Vortex86EX2

Fact Sheet

32-bit x86 Micro Processor / Master Core

1. Overview

The **Vortex86EX2** is a low-power SoC. It integrates two fully static 32-bit X86 processor, Master System with the compatibility of Windows,Linux, DOS and most popular 32-bit RTOS; Slave System with also the compatibility of Windows, Linux , DOS and most popular 32-bit RTOS. Master System integrates I-16KB and D-16KB write through 4-way L1 cache, 128KB write through/write back 4-way L2 cache. Slave System integrates I-16KB and D-16KB write through 4-way L1 cache. It also integrates PCIe Gen1 x2 Port, DDR3, CrossBar Interface, ISA, I^{2}C, SPI, IPC (Internal Peripheral Controllers with DMA and interrupt timer/counter included), Fast Ethernet, FIFO UART, USB2.0 Host, USB2.0 Device, PCIe Device, SD and CAN controller within a single package to form a system-on-chip (SoC). It provides an ideal solution for low-cost and power-efficiency embedded system with desired performance.

2. Features

- ■ **X86 Processor Core**
    - – 6-stage pipeline
    - – Clock support up to 600 MHz
    - – X86 instruction set

- ■ **Floating Point Unit Support**
    - – Extends CPU instruction set to include Trigonometric, Logarithmic and Exponential

- ■ **Embedded I/D Separated L1 Cache**
    - – 16-KByte I-Cache, 16-KByte D-Cache

- ■ **Embedded L2 Cache**
    - – 4-way 128KB L2 Cache
    - – Write through or write back policy

- ■ **DDR3 Control Interface**
    - – 16 bits data bus
    - – 2 ranks
    - – Support ECC
    - – DDR3 clock support up to 400MHz
    - – DRAM size maximum 2GB

- ■ **LCD Interface**

- ■ **SD/eMMC Interface x3**

- – Support SDSC, SDHC and SDXC card.
- – Compliant upto eMMC Version 5.1

- ■ **Fast Ethernet MAC x2**

- ■ **Fast Ethernet PHY**

- ■ **PCIE Host/Target Gen 1 x1 Interface**

- ■ **PCIE Host Gen 1 x1 Interface**

- ■ **USB Interface**
    - – Port 0/1 support Host (HS, FS and LS)
    - – Port 1 support HS Device

- ■ **HDA Controller**
    - – 4 input streams, 4 output streams

- ■ **12bit ADC Interface x 2**
    - – 8 channels
    - – Support DMA Controller

- ■ **I^{2}C bus x2**
    - – Compliant with V2.1
    - – Some master code (general call, START and CBUS) not support.
    - – Up to the fast speed.

- **Boot SPI Interface**
  - For boot up function from SPI flash
  - Support flash size up to 128 MBytes
  - Support 4 IO mode

- **General Purpose SPI Controller x2**
  - Some master code (general call, START and CBUS) not support.
  - Support SPI Device x2 (SPI_CS# x2)
  - Support DMA Controller

- **CAN Bus Interface x2**
  - Compatible with CAN 2.0A/2.0B

- **Motion Control Interface x3**
  - 1groups of controller, 4 controllers per group
  - Each controller can be configured to PWM/Servo/Sensor Interface mode
  - Controller interconnect to the other with routing network in the same group

- **CrossBar Interface**
  - 16 CrossBar ports for digital function select. (each port is 8 pins, total 128 pins)

- **ISA Bus Interface**
  - AT clock programmable
  - 8/16 Bit ISA device w/ Zero-Wait-State
  - Generate refresh signals to ISA interface during DRAM refresh cycle
  - Support Max ISA Clock 33M
  - Support 7 channel ISA DMA
  - Support ISA IRQ x 11

- **DMA Controller x2**

- **Programmable Interrupt Controller**

- **MTBF Counter**

- **Counter / Timers**

  - 1 set of 8254 timer controller

- **Real Time Clock**
  - Less than 2.5uA (3.0V) power consumption while chip is power-off.

- **FIFO UART Port x 10 (10 sets COM Port)**
  - Compatible with 16C550 / 16C552
  - Default internal pull-up
  - Support programmable baud rate generator with data rate from 50 to 20M bps
  - The character options are programmable for 1 start bits; 1, 1.5 or 2 stop bits; even, odd or no parity; 5~8 data bits
  - Support TXD_En Signal on COM1-10
  - Port 80h output data could be sent to COM1 by software programming
  - Support half-duplex mode
  - Enhanced low I/O access latency

- **Parallel Port**
  - Support SPP/EPP/ECP mode

- **General Programmable I/O**
  - Support 128 programmable I / O pins
  - Each GPIO pin can be individually configured to be an input/output pin
  - GPIO_P0~GPIO_P9 can be programed by 8051A
  - Support GPIO Interrupt Controller (input/output) x2
  - Support DMA Controller

- **PS / 2 Keyboard and Mouse Interface Support**
  - Compatible with 8042 controller

- **Temperature Sensor**

- **Speaker out**

- **JTAG Interface supported for Software Debugging**

- **Input Clock**
  - 25 MHz
  - 32.768 KHz

- **Output Clock**
  - one clock output select from 14.318MHz /24MHz /25MHz /50MHz /ISA Clock

- **Operating Voltage Range**
  - Core voltage: 1.2 V ± 5%
  - I/O voltage: 1.5V (or 1.35V) ± 5%, 1.8V ± 5 %, 3.3 V ± 10 %

- **Operating Temperature**
  - -40°C ~ 85°C

- **Package Type**
  - 19x19mm LFBGA-441

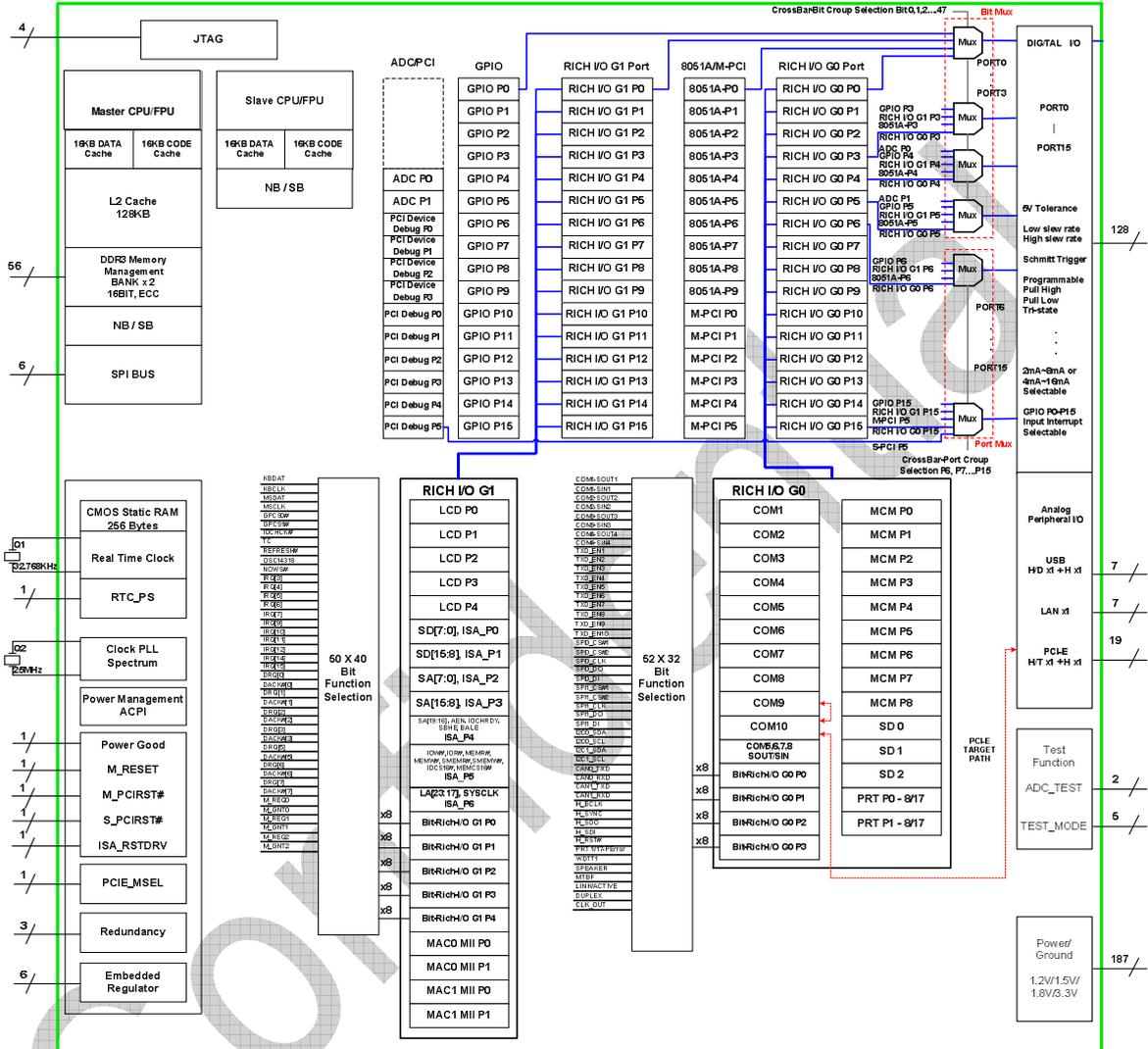# 3. Block Diagram & PCI Device List

3.1. VORTEX86EX2 Block Diagram

3.2. Clock Domain Diagram

3.3. PCI Device List

ID SEL	ADD 11	ADD 12	ADD 13	ADD 14	ADD 15	ADD 16	ADD 17	ADD 18	ADD 19	ADD 20	ADD 21	ADD 22	ADD 23	ADD 24	ADD 25	ADD 26	ADD 27	ADD 28	ADD 29	ADD 30
Device #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Function	NB	PCI slot 0	PCI slot 1	PCI slot 2	PCIE1 (Host)	PCIE0 (Host/Target)	Hybrid	SB	MAC0	MAC1	USB 2.0	USB 2.0 Device	SD	SD	HDA	MCM 0	MCM 1	MCM 2	CAN	LCD
Fun 0	NB0							SB0			OHCI		SD 0/1	SD 2					CAN0	
Fun 1	NB1							SB1			EHCI								CAN1	

Note: 1. USB 2.0 Host supports 2 ports. USB 2.0 Device supports 1 ports(PHY share with USB 2.0 Host Port 1. Select by flash strap).

2. PCIE Host, Interrupt Routing: INTD, INTA, INTB, INTC

3. PCIE Host/Target, Interrupt Routing: INTA, INTB, INTC, INTD

4. Terminology Explanation

4.1. Input & Output and Register AttributeTerms

Term	Term Description
I	Input Pin.
O	Output Pin.
I/O	Bi-directional Input/Output Pin.
P	Power Pin.
G	Ground Pin.
RO	Read Only.
WO	Write Only.
R/W	Read and Write.
R/W1C	Read and Write Clear. A register bit with this attribute can be read and written. However, a write of 1 clears the corresponding bit and a write of 0 will have no effect.
W1C	A write of 1 clears this bit.
RC	Read Clear.

4.2. General Terms

Term	Term Description
EIP	The instruction pointer is called EIP in 32-bit mode.

5. PIN Function List

5.1. Ball Map

	D	C	B	A	
	CBAR_P5[6]	CBAR_P5[7]	RTC_VDD33	RTC_VSS	1
	CBAR_P5[5]	RTC_PS	RTC_XOUT	RTC_XIN	2
	CBAR_P5[4]	SPI_HOLD	SPI_WP	SPI_DO	3
	VSS	SPI_CS#	SPI_CK	SPI_DI	4
	TDO	TCK	VSS	VDD33	5
	S_PCIRST#	PE0_RST	M_PCIRST#	ISA_RSTDRV	6
	CBAR_P15[4]	CBAR_P15[5]	CBAR_P15[6]	CBAR_P15[7]	7
	VSS	CBAR_P15[0]	CBAR_P15[1]	CBAR_P15[2]	8
	CBAR_P14[4]	CBAR_P14[5]	CBAR_P14[6]	CBAR_P14[7]	9
	CBAR_P14[1]	CBAR_P14[2]	VSS	VDD33	10
	CBAR_P13[4]	CBAR_P13[5]	CBAR_P13[6]	CBAR_P13[7]	11
	VSS	CBAR_P13[0]	CBAR_P13[1]	CBAR_P13[2]	12
	CBAR_P12[4]	CBAR_P12[5]	CBAR_P12[6]	CBAR_P12[7]	13
	CBAR_P12[1]	CBAR_P12[2]	VSS	VDD33	14
	CBAR_P11[4]	CBAR_P11[5]	CBAR_P11[6]	CBAR_P11[7]	15
	VSS	CBAR_P11[0]	CBAR_P11[1]	CBAR_P11[2]	16
	CBAR_P10[4]	CBAR_P10[5]	CBAR_P10[6]	CBAR_P10[7]	17
	CBAR_P10[1]	CBAR_P10[2]	VSS	VDD33	18
	CBAR_P9[4]	CBAR_P9[5]	CBAR_P9[6]	CBAR_P9[7]	19
	CBAR_P9[0]	CBAR_P9[1]	CBAR_P9[2]	CBAR_P9[3]	20
	CBAR_P7[6]	CBAR_P8[1]	CBAR_P8[2]	CBAR_P8[5]	21
	CBAR_P7[7]	VSS	CBAR_P8[3]	CBAR_P8[6]	22
	CBAR_P8[0]	VDD33	CBAR_P8[4]	CBAR_P8[7]	23

J	H	G	F	E	
USB_DP	CBAR_P4[3]	ADC_TEST0	ADC_TEST1	CBAR_P5[3]	1
USB_DM	CBAR_P4[4]	CBAR_P4[1]	CBAR_P5[1]	CBAR_P5[2]	2
CBAR_P4[6]	CBAR_P4[5]	CBAR_P4[2]	CBAR_P4[0]	CBAR_P5[0]	3
CBAR_P4[7]	AVDD_ADC033	AVDD_ADC0REF33	TMS	TDI	4
AVSS_ADC033	AVSS_ADC0REF33	AVSS_TEMP33	AVDD_TEMP33	PCIE_MSEL	5
				PWRGOOD	6
				CBAR_P15[3]	7
AVDD_ADC1REF33	AVDD_ADC133			VDD33	8
AVSS_ADC1REF33	AVSS_ADC133			CBAR_P14[3]	9
SYSFAIL	ExtSysFailIn			CBAR_P14[0]	10
M_RESET	Ext_Switch_Fail			CBAR_P13[3]	11
VSS	VSS			VDD33	12
VSS	VDD12			CBAR_P12[3]	13
VSS	VSS			CBAR_P12[0]	14
USBD_VBUS	TEST_MODE4			CBAR_P11[3]	15
TEST_MODE1	TEST_MODE0			VDD33	16
				CBAR_P10[3]	17
				CBAR_P10[0]	18
CBAR_P3[3]	VDD33	CBAR_P6[1]	CBAR_P6[6]	VDD33	19
CBAR_P3[4]	VSS	CBAR_P6[2]	CBAR_P6[7]	VSS	20
CBAR_P3[5]	CBAR_P3[6]	CBAR_P6[3]	CBAR_P7[0]	CBAR_P7[3]	21
VSS	CBAR_P3[7]	CBAR_P6[4]	CBAR_P7[1]	CBAR_P7[4]	22
VDD33	CBAR_P6[0]	CBAR_P6[5]	CBAR_P7[2]	CBAR_P7[5]	23

P	N	M	L	K	
PE1_RXP	PE1_CLKN	AVDD_USB12	USB1_DP	AVDD_USB33	1
PE1_RXN	PE1_CLKP	VSS	USB1_DM	AVSS_USB	2
AVDD_PETX12	VSS	VSS	AVSS_USBPLL12	AVDD_USBBAS33	3
AVSS_PETX12	VSS	VSS	AVDD_USBPLL12	AVSS_USBBAS33	4
AVDD_PE33	VSS	VSS	USB1_REXT	USB_REXT	5
					6
					7
VSS	VDD12	VSS	VDD12	VSS	8
VSS	VSS	VSS	VSS	VSS	9
VSS	VSS	VSS	VSS	VSS	10
VSS	VSS	VSS	VSS	VSS	11
VSS	VSS	VSS	VSS	VSS	12
VSS	VSS	VSS	VSS	VSS	13
VSS	VSS	VSS	VSS	VSS	14
VSS	VSS	VSS	VSS	TEST_MODE3	15
VSS	VDD12	VSS	VDD12	TEST_MODE2	16
					17
					18
CBAR_P0[6]	CBAR_P1[1]	VDD33	CBAR_P2[1]	CBAR_P2[6]	19
CBAR_P0[7]	CBAR_P1[2]	VSS	CBAR_P2[2]	CBAR_P2[7]	20
CBAR_P1[0]	CBAR_P1[3]	CBAR_P1[6]	CBAR_P2[3]	CBAR_P3[0]	21
VSS	CBAR_P1[4]	CBAR_P1[7]	CBAR_P2[4]	CBAR_P3[1]	22
VDD33	CBAR_P1[5]	CBAR_P2[0]	CBAR_P2[5]	CBAR_P3[2]	23

W	V	U	T	R	
AVSS_EPHYTX18	PE0_CLKP	PE0_RXN	PE0_TXN	PE1_TXP	1
AVDD_EPHYBG18	PE0_CLKN	PE0_RXP	PE0_TXP	PE1_TXN	2
AVSS_EPHYBG18	AVDD_EPHY33	VSS	AVDD_PEPLL12	AVDD_PERX12	3
VSS	VSS	VSS	AVSS_PEPLL12	AVSS_PERX12	4
AVSS_PEPLL	VSS	VDD18	VSS	VSS	5
AVDD_PEPLL12_1					6
AVSS_NBPLL12					7
VSS			VSS	VDD12	8
VSS			VDD12	VSS	9
DDR_DQ[8]			VSS	VSS	10
VSS			VDD12	VSS	11
DDR_VREF			VSS	VSS	12
DDR_ZQ			VDD12	VSS	13
VDD15			VSS	VSS	14
DDR_DQ[15]			VDD12	VSS	15
VSS			VSS	VDD12	16
DDR_ODT[1]					17
VSS					18
DDR_BA[2]	DDR_MA[2]	VDD15	CBAR_P0[0]	CBAR_P0[1]	19
DDR_MA[1]	VSS	DDR_MA[13]	VSS	CBAR_P0[2]	20
DDR_MA[6]	DDR_MA[8]	DDR_MA[14]	VDD15	CBAR_P0[3]	21
VDD15	DDR_MA[7]	DDR_MA[11]	VSS	CBAR_P0[4]	22
DDR_MA[5]	VSS	DDR_MA[9]	DDR_RST#	CBAR_P0[5]	23

AC	AB	AA	Y	
AVDD_EPHYPLL18	EPHY_ATSTP	EPHY_TXP	EPHY_RXP	1
AVSS_EPHYPLL18	EPHY_ATSTN	EPHY_TXN	EPHY_RXN	2
REG_AVDD33	REG_AVSS33	REG_VCTRL18	EPHY_ISET	3
XOUT_25	XIN_25	REG_FB12	REG_FB18	4
DIF2_PCIE_PLLCLK100_P	DIF2_PCIE_PLLCLK100_N	REG_VCTRL12	VSS	5
DIF1_PCIE_PLLCLK100_P	DIF1_PCIE_PLLCLK100_N	AVDD_PEPLL33	AVSS_PEPLL33	6
DIF0_PCIE_PLLCLK100_P	DIF0_PCIE_PLLCLK100_N	DIF_VSS12	AVDD_NBPLL12	7
AVDD_SBPLL18	AVSS_SBPLL	AVDD_SBPLL33	DIF_VDD12	8
VDD15	VSS	VSS	VDD15	9
DDR_DQ[2]	DDR_DQ[1]	DDR_DQ[0]	DDR_DQ[9]	10
DDR_DQ[3]	VSS	DDR_DQ[11]	DDR_DQ[10]	11
DDR_DQS[0]	VDD15	VSS	VDD15	12
DDR_DQS#[0]	DDR_DM[0]	DDR_DM[1]	VSS	13
VSS	DDR_DQ[5]	DDR_DQS#[1]	DDR_DQS[1]	14
DDR_DQ[4]	VDD15	VSS	DDR_DQ[14]	15
DDR_DQ[6]	DDR_DQ[7]	DDR_DQ[12]	VDD15	16
DDR_CLK0	VSS	DDR_DQ[13]	DDR_CLK1#	17
DDR_CLK0#	VDD15	DDR_RAS#	DDR_CLK1	18
VSS	DDR_ODT[0]	DDR_CAS#	VDD15	19
DDR_CKE	DDR_WE#	VSS	DDR_BA[1]	20
VDD15	DDR_MA[10]	DDR_CS1#	DDR_MA[4]	21
DDR_CS0#	DDR_MA[15]	DDR_MA[0]	DDR_MA[12]	22
VSS	DDR_BA[0]	VSS	DDR_MA[3]	23

5.2. **Pin Out Table**

Ball No.	Function	Ball No.	Function	Ball No.	Function	Ball No.	Function
A1	RTC_VSS	E20	VSS	M14	VSS	W8	VSS
A2	RTC_XIN	E21	CBAR_P7[3]	M15	VSS	W9	VSS
A3	SPI_DO	E22	CBAR_P7[4]	M16	VSS	W10	DDR_DQ[8]
A4	SPI_DI	E23	CBAR_P7[5]	M19	VDD33	W11	VSS
A5	VDD33	F1	ADC_TEST1	M20	VSS	W12	DDR_VREF
A6	ISA_RSTDRV	F2	CBAR_P5[1]	M21	CBAR_P1[6]	W13	DDR_ZQ
A7	CBAR_P15[7]	F3	CBAR_P4[0]	M22	CBAR_P1[7]	W14	VDD15
A8	CBAR_P15[2]	F4	TMS	M23	CBAR_P2[0]	W15	DDR_DQ[15]
A9	CBAR_P14[7]	F5	AVDD_TEMP33	N1	PE1_CLKN	W16	VSS
A10	VDD33	F19	CBAR_P6[6]	N2	PE1_CLKP	W17	DDR_ODT[1]
A11	CBAR_P13[7]	F20	CBAR_P6[7]	N3	VSS	W18	VSS
A12	CBAR_P13[2]	F21	CBAR_P7[0]	N4	VSS	W19	DDR_BA[2]
A13	CBAR_P12[7]	F22	CBAR_P7[1]	N5	VSS	W20	DDR_MA[1]
A14	VDD33	F23	CBAR_P7[2]	N8	VDD12	W21	DDR_MA[6]
A15	CBAR_P11[7]	G1	ADC_TEST0	N9	VSS	W22	VDD15
A16	CBAR_P11[2]	G2	CBAR_P4[1]	N10	VSS	W23	DDR_MA[5]
A17	CBAR_P10[7]	G3	CBAR_P4[2]	N11	VSS	Y1	EPHY_RXP
A18	VDD33	G4	AVDD_ADC0REF33	N12	VSS	Y2	EPHY_RXN
A19	CBAR_P9[7]	G5	AVSS_TEMP33	N13	VSS	Y3	EPHY_ISET
A20	CBAR_P9[3]	G19	CBAR_P6[1]	N14	VSS	Y4	REG_FB18
A21	CBAR_P8[5]	G20	CBAR_P6[2]	N15	VSS	Y5	VSS
A22	CBAR_P8[6]	G21	CBAR_P6[3]	N16	VDD12	Y6	AVSS_PEPLL33
A23	CBAR_P8[7]	G22	CBAR_P6[4]	N19	CBAR_P1[1]	Y7	AVDD_NBPLL12
B1	RTC_VDD33	G23	CBAR_P6[5]	N20	CBAR_P1[2]	Y8	DIF_VDD12
B2	RTC_XOUT	H1	CBAR_P4[3]	N21	CBAR_P1[3]	Y9	VDD15
B3	SPI_WP	H2	CBAR_P4[4]	N22	CBAR_P1[4]	Y10	DDR_DQ[9]
B4	SPI_CK	H3	CBAR_P4[5]	N23	CBAR_P1[5]	Y11	DDR_DQ[10]
B5	VSS	H4	AVDD_ADC033	P1	PE1_RXP	Y12	VDD15
B6	M_PCIRST#	H5	AVSS_ADC0REF33	P2	PE1_RXN	Y13	VSS

B7	CBAR_P15[6]	H8	AVDD_ADC133	P3	AVDD_PETX12	Y14	DDR_DQS[1]
B8	CBAR_P15[1]	H9	AVSS_ADC133	P4	AVSS_PETX12	Y15	DDR_DQ[14]
B9	CBAR_P14[6]	H10	EXTSYSFAILIN	P5	AVDD_PE33	Y16	VDD15
B10	VSS	H11	EXT_SWITCH_FAIL	P8	VSS	Y17	DDR_CLK1#
B11	CBAR_P13[6]	H12	VSS	P9	VSS	Y18	DDR_CLK1
B12	CBAR_P13[1]	H13	VDD12	P10	VSS	Y19	VDD15
B13	CBAR_P12[6]	H14	VSS	P11	VSS	Y20	DDR_BA[1]
B14	VSS	H15	TEST_MODE4	P12	VSS	Y21	DDR_MA[4]
B15	CBAR_P11[6]	H16	TEST_MODE0	P13	VSS	Y22	DDR_MA[12]
B16	CBAR_P11[1]	H19	VDD33	P14	VSS	Y23	DDR_MA[3]
B17	CBAR_P10[6]	H20	VSS	P15	VSS	AA1	EPHY_TXP
B18	VSS	H21	CBAR_P3[6]	P16	VSS	AA2	EPHY_TXN
B19	CBAR_P9[6]	H22	CBAR_P3[7]	P19	CBAR_P0[6]	AA3	REG_VCTRL18
B20	CBAR_P9[2]	H23	CBAR_P6[0]	P20	CBAR_P0[7]	AA4	REG_FB12
B21	CBAR_P8[2]	J1	USB_DP	P21	CBAR_P1[0]	AA5	REG_VCTRL12
B22	CBAR_P8[3]	J2	USB_DM	P22	VSS	AA6	AVDD_PEPLL33
B23	CBAR_P8[4]	J3	CBAR_P4[6]	P23	VDD33	AA7	DIF_VSS12
C1	CBAR_P5[7]	J4	CBAR_P4[7]	R1	PE1_TXP	AA8	AVDD_SBPLL33
C2	RTC_PS	J5	AVSS_ADC033	R2	PE1_TXN	AA9	VSS
C3	SPI_HOLD	J8	AVDD_ADC1REF33	R3	AVDD_PERX12	AA10	DDR_DQ[0]
C4	SPI_CS#	J9	AVSS_ADC1REF33	R4	AVSS_PERX12	AA11	DDR_DQ[11]
C5	TCK	J10	SYSFAIL	R5	VSS	AA12	VSS
C6	PE0_RST	J11	M_RESET	R8	VDD12	AA13	DDR_DM[1]
C7	CBAR_P15[5]	J12	VSS	R9	VSS	AA14	DDR_DQS#[1]
C8	CBAR_P15[0]	J13	VSS	R10	VSS	AA15	VSS
C9	CBAR_P14[5]	J14	VSS	R11	VSS	AA16	DDR_DQ[12]
C10	CBAR_P14[2]	J15	USBD_VBUS	R12	VSS	AA17	DDR_DQ[13]
C11	CBAR_P13[5]	J16	TEST_MODE1	R13	VSS	AA18	DDR_RAS#
C12	CBAR_P13[0]	J19	CBAR_P3[3]	R14	VSS	AA19	DDR_CAS#
C13	CBAR_P12[5]	J20	CBAR_P3[4]	R15	VSS	AA20	VSS
C14	CBAR_P12[2]	J21	CBAR_P3[5]	R16	VDD12	AA21	DDR_CS1#
C15	CBAR_P11[5]	J22	VSS	R19	CBAR_P0[1]	AA22	DDR_MA[0]

C16	CBAR_P11[0]	J23	VDD33	R20	CBAR_P0[2]	AA23	VSS
C17	CBAR_P10[5]	K1	AVDD_USB33	R21	CBAR_P0[3]	AB1	EPHY_ATSTP
C18	CBAR_P10[2]	K2	AVSS_USB	R22	CBAR_P0[4]	AB2	EPHY_ATSTN
C19	CBAR_P9[5]	K3	AVDD_USBBAS33	R23	CBAR_P0[5]	AB3	REG_AVSS33
C20	CBAR_P9[1]	K4	AVSS_USBBAS33	T1	PE0_TXN	AB4	XIN_25
C21	CBAR_P8[1]	K5	USB_REXT	T2	PE0_TXP	AB5	DIF2_PCIE_PLLCLK100_N
C22	VSS	K8	VSS	T3	AVDD_PEPLL12	AB6	DIF1_PCIE_PLLCLK100_N
C23	VDD33	K9	VSS	T4	AVSS_PEPLL12	AB7	DIF0_PCIE_PLLCLK100_N
D1	CBAR_P5[6]	K10	VSS	T5	VSS	AB8	AVSS_SBPLL
D2	CBAR_P5[5]	K11	VSS	T8	VSS	AB9	VSS
D3	CBAR_P5[4]	K12	VSS	T9	VDD12	AB10	DDR_DQ[1]
D4	VSS	K13	VSS	T10	VSS	AB11	VSS
D5	TDO	K14	VSS	T11	VDD12	AB12	VDD15
D6	S_PCIRST#	K15	TEST_MODE3	T12	VSS	AB13	DDR_DM[0]
D7	CBAR_P15[4]	K16	TEST_MODE2	T13	VDD12	AB14	DDR_DQ[5]
D8	VSS	K19	CBAR_P2[6]	T14	VSS	AB15	VDD15
D9	CBAR_P14[4]	K20	CBAR_P2[7]	T15	VDD12	AB16	DDR_DQ[7]
D10	CBAR_P14[1]	K21	CBAR_P3[0]	T16	VSS	AB17	VSS
D11	CBAR_P13[4]	K22	CBAR_P3[1]	T19	CBAR_P0[0]	AB18	VDD15
D12	VSS	K23	CBAR_P3[2]	T20	VSS	AB19	DDR_ODT[0]
D13	CBAR_P12[4]	L1	USB1_DP	T21	VDD15	AB20	DDR_WE#
D14	CBAR_P12[1]	L2	USB1_DM	T22	VSS	AB21	DDR_MA[10]
D15	CBAR_P11[4]	L3	AVSS_USBPLL12	T23	DDR_RST#	AB22	DDR_MA[15]
D16	VSS	L4	AVDD_USBPLL12	U1	PE0_RXN	AB23	DDR_BA[0]
D17	CBAR_P10[4]	L5	USB1_REXT	U2	PE0_RXP	AC1	AVDD_EPHYPLL18
D18	CBAR_P10[1]	L8	VDD12	U3	VSS	AC2	AVSS_EPHYPLL18
D19	CBAR_P9[4]	L9	VSS	U4	VSS	AC3	REG_AVDD33
D20	CBAR_P9[0]	L10	VSS	U5	VDD18	AC4	XOUT_25
D21	CBAR_P7[6]	L11	VSS	U19	VDD15	AC5	DIF2_PCIE_PLLCLK100_P

D22	CBAR_P7[7]	L12	VSS	U20	DDR_MA[13]	AC6	DIF1_PCIE_PLLCLK100_P
D23	CBAR_P8[0]	L13	VSS	U21	DDR_MA[14]	AC7	DIF0_PCIE_PLLCLK100_P
E1	CBAR_P5[3]	L14	VSS	U22	DDR_MA[11]	AC8	AVDD_SBPLL18
E2	CBAR_P5[2]	L15	VSS	U23	DDR_MA[9]	AC9	VDD15
E3	CBAR_P5[0]	L16	VDD12	V1	PE0_CLKP	AC10	DDR_DQ[2]
E4	TDI	L19	CBAR_P2[1]	V2	PE0_CLKN	AC11	DDR_DQ[3]
E5	PCIE_MSEL	L20	CBAR_P2[2]	V3	AVDD_EPHY33	AC12	DDR_DQS[0]
E6	PWRGOOD	L21	CBAR_P2[3]	V4	VSS	AC13	DDR_DQS#[0]
E7	CBAR_P15[3]	L22	CBAR_P2[4]	V5	VSS	AC14	VSS
E8	VDD33	L23	CBAR_P2[5]	V19	DDR_MA[2]	AC15	DDR_DQ[4]
E9	CBAR_P14[3]	M1	AVDD_USB12	V20	VSS	AC16	DDR_DQ[6]
E10	CBAR_P14[0]	M2	VSS	V21	DDR_MA[8]	AC17	DDR_CLK0
E11	CBAR_P13[3]	M3	VSS	V22	DDR_MA[7]	AC18	DDR_CLK0#
E12	VDD33	M4	VSS	V23	VSS	AC19	VSS
E13	CBAR_P12[3]	M5	VSS	W1	AVSS_EPHYTX18	AC20	DDR_CKE
E14	CBAR_P12[0]	M8	VSS	W2	AVDD_EPHYBG18	AC21	VDD15
E15	CBAR_P11[3]	M9	VSS	W3	AVSS_EPHYBG18	AC22	DDR_CS0#
E16	VDD33	M10	VSS	W4	VSS	AC23	VSS
E17	CBAR_P10[3]	M11	VSS	W5	AVSS_PEPLL		
E18	CBAR_P10[0]	M12	VSS	W6	AVDD_PEPLL12_1		
E19	VDD33	M13	VSS	W7	AVSS_NBPLL12		

5.3. Pin List Table

Function	Symbol	PIN Sum
SYSTEM	XOUT_25, XIN_25, PWRGOOD, M_RESET, M_PCIRST#, S_PCIRST#, ISA_RSTDRV, PCIE_MSEL	8 PINs
DDR3 Interface	DDR_RST#, DDR_CLK0, DDR_CLK0#, DDR_CLK1, DDR_CLK1#, DDR_RAS#, DDR_CAS#, DDR_WE#, DDR_CKE, DDR_CS1#, DDR_CS0#, DDR_DQM[1:0], DDR_DQS[1:0], DDR_DQS#[1:0], DDR_ODT[1], DDR_ODT[0], DDR_BA[2:0], DDR_DQ[15:0], DDR_MA[15:0], DDR_ZQ, DDR_VREF	56 PINs
CrossBar Interface	CBAR_P0[7:0], CBAR_P1[7:0], CBAR_P2[7:0], CBAR_P3[7:0], CBAR_P4[7:0], CBAR_P5[7:0], CBAR_P6[7:0], CBAR_P7[7:0], CBAR_P8[7:0], CBAR_P9[7:0], CBAR_P10[7:0], CBAR_P11[7:0], CBAR_P12[7:0], CBAR_P13[7:0], CBAR_P14[7:0], CBAR_P15[7:0]	128 PINs
USB Interface	USB_DP, USB_DM, USB1_DP, USB1_DM, USB_REXT, USB1_REXT	6 PINs
USB Device Interface	USBD_VBUS	1 PINs
PCIE Bus Interface	PE0_RST, PE0_TXP, PE0_TXN, PE0_RXP, PE0_RXN, PE0_CLKP, PE0_CLKN, PE1_TXP, PE1_TXN, PE1_RXP, PE1_RXN, PE1_CLKP, PE1_CLKN, DIF0_PCIE_PLLCLK100_P, DIF0_PCIE_PLLCLK100_N,DIF1_PCIE_PLLCLK100_P, DIF1_PCIE_PLLCLK100_N,DIF2_PCIE_PLLCLK100_P, DIF2_PCIE_PLLCLK100_N	19 PINs
Ethernet Interface	EPHY_ISET, EPHY_TXP, EPHY_TXN, EPHY_RXP, EPHY_RXN, EPHY_ATSTP, EPHY_ATSTN	7 PINs
SPI Interface	SPI_CS#/STRAP_BMS,SPI_CK/STRAP_JTAGDIS, SPI_DO/STRAP_HDM, SPI_DI, SPI_HOLD, SPI_WP	6 PINs
RTC Interface	RTC_PS, RTC_XOUT, RTC_XIN	3 PINs
JTAG Interface	TDO, TMS, TCK, TDI	4 PINs
ADC Interface	ADC_TEST0, ADC_TEST1	2 PINs
Embedded Regulator	REG_AVDD33, REG_AVSS33, REG_VCTRL18, REG_FB18, REG_VCTRL12, REG_FB12	6 PINs
Redundancy Interface	ExtSysFailIn, SysFail, Ext_Switch_Fail	3 PINs
TEST Interface	TEST_MODE0/STRAP_JTAGM, TEST_MODE1, TEST_MODE2,	5 PINs

Function	Symbol	PIN Sum
	TEST_MODE3, TEST_MODE4	
System PLL Power	AVDD_NBPLL12, AVSS_NBPLL12, AVDD_SBPLL18, AVDD_SBPLL33, AVSS_SBPLL	5 PINs
USB Interface Power	AVDD_USBPLL12, AVSS_USBPLL12, AVDD_USBBAS33, AVSS_USBBAS33, AVDD_USB33, AVSS_USB, AVDD_USB12	7 PINs
PCIE Interface Power	AVDD_PE33, AVDD_PEPLL12, AVSS_PEPLL12, AVDD_PERX12, AVSS_PERX12, AVDD_PETX12, AVSS_PETX12, AVDD_PEPLL33, AVSS_PEPLL33, AVDD_PEPLL12_1, AVSS_PEPLL	11 PINs
Ethernet Power	AVDD_EPHY33, AVDD_EPHYPLL18, AVSS_EPHYPLL18, AVDD_EPHYBG18, AVSS_EPHYBG18, AVSS_EPHYTX18	6 PINs
ADC Power	AVDD_ADC0REF33, AVSS_ADC0REF33, AVDD_ADC1REF33, AVSS_ADC1REF33, AVDD_ADC033, AVSS_ADC033, AVDD_ADC133, AVSS_ADC133	8 PINs
Temperature Sensor Power	AVDD_TEMP33, AVSS_TEMP33	2 PINs
Battery Power	RTC_VDD33, RTC_VSS	2 PINs
Differential PAD Power	DIF_VDD12, DIF_VSS12	2 PINs
1.2V Power	VDD12	11 PINs
1.5 Power	VDD15	13 PINs
1.8V Power	VDD18	1 PINs
3.3V Power	VDD33	13 PINs
Digital Ground	VSS	106 PINs

5.4. <u>Signal Description</u>

This chapter provides a detailed description of SoC signals. A signal with the symbol "#" at the end of itself indicates that this pin is low active. Otherwise, it is high active.

The following table gives knowledge of pins before you proceed with this chapter.

Pin	Description
Strap Pins (ex: STRAP_JTAGDIS, STRAP_BMS…)	When POWERGOOD is low, the Strap Pin value will be latched. The latched Strap Pin value will be kept after POWERGOOD goes high.

● **System (8 PINs)**

PIN No.	Symbol	Type	Description
AC4	XOUT_25	O	***Crystal-out***. Frequency output from the inverting amplifier (oscillator).
AB4	XIN_25	I	***Crystal-in.*** 25MHz frequency input, <u>within 100 ppm tolerance</u>, to the amplifier (oscillator).
E6	PWRGOOD	I	***Power-Good Input.*** This signal comes from Power Good of the power supply to indicate that the power is available. The SoC uses this signal to generate reset sequence for the system. In addition, the SoC is in reset state when Power Good is low.
J11	M_RESET	I	***Master System Reset Input.*** The SoC uses this signal to generate PCI reset sequence for the system. In addition, the SoC is in reset state when M_RESET is low.
B6	M_PCIRST#	O	***Master System PCI Reset.*** This pin is used to reset Master System PCI devices. When it is asserted low, all the PCI devices will be reset.
D6	S_PCIRST#	O	***Slave System PCI Reset.*** This pin is used to reset Slave System PCI devices. When it is asserted low, all the PCI devices will be reset.
A6	ISA_RSTDRV	O	***ISA Reset Driver***: This output signal is driven active during system power up
E5	PCIE_MSEL	I	***PCIe Host / Target Select.*** Strap pin for PCIe Interface is selected to Host or Target mode. Pull low to PCIe Target.

			Pull high to PCIe Host. (default internal pull-high)

● **DDR3 Interface (56 PINs)**

PIN No.	Symbol	Type	Description
T23	DDR_RST#	O	*Active Low Asynchronous Reset.* Reset is active when RESET# is LOW and otherwise. RESET# must be set as HIGH during normal operation.
AC17, Y18	DDR_CLK0 DDR_CLK1	O	*Clock output.* This pin provides the fundamental timing for the DRAM Controller.
AC18, Y17	DDR_CLK0# DDR_CLK1#	O	*Clock output.* This pin provides the fundamental timing for the DRAM controller.
AA18	DDR_RAS#	O	*Row Address Strobe.* When asserted, this signal latches row address on positive edge of the DRAM clock. This signal also allows row access and pre-charge.
AA19	DDR_CAS#	O	*Column Address Strobe.* When asserted, this signal latches column address on the positive edge of the DRAM clock. This signal also allows column access and pre-charge.
AB20	DDR_WE#	O	*Memory Write Enable.* This pin is used as a write enable for the memory data bus.
AC20	DDR_CKE	O	*Clock Enable.* CKE HIGH activates, and CKE LOW deactivates internal clock signals, and device input buffers and output drivers.
AA21, AC22	DDR_CS1# DDR_CS0#	O	*Chip Select CS0# & CS1#.* These two pins activate the DRAM devices. First Bank of DRAM accepts any command when the CS0# pin is active low. Second Bank of DRAM accepts any command when the CS1# pin is active low.
AA13,AB13	DDR_DQM[1:0]	O	*Data Mask DQM[1-0].* These pins act as synchronized output enables during read cycles and byte masks during write cycles.
Y14, AC12	DDR_DQS[1:0]	I/O	*Data Strobe DQS[1-0] for DRAM.* Output with write data, input with the read data for source synchronous operation.
AA14, AC13	DDR_DQS#[1:0]	I/O	*Data Strobe DQS#[1-0] for DRAM.* Output with write data, input with the read data for source synchronous operation.
AB19	DDR_ODT[0]	O	*On Die Termination Control for DRAM.* ODT(registered HIGH) enables on die termination resistance internal to the DDR3 SDRAM.

PIN No.	Symbol	Type	Description
W17	DDR_ODT[1]	O	***On Die Termination Control for DRAM.*** ODT(registered HIGH) enables on die termination resistance internal to the DDR3 SDRAM.
W19, Y20, AB23	DDR_BA[2:0]	O	***Bank Address BA[2-0].*** These pins are connected to DDR3 as bank address pins.
W15, Y15, AA17, AA16, AA11,Y11,Y10, W10, AB16, AC16, AB14, AC15, AC11, AC10, AB10, AA10	DDR_DQ[15:0]	I/O	***Memory Data DQ[15-0].*** These pins are connected to the DDR3 data bus.
AB22,U21,U20, Y22,U22,AB21, U23,V21,V22, W21,W23,Y21, Y23,V19,W20, AA22	DDR_MA[15:0]	O	***Memory Address MA[15-0].*** Normally, these pins are used as the row and column address for DDR3.
W13	DDR_ZQ	O	***Reference Voltage for DDR3 only. Reference Pin for ZQ calibration***
W12	DDR_VREF	I	***Reference voltage for DDR3 only. Reference voltage for inputs for SSTL interface.***

● **CrossBar Interface (128 PINs)**

PIN No.	Symbol	Type	Description
P20,P19,R23, R22,R21,R20, R19,T19	CBAR_P0[7:0]	I/O	***CrossBar Port 0[7:0].*** PIN function is select by CrossBar mechanism. This port support CrossBar-Bit select by group.
M22,M21,N23, N22,N21N20, N19,P21	CBAR_P1[7:0]	I/O	***CrossBar Port 1[7:0].*** PIN function is select by CrossBar mechanism. This port support CrossBar-Bit select by group.
K20,K19,L23, L22,L21,L20, L19,M23	CBAR_P2[7:0]	I/O	***CrossBar Port 2[7:0].*** PIN function is select by CrossBar mechanism. This port support CrossBar-Bit select by group.
H22,H21,J21,	CBAR_P3[7:0]	I/O	***CrossBar Port 3[7:0].***

PIN No.	Symbol	Type	Description
J20,J19,K23, K22,K21			PIN function is select by CrossBar mechanism. This port support CrossBar-Bit select by group.
J4, J3, H3, H2, H1, G3, G2, F3	CBAR_P4[7:0]	I/O	***CrossBar Port 4[7:0].*** PIN function is select by CrossBar mechanism. This port support CrossBar-Bit select by group. This port also support ADC, but only support CrossBar-Port select by group.
C1,D1,D2, D3,E1,E2, F2,E3	CBAR_P5[7:0]	I/O	***CrossBar Port 5[7:0].*** PIN function is select by CrossBar mechanism. This port support CrossBar-Bit select by group. This port also support ADC, but only support CrossBar-Port select by group.
F20,F19,G23, G22,G21,G20, G19,H23	CBAR_P6[7:0]	I/O	***CrossBar Port 6[7:0].*** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
D22,D21,E23, E22,E21,F23, F22,F21	CBAR_P7[7:0]	I/O	***CrossBar Port 7[7:0].*** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
A23,A22,A21, B23,B22,B21, C21,D23	CBAR_P8[7:0]	I/O	***CrossBar Port 8[7:0].*** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
A19,B19,C19, D19,A20,B20, C20,D20	CBAR_P9[7:0]	I/O	***CrossBar Port 9[7:0].*** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
A17,B17,C17, D17,E17,C18, D18,E18	CBAR_P10[7:0]	I/O	***CrossBar Port 10[7:0].*** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
A15,B15,C15, D15,E15,A16, B16,C16	CBAR_P11[7:0]	I/O	***CrossBar Port 11[7:0].*** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
A13,B13,C13, D13,E13,C14, D14,E14	CBAR_P12[7:0]	I/O	***CrossBar Port 12[7:0].*** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
A11,B11,C11, D11,E11,A12, B12,C12	CBAR_P13[7:0]	I/O	***CrossBar Port 13[7:0].*** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
A9,B9,C9,	CBAR_P14[7:0]	I/O	***CrossBar Port 14[7:0].***

PIN No.	Symbol	Type	Description
D9,E9,C10, D10,E10			PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.
A7,B7,C7, D7,E7,A8, B8,C8	CBAR_P15[7:0]	I/O	**_CrossBar Port 15[7:0]._** PIN function is select by CrossBar mechanism. This port only support CrossBar-Port select by group.

● **USB Interface (6 PINs)**

PIN No.	Symbol	Type	Description
J1, J2	USB_DP USB_DM	I/O	Universal Serial Bus Controller 0 Port 0. These are the serial data pair for USB Port 0. 15kΩ pull down resistors are connected to DP and DM internally.
L1, L2	USB1_DP USB1_DM	I/O	Universal Serial Bus Controller 0 Port 1. These are the serial data pair for USB Port 1. 15kΩ pull down resistors are connected to DP and DM internally.
K5	USB_REXT	I	Universal Serial Bus Controller 0 Port 0 External Reference Resistance 12kΩ ±1%
L5	USB1_REXT	I	Universal Serial Bus Controller 0 Port 1 External Reference Resistance 12kΩ ±1%

● **USB Device Interface (1 PINs)**

PIN No.	Symbol	Type	Description
J15	USBD_VBUS	I	**Universal Serial Bus Voltage Detect.** For detecting host connection.

● **PCIe Bus Interface (19 PINs)**

PIN No.	Symbol	Type	Description
C6	PE0_RST	I/O	**_PCIe Reset._** PCIe Host Mode: (Output) This pin is used to reset PCIe devices. When it is asserted low, PCIe devices will be reset. PCIe Target Mode: (Input) This pin is used to reset PCIe target Controller.
T2, T1	PE0_TXP	O	PCIe Differential serial data output. P: positive; N: negative

PIN No.	Symbol	Type	Description
	PE0_TXN		
U2, U1	PE0_RXP PE0_RXN	I	PCIe Differential serial data input. P: positive; N:negative
V1, V2	PE0_CLKP PE0_CLKN	I	PCIe Differential reference clock. P: positive; N: negative
R1, R2	PE1_TXP PE1_TXN	O	PCIe Differential serial data output. P: positive; N: negative
P1, P2	PE1_RXP PE1_RXN	I	PCIe Differential serial data input. P: positive; N: negative
N2, N1	PE1_CLKP PE1_CLKN	I	PCIe Differential reference clock. P: positive; N: negative
AC7, AB7	DIF0_PCIE_PLL CLK100_P DIF0_PCIE_PLL CLK100_N	O	PCIe Differential Clock 100MHz to PE1_CLKP and PE1_CLKN. P: positive; N: negative
AC6, AB6	DIF1_PCIE_PLL CLK100_P DIF1_PCIE_PLL CLK100_N	O	PCIe Differential Clock 100MHz to Port0 P: positive; N: negative
AC5, AB5	DIF2_PCIE_PLL CLK100_P DIF2_PCIE_PLL CLK100_N	O	PCIe Differential Clock 100MHz to Port1 P: positive; N: negative

● **Ethernet Interface (7 PINs)**

PIN No.	Symbol	Type	Description
Y3	EPHY_ISET	I	*ISET.* External resistor 6kΩ ±1%connecting pin for BIAS
AA1	EPHY_TXP	O	*TXP.* 10B-T/100BT transmitting output pin/ receiving input pin (positive)
AA2	EPHY_TXN	O	*TXN.* 10B-T/100BT transmitting output pin/ receiving input pin (negative)
Y1	EPHY_RXP	I	*RXP.* 10B-T/100BT receiving input pin/ transmitting output pin (positive)
Y2	EPHY_RXN	I	*RXN.* 10B-T/100BT receiving input pin/ transmitting output

PIN No.	Symbol	Type	Description
			pin (negative)
AB1	EPHY_ATSTP	I/O	**ATSTP.** Variable Gain Amplifier and ADC testing pin for input and output (positive)
AB2	EPHY_ATSTN	I/O	**ATSTN.** Variable Gain Amplifier and ADC testing pin for input and output (negative)

● **SPI Interface (6 PINs)**

Ball No.	Symbol	Type	Description
C4	SPI_CS#	O	**SPI Chip Select.**
	STRAP_BMS	I	**Boot Mode Select.** Pull it low to select Normal boot (250ms reset time). Pull it high to select Fast boot Default internal pull-high.
B4	SPI_CK	O	**SPI Clock.**
	STRAP_JTAGDIS	I	**JTAG Disable.** Pull it low to enable JTAG. Default internal pull-low
A3	SPI_DO	O	**SPI Data Output / Output pin,** connected with input of flash.
	STRAP_HDM	I	**Flash Strap Hardware Default Mode.** Pull it low to select Strap mode from Flash Pull it high to select Hardware Default Mode Default internal pull-low.
A4	SPI_DI	I	**SPI Data Input / Input pin, connected with output of flash.**
C3	SPI_HOLD	O	**SPI Hold**
B3	SPI_WP	O	**SPI Write Protection**

● **RTC Interface (3 PINs)**

PIN No.	Symbol	Type	Description
C2	RTC_PS	I	**RTC Battery Power Sense.**
B2	RTC_XOUT	O	**Crystal-out**. Frequency output from the inverting amplifier (oscillator)

PIN No.	Symbol	Type	Description
A2	RTC_XIN	I	**Crystal-in.** 32.768KHz frequency input, <u>within 100 ppm tolerance</u>, to the amplifier (oscillator).

- **JTAG Interface (4 PINs)**

PIN No.	Symbol	Type	Description
D5	TDO	O	**TDO:** JTAG Test Data Output pin.
F4	TMS	I	**TMS:** JTAG Test Mode Select pin.
C5	TCK	I	**TCK:** JTAG Test Clock Input pin.
E4	TDI	I	**TDI:** JTAG Test Data Input pin.

- **ADC Interface (2 PINs)**

PIN No.	Symbol	Type	Description
G1	ADC_TEST0	G	**ADC0 Test Pin.** Connect to AVSS_ADC0REF33
F1	ADC_TEST1	G	**ADC1 Test Pin.** Connect to AVSS_ADC1REF33

- **Embedded Regulator (6 PINs)**

PIN No.	Symbol	Type	Description
AC3	REG_AVDD33	P	**Analog Power**: Embedded Regulator 3.3V PAD Power.
AB3	REG_AVSS33	G	**Analog Ground** Embedded Regulator 3.3V PAD Ground
AA3	REG_VCTRL18	O	**Voltage Control for 1.8 Regulator**
Y4	REG_FB18	I	**Feedback from 1.8V Regulator**
AA5	REG_VCTRL12	O	**Voltage Control for 1.2 Regulator**
AA4	REG_FB12	I	**Feedback from 1.2V Regulator**

- **Redundant (3 PINs)**

PIN No.	Symbol	Type	Description
H10	ExtSysFailIn	I	**External system fail input.** This pin is the system fail in for redundant.

J10	SysFail	O	**System fail output.** This pin is the system fail out for re-dundant.
H11	Ext_Switch_Fail	I	**External System fail Switch Control input.** This pin is use to set to System Fail.

- **TEST Mode (5 PINs)**

PIN No.	Symbol	Type	Description
H16	TEST_MODE0	O	**Test Function Pin.**
	STRAP_JTAGM	I	**JTAG Mode Select.** Pull it low to select Old JTAG Mode. Pull it high to select New JTAG Mode. Default internal pull-high.
J16	TEST_MODE1	O	**Test Function Pin.**
K16	TEST_MODE2	I	**Test Function Pin.**
K15	TEST_MODE3	O	**Test Function Pin.** Add External pull-low resistor is recommended.
H15	TEST_MODE4	I	**Test Function Pin.** Add External tie-low is recommended.

- **System PLL Power & Ground (5 PINs)**

PIN No.	Symbol	Type	Description
Y7	AVDD_NBPLL12	P	**Analog Power.** CPU/NB PLL Analog Power 1.2V.
W7	AVSS_NBPLL12	G	**Analog Ground.** CPU/NB PLL Analog Ground
AC8	AVDD_SBPLL18	P	**Analog Power.** SB System PLL Analog Power 1.8V.
AA8	AVDD_SBPLL33	P	**Analog Power.** SB System PLL Analog Power 3.3V.
AB8	AVSS_SBPLL	G	**Analog Ground.** SB System PLL Analog Ground

- **USB Power & Ground (7 PINs)**

PIN No.	Symbol	Type	Description
L4	AVDD_USBPLL12	P	*Analog Power*: USB PLL Power
L3	AVSS_USBPLL12	G	*Analog Ground*: USB PLL Ground
K3	AVDD_USBBAS33	P	*Analog Power*: USB Base Voltage 3.3V Power
K4	AVSS_USBBAS33	G	*Analog Ground*: USB Base Voltage 3.3V Ground
K1	AVDD_USB33	P	*Analog Power*: USB 3.3V Power
M1	AVDD_USB12	P	*Analog Power*: USB 1.2V Power
K2	AVSS_USB	G	*Analog Ground*: USB Ground

● **PCIe Power & Ground (11 PINs)**

PIN No.	Symbol	Type	Description
P5	AVDD_PE33	P	*Analog Power* PCIe 3.3V Power
T3	AVDD_PEPLL12	P	*Analog Power* PCIe PLL 1.2V Power
T4	AVSS_PEPLL12	G	*Analog Ground* PCIe PLL 1.2V Ground
R3	AVDD_PERX12	P	*Analog Power* PCIe Receiver 1.2V Power
R4	AVSS_PERX12	G	*Analog Ground* PCIe Receiver 1.2V Ground
P3	AVDD_PETX12	P	*Analog Power* PCIe Transceiver 1.2V Power
P4	AVSS_PETX12	G	*Analog Ground* PCIe Transceiver 1.2V Ground
AA6	AVDD_PEPLL33	P	*Analog Power* PCIe PLL 3.3V Power

PIN No.	Symbol	Type	Description
Y6	AVSS_PEPLL33	G	*Analog Ground* PCIe PLL 3.3V Ground
W6	AVDD_PEPLL12_1	P	*Analog Power* PCIe 1.2V Power
W5	AVSS_PEPLL	G	*Analog Ground* PCIe PLL Ground

● **Ethernet Power & Ground (6 PINs)**

PIN No.	Symbol	Type	Description
V3	AVDD_EPHY33	P	*Analog Power* Internal Ethernet PHY 3.3V Power
AC1	AVDD_EPHYPLL18	P	*Analog Power* Internal Ethernet PHY PLL 1.8V Power
AC2	AVSS_EPHYPLL18	G	*Analog Ground* Internal Ethernet PHY PLL 1.8V Ground
W2	AVDD_EPHYBG18	P	*Analog Power* Internal Ethernet PHY Band Gap 1.8V Power
W3	AVSS_EPHYBG18	G	*Analog Ground* Internal Ethernet PHY Band Gap 1.8V Ground
W1	AVSS_EPHYTX18	G	*Analog Ground* Internal Ethernet PHY TX 1.8V Ground

● **ADC Power & Ground (8 PINs)**

PIN No.	Symbol	Type	Description
G4	AVDD_ADC0REF33	P	*Analog Power*: ADC Reference 3.3V Power
H5	AVSS_ADC0REF33	G	*Analog Ground*: ADC Reference 3.3V Ground
J8	AVDD_ADC1REF33	P	*Analog Power*: ADC Reference 3.3V Power
J9	AVSS_ADC1REF33	G	*Analog Ground*: ADC Reference 3.3V Ground

PIN No.	Symbol	Type	Description
H4	AVDD_ADC033	P	**Analog Power**: ADC 3.3V Power
J5	AVSS_ADC033	G	**Analog Ground**: ADC 3.3V Ground
H8	AVDD_ADC133	P	**Analog Power:** **ADC 3.3V Power**
H9	AVSS_ADC133	G	**Analog Ground:** **ADC 3.3V Ground**

● **Temperature Sensor Power & Ground (2 PINs)**

PIN No.	Symbol	Type	Description
F5	AVDD_TEMP33	P	**Analog Power**: Temperature Sensor 3.3V Power
G5	AVSS_TEMP33	G	**Analog Ground**: Temperature Sensor 3.3V Ground

● **Battery POWER & Ground (2 PINs)**

PIN No.	Symbol	Type	Description
B1	RTC_VDD33	P	**Battery power for RTC.**
A1	RTC_VSS	G	**Battery ground for RTC.**

● **Differential PAD Power & Ground (2 PINs)**

PIN No.	Symbol	Type	Description
Y8	DIF_VDD12	P	**Differential PAD 1.2V Power.**
AA7	DIF_VSS12	G	**Differential PAD 1.2V Ground.**

● **1.2V POWER (11 PINs)**

PIN No.	Symbol	Type	Description
H13,L8,L16, N8,N16,R8,	VDD12	P	**Core power.**

PIN No.	Symbol	Type	Description
R16,T9,T11, T13,T15			

- **1.5V POWER (13 PINs)**

PIN No.	Symbol	Type	Description
T21,U19,W14 ,W22,Y9,Y12, Y16,Y19, AB12,AB15, AB18,AC9, AC21	VDD15	P	*1.5V Power.*

- **1.8V POWER (1 PINs)**

PIN No.	Symbol	Type	Description
U5	VDD18	P	*1.8V Power.*

- **3.3V Power (13 PINs)**

PIN No.	Symbol	Type	Description
A5,A10,A14, A18,C23,E8, E12,E16,E19, H19,J23,M19, P23	VDD33	P	*I/O PAD Power.*

- **Digital Ground (106 PINs)**

PIN No.	Symbol	Type	Description
B5,B10,B14, B18,C22,D4, D8,D12,D16, E20,H12,H14, H20,J12, J13,J14,J22, K8-K14,L9-L15,	VSS	G	*Digital Ground.*

M2-M5,M8-M16, M20,N3-N5, N9-N15, P8-P16,P22,R5, R9-R15,T5,T8, T10,T12,T14, T16,T20,T22, U3,U4,V4,V5, V20,V23,W4, W8,W9,W11, W16,W18,Y5, Y13,AA9,AA12, AA15,AA20, AA23,AB9, AB11,AB17, AC14,AC19, AC23		

5.5. <u>PIN Capacitance Description</u>

Symbol	Parameter	Min.	Max.	Unit
C_{IN}	3.3V Input Capacitance (5V Tolerent I/O)	3	5	pF
C_{IN_ADC}	3.3V Input Capacitance (GPIO/ADC Comb PAD, ADC mode)	6	8	pF
C_{IN_Normal}	3.3V Input Capacitance (GPIO/ADC Comb PAD, GPIO mode)	1	3	pF

5.6. PIN Pull-up / Pull-down Description

PIN Name	Type (Voltage)	Driving Current	Pull-Up	Pull-Down	Schmitt Trigger	5V I/O Tolerant	Slew Rate	Description
XOUT_25M	O (1.2V)	--	--	--	--	--	--	
XIN_25M	I (1.2V)	--	--	--	--	--	--	
PWRGOOD	I (3.3V)	--	--	--	Y	Y	--	
M_RESET	I (3.3V)	--	--	--	Y	Y	--	
M_PCIRST#	O (3.3V)	Note 4	--	Y	--	Y	Note 4	PCI type I/O pad
S_PCIRST#	O (3.3V)	Note 4	--	Y	--	Y	Note 4	PCI type I/O pad
ISA_RSTDRV	O (3.3V)	Note 4	--	--	--	Y	Note 4	
PCIE_MSEL	I (3.3V)	--	--	--	--	Y	--	
DDR_RST#	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_CLK0	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_CLK0#	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_CLK1	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_CLK1#	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_RAS#	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_CAS#	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_WE#	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_CKE	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_CS0#	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_CS1#	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_DQM[1:0]	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_DQS[1:0]	I/O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_DQS#[1:0]	I/O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_ODT[1:0]	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_BA[2:0]	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_DQ[15:0]	I/O (1.5V)	Note 1			--	--	FIX	
DDR_MA[15:0]	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_ZQ	O (1.5V)	Note 1	--	--	--	--	FIX	
DDR_VREF	I (1.5V)							
CBAR_P0[7:0] ~	I/O (3.3V)	Note 3	Note 3	Note 3	Note 3	Y	Note 3	

PIN Name	Type (Voltage)	Driving Current	Pull-Up	Pull-Down	Schmitt Trigger	5V I/O Tolerant	Slew Rate	Description
CBAR_P3[7:0]								
CBAR_P4[7:0] ~ CBAR_P5[7:0]	I/O (3.3V)	Note 3	Note 3	Note 3	Note 3	--	Note 3	
CBAR_P6[7:0] ~ CBAR_P15[7:0]	I/O (3.3V)	Note 3	Note 3	Note 3	Note 3	Y	Note 3	
USB_DP	I/O	--	--	Note 2	--	--	--	USB analog I/O pad
USB_DM	I/O	--	--	Note 2	--	--	--	USB analog I/O pad
USB_DP1	I/O	--	--	Note 2	--	--	--	USB analog I/O pad
USB_DM1	I/O	--	--	Note 2	--	--	--	USB analog I/O pad
USB_REXT	I							
USB_REXT1	I							
USBD_VBUS	I (3.3V)					Y		
PE0_RST	I/O (3.3V)	Note 4	--	--	--	--	Note 4	
PE0_TXP, PE0_TXN	O							PCIE analog I/O pad
PE0_RXP, PE0_RXN	I							PCIE analog I/O pad
PE0_CLKP, PE0_CLKN	I							Differential Clock
PE1_TXP, PE1_TXN	O							PCIE analog I/O pad
PE1_RXP, PE1_RXN	I							PCIE analog I/O pad
PE1_CLKP, PE1_CLKN	I							Differential Clock
DIF0_PCIE_PLL CLK100_P, DIF0_PCIE_PLL CLK100_N	O							Differential Clock
DIF1_PCIE_PLL CLK100_P, DIF1_PCIE_PLL CLK100_N	O							Differential Clock

PIN Name	Type (Voltage)	Driving Current	Pull-Up	Pull-Down	Schmitt Trigger	5V I/O Tolerant	Slew Rate	Description
DIF2_PCIE_PLL CLK100_P, DIF2_PCIE_PLL CLK100_N	O							Differential Clock
EPHY_ISET	I	--	--	--	--	--	--	
EPHY_TXN	O	--	--	--	--	--	--	EtherPHY analog I/O pad
EPHY_TXP	O	--	--	--	--	--	--	EtherPHY analog I/O pad
EPHY_RXN	I	--	--	--	--	--	--	EtherPHY analog I/O pad
EPHY_RXP	I	--	--	--	--	--	--	EtherPHY analog I/O pad
EPHY_ATSTN	I/O	--	--	--	--	--	--	
EPHY_ATSTP	I/O	--	--	--	--	--	--	
SPI_CS#	O (3.3V)	8~10mA	Y	--	--	--	--	
SPI_CK	O (3.3V)	8~10mA	--	Y	--	--	S	
SPI_DO	I/O (3.3V)	8~10mA	--	Y	--	--	S	
SPI_DI	I/O (3.3V)		--	--	--	--	S	
SPI_HOLD	I/O (3.3V)					--		
SPI_WP	I/O (3.3V)					--		
RTC_PS	I (3.3V)	--	--	--	--	--	--	
RTC_XOUT	O (3.3V)	--	--	--	--	--	--	
RTC_XIN	I (3.3V)	--	--	--	--	--	--	
TDO	O (3.3V)	8mA	--	--	--	Y	S	
TMS	I (3.3V)	8mA	--	--	--	Y	S	
TCK	I (3.3V)	8mA	--	--	--	Y	S	
TDI	I (3.3V)	8mA	--	--	--	Y	S	
ADC_TEST0	G	--	--	--	--	--	--	
ADC_TEST1	G	--	--	--	--	--	--	
REG_AVDD33	P							

PIN Name	Type (Voltage)	Driving Current	Pull-Up	Pull-Down	Schmitt Trigger	5V I/O Tolerant	Slew Rate	Description
REG_AVSS33	G							
REG_VCTRL18	O (1.8V)							
REG_FB18	I (1.8V)							
REG_VCTRL12	O (1.2V)							
REG_FB12	I (1.2V)							
ExtSysFailIn	I (3.3V)							
SysFail	O (3.3V)							
Ext_Switch_Fail	I (3.3V)							
TEST_MODE0	O (3.3V)		Y					
TEST_MODE1	O (3.3V)			Y				
TEST_MODE2	I (3.3V)		Y					
TEST_MODE3	O (3.3V)			Y				
TEST_MODE4	I (3.3V)							

Definition:
1. --: Not need to specify
2. Y: Yes
3. F: Fast
4. S: Slow
5. P: Power
6. G: Ground
7. The pull-up/pull-down resistance is 75KΩ

Note:
1. Define by North Function1 84h.
2. The pull-down resistance is 15KΩ.
3. The setting is depending on CrossBar Setting BA(Base Address) +70h ~EFh. (CrossBar Base Address refers to the Register of 23h-20h, Hybrid Function Control Register)
4. The setting is depending on South Bridge Configuration Function 0 Register 48h.

6. Register Sets

The SoC contains four sets of software accessible registers (Core registers, MSR, I/O Mapped regis-

ters and Configuration registers).

6.1. Core Registers

The SoC provides 24 Core Registers. The 16 Base Architecture Registers (General-purpose Registers, Segment Registers, Flags Register and Instruction Pointer) are used in general system and application programming. The other 8 system-level registers (Control Registers and System Address Registers) can be used only by system-level programs. These registers are shown below. The details will be described in Register Description in Chapter 8.

6.1.1. General-Purpose Registers

Register Name
EAX Register
EBX Register
ECX Register
EDX Register
ESI Register
EDI Register
EBP Register
ESP Register

6.1.2. Segment Registers

Register Name
Code Segment Register – CS
Stack Segment Register – SS
Data Segment Register – DS
Data Segment Register – ES
Data Segment Register – FS
Data Segment Register – GS

6.1.3. Instruction Pointer Register

Register Name
Instruction Pointer Register

6.1.4. Flags Register

Register Name
Flags Register

6.1.5. Control Registers

Register Name
Control Register 0
Control Register 1
Control Register 2
Control Register 3

6.1.6. System Address Registers

Register Name
Global Descriptor Table Register
Interrupt Descriptor Table Register
Local Descriptor Table Register
Task State Segment Register

6.2. CPU MSR Registers

6.2.1. MSR Registers

MSR Index	MSR Name
10h	Time-Stamp Counter
1Bh	Confidential
174h	Confidential
175h	Confidential
176h	Confidential
277h	Confidential
52444300h	CPU_ID String Bit 63:0
52444301h	CPU_ID String Bit 95:64
52444303h	Confidential
52444304h	Confidential
52444305h	Confidential
52444306h	Confidential
C0000080h	Confidential
CFCFCF00h	Confidential
D0D0D000h	Instruction Counter Register
D0D0D001h	User Instruction Counter Register
D0D0D002h	Instruction Counter Control Register

6.3. I / O Mapped Registers

The I/O Mapped Registers are usually used to control the SoC integrated peripherals or to store the

peripherals' data, addresses and statuses. We divided these I/O Mapped Registers into below sub-sets.

These registers are listed as below. In another chapters, Functions and Registers Description will show more detailed information about these registers.

7. System Address Map

The SoC supports 4-Gbyte addressable memory space and 64-Kbyte addressable I/O space. In order to be compatible with PC/AT system, the lower 1 Mbytes of this addressable memory is divided into regions which can be individually controlled with programmable attributes such as disable, read/write, write only, or read only.

7.1. Memory Address Ranges

Figure 7-1 represents SoC memory address map. It shows the main memory regions defined and supported by the SoC. At the highest level, the address space is divided into two main conceptual regions. One is the 0–1-Mbyte DOS compatibility region and the other is 1-Mbyte to 4-Gbyte extended memory region. The SoC supports several main memory sizes. The main memory type and size in the system will be auto-detected by the system BIOS.
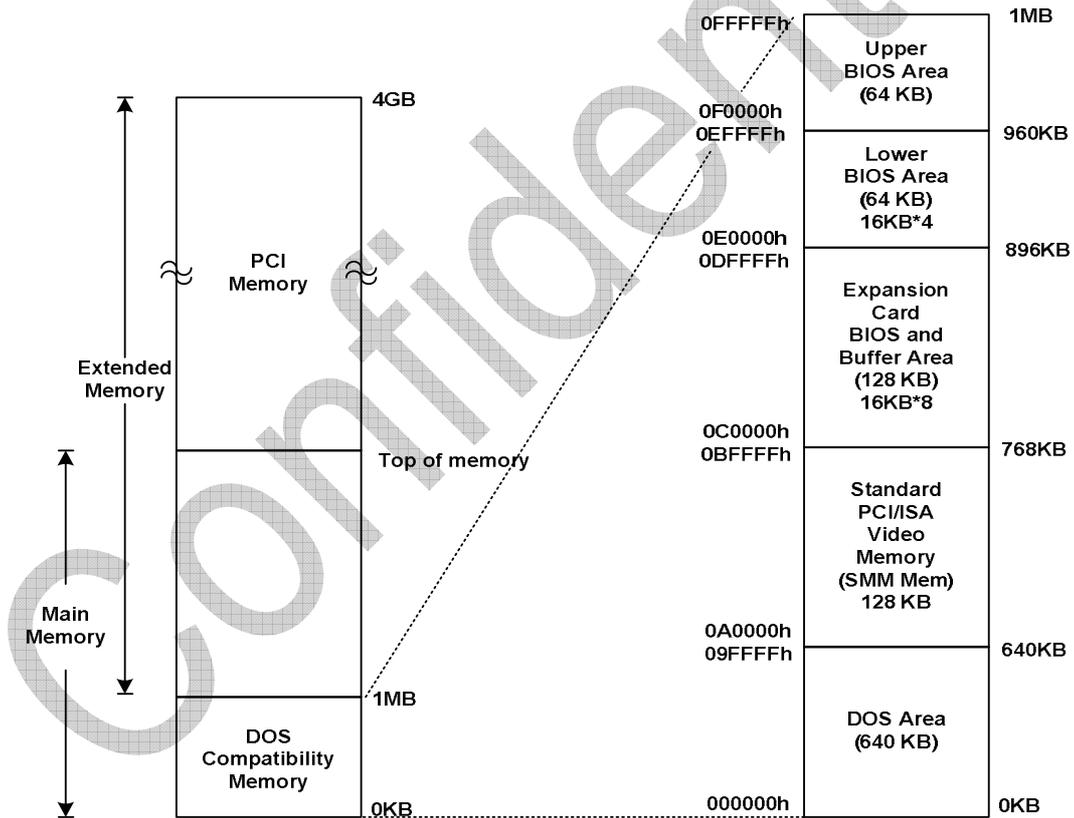
Figure 7-1. Memory Address Map

7.1.1. Dos Compatibility Region

The first region of memory is called the Dos Compatibility Region because it is defined for early PC. This area is divided into the following address regions:

- 0–640-KByte DOS Area

- 640–768-KByte Video Buffer Area

- 768–896-KByte in 16-Kbyte sections (total of 8 sections) - Expansion Area

- 896–960-KByte in 16-Kbyte sections (total of 4 sections) - Extended System BIOS Area

- 960-Kbyte–1-MByte Memory (BIOS Area) - System BIOS Area

From 640 KBytes – 1Mbytes: it can be divided into fourteen ranges which can be enabled or disabled independently for both read and write. These regions can also be mapped to either main DRAM or PCI by system BIOS. (See A/B Page control Register and Memory Attribute Register in North Bridge PCI configuration registers)

DOS Area (00000–9FFFFh)

The DOS area (00000h – 9FFFFh) is 640 Kbytes in size. It is always mapped to the main memory controlled by the SoC.

Video Buffer Area (A0000–BFFFFh)

The 128-Kbyte graphics adapter memory region is normally mapped to a video device on the PCI bus (typically VGA controller). This area is controlled by the A/B Page control Register. It can be mapped to either main DRAM or PCI for both read and write command.

ISA Expansion Area (C0000–DFFFFh)

This 128-Kbyte ISA Expansion region is divided into eight 16-Kbyte segments. Each segment can be assigned one of four Read/Write states: read-only, write-only, read/write, or disabled. Typically, these blocks are mapped through the PCI bridge to ISA space. Memory that is disabled is not remapped.

Extended System BIOS Area (E0000–EFFFFh)

This 64-Kbyte area is divided into four 16-Kbyte segments. Each segment can be assigned independent read and write attributes so it can be mapped either to main DRAM or to PCI. Typically, this area is used for RAM or ROM. Memory that is disabled is not remapped.

System BIOS Area (F0000–FFFFFh)

This area is a single 64-Kbyte segment that can be assigned read and write attributes. It is by default (after reset) read/write disabled and cycles are forwarded to PCI. By manipulating the read/write attributes, the SoC can "shadow" BIOS into main memory. Memory that is disabled is not remapped.

7.1.2. Extended Memory Region

This memory region covers 10_0000h (1 Mbytes) to FFFF_FFFFh (4 Gbytes minus 1) address range and is divided into the following regions:

- DRAM memory from 1 Mbytes to a top of memory

- PCI Memory space from the top of memory to 4 GBytes

- High BIOS area from 4 GBytes to 4 GBytes minus 16 MBytes

Main DRAM Address Range (0010_0000h to Top of Main Memory)

The address range from 1 Mbytes to the top of main memory is mapped to the main memory address range controlled by the SoC. All accesses to addresses within this range are forwarded to the main memory.

PCI Memory Address Range (Top of Main Memory to 4 GBytes)

The address range from the top of main DRAM to 4 GBytes is normally mapped to PCI. The PMC forwards all accesses within this address range to PCI.

1. High BIOS Area (FF00_0000–FFFF_FFFFh)

The top 16 Mbytes of the Extended Memory Region is reserved for System BIOS (High BIOS), extended BIOS for PCI devices, and the A20 alias of the system BIOS. The CPU begins execution from the High BIOS after reset. This region is mapped to the PCI so that the upper subset of this region is aliased to 16 Mbytes minus 256 Kbytes range. The actual address space required for the BIOS is less than 2 Mbytes.

7.2. <u>Memory Shadowing</u>

Any block of memory that can be designated as read only or write only can be "shadowed" into PMC DRAM memory. Typically, this is done to allow ROM code to execute more rapidly out of main DRAM. ROM is used as read only during the copy process while DRAM at the same time is designated write only. After copying, the DRAM is designated read only so that ROM is shadowed. CPU bus transactions are routed accordingly. The PMC does not respond to transactions originating from PCI or ISA masters and targeted at shadowed memory blocks.

7.3. <u>I/O Address Space</u>

The SoC positively decodes accesses to all internal registers, including PCI configuration registers (CF8h and CFCh), PC/AT Compatible I/O registers (8237, 8254 & 8259), and all relocatable I/O space registers (UART).

8. CPU Description

The VORTEX86EX2 is a 32-bit highly integrated SoC with 6-stage pipeline. It provides the ideal solution with low power consumption for embedded system integration. The following sections provide more detail on the sub-functions of the SoC.

8.1. SoC Core

The SoC integrates a high speed and high performance CPU core that is designed on advanced 32-bit, 6-stage pipeline architecture. The CPU core of SoC implements an MMU (Memory Management Unit).

8.1.1. Bus Unit

The bus unit manages data transformations, instruction prefetches and control functions between the processor's internal units and the SoC peripheral. Internally, the bus unit communicates with the cache and the instruction prefetch units through the 32-bit bus. Externally, the bus unit provides the processor with bus functions, including external bus cycles, memory read/write, instruction fetch, cache line fill, etc.,

8.1.2. Prefetch Unit

When the BUS UNIT is not performing bus cycles to execute an instruction, the instruction prefetch unit uses the BUS UNIT to prefetch instructions. By reading instructions before they are needed, the processor rarely needs to wait for an instruction prefetch cycle on the processor bus.

Instruction prefetch cycles read 32-byte blocks of instructions, starting at addresses numerically greater than the last-fetched instruction. The prefetch unit, which has a direct connection to the paging unit, generates the starting address. The 32-byte prefetched blocks are read into both the prefetch and cache units simultaneously. The prefetch queue in the prefetch unit stores 64 bytes of instructions. As each instruction is fetched from the queue, the code part is sent to the instruction decode unit and (depending on the instruction) the displacement part is sent to the segmentation unit, where it is used for address calculation. If loops are encountered in the program being executed, the prefetch unit gets copies of previously executed instructions from the cache.

8.1.3. Decode Unit

The instruction decode unit receives instructions from the instruction prefetch unit and translates them in a two-stage process into low-level control signals and microcode entry points. Most instructions can be decoded at a rate of one per clock.

The decode unit simultaneously processes instruction prefix bytes, opcodes, modR/M bytes, and displacements. The outputs include hardwired microinstructions to the segmentation, and integer units. The instruction decode unit is flushed whenever the instruction prefetch unit is flushed.

8.2. L1 Cache

In order to maximize the performance, the SoC integrated a 4-way, 16-KByte code and 16-KByte data cache in it. The level 1 cache supports write through policy. The on-chip L1 cache allows frequently used data and code to be stored on chip reducing accesses to the external bus. It significantly reduces the penalty of performance to access these codes and data from external slower memory devices.

8.3. 32-Bit Memory Addressing Mode

The SoC core provides several addressing modes for instructions to specify operands. The addressing modes are optimized to allow the efficient execution of high-level languages such as C and C++, and they cover the vast majority of data references needed by high-level languages.

8.3.1. Register and Immediate Modes

Two of the addressing modes provide instructions that operate on register or immediate operands:

Register Operand Mode: The operand is located in one of the 8-, 16- or 32-bit general registers. For example, the ADD instruction adds the EAX and EBX register value and save the result to the EAX register

Example: ADD EAX, EBX

Immediate Operand Mode: An operand that is directly encoded as part of an instruction is called an **immediate operand**. For example, the MOV instruction moves the immediate value 12345678h (HEX) to the EAX register.

Example: MOV EAX, 12345678h

8.3.2. <u>32-bit Memory Addressing Modes</u>

The memory addressing modes provide a mechanism for specifying the effective address of an operand. The effective address is calculated by using combinations of the following four address elements:

Displacement: An 8-, 16-, or 32-bit immediate value, following the instruction.

Base: The contents of any general-purpose register. The base registers are generally used by compilers to point to the start of the local variable area.

Index: The contents of any general-purpose register except for ESP. The index registers are used to access the elements of an array, or a string of characters.

Scale: The index register's value can be multiplied by a scale factor: 1, 2, 4 or 8. Scaled index mode is especially useful for accessing arrays or structures.

Combinations of these 4 components make up the 9 additional addressing modes. There is no performance penalty for using any of these addressing combinations, since the effective address calculation is pipelined with the execution of other instructions.

The **Effective Address (EA)** of an operand is calculated according to the following formula.

EA = Base + (Index * Scale) + Displacement

Direct Mode	:	The operand's offset is contained as part of the instruction as an 8-, 16- or 32-bit displacement.
Example	:	**INC Word PTR [50000]**
Register Indirect Mode	:	A **Base** register contains the address of the operand.
Example	:	**MOV [ECX], EDX**
Based Mode	:	A **Base** register's contents are added to a **Displacement** to form the operand's offset.
Example	:	**MOV ECX, [EAX + 24]**

Index Mode	**:**	An **Index** register's contents are added to a **Displacement** to form the operand's offset.
Example	**:**	**ADD EAX, TABLE[ESI]**

Scaled Index Mode	**:**	An **Index** register's contents are multiplied by a **Scaling** factor that is added to a **Displacement** to form the operand's offset.
Example	**:**	**IMUL EBX, TABLE[ESI • 4], 7**

Based Index Mode	**:**	The contents of a **Base** register are added to the contents of an **Index** register to form the effective address of an operand.
Example	**:**	**MOV EAX, [ESI] [EBX]**

Based Scaled Index Mode	**:**	The content of an **Index** register is multiplied by a **Scaling** factor and the result is added to the contents of a **Base** register to obtain the operand's offset.
Example	**:**	**MOV ECX, [EDX • 8] [EAX]**

Based Index Mode with Displacement	**:**	The contents of an **Index** register and a **Base** register's contents and a **Displacement** are all summed together to form the operand offset.
Example	**:**	**ADD EDX, [ESI] [EBP + 00FFFFF0H]**

Based Scaled Index Mode with Displacement	**:**	The contents of an **Index** register are multiplied by a **Scaling** factor and the result is added to the contents of a **Base** register and a **Displacement** to form the operand's offset.
Example	**:**	**MOV EAX, LOCALTABLE [EDI • 4] [EBP + 80]**

8.3.3. 32-bit Addressing Map

In the 32-bit addressing mode, the physical memory addresses range from 0000_0000h to FFFF_FFFFh (4 Gbytes).

● CPU start address at 0FFF_FFFF0h after reset

● Interrupt Vector Table with 256 interrupts at 0000_0000h to 0000_03FFh (1 Kbyte)
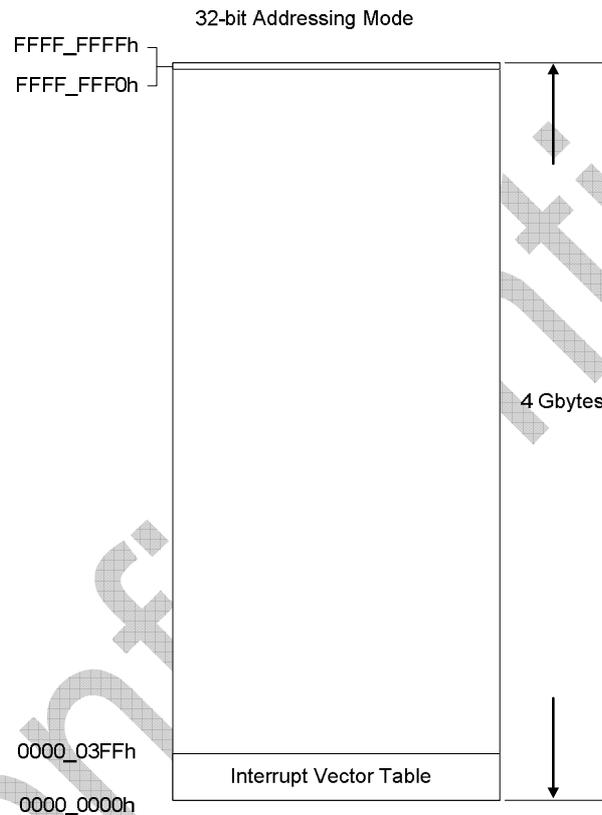

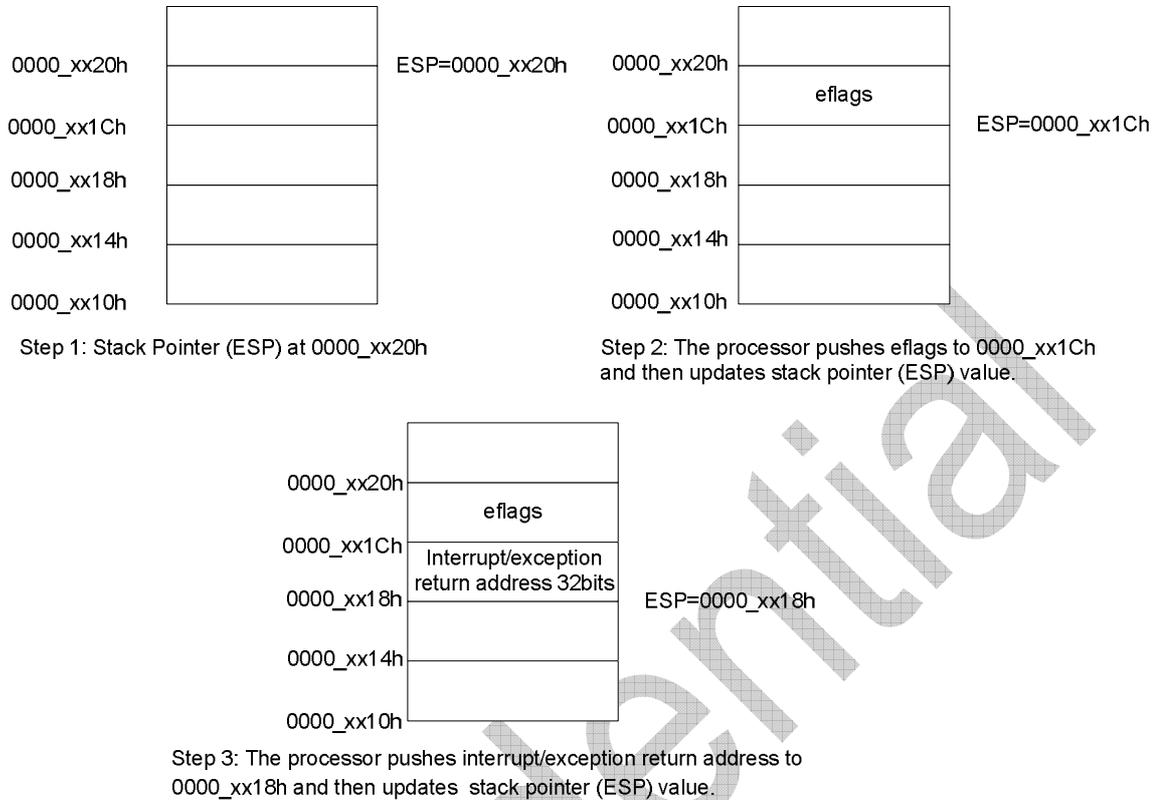**Figure 8-1. 32-bit Addressing Map and Interrupt Vector Table**

8.3.4. Interrupt Vector Table

● Interrupt Vector Table with 256 interrupts at 0000_0000h~0000_03FFh (1 Kbyte)

● Each interrupt has 4-byte space to store Interrupt Service Routine (ISR) entry point

INT255 ISR entry point	0000_03FFh
INT254 ISR entry point	0000_03FCh
INT253 ISR entry point	0000_03F8h
. . . . .	0000_03F4h
INT2 ISR entry point	0000_000Ch
INT1 ISR entry point	0000_0008h
INT0 ISR entry point	0000_0004h
	0000_0000h

Figure 8-2. Interrupt Vector Table

8.3.5. Interrupt & Exception Flow

Step 1: Stack Pointer (ESP) at 0000_xx20h

Step 2: The processor pushes eflags to 0000_xx1Ch and then updates stack pointer (ESP) value.

Step 3: The processor pushes interrupt/exception return address to 0000_xx18h and then updates stack pointer (ESP) value.

Figure 8-3. Interrupt & Exception Flow

PUSH/POP Behavior

Step 1: Stack pointer at 0000_xx20h

PUSH EAX

Step 2: The processor pushes EAX to 0000_xx1Ch and then updates stack pointer.
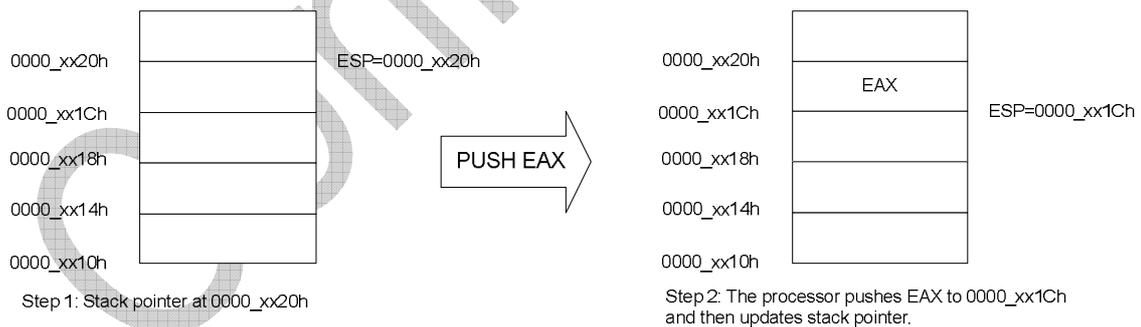
Figure 8-4. PUSH/POP Behavior

8.3.6. **Exception Type**

Exception	Vector	Condition
Division by zero	0	Attempting to execute a DIV or an IDIV instruction with a divisor which equals zero.
Breakpoint	3	A Breakpoint exception occurs when an INT3 instruction is executed. The INT3 is normally used by debug software to set instruction breakpoints by replacing instruction-opcode bytes with the INT3 opcode.
Overflow	4	It indicates that an overflow trap occurred when an INTO instruction was executed. The INTO instruction checks the state of the OF flag in the EFLAGS register. If the OF flag is set, an overflow trap is generated.
Bounds check	5	A bound check exception can occur as a result of executing the BOUND instruction. The BOUND instruction compares an array index with the lower bounds and upper bounds of an array. If the array index is not within the array boundary, the bound check exception occurs
Invalid opcode	6	An invalid opcode exception occurs when an attempt is made to execute an invalid or undefined opcode.
Floating-point error	16	Indicates FPU has detected a floating-point error conditions: Divide-by-Zero, Underflow, Overflow....etc.