EXAR PCI UART- EEPROM INTERFACE

Author: **BL**

1.0 INTRODUCTION

The Exar's PCI family of UARTs provides an interface to an Electrically Erasable Programmable Read Only Memory (EEPROM). The EEPROM must be 93C46-like, with its memory configured as 16-bit words. This interface is provided in order to program the registers in the PCI Configuration Space of the PCI UART during power-up. This is essential for the operating system to correctly identify the device and load the appropriate device driver. The following table lists all the UARTs that have this EEPROM interface and are covered by this application note. In this application note, the term 'PCI UART' applies to any of the UARTs listed in the table below.

TABLE 1: EXAR PCI UARTS WITH EEPROM (93C46) INTERFACE

PCI FAMILY	UARTS
5V PCI ('C' Family)	XR17C158, XR17C154, XR17C152
3.3V PCI ('L' Family)	XR17L154, XR17L152
Universal PCI ('D' Family)	XR17D158, XR17D154, XR17D152

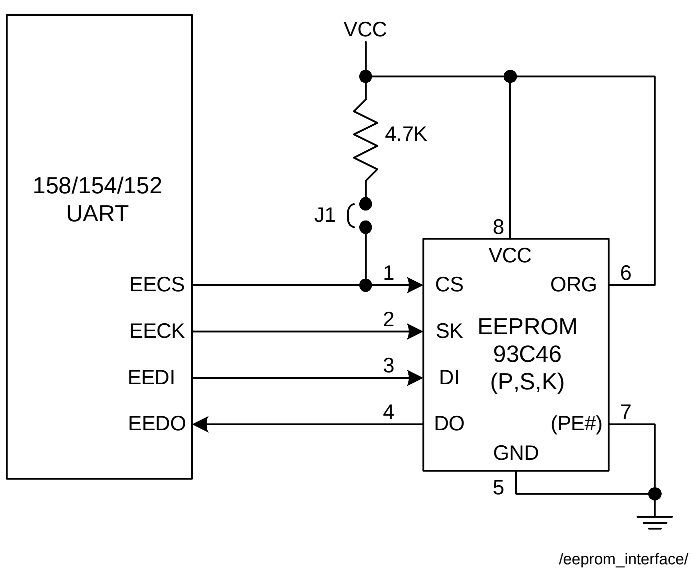
2.0 HARDWARE INTERFACE

The Exar PCI UART works with a 93C46-type EEPROM whose memory is configured as 16-bit words. The UART reads only the first four words (16-bit registers) of the EEPROM. Exar selected the 93C46 for its small size and there is no limitation on the size of the EEPROM used. The 93C46 is an industry standard part and is offered by Catalyst, Exel, Fairchild, Microchip, Xicor and other vendors. The 93C46 is a 1K-bit memory device that can be configured as 64 x 16 (sixty four 16-bit words) or 128 x 8 (128 bytes). In order to work with Exar PCI UARTs, it needs to be configured as 64 x 16. The UART interfaces to the EEPROM through 4 input/output pins:

- EECS (output) is the EEPROM chip select signal (pin 115 of the 154/158; pin 83 of the 152)
- EECK (output) is the EEPROM clock (pin 116 of the 154/158; pin 84 of the 152)
- EEDI (output) is the serial data input to the EEPROM (pin 114 of the 154/158; pin 82 of the 152)
- EEDO (input) is serial data output from the EEPROM (pin 113 of the 154/158; pin 81 of the 152)

The figure below shows the connections.

FIGURE 1. PCI UART CONNECTIONS TO EEPROM



3.0 PCI CONFIGURATION

The system in which the PCI UART resides, configures the UART during power-up based on the information provided by the UART in its PCI Configuration Registers. Only four 16-bit words are programmable via the EEPROM interface. These are given in Table 2 below. During power up of the PCI UART, on the rising edge of the active-LOW reset pin (RST#), the state of the EECS pin is sampled.

- A LOW (logic 0) on the EECS pin means no EEPROM is present => Exar's default values will be retained.
- A HIGH (logic 1) on the EECS pin means an EEPROM is present => values from the EEPROM will be loaded (see Table 3).

The PCI UART has a weak internal pull down resistor making the default to "no EEPROM". The system designer must provide an external pull-up resistor (4.7kΩ recommended) to enable the reading of external EEPROM memory values. If EECS pin is sampled HIGH, the PCI UART reads four word data (described below) from the EEPROM's first four word locations and places those values into the PCI configuration space registers for chipset identification. The PCI UART reads data from the EEPROM using a clock (EECK) rate of roughly 128.9KHz when the PCI bus clock is 33MHz (= the PCI clock frequency divided by 256). It is the responsibility of the system design engineer to ensure that the selected EEPROM device supports this clock rate.

FIGURE 2. PCI CONFIGURATION DURING POWER-UP

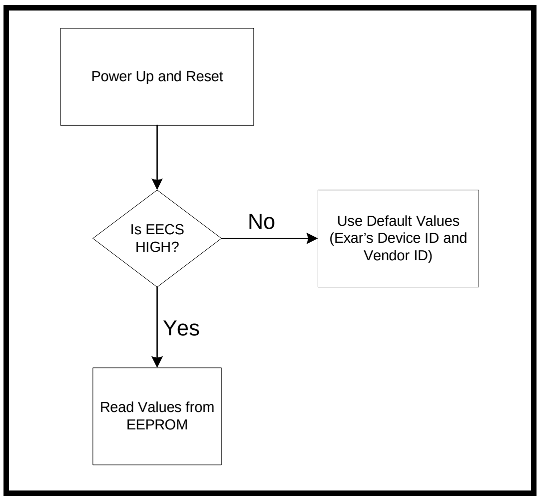


TABLE 2: PCI CONFIGURATION REGISTERS

Configuration Register	PCI Configuration 16-bit Register Address	Exar's Default Values
Vendor ID	0x00	0x13A8
Device ID	0x02	0x0158, 0x0154 or 0x0152
SubSystem Vendor ID	0x2C	0x0000
SubSystem ID	0x2E	0x0000

TABLE 3: EEPROM ADDRESS DEFINITIONS

EEPROM MEMORY ADDRESS	EEPROM DATA [D15:D0]
0x00	Vendor ID
0x01	Device ID
0x02	Subsystem Vendor ID
0x03	Subsystem ID

4.0 PROGRAMMING THE EEPROM

The 93C46 EEPROM can be programmed using a stand-alone device programmer with the required values in its first four word locations. The EEPROM can also be programmed through the PCI UART interface. The PCI UART provides a 32-bit register, REGB (offset 0x8C from base address in the PCI Base Address Register), which can be used to write to/read from the EEPROM. Only 4 bits [23:20] of the REGB register are used and are shown below:

TABLE 4: REGB REGISTER BITS DEFINITION

REGB[20] (Write-Only)	Control the EECK, clock input of the EEPROM.
REGB[21] (Write-Only)	Control the EECS, chipselect of the EEPROM device.
REGB[22] (Write-Only)	EEDI data input (for the UART). Write data to the EEPROM device.
REGB[23] (Read-Only)	EEDO data output (for the UART). Read data from the EEPROM device.

The programming routine for operating the EEPROM must meet the timing requirement of the EEPROM device as published in its datasheet. Please refer to the datasheet of the specific EEPROM device used in order to obtain the correct timing specification.

5.0 EXAR WINDOWS 2000/XP APPLICATION PROGRAM

Exar offers a Windows 2000/XP application software that provides a graphic user interface (GUI) to assist customers to modify the Vendor ID, Device ID, SubSystem Vendor ID and SubSystem ID in the EEPROM with your own identifications. The device driver provided by Exar must be loaded before this application can be used. The following steps describe this procedure:

- Remove the pull-up resistor, if any, on the EECS pin of the EEPROM. Or, you can remove the jumper J1 shown in Figure 1.
- Shut down the Win 2000/XP system. Install the PCI Card and boot up the system. On power-up, the system BIOS will use the Exar’s default values for Device ID, Vendor ID, Subsystem ID and Subsystem Vendor ID.
- Load Exar’s XR17C15x driver when prompted by the operating system. This usually involves just selecting the path where the driver files are located.
- Start the EEPROM Application and modify the Device, Vendor, Subsystem and Subsystem Vendor ID’s as desired
- Power down the system and install a pull-up resistor on the EECS pin (or install the jumper J1).
- Upon power-up, the values from the EEPROM will be loaded into the PCI Configuration register of the PCI UART.
- Now, your own device driver based on the new Device and Vendor ID’s can be loaded for the PCI UART.

6.0 EMBEDDED SYSTEMS

In embedded system applications, there is no need for having a device driver to link the hardware and users application. As long as the host CPU has a direct access to the PCI UART internal registers, the user can use the REGB register bits to control the 4 Input/output pins of the EEPROM. Please follow the instructions given in the 93C46 datasheet for the read/write cycle timing for writing custom values into the first four words of the EEPROM.

REVISION HISTORY

DATE	DESCRIPTION
July 2002	Initial Release
June 2003	Updated Device List. Updated EEPROM Utility information (Win 2K and XP).

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.

Datasheet June 2003