



*Your Imagination, Our Creation*

---

# GL640USB GL640USB-A

## IEEE-1284 to USB Bridge

*SPECIFICATION 1.1*  
*June 7, 1999*

**Genesys Logic, Inc.**

10F, No.11, Ln.3, Tsao Ti Wei, Shenkeng, Taipei, Taiwan

Tel: +886-2-2664-6655 Fax: +886-2-2664-5757

<http://www.genesyslogic.com>

## **Index**

1. Features.....	2
2. Description .....	3
3. IEEE 1284 Interface.....	3
4. Pin Configuration .....	4
5. Pin Descriptions .....	5
6. Block Diagram.....	7
7. 8-BIT RISC MCU INSTRUCTION .....	9
7.1 MEMORY ORGANIZATION .....	9
7.2 MCU FUNCTION REGISTERS .....	11
7.3 I/O Register Summary .....	15
7.4 USB Register Summary.....	22
8. INSTRUCTION SET SUMMARY .....	27
9. Advantages of GL640USB .....	30
10. EPP Timing .....	31
10.1 EPP Burst Data Write .....	31
10.2 EPP Burst Data Read.....	32
11. Electrical Characteristics .....	33
11.1 Absolute Maximum Ratings (Voltages referenced to GND).....	33
11.2 DC Characteristics (Digital Pins).....	33
11.3 DC Characteristics (VCP/D+/D-).....	34
11.4 Switching Characteristics.....	35
12. Package .....	36

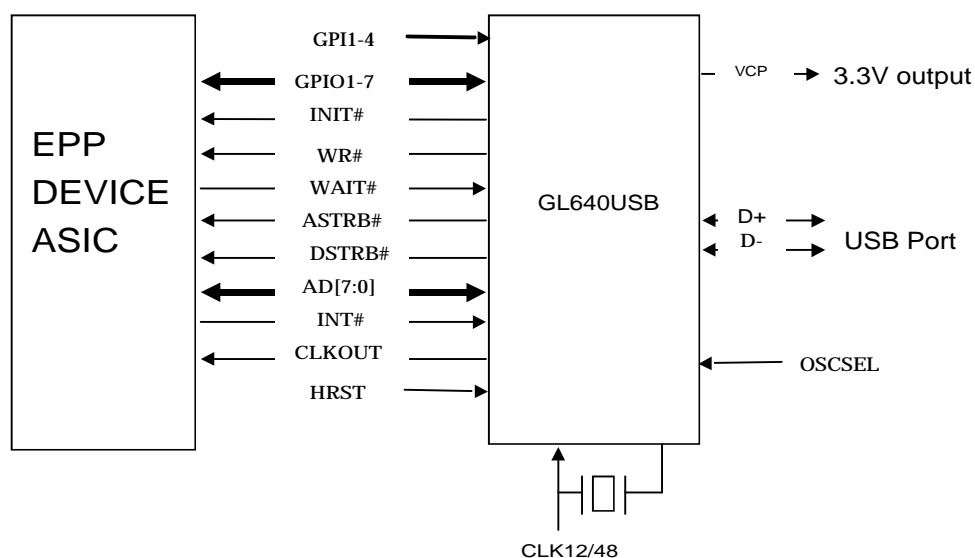
## 1. Features

- Low-cost solution for full-speed USB device
- Applications – Printer, Scanner, PC Camera, External Storage Device, CD-ROM
- IEEE 1284 interface supports bi-directional communication
- On-chip EPP engine
- On-chip 8-bit micro-controller
- USB Specification Compliance
  - Conforms to USB 12Mbps Specification, Version 1.0
  - Supports 1 device address and 4 endpoints
- 2 sets of 64-byte FIFO with dynamic allocation
  - a. 64-In/64-Out
  - b. 64-In0/64-In1 ping pong buffer
  - c. 64-Out0/64-Out1 ping pong buffer
- Integrated USB transceiver
- On-chip 3.3v regulator
- Improved output drivers with slew-rate control to reduce EMI
- 12MHz external clock
- On-chip PLL to support 48MHz internal clock
- Internal power-on reset(POR)
- Supports suspend/normal mode power management
- Customized firmware/driver for scanner application
- The driver allows current Win98/Win95 OSR-2 printer drivers to print seamlessly to USB
- Fully plug & play compatible
- Single-chip solution in cost saving 40-pin SOJ(GL640USB) / 48-pin LQFP(GL640USB-A)
- Special pad-design to support EPP/USB dual-interface application

## 2. Description

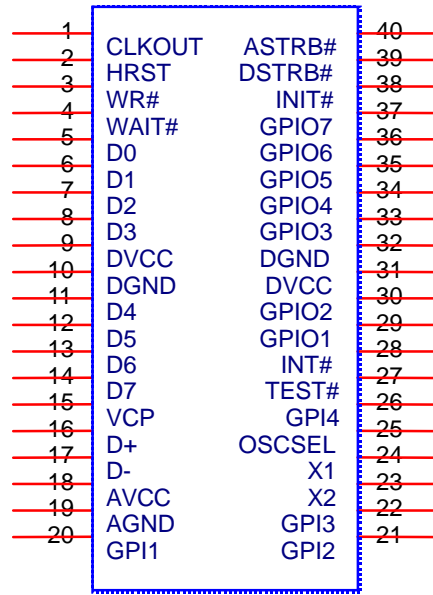
The GL640USB is a low-cost, single-chip embedded controller designed for connecting an IEEE 1284 parallel port peripheral to USB. It is suitable for applications where the IC is mounted on a PCB inside a product or stand-alone applications where the chip resides in a cable connecting a standard parallel port device to an USB-capable computer. The GL640USB firmware/driver will allow peripheral vendors to easily migrate parallel port device to USB.

## 3. IEEE 1284 Interface

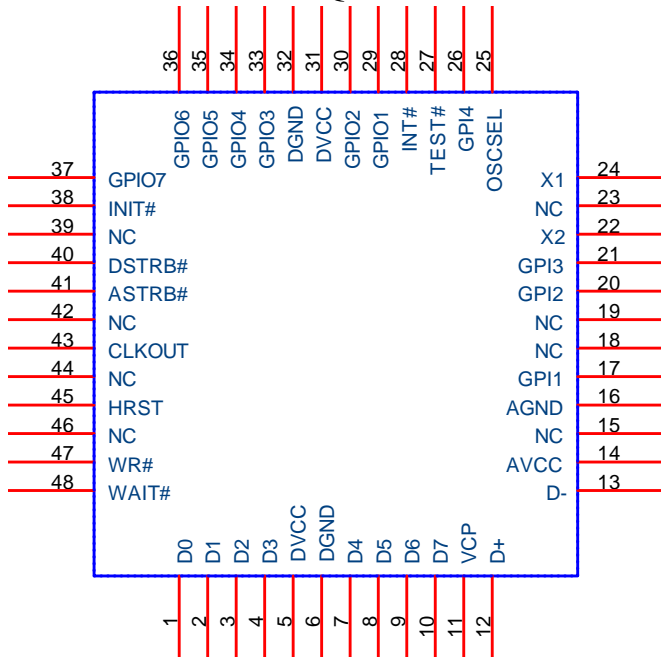


## 4. Pin Configuration

### 40-SOJ



### 48-LQFP

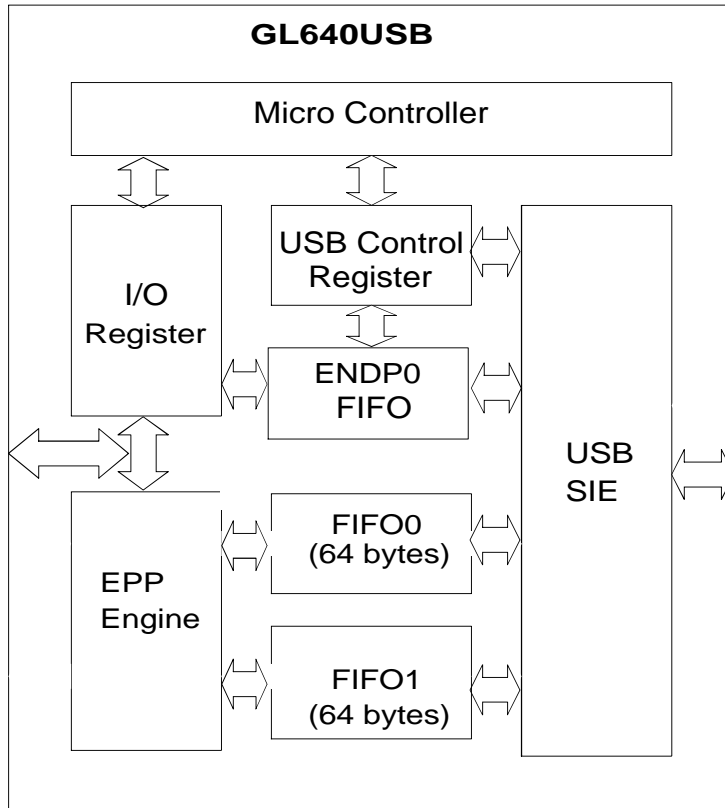


## 5. Pin Descriptions

GL640USB Pin No.	GL640USB-A Pin No.	SYMBOL	I/O	DESCRIPTION
1	43	CLKOUT	O	Clock output to external device The clock frequency is selected by setting DEVCTL register.
2	45	HRST	I	High active hardware reset, internal pull low With internal power-on reset circuit, this pin can be left unconnected or pulled low.
3	47	WR#	O	This active low signal indicates the EPP write operation.
4	48	WAIT#	I	this signal acts as the acknowledge from external EPP device, internal pull up
5-8	1-4	D0-D3	I/O	Address/Data bus bit 0 to bit 3
9	5	DVCC	-	Digital 5V input
10	6	DGND	-	Digital ground
11-14	7-10	D4-D7	I/O	Address/Data bus bit 4 to bit 7
15	11	VCP	-	3.3 V output. This output voltage is used to pull up D+ line to indicate the full-speed device.
16	12	D+	I/O	USB differential data
17	13	D-	I/O	USB differential data
18	14	AVCC	-	Analog 5V input
19	16	AGND	-	Analog ground
20	17	GPI1	I	General purpose input 1
21	20	GPI2	I	General purpose input 2
22	21	GPI3	I	General purpose input 3
23	22	X2	I/O	Crystal output
24	24	X1	I	12/48 MHz crystal or external clock input
25	25	OSCSEL	I	Oscillator Select, internal pull down 0=12MHz 1=48MHz
26	26	GPI4	I	General purpose input 4, internal pull up

27	27	TEST#	I	Mode select, internal pull up 1=Normal Mode 0=Chip Test Mode
28	28	INT#	I	this is an input pin to accept the interrupt signal from external EPP controller, internal pull up
29	29	GPIO1	I/O	this signal is served as an general purpose I/O pin, internal pull up
30	30	GPIO2	I/O	this signal is served as an general purpose I/O pin, internal pull up
31	31	DVCC	-	Digital 5V input
32	32	DGND	-	Digital ground
33	33	GPIO3	I/O	this signal is served as an general purpose I/O pin, internal pull up
34	34	GPIO4	I/O	this signal is served as an general purpose I/O pin, internal pull down
35	35	GPIO5	I/O	this signal is served as an general purpose I/O pin
36	36	GPIO6	I/O	this signal is served as an general purpose I/O pin
37	37	GPIO7	I/O	this signal is served as an general purpose I/O pin
38	38	INIT#	O	output signal to initialize the EPP device.
39	40	DSTRB#	O	EPP data strobe
40	41	ASTRB#	O	EPP address strobe

## 6. Block Diagram



This USB controller contains 3 sets of FIFO. The **ENDP0 FIFO** is an 8-byte FIFO to store/transmit the endp0 control packet. The **FIFO0** and **FIFO1** are a set of 64-byte ping-pong FIFO for endp1 and endp2. FIFO0 is used for DATA0 packet and FIFO1 is for DATA1 packet. When they are served as the endp1 FIFO, the LINKFF bit of FFCFG should be set and LINKDIR bit of FFCFG should be cleared. If they are allocated as the endp1 FIFO, then LINKDIR should be set.

An EPP engine is included to automatically accessing data to/from external device ASIC. Firmware can set the data length and hardware will count down to decide the last byte.

There are two sets of registers : **I/O register** and **USB register**. The **USB register** is controlled by micro-controller to implement the USB endpoint 0 functions. The **I/O register** is the main register to interface the data accessing between this controller and external device.





## **GL640USB, GL640USB-A**

---

The internal micro-controller is a proprietary RISC-like, Harvard-architecture 8-bit micro-controller.

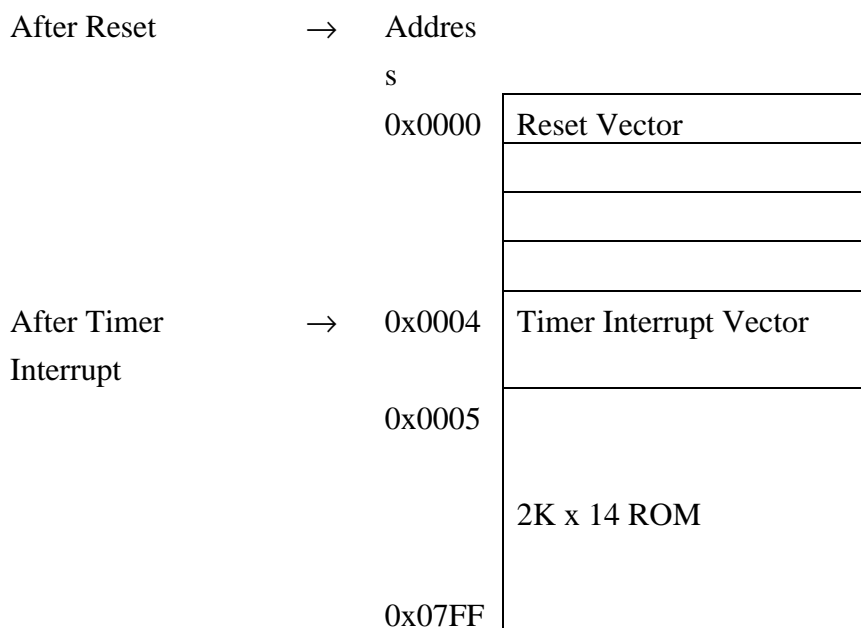
## 7. 8-BIT RISC MCU INSTRUCTION

### 7.1 MEMORY ORGANIZATION

The memory in the microcontroller is organized into user program memory in program ROM and data memory in SRAM space.

#### *Program Memory Organization*

The 11-bit Program Counter (PC) is capable of addressing 2.25K x 14 of program space. However, the program space of the GL640USB is 2K x 14. The program memory space is divided into two functional groups: Interrupt Vectors and program code. After a reset, the Program Counter points to location zero of the program space. After a timer interrupt, the Program Counter points the location 0x0004 of the program space.



*Figure 7-1 Program Memory Space*

### *Data Memory Organization*

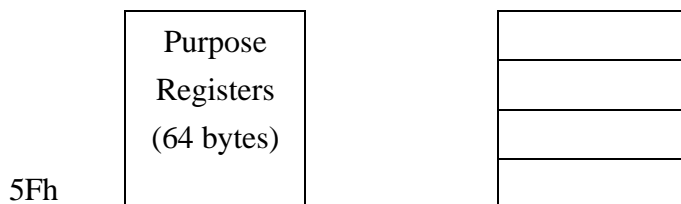
The data memory is partitioned into two Banks which contain the General Purpose Registers, MCU Function Registers and USB Function Registers. Bit RP0 is the bank select bit.

RP0 (STATUS<5>) = 1 → Bank 1

RP0 (STATUS<5>) = 0 → Bank 0

The lower locations of each Bank are reserved for MCU Function Registers. Above the MCU Function Registers are the USB Registers, I/O Registers and General Purpose Registers implemented as SRAM. Both Bank 0 and Bank 1 contain MCU Function Registers. USB Registers are located in Bank 0, and I/O Registers are located in Bank 1. Some “high use” MCU Function Registers from Bank 0 are mirrored in Bank 1 for code reduction and quicker access.

Data Memory Address		Data Memory Address	
00h	INDR	80h	INDR
01h	TIMER	81h	PSCON
02h	PCL	82h	PCL
03h	STATUS	83h	STATUS
04h	INDAR	84h	INDAR
05h		85h	
06h	PORT1	86h	PORT1CON
07h	PORT2	87h	PORT2CON
08h		88h	
09h		89h	
0Ah	PCHBUF	8Ah	PCHBUF
0Bh	INTEN	8Bh	INTEN
10~1Fh	<b>USB Registers</b>	90h~9Fh	<b>I/O Registers</b>
20h			
	General		



*Figure 7-2 Data Memory Space*

## **7.2 MCU FUNCTION REGISTERS**

Address	Name	Function
00h	INDR	Addressing this location will use the content of INDAR to address data memory (not a physical address)
01h	TIMER	Timer register
02h	PCL	Program Counter's low byte
03h	STATUS	Status register
04h	INDAR	Indirect address register
06h	PORT1	Port 1 data register
07h	PORT2	Port 2 data register
0Ah	PCHBUF	Write buffer of Program Counter's bit 10-8
80h	INDR	Addressing this location will use the content of INDAR to address data memory (not a physical address)
81h	PSCON	Prescaler control register
82h	PCL	Program Counter's low byte
83h	STATUS	Status register
84h	INDAR	Indirect address register
86h	PORT1CON	Port 1 direction control register
87h	PORT2CON	Port 2 direction control register
8Ah	PCHBUF	Write buffer of Program Counter's bit 10-8

*Table 7-1 MCU Function Register Summary*

INDR (Address 00h/80h)

INDR is not a physical register. Addressing INDR register will cause indirect addressing. Any

instruction using the INDF register actually accesses the register pointed by the INDAR register.

**TIMER** (Address 01h, Timer register)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TIMER7	TIMER6	TIMER5	TIMER4	TIMER3	TIMER2	TIMER1	TIMER0

After timer is enable, the timer will start to count up. The Timer Interrupt is generated when the TIMER register overflows from FFh to 00h.

Value on POR: “0 0 0 0 0 0 0 0”

**PCL** (Address 02h/82h, Program Counter's low byte)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0

The Program Counter (PC) is 11-bit wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte is not directly readable or writable and comes from PCHBUF. The GL600USB has a 4 level deep x 11-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is pushed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is popped in the event of a RETIA, RETI or a RET instruction execution. PCHBUF is not affected by a push or pop operation.

Value on POR: “0 0 0 0 0 0 0 0”

**Status** (Address 03h, Status register)

		R/W			R/W	R/W	R/W
		BS			ZO	HC	CA

**BS:** Bank Select

1: Bank 1 (80h-FFh)

0: Bank 0 (00h-7Fh)

**ZO:** Zero bit

1: The result of an arithmetic or logic operation is zero

0: The result of an arithmetic or logic operation is not zero

**HC:** Half Carry/Borrow bit

1: A carry-out from the 4<sup>th</sup> low order bit

0: No carry-out from the 4<sup>th</sup> low order bit

CA: Carry/Borrow bit

1: A carry-out from the most significant bit

0: No carry-out from the most significant bit

Value on POR: “- - 0 - - 0 0 0”

INDAR: (Address 04h/84h, Indirect address register)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
INDAR7	INDAR6	INDAR5	INDAR4	INDAR3	INDAR2	INDAR1	INDAR0

Any instruction using the INDF register actually accesses the register pointed by the INDAR register.

Value on POR: “x x x x x x x”<sup>[1]</sup>

Note 1: “x” means unknown

PORT1 (Address 06h, Port 1 data register)

			R/W	R/W	R/W	R/W	R/W
			PORT1.	PORT1.	PORT1.	PORT1.	PORT1.
			4	3	2	1	0

PORT1 is a 5-bit latch for Port 1. Reading the PORT1 register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations.

Value on POR: “- - - x x x x x”

PORT2 (Address 07h, Port 2 data register)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PORT2.	PORT2.	PORT2.	PORT2.	PORT2.	PORT2.	PORT2.	PORT2.
7	6	5	4	3	2	1	0

PORT2 is an 8-bit latch for Port 2. Reading the PORT2 register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations.

Value on POR: “x x x x x x x x”

PCHBUF (Address 0Ah/8Ah, Write buffer of Program Counter’s bit 10-8)

					R/W	R/W	R/W
--	--	--	--	--	-----	-----	-----

					PCHBU F2	PCHBU F1	PCHBU F0
--	--	--	--	--	-------------	-------------	-------------

Write buffer for upper 3-bit of Program Counter. The upper byte of Program Counter is not directly accessible. PCHBUF is a holding register for the PC[10:8] whose contents are transferred to the upper byte of the Program Counter.

Value on POR: “- - - - 0 0 0”

INTEN (Address 0Bh/8Bh, Interrupt enable register)

R/W		R/W			R/W		
GIE		TMROE N			TMROF		

GIE: Global interrupt enable bit

1: Enable all interrupts

0: Disable all interrupts

TMROEN: Timer overflow interrupt enable bit

1: Enable timer interrupt

0: Disable timer interrupt

TMROF: Timer overflow interrupt flag bit

1: Timer register has overflowed

0: Timer register did not overflow

Value on POR: “0 - 0 - - 0 - -”

PSCON (Address 81h, Prescaler control register)

				R/W	R/W	R/W	R/W
				PSDIS	PS2	PS1	PS0

PSEN: Prescaler disable bit

1: Disable prescaler for timer

0: Enable prescaler for timer

PS[2:0]: Prescaler rate select bits

Bit Value	Timer Rate
000	1:2
001	1:4

010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

Value on POR: “- - - 1 1 1 1”

PORT2CON (Address 86h, Port 1 direction control register)

				R/W	R/W	R/W	R/W
				P1CON3	P1CON2	P1CON1	P1CON0

All Port 1 pins have data direction control bits which can configure these pins as output or input. Setting a PORT1CON register bit put the corresponding output driver in a hi-impedance mode. Clearing a bit in the PORT1CON register puts the contents of the output latch on the selected pin. Value on POR: “- - - 1 1 1 1”

PORT2CON (Address 87h, Port 2 direction control register)

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
P2CON7	P2CON6	P2CON5	P2CON4	P2CON3	P2CON2	P2CON1	P2CON0

All Port 2 pins have data direction control bits which can configure these pins as output or input. Setting a PORT2CON register bit put the corresponding output driver in a hi-impedance mode. Clearing a bit in the PORT2CON register puts the contents of the output latch on the selected pin. Value on POR: “1 1 1 1 1 1 1 1”

### **7.3 I/O Register Summary**

Mnemonic	Address (Hex)	Description
IODEVCTL	90	device control for I/O
ENDPCTL	91	endpoint control
GPI	92	GPI3-1 register



INTFLG	93	interrupt flag
UINTEN	94	USB interrupt enable
USBFLG	95	USB interrupt flag
CTLDAT/ STSDAT	96	Control/status data buffer
CTLLEN/ STSLEN	97	Control/status data length
FFCFG	98	FIFO configuration register
FFCTL	99	FIFO control
FF0DAT	9A	data port for FIFO0
FF1DAT	9B	data port for FIFO1
LINE_L	9C	Read : max. length of TX data packet Write : line size (low byte)
LINE_H	9D	Read : received data length / Write : line size (high byte)
EPPCTL	9E	EPP control / status
EPPAD	9F	EPP interface AD0-7

**IODEVCTL** (offset 90h)

<i>R/W</i>	<i>R/O</i>	<i>R/O</i>	<i>R/O</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/O</i>
EXTCLK EN	NC	NC	GPI4	CKSEL2	CKSEL1	CKSEL0	CLKTYP

CLKTYP - select crystal frequency, this bit reflects the OSCSEL pin

0 - 12MHz, 1 - 48MHz

CKSEL[2:0] - select CLKOUT frequency, default=8MHz

000 - stop CLKOUT                      100 - 8MHz

001 - 24MHz                              101 - 6MHz

010 - 16MHz                              110 - 12KHz

011 - 12MHz

GPI4 - this bit reflects the status of GPI4 input pad

EXTCLKEN - Enable EXTCLK output

0 - set EXTCLK to tri-state              1 - enable EXTCLK

**ENDPCTL**(offset 91h)

				<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
				EP3STL	EP2STL	EP1STL	EP0STL

Default=8'h00

EPnSTL - Endpoint *n* is stalled

By setting EPnSTL to '1', endpoint *n* will respond with a STALL to incoming USB packet.

#### GPI (offset 92h)

					<i>R/O</i>	<i>R/O</i>	<i>R/O</i>
					GPI3	GPI2	GPI1

#### INTFLG ( offset 93h )

<i>R/O</i>	<i>R/WIC</i>	<i>R/WIC</i>	<i>R/WIC</i>	<i>R/O</i>	<i>R/O</i>	<i>R/WIC</i>	<i>R/WIC</i>
Reserved	RESUME	SUSPND	EPPINT	Reserved	EP3TX	DATARX	DATATX

Default=8'h00

This register is used to identify the exact interrupt event. When the external controller receives an interrupt, it should first read this register to check the interrupt event. Writing '1' to clear the individual interrupt bit.

DATATX - The endp1 transmits a data packet completely

DATARX - The endp2 receives a data packet

EP3TX - Endp3 transaction is detected

EPPINT -This bit means external EPP interrupt is detected.

The EPPTXEN/EPPTXEN is cleared by hardware when this bit is set.

SUSPND - USB suspend request is detected.

RESUME - USB resume request is detected.

#### UINTEN ( offset 94h )

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/O</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
FFINT	REMINT	SUSINT	DMAINT	Reserved	UINTEN	DRXINT	DTXINT

These are the interrupt enable bits of INTFLG register.

Default = 8'h00

#### USBFLG ( offset 95h )

					<i>R/WIC</i>	<i>R/O</i>	<i>R/O</i>
--	--	--	--	--	--------------	------------	------------

					STSTX	Reserved	Reserved
--	--	--	--	--	-------	----------	----------

Default=8'h00

STSTX - The interrupt endpoint 3 transmits a status packet completely

**CTLDAT/STSDAT ( offset 96h )**

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
CTLDAT7	CTLDAT6	CTLDAT5	CTLDAT4	CTLDAT3	CTLDAT2	CTLDAT1	CTLDAT0

Read: Pop data from endp0/endp3 FIFO.

Write: Push data into endp0/endp3 FIFO.

**CTLLEN/STSLEN ( offset 97h )**

	R/O	R/O	R/W	R/W	R/W	R/W	R/W
	RXSETUP	RXOUT	<b>DATOG</b>	CTLLEN3	CTLLEN2	CTLLEN1	CTLLEN0

For the received OUT/SETUP transaction on endp0/endp3, read this register to check the data size and data toggle, and then read the received data from CTLDAT/STSDAT register. To transmit data on endp0/endp3, first push data into endp0/endp3 FIFO, and then write this register to set data size and data toggle. Finally, turn on CTLTXEN or STSTXEN bit to enable the data transmission.

CTLLEN3-0 - Length of the received/transmitted endp0/endp3 data

DATOG - Data toggle of the received/transmitted endp0/endp3 data

RXOUT - The received transaction is an OUT transaction

RXSETUP - The received transaction is a SETUP transaction

**FFCFG ( offset 98h )**

		R/W	R/W	R/W	R/W	R/W	R/W
		TX64ONLY	TXNULL	DMARXEN/ EPPRXEN	DMATXEN/E PPTXEN	LINKDIR	LINKFF

Default=8'h00

This register is used to control TX/RX FIFO and DMA/EPP engine operation.

LINKFF - link RXFIFO to TXFIFO together to form ping-pong FIFO

LINKDIR - This bit is valid only when LINKFF is set to 1.

1: the ping-pong FIFO is used for end2

- 0: the ping-pong FIFO is used for end1
- DMATXEN - enable DMA/EPP engine to access data into TXFIFO
- DMARXEN - enable DMA/EPP engine to move data from RXFIFO into external Device
- TXNULL - Set this bit to transmit a zero-byte data on endp1  
This bit will be cleared by hardware automatically.
- TX64ONLY - Transmit 64-byte packet only  
When this bit is set, data left in FIFO will not be sent if it is not 64 bytes.

*Note: When LINKFF is set and LINKDIR is gonna to change, LINKFF should be cleared to '0' first and then change LINKDIR and set LINKFF again.*

**FFCTL** ( offset 99h )

W/O	W/O	W/O	R/W	R/W	R/W	R/W	R/W
FFORST	RXFFRST	TXFFRST	STSTXEN	DRXDIS/ DTX1EN	DTX0EN	CTLRXDIS	CTLTXEN

- CTLTXEN - enable endpoint 0 transmitting  
This bit is cleared by hardware when CTLTX interrupt is set.
- CTLRXDIS - disable endpoint 0 receiving
- DTXEN - turn on this bit to enable endpoint 1 data packet transmission.  
If LINKFF=1, it indicates the TXFIFO0 is ready.  
This bit is cleared by hardware when DATATX interrupt is set.
- DRXDIS - disable RXFIFO receiving  
If this bit is set, NAK will be responded to the OUT token.  
Data in RXFIFO is kept unchanged.  
If LINKFF=1, this bit means the TXFIFO1 is ready.
- STSTXEN - enable endpoint 3 transmitting  
This bit is cleared by hardware when STSTX interrupt is set.
- TXFFRST - reset TXFIFO, cleared by hardware itself.
- RXFFRST - reset RXFIFO, cleared by hardware itself
- FFORST - reset CTL/STS FIFO, cleared by hardware itself

**FF0DAT** ( offset 9Ah )

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
FFDAT7	FFDAT6	FFDAT5	FFDAT4	FFDAT3	FFDAT2	FFDAT1	FFDAT0

Read this register will pop data from FIFO0. Write this register will push data into FIFO0.

**FF1DAT** ( offset 9Bh )

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
FFDAT7	FFDAT6	FFDAT5	FFDAT4	FFDAT3	FFDAT2	FFDAT1	FFDAT0

Read this register will pop data from FIFO1. Write this register will push data into FIFO1.

**MAXLEN/LINE\_L** ( offset 9Ch )

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
LINE7	LINE6	LINE5	LINE4	LINE3	LINE2	LINE1	LINE0

Read:

MAXLEN - Equal to 64

Write:

LINE\_L - The low byte of EPP accessing length

**DATLEN / LINE\_H** ( offset 9Dh )

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
LINE15	LINE14	LINE13	LINE12	LINE11	LINE10	LINE9	LINE8

Read:

DATLEN - The length of data received by the RX endpoint

Write:

LINE\_H - The high byte of EPP accessing length

**EPPCTL** ( offset 9Eh )

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/O</i>	<i>R/W</i>
GLTHFLT	WRCHK	ACTIVE	A2CHK	A1CHK	A0CHK	INT_	ADOE

This register is available only in EPP mode.

ADOE - Set this bit to '1' can drive data in EPPAD register to pins D7-0

INT\_ - This bit reflects the status of INT\_ pin.

- AnCHK* - This bit is used to select the type of EPP read engine
  - 1 – EPP engine will check *An* pin before starting an EPP cycle.
  - 0 – EPP engine doesn't check *An* pin ,  $n=0\sim2$
- ACTIVE* - Select the *An* pin polarity
  - 1 – active high for data ready 0 – active low for data ready
- WRCHK* - Select EPP check status during write or read
  - 1- Check during EPP write cycle
  - 0 – check during EPP read cycle
- GLTHFLT* - Filter glitches on WAIT\_
  - 1- Enable filter, performance maybe slow down
  - 0 – Disable filter

**EPPAD** ( offset 9Fh )

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
EPPAD7	EPPAD6	EPPAD5	EPPAD4	EPPAD3	EPPAD2	EPPAD1	EPPAD0

This register is used to set/read pins D7-0.

To set D7-0, except writing EPPAD, the ADOE bit of EPPCTL register is set to 1

Read this register can get the status of pins D7-0.

## 7.4 USB Register Summary

Mnemonic	Offset	Description	Remarks
DEVCTL	10h	Device control register	
EVTFLG	11h	USB function interrupt flag	
DEVADR	12h	USB Device address	
RXCTL0	13h	Endpoint 0 RX control	
TXCTL0	14h	Endpoint 0 TX control	
CTLDAT	15h	Endpoint 0/3 FIFO data port	
MISC	16h	Miscellaneous register	
GPIO	17h	GPIO value	
GPIOCTL	18h	GPIO direction control	

### **DEVCTL** ( offset 10h )

			R/W	R/W	R/W	R/W	R/W
			DISGLUSB	EP0STL	WAKE	WAKEDIS	PWRDN

Default=8'h10

- PWRDN** - Power down mode
- If USB suspend is detected, firmware can set PWRDN to put the controller into power down mode. Power down mode stops oscillator and freezes all clocks at known states, and no more command can be executed.
- Hardware will automatically clear PWRDN upon hardware reset or interrupt event.
- WAKEDIS** - Disable remote wakeup capability
- WAKE** - Wake up host
- Write '1' to this bit will place USB bus to K state.
- EP0STL** - Endpoint 0 stall
- Endpoint 0 will respond with a STALL to a valid transaction.
- DISGLUSB** - Disable GL640USB, and bypass all EPP signals

When this bit is set to '1', D+ and D- pin will be driven to low so that no connect will be detected on the host side.

**EVTFLG ( offset 11h )**

		<i>W/O</i>	<i>R/WIC</i>	<i>R/WIC</i>	<i>R/WIC</i>	<i>R/WIC</i>	<i>R/WIC</i>
		S_CTLRX	WAKEUP	RESUME	SUSPD	EP0TX	EP0RX

This register is the main USB interrupt flag for endpoint 0 and power management. The firmware detects endpoint 0 transaction via EP0RX and EP0TX. If firmware can't handle the received endpoint 0 control data, then firmware should write '1' to S\_CTLRX and EP0RX. By this, the interrupt will redirect to the external DMA interrupt and the CTLRX bit of USBFLG register will be set. For power management, when a USB suspend is detected, the SUSPD bit will be set to '1'. If the USB host put the bus to 'K' state during suspend, then the RESUME bit will be set to '1'. If a remote-wake-up event is detected, then the WAKEUP bit will be set to '1'.

All those interrupt status bits can be written '1' to clear.

- EP0RX - Endpoint 0 receives a data packet.
- EP0TX - Endpoint 0 transmits a data packet completely.
- SUSPD - USB suspend detected
- RESUME - USB resume detected
- WAKEUP- remote-wake-up event is detected during suspend state

The event is

DMA mode: WAKEVT bit is set

EPP mode: INT\_ pin is asserted low

S\_CTLRX- Write 1 to set CTLRX bit of USBFLG

All the data in endp0/endp3 FIFO are left unchanged.

**DEVADR ( offset 12h )**

	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
	DEVADR6	DEVADR5	DEVADR4	DEVADR3	DEVADR2	DEVADR1	DEVADR0

This register is used to set USB device address.

Default=8'h00



**RXCTL0** ( offset 13h )

R/W	R/O	R/O	R/O	R/O	R/O	R/O	R/O
RXDIS	RXSETUP	RXOUT	RXSEQ	RXCNT3	RXCNT2	RXCNT1	RXCNT0

Default=8'h0e

This register is used to check the received data byte count, data toggle, and transaction token on endpoint 0. When the EP0RX interrupt is detected, firmware should first check this register to decide the received data is valid or not. At this time, RXDIS bit is set by hardware to prevent the current data in endp0 FIFO overwritten by next incoming data. After extracting data from endp0 FIFO, firmware should clear RXDIS to enable receiving capability on endp0.

RXCNT[3:0] - Received data byte count.

RXSEQ - 1 - The received data is DATA1  
0 - The received data is DATA0

RXOUT - 1 - The received token is OUT.

RXSETUP - 1 - The received token is SETUP.

RXDIS - Disable receiving capability on endpoint 0

Upon successfully receiving a data packet on endpoint 0, hardware will automatically set this bit to '1'. At this time, no more SETUP/OUT data on endpoint 0 can be accepted, hardware will respond with NAK.

0 - Endp0 FIFO is available for data receiving.

1 - Endp0 FIFO is not available

*Note: Firmware must take care the data toggle check and decide if current RX data is valid.*

**TXCTL0** ( offset 14h )

		R/W	R/W	R/W	R/W	R/W	R/W
		TXOE	TXSEQ	TXCNT3	TXCNT2	TXCNT1	TXCNT0

Default=8'h00

This register is used to control the data byte count, data toggle of the transmitted data on endpoint 0. If there is data to be sent to endpoint 0, firmware should first push data into FFDAT0 register, and then set the pushed byte count to TXCNT3-0, set the data toggle to TXSEQ, and finally turn on TXOE to enable the data transmission.

TXCNT3-0 - number of bytes to send  
 TXSEQ - 0 - TX DATA0 1 - TX DATA1  
 TXOE - ready to transmit control data

**CTLDAT ( offset 15h )**

<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
FFDAT7	FFDAT6	FFDAT5	FFDAT4	FFDAT3	FFDAT2	FFDAT1	FFDAT0

This register is the window to push/pop data from endp0/3 FIFO. Write to push FIFO, read to pop.

**MISC ( offset 16h )**

<i>R/O</i>	<i>R/O</i>	<i>R/O</i>	<i>R/O</i>	<i>R/O</i>	<i>W/O</i>	<i>R/W</i>	<i>R/W</i>
SUSPD	ADDR	DEFLT	POWER	Reserved	FFRST0	SF	SUS_DIS

Default = 8'h10

This register is mainly used in testing purpose.

SUS\_DIS - Disable suspend detection  
 SF - Short frame mode, used in suspend detection  
           0=normal mode, needs 3ms bus idle to enter suspend mode  
           1=short frame mode, needs only 200us to enter suspend mode  
 FFRST0 - Reset endpoint 0 FIFO read/write pointer.  
           Data in FIFO remain unchanged.  
 POWER - Device is in the powered state  
 DEFLT - Device is in the default state  
 ADDR - Device is in the address state  
 SUSPD - Device is in the suspend state

**GPIO ( offset 17h )**

	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
	GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1

**GPIOCTL ( offset 18h )**

	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>	<i>R/W</i>
	GPIO7OE	GPIO6OE	GPIO5OE	GPIO4OE	GPIO3OE	GPIO2OE	GPIO1OE

GPIO and GPIOCTL registers are available only in EPP mode. GPIOCTL register determines pin GPIO1-7 is input mode or output mode, GPIO register is used to set the output data to pin GPIO1-7 or read data from pin GPIO1-7 input.

GPIO $n$ OE -    1= set pin GPIO $n$  to output mode  
                  0= set pin GPIO $n$  to input mode

## 8. INSTRUCTION SET SUMMARY

### *Operand Field Descriptions*

Field	Description
r	Register address
A	Accumulator
i	Immediate data
b	Bit address within a 8-bit register

### *Instruction Set*

Mnemonic, Operands	Description	Cycle s	Flags Affected
<b>Arithmetic Operations</b>			
ADDAR r, A	Add r and A, $r \leftarrow r + A$	1	CA, HC, ZO
ADDAR A, r	Add A and r, $A \leftarrow A + r$	1	CA, HC, ZO
ADDAI i	Add A and i, $A \leftarrow A + i$	1	CA, HC, ZO
INCR r	Increment r, $r \leftarrow r + 1$	1	ZO
INCR A, r	Increment r, $A \leftarrow r + 1$	1	ZO
INCRSZ r	Increment r, $r \leftarrow r + 1$ , skip if ( $r = 0$ )	1 or 2	
INCRSZ A, r	Increment r, $A \leftarrow r + 1$ , skip if ( $A = 0$ )	1 or 2	
SUBAR r, A	Subtract A from r, $r \leftarrow r - A$	1	CA, HC, ZO
SUBAR A, r	Subtract A from r, $A \leftarrow r - A$	1	CA, HC, ZO
SUBIA i	Subtract A from i, $A \leftarrow i - A$	1	CA, HC,

			ZO
DECR r	Decrement r, $r \leftarrow r - 1$	1	ZO
DECR A, r	Decrement r, $A \leftarrow r - 1$	1	ZO
DECRSZ r	Decrement r, $r \leftarrow r - 1$ , skip if ( $r = 0$ )	1 or 2	
DECRSZ A, r	Decrement r, $A \leftarrow r - 1$ , skip if ( $A = 0$ )	1 or 2	
CLRR r	Clear r, $r \leftarrow 0$	1	ZO
CLRA	Clear A, $A \leftarrow 0$	1	ZO
NOP	No operation	1	
<b>Logical Operations</b>			
ANDAR r, A	And r and A, $r \leftarrow r \& A$	1	ZO
ANDAR A, r	And A and r, $A \leftarrow A \& r$	1	ZO
ANDAI i	And A and i, $A \leftarrow A \& i$	1	ZO
CMPR r	Complement r, $r \leftarrow r \wedge FF$	1	ZO
CMPR A, r	Complement r, $A \leftarrow r \wedge FF$	1	ZO
ORAR r, A	Inclusive OR r with A, $r \leftarrow r   A$	1	ZO
ORAR A, r	Inclusive OR A with r, $A \leftarrow A   r$	1	ZO
ORIA i	Inclusive OR i with A, $A \leftarrow A   i$	1	ZO
XORAR r, A	Exclusive OR r with A, $r \leftarrow r \wedge A$	1	ZO
XORAR A, r	Exclusive OR A with r, $A \leftarrow A \wedge r$	1	ZO
XORIA i	Exclusive OR i with A, $A \leftarrow A \wedge i$	1	ZO
<b>Bit-wise Operations</b>			
BCR r, b	Bit clear r, $r.b \leftarrow 0$	1	
BSR r, b	Bit set r, $r.b \leftarrow 1$	1	
BTRSC r, b	Bit test r, skip if ( $r.b = 0$ )	1 or 2	
BTRSS r, b	Bit test r, skip if ( $r.b = 1$ )	1 or 2	
<b>Data Movement Operations</b>			
MOV r, A	Move A into r, $r \leftarrow A$	1	
MOV A, r	Move r into A, $A \leftarrow r$	1	ZO
MOVIA i	Move i into A, $A \leftarrow i$	1	
<b>Shift Operations</b>			
SWAPR r	Swap high and low nibbles in r	1	

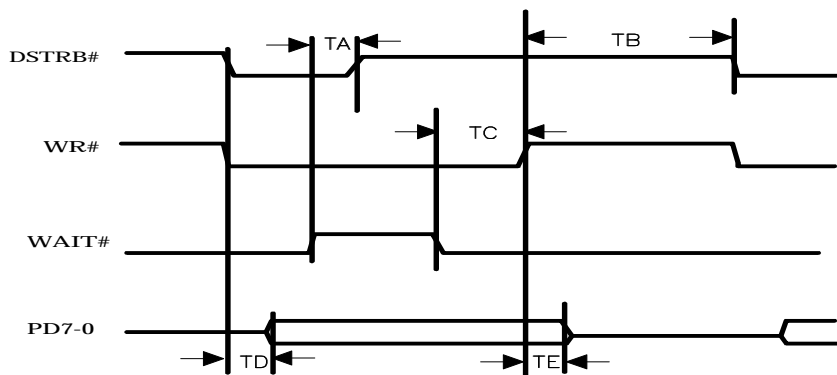
RLR r	Rotate r left through C	1	CA
RRR r	Rotate r right through C	1	CA
<b>Control Transfer Operations</b>			
CALL i	Call subroutine	2	
JUMP i	Jump to address	2	
RETIA	Return and load i to A	2	
RETI	Return from interrupt	2	
RET	Return from subroutine	2	

## 9. Advantages of GL640USB

Advantage	Description
EMI consideration	<ol style="list-style-type: none"> <li>1. Operates at 12MHz frequency to reduce EMI radiation</li> <li>2. Slew rate controlled output pads: All the transition on output pads are well controlled to reduce the EMI radiation.</li> </ol>
Cost Saving	<ol style="list-style-type: none"> <li>1. Provide clock output, no other crystal required. The clock frequency can be 24, 16, 12, 8, 6 MHz</li> <li>2. Build in power-on reset circuit. No reset circuit required.</li> <li>3. Provide 3.3V output to pull-up USB bus</li> </ol>
Flexibility	<ol style="list-style-type: none"> <li>1. Provide self-Isolation capability. With this capability, a device can be designed to support both EPP and USB interface.</li> <li>2. Provide additional 7 GPIO pins and 4 GPI pins. These pins can be used as testing purpose, status indication, product selection, ...and so on.</li> <li>3. Customized firmware: Contains a RISC controller with 2K ROM, user can developing its own firmware to achieve best performance and satisfy special applications</li> <li>4. Support 93C46 to provide external VID and PID</li> </ol>
Performance	<ol style="list-style-type: none"> <li>1. Build in hardware EPP engine and 2 sets of 64 byte ping-pong FIFO, the performance for Bulk transfer will be up to 1.1 Mbytes /sec</li> <li>2. Balanced architecture: Both the Bulk-In and Bulk-Out can achieve 1.1 Mbytes/sec transfer rate.</li> </ol>

## 10. EPP Timing

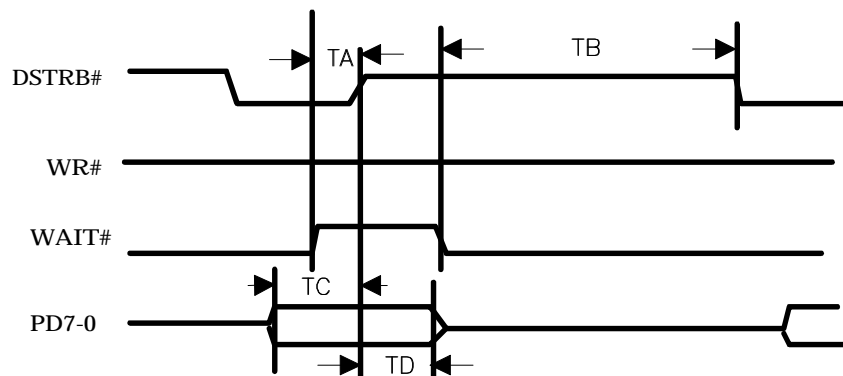
### 10.1 EPP Burst Data Write



NAME	Description	Min	Max	UNIT
TA	Wait# de-assert to DSTRB# de-assert	42	83	ns
TB	End of EPP write cycle to beginning of next EPP write cycle	83	83	ns
TC	Wait# assert to WR# de-assert	42	83	ns
TD	WR# assert to data available	0	10	ns
TE	WR# de-assert to data invalid	2		ns



## 10.2 EPP Burst Data Read



NAME	Description	Min	Max	UNIT
TA	Wait# de-assert to DSTRB# de-assert	42	83	ns
TB	Wait# assert to next DSTRB# assert	83	124	ns
TC	Data available to DSTRB# deasserted	5		ns
TD	DSTRB# de-assert to data invalid	5		ns

## 11. Electrical Characteristics

### 11.1 Absolute Maximum Ratings (Voltages referenced to GND)

SYMBOL	Description	MIN	MAX
VCC	DC supply voltage	-0.5V	+7V
V <sub>I</sub>	DC input voltage	-0.5V	VCC+0.5V
V <sub>I/O</sub>	DC input voltage range for I/O	-0.5V	VCC+0.5V
V <sub>AI/O</sub>	DC input voltage for USB D+/D- pins	-0.5	VCC+0.5V
V <sub>I/OZ</sub>	DC voltage applied to outputs in High Z state	-0.5V	VCC+0.5V
V <sub>ESD</sub>	static discharge voltage	4000V	

### 11.2 DC Characteristics (Digital Pins)

SYMBOL	Description	MIN	TYP	MAX	UNIT
P <sub>D</sub>	Power Dissipation	9	10	11	mA
V <sub>DD</sub>	Power Supply Voltage	4.5	5.0	5.5	V
I <sub>O</sub>	DC output sink current excluding D+/D-/VCC/GND	4			mA
V <sub>IL</sub>	LOW level input voltage			0.9	V
V <sub>IH</sub>	HIGH level input voltage	2.0			V
V <sub>TLH</sub>	LOW to HIGH threshold voltage	1.7	1.8	2.0	V
V <sub>THL</sub>	HIGH to LOW threshold voltage	0.9	1.1	1.3	V
V <sub>HYS</sub>	Hysteresis voltage	0.6	0.7	0.85	V
V <sub>OL</sub>	LOW level output voltage when I <sub>OL</sub> =4mA			0.4	V
V <sub>OH</sub>	HIGH level output voltage when I <sub>OH</sub> =4mA	2.4			V
I <sub>OLK</sub>	Leakage current for pads with internal pull up or pull down resistor			38	μA
R <sub>DN</sub>	Pad internal pulldown resister ( <i>Note 1</i> )	20K	28K	41K	Ohms
R <sub>UP</sub>	Pad internal pullup resister ( <i>Note 2</i> )	24K	33K	48K	Ohms

*Note 1 : Pins with internal pullup resister : PIN3, 4, 26, 27, 28, 29, 30, 33, 38, 39, 40*

*Note 2 : Pins with internal pulldown resister: PIN25, 34*

### **11.3 DC Characteristics (VCP/D+/D-)**

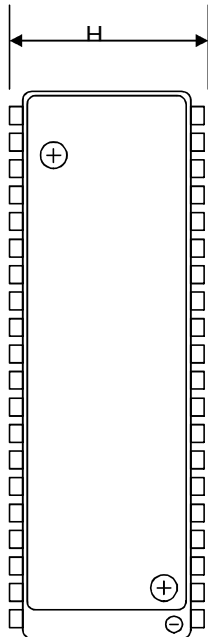
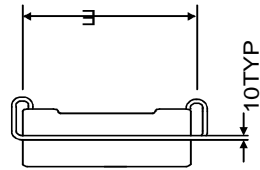
SYMBOL	Description	MIN	TYP	MAX	UNIT
V <sub>3.3</sub>	VCP regulator output	3.0	3.3	3.6	V
I <sub>3.3</sub>	VCP maximum supply current	27	41	56	mA
V <sub>OL</sub>	D+/D- static output LOW(R <sub>L</sub> of 1.5K to 3.6V )			0.3	V
V <sub>OH</sub>	D+/D- static output HIGH (R <sub>L</sub> of 15K to GND )	2.8		3.6	V
V <sub>DI</sub>	Differential input sensitivity	0.2			V
V <sub>CM</sub>	Differential common mode range	0.8		2.5	V
V <sub>SE</sub>	Single-ended receiver threshold	0.2			V
C <sub>IN</sub>	Transceiver capacitance			20	pF
I <sub>LO</sub>	Hi-Z state data line leakage	-10		+10	μA
Z <sub>DRV</sub>	Driver output resistance	28		43	Ohms

## 11.4 Switching Characteristics

SYMBOL	Description	MIN	TYP	MAX	UNIT
$F_{X1}$	X1 crystal frequency	11.97	12	12.03	MHz
$T_{CYC}$	X1 cycle time		83.3		ns
$T_{X1L}$	X1 clock LOW time	$0.45T_{CYC}$			ns
$T_{X1H}$	X1 clock HIGH time	$0.45T_{CYC}$			ns
$T_{r30pf}$	Output pad rise time from 10% to 90% swing with 30pF loading	6	8	12	ns
$T_{f30pf}$	Output pad fall time from 10% to 90% swing with 30pF loading	5.3	7	10.4	ns
$T_{r50pf}$	Output pad rise time from 10% to 90% swing with 50pF loading	9.5	13	19	ns
$T_{f50pf}$	Output pad fall time from 10% to 90% swing with 50pF loading	7.7	10	16	ns
$T_{rUSB}$	D+/D- rise time with 50pF loading	4		20	ns
$T_{fUSB}$	D+/D- fall time with 50pF loading	4		20	ns

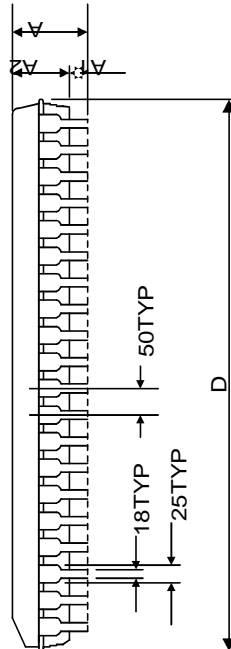
## 12. Package

### 40-SOJ



UNIT : MIL

SYMBOLS	MIN	NOR	MAX.
A	130	134	138
A1	0.24	-	-
A2	106	110	114
D	1025 BSC.		
E	400 BSC.		
H	430	440	450



48-LQFP

SYVBOIS	MIN	MAX
A		1.6
A1	0.05	0.15
A2	1.35	1.45
C1	0.09	0.16
D	9.00BSC	
D1	7.00BSC	
E	9.00BSC	
E1	7.00BSC	
e	0.5BSC	
b	0.17	0.27
L	0.45	0.75
L1	1 REF	

