Part 3.

TMP68301AKF-8
TMP68301AKFR-8

1. Introduction

The TMP68301AK uses the 68HC000, the CMOS version of the 68000, as its core processor. The TMP68301AK includes a serial interface, parallel interface, timer, and interrupt controller. The TMP68301AK operates at a low 3.3voltage (3.3 V±10 %), making it suitable for fields where low power consumption is required. It incorporates peripheral circuits, such as the address decoder.

68000 development environments and software resources can be used directly with the TMP68301AK.

- Core processor 68HC000
- 17 32-bit registers
- 16M-byte direct addressing
- 56 powerful basic instructions
- 14 addressing modes

- 3-channel asynchronous serial interface
- 16-bit parallel I/O interface
- 3-channel, 16-bit timer/counter
- 10-channel interrupt controller (3 external channels,7 internal channels)
  (can be extended by software to 10 external channels)
- 2-channel chip-select signal output ($\overline{CS0}$, $\overline{CS1}$)
- Automatic wait insertion
- Bus monitor function
- Low power consumption (CMOS)
- 2 types of packages: 100-pin QFP and 100-pin RFP
- Low-voltage operation (Vcc = 3.3 V±10 %)

The TMP68301AK has two operating modes: normal operating mode, and emulation mode for using an in-circuit emulator (ICE), which is a 68000 development tool. In emulation mode, the 68HC000 core built into the TMP68301AK is disconnected from the bus, and the internal peripheral circuits are controlled using the address, data, and control signals from the development tool.

The 68HC000 core built into the TMP68301AK is the same as the standard TMP68HC000, except that the 8-bit peripheral device control signals E, $\overline{VPA}$, and $\overline{VMA}$ are not used. For information on 68HC000 operation and instructions, refer to the TLCS-68000 Data Book .

Unlike the 68301A, the 68301AK parallel interface does not support a Centronics interface (can be used as an I/O port).

Figure 1.1 shows the TMP68301AK block diagram.

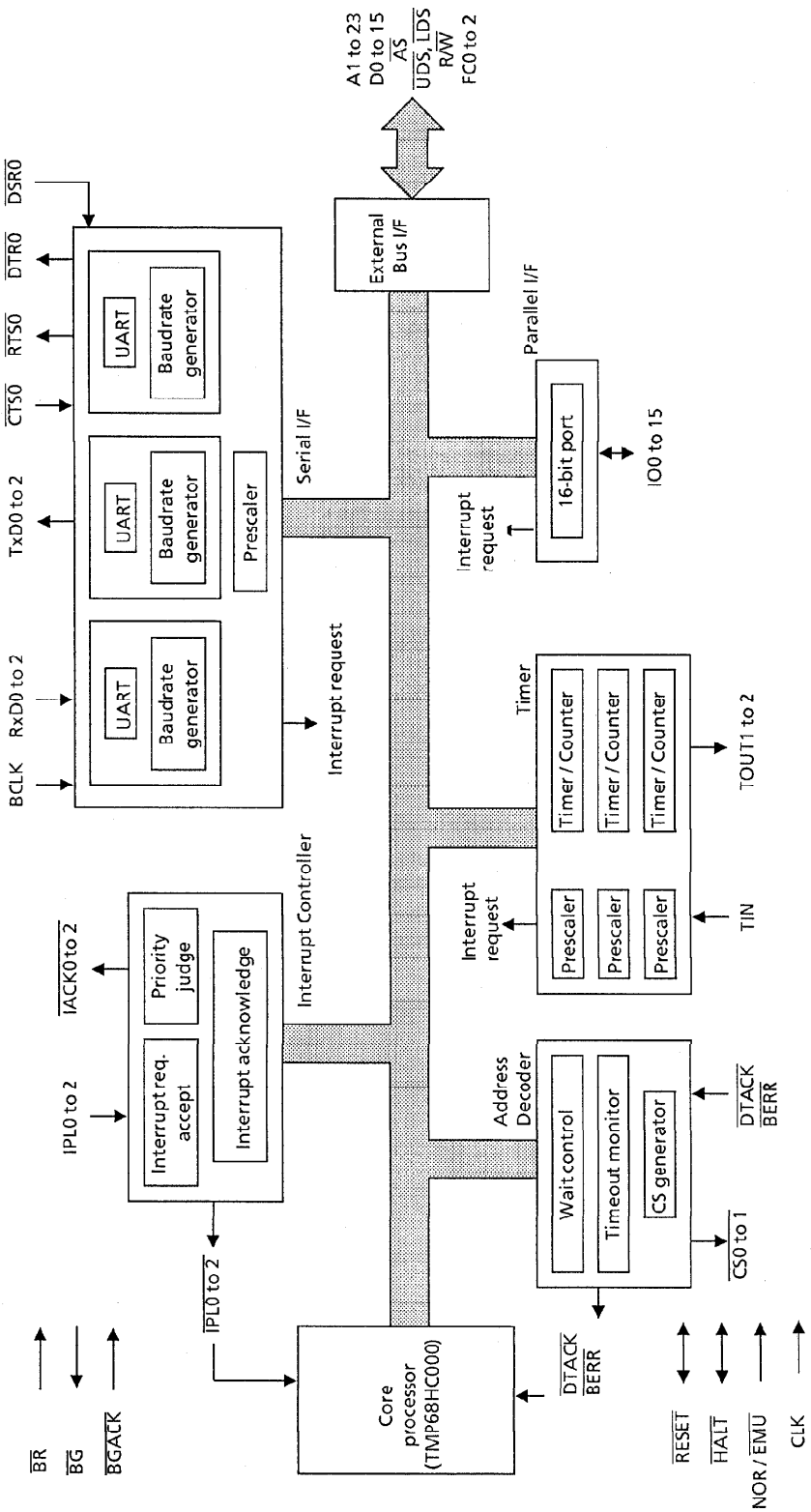


Figure 1.1 TMP68301AK Block Diagram

2. SIGNAL AND BUS OPERATION DESCRIPTION

This section briefly describes input and output signals. The terms "assert" and "negate" appear frequently. These terms are used to avoid ambiguity where terms such as "active high" and "active low" might cause confusion. "Assert" is used to show that signals are active or true, irrespective of whether the signal is electrically high or low. "Negate" is used to show that signals are inactive or false.

2.1 Pin Assignment Diagram

Figures 2.1 shows the pin assignments.

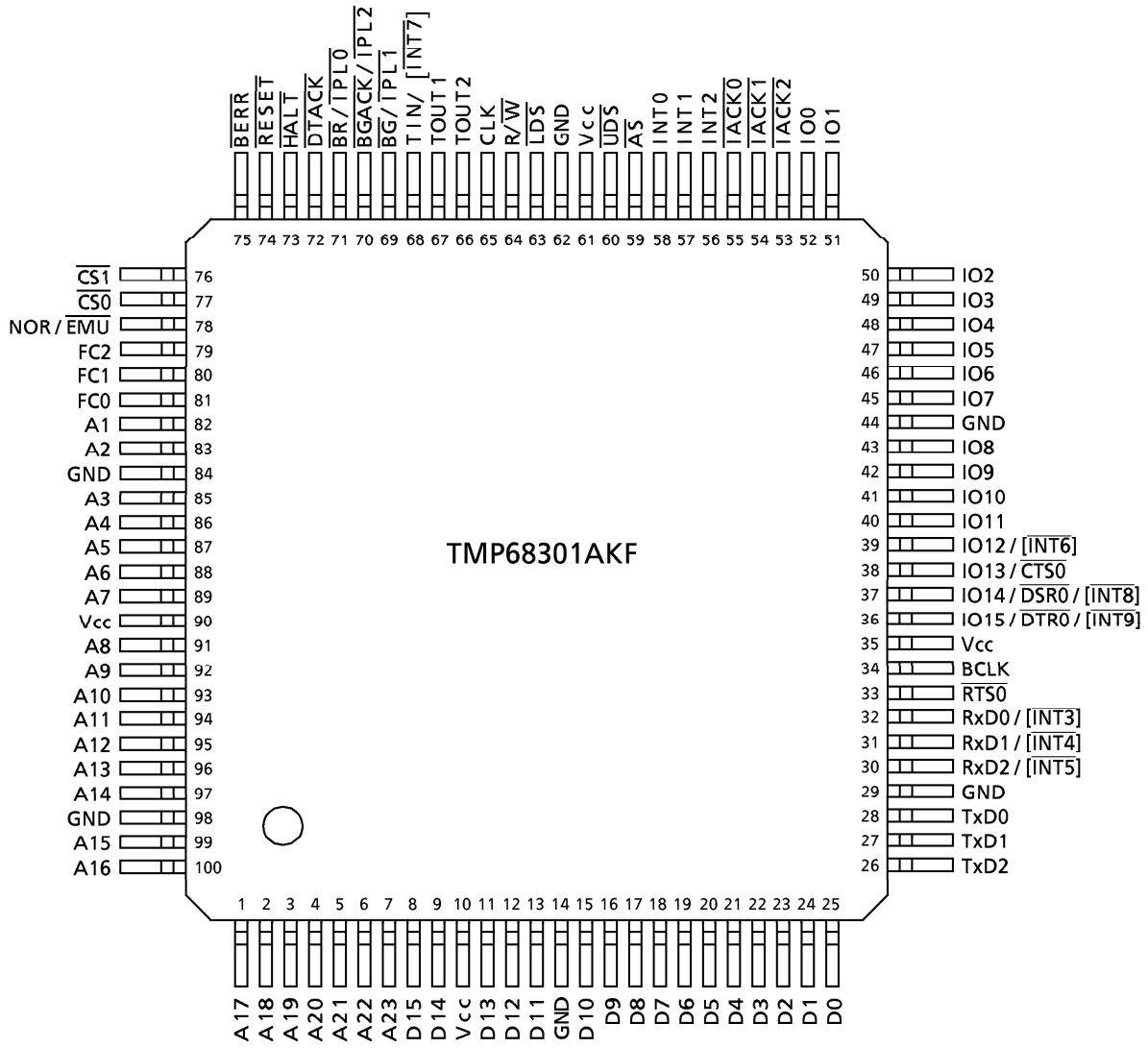


Figure 2.1 Pin Assignments (top view) (1/2)

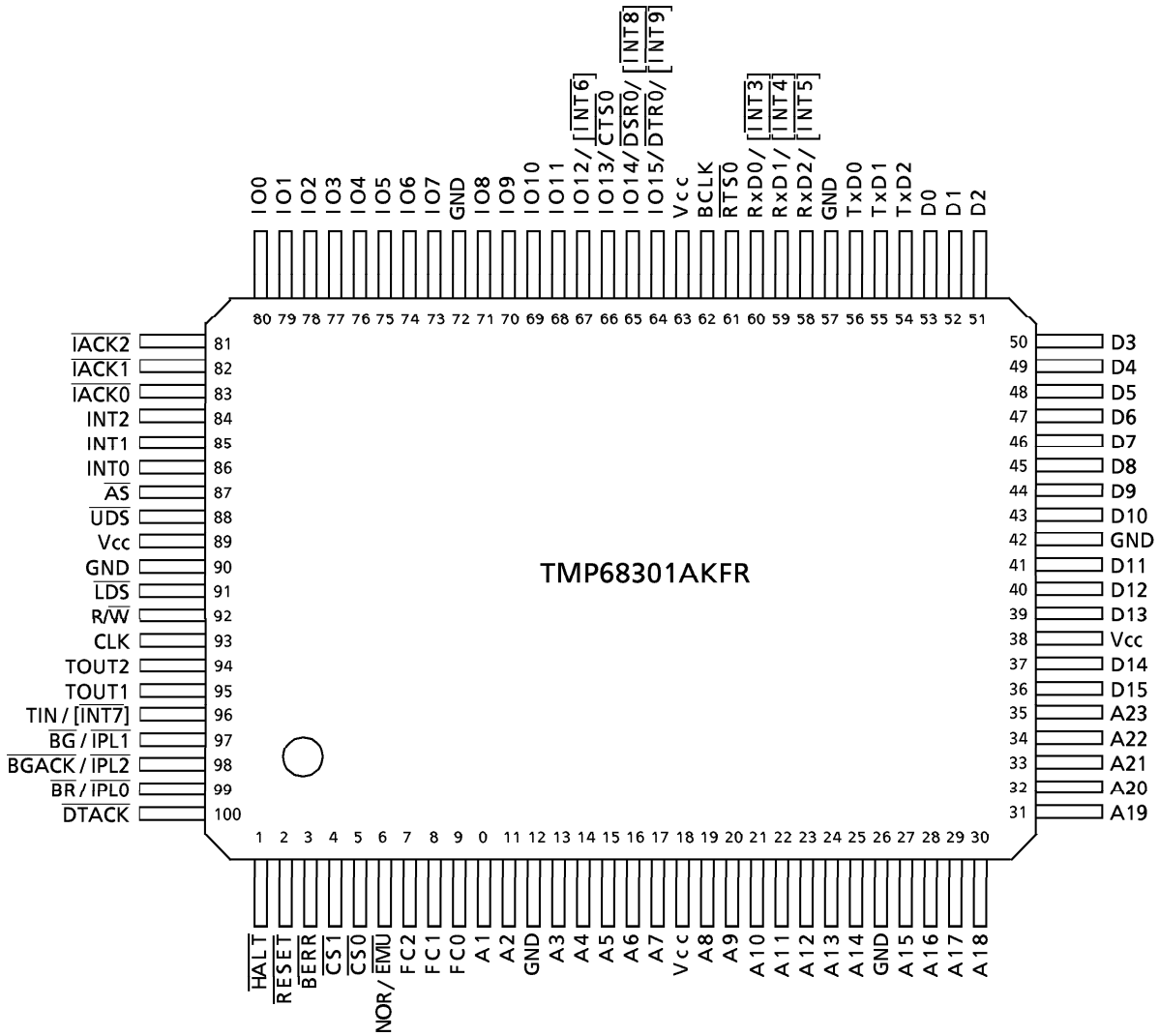


Figure 2.1 Pin Assignments (top view) (2/2)

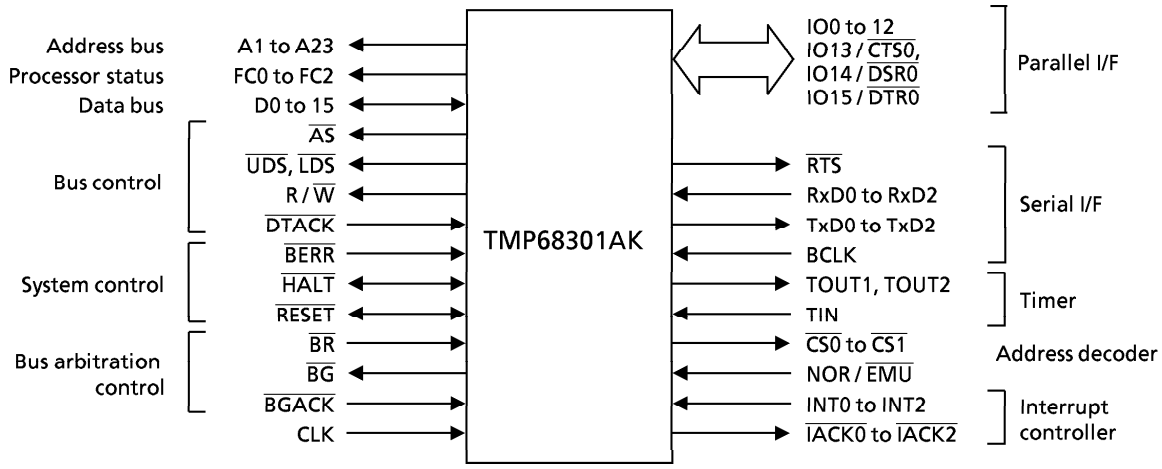


Figure 2.2 Normal Mode Input / Output Signals When Configured as Bus Master

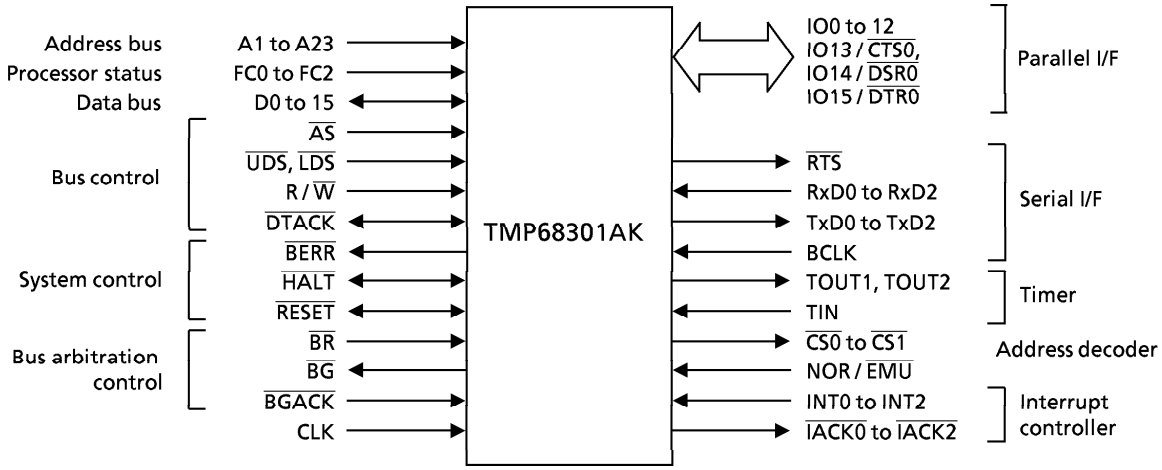


Figure 2.3 Normal Mode Input / Output Signals When Bus Mastership Released

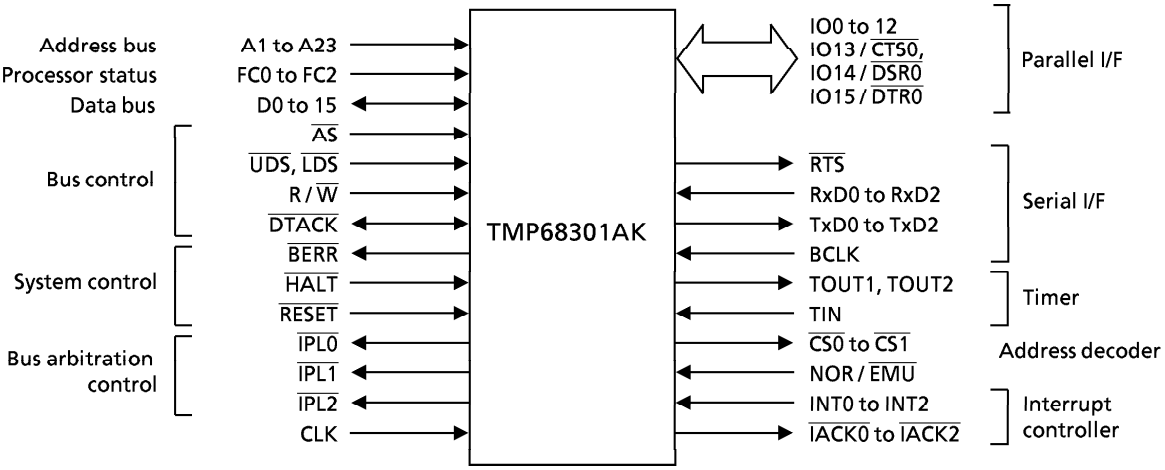


Figure 2.4 Emulation Mode Input / Output Signals

2.2 Pin Names and Functions

The following describes pin states and functions in normal and emulation modes.

NOR : Normal mode EMU : Emulation mode
O.D : Open drain output O : Output I : Input I/O : Input/output

Signal	Pin Status		Function
	NOR	EMU	
A1 to A23	O	I	23-bit address bus. Can directly access 16M bytes of memory.
FC0 to FC2	O	I	Show the status of the processor and the current cycle type. For details, see Table 2.2.
D0 to D15	I/O	I/O	16-bit general-purpose data bus.
$\overline{AS}$	O	I	Indicates that there is a valid address on the address bus.
R/$\overline{W}$	O	I	Indicates whether the data transfer is read (high) or write (low).
$\overline{UDS}$ / $\overline{LDS}$	O	I	Control the data on the data bus. For details, see Table 2.1.
$\overline{DTACK}$	I	O.D	Indicates the end of a data transfer.
$\overline{BR}$ ($\overline{IPL0}$)	I	O	Wire-ORed with all other devices that could become bus masters. Indicates that another device is requesting bus mastership. Becomes the $\overline{IPL0}$ output in emulation mode. The $\overline{IPL}$ signal codes the priority level of devices requesting interrupts.
$\overline{BG}$ ($\overline{IPL1}$)	O	O	Indicates to devices that could become bus masters that the processor will release control of the bus at the end of the current bus cycle. Becomes the $\overline{IPL1}$ output in emulation mode.
$\overline{BGACK}$ ($\overline{IPL2}$)	I	O	Indicates that another device has become the bus master. Becomes the $\overline{IPL2}$ output in emulation mode.
$\overline{BERR}$	I	O.D	Reports to the processor that there is a problem in the current cycle.
$\overline{RESET}$	O.D	I	In combination with $\overline{HALT}$, resets the processor. If the RESET instruction is executed, functions as a reset signal for external devices.
$\overline{HALT}$	O.D	I	When input, halts the processor when the current bus cycle ends. Also acts as an output signal when a double bus fault condition occurs.
CLK	I	I	Clock input pin.
INT0, INT1, INT2	I	I	External interrupt request input.
$\overline{IACK0}$, $\overline{IACK1}$ $\overline{IACK2}$	O	O	Indicate the interrupt acknowledge cycles corresponding to INT0, INT1, and INT2.
NOR / $\overline{EMU}$	I	I	Normal mode / Emulation mode switch signal.

NOR : Normal mode EMU : Emulation mode
O : Output I : Input I/O : Input/output

Signal	Pin Status		Function
	NOR	EMU	
RxD0 / [$\overline{INT3}$] RxD1 / [$\overline{INT4}$] RxD2 / [$\overline{INT5}$]	I	I	Serial interface data inputs. Also act as interrupt inputs according to the expansion interrupt register.
TxD0, TxD1, TxD2	O	O	Serial interface data output.
BCLK	I	I	The standard clock for generating the serial interface baudrates.
$\overline{RTS0}$	O	O	$\overline{RTS}$ signal for serial interface channel 0.
TIN [$\overline{INT7}$]	I	I	Input signal to each timer channel. Also acts as an interrupt input according to the expansion interrupt register.
TOUT1, TOUT2	O	O	Output signals from Timer channel 1 and 2.
IO0 to IO11	I/O	I/O	General-purpose I/O ports.
IO12 [$\overline{INT6}$]	I/O	I/O	General-purpose I/O port. Also acts as an interrupt input according to the expansion interrupt register.
IO13 / $\overline{CTS0}$	I/O	I/O	General-purpose I/O port or $\overline{CTS}$ signal for serial interface channel 0.
IO14 / $\overline{DSR0}$ [$\overline{INT8}$]	I/O	I/O	General-purpose I/O port or $\overline{DSR}$ signal for serial interface channel 0. Also acts as an interrupt input according to the expansion interrupt register.
IO15 / $\overline{DTR0}$ [$\overline{INT9}$]	I/O	I/O	General-purpose I/O port or $\overline{DTR}$ signal for serial interface channel 0. Also acts as an interrupt input according to the expansion interrupt register.
$\overline{CS0}$, $\overline{CS1}$	O	O	Chip select output from the address decoder.
Vcc	–	–	Power input (+ 3.3 V).
GND	–	–	Ground (0 V).

Table 2.1 Data Bus Control By Data Strobe

$\overline{UDS}$	$\overline{LDS}$	R/$\overline{W}$	D8 to D15	D0 to D7
H	H	–	Data invalid	Data invalid
L	L	H	Valid data bits 8 to 15	Valid data bits 0 to 7
H	L	H	Data invalid	Valid data bits 0 to 7
L	H	H	Valid data bits 8 to 15	Data invalid
L	L	L	Valid data bits 8 to 15	Valid data bits 0 to 7
H	L	L	Valid data bits 0 to 7*	Valid data bits 0 to 7
L	H	L	Valid data bits 8 to 15	Valid data bits 8 to 15*

L : Low H : High

* : This condition is a result of the current implementation and may not be available in the future.

2.2 Function Code Output

FC2	FC1	FC0	Cycle Types
L	L	L	*
L	L	H	User data
L	H	L	User program
L	H	H	*
H	L	L	*
H	L	H	Supervisor data
H	H	L	Supervisor program
H	H	H	Interrupt acknowledge

(Note)

L : Low H : High

* : Undefined, Reserved use

Note : If FC0, FC1, and FC2 are pulled up, when the bus is released, the state is the same as that at an interrupt acknowledge cycle, and the interrupt controller malfunctions. To prevent such malfunctions, pull down one of FC0, FC1, and FC2.

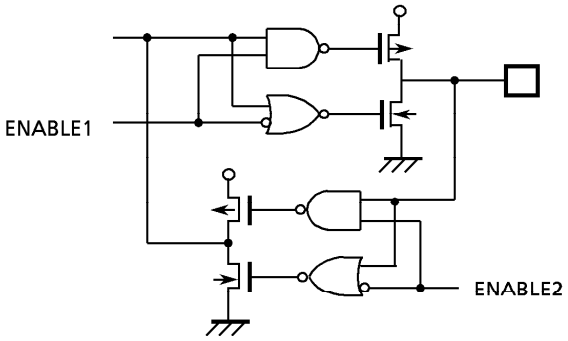
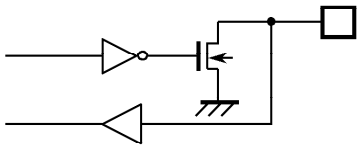
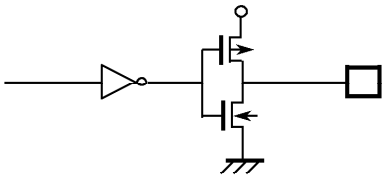
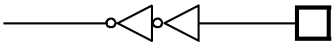
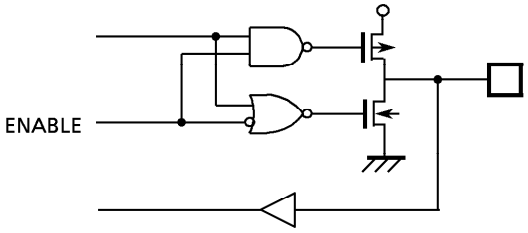
2.3 Signal Summary

Table 2.3 Signal Summary

Signal Name	Mnemonic	Input / Output	Active State	Tri-state output	On $\overline{HALT}$	On $\overline{BGACK}$	On $\overline{RESET}$ and $\overline{HALT}$
Address bus	A1 to A23	Output (Input)	High	Yes	Hi-Z	Hi-Z	Hi-Z
Data bus	D0 to D15	Input / Output (Output / Input)	High	Yes	Hi-Z	Hi-Z	Hi-Z
Address strobe	$\overline{AS}$	Output (Input)	Low	Yes	High	Hi-Z	Hi-Z
Read / Write	R/$\overline{W}$	Output (Input)	High (Read) Low (Write)	Yes	High	Hi-Z	Hi-Z
Upper and lower data strobe	$\overline{UDS}$, $\overline{LDS}$	Output (Input)	Low	Yes	High	Hi-Z	Hi-Z
Data transfer acknowledge	$\overline{DTACK}$	Input (Output)	Low	Open drain	–	–	–
Bus request	$\overline{BR}$ (/ $\overline{IPL0}$)	Input (Output)	Low	No	–	–	–
Bus grant	$\overline{BG}$ (/ $\overline{IPL1}$)	Output	Low	No	High	Low	High
Bus grant acknowledge	$\overline{BGACK}$ (/ $\overline{IPL2}$)	Input (Output)	Low	No	–	–	–
Bus error	$\overline{BERR}$	Input (Output)	Low	Open drain	–	–	–
Reset	$\overline{RESET}$	Input / Output	Low	Open drain	Open drain	Open drain	Low
Halt	$\overline{HALT}$	Input / Output	Low	Open drain	Low	Open drain	Low
Function code	FC0, FC1, FC2	Output (Input)	High	Yes	–	Hi-Z	Hi-Z
Clock	CLK	Input	High	–	–	–	–
I/O port	IO0 to 15	Input/Output	High	–	–	–	–
Timer output	TOUT1, TOUT2	Output	High	No	–	–	Low
Timer input	TIN	Input	Low	–	–	–	–
Request to send	$\overline{RTS0}$	Output	Low	No	–	–	High
Receive data	RxD0 to 2	Input	High	–	–	–	–
Transfer data	TxD0 to 2	Output	High	No	–	–	High
Baudrate clock	BCLK		High	–	–	–	–
Mode switch	NOR / $\overline{EMU}$			–	–	–	–
Interrupt request	INT0 to 2		High	–	–	–	–
Interrupt acknowledge	$\overline{IACK0}$ to $\overline{2}$	Output	Low	No	–	–	High
Chip select	$\overline{CS0}$, $\overline{CS1}$	Output	Low	No	–	–	High
Power input	V_{CC}		–	–	–	–	–
Ground	GND		–	–	–	–	–

The parentheses in the Input / Output column indicate emulation mode.
Notes : – : Optional

2.4 Pin Input / Output Circuits

Pin	Input / Output Circuit	Remarks
A1 to A23, D0 to D15 FC0 to FC2, $\overline{AS}$, $\overline{UDS}$, $\overline{LDS}$ R/$\overline{W}$		Tri-state output
$\overline{RESET}$, $\overline{HALT}$, $\overline{DTACK}$, $\overline{BERR}$		Sink open drain output
$\overline{BG}$ TOUT1, TOUT2 $\overline{RTS0}$ TxD0, TxD1, TxD2 $\overline{IACK0}$, $\overline{IACK1}$, $\overline{IACK2}$ $\overline{CS0}$, $\overline{CS1}$		Push / pull output
CLK TIN RxD0, RxD1, RxD2 BCLK INT0, INT1, INT2 NOR / $\overline{EMU}$		
$\overline{BR}$, $\overline{BGACK}$ IO0 to IO7 IO8 to IO15		

3. Address Decoder

Refer to 3. Address Decoder in the TMP68301A manual.

4. Interrupter Controller

4.1 Overview

The interrupt controller provides 10 interrupt channels. Seven of the channels are for internal peripheral circuits, while the other three are for external interrupts from interrupt request input pins INT0, 1, and 2. Interrupt request levels input to the core processor (the pattern of "0"s and "1"s input to the core processor IPL0, 1, and 2) can be set for each channel. Priorities can be set independently. Interrupt request input mode (input level, rising / falling edge) for external interrupts can also be set independently. In addition, the external interrupt vector number can be selected as either an internal vector number in the interrupt controller, or an externally input vector number.

If an external vector is input, the IACK output (IACK0, 1, or 2) corresponding to an external interrupt channel is asserted in accordance with the interrupt acknowledge cycle.

The auto-vectored interrupt function supported by the TMP68HC000 is not available because the TMP68301AK does not have 68000 interface signals.

Figure 4.1 shows a block diagram of the interrupt controller.

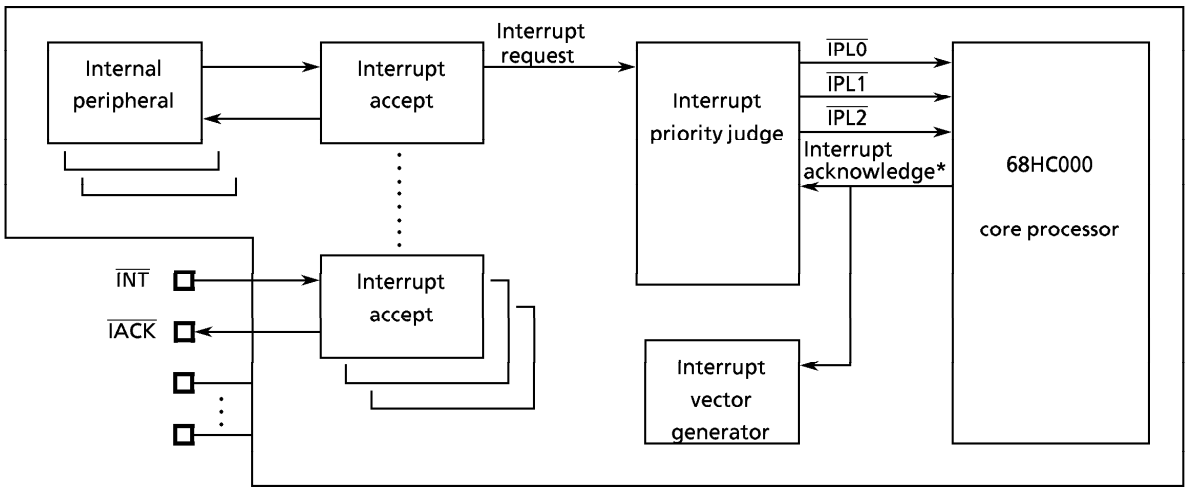


Figure 4.1 Interrupt Controller Block Diagram

4.2 Interrupt Request

When an interrupt request is present on an interrupt channel, the interrupt controller uses the internal $\overline{IPL0}$ to $\overline{IPL2}$ signals (not output to external pins in normal mode) to issue an interrupt request to the core processor at the previously set interrupt level. If requests are generated on more than one channel at the same time, the request with the highest priority level is issued.

The following input modes can be selected for external interrupt requests (interrupts using INT0, 1, or 2).

- Low-level interrupt
- High-level interrupt
- Rising-edge interrupt
- Falling-edge interrupt

Interrupt request inputs (INT0, 1, or 2) are detected on the falling edge of the system clock. Level-triggered interrupts must be asserted until the $\overline{IACK}$ output ($\overline{IACK0}$, $\overline{1}$, $\overline{2}$) for the channel corresponding to the interrupt request is asserted. Edge-triggered interrupts must be held at the same state for at least two system clocks after the edge to prevent malfunction due to noise.

Note : When switching from level mode to edge mode, the interrupt requests (pending bits) received in level mode are not cleared. To avoid this, clear any interrupt requests as follows:

1. Mask interrupt requests
2. Switch from level mode to edge
3. Clear the pending bits
4. Release the interrupt request mask

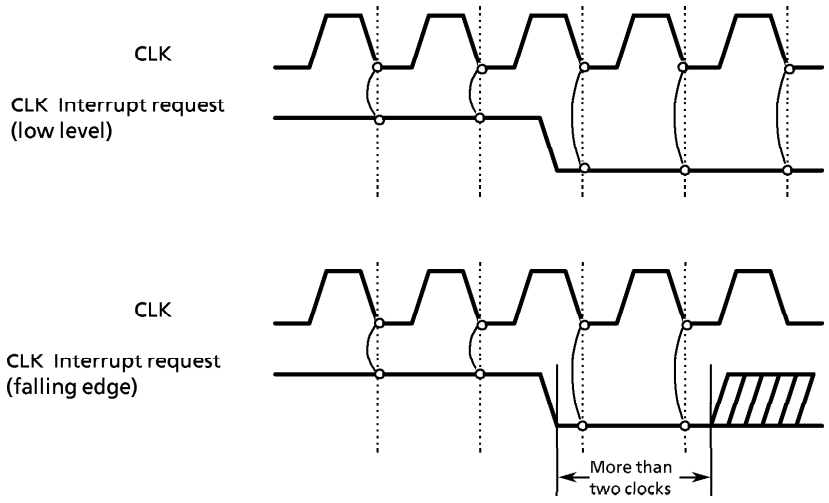


Figure 4.2 External Interrupt Request

4.3 Priority Between Channels

The interrupt controller can set the interrupt level of IPL0, 1, and 2 for interrupting the core processor using the level bit of the interrupt control register for each channel. This sets the relative priority of each channel. The following priority applies if more than one channel is set to the same interrupt level:

Priority	Channel	
High	External interrupt request	Channel 0
	Timer	Channel 0
	Serial interface	Channel 0
	External interrupt request	Channel 1
	Timer	Channel 1
	Serial interface	Channel 1
	Serial interface	Channel 2
	Timer	Channel 2
Low	External interrupt request	Channel 2

Table 4.1 Priority of Channels Set to Same Interrupt Request Level

4.4 Interrupt Acknowledge Cycle (IACK Cycle)

In 68000 interrupt processing, if an interrupt is accepted, the interrupt acknowledge cycle (IACK cycle) is performed. The interrupt request level is released on addresses A1 to A3 is output and the corresponding interrupt vector number is read from data bus D0 to D7.

The 68301A interrupt controller has a function to automatically generate the vector number to be read by the core processor during the $\overline{\text{IACK}}$ cycle. This function allows the interrupt controller to automatically perform the above interrupt processing. Also, when an external interrupt is accepted, the interrupt controller can assert the IACKn signal corresponding to the interrupt and obtain the vector externally.

If more than one channel issues requests at the same level, an interrupt acknowledge signal is asserted for the channel with the highest priority in Table 4.1. The interrupt acknowledge signal asserted here only applies internally to the 68301A and is not output externally. However, depending on the register setting, external interrupts can be output externally as $\overline{\text{IACK}}$ signals. $\overline{\text{IACK}}$ signals are asserted using the same timing as that for $\overline{\text{AS}}$ signals. Area control register 2 in the address decoder can be set to insert a WAIT into IACK cycles. For details, see the address decoder section.

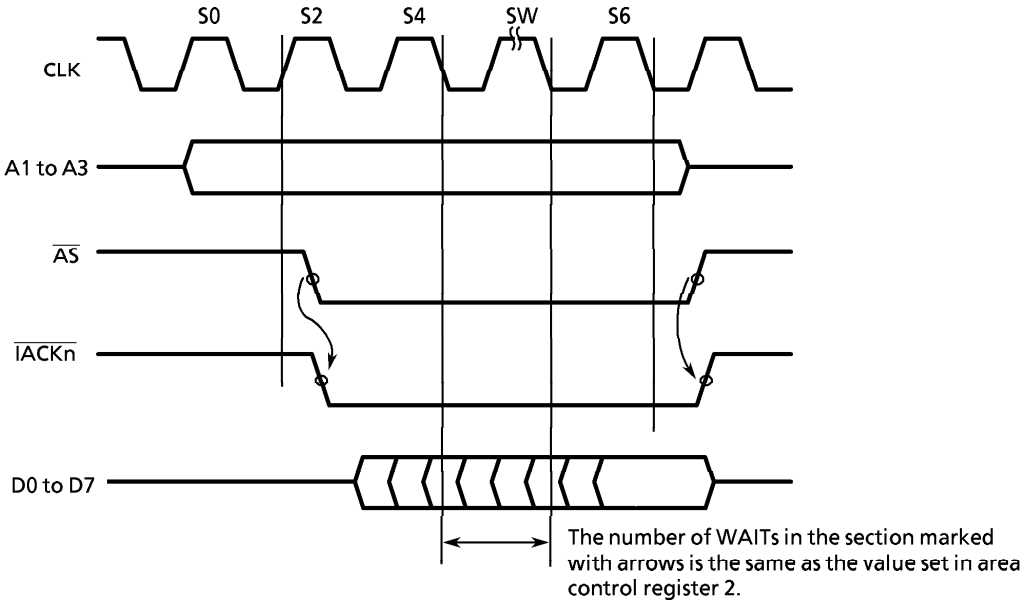


Figure 4.3 IACK Signals for External Vector Fetch by External Interrupt

4.5 Automatic Generation of Vector Numbers

The interrupt controller automatically generates vector numbers during IACK cycles and these are read by the core processor. The five lower bits of the vector are determined depending on the interrupt channel or request. The three upper bits are set using the interrupt vector number register (IVNR). See Table 4.2 for a list of vector numbers.

In the case of external interrupt requests, automatic generation or external vector input can be selected by setting the vector generation mode bit (V bit) of the interrupt control register (ICR0, 1, or 2).

Serial interface interrupt requests generate vector numbers not only corresponding to the channel generating the interrupt, but also in accordance with the cause of the interrupt.

Channel	Cause	Vector Number
External interrupt Channel 0		XXX00000
External interrupt Channel 1		XXX00001
External interrupt Channel 2		XXX00010
Timer 0 Channel 0		XXX00100
Timer 1 Channel 1		XXX00101
Timer 2 Channel 2		XXX00110
Serial interface Channel 0	Receive error, break detection	XXX01000
	Receive complete	XXX01001
	Transmit ready	XXX01010
	Interrupt source cleared while interrupt pending (Note 1)	XXX01011
Serial interface Channel 1	Receive error, break detection	XXX01100
	Receive complete	XXX01101
	Transmit ready	XXX01110
	Interrupt source cleared while interrupt pending (Note 1)	XXX01111
Serial interface Channel 2	Receive error, break detection	XXX10000
	Receive complete	XXX10001
	Transmit ready	XXX10010
	Interrupt source cleared while interrupt pending (Note1)	XXX10011
Other	Default vector (Note 2)	XXX11111

XXX : Set by the three upper bits of the IVNR.

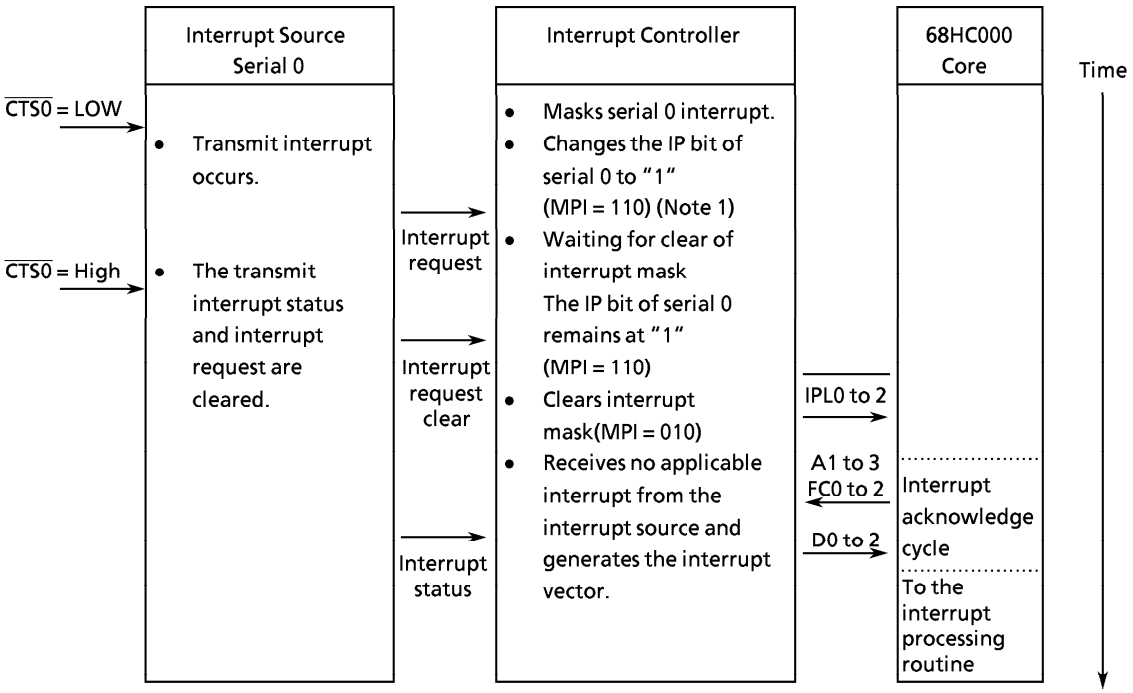
Table 4.2 Vector Number List

Note1: This vector number is generated when the cause of the interrupt becomes unknown if the interrupt source is cleared before the interrupt acknowledge is returned for a pending interrupt.

Note2: If the CPU accepts an interrupt, the IACK cycle starts after the following instruction is completed. However, if that instruction masks the interrupt, the cause becomes masked and the vector is fixed at "11111" because the vector cannot now be generated by the IACK cycle.

4.5.1 Interrupt Source Cleared While Interrupt Pending

This interrupt is generated under the following conditions. The following description is based on the generation of a serial 0 interrupt.



When an interrupt is generated by the interrupt source, an interrupt request is passed to the interrupt controller. As a result, the interrupt controller sets the relevant IP bit to "1". If the relevant interrupt is masked, the interrupt controller waits for the interrupt request to be issued to the 68HC000 core. (Note 1: MPI represents the bit status of the mask register, pending register, and in-service register for the generated interrupt). During this period, the interrupt condition may be cleared at the interrupt source. (The above example shows when the $\overline{CTS0}$ input changes to high in the transmit interrupt state). Even if the interrupt cause is cleared at the interrupt source, because interrupt control is still pending, the interrupt controller sends an interrupt request to the 68HC000 core when the interrupt mask is cleared. This starts the interrupt sequence and generates the interrupt vector. Because the vector reflects the state of the interrupt source, "no applicable interrupt" indicates that the interrupt cause was cleared at the interrupt source while the interrupt was pending, as mentioned previously. For serial interfaces, these kinds of interrupts occur under the following conditions:

(a) The interrupt is masked by the serial mode register (SMRn) while it is pending.

(b) The $\overline{CTS0}$ input changes to high while a transmit ready interrupt is pending. (TxRDY is set to "0" when $\overline{CTS0}$ changes to high, clearing the transmit interrupt cause.)

(c) In the serial command register (SCMRn), TxEN is set to "0" while a transmit ready interrupt is pending, or RxEN is set to "0" while a receive complete interrupt is pending.

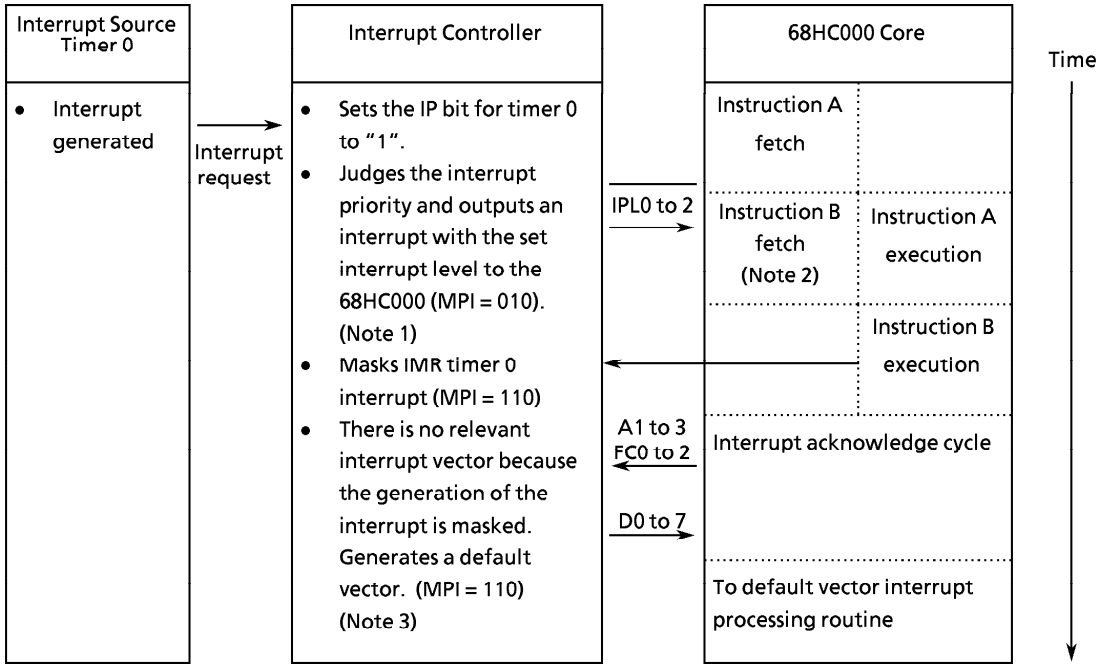
(d) The receive buffer is read by the error interrupt processing routine.

(If both ERINTM and RxINTM of the serial mode register [SMRn] are "0" and both interrupt masks are cleared, both error interrupt and receive interrupt are generated when an error occurs. Since the priority of the error interrupt is higher, the error interrupt processing routine is executed first. If the serial data register [SDRn] is read by this processing routine, the pending receive interrupt is cleared).

These interrupts occur due to software processing problems as described above. Therefore, check your software and make necessary modification so that these interrupts will not occur.

4.5.2 Default Vector Interrupts

Default vector interrupts occur under the following conditions. The following description is based on the generation of a timer 0 interrupt.



When an interrupt is generated, an interrupt request is passed to the interrupt controller. As a result, the interrupt controller sets the relevant IP bit to "1", checks that the interrupt is the highest priority interrupt, and outputs an interrupt with the set interrupt level to the 68HC000 core ($\overline{IPL0}$ to $\overline{2}$ output). The internal status of the interrupt controller at this time is as follows: mask bit = "1" (M=1), pending bit = "1" (P=1), and in-service bit = "0" (I=0). (Note 1: MPI represents the bit status of the mask register, pending register, and in-service register for the generated interrupt).

When the 68HC000 core accepts the $\overline{IPL0}$ to $\overline{2}$ interrupts, it attempts to jump to the interrupt sequence operations. However the interrupt sequence is delayed until instruction B fetched by the 68HC000 core is executed. If instruction B masks the generated interrupt (Note 2), the instruction is executed and the interrupt request by the interrupt controller is cancelled. Subsequently, even if the 68HC000 core starts the interrupt acknowledge cycle, the interrupt controller does not have a corresponding interrupt and so generates a default vector indicating that there is no relevant interrupt and ends the interrupt acknowledge cycle.

Even if a default vector interrupt is generated, the interrupt generated remains in the interrupt controller in the pending state (Note 3). Accordingly, after clearing the interrupt mask, a normal interrupt sequence can be performed.

If you simply wish to return to the main processing and disable vector generation when this default interrupt occurs, insert an instruction before the interrupt mask instruction (instruction B), to set the 68HC000 core interrupt level to the highest level. This prevents the 68HC000 core from receiving interrupts, apart from interrupt level 7, thus preventing initiation of the interrupt sequence midway through masking the interrupt. However, if the level of the interrupt generated is 7, the 68HC000 core cannot disable the interrupt request and the default vector is generated. In this case, perform default vector processing.

4.6 Interrupt Status

The interrupt status for each channel is represented by the mask register (IMR), pending register (IPR), and in-service register (IISR) bits corresponding to the channel. The meaning of these bits are described below. The set (setting the bit to "1") and reset (setting the bit to "0") methods vary according to a bit.

Mask Bit (M)	Control Bit for masking interrupt requests
1	Masks the interrupt request.
0	Unmasks the interrupt request.
set	Set by hardware reset or by writing "1" by software.
reset	Set to "0" by software.

Pending bit (P)	Indicates an interrupt request occurred and is pending (waiting for interrupt processing).
1	An interrupt request occurred and is pending.
0	No interrupt request occurred
set	An interrupt request occurred (this bit cannot be set to "1" by software).
reset	Reset by hardware reset or by writing "0" by software when the interrupt request is accepted by the core processor. (See Notes)

In-service bit (I)	Indicates that an interrupt request has been accepted by the core processor.
1	Indicates that an interrupt request has been accepted.
0	Indicates that no interrupt request has been accepted.
set	An interrupt request is accepted by the core processor (this bit cannot be set to "1" by software).
reset	Hardware reset Set to "0" by software

Note1 : The function to clear pending bits by software is used when initializing the whole system or when pending bits are set by unnecessary interrupts. However, if pending bits set according to interrupts generated by internal peripheral circuits are cleared to "0", interrupts will no longer occur. This is because the pending bit is only set when the interrupt request from the interrupt source changes from "0" to "1" (when an interrupt is generated). To re-enable interrupts after the pending bit has been cleared, the interrupt source must also be operated as follows.

Timer

First clear the interrupt request bit (INT bit) of the timer control register (TCRn) to "0", then set to "1".

Serial Interface

First set the interrupt mask bit (INTM bit) of the serial control register (TCRn) to "1", then clear to "0". If, at that time, an interrupt request is present due to an interrupt source in a channel, the corresponding IP bit is set immediately after the INTM bit is cleared to "0". To avoid this, disable interrupt requests for the channel (for example, by setting an interrupt mask for each channel, by reading the receive data, or by performing an error reset) before performing the above procedure.

External Interrupt

For edge mode interrupts, the IP bit is set the next time the interrupt edge is input. For level mode interrupts, the IP bit is set the next time the interrupt input is asserted.

Note2 : For level mode interrupts, clearing the IP bit by software requires first negating the interrupt input. The IP bit cannot be cleared by software while the interrupt input is still asserted.
When clearing the pending bit by software, write "1" to all bits except for the bit to be cleared. Writing "1" to the pending bit will not affect operation but enables interrupts to be generated even though the pending bit is mistakenly cleared.

The interrupt states for the mask bit (M), pending bit (P), and in-service bit (I) values are shown in Table 4.4.

M	P	I	
0	0	0	No interrupt request
0	0	1	During interrupt processing routine
0	1	0	Interrupt request generated
0	1	1	Another interrupt request is generated during an interrupt processing routine.
1	0	0	No interrupt request
1	0	1	Interrupt is masked during an interrupt processing routine
1	1	0	Interrupt request generated while interrupt is being masked
1	1	1	Interrupt request generated after masking interrupt during an interrupt processing routine.

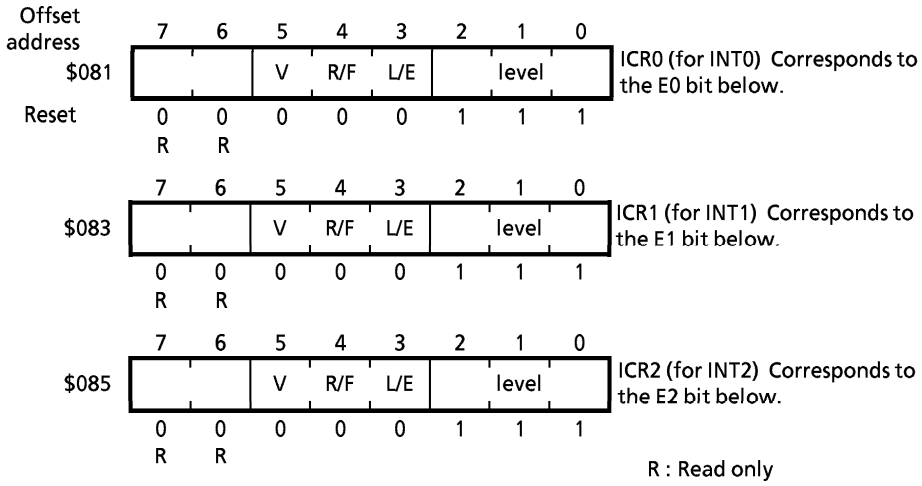
Table 4.3 Interrupt Status

4.7 Register Configuration

4.7.1 Interrupt Controller Registers 0, 1, and 2 (ICR0, 1, and 2)

These registers control the external interrupt inputs (INT0, 1, and 2). The registers set the interrupt level and select external vector input or automatic generation of the vector number set for input mode.

After a hardware reset, ICR0, 1, and 2 are all initialized to $07 (vector number from external source, falling edge mode, interrupt request level 7). These registers can only be written to when the interrupt is masked by the interrupt mask register (IMR).



V : Vector number automatic generation control

0 : Reads vector number from an external source instead of automatic generation of vector number

1 : Vector number automatic generation

R/F, L/E : Request input mode for external interrupts

R/F	L/E	Interrupt Request Input Mode
0	0	Falling edge
1	0	Rising edge
0	1	Low level
1	1	High level

Level : Interrupt request level

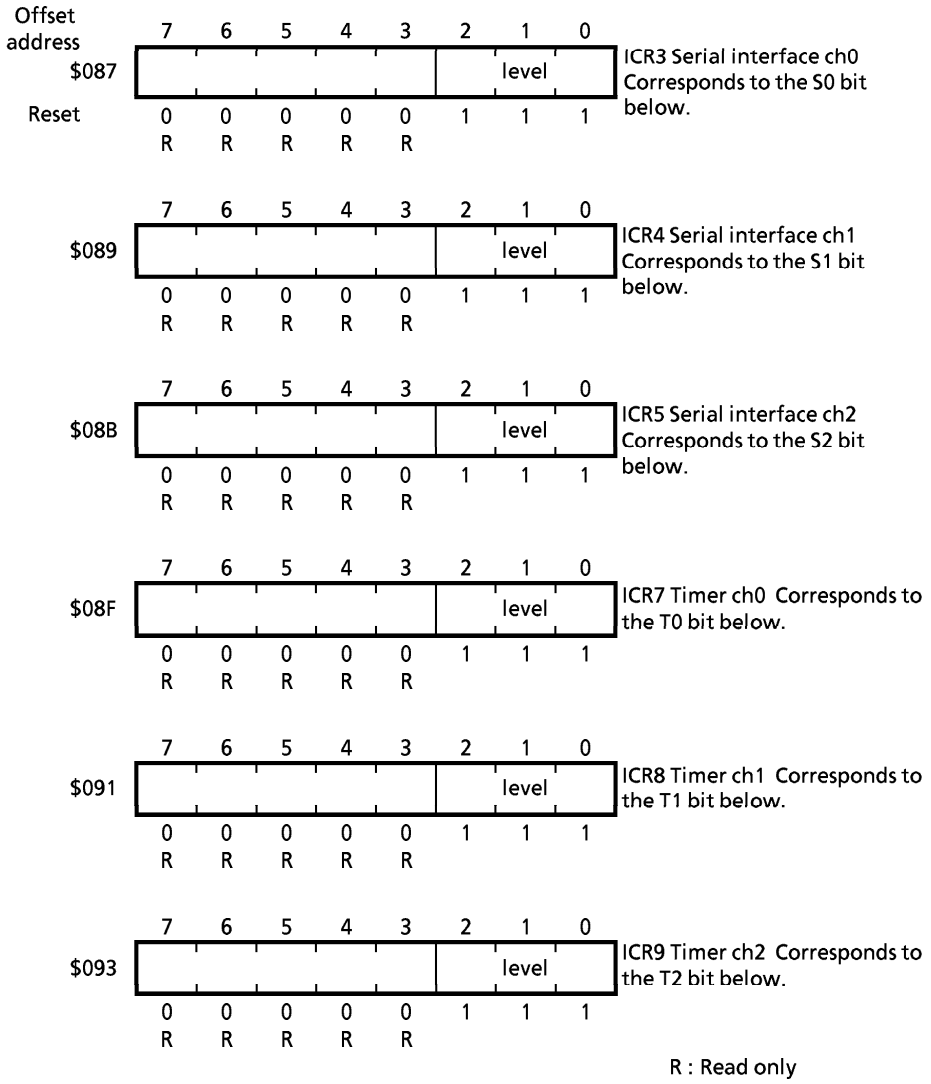
0 to 7 : Indicate the interrupt request level corresponding to $\overline{IPL0}$, $\overline{1}$, and $\overline{2}$ to be input to the core processor. For example, $\overline{IPL2} = \overline{1}$, $\overline{IPL1} = \overline{0}$, and $\overline{IPL0} = \overline{0}$ indicate request level 3. The following table shows the correspondence between IPLx and interrupt levels.

Interrupt level	$\overline{IPL2}$	$\overline{IPL1}$	$\overline{IPL0}$
0	1	1	1
1	1	1	0
2	1	0	1
3	1	0	0
4	0	1	1
5	0	1	0
6	0	0	1
7	0	0	0

4.7.2 Interrupt Control Registers 3 to 5, 7 to 9 (ICR3 to 5, 7 to 9)

These are the interrupt control registers for interrupts from internal peripheral circuits.

After a hardware reset, ICR3 to 5 and 7 to 9 are all initialized to $07 (interrupt request level 7). These registers can be written to only when the interrupt is masked by the interrupt mask register (IMR).



Level : Interrupt request level

0 to 7 : Indicate the interrupt request level corresponding to IPL0, 1, and 2 to be input to the core processor.

4.7.3 Interrupt Mask Register (IMR)

This register sets the interrupt mask for a channel. A "1" masks the interrupt (ignore interrupt). A "0" unmasks the interrupt (allow interrupt). If masking an interrupt during operation, set the channels corresponding to the bits to be modified to a state whereby, even if an interrupt is generated, it will not be accepted by the core processor. (For example, set the interrupt level in the core processor status register to 7). This is necessary because, if an interrupt occurs during masking, the interrupt to be masked may be generated due to the timing mismatch.

After a hardware reset, this register is initialized to $07F7 (all interrupt channels masked).

Offset address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$094						T2	T1	T0	P	S2	S1	S0		E2	E1	E0	IMR
Reset	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	
	R	R	R	R	R								R				

T2 : Timer channel 2
T1 : Timer channel 1
T0 : Timer channel 0
S2 : Serial interface channel 2
S1 : Serial interface channel 1
S0 : Serial interface channel 0
E2 : External interrupt channel 2
E1 : External interrupt channel 1
E0 : External interrupt channel 0
P : Parallel interface (only the case that IO12 is used for $\overline{INT6}$ (External interrupt))

R : Read only

Note : Bit 7 is normally set to "1".

4.7.4 Interrupt Pending Register (IPR)

This register indicates whether there is an interrupt request and that the interrupt request is not yet accepted by the core processor. A "1" indicates that there is an interrupt request that has not yet been accepted by the core processor. A "0" indicates that there is no interrupt request.

Each bit is automatically cleared when the request is accepted by the core processor. Clearing a bit using software cancels the interrupt request. However, to enable subsequent interrupts, the interrupt source must be cleared.

After a hardware reset, this register is initialized to $0000 (no interrupt requests).

Offset address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$096						T2	T1	T0	P	S2	S1	S0		E2	E1	E0	IPR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	R	R	R	R	R								R				

R : Read only

4.7.5 Interrupt In-Service Register (IISR)

This register indicates whether or not an interrupt request has been accepted by the core processor. A "1" indicates that a request has been accepted. A "0" indicates that a request has not been accepted.

While operating with the register set to "1" does not affect operation, clear each bit during the interrupt processing routine. (These bits are not cleared automatically).

After a hardware reset, the register is initialized to $0000 (no interrupt requests accepted).

Offset address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$098						T2	T1	T0	P	S2	S1	S0		E2	E1	E0	IISR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	R	R	R	R	R								R				

R : Read only

4.7.6 Interrupt Vector Number Register (IVNR)

This register specifies the three upper bits of a vector number. The five lower bits of a vector number are determined by the interrupt channel or the interrupt source.

After a hardware reset, the register is initialized to $00.

Note : After a reset is released, the 68HC000 core interrupt vectors overlap the internal peripheral circuit interrupt vectors. Therefore, set the value of IVNR to $40 or more by software.

Offset address	7	6	5	4	3	2	1	0	
$09B	Vector								IVNR
Reset	0	0	0	0	0	0	0	0	
				R	R	R	R	R	

R : Read only

Vector : Three upper bits of the vector number

4.8 Interrupt Expansion Function

In the TMP68301A, interrupt channels assigned to unused peripheral circuits can be used as external interrupt inputs by setting the expansion interrupt register. Thus, a maximum of seven channels can be used for external interrupt inputs in addition to the three standard external interrupt channels.

The interrupt input pins are assigned as follows:

Peripheral circuit interrupt channel	Interrupt input pin	Interrupt input name
Serial interface ch0	RxD0	$\overline{INT3}$
Serial interface ch1	RxD1	$\overline{INT4}$
Serial interface ch2	RxD2	$\overline{INT5}$
	IO12	$\overline{INT6}$ (Note)
Timer ch0	TIN	$\overline{INT7}$
Timer ch1	IO14 / $\overline{DSR0}$	$\overline{INT8}$ (Note)
Timer ch2	IO15 / $\overline{DTR0}$	$\overline{INT9}$ (Note)

Note : When used as interrupt inputs, the pins must be first set to input in the parallel direction register.

Only falling-edge input mode is available for expanded external interrupts. However, like the standard external interrupt inputs, the state must be held for at least two clocks after the falling edge.

Multiple falling edges may occur before the processor accepts an interrupt request. The processor treats these edges as if they were the same interrupt request.

There is no IACK output signal corresponding to the expanded external interrupts. Therefore, the vector number is generated automatically by the interrupt controller because the vector number cannot be input from an external source during the interrupt acknowledge cycle. Vector numbers generated at this time are as follows.

Interrupt channel	Interrupt input name	Vector number
Serial interface ch0	$\overline{INT3}$	XXX010**
Serial interface ch1	$\overline{INT4}$	XXX011**
Serial interface ch2	$\overline{INT5}$	XXX100**
	$\overline{INT6}$	XXX101**
Timer ch0	$\overline{INT7}$	XXX00100
Timer ch1	$\overline{INT8}$	XXX00101
Timer ch2	$\overline{INT9}$	XXX00110

XXX : The three upper bits specified by the vector number register

** : The two lower bits of the interrupt vector number assigned to the serial interface depend on the status of the standard interrupt channels. Accordingly, when using these interrupts, set the same destination address in all four. The same applies to $\overline{INT6}$.

4.8.1 Expansion Interrupt Register (IEIR)

This register controls the switching of interrupt channels between internal peripheral circuits and external interrupt inputs. A "1" indicates external interrupt input. A "0" indicates interrupts from the internal peripheral circuits (external interrupt input disabled).

After a hardware reset, this register is initialized to $00 (all channels set to internal peripheral circuit interrupts).

Offset address	7	6	5	4	3	2	1	0	
$09D		T2	T1	T0	P	S2	S1	S0	IEIR
Reset	0	0	0	0	0	0	0	0	
	R								R : Read only

T2 : Timer ch2

T1 : Timer ch1

T0 : Timer ch0

P : (Note 2)

S2 : Serial interface ch2

S1 : Serial interface ch1

S0 : Serial interface ch0

Note1: Even if external interrupt input is enabled, the internal peripheral circuits can perform their original functions. However, the pins assigned as interrupt inputs (for example, the serial interface RxD) cannot be used for their original functions.

The peripheral circuits can function but cannot generate interrupts if the interrupt channels are used for interrupt inputs.

Note2: When using IO12 for $\overline{INT6}$, set this register to "1".

5. Serial Interface

Refer to 5. Serial Interface in the TMP68301A manual.

6. Parallel Interface

6.1 Overview

This parallel interface may be used as a general-use,16-bit I/O port where input or output may be specified for each bit.

6.2 Operation Mode

The parallel interface uses mode 0 as its operation mode. Several types of pin functions can be selected, and part of the parallel interface I/O port can be used as a serial interface.

6.2.1 Mode 0 Operation

In mode 0, the port operates as a 16-bit I/O port.

Each parallel interface bit can be configured for either input or output using the direction register. The input pins pass data received from an external devices through the internal buffer and set them in the data register. The output pins pass data from the data register through the internal buffer and output them to external devices.

6.3 Register Configuration

6.3.1 Parallel Direction Register

This register is used to program each bit of the 16-bit port for input or output. A "1" specifies output. A "0" specifies input.

Offset address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$100	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0	PDIR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

6.3.2 Parallel Data Register

This register is used to read or write data input from or output to the I/O ports. When a port is in input mode, data from an external device received by the receive buffer are immediately set in this register. Any data written in at this time are ignored.

When a port is in output mode, data written to the corresponding bit in this register are output to the external port. If the register is read at this time, the data currently output from the port will be read.

Offset address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$10A	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	PDR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

6.3.3 Parallel Control Register

This register selects operation modes and pin functions. M0 and M1 specify operation modes. PF0 and PF1 specify pin functions. The relationship between operation modes and pin functions is shown in the diagram below:

Offset address	7	6	5	4	3	2	1	0	
$103					PF1	PF0	M1	M0	PCR
Reset	0 R	0 R	0 R	0 R	0	0	0	0	

Note : Fix both M0 and M1 to "0".

R : Read only

Mode 0 (M0 = 0, M1 = 0)	
PF1 = X	
PF0 = 0	PF0 = 1
I/O 0	I/O 0
I/O 1	I/O 1
I/O 2	I/O 2
I/O 3	I/O 3
I/O 4	I/O 4
I/O 5	I/O 5
I/O 6	I/O 6
I/O 7	I/O 7
I/O 8	I/O 8
I/O 9	I/O 9
I/O 10	I/O 10
I/O 11	I/O 11
I/O 12	I/O 12
I/O 13	$\overline{\text{CTS0}}$
I/O 14	$\overline{\text{DSR0}}$
I/O 15	$\overline{\text{DTR0}}$

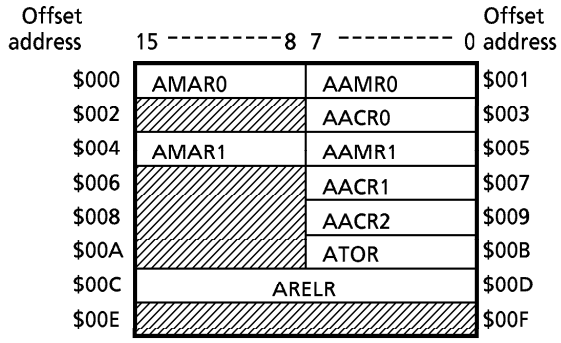
7. Timer

Refer to 7. Timer in the TMP68301A manual.

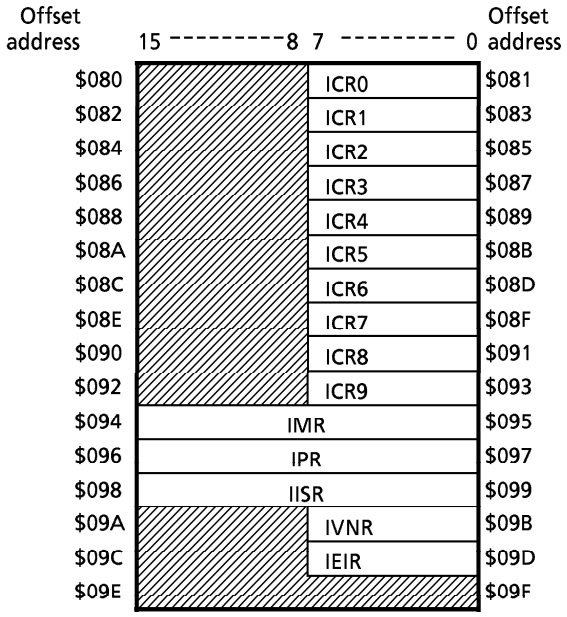
8. Internal Peripheral Circuit Register Map

8.1 Register Map (1)

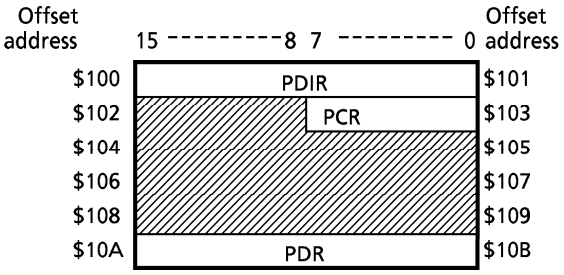
○ Address Decoder



○ Interrupt Controller



○ Parallel Interface



○ Serial Interface

Offset address	15 ---------8	7 --------- 0	Offset address
$180		SMR0	$181
$182		SCMR0	$183
$184		SBRR0	$185
$186		SSR0	$187
$188		SDR0	$189
$18A			$18B
$18C		SPR	$18D
$18E		SCR	$18F
$190		SMR1	$191
$192		SCMR1	$193
$194		SBRR1	$195
$196		SSR1	$197
$198		SDR1	$199
$19A			$19B
$19C			$19D
$19E			$19F
$1A0		SMR2	$1A1
$1A2		SCMR2	$1A3
$1A4		SBRR2	$1A5
$1A6		SSR2	$1A7
$1A8		SDR2	$1A9
$1AA			$1AB
$1AC			$1AD
$1AE			$1AF

○ 16-Bit Timer

Offset address	15 ---------8	7 --------- 0	Offset address
$200		TCR0	$201
$202			$203
$204		TMCR01	$205
$206			$207
$208			$209
$20A			$20B
$20C		TCTR0	$20D
$20E			$20F
$210			$211
$212			$213
$214			$215
$216			$217
$218			$219
$21A			$21B
$21C			$21D
$21E			$21F
$220		TCR1	$221
$222			$223
$224		TMCR11	$225
$226			$227
$228		TMCR12	$229
$22A			$22B
$22C		TCTR1	$22D
$22E			$22F
$230			$231
$232			$233
$234			$235
$236			$237
$238			$239
$23A			$23B
$23C			$23D
$23E			$23F
$240		TCR2	$241
$242			$243
$244		TMCR21	$245
$246			$247
$248		TMCR22	$249
$24A			$24B
$24C		TCTR2	$24D
$24E			$24F

8.2 Register Map (2)

○ Address Decoder

Symbol	Name	Offset address	Data bus Upper bytes : 15 to 8 (even-numbereaddresses) Lower bytes : 7 to 0 (odd-numbered d addresses)							
			15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
AMAR0	Memory Address Register 0	$000	A23	A22	A21	A20	A19	A18	A17	A16
			0	0	0	0	0	0	0	0
	For CS0		R/W							
AAMR0	Address Mask Register 0	$001	M21	M20	M19	M18	M17	M16	M15-M9	M8
			1	1	1	1	1	1	1	1
	For CS0		R/W							
AACR0	Area Control Register 0	$003			EN	ED	ID	WAIT		
			0	0	1	1	1	1	0	1
	For CS0		R		R/W					
AMAR1	Memory Address Register 1	$004	A23	A22	A21	A20	A19	A18	A17	A16
			–	–	–	–	–	–	–	–
	For CS0		R/W							
AAMR1	Address Mask Register 1	$005	M21	M20	M19	M18	M17	M16	M15-M9	M8
			–	–	–	–	–	–	–	–
	For CS1		R/W							
AACR1	Area Control Register 1	$007			EN	ED	ID	WAIT		
			0	0	0	1	1	0	0	0
	For CS1		R		R/W					
AACR2	Area Control Register 2	$009				ED	ID	WAIT		
			0	0	0	1	1	0	0	0
	For IACK cycle		R			R/W				
ATOR	Time Out Register	$00B					256	128	64	32
			0	0	0	0	1	0	0	0
	For BERR generation		R				R/W			
ARELR	Relocaton Register	$00C	A23	A22	A21	A20	A19	A18	A17	A16
			1	1	1	1	1	1	1	1
			R/W							
		$00D	A15	A14	A13	A12	A11	A10		
			1	1	1	1	1	1	0	0
	For internal register		R/W						R	

○ Interrupt Controller (1)

Symbol	Name	Offset address	Data bus Upper bytes : 15 to 8 (even-numbered addresses) Lower bytes : 7 to 0 (odd-numbered addresses)							
			15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
ICR0	Interrupt Control Register 0 For external interrupt (INT0)	$081			V	R/F	L/E	Level		
			0	0	0	0	0	1	1	1
			R		R/W					
ICR1	Interrupt Control Register 1 For external interrupt (INT0)	$083			V	R/F	L/E	Level		
			0	0	0	0	0	1	1	1
			R		R/W					
ICR2	Interrupt Control Register 2 For external interrupt (INT0)	$085			V	R/F	L/E	Level		
			0	0	0	0	0	1	1	1
			R		R/W					
ICR3	Interrupt Control Register 3 For serial ch0 (INT3)	$087						Level		
			0	0	0	0	0	1	1	1
			R					R/W		
ICR4	Interrupt Control Register 4 For serial ch1 (INT4)	$089						Level		
			0	0	0	0	0	1	1	1
			R					R/W		
ICR5	Interrupt Control Register 5 For serial ch2 (INT5)	$08B						Level		
			0	0	0	0	0	1	1	1
			R					R/W		
ICR6	Interrupt Control Register 6 For parallel (INT6)	$08D						Level		
			0	0	0	0	0	1	1	1
			R					R/W		
ICR7	Interrupt Control Register 7 For timer ch0 (INT7)	$08F						Level		
			0	0	0	0	0	1	1	1
			R					R/W		
ICR8	Interrupt Control Register 8 For timer ch1 (INT8)	$091						Level		
			0	0	0	0	0	1	1	1
			R					R/W		
ICR9	Interrupt Control Register 9 For timer ch2 (INT9)	$093						Level		
			0	0	0	0	0	1	1	1
			R					R/W		

○ Interrupt Controller (2)

Symbol	Name	Offset address	Data bus Upper bytes: 15 to 8 (even-numbered addresses) Lower bytes: 7 to 0 (odd-numbered addresses)							
			15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
IMR	Interrupt Mask Register	$094						T2	T1	T0
			0	0	0	0	0	1	1	1
			R					R/W		
		$095	P	S2	S1	S0		E2	E1	E0
			1	1	1	1	0	1	1	1
			R/W				R	R/W		
IPR	Interrupt Pending Register	$096						T2	T1	T0
			0	0	0	0	0	0	0	0
			R					R/W		
		$097	P	S2	S1	S0		E2	E1	E0
			0	0	0	0	0	0	0	0
			R/W				R	R/W		
IISR	Interrupt In Service Register	$098						T2	T1	T0
			0	0	0	0	0	0	0	0
			R					R/W		
		$099	P	S2	S1	S0		E2	E1	E0
			0	0	0	0	0	0	0	0
			R/W				R	R/W		
IVNR	Interrupt Vector Number Register	$09B	Vector							
			0	0	0	0	0	0	0	0
			R/W				R			
IEIR	Expansion Interrupt Register For external interrupt expansion	$09D		T2	T1	T0	P	S2	S1	S0
			0	0	0	0	0	0	0	0
			R	R/W						

○ Parallel Interface

Symbol	Name	Offset address	Data bus Upper bytes : 15 to 8 (even-numbered addresses) Lower bytes : 7 to 0 (odd-numbered addresses)							
			15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
PDIR	Parallel Direction Register	$100	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8
			0	0	0	0	0	0	0	0
			R/W							
		$101	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
			0	0	0	0	0	0	0	0
			R/W							
PCR	Parallel Control Register	$103					PF1	PF0	M1	M0
			0	0	0	0	0	0	0	0
			R				R/W			
PDR	Parallel Data Register	$10A	D15	D14	D13	D12	D11	D10	D9	D8
			0	0	0	0	0	0	0	0
			R/W							
		$10B	D7	D6	D5	D4	D3	D2	DR	D0
			0	0	0	0	0	0	0	0
			R/W							

○ Serial Interface (1)

Symbol	Name	Offset address	Data bus Upper bytes : 15 to 8 (even-numbered addresses) Lower bytes : 7 to 0 (odd-numbered addresses)							
			15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
SMR0	Serial Mode Register 0 For ch0	$181	RxINTM	ErINTM	PE0	PEN	CL1	CL0	TxINTM	ST
			1	1	–	–	–	–	1	–
			R/W							
SCMR0	Serial Command Register 0 For ch0	$183			RTS	ERS	SBRK	RxEN	DTR	TxEN
			0	0	0	1	0	0	0	0
			R		R/W					
SBRR0	Serial Baudrate Register 0 For ch0	$185	B7	B6	B5	B4	B3	B2	B1	B0
			0	0	0	0	0	0	0	0
			R/W							
SSR0	Serial Status Register For ch0	$187	DSR	RBRK	FE	OE	PE	TxE	RxRDY	TxRDY
			0	0	0	0	0	1	0	0
			R							
SDR0	Serial Data Register 0 For ch0	$189	D7	D6	D5	D4	D3	D2	D1	D0
			–	–	–	–	–	–	–	–
			R/W							
SPR	Serial Prescaler Register	$18D	P7	P6	P5	P4	P3	P2	P1	P0
			–	–	–	–	–	–	–	–
			R/W							
SCR	Serial Control Register	$18F	CKSE		RES					INTM
			1	0	1	0	0	0	0	1
			R/W	R	R/W	R				R/W

○ Serial Interface (2)

Symbol	Name	Offset address	Data bus Upper bytes : 15 to 8 (even-numbered addresses) Lower bytes : 7 to 0 (odd-numbered addresses)							
			15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
SMR1	Serial Mode Register 1	$191	RxINTM	ErINTM	PE0	PEN	CL1	CL0	TxINTM	ST
			1	1	–	–	–	–	1	–
	For ch1		R/W							
SCMR1	Serial Command Register 1	$193				ERS	SBRK	RxEN		TxEN
			0	0	0	1	0	0	0	0
	For ch1		R			R/W			R	R/W
SBRR1	Serial Baudrate Register 1	$195	B7	B6	B5	B4	B3	B2	B1	B0
			0	0	0	0	0	0	0	0
	For ch1		R/W							
SSR1	Serial Status Register 1	$197		RBRK	FE	OE	PE	TxE	RxRDY	TxRDY
			0	0	0	0	0	1	0	0
	For ch1		R							
SDR1	Serial Data Register 1	$199	D7	D6	D5	D4	D3	D2	D1	D0
			–	–	–	–	–	–	–	–
	For ch1		R/W							
SMR2	Serial Mode Register 2	$1A1	RxINTM	RxINTM	PE0	PEN	CL1	CL0	TxINTM	ST
			1	1	–	–	–	–	1	–
	For ch2		R/W							
SCMR2	Serial Command Register 2	$1A3				ERS	SBRK	RxEN		TxEN
			0	0	0	1	0	0	0	0
	For ch2		R			R/W			R	R/W
SBRR2	Serial Baudrate Register 2	$1A5	B7	B6	B5	B4	B3	B2	B1	B0
			0	0	0	0	0	0	0	0
	For ch2		R/W							
SSR2	Serial Status Register 2	$1A7		RBRK	FE	OE	PE	TxE	RxRDY	TxRDY
			0	0	0	0	0	1	0	0
	For ch2		R							
SDR2	Serial Data Register 2	$1A9	D7	D6	D5	D4	D3	D2	D1	D0
			–	–	–	–	–	–	–	–
	For ch2		R/W							

○ 16-Bit Timer (1)

Symbol	Name	Offset address	Data bus Upper bytes : 15 to 8 (even-numbered addresses) Lower bytes : 7 to 0 (odd-numbered addresses)							
			15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
TCR0	Timer Control Register 0	$200	CK2	CK1	P4	P3	P2	P1	T2	T1
			0	0	0	0	0	0	0	0
			R/W							
		$201	N/1					INT	CS	TS
			0	1	0	1	0	0	0	1
	For ch0		R/W	R				R/W		
TMCR01	Timer MAX Count Register 01	$204	M15	M14	M13	M12	M11	M10	M9	M8
			–	–	–	–	–	–	–	–
			R/W							
		$205	M7	M6	M5	M4	M3	M2	M1	M0
			–	–	–	–	–	–	–	–
	For ch0		R/W							
TCTR0	Timer Count Register 0	$20C	C15	C14	C13	C12	C11	C10	C9	C8
			0	0	0	0	0	0	0	0
			R							
		$20D	C7	C6	C5	C4	C3	C2	C1	C0
			0	0	0	0	0	0	0	0
	For ch0		R							
TCR1	Timer Control Register 1	$220	CK2	CK1	P4	P3	P2	P1	T2	T1
			0	0	0	0	0	0	0	0
			R/W							
		$221	N/1	R/P	MR2	MR1		INT	CS	TS
			0	0	0	1	0	0	1	0
	For ch1		R/W				R	R/W		
TMCR11	Timer MAX Count Register 11	$224	M15	M14	M13	M12	M11	M10	M9	M8
			–	–	–	–	–	–	–	–
			R/W							
		$225	M7	M6	M5	M4	M3	M2	M1	M0
			–	–	–	–	–	–	–	–
	For ch1		R/W							
TMCR12	Timer MAX Count Register 12	$228	M15	M14	M13	M12	M11	M10	M9	M8
			–	–	–	–	–	–	–	–
			R/W							
		$229	M7	M6	M5	M4	M3	M2	M1	M0
			–	–	–	–	–	–	–	–
	For ch1		R/W							

○ 16-Bit Timer (2)

Symbol	Name	Offset address	Data bus Upper bytes : 15 to 8 (even-numbered addresses) Lower bytes : 7 to 0 (odd-numbered addresses)							
			15/7	14/6	13/5	12/4	11/3	10/2	9/1	8/0
TCTR1	Timer Count Register 1	$22C	C15	C14	C13	C12	C11	C10	C9	C8
			0	0	0	0	0	0	0	0
			R							
		$22D	C7	C6	C5	C4	C3	C2	C1	C0
			0	0	0	0	0	0	0	0
	For ch1		R							
TCR2	Timer Control Register 2	$240	CK2	CK1	P4	P3	P2	P1	T2	T1
			0	0	0	0	0	0	0	0
			R/W							
		$241	N/1	R/P	MR2	MR1		INT	CS	TS
			0	0	0	1	0	0	1	0
	For ch2		R/W				R	R/W		
TMCR21	Timer MAX Count Register 21	$244	M15	M14	M13	M12	M11	M10	M9	M8
			–	–	–	–	–	–	–	–
			R/W							
		$245	M7	M6	M5	M4	M3	M2	M1	M0
			–	–	–	–	–	–	–	–
	For ch2		R/W							
TMCR22	Timer MAX Count Register 22	$248	M15	M14	M13	M12	M11	M10	M9	M8
			–	–	–	–	–	–	–	–
			R/W							
		$249	M7	M6	M5	M4	M3	M2	M1	M0
			–	–	–	–	–	–	–	–
	For ch2		R/W							
TCTR2	Timer Count Register 2	$24C	C15	C14	C13	C12	C11	C10	C9	C8
			0	0	0	0	0	0	0	0
			R							
		$24D	C7	C6	C5	C4	C3	C2	C1	C0
			0	0	0	0	0	0	0	0
	For ch2		R							

9. Electrical Specifications

9.1 Maximum Ratings

This section describes the electrical characteristics and timing of the TMP68301AK.

Parameter	Symbol	Rating	Unit
		TMP68301AK	
Power supply voltage	Vcc	– 0.3 to + 6.5	V
Input voltage	Vin	– 0.3 to + 6.5	V
Operating temperature	Ta	– 0 to + 70	°C
Storage temperature	Tstg	– 55 to + 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, normal precautions should be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or Vcc).

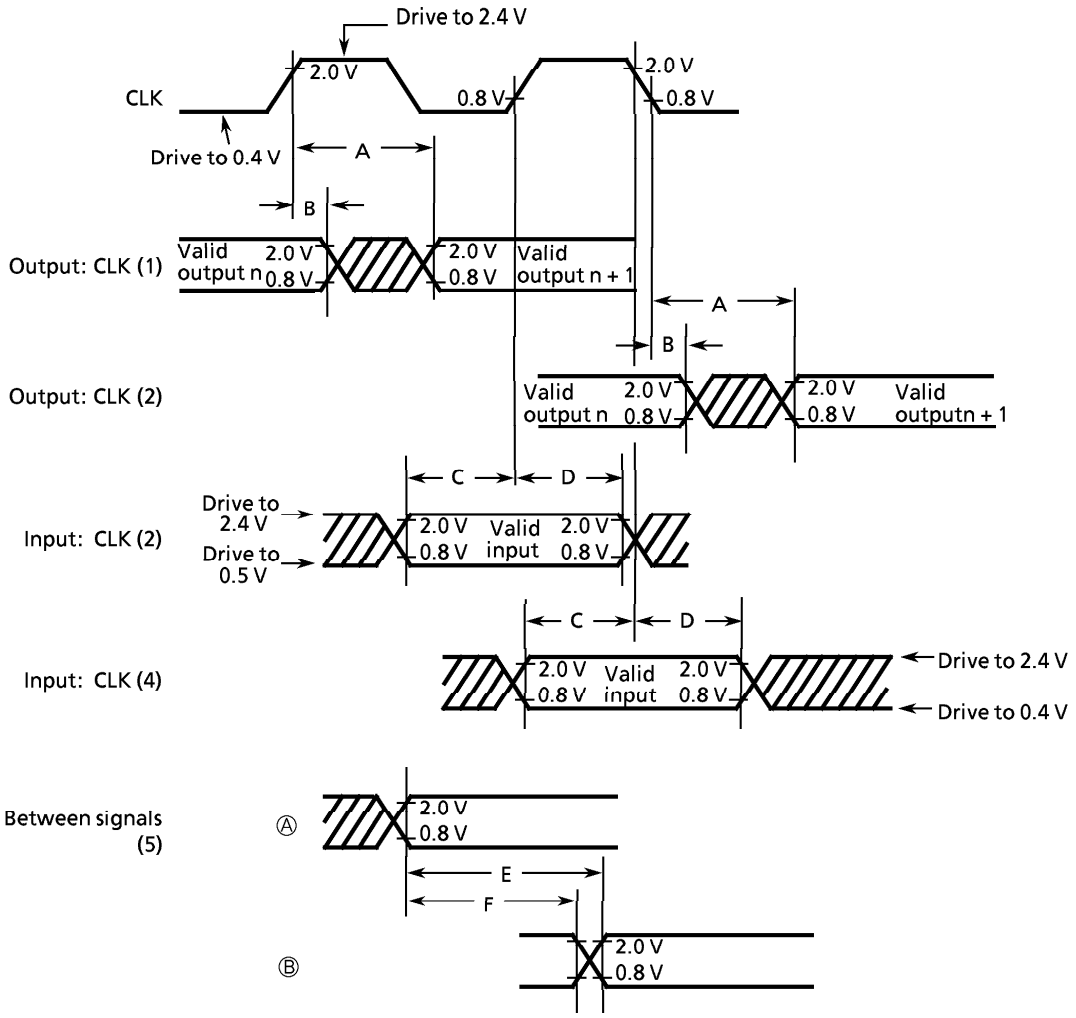
9.2 DC Electrical Specifications

(GND = 0 V, Ta = 0 to 70 °C)

Characteristic		Symbol	VCC = 3.3 V ± 10 %		Unit
			Min	Max	
Supply voltage		V_{CC}	3.0	3.6	V
Input high voltage except CLK	CLK CLK	V_{IH}	2.0 2.0	V_{CC} V_{CC}	V
Input low voltage except CLK	CLK CLK	V_{IL}	GND-0.3 GND-0.3	0.6 0.5	V
Input leakage current (3.6 V)	$\overline{BERR}$, $\overline{BGACK}$, $\overline{BR}$, $\overline{DTACK}$, CLK, $\overline{HALT}$, $\overline{RESET}$, NOR / $\overline{EMU}$, TIN, INT0 to INT2, RxD0 to RxD2	I_{IN}	– – –	2.5 2.5 20	μA
Tri-state (off state) input current	(2.4 V / 0.4 V) $\overline{AS}$, A1 to A23, D0 to D15, FC0 to FC2, $\overline{LDS}$, $\overline{UDS}$, R / $\overline{W}$, IO0 to IO15	I_{TSI}	 – – –	 20 20 20	μA
Output high voltage (IOH = – 400 μA)	$\overline{AS}$, A1 to A23, D0 to D15, $\overline{BG}$, FC0 to FC2, $\overline{LDS}$, $\overline{UDS}$, R / $\overline{W}$, IO0 to IO15, TOUT1, TOUT2, CS0, CS1, $\overline{RTS0}$, INT0 to INT2, TxD0 to TxD2	V_{OH}	– V_{CC}-0.5 V_{CC}-0.5 V_{CC}-0.5 V_{CC}-0.5	 – – – –	V
Output low voltage (IOL = 0.8 mA) (IOL = 1.6 mA) (IOL = 0.8 mA) (IOL = 2.7 mA)	 $\overline{HALT}$ A1 to A23, $\overline{BG}$, FC0 to FC2 $\overline{RESET}$ $\overline{AS}$, D0 to D15, $\overline{LDS}$, $\overline{UDS}$, R / W, IO0 to IO15, TOUT1, TOUT2, CS0, CS1, $\overline{RTS0}$, $\overline{DTACK}$, $\overline{BERR}$, INT0 to INT2, TxD0 to TxD2	V_{OL}	 – – – –	 0.5 0.5 0.5 0.5	V
Current dissipation	f = 8.0 MHz	I_D	–	50	mA
Power dissipation	f = 8.0 MHz	P_D	–	0.18	W
Input capacitance Frequency (Vin = 0 V, Ta = 25 °C : f = 1 MHz)*		C_{IN}	–	20.0	pF
Load capacitance	$\overline{HALT}$ Other pins	C_L	– –	70 130	pF

* : Input capacitance is periodically sampled rather than 100 % tested.

AC Electrical Specification Definitions (V_{CC} = 3.3 V ± 10 %)

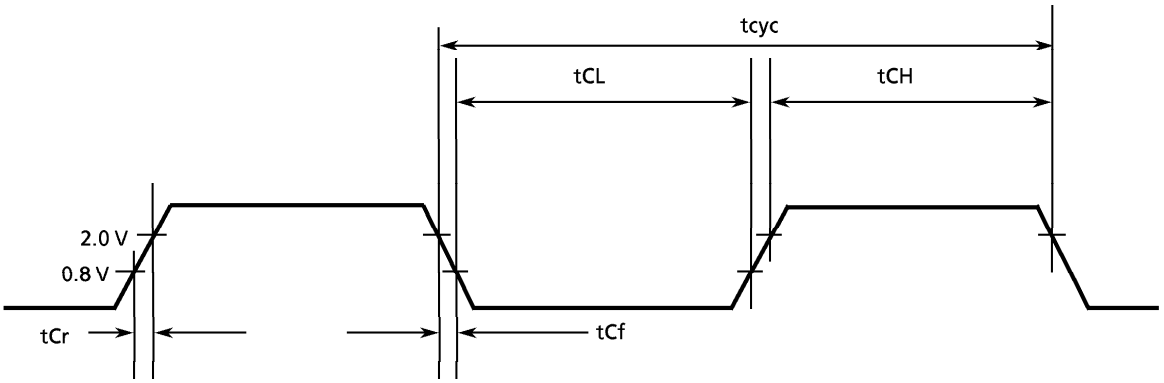


A : Maximum output delay
B : Minimum output hold time
C : Minimum input setup time
D : Minimum input hold time
E : Signal Ⓐ valid - signal Ⓑ valid time
F : Signal Ⓐ valid - signal Ⓑ invalid time

9.3 AC Electrical Specifications - Clock Timing

VCC = 3.3 V ± 10 % See Figure 9.1
GND = 0 V, Ta = 0 to 70 °C

Characteristic	Symbol	8 MHz		Unit
		Min	Max	
Operating frequency	f	4.0	8	MHz
Cycle time	tcyc	125	1000	ns
Clock pulse width	tCL tCH	52 52	500 500	ns
Rise and fall times	tCr tCf	– –	10 10	ns



Note : Timing measurements are referenced between a low voltage of 0.8 V and a high voltage of 2.0 V unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8 V and 2.0 V.

Figure 9.1 Clock Input Timing

9.4 AC Electrical Specifications - Read and Write Cycles (1/4)

(V_{CC} = 3.3 V ± 10 %, GND = 0 V, Ta = 0 to 70 °C; See figures 9.2 and 9.3.)

Number	Characteristic	Symbol	8.0 MHz		Unit
			Min	Max	
1	Clock period	tCYC	125	1000	ns
2	Clock width low	tCL	52	500	ns
3	Clock width high	tCH	52	500	ns
4	Clock fall time	tCf	–	10	ns
5	Clock rise time	tCr	–	10	ns
6	Clock low to address valid	tCLAV	–	80	ns
6A	Clock high to FC valid	tCHFCV	–	72	ns
7	Clock high to address, data bus high impedance (maximum)	tCHADZ	–	80	ns
8	Clock high to address, FC invalid (minimum)	tCHAFI	0	–	ns
9[1]	Clock high to $\overline{AS}$, $\overline{DS}$ low	tCHSL	3	60	ns
11[2]	Address valid to $\overline{AS}$, $\overline{DS}$ low (read) / Address valid to $\overline{AS}$ low (write)	tAVSL	20	–	ns
11A[2]	FC valid to $\overline{AS}$, $\overline{DS}$ low (read) / FC valid to $\overline{AS}$ low (write)	tFCVSL	90	–	ns
12[1]	Clock low to $\overline{AS}$, $\overline{DS}$ high	tCLSH	–	72	ns
13[2]	AS, DS high to address / FC invalid	tSHAFI	40	–	ns
14[2]	$\overline{AS}$, $\overline{DS}$ width low (read) / $\overline{AS}$ width low (write)	tSL	270	–	ns
14A[2]	$\overline{DS}$ width low (write)	tDSL	140	–	ns
15[2]	$\overline{AS}$, $\overline{DS}$ width high	tSH	150	–	ns
16	Clock high to control bus high impedance	tCHCZ	–	80	ns
17[2]	$\overline{AS}$, $\overline{DS}$ high to R/$\overline{W}$ high (read)	tSHRH	40	–	ns
18[1]	Clock high to R/$\overline{W}$ high	tCHRH	0	62	ns
20[1]	Clock to R/$\overline{W}$ low (write)	tCHRL	0	62	ns
20A[2,6]	$\overline{AS}$ low to R/$\overline{W}$ valid (write)	tASRV	–	10	ns
21[2]	Address valid to R/$\overline{W}$ low (write), FC valid to R/$\overline{W}$ low (write)	tAVRL	0	–	ns
21A[2]	FC valid to R/$\overline{W}$ low (write)	tFCVRL	60	–	ns
22[2]	R/$\overline{W}$ low to $\overline{DS}$ low (write)	tRLSL	80	–	ns
23	Clock low to data out valid (write)	tCLDO	–	80	ns
25[2]	$\overline{AS}$, $\overline{DS}$ high to data out invalid (write)	tSHDOI	40	–	ns
26[2]	Data out valid to $\overline{DS}$ low (write)	tDOSL	30	–	ns
27[5]	Data in to clock low (setup time on read)	tDICL	15	–	ns
28[2]	$\overline{AS}$, $\overline{DS}$ high to $\overline{DTACK}$ high	tSHDAH	0	240	ns
29	$\overline{AS}$, $\overline{DS}$ negate to data invalid (hold time on read)	tSHDII	0	–	ns

9.4 AC Electrical Specifications - Read and Write Cycles (2/4)

(V_{CC} = 3.3 V ± 10 %, GND = 0 V, Ta = 0 to 70 °C; See figures 9.2 and 9.3)

Number	Characteristic	Symbol	8.0MHz		Unit
			Min	Max	
30	$\overline{AS}$, $\overline{DS}$ high to $\overline{BERR}$ high	tSHBEH	0	–	ns
31[2,5]	$\overline{DTACK}$ low to data in (setup time)	tDALDI	–	90	ns
32	$\overline{HALT}$ and $\overline{RESET}$ input transition	tRHr, f	0	200	ns
33	Clock high to $\overline{BG}$ low	tCHGL	–	62	ns
34	Clock high to $\overline{BG}$ high	tCHGH	–	62	ns
35	$\overline{BR}$ low to BG low	tBRLGL	1.5	3.5	Clk. Per.
36[7]	$\overline{BR}$ high to $\overline{BG}$ high	tBRHGH	1.5	3.5	Clk. Per.
37	$\overline{BGACK}$ low to $\overline{BG}$ low	tGALGH	1.5	3.5	Clk. Per.
37A[8]	$\overline{BGACK}$ low to $\overline{BR}$ high	tGALBRH	20	1.5 Clocks	ns
38	$\overline{BG}$ low to control, address, data bus, high impedance ($\overline{AS}$ high)	tGLZ	–	80	ns
39	$\overline{BG}$ width high	tGH	1.5	–	Clk. Per.
46	$\overline{BGACK}$ width low	tGAL	1.5	–	Clk. Per.
47[5]	Asynchronous input setup time	tASI	15	–	ns
48[2,3]	$\overline{BERR}$ low to $\overline{DTACK}$ low	tBELDAL	20	–	ns
53	Clock high to data out invalid	tCHDOI	0	–	ns
55	R/$\overline{W}$ low to data bus drive	tRLDBD	30	–	ns
56[4]	$\overline{HALT}$ / $\overline{RESET}$ pulse width	tHRPW	10	–	Clk. Per.
57	$\overline{BGACK}$ high to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ drive	tGASD	1.5	–	Clk. Per.
58[7]	$\overline{BR}$ high to control bus driven	tRHSD	1.5	–	Clk. Per.

Note1 : For a loading capacitance of 50 [pF] or less, subtract 3 [ns] from the value given in the maximum columns.

Note2 : Actual value depends on clock period.

Note3 : If #47 is satisfied for both $\overline{DTACK}$ and $\overline{BERR}$, #48 may be 0 [ns]

Note4 : For power up, the MPU must be help in $\overline{RESET}$ state from 1 ms to 100 ms to allow stabilization of on-chip circuitry. After the system is powered up, #56 refers to the minimum pulse width required to reset the system.

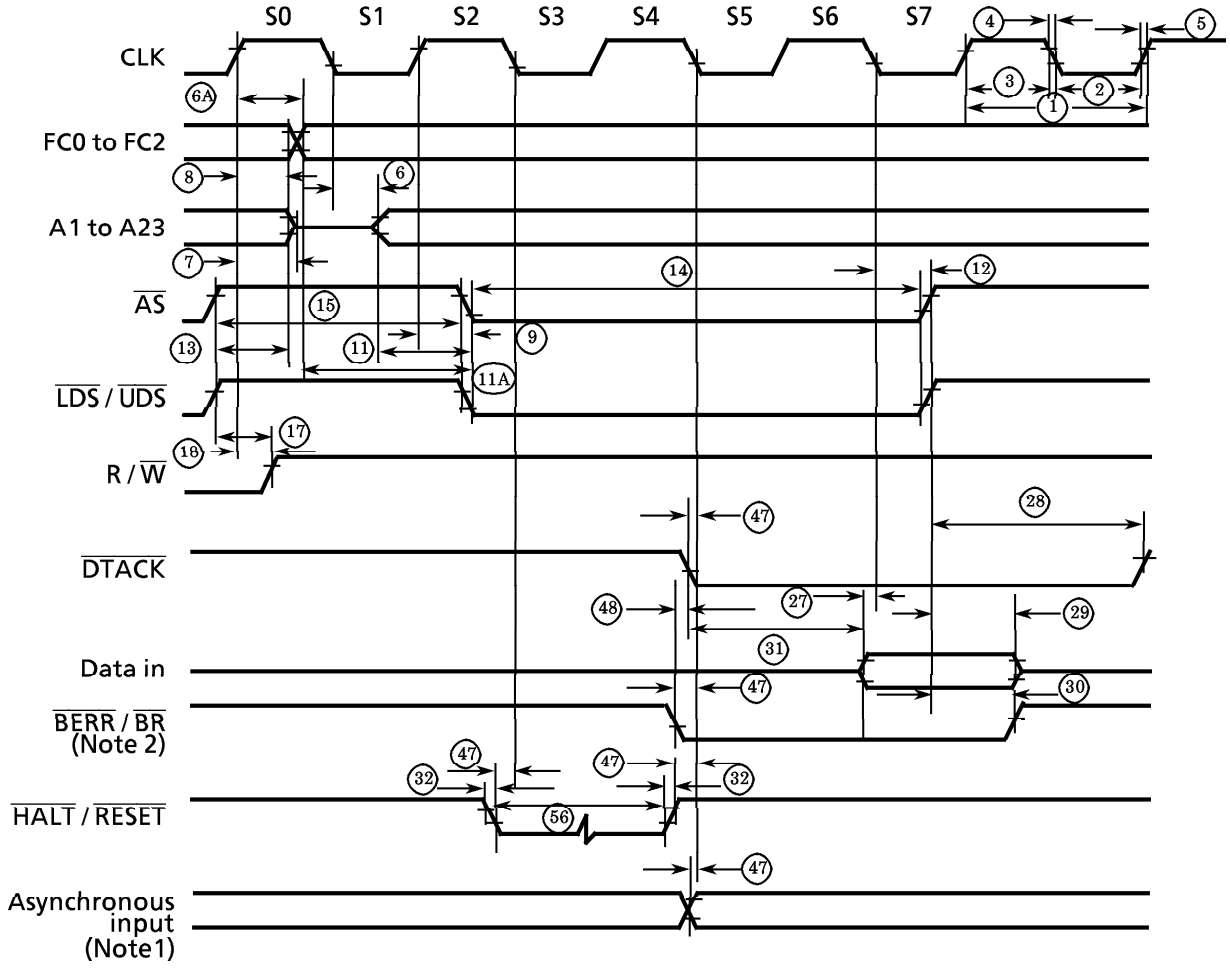
Note5 : If the asynchronous setup time (#47) requirements are satisfied, the $\overline{DTACK}$ low-to-data setup time (#31) requirement can be ignored. The data must satisfy the data-in to clock-low setup time (#27) requirement for the following cycle.

Note6 : When $\overline{AS}$ and R/$\overline{W}$ are equally loaded (±20 %), subtract 3 [ns] from the tASRV maximum values.

Note7 : If external arbitration logic negates $\overline{BR}$ before asserting $\overline{BGACK}$, the TMP68301 will negate $\overline{BG}$ and begin driving the bus again .

Note8 : The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{BG}$ may be reasserted.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related device operation diagrams.



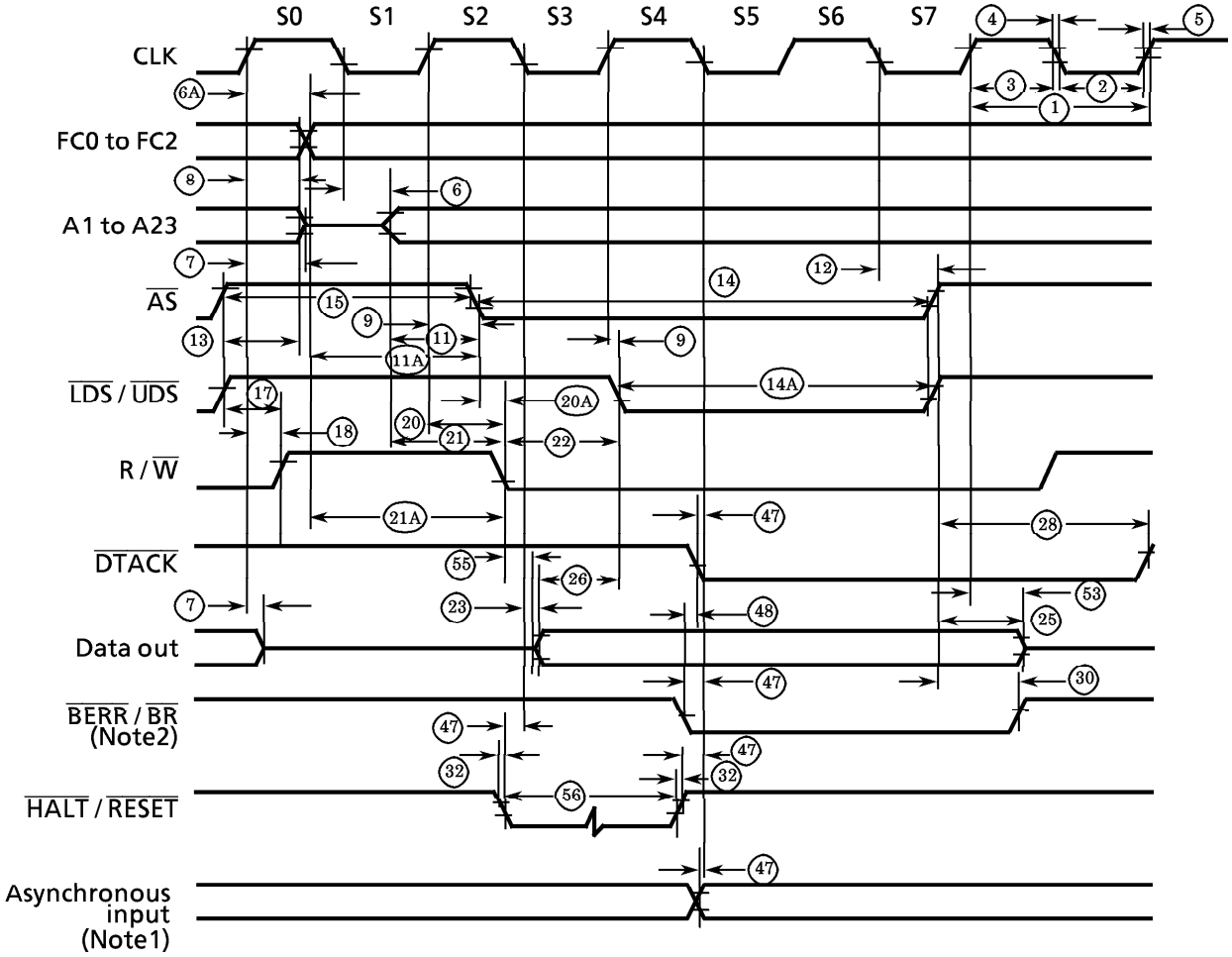
Note1 : The asynchronous input $\overline{\text{BGACK}}$ is detected on the falling edge of the clock.

Note2 : Asserting $\overline{\text{BR}}$ at this timing is only necessary when BR is recognized at the end of this bus cycle.

Note3 : Timing measurements are referenced between a low voltage of 0.8 V and a high voltage of 2.0 V unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8 V and 2.0 V.

Figure 9.2 Read Cycle Timing Diagram

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related device operation diagrams.



Note1 : Timing measurements are referenced between a low voltage of 0.8 V and a high voltage of 2.0 V unless otherwise noted. The voltage swing through this range should start outside and pass through the range such that the rise or fall will be linear between 0.8 V and 2.0 V.

Note2 : Because of loading variation, R/$\overline{W}$ may be valid after $\overline{AS}$ even though both are asserted at the rising edge of S2 (Specification 20 A).

Figure 9.3 Write Cycle Timing Diagram

9.5 AC Electrical Specifications - Bus Arbitration

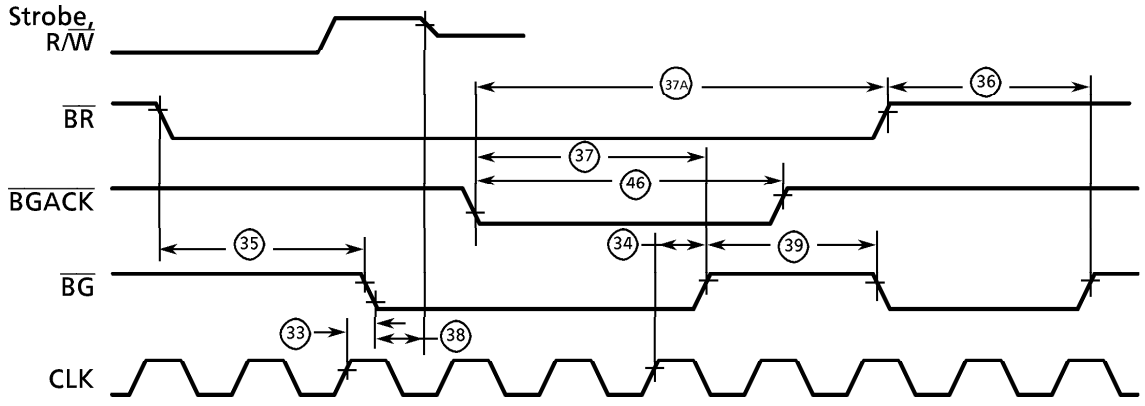
(V_{CC} = 3.3 V ± 10 %, GND = 0 V, Ta = 0 to 70 °C; Figure 9.4)

Number	Characteristic	Symbol	8.0 MHz		Unit
			Min	Max	
7	Clock high to address, data bus high impedance	tCHADZ	–	80	ns
16	Clock high to control bus high impedance	tCHCZ	–	80	ns
33	Clock high to $\overline{BG}$ low	tCHGL	–	62	ns
34	Clock high to $\overline{BG}$ high	tCHGH	–	62	ns
35	$\overline{BR}$ low to $\overline{BG}$ low	tBRLGL	1.5	3.5	Clk. Per.
36[1]	$\overline{BR}$ high to $\overline{BG}$ high	tBKHGH	1.5	3.5	Clk. Per.
37	$\overline{BGACK}$ low to $\overline{BG}$ high	tGALGH	1.5	3.5	Clk. Per.
37A[2]	$\overline{BGACK}$ low to $\overline{BR}$ high	tGALBRH	20	1.5 Clocks	ns
38	$\overline{BG}$ low to control, address, data bus high impedance (AS high)	tGLZ	–	80	ns
39	$\overline{BG}$ width high	tGH	1.5	–	Clk. Per.
46	$\overline{BGACK}$ width low	tGAL	1.5	–	Clk. Per.
47	Asynchronous input setup time	tASI	15	–	ns
57	$\overline{BGACK}$ high to control bus driven	tGABD	1.5	–	Clk. Per.
58[1]	$\overline{BG}$ high to control bus driven	tGHBD	1.5	–	Clk. Per.

Note1 : If external arbitration logic negates $\overline{BR}$ before asserting $\overline{BGACK}$, the processor will negate $\overline{BG}$ and begin driving the bus again .

Note2 : The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{BG}$ may be reasserted.

These waveforms should only be referenced in regard to the edge-to-edge measurement of the timing specifications. They are not intended as a functional description of the input and output signals. Refer to other functional descriptions and their related device operation diagrams.



Note : Asynchronous inputs $\overline{BERR}$, $\overline{BGACK}$, $\overline{BR}$, and $\overline{DTACK}$ are detected at the falling edge of the clock.

Figure 9.4 Bus Arbitration Timing Diagram

When the external bus master accesses the registers of internal devices, addresses, data, and control signals must be input in accordance with the read / write cycle timing.

9.6 AC Electrical Specifications - Peripherals

(V_{CC} = 3.3 V ± 10 %, GND = 0 V, Ta = 0 to 70 °C; See figures 9.5 - 9.11)

Number	Characteristic	Symbol	8.0 MHz		Unit
			Min	Max	
47	Asynchronous input setup time	tASI	15	—	ns
101	Clock to $\overline{CS}$, $\overline{IACK}$	tCDS	—	90	ns
102	Clock high to TOUT	tCHTO	—	80	ns
103	BCLK cycle time	tBCYC	125	—	ns
104	BCLK width low	tBCL	55	—	ns
105	BCLK width high	tBCH	55	—	ns
106	BCLK rise time	tBCr	—	10	ns
107	BCLK fall time	tBCf	—	10	ns
108	LDS high to DTR, RTS	tDSMC	—	140	ns
109	$\overline{DSR}$ to $\overline{LDS}$ low	tMCDS	50	—	ns
110	$\overline{DS}$ high to I/O output	tDSIO	—	60	ns
111	I/O input setup to CLK low	tIOsCL	50	—	ns
112	I/O input hold from CLK low	tIOhCL	50	—	ns

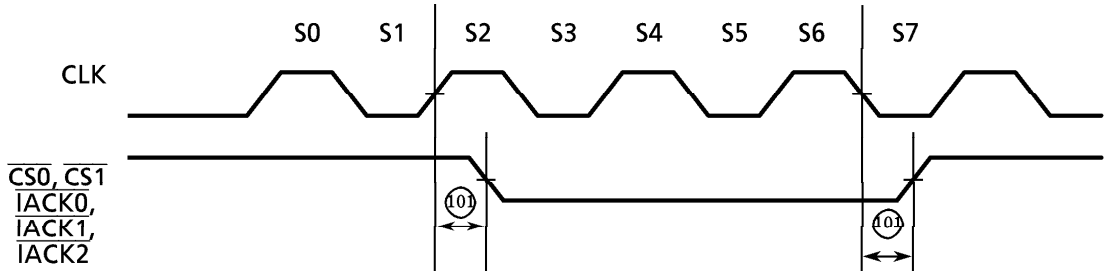


Figure 9.5 CS, IACK Timing Diagram

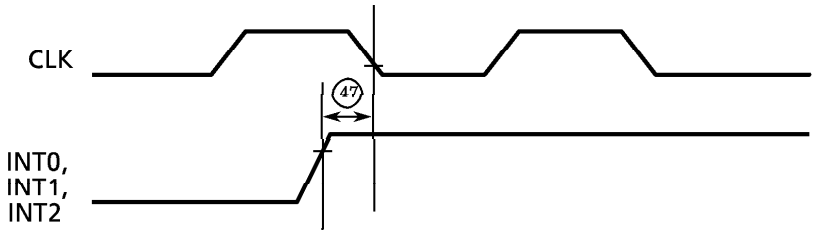


Figure 9.6 Interrupt Request Timing Diagram

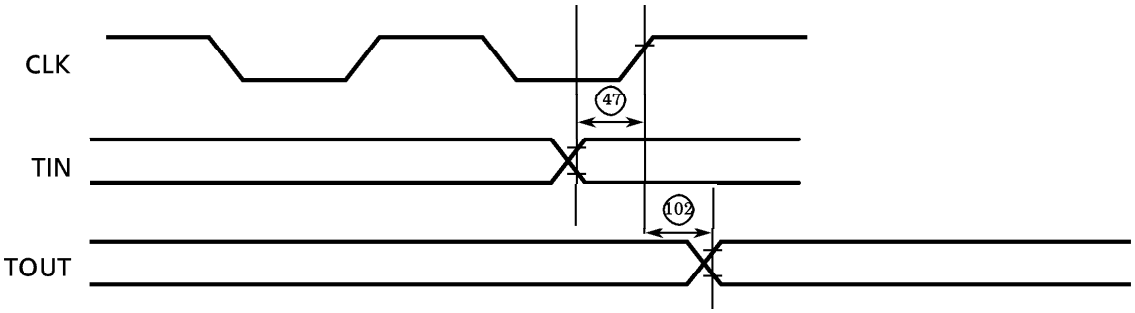


Figure 9.7 Timer Input / Output Timing Diagram

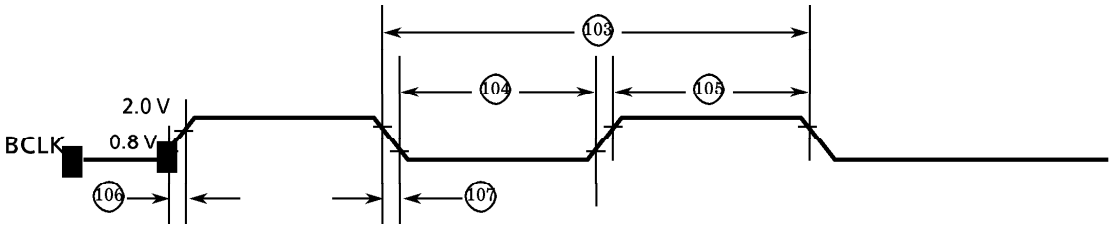


Figure 9.8 Baud rate Clock Timing Diagram

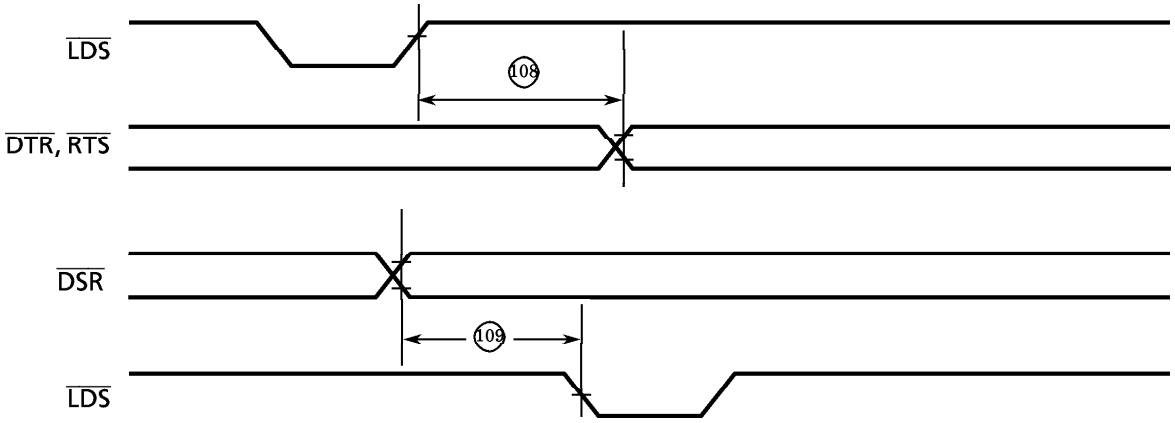


Figure 9.9 Serial Port Timing Diagram

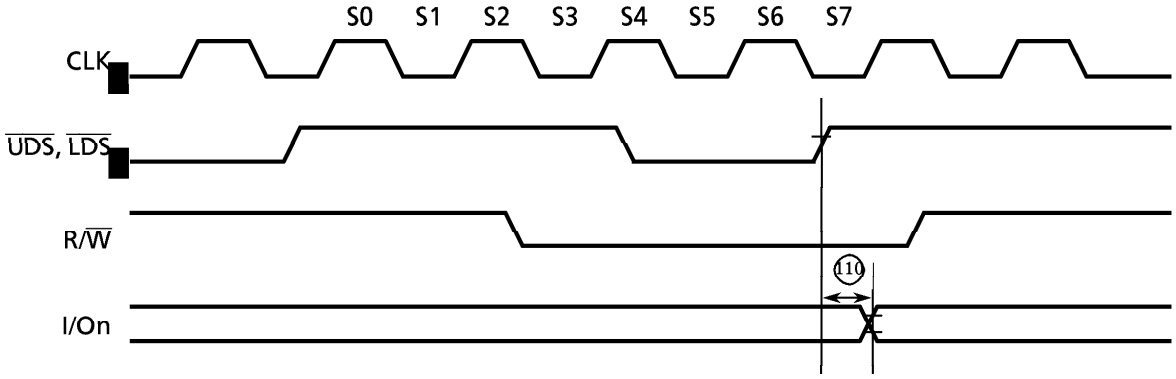


Figure 9.10 I/O Port Output Timing Diagram

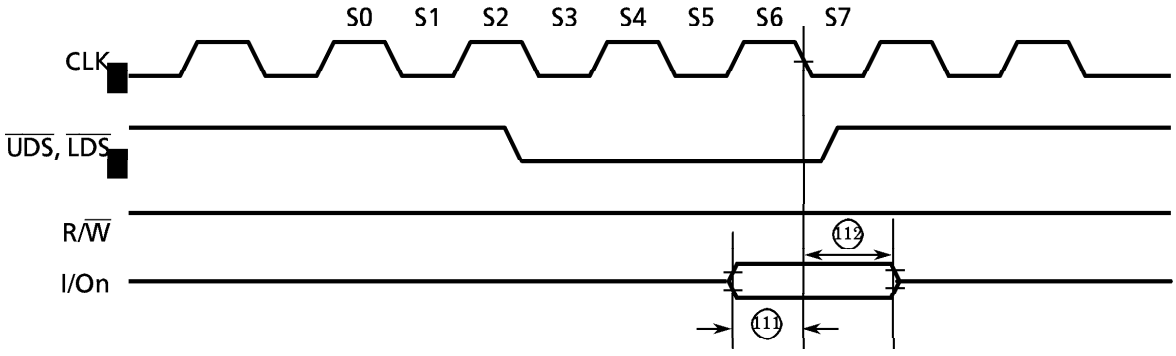


Figure 9.11 I/O Port Input Timing Diagram

10. Development Environment

Refer to 10. Development Environment in the TMP68301A manual.

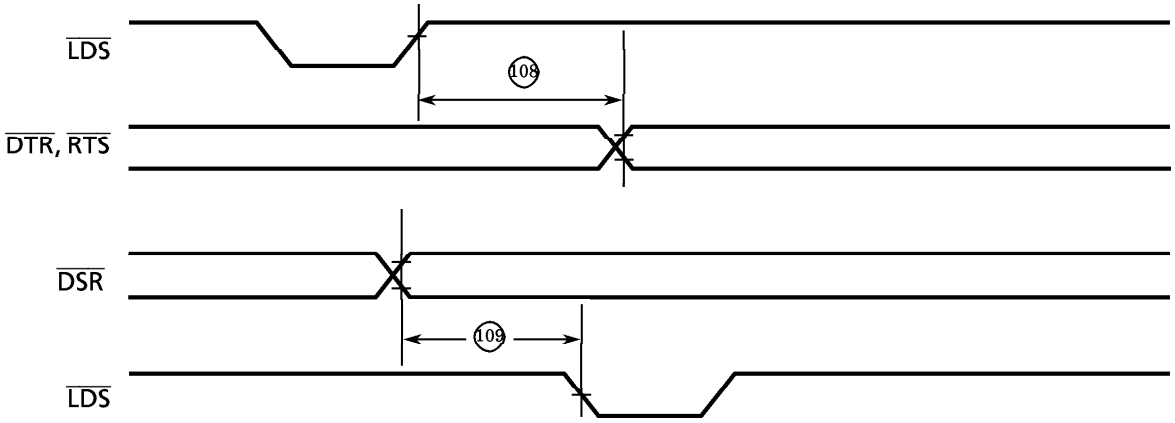


Figure 9.9 Serial Port Timing Diagram

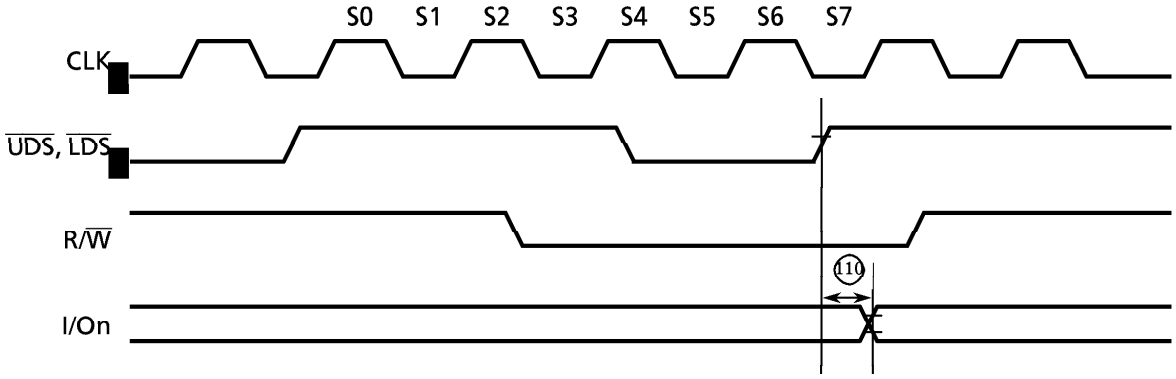


Figure 9.10 I/O Port Output Timing Diagram

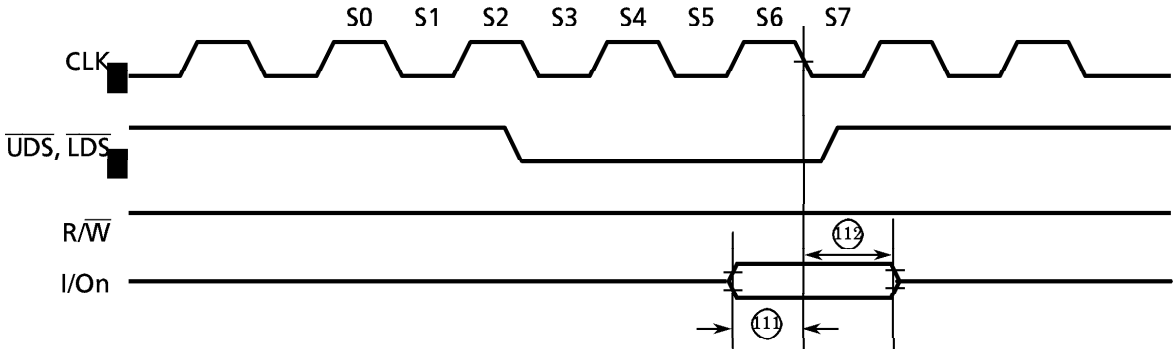


Figure 9.11 I/O Port Input Timing Diagram

10. Development Environment

Refer to 10. Development Environment in the TMP68301A manual.