National Radio Astronomy Observatory
PO Box O
Socorro, NM 87801

Title: Module Interface Board Hardware Definition
Document: A23710N0001 **Date:** 2004-10-08
Type: *Hardware Definition* **Revision: D**

Authors: W. M. Koski, George Peck

EVLA Monitor and Control Group

Module Interface Board Hardware Definition

W. M. Koski wkoski@nrao.edu
G. Peck gpeck@nrao.edu

Keywords:	
Author Signatures:	Date:
Approved by:	Date:
Released by:	Date:

Change Record

Revision	Date	Author	Sections Affected	Remarks
None	8/1/2002	W. Koski, G. Peck	All	Initial Document
D	10/8/2004	W. Koski, G. Peck	All	Revised Document to MIB Board Revision D

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figures:

Tables:

1 Introduction

This document describes the Module Interface Board (MIB) for the Expanded Very Large Array (EVLA) project.

The MIB has been developed to serve the needs of Device Designers who require varying degrees of complexity in local hardware monitoring and control with known, but flexible interface methods.

In this report, the word “device” refers to the electronics that the MIB is monitoring and controlling. A single MIB could monitor and control one or more devices. In most cases, the MIB will be connected directly to one of the devices that it is monitoring and controlling via its connectors.

The MIB has two primary functions. The first is to interface the device to the Ethernet backbone of the EVLA. The second is to perform the monitor and control tasks that are required by the device. In some cases, the device may have its own microprocessor, in addition to the MIB, to perform some of the monitor and control interface tasks.

The MIB provides a powerful 32-bit micro-controller with firmware that facilitates communication between the MIB and the device that it controls. Communication between the antenna control computer and the MIB is via fiber optics using the IEEE 802.3 standard commonly called Ethernet. The particular IEEE 802.3 sub-standard used is known as 100BASE-FX.

The MIB normally is the main micro-controller that carries out most device tasks. Between the MIB and device are various signal lines that provide serial communications, thirty-two general-purpose I/O lines, eight interrupts, and other miscellaneous signals. These lines provide the device designer some degree of flexibility, but also standardization.

The main area of deployment for the MIB will be for devices located at antennas or that are antenna related. Due to differing requirements, the Correlator design team will develop a different MIB for the Correlator called the Commercial MIB (CMIB). A separate document from this one will be developed for the CMIB.

The structure of this document is as follows. Section 2 describes the general specifications of the MIB. In Section 3, the MIB connections are listed in tables for each connector. Firmware considerations are discussed in Sections 4 through 9. The theories of operations or implementation details are given in Section 10. Section 11 lists References. Section 12 contains the list of MIB drawings.

2 General Specifications

2.1 Background

The MIB provides the basic need to act as a transceiver for messages on the Ethernet, provides monitor and control duties, and performs device tasks. Much of the functionality of the MIB reflects the micro-controller selected to operate it. In this design an Infineon Tricore TC11IB micro-controller was selected. Table 1 lists the basic functionality for the TC11IB as well as the MIB.

Table 1: Basic Functionality for the MIB

Function	**MIB**
External Parallel Memory	8M/16M Bytes (4M/8M x 16 Bits) Parallel Flash
External Serial Memory	4K Bits Serial Flash
Internal RAM	1.5M Bytes (TC11IB)
External Oscillators	12 MHz – TC11IB, 25MHz – LXT971A
Operating Temperature Range	0° C - 70° C
Watchdog Timer	1
Internal Timers	7
Ethernet Ports	1: Media Independent Interface (MII) to LXT971A
Synchronous Serial Ports	2: One Serial Peripheral Interface (SPI), One Programmable
Asynchronous Serial Ports	2: One RS232 Modem, One Programmable
General Purpose I/O Lines	1: 32 Bits Total – Programmable by Bytes
Interrupts	9: Two used by the MIB, Eight External
Resets	4: One Power-On, One for all MIB's, 2 External Outputs
Connectors	12
Power	5 Volts ± 250mV @ 1A Maximum

The MIB Block Diagram (Figure 1) helps to show the basic use for the MIB in a Device Designer's system. The left hand side shows the signals and connections that are used by the MIB to support the TC11IB and communications over Ethernet. The right hand side shows the signals and connections that can be used by the Device Designer. There are a total of fourteen connectors. Certain signals are available on more than one connector. Section 2.3 will explain their general use to the Device Designer.

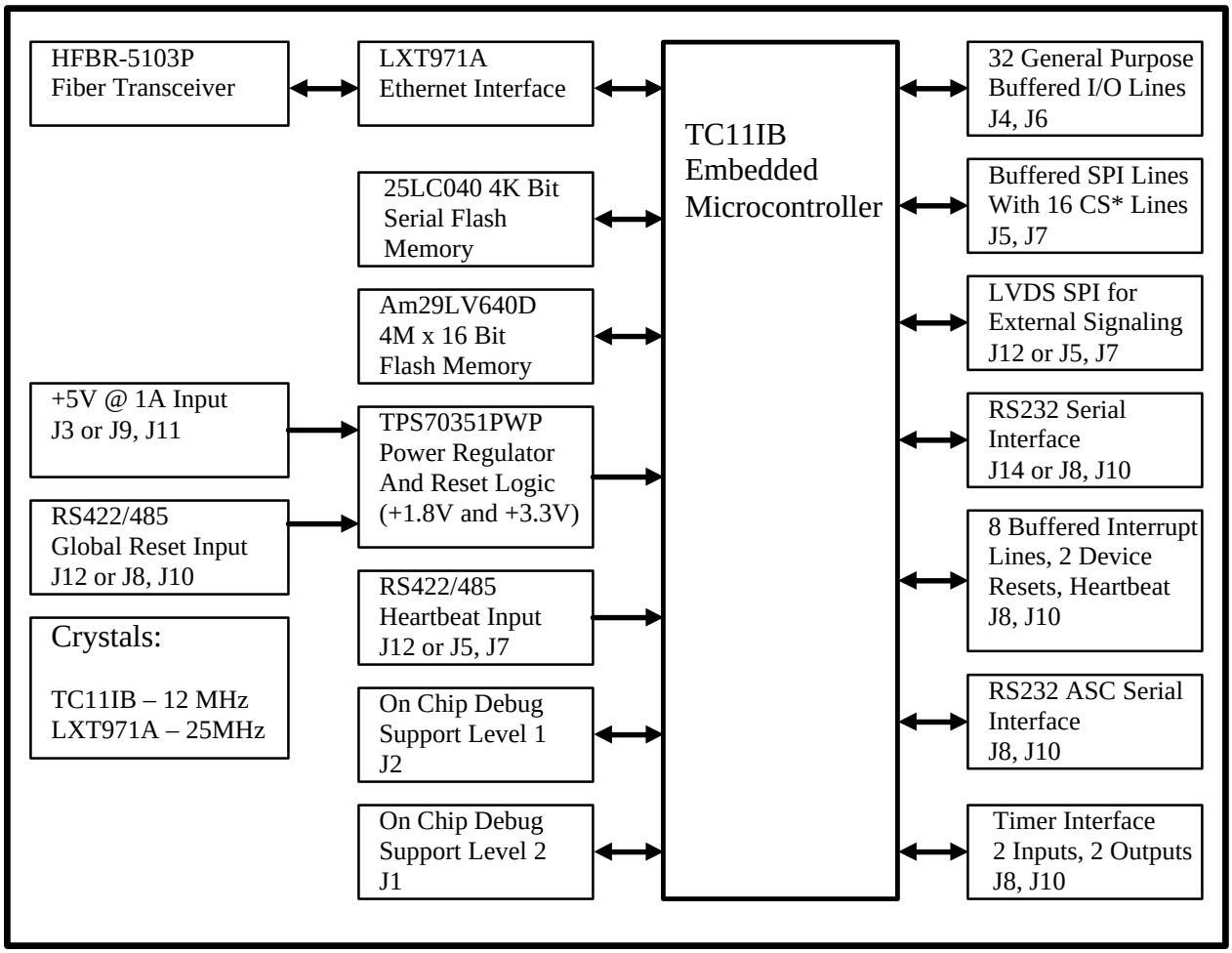

Figure 1: MIB Block Diagram

2.2 Mechanical Details

The MIB uses a 14-layer surface-mount printed circuit board measuring 6.0 in. by 4.5 in. by 0.08 in. thick. The Drawing shown in Figure 2: MIB Mechanical and Connector Pin Location Drawing C23710M0001 shows mechanical details that the Device Designer needs. From this drawing, the Device Designer can determine optional mounting hole placement as well as placement of the connectors. The height dimensions assume the usage of a Samtec FTSH series male header and a Samtec FLE series socket for stacking the MIB and device PCB's together. The mounting holes allow for #4-40 style mounting hardware. . The MIB requires a printed circuit board area of 6 in. by 4.5 in. The stacked height distance between the MIB printed circuit board and the device printed circuit board that it is mounted on is 0.3 in. The total height of the MIB is approximately 0.7 in.

2.3 Connectors

The MIB has twelve connectors, denoted J1 through J12. The Device Designer needs to take note of only J3 through J12. Most device designers may only need to contend with a subset of these connectors. Table 2: Connector Functions and Mating Connectors, describes each connector, its primary purpose, and the proper mating connector to be used by the Device Designer.

Table 2: Connector Functions and Mating Connectors

Connector	Function	Mating Connectors
J1	On Chip Debug Service Level 2	Samtec QTH-030-01-L-D-A-K
J2	On Chip Debug Service Level 1	Samtec HCSD-08-S-02-01-S-N-ST (Note 1)
J3	+5V Input Power	Hirose DF7-4S-3.96C
J4	General Purpose I/O Lines	Samtec FLE-125-01-G-DV-A (Note 2)
J5	SPI Signal Lines	Samtec FLE-125-01-G-DV-A (Note 2)
J6	General Purpose I/O Lines	Samtec FTSH-125-02-L-DV-A
J7	SPI Signal Lines	Samtec FTSH-125-02-L-DV-A
J8	Interrupts, Resets, Serial Lines	Samtec FLE-125-01-G-DV-A (Note 2)
J9	+5V Input Power	Samtec FLE-113-01-G-DV-A
J10	Interrupts, Resets, Serial Lines	Samtec FTSH-125-02-L-DV-A
J11	+5V Input Power	Samtec FTSH-113-02-L-DV-A
J12	Required MIB Signals	Samtec FFSD-13-S-02-N (Note 1)
Note 1	Ribbon Cable General Specs.	Cable Length 2 in, One End Terminated
Note 2	Ribbon Cable Replacement	Samtec FFSD-25-S-02-N (Note 1)

Before the discussion continues further, it must be stated that there are three methods for the MIB to be connected to the device. They are MIB as daughter card, device as daughter card, and MIB as stand-alone. Each method assumes that the MIB has the TC11IB component on top. For the MIB as daughter card case, the MIB would connect to the device via connectors J6, J7, J10, and J11. For the device as daughter card case, the device would connect to the MIB via connectors J4, J5, J8, and J9. For the MIB as stand-alone case, the MIB would connect to the device via ribbon cable using connectors J4, J5, and J8. Combinations of the above can also exist as well. A device designer can easily make an electronic Dagwood sandwich using the MIB.

The ribbon cable, described in Note 2 of Table 2, can be changed to a double ended connector by replacing the “–S” with a “-D”. This would allow an FTSH-125-02-L-DV-A to be mounted on the device PCB and the ribbon cable could then be plugged into it. Please see the Samtec FFSD series fact sheet for further ribbon cable options and mechanical information.

2.3.1 Connectors J1, J2:

Connectors J1 and J2 are meant to be connected to software emulators, analyzers, etc. At present, no usage by the Device Designer for their devices is seen.

Please see Section 3, Table 3 for J1 Pin Assignments and Section 3, Table 4 for J2 pin assignments.

2.3.2 Connectors J3, J9, J11:

Connector J3 is where the main +5V input power should be applied in most designs. However, if the MIB is a daughter card to the device PCB, then the +5V input power can be supplied via J11. For the device as daughter card, the +5V input power can be supplied via J9. For the stand-alone case, power would be routed to each board separately, and to J3 specifically for the MIB.

Please see Section 3, Table 5 for J3 Pin Assignments and Section 3, Table 9 for J9 and J11 Pin Assignments.

2.3.3 Connectors J4, J6:

Connectors J4 and J6 provide the thirty-two General Purpose I/0 Lines to the Device. They have been broken into four groupings called Bank 0, Bank 1, Bank 2, and Bank 3. Each Bank as a whole can be designated as Input, Output, or Bi-directional. If the device designer uses Bi-directional, it is highly recommended that Bank 0 is used as the first Bi-directional group, and that Bank 3 provide the directional control signals necessary to carry out the Bi-directional Option. This leaves Bank 1 and Bank 2 to further widen the Bi-directional needs of the device, if necessary.

It is also recommended that the device designer carefully group each Bank with similar signals when possible. Having as many related signal lines together as possible eases software design.

Please see Section 3, Table 6 for J4 and J6 Pin Assignments**.**

2.3.4 Connectors J5, J7:

Connectors J5 and J7 provide the Serial Peripheral Interface (SPI) signals in and out of the device. The first thirty signals are devoted to device functions while the latter twenty signals are for external MIB needs. The MIB provides the device with sixteen low-true SPI Chip Selects. Each SPI device will be connected to one of these SPI Chip Select lines and will only operate when this line is logic low. The MIB is the SPI Master, so it provides the SPI Clock signal and the Master-Output-Slave-Input (MOSI) signal. All SPI device outputs will be connected to the Master-Input-Slave-Output (MISO) line. If a device uses more then sixteen SPI ports, then additional SPI Chip Selects must be developed. Thirty-two additional select lines can be obtained from the general purpose I/O lines, if they are available. If still more select lines are needed, then the SPI Chip Select lines would be grouped together to provide an address value, from which the device could provide individual chip select lines. Finally, if a SPI only solution were required, then SPI Chip Select 14 would provide an address to a shift register, while SPI Chip Select 15 would transfer data to or from a device function that has been addressed.

For Slot-ID, these SPI signals that go external to the module, the MOSI, MISO, and SPI Clock signals have been converted to LVDS. These differential signals, along with a LVDS signal called IPREQ*, provide the means by which any module obtains its address from the slot that it has been placed in.

The final input signal on these connectors is the differential HEARTBEAT signal that provides a hardware timing interrupt for the MIB. This has been currently set to 19.2 Hz by an external timing module that distributes such signals. For certain designs, this HEARTBEAT signal is not required and the MIB will track time it gets from the network. Finally, a timing signal other then 19.2Hz could be input to the MIB for a specialized device design.

Please see Section 3, Table 7 for J5 and J7 Pin Assignments**.**

2.3.5 Connectors J8, J10:

Connectors J8 and J10 provide for additional communication methods, notably two RS232 ports. One port is a simple transmit and receive type, while the other port is a full-featured hardware handshaking type. Additionally, eight interrupt inputs are provided on this connector. Next, two buffered input and output lines have been provided such that hardware can access a TC11IB internal timer unit. Finally, another use for these connectors is to provide reset signals both into and out of the MIB.

The first RS232 port only has transmit and receive lines, plus a ground reference. This port has been dedicated to reporting MIB status information for software programming use.

The second RS232 port includes modem signals that allow the device designer to connect to a single complex or simple RS232 device in order to carry out device communications.

Interrupts produced by the device are also on this connector. A low going edge, high going edge, or both can trigger the software interrupt. If the interrupt mode isn't used then these signals could be used as simple inputs instead.

Access to the TC11IB General Purpose Timer Unit 1 (GPTU1) has been provided. It consists of two input lines in order to have the capability for two unique external clocks that feed counter stage within GPTU1. Two buffered unique outputs can then be utilized in order to provide device hardware timing signals as well.

The RS422/485 Global Reset can be input to the MIB on this connector. Additionally the MIB can output a buffered software generated reset to the device. A buffered power-on reset is also on this connector, which also could be utilized by the device.

The final signal is the single-ended output HEARTBEAT signal, which the device can utilize to synchronize its hardware with said signal.

Please see Section 3, Table 8 for J8 and J10 Pin Assignments**.**

2.3.6 Connector J12:

Connector J12 provides a single connector for all external signals that the MIB requires in order to operate. They are the RS422/485 Global Reset, the LVDS HEARTBEAT, and the LVDS SPI lines and IPREQ*.

Other connectors can also provide these signals, so the Device Designer can chose which connector to use for these signals as a matter of convenience.

Please see Section 3, Table 10 for J12 Pin Assignments**.**

2.4 Power

The MIB requires +5VDC ±250mV regulated power supply. The entire board, excluding external peripherals, consumes less then 1A at +5VDC.

2.5 Operating Temperature Range

All components currently used have the commercial temperature operating range specifications of 0° C through 70° C.

3 Connector Signal Name Pin Assignments

3.1 Connector Signal Naming Convention

In the following tables each signal assigned to a pin number is assigned a signal name that matches as closely as possible the name used on the MIB schematic. Furthermore, the signal name is related to the function it provides, so that a device designer can quickly understand the function from the name with practice.

All signals are considered to be High-True unless an asterisk "*" appears in the name, then that signal is a Low-True signal. If the signal name has a "+" or "-" in it, then that signal is part of a differential pair.

All signals are to be considered to be LVTTL. That is to say they are meant to connect to +3.3V TTL devices. However, the interface circuitry can accept +5V TTL logic signals, with care.

Most of the differential signals are LVDS devices. A few signals are RS422/485 devices. In the following tables, the RS422/485 cases will be noted.

The column designated I/O will have "I", "O", "I/O", or "-" to denote signal direction from the MIB's point of view. That is to say, an "I" is a signal that is an Input into the MIB.

3.2 Connector Signal Pin Assignment Tables

Table 3: J1, On-Chip Debug Service Level 2 Connector

PIN	Signal Name	I/O	Function
1	PORST*	O	Power On Reset
2	BRKPT2	O	On-Chip Debug Service Break Qualification Line 2
3	BRKIN*	I	On-Chip Debug Service Break Input
4	BRKPT1	O	On-Chip Debug Service Break Qualification Line 1
5	TRST*	I	JTAG Module Reset/Enable
6, 7, 9	No Connection	-	No Connection
8	VCC	-	3.3 Volts
10	BRKPT0	O	On-Chip Debug Service Break Qualification Line 0
11	OCDSEN*	I	On-Chip Debug Service Enable
12	STATUS4	O	On-Chip Debug Service Pipeline Status Line 4
13	TDI	I	JTAG Module Serial Data In
14	STATUS3	O	On-Chip Debug Service Pipeline Status Line 3
15	TCLK	I	JTAG Module Clock
16	STATUS2	O	On-Chip Debug Service Pipeline Status Line 2
17, 19	No Connection	-	No Connection
18	STATUS1	O	On-Chip Debug Service Pipeline Status Line 1
20	STATUS0	O	On-Chip Debug Service Pipeline Status Line 0
21-26	No Connection	-	No Connection
27	BRKOUT*	O	On-Chip Debug Service Break Output
28	TDO	O	JTAG Module Serial Data Out
29-34	No Connection	-	No Connection
35	GND (Opt)	-	Ground (Optional)
36	IND_PC7	O	On-Chip Debug Service Indirect PC Address 7
37, 39	No Connection	-	No Connection
38	IND_PC6	O	On-Chip Debug Service Indirect PC Address 6
40	IND_PC5	O	On-Chip Debug Service Indirect PC Address 5
41, 43	No Connection	-	No Connection
42	IND_PC4	O	On-Chip Debug Service Indirect PC Address 4
44	TMS	I	JTAG Module State Machine Control Input
45-47	No Connection	-	No Connection
48	IND_PC3	O	On-Chip Debug Service Indirect PC Address 3
49, 51	No Connection	-	No Connection
50	IND_PC2	O	On-Chip Debug Service Indirect PC Address 2
52	IND_PC1	O	On-Chip Debug Service Indirect PC Address 1
53, 55	No Connection	-	No Connection
54	IND_PC0	O	On-Chip Debug Service Indirect PC Address 0
56	VCC	-	3.3 Volts
57-59	No Connection	-	No Connection
60	CPUCLK	O	TC11IB Clock Out

Table 4: J2, On-Chip Debug Service Level 1 Connector

PIN	Signal	I/O	Function
1	TMS	I	JTAG Module State Machine Control Input
2	VCC	-	3.3 Volts
3	TDO	O	JTAG Module Serial Data Out
4	GND	-	Ground
5	CPUCLK	O	TC11IB Clock Out
6	GND	-	Ground
7	TDI	I	JTAG Module Serial Data In
8	PORST*	O	Power on Reset
9	TRST*	I	JTAG Module Reset/Enable
10	BRKOUT*	O	On-Chip Debug Service Break Output
11	TCLK	I	JTAG Module Clock
12	GND	-	Ground
13	BRKIN*	I	On-Chip Debug Service Break Input
14	OCDSEN*	I	On-Chip Debug Service Enable
15, 16	No Connection	-	No Connection

Table 5: J3, Main Power Input Connector

PIN	Signal	I/O	Function
1	+5VCC	-	+5 Volts
2	+5VCC	-	+5 Volts
3	GND	-	Ground
4	GND	-	Ground

Table 6: J4 And J6, General Purpose I/O Connectors (Continues)

PIN	Signal	I/O	Function
1	GND	-	Ground
2	IO0	I/O	General Purpose I/O Pin 0 (Bank 0)
3	IO1	I/O	General Purpose I/O Pin 1 (Bank 0)
4	GND	-	Ground
5	IO2	I/O	General Purpose I/O Pin 2 (Bank 0)
6	IO3	I/O	General Purpose I/O Pin 3 (Bank 0)
7	GND	-	Ground
8	IO4	I/O	General Purpose I/O Pin 4 (Bank 0)
9	IO5	I/O	General Purpose I/O Pin 5 (Bank 0)
10	GND	-	Ground
11	IO6	I/O	General Purpose I/O Pin 6 (Bank 0)
12	IO7	I/O	General Purpose I/O Pin 7 (Bank 0)
13	GND	-	Ground
14	IO8	I/O	General Purpose I/O Pin 8 (Bank 1)
15	IO9	I/O	General Purpose I/O Pin 9 (Bank 1)
16	GND	-	Ground
17	IO10	I/O	General Purpose I/O Pin 10 (Bank 1)
18	IO11	I/O	General Purpose I/O Pin 11 (Bank 1)
19	GND	-	Ground
20	IO12	I/O	General Purpose I/O Pin 12 (Bank 1)
21	IO13	I/O	General Purpose I/O Pin 13 (Bank 1)
22	GND	-	Ground
23	IO14	I/O	General Purpose I/O Pin 14 (Bank 1)
24	IO15	I/O	General Purpose I/O Pin 15 (Bank 1)
25	GND	-	Ground
26	IO16	I/O	General Purpose I/O Pin 16 (Bank 2)
27	IO17	I/O	General Purpose I/O Pin 17 (Bank 2)
28	GND	-	Ground
29	IO18	I/O	General Purpose I/O Pin 18 (Bank 2)
30	IO19	I/O	General Purpose I/O Pin 19 (Bank 2)
31	GND	-	Ground
32	IO20	I/O	General Purpose I/O Pin 20 (Bank 2)
33	IO21	I/O	General Purpose I/O Pin 21 (Bank 2)
34	GND	-	Ground
35	IO22	I/O	General Purpose I/O Pin 22 (Bank 2)
36	IO23	I/O	General Purpose I/O Pin 23 (Bank 2)
37	GND	-	Ground
38	IO24	I/O	General Purpose I/O Pin 24 (Bank 3)
39	IO25	I/O	General Purpose I/O Pin 25 (Bank 3)
40	GND	-	Ground
41	IO26	I/O	General Purpose I/O Pin 26 (Bank 3)
42	IO27	I/O	General Purpose I/O Pin 27 (Bank 3)
43	GND	-	Ground

Table 6: J4 And J6, General Purpose I/O Connectors (Continued)

PIN	Signal	I/O	Function
44	IO28	I/O	General Purpose I/O Pin 28 (Bank 3)
45	IO29	I/O	General Purpose I/O Pin 29 (Bank 3)
46	GND	-	Ground
47	IO30	I/O	General Purpose I/O Pin 30 (Bank 3)
48	IO31	I/O	General Purpose I/O Pin 31 (Bank 3)
49	GND	-	Ground
50	GND	-	Ground

Table 7: J5 And J7, Serial Peripheral Interface (SPI) Connectors (Continues)

PIN	Signal	I/O	Function
1	GND	-	Ground
2	MISO	I	Master Input Slave Output
3	GND	-	Ground
4	MOSI	O	Master Output Slave Input
5	GND	-	Ground
6	SCLK	O	SPI Clock
7	GND	-	Ground
8	CSSPI0*	O	SPI Chip Select 0
9	CSSPI1*	O	SPI Chip Select 1
10	GND	-	Ground
11	CSSPI2*	O	SPI Chip Select 2
12	CSSPI3*	O	SPI Chip Select 3
13	GND	-	Ground
14	CSSPI4*	O	SPI Chip Select 4
15	CSSPI5*	O	SPI Chip Select 5
16	GND	-	Ground
17	CSSPI6*	O	SPI Chip Select 6
18	CSSPI7*	O	SPI Chip Select 7
19	GND	-	Ground
20	CSSPI8*	O	SPI Chip Select 8
21	CSSPI9*	O	SPI Chip Select 9
22	GND	-	Ground
23	CSSPI10*	O	SPI Chip Select 10
24	CSSPI11*	O	SPI Chip Select 11
25	GND	-	Ground
26	CSSPI12*	O	SPI Chip Select 12
27	CSSPI13*	O	SPI Chip Select 13
28	GND	-	Ground
29	CSSPI4*	O	SPI Chip Select 14
30	CSSPI5*	O	SPI Chip Select 15
31	GND	-	Ground
32	MISO+	I	Master Input Slave Output Differential High True
33	MISO-	I	Master Input Slave Output Differential Low True
34	GND	-	Ground
35	MOSI+	O	Master Output Slave Input Differential High True
36	MOSI-	O	Master Output Slave Input Differential Low True
37	GND	-	Ground
38	SCLK+	O	SPI Clock Differential High True
39	SCLK-	O	SPI Clock Differential Low True
40	GND	-	Ground
41	IPREQ*+	O	IP Address Request Differential High True
42	IPREQ*-	O	IP Address Request Differential Low True
43	GND	-	Ground

Table 7: J5 And J7, Serial Peripheral Interface (SPI) Connectors (Continued)

PIN	Signal	I/O	Function
44	No Connection	-	No Connection
45	No Connection	-	No Connection
46	No Connection	-	No Connection
47	HEARTBEAT+	I	Heartbeat RS422/485 Differential High True
48	HEARTBEAT-	I	Heartbeat RS422/485 Differential Low True
49	GND	-	Ground
50	GND	-	Ground

Table 8: J8 And J10, Miscellaneous I/O Connectors (Continues)

PIN	Signal	I/O	Function
1	GND	-	Ground
2	GRESET+	I	Global Reset RS422/485 Differential High True
3	GRESET-	I	Global Reset RS422/485 Differential Low True
4	GND	-	Ground
5	ASC RXD	I	RS232 ASC RXD
6	No Connection	-	No Connection
7	GND	-	Ground
8	ASC TXD	O	RS232 ASC TXD
9	No Connection	-	No Connection
10	GND	-	Ground
11	INT IN0	I	Interrupt 0
12	TIMER IN0	I	Timer Input 0
13	INT IN1	I	Interrupt 1
14	TIMER IN1	I	Timer Input 1
15	INT IN2	I	Interrupt 2
16	TIMER OUT0	O	Timer Output 0
17	INT IN3	I	Interrupt 3
18	TIMER OUT1	O	Timer Output 1
19	INT IN4	I	Interrupt 4
20	GND	-	Ground
21	INT IN5	I	Interrupt 5
22	GND	-	Ground
23	INT IN6	I	Interrupt 6
24	GND	-	Ground
25	INT IN7	I	Interrupt 7
26	GND	-	Ground
27	PORST*	O	Power On Reset
28	GND	-	Ground
29	LRESET*	O	Local MIB Software Generated Reset
30	GND	-	Ground
31	HEARTBEAT	-	Heartbeat (Un-buffered)
32	GND	-	Ground
33	RS RXD	I	RS232 Receive Data
34	GND	-	Ground
35	RS TXD	O	RS232 Transmit Data
36	GND	-	Ground
37	RS DTR	O	RS232 Data Terminal Ready
38	GND	-	Ground
39	RS RTS	O	RS232 Request to Send
40	GND	-	Ground
41	RS DCD	I	RS232 Data Carrier Detect
42	GND	-	Ground
43	RS CTS	I	RS232 Clear to Send

Table 8: J8 And J10, Miscellaneous I/O Connectors (Continued)

PIN	Signal	I/O	Function
44	GND	-	Ground
45	RS DSR	I	RS232 Data Set Ready
46	GND	-	Ground
47	RS RI	I	RS232 Ring Indicator
48	GND	-	Ground
49	GND	-	Ground
50	GND	-	Ground

Table 9: J9 And J11, Power Connectors

PIN	Signal	I/O	Function
1-12	GND	-	Ground
13-24	+5VCC	-	+5 Volts
25	GND	-	Ground
26	GND	-	Ground

Table 10: J12, MIB Required External Signals

PIN	Signal	I/O	Function
1	GND	-	Ground
2	GRESET+	I	Global Reset RS422/485 Differential High True
3	GRESET-	I	Global Reset RS422/485 Differential Low True
4	GND	-	Ground
5	HEARTBEAT+	I	Heartbeat RS422/485 Differential High True
6	HEARTBEAT-	I	Heartbeat RS422/485 Differential Low True
7	GND	-	Ground
8	MOSI+	O	Master Output Slave Input Differential High True
9	MOSI-	O	Master Output Slave Input Differential Low True
10	GND	-	Ground
11	MISO+	I	Master Input Slave Output Differential High True
12	MISO-	I	Master Input Slave Output Differential Low True
13	GND	-	Ground
14	SCLK+	O	SPI Clock Differential High True
15	SCLK-	O	SPI Clock Differential Low True
16	GND	-	Ground
17	IPREQ*+	O	IP Address Request Differential High True
18	IPREQ*-	O	IP Address Request Differential Low True
19	GND	-	Ground
20-26	No Connection	-	No Connection

4 Built-in Firmware

4.1 General MIB Firmware Considerations

The MIB contains software that is common to all MIB's, as well as software that is specific to the device or devices that the MIB controls (module software). The device software is downloaded over the network into the Flash memory on the MIB. Once the MIB has loaded the module software, it will not rewrite that module software, unless it determines it has been placed into a different module, or there has been a software revision for that particular module.

The MIB software contains the Real Time Operating System (RTOS) as well as the module operations code. The RTOS will carry out the communication duties as well as insure that tasks are carried out on time. The module operations code is broken into functions for initialization, operation, monitor and control, screen interface, and testing.

Upon power-up the MIB will begin executing code inside the Flash memory. This code contains the RTOS and MIB operations code. Please note that this stand-alone code is separate from the module software. This startup code will initialize the TC11IB peripheral functions so that the MIB can enact an IP Request using SPI. This SPI access will load a block of information that includes Antenna ID, Slot ID, and Slot parameters. The MIB will then send an Ethernet message denoting its presence on the network. From the Slot ID, the type of module that the MIB is expected to be will be determined. If the module software doesn't match, then the MIB will signal this condition over the network. The normal response by the network will be to load this MIB with the proper module software. This module software would then be written into the Flash memory.

At this point, the MIB will transfer the module software from Flash into the internal TC11IB 1.5M byte memory. Code would then execute from the TC11IB and the Flash memory would go dormant. The Flash memory might be used for module testing or update purposes, but should not be used for normal module software operations.

Once the MIB begins executing the module software, it will initialize the interface hardware and initialize the device, if required. It will then drop into normal operation, executing monitor and control tasks for the device as well as handling Ethernet communications. The screen interface is one example of using Ethernet communications for the MIB.

Finally, module software code updates could be managed by two methods. The first method is that just prior to the module software transfer, the MIB would request the code revision for the software. If the revision didn't match, the MIB would signal the network for an update. The other method would be that the network could command the MIB to load the update after getting the revision level of the code.

5 General Purpose I/O

5.1 General Purpose I/O Considerations

Each MIB, as stated before, has 32 bits of General Purpose I/O broken into four banks. In order to provide the Device Designer and Software Designer some common ground so that the MIB can be programmed to carry out the proper tasks, a couple of tables should be filled out, or other documentation should be provided that defines the interface between the MIB and the device. This documentation should specify any monitor and control tasks that the MIB is expected to perform on the device with the General Purpose I/O.

The first table, Table 11: Bank Settings, defines each bank. The column marked Signal Direction should be filled with Input, Output, Bi-directional, or Unused in order to tell the Software Designer what the Bank should be set for. For Bi-directional, placing an (I) or (O) after the entry will tell the Software Designer the quiescent state it should be left in between accesses. The column marked Signal Group Name provides the name that the particular Bank can be called or the name of the group function it may carry out. For example, Address [8..15] would denote a high order address bus that has been assigned to a Bank. For a Bank that carries out bit functions, it probably would be best that an additional separate description be provided. The Initial Value column is mainly meant for an Output Bank and simply describes what the Bank should output when the module software initializes.

The next table, Table 12: General Purpose Bit Functions, defines the function for each of the 32 bits of General Purpose I/O. This table can be supplemented or replaced with more tables and documentation that describe functions and software requirements in a more complete manner. This additional information would be excellent for cases where multiple bits carry out multiple functions for the Device.

Finally, each Bit Function should have a name, what logic level is the true state, MIB response, etc. For example an input could be:

PLL Lock, High=Lock, Low=Report Astronomy Dead at 20 cm.

An Output could be:

6 CM Refrigerator Active, Low=Active, MIB Activates via Command Request/Initialization

Table 11: Bank Settings

Bank	Signal Direction	Signal Group Name	Initial Value
0			
1			
2			
3			

Table 12: General Purpose Bit Functions

IO #	Function
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	

6 Serial Peripheral Interface (SPI)

6.1 SPI Considerations

Each MIB, as stated before, has been given a standard capability to work with sixteen SPI devices. Each SPI device ranges from simple registers to very complex devices such as microprocessors. The Device Designer will have to provide documentation and work closely with the Software Designer in order to provide the best fit for every SPI device in his design.

SPI is a form of synchronous serial transmission. The basic aspect of SPI is that each SPI device has a serial input, a serial output, and a serial clock. The MIB will be considered to be the master and all additional SPI devices will be considered slaves. The MIB provides the serial clock and controls all data transfers.

The simplest form of SPI communications is one master and one slave. In this form, only the three lines mentioned above need to be connected in the proper fashion. If more then one slave exists, then an additional line, usually called the Slave Select (SS), is used to select a particular slave. The SS line acts as a device select and as the output enable line for a SPI device. In this document, these lines generated by the MIB are called CSSPIn* for Chip Select SPI. The letter “n” is a number from zero to fifteen that in turn denotes which SPI device it is.

Figure 3 shows the basic connections needed to connect the MIB SPI Master to multiple SPI slaves. The line designated Master-Output-Slave-Input (MOSI) simply states that the Master-Output connects to all Slave-Inputs. The same is true for the Master-Input-Slave-Output (MISO) in that the Master-Input must connect to all the Slave-Outputs. The SPI Clock (SCLK) is connected together for all the SPI devices. As stated before, the MIB is the source for the SPI Clock. The active low Slave Select (SS*) lines must be generated one per slave device. The MIB provides sixteen lines that the Device Designer can utilize, but additional lines can be generated via digital logic controlled by the MIB. In Figure 3, the first slave shows the method to daisy chain multiple slaves together, where the SS* line on its low to high transition normally acts as a data valid signal for each device. This methodology has the problem in that when one needs to update one slave in the chain, the MIB must remember the other slave values. Over time the MIB could lose track and thus it would be impossible to load the current values, so the daisy chain method would best be avoided. The depictions for Slave 2 and Slave 3 in Figure 3 denote the expected usage by which SPI devices are connected to the MIB.

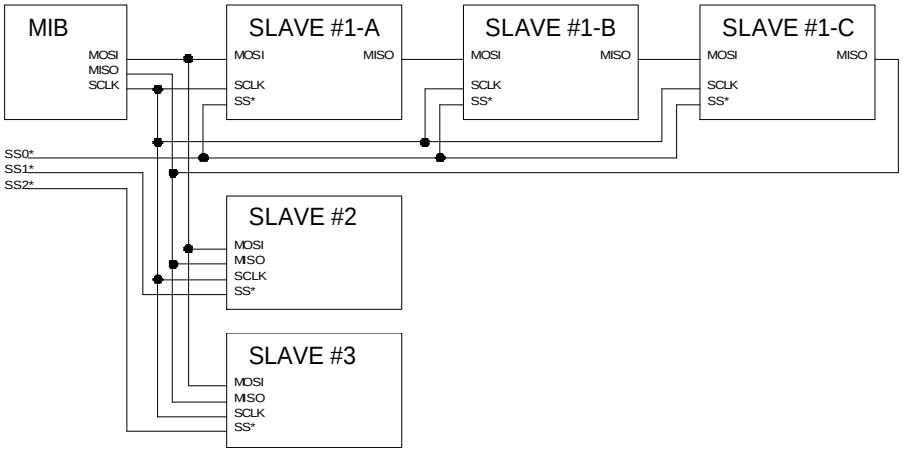

Figure 3: SPI Connections – MIB Master to Slaves

The actual communication using SPI is straightforward. The master and slave simply form a ring buffer, such that when the master transmits its buffer to the slave, the slave transmits its buffer to the master. One transition of the clock shifts the data out of each buffer. The next transition of the clock latches the data into each buffer.

Figure 4: Basic SPI Transfer Timing Example shows the timing for a typical byte transfer. In this transfer, the SS* is driven low to select the specific slave. Next the master clocks the data transfer. In this case, the falling edge of the clock causes the shift registers to place the next data bit onto MISO and MOSI. On the rising edge, this bit will be sampled and loaded into each buffer. When the transfer ends, the SS* will return high, thus deactivating the slave. The master has complete control over the clock's idle state, the transfer rate, which clock edge that shifts/latches the data, MSB/LSB first, and how many bits are to be transferred. Having that level of control calls for standardization, however, each slave may require different setups. If the Device Designer can select devices that fall into the following categories, then software would be greatly simplified.

? Clock Idle = High
? Clock Transition = Low to High Transition Latches Data
? Clock Speed = Less Then 6 MHz
? Transfer Size = Byte (8 bits) or Word (16 bits)
? MSB/LSB First = MSB First

Usage of logic gates that change any of the above characteristics can easily implement different SPI devices. If an unusual SPI device is selected which doesn't fit the above, and gating cannot provide for its operation, then it would have to be coded as a special case.

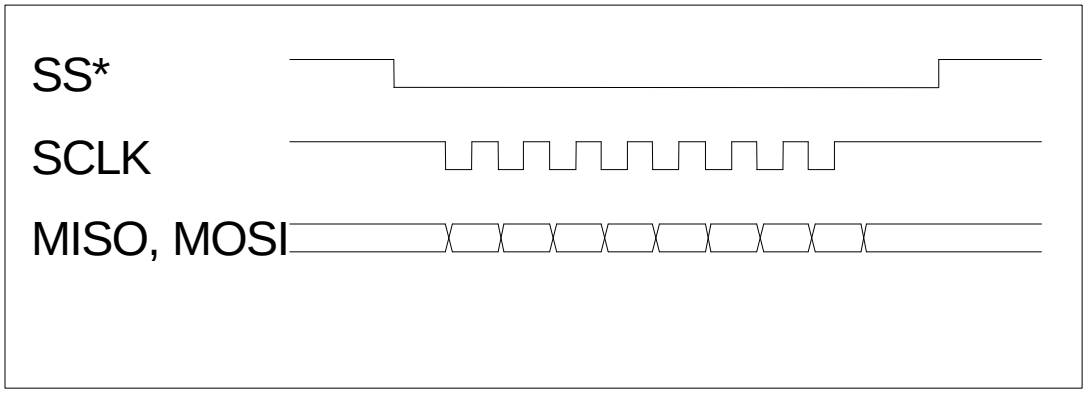

Figure 4: Basic SPI Transfer Timing Example

The next aspect is how to generate the SS lines for the case where the number of SPI devices exceeds sixteen, or when the device hardware will not use the sixteen lines the MIB can provide. For the first case where the devices exceed sixteen, one or more Banks of the General Purpose I/O can be dedicated for this purpose. One would simply designate the Bank to be Output, Initial Value = 0xFF. Each Bank used, would add eight more SS lines up to an additional thirty-two. The device designer would then need only to be certain to provide buffers for the three major SPI lines MOSI, MISO, and SCLK if applicable. For a design that goes beyond forty-eight SPI devices, or where the General Purpose I/O cannot be used, the Device Designer would have to switch to an addressing scheme.

The basic idea behind the addressing scheme is to provide an address value to the hardware, which then decodes each value into a distinct SS line. This address value can be passed to hardware via the SPI bus itself or by redefining CSSPI0* through CSSPI15* to be an address instead of an individual SS line. Device hardware would be responsible for the decoding implementation. Also, an address value of 0xFFFF defines that no SPI device is selected.

Figure 5: SS* Line Generator Using SPI Bus, shows a method of developing SS* lines via the SPI bus. This case would be used when the MIB is fairly remote or external from a device and limited cabling is desired. In order to minimize RFI generation, when sending SPI signals external to the module, LVDS signals for MOSI, MISO, SCLK, and any CSSPI* lines should be developed in a device. Device Designers would use these LVDS lines for external SPI communications

The circuit design example uses Xilinx logic FPGA blocks. It could be easily implemented into other hardware devices as the Device Designer sees fit. For this design the MOSI line feeds a shift register at U3. In order for the shift register to shift in a new address value, the MIB must set ADDRSEL* to the active low state. ADDRSEL* also disables U12 so that when a new address is being loaded, it will not cause SS* lines to glitch during the shift period. The SPICLK is the SPI Clock with the assumption that the low to high is the transition to latch a bit. The PORST* is the power on reset line that clears the shift register and also via U3-Q15 disables U12. This hardware provides for 32K addresses and therefore should provide enough capability for very complex SPI buses. When DATASEL* is driven low, provided U3-Q15 had a high shifted into it, will cause one of the SS* lines to go low. This occurs because DATASEL* enables U12 that feeds the U8 decoder. One output of U8 will enable another decoder such as U4. The single

output of the secondary decoder will be inverted, and that line then drives a single SS* to the active low state. Additionally the U8 outputs are also inverted and brought out in order to provide control over the external SPI bus by grouping it into banks of sixteen. By simple logic expansion, each U8 output can control another 16 line decoder, so this design can provide a total of 256 SS* lines. In order to produce more lines, one would need to supply an additional layer of decode logic. The MISO line isn't needed, as setting the address is a write-only function. Writing an address value of zero deactivates the decode logic. The ADDRSEL* line will be equivalent to CSSPI14* and DATASEL* is equivalent to CSSPI15* for standardization purposes.

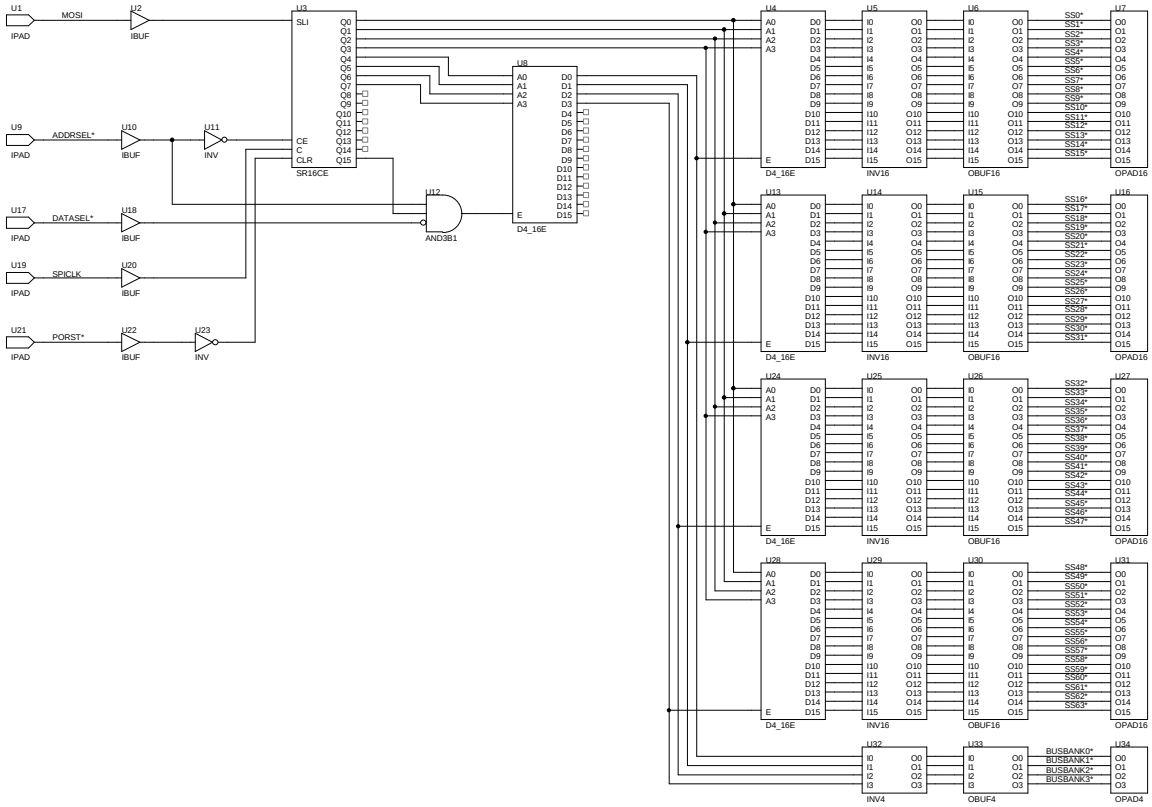

Figure 5: SS Line Generator Using SPI Bus*

Finally, each SPI device that is selected will have a protocol in order to utilize it properly. Device Designers should attempt to communicate with other designers to find SPI devices that can be used across designs. This will simplify the software development, in that the number of protocols would be minimal. For the unique case where we might have MIB to microprocessor communications, NRAO would generate a standard protocol for that communication. In fact via ALMA, a protocol was established by NRAO for such a purpose. It would be necessary for the EVLA to modify this protocol in order to use it more effectively.

7 Asynchronous Synchronous Communications (ASC)

7.1 ASC Considerations

Each MIB has, in addition to SPI, the capability to communicate with a separate Asynchronous Synchronous Communications (ASC) port. In effect, this is a two-wire port that has been connected to RS232 interface devices. This was done to allow the MIB and a simple terminal to act in concert, in order to have local dedicated communications for software development with this port.

8 RS232 Communications

8.1 RS232 Considerations

Each MIB has capability to communicate with another separate RS232 communications port. This port is what one would call a modem style port, in that it has control signals such as Clear to Send (CTS), in addition to the normal RS232 RxD and TxD. A standard RS232 interface device was used and thus RS232 signal levels are present. This was done because it was thought that this port will allow the MIB and a commercial device to communicate using the well known RS232C interface standard. Thus a unique commercial device using RS232 can communicate to a MIB for monitor and control purposes and information can be provided back to the system via Ethernet.

To fully implement this port, the Device Designer will work with the Software Designer in order to work out details of communication protocols with such a commercial device.

9 Timer

9.1 Timer Considerations

Each MIB has capability to bring in two external clocks from the device that can drive the internal TC11IB General Purpose Timing Unit 1 (GPTU1). Additionally, two output lines from the GPTU1 have also been provided to the device. The GPTU1 can be programmed various ways in order to utilize these inputs clocks and output section of this timer. An example could be that after the timer counts a particular value of input clocks, a TC11IB timer interrupt is generated to cause the MIB to carry some software function. Further clocks or even a different clock can be counted and an output timing pulse to the device could be generated.

To fully implement any timing capabilities, the Device Designer will work with the Software Designer in order to work out details of his device's timing requirements with the MIB.

10 MIB Implementation Details

10.1 General Implementation Considerations

The MIB schematic D23710S0001 consists of seven sheets, developed using the schematic capture program known as Protel. The printed circuit board layout D23710Q0001 was done using Protel as well.

The first sheet in the schematic is meant to clarify the sheet-to-sheet interconnections. The second sheet is the power management sheet. The third sheet is the Infineon Tricore TC11IB microcontroller with some support logic. The fourth sheet deals with the parallel devices in this design. The fifth sheet deals with the serial devices in this design. Finally, the sixth and seventh sheets deal with the extensive filtering in this design in order to reduce RFI, both generated by the MIB and external to the MIB.

In the following subsections, each sheet will be described as to the theory of operation for the MIB design.

10.2 Sheet 1: Sheet-to-Sheet Interconnections

This MIB schematic sheet doesn't contain any design information per se. Rather it is the hierarchical sheet that glues all of the other sheets in this design together. However, it does allow one to follow a signal or signals from one sheet to another. The notes on this sheet also explain the naming convention used for this schematic diagram.

10.3 Sheet 2: Power Management

This MIB schematic sheet deals with the power requirements that the MIB has. The core of this sheet is a Texas InstrumentsTPS70351PWP regulator, power sequencer, and supervisor. The regulator produces +3.3V and +1.8V from a nominal +5V input. The power sequencer powers one voltage output and then the other voltage output dependent upon a logic setting at the pin SEQ. This pin has been tied to a logic low, such that the +3.3V powers first. When the +3.3V reaches approximately 85% of it nominal output, the +1.8V then powers up. Blue LED's visually indicate the input and output power status. When both supplies reach their output levels, after 140ms the RESET pin, which has remained low during all this time, will go high. This provides a clean reset during the power up state. The device will remain in this operational state provided that the power demands remain within specification. If demands are too great, the power supervisor of this device will then power down in order to protect the MIB, and cause the RESET pin to go low. Switch S1 allows the ability to reset the MIB independent of power conditions. The other circuitry allows an additional means via a RS422/485 bus connection to reset the MIB using a signal external to the module known as the Global Reset. The RESET pin, called PORST*, goes directly to the TC11IB.

The decoupling capacitors are shown on this sheet as well. They have been broken down such that each additional sheet has its decoupling capacitors designated. For the TC11IB, ferrite beads have been used to further prevent possible digital noise produced by the TC11IB from tainting other MIB devices on or off board via power connections.

10.4 Sheet 3: Infineon Tricore TC11IB Microcontroller

This sheet details the Infineon Tricore TC11IB microcontroller. The TC11IB was broken into six sections in order to group logical blocks of the TC11IB together. These six sections are Operation and Support, External Bus Unit, Power, Ports, PCI Bus, and Ethernet.

10.4.1 Operation and Support Section:

The Operation and Support Section details the oscillator, the PLL power, reset, On-Chip Debug Service (OCDS), and configuration.

The oscillator consists of an external 12MHz crystal that is connected to the XTAL1 and XTAL2 pins that in turn connect to an internal TC11IB oscillator. This oscillator circuit feeds the 12MHz to an internal TC11IB PLL that produces 48MHz and 96MHz clocks. Power is supplied via 10µH inductors and 10µF capacitors in order to filter both the oscillator and PLL power sections in order to provide isolation for these clock generators.

The reset going into the TC11IB, known as PORST*, originates from the Power Management sheet. This reset in turn produces a TC11IB synchronous reset known as HDRST*. This reset drives all the rest of the reset requirements internal and external to the MIB.

The OCDS is connected to two connectors, J1 and J2. The J2 connector provides level one OCDS, which are your basic JTAG signals. J1 provides level two OCDS that improves capabilities for software debugging or emulation efforts.

Configuration is set by tying CFG0 through CFG3 to the levels as indicated on the schematic. Along with the setting of the OCDS ENABLE, BRKIN, and Port 0: 11 thru 13 pins, these provide sub-modes for booting up the operating code or placing the TC11IB into a defined state. In order to simplify the number of possibilities Port 0: 11 is a high, whereas Port 0: 12, 13 are low.

In normal operation both the OCDS ENABLE and BRKIN are high disabling the external controls for debugging or emulation. The configuration is set such that the TC11IB will execute code contained in Flash Memory that is shown on sheet 4.

Dependent on when either OCDS ENABLE or BRKIN are low, the TC11IB can be placed into a deep sleep mode, a halted state, or in emulator mode. When the PORST* signal goes high, all of the above states are loaded into the TC11IB and stored into a register.

10.4.2 External Bus Unit Section:

The External Bus Unit is the core for all of the parallel functions for the MIB. It provides a thirty-two bit data bus and a twenty-four-bit address bus. Also, it provides normal controls such as read and write strobes, data bus bank controls, and programmable chip selects. All of these lines connect to the devices on sheet 4, the Parallel I/O sheet.

Only three programmable chip select lines are used. The flash memory is connected to both CS0* and CS1*. The CS2* chip select, along with the four data bus banks, controls access to the registered transceiver buffers that provide the MIB General Purpose I/O function.

10.4.3 Power Section:

This section shows the bulk of the pins on the TC11IB dedicated to power. These pins have been grouped together and are isolated and decoupled using the ferrite beads and capacitors as shown on sheet 1. Certain power pins specific to a different block are shown on that block for the TC11IB. This was done in order to make clear those specific power needs and circuitry.

10.4.4 Port Section:

This section shows the usage of the TC11IB ports, except Port 2, which is dedicated to Ethernet and has been placed by itself on the schematic.

Port 0 is used to interface with an internal TC11IB timer group. Port 0: 8 and Port 0: 9 provides the means to bring in two independent external clocks. Next, Port 0: 14 and Port 0: 15 are independent outputs from the timer group, so timing signals can be fed back to device hardware. All four signals are buffered as shown on the schematic. A software generated reset for use by the device is developed from Port 0: 10. This output line can also be connected to an internal TC11IB timer in order to produce a reset or other pulses of a programmable length. Finally, Port 0: 11 thru 13 are inputs that provide required configuration options at power-up. Port 0: 11 is tied high, whereas Port 0: 12, 13 are tied low.

Port 1 provides the RS232, SPI, and ASC serial capabilities. All of these lines, except for SPI, connect to the devices on sheet 5, the Serial I/O sheet. For SPI, those lines connect to the devices on sheet 4, the Parallel I/O sheet as well as sheet 5, the Serial I/O sheet.

Port 3 provides input into the interrupt logic of the TC11IB. Eight interrupts INT [0..7] are devoted to Device Designer use. Another interrupt is for the HEARTBEAT signal that provides the MIB its core-timing signal. The last interrupt comes from the Ethernet Controller in order to provide interrupts due to Ethernet communications. The device interrupts are buffered on sheet 4, the Parallel I/O sheet, and the HEARTBEAT and Ethernet Controller interrupt signals are from sheet 5, the Serial I/O sheet.

Port 4 develops the sixteen SPI device select lines CSSPI [0..15]. These lines are buffered on sheet 4, the Parallel I/O sheet.

Port 5 consists of control signals for MIB interface hardware. Ten bits from this port control different sections of the MIB design, enabling these sections to be active or inactive, as the MIB or device requires. All of these signals go to sheet 4, the Parallel I/O sheet.

SPIEN0*, when low, activates the LVDS SPI interface that is used to access the external Slot-ID device outside the module. SPIEN1*, when low, activates the LVTTL SPI interface that is used for internal device SPI accesses. If SPIEN0* is also low with SPIEN1*, the LVTTL MISO signal is deactivated and the LVDS MISO signal is active.

INTEN*, when low, allows the device interrupts to drive the TC11IB Port 3 lines. The HEARTBEAT or Ethernet Controller interrupt signal are not affected by the state of this signal.

IPREQ*, when low, indicates that the MIB is looking to receive the slot information over SPI. This information is stored in an external SPI flash EEPROM. This EEPROM will have Ethernet addresses, the antenna identification, and slot information. For detailed information on Slot-ID see document A23010N0003: EVLA Hardware Networking Specification.

The line SFLASHEN* enables an on-board SPI flash EEPROM. Contained in this EEPROM is the MIB Media Access Control (MAC) address, MIB serial number, and MIB revision level. Other MIB information including software-generated information will also be stored in this EEPROM. A later section will describe the location for this information.

The lines WRMODE0* thru WRMODE3* control the General Purpose I/O registered buffer transceiver banks. When these lines are low, that particular bank will be an output bank.

10.4.5 PCI Bus Section:

This section simply documents the capability for PCI on the TC11IB. For the MIB design, this section is simply not used.

10.4.6 Ethernet Section:

Port 2, along with three other lines, produces the Media Independent Interface (MII) that is the core for the Ethernet support by the TC11IB. These interface signals go to sheet 5, the Serial I/O sheet, and connect to the Intel LXT971A device in order to send or receive the Ethernet serial data stream over fiber optics.

10.5 Sheet 4: Parallel I/O

This sheet deals with the parallel device needs for the MIB and device. The bulk of this sheet is the buffering that drives the signals to the outside world of the device. The sheet is broken into six sections that are General Purpose I/O, Parallel Flash, Serial Flash, Reset Buffering, SPI Input Buffering, SPI and Interrupt Buffering, and the Main I/O Connectors.

10.5.1 General Purpose I/O Section:

The General Purpose I/O Section details the registered transceiver buffers that are the parallel interface between the TC11IB and the Device.

These buffers are two Texas Instruments 74ALVC16543DL devices. On the MIB side, they are connected to the thirty-two-bit data bus. Reading or writing to the transparent registers is controlled by the MIB RD* and WR* along with the CS2* signal. As far as the MIB is concerned, these devices act as a simple memory element. The WRMODE0* thru WRMODE3* signals determine whether or not a particular bank of 8-bits acts as an output. When low, they force the particular bank to act like an output. When in the output state, any read reflects the state of the output pin. This automatically provides the read after write or command read-back function.

10.5.2 Parallel Flash Section:

The Parallel Flash section consists of Flash Memory that provides 8M or 16M bytes of memory storage.

Both the MIB address and data busses are connected to this memory. The memory device is an AMD AM29LV640D or AM29LV642D for the 8M-byte version or 16M byte versions respectively. Each version of Flash memory provides sixteen bit memory accesses. It will contain all the code needed to operate the MIB in a stand-alone state, as well as the code to operate the particular device the MIB is installed in. Standard strobes for reads and writes are fed to this memory device. Also either chip select line CS0* or CS1* will select an 8M byte section of the AM29LV642D, while CS1* isn't used for the AM29LV640D. The Flash memory is meant only to operate in a temporary fashion. All code will be transferred to the TC11IB as quickly as possible. Certain specialized code could operate out of the Flash memory such as module testing code, or Device code loading software. Once the Device code is in operation, the Flash memory will be dormant.

10.5.3 Serial Flash Section:

This section consists of a Microchip 25LC040 SPI Serial Flash Memory device. It is a 4K-bit device that communicates using SPI access commands. The device is enabled when the SFLASHEN* goes low. From that point an eight-bit value is sent to it, which then enables a read or write function. Signals SPIEN0* and SPIEN1* are high so they disable external SPI accesses during this time. Resister R101 is used to protect this EEPROM from further writes into the EEPROM if it is installed.

10.5.4 Reset Buffering and Test Light Sections:

This section shows the two resets provided to the device as well as the on-board test light.

The RESET* signal occurs at the same time the TC11IB is reset. This provides the device with the basic power-on or MIB reset, for use by the device. The next reset, LRESET*, is generated by MIB software. This reset allows the MIB to reset the device

completely independent from a hardware reset. This buffering provides LVTTL levels to the device. Finally, both of these signals are optional. It is entirely up to the Device Designer to utilize these signals or to provide his own reset as needed.

A green LED test light has been provided so that software programmers have some indication of MIB operation. Currently, when the MIB is running its software, this light is blinking off and on approximately for one second for each state. This light can be used for other purposes as well, such as lighting when a particular software routine is executed.

10.5.5 SPI Input Buffering Section:

There are two SPI buses, one with LVTTL levels and the other with LVDS levels. This section controls which MISO signal is inputted into the TC11IB.

When the control signal SPIEN0* is low, this enables the LVDS MISO signal to be input into the TC11IB despite the state of SPIEN1*. When only SPIEN1* is low, then the LVTTL MISO signal is inputted into the TC11IB. When both SPIEN0* and SPIEN1* are high, the on-board EEPROM MOSI signal can be sent to the TC11IB if that device has been enabled by the SFLASHEN* signal.

10.5.6 SPI and Interrupt Buffering Section:

This section is simply a Texas Instruments SN74LVTH32244, a large LVTTL buffer that buffers signals off or onto the MIB.

Most of the buffer is devoted to the LVTTL SPI signals MOSI, MISO, SCLK, and CSSPI0* thru CSSPI15* SPI select lines. A small subsection buffers the SPI signals that produce the LVDS SPI levels. As stated previously SPIEN0* enables the LVDS level section of this buffer, while SPIEN1* enables the LVTTL section of this buffer.

The eight external interrupts from the device are buffered here as well. The state of INTEN* enables or disables these interrupt signals.

All outputs of this buffer are weakly pulled high so that any sections of the buffer, when disabled, will produce a high level.

10.5.7 Main I/O Connectors and Common External Connector Sections:

These eight connectors provide the MIB with flexible methods of device interconnections. Four female socket connectors will be mounted on the MIB's bottom surface in order to provide a means by which the MIB can connect to a motherboard. Four male header connectors will be mounted on the MIB's top surface in order to provide a means by which the MIB can connect to a daughterboard or connect to a device via ribbon cable. Each matching pair of connectors are directly across from one another, thus providing a means by which the MIB can be stacked together with multiple PCB's. The power connector set is smaller then the rest, so a method of keying PCB's exists.

The common external connector provides a one-stop means to link signals that are required by the MIB such as the Slot-ID signals, as well as signals that enhance MIB operations such as the global reset and heartbeat signals in one place. This connector is mounted only one side of the MIB, unlike the above connectors. The signals that are present on this connector are on the above connectors as well, so that one can utilize them in a flexible manner.

These signals are the LVDS SPI signals, the LVDS IPREQ* signal, the Global Reset signal and the RS422/485 HEARTBEAT signals.

The LVDS SPI and IPREQ* signals are the means by which the MIB identifies its slot address and module type.

The RS422/485 HEARTBEAT signal can also be inputted to the MIB at this connector.

10.6 Sheet 5: Serial I/O

This sheet deals with the serial device needs for the MIB and device. The sheet is broken into four sections that are both RS232 ports, LVDS Interface, RS85 Heartbeat, and Ethernet Controller with the Fiber Optic Transceiver as well.

10.6.1 RS232 Port 1 and Port 2 Sections:

The RS232 Port 1 signals interface via a Maxim MAX3244. This device provides RS232 levels from the signals generated by the TC11IB serial modem port. These signals are brought to the main connectors on the Parallel Sheet.

The RS232 Port 2 signals interface via a Maxim MAX218CWP. This device provides RS232 levels from the signals generated by the TC11IB ASC port. These two signals are also brought to the main connectors on the Parallel Sheet.

10.6.2 LVDS Interface Section:

The LVDS Interface Section details the signal level translation devices that convert various output signals from LVTTL to LVDS and vice versa for input signals.

A Texas Instruments SN65LVDS180 provides the LVDS interface for the SPI serial streams. The enable control lines are always active because the MIB SPI is full duplex.

Another Texas Instrument part, a SN65LVDS3487D, provides the LVDS interface for the rest of the signals. These signals, all outputs, are SCLK, and IPREQ*.

All of these signals are also brought to the main connectors on the Parallel Sheet as well as the external common connector.

10.6.3 RS485 Heartbeat Section:

The distribution for this signal was chosen to be RS422/485 for all MIB's. This signal is fed into a Maxim MAX3362EKA in order to provide the LVTTL signal for the TC11IB. The input Heartbeat signal is present on the main connectors on the Parallel Sheet as well as the external common connector.

This generated LVTTL signal goes to a main connector on the Parallel sheet as well, but it is un-buffered. If the device uses this signal, the device should buffer it. If the device requires high hardware timing accuracy, another related timing signal has been developed for that purpose and should be used instead.

10.6.4 Ethernet Section:

TheTC11IB provides a Media Independent Interface (MII). The signals are tied to an Intel LXT971A, that in turn provides a full duplex Ethernet serial stream interface. The LXT971A is clocked with a 25MHz crystal in order to produce the correct bit rates for the Ethernet stream. All MIB communications, control, and data are via MII. The LXT971A provides signals that interface to physical devices that carry the actual Ethernet signals over fiber optics, in this case.

The fiber optic transceiver is an Agilent HBFR-5103P. This device provides Ethernet over fiber optics under the IEEE 802.3 sub-standard known as 100Base-FX.

10.7 Sheets 6,7: Digital Filtering

These two sheets provide filtering on the digital signals on the MIB that go external between the devices. Each signal has been analyzed for its characteristics and a filter type selected to give the best results that doesn't compromise its use. This is another part of the MIB design that is intended to reduce the RFI characteristics of digital signals. Additionally, careful attention was paid during the PCB layout process to all signal lines in order to further reduce the MIB's potential for RFI.

If each signal line was explained as to this filtering, this document would too complex to follow. However, using the documentation on the filter devices and the signal that is involved, interested parties can analyze what was done in the design.

Each signal that is filtered has a name as it enters these sheets. The filtered signal simply has the prefix "FL_" added to the original signal name. For example, the signal "ABCD" after filtering would be named "FL_ABCD".

Figure 6 shows a simulation for the filter that is used for the I/O signals. The simulation was done for rise and fall times of 2 ns, and for a maximum data rate of 10 MHz.

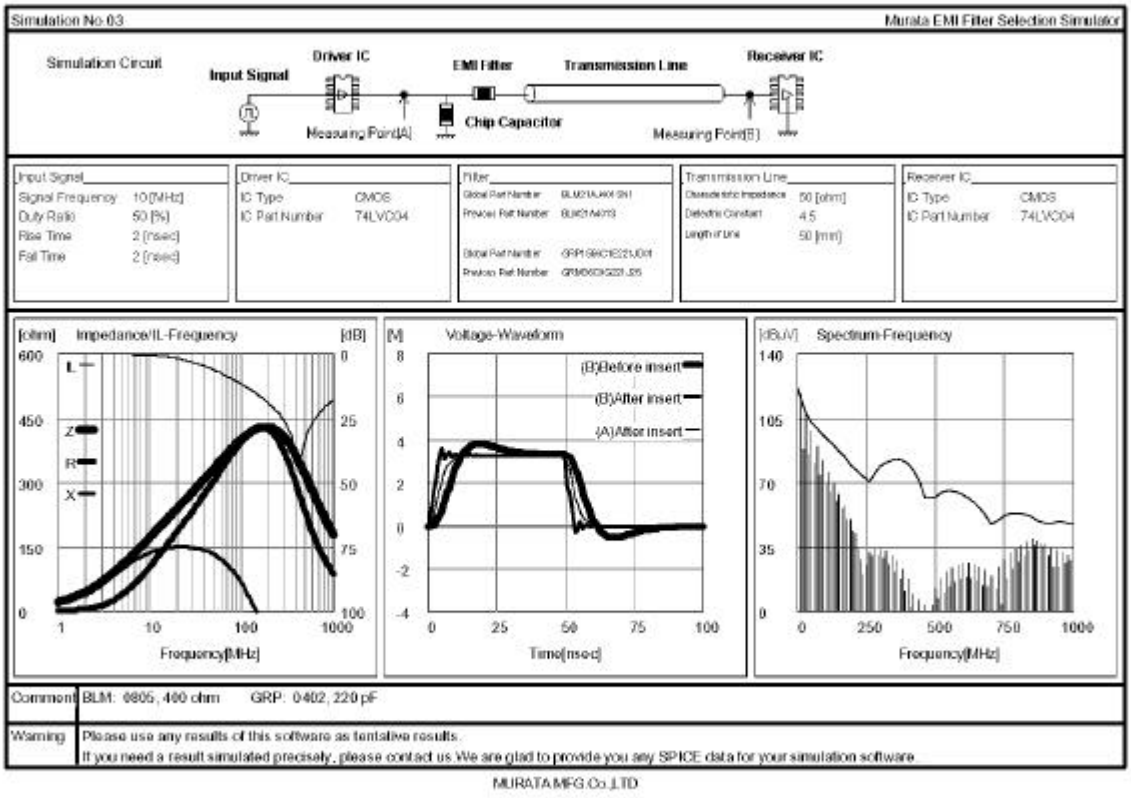

Figure 6: Example Filter Simulation for the I/O Lines

11 References

As of October 8, 2004, this is still to be determined.

12 MIB Drawing List

As of October 8, 2004, this is still to be determined.