ADVANCE INFORMATION

TIGER560

USB Controller for Low Cost VoIP solutions

Includes SLIC and Codec interfaces and support circuitry to enable high integration/low cost telephone and handset to USB interfaces

Implements the Microsoft interface for USB audio devices and is able to use the standard Microsoft driver included with Windows

Features

- Performs all required USB interfacing and control functions for interfacing a regular phone or phone handset to the PC via USB.
- Audio functions
  - USB audio class device mode
  - Able to use Microsoft audio USB driver
  - 8 bit μ-Law CODEC interface
  - μ-Law to PCM16 translation
  - Record volume control
  - Playback volume control
  - Automatic audio mute
- PCM interface, support for;
  - Silicon Labs Si3211 ProSLIC
  - MC145480 audio codec
  - Many popular codecs/SLICs
  - Master/Slave operation
  - TDM, IOM2, GCI
  - Short and long frames
  - Multiple configuration options
- USB interface
  - Full speed 12MBps USB node
  - On chip USB transceiver
  - Digital PLL
  - Physical Layer Interface (PHY)
  - Media access controller (MAC)
  - Bus or self powered
  - Suspend/resume supported
  - USB specification 1.1 compliant
  - On chip 3.3V regulator
- SPI uP interface Bus
  - 4-wire interface
  - Byte serial data transfer
- Choice of USB descriptor tables
  - Audio device class
  - EPROM download
  - Legacy Tiger500
- Peripheral Interface Bus (PIB)
  - Easy connection of most popular peripheral chips
  - Byte-wide data
  - 6 address lines
  - 7 general purpose control lines
  - Read control signal
  - Write control signal
  - Reset control signal
- 4Mbit parallel interface
  - 8bit bi-directional port
  - ISO transfer
  - Up to 4Mbit
- Device features
  - Single 12MHz crystal oscillator
  - 5V operation
  - Onboard 3.3V regulator
  - 100 pin PQFP package
  - 0.65mm lead pitch

General Description

VoIP (Internet Phone) applications are becoming increasing popular as VoIP can provide free or low-cost calling worldwide. Early users of VoIP have attempted to use the existing PC sound card and have suffered poor call quality due to echoes and other problems.

To provide a VoIP experience that is the same as using a regular phone and eliminate the poor call quality that results from using the PC sound card, TigerJet has developed the Tiger560 USB controller that enables a regular phone, handset or headset to be interfaced to the USB port on the PC. With the Tiger560 OEMs can quickly bring to market a family of low cost high quality VoIP products.

The Tiger560 USB registers are compatible with the Microsoft USB audio driver and the Tiger560 is able to use the Microsoft written and supported driver directly. For many audio applications such as phone handsets or headsets, no additional software drivers are required. By using the standard Microsoft USB audio driver all popular Internet Telephony applications are supported and users will be able to make free or low cost calls with the same user experience as using a regular phone.

For phone handset or headset to USB applications, a low cost 8bit u-law “telephone” codec can be directly interfaced to the Tiger560 to provide a very highly integrated and low cost solution. For this type of product only the Microsoft USB audio driver is required.

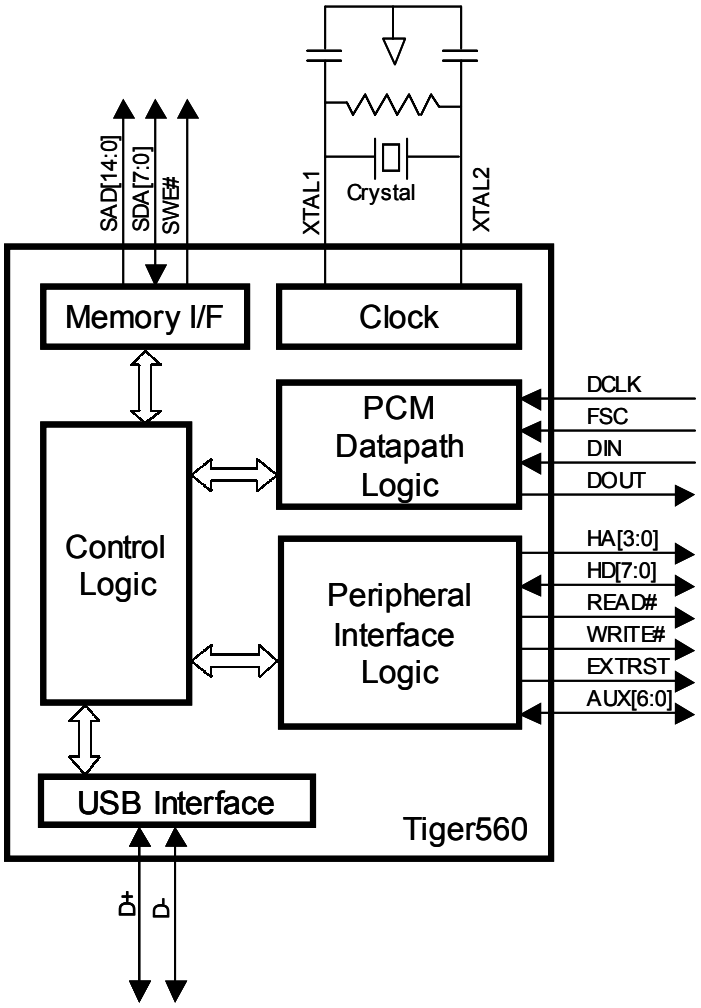
For regular phone to USB applications, a SLIC (subscriber line interface chip) can be interfaced to the Tiger560. The SLIC interfaces to the phone via the standard RJ-11 connector. The Tiger560 provides the USB interfacing and control. Reference drivers are available that enable the phone to ring, provide on hook/off hook detection and implement DTMF decode.

The Tiger560 incorporates an SPI interface, TDM/IOM2/GCI serial interface and a peripheral Interface Bus (PIB). The PIB has byte wide data, 6 address lines, 8 configurable control lines and read and write lines.

Ordering information

The order code for the Tiger560 is Tiger560.

Functional Block Diagram



Typical Implementation

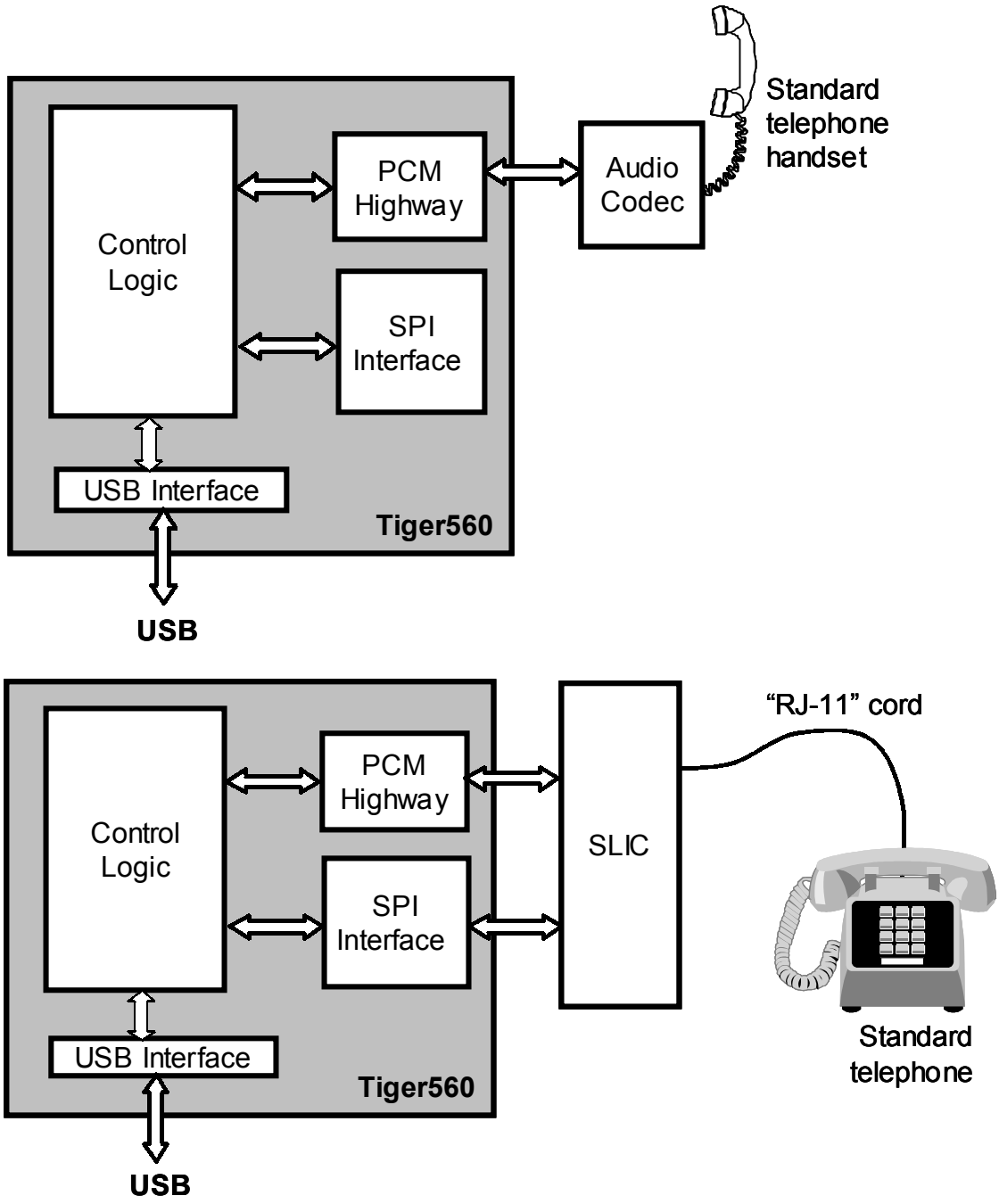
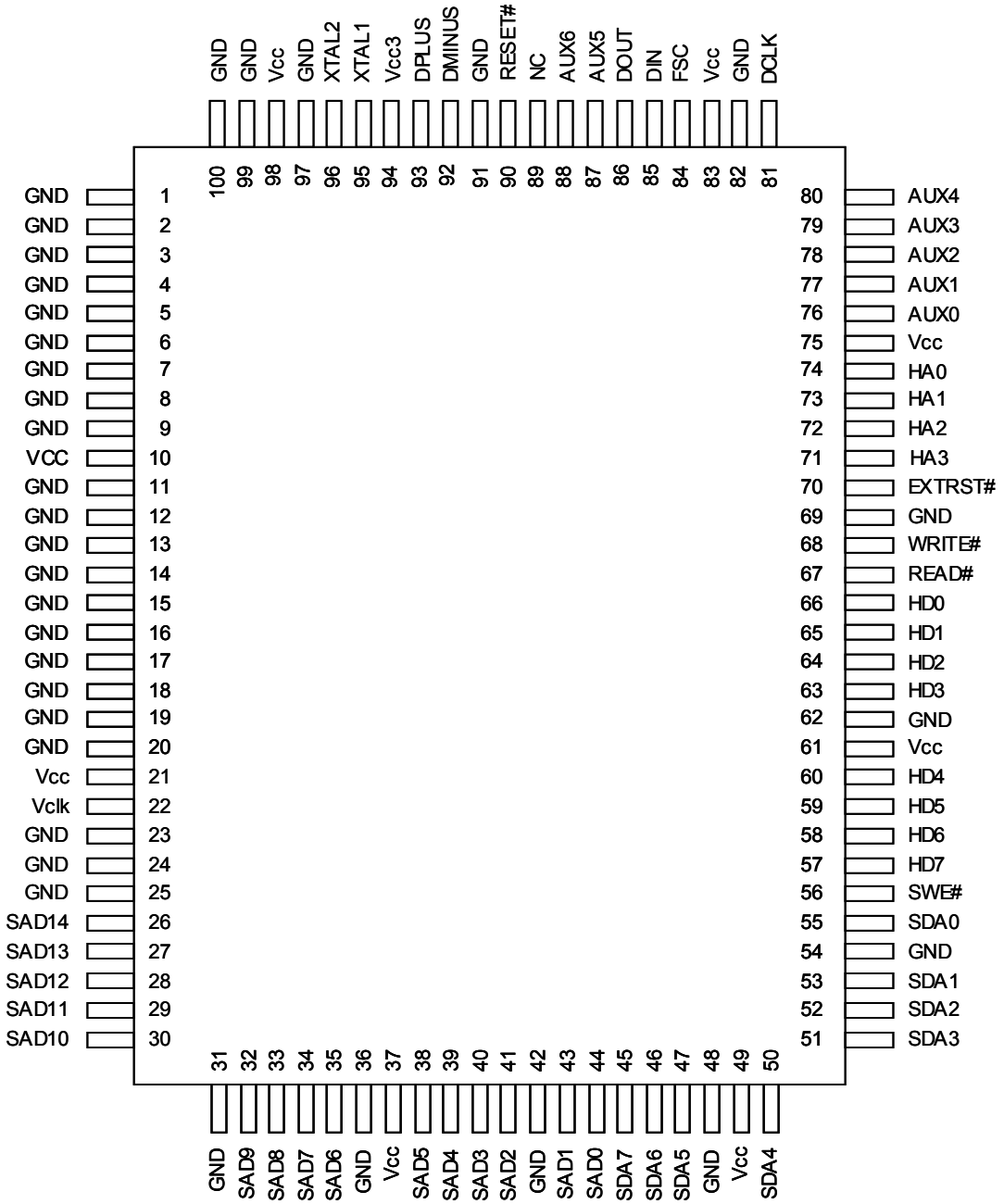


Table of Contents

Pin-out

Pin-out diagram



Pin assignment by pin number

Pin	Name
1	GND
2	GND
3	GND
4	GND
5	GND
6	GND
7	GND
8	GND
9	GND
10	$V_{CC}5$
11	GND
12	GND
13	GND
14	GND
15	GND
16	GND
17	GND
18	GND
19	GND
20	GND
21	$V_{CC}5$
22	VCLK
23	VS
24	GND
25	GND
26	SAD14
27	SAD13
28	SAD12
29	SAD11
30	SAD10
31	GND
32	SAD9
33	SAD8
34	SAD7

Pin	Name
35	SAD6
36	GND
37	$V_{CC}5$
38	SAD5
39	SAD4
40	SAD3
41	SAD2
42	GND
43	SAD1
44	SAD0
45	SDA7
46	SDA6
47	SDA5
48	GND
49	$V_{CC}5$
50	SDA4
51	SDA3
52	SDA2
53	SDA1
54	GND
55	SDA0
56	SWE#
57	HD7
58	HD6
59	HD5
60	HD4
61	$V_{CC}5$
62	GND
63	HD3
64	HD2
65	HD1
66	HD0
67	READ#

Pin	Name
68	WRITE#
69	GND
70	EXTRST#
71	HA3
72	HA2
73	HA1
74	HA0
75	$V_{CC}5$
76	AUX0
77	AUX1
78	AUX2
79	AUX3
80	AUX4
81	DCLK
82	GND
83	$V_{CC}5$
84	FSC
85	DIN
86	DOUT
87	AUX5
88	AUX6
89	NC
90	RESET#
91	GND
92	DMINUS
93	DPLUS
94	$V_{CC}3$
95	XTAL1
96	XTAL2
97	GND
98	$V_{CC}5$
99	GND
100	GND

Signal assignments by functional category

Ground pins (these pins to be grounded for minimum power consumption)	
Name	**Pin**
Gnd	19
Gnd	24
Gnd	99
Gnd	100
Gnd	1
Gnd	2
Gnd	3
Gnd	4
Gnd	6
Gnd	7
Gnd	23
Gnd	8
Gnd	9
Gnd	12
Gnd	13
Gnd	14
Gnd	16
Gnd	17
Gnd	18

Memory Interface	
Name	**Pin**
SAD0	44
SAD1	43
SAD10	30
SAD11	29
SAD12	28
SAD13	27
SAD14	26
SAD2	41
SAD3	40
SAD4	39
SAD5	38
SAD6	35
SAD7	34
SAD8	33
SAD9	32
SDA0	55
SDA1	53
SDA2	52
SDA3	51
SDA4	50
SDA5	47
SDA6	46
SDA7	45
SWE#	56
VCLK	22

Peripheral Interface Bus	
Name	**Pin**
AUX0	76
AUX1	77
AUX2	78
AUX3	79
AUX4	80
AUX5	87
AUX6	88
EXTRST#	70
HA0	74
HA1	73
HA2	72
HA3	71
HD0	66
HD1	65
HD2	64
HD3	63
HD4	60
HD5	59
HD6	58
HD7	57
READ#	67
WRITE#	68

Serial Ports	
Name	**Pin**
DCLK	81
DIN	85
DOUT	86
FSC	84

USB Ports	
Name	**Pin**
DMINUS	92
DPLUS	93

Control Signals	
Name	**Pin**
RESET#	90
XTAL1	95
XTAL2	96

Power and Ground	
Name	**Pin**
$V_{CC}5$	10, 21, 37, 49, 61, 75, 83, 98
$V_{CC}3$	94
GND	5, 11, 15, 20, 25, 31, 36, 42, 48, 54, 62, 69, 82, 91, 97

Signal descriptions

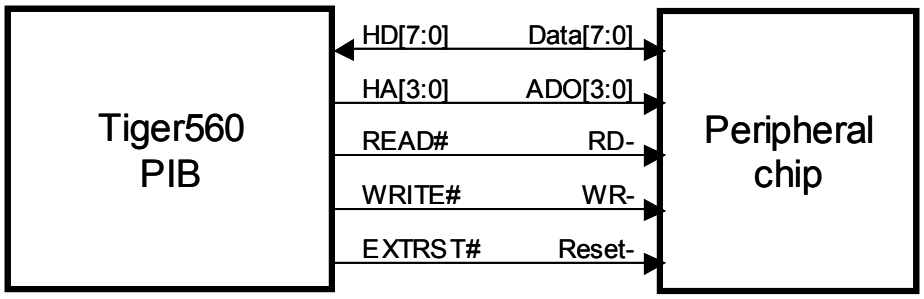
Signal Name	Type	Description	Alternate function(s)
AUX0	I/O	PIB aux port bit 0	PIB HA4 SPI interface CDIN
AUX1	I/O	PIB aux port bit 1	PIB HA5 SPI interface CDOUT
AUX2	I/O	PIB aux port bit 2	USB suspend output
AUX3	I/O	PIB aux port bit 3	
AUX4	I/O	PIB aux port bit 4	
AUX5	I/O	PIB aux port bit 5	Audio feedback comparison output
AUX6	I/O	PIB aux port bit 6	ROMCS#
DCLK	I/O	Serial port data clock	
DIN	I	Serial port data input	
DMINUS	I/O	USB D-	
DOUT	I	Serial port data output	
DPLUS	I/O	USB D+	
EXTRST#	O	PIB reset	
FSC	I/O	Serial port frame sync	
HA0	I/O	PIB address 0	Product ID [15] input on reset
HA1	I/O	PIB address 1	Self/bus power input on reset
HA2	I/O	PIB address 2	Serial uP interface CSB
HA3	I/O	PIB address 3	Serial uP interface CCLK HA[3:2] : mod selection 00: Tiger500 mode 01: New Tiger560 USB DP table 10: Testing mode 11: Use external ROM for USB DP
HD0 – HD7	I/O	PIB data bus	Vendor ID [8 – 15] input on reset
READ#	O	PIB read	
RESET#	I	System reset input	
SAD0 – SAD12	O	Memory address bus	Product ID [0 – 12] input on reset SAD0 -10 for ROMAD0 -10
SAD13	O	Memory address bus	Product ID [13] input on reset 4Mbit parallel port WR#
SAD14	O	Memory address bus	Product ID [13 – 14] input on reset 4Mbit parallel port RD#
SDA0 – SDA 7	I/O	Memory data bus	Vendor ID [0 – 7] input on reset 4Mbit parallel port data bus ROMDA0 - 7
$V_{CC}3$	O	3.3V output for USB port	
VCLK	I/O	Memory clock	
WRITE#	O	PIB write	
XTAL1	I	Crystal oscillator	
XTAL2	O	Crystal oscillator	

Functional description

Peripheral Interface Bus (PIB)

To enable a “glueless” interface to most popular peripheral chips, the Tiger560 implements a Peripheral Interface Bus (PIB). The PIB consists of a 6 bit address bus, HA[5:0], 8 bit data bus, HD[7:0], READ#, WRITE#, EXTRST# (external reset) and 8 AUX lines.

Typical connection of a Peripheral using the Tiger560 PIB



All of the address, data and control lines are fully qualified and can be connected without any additional glue logic to a wide range of peripheral chips.

AUX lines

AUX[6:0] can be individually programmed as inputs or outputs. Register 0x13 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 defines the appropriate line as input. On hard reset all AUX lines float and are defined as inputs.

The status of the AUX lines can be read from register 0x12. The actual AUX line value will be read irrespective of it being an input or output.

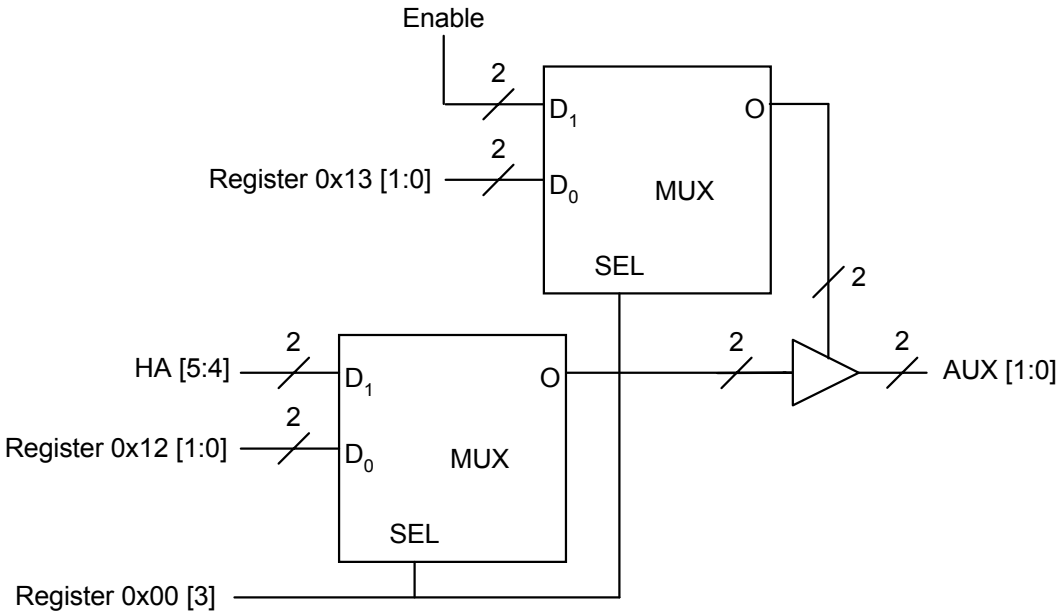
AUX line inputs can be inverted individually with register 0x15. This is useful for determining the active polarity of the AUX line when used for Wake-up. This register does not change signal polarity when the AUX lines are used for Interrupt or Suspend.
Please note: on the current silicon revision of the Tiger560 AUX3 and AUX4 must both be either an input or an output, please contact TigerJet if you need further clarification.

Dual function AUX lines

Register 0x00 enables individual enabling of HA[5:4] and Suspend. Register 0x29 bit 5 enables SPI pin definition. These functions are described below in turn.

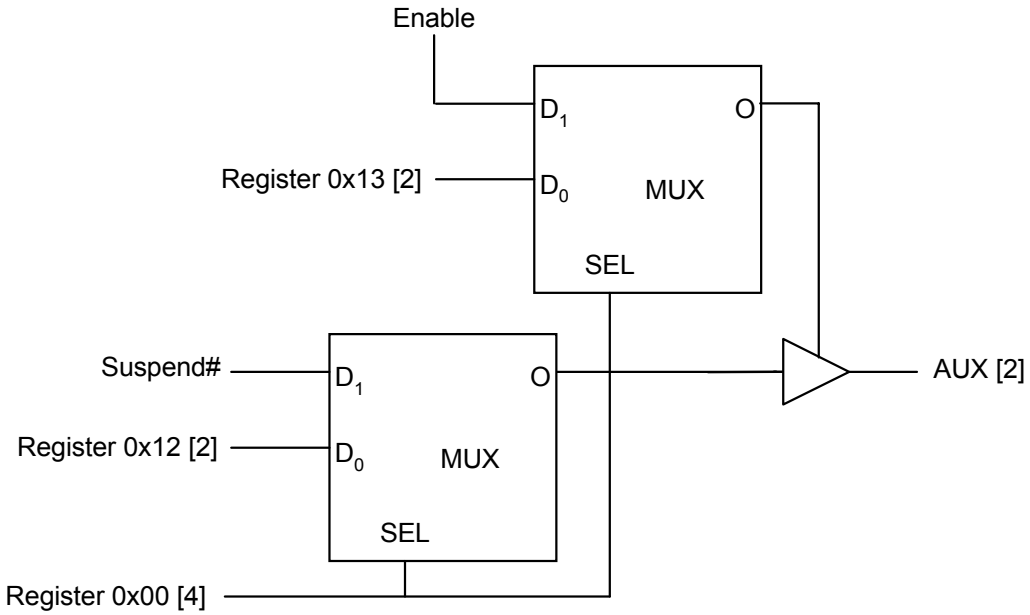
Address lines HA4 and HA5

AUX pins 1 and 0 can be set as two additional address lines, HA[5:4], to increase the address space. The control logic is detailed below.

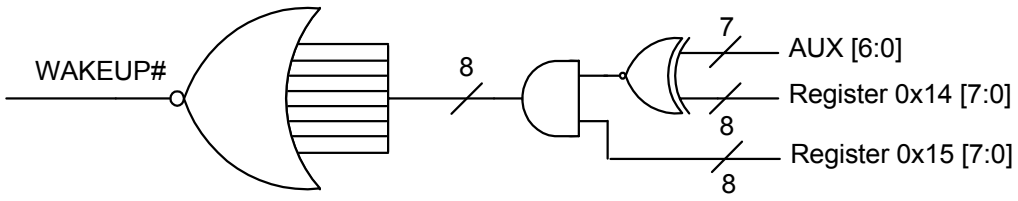


Suspend#

The control logic for switching between AUX line 2 and Suspend# is detailed below.



Wake up logic



The wake up signal can be generated from any of the AUX lines, Register 0x15 selects the AUX lines to be monitored, writing a 1 to the appropriate register location will enable monitoring of a appropriate AUX line. Register 0x14 selects the polarity of the AUX input, a 1 will invert the incoming signal and result in wake up on a high input.

USB Interface

The Tiger560 implements a fully compliant USB 1.1 interface. The USB line drivers and 3.3 volt regulator are included on chip and the Tiger560 can be connected directly to the USB bus.

Chip mode selection
HA[3:2] reset latched value determines the operating mode of the Tiger560.

HA[3:2]	Chip mode
00	USB device descriptor table is the same as Tiger500
01	USB device descriptor table with AUDIO class device descriptor table. Product ID [15:2] with chip operation setting
10	Test mode
11	USB device descriptor table from external ROM. ROMCs# from AUX[6] ROMAD[9:0] = SAD[9:0] ROMDA[7:0] = SDA[9:0]

USB vendor and product I.D.s

The USB vendor and product I.D.s are set by pull-up and pull down resistors on the following pins.

Vendor I.D. bits 0 – 7 on SRAM data bus SDA0 – SDA7
Vendor I.D. bits 8 – 15 on PIB data bus HD0 – HD7
Product I.D. bits 0 – 14 on SRAM address bus SAD0 – SDA14
Product I.D. bit 15 on PIB address HA0

On reset the inputs are read into internal registers, it is important to ensure that peripheral devices connected to these inputs tri-state their outputs on reset.

ProductID[15:14]	00: Video only USB device descriptor table 01: Audio only USB device descriptor table 10: Video + Audio USB device descriptor table
ProductID[13:12]	00: max power 500 ma 01: max power 400 ma 10: max power 300 ma 11: max power 250 ma
ProductID[11]	0: No USB wake up function support 1: Support USB wake up function
ProductID[10]	0: Audio data from serial port audio codec channel number 0 1: Audio data from serial port audio codec channel number 1
ProductID[9]	0: Audio data shift direction MSB first 1: Audio data shift direction LSB first
ProductID[8]	0: Normal Audio data clock input 1: Inverted Audio data clock input
ProductID[7:6]	00: no change 01: Double Audio data clock input with 30ns delay 10: Double Audio data clock input with 40ns delay 11: Double Audio data clock input with 50ns delay
ProductID[5]	0: AUX[2]/Suspend# output low active 1: AUX[2]/Suspend# output high active
ProductID[4]	0: Disable internal PLL, use 48Mhz crystal 1: Enable internal PLL, use 12Mhz crystal
ProductID[3]	0: Normal operation 1: Force serial port clock DCLK as input pad
ProductID[2]	0: Normal operation 1: Force serial port FSC as input pad

USB descriptor table – EPROM download

The Tiger560 can be programmed to download the USB descriptor table from EPROM

If Tiger560 set to use the descriptor table from EPROM, the product ID[15:2] is set to program the internal function setting.

Vendor Commands

The Tiger560 uses the USB vendor command to access the internal registers and parallel port interface. For the most efficient transfer of data, the Tiger560 allows for multiple transfers in a single vendor command, in addition, multiple transfers can be to a single location or can increment with each transfer. For each vendor command that is issued the access timing can be defined.

An additional feature of the Tiger560 is that a vendor command can implement “check-before-doing”. The vendor command specifies the status source location and specifies the mask for the status value. Tiger560 will first read the status value from the specified location and AND it with the mask value. If the ANDed value is TRUE, the read operation will be performed.
The checking is repeated until the FAIL condition is detected. When a FAIL condition occurs the transfer will be halted and no more transfers will take place until next vendor command. The number of operations prior to the fail can be read from internal registers 0x05 and 0x06.

Vendor Command byte 0: Request-type

Bit 7 indicates the type of operation. If this bit is set to 1, this vendor command will read data from the Tiger560 to the host. If this bit is set to 0, the command is for write operation.
For the read operation, set this byte to 0xC0. For the write operation, set this byte to 0x40.

Vendor Command byte 1: Request

Bit 0 indicate the operation of address lines. When this bit is 0, internal address counter will automatically increase by one for the next data transfer. Setting this bit to 1 will force the address counter to keep the same value for all the data transfer until next vendor command.
Bits 2 and 1 specify the pulse width for the command. When set to 0, will have 2 cycles of command pulse. The timing is based on a 24 MHz system clock for about 42 ns per cycle. Value 1 of bit 2 and 1 has 3 cycle; set to 2 has 8 cycle and set to 3 has the longest pulse width for 16 cycles.
Setting bit 3 to 1 will activate the mask and status checking operation.

Vendor Command byte 3 and 2: Value

Byte 2 is the status address and byte 3 is the mask value. Tiger 560 will read the status value from the status address and AND it with byte 3 mask value for the status checking.

Vendor Command byte 5 and 4: Index

This is the address field for the vendor command. Byte 5 is not used because the Tiger560 only requires a 256 location address space. The address space 0x00 to 0xBF is used for the Tiger560’s internal registers. Address space 0xC0 to 0xFF is used for the PIB. When addresses above 0xC0 are accessed, the PIB READ# or WRITE# signals will be generated.

Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0x00	0	0							Internal register address space
to	0	1							
0xBF	1	0							
0xC0	1	1	HA5	HA4	HA3	HA2	HA1	HA0	PIB address space
to									
0xFF									

Vendor Command byte 7 and 6: Count

Byte 6 specifies the number of bytes to transfer. Byte 7 is not used. The Tiger560 can support up to 255 transfer operations. For the check-before-doing operation, this field indicates the maximum

number of transfers to be performed. The actual transfer count will be in register 0x05 and 0x06 depending the type of operation.

Interrupt Transfer

USB Endpoint 5 is used for the interrupt transfer on the Tiger560. The polling interval is 1ms. Each time polling occurs, 2 bytes of data will be transferred from the Tiger 560 to the host. The first byte of data is the value of AUX pins current state. The second byte is the value of specified source. Register 0x18 is defined as the location for this byte. Only the external PIB will be used for the polling. Setting bits 7 and 6 in register 0x18 to 1 will enable the PIB interrupt status polling. The polling operation will not conflict with any vendor command because the polling will be performed in every USB SOF (Start-Of-Frame) package. Byte one will show the current status of any interrupt line connected to the AUX pins and byte two will be the value of the interrupt status of the peripheral (external) device.

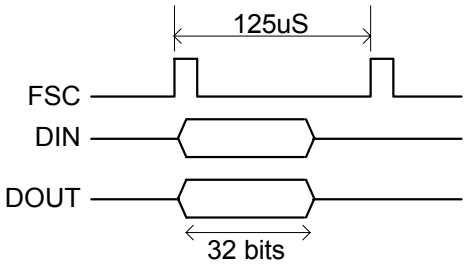
Serial Port Interface

The Tiger560 serial port consists of one data clock (DCLK), one frame synchronization clock (FSC) and two data lines (DIN and DOUT).
DCLK and FSC can either be inputs or outputs to the Tiger560. DCLK can either be the same as the data rate or twice the data rate. If DCLK is the same as the data rate the internal clock doubler should be turned on for correct operation.

The serial port interface is controlled by two sets of registers based on the Tiger560 chip mode selection.
If Tiger560 is set to use original Tiger500 USB descriptor mode(HA[3:2]==2'b00), the serial port interface is controlled by the same set of registers as Tiger500. If Tiger560 is set to use the Tiger560 USB descriptor (HA[3:2]==2'b01) and the audio class device is enabled (productID[15:14] == 2'b01 or 2'b10), the serial port interface is controlled by productID definition and a set of registers that is specific to the Tiger560. For detailed register definition please see the registers section.

Tiger500 mode: Serial port signals

The FSC input 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 pins can be forced to drive low by setting bit 3 in register 0x01. This option allows certain types of peripherals to activate the data clock.

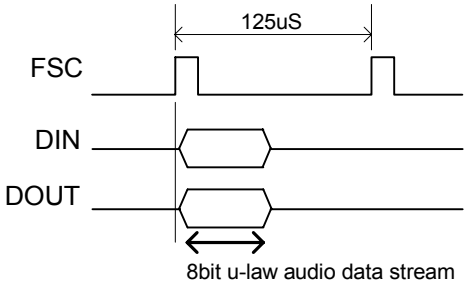
Tiger500 mode: Serial port data transfer

Tiger560 USB Endpoint 3 and Endpoint 4 are used for the serial port data transfer. Each pipe will transfer 37 bytes of data each USB isochronous transfer. Endpoint 3 will communicate the serial port data back to host and Endpoint 4 will transfer output data to serial port.

The serial port clock is asynchronous to the USB clock, so an adaptive endpoint design is used. Endpoint 3 provides the data rate information and feedback to the host. Host receives the data rate information and decides the transfer rate to Endpoint 4. The data rate information is in the byte one of the transferred data stream. Three possible value are defined, 32, 28 and 36. For 8 KHz of frame clock, every 1 ms, 32 bytes of data will be transferred. Endpoint 3 will adjust the transfer count with either 28 or 36 based on the difference between the serial port clock and USB clock.

The host is required to send the same amount of data to Endpoint 4 for the synchronization. Both endpoints have the same data format. The first byte is the number of valid data bytes. The following bytes are the valid data. Stuff bytes are attached at end of valid data so the total number of bytes is 37.

Tiger560 mode (Audio class device): Serial port signals
The FSC should be an 8Khz clock. Within one FSC period, 8 bits of μ-law audio data are transmitted and received. The channel/slot number can be programmed through productID[10]. When productID[10] is set to 0 the first 8bits after the frame sync is used, when productID[10] is set to 1 the second 8bits is used.



Tiger560 mode (Audio class device): Serial port data transfer
Tiger560 USB Endpoint 6 and Endpoint 7 are used for the audio data transfer. 16-bit PCM audio format is supported. Each USB isochronous transfer will carry 8 samples with 16 bytes of data. Endpoint 6 is for wave-out device and Endpoint 7 is for wave-in device.

The Tiger560 translates audio samples between 8bit μ-law and 16-bit PCM format. Each audio stream direction has a USB feature unit to control the volume. Tiger560 will perform hardware volume scaling based on the USB SET_CUR volume command. The mute control is supported for both the speaker and microphone audio streams.

Auto audio feedback comparison logic will monitor the input audio stream. When the value exceeds the set reference value, it will automatically mute the audio output stream to prevent audio feedback or echo. The reference value for the muting is controlled through internal registers 0x2f and 0x30.

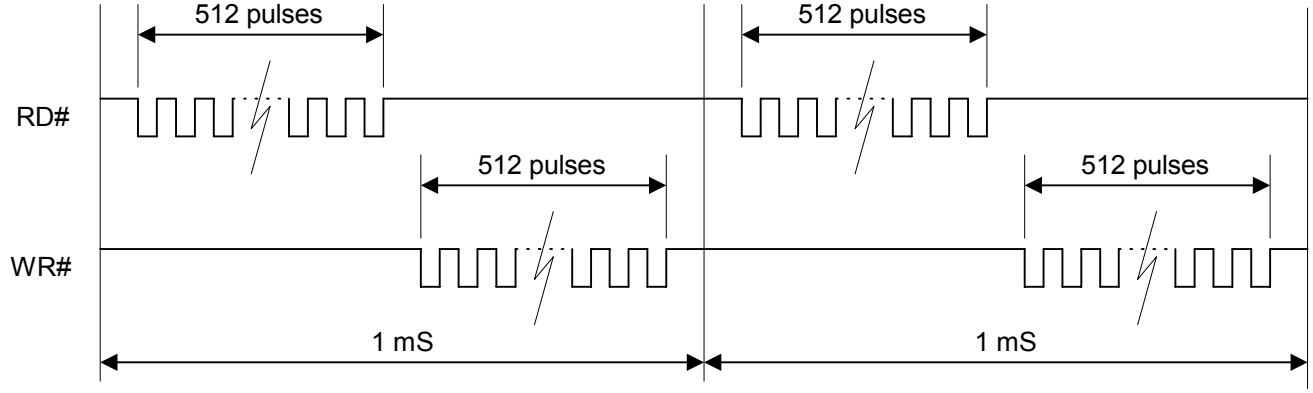
4Mbit Parallel Port

The Tiger560 provides a 4Mbit, 8-bit parallel port interface for support of external peripherals that require a high-speed USB interface using ISO transfer. Data can be transferred at 4Mbits/second in each direction.
The circuitry used is shared with the EPROM interface and the 4Mbit Parallel port can only be used in a system that does not implement the optional EPROM.

Interface signals

There are 8 bi-directional data lines and 3 control lines, RD#, WR# and VCLK.
The 11 interface lines are dual function shared pins. The data bus is shared with SDA[7:0], RD# is shared with SAD[14], WR# is shared with SAD[13].
VCLK should be configured as an output by setting 0x00 bit 7. When set as an output VCLK will produces a 48MHz reference clock.

Signal timing for the 4Mbit Parallel Port



1. In each 1mS period there will be 512 read pulses and 512 write pulses.
2. Either read or write will occur first, the determination is made by the USB hub.
3. The read and write cycles will not overlap.

VCLK is the 48MHz reference clock.

Software settings

Two software settings are required to enable the 4Mbit parallel port.

1. Set the alternate interface setting to 4. This will enable the first ISO endpoint as 512 byte of read operation and the second ISO endpoint as 512 byte of write operation.
2. Set the internal register index 7 bit 0 to 1 to enable the high speed ISO.
3. Set register 0x00 bit 7 to 1 to enable VCLK as an output

Data transfer

When the interface is enabled, Tiger560 will start to monitor the USB interface for any matching ISO package. When it detects a matching ISO package, it will generate a read or write cycle for each byte transfer. The endpoint is defined as a 512-byte interface, so there will be 512 cycles in each ISO package.

Each ISO package is a contiguous block of 512 bytes and will be completed in a 1ms interval. In each 1ms interval there is one ISO read and one ISO write. The order of read and write is scheduled by the USB hub controller and can be any combination and can change from one interval to the next.

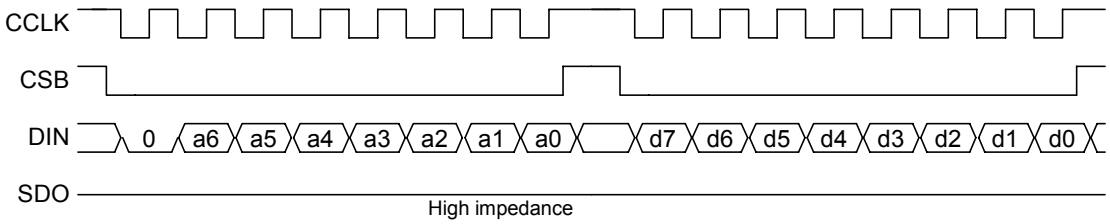
SPI micro-controller interface

To enable interface to many popular micro-controllers and serial peripheral devices such as SLICs etc, a 4-wire SPI interface has been implemented.

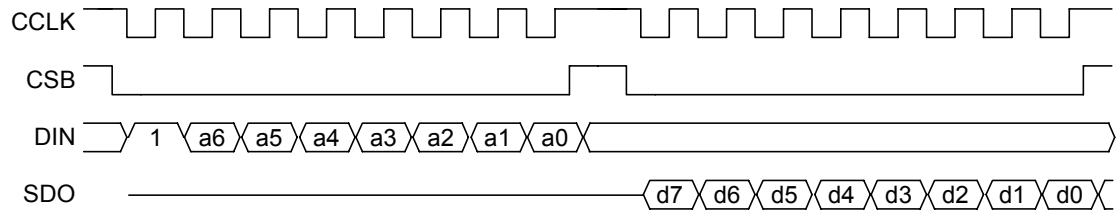
The interface consists of a clock signal (CCLK), chip select (CSB), data input (CDIN) and data output (CDOUT). These signals share the HA[3], HA[2], AUX[0] and AUX[1] pins. The SPI interface is enabled when bit 5 of register0x29 is set to 1.

Registers 0x26 to 0x29 are used to control the transfer of data. Each transfer can be either a 2-byte transfer or a 3-byte transfer. For many micro-controllers, the first byte of the transfer is the command/address byte. The second byte is the data read or write. Generally the MSB of the first byte indicates a read or write operation, 0 indicates a write and a 1 indicates a read.

The transfer can be programmed with a chip select (CBS) break between each byte transfer or without CSB break between each transfer. The speed and the phase of transfer clock also can be programmed through the Tiger560 register 0x29.



Serial write timing diagram



Serial read timing diagram

Registers

Register addressing

Tiger560 internal registers are accessed via USB end point. The internal registers are mapped from address 0x00 to 0x30. The PIB address lines HA[5:0] are mapped from 0xc0 to 0xff.

General Control Register

General Control 0x00

Bit	7	6	5	4	3	2	1	0
	VCLK OE	USB reset enable	Reserved	Aux / Suspend	Aux / HA	IOM reset	Reserved	EXTRST#

Type; R/W
Default value: 40

VCLK OE

0 = Enable VCLK output (for 4Mbit Parallel Port)
1 = Disable VCLK output

USB reset enable (Not used in Audio class device mode)

0 = Enable USB reset
1 = Disable USB reset

Aux / Suspend

0 = AUX2 = AUX2
1 = AUX2 = SUSPEND#

Aux / HA

0 = AUX[1:0] = AUX[1:0]
1 = AUX[1:0] = HA[5:4]

IOM reset (Not used in Audio class device mode)

0 = Normal operation
1 = Reset IOM bus

EXTRST#

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

Serial Bus Registers

Serial Bus Control Register 0x01

Bit	7	6	5	4	3	2	1	0
	Serial CLK select	Serial CLK select	Reserved	Reserved	DOUT force low	Serial CLK polarity	Reserved	DOUT enable

Type; R/W
Default value: 00

Serial CLK select (Not used in Audio class device mode)

00 = Normal
01 = Clock doubler enabled, high period = 30nS
10 = Clock doubler enabled, high period = 60nS
11 = Clock doubler enabled, high period = 90nS

DOUT force low (Not used in Audio class device mode)

0 = Normal operation
1 = Force DOUT to low

Serial CLK polarity (No use in Audio class device mode)

0 = Normal
1 = Invert

DOUT enable (No use in Audio class device mode)

0 = Tristate DOUT
1 = Enable DOUT

Second Frame Sync Time Slot 0x02

Bit	7	6	5	4	3	2	1	0
	FSC master mode	Serial CLK Output Invert	Serial Master Enable	Reserved	Reserved	Reserved	Reserved	Reserved

Type: R/W
Default value: 00

FSC master mode source selection

0: from internal 48Mhz clock
1: from DCLK and divide by 256, DCLK needs to be 2.048Mhz.

Serial CLK Output Invert (Not used in audio class device mode)

Applies to TDM clock output in master mode only.
0 = Normal operation
1 = Invert TDM CLK

Serial Master Enable (Not used in audio class device mode)

0 = Enable Serial Master, Serial CLK and FSC pins are set as outputs.
1 = Disable Serial Master, Serial CLK and FSC pins set as inputs.

USB Vendor Command Registers

Status Write Counter 0x05

Bit	7	6	5	4	3	2	1	0
	SWC7	SWC6	SWC5	SWC4	SWC3	SWC2	SWC1	SWC0

Type; RO
Default value:

SWC[7:0]
Write transferred byte number

Status Read Counter 0x06

Bit	7	6	5	4	3	2	1	0
	SRC7	SRC6	SRC5	SRC4	SRC3	SRC2	SRC1	SRC0

Type; RO
Default value: 00

SRC[7:0]
Read transferred byte number

PIB Aux Port Control

PIB Aux Port Data 0x12

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

Type; R/W
Default value: 00

AuxD[6:0]
Write = Sets the state of Aux lines configured as outputs
Read = Reads the status of all Aux lines both inputs and outputs

Note: Aux[1:0] can be configured to function as address lines 4 and 5

PIB Aux Control 0x13

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

Type; R/W
Default value: 00

AuxC[6:0]
0 = Line configured as an input
1 = Line configured as an output

Note; On current Tiger560 silicon AuxC4 is not implemented, AUX4 is controlled by AuxC3.

PIB Aux Polarity 0x14

N.B. this register only changes polarity when used for wake up

Bit	7	6	5	4	3	2	1	0
	Reserved	AuxP6	AuxP5	AuxP4	AuxP3	AuxP2	AuxP1	AuxP0

Type; R/W
Default value: 00

AuxP[6:0]

0 = Normal operation
1 = Invert the signals on the Aux pins when used for wake up

PIB Wake up input 0x15

Bit	7	6	5	4	3	2	1	0
	Reserved	AuxW6	AuxW5	AuxW4	AuxW3	AuxW2	AuxW1	AuxW0

Type; R/W
Default value: 00

AuxW[6:0]

0 = Ignore input for wake up
1 = Select an input(s) to generate a wake up event

SRAM Delay 0x17

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Delay SRAM WR# 1	Delay SRAM WR# 0	Reserved	Reserved	Reserved	Reserved

Default value: 00

Delay SRAM WR# [1:0]

00 = Delay 8nS
01 = Delay 4nS
10 = Delay 12nS
11 = Delay 16nS

Interrupt Polling Address 0x18

Bit	7	6	5	4	3	2	1	0
	Int PE 1	Int PE 0	Int P Add 5	Int P Add 4	Int P Add 3	Int P Add 2	Int P Add 1	Int P Add 0

Type: R/W
Default value: 00

Int PE [1:0]

00 = Disable PIB polling
01 = Disable PIB polling
10 = Disable PIB polling
11 = Enable PIB polling

Int P Add [5:0]

Interrupt PIB Polling Address
Sets the address at which data from the PIB will be read into the interrupt pipe

Audio control register 0x1f

Bit	7	6	5	4	3	2	1	0
	Reserved	DOUT force low	Disable DOUT	Enable USB reset	Invert DCLK	TDM master mode	Reset TDM	Reserved

Type: R/W
Default value: 00

Reset TDM (only under Audio Class device mode)

0: normal operation
1: reset TDM block

TDM master mode (only under Audio Class device mode)

0: TDM clock in slave mode
1: TDM clock in master mode

Invert DCLK (only under Audio Class device mode)

0: normal operation
1: invert DCLK

Enable USB reset (only under Audio Class device mode)

0: disable USB reset
1: enable USB reset

Disable DOUT (only under Audio Class device mode)

0: normal operation
1: disable DOUT output

DOUT force low (only under Audio Class device mode)

0: normal operation
1: force DOUT to low

Microphone mute Control 0x20

Bit	7	6	5	4	3	2	1	0
	MPMC[7]	MPMC[6]	MPMC[5]	MPMC[4]	MPMC[3]	MPMC[2]	MPMC[1]	MPMC[0]

Type: R/W
Default value: 00

Microphone mute Control

0: normal audio
not equal to 0: mute microphone

Speaker Mute Control 0x21

Bit	7	6	5	4	3	2	1	0
	SMC[7]	SMC[6]	SMC[5]	SMC[4]	SMC[3]	SMC[2]	SMC[1]	SMC[0]

Type: R/W
Default value: 00

Speaker mute control

0: normal audio
not equal to 0: mute Speaker

Microphone volume low byte 0x22

Bit	7	6	5	4	3	2	1	0
	MPV[7]	MPV[6]	MPV[5]	MPV[4]	MPV[3]	MPV[2]	MPV[1]	MPV[0]

Microphone volume high byte 0x23

Bit	7	6	5	4	3	2	1	0
	MPV[15]	MPV[14]	MPV[13]	MPV[12]	MPV[11]	MPV[10]	MPV[9]	MPV[8]

Type: R/W
Default value: 0xF600

Microphone Volume value [15:0]

5 level Volume control
0x0000 ~ 0xFE00 : Scale up X4
0xFDFF ~ 0xFB00 : Scale up X2
0xFAFF ~ 0xF200 : No Scaling
0xF1FF ~ 0xD800 : Scale down X2
0xD7FF ~ 0x8000 : Scale down X4

Speaker Volume low byte 0x24

Bit	7	6	5	4	3	2	1	0
	SPV[7]	SPV[6]	SPV[5]	SPV[4]	SPV[3]	SPV[2]	SPV[1]	SPV[0]

Speaker Volume high byte 0x25

Bit	7	6	5	4	3	2	1	0
	SPV[15]	SPV[14]	SPV[13]	SPV[12]	SPV[11]	SPV[10]	SPV[9]	SPV[8]

Type: R/W
Default value: 0xD800

Speaker Volume value [15:0]

5 level Volume control
0x0000 ~ 0xFB00 : Scale up X4
0xFAFF ~ 0xE700 : Scale up X2
0xE6FF ~ 0xC900 : No Scaling
0xC8FF ~ 0xA200 : Scale down X2
0xA1FF ~ 0x8000 : Scale down X4

Serial uP interface first data 0x26

Bit	7	6	5	4	3	2	1	0
	SPI1B7	SPI1B6	SPI1B5	SPI1B4	SPI1B3	SPI1B2	SPI1B1	SPI1B0

Type: R/W
Default value: 00

SPI1B [7:0]

Write as first serial uP interface write data byte
Read ad first serial uP interface read data byte

Serial uP interface second data 0x27

Bit	7	6	5	4	3	2	1	0
	SPI2B7	SPI2B6	SPI2B5	SPI2B4	SPI2B3	SPI2B2	SPI2B1	SPI2B0

Type: R/W
Default value: 00

SPI2B [7:0]

Write as second serial uP interface write data byte
Read ad second serial uP interface read data byte

Serial uP interface third data 0x28

Bit	7	6	5	4	3	2	1	0
	SPI3B7	SPI3B6	SPI3B5	SPI3B4	SPI3B3	SPI3B2	SPI3B1	SPI3B0

Type: R/W
Default value: 00

SPI3B [7:0]

Write as third serial uP interface write data byte
Read ad third serial uP interface read data byte

Serial uP interface control register 0x29

Bit	7	6	5	4	3	2	1	0
	3 byte operation	Transfer mode	Enable serial uP pin definition	Clock speed selection[2]	Clock speed selection[1]	Clock speed selection[0]	Break between byte transfer	Start transfer / status

Type: R/W
Default value: 00

3 byte operation

0: 2 byte transfer
1: 3 byte transfer

Transfer Mode

0: normal operation
1: uneven clock transfer mode

Enable serial uP pin definition

0: normal operation
1: enable serial uP pin definition

Clock speed selection[2:0]

0: 320 ns per cycle
1: 640 ns per cycle
2: 960 ns per cycle
~
15: 2560 ns per cycle

Break between byte transfer

0: continue bit transfer without byte break
1: CSB pull high between byte transfer

Start transfer /status

Write 1 to start the data transfer
Read 1 as busy and 0 for idle

TDM FS delay control 0x2a

Bit	7	6	5	4	3	2	1	0
	TFSDC7	TFSDC6	TFSDC5	TFSDC4	TFSDC3	TFSDC2	TFSDC1	TFSDC0

Type: R/W
Default value: 00

TFSDC [7:0]

Number of clock delay for FS input
0: no delay
1: one cycle delay
255: 255 cycle delay

AUX pin input level / edge selection 0x2b

Bit	7	6	5	4	3	2	1	0
	APS7	APS6	APS5	APS4	APS3	APS2	APS1	APS0

Type: R/W
Default value: 00

APS [7:0]

Each bit corresponding to one AUX pin
0: AUX pin input as level
1: AUX pin input as edge trigger

AUX pin edge selection 0x2c

Bit	7	6	5	4	3	2	1	0
	APES7	APES6	APES5	APES4	APES3	APES2	APES1	APES0

Type: R/W
Default value: 00

APES [7:0]

Each bit corresponds to one AUX pin
0: Trigger by rising edge of AUX input
1: Trigger by falling edge of AUX input

AUX trigger register reset 0x2d

Bit	7	6	5	4	3	2	1	0
	ATRR7	ATRR6	ATRR5	ATRR4	ATRR3	ATRR2	ATRR1	ATRR0

Type: R/W
Default value: 00

ATRR [7:0]

Each bit corresponding to one AUX pin
0: normal operation
1: reset AUX trigged register

Audio Feedback control 0x2e

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Polarity of output	Enable AUX[5] output	Audio data format	Enable audio feeback

Type: R/W
Default value: 00

Polarity of output

0: normal operation
1: invert AUX[5] output

Enable AUX[5] output

0: normal operation
1: set AUX[5] as output for the comparison result

Audio data format

0: 2's complemented audio data format
1: signed audio data format

Enable audio feedback

0: no audio feedback comparison
1: enable audio feedback comparison

Audio reference value low byte 0x2f

Bit	7	6	5	4	3	2	1	0
	ARV[7]	ARV[6]	ARV[5]	ARV[4]	ARV[3]	ARV[2]	ARV[1]	ARV[0]

Audio reference value high byte 0x30

Bit	7	6	5	4	3	2	1	0
	Reserved	ARV[14]	ARV[13]	ARV[12]	ARV[11]	ARV[10]	ARV[9]	ARV[8]

Type: R/W
Default value: 0xD800

Audio reference value [14:0]

When the average (CONFIRM) audio input stream exceeds this value the output audio stream is muted.

Absolute Maximum Ratings

Symbol	Parameter	Rating	Units
Vcc	Power Supply	-0.3 to 6.0	V
Vin	Input Voltage	-0.3 to Vcc+0.3	V
Vout	Output Voltage	-0.3 to Vcc+0.3	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(5V)	Power Supply	4.5	5.0	5.5	V
Vcc(3.3V)	Input Voltage	2.7	3.3	3.6	V
V_{IN}	Input Voltage	0		Vcc	V
T_{OPR}	Storage Temperature	0		70	°C

D.C. characteristics

Symbol	Parameter	Min	Typ	Max	Units
VIL	Input Low Voltage			0.8	V
VIH(3.3V)	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

A.C. Characteristics

PIB timing

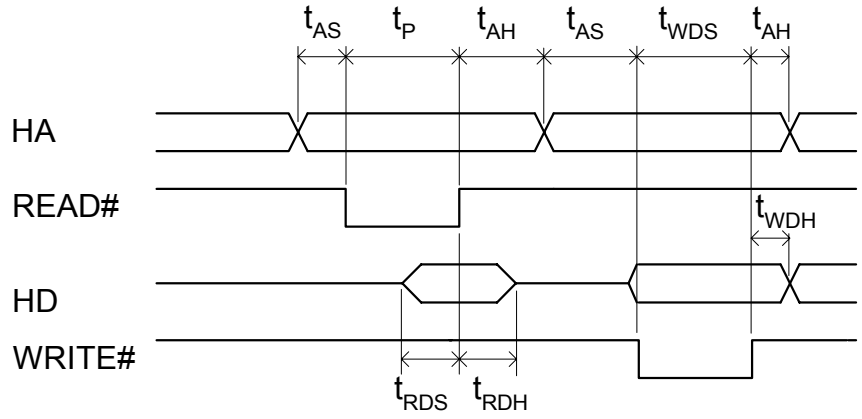
Symbol	Parameter	Min(nS)	Max(nS)
t_{AS}	Address setup time	120	
t_{AH}	Address hold time	40	
t_{P}	Command pulse width	80	640
t_{RDS}	Read data setup time	5	
t_{RDH}	Read data hold time	0	
t_{WDS}	Write data setup time	80	640
t_{WDH}	Write data hold time	40	

Serial bus timing

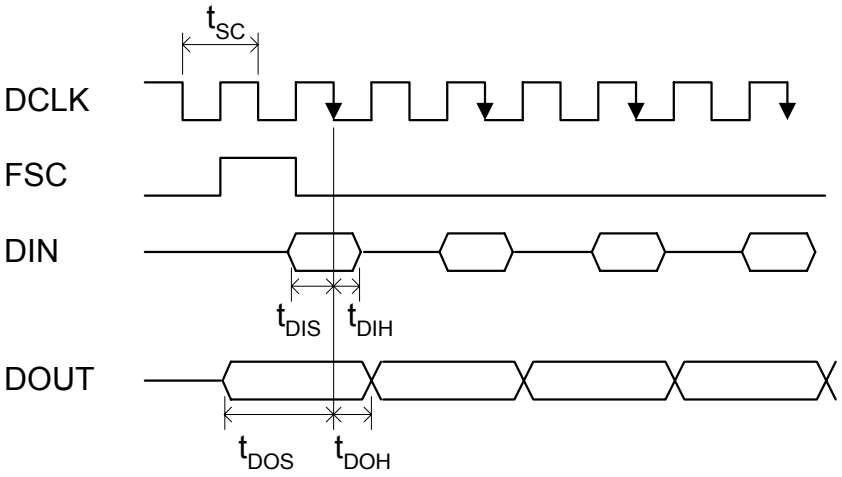
Symbol	Parameter	Min(nS)	Max(nS)
t_{SC}	Serial clock period		
t_{DIS}	Data input setup time		
t_{DIH}	Data input hold time		
t_{DOS}	Data output setup time		
t_{DOH}	Data output hold time		

Waveforms

PIB waveforms



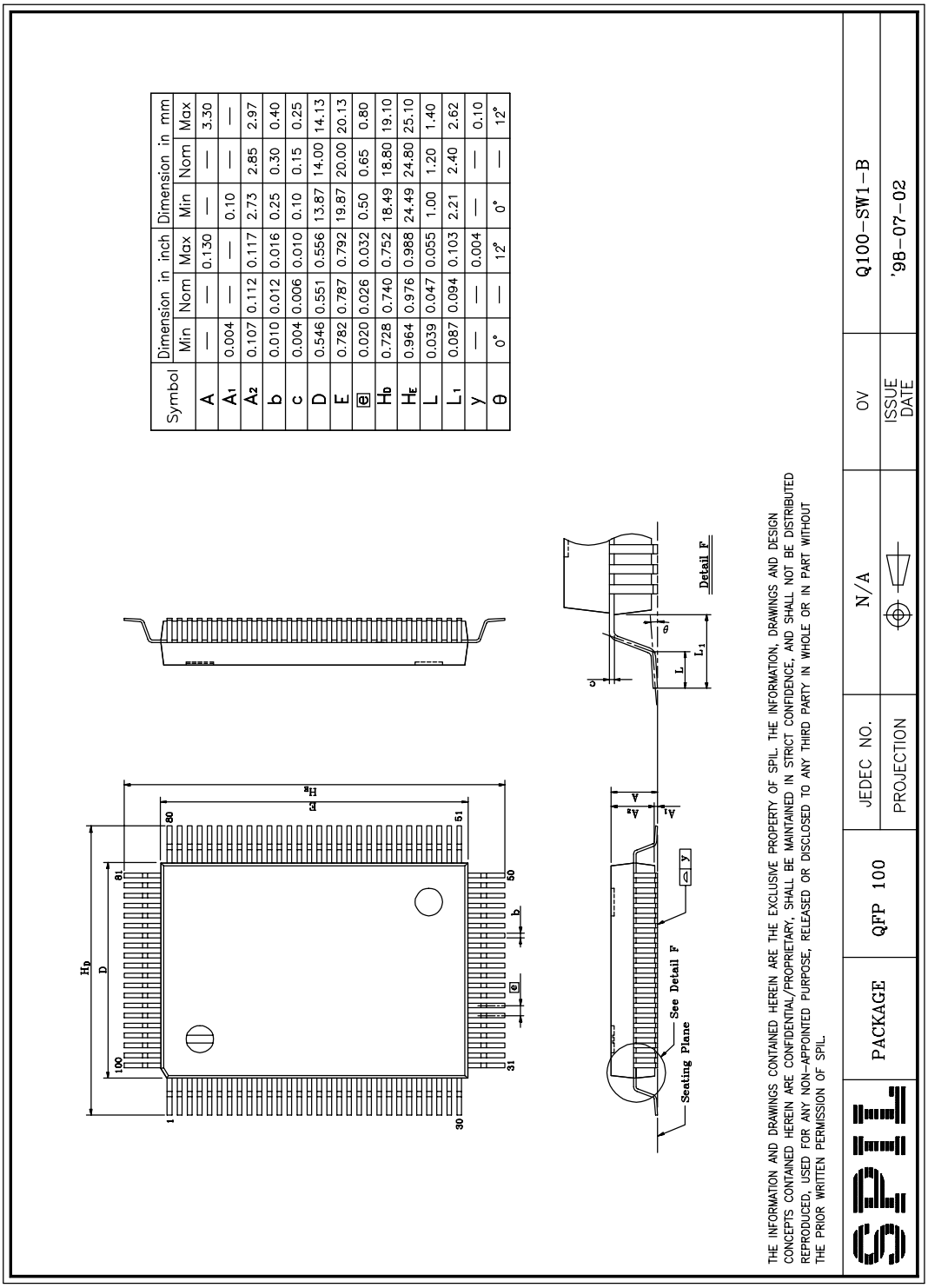
Serial bus waveforms



Application Schematics

For application schematics please visit www.tjnet.com.

Physical Dimensions



Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.130	—	—	3.30
A_1	0.004	—	—	0.10	—	—
A_2	0.107	0.112	0.117	2.73	2.85	2.97
b	0.010	0.012	0.016	0.25	0.30	0.40
c	0.004	0.006	0.010	0.10	0.15	0.25
D	0.546	0.551	0.556	13.87	14.00	14.13
E	0.782	0.787	0.792	19.87	20.00	20.13
e	0.020	0.026	0.032	0.50	0.65	0.80
H_D	0.728	0.740	0.752	18.49	18.80	19.10
H_E	0.964	0.976	0.988	24.49	24.80	25.10
L	0.039	0.047	0.055	1.00	1.20	1.40
L_1	0.087	0.094	0.103	2.21	2.40	2.62
y	—	—	0.004	—	—	0.10
θ	0°	—	12°	0°	—	12°

Legal Words

How to reach TigerJet

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

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