TIGER320

PCI interface that is compliant with PC99, PCI 2.2 and Power Management 1.1

Internet Phones (VoIP), Software modems, ISDN TAs and other applications that require a low cost PCI 2.2 interface with bus mastering

Features

- PCI 2.2
  - PCI-SIG tested and approved
- Power Management 1.1
- PC99
  - Meets the requirements of WHQL
  - Enables Windows Logo certification
- TDM/IOM2 Industry standard serial bus
  - Flexible with many configuration options
  - Wide range of codecs supported
  - 32bit interface
- SLIC support
  - Silicon Labs Si3210
- ISDN AFEs supported
  - Siemens ISAC for S/T ISDN
  - Siemens IECQ for U ISDN
- Analog Modem Codecs supported
  - Silicon Labs Si3021/Si3012 (US)
  - Silicon Labs Si3021/Si3014 (Int.)
- Peripheral Interface Bus (PIB)
  - Byte wide data
  - 4 address lines
  - 8 control lines (AUX)
  - Read
  - Write
  - Reset
- Fully programmable PCI Subsystem ID
  - Uses resistor pull-up/downs
  - No EEPROM required
- Class code programmable
  - Network device
  - Communications device
- Microsoft Windows Plug and Play
  - Sample driver code available
- ISDN drivers
  - Windows95
  - Windows98
  - WindowsNT 4.0
  - Windows 2000 (beta)
  - U interface versions
  - S/T interface versions
- On-chip crystal oscillator
  - Can be used for codec clocks
  - Any external clock requirement
  - Supports low cost crystal
  - Not required by Tiger320 logic
- 3.3V power supply
  - Supports 5V and 3.3V signaling
- 100 PQFP package
  - 0.65mm lead spacing
  - Low cost
  - Easy to handle

This document contains information on a new product. Specifications and information herein are subject to change without notice.
All brand names and product names appearing in this document are registered trademarks or trademarks of their respective holders.

©TigerJet Network Inc

This datasheet has been downloaded from http://www.digchip.com at this page

General Description

The Tiger320 has been designed to enable easy implementation of low cost PCI interfacing solutions for general-purpose requirements. Typical examples are Internet phones where the Tiger320 would be partnered with a SLIC, software modems where the Tiger320 would be partnered with a DAA and codec, ISDN modems (terminal adaptors) where the Tiger320 would be partnered with an ISDN AFE and hard analog modems where the Tiger320 could enable a 3.3V PC99 solution.

Two interfaces are provided, the PIB (Peripheral Interface Bus) and the TDM/IOM2/PCM highway serial interface.

The PIB consists of an 8 bit data port, 4 address line control signals (read, write, reset) and 8 aux lines that can be configured for chip selects, interrupts etc. If the chip to be interfaced has an address and data bus this is the interface that would be used. An example of this would be a UART or a hard modem chip set.

The TDM interface can be used for interfacing most popular SLIC, ISDN, Analog modem DAAs and audio codecs. Bus mastering can be used to transfer data to the main memory of the host PC.

PCI interfaces that both meet the PCI specification and work across the full range of PC systems are a significant design challenge. The Tiger320 leverages the design expertise of the very successful Tiger300, Tiger100APC and Tiger600 chips and provides a PCI interface that meets the requirements of PCI 2.2 and the Power management 1.1 specifications.

PCI subsystem vendor and subsystem IDs are set by pull up/down resistors, this provides a significant cost saving in comparison with the expensive EEPROM approach that is used by other vendors.

In addition the Tiger320 provides a crystal oscillator cell that can be used to implement a clock source for codecs etc. This oscillator is not required by any circuitry of the Tiger320.

Ordering Information

The order code for the Tiger320 is Tiger320.

Block Diagram

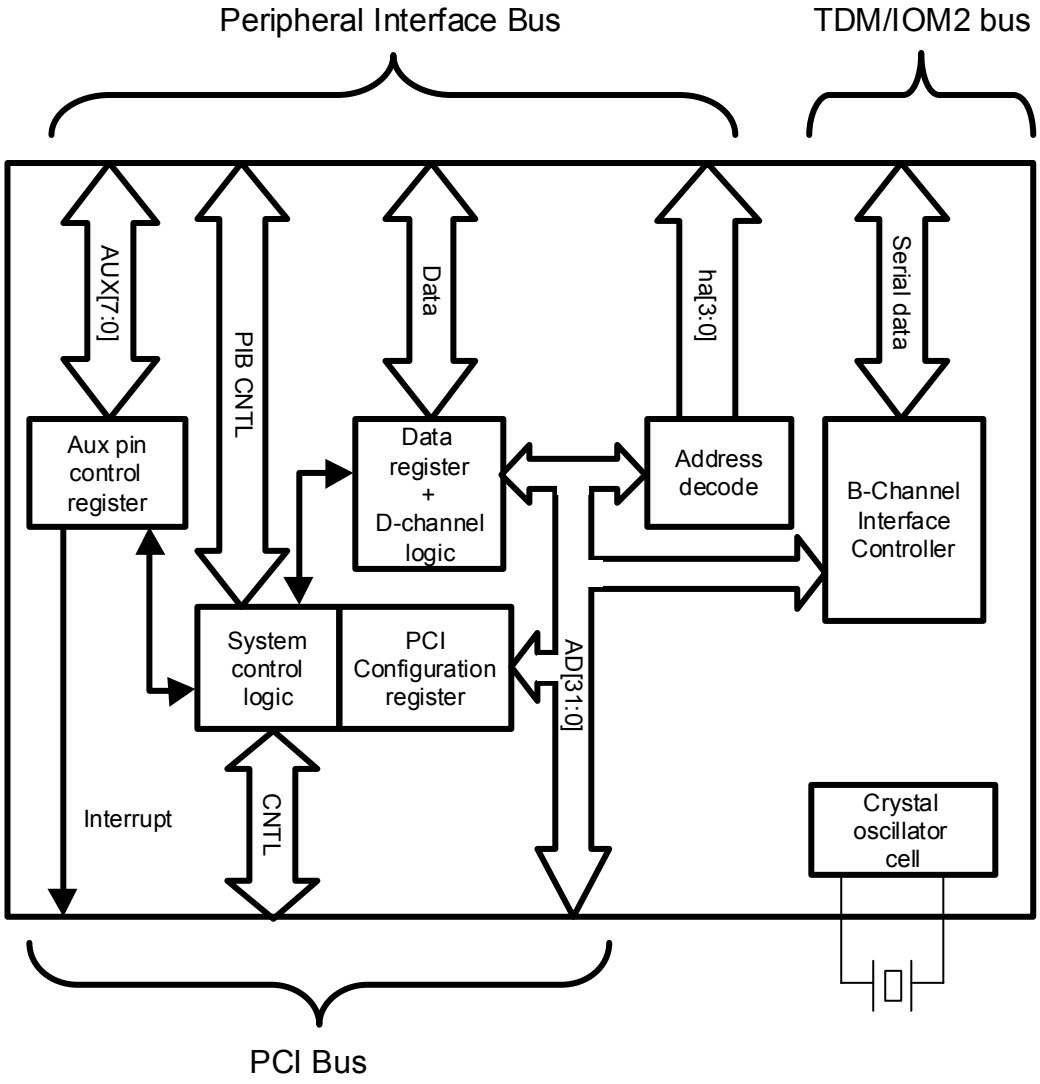
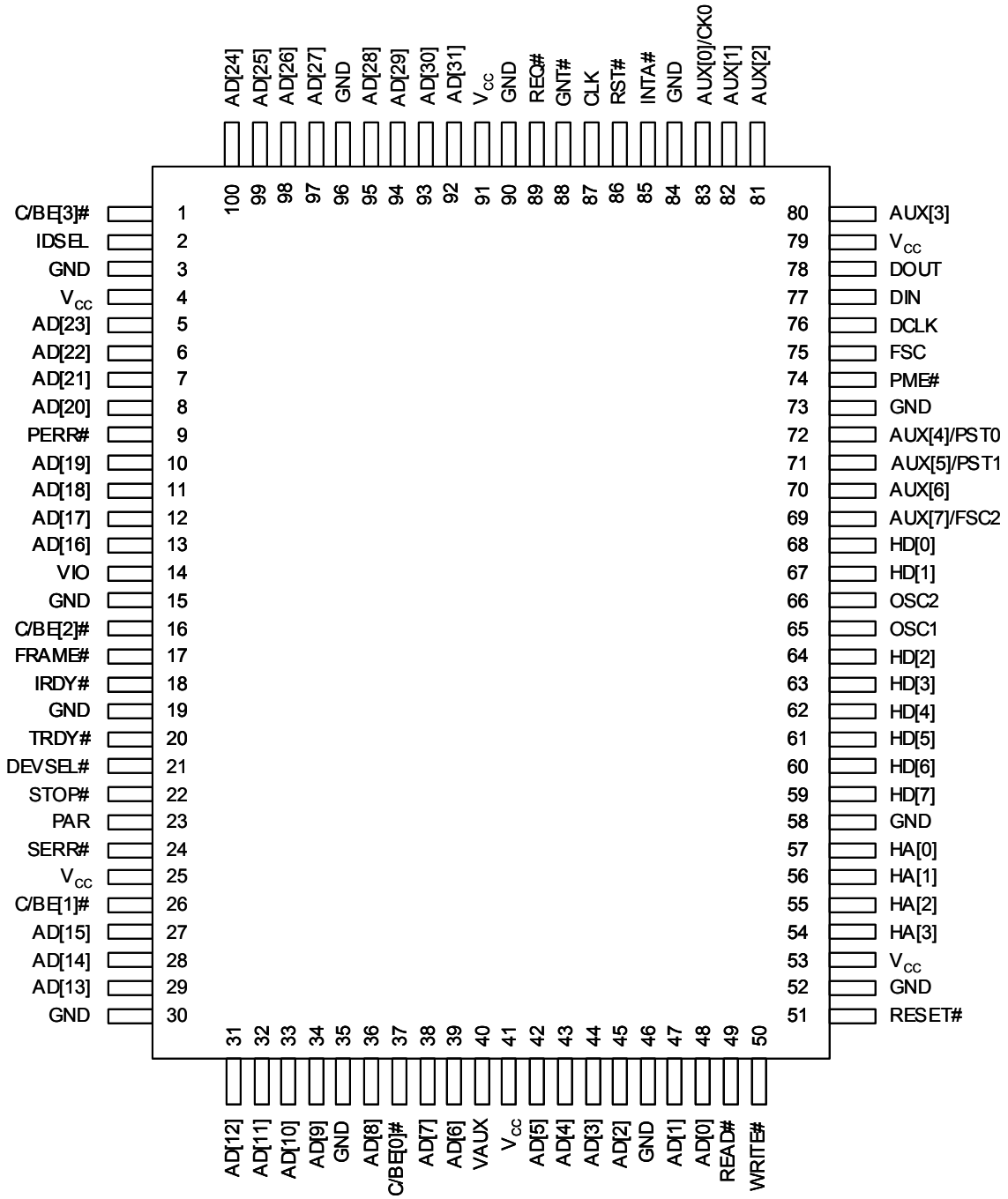


Table of Contents

Pin-out

Pin-out diagram



Pin assignment by pin number

Pin	Name
1	C/BE[3]#
2	IDSEL
3	GND
4	V_{CC}
5	AD[23]
6	AD[22]
7	AD[21]
8	AS[20]
9	PERR#
10	AD[19]
11	AD[18]
12	AD[17]
13	AD[16]
14	VIO
15	GND
16	C/BE[2]#
17	FRAME#
18	IDRY#
19	GND
20	TRDY#
21	DEVSEL#
22	STOP#
23	PAR
24	SERR#
25	V_{CC}
26	C/BE[1]#
27	AD[15]
28	AD[14]
29	AD[13]
30	GND
31	AD[12]
32	AD[11]
33	AD[10]
34	AD[9]

Pin	Name
35	GND
36	AD[8]
37	C/BE[0]#
38	AD[7]
39	AD[6]
40	VAUX
41	V_{CC}
42	AD[5]
43	AD[4]
44	AD[3]
45	AD[2]
46	GND
47	AD[1]
48	AD[0]
49	READ#
50	WRITE#
51	EXTRST#
52	GND
53	V_{CC}
54	HA[3]
55	HA[2]
56	HA[1]
57	HA[0]
58	GND
59	HD[7]
60	HD[6]
61	HD[5]
62	HD[4]
63	HD[3]
64	HD[2]
65	OSC1
66	OSC2
67	HD[1]

Pin	Name
68	HD[0]
69	AUX[7]/FSC2
70	AUX[6]
71	AUX[5]/PST1
72	AUX[4]/PST0
73	GND
74	PME#
75	FSC
76	DCLK
77	DIN
78	DOUT
79	V_{CC}
80	AUX[3]
81	AUX[2]
82	AUX[1]
83	AUX[0]/CKO
84	GND
85	INTA#
86	RST#
87	CLK
88	GNT#
89	REQ#
90	GND
91	V_{CC}
92	AD[31]
93	AD[30]
94	AD[29]
95	AD[28]
96	GND
97	AD[27]
98	AD[26]
99	AD[25]
100	AD[24]

Signal assignments by functional category

PCI Control Signals	
Name	**Pin**
C/BE[0]#	37
C/BE[1]#	26
C/BE[2]#	16
C/BE[3]#	1
IDSEL	2
FRAME#	17
IRDY#	18
TRDY#	20
DEVSEL#	21
STOP#	22
PAR	23
INTA#	85
RST#	86
GNT#	88
REQ#	89
CLK	87
PERR#	9
SERR#	24
VIO	14
VAUX	40
PME#	74

Serial Ports	
Name	**Pin**
DCLK	76
DIN	77
DOUT	78
FSC	75
FSC2/AUX[7]	69

PCI Address/Data	
Name	**Pin**
AD[0]	48
AD[1]	47
AD[2]	45
AD[3]	44
AD[4]	43
AD[5]	42
AD[6]	39
AD[7]	38
AD[8]	36
AD[9]	34
AD[10]	33
AD[11]	32
AD[12]	31
AD[13]	29
AD[14]	28
AD[15]	27
AD[16]	13
AD[17]	12
AD[18]	11
AD[19]	10
AD[20]	8
AD[21]	7
AD[22]	6
AD[23]	5
AD[24]	100
AD[25]	99
AD[26]	98
AD[27]	97
AD[28]	95
AD[29]	94
AD[30]	93
AD[31]	92

Peripheral Interface Bus (PIB)	
Name	**Pin**
AUX[0]/CKO	83
AUX[1]	82
AUX[2]	81
AUX[3]	80
AUX[4]/PST0	72
AUX[5]/PST1	71
AUX[6]	70
AUX[7]FSC2	69
EXTRST#	51
HA0	57
HA1	56
HA2	55
HA3	54
HD0	68
HD1	67
HD2	64
HD3	63
HD4	62
HD5	61
HD6	60
HD7	59
READ#	49
WRITE#	50

OSC interface	
Name	**Pin**
OSC1	66
OSC2	65
CKO/AUX[0]	83

Power and Ground	
Name	**Pin**
V_{CC}	4, 25, 41, 53, 79, 91
GND	3, 15, 19, 30, 35, 46, 52, 58, 73, 84, 90, 96

Signal descriptions

Signal Name	Type	Description	Alternate Function
AD[0:31]	I/O	Multiplexed address and data	
AUX[0]	I/O	PIB AUX port bit 0	Crystal Clock output
AUX[1]	I/O	PIB AUX port bit 1	Subsystem Vendor ID bit 14
AUX[2:3]	I/O	PIB AUX port bits 2 – 3	
AUX[4]	I/O	PIB AUX port bit 4	Power State bit 0 output Configuration register 0x44 bit 0
AUX[5]	I/O	PIB AUX port bit 5	Power State bit 1, output Configuration register 0x44 bit 1
AUX[6]	I/O	PIB AUX port bit 6	Subsystem Vendor ID bit 15
AUX[7]	I/O	PIB AUX port bits 7	1. Subsystem Vendor ID bit 13 2. FSC2 output
CLK	I	PCI bus clock	
DCLK	I	Serial port data clock	
DIN	I	Serial port data input	
DOUT	O	Serial port data output	
C/BE[0:3]#	I/O	Bus command and byte enable	
DEVSEL#	I/O	Device detect for configuration read and write cycles	
EXTRST#	O	Reset for IOM2 & PIB peripherals	Subsystem Vendor ID bit 10
FRAME#	I/O	Cycle frame indicates the beginning and duration of access	
FSC	I	Serial port frame sync	
GNT#	I/O	Bus grant to master request	
HA[0]	I/O	PIB address bit 0	Subsystem Vendor ID [0]
HA[1]	I/O	PIB address bit 1	Subsystem Vendor ID [1] Class code, 0=Network, 1=Com.
HA[2:3]	I/O	PIB address bit 2, 3	Subsystem Vendor ID bit [8:9]
HD[0:7]	I/O	PIB data bus	Subsystem Vendor ID bit [0:7]
IDSEL	I	Chip select for configuration read and write cycles	
INTA#	O	PCI bus interrupt	
IRDY#	I/O	Initiator ready	
PAR#	I/O	Even parity across AD[0:31] and C/BE[3:0]#	
PME#	O	Power management	
OSC1	I	OSC input	
OSC2	O	OSC output	
PERR#	I/O	Parity Error	
REQ#	I/O	Bus master request	
READ#	I/O	PIB read command	Subsystem Vendor ID bit 11
RST#	I	Reset from the PCI bus	
STOP#	I/O	Indicates the current target is requesting the master to stop the current transaction	
SERR#	OD	System Error	
TRDY#	I/O	Target ready, completion of current phase as a receiver	
WRITE#	I/O	PIB write command	Subsystem Vendor ID bit 12

Functional Description

PCI bus interface

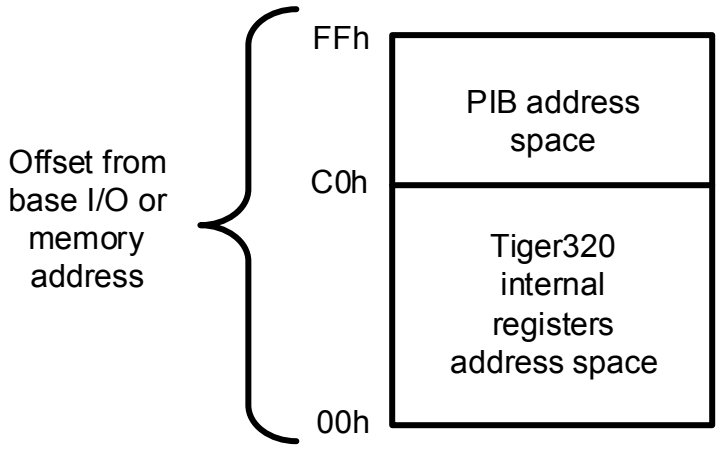
The Tiger320 implements a PCI 2.2 compliant bus interface and implements revision 1.1 of the PCI power management specification. The Tiger320 is built in a 3.3V process and supports both 5V and 3.3V signaling environments.
PCI specifications are available from the PCI special interest group at; www.pcisig.com

Address space

Internal registers and the PIB address bus of the Tiger320 can be accessed either by I/O or memory cycles.

The Tiger320 address space consists of FFh memory or I/O locations, actual base addresses are set by PCI configuration register 10h for I/O or 14h for memory. I/O and memory cycles can be used interchangeably to access the internal registers and PIB address space.

The first C0h locations from the base address enable access the Tiger320 internal registers. The following 30h locations enable access to the PIB address space. The diagram below illustrates the addressing concept.



Subsystem Vendor ID

The PCI subsystem vendor I.D. is set by pull-up and pull-down resistors on the following pins.

Subsystem Vendor I.D. bits 0 – 7 on PIB data bus HD0 – HD7
Subsystem Vendor I.D. bits 8 – 9 on PIB address bus HA2 – HA3
Subsystem Vendor I.D. bit 10 on PIB external reset EXTRST#
Subsystem Vendor I.D. bit 11 on PIB read command READ#
Subsystem Vendor I.D. bit 12 on PIB write command WRITE#
Subsystem Vendor I.D. bit 13 on PIB AUX[7]
Subsystem Vendor I.D. bit 14 on PIB AUX[1]
Subsystem Vendor I.D. bit 15 on PIB AUX[6]

On reset the inputs are read into configuration register 2Ch, it is important to ensure that peripheral devices connected to these inputs tri-state the pins connected to the Tiger320 on reset. The default value for the PCI subsystem vendor ID is 0001H. 50K pull-up/down resistors

built into the input pads of the chip establish the default values. When defining different subsystem vendor ID resistors are only required on pads that have to be pulled to the opposite state.

The mapping is as follows;

Subsystem Vendor ID																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Input	AUX6	AUX1	AUX7	WRITE#	READ#	EXTRST#	HA3	HA2	HD7	HD6	HD5	HD4	HD3	HD2	HD1	HD0

Subsystem ID and class code

Subsystem ID and class code are set by pull-up and pull down resistors on the following pins.

Subsystem I.D. bits 0 – 1 on PIB address bus HA0 – HA1

The mapping is as follows;

Subsystem ID																
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Input	0	0	0	0	0	0	0	0	0	0	0	0	0	0	HA1	HA0

In addition the class code of the PCI card is set by bit 17,

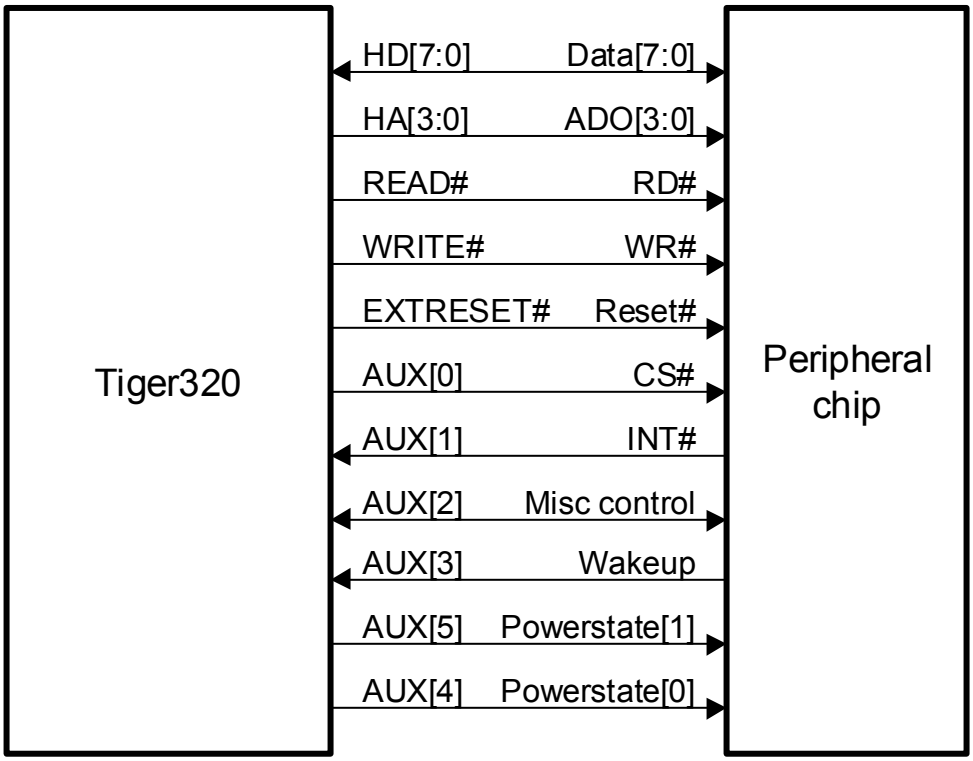
Bit17 = 0, Network device
Bit17 = 1, Communication device

Configuration space

The Tiger320 implements the standard PCI configuration space, for full details please consult the PCI 2.2 specification.

Peripheral Bus Interface (PIB)

The PIB enables a “glueless” interface to most popular peripheral chips. Address, data and control lines are fully qualified and the access cycle time can be adjusted to support slow devices.



Typical connection of a peripheral chip to the Tiger320 PIB

The connection details will vary with application, the diagram above shows a typical application. AUX[0] is used to generate the chip select to the peripheral chip, in the event that more that one peripheral chip is required in a system, additional AUX lines can be used to generate chip select signals to other chips. AUX[1] is used for generating the interrupt, AUX[3] for wake-up of the host PC, for more information in the register setting for handling the interrupt and the PME#, please see the section on PME# and PCI INT# generation logic. In the event that the peripheral can be set into different power states, AUX [5] and [4] can be used to set these.

PCI to PIB address mapping – I/O cycle

The diagram below illustrates how the PCI address is mapped to the PIB address.

A31	-	-	-	-	-	-	A8	A7	A6	A5	A4	A3	A2	A1	A0
PCI A8 to A31 equal to config register 10h and A6 and A7 both equal 1								1	1	ha3	ha2	ha1	ha0	0	0
										A2 to A5 mapped to ha0 to ha3					

PCI to PIB address mapping – memory cycle

The diagram below illustrates how the PCI address is mapped to the PIB address.

A31	-	-	-	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
PCI A11 to A31 equal to config register 14h A8, A9 and A10 equal to 0 A6 and A7 both equal 1					0	0	0	1	1	ha3	ha2	ha1	ha0	0	0
										A2 to A5 mapped to ha0 to ha3					

Data mapping

The PIB has an 8bit data bus that is mapped to the lowest 8 bits of the 32 bit PCI bus. This means that when data is read from (or written to) a specific address the next valid address is 4 bytes higher. So, if you read date from C0h the next valid address is C4h.

Read cycle

When a PCI read cycle takes place and provided the address has been qualified, the READ# signal will be active for 3 – 12 PCI cycles, the actual number of cycles is determined by the setting of bits 4 and 5 in the internal register 0x00.

Data from the PIB data bus will be latched on the rising edge of READ# and transferred to the PCI data bus. The PIB address bus is driven by the Tiger320 one PCI clock cycle prior to the falling edge of the READ# pulse and at least one cycle after the rising edge of the READ# pulse. The setup and hold time for latching data from the PIB is 3nS.

Write cycle

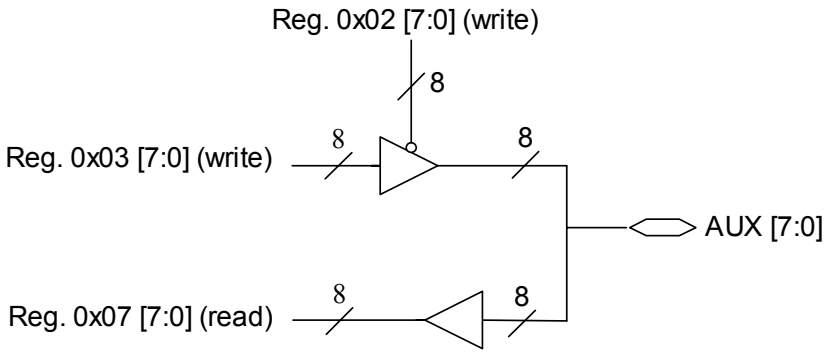
When a PCI write cycle takes place and provided the address has been qualified, the WRITE# signal will be active for 3 – 12 PCI cycles, the actual number of cycles is determined by the setting of bits 4 and 5 in the internal register 0x00.

Data and address information on the PIB data bus will be valid one cycle prior to the falling edge of the PIB WRITE# line. The data will remain valid until one cycle prior to the falling edge of the PIB WRITE# signal.

AUX lines

Setting to input or output
The PIB AUX[7:0] pins can be individually programmed as inputs or outputs. Register 0x02 determines which AUX pins are defined as inputs and which are defined as outputs. Bit0 in the register controls the state of AUX0, bit1 controls AUX1 etc. A 1 in the register defines an AUX line as an output. A 0 in the register location defines the appropriate line as input. On a hard reset all AUX lines float and are defined as inputs.

Reading status
The status of the AUX pins can be read from register 0x07. The actual AUX pin value will be read irrespective of it being an input or output.



AUX lines - additional functions

Setting Subsystem Vendor ID
AUX[7] is used as Subsystem vendor ID bit [13], AUX[6] as Subsystem vendor ID bit 15 and AUX[1] as Subsystem Vendor ID bit 14 during PCI hardware reset. After the PCI hardware reset, these pins will revert back to normal operation. It is important that the board level design to ensures these pins will not be driven by other devices during the PCI hardware reset time and the states during reset are controlled by the pull-up/down resistors either internal to the pads or by external resistors.

Crystal oscillator output
AUX[0] can be programmed to function as the crystal clock output. Internal register 0x2b bit 2 is used to enable this function. When it is set to 1, AUX[0] will be the output for the crystal oscillator.

Second FSC signal
AUX[7] can be programmed to function as the second FSC signal for the serial bus. Internal register 0x2b bit 1 is used to enable this function. When it is set to 1, AUX[7] will be the second serial FSC signal. Please refer to serial port interface for the definition of second serial FSC signal.

Power state output
AUX[5:4] can be programmed to output the power state bits[1:0] from the PCI configuration register 44h bits[1:0]). Internal register 0x2b bit 0 is used to enable this function When bit 0 set to 1 this function will be enabled.

AUX lines – control priority

The actual state definition of the AUX[0:7] pins follows the priorities detailed below;

Highest priority - PCI hardware reset

The PCI hardware reset has the highest priority. When it occurs, AUX[6:1:7] will be used to set the Subsystem Vendor ID[15:14:13]. It will also clear the internal registers 0x2 and 0x2b to 0 which will result in AUX[7:0] set to input mode.

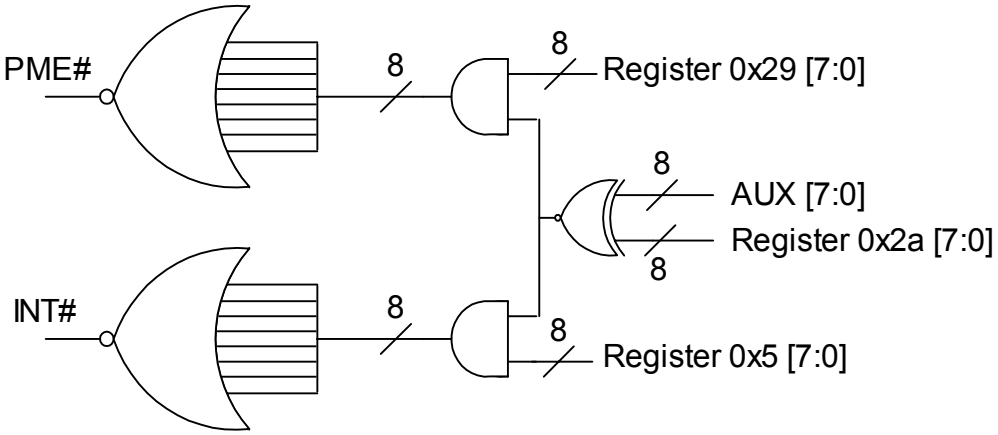
Lowest priority - Internal register 0x2b

Internal register 0x2b has second priority. When register 0x2b bit 0 set to 1, AUX[7:6] will be set as outputs for PCI power state bits[1:0] no matter what is the value in register 0x2. When register 0x2b bit 1 set to 1, AUX[1] will be used as FSC2 and when register 0x2b bit 2 set to 1, AUX[0] will be the CKO (Crystal Oscillator Output).

PME# and PCI INT# generation logic

PME# (Power Management Event) can be used to wake-up a PC that is in sleep or power down mode, for example this could be used in a modem to wake up the PC when the phone line rings. PME# can be generated from any of the AUX lines and the external event can be either active high or low. Register 0x2a is used to invert the inputs from the AUX pins. Register 0x29 is used as the mask register. The diagram below shows the logical equivalence of the circuitry.

INT# (Interrupt) is generated in exactly the same manner with the difference in that register 0x5 is used as the mask register.



TDM/IOM2 serial port

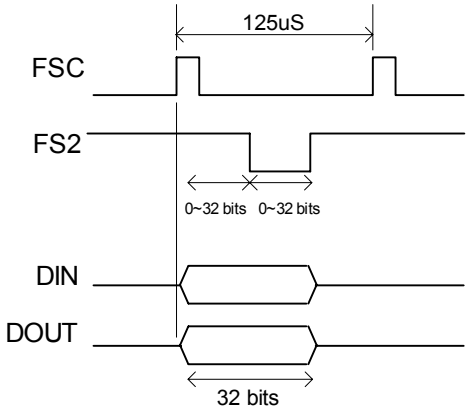
The Tiger320 serial port interface provides a symmetrical full-duplex communication link to either an ISDN front end such as the Siemens 2091 (U interface) or 2186 (S/T interface). In addition analog modem front ends such as the Silicon Labs and general purpose audio codecs are also supported.

Serial port interface

The interface consists of one data clock (DCLK), one frame synchronization clock (FSC) and two data lines (DIN and DOUT).
DCLK and FSC are inputs to the Tiger320. DCLK can either be the same as the data rate or twice the data rate. If the DCLK input to Tiger320 is the same rate as the data rate, the internal clock doubler should be turned on for correct operation. There are software controlled switches for inverting the DCLK and to control the high pulse time for the clock doubler for different applications.
This interface is fully compatible with IOM2 bus timing. IOM channel 0 is the channel used for the data communication. The 32 bit channel 0 data will be stored and transferred through the Tiger320 PCI bus master logic to and from main system memory. The serial port incoming data is written to main memory through the Bus master DMA read port (read in form the codec and written to RAM). The outgoing data is read from the main system memory through the Bus master DMA write port (read from RAM and written out).

Input clock

The FSC input to the Tiger320 should be an 8 KHz clock. Within one FSC period, the first 32 bits of data are transmitted and received. The data is clocked in and out on by the DCLK. For details on the serial port timing please see the timing diagram in the section on A.C. characteristics.



DOUT is open-collector, it should be pulled high with an external pull-up resistor. Outside the 32 bit data transfer window, DOUT can either be set to float or DOUT pins can be forced to drive low. This option allows some peripherals to activate the data clock.

Data transfer

Data is transferred from the IOM2 bus to main system memory by bus mastering.

Second FSC signal

The FSC2 signal is an option pin for support of a second codec. FSC2 is shared with AUX[7] pin and can be enabled by register 0x2B bit 1. The timing of FSC2 is controlled by register 0x2C bit[4:0]. The value in register 0x2C bit[4:0] controls the starting cycle for the FSC2 to go low after the FSC signal. The value in register 0x2E bit[4:0] controls the cycle number for FSC2 to stay low before going high. There is no hardware guarantee for the FSC2 timing if FSC stroke happens during the low time of FSC2.

Parallel to Serial/ Serial to Parallel shift direction

The serial port data is serially shifted in and out from the PCI bus master channel. The shift direction can be controlled by register 0x2D bits 6 and bit 7. One bit is used for input and the other is used for output.
Set to 0, the 32 bit shift register will shift to right (LSB first).
Set to 1, the 32 bit shift register will shift to left (MSB first).

Important to note that the shift register is 32 bits long.
The software driver needs to arrange the byte order of the data before it is DMAed to the IOM2 bus.

Bus mastering DMA

Tiger320 PCI bus master supports bi-directional data transfer.
DMA-READ is referred as the data transfer from the serial input port to the system memory.
DMA-WRITE is for data transferred from system memory to the serial output port.
Each direction of DMA has it own set of registers and logic. They can be programmed for different configurations of the DMA transfer.

Setup of DMA operation

To setup the DMA operation, first the DMA start address and the DMA end address registers need to be set to the relevant physical system memory locations. The Tiger320 DMA logic does not require a counter value.

The difference between the DMA end address and DMA start address will be the amount of data that is to be transferred by the DMA. If the start address is “X” and the data transferred is “n” bytes, the end address will be “X + n - 4”. The minus 4 is needed to point to the very last location as the first location is 0 and not 1.
The serial port will always transfer 32 bits of data for each frame sync.
Important to note that the DMA start and end addresses must be on a double word boundary.

Status register

There is a DMA interrupt address register to enable DMA status checking. When the DMA reaches the interrupt address, it will set the flag in Interrupt 0 status register (Index 0x6). Reading the status register and writing back the read value of 1 (if an interrupt has occurred) will clear the flag. Please note, writing a 0 will not clear the flag.

Control register

Control register (Index 0x0) bit 6 controls the mode for generating the interrupt.
Set to 0 for level trigger, in this mode the interrupt will occur when the event happens and the status bit will remain set to 1 only as long as the conditions that created the interrupt still exist.
Set to 1 for edge trigger, in this mode the interrupt will occur when the event happens and the status bit will remain set until it is cleared by writing a 1 to the DMA status register.

Control register (Index 0x0) bit 7 is used to configure the DMA operation mode. When it is set to 0, the bus master DMA is set as self-address mode. The DMA will automatically wrap around to the starting address if the ending address is reached. If it is set 1, normal mode, the DMA will stop DMA when the end address is reached. The major difference between the self-address DMA mode and the normal DMA mode is the hardware will continuously transfer data without CPU attention in the self-address mode.

To start the DMA operation, set the Operation register (Index 0x1) bit 0 to 1. To stop the DMA operation, set this bit to 0. Bit 1 of this register is used to restart the DMA operation in normal mode. This bit is not used in the self-address mode.

The status of DMA transfer can be monitored by reading the current DMA address. This register contains the current DMA address pointer. Interrupt 0 status register (Index 0x6) bit 0 to 3 has the status of the current DMA operation. Each bit can generate interrupt if the mask bit is set.

In the event that the PCI bus master logic detects an abort condition, Interrupt 0 status register (Index 0x6) bit 4 and 5 will be set based on the abort condition. It can be cleared by reading the status register. To recover the PCI master state machine, set the reset bit in register 0 to 1 and reset it to 0, also enable the DMA operation by setting register 1 bit 0 to 1.

DMA register addresses

Index 0xB ~ 0x8	DMA write starting address
Index 0xF ~ 0xC	DMA write interrupt address
Index 0x13 ~ 0x10	DMA write end address
Index 0x17 ~ 0x14	DMA write current address
Index 0x1B ~ 0x18	DMA read starting address
Index 0x1F ~ 0x1C	DMA read interrupt address
Index 0x23 ~ 0x20	DMA read end address
Index 0x27 ~ 0x24	DMA read current address

Software considerations

The basic setup flow for the DMA is:
1. Allocate non-cacheable continuous DMA memory.
2. Set DMA start and end address registers.
3. Fill the DMA write buffer.
4. Set DMA start bit.

Self-address DMA mode is the most powerful feature of Tiger320. The DMA hardware only requires to be initialized once and no further software setup is necessary. It provides the status registers and interrupt for the host processor to detect current DMA status. Also, it provides an almost unlimited system memory buffer for the data transfer.

Once the DMA is setup, the main CPU can use a timer polling routine to monitor the current DMA status and process the downstream and upstream data buffer. The interval time is based on the size of allocated system memory.

The normal DMA mode has a similar control flow as PC DMA controller. The major difference is there is no counter register. The end address is used instead. Also, the trigger of the DMA has it's own control register.

Watchdog timer

A 24-bit internal free running counter is used for the DMA watchdog timer. The counter is clocked by the PCI system clock and is free running. It will be reset by two conditions.
1. The PCI hardware reset.
2. Any internal register read access.

If the watchdog timer is enabled, register 0x2b bit 3 set to 1. The counter will overflow in about 500ms without any internal register access.

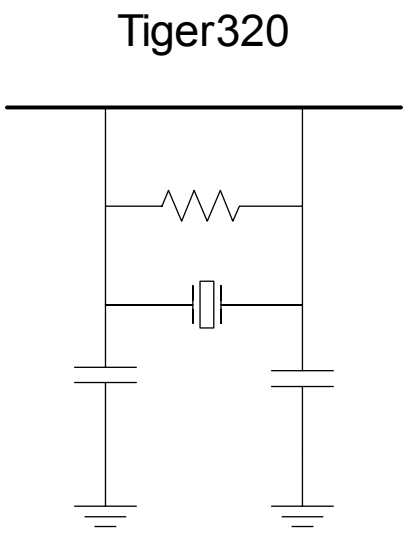
In the event of an overflow the DMA logic will do following;
1. Generate a reset pulse to PCI bus master block. This will stop the DMA operation.
2. Set status bit (bit 4) in register 0x2B.
3. Reset AUX control register 0x2 to 0.
4. Reset bits[2:0] of AUX selection register 0x2B to 0.

The 24-bit counter is divided into three 8 bit counters. Each counter can be set to 0xFF to reduce the time for the 24 bit counter to overflow. Setting register 0x2B bit 5 to 1 will set counter bit [7:0] to all 1's. Setting register 0x2B bit 6 to 1 will set counter bit [15:8] to all 1's. Setting register 0x2B bit 7 to 1 will set counter bit [23:16] to all 1's. For the watchdog counter to operate, all 3 bits need to be set to 0.

Crystal Oscillator

The crystal oscillator cell is in essence an inverting buffer that is biased into a high gain amplifier with a feedback resistor. A series mode crystal sets the frequency of oscillation. To ensure the reliable starting the reset pulse will restart the oscillator. Crystals between 1MHz and 20MHz can be used.

Typical circuit



The recommended resister value is 1M ohm, capacitors should be 10pF. The crystal should be placed close to the Tiger320 and good layout practice followed.

Registers

Tiger320 Internal Register Description

Reset and PIB cycle time offset 0x00

Bit	7	6	5	4	3	2	1	0
	DMA operation mode	DMA interrupt mode	PIB cycle time	PIB cycle time	Reserved	Reset serial port	Reset master	EXTRST#

Type: R/W
Default: 00h

DMA bus master operation mode

0 = Self address mode (continues loop)
1 = Normal DMA operation

DMA interrupt mode

0 = Status bit will be set only when DMA current address is equal to interrupt address or end address.
1 = Status bit will be set until the status bit has been cleared by writing a 1.

PIB cycle time

00 = PIB 3 cycle operation (fastest)
01 = PIB 8 cycle operation
10 = PIB 15 cycle operation (slowest)
11 = PIB 15 cycle operation (slowest)
NOTE cycle numbers are for read operation, write operation will be one cycle less.

Reset serial port

0 = Normal operation
1 = Reset of serial port logic

Reset DMA logic

0 = Normal operation
1 = Reset of DMA logic

EXTRST# pin state

0 = External reset pin on PIB low
1 = External reset pin on PIB high

DMA operation offset 0x01

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Restart DMA	Enable DMA

Type: R/W
Default: 00h

Restart DMA

0 = Hold
1 = Restart DMA using current DMA setup, used for “normal DMA operation”

Enable DMA

0 = Stop DMA
1 = Run DMA

PIB Aux Port Control 0x02

Bit	7	6	5	4	3	2	1	0
	AuxC7	AuxC6	AuxC5	AuxC4	AuxC3	AuxC2	AuxC1	AuxC0

Type: R/W
Default value: 00h

AuxC[7:0]

0 = Line configured as an input
1 = Line configured as an output

PIB Aux Port Data 0x03

Bit	7	6	5	4	3	2	1	0
	AuxD7	AuxD6	AuxD5	AuxD4	AuxD3	AuxD2	AuxD1	AuxD0

Type: R/W
Default value: 00h

AuxD[7:0]

Write = Sets the state of Aux lines configured as outputs
Read = Reads the status of all Aux lines both inputs and outputs

Interrupt 0 mask 0x04

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Enable PCI target abort	Enable PCI master abort	Enable DMA read address reach end of buffer address	Enable DMA read address reach interrupt address	Enable DMA write address reach end of buffer address	Enable DMA write address reach interrupt address

Type: R/W
Default: 00h

Enable PCI target abort

0 = No interrupt generated
1 = Interrupt generated when PCI target abort occurs

Enable PCI master abort

0 = No interrupt generated
1 = Interrupt generated when PCI master abort occurs

Enable DMA read address reach end of buffer address

0 = No interrupt generated
1 = Interrupt generated when the DMA read address reaches the end of buffer address

Enable DMA read address reach interrupt address

0 = No interrupt generated
1 = Interrupt generated when the DMA read address reaches the interrupt address

Enable DMA write address reach interrupt address

0 = No interrupt generated
1 = Interrupt generated when the DMA write address reaches the interrupt address

Enable DMA write address reach end of buffer address

0 = No interrupt generated
1 = Interrupt generated when the DMA write address reaches the end of buffer address

Interrupt 1 mask register 0x05

Bit	7	6	5	4	3	2	1	0
	AuxInt7	AuxInt6	AuxInt5	AuxInt4	AuxInt3	AuxInt2	AuxInt1	AuxInt0

Type: R/W
Default value: 00h

AuxInt[7:0]

0 = Ignore input for generation of interrupt 1
1 = Select an input(s) to generate an interrupt 1

Interrupt 0 status 0x06

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	PCI target abort	PCI master abort	DMA read address reach end of buffer address	DMA read address reach interrupt address	DMA write address reach end of buffer address	DMA write address reach interrupt address

Type: R/W1TC
Default: No default value, depending on chip status

Note: when reading this register, immediately write back the value that has been read for correct operation.

PCI target abort

0 = No interrupt occurred
1 = Interrupt occurred due to PCI target abort

PCI master abort

0 = No interrupt occurred
1 = Interrupt occurred due to PCI master abort

DMA read address reach end of buffer address

0 = No interrupt occurred
1 = Interrupt occurred due to the DMA read address reaching the end of buffer address

DMA read address reach interrupt address

0 = No interrupt occurred
1 = Interrupt occurred due to the DMA read address reaching the interrupt address

DMA write address reach interrupt address

0 = No interrupt occurred
1 = Interrupt occurred due to the DMA write address reaching the interrupt address

DMA write address reach end of buffer address

0 = No interrupt occurred
1 = Interrupt occurred due to the DMA write address reaching the end of buffer address

Interrupt 1 status 0x07

Bit	7	6	5	4	3	2	1	0
	AUX7	AUX6	AUX5	AUX4	AUX3	AUX2	AUX1	AUX0

Type: RO
Default: AUX pins current value

DMA registers

DMA write starting address register 0x08 – 0x0B

Type: R/W
Default: Undetermined.
The 32bit starting address for DMA read from main memory and write to the IOM2 port

DMA write interrupt address register 0x0C – 0x0F

Type: R/W
Default: Undetermined.
The 32bit address of the interrupt to be generated during DMA read from main memory and write to the IOM2 port

DMA write end address register 0x10 – 0x13

Type: R/W
Default: Undetermined.
The 32bit ending address for DMA read from main memory and write to the IOM2 port

Current DMA write address register 0x14 – 0x17

Type: RO
Default: Undetermined.
The 32bit address at which the write DMA process is currently at.

DMA read starting address register 0x18 – 0x1B

Type: R/W
Default: Undetermined.
The 32bit starting address for DMA read from the IOM2 port and write to main memory

DMA read interrupt address register 0x1C – 0x1F

Type: R/W
Default: Undetermined.
The 32bit address of the interrupt to be generated during DMA read from the IOM2 port and write to main memory

DMA read end address register 0x20 – 0x23

Type: R/W
Default: Undetermined.
The 32bit end address for DMA read from the IOM2 port and write to main memory

Current DMA read address register 0x24 – 0x27

Type: RO
Default: Undetermined.
The 32bit address at which the DMA read process is currently at.

AUX control registers

AUX mask register 0x29

Type: R/W
Default: 00h
Set to 1 to select the AUX input to be used to enable PME# output. Enable an external event (such as ring) to switch on computer.

AUX polarity control register 0x2A

Type: R/W
Default: 00h
Set to 1 to invert the polarity of AUX pin data.

AUX select register 0x2B

Bit	7	6	5	4	3	2	1	0
	Set watchdog timer[23:16]	Set watchdog timer[15:8]	Set wahtchdog timer[7:0]	DMA watchdog overflow	Enable DMA watchdog timer	AUX[0] function selection	AUX[7] function selection	AUX[5:4] function selection

Type : R/W, bit 4 R/W1TC
Default: 00h

Bit 0 set to 1, AUX[5:4] becomes power state bit (reference PCI config register 44h[1:0])
Bit 1 set to 1, AUX[7] becomes FSC2
Bit 2 set to 1, AUX[0] becomes crystal clock output CKO
Bit 3 set to 1, enable DMA watchdog timer
Bit 4 Watchdog timer status. Set to 1 when watchdog timer expires. Writing 1 will clear this bit.
Bit 5 set to 1 will set watchdog timer bit [7:0] to all 1's.
Bit 6 set to 1 will set watchdog timer bit [15:8] to all 1's.
Bit 7 set to 1 will set watchdog timer bit [23:16] to all 1's.

Second Serial Port Registers

Second Frame delay count 0x2C

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	FSC_D5	FSC_D4	FSC_D3	FSC_D2	FSC_D1	FSC_D0

Type: R/W
Default: 00h

Bit 5:0, FSC_D[5:0]
FSC2 delay count

0x00 2 cycle delay from FSC
0x1F 34 cycle delay from FSC

Serial port Control Register 0x2D

Bit	7	6	5	4	3	2	1	0
	Serial port shift-out direction	Serial port shift-in direction	Reserved	Reserved	Invert DCLK	Clock width selection bit 1	Clock width selection bit 0	Double Serial port clock DCLK

Type : R/W
Default: 00h

Bit 0, 2XDCLK
When set to 1, doubles the DCLK internally. Reset to 0 during power up.

Bit [2:1], DCLK_width
Control the internal double clk DCLK clock width.
00 : 10ns
01 : 20ns
10 : 30ns
11 : 40ns

Bit 3. Invert DCLK
When set to 1, invert the DCLK(After clock doubler) to the serial port block.

Bit 6, Serial port shift-in direction
0: LSB first
1: MSB first

Bit 7, Serial port shift-out direction
0: LSB first
1: MSB first

Second Frame low count 0x2E

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	FSC_LD5	FSC_LD4	FSC_LD3	FSC_LD2	FSC_LD1	FSC_LD0

Type: R/W
Default: 00h

Bit 5:0, FSC_LD[5:0]
FSC2 low count

0x00 2 cycle low time
0x1F 34 cycle low time

Frame Sync (FSC) delay count 0x2F

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	FSC_DC5	FSC_DC4	FSC_DC3	FSC_DC2	FSC_DC1	FSC_DC0

Type: R/W
Default: 00h

Bit 5:0, FSC_DC[5:0]
FSC delay count

PCI Configuration Space Description

Chip Device and Vendor ID offset 0x00

Bit	31	30	29	28	27	26	25	24
	0	1	0	1	0	1	1	0

Bit	23	22	21	20	19	18	17	16
	0	0	0	0	1	0	0	0

Bit	15	14	13	12	11	10	9	8
	0	0	0	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	0	1	0	1	0	1	1	1

Type: RO,
00h = 0xE159h, PCI ID number
02h = 0x0001h, device ID

Status and command offset 0x04

Bit	31	30	29	28	27	26	25	24
	Detected Parity Error	Signaled System Error	Master abort	Target abort	0	Medium device 1	Medium device 0	Data Parity Error Detected

Bit	23	22	21	20	19	18	17	16
	0	0	0	New Caps (APC)	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	SERR# enable

Bit	7	6	5	4	3	2	1	0
	0	Parity Error Response	0	0	0	Master enable	Memory enable	I/O enable

Default: 0x02100000

Detected Parity Error, bit 31
Type: R/W1TC
0 = No Error
1 = Detect address parity error

Signaled System Error, bit 30
Type: R/W1TC
0 = No Error
1 = Asserted SERR#

Master abort, bit 29
Type: R/W1TC
0 = Normal operation
1 = Transaction terminated with a Master-Abort

Target abort, bit 28
Type: R/W1TC
0 = Normal operation
1 = Transaction terminated with a Target-Abort

Medium device, bit [26:25]
Type: RO
01 = Medium speed to assert DEVSEL#

Data Parity Error Detected, bit 24
Type: R/W1TC
0 = No Error
1 = Detect data parity error

New Capabilities, bit 20
Type: RO
1 = Support new capability

SERR# enable, bit 8
Type: RW
0 = Disable SERR# generation
1 = Enable SERR# generation

New Capabilities, bit 6
Type: RW
0 = Disable Parity detection and response
1 = Enable Parity detection and response

Master enable
Type: RW
0 = Disables Bus mastering
1 = Enable Bus mastering
Note: set to 0 (disable) on reset

Memory enable
Type: RW
0 = Disables response to Memory Space accesses
1 = Enables response to Memory Space accesses
Note: set to 0 (disable) on reset

I/O enable
Type: RW
0 = Disables response to I/O Space accesses
1 = Enables response to I/O Space accesses
Note: set to 0 (disable) on reset

Class code and revision ID offset 0x08

Bit	31	30	29	28	27	26	25	24
	0	0	0	0	0	1	1	1

Bit	23	22	21	20	19	18	17	16
	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Class code
Type: RO
Subsystem ID [1]
0: 028000h = PCI network device
1: 078000h = PCI communication device

Revision ID
Type: RO
0 = revision 0

Header type and Latency timer offset 0x0C

Bit	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	LatTim7	LatTim6	LatTim5	LatTim4	LatTim3	LatTim2	LatTim1	LatTim0

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Default: 0x00000000

Header type
Type: RO
00h = Defines the layout of the predefined header

LatTim [0:7]
Type: RW
Sets the value in PCI clocks, for the latency timer of the PCI bus master.
Note: set to 0 on reset

I/O Base address 0x10

Bit	31	30	29	28	27	26	25	24
	I/Obase 31	I/Obase 30	I/Obase 29	I/Obase 28	I/Obase 27	I/Obase 26	I/Obase 25	I/Obase 24

Bit	23	22	21	20	19	18	17	16
	I/Obase 23	I/Obase 22	I/Obase 21	I/Obase 20	I/Obase 19	I/Obase 18	I/Obase 17	I/Obase 16

Bit	15	14	13	12	11	10	9	8
	I/Obase 15	I/Obase 14	I/Obase 13	I/Obase 12	I/Obase 11	I/Obase 10	I/Obase 9	I/Obase 8

Bit	7	6	5	4	3	2	1	0
	I/Osize	I/Osize	I/Osize	I/Osize	I/Osize	I/Osize	I/Osize	I/Ospace

Default: 0x00000001

I/Obase [31 : 8]
Type: RW
The I/O base address at which the device is located, set by the BIOS
Note: register access can either be by I/O or memory

I/Osize
Type: RO
00h, sets the I/O space to 256 bytes.

I/Ospace
Type: RO
1, requests an I/O space allocation from the BIOS.

Memory Base address 0x14

Bit	31	30	29	28	27	26	25	24
	Mbase 31	Mbase 30	Mbase 29	Mbase 28	Mbase 27	Mbase 26	Mbase 25	Mbase 24

Bit	23	22	21	20	19	18	17	16
	Mbase 23	Mbase 22	Mbase 21	Mbase 20	Mbase 19	Mbase 18	Mbase 17	Mbase 16

Bit	15	14	13	12	11	10	9	8
	Mbase 15	Mbase 14	Mbase 13	Mbase 12	Msize	Msize	Msize	Msize

Bit	7	6	5	4	3	2	1	0
	Msize	Msize	Msize	Msize	Msize	Msize	Msize	Mspace

Default: 0x00000000

Mbase [31 : 12]
Type: RW

The memory base address at which the device is located, set by the BIOS
Note: register access can either be by I/O or memory

Msize
Type: RO

00h, sets the memory space to 4K bytes.

Mspace
Type: RO

0, requests a memory space allocation from the BIOS.

Subsystem and Subsystem vendor ID 0x2C

Bit	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	SubSys1	SubSys0

Bit	15	14	13	12	11	10	9	8
	SubSysV-15	SubSysV-14	SubSysV-13	SubSysV-12	SubSysV-11	SubSysV-10	SubSysV9	SubSysV8

Bit	7	6	5	4	3	2	1	0
	SubSysV7	SubSysV6	SubSysV5	SubSysV4	SubSysV3	SubSysV2	SubSysV1	SubSysV0

SubSys [15 : 0]
Type: RO
Subsystem ID: From, {AUX[6], AUX[1:0], WRITE#, READ#, EXTRST#,HA[3:2], HD[7:0]}

SubSysV [1 : 0]
Type: RO
Subsystem Vendor ID: From HA[1:0]

Capabilities Pointer [7:0] 0x34

Bit	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	0	1	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

Type: RO, Default value = 0x40

Interrupts 0x3C

Bit	31	30	29	28	27	26	25	24
	1	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	INTA#

Bit	7	6	5	4	3	2	1	0
	IntLn7	IntLn6	IntLn5	IntLn4	IntLn3	IntLn2	IntLn1	IntLn0

Default: 0x80010100

Maximum Latency:
Type: RO
Value = 0x80:

Minimum Grant:
Type : RO
Value = 0x01

INTA#
Type: RO
1, Indicates that the interrupt pin used is INTA#.

Interrupt Line [7 : 0]
Type; RW
Used by POST software for interrupt line routing information.

PMC-Power Management Capabilities (APC) 0x40

Bit	31	30	29	28	27	26	25	24
	PME support D3cold	PME support D3hot	PME support D2	PME support D1	PME support D0	D2 Support	D1 Support	Aux current Bit 8

Bit	23	22	21	20	19	18	17	16
	Aux current Bit 7	Aux current Bit 6	Device Specific	0	PME clock	Version Bit 2	Version Bit 1	Version Bit 0

Bit	15	14	13	12	11	10	9	8
	Next Item	Next Item	Next Item	Next Item	Next Item	Next Item	Next Item	Next Item

Bit	7	6	5	4	3	2	1	0
	CapID bit 7	CapID bit 6	CapID bit 5	CapID bit 4	CapID bit 3	CapID bit 2	CapID bit 1	CapID bit 0

Default : If VAUX is 0, default value is 0x6C220001, if VAUX is 1, default value is 0xEC620001

PME support, bit [31:27]
Type: RO
If VAUX is 0, the value is 01101b: Support D3hot, D2 and D0.
If VAUX is 1, the value is 11101b: Support D3cold, D3hot, D2 and D0

D2 Support, bit 26
Type: RO
1: support D2

D1 Support, bit 25
Type: RO
0: not support D1

Aux_current, bit [24:22]
Type: RO
If VAUX is 0, the value is 000b. If VAUX is 1, the value is 001b.

Device Specific, bit 21
Type: RO
1: require a device specific initialization sequence.

PME clock, bit 19
Type: RO

0: no PCI clock for PME# generation.

Version, bit [18:16]
Type: RO
010b: Compiles with Revision 1.1 of PCI Power Management Interface Specification.

Next Item, bit [15:8]
Type: RO
0h: No next Item

Capability ID, bit [7:0]
Type: RO
01h: List item as PCI power Management registers.

Data register, PMCSR(APC) 0x44

Bit	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PME Status	0	0	0	0	0	0	PME enable

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	Power State Bit 1	Power State Bit 0

Default: 0x00000000

PME status, bit 15
Type: R/W1TC
If VAUX is 0, this bit is reset as 0 and set when assert PME# signal.
If VAUX is 1, this bit is sticky bit and set when assert PME# signal.

PME enable, bit 8
Type: RW
If VAUX is 0, this bit is reset as 0. If VAUX is 1, this bit is sticky bit. Set to 1 to enable the PME# generation.

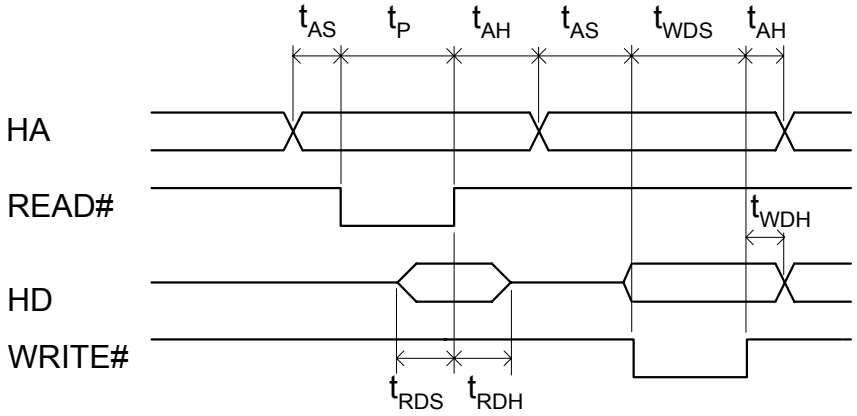
Power State, bit [1:0]
Type: RW
If VAUX is 0, these bits are reset to 0. If VAUX is 1, these bits are sticky.
Current power state. Only support D0, D2 and D3. The definition of the field values is given below.

00b ---- D0
10b ---- D2
11b ---- D3

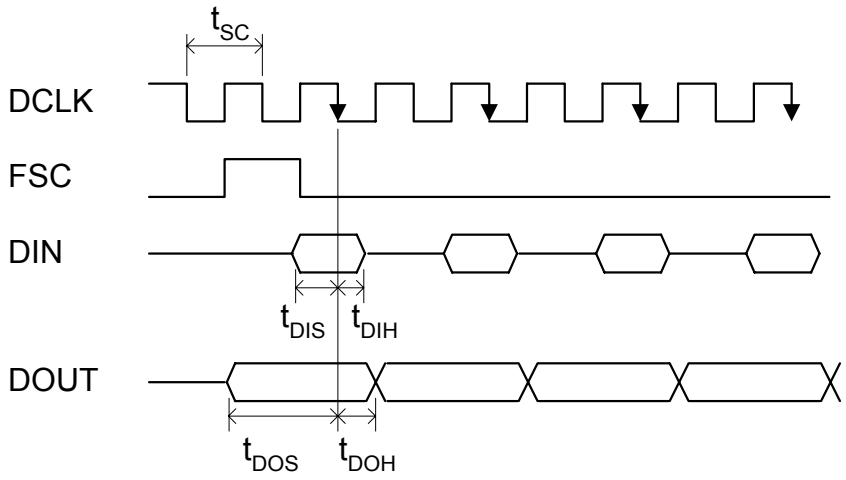
Write with 01b will have no effect.

Waveforms

PIB waveforms



Serial bus waveforms



A.C. Characteristics

PIB timing

Symbol	Parameter	Min(nS)	Max(nS)
t_{AS}	Address setup time	120	
t_{AH}	Address hold time	40	
t_{P}	Command pulse width	80	640
t_{RDS}	Read data setup time	10	
t_{RDH}	Read data hold time	5	
t_{WDS}	Write data setup time	80	640
t_{WDH}	Write data hold time	40	

Serial bus timing

Symbol	Parameter	Min(nS)	Max(nS)
t_{SC}	Serial clock period	500	
t_{DIS}	Data input setup time	10	
t_{DIH}	Data input hold time	5	
t_{DOS}	Data output setup time	½ DCLK	1 DCLK
t_{DOH}	Data output hold time	50	½ DCLK

D.C. characteristics

Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
Vcc	Power Supply	-0.3 to 6.0	V
Vin	Input Voltage	-0.3 to 6.0	V
Vout	Output Voltage	-0.3 to 6.0	V
T_{STG}	Storage Temperature	-40 to 125	°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability.

Operating ranges

Symbol	Parameter	Min	Typ	Max	Units
Vcc	Power Supply	3.0		5.5	V
V_{IN}	Input Voltage	0		5.5	V
T_{OPR}	Operating Temperature	0		70	°C
Symbol	Parameter	Min	Typ	Max	Units
VIL	Input Low Voltage			0.8	V
VIH	Input Voltage	2.2			V
VOL	Output Low Voltage, IOL= 4mA			0.4	V
V_{OH}	Output High Voltage, IOH = 4mA	3.5			V
RI	Pull-up / Pull-down resistors		50		KΩ

Typical Application Schematics

Please visit www.tjnet.com for schematics.

Physical Dimensions

Please visit www.tjnet.com for a 100pin pqfp package drawing.

Legal Words

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

TigerJet and TJNET are registered trademarks of TigerJet Network Inc.

How to reach TigerJet

TigerJet Network Inc.
50 Airport Parkway
San Jose
CA 95110
USA

E-mail: tiger@tjnet.com
www.tjnet.com