ADVANCE INFORMATION

TIGER560B

VoIP processor with USB for Low Cost / High Quality Internet Telephony solutions

Features

- Built in glueless support for;
  - Keypad
  - LCD
  - Phone interface (SLIC)
  - Ringer / Buzzer
- H.323 standard supported
- SIP standard supported
- No drivers required, all drivers are embedded in Windows. No hassle for installation or upgrade.
- Implements all of the required VoIP functions and USB interfacing for Internet telephony.
- Fully integrated with IP Phone Center application, provides easy to use functions;
  - PC to PC calls
  - PC to gateway calls
- "Drop in" replacement for original Tiger560
- USB HID implementation
  - Microsoft USB HID definition 1.1
  - Windows USB Telephony
  - Keypad support
  - LCD interface
  - Phone ringer/buzzer
  - Serial number support
- USB suspend mode for power save
- Remote wakeup support
- Audio functions
  - USB audio class device mode
  - Uses 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
  - Winbond W681511audio 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 for clocking
  - 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
- USB descriptor tables
  - Audio device class
  - HID
- Peripheral Interface Bus (PIB)
  - All popular peripheral chips
  - Byte-wide data
  - 6 address lines
  - 22 general purpose control lines
  - Read control signal
  - Write control signal
  - Reset control signal
- Device features
  - Single 12MHz crystal oscillator
  - 5V operation
  - Onboard 3.3V regulator
  - 100 pin PQFP/0.65mm lead pitch

General Description

VoIP (Internet Phone) applications are very popular as VoIP can provide free and/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 identical to using a regular phone and eliminate the poor call quality that results from using the PC sound card, TigerJet has developed the Tiger560B VoIP processor with USB that enables the manufacture of both low cost USB phones and USB to RJ11 adapters that enable regular phones to be used to place and receive VoIP calls. Reference schematics are available that enable OEMs to quickly bring to market a family of low cost high quality VoIP products. In addition the world leading IP Phone Center telephony application and SDK is available to manufacturers who are interested in manufacturing a complete hardware/software product.

The Tiger560B is compatible with the Microsoft USB audio driver and Windows HID this enables the Tiger560B to use the Microsoft written and supported drivers directly.

Glueless support is included for keypad, buzzer, LCD and phone interface (SLIC) these features can be implemented without the requirement for a driver (no hassle for installation or upgrade).

A low cost 4x4 or 4x8 matrix keypad can be directly connected to the Tiger560B. Key presses are passed to the Internet telephony application. To enable the phone to ring a low cost buzzer is supported. For applications that call for an LCD to display called number and phone status an LCD can be directly connected.

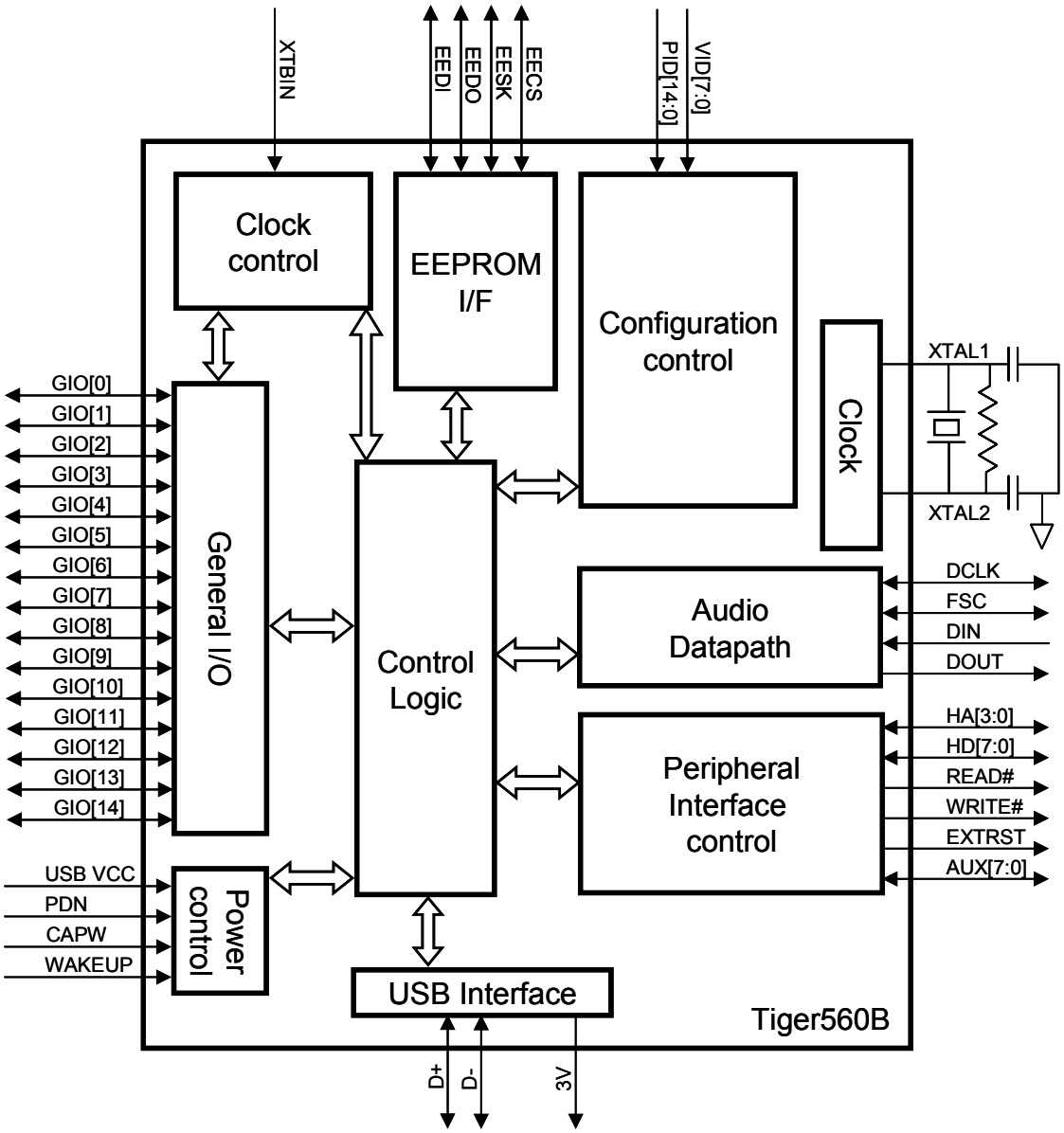
For the keypad, buzzer, LCD and phone interface (SLIC), there is no need to load a driver as the embedded Windows drivers are used.

The world leading IP Phone Center telephony application is available to bundle with VoIP products that are based on the Tiger560B chip.

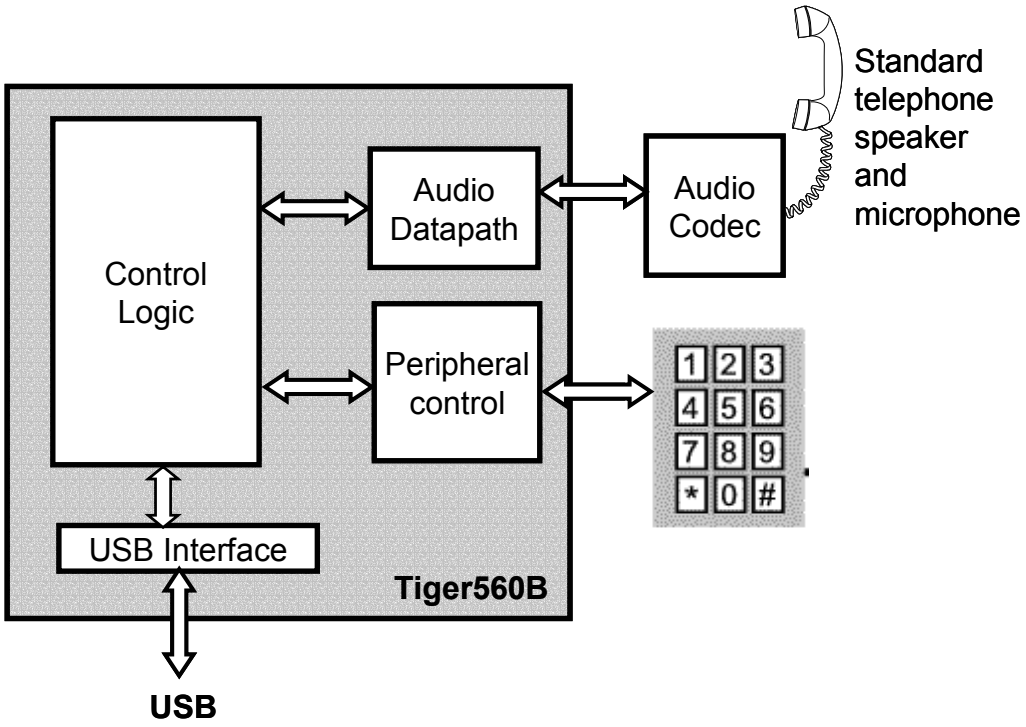
Ordering information

The order code for the Tiger560B is Tiger560B.

Functional Block Diagram



Typical Implementation



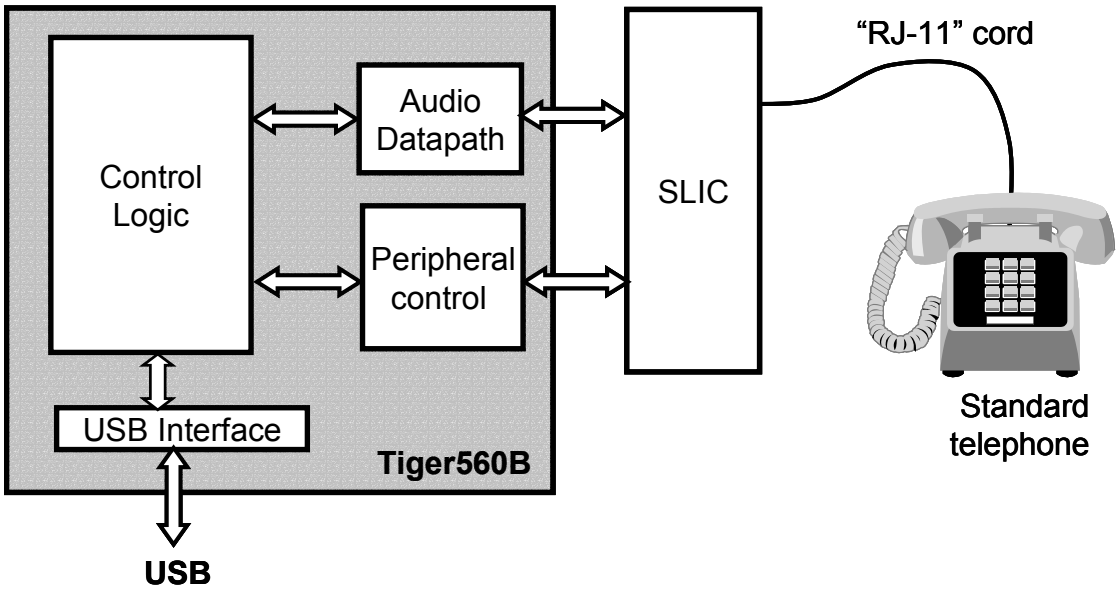
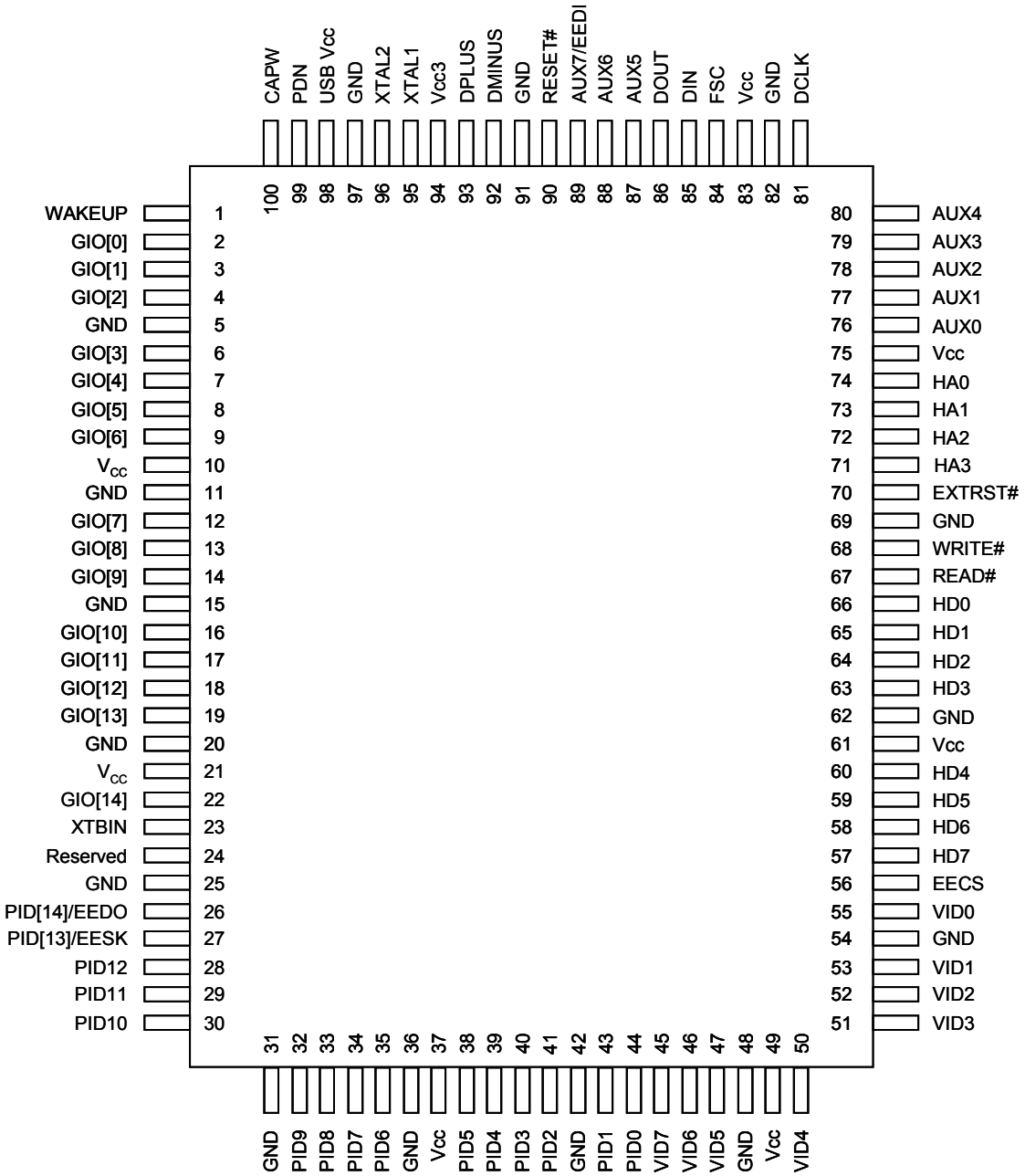


Table of Contents

Pin-out

Pin-out diagram



Pin assignment by pin number

Pin	Name
1	WAKEUP
2	GIO[0]
3	GIO[1]
4	GIO[2]
5	GND
6	GIO[3]
7	GIO[4]
8	GIO[5]
9	GIO[6]
10	$V_{CC}5$
11	GND
12	GIO[7]
13	GIO[8]
14	GIO[9]
15	GND
16	GIO[10]
17	GIO[11]
18	GIO[12]
19	GIO[13]
20	GND
21	$V_{CC}5$
22	GIO[14]
23	XTBIN
24	Reserved
25	GND
26	PID[14]/EEDO
27	PID[13]/EESK
28	PID12
29	PID11
30	PID10
31	GND
32	PID9
33	PID8
34	PID7

Pin	Name
35	PID6
36	GND
37	$V_{CC}5$
38	PID5
39	PID4
40	PID3
41	PID2
42	GND
43	PID1
44	SAD0
45	VID7
46	VID6
47	VID5
48	GND
49	$V_{CC}5$
50	VID4
51	VID3
52	VID2
53	VID1
54	GND
55	VID0
56	EECS
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	AUX7/EEDI
90	RESET#
91	GND
92	DMINUS
93	DPLUS
94	$V_{CC}3$
95	XTAL1
96	XTAL2
97	GND
98	$V_{CC}5$
99	PDN
100	CAPW

Signal assignments by functional category

General I/O pins	
Name	**Pin**
GIO[0]	2
GIO[1]	3
GIO[2]	4
GIO[3]	6
GIO[4]	7
GIO[5]	8
GIO[6]	9
GIO[7]	12
GIO[8]	13
GIO[9]	14
GIO[10]	16
GIO[11]	17
GIO[12]	18
GIO[13]	19
GIO[13]	22

Control Signals	
Name	**Pin**
CAPW	100
PDN	99
PID[13]	27
PID[14]	26
Reserved	24
RESET#	90
WAKEUP	1
XTAL1	95
XTAL2	96
XTBIN	23

Configuration Interface	
Name	**Pin**
PID0	44
PID1	43
PID2	41
PID3	40
PID4	39
PID5	38
PID6	35
PID7	34
PID8	33
PID9	32
PID10	30
PID11	29
PID12	28
VID0	55
VID1	53
VID2	52
VID3	51
VID4	50
VID5	47
VID6	46
VID7	45
EESK	27
EECS	56
EEDO	26
EEDI	89

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

Peripheral Interface Bus	
Name	**Pin**
AUX0	76
AUX1	77
AUX2	78
AUX3	79
AUX4	80
AUX5	87
AUX6	88
AUX7	89
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

Power and Ground	
Name	**Pin**
USB V_{CC}	98
$V_{CC}5$	10, 21, 37, 49, 61, 75, 83
$V_{CC}3$	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#
AUX7	I/O	PIB aux port bit 7	EEDI EEPROM DI pin
CAPW	I	Remote wakeup 1: Enable 0: Disable	
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+	
EECS	I/O	EEPROM CS pin Pull high for EEPROM present. Internal pull low.	
EESK	I/O		
EEDO	I/O		
EXTRST#	O	PIB reset	
FSC	I/O	Serial port frame sync	
GIO[0 – 14]	I/O	General I/O pins	
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] : mode selection 00: Keypad scan with HD[3:0] (default mode) 01: Tiger560 mode 10: Test mode 11: Keypad scan with HD[7:0]
HD0 – HD7	I/O	PIB data bus	
PDN	O	Power control 1: Normal operation 0: Turn off power switch	
READ#	O	PIB read	
RESET#	I	System reset input	
PID[14:0]	I	Product ID [0 – 14] input on reset	
VID[0:7]	I	Vendor ID [0:7]	
USB V_{CC}	Power	5V when in suspend mode	
$V_{CC}3$	O	3.3V output for USB port	

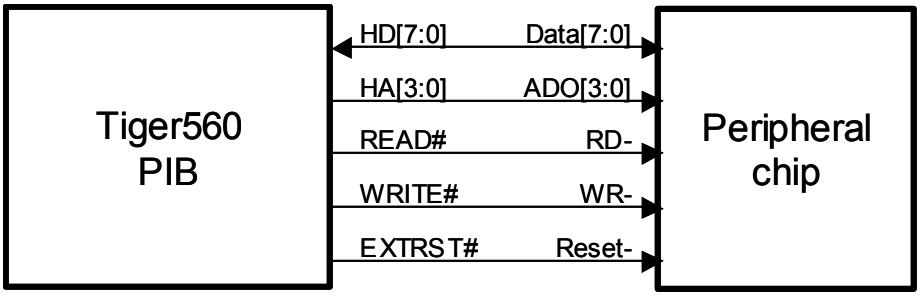
VCLK	I/O	Memory clock	
WAKEUP	I	Wakeup signal 1: Wakeup system 0: Idle	
WRITE#	O	PIB write	
XTAL1	I	Crystal oscillator	
XTAL2	O	Crystal oscillator	
XTBIN	I	Oscillator input	

Functional description

Peripheral Interface Bus (PIB)

To enable a “glueless” interface to most popular peripheral chips, the Tiger560B 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 Tiger560B 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[7:0] 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.

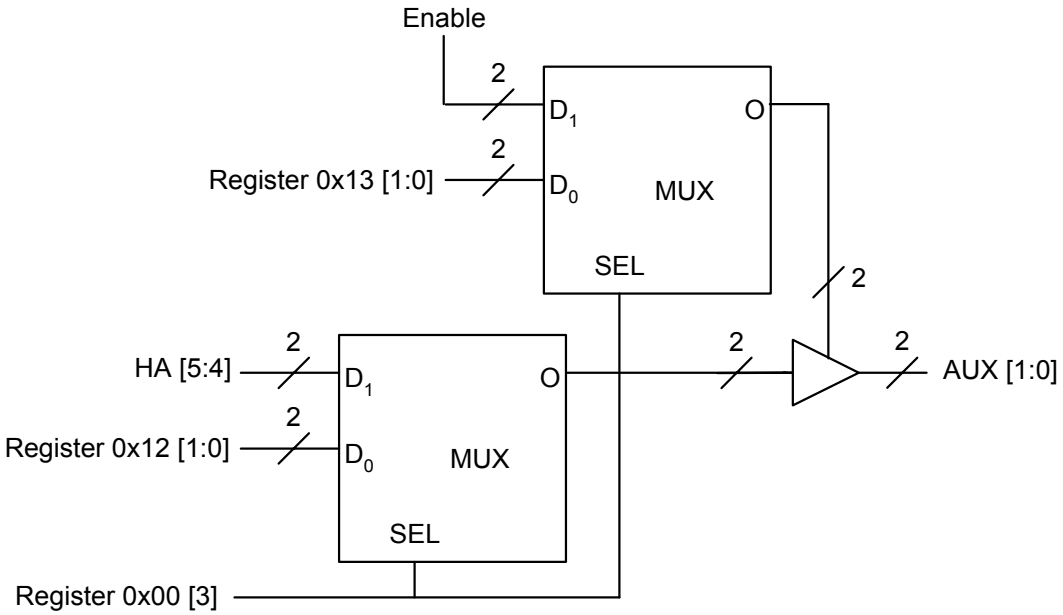
AUX line inputs can be inverted individually with register 0x15. This is useful for determining the active polarity of the AUX line when used for Wake-up. This register does not change signal polarity when the AUX lines are used for Interrupt or Suspend.

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.

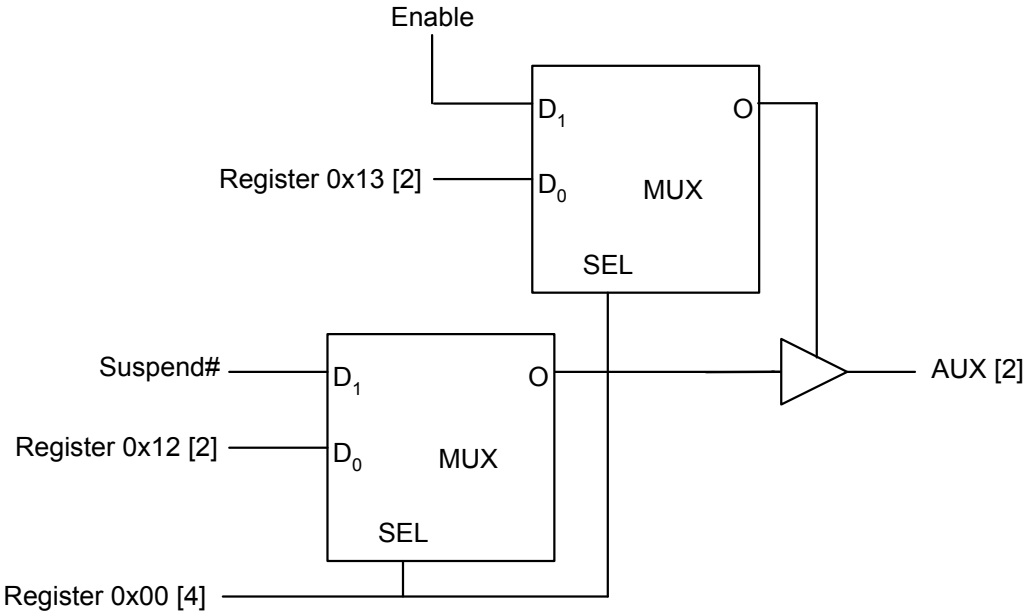
Address lines HA4 and HA5

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.



USB Interface

The Tiger560B implements a fully compliant USB 1.1 interface. The USB line drivers and 3.3 volt regulator are included on chip and the Tiger560B can be connected directly to the USB bus.

Chip mode selection
HA[3:2] reset latched value determines the operating mode of the Tiger560B.

HA[3:2]	Chip mode
00	Hardware keypad scan with HD[3:0] 4x4 keypad matrix
01	No keypad scan
10	Test mode
11	Keypad scan with HD[7:0] 4x8 keypad matrix

USB vendor and product I.D.s

The USB vendor and product I.D.s are set by pull-up/down resistors on the following pins.
Vendor I.D. bits 0 – 7 on VID[7:0]
Vendor I.D. bits 8 – 15 hard coded as 0x06
Product I.D. bits 0 – 14 PID[14:0]
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: Audio + HID + Composite USB device descriptor table 01: Audio USB device descriptor table 10: Audio + Composite USB device descriptor table 11: Audio + HID USB device descriptor table
ProductID[13:12]	00: max power 500 ma 01: max power 300 ma 10: max power 200 ma 11: max power 100 ma
ProductID[11]	0: No USB wake up function support 1: Support USB wake up function
ProductID[10]	Reserved
ProductID[9]	0: Audio data shift direction MSB first 1: Audio data shift direction LSB first
ProductID[8]	Reserved
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]	Reserved
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
Product ID[11]=1 .AND. Product ID[5]=1	Use 48MHz, otherwise 12MHz crystal

USB descriptor table – EEPROM download

The Tiger560B can be programmed to download the USB ID table from EEPROM

If Tiger560B set to use the descriptor table from EEPROM, the product ID[15:2] is set to program the internal function setting.

Vendor Commands

The Tiger560B uses the USB vendor command to access the internal registers and parallel port interface. For the most efficient transfer of data, the Tiger560B 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 Tiger560B 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. Tiger560B 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 Tiger560B 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 560B 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 Tiger560B only requires a 256 location address space. The address space 0x00 to 0xBF is used for the Tiger560B'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 Tiger560B 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

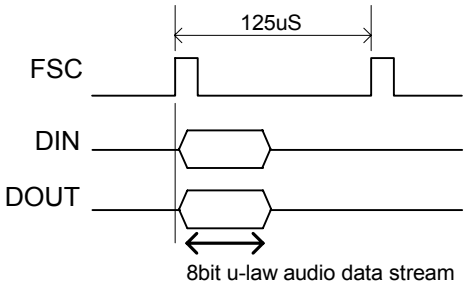
USB Endpoint 5 is used for the interrupt transfer on the Tiger560B. The polling interval is 1ms. Each time polling occurs, 2 bytes of data will be transferred from the Tiger 560B to the host. The first byte of data is the value of AUX pins current state. The second byte is the value of specified source. 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 Tiger560B 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 Tiger560B. 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 productID definition and a set of registers that is specific to the Tiger560B. For the detailed register definition please see the registers section.

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 MSB position can be programmed using internal register 0x29.



Serial port data transfer
Tiger560B 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 Tiger560B 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. Tiger560B 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.

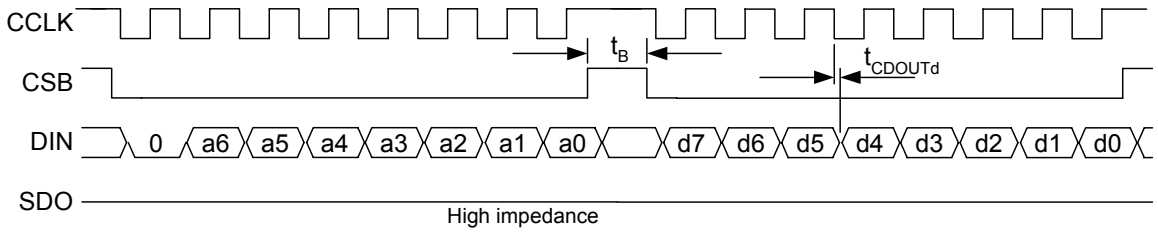
SPI micro-controller interface

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

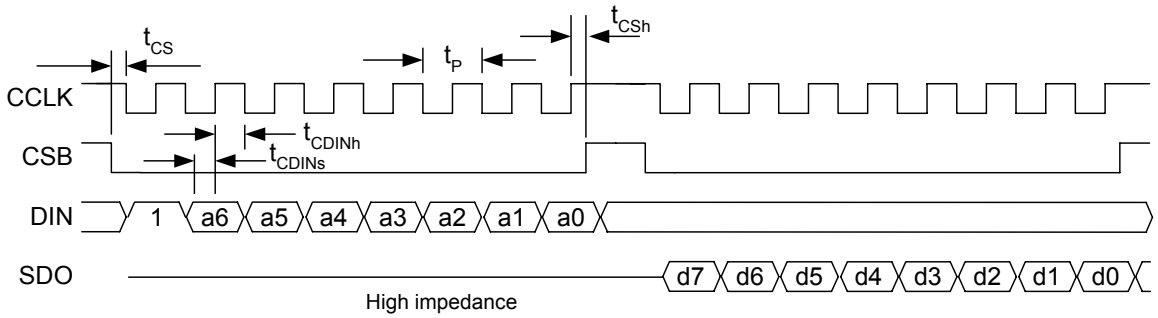
The interface consists of a clock signal (CCLK), chip select (CSB), data input (CDIN) and data output (CDOUT). These signals share the HA[3], HA[2], AUX[0] and AUX[1] pins. The SPI interface is enabled when bit 5 of register0x29 is set to 1.

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 Tiger560B register 0x29.



Serial write timing diagram



Serial read timing diagram

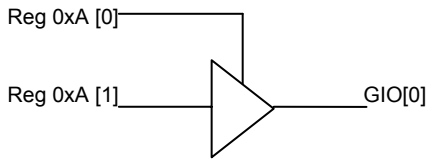
Wakeup support

The Tiger560B can generate a wakeup signal to the host computer when the WAKEUP input goes high. To enable the Wakeup function set PID[11] and CAPW = 1.

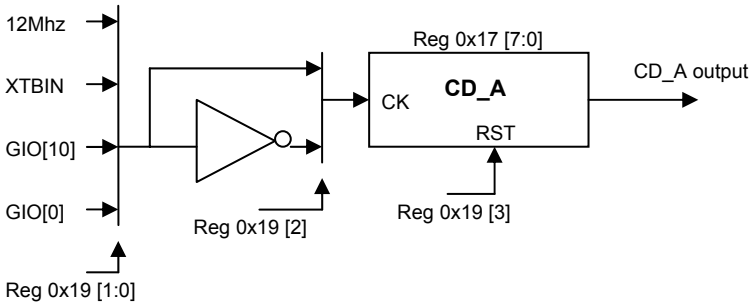
General purpose I/O (GIO) and Internal Clock divider Definition

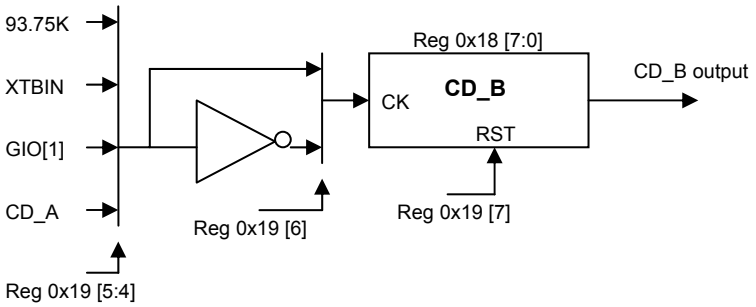
Tiger560B has 15 GIO pins for general purpose use. Each pin can be individually programmed through registers 0x0a, 0x0f, 0x11.

GIO[7:0] are bidirectional In/Out pads. Register 0xA controls GIO[3:0] and Register 0xF controls GIO[7:4]. Below is the logic diagram for GIO[0]. One register read/write can control 4 GIO pins.

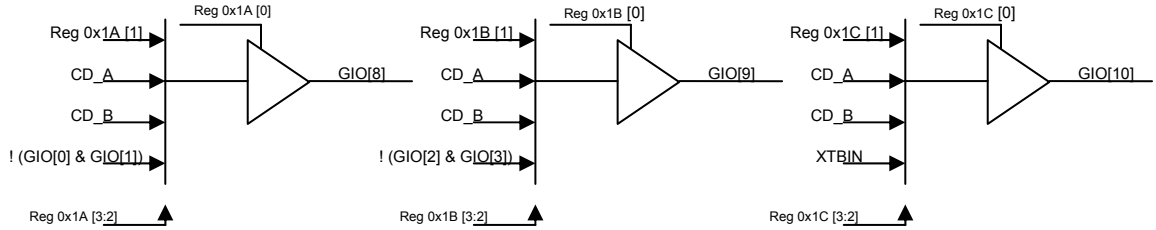


There are two internal clock dividers, CD_A and CD_B. The source of the clock and the ratio can be programmed by internal registers. Following is the logic diagram for CD_A and CD_B.

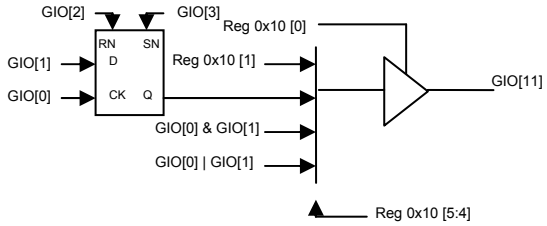


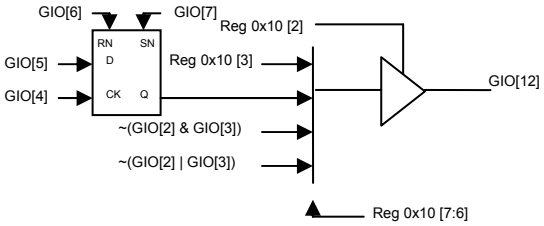


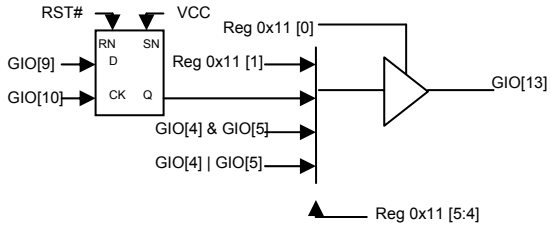
GIO[10:8] can be programmed as the clock divider outputs. Each pin is controlled by a single register. Below is the logic diagram for GIO[10:8].

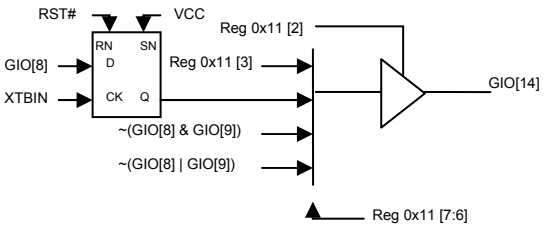


GIO[14:11] have different combination logic or flip-flop output. Register 0x10 control GIO[12:11] and register 0x11 control GIO[14:13]. Each register controls two GIO pins. Below is the logic diagram for GIO[14:11].









Registers

Register addressing

Tiger560B 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
	Reserved	USB reset enable	Reserved	Reserved	Aux / HA	Reserved	Reserved	EXTRST#

Type; R/W
Default value: 40

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

0 = Enable USB reset
1 = Disable USB reset

Aux / HA

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

EXTRST#

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

USB Vendor Command Registers

Register 0x07: HID keypad scanning column mask

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

Type; R/W
Default value: 00

HMask[n] :

0 = Disable HD[n] line
1 = Enable HD[n] line

Register 0x08: HID Report Value low byte

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

Register 0x09: HID Report Value high byte

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

Type; RO
Default value: 00

HReport[15:0] : Current HID report value

Register 0x0a: GIO[3:0] control register

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

Type; R/W
Default value: 00

GIO_DIR[n] :
0: GIO[n] input mode
1: GIO[n] output mode

GIO_O[n] :
GIO[n] Output value, Read as current GIO[n] value.

Register 0x0b: HID keypad scanning debounce period

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

Type; R/W
Default value: 48

HDeb [7:0] : Debounce period
Number of scan cycles to ignore when detect changes.
Scan clock cycle = 0.5ms, HDeb = 48, debounce period = 48 * 0.5ms = 24ms.

Register 0x0c: Software HID report value low byte

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

Register 0x0d: Software HID report value high byte

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

Type; R/W
Default value: 0x00

SHReport [15:0] : Software HID report value

Register 0x0e: Software HID report control register

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	ValidSH	EnableSH	Kspeed[2]	Kspeed[1]	Kspeed[0]

Type; R/W
Default value: 0x00

Kspeed [2:0] : HID keypad scan speed

0: scan clock = 0.5ms, 1ms per line
1: scan clock = 1 ms, 2 ms per line
2: scan clock = 2 ms, 4 ms per line
3: scan clock = 4 ms, 8 ms per line
other: reserved

EnableSH: Software HID enable bit. Set to high to enable software HID report.

ValidSH: Valid software HID report. Set to high will trigger HID interface send SHReport[15:0] to system. It will be cleared by hardware when report has been send.

Register 0x0f: GIO[7:4] control register

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

Type; R/W
Default value: 00

GIO_DIR[n] :

0: GIO[n] input mode
1: GIO[n] output mode

GIO_O[n] :

GIO[n] Output value, Read as current GIO[n] value.

Register 0x10: GIO[12:11] control register

Bit	7	6	5	4	3	2	1	0
	GIO12O SEL 1	GIO12O SEL 0	GIO11O SEL 1	GIO11O SEL 0	GID_O 12	GIO_DIR 12	GID_O 11	GIO_DIR 11

Type; R/W
Default value: 00

GIO_DIR[n] :
0: GIO[n] input mode
1: GIO[n] output mode

GIO_O[n] :
GIO[n] Output value, Read as current GIO[n] value.

GIO11OSEL[1:0]: GIO[11] output source selection:
0: Register 0x10 bit[1]
1: D-type Flip Flop, CK=GIO[0], D=GIO[1], RSTN=GIO[2], SETN=GIO[3]
2: AND gate, GIO[0] & GIO[1]
3: OR gate, GIO[0] | GIO[1]

GIO12OSEL[1:0]: GIO[12] output source selection:
0: Register 0x10 bit[3]
1: D-type Flip Flop, CK=GIO[4], D=GIO[5], RSTN=GIO[6], SETN=GIO[7]
2: NAND gate, ~(GIO[2] & GIO[3])
3: NOR gate, ~(GIO[2] | GIO[3])

Register 0x11: GIO[14:13] control register

Bit	7	6	5	4	3	2	1	0
	GIO14O SEL 1	GIO14O SEL 0	GIO13O SEL 1	GIO13O SEL 0	GID_O 14	GIO_DIR 14	GID_O 13	GIO_DIR 13

Type; R/W
Default value: 00

GIO_DIR[n] :
0: GIO[n] input mode
1: GIO[n] output mode

GIO_O[n] :
GIO[n] Output value, Read as current GIO[n] value.

GIO13OSEL[1:0]: GIO[13] output source selection:
0: Register 0x11 bit[1]
1: D-type Flip Flop, CK=GIO[10], D=GIO[9], RSTN=RST#, SETN=1
2: AND gate, GIO[4] & GIO[5]
3: OR gate, GIO[4] | GIO[5]

GIO14OSEL[1:0]: GIO[14] output source selection:
0: Register 0x11 bit[3]
1: D-type Flip Flop, CK=XTBIN, D=GIO[8], RSTN=RST#, SETN=1
2: NAND gate, ~(GIO[8] & GIO[9])
3: NOR gate, ~(GIO[8] | GIO[9])

PIB Aux Port Control

PIB Aux Port Data 0x12

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

Type; R/W
Default value: 00

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

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
	AuxC7	AuxC6	AuxC5	AuxC4	AuxC3	AuxC2	AuxC1	AuxC0

Type; R/W
Default value: 00

AuxC[7:0]

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

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
	AuxP7	AuxP6	AuxP5	AuxP4	AuxP3	AuxP2	AuxP1	AuxP0

Type; R/W
Default value: 00

AuxP[7:0]

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
	AuxW7	AuxW6	AuxW5	AuxW4	AuxW3	AuxW2	AuxW1	AuxW0

Type; R/W
Default value: 00

AuxW[7:0]

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

Register 0x16: EEPROM control register

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	EPRLD	EEDI_I	EEDO_O	EESK_O	EECS_O	EPEN

Type; R/W
Default value: 00

EPEN: Enable software control of EEPROM pin.
EECS_O: EECS output value
EESK_O: EECK output value
EEDO_O: EEDO output value
EEDI_I: Read only bit, current EEDI value
EPRLD: EEPROM reload. Set this bit from 1 to 0 will trigger hardware reload the EEPROM.

Register 0x17: Clock divider (A) Ratio Value register

Bit	7	6	5	4	3	2	1	0
	CARatio_7	CARatio_6	CARatio_5	CARatio_4	CARatio_3	CARatio_2	CARatio_1	CARatio_0

Type; R/W
Default value: 00

Clock Divide Ratio = (CARatio + 1) * 2
Example: Input clock = 12Mhz, CARatio = 0, Clock divider A output = 12 / ((0+1)*2) = 6 Mhz

Register 0x18: Clock divider (B) Ratio Value register

Bit	7	6	5	4	3	2	1	0
	CBRatio _7	CBRatio _6	CBRatio _5	CBRatio _4	CBRatio _3	CBRatio _2	CBRatio _1	CBRatio_0

Type; R/W
Default value: 00

Clock Divide Ratio = (CBRatio + 1) * 2
Example: Input clock = 1 Khz, CBRatio = 9, Clock divider B output = 1K / ((9+1)*2) = 50 hz

Register 0x19: Clock divider Control register

Bit	7	6	5	4	3	2	1	0
	CB_RST	CB_Inv	CBSrc _1	CBSrc _0	CA_RST	CA_Inv	CASrc _1	CASrc_0

Type; R/W
Default value: 00

CASrc[1:0] : Clock divider A Source Selection:

0: 12Mhz
1: XTBIN
2: GIO[10]
3: GIO[0]

CA_Inv: Invert input clock
CA_RST: Reset Clock divider A

CBSrc[1:0] : Clock divider B Source Selection:

0: 93.75Khz
1: XTBIN
2: GIO[1]
3: Clock divider A output

CB_Inv: Invert input clock
CB_RST: Reset Clock divider B

Register 0x1a: GIO[8]Control register

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	GIO8S_1	GIO8S_0	GID_O 8	GIO_DIR 8

Type; R/W
Default value: 00

GIO_DIR 8: GIO[8] Direction

0: Input Mode
1: Output Mode

GIO_O 8: GIO[8] Register Output Value
GIO8S[1:0] : GIO[8] Output Source Selection

0: From GIO_O 8
1: Counter Divider A output
2: Counter Divider B output
3: NAND gate, ~(GIO[0] & GIO[1])

Register 0x1b: GIO[9]Control register

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	GIO9S_1	GIO9S_0	GID_O 9	GIO_DIR 9

Type; R/W
Default value: 00

GIO_DIR 9: GIO[9] Direction
0: Input Mode
1: Output Mode
GIO_O 9: GIO[9] Register Output Value
GIO9S[1:0] : GIO[9] Output Source Selection
0: From GIO_O 9
1: Counter Divider A output
2: Counter Divider B output
3: NAND gate, ~(GIO[2] & GIO[3])

Register 0x1c: GIO[10]Control register

Bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	GIO10S_1	GIO10S_0	GID_O 10	GIO_DIR 10

Type; R/W
Default value: 00

GIO_DIR 10: GIO[10] Direction
0: Input Mode
1: Output Mode
GIO_O 10: GIO[10] Register Output Value
GIO10S[1:0] : GIO[10] Output Source Selection
0: From GIO_O 10
1: Counter Divider A output
2: Counter Divider B output
3: From XTBIN

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 ~ 0xF000 : Scale up X4
0x0EFF ~ 0xC000 : Scale up X2
0x0BFF ~ 0x0500 : No Scaling
0x04FF ~ 0x0100 : Scale down X2
0x00FF ~ 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 ~ 0x0E00 : Scale up X4
0x0DFF ~ 0x0B00 : Scale up X2
0x0AFF ~ 0x0500 : No Scaling
0x04FF ~ 0x0200 : Scale down X2
0x01FF ~ 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

EEPROM definition

If the EECS pin is high during RESET# the Tiger560B will access the EEPROM after hardware RESET# signal. Tiger560B supports 9346 (64*16) EEPROM format. Following table shows the definition of the entries. If the EEPROM contents pass the check sum comparison, the valid serial number can be access through standard USB GET_STRING command and the VendorID and ProductID will be used. If the check sum value is not correct, no serial number will be reported and default ID will be used.

Word Offset	Description	Example
0	Must be 0x0000	0x0000
1	Must be 0x0000	0x0000
2	USB Vendor ID	0x06E6
3	USB Product ID	0xABCD
4	Serial number word 0, Byte 0 at LSB, byte 1 at MSB	0x3130
5	Serial number word 1	0x3332
6	Serial number word 2	0x3534
7	Serial number word 3	0x3736
8	Serial number word 4	0x3938
9	Serial number word 5	0x3130
10	Sum from byte 0(low byte of word 0) to byte 17(high byte of word 9) + 0x1234	0xE6 + 0x06 + 0xCD + 0xAB +0x30 + 0x31 + ….. + 0x1234 = SUM

- Example shows the serial number = 012345678901

HID Keypad definition

Tiger560B supports hardware keypad scanning based on the following table definition. Each HD[n] bit can be masked by register 0x07. HA[3:2] controls the format of the hardware keypad scanning.

	HD0	HD1	HD2	HD3	HD4	HD5	HD6	HD7
	Col 0	**Col 1**	**Col2**	**Col 3**	**Col 4**	**Col 5**	**Col 6**	**Col 7**
Row 0/AUX0	1	2	3	Drop	Hold	PageUp	Return	Button1
Row 1/AUX1	4	5	6	Sent	Redial	PageDown	Home	Button2
Row 2/AUX3	7	8	9	Delete	Flash	Up	Volume+	Button3
Row 3/AUX4	*	0	#	Mute	Transfer	Down	Voume-	Button4

AUX5: Hook pin, 0: On-Hook
1: Off-Hook

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

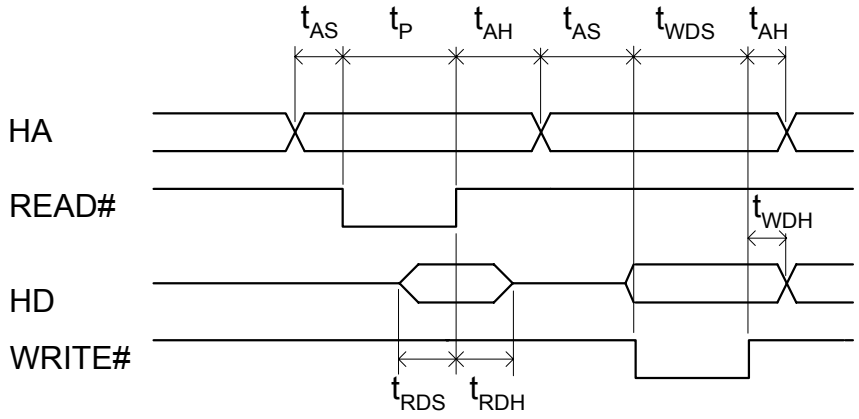
Symbol	Parameter	Min(nS)	Max(nS)
t_{AS}	Address setup time	120	
t_{AH}	Address hold time	40	
t_{P}	Command pulse width	80	640
t_{RDS}	Read data setup time	5	
t_{RDH}	Read data hold time	0	
t_{WDS}	Write data setup time	80	640
t_{WDH}	Write data hold time	40	

Serial bus timing

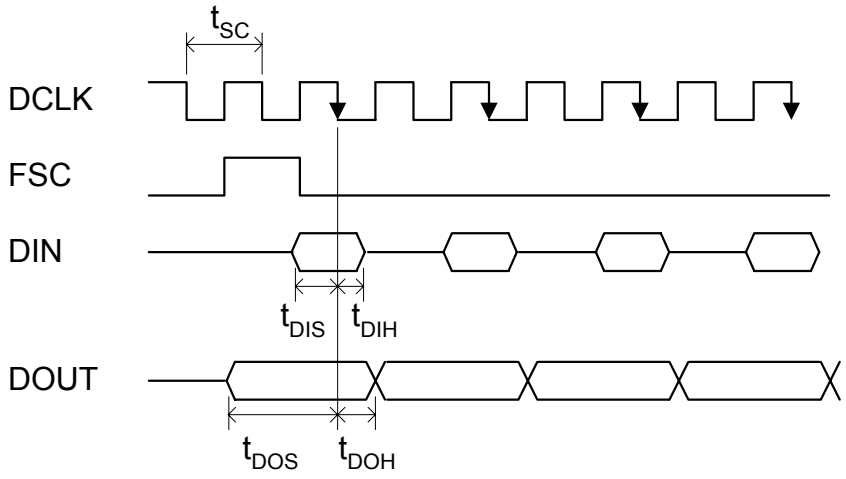
Symbol	Parameter	Min(nS)	Max(nS)
t_{P}	Serial clock period	640	5120
t_{CS}	Chip Select setup time	320	
t_{CSh}	Chip Select hold time	320	
t_{CDINs}	CDIN setup time	40	
t_{CDINh}	CDIN hold time	40	
t_{CDOUTd}	CDOUT data out delay		30
t_{B}	Break between address	1280	

Waveforms

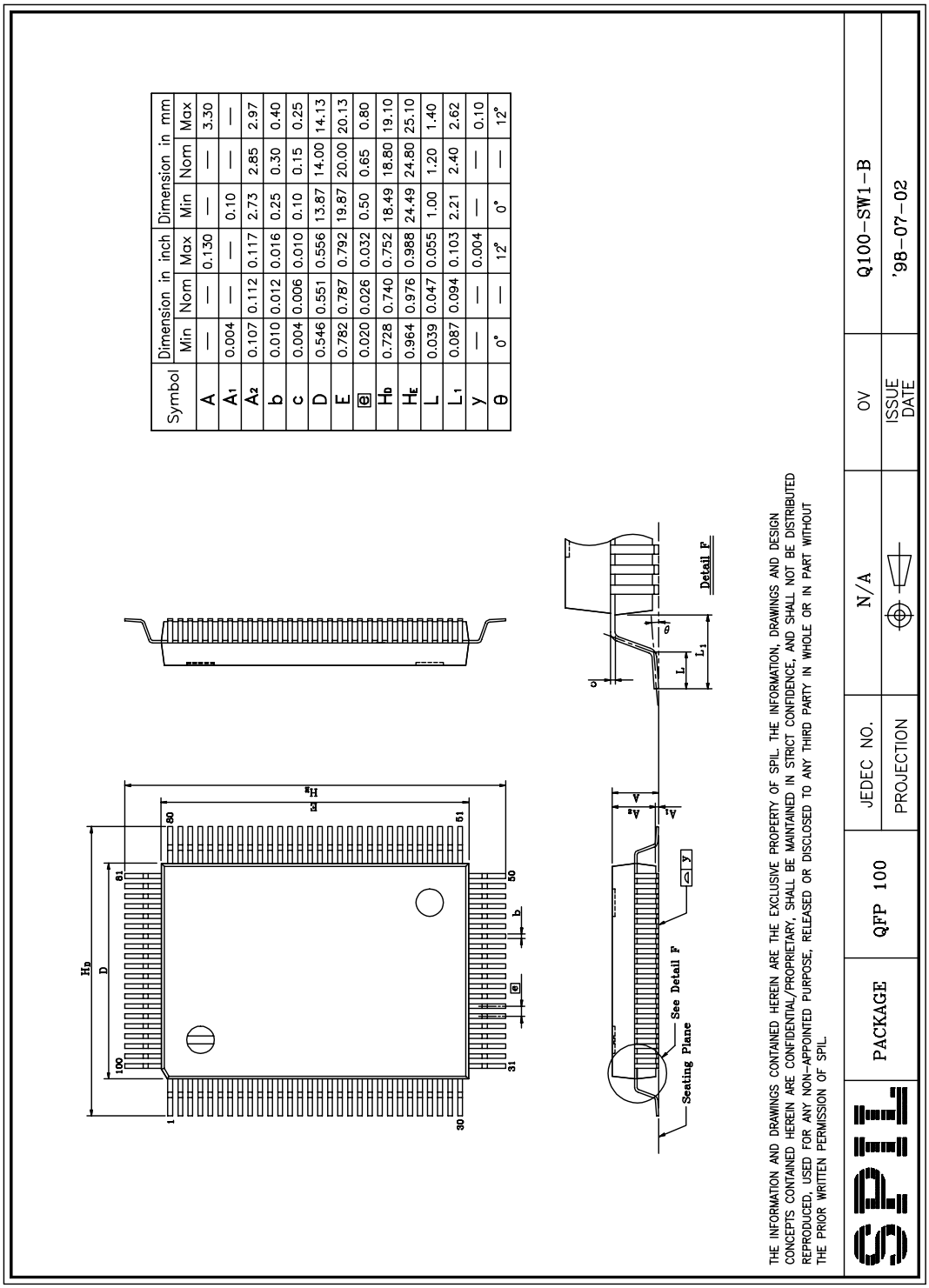
PIB waveforms



Serial bus waveforms



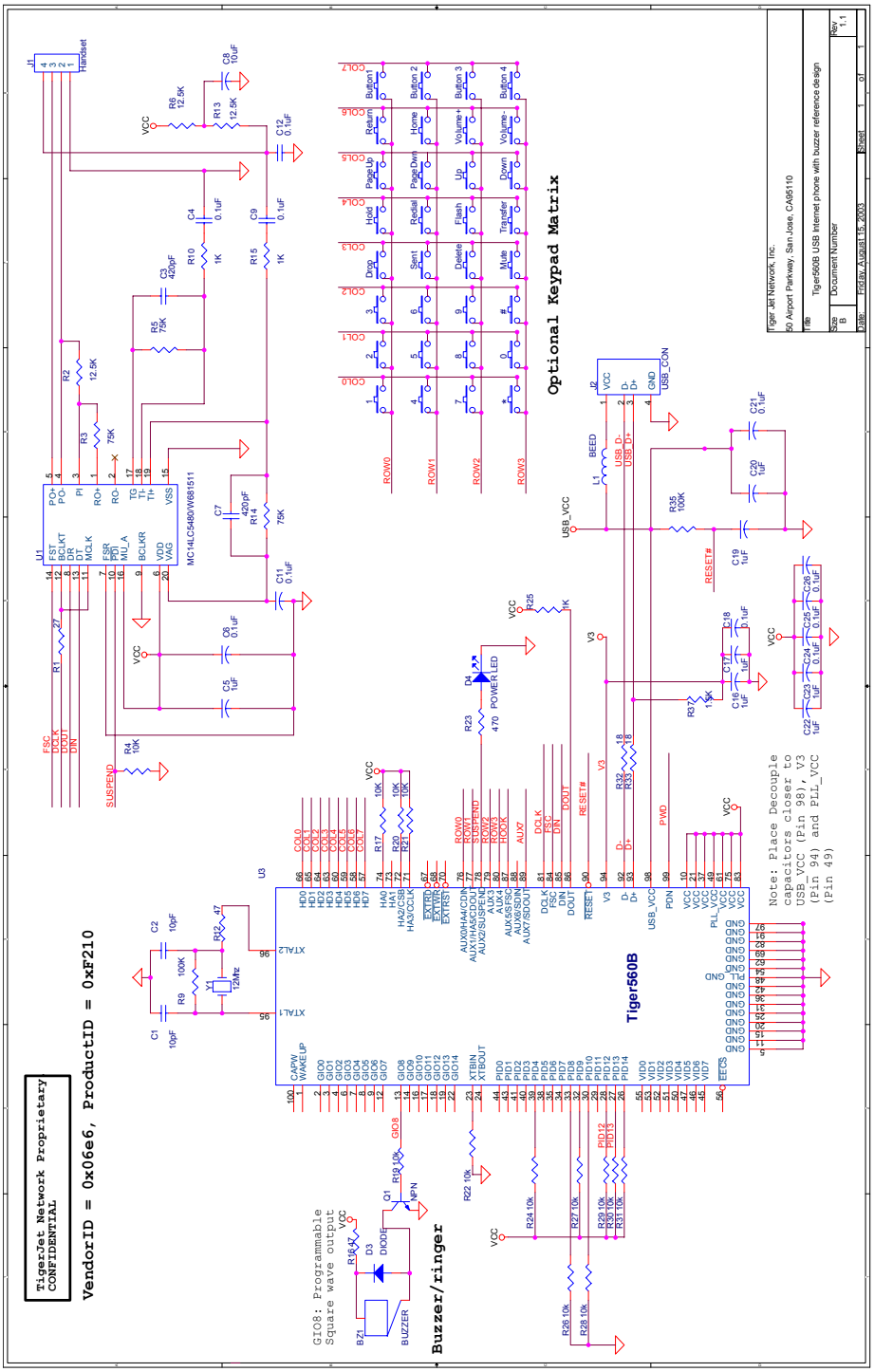
Physical Dimensions



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°

Typical Application Schematic

Typical application schematic that includes a buzzer/ringer. Please contact TigerJet or your TigerJet representative for the latest reference schematics, including LCD, and phone (SLIC) interface.



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