Data Sheet, DS 1, Oct. 2001

SIUC-BA
Single Chip ISDN USB Controller - Basic
PSB 2155 Version 1.3

Wired Communications

Infineon technologies

Never stop thinking.

Edition 2001-10-01

Published by Infineon Technologies AG,
St.-Martin-Strasse 53,
D-81541 München, Germany

Attention please!

The information herein is given to describe certain components and shall not be considered as warranted characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Infineon Technologies is an approved CECC manufacturer.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office in Germany or our Infineon Technologies Representatives worldwide (see address list).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Data Sheet, DS 1, Oct. 2001

SIUC-BA

Single Chip ISDN USB Controller - Basic

PSB 2155 Version 1.3

Wired Communications

Infineon technologies

Never stop thinking.

Revision History:	**2001-10-01**	DS 1
Previous Version:	None	

Page	Subjects (major changes since last revision)

For questions on technology, delivery and prices please contact the Infineon Technologies Offices in Germany or the Infineon Technologies Companies and Representatives worldwide: see our webpage at http://www.infineon.com

Table of Contents

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

Table of Contents **Page**

List of Figures

List of Figures

List of Figures

List of Figures

List of Tables

Preface

The Single Chip ISDN USB Controller - Basic (SIUC-BA) is an optimized low cost solution for host based connectivity to ISDN through USB. This document provides reference information on the features and possible applications.

Organization of this Document

This Data Sheet is divided into 11 chapters. It is organized as follows:

- Chapter 1, Overview
  Gives a general description of the product, lists the key features, describes functional modules and presents some typical applications.
- Chapter 2, Pin Description
  Lists pins with associated signals, categorizes signals according to function, and describes signals.
- Chapter 3, C800 Microcontroller
  Describes the embedded C800 8-bit microcontroller, memory organisation and special function registers.
- Chapter 4, USB Module
  Covers the USB implementation on the SIUC-BA describing the transfer modes, the memory buffer operation, the USB device framework and the USB registers.
- Chapter 5, ISDN Module
  Includes all the modules, modes and interfaces related to ISDN, especially the IOM-2 interface, the S-transceiver and the HDLC Controllers.
- Chapter 6, Interrupt System
  Describes the microcontroller interrupt sources along with all interrupt status and interrupt enable registers, and describes interrupt priority handling.
- Chapter 7, Firmware
  Describes the way of running firmware, the memory map, and explains the different USB models used with the SIUC-BA.
- Chapter 8, General Features
  Includes the clock and reset generation within the device, the auxiliary and SPI interfaces, and the voltage regulator.
- Chapter 9, Operational Description
  Briefly elaborates the operational modes like active, idle, suspend etc. within the device along with the programming sequence to be followed in different modes.
- Chapter 10, Electrical Characteristics
- Chapter 11, Package Outlines

Related Documents

- USB Specification V1.1, September 23, 1998
- USB Class Definitions for Communication Devices (CDC) V1.1, January 19, 1999
- USB Device Class Specification for Device Firmware Upgrade (DFU) V1.0, May 13, 1999
- ISDN PC Adapter Circuit (IPAC-X) PSB 21150, Version 1.1, Preliminary Data Sheet, 12.99
- C540U / C541U 8-Bit CMOS Microcontroller, User's Manual 11.97.
- C501 8-Bit Single-Chip Microcontroller, User's Manual 04.97
- Embedded C165 with USB, ISDN Terminal adapter and HDLC, UTAH, Data Sheet.

1 Overview

The Single Chip ISDN USB Controller - Basic (SIUC-BA) integrates all functions on a single chip for a host based ISDN S-interface access solution through USB. It is a derivate with reduced functionality of the full feature device SIUC-X PSB 2154, which combines the basic features of the ISDN PC Adapter Circuit (IPAC PSB 2115) and the C541U 8-bit Microcontroller with USB, but in 3.3 V technology. It is dedicated for pure single chip designs without memory extension and peripheral devices.

On the ISDN side, it includes the S-transceiver (Layer 1), an HDLC controller for the D-channel and two protocol controllers for both B-channels (Layer 2). They can be used for HDLC protocol or transparent access. The FIFO size of the B-channel buffers is 128 bytes per channel and per direction.

On the USB side, it includes a full speed USB transceiver, supports bus powered operation and is compliant with USB Specification V1.1 and the Communication Device Class (CDC) Specification V1.1 for ISDN devices. The endpoints can be controlled by the microcontroller by special function registers. A boot loader in the internal ROM allows firmware download to the internal memory via USB according to the USB Device Class Specification for Device Firmware Upgrade (DFU) V1.0.

The embedded new C800 Microcontroller core (8-bit) enables transparent or HDLC-framed exchange of B-channel data between the S-interface and USB. In addition, it provides lower level D-channel access control functions. The operational firmware which is downloaded via USB is contained in internal 16K RAM.

Firmware can be developed with the full feature version SIUC-X PSB 2154 using internal memory (single chip mode) or using additional external memory (shared or separate memory mode), whereby emulation is supported through Enhanced Hooks Technology™. Emulation is not supported by the SIUC-BA.

The programmable I/O lines of the 8-bit auxiliary port may be used for external control functions and to forward status information to the µC. The SPI interface for serial EEPROM communication is multiplexed onto these lines. 3 programmable LED output ports are available, one of them can indicate the activation status of the S-interface automatically.

The onchip voltage regulator supports the design of bus powered applications, i.e. the regulated 3.3V supply is generated out of the USB supply by the SIUC-BA itself without requiring an external voltage regulator.

The SIUC-BA is produced in advanced CMOS technology.

Single Chip ISDN USB Controller - Basic SIUC-BA

PSB 2155

Version 1.3

1.1 Features

P-MQFP-80-1

General

- Single chip host based ISDN solution for USB
- Derivate of SIUC-X with reduced features
- 3.3V power supply
- Programmable reset sources
- Onchip PLL for 48 MHz clock generation
- 5V tolerant I/Os
- Onchip voltage regulator for bus-powered operation (patent pending)

ISDN (S-Interface, 2B+D Channels)

- S/T-transceiver (ITU-T I.430) operating in TE mode
- D-channel and B-channel protocol controllers (HDLC)
- Different types of protocol support depending on operating mode (Non-auto mode, Transparent mode 1-3, extended transparent mode)
- IOM-2 interface, single/double clock with strobe signal
- Monitor and C/I-channel protocol to control peripheral devices
- 128 byte FIFO buffers with programmable FIFO thresholds per B-channel per direction
- 64 byte FIFO buffers per direction with programmable FIFO thresholds for D-channel
- D-channel access mechanism
- Transformer ratio 1:1
- 2 timers programmable between 1 ms to 14.336 s.

Firmware

- SIUC-BA comes along with firmware and drivers fully supporting ISDN data access according to USB CDC V1.1

Type	Package
PSB 2155	P-MQFP-80-1

- Onchip bootloader supports DFU class so firmware can easily be downloaded via USB to internal memory (flexibility for firmware upgrades)
- Customer can develop own firmware using internal memory

Microcontroller & Peripherals

- 8-bit C800 CPU, full software/toolset compatible to standard 80C51/80C52 microcontrollers
- 48 MHz operating frequency, equivalent to 4 MIPS
- 4 Kbyte onchip standard ROM program memory (boot loader)
- 256 byte onchip data RAM
- 16 Kbyte onchip RAM (XRAM) flexibly programmable as program and/or data space
- Two lines can be used as external interrupt source, one of them can indicate the USB device attached status in self powered mode
- 14 interrupt sources to CPU selectable at 4 priority levels
- 8 data pointer
- Two 16-bit timers: timer 0 and timer 1

USB

- Compliant to
  - USB Specification V1.1
  - USB Communication Device Class (CDC) Specification V1.1
  - USB Device Firmware Upgrade (DFU) Specification V1.0
- Onchip USB transceiver
- 12 Mbit/s full speed operation
- 7 software configurable endpoints, in addition to the bi-directional Control Endpoint 0
- Firmware supports 2 configurations by default:
  - USB Device Firmware Upgrade (DFU)
  - USB Communication Device Class (CDC)
- DFU Configuration with 1 interface and 1 endpoint (EP0)
- CDC Configuration with 4 additional interfaces and 2 alternate settings each, supporting 8 endpoints (EP0 - EP7)
- All USB transfer modes supported (bulk, isochronous, interrupt and control)
- Bus-powered operation possible (no external power supply necessary)
- Low Power Device, <100mA (operational), <500 µA (suspend)
- Optional loading of customized configuration data (e.g. Vendor ID, Product ID, ...) from external serial EEPROM

Miscellaneous

- 8 line programmable auxiliary I/O interface with interrupt inputs
- SPI Interface for optional connection to an external EEPROM
- 3 LED output ports (one is capable to indicate S-bus activation status automatically)
- Strap pins for identification of different HW configurations

1.2 Logic Symbol

The logic symbol shows all functions of the SIUC-BA. It must be noted, that not all functions are available simultaneously, but depend on the selected mode.

Pins which are marked with a " * " are multiplexed and not available in all modes.

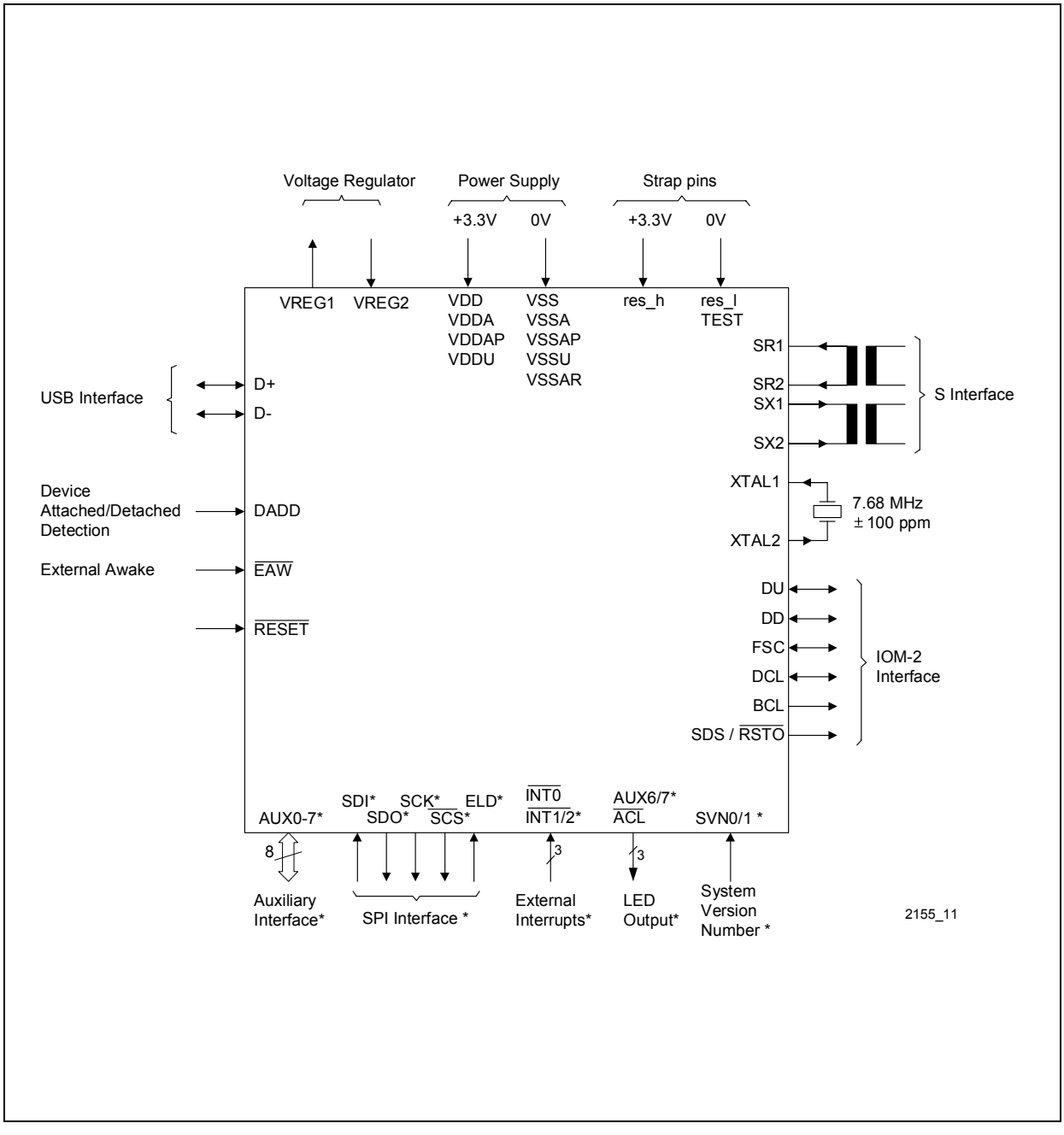


Figure 1 Logic Symbol

1.3 Typical Application

The SIUC-BA is dedicated to USB host based, single chip applications. The S interface is a 4-line 192 kbit/s interface while the 2-line USB interface works at 12 MHz.

An ISDN adapter for a PC is built around the SIUC-BA using the USB interface (**Figure 2**). The onchip voltage regulator allows bus powered operation without the necessity of an external regulator. 3 LED ports can be used to indicate different status information to the user.

This single chip solution enables cost optimized design of an embedded device cable linking USB to ISDN ("cable with a bump").

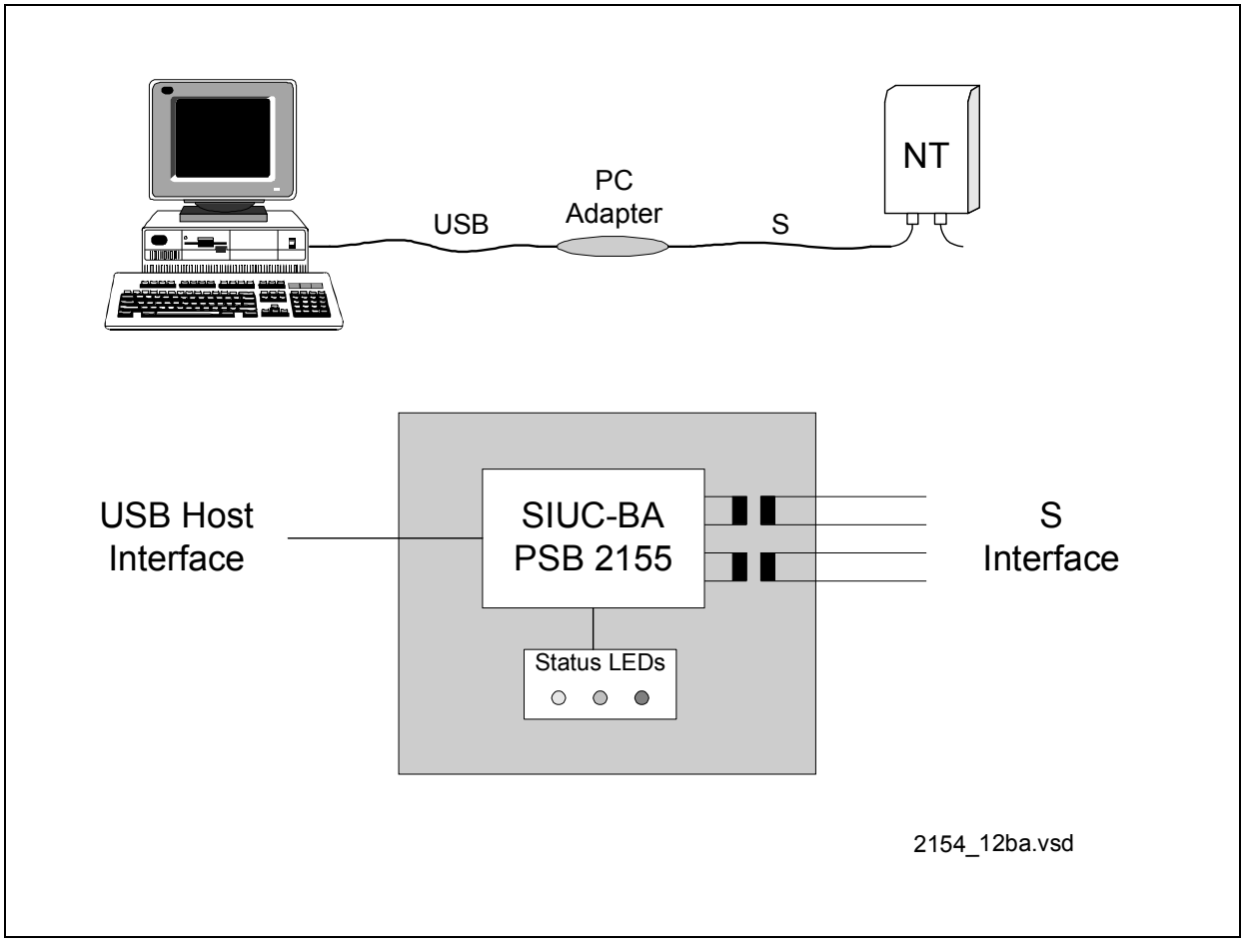


Figure 2 ISDN PC Adapter for S Interface

1.4 Functional Description

The data transfer from USB to S-Interface and vice versa takes place through the µC XRAM and a set of FIFOs for each channel. The IOM-2 interface allows connection of other communication devices.

The auxiliary interface serves as a general purpose I/O port in addition to providing LED drivers and lines for the SPI interface. Two auxiliary pins are used to latch a number for vendor specific system identification during reset.

A bootloader firmware contained in 4K ROM provides the ability to download operational firmware via USB to internal RAM. All registers corresponding to different peripherals are available in the Special Function Register (SFR) map of the µC. The ISDN specific registers are located in the external memory map of the µC.

Onchip PLL provide clock generation for the S-transceiver and the 48 MHz USB clock. An onchip voltage regulator allows the flexibility for bus and self powered system architectures.

1.4.1 Functional Block Diagram

Figure 3 gives an overview of the functional blocks.

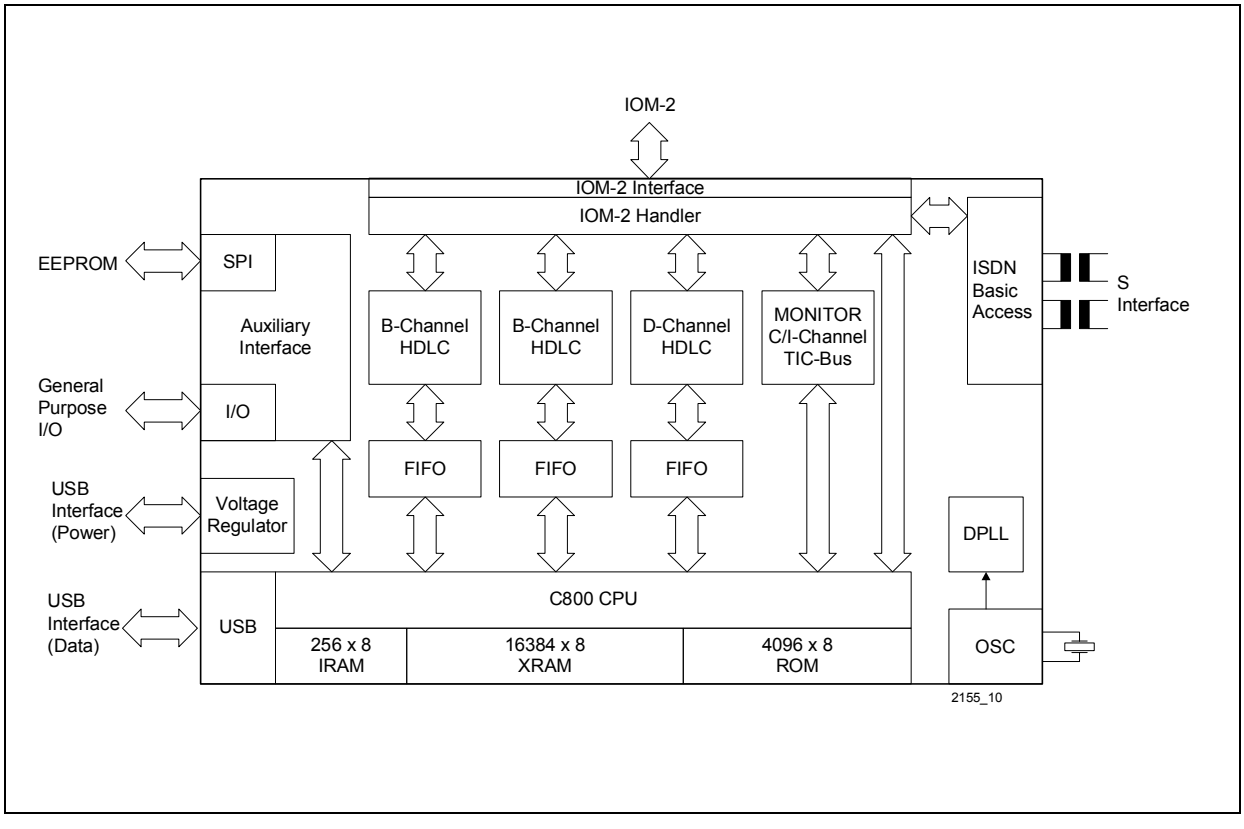


Figure 3 Block Diagram

2 Pin Description

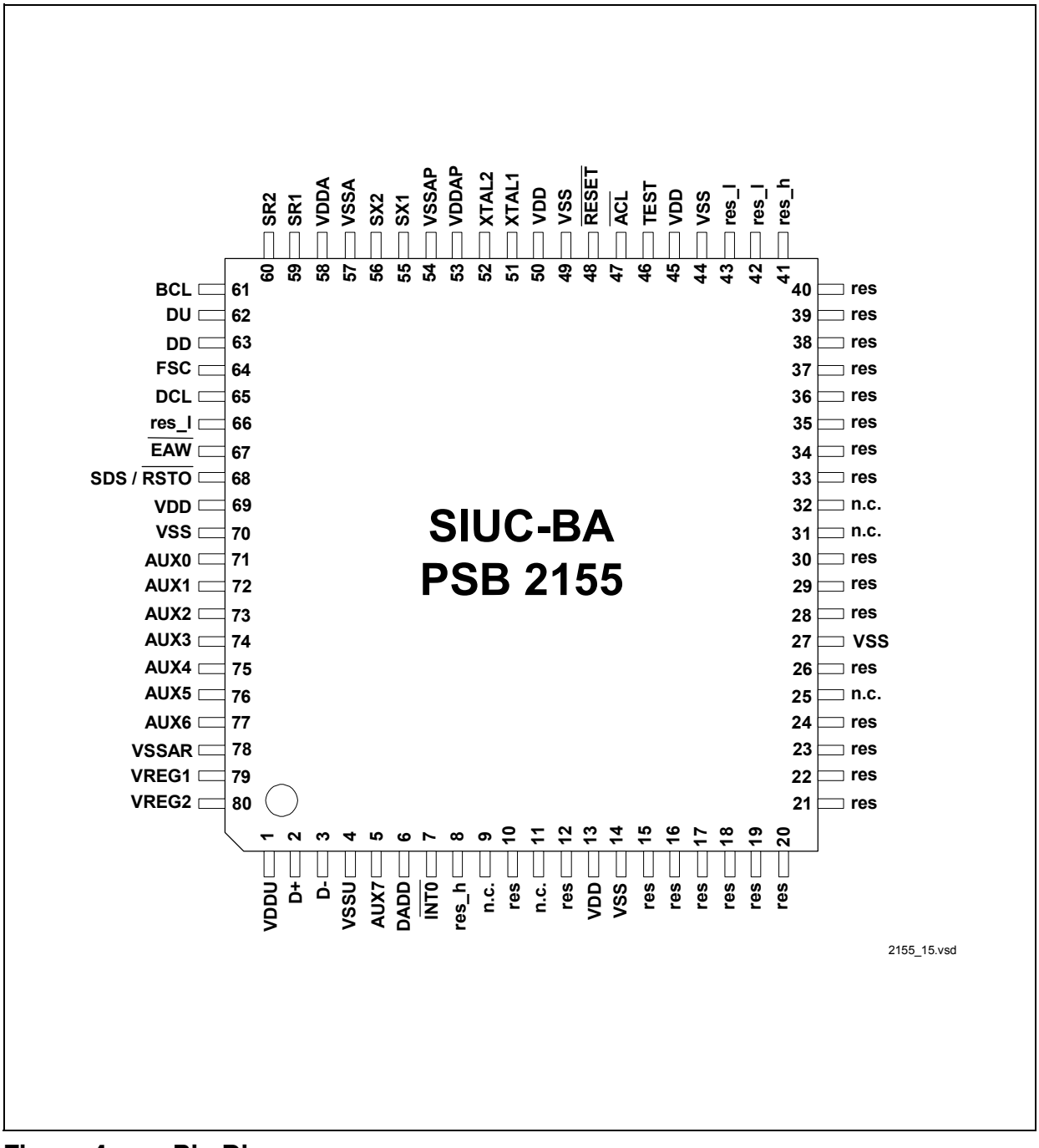


Figure 4 Pin Diagram

Table 1 Pin Definition - IOM-2 Interface

Pin No.	Symbol	Input (I) Output (O)	Function
64	FSC	O	**Frame Sync** 8-kHz frame synchronisation signal. The rising edge indicates the beginning of the IOM frame (HIGH during channel 0).
65	DCL	O	**Data Clock** IOM clock signal of twice the IOM data rate (1.536 MHz). The first rising edge is used to transmit data, the second falling edge is used to sample data.
62	DU	I	**Data Upstream** IOM data signal in upstream direction.
63	DD	O(OD)	**Data Downstream** IOM data signal in downstream direction.
61	BCL	O	**Bit Clock** Bit clock output, identical to IOM data rate , derived from the DCL output clock (BCL = DCL/2 = 768 kHz).
68	SDS / $\overline{\text{RSTO}}$	O (OD)	**Serial Data Strobe / Reset Output** Programmable strobe signal (push pull characteristic) for time slot and/or D-channel indication on IOM-2. It can optionally be used as reset output (open drain characteristic).

Table 2 Pin Definition - Auxiliary Interface

Pin No.	Symbol	Input (I) Output (O)	Function
71 72 73	AUX0 AUX1 AUX2	I/O	**Auxiliary Port 0-2 - General input/output Ports** These pins are individually programmable as general input/output. The state of the pin can be read from (input) / written to (output) a register.
74	AUX3	I/O	**Auxiliary Port 3** **Non SPI Mode: AUX3 (input/output)** If not used for the SPI interface, this pin is programmable as general input/output. The state of the pin can be read from (input) / written to (output) a register. **SPI Mode: ELD (input, during reset) - EEPROM Load** This pin is strapped HIGH during reset to indicate the µC that an EEPROM is connected (e.g. for loading USB ID values). **SPI Mode: $\overline{\text{SCS}}$ (output, after reset)** Serial Chip Select to EEPROM. This pin has an internal pulldown resistor, i.e. for the ELD function an external pullup resistor must be connected to indicate that an EEPROM is connected.

Table 2 Pin Definition - Auxiliary Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
75	AUX4	I/O	**Auxiliary Port 4** **Non SPI Mode: AUX4 (input/output)** If not used for the SPI interface, this pin is programmable as general input/output. The state of the pin can be read from (input) / written to (output) a register. **SPI Mode: SDI (input)** Serial Data Input on the SPI interface to be connected to the SO pin of the EEPROM. **MBIT - Multiframe Synchronization (output)** If selected via ACFG2.A4SEL=1 the pin AUX4 is used for multiframe synchronization, i.e. it is an M-bit output. **All modes: SVN0 - System Version Number 0 (input; during reset only)** During reset the state of this pin (pull up/down resistor) is latched to the internal System Version Number register. After reset this pin performs the functions described above. An internal pull down resistor is provided.
76	AUX5	I/O	**Auxiliary Port 5** **Non SPI Mode: AUX5 (input/output)** If not used for the SPI interface, this pin is programmable as general input/output. The state of the pin can be read from (input) / written to (output) a register. **SPI Mode: SDO (output)** Serial Data Output on the SPI interface to be connected to the SI pin of the EEPROM. **All modes: SVN1 - System Version Number 1 (input)** During reset the state of this pin (pull up/down resistor) is latched to the internal System Version Number register. After reset this pin performs the functions described above. An internal pull down resistor is provided.

Table 2 Pin Definition - Auxiliary Interface (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
77	AUX6	I/O	**Auxiliary Port 6** **Non SPI Mode: AUX6 (input/output), $\overline{\text{INT1}}$** If not used for the SPI interface, this pin is programmable as general input/output. The state of the pin can be read from (input) / written to (output) a register. In addition to that, as an input, it can generate an interrupt (ISTAA:INT1) which is maskable in MASKA: INT1. The interrupt input is either edge or level triggered (ACFG2:EL1). As an output it is able to sink higher current and so allows for direct connection of an LED in stand-alone applications. An internal pullup resistor is connected to this pin. **SPI Mode: SCK (output)** Serial Clock output to EEPROM
5	AUX7	I/O	**Auxiliary Port 7** **AUX7 (input/output), $\overline{\text{INT2}}$** This pin is programmable as general input/output. The state of the pin can be read from (input) / written to (output) a register. In addition to that, as an input, it can generate an interrupt (ISTAA:INT2) which is maskable in MASKA:INT2. The interrupt input is either edge or level triggered (ACFG2:EL2). As an output it is able to sink higher current and so allows for direct connection of an LED in stand-alone applications. An internal pullup resistor is connected to this pin.

Table 3 Pin Definition - Miscellaneous

Pin No.	Symbol	Input (I) Output (O)	Function
2	D+	I/O	**USB D+ Data Line** The pin D+ can directly be connected to the USB cable (Transceiver is integrated onchip). A pull up resistor (1.5K ± 5%) must be connected to D+ to select full speed operation according to the USB spec.
3	D-	I/O	**USB D- Data Line** The pin D- can directly be connected to the USB cable. (Transceiver is integrated onchip).
48	$\overline{\text{RESET}}$	I	**Reset** A LOW on this input forces the SIUC-BA into a reset state. The duration of this pulse must be at least 4 ms to stabilize the internal oscillator. Following the reset, the microcontroller executes a complete machine cycle to initialize indirectly resetable registers.
67	$\overline{\text{EAW}}$	I	**External Awake** If a negative level on this input is detected, the SIUC-BA generates an interrupt (AUXI.EAW), and if enabled, a reset pulse.
47	$\overline{\text{ACL}}$	O	**Activation LED** This pin can either function as a programmable output or automatically indicate the activated state of the S interface by a logic 0. An LED with pre-resistance may directly be connected to $\overline{\text{ACL}}$.
6	DADD	I	**Device Attached input** For self powered applications this pin together with some external logic can be used to detect whether the device is connected to the USB or not.
7	$\overline{\text{INT0}}$	I	**External Interrupt 0 input** This interrupt input can be used to indicate external events to the µC.
55 56	SX1 SX2	O O	**S-Bus Transmitter Output** Differential output for the S-transmitter. positive negative

Table 3 Pin Definition - Miscellaneous (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
59 60	SR1 SR2	I I	**S-Bus Receiver Input** Differential inputs for the S-receiver.
51	XTAL1	I	**Oscillator Input** Input pin of oscillator or input from external clock source. 7.68 MHz crystal or clock required.
52	XTAL2	O	**Oscillator Output** Output pin of oscillator. Not connected if external clock source is used.
46	TEST	I	**TEST** This pin is reserved for test purposes during manufacturing and should be connected to GND.
10, 12, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 28, 29, 30, 33, 34, 35, 36, 37, 38, 39, 40,	res	I	**reserved** These pins are reserved and must be left not connected.
42, 43, 66	res_l	I	**reserved, connect LOW** These pins are reserved and must be connected to VSS.
8, 41	res_h	I	**reserved, connect HIGH** These pins are reserved and must be connected to VDD.
9, 11, 25, 31, 32	n.c.		**not connected** These pins are not connected internally and can be connected in any way.

Table 4 Pin Definition - Power Supply

Pin No.	Symbol	Input (I) Output (O)	Function
13, 45, 50, 69	VDD	I	**Digital Supply Voltage,** +3.3V for core logic and oscillator
58	VDDA	I	**Analog Supply Voltage,** +3.3V for S-transceiver

Table 4 Pin Definition - Power Supply (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
53	VDDAP	I	**Analog Supply Voltage,** +3.3V for PLL
1	VDDU	I	**Analog Supply Voltage,** +3.3V for USB module
14, 27, 44, 49, 70	VSS	I	**Digital GND**, for core logic and oscillator
57	VSSA	I	**Analog GND**, for S-transceiver
54	VSSAP	I	**Analog GND**, for PLL
4	VSSU	I	**Analog GND**, for USB module
79 80	VREG1 VREG2		**Voltage Regulator** These two pins from the internal voltage regulator are used to connect some additional external components for regulation. The regulator uses the USB power supply (bus-powered mode) to generate the +3.3V supply for the SIUC-BA which must externally be connected to the VDDx pins (the supply is not connected internally). If the voltage regulator is not used (e.g. USB self-powered mode) VREG1/2 are left not connected and the external power supply is connected to the VDDx pins.
78	VSSAR	I	**Analog GND**, for Voltage Regulator

Note: Some of the pins are used to latch certain values during reset. These are ALE/$\overline{CS}$, AUX3/ELD, AUX4/SVN0 and AUX5/SVN1. The values on these pins must be held stable for at least 600 ns after reset.

2.1 Pin States in Operating Modes

The following table provides an overview on the behaviour of the pins and ports in the different operating modes. In normal operating mode most of the pins can be used in different ways therefore the behaviour in operational mode is not listed.

The following abbreviations are used:

I input pin
this pin needs to be terminated externally if no internal pull up/down resistor is available or if the internal resistor is switched off, to avoid malfunctions due to floating input.

O output pin
this pin keeps the level (“0” or “1”) which was driven just before suspend mode was entered.

Z high impedant
this pin is an output pin but the output driver is switched off (floating), therefore this pin does not need to be terminated but can be left open.

Note: "Pull up" at any place in the tables below refers to internal pull up resistors only and not to any external resistors.
Internal resistors are switched off in suspend and idle mode to avoid an increased leakage current.
Some pins can show different characteristic depending on the mode which was used before suspend mode was entered.

Table 5 Pin Definition - IOM-2 Interface

Pin No.	Symbol	During Reset	In Suspend Mode	In Idle Mode	Comment
64	FSC	O	O or I	O or I	Note 1
65	DCL	O	O or I	O or I	
62	DU	Z	O or I	O or I	Note 2
63	DD	Z	O or I	O or I	
61	BCL	O	O	O	
68	SDS / $\overline{\text{RSTO}}$	O	Z or O	Z or O	Note 3

Table 6 Pin Definition - Auxiliary Interface

Pin No.	Symbol	During Reset	In Suspend Mode	In Idle Mode	Comment
71 72 73	AUX0 AUX1 AUX2	I I I	I or O I or O I or O	I or O I or O I or O	Note 4
74	AUX3	I	I or O	I or O	
75	AUX4	I	I or O	I or O	
76	AUX5	I	I or O	I or O	
77	AUX6	I (with pull up)	I or O (w/o pull up)	I or O (w/o pull up)	
5	AUX7	I (with pull up)	I or O (w/o pull up)	I or O (w/o pull up)	

Table 7 Pin Definition - Miscellaneous

Pin No.	Symbol	During Reset	In Suspend Mode	In Idle Mode	Comment
41	MMOD	I	I	I	
8	$\overline{EA}$	I	I	I	Note 5
6	DADD	I	O or I	O or I	
7	$\overline{INT0}$	I	O or I	O or I	
2	D+	I	I	I	
3	D-	I	I	I	
48	$\overline{RESET}$	I (with pull up)	I (w/o pull up)	I (with pull up)	
43 42	BMOD0 BMOD1	I I	I I	I I	
67	$\overline{EAW}$	I	I	I	
47	$\overline{ACL}$	O	O	O	
55 56	SX1 SX2	Z Z	Z Z	Z Z	
59 60	SR1 SR2	I I	I I	I I	
51	XTAL1	I	I	I	
52	XTAL2	O	O	O	

Table 7 Pin Definition - Miscellaneous (cont'd)

Pin No.	Symbol	During Reset	In Suspend Mode	In Idle Mode	Comment
46	TEST	I	I	I	
66	res_I	I	I	I	

Note: 1) In normal mode (i.e. the transceiver is not switched off) FSC and DCL become output in suspend/idle mode. However, if the remote wakeup feature from the S transceiver should be disabled, the transceiver must be switched off (TR_CONF0.DIS_TR = 1) which has the effect that FSC and DCL become input and therefore must be terminated externally (pull up resistors).

2) If DU and DD are programmed to open drain characteristic (default mode) they become input ports during suspend/idle mode. If they are programmed to push pull drivers (IOM_CR.DIS_OD=1), they become output and so no terminating resistor is required.

3) The behaviour of pin SDS/$\overline{\text{RSTO}}$ in suspend/idle mode depends on its configuration in operational mode before suspend was entered. If the pin was used as $\overline{\text{RSTO}}$ then it will be "Z" in suspend/idle mode, if it was used as SDS then it keeps the output characteristic is suspend mode.

4) All pins from the auxiliary port keep their selected direction (input/output) and their programmed level as outputs ("0" or "1") when going to suspend/idle mode.

5) This pin is directly connected to VDD on the system board.

Important Note: To avoid malfunctions on unused signals the following register programming must be done right after reset:
write 03_H to register address 80_H
write 00_H to register address $A0_H$
write 00_H to register address $B0_H$

3 C800 Microcontroller

3.1 CPU

The CPU is designed to operate on bits and bytes. The instructions, which consist of up to 3 bytes, are performed in one, two or four machine cycles. One machine cycle requires twelve microcontroller clock cycles. The instruction set has extensive facilities for data transfer, logic and arithmetic instructions. The boolean processor has its own full-featured and bit-based instructions within the instruction set. Efficient use of program memory results from an instruction set consisting of 44% one-byte, 41% two-byte, and 15% three-byte instructions.

A machine cycle (12 microcontroller clocks) consists of 6 states. Each state is divided into a phase 1 half, during which the phase 1 clock is active, and a phase 2 half, during which the phase 2 clock is active. Typically, arithmetic and logic operations take place during phase 1 and internal register-to-register transfers take place during phase 2. The diagrams in **Figure 5** show the fetch/execute timing related to the internal states and phases.

Execution of a one-cycle instruction begins at S1P2, when the op-code is latched into the instruction register. If it is a 2-byte instruction, the second reading takes place during S4 of the same machine cycle. If it is a one-byte instruction, there is still a fetch at S4, but the byte read (which would be the next op-code) is ignored (discarded fetch), and the program counter is not incremented. In any case, execution is completed at the end of S6P2.

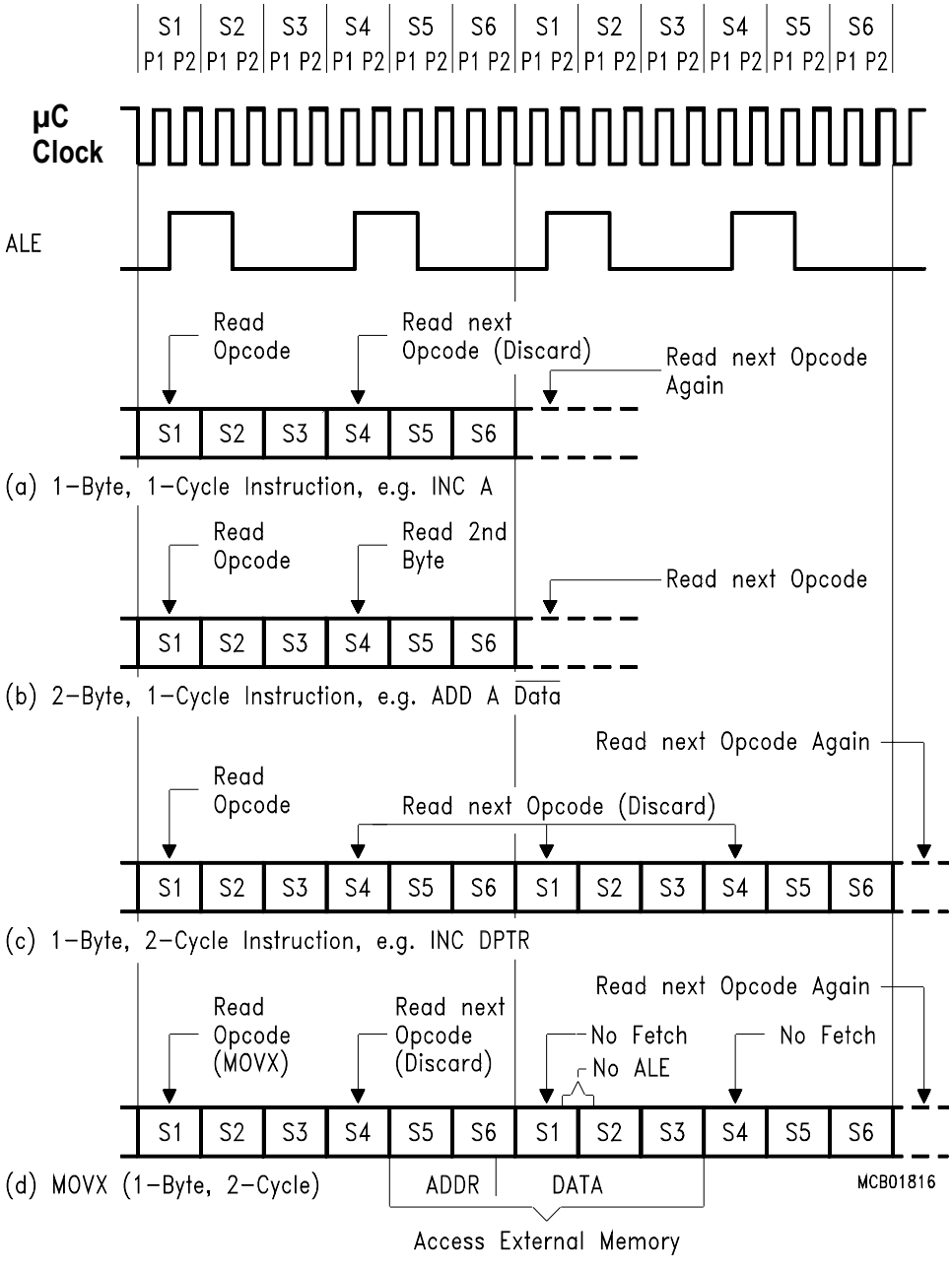
Most C800 instructions are executed in one cycle. MUL (multiply) and DIV (divide) are the only instructions that take more than 2 cycles to complete: they take 4 cycles. Normally, 2 code bytes are fetched from the program memory every machine cycle. The only exception is when a MOVX instruction is executed. MOVX is a one-byte, 2-cycle instruction that accesses external data memory. During a MOVX, the two fetches in the second cycle are skipped while the external data memory is being addressed and strobed.

Fetches from external program memory always use a 16-bit address. Accesses to external data memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri). “External” in this context refers to the CPU-module (XRAM is internal, additional onchip RAM is external) and not to the chip.
For 8-bit addressing the XRAM page register (XPAGE) is used as the high byte, which allows effective addressing in critical loops:

Loop using 16-bit address:	movx a, @dptr	(2 cycles)
	inc dptr	(2 cycles)
	loop_based_on_some_condition	

Loop using 8-bit address:

```
movx a, @Ri                    (2 cycles)
inc Ri                         (1 cycle)
loop_based_on_some_condition
```



Note: The signal ALE is only available in SIUC-X but not in SIUC-BA. It is shown here to allow complete understanding of CPU functionaliy.

Figure 5 Fetch Execute Sequence

3.2 Memory Organisation

The C800 CPU manipulates operands in the following address spaces:

– 4 Kbyte onchip ROM program memory (boot loader); see note
– 16 Kbyte onchip RAM (XRAM) memory for program and/or data (configurable)
– 256 bytes of internal data memory (IRAM)
– 128 byte special function register area

Figure 6 illustrates the memory address spaces of the C800.

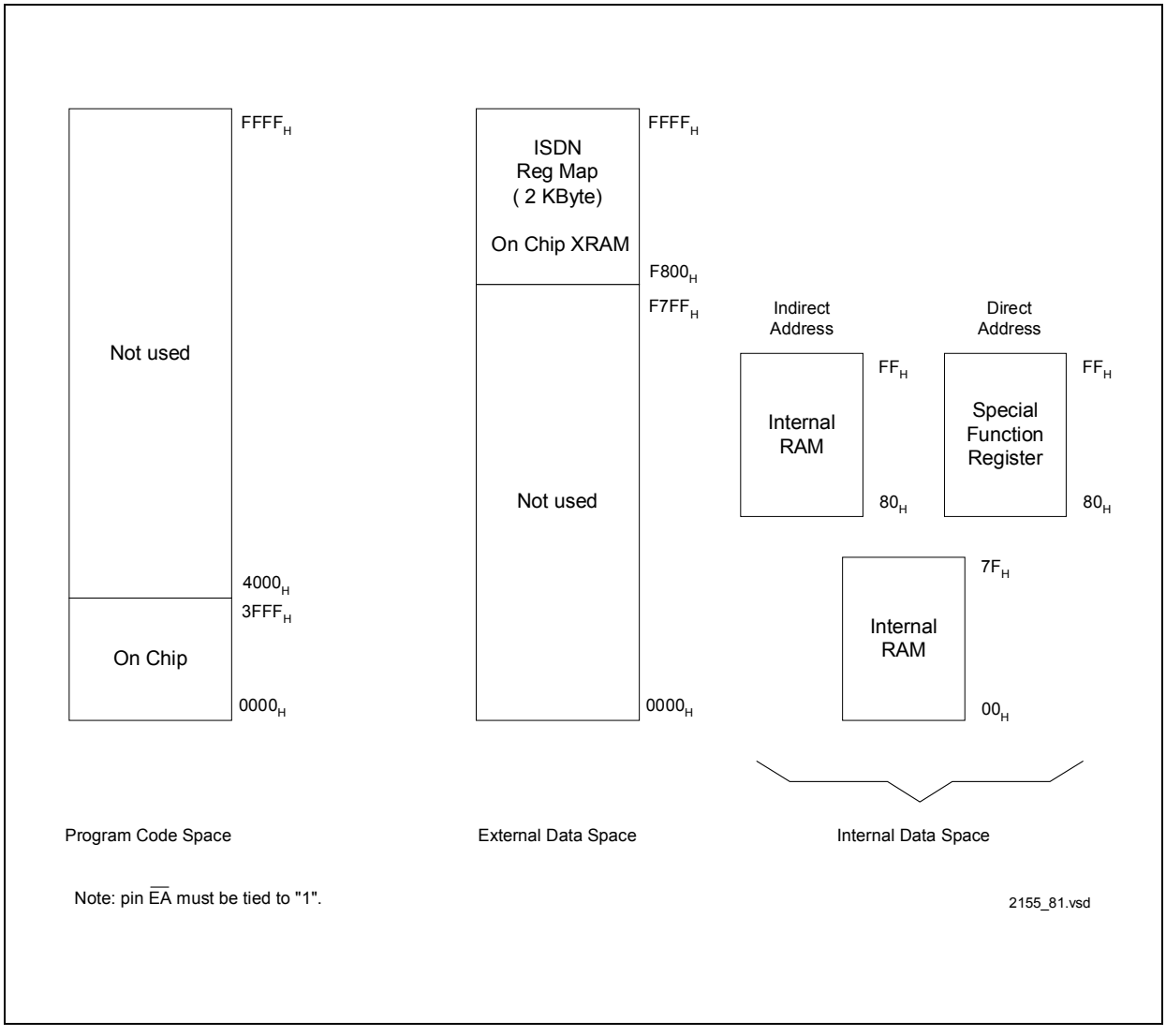


Figure 6 C800 Default Memory Map (Firmware Execution Mode)

Note: For simplification the 4K onchip ROM is not shown in **Figure 6** but described in detail in **Figure 7**.

3.2.1 Program Memory

The C800 has 4 Kbyte of ROM program memory which stores the download routines (boot loader). The firmware can be downloaded into onchip program RAM. If the $\overline{EA}$ pin is held high, the C800 executes program code out of internal program memory .

Important Note: Strapping $\overline{EA}$ to low must not be done for SIUC-BA.

Note: For further details on memory configuration please refer to **Chapter 7.3**.

3.2.2 Data Memory

The data memory address space consists of core-internal memory and core-external memory space (both are onchip). The internal data memory is divided into 3 physically separate and distinct blocks: the lower 128 bytes of RAM, the upper 128 bytes of RAM and the 128 byte special function register (SFR) area. The external data memory is divided into onchip XRAM and another 2 Kbyte of address space reserved for ISDN registers.

Note: The registers of the USB module are accessed through special function registers in the SFR area.

Note: After reset the access to onchip XRAM is disabled, i.e. the µC must set bit SYSCON1.XMAP0=1 in order to enable internal XRAM access, otherwise all accesses to these locations are undefined. The bootloader running on SIUC-BA is taking care of this.

3.2.3 General Purpose Registers - Overview

The lower 32 locations of the internal RAM (data memory addresses from 00 to 1F) are assigned to 4 banks with 8 general purpose registers (GPRs) each. Only one of these banks may be enabled at a time. 2 bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank. This allows fast context switching, which is useful when entering subroutines or interrupt service routines.

The 8 general purpose registers of the selected register bank may be accessed by register addressing. With register addressing, the instruction opcode indicates which register is to be used. For indirect addressing, R0 and R1 are used as pointer or index registers to address internal or external memory (e.g. MOV @R0).

Reset initializes the stack pointer to location 07 and increments it once to start from location 08, which is also the first register (R0) of register bank 1. Thus, if one is going to use more than one register bank, the SP should be initialized to a different location of the RAM which is not used for data storage.

16 bytes of data memory (addresses 20 to 2F) are bit-addressable. Direct addresses from 30 to 7F can be used as scratch pad registers or for a stack.

3.2.4 Partitioning of RAM, Switching from ROM to RAM

In download mode the bootloader contained in ROM performs the download of the firmware to 16K internal RAM (**Figure 7**) which is used as data RAM where the µC writes the downloaded data to.

After the download is finished the µC first partitions the internal RAM for program and data memory by writing the PSIZ and DSIZ registers. Then it sets the bit SYSCON2.STAT2=1 to switch from ROM to RAM and with a successive register access setting SYSCON2.STAT1=1 a reset to the µC is performed which has the effect that the program counter starts execution in RAM at address 0000_H.

Note: **Figure 7** shows a memory partitioning of 6K Program and 10K Data RAM as an example. Any other partitioning with a granularity of 1K is possible.

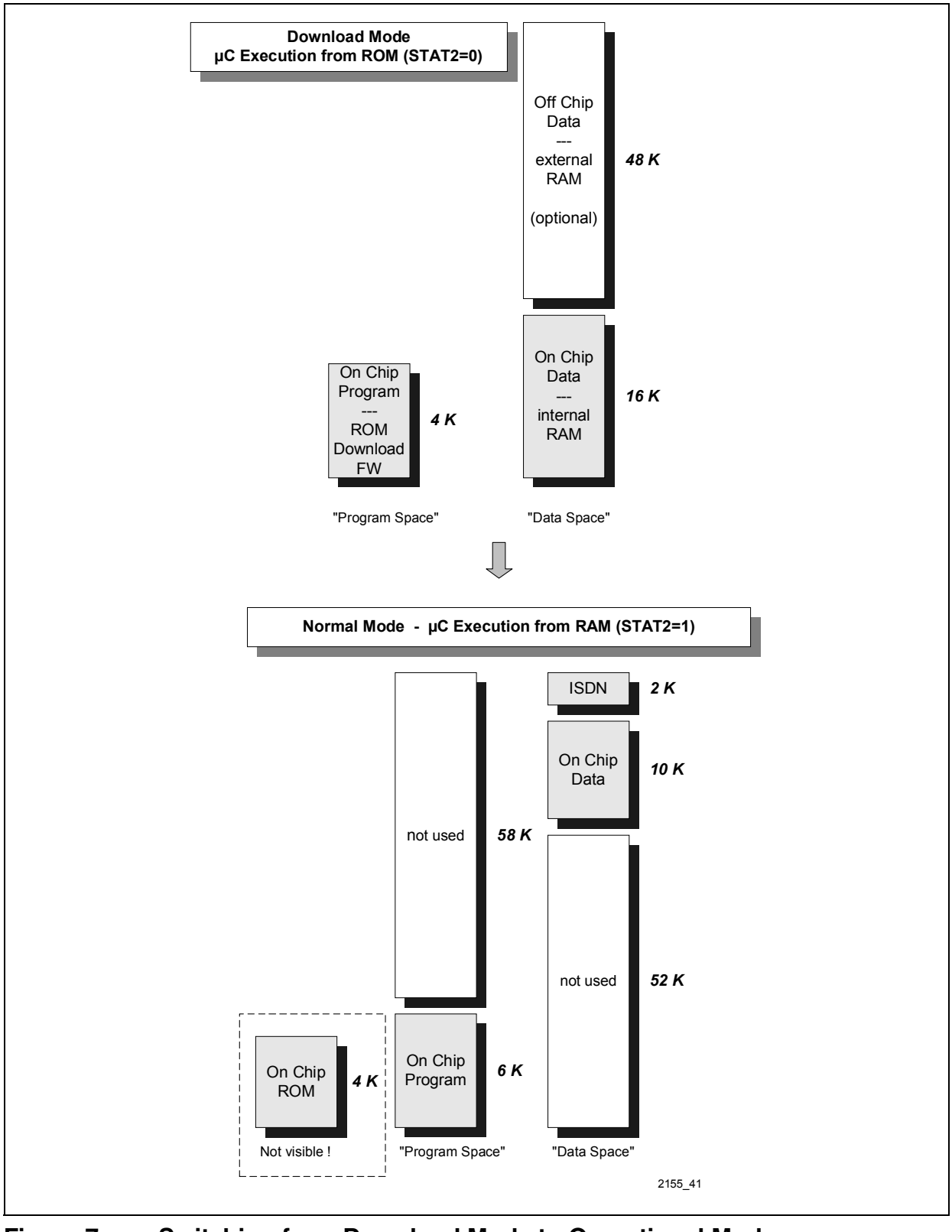


Figure 7 Switching from Download Mode to Operational Mode

3.2.5 Special Function Registers - Overview

The registers, except the program counter, the four general purpose register banks and the ISDN registers, reside in the special function register (SFR) area. All SFRs with addresses where address bits 0-2 are 0 (e.g. 80_H, 88_H, 90_H, 98_H,..., $F8_H$) are bit-addressable.

The SFRs include pointers and registers that provide an interface between the CPU and other onchip peripherals. The SFRs are listed in **Table 8** and **Table 9**.
In **Table 8** they are organised in functional groups.
Table 9 illustrates the contents of the SFRs in numeric order of their addresses.

Table 8 Special Function Registers - Functional Blocks

Block	Symbol	Name	Addr	Contents after Reset	Page No.	Bit Addr
CPU	SP	Stack Pointer	81	07	38	no
	DPL	Data Pointer, Low Byte	82	00	38	no
	DPH	Data Pointer, High Byte	83	00	38	no
	DPSEL	Data Pointer Selector	84	00	39	no
	PSW	Program Status Word	D0	00	42	yes
	ACC	Accumulator	E0	00	47	yes
	B	B Register	F0	00	47	yes
	PSIZ	Program RAM Size	AB	06	48	no
	DSIZ	Data RAM Size	BC	0A	49	no
	SYSCON1	System Control 1	AD	21	50	no
	SYSCON2	System Control 2	A4	00	51	no
	XPAGE	XRAM Page	AE	00	52	no
Interrupt System	IEN0	Interrupt Enable 0	A8	00	283	yes
	IEN1	Interrupt Enable 1	A9	00	284	no
	IEN2	Interrupt Enable 2	AA	00	284	no
	IP0	Interrupt Priority 0	B8	00	290	yes
	IP1	Interrupt Priority 1	AC	00	290	no
Power	PCON	Power Control	87	00	41	no
	WCON	WakeUp Control	8E	00	43	no
Timers	TCON	Timer Control	88	00	36	yes
	TMOD	Timer Mode	89	00	37	no
	TL0	Timer 0, Low Byte	8A	00	35	no
	TL1	Timer 1, Low Byte	8B	00	35	no
	TH0	Timer 0, High Byte	8C	00	35	no
	TH1	Timer 1, High Byte	8D	00	35	no

Table 8 Special Function Registers - Functional Blocks (cont'd)

Block	Symbol	Name	Addr	Contents after Reset	Page No.	Bit Addr
SPI	EEPINT	EEPROM Interrupt Control	93	00	276	no
	EEPCMD	EEPROM Command	94	00	325	no
	EEPADR	EEPROM Byte Address	95	00	326	no
	EEPDAT	EEPROM Data	96	00	326	no
	EEPSL	EEPROM Start/Load	97	00	327	no
PLL	PLCONA	PLL Configuration A	A1	C1	45	no
	PLCONB	PLL Configuration B	A2	80	45	no
HW Config	HCON	Boot mode System Version Number	A3	BMOD pins SVN pins	44	no
USB Module	EPSEL	Endpoint Select	D2	80	86	no
	USBVAL	USB Data	D3	00	88	no
	ADROFF	Address Offset	D4	00	89	no
	GESR	Global Endpoint Stall	DA	00	85	no
	GEPIR	Global Endpoint Interrupt Request Flag	D6	00	280	no
	CIARI	Config Request Interrupt	D7	00	281	no
	CIARIE	Config Interrupt Enable	D8	00	287	no
	CIAR	Configuration Request Status	D9	00	90	no
	DCR	Device Control	C1	000x0000	91	no
	DPWDR	Device Power Down	C2	00	93	no
	DIER	Device Interrupt Enable	C3	00	285	no
	DIRR	Device Interrupt Request	C4	00	276	no
	DSIR	Device Setup Interrupt	C5	00	278	no
	DGSR	Device Get_Status Register	C9,CA	0000	95	no
	FNRL	Frame Number, Low Byte	C6	xx	94	no
	FNRH	Frame Number, High Byte	C7	00000xxx	94	no
	EPBCn	Endpoint n Buffer Control	C1	00	97	no
	EPBSn	Endpoint n Buffer Status	C2	20	98	no
	EPIEn	Endpoint n Interrupt Enable	C3	00	286	no
	EPIRn	Endpoint n Interrupt Request	C4	01 / 00	278	no
	EPBAn	Endpoint n Base Address	C5	00	100	no
	EPLENn	Endpoint n Buffer Length	C6	0xxxxxxx	101	no
	EGSR	Endpoint Get_Status Register	C9,CA	0000	102	no
	IFCSEL	Interface Select	DB	00	87	no
	IGSR	Interface Get_Status Register	CB,CC	0000	96	no

Table 8 Special Function Registers - Functional Blocks (cont'd)

Block	Symbol	Name	Addr	Contents after Reset	Page No.	Bit Addr
ISDN Register Map (Non SFR Registers)						
ISDN		D and C/I Channel Registers	FF00 - FF2F		205	no
ISDN		Transceiver Registers	FF30 - FF3B		222	no
ISDN		Auxiliary Interface Registers	FF3C - FF3F		232	no
ISDN		IOM-2 and Monitor Handler	FF40 - FF5F		235	no
ISDN		Interrupt & General Configuration Registers	FF60 - FF6F		252	no
ISDN		B-channel A Registers	FF70 - FF7F		258	no
ISDN		B-Channel B Registers	FF80 - FF8F			no

Table 9 Special Function Registers - Numerically ordered addresses

Addr	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
80	reserved	(value 03_H to be written after reset)							
81	SP	.7	.6	.5	.4	.3	.2	.1	.0
82	DPL	.7	.6	.5	.4	.3	.2	.1	.0
83	DPH	.7	.6	.5	.4	.3	.2	.1	.0
84	DPSEL	0	0	0	0	0	D2	D1	D0
87	PCON	res.	SMS	IDLS	res.	GF1	GF0	SME	IDLE
88	TCON	TF1	TR1	TF0	TR0	0	res.	IE0	IT0
89	TMOD	res.	C/$\overline{T}$1	M1(1)	M0(1)	GATE0	C/$\overline{T}$0	M1(0)	M0(0)
8A	TL0	.7	.6	.5	.4	.3	.2	.1	.0
8B	TL1	.7	.6	.5	.4	.3	.2	.1	.0
8C	TH0	.7	.6	.5	.4	.3	.2	.1	.0
8D	TH1	.7	.6	.5	.4	.3	.2	.1	.0
8E	WCON	EWPD	0	0	0	WPUS	WPI0	WPCI	WPTR
93	EEPINT	0	0	0	0	0	0	0	ECINT
94	EEPCMD	.7	.6	.5	.4	.3	.2	.1	.0
95	EEPADR	.7	.6	.5	.4	.3	.2	.1	.0
96	EEPDAT	.7	.6	.5	.4	.3	.2	.1	.0
97	EEPSL	ELD	0	0	0	0	0	0	ESTA
A0	reserved	(value 00_H to be written after reset)							
A1	PLCONA	N4	N3	N2	N1	N0	M3	M2	M1
A2	PLCONB	M0	0	0	PSCVAL	PSCEN	LOCK	SWCK	PCLK
A3	HCON	0	BMOD1	BMOD0	SVN4	SVN3	SVN2	SVN1	SVN0
A4	SYSCON2	0	0	0	0	0	0	STAT2	STAT1
A8	IEN0	EAL	0	EX5	ES	ET1	res.	ET0	EX0
A9	IEN1	0	0	EX11	EX10	EX9	EX8	EX7	EX6
AA	IEN2	0	0	res.	res.	res.	EX14	EX13	EX12
AB	PSIZ	0	0	0	.4	.3	.2	.1	.0
AC	IP1	0	0	.5	.4	.3	.2	.1	.0
AD	SYSCON1	0	0	1	0	0	0	0	XMAP0
AE	XPAGE	.7	.6	.5	.4	.3	.2	.1	.0

Table 9 Special Function Registers - Numerically ordered addresses (cont'd)

Addr	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B0	reserved	(value 00_H to be written after reset)							
B8	IP0	0	0	.5	.4	.3	.2	.1	.0
BC	DSIZ	0	0	0	.4	.3	.2	.1	.0
C1	USB Device and Endpoint Registers. See **Table 10**.								
C2									
C3									
C4									
C5									
C6									
C7									
C8									
C9									
CA									
CB									
CC									
D0	PSW	CY	AC	GF3	RS1	RS0	OV	GF2	P
D2	EPSEL	EPS7	0	0	0	0	EPS2	EPS1	EPS0
D3	USBVAL	.7	.6	.5	.4	.3	.2	.1	.0
D4	ADROFF	0	0	AO5	AO4	AO3	AO2	AO1	AO0
D6	GEPIR	EPI7	EPI6	EPI5	EPI4	EPI3	EPI2	EPI1	EPI0
D7	CIARI	0	0	0	0	0	0	0	DRVI
D8	CIARIE	0	0	0	0	0	0	0	DRVIE
D9	CIAR	0	0	CFG	0	IFC1	IFC0	0	AS
DA	GESR	EPST7	EPST6	EPST5	EPST4	EPST3	EPST2	EPST1	EPST0
DB	IFCSEL	0	0	0	0	0	0	IF1	IF0
E0	ACC	.7	.6	.5	.4	.3	.2	.1	.0
F0	B	.7	.6	.5	.4	.3	.2	.1	.0

Table 10 USB Device and Endpoint Registers

Addr	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Device Registers (selected via EPSEL.EPS7 = 1, EPS2-0 = don't care)									
C1	DCR	0	DA	SWR	SUSP	DINIT	RSM	UCLK	0
C2	DPWDR	0	0	0	0	0	0	TPWD	RPWD
C3	DIER	SE0IE	DAIE	DDIE	SBIE	SEIE	STIE	SUIE	SOFIE
C4	DIRR	SE0I	DAI	DDI	SBI	SEI	STI	SUI	SOFI
C5	DSIR	0	0	0	0	0	0	GSIE	GSIR
C6	FNRL	FNR7	FNR6	FNR5	FNR4	FNR3	FNR2	FNR1	FNR0
C7	FNRH	0	0	0	0	0	FNR10	FNR11	FNR9
C8	reserved								
C9	DGSR	DST7	DST6	DST5	DST4	DST3	DST2	RWUP	PSTAT
CA		DST15	DST14	DST13	DST12	DST11	DST10	DST9	DST8
CB	IGSR	IST7	IST6	IST5	IST4	IST3	IST2	IST1	IST0
CC		IST15	IST14	IST13	IST12	IST11	IST10	IST9	IST8
Endpoint Registers(8 sets of the registers below for endpoints 0...7, selected via EPSEL.EPS7 = 0, EPS2-0 = 000_B ... 111_B)									
C1	EPBC	STALL	0	0	GEPIE	SOFDE	INCE	0	DBM
C2	EPBS	UBF	CBF	DIR	ESP	SETRD	SETWR	CLREP	DONE
C3	EPIE	AIE	NAIE	RLEIE	0	DNRIE	NODIE	EODIE	SODIE
C4	EPIR	ACK	NACK	RLE	0	DNR	NOD	EOD	SOD
C5	EPBA	PAGE	0	0	0	An6	An5	An4	An3
C6	EPLEN	0	Ln6	Ln5	Ln4	Ln3	Ln2	Ln1	Ln0
C7	reserved								
C8	reserved								
C9	EGSR	EST7	EST6	EST5	EST4	EST3	EST2	EST1	STALL
CA		EST15	EST14	EST13	EST12	EST11	EST10	EST9	EST8

3.3 Timer 0 and 1

The SIUC-BA contains four timers, two 16-bit timers embedded in the C800 which are described below, and another two timers in the ISDN section described in **Chapter 5.1.1**.

The C800 contains two 16-bit timers, timer 0 and 1, which are useful in many applications. The timer register is incremented every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 12 oscillator periods, the counter rate is 1/12 of the oscillator frequency.

The timers 0 and 1 of the C800 are fully compatible with timers 0 and 1 of the 80C51 and can be used in the same four operating modes:

Mode 0: 8-bit timer with a divide-by-32 prescaler (M1=0; M0=0)

Mode 1: 16-bit timer (M1=0; M0=1)

Mode 2: 8-bit timer with 8-bit auto-reload (M1=1; M0=0)

Mode 3: Timer 0 is configured as two 8-bit timers;
Timer 1 in this mode holds its count. The effect is the same as setting TR1 = 0. (M1=1; M0=1)

The external input $\overline{\text{INT0}}$ can be programmed to function as a gate for timer 0 to facilitate pulse width measurement.

Each timer consists of two 8-bit registers (TH0 and TL0 for timer 0, TH1 and TL1 for timer 1) which may be combined to one timer configuration depending on the mode that is selected. The functions of the timers are controlled by two special function registers TCON and TMOD.

In the following descriptions the symbols TH0 and TL0 are used to specify the high-byte and the low-byte of timer 0 (TH1 and TL1 for timer 1, respectively). The operating modes are similar for both timers, except timer 0 only provides the gate function (bit GATE0) with the external signal $\overline{\text{INT0}}$.

The timers are fully compliant to the C500 series of microcontrollers (e.g. C501).

3.3.1 Mode 0

Putting timer 0 and 1 into mode 0 configures it as an 8-bit timer with a divide-by-32 prescaler (**Figure 8**).

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1's to all 0's, it sets the timer overflow flag TFx (x = 0 or 1). The overflow flag TFx then can be used to request an interrupt. The counted input is enabled to the timer when TRx = 1 and either GATE0 = 0 or $\overline{INT0}$ = 1 (setting GATE0 = 1 allows the timer to be controlled by external input $\overline{INT0}$, to facilitate pulse width measurements; for timer 0 only). TRx is a control bit in the special function register TCON; bit GATE0 is in TMOD.

The 13-bit register consists of all 8 bits of THx and the lower 5 bits of TLx. The upper 3 bits of the TLx register are indeterminate and should be ignored. Setting the run flag (TRx) does not clear the registers.

Mode 0 operation is similar for timer 0 and timer 1, however only for timer 0 there is the gate bit GATE0 and the external signal $\overline{INT0}$.

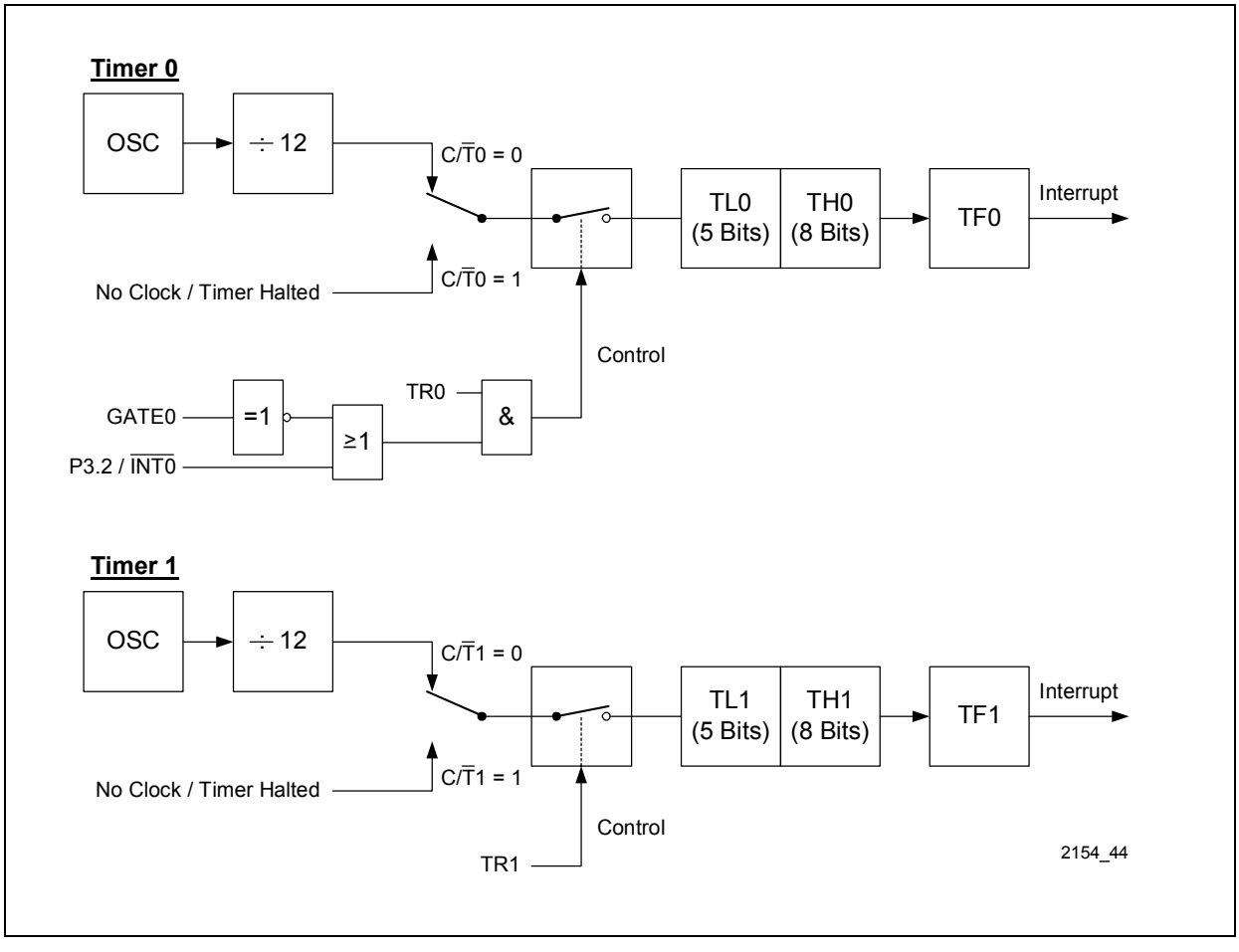


Figure 8 Timer Mode 0: 13-Bit Timer

3.3.2 Mode 1

Mode 1 is the same as mode 0, except that the timer register is running with all 16 bits (**Figure 9**).

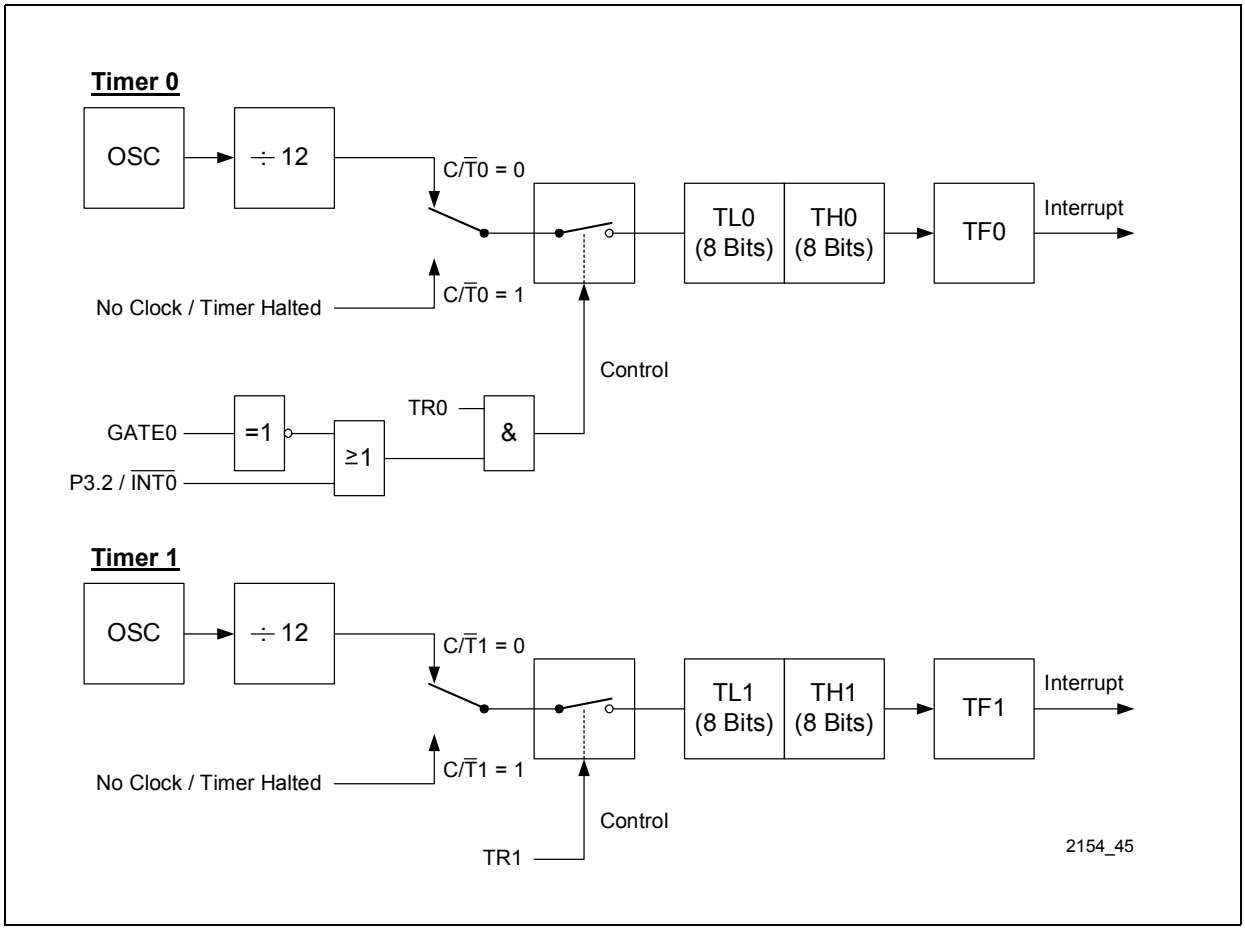


Figure 9 Timer Mode 1: 16-Bit Timer

3.3.3 Mode 2

Mode 2 configures the timer register as an 8-bit counter (TLx) with automatic reload, as shown in **Figure 10**. Overflow from TLx not only sets TFx, but also reloads TLx with the contents of THx, which is preset by software. The reload leaves THx unchanged.

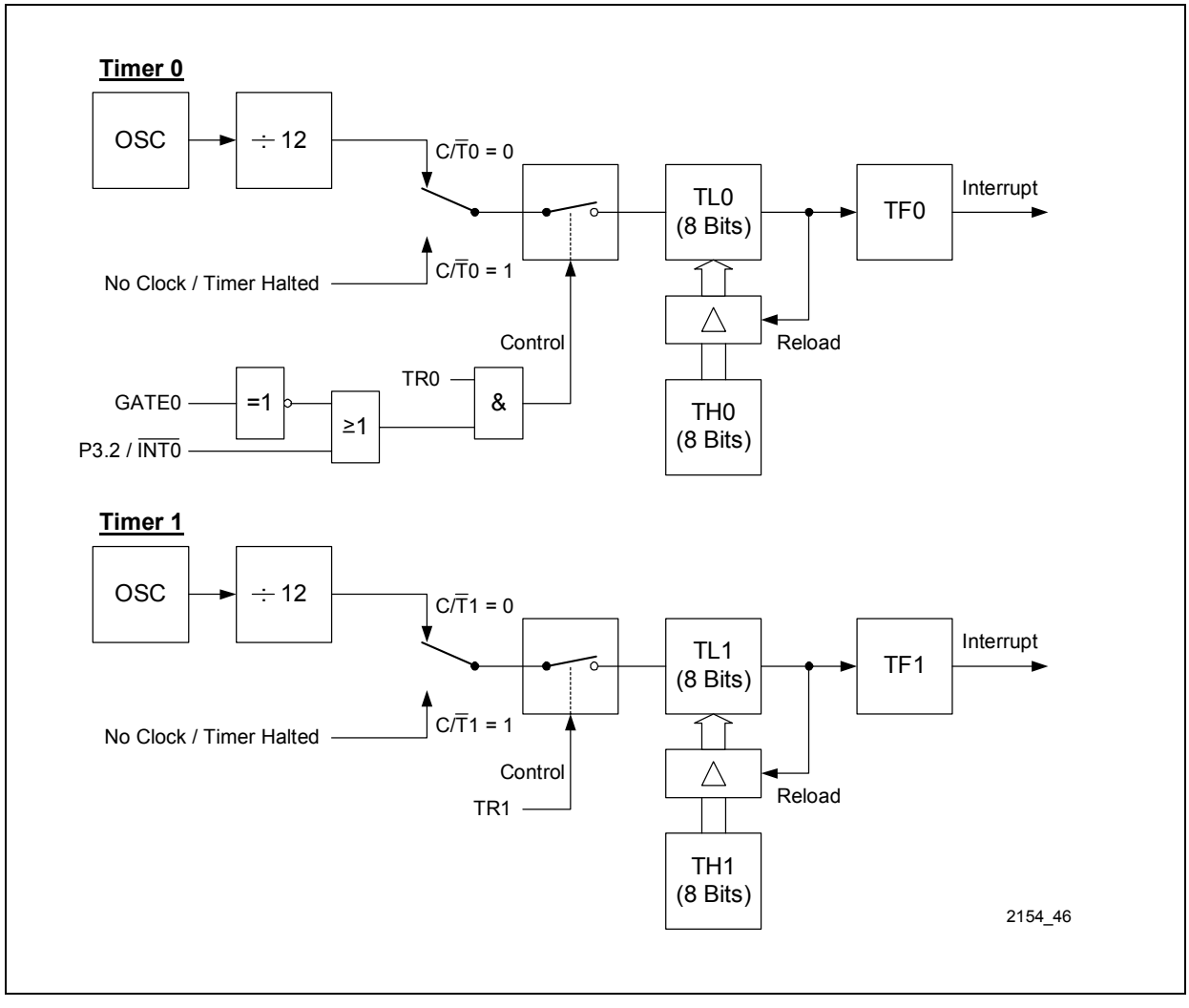


Figure 10 Timer Mode 2: 8-Bit Timer with Auto-Reload

3.3.4 Mode 3

Mode 3 has different effects on timer 0 and timer 1. Timer 1 in mode 3 simply holds its count. The effect is the same as setting TR1=0. Timer 0 in mode 3 establishes TL0 and TH0 as two seperate counters. The logic for mode 3 on timer 0 is shown in **Figure 11**. TL0 uses the timer 0 control bits: C/$\overline{T}$0, GATE0, TR0, $\overline{INT0}$ and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from timer 1. Thus, TH0 now controls the “timer 1” interrupt.

Mode 3 is provided for applications requiring an extra 8-bit timer or counter. When timer 0 is in mode 3, timer 1 can be turned on and off by switching it out of and into its own mode 3, or can still be used by the serial channel as a baud rate generator, or in fact, in any application not requiring an interrupt from timer 1 itself.

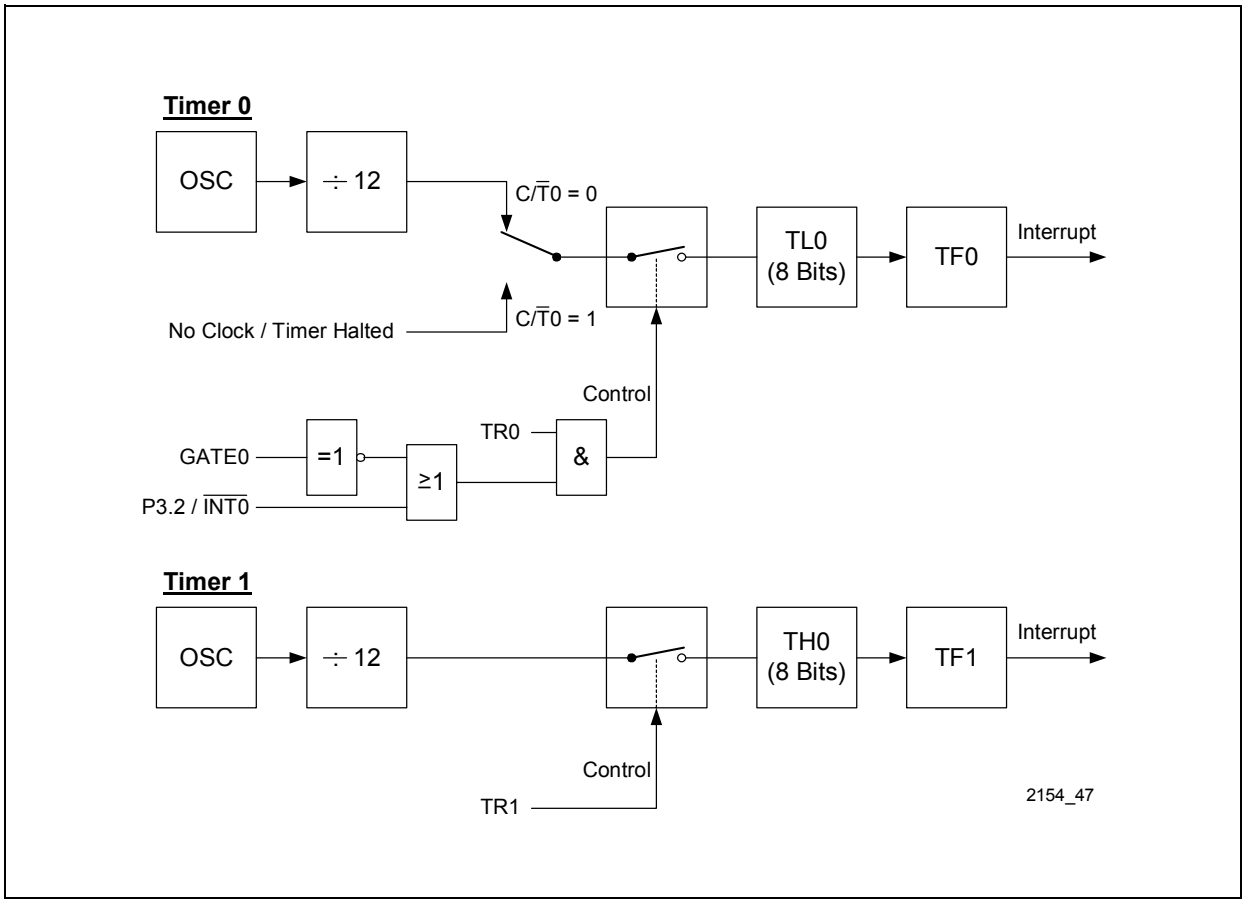


Figure 11 Timer Mode 3: Two 8-Bit Timers

3.4 Timer 0 and 1 Registers

Totally six special function registers control the timer 0 and 1 operation :

- TL0/TH0 and TL1/TH1 - counter registers, low and high part
- TCON and TMOD - control and mode select registers

3.4.1 TLx / THx - Timer Low / High Registers

Reset value: 00_H Address: $8A_H$

	7	6	5	4	3	2	1	0
TL0								
	rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: $8B_H$

	7	6	5	4	3	2	1	0
TL1								
	rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: $8C_H$

	7	6	5	4	3	2	1	0
TH0								
	rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: $8D_H$

	7	6	5	4	3	2	1	0
TH1								
	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Function
TLx.7-0 x=0-1	Timer 0/1 low register

Operating Mode	Description
0	“TLx” holds the 5-bit prescaler value.
1	“TLx” holds the lower 8-bit part of the 16-bit timer value.
2	“TLx” holds the 8-bit timer value.
3	TL0 holds the 8-bit timer value; TL1 is not used.

Bit	Function
THx.7-0 x=0-1	Timer 0/1 high register

Operating Mode	Description
0	“THx” holds the 8-bit timer value.
1	“THx” holds the higher 8-bit part of the 16-bit timer value
2	“THx” holds the 8-bit reload value.
3	TH0 holds the 8-bit timer value; TH1 is not used.

3.4.2 TCON - Timer Control Register

Reset value: 00_H Address: 88_H

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	0	res.	IE0	IT0
r	rw	r	rw	r	rw	r	rw

The **Timer Control Register (TCON)** is also described in the section on interrupts.

Bit	Function
TR0	**Timer 0 run control bit** Set/cleared by software to turn timer 0 ON/OFF.
TF0	**Timer 0 overflow flag** Set by hardware on timer overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	**Timer 1 run control bit** Set/cleared by software to turn timer 1 ON/OFF.

Bit	Function
TF1	**Timer 1 overflow flag** Set by hardware on timer overflow. Cleared by hardware when processor vectors to interrupt routine.
IE0	**External Interrupt 0 Edge Flag** Set by hardware when an external interrupt condition at pin $\overline{INT0}$ is detected (low level if IT0=0 or falling edge if IT0=1)
IT0	**Interrupt 0 Control Bit** If 1, a falling edge triggers an interrupt; if 0, a low level triggers an interrupt.

Note: res. = Bit is reserved and must not be changed.

3.4.3 TMOD - Timer Mode Register

Reset value: 00_H Address: 89_H

7	6	5	4	3	2	1	0
res.	C/$\overline{T}$1	M1(1)	M0(1)	GATE0	C/$\overline{T}$0	M1(0)	M0(0)
rw	rw	rw	rw	rw	rw	rw	rw

Timer 1 Control: C/$\overline{T}$1, M1-0(1)
Timer 2 Control: GATE0, C/$\overline{T}$0, M1-0(0)

Bit	Function
C/$\overline{T}$x (x=0, 1)	**Timer x Selector** 0: Input is from internal system clock. 1: No input clock (timer is halted).
GATE0	**Timer 0 Gate Flag** 0: Timer 0 is enabled whenever TCON.TR0 is set to ’1’ (software control). 1: Timer 0 is enabled only if TCON.TR0 is set to ’1’ and if the $\overline{INT0}$ pin is set to ’1’ (hardware control).

M1-0 (x)
(x= 0, 1)

Timer x Mode Control bit M1-0

M1	M0	Function
0	0	8-bit timer: “THx” operates as 8-bit timer “TLx” serves as 5-bit prescaler
0	1	16-bit timer. “THx” and “TLx” are cascaded; there is no prescaler
1	0	8-bit auto-reload timer. “THx” holds a value which is to be reloaded into “TLx” each time it overflows
1	1	Timer 0 : TL0 is an 8-bit timer controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only controlled by timer 1 control bits. Timer 1 : Timer 1 stops

Note: res. = Bit is reserved and must not be changed.

3.5 Microcontroller Registers

Stack Pointer

The SP register (address 81_H) contains the Stack Pointer. The Stack Pointer is used to load the program counter into memory during LCALL and ACALL instructions and is used to retrieve the program counter from memory in RET and RETI instructions. The stack may also be saved or loaded using PUSH and POP instructions, which also increment and decrement the Stack Pointer. The Stack Pointer points to the top location of the stack. On reset, the Stack Pointer is set to 07_H.

General Purpose Flags

The SIUC-BA provides 4 general purpose flags GF3-0 which can be read/written by the microcontroller without effect on any function of the SIUC-BA. Their value is only set to their default value with a reset, otherwise it’s only changed by a µC write access. GF0 and GF1 are located in the PCON register, GF2 and GF3 are part of the PSW register.

3.5.1 DPSEL - Data Pointer Select Register

The microcontroller architecture provides 8 16-bit pointers for indirect addressing of external devices, for offset code byte fetches and for offset program jumps. The current data pointer is given by the contents of the **Data Pointer High (DPH) register** and the **Data Pointer Low (DPL) register** (see **Chapter 3.5.2**). The 3 least significant bits of the **DPSEL** register are used to select among the 8 data pointers DPTR0 - DPTR7.

Reset value: 00_H Address: 84_H

7	6	5	4	3	2	1	0
0	0	0	0	0	D2	D1	D0
r	r	r	r	r	rw	rw	rw

Bit	Function
D2 - D0	**Data Pointer Select Bits** 000: DPTR0 selected (default after reset) 001: DPTR1 selected 010: DPTR2 selected 011: DPTR3 selected 100: DPTR4 selected 101: DPTR5 selected 110: DPTR6 selected 111: DPTR7 selected

The addresses of the DPH and DPL registers remain 83H and 82H, respectively, under all conditions. **Figure 12** illustrates the access mechanism.

3.5.2 Data Pointer Register Low / High - DPL / DPH

Reset value: 00_H Address: 82_H

	7	6	5	4	3	2	1	0
DPL								
	rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: 83_H

	7	6	5	4	3	2	1	0
DPH								
	rw	rw	rw	rw	rw	rw	rw	rw

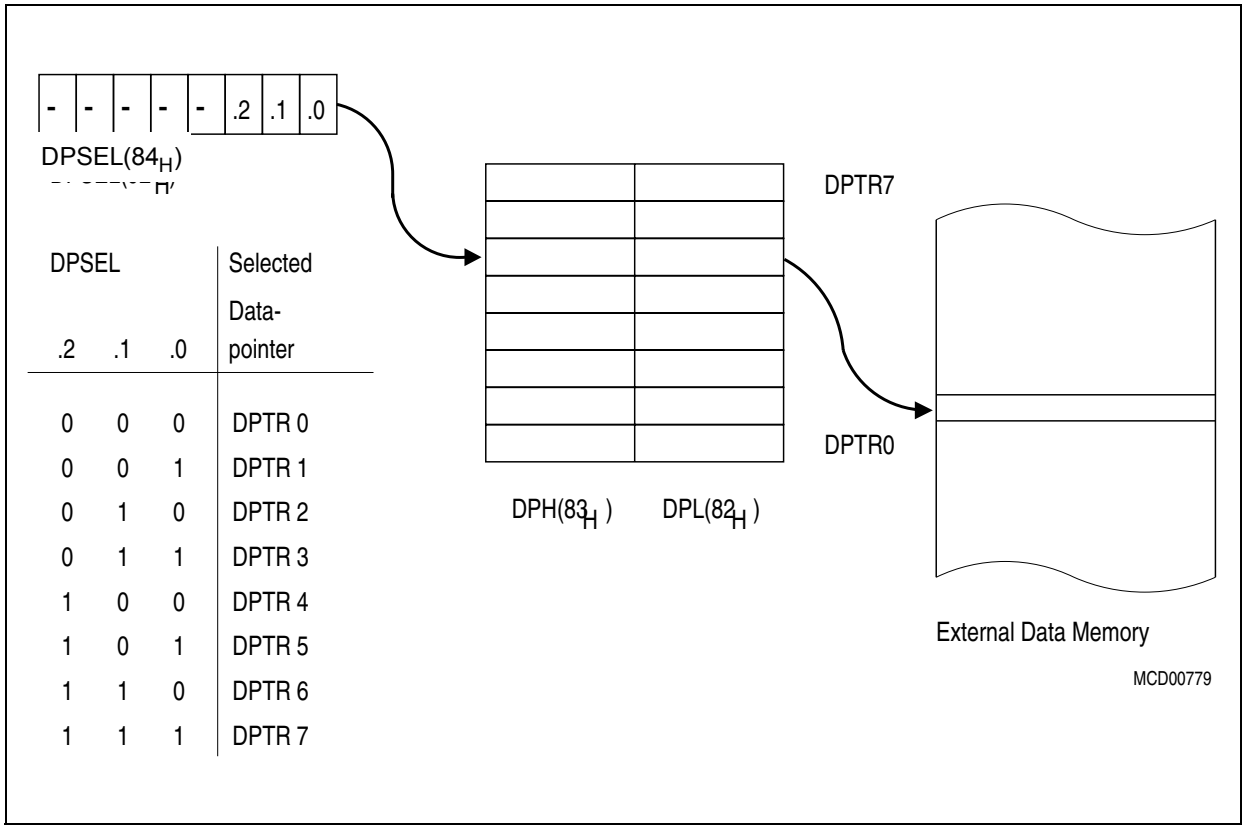


Figure 12 Data Memory Access via 8 data pointers

3.5.3 PCON - Power Control Register

Reset value: 00_H Address: 87_H

7	6	5	4	3	2	1	0
res.	SMS	IDLS	res.	GF1	GF0	SME	IDLE
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Function
SMS	**Suspend Mode Start** When set to 1, power-down (suspend) mode is started, provided SME (PCON.1) was set by the previous instruction. The bit is then automatically cleared by hardware. It always returns 0 when read. The power down state is described in **Chapter 9.2.2**.
IDLS	**Idle Start** When written to 1, idle mode is started, provided IDLE (PCON.0) was set by the previous instruction. The bit is then automatically cleared by hardware. It always returns a 0 when read. The idle state is described in **Chapter 9.2.1**.
GF1	**General Purpose Flag 1**
GF0	**General Purpose Flag 0**
SME	**Suspend Mode Enable** Writing a 1 starts a delay cycle whereby the following instruction can put the core into power-down mode by writing a 1 to PCON.6 (SMS). The bit is then automatically cleared by hardware. It always returns a 0 when read.
IDLE	**Idle Mode Enable** Writing a 1 starts a delay cycle whereby the following instruction can put the core into idle mode by writing a 1 to PCON.5 (IDLS). The bit is then automatically cleared by hardware. It always returns a 0 when read.

Note: res. = Bit is reserved and must not be changed.

3.5.4 PSW - Program Status Word Register

The program status word register contains status information resulting from CPU and ALU operations.

Reset value: 00_H Address: $D0_H$

7	6	5	4	3	2	1	0
CY	AC	GF3	RS1	RS0	OV	GF2	P
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Function
CY	**ALU Carry Flag**
AC	**ALU Auxiliary Carry Flag**
GF3	**General Purpose Flag 3**
RS1,RS0	**Register Bank Select Bit 1,0** RS1=0, RS0=0: RB0. Registers from 00 - 07_H. RS1=0, RS0=1: RB1. Registers from 08 - $0F_H$. RS1=1, RS0=0: RB2. Registers from 10 - 17_H. RS1=1, RS0=1: RB3. Registers from 18 - $1F_H$.
OV	**ALU Overflow Flag**
GF2	**General Purpose Flag 2**
P	**Parity Flag** Set each instruction cycle to indicate odd/even parity in the accumulator.

3.5.5 WCON - WakeUp Control Register

Reset value: 00_H Address: 8EH

7	6	5	4	3	2	1	0
EWPD	0	0	0	WPUS	WPI0	WPCI	WPTR
rw	r	r	r	rw	rw	rw	rw

Bit	Function
EWPD	**External WakeUp from Power Down Enable** Setting EWPD before entering power down mode enables the external wakeup capability from power down mode, via the pin $\overline{INT0}$, the USB module, the S-Transceiver, a C/I Code change or from one of the interrupt signals $\overline{INT1}$, $\overline{INT2}$ or $\overline{EAW}$. Each of the four groups of sources can individually be enabled. However, enabling one of the four wakeup sources (WPUS, WPIO, WPCI or WPTR) has no effect if EWPD is not set to ’1’ at the same time.
WPUS	**Wakeup via USB bus enabled** Any activity on the bus (USB resume) will wakeup the SIUC-X from suspend mode. In normal operation mode this event is indicated in the interrupt status bit DIRR.SEI.
WPI0	**WakeUp via $\overline{INT0}$ enabled** An active signal on pin $\overline{INT0}$ will wakeup the SIUC-X from suspend mode. In normal operation mode this event is indicated in the interrupt status bit TCON.IE0
WPCI	**WakeUp via C/I-Code Change, $\overline{EAW}$ or $\overline{INT1/2}$ interrupt pins enabled** A C/I-code change or an active signal on one of the interrupt pins $\overline{EAW}$, $\overline{INT1}$ or $\overline{INT2}$ will wakeup the SIUC-X from suspend mode. In normal operation mode these events are indicated in the corresponding interrupt status bits ISTA.CIC, AUXI.EAW and AUXI.INT1/2, respectively.
WPTR	**WakeUp via S-Transceiver level detect enabled** Any activity on the S bus will wakeup the SIUC-X from suspend mode. In normal operation mode this event is indicated in the interrupt status bit ISTATR.LD.

Note: Setting to ’1’ means ’enabled’, setting to ’0’ means ’disabled’.

For further information see **Chapter 6.3** and **Chapter 9.3.6**.

3.5.6 HCON - Hardware Configuration Register

Reset value: 00_H Address: $A3_H$

7	6	5	4	3	2	1	0
0	BMOD1	BMOD0	SVN4	SVN3	SVN2	SVN1	SVN0
r	r	r	rw	rw	rw	rw	rw

Bit	Function
BMOD1- BMOD0	**Boot Mode** The state of the boot mode pins is available at this address. BMOD1/0 together with pin $\overline{EA}$ determine in which operation mode the µC starts after reset (see **Chapter 7.1**)
SVN4-0	**System Version Number** These bits have no effect on the functions of the device but they can be used to identify different hardware configurations as this number is forwarded to the host for identification purposes. • SVN1-0: The state of the SVN1 and SVN0 pins is latched during reset and can be read from this bit position. If an EEPROM is connected, the µC overwrittes these bits with a system version number loaded from the EEPROM. • SVN4-2: If an EEPROM is connected, the µC overwrittes these bits with a system version number loaded from the EEPROM. If no EEPROM is connected these bits keep their reset values (000_B).

The HCON register is transferred to the host after power on reset by means of a string descriptor. The values can be used by the host to uniquely identify the system configuration built around the SIUC-BA in order to select the appropriate firmware and driver software.

3.5.7 PLCONA/B - PLL Configuration Registers A, B

Reset value: $C1_H$ Address: $A1_H$

	7	6	5	4	3	2	1	0
PLCONA	N4	N3	N2	N1	N0	M3	M2	M1
	rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 80_H Address: $A2_H$

	7	6	5	4	3	2	1	0
PLCONB	M0	0	0	PSCVAL	PSCEN	LOCK	SWCK	PCLK
	rw	r	r	rw	rw	r	rw	rw

Bit	Function
N4 - N0	**Factor N** Determines the multiplication factor of the internal PLL. Values between 0 and 31 are possible (default = 24_D => multiplication by 25)
M3 - M0	**Factor M** Determines the division factor of the internal PLL. Values between 0 and 15 are possible (default = 3_D => division by 4)
PSCVAL	**Prescaler Value** The clock for the microcontroller is devided by a prescaler if PSCEN=1: 0: µC clock divided by 2 1: µC clock divided by 1.5 PSCVAL has no effect if PSCEN=0 (prescaler disabled).
PSCEN	**Prescaler Enable** Setting this bit to 1 enables the prescaler which is configured in PSCVAL.
LOCK	**PLL Locked** When 0, the PLL frequency is not stabilized and setting of bit SWCK is not allowed. When 1, the PLL frequency has stabilized and the setting of bit SWCK is allowed.

Bit	Function
SWCK	**Switch Clock** When 0, the PLL is bypassed and the output clock is equal to the input clock. When 1, the output clock of the PLL is derived from the input clock according to the following equation: $$\text{output} = \frac{N+1}{M+1} \times \text{input}$$
PCLK	**PLL Clock Enable** Bit PCLK controls the 48 MHz PLL. If PCLK=0, the 48 MHz PLL is disabled (default after reset). If PCLK=1, the 48 MHz PLL is enabled As the PLL is disabled after reset, the microcontroller and USB blocks are clocked at crystal speed (7.68 MHz). For normal operation the PLL has to be programmed first.

Please also refer to **Chapter 8.1**.

3.5.8 ACC / B - Accumulator / B Register

The **Accumulator (ACC)** provides one of the operands for most ALU operations. The **B Register (B)** provides the second operand for multiply or divide instructions. At other times, it may be used as a scratch pad register.

Reset value: 00_H Address: $E0_H$

	7	6	5	4	3	2	1	0
ACC								
	rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: $F0_H$

	7	6	5	4	3	2	1	0
B								
	rw	rw	rw	rw	rw	rw	rw	rw

3.5.9 PSIZ - Program RAM Size Register

The PSIZ and DSIZ registers partition the onchip 16 Kbyte RAM into Program space and Data space. Their contents are valid only in certain firmware modes as described in **Chapter 7**.

Reset value: 06_H Address: AB_H

7	6	5	4	3	2	1	0
0	0	0	.4	.3	.2	.1	.0
r	r	r	rw	rw	rw	rw	rw

Bit	Function
PSIZ.4 - PSIZ.0	**Program RAM Allocation** These bits define the size of the Program RAM within the 16 kByte internal RAM. The selection of PRAM size has a granularity of 1 kByte. 00_H: No Program RAM is preset. 01_H: 1 kByte Program RAM is present. 02_H: 2 kByte Program RAM is present. . . 10_H - $1F_H$: 16 kByte Program RAM is present.

3.5.10 DSIZ - Data RAM Size Register

Reset value: $0A_H$ Address: BC_H

7	6	5	4	3	2	1	0
0	0	0	.4	.3	.2	.1	.0
r	r	r	rw	rw	rw	rw	rw

Bit	Function
DSIZ.4 - DSIZ.0	**Data RAM Allocation** These bits define the size of the Data RAM within the 16 kByte internal RAM. The selection of DRAM size has a granularity of 1 kByte. 00_H: No Data RAM is preset. 01_H: 1 kByte Data RAM is present. 02_H: 2 kByte Data RAM is present. . . 10_H - $1F_H$: 16 kByte Data RAM is present.

Note: The sum of PSIZ and DSIZ must not exceed the internal RAM space of 16 kByte.

3.5.11 SYSCON1 - System Control Register 1

Reset value: 21_H Address: AD_H

7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	XMAP0
r	r	rw	r	r	rw	rw	rw

Bit	Function
XMAP0	**Global XRAM Access Enable/Disable Control** 0: The access to onchip XRAM is enabled. 1: The access to onchip XRAM is disabled (default after reset). All MOVX accesses are performed via the external bus. After reset the µC starts program execution from internal ROM. The XMAP0 bit must be set to 0 before any access to XRAM address space (download to 16K internal RAM, or access to ISDN registers) can be performed. Once cleared, this bit can only be set again by a reset operation on the device.

3.5.12 SYSCON2 - System Control Register 2

Reset value: 00_H Address: $A4_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	STAT2	STAT1
r	r	r	r	r	rw	rw	rw

Bit	Function
STAT2	**Status Bit 2** This bit is used to switch executable firmware: 0: the boot loader in internal ROM is connected to the µC 1: the operational firmware in RAM is connected to the µC This bit is only activated with the rising edge of SYSCON2.STAT1, i.e. for switching of firmware from ROM to RAM, first the STAT2 bit must be set to 0 or 1 and then with a consecutive register access the STAT1 bit must be set to 1. This bit is not reset to its default value if STAT1 is set (µC reset), but the programmed value is retained and can be read back after the reset.
STAT1	**Status Bit 1** This bit is the software reset bit for the µC. Setting it to 1 initiates a reset to the µC and all its special function registers (SFR). This bit is automatically reset by the hardware.

3.5.13 XPAGE - XRAM Page Register

Reset value: 00_H Address: AE_H

7	6	5	4	3	2	1	0
.7	.6	.5	.4	.3	.2	.1	.0
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Function
XPAGE.7 - XPAGE.0	**XRAM High Address Byte** The contents of this register are used as the high byte of the XRAM address for paged accesses, for 8-bit MOVX instructions with Ri (i=0,1).

For further information refer to **Chapter 3.1**.

4 USB Module

The USB module handles all transactions between the universal serial bus USB and the internal (parallel) bus of the microcontroller. The USB module includes several units which are required to support data handling with the USB bus:

- An onchip USB bus transceiver
- A USB memory with 2 pages of 128 bytes each
- The memory management unit (MMU) for USB and µC memory access control
- The USB device core (UDC) for USB protocol handling
- A µC interface with the USB specific special function registers and interrupt control logic

Figure 13 shows the block diagram of the functional units of the USB module with their interfaces.

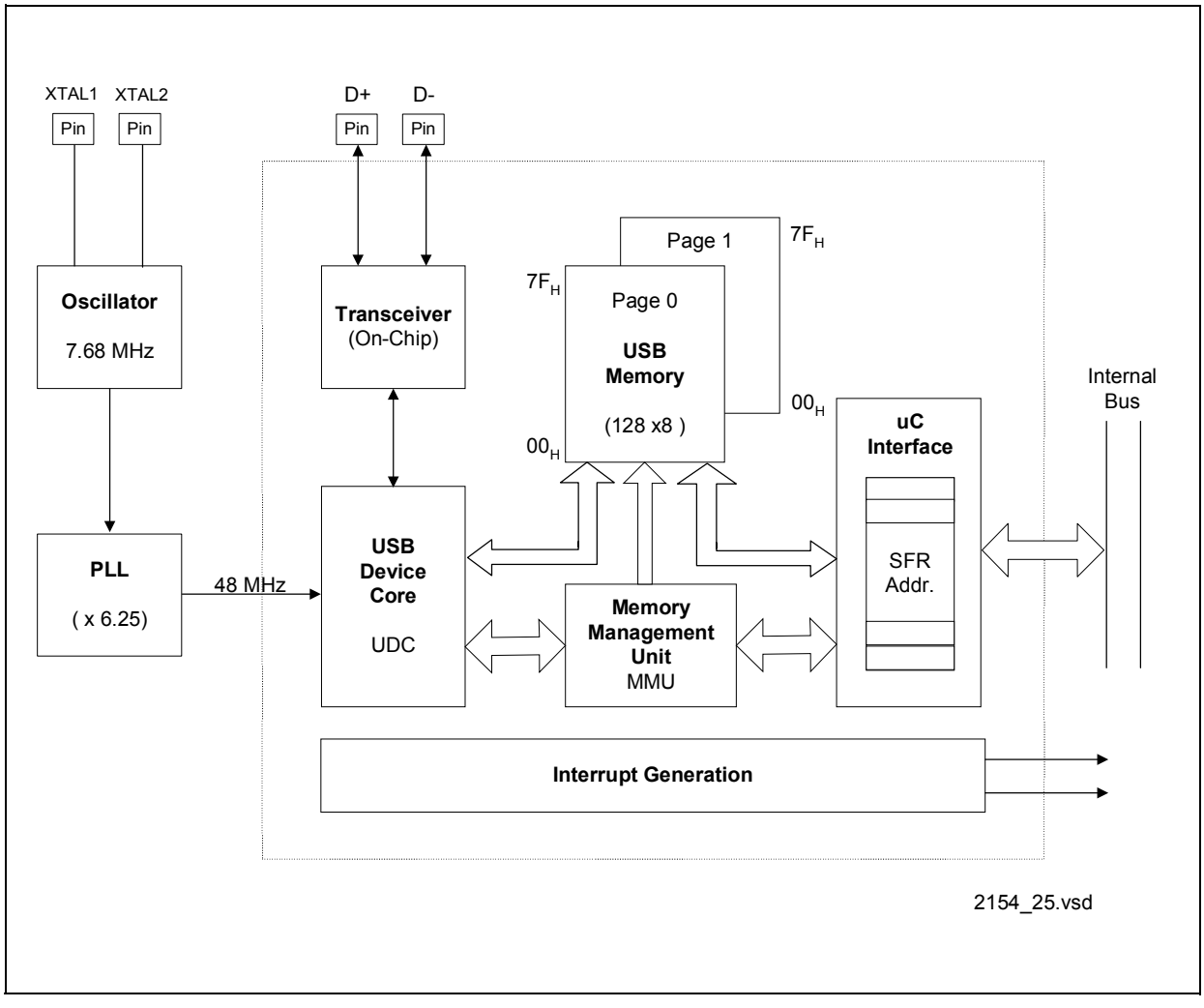


Figure 13 USB Module Block Diagram

4.1 Transfer Modes

USB data transfers take place between host and a particular endpoint on a USB device. A given USB device may support multiple data transfer endpoints. The USB host treats communications with any endpoint of a USB device independently from any other endpoint. Such associations between the host software and a USB device endpoint are called pipes. As an example, a given USB device could have an endpoint supporting a pipe for transporting B1-channel data from the host to the USB device and another endpoint supporting a pipe for transporting B1-channel data from the USB device to the host. The USB architecture of the SIUC-BA comprehends all four basic types of data transfers, i.e. Control, Isochronous, Interrupt and Bulk.

Table 11 USB Transfer Modes

Mode	Function
Control	Control data are used to configure devices, data transmission is lossless. Control pipes are bidirectional, data transfer is possible in both directions via one pipe. Endpoint 0 is always configured as control endpoint with a maximum buffer length of 8 bytes. The control endpoint can be configured to handle data packets of 64 bytes maximum length.
Isochronous	Isochronous data are continuous and real-time in creation and consumption, such as voice data. In this case, real-time is defined from frame to frame. Isochronous data transfer has the highest priority, but is not always lossless. Isochronous pipes are always unidirectional, so one endpoint can be associated to an IN pipe or an OUT pipe. The SIUC-X supports up to 64 bytes.
Interrupt	Interrupt data are a small amount of data, which are transferred to the host every n frames, with n being programmable by the host. Data delivery is lossless. Interrupt pipes are always unidirectional IN pipes, the maximum data packet length is limited to 64 bytes.
Bulk	Bulk data can be a larger amount of data, which can be split by the host in several data packets within one frame. Data delivery is lossless. Bulk pipes are always unidirectional, so one endpoint can be associated to an IN pipe or an OUT pipe. The maximum data packet length is limited to 64 bytes.

4.2 Memory Buffer Modes

4.2.1 Overview

Every endpoint of the USB module can operate in 2 modes, dual buffer mode and single buffer mode. Each mode provides random or sequential access to the USB memory. **Figure 14** shows the possible buffer modes.

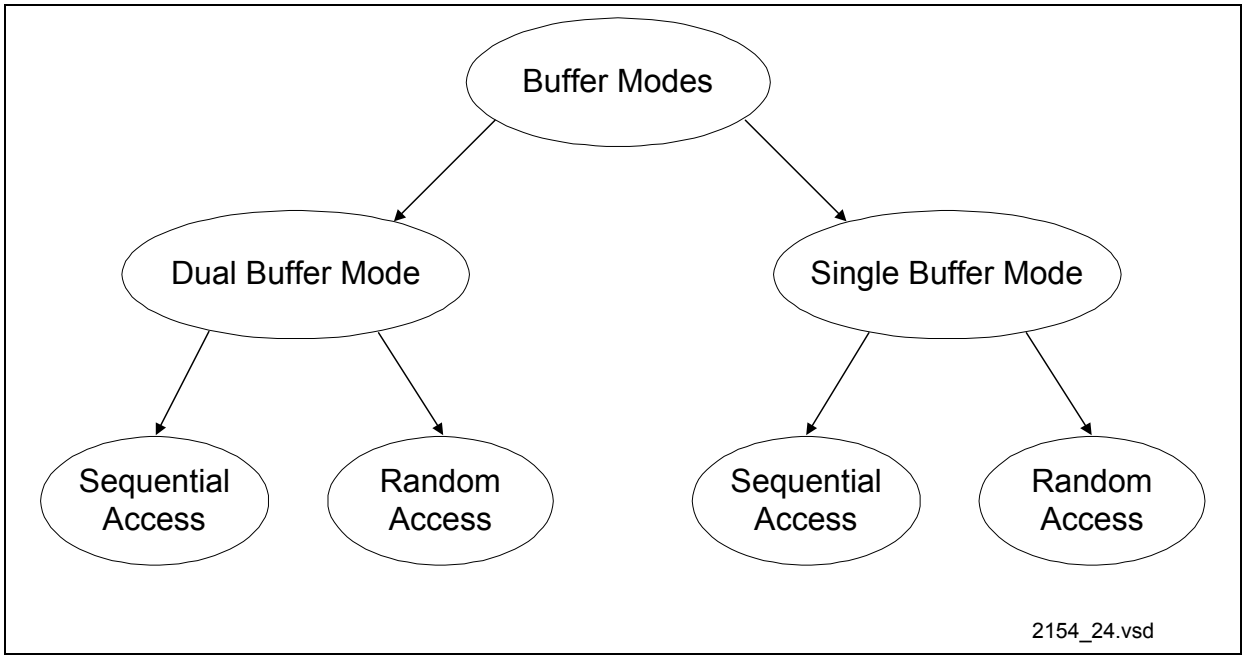


Figure 14 Memory Buffer Modes

Single Buffer Mode

In single buffer mode, the USB and the CPU use one common memory page. The active buffer page is either page 0 or page 1. This mode is dedicated for data strings with a maximum length of 128 bytes.

Dual Buffer Mode

In dual buffer mode the USB and the CPU write into different USB memory pages allowing back-to-back data transfers. Switching between the pages is done automatically, enabling a high data transfer rate between the CPU and the USB module. This mode is dedicated for large data packets per frame (endless strings).

Random Access

Random access is available in both modes. Random access allows to change only a few bytes in a data block of the USB memory buffer without requiring the µC to enter a complete data block. When the CPU has modified the bytes in the data block, setting of

bit DONE by software marks the buffer ready for transmission or reception of data over the USB pipe. For modification of a specific byte in the buffer, the CPU must write the address to SFR ADROFF and read/write the data byte from/to register USBVAL.

Sequential Access

In sequential access mode the CPU accesses the data register USBVAL continuously without setting the address of the next USB memory buffer location. This is done automatically if bit INCE (increment enable) in the related SFR EPBCRn is set. After a specific number of CPU accesses (as done in SFR EPLENn), the buffer has been read/written by the CPU and is empty/full. Setting of bit DONE in software, manually or automatically, marks the USB buffer ready.

4.2.2 Single Buffer Mode

In single buffer mode the USB and the CPU share one common USB memory page. The active buffer page can be either page 0 or page 1. Back-to-back transfers are not possible in this mode. Easy data storage and controlling can be achieved in this mode. E.g. a once created data set for an interrupt endpoint can be stored permanently in USB memory. As a result, an additional memory space for data storage is no longer needed.

4.2.2.1 USB Write Access

Figure 15 shows the basic flowchart of a USB write access to one USB memory buffer in single buffer mode.

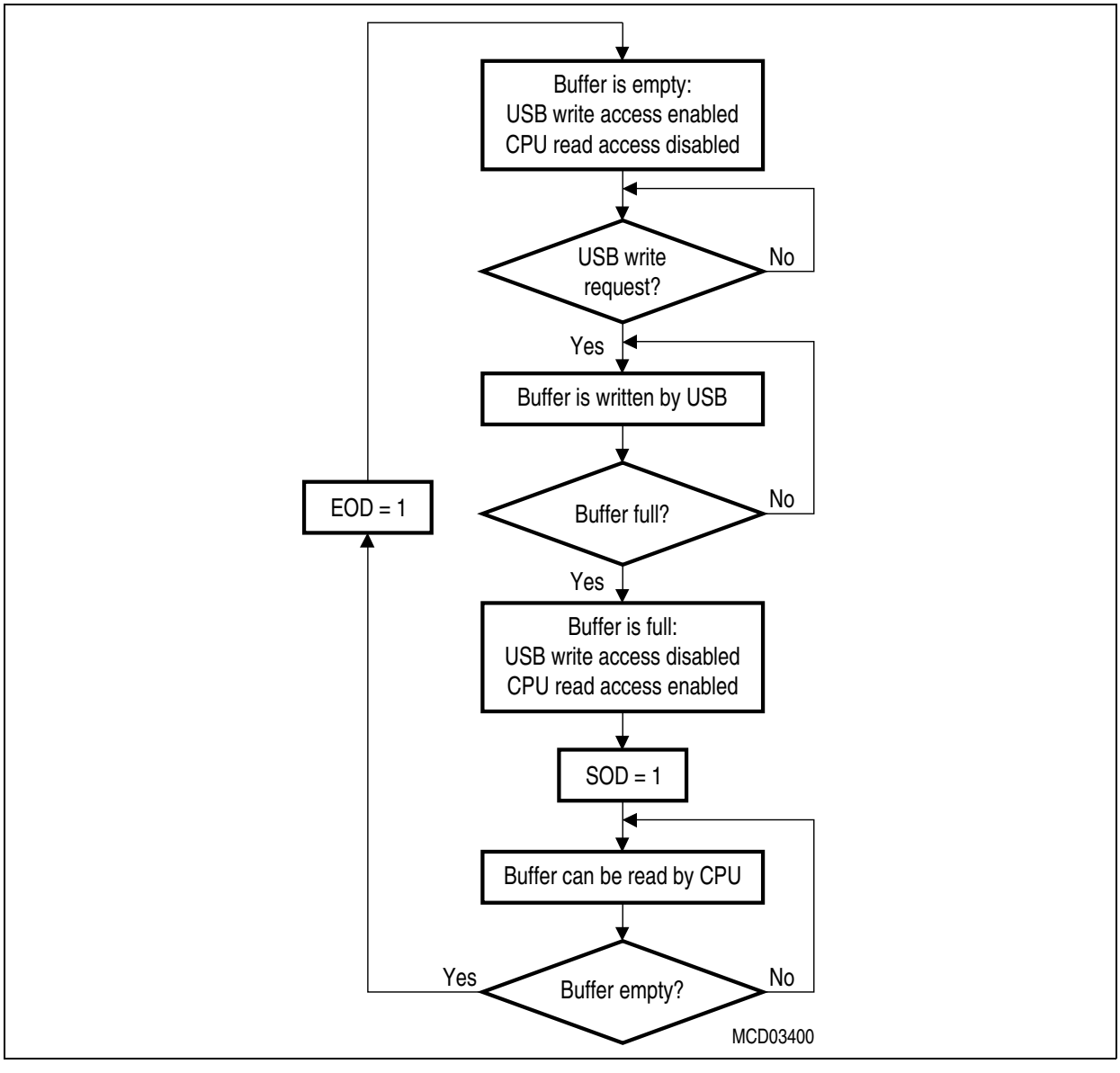


Figure 15 USB Write Access in Single Buffer Mode - Buffer Handling

Figure 16 shows more details of a USB write access to USB memory in single buffer mode. After SOF(n) (start of frame) occured at ①, the USB starts writing at ② a fixed number of bytes into the USB memory. A byte counter is incremented after every USB memory write operation. When the USB memory write operation (Len(n)) is finished correctly, bit SOD (start of data) is set at ③, indicating a full USB memory buffer. Furthermore, the byte counter value is stored in the corresponding length register, indicating the number of bytes which have been transferred and can be now read by the CPU. Subsequently, the CPU can read data bytes from USB memory, generating an EOD (end of data) at ④ after the last byte has been read. Bit EOD set indicates an empty USB buffer, which now can be written again by the USB.

Figure 16 also shows a second USB write access operation with a different number of bytes (Len(n+1)), where the CPU read operation from the USB memory is interrupted twice.

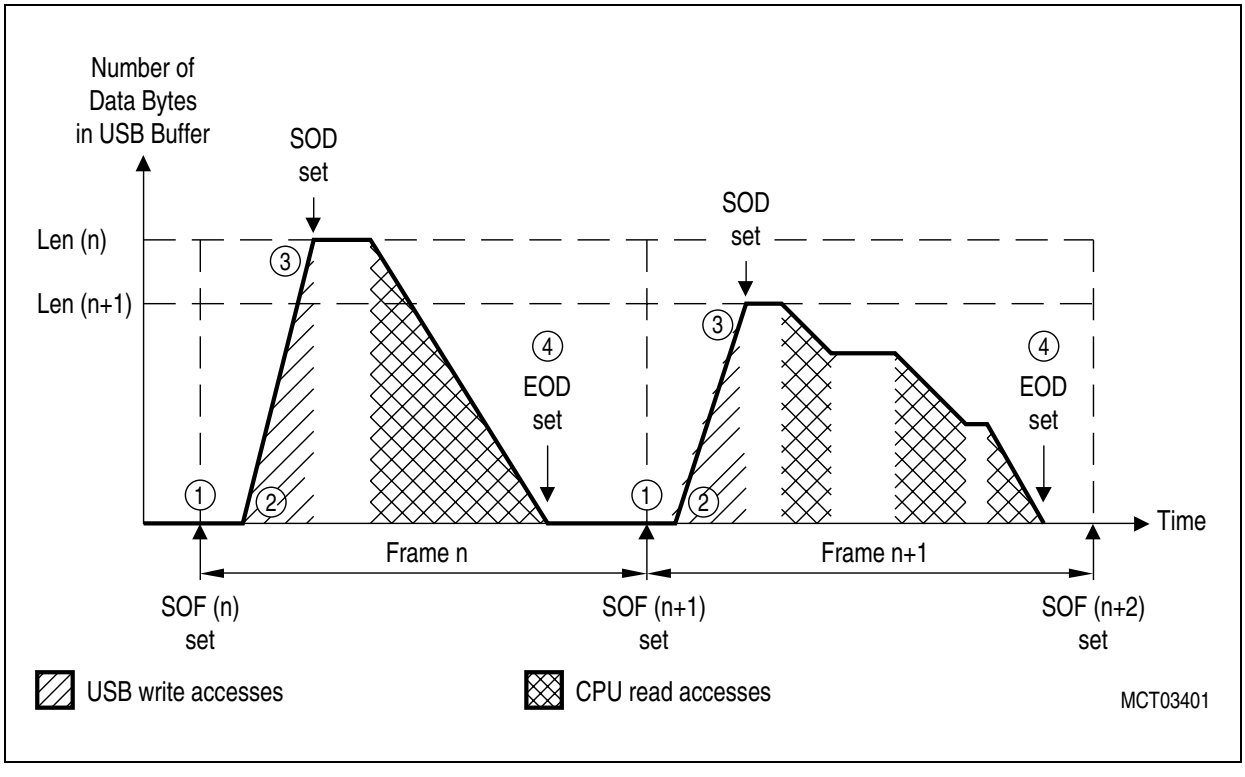


Figure 16 Single Buffer Mode : Standard USB Write Access

Note: The CPU accesses shown in the following diagrams assume that bit INCE in the corres-ponding endpoint control register is set.
A frame is the 1 ms time interval defined by the USB host.
Every frame begins with a SOF token (start-of-frame).

4.2.2.2 USB Read Access

Figure 17 shows the basic flowchart of a USB read access from one USB memory buffer in single buffer mode.

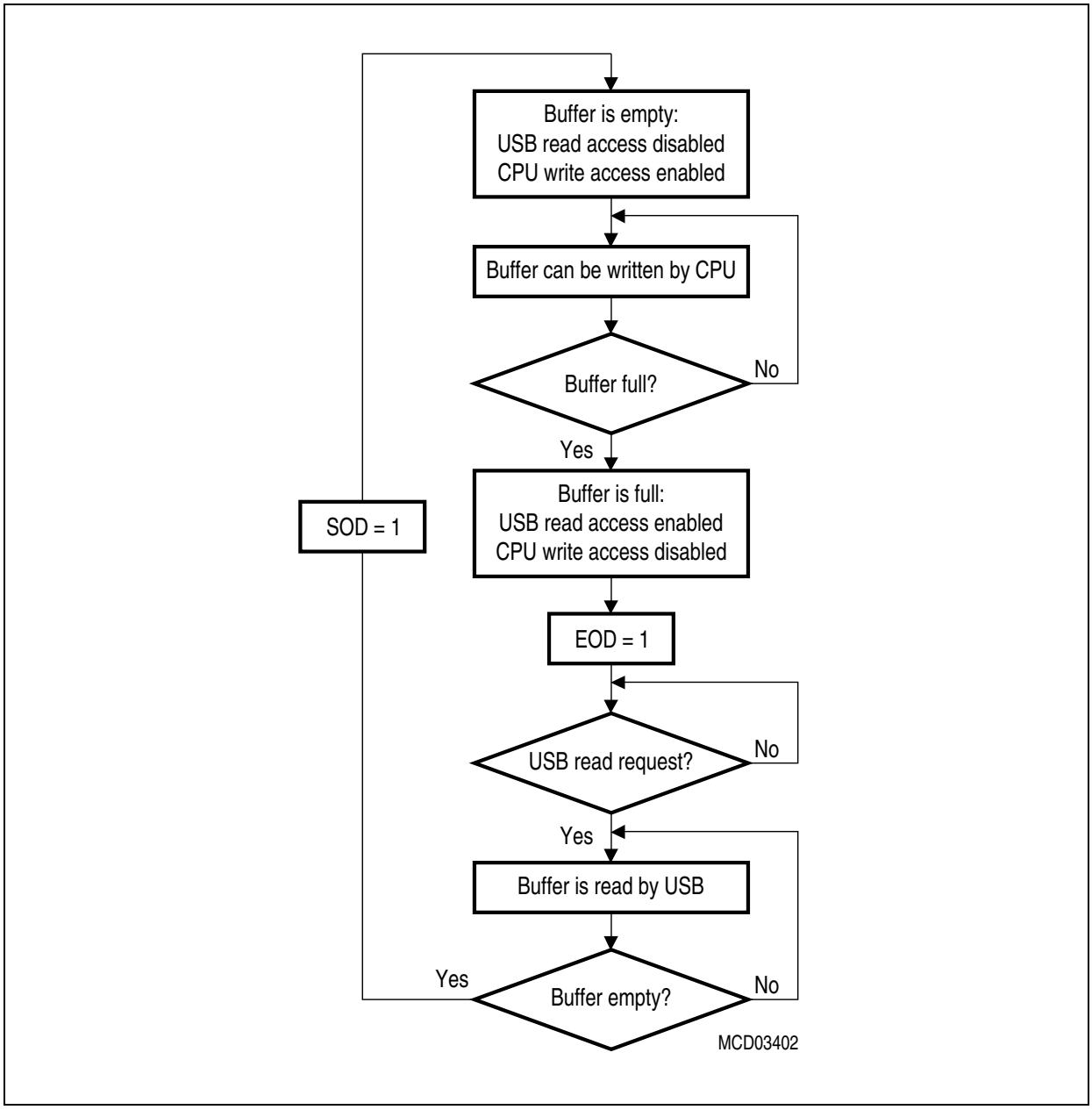


Figure 17 USB Read Access in Single Buffer Mode - Buffer Handling

The standard USB read access as shown in **Figure 18** supports random and sequential CPU access mode of the USB memory. The memory buffer full condition is true when a predefined number of bytes (MaxLen) has been written by the CPU or when bit DONE has been set by software.

After SOF(n) occured at ① with a full USB memory buffer, the USB reads the buffer. Bit SOD is set at the end of the USB buffer read operation at ②, indicating an empty USB memory buffer. Now, the CPU can write again data into the USB memory buffer until a determined number (MaxLen) of bytes are transfered or until bit DONE has been set by software. The MaxLen value must be previously set by software. When the actual USB memory buffer address offset is equal to MaxLen, bit EOD is set at ③ to indicate a full buffer. The USB memory buffer address offset is automatically incremented with every CPU write access to USB memory buffer if bit INCE is set.

During the next frame (after SOF(n+1)) is set at ④) the USB memory buffer can be read by the USB. Bit SOD is set again when the USB memory buffer becomes empty again. If bit DONE is set by the CPU (at ⑤), the buffer is declared by the CPU to be full, even if the address offset does not reach the value of MaxLen.

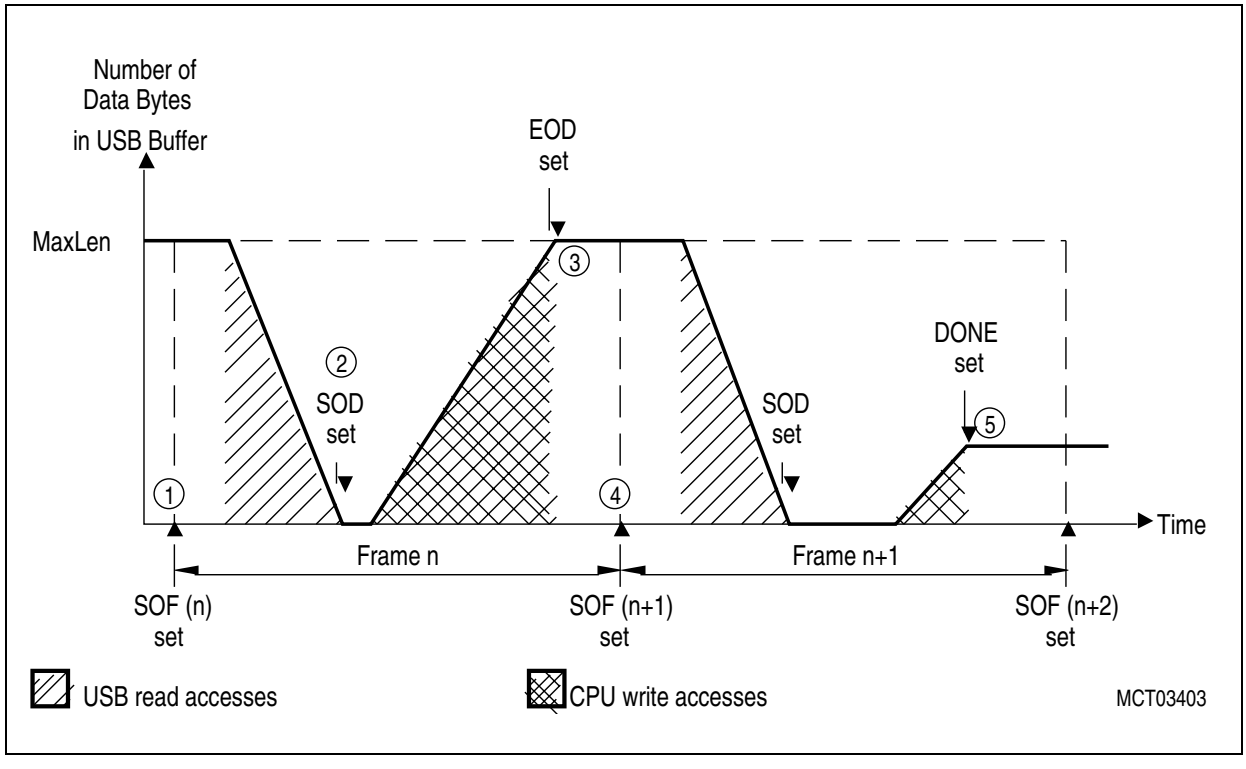


Figure 18 Single Buffer Mode : Standard USB Read Access

The start-of-frame-done enable feature (SOFDE=1) is useful for USB memory read accesses when the number of data bytes to be transferred from CPU to USB is not predictable (see **Figure 19**). The CPU can write data as desired to USB memory until a SOF occures (every 1 ms). The automatic setting of bit SOF causes bit EOD to be set (at ①). This indicates the CPU that no CPU action on this buffer is required until a USB read operation has been finished (bit SOD set at ②). Setting of SOD indicates an empty USB memory to the CPU which can start again writing data into USB memory.

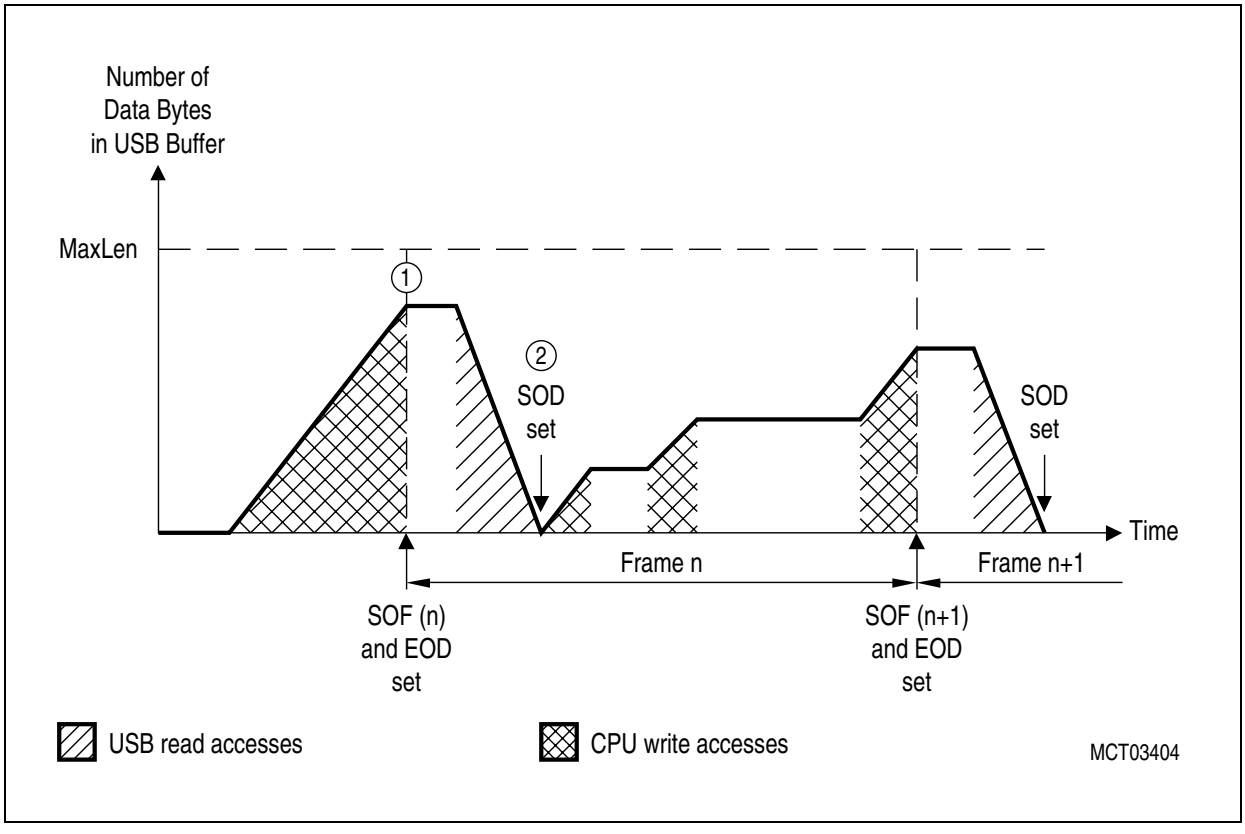


Figure 19 Single Buffer Mode : USB Read Access with Start-of-Frame-Done Enabled

4.2.3 Dual Buffer Mode

In dual buffer mode, both USB memory pages (page 0 and page 1) are used for data transfers. The logical assignment of the memory pages to CPU or USB is automatically switched. The following two figures show the buffer handling concept in dual buffer mode for the USB read access and USB write access.

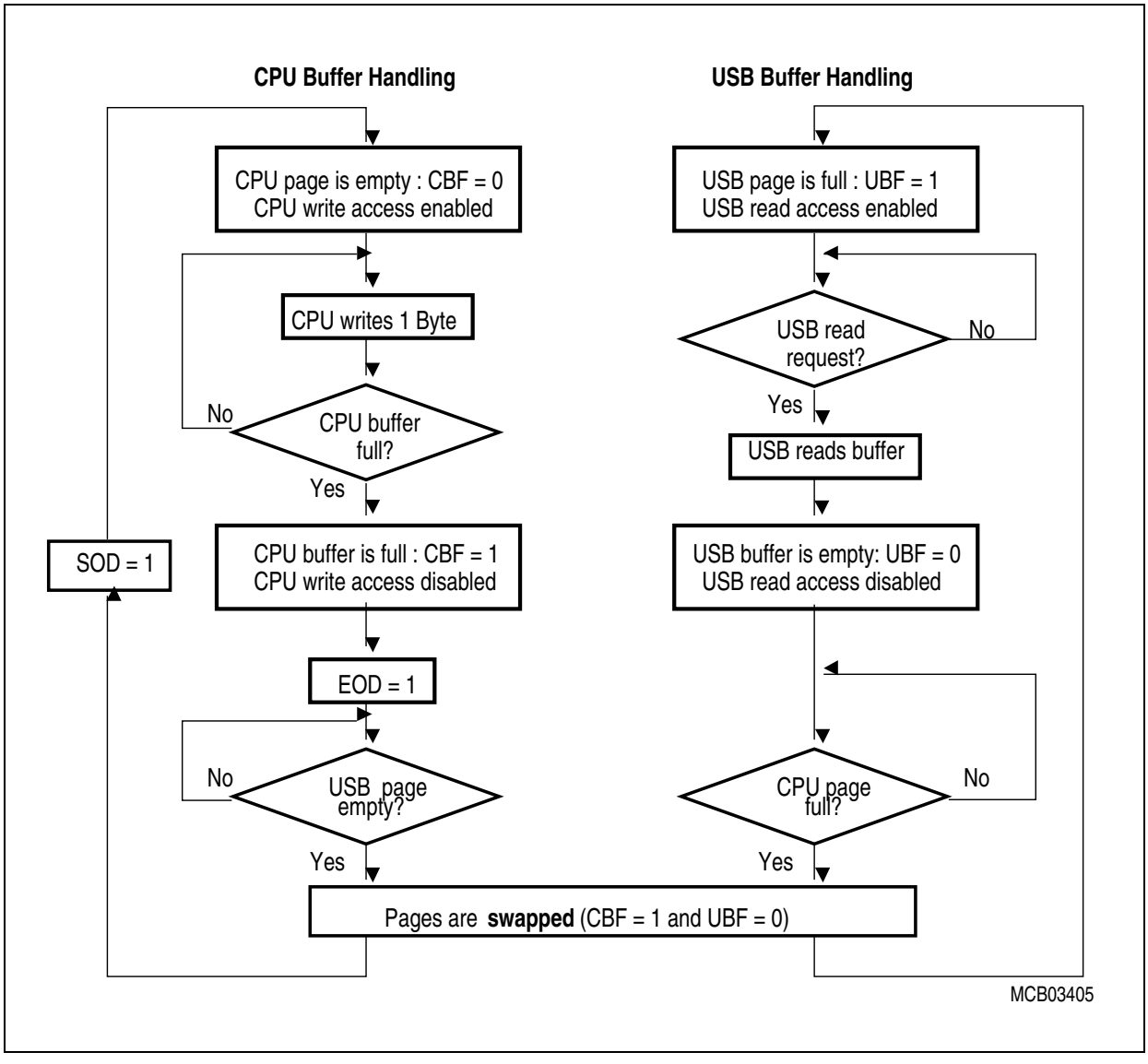


Figure 20 USB Read Access in Dual Buffer Mode - Buffer Handling

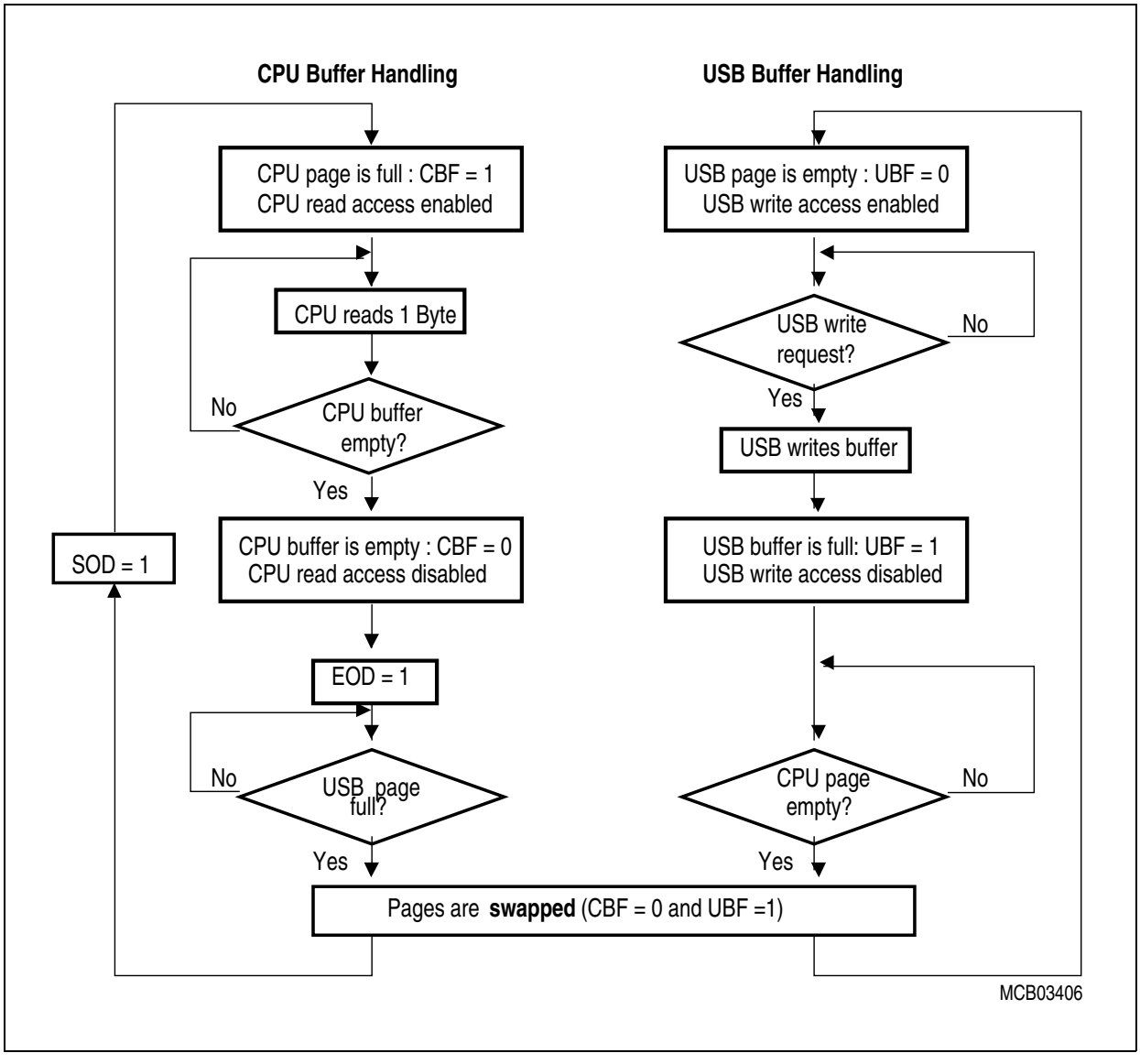


Figure 21 USB Write Access in Dual Buffer Mode - Buffer Handling

Figure 22 describes an example of a USB read operation in sequential mode with both buffers empty at the beginning of the USB read operation.

The CPU starts writing data with sequential access (INCE=1) to the buffer assigned to the CPU at ①. By definition, the buffer is full when MaxLen is reached at ②. The second buffer assigned to the USB is empty (UBF=0) and as a result both buffers are logically swapped. Now the buffer assigned to USB is full (UBF=1) and an USB read access can take place. After the USB read access, the buffer assigned to the USB is empty again with UBF=0. During the USB read access the CPU is still allowed to write into its assigned buffer. When reaching MaxLen at ③, the CPU buffer is full and both buffers are again logically swapped. The USB further execute its read access.

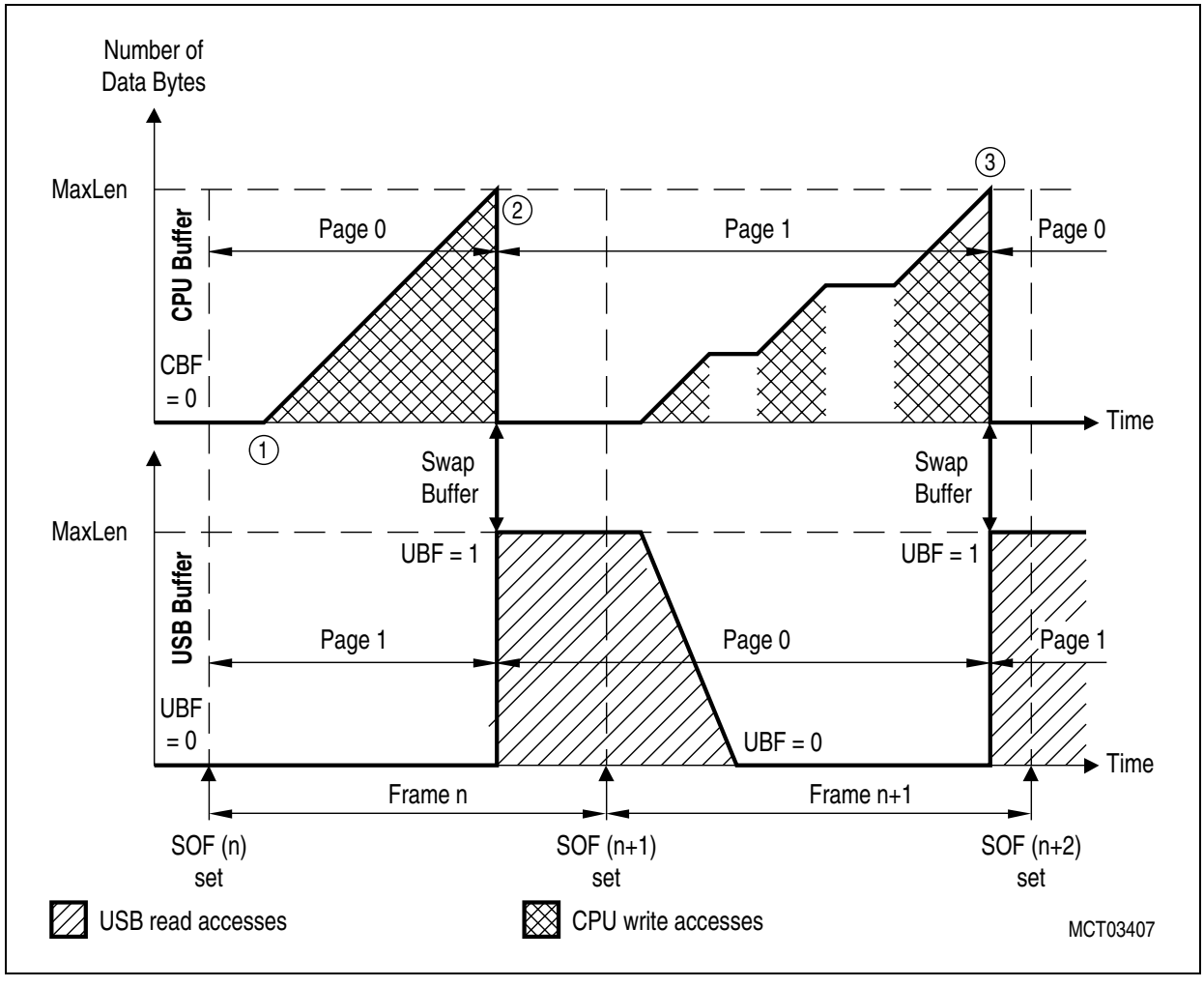


Figure 22 Dual Buffer Mode USB Read Access: Buffer Switching when MaxLen is reached

In dual buffer mode, the physical assignment of the USB memory pages (page 0 or page 1) to either CPU buffer or USB buffer is controlled automatically in the USB module and cannot be selected by software.

Another way to initiate buffer switching is setting bit DONE by software. This feature, which is shown in **Figure 23** for USB read access, can be used to transfer a variable number of bytes. The maximum number of bytes to be transferred is still determined by MaxLen, which is not changed when bit DONE is set. The actual packet length (Len1 or Len2) is the number of bytes which have been written to the buffer before bit DONE is set.

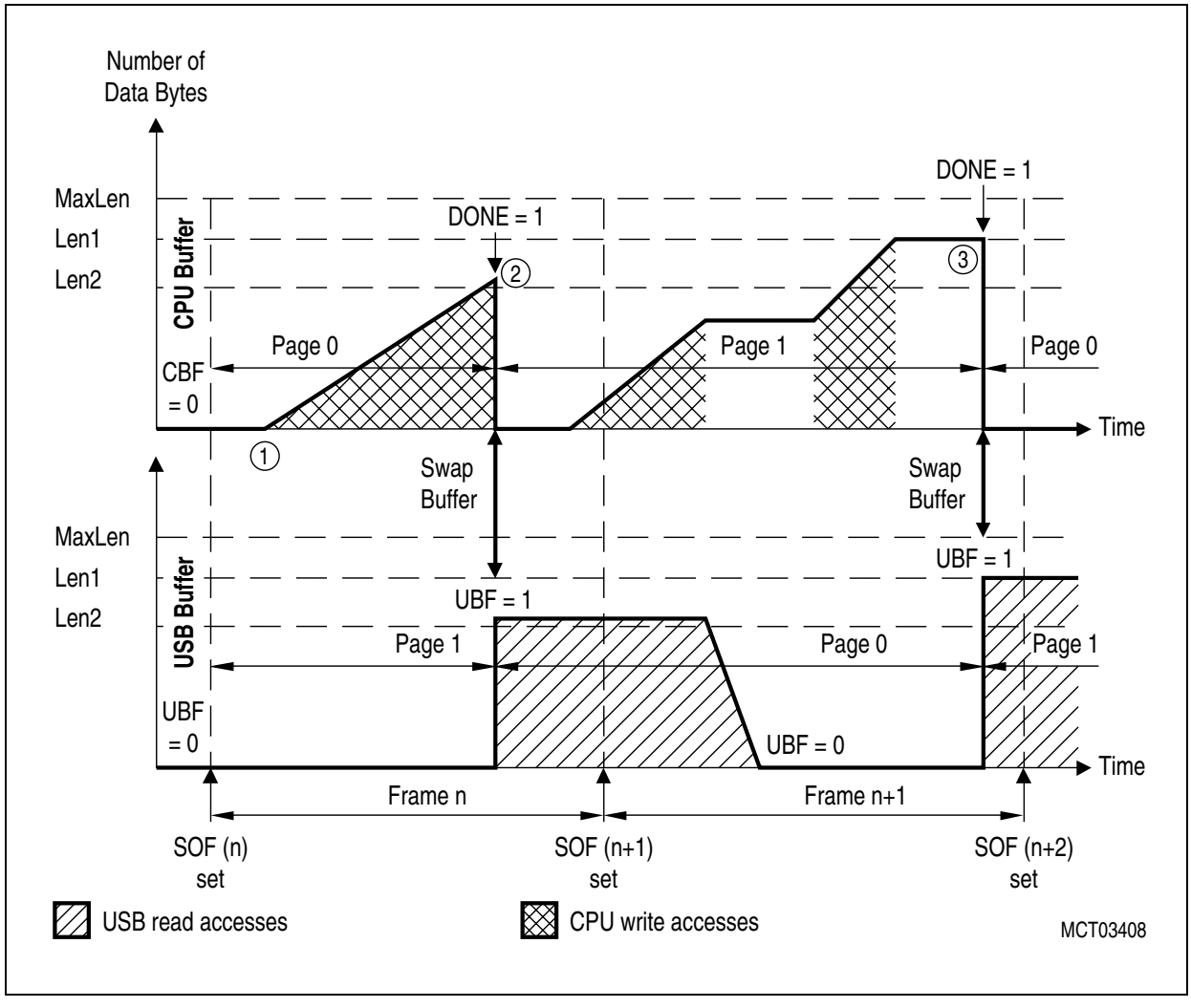


Figure 23 Dual Buffer Mode USB Read Access: Buffer Switching by Setting Bit DONE

If bit SOFDE is set, buffer switching is done automatically after SOF (start of frame) has been detected by the USB. **Figure 24** describes this functionality for USB read access for this case. The buffer which contains the latest data from the CPU is tagged valid for USB access (UBF=1) at ① and the buffers are swapped if the USB buffer is empty. After the USB read access has occured at ②, this buffer assigned to USB is empty again (UBF=0) and can be swapped again as soon as the CPU has filled its buffer (at ③). The number of bytes in the buffer is less or equal MaxLen. The MaxLen threshold is always active, but an occurrence of SOF (if SOFDE=1) or setting bit DONE by software are used to tag the CPU buffer full before reaching MaxLen.

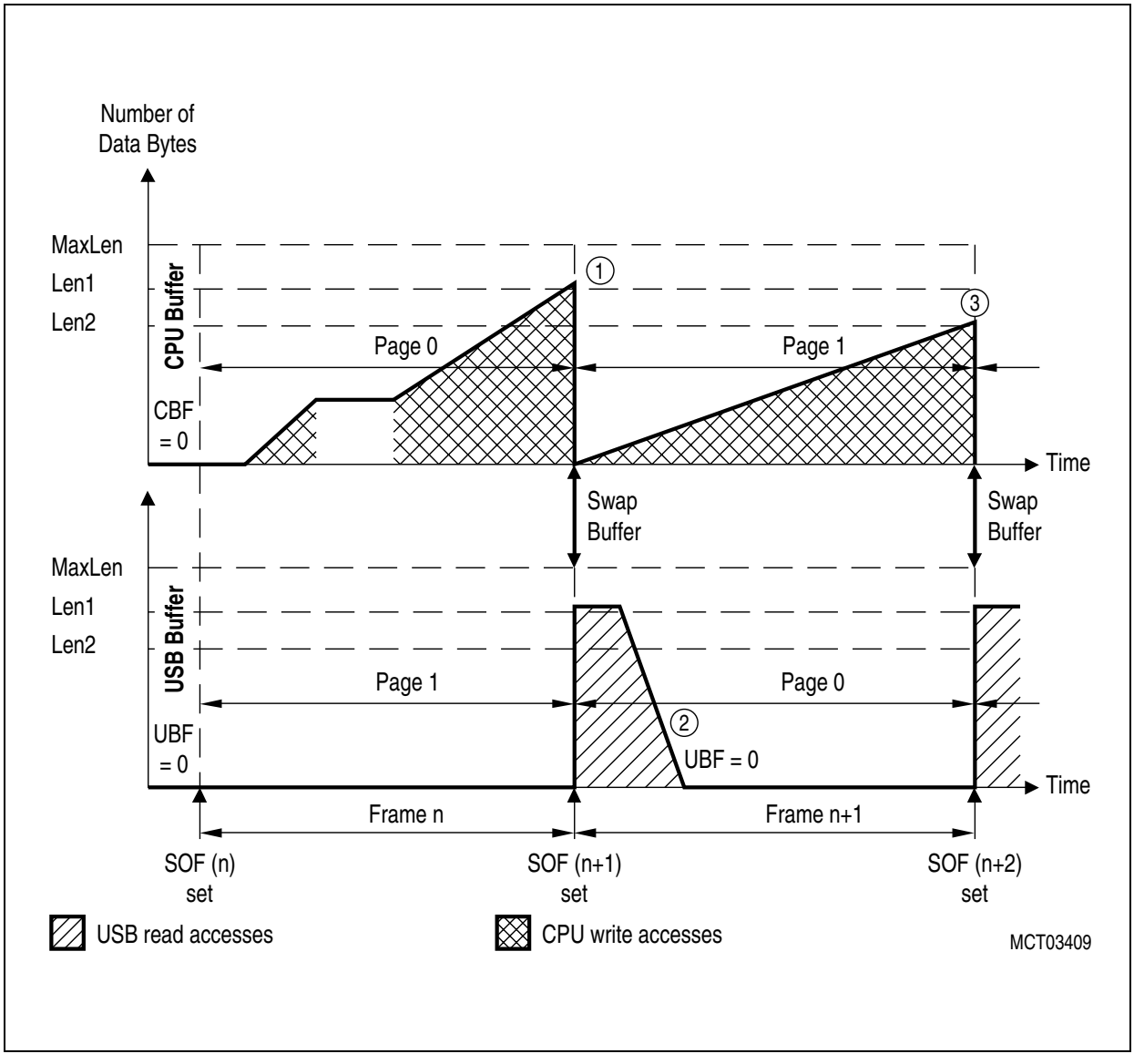


Figure 24 Dual Buffer Mode USB Read Access: Buffer Switching on SOF with SOFDE=1

If the number of data bytes to be transferred is greater than the maximum packet size (given by MaxLen), the data is split up automatically into packets, which are transferred one after the other. **Figure 25** gives an example of an USB read access, where data from the CPU is split up into two packets. When MaxLen is reached during the CPU write access, the currently active buffer is switched to USB side (UBF=1). The CPU continues writing data to the buffer. When the complete data packet has been written to the buffer by the CPU, bit DONE is set by software to indicate the end of the data packet (CBF=1). In the example, the USB buffer has not been read out. It is still full for the USB and can not be swapped (CBF=UBF=1). When the USB read access has occured (CBF=0), the buffers are automatically swapped and bit SOD is set.

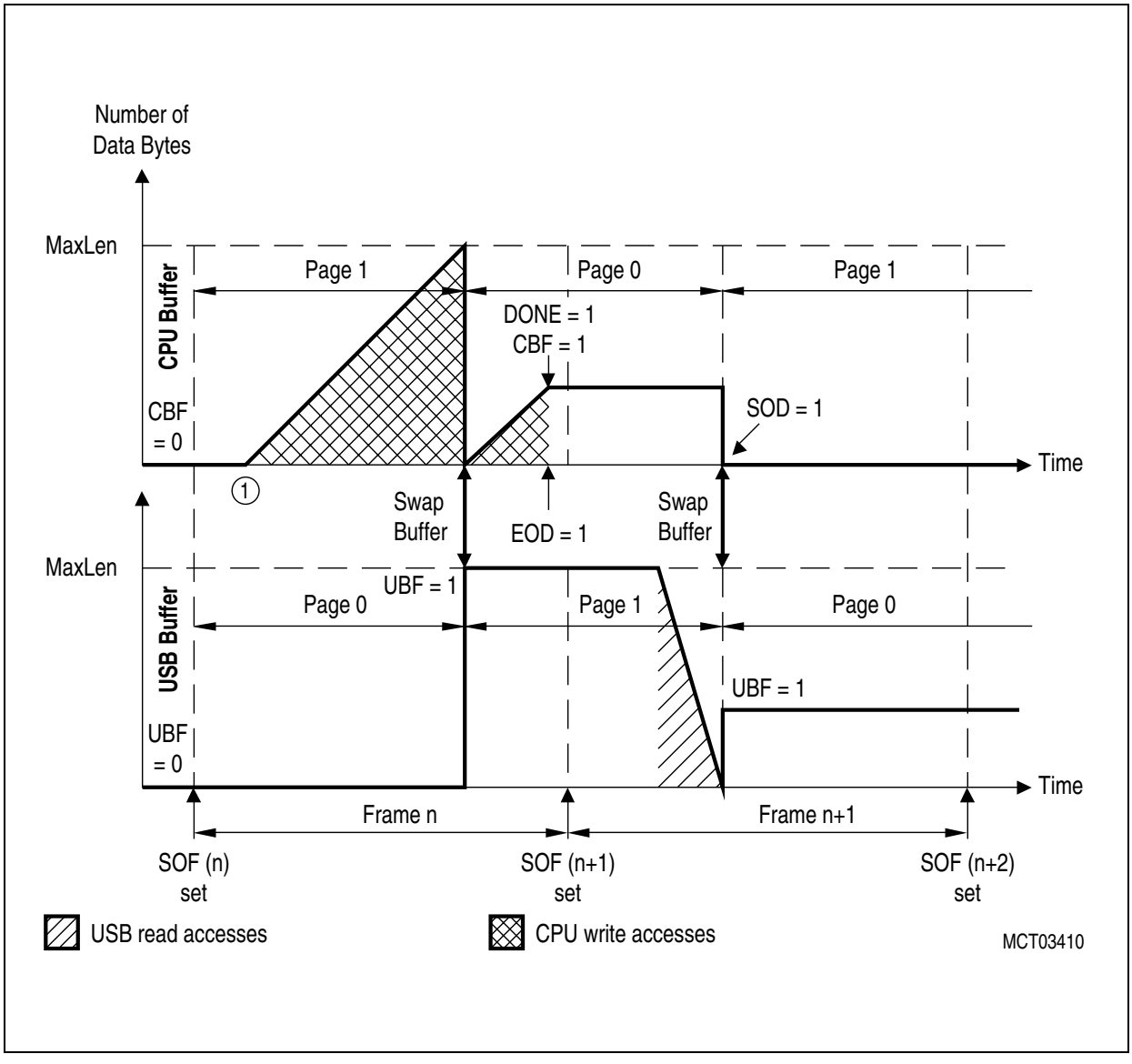


Figure 25 Double Buffer Mode USB Read Access: Data Length greater than Packet Length (MaxLen)

In general, three criteria for buffer switching are implemented in the USB module:

a) For sequential access, the address offset register ADROFF is automatically incremented after each read or write action of the CPU. The address offset value (before incrementing) represents the number of bytes stored in USB memory for a specific endpoint. If the address offset value (after incrementing) reaches the value stored in endpoint length register EPLENn, the currently active buffer is tagged full (USB read access - all bytes have been written by CPU, CBF=1) or empty (USB write access - all bytes have been read by CPU, CBF=0).

b) When Bit DONE, which is located in the endpoint buffer status register EPBSn, is set, software buffer switching is initiated. This action is independent from the number of bytes which have been handled by the CPU (possible in sequential access mode (INCE=1) and random access mode (INCE=0)).
On CPU read accesses, the buffer is declared empty and bit CBF is cleared. If the buffer assigned to the USB is full (UBF=1), the buffers are immediately swapped. In this case, register EPLENn contains the number of received bytes.
On CPU write accesses, two different cases must be distinguished. For random accesses, the number of bytes of one packet is fixed by the value in register EPLENn and does not change. For sequential accesses, the number of written bytes represents the packet size. In this case, the actual value of register ADROFF is transferred to register EPLENn when bit DONE is set.

c) The third criteria for buffer switching is the automatic buffer switching on detection of SOF (see **Figure 25**). This feature can be individually enabled (SOFDE=1) or disabled (SOFDE=0) by software selectively for each endpoint.

4.2.4 Buffer Underrun / Overflow

For the USB transfer modes control, interrupt and bulk, buffer underrun and buffer overflow conditions in the USB module are automatically handled by the UDC using the USB specific low level control mechanism (ACK, NAK).

For isochronous transfer no such control procedures are used. If a buffer underrun condition occurs at isochronous IN transfers (µC has failed to write data fast enough to the buffers), data packets with length "0" are transmitted to the host. In case of buffer overflow at isochronous OUT transfers (µC has failed to read data fast enough from the buffers), new data packets received from the host will be discarded.

4.3 Memory Buffer Organisation

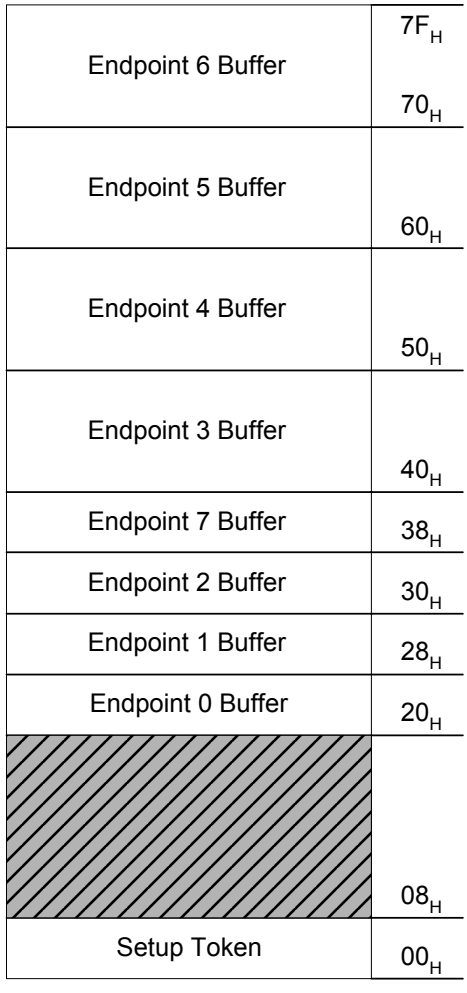
The address generation of the USB memory buffer is based on the address offset and base address pointer. This scheme allows flexible and application specific buffer allocation and management. The length of an endpoint buffer can be up to 8, 16, 32 or 64 bytes. The start address of each endpoint buffer can be located to memory locations according to **Table 12**.

Table 12 Buffer Length and Base Address Values

Buffer Length	Valid Buffer Base Addresses
8 bytes	08_H, 10_H, 18_H, 20_H, 28_H, 30_H, 38_H, 40_H, 48_H, 50_H, 58_H, 60_H, 68_H, 70_H, 78_H
16 bytes	10_H, 20_H, 30_H, 40_H, 50_H, 60_H, 70_H
32 bytes	20_H, 40_H, 60_H
64 bytes	40_H

In order to avoid unused memory space between 2 endpoint buffers, the largest buffer should be located at the highest address. This structure should be used to allocate USB memory for all endpoint buffers. The base address for the setup packet is always located at address 00H. This leads to a typical USB buffer structure as shown in **Figure 26**. In this example, a buffer length of 8 bytes has been allocated to endpoints 0, 1, 2 and 7 while a buffer length of 16 bytes has been reserved for endpoints 3, 4, 5 and 6.

The µC allocates the USB memory buffers for all endpoints by programming the endpoint base address (EPBAn) and the block size (EPLEn) within the total USB buffers.



Buffer Block	EPBAn	EPLENn
Endpoint 6	EPBA6=0E$_H$	EPLEN6=10$_H$
Endpoint 5	EPBA5=0C$_H$	EPLEN5=10$_H$
Endpoint 4	EPBA4=0A$_H$	EPLEN4=10$_H$
Endpoint 3	EPBA3=08$_H$	EPLEN3=10$_H$
Endpoint 7	EPBA7=07$_H$	EPLEN7=08$_H$
Endpoint 2	EPBA2=06$_H$	EPLEN2=08$_H$
Endpoint 1	EPBA1=05$_H$	EPLEN1=08$_H$
Endpoint 0	EPBA0=04$_H$	EPLEN0=08$_H$
Setup Token	Address 00$_H$ on page 0	8 bytes

Note: ADDROFF = 00$_H$

2154_26.vsd

Figure 26 Endpoint Buffer Allocation (Example: 7+1 Endpoints)

4.4 Memory Buffer Address Generation

The generation of a USB memory address for USB access (read or write) depends on the EPNum (endpoint number) information, which has been transmitted to the USB module during the software initialization procedure. The EPNum information is used for the selection of an endpoint specific base address register. As the maximum data packet length of each endpoint can individually be programmed, there are some fixed start addresses for the endpoints. The user program defines the base address for the first data byte of the corresponding endpoint by writing the endpoint specific base address register EPBAn. The length depends on the amount of data to be read or written. The user must take care to assign a buffer space at least as large as the maximum packet size of the endpoint.

The address of the currently accessed byte in the USB memory area of the selected endpoint is defined by an address offset which must be added to the endpoint base address in order to get the correct address for the USB memory buffer. The structure is shown in **Figure 27**.

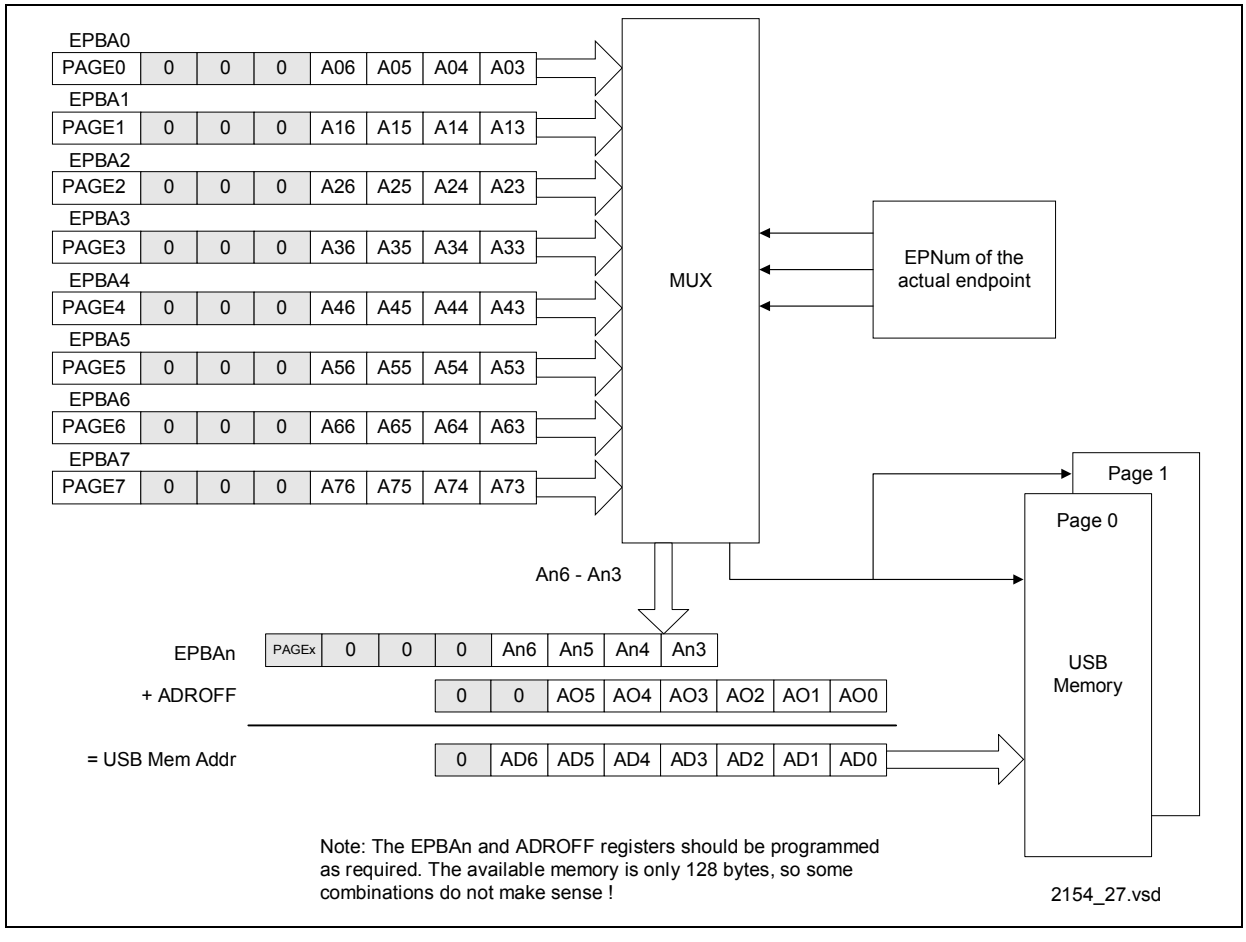


Figure 27 USB Memory Address Generation

4.5 USB Initialization

After a hardware reset operation, bit DCR.UCLK is set to 0. A well defined procedure must be executed for switching on the clock for the USB module. This procedure is described in the Operational Description section. This switch-on procedure after a hardware reset assures proper operation of the USB clock system.

The USB module must be functionally initialized from the µC by writing five configuration bytes for each endpoint to the USBVAL register. **Table 13** shows the 5-byte configuration block which must be transmitted by the µC to the USB module via the USBVAL register for each endpoint. The gray shaded fields have a fixed 0 or 1 value for each endpoint while the white bitfields have to be filled by parameters listed in **Table 14**.

Table 13 USB Configuration Block

	bit 39	bit 38	bit 37	bit 36	bit 35	bit 34	bit 33	bit 32
Byte 0	0	EpNum			0	1	EpInterface	
	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24
Byte 1	EpAltSetting		EpType		EpDir	msb	EPPackSize	
	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16
Byte 2	EPPackSize						lsb	0
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
Byte 3	0	0	0	0	0	EpNum		
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Byte 4	0	0	0	0	0	0	0	0

= Constant data

The five byte USB configuration block must be transferred sequentially (byte 0 to byte 4) from the µC to the USB module for each endpoint beginning with endpoint 0, followed by the USB configuration block for endpoint 1 and so on up to the USB configuration block for endpoint 7. EPNum is set to 000B, 001B,... up to 111B for endpoints 0 up to 7. After this action, bit DINIT is reset by hardware and the software reset and initialization sequence is finished.

This configuration can only be reset by a hardware or software reset (DCR.SWR). A USB reset has no effect on the endpoint configuration, but will only reset the parameters Address, Configuration, Interface and Alternate Setting to its default value 0.

Table 14 Bitfield Definition of USB Configuration Block

Bitfield	Description
EPNUm	This 3-bit field specifies the number of the endpoint (0-7) for which the actual configuration byte block is valid. This 3-bit field must be entered in byte 0 and byte 3 of a configuration byte block
EpInterface	This 2-bit field specifies the number of the interface (0-3) for which the configuration byte block is valid
EpAltSetting	This 2-bit field selects the alternate setting of the corresponding interface (EpInterface) for which the configuration byte block is valid.
EPType	This 2-bit field defines the transfer type of the endpoint 00: Control Endpoint 01: Isochronous Endpoint 10: Bulk Endpoint 11: Interrupt Endpoint Endpoint 0 must be setup as a control endpoint
EPDir	This byte defines the direction of the endpoint 0: Out (packets to be transferred from Host to CPU) 1: In (packets to be transferred from CPU to Host)
EPPackSize	This 10-bit field defines the maximum packet size to be transferred to this endpoint within the range from 0 up to 1023 bytes. The configuration of EPPackSize must be in harmony with the USB specification.

4.6 USB Device Framework

4.6.1 Enumeration Process

The bus enumeration process consists of an interrogation sequence through which the USB host acquires information from the connected device, gives it a unique address, and assigns it a configuration value. In the simplest form, the process takes four steps:

Step 1: The host issues a Get_Descriptor command to the device through the default address using the control pipe. The device then provides information about itself, such as device class, vendor id, maximum packet size for endpoint 0, etc.

Step 2: The host sends a unique address in a data packet to the device using the Set_Address command. The device, under software control, gets the address through endpoint 0 and stores it.

Step 3: The host requests and reads the device configuration descriptor using the Get_Configuration command. The device responds with information about the number of interfaces and endpoints, endpoint transfer type, packet size and direction, maximum power requirements, power source, etc.

Step 4: The last step of the enumeration process is handled using the Set_Configuration command through which the host assigns a configuration value to the device.

After the enumeration process is complete, the device is configured and ready for USB data transmit and receive transactions.

4.6.2 Control Transfers

A control transfer consists of at least two and perhaps three stages. This chapter gives a short description of these stages of a control transfer and the associated control and status bits.

4.6.2.1 Setup Stage

A control transfer always begins with a setup stage that transfers information to a target device, defining the type of request being made to the USB device. The standard commands except the Set_Descriptor, Get_Descriptor and Synch_Frame commands are handled by the USB module automatically without CPU interaction. If the command is not handled by the USB module automatically, a setup interrupt (bit SUI) indicates the end of a setup phase. Additionally, the status and control bits UBF (USB Buffer Full), CBF (CPU Buffer Full) and SOD (Start of Data) are reset.

4.6.2.2 Data Stage

This stage occurs only for requests that require data transfers. The direction of this data stage is always predicted to be from Host to Device (bit EPBSn.DIRn is automatically cleared after the setup stage occurred). The first data packet may immediately be sent

from the Host to the control endpoint according to this configuration of bit EPBSn.DIRn, while NACK will automatically be returned from the Device to the Host in case of a USB read access. It also causes the direction bit to be changed (EPBSn.DIRn=1, USB read access).

The direction of the next transfer can also be predicted under software control (bit EPBSn.SETRDn) to be a USB read access (EPBSn.DIRn=1). This feature is used, if the direction of the data stage is known and the data packet to be transferred from the CPU to the Host is setup before the next USB access occurs.

Therefore, the direction bit must be changed under software control, to be able to transfer the data packet within the first USB read access. Status bit SOD is set under hardware control to indicate valid data to be read by the CPU in case of a USB write access, or data to be written by the CPU in case of a USB read access.

4.6.2.3 Status Stage

The status stage always occurs to report the result of the requested operation. A status stage initiated by the Host, but not terminated according to the configuration of EPBS0.ESP0 (ESP0=0) is indicated by a status interrupt (DIRR.STI). Bit EPBS0.ESP0 has to be set under software control to enable the acknowledge of the status stage.

4.6.3 Standard Device Requests

Table 15 lists the standard requests possible. Only the Set_Descriptor, Get_Descriptor and Synch_Frame requests require intervention from the CPU (DIRR.SUI interrupt). All other standard device requests are handled automatically by the UDC and the CPU only gets a notification. Since the device supports multiple device configurations, interfaces and alternate settings, a separate register CIAR informs the CPU which configuration, interface and alternate setting is active. For further information please refer to the USB Specification Version 1.1, Chapter 9.4 .

Table 15 Standard Device Requests

Request	Description
Get_Status	This request returns status for the specified recipient, which can be a device, interface or endpoint.
Clear_Feature	This request is used to clear or disable a specific feature. The UDC supports this command to clear the Endpoint_Stall feature for all supported logical endpoints and to clear the Device_Remote_ WakeUp feature.

Table 15 Standard Device Requests (cont'd)

Request	Description
Set_Feature	This request is used to set or enable a specific feature. The UDC supports the Set_Feature command to set the Endpoint_Stall feature for all supported logical endpoints and the Device_Remote _WakeUp feature.
Set_Address	This request sets the device address for all future device accesses. Stages after the initial Setup packet assume the same device address as the Setup packet. The USB device does not change its device address until after the Status stage of this request is completed successfully. This is the difference between this request and all other requests. For all other requests, the operation indicated must be completed before the Status stage.
Get_Descriptor *	This request returns the specified descriptor if the descriptor exists. The standard request to a device supports three types of descriptors: DEVICE, CONFIGURATION & STRING. A request for a configuration descriptor returns the configuration descriptor, all interface descriptors, and endpoint descriptors for all the interfaces in a single request. Class-specific and/or vendor-specific descriptors follow the standard descriptors they extend or modify. This request must always be handled by the µC.
Set_Descriptor *	This request may be used to update existing descriptors or add new descriptors. This request must always be handled by the µC.
Get_Configuration	This request returns the current device configuration value. If the returned value is zero, the device is not configured.
Set_Configuration	This request sets the device configuration.
Get_Interface	This request returns the selected alternate setting for the specified interface. This is a valid request only when the device is in the Configured state.

Table 15 Standard Device Requests (cont'd)

Request	Description
Set_Interface	This request allow the host to select an alternate setting for the specified interface.
Synch_Frame *	This request is used to set and then report an endpoint's synchronization frame. When an endpoint supports isochronous transfers, the endpoint may also require per-frame transfers to vary in size according to a specific pattern. The host and the endpoint must agree on which frame the repeating pattern begins. The number of the frame in which the pattern began is returned to the host. This frame number is the one conveyed to the endpoint by the last SOF prior to the first frame of the pattern. The request Sync_Frame is used for isochronous endpoints only. This request must always be handled by the µC.

Note: All 11 standard device requests generate an interrupt via the DSIR register. The requests that always require host intervention are indicated with a " * " in the table above (Get_Descriptor, Set_Descriptor and Sync-Frame).
All other 8 requests are automatically handled by the UDC, but they are transparent to the µC (the request Get_Status can be initialized by the µC if required, the remaining 7 requests cannot be controlled by the µC).

4.7 Onchip USB Transceiver

The SIUC-X provides onchip receiver and transmitter circuitries which allows to connect the SIUC-X directly to the USB bus. The USB driver circuitry is shown in **Figure 28**. The USB transceiver is capable of transmitting and receiving serial date at full speed (12 MBits/s) data rate. Transceiver and receiver can be separately disabled for power down mode operation. A single ended zero error condition (D+ and D- both at low level) can be detected.

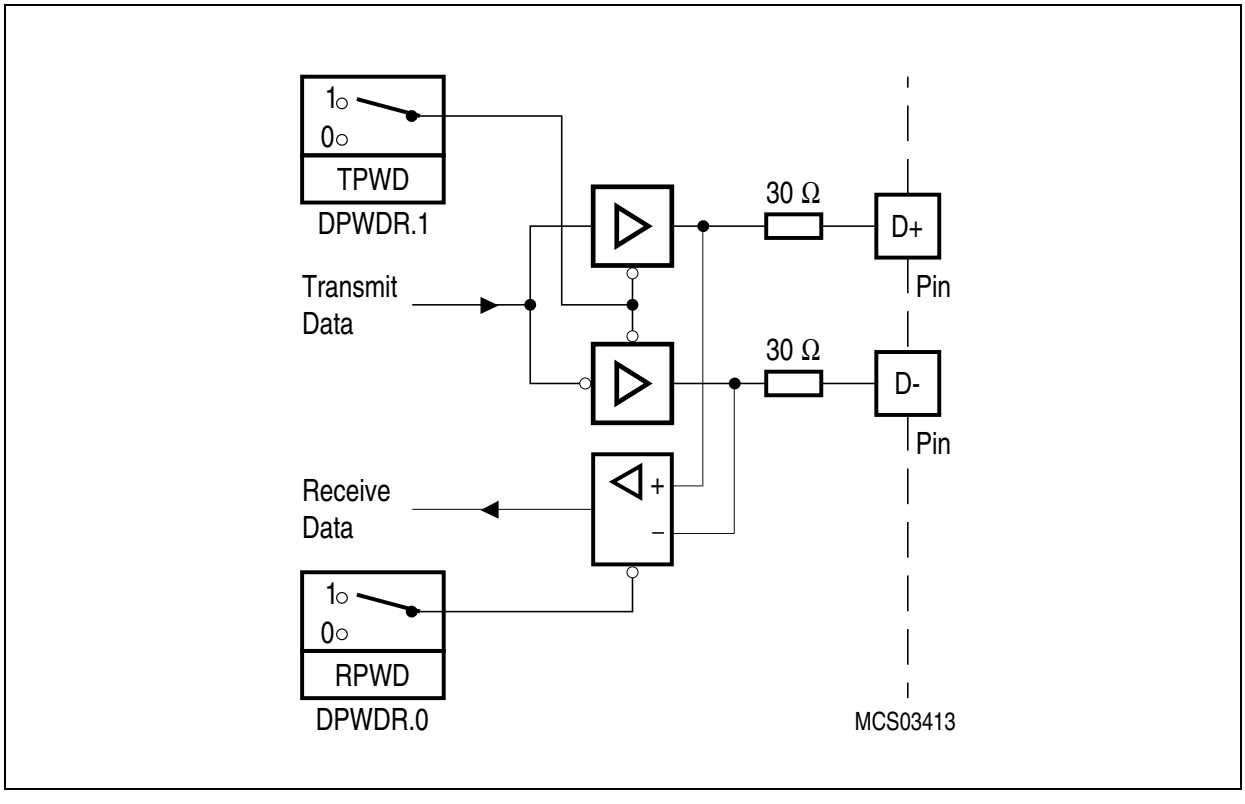


Figure 28 USB Onchip Driver Circuitry

The USB driver circuitry is a differential output driver which drives the USB data signal onto the cable of the USB bus. The static output swing of the transmitter is in low state below 0.3 V with a 1.5 kΩ load to 3.6 V and in high state above 2.8 V with a 15 kΩ load to V_{SS}. The driver outputs support tri-state operation to achieve bi-directional half duplex operation (control bits RPWD and TPWD). High impedance is also required to isolate the port from devices that are connected but powered down. The driver tolerates exposure to waveforms as specified in chapter 7.1 of the USB V1.1 Specification..

For a full speed USB connection the impedance of the USB driver must be between 29 Ω and 44 Ω. The data line rise and fall times are between 4 ns and 20 ns, smoothly rising or falling (monotonic), and are well matched to minimize RFI emissions and signal skew. **Figure 29** shows how the full speed driver is realized using two identical CMOS buffers. **Figure 29** shows the full speed driver signal waveforms.

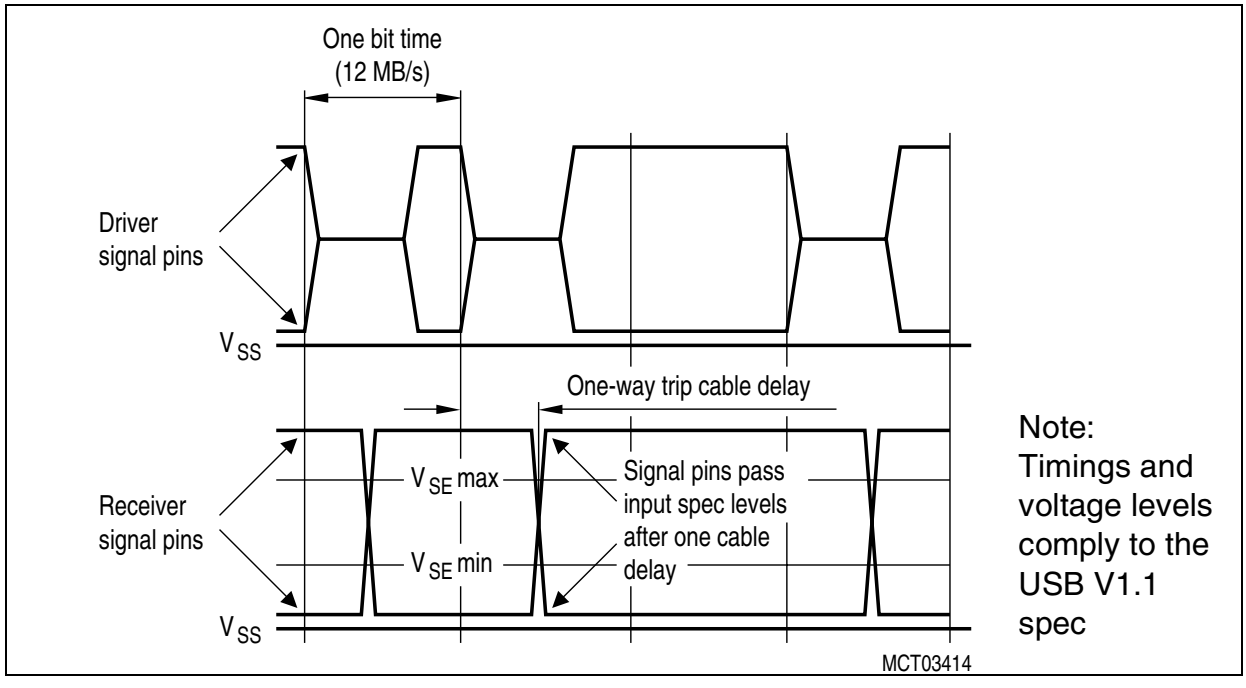


Figure 29 Full Speed USB Driver Signal Waveforms

Generally, full speed and low speed USB devices are differentiated by the position of the pullup resistor on the downstream end of the cable. Full speed devices are terminated with the pullup on the D+ line and low speed devices are terminated with the pullup in the D- line.

As the SIUC-BA is a full speed device an external pull-up resistor (R_2) must be connected to the D+ line as shown in **Figure 30**. The pullup terminator is a 1.5 kΩ resistor tied to a voltage source between 3.0 and 3.6 V referenced to the local ground. In some cases it might be necessary to hide the USB device from the host even when it is plugged in. To accomplish this, the pull-up resistor R_2 can be made switchable with a port a transistor.

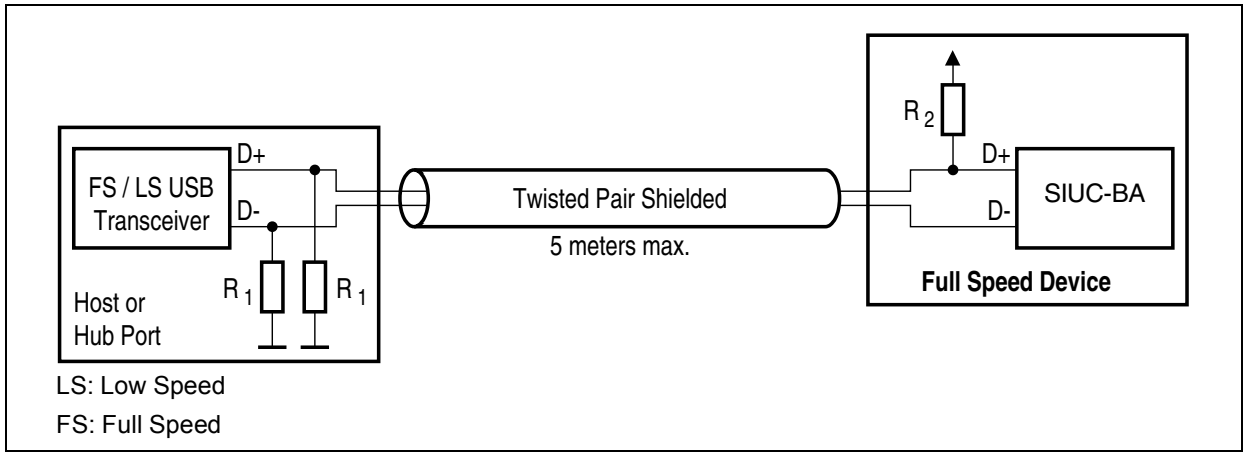


Figure 30 High Speed Device Cable and Resistor Connection

4.8 Detach / Attach Detection and USB Power Modes

The USB device can be used in two different modes concerning its power supply, the bus-powered mode and the self-powered mode.

4.8.1 Self-Powered Mode

In self-powered mode, the USB device has its own power supply. The USB device has to detect whether it is connected to USB bus or not. This detection is done by hardware by using the Device-Attached Device-Detached pin DADD as shown in **Figure 31**. Bit DCR.DA reflects the state of pin DADD. When the device-attached condition is detected, bit DA is set and a Device-Attached Interrupt (DIRR.DAI) can be generated if required. The interrupt service routine of this device interrupt must completely initialize the USB device/module. The device-detached detection resets bit DA, sets bit DIRR.DDI (Device-Detached Interrupt) and can generate a device interrupt, too.

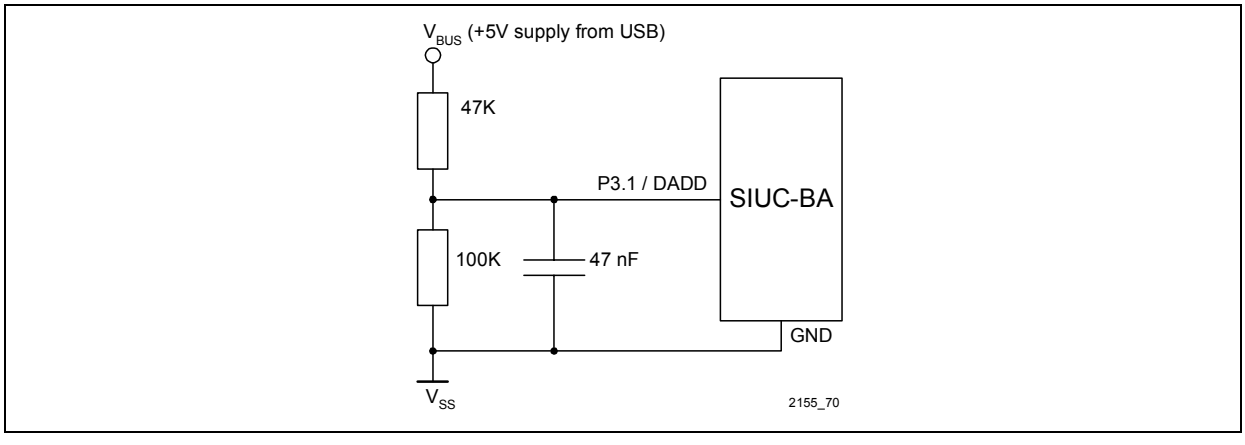


Figure 31 Device Attached - Device Detached Detection in Self-Powered Mode

4.8.2 Bus-Powered Mode

In bus-powered mode, the USB device is driven by the power supply from the USB bus. The maximum power consumption is given by the USB specification, i.e. the power consumption of the total system must not exceed 500 µA in suspend mode and 100mA (for low power devices) in operational mode.

An explicit device-attached detection in this mode is not necessary. If the CPU is running, the device is attached, so the USB device/module has to be configured only after power-on. The device-detach action has no significance concerning software, because the device is no longer powered and the CPU stops. As a result, no attach-detach detection is needed. In this mode, pin DADD can be used as standard IO pin with bit DA monitoring its status and the interrupt generation on DA should not be used. If the interrupt generation on bit DA remains activated, a request must not be interpreted as attached-detached action, but as an external interrupt request on pin DADD, which is generating a device interrupt.

4.9 USB Registers

Two different kinds of registers are implemented in the USB module.

1.) The global registers describe the basic functionality of the complete USB module and can be accessed via unique SFR addresses. These are the:

- GEPIR (Global Endpoint Interrupt Request Register)
- GESR (Global Endpoint Stall Register)
- EPSEL (Endpoint Select Register)
- ADROFF (Address Offset Register)
- USBVAL (USB Value Register)
- CIAR (Configuration Request Register)
- CIARI (Configuration Request Interrupt Register)
- CIARIE (Configuration Request Interrupt Enable Register)
- IFCSEL (Interface Select Register)

2.) To reduce the number of SFR addresses needed to control the USB module, device registers and endpoint registers are mapped into an SFR address block of nine SFR addresses ($C1_H$ to CC_H) for device and seven SFR addresses ($C1_H$ to CA_H) for endpoint. The endpoint specific functionality of the USB module is controlled via the device registers:

- DCR (Device Control Register)
- DPWDR (Device Power Down Register)
- DIER (Device Interrupt Enable Register)
- DIRR (Device Interrupt Request Register)
- FNRH, FNRL (Frame Number High/Low Registers)
- DSIR (Device Setup Interrupt Register)
- DGSR (Device Get Status Register)
- IGSR (Interface Get Status Register)

An endpoint register set is available for each endpoint (n=0...7) and describes the functionality of the selected endpoint. **Figure 32** explains the structure of the USB module registers.

Note: In the description of the USB module registers, bits are marked as rw, r or w. Bits marked as rw can be read and written. Bits marked as r can be read only. Writing any value to r bits has no effect. Bits marked as w are used to execute internal commands which are triggered by writing a 1. Writing a 0 to w bits has no effect. Reading w bits returns a 0.

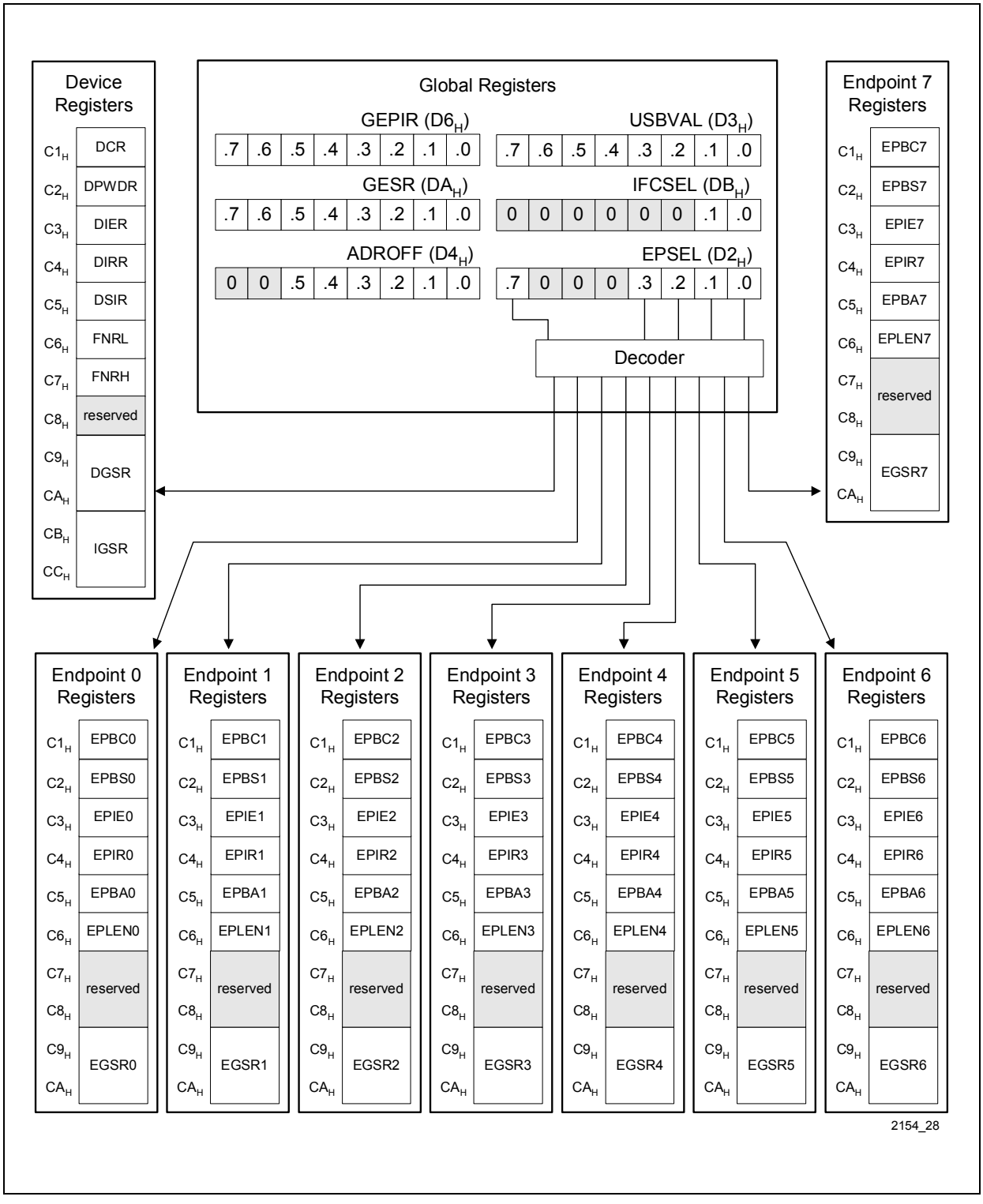


Figure 32 USB Register Set

For each of the 4 interfaces (besides the default interface with Endpoint 0) there is a 16-bit register that contains a status value which is transmitted to the USB host upon a Get_Status request (**Figure 33**).

For setting the Interface Get Status Register (IGSR) for a specific interface, the µC first selects the interface number in the Interface Select Register (IFCSEL) and then writes the status value for the corresponding interface to IGSR (Note: IGSR can only be accessed with EPSEL=80_H). The status value itself (contained in IGSR) is fully specified by the USB spec. There is no function defined for that yet in USB V1.1, however SIUC-BA fully supports the Get_Status request for potential changes in future.

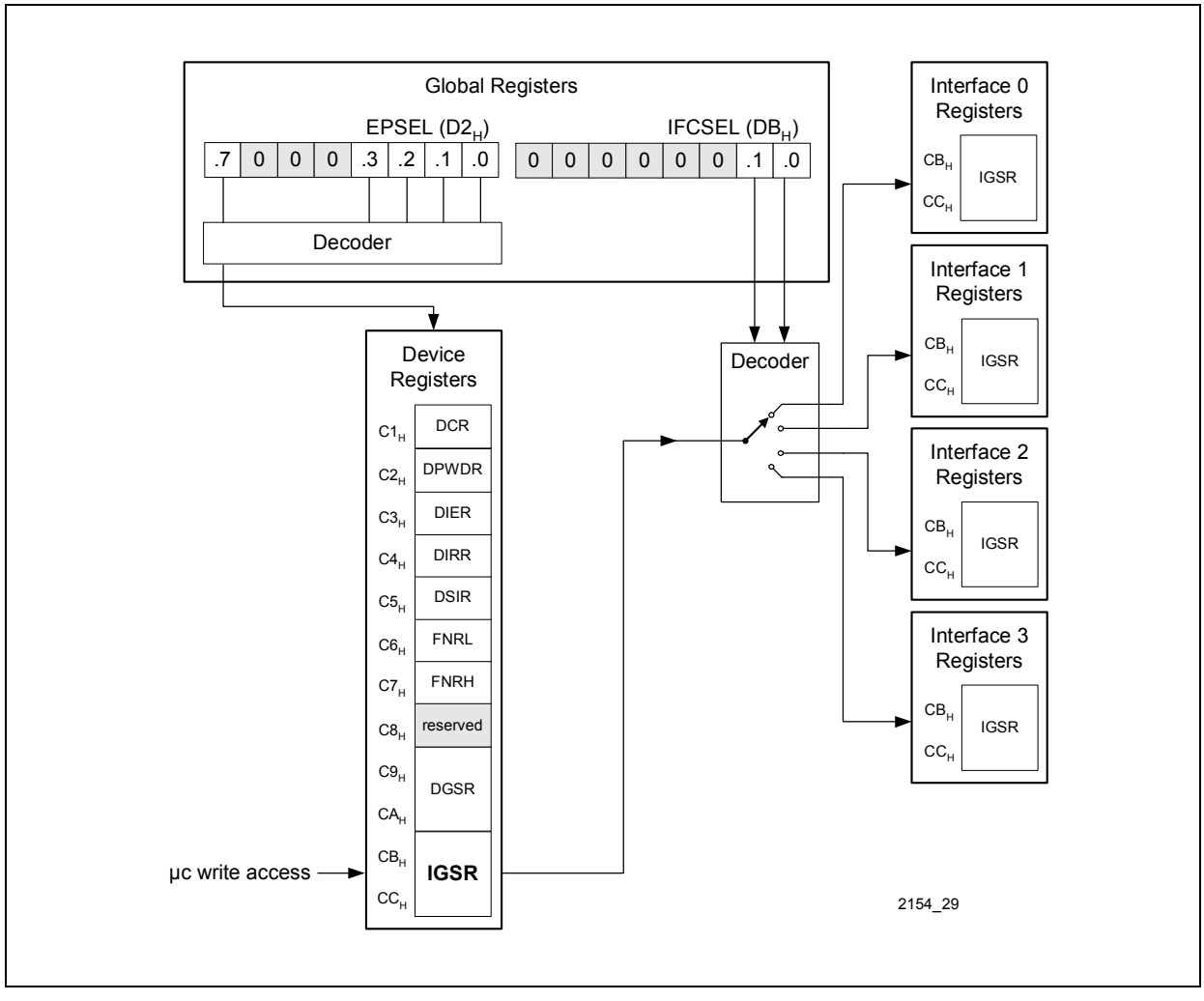


Figure 33 USB Interface Get_Status Registers

Global Registers

The global registers **GEPIR, GESR, EPSEL, ADROFF, IFCSEL** and **USBVAL** describe the global functionality of the USB module and can be accessed via unique SFR addresses. The **Global Endpoint Interrupt Request Register (GEPIR)** is described in the section on interrupts (**Chapter 6**), the other global registers are described below.

Standard Command Registers

All 11 standard device requests generate an interrupt via the DSIR register. The requests Get_Descriptor, Set_Descriptor and Sync-Frame always require host intervention. All other 8 requests are automatically handled by the UDC, but they are transparent to the µC (the request Get_Status can be initialized by the µC if required, the remaining 7 requests cannot be controlled by the µC at all).

Since the device supports multiple device configurations, interfaces and alternate settings, a separate register CIAR is needed to inform the device which configuration, interface and alternate setting is active. Any changes in the CIAR register through Set_Configuration or Set_Interface will generate an interrupt in the **Configuration Request Interrupt Register (CIARI)**. This interrupt can be enabled via a bit in the CIARIE register. These registers are described in the section on interrupts (**Chapter 6**).

Device Registers

The device registers can only be accessed when the endpoint select register (adr. $D2_H$) is set to EPSEL = 80_H. The **Device Interrupt Enable Register (DIER), the Device Interrupt Request Register (DIRR) and the Device Setup Interrupt Register (DSIR)** are described in the section on interrupts (**Chapter 6**), all other device registers are described below.

Endpoint Registers

Each of the 8 endpoints has its own endpoint register set. The **Endpoint Interrupt Enable Register (EPIE)** and the **Endpoint Interrupt Request Register (EPIR)** are described in the section on interrupts (**Chapter 6**), all other endpoint registers are described below.

4.9.1 GESR- Global Endpoint Stall Register

Reset value: 00_H Address: DA_H

7	6	5	4	3	2	1	0
EPST7	EPST6	EPST5	EPST4	EPST3	EPST2	EPST1	EPST0
r	r	r	r	r	r	r	r

Bit	Function
EPST7-0	**Endpoint x Stalled** EPSTx indicates to the µC if the corresponding endpoint is stalled (’1’) or not (’0’).

The GESR register contains the least significant bits of the EGSR registers of all 8 endpoints.

4.9.2 EPSEL - Endpoint Select Register

Reset value: 80_H Address: $D2_H$

7	6	5	4	3	2	1	0
EPS7	0	0	0	0	EPS2	EPS1	EPS0
rw	r	r	r	r	rw	rw	rw

Bit	Function
EPS7 EPS2 EPS1 EPS0	**Endpoint / Device Register Block Select Bits** These five bits select the active register block of endpoint or device registers. 1XXX: Device register set selected 0000: Endpoint 0 register set selected 0001: Endpoint 1 register set selected 0010: Endpoint 2 register set selected 0011: Endpoint 3 register set selected 0100: Endpoint 4 register set selected 0101: Endpoint 5 register set selected 0110: Endpoint 6 register set selected 0111: Endpoint 7 register set selected

4.9.3 IFCSEL - Interface Select Register

Reset value: 00_H Address: DB_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	IF1	IF0
r	r	r	r	r	r	rw	rw

Bit	Function
IF1,0	**Interface Select Bits** These two bits are used to select the interface number for setting the corresponding 16-bit status value in IGSR. 00: interface 0 01: interface 1 10: interface 2 11: interface 3

Also see **Figure 33**.

4.9.4 USBVAL - USB Data Register

Reset value: 00_H Address: $D3_H$

7	6	5	4	3	2	1	0
.7	.6	.5	.4	.3	.2	.1	.0
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Function
USBVAL.7 - USBVAL.0	**USB Data Value** USBVAL stores the 8-bit data byte during transfers from µC to USB memory and from USB memory to the µC. Bit NOD in the EPIRn register indicates when the µC processes a USBVAL read operation with an empty USB buffer or a USBVAL write operation to a full USB buffer.

The data transfers between USB memory and the µC (C800 CPU) are handled via the SFR USBVAL. With a µC write access to USBVAL, the value written into it is transferred to the USB memory location defined by the content of the endpoint specific base address register EPBAn and the address offset register ADROFF. During USB memory read accesses by the µC, data is written in the reverse direction. Access to USBVAL is only successful if either EPBSn.DIRn=0 and CBF=1 (USB write operation) or EPBSn.DIRn=1 and CBF=0 (USB read operation).

4.9.5 ADROFF - Address Offset Register

Reset value: 00_H Address: $D4_H$

7	6	5	4	3	2	1	0
0	0	AO5	AO4	AO3	AO2	AO1	AO0
r	r	rw	rw	rw	rw	rw	rw

Bit	Function
AO5 - AO0	**USB Address Offset** ADROFF stores the 6-bit offset address for USB memory buffer addressing by the µC.

In most cases the µC accesses only one endpoint buffer until it is full (CBF=1 during µC write access) or empty (CBF=0 during µC read access). As the USB memory size is 128 bytes per page, the maximum packet length is limited to 64 bytes. Therefore, only the lowest 6 bits of ADROFF (AO5...AO0) are required for offset definition. A write operation to ADROFF is only successful if either EPBSn.DIRn=0 and CBF=1 (USB write operation) or EPBSn.DIRn=1 and CBF=0 (USB read operation).

4.9.6 CIAR - Configuration Request Register

Reset value: 00_H Address: $D9_H$

7	6	5	4	3	2	1	0
0	0	CFG	0	IFC1	IFC0	0	AS
r	r	r	r	r	r	r	r

Bit	Function
CFG	**Configuration Value** 0: Unconfigured 1: Indicates configuration setting 1.
IFC1-0	**Interface Number** IFC indicates the interface number for this configuration setting. A maximum of 4 interfaces (excluding interface 0) per configuration are possible.
AS	**Alternate Setting** AS indicates the selected alternate setting for that interface (IFC1-0). 0: Alternate Setting 0 1: Alternate Setting 1

This register is used for the standard requests Set_Configuration and Set_Interface.

4.9.7 DCR - Device Control Register

The device control register includes control and status bits which indicate the current status of the USB module and the status of the USB bus. This register can only be accessed when the endpoint select register (adr. $D2_H$) is set to EPSEL = 80_H.

Reset value: $000X0000_B$ Address: $C1_H$

7	6	5	4	3	2	1	0
0	DA	SWR	SUSP	DINIT	RSM	UCLK	0
r	r	rw	r	r	rw	rw	r

Bit	Function
DA	**Device Attached** Bit DA reflects the state of pin DADD, which can be used to indicate whether the device is attached to the USB bus or not in self-powered mode. If pin DADD is 0, bit DA=0. If pin DADD is 1, bit DA=1. If the "Device-Attached/Device-Detached" feature is not required (e.g. in bus-powere mode), pin DADD and bit DA can be used as general purpose input (for further information see **chapter 4.8**).
SWR	**Software Reset** Setting bit SWR initiates a software reset operation of the USB device. This bit is cleared by hardware after a successful reset operation. SWR can not be reset by software.
SUSP	**Suspend Mode** This bit is set when the USB is idle for more than 3 ms. It will remain set until there is a non idle state on the USB cable or when bit RSM is set. In addition to SUSP an interrupt can be generated when the suspend mode begins (DIRR.SBI) and when it ends (DIRR.SEI)
DINIT	**Device Initialization in Progress** At the end of a software reset, bit DINIT is set by hardware. After software reset of the USB module, it must be initialized by the CPU. When DINIT is set after a software reset, 5 bytes for each endpoint must be written to SFR USBVAL. After the 40th byte, bit DONE0 has to be set by software. Bit DINIT is reset by software after a successful initialization sequence.

RSM	**Resume Bus Activity** When the USB device is in suspend mode, setting bit RSM resumes bus activity. In response to this action, the USB will deassert the suspend bit and perform the remote wake-up operation. Writing 0 to RSM has no effect, the bit is reset if bit SUSP is 0.
UCLK	**UDC Clock Selection** Bit UCLK controls the functionality of the USB core clock. If UCLK=0, the 48 MHz USB core clock is disabled. If UCLK=1, the 48 MHz USB core clock is enabled.

4.9.8 DPWDR - Device Power Down Register

The device power down register (DPWDR) includes 2 bits which allow to switch off the USB transmitter and receiver circuitry selectively for power down mode operation. This register can only be accessed when the endpoint select register (adr. $D2_H$) is set to EPSEL = 80_H.

Reset value: 00_H Address: $C2_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	TPWD	RPWD
r	r	r	r	r	r	rw	rw

Bit	Function
TPWD	**USB Transmitter Power Down** Setting bit TPWD puts the USB transmitter into power down mode. After a wake-up from software power down mode, bit TPWD must be cleared by software to enable data transmission again.
RPWD	**USB Receiver Power Down** Setting bit RPWD puts the USB receiver into power down mode. After a wake-up from software power down mode, bit RPWD must be cleared by software to enable data reception again. Note: If RPWD is set, the USB bus can not wake-up the device from power down mode.

4.9.9 FNRH / FNRL - Frame Number Register High / Low Byte

The frame number registers store an 11-bit value which defines the number of a USB frame. The frame number rolls over upon reaching its maximum value of $7FF_H$. The FNRH/FNRL registers are read only registers which are reset to 00_H by a hardware reset. These registers can only be accessed when the endpoint select register (adr. $D2_H$) is set to EPSEL = 80_H.

Reset value: $00000XXX_B$ Address: $C7_H$

	7	6	5	4	3	2	1	0
FNRH	0	0	0	0	0	FNR10	FNR9	FNR8
	r	r	r	r	r	r	r	r

Reset value: XX_H Address: $C6_H$

	7	6	5	4	3	2	1	0
FNRL	FNR7	FNR6	FNR5	FNR4	FNR3	FNR2	FNR1	FNR0
	r	r	r	r	r	r	r	r

Bit	Function
FNR10 - FNR0	**Frame Number Value** FNR10-8 and FNR7-0 hold the current 11-bit frame number of the latest SOF token.

4.9.10 DGSR - Device Get_Status Register

These two registers hold a 16-bit value that is sent to the USB host upon a "Device Get Status" command. These registers can only be accessed when the endpoint select register (adr. $D2_H$) is set to EPSEL = 80_H.

Reset value: 00_B Address: CA_H

7	6	5	4	3	2	1	0
DST15	DST14	DST13	DST12	DST11	DST10	DST9	DST8
rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: $C9_H$

7	6	5	4	3	2	1	0
DST7	DST6	DST5	DST4	DST3	DST2	RWUP	PSTAT
rw	rw	rw	rw	rw	rw	r	rw

Bit	Function
RWUP	**Remote Wakeup (Set/Cleared by the USB host)** The remote wakeup status is configured by the host using the Set_Feature_Remote_Wakeup and the status is returned to the host with every Get_Status request. 0: remote wakeup disabled 1: remote wakeup enabled This bit cannot be written by the µC.
PSTAT	**USB Power Status** This bit indicates to the host whether the USB device is operating in bus-powered mode ("0") or in self-powered mode ("1"). This bit has no effect on the functions of the device.

The function of this register is completely determined by the USB specification. For further functional additions the higher bits of this register (DST15-2) can be programmed to "1" to fulfill future requirements.

4.9.11 IGSR - Interface Get_Status Register

These two registers hold a 16-bit value that is sent to the USB host upon an "Interface Get Status" command. These registers can only be accessed when the endpoint select register (adr. $D2_H$) is set to EPSEL = 80_H. The interface number (0...3) is selected in IFCSEL.

Reset value: 00_B Address: CC_H

7	6	5	4	3	2	1	0
IST15	IST14	IST13	IST12	IST11	IST10	IST9	IST8
rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: CB_H

7	6	5	4	3	2	1	0
IST7	IST6	IST5	IST4	IST3	IST2	IST1	IST0
rw	rw	rw	rw	rw	rw	rw	rw

The function of this register is completely determined by the USB specification. For further functional additions the bits of this register (IST15-0) can be programmed to "1" to fulfill future requirements. Also see **Figure 33**.

4.9.12 EPBCn - Endpoint Buffer Control Register

The endpoint buffer control register controls the endpoint specific operations. The index n corresponds to the selected endpoint.

Reset value: 00_H Address: $C1_H$

7	6	5	4	3	2	1	0
STALLn	0	0	GEPIEn	SOFDEn	INCEn	0	DBMn
rw	r	r	rw	rw	rw	r	rw

Bit	Function
STALLn	**Endpoint Stall** Bit STALL can be set to indicate that the endpoint is stalled. If the stall bit for endpoint 0 (STALL0) is set, the next incoming setup token will automatically clear it. If STALL=0, the endpoint n is active If STALL=1, the endpoint n is stalled
GEPIEn	**Global Endpoint Interrupt Enable** Bit GEPIEn enables or disables the generation of the global endpoint interrupt n based on the endpoint specific interrupt request bits in register EPIRn. If GEPIE=0, the USB endpoint n interrupt is disabled If GEPIE=1, the USB endpoint n interrupt is enabled
SOFDEn	**Start of Frame Done Enable** If bit SOFDE is set, the current CPU buffer in USB memory is automatically tagged full (data flow from the CPU to USB) or empty (data flow from USB to the CPU) on each detection of a start of frame on the USB (auto-done). If SOFDE=0, no action takes place on SOF If SOFDE=1, automatic generation of DONE on SOF is enabled

INCEn	**Auto Increment Enable** If bit INCE is set, the address offset register ADROFF for CPU access to USB memory is automatically incremented after each data write or data read action of the USBVAL register. This allows the user to handle the USB memory like a FIFO without modification of the address of the desired memory location by software.
DBMn	**Dual Buffer Mode** Bit DBM allows the selection between single buffer mode and dual buffer mode. If DBM=0, single buffer mode is selected If DBM=1, dual buffer mode is selected

4.9.13 EPBSn - Endpoint Buffer Status Register

The bits of the endpoint buffer status registers indicate the status of the endpoint specific USB memory buffers and allows setting of certain USB memory buffer conditions.

Reset value: 20_H Address: $C2_H$

7	6	5	4	3	2	1	0
UBFn	CBFn	DIRn	ESP0	SETRDn	SETWRn	CLREPn	DONEn
r	r	r	w	w	w	w	w

Bit	Function
UBFn	**USB Buffer Full** Bit UBFn indicates the status of the USB memory buffer for endpoint n. USB read access: If UBFn=0, the USB buffer for endpoint n is empty. If UBFn=1, the USB buffer for endpoint n is not empty. USB write access: If UBFn=0, the USB buffer for endpoint n is not full. If UBFn=1, the USB buffer for endpoint n is full.

CBFn	**CPU Buffer Full** Bit CBFn indicates the status of the CPU memory buffer for endpoint n. CPU read access: If CBFn=0, the CPU buffer for endpoint n is empty. If CBFn=1, the CPU buffer for endpoint n is not empty. CPU write access: If CBFn=0, the CPU buffer for endpoint n is not full. If CBFn=1, the CPU buffer for endpoint n is full.
DIRn	**Direction of USB Memory Access** Bit DIRn indicates the direction of the last USB memory access for endpoint n. If DIRn=0, the last data flow for endpoint n was from host to CPU If DIRn=1, the last data flow for endpoint n was from CPU to host
ESP0	**Enable Status Phase** If bit ESP0 is set, the next status phase of endpoint n will automatically be acknowledged by an ACK except if the endpoint n is stalled. If the status phase is successfully completed, bit ESP0 is automatically reset by hardware and no status interrupt request (DIRR.STI) is generated. If the CPU detects a corrupted control transfer (endpoint 0), bit STALL0 should be set by software instead of bit ESP0 in order to indicate an error condition from which the USB device can not recover by itself. Note: This bit exists for endpoint 0 only.
SETRDn	**Set Direction of USB Memory Buffer to Read** Bit SETRDn is used to predict the direction of the next USB access for endpoint n as a USB read access. A faulty prediction causes no errors since the USB module determines the real direction. A change in the data direction is only executed if both USB memory buffers are empty. SETRDn can not be set together with CLREPn because a change of bit DIRn during a transfer is not allowed. Note: bits SETRDn and SETWRn must not be set at the same time.
SETWRn	**Set Direction of USB Memory Buffer to Write** Bit SETWRn is used to predict the direction of the next USB access for endpoint n as a USB write access. A faulty prediction causes no errors since the USB module determines the real direction. A change in the data direction is only executed if both USB memory buffers are empty. SETWR can not be set together with CLREPn because a change of EPBSn.DIRn during a transfer is not allowed. Note: bits SETWRn and SETRDn must not be set at the same time.

CLREPn	**Clear Endpoint** Setting bit CLREPn will set the address offset register for a CPU access to USB memory to 0. The bits CBFn and UBFn will be reset when CLREPn is set. Bit CLREPn is reset by hardware. A read operation of this bit will always deliver 0. Setting of bit CLREPn does not change the direction of endpoint n. This means, bit DIRn is not changed. Note: When bits CLREPn and ESP0 are set simultaneously with one instruction, bit ESP0 remains set and the next status phase is enabled. If only CLREPn is set, bit ESP0 is reset and the status phase is disabled. Setting bits CLREPn and SETRDn or SETWRn simultaneously with one instruction is not allowed. This means that the setting of SETRDn or SETWRn is ignored.
DONEn	**Buffer Done by CPU** If bit DONE is set, the current USB memory buffer assigned to CPU is automatically tagged full (data flow from the CPU to USB) or empty (data flow from USB to the CPU). This bit is reset by hardware after it has been set. A read operation of this bit always delivers a 0. Note: If the direction of the endpoint is read (USB read access) and auto-increment is enabled (INCEn=1) and DONEn is set, the content of register ADROFF is copied automatically to register EPLENn of the actual endpoint. Register EPLENn is not changed if the auto-increment capability is disabled (INCEn=0).

4.9.14 EPBAn - Endpoint Base Address Register

The endpoint base address and length registers define the location and size (start address and length) of the endpoint specific buffers in the USB memory (also see **Chapter 4.9.15**).

Reset value: 00_H Address: $C5_H$

7	6	5	4	3	2	1	0
PAGEn	0	0	0	An6	An5	An4	An3
r	r	r	r	rw	rw	rw	rw

Bit	Function
PAGEn	**Buffer Page for endpoint n (single buffer mode only)** In single buffer mode, the endpoint n can be either located on USB memory buffer page 0 (PAGEn=0) or on USB memory buffer page 1 (PAGEn=1) by clearing or setting this bit. In dual buffer mode this bit has no effect. Note: The SETUP token is always stored on USB memory buffer page 0 at address 00_H to 07_H.
An6-An3	**Endpoint n Buffer Start Address** The bits 0 to 3 of EPBAn are the address bits A6 to A3 of the USB memory buffer start address for endpoint n. A7 and A2-A0 of the resulting USB memory buffer start address are set to 0.

4.9.15 EPLENn - Endpoint Buffer Length Register

Reset value: $0XXXXXXX_B$ Address: $C6_H$

7	6	5	4	3	2	1	0
0	Ln6	Ln5	Ln4	Ln3	Ln2	Ln1	Ln0
r	rw	rw	rw	rw	rw	rw	rw

Bit	Function
Ln6 - Ln0	**Endpoint n Buffer Length** The bits 0 to 6 of EPLENn define the length of the USB memory buffer for endpoint n and can not be written if DINIT=1.

4.9.16 EGSR - Endpoint Get_Status Register

These two registers hold a 16-bit value that is sent to the USB host upon an "Endpoint Get Status" command.

Reset value: 00_B Address: CA_H

7	6	5	4	3	2	1	0
EST15	EST14	EST13	EST12	EST11	EST10	EST9	EST8
rw	rw	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: $C9_H$

7	6	5	4	3	2	1	0
EST7	EST6	EST5	EST4	EST3	EST2	EST1	STALL
rw	rw	rw	rw	rw	rw	rw	r

Bit	Function
STALL	**Endpoint Stalled (Set/Cleared by the USB host)** The endpoint stall status can be read by the µC in the GESR register and the status is returned with every Get_Status request for Endpoint. 0: endpoint is not stalled 1: endpoint is stalled This bit cannot be written by the µC.

The function of this register is completely determined by the USB specification. For further functional additions the higher bits of this register (EST15-1) can be programmed to "1" to fulfill future requirements.

5 ISDN Module

5.1 General Functions and Architecture

Figure 34 shows the architecture of the ISDN block containing the following functions:

- S/T-interface transceiver operating in terminal mode (TE)
- Serial or parallel microcontroller interface
- Two B-channel HDLC-controller with 128 byte FIFOs per channel and per direction with programmable FIFO block size (threshold)
- One D-channel HDLC-controller with 64 byte FIFOs per direction with programmable FIFO block size (threshold)
- IOM-2 interface for terminal applications (TE mode)
- D-channel access mechanism
- C/I- and Monitor channel handler
- Auxiliary interface with interrupt and general purpose I/O lines and LED drivers
- Clock and timing generation
- Digital PLL to synchronize the transceiver to the S/T interface
- Reset generation (watchdog timer)
- SPI interface for connection of a serial EEPROM

The functional blocks are described in the following chapters.

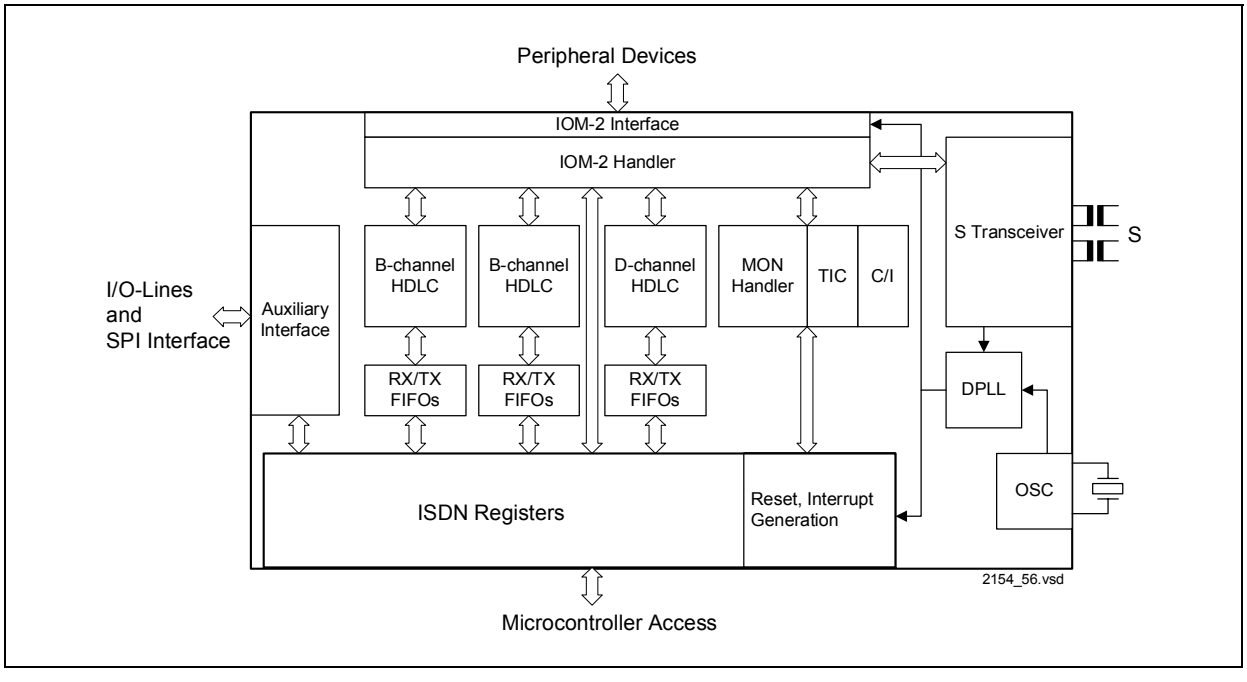


Figure 34 Functional Block Diagram of the ISDN Module

Note: All addresses mentioned in the ISDN chapter must be prefixed by $F8_H$ to correspond to the ISDN address space $F800_H$ - $F8FF_H$ (see **Chapter 5.8**).

5.1.1 Timer 2 and 3

The SIUC-BA provides 4 timers (Timer 0, 1, 2 and 3). Timer 0 and timer 1 are located in the microcontroller module (see **Chapter 3.3**), timer 2 and 3 are embedded in the ISDN module and described below.

Each of both timers provide two modes (**Table 16**), a count down timer interrupt, i.e. an interrupt is generated only once after expiration of the selected period, and a periodic timer interrupt, which means an interrupt is generated continuously after every expiration of that period.

Table 16 SIUC-BA Timers

Address	Register	Modes	Period
24_H	TIMR2	Periodic	64 ... 2048 ms
		Count Down	64 ms ... 14.336 s
65_H	TIMR3	Periodic	1 ... 63 ms
		Count Down	1 ... 63 ms

When the programmed period has expired an interrupt is generated and indicated in the auxiliary interrupt status ISTA.AUX (enabled/disabled via IEN2.AUX). The source of the interrupt can be read from AUXI (TIN2, TIN3) and each of the interrupt sources can individually be masked in AUXM.

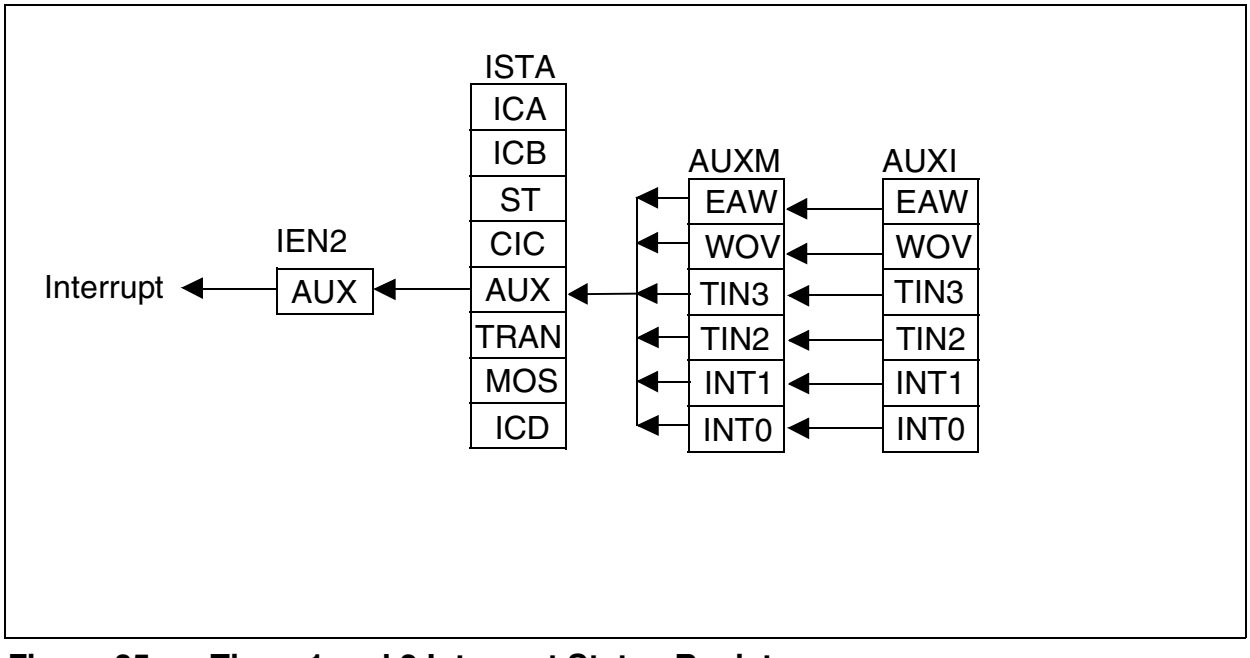


Figure 35 Timer 1 and 2 Interrupt Status Registers

Timer 2

The host controls the timer 2 by setting bit CMDRD.STI to start the timer and by writing register TIMR2 to stop the timer. After time period T1 an interrupt (AUXI.TIN2) is generated continuously if CNT=7 or a single interrupt is generated after timer period T if CNT<7 (**Figure 36**).

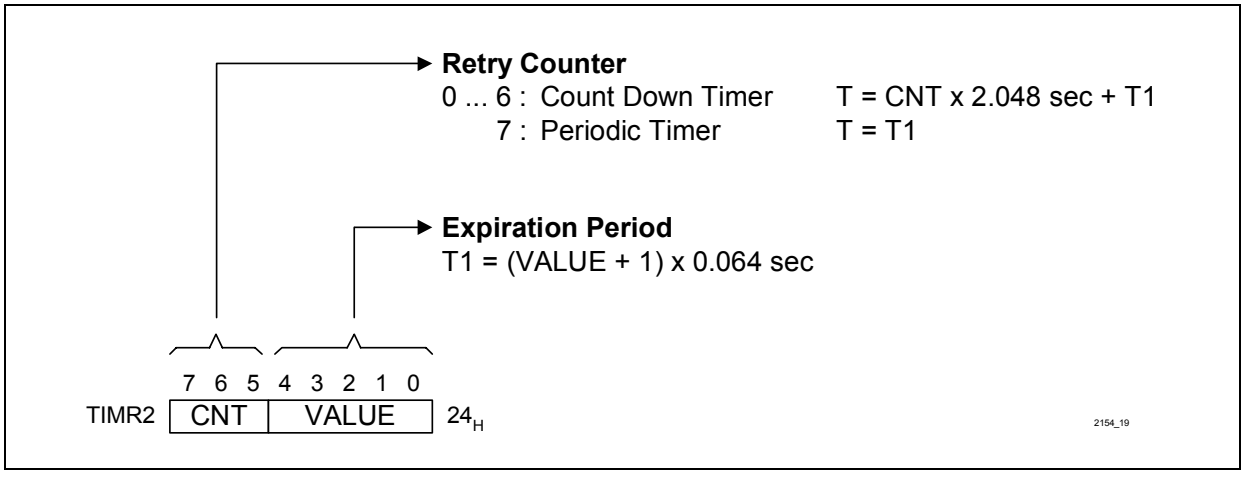


Figure 36 Timer 2 Register

Timer 3

The host starts and stops timer 3 in TIMR3.CNT (**Figure 37**). If TIMR3.TMD=0 the timer is operating in count down mode, for TIMR3.TMD=1 a periodic interrupt AUXI.TIN3 is generated. The timer length (for count down timer) or the timer period (for periodic timer), respectively, can be configured to a value between 1 - 63 ms (TIMR3.CNT).

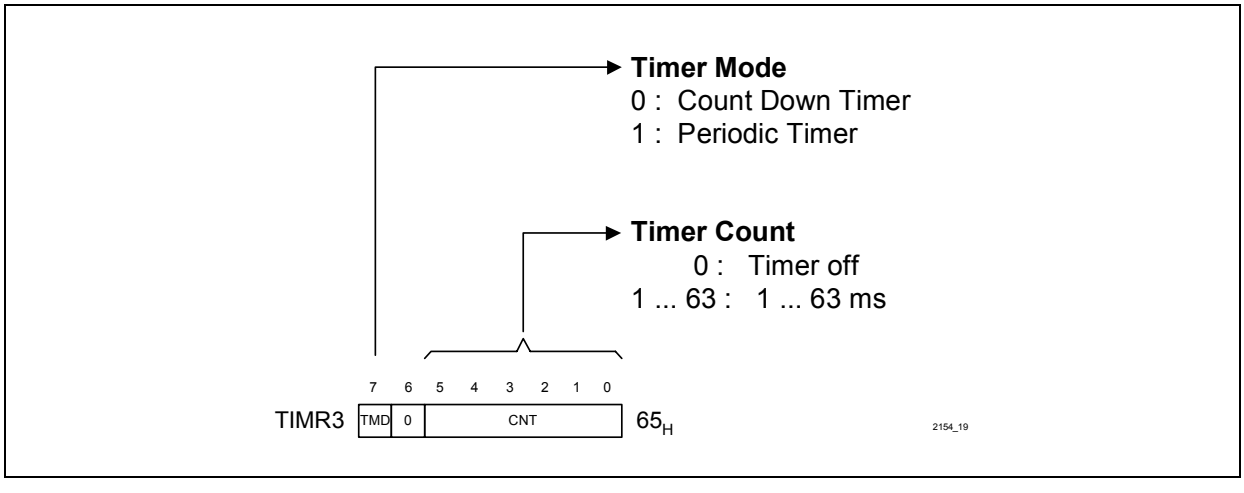


Figure 37 Timer 3 Register

Further timers are available in the microcontroller module (see **Chapter 3.3**).

5.1.2 Activation Indication via Pin ACL

The activated state of the S-interface is directly indicated via pin $\overline{\text{ACL}}$ (Activation LED). An LED with pre-resistance may directly be connected to this pin and a low level is driven on $\overline{\text{ACL}}$ as soon as the layer 1 state machine reaches the activated state (see **Figure 38**).

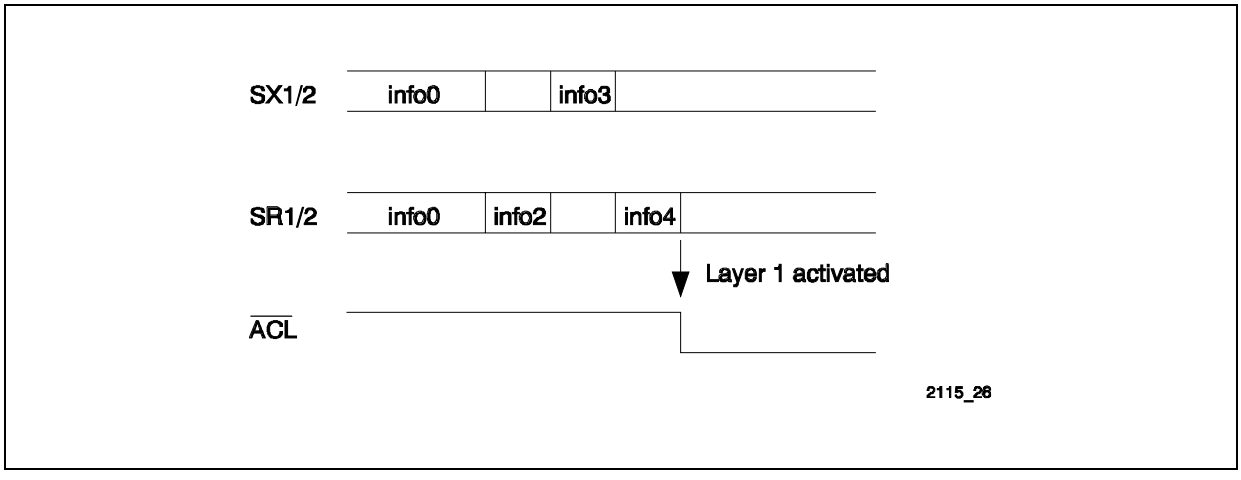


Figure 38 ACL Indication of Activated Layer 1

By default (ACFG2.ACL=0) the state of layer 1 is indicated at pin $\overline{\text{ACL}}$. If the automatic indication of the activated layer 1 is not required, the state on pin $\overline{\text{ACL}}$ can also be controlled by the host (see **Figure 39**).
If ACFG2.ACL=1 the LED on pin $\overline{\text{ACL}}$ can be switched on (ACFG2.LED=1) and off (ACFG2.LED=0) by the host.

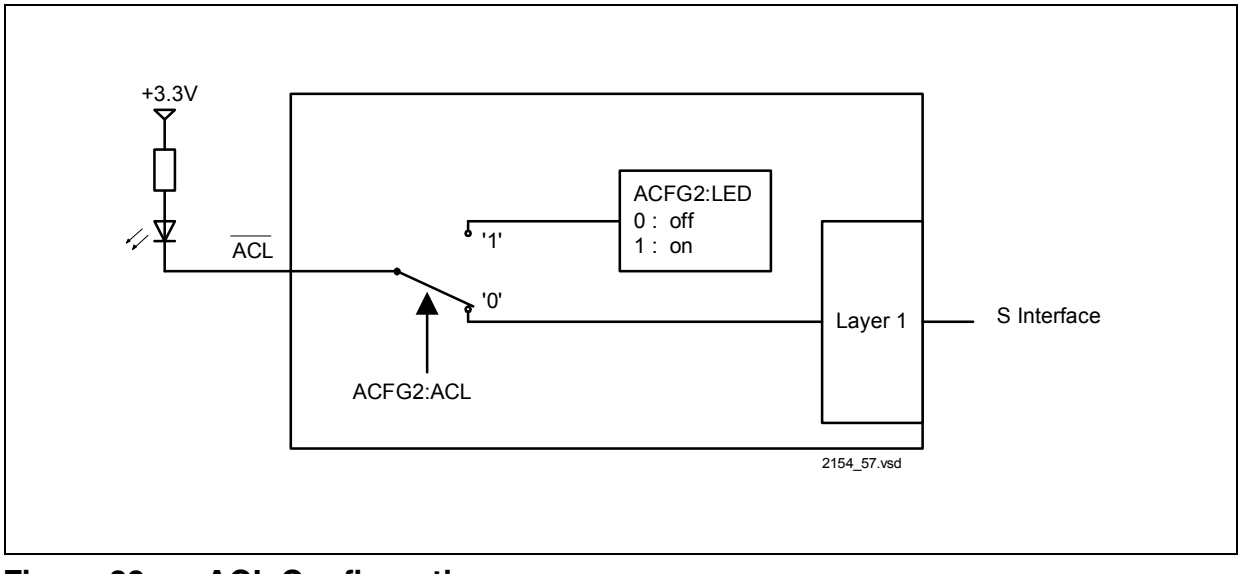


Figure 39 ACL Configuration

5.2 S/T-Interface

The layer-1 functions for the S/T interface of the SIUC-BA are:

- line transceiver functions for the S/T interface according to the electrical specifications of ITU-T I.430;
- conversion of the frame structure between IOM-2 and S/T interface;
- conversion from/to binary to/from pseudo-ternary code;
- level detection
- receive timing recovery for point-to-point, passive bus and extended passive bus configuration
- S/T timing generation using IOM-2 timing synchronous to system, or vice versa;
- D-channel access control and priority handling;
- D-channel echo bit generation by handling of the global echo bit;
- activation/deactivation procedures, triggered by primitives received over the IOM-2 C/I channel or by INFO's received from the line;
- execution of test loops.

The wiring configurations in user premises, in which the SIUC-BA can be used, are illustrated in **Figure 40**.

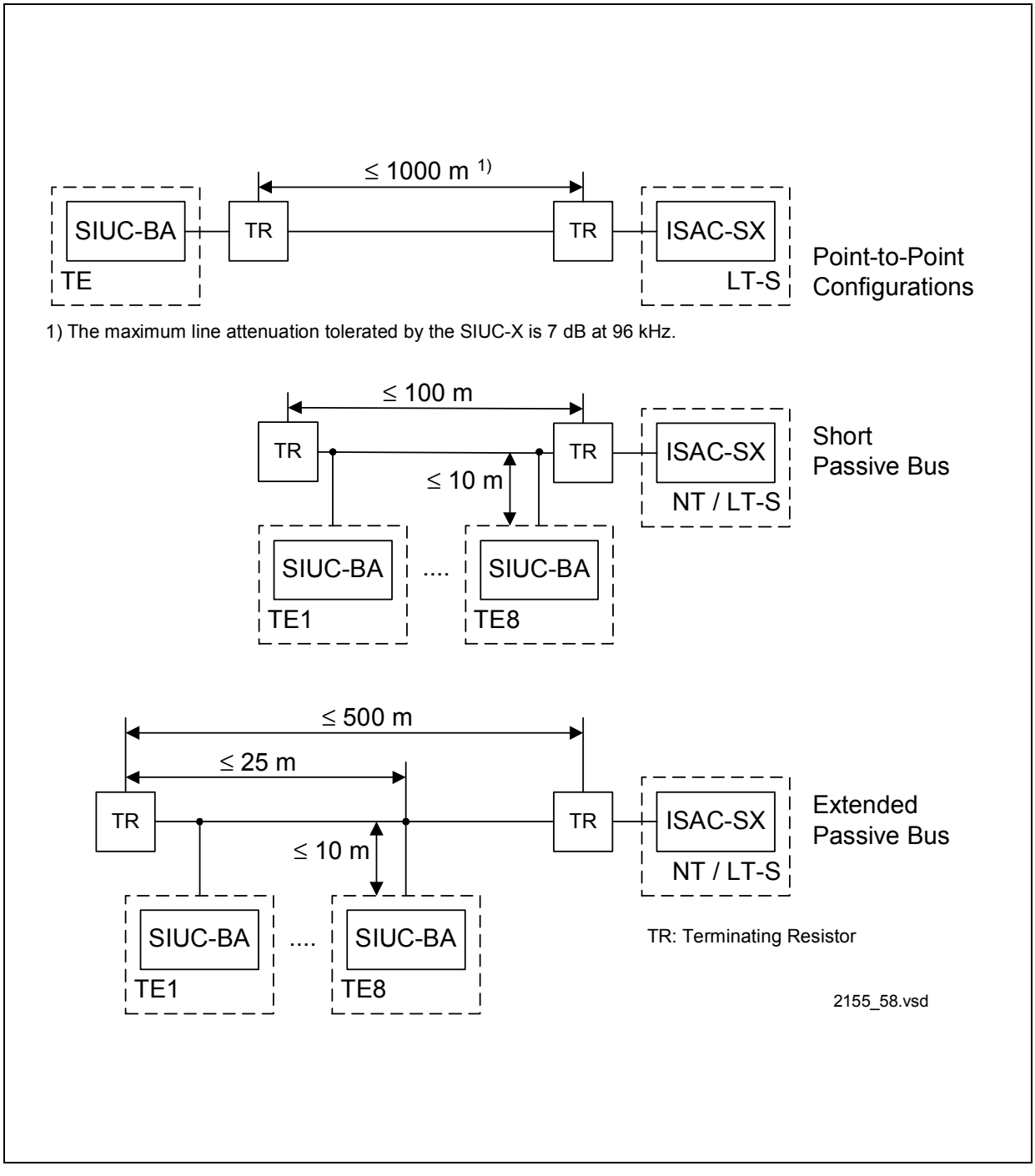


Figure 40 Wiring Configurations in User Premises

5.2.1 S/T-Interface Coding

Transmission over the S/T-interface is performed at a rate of 192 kbit/s. 144 kbit/s are used for user data (B1+B2+D), 48 kbit/s are used for framing and maintenance information.

Line Coding

The following figure illustrates the line code. A binary ONE is represented by no line signal. Binary ZEROs are coded with alternating positive and negative pulses with two exceptions:
For the required frame structure a code violation is indicated by two consecutive pulses of the same polarity. These two pulses can be adjacent or separated by binary ONEs.
In bus configurations a binary ZERO always overwrites a binary ONE.

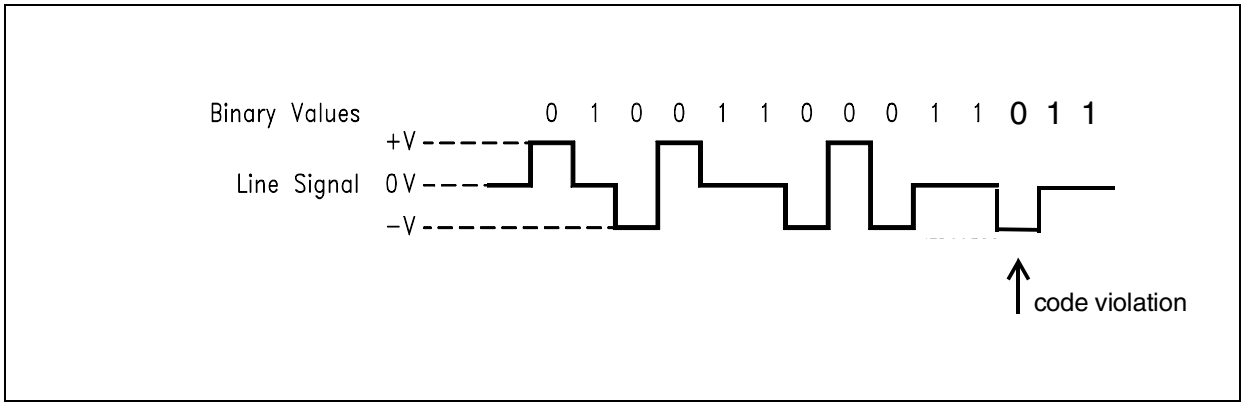


Figure 41 S/T -Interface Line Code

Frame Structure

Each S/T frame consists of 48 bits at a nominal bit rate of 192 kbit/s. For user data (B1+B2+D) the frame structure applies to a data rate of 144 kbit/s (see **Figure 42**).
In the direction TE → NT the frame is transmitted with a two bit offset. For details on the framing rules please refer to ITU I.430 section 6.3. The following figure illustrates the standard frame structure for both directions (NT → TE and TE → NT) with all framing and maintenance bits.

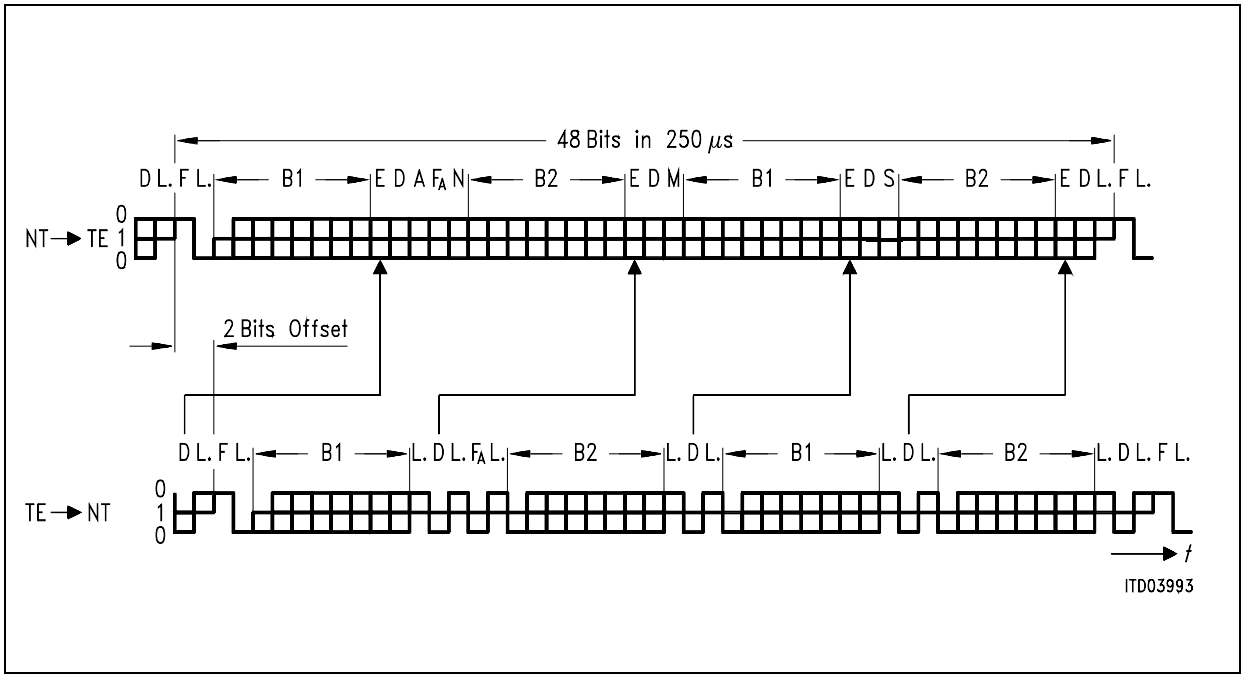


Figure 42 Frame Structure at Reference Points S and T (ITU I.430)

– F	Framing Bit	F = (0b) → identifies new frame (always positive pulse, always code violation)
– L.	D.C. Balancing Bit	L. = (0b) → number of binary ZEROs sent after the last L. bit was odd
– D	D-Channel Data Bit	Signaling data specified by user
– E	D-Channel Echo Bit	E = D → received E-bit is equal to transmitted D-bit
– F_A	Auxiliary Framing Bit	See section 6.3 in ITU I.430
– N		N = $\overline{F_A}$
– B1	B1-Channel Data Bit	User data
– B2	B2-Channel Data Bit	User data
– A	Activation Bit	A = (0b) → INFO 2 transmitted A = (1b) → INFO 4 transmitted
– S	S-Channel Data Bit	S_1 channel data (see note below)
– M	Multiframing Bit	M = (1b) → Start of new multiframe

Note: The ITU I.430 standard specifies S1 - S5 for optional use.

5.2.2 S/T-Interface Multiframing

According to ITU recommendation I.430 a multiframe provides extra layer 1 capacity in the TE-to-NT direction by using an extra channel between the TE and NT (Q-channel). The Q bits are defined to be the bits in the F_A bit position.

In the NT-to-TE direction the S-channel bits are used for information transmission. One S channel (S1) out of five possible S-channels can be accessed by the SIUC-BA.

The S and Q channels are accessed via the µC interface or the IOM-2 MONITOR channel, respectively, by reading/writing the SQR or SQX bits in the S/Q channel registers (SQRRx, SQXRx).

Table 17 shows the S and Q bit positions within the multiframe.

Table 17 S/Q-Bit Position Identification and Multiframe Structure

Frame Number	NT-to-TE F_A Bit Position	NT-to-TE M Bit	NT-to-TE S Bit	TE-to-NT F_A Bit Position
1	ONE	ONE	S11	Q1
2	ZERO	ZERO	S21	ZERO
3	ZERO	ZERO	S31	ZERO
4	ZERO	ZERO	S41	ZERO
5	ZERO	ZERO	S51	ZERO
6	ONE	ZERO	S12	Q2
7	ZERO	ZERO	S22	ZERO
8	ZERO	ZERO	S32	ZERO
9	ZERO	ZERO	S42	ZERO
10	ZERO	ZERO	S52	ZERO
11	ONE	ZERO	S13	Q3
12	ZERO	ZERO	S23	ZERO
13	ZERO	ZERO	S33	ZERO
14	ZERO	ZERO	S43	ZERO
15	ZERO	ZERO	S53	ZERO
16	ONE	ZERO	S14	Q4
17	ZERO	ZERO	S24	ZERO
18	ZERO	ZERO	S34	ZERO
19	ZERO	ZERO	S44	ZERO
20	ZERO	ZERO	S54	ZERO
1	ONE	ONE	S11	Q1
2	ZERO	ZERO	S21	ZERO

After multiframe synchronization has been established, the Q data will be inserted at the upstream (TE → NT) F_A bit position in each 5th S/T frame (see **Table 17**).

When synchronization is not achieved or lost, each received F_A bit is mirrored to the next transmitted F_A bit.
Multiframe synchronization is achieved after two complete multiframes have been detected with reference to F_A/N bit and M bit positions. Multiframe synchronization is lost if bit errors in F_A/N bit or M bit positions have been detected in two consecutive multiframes. The synchronization state is indicated by the MSYN bit in the S/Q-channel receive register (SQRR1).
The multiframe synchronization can be enabled or disabled by programming the MFEN bit in the S/Q-channel transmit register (SQXR1).

If enabled (TR_CONF1.EN_SFSC=1) the first frame within a multiframe generates a short FSC, i.e. every 40th IOM-frame a short FSC is generated.

Interrupt Handling for Multiframing

To trigger the microcontroller for a multiframe access an interrupt can be generated once per multiframe (SQW) or if the received S-channels have changed (SQC).
In both cases the microcontroller has access to the multiframe within the duration of one multiframe (5 ms).

5.2.3 Multiframe Synchronization (M-Bit)

The SIUC-BA offers the capability to control the start of the multiframe from external signals, so applications which require synchronization between different S-interfaces are possible. Such an application is the connection of DECT base stations to PBX line cards.

For this purpose a multiplexed function of the AUX4 pin is used. If the ACFG2.A4SEL is set to “1” the pin is not used as general pupose I/O pin but as M-Bit output.

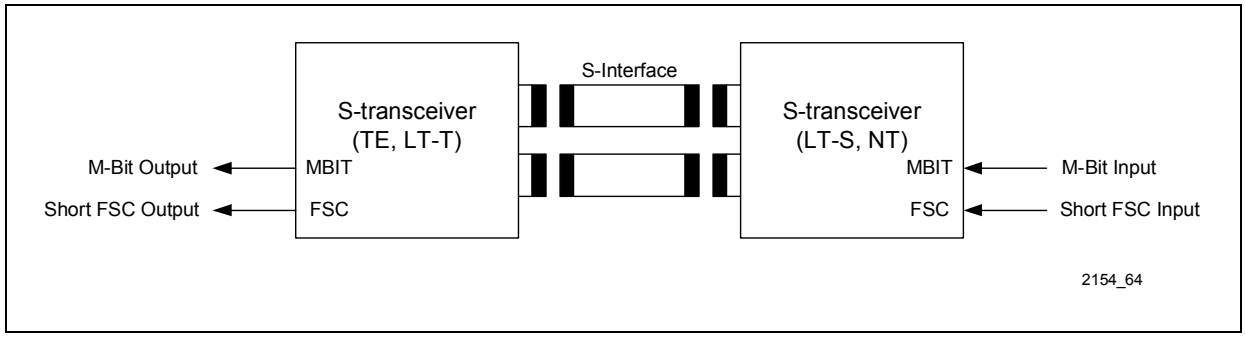


Figure 43 Multiframe Synchronization using the M-Bit

In TE mode the SIUC-BA outputs the value of the M-bit on the MBIT pin. The value of M should be sampled at the falling edge of FSC.

Frame Relationship

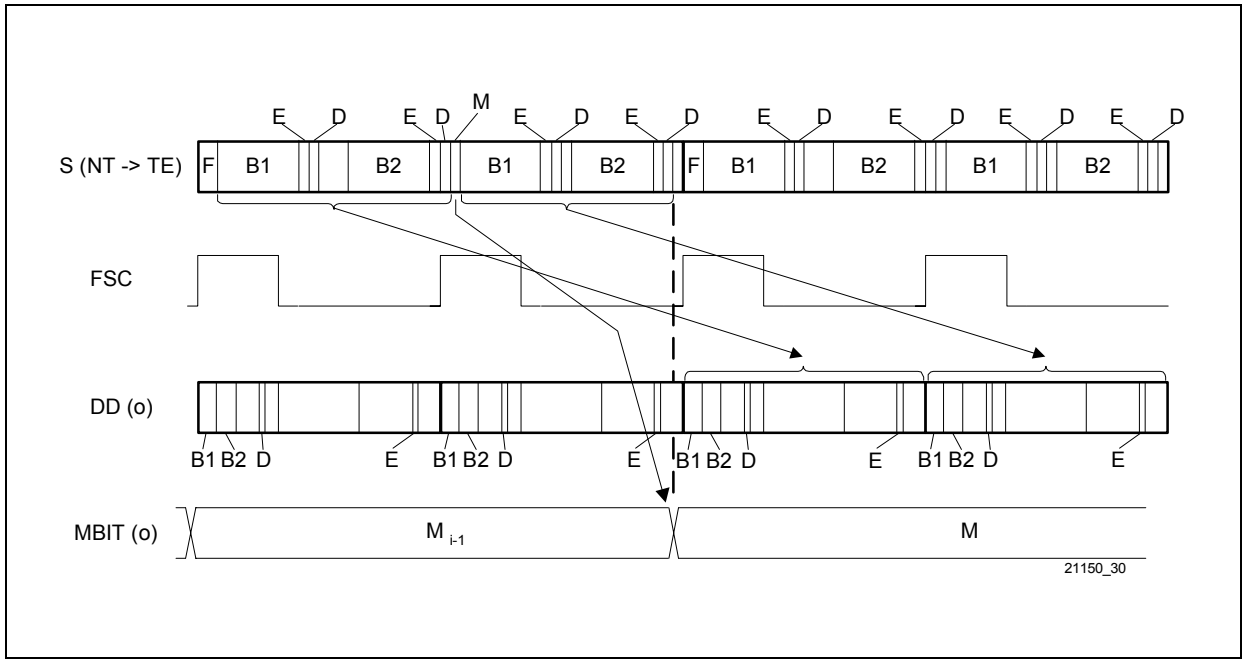


Figure 44 Frame Relationship in TE mode (M-Bit output)

5.2.4 Data Transfer and Delay between IOM-2 and S/T

In the state F7 (Activated) or if the internal layer-1 statemachine is disabled and XINF of register TR_CMD is programmed to ’011’ the B1, B2, D and E bits are transferred transparently from the S/T to the IOM-2 interface. In all other states ’1’s are transmitted to the IOM-2 interface.

To transfer data transparently to the S/T interface any activation request C/I command (AR8, AR10 or ARL) is additionally necessary or if the internal layer-1 statemachine is disabled, bit TDDIS of register TR_CMD has additionally to be programmed to ’0’.

Figure 45 shows the data delay between the IOM-2 and the S/T interface and vice versa.

For the D channel the delay from the IOM-2 to the S/T interface is only valid if S/G evaluation is disabled (MODED:DIM0=0). If S/G evaluation is enabled (MODED.DIM2-0=0x1$_B$) the delay depends on the selected priority and the relation between the echo bits on S and the D channel bits on the IOM-2, e.g. for priority 8 the timing relation between the 8th D-bit on S bus and the D-channel on IOM-2.

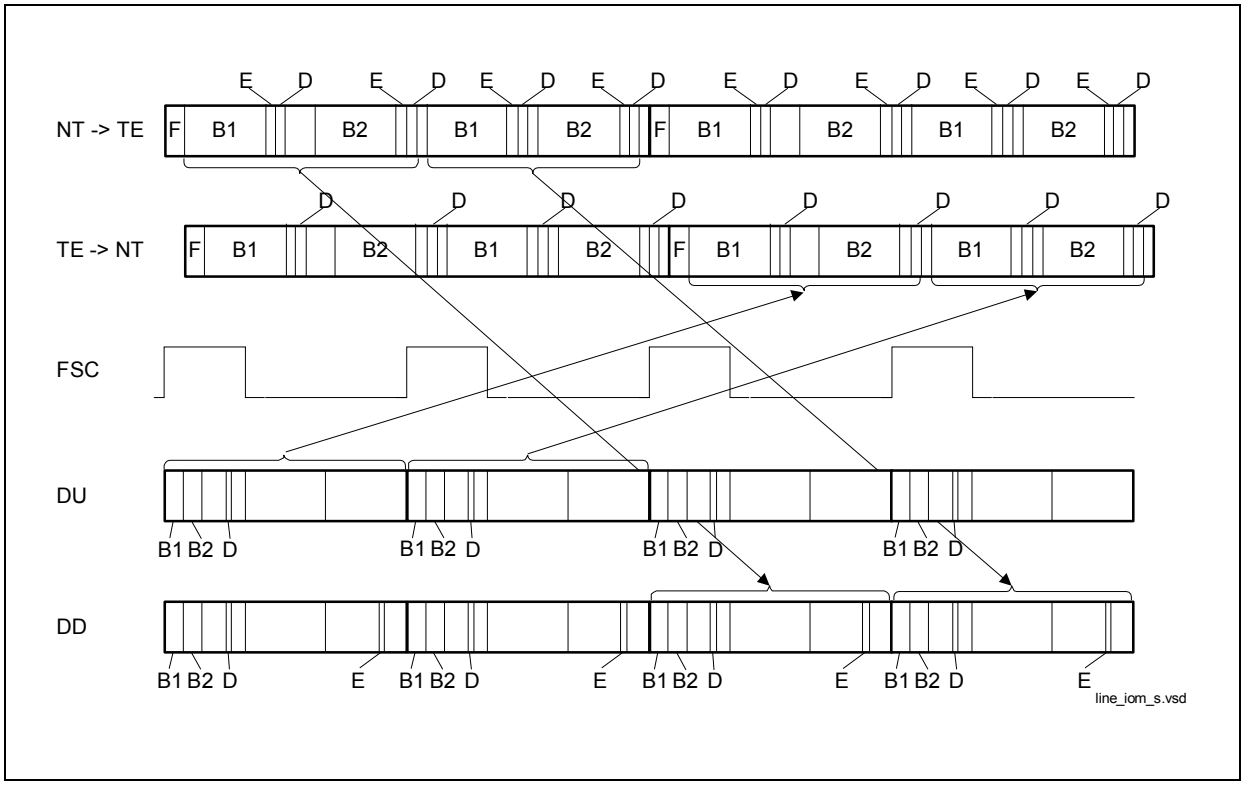


Figure 45 Data Delay between IOM-2 and S/T Interface (TE mode)

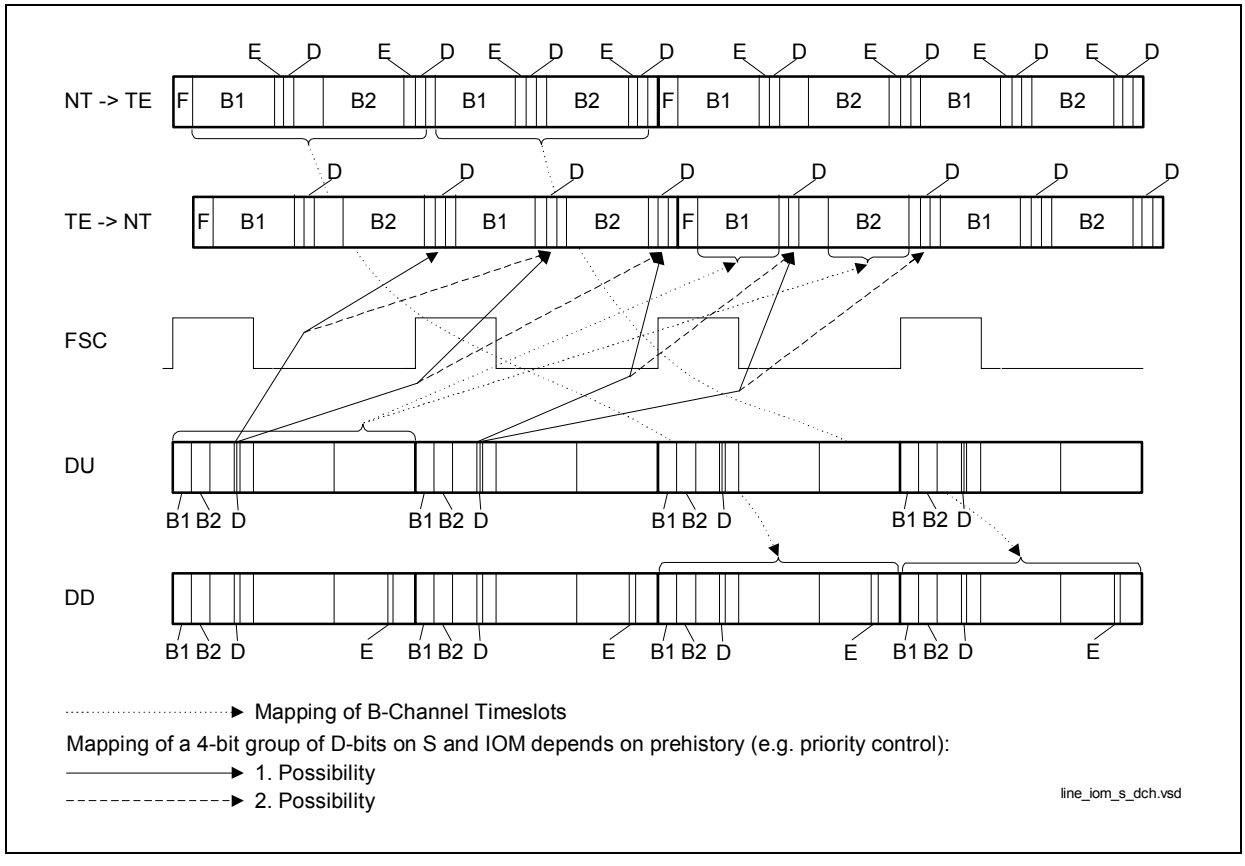


Figure 46 Data Delay between IOM-2 and S/T Interface with S/G Bit Evaluation (TE mode)

5.2.5 Transmitter Characteristics

The full-bauded pseudo-ternary pulse shaping is achieved with the integrated transmitter which is realized as a symmetrical current limited voltage source ($V_{SX1/SX2}$ = +/-1.05V; I_{max} = 26 mA). The equivalent circuit of the transmitter is shown in **Figure 47**.
The nominal pulse amplitude on the S-interface 750 mV (zero-peak) is adjusted with external resistors (see **Chapter 5.2.7.1**).

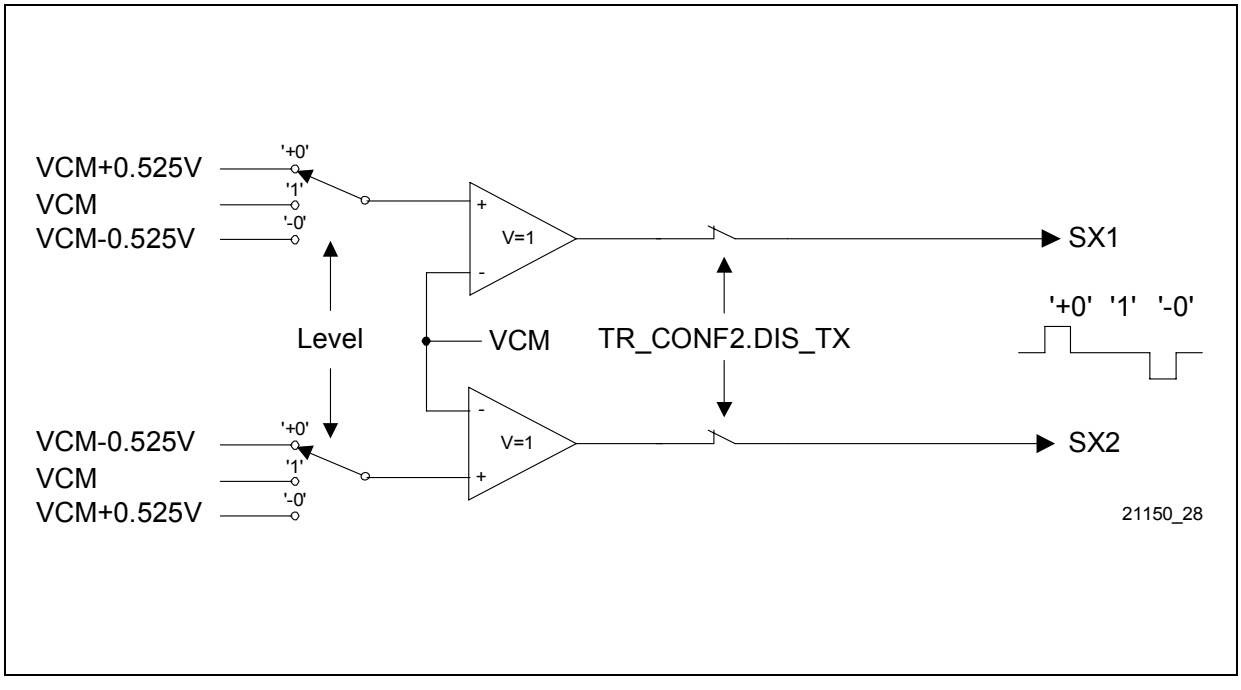


Figure 47 Equivalent Internal Circuit of the Transmitter Stage

5.2.6 Receiver Characteristics

The receiver consists of a differential input stage, a peak detector and a set of comparators. Additional noise immunity is achieved by digital oversampling after the comparators. A simplified equivalent circuit of the receiver is shown in **Figure 48**.

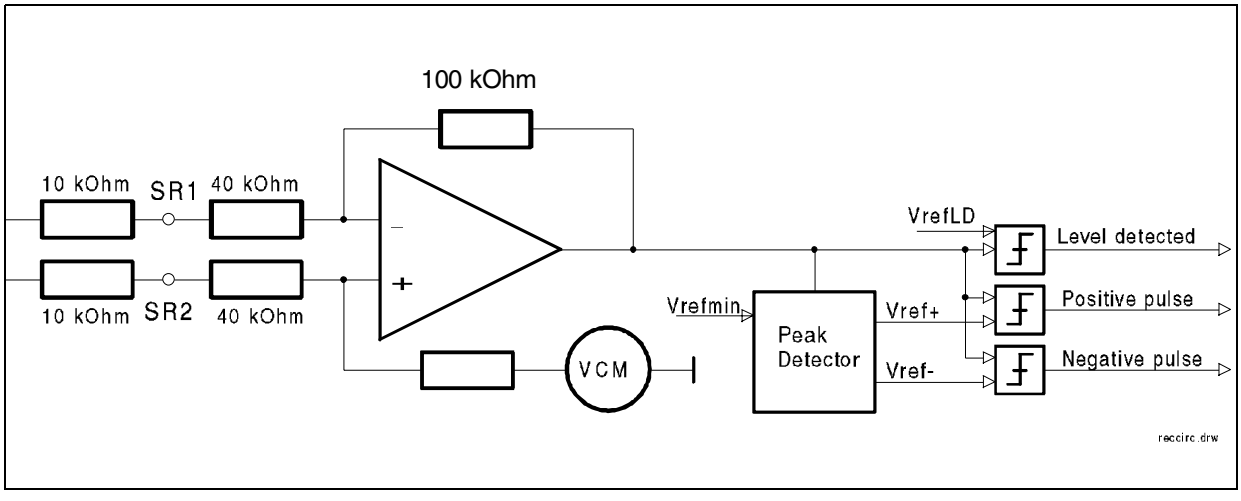


Figure 48 Equivalent Internal Circuit of the Receiver Stage

The input stage works together with external 10 kΩ resistors to match the input voltage to the internal thresholds. The data detection threshold Vref is continiously adapted between a maximal (Vrefmax) and a minimal (Vrefmin) reference level related to the line level. The peak detector requires maximum 2 µs to reach the peak value while storing the peak level for at least 250 µs (RC > 1 ms).

The additional level detector for power up/down control works with a fixed threshold VrefLD. The level detector monitors the line input signals to detect whether an INFO is present. When closing an analog loop it is therefore possible to indicate an incoming signal during activated loop.

5.2.7 S/T Interface Circuitry

For both, receive and transmit direction a 1:1 transformer is used to connect the SIUC-BA transceiver to the 4 wire S/T interface. Typical transformer characteristics can be found in the chapter on electrical characteristics. The connections of the line transformers is shown in **Figure 49**.

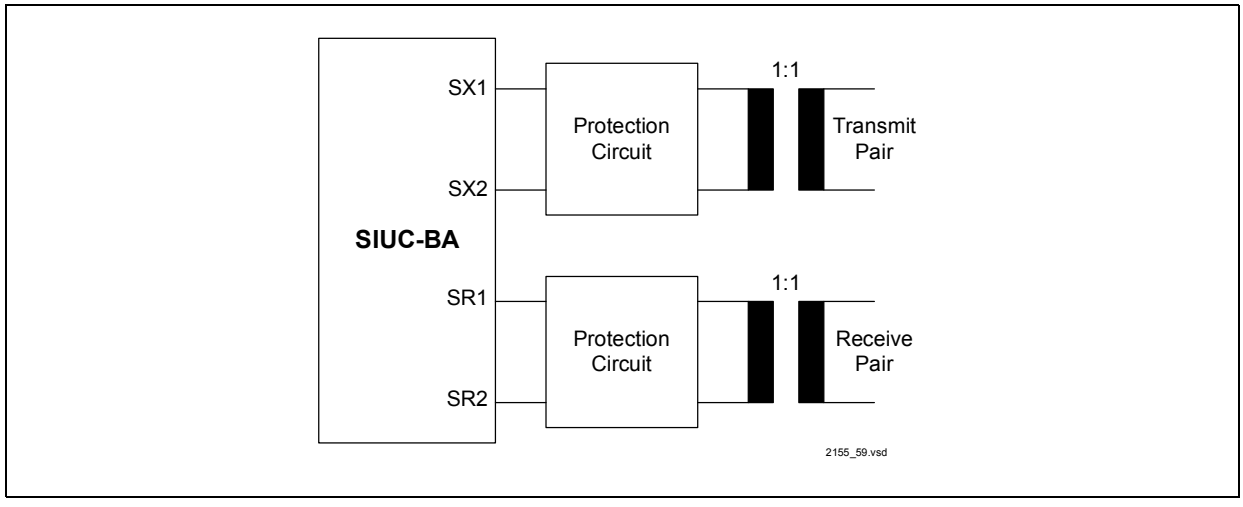


Figure 49 Connection of Line Transformers and Power Supply to the SIUC-BA

For the transmit direction an external transformer is required to provide isolation and pulse shape according to the ITU-T recommendations.

5.2.7.1 External Protection Circuitry

The ITU-T I.430 specification for both transmitter and receiver impedances in TEs results in a conflict with respect to external S-protection circuitry requirements:

– To avoid destruction or malfunction of the S-device it is desirable to drain off even small overvoltages reliably.
– To meet the 96 kHz impedance test specified for transmitters and receivers (for TEs only, ITU-T I.430 sections 8.5.1.2a and 8.6.1.1) the protection circuit must be dimensioned such that voltages below 1.2 V (ITU-T I.430 amplitude) x transformer ratio are not affected.

This requirement results from the fact that this test is also to be performed with no supply voltage being connected to the TE. Therefore the second reference point for overvoltages V_{DD}, is tied to GND. Then, if the amplitude of the 96 kHz test signal is greater than the combined forward voltages of the diodes, a current exceeding the specified one may pass the protection circuit.

The following recommendations aim at achieving the highest possible device protection against overvoltages while still fulfilling the 96 kHz impedance tests.

Protection Circuit for Transmitter

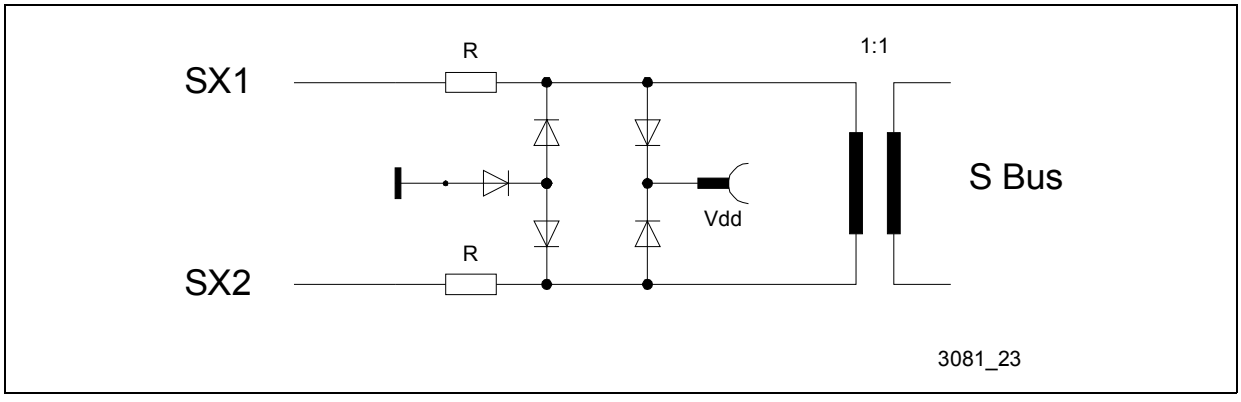


Figure 50 External Circuitry for Transmitter

Figure 50 illustrates the secondary protection circuit recommended for the transmitter.

The external resistors (R = 5 ... 10 Ω) are required in order to adjust the output voltage to the pulse mask on the one hand and in order to meet the output impedance of minimum 20 Ω (transmission of a binary zero according to ITU-T I.430) on the other hand.
Two mutually reversed diode paths protect the device against positive or negative overvoltages on both lines.

An ideal protection circuit should limit the voltage at the SX pins from – 0.4 V to V_{DD} + 0.4 V. With the circuit in **Figure 50** the pin voltage range is increased from – 1.4 V to V_{DD} + 0.7 V. The resulting forward voltage of 1.4 V will prevent the protection circuit from becoming active if the 96 kHz test signal is applied while no supply voltage is present.

Protection Circuit for Receiver

Figure 51 illustrates the external circuitry used in combination with a symmetrical receiver. Protection of symmetrical receivers is rather simple.

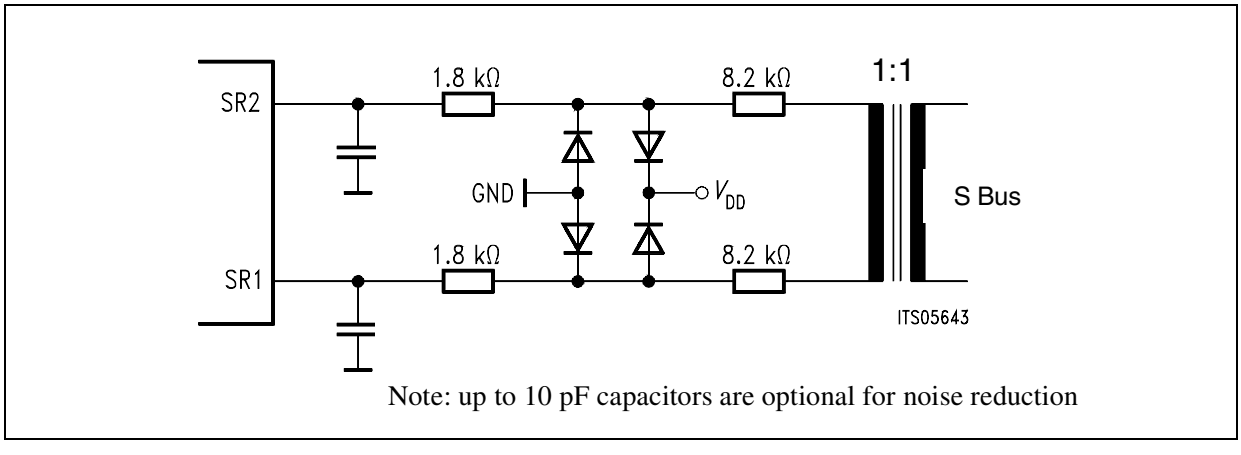


Figure 51 External Circuitry for Symmetrical Receivers

Between each receive line and the transformer a 10 kΩ resistor is used. This value is split into two resistors: one between transformer and protection diodes for current limiting during the 96 kHz test, and the second one between input pin and protection diodes to limit the maximum input current of the chip.

With symmetrical receivers no difficulties regarding LCL measurements are observed; compensation networks thus are obsolete.

In order to comply to the physical requirements of ITU-T recommendation I.430 and considering the national requirements concerning overvoltage protection and electromagnetic compatibility (EMC), the SIUC-BA may need additional circuitry.

5.2.8 S/T Interface Delay Compensation

The S/T transmitter is shifted by two S/T bits minus 7 oscillator periods (plus analog delay plus delay of the external circuitry) with respect to the received frame. To compensate additional delay introduced into the receive and transmit path by the external circuit the delay of the transmit data can be reduced by another two oscillator periods (2 x 130 ns). Therefore PDS of the TR_CONF2 register must be programmed to ’1’. This delay compensation might be necessary in order to comply with the "total phase deviation input to output" requirement of ITU-T recommendation I.430 which specifies a phase deviation in the range of – 7% to + 15% of a bit period.

5.2.9 Level Detection Power Down

If MODE1.CFS is set to ’0’, the clocks are also provided in power down state, whereas if CFS is set to ’1’ only the analog level detector is active in power down state. All clocks, including the IOM-2 interface, are stopped (DD, DU are ’high’, DCL and BCL are ’low’).

An activation initiated from the exchange side will have the consequence that a clock signal is provided automatically if TR_CONF0.LDD is set to ’0’. If TR_CONF0.LDD is set to ’1’ the microcontroller has to take care of an interrupt caused by the level detect circuit (ISTATR.LD)

From the terminal side an activation must be started by setting and resetting the SPU-bit in the IOM_CR register and writing TIM to the CIX0 register or by resetting MODE1.CFS=0.

5.2.10 Transceiver Enable/Disable

The layer-1 part of the SIUC-BA can be enabled/disabled by configuration (see **Figure 52**) with the two bits TR_CONF0.DIS_TR and TR_CONF2.DIS_TX .
By default all layer-1 functions with the exception of the transmitter buffer is enabled (DIS_TR = ’0’, DIS_TX = ’1’). With several terminals connected to the S/T interface, another terminal may keep the interface activated although the SIUC-BA does not establish a connection. The receiver will monitor for incoming calls in this configuration. If the transceiver is disabled (DIS_TR = ’1’) all layer-1 functions are disabled including the level detection circuit of the receiver. In this case the power consumption of the Layer-1 is reduced to a minimum. The HDLC controller and codec part can still operate via IOM-2. The DCL and FSC pins become input.

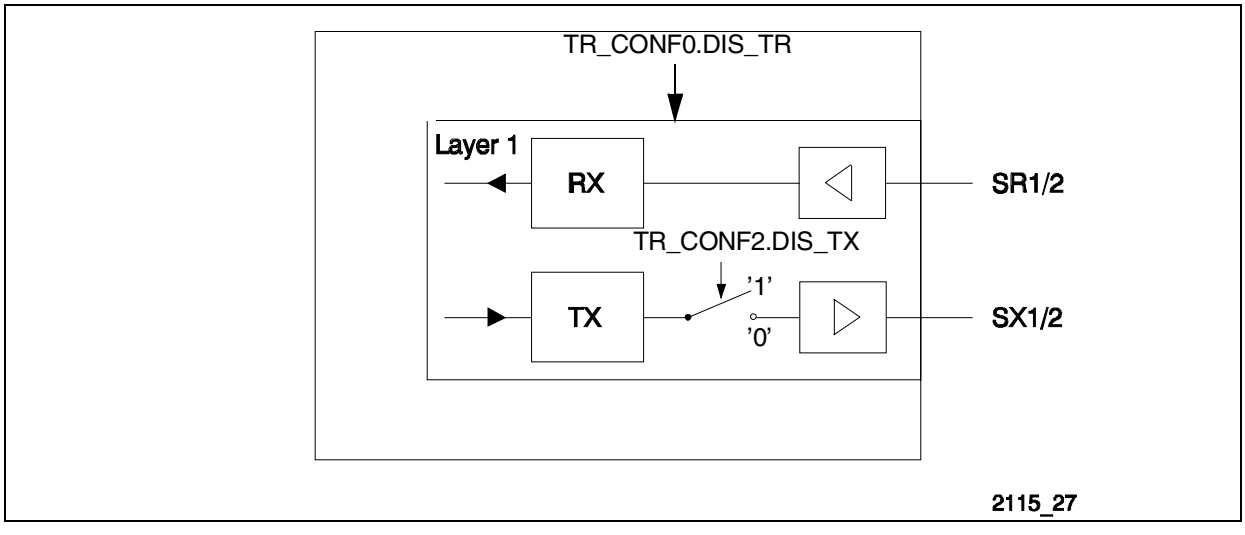


Figure 52 Disabling of S/T Transmitter

5.2.11 Test Functions

The SIUC-BA provides test and diagnostic functions for the S/T interface:

– The **internal local loop** (internal Loop A) is activated by a C/I command ARL or by setting the bit LP_A (Loop Analog) in the TR_CMD register if the layer-1 statemachine is disabled.
The transmit data of the transmitter is looped back internally to the receiver. The data of the IOM-2 input B- and D-channels are looped back to the output B- and D-channels.
The S/T interface level detector is enabled, i.e. if a level is detected this will be reported by the Resynchronization Indication (RSY) but the loop function is not affected.
Depending on the DIS_TX bit in the TR_CONF2 register the internal local loop can be transparent or non transparent to the S/T line.

– The **external local loop** (external Loop A) is activated in the same way as the internal local loop described above. Additionally the EXLP bit in the TR_CONF0 register has to be programmed and the loop has to be closed externally as described in **Figure 53**. The S/T interface level detector is disabled.
This allows complete system diagnostics.
– In **remote line loop** (RLP) received data is looped back to the S/T interface. The D-channel information received from the line card is transparently forwarded to the output IOM-2 D-channel. The output B-channel information on IOM-2 is fixed to 'FF'$_H$ while this test loop is active. The remote loop is programmable in TR_CONF2.RLP.

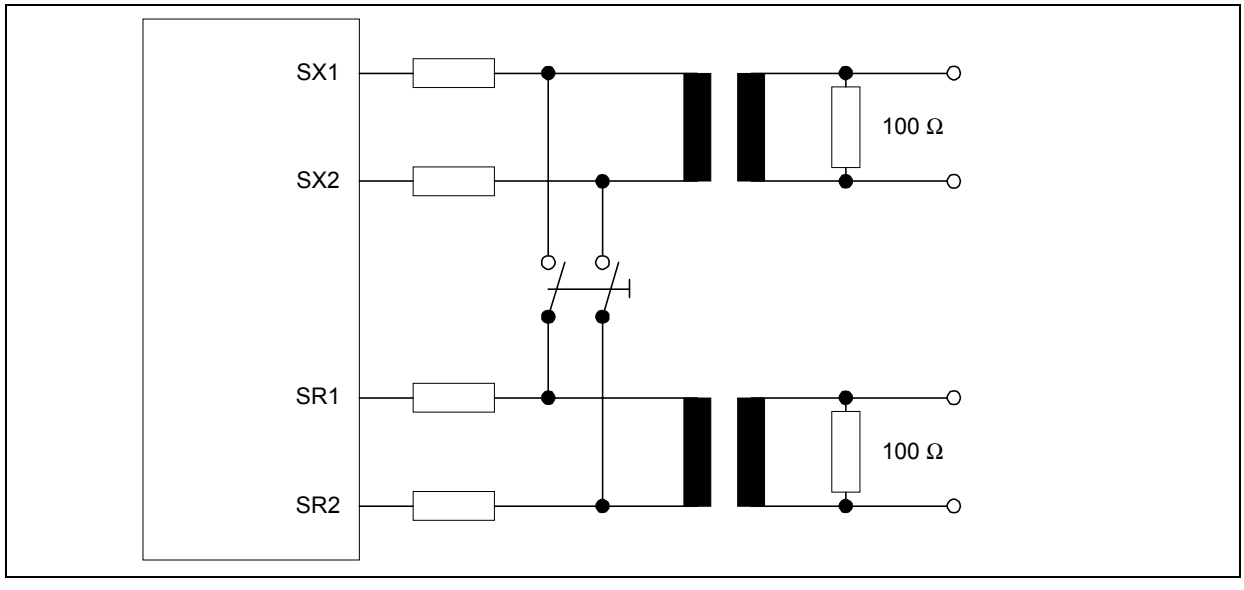


Figure 53 External Loop at the S/T-Interface

– transmission of special test signals on the S/T interface according to the modified AMI code are initiated via a C/I command written in CIX0 register (see **Chapter 5.3.2**)

Two kinds of test signals may be transmitted by the SIUC-X:

– The single pulses are of alternating polarity. One pulse is transmitted in each frame resulting in a frequency of the fundamental mode of 2 kHz. The corresponding C/I command is SSP (Send Single Pulses).
– The continuous pulses are of alternating polarity. 48 pulses are transmitted in each frame resulting in a frequency of the fundamental mode of 96 kHz. The corresponding C/I command is SCP (Send Continuous Pulses).

5.3 Control of Layer-1

The layer-1 activation/ deactivation can be controlled by an internal state machine via the IOM-2 C/I0 channel or by software via the microcontroller interface directly. In the default state the internal layer-1 state machine of the SIUC-BA is used. By setting the L1SW bit in the TR_CONF0 register the internal state machine can be disabled and the layer-1 commands, which are normally generated by the internal state machine are written directly in the TR_CMD register or indications read from the TR_STA register respectively. The SIUC-BA layer-1 control flow is shown in **Figure 54**.

It should be noted that the TR_CMD and TR_STA registers can always be read even if the layer-1 statemachine is enabled. This may be usefull for test purposes.

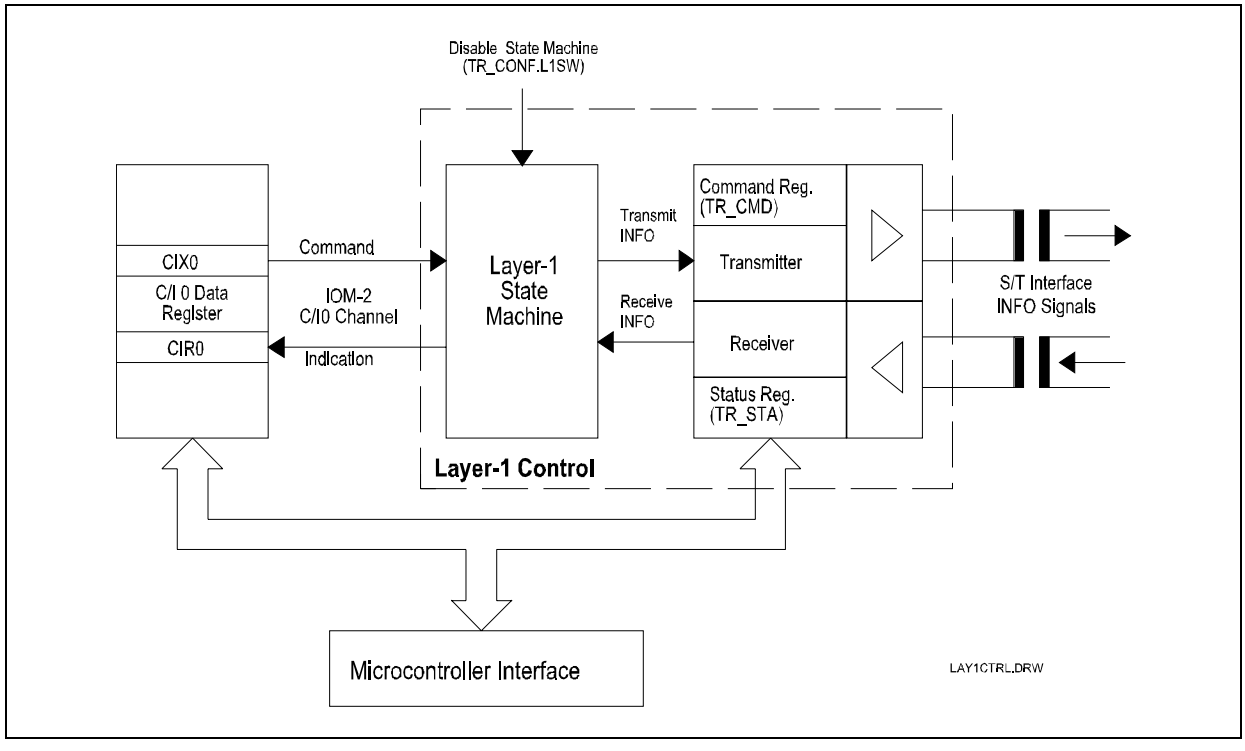


Figure 54 Layer-1 Control

In the following sections the layer-1 control by the SIUC-BA state machine will be described. For the description of the IOM-2 C/I0 channel see also **Chapter 5.5.5**.

The layer-1 functions are controlled by commands issued via the CIX0 register. These commands, sent over the IOM-2 C/I channel 0 to layer 1, trigger certain procedures, such as activation/deactivation, switching of test loops and transmission of special pulse patterns. These procedures are governed by layer-1 state diagrams. Responses from layer 1 are obtained by reading the CIR0 register after a CIC interrupt (ISTA).

The state diagrams of the SIUC-BA are shown in **Figure 56** and **Figure 57**. The activation/deactivation implemented by the SIUC-BA agrees with the requirements set forth in ITU recommendations. State identifiers F1-F8 are in accordance with ITU I.430.

State machines are the key to understanding the transceiver part of the SIUC-BA. They include all information relevant to the user and enable him to understand and predict the behaviour of the SIUC-BA. The state diagram notation is given in **Figure 55**. The informations contained in the state diagrams are:

– state name (based on ITU I.430)
– S/T signal received (INFO)
– S/T signal transmitted (INFO)
– C/I code received
– C/I code transmitted
– transition criteria

The coding of the C/I commands and indications are described in detail in **Chapter 5.3.2**.

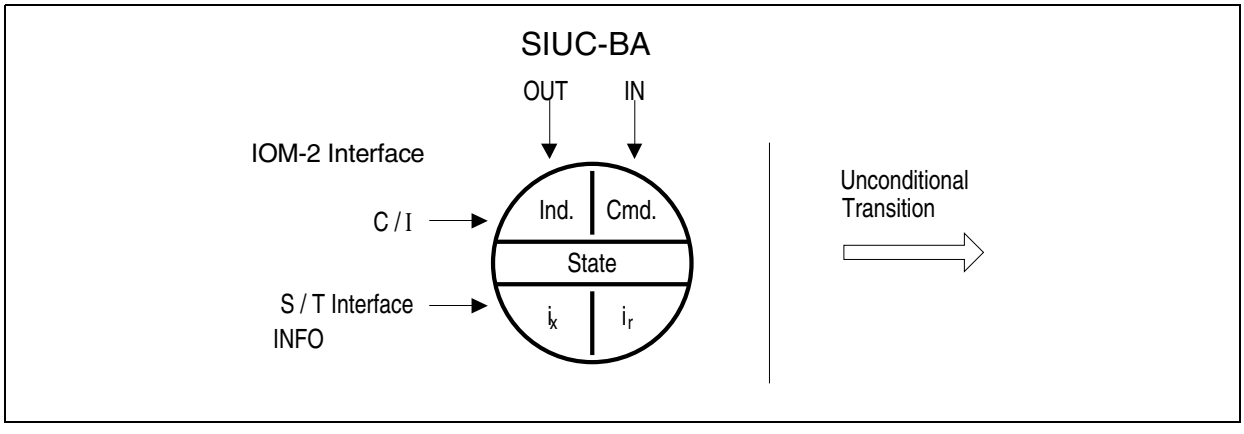


Figure 55 State Diagram Notation

The following example illustrates the use of a state diagram with an extract of the TE state diagram. The state explained is “F3 deactivated”.

The state may be entered:

– from the unconditional states (ARL, RES, TM)
– from state “F3 pending deactivation”, “F3 power up”, “F4 pending activation” or “F5 unsynchronized” after the C/I command “DI” has been received.

The following informations are transmitted:

– INFO 0 (no signal) is sent on the S/T-interface.
– C/I message “DC” is issued on the IOM-2 interface.

The state may be left by either of the following methods:

– Leave for the state “F3 power up” in case C/I = “TIM” code is received.
– Leave for state “F4 pending activation” in case C/I = AR8 or AR10 is received.
– Leave for the state “F6 synchronized” after INFO 2 has been recognized on the S/T-interface.
– Leave for the state “F7 activated” after INFO 4 has been recognized on the S/T-interface.

– Leave for any unconditional state if any unconditional C/I command is received.

As can be seen from the transition criteria, combinations of multiple conditions are possible as well. A “∗” stands for a logical AND combination. And a “+” indicates a logical OR combination.

The sections following the state diagram contain detailed information on all states and signals used.

Test Signals

- Send Single Pulses (SSP)
  One pulse with a width of one bit period per frame with alternating polarity.
- Send Continuous Pulses (SCP)
  Continuous pulses with a pulse width of one bit period.

External Layer-1 Statemachine

Instead of using the integrated layer-1 statemachine it is also possible to implement the layer-1 statemachine completely in software.

The internal layer-1 statemachine can be disabled by setting the L1SW bit in the TR_CONF0 register to ’1’.

The transmitter is completely under control of the microcontroller via register TR_CMD.

The status of the receiver is stored in register TR_STA and has to be evaluated by the microcontroller. This register is updated continuously. If not masked a RIC interrupt is generated by any change of the register contents. The interrupt is cleared after a read access to this register.

The RIC interrupt can also be used if the inchip layer-1 statemachine is enabled. This is not required in most applications as the important status changes will result in C/I code change interrupts. However, RIC provides the advantage that status changes on S can be indicated much faster to the µC then a C/I code change interrupt.

Reset States

An active signal on the reset pin $\overline{\text{RESET}}$ brings the transceiver state machine to the reset state. The function of this reset event is identical to the C/I code RES concerning the state machine.

C/I Codes in Reset State

In the reset state the C/I code 0001 (RES) is valid. This state is entered either after a hardware reset ($\overline{\text{RESET}}$) or after the C/I code RES.

5.3.1 State Machine TE Mode

5.3.1.1 State Transition Diagram (TE)

Figure 56 shows the state transition diagram of the SIUC-BA state machine. **Figure 57** shows this for the unconditional transitions (Reset, Loop, Test Mode i).

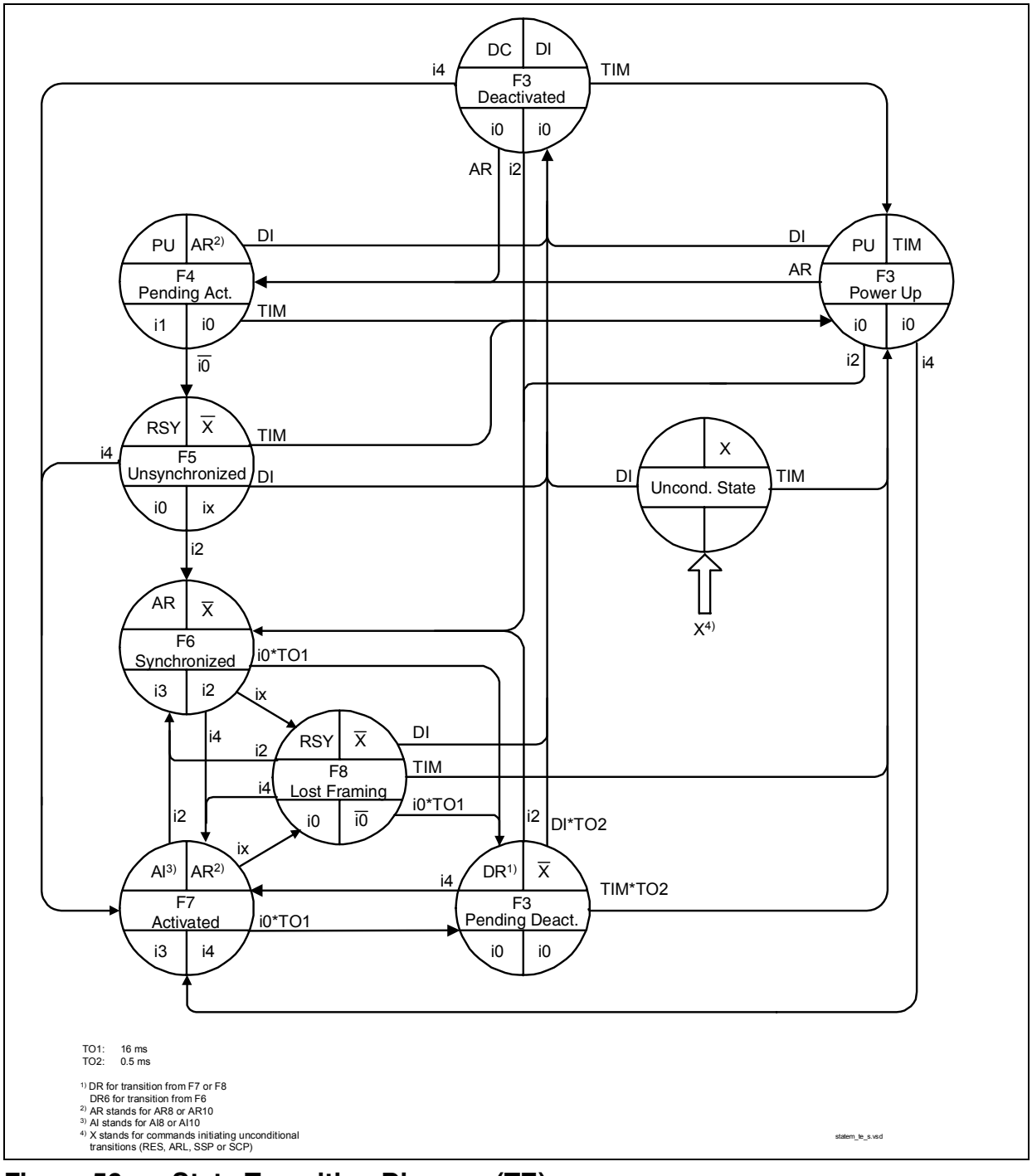


Figure 56 State Transition Diagram (TE)

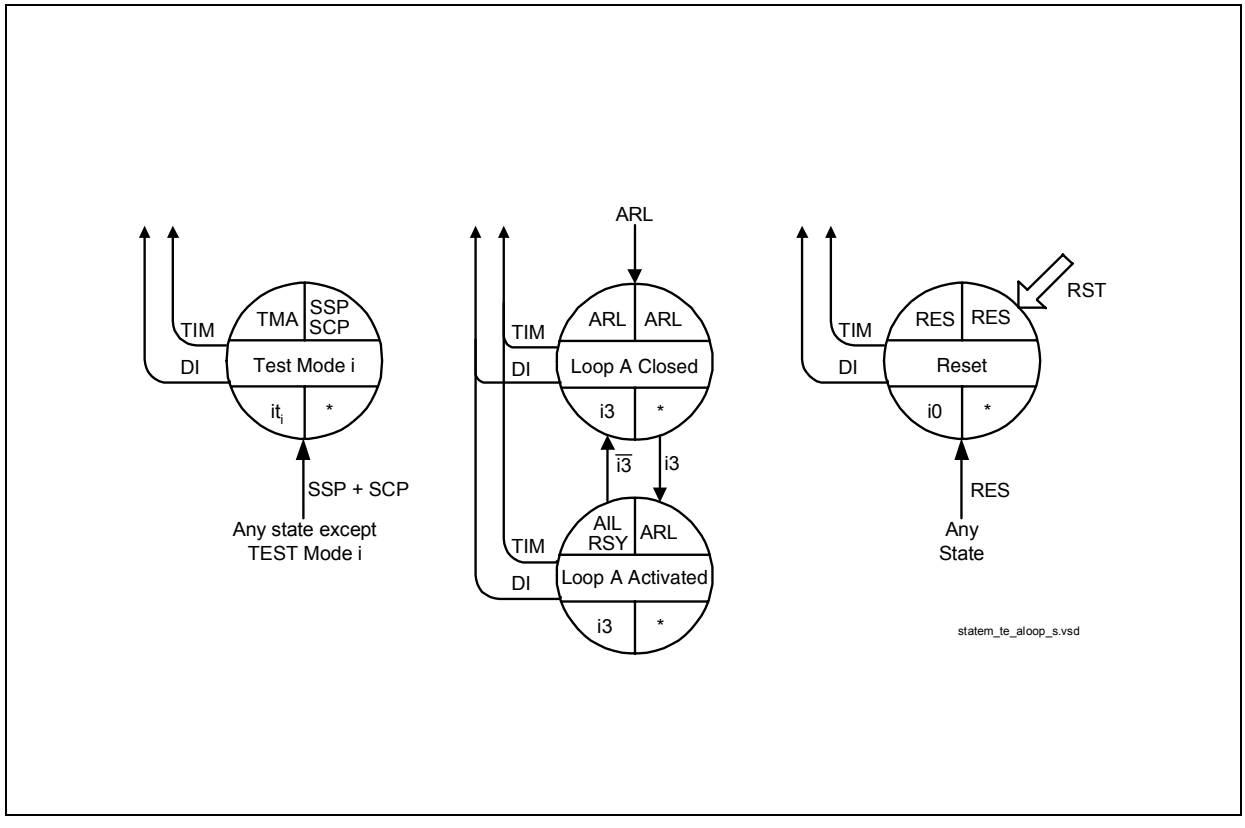


Figure 57 State Transition Diagram of Unconditional Transitions (TE)

5.3.1.2 States (TE)

F3 Pending Deactivation

State after deactivation from the S/T interface by info 0. Note that no activation from the terminal side is possible starting from this state. A 'DI' command has to be issued to enter the state 'Deactivated State'.

F3 Deactivated State

The S/T interface is deactivated and the clocks are deactivated 500 µs after entering this state and receiving info 0 if the CFS bit of the SIUC-BA Configuration Register is set to "0". Activation is possible from the S/T interface and from the IOM-2 interface. The bit TR_CMD.PD is set and the analog part is powered down (TR_CMD.PD is only set if ISTATR.LD is inactive).

F3 Power Up

The S/T interface is deactivated (info 0 on the line) and the clocks are running.

F4 Pending Activation

The SIUC-X transmits info 1 towards the network, waiting for info 2.

F5 Unsynchronized

Any signal except info 2 or 4 detected on the S/T interface.

F6 Synchronized

The receiver has synchronized and detects info 2. Info 3 is transmitted to synchronize the NT.

F7 Activated

The receiver has synchronized and detects info 4. All user channels are now conveyed transparently to the IOM-2 interface.
To transfer user channels transparently to the S/T interface either the command AR8 or AR10 has to be issued and TR_STA.FSYN must be “1” (signal from remote side must be synchronous).

F8 Lost Framing

The receiver has lost synchronization in the states F6 or F7 respectively.

Unconditional States

Loop A Closed (internal or external)

The SIUC-X loops back the transmitter to the receiver and activates by transmission of info 3. The receiver has not yet synchronized.
For a non transparent internal loop the DIS_TX bit of register TR_CONF2 has to be set to ’1’.

Loop A Activated (internal or external)

The receiver has synchronized to info 3. Data may be sent. The indication “AIL” is output to indicate the activated state. If the loop is closed internally and the S/T line awake detector detects any signal on the S/T interface, this is indicated by “RSY”.

Test Mode - SSP

Single alternating pulses are transmitted to the S/T-interface resulting in a frequency of the fundamental mode of 2 kHz.

Test Mode - SCP

Continuous alternating pulses are transmitted to the S/T-interface resulting in a frequency of the fundamental mode of 96 kHz.

5.3.1.3 C/I Codes (TE)

Command	Abbr.	Code	Remark
Activation Request with priority class 8	AR8	1000	Activation requested by the SIUC-X, D-channel priority set to 8 (**see note**)
Activation Request with priority class 10	AR10	1001	Activation requested by the SIUC-X, D-channel priority set to 10 (**see note**)
Activation Request Loop	ARL	1010	Activation requested for the internal or external Loop A (**see note**). For a non transparent internal loop bit DIS_TX of register TR_CONF2 has to be set to '1' additionally.
Deactivation Indication	DI	1111	Deactivation Indication
Reset	RES	0001	Reset of the layer-1 statemachine
Timing	TIM	0000	Layer-2 device requires clocks to be activated
Test mode SSP	SSP	0010	One AMI-coded pulse transmitted in each frame, resulting in a frequency of the fundamental mode of 2 kHz
Test mode SCP	SCP	0011	AMI-coded pulses transmitted continuously, resulting in a frequency of the fundamental mode of 96 kHz

Note: In the activated states (AI8, AI10 or AIL indication) the 2B+D channels are only transferred transparently to the S/T interface if one of the three "Activation Request" commands is permanently issued.

Indication	Abbr.	Code	Remark
Deactivation Request	DR	0000	Deactivation request via S/T-interface if left from F7/F8
Reset	RES	0001	Reset acknowledge
Test Mode Acknowledge	TMA	0010	Acknowledge for both SSP and SCP
Resynchronization during level detect	RSY	0100	Signal received, receiver not synchronous
Deactivation Request from F6	DR6	0101	Deactivation Request from state F6

Indication	Abbr.	Code	Remark
Power up	PU	0111	IOM-2 interface clocking is provided
Activation request	AR	1000	Info 2 received
Activation request loop	ARL	1010	Internal or external loop A closed
Illegal Code Violation	CVR	1011	Illegal code violation received. This function has to be enabled by setting the EN_ICV bit of register TR_CONF0.
Activation indication loop	AIL	1110	Internal or external loop A activated
Activation indication with priority class 8	AI8	1100	Info 4 received, D-channel priority is 8 or 9.
Activation indication with priority class 10	AI10	1101	Info 4 received, D-channel priority is 10 or 11.
Deactivation confirmation	DC	1111	Clocks are disabled if CFS bit of register MODE1 is set to ’1’, quiescent state

5.3.1.4 Infos on S/T (TE)

Receive Infos on S/T (Downstream)

Name	Abbr.	Description
info 0	i0	No signal on S/T
info 2	i2	4 kHz frame A=’0’
info 4	i4	4 kHz frame A=’1’
info X	ix	Any signal except info 2 or info 4

Transmit Infos on S/T (Upstream)

Name	Abbr.	Description
info 0	i0	No signal on S/T
info 1	i1	Continuous bit sequence of the form ’00111111’
info 3	i3	4 kHz frame
Test info 1	it_1	SSP - Send Single Pulses
Test info 2	it_2	SCP - Send Continuous Pulses

5.3.2 Command/ Indicate Channel Codes (C/I0) - Overview

The table below presents all defined C/I0 codes. A command needs to be applied continuously until the desired action has been initiated. Indications are strictly state orientated. Refer to the state diagrams in the previous sections for commands and indications applicable in various states.

Code				TE	
				Cmd	**Ind**
0	0	0	0	TIM	DR
0	0	0	1	RES	RES
0	0	1	0	SSP	TMA
0	0	1	1	SCP	–
0	1	0	0	–	RSY
0	1	0	1	–	DR6
0	1	1	0	–	–
0	1	1	1	–	PU
1	0	0	0	AR8	AR
1	0	0	1	AR10	–
1	0	1	0	ARL	ARL
1	0	1	1	–	CVR
1	1	0	0	–	AI8
1	1	0	1	–	AI10
1	1	1	0	–	AIL
1	1	1	1	DI	DC

5.4 Control Procedures

5.4.1 Example of Activation/Deactivation

An example of an activation/deactivation of the S/T interface initiated by the terminal with the time relationships mentioned in the previous chapters is shown in **Figure 58**.

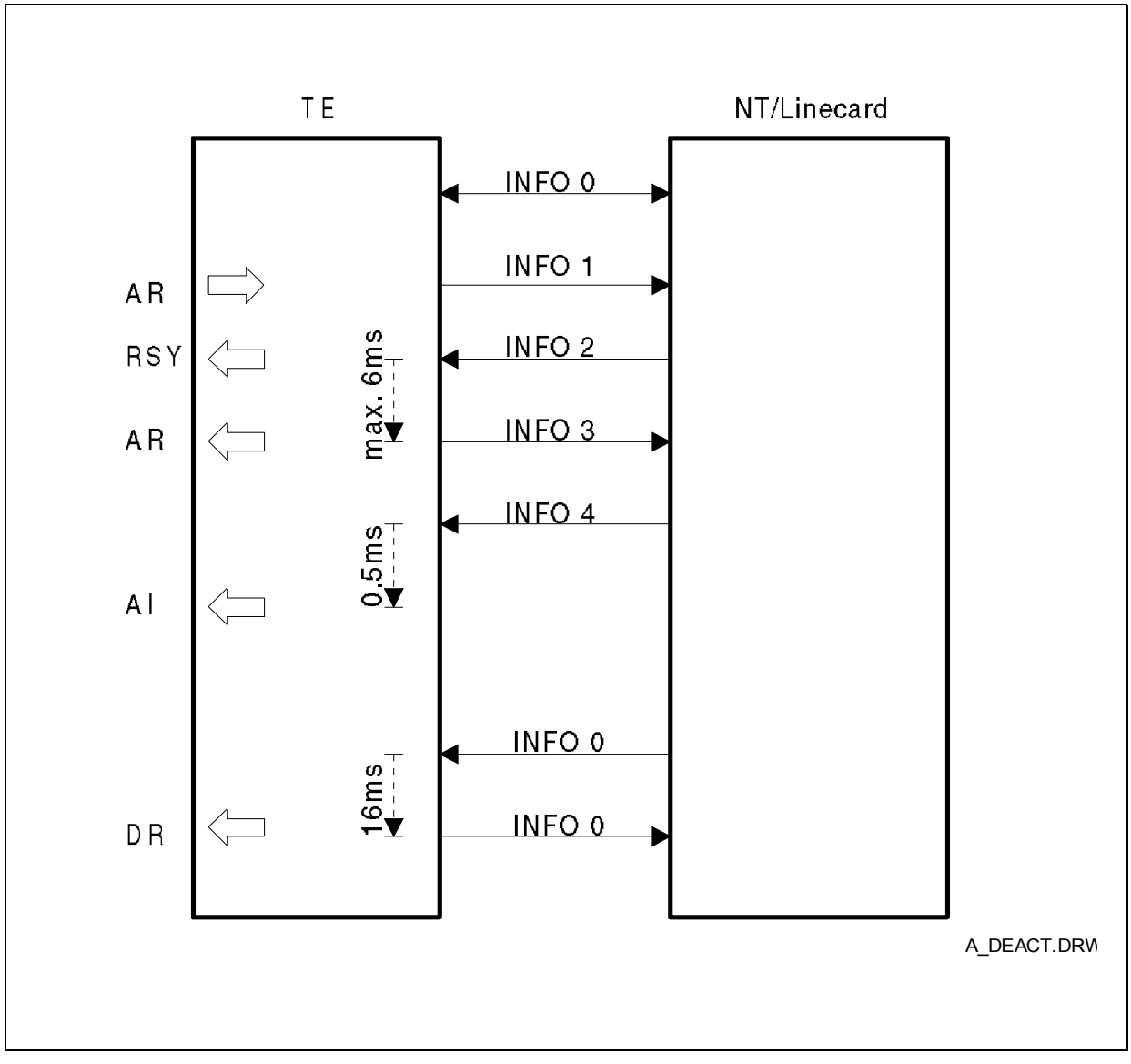
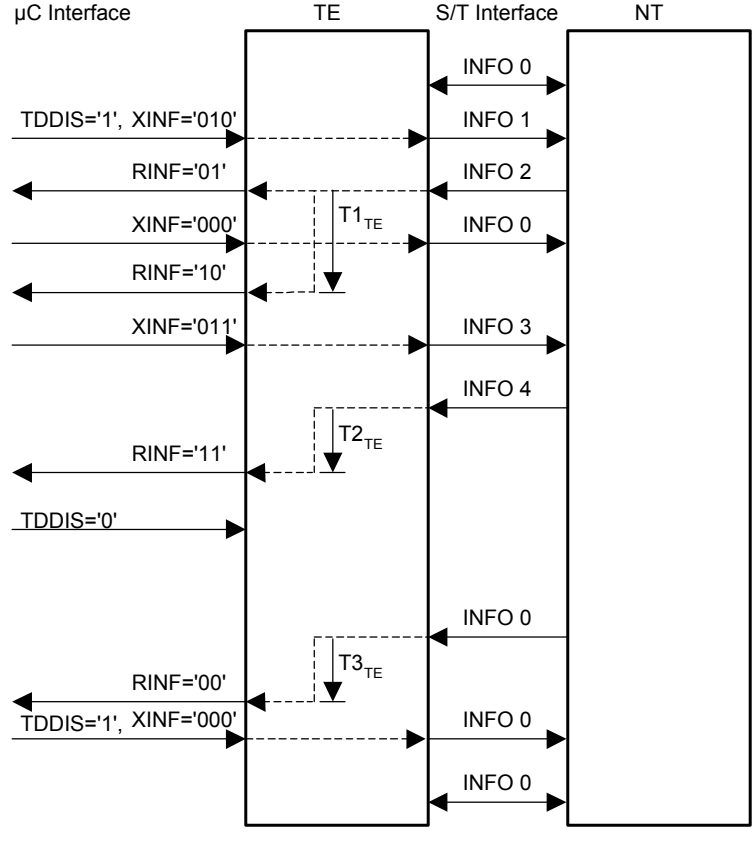


Figure 58 Example of Activation/Deactivation Initiated by the Terminal

5.4.2 Activation initiated by the Terminal

INFO 1 has to be transmitted as long as INFO 0 is received.
INFO 0 has to be transmitted thereafter as long as no valid INFO (INFO 2 or INFO 4) is received.
After reception of INFO 2 or INFO 4 transmission of INFO 3 has to be started.
Data can be transmitted if INFO 4 has been received.



$T1_{TE}$: 2 to 6 frames (0.5 ms to 1.5 ms)
$T2_{TE}$: 2 frames (0.5 ms)
$T3_{TE}$: 4 frames (1 ms)

act_deac_te-ext_s.vsd

Figure 59 Example of Activation/Deactivation initiated by the Terminal (TE). Activation/Deactivation completely under Software Control

Note: RINF and XINF are Receive- and Transmit-INFOs of register TR_STA.

5.4.3 Activation initiated by the Network Termination NT

INFO 0 has to be transmitted as long as no valid INFO (INFO 2 or INFO 4) is received. After reception of INFO 2 or INFO 4 transmission of INFO 3 has to be started. Data can be transmitted if INFO 4 has been received.

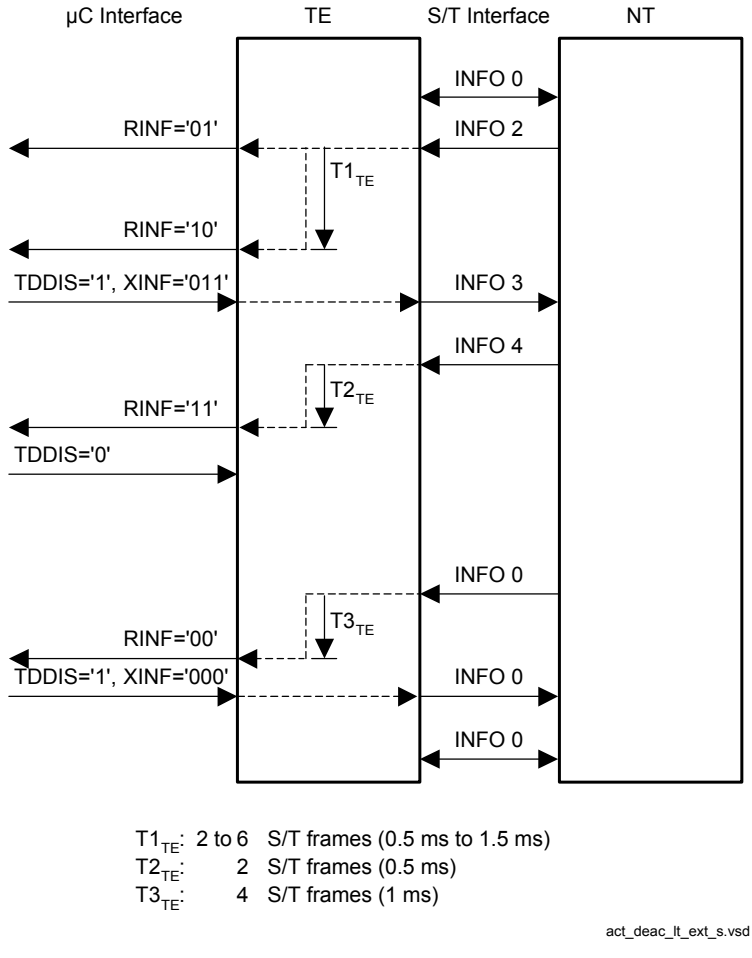


Figure 60 Example of Activation/Deactivation initiated by the Network Termination (NT). Activation/Deactivation completely under Software Control

Note: RINF and XINF are Receive- and Transmit-INFOs of register TR_STA.

5.5 IOM-2 Interface

The SIUC-BA supports the IOM-2 interface in linecard mode and in terminal mode with single clock and double clock. The IOM-2 interface consists of four lines: FSC, DCL, DD and DU. The rising edge of FSC indicates the start of an IOM-2 frame. The DCL and the BCL clock signals synchronize the data transfer on both data lines DU and DD. The DCL is twice the bit rate, the BCL rate is equal to the bit rate. The bits are shifted out with the rising edge of the first DCL clock cycle and sampled at the falling edge of the second clock cycle.

The IOM-2 interface can be enabled/disabled with the DIS_IOM bit in the IOM_CR register.

A DCL signal and BCL signal (pin BCL/SCLK) output is provided and the FSC signal is generated by the receive DPLL which synchronizes it to the received S/T frame.

The BCL clock together with the serial data strobe signal SDS/$\overline{\text{RSTO}}$ (multiplexed with reset output) can be used to connect time slot oriented standard devices to the IOM-2 interface. If the transceiver is disabled (TR_CONF0.DIS_TR) the DCL and FSC pins become input and the HDLC part can still work via IOM-2. In this case the clock mode bit (IOM_CR.CLKM) selects between a double clock and a single clock input for DCL.

The clock rate/frequency of the IOM-2 signals in TE mode are:
DD, DU: 768 kbit/s
FSC (o): 8 kHz
DCL (o): 1536 kHz (double clock rate)
BCL (o): 768 kHz (single clock rate)
Option - Transceiver disabled (DIS_TR = ’1’):
FSC (i): 8 kHz
DCL (i): 1536 ... 4096 kHz, in steps of 512 kHz (double clock rate)

IOM-2 Frame Structure (TE Mode)

The frame structure on the IOM-2 data ports (DU,DD) of a master device in IOM-2 terminal mode is shown in **Figure 61**.

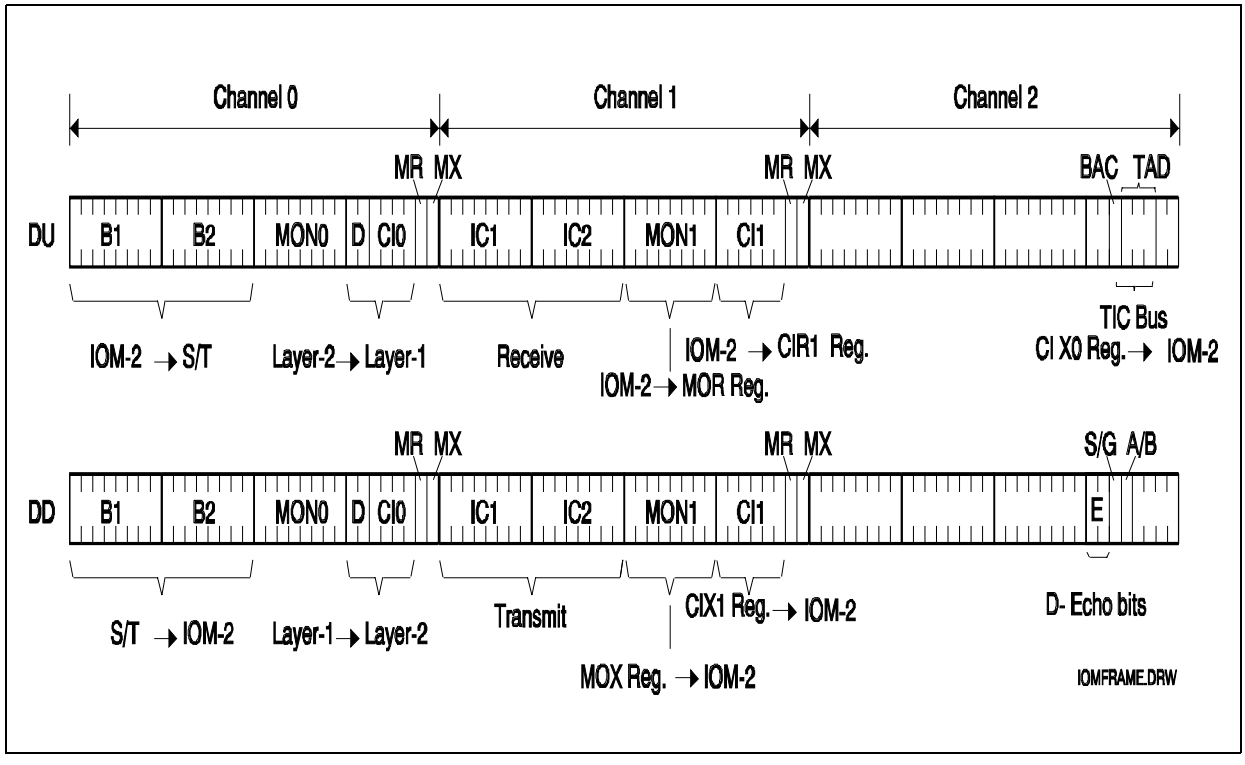


Figure 61 IOM-2 Frame Structure in Terminal Mode

The frame is composed of three channels

- Channel 0 contains 144-kbit/s of user and signaling data (2B + D), a MONITOR programming channel (MON0) and a command/indication channel (CI0) for control and programming of the layer-1 transceiver.
- Channel 1 contains two 64-kbit/s intercommunication channels (IC) plus a MONITOR and command/indicate channel (MON1, CI1) to program or transfer data to other IOM-2 devices.
- Channel 2 is used for the D-channel access mechanism (TIC-bus, S/G-bit). Additionally, channel 2 supports further IC and MON channels.

5.5.1 IOM-2 Handler

The IOM-2 handler offers a great flexibility for handling the data transfer between the different functional units of the SIUC-BA and voice/data devices connected to the IOM-2 interface. Additionally it provides a microcontroller access to all timeslots of the IOM-2 interface via the four controller data access registers (CDA). **Figure 62** shows the architecture of the IOM-2 handler. For illustrating the functional description it contains all configuration and control registers of the IOM-2 handler.

The PCM data of the functional units

- Transceiver (TR) and the
- Controller data access (CDA)
- B-channel HDLC controllers

can be configured by programming the time slot and data port selection registers (TSDP). With the TSS bits (Time Slot Selection) the PCM data of the functional units can be assigned to each of the 32 PCM time slots of the IOM-2 frame. With the DPS bit (Data Port Selection) the output of each functional unit is assigned to DU or DD respectively. The input is assigned vice versa. With the data control registers (xxx_CR) the access to the data of the functional units can be controlled by setting the corresponding control bits (EN, SWAP).

The IOM-2 handler also provides access to the

- MONITOR channel (MON)
- C/I channels (C/I0,C/I1)
- TIC bus (TIC) and
- HDLC control

The access to these channels is controlled by the registers TR_CR, MON_CR, DCI_CR and BCHx_CR.
The IOM-2 interface with the Serial Data Strobe SDS is controlled by the control registers IOM_CR and SDS_CR.

The reset configuration of the SIUC-BA IOM-2 handler corresponds to the defined frame structure and data ports of a master device in IOM-2 terminal mode (see **Figure 61**).

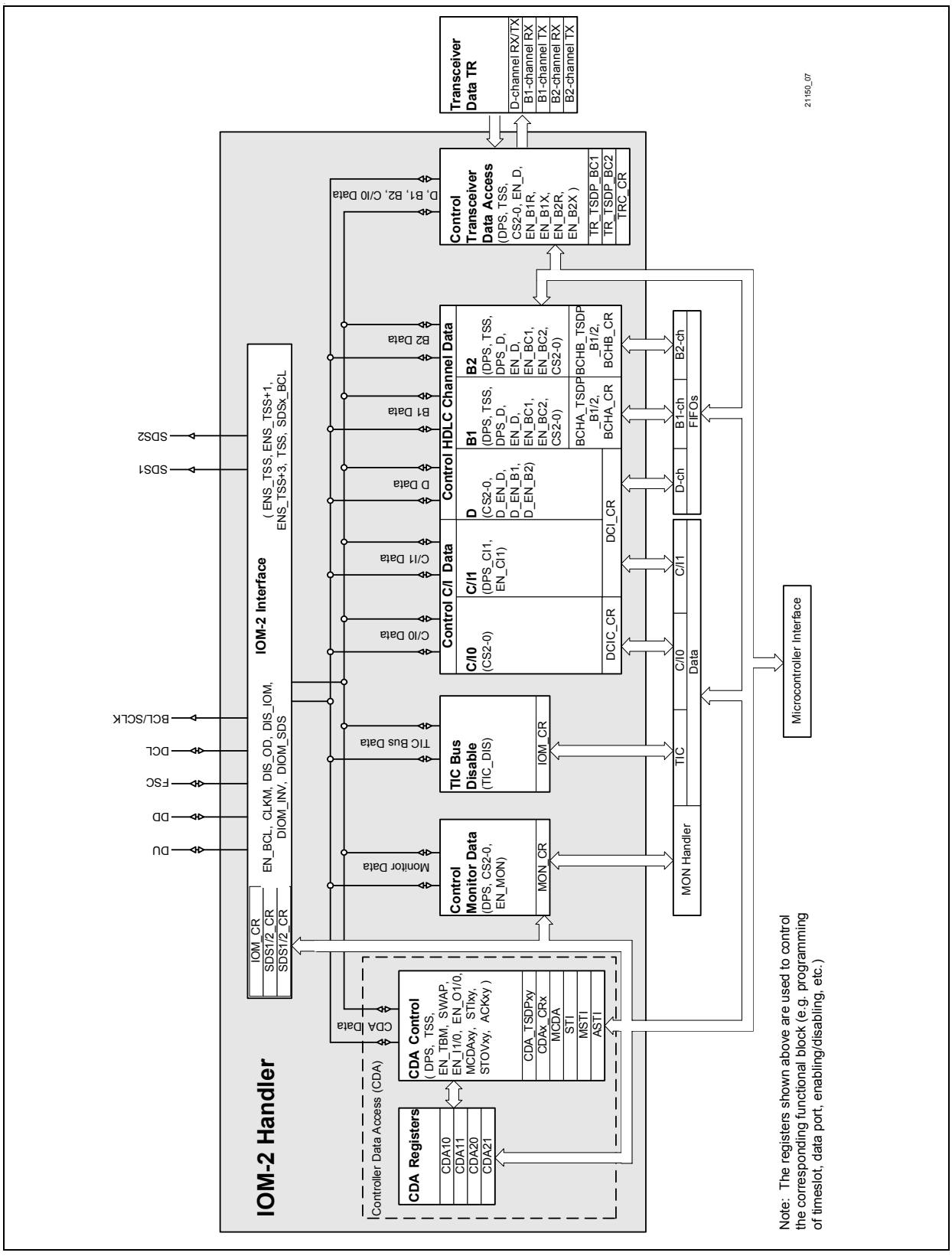


Figure 62 Architecture of the IOM Handler (Example Configuration)

5.5.1.1 Controller Data Access (CDA)

With its four controller data access registers (CDA10, CDA11, CDA20, CDA21) the SIUC-BA IOM-2 handler provides a very flexible solution for the host access to up to 32 IOM-2 time slots.

The functional unit CDA (controller data access) allows with its control and configuration registers

- looping of up to four independent PCM channels from DU to DD or vice versa over the four CDA registers
- shifting of two independent PCM channels to another two independent PCM channels on both data ports (DU, DD). Between reading and writing the data can be manipulated (processed with an algorithm) by the microcontroller. If this is not the case a switching function is performed
- monitoring of up to four time slots on the IOM-2 interface simultaneously
- microcontroller read and write access to each PCM timeslot

The access principle which is identical for the two channel register pairs CDA10/11 and CDA20/21 is illustrated in **Figure 63**. Each of the index variables x,y used in the following description can be 1 or 2 for x and 0 or 1 for y. The prefix 'CDA_' from the register names has been omitted for simplification.

To each of the four CDAxy data registers a TSDPxy register is assigned by which the time slot and the data port can be determined. With the TSS (Time Slot Selection) bits a time slot from 0...31 can be selected. With the DPS (Data Port Selection) bit the output of the CDAxy register can be assigned to DU or DD respectively. The time slot and data port for the output of CDAxy is always defined by its own TSDPxy register. The input of CDAxy depends on the SWAP bit in the control registers CRx.

- If the SWAP bit = '0' (swap is disabled) the time slot and data port for the input and output of the CDAxy register is defined by its own TSDPxy register.
- If the SWAP bit = '1' (swap is enabled) the input port and timeslot of the CDAx0 is defined by the TSDP register of CDAx1 and the input port and timeslot of CDAx1 is defined by the TSDP register of CDAx0. The input definition for timeslot and data port CDAx0 are thus swapped to CDAx1 and for CDAx1 swapped to CDAx0. The output timeslots are not affected by SWAP.

The input and output of every CDAxy register can be enabled or disabled by setting the corresponding EN (-able) bit in the control register CDAx_CR. If the input of a register is disabled the output value in the register is retained. In the normal mode (SWAP=0) the input of CDAx0 and CDAx1 is enabled via EN_I0 and EN_I1, respectively. If SWAP=1 EN_I0 controls the input of CDAx1 and EN_I1 controls the input of CDAx0. The output control (EN_O0 and EN_O1) is not affected by SWAP.

Usually one input and one output of a functional unit (transceiver, HDLC controllers, CDA registers) is programmed to a timeslot of IOM-2 (e.g. for B-channel transmission in upstream direction the HDLC controller writes data onto IOM and the transceiver reads

data from IOM). For monitoring data in such cases a CDA register is programmed as described below under “Monitoring Data”. Besides that none of the IOM timeslots must be assigned more than one input and output of any functional unit.

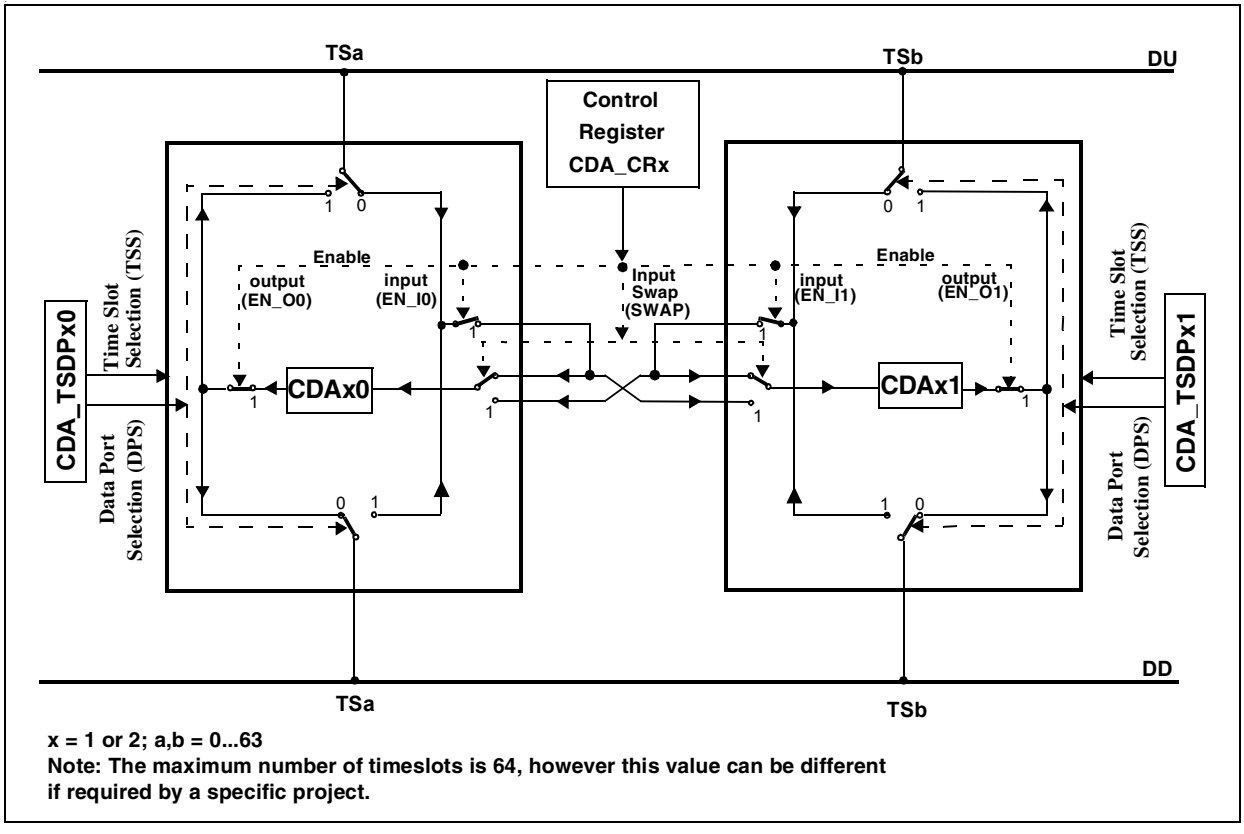


Figure 63 Data Access via CDAx1 and CDAx2 register pairs

Looping and Shifting Data

Figure 64 gives examples for typical configurations with the above explained control and configuration possibilities with the bits TSS, DPS, EN and SWAP in the registers TSDPxy or CDAx_CR:

a) looping IOM-2 time slot data from DU to DD or vice versa (SWAP = 0)

b) shifting data from TSa to TSb and TSc to TSd in both transmission directions (SWAP = 1)

c) switching data from TSa to TSb and looping from DU to DD or TSc to TSd and looping from DD to DU respectively

TSa is programmed in TSDP10, TSb in TSDP11, TSc in TSDP20 and TSd in TSDP21. It should also be noted that the input control of CDA registers is swapped if SWAP=1 while the output control is not affected (e.g. for CDA11 in example a: EN_I1=1 and EN_O1=1, whereas for CDA11 in example b: EN_I0=1 and EN_O1=1).

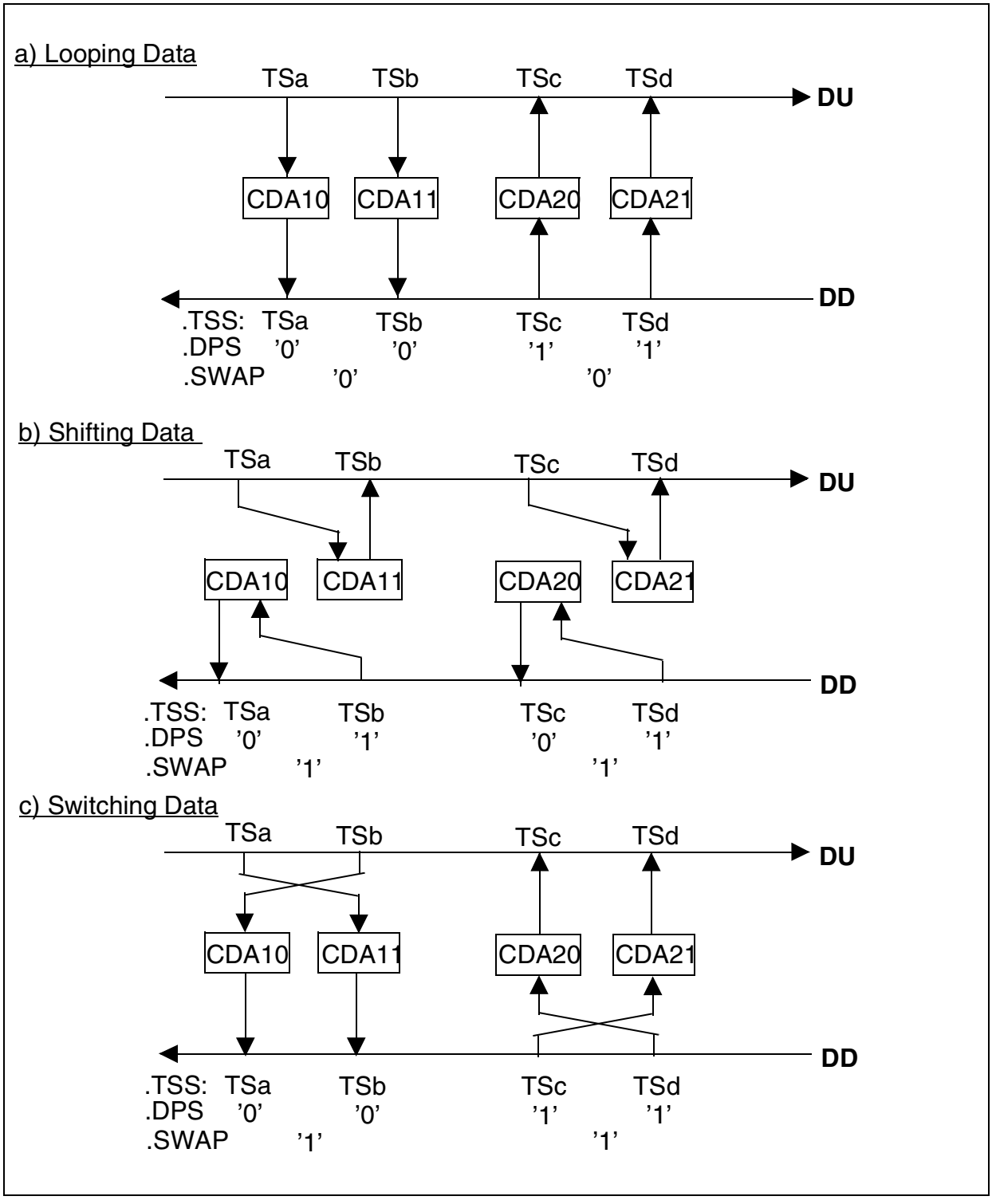


Figure 64 Examples for Data Access via CDAxy Registers
a) Looping Data
b) Shifting (Switching) Data
c) Shifting and Looping Data

Figure 65 shows the timing of looping TSa from DU to DD (a = 0...11) via CDAxy register. TSa is read in the CDAxy register from DU and is written one frame later on DD.

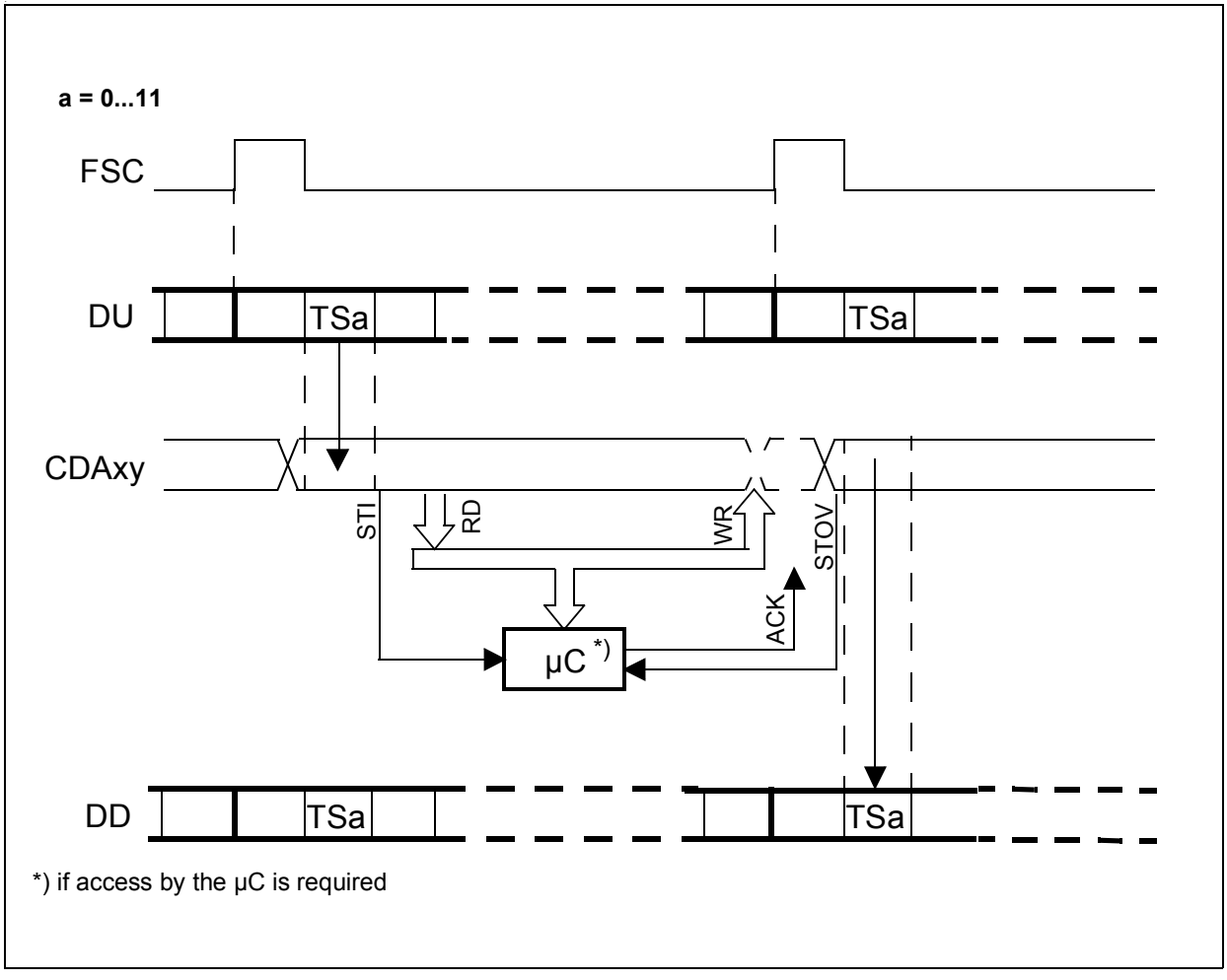


Figure 65 Data Access when Looping TSa from DU to DD

Figure 66 shows the timing of shifting data from TSa to TSb on DU(DD). In **Figure 66a)** shifting is done in one frame because TSa and TSb didn't succeed direct one another (a,b = 0...9 and b ≥ a+2). In **Figure 66b)** shifting is done from one frame to the following frame. This is the case when the time slots succeed one other (b = a+1) or b is smaller than a (b < a).

At looping and shifting the data can be accessed by the controller between the synchronous transfer interrupt (STI) and the status overflow interrupt (STOV). STI and STOV are explained in the section 'Synchronous Transfer'. If there is no controller intervention the looping and shifting is done autonomous.

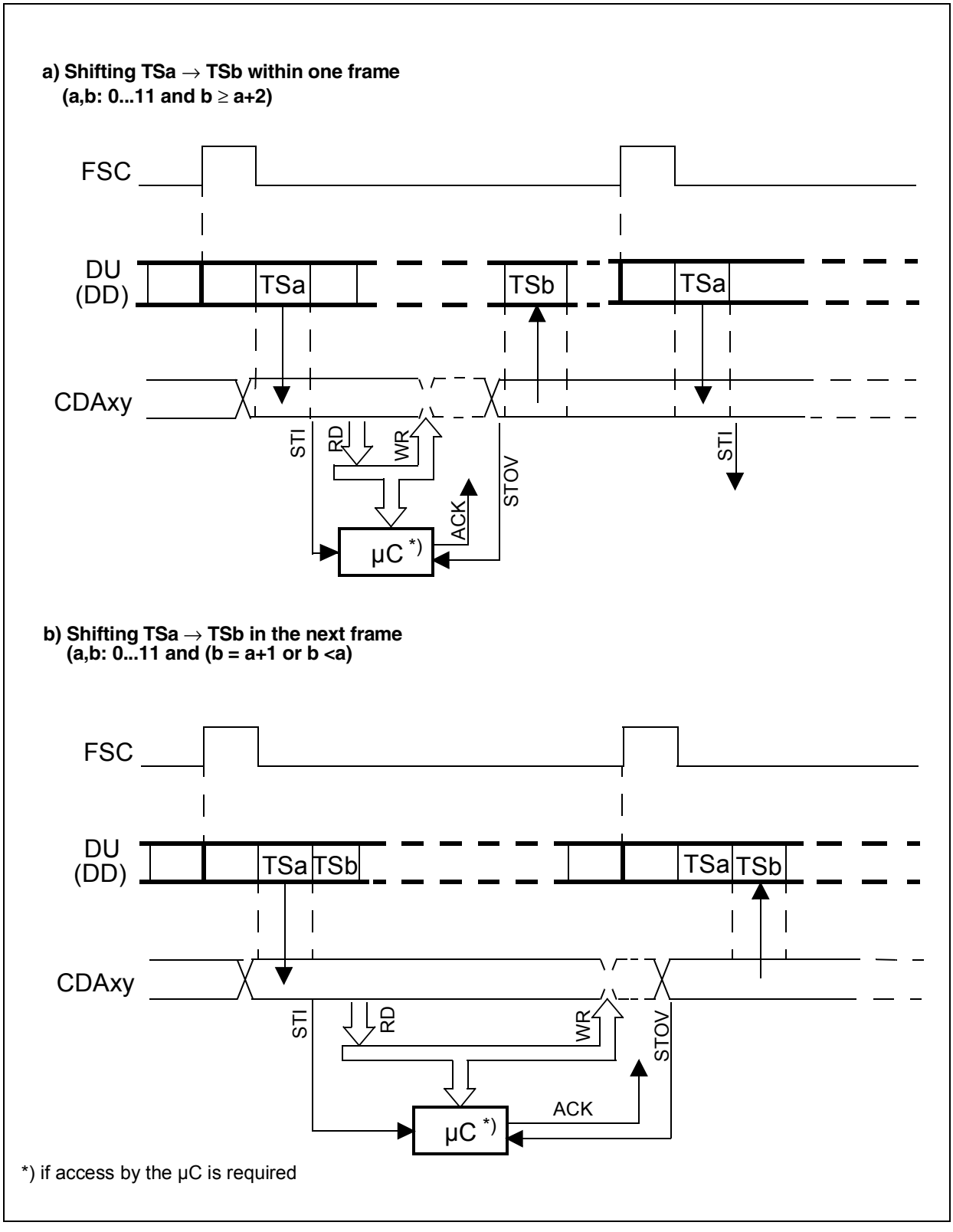


Figure 66 Data Access when Shifting TSa to TSb on DU (DD)

Monitoring Data

Figure 67 gives an example for monitoring of two IOM-2 time slots each on DU or DD simultaneously. For monitoring on DU and/or DD the channel registers with even numbers (CDA10, CDA20) are assigned to time slots with even numbers TS(2n) and the channel registers with odd numbers (CDA11, CDA21) are assigned to time slots with odd numbers TS(2n+1). The user has to take care of this restriction by programming the appropriate time slots.

This mode is only valid if two blocks (e.g. transceiver and HDLC controller) are programmed to these timeslots and communicate via IOM-2. However, if only one block is programmed to this timeslot the timeslots for CDAx0 and CDAx1 can be programmed completely independently.

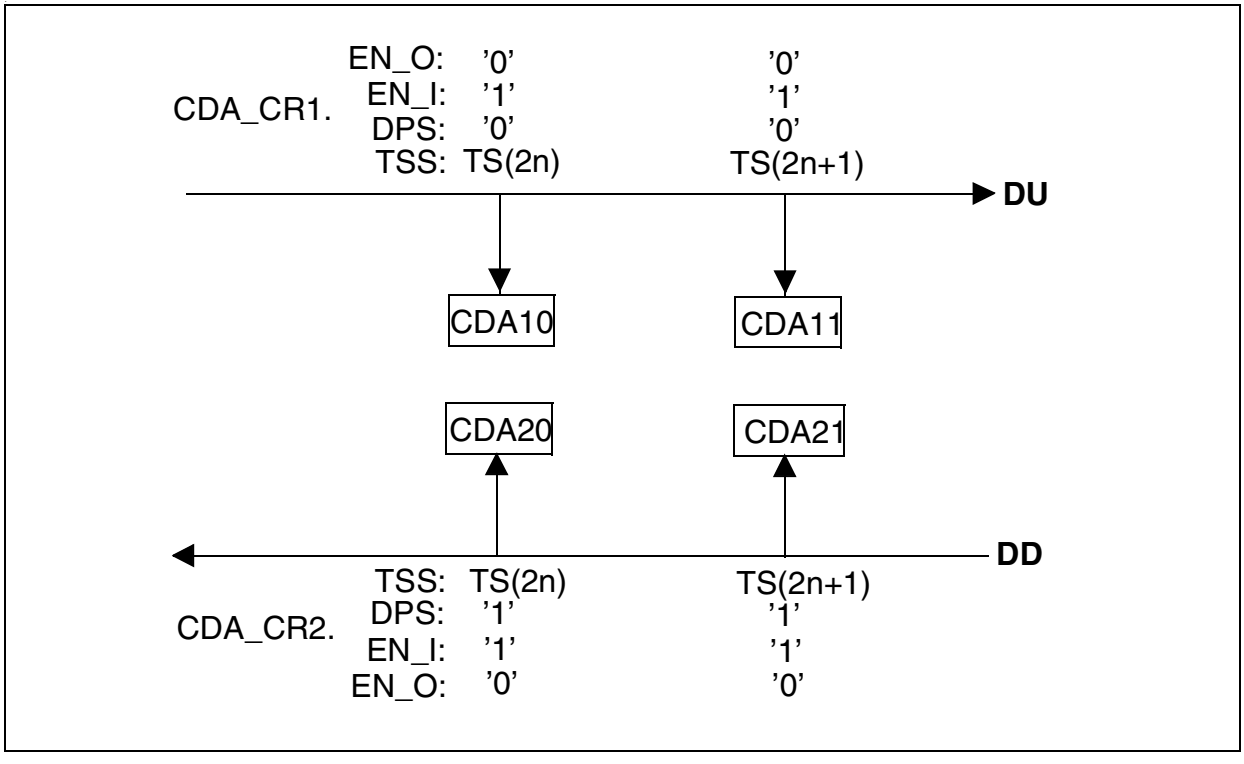


Figure 67 Example for Monitoring Data

Monitoring TIC Bus

Monitoring the TIC bus (TS11) is handled as a special case. The TIC bus can be monitored with the registers CDAx0 by setting the EN_TBM (Enable TIC Bus Monitoring) bit in the control registers CRx. In this special case the TSDPx0 must be set to 08_h for monitoring from DU or 88_h for monitoring from DD respectively. By this it is possible to monitor the TIC bus (TS11) and the odd numbered D-channel (TS3) simultaneously on DU and DD.

Synchronous Transfer

While looping, shifting and switching the data can be accessed by the controller between the synchronous transfer interrupt (STI) and the status overflow interrupt (STOV).

The microcontroller access to the CDAxy registers can be synchronized by means of four programmable synchronous transfer interrupts (STIxy) and synchronous transfer overflow interrupts (STOVxy) in the STI register.
Depending on the DPS bit in the corresponding CDA_TSDPxy register the STIxy is generated two (for DPS='0') or one (for DPS='1') BCL clock after the selected time slot (CDA_TSDPxy.TSS). One BCL clock is equivalent to two DCL clocks. It should be noted that synchronous interrupts are only generated if the corresponding CDA register input is enabled even if the synchronous interrupts are used for any other purpose than CDA register access. In order to enable the STI interrupts the input of the corresponding CDA register has to be enabled. This is also valid if only a synchronous write access (output) is wanted. The enabling of the output alone does not effect an STI interrrupt. In order to enable the STOV interrupts the output of the corresponding CDA register has to be enabled.

In the following description the index xy_0 and xy_1 are used to refer to two different interrupt pairs (STI/STOV) out of the four CDA interrupt pairs (STI10/STOV10, STI11/ STOV11, STI20/STOV20, STI21/STOV21).

An $STOVxy_0$ is related to its $STIxy_0$ and is only generated if $STIxy_0$ is enabled and not acknowledged. However, if $STIxy_0$ is masked, the $STOVxy_0$ is generated for any other $STIxy_1$ which is enabled and not acknowledged.

Table 18 gives some examples for that. It is assumed that an STOV interrupt is only generated because an STI interrupt was not acknowledged before.

In example 1 only the $STIxy_0$ is enabled and thus $STIxy_0$ is only generated. If no STI is enabled, no interrupt will be generated even if STOV is enabled (example 2).
In example 3 $STIxy_0$ is enabled and generated and the corresponding $STOVxy_0$ is disabled. $STIxy_1$ is disabled but its $STOVxy_1$ is enabled, and therefore $STOVxy_1$ is generated due to $STIxy_0$. In example 4 additionally the corresponding $STOVxy_0$ is enabled, so $STOVxy_0$ and $STOVxy_1$ are both generated due to $STIxy_0$.
In example 5 additionally the $STIxy_1$ is enabled with the result that $STOVxy_0$ is only generated due to $STIxy_0$ and $STOVxy_1$ is only generated due to $STIxy_1$.

Compared to the previous example $STOVxy_0$ is disabled in example 6, so $STOVxy_0$ is not generated and $STOVxy_1$ is only generated for $STIxy_1$ but not for $STIxy_0$.

Compared to example 5 in example 7 a third $STOVxy_2$ is enabled and thus STOVxy2 is generated additionally for both $STIxy_0$ and $STIxy_1$.

Table 18 Examples for Synchronous Transfer Interrupts

Enabled Interrupts (Register MSTI)		Generated Interrupts (Register STI)		
STI	**STOV**	**STI**	**STOV**	
xy_0	-	xy_0	-	Example 1
-	xy_0	-	-	Example 2
xy_0	xy_1	xy_0	xy_1	Example 3
xy_0	xy_0 ; xy_1	xy_0	xy_0 ; xy_1	Example 4
xy_0 ; xy_1	xy_0 ; xy_1	xy_0 xy_1	xy_0 xy_1	Example 5
xy_0 ; xy_1	xy_1	xy_0 xy_1	- xy_1	Example 6
xy_0 ; xy_1	xy_0 ; xy_1 ; xy_2	xy_0 xy_1	xy_0 ; xy_2 xy_1 ; xy_2	Example 7

An STOV interrupt is not generated if all stimulating STI interrupts are acknowledged.

An STIxy must be acknowledged by setting the ACKxy bit in the ASTI register until two BCL clocks (for DPS='0') or one BCL clocks (for DPS='1') before the time slot which is selected for the appropriate STIxy.

The interrupt structure of the synchronous transfer is shown in **Figure 68**.

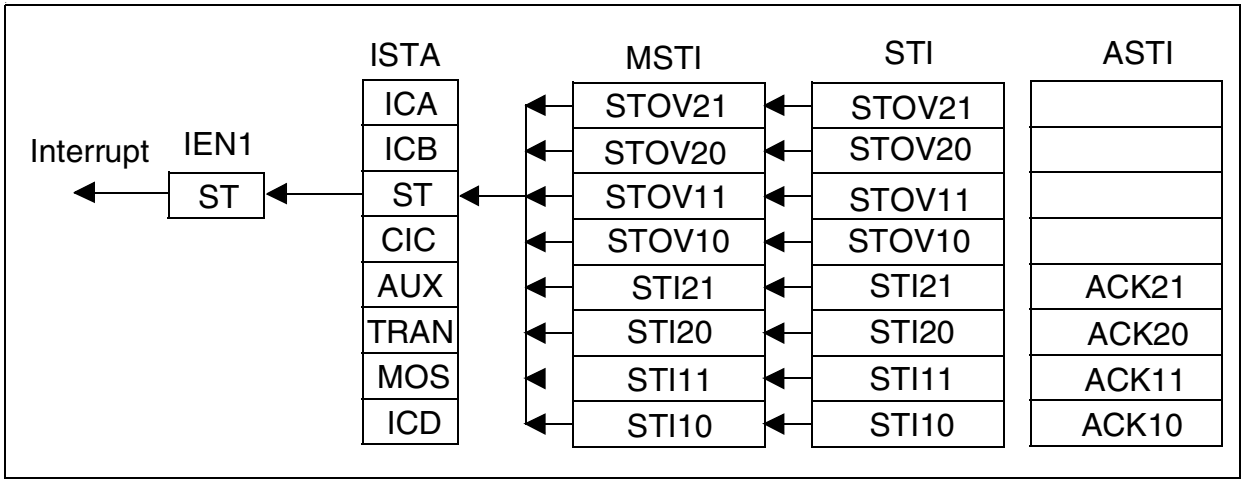


Figure 68 Interrupt Structure of the Synchronous Data Transfer

Figure 69 shows some examples based on the timeslot structure. Figure a) shows at which point in time an STI and STOV interrrupt is generated for a specific timeslot. Figure b) is identical to example 3 above, figure c) corresponds to example 5 and figure d) shows example 4.

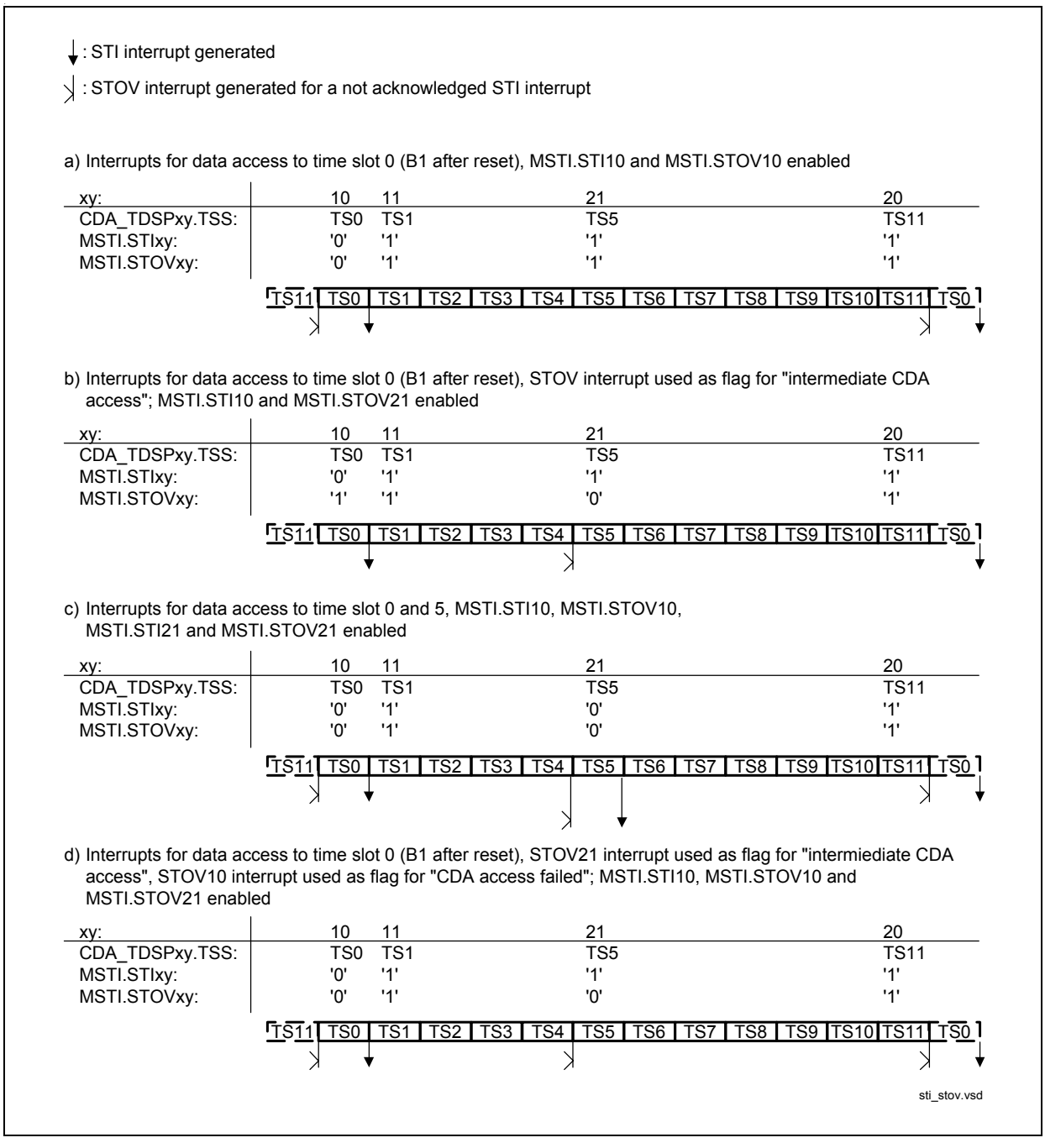


Figure 69 Examples for the Synchronous Transfer Interrupt Control with one enabled STIxy

5.5.2 IDSL Support

5.5.2.1 IOM-2 Interface

The IOM handler of the SIUC-X provides a flexible access of the B-channel HDLC controllers to the timeslots on IOM-2 which may be used for IDSL applications.

One of the two B-channel HDCL controllers is programmed to transparent mode and its FIFO is programmed to a certain timeslot on IOM-2, while the second B-channel controller and the D-channel controller is unused (**Figure 70**)

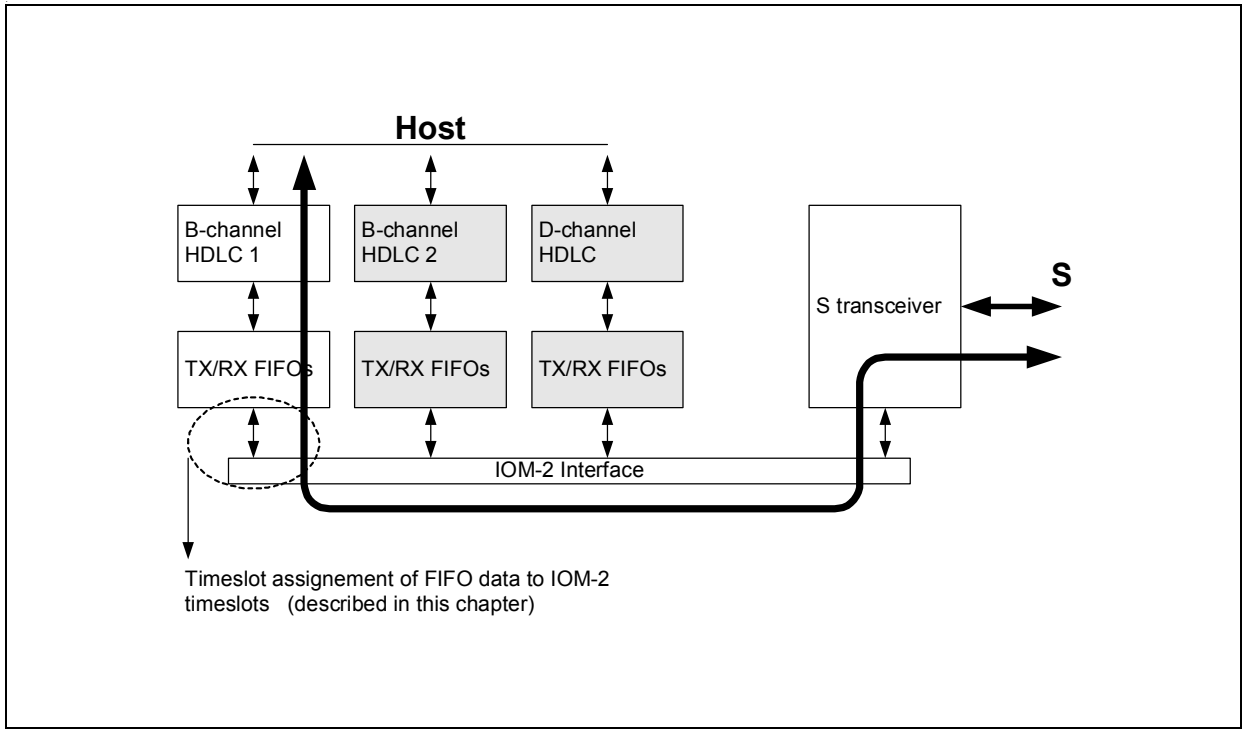


Figure 70 Timeslot Assignment on IOM-2

This B-channel HDLC controller is assigned to three timeslots on IOM-2, which are two 8-bit timeslots and one 2-bit timeslot. For each of the 3 timeslots the timeslot position (timeslot number) and data port (DU, DD) can individually be selected. Additionally, each of the 3 timeslots can individually be enabled/disabled so any combination of the 3 timeslots can be configured, i.e. during each FSC frame the HDLC/FIFO will access 2 bit, 8 bit, 10 bit, 16 bit or 18 bit.

Some examples for access to IOM timeslots are given in **Figure 71**:

- Example 1 shows 18-bit access to B1 + B2 + D
- Example 2 shows 10-bit access to B2 + D
- Example 3 shows 10-bit access to B1 + D in channel 1
- Example 4 shows 16-bit access to MON0 + MON1.

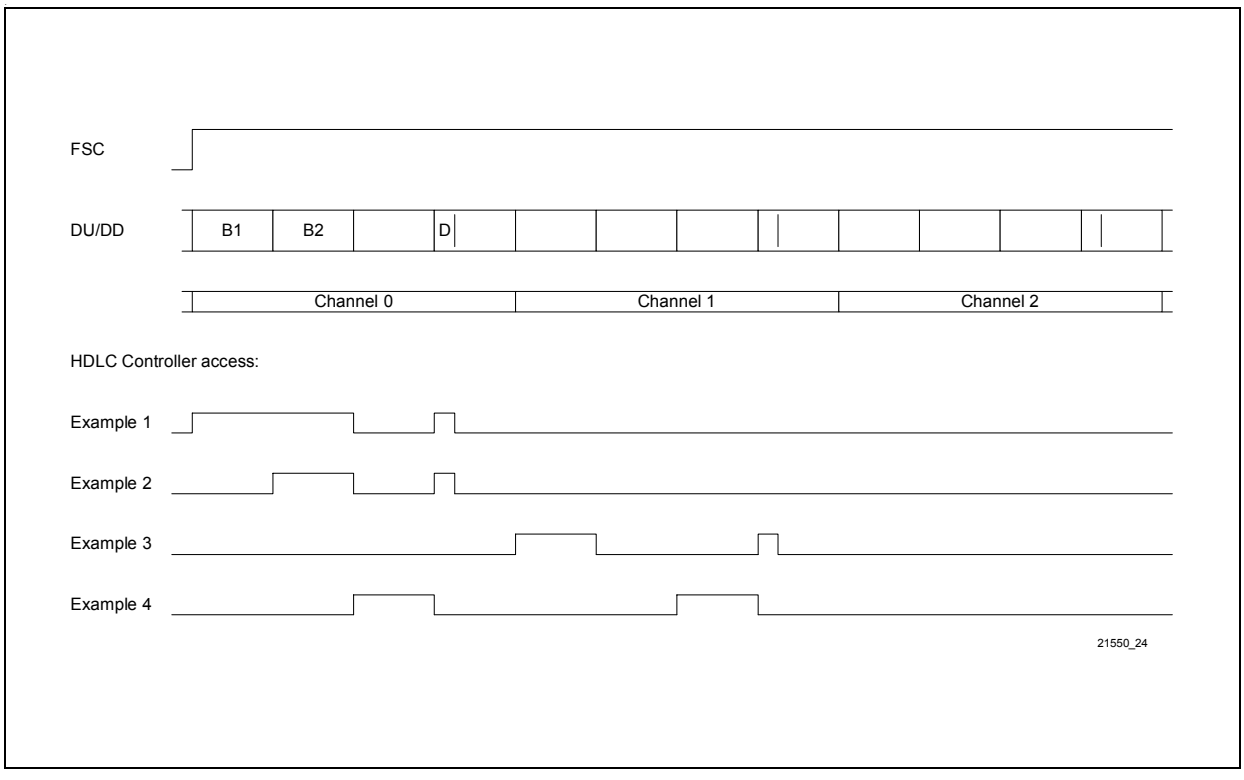


Figure 71 Examples for HDLC Controller Access

The following registers are used to configure one of the two B-channel HDLC controllers (channel A or B) for that (x = A or B):

- BCHx_TSDP_BC1 consists of bits for timeslot selection (TSS) and data port selection (DPS) to program the first 8-bit timeslot.
- BCHx_TSDP_BC2 consists of bits for timeslot selection (TSS) and data port selection (DPS) to program the second 8-bit timeslot.
- BCHx_CR consists of bits for channel selection (CS2-0) and data port selection (DPS_D) to program the 2-bit timeslot. Another 3 bits are used to selectively enable/ disable the first 8-bit timeslot (EN_BC1), the second 8-bit timeslot (EN_BC2) and the 2-bit timeslot (EN_D).

5.5.2.2 S Interface

Data which is read from and written to the IOM-2 interface by the B-channel controller as described in the previous **Chapter 5.5.2.1** is received from and transmitted to the S interface (**Figure 72**).

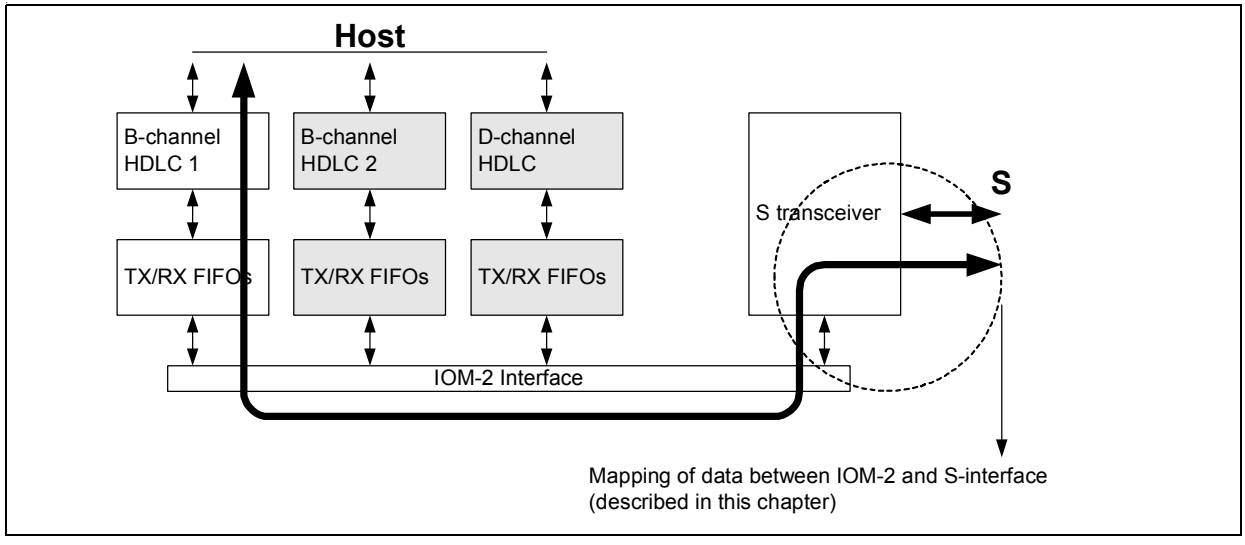


Figure 72 Timeslot Assignment on S

As the timeslot structure of the IOM-2 interface is different from the S interface, it is important to consider the delay and mapping of data between both interfaces

Figure 72 shows the example for bundling 2B+D channels for transmission of 144 kbit/s. Serial data from the FIFO is mapped to the corresponding B- and D-channel timeslots on IOM-2. The ITU I.430 specifies the order and timeslot position of B- and D-channel data on the S-frame. Due to that the order of B- and D-channel data on S is different from IOM-2 which has the effect that mapping of data from IOM-2 to S will change the original order of the serial data stream. However, this has no effect as the remote receiver is using the same mechanism for mapping data between S and IOM-2. In SIUC-BA B- and D-channel bits of one IOM-frame are mapped to the corresponding timeslots of the same S-frame.

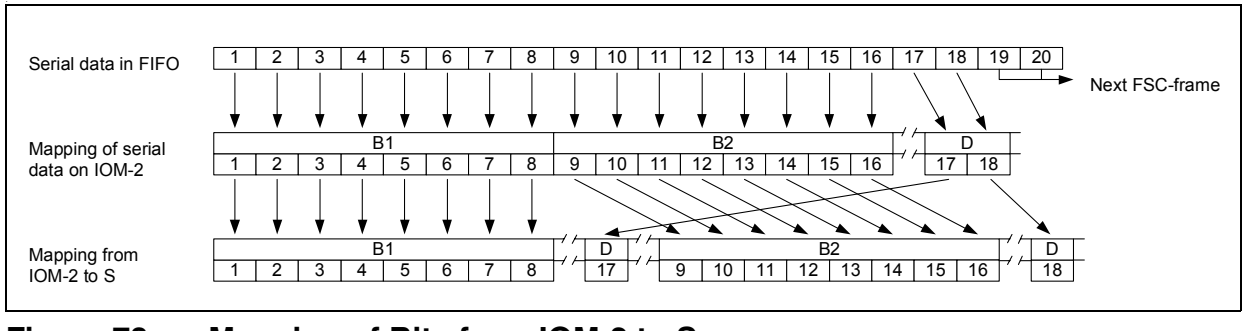


Figure 73 Mapping of Bits from IOM-2 to S

5.5.3 Serial Data Strobe Signal and Strobed Data Clock

For time slot oriented standard devices connected to the IOM-2 interface the SIUC-BA provides an independent data strobe signal SDS which is multiplexed with the reset output signal (SDS/$\overline{\text{RSTO}}$). Instead of a data strobe signal a strobed IOM-2 bit clock can be provided on pin SDS.

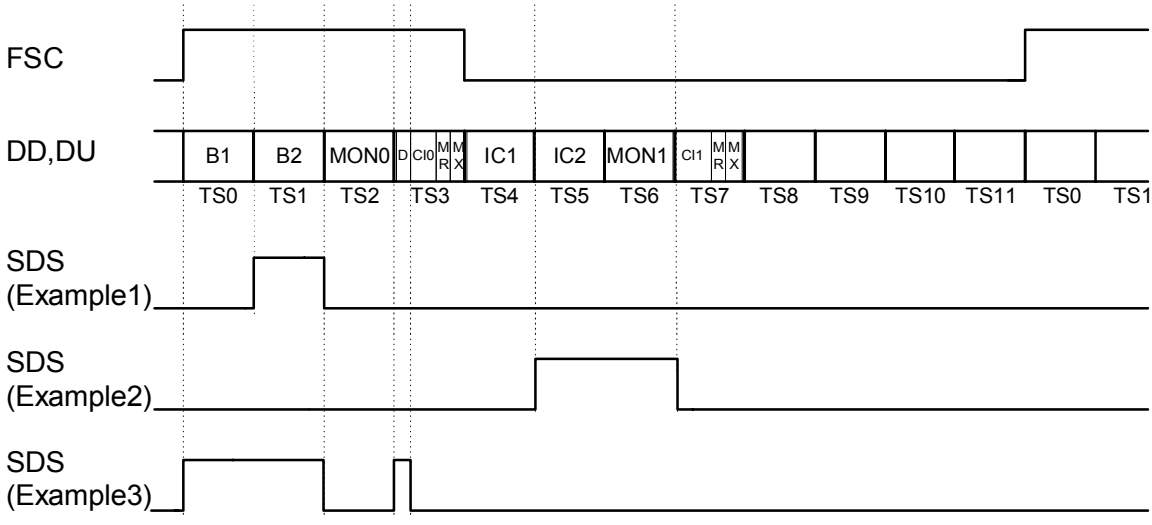
5.5.3.1 Serial Data Strobe Signal

The strobe signal can be generated with every 8-kHz frame and is controlled by the registers SDS_CR. By programming the TSS bits and three enable bits (ENS_TSS, ENS_TSS+1, ENS_TSS+3) a data strobe can be generated for the IOM-2 time slots TS, TS+1 and TS+3 and any combination of them.

The data strobes for TS and TS+1 are always 8 bits long (bit7 to bit0) whereas the data strobe for TS+3 is always 2 bits long (bit7, bit6).

Figure 74 shows three examples for the generation of a strobe signal. In example 1 the SDS is active during channel B2 on IOM-2 whereas in the second example during IC2 and MON1. The third example shows a strobe signal for 2B+D channels which can be used e.g. for an IDSL (144kbit/s) transmission.

The timeslot programming for the SDS signals can be used for another purpose besides SDS signal generation. If enabled (SDS_CONF.DIOM_SDS=1) the DU/DD lines on IOM are high impedant in the selected timeslots (SDS_CONF.DIOM_INV=1) or vice versa (DIOM_INV=0), meaning that only the selected timeslot is active and DU/DD are high impedant outside this timeslot. In this way the IOM-2 interface can be used as real standard PCM interface where data can be transferred in any timeslot.



Example 1: TSS = '0_H'
ENS_TSS = '0'
ENS_TSS+1 = '1'
ENS_TSS+3 = '0'

Example 2: TSS = '5_H'
ENS_TSS = '1'
ENS_TSS+1 = '1'
ENS_TSS+3 = '0'

Example 3: TSS = '0_H'
ENS_TSS = '1'
ENS_TSS+1 = '1'
ENS_TSS+3 = '1'

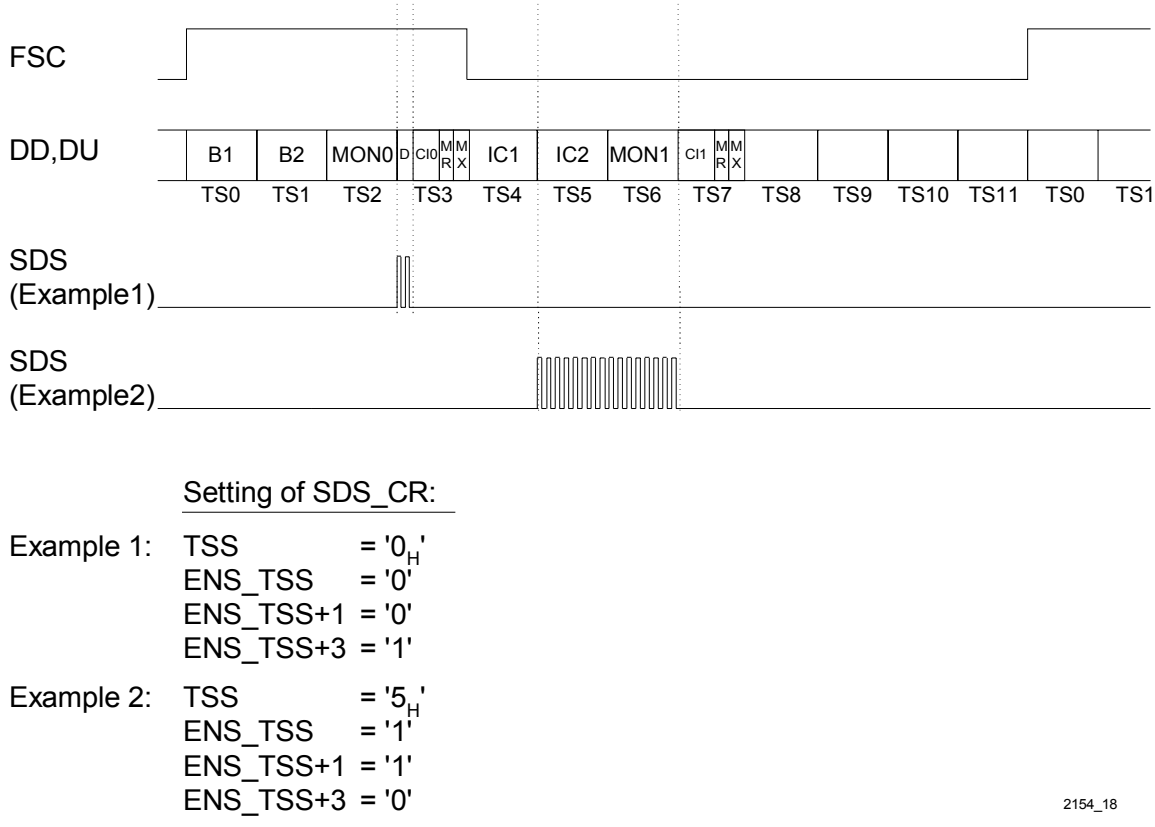
2154_17

For all examples SDS_CONF.SDS_BCL must be set to “0”.

Figure 74 Data Strobe Signal

5.5.3.2 Strobed IOM-2 Bit Clock

The strobed IOM-2 bit clock is active during the programmed window. Outside the programmed window a ’0’ is driven. Two examples are shown in **Figure 75**.



For all examples SDS_CONF.SDS_BCL must be set to “1”.

Figure 75 Strobed IOM-2 Bit Clock. Register SDS_CONF programmed to 01_H

The strobed bit clock can be enabled in SDS_CONF.SDS_BCL.

5.5.4 IOM-2 Monitor Channel

The IOM-2 MONITOR channel (see **Figure 76**) is utilized for information exchange in the MONITOR channel between a master mode device and a slave mode device.

The MONTIOR channel data can be controlled by the bits in the MONITOR control register (MON_CR). For the transmission of the MONITOR data one of the IOM-2 channels (3 IOM-2 channels in TE mode) can be selected by setting the MONITOR channel selection bits (MCS) in the MONITOR control register (MON_CR).

The DPS bit in the same register selects between an output on DU or DD respectively and with EN_MON the MONITOR data can be enabled/disabled. The default value is MONITOR channel 0 (MON0) enabled and transmission on DD.

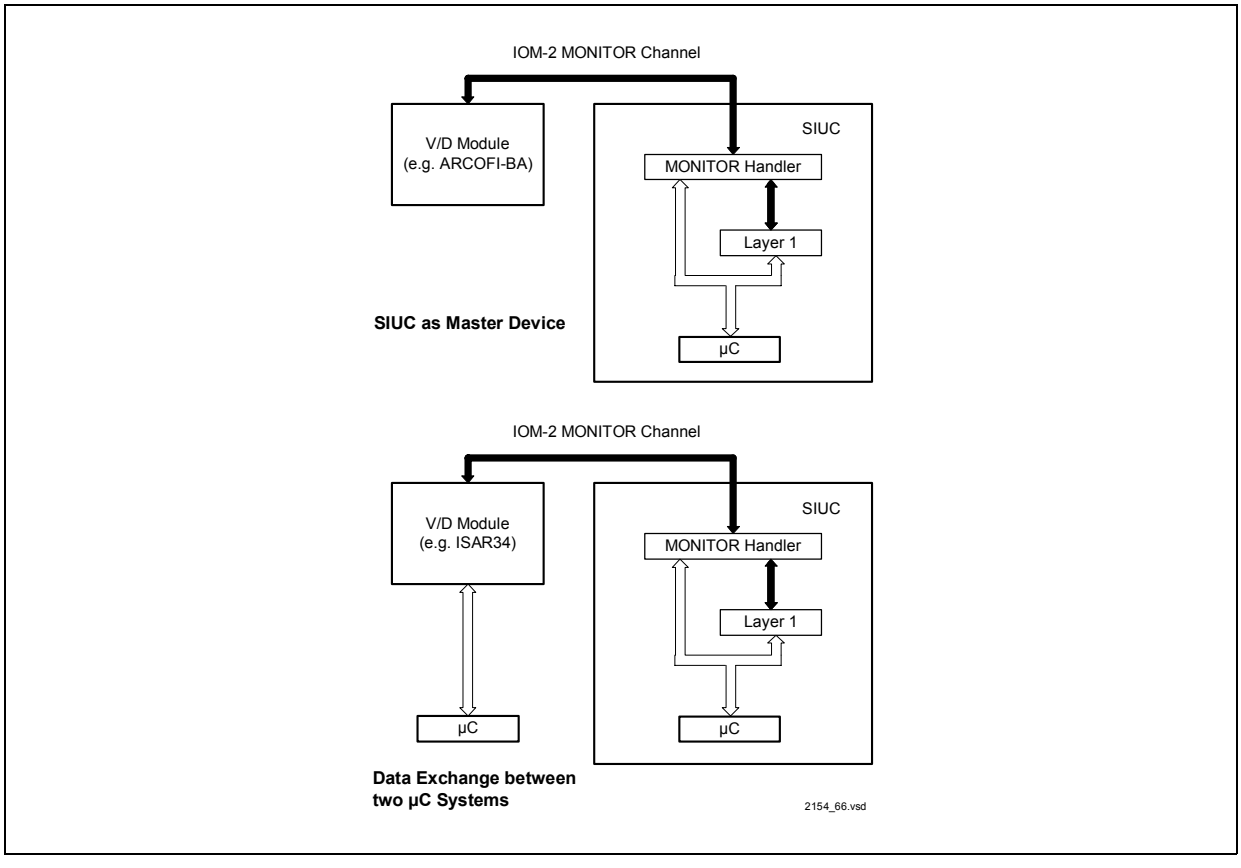


Figure 76 Examples of MONITOR Channel Applications in IOM-2 TE Mode

The MONITOR channel of the SIUC-BA can be used in following applications which are illustrated in **Figure 76**:

- As a **master device** the SIUC-BA can program and control other devices attached to the IOM-2 which do not need a parallel microcontroller interface e.g. ARCOFI-BA PSB 2161. This facilitates redesigning existing terminal designs in which e.g. an interface of an expansion slot is realized with IOM-2 interface and monitor programming.

- For **data exchange** between two microcontroller systems attached to two different devices on one IOM-2 backplane. Use of the MONITOR channel avoids the necessity of a dedicated serial communication path between the two systems. This simplifies the system design of terminal equipment.

5.5.4.1 Handshake Procedure

The MONITOR channel operates on an asynchronous basis. While data transfers on the bus take place synchronized to frame sync, the flow of data is controlled by a handshake procedure using the MONITOR Channel Receive (MR) and MONITOR Channel Transmit (MX) bits. Data is placed onto the MONITOR channel and the MX bit is activated. This data will be transmitted once per 8-kHz frame until the transfer is acknowledged via the MR bit.

The MONITOR channel protocol is described in the following section and **Figure 77** illustrates this. The relevant control and status bits for transmission and reception are listed in **Table 19** and **Table 20**.

Table 19 Transmit Direction

Control/ Status Bit	Register	Bit	Function
Control	MOCR	MXC	MX Bit Control
		MIE	Transmit Interrupt Enable
Status	MOSR	MDA	Data Acknowledged
		MAB	Data Abort
	MSTA	MAC	Transmission Active

Table 20 Receive Direction

Control/ Status Bit	Register	Bit	Function
Control	MOCR	MRC	MR Bit Control
		MRE	Receive Interrupt Enable
Status	MOSR	MDR	Data Received
		MER	End of Reception

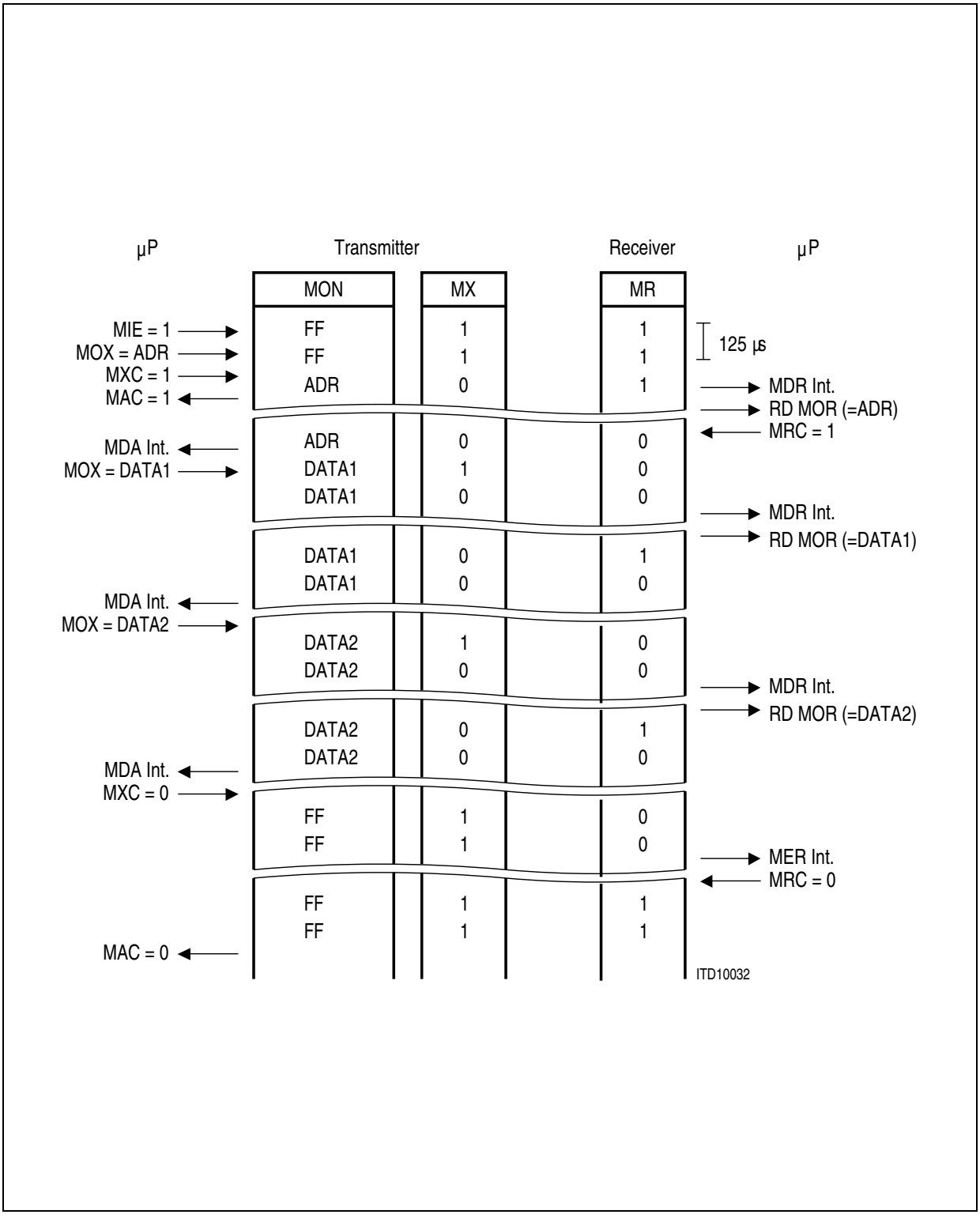


Figure 77 MONITOR Channel Protocol (IOM-2)

Before starting a transmission, the microprocessor should verify that the transmitter is inactive, i.e. that a possible previous transmission has been terminated. This is indicated by a ’0’ in the MONITOR Channel Active MAC status bit.

After having written the MONITOR Data Transmit (MOX) register, the microprocessor sets the MONITOR Transmit Control bit MXC to ’1’. This enables the MX bit to go active (0), indicating the presence of valid MONITOR data (contents of MOX) in the corresponding frame. As a result, the receiving device stores the MONITOR byte in its MONITOR Receive MOR register and generates an MDR interrupt status.

Alerted by the MDR interrupt, the microprocessor reads the MONITOR Receive (MOR) register. When it is ready to accept data (e.g. based on the value in MOR, which in a point-to-multipoint application might be the address of the destination device), it sets the MR control bit MRC to ’1’ to enable the receiver to store succeeding MONITOR channel bytes and acknowledge them according to the MONITOR channel protocol. In addition, it enables other MONITOR channel interrupts by setting MONITOR Interrupt Enable (MIE) to ’1’.

As a result, the first MONITOR byte is acknowledged by the receiving device setting the MR bit to ’0’. This causes a MONITOR Data Acknowledge MDA interrupt status at the transmitter.

A new MONITOR data byte can now be written by the microprocessor in MOX. The MX bit is still in the active (0) state. The transmitter indicates a new byte in the MONITOR channel by returning the MX bit active after sending it once in the inactive state. As a result, the receiver stores the MONITOR byte in MOR and generates a new MDR interrupt status. When the microprocessor has read the MOR register, the receiver acknowledges the data by returning the MR bit active after sending it once in the inactive state. This in turn causes the transmitter to generate an MDA interrupt status.

This "MDA interrupt – write data – MDR interrupt – read data – MDA interrupt" handshake is repeated as long as the transmitter has data to send. Note that the MONITOR channel protocol imposes no maximum reaction times to the microprocessor.

When the last byte has been acknowledged by the receiver (MDA interrupt status), the microprocessor sets the MONITOR Transmit Control bit MXC to ’0’. This enforces an inactive (’1’) state in the MX bit. Two frames of MX inactive signifies the end of a message. Thus, a MONITOR Channel End of Reception MER interrupt status is generated by the receiver when the MX bit is received in the inactive state in two consecutive frames. As a result, the microprocessor sets the MR control bit MRC to 0, which in turn enforces an inactive state in the MR bit. This marks the end of the transmission, making the MONITOR Channel Active MAC bit return to ’0’.

During a transmission process, it is possible for the receiver to ask a transmission to be aborted by sending an inactive MR bit value in two consecutive frames. This is effected by the microprocessor writing the MR control bit MRC to ’0’. An aborted transmission is indicated by a MONITOR Channel Data Abort MAB interrupt status at the transmitter.

The MONITOR transfer protocol rules are summarized in the following section:

- A pair of MX and MR in the inactive state for two or more consecutive frames indicates an **idle state** or an **end of transmission**.
- A **start of a transmission** is initiated by the transmitter by setting the MXC bit to ’1’ enabling the internal MX control. The receiver acknowledges the received first byte by setting the MR control bit to ’1’ enabling the internal MR control.
- The internal MX,MR control indicates or acknowledges a new byte in the MON slot by toggling MX,MR from the active to the inactive state for one frame.
- Two frames with the MR-bit set to inactive indicate a receiver request for **abort**.
- The transmitter can **delay a transmission** sequence by sending the same byte continuously. In that case the MX-bit remains active in the IOM-2 frame following the first byte occurrence. Delaying a transmission sequence is only possible while the receiver MR-bit and the transmitter MX-bit are active.
- Since a **double last-look criterion** is implemented the receiver is able to receive the MON slot data at least twice (in two consecutive frames), the receiver waits for the acknowledge of the reception of two identical bytes in two successive frames.
- To control this handshake procedure a collision detection mechanism is implemented in the transmitter. This is done by making a **collision check** per bit on the transmitted MONITOR data and the MX bit.
- Monitor data will be transmitted repeatedly until its reception is acknowledged or the transmission time-out timer expires.
- Two frames with the MX bit in the inactive state indicates the **end of a message** (EOM).
- Transmission and reception of monitor messages can be performed simultaneously. This feature is used by the SIUC-BA to send back the response before the transmission from the controller is completed (the SIUC-BA does not wait for EOM from controller).

5.5.4.2 Error Treatment

In case the SIUC-BA does not detect identical monitor messages in two successive frames, transmission is not aborted. Instead the SIUC-BA will wait until two identical bytes are received in succession.

A transmission is aborted of the SIUC-BA if

- an error in the MR handshaking occurs
- a collision on the IOM-2 bus of the MONITOR data or MX bit occurs
- the transmission time-out timer expires

A reception is aborted by the device if

- an error in the MX handshaking occurs or
- an abort request from the opposite device occurs

MX/MR Treatment in Error Case

In the master mode the MX/MR bits are under control of the microcontroller through MXC or MRC, respectively. An abort is indicated by an MAB interrupt or MER interrupt, respectively.

In the slave mode the MX/MR bits are under control of the device. An abort is always indicated by setting the MX/MR bit inactive for two or more IOM-2 frames. The controller must react with EOM.

Figure 78 shows an example for an abort requested by the receiver, **Figure 79** shows an example for an abort requested by the transmitter and **Figure 80** shows an example for a successful transmission.

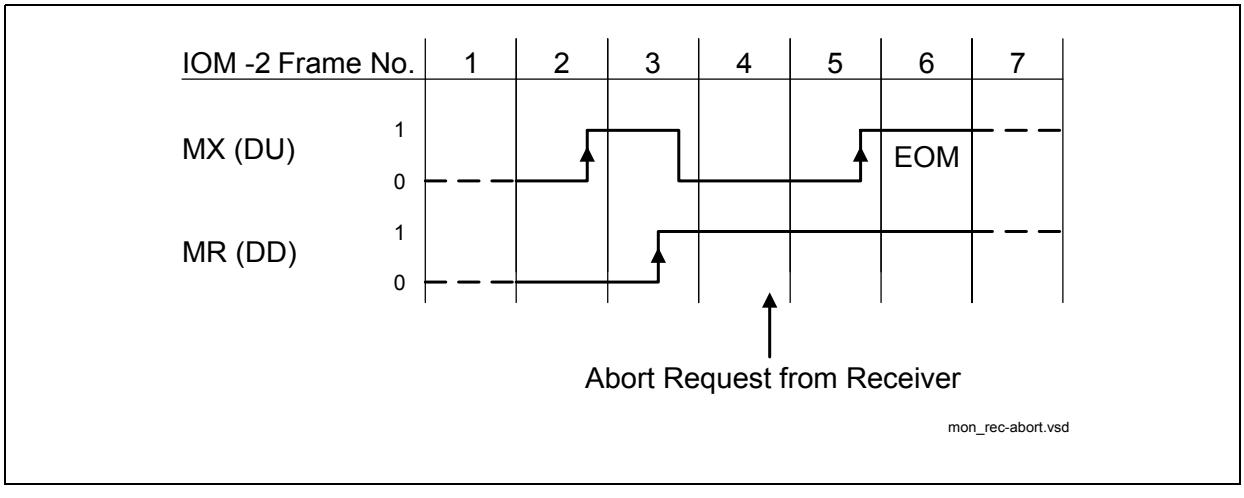


Figure 78 Monitor Channel, Transmission Abort requested by the Receiver

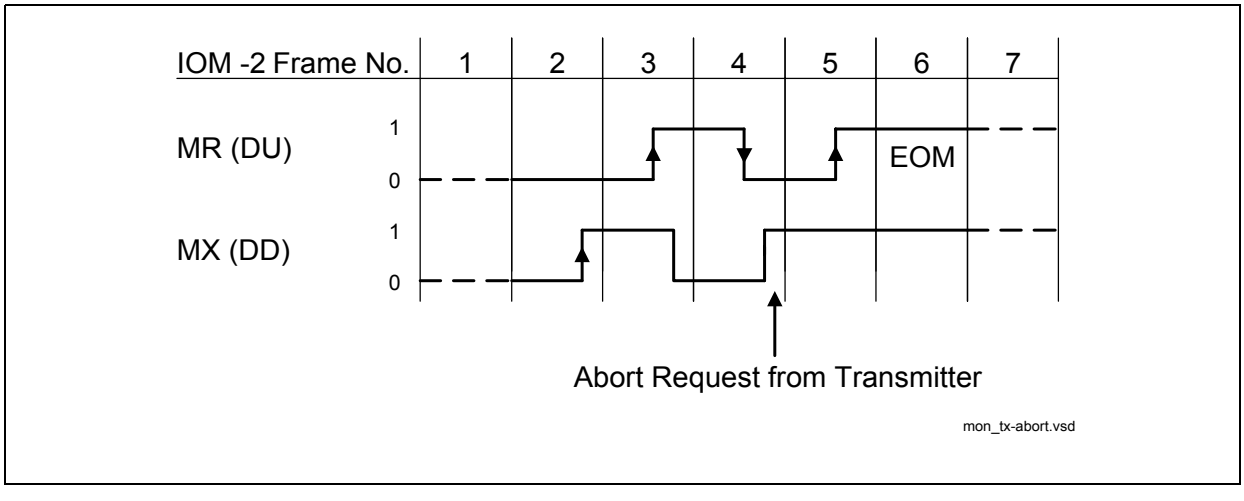


Figure 79 Monitor Channel, Transmission Abort requested by the Transmitter

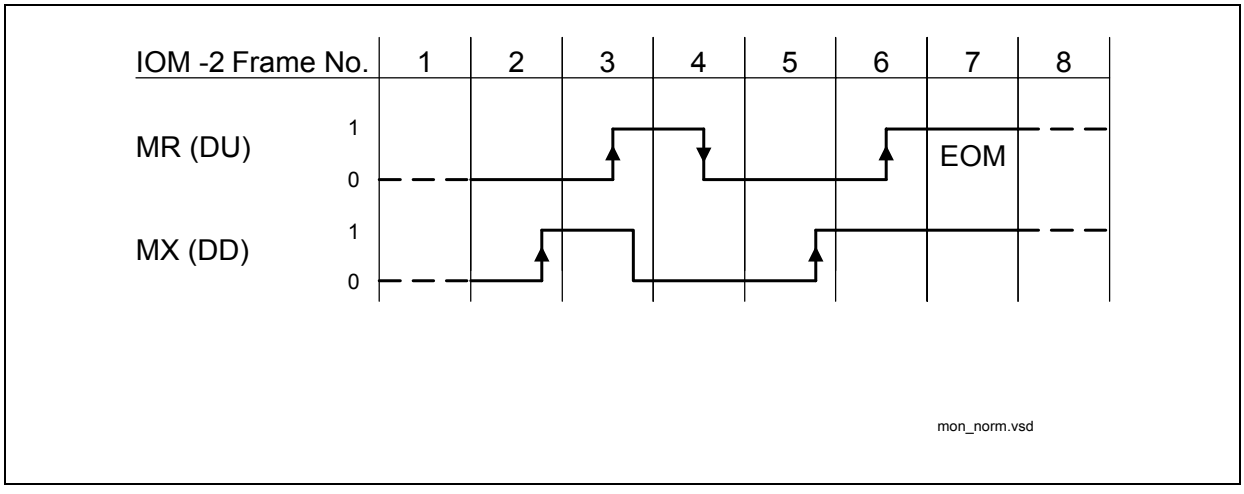


Figure 80 Monitor Channel, Normal End of Transmission

5.5.4.3 MONITOR Channel Programming as a Master Device

As a master device the SIUC-BA can program and control other devices attached to the IOM-2 interface. The master mode is selected by default if one of the possible microcontroller interfaces are selected. The monitor data is written by the microprocessor in the MOX register and transmitted via IOM-2 DD (DU) line to the programmed/controlled device e.g. ARCOFI-BA PSB 2161 or IEC-Q TE PSB 21911. The transfer of the commands in the MON channel is regulated by the handshake protocol mechanism with MX, MR which is described in the previous chapter **Chapter 5.5.4.1**.
If the transmitted command was a read command the slave device responds by sending the requested data.

The data structure of the transmitted monitor message depends on the device which is programmed. Therefore the first byte of the message is a specific address code which contains in the higher nibble a MONITOR channel address to identify different devices. The length of the messages depends on the accessed device and the type of MONITOR command.

The SIUC-BA does not support MONITOR channel Slave Mode.

5.5.4.4 Monitor Time-Out Procedure

To prevent lock-up situations in a MONITOR transmission a time-out procedure can be enabled by setting the time-out bit (TOUT) in the MONITOR configuration register (MCONF). An internal timer is always started when the transmitter must wait for the reply of the addressed device. After 5 ms without reply the timer expires and the transmission will be aborted with a EOM (End of Message) command by setting the MX bit to ’1’ for two consecutive IOM-2 frames.

5.5.4.5 MONITOR Interrupt Logic

Figure 81 shows the MONITOR interrupt structure of the SIUC-BA. The MONITOR Data Receive interrupt status **MDR** has two enable bits, MONITOR Receive interrupt Enable (**MRE**) and MR bit Control (**MRC**). The MONITOR channel End of Reception **MER**, MONITOR channel Data Acknowledged **MDA** and MONITOR channel Data Abort **MAB** interrupt status bits have a common enable bit MONITOR Interrupt Enable **MIE**.

MRE prevents the occurrence of **MDR** status, including when the first byte of a packet is received. When **MRE** is active (1) but **MRC** is inactive, the **MDR** interrupt status is generated only for the first byte of a receive packet. When both **MRE** and **MRC** are active, **MDR** is always generated and all received MONITOR bytes - marked by a 1-to-0 transition in MX bit - are stored. (Additionally, an active **MRC** enables the control of the MR handshake bit according to the MONITOR channel protocol.)

The MONITOR status interrupt in ISTA can be masked in the IEN1 register.

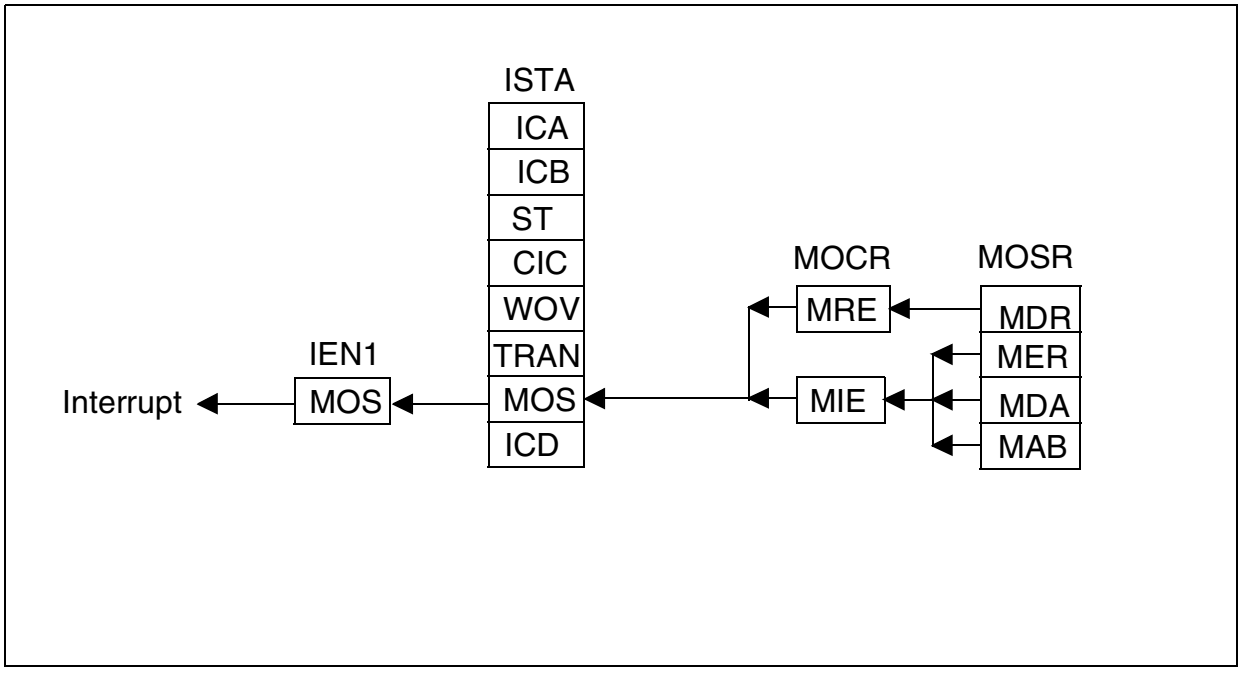


Figure 81 MONITOR Interrupt Structure

5.5.5 C/I Channel Handling

The Command/Indication channel carries real-time status information between the SIUC-BA and another device connected to the IOM-2 interface.

1) One C/I channel (called C/I0) conveys the commands and indications between the layer-1 and the layer-2 parts of the SIUC-BA. It can be accessed by an external layer-2 device e.g. to control the layer-1 activation/deactivation procedures. C/I0 channel access may be arbitrated via the TIC bus access protocol. In this case the arbitration is done in IOM-2 channel 2 (see **Figure 61**).

The C/I0 channel is accessed via register CIR0 (in receive direction, layer-1 to layer-2) and register CIX0 (in transmit direction, layer-2 to layer-1). The C/I0 code is four bits long. A listing and explanation of the layer-1 C/I codes can be found in **Chapter 5.3.2**.
In the receive direction, the code from layer-1 is continuously monitored, with an interrupt being generated anytime a change occurs (ISTA.CIC). A new code must be found in two consecutive IOM-2 frames to be considered valid and to trigger a C/I code change interrupt status (double last look criterion).

In the transmit direction, the code written in CIX0 is continuously transmitted in C/I0.

2) A second C/I channel (called C/I1) can be used to convey real time status information between the SIUC-BA and various non-layer-1 peripheral devices e.g. PSB 2161 ARCOFI-BA. The C/I1 channel consists of four or six bits in each direction.The width can be changed from 4bit to 6bit by setting bit CIX1.CICW.

In 4-bit mode 6-bits are written whereby the higher 2 bits must be set to “1” and 6-bits are read whereby only the 4 LSBs are used for comparison and interrupt generation (i.e. the higher two bits are ignored).

The C/I1 channel is accessed via registers CIR1 and CIX1. A change in the received C/I1 code is indicated by an interrupt status without double last look criterion.

CIC Interrupt Logic

Figure 82 shows the CIC interrupt structure.

A CIC interrupt may originate

– from a change in received C/I channel 0 code (CIC0)

or

– from a change in received C/I channel 1 code (CIC 1).

The two corresponding status bits CIC0 and CIC1 are read in CIR0 register. CIC1 can be individually disabled by clearing the enable bit CI1E in the CIX1 register. In this case the occurrence of a code change in CIR1 will not be displayed by CIC1 until the corresponding enable bit has been set to one.
Bits CIC0 and CIC1 are cleared by a read of CIR0.
An interrupt status is indicated every time a valid new code is loaded in CIR0 or CIR1.

The CIR0 is buffered with a FIFO size of two. If a second code change occurs in the received C/I channel 0 before the first one has been read, immediately after reading of CIR0 a new interrupt will be generated and the new code will be stored in CIR0. If several consecutive codes are detected, only the first and the last code is obtained at the first and second register read, respectively.

For CIR1 no FIFO is available. The actual code of the received C/I channel 1 is always stored in CIR1.

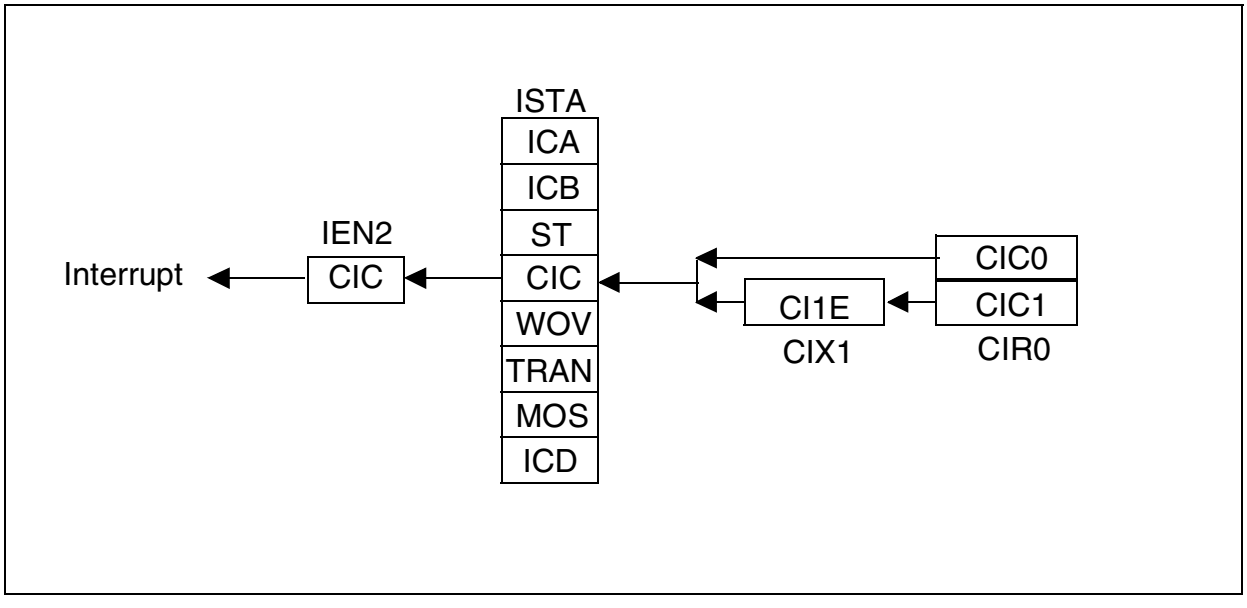


Figure 82 CIC Interrupt Structure

5.5.6 D-Channel Access Control

D-channel access control is defined to guarantee all connected TEs and HDLC controllers a fair chance to transmit data in the D-channel. Collisions are possible

- on the IOM-2 interface if there is more than one HDLC controller connected or
- on the S-interface when there is more than one terminal connected in a point to multipoint configuration (NT → TE1 … TE8).

Both arbitration mechanisms are implemented in the SIUC-BA and will be described in the following two chapters.

5.5.6.1 Stop/Go Bit Handling

The two D-channel access procedures are handled via the Stop/Go bit handling described in this chapter.

The availability of the S/T interface D channel is indicated in bit 5 "Stop/Go" (S/G) of the last octet in DD channel 2 (**Figure 83**).

S/G = 1 : stop
S/G = 0 : go

The Stop/Go bit is available to other layer-2 devices connected to the IOM-2 interface to determine if they can access the S/T bus D channel in upstream direction.

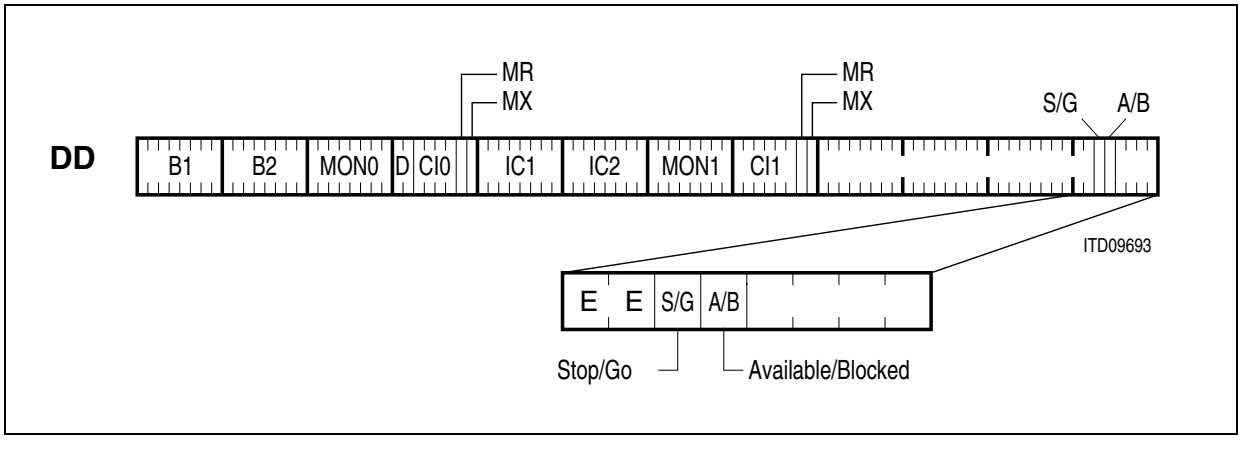


Figure 83 Structure of Last Octet of Ch2 on DD

The S/G bit can also be output on pin AUX7/SGO (Stop/Go bit output is auxiliary function of pin AUX7). In this case the SGO signal changes with the D-bits of the IOM channel. The signal length depends on TR_CONF2.SGD and the polarity is selected via TR_CONF2.SGP.

5.5.6.2 TIC Bus D-Channel Access Control

The TIC bus is imlemented to organize the access to the layer-1 functions provided in the SIUC-BA (C/I-channel) and to the D-channel from up to 7 external communication controllers (see **Figure 84**).
To this effect the outputs of the D-channel controllers (e.g. ICC - ISDN Communication Controller PEB 2070) are wired-or (negative logic, i.e. a “0” wins) and connected to pin DU. The inputs of the ICCs are connected to pin DD. External pull-up resistors on DU/DD are required. The arbitration mechanism must be activated by setting MODED.DIM2-0=00x.

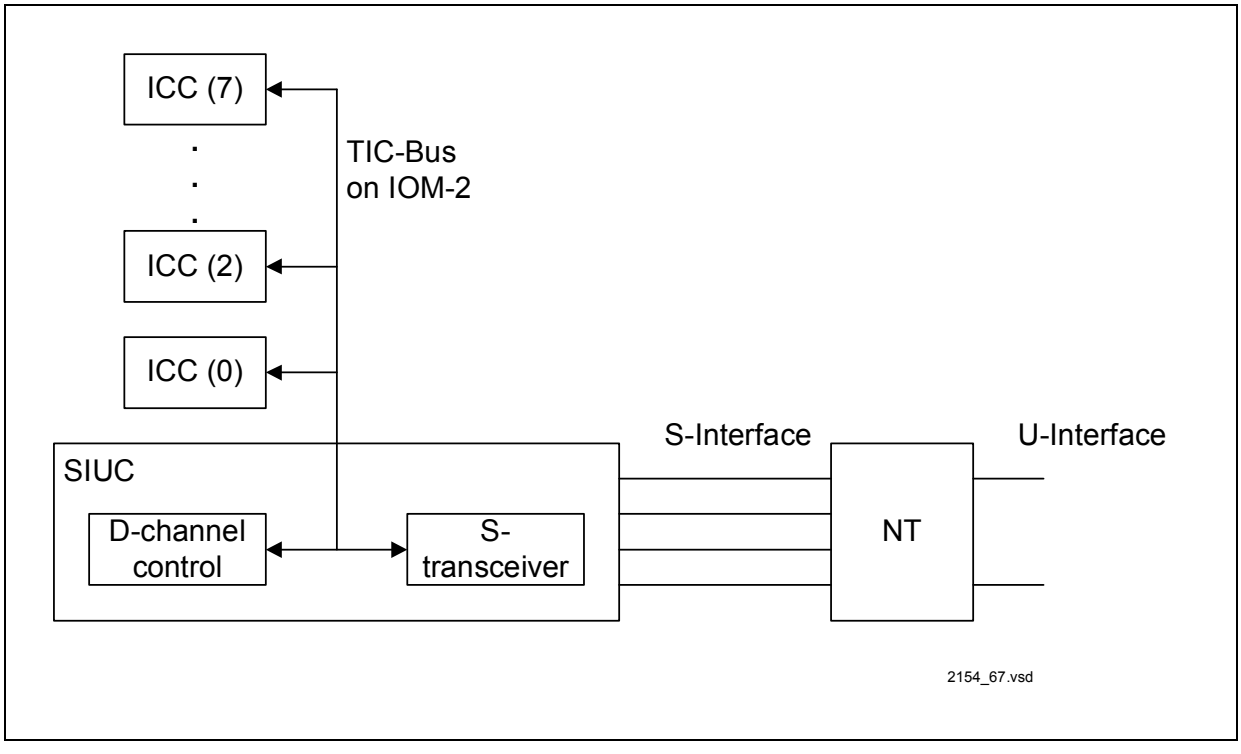


Figure 84 Applications of TIC Bus in IOM-2 Bus Configuration

The arbitration mechanism is implemented in the last octet in IOM-2 channel 2 of the IOM-2 interface (see **Figure 85**). An access request to the TIC bus may either be generated by software (µP access to the C/I channel) or by the SIUC-BA itself (transmission of an HDLC frame in the D-channel). A software access request to the bus is effected by setting the BAC bit (CIX0 register) to ’1’.

In the case of an access request, the SIUC-BA checks the Bus Accessed-bit BAC (bit 5 of last octet of CH2 on DU, see **Figure 85**) for the status "bus free“, which is indicated by a logical ’1’. If the bus is free, the SIUC-BA transmits its individual TIC bus address TAD programmed in the CIX0 register (CIX0.TBA2-0). The SIUC-BA sends its TIC bus address TAD and compares it bit by bit with the value on DU. If a sent bit set to ’1’ is read back as ’0’ because of the access of another D-channel source with a lower TAD, the SIUC-BA withdraws immediately from the TIC bus, i.e. the remaining TAD bits are not

transmitted. The TIC bus is occupied by the device which sends its address error-free. If more than one device attempt to seize the bus simultaneously, the one with the lowest address values wins. This one will set BAC=0 on TIC bus and starts D-channel transmission in the same frame.

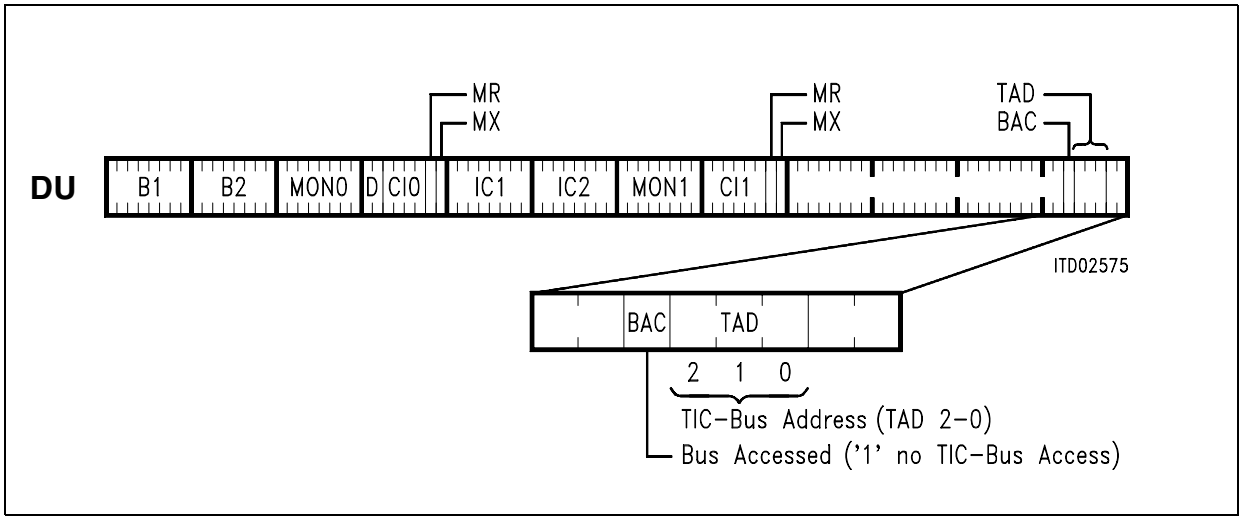


Figure 85 Structure of Last Octet of Ch2 on DU

When the TIC bus is seized by the SIUC-BA, the bus is identified to other devices as occupied via the DU Ch2 Bus Accessed-bit state '0' until the access request is withdrawn. After a successful bus access, the SIUC-BA is automatically set into a lower priority class, that is, a new bus access cannot be performed until the status "bus free" is indicated in two successive frames.

If none of the devices connected to the IOM-2 interface request access to the D and C/I channels, the TIC bus address 7 will be present. The device with this address will therefore have access, by default, to the D and C/I channels.

Note: Bit BAC (CIX0 register) should be reset by the µP when access to the C/I channels is no more requested, to grant other devices access to the D and C/I channels.

5.5.6.3 S-Bus Priority Mechanism for D-Channel

The S-bus access procedure specified in ITU I.430 was defined to organize D-channel access with multiple TEs connected to a single S-bus (see **Figure 86**).

To implement collision detection the D (channel) and E (echo) bits are used. The D-channel S-bus condition is indicated towards the IOM-2 interface with the S/G bit (see above).

The access to the D-channel is controlled by a priority mechanism which ensures that all competing TEs are given a fair access chance. This priority mechanism discriminates among the kind of information exchanged and information exchange history: Layer-2 frames are transmitted in such a way that signalling information is given priority (priority class 1) over all other types of information exchange (priority class 2). Furthermore, once a TE having successfully completed the transmission of a frame, it is assigned a lower level of priority of that class. The TE is given back its normal level within a priority class when all TEs have had an opportunity to transmit information at the normal level of that priority class.

The priority mechanism is based on a rather simple method: A TE not transmitting layer-2 frames sends binary 1s on the D-channel. As layer-2 frames are delimited by flags consisting of the binary pattern “01111110” and zero bit insertion is used to prevent flag imitation, the D-channel may be considered idle if more than seven consecutive 1s are detected on the D-channel. Hence by monitoring the D echo channel, the TE may determine if the D-channel is currently used by another TE or not.

A TE may start transmission of a layer-2 frame first when a certain number of consecutive 1s has been received on the echo channel. This number is fixed to 8 in priority class 1 and to 10 in priority class 2 for the normal level of priority; for the lower level of priority the number is increased by 1 in each priority class, i.e. 9 for class 1 and 11 for class 2.

A TE, when in the active condition, is monitoring the D-echo channel (E-bits), counting the number of consecutive binary 1s. If a 0 bit is detected, the TE restarts counting the number of consecutive binary 1s. If the required number of 1s according to the actual level of priority has been detected, the TE may start transmission of an HDLC frame. If a collision occurs, the TE immediately shall cease transmission, return to the D-channel monitoring state, and send 1s over the D-channel.

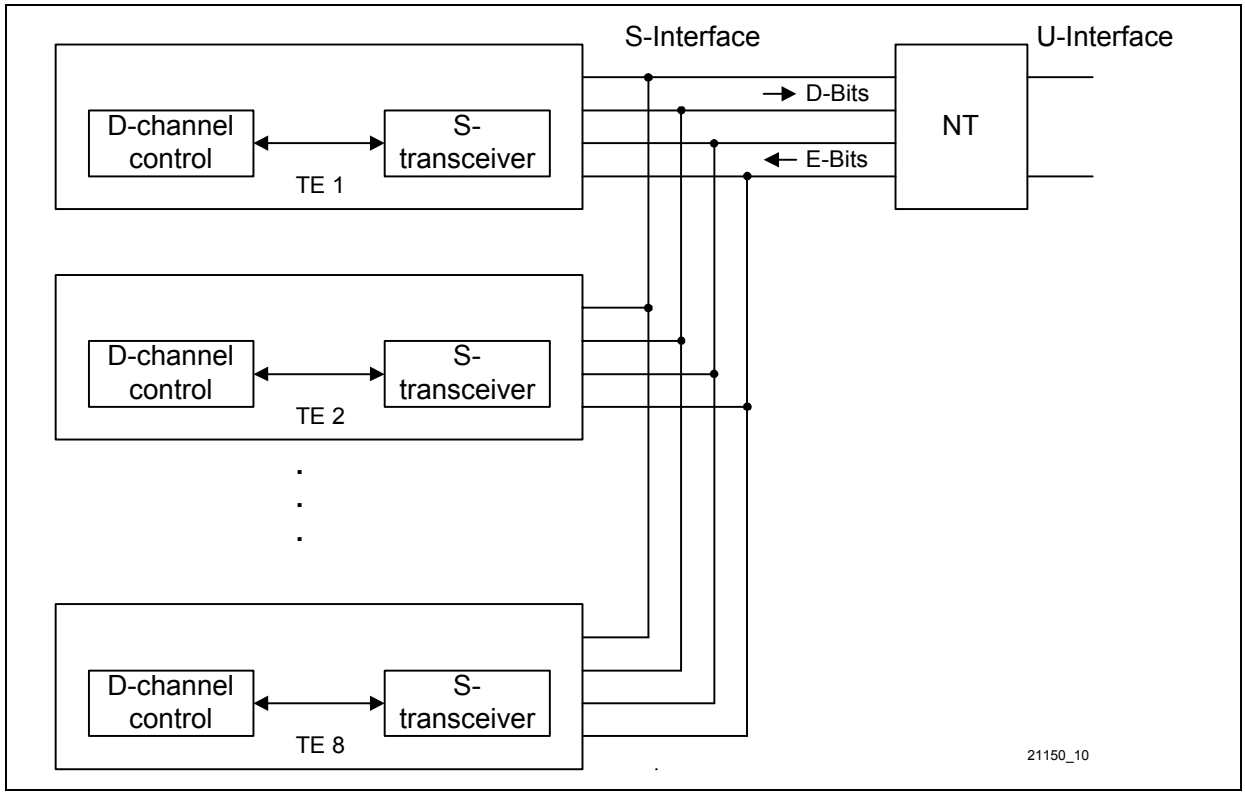


Figure 86 D-Channel Access Control on the S-Interface

The above described priority mechanism is fully implemented in the transceiver. For this purpose the D-channel collission detection according to ITU I.430 must be enabled by setting MODED.DIM2-0 to ’0x1’. In this case the transceiver continuously compares the received E-echo bits with its own transmitted D data bits.

Depending on the priority class selected, 8 or 10 consecutive ONEs (high priority level) need to be detected before the transceiver sends valid D-channel data on the upstream D-bits on S. In low priority level 9 or 11 consecutive ONEs are required.

The transceiver controls the S/G bit on IOM-2 in a way that internal delays in the transceiver path are compensated, i.e. the S/G is set to 0 (“go”) before the required number of ONEs is counted. Due to that reason D-channel bits in the transceiver transmit path must be discarded and the S/G bit must be set to “1” if a collision on S is detected.

The priority class (priority 8 or priority 10) is selected by transferring the appropriate activation command via the Command/Indication (C/I) channel of the IOM-2 interface to the transceiver. If the activation is initiated by a TE, the priority class is selected implicitly by the choice of the activation command. If the S-interface is activated from the NT, an activation command selecting the desired priority class should be programmed at the TE on reception of the activation indication (AI8 or AI10). In the activated state the priority class may be changed whenever required by simply programming the desired activation request command (AR8 or AR10).

5.5.6.4 State Machine of the D-Channel Arbiter

Figure 87 gives a simplified view of the state machine of the D-channel arbiter. CNT is the number of ’1’ on the IOM-2 D-channel and BAC corresponds to the BAC-bit on IOM-2. The number n depends on configuration settings (selected priority 8 or 10) and the condition of the previous transmission, i.e. if an abort was seen (n = 8 or 10, respectively) or if the last transmission was successful (n = 9 or 11, respectively).

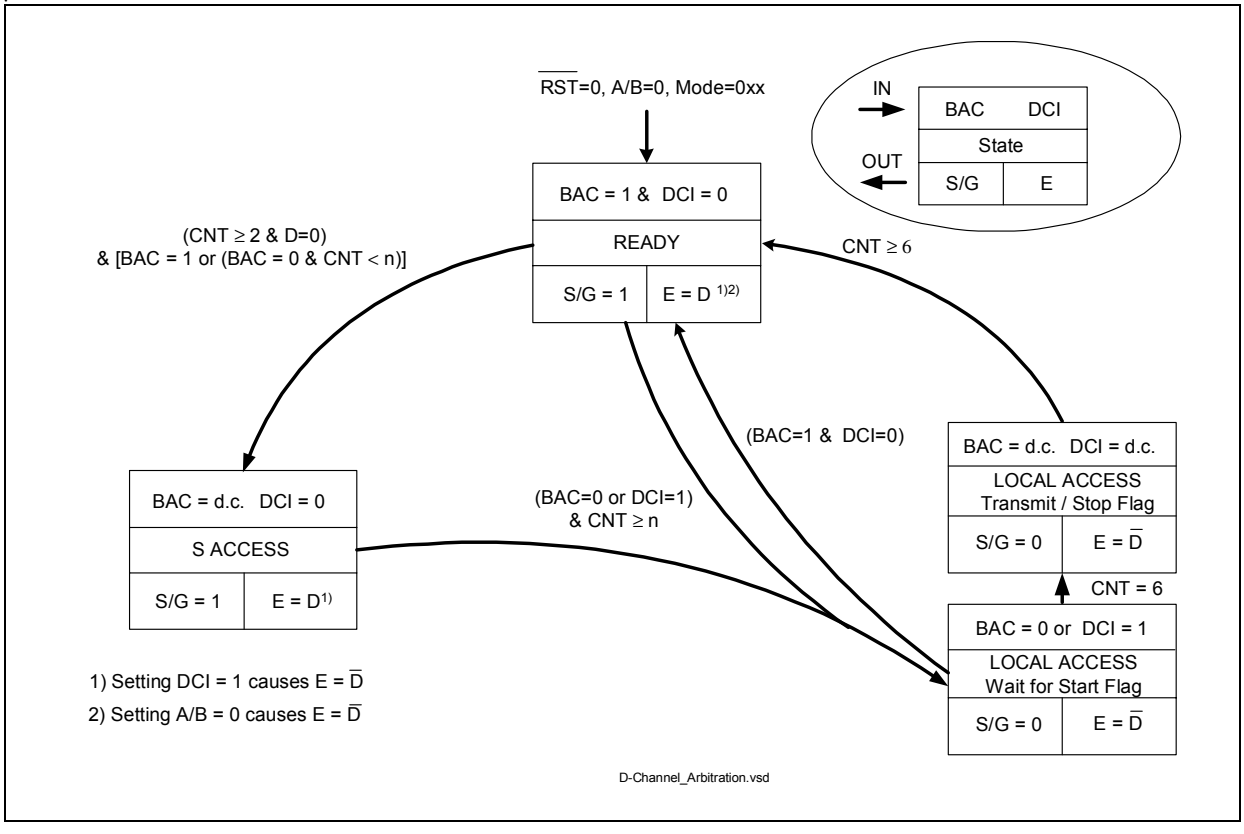


Figure 87 State Machine of the D-Channel Arbiter (Simplified View)

Note: The figure above provides a simplified view only. If the S-transceiver is reset by SRES.RES_S = ’1’ or disabled by TR_CONF0.DIS_TR = ’1’, then the D-channel arbiter is in state Ready (S/G = ’1’), too. The S/G evaluation of the HDLC controller has to be disabled in this case; otherwise, the HDLC is not able to send data.

1. Local D-Channel Controller Transmits Upstream

In the initial state (’Ready’ state) neither the local D-channel sources nor any of the terminals connected to the S-bus transmit in the D-channel.

The S-transceiver thus receives BAC = “1” (IOM-2 DU line) and transmits S/G = “1” (IOM-2 DD line). The access will then be established according to the following procedure:

- Local D-channel source verifies that BAC bit is set to ONE (currently no bus access).
- Local D-channel source issues TIC bus address and verifies that no controller with higher priority requests transmission (TIC bus access must always be performed even if no other D-channel sources are connected to IOM-2).
- Local D-channel source issues BAC = “0” to block other sources on IOM-2 and to announce D-channel access.
- S-transceiver pulls S/G bit to ZERO (’Local Access’ state) as soon as CNT ≥ n (see note) to allow for further D-channel access.
- S-transceiver transmits inverted echo channel (E bits) on the S-bus to block all connected S-bus terminals (E = $\overline{D}$).
  Blocking the S-bus by inverting the D-bits in the Echo channel (E = $\overline{D}$) can be enforced by the host via TR_MODE.DCH_INH = 1 and/or any project specific pin signal like DCI.
- Local D-channel source commences with D data transmission on IOM-2 as long as it receives S/G = “0”.
- After D-channel data transmission is completed the controller sets the BAC bit to ONE.
- S-transceiver transmits non-inverted echo (E = D).
- S-transceiver pulls S/G bit to ONE (’Ready’ state) to block the D-channel controller on IOM-2.

Note: If right after D-data transmission the D-channel arbiter goes to state ’Ready’ and the local D-channel source wants to transmit again, then it may happen that the leading ’0’ of the start flag is written into the D-channel before the D-channel source recognizes that the S/G bit is pulled to ’1’ and stops transmission. In order to prevent unintended transitions to state ’S-Access’, the additional condition CNT ≥ 2 is introduced. As soon as CNT ≥ n, the S/G bit is set to ’0’ and the D-channel source may start transmission again (if TIC bus is occupied). This allows an equal access for D-channel sources on IOM-2 and on the S interface.

2. Terminal Transmits D-Channel Data Upstream

The initial state is identical to that described in the last paragraph. When one of the connected S-bus terminals needs to transmit in the D-channel, access is established according to the following procedure:

- S-transceiver recognizes that the D-channel on the S-bus is active via D = ’0’.
- S-transceiver transfers S-bus D-channel data transparently through to the upstream IOM-2 bus.

5.5.7 Activation/Deactivation of IOM-2 Interface

The IOM-2 interface can be switched off in the inactive state, reducing power consumption to a minimum. In this deactivated state is FSC = ’1’, DCL and BCL = ’0’ and the data lines are ’1’.

The IOM-2 interface can be kept active while the S interface is deactivated by setting the CFS bit to "0" (MODE1 register). This is the case after a hardware reset. If the IOM-2 interface should be switched off while the S interface is deactivated, the CFS bit should be set to ’1’. In this case the internal oscillator is disabled when no signal (info 0) is present on the S bus and the C/I command is ’1111’ = DIU. If the TE wants to activate the line, it has first to activate the IOM-2 interface either by using the "Software Power Up" function (IOM_CR.SPU bit) or by setting the CFS bit to "0" again.

The deactivation procedure is shown in **Figure 88**. After detecting the code DIU (Deactivate Indication Upstream) the layer 1 of the SIUC-BA responds by transmitting DID (Deactivate Indication Downstream) during subsequent frames and stops the timing signals synchronously with the end of the last C/I (C/I0) channel bit of the fourth frame.

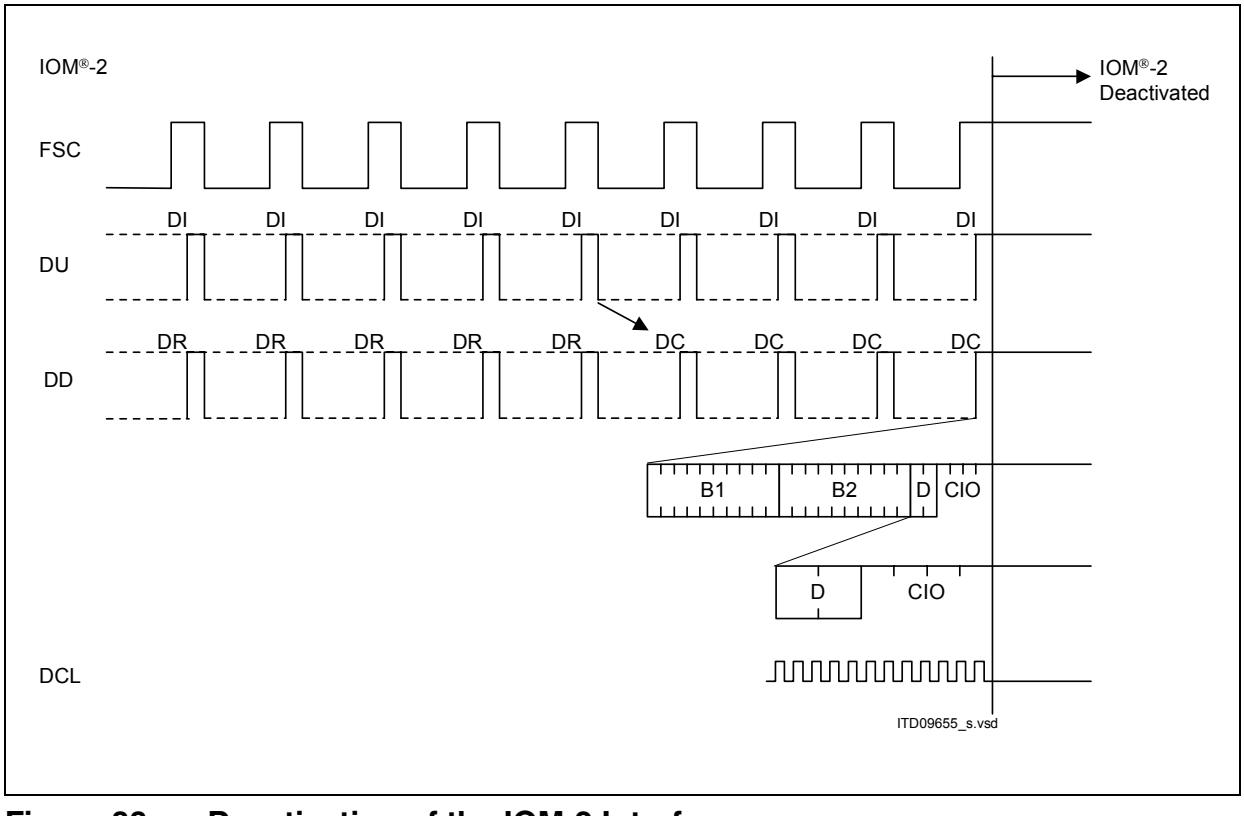


Figure 88 Deactivation of the IOM-2 Interface

The clock pulses will be enabled again when the DU line is pulled low (bit SPU in the IOM_CR register), i.e. the C/I command TIM = "0000" is received by layer 1, or when a non-zero level on the S-line interface is detected (if TR_CONF0.LDD=0). The clocks are turned on after approximately 0.2 to 4 ms depending on the oscillator.

DCL is activated such that its first rising edge occurs with the beginning of the bit following the C/I (C/I0) channel.

After the clocks have been enabled this is indicated by the PU code in the C/I channel and, consequently, by a CIC interrupt. The DU line may be released by resetting the Software Power Up bit IOM_CR ='0' and the C/I code written to CIX0 before (e.g. TIM or AR8) is output on DU.

The SIUC-BA supplies IOM-2 timing signals as long as there is no DIU command in the C/I (C/I0) channel. If timing signals are no longer required and activation is not yet requested, this is indicated by programming DIU in the CIX0 register.

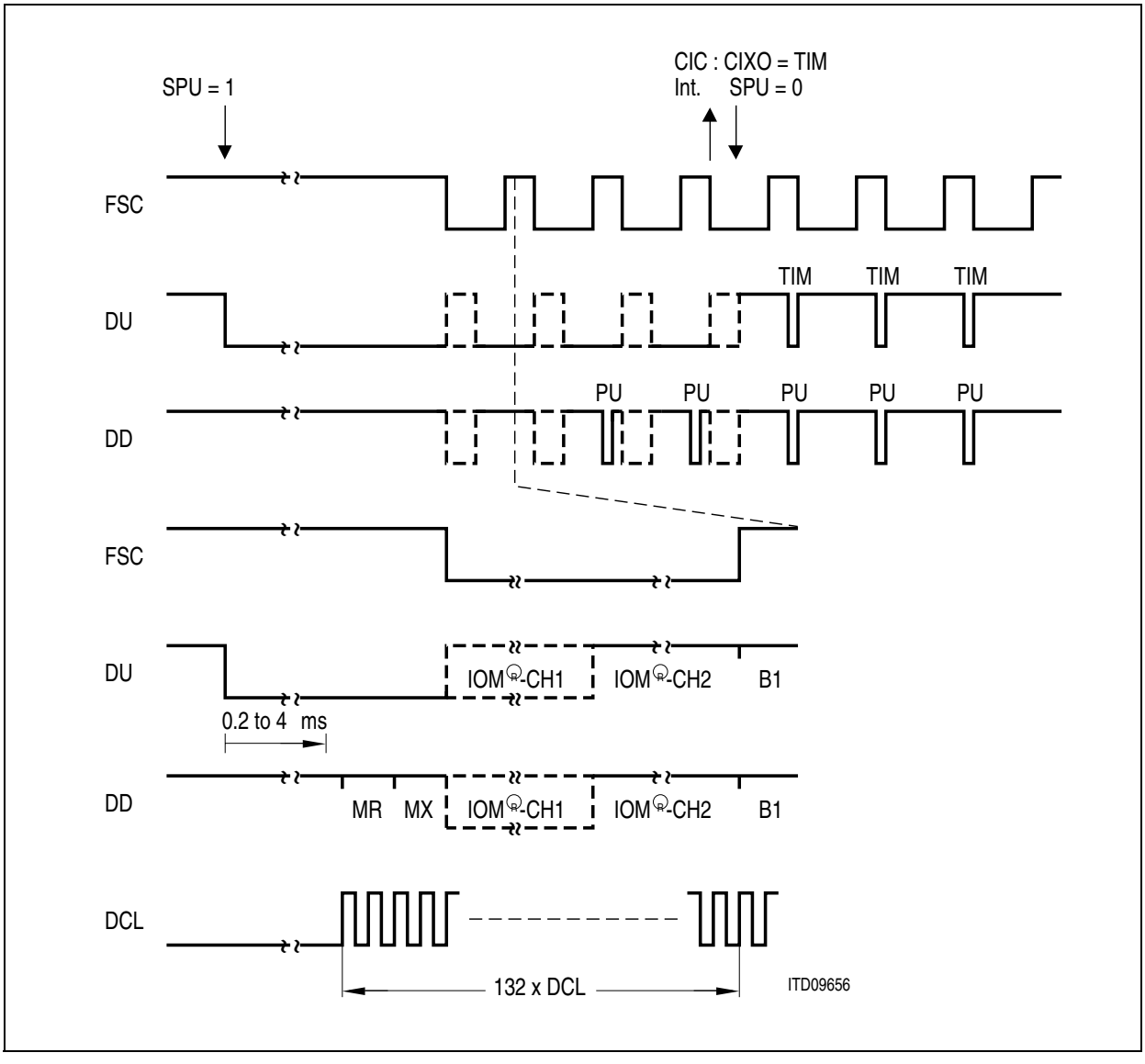


Figure 89 Activation of the IOM-2 interface

5.6 HDLC Controllers

The SIUC-BA contains three HDLC controllers which can arbitrarily be used for the layer-2 functions of the D- channel protocol (LAPD) and B-channel protocols. By setting the Enable HDLC channel bits (EN_D, EN_B1H, EN_B2H) in the DCI_CR/BCH_CR registers each of the HDLC controllers can access the D or B-channels or any combination of them e.g. 18 bit IDSL data (2B+D).

They perform the framing functions used in HDLC based communication: flag generation/recognition, bit stuffing, CRC check and address recognition.

The D-channel FIFO has a size of 64 byte per direction. Each of the two B-channel FIFOs has a size of 128 bytes per direction. They are implemented as cyclic buffers. The transceiver reads and writes data sequentially with constant data rate whereas the data transfer between FIFO and microcontroller uses a block oriented protocol with variable block sizes.

The configuration, control and status bits related to the HDLC controllers are all assigned to the following address ranges:

Table 21 HDLC Controller Address Range

	FIFO Address	**Config/Ctrl/Status Registers**
D-channel	00_H-$1F_H$	20_H-29_H
B-channel A	$7A_H$	70_H-79_H
B-channel B	$8A_H$	80_H-89_H

Note: For B-channel data access a single address location is used to read from and write to the FIFO. For D-channel access the address range 00_H-$1F_H$ is used (similar as in ISAC-S PEB 2086), however a single address from this range is sufficient to access the FIFO as the internal FIFO pointer is incremented automatically independent from the external address.

The mechanisms for access to the FIFOs are identical for D- and B-channels, therefore the following description applies to both of them and for simplification specific references like registers are indicated by an “x” (stands for “D” and “B”) to indicate it is relevant for D- and B-channel (e.g. ISTAx means ISTAD/ISTAB).

5.6.1 Message Transfer Modes

The HDLC controllers can be programmed to operate in various modes, which are different in the treatment of the HDLC frame in receive direction. Thus the receive data flow and the address recognition features can be programmed in a flexible way to satisfy different system requirements.

The structure of a D-channel two-byte address (LAPD) is shown below:

High Address Byte			Low Address Byte	
SAPI1, 2, SAPG	C/R	0	TEI 1, 2, TEIG	EA

For address recognition on the D-channel the SIUC-BA contains four programmable registers for individual SAPI and TEI values (SAP1, 2 and TEI1, 2), plus two fixed values for the “group” SAPI (SAPG = ’FE’ or ’FC’) and TEI (TEIG = ’FF’).
The received C/R bit is excluded from the address comparison. EA is the address field extension bit which must be set to ’1’ according to HDLC LAPD.

The structure of a B-channel two-byte address is as follows:

High Address Byte			Low Address Byte
RAH1, 2, Group Address	C/R	0	RAL1, 2, Group Address

For address recognition on the B-channel the SIUC-BA contains four programmable registers for individual Receive Address High and Low values (RAH1, 2 and RAL1, 2), plus two fixed values for the High Address Byte (Group Address = ’FE’ or ’FC’) and one fixed value for the Low Address Byte (Group Address = ’FF’).
The received C/R bit is excluded from the address comparison. EA is the address field extension bit which must be set to ’1’ according to HDLC LAPD.

Operating Modes

There are 5 different operating modes which can be selected via the mode selection bits MDS2-0 in the MODEx registers:

Non-Auto Mode (MDS2-0 = ’01x’)

Characteristics: Full address recognition with one-byte (MDS = ’010’) or two-byte (MDS = ’011’) address comparison

All frames with valid addresses are accepted and the bytes following the address are transferred to the µP via RFIFOx. Additional information is available in RSTAx.

Transparent mode 0 (MDS2-0 = ’110’).

Characteristics: no address recognition

Every received frame is stored in RFIFOx (first byte after opening flag to CRC field). Additional information can be read from RSTAx.

Transparent mode 1 (MDS2-0 = ’111’).

Characteristics: SAPI recognition (D-channel)
High byte address recognition (B-channel)

A comparison is performed on the first byte after the opening flag with SAP1, SAP2 and “group” SAPI (FE_H/FC_H) for D-channel, and with RAH1, RAH2 and group address (FE_H/FC_H) for B-channel. In the case of a match, all the following bytes are stored in RFIFOx. Additional information can be read from RSTAx.

Transparent mode 2 (MDS2-0 = ’101’).

Characteristics: TEI recognition (D-channel)
Low byte address recognistion (B-channel)

A comparison is performed only on the second byte after the opening flag, with TEI1, TEI2 and group TEI (FF_H) for D-channel, and with RAL1 and RAL2 for B-channel. In case of a match the rest of the frame is stored in the RFIFOx. Additional information is available in RSTAx.

Extended transparent mode (MDS2-0 = ’100’).

Characteristics: fully transparent

In extended transparent mode fully transparent data transmission/reception without HDLC framing is performed i.e. without FLAG generation/recognition, CRC generation/check, bitstuffing mechanism. This allows user specific protocol variations.
Also refer to **Chapter 5.6.5**.

5.6.2 Data Reception

5.6.2.1 Structure and Control of the Receive FIFO

The cyclic receive FIFO buffers with a length of 64-byte for D-channel and 128 byte for each of the two B-channels have variable FIFO block sizes (thresholds) of

- 4, 8, 16 or 32 bytes for D-channel and
- 8, 16, 32 or 64 bytes for B-channels

which can be selected by setting the corresponding RFBS bits in the EXMx registers. The variable block size allows an optimized HDLC processing concerning frame length, I/O throughput and interrupt load.
The transfer protocol between HDLC FIFO and microcontroller is block oriented with the microcontroller as master. The control of the data transfer between the CPU and the SIUC-BA is handled via interrupts (SIUC-BA → Host) and commands (Host → SIUC-BA).

There are three different interrupt indications in the ISTAx registers concerned with the reception of data:

– **RPF** (**R**eceive **P**ool **F**ull) interrupt, indicating that a data block of the selected length (EXMx.RFBS) can be read from RFIFOx. The message which is currently received exceeds the block size so further blocks will be received to complete the message.
– **RME** (**R**eceive **M**essage **E**nd) interrupt, indicating that the reception of one message is completed, i.e. either
  - a short message is received

  (message length ≤ the defined block size (EXMx.RFBS)) or
  - the last part of a long message is received

  (message length > the defined block size (EXMx.RFBS))
  and is stored in the RFIFOx.
– **RFO** (**R**eceive **F**rame **O**verflow) interrupt, indicating that a complete frame could not be stored in RFIFOx and is therefore lost as the RFIFOx is occupied. This occurs if the host fails to respond quickly enough to RPF/RME interrupts since previous data was not read by the host.

There are two control commands that are used with the reception of data:

– **RMC** (**R**eceive **M**essage **C**omplete) command, telling the SIUC-BA that a data block has been read from the RFIFOx and the corresponding FIFO space can be released for new receive data.
– **RRES** (**R**eceiver **R**eset) command, resetting the HDLC receiver and clearing the receive FIFO of any data (e.g. used before start of reception). It has to be used after a change of the message transfer mode. Pending interrupt indications of the receiver are not cleared by RRES, but have to be cleared by reading these interrupts.

Note: The significant interrupts and commands are underlined as only these are commonly used during a normal reception sequence.

The following description of the receive FIFO operation is illustrated in **Figure 90** for a RFIFOx block size (threshold) of 16 and 32 bytes.

The RFIFOx requests service from the microcontroller by setting a bit in the ISTAx register, which causes an interrupt (RPF, RME, RFO). The microcontroller then reads status information (RBCHx,RBCLx), data from the RFIFOx and then may change the receive FIFO block size (EXMx.RFBS). A block transfer is completed by the microcontroller via a receive message complete (CMDRx.RMC) command. This causes the space of the transferred bytes being released for new data and in case the frame was complete (RME) the reset of the receive byte counter RBC (RBCHx,RBCLx).
The total length of the frame is contained in the RBCHx and RBCLx registers which contain a 12 bit number (RBC11...0), so frames up to 4095 byte length can be counted. If a frame is longer than 4095 bytes, the RBCH.OV (overflow) bit will be set. The least significant bits of RBCLx contain the number of valid bytes in the last data block indicated by RMEx (length of last data block ≤ selected block size). **Table 22** shows which RBC bits contain the number of bytes in the last data block or number of complete data blocks respectively. If the number of bytes in the last data block is ’0’ the length of the last received block is equal to the block size.

Table 22 Receive Byte Count with RBC11...0 in the RBCHx/RBCLx registers

EXMD1.RFBS bits (D-channel)	EXMB.RFBS bits (B-channel)	Selected block size	Number of	
			complete data blocks in	bytes in the last data block in
--	’00’	64 byte	RBC11...6	RBC5...0
’00’	’01’	32 byte	RBC11...5	RBC4...0
’01’	’10’	16 byte	RBC11...4	RBC3...0
’10’	’11’	8 byte	RBC11...3	RBC2...0
’11’	--	4 byte	RBC11...2	RBC1...0

The transfer block size (EXMx.RFBS) is 32 bytes for D-channel and 64 bytes for B-channel by default. If it is necessary to react to an incoming frame within the first few bytes the microcontroller can set the RFIFOx block size to a smaller value. Each time a CMDRx.RMC or CMDRx.RRES command is issued, the RFIFOx access controller sets its block size to the value specified in EXMR.RFBS, so the microcontroller has to write the new value for RFBS before the RMC command. When setting an initial value for RFBS before the first HDLC activities, a RRES command must be issued afterwards.

The RFIFOx can hold any number of frames fitting in the 64 bytes (D-channel)/128 bytes (B-channel) independent of RFBS (but the RFIFO is read blockwise according to the selected threshold). At the end of a frame, the RSTAx byte is always appended.

All generated interrupts are inserted together with all additional information into a wait line to be individually passed to the host. For example if several data blocks have been received to be read by the host and the host acknowledges the current block, a new RPF or RME interrupt from the wait line is immediately generated to indicate new data.

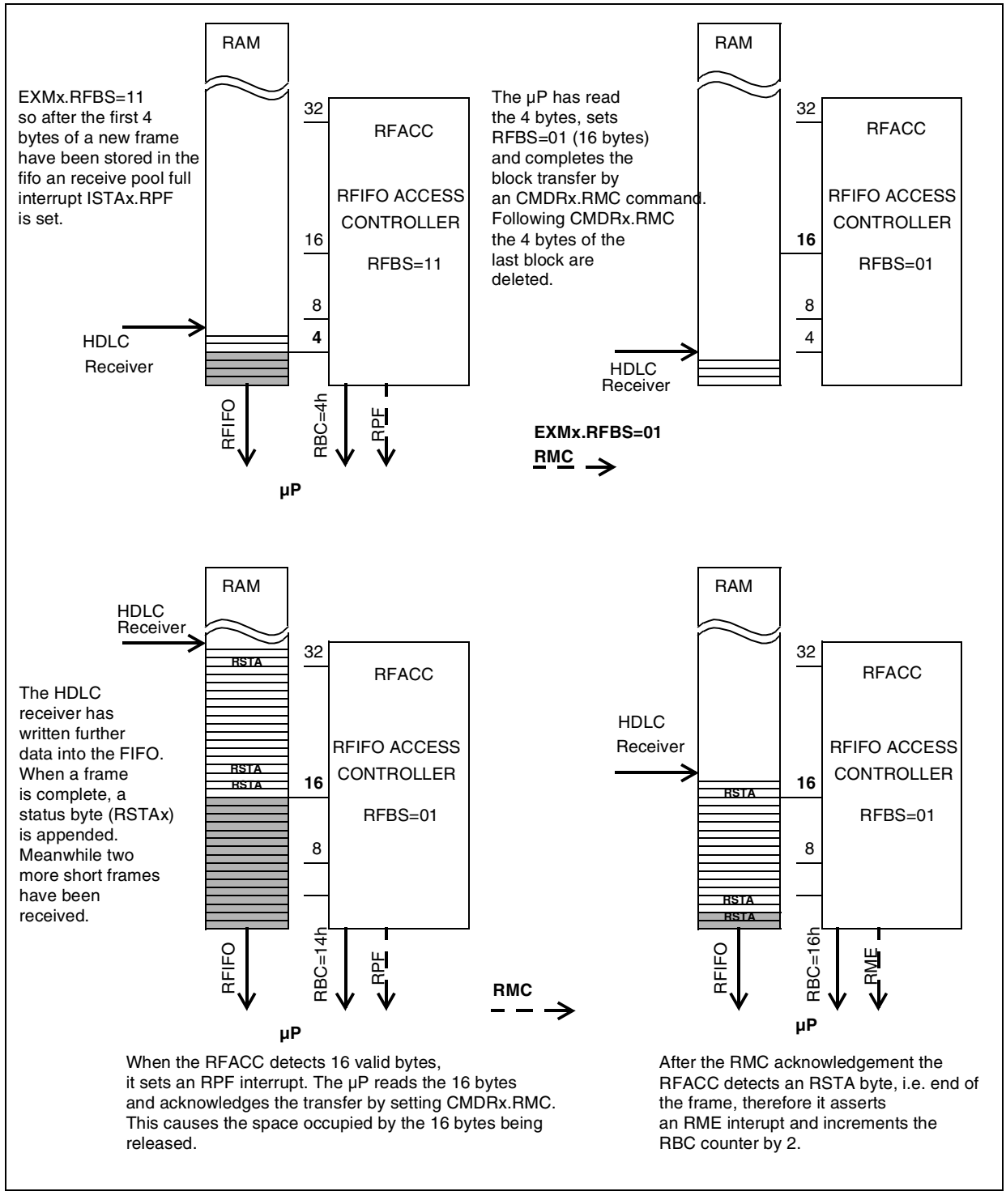


Figure 90 RFIFO Operation

Possible Error Conditions during Reception of Frames

If parts of a frame get lost because the receive FIFO is full, the Receive Data Overflow (RDO) byte in the RSTAx byte will be set. If a complete frame is lost, i.e. if the FIFO is full when a new frame is received, the receiver will assert a Receive Frame Overflow (RFO) interrupt.

The microcontroller sees a cyclic buffer, i.e. if it tries to read more data than available, it reads the same data again and again. On the other hand, if it doesn't read or doesn't want to read all data, they are deleted anyway after the RMC command.

If the microcontroller reads data without a prior RME or RPF interrupt, the content of the RFIFOx would not be corrupted, but new data is only transferred to the host as long as new valid data is available in the RFIFOx, otherwise the last data is read again and again.

The general procedures for a data reception sequence are outlined in the flow diagram in **Figure 91**.

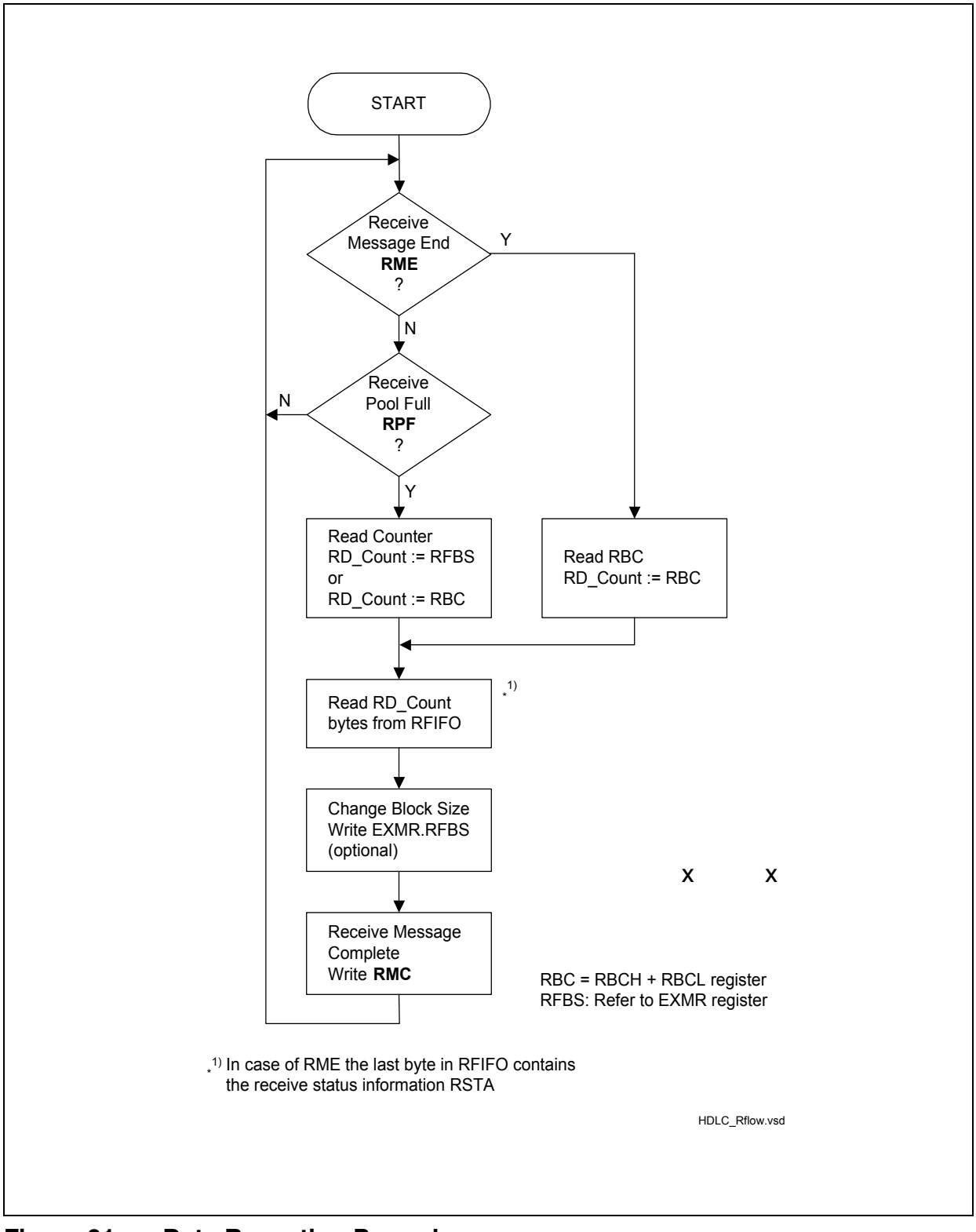


Figure 91 Data Reception Procedures

Figure 92 gives an example of an interrupt controlled reception sequence, supposed that a long frame (68 byte) followed by two short frames (12 byte each) are received. The FIFO threshold (block size) is set to 32 byte in this example:

- After 32 byte of frame 1 have been received an RPF interrupt is generated to indicate that a data block can be read from the RFIFOx.
- The host reads the first data block from RFIFOx and acknowledges the reception by RMC. Meanwhile the second data block is received and stored in RFIFOx.
- The second 32 byte block is indicated by RPF which is read and acknowledged by the host as described before.
- The reception of the remaining 4 bytes plus RSTAx are indicated by RME (i.e. the receive status is always appended to the end of the frame).
- The host gets the number of bytes (COUNT = 5) from RBCLx/RBCHx and reads out the RFIFOx and optionally the status register RSTA. The frame is acknowledged by RMC.
- The second frame is received and indicated by RME interrupt.
- The host gets the number of bytes (COUNT = 13) from RBCLx/RBCHx and reads out the RFIFOx and optionally the status register. The RFIFOx is acknowledged by RMC.
- The third frame is transferred in the same way.

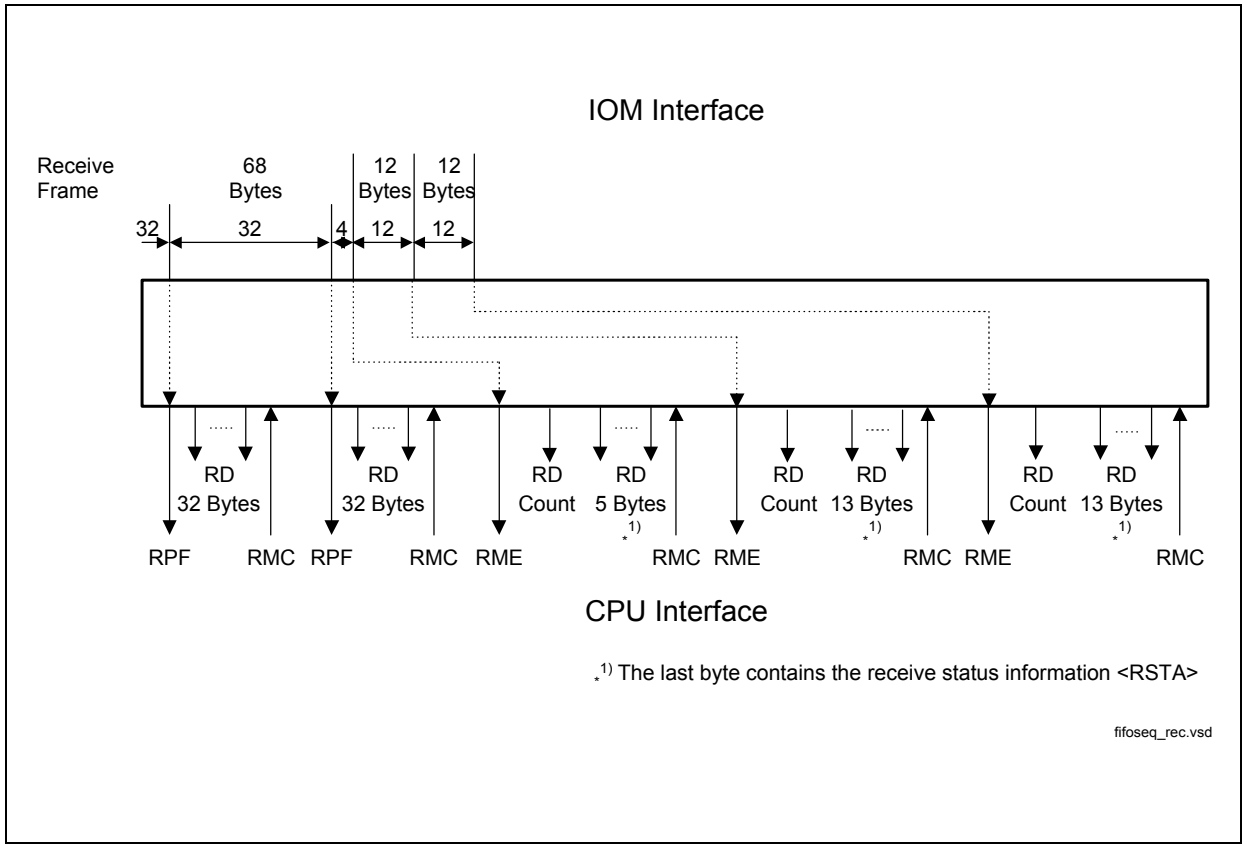


Figure 92 Reception Sequence Example

5.6.2.2 Receive Frame Structure

The management of the received HDLC frames as affected by the different operating modes (see **Chapter 5.6.1**) is shown in **Figure 93**.

Figure 93 Receive Data Flow

The SIUC-BA indicates to the host that a new data block can be read from the RFIFOx by means of an RPF interrupt (see previous chapter). User data is stored in the RFIFOx and information about the received frame is available in the RBCLx and RBCHx registers and the RSTAx bytes which are listed in **Table 23**.

Table 23 Receive Information at RME Interrupt

Information	Register	Bit	Mode
Type of frame (Command/ Response)	RSTAx	C/R	Non-auto mode, 2-byte address field Transparent mode 1
Recognition of SAPI	RSTAD RSTAB	SA1, 0 HA1, 0	Non-auto mode, 2-byte address field Transparent mode 1
Recognition of TEI	RSTAD RSTAB	TA LA	All except transparent mode 0
Result of CRC check (correct/incorrect)	RSTAx	CRC	All
Valid Frame	RSTAx	VFR	All
Abort condition detected (yes/no)	RSTAx	RAB	All
Data overflow during reception of a frame (yes/no)	RSTAx	RDO	All
Number of bytes received in RFIFO	RBCL	RBC4-0	All (also see **Table 22**)
Message length	RBCLx RBCHx	RBC11-0	All
RFIFO Overflow	RBCHx	OV	All

The RSTAx register is always appended in the RFIFOx as last byte to the end of a frame.

5.6.3 Data Transmission

5.6.3.1 Structure and Control of the Transmit FIFO

The cyclic transmit FIFO buffers with a length of 64-byte for D-channel and 128 byte for each of the two B-channels have variable FIFO block sizes (thresholds) of

- 16 or 32 bytes for D-channel and
- 32 or 64 bytes for B-channels

which can be selected by setting the corresponding XFBS bits in the EXMx registers. There are three different interrupt indications in the ISTAx registers concerned with the transmission of data:

– **XPR** (**T**ransmit **P**ool **R**eady) interrupt, indicating that a data block of up to 16 or 32 byte (D-channel), 32 or 64 byte (B-channel) can be written to the XFIFOx (block size selected via EXMx.XFBS).
An XPR interrupt is generated either
  - after an XRES (Transmitter Reset) command (which is issued for example for frame abort) or
  - when a data block from the XFIFOx is transmitted and the corresponding FIFO space is released to accept further data from the host.

– **XDU** (**T**ransmit **D**ata **U**nderrun) interrupt, indicating that the transmission of the current frame has been aborted (seven consecutive ’1’s are transmitted) as the XFIFOx holds no further transmit data. This occurs if the host fails to respond to an XPR interrupt quickly enough.

– Only valid for D-channel:
XMR (**T**ransmit **M**essage **R**epeat) interrupt, indicating that the transmission of the complete last frame has to be repeated as a collision on the S bus has been detected and the XFIFOx does not hold the first data bytes of the frame (collision after the 16th/32nd byte or after the 32nd/64th byte of the frame, respectively).

The occurence of an XDU or XMR interrupt clears the XFIFOx and an XMR interrupt is issued together with an XDU or XMR interrupt, respectively. Data cannot be written to the XFIFOx as long as an XDU/XMR interrupt is pending.

Three different control commands are used for transmission of data:

– **XTF** (**T**ransmit **T**ransparent **F**rame) command, telling the SIUC-BA that up to 16 or 32 byte (D-channel) or 32 or 64 byte (B-channel) have been written to the XFIFOx and should be transmitted. A start flag is generated automatically.

– **XME** (**T**ransmit **M**essage **E**nd) command, telling the SIUC-BA that the last data block written to the XFIFOx completes the corresponding frame and should be transmitted. This implies that according to the selected mode a frame end (CRC + closing flag) is generated and appended to the frame.

– **XRES** (**T**ransmitter **R**eset) command, resetting the HDLC transmitter and clearing the transmit FIFO of any data. After an XRES command the transmitter always sends an abort sequence, i.e. this command can be used to abort a transmission. Pending interrupt indications of the transmitter are not cleared by XRES, but have to be cleared by reading these interutps.

Optionally two additional status conditions can be read by the host:

– **XDOV** (**T**ransmit **D**ata **O**verflow), indicating that the data block size has been exceeded, i.e. more than 16 or 32 byte (D-channel) or 32 or 64 byte (B-channel) were entered and data was overwritten.
– **XFW** (**T**ransmit **F**IFO **W**rite Enable), indicating that data can be written to the XFIFOx. This status flag may be polled instead of or in addition to XPR.

Note: The significant interrupts and commands are underlined as only these are usually used during a normal transmission sequence.

The XFIFO requests service from the microcontroller by setting a bit in the ISTAx register, which causes an interrupt (XPR, XDU, XMR). The microcontroller can then read the status register STARx (XFW, XDOV), write data in the FIFO and it can change the transmit FIFO block size (EXMx.XFBS) if required.
The instant of the initiation of a transmit pool ready (XPR) interrupt after different transmit control commands is listed in **Table 24**.

Table 24 XPR Interrupt (availability of XFIFOx) after XTF, XME Commands

CMDRx Register	Transmit pool ready (XPR) interrupt initiated ...
XTF	as soon as the selected buffer size in the FIFOx is available.
XTF & XME	after the successful transmission of the closing flag. The transmitter always sends an abort sequence.
XME	as soon as the selected buffer size in the FIFO is available, two consecutive frames share flags.

When setting XME the transmitter appends the CRC and the endflag at the end of the frame. When XTF & XME has been set, the XFIFOx is locked until successful transmission of the current frame, so a consecutive XPR interrupt also indicates successful transmission of the frame whereas after XME or XTF the XPR interrupt is asserted as soon as there is space for one data block in the XFIFOx.

The transfer block size is 32 bytes (for D-channel) or 64 bytes (for B-channel) by default, but sometimes, if the microcontroller has a high computational load, it is useful to increase the maximum reaction time for an XPR interrupt. The maximum reaction time is:

t_{max} = (XFIFOx size - XFBS) / data transmission rate

With a selected block size of 16 bytes (D-channel only) an XPR interrupt indicates when a transmit FIFO space of at least 16 bytes is available to accept further data, i.e. there are still a maximum of 48 bytes (64 bytes - 16 bytes) to be transmitted. With a 32 bytes block size (D- or B-channel) the XPR is initiated when a transmit FIFO space of at least 32 bytes is available to accept further data, i.e. there are still a maximum of 32 bytes (D-channel: 64 bytes - 32 bytes) or 96 bytes (B-channel: 128 bytes - 32 bytes) to be transmitted. The maximum reaction time for the smaller block size is 50 % higher with the trade-off of a doubled interrupt load. With a selected block size an XPR always indicates the available space in the XFIFOx, so any number of bytes smaller than the selected XFBS may be stored in the FIFO during one “write block“ access cycle.

Similar to RFBS for the receive FIFO, a new setting of XFBS takes effect after the next XTF,XME or XRES command. XRES resets the XFIFOx.

The XFIFOx can hold any number of frames fitting in the 64 bytes (D-channel) or 128 bytes (B-channel), respectively.

Possible Error Conditions during Transmission of Frames

If the transmitter sees an empty FIFO, i.e. if the microcontroller doesn’t react fast enough to an XPR interrupt, an XDU (transmit data underrun) interrupt will be generated. If the HDLC channel becomes unavailable during transmission the transmitter tries to repeat the current frame as specified in the LAPD protocol. This is impossible after the first data block has been sent (16 or 32 bytes for D-channel; 32 or 64 byte for B-channel), in this case an XMR transmit message repeat interrupt is set and the microcontroller has to send the whole frame again.

Both XMR and XDU interrupts cause a reset of the XFIFOx. The XFIFOx is locked while an XMR or XDU interrupt is pending, i.d. all write actions of the microcontroller will be ignored as long as the microcontroller hasn’t read the ISTAx register with the set XDU, XMR interrupts.

If the microcontroller writes more data than allowed (block size), then the data in the XFIFOx will be corrupted and the STARx.XDOV bit is set. If this happens, the microcontroller has to abort the transmission by CMDRx.XRES and start new.

The general procedures for a data transmission sequence are outlined in the flow diagram in **Figure 94**.

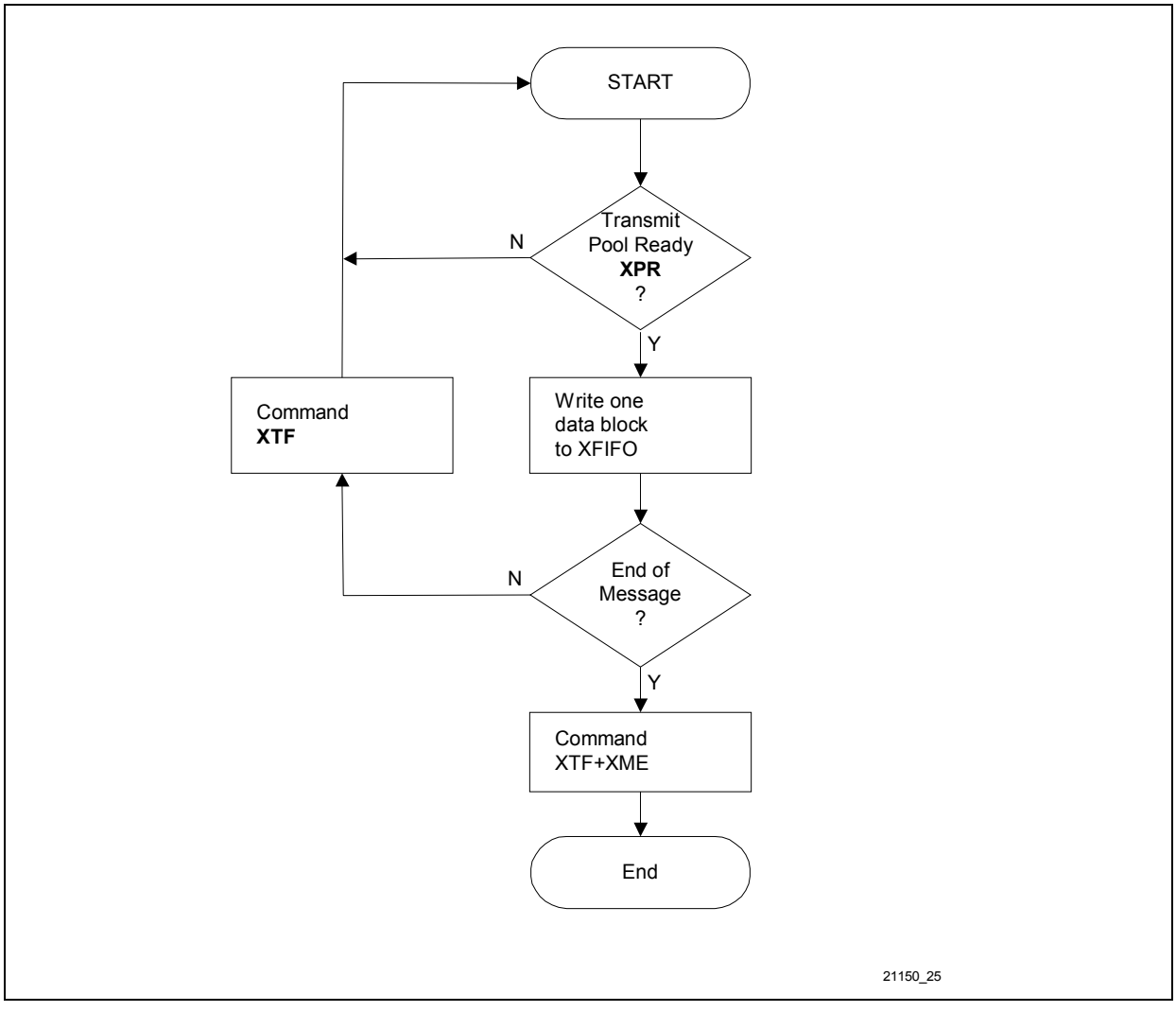


Figure 94 Data Transmission Procedure

The following description gives an example for the transmission of a 76 byte frame with a selected block size of 32 byte:

- The host writes 32 bytes to the XFIFOx, issues an XTF command and waits for an XPR interrupt in order to continue with entering data.
- The SIUC-BA immediately issues an XPR interrupt (as remaining XFIFOx space is not used) and starts transmission.
- Due to the XPR interrupt the host writes the next 32 bytes to the XFIFOx, followed by the XTF command, and waits for XPR.
- As soon as the last byte of the first block is transmitted, the SIUC-BA releases an XPR (XFIFOx space of first data block is free again) and continues transmitting the second block.
- The host writes the remaining 12 bytes of the frame to the XFIFOx and issues the XTF command together with XME to indicate that this is the end of frame.
- After the last byte of the frame has been transmitted the SIUC-BA releases an XPR interrupt and the host may proceed with transmission of a new frame.

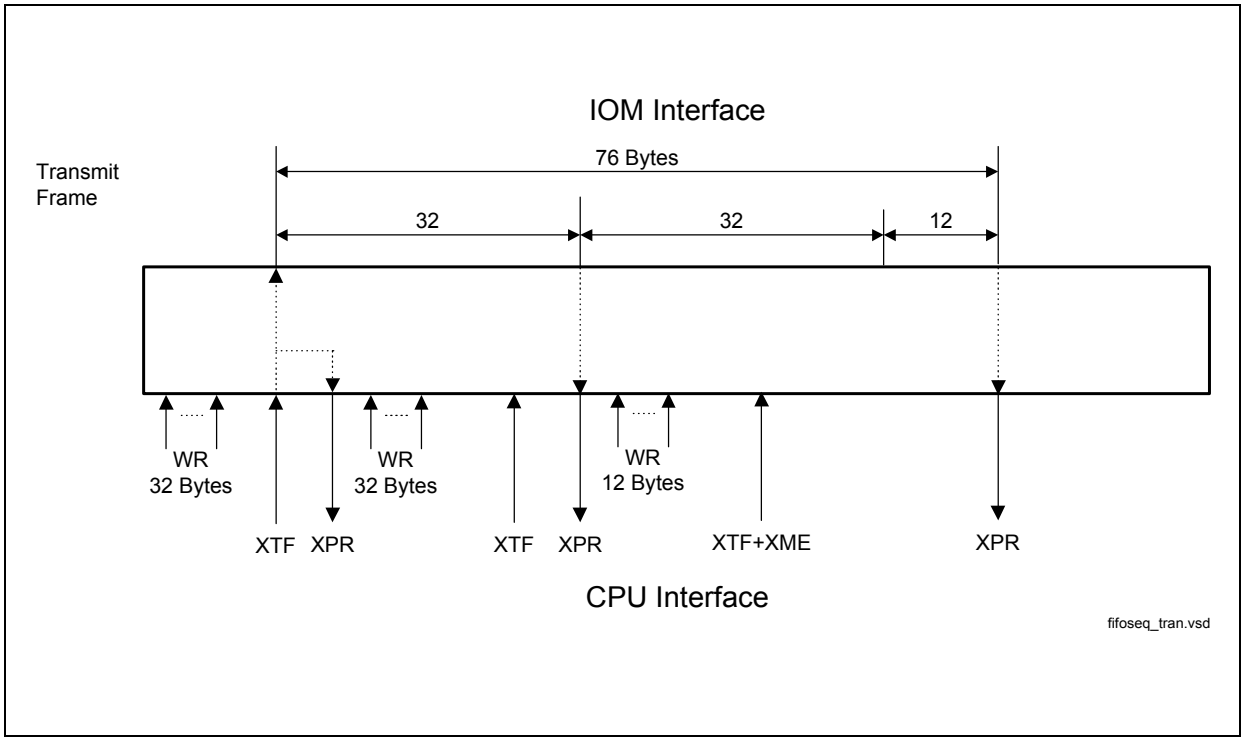


Figure 95 Transmission Sequence Example

5.6.3.2 Transmit Frame Structure

The transmission of transparent frames (XTF command) is shown in **Figure 96**.

For transparent frames, the whole frame including address and control field must be written to the XFIFOx. The host configures whether the CRC is generated and appended to the frame (default) or not (selected in EXMx.XCRC).

Further, the host selects the interframe time fill signal which is transmitted between HDCL frames (EXMx.ITF). One option is to send continuous flags (’01111110’), however if D-channel access handling (collision resolution on the S bus) is required, the signal must be set to idle (continuous ’1’s are transmitted). Reprogramming of ITF takes effect only after the transmission of the current frame has been completed or after an XRES command.

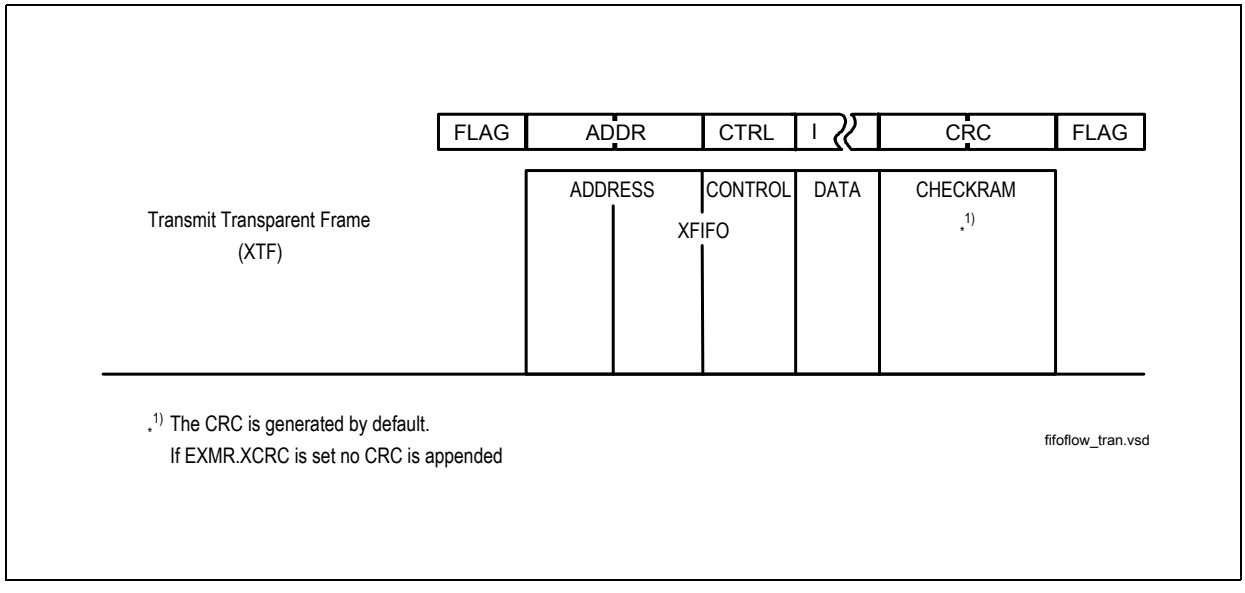


Figure 96 Transmit Data Flow

5.6.4 Access to IOM-2 channels

By setting the enable HDLC data bits (EN_D, EN_B1H, EN_B2H) in the DCI_CR register (D-channel) and in the BCH_CR register (B-channel) the HDLC controller can access the D, B1 and B2 channels or any combination of them (e.g. 18 bit IDSL data 2B+D). In all modes (except extended transparent mode) transmission always works frame aligned, i.e. it starts with the first selected channel, whereas reception searches for a flag anywhere in the serial data stream.

5.6.5 Extended Transparent Mode

This non-HDLC mode is selected by setting MODE2...0 to ’100’. In extended transparent mode fully transparent data transmission/reception without HDLC framing is performed i.e. without FLAG generation/recognition, CRC generation/check, bitstuffing mechanism. This allows user specific protocol variations.

Transmitter

The transmitter sends the data out of the FIFO without manipulation. Transmission is always IOM-2 frame aligned and byte aligned, i.e. transmission starts in the first selected channel (B1, B2, D, according to the setting of register DCI_CR or BCH_CR in the IOM-2 Handler) of the next IOM-2 frame.

The FIFO indications and commands are the same as in other modes.

If the microcontroller sets XTF & XME the transmitter responds with an XPR interrupt after sending the last byte, then it returns to its idle state (sending continuous ‘1’).

If the collision detection is enabled in D-channel (MODE.DIM = ’0x1’) the stop go bit (S/G) can be used as clear to send indication as in any other mode. If the S/G bit is set to ’1’ (stop) during transmission the transmitter responds always with an XMR (transmit message repeat) interrupt.

If the microcontroller fails to respond to a XPR interrupt in time and the transmitter runs out of data then it will assert an XDU (transmit data underrun) interrupt.

Receiver

The reception is IOM-2 frame aligned and byte aligned, like transmission, i.e. reception starts in the first selected channel (B1, B2, D, according to the setting of registers DCI_CR and BCH_CR in the IOM-2 Handler) of the next IOM-2 frame. The FIFO indications and commands are the same as in others modes.

All incoming data bytes are stored in the RFIFOx and is additionally made available in RSTAx. If the FIFO is full an RFO interrupt is asserted (EXMx.SRA = ’0’).

Note: In the extended transparent mode the EXMx register has to be set to ’xxx00000’

5.6.6 HDLC Controller Interrupts

The cause of an interrupt related to the HDLC controllers is indicated in the ISTA register by the ICD bit for D-channel, ICA for B-channel A and ICB for B-channel B. These bits point to the different interrupt sources of the HDLC controllers in the ISTAD and ISTAB registers. The individual interrupt sources of the HDLC controllers during reception and transmission of data are explained in **Chapter 5.6.2.1** or **Chapter 5.6.3.1** respectively.

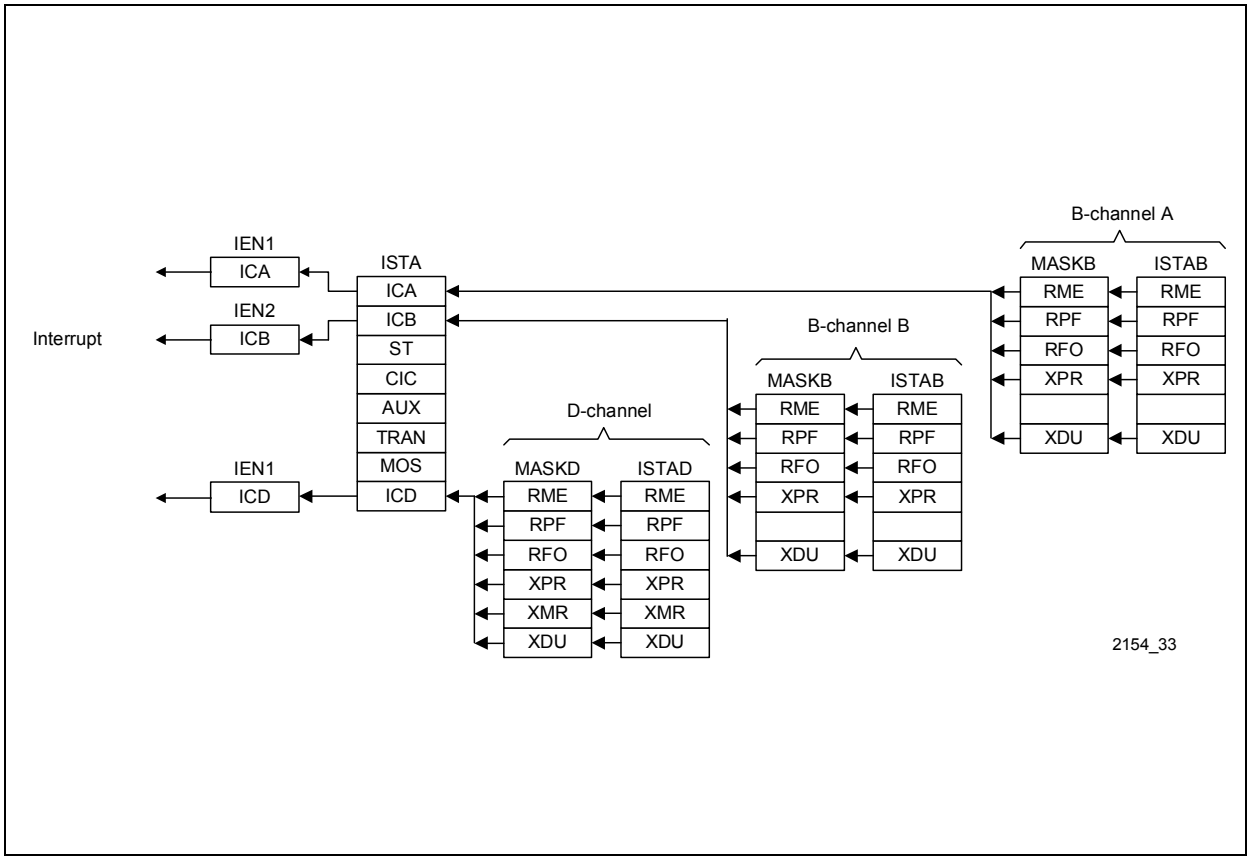


Figure 97 Interrupt Status Registers of the HDLC Controllers

Each interrupt source in the ISTAD and ISTAB registers can selectively be masked by setting the corresponding bit in MASKD/MASKB to “1”.

5.7 Test Functions

The SIUC-BA provides test and diagnostic functions for the S-interface, the D-channel and each of the two B-channels:

- Digital loop via TLP (Test Loop, TMD and TMB registers) command bit (**Figure 98**): The TX path of layer 2 is internally connected with the RX path of layer 2. The output from layer 1 (S/T) on DD is ignored. This is used for testing SIUC-BA functionality excluding layer 1 (loopback between XFIFOx and RFIFOx).

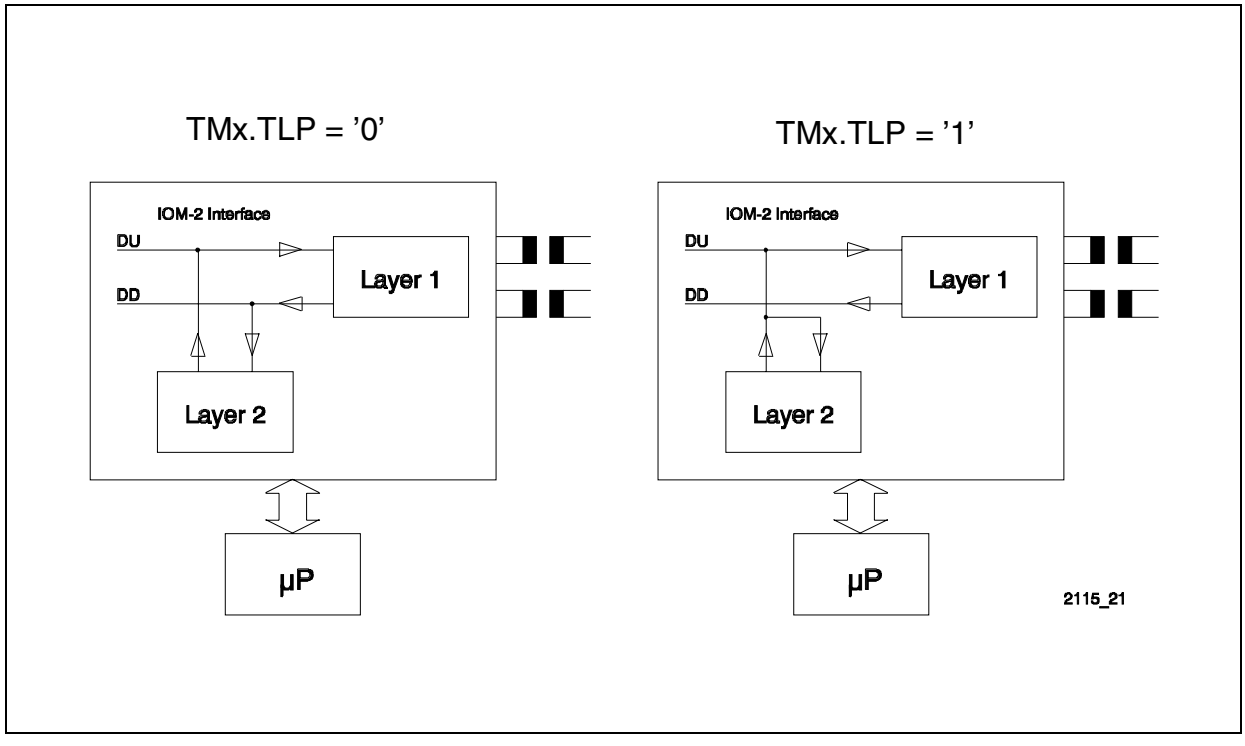


Figure 98 Layer 2 Test Loops

- Test of layer-2 functions while disabling all layer-1 functions and pins associated with them (including clocking) via bit TR_CONF0.DIS_TR. The HDLC controllers can still operate via IOM-2. DCL and FSC pins become input.

- loop at the analog end of the S interface;

  Test loop 3 is activated with the C/I channel command Activate Request Loop (ARL). An S interface is not required since INFO3 is looped back internally to the receiver. When the receiver has synchronized itself to this signal, the message "Test Indication" (or "Awake Test Indication") is delivered in the C/I channel. No signal is transmitted over the S interface.
  In the test loop mode the S interface awake detector is enabled, i.e. if a level is detected (e.g. Info 2/Info 4) this will be reported by the Resynchronization Indication (RSY). The loop function is not effected by this condition and the internally generated 192-kHz line clock does not depend on the signal received at the S interface.

- transmission of special test signals on the S/T interface according to the modified AMI code are initiated via a C/I command written in CIX0 register.

  Two kinds of test signals may be sent by the SIUC-BA:

  – single pulses and
  – continuous pulses.

  The single pulses are of alternating polarity, one S interface bit period wide, 0.25 ms apart, with a repetition frequency of 2 kHz. Single pulses can be sent in all applications. The corresponding C/I command in TE, LT-S and LT-T applications is TM1.

  Continuous pulses are likewise of alternating polarity, one S-interface bit period wide, but they are sent continuously. The repetition frequency is 96 kHz. Continuous pulses may be transmitted in all applications. This test mode is entered in LT-S, LT-T and TE applications with the C/I command TM2.

5.8 ISDN Register Description

The register mapping of the SIUC-BA is shown in **Figure 99**. All addresses mentioned must be prefixed by $F8_H$ to correspond to the ISDN address space $F800_H$ - $F8FF_H$.

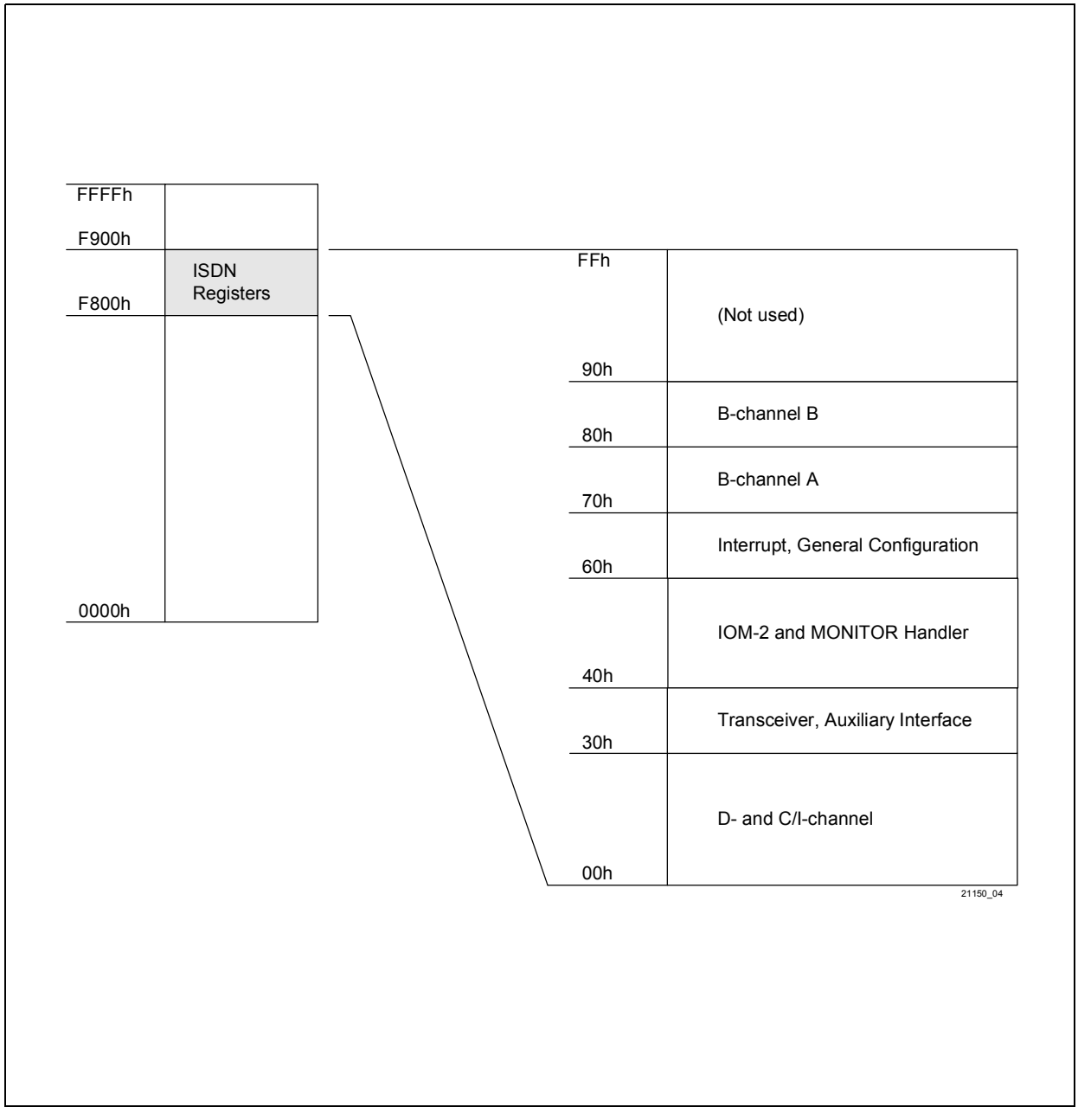


Figure 99 Register Mapping of the SIUC-BA

The register address range from 00_H-$2F_H$ is assigned to the D-channel HDLC controller and the C/I-channel handler.

The register set ranging from 30_H-$3F_H$ pertains to the transceiver and auxiliary interface registers.

The address range from 40_H-$5B_H$ is assigned to the IOM handler with the registers for timeslot and data port selection (TSDP) and the control registers (CR) for the transceiver data (TR), Monitor data (MON), HDLC/CI data (HCI) and controller access data (CDA), serial data strobe signal (SDS), IOM interface (IOM) and synchronous transfer interrupt (STI).

The address range from $5C_H$-$5F_H$ pertains to the MONITOR handler.

General interrupt and configuration registers are contained in the address range 60_H-65_H.

The address range 70_H-$8F_H$ is assigned to the two B-channel FIFOs and HDLC controllers having an identical set of registers.

The register summaries of the SIUC-BA are shown in the following tables containing the abbreviation of the register name and the register bits, the register address, the reset values and the register type (Read/Write). A detailed register description follows these register summaries.

The register summaries and the description are sorted in ascending order of the register address.

D-channel HDLC, C/I-channel Handler

Name	7	6	5	4	3	2	1	0	ADDR	R/W	RES
RFIFOD	D-Channel Receive FIFO								00_H-$1F_H$	R	
XFIFOD	D-Channel Transmit FIFO								00_H-$1F_H$	W	
ISTAD	RME	RPF	RFO	XPR	XMR	XDU	0	0	20_H	R	10_H
MASKD	RME	RPF	RFO	XPR	XMR	XDU	1	1	20_H	W	FF_H
STARD	XDOV	XFW	0	0	RACI	0	XACI	0	21_H	R	40_H
CMDRD	RMC	RRES	0	STI	XTF	0	XME	XRES	21_H	W	00_H
MODED	MDS2	MDS1	MDS0	0	RAC	DIM2	DIM1	DIM0	22_H	R/W	$C0_H$
EXMD1	XFBS	RFBS		SRA	XCRC	RCRC	0	ITF	23_H	R/W	00_H
TIMR2	CNT			VALUE					24_H	R/W	00_H
SAP1	SAPI1						0	MHA	25_H	W	FC_H
SAP2	SAPI2						0	MLA	26_H	W	FC_H
RBCLD	RBC7							RBC0	26_H	R	00_H
RBCHD	0	0	0	OV	RBC11			RBC8	27_H	R	00_H
TEI1	TEI1							EA1	27_H	W	FF_H
TEI2	TEI2							EA2	28_H	W	FF_H
RSTAD	VFR	RDO	CRC	RAB	SA1	SA0	C/R	TA	28_H	R	$0F_H$
TMD	0	0	0	0	0	0	0	TLP	29_H	R/W	00_H
	reserved								$2A-2D_H$		
CIR0	CODR0				CIC0	CIC1	S/G	BAS	$2E_H$	R	$F3_H$
CIX0	CODX0				TBA2	TBA1	TBA0	BAC	$2E_H$	W	FE_H
CIR1	CODR1						CICW	CI1E	$2F_H$	R	FE_H

CIX1	CODX1	CICW	CI1E	$2F_H$	W	FE_H

Transceiver, Auxiliary Interface

Name	7	6	5	4	3	2	1	0	ADDR	R/W	RES
TR_CONF0	DIS_TR	0	EN_ICV	0	L1SW	0	EXLP	LDD	30_H	R/W	01_H
TR_CONF1	0	RPLL_ADJ	EN_SFSC	0	0	x	x	x	31_H	R/W	
TR_CONF2	DIS_TX	PDS	0	RLP	0	0	SGP	SGD	32_H	R/W	80_H
TR_STA	RINF		0	ICV	0	FSYN	0	LD	33_H	R	00_H
TR_CMD	XINF			DPRIO	TDDIS	PD	LP_A	0	34_H	R/W	08_H
SQRR1	MSYN	MFEN	0	0	SQR11	SQR12	SQR13	SQR14	35_H	R	40_H
SQXR1	0	MFEN	0	0	SQX11	SQX12	SQX13	SQX14	35_H	W	$4F_H$
SQRR2	SQR21	SQR22	SQR23	SQR24	SQR31	SQR32	SQR33	SQR34	36_H	R	00_H
	reserved								36_H	W	
SQRR3	SQR41	SQR42	SQR43	SQR44	SQR51	SQR52	SQR53	SQR54	37_H	R	00_H
	reserved								37_H	W	
ISTATR	0	x	x	x	LD	RIC	SQC	SQW	38_H	R	00_H
MASKTR	1	1	1	1	LD	RIC	SQC	SQW	39_H	R/W	FF_H
TR_MODE	0	0	0	0	DCH_INH	MODE 2	MODE 1	MODE 0	$3A_H$	R/W	00_H
	reserved								$3B_H$		
ACFG1	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0	$3C_H$	R/W	00_H
ACFG2	A7SEL	A5SEL	FBS	A4SEL	ACL	LED	EL2	EL1	$3D_H$	R/W	00_H

Transceiver, Auxiliary Interface

Name	7	6	5	4	3	2	1	0	ADDR	R/W	RES
AOE	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0	$3E_H$	R/W	FF_H
ARX	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0	$3F_H$	R	
ATX	AT7	AT6	AT5	AT4	AT3	AT2	AT1	AT0	$3F_H$	W	00_H

IOM Handler (Timeslot , Data Port Selection, CDA Data and CDA Control Register)

Name	7	6	5	4	3	2	1	0	ADDR	R/W	RES
CDA10	Controller Data Access Register (CH10)								40_H	R/W	FF_H
CDA11	Controller Data Access Register (CH11)								41_H	R/W	FF_H
CDA20	Controller Data Access Register (CH20)								42_H	R/W	FF_H
CDA21	Controller Data Access Register (CH21)								43_H	R/W	FF_H
CDA_ TSDP10	DPS	0	0	TSS					44_H	R/W	00_H
CDA_ TSDP11	DPS	0	0	TSS					45_H	R/W	01_H
CDA_ TSDP20	DPS	0	0	TSS					46_H	R/W	80_H
CDA_ TSDP21	DPS	0	0	TSS					47_H	R/W	81_H
BCHA_ TSDP_ BC1	DPS	0	0	TSS					48_H	R/W	80_H
BCHA_ TSDP_ BC2	DPS	0	0	TSS					49_H	R/W	81_H

BCHB_ TSDP_ BC1	DPS	0	0	TSS					$4A_H$	R/W	81_H
BCHB_ TSDP_ BC2	DPS	0	0	TSS					$4B_H$	R/W	85_H
TR_ TSDP_ BC1	DPS	0	0	TSS					$4C_H$	R/W	00_H
TR_ TSDP_ BC2	DPS	0	0	TSS					$4D_H$	R/W	01_H
CDA1_ CR	0	0	EN_ TBM	EN_I1	EN_I0	EN_O1	EN_O0	SWAP	$4E_H$	R/W	00_H
CDA2_ CR	0	0	EN_ TBM	EN_I1	EN_I0	EN_O1	EN_O0	SWAP	$4F_H$	R/W	00_H

IOM Handler (Control Registers, Synchronous Transfer Interrupt Control), MONITOR Handler

Name	7	6	5	4	3	2	1	0	ADDR	R/W	RES
TR_CR (CI_CS=0)	EN_ D	EN_ B2R	EN_ B1R	EN_ B2X	EN_ B1X	CS2-0			50_H	R/W	$F8_H$
TRC_CR (CI_CS=1)	0	0	0	0	0	CS2-0			50_H	R/W	00_H
BCHA_ CR	DPS_ D	0	EN_D	EN_ BC2	EN_ BC1	CS2-0			51_H	R/W	80_H
BCHB_ CR	DPS_ D	0	EN_D	EN_ BC2	EN_ BC1	CS2-0			52_H	R/W	81_H
DCI_CR (CI_CS=0)	DPS_ CI1	EN_ CI1	D_ EN_D	D_ EN_B2	D_ EN_B1	CS2-0			53_H	R/W	

DCIC_CR (CI_CS=1)	0	0	0	0	0	CS2-0			53_H	R/W	00_H
MON_CR	DPS	EN_ MON	0	0	0	CS2-0			54_H	R/W	40_H
SDS_CR	ENS_ TSS	ENS_ TSS+1	ENS_ TSS+3	TSS					55_H	R/W	00_H
	reserved								56_H		
IOM_CR	SPU	0	CI_CS	TIC_ DIS	EN_ BCL	CLKM	DIS_ OD	DIS_ IOM	57_H	R/W	08_H
STI	STOV 21	STOV 20	STOV 11	STOV 10	STI 21	STI 20	STI 11	STI 10	58_H	R	00_H
ASTI	0	0	0	0	ACK 21	ACK 20	ACK 11	ACK 10	58_H	W	00_H
MSTI	STOV 21	STOV 20	STOV 11	STOV 10	STI 21	STI 20	STI 11	STI 10	59_H	R/W	FF_H
SDS_ CONF	0	0	0	0	DIOM_ INV	DIOM_ SDS	0	SDS_ BCL	$5A_H$	R/W	00_H
MCDA	MCDA21		MCDA20		MCDA11		MCDA10		$5B_H$	R	FF_H
MOR	MONITOR Receive Data								$5C_H$	R	FF_H
MOX	MONITOR Transmit Data								$5C_H$	W	FF_H
MOSR	MDR	MER	MDA	MAB	0	0	0	0	$5D_H$	R	00_H
MOCR	MRE	MRC	MIE	MXC	0	0	0	0	$5E_H$	R/W	00_H
MSTA	0	0	0	0	0	MAC	0	TOUT	$5F_H$	R	00_H
MCONF	0	0	0	0	0	0	0	TOUT	$5F_H$	W	00_H

Interrupt, General Configuration Registers

NAME	7	6	5	4	3	2	1	0	ADDR	R/W	RES
ISTA	ICA	ICB	ST	CIC	AUX	TRAN	MOS	ICD	60_H	R	00_H
ISTA_INIT									60_H	W	FF_H
AUXI	0	0	EAW	WOV	TIN3	TIN2	INT2	INT1	61_H	R	00_H
AUXM	1	1	EAW	WOV	TIN3	TIN2	INT2	INT1	61_H	W	FF_H
MODE1	0	0	0	WTC1	WTC2	CFS	RSS2	RSS1	62_H	R/W	00_H
	reserved								63_H	R/W	00_H
ID	0	0	DESIGN						64_H	R	01_H
SRES	RES_CI	RES_BCHA	RES_BCHB	RES_MON	RES_DCH	RES_IOM	RES_TR	RES_RSTO	64_H	W	00_H
TIMR3	TMD	0	CNT						65_H	R/W	00_H
	reserved								66_H-$6F_H$		

B-channel HDLC Control Registers (channel A / B)

Name	7	6	5	4	3	2	1	0	ADDR	R/W	RES
ISTAB	RME	RPF	RFO	XPR	0	XDU	0	0	$70_H/80_H$	R	10_H
MASKB	RME	RPF	RFO	XPR	1	XDU	1	1	$70_H/80_H$	W	FF_H
STARB	XDOV	XFW	0	0	RACI	0	XACI	0	$71_H/81_H$	R	40_H
CMDRB	RMC	RRES	0	0	XTF	0	XME	XRES	$71_H/81_H$	W	00_H
MODEB	MDS2	MDS1	MDS0	0	RAC	0	0	0	$72_H/82_H$	R/W	$C0_H$
EXMB	XFBS	RFBS		SRA	XCRC	RCRC	0	ITF	$73_H/83_H$	R/W	00_H
	reserved								$74_H/84_H$		
RAH1	RAH1						0	MHA	$75_H/85_H$	W	00_H
RAH2	RAH2						0	MLA	$76_H/86_H$	W	00_H
RBCLB	RBC7							RBC0	$76_H/86_H$	R	00_H
RBCHB	0	0	0	OV	RBC11			RBC8	$77_H/87_H$	R	00_H
RAL1	RAL1								$77_H/87_H$	W	00_H
RAL2	RAL2								$78_H/88_H$	W	00_H
RSTAB	VFR	RDO	CRC	RAB	HA1	HA0	C/R	LA	$78_H/88_H$	R	$0E_H$
TMB	0	0	0	0	0	0	0	TLP	$79_H/89_H$	R/W	00_H
RFIFOB	B-Channel Receive FIFO								$7A_H/8A_H$	R	
XFIFOB	B-Channel Transmit FIFO								$7A_H/8A_H$	W	
	reserved								$7B_H$-$7F_H$ $8B_H$-$8F_H$		

5.8.1 D-channel HDLC Control and C/I Registers

5.8.1.1 RFIFOD - Receive FIFO D-Channel

	7						0	
RFIFOD	Receive data							RD (00-1F)

A read access to any address within the range 00h-1Fh gives access to the “current” FIFO location selected by an internal pointer which is automatically incremented after each read access.

The RFIFOD contains up to 32 bytes of received data.

After an ISTAD.RPF interrupt, a complete data block is available. The block size can be 4, 8, 16 or 32 bytes depending on the EXMD2.RFBS setting.

After an ISTAD.RME interrupt, the number of received bytes can be obtained by reading the RBCLD register.

5.8.1.2 XFIFOD - Transmit FIFO D-Channel

	7						0	
XFIFOD	Transmit data							WR (00-1F)

A write access to any address within the range 00-1F$_H$ gives access to the “current” FIFO location selected by an internal pointer which is automatically incremented after each write access.

Depending on EXMD2.XFBS up to 16 or 32 bytes of transmit data can be written to the XFIFOD following an ISTAD.XPR interrupt.

5.8.1.3 ISTAD - Interrupt Status Register D-Channel

Value after reset: 10_H

	7							0	
ISTAD	RME	RPF	RFO	XPR	XMR	XDU	0	0	RD (20)

RME ... Receive Message End

One complete frame of length less than or equal to the defined block size (EXMD1.RFBS) or the last part of a frame of length greater than the defined block size has been received. The contents are available in the RFIFOD. The message length and additional information may be obtained from RBCHD and RBCLD and the RSTAD register.

RPF ... Receive Pool Full

A data block of a frame longer than the defined block size (EXMD1.RFBS) has been received and is available in the RFIFOD. The frame is not yet complete.

RFO ... Receive Frame Overflow

The received data of a frame could not be stored, because the RFIFOD is occupied. The whole message is lost.

This interrupt can be used for statistical purposes and indicates that the microcontroller does not respond quickly enough to an RPF or RME interrupt (ISTAD).

XPR ... Transmit Pool Ready

A data block of up to the defined block size 16 or 32 (EXMD1.XFBS) can be written to the XFIFOD.

An XPR interrupt will be generated in the following cases:

- after an XTF or XME command as soon as the 16 or 32 bytes in the XFIFO are available and the frame is not yet complete
- after an XTF together with an XME command is issued, when the whole frame has been transmitted
- after a reset of the transmitter (XRES)
- after a device reset

XMR ... Transmit Message Repeat

The transmission of the last frame has to be repeated because a collision on the S bus has been detected after the 16th/32nd data byte of a transmit frame.

If an XMR interrupt occurs the transmit FIFO is locked until the XMR interrupt is read by the host (interrupt cannot be read if masked in MASKD).

XDU ... Transmit Data Underrun

The current transmission of a frame is aborted by transmitting seven ’1’s because the XFIFOD holds no further data. This interrupt occurs whenever the microcontroller has failed to respond to an XPR interrupt (ISTAD register) quickly enough, after having initiated a transmission and the message to be transmitted is not yet complete.

If an XDU interrupt occurs the transmit FIFO is locked until the XDU interrupt is read by the host (interrupt cannot be read if masked in MASKD).

5.8.1.4 MASKD - Mask Register D-Channel

Value after reset: FF_H

	7							0	
MASKD	RME	RPF	RFO	XPR	XMR	XDU	1	1	WR (20)

Each interrupt source in the ISTAD register can selectively be masked by setting the corresponding bit in MASKD to ’1’. Masked interrupt status bits are not indicated when ISTAD is read. Instead, they remain internally stored and pending until the mask bit is reset to ’0’.

5.8.1.5 STARD - Status Register D-Channel

Value after reset: 40_H

	7							0	
STARD	XDOV	XFW	0	0	RACI	0	XACI	0	RD (21)

XDOV ... Transmit Data Overflow

More than 16 or 32 bytes (according to selected block size) have been written to the XFIFOD, i.e. data has been overwritten.

XFW ... Transmit FIFO Write Enable

Data can be written to the XFIFOD. This bit may be polled instead of (or in addition to) using the XPR interrupt.

RACI ... Receiver Active Indication

The D-channel HDLC receiver is active when RACI = ’1’. This bit may be polled. The RACI bit is set active after a begin flag has been received and is reset after receiving an abort sequence.

XACI ... Transmitter Active Indication

The D-channel HDLC-transmitter is active when XACI = ’1’. This bit may be polled. The XACI-bit is active when an XTF-command is issued and the frame has not been completely transmitted

5.8.1.6 CMDRD - Command Register D-channel

Value after reset: 00_H

	7							0	
CMDRD	RMC	RRES	0	STI	XTF	0	XME	XRES	WR (21)

RMC ... Receive Message Complete

Reaction to RPF (Receive Pool Full) or RME (Receive Message End) interrupt. By setting this bit, the microcontroller confirms that it has fetched the data, and indicates that the corresponding space in the RFIFOD may be released.

RRES ... Receiver Reset

HDLC receiver is reset, the RFIFOD is cleared of any data.

STI ... Start Timer 2

The SIUC-BA timer 2 is started when STI is set to one. The timer is stopped by writing to the TIMR2 register.

Note: Timer 3 is controlled by the TIMR3 register only.

XTF ... Transmit Transparent Frame

After having written up to 16 or 32 bytes (EXMD1.XFBS) to the XFIFOD, the microcontroller initiates the transmission of a transparent frame by setting this bit to ’1’. The opening flag is automatically added to the message by the SIUC-BA (except in the extended transparent mode where no flags are used).

XME ... Transmit Message End

By setting this bit to ’1’ the microcontroller indicates that the data block written last to the XFIFOD completes the corresponding frame. The SIUC-BA terminates the transmission by appending the CRC (if EXMD1.XCRC=0) and the closing flag sequence to the data (except in the extended transparent mode where no such framing is used).

XRES ... Transmitter Reset

The D-channel HDLC transmitter is reset and the XFIFOD is cleared of any data. This command can be used by the microcontroller to abort a frame currently in transmission.

Note: After an XPR interrupt further data has to be written to the XFIFOD and the appropriate Transmit Command (XTF) has to be written to the CMDRD register again to continue transmission, when the current frame is not yet complete (see

also XPR in ISTAD).
During frame transmission, the 0-bit insertion according to the HDLC bit-stuffing mechanism is done automatically.

5.8.1.7 MODED - Mode Register

Value after reset: $C0_H$

	7							0	
MODED	MDS2	MDS1	MDS0	0	RAC	DIM2	DIM1	DIM0	RD/WR (22)

MDS2-0 ... Mode Select

Determines the message transfer mode of the HDLC controller, as follows:

MDS2-0	Mode	Number of Address Bytes	Address Comparison 1.Byte	Address Comparison 2.Byte	Remark
0 0 0	Reserved				
0 0 1	Reserved				
0 1 0	Non-Auto mode	1	TEI1,TEI2	–	One-byte address compare.
0 1 1	Non-Auto mode	2	SAP1,SAP2,SAPG	TEI1,TEI2,TEIG	Two-byte address compare.
1 0 0	Extended transparent mode				
1 1 0	Transparent mode 0	–	–	–	No address compare. All frames accepted.
1 1 1	Transparent mode 1	> 1	SAP1,SAP2,SAPG	–	High-byte address compare.
1 0 1	Transparent mode 2	> 1	–	TEI1,TEI2,TEIG	Low-byte address compare.

Note: SAP1, SAP2: two programmable address values for the first received address byte (in the case of an address field longer than 1 byte);
SAPG = fixed value FC / FE_H.
TEI1, TEI2: two programmable address values for the second (or the only, in the case of a one-byte address) received address byte; TEIG = fixed value FF_H

Two different methods of the high byte and/or low byte address comparison can be selected by setting SAP1.MHA and/or SAP2.MLA.

RAC ... Receiver Active

The D-channel HDLC receiver is activated when this bit is set to ’1’. If set to ’0’ the HDLC data is not evaluated in the receiver.

DIM2-0 ... Digital Interface Modes

These bits define the characteristics of the IOM Data Ports (DU, DD). The DIM0 bit enables/disables the collission detection. The DIM1 bit enables/disables the TIC bus access. The effect of the individual DIM bits is summarized in the table below**.**

DIM2	DIM1	DIM0	Characteristics
0		0	Transparent D-channel, the collission detection is disabled
0		1	Stop/go bit evaluated for D-channel access handling
0	0		Last octet of IOM channel 2 used for TIC bus access
0	1		TIC bus access is disabled
1	x	x	Reserved

5.8.1.8 EXMD1- Extended Mode Register D-channel 1

Value after reset: 00_H

	7						0	
EXMD1	XFBS	RFBS	SRA	XCRC	RCRC	0	ITF	RD/WR (23)

XFBS … Transmit FIFO Block Size

0 … Block size for the transmit FIFO data is 32 byte
1 … Block size for the transmit FIFO data is 16 byte

Note: A change of XFBS will take effect after a receiver command (CMDRD.XME, CMDRD.XRES, CMDRD.XTF) has been written.

RFBS … Receive FIFO Block Size

RFBS		Block Size Receive FIFO
Bit 6	Bit5	
0	0	32 byte
0	1	16 byte
1	0	8 byte
1	1	4 byte

Note: A change of RFBS will take effect after a transmitter command (CMDR.RMC, CMDR.RRES,) has been written

SRA … Store Receive Address

0 … Receive Address isn't stored in the RFIFOD
1 … Receive Address is stored in the RFIFOD

XCRC … Transmit CRC

0 … CRC is transmitted
1 … CRC isn't transmitted

RCRC… Receive CRC

0 … CRC isn't stored in the RFIFOD
1 … CRC is stored in the RFIFOD

ITF… Interframe Time Fill

Selects the inter-frame time fill signal which is transmitted between HDLC-frames.

0 … idle (continuous ’1’)
1 … flags (sequence of patterns: ‘0111 1110’)

Note: ITF must be set to ’0’ for power down mode.
In applications with D-channel access handling (collision resolution), the only possible inter-frame time fill is idle (continuous ’1’). Otherwise the D-channel on the S/T-bus cannot be accessed

5.8.1.9 TIMR2 - Timer 2 Register

Value after reset: 00_H

	7 5	4 0	
TIMR2	CNT	VALUE	RD/WR (24)

CNT ... Timer Counter

CNT together with VALUE determines the time period T after which a AUXI.TIN2 interrupt will be generated:

CNT=0...6: T = CNT x 2.048 sec + T1 with T1 = (VALUE+1) x 0.064 sec

CNT=7: T = T1 = (VALUE+1) x 0.064 sec (generated periodically)

The timer can be started by setting the STI-bit in CMDRD and will be stopped when a TIN2 interrupt is generated or the TIMR2 register is written.

Note: If CNT is set to 7, a TIN interrupt is indefinitely generated after every expiration of T1 (i.e. T = T1).

VALUE ... Timer Value

Determines the value of the timer value **T1 = (VALUE + 1) x 0.064 sec**.

5.8.1.10 SAP1 - SAPI1 Register

Value after reset: FC_H

	7			0	
SAP1	SAPI1		0	MHA	WR (25)

SAPI1 ... SAPI1 value

Value of the first programmable Service Access Point Identifier (SAPI) according to the ISDN LAPD protocol.

MHA... Mask High Address

0 …The SAPI address of an incomming frame is compared with SAP1, SAP2, SAPG.
1 …The SAPI address of an incomming frame is compared with SAP1 and SAPG.
SAP1 can be masked with SAP2 thereby bit positions of SAP1 are not compared if they are set to ’1’ in SAP2.

5.8.1.11 SAP2 - SAPI2 Register

Value after reset: FC_H

	7			0	
SAP2	SAPI2		0	MLA	WR (26)

SAPI2 ... SAPI2 value

Value of the second programmable Service Access Point Identifier (SAPI) according to the ISDN LAPD-protocol.

MLA... Mask Low Address

0 … The TEI address of an incomming frame is compared with TEI1, TEI2 and TEIG.
1 … The TEI address of an incomming frame is compared with TEI1 and TEIG.
TEI1 can be masked with TEI2 thereby bit positions of TEI1 are not compared if they are set to ’1’ in TEI2.

5.8.1.12 RBCLD - Receive Frame Byte Count Low D-Channel

Value after reset: 00_H

	7						0	
RBCLD	RBC7						RBC0	RD (26)

RBC7-0 ... Receive Byte Count

Eight least significant bits of the total number of bytes in a received message (see RBCHD register).

5.8.1.13 RBCHD - Receive Frame Byte Count High D-Channel

Value after reset: 00_H.

	7						0	
RBCHD	0	0	0	OV	RBC11		RBC8	RD (27)

OV ... Overflow

A ’1’ in this bit position indicates a message longer than $(2^{12} - 1) = 4095$ bytes .

RBC8-11 ... Receive Byte Count

Four most significant bits of the total number of bytes in a received message (see RBCLD register).

Note: Normally RBCHD and RBCLD should be read by the microcontroller after an RME-interrupt in order to determine the number of bytes to be read from the RFIFOD, and the total message length. The contents of the registers are valid only after an RME or RPF interrupt, and remain so until the frame is acknowledged via the RMC bit or RRES.

5.8.1.14 TEI1 - TEI1 Register 1

Value after reset: FF_H

	7			0	
TEI1	TEI1			EA1	WR (27)

TEI1 ... Terminal Endpoint Identifier

In all message transfer modes except in transparent modes 0, 1 and extended transparent mode, TEI1 is used by the SIUC-BA for address recognition. In the case of a two-byte address field, it contains the value of the first programmable Terminal Endpoint Identifier according to the ISDN LAPD-protocol.
In non-automodes with one-byte address field, TEI1 is a command address, according to X.25 LAPB.

EA1 ... Address field Extension bit

This bit is set to ’1’ according to HDLC/LAPD.

5.8.1.15 TEI2 - TEI2 Register

Value after reset: FF_H

	7			0	
TEI2	TEI2			EA2	WR (28)

TEI2 ... Terminal Endpoint Identifier

In all message transfer modes except in transparent modes 0, 1 and extended transparent mode, TEI2 is used by the SIUC-BA for address recognition. In the case of a two-byte address field, it contains the value of the second programmable Terminal Endpoint Identifier according of the ISDN LAPD-protocol.
In non-auto-modes with one-byte address field, TEI2 is a response address, according to X.25 LAPD.

EA2 ... Address field Extension bit

This bit is to be set to ’1’ according to HDLC/LAPD.

5.8.1.16 RSTAD - Receive Status Register D-Channel

Value after reset: $0F_H$

	7							0	
RSTAD	VFR	RDO	CRC	RAB	SA1	SA0	C/R	TA	RD (28)

For general information please refer to **Chapter 5.6**.

VFR... Valid Frame

Determines whether a valid frame has been received.
The frame is valid (1) or invalid (0). A frame is invalid when there is not a multiple of 8 bits between flag and frame end (flag, abort).

RDO ... Receive Data Overflow

If RDO=1, at least one byte of the frame has been lost, because it could not be stored in RFIFOD. As opposed to the ISTAD.RFO an RDO indicates that the beginning of a frame has been received but not all bytes could be stored as the RFIFOD was temporarily full.

CRC ... CRC Check

The CRC is correct (1) or incorrect (0).

RAB ... Receive Message Aborted

The receive message was aborted by the remote station (1), i.e. a sequence of seven 1’s was detected before a closing flag.

SA1-0 ... SAPI Address Identification
TA ... TEI Address Identification

SA1-0 are significant in non-automode with a two-byte address field, as well as in transparent mode 3. TA is significant in all modes except in transparent modes 0 and 1.

Two programmable SAPI values (SAP1, SAP2) plus a fixed group SAPI (SAPG of value FC_H/FE_H), and two programmable TEI values (TEI1, TEI2) plus a fixed group TEI (TEIG of value FF_H), are available for address comparison.

The result of the address comparison is given by SA1-0 and TA, as follows:

C/R ... Command/Response

The C/R bit contains the C/R bit of the received frame (Bit1 in the SAPI address).

Note: The contents of RSTAD corresponds to the last received HDLC frame; it is duplicated into RFIFOD for every frame (last byte of frame)

MDS2-0	Mode	SA1	SA0	TA	Address Match with 1st Byte	2nd Byte
010	Non-Auto/8 Mode	x x	x x	0 1	TEI2 TEI1	- -
011	Non-Auto/16 Mode	0 0 0 0 1 1	0 0 1 1 0 0	0 1 0 1 0 1	SAP2 SAP2 SAPG SAPG SAP1 SAP1	TEIG TEI2 TEIG TEI1 or TEI2 TEIG TEI1
111	Transparent Mode 1	0 0 1	0 1 0	x x x	SAP2 SAPG SAP1	- - -
101	Transparent Mode 2	- -	- -	0 1	- -	TEIG TEI1 or TEI2
		1	1	x	reserved	

Note: If SAP1 and SAP2 contain identical values, the combination SAP1/2-TEIG will only be indicated by SA1,0=”10” (i.e. the value “00” will not occur in this case).

5.8.1.17 TMD -Test Mode Register D-Channel

Value after reset: 00_H

	7							0	
TMD	0	0	0	0	0	0	0	TLP	RD/WR (29)

For general information please refer to **Chapter 5.2.11**.

TLP ... Test Loop

The TX path of layer-2 is internally connected with the RX path of layer-2. Data coming from the layer 1 controller will not be forwarded to the layer 2 controller.
The setting of TLP is only valid if the IOM interface is active.

5.8.1.18 CIR0 - Command/Indication Receive 0

Value after reset: $F3_H$

	7				0	
CIR0	CODR0	CIC0	CIC1	S/G	BAS	RD (2E)

CODR0 ... C/I Code 0 Receive

Value of the received Command/Indication code. A C/I-code is loaded in CODR0 only after being the same in two consecutive IOM-frames and the previous code has been read from CIR0.

CIC0 ... C/I Code 0 Change

A change in the received Command/Indication code has been recognized. This bit is set only when a new code is detected in two consecutive IOM-frames. It is reset by a read of CIR0.

CIC1 ... C/I Code 1 Change

A change in the received Command/Indication code in IOM-channel 1 has been recognized. This bit is set when a new code is detected in one IOM-frame. It is reset by a read of CIR0.

S/G ... Stop/Go Bit Monitoring

Indicates the availability of the upstream D-channel on the S/T interface.
1: Stop
0: Go

BAS ... Bus Access Status

Indicates the state of the TIC-bus:

0: the SIUC-BA itself occupies the D- and C/I-channel
1: another device occupies the D- and C/I-channel

Note: The CODR0 bits are updated every time a new C/I-code is detected in two consecutive IOM-frames. If several consecutive valid new codes are detected and CIR0 is not read, only the first and the last C/I code is made available in CIR0 at the first and second read of that register, respectively.

5.8.1.19 CIX0 - Command/Indication Transmit 0

Value after reset: FE_H

	7				0	
CIX0	CODX0	TBA2	TBA1	TBA0	BAC	WR (2E)

CODX0 ... C/I-Code 0 Transmit

Code to be transmitted in the C/I-channel 0.
The code is only transmitted if the TIC bus is occupied. If TIC bus is enabled but occupied by another device, only “1s” are transmitted.

TBA2-0 ... TIC Bus Address

Defines the individual address for the SIUC-BA on the IOM bus.

This address is used to access the C/I- and D-channel on the IOM interface.

Note: If only one device is liable to transmit in the C/I- and D-channels of the IOM it should always be given the address value ’7’.

BAC ... Bus Access Control

Only valid if the TIC-bus feature is enabled (MODED.DIM2-0).

If this bit is set, the SIUC-BA will try to access the TIC-bus to occupy the C/I-channel even if no D-channel frame has to be transmitted. It should be reset when the access has been completed to grant a similar access to other devices transmitting in that IOM-channel.

Note: Access is always granted by default to the SIUC-BA with TIC-Bus Address (TBA2-0, STCR register) ’7’, which has the lowest priority in a bus configuration.

5.8.1.20 CIR1 - Command/Indication Receive 1

Value after reset: FE_H

	7			0	
CIR1	CODR1		CICW	CI1E	RD (2F)

CODR1 ... C/I-Code 1 Receive

CICW, CI1E ... C/I-Channel Width, C/I-Channel 1 Interrupt Enable

These two bits contain the read back values from CIX1 register (see below).

5.8.1.21 CIX1 - Command/Indication Transmit 1

Value after reset: FE_H

	7			0	
CIX1	CODX1		CICW	CI1E	WR (2F)

CODX1 ... C/I-Code 1 Transmit

Bits 7-2 of C/I1-channel timeslot.

CICW... C/I-Channel Width

CICW selects between a 4 bit (’0’) and 6 bit (’1’) C/I1 channel width.

The C/I1 handler always reads and writes 6-bit values but if 4-bit is selected, the higher two bits are ignored for interrupt generation. However in write direction the full CODX1 code is transmitted, i.e. the host must write the higher two bits to “1”.

CI1E ... C/I-Channel 1 Interrupt Enable

Interrupt generation ISTA.CIC of CIR0.CIC1 is enabled (1) or masked (0).

5.8.2 Transceiver Registers

5.8.2.1 TR_CONF0 - Transceiver Configuration Register 0

Value after reset: 01_H

	7							0	
TR_CONF0	DIS_TR	0	EN_ICV	0	L1SW	0	EXLP	LDD	RD/WR (30)

DIS_TR ... Disable Transceiver

All layer-1 functions are disabled by setting DIS_TR = 1. The D- and B-channel HDLC controllers can still operate via IOM-2 while DCL and FSC pins become input.

In order to reenable the transceiver again, a reset to the transceiver must be issued (SRES.RES_TR = 1). The transceiver must not be reenabled by setting DIS_TR from “1” to “0”.

For general information please refer to **Chapter 5.2.10**.

EN_ICV ... Enable Illegal Code Violation

0: normal operation

1: ICV enabled. The receipt of at least one illegal code violation within one multiframe is indicated by the C/I indication ’1011’ (CVR) in two consecutive IOM frames.

L1SW ... Enable Layer 1 State Machine in Software

0: Layer 1 state machine of the SIUC-BA is used

1: Layer 1 state machine is disabled. The functionality can be realized in software. The commands can be written to register TR_CMD and the status can be read from TR_STA.

For general information please refer to **Chapter 5.3**.

EXLP ... External loop

In case the analog loopback is activated with C/I = ARL or with the LP_A bit in the TR_CMD register the loop is a

0: internal loop next to the line pins

1: external loop which has to be closed between SR1/2 and SX1/SX2

Note: The external loop is only useful if bit DIS_TX of register TR_CONF2 is set to ’0’.

For general information please refer to **Chapter 5.2.11**.

LDD ... Level Detection Discard

0: Automatic clock generation after detection of any signal on the line in power down state

1: No clock generation after detection of any signal on the line in power down state

Note: If an interrupt by the level detect circuitry is generated, the microcontroller has to set this bit to '0' for an activation of the S/T interface.

For general information please refer to **Chapter 5.2.9** and **Chapter 5.5.7**.

5.8.2.2 TR_CONF1 - Transceiver Configuration Register 1

Value after reset: $0x_H$

	7							0	
TR_CONF1	0	RPLL_ADJ	EN_SFSC	0	0	x	x	x	RD/WR (31)

RPLL_ADJ ... Receive PLL Adjustment

0: DPLL tracking step is 0.5 XTAL period per S-frame

1: DPLL tracking step is 1 XTAL period per S-frame

EN_SFSC ... Enable Short FSC

0: No short FSC is generated

1: A short FSC is generated once per multiframe (every 40th IOM frame)

x ... Undefined

The value of these bits depends on the selected mode. It is important to note that these bits must not be overwritten to a different value when accessing this register.

5.8.2.3 TR_CONF2 - Transmitter Configuration Register 2

Value after reset: 80_H

	7							0	
TR_CONF2	DIS_TX	PDS	0	RLP	0	0	SGP	SGD	RD/WR (32)

DIS_TX ... Disable Line Driver

0: Transmitter is enabled

1: Transmitter is disabled

For general information please refer to **Chapter 5.2.10**.

PDS ... Phase Deviation Select

Defines the phase deviation of the S-transmitter.
0: The phase deviation is 2 S-bits minus 7 oscillator periods plus analog delay plus delay of the external circuitry.
1: The phase deviation is 2 S-bits minus 9 oscillator periods plus analog delay plus delay of the external circuitry.

For general information please refer to **Chapter 5.2.8**.

RLP ... Remote Loop

0: Remote Loop open
1: Remote Loop closed
This test mode can also be programmed in TR_CMD.LP_A.

For general information please refer to **Chapter 5.2.11**.

SGP ... Stop/Go Bit Polarity

Defines the polarity of the S/G bit output on pin SGO (multiplexed function of AUX7).

0: low active (SGO=0 means “go”; SGO=1 means “stop”)

1: high active (SGO=1 means “go”; SGO=0 means “stop”)

SGD ... Stop/Go Bit Duration

Defines the duration of the S/G bit output on pin SGO (multiplexed function of AUX7).

0: active during the D-channel timeslot

1: active during the whole corresponding IOM frame (starts and ends with the beginning of the D-channel timeslot)

Note: Outside the active window of SGO (defined in SGD) the level on pin SGO remains in the “stop”-state depending on the selected polarity (SGP), i.e. SGO=1 (if SGP=0) or SGO=0 (if SGP=1) outside the active window.

5.8.2.4 TR_STA - Transceiver Status Register

Value after reset: 00_H

	7						0	
TR_STA	RINF	0	ICV	0	FSYN	0	LD	RD (33)

Important: This register is used only if the Layer 1 state machine of the SIUC-BA is disabled (TR_CONF0.L1SW = 1) and implemented in software! With the SIUC-BA layer 1 state machine enabled, the signals from this register are automatically evaluated.

For general information please refer to **Chapter 5.3**.

RINF ... Receiver INFO

00: Received INFO 0

01: Received any signal except INFO 1 - 4

10: Received INFO 2

11: Received INFO 4

ICV ... Illegal Code Violation

0: No illegal code violation is detected
1: Illegal code violation (ANSI T1.605) in data stream is detected

FSYN ... Frame Synchronization State

0: The S/T receiver is not synchronized

1: The S/T receiver has synchronized to the framing bit F

LD ... Level Detection

0: No receive signal has been detected on the line.
1: Any receive signal has been detected on the line.

5.8.2.5 TR_CMD - Transceiver Command Register

Value after reset: 08_H

	7					0	
TR_CMD	XINF	DPRIO	TDDIS	PD	LP_A	0	RD/WR (34)

Important: This register is used only if the Layer 1 state machine of the SIUC-BA is disabled (TR_CONF0.L1SW = 1) and implemented in software! With the SIUC-BA layer 1 state machine enabled, the signals from this register are automatically generated.

XINF ... Transmit INFO

000: Transmit INFO 0

001: reserved

010: Transmit INFO 1

011: Transmit INFO 3

100: Send continous pulses at 192 kbit/s alternating or 96 kHz rectangular, respectively (SCP)

101: Send single pulses at 4 kbit/s with alternating polarity corresponding to 2 kHz fundamental mode (SSP)

11x: reserved

DPRIO ... D-Channel Priority

0: Priority Class 1 for D channel access on S interface

1: Priority Class 2 for D channel access on S interface

TDDIS ... Transmit Data Disabled (TE mode)

0: The B and D channel data are transparently transmitted on the S/T interface if INFO 3 is being transmitted

1: The B and D channel data are set to logical ’1’ on the S/T interface if INFO 3 is being transmitted

PD ... Power Down

0: The transceiver is set to operational mode

1: The transceiver is set to power down mode

Note: This bit should not be used with active layer 1 state machine of the SIUC-BA to power down the device. Instead, the C/I commands should be used.

For general information please refer to **Chapter 5.3.1.2**.

LP_A ... Loop Analog

The setting of this bit corresponds to the C/I command ARL.

0:Analog loop is open

1:Analog loop is closed internally or externally according to the EXLP bit in the TR_CONF0 register

For general information please refer to **Chapter 5.2.11**.

5.8.2.6 SQRR1 - S/Q-Channel Receive Register 1

Value after reset: 40_H

	7							0	
SQRR	MSYN	MFEN	0	0	SQR11	SQR12	SQR13	SQR14	RD (35)

For general information please refer to **Chapter 5.2.2**.

MSYN ... Multiframe Synchronization State

0: The S/T receiver has not synchronized to the received F_A and M bits

1: The S/T receiver has synchronized to the received F_A and M bits

MFEN ... Multiframe Enable

Read-back of the MFEN bit of the SQXR register

SQR11-14 ... Received S Bits

Received S bits in frames 1, 6, 11 and 16 (TE mode).

5.8.2.7 SQXR1- S/Q-Channel TX Register 1

Value after reset: $4F_H$

	7							0	
SQXR1	0	MFEN	0	0	SQX11	SQX12	SQX13	SQX14	WR (35)

MFEN ... Multiframe Enable

Used to enable or disable the multiframe structure (see **Chapter 5.2.2**)

0: S/T multiframe is disabled
1: S/T multiframe is enabled
Readback value in SQRR1.

SQX11-14 ... Transmitted S/Q Bits

Transmitted Q bits (F_A bit position) in frames 1, 6, 11 and 16 (TE mode).

5.8.2.8 SQRR2 - S/Q-Channel Receive Register 2

Value after reset: 00_H

	7							0	
SQRR2	SQR21	SQR22	SQR23	SQR24	SQR31	SQR32	SQR33	SQR34	RD (36)

SQR21-24, SQR31-34... Received S Bits (TE mode only)

Received S bits in frames 2, 7, 12 and 17 (SQR21-24, subchannel 2), and in frames 3, 8, 13 and 18 (SQR31-34, subchannel 3).

5.8.2.9 SQRR3 - S/Q-Channel Receive Register 3

Value after reset: 00_H

	7							0	
SQRR3	SQR41	SQR42	SQR43	SQR44	SQR51	SQR52	SQR53	SQR54	RD (37)

SQR41-44, SQR51-54... Received S Bits (TE mode only)

Received S bits in frames 4, 9, 14 and 19 (SQR41-44, subchannel 4), and in frames 5, 10, 15 and 20 (SQR51-54, subchannel 5).

5.8.2.10 ISTATR - Interrupt Status Register Transceiver

Value after reset: 00_H

	7							0	
ISTATR	x	x	x	x	LD	RIC	SQC	SQW	RD (38)

For all interrupts in the ISTATR register the following logical states are defined:

0: Interrupt is not acitvated
1: Interrupt is acitvated

x ... Reserved

Bits set to “1” in this bit position must be ignored.

LD ... Level Detection

Any receive signal has been detected on the line. This bit is set to “1” (i.e. an interrupt is generated if not masked) as long as any receiver signal is detected on the line.

RIC ... Receiver INFO Change

RIC is activated if one of the TR_STA bits RINF or ICV has changed. This bit is reset by reading this register.

SQC ... S/Q-Channel Change

A change in the received S-channel has been detected. The new code can be read from the SQRxx bits of registers SQRR1-3 within the next multiframe. This bit is reset by a read access to the corresponding SQRRx register.

SQW ... S/Q-Channel Writable

The S/Q channel data for the next multiframe is writable.
The register for the Q (S) bits to be transmitted (received) has to be written (read) within the next multiframe. This bit is reset by writing register SQXRx.

This timing signal is indicated with the start of every multiframe. Data which is written right after SQW-indication will be transmitted with the start of the following multiframe. Data which is written before SQW-indication is transmitted in the multiframe which is indicated by SQW.

SQW and SQC could be generated at the same time.

5.8.2.11 MASKTR - Mask Transceiver Interrupt

Value after reset: FF_H

	7							0	
MASKTR	1	1	1	1	LD	RIC	SQC	SQW	RD/WR (39)

The transceiver interrupts LD, RIC, SQC and SQW are enabled (0) or disabled (1).

5.8.2.12 TR_MODE - Transceiver Mode Register 1

Value after reset: 00_H

	7						0	
TR_MODE	D_SLICE	0	0	0	MODE 2	MODE 1	MODE 0	RD/WR (3A)

D_SLICE ... D-channel Slice

Determines the 2-bit position of the D-channel within the selected octett.
00: bit 7 and 6 (default position on IOM-2 interface)
01: bit 5 and 4
10: bit 3 and 2
11: bit 1 and 0

Note: Shifting the D-channel to a different position may be usefull for special test purposes.

MODE2-0 ... Transceiver Mode

000: TE mode
$\overline{000}$: all other codes reserved

5.8.3 Auxiliary Interface Registers

5.8.3.1 ACFG1 - Auxiliary Configuration Register 1

Value after reset: 00_H

	7							0	
ACFG1	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0	RD/WR (3C)

For general information please refer to **Chapter 8.3.1**.

OD7-0 ... Output Driver Select for AUX7 - AUX0

0: output is open drain

1: output is push/pull

Note: The ODx configuration is only valid if the corresponding output is enabled in the AOE register.
AUX7 and AUX6 provide internal pull up resistors which are only available as inputs and in output/open drain mode, but disabled in output / push/pull mode.

5.8.3.2 ACFG2 - Auxiliary Configuration Register 2

Value after reset: 00_H

	7							0	
ACFG2	A7SEL	A5SEL	FBS	A4SEL	ACL	LED	EL2	EL1	RD/WR (3D)

A7SEL ... AUX7 Function Select

0: pin AUX7 provides normal I/O functionality.

1: pin AUX7 provides the S/G bit output (SGO) from the IOM DD-line. Bit AOE.OE7 is don't care, the output characteristic (push pull or open drain) can be selected via ACFG1.OD7.

A5SEL ... AUX5 Function Select

0: pin AUX5 provides normal I/O functionality.

1: pin AUX5 provides an FSC or BCL signal output (FBOUT) which is selected in ACFG2.FBS. Bit AOE.OE5 is don't care, the output characteristic (push pull or open drain) can be selected via ACFG1.OD5.

For general information please refer to **Chapter 8.1**.

FBS ... FSC/BCL Output Select

0: FSC is output on pin AUX5.

1: BCL (single bit clock) is output on pin AUX5.

Note: This selection has only effect on pin AUX5 if FBOUT is enabled (A5SEL=1).

For general information please refer to **Chapter 8.1**.

A4SEL ... AUX4 Function Select

0: pin AUX4 provides normal I/O functionality.

1: pin AUX4 supports multiframe synchronization and is used as M-bit output in TE mode. Bit AOE.OE4 is don't care, the output characteristic (push pull or open drain) can be selected via ACFG1.OD4.

For general information please refer to **Chapter 5.2.3**.

ACL ... ACL Function Select

0: Pin ACL automatically indicates the S-bus activation status by a LOW level.

1: The output state of ACL is programmable by the host in bit LED.

Note: An LED with preresistance may directly be connected to ACL.

LED ... LED Control

If enabled (ACL=1) the LED with preresistance connected between VDD and ACL is switched ...

0: Off (high level on pin ACL)

1: On (low level on pin ACL)

EL1, 2 ... Edge/Level Triggered Interrupt Input for $\overline{\text{INT1}}$, $\overline{\text{INT2}}$

0: A negative level ...

1: A negative edge ... on $\overline{\text{INT1/2}}$ (pins AUX6/7) generates an interrupt to the SIUC-BA.

Note: An interrupt is only generated if the corresponding mask bit in AUXM is reset. This configuration is only valid if the corresponding output enable bit in AOE is disabled.

For general information please refer to **Chapter 8.3.1**.

5.8.3.3 AOE - Auxiliary Output Enable Register

Value after reset: FF_H

	7							0	
AOE	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0	RD/WR (3E)

For general information please refer to **Chapter 8.3.1**.

OE7-0 ... Output Enable for AUX7 - AUX0

0: Pin AUX7-0 is configured as output. The value of the corresponding bit in the ATX register is driven on AUX7-0.

1: Pin AUX7-0 is configured as input. The value of the corresponding bit can be read from the ARX register.

Note: If pins AUX7, AUX6 are to be used as interrupt input, OE7, OE6 must be set to 1. If pins AUX7, AUX5 and AUX4 are not used as I/O pins (see ACFG2), the corresponding OEx bit cannot be set, but delivers the mode dependent direction (input/output) in that function upon a read access. If the secondary function is disabled, the direction of the pin as I/O pin is valid again.

5.8.3.4 ARX - Auxiliary Interface Receive Register

Value after reset: (not defined)

	7							0	
ARX	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0	RD (3F)

AR7-0 ... Auxiliary Receive

The value of AR7-0 always reflects the level at pin AUX7-0 at the time when ARX is read by the host even if a pin is configured as output. If the mask bit for AUX7, 6 is set in the MASKA register, no interrupt is generated to the SIUC-BA, however, the current state at pin AUX7,6 can be read from AR7,6

5.8.3.5 ATX - Auxiliary Interface Transmit Register

Value after reset: 00_H

	7							0	
ATX	AT7	AT6	AT5	AT4	AT3	AT2	AT1	AT0	WR (3F)

AT7-0 ... Auxiliary Transmit

A ’0’ or ’1’ in AT7-0 will drive a low or a high level at pin AUX7-0 if the corresponding output is enabled in the AOE register.

5.8.4 IOM-2 and MONITOR Handler

5.8.4.1 CDAxy - Controller Data Access Register xy

	7							0	
CDAxy	Controller Data Access Register								RD/WR (40-43)

Data registers CDAxy which can be accessed from the controller.

Register	Register Address	Value after Reset
CDA10	40_H	FF_H
CDA11	41_H	FF_H
CDA20	42_H	FF_H
CDA21	43_H	FF_H

5.8.4.2 XXX_TSDPxy - Time Slot and Data Port Selection for CHxy

	7				0	
XXX_ TSDPxy	DPS	0	0	TSS		RD/WR (44-4D)

Register	Register Address	Value after Reset
CDA_TSDP10	44_H	00_H (= output on B1-DD)
CDA_TSDP11	45_H	01_H (= output on B2-DD)
CDA_TSDP20	46_H	80_H (= output on B1-DU)
CDA_TSDP21	47_H	81_H (= output on B2-DU)
BCHA_TSDP_BC1	48_H	80_H (= output on B1-DU)
BCHA_TSDP_BC2	49_H	81_H (= output on B2-DU)
BCHB_TSDP_BC1	$4A_H$	81_H (= output on B2-DU)
BCHB_TSDP_BC2	$4B_H$	85_H (= output on IC2-DU)
TR_TSDP_BC1	$4C_H$	00_H (= transceiver output on B1-DD)
TR_TSDP_BC2	$4D_H$	01_H (= transceiver output on B2-DD)

This register determines the time slots and the data ports on the IOM-2 interface for the data channels ’xy’ of the functional units ’XXX’ which are Controller Data Access (CDA), B-channel controllers (BCHA, BCHB) and Transceiver (TR).

Each of the two B-channel controllers (BCHA, BCHB) can access any combination of two 8-bit timeslots and one 2-bit timeslot (e.g. 16-bit access to B1+B2 or 18-bit IDSL in 2B+D). The position of the two 8-bit timeslots is programmed in BCHx_TSDP_BC1 and BCHx_TSDP_BC2. The position of the 2-bit timeslot is set to the D-timeslot of one IOM channel programmed in BCHA_CR and BCHB_CR. In the same registers each of the three timeslots is enabled/disabled.

The position of B-channel data from the S-interface is programmed in TR_TSDP_BC1 and TR_TSDP_BC2.

DPS ... Data Port Selection

0: The data channel xy of the functional unit XXX is output on DD.
The data channel xy of the functional unit XXX is input from DU.
1: The data channel xy of the functional unit XXX is output on DU.
The data channel xy of the functional unit XXX is input from DD.

Note: For the CDA (controller data access) data the input is determined by the CDA_CRx.SWAP bit. If SWAP = ’0’ the input for the CDAxy data is vice versa to the output setting for CDAxy. If the SWAP = ’1’ the input from CDAx0 is vice versa to the output setting of CDAx1 and the input from CDAx1 is vice versa to the output setting of CDAx0. See controller data access description in **Chapter 5.5.1.1**

TSS ... Timeslot Selection

Selects one of 32 timeslots (0...31) on the IOM-2 interface for the data channels.

5.8.4.3 CDAx_CR - Control Register Controller Data Access CH1x

	7							0	
CDAx_CR	0	0	EN_TBM	EN_I1	EN_I0	EN_O1	EN_O0	SWAP	RD/WR (4E-4F)

Register	Register Address	Value after Reset
CDA1_CR	$4E_H$	00_H
CDA2_CR	$4F_H$	00_H

For general information please refer to **Chapter 5.5.1.1**.

EN_TBM ... Enable TIC Bus Monitoring

0: The TIC bus monitoring is disabled

1: The TIC bus monitoring with the CDAx0 register is enabled. The TSDPx0 register must be set to 08_H for monitoring from DU or 88_H for monitoring from DD, respectively.

EN_I1, EN_I0 ... Enable Input CDAx0, CDAx1

0: The input of the CDAx0, CDAx1 register is disabled

1: The input of the CDAx0, CDAx1 register is enabled

EN_O1, EN_O0 ... Enable Output CDAx0, CDAx1

0: The output of the CDAx0, CDAx1 register is disabled

1: The output of the CDAx0, CDAx1 register is enabled

SWAP ... Swap Inputs

0: The time slot and data port for the input of the CDAxy register is defined by its own TSDPxy register. The data port for the CDAxy input is vice versa to the output setting for CDAxy.

1: The input (time slot and data port) of the CDAx0 is defined by the TSDP register of CDAx1 and the input of CDAx1 is defined by the TSDP register of CDAx0. The data port for the CDAx0 input is vice versa to the output setting for CDAx1. The data port for the CDAx1 input is vice versa to the output setting for CDAx0. The input definition for time slot and data port CDAx0 are thus swapped to CDAx1 and for CDAx1 to CDAx0. The outputs are not affected by the SWAP bit.

5.8.4.4 TR_CR - Control Register Transceiver Data (IOM_CR.CI_CS=0)

Value after reset: $F8_H$

	7					0	
TR_CR	EN_ D	EN_ B2R	EN_ B1R	EN_ B2X	EN_ B1X	CS2-0	RD/WR (50)

Read and write access to this register is only possible if IOM_CR.CI_CS=0.

EN_D ... Enable D-Channel Data
EN_B2R ... Enable B2 Receive Data (transceiver receives from IOM)
EN_B1R ... Enable B1 Receive Data (transceiver receives from IOM)
EN_B2X ... Enable B2 Transmit Data (transceiver transmits to IOM)
EN_B1X ... Enable B1 Transmit Data (transceiver transmits to IOM)

This register is used to individually enable/disable the D-channel (both, RX and TX direction) and the receive/transmit paths for the B-channels for the S-transceiver.

0: The corresponding data path to the transceiver is disabled.

1: The corresponding data path to the transceiver is enabled.

CS2-0 ... Channel Select for Transceiver D-channel

This register is used to select one of eight IOM channels to which the transceiver D and C/I channel data are related to (also see register TRC_CR below).

Note: It should be noted that writing TR_CR.CS2-0 will also write to TRC_CR.CS2-0 and therefore modify the channel selection for the transceiver C/I data.

5.8.4.5 TRC_CR - Control Register Transceiver C/I (IOM_CR.CI_CS=1)

Value after reset: 00_H

	7					0	
TRC_CR	0	0	0	0	0	CS2-0	RD/WR (50)

Write access to this register is possible if IOM_CR.CI_CS = 0 or IOM_CR.CI_CS = 1.
Read access to this register is possible only if IOM_CR.CI_CS = 1.

CS2-0 ... Channel Select for the Transceiver C/I Channel

This register is used to select one of eight IOM channels to which the transceiver C/I channel are related to.

5.8.4.6 BCHx_CR - Control Register B-Channel Controller Data

	7					0	
BCHx_CR	DPS_D	0	EN_D	EN_ BC2	EN_ BC1	CS2-0	RD/WR (51,52)

Register	Register Address	Value after Reset
BCHA_CR	51_H	08_H
BCHB_CR	52_H	81_H

The registers BCHA_TSDP_BC1/2 and BCHB_TSDP_BC1/2 (see above) select the IOM-2 timeslots for B-channel access. For each of the B-channel controllers (BCHA, BCHB) two 8-bit timeslots can be selected (position and direction).

This register BCHx_CR is used to select the position and direction of the 2-bit timeslot for each of the two B-channel controllers and each of the three selected timeslots (2 x 8-bit and 2-bit) is individually enabled/disabled.

DPS ... Data Port Selection for D-Channel Timeslot access

0: The B-channel controller data is output on DD.
The B-channel controller data is input from DU.
1: The B-channel controller data is output on DU.
The B-channel controller data is input from DD.

EN_D ... Enable D-Channel Timeslot (2-bit) for B-Channel controller access
EN_BC2 ... Enable B2-Channel Timeslot (8-bit) for B-Channel controller access
EN_BC1 ... Enable B1-Channel Timeslot (8-bit) for B-Channel controller access

These bits individually enable/disable the B-channel access to the 2-bit and the two 8-bit timeslots.

0: B-channel B/A does not access timeslot data B1, B2 or D, respectively.

1: B-channel B/A does access timeslot data B1, B2 or D, respectively.

Note: The terms B1/B2 should not imply that the 8-bit timeslots must be located in the first/second IOM-2 timeslots, it's simply a placeholder for the 8-bit timeslot position selected in the registers BCHA_TSDP_BC1/2 and BCHB_TSDP_BC1/2.

CS2-0 ... Channel Select

This register is used to select one of eight IOM channels. If enabled (EN_D=1), the B-channel controller is connected to the 2-bit D-channel timeslot of that IOM channel.

5.8.4.7 DCI_CR - Control Register for D and CI1 Handler (CI_CS=0)

Value after reset: $A0_H$

	7					0	
DCI_CR	DPS_ CI1	EN_ CI1	D_ EN_D	D_ EN_B2	D_ EN_B1	CS2-0	RD/WR (53)

Read and write access to this register is possible only if IOM_CR.CI_CS = 0.

DPS_CI1 ... Data Port Selection CI1 Handler

0: The CI1 data is output on DD and input from DU

1: The CI1 data is output on DU and input from DD

EN_CI1 ... Enable CI1 Handler

0: CI1 data access is disabled

1: CI1 data access is enabled

Note: The timeslot for the C/I1 handler cannot be programmed but is fixed to IOM channel 1.

D_EN_D ... Enable D-timeslot for D-channel controller
D_EN_B2 ... Enable B2-timeslot for D-channel controller
D_EN_B1 ... Enable B1-timeslot for D-channel controller

These bits are used to select the timeslot length for the D-channel HDLC controller as it is capable to access not only the D-channel timeslot. The host can individually enable two 8-bit timeslots B1- and B2-channel (D_EN_B1, D_EN_B2) and one 2-bit timeslot D-channel (D_EN_D) on IOM-2. The position is selected via CS2-0.

0: D-channel controller does not access timeslot data B1, B2 or D, respectively

1: D-channel controller does access timeslot data B1, B2 or D, respectively

For D-channel HDLC only (not for B-channel HDLC) the position of the two 8-bit timeslots is fixed to the first and second octett of the selected IOM channel.

CS2-0 ... Channel Select for D-channel controller

This register is used to select one of eight IOM channels. If enabled, the D-channel data is connected to the corresponding timeslot of that IOM channel.

Note: It should be noted that writing DCI_CR.CS2-0 will also write to DCIC_CR.CS2-0 and therefore modify the channel selection for the data of the C/I0 handler.

5.8.4.8 DCIC_CR - Control Register for CI0 Handler (CI_CS=1)

Value after reset: 00_H

	7					0	
DCIC_CR	0	0	0	0	0	CS2-0	RD/WR (53)

Write access to this register is possible if IOM_CR.CI_CS = 0 or IOM_CR.CI_CS = 1.
Read access to this register is possible only if IOM_CR.CI_CS = 1.

CS2-0 ... Channel Select for C/I0 Handler

This register is used to select one of eight IOM channels. If enabled, the data of the C/I0 handler is connected to the corresponding timeslots of that IOM channel.

5.8.4.9 MON_CR - Control Register Monitor Data

Value after reset: 40_H

	7					0	
MON_CR	DPS	EN_ MON	0	0	0	CS2-0	RD/WR (54)

For general information please refer to **Chapter 5.5.4**.

DPS ... Data Port Selection

0: The Monitor data is output on DD and input from DU

1: The Monitor data is output on DU and input from DD

EN_MON ... Enable Output

0: The Monitor data input and output is disabled

1: The Monitor data input and output is enabled

CS2-0 ... MONITOR Channel Selection

000: The MONITOR data is input/output on MON0 (3rd timeslot on IOM-2)

001: The MONITOR data is input/output on MON1 (7th timeslot on IOM-2)

010: The MONITOR data is input/output on MON2 (11th timeslot on IOM-2)

:

111: The MONITOR data is input/output on MON7 (31st timeslot on IOM-2)

5.8.4.10 SDS_CR - Control Register Serial Data Strobe

Value after reset: 00_H

	7				0	
SDS_CR	ENS_ TSS	ENS_ TSS+1	ENS_ TSS+3	TSS		RD/WR (55)

This register is used to select position and length of the strobe signal. The length can be any combination of two 8-bit timeslot (ENS_TSS, ENS_TSS+1) and one 2-bit timeslot (ENS_TSS+3).

For general information please refer to **Chapter 5.5.3** and **Chapter 5.5.3.2**.

ENS_TSS ... Enable Serial Data Strobe of timeslot TSS
ENS_TSS+1 ... Enable Serial Data Strobe of timeslot TSS+1

0: The serial data strobe signal SDS is inactive during TSS, TSS+1

1: The serial data strobe signal SDS is active during TSS, TSS+1

ENS_TSS+3 ... Enable Serial Data Strobe of timeslot TSS+3 (D-Channel)

0: The serial data strobe signal SDS is inactive during the D-channel (bit7, 6) of TSS+3

1: The serial data strobe signal SDS is active during the D-channel (bit7, 6) of TSS+3

TSS ... Timeslot Selection

Selects one of 32 timeslots on the IOM-2 interface (with respect to FSC) during which SDS is active high or provides a strobed BCL clock output (see SDS_CONF.SDS_BCL). The data strobe signal allows standard data devices to access a programmable channel.

5.8.4.11 IOM_CR - Control Register IOM Data

Value after reset: 08_H

	7							0	
IOM_CR	SPU	0	CI_CS	TIC_ DIS	EN_ BCL	CLKM	DIS_ OD	DIS_ IOM	RD/WR (57)

SPU ... Software Power Up

0: The DU line is normally used for transmitting data

1: Setting this bit to '1' will pull the DU line to low. This will enforce connected layer 1 devices to deliver IOM-clocking.

After a subsequent ISTA.CIC-interrupt (C/I-code change) and reception of the C/I-code "PU" (Power Up indication in TE-mode) the microcontroller writes an AR or TIM command as C/I-code in the CIX0-register, resets the SPU bit and waits for the following CIC-interrupt.

For general information please refer to **Chapter 5.5.7**.

CI_CS ... C/I Channel Selection

The channel selection for D-channel and C/I-channel is done in the channel select bits CH2-0 of register TR_CR (for the transceiver) and DCI_CR (for the D-channel controller and C/I-channel controller).

0: A write access to CS2-0 has effect on the configuration of D- and C/I-channel, whereas a read access delivers the D-channel configuration only.

1: A write access to CS2-0 has effect on the configuration of the C/I-channel only, whereas a read access delivers the C/I-channel configuration only.

TIC_DIS ... TIC Bus Disable

0: The last octet of IOM channel 2 (12th timeslot) is used as TIC bus (TE mode only).

1: The TIC bus is disabled. The last octet of the last IOM time slot (TS 11) can be used as every time slot. This means that the timeslots TIC, A/B, S/G and BAC are not available any more.

EN_BCL ... Enable Bit Clock BCL/SCLK

0: The BCL/SCLK clock is disabled
1: The BCL/SCLK clock is enabled.

CLKM ... Clock Mode

If the transceiver is disabled (DIS_TR = ’1’) the DCL from the IOM-2 interface is an input.

0: A double bit clock is connected to DCL

1: A single bit clock is connected to DCL

For general information please refer to **Chapter 5.5**.

DIS_OD ... Disable Open Drain Drivers

0: DU/DD are open drain drivers

1: DU/DD are push pull drivers

DIS_IOM ... Disable IOM

DIS_IOM should be set to ’1’ if external devices connected to the IOM interface should be “disconnected“ e.g. for power saving purposes or for not disturbing the internal IOM connection between layer 1 and layer 2. However, the SIUC-BA internal operation between S-transceiver, B-channel and D-channel controller is independent of the DIS_IOM bit.

0: The IOM interface is enabled

1: The IOM interface is disabled (FSC, DCL clock outputs have high impedance; clock inputs are active; DU, DD data line inputs are switched off and outputs have high impedance)

5.8.4.12 STI - Synchronous Transfer Interrupt

Value after reset: 00_H

	7							0	
STI	STOV 21	STOV 20	STOV 11	STOV 10	STI 21	STI 20	STI 11	STI 10	RD (58)

For all interrupts in the STI register the following logical states are applied:

0: Interrupt is not activated

1: Interrupt is activated

For general information please refer to **Chapter 5.5.1.1**.

STOVxy ... Synchronous Transfer Overflow Interrupt

Enabled STOV interrupts for a certain STIxy interrupt are generated when the STIxy has not been acknowledged in time via the ACKxy bit in the ASTI register. This must be one (for DPS='0') or zero (for DPS='1') BCL clocks before the time slot which is selected for the STOV.

STIxy ... Synchronous Transfer Interrupt

Depending on the DPS bit in the corresponding TSDPxy register the Synchronous Transfer Interrupt STIxy is generated two (for DPS='0') or one (for DPS='1') BCL clock after the selected time slot (TSDPxy.TSS).

Note: ST0Vxy and ACKxy are useful for synchronizing microcontroller accesses and receive/transmit operations. One BCL clock is equivalent to two DCL clock cycles.

5.8.4.13 ASTI - Acknowledge Synchronous Transfer Interrupt

Value after reset: 00_H

	7							0	
ASTI	0	0	0	0	ACK 21	ACK 20	ACK 11	ACK 10	WR (58)

For general information please refer to **Chapter 5.5.1.1**.

ACKxy ... Acknowledge Synchronous Transfer Interrupt

After an STIxy interrupt the microcontroller has to acknowledge the interrupt by setting the corresponding ACKxy bit to "1".

5.8.4.14 MSTI - Mask Synchronous Transfer Interrupt

Value after reset: FF_H

	7							0	
MSTI	STOV 21	STOV 20	STOV 11	STOV 10	STI 21	STI 20	STI 11	STI 10	RD/WR (59)

For the MSTI register the following logical states are applied:

0: Interrupt is not masked
1: Interrupt is masked

For general information please refer to **Chapter 5.5.1.1**.

STOVxy ... Synchronous Transfer Overflow for STIxy

Mask bits for the corresponding STOVxy interrupt bits.

STIxy ... Synchronous Transfer Interrupt xy

Mask bits for the corresponding STIxy interrupt bits.

5.8.4.15 SDS_CONF - Configuration Register for Serial Data Strobes

Value after reset: 00_H

	7							0	
SDS_CONF	0	0	0	0	DIOM_INV	DIOM_SDS	0	SDS_BCL	RD/WR (5A)

For general information on SDS_BCL please refer to **Chapter 5.5.3**.

DIOM_INV ... DU/DD on IOM Timeslot Inverted

0:DU/DD are active during SDS HIGH phase and inactive during the LOW phase.
1:DU/DD are active during SDS LOW phase and inactive during the HIGH phase.
This bit has only effect if DIOM_SDS is set to ’1’ otherwise DIOM_INV is don’t care.

DIOM_SDS ... DU/DD on IOM Controlled via SDS

0: The pin SDS and its configuration settings are used for serial data strobe only. The IOM-2 data lines are not affected.
1: The DU/DD lines are deactivated during the during High/Low phase (selected via DIOM_INV) of the SDS signal. The SDS timeslot is selected in SDS_CR.

SDS_BCL ... Enable IOM Bit Clock for SDS

0: The serial data strobe is generated in the programmed timeslot.
1: The IOM bit clock is generated in the programmed timeslot.

5.8.4.16 MCDA - Monitoring CDA Bits

Value after reset: FF_H

	7							0	
MCDA	MCDA21		MCDA20		MCDA11		MCDA10		RD (5B)
	Bit7	Bit6	Bit7	Bit6	Bit7	Bit6	Bit7	Bit6	

MCDAxy ... Monitoring CDAxy Bits

Bit 7 and Bit 6 of the CDAxy registers are mapped into the MCDA register.

This can be used for monitoring the D-channel bits on DU and DD and the 'Echo bits' on the TIC bus with the same register

5.8.4.17 MOR - MONITOR Receive Channel

Value after reset: FF_H

	7							0	
MOR	Monitor Receiver Data								RD (5C)

Contains the MONITOR data received in the IOM-2 MONITOR channel according to the MONITOR channel protocol. The MONITOR channel (0-7) can be selected by setting the monitor channel select bit MON_CR.MCS.

5.8.4.18 MOX - MONITOR Transmit Channel

Value after reset: FF_H

	7							0	
MOX	Monitor Transmit Data								WR (5C)

Contains the MONITOR data to be transmitted in IOM-2 MONITOR channel according to the MONITOR channel protocol.The MONITOR channel (0-7) can be selected by setting the monitor channel select bit MON_CR.MCS.

5.8.4.19 MOSR - MONITOR Interrupt Status Register

Value after reset: 00_H

	7							0	
MOSR	MDR	MER	MDA	MAB	0	0	0	0	RD (5D)

MDR ... MONITOR channel Data Received

MER ... MONITOR channel End of Reception

MDA ... MONITOR channel Data Acknowledged

The remote end has acknowledged the MONITOR byte being transmitted.

MAB ... MONITOR channel Data Abort

5.8.4.20 MOCR - MONITOR Control Register

Value after reset: 00_H

	7							0	
MOCR	MRE	MRC	MIE	MXC	0	0	0	0	RD/WR (5E)

MRE ... MONITOR Receive Interrupt Enable

0: MONITOR interrupt status MDR generation is masked

1: MONITOR interrupt status MDR generation is enabled

MRC ... MR Bit Control

Determines the value of the MR bit:

0: MR is always ’1’. In addition, the MDR interrupt is blocked, except for the first byte of a packet (if MRE = 1).

1: MR is internally controlled by the SIUC-BA according to MONITOR channel protocol. In addition, the MDR interrupt is enabled for all received bytes according to the MONITOR channel protocol (if MRE = 1).

MIE ... MONITOR Interrupt Enable

MONITOR interrupt status MER, MDA, MAB generation is enabled (1) or masked (0).

MXC ... MX Bit Control

Determines the value of the MX bit:
0:The MX bit is always ’1’.

1:The MX bit is internally controlled by the SIUC-BA according to MONITOR channel protocol.

5.8.4.21 MSTA - MONITOR Status Register

Value after reset: 00_H

MSTA	0	0	0	0	0	MAC	0	TOUT	RD (5F)

MAC ... MONITOR Transmit Channel Active

The data transmisson in the MONITOR channel is in progress.

TOUT ... Time-Out

Read-back value of the TOUT bit.

5.8.4.22 MCONF - MONITOR Configuration Register

Value after reset: 00_H

MCONF	0	0	0	0	0	0	0	TOUT	WR (5F)

TOUT... Time-Out

0: The monitor time-out function is disabled

1: The monitor time-out function is enabled

5.8.5 Interrupt and General Configuration

5.8.5.1 ISTA - Interrupt Status Register

Value after reset: 00_H

	7							0	
ISTA	ICA	ICB	ST	CIC	AUX	TRAN	MOS	ICD	RD (60)

For all interrupts in the ISTA register following logical states are applied:

0: Interrupt is not acitvated
1: Interrupt is acitvated

ICA, ICB, ICD ... HDLC Interrupt from B-channel A, B or D-channel

An interrupt originated from the HDLC controllers of B-channel A, B or of the D-channel has been recognized.

ST ... Synchronous Transfer

This interrupt is generated to enable the microcontroller to lock on to the IOM timing for synchronous transfers. The source can be read from the STI register.

CIC ... C/I Channel Change

A change in C/I channel 0 or C/I channel 1 has been recognized. The actual value can be read from CIR0 or CIR1.

AUX ... Auxiliary Interrupts

Singals an interrupt generated from external awake (pin $\overline{EAW}$), watchdog timer overflow, timer2, timer3 or from one of the interrupt input pins ($\overline{INT1}$, $\overline{INT2}$). The source can be read from the auxiliary interrupt register AUXI.

TRAN ... Transceiver Interrupt

An interrupt originated in the transceiver interrupt status register (ISTATR) has been recognized.

MOS ... MONITOR Status

A change in the MONITOR Status Register (MOSR) has occured.

Note: A read of the ISTA register clears none of the interrupts. They are only cleared by reading the corresponding status register.

5.8.5.2 ISTA_INIT - Interrupt Status Register Initialize

Value after reset: FF_H

	7						0	
ISTA_INIT								WR (60)

After reset all interrupts from the ISDN module are disabled. In order to enable the ISDN Interrrupt Status register (ISTA) to generate interrupts to the microcontroller, the ISTA_INIT register must be set to 00_H. This value should not be changed again afterwards during normal operation.

Enabling and disabling of certain ISDN interrupts from ISTA should be done in the interrupt enable registers of the microcontroller IEN0, IEN1 and IEN2 (see **Chapter 6.1.2**).

5.8.5.3 AUXI - Auxiliary Interrupt Status Register

Value after reset: 00_H

	7							0	
AUXI	0	0	EAW	WOV	TIN3	TIN2	INT2	INT1	RD (61)

For all interrupts in the ISTA register following logical states are applied:

0: Interrupt is not acitvated
1: Interrupt is acitvated

EAW ... External Awake Interrupt

An interrupt from the $\overline{EAW}$ pin has been detected.

WOV ... Watchdog Timer Overflow

Signals the expiration of the watchdog timer, which means that the microcontroller has failed to set the watchdog timer control bits WTC1 and WTC2 (MODE1 register) in the correct manner. A reset pulse has been generated by the SIUC-X.

TIN3, 2 ... Timer Interrupt 3, 2

An interrupt originated from timer 2 or timer 3 is recognized, i.e the timer has expired.

INT2, 1 ... Auxiliary Interrupt from external devices 2, 1

A low level or a negative state transition (programmable in ACFG2.EL2/1) is detected at pin AUX7 or AUX6, respectively.

5.8.5.4 AUXM - Auxiliary Mask Register

Value after reset: FF_H

	7							0	
AUXM	1	1	EAW	WOV	TIN3	TIN2	INT2	INT1	WR (61)

For the Auxiliary MASK register following logical states are applied:

0: Interrupt is enabled
1: Interrupt is disabled

Each interrupt source in the AUXI register can selectively be masked/disabled by setting the corresponding bit in AUXM to ’1’. Masked interrupt status bits are not indicated when AUXI is read. Instead, they remain internally stored and pending, until the mask bit is reset to ’0’.

5.8.5.5 MODE1 - Mode1 Register

Value after reset: 00_H

	7							0	
MODE1	0	0	0	WTC1	WTC2	CFS	RSS2	RSS1	RD/WR (62)

WTC1, 2 ... Watchdog Timer Control 1, 2

After the watchdog timer mode has been selected (RSS = ’11’) the watchdog timer is started. During every time period of 128 ms the microcontroller has to program the WTC1 and WTC2 bit in the following sequence

	WTC1	WTC2
1.	1	0
2.	0	1

to reset and restart the watchdog timer, i.e. in order to reset the timer again the µC has to make 2 write accesses to WTC1,2 after every 128 ms period.

If WTC1/2 is not written fast enough in this way, the timer expires and a WOV-interrupt (AUXI register) together with a reset pulse is generated.

CFS ... Configuration Select

This bit determines clock relations and recovery on S/T and IOM interfaces.

0: The IOM interface clock and frame signals are always active, "Power Down" state included.
The states "Power Down" and "Power Up" are thus functionally identical except for the indication: PD = 1111 and PU = 0111.
With the C/I command Timing (TIM) the microcontroller can enforce the "Power Up" state and with C/I command Deactivation Indication (DI) the "Power Down" state is reached again.
However, it is also possible to activate the S-interface directly with the C/I command Activate Request (AR 8/10/L) without the TIM command.

1: The IOM interface clock and frame signals are normally inactive ("Power Down").
For activating the IOM-2 clocks the "Power Up" state can be induced by software (IOM_CR.SPU) or by resetting CFS again.
After that the S-interface can be activated with the C/I command Activate Request (AR 8/10/L). The "Power Down" state can be reached again with the C/I command Deactivation Indication (DI).

Note: After reset the IOM interface is always active. To reach the "Power Down" state the CFS-bit has to be set.

For general information please refer to **Chapter 5.2.9**.

RSS2, RSS1... Reset Source Selection 2,1

The SIUC-BA reset sources and the SDS functionality for the SDS/$\overline{\text{RSTO}}$ output pin can be selected according to the table below.

RSS		C/I Code Change	EAW	Watchdog Timer	SDS Functionality
Bit 1	Bit 0				
0	0	--	--	--	--
0	1	--	--	--	x
1	0	x	x	--	--
1	1	--	--	x	--

- If RSS = ’00’ no above listed reset source is selected and therefore no reset is generated at SDS/$\overline{\text{RSTO}}$.
- If RSS = ’01’ the SDS/$\overline{\text{RSTO}}$ pin has SDS functionality and a serial data strobe signal is output at the SDS/$\overline{\text{RSTO}}$ pin. In this mode no reset is output at SDS/$\overline{\text{RSTO}}$.
- **Watchdog Timer**
  After the selection of the watchdog timer (RSS = ’11’) the timer is reset and started. During every time period of 128 ms the microcontroller has to program the WTC1 and WTC2 bits in two consecutive bit pattern (see description above of the WTC1, 2 bits) otherwise the watchdog timer expires and a reset pulse of 125 µs ≤ t≤ 250 µs is generated. Once RSS1,2 is programmed to ’11’ the value cannot be reprogrammed to any other value, i.e. deactivation of the watchdog timer is only possible with a hardware reset.
- If RSS = ’10’ is selected the following two reset sources generate a reset pulse of 125 µs ≤ t ≤ 250µs at the SDS/$\overline{\text{RSTO}}$ pin:
  - External (Subscriber) Awake ($\overline{\text{EAW}}$)
  The **$\overline{\text{EAW}}$** input pin serves as a request signal from the subscriber to initiate the awake function in a terminal and generates a reset pulse (in TE mode only).
  - Exchange Awake (C/I Code)
  A C/I Code change generates a reset pulse.

After a reset pulse generated by the SIUC-BA and the corresponding interrupt (WOV or CIC) the actual reset source can be read from the ISTA.

5.8.5.6 ID - Identification Register

Value after reset: 01_H

	7			0	
ID	0	0	DESIGN		RD (64)

DESIGN ... Design Number

The design number allows to identify different hardware designs of the SIUC-BA by software.

01_H: Version 1.3

(all other codes reserved)

5.8.5.7 SRES - Software Reset Register

Value after reset: 00_H

	7							0	
SRES	RES_ CI	RES_ BCHA	RES_ BCHB	RES_ MON	RES_ DCH	RES_ IOM	RES_ TR	RES_ RSTO	WR (64)

RES_xx ... Reset Functional Block xx

A reset can be activated on the functional block C/I-handler, B-channel A and B, Monitor channel, D-channel, IOM handler, S-transceiver and to pin $\overline{\text{RSTO}}$.

Setting one of these bits to “1” causes the corresponding block to be reset for a duration of 4 BCL clock cycles, except RES_RSTO which is activated for a duration of 125 ... 250µs. The bits are automatically reset to “0” again.

5.8.5.8 TIMR3 - Timer 3 Register

Value after reset: 00_H

	7			0	
TIMR3	TMD	0	CNT		RD/WR (65)

TMD ... Timer Mode

Timer 3 can be used in two different modes of operation.

0: Count Down Timer.
An interrupt is generated only once after a time period of 1 ... 63 ms.

1: Periodic Timer.
An interrupt is periodically generated every 1 ... 63 ms (see CNT).

CNT ... Timer Counter

0: Timer off.

1 ... 63: Timer period = 1 ... 63 ms

By writing ’0’ to CNT the timer is immediately stopped. A value different from that determines the time period after which an interrupt will be generated.
If the timer is already started with a certain CNT value and is written again before an interrupt has been released, the timer will be reset to the new value and restarted again. An interrupt is indicated to the host in AUXI.TIN3.

Note: Reading back this value delivers back the current counter value which may differ from the programmed value if the counter is running.

5.8.6 B-Channel Registers

The registers for B-channel A are contained in the address space 70_H - $7A_H$ and for B-channel B in the address space 80_H - $8A_H$.

5.8.6.1 ISTAB - Interrupt Status Register B-Channels

Value after reset: 10_H

	7							0	
ISTAB	RME	RPF	RFO	XPR	0	XDU	0	0	RD (70/80)

For general information please refer to **Chapter 5.6.6**.

RME ... Receive Message End

One complete frame of length less than or equal to the defined block size (EXMB.RFBS) or the last part of a frame of length greater than the defined block size has been received. The contents are available in the RFIFOB. The message length and additional information may be obtained from RBCHB and RBCLB and the RSTAB register.

RPF ... Receive Pool Full

A data block of a frame longer than the defined block size (EXMB.RFBS) has been received and is available in the RFIFOB. The frame is not yet complete.

RFO ... Receive Frame Overflow

The received data of a frame could not be stored, because the RFIFOB is occupied. The whole message is lost.

This interrupt can be used for statistical purposes and indicates that the microcontroller does not respond quickly enough to an RPF or RME interrupt (ISTAB).

XPR ... Transmit Pool Ready

A data block of up to the defined block size 32 or 64 (EXMB.XFBS) can be written to the XFIFOB.

An XPR interrupt will be generated in the following cases:

- after an XTF or XME command as soon as the 32 or 64 bytes in the XFIFOB are available and the frame is not yet complete
- after an XTF together with an XME command is issued, when the whole frame has been transmitted
- after a reset of the transmitter (XRES)
- after a device reset

XDU ... Transmit Data Underrun

The current transmission of a frame is aborted by transmitting seven ’1’s because the XFIFOB holds no further data. This interrupt occurs whenever the microcontroller has failed to respond to an XPR interrupt (ISTAB register) quickly enough, after having initiated a transmission and the message to be transmitted is not yet complete.

5.8.6.2 MASKB - Mask Register B-Channels

Value after reset: FF_H

	7							0	
MASKB	RME	RPF	RFO	XPR	1	XDU	1	1	WR (70/80)

Each interrupt source in the ISTAB register can selectively be masked by setting the corresponding bit in MASKB to ’1’. Masked interrupt status bits are not indicated when ISTAB is read. Instead, they remain internally stored and pending until the mask bit is reset to ’0’.

For general information please refer to **Chapter 5.6.6**.

5.8.6.3 STARB - Status Register B-Channels

Value after reset: 40_H

	7							0	
STARB	XDOV	XFW	0	0	RACI	0	XACI	0	RD (71/81)

XDOV ... Transmit Data Overflow

More than 16 or 32 bytes (according to selected block size) have been written to the XFIFOB, i.e. data has been overwritten.

XFW ... Transmit FIFO Write Enable

Data can be written to the XFIFOB. This bit may be polled instead of (or in addition to) using the XPR interrupt.

RACI ... Receiver Active Indication

The B-channel HDLC receiver is active when RACI = ’1’. This bit may be polled. The RACI bit is set active after a begin flag has been received and is reset after receiving an abort sequence.

XACI ... Transmitter Active Indication

The B-channel HDLC-transmitter is active when XACI = ’1’. This bit may be polled. The XACI-bit is active when an XTF-command is issued and the frame has not been completely transmitted.

5.8.6.4 CMDRB - Command Register B-channels

Value after reset: 00_H

	7							0	
CMDRB	RMC	RRES	0	0	XTF	0	XME	XRES	WR (71/81)

RMC ... Receive Message Complete

Reaction to RPF (Receive Pool Full) or RME (Receive Message End) interrupt. By setting this bit, the microcontroller confirms that it has fetched the data, and indicates that the corresponding space in the RFIFOB may be released.

RRES ... Receiver Reset

HDLC receiver is reset, the RFIFOB is cleared of any data.

XTF ... Transmit Transparent Frame

After having written up to 32 or 64 bytes (EXMB.XFBS) to the XFIFOB, the microcontroller initiates the transmission of a transparent frame by setting this bit to ’1’. The opening flag is automatically added to the message by the SIUC-BA except in the extended transparent mode.

XME ... Transmit Message End

By setting this bit to ’1’ the microcontroller indicates that the data block written last to the XFIFOB completes the corresponding frame. The SIUC-BA terminates the transmission by appending the CRC (if EXMB.XCRC=0) and the closing flag sequence to the data except in the extended transparent mode.

XRES ... Transmitter Reset

The B-channel HDLC transmitter is reset and the XFIFOB is cleared of any data. This command can be used by the microcontroller to abort a frame currently in transmission.

Note: After an XPR interrupt further data has to be written to the XFIFOB and the appropriate Transmit Command (XTF) has to be written to the CMDRB register again to continue transmission, when the current frame is not yet complete (see also XPR in ISTAB).
During frame transmission, the 0-bit insertion according to the HDLC bit-stuffing mechanism is done automatically except in the extended transparent mode.

5.8.6.5 MODEB - Mode Register

Value after reset: $C0_H$

	7							0	
MODEB	MDS2	MDS1	MDS0	0	RAC	0	0	0	RD/WR (72/82)

MDS2-0 ... Mode Select

Determines the message transfer mode of the HDLC controller, as follows:

MDS2-0	Mode	Number of Address Bytes	Address Comparison 1.Byte	2.Byte	Remark
0 0 0	Reserved				
0 0 1	Reserved				
0 1 0	Non-Auto mode	1	RAL1,RAL2	–	One-byte address compare.
0 1 1	Non-Auto mode	2	RAH1,RAH2, Group Address	RAL1,RAL2, Group Address	Two-byte address compare.
1 0 0	Extended transparent mode				
1 1 0	Transparent mode 0	–	–	–	No address compare. All frames accepted.
1 1 1	Transparent mode 1	> 1	RAH1,RAH2, Group Address	–	High-byte address compare.
1 0 1	Transparent mode 2	> 1	–	RAL1,RAL2, Group Address	Low-byte address compare.

Note: - RAH1, RAH2: two programmable address values for the first received address byte (in the case of an address field longer than 1 byte);
Group Address= fixed value FC / FE_H.
- RAL1, RAL2: two programmable address values for the second (or the only, in the case of a one-byte address) received address byte;
Group Address= fixed value FF_H.

RAC ... Receiver Active

The B-channel HDLC receiver is activated when this bit is set to ’1’. If set to ’0’ the HDLC data is not evaluated in the receiver.

5.8.6.6 EXMB - Extended Mode Register B-channels

Value after reset: 00_H

	7						0	
EXMB	XFBS	RFBS	SRA	XCRC	RCRC	0	ITF	RD/WR (73/83)

XFBS … Transmit FIFO Block Size

0 … Block size for the transmit FIFO data is 64 byte
1 … Block size for the transmit FIFO data is 32 byte

Note: A change of XFBS will take effect after a receiver command (CMDRB.XME, CMDRB.XRES, CMDRB.XTF) has been written.

RFBS … Receive FIFO Block Size

RFBS		Block Size Receive FIFO
Bit 6	Bit5	
0	0	64 byte
0	1	32 byte
1	0	16 byte
1	1	8 byte

Note: A change of RFBS will take effect after a transmitter command (CMDRB.RMC, CMDRB.RRES,) has been written

SRA … Store Receive Address

0 … Receive Address is not stored in the RFIFOB
1 … Receive Address is stored in the RFIFOB

XCRC … Transmit CRC

0 … CRC is transmitted
1 … CRC is not transmitted

RCRC… Receive CRC

0 … CRC is not stored in the RFIFOB
1 … CRC is stored in the RFIFOB

ITF… Interframe Time Fill

Selects the inter-frame time fill signal which is transmitted between HDLC-frames.

0 … idle (continuous ’1’)
1 … flags (sequence of patterns: ‘0111 1110’)

5.8.6.7 RAH1 - RAH1 Register

Value after reset: 00_H

	7			0	
RAH1	RAH1		0	MHA	WR (75/85)

RAH1 ... Value of the first individual programmable high address byte

In operating modes that provide high byte address recognition, the high byte of the received address is compared with the individual programmable values in RAH1, RAH2 or group address FC_H/FE_H.

MHA ... Mask High Address

0: The RAH1 address of an incoming frame is compared with RAH1, RAH2 and Group Address.
1: The RAH1 address of an incoming frame is compared with RAH1 and Group Address. RAH1 can be masked with RAH2 thereby bitpositions of RAH1 are not compared if they are set to ’1’ in RAH2.

5.8.6.8 RAH2 - RAH2 Register

Value after reset: 00_H

	7			0	
RAH2	RAH2		0	MLA	WR (76/86)

RAH2 ... Value of the second individual programmable high address byte

See RAH1 register above. RAH1 and RAH2 are used in non-auto mode when a 2-byte address field has been selected and in the transparent mode 1.

MLA ... Mask Low Address

0: The address of an incoming frame is compared with RAL1, RAL2 and Group Address.
1: The address of an incoming frame is compared with RAL1 and Group Address. RAL1 can be masked with RAL2 thereby bitpositions of RAL1 are not compared if they are set to ’1’ in RAL2.

5.8.6.9 RBCLB - Receive Frame Byte Count Low B-Channels

Value after reset: 00_H

	7		0	
RBCLB	RBC7		RBC0	RD (76/86)

RBC7-0 ... Receive Byte Count

Eight least significant bits of the total number of bytes in a received message (see RBCHB register).

5.8.6.10 RBCHB - Receive Frame Byte Count High B-Channels

Value after reset: 00_H.

	7					0	
RBCHB	0	0	0	OV	RBC11	RBC8	RD (77/87)

OV ... Overflow

A ’1’ in this bit position indicates a message longer than (2^{12} - 1) = 4095 bytes .

RBC8-11 ... Receive Byte Count

Four most significant bits of the total number of bytes in a received message (see RBCLB register).

Note: Normally RBCHB and RBCLB should be read by the microcontroller after an RME-interrupt in order to determine the number of bytes to be read from the RFIFOB, and the total message length. The contents of the registers are valid only after an RME or RPF interrupt, and remain so until the frame is acknowledged via the RMC bit or RRES.

5.8.6.11 RAL1 - RAL1 Register 1

Value after reset: 00_H

	7						0	
RAL1	RAL1							WR (77/87)

RAL1 ... Receive Address Byte Low Register 1

The general function (READ/WRITE) and the meaning or contents of this register depends on the selected operating mode:

– Non-auto mode (16-bit address):
RAL1 can be programmed with the value of the first individual low address byte.
– Non-auto mode (8-bit address):
According to X.25 LAPB protocol, the address in RAL1 is recognized as COMMAND address.

5.8.6.12 RAL2 - RAL2 Register

Value after reset: 00_H

	7							0	
RAL2	RAL2								WR (78/88)

RAL2 ... Receive Address Byte Low Register 2

Value of the second individual programmable low address byte. If a one byte address field is selected, RAL2 is recognized as RESPONSE according to X.25 LAPB protocol.

5.8.6.13 RSTAB - Receive Status Register B-Channels

Value after reset: $0E_H$

	7							0	
RSTAB	VFR	RDO	CRC	RAB	HA1	HA0	C/R	LA	RD (78/88)

VFR... Valid Frame

Determines whether a valid frame has been received.
The frame is valid (1) or invalid (0).
A frame is invalid when there is not a multiple of 8 bits between flag and frame end (flag, abort).

RDO ... Receive Data Overflow

If RDO=1, at least one byte of the frame has been lost, because it could not be stored in RFIFOB. As opposed to ISTAB.RFO an RDO indicates that the beginning of a frame has been received but not all bytes could be stored as the RFIFOB was temporarily full.

CRC ... CRC Check

The CRC is correct (1) or incorrect (0).

RAB ... Receive Message Aborted

The receive message was aborted by the remote station (1), i.e. a sequence of seven 1’s was detected before a closing flag.

HA1, HA0 … High Byte Address Compare; significant only in non automode 16 and in transparent mode 1

In operating modes which provide high byte address recognition, the SIUC-BA compares the high byte of a 2-bytes address with the contents of two individual programmable registers (RAH1, RAH2) and the fixed values FE_H and FC_H (group address).

Depending on the result of this comparison, the following bit combinations are possible:

10 … RAH1 has been recognized
00 … RAH2 has been recognized
01 … group address has been recognized

C/R ... Command/Response

The C/R bit contains the C/R bit of the received frame (Bit1 in the SAPI address, LAPD).

LA … Low Byte Address Compare; significant only in non automodes 8 and 16 and in transparent mode 2

The low byte address of a 2-byte address field, or the single address byte of a 1-byte address field is compared with two programmable registers (RAL1, RAL2) and with the group address (fixed value FF_H)
0 … Group address has been recognized
1 … RAL1 or RAL2 has been recognized

Note: RSTAB corresponds to the last received HDLC frame; it is duplicated into RFIFOB for every frame (last byte of frame).
If several frames are contained in the RFIFOB the corresponding status information for each frame should be evaluated from the FIFO contents (last byte) as RSTAB only refers to last frame in the FIFO.

5.8.6.14 TMB -Test Mode Register B-Channels

Value after reset: 00_H

	7							0	
TMB	0	0	0	0	0	0	0	TLP	RD/WR (79/89)

TLP ... Test Loop

The TX path of layer-2 is internally connected with the RX path of layer-2. Data coming from the layer 1 controller will not be forwarded to the layer 2 controller.

5.8.6.15 RFIFOB - Receive FIFO B-Channels

	7	0	
RFIFOB	Receive data		RD (7A/8A)

A read access to this register gives access to the “current” FIFO location selected by an internal pointer which is automatically incremented after each read access.

The RFIFOB contains up to 128 bytes of received data.

After an ISTAB.RPF interrupt, a complete data block is available. The block size can be 8, 16, 32 or 64 bytes depending on the EXMB.RFBS setting.

After an ISTAB.RME interrupt, the number of received bytes can be obtained by reading the RBCLB register.

5.8.6.16 XFIFOB - Transmit FIFO B-Channels

	7	0	
XFIFOB	Transmit data		WR (7A/8A)

A write access to this register gives access to the “current” FIFO location selected by an internal pointer which is automatically incremented after each write access.

Depending on EXMB.XFBS up to 32 or 64 bytes of transmit data can be written to the XFIFOB following an ISTAB.XPR interrupt.

6 Interrupt System

The SIUC-BA provides 14 interrupt sources with 4 priority levels. 13 interrupts can be generated by the onchip peripherals (timer0, timer1, USB module, ISDN module), and 1 interrupt may be triggered externally ($\overline{INT0}$). Further interrupt sources are combined in the ISDN registers (e.g. interrupts from level detect, watchdog, timers and external interrupts, see **Figure 104**). **Figure 100**, **Figure 101**, **Figure 102** and **Figure 103** give a general overview of the interrupt sources and illustrate the request and control flags which are described in the next sections.

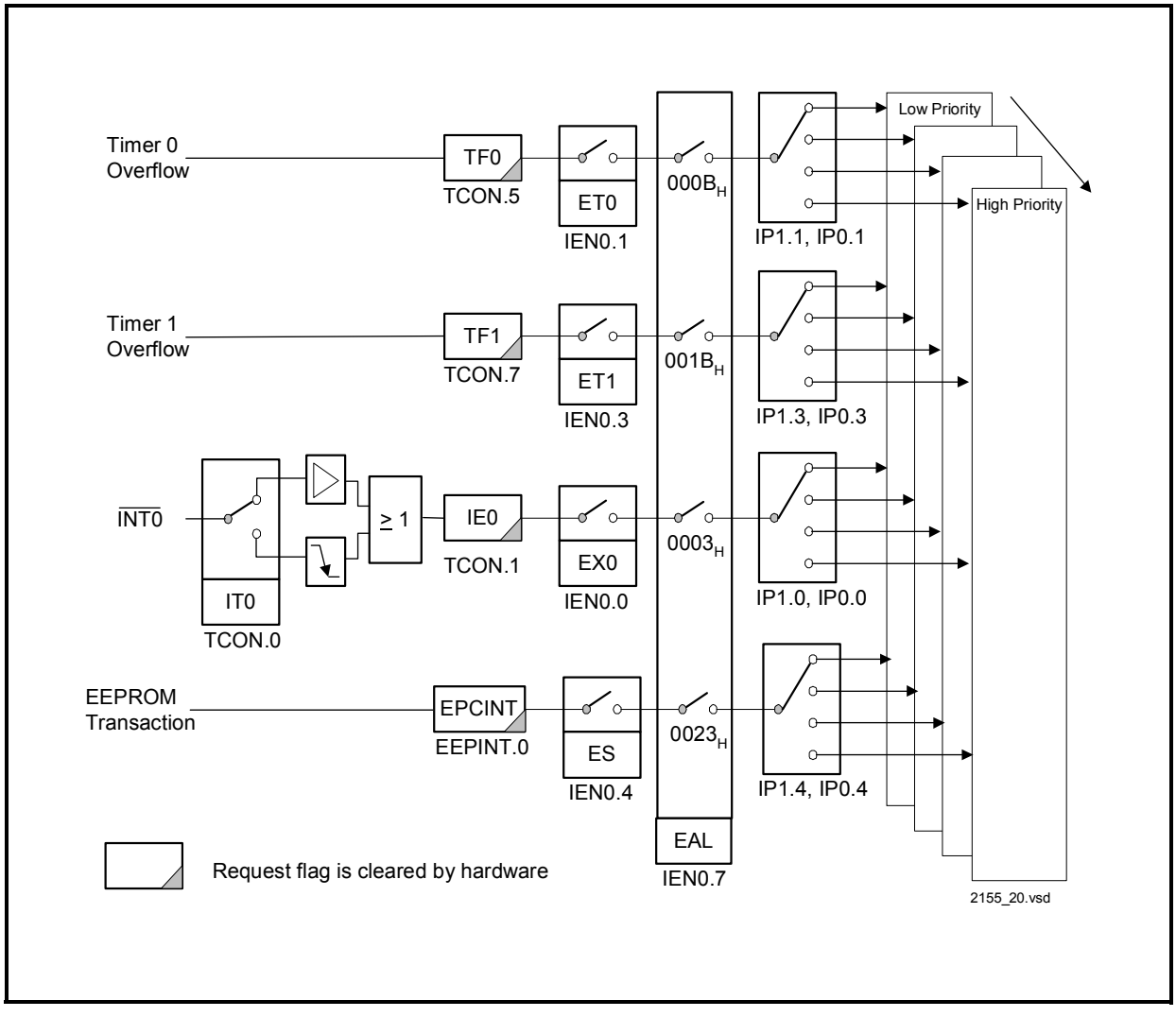


Figure 100 Interrupt Request Sources (Part 1) - Miscellaneous Interrupts

Some of the interrupts can wakeup the SIUC-BA from suspend mode (**Chapter 3.5.5**). However, in suspend mode the $\overline{INT0}$ interrupt is directly routed to the NMI interrupt, i.e. the IEN0.EX0 bit has only effect on the generation of the $\overline{INT0}$ interrupt in operational and idle mode, but not is suspend mode (see **Chapter 6.3** for wakeup from suspend).

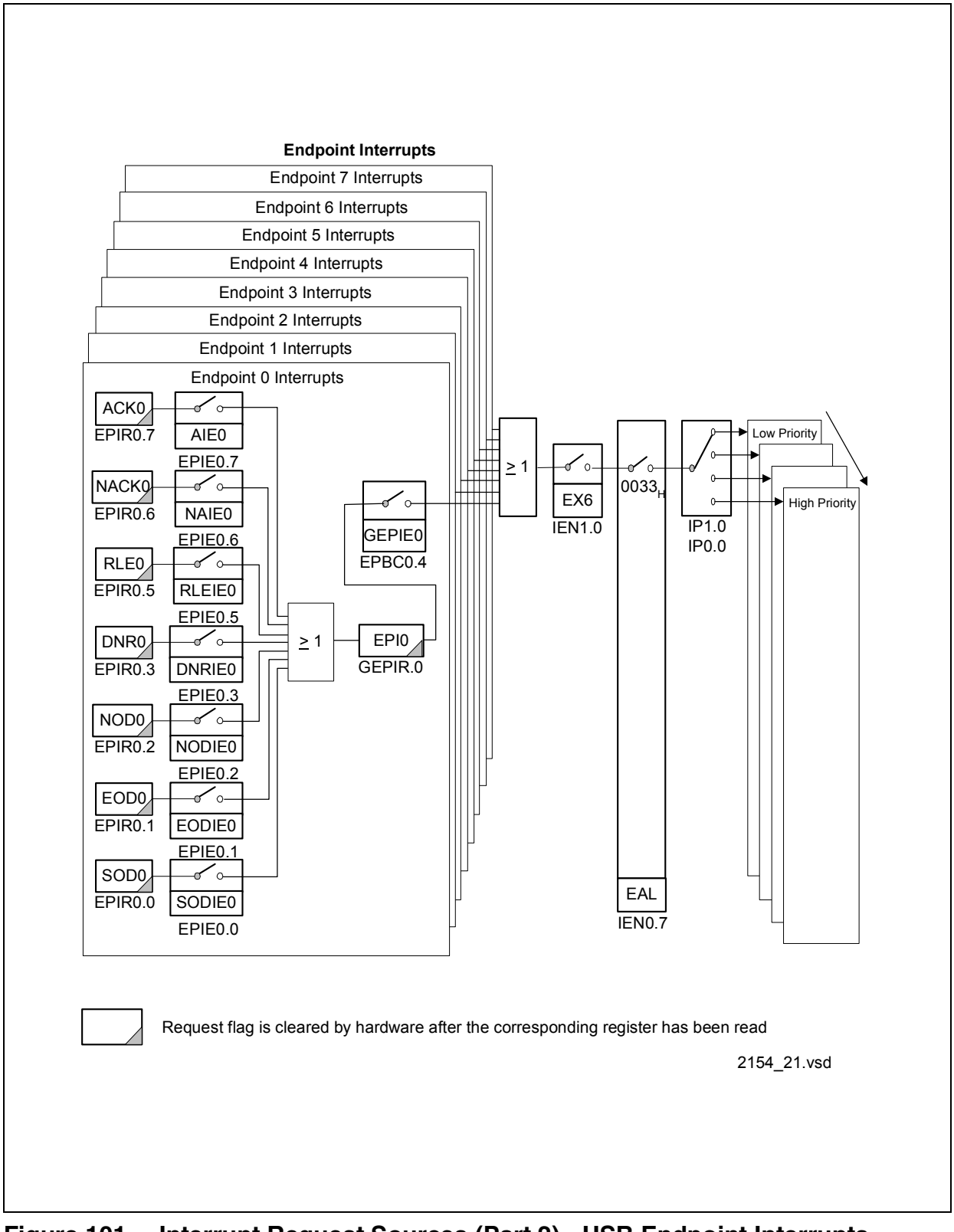


Figure 101 Interrupt Request Sources (Part 2) - USB Endpoint Interrupts

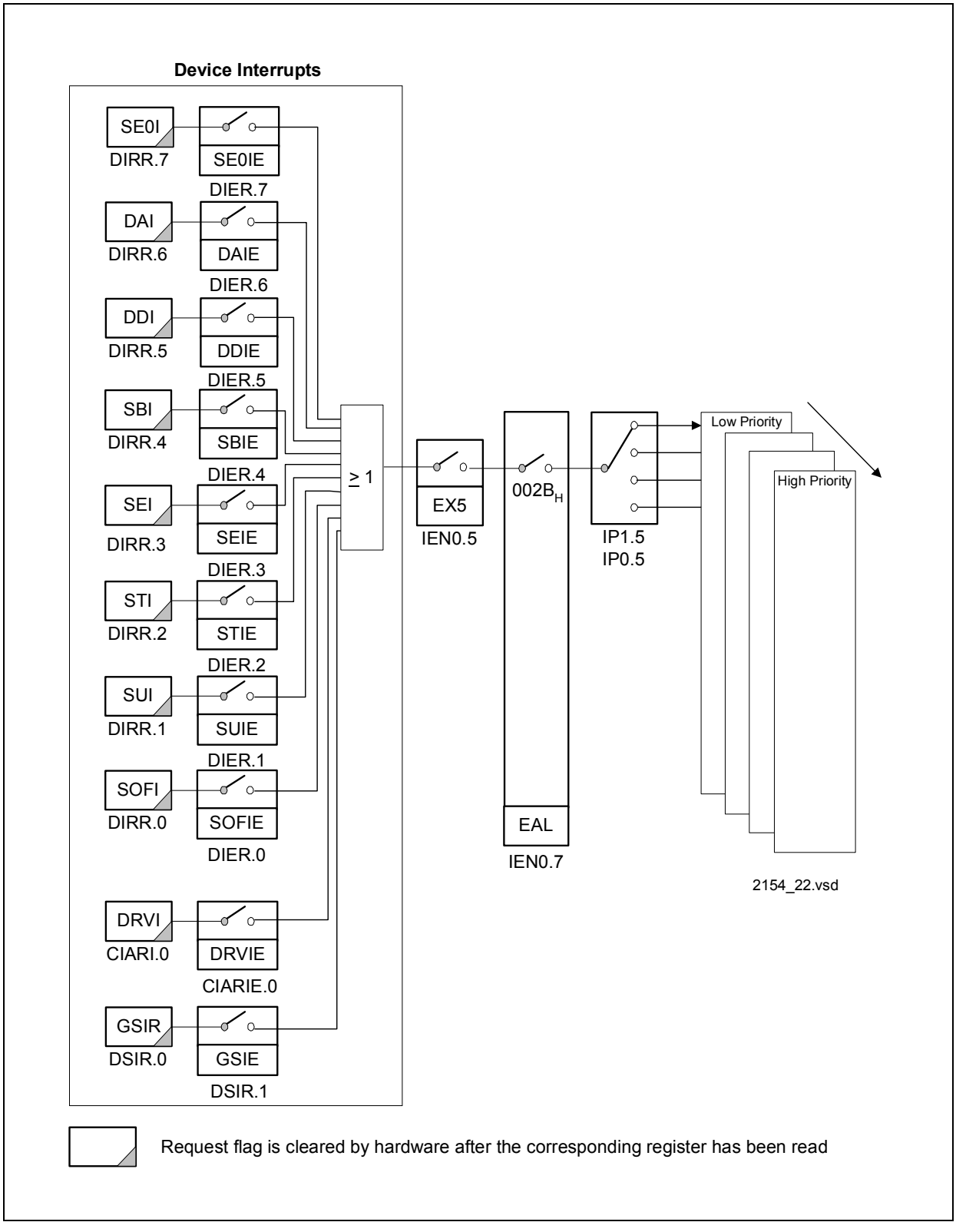


Figure 102 Interrupt Request Sources (Part 3) - USB Device Interrupts

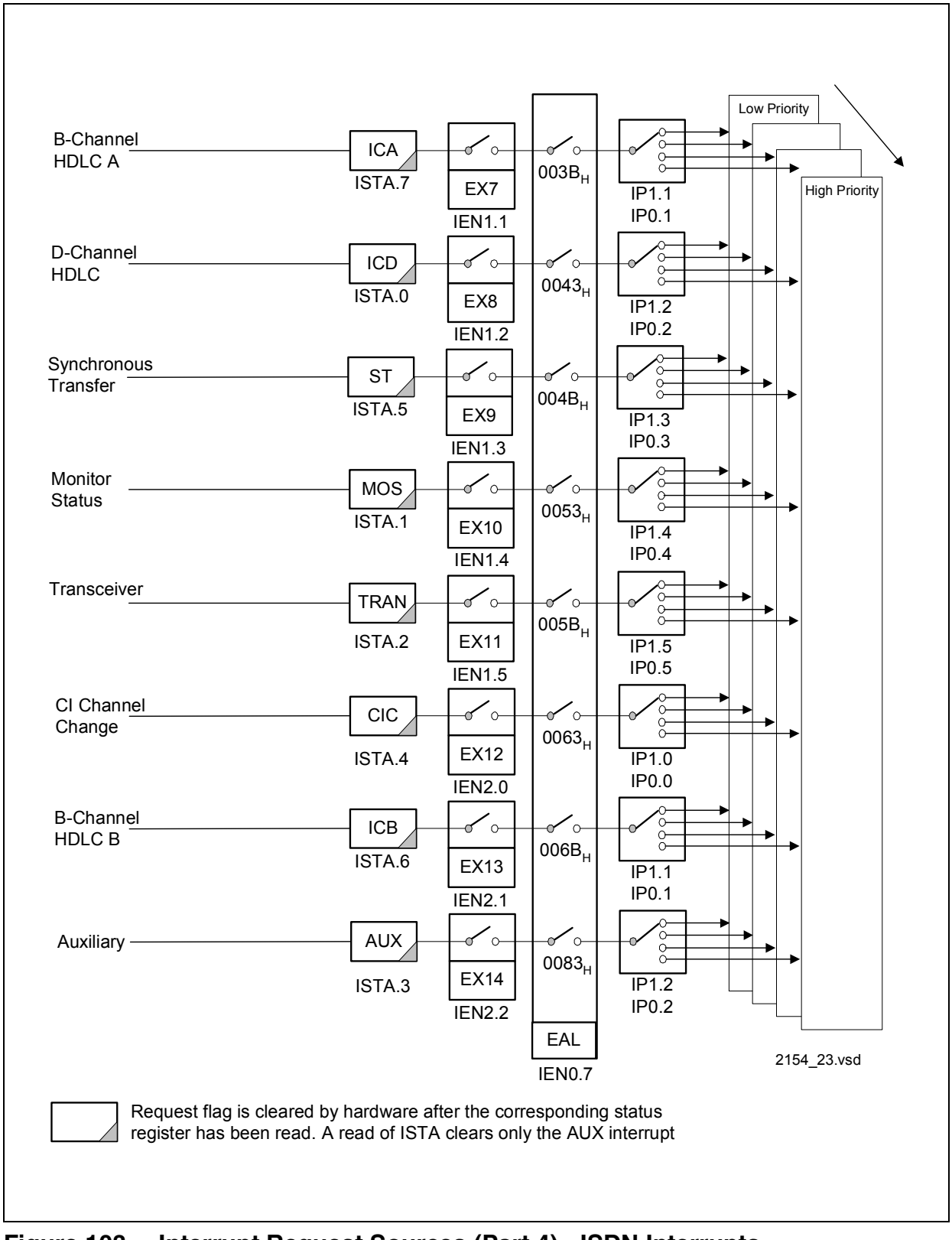


Figure 103 Interrupt Request Sources (Part 4) - ISDN Interrupts

Special events in the ISDN part are indicated by means of eight interrupt outputs (ICA, ICB, ST, CIC, AUX, TRAN, MOS, ICD), which request the µC to read status information or transfer data from/to the ISDN registers.

The cause of an interrupt must be determined by the µC by reading the corresponding interrupt status registers. For all eight individual interrupt sources the µC can read one interrupt status register (ISTA) to determine the source.

The structure of the ISDN interrupt status registers is shown in **Figure 104**.

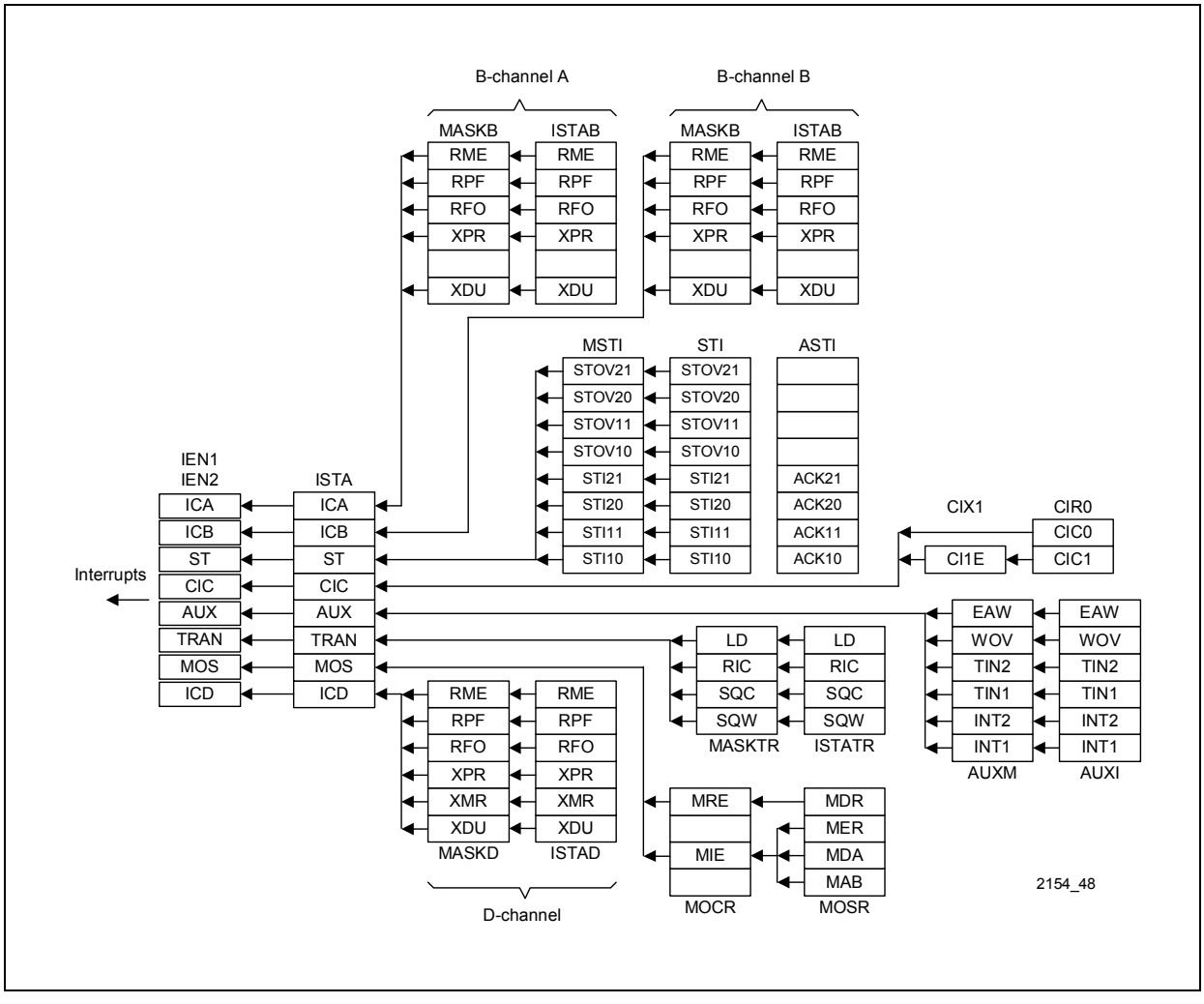


Figure 104 ISDN Interrupt Status Registers

The eight interrupts point at interrupt sources in the D-channel HDLC Controller (ICD), B-channel HDLC controllers (ICA, ICB), Monitor- (MOS) and C/I- (CIC) handler, the transceiver (TRAN), the synchronous transfer (ST) and the auxiliary interrupts (AUXI).

The ISDN interrupts from ISTA are enable/disabled in the IEN1 and IEN2 registers, however after reset the ISTA_INIT register must be set to 00H in order to enable ISDN interrupts.

6.1 Interrupt Registers

6.1.1 Interrupt Request / Control Flags

The external interrupt 0 ($\overline{INT0}$) can be either level-activated or negative transition activated, depending on bit IT0 in register TCON. The flag that generates this interrupt is bit IE0 in TCON. When the external interrupt is generated, the flag that generated this interrupt is cleared by the hardware when the service routine is vectored to, but only if the interrupt was transition-activated. If the interrupt was level-activated, then the requesting external source directly controls the request flag, rather than the onchip hardware.

The timer 0 and timer 1 interrupts are generated by TF0 and TF1 in register TCON, which are set by a rollover in their respective timer/counter registers. When a timer interrupt is generated, the flag that generated it is cleared by the onchip hardware when the service routine is vectored to.

6.1.1.1 TCON - Timer Control Register

Reset value: 00_H Address: 88_H

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	0	0	IE0	IT0
r	rw	r	rw	r	rw	r	rw

Note: The shaded bits are not used for interrupt control.

Bit	Function
TF1	**Timer 1 Overflow Flag** Set by hardware when Timer/Counter 1 overflow. Cleared by hardware when processor calls the interrupt service routine.
TR1	**Timer 1 Run Control** If 1, timer runs; if 0, timer is halted.
TF0	**Timer 0 Overflow Flag** Set by hardware when Timer/Counter 0 overflow. Cleared by hardware when processor calls the interrupt service routine.
TR0	**Timer 0 Run Control** If 1, timer runs; if 0, timer is controlled by $\overline{INT0}$ and GATE0.

IE0	**External Interrupt 0 Edge Flag (pin $\overline{INT0}$)** Set by hardware when an external interrupt condition at pin $\overline{INT0}$ is detected (low level if IT0=0 or falling edge if IT0=1).
IT0	**Interrupt 0 Control Bit (pin $\overline{INT0}$)** If 1, a falling edge triggers an interrupt; if 0, a low level triggers an interrupt.

6.1.1.2 EEPINT - EEPROM Interrupt Control Register

Reset value: 00_H Address: 93_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	EPCINT
r	r	r	r	r	r	r	rw

Bit	Function
EPCINT	**EEPROM Control Interrupt** This bit is set when an EEPROM transaction, which is started by setting the ESTA bit in the EEPSL register, is finished. The microcontroller has to acknowledge this bit by writing a ’0’ to it.

6.1.1.3 DIRR - USB Device Interrupt Request Register

Reset value: 00_H Address: $C4_H$

7	6	5	4	3	2	1	0
SE0I	DAI	DDI	SBI	SEI	STI	SUI	SOFI
r	r	r	r	r	r	r	r

For accessing DIRR, the SFR EPSEL must be 80_H.

The USB device interrupt request register contains the device specific interrupt flags of the USB module. These flags are set after the occurrence of special events. If a request flag is set, it is automatically cleared after a read operation of the DIRR register.

The interrupts contained in the DIRR register can individually be masked in the DIER register.

Bit	Function
SE0I	**Single Ended Zero Interrupt** SE0I is set each time a single ended zero is detected for equal or greater than 2.5 µs. EOP (2 bit times) is not detected.
DAI	**Device Attached Interrupt** Bit DAI is automatically set after detection of the USB device being attached to the USB bus (for further information see **Chapter 4.8**).
DDI	**Device Detached Interrupt** Bit DDI is automatically set after detection of the device being detached from the USB bus (for further information see **Chapter 4.8**).
SBI	**Suspend Begin Interrupt** SBI is automatically set when the suspend mode is entered.
SEI	**Suspend End Interrupt** SEI is automatically set when the suspend mode is left.
STI	**Status Interrupt** STI is set if the host requires a status transfer and the device answers with NACK (if bit ESP is set, the device answers with ACK and then STI is not set).
SUI	**Setup Interrupt** SUI is automatically set after a successful reception of a setup packet which is not handled by the USB module and must be forwarded to the CPU. The setup packet itself is limited to 8 bytes and stored at USB memory addresses 00H to 07H. If a setup interrupt occurs, the control and status bits UBF0, CBF0, SOD0 and DIR0 in the endpoint registers EPBS0 and EPIR0 are cleared. DIR0=0 predicts the direction of the next USB access (data phase) to be from host to CPU. Bit DIR0 is automatically set (CPU to host), if the host tries to perform a read access in the first data packet.
SOFI	**Start of Frame Interrupt** SOF is automatically set after detection of a start of frame packet on the USB.

The register EPIRn (n=0-7) contains USB endpoint specific interrupt request flags. This SFR is available for each endpoint. If a request flag in EPIRn is set, it is automatically cleared after a read operation of the EPIRn register.

6.1.1.4 DSIR - Device Setup Interrupt Register

This register can only be accessed when the endpoint select register (adr. $D2_H$) is set to EPSEL = 80_H.

Reset value: 00_H Address: $C5_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	GSIE	GSIR
r	r	r	r	r	r	rw	r

Bit	Function
GSIE	**Global SETUP packet interrupt enable** 0: interrupt generation is disabled 1: interrupt generation is enabled
GSIR	**Global SETUP packet interrupt request** A setup packet has been received.

A read access to the DSIR register will clear a GSIR interrupt request. This interrupt is generated for all USB standard device requests (see **Chapter 4.6.3**).

6.1.1.5 EPIRn - Endpoint Interrupt Request Register

Reset value EPIR0: 01_H, Reset value EPIR1-EPIR7: 00_H Address: $C4_H$

7	6	5	4	3	2	1	0
ACKn	NACKn	RLEn	0	DNRn	NODn	EODn	SODn
r	r	r	r	r	r	r	r

For accessing EPIRn, the SFR EPSEL must be $0n_H$.

The interrupts contained in the EPIRn register can individually be masked in the EPIEn register.

Bit	Function
ACKn	**USB Acknowledge** Bit ACKn=1 indicates a successful action on the USB.
NACKn	**USB Not Acknowledge** Bit NACK is set for all unsuccessful actions on the USB.
RLEn	**Read Length Error** Bit RLEn is automatically set if the number of bytes read by the USB does not correspond to the packet length programmed by the CPU.
DNRn	**Data Not Ready** This bit is set by hardware if the USB module requires an access to USB memory, but no buffer is available. USB Read Action: DNRn is set if UBF is not set. USB Write Action: DNRn is set if UBF is set.
NODn	**No Data** This bit indicates an incorrect CPU read or write access to USB memory. It is set if the CPU processes a read access to an empty USB buffer or a write access to a full buffer. NODn is also set if the direction is write (DIRn=0 for USB write access) and the CPU tries to write to the USB memory buffer.
EODn	**End of Data** During a **USB read** access EODn is set if the CPU has written a programmable number (MaxLen) of bytes in the transmit buffer. As a result, the buffer is full and no more write actions from the CPU are allowed. During a **USB write** access EODn is set if the CPU has read a programmable number (USBLen) of bytes out of the receive buffer. As a result, the buffer is empty now and no more read actions from the CPU are allowed.
SODn	**Start of Data** During a **USB read** access SODn is set if the USB has read a fixed number (USBLen) of bytes from the transmit buffer. As a result, the buffer is now empty and the CPU can process write actions again. During a **USB write** access SODn is set if the USB has written a fixed number (USBLen) of bytes to the receive buffer. As a result, the buffer is full and the CPU can start read actions.

In dual buffer mode, bits SODn and EODn can be set simultaneously if the corresponding buffer page is swapped.

6.1.1.6 GEPIR - Global Endpoint Interrupt Request Register

The global endpoint interrupt request register GEPIR contains one flag for each endpoint which indicates whether one or more of the seven endpoint specific interrupt requests has become active. If a request flag in GEPIR is set, it is automatically cleared after a read operation of the GEPIR register.

Reset value: 00_H Address: $D6_H$

7	6	5	4	3	2	1	0
EPI7	EPI6	EPI5	EPI4	EPI3	EPI2	EPI1	EPI0
r	r	r	r	r	r	r	r

Bit	Function
EPI7	**Endpoint Interrupt 7 request flag** If EPI7 is set, an endpoint interrupt 7 request is pending.
EPI6	**Endpoint Interrupt 6 request flag** If EPI6 is set, an endpoint interrupt 6 request is pending.
EPI5	**Endpoint Interrupt 5 request flag** If EPI5 is set, an endpoint interrupt 5 request is pending.
EPI4	**Endpoint Interrupt 4 request flag** If EPI4 is set, an endpoint interrupt 4 request is pending.
EPI3	**Endpoint Interrupt 3 request flag** If EPI3 is set, an endpoint interrupt 3 request is pending.
EPI2	**Endpoint Interrupt 2 request flag** If EPI2 is set, an endpoint interrupt 2 request is pending.
EPI1	**Endpoint Interrupt 1 request flag** If EPI1 is set, an endpoint interrupt 1 request is pending.
EPI0	**Endpoint Interrupt 0 request flag** If EPI0 is set, an endpoint interrupt 0 request is pending.

6.1.1.7 CIARI - Configuration Request Interrupt Register

The Configuration, Interface & Alternate Setting Interrupt Register (CIARI) sends an interrupt to the µC whenever the host programs multiple device configurations or interfaces.

Reset value: 00_H Address: $D7_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	DRVI
r	r	r	r	r	r	r	r

The interrupt contained in the CIARI register can be masked in the CIARIE register.

Bit	Function
DRVI	**Device Request Value Interrupt** Bit DRVI is set each time the host sends a device request that contains one or more of the following: • Configuration Value • Interface, Alternate Setting This flag can only be cleared by writing a 0 to the bit. Writing a 1 to the bit will be ignored. The interrupt has to be enabled by bit DRVIE before being used.

Note: This interrupt is an additional USB device interrupt and shares the vector address $002B_H$.

6.1.1.8 ISTA - ISDN Status Register

The ISTA register is described in detail with the ISDN registers in **Chapter 5.8**.

Reset value: 01_H Address: $F860_H$

7	6	5	4	3	2	1	0
ICA	ICB	ST	CIC	AUX	TRAN	MOS	ICD
r	r	r	r	r	r	r	r

The interrupts contained in the ISTA register can individually be masked in the MASK register. For all interrupts in the ISDN status register ISTA, the following logical states are applied:

0: Interrupt is not active
1: Interrupt is active

Bit	Function
ICA	**HDLC Interrupt from B-channel A**
ICB	**HDLC Interrupt from B-channel B**
ST	**Synchronous Transfe**r To enable the microcontroller to lock on to the IOM timing for synchronous transfers, the source can be read from the STI register
CIC	**CI Channel Change** A change on C/I Channel 0 or C/I Channel 1 has been recognised. The actual value can be read from CIR0 or CIR1
AUX	**Auxiliary Interrupt** Indicates watchdog timer overflow, timer 1 or timer 2 time-out or an external interrupt $\overline{INT1}$ / $\overline{INT2}$, the source can be read from the auxiliary interrupt register AUXI
TRAN	**Transceiver Interrupt** Indicates an interrupt from the transceiver interrupt status register ISTATR
MOS	**Monitor Status** Indicates a change in the Monitor Status Register (MOSR)
ICD	**HDLC Interrupt from D-channel**

Note: A read of the ISTA register clears none of the interrupts. They are only cleared by reading the corresponding status register.

6.1.2 Interrupt Enable Registers

Each interrupt can be individually enabled or disabled by setting or clearing the corresponding bit in the global interrupt enable registers IEN0, IEN1 and IEN2, the USB specific DIER, EPIEn and EPBCn registers or the ISDN specific enable registers. Register IEN0 also contains the global disable bit (EAL), which can be cleared to disable all interrupts at once. The USB and ISDN interrupt sources have further enable bits for individual interrupt control.

The IEN0 register contains the general enable/disable flags of the external interrupt 0, the timer interrupts and 2 other interrupts. The 2 USB interrupts are enabled/disabled by bits in the IEN0/1 registers. The 8 ISDN interrupts are enabled/disabled by bits in the IEN1/2 registers. After reset, the enable bits of IEN0, IEN1 and IEN2 are set to 0, i.e. the corresponding interrupts are disabled.

6.1.2.1 IEN0 - Interrupt Enable Register 0

Reset value: 00_H Address: $A8_H$

7	6	5	4	3	2	1	0
EAL	0	EX5	ES	ET1	res.	ET0	EX0
rw	r	rw	rw	rw	rw	rw	rw

Bit	Function
EAL	**Enable all interrupts** When set to 0, all interrupts are disabled. When set to 1, interrupts are individually enabled/disabled according to their respective bit selection.
EX5	**Enable External Interrupt 5 - UDCI** (USB Device Interrupt)
ES	**Enable SPI EEPROM Control Interrupt - EPCINT** (EEPROM Control Int)
ET1	**Enable Timer 1 Overflow Interrupt**
ET0	**Enable Timer 0 Overflow Interrupt**
EX0	**Enable External Interrupt 0 - $\overline{\text{INT0}}$** The $\overline{\text{INT0}}$ interrupt can only be disabled in operational and idle mode. In suspend mode this interrupt is directly routed to the NMI service routine, i.e. bit EX0 is don't care in suspend mode.

Note: res. = bit is reserved and must not be changed.

6.1.2.2 IEN1- Interrupt Enable Register 1

Reset value: 00_H Address: $A9_H$

7	6	5	4	3	2	1	0
0	0	EX11	EX10	EX9	EX8	EX7	EX6
r	r	rw	rw	rw	rw	rw	rw

Bit	Function
EX11	**Enable External Interrupt 11 -TRAN** (Transceiver)
EX10	**Enable External Interrupt 10 -MOS** (Monitor Status)
EX9	**Enable External Interrupt 9 -ST** (Synchronous Transfer)
EX8	**Enable External Interrupt 8 -ICD** (HDLC D-Channel)
EX7	**Enable External Interrupt 7 -ICA** (HDLC B-Channel A)
EX6	**Enable External Interrupt 6 -EPI** (USB Endpoint Interrupt)

6.1.2.3 IEN2 - Interrupt Enable Register 2

Reset value: 00_H Address: AA_H

7	6	5	4	3	2	1	0
0	0	res.	res.	res.	EX14	EX13	EX12
r	r	r	r	r	rw	rw	rw

Bit	Function
EX14	**Enable External Interrupt 14-AUX** (Auxiliary Functions)
EX13	**Enable External Interrupt 13 -ICB** (HDLC B-Channel B)
EX12	**Enable External Interrupt 12 -CIC** (C/I Code Change)

Note: res. = bit is reserved and must not be changed.

6.1.2.4 DIER - USB Device Interrupt Enable Register

The device interrupt enable register DIER contains the enable bits for the different USB device interrupts. A device interrupt can only be generated if IEN0.EX5 and EAL are set too.

Reset value: 00_H Address: $C3_H$

7	6	5	4	3	2	1	0
SE0IE	DAIE	DDIE	SBIE	SEIE	STIE	SUIE	SOFIE
rw	rw	rw	rw	rw	rw	rw	rw

For accessing DIER, the SFR EPSEL must be 80_H.

The interrupts contained in the DIRR register can individually be masked in the DIER register.

Bit	Function
SE0IE	**Single Ended Zero Interrupt Enable**
DAIE	**Device Attached Interrupt Enable**
DDIE	**Device Detached Interrupt Enable**
SBIE	**Suspend Begin Interrupt Enable**
SEIE	**Suspend Change Interrupt Enable**
STIE	**Status Interrupt Enable**
SUIE	**Setup Interrupt Enable**
SOFIE	**Start of Frame Interrupt Enable**

6.1.2.5 EPIEn - USB Endpoint Interrupt Enable Register

Reset value: 00_H Address: $C3_H$

7	6	5	4	3	2	1	0
AIEn	NAIEn	RLEIEn	0	DNRIEn	NODIEn	EODIEn	SODIEn
rw	rw	rw	r	rw	rw	rw	rw

The endpoint interrupt enable registers contain the endpoint specific interrupt enable bits. With these bits, the endpoint specific interrupts can be individually enabled or disabled. In addition to a bit in an EPIEn register, the global interrupt bit EPIn in GEPIR for endpoint n and the general endpoint interrupt bit IEN1.EX6 and the general interrupt enable bit IEN0.EAL must be set for the interrupt to become active.

For accessing EPIEn, the SFR EPSEL must be $0n_H$.

Bit	Function
AIEn	**Acknowledge Interrupt Enable** Bit AIEn enables the generation of an endpoint specific acknowledge interrupt when bit ACKn in register EPIRn is set.
NAIEn	**Not Acknowledged Interrupt Enable** Bit NAIEn enables the generation of an endpoint specific not acknowledged interrupt when bit NACKn in register EPIRn is set.
RLEIEn	**Read Length Error Interrupt Enable** Bit RLEIEn enables the generation of an endpoint specific read length error interrupt when bit RLEn in register EPIRn is set.
DNRIEn	**Data Not Ready Interrupt Enable** Bit DNRIEn enables the generation of an endpoint specific data not ready interrupt when bit DNRn in register EPIRn is set.
NODIEn	**No Data Interrupt Enable** Bit NODIEn enables the generation of an endpoint specific no data interrupt when bit NODn in register EPIRn is set.
EODIEn	**End of Data Interrupt Enable** Bit EODIEn enables the generation of an endpoint specific end of data interrupt when bit EODn in register EPIRn is set.
SODIEn	**Start of Data Interrupt Enable** Bit SODIEn enables the generation of an endpoint specific start of data interrupt when bit SODn in register EPIRn is set.

6.1.2.6 EPBCn - Endpoint n Buffer Control Register (n=0-7)

Reset value: 00_H Address: $C1_H$

7	6	5	4	3	2	1	0
STALLn	0	0	GEPIEn	SOFDEn	INCEn	0	DBMn
rw	r	r	rw	rw	rw	r	rw

For accessing EPBCn registers, the SFR EPSEL must be $0n_H$. Note: The shaded bits are not used for interrupt control.

Bit	Function
GEPIEn	**Global Endpoint Interrupt Enable** Bit GEPIEn enables or disables the generation of the global endpoint interrupt for endpoint n based on the endpoint specific interrupt request bits in register EPIRn. GEPIEn is used to enable/disable a specific endpoint interrupt, whereas IEN1.EX6 enables/disables all endpoint interrupts irrespective of GEPIEn.

6.1.2.7 CIARIE - Configuration Request Interrupt Enable

Reset value: 00_H Address: $D8_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	DRVIE
r	r	r	r	r	r	r	rw

Bit	Function
DRVIE	**Device Request Value Interrupt Enable** When 1, this bit enables the device request value interrupt.

6.1.2.8 ISTA_INIT - ISDN Interrupt Status Register Initialize

Reset value: FF_H Address: $F860_H$

7	6	5	4	3	2	1	0
rw	rw	rw	rw	rw	rw	rw	rw

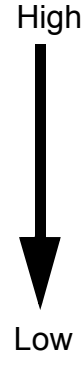
In order to enable ISDN interrupt generation to the microcontroller, the ISTA_INIT register must be programmed to 00_H after reset. After that this register should not be changed any more.

Enabling and disabling of ISDN interrupts (ISTA register) should be done in the Interrupt Enable Registers IEN1 and IEN2.

6.1.3 Interrupt Priority

For the purposes of assigning priority, the 14 interrupt sources are divided into groups determined by their bit position in the Interrupt Enable Registers and their respective requests are scanned in the order shown below.

Table 25 Interrupt Priority Order



Interrupt Group			
0 (bit 0)	External Interrupt 0	Ext Int 6 (USB Endpoint)	Ext Int 12 (CIC)
1 (bit 1)	Timer 0 Overflow	Ext Int 7 (ICA)	Ext Int 13 (ICB)
2 (bit 2)	Not Used	Ext Int 8 (ICD)	Ext Int 14 (AUX)
3 (bit 3)	Timer 1 Overflow	Ext Int 9 (ST)	
4 (bit 4)	ES (EEPROM Control)	Ext Int 10 (MOS)	
5 (bit 5)	Ext Int 5 (USB Device)	Ext Int 11 (TRAN)	

In the standard configuration, each interrupt group may be individually assigned to one of four priority levels by writing to the IP0 and IP1 Interrupt Priority registers at the corresponding bit position.

An interrupt service routine may only be interrupted by an interrupt of higher priority, and if two interrupts of different priority occur at the same time, the higher level interrupt will be serviced first. An interrupt can not be interrupted by another interrupt of the same or a lower priority level. If two interrupts of the same priority level occur simultaneously, the order in which the interrupts are serviced is determined by the scan order shown above, i.e. if two groups are programmed to the same priority level, the priority of these two groups within this level is as shown in **Table 25**.

6.1.3.1 IP0 / IP1 - Endpoint Priority Registers

Reset value: 00_H Address: $B8_H$

	7	6	5	4	3	2	1	0
IP0	0	0	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	r	r	rw	rw	rw	rw	rw	rw

Reset value: 00_H Address: AC_H

	7	6	5	4	3	2	1	0
IP1	0	0	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	r	r	rw	rw	rw	rw	rw	rw

Bit	Function
IP1.0 - IP0.0 (group 0) IP1.1 - IP0.1 (group 1) IP1.2 - IP0.2 (group 2) IP1.3 - IP0.3 (group 3) IP1.4 - IP0.4 (group 4) IP1.5 - IP0.5 (group 5)	Interrupt Group Priority Level set as follows: 00 = Group x set to Priority Level 0 (lowest) 01 = Group x set to Priority Level 1 10 = Group x set to Priority Level 2 11 = Group x set to Priority Level 3 (highest) A pair of bits from the IP0 and IP1 registers is used for one group to select its priority.

Note: x = Interrupt group as shown in **Table 25**.

6.2 Interrupt Handling

The interrupt flags are sampled at S5P2 in each machine cycle. The samples flags are polled during the following machine cycle. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate a long call (LCALL) to the appropriate service routine. In some cases it also clears the flag that generated the interrupt, while in other cases it does not; then this has to be done by the user's software. If any interrupt flag is active but not being responded to, or if the flag is no longer active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. When an interrupt is serviced, a long call instruction is executed to one of the locations listed in the following table:

Table 26 Interrupt Vectors

Interrupt Source	Interrupt Enable		Vector Address	Interrupt Request Flag
	Register	Bit	(hex)	
External Interrupt 0	IEN0	EX0	0003	TCON.IE0
Timer 0 Overflow	IEN0	ET0	000B	TCON.TF0
			0013	(not used)
Timer 1 Overflow	IEN0	ET1	001B	TCON.TF1
SPI Interface	IEN0	ES	0023	EEPINT.EPCINT
USB Device Interrupt	IEN0	EX5	002B	DIRR, CIARI, DSIR
USB Endpoint Interrupt	IEN1	EX6	0033	GEPIR
ISDN B-channel A	IEN1	EX7	003B	ISTA.ICA
ISDN D-channel	IEN1	EX8	0043	ISTA.ICD
ISDN Synchronous Transfer	IEN1	EX9	004B	ISTA.ST
ISDN MONITOR Channel	IEN1	EX10	0053	ISTA.MOS
ISDN Transceiver Interrupt	IEN1	EX11	005B	ISTA.TRAN
ISDN C/I Channel Interrupt	IEN2	EX12	0063	ISTA.CIC
ISDN B-channel B	IEN2	EX13	006B	ISTA.ICB
Auxiliary Interface Interrupt	IEN2	EX14	0083	ISTA.AUX
			007B	Non Maskable Interrupt

Note: The Non Maskable Interrupt (NMI) input to the C800 core is activated by a wakeup from Suspend mode. The µC then branches to address $007B_H$.

The external interrupt from pin $\overline{\text{INT0}}$ which is indicated in TCON.IE0 can only be disabled via IEN0.EX0 in operational and idle mode, but in suspend mode EX0 has no effect. The vector address shown in the table above is only valid for operational and idle mode, in suspend mode the $\overline{\text{INT0}}$ interrupt is directly routed to the NMI.

6.3 Wakeup from Suspend

In suspend mode certain events can wakeup the device which are

- USB resume (activity is detected on the bus)
- External interrupt from pin $\overline{\text{INT0}}$, $\overline{\text{INT1}}$, $\overline{\text{INT2}}$ and $\overline{\text{EAW}}$
- Level detect on S interface (incoming call is detected)
- C/I-code change is detected

Before going into suspend mode the wakeup sources are individually enabled and disabled in the Wakeup Control Register (WCON) and in some specific Mask Registers (**Figure 105**).

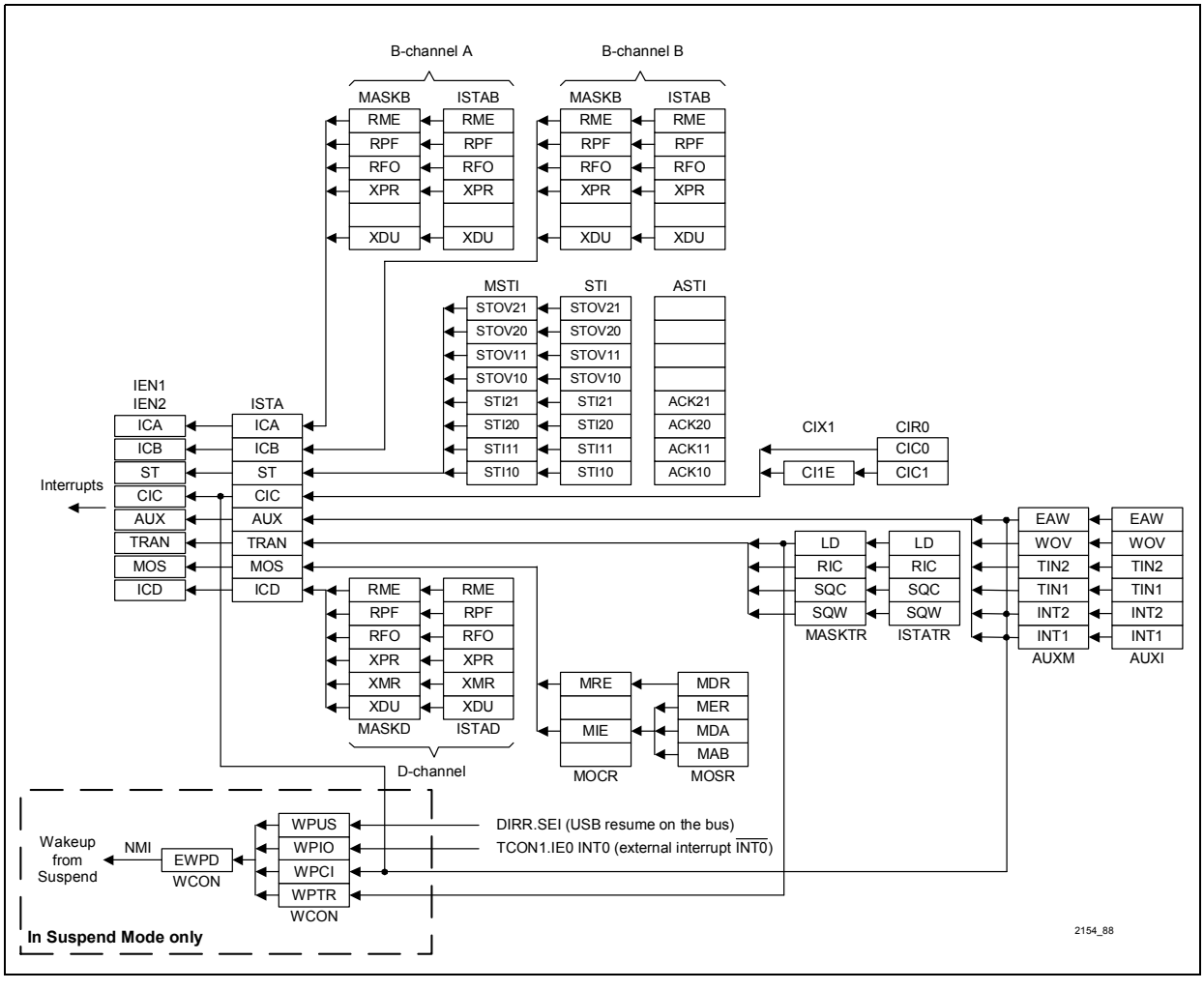


Figure 105 Wakeup Sources in Suspend Mode

The WCON.EWPD bit (External Wakeup from Power Down Enable) is used to enable/ disable all wakeup sources in general. Another four bits in the same register are used to individually control four groups of wakeup sources (**Table 27**). For all wakeup events from the ISDN block (except CIC0) the corresponding mask bit can be used to disable the event as wakeup source individually.

Table 27 Enabling / Disabling of Wakeup Sources

Enable / Disable via			Wakeup Source	Description
WCON-Register		Mask-Register		
EWPD	WPUS		DIRR.SEI	USB resume on the bus
	WPIO		TCON.IE0	External interrupt from $\overline{\text{INT0}}$
	WPCI	AUXM.INT1	AUXI.INT1	External interrupt from $\overline{\text{INT1}}$
		AUXM.INT2	AUXI.INT2	External interrupt from $\overline{\text{INT2}}$
		AUXM.EAW	AUXI.EAW	External interrupt from $\overline{\text{EAW}}$
			CIR0.CIC0	C/I0-Code Change
		CIX1.CI1E	CIR0.CIC1	C/I1-Code Change
	WPTR	MASKTR.LD	ISTATR.LD	Level detect on S interface

The vector address of the interrupt source in normal operation mode (see **Table 26**) is not valid in suspend mode and the corresponding interrupt status bit is not set when a wakeup event occurs. Instead, in suspend mode the wakeup source is directly routed to the NMI with the corresponding interrupt vector address $007B_H$. This means that the actual wakeup source cannot be determined if more than one wakeup source was enabled.

Note: It should be noted that the control function "level detect discard" (TRCONF0.LDD) should not be used to disable the wakeup function from the S interface in suspend mode. Instead, the WCON.WPTR should be used to control this wakeup source.

Suspend Mode with Disabled Remote Wakeup

Special care should be taken if the device is going into suspend mode while the remote wakeup capability is disabled (e.g. the host previously sent a Set_Feature command with “remote wakeup disabled”). Before going to suspend mode all wakeup sources - except USB resume on the bus (DIRR.SEI) - should be disabled via the WCON-register and their respective mask registers. Additionally, the S-transceiver should be switched off completely (TR_CONF0.DIS_TR = 1) and the reset source selection should be disabled (MODE1.RSS2,1 = 00).

Note: To avoid floating inputs at FSC and DCL, pull up resistors should be provided as both clock signals become input when the transceiver is switched off.

7 Firmware

The firmware that is provided with the SIUC-BA implements various functions that allow fast and efficient system design. It consists of an embedded universal boot loader contained in ROM and downloadable operational firmware.

7.1 Firmware Operation Modes

The SIUC-BA provides the ability to download microcontroller code into internal memory. This allows for flexible firmware upgrades and reduces the risk of changing standards (e.g. USB or ISDN Specifications).

Table 28 shows which boot modes are selected by pin strapping. A boot loader is contained in ROM and after reset the µC reads the BMOD1-0 pins (register HCON) to perform one of the described operations.

Table 28 Boot Mode Selection

State of Pin $\overline{EA}$	Selection by Pinstrapping		Mode Description
	BMOD1	BMOD0	
0	X	X	(this setting is not defined and therfore must not be used)
1	0	0	**Download Mode** The firmware is downloaded from the host via USB into the internal RAM of the SIUC-BA. This is controlled by the onchip bootloader contained in ROM. The download firmware can either be the ready to use firmware provided by Infineon Technologies, or it can be propietary customer firmware.
1	0	1	Reserved for further use.
1	1	0	
1	1	1	**Test Mode** This mode is reserved for manufacturing test.

Note: X = don’t care

7.2 Boot Loader Firmware

The operations performed after reset are shown in **Figure 106**.

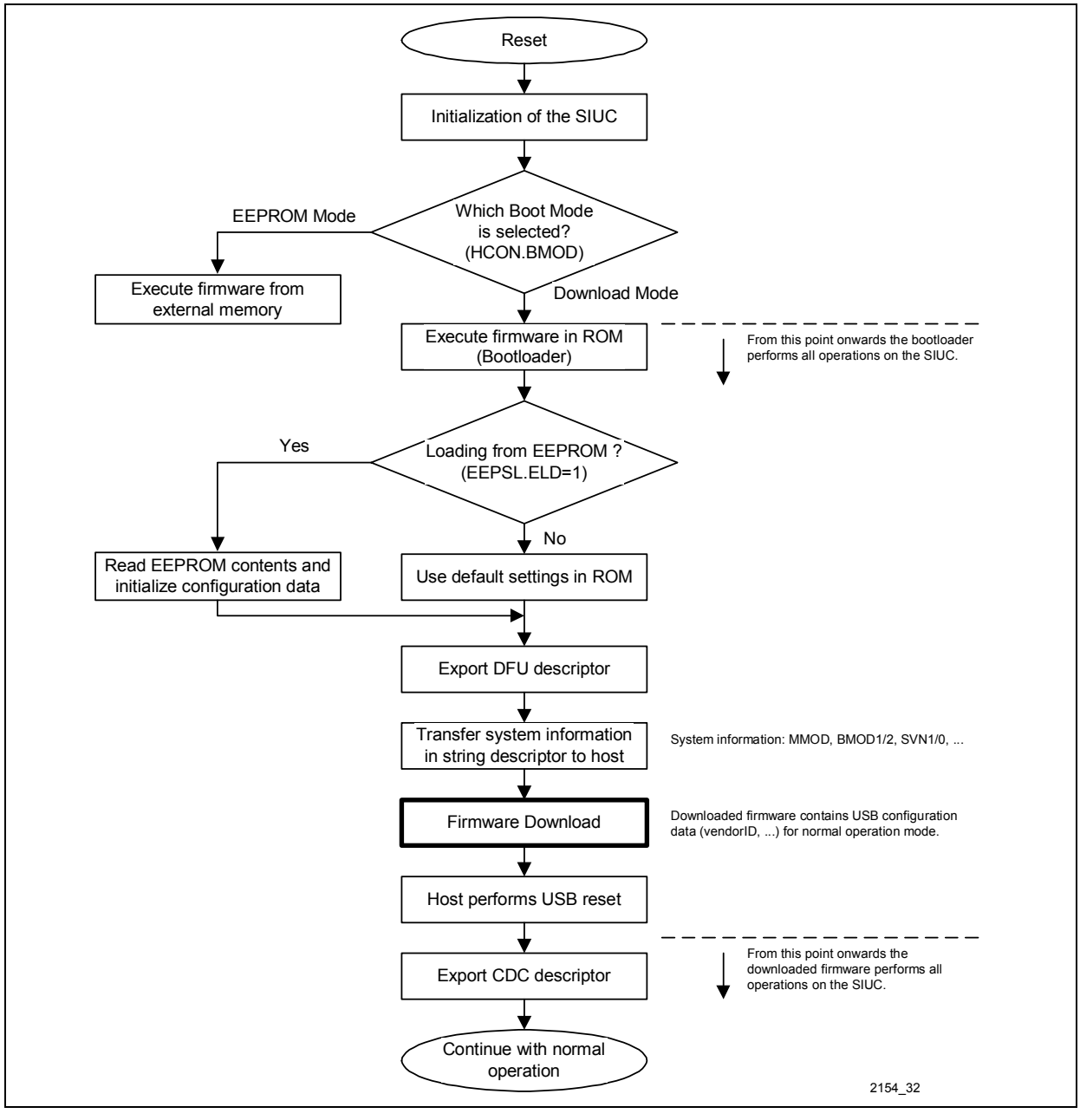


Figure 106 Bootmode Procedure

After a power on reset, the µC executes the bootloader from internal ROM.

Pin ELD (a multiplexed function of pin AUX3) which is read via register EEPSL.ELD indicates whether the µC should load USB configuration data (Vendor ID, Device ID,...) from external EEPROM or whether default settings in internal ROM should be used.

The boot loader initializes the CPU and the USB core with the DFU (Device Firmware Upgrade) configuration and - upon host request - exports a DFU descriptor to the host, indicating that this device requires a firmware download. This USB configuration uses only the default endpoint (EP0) with the buffer size set to 64 bytes.

After initialization, the boot loader waits for commands (some of them are vendor specific) to get:

- the download firmware including a header with information about it (size, Vendor ID and Product ID or their sources, ...)
  Note: As the SIUC-BA supports an SPI interface to an EEPROM, the connected EEPROM can be the source for a vendor specific encryption key. The IDs are always downloaded.
- commands to program the EEPROM
- a command, which aborts the download on request from the host
- a command, which finishes the download and says "I will send you a RESET within the next few milliseconds. You have to restart and execute the operational firmware". After the command the device waits for a RESET on the USB bus. Then it activates the firmware reset (set the µC software reset bit SYSCON2.STAT1 and execute the operational firmware).

The download of operational firmware and USB configuration data is done according to the USB Device Class Spec DFU (Device Firmware Upgrade). The validity of the download is signaled to the host on a GET_STATUS request (vendor specific). The bootloader is able to do another download if required.

This switching between download and operational firmware is controlled by bit SYSCON2.STAT2. The switching from bootloader to the operational firmware is only done when triggered by software; it is also possible to switch back to the bootloader for a run-time update.

After this download a USB reset is executed on the SIUC-BA and the µC executes the downloaded firmware in RAM. The firmware provided together with the SIUC-BA will then export a CDC descriptor (USB Communication Devices Class) and identify itself as an ISDN communications device. From this point on normal operation according to CDC is done.

System Identification

The SIUC-BA provides the possibility to identify different system configurations of the device hardware to the host. This enables the host to select the appropriate download firmware for a specific hardware configuration.

Three strap pins are latched during reset and can be read from the Hardware Configuration register HCON ($A3_H$):

- MMOD - Is used to differentiate between shared memory (’0’) and separate memories (’1’) on the external memory interface. This pin has no effect on the HW functions of the SIUC-BA, but is only used as indication for the µC.
- SVN0, SVN1 - Two pins on the auxiliary interface are used as strap pins during reset to set the corresponding SVN0,1 bits in the HCON register.
  In addition to that another 3 bits (SVN4-2) are available in the HCON register which can be overwritten by a value from a connected EEPROM. If no EEPROM is used, SVN5-0 is not overwritten by the µC (i.e. SVN4-2 remains 000_B and SVN1-0 contain the pin strap values). The SVNx bits have no influence on the HW functions of the device but they can be used as an identification number for different HW configurations.
- BMOD0, BMOD1 - The bootmode pins are used to select the firmware download mode (see **Chapter 7.1**) which can be read from the HCON register.

The HCON register is transferred to the host by means of a string descriptor. This allows the host software to identify the system and load the appropriate firmware and driver software.

This identification requires no EEPROM (single chip application). With external EEPROM connected another 3 bits (SVN4-2) can be loaded, i.e. a 5-bit value for System Version Number is available and can be used by host software for identification.

7.3 Memory Modes

Embedded in the SIUC-BA are several memories that become active depending on the MMOD and BMOD strapping and the USB configuration.

- 4 Kbyte Boot Loader ROM
- 16 Kbyte mixed Program / Data RAM that can be configured using the PSIZ, DSIZ registers (e.g. 6 Kbyte Program and 10 Kbyte Data)

The following chapters describe examples for different memory configurations:

- Firmware Download Mode
  The bootloader in internal ROM performs firmware download via USB into internal/ external memory.
- Firmware Execution in internal RAM
  Normal operation mode with no memory extension.

7.3.1 Firmware Download Mode

The memory map used for the download mode is shown in **Figure 107**. The boot loader is present inside the 4 Kbyte Program ROM which is not visible during normal operation mode.

Firmware is downloaded to the 16 Kbyte internal RAM. After the download is finished this memory can be used as program or data memory. The address range 4000H to FFFFH is not used.

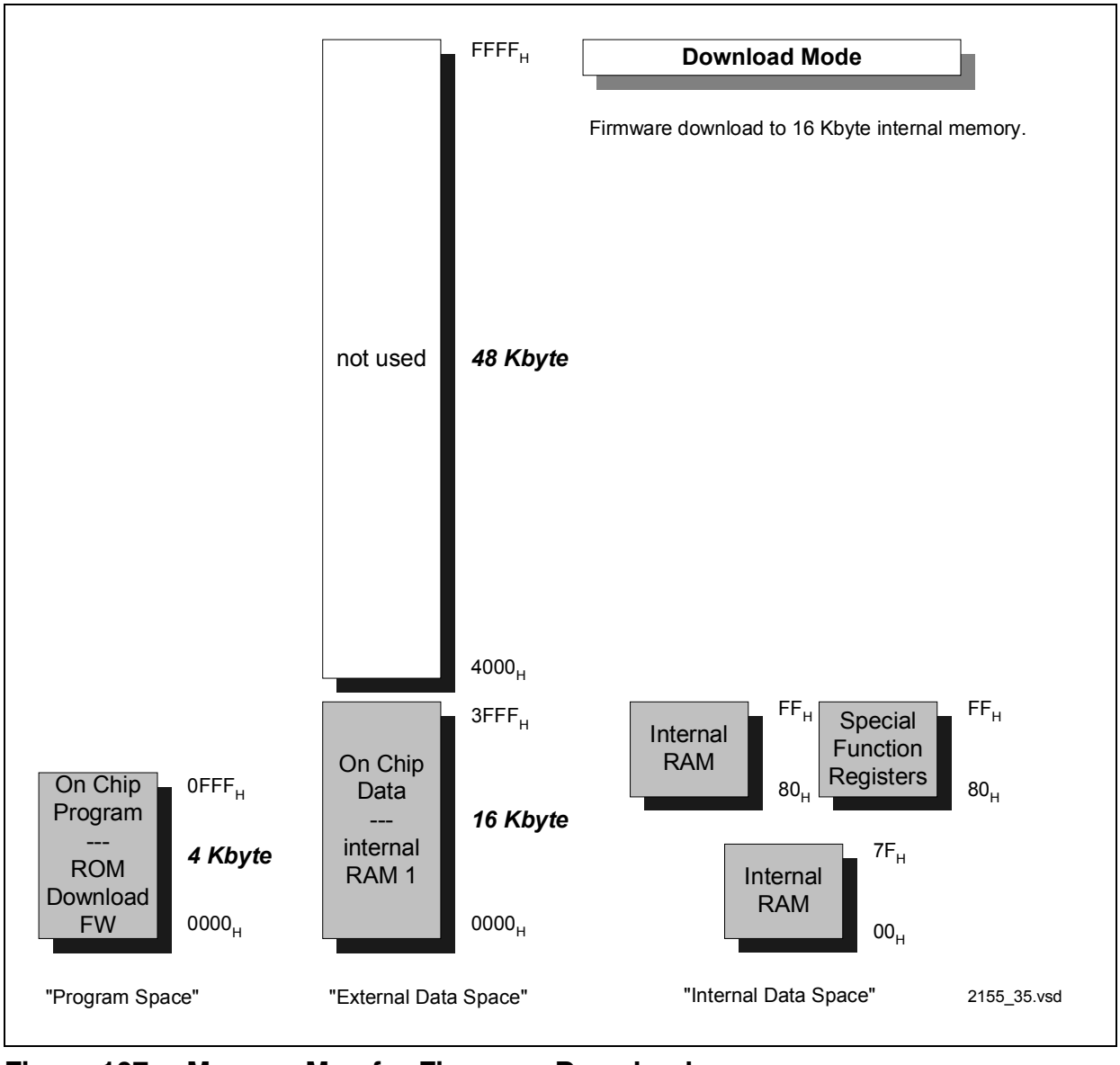


Figure 107 Memory Map for Firmware Download

7.3.2 Firmware Execution in RAM

Figure 108 shows the memory map used during execution of the downloaded firmware (normal operation mode). During execution the bootloader Program ROM is invisible and the downloaded firmware takes over. A switch from download mode to execution mode takes place after the download is finished (SYSCON2.STAT2).

In this example, the 16 kByte internal RAM has been configured to work as 6 kByte Program (RAM1) and 10 kByte Data (RAM2). The size of internal program memory is programmed in PSIZ, the remaining space is used as data memory (DSIZ) which must be programmed before giving a software reset to the µC and switching back the data space to program memory space (normal execution mode).

The onchip data memory (DSIZ) is always located right below the ISDN registers (below address F800H).

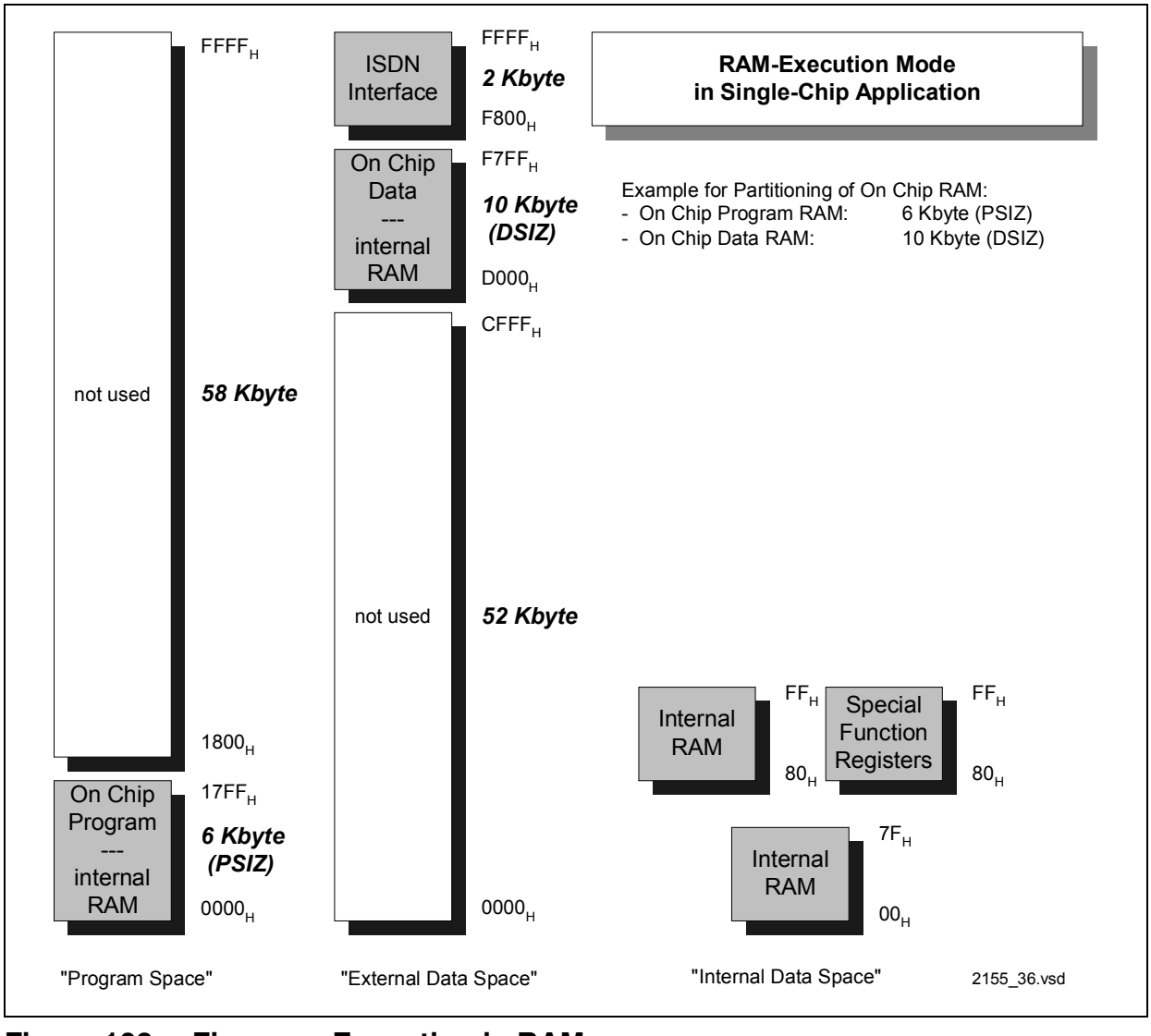


Figure 108 Firmware Execution in RAM

7.4 USB Models

7.4.1 General USB Model in SIUC-BA

Figure 110 gives a general view on the USB models that can be built with the SIUC-BA and its USB device core (UDC). It provides 4 interfaces in addition to the default interface 0, 2 alternate settings for each interface and 7 configurable endpoints besides the default endpoint 0. The assignment of endpoints to an interface is just depending on the application, so the device class supported by the SIUC-BA is determined by the firmware running on it.

The following two chapters already show specific implementations of specific device classes which are used for ISDN data access application. However, the SIUC-BA allows any class specific implementation within its USB resources of interfaces and endpoints.

It should be noted that the UDC counts the functional Interfaces 1 - 4 with the interface numbers Ifc = 0 - 3. However the USB specifications considers the default endpoint 0 as a separate interface (Interface 0).

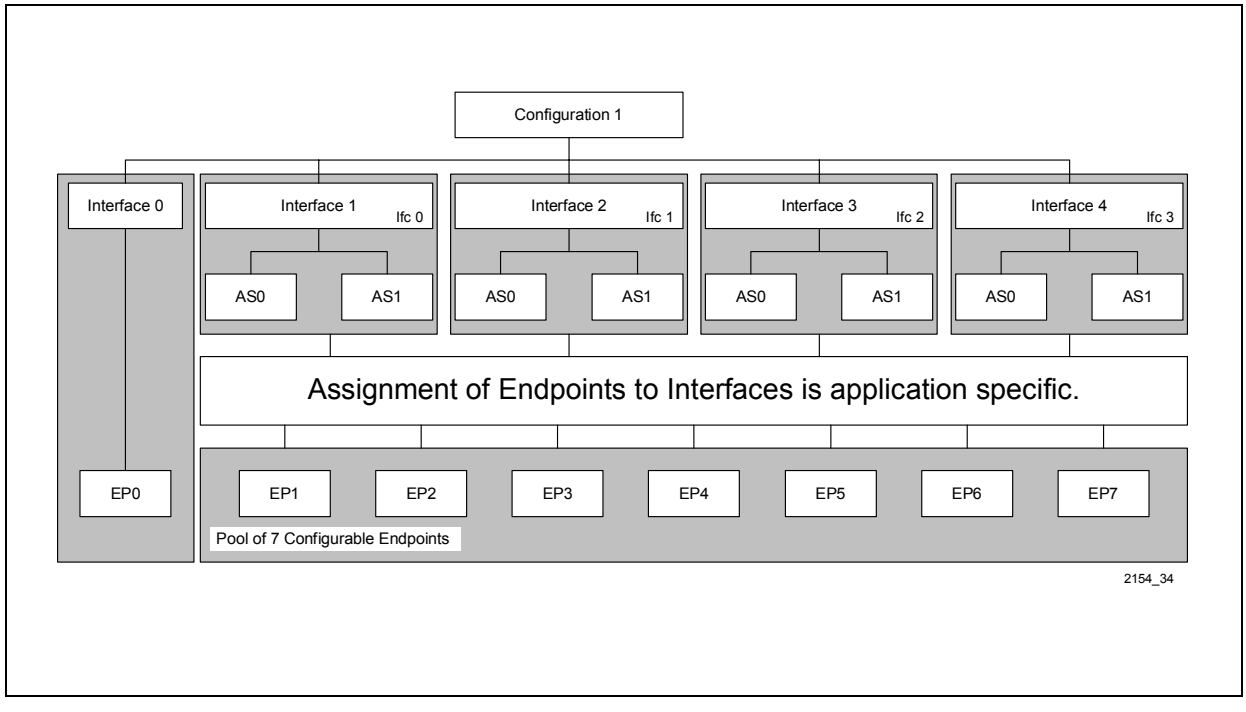


Figure 109 General USB Model

7.4.2 USB Model in Download Mode (DFU)

The SIUC-BA supports download of µC code via USB into internal and external memory. The boot loader program which resides in internal PROM is operating compliant to the USB Class Specification for Device Firmware Upgrade (DFU).

The USB configuration on the SIUC-BA used in this mode is shown in **Figure 110**. Only the default endpoint (EP0) is used for bulk data transfer with a buffer size of 2x64-byte.

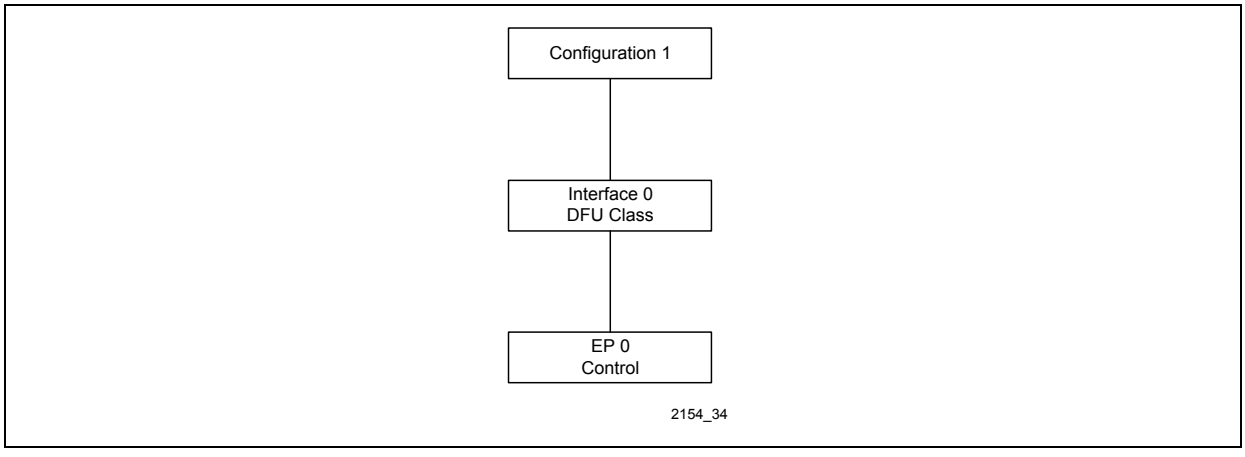


Figure 110 USB Configuration in DFU Mode

7.4.3 USB Model in Operational Mode (CDC)

The firmware that is provided with the SIUC-BA provides a logical link between the ISDN part of the device and the USB interface, and performs the access to the ISDN specific registers for transfer of control/status information and B- and D-channel data (**Figure 111**).

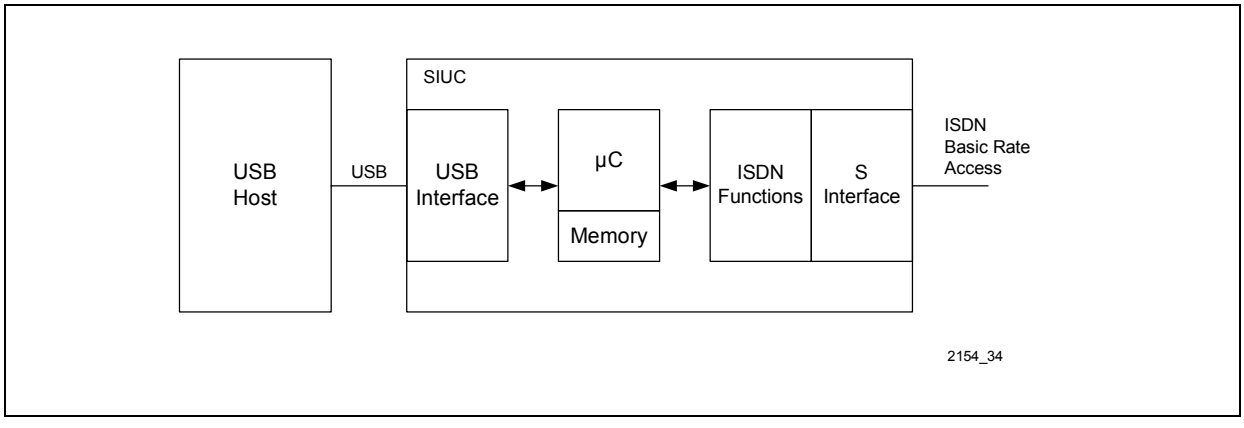


Figure 111 SIUC-BA Firmware Operation

The microcontroller provides the access to the FIFOs and HDLC controllers and exchanges this data between the host and the ISDN registers in a specific way which is specified in the USB Communication Device Class Specification (CDC) Version 1.1. In

other words, the host does not need to initiate the µC to read and write data from the FIFOs, this is done automatically by the provided firmware. The mechanism how and in which format data is exchanged is described in the USB CDC spec.

Communication devices present data to the host in a form defined by another class such as Audio, Data or Human Interface. To allow the appropriate class driver to manage that data, the host is presented with an interface, which obeys the specification of that class. The interface that is required may change according to events that are initiated by the user or the network during a communication session, e.g. the transition from a data only call to a data and voice call.

The functional characteristics include:

- Device Organisation
  - 7 Configurable Endpoints + Default Endpoint (EP0)
  - Construction of Interfaces from Endpoints (4 Interfaces + Default Interface 0)
  - Construction of Configurations from Interfaces and Alternate Settings (SIUC-BA firmware supports one configuration at a time)
- Device Operation
  A communication device has three basic responsibilities:
  - Device Management
  - Call Management
  - Data Transmission

The device uses a Communication Class Interface to perform device management and optionally, call management (see **Figure 112**) via default endpoint 0 which is a bidirectional endpoint (control transfer).

Device management refers to requests and notifications that control and configure the operational state of the device, as well as notify the host of events occurring on the device. Call management refers to setting up and tearing down of calls. This same process also controls the operational parameters of the call.

Data transmission is accomplished using Data Class interfaces for D-channel data and for each of the two B-channels (two unidirectional endpoints each). Another interface may be used to provide synchronisation data to the host for B-channel data transfer. Basically, all device endpoints (EP1-EP7) beside EP0 can support all four USB transfer modes. As data class interface in this firmware model they are used as isochronous or bulk endpoints.

All data class interfaces support two alternate settings, one for operational state (AS1) and a default alternate setting (AS0) which means "no operation". This releases bandwidth in idle mode and conforms to the bandwidth managment requirements on USB.

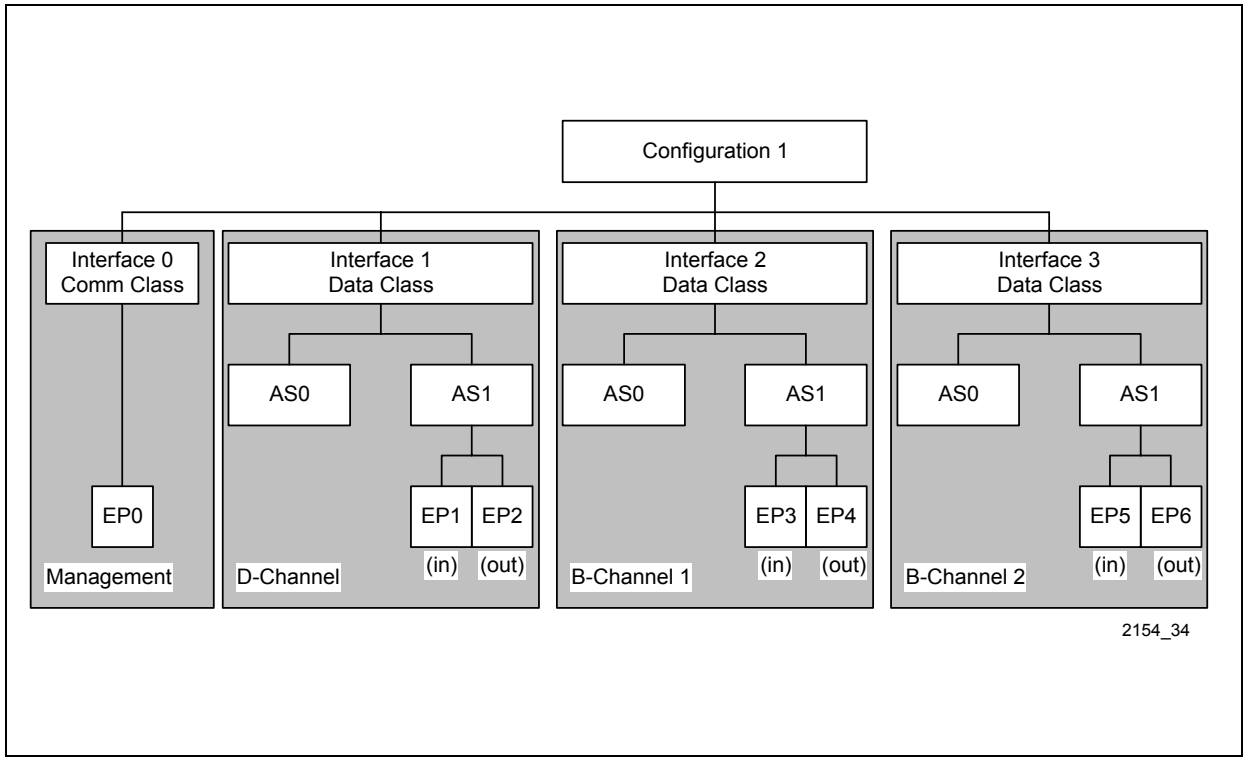


Figure 112 USB Configuration in CDC Mode

Data Buffer Size (CDC)

Within the USB module the data buffer sizes are programmable for each endpoint. The SIUC-BA firmware supporting CDC uses the following data buffer sizes for each endpoint:

EP0: 2x8-byte (Control)

EP1: 2x8-byte (D-Channel)

EP2: 2x8-byte (D-Channel)

EP3: 2x16-byte (B1-Channel)

EP4: 2x16-byte (B1-Channel)

EP5: 2x16-byte (B2-Channel)

EP6: 2x16-byte (B2-Channel)

EP7: 2x8-byte (not used here)

The endpoint 7 (EP7) is not used in the configuration supported by the SIUC-BA firmware from Infineon Technologies. However, it may be used for different implementations than this (e.g. B-channel Sync in CDC spec).

7.4.4 USB Configuration Data

During the USB configuration procedures the SIUC-BA identifies itself by means of several ID values and set of strings. If no EEPROM is connected the values are loaded with default settings from internal ROM. An external EEPROM is used to initialize all values after power on reset with customer specific values.

The SIUC-BA uses two operational modes where it identifies itself with different sets of descriptors:

- DFU (Device Firmware Upgrade) Class - Descriptor
  This descriptor is exported by the bootloader firmware located in ROM which performs the download of the functional firmware into the SIUC-BA's memory. If no EEPROM is connected, the configuration parameters are set by default values in ROM.
- CDC (Communication Device Class) - Descriptor
  After the firmware download is finished and a USB reset is generated by the host, the SIUC-BA exports the CDC descriptor, i.e. the SIUC-BA identifies itself as an ISDN communications device compliant to the USB CDC V1.1 spec. As the configuration parameters are loaded together with the downloaded firmware, a file containing these configuration data is provided by the system manufacturer which are downloaded together with the firmware file. If no file is provided default settings are used.

Table 29 SIUC-BA Configuration Data for USB Descriptors

Value	No EEPROM connected	EEPROM connected
idVendor	DFU:0x058B ("Infineon Technologies") CDC:downloaded by FW (default: 0x058B)	DFU: value from EEPROM CDC downloaded by FW (default: 0x058B)
idProduct	DFU:0x8002 CDC:downloaded by FW (default: 0x0002)	DFU: value from EEPROM CDC downloaded by FW (default: 0x0002)
bcdDevice	DFU:0x0101 (Version 1.1) CDC: downloaded by FW (default: 0x0101)	DFU: value from EEPROM CDC downloaded by FW (default: 0x0101)
bcdUSB	DFU:0x0101 ("USB V1.1") CDC:downloaded by FW (default: 0x0101)	DFU: value from EEPROM CDC downloaded by FW (default: 0x0101)

Table 29 SIUC-BA Configuration Data for USB Descriptors (cont'd)

Value	No EEPROM connected	EEPROM connected
bDeviceClass	DFU:0xFEh ("Application Specific Class") CDC:downloaded by FW (default: 0x02, i.e. USB CDC)	DFU: value from EEPROM CDC downloaded by FW (default: 0x02, i.e. USB CDC)
bDeviceSubClass	DFU:0x01h ("DFU") CDC:downloaded by FW (default: 0x04, i.e. Multichannel Control Model)	DFU: value from EEPROM CDC downloaded by FW (default: 0x04, i.e. Multichannel Control Model)
bDeviceProtocol	DFU:0x00 ("no specific protocol") CDC:downloaded by FW (default: 0xFF; Vendor Specific)	DFU: value from EEPROM CDC downloaded by FW (default: 0xFF; Vendor Specific)
bmAttributes	DFU:0x80 (not self-powered, no rem. wakeup) CDC:downloaded by FW (default: 0x80, not self-powered, no rem. wakeup)	DFU: value from EEPROM CDC downloaded by FW (default: 0x80, not self-powered, no rem. wakeup)
MaxPower	DFU:0x32 (100 mA, see note) CDC: downloaded by FW (default: 0x32)	DFU: value from EEPROM CDC downloaded by FW (default: 0x32)
strManufacturer	DFU:"Infineon Technologies" CDC: downloaded by FW (default: "Infineon Technologies")	DFU: value from EEPROM CDC downloaded by FW (default: "Infineon Technologies")
strProduct	DFU:"SIUC PSB2154" CDC: downloaded by FW (default: "SIUC PSB2154")	DFU: value from EEPROM CDC downloaded by FW (default: "SIUC PSB2154")

Table 29 SIUC-BA Configuration Data for USB Descriptors (cont'd)

Value	No EEPROM connected	EEPROM connected
strSerialNumber	DFU:(no string) CDC:downloaded by FW (no default value)	DFU: value from EEPROM CDC downloaded by FW (no default value)
strConfiguration	DFU:"DFU" CDC:downloaded by FW (default: "CDC")	DFU: value from EEPROM CDC downloaded by FW (default: "CDC")

Note: MaxPower: The actual power consumption of the SIUC-BA is far below 100 mA, however, the real value was not known at the time of ROM mask setting. Moreover, external components add further power consumption. Therefore the worst case value was set for MaxPower to allow low power device operation.
Since the SIUC-BA is a derivative of the full-featured version SIUC-X PSB2154, it uses the same download firmware as SIUC-X, therefore the prodcut string has the unchanged value “PSB2154”.

The configuration data described above is stored in external EEPROM in the following format:

Table 30 Organisation of EEPROM Memory

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	idVendor		idProduct		bcd Device		bcdUSB		bDC	bSC	bDP	bmAt	MP	SVN	55_H	AA_H
16	LEN	strManufacturer														
32	strManufacturer (continued)															
48	LEN	strProduct														
64	strProduct (continued)															
80	LEN	strSerialNumber								LEN	strConfiguration					
96	strConfiguration (continued)															
112																
128																
:	:															
255																

= contents not defined and available for propietary extensions (if not used by strings)

Note: MP = MaxPower
SVN = System Version Number
Byte number 14 and 15 must be programmed to the values 55_H and AA_H, respectively. This is used by the bootloader firmware to detect whether the connected EEPROM has been programmed (i.e. data is valid) or the EEPROM is unprogrammed (i.e. data is invalid). In the latter case the EEPROM is not read by the firmware and the data is ignored.

The total memory space depends on the selected EEPROM (e.g. 1024 or 2048 bit) which has effect on the remaining memory space available for propietary extensions.

All strings consist of one byte for length information (LEN) followed by the character information. The actual start position of strings (e.g. strProduct, ...) is not fixed (as shown above) but depends on the length of the preceeding strings, i.e. a string starts after the last byte of the preceeding string. For strings with the length 0 the corresponding index of a descriptor will be set to "No String" (0).

7.5 Remote Wakeup

Remote Wakeup means an incoming call from the ISDN line can wakeup the SIUC-BA and the host, both of which are in suspend mode.

In order to achieve a reasonably working wakeup functionality it is necessary to support major parts of the D-channel layer 2 and layer 3 protocols (see "Implementation Hints" below). Additionally a power management concept needs to be provided on the system.

The firmware provided with the SIUC-BA is focused on single chip and low-cost ISDN terminal applications and does not include the relevant parts of the D-channel protocol. Therefore the SIUC-BA firmware does not fully support Remote Wakeup. The trade-off is that the host is always waked up on an incoming call regardless whether the call is dedicated for this terminal or another terminal on the same S-bus.

However, SIUC-BA provides an excelent migration path towards the SIUC-X which provides the “Intelligent Remote Wakeup” feature by connection of external memory. Some guidelines and hints for a possible implementation are given below.

Implementation Hints

When the system is in power down mode (USB suspend mode) an incoming call will wakeup the host and the µC in the SIUC-X. As the response time of the host is considerably high due to the long time the OS usually needs to return to operational state, the µC handles major parts of the D-channel layer-2 and layer-3 protocol to setup the call and to provide enough time for the host to resume.

Layer 1

The ISDN Layer 1, based on the ITU standard I.430, is handled mainly by the SIUC-X. To be fully compliant to the TBR3 conformance test suite, the software supports additional timers for activation from TE side and for delayed deactivation.

The SIUC-X correctly handles activation from the network (only activation of S transceiver necessary). It is not required that the software is active before a complete wakeup of the SIUC-X.

Note: It is possible that the layer 1 is always activated because the network or other devices on the S-bus prevent the line from deactivation. In this case wakeup can not be triggered by activation of the layer 1 interface.

Layer 2/3

The Layer 2 and 3 are based on the ITU standards Q.921 and Q.931 (or for Europe ETS 300 125 and ETS 300 102-1/ETS 300 102-2).

The first frame on an incoming call is the Layer 3 message SETUP. The SIUC-X decodes this message (only the HDLC receiver is activated). If it is not the right message (e.g.

called party number is wrong or not relevant information like Time received) the controller ignores it and switches back to suspend mode.

After the controller has found a correct and matching SETUP message it signals RESUME to the host (activation of USB module).

Until the time the host is available, the SIUC-X answers with an "ALERT" (Layer 3). To do this the software sets up a Layer 2 connection for sending Layer 3 frames. At this moment the first data has to be sent over the S-bus (the HDLC transmitter is activated).

The layer 2 consists of a state machine for Q.921 and handling of a TEI value (request, assign and verify).

Power Management

For the power consumption requirements the USB specification differentiates between low power and high power devices:

Table 31 USB Power Consumption Limits

State	High Power Device	Low Power Device
Suspend	2.5 mA	500 µA
Unconfigured	100 mA	100 mA
Configured	500 mA	100 mA

As high power devices can only be operated at root hubs or self powered hubs, most system architectures for bus powered ISDN terminal equipment will be implemented as low power devices.

Remote wakeup requires the µC to handle the call management while the USB host is still in suspend mode. Only if a valid incoming call is detected, a wakeup to the host is initiated. So while the USB is still in suspend mode the µC is operating which results in a higher power consumption of the device.

Due to these reasons a reasonably functioning ISDN terminal device for USB can only be built as self powered device, otherwise the host would see a remote wakeup with any signal on the S interface.

8 General Features

8.1 Clock Generation

The SIUC-BA derives its system clocks from an external clock connected to XTAL1 (while XTAL2 is not connected) or from a 7.68 MHz crystal connected across XTAL1 and XTAL2 (**Figure 113**). A description of the oscillator circuit is provided in **Chapter 10.5**.

8.1.1 USB / Microcontroller

The crystal output passes through a programmable PLL to generate the 48MHz clock for the USB device controller and the microcontroller. The input / output frequency equation is:

$$\text{Clock Output} = \frac{N + 1}{M + 1} \text{ Clock Input}$$

where, N = 0.....31 and M = 0.....15

N and M can be programmed in the PLCONA and PLCONB registers. The default values $N=24_D$ and $M=3_D$ determine the output clock = 25/4 x 7.68 MHz = 48 MHz.

Programming Sequence

After reset the PLL is disabled and the µC is operating at 7.68 MHz crystal frequency. The µC sets the factors M and N and enables the PLL (PCLK). It takes a few ms for the PLL to indicate the status "PLL locked" (bit LOCK is set), after which the µC can switch the PLL clock from "bypass" to "connected" (SWCK). Now the µC is operating at 48 MHz.

Prescaler

For the microcontroller clock a prescaler can be used to divide the 48 MHz clock from the PLL. If enabled (PLCONB.PSCEN=1) the prescaler divides by 2 (PLCON.PSCVAL=0) resulting in 24 MHz frequency or it divides by 1.5 (PLCON.PSCVAL=1) resulting in 32 MHz frequency. This may be used for test purposes or to reduce power consumption if less µC performance is sufficient.

USB Clock Enable

After reset the clock for the USB device controller is disabled. As soon as the PLL is programmed and provides the 48 MHz clock, the µC can switch on the USB clock by setting DCR.UCLK.

8.1.2 S-Transceiver PLL

The IOM clocks are summarized with the respective duty cycles in **Table 32**.

Table 32 IOM Clocks

Mode	DCL	FSC	BCL
TE	o: 1536 kHz (1:1)	o: 8 kHz (1:2)	o: 768 kHz (1:1)

All output clocks are synchronous to the S-transceiver. The FSC signal is used to generate the pulse lengths of the different reset sources software reset, C/I Code, $\overline{\text{EAW}}$ pin and Watchdog (see **Chapter 8.2**).

In TE applications, the transmit and receive bit clocks are derived, with the help of the PLL, from the S interface receive data stream. The received signal is sampled several times inside the derived receive clock period, and a majority logic is used to additionally reduce bit error rate in severe conditions. The transmit frame is shifted by two bits with respect to the received frame.

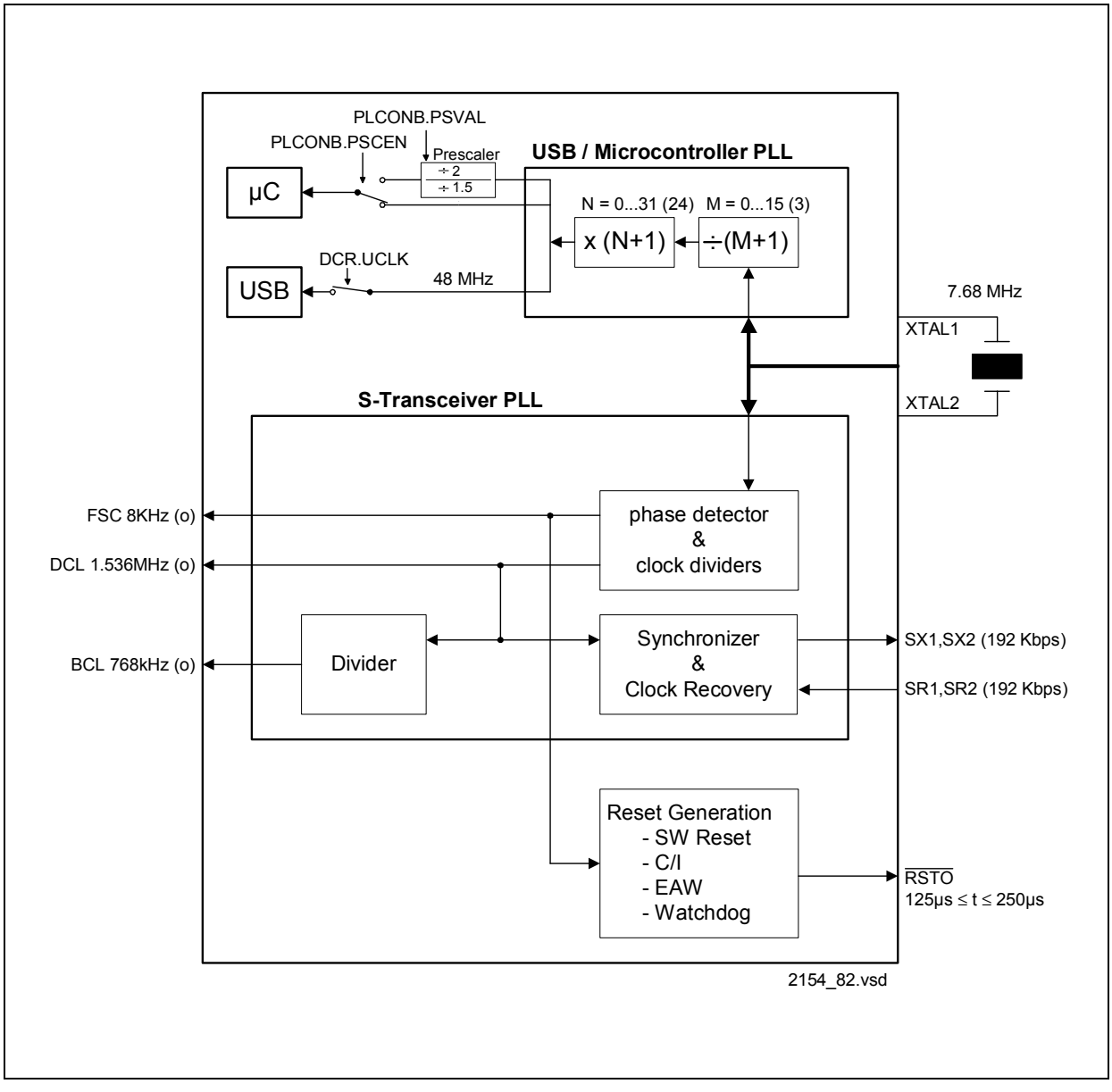


Figure 113 SIUC-BA Clock System

8.1.2.1 Receive PLL (RPLL)

The receive PLL performs phase tracking between the F/L transition of the receive signal and the recovered clock. Phase adjustment is done by adding or subtracting 0.5 or 1 XTAL period to or from a 1.536-MHz clock cycle. The 1.536-MHz clock is than used to generate any other clock synchronized to the line.

During (re)synchronization an internal reset condition may effect the 1.536-MHz clock to have high or low times as short as 130 ns. After the S/T interface frame has achieved the synchronized state (after three consecutive valid pairs of code violations) the FSC output in TE mode is set to a specific phase relationship, thus causing once an irregular FSC timing.

The phase relationships of the clocks are shown in **Figure 114.**

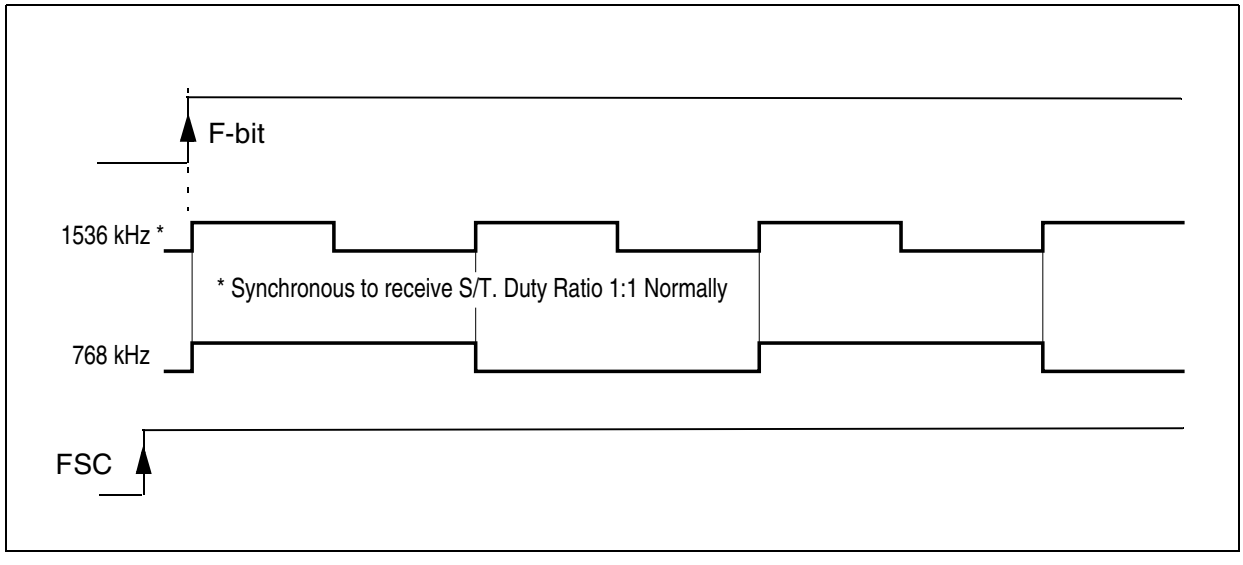


Figure 114 Phase Relationships of SIUC-BA Clock Signals

8.1.2.2 Jitter

The timing extraction jitter of the SIUC-BA conforms to ITU-T Recommendation I.430 (– 7% to + 7% of the S-interface bit period).

8.2 Reset Generation

8.2.1 Hardware Reset Operation

The external hardware reset forces the chip components to a predefined default state. It must be of at least 4 ms duration to allow the 7.68MHz oscillator to stabilize. The Reset input is an active LOW input ($\overline{\text{RESET}}$). An internal Schmitt trigger is used at the input for noise rejection. Following reset, the device performs a complete machine cycle (12 clocks) during which other indirectly reset registers are initialized.

An automatic power-on reset can be obtained by a capacitor connected to VSS and a resistor connected to VDD. After VDD has been turned on, the capacitor holds the voltage level at the reset pin for a specific time to effect a complete reset.

This external reset signal is additionally fed to the multiplexed $\overline{\text{RSTO}}$/SDS output if enabled (see **Chapter 8.2.2**). The length of the reset signal is specified in the **Electrical Specification**.

A reset operation of the USB module can only be achieved under software control.

8.2.2 Software Reset

The microcontroller can issue resets through programmable bits to each of the functional blocks (**Figure 115**). They have different functionality compared to the hardware reset. The Reset pin is not activated.

USB Reset

The USB module has its own reset bit. This software reset, which **MUST** be executed after a hardware reset, is initiated by setting bit DCR.SWR by software. This bit is reset automatically by hardware when the software reset operation of the USB module is finished. Further, with the reset of bit SWR, bit DCR.DINIT is set indicating to the CPU to initialize the endpoints of the USB module.

A USB reset can also be initiated by the upstream USB port (host or hub controller) issuing a reset signalling on the bus according to the USB specification.

µC Reset

The µC has its own reset bit, too. Bit SYSCON2.STAT1 is required when reconfiguring the device from Download mode to Firmware Execution mode (i.e. switching program execution from ROM to RAM).

Reset Output

Depending on the setting of bits RSS1/2 (see MODE1 register) several sources can cause a reset of the MODE1 register. These reset sources are C/I code change (exchange awake), pin $\overline{\text{EAW}}$ (subscriber awake) and watchdog timeout. Additionally, these sources and a reset signal on pin RESET can be output on the low active reset output pin $\overline{\text{RSTO}}$. A programmable software reset (SRES.RES_RSTO) is output on pin $\overline{\text{RSTO}}$ irrespective of the MODE1.RSS1/2 setting.

Watchdog Reset

If the watchdog timer expires a reset is issued to the SIUC-X which has the same effect as a hardware reset (RESET pin). After the selection of the watchdog timer (RSS2/1 = ’11’) an internal timer is reset and started. During every time period of 128 ms the µC has to program the WTC1- and WTC2 bits in the following sequence to reset and restart the watchdog timer:

	WTC1	WTC2
1.	1	0
2.	0	1

If not, the timer expires and a WOV-interrupt (AUXI Register) together with a reset pulse of 125 µs is generated. It may take a few ms after the interrupt until the the reset is going active. Deactivation of the watchdog timer is only possible with a hardware reset, i.e. if RSS2,1 is programmed to ’11’ the value cannot be reprogrammed again afterwards.

In suspend mode all clocks are disabled, so the watchdog is also stopped. However, the watchdog timer continues again when suspend mode is left and clocks are switched on again. It is recommended to write to WTC1,2 according to the above description just before entering suspend mode to avoid the timer to expire accidently right after suspend mode is left again.

ISDN Reset

The **Software Reset Register (SRES)** provides reset bits for each functional block of the ISDN module. A reset to external devices (pin $\overline{\text{RSTO}}$/SDS) can also be controlled in this way. The reset state is activated when the bit is set to ’1’ and the reset state is deactivated again automatically. The address range of the registers which will be reset at each SRES bit is listed in **Figure 115**.

For the ISDN Layer-1 state machine, a hardware or software reset (bit **RES_TR**) brings it to the reset or IDLE state in which the analog components are disabled (transmission of INFO 0) and the S/T line awake detector is inactive. Reset signals should be a minimum of 2 DCL clock cycles wide. These reset events are identical to the C/I code RES with respect to the state machine.

$\overline{\text{RSTO}}$ Reset Source Selection

The selection of the reset sources on pin $\overline{\text{RSTO}}$ can be done with the RSS2,1 bits in the MODE1 register according to **Table 33**. If RSS2,1 = '01' the $\overline{\text{RSTO}}$/SDS pin has SDS functionality and a serial data strobe signal is output at this pin and no reset is output at $\overline{\text{RSTO}}$/SDS. The internal reset sources only set the MODE1 register to its reset value.

Table 33 Reset Source Selection (MODE1.RSS2,1)

RSS2	RSS1	C/I Code Change	$\overline{\text{EAW}}$	Watchdog Timer	SDS Functionality
0	0	--	--	--	--
0	1	--	--	--	x
1	0	x	x	--	--
1	1	--	--	x	--

- **C/I Code Change (Exchange Awake)**
  A change in the downstream C/I channel (C/I0) generates a reset pulse of $125\mu s \leq t \leq 250\mu s$.
- **$\overline{\text{EAW}}$ (Subscriber Awake)**
  A low level on the $\overline{\text{EAW}}$ input starts the oscillator from the power down state and generates a reset pulse of $125\mu s \leq t \leq 250\mu s$.
  (Besides that $\overline{\text{EAW}}$ can also generate an interrupt ISTA_TR.LD to the µC)
- Watchdog Timer
  If the watchdog timer expires a reset is generated as described above.
  If RSS2,1 is programmed to '11' the value cannot be reprogrammed afterwards, i.e. the watchdog timer can only be stopped again by a reset.
- SDS Functionality
  SDS is not related to reset functionality. A serial data strobe signal can be programmed in SDS_CR and output on pin $\overline{\text{RSTO}}$/SDS (see **Chapter 5.5.3**).

Note: A reset on $\overline{\text{RSTO}}$ can always be activated by setting the software reset bit SRES.RES_RSTO irrespective of the MODE1.RSS1,2 setting (i.e. even if SDS functionality is selected).

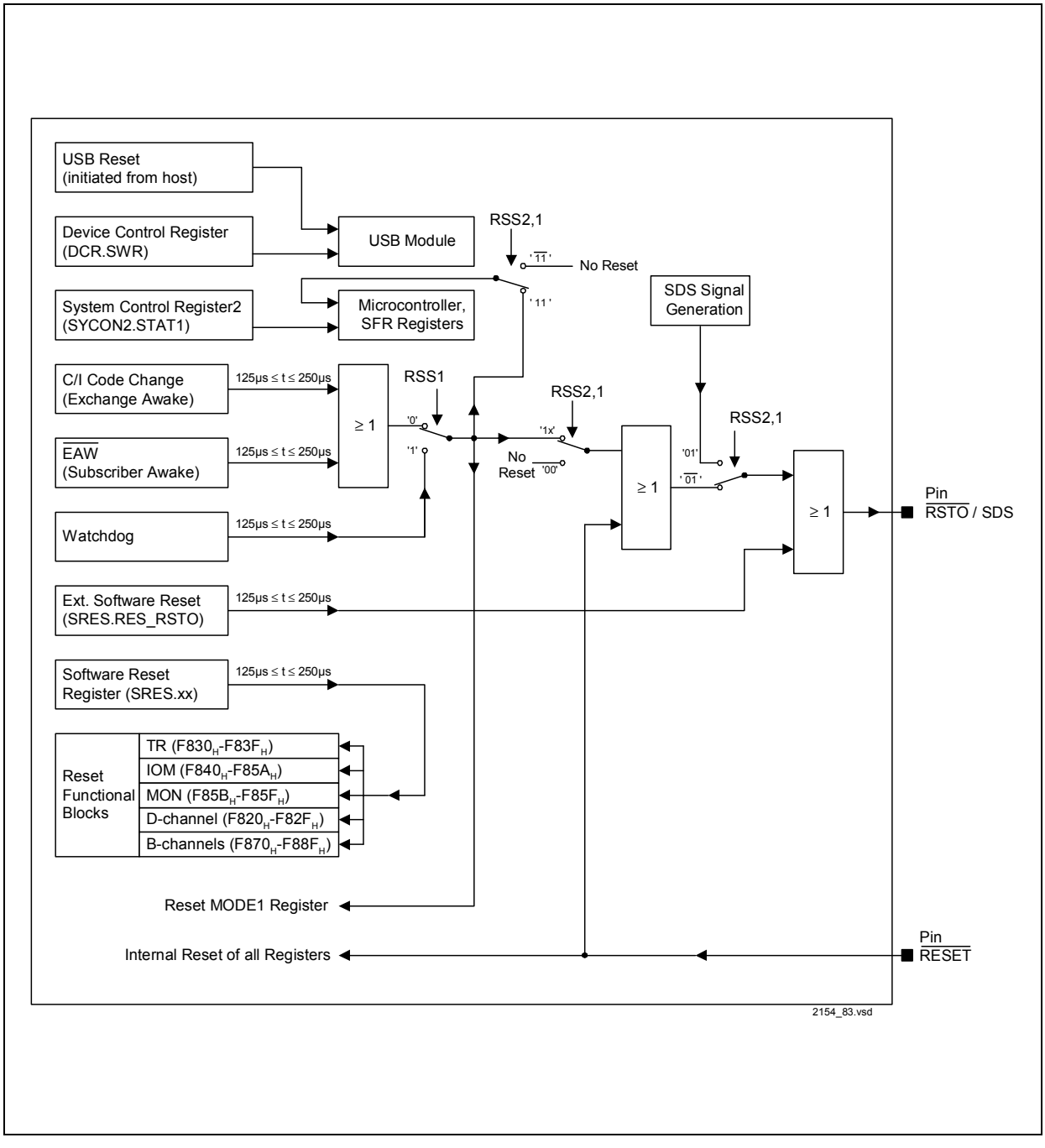


Figure 115 Reset Generation

8.3 Auxiliary Interface

8.3.1 Mode Dependent Functions

The AUX interface provides various functions as shown in **Table 34**. After reset the pins are switched as inputs until further configuration is done by the microcontroller. The system designer must use this interface with care in case the SPI functionality is required simultaneously with other auxiliary functions, for example if an EEPROM is connected (chip select signal on AUX3).

The registers of the auxiliary interface are located in the ISDN register map and described in **Chapter 5.8.3**.

Table 34 AUX Pin Functions

Pin	Function
AUX0	AUX0 (i/o)
AUX1	AUX1 (i/o)
AUX2	AUX2 (i/o)
AUX3	AUX3 (i/o) / ELD (i) / $\overline{\text{SCS}}$ (o)
AUX4	AUX4 (i/o) / SDI (i) / MBIT (i/o) / SVN0 (i)
AUX5	AUX5 (i/o) / SDO (o) / FBOUT (o) / SVN1 (i)
AUX6	INT1 (i/o) / SCK (o)
AUX7	INT2 (i/o) / SGO

AUX0-5

These pins can be used as programmable I/O lines.

As inputs (AOE.OEx=1) the state at the pin is latched in when the host performs a read operation to register ARX.

As outputs (AOE.OEx=0) the value in register ATX is driven on the pins with a minimum delay after the write operation to this register is performed. They can be configured as open drain (ACFG1.ODx=0) or push/pull outputs (ACFG1.ODx=1). The status (’1’ or ’0’) at output pins can be read back from register ARX, which may be different from the ATX value, e.g. if another device drives a different level.

$\overline{\text{INT1}}$, $\overline{\text{INT2}}$ and LED Ports

In all modes two pins can be used as programmable I/O with optional interrupt input capability (default after reset, i.e. both interrupts masked).

The $\overline{\text{INT1/2}}$ pins are general input or output pins like AUX0-5 (see description above). In addition to that, as inputs they can generate an interrupt to the host (AUXI.INT1/2) which is maskable in AUXM.INT1/2. The interrupt input is either edge or level triggered (ACFG2.EL1/2).
As outputs both pins can directly be connected to an LED with preresistor.

For both pins AUX6/7 internal pull-up resistors are provided if the pin is configured as input or as output with open drain chracteristic. The internal pull-ups are disabled if output mode with push/pull characteristic is selected.

FBOUT

AUX5 is multiplexed with the selectable FSC/BCL output FBOUT, i.e. the host can select either standard I/O characteristic (ACFG2.A5SEL=0, default) or FBOUT functionality (ACFG2.A5SEL=1). FBOUT provides either an FSC (ACFG2.FBS=0, default) or BCL signal (ACFG2.FBS=1) which are derived from the DCL clock (also see **Chapter 8.1**).

SGO

AUX7 provides the additional capability to output the S/G bit from the IOM-2 interface by setting ACFG2.A7SEL=1.

MBIT

If ACFG2.A4SEL is set to “1” the pin AUX4 is used for Multiframe Synchronization output (see **Chapter 5.2.3**) and all configuration as general purpose I/O pin is don’t care.

System Version Number (SVN0, 1)

These two pins are used as input during reset to latch a logical ’1’ (external pull-up resistors required) or ’0’ (internal pull-downs provided) to the SVN0, 1 bits in register HCON. The firmware transfers this information to the host so the system configuration built around the SIUC-BA can be identified by the host and the appropriate firmware and software drivers can be loaded. For further information refer to **Chapter 7.2**.

SPI-Signals (ELD, SDI, SDO, SCK, $\overline{\text{SCS}}$)

These signals are multiplexed with the functions described above. As long as the SPI interface is not accessed by activating the chip select signal $\overline{\text{SCS}}$, the non-SPI functionality is available. The SPI interface is described in detail in **Chapter 8.3.2**.

8.3.2 SPI Interface

The SIUC-BA includes a Serial Peripheral Interface bus to connect an external EEPROM. This EEPROM can optionally be used to load any vendor specific data. The physical interface for the EEPROM is a serial 3-wire interface to connect standard memory devices like 25C20, supporting devices up to 4K. The following signals are used:

SDI, SDO, SCK

During all EEPROM transactions, these pins are used as data input pin, data output pin and clock pin, respectively. The clock signal is driven by SIUC-BA. Data is clocked out on the negative edge of SCK and clocked in on the positive edge of SCK.

$\overline{\text{SCS}}$

This is the active low Chip Select pin for the EEPROM.

ELD

This pin is strapped during the hardware reset and stored in the EEPSL register to indicate to the µC that USB IDs should be loaded from EEPROM after reset. A logic 1 indicates to the µC whether an EEPROM is connected. If set to 1, the µC will read the IDs from the external EEPROM and load the corresponding ID registers. If set to 0 no EEPROM access is performed and the ID registers are loaded with default values. During normal operation, a connected EEPROM can be accessed through the EEPROM Configuration registers, e.g. to read/write any proprietary data irrespective of the state of ELD after a reset.

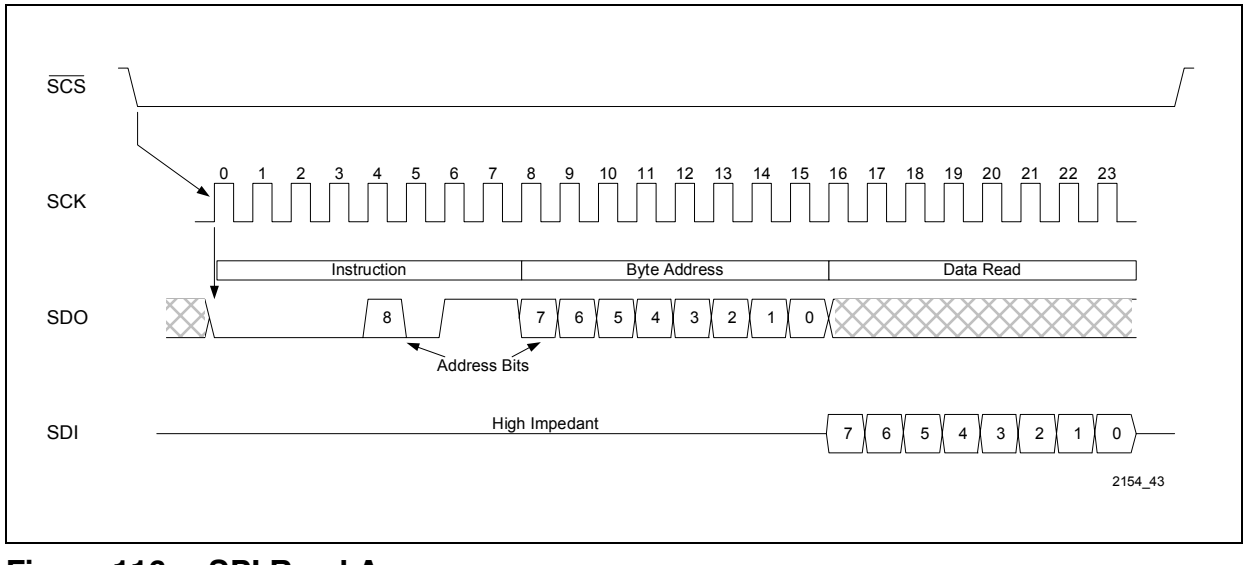


Figure 116 SPI Read Access

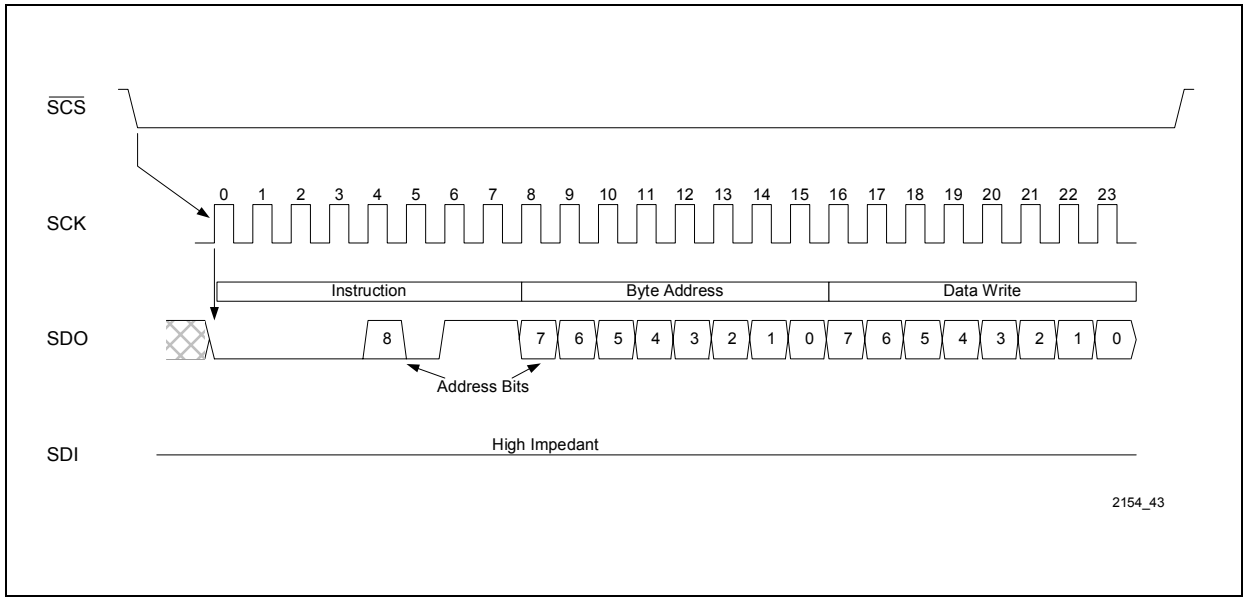


Figure 117 SPI Write Access

8.3.2.1 Direct Microcontroller Access to the EEPROM

EEPROM Configuration registers are implemented to enable access to the EEPROM directly. For example, the microcontroller can write the EEPROM contents during board manufacturing and read back the contents for verification or it can use the EEPROM to store customized information.

To control all EEPROM transactions the following registers are used. The registers are located at address 93h - 97h in the special function register address space.

- **EEPCMD - EEPROM Command Register**
  The microcontroller writes the command for the next EEPROM transfer to this register. The following SPI commands are supported (all other codings are reserved):
  ”00000110” ⇒ WREN,Set Write Enable Latch
  ”00000100” ⇒ WRDI,Reset Write Enable Latch
  ”00000101” ⇒ RDSR,Read Status Register
  ”00000001” ⇒ WRSR,Write Status Register
  ”0000A011” ⇒ READ,Read Data from Memory Array
  ”0000A010” ⇒ WRITE,Write Data to Memory Array
  "A" represents the MSB address bit A8, the lower 8 bits (A7-0) is contained in the EEPROM Byte Address Register EEPADR.
- **EEPADR - EEPROM Byte Address Register**
  For read and write transactions to the connected EEPROM, the EEPROM byte address is written to this register before the transaction is started. The MSB address bit A8 is contained in the EEPROM command register EEPCMD.
- **EEPDAT - EEPROM Data Register**
  For the transactions

- 'Write Status Register - WRSR' and
- 'Write Data to Memory Array - WRITE'

the data that has to be transferred to the EEPROM is written to this register before the transaction is started.
After the transactions

- 'Read Status Register - RDSR' or
- 'Read Data from Memory Array - READ'

are finished (ESTA bit is reset and an interrupt is generated), the byte received from the EEPROM is available in this register.

- **ESTA - EEPROM Start Bit**
  The microcontroller sets this bit to '1' in order to start an EEPROM transaction. It must be ensured that the EEPCMD, EEPADR and EEPDAT registers are configured with the correct values before the transaction is started.
  After the transaction is finished the ESTA bit is reset by hardware and an EEPROM Control Interrupt ECINT is released to the microcontroller (if enabled in register IEN0).

To start a read/write transaction to a connected EEPROM, the microcontroller sets the EEPROM command EEPCMD, the EEPROM byte address EEPADR (for EEPROM read/write data commands), the data EEPDAT that is to be written to the EEPROM and finally sets the EEPROM Start bit ESTA.

The EEPROM Command is then interpreted and the SIUC-BA starts the read/write transaction to the connected EEPROM. After the transaction has finished, the ESTA bit is reset to '0' and an ECINT interrupt is generated to the µC (EEPINT-EEPROM Interrupt Control Register), if the interrupt is enabled (IEN0.ES=1).

If the EEPROM Command is a read command (Read Status Register, Read Data from Memory Array), the byte that is read out of the EEPROM is available in the EEPDAT register when the transfer is finished.

8.3.3 SPI Registers

The **EEPROM Interrupt Control Register (EEPINT)** is described in the section on interrupts.

8.3.3.1 EEPCMD - EEPROM Command Register

Reset value: 00_H Address: 94_H

7	6	5	4	3	2	1	0
.7	.6	.5	.4	.3	.2	.1	.0
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Function
EEPCMD.7 - EEPCMD.0	**EEPROM Command** Command for the next EEPROM transaction. The following SPI commands are supported: ”00000110” ⇒ WREN,Set Write Enable Latch ”00000100” ⇒ WRDI,Reset Write Enable Latch ”00000101” ⇒ RDSR,Read Status Register ”00000001” ⇒ WRSR,Write Status Register ”0000A011” ⇒ READ,Read Data from Memory Array ”0000A010” ⇒ WRITE,Write Data to Memory Array "A" contains the MSB address bit A8 for read/write transactions.

8.3.3.2 EEPADR - EEPROM Byte Address Register

Reset value: 00_H Address: 95_H

7	6	5	4	3	2	1	0
.7	.6	.5	.4	.3	.2	.1	.0
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Function
EEPADR.7 - EEPADR.0	**EEPROM Byte Address** Byte address for the next EEPROM transaction. The MSB address bit A8 for read/write transactions is contained in the EEPROM command byte (see above).

8.3.3.3 EEPDAT - EEPROM Data Register

Reset value: 00_H Address: 96_H

7	6	5	4	3	2	1	0
.7	.6	.5	.4	.3	.2	.1	.0
rw	rw	rw	rw	rw	rw	rw	rw

Bit	Function
EEPDAT.7 - EEPDAT.0	**EEPROM Data** Transaction with a read command: After the transaction has finished, the register contains the byte that has been read from the EEPROM. Transaction with a write command: The contents of this register will be written at the relevant EEPROM address, after the ESTA start bit is set.

8.3.3.4 EEPSL - EEPROM Start / Load Register

Reset value: 00_H Address: 97_H

7	6	5	4	3	2	1	0
ELD	0	0	0	0	0	0	ESTA
r	r	r	r	r	r	r	rw

Bit	Function
ELD	**EEPROM Load** The ELD pin is latched with the falling edge of reset. A logic 1 (which overwrites the internal pulldown resistor) indicates that an EEPROM is connected to the SIUC-BA and that the µC could load USB IDs from it.
ESTA	**EEPROM Start Bit** Setting this bit to ’1’ starts an EEPROM transaction with the EEPROM Command, EEPROM Data and the EEPROM Byte Address. After the transaction is finished, this bit is reset by hardware to indicate that another transaction may be started by the microcontroller.

8.4 Voltage Regulator

The SIUC-BA provides an onchip voltage regulator which allows direct connection to the power lines of the USB port using some external components. In this way bus powered operation without an external voltage regulator can be realized.

The functional part of the SIUC-BA is isolated from the onchip regulator, so the regulated output voltage must be connected to the power supply pins of the SIUC-BA. Further devices (e.g. external SRAM) can be connected as long as the total power budget of the regulator is not exceeded. **Figure 118** shows how the external components are to be connected to the VREG1/2 pins.

If self-powered operation is required (separate power supply) the voltage regulator is not used and its pins VREG1/2 are left open.

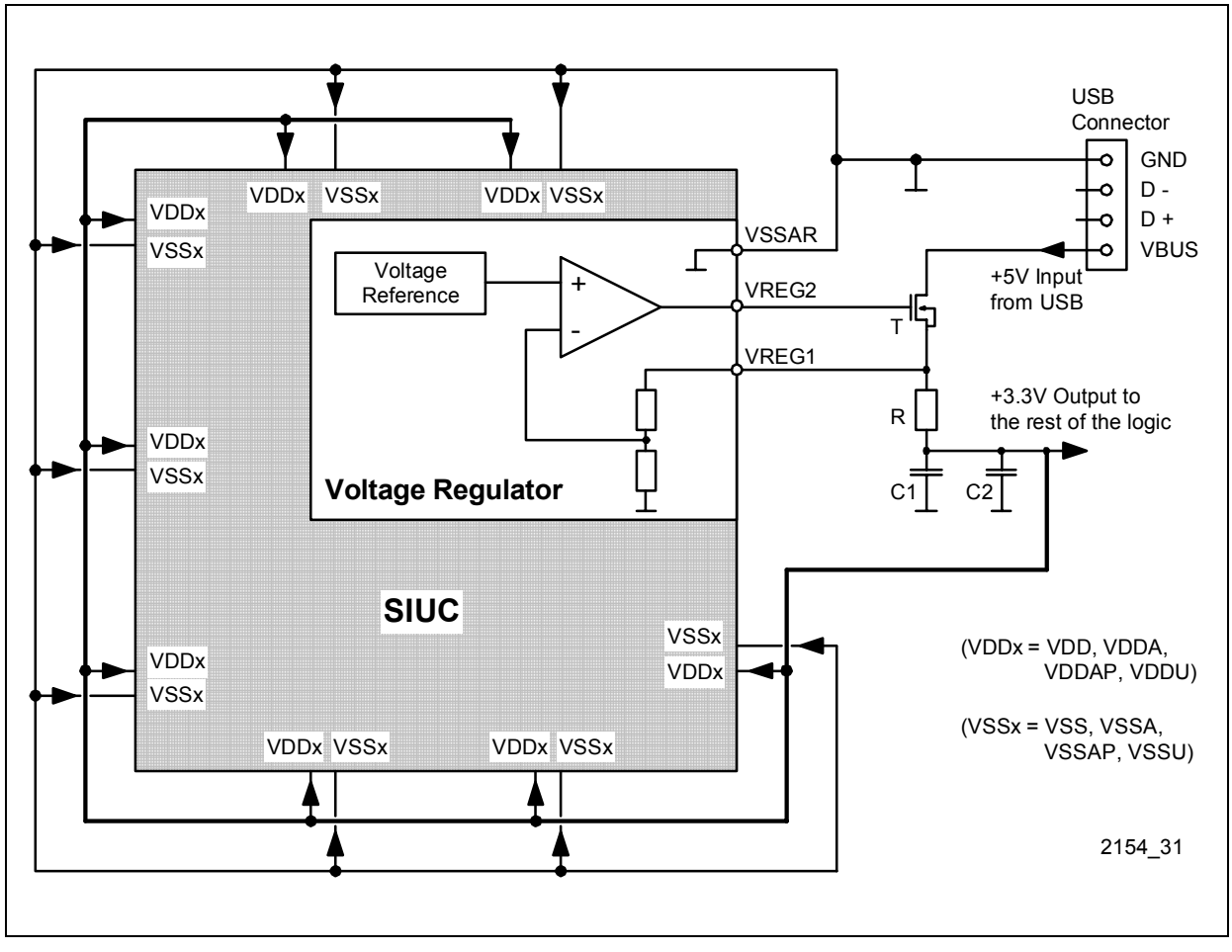


Figure 118 External Circuitry of the Voltage Regulator

Note: The figure does not show any general power supply requirements such like blocking capacitors.

9 Operational Description

9.1 Configuration of Functional Blocks

This chapter contains a description how the functional blocks of the SIUC-BA must be programmed after reset. A detailed description in which sequence the programming is to be done can be found in **Chapter 9.3**.

On the right side the relevant pins, registers (in bold), register bits and cross references for a detailed description are given:

Register Initialization

To avoid malfunctions on unused port signals, the I/O direction must be programmed right after reset

write value to register:
03_H => reg. 80_H
00_H => reg. $A0_H$
00_H => reg. $B0_H$

Firmware Operation Mode

After a reset of the SIUC-BA, the microcontroller starts operation depending on the strapping of the $\overline{EA}$ and BMOD pins. For this purpose the value of the BMOD pins is available in the internal special function register HCON (Hardware Configuration Register).
In downalod mode the bootloader in ROM is executed first and only after the firmware download is finished, execution of operational firmware starts.

HCON
BMOD0/1
$\overline{EA}$=1,
(p. 44, 295)

PLL Configuration

Right after reset the PLL is bypassed and the µC is operating at crystal speed 7.68 MHz. The PLL must be programmed before operation at 48 MHz is possible.

PLCONA/B
M3-0, N4-0, PCLK, LOCK, SWCK
(p. 312, 45)

USB Clock Enable

After the PLL is programmed (see above) the 48 MHz clock for the USB device core (UDC) must be enabled by setting UCLK.

DCR.UCLK
(p. 312, 91)

Internal Memory Access Enable

In order to enable access to onchip RAM the XMAP0 bit must be set.

SYSCON1.XMAP0
(p. 21, 50)

Program and Data RAM Partitioning

After download is finished and firmware execution is switched from ROM to RAM, the partitioning of internal RAM into program and data RAM (single chip applications) must be done via the registers PSIZ and DSIZ.

PSIZ

DSIZ

(p. 22, 48, 49)

Switching Program Execution from ROM to RAM

In download mode the bootloader performs the download to internal RAM. When this is finished program execution is switched from ROM to RAM which is done by first setting STAT2 to switch from ROM to RAM and then setting STAT1 to reset the µC.

SYSCON2

STAT1, STAT2 (p. 22, 51)

The description above contains a minimum set of configuration registers that is loaded with specific values. Other configuration registers may optionally be programmed depending on the required features and operating modes, that means the SIUC-BA has reached normal operation mode and the USB and ISDN parts can be programmed to operate in the way required by the individual application.

The microcontroller may switch between power-up and suspend mode. This does not influence the register contents, i.e. the internal states remain stored. In suspend mode however, all internal clocks are disabled, and no interrupts can be forwarded to the microcontroller. This state is used as a standby mode, when there is no activity on either the USB or the S-interface, thus minimizing power consumption. The device can be waked up from this suspend state by different sources which can be enabled before entering suspend mode (see **Chapter 9.3.6**).

The communication between the microcontroller and IOM-2, S-Interface is done via a set of directly accessible 8-bit registers. The microcontroller sets the operating modes, controls function sequences and gets status information by writing or reading these registers (Command/Status transfer). Each of the two B-channels is controlled via an equal, but totally independent register file. Additional registers are available for D-channel control and the Auxiliary interface.

Data transfer between the microcontroller memory, IOM-2, S-Interface and USB for both transmit and receive direction is controlled by interrupts.

Special events from the ISDN module are indicated by means of a 8 interrupt outputs, which requests the microcontroller to read status information or transfer data. Events on the USB interface are indicated by means of 2 interrupts.

9.2 Power Saving Modes

The SIUC-BA provides two power saving modes:

– idle mode
– suspend mode

If the suspend mode and the idle mode are set at the same time, suspend takes precedence.

This chapter contains a general overview on the power saving modes, **Chapter 9.3** describes the sequence of operations to reach these modes.

9.2.1 Idle Mode

The idle mode is configured in the PCON register in the following way.
First the Idle_Mode_Enable bit IDLE is set and
with the following instruction the Idle_Start bit IDLS is set.

PCON
IDLE, IDLS
(p. 41)

The idle mode is characterized by the following behaviour:

- the oscillator continues to run
- the µC is gated off from the clock signal
- the rest of the modules are still provided with the clock
- the µC status is preserved in its entirety (i.e. stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode)

The reduction of power consumption, which can be achieved by this feature depends on the number of peripherals running. If all the peripherals are disabled or stopped, maximum power reduction can be achieved. The idle mode is a useful feature which makes it possible to "freeze" the processor's status - either for a predefined time, or until an external event reverts the controller to normal operation.

The following functions can optionally be disabled to achieve several levels for power consumption in idle mode:

USB Transmitter Power Down

The transmitter can be disabled by setting TPWD which is usefull if no data is sent on the USB but receive data is still monitored on the bus.

DPWDR
TPWD
(p. 93)

USB Receiver Power Down

The receiver can be disabled by setting RPWD which is usefull if no data needs to be received from the bus. However, if wakeup from USB (i.e. resume initiated by the host) should wakeup the µC, the receiver should not be powered down.

DPWDR
RPWD
(p. 93)

USB Clock Disable

The clock for the USB controller can be disabled by resetting UCLK, so the USB part is in its lowest power down state.

DCR
UCLK
(p. 91)

ISDN Functions Power Down

As the ISDN module is still provided with clocks in idle mode any of the functional units (transceiver, IOM-2, HDLC controllers) which are all disabled for suspend mode (clocks are off) can individually be disabled to reduce power consumption (see **chapter 9.2.2.1**). However, the level detect circuit of the transceiver must be enabled if incoming calls from the line should be detected.

9.2.2 Suspend Mode

In order so switch the SIUC-BA to suspend mode (power down mode) special care must be taken when switching off the clocks. Several conditions must be met so the oscillator can be powered down (**Figure 119**).

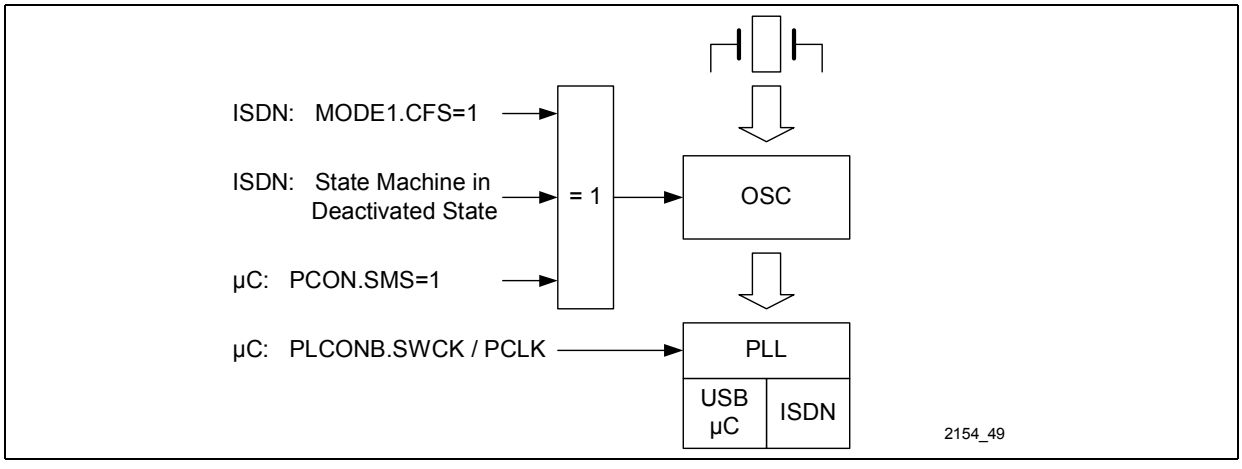


Figure 119 Clocks in Suspend Mode

The following programming procedure must be followed to reach suspend state:

- Disable IOM clocks in power down mode (MODE1.CFS=1) and set layer 1 state machine to "Deactivated State" (write the C/I command DI and wait for C/I indication DC).
- Enable the sources that can wakeup SIUC-BA from suspend mode (WCON register). If no wakeup sources are enabled, the suspend mode can only be left by a reset.
- Set USB transmitter to power down mode (DPWDR.TPWD=1). Do not set USB receiver to power down mode (DPWDR.RPWD=0) if wakeup from USB (resume on the bus) should be possible. Then switch off the USB clock (DCR.UCLK=0).
- First the PLL is switched to bypass mode (PLCONB.SWCK=0), i.e. the µC operates at crystal speed 7.68 MHz, and then the PLL for USB and µC is switched off (PLCONB.PCLK=0).
- Finally the microcontroller switches itself to suspend mode by first writing PCON.SME=1 and with a successive access setting PCON.SMS=1.
  Only with the last access the oscillator is finally powered down.

MODE1.CFS
C/I = DIU, DC
(p. 254)

WCON
(p. 340)

DPWDR
TPWD, RPWD,
(p. 93)

DCR .UCLK
(p. 91)

PLCONB
SWCK, PCLK
(p. 45)

PCON
SME, SMS
(p. 41)

The suspend mode is characterized by the following behaviour:

- the XTAL oscillator is deactivated
- all functions of the microcontroller are stopped
- only the contents of the onchip IRAM, XRAM and SFRs are maintained
- all internal pull up and pull down resistors are switched off

The USB module enters the suspend state when it detects no activity on the USB bus for more than 3 ms. The suspend mode can be left in one of the following ways which is enabled in the WCON register (also see **Chapter 6.3** and **Chapter 9.3.6**):

- a low signal at pin $\overline{\text{INT0}}$, $\overline{\text{INT1}}$, $\overline{\text{INT2}}$ or $\overline{\text{EAW}}$
- a C/I code change
- any activity on the S bus
- any activity on the USB bus
- an active reset signal at $\overline{\text{RESET}}$

If the suspend state is left due to one of the events described above (except "reset") the device is waked up.

Using the reset to leave suspend mode puts the microcontroller with its SFRs into the reset state. Using any of the other sources to exit the suspend mode maintains the state of the SFRs, which has been frozen when suspend mode was entered.

In suspend mode the internal pull up resistor at $\overline{\text{INT0}}$ is switched off to avoid high leakage current. If $\overline{\text{INT0}}$ is used as wakeup source, the port must be programmed to push pull characteristic, or if open drain characteristic is required, an external pull up must be connected. If $\overline{\text{INT0}}$ is not used as interrupt during suspend, the port should be programmed as output by writing a ’1’ to it.

If the IOM-2 interface is not used in the application system, the open drain characteristic of the data lines DU and DD must be disabled (IOM_CR.DIS_OD=1) as the missing pull up resistors would cause malfunction in suspend mode.

Further information on the topics enabling/disabling certain wakeup sources and on suspend state with disabled wakeup capability can be found in **Chapter 6.3**.

9.2.2.1 ISDN Module Power Down

Configuring the suspend mode as described above has no effect on the configuration of the ISDN module. The functional blocks there must individually be programmed to power down mode which is done before going into suspend mode.

However, setting some of these blocks to power down mode may also be used to implement an idle mode (see **Chapter 9.2.1**).

Power Down

The ISDN module has configuration bits that can initiate a local power down when the rest of the chip is in the active state. There are 2 modes of entering power down:

- Giving the C/I command 1111 = DIU - Deactivation Indication Upstream (written to CIX0) when the layer-1 state machine is enabled. No signal (info 0) is now present on the S bus (asuming no other device is transmitting on S). **CIX0**
- Setting the TR_CMD.PD bit when the layer-1 state machine is disabled (TR_CONF0.L1SW = 1). **TR_CMD**.PD **TR_CONF0**.L1SW

During power down, if MODE1.CFS = 0, clocks to/from the module are active. In this case Power Up and Power Down are functionally identical except for the indication PD = 1111 and PU = 0111. If MODE1.CFS = 1, only the analog level detector is active and all clocks to/from the module are stopped. **MODE1**.CFS

Wake Up

Wake up can take place either from the exchange or the terminal.

- If TR_CONF0.LDD = 0, activation initiated from the exchange side will cause the clock signal to be provided immediately. If TR_CONF0.LDD = 1, the µC has to take care of an interrupt caused by the level detect circuit (ISTATR.LD). The µC must then set this bit to 0 to activate the S/T interface again. **TR_CONF0**.LDD **ISTATR**.LD
- From the terminal side, wakeup can be initiated by a set/reset of IOM_CR.SPU and writing TIM to the CIX0 register or by resetting MODE1.CFS. **IOM_CR**.SPU **CIX0** **MODE1**.CFS

Transceiver

TR_CONF0.DIS_TX can be used to disable the transmitter (while the receiver is still active) and TR_CONF0.DIS_TR disables the complete transceiver resulting in minimum power consumption. In this mode DCL and FSC are inputs. When the S-transceiver is completely switched off, no activation from the line can be detected. To overcome this, the level detect circuit can be enabled before switching off the transceiver (see above). The power consumption is minimal, but only a level on the line is detected, the S-transceiver will not automatically start to setup layer-1. The transceiver has to be switched on first.

TR_CONF0

DIS_TX
DIS_TR

IOM-2

IOM_CR.DIS_IOM is used to disconnect external devices from IOM-2. Setting IOM_CR.SPU pulls the DU line low. This will force connected layer-1 devices to deliver IOM clocking. After a subsequent ISTA.CIC interrupt, and reception of the C/I code PU, the µC writes an AR or TIM command as C/I code in the CIX0 register and resets the SPU bit.

IOM_CR

DIS_IOM
SPU

ISTA.CIC

CIX0

9.3 Sequence of Operations

The SIUC-BA uses 4 operational states which are

- Reset State (after power on reset, hardware reset)
- Active Mode (normal operation)
- Idle Mode (chapter 9.2.1)
- Suspend Mode (chapter 9.2.2)

A well defined procedure must be executed for going from one state to another, so the following state transitions and required sequence of operations are described in the chapters below.

- Reset to Active
- Active to Idle
- Idle to Active
- Active to Suspend
- Suspend to Active

Of utmost importance is initialization of the USB module.

The previous chapters describe in detail how each functional unit is configured. The following chapters describe the sequence in which configuration has to take place.

9.3.1 Reset to Active

- After a hardware reset which lasts 4 ms, all chip components excluding the USB module are initialized. A hardware reset operation puts only the internal µC interface of the USB module and its memory management unit into a well defined reset state.
- The microcontroller starts execution at the 7.68 MHz XTAL frequency.
- The microcontroller executes one machine cycle to reset all indirectly resetable registers.
- The µC programs the PLL for the USB module and the µC (**Chapter 9.1**).
- The USB module clock is switched on by programming DCR.UCLK (**Chapter 9.1**).
- Setting the USB reset bit DCR.SWR starts the software reset operation of the complete USB module.
- When the software reset is finished, DCR.SWR is automatically cleared by hardware and bit DCR.DINIT is set to indicate the start of the initialization sequence.
- The USB module is functionally initialized by the microcontroller by writing the configuration bytes for each endpoint. Thereafter, bit EPBSn.DONEn is set by software. After this action, bit DCR.DINIT is automatically reset by hardware and the software reset and initialization sequence are finished.
- Programming of required configuration registers (e.g. ISDN) takes place. This depends on the required operation mode.

This switch-on procedure after a hardware reset assures proper operation of the USB clock system.

9.3.2 Active to Idle

This chapter provides a description for the operational sequence, for a detailed description about the idle mode itself please refer to **Chapter 9.2.1**.

- The idle mode is entered by programming the bits IDLE and IDLS in the PCON register.
- In idle mode, different sub-modules (e.g. USB module, S-Interface, HDLC controllers) can be fully functional or can be switched off depending on system requirements.
- Special care is necessary to switch off the USB module. The following steps must be processed before entering the idle mode:
  - µC switches the USB module clock off by resetting DCR.UCLK.
  - PLL output is synchronously switched to low frequency input (PLCONB.SWCK=0), i.e. the PLL is actually bypassed and the µC is operating at 7.68 MHz
  - PLL is switched off (PLCONB.PCLK=0)

9.3.3 Idle to Active

There are two ways to terminate the idle mode

- hardware reset
- interrupt

Hardware Reset

The hardware reset must be kept active for 4ms till the oscillator stabilizes. The **"Reset to Active"** sequence applies now (**Chapter 9.3.1**).

Interrupt

The idle mode is left by receiving any enabled interrupt. This interrupt will be serviced and normally the instruction to be executed following the RETI instruction will be the one following the instruction that set the idle bit.

- After leaving the idle mode through an interrupt (e.g. from the USB module), a well defined procedure must be executed again to switch on the USB module and PLL.
- First the µC programs the PLL (see **Chapter 9.1**).
- The USB module clock is switched on by programming DCR.UCLK (**Chapter 9.1**).
- The switch off/on procedure assures proper operation of the USB clock system. As opposed to the hardware reset, after an interrupt the USB module does not need to be reconfigured but the previous settings are retained.
- Programming of required configuration registers (e.g. ISDN) takes place. This depends on the required operation mode.

9.3.4 Active to Suspend

This chapter provides a description of the operational sequence, for a detailed description about the suspend mode itself please refer to **Chapter 9.2.2**.

- In suspend mode, the oscillator is stopped. Therefore all functions of the µC are stopped and only the contents of the onchip IRAM, XRAM and SFRs are maintained.
- Special care must be taken for configuration of the suspend mode. For a detailed description please refer to **Chapter 9.2.2**.

9.3.5 Suspend to Active

There are two ways to terminate the suspend mode

- hardware reset
- wakeup event

Hardware Reset

The hardware reset must be kept active for 4ms till the oscillator stabilizes. The **"Reset to Active"** sequence applies now (**Chapter 9.3.1**).

Wakeup Event

If the wakeup capability from suspend initiated by certain events is required, this function must be enabled (**Chapter 9.3.6**) prior to setting the power down configuration bit. In normal operation mode and in idle mode these events will set their corresponding interrrupt status bit and can issue a maskable interrupt. However, in suspend mode these wakeup sources are directly routed to the NMI (non-maskable interrupt) and the interrupt status bits are not affected. In case of a wakeup event, the NMI input to the C800 core is activated by hardware. The core then executes an interrupt service routine at $7B_H$, and continues normal program execution.

The following two cases to exit suspend must are differentiated:

- Exit via pin $\overline{\text{INT0}}$, $\overline{\text{EAW}}$, C/I-Code Change or S-Bus Level Detect
  If the wake-up capability from one of these sources has been selected and if the standard request Set_Feature Remote_WakeUp was given before entering suspend mode, any activity from these sources can initiate termination of the suspend state.
  When the onchip oscillator clock is detected for stable nominal frequency (after approx. 4 ms), a synchronous multiplexer switches this 7.68 MHz clock to the microcontroller. The interrupt address of the first instruction to be executed after wake-up is 007BH. This interrupt will be serviced and normally the instruction to be executed following the RETI instruction will be the one following the instruction that set the power down bit.
  The PLL is switched on and the rest of the steps follow the **"Reset to Active"** programming sequence.

- Exit via the USB (resume from host)
  Any activity on the USB bus (in this case host initiated) will terminate the suspend state. The wake-up procedure now starts with oscillator stabilization. The wake-up trigger signal from the USB module can only be generated if the USB receiver circuitry was enabled in suspend mode (**Chapter 9.2.2**).
  The rest of the steps follow the **"Reset to Active"** programming sequence.

9.3.6 Interrupt Wakeup Control

Before the µC sets the ISDN and USB part and finally itself into suspend mode it can determine which external event will be allowed to terminate the suspend state and initiate a resume.

The configuration is done in the Wakeup Control register (WCON) with the following bits:

- EWPD (External Wakup from Power Down Enable)
  selects if external wakeup from power down mode is generally enabled, the other bits in register WCON enable the wakeup source individually. If EWPD is configured to "wakeup disabled" all other configuration bits are don't care.
- WPUS (Wakeup via USB Bus Enable)
  Wakeup from USB device core wakes up the SIUC-BA, i.e. a resume signalling on the bus initated from the USB host is detected. The USB suspend mode end interrupt is indicated in DIRR.SEI.
- WPIO (Wakeup via $\overline{\text{INT0}}$ Enable)
  An external device activating pin $\overline{\text{INT0}}$ wakes up the SIUC-BA. An external interrupt on $\overline{\text{INT0}}$ is indicated in TCON.IE0.
- WPTR (Wakeup via S transceiver Enable)
  Any signal level different from INFO0 (INFO0 = no signal) indicates an incoming call on the S interface, so the SIUC-BA is waked up to receive the call. A level detect interrupt is indicated via ISTA.TRAN.
- WPCI (Wakeup from C/I-Code Change or from $\overline{\text{EAW}}$ Enable)
  A code change in the downstream C/I channel or an external awake signal on pin $\overline{\text{EAW}}$ wakes up the SIUC-BA. The interrupt is indicated in ISTA.CIC or in ISTA.AUX, respectively.

10 Electrical Characteristics

10.1 Absolute Maximum Ratings

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Ambient temperature under bias	T_A	0	70	°C
Storage temperature	T_{STG}	– 55	150	°C
Input/output voltage on any pin with respect to ground	V_S	– 0.3	5.25	V
Maximum voltage on any pin with respect to ground	V_{max}		5.5	V

Note: Stresses above those listed here may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.
Voltage applied to any signal pin without the power supply connected can damage the device, i.e. it must be ensured that power supply is connected before any pin signal and the supply voltage must show a monotonic rise.
If not otherwise noted in this chapter 10,
V_{DD} is used synonymous for V_{DD}, V_{DDA}, V_{DDAP}, V_{DDU} and V_{DDR}, and
V_{SS} is used synonymous for V_{SS}, V_{SSA}, V_{SSAP}, V_{SSU} and V_{SSAR}.

10.2 DC Characteristics

V_{DD}/V_{SS} = 3.3V ± 0.2V; T_A = 0 to 70 °C

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
H-input level (except pin SR1/2)	V_{IH}	2.0		5.25	V	
L-input level (except pin SR1/2)	V_{IL}	– 0.3		0.8	V	
H-output level (except pin XTAL2, SX1/2)	V_{OH}	2.4			V	I_{OH} = - 400 µA
L-output level (except pin XTAL2, SX1/2)	V_{OL}			0.45	V	I_{OL} = 6 mA (DU, DD, C768) I_{OL} = 4.5 mA ($\overline{ACL}$, AUX6, AUX7) I_{OL} = 2 mA (all others)
Input leakage current Output leakage current (all pins except SX1/2, SR1/2, XTAL1/2, AUX7/6)	I_{LI} I_{LO}			±1 ±1	µA µA	0V< V_{IN}<V_{DD} 0V< V_{OUT}<V_{DD}
Input leakage current Output leakage current (AUX7/6)	I_{LI} I_{LO}	50 50		200 200	µA µA	0V< V_{IN}<V_{DD} 0V< V_{OUT}<V_{DD} (only if AUX7/6 is input or output/open-drain; not relevant if output/push-pull)

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
V_{DD}= 3.3V ± 0.2V; V_{SS}= 0V; T_A = 0 to 70 °C						
Power supply current-Power Down (suspend)	I_{PD}			400	µA	Inputs at V_{SS} / V_{DD} No output loads except SX1,2 (50Ω)
Power supply current-S operational (96 kHz Testsignal) B1=FF_H,B2=FF_H, D=1)	I_{OP1} I_{OP2}			70 65	mA mA	Inputs at V_{SS} / V_{DD} No output loads except SX1,2 (50Ω); USB controller operational;
Power supply current-Operational (96 kHz Testsignal)	I_{OP3}			60	mA	Inputs at V_{SS} / V_{DD} No output loads except SX1,2 (50Ω); USB core disabled (DCR.UCLK);
Absolute value of output pulse amplitude (VSX2 – VSX1)	Vx			1.17	V	R_L = ∞
Transmitter output current (SX1,2)	Ix			26	mA	R_L = 5.6 Ω
Transmitter output impedance (SX1,2)	Zx	10 0			kΩ Ω	- Inactive or during binary one; - during binary zero R_L = 50 Ω

10.3 Voltage Regulator

T_A = 0 to 70 °C

Parameter	Symbol	Limit Values		Unit	Conditions
		min.	max.		
Input Voltage	V_{IN}	4.10	5.30	V	
Output Voltage	V_{OUT}	3.10	3.50	V	
Quiescent current	I_Q		65	µA	
Output current - suspend	I_{OUT1}	435		µA	
The output current in operational mode depends on the external circuitry. Examples are given below:					
Output current - operational	I_{OUT2}	80 100		mA mA	V_{IN} = 4.10 V BSS129 BSS149

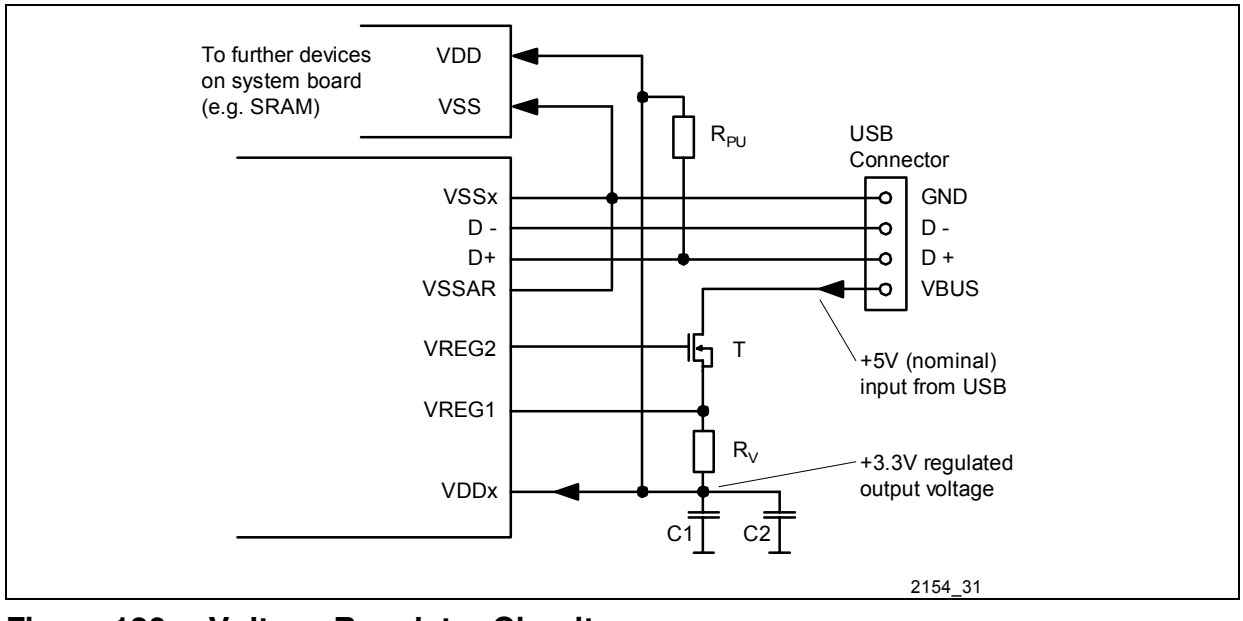


Figure 120 Voltage Regulator Circuit

T = BSS129 or BSS149(N-channel depletion)
R_V = 1 Ω
C1 = 33 µF
C2= 100 nF

R_{PU} = 1.5 kΩ ± 5% (Pull up resistor on D+ to indicate a full speed device; R_{PU} is not required for the voltage regulator)

10.4 Capacitances

T_A = 25 °C, V_{DD} = 3.3V ± 0.2V, V_{SS} = 0 V, f_c = 1 MHz, unmeasured pins grounded.

Parameter	Symbol	Limit Values		Unit	Remarks
		min.	max.		
Input Capacitance I/O Capacitance	C_{IN} $C_{I/O}$		7 7	pF pF	All pins except SX1,2 and XTAL1,2
Output Capacitance against V_{SS}	C_{OUT}		10	pF	pins SX1,2
Load Capacitance	C_L		40	pF	pins XTAL1,2

10.5 Oscillator Specification

Recommended Oscillator Circuits

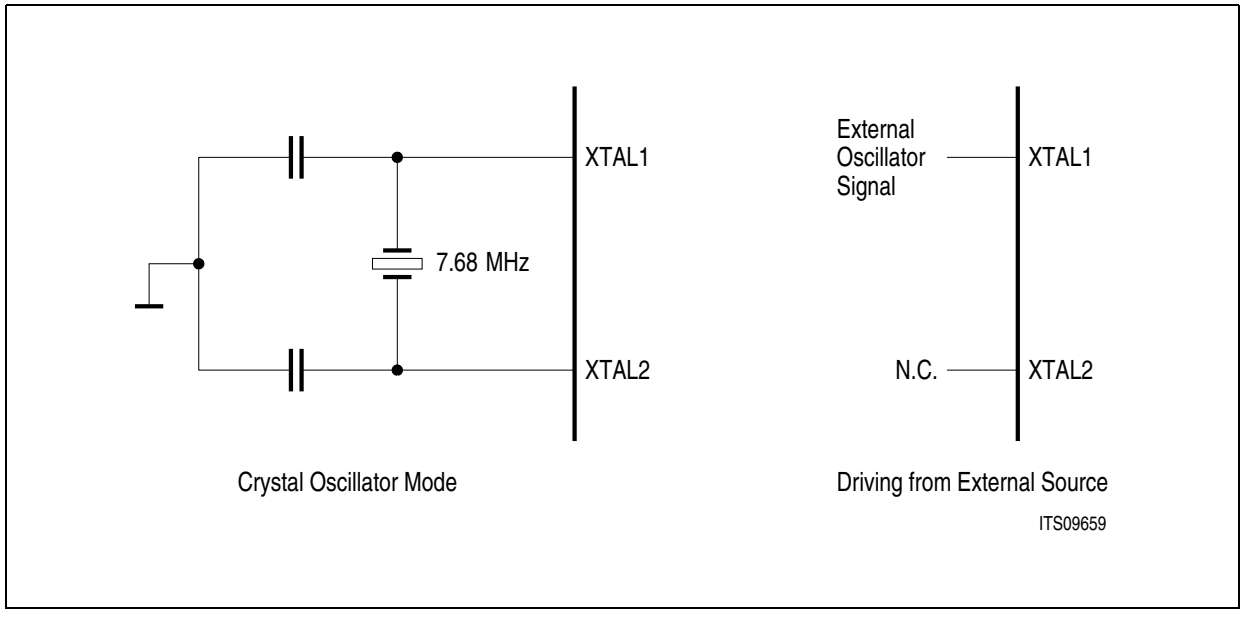


Figure 121 Oscillator Circuits

Crystal Specification

Parameter	Symbol	Limit Values	Unit
Frequency	f	7.680	MHz
Frequency calibration tolerance		max. 100	ppm
Load capacitance	C_L	max. 40	pF
Oscillator mode		fundamental	

Note: The load capacitance C_L depends on the recommendation of the crystal specification. Typical values for C_L are 22 ... 33 pF.

XTAL1 Clock Characteristics (external oscillator input)

Parameter	Limit Values	
	min.	max.
Duty cycle	1:2	2:1

10.6 Recommended Transformer Specification

Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
Transformer ratio			1:1			
Main inductance	L	25 20			mH mH	no DC current, 10 kHz 2.5 mA DC current, 10 kHz
Leakage inductance	L_L			8	µH	10 kHz
Capacitance between primary and secondary side	C			80	pF	1 kHz
Copper resistance	R	1.7	2.0	2.3	W	

10.7 AC Characteristics

T_A = 0 to 70 °C, V_{DD} = 3.3 V ± 5 %

Inputs are driven to 2.4 V for a logical "1" and to 0.45 V for a logical "0". Timing measurements are made at 2.0 V for a logical "1" and 0.8 V for a logical "0". The AC testing input/output waveforms are shown in **figure 122.**

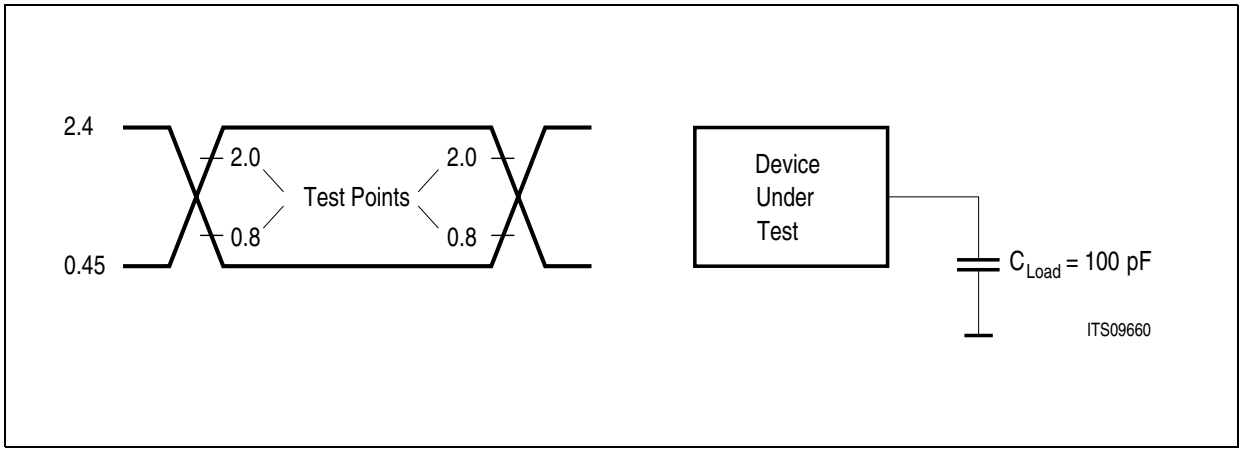


Figure 122 Input/Output Waveform for AC Tests

10.8 IOM-2 Interface Timing

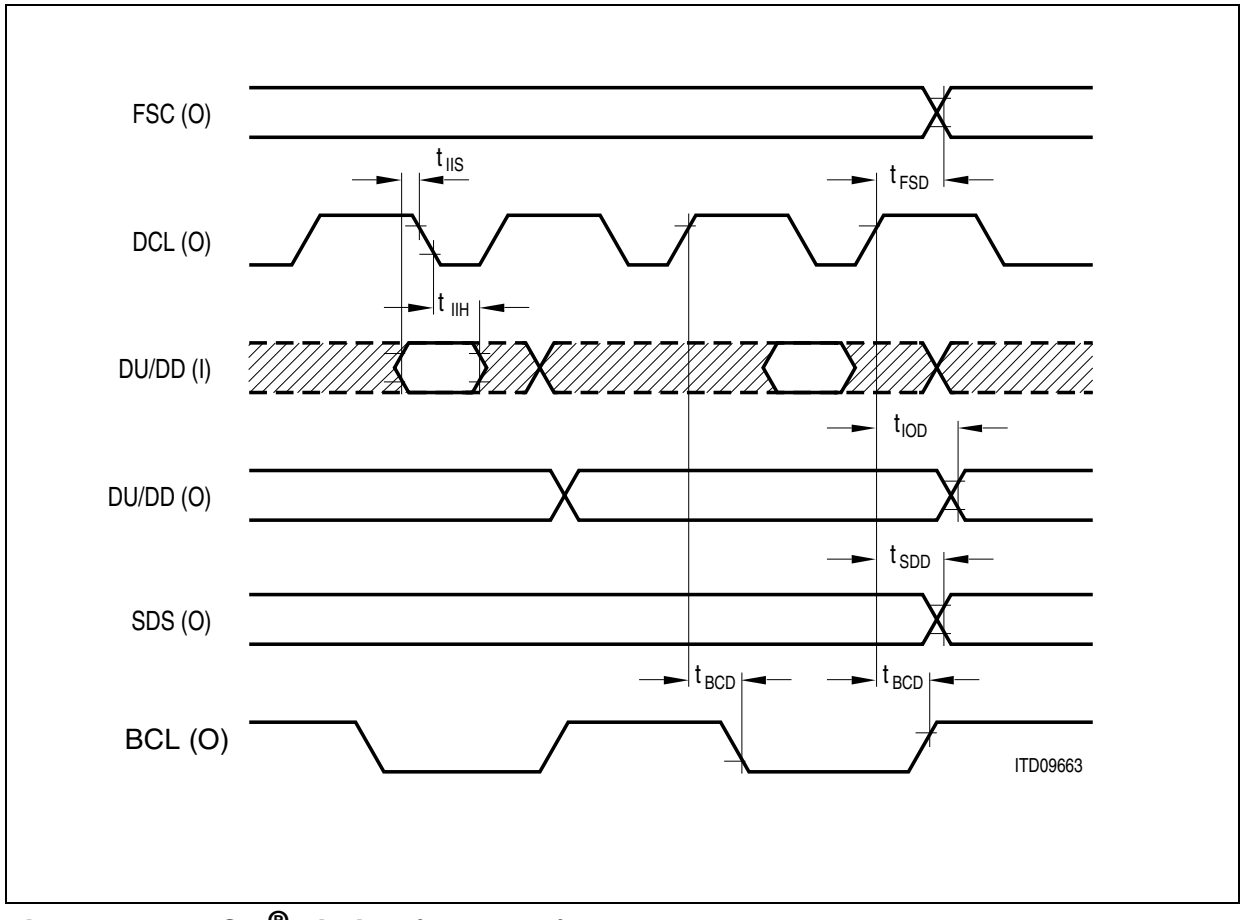


Figure 123 IOM® Timing (TE mode)

Parameter	Symbol	Limit Values		Unit
		min.	max.	
IOM output data delay	t_{IOD}		100	ns
IOM input data setup	t_{IIS}	20		ns
IOM input data hold	t_{IIH}	20		ns
FSC strobe delay	t_{FSD}	-130		ns
Strobe signal delay	t_{SDD}		120	ns
BCL delay	t_{BCD}		100	ns
Frame sync setup	t_{FSS}	50		ns
Frame sync hold	t_{FSH}	30		ns
Frame sync width	t_{FSW}	40		ns

DCL Clock Characteristics

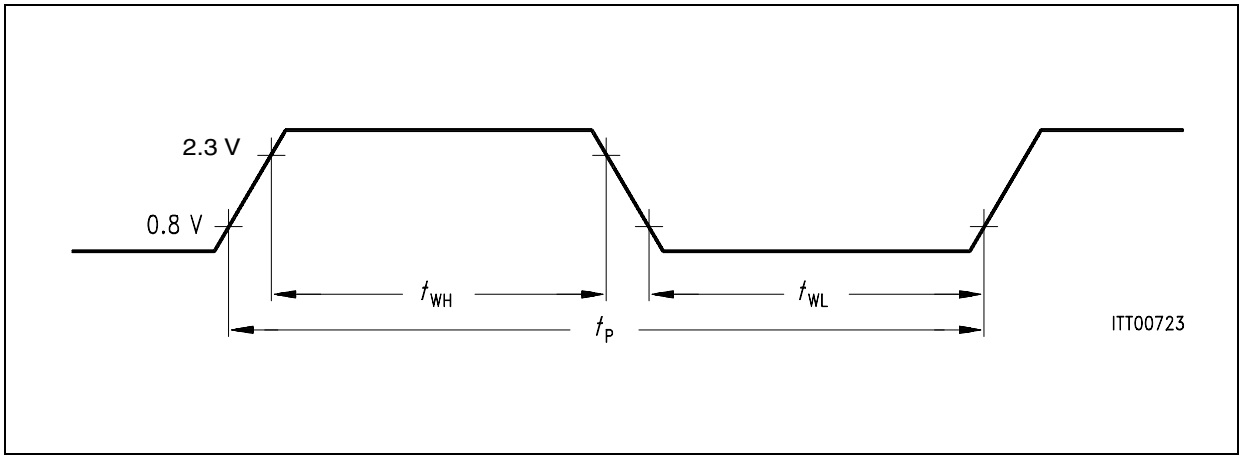


Figure 124 Definition of Clock Period and Width

Symbol	Limit Values			Unit	Test Condition
	min.	typ.	max.		
t_{PO}	585	651	717	ns	osc ± 100 ppm
t_{WHO}	260	325	391	ns	osc ± 100 ppm
t_{WLO}	260	325	391	ns	osc ± 100 ppm

10.9 Auxiliary Interface Timing

The pins from the auxiliary interface can be used as standard I/O pins. Their timing conditions either as input or as output is shown in **figure 125**. The read and write signals shown below indicate the corresponding access to the SIUC-BA register, they are not control signals on the auxilliary interface.

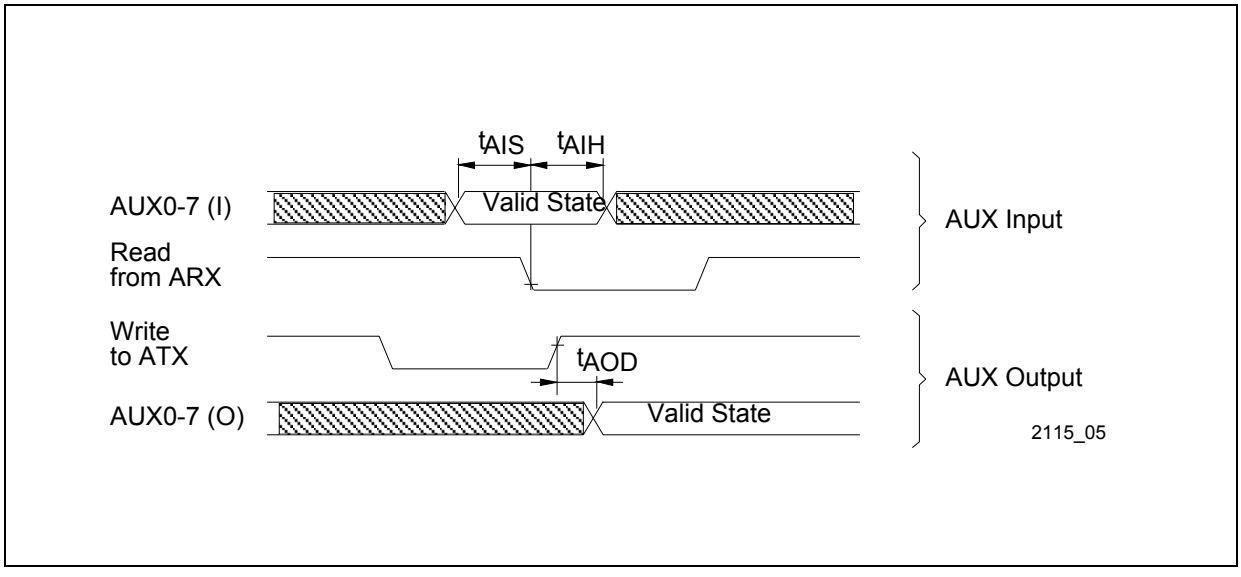


Figure 125 AUX Interface I/O Timing

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Auxiliary input data setup	t_{AIS}	30		ns
Auxiliary input data hold	t_{AIH}	30		ns
Auxiliary output data delay	t_{AOD}		200	ns

10.10 SPI Interface Timing

Some pins from the auxiliary interface can be used to realize an SPI interface in order to connect a serial EEPROM.

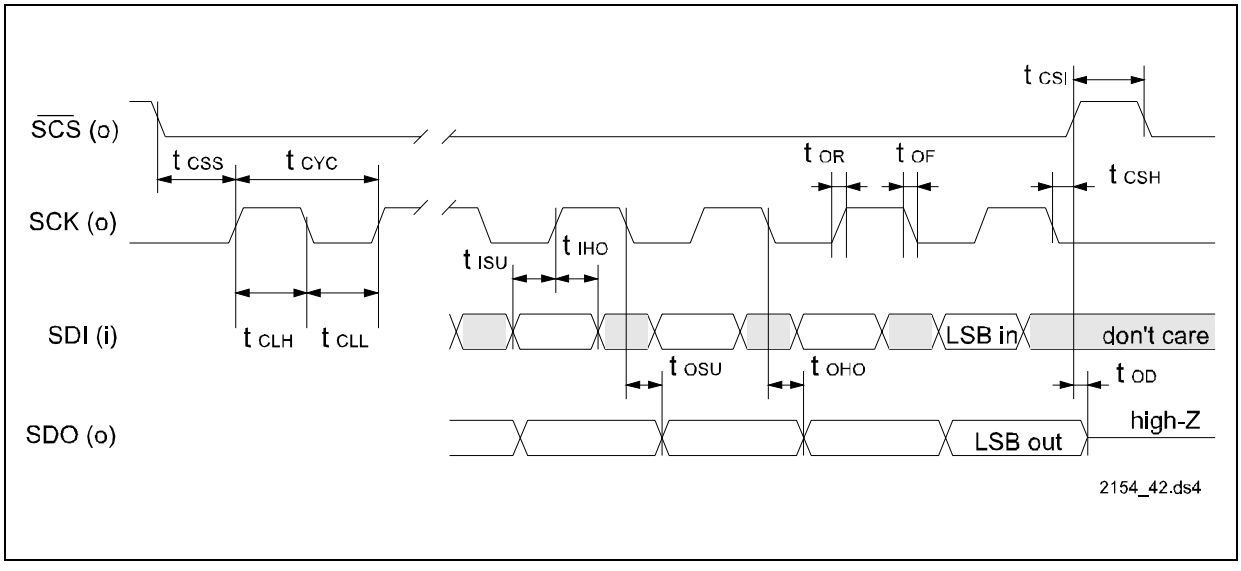


Figure 126 AUX Interface I/O Timing

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Chip Select Setup Time	t_{CSS}	500		ns
Chip Select Hold Time	t_{CSH}	500		ns
Chip Select Inactive	t_{CSI}	500		ns
Clock Cycle Time	t_{CYC}	1000		ns
Clock HIGH Time	t_{CLH}	410		ns
Clock LOW Time	t_{CLL}	410		ns
Clock Output Rise Time	t_{OR}		2	ns
Clock Output Fall Time	t_{OF}		2	ns
Input Data Setup Time	t_{ISU}	100		ns
Input Data Hold Time	t_{IHO}	100		ns
Output Data Setup Time	t_{OSU}		500	ns
Output Data Hold Time	t_{OHO}	0	500	ns
Output Disable Time	t_{OD}		500	ns
Write Cycle Time	t_{WC}		10	ns

10.11 USB Transceiver Characteristics

The electrical characteristics of the USB device core in SIUC-BA are compliant to the USB V1.1 specification.

V_{DDU} = 3.3 V ± 0.2 V, V_{SSU} = 0 V, T_A = 0 to 70 °C

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Output impedance (high state)	R_{DH}	28	43	Ω	1)
Output impedance (low state)	R_{DL}	28	51	Ω	
Input leakage current	I_I		5	µA	$V_{IN}=V_{SS}$ or V_{CC}
Tristate output off-state current	I_{OZ}		10	µA	$V_{OUT}=V_{SS}$ or V_{CC} 1)
Crossover point	V_{CR}	1.3	2.0	V	2)

Note: 1) This value includes an external resistor of 30Ω ± 1 % (for testing details see diagram “Load for D+/D-”).
2) The crossover point in in the range of 1.3V to 2.0V with a 50 pF capacitance.

Parameter	Symbol	Limit Values		Unit
		min.	max.	
Rise Time	t_{FR}	4	20	ns
Fall Time	t_{FF}	4	20	ns

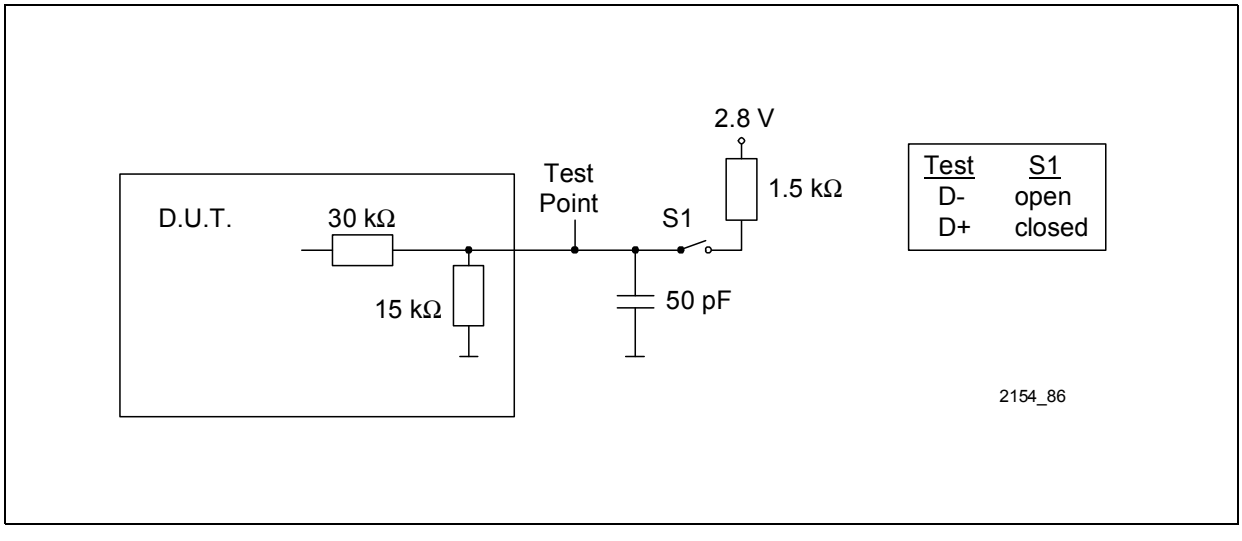


Figure 127 Load for D+/D-

Receiver Sensitivity

The input sensitivity is at least 200 mV when both differential data inputs are in the differential common mode range of 0.8V to 2.5V.

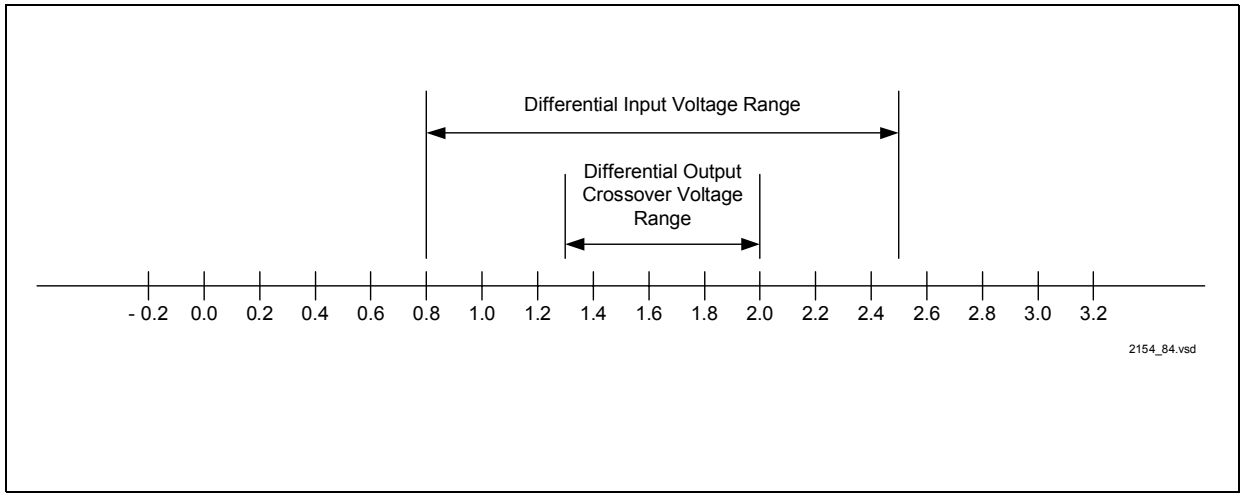


Figure 128 Differential Input Sensitivity Range

10.12 Reset

Parameter	Symbol	Limit Values	Unit	Test Conditions
		min.		
Length of active high state	t_{RES}	4	ms	Power On/Power Down to Power Up (Standby)
		2 x DCL clock cycles		During Power Up (Standby)

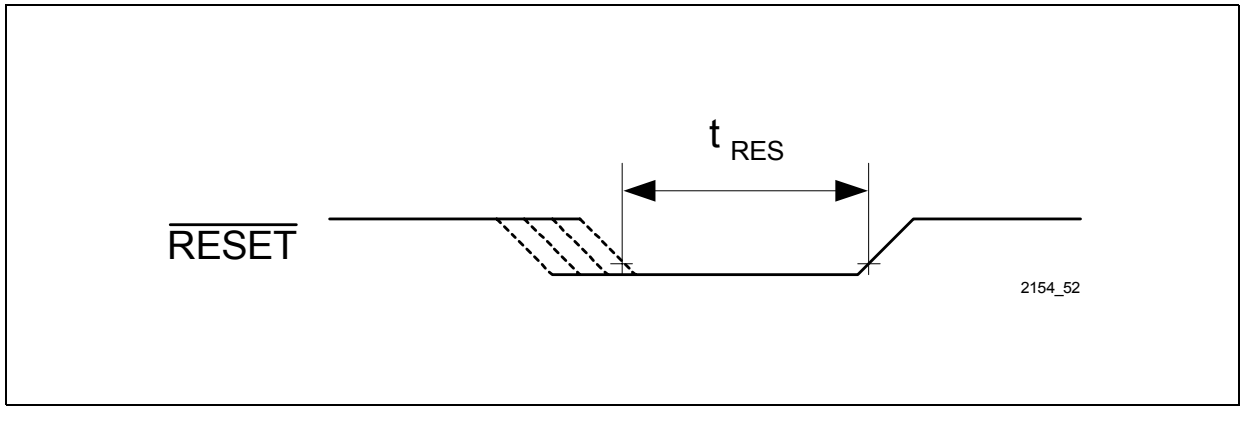
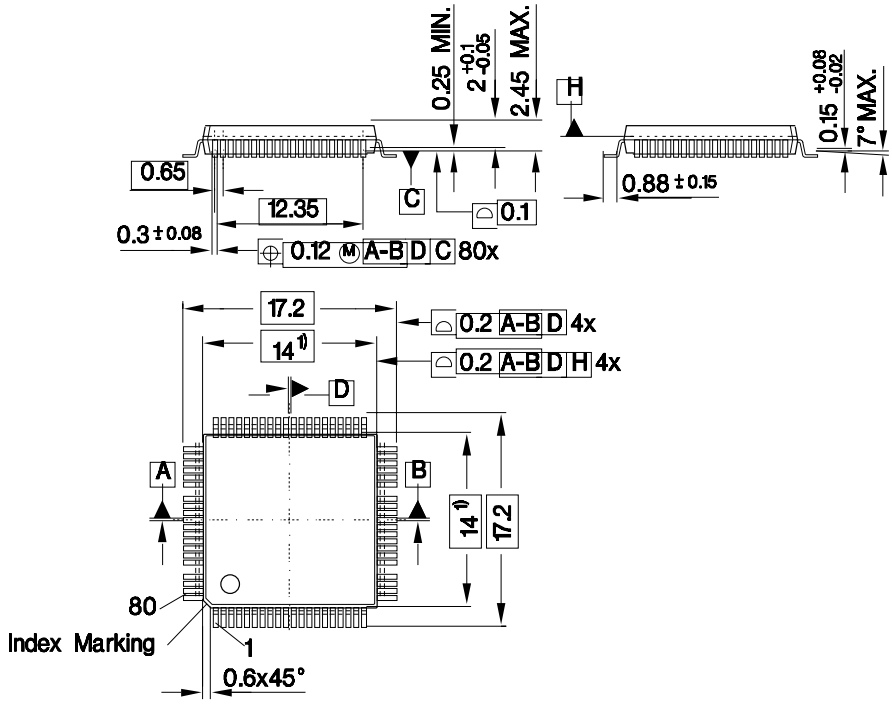


Figure 129 Reset Signal

11 Package Outlines

P-MQFP-80-1
(Plastic Metric Quad Flat Package)



1) Does not include plastic or metal protrusion of 0.25 max. per side

GPM05249

Sorts of Packing

Package outlines for tubes, trays etc. are contained in our Data Book “Package Information”.
SMD = Surface Mounted Device

Dimensions in mm

A

B

C

D

E

F

G

H

I

N

O

P

R

S

T

U

V

W

X

Infineon goes for Business Excellence

“Business excellence means intelligent approaches and clearly defined processes, which are both constantly under review and ultimately lead to good operating results.
Better operating results and business excellence mean less idleness and wastefulness for all of us, more professional success, more accurate information, a better overview and, thereby, less frustration and more satisfaction.”

Dr. Ulrich Schumacher

http://www.infineon.com

Published by Infineon Technologies AG