

May 2001

### EXAR XR17C158/154/152 PCI UART EEPROM INTERFACE

Author: KG

#### 1.0 INTRODUCTION

The Exar XR17C158/154/152 (XR17C15x) PCI UART provides an interface to the 93C46 non-volatile memory device, also known as Electrically Erasable Programmable Read Only Memory (EEPROM) as we will call it in this application note. The 93C46 is an industry standard part and supplied by several vendors. The 93C46 is a 1K-bit memory device that data can be formatted to 64x16 or 128x8. The XR17C15x EEPROM interface uses the 64x16 data format. Exar selected the 93C46 for its small size and there is no limitation on the size of the EEPROM used. Customer can use a larger memory size device when its system needs the bigger storage space, and the controlling software determines that.

#### 2.0 HARDWARE INTERFACE

The 93C46 EEPROM is offered by Catalyst, Exel, Fairchild, Microchip and Xicor. The XR17C15x interfaces to the EEPROM through 4 input/output pins. The EECS, pin 115, is the chip select signal output; EECK, pin 116, is the clock output; EEDI, pin 114, is the serial data output or input to the EEPROM and EEDO, pin 113, is serial data input or output from the EEPROM. Figure 1 below shows the connections.

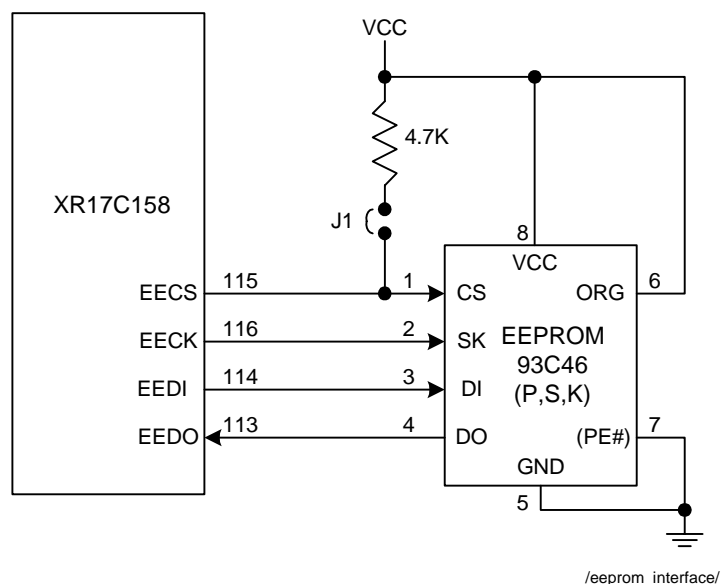


FIGURE 1. XR17C15x CONNECTIONS TO EEPROM

#### 3.0 SOFTWARE/FIRMWARE

During power up, on the low to high transition of the reset pin, the state of the EECS pin is sampled to determine an external EEPROM is present or absent. The XR17C15x has a weak internal pull down resistor making the default to "no EEPROM". The printed circuit board must provide an external 4.7K Ohms pull-up resistor to enable the reading of external EEPROM memory values. If EECS pin is sampled at logic "1", the XR17C15x reads four word data (described below) from the EEPROM's first four word locations and places those values

**DAN112**

into the PCI configuration space registers for chipset identification. If EECS pin is a logic "0", the XR17C15x uses the default value of Exar's Vendor and Device ID. The ID's are stored in the configuration space for chipset identification during the auto-configuration sequence. The external pull-up resistor is required when the external EEPROM is required by the system design.

It is suggested to pre-program the EEPROM device before assembly during manufacturing. The EEPROM in the XR17C15x card must be pre-programmed with the proper vendor and sub-vendor identifications, and vendor and sub-vendor device numbers for the PC system to boot up correctly during the auto-configuration sequence. Otherwise, the PC system will not recognize the card.

Notice that the XR17C15x device will write to the external EEPROM even when EECS pin was detected with a logic "0" during power up. So here is a more complex method to store your ID values to the external EEPROM. It enables you to program the EEPROM via the PCI Bus. You will need to pre-install both, the Exar's and your own, device drivers to the PC system and provide a jumper option (J1) for the pull-up resistor on pin EECS as shown in Figure 1 on previous page. The procedure goes like this. With J1 out, no pull-up resistor in the circuit, power up the system to boot up with the Exar ID values, run an utility program to store your ID values to the external EEPROM via the PCI bus, install J1, the pull-up resistor on pin EECS in the circuit, and reboot the PC system. The system will now boot up with your ID values for your own device driver.

#### 4.0 SOFTWARE ACCESS AND CONTROL

The XR17C15x provides the REGB register (offset 0x8E from base address in the PCI Base Address Register) to control the four pins described above. REGB register is a 32-bit register but it uses only 4 bits to offer this function. Bit-20 is EECK, bit-21 is EECS, bit-22 is EEDI and bit-23 is EEDO. Each bit has direct control over its associated I/O pin.

According the 93C46 data sheet, there are three data operation sequences; read instruction timing, write instruction timing, and erase instruction timing. The duration of high/low signal level of EECK pin determines the serial read/write rate. Do not go over the maximum clock rate of the 93C46 to avoid read/write operation failures. The XR17C15x reads data from the EEPROM during power up at around 128.9KHz when the PCI bus clock is 33MHz or the PCI clock frequency divided by 256. The programming routine for operating the EEPROM must meet the timing requirement of the EEPROM device as published in its datasheet. Each EEPROM's vendor may have different specifications so check the specifications over before making any substitution.

#### 5.0 DATA FORMAT AND STORAGE

The interface uses a 16-bit data format (Word). The very first four words in EEPROM will be read by XR17C15x during power up and loaded in the PCI configuration space. They are as follows:

**TABLE 1: EEPROM DATA DEFINITIONS**

EEPROM Memory Word	Configuration Register	Configuration Register Offset
1st word	Vendor ID	Index 2
2nd word	Vendor Device ID	Index 0
3rd word	SubSystem Vendor ID	Index 2E
4th word	SubSystem Device ID	Index 2C

Note also that words 5 through 64 are not used by the PCI configuration. Customer's software driver can use this data area for own use.

**TABLE 2: EEPROM ADDRESS DEFINITIONS**

EEPROM MEMORY ADDRESS	WORD (32-BIT) DATA FORMAT	
	Bit-31 to Bit-16	Bit-15 to Bit-0
0x00	Device ID	Vendor ID
0x02	Subsystem Device ID	Subsystem Vendor ID

## **6.0 EXAR WINDOWS NT4.0 APPLICATION PROGRAM**

Exar offers a Windows NT4.0 application software that provides a graphic user interface (GUI) to assist customers to modify the Vendor ID, Vendor Device Model Number, SubSystem Vendor ID and SubSystem Model Number in the EEPROM with your own identifications.

The application software works with a kernel device driver under Windows NT4.0 operating system. The operating system does not allow user's application software to have direct communication with the hardware in the PCI slots. Thus, a kernel device driver is needed to bridge the hardware and the application. The application opens the driver device and requests the PCI card base address. Once the application gets the device base address, it could access the internal registers in the XR17C15x. Through the REGB register, the application can directly control the four input/output signal lines of XR17C15x device to the EEPROM as described earlier.

The kernel device driver does the lowest level tasks in kernel mode in order to allow the application software to contact the PCI card. First, the device driver under the operating system's assistance enumerates every PCI slots to find the right PCI card; second, it initializes the XR17C15x device in PCI card; third, it allocates the resources in the PC system and creates a device symbolic name which will be called by the application. Finally, it maps the hardware physical address into user mode linear address that will be used by higher level applications in user mode.

## **7.0 EMBEDDED APPLICATION**

In embedded applications, there is no need for having a device driver to link the hardware and users application alike with Windows O/S. As long as the host CPU has a direct access to the XR17C15x internal registers. The user can use the 4 Input/output pins for control and store data proprietary data to the EEPROM.

/DAN112/

### **NOTICE**

EXAR Corporation reserves the right to make changes to the products contained in this publication in order to improve design, performance or reliability. EXAR Corporation assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representation that the circuits are free of patent infringement. Charts and schedules contained here in are only for illustration purposes and may vary depending upon a user's specific application. While the information in this publication has been carefully checked; no responsibility, however, is assumed for inaccuracies.

EXAR Corporation does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless EXAR Corporation receives, in writing, assurances to its satisfaction that: (a) the risk of injury or damage has been minimized; (b) the user assumes all such risks; (c) potential liability of EXAR Corporation is adequately protected under the circumstances.

Copyright 2000 EXAR Corporation

Datasheet March 2001

Reproduction, in part or whole, without the prior written consent of EXAR Corporation is prohibited.

---