Freescale Semiconductor
Design Workbook

P2020DS — a P2020 Development Platform

by
NMG System Design and Architecture
Freescale Semiconductor, Inc.
Austin, TX

Rev. 1.0 alpha
4/2009

1 Overview

P2020DS is a high-performance computing, evaluation and development platform supporting the **P2020** Power Architecture™ processor. P2020DS

P2020DS's official designation is "P2020DS", and may be ordered using this part number.

P2020DS is designed to the ATX form-factor standard, allowing it to be used in 2U rack-mount chassis', as well as in a standard ATX chassis. The system is lead-free and RoHS-compliant.

Contents

freescale™ semiconductor

2 Features

The features of the P2020DS development board are as follows:

- P2020 Processor
  - Core Processors
    - dual e500v2 cores at up to 1.2GHz
    - 45 nm SOI process technology
  - High-speed serial port (SERDES)
    - 4 lanes, dividable into combinations of x4, x2 and x1 lanes
    - supports PCIExpress, SGMII and Serial RapidIO
  - DDR memory controller
    - Designed for DDR3 support
    - One 240-pin socket for up to 4GB of 800 MHz DDR3
  - Triple-speed ethernet controller
    - Three 10/100/1G ports
    - Three ports use on-board VSC8244 PHY in RGMII mode
    - One port may be used in SGMII-mode with off-board card.
    - IEEE-1588 (PTP) support
  - Local bus
    - 128 MB NOR flash (fast boot)
    - 1GB NAND flash (slow boot/storage)
  - eSDHC
    - Connects to SDMedia card slot for boot code or mass storage
  - SPI
    - 16MB EEPROM device for boot code and storage
    - Connects to SDMedia card slot for boot code or mass storage
  - I2C
    - 2 controllers
    - I2C-based real-time clock and battery-backed SRAM
    - EEPROM storage for boot-sequencer, SystemID, ngPIXIS processor code, etc.
  - UART
    - Two serial ports at up to 115200 kbps
  - Miscellaneous features
    - DMA exerciser
  - GPIO
    - 16 GPIO pins user-accessible
  - Package
    - 31x31mm 689-pin 1mm pitch TEPBGA II (temperature-enhanced plastic BGA)
    - Socket supported
- NVidia M1575 South Bridge

  - Serial ATA 2 (RAID-1 Support)
  - Parallel ATA (CDROM support)
  - AC97 Audio support
  - Realtime Clock
  - 256 bytes of NVRAM
- System Logic
  - Manages system reset sequencing
  - Manages system clock and DDR clock speed selections
  - Controls system and monitoring
  - Implements registers for system control and monitoring
  - Internal 8-bit MCU allows independant VCore/temperature monitoring and reconfiguration.
- Clocks
  - System clock
    - SYSCLK switch settable to one of eight common settings in the interval 33MHz-166MHz.
    - Software settable in 1MHz increments from 1-200MHz.
  - DDR clock
    - DRCLK switch settable to one of eight common settings in the interval 33MHz-166MHz.
    - Software settable in 1MHz increments from 1-200MHz.
- Power Supplies
  - Dedicated VDD supplying VDD (CPU core power) and VPLAT
  - Dedicated SERDES power (nominal 1.05V).
  - GVDD (DDR power) and VTT/VREF adjustable for DDR3 or DDR2
  - 2.5V power for ethernet PHY

3 Block Diagram

Figure 1 shows the major functions of the **P2020**, while Figure 3 shows the overall architecture of the P2020DS system which surrounds it.

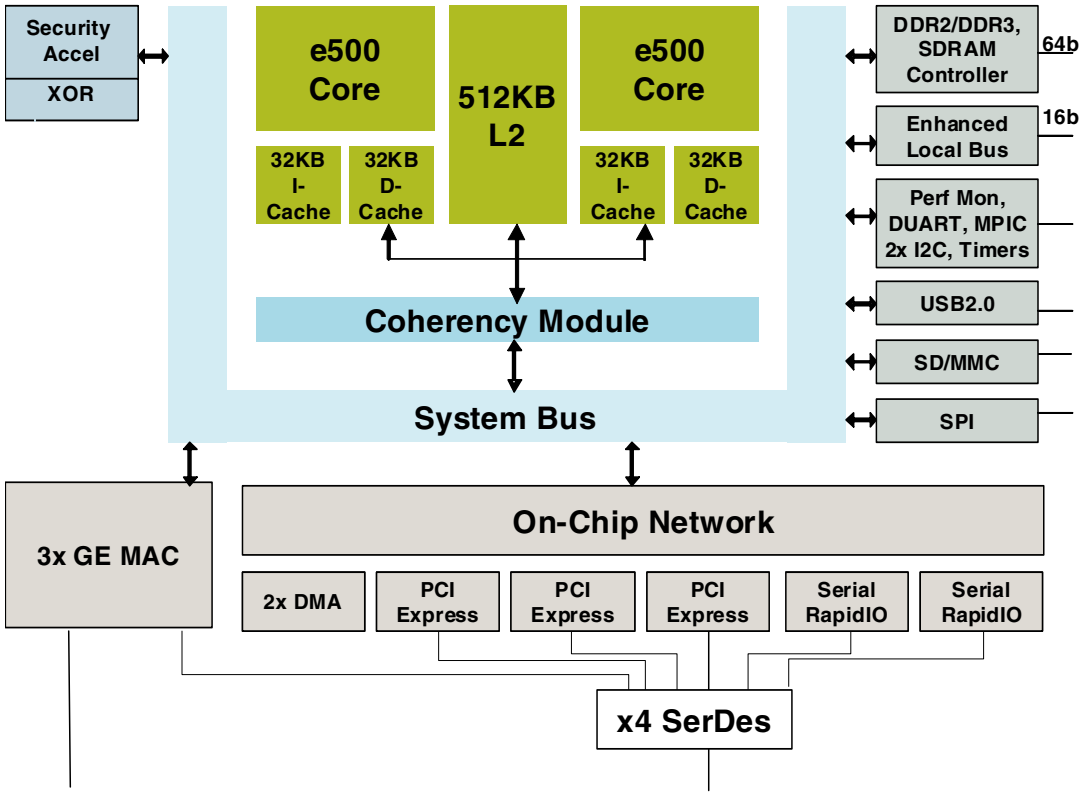

Figure 1. Block Diagram

4 Evaluation Support

4.1 Development System Use

P2020DS is intended to evaluate as many features of the **P2020** as are reasonable within a limited amount of board space and cost limitations.

For general hardware and/or software development and evaluation purposes, P2020DS can be used just like an ordinary desktop computer. In the absence of special hardware or software configuration, P2020DS operates

identically to a development/evaluation system such as ArgoNavis(8641DS) or other members the HPC family. Figure 2 shows an example of P2020DS system in a desktop configuration:

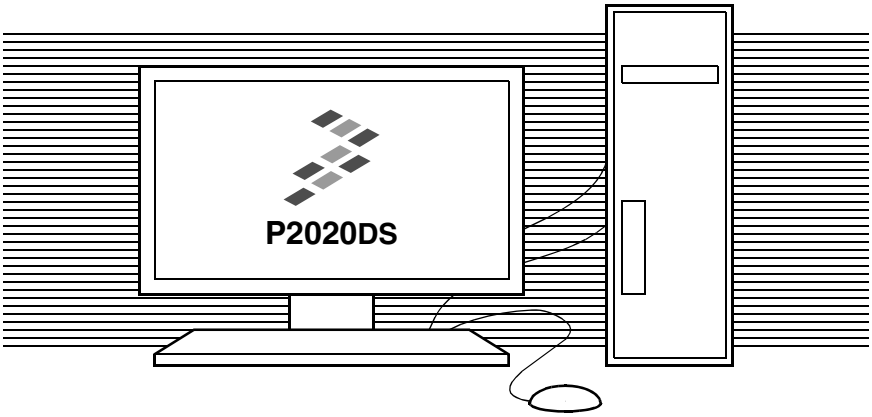

Figure 2. P2020DS **Desktop Configuration**

4.2 Rackmount Server Use

For use in a rackmount chassis, P2020DS requires the following modifications:

- low-profile heatsink
- non-socketed board

Otherwise, it is similar to the desktop case.

4.3 Embedded Use

For general embedded hardware and/or software development and evaluation purposes, P2020DS can be used just like an ordinary desktop computer. The core voltage and PLL settings might be adjusted to allow the heatsink to be replaced or even removed. Perpiherals and embedded storage can be connected to the PromJet superset connector.

As before, the **ngPIXIS** is used to provide startup configuration information for DINK, UBOOT or Linux and other advanced features are used or ignored.

4.4 AVP-controlled Evaluation

For many test situations, it is desireable to download a test vector program, and run the results. P2020DS can do this by using a PCI-based control card such as the DataBlizzard or a PCI-Express based control card such as “Komodo”;

either stand-alone, or in coordination with the **ngPIXIS**. Table 1 lists an overview of the steps required to accomplish this..

Table 1. AVP Execution Steps

Step	Details	
Assert Target Reset	Set target reset	Control card asserts “flying lead” reset line; alternately the **ngPIXIS** register bit PX_RST[RSTL] is set to ‘0’.
	Target processor (not the system) is reset.	
Setup New Target Environment	Set target core VDD	VCTL[VCORE]=1 VCORE=xxxxxxxx
	Set requested SYSCLK	VCTL[SYSCLK]=1 VSPEED[SYS]=xxx
Restart Target	Set target reconfiguration	VCTL[GO]=1
	System is reconfigured, target processor is remains in reset. This may take several milliseconds	
Download Target	Download to target execution space. Presumably the DDR and PCIExpress resources were configured by the I2C sequencer. If so, a PCIMaster such as the DataBlizzard can simply write test code to system memory via PCI->DDR path.	
Release Target Reset	Release target reset	Control card deasserts “flying lead” reset line; alternately the **ngPIXIS** register bit PX_RST[RSTL] is set to ‘1’.
	Target processor executes code.	
Collect Results	Results can be extracted from system DDR, PCIExpress graphics memory (used as a buffer), or other memory (SDMedia, flash, PromJet)	

5 Architecture

The P2020DS architecture is primarily determined by the Freescale Semiconductor **P2020** Power Architecture(TM) processor, and by the need to provide typical OS-dependant resources (disk, ethernet, etc.).

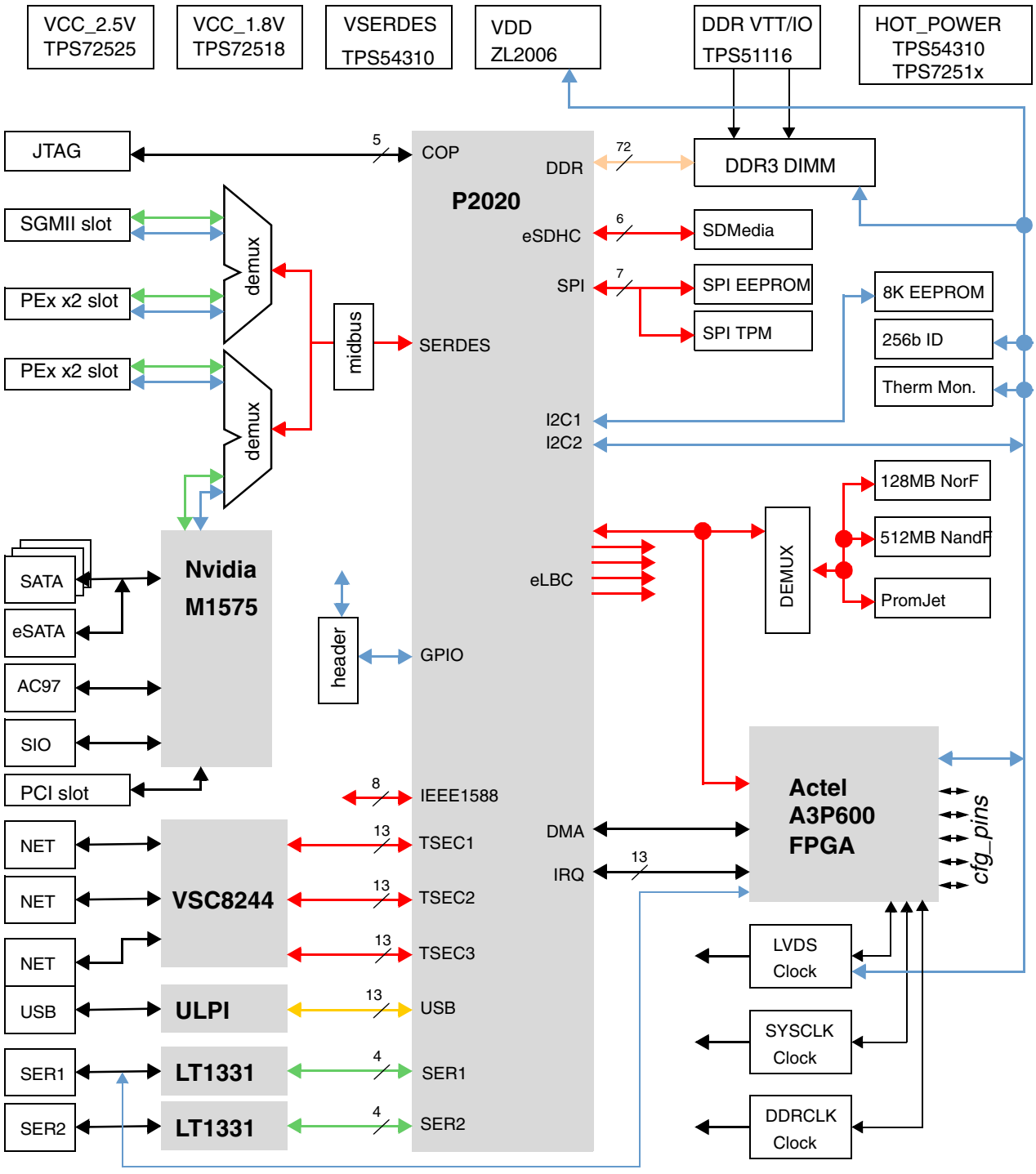

Figure 3. Detailed Block Diagram

Table 2 summarizes some of the unique components used on P2020DS.

Table 2. P2020DS Principal Components

Category	Component	Link	New? (where used)	Notes
Processor	Freescale **P2020**	**P2020**	Yes	
Power	ZilkerLabs ZL2006	ZL2006	Yes	VDD
	TI 2.5V Power	TPS72525	No (Argo)	LVDD (Enet) 2.5V
	TI 1.2V Power		No (FCL)	Phy Core power
	TI 1.0V TPS54910	tPS54910	No (Argo)	VSERDES (XVDD + SVDD)
	TI TPS51116	TPS51116	No (Argo)	GVDD/VTT/MVREF
	TI TPS54310	TPS54310	No (Argo)	HOT 3.3V
	TI TPS72525	TPS72525	No (Argo)	HOT 2.5V
	TI TPS72518	TPS72518	No (Argo)	HOT 1.8V
	TI TPS72515	TPS72515	No (Argo)	HOT 1.5V
DDR3 Socket	n/a	n/a	No (Calamari)	
Serdes	Pericom LVDS Mux	PI2PCIE2412	Yes	
	SGMII Riser Slot		No (Intrepid)	
	PCI Express Slot		No (Argo)	
Ethernet	Vitesse PHY	VSC8244XHG	No (Argo)	
	BelFuse dual MagJack	0845-2R1T-E4	No (Argo)	
	Ethernet over USB		No (Lyra)	
IEEE-1588	Reference Clock	E13C7E2F-100.000M	No (Calamari)	Precision reference
Local Bus	Demux control	SN74ALVCH32973	No (Argo)	Local bus address latch/buf
	Spansion 128MB Flash	S29GL01GP11FFI010		NOR-type flash
	Spansion 1Gb Flash	K9NBG08U5M		NAND-type flash
USB	SMSC USB Transceiver	USB3300-EZK	No (Calamari)	ULPI transceiver
	Micrel USB Power	MIC2076-1YM	No (exists)	USB power
	Crystal	ECCM7AA10-24.000M	No (Calamari)	USB Clock
	Ethernet over USB		No (Lyra)	
eSDHC	SDMedia Connector		No (Lyra)	
SPI	SDMedia Connector		No (Lyra)	(SDHC using SPI mode)
	Spansion 16MB EEPROM		No (Calamari)	
Clock	IDT ICS951412	ICS951412A	No (Lyra)	System + PEX clocks
	IDT ICS307	ICS307M02LF	No (Lyra)	Serial SYSCLK/DDRCLK
System Controller	Actel FPGA	A3P600-FGG484FSL	Yes	System Controller
Serial	LinearTech Transceiver	LT1331CSW#PBF	No (Lyra)	Serial ports
I2C	Atmel 24C64	24C64	No (Lyra)	8K boot EEPROM
	Atmel 24C02	24C02	No (Lyra)	System ID/MAC EEPROM
	Analog ADT7461	ADT7461ARZ	No (Lyra)	Thermal diode monitor
South Bridge	Nvidia M1575	M1575	No (Argo)	South bridge

Table 2. P2020DS Principal Components

Category	Component	Link	New? (where used)	Notes
PCI	n/a	n/a	No (Argo)	PCI Slots
AC97 Audio	Realtek ALC655	ALC655-LF	Yes	

5.1 Processor

P2020DS supports the Freescale Semiconductor **P2020** processor. Table 3 lists the major pin groupings of the **P2020**.

Table 3. P2020 Summary

Signal Group	Pin Count	Details
Memory Controller	150	Section 5.1.1
SerDes x4	28	Section 5.1.2
Ethernet	53	Section 5.1.3
IEEE 1588	8	Section 5.1.4
Local Bus	56	Section 5.1.5
eSDHC	6	Section 5.1.6
SPI	7	Section 5.1.7
USB	13	Section 5.1.8
DMA	6	Section 5.1.9
eOpenPIC	17	Section 5.1.10
GPIO	16	Section 5.1.11
System Control	7	Section 5.1.12
UART	8	Section 5.1.13
I2C	4	Section 5.1.14
Debug/Power Management	11	Section 5.1.15
Clock	3	Section 5.1.16
JTAG/Test	9	Section 5.1.17
Thermal	2	Section 5.1.18
Power	280	Section 5.1.19
reserved	5	
TOTAL	**689**	

5.1.1 DDR

The **P2020** contains a memory controller capable of supporting DDR1, DDR2 and DDR3 devices. P2020DS supports DDR-3 only, using industry-standard DDR3 DIMM modules, for a maximum total of 4GB of memory. The memory interface includes all the necessary termination and I/O power and is routed so as to achieve maximum performance on the memory bus. In particular, the DDR components are placed and routed so as to achieve 2T timing with unbuffered DIMMs at 800MHz or faster. 1T timing may be possible if the DIMM is lightly loaded (one rank only, with wide (16-bit) components).

The general DDR memory architecture is shown in Figure 4.

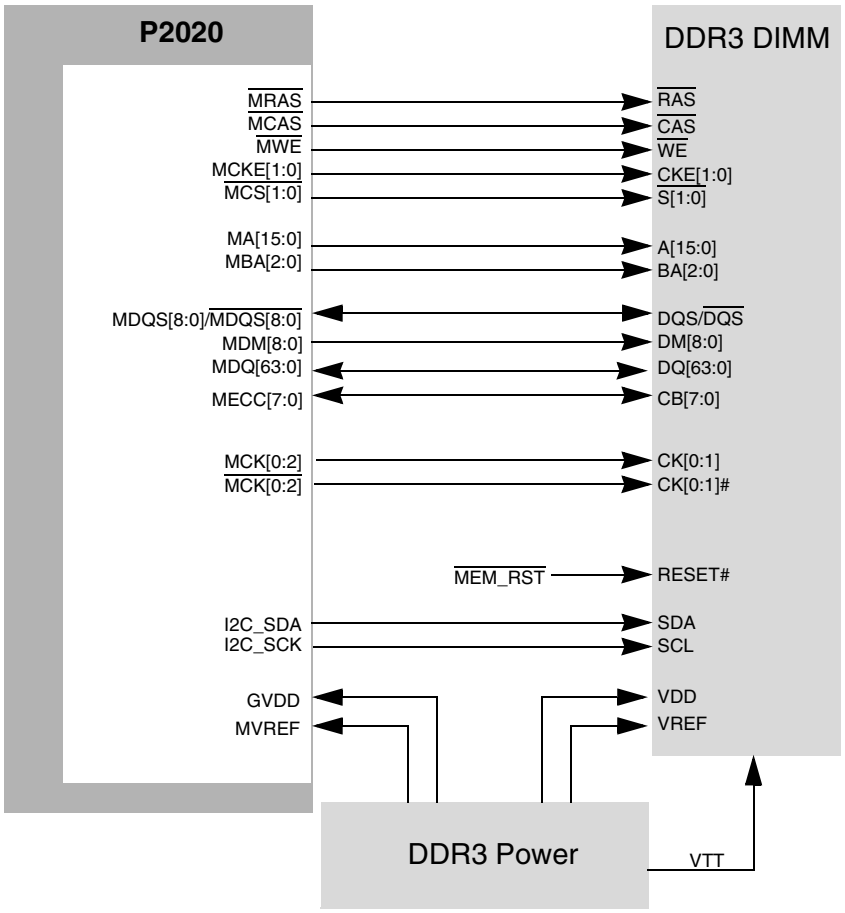

Figure 4. P2020DS **Memory Architecture**

Note also that P2020DS does not directly support the use of the MECC pins to access internal debug information, as P2020DS does not provide the special multiplexer and thus has a simpler routing and signal integrity status. On the other hand, P2020DS does not interfere with this path, so access to debug information on the MECC pins is possible with the use of a NextWave (or equivalent) DDR logic analyzer connector and the use of non-ECC DDR modules.

32-bit DDR3 interface mode is supported; from the viewpoint of the P2020DS board, the unused lower MDQ/MDS/MDM signals are simply inactive.

The DDR3 power supply the following interface voltages:

- VDD_IO up to 10W (6A at 1.5V nominal)
- VDDQ+VTT up to 3A
- MVREF up to 10mA

The DDR memory port signals and connections are summarized in Table 4.

Table 4. DDR Memory Connections

Pin Count	Signal Names	Connections
64	MDQ[0:63]	**P2020**, DIMM
8	MECC[0:7]	**P2020**, DIMM
9	MDM[0:8]	**P2020**, DIMM
18	MDQS[0:8](p,n)	**P2020**, DIMM
3	MBA[0:2]	**P2020**, DIMM
16	MA[0:15]	**P2020**, DIMM
1	MWE_B	**P2020**, DIMM
1	MRAS_B	**P2020**, DIMM
1	MCAS_B	**P2020**, DIMM
2	MCS_B[0:1]	**P2020**, DIMM
2	MCS_B[2:3]	unused
2	MCKE[0:1]	**P2020**, DIMM
2	MCKE[2:3]	unused
6	MCK[0:2](p,n)	**P2020**, DIMM
6	MCK[3:5](p,n)	unused
2	MODT[0:1]	**P2020**, DIMM
2	MODT[2:3]	unused
2	MDIC[0:1]	MPC8610
1	MVREF	**P2020**, DIMM
1	MAPAR_ERR*	
1	MAPAR_OUT	
150	Total pins in this group	

5.1.1.1 Compatible DDR-3 Modules

The DDR interface of P2020DS and the **P2020** should work with any JEDEC-compliant 240-pin DDR-3 DIMM module. Table 5 shows several DIMM modules which are believed compatible; those which have been tested and confirmed are noted as such.

Table 5. DDR-3 Modules

Mfg.	Part Number	Size	Ranks	ECC	Data Rate	Verified?	Notes
Elpida	EBJ21EE8BAFA-AE-E	2 GB	2	Y	1066	Yes	Or later revs.

5.1.2 SerDes x4

The SerDes block provides high-speed serial communications interfaces for several internal devices. The SerDes block provides 4 serial lanes which may be partitioned as shown in the following Table 6:

Table 6. SerDes Partition Capabilities

cfg_io_ports(0:3)	Lane				Supported?	Communications Rate	
	0 (A)	1 (B)	2 (E)	3 (F)		A+B	E+F
0000	PEX1: x1	off	off	off	Yes	2.5 Gb	-
0010	PEX1: x1	PEX2: x1	PEX3: x2		Yes	2.5 Gb	2.5 Gb
0100	PEX1: x2		PEX3: x2		Yes	2.5 Gb	2.5 Gb
0110	PEX1: x4				No	2.5 Gb	
1000	SRIO1: x4				No	1.25 Gb	
1001	SRIO1: x4				No	2.5 Gb	
1011	PEX1: x1	PEX2: x1	SGMII2	SGMII3	Yes	2.5 Gb	1.25 Gb
1100	SRIO1: x1	SRIO2: x1	SGMII2	SGMII3	No	1.25 Gb	1.25 Gb
1101	SRIO1: x1	SRIO2: x1	SGMII2	SGMII3	No	2.5 Gb	1.25 Gb
1110	PEX1: x1	SRIO2: x1	SGMII2	SGMII3	No	2.5 Gb	1.25 Gb
1111	PEX1: x2		SGMII2	SGMII3	Yes	2.5 Gb	1.25 Gb

Note that the term “lane” is used to describe the minimum number of signals needed to create a bidirectional communications channel; in the case of PCI Express or Serial RapidIO, a lane consists of two differential pairs, one for receive and one for transmit, or four in all.

In order to make the maximal amount of use from these lanes, high-speed LVDS multiplexers from Pericom are used to route the SERDES lanes to various destinations. The insertion loss from using these devices is guaranteed to less than 2dB. To keep the routing simple, and to avoid multiple levels of multiplexing, the PEX x4 and SRIO x4 facilities are not supported. When those cases are eliminated, lanes 0&1 can be treated as one independant set, while lanes 2&3 can be treated as a second.

Figure 5 shows an overview of the routing for lanes 0 & 1.

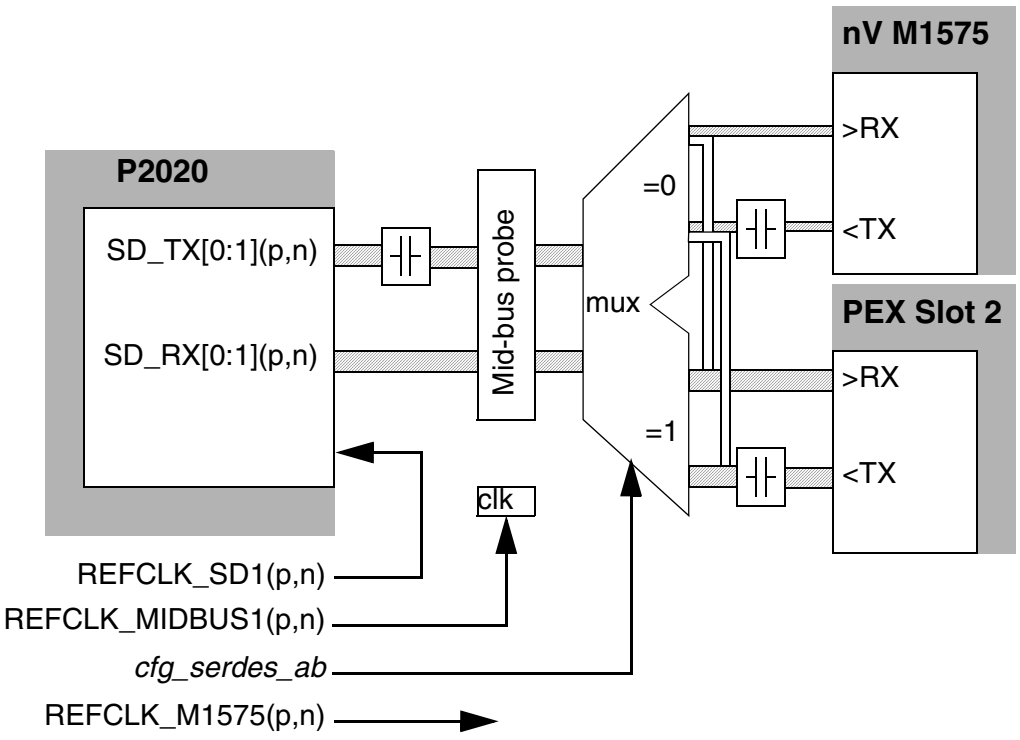

Figure 5. SerDes x4 Lanes 0&1 Routing

Lanes 0 & 1 are routed in a slightly more complicated manner as compared to lanes 2 & 3; both lanes are sent through the mid-bus probe to the mux, where they are routed as a pair to the PCI Express slot #2; OR, lane 1 is split off and sent to the nVIDIA M1575 while lane 0 remains with the PCI Express slot. This is summarized in Table 7:

Table 7. SerDes Lanes 0&1 Routing Summary

cfg_serdes_ab	Source Lane		Mode Description
	0 (A)	1 (B)	
0	SLOT 2 RX/TX0	M1575 RX/TX0	Slot 2 in PEX x1 mode, nVidia available
1	SLOT 2 RX/TX0	SLOT 2 RX/TX1	Slot 2 in PEX x2 mode, nVidia not available

To keep propagation time and effects constant between lanes, lane 0 is passed through the multiplexer even though it ends up at the same location. This requires specialized PCB routing and constaints.

Figure 6 shows an overview of the routing for lanes 2 & 3.

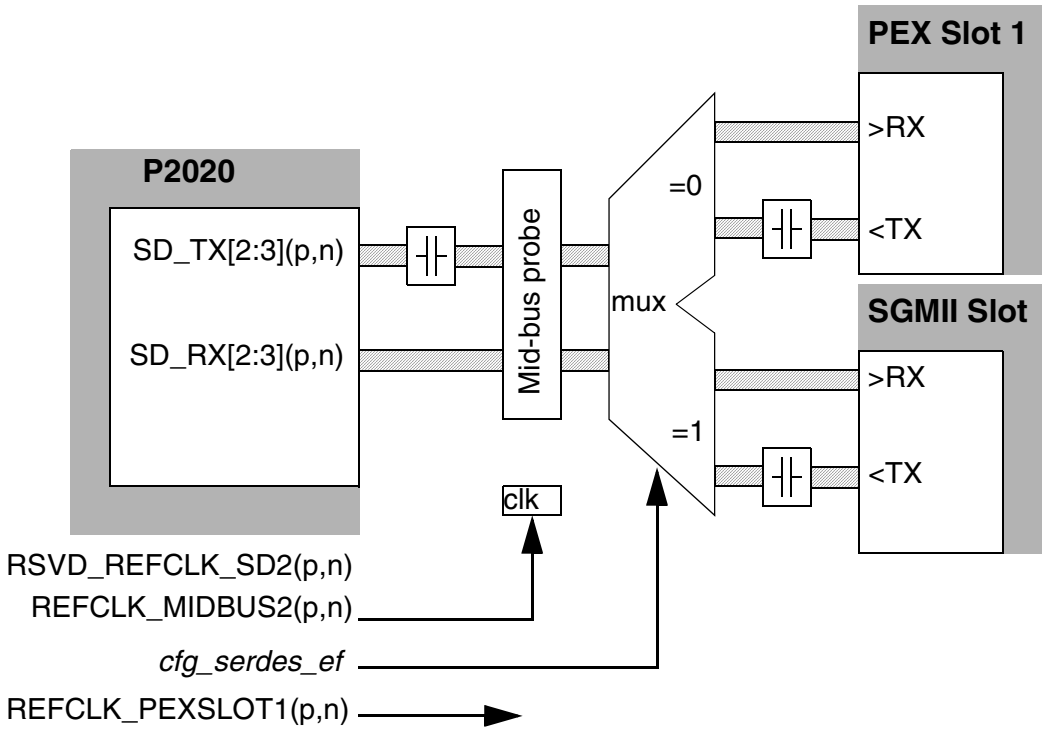

Figure 6. SerDes x4 Lanes 2&3 Routing

Lanes 2 & 3 routing is relatively simple; both lanes are sent through the mid-bus probe to the mux, where they are routed as a pair to either the SGMII slot, or to the PCI Express slot 1.

SerDes connections are summarized in Table 8.

Table 8. SerDes x4 Port Connections

Pin Count	Signal Names	Connections
8	SD_RX[0:3](p,n)	**P2020** PI2PCIE2422 mux
8	SD_TX[0:3](p,n)	**P2020** PI2PCIE2422 mux
2	SD_REFCLK(p,n)	**P2020**
2	SD_TXCLK(p,n)	unused
2	SD_PLL_TPA SD_PLL_TPD	**P2020** Test point
2	SD_IMP_CAL_TX SD_IMP_CAL_RX	calibration resistors
24	Total pins in this group	

5.1.3 Ethernet (TSEC)

The **P2020** supports up to three 10/100/1000baseT triple-speed Ethernet controllers (TSEC). These controllers may be routed internally to one of three GMAC ports, where P2020DS connects them to the on-board Vitesse VSC8244 quad-PHY (the fourth port is unused) using the RGMII protocol. Alternately, TSEC2 and/or TSEC3 may be independantly connected to the SGMII interface, where P2020DS, routes them to connect to a multi-channel SGMII ethernet card (sold separately).

The management interface (MI) connects to both on-board PHYs and SGMII-card PHYs.

Connections and routing is summarized in Table 9.

Table 9. Ethernet Port Locations

P2020 TSEC #	Connection Port	PHY Address	Location	Notes
1	ETSEC	0	Top port of RJ45 stack	
	SGMII			
2	ETSEC	1	Bottom port of RJ45 stack	
	SGMII			
3	ETSEC	2	Top of combo USB/RJ45 stack	
	SGMII	TBD	Top port of card	

The Ethernet port signals are summarized in Table 10.

Table 10. Ethernet Port Connections

Pin Count	Signal Names	Category	Connections
2	EC_MDC, EC_MDIO	Management	**P2020**, VSC8244
1	EC_GTX_CLK125	Clocking	**P2020**, VSC8244
12	TSEC1_TXD(3:0) TSEC1_TX_EN TSEC1_GTX_CLK TSEC1_RXD(3:0) TSEC1_RX_DV TSEC1_RX_CLK	TSEC1	**P2020**, VSC8244
12	TSEC2_TXD(3:0) TSEC2_TX_EN TSEC2_GTX_CLK TSEC2_RXD(3:0) TSEC2_RX_DV TSEC2_RX_CLK	TSEC2	**P2020**, VSC8244

Table 10. Ethernet Port Connections

Pin Count	Signal Names	Category	Connections
15	TSEC1_TXD(7:4) (TSEC3_TXD(3:0) TSEC2_TXD4 (TSEC3_TXCLK) TSEC2_TXD5 (TSEC3_TXCTL) TSEC1_TX_ER TSEC1_TX_CLK TSEC1_CRS (TSEC3_TXCTL) TSEC1_COL (TSEC3_TXCLK) TSEC1_RXD(7:4) (TSEC3_RXD(3:0)) TSEC1_RX_ER	TSEC3	**P2020**, VSC8244
11	TSEC2_TXD(7:6) TSEC2_TX_ER TSEC2_TX_CLK TSEC2_CRS TSEC2_COL TSEC2_RXD(7:4) TSEC2_RX_ER	TSEC2	**P2020**, ground or n/c (excluding config-pin use)
53	Total pins in this group		

The general organization of the ethernet system is shown in Figure 7.

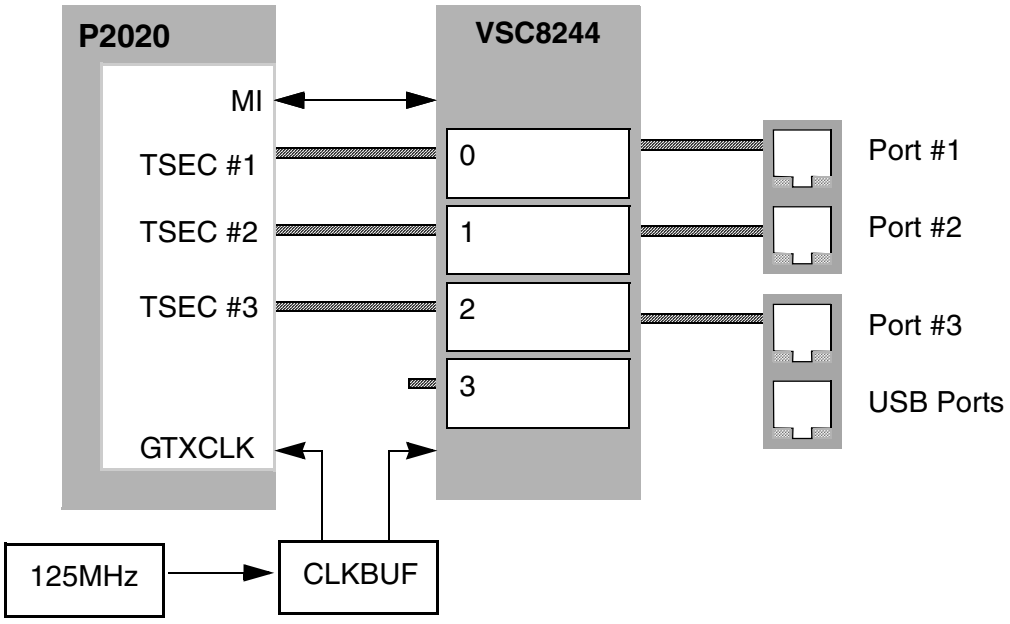

Figure 7. Ethernet Architecture

P2020DS uses the ICS8304AMLF to drive the ethernet GTX clocks with the correct edge rate at 2.5V.

Refer to the Vitesse website for programming information for the VSC8244 PHY, or use existing drivers as used with the ArgoNavis (HPCN8641DS) and/or Intrepid (MPC8544DS).

5.1.4 IEEE 1588

The **P2020** includes support for the IEEE 1588™ Precision Time Protocol (PTP). This facility works in tandem with the Ethernet controller to time-stamp incoming packets.

Figure 8 shows an overview of this block.

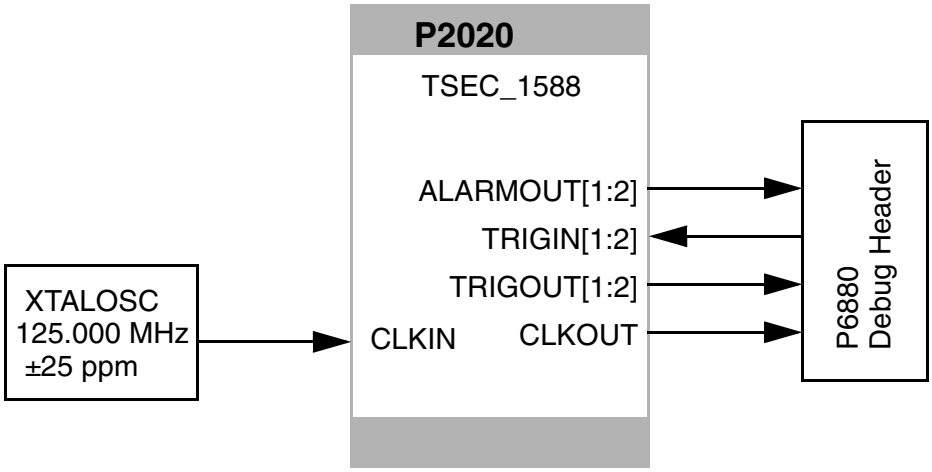

Figure 8. IEEE-1588 Interface Overview

The IEEE-1588 signals are summarized in Table 11.

Table 11. IEEE-1588 Support Connections

Pin Count	Signal Names	Connections
1	TSEC_1588_CLKIN	**P2020**, PTP reference clock
2	TSEC_1588_TRIG_IN[1:2]	**P2020**, Debug header
2	TSEC_1588_PULSE_OUT[1:2]	**P2020**, Debug header
1	TSEC_1588_CLKOUT	**P2020**, Debug header
2	TSEC_1588_ALARM_OUT[1:2]	**P2020**, Debug header
8	Total pins in this group	

5.1.5 Local Bus

The eLBC (embedded Local Bus Controller) is relatively simple. For P2020DS, the local bus connects to various flash devices and the **ngPIXIS** internal register space. The **P2020** only supports 16-bit devices, so the eLBC interface is comparitively simpler than past development systems. In particular, a single 16-bit latch/buffer is used to latch the portion of the address that is not already provided by the latched address pins, and also to buffer the data.

Figure 9 shows an overview.

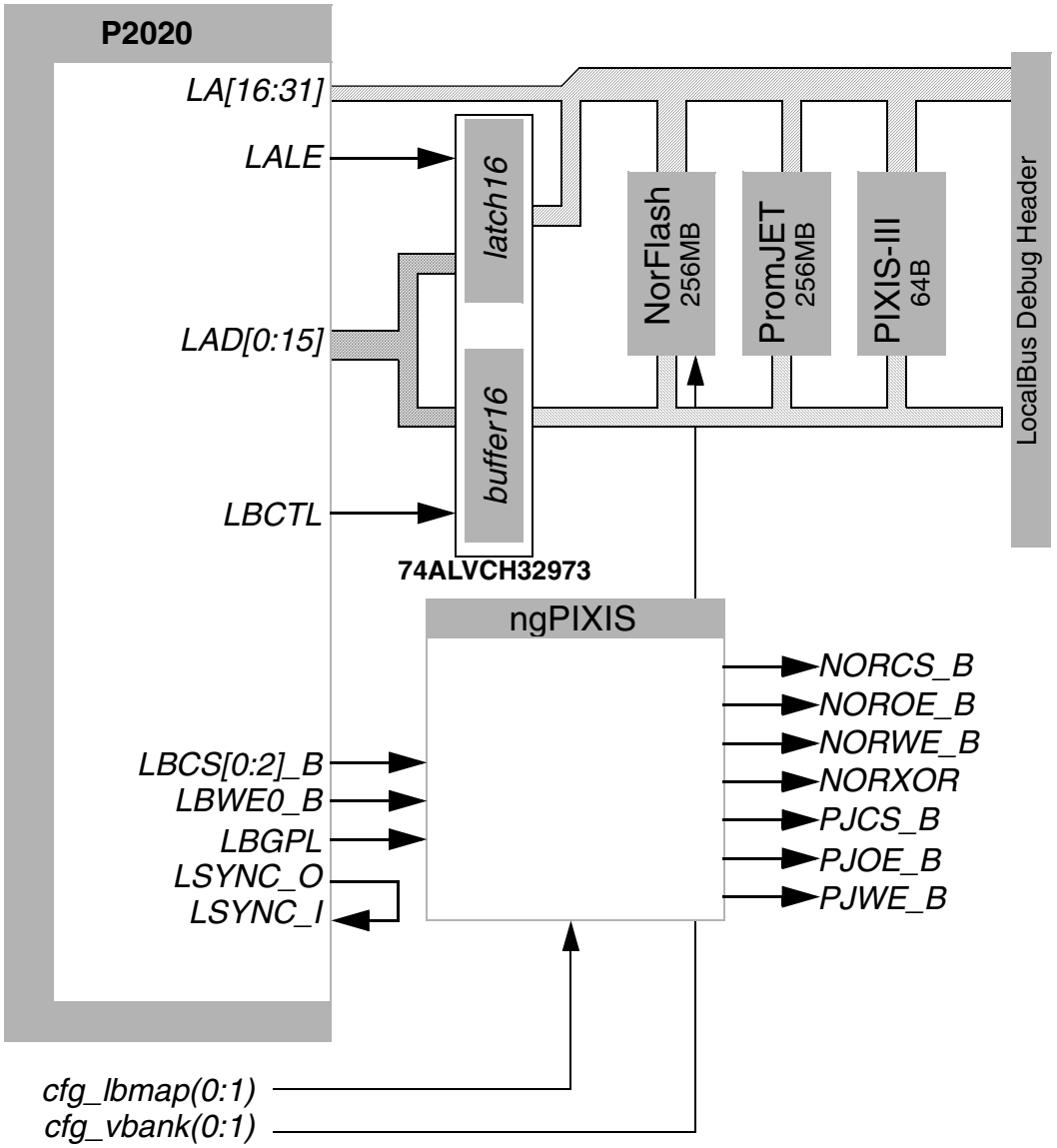

Figure 9. Local Bus Overview

Local bus signals are summarized in Table 12.

Table 12. Local Bus Connections

Pin Count	Signal Names	Connections
16	LAD[0:15]	**P2020** Address/data latch/buffer Debug header

Table 12. Local Bus Connections

Pin Count	Signal Names	Connections
2	LDP[0:1]	**P2020** Debug header
16	LA[16:31]	**P2020** NORFlash PromJET **ngPIXIS** Debug header
8	LCS[0:7]_B	**P2020** **ngPIXIS** Debug header
1	LWE0_B / LBS0*	**P2020** **ngPIXIS** Debug header
3	LWE1_B / LBS1_B	**P2020** **ngPIXIS** Debug header
1	LBCTL	**P2020** Data bus buffer Debug header
1	LALE	**P2020** Address latch Debug header
1	LGPL0	**P2020** **ngPIXIS** Debug header
1	LGPL1	**P2020** **ngPIXIS** Debug header
1	LGPL2 / LOE_B	**P2020** **ngPIXIS** Debug header
1	LGPL3	**P2020** Debug header
1	LGPL4 / LUPWAIT_B / LGTA_B	**P2020** **ngPIXIS** Debug header
1	LGPL5	**P2020** Debug header

Table 12. Local Bus Connections

Pin Count	Signal Names	Connections
2	LCLK[0:1]	**P2020** **ngPIXIS** Debug header
1	LSYNC_OUT	**P2020**
1	LSYNC_IN	**P2020**
56	Total pins in this group	

The **P2020** can redirect boot fetches to the eLBC, where it is routed to the device attached to LCS0_B. To support greater flexibility, the **ngPIXIS** can re-route the LCS0_B pin to other devices, allowing the P2020DS to boot from the following devices:

- NORFlash
- NORFlash with MSB[0:1] address lines XOR'd (virtual bank swapping)
- PromJet

The eLBC chip-select connections are summarized in Table 13.

Table 13. Local Bus Chip Select Mapping

Chip Select	cfg_lbmap	cfg_vbank	Destination	Description
0	0X	00	NORFlash	Boot from NORFlash (default).
		01		Boot from NORFlash (A1 inverted)
		10		Boot from NORFlash (A0 inverted)
		11		Boot from NORFlash (A0:1 inverted)
	1X	XX	PromJet	Boot from PromJet. PromJets do not have address swapping.
1	0X	XX	PromJet	PromJet. Note that PromJets do not have address swapping.
	1X	00	NORFlash	Access NORFlash.
		01/10/11	NORFlash	Access bank-swapped NORFlash.
3	XX	XX	PIXIS-III	Internal registers.
2, 4-7	*reserved*			

The "cfg_vbank" column mentioned in Table 13 is used to rearrange the internal addresses of NOR flash devices, based upon user configuration options. Simplistically, no matter what state the switches are in, to the end-user toggling the switch results in toggling the halves or quarters of the NOR flash and toggling the CS lines of the NAND flash. If different program images are stored therein, upon reset different startup code will be executed.

For NOR flashes, which have one chip select but a large number of address pins, the “CFG_VBANK[0:1]” signals drive a pair of XOR gates in-line with the most-significant address bits of the NORFlash, as shown in Figure 10.

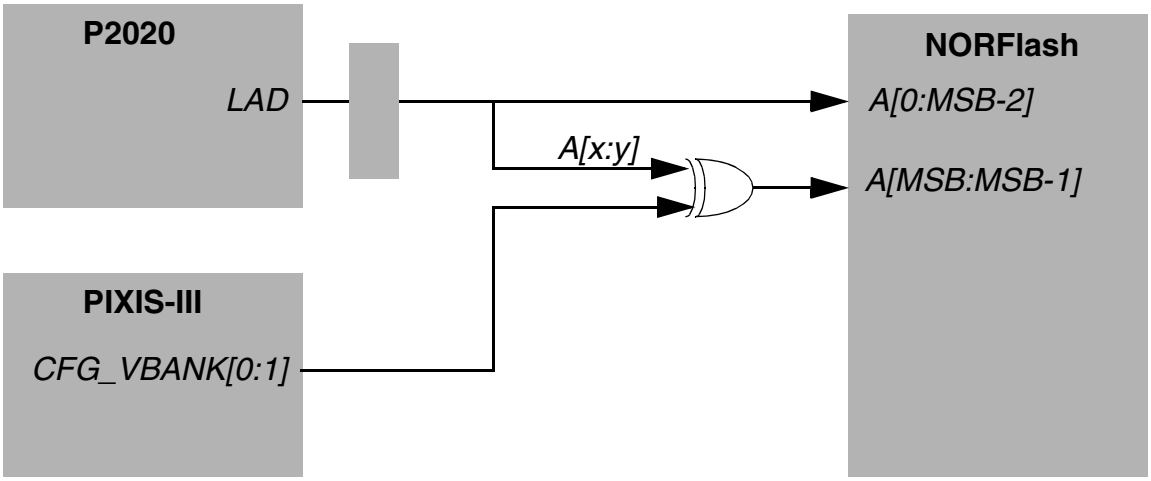

Figure 10. Flash Address Toggle

When CFG_VBANK[0:1] is “00”, A[MSB:MSB-1] is not altered at all, and so the flash behaves as normal. When CFG_VBANK[0:1] is “10”, A[MSB] is toggled such that data in the high half of the flash appears at the bottom, and vice-versa. A similar process applies to CFG_VBANK[1] and A[MSB-1]. The end-result is that the 128MB flash can be partitioned into four 32MB boot images, or two 64MB images, based upon how CFG_VBANK[0:1] are used.

Note that CFI flash programming algorithms do not use higher address bits of flash devices, so program/erase algorithms are not affected.

NOTE: To meet LALE/LAD setup and hold time restrictions, at high platform speeds (>500 MHz), additional PCB trace delay will be required for the LAD(0:15) bus.

5.1.6 eSDHC

The **P2020** has an enhanced secure digital host controller (eSHDC). P2020DS connects this to a SDMedia card slot, and uses GPIO signals for sideband signals such as write-protect-detect and card-detect. Both x4 and x8 cards are supported; the latter using the SPI_CS_B[0:3] signals which can be reassigned as eSHDC_D[4:7].

Figure 16 shows the overall connections of the eSDHC block.

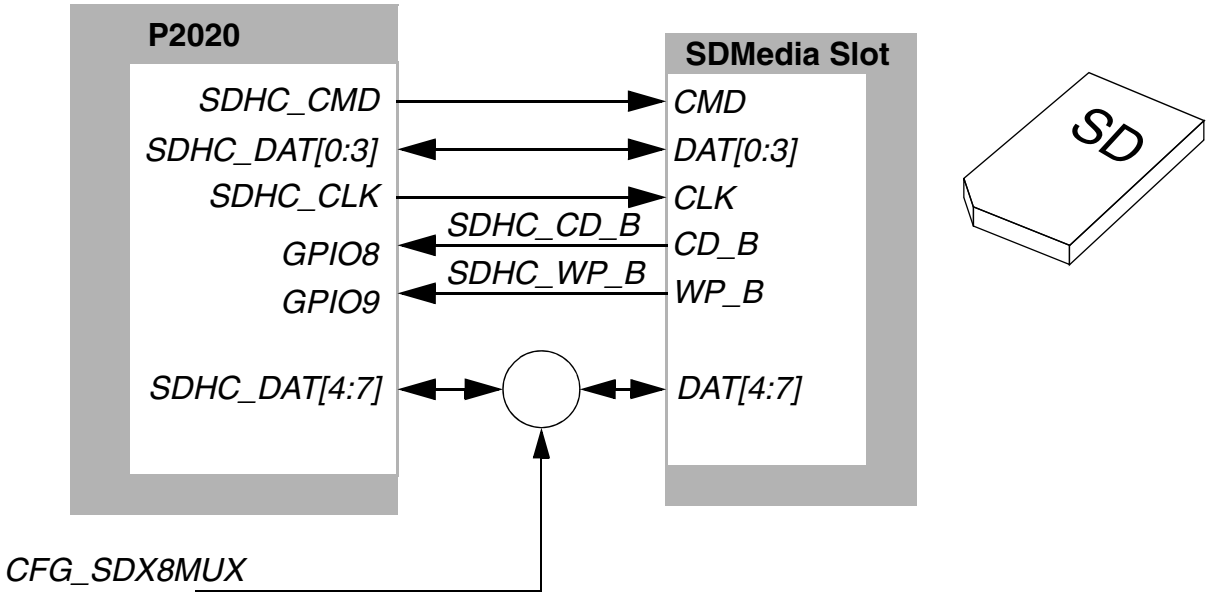

Figure 11. eSDHC Architecture

eSDHC port signals are summarized in Table 14.

Table 14. eSDHC Connections

Pin Count	Signal Names	Connections
1	SDHC_CMD	**P2020**, SDMediaSlot
4	SDHC_DAT[0:3]	**P2020**, SDMediaSlot
-	SDHC_DAT[4:7]	**P2020**, multiplexer
1	SDHC_CLK	**P2020**, SDMediaSlot
6	Total pins in this group	

The SDHC_DAT[4:7] signals are shared with the SPI CS pins; software may select the routing of those pins to either the SDHC devices or the SPI devices; both cannot be used simultaneously.

In addition, the following GPIO signals are used (see Section 5.1.11):

- GPIO8 - eSDHC card detect
- GPIO9 - eSDHC card write protect

5.1.7 SPI Interface

The **P2020** has a Serial Perpheral Interface (SPI), which is used to communicate with various peripherals. P2020DS connects a conventional 16MB serial EEPROM to one chip select, and an SPI-based SDMedia card slot (connected for MMC/SPI interfacing) to a second. The remaining two chip-selects are unused.

Figure 12 shows the overall connections of the SPI portion.

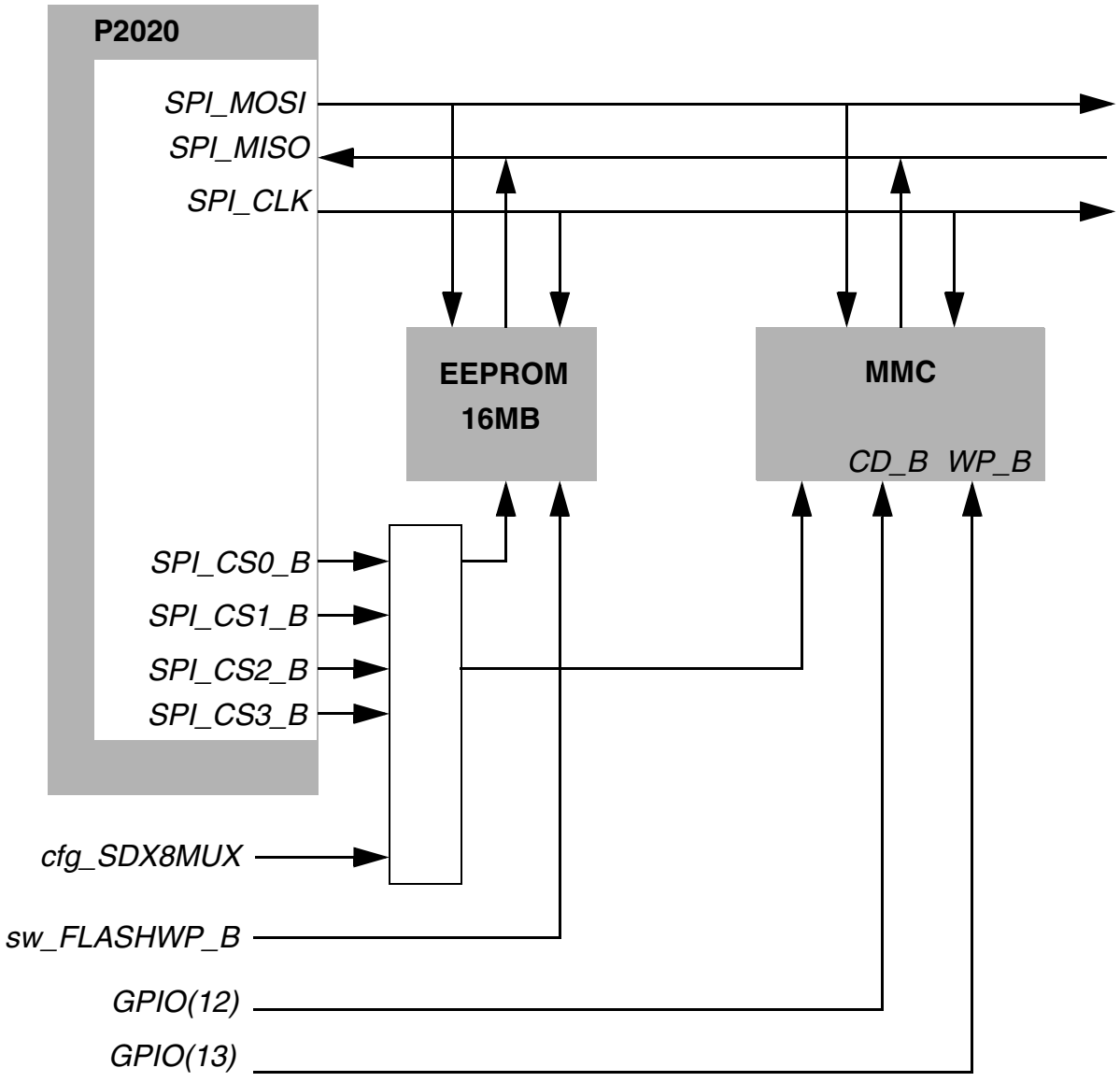

Figure 12. SPI Architecture

SPI port signals are summarized in Table 15.

Table 15. SPI Connections

Pin Count	Signal Names	Connections
1	SPI_MOSI	**P2020** Spansion S25FL128 SDMedia CardSlot #2

Table 15. SPI Connections

Pin Count	Signal Names	Connections
1	SPI_MISO	**P2020** Spansion S25FL128 SDMedia CardSlot #2
1	SPI_CLK	**P2020** Spansion S25FL128 SDMedia CardSlot #2
1	SPI_CS0_B	**P2020** Spansion S25FL128
1	SPI_CS1_B	**P2020**
1	SPI_CS2_B	**P2020** SDMedia CardSlot #2
1	SPI_CS3_B	**P2020**
7	Total pins in this group	

5.1.8 USB Interface

The **P2020** has a USB 2.0 port that supports high-speed as well as slower speeds. It uses the UTMI+ protocol to connect to an external USB PHY, and may be configured for host or device modes.

Figure 13 shows the overall connections of the USB portion.

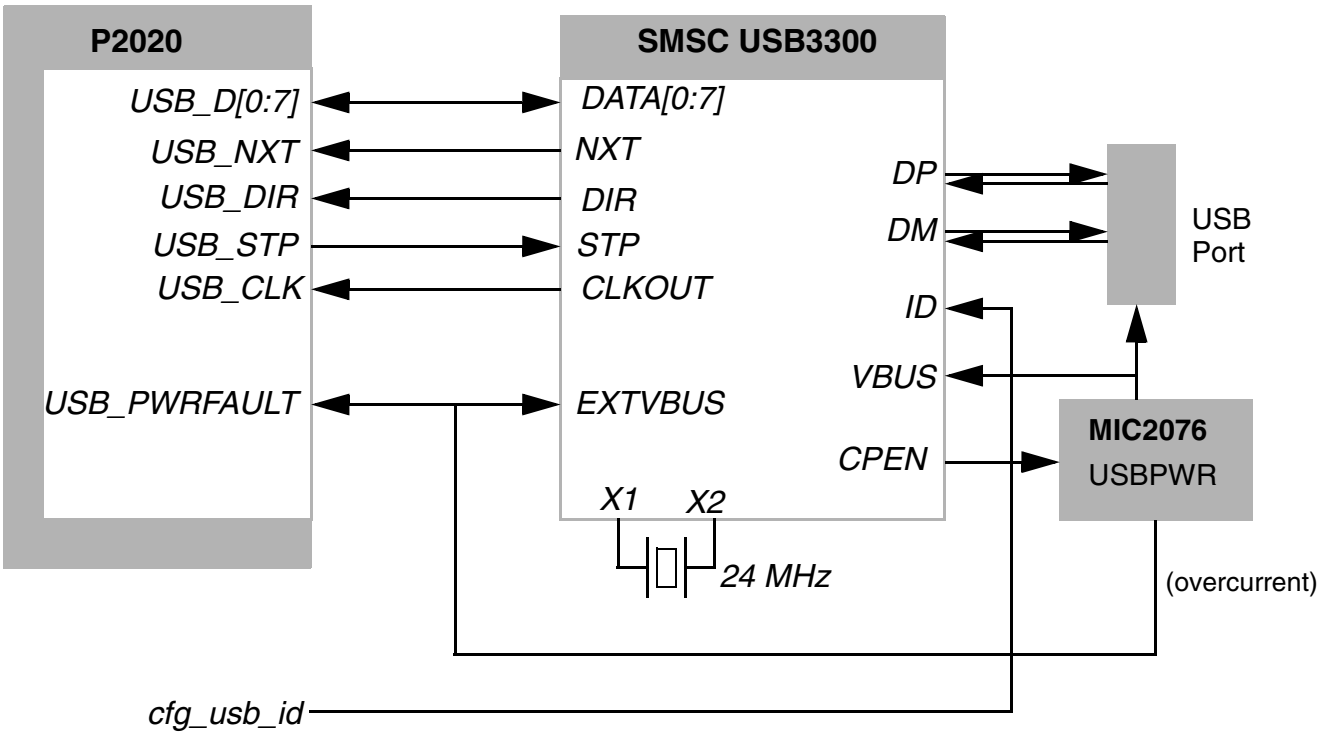

Figure 13. USB Architecture

The USB port connector is a female “Type A”, the standard connector for a host to communicate with keyboards, mice, memory sticks, etc. To support evaluating peripheral mode, the ID pin of the USB3300 can be controlled with **ngPIXIS** using the “On-The-Go” mode to switch to peripheral mode. This requires a special adapter, as a host will expect the target device to be either a male “Type A” or a female “Type B”.

USB port signals are summarized in Table 16.

Table 16. USB Connections

Pin Count	Signal Names	Connections
8	USB_D[7:0]	**P2020**, USB PHY
1	USB_NXT	**P2020**, USB PHY
1	USB_DIR	**P2020**, USB PHY
1	USB_STP	**P2020**, USB PHY
1	USB_CLK	**P2020**, USB PHY
1	USB_PWRFAULT	**P2020**, USB Power Supply
13	Total pins in this group	

5.1.9 DMA Controller

The **P2020** DMA controllers have internal and external controls to initiate and monitor DMA activity. P2020DS does not incorporate any specific devices which make use of the external pin-controlled DMA; consequently, DMA1 connects to the **ngPIXIS** where it may be controlled and monitored by software. Refer to the PX_DMACTL register in Section 9.3 for further details.

The DMA2 ports are also connected to test points to allow external hardware control as well.

Figure 14 shows an overview.

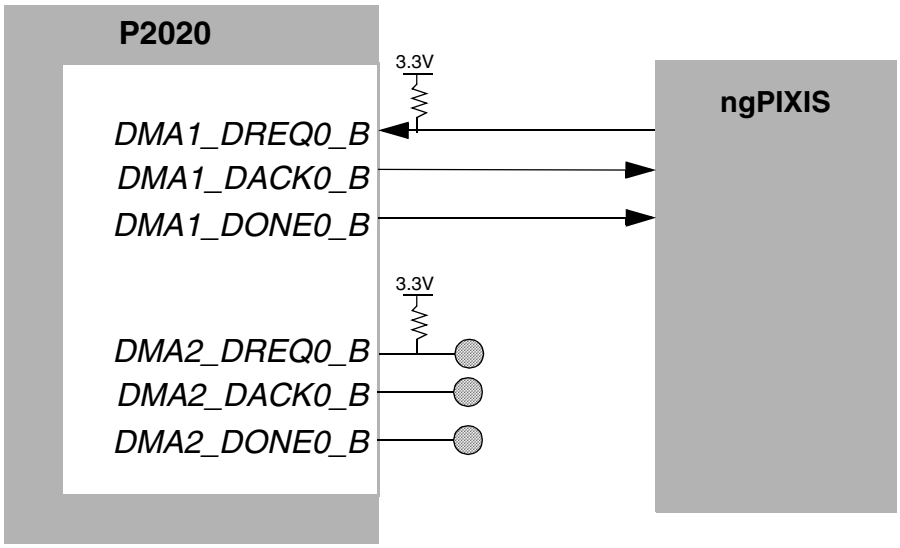

Figure 14. DMA Architecture

The DMA controller signals are summarized in Table 17.

Table 17. P2020 DMA Connections

Pin Count	Signal Names	Connections
3	DMA1_DREQ0_B DMA1_DACK0_B DMA1_DDONE0_B	**P2020** **ngPIXIS**
3	DMA2_DREQ0_B DMA2_DACK0_B DMA2_DDONE0_B	**P2020** testpoints
6	Total pins in this group	

5.1.10 eOpenPIC Interrupt Controller

P2020DS contains numerous interrupt connections; Figure 15 shows an overview.

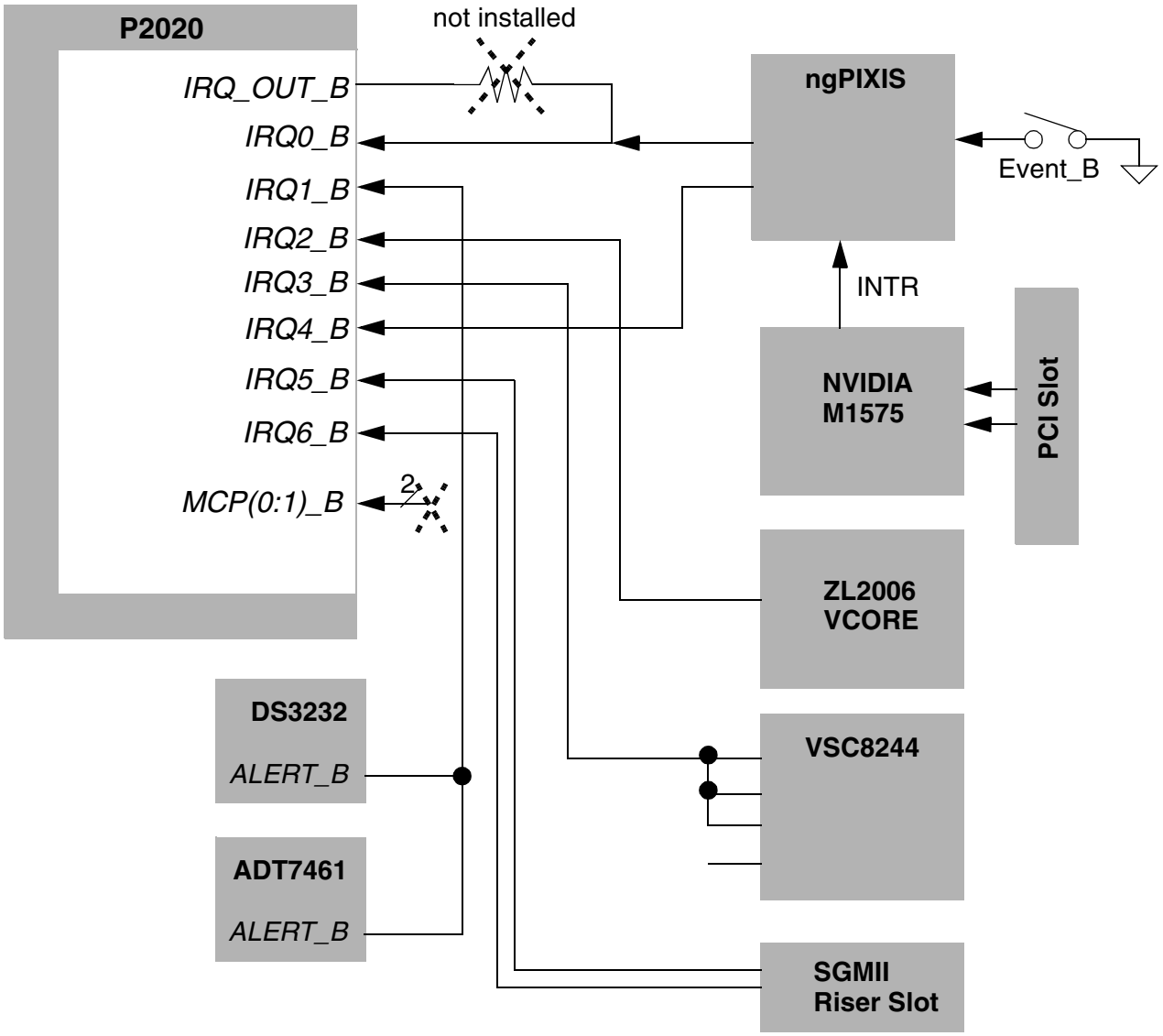

Figure 15. Interrupt Architecture

Note that the M1575 INTR output is a legacy 8259-style active-high/edge-triggered interrupt; **ngPIXIS** inverts this signal and drives it as an open-drain output onto the IRQ4_B interrupt line. IRQ4_B is essential for Linux 8259 interrupt driver support - it must align with the PEX legacy interrupts.

The **P2020** eOpenPIC connections are as shown in Table 18.

Table 18. Interrupt Connections

Pins	Signal Names	Connections
2	MCP0_B MCP1_B	pullup (unused)
2	UDE0_B UDE1_B	**ngPIXIS** driven by EVENT switch (s/w config option)
1	IRQ0_B	**ngPIXIS** Event Switch (s/w config option) optional resistor connection to IRQ_OUT_B
1	IRQ1_B	ADT7641 thermal alarm output DS3232 RTC alarm output
1	IRQ2_B	ZL2006 VCORE alert output
1	IRQ3_B	VSC8244 PHY interrupts 0-2 (wire-ORed)
1	IRQ4_B	**ngPIXIS** M1575 8259 PIC output (INTR, converted to active-low open-drain)
1	IRQ5_B	SGMII riser slot INT0_B
1	IRQ6_B	SGMII riser slot INT1_B
1	IRQ_OUT_B	optional connection to IRQ0_B
10	Total pins in this group	

5.1.11 GPIO Controller Port

Several pins of the **P2020** can be used for customer-specific applications. Some of these pins have alternate **P2020**-defined purposes to which they may also be used. All GPIO signals are connected to test points (in the form of a depopulated header) on the on the P2020DS board; for those that have additional functions, there are additional connections as noted. In general, additional functions are used so as not to interfere with use as GPIO unless otherwise noted.

GPIO signals are summarized in Table 19.

Table 19. GPIO Connections

Pin Count	Signal Names	Stingray Function	Connections
4	GPIO[0:3]		Header Optional IRQ(8:11)
4	GPIO[4:7]		Header
1	GPIO8	SDHC_CD_B	SDMedia CardDetect_B Header
1	GPIO9	SDHC_WP_B	SDMedia WriteProtect_B Header

Table 19. GPIO Connections

Pin Count	Signal Names	Stingray Function	Connections
2	GPIO[10:11]	USB_PCTL[0:1]	Header
1	GPIO[12]	MMC_CD_B	Header
1	GPIO[13]	MMC_WP_B	Header
2	GPIO[14:15]		Header
16	Total pins in this group		

5.1.12 Control Group

The **P2020** control group signals are principally related to halting or restarting exceution. Figure 16 shows an overview of the connections.

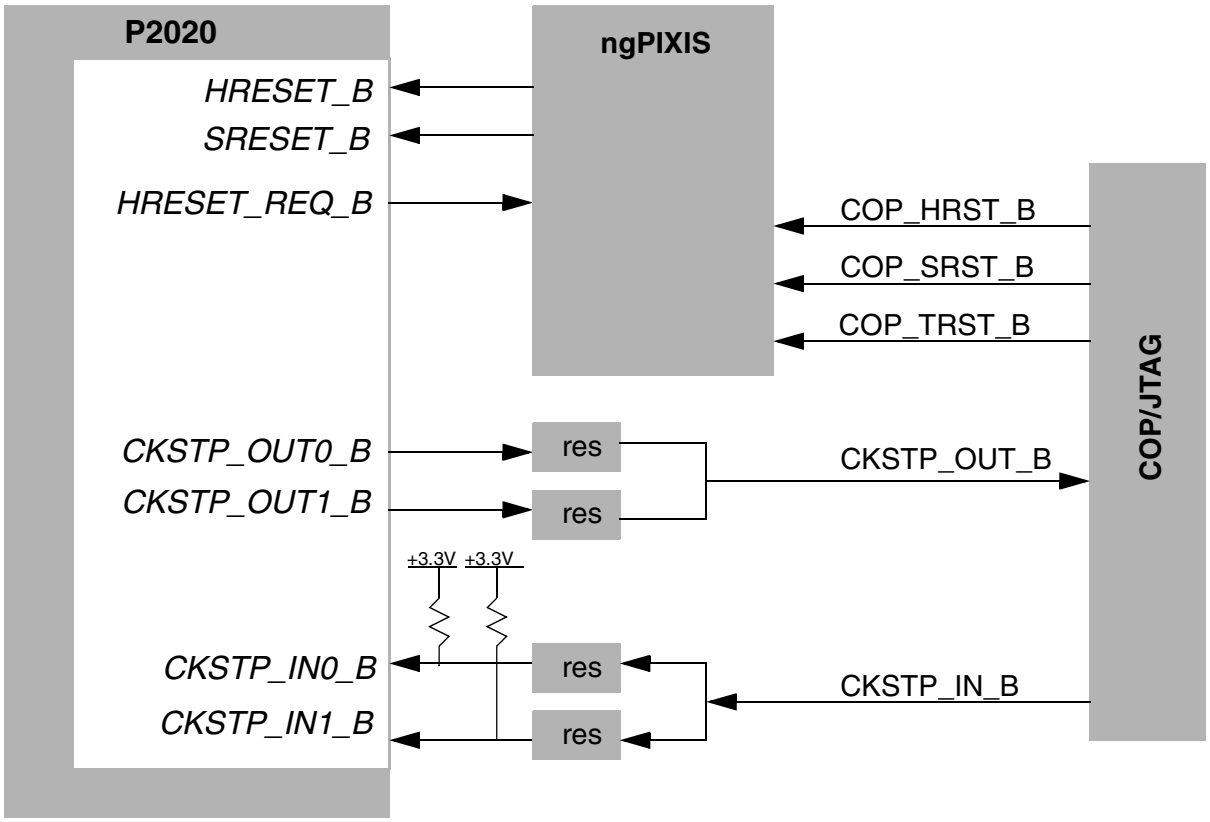

Figure 16. Control Architecture

The signal flow is fairly straightforward. The HRESET_B and SRESET_B signals are merged from the COP/JTAG header with internal controls from the reset/powerup state machines.

The “checkstop” signals from the JTAG header may be routed to CPU 0, CPU1 or both, based upon resistor stuffing options when the board is assembled. By default, both resistors are installed, and so when CHKSTP_IN_B is asserted by the COP/JTAG tool, both processors stop. The CHKSTP_OUT[0:1]_B outputs are wire-ORed to provide an active-low signal whenever either processor asserts its corresponding CHKSTP_OUT signal.

The control connections are summarized in Table 20.

Table 20. P2020 Control Connections

Pin Count	Signal Names	Connections
1	HRESET_B	
1	HRESET_REQ_B	
1	SRESET_B	To **ngPIXIS**
2	CKSTP_IN0_B CKSTP_IN1_B	COP Header
2	CKSTP_OUT0_B CKSTP_OUT1_B	COP Header
7	Total pins in this group	

5.1.13 UART Serial Ports

P2020DS connects both 4-wire serial ports to serial level transceivers, and from there to a stacked dual DB9 male connector placed in the ATX I/O gasket area. The default mode is 4-wire, so RTS/CTS flow control is supported on these connectors..

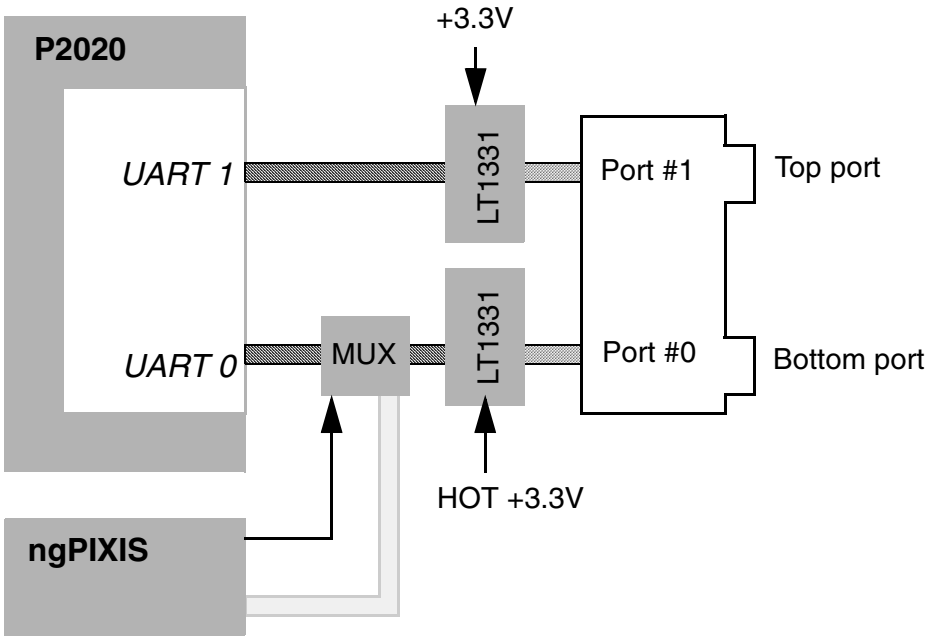

Figure 17. Serial Architecture

Serial port signals are summarized in Table 21.

Table 21. Serial Port Connections

Pin Count	Signal Names	Connections
4	UART_SOUT0 UART_SIN0 UART_CTS0_B UART_RST0_B	**P2020** LT1331 RS232 transceiver.
4	UART_SOUT0 UART_SIN0 UART_CTS0_B UART_RST0_B	**P2020** LT1331 RS232 transceiver.
8	Total pins in this group	

The UART programming model is a standard PC16550-compatible register set. Baud rate calculations for the divisor latch registers (DLL and DML) is typically done by by reading the **ngPIXIS** PX_CLK register to determine the **P2020** SYSCLK clock input (typically 133 MHz) frequency, but possibly any value. The baud rate divisors can then be calculated using the formula described in the User's Manual.

Programming Note: If the dynamic reconfiguration capabilities of **ngPIXIS** are used to set the SYSCLK input to an arbitrary value, the three-bits in the PX_CLK register are not valid. In this case, the PX_AUX register is by convention set to the value of SYSCLK, in MHz, which is used in lieu of PX_CLK.

Note that the primary serial port is powered from the 3.3V hot power rail, and thus may be used even with the system is powered down. This facility is used by the **ngPIXIS** processor to run programs and interact with the user, allowing reconfiguration of the board when sealed in the chassis.

5.1.14 I2C

The **P2020** has two separate I2C/SMB buses. Bus 1 is dedicated to a single I2C-based EEPROM which primarily contains boot initialization EEPROM. Bus 2 is normally dedicated to the SDRAM I2C-based SPD/EEPROMs (for proper memory initialization), and other devices as needed.

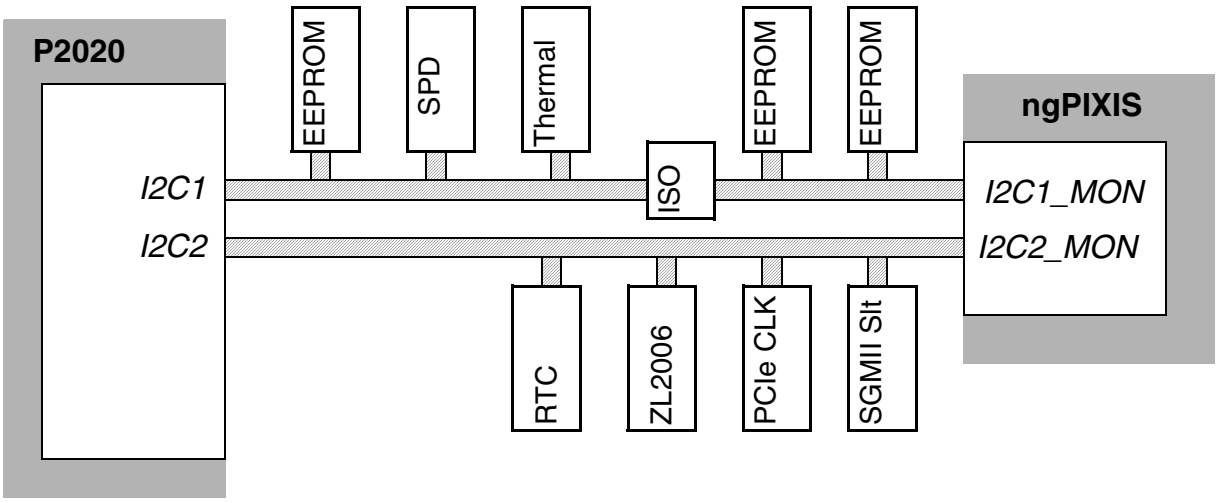

Figure 18. I2C Architecture

The I2C bus signals are summarized in Table 22.

Table 22. I2C Bus Connections

Pin Count	Signal Names	Connections
2	I2C1_SDA I2C1_SCL	**P2020** Devices: See table below
2	I2C2_SDA I2C2_SCL	**P2020** Devices: See table below
4	Total pins in this group	

I2C bus device addresses are summarized in Table 23.

Table 23. I2C Bus Device Map

I2C Bus	I2C Address	Device	Notes
1	0x0C	DIMM Thermal Monitor Microchip MCP98242 or equivalent	Presence and type of device depends on the DIMM vendor; the default Elpida device supplies an MCP98242.
1	0x31	DIMM Write Protection Microchip MCP98242 or equivalent	Presence and type of device depends on the DIMM vendor; the default Elpida device supplies an MCP98242.

Table 23. I2C Bus Device Map

I2C Bus	I2C Address	Device	Notes
1	0x4C	Processor Thermal Monitor Analog ADT7461A or equivalent	V1: present V2: deleted, replaced by ZL2006.
1	0x50	4KiB EEPROM Atmel AT24C64A or equivalent.	Processor reset initialization code (boot sequencer). May also be used for data. Write protectable.
1	0x51	DDR3 DIMM Socket Atmel AT24C02, Microchip MCP98242. or equivalent	SPD EEPROM Type of device depends on the DIMM vendor; the default Elpida device supplies an MCP98242.
1	0x55	4KiB EEPROM Atmel AT24C64A or equivalent.	Stores ngPIXIS-accessed configuration data. Accessible while board is powered off. Write protectable.
1	0x56	4KiB EEPROM Atmel AT24C64A or equivalent.	Stores ngPIXIS GMSA program code. Accessible while board is powered off. Write protectable.
1	0x57	256B SYSTEM ID EEPROM Atmel AT24C02A or equivalent.	Stores board-specific data, including MAC addresses, serial number/errata, etc. Write protectable.
1	n/a	**ngPIXIS** I2C port	Used for bus reset, monitoring, and master-only data collection.
1	n/a	I2C Access Header	For remote programming of boot sequencer startup code (if needed).
2	0x11	VCORE PMBus ZL2006	
2	0x50	SGMII Card Slot	Device address depends on attached device(s), if any.
2	0x68	Real-time clock DS3232	optional
2	0x6E	SERDES clock generator ICS9FG108	
2	n/a	**ngPIXIS** I2C port	Used for bus reset, monitoring, and master-only data collection.
2	n/a	I2C Access Header	For remote programming of boot sequencer startup code (if needed).

Note: These are “DINK”-style addresses, which do not include the position of the LSB of the transmitted address (the read/write bit).

5.1.15 Debug and Power Management

The debug and power management signals of the **P2020** are summarized in Table 24.

Table 24. Debug and Power Management Connections

Pin Count	Signal Names	Connections
1	TRIG_IN	P6880 Debug header
1	TRIG_OUT / READY / QUIESCE_B	P6880 Debug header
1	READY_P1	P6880 Debug header
5	MSRCID[0:4]	P6880 Debug header
1	MDVAL	P6880 Debug header
1	CLK_OUT	Test point w/adjacent ground.
1	ASLEEP	**ngPIXIS** P6880 Debug header
11	Total pins in this group	

With the exception of the TRIG_OUT/READY/QUIESCE_B and ASLEEP pins, which are also connected to the **ngPIXIS** system supervisor for power control and monitoring, the remaining signals are simply connected to a Tektronix P6880 debug connector (which is a PCB pattern, and does not require any components).

5.1.16 Clock

The clocks for the **P2020** are summarized in Table 25. Further details on the clock architecture are covered in Section 5.5, “Clocks,” on page 52.

Table 25. P2020 Clock Connections

Pin Count	Signal Names	Connections
1	SYSCLK	ICS307 System clock synthesizer
1	DDRCLK	ICS307 DDR clock synthesizer
1	RTC	Arbitrary timebase frequency
11	Total pins in this group	

Note: the SERDES and ethernet clocks are included in their respective sections.

5.1.17 JTAG/Test

The **P2020** JTAG(COP) and test signals are summarized in Table 26.

Table 26. P2020 JTAG(COP) and Test Connections

Pin Count	Signal Names	Connections
1	TCK	COP Header
1	TDI	COP Header
1	TDO	COP Header
1	TMS	COP Header
1	TRST_B	**ngPIXIS**
1	TEST_MODE_B	Pullup
1	TEST_SEL_B	Pullup
2	FA_ANALOG_1 FA_ANALOG_2	Test
9	Total pins in this group	

5.1.18 Temperature

The **P2020** has two pins connected to a thermal body diode on the die, allowing direct temperature measurement. These pins are connected to either the Analog Devices ADT7461 thermal monitor (for Stingray V1) or to the Zilker ZL2006 (for V2 and later), which allows direct reading of the temperature of the die and is accurate to ±1 °C.

Thermal management signals are summarized in Table 27.

Table 27. Thermal Management Connections

Pin Count	Signal Names	Connections
2	TEMP_ANODE TEMP_CATHODE	**P2020** V1: ADT7461 V2+: ZL2006
2	Total pins in this group	

5.1.19 Power

The power requirements of the **P2020** are estimated at this time; based on historical precedents, estimated power requirements are summarized in Table 28.

Table 28. P2020 Power Supply Details

Power Pins	Description	V_{NOM}	Voltage Range	I_{MAX} @ V_{NOM}	Supplier
OVDD	General I/O Supply	3.3V	3.3V	< 2A	ATX PSU
LVDD	TSEC[1:3] Supply	2.5V	2.5V	< 2A	TPS75225
GVDD	DDR SSTL-1.5 Supply	1.5V	1.5V-1.8V	< 4A	TPS51116
BVDD	LocalBus/GPIO Supply	3.3V	3.3V	< 2A	ATX PSU
SVDD	SERDES Core Supply	1.0V	1.00 / 1.05V	< 0.6A	TPS54910
XVDD	SERDES I/O Supply			< 0.3A	
VDD	VCORE0, VCORE1, Platform	1.0V	0.90-1.2V	< 6A	ZL2006
AVDD_x	PLL filter for CORE0, CORE1, DDR, PLAT, LBIU and SERDES			0.1 A	

Note that this is the power for the **P2020** only, it does not include external devices, memory, etc. Since these are estimates, and because alpha silicon tends to be ‘hot’, the VDD rail needs to have excess capacity of approximately 20%.

Note also that these voltage levels are not all that are supported by **P2020**, they are the voltage levels supported by P2020DS.

Because of the high current transients present on the VDD power pins, careful attention should be paid to properly bypass these power pins, and to provide a good connection between the BGA pads and the power and ground planes. In particular, the SMD capacitors should have pads directly attached to the via ring (or within it, if the PCB costs are not prohibitive).

Power supply connections are summarized in Table 29.

Table 29. Power Supply Connections

Pin Count	Signal Names	Connections
31	VDD	**P2020**, ZL2006 PSU
1	AVDD_CORE0	
1	AVDD_CORE1	
1	AVDD_PLAT	
1	AVDD_DDR	
1	AVDD_LBIU	
1	VDD_SENSE	
1	POVDD	
1	FA_VDD	
6	XVDD	**P2020**, TPS 54910
6	SVDD	
1	AVDD_SRDS	
1	RSVD_SVDD	
1	RSVD_AVDD2_SRDS	
35	GVDD (VDD_DDR)	**P2020**, TPS51116 PSU
9	LVDD	**P2020**, TPS72525 PSU
6	OVDD	**P2020**, ATX PSU
7	BVDD	
3	CVDD	
150	GND	**P2020**, common ground plane
8	XGND	
11	SGND	
1	AGND_SRDS	
1	RSVD_SGND	
1	RSVD_AGND2_SRDS	
1	VSS_SENSE	
280	Total pins in this group	

5.1.20 Mechanical Clearance

The **P2020** is a 31x31mm 689-pin 1mm pitch TEPBGA II (temperature-enhanced plastic BGA) package (a incompletely filled array). In addition to providing a socketable board, additional considerations are required to accomodate the heatsink both for socketed systems and non-socketed systems. Refer to Section 8, "PCB Development Issues," on page 74 for more details.

The expected thermal dissipation requirements are 7.26W max. A passive heatsink rated for 8W or greater will be required; a cooling fan is not desired but connections are provided for one in order to accomodate more "off the shelf" solutions.

5.2 South Bridge

P2020DS uses the NVidia M1575 "Super South Bridge" to provide access to standard Linux I/O devices, including:

- SATA 2 ("serial IDE")
- Real-time clock
- BBRAM
- AC97 Audio

Figure 19 shows an overview of the NVidia M1575.

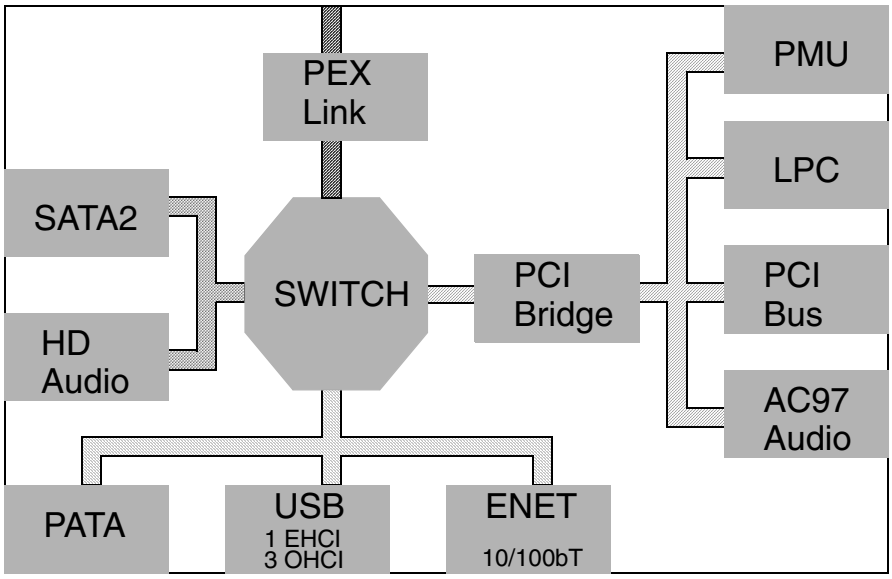

Figure 19. NVidia M1575 Overview

The M1575 is operated in "end-point" mode, as compared to "south-bridge" mode. The NVidia M1575 supplies all the IO machinery needed for full Linux, QNX or other OS desktop support.

The M1575 is in a 628-Ball (31mmx31mm) BGA package, and requires several clock and power sources as detailed in Section 5.4, "System Power"" and Section 5.5, "Clocks"".

5.2.1 NVidia SATA Controller

The NVidia M1575 supports a high-speed serial ATA (“SATA”) connections. The SATA controller supports four ports at a 1.5 Gib/s and 3.0 Gib/s data rates, for SATA I and SATA II modes, respectively. AHCI features are also supported. Figure 20 shows the overall connections of the SATA bus.

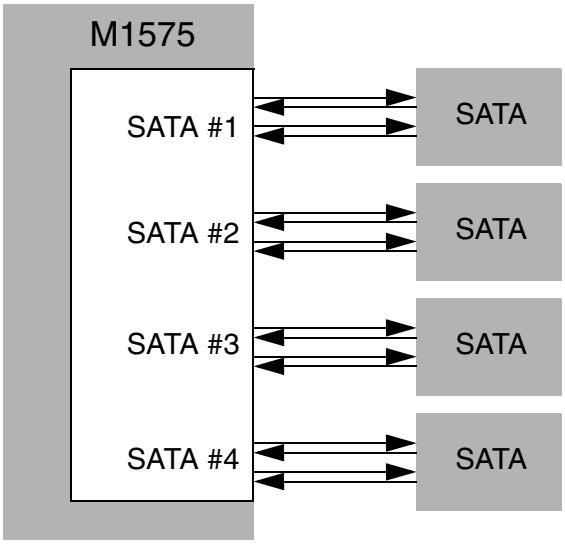

Figure 20. SATA Architecture

5.2.2 NVidia PCI Controller

The NVidia M1575 provides a conventional 33MHz, 5V PCI interface for communication with legacy PCI boards, and most importantly (for test purposes) provides a channel for remote control of the P2020DS using the “DataBlizzard” PCI bridge card. The “DataBlizzard” can control many features of the board remotely via PCI configuration cycles.

The PCI bus is connected only to one PCI slot, reflecting both the high integration of Freescale Power Architecture™ devices and the transition to PCIExpress and other channels.

The NVidia M1575 provides all clocks and arbitration signals for the slow. Figure 21 shows the bus organization.

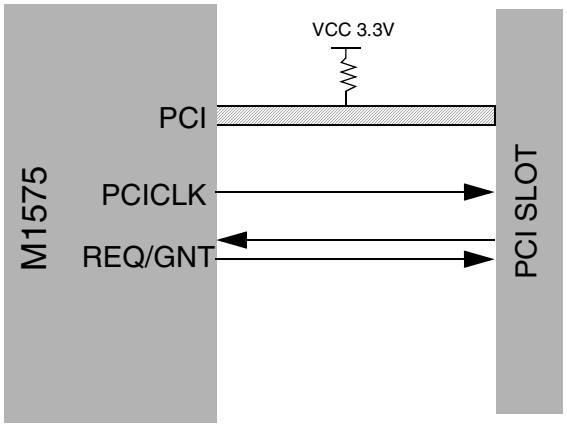

Figure 21. PCI Bus Architecture

Note that the M1575 drives signals to 3.3V levels, and bus pullups are 3.3V; it is tolerant to 5V signalling levels. The PCI slots are configured to 5V and the PCI IO pins are connected to 5V, reflecting the large majority of cards which only support that option.

Table 30 summarizes the PCI bus arbitration and interrupt connections.

Table 30. P2020DS PCI Bus Information

Device	Vendor Device	IDSEL	Arbiter Port	Clock Port	PCI Interrupt Mapping				Notes
					INTA#	INTB#	INTC #	INTD #	
M1575	0x10B9 0x5249	AD16							Device is the PCIBridge, there are other devices (see spec).
Slot 1	varies	AD17	0	0	0	1	2	3	Only slot

5.2.3 NVidia Interrupts

The NVidia M1575, which configured as an end-point, routes internal interrupt signals to pins. This pins are collected in the **ngPIXIS**, which maps them to specific **P2020** interrupt inputs.

5.2.4 NVidia Audio

The NVidia AC97 audio controller logic is connected to an AC97 codec, and then to a standard combined AC97 audio line in/mic in/line out mini jack. Figure 22 shows the overall connections of the audio portion.

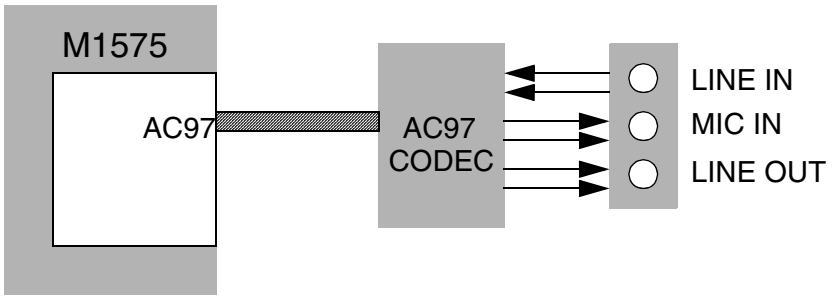

Figure 22. Audio Architecture

5.2.5 NVidia Power/Power Control

Other than standby real-time clock/NVRAM battery power, all NVidia power supplies are supplied by the ATX power supply or other sources derived from it. VCC_HOT_1.8V is constantly provided to power the APM/ACPI section.

5.2.6 NVidia Other

The NVidia M1575 has several useful features which are supported. These include:

- RTC

- NVRAM - 256 bytes

Note: The M1575 locks the RTC until a memory fetch is processed by internal device 0x29 (system/legacy interface). This can be handled by one of the following methods:

- Do a read access from the LPCFlash space.
- Do a read from the PCI memory space.
- Setup a memory space at an unused location.

5.2.7 NVidia Unsupported Interfaces

The ethernet, parallel IDE (PATA), USB controller, floppy disk controller, and other interfaces are not supported.

5.3 System Control Logic

P2020DS contains an FPGA, the "**ngPIXIS**", which implements the following functions:

- Reset sequencing/timing combined with COP/JTAG connections.
- Map/re-map **P2020** local bus chip selects to flash, compact flash, etc.
- Transfer switch settings to processor/board configuration signals.
- Load configuration data from RAM (registers) or EEPROM to override configuration for self-test.
- Miscellaneous system logic
  - COP reset merging
  - DMA trigger/monitor regs.
  - I2C timeout reset.

The FPGA is powered from standby power supplies and an independant clock. This allows the FPGA to control all aspects of board bringup, including power, clocking and reset.

The **ngPIXIS** is implemented in an Actel A3P600 in a 484-256-pad micro-BGA. Figure 23 shows the overall **ngPIXIS** architecture.

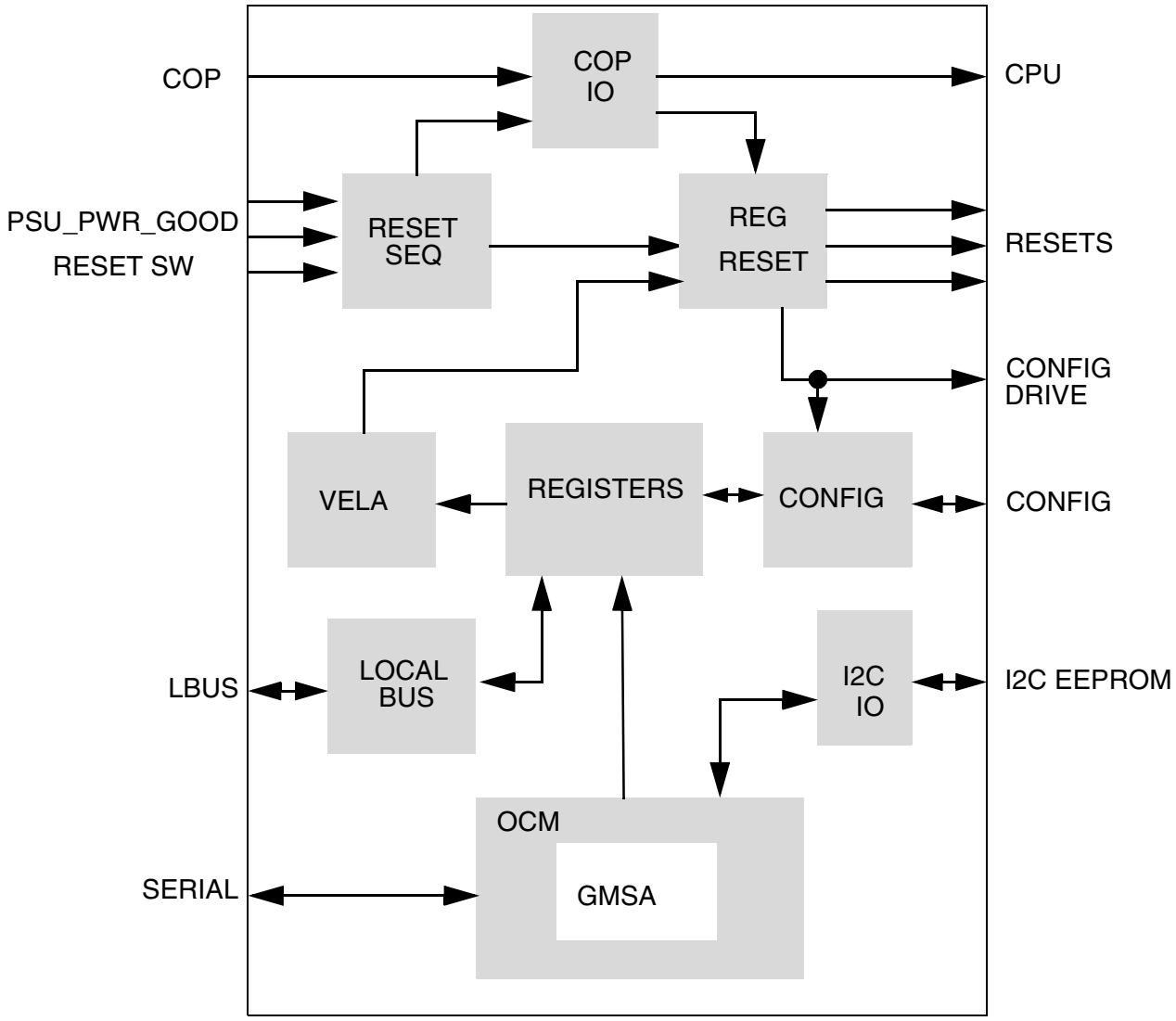

Figure 23. ngPIXIS Overview

The principal portions of **ngPIXIS** are:

COP	Handles merging COP header resets with on-board resets in a transparent manner.
RESETSEQ	Collects various reset/power-good signals and starts the global reset sequencer.
REGRESETS	Drives resets from the sequencer, from register-based software control, or from VELA.
REGISTERS	A multi-ported register file containing status and configuration data.
LOCALBUS	Interface between processor and REGFILE
CONFIG	Monitors and/or sets selected configuration signals
VELA	VELA is a simple machine to monitor requested changes in board configuration and when detected, perform a power-on-reset / re-configuration of the target system.

OCM	Offline Configuration Manager - a machine which initializes the **ngPIXIS** registers, including those used for **P2020** initialization, from external I2C EEPROMs. The OCM can talk to the user or another computer using the serial port while the system is powered down.
GMSA	General Microprocessor/Stack Architecture - a stack-based microprocessor which loads executes before and during power-down/-up events. It can query I2C devices and collect data during normal system operation, as well as allow setting configuration switches without opening the chassis.

5.3.1 Subsections

5.3.1.1 COP

Handles merging COP header resets with on-board resets in a transparent manner. It is critical that the COP HRST* input resets the entire system **EXCEPT** for the COP JTAG controller (i.e. TRST* must not be asserted). With COP not attached, it is critical that reset **does** assert TRST*. The COP core manages these modal operation.

5.3.1.2 RESETSEQ

Collects various reset/power-good signals and starts the global reset sequencer.

Note that ASLEEP indicates the processors(s) have exited the reset state. It does not cause a reset, as the processor can sleep for any number of reasons after hard reset has completed.

Note also that during power-down ALL I/O and output drivers must be tristated. After power up, drivers MAY be driven. Normal operation and/or use of the VELA engine may cause some I/Os to be tristated.

5.3.1.3 REGRESETS

Copies reset signals from the sequencer, but also allows register-based software to individually asserted reset tot the local bus, memory, and/or compact flash interfaces.

5.3.1.4 REGFILE

A dual-ported register file containing several sorts of registers.

Note that REGFILE must be able to accept (or arbitrate for) concurrent writes to the same register, though this is not a statistically likely occurrance.

5.3.1.5 LOCALBUS

Interface between processor and REGFILE. Since access to the internal registers may be blocked, asynchronous (not ready) signalling is used.

5.3.1.6 CONFIG

Monitors and/or sets selected configuration signals.

In some instances, CONFIG maps switch settings into direct configuration outputs, while in others (such as SYSCLK) it maps a 3-position switch into a 16-bit register initialization pattern, which is subsequently used to initialize the clock generator.

5.3.1.7 VELA

VELA is a simple microsequencer used to monitor sequence in requested changes in board configuration upon a signal (generally a register write from PCI). When detected, bits in a PX_EN[1:8] register allow a corresponding PX_SW[1:8] register to be driven onto configuration pins during a system restart.

5.3.1.8 OCM

The OCM, or off-line configuration manager, is a small microprocessor (GMSA) which contains an embedded CPU core, 8K SRAM and I/O peripherals (UART, I2C, GPIO and timers). The primary goal of the OCM and processor is:

All users and third-parties to customize the configuration and data collection as needed, without requiring a custom FPGA to be designed.

Previous generation FPGA (PIXIS) supported many OCM features, but if any changes were needed, it not only required a full FPGA design tool flow, but the time needed to analyze and design the results. In addition, replacing the FPGA image requires a special hardware module and host PC. By replacing much of the PIXIS logic with a general purpose processor and support logic, end-users can customize many features by changing the program stored in a easily accessible (but protected) I2C-based EEPROM. An assembler and simulator toolset is available.

The standard OCM software performs the following functions:

- Monitor PX_VCTL[GO] to avoid interfering with self-shmoo
- Load **ngPIXIS** SW/EN registers from external I2C EEPROM
- Modify **ngPIXIS** misc registers from external I2C EEPROM
- Modify VCORE output voltage based on SW_VCORE(0:1) settings
- User interaction to allow programming I2C EEPROM (even with power off).
- Background data collection on VCORE, ICORE, TEMP, etc.
- Other system control functions (reset, power cycle, etc.

A block diagram of the OCM component is shown in Figure 24.

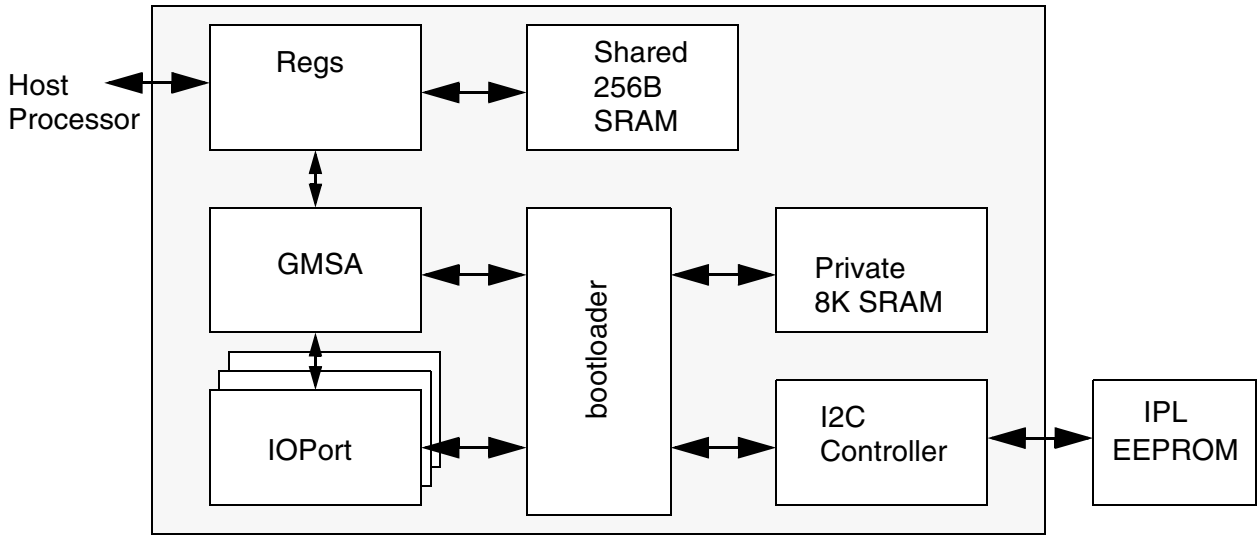

Figure 24. OCM GMSA Implementation

A great portion of the OCM is defined by the software, as this can be changed by the end-user or updated with new functions at any time, refer to the OCM documentation for definitive details.

5.3.1.9 OCM Configuration Functions

One of the two primary functions of the OCM is to configure the target

5.3.1.10 OCM Interactive Functions

5.3.1.11 OCM Message Protocol

When the OCM is running, it monitors the PX_OCMCSR register to see if the MSG bit is set. If so, it will use the data in the PX_OCMMSG register, and data in the shared 256-byte SRAM (accessed via PX_ADDR/PX_DATA) to determine the next action to perform.

Table 31 summarizes the MSG/ACK sequence.

Table 31. OCM Message Protocol Handling

ACK	MSG	Description	System Action	OCM Action	Notes
0	0	Idle	May set PX_OCMMSG and SRAM	None	System may NOT send message unless ACK is clear.
0	1	New Message	Wait for ACK (if desired)	On detect, examine PX_OCMMSG and SRAM. When done, set ACK.	System is very much faster than OCM, so no need to wait for ACK unless you want to.
1	1	Completed	May clear MSG.	Wait for MSG=0	
1	0	Idle	None	Clear ACK.	

The OCM treats the value in PX_OCMMSG as a pointer to a list of commands in the shared SRAM. Each element is examined in turn, and the command is processed until an end sequence is found. Commands may have optional arguments, etc. Note that user interaction is suspended during command processing. Unless you are in “Interactive” mode, this will not affect anything.

Table 32 summarizes the message codes.

Table 32. OCM Message Protocol Codes

Message	Code	Operands?	Definition
END	0x00	-	Terminate program.
DLY	0x01	n (1 byte)	Delay program n seconds.
RST0	0x02	-	Assert HRESET to target.
RST1	0x03	-	De-assert HRESET to target.
PWR	0x05	-	Toggle power to target.
GETMEM	0x08	a (two bytes)	Set PX_OCMMSG to value of program memory at specified address (MSB first).

Table 32. OCM Message Protocol Codes

Message	Code	Operands?	Definition
SETMEM	0x09	a (two bytes)	Store PX_OCMMSG value to program memory at specified address (MSB first).
SCLR	0x10	-	Clear all accumulated snapshot data.
START	0x11	-	Begin background snapshot data collection using programmed timer rate (default = 0.5s)
STOP	0x12	-	Stop background snapshot data collection.
GET	0x13	a (one byte)	Store snapshot data to SRAM at supplied address.
ENABLE	0x14	a (one byte)	Select data sources to collect. (default: all sources) b1: temp b0: core 0: V + I NOTE: These bits will vary by platform.
TSLOW	0x20	-	Set hardware time clock to slow clock (240 Hz). Default.
TFAST	0x21	-	Set hardware time clock to slow clock (38 kHz).
TIMER	0x22	v (two bytes)	Set timer rate to supplied value. Default: 0x10 Timer speed depends on timer rate selected and prescaler (default is slow, and 0x56).

Table 32 shows some timer encoding values (note that the timer hardware prescale value (24) is not changable by user program):

Table 33. OCM Timer Values

Timer Rate	Hardware Prescale	TIMER	Definition
SLOW	24	10	1 Hz: 240 / 24 / 10
		1	10 Hz : 240 / 24
		100	0.1 Hz : 240 / 24 / 100
FAST	24	1583	1 Hz: 38000 / 24 / 1583
		1	1583 Hz : 38000 / 24
		16	~98 Hz : 38000 / 24 / 16

A sample representation of an OCM message is shown in Figure 25.

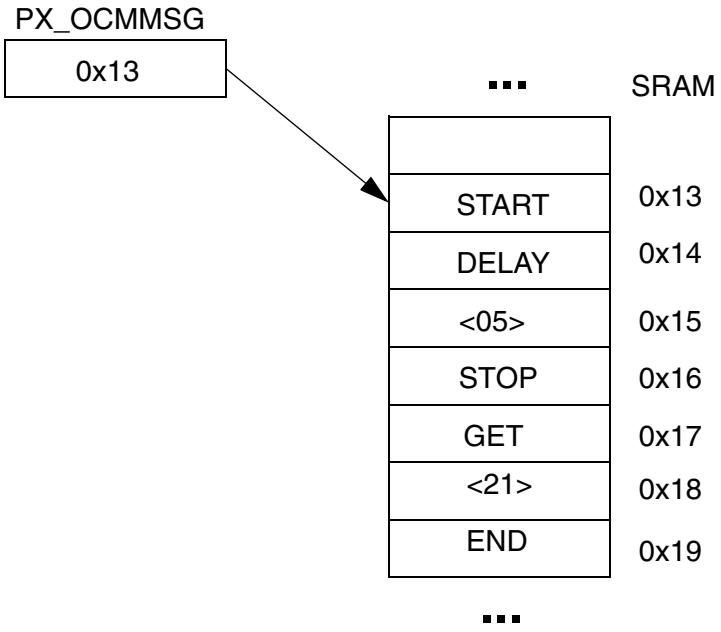

Figure 25. OCM Sample Message

In this example, the **P2020** preloaded the contents of the SRAM with the above program, starting at address 0x13. The PX_ADDR and PX_DATA registers were used to accomplish this.

This sequence instructs the OCM to:

- initiate PMBus data collection
- delay 5 seconds
- stop data collection
- stores collection result in SRAM at location 0x21
- stop program execution

Then, the address 0x13 is stored in the PX_OCMMSG register. At this point, all data has been setup; all that is left is start the program. When ready, the host system signals the OCM by setting PX_OCMCSR. When the program has completed, the OCM sets the ACK bit and stops further activity (except data collection, unless specifically commanded to stop).

Note that the host system can setup multiple programs in the SRAM (start collection, stop collection, get data) and trigger them at will just by setting the OCMMSG/OCMCSR registers alone.

5.3.2 Power

Power for **ngPIXIS** is supplied from dedicated VCC_HOT_3.3 and VCC_HOT_1.5V power supplies based upon the ATX power supply +5V standby power, VCC5STDBY.

5.3.3 Register Summary

ngPIXIS contains several registers as detailed in Table 34; for further details, see Section 9.3, "ngPIXIS Registers," on page 81".

Table 34. ngPIXIS Register Map

BASE ADDRESS OFFSET	REGISTER	NAME	ACCESS	RESET
0x00	System ID register	PX_ID	R	0x16
0x01	System architecture version register	PX_ARCH	R	0x01
0x02	**ngPIXIS** version register	PX_SCVER	R	0x02
0x03	General control/status register	PX_CSR	R/W	0x00
0x04	Reset control register	PX_RST	R/W	0x00
0x05	*reserved*	*reserved*	*reserved*	*undefined*
0x06	Auxiliary register	PX_AUX	R/W	0x00
0x07	Speed register	PX_SPD	R	0x00
0x08	Board Configuration register 0	PX_BRDCFG0	R/W	0xF7
0x09	DMA Control/Status register	PX_DMA	R/W	0x00
0x0A	SRAM Address Register	PX_ADDR	R/W	0x00
0x0B-0x0C	*reserved*	*reserved*	*reserved*	*undefined*
0x0D	SRAM Data Register	PX_DATA	R/W	*undefined*
0x0E	LED Data Register	PX_LED	R/W	0x00
0x0F	*reserved*	*reserved*	*reserved*	*undefined*
0x10	VELA Control Register	PX_VCTL	R/W	0x00
0x11	VELA Status Register	PX_VSTAT	R	0x00
0x12	VELA Configuration Enable Register 0	PX_VCFGEN0	R/W	0x00
0x13	*reserved*	*reserved*	*reserved*	*undefined*
0x14	OCMCSR Register	PX_OCMCSR	R/W	0x00
0x15	OCMMSG Register	PX_OCMMSG	R/W	0x00
0x16	GMDBG Register	PX_GMDBG	R/W	0x00
0x17-0x18	*reserved*	*reserved*	*reserved*	*undefined*
0x19	SCLK0 Register	PX_SCLK0	R/W	*varies*
0x1A	SCLK1 Register	PX_SCLK1	R/W	*varies*
0x1B	SCLK2 Register	PX_SCLK2	R/W	*varies*
0x1C	DCLK0 Register	PX_DCLK0	R/W	*varies*
0x1D	DCLK1 Register	PX_DCLK1	R/W	*varies*
0x1E	DCLK2 Register	PX_DCLK2	R/W	*varies*
0x1F	WATCH Register	PX_WATCH	R/W	0x7F
0x20	SW1 Registers	PX_SW1	R/W	*varies*
0x21	EN1 Registers	PX_EN1	R/W	*varies*
0x22-0x2D	SW/EN (2:6) Registers	*varies*	*varies*	*varies*
0x2E	SW8 Registers	PX_SW8	R/W	*varies*

Table 34. ngPIXIS Register Map

BASE ADDRESS OFFSET	REGISTER	NAME	ACCESS	RESET
0x2F	EN8 Registers	PX_EN8	R/W	*varies*
0x30-0x3F	*reserved*	*reserved*	*reserved*	*undefined*

5.4 System Power

The 12V, 5V and 3.3V power requirements are met by the attached ATX-12V compatible power supply unit (PSU). 5V and 3.3V are connected to individual power planes in the P2020DS PCB stackup. The 12V power from the standard ATX header treated as separate from the ATX-12V power, which supplies a large amount of current and is referred to as “VCC_12V_BULK”. The latter is used solely for the VCORE power supply rail, while the former is used for miscellaneous purposes such as fan power and PCI slots.

Note that to support **ngPIXIS** standby operation and to support video cards or other high-power-dissipation cards in the PCIExpress slot, the PSU should support the following minimum specification:

- minimum 450W overall, 500W recommended
- supports one PCIE 12V connector
- PCIE 12V support a minimum of 150W
- mimimum 5V 2A standby current

All other power sources are derived from the ATX PSU. Figure 26 shows the principal clock connections (DDR and miscellaneous clocks are not shown).

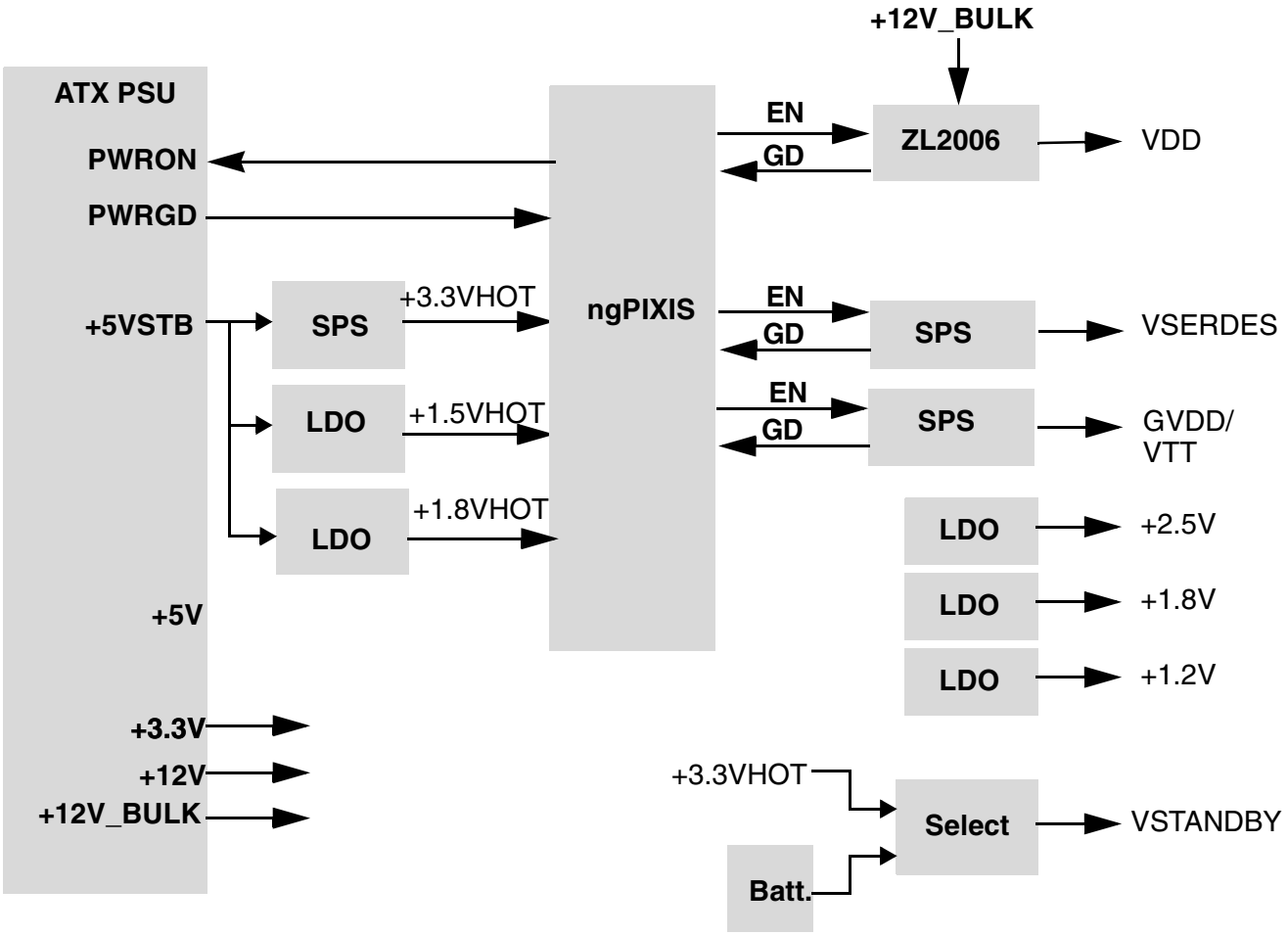

Figure 26. Stingray Power Architecture

5.4.1 Core Power

P2020DS uses the Zilker Labs ZL2006 multi-phase switching power controller. P2020DS uses this device as a single-phase controller for up to 10A of power at a nominal 1.10V output. Of particular interest is the PMBus capabilities of the ZL2006; P2020DS uses hardware configuration pins to set the nominal voltage to 1.10V, but using PMBus commands, nearly any parameter of the design can be adjusted by software, including:

- output voltage
- current limit
- slew rate
- power-up delay
- droop compensation
- margining

and in addition, the “SNAPSHOT” command allows collection of data, including paired voltage and current measurements. Refer to ZL2006 datasheet and the PMBus association specifications for further details.

To allow a limited amount of test support without requiring software, the **ngPIXIS** converts two switch settings into three commands that are issued during the startup reset sequence (if the switch is “00”, no such action is taken, and the hardware-selected default applies).

Table 35. ZL2006 Settings

cfg_vcore	Action Taken	VCORE Voltage
00	None	1.050 V
01	Issue PMBus command to set voltage	1.000 V
10		0.950 V
11		0.925 V

Note that the system has full control of the VCORE setting using the PMBus.

5.4.2 VSERDES

P2020DS uses the TI TPS54310 switching power supply to provide up to 3A at a nominal 1.05V. A single output controlled by the **ngPIXIS** allows selecting an alternate voltage position; or further details, see Table 35”.

Table 36. VSERDES Selection

cfg_vserdes	VSERDES Voltage
0	1.05V (default)
1	1.00V

VSERDES supplies both the XVDD (eXternal, or I/O, voltage) and SVDD (internal) SERDES power supplies. The power planes are electrically isolated through a ferrite bead.

5.4.3 GVDD/VTT

Power for the DDR interface is derived from the TPS51116 switching power supply. This device supplies both GVDD (DDR IO power), VTT (termination power), and MVREF (switching reference voltage) for the DDR DIMM and the **P2020** DDR interface.

5.5 Clocks

Table 37 summarizes the clock requirements of P2020DS. Note that the DDR clocks are not included, as they are provided by the **P2020**.

Table 37. P2020DS Clock Requirements

Clock	Destination	Clock Frequency	Specs	Type	Notes
SYSCLK	**P2020** SYSCLK	33-200 MHz	t_R <= 1ns t_F <= 1ns <= 60% duty <= 150 ps jitter	LVTTL	133.33 nominal closed loop jitter bandwidth should be <500 kHz at -20 dB.
DDRCLK	**P2020** DDRCLK	33-200 MHz	t_R <= 1ns t_F <= 1ns <= 60% duty <= 150 ps jitter	LVTTL	133.33 nominal closed loop jitter bandwidth should be <500 kHz at -20 dB.
BCLK	M1575 CLK14M	14.318 MHz	none	LVTTL	Traditional ISA clock reference.
	P2020 RTCCLK				
	AC97 Codec AUDCLK				
REFCLK	**P2020** SD1_REFCLK(p,n)	100.00 MHz or 125.00 MHz	jitter: 80-100 ps skew: 330 ps	LVDS	100.00 MHz 100 ps jitter
	PEXSLOT1 REFCLK(p,n)				
	PEXSLOT2 REFCLK(p,n)				
	MIDBUS TAP1 REFCLK				
	NVidia M1575 REFCLK(p,n)				
GTXCLK	**P2020** EC_GTX_CLK125 VSC8244 XTAL1				
UPHYCLK	USB PHY clock	26.000 MHz		LVTTL	
SATACLK	M1575 X25M(1:2)	25.000 MHz		LVTTL	
PCICLK	M1575 PCI1_CLK0	33.333 MHz	>47-53% duty	LVTTL	
	PCI Slot PCI1_CLK1				
CLKCLK	M1575 X32KI	32.768 kHz		analog	

Figure 27 shows the principal clock connections (DDR and miscellaneous clocks are not shown).

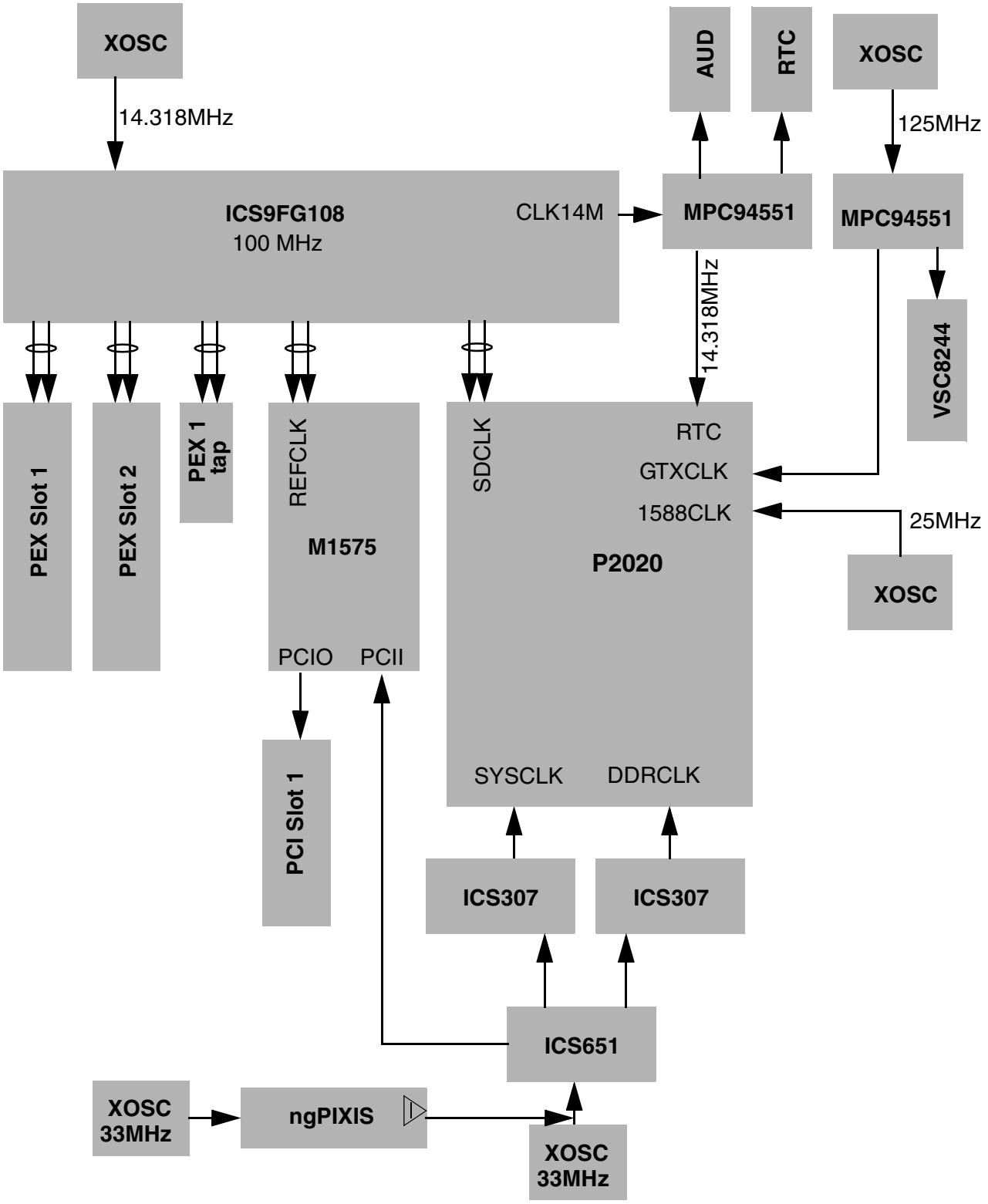

Figure 27. P2020DS Clock Architecture

5.5.1 SYSCLK

Much of the timing within the **P2020** is derived from the SYSCLK input. On P2020DS this pin is controlled by an IDT ICS307-02 frequency synthesizer. This device is serially configured by twenty-found bits of data by the **ngPIXIS** as part of the reset/power-up sequence. These 24 bits can be controlled to set the SYSCLK speed to fine increments using the dynamic (re)configuration facilities of remote access **ngPIXIS**. To make configuration easy, **ngPIXIS** pre-loads the 24-bit configuration pattern using one of eight popular values by sampling three switches located on the motherboard.

Table 38 summarizes the switch-selectable clock generation possibilities.

Table 38. SYSCLK Frequency Options

cfg_sysclk	Selected SYSCLK	Actual SYSCLK	Error	ICS307 Control Word	Notes
0 0 0	33.333 MHz	33.3330 MHz	0 ppm	0x200381	
0 0 1	40.000 MHz	39.9996 MHz	10 ppm	0x200501	
0 1 0	50.000 MHz	49.9995 MHz	10 ppm	0x220501	
0 1 1	66.666 MHz	66.6660 MHz	0 ppm	0x270501	
1 0 0	83.333 MHz	83.3325 MHz	6 ppm	0x230381	
1 0 1	100.000 MHz	99.9990 MHz	10 ppm	0x230501	
1 1 0	133.333 MHz	133.332 MHz	7.5 ppm	0x210201	Default
1 1 1	166.666 MHz	166.6650 MHz	6 ppm	0x210381	

Table 38 is based upon a 33.333 MHz reference clock input. The “Control Word” field is the data sent to the ICS307 upon startup, or when commanded to by the VELA controller. This value can be calculated from the ICS307 datasheet examples, or using the convenient on-line calculator IDT provides. In the cases above, whenever different values are calculated for frequency accuracy vs. lowest-jitter, the lowest-jitter parameter was chosen.

5.5.2 DDRCLK

The DDR memory controller operates asynchonously to the processor/platform speed (which is derived from SYSCLK). The DDRCLK input is used, along with corresponding PLL configuration pins, to set the DDR clocks. The same ICS307-02 clock generator is used, with the same **ngPIXIS** generated serial settings. The only difference is that there are an additional three programming registers (PX_DCLK[0:2]) to control the fine frequency generation capabilities.

Table 38 summarizes the switch-selectable clock generation possibilities.

Table 39. DDRCLK Frequency Options

cfg_ddrclk	Selected DDRCLK	Actual DDRCLK	Error	ICS307 Control Word	Notes
0 0 0	33.333 MHz	33.3330 MHz	0 ppm	0x200381	
0 0 1	40.000 MHz	39.9996 MHz	10 ppm	0x200501	

Table 39. DDRCLK Frequency Options

cfg_ddrclk	Selected DDRCLK	Actual DDRCLK	Error	ICS307 Control Word	Notes
0 1 0	50.000 MHz	49.9995 MHz	10 ppm	0x220501	
0 1 1	66.666 MHz	66.6660 MHz	0 ppm	0x270501	
1 0 0	83.333 MHz	83.3325 MHz	6 ppm	0x230381	
1 0 1	100.000 MHz	99.9990 MHz	10 ppm	0x230501	
1 1 0	133.333 MHz	133.332 MHz	7.5 ppm	0x210201	Default
1 1 1	166.666 MHz	166.6650 MHz	6 ppm	0x210381	

5.5.3 BCLK

BCLK is a reference 14.318MHz clock source, used by the **P2020** as a real-time clock input, and by the NVidia M1575 for general internal timing, and by the AC97 codec for audio timing.

5.5.4 REFCLK

REFCLK is the clock used by devices connected to the SERDES block, and is provided by the IDT ICS9FG108. It is a differential clock and is routed to each SERDES endpoint. The default frequency is 100.00MHz for PCI Express, SGMII and RapidIO; 125.00MHz is also supported.

Other frequencies may be selected (for test purposes) by programming the device from the I2C bus.

5.6 System Reset

Figure 28 shows the reset connections of P2020DS.

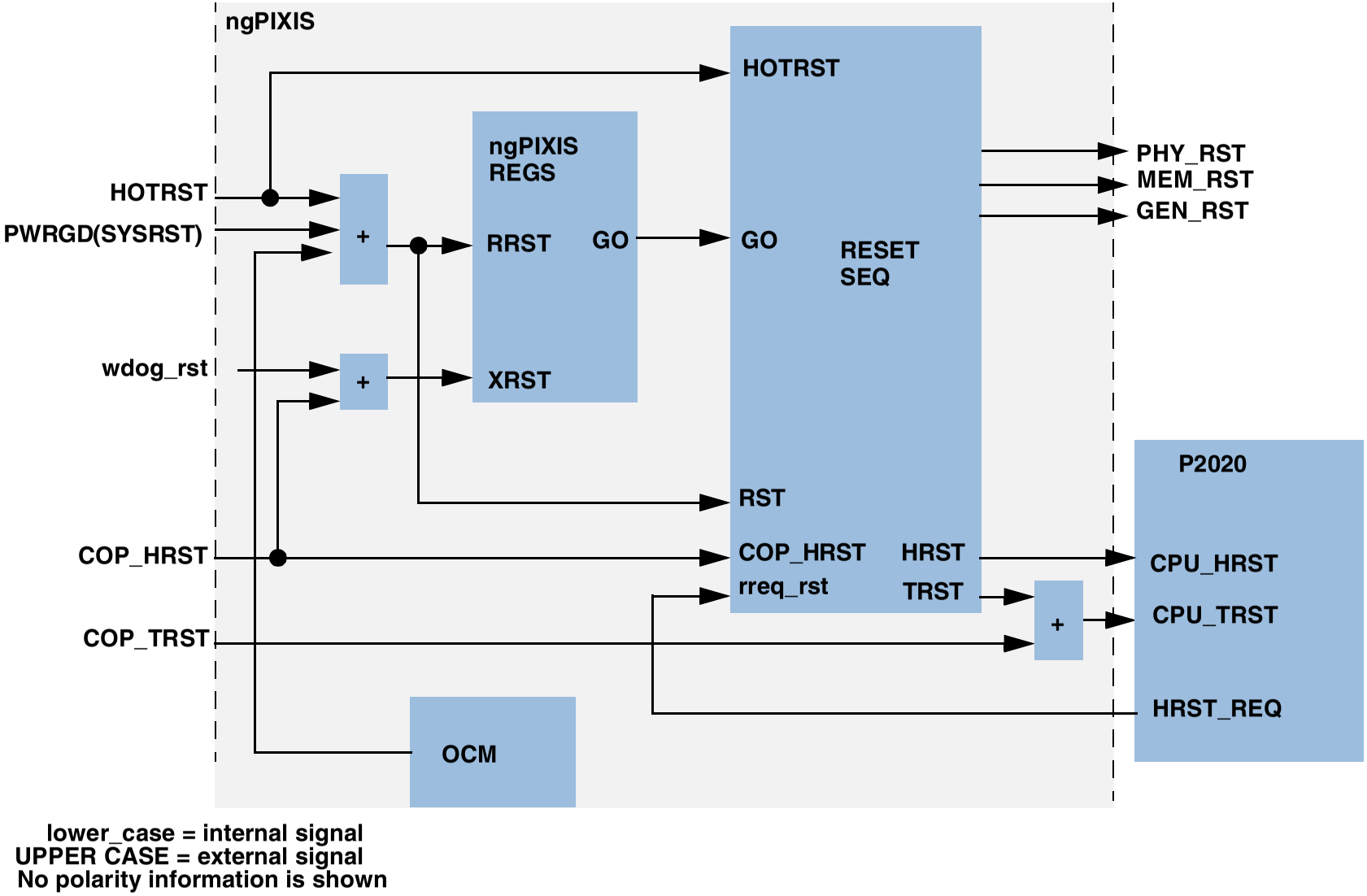

Figure 28. Stingray Reset Hierarchy

From the above figure, the following can be inferred:

- PIXIS registers are reset by every reset input and GO (which is an output controlled by VELA, in turn controlled by PIXIS registers).
- Most PIXIS registers are reset by either RRST or XRST, except one, PX_AUX, which is reset ONLY by RRST (it is unaffected by COP_HRST and wdog_rst).
- If the watchdog timer expires, all internal settings (including VELA-controlled configuration) are reset.
- If the COP COP_HRST signal is asserted, all internal settings (including VELA-controlled configuration) are reset.
- The reset sequencer is triggered upon “GO”, “COP_HRESET”, or “RST”. The sequencer performs identically, except that when triggered by COP_HRST” it does NOT assert CPU_TRST; in all other cases, it does.
- The reset sequencer controls CPU_HRST; it must run for the COP_HRST signal to be passed through.
- Conversely, CPU_TRST is wire-OR’ed with the sequencer, so COP has control of CPU_TRST directly (essentially).

All reset operations are conducted within various portions of the **ngPIXIS**; refer to Section 5.3.1.2 for details. Due to the many reset resources and outputs, reset generation is a little more complicated than normal. Table 40 summarizes reset terms.

Table 40. Reset Terms

Term	Description	Notes
INPUT TERMS		
HOT_RST*	Low until VCC_HOT_3.3 is stable, high thereafter.	Only toggles when power supply is removed/unplugged.
PWRGD	Low until ATX power supply is stable, or while system reset is asserted (motherboard switch or chassis-cabled switch)	Asserted after PWRON* asserted by NVidia, or by manual user intervention.
COP_HRST*	Asserted under COP control.	Must never cause CPU_TRST* to be asserted.
COP_TRST*	Asserted under COP control. Drives CPU_TRST*.	
VELA “GO”	Assered by s/w (local or remote). Triggers configuration-controlled startup.	
RESET_REQ*	Asserted by CPU(s) to start self-reset.	Short duration -- needs stretching.
OUTPUT TERMS		
CPU_HRST*	Restarts **P2020** cores.	Cannot directly cause CPU_TRST*
CPU_TRST*	Resets **P2020** JTAG controller.	Must be asserted by others when COP is not attached. Must not be asserted by others when COP is attached.
PHY_RST*	Soft-reset of PHY.	
GEN_RST*	Hard-reset of PHY and other devices.	
CFG_DRV*	Asserted one clock beyond CPU_HRST* to insure adequate configuration sampling.	

Some of the important guidelines for creating the reset controller are:

- PWRGD from the ATX power supply is also the general system reset
- COP_TRST* must be asserted during normal, non-COP startup.
- COP_TRST* must not be asserted if COP asserts COP_HRST*
- COP_HRST* must reset the target system as well as the processor HRESET* inputs.
- HRESET_REQ* is only 2-3 clock cycles and requires pulse stretching.
- For shmoo/test tracking, one register (PX_AUX) must be reset by all reset sources EXCEPT COP_HRST and WDOG_RST.

6 Configuration

There are three categories of configuration options:

- those options which require software-configuration to support evaluation,
- those options which are expected to be easily and often changed by the end-user/developer, and
- those which should rarely or never be changed.

The first two options are implemented with “DIP switches” and/or software-settable options, while the latter set are usually implemented by resistors which must be added or removed by competent technicians.

For those signals configured using switches, the configuration logic is as shown in Figure 29.

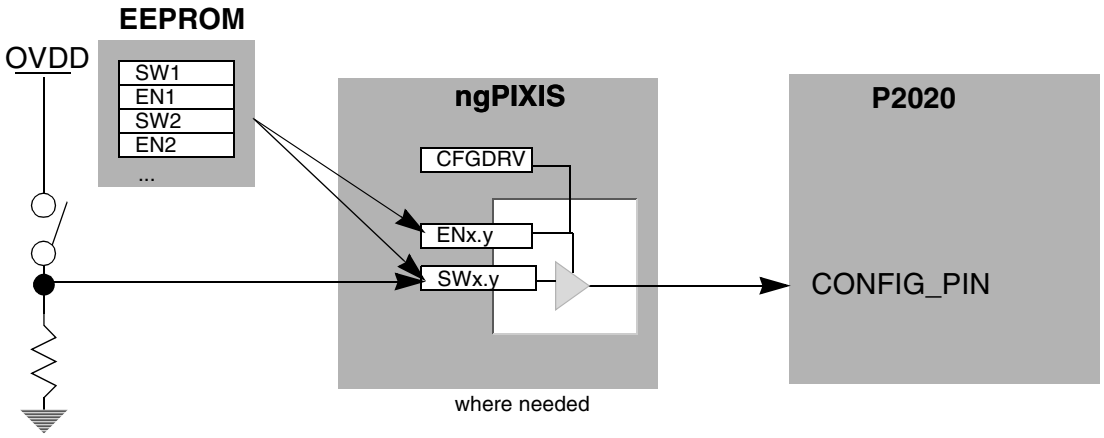

Figure 29. Configuration Logic

The default action is for the **ngPIXIS** to transfer the switch setting to the processor cofiguration pin during the HRESET_B assertion interval.

In addition, software running on the **P2020** can initialize internal registers (SWx, ENx) to allow a board to configure itself for the next restart (termed “self-shmoo” or “self-characterization”).

A third option allows the **ngPIXIS** to copy configuration data from an external I2C EEPROM upon reset, and apply those values to the SWx/ENx registers (ignoring the external hardware switches). This allows dispensing with the expensive DIP switches and their corresponding difficulties with setup and configuration preservation.

6.1 Hardware Configuration

Table 48 summarizes the configuration options supported by Stingray.

Table 41.

Config Definition	Pins	Internal Pullup?	Type	ngPIXIS Mapping	Schematic Implementation	Notes
cfg_srds_refclk	TSEC1588_ ALARMOUT1	Y	pulldown	-	1K pulldown (DNP)	1=100, 0=125
cfg_sgmii3	TSEC1588_ ALARMOUT2	Y	SW5#7	SW_SGMII3	CFG_T1588(0)	1=RGMII 0=SGMII
cfg_ddr_pll[0]	TSEC1588_ CLKOUT	Y	SW2#1	SW_DDRPLL(0)	CFG_T1588(3)	3/4/6/8/10/12 /14/rsvd : 1
cfg_ddr_pll[1]	TSEC1588_ PULSEOUT1	Y	SW2#2	SW_DDRPLL(1)	CFG_T1588(1)	
cfg_ddr_pll[2]	TSEC1588_ PULSEOUT2	Y	SW2#3	SW_DDRPLL(2)	CFG_T1588(2)	
cfg_tsec_reduce	EC_MDC	Y	pulldown	-	1K pulldown	always ’0’
cfg_tsec1_prtcl[1]	TSEC1_TXD7	Y	pulldown	-	1K pulldown	always ’0’ (RGMII=10)
cfg_rom_loc[0]	TSEC1_TXD6	Y	SW4#5	SW_ROMLOC(0)	CFG_TSEC1(6)	
cfg_rom_loc[1]	TSEC1_TXD5	Y	SW4#6	SW_ROMLOC(1)	CFG_TSEC1(5)	
cfg_rom_loc[2]	TSEC1_TXD4	Y	SW4#7	SW_ROMLOC(2)	CFG_TSEC1(4)	
cfg_io_ports[0]	TSEC1_TXD3	Y	SW4#1	SW_IOPORTS(0)	CFG_TSEC1(3)	
cfg_io_ports[1]	TSEC1_TXD2	Y	SW4#2	SW_IOPORTS(1)	CFG_TSEC1(2)	
cfg_io_ports[2]	TSEC1_TXD1	Y	SW4#3	SW_IOPORTS(2)	CFG_TSEC1(1)	
cfg_tsec1_prtcl[0]	TSEC1_TXD0	Y	pullup	-	n/c	always ’1’ (RGMII=10)
cfg_rom_loc[3]	TSEC1_TX_ER	Y	SW4#8	SW_ROMLOC(3)	CFG_TSEC1(0)	
cfg_tsec2_prtcl[1]	TSEC2_TXD7	Y	pulldown	-	1K pulldown	always ’0’ (RGMII=10)
cfg_fuse_rd_en	TSEC2_TXD6	Y	pullup/down	-	1K pulldown (DNP)	
cfg_sdhc_cd_pol_sel	TSEC2_TXD5	Y	SW8#8	SW_SDHC_POL	CFG_TSEC2(3)	
cfg_device_ID[7]	TSEC2_TXD4	Y	SW5#8	-	CFG_TSEC2(2)	
cfg_device_ID[6]	TSEC2_TXD3	Y	pullup/down	-	1K pulldown (DNP)	
cfg_device_ID[5]	TSEC2_TXD2	Y	pullup/down	-	1K pulldown (DNP)	

Table 41.

Config Definition	Pins	Internal Pullup?	Type	ngPIXIS Mapping	Schematic Implementation	Notes
cfg_dram_type	TSEC2_TXD1	Y	FPGA	DDR2_TEST_ DETECT_B	CFG_TSEC2(1)	from DIMM module: 0 for DDR3, else DDR2
cfg_tsec2_prtcl[0]	TSEC2_TXD0	Y	pullup	-	n/c	always '1' (RGMII=10)
cfg_io_ports[3]	TSEC2_TX_ER	Y	SW4#4	SW_IOPORTS(3)	CFG_TSEC2(0)	
cfg_gpinput[0:15]	LAD[0:15]	Y	SW8#(3:4)	SW_GPCFG(0:1)	(via databus)	LSBs only
cfg_cpu1_boot	LA[16]	Y	SW1#8	SW_CPU1_BOOT	CFG_CPU1_BOOT	
cfg_ddr_pll_fdbk_se l	LA[17]	Y	SW2#4	SW_DDRPLLFD	CFG_DDRPLLFD	
cfg_host_agt[1]	LA[18]	Y	SW5#2	SW_HOSTAGT(1)	CFG_HOSTAGT(1)	
cfg_host_agt[2]	LA[19]	Y	SW5#3	SW_HOSTAGT(2)	CFG_HOSTAGT(2)	
cfg_eng_use[0:2]	LA[20:22]	Y	pullup/do wn	-	-	
cfg_cpu0_boot	LA[27]	Y	SW1#4	SW_CPU0_BOOT	CFG_CPU0_BOOT	
cfg_eng_use3	LA[28]	Y	n/a	-	pulldown	8572 interposer only: 8572 SGMII1 config
cfg_sys_pll[0:2]	LA[29:31]	Y	SW2#5-7	SW_SYSPLL(0:2)	CFG_SYSPLL(0:2)	4/5/6/8/10/12 /rsv/rsv : 1
cfg_core1_pll[0]	LWE_B[0]	Y	SW1#5	SW_CORE1PLL(0)	CFG_CORE1_PLL0	
cfg_host_agt[0]	LWE_B[1]	Y	SW5#1	SW_HOSTAGT(0)	CFG_HOSTAGT(0)	
cfg_core0_pll[0]	LBCTL	Y	SW1#1	SW_CORE0_PLL(0)	CFG_CORE0_PLL(0)	
cfg_core0_pll[1]	LALE	Y	SW1#2	SW_CORE0_PLL(1)	CFG_CORE0_PLL(1)	
cfg_rio_sys_size	LGPL0	Y	pullup	-	pullup	
cfg_sgmii2	LGPL1	Y	SW5#6	SW_SGMII2	CFG_SGMII2	1=RGMII 0=SGMII
cfg_core0_pll[2]	LGPL2	Y	SW1#3	SW_CORE0_PLL(2)	CFG_CORE0_PLL(2)	
cfg_boot_seq[0]	LGPL3	Y	SW5#4	SW_BOOTSEQ(0)	CFG_BOOTSEQ(0)	
cfg_boot_seq[1]	LGPL5	Y	SW5#5	SW_BOOTSEQ(1)	CFG_BOOTSEQ(1)	

Table 41.

Config Definition	Pins	Internal Pullup?	Type	ngPIXIS Mapping	Schematic Implementation	Notes
cfg_svr[1]	USB_STP	Y	pullup	SW_SVR(1)	CFG_SVR(1)	SVR (CPU type)
cfg_svr[0]	DMA1_DACK_B	Y	pullup	SW_SVR(0)	DMASET(1)	SVR (CPU type)
cfg_testsel_b	TESTSEL_B	Y	pullup	SW_TESTSEL_B	-	SVR (CPU type)
cfg_mem_debug	DMA2_DACK_B	Y	pulldown	-	-	
cfg_test_port_mux_sel	DMA1_DDONE_B	Y	pullup	-	-	never pulldown
cfg_ddr_debug	DMA2_DDONE_B	Y	pulldown	-	-	
cfg_cvdd_vsel[0]	HRESET_REQ_B	Y	pullup	-	-	
cfg_lvdd_vsel	UART0_SOUT	Y	SW8#7	-		default = 0 = 3.3V; always pulled up
cfg_core1_pll[1]	UART1_SOUT	Y (PD)	pullup	SW_CORE1_PLL(1)	CFG_CORE1_PLL1	
cfg_tsec3_prtcl[0]	UART0_RTS_B	Y	pullup	-	-	always ’1’ (RGMII=10)
cfg_tsec3_prtcl[1]	UART1_RTS_B	Y	pulldown	-	pulldown	always ’0’ (RGMII=10)
cfg_por_abist	TRIG_OUT	Y	pullup	-	-	never pulldown
cfg_core1_pll[2]	READY_P1	Y	SW1#7	SW_CORE1_PLL(2)	CFG_CORE1_PLL2	
cfg_elbc_ecc	MSRCID[0]	Y	SW7#6	SW_ELBC_ECC	CFG_ELBC_ECC	
cfg_bvdd_vsel[1]	MSRCID[1]	Y	SW8#6	-	-	
cfg_60x	MSRCID[2]	Y	n/a	-	-	never pulldown
cfg_global_sfto	MSRCID[3]	Y	n/a	-	-	never pulldown
cfg_bvdd_vsel[0]	MSRCID[4]	Y	n/a	-	-	never pulldown
cfg_test_port_dis	MDVAL	Y	n/a	-	-	never pulldown
cfg_cvdd_vsel[1]	ASLEEP	Y	pullup	-	-	
NON-CPU-CFG						
cfg_lbmap				SW_LBMAP(0:2)	-	

Table 41.

Config Definition	Pins	Internal Pullup?	Type	ngPIXIS Mapping	Schematic Implementation	Notes
cfg_swap				SW_VBANK(0:1)	CFG_VBANK(0:1)	
cfg_flashwp				SW_FLASHWP_B	-	
cfg_pixisopt				SW_PIXISOPT(0:1)	-	
cfg_vcore(6:0)				SW_VCORE(6:0)	CFG_VCORE(6:0)	
cfg_vserdes				SW_VSERDES	CFG_VSERDES	
cfg_refspread				SW_REFSPREAD	CFG_REFSPREAD	
cfg_refclksel				-	CFG_REFCLKSEL	
cfg_serdes_ef				SW_SERDES_EF	CFG_SERDES_EF	
cfg_serdes_ab				SW_SERDES_AB	CFG_SERDES_AB	
cfg_sysclk(0:2)				SW_SYSCLK(0:2	-	
cfg_ddrclk(0:2)				SW_DDRCLK(0:2)	-	
cfg_sd8xmux					CFG_SD8XMUX	

6.2 Configuration Options

The P2020DS uses a function in the **ngPIXIS** called the “OCM” - the Offline Configuration Manager - to control how the board is configured as part of the normal power-up sequence. The OCM monitors a two-position switch, which allows selecting one of the configuration modes shown in Table 45.

Table 42. P2020DS Configuration Options

SW_CFGOPT(0:1)	Configuration Mode	Description
00	Normal	Switch settings control all system configurations. No software in the OCM is executed. The following functions are NOT available: * Real-time PVT data collection * VCORE voltage changes
01	n/a	
10	Memory	If VCTL=1 (self-shmoo is in process), no changes are made; otherwise, Pixis SW/EN registers are loaded from external EEPROM, and for every bit in acontrol all system settings, For every bit in an enable register that is set to one, the corresponding bit in a switch register is driven onto the board during HRESET.
11	Interactive	Same as “Memory” mode, except that during startup, the OCM pauses and allows the user to alter and optionally re-write the memory with new configuration settings with a UART connected to COM1. If no IO is received within 20 seconds, IO is stopped and “Memory” mode is used.

6.2.1 Configuration Mode: Normal

In this mode, the OCM does not change any switch settings; whatever is selected on the configuration switches will be used during the HRESET configuration sampling interval.

It is also possible for the system to change its own configuration registers and initiate a software-controlled restart (“self-shmooing”), just as with previous platforms.

This mode is backwards-compatible with previous generation platforms.

6.2.2 Configuration Mode: Memory

In memory configuration mode, the **ngPIXIS** registers are initialized as before. If the system is not in self-shmoo mode, the OCM loads values from the SW(1:8) and EN(1:8) registers from the I2C-based EEPROM storage at device address 0x55. This device is isolated and powered from the standby power supply and so is available even before the system has been powered up.

The OCM transfers data from the EEPROM into the **ngPIXIS** SW/EN registers, as shown in Table 43.

Table 43. OCM Configuration Data Format

I2C EEPROM	Description	ngPIXIS Register
Source Addresses		Destination Address
0x00 - 0x03	EEPROM header	n/a
0x08	Set Selection bit 3=1: Use VCORE value bit 3=1: Do not use VCORE bit 0(lsb)=0: Use Set A bit 0(lsb)=1: Use Set B	n/a
0x20..0x3F	Set A: SW1, EN1 SW2, EN2 SW3, EN3 SW4, EN4 SW5, EN5 SW6, EN6 SW7, EN7	SW(x), EN(x) 20, 21, 22, 23, 24, 25 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F
0x40..0x5F	Set B: SW1, EN1 SW2, EN2 SW3, EN3 SW4, EN4 SW5, EN5 SW6, EN6 SW7, EN7	SW(x), EN(x) 20, 21, 22, 23, 24, 25 26, 27, 28, 29, 2A, 2B, 2C, 2D, 2E, 2F
0x70, 0x71	VCORE output code (if enabled)	PMBus output code (MSB first)
0xA0, 0xA1	**ngPIXIS** register edits. Ends on values of 0x00 or 0xFF.	Address, then Data

The OCM examines the LSB of byte 0x08, and transfers either “Set A” (LSB=0) or “Set B” (LSB=1).

In all cases, when the registers have been loaded, during reset configuration, whereever an EN register has a bit set to “1”, the corresponding bit in a “SW” register is used to replace the selected switch setting.

For example, that if all EN(1:8) registers are set to “11111111”, then no external switch settings will be used, and so the system will be purely EEPROM-configured. This allows a system cost reduction by eliminating all but one of the (fairly expensive) DIP switches.

Conversely, that if all EN(1:8) registers are set to “00000000”, then the register settings are unchanged and the external switch swttings will be used.

During “Memory” mode, the OCM is still running code. The target system may communicated with it via message passing; refer to the OCM section for details.

6.2.3 Configuration Mode: Interactive

In this mode, the OCM prints a message to the COM1 serial port (115200, 8/N/1) and waits up to 20 seconds for a keypress. During this time, if the system is power up, it will pause until this delay has completed. If no character is is received, the OCM mode falls back to “Memory” mode.

Otherwise, while the system is still prevented from powering up, commands are accepted over the serial port, allowing SW/EN registers to be edited in memory. When the user has completed any configuration commands, the “GO” command allows the normal power-up sequencer to proceed.

The OCM software supports several commands, as described in Table 44.

Table 44. OCM Commands

Command	Function
E	Edit all SW registers, with EN defaulting to ‘1111_1111’.
EN	Edit EN register.
GO	GO: Allow power and reset startup sequences to proceed when needed.
HE	Help: list commands.
I	Initialize EEPROMs to factory defaults.
IN	Show system INfo
OP	Options: set various options.
PD	Pixis Display: show contents of PIXIS registers.
PJ	Flash PromJet: toggle between Flash and PromJet mode.
PM	Pixis Modify: alter contents of PIXIS register.
PW	Power: toggle power on/off.
Q	Quit: exit all interaction and shut down. Power and reset sequencers are allowed to proceed.
RS	Reset: assert, then de-assert HRESET to the target.
S	Show SW and EN registers.
SA	Select and use switch “Set A”.
SB	Select and use switch “Set B”.
ST	STOP: Do not allow power and reset startup sequences to proceed when needed.
SV	Set a VCORE voltage value to be stored in EEPROM. Will be sent to the VDD supply on startup, if enabled.
SW	Edit an SW register.
V	Set VCORE to entered value immediately. Note: there is limited protection against destroying the processor.
VB	Flash VBank: toggle flash VBANK(0).

Table 44. OCM Commands

Command	Function
WR	Write: Save SW/EN/SA/SB/SW edits in EEPROM. If this command is not used, cycling the power will result in previous values being used.
OPTIONAL COMMANDS	
MD	Memory Display: show internal memory.
MM	Memory Modify: alter internal memory.

Here is a sample transcript, showing the OCM starting up, and the user altering the configuration for the CPU0 core PLL setting (from '111' to '100') and changing CPU1 to boot-hold-off mode:

```
OCM: Press any key to enter setup: [####################]
OCM: Offline Config Manager V6/2009_0203
     Cohost: P2020DS/Stingray
OCM>>PD
  PIXIS Registers:
  R00=16          R01=01          R02=01          R03=00
  R04=00          R05=00          R06=00          R07=00
  R08=F7          R09=00          R0A=00          R0B=00
  R0C=00          R0D=67          R0E=00          R0F=00
  R10=10          R11=11          R12=12          R13=00
  R14=00          R15=00          R16=00          R17=00
  R18=00          R19=00          R1A=00          R1B=00
  R1C=00          R1D=00          R1E=00          R1F=00
  R20=11          R21=FF          R22=12          R23=FF
  R24=13          R25=FF          R26=14          R27=FF
  R28=15          R29=FF          R2A=16          R2B=FF
  R2C=17          R2D=FF          R2E=18          R2F=FF
OCM>>SW
  SW1=79 ? 98
OCM>>EN
  EN1=FF ? F1
OCM>>S
  Using set "A"
  Switch Settings:
  SW1=98          EN1=F1
  SW2=08          EN2=FF
  SW3=F6          EN3=FF
  SW4=4D          EN4=FF
  SW5=FE          EN5=FF
  SW6=07          EN6=FF
  SW7=4F          EN7=3F
  SW8=07          EN8=FF
OCM>>WR
  Writing to EEPROM...
OCM>>Q

...Shutting down...
```

After ‘Q’ is entered, assuming the system was powered up and waiting, the configuration registers are set and the ngPIXIS will drive this data onto the pins accordingly.

6.2.4 Configuration Switches

The SW registers are formatted as shown in Table 45.

Table 45. Configuration Switches Format

DIP Switch Label	1	2	3	4	5	6	7	8
ngPIXIS Register Bit (Power Arch. “big endian” format)	0	1	2	3	4	5	6	7

Switch names exactly match the name on the schematics and on the printed-circuit board. The switches are summarized in Table 46.

Table 46. Configuration Switches

DIP Switch	Individual Switch(es)	Register	Configuration Controlled	Class
SW1	1/2/3	SW1[0:2]	cfg_core0_pll(0:2)	Dynamic
	4	SW1[3]	cfg_cpu0boot	
	5/6/7	SW1[4:6]	cfg_core1_pll(0:2)	
	8	SW1[7]	cfg_cpu1boot	
SW2	1/2/3	SW2[0:2]	cfg_ddrpll(0:2)	Dynamic
	4	SW2[3]	cfg_ddrpllfd	
	5/6/7	SW2[4:6]	cfg_syspll(0:2)	
	8	SW2[7]	n/a	
SW3	1	SW3[0]	cfg_refspread	Static
	2	SW3[1]	n/a	
	3/4/5	SW3[2:4]	cfg_ddrclk(0:2)	Static
	6/7/8	SW3[5:7]	cfg_sysclk(0:2)	
SW4	1/2/3/4	SW4[0:3]	cfg_ioports(0:3)	Dynamic
	5/6/7/8	SW4[4:7]	cfg_romloc(0:3)	
SW5	1/2/3	SW5[0:2]	cfg_hostagt(0:2)	Dynamic
	4/5	SW5[3:4]	cfg_bootseq(0:1)	
	6	SW5[5]	cfg_sgmii2	
	7	SW5[6]	cfg_sgmii3	
	8	SW5[7]	cfg_sdhc_pol	

Table 46. Configuration Switches

DIP Switch	Individual Switch(es)	Register	Configuration Controlled	Class
SW6	1/2	SW6[0:1]	cfg_vcore(0:1)	Static
	3	SW6[2]	cfg_vserdes	
	4	SW6[3]	cfg_serdes_ab	
	5	SW6[4]	cfg_serdes_ef	
	6/7	SW6[5:6]	cfg_svr(0:1)	Dynamic
	8	SW6[7]	cfg_testsel_b	
SW7	1/2	SW7[0:1]	cfg_lbmap(0:1)	Static
	3/4	SW7[2:3]	cfg_vbank(0:1)	
	5	SW7[4]	n/a	
	6	SW7[5]	cfg_elbc_ecc	Dynamic
	7/8	SW7[6:7]	n/a	
SW8	1	SW8[0]	cfg_core0speed	Dynamic
	2	SW8[1]	cfg_core1speed	
	3	SW8[2]	cfg_platspeed	
	4	SW8[3]	cfg_ddrspeed	
	5	SW8[4]	cfg_sysspeed	
	6/7/8	SW8[5:7]	n/a	

where “Dynamic” are those configuration pins which are only asserted during HRESET_B (these are also processor-only configuration pins), while “Static” configuration pins remain constant as long as the system power is operational.

6.2.5 Standard Configuration Settings

The following table shows the standard configuration settings for the P2020DS board. Note that processor and DDR3 speed changes may result in a shipped system having different configuration values - do not change shipped systems unless you know what you are doing.

Table 47 is accurate for a P2020 running at:

- 1200 MHz core speed
- 600 MHz platform speed
- 666 MHz DDR3 speed
- 100 MHz SYSCLK speed

As usual, a ‘1’ means the switch is ON, or UP (depending on board orientation).

Table 47. Default Configuration Switches

DIP Switch	Setting (hex)	Setting (binary)	Description
SW1	98	1001_1000	Core ratios are set to 2X CPU0 may boot CPU1 is in boot-holdoff.
SW2	96	1001_0101	DDR raio is 10X PLAT ratio is 6X
SW3	5D	0101_1101	Spread-spectrum clocking disabled DDRCLK is 66 MHz SYSCLK is 100 MHz
SW4	2E	0010_1110	PEX1: Enabled, 1x PEX2: Enabled, 1x PEX3: Enabled, 2x Boot location: GPCM 16-bit
SW5	FF	1111_1111	PEX1: host PEX2: host PEX3: host Boot sequencer: disabled TSEC2: RGMII mode TSEC3: RGMII mode SDHC CD: active-low
SW6	17	0001_0111	VCORE: hardware-defined VSERDES: default SERDES_AB: Split PEX to slot and NVidia M1575. SERDES_EF: Route to Slot. SVR: default
SW7	4F	0000_1111	LBMAP: Boot from NOR flash VBANK: No virtual banking. ECC: disabled IDWP: protected FLASHWP: not protected.
SW8	FF	1111_1111	CORE1: >1GHz CORE2: >1GHz PLAT: >= 333 MHz DDRCLK: >= 500 MHz SYSCLK: >= 66 MHz
SW9	7C	0111_1101	OCM: reset GPCFG: “11” IPLWP: protected. CFGWP: protected CFGOPT: use switches.

7 Applications Support

The following functions are present on the board to support the Freescale Applications/Helpline group. These functions, while possibly of use to customers, may or may not be applicable to other applications and can generally be removed from the board

7.1 Parametric Support

In order to support testing under various voltage, frequency, and temperature conditions, P2020DS has several features already covered in the various power and clock sections. Additional features for clocks are:

- Ability to inject external clocks.
  This may involved adding or removing connectors, resistors or other components on the board, as signal integrity considerations trump convenience factors.

Additional features for current measurement are at least of the following for each power supply that consumes at least 1 amp:

- Banana jack Series low-ohm resistor
  By measuring voltage drop across the resistor, current can be determined.
- Current-mirror across low-ohm resistor
  The developed voltage proportional to the current is measured by an A/D converter.
- Directly measurable current reporting.
  Using dedicated pins of the power supplies to measure/infer power levels or PMBus-controlled devices.

7.2 Debug Support

For debug purposes, Table 48 summarizes the debug support options for various P2020DS subsystems. All debug attachment methods are non-intrusive.

Table 48. P2020DS Debug Options

Subsystem	Debug Method	Notes
SERDES	Mid-point TAP PCI Express analyzer cards	
DDR	Nexus DDR3 breakout card	
Local Bus	P6880 logic analyzer attachments	
GPIO	2x8 “Berg” header	Alternate functions must be disabled in many cases.
PCI	PCI Slot Analyzer card	

Debugging information is provided on 8 LEDs; normally these LEDs indicate hardware activity as shown in Table 49, but software can override these and use them for software debugging purposes.

Table 49. LED functions

LED	Reset Active OCM Debug Off	Reset Active OCM Debug Active	Reset Inactive PX_CSR[LED]=0	Reset Inactive PX_CSR[LED]=0
0	All on (lamp test)	OCM defined	Boot (LCS0_B) activity	User-defined
1			Flash (LCS1_B) activity	
2			IC2 (SCL(0:1)) activity	
3			INTR signal transitions	
4			ASLEEP - set if CPU config err.	
5			OCM Fail - set if errors occurred.	
6			OCM software activity.	
7			Heartbeat - ngPIXIS clock monitor	

8 PCB Development Issues

General requirements include:

- All vias open (unmasked) so that flying probe can make contact to vias.
- Silkscreen should be placed so that soldermask openings don't clip text (wherever possible).

8.1 Material

The PCB shall be constructed using non-lead processing in compliance with RoHS standards.

The PCB shall be immersion-gold plated or OSP protected.

8.2 Dimensions

The PCB dimensions and mounting holes are based on the standard full-size ATX formfactor (formfactor.org).

8.3 Pad Stack

For the **P2020**:

- .010 drill
- .019 pad
- .030 antipad/plane clearance.

This is for 4mil trace routing between pads (dual track).

For all other components traditional (though lead-free) padstacks may be used.

8.4 Stackup

The recommended 14-layer stackup is shown in Figure 30

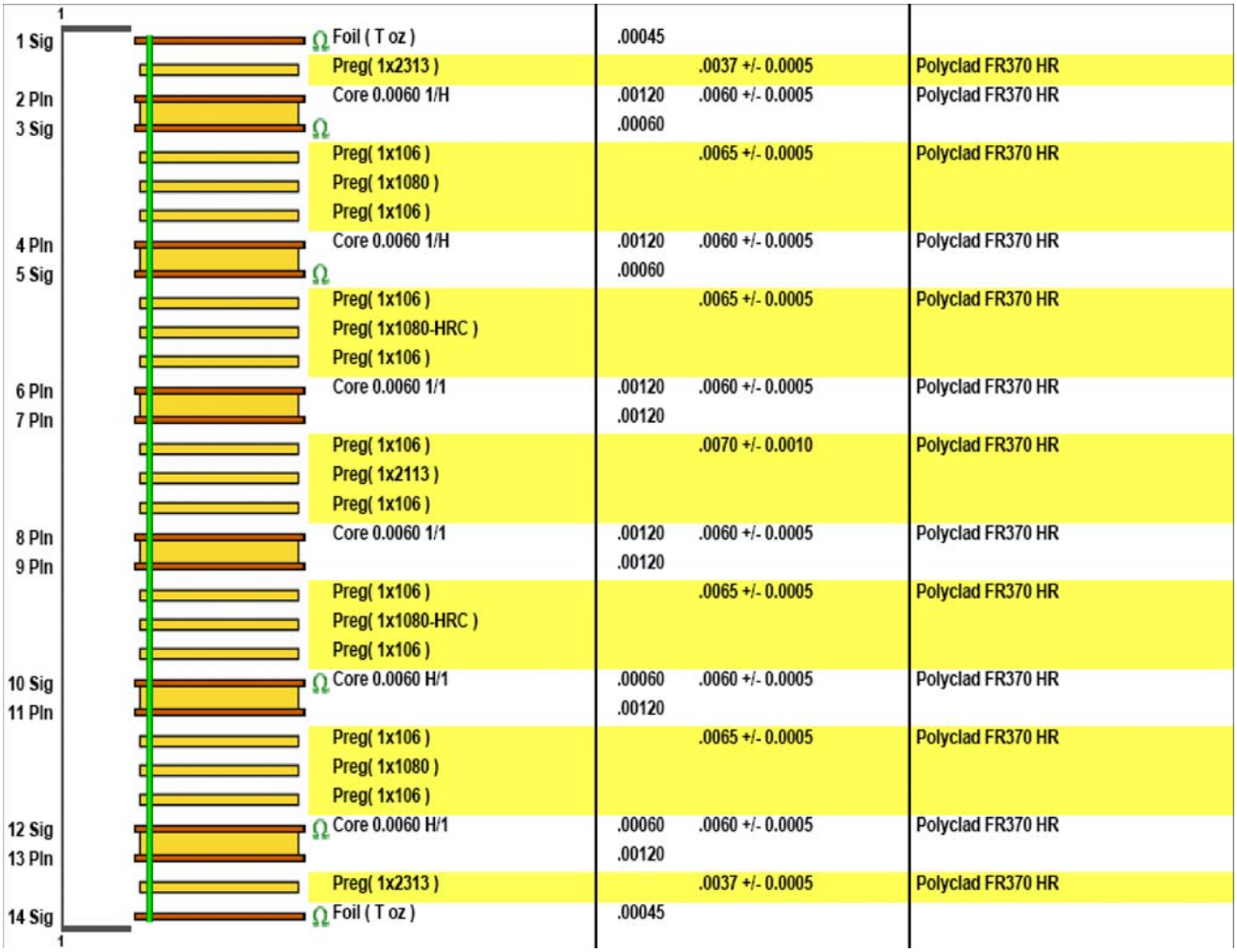

Figure 30. PCB Stackup

Although the **P2020** is compatible with a 4-signal-layer escape, the development board is not slated for a high-volume production line where the effort to do a 4-signal-layer design is worth the return on investment.

8.5 Components

All components shall be lead-free. All SMT components are 0402, unless otherwise noted.

Components are as noted in the bill of material (BOM), except for these “non-components” (custom) gemoetries:

- “mtg” is an ATX chassis mounting hole, connected to ground.
- “tp_pth” is a test point implemented as a plated-through hole.
  “tp_pth” should be smaller than on FS1/FS2 (size TBD).
  All other test points are “25 mil” pads, or circular 0.1”.

8.6 Placement

8.6.1 General Rules

Resistors, ceramic capacitors (no tantalums or lytics) and ferrites should be placed on the bottom of the board.

ICs should be placed on the top, with the following exceptions:

- single-gate ICs
- any IC in a SO8 or smaller that dissipates <10mW
- FET-gate buffers

In general, nothing over ~0.15” should be on the back.

Keep fiducials 200 mils away from all card edges.

Bulk and high-frequency bypass capacitors are shown on the power supply pins of each device on the schematic page; these caps must be located very near the power pin.

No tall components may exist in the area on the top of the board, such that it might interfere with installing a long the PCI or PCI-Express cards. Components smaller than 0.3” are acceptable.

ICT Probe Rules

- 28-30 mil for standard test point.
- 20-28 mil for smaller test point (less accurate, greater cost on ICT fixture).
- No test points for ICT can be less than 20 mil.

8.6.2 Suggested Placement

The die escape pattern, coupled with the ATX form-factor and recommended component placement produces the placement/escape shown in Figure 31.

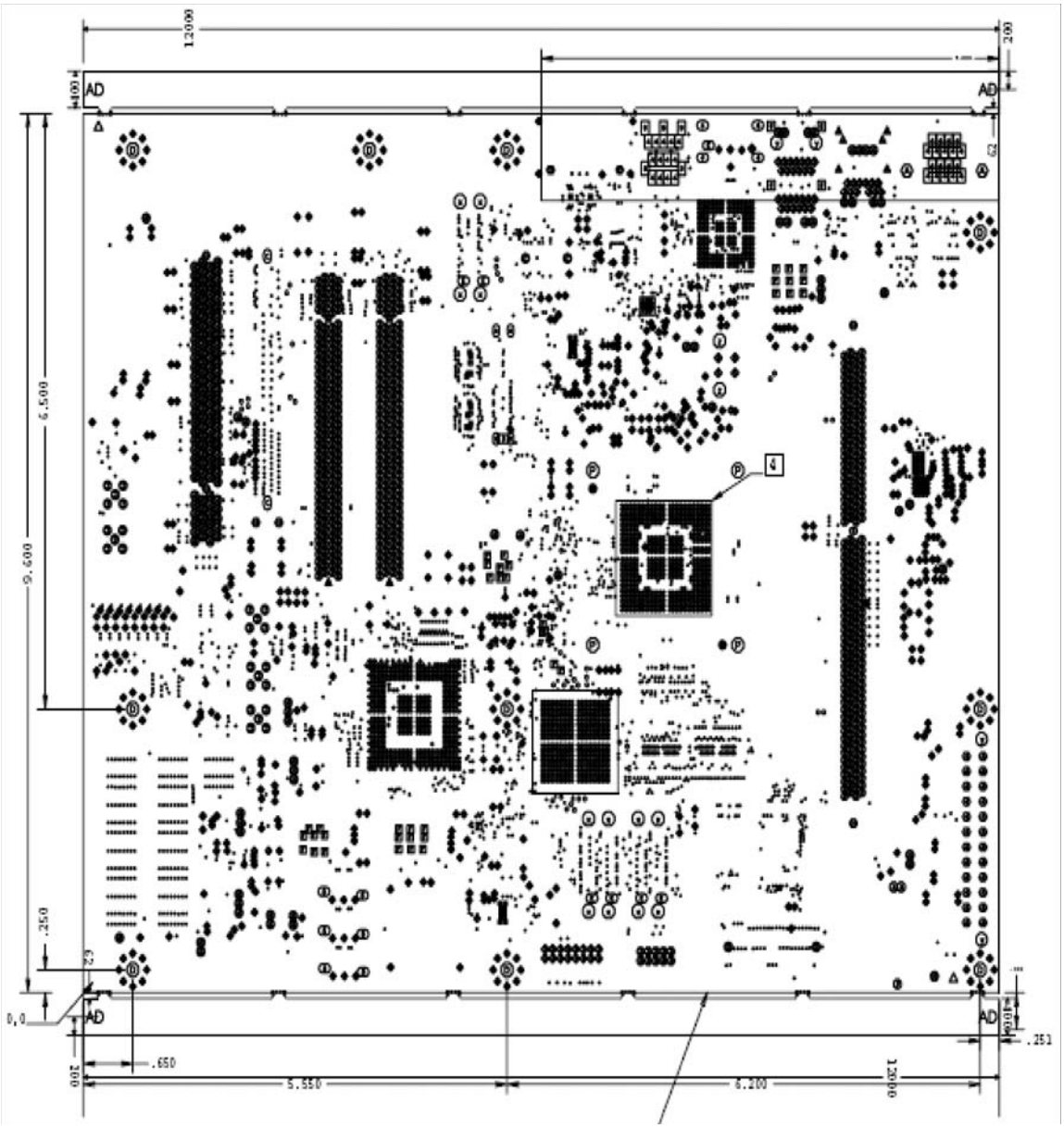

Figure 31. Stingray Layout

To work with existing ATX “I/O shield” hardware, connector placement must follow the alignment shown in Figure 32.

Figure 32. ATX Rear-Panel Escape I/O Shield

8.6.3 Routing

Routing is constrained by a combination of electrical classes and differential pair notations.

Nets should accomodate ICT with testpoints on the bottom where no other pads are accessible (e.g. BGA via pads), where space permits.

All BGA pads should be open on the bottom layer for probing.

All BGA devices should have silkscreen pin number labelling on the bottom layer (grid).

8.6.4 Constraints

There are extensive constraints applicable to this layout, including:

- DDR3 routing
- P2020 wire-bond package compensation for DDR3
- SERDES routing
- SERDES multiplexing and T connections
- Kelvin connections for thermal diodes
- LALE vs LAD setup/hold time adjustments
- Short traces between filtered power and clocks/processor inputs.

Rather than verbally describe these, the Cadence 16.0.1 Allegro tool has the constraints applied to the design so as to produce a controlled layout. Refer to the PCB design database for constraints.

8.6.5 Power Nets

Power Net Name	Description	Restrictions
VCC_HOT_5	Powers 3.3, 2.5 and 1.5V HOT supplies.	Heavy trace with 2A capacity, or localized area fill.
VCC_12	Slot and Fan power	Heavy trace with 1A capacity. Used by slots and fans so should be routed around the periphery of the PCB.
VCC_12_BULK	VCORE power	Area fill for connector and all attached components.
VCC_HOT_3.3	FPGA I/O	Heavy trace/fill between PSU and Actel; 25 mil traces to all other users.
VCC_HOT_1.5	FPGA core	Heavy trace/fill between PSU and Actel; 25 mil traces to all other users.
VCC_HOT_1.8	NVidia	Heavy trace.
VCC_5V	Main power	Shared power plane w/3.3V
VCC_3.3V	Main 3.3V power	Shared power plane w/5V
VCORE	**P2020** core power.	Requires 60A path between PSU endpoint and VCORE vias. Requires 3 planes/fills - see Section 5.1.19 for details.
VCC_1p0	**P2020** IP power	Area fill on inner power plane. Second priority after VCORE.
VCC_1.2	VSC8244 core power,	Area fill
VCC_SERDES	**P2020** XVDD/SVDD pins.	Area fill
VCC_1.8V	NVidia, AC codec, etc.	Area fill
VCC_DDRA_IO	I/O power for DDR interface	Area fill on plane 1

9 Programming Model

This section covers general programming information, and is presented here to help guide the writing of board support packages.

9.1 Address Map

Table 50 shows a typical map of the P2020DS. Since all chip-selects are programmable, almost all devices may be located with impunity, so this map is subject to change.

Table 50. Address Map

Start Address	End Address	SIZE	BAT	LAW	LCS	Register
0_0000_0000	0_3FFF_FFFF	1G	1	1	-	DDR memory
0_4000_0000	0_47FF_FFFF	128M	5	7	2	NAND, rank 1
0_4800_0000	0_4FFF_FFFF	128M	5	7	4	NAND, rank 2
0_5000_0000	0_57FF_FFFF	128M	5	7	5	NAND, rank 3
0_5800_0000	0_5FFF_FFFF	128M	5	7	6	NAND, rank 4
0_6000_0000	0_7FFF_FFFF	512M	6	8	0	FLASH (large space)
0_8000_0000	0_9FFF_FFFF	512M	2	2	-	PEX 1: MEM Space (NVidia M1575)
0_A000_0000	0_AFFF_FFFF	512M	3	3	-	PEX 2: MEM Space (Slot)
0_B000_0000	0_E1FF_FFFF	452M	*reserved*			
0_E200_0000	0_E2FF_FFFF	16M	4	4	-	PEX 2: IO Space (NVidia M1575)
0_E300_0000	0_E3FF_FFFF	16M	4	5	-	PEX 2: IO Space (Slot)
0_E400_0000	0_F7FF_FFFF	335M	*reserved*			
0_F800_0000	0_F80F_FFFF	1M	0	-	-	CCSRBAR space (internal **P2020** registers)
0_F810_0000	0_F81F_FFFF	1M	0	0	3	PIXIS register space
0_F820_0000	0_F82F_FFFF	1M	0	0	7	MRAM space
0_F830_0000	0_FDFF_FFFF	128M	*reserved*			
0_FE00_0000	0_FEFF_FFFF	16M	0	6	1	PromJet (or flash)
0_FFF0_0000	0_FFFF_FFFF	16M	0	6	0	Flash (or PromJet)
1_0000_0000	F_FFFF_FFFF	<64G	*reserved*			

Note that accessing the full 128MB of NOR flash using this space requires that it be relocated to address 0x6000_0000 duing use.

9.2 Software Notes

The following notes may be useful for bringing up new software.

- TBD

9.3 ngPIXIS Registers

The **ngPIXIS** device contains several software-accessible registers which are accessed from the base address programmed for LCS3 (see Section 5.1.5). Table 51 shows the register map of the **ngPIXIS** device.

Table 51. ngPIXIS Register Map

Base Address Offset	Register	Name	Access	Reset
0x00	System ID register	PX_ID	R	0x16
0x01	System architecture version register	PX_ARCH	R	0x01
0x02	**ngPIXIS** version register	PX_SCVER	R	0x02
0x03	General control/status register	PX_CSR	R/W	0x00
0x04	Reset control register	PX_RST	R/W	0x00
0x05	*reserved*	*reserved*	*reserved*	*undefined*
0x06	Auxiliary register	PX_AUX	R/W	0x00
0x07	Speed register	PX_SPD	R	0x00
0x08	Board Configuration register 0	PX_BRDCFG0	R/W	0x11
0x09	DMA Control/Status register	PX_DMA	R/W	0x06
0x0A	SRAM Address Register	PX_ADDR	R/W	0x00
0x0B-0x0C	*reserved*	*reserved*	*reserved*	*undefined*
0x0D	SRAM Data Register	PX_DATA	R/W	*undefined*
0x0E	LED Data Register	PX_LED	R/W	0x00
0x0F	*reserved*	*reserved*	*reserved*	*undefined*
0x10	VELA Control Register	PX_VCTL	R/W	0x00
0x11	VELA Status Register	PX_VSTAT	R	0x00
0x12	VELA Configuration Enable Register 0	PX_VCFGEN0	R/W	0x00
0x13	*reserved*	*reserved*	*reserved*	*undefined*
0x14	OCMCSR Register	PX_OCMCSR	R/W	0x00
0x15	OCMMSG Register	PX_OCMMSG	R/W	0x00
0x16	GMDBG Register	PX_GMDBG	R/W	0x00
0x17-0x18	*reserved*	*reserved*	*reserved*	*undefined*
0x19	SCLK0 Register	PX_SCLK0	R/W	*varies*
0x1A	SCLK1 Register	PX_SCLK1	R/W	*varies*
0x1B	SCLK2 Register	PX_SCLK2	R/W	*varies*
0x1C	DCLK0 Register	PX_DCLK0	R/W	*varies*
0x1D	DCLK1 Register	PX_DCLK1	R/W	*varies*
0x1E	DCLK2 Register	PX_DCLK2	R/W	*varies*
0x1F	WATCH Register	PX_WATCH	R/W	0x7F
0x20, 0x22, ...	SW(1:8) Registers	PX_SW(1:8)	R/W	0x00
0x21, 0x23, ...	EN(1:8) Registers	PX_EN(1:8)	R/W	0x00

The corresponding header file definitions are in Section 9.4.

9.3.1 ID Register (PX_ID)

The ID register contains a unique classification number; this ID number is used by DINK/eDINK and other software to identify board types. This number does not change for any P2020DS revision.

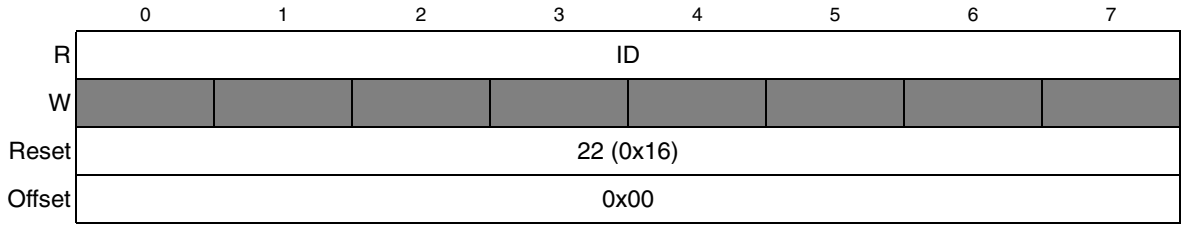

Figure 33. ID Register (PX_ID)

Table 52. PX_ID Field Descriptions

Bits	Name	Description
0-7	ID	Board identification

9.3.2 Architectural Version Register (PX_ARCH)

The version register contains the architectural revision of the P2020DS board.

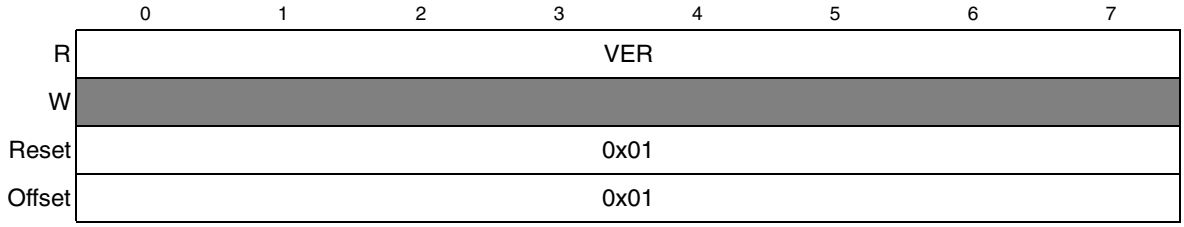

Figure 34. Version Register (PX_ARCH)

Table 53. PX_ARCH Field Descriptions

Bits	Name	Description
0-7	VER	Version Number: %00000001 : Version 1 %00000010 : Version 2 etc.

This register only changes when significant (i.e. software-visible and software-impacting) changes to the board have occurred. Note that changing flash manufacturers, etc. would not be considered an architectural change, as a CFI-compliant flash programmer should be able to handle such a change.

9.3.3 System Control FPGA Version Register (PX_SCVER)

The version register contains the major and minor revision information of the **ngPIXIS** system controller FPGA.

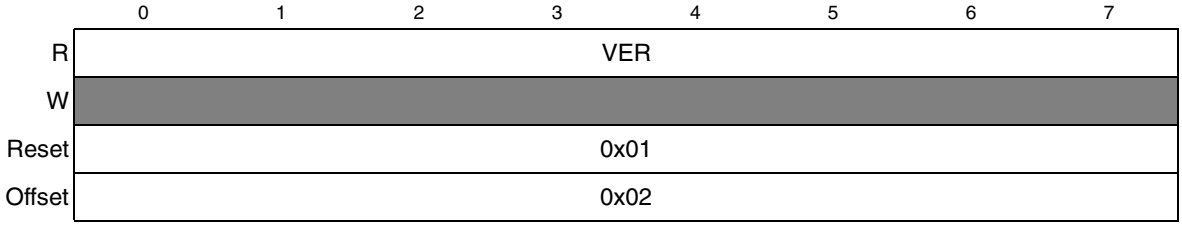

Figure 35. Version Register (PX_SCVER)

Table 54. PX_SCVER Field Descriptions

Bits	Name	Description
0-7	VER	Version Number: %00000001 : Version 1 %00000010 : Version 2 etc.

9.3.4 General Control/Status Register (PX_CSR)

The PX_CSR register contains various control and status fields, as described below.

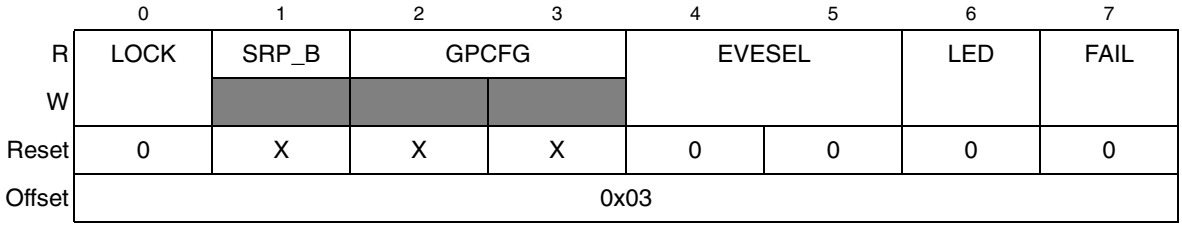

Figure 36. General Control/Status Register (PX_CSR)

Table 55. PX_CSR Field Descriptions

Bits	Name	Description
0	LOCK	If 0: If clear, all registers may be read from or written to. If 1: If set, the following registers may not be written to: PX_RST PX_VCTL
1	SRB_B	If 0: An SGMII-riser card is installed in the slot. If 1: No SGMII-riser card is present.
2-3	GPCFG	This field represents the status present on the GPCFG(0:1) switches, and may be used by software for general-purpose configuration purposes. SW_GPCFG(0) is sampled in PX_CSR[2]. SW_GPCFG(1) is sampled in PX_CSR[3].
4-5	EVESEL	If 00: The EVENT_B switch asserts IRQ0_B (debugger switch). If 01: The EVENT_B switch asserts SRESET_B. If 10: The EVENT_B switch asserts UDE0_B. If 11: The EVENT_B switch asserts UDE1_B.

Table 55. PX_CSR Field Descriptions

Bits	Name	Description
6	LED	If set, the diagnostic LEDs are driven by the value in the PX_LED register; otherwise, the LEDs default to activity monitors.
7	FAIL	If set, the external LED labelled “FAIL” is turned on and the one labelled “PASS” is off; otherwise, if clear, the opposite is true.

9.3.5 Reset Control Register (PX_RST)

The PX_RST register may be used to assert system resets. PX_RST is usually only written -- reads return the value in the register, and do not (necessarily) reflect the value of the system reset.

	0	1	2	3	4	5	6	7
R	ALL	*-rsv-*	*-rsv-*	*-rsv-*	SGMII	PHY	*-rsv-*	GEN
W								
Reset	1	1	1	1	1	1	1	1
Offset	0x04							

Figure 37. Reset Control Register (PX_RST)

Table 56. PX_RST Field Descriptions

Bits	Name	Description
0	ALL	If set to 0, a full system reset is initiated.
1-3	*-rsvd-*	-reserved-
4	SGMII	If 0: SGMII_RST_B is asserted. If 1: SGMII_RST_B is deasserted.
5	PHY	If 0: PHY_RST_B is asserted. If 1: PHY_RST_B is deasserted
6	*-rsvd-*	-reserved-
7	GEN	If 0: GEN_RST_B is asserted. If 1: GEN_RST_B is deasserted

NOTE: the PX_RST register bits are not self-resetting. PX_RST[ALL] is reset only as a side-effect of trigging a full system reset. The other bits must be cleared with software.

NOTE: these register-based resets are OR’d with existing reset sequencer outputs. Setting these bits while a VELA configuration cycle is active may have unpredictable results.

9.3.6 Auxiliary Register (PX_AUX)

The PX_AUX register is a general-purpose read/write register. It reset upon initial power activation, or by chassis reset sources; PX_AUX preserves its value between COP- or watchdog-initiated resets.

	0	1	2	3	4	5	6	7
R	USER							
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x06							

Figure 38. Auxiliary Register (PX_AUX)

Table 57. PX_AUX Field Descriptions

Bits	Name	Description
0-7	USER	User defined.

9.3.7 Speed Register (PX_SPD)

The PX_SPD register is used to communicate the current user settings for the SYSCLK and DDRCLK clock generators. These values are used to specify one of eight “startup” values for each clock. This register is typically needed for software to be able to accurately initialize timing-dependant parameters, such as those for local bus, DDR memory, I2C clock rates, and more..

	0	1	2	3	4	5	6	7
R	PIXOPT		DDRCLK			SYSCLK		
W								
Reset	X	X	X	X	X	X	X	X
Offset	0x07							

Figure 39. Power Status Register (PX_SPD)

Table 58. PX_SPD Field Descriptions

Bits	Name	Description
0-1	PIXOPT	Reflects the SW_PIXOPT(0:1) switch settings.
2-4	DDRCLK	Reflects switch settings as described in Table 38.
5-7	SYSCLK	Reflects switch settings as described in Table 38.

9.3.8 Board Configuration Register (PX_BRDCFG0)

The PX_BRDCFG0 register is used to control the board configuration. Unlike other configuration settings, those controlled by this register may be changed at any time.

	0	1	2	3	4	5	6	7
R W	VCORE_MGN		NGI2C_ ACC	DEVID7	PJWP_B	REFCLK	USB_ID	SD8X
Reset	0	0	0	1	0	0	0	1
Offset	0x08							

Figure 40. Board Configuration Register 0 (PX_BRDCFG0)

Table 59. PX_BRDCFG0 Field Descriptions

Bits	Name	Description
0-1	VCORE_MGN	This field controls the MGN pin of the ZL2006, for VDD (VCORE) power supply margining. 0X: MGN is tri-state; no margining is active. 10: MGN is driven low; VCORE is reduced 5%. 11: MGN is driven high; VCORE is increased 5%.
2	NGI2C_ACC	Controls the whether the system may access the private I2C devices owned by the **ngPIXIS** device. 0: The system cannot access those devices. 1: The system may access those devices.
3	DEVID7	Assigns the MSB of the Serial RapidIO Device ID (bits 5-6 are set to ‘1’ always). NOTE: As this bit is samples upon reset, it can only be changed with EEPROM initialization is used. See Section 6.2.2 for details.
4	PJWP_B	Controls the whether the PromJET may be written to or not. 0: The PromJet cannot be written to. 1: The PromJet may be written to.
5	REFCLK	Controls the reference (SERDES) clock frequency. 0: SERDES ports are supplied with a 100 MHz clock. 1: SERDES ports are supplied with a 125 MHz clock.
6	USB_ID	If the USB port is adapted to On-The-Go (OTG) form-factor, this bit controls the state of the USB ID pin.
7	SD8X	0: **P2020** SPI_CS(0:3)_B pins are used as SDHC data bits 4:7 for SDHC-8bit mode. SPI CS_B pins are pulled high. 1: **P2020** SPI_CS(0:3)_B pins are used with the SPI controller. SDHC data bits 4:7 are pulled high and only SDHC-4bit mode is used.

9.3.9 DMA Control/Status Register (PX_DMA)

The PX_DMA register provides direct, bit-level control to the external DMA signals.

	0	1	2	3	4	5	6	7
R	*-rsv-*	*-rsv-*	*-rsv-*	*-rsv-*	*-rsv-*	DDONE	DACK	DREQ
W								
Reset	0	0	0	0	0	1	1	0
Offset	0x09							

Figure 41. DMA Control/Status Register (PX_DMA)

Table 60. PX_DMA Field Descriptions

Bits	Name	Description
0-4	*-rsvd-*	-reserved-
5	DONE	Value is the current status of the DMA1_DDONE_B signal.
6	DACK	Value is the current status of the DMA1_DACK_B signal.
7	DREQ	Value is driven on the DMA1_DREQ_B signal.

9.3.10 Address Register (PX_ADDR)

The PX_ADDR register is a general-purpose read/write register which is used to index an internal 256-byte SRAM array. It reset upon initial power activation, or by chassis reset sources; PX_ADDR preserves its value between COP- or watchdog-initiated resets.

	0	1	2	3	4	5	6	7
R	ADDR							
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x0A							

Figure 42. SRAM Address Register (PX_ADDR)

Table 61. PX_ADDR Field Descriptions

Bits	Name	Description
0-7	ADDR	Address of SRAM array to which PX_DATA will read/write.

NOTE: Writing PX_ADDR and reading/write PX_DATA is non-atomic. Care should be exercised when sharing SRAM between processors and/or the **ngPIXIS** GMSA core.

9.3.11 Data Register (PX_DATA)

The PX_DATA register is a general-purpose read/write register which is used to read or write to an internal 256-byte SRAM array. It reset upon initial power activation, or by chassis reset sources; PX_DATA preserves its value between COP- or watchdog-initiated resets.

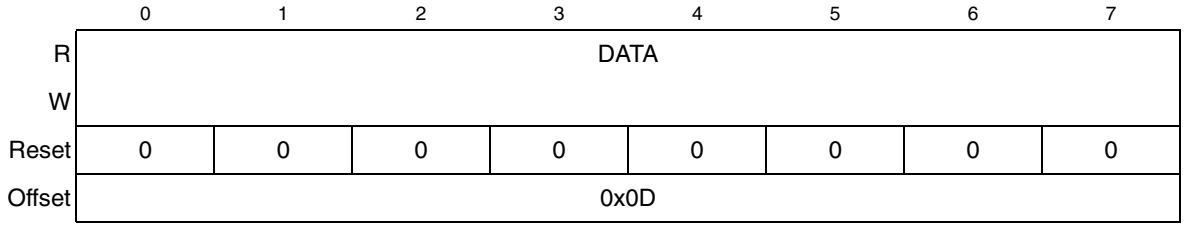

Figure 43. Power Status Register (PX_DATA)

Table 62. PX_DATA Field Descriptions

Bits	Name	Description
0-7	DATA	Contents of SRAM array as indexed by PX_ADDR.

9.3.12 LED Data Register (PX_LED)

The LED data register can be used to directly control the monitoring LEDs (for software message purposes, as an example). Direct control of the LEDs is possible only when PX_CSR[LED] is set to one.

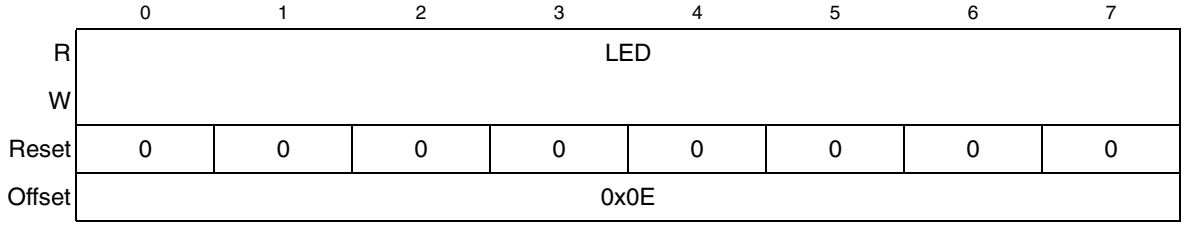

Figure 44. LED Control Register (PX_LED)

Table 63. FS_LED Field Descriptions

Bits	Name	Description
0-7	LED	Corresponding values for monitoring LEDs L0-L7. Setting a bit to one illuminates the LED.

9.3.13 VELA Control Register (PX_VCTL)

The PX_VCTL register may be used to start and control the configuration reset sequencer as well as other configuration/test-related features..

	0	1	2	3	4	5	6	7
R	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	WDEN	*-rsvd-*	PWROFF	GO
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x10							

Figure 45. Configuration Sequencer Control Register (PX_VCTL)

Table 64. PX_VCTL Field Descriptions

Bits	Name	Description
0-3	*-rsvd-*	*-reserved-*
4	WDEN	Watchdog Enable 0: Watchdog disabled. 1: Watchdog enabled. If not disabled with 2^29 clock cycles (> 5 minutes at 30ns clock), the system will be reset. NOTE: This is not a highly-secure watchdog; software can reset this bit at any time, disabling the watchdog.
5	*-rsvd-*	*-reserved-*
6	PWROFF	Power Off 0: Power is controlled as normal (by NVidia or by switch). 1: Power is forced off. NOTE: Hardware must restore power; software cannot force power on.
7	GO	Go 0: The VELA sequencer remains idle. 1: The VELA sequencer starts. NOTE: The sequencer halts after running until software resets GO to 0.

Note that the default value of PWROFF is zero, so that normal operations do not interfere with the power switches. Setting PWROFF to one overrides any user- or APM-initiated power switch event.

9.3.14 VELA Status Register (PX_VSTAT)

The PX_VSTAT register may be used to monitor the configuration sequencer activity.

	0	1	2	3	4	5	6	7
R	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	BUSY
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x11							

Figure 46. Configuration Sequencer Status Register (PX_VSTAT)

Table 65. PX_VSTAT Field Descriptions

Bits	Name	Description
0-6	*-rsvd-*	*-reserved-*
7	BUSY	0: The VELA sequencer is idle. 1: The VELA sequencer is busy.

9.3.15 VELA Status Register (PX_VSTAT)

The PX_VSTAT register may be used to monitor the configuration sequencer activity.

	0	1	2	3	4	5	6	7
R	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	BUSY
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x11							

Figure 47. Configuration Sequencer Status Register (PX_VSTAT)

Table 66. PX_VSTAT Field Descriptions

Bits	Name	Description
0-6	*-rsvd-*	*-reserved-*
7	BUSY	0: The VELA sequencer is idle. 1: The VELA sequencer is busy.

9.3.16 VELA Configuration Enable Register 0 (PX_VCFGEN0)

The PX_VCFGEN0 registers are used to enable special events which cannot be controlled with the more general SWx/ENx registers.

	0	1	2	3	4	5	6	7
R	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	*-rsvd-*	DDRCLK	SYSCLK
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x12							

Figure 48. VELA Configuration Enable 0 Register (PX_VCFGEN0)

Table 67. PX_VCFGEN0 Field Descriptions

Bits	Name	Description
0-7	*-rsvd-*	*-reserved-*
6	DDRCLK	If set, the values in PX_DCLK[0:2] are used to select a new DDRCLK frequency when VCTL[GO] is asserted.
7	SYSCLK	If set, the values in PX_SCLK[0:2] are used to select a new SYSCLK frequency when VCTL[GO] is asserted.

9.3.17 OCM Control/Status Register (PX_OCMCSR)

The PX_OCMCSR is a general-purpose read/write register, used to communicate between the **P2020** and the FPGA GMSA processor.

	0	1	2	3	4	5	6	7
R	ACK	U0	DBGSEL				U1	MSG
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x14							

Figure 49. Configuration Sequencer Status Register (PX_OCMCSR)

Table 68. PX_OCMCSR Field Descriptions

Bits	Name	Description
0	ACK	0: No acknowledgement is present. 1: OCM software signals that message has been processed.
1	U0	Unassigned values.
2-5	DBGSEL	Selects information to provide in the PX_GMDBG register.
6	U1	Unassigned values.
7	MSG	0: No message is present. 1: Signal OCM software that a message is present.

Note: The above definitions are enforced only by the OCM software; it is possible to redefine the register meanings by using different software with the GMSA processor.

9.3.18 OCM Message Register (PX_OCMMSG)

The PX_OCMMSG is a general-purpose read/write register, used to communicate between the **P2020** and the FPGA GMSA processor.

	0	1	2	3	4	5	6	7
R	MSGADDR							
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x15							

Figure 50. Configuration Sequencer Status Register (PX_OCMMSG)

Table 69. PX_OCMMSG Field Descriptions

Bits	Name	Description
0-7	ADDR	Address within shared SRAM of a message to process.

Note: The above definitions are enforced by software; it is possible to redefine the meanings by using different software in the GMSA processor.

9.3.19 GMSA Debug Register (PX_GMDBG)

The PX_GMDBG register allows access to information regarding the GMSA processor in the **ngPIXIS**.

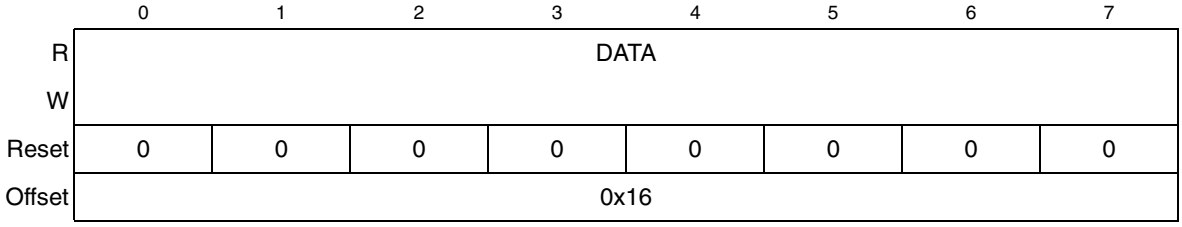

Figure 51. Configuration Sequencer Status Register (PX_GMDBG)

Table 70. PX_GMDBG Field Descriptions

Bits	Name	Description
0-7	DATA	Requested data, as selected by PX_OCMCSR[2:5]: 0001: PC[7:0] 0010: PC[11:0] 0011: opcode 0100: status 0101: TOS Other values are undefined.

9.3.20 SCLK[0:2] Registers (PX_SCLK[0:2])

The PX_SCLK[0:2] registers control the 24-bit configuration word of the ICS307 system clock generator.

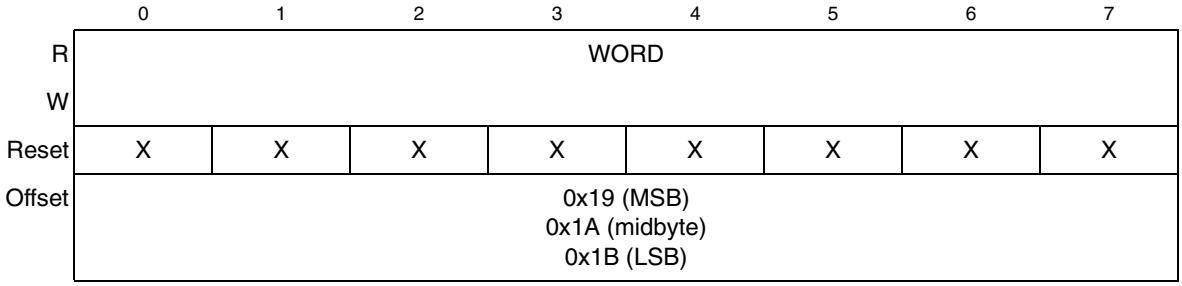

Figure 52. SCLK[0:2] Register (PX_SCLK[0:2])

Table 71. PX_SCLK[0:2] Field Descriptions

Bits	Name	Description
0-7	WORD	Read: returns the current programmed values. Write: values written to WORD are driven into the ICS307 during reset sequencing if PX_VCFGEN0[SCLK]=1; otherwise, the encoded value of CFG_SYSCLK(0:2) is used.

9.3.21 DCLK[0:2] Registers (PX_DCLK[0:2])

The PX_DCLK[0:2] registers control the 24-bit configuration word of the ICS307 DDR clock generator.

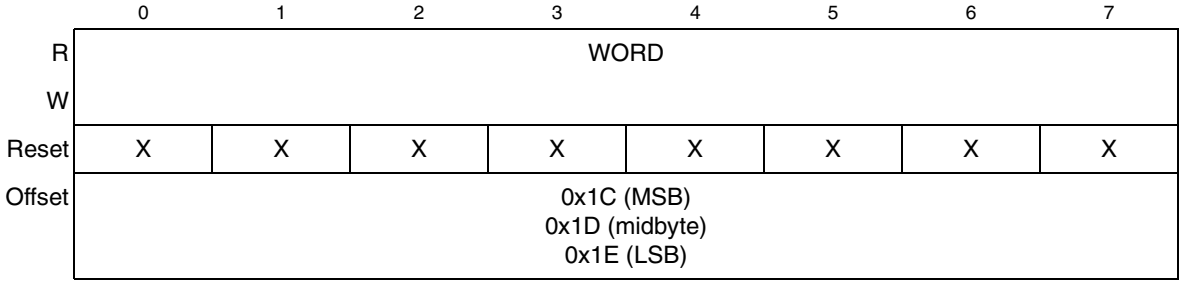

Figure 53. DCLK[0:2] Register (PX_DCLK[0:2])

Table 72. PX_DCLK[0:2] Field Descriptions

Bits	Name	Description
0-7	WORD	Read: returns the current programmed values. Write: values written to WORD are driven into the ICS307 during reset sequencing if PX_VCFGEN0[SCLK]=1; otherwise, the encoded value of CFG_DDRCLK(0:2) is used.

9.3.22 Watchdog Register (PX_WATCH)

The PX_WATCH register selects the appropriate watchdog timer event for the VELA-controlled sequencer. Note that this watchdog works independantly of any other watchdog timers, such as those within the **P2020**.

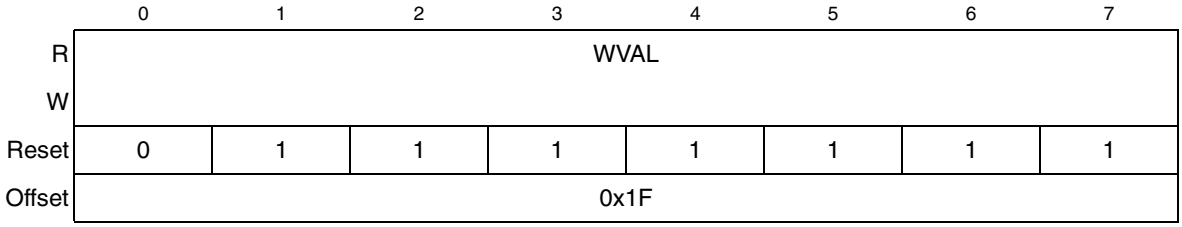

Figure 54. Watchdog Register (PX_WATCH)

Table 73. PX_WATCH Field Descriptions

Bits	Name	Description
0-7	WVAL	Read: returns the current programmed values. Write: sets watchdog timer.

The PX_WATCH reg represent the 8 most significant bits of an internal 34-bit watchdog timer. Any new value must be written BEFORE the PX_VCTL[WDEN] bit is set to one, and must be written after every reset of this register (i.e. it resets just like any other general register).

If the watchdog times out or any other reset/restart condition occurs (except the PX_VCTL[GO] bit), then repeat above.

Time formulae:

The base of the timer = 26 bits X 30ns interval = 2.01326592 seconds.

Where the upper 8 bit field represents (seconds) [(decimal value of the 8 bit field) x (2.01326592sec)] + 2.01326592sec.

Some examples values for PX_WATCH register values:

Table 74. Watchdog Timer Values

Timeout Value		Timeout
Binary	Hex	
11111111	0xFF	0.59 min
01111111	0x7F	4.29 min
00111111	0x3F	2.15min
00011111	0x1F	1.07 min
00001111	0x0F	32.1 sec
00000111	0x07	16.1 sec
00000011	0x03	8.05 sec
00000001	0x01	4.027 sec
00000000	0x00	2.013 sec

9.3.23 Switch Register (PX_SWx)

The PX_SWx registers are used to define configuration switch overrides. Each bit in each SWx register corresponds to a similarly named switch on the board. If a bit is ‘1’, the switch will be ‘1’ and vice-versa, provided the matching ENx bit is set..

	0	1	2	3	4	5	6	7
R	SWx #1	SWx #2	SWx #3	SWx #4	SWx #5	SWx #6	SWx #7	SWx #8
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x20, 0x22, 0x24, ...							

Figure 55. Switch Register (PX_SW(1:8))

Table 75. PX_SW (1:8) Field Descriptions

Bits	Name	Description
0-7	SWx #b	Value to replace for switch SWx #b.

Note that SW registers do NOT reflect the contents of the physical switches.

9.3.24 Switch Enable Register (PX_ENx)

The PX_ENx registers control whether the corresponding bits of corresponding PX_SWx register are used, or ignored.

	0	1	2	3	4	5	6	7
R	ENx #1	ENx #2	ENx #3	ENx #4	ENx #5	ENx #6	ENx #7	ENx #8
W								
Reset	0	0	0	0	0	0	0	0
Offset	0x21, 0x23, 0x25, ...							

Figure 56. Switch Enable Register (PX_EN(1:8))

Table 76. PX_EN(1:8) Field Descriptions

Bits	Name	Description
0-7	ENx #b	0: External switch SWx #b controls the corresponding configuration pin; the system controller does not affect the value. 1: Internal register SWx #b controls the corresponding configuration pin; the external switches do not affect the value.

9.4 Header File

```
// STINGRAY include file
#ifndef STINGRAY_H
#define STINGRAY_H
#define ngPIXIS_BASE    0xFD000000

// Registers
#define PX_ID           (ngPIXIS_BASE + 0x00)
#define PX_ARCH         (ngPIXIS_BASE + 0x01)
#define PV_SCVER        (ngPIXIS_BASE + 0x02)
#define PX_CSR          (ngPIXIS_BASE + 0x03)
#define PX_RST          (ngPIXIS_BASE + 0x04)
#define PX_AUX          (ngPIXIS_BASE + 0x06)
#define PX_SPD          (ngPIXIS_BASE + 0x07)
#define PX_BRDCFG0      (ngPIXIS_BASE + 0x08)
#define PX_DMA          (ngPIXIS_BASE + 0x09)
#define PX_ADDR         (ngPIXIS_BASE + 0x0A)
#define PX_DATA         (ngPIXIS_BASE + 0x0D)
#define PX_LED          (ngPIXIS_BASE + 0x0E)
#define PX_VCTL         (ngPIXIS_BASE + 0x10)
#define PX_VSTAT        (ngPIXIS_BASE + 0x11)
#define PX_VCFGEN0      (ngPIXIS_BASE + 0x12)
#define PX_OCMCSR       (ngPIXIS_BASE + 0x14)
#define PX_OCMMSG       (ngPIXIS_BASE + 0x15)
#define PX_OCMDBG       (ngPIXIS_BASE + 0x16)
#define PX_SCLK0        (ngPIXIS_BASE + 0x19)
#define PX_SCLK1        (ngPIXIS_BASE + 0x1A)
#define PX_SCLK2        (ngPIXIS_BASE + 0x1B)
#define PX_DCLK0        (ngPIXIS_BASE + 0x1C)
```

```
#define PX_DCLK1        (ngPIXIS_BASE + 0x1D)
#define PX_DCLK2        (ngPIXIS_BASE + 0x1E)
#define PX_WATCH        (ngPIXIS_BASE + 0x1F)
#define PX_SW(n)        (ngPIXIS_BASE + 0x20 + ((n)*2))
#define PX_EN(n)        (ngPIXIS_BASE + 0x21 + ((n)*2))

// Define functions to read and write byte registers.
// Could be replaced by asm or other functions.

#define PXR(r)          read_byte(r)
#define PXW(r,b)        write_byte(r,w)

// Register bit read/write controls
#define PX_GET_ID               ( PXR(PX_ID)       & 0xFF)
#define PX_GET_ARCH             ( PXR(PX_ARCH)     & 0xFF)


#endif /* STINGRAY_H */
```

9.5 EEPROM Data

The SystemID EEPROM stores important data about the Stingray system, including:

- Board ID
- Errata Level (as shipped)
- Manufacturing Date
- Ethernet MAC address

This device is programmed at the factory and is write-protected. The contents of the data are described in detail in application note “AN3638 : The SystemID Format for Power Architecture™ Development Systems” available at the freescale.com website.

Appendix A- References

Table 77. Useful References

Topic	Reference
SystemID Format	AN3638 from www.freescale.com
PromJet	“PromJet” modules are flash memory emulators available from Emutec (www.emutec.com). Stingray uses the 16-bit wide devices (size is user-dependant). The “low-voltage” option for use on the 3.3V local bus.

How to Reach Us:

USA/Europe/Locations Not Listed:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405,
Denver, Colorado 80217
1-480-768-2130
(800) 521-6274

Japan:
Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
852-26668334

Home Page:
www.freescale.com

Learn More: For more information about Freescale Semiconductor products, please visit www.freescale.com

DWP2020DS
Rev. 1.0 alpha
4/2009

freescale™
semiconductor