

# MSC8122PFC-HV Packet Telephony Farm Card User's Guide

The high-volume MSC8122 packet telephony farm card (MSC8122PFC-HV) is a PCI telephony mezzanine card (PTMC) for evaluating and developing media gateway products. This card, which uses the StarCore™ MSC8122 16-bit fixed-point DSP devices, demonstrates a cost versus performance optimized implementation. The design can become a subsystem component of a larger telecommunications product.

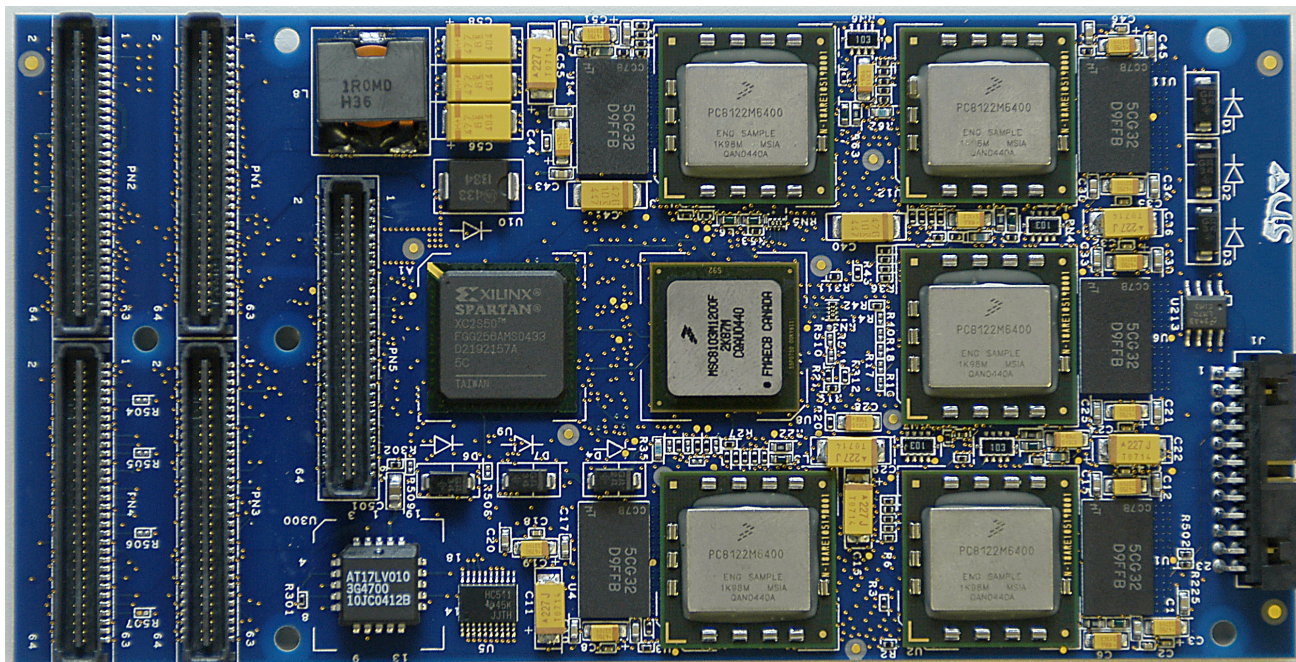
The MSC8122PFC-HV DSP farm consists of five MSC8122 devices running at up to 500 MHz and one MSC8103 device to aggregate the data to/from the DSP farm. Each MSC8122 device has two associated 2M × 32 (8 MB) SDRAM, giving 16 MB of 64-bit wide memory. The MSC8103 aggregator has an optional 4M × 8 (4 MB) flash device to store boot code.

The MSC8122PFC-HV (see **Figure 1**) interfaces with a baseboard platform via its PTMC site. The PTMC is a PCI mezzanine card (PMC) module that conforms to the PMC standard for PN1 and PN2 but uses PN3 and PN4 to support a variety of telecommunications interfaces. The PTMC site on the PFC is configured to meet the PT3MC standard, a subset of the PTMC specification that supports UTOPIA, Ethernet reduced media-independent interface (RMII), and computer telephony bus interfaces on PN3/4. An optional fifth connector (PN5) supports an MII interface to the MSC8103 device. PN5 is a proprietary connector, effectively supporting an enhanced PTMC that is backward-compatible with existing PTMCs.

## CONTENTS

<b>1</b>	Packet Telephony Development Kit .....	3
<b>2</b>	Getting Started with the MSC8122PFC-HV .....	4
<b>2.1</b>	Board Configuration Options .....	5
<b>2.2</b>	JTAG Chain Configuration .....	11
<b>2.3</b>	Optional Flash Memory .....	12
<b>2.4</b>	Probing Board Voltages .....	12
<b>3</b>	MSC81x2PFC-HV Hardware Components .....	13
<b>3.1</b>	MSC8103 Aggregator .....	13
<b>3.2</b>	MSC8122 DSP Farm .....	16
<b>4</b>	FPGA Firmware .....	18
<b>4.1</b>	PT3MC-Compliant FPGA Firmware .....	18
<b>4.2</b>	PDK-Compliant FPGA Firmware .....	20
<b>4.3</b>	FPGA Registers .....	25
<b>4.4</b>	PCI Bus Interface .....	32

Data is transferred primarily through 10/100 Mbps Ethernet (MII or RMII interfaces) or UTOPIA and a computer telephony TDM bus through the PTMC connectors. An I<sup>2</sup>C management interface is available through the PTMC PN1 connector. Additional I/O interfaces include HDI16/PCI, serial port, and EOnCE JTAG TAP for debug. The MSC8122PFC-HV interfaces with Freescale packet telephony enhanced PTMC baseboards such as the PDK demonstration system, as well as with standard customer PTMC type III baseboards.



**Figure 1.** MSC1822 Packet Telephony Farm Card (MSC8122PFC-HV)

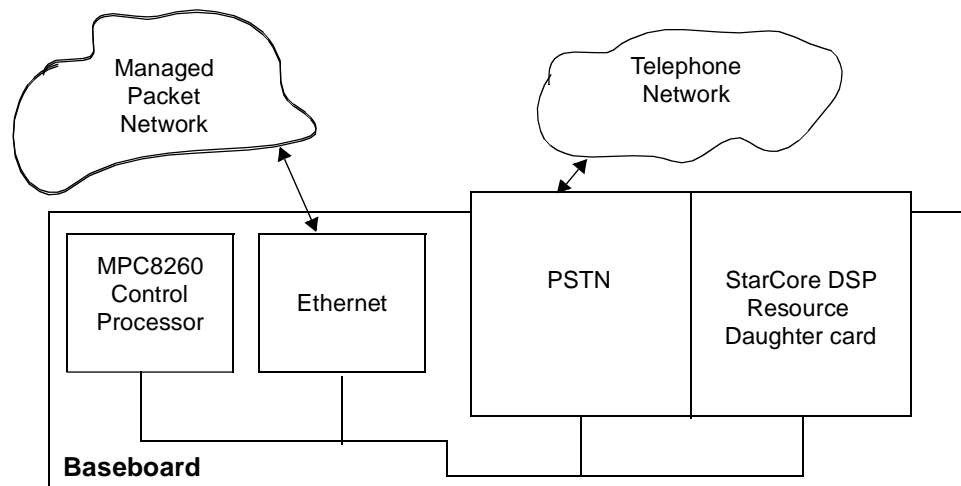
Features of the MSC8102PFC-HV are as follows:

- MSC8122PFC-HV platform:
  - Digital support for up to 672 channels.
  - PTMC Type 3 (PT3MC)-compliant card for interfacing to standard subsystems.
  - Optional PN5 connector to allow integration with the Freescale PDK.
- MSC8103 aggregator DSP:
  - 1 RMII fast Ethernet via a PN3 PTMC interface.
  - 1 MII/RMII fast Ethernet or UTOPIA via PN4 PTMC interface.
  - 1 MII fast Ethernet via PN5 (channel shared with PN3 RMII).
  - Host control (over HDI16) via PTMC PCI standard interface.
  - 32-bit asynchronous 60x bus interface to the MSC8122 DSI ports for on-board data distribution to the DSP farm.
  - 4 MB of Flash memory for system bootstrap.
  - Control over the programming sequence of the FPGA.

- MSC8122 farm consisting of five MSC8122 DSPs:
  - Standard CT-Bus TDM via the PN3 PTMC interface.
  - 32-bit DSI asynchronous slave port interfacing the MSC8103 60x bus.
  - 16 MB of 64-bit wide SDRAM.
- FPGA:
  - Direct connection or PCI interface from the host to the MSC8103 host port (HDI16) interface.
  - MII-to-RMII conversion for PN3 PTMC (and non-standard RMII on PN4).
  - CT-bus back-up clock and frame signal selection.
  - DSP  $\overline{\text{HRESET}}$  control.
- Debug:
  - Chained DSP EOnCE port with optional jumpers so that individual MSC8122 farm DSPs can be depopulated.
  - DSP EOnCE chain on the J1 debug connector.
  - Jumper to connect DSP EOnCE chain to the PN1/PN2 PTMC interface.
  - FPGA configuration EEPROM ISP through the J1 debug connector.
  - Boot mode option signals on the J1 debug connector.
  - J1 debug connector in envelope provided by the PTMC specification.

## 1 Packet Telephony Development Kit

The Packet Telephony Development kit (PDK) is a Freescale platform for evaluating and developing voice-over packet applications. The PDK has an MPC8260 host network processor that runs Linux, StarCore DSP resource cards that run DSP code, and a public switched telephone network (PSTN) card with interfaces such as E1/T1 and analog telephone lines (see **Figure 2**).



**Figure 2.** Components of the Packet Telephony Development Kit (PTK)

The documentation for the kit components is listed in **Table 1**. Reference documents for the MSC8122PFC-HV are listed in **Table 2**.

**Table 1.** PTK Components and Their Associated Documents

Component	Document	Document ID
Baseboard	<i>Packet Development Kit Baseboard Hardware User's Guide</i>	PTKITBASEUG
MPC8260 control processor	<i>MPC8260 PowerQUICC II™ Family Reference Manual</i> (Available at the website listed on the back page of this user's guide.)	MPC8260UM
PSTN card	<i>Packet Development Kit PSTN Mezzanine User's Guide</i>	PTKITPSTNUG
StarCore DSP resource daughter card	<ul style="list-style-type: none"> <li>• <i>MSC8102 Packet Telephony Farm Card (MSC8102PFC) User's Guide</i></li> <li>• <i>MSC8101 Packet Telephony Farm Card (MSC8101PFC) User's Guide</i></li> <li>• <i>MSC8122 Packet Telephony Farm Card (MSC8122PFC-HV) User's Guide</i></li> </ul>	PTKIT8102UG PTKIT8101UG PTKIT8122UG
MSC8122 processor	<i>MSC8122 Reference Manual</i> and other MSC8122 documentation are located at the web site listed on the back cover of this user's guide.	MSC8122RM
MSC8103 processor	<i>MSC8103 Reference Manual</i> and other MSC8103 documentation are located at the web site listed on the back cover of this user's guide.	MSC8103RM
Software	<i>Packet Telephony Development Kit Software User's Guide</i>	PTKITSOFTUG

**Caution:** The packet telephony development kit includes open-construction printed circuit boards that contain static-sensitive components. These boards are subject to damage from electrostatic discharge (ESD). To prevent such damage, you must use static-safe work surfaces and grounding straps, as defined in ANSI/EOS/ESD S6.1 and ANSI/EOS/ESD S4.1. All handling of these boards must be in accordance with ANSI/EAI 625.

**Table 2.** Reference Documents

Document	Revision	Date	Document ID
Standard Physical and Environmental layers for PCI Mezzanine Cards: PMC	Draft 2.4	January 12, 2001	IEEE: P1386.1
Standard for a Common Mezzanine Card Family: CMC	Draft 2.4a	March 21, 2001	IEEE: P1386
CompactPCI PCI Telecom Mezzanine Card Specification	R1.0	April 11, 2001	PICMG 2.15
H.100 Hardware Compatibility Specification: CT Bus	1.0		H.100

## 2 Getting Started with the MSC8122PFC-HV

**Figure 2** shows the position of major components on the MSC8122PFC-HV. In compliance with the PMC specification, all heat generating components are placed on the side facing the base card. A goal of the specification is to contain heat within one card slot in a card rack. It seeks to limit heat transfer from one base card in a rack to the next. However, with the heat generating components placed as they are, heat dissipation through ambient air convection is hindered because of the limited air volume and non-optimal air movement path. Therefore, forced air movement is required to cool the MSC8122PFC-HV, especially when the PFC is mounted horizontally. The amount of air flow required depends on ambient air temperature, DSP work load, and PFC mounting orientation. Read the data sheets of all components on the board and design a system with adequate forced air flow over the PFC to comply with maximum junction temperature limits of all components. The major components of interest are the MSC8122 DSP devices when they perform heavy loads. An on-board temperature sensor aids with thermal management software.

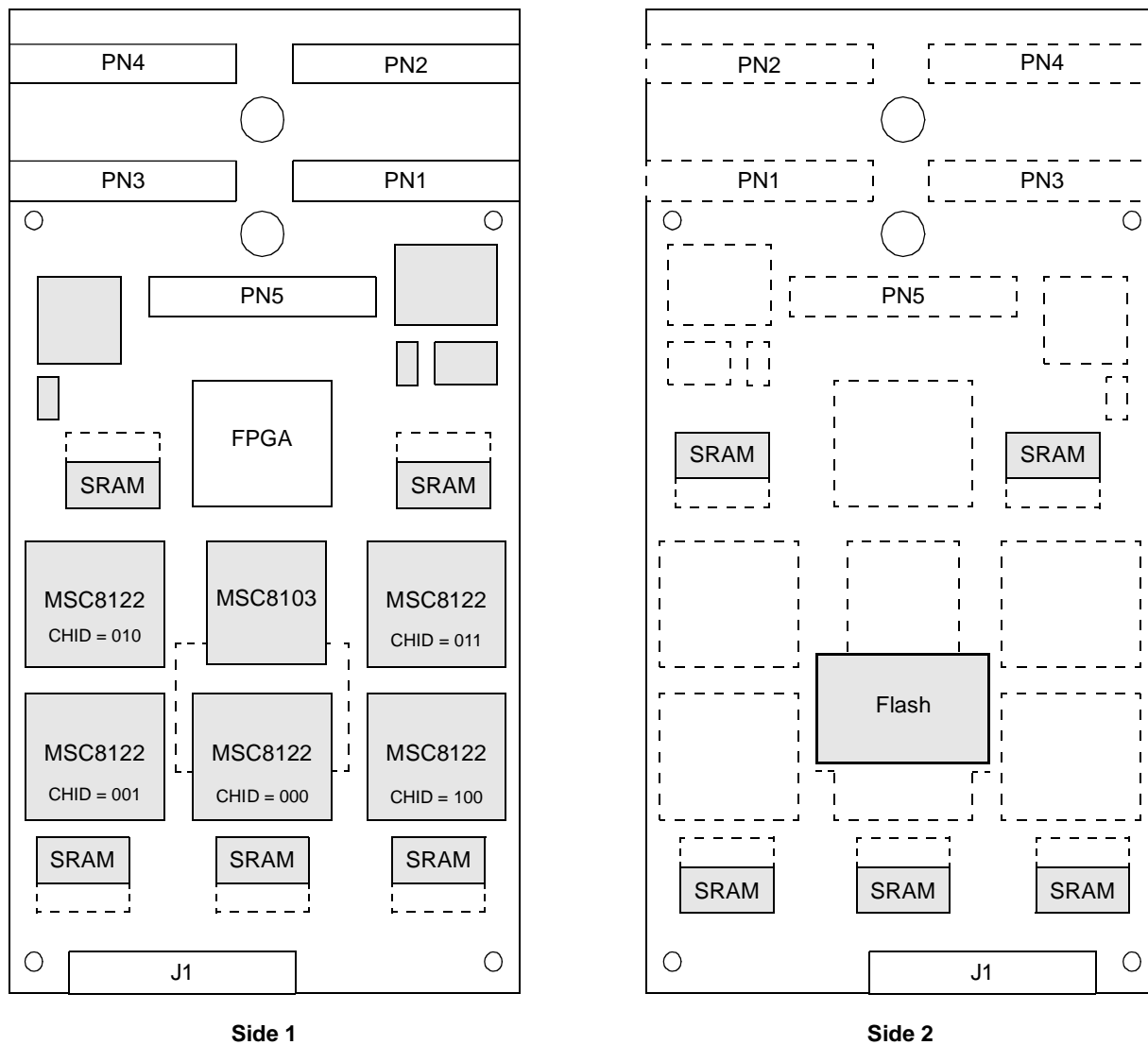


Figure 3. MSC8122PFC-HV Major Components

## 2.1 Board Configuration Options

The MSC8122PFC-HV board is designed to maximize cost versus performance. It is very important to understand the board configuration options before you use the board as a subsystem component of a larger telecommunications product. There are no switches on the MSC81x2PFC-HV. All options for board configuration are set by populating, changing, or removing surface mount resistors. A field-programmable gate array (FPGA) device is connected between the MSC8103 host DSP on the card and the PMC signals on PN1 and PN2. This feature promotes flexibility when you are choosing or designing a carrier card to mate with the MSC8122PFC-HV. The core power supply voltage level and sequencing are adjustable to meet the requirements for either device. The MODCK[1–2] boot option pins can be adjusted to allow different clocking modes.

Two different versions of the FPGA firmware allow the card to interface with either a Type 3 PCI mezzanine telecom carrier card (PT3CC) or the PDK base card. The FPGA configuration data is programmed into an on-board non-volatile serial EEPROM memory via the J1 connector. Contact a Freescale marketing representative for alternate binary files and related source files. **Table 3** through **Table 8** present the PN[1–5] signals and the J1 (debug) signals, respectively.

**Table 3. PN1 Signals**

Signal		Pin		Signal	
PDK	PT3MC			PT3MC	PDK
PMC_JTAG		1	2	NC	
GND		3	4	PCI_nINTA	NA
NC		5	6	NC	
NA	CMC_BUSMODE1	7	8	5V	
NC		9	10	NC	
GND		11	12	NC	
NA	PCI_CLK	13	14	GND	
GND		15	16	PCI_nGNT	NA
NA	PCI_nREQ	17	18	5V	
V I/O		19	20	PCI_AD31	NA
NA	PCI_AD28	21	22	PCI_AD27	NA
NA	PCI_AD25	23	24	GND	
GND		25	26	PCI_CBE3	NA
NA	PCI_AD22	27	28	PCI_AD21	NA
NA	PCI_AD19	29	30	5V	
V I/O		31	32	PCI_AD17	NA
NA	PCI_nFRAME	33	34	GND	
GND		35	36	PCI_nIRDY	NA
NA	PCI_nDEVSEL	37	38	5V	
GND		39	40	NC	
I2C_SCL		41	42	I2C_SDA	
NA	PCI_PAR	43	44	GND	
V I/O		45	46	PCI_AD15	NA
NA	PCI_AD12	47	48	PCI_AD11	NA
NA	PCI_AD9	49	50	5V	
GND		51	52	PCI_CBE0	NA
NA	PCI_AD6	53	54	PCI_AD5	NA
NA	PCI_AD4	55	56	GND	
V I/O		57	58	PCI_AD3	NA
NA	PCI_AD2	59	60	PCI_AD1	NA
NA	PCI_AD0	61	62	5V	
GND		63	64	NC	

**Notes:** 1. NC = Not connected.  
2. NA = Not applicable; connected but unused.

Table 4. PN2 Signals

Signal		Pin		Signal	
PDK	PT3MC			PT3MC	PDK
NC		1	2	PMC_nTRST	
PMC_TMS		3	4	PMC_TDO	
PMC_TDI		5	6	GND	
GND		7	8	PDK_TCK	
HDI16_HD0	NA	9	10	NA	HDI16_HA1
HDI16_HD1	CMC_BUSMODE2	11	12	3.3V	
PTMC_nRST	PCI_nRST	13	14	CMC_BUSMODE3	HDI16_HA2
3.3V		15	16	CMC_BUSMODE4	HDI16_HA3
HDI16_HD2	NA	17	18	GND	
HDI16_HD3	PCI_AD30	19	20	PCI_AD29	HDI16_HRW
GND		21	22	PCI_AD26	HDI16_HDS
HDI16_HD4	PCI_AD24	23	24	3.3V	
HDI16_HD5	PCI_IDSEL	25	26	PCI_AD23	NA
3.3V		27	28	PCI_AD20	NA
HDI16_HD6	PCI_AD18	29	30	GND	
HDI16_HD7	PCI_AD16	31	32	PCI_CBE2	NA
GND		33	34	NA	HDI16_HCS
HDI16_HD8	PCI_nTRDY	35	36	3.3V	
GND		37	38	PCI_nSTOP	NA
HDI16_HD9	PCI_nPERR	39	40	GND	
3.3V		41	42	PCI_nSERR	NA
HDI16_HD10	PCI_CBE1	43	44	GND	
HDI16_HD11	PCI_AD14	45	46	PCI_AD13	NA
NA	PCI_M66EN	47	48	PCI_AD10	HDI16_HA_DSP
HDI16_HD12	PCI_AD8	49	50	3.3V	
HDI16_HD13	PCI_AD7	51	52	NC	
3.3V		53	54	NC	
HDI16_HD14	NA	55	56	GND	
HDI16_HD15	NA	57	58	NA	
GND		59	60	NA	
HDI16_HA0	NA	61	62	3.3V	
GND		63	64	NA	

**Notes:**

1. NC = Not connected.
2. NA = Not applicable; connected but unused.

**Table 5. PN3 Signals**

Signal	Pin		Signal
MII_MDIO	1	2	GND
GND	3	4	SRX
MII_MDC	5	6	STX
RMII2_RXER	7	8	GND
PTMC_PTID2	9	10	RMII2_RXD0
NC	11	12	RMII2_RXD1
RMII_REFCLK	13	14	GND
GND	15	16	RMII2_TXD0
CT_FA	17	18	RMII2_TXD1
CT_FB	19	20	GND
PTMC_PTID0	21	22	RMII2_TXEN
NC	23	24	RMII2_CRS_DV
CT_C8A	25	26	GND
GND	27	28	CT_D19
CT_D18	29	30	CT_D17
CT_D16	31	32	GND
GND	33	34	NC
CT_D14	35	36	NC
CT_D12	37	38	GND
PTMC_PTENB	39	40	NC
NC	41	42	NC
CT_C8B	43	44	GND
GND	45	46	CT_D15
CT_D10	47	48	CT_D13
CT_D8	49	50	CT_D11
GND	51	52	CT_D9
CT_D6	53	54	CT_D7
CT_D4	55	56	GND
PTMC_PTID1	57	58	CT+D5
CT_D2	59	60	CT_D3
CT_D0	61	62	GND
GND	63	64	CT_D1

Table 6. PN4 Signals

Signal	Pin		Signal
UT_RXSOC	1	2	GND
GND	3	4	UT_TXADR4
UT_RXCLAV	5	6	UT_RXADR4
UT_TXADR3	7	8	GND
PTMC_USER0	9	10	GND
GND	11	12	NC
NC	13	14	GND
GND	15	16	UT_nTCENB
UT_RXADR3	17	18	UT_TXCLAV
UT_RXADR2	19	20	GND
PTMC_USER1	21	22	UT_nRXENB
GND	23	24	UT_TXADR2
UT_RXCLK	25	26	GND
GND	27	28	UT_RXADR1
UT_RXADR0	29	30	UT_TXADR1
UT_RXPRTY	31	32	GND
GND	33	34	UT_TXADR0
UT_RXD7	35	36	UT_TXPRTY
UT_RXD6	37	38	GND
PTMC_USER2	39	40	UT_TXD7
GND	41	42	UT_TXD6
UT_TXCLK	43	44	GND
GND	45	46	UT_TXD5
UT_RXD5	47	48	UT_TXD4
UT_RXD4	49	50	GND
GND	51	52	UT_TXD3
UT_RXD3	53	54	UT_TXD2
UT_RXD2	55	56	GND
PTMC_USER3	57	58	UT_TXD1
UT_RXD1	59	60	UT_TXD0
UT_RXD0	61	62	GND
GND	63	64	UT_TXSOC

**Table 7. PN5 Signals**

Signal	Pin		Signal
NC	1	2	NC
NC	3	4	NC
NC	5	6	NC
NC	7	8	NC
NC	9	10	NC
NC	11	12	GND
NC	13	14	NC
NC	15	16	NC
NC	17	18	NC
GND	19	20	MII2_TCLK
MII2_TXD0	21	22	NC
MII2_TXD1	23	24	MII2_RXDV
MII2_TXD2	25	26	MII2_RXD0
MII2_TXD3	27	28	MII2_RXD1
NC	29	30	MII2_RXD2
NC	31	32	GND
MII2_TXEN	33	34	MII2_RXD3
MII2_TXER	35	36	MII2_RXER
MII2_COL	37	38	MII2_CRS
NC	39	40	MII2_RCLK
NC	41	42	NC
NC	43	44	NC
NC	45	46	NC
NC	47	48	NC
NC	49	50	NC
NC	51	52	GND
NC	53	54	NC
NC	55	56	NC
NC	57	58	NC
GND	59	60	NC
NC	61	62	GND
NC	63	64	NC

**Table 8. J1 (Debug) Signals**

Signal	Pin		Signal
TDI	1	2	GND
TDO	3	4	GND
TCK	5	6	GND

**Table 8.** J1 (Debug) Signals (Continued)

Signal	Pin		Signal
FPGA_nSER_EN	7	8	KEY
HRESET	9	10	TMS
3.3V	11	12	FPGA_DIN
FPGA_CCLK	13	14	nTRST
SRX	15	16	STX
3.3V	17	18	A_RSTCONF
3.3V	19	20	A_BOOT
NC	21	22	NC
NC	23	24	NC

## 2.2 JTAG Chain Configuration

The JTAG chain on the card links all six DSPs. The chain is always connected to the J1 debug and programming connector. The board JTAG chain can also be connected to the PMC JTAG chain by populating resistors R209, R224, R287, R288, R269, and R244 (see schematic). The chain is organized as shown in **Table 9**.

**Table 9.** JTAG Chain Index

JTAG Chain Index	Device	
0	MSC8122 CHIP ID 4	SYNC
1		Core 3
2		Core 2
3		Core 1
4		Core 0
5	MSC8122 CHIP ID 3	SYNC
6		Core 3
7		Core 2
8		Core 1
9		Core 0
10	MSC8122 CHIP ID 2	SYNC
11		Core 3
12		Core 2
13		Core 1
14		Core 0
15	MSC8122 CHIP ID 1	SYNC
16		Core 3
17		Core 2
18		Core 1
19		Core 0

**Table 9.** JTAG Chain Index (Continued)

20	MSC8122 CHIP ID 0	SYNC
21		Core 3
22		Core 2
23		Core 1
24		Core 0
25	MSC8103	

## 2.3 Optional Flash Memory

An optional 4 MB non-volatile Flash memory device can be installed on the MSC8122PFC-HV card so that the card can boot without host control. To enable bootstrapping from reset, the Flash memory device is connected to  $\overline{CS0}$  and the signals described in **Table 10**. On the Flash memory device, the BYTE signal is pulled down for byte mode, which enables D[0–7] and tri-states D[8–14]. A1/D15 is used as the LSB address bit input.

**Table 10.** MSC8103 Flash Memory Signal Connections

MSC8103	Flash	Description
$\overline{CS0}$	$\overline{CS}$	Chip select
$\overline{PSDRAS}$	$\overline{OE}$	Output enable
$\overline{PSDDQM0}$	$\overline{WE}$	Write enable
A[7–30]	A[0–21]	Address
A31	A1/D15	Byte select
D[0–7]	D[0–7]	Byte wide data

By default, the MSC8103 attempts to read the boot data and program from the Flash memory device, even if it is not installed. The host should reset the MSC8103 and MSC8122 devices via the FPGA and change the boot mode if the system should not follow the default behavior. Consult the *MSC8103 Reference Manual* for information on the boot process and on FPGA control over DSP reset and boot mode. The MSC8122PFC-HV uses the same Flash memory (AM29LV320DB) as the MSC8102ADS, so you can use either the Metrowerks CodeWarrior™ or PFC-specific Flash programmer. Consult Freescale or the device manufacturer data sheet for additional details on programming Flash memory.

## 2.4 Probing Board Voltages

The MSC8122PFC-HV board requires five different voltages for proper operation, including 3.3 V, 5 V, MSC8122 core voltage, MSC8103 core voltage, and Xilinx FPGA core voltage. The MSC8103 and Xilinx core voltages are regulated from 3.3 V using 1 Amp MAX8869 linear regulators. The MSC8122 core voltage is generated from 5 V with a switching regulator circuit. **Figure 4** shows one ground and five positive voltage probe points that allow easy probing of all on board voltages.

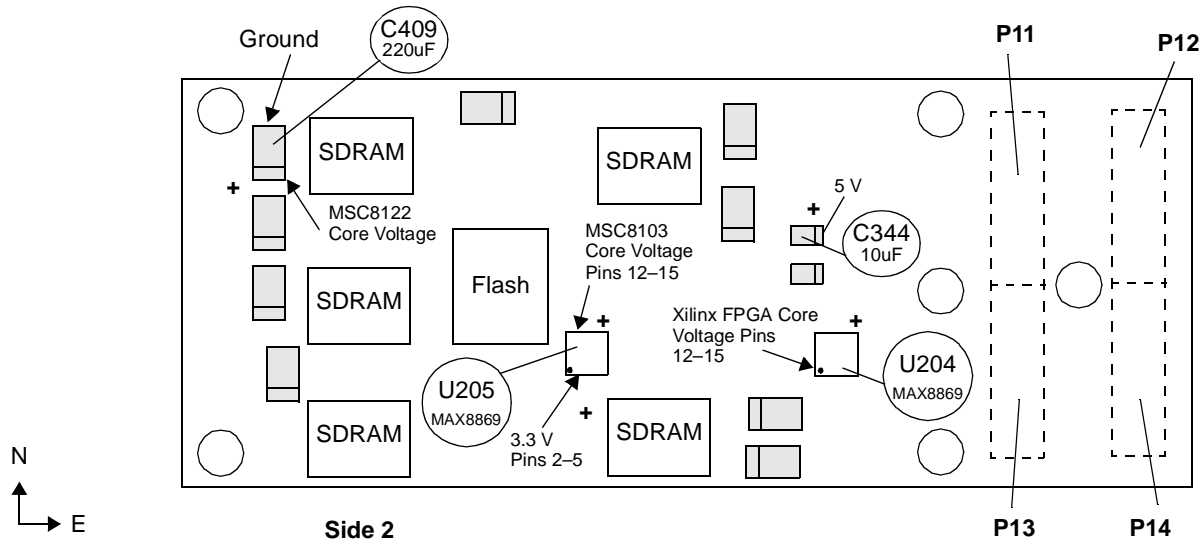


Figure 4. Board Voltage Probe Points

### 3 MSC81x2PFC-HV Hardware Components

As Figure 5 shows, the principal components of the MSC122PFC-HV board are the MSC8103 aggregator DSP and the five MSC8122 DSP devices. This section discusses these components and the buses that connect them.

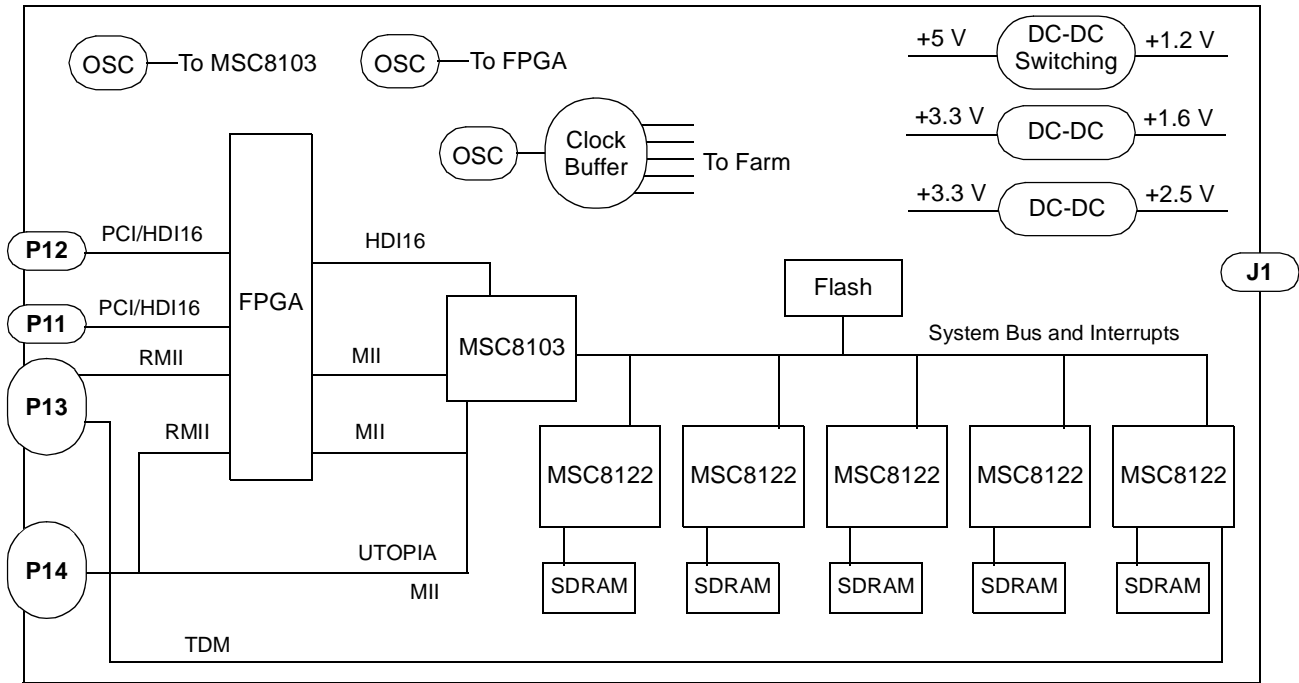


Figure 5. MSC122PFC-HV Block Diagram

#### 3.1 MSC8103 Aggregator

Under typical operating conditions, the MSC8103 is used to terminate ATM or 10/100BaseT Ethernet packet traffic from a host card via its PTMC interface. The subsequent data is placed into the MSC8103 internal SRAM. The data is then distributed to the MSC8122 farm for processing via the MSC8122 DSI port. After MSC8122

processing, the data is dispatched through the MSC8122 TDM interfaces to the CT bus. Data also flows in the opposite path. An FPGA connects the PTMC I/O pins on PN1 and PN2 to the HDI16 port on the MSC8103. This connection provides a slave interface to the MSC8103 on the PTMC or PDK data buses. The MSC8103 then connects to the optional Flash memory device and the MSC8122 DSP farm through a 60x system bus. MSC8103 system bus chip selects are connected as shown in **Table 11**.

**Table 11.** MSC8103 Chip Select Assignment

Signal	Peripheral
$\overline{\text{CS0}}$	Flash
$\overline{\text{CS3}}$	MSC8122 DSI Broadcast
$\overline{\text{CS4}}$	MSC8122 DSI Single

The MSC8103 interrupts are connected to the FPGA and to the DSP farm. Each DSP in the farm has an interrupt line from and another interrupt line to the MSC8103 host. All interrupt lines are pulled high with weak resistors. The interrupts are connected as shown in **Table 12**.

**Table 12.** MSC8103 Interrupt Signals

Signal	Description
$\overline{\text{IRQ1}}$	Interrupt from farm DSP 1
$\overline{\text{IRQ2}}$	Interrupt from FPGA (defined in FPGA)
$\overline{\text{IRQ4}}$	Interrupt from farm DSP 2
$\overline{\text{IRQ5}}$	Interrupt from farm DSP 3
$\overline{\text{IRQ6}}$	Interrupt from farm DSP 4
$\overline{\text{IRQ7}}$	Interrupt from farm DSP 5
PC4	Interrupt to farm DSP 1
PC5	Interrupt to farm DSP 2
PC22	Interrupt to farm DSP 3
PC23	Interrupt to farm DSP 4
PC24	Interrupt to farm DSP 5

### 3.1.1 HDI16 to FPGA

The MSC8103 HDI16 port connects directly to the FPGA. The FPGA is then connected to the PMC PN1/2 connectors. These connections allow creative implementation of slave interfaces to a base card.

### 3.1.2 System Bus to MSC8122 DSI

The direct slave interface (DSI) on each MSC8122 device gives the aggregator direct access to the MSC8122 internal memory space, including on-chip memory and registers. The DSI write buffer stores the address and the data of the accesses until they are performed. The external host can therefore perform multiple writes without waiting for those accesses to complete. Latencies that are typical during accesses to on-chip memories are greatly reduced by a DSI read prefetch mechanism. The DSI on each MSC8122 device connects to the external 60x system bus on the MSC8103 device. The same chip select is connected to all the DSI ports, but each device is given a

different CHIP ID via boot option pins. Each MSC8122 is accessible at different offsets from the base address associated with chip select 4.

**Table 13.** MSC8122 DSI Port Offsets on 60x Bus

DSI Device ID	DSI Base Offset
0	0x00000000
1	0x00200000
2	0x00400000
3	0x00600000
4	0x00800000

The DSI bus is configured for 32-bit asynchronous operation. Therefore, the MSC8103 should be configured to use a user-programmable machine (UPM) memory controller for single-chip access. The DSI are asynchronously controlled via the UPM signals listed in **Table 14**.

**Table 14.** MSC8103 60x Bus Signals to DSI Ports

MSC8103	MSC8122	Description
CS3	HBCS	Broadcast chip select
CS4	HCS	Individual addressable chip select
PSDDQM[0–3]	HWBS[0–3]	Byte lane strobes
PSDWE	HR/W	Read / write command
UPMWAIT	HTA	Transfer acknowledge (MSC8122 can pause the UPM state machine)
TBST	HBRST	Burst transfer command
A[7–10]	HCID[0–3]	DSI chip select address
A[11–29]	HA[11–29]	DSI transfer address
D[0–31]	HD[0–31]	DSI transfer data

All devices can be accessed simultaneously via the broadcast chip select, which is connected to chip select 3 on the MSC8103. This chip select should be configured to use the general-purpose chip select machine (GPCM) because the DSI ports do not drive the transfer acknowledge in this mode.

### 3.1.3 Ethernet RMII/MII

The MSC8103 has two fast communication controller (FCC) interfaces for packet transfers. You can configure them for two MII ports (FCC1 and FCC2) or an MII port (FCC1) and a UTOPIA port (FCC2). Both configurations are routed to the FPGA as depicted in **Figure 6**. The MII and UTOPIA pins are dual-function on the MSC8103. The UTOPIA-only pins connect directly to PN4. The dual-function UTOPIA and MII pins connect to both the FPGA and PN4. RMII signals, which are converted from MII signals in the FPGA, are routed to UTOPIA-only pins. In UTOPIA mode, the FPGA tri-states its RMII output pins. In RMII mode, the UTOPIA-only output pins from the MSC8103 are tri-stated. The FPGA then converts MII to RMII, and the RMII pins on PN4 are driven from the FPGA. Therefore, in RMII mode, both RMII and MII signals are present on PN4.

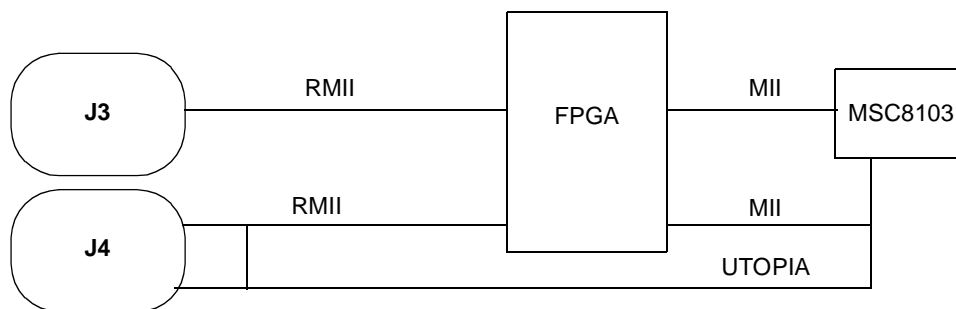


Figure 6. R/MII to PMC Connectors

### 3.1.4 Serial Port

The MSC8103 STX and SRX pins connect directly to the J1 debug connector. The signals at the connector are 0–3.3 V signals, not RS-232. A transceiver must be supplied to connect to a computer RS-232 serial port.

## 3.2 MSC8122 DSP Farm

The DSP farm consisting of five MSC8122 DSP devices is another main component on the MSC8122PFC-HV board. The remainder of this section considers the SDRAM devices connected to these DSPs and the CT bus for time-division multiplexing (TDM). Each MSC8122 device has two 8 MB 16-bit wide SDRAM devices attached. Following is a C language subroutine example of SDRAM initialization code for Metrowerks CodeWarrior:

```

/*****
HV_init_sdram() :
    Initialize the Bank 2 for 64-bit SDRAM.
    Return 0 on success, 1 on error.
*/
int HV_init_sdram(unsigned long base_address)
{
    Word64 zero64;
    Int x;
    msc8102_sys_regs_t* SysRegs = (msc8102_sys_regs_t*) SYSBUS_BASE;
    zero64.msb = 0;
    zero64.lsb = 0;

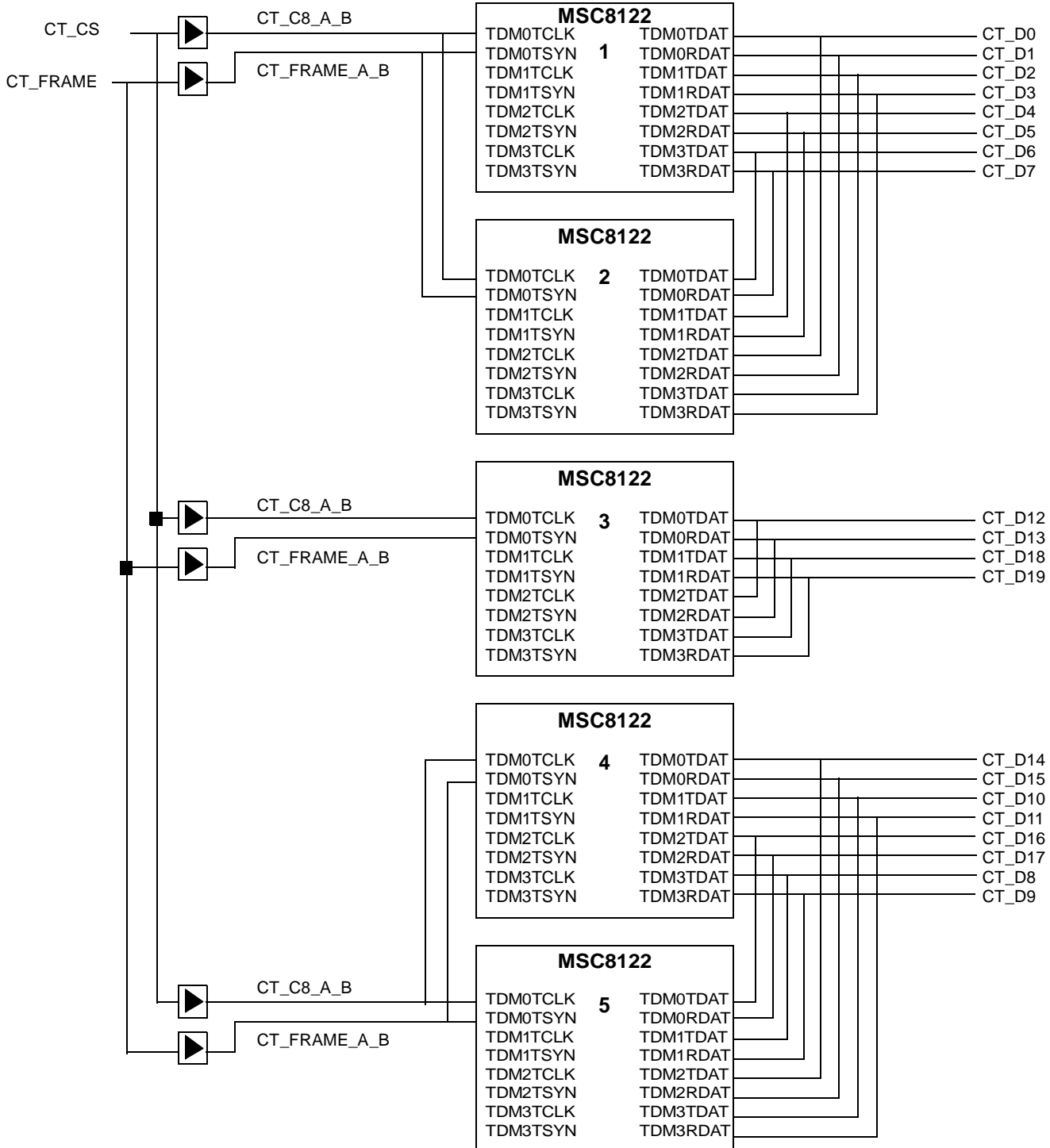
    SysRegs->MEMCBROR[2].OR = 0xFE002CC0;
    SysRegs->MEMCBROR[2].BR = base_address + 0x0041;

    SysRegs->PSDMR = 0x606B20A2; /* PSDMR- precharge */
    Write_64((Word64*) base_address, zero64);
    SysRegs->PSDMR = 0x486B2062; /* PSDMR- issue 8 CBR's */
    for (x = 0; x < 8; x++)
    { Write_64((Word64*) base_address, zero64); }
    SysRegs->PSDMR = 0x586B2062; /* PSDMR- mode register write */
    Write_64((Word64*) base_address, zero64);
    SysRegs->PSDMR = 0x406B2062; /* PSDMR- normal mode */
    SysRegs->PSRT = 0x2C;
    SysRegs->MPTPR = 0x2000;
    return 0;
}

```

### 3.2.1 CT Bus (TDM)

A standard CT bus is split into 32 streams of 128 time slots each, giving 4096 time slots. Each CT stream is uni-directional only, so two CT streams interface with an MSC8122 TDM. The MSC8122PFC-HV follows the PT3MC specification, which further restricts the number of streams to 20, or 10 TDM channels. MSC8122 devices are configured for four-pin operation with common clock and frame syncs for receive/transmit. The primary and backup frame and clock (CT\_F[A-B] and CT\_C8[A-B]) signals are routed to the FPGA. One set is selected and routed out to the DSP farm. The FPGA provides adequate buffering of the frame and clock signals for the DSP farm.



**Figure 7.** CT Bus Connections to DSP Farm  
**MSC8122PFC-HV Packet Telephony Farm Card User's Guide, Rev. 0**

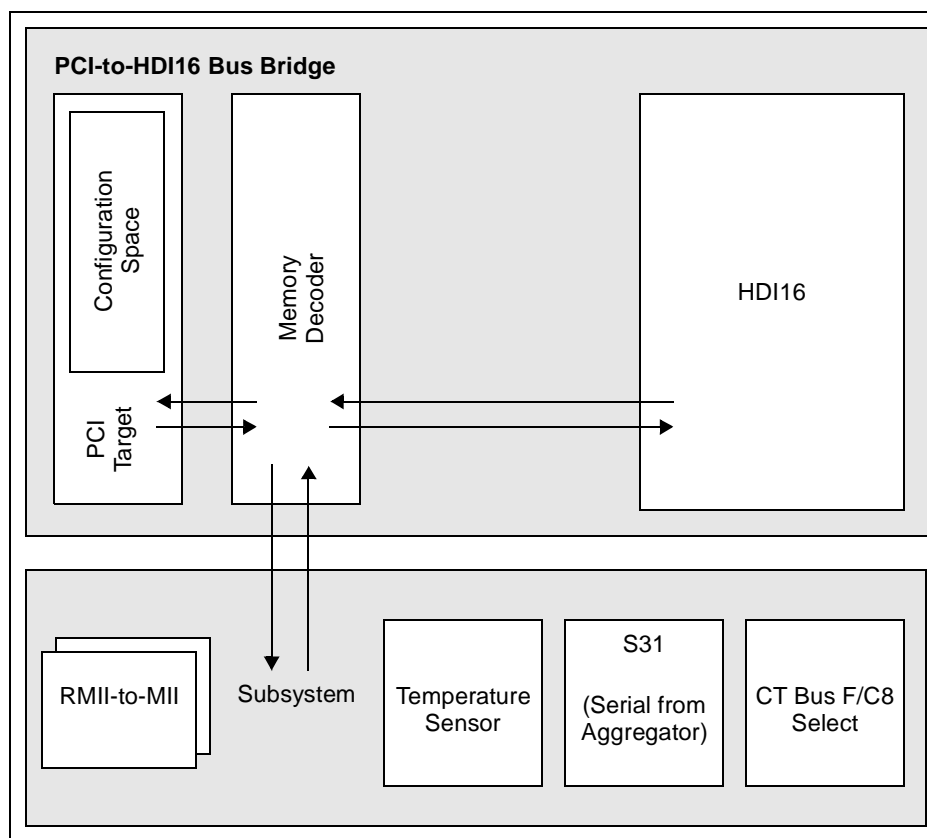
## 4 FPGA Firmware

There are two versions of the FPGA architecture:

- Standard PT3MC FPGA for interfacing with a 32-bit PCI, which includes modules for the PCI-to-HDI16 bus bridge and the collection of subsystem capabilities.
- Non-standard PDK FPGA for interfacing with the HDI16 (pass-through) on the packet telephony development kit (PDK) local bus. The PDK FPGA architecture is much less complex and reuses many PT3MC modules. The PDK FPGA replaces the bus bridge with an HDI16 pass-through module but maintains all the subsystem capabilities of the PT3MC FPGA.

### 4.1 PT3MC-Compliant FPGA Firmware

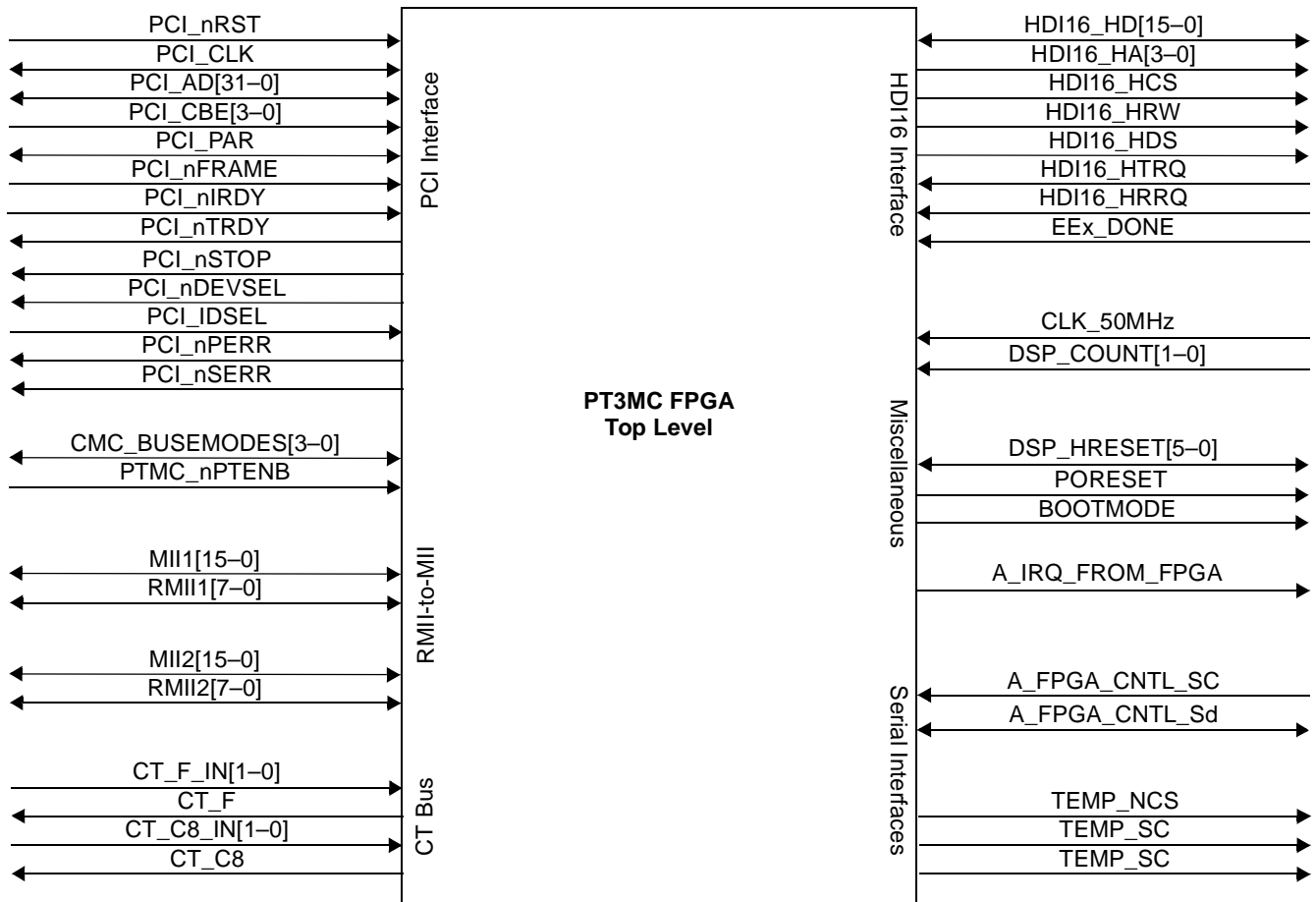
The PT3MC FPGA connects the MSC8122PFC-HV with a standard PT3CC through a 33 MHz, 32-bit, 3.3 V PCI, and through RMI, UTOPIA, and the CT bus (TDM). To support the PT3CC interface, the FPGA contains a PCI target, RMI-to-MII converters, and a CT bus back-up frame and clock selector. The hardware description language (HDL) for the FPGA is written in Verilog and has a hierarchical design. The top-level design is divided into the PCI-to-HDI16 bus bridge and the subsystem as shown in **Figure 8**.



**Figure 8.** PT3MC FPGA Block Diagram

The main components of the PCI-to-HDI16 bus bridge are the PCI target module, memory decoder module, and HDI16 module. The PCI target connects directly to the PCI bus of the PT3CC, handles PCI target configuration, and converts PCI transfers into internal bus transfers. The memory decoder splits the internal address space into HDI16 access and subsystem control register access. The HDI16 module controls the HDI16 interface and converts internal bus transactions to HDI16 bus transactions. The subsystem combines all modules that handle subsystem

functions including RMII-to-MII conversion, temperature sensor interfacing, aggregator serial interfacing, and CT bus back-up signal selection. This module contains the control and status registers to configure, control, and interface the subsystem modules. A control register in the subsystem module also controls the DSP  $\overline{\text{HRESET}}$  commands. The PT3MC FPGA top-level HDL module combines the PCI-to-HDI16 bus bridge and the subsystem module into one. The two modules connect via an internal register interface for PCI transfers to and from the subsystem control and status registers. This module also contains the logic for the CMC bus mode and PTMC PTENB signals. The I/O signals for this module are illustrated in **Figure 9**.



**Figure 9.** FPGA Top-Level Module Signals

FPGA internal space is split into subsystem registers, scratch pad RAM, and HDI16 access space. The subsystem register space contains registers for modules including the temperature sensor or RMII-to-MII converters. Scratch pad RAM is 32-bit general-purpose RAM. HDI16 data transfers are handled directly on the HDI16 interface and can access the HDI16 data, control, status, and reset registers. The memory map, as viewed by the host, is offset from the PCI base address assigned to the device. The memory map of the internal space is shown in **Table 15**.

All transfers through the PCI interface must be 32-bit word aligned. A target abort is issued to any transfer not conforming to this rule. Transactions use all 32 bits, and the least significant byte enable is used to select the entire 32-bit word. Transactions with HDI16 access space use only the least significant 16 bits. The unused most significant 16 bits should be cleared during writes and is cleared during reads.

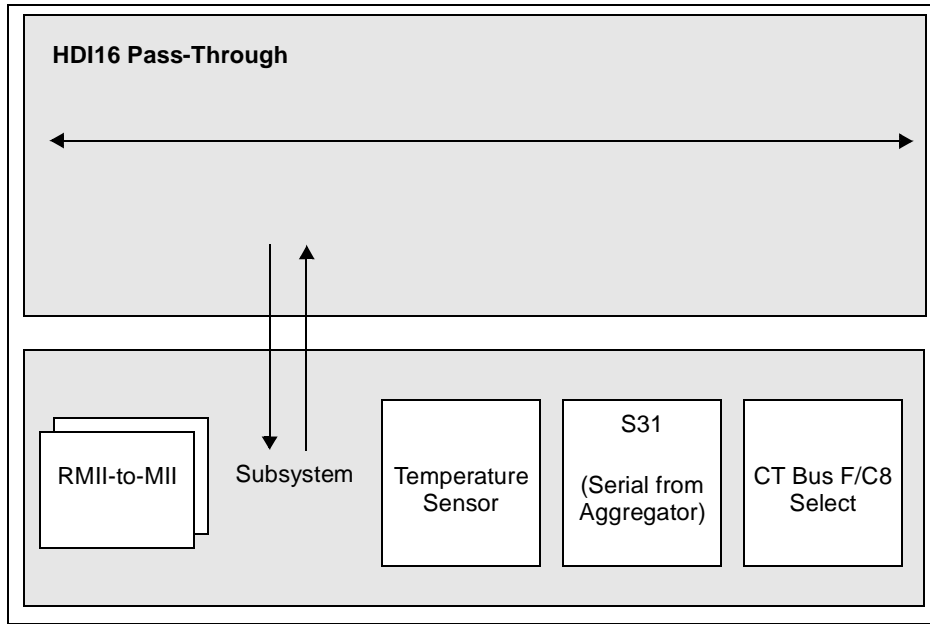
**Table 15. PT3MC FPGA Memory Map**

Size	Byte Address Offset	Interface/Register
	0x0000–0x03FF	Register interface to subsystem module
32 Bits	0x0000	Host Control/Status Register (HCSR)
16 Bits	0x0004	Temperature Sensor Output Register (TSOR)
N/A	0x0008	Reserved
16 Bits	0x000C	RMII Control Register (RCR)
16 Bits	0x0010	Aggregator Command Register (ACR)
16 Bits	0x0014	$\overline{\text{HRESET}}$ Status Register (HSR)
16 Bits	0x001C	FPGA Version Register (FVR)
32 Bits	0x0020–0x03FF	Scratch pad RAM
	0x0400–0x043F	Direct to MSC8103 HDI16 registers
16 Bits	0x0400	HDI16 ICR
16 Bits	0x0404	HDI16 CVR
16 Bits	0x0408	HDI16 ISR
16 Bits	0x0410	HDI16 RX3/TX3
16 Bits	0x0414	HDI16 RX2/TX2
16 Bits	0x0418	HDI16 RX1/TX1
16 Bits	0x041C	HDI16 RX0/TX0
16 Bits	0x0420	HDI16 RST3
16 Bits	0x0424	HDI16 RST2
16 Bits	0x0428	HDI16 RST1
16 Bits	0x042C	HDI16 RST0
<b>Note:</b> Address ranges are given in byte alignment, but accesses must be performed in 32-bit widths and alignment.		

## 4.2 PDK-Compliant FPGA Firmware

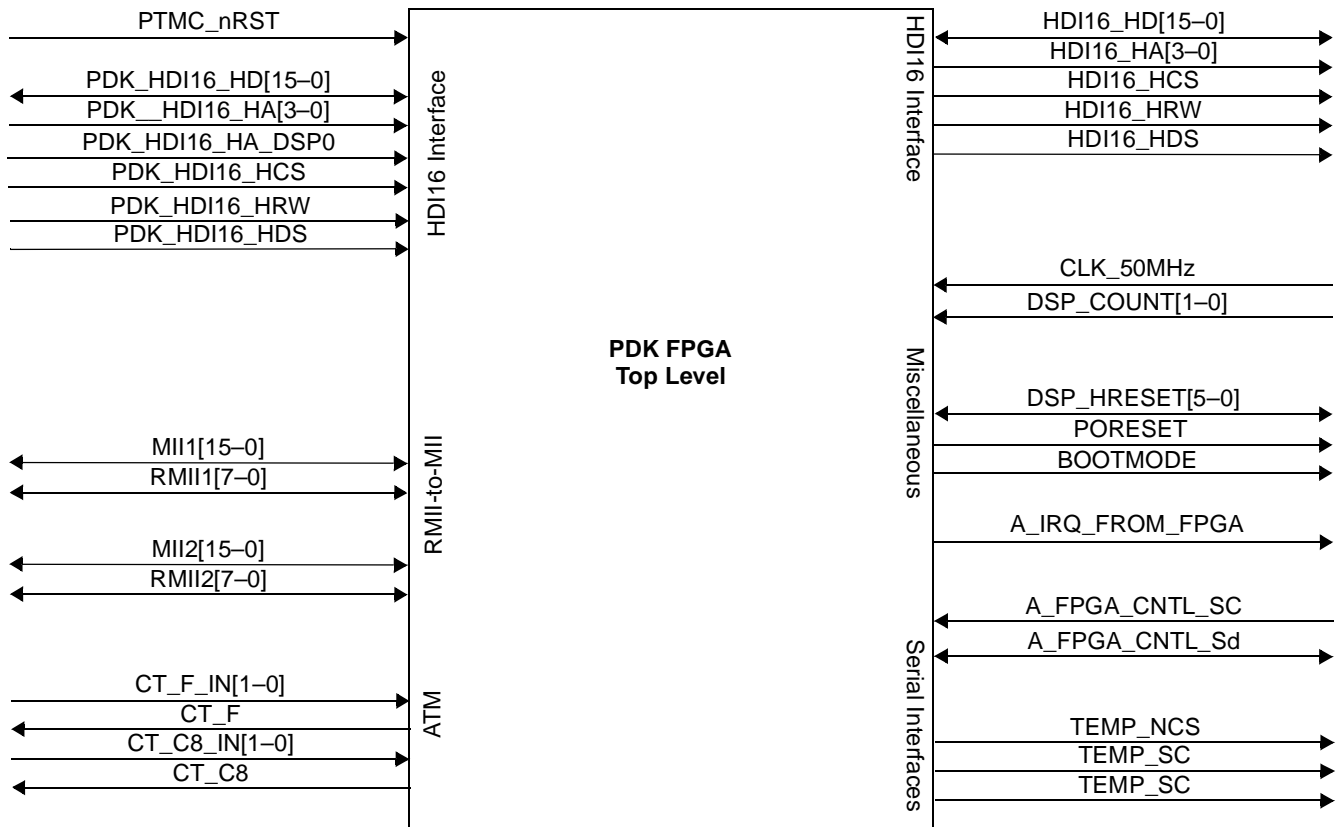
The PDK FPGA connects the MSC8122PFC-HV with the packet telephony development kit (PDK) through the HDI16 over the MPC8XXX local bus, MII, and CT Bus (TDM). For future expansion, the PDK FPGA also contains two RMII-to-MII converter modules. The HDI16 pass-through module uses the PDK\_HDI16\_HA\_DSP0 input to select either aggregator HDI16 transfers or internal subsystem register transfers. If the signal is low during a transfer cycle, the module performs an aggregator DSP HDI16 transfer. If the signal is high during a transfer cycle, the module performs an internal subsystem register transfer.

The main difference between the subsystem module in the PT3MC FPGA version and that in the PDK FPGA version is the register transfer interface. The PT3MC FPGA version has a 32-bit data, 8-bit address interface, where the PDK FPGA version has a 16-bit data, 4-bit address interface. Therefore, the subsystem registers in the PDK FPGA version are partitioned into low and high 16-bit words, accessed individually through the register transfer interface. Also, the internal scratch pad RAM is not included in the PDK FPGA version due to a lack of addressing space.



**Figure 10.** PDK FPGA Block Diagram

The PDK FPGA top-level HDL module combines both the HDI16 pass-through and the subsystem module into one. The two modules are connected via an internal register interface for PDK local bus transfers and from the subsystem control and status registers. The I/O signals for this module are illustrated in **Figure 11**.



**Figure 11.** PDK FPGA Top-Level Module Signals

The internal space of the FPGA is partitioned into subsystem register and HDI16 direct access space. The subsystem register space contains registers that interconnect the subsystem modules, such as the temperature sensor or RMI to MII converters. HDI16 direct access space transfers perform transactions directly on the HDI16 interface of the aggregator DSP.

The subsystem registers of the PDK FPGA are identical to the registers of the PT3MC FPGA. However, the registers are little-endian (bit 31 is the MSB). On the PCI bus, the PDK uses big-endian (bit 0 is the MSB). The FPGA uses the same bit masks on both endian types by aligning bit magnitudes on 16-bit aligned words. Because the data path to the PDK host processor is only 16-bits wide, the subsystem registers are partitioned into low and high 16-bit words. Therefore, two accesses must be completed to read or write an entire subsystem register, one for each half. The PDK FPGA memory map is offset from the base address assigned by the host processor to the PDK\_HDI16\_HCS chip select signal. **Table 16** lists the base address offsets for the registers. The least significant word of each register is accessed by a 16-bit transfer to the base address of the register, and the most significant word is accessed by a 16-bit transfer to the base address plus a 2-byte offset.

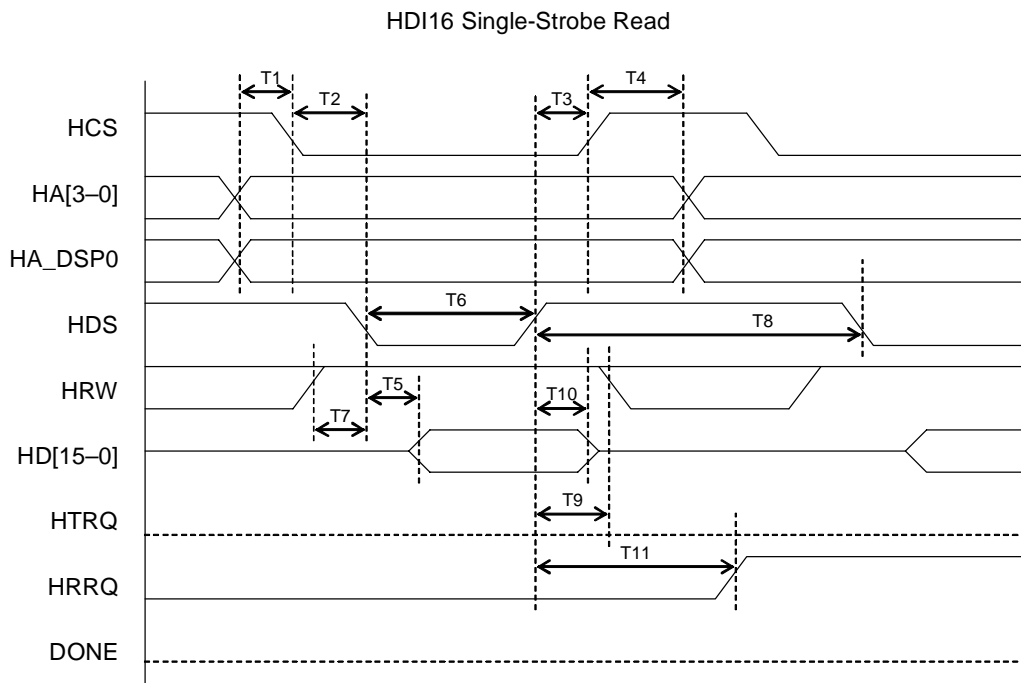
**Table 16.** PDK FPGA Memory Map

Byte Address Offset	Interface
0x00000–0x07FFF	Direct to HDI16 Registers
0x00000	HDI16 ICR
0x00002	HDI16 CVR
0x00004	HDI16 ISR
0x02000	HDI16 RX3/TX3
0x02002	HDI16 RX2/TX2
0x02004	HDI16 RX1/TX1
0x02006	HDI16 RX0/TX0
0x04000	HDI16 RST3
0x04002	HDI16 RST2
0x04004	HDI16 RST1
0x04006	HDI16 RST0
0x38000–0x3FFFF	Register interface to subsystem module
0x38000	Host Control/Status Register (HCSR)
0x38004	Temperature Sensor Output Register (TSOR)
0x3A000	Reserved
0x3A004	RMI Control Register (RCR)
0x3C000	Aggregator Command Register (ACR)
0x3C004	HRESET Status Register (HSR)
0x3E004	FPGA Version Register (least significant 16 bits)
0x3E006	16-bit Scratch Pad Register
<b>Note:</b> Address ranges are given in byte width, but accesses must be performed in 16-bit widths and alignment.	

The host processor on the PDK platform interacts with the FPGA through its local bus. The FPGA presents an HDI16-type interface to the host processor, with the addition of a PDK\_HA\_DSP0 signal that addresses the aggregator or FPGA internal registers. The FPGA HDI16 interface is preconfigured for single-strobe, active low operation. Read data cycles on the FPGA HDI16 interface follows the same waveform pattern as the HDI16 interface on the aggregator DSP but with altered timing requirements. The timing requirements have been increased where necessary to allow for propagation through the FPGA logic cell structures.

**Table 17.** HDI16 Read Timing Parameters

Label	Description	Constraint	Unit
Tc	One aggregator DSP clock period	N/A	ns
T1	Address set-up to chip select assertion	> 5.0	
T2	Address and chip select decoding delay to data strobe assertion	> 5.0	
T3	Chip select hold time after data strobe release	0	
T4	Address hold time after chip select release	> 5.0	
T5	Data strobe to valid data present	> (2.0 x Tc) + 15.0	
T6	Data strobe minimum assertion time	> (1.5 x Tc) + 5.0	
T7	Read/Write signal setup to data strobe assertion	> 5.0	
T8	Minimum data strobe deassertion width	> (2.5 x Tc) + 5.0	
T9	Read/Write signal hold time after data strobe release	> 5.0	
T10	Data valid after data strobe release	> 5.0	
T11	Data strobe release to host request release	> (3.5 x Tc) + 15.0	

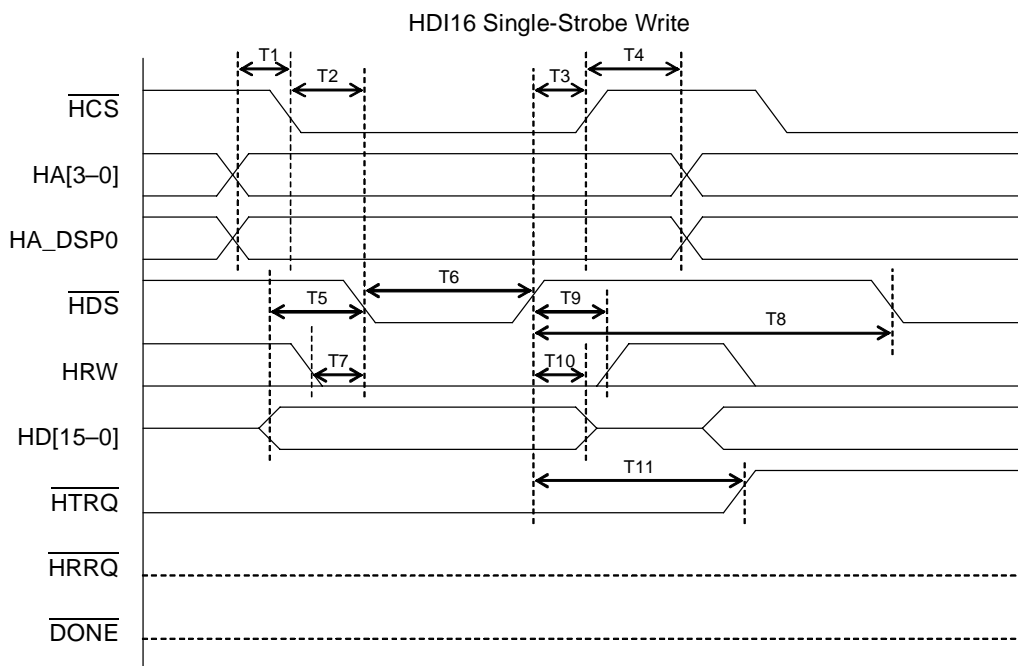


**Figure 12.** HDI16 Read Waveform

Write data cycles on the FPGA HDI16 interface have the same waveform pattern as the HDI16 interface on the aggregator DSP, but the timing requirements are increased to allow propagation through the FPGA logic cell structures. Also, there are differing latch requirements for FPGA internal registers (see **Table 18** and **Figure 13**). The DSP HDI16 interface latches data on the rising edge (release) of data strobe (HDS), whereas the FPGA latches data on the falling edge (assertion) of HDS. Therefore, data must be valid before the assertion of HDS and remain valid until after the release of HDS.

**Table 18.** HDI16 Write Timing Parameters

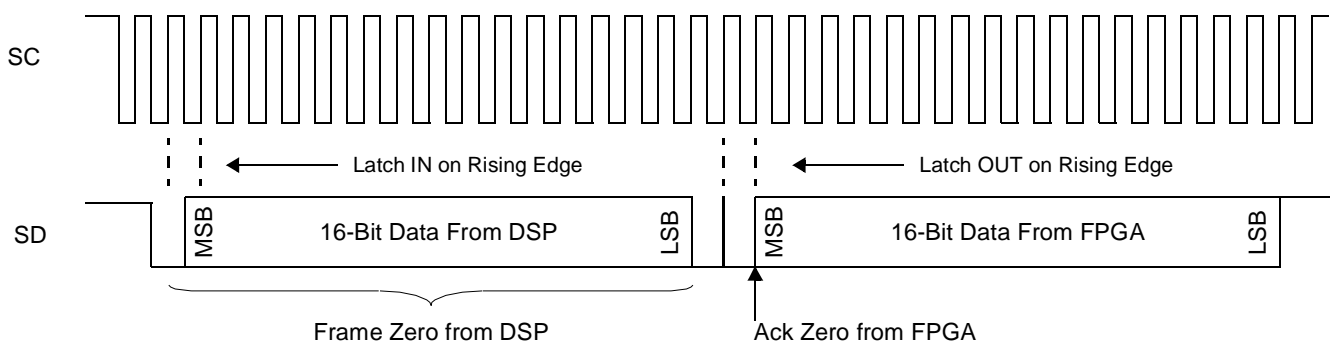
Label	Description	Constraint	Unit
Tc	One aggregator DSP clock period	N/A	ns
T1	Address set-up to chip select assertion	> 5.0	
T2	Address and chip select decoding delay to data strobe assertion	> 5.0	
T3	Chip select hold time after data strobe release	0	
T4	Address hold time after chip select release	> 5.0	
T5	Valid data present to data strobe assertion	> 10.0	
T6	Data strobe minimum assertion time	> (1.5 x Tc) + 5.0	
T7	Read/Write signal setup to data strobe assertion	> 5.0	
T8	Minimum data strobe deassertion width	> (2.5 x Tc) + 5.0	
T9	Read/Write signal hold time after data strobe release	> 5.0	
T10	Data valid after data strobe release	> 5.0	
T11	Data strobe release to host request release	> (3.0 x Tc) + 15.0	



**Figure 13.** HDI16 Write Waveform

## 4.2.1 Aggregator Serial Interface

A simple synchronous serial interface provides a control and status data path between the aggregator DSP and the FPGA. This interface transfers 16 data bits to the FPGA from the DSP and allows the DSP to read back 16 data bits. The 16-bit data from the DSP is latched into the Aggregator Command Register (ACR) (see [page 30](#)) on the rising edge of SC during the second framing zero from the DSP. The returned 16-bit data from the FPGA is the mirror of the ACR, with the exception that the  $\overline{\text{HRESET}}[4-0]$  bits indicate the true status of the  $\overline{\text{HRESET}}$  lines to the DSP farm, taken from the  $\overline{\text{HRESET}}$  Status Register (HSR). The transfer of data between the DSP and the FPGA must follow the simple frame format shown in [Figure 14](#). There are two signals, Serial Data (SD) and Serial Clock (SC). SD is wire-and with a pull-up, and SC is actively driven by the DSP only. Data is clocked into the FPGA on the rising edge of SC, starting with the first framing zero until the second framing zero. A frame starts with a zero, then the 16-bit data (MSB first), then another zero, all clocked at the rising edge of SC. The FPGA does not start latching data until it receives the first zero, and it triggers a framing error condition, resetting the interface if it does not receive the second framing zero after the 16-bit data.



**Figure 14.** Aggregator Serial Interface Waveform

After the second framing zero is clocked, the FPGA begins to drive SD low if no framing error is triggered. The DSP should release SD and then sample SD to check for the ACK zero from the FPGA. If ACK zero is sent, the FPGA shifts the return data out onto SD with each rising edge of SC. The DSP should sample the return data after each rising edge of SC until all 16 bits are received. The FPGA then releases SD and begins to search for the next beginning frame zero from the DSP on each rising edge of SC. To reset the interface from the DSP, release SD and clock SC at least 18 times. Nothing happens if the interface is already reset. If the FPGA is driving SD, any remaining return data clocks out. If there is a partially unfinished transfer from the DSP to the FPGA, a reset triggers a framing error and resets the interface. The ACR is untouched under all these conditions. To clear the ACR, perform a complete transfer with all zero data to the FPGA.

## 4.3 FPGA Registers

The FPGA registers discussed in this section are as follows:

- Host Control/Status Register (HCSR), [page 26](#).
- Temperature Sensor Output Register (TSOR), [page 28](#).
- RMI Control Register (RCR), [page 29](#).
- Aggregator Command Register (ACR), [page 30](#).
- $\overline{\text{HRESET}}$  Status Register (HSR), [page 31](#).
- FPGA Version Register (FVR), [page 32](#).
- Scratch Pad RAM (SPR), [page 32](#).

HCSR		Host Control/Status Register														Byte Offset 0x0	
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	—														BM	POR	
TYPE	R														R/W		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
	RAH	CFS	CFC1	CFC2	CAC	IFH	ITHE	ITH	DC1	DC0	HRA	HR4	HR3	HR2	HR1	HR0	
TYPE	R/W	R	R/W				R			R/W							
RESET	0	—	0	0	0	0	0	—	—	—	0	0	0	0	0	0	

**Table 19.** HCSR Bit Descriptions

Name	Reset	Description	Settings
— 31–18	0	Reserved. Write to zero for future compatibility.	
<b>BM</b> 17	0	<b>Boot Mode</b> Sets the boot mode of the aggregator DSP after the next PORESET. When power is first applied to the board, the FPGA asserts the PORESET signal for a few micro-seconds while it drives the BOOTMODE <sub>external</sub> signal low. The FPGA then release the PORESET signal, booting the DSP via Flash. To boot the aggregator via HDI16, set the BM bit and the POR bit of the HCSR for some time, then clear the POR bit but keep the BM bit set. Finally, send the Hardware Reset Configuration Word (HRCW) settings to the DSP via the HDI16 direct access space.	0 Boot via Flash device (default). 1 Boot via the HDI16.
<b>POR</b> 16	0	<b>Power-On Reset (PORESET)</b> Controls the state of the PORESET pin on the FPGA.	0 PORESET pin on the FPGA is set to High-Z. 1 PORESET pin on the FPGA is driven low.
<b>RAH</b> 15	0	<b>HDI16 Request Active High</b> Sets the polarity of the request lines from the aggregator to the FPGA for buffered accesses. These signals, HRRQ and HTRQ, are driven by the aggregator. The polarity programmed in the aggregator should match the polarity programmed with this bit if buffered (FIFO) accesses to the HDI16 data registers are performed.	0 Polarity is active low. 1 Polarity is active high.
<b>CFS</b> 14	—	<b>CT Bus Frame/C8 Failover Status</b> Indicates whether a failover on the CT bus frame and C8 signals was performed. This bit is cleared when a manual selection CT bus failover command (CFC) is issued. This bit is set when an active failover CT bus failover command (CFC) is issued and a failover is triggered.	

**Table 19. HCSR Bit Descriptions (Continued)**

Name	Reset	Description	Settings
<b>CFC[1–2]</b> 13–12	0	<b>CT Bus Frame/C8 Failover Command</b> Control the mode of the CT bus frame/C8 failover module. By default, the failover command is set to SELECT A. If an active mode type is selected, the first channel tested is the channel last selected by a manual mode. Failure of C8 is triggered if the selected C8 signal oscillates at a rate less than 7.14 MHz. Failure of frame is triggered if the selected frame signal oscillates at a rate less than 512 C8 clocks. If a failure is triggered, the CFS bit of the HCSR is set.	00 SELECT A. Manually select channel C8A and FA. 01 SELECT B. Manually select channel C8B and FB. 10 SENSE C8. Switch channels if a failure occurs on C8. 11 SENSE C8 and F. Switch channels if a failure occurs on C8 or frame.
<b>CAC</b> 11	0	<b>Clear (Disable) Aggregator Command</b> Resets and disables the serial command transferred from the aggregator to the FPGA.	0 Release the serial interface reset and allow normal operation to begin. 1 Clear the Aggregator Command Register (ACR) and place the aggregator serial interface into continuous reset.
<b>IFH</b> 10	0	<b>Interrupt Request from Host to Aggregator</b> Directly mapped to an output pin connected to an aggregator interrupt input.	0 Drive the interrupt pin low (default). 1 Drive the interrupt pin high.
<b>ITHE</b> 9	0	<b>Interrupt Request to Host from Aggregator Enable</b> Allows the aggregator to signal an interrupt to the host through a bus-specific interrupt signal.	0 No interrupt (default). 1 Interrupts as indicated by the ITH bit of the HCSR trigger an interrupt.
<b>ITH</b> 8	—	<b>Interrupt Request to Host from Aggregator Status</b> Indicates an interrupt service request to the host. This bit is set if the IRQ bit in the Aggregator Command Register (ACR) is set.	
<b>DC[1–0]</b> 7–6	—	<b>DSP Count</b> Indicates the intended size of the DSP farm. The number of DSPs in the farm is equal to the binary value of the DC field plus two. For example, if the DC value is 0b11, which equals three in binary, the number of DSPs installed is five. If the DC value is 0b00, which equals zero in binary, the number of DSPs installed is two. If two DSPs are indicated, the host should use the $\overline{\text{HRESET}}$ Status Register (HSR) and the HR field of the HCSR to confirm the number of DSPs.	
<b>HRA</b> 5	0	<b><math>\overline{\text{HRESET}}</math> Aggregator</b> Controls the aggregator $\overline{\text{HRESET}}$ pin.	0 The FPGA tri-states its output pin and does not drive a value onto the aggregator $\overline{\text{HRESET}}$ pin. 1 Aggregator $\overline{\text{HRESET}}$ pin is driven low.
<b>HR[4–0]</b> 4–0	0	<b><math>\overline{\text{HRESET}}</math> DSP Farm</b> Control the $\overline{\text{HRESET}}$ pins of the DSP farm. These bits are ORed with the HR field of the Aggregator Command Register (ACR) to determine the actual $\overline{\text{HRESET}}$ commands. If one of these bits is set, the corresponding farm DSP $\overline{\text{HRESET}}$ pin is driven low. The FPGA tri-states the corresponding $\overline{\text{HRESET}}$ pin if both bits of the same index are cleared.	

TSOR		Temperature Sensor Output Register														Byte Offset 0x4	
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		—															
TYPE		R															
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		V11	V10	V9	V8	V7	V6	V5	V4	V3	V2	V1	V0	—		TSD	
TYPE		R															
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Table 20.** TSOR Bit Descriptions

Name	Reset	Description	Settings
— 31–16	0	Reserved. Write to zero for future compatibility.	
<b>V[11–0]</b> 15–4	0	<b>Temperature Sensor Output Value</b> Indicates the last sampled temperature taken on the board. The temperature is updated at approximately 22 kHz. To get the temperature in Celsius, multiply the temperature value by 0.0625 or divide by 16. Dividing the entire TSOR register value by 256 provides the same result.	
— 3–1	0	Reserved. Write to zero for future compatibility.	
<b>TSD</b> 0	0	<b>Temperature Sensor Disable</b> Disables the temperature sensor and puts the device into low-power shutdown mode, which reduces the typical operating current from 265 micro-amps to 3 micro-amps.	0 Place the device into operational mode with the temperature sensor in operational mode (default). 1 Place the device into low-power shutdown mode.

RCR		RMII Control Register														Byte Offset 0xC	
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE		—															
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE		—										MLE2	MLE1	RCE2	RE2	RCE1	RE1
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Table 21. RCR Bit Descriptions**

Name	Reset	Description
— 31–6	0	Reserved. Write to zero for future compatibility.
<b>MLE[2–1]</b> 5–4	0	<b>MII Loop-back Enable</b> Control the internal loop-back of MII signals from the aggregator. The selected MII channel from the aggregator is looped back internal to the FPGA if the bit is set. The RMII to MII converter module must also be enabled. Set the proper RMII Clock Enable bit to enable the FPGA to source the MII Tx and Rx clocks.
<b>RCE2</b> 3	0	<b>RMII Clock Enable 2</b> Enable the FPGA to source the 50 MHz RMII reference clocks. The RCE[2–1] bits are ORed with the RCE bits of the Aggregator Command Register (ACR) to determine the actual RCE commands. If one of these bits is set, the FPGA drives the 50 MHz reference clock for the corresponding RMII interface. The FPGA tri-states the corresponding RMII clock out pin if both bits of the same index are cleared. The corresponding RMII interface must also be enabled for the FPGA to begin driving the RMII clock out pin.
<b>RE2</b> 2	0	<b>RMII Enable 2</b> Enable the FPGA RMII-to-MII converter modules. The RE[2–1] bits are ORed with the RE bits of the Aggregator Command Register (ACR) to establish the actual RE commands. If one of these bits is set, the FPGA enables the corresponding RMII-to-MII converter module. The FPGA disables the corresponding converter module if both bits of the same index are cleared. Disabling a module causes the FPGA to tri-state all output pins connected to the module, allowing normal R/MII traffic to proceed.
<b>RCE1</b> 1	0	<b>RMII Clock Enable 1</b> Enable the FPGA to source the 50 MHz RMII reference clocks. The RCE[2–1] bits are ORed with the RCE bits of the Aggregator Command Register (ACR) to determine the actual RCE commands. If one of these bits is set, the FPGA drives the 50 MHz reference clock for the corresponding RMII interface. The FPGA tri-states the corresponding RMII clock out pin if both bits of the same index are cleared. The corresponding RMII interface must also be enabled for the FPGA to begin driving the RMII clock out pin.
<b>RE1</b> 0		<b>RMII Enable 1</b> Enable the FPGA RMII-to-MII converter modules. The RE[2–1] bits are ORed with the RE bits of the Aggregator Command Register (ACR) to establish the actual RE commands. If one of these bits is set, the FPGA enables the corresponding RMII-to-MII converter module. The FPGA disables the corresponding converter module if both bits of the same index are cleared. Disabling a module causes the FPGA to tri-state all output pins connected to the module, allowing normal R/MII traffic to proceed.

ACR		Aggregator Command Register														Byte Offset 0x10	
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		—															AFE
TYPE		R															
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		AF5	AF4	AF3	AF2	AF1	AF0	RCE2	RE2	RCE1	RE1	IRQ	HR4	HR3	HR2	HR1	HR0
TYPE		R															
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

**Table 22.** ACR Bit Descriptions

Name	Reset	Description
— 31–17	0	Reserved. Write to zero for future compatibility.
<b>AFE</b> 16	0	<b>Aggregator Command Frame Error</b> Indicates an error condition generated on the transfer of command data from the aggregator to the FPGA. The transfer must follow a specific format; otherwise, a frame error is generated. The frame error is cleared if the CAC bit in the Host Control/Status Register (HCRS) is set or when the next successful transfer completes.
<b>AF[5–0]</b> 15–10	0	<b>Aggregator Flags</b> General-purpose flags by which the aggregator communicates information to the host, such as system states or service requests. This field can be used with the IRQ field of this register to indicate the types of service requested by the aggregator.
<b>RCE2</b> 9	0	<b>RMII Clock Enable 2</b> Enables the FPGA to source the 50 MHz RMII reference clocks. The RCE bits are ORed with the RCE bits of the RMII Control Register (RCR) to determine the actual RCE commands. If one of these bits is set, the FPGA drives the 50 MHz reference clock for the corresponding RMII interface. The FPGA tri-states the corresponding RMII clock out pin if both bits of the same index are cleared. The corresponding RMII interface must also be enabled for the FPGA to begin driving the RMII clock out pin.
<b>RE2</b> 8	0	<b>RMII Enable 2</b> Enables the FPGA RMII-to-MII converter modules. These bits are ORed with the RE bits of the RMII Control Register (RCR) to determine the actual RE commands. If one of these bits is set, the FPGA enables the corresponding RMII-to-MII converter module. The FPGA disables the corresponding RMII-to-MII converter module if both bits of the same index are cleared. Disabling a module causes the FPGA to tri-state all output pins connected to the module, allowing normal R/MII traffic to proceed.
<b>RCE1</b> 7	0	<b>RMII Clock Enable 1</b> Enables the FPGA to source the 50 MHz RMII reference clocks. The RCE bits are ORed with the RCE bits of the RMII Control Register (RCR) to determine the actual RCE commands. If one of these bits is set, the FPGA drives the 50 MHz reference clock for the corresponding RMII interface. The FPGA tri-states the corresponding RMII clock out pin if both bits of the same index are cleared. The corresponding RMII interface must also be enabled for the FPGA to begin driving the RMII clock out pin.

**Table 22.** ACR Bit Descriptions (Continued)

Name	Reset	Description
<b>RE1</b> 6	0	<b>RMII Enable 1</b> Enables the FPGA RMII-to-MII converter modules. These bits are ORed with the RE bits of the RMII Control Register (RCR) to determine the actual RE commands. If one of these bits is set, the FPGA enables the corresponding RMII-to-MII converter module. The FPGA disables the corresponding RMII-to-MII converter module if both bits of the same index are cleared. Disabling a module causes the FPGA to tri-state all output pins connected to the module, allowing normal R/MII traffic to proceed.
<b>IRQ</b> 5	0	<b>Interrupt Request to Host</b> Indicates that the aggregator is requesting service from the host. When this bit is set, an interrupt request is generated.
<b>HR[4-0]</b> 4-0	0	<b>HRESET DSP Farm</b> Controls the $\overline{\text{HRESET}}$ pins of the DSP farm. These bits are ORed with the HR field of the Host Control/Status Register (HCSR) to determine the actual $\overline{\text{HRESET}}$ commands. If one of these bits is set, the corresponding farm DSP $\overline{\text{HRESET}}$ pin is driven low. The FPGA tri-states the corresponding $\overline{\text{HRESET}}$ pin if both bits of the same index are cleared.

**HSR**  $\overline{\text{HRESET}}$  Status Register Byte Offset 0x14

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TYPE	—															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE	—										HRA	HR4	HR3	HR2	HR1	HR0
RESET	0	0	0	0	0	0	0	0	0	0	—	—	—	—	—	—

**Table 23.** HSR Bit Descriptions

Name	Reset	Description	Settings
— 31-6	0	Reserved. Write to zero for future compatibility.	
<b>HRA</b> 5	—	<b>HRESET Aggregator Status</b> Indicates the current state of the aggregator $\overline{\text{HRESET}}$ pin. Since multiple sources can drive the $\overline{\text{HRESET}}$ pin of the aggregator, this bit allows a true indication of the state of the pin.	0 The aggregator $\overline{\text{HRESET}}$ pin is low and is in reset mode. 1 The aggregator $\overline{\text{HRESET}}$ pin is high and is no longer in reset.
<b>HR[4-0]</b> 4-0	—	<b>HRESET DSP Farm Status Bits</b> Indicates the current state of the farm $\overline{\text{HRESET}}$ pins. Since multiple sources can drive the $\overline{\text{HRESET}}$ pins of the farm, these bits allow to a true indication of the state of these pins.	0 The farm $\overline{\text{HRESET}}$ pin is low and is in reset mode. 1 The farm $\overline{\text{HRESET}}$ pin is high and is no longer in reset.

FVR		FPGA Version Register														Byte Offset 0x1C	
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		—															
TYPE		R															
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		MV7	MV6	MV5	MV4	MV3	MV2	MV1	MV0	mV7	mV6	mV5	mV4	mV3	mV2	mV1	mV0
TYPE		R															
RESET		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

**Table 24.** FVR Bit Descriptions

Name	Reset	Description
— 31–16	0	Reserved. Write to zero for future compatibility.
<b>MV[7–0]</b> 15–8	—	<b>Major Version</b> Indicates the 2-digit major version number of the FPGA firmware. The version number is encoded in binary coded decimal (BCD) format, so each 4-bit group specifies a number from 0–9, using binary numbers 0b0000–0b1001.
<b>mV[7–0]</b> 7–0	—	<b>Minor Version</b> Indicates the 2-digit minor version number for the FPGA firmware. The version number is encoded in binary coded decimal (BCD) format, so each 4-bit group specifies a number from 0–9, using binary numbers 0b0000 to 0b1001.

## SPR Scratch Pad RAM

The subsystem module provides approximately 1 kB of general-purpose RAM that must be accessed in 32-bit widths and alignments. The address range assigned to the RAM is 0x0020–0x03FF, as shown in **Table 15**.

## 4.4 PCI Bus Interface

The PT3MC FPGA version interfaces with the host through a 32-bit, 33 MHz, 3.3 V PCI bus. The interface acts only as a PCI target. The registers to configure the PCI interface are accessible through PCI configuration read/write cycles. See the PCI specification for details. Since the device acts only as a PCI target, these registers are a subset of the PCI configuration space registers. These registers are as follows:

- Vendor/Device ID Register (VDID), **page 33**.
- Command Status Register (CSR), **page 33**.
- Class Code/Rev ID Register (CCRIDR), **page 34**.
- Header Type Register (HTR), **page 34**.
- Base Address 0 Register (BADDR0), **page 35**.
- Subsystem ID Register (SIDR), **page 35**.
- Configuration Space Scratch Pad Register (), **page 35**.

VDID		Vendor/Device ID Register														Byte Offset 0x00	
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DEVICE ID															
TYPE		R															
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		VENDOR ID															
TYPE		R															
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Vendor ID and Device ID Register is scanned by the PCI configuration host to determine the card in each slot. The IDs uniquely identify the cards. The vendor ID should be set in the source code to a number assigned to the board designer by the PCI Special Interest Group (PSIG). The vendor assigns the device ID to identify the card. New for PCI 2.2 is the Subsystem ID Register, by which the vendor can further differentiate the cards.

CSR		Command/Status Register														Byte Offset 0x04	
Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		STATUS															
		DPE	SSE	—	STA	—											
TYPE		R/C	R/C	R	R/C	R											
RESET		—	—	0	0	—	0	0	0	0	0	0	0	0	0	0	0
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		COMMAND															
		—						SE	—	PER	—				MS	—	
TYPE		R						R/W	R	R/W							
RESET		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

The Command/Status Registers are required by the PCI specification to allow the host to probe interface status and to control interface capability options for each card on the bus. The Status Register in the PT3MC FPGA version contains only a subset of the many indicator flag bits in the PCI Specification. Only the Signaled Target Abort (bit 27), Signaled System Error (bit 30), and Detected Parity Error (bit 31) indicator flag bits are implemented in the Status Register. These bits are set when the underlying condition occurs and cleared only when the host writes a one into the bit location. Writing a zero into the bit location has no effect.

The Command Register in the PT3MC FPGA version contains only a subset of the many option enable bits in the PCI specification. Only the Memory Space (bit 1), Parity Error Response (bit 6), and SERR# Enable (bit 8) are implemented. The host sets these bits to enable optional functionality and clears them to disable the functionality. By default, all optional functionality is disabled.

**CCRDR** Class Code/Revision ID Register Byte Offset 0x08

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLASS/SUBCLASS CODE															
TYPE	R															
RESET	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROG INTERFACE								REV ID							
TYPE	R															
RESET	0	0	0	0	0	0	0	0	—	—	—	—	—	—	—	—

The Class Code and Rev ID Register identifies the device class and revision number of the card. The FPGA has a class code set to 0x078000, which indicates a simple communications controller, with an “others” subclass. See the PCI specification for details. The revision number should be incremented for every release of firmware that requires a PCI bus driver change or affects performance.

**HTR** Header/Type Register Byte Offset 0x0C

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—								HEADER TYPE							
TYPE	R															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—															
TYPE	R															
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Header Type Register identifies the format of the remaining PCI Configuration space. The FPGA has a header type set to 0x00, indicating a single-function standard device header.

<b>BADDR0</b>		<b>Base Address 0 Register</b>														<b>Byte Offset 0x10</b>			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
	BASE ADDRESS																		
TYPE	R/W																		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
																—	IOS		
TYPE	R																		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

The Base Address 0 Register is a means for the host to configure the PCI base address for the FPGA. The FPGA responds to any transfer on the PCI bus within the address range specified by the base address and space size. The requested space size is set to 16 kB. The I/O Space bit (bit 0) of the register indicates whether the target is requesting a region of memory or I/O space. By default, this bit cleared, indicating a request for a region of memory space. The FPGA responds only to transfers within the assigned region if the Memory Space bit is set in the Command Register. Until it is set, the host is free to manipulate the base address register in any way. To determine the space size requested by the card, the host attempts to set all bits in the register and then read the result. The width of the bit field set during this operation determines the requested space size. The host reserves a section of memory in which the bits in the same field remain constant. The host sees bits 31—16 as set and bits 15—0 cleared after it performs this operation on the FPGA. Therefore, the host reserves a 16 kB region of memory space and writes the 16 most significant bits of the address space into bits 31—16 of the base address register.

<b>SIDR</b>		<b>Subsystem ID Register</b>														<b>Byte Offset 0x2C</b>			
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16			
	SUBSYSTEM ID																		
TYPE	R																		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
	SUBSYSTEM VENDOR ID																		
TYPE	R																		
RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

The Subsystem ID Register contains two values, both set by the vendor. These fields can be used to differentiate further between devices. By default, these fields are cleared to zero.

### Configuration Space Scratch Pad Register

A 32-bit Scratch Pad Register resides in configuration register space to help with PCI bus testing and debugging. The register is located at an address that is reserved and unused in the PCI 2.2 specification. The register is located at byte offset 0x38 of the configuration register space.



# Appendix A      Schematics



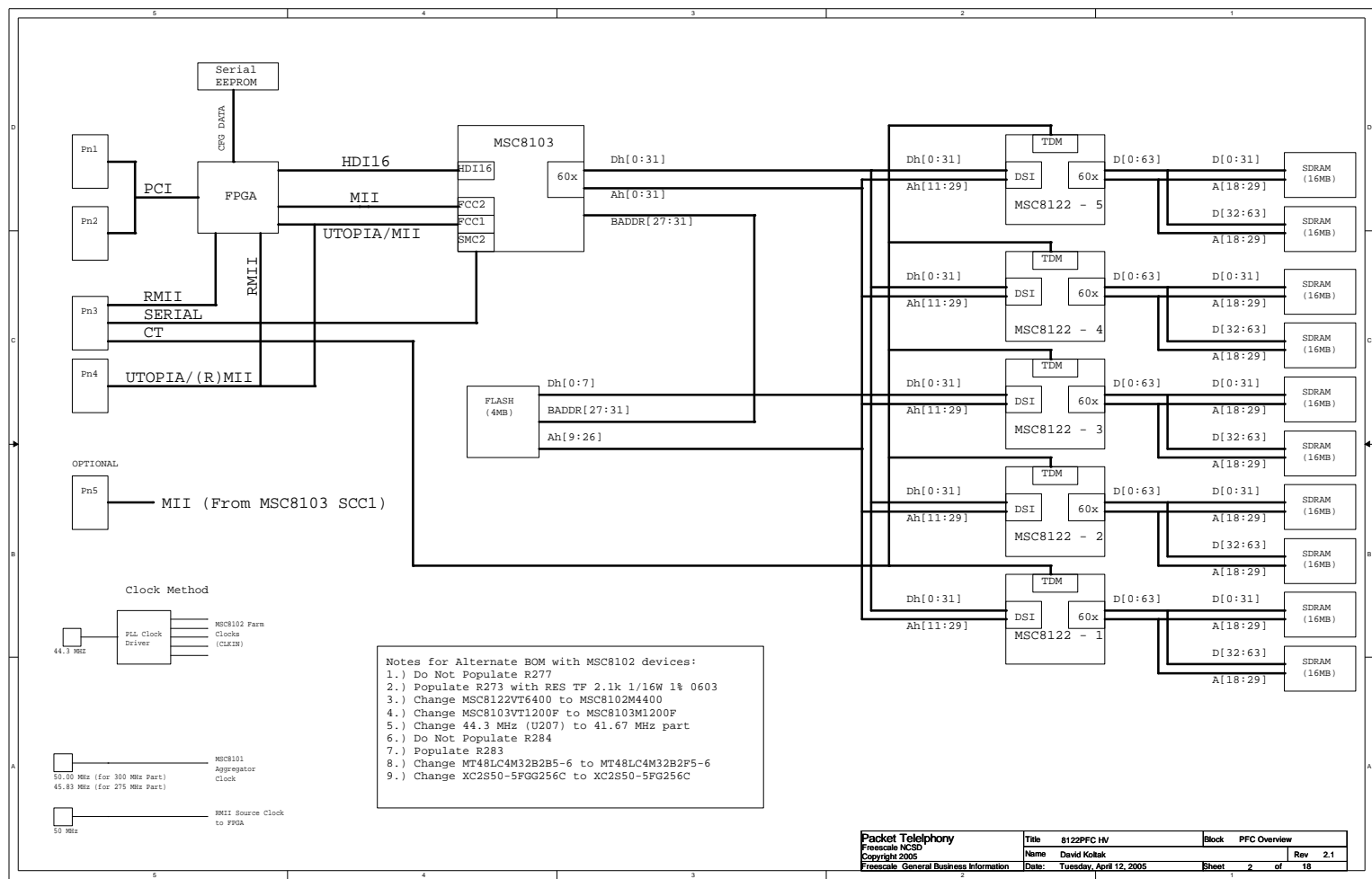
MSC8122 - PACKET TELEPHONY FARMCARD (8122PFC)  
HIGH VOLUME

**Table of Contents**

Page	Description
1	This Page
2	PFC Overview
3	PTMC Connectors
4	MSC8103 Core
5	MSC8103 Memory
6	FPGA
7	POWER / RESET
8	DSP FARM
9	DSP1 FARM CORE
10	DSP1 FARM MEMORY
11	DSP2 FARM CORE
12	DSP2 FARM MEMORY
13	DSP3 FARM CORE
14	DSP3 FARM MEMORY
15	DSP4 FARM CORE
16	DSP4 FARM MEMORY
17	DSP5 FARM CORE
18	DSP5 FARM MEMORY

**Revision History**

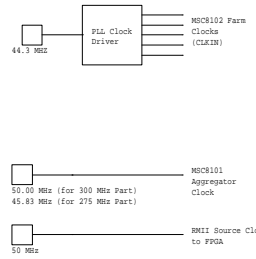
1.0	First Release
1.1	DRC clean-up and incorporated comments
1.2	STX/SRX, Temp IC, PORESET-J1, P5
1.3	64-bit SRAM and clean up to get ready for layout
1.4	Post Design Review
1.5	Modified FPGA Pinout to Optimize Layout
1.6	Added Third Oscillator to FPGA for optional RMII clock source
1.7	Back Annotated from Layout Design
1.8	Added Serial EEPROM for FPGA Boot
2.0	TDM to 8101, J1, TDM Mismatch, Boot Mode, Farm Caps
2.1	Heat Sink on Linear Regulators, Caps, Inductor



OPTIONAL

Pn5 → MII (From MSC8103 SCC1)

**Clock Method**



Notes for Alternate BOM with MSC8102 devices:

- 1.) Do Not Populate R277
- 2.) Populate R273 with RES TF 2.1k 1/16W 1% 0603
- 3.) Change MSC8122VT6400 to MSC8102M4400
- 4.) Change MSC8103VT1200F to MSC8103M1200F
- 5.) Change 44.3 MHz (U207) to 41.67 MHz part
- 6.) Do Not Populate R284
- 7.) Populate R283
- 8.) Change MT48LC4M32B2B5-6 to MT48LC4M32B2P5-6
- 9.) Change XC2S50-5FGG256C to XC2S50-5PG256C

<b>Packet Telephony</b>	<b>Title</b> 8122PFC HW	<b>Block</b> PFC Overview
Freescale WCD	<b>Name</b> David Koltak	<b>Rev</b> 2.1
Copyright 2005	<b>Date</b> Tuesday, April 12, 2005	<b>Sheet</b> 2 of 18
Freescale. General Business Information		



- Notes :
- 1.) Populate resistors dependent on intended PTMC board type. By default, the board should be type 3. Therefore, populate resistors R201 and R226 only. The carrier should have pull-down resistors. See Table 1 for a definition of PTMC board types.
  - 2.) Signal to PCI that card is 66MHz capable. By default, populate the resistor to disable 66MHz Bus Mode.
  - 3.) By default, do not populate. If FMC JTAG is not connected, populate to not brake chain.

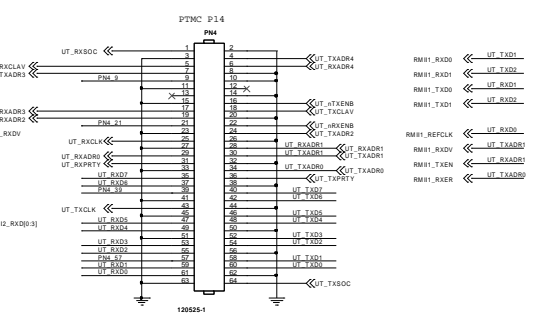
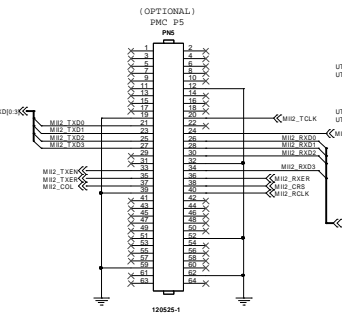
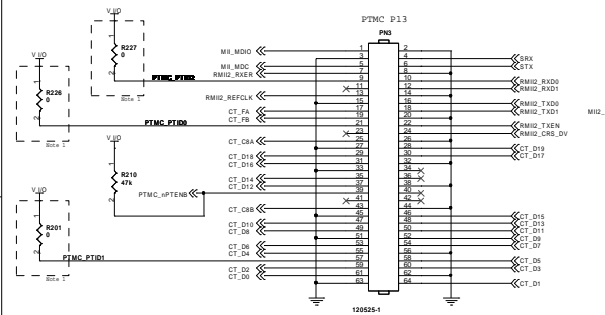
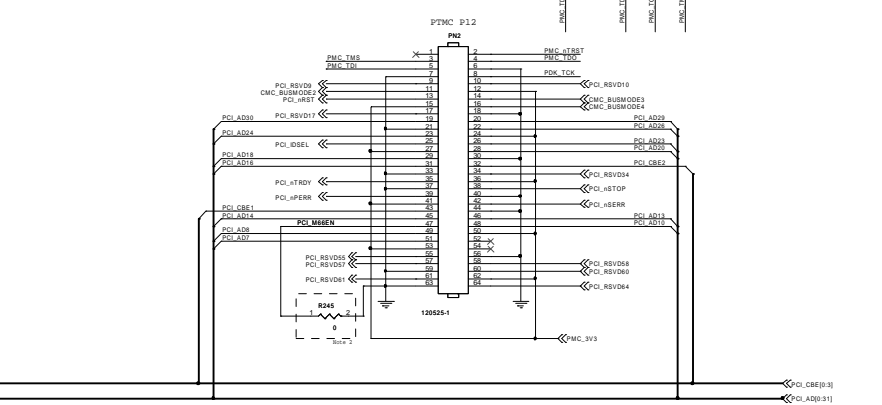
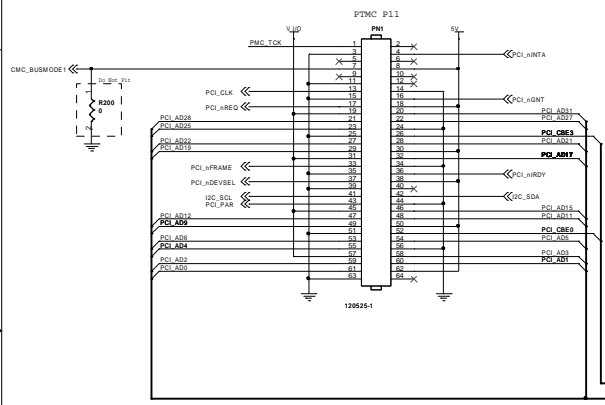
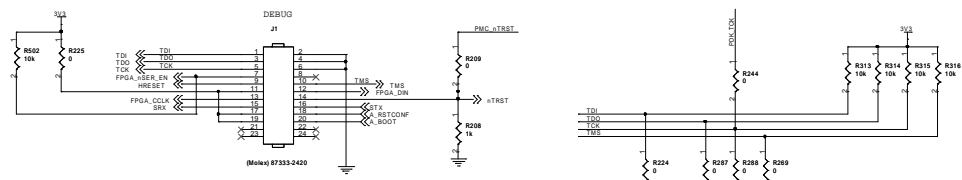
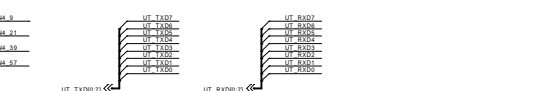
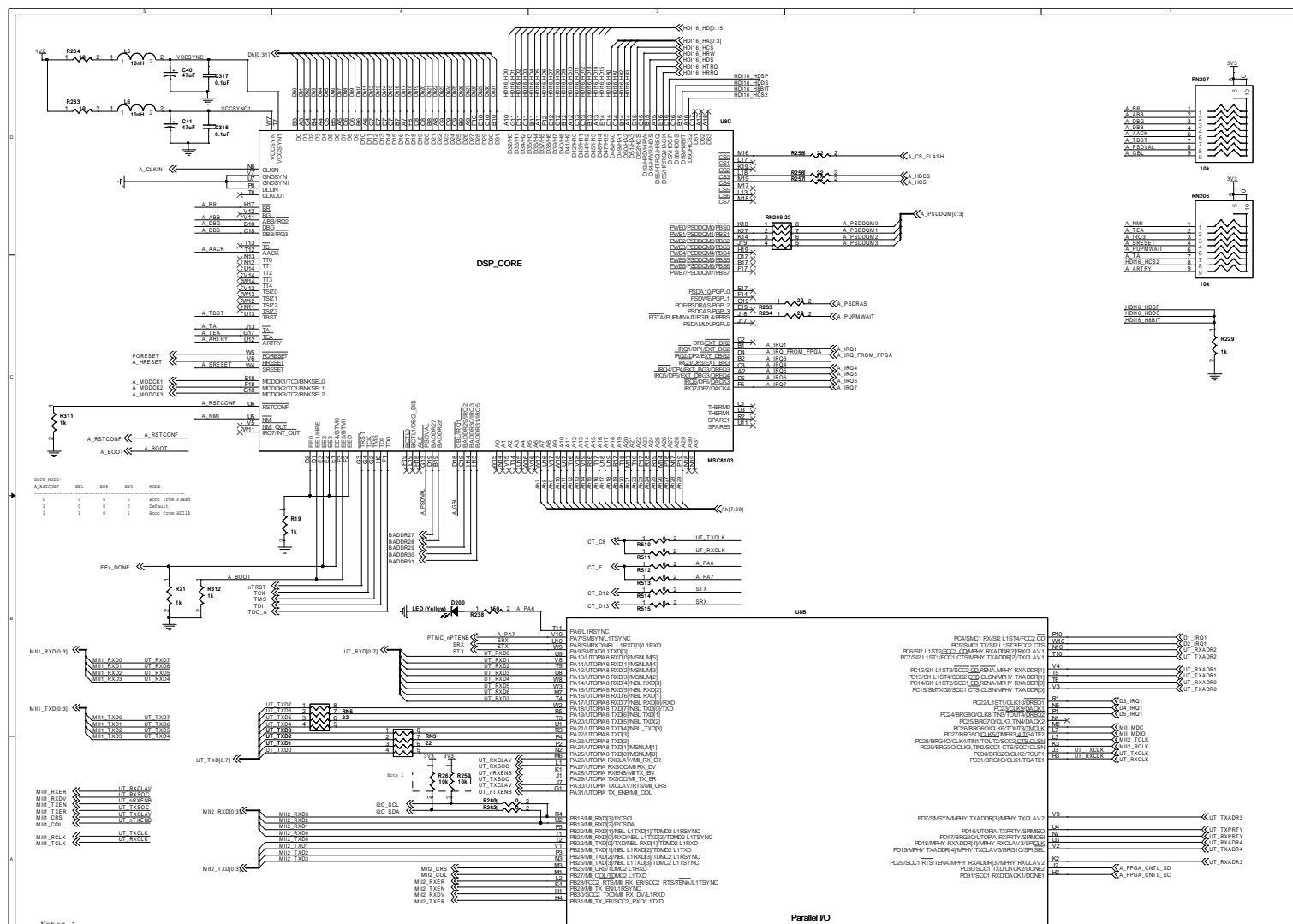
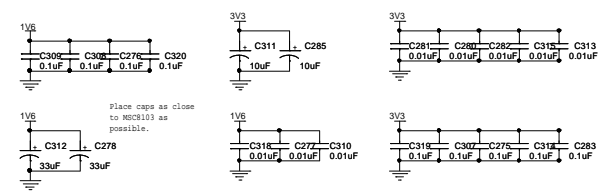
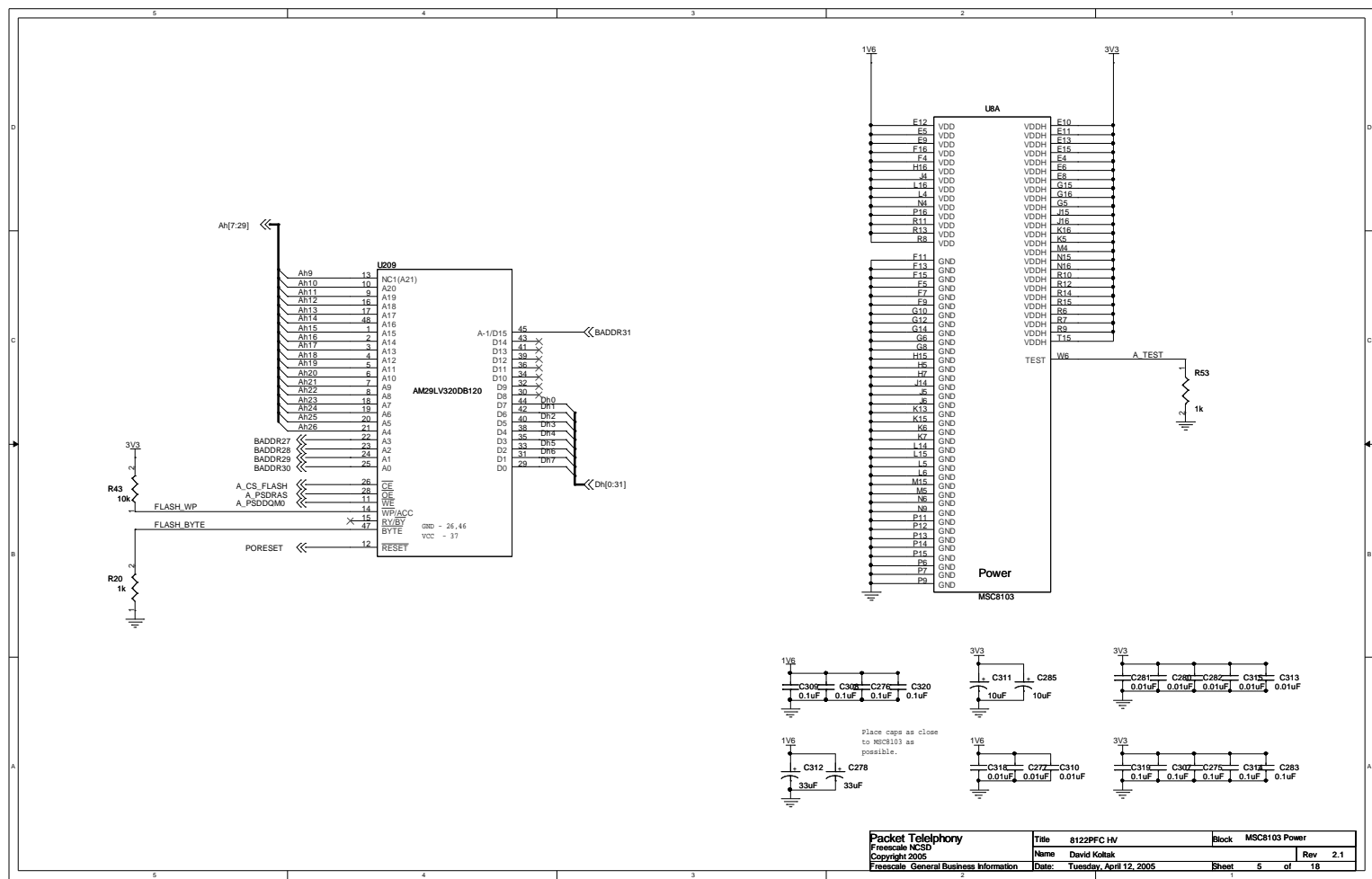


TABLE 1 - PTMC Board Types

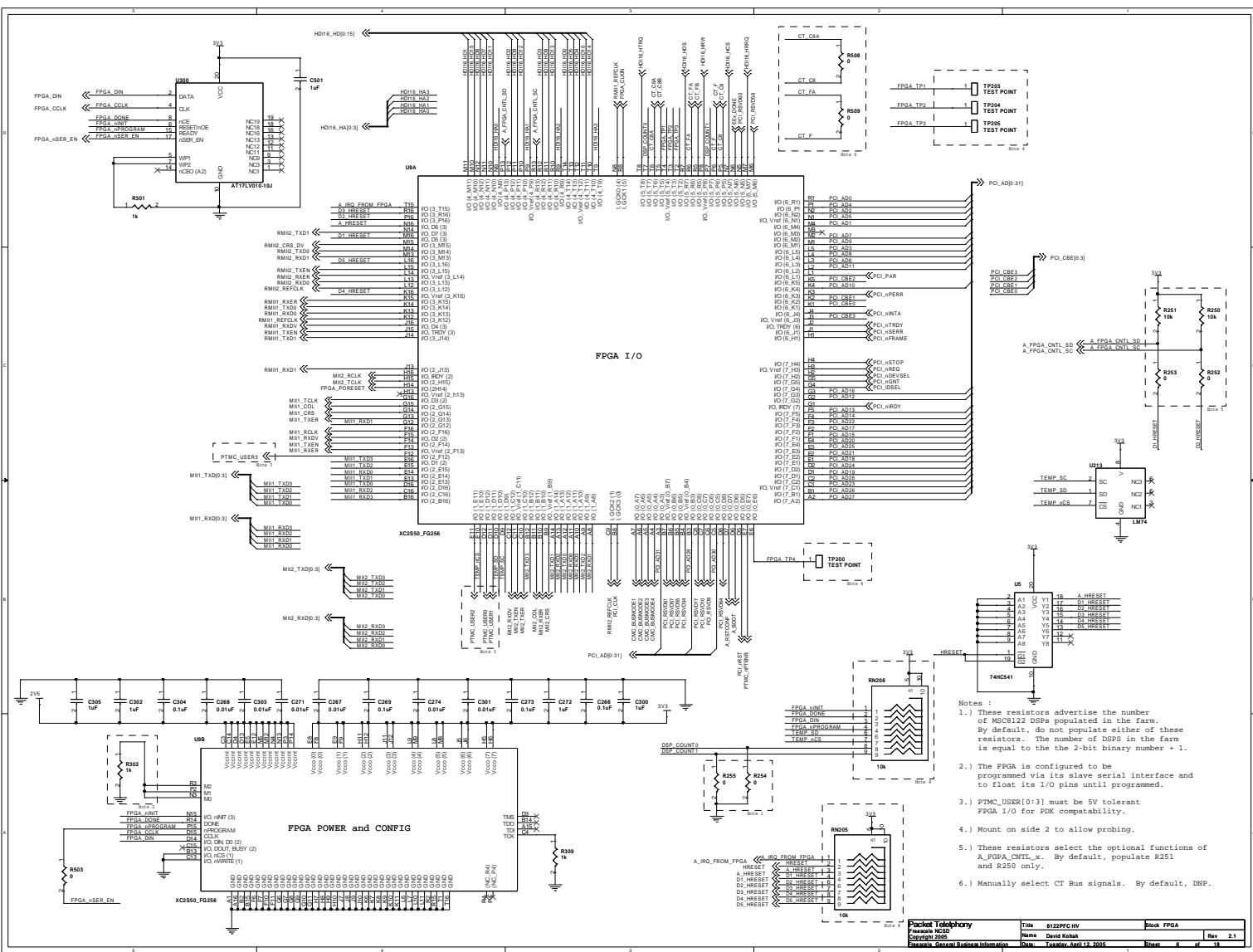
	0	1	2	3	4	5	6	7
R200	X	X	X	X	X	X	X	X
R208	X	X	X	X	X	X	X	X
R226	X	X	X	X	X	X	X	X
R244	X	X	X	X	X	X	X	X
R245	X	X	X	X	X	X	X	X
R246	X	X	X	X	X	X	X	X
R247	X	X	X	X	X	X	X	X
R248	X	X	X	X	X	X	X	X
R249	X	X	X	X	X	X	X	X
R250	X	X	X	X	X	X	X	X
R251	X	X	X	X	X	X	X	X
R252	X	X	X	X	X	X	X	X
R253	X	X	X	X	X	X	X	X
R254	X	X	X	X	X	X	X	X
R255	X	X	X	X	X	X	X	X
R256	X	X	X	X	X	X	X	X
R257	X	X	X	X	X	X	X	X
R258	X	X	X	X	X	X	X	X
R259	X	X	X	X	X	X	X	X
R260	X	X	X	X	X	X	X	X
R261	X	X	X	X	X	X	X	X
R262	X	X	X	X	X	X	X	X
R263	X	X	X	X	X	X	X	X
R264	X	X	X	X	X	X	X	X
R265	X	X	X	X	X	X	X	X
R266	X	X	X	X	X	X	X	X
R267	X	X	X	X	X	X	X	X
R268	X	X	X	X	X	X	X	X
R269	X	X	X	X	X	X	X	X
R270	X	X	X	X	X	X	X	X
R271	X	X	X	X	X	X	X	X
R272	X	X	X	X	X	X	X	X
R273	X	X	X	X	X	X	X	X
R274	X	X	X	X	X	X	X	X
R275	X	X	X	X	X	X	X	X
R276	X	X	X	X	X	X	X	X
R277	X	X	X	X	X	X	X	X
R278	X	X	X	X	X	X	X	X
R279	X	X	X	X	X	X	X	X
R280	X	X	X	X	X	X	X	X
R281	X	X	X	X	X	X	X	X
R282	X	X	X	X	X	X	X	X
R283	X	X	X	X	X	X	X	X
R284	X	X	X	X	X	X	X	X
R285	X	X	X	X	X	X	X	X
R286	X	X	X	X	X	X	X	X
R287	X	X	X	X	X	X	X	X
R288	X	X	X	X	X	X	X	X
R289	X	X	X	X	X	X	X	X
R290	X	X	X	X	X	X	X	X
R291	X	X	X	X	X	X	X	X
R292	X	X	X	X	X	X	X	X
R293	X	X	X	X	X	X	X	X
R294	X	X	X	X	X	X	X	X
R295	X	X	X	X	X	X	X	X
R296	X	X	X	X	X	X	X	X
R297	X	X	X	X	X	X	X	X
R298	X	X	X	X	X	X	X	X
R299	X	X	X	X	X	X	X	X
R300	X	X	X	X	X	X	X	X
R301	X	X	X	X	X	X	X	X
R302	X	X	X	X	X	X	X	X
R303	X	X	X	X	X	X	X	X
R304	X	X	X	X	X	X	X	X
R305	X	X	X	X	X	X	X	X
R306	X	X	X	X	X	X	X	X
R307	X	X	X	X	X	X	X	X
R308	X	X	X	X	X	X	X	X
R309	X	X	X	X	X	X	X	X
R310	X	X	X	X	X	X	X	X
R311	X	X	X	X	X	X	X	X
R312	X	X	X	X	X	X	X	X
R313	X	X	X	X	X	X	X	X
R314	X	X	X	X	X	X	X	X
R315	X	X	X	X	X	X	X	X
R316	X	X	X	X	X	X	X	X





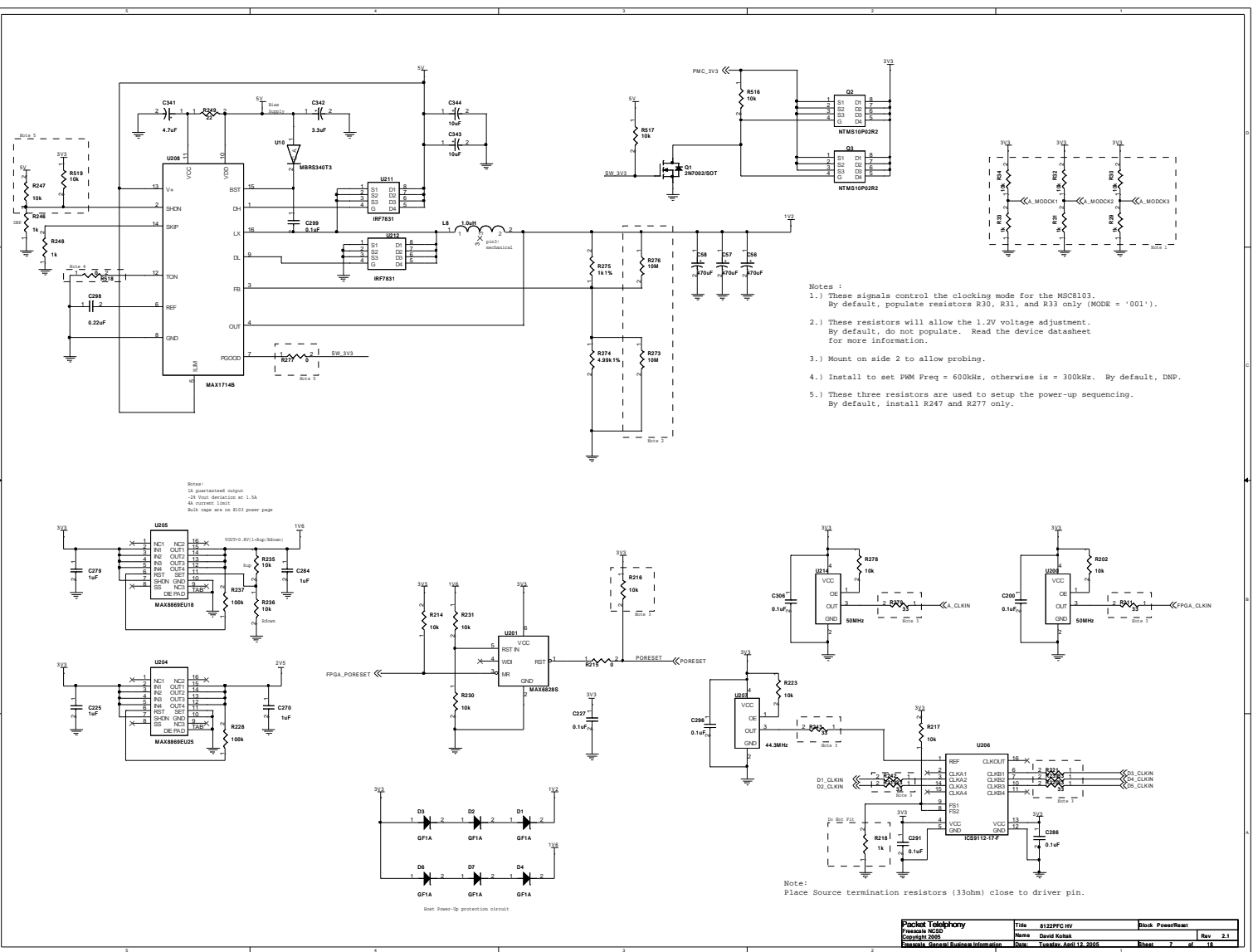


Packet Telephone	Title	8122PFC HV	Block	MSC8103 Power
Freescale MC3D	Name	David Koltak	Date	Tuesday, April 12, 2005
Copyright 2005	Rev	2.1	Sheet	5 of 18
Freescale General Business Information				



- Notes:
- 1.) These resistors advertise the number of MS08122 DSPs populated in the farm. By default, do not populate either of these resistors. The number of DSPs in the farm is equal to the the 2-bit binary number + 1.
  - 2.) The FPGA is configured to be programmed via its slave serial interface and to float its I/O pins until programmed.
  - 3.) PTMC\_USER[0:3] must be 5V tolerant FPGA I/O for PDK compatibility.
  - 4.) Mount on side 2 to allow probing.
  - 5.) These resistors select the optional functions of A\_FPGA\_CNTL\_x. By default, populate R251 and R250 only.
  - 6.) Manually select CT Bus signals. By default, DNP.

<b>Packet Telephony</b>	Title	8122PFC HW	Block	FPGA
Tweezer PCB	Name	David Kohan		
Copyright 2005	Date	Tuesday, April 12, 2005	Sheet	6 of 18
David Kohan	Author			



Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

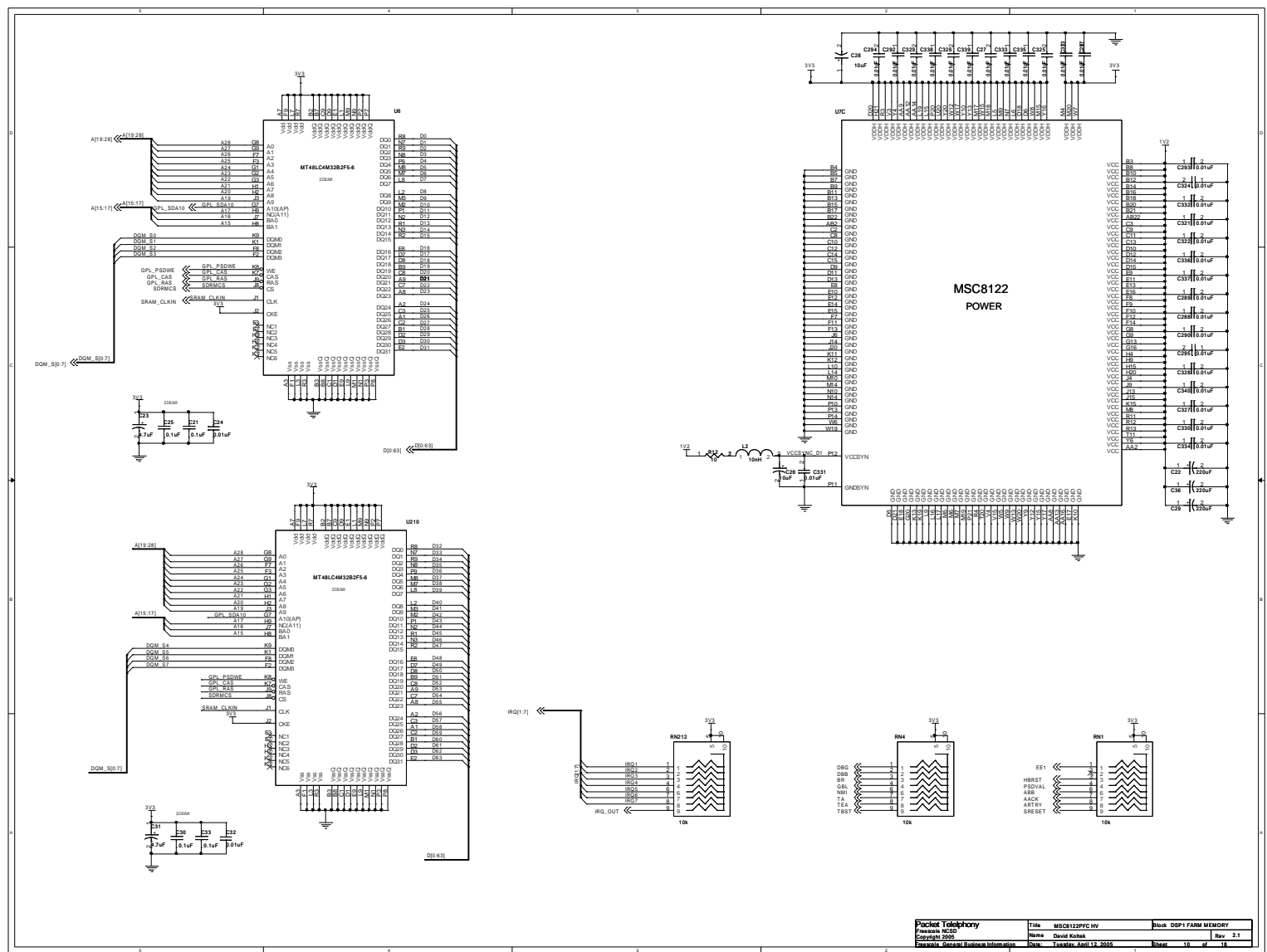
Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

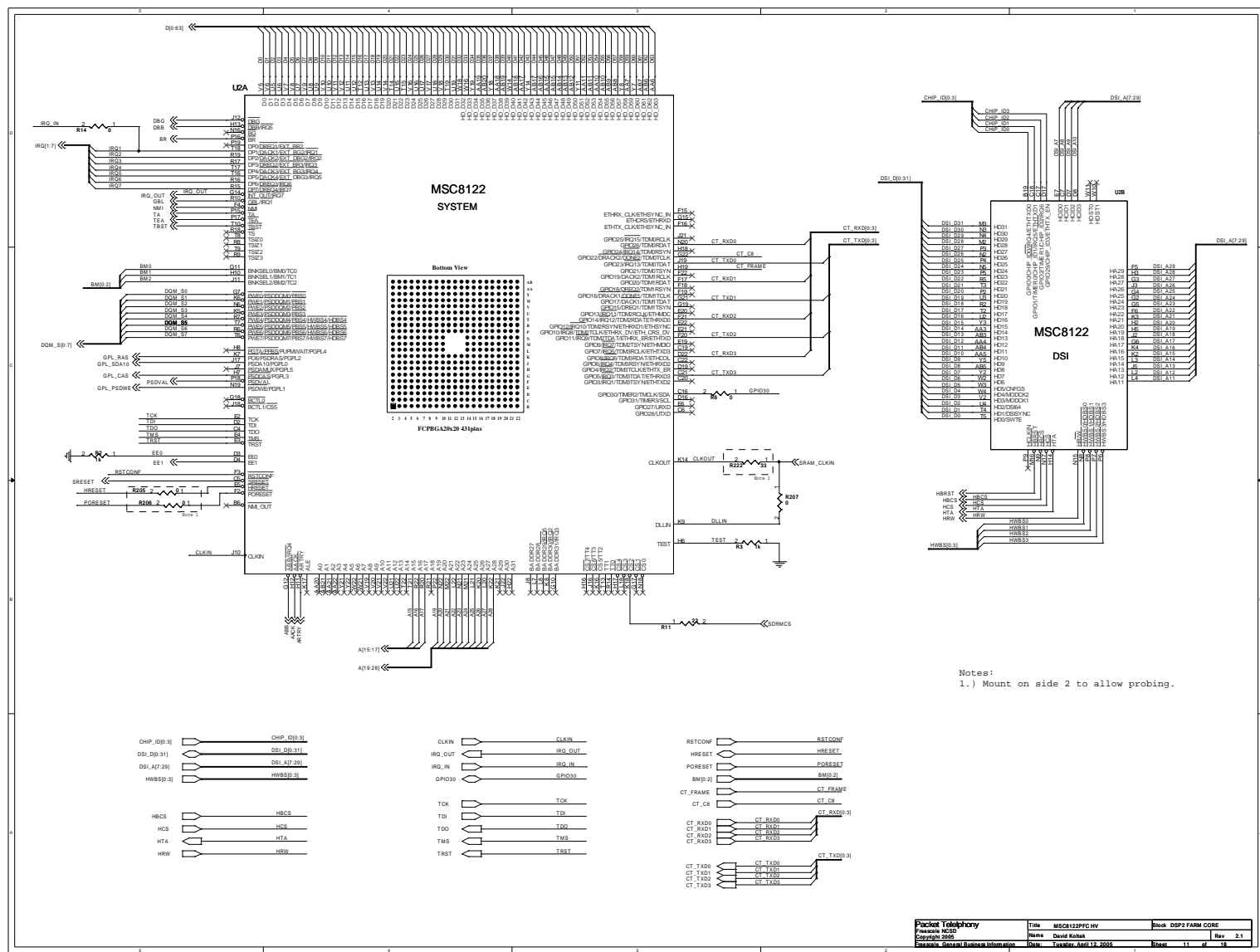
Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page

Notes:  
 18. Overcurrent output  
 -28 Volt deviation at 1.5A  
 40. Current Limit  
 Bulk caps are on B03 power page



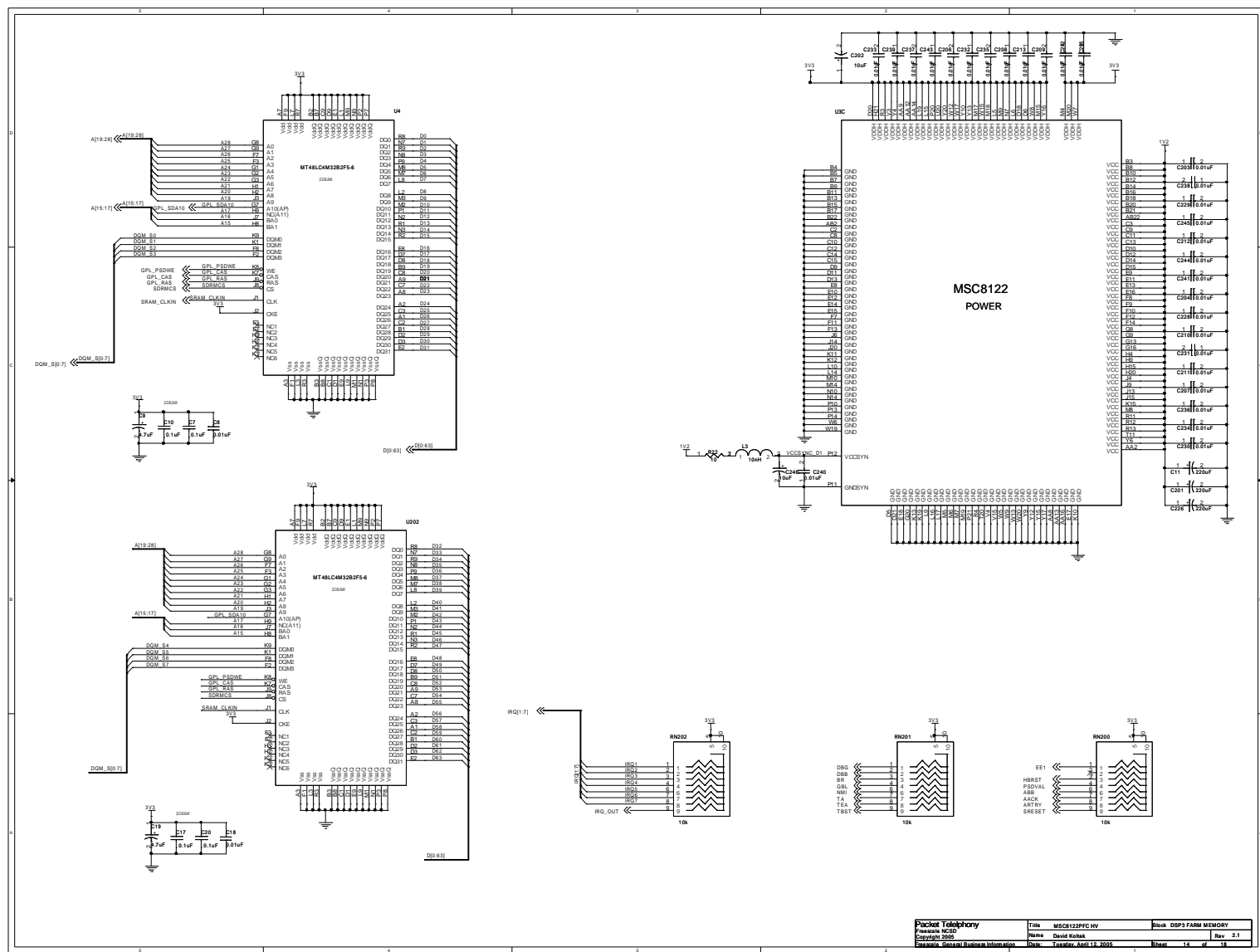




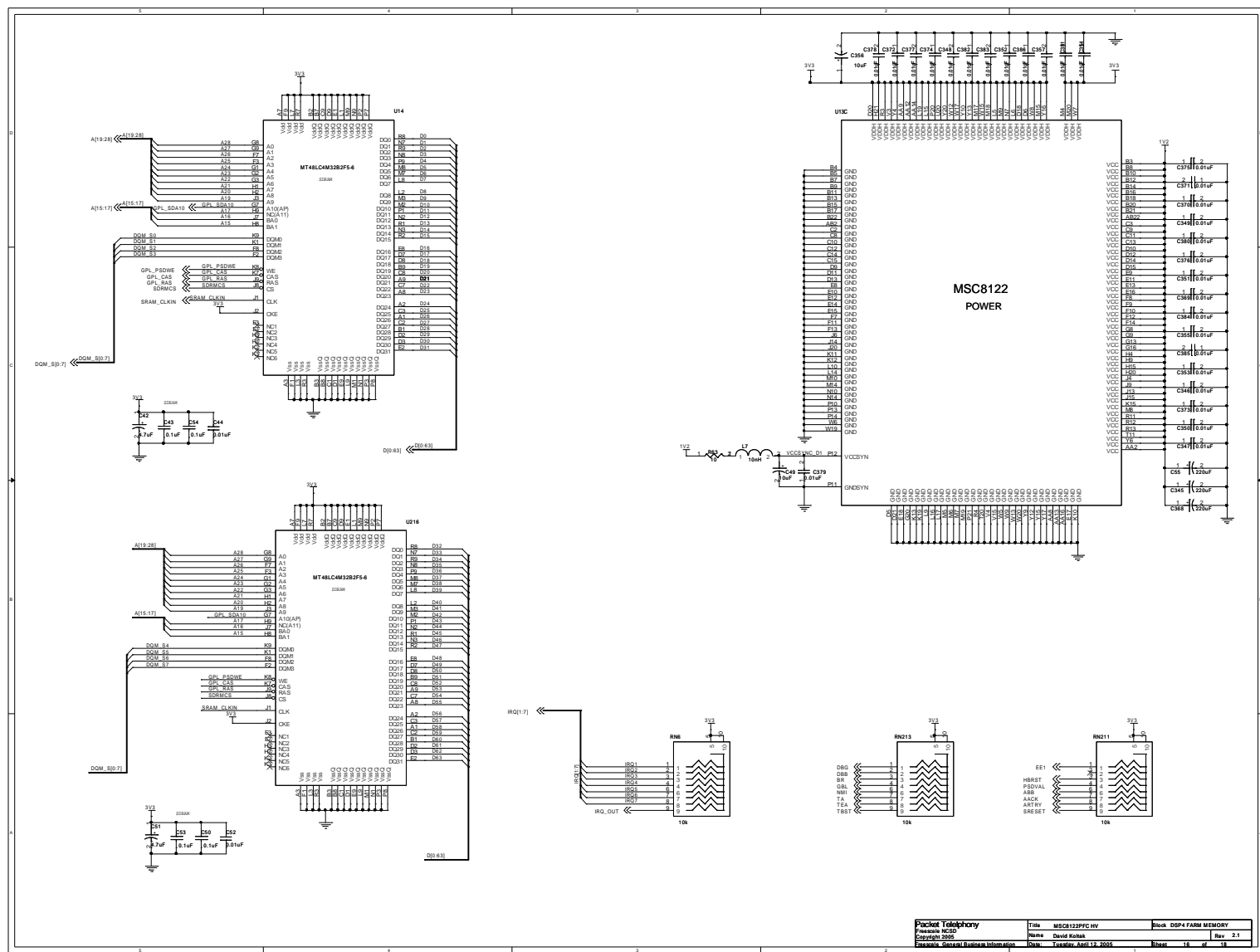


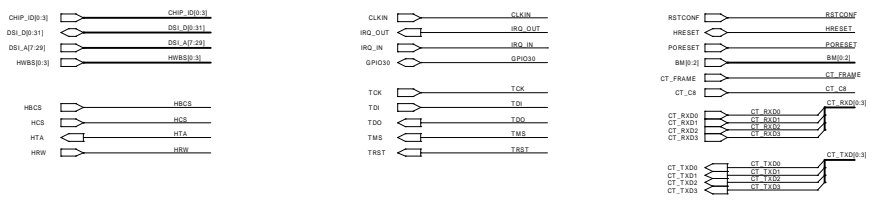
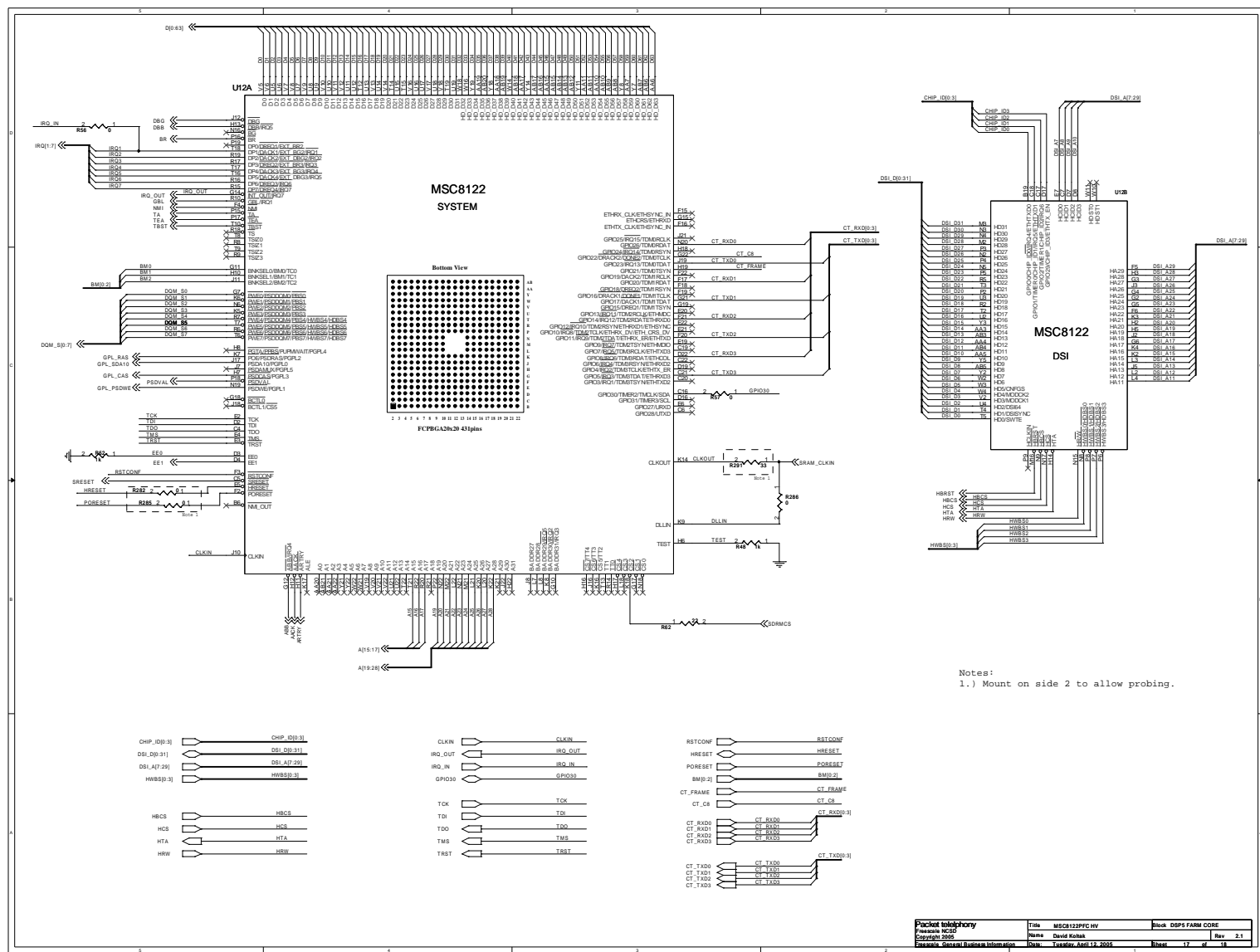


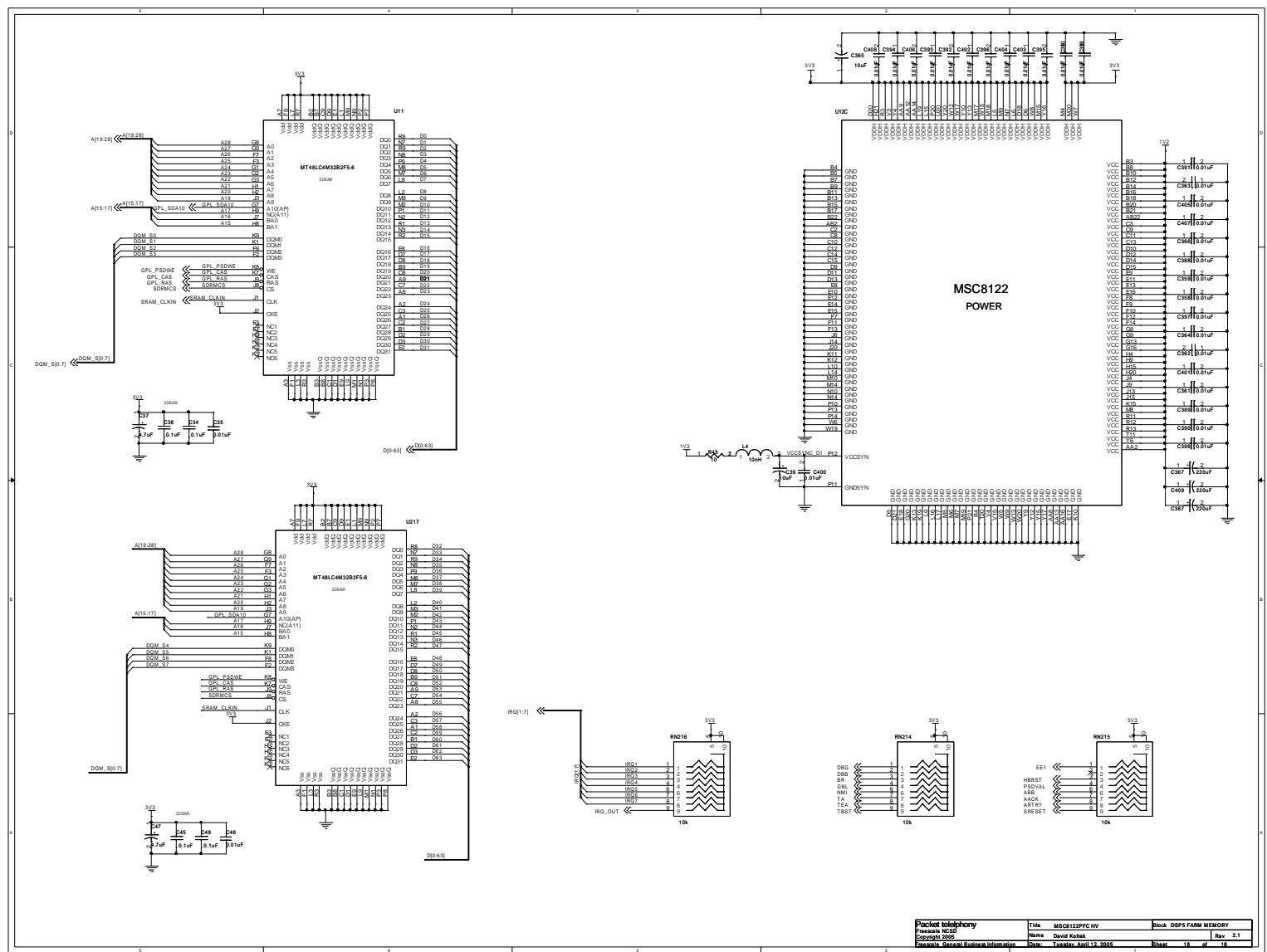














## How to Reach Us:

### Home Page:

[www.freescale.com](http://www.freescale.com)

### E-mail:

[support@freescale.com](mailto:support@freescale.com)

### USA/Europe or Locations not listed:

Freescale Semiconductor  
 Technical Information Center, CH370  
 1300 N. Alma School Road  
 Chandler, Arizona 85224  
 +1-800-521-6274 or +1-480-768-2130  
[support@freescale.com](mailto:support@freescale.com)

### Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GMBH  
 Technical Information Center  
 Schatzbogen 7  
 81829 München, Germany  
 +44 1296 380 456 (English)  
 +46 8 52200080 (English)  
 +49 89 92103 559 (German)  
 +33 1 69 35 48 48 (French)  
[support@freescale.com](mailto:support@freescale.com)

### Japan:

Freescale Semiconductor Japan Ltd.  
 Headquarters  
 ARCO Tower 15F  
 1-8-1, Shimo-Meguro, Meguro-ku,  
 Tokyo 153-0064, Japan  
 0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.  
 Technical Information Center  
 2 Dai King Street  
 Tai Po Industrial Estate  
 Tai Po, N.T. Hong Kong  
 +800 2666 8080

### For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center  
 P.O. Box 5405  
 Denver, Colorado 80217  
 1-800-441-2447 or 303-675-2140  
 Fax: 303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc. CodeWarrior is a trademark or registered trademark of Metrowerk Corp. in the US and/or other countries. StarCore is a licensed trademark of StarCore LLC. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2005.