PCIe-Mini-ESCC

2- Channels
Serial Communication Controller

PCIexpress Mini

927-10-000-4000

REFERENCE MANUAL

Revision 1.0
August 2020

ALPHI TECHNOLOGY CORPORATION
1898 E. Southern Ave
Tempe, AZ 85282 USA
Tel : (480) 838-2428
Fax: (480) 838-4477

NOTICE

1 Manual update

Release Rev.10 8/4/2020

2 GENERAL DESCRIPTION

2.1 INTRODUCTION

The PCIe-Mini-ESCC module provides an ESCC Z85C30 with 2 serial channels, and the appropriate buffers to use it as RS-232 or RS-422.

The PCIe-Mini form factor provides easy installation.

The PCIe-Mini-ESCC is installed with the following resources:

- RS-232, RS-422, RS-485 compliant
- On board line driver and receiver directly to provide the appropriate voltage levels
- 2048-byte receive and transmit FIFOs for each channel
- Multiple clocks available to fulfill the user's requirements
- Extended temperature ranges

2.2 FUNCTIONAL DESCRIPTION

A functional block diagram of the PCIexpress Mini module is depicted below in Figure 1. The PCIe-Mini-ESCC is designed around the Zilog Z8530 that is used to provide a large selection of asynchronous and synchronous protocols..

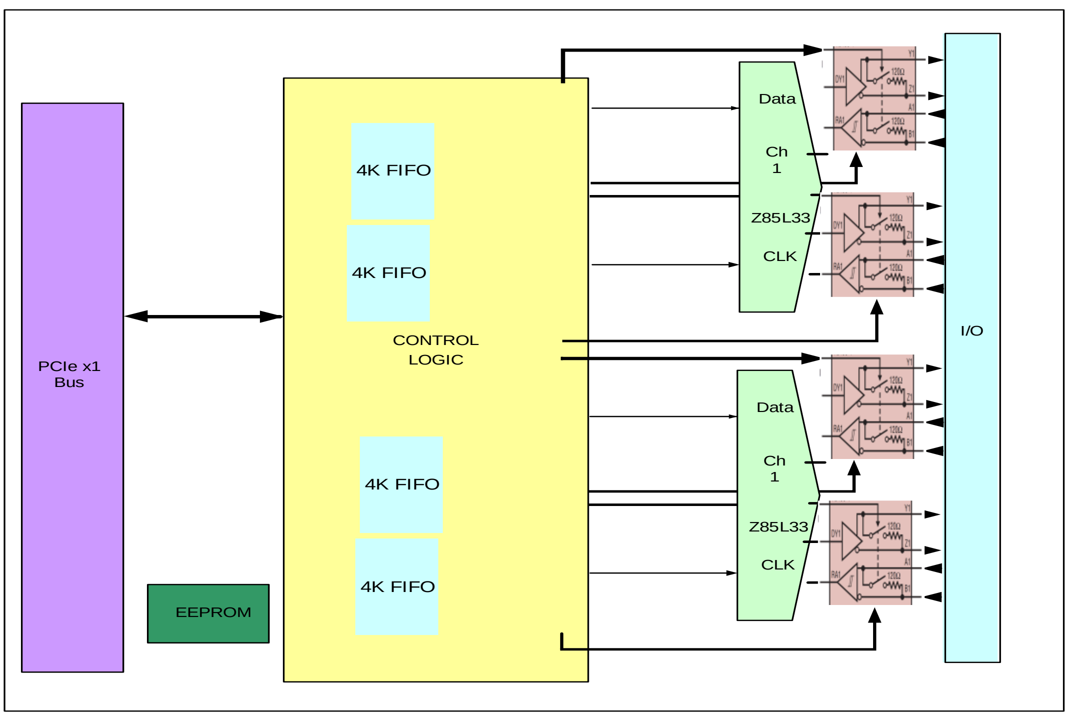

Figure 1.0: PCIe-Mini-ESCC **Block Diagram**

2.3 REFERENCE MATERIALS LIST

The reader should refer to the “Z85C30 Data sheet”, from Zilog , that provides detailed descriptions about the Z85C30 registers.

http://www.zilog.com/docs/serial/z85c30.pdf

WWW Home Page:
http://www.Zilog.com

The reader should refer to the PCIe Local Bus Specification for a detailed explanation of the PCIe bus architecture and timing requirements.

3 HOST (PCIe-Mini) SIDE

3.1 Interface to HOST (PCIe-Mini)

All PCIe-Mini) devices contain a set of registers in Configuration Space which allow for determining the manufacturer and model of the device, determining the resources necessary for the correct operation of the device, and other configuration information. Configuration space is decoded on a per slot / device basis via a mechanism described in the Mini-PCI specification.

All PCIe-Mini devices can be relocated in physical memory by means of several Base Address Registers. These registers contain the high address bits, which must be matched for any access to the card to be successful. Part of the protocol of programming the Base Address Registers allows for the transfer of information regarding the size of the regions.

The actual Base Address Registers are located in Configuration Space and are initialized by the BIOS or the OS during startup. One of the primary functions of the device driver under Windows NT/XP/WIN7 is to map these resources to the user application.

The card is actually accessed through the decoded base address registers.

3.2 PCIe-Mini Configuration Space

PCI Address: *CONFIG:0x00 – 0x3C*
Mode of Access: *Read/Write*
Reset By *Mini-PCI Hardware Reset*

The card has the following registers available to PCIe-Mini Configuration Space. They are implemented in the FPGA chip.

Offset Into PCI CFG	31 – 24	23 – 16	15 – 8	7 – 0
0x00	Device ID		Vendor ID	
0x04	Status		Command	
0x08	Class Code			Revision ID
0x0C	BIST	Header Type	PCI Latency Timer	Cache Line Size
0x10	PCI Base Address 0 (Memory Access to PLX Registers)			
0x14	PCI Base Address 1 (I/O Access to PLX Registers)			
0x18	PCI Base Address 2 (Memory Access to DSP SRAM and card registers)			
0x1C	PCI Base Address 3 (Not Used for this card)			
0x20	Unused PCI Base Address 4			
0x24	Unused PCI Base Address 5			
0x28	Cardbus CIS Pointer (Not Supported)			
0x2C	Subsystem ID		Subsystem Vendor ID	
0x30	PCI Base Address for Expansion ROM			
0x34	Reserved			
0x38	Reserved			
0x3C	Max Latency	Min Grant	Interrupt Pin	Interrupt Line

Table 2.1: PCIe-Mini Configuration Space

The card presents the following initial configuration values to the PCIe-Mini system, based on the values stored in the device.

Register	Value (Meaning)
Vendor ID	0x13C5 (Alphi technology Corporation)
Device ID	0x0508 (PCIe-Mini-ESCC)
Revision ID	0x00
Class Code	0xff0000 (Device does not fit into defined class codes)
Interrupt Line	0xff
Interrupt Pin	A
Multifunction Device	No
Build In Self Test	No
Latency Timer	0x00
Minimum Grant	0x00
Maximum Latency	0x00
Base Address 0 Size	16K Bytes Allocated
Base Address 1 Size	Not used
Base Address 2 Size	4K Bytes Allocated
Base Address 3 Size	No used
Expansion ROM Size	None

Table 2.2: PCIe-Mini-ARIC429 Default Configuration

3.3 PCIe-Mini Base Address Regions

HOST Address	WIDTH USED	Description	TYPE
BAR0	16 K bytes	PCIexpress Control Register	I/O
BAR2	4 K bytes	ESCC IO Space	
BAR3	Not used		
BAR4	Not used		
BAR5	Not used		

Table 2.3: PCIe-Mini-ARIC429 Base Address Regions

4 Register Description

The SCC registers are located in BAR 2, at offsets between 0x00 and 0x37. The following table contains an example of C structure mapping the registers.

4.1 BAR2 Address Map

Address Offset	Module	R/W	Size	Description
0x0000	System ID	R	32-bit	System Identification
0x0004	FPGA Version	R		Time stamp corresponding to the FPGA version
0x0010	LED	R/W	8-bit	LED outputs
0x0020	SCC #0			Channel 1 Module
0x0080	SCC #1			Channel 2 Module
0x0800	Flash			FLASH memory

Figure 1: Bar 2 Address Map

4.2 Channel Description

Address Offset	Register	R/W	Size	Description
0x00	SCC Control Reg.	R/W	8-bit	Refer to the Z8530 manual
0x04	SCC Data Reg.	R/W	8-bit	Refer to the Z8530 manual
0x08	Esc Char Reg.	R/W	8-bit	Used in bisync escape character stripping
0x0c	RxFIFO Status Reg.	R/W	32-bit	FIFO usage level and reset
0x10	TxFIFO Status Reg.	R/W	32-bit	FIFO usage level and reset
0x20	RxFIFO Data out	R	32-bit	FIFO data out
0x24	RxFIFO Data Valid	R	32--bit	Which data byte is valid
0x28	TxFIFO Data in	W	32-bit	FIFO data in
0x30	I/O buffer control	R/W	32-bit	Specify serial interface protocol

Figure 2: SCC Address Map

4.3 SCC Control and Data Register

These addresses allow direct access to the SCC registers. Please refer to the SCC manual.

4.3.1 Channel Control Register

BIT	18	17	16	15	13
Signal	Rx Fifo IRQ	Tx Fifo IRQ	SCC IRQ	Reset interface	DLE/SYNC enabled

BIT	12	11	10	7-0
Signal	Tx DMA Enable	Rx DMA Enable	SCC IRQ enable	Escape Character

1. **Rx Fifo IRQ (R)**: This status bit indicates if the Rx FIFO logic is requesting an interrupt.
2. **Tx Fifo IRQ (R)**: This status bit indicates if the Tx FIFO logic is requesting an interrupt.
3. **SCC IRQ (R)**: This status bit indicates if the SCC is requesting an interrupt.
4. **Reset Interface (R/W)**: This bit allows resetting the interface. It clears the FIFO and sends a reset signal to the SCC.
5. **DLE/SYNC support (R/W)**: This bit is used to enable the Stripping Algorithm and to support the BYSYNC mode. It allows stripping some bytes from the input stream. These Bytes must be defined as the escape character register defined in bits 7-0. At reset this bit is disabled.
6. **Tx DMA Enable (R/W)**: This bit allows using the FIFO to send data bytes to the SCC. The DMA transfer is triggered by the signal /DTR//REQ. The SCC must be programmed to assert it when ready to transmit.
7. **Rx DMA Enable (R/W)**: This bit allows using the FIFO to receive data bytes from the SCC. The DMA logic uses the signal /W//REQ for transferring data from the SCC. The SCC must be programmed to assert it when received data are ready.
8. **Escape character**: This register stores the character that will be escaped by the Algorithm described below.

The Escape algorithm implemented as follows:
No strings of DLEs are allowed, only a DLE and the following character.
The only time that a character following a DLE would enter the FIFO is if it is a second DLE.
This works somewhat like the '\' character in C string literals.
Example:

"A, DLE, DLE, DLE, DLE, B, C" enters the FIFO as "A, DLE, DLE, B, C".

In the FPGA the register is implemented to set the DLE character (both DLE and SYNC would be set on the SCC for transmission).
The algorithm below is implemented.

```
escape = false;
do {
```



```
  c = nextChar();
  if (c == DLE ^^ escape) {
    escape = !escape;
  } else {
    sendToFifo(c);
    escape = false;
  }
}
```

Note: Any character can be Escaped.

4.3.2 I/O Buffer Control Register

BIT	9	8	7	6
Signal	Clk Source	DCD disable	Driver Enable 0	Driver Enable 1

BIT	5	4	3	2	1	0
Signal	RS485 termination	Loop Back	Fast Mode	RS485/ RS232	Half duplex	Receiver enable

Clock source: When ‘1’, the ESCC tx/rx clock input is fed by the local 14.7456 MHz oscillator. When ‘0’, it uses the clock signal from the serial interface.
DCD disable: When ‘1’, the DCD input of the ESCC is set to ‘0’, always asserting it. When ‘0’ the DCD signal from the serial interface is sent to the ESCC.
Driver Enable: The output drivers can be always enabled, always disabled, or enabled by the RTS signal:
- **00**: Transmitter is disabled.
- **01**: Transmitter is enabled.
- **02**: Transmitter is enabled when the SCC’s RTS pin is low.
- **03**: Transmitter is enabled when the SCC’s RTS pin is high.

RS485 termination: When ‘1’, the terminations are enabled. It enables a 120Ω resistor between the 2 differential lines.
Receiver enable: When ‘0’, the receiver is enabled.
Half Duplex: When ‘0’, the buffers are in half-duplex mode.
RS485/RS232: A “1” enables the RS485 transceivers.
Fast: A “1”selects tells the transceivers power supply to stay on, independently of the state of the driver, enabling a faster turn on/off at the expense of a slightly higher power consumption.
Loop Back: A “1” selects loop-back diagnostic mode. The output pins connected to the input pins before the transceivers.

Upon reset, the register contains all ‘0’.

4.3.3 Transmit FIFO Control Register

Each register allows a separate condition which the FIFO can use to generate an interrupt. See chart below for configuration. The bits are enabled for interrupt when set (1). However some status bits request interrupts when high and some low. So, for instance the receive FIFO can be program to request an interrupt when NOT empty (Receive FIFO Empty is 0) or when the receive FIFO is full (Receive FIFO Full is 1).

BIT	31	27	26	25	24
Signal	FIFO Clear	TxFIFO Empty IRQ enable	TxFIFO Almost Empty IRQ enable	TxFIFO Almost Full IRQ enable	TxFIFO Full IRQ enable
IRQ when		true	true	false	false

BIT	19	18	17	16	15:0
Signal	TxFIFO Empty	TxFIFO Almost Empty	TxFIFO Almost Full	TxFIFO Full	FIFO level

FIFO Clear: Resetting the FIFO is made by toggling the correspondent bit from”0” to 1” then “0”, will reset the corresponding channel.
FIFO enabled: When the transmit FIFO is enabled, the signal /DTR//REQ is for transferring data to the SCC. The SCC must be programmed to assert it when ready to transmit.
SCC IRQ enable: A “1” allows the SCC Z85233 interrupt to assert the VMEbus interrupt request lines.
TxFIFO empty: When set to “1”, the board requests a VME interrupt when the FIFO is empty.
TxFIFO almost empty: When set to “1”, the board requests a VME interrupt when the FIFO contains less than 256 bytes
TxFIFO almost full: : When set to “1”, the board requests a VME interrupt when the FIFO contains less than 1792 bytes (256 bytes of space still available).
TxFIFO full: When set to “1”, the board requests a VME interrupt when the FIFO is not full.

4.3.4 Receive FIFO Control Register

BIT	31	27	26	25	24
Signal	FIFO Clear	RxFIFO Empty IRQ enable	RxFIFO Almost Empty IRQ enable	RxFIFO Almost Full IRQ enable	RxFIFO Full IRQ enable
IRQ when		false	false	true	true

BIT	19	18	17	16	15:0
Signal	RxFIFO Empty	RxFIFO Almost Empty	RxFIFO Almost Full	RxFIFO Full	FIFO level

FIFO Reset: Resetting the FIFO is made by toggling the correspondent bit from”0” to 1” then “0”, will reset the corresponding channel.
RxFIFO empty: When set to “1”, the board requests a VME interrupt when the FIFO is not empty.
RxFIFO almost empty: When set to “1”, the board requests a VME interrupt when the FIFO contains more than 256 bytes.
RxFIFO almost full: : When set to “1”, the board requests a VME interrupt when the FIFO contains more than approximately 1792 bytes (256 bytes of space still available).
RxFIFO full: : When set to “1”, the board requests a VME interrupt when the FIFO is full.

Please note that there is a total of up to 4 bytes in buffers after the FIFO. So, for instance, the FIFO might show as empty while there are few characters waiting. The bits 16 18 of the Receive FIFO adapter status indicates how many bytes, if any, are available. In general, if the FIFO is not empty the adapter buffer will contains 4 extra bytes.

4.3.5 Receive FIFO Width Adapter Status

BIT	18:16	2:0
Sign al	Number of bytes available	Valid bytes in last transfer

To speed up the data transfer from the serial converter across the VME, the incoming characters are compiled into a 32-bit word. This register indicates the state of the 8 to 32 bit data adapter.

- Number of bytes available: This indicates how many bytes are already in the converter.
- Valid bytes in last transfer: Because bytes arrive asynchronously, between the last check for the number of bytes and the actual read operation, the byte count might have

increased. So, if there wasn't already 4 bytes available in the register, one must check this value to see how many bytes are valid in the 32-bit.

4.4 Baud rate generator

The BRG has 3 possible source of Clock.

On board oscillator is 14.7456 MHz

Source	Description
PCLK	PCLK = 16 MHz
RTXC	RTXC can be Driven External in RS-485 Mode
RTXC	RTXC can be Driven by on Board Oscillator of 14.7456MHZ

The Z85233 provides a baud rate generator for each channel, consisting of one16-bit time-constant register, a 16-bit down-counter, and a flip-flop on the output that makes the output a square wave. The down-counter clock is divided by the clock mode value, and the clock mode source allows selecting the master clock between the system clock PCLK and the input pin RTxC

4.5 Hardware information

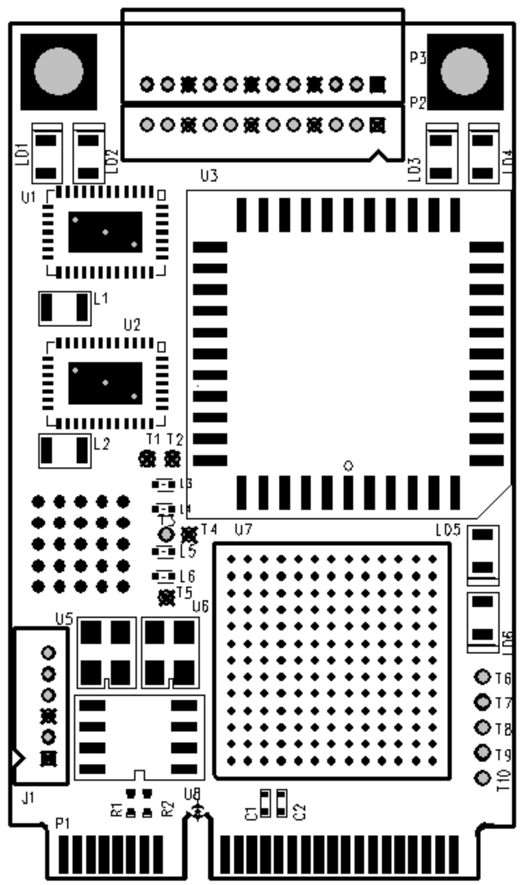

Figure 2.3: **Connector LOCATION**

J1 is Factory use only

4.6 Jumpers Description

None

4.7 LED Description

Six Led are located on the board that can be used for debugging purpose.

LED #	Description
1	CH 1 Transmit
2	CH 1 Receive
3	CH 0 Transmit
4	CH 0 Receive
5	User LED 1
6	User LED 2

Figure 2: LED Description

4.8 Connectors Description

4.8.1 Connector Model

Two 12-pin connectors are used to route the ESCC signals off the card.

The Connectors are manufactured by Molex
.

Use	Model
On PC Board	53048-1210
Suggested Mate	51021-1200
Suggested Female Terminal	50058-8100

Table 4.2: Connector Model Numbers

4.8.2 External I/O Connector P3

The SCC # 0 signals are routed as follow for PCIe-Mini-ESCC

Pin	Connection	LTC2872	RS232	RS422
1	GND			
2	TXD0+	Y1	TxD	TxD+
3	TXD0-/nRTS0	Z1	nRTS	TxD-
4	GND			
5	RXD0+	A1	RxD	RxD+
6	RXD0-/nCTS0	B1	nCTS	RxD-
7	GND			
8	TXC0+	Y2	TxC	TxC+
9	TXC0-/DTR0	Z2	nDTR	TxC-
10	GND			
11	RXC0+	A2	RxC	RxC+
12	RXC0-/DCD0	B2	nDCD	RxC-

Table 4.1: P3 external I/O connector PCIe-Mini-ESCC

4.8.3 External I/O Connector P2

The SCC # 1 signals are routed as follow for PCIe-Mini-ESCC

Pin	Connection	LTC2872	RS232	RS422
1	GND			
2	TXD1+	Y1	TxD	TxD+
3	TXD1-/nRTS1	Z1	nRTS	TxD-
4	GND			
5	RXD1+	A1	RxD	RxD+
6	RXD1-/nCTS1	B1	nCTS	RxD-
7	GND			
8	TXC1+	Y2	TxC	TxC+
9	TXC1-/DTR1	Z2	nDTR	TxC-
10	GND			
11	RXC1+	A2	RxC	RxC+
12	RXC1-/DCD1	B2	nDCD	RxC-

Table 4.1: P3 external I/O connector PCIe-Mini-ESCC

4.9 P1 PCIexpress Mini Connections

P4	Signal	P4	Signal
1	NC	2	3.3V
3	NC	4	GND
5	NC	6	1.5V
7	GND: CLKreq	8	NC
9	GND	10	NC
11	REF CLK+	12	NC
13	REF CLK+	14	NC
15	GND	16	NC
17	NC	18	GND
19	NC	20	NC
21	GND	22	PCIe reset
23	PCIe TX -	24	NC
25	PCIe TX +	26	GND
27	GND	28	1.5V
29	GND	30	NC
31	PCIe RX -	32	NC
33	PCIe RX +	34	GND
35	GND	36	NC
37	GND	38	NC
39	3.3V	40	GND
41	3.3V	42	NC
43	GND	44	NC
45	NC	46	NC
47	NC	48	1.5V
49	NC	50	GND
51	NC	52	3.3V

NC: No Connection to the board

Table 3.4: PCIe-Mini Connections