STMicroelectronics - UPSD3422E-40U6 Datasheet

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8032
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, IrDA, SPI, UART/USART, USB
Peripherals	LVD, POR, PWM, WDT
Number of I/O	46
Program Memory Size	80KB (80K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3422e-40u6

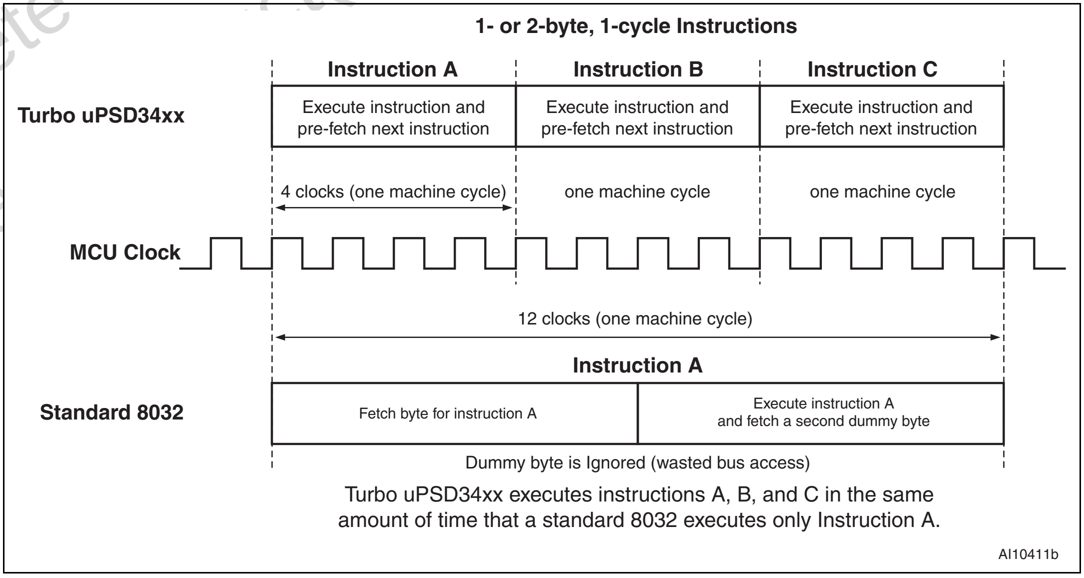
5 8032 MCU core performance enhancements

Before describing performance features of the UPSD34xx, let us first look at standard 8032 architecture. The clock source for the 8032 MCU creates a basic unit of timing called a machine-cycle, which is a period of 12 clocks for standard 8032 MCUs. The instruction set for traditional 8032 MCUs consists of 1, 2, and 3 byte instructions that execute in different combinations of 1, 2, or 4 machine-cycles. For example, there are one-byte instructions that execute in one machine-cycle (12 clocks), one-byte instructions that execute in four machine-cycles (48 clocks), two-byte, two-cycle instructions (24 clocks), and so on. In addition, standard 8032 architecture will fetch two bytes from program memory on almost every machine-cycle, regardless if it needs them or not (dummy fetch). This means for one-byte, one-cycle instructions, the second byte is ignored. These one-byte, one-cycle instructions account for half of the 8032's instructions (126 out of 255 opcodes). There are inefficiencies due to wasted bus cycles and idle bus times that can be eliminated.

The UPSD34xx 8032 MCU core offers increased performance in a number of ways, while keeping the exact same instruction set as the standard 8032 (all opcodes, the number of bytes per instruction, and the native number a machine-cycles per instruction are identical to the original 8032). The first way performance is boosted is by reducing the machine-cycle period to just 4 MCU clocks as compared to 12 MCU clocks in a standard 8032. This shortened machine-cycle improves the instruction rate for one- or two-byte, one-cycle instructions by a factor of three (*Figure 6 on page 33*) compared to standard 8051 architectures, and significantly improves performance of multiple-cycle instruction types.

The example in *Figure 6 on page 33* shows a continuous execution stream of one- or two-byte, one-cycle instructions. The 5 V UPSD34xx will yield 10 MIPS peak performance in this case while operating at 40 MHz clock rate. In a typical application however, the effective performance will be lower since programs do not use only one-cycle instructions, but special techniques are implemented in the UPSD34xx to keep the effective MIPS rate as close as possible to the peak MIPS rate at all times. This is accomplished with an instruction pre-fetch queue (PFQ), a branch cache (BC), and a 16-bit program memory bus as shown in *Figure 7 on page 34*.

Figure 6. Comparison of UPSD34xx with standard 8032 performance


1- or 2-byte, 1-cycle Instructions
Instruction A
Instruction B
Instruction C
Turbo uPSD34xx
Execute instruction and pre-fetch next instruction
Execute instruction and pre-fetch next instruction
Execute instruction and pre-fetch next instruction
4 clocks (one machine cycle)
one machine cycle
one machine cycle
MCU Clock
12 clocks (one machine cycle)
Instruction A
Standard 8032
Fetch byte for instruction A
Execute instruction A and fetch a second dummy byte
Dummy byte is Ignored (wasted bus access)
AI10411b

Turbo uPSD34xx executes instructions A, B, and C in the same amount of time that a standard 8032 executes only Instruction A.

12 Debug unit

The 8032 MCU module supports run-time debugging through the JTAG interface. This same JTAG interface is also used for In-System Programming (ISP) and the physical connections are described in the PSD module section, *Section 28.6.1: JTAG ISP and JTAG debug on page 257*.

Debugging with a serial interface such as JTAG is a non-intrusive way to gain access to the internal state of the 8032 MCU core and various memories. A traditional external hardware emulator cannot be completely effective on the UPSD34xx because of the Pre-Fetch Queue and Branch Cache. The nature of the PFQ and BC hide the visibility of actual program flow through traditional external bus connections, thus requiring on-chip serial debugging instead.

Debugging is supported by Windows PC based software tools used for 8051 code development from 3rd party vendors listed at www.st.com/psm. Debug capabilities include:

- Halt or start MCU execution
- Reset the MCU
- Single step
- 3 match breakpoints
- 1 range breakpoint (inside or outside range)
- Program tracing
- Read or modify MCU core registers, DATA, IDATA, SFR, XDATA, and code
- External debug event pin, input or output

Some key points regarding use of the JTAG debugger.

- The JTAG debugger can access MCU registers, data memory, and code memory while the MCU is executing at full speed by cycle-stealing. This means “watch windows” may be displayed and periodically updated on the PC during full speed operation. Registers and data content may also be modified during full speed operation.
- There is no on-chip storage for Program Trace data, but instead this data is scanned from the UPSD34xx through the JTAG channel at run-time to the PC host for proccessing. As such, full speed program tracing is possible only when the 8032 MCU is operating below approximately one MIPS of performance. Above one MIPS, the program will not run real-time while tracing. One MIPS performance is determined by the combination of choice for MCU clock frequency, and the bit settings in SFR registers BUSCON and CCON0.
- Breakpoints can optionally halt the MCU, and/or assert the external Debug Event pin.
- Breakpoint definitions may be qualified with read or write operations, and may also be qualified with an address of code, SFR, DATA, IDATA, or XDATA memories.
- Three breakpoints will compare an address, but the fourth breakpoint can compare an address and also data content. Additionally, the fouth breakpoint can be logically combined (AND/OR) with any of the other three breakpoints.
- The Debug Event pin can be configured by the PC host to generate an output pulse for external triggering when a break condition is met. The pin can also be configured as an event input to the breakpoint logic, causing a break on the falling-edge of an external event signal. If not used, the Debug Event pin should be pulled

period (12 / $f_{OSC,}$ seconds). However, if MCU_CLK is divided by the SFR CCON0, then the sample period must be calculated based on the resultant, longer, MCU_CLK frequency. In this case, an external clock signal on pins C0, C1, or T2 should have a duration longer than one MCU machine cycle, t_{MACH_CYC}. *Section 19.5: Watchdog timer, WDT on page 92* explains how to estimate t_{MACH_CYC}.

Table 56. TCON: Timer control register (SFR 88h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Table 57. TCON register bit definition

Bit	Symbol	R/W	Definition
7	TF1	R	Timer 1 overflow interrupt flag. Set by hardware upon overflow. Automatically cleared by hardware after firmware services the interrupt for Timer 1.
6	TR1	R,W	Timer 1 run control. 1 = Timer/Counter 1 is on, 0 = Timer/Counter 1 is off.
5	TF0	R	Timer 0 overflow interrupt flag. Set by hardware upon overflow. Automatically cleared by hardware after firmware services the interrupt for Timer 0.
4	TR0	R,W	Timer 0 run control. 1 = Timer/Counter 0 is on, 0 = Timer/Counter 0 is off.
3	IE1	R	Interrupt flag for external interrupt pin, EXTINT1. Set by hardware when edge is detected on pin. Automatically cleared by hardware after firmware services EXTINT1 interrupt.
2	IT1	R,W	Trigger type for external interrupt pin EXTINT1. 1 = falling edge, 0 = low-level
1	IE0	R	Interrupt flag for external interrupt pin, EXTINT0. Set by hardware when edge is detected on pin. Automatically cleared by hardware after firmware services EXTINT0 interrupt.
0	IT0	R,W	Trigger type for external interrupt pin EXTINT0. 1 = falling edge, 0 = low-level

20.3 SFR, TCON

Timer 0 and Timer 1 share the SFR, TCON, that controls these timers and provides information about them. See *Table 56 on page 97*.

Bits IE0 and IE1 are not related to Timer/Counter functions, but they are set by hardware when a signal is active on one of the two external interrupt pins, EXTINT0 and EXTINT1. For system information on all of these interrupts, see *Table 17 on page 63*, Interrupt Summary.

Bits IT0 and IT1 are not related to Timer/Counter functions, but they control whether or not the two external interrupt input pins, EXTINT0 and EXTINT1 are edge or level triggered.

20.4 SFR, TMOD

Timer 0 and Timer 1 have four modes of operation controlled by the SFR named TMOD (*Table 58*).

20.5 Timer 0 and Timer 1 operating modes

The “Timer” or “Counter” function is selected by the $C/\overline{T}$ control bits in TMOD. The four operating modes are selected by bit-pairs M[1:0] in TMOD. Modes 0, 1, and 2 are the same for both Timer/Counters. Mode 3 is different.

20.5.1 Mode 0

Putting either Timer/Counter into Mode 0 makes it an 8-bit Counter with a divide-by-32 prescaler. *Figure 24* shows Mode 0 operation as it applies to Timer 1 (same applies to Timer 0).

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all '1s' to all '0s,' it sets the Timer Interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or EXTINT1 = 1. (Setting GATE = 1 allows the Timer to be controlled by external input pin, EXTINT1, to facilitate pulse width measurements). TR1 is a control bit in the SFR, TCON. GATE is a bit in the SFR, TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag, TR1, does not clear the registers.

Mode 0 operation is the same for the Timer 0 as for Timer 1. Substitute TR0, TF0, C0, TL0, TH0, and EXTINT0 for the corresponding Timer 1 signals in *Figure 24*. There are two different GATE Bits, one for Timer 1 and one for Timer 0.

20.5.2 Mode 1

Mode 1 is the same as Mode 0, except that the Timer register is being run with all 16 bits.

20.5.3 Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in *Figure 25 on page 100*. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset with firmware. The reload leaves TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

20.5.4 Mode 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in *Figure 26 on page 100*. TL0 uses the Timer 0 control Bits: $C/\overline{T}$, GATE, TR0, and TF0, as well as the pin EXTINT0. TH0 is locked into a timer function (counting at a rate of 1/12 f_{OSC}) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the “Timer 1“ interrupt flag.

Mode 3 is provided for applications requiring an extra 8-bit timer on the counter (see *Figure 26 on page 100*). With Timer 0 in Mode 3, a UPSD34xx device can look like it has three Timer/Counters (not including the PCA). When Timer 0 is in Mode 3, Timer 1 can be

20.6.1 Capture mode

In Capture mode there are two options which are selected by the bit EXEN2 in T2CON. *Figure 27 on page 105* illustrates Capture mode.

If EXEN2 = 0, then Timer 2 is a 16-bit timer if $C/\overline{T2}$ = 0, or it is a 16-bit counter if $C/\overline{T2}$ = 1, either of which sets the interrupt flag bit TF2 upon overflow.

If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input pin T2X causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2X causes interrupt flag bit EXF2 in T2CON to be set. Either flag TF2 or EXF2 will generate an interrupt and the MCU must read both flags to determine the cause. Flags TF2 and EXF2 are not automatically cleared by hardware, so the firmware servicing the interrupt must clear the flag(s) upon exit of the interrupt service routine.

20.6.2 Auto-reload mode

In the Auto-reload mode, there are again two options, which are selected by the bit EXEN2 in T2CON. *Figure 28 on page 106* shows Auto-reload mode.

If EXEN2 = 0, then when Timer 2 counts up and rolls over from FFFFh it not only sets the interrupt flag TF2, but also causes the Timer 2 registers to be reloaded with the 16-bit value contained in registers RCAP2L and RCAP2H, which are preset with firmware.

If EXEN2 = 1, then Timer 2 still does the above, but with the added feature that a 1-to-0 transition at external input T2X will also trigger the 16-bit reload and set the interrupt flag EXF2. Again, firmware servicing the interrupt must read both TF2 and EXF2 to determine the cause, and clear the flag(s) upon exit.

Note: The UPSD34xx does not support selectable up/down counting in Auto-reload mode (this feature was an extension to the original 8032 architecture).

Table 60. T2CON: Timer 2 control register (SFR C8h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$

Table 61. T2CON register bit definition

Bit	Symbol	R/W	Definition
7	TF2	R,W	Timer 2 flag, causes interrupt if enabled. TF2 is set by hardware upon overflow. Must be cleared by firmware. TF2 will not be set when either RCLK or TCLK =1.
6	EXF2	R,W	Timer 2 flag, causes interrupt if enabled. EXF2 is set when a capture or reload is caused by a negative transition on T2X pin and EXEN2 = 1. EXF2 must be cleared by firmware.
5	RCLK(1)	R,W	UART0 Receive Clock control. When RCLK = 1, UART0 uses Timer 2 overflow pulses for its receive clock in Modes 1 and 3. RCLK=0, Timer 1 overflow is used for its receive clock

Figure 36. UART mode 3, block diagram

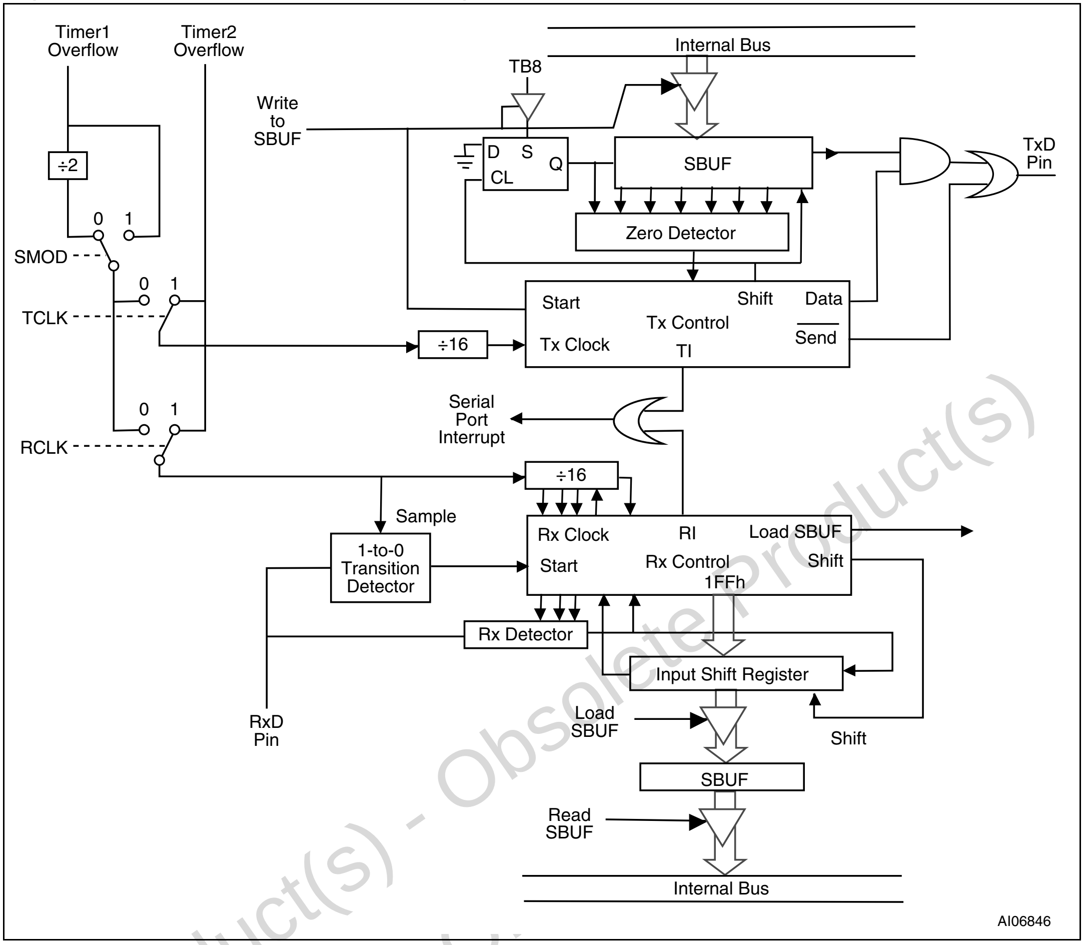
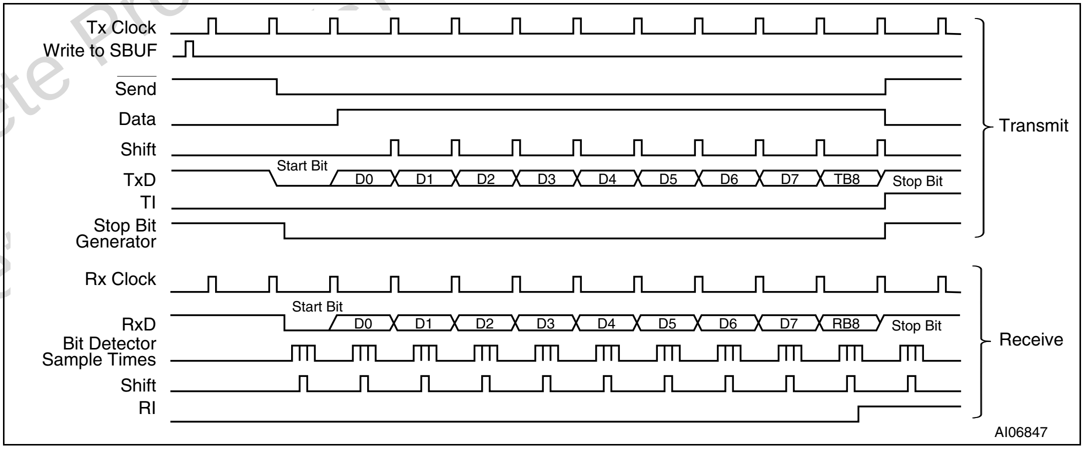


Figure 37. UART mode 3, timing diagram



23.12 I^2C Start sample setting (S1SETUP)

The S1SETUP register (*Table 84*) determines how many times an I^2C bus Start condition will be sampled before the SIOE validates the Start condition, giving the SIOE the ability to reject noise or illegal transmissions.

Because the minimum duration of an Start condition varies with I^2C bus speed (f_{SCL}), and also because the UPSD34xx may be operated with a wide variety of frequencies (f_{OSC}), it is necessary to scale the number of samples per Start condition based on f_{OSC} and f_{SCL}.

In Slave mode, the SIOE recognizes the beginning of a Start condition when it detects a '1'-to-'0' transition on the SDA bus line while the SCL line is high (see *Figure 41 on page 125*). The SIOE must then validate the Start condition by sampling the bus lines to ensure SDA remains low and SCL remains high for a minimum amount of hold time, t_{HLDSTA}. Once validated, the SIOE begins receiving the address byte that follows the Start condition.

If the EN_SS Bit (in the S1SETUP register) is not set, then the SIOE will sample only once after detecting the '1'-to-'0' transition on SDA. This single sample is taken $1/f_{OSC}$ seconds after the initial 1-to-0 transition was detected. However, more samples should be taken to ensure there is a valid Start condition.

To take more samples, the SIOE should be initialized such that the EN_SS Bit is set, and a value is written to the SMPL_SET[6:0] field of the S1SETUP register to specify how many samples to take. The goal is to take a good number of samples during the minimum Start condition hold time, t_{HLDSTA}, but no so many samples that the bus will be sampled after t_{HLDSTA} expires.

Table 86 on page 134 describes the relationship between the contents of S1SETUP and the resulting number of I^2C bus samples that SIOE will take after detecting the 1-to-0 transition on SDA of a Start condition.

Important note: Keep in mind that the time between samples is always $1/f_{OSC}$.

The minimum Start condition hold time, t_{HLDSTA}, is different for the three common I^2C speed categories per *Table 87 on page 134*.

Table 84. S1SETUP: I^2C Start condition sample setup register (SFR DBh, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EN_SS	SMPL_SET[6:0]						

Table 85. S1SETUP register bit definition

Bit	Symbol	R/W	Function
7	EN_SS	R/W	Enable Sample Setup EN_SS = 1 will force the SIOE to sample(1) a Start condition on the bus the number of times specified in SMPL_SET[6:0]. EN_SS = 0 means the SIOE will sample(1) a Start condition only one time, regardless of the contents of SMPL_SET[6:0].
6:0	SMPL_SET [6:0]	–	Sample Setting Specifies the number of bus samples(1) taken during a Start condition. See *Table 86* for values.

1. Sampling SCL and SDA lines begins after '1'-to-'0' transition on SDA occurred while SCL is high. Time between samples is $1/f_{OSC}$.

```
Enable I2C SIOE
    –   SFR S1CON.INI1 = 1
Transmit Address and R/W bit = 1 to Slave
    –   Is bus not busy? (SFR S1STA.BBUSY = 0?)
    <If busy, then test until not busy>
    –   SFR S1DAT[7:0] = Load Slave Address # 01h
    –   SFR S1CON.STA = 1, send Start on bus
    <bus transmission begins>
Enable All Interrupts and go do something else
    –   SFR IE.EA = 1
```

Slave-Transmitter

```
Disable all interrupts
    –   SFR IE.EA = 0
Set pointer to global data xmit buffer, set count
    –   *xmit_buf = *pointer to data
    –   buf_length = number of bytes to xmit
Set global variables to indicate Master-Xmitter
    –   I2C_master = 0, I2C_xmitter = 1
Enable SIOE
    –   SFR S1CON.INI1 = 1
Prepare to Xmit first data byte
    –   SFR S1DAT[7:0] = xmit_buf[0]
Enable All Interrupts and go do something else
    –   SFR IE.EA = 1
```

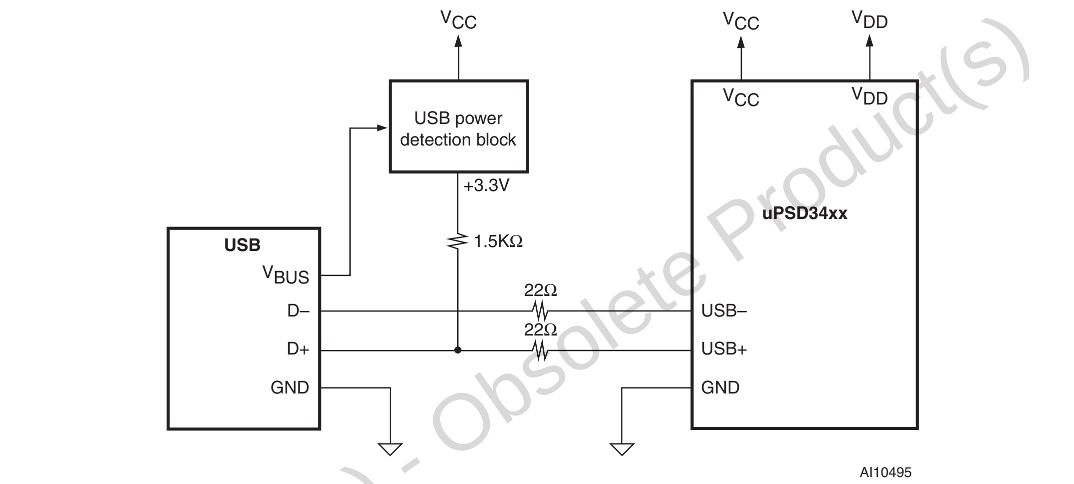
Slave-Receiver

```
Disable all interrupts
    –   SFR IE.EA = 0
Set pointer to global data recv buffer, set count
    –   *recv_buf = *pointer to data
    –   buf_length = number of bytes to recv
Set global variables to indicate Master-Xmitter
    –   I2C_master = 0, I2C_xmitter = 0
Enable SIOE
    –   SFR S1CON.INI1 = 1
Enable All Interrupts and go do something else
    –   SFR IE.EA = 1
```

25.5 Typical connection to USB

Connecting the UPSD34xx to the USB is simple and straightforward. *Figure 55* shows a typical self-powered example requiring only three resistors and a USB power detection circuit. The USB power detection circuit detects when the device has been connected to the USB. When V_{BUS} is detected, it switches 3.3 V to the pull-up resistor on the D+ line. Per the USB specification, the pull-up resistor on D+ is required to signal to the upstream USB port when a full speed device has been connected to the bus. The resistors in series in the D+ and D– lines are recommended per the USB specification to reduce transients on the data lines.

Figure 55. Typical self powered example



26 Analog-to-digital convertor (ADC)

The ADC unit in the UPSD34xx is a SAR type ADC with an SAR register, an auto-zero comparator and three internal DACs. The unit has 8 input channels with 10-bit resolution. The A/D converter has its own AV_{REF} input (80-pin package only), which specifies the voltage reference for the A/D operations. The analog to digital converter (A/D) allows conversion of an analog input to a corresponding 10-bit digital value. The A/D module has eight analog inputs (P1.0 through P1.7) to an 8x1 multiplexor. One ADC channel is selected by the bits in the configuration register. The converter generates a 10-bits result via successive approximation. The analog supply voltage is connected to the AV_{REF} input, which powers the resistance ladder in the A/D module.

The A/D module has 3 registers, the control register ACON, the A/D result register ADAT0, and the second A/D result register ADAT1. The ADAT0 register stores Bits 0.. 7 of the converter output, Bits 8.. 9 are stored in Bits 0..1 of the ADAT1 register. The ACON register controls the operation of the A/D converter module. Three of the bits in the ACON register select the analog channel inputs, and the remaining bits control the converter operation.

ADC channel pin input is enabled by setting the corresponding bit in the P1SFS0 and P1SFS1 registers to '1' and the channel select bits in the ACON register.

The ADC reference clock (ADCCLK) is generated from f_{OSC} divided by the divider in the ADCPS register. The ADC operates within a range of 2 to 16MHz, with typical ADCCLK frequency at 8MHz.

The conversion time is 4µs typical at 8MHz.

The processing of conversion starts when the Start Bit ADST is set to '1.' After one cycle, it is cleared by hardware. The ADC is monotonic with no missing codes. Measurement is by continuous conversion of the analog input. The ADAT register contains the results of the A/D conversion. When conversion is complete, the result is loaded into the ADAT. The A/D Conversion Status Bit ADSF is set to '1.' The block diagram of the A/D module is shown in *Figure 56*. The A/D status bit ADSF is set automatically when A/D conversion is completed and cleared when A/D conversion is in process.

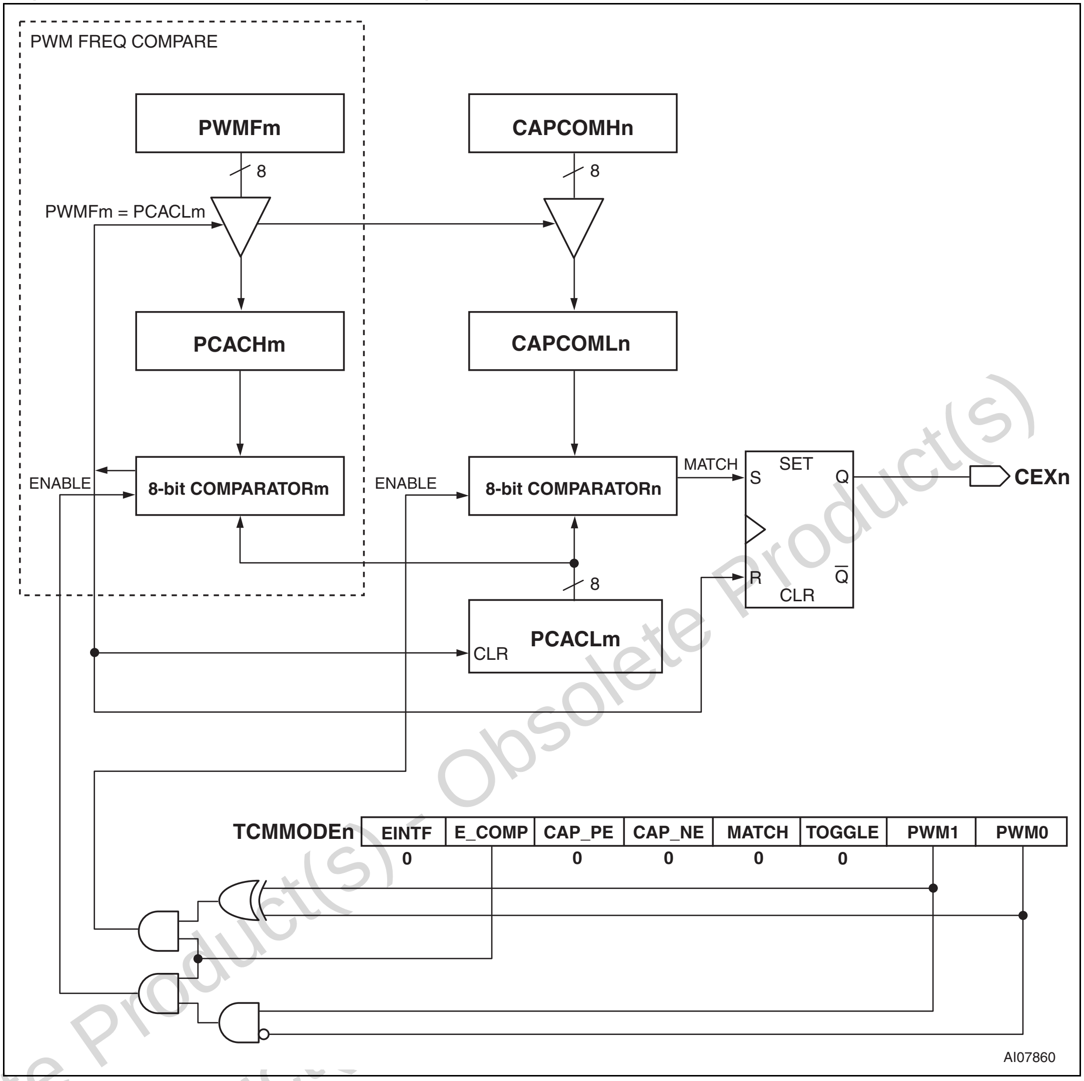
In addition, the ADC unit sets the interrupt flag in the ACON register after a conversion is complete (if AINTEN is set to '1'). The ADC interrupts the CPU when the enable bit AINTEN is set.

26.1 Port 1 ADC channel selects

The P1SFS0 and P1SFS1 registers control the selection of the Port 1 pin functions. When the P1SFS0 Bit is '0,' the pin functions as a GPIO. When bits are set to '1,' the pins are configured as alternate functions. A new P1SFS1 register selects which of the alternate functions is enabled. The ADC channel is enabled when the bit in P1SFS1 is set to '1.'

Note: In the 52-pin package, there is no individual AV_{REF} pin because AV_{REF} is combined with AV_{CC} pin.

Figure 60. PWM mode - (x8) programmable frequency



1. m = 0: n = 0, 1, or 2
   m = 1: n = 3, 4, or 5

27.9 PWM mode - fixed frequency, 16-bit

The operation of the 16-bit PWM is the same as the 8-bit PWM with fixed frequency. In this mode, one or all the TCM can be configured to have a fixed frequency PWM output on the port pins. The PWM frequency is depending on the clock input frequency to the 16-bit Counter. The duty cycle of each TCM module can be specified in the CAPCOMHn and CAPCOMLn registers. When the 16-bit PCA_Counter is equal or greater than the values in registers CAPCOMHn and CAPCOMLn, the PWM output is switched to a high state. When the PCA_Counter overflows, CEXn is asserted low.

from the Secondary Flash memory in program space. After the writing is complete, the Main Flash can be “reclassified” back to program space, then execution can continue from the new code in Main Flash memory. The mapping example of *Figure 67* will accommodate this operation.

28.2.6 Memory sector select rules

When defining sector select signals (FSx, CSBOOTx, RS0, CSIOP, PSELx) in PSDsoft Express, the user must keep these rules in mind:

- Main Flash and Secondary Flash memory sector select signals may not be larger than their physical sector size as defined in *Table 157 on page 193*.
- Any Main Flash memory sector select may not be mapped in the same address range as another Main Flash sector select (cannot overlap segments of Main Flash on top of each other).
- Any Secondary Flash memory sector select may not be mapped in the same address range as another Secondary Flash sector select (cannot overlap segments of Secondary Flash on top of each other).
- A Secondary Flash memory sector may overlap a Main Flash memory sector. In the case of overlap, priority is given to the Secondary Flash memory sector.
- SRAM, CSIOP, or PSELx may overlap any Flash memory sector. In the case of overlap, priority is given to SRAM, CSIOP, or PSELx.

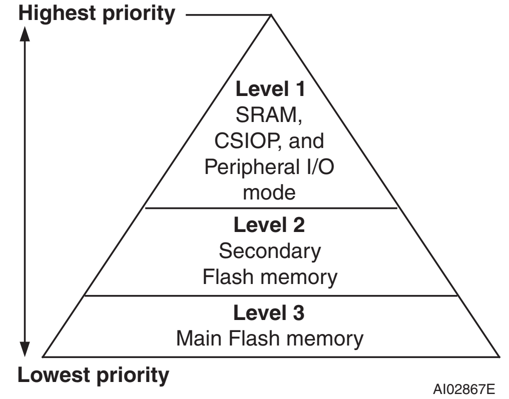
Note: PSELx is for optional Peripheral I/O Mode on Port A.

- The address range for sector selects for SRAM, PSELx, and CSIOP must not overlap each other as they have the same priority, causing contention if overlapped.

Figure 68 illustrates the priority scheme of the memory elements of the PSD module. Priority refers to which memory will ultimately produce a byte of data or code to the 8032 MCU for a given bus cycle. Any memory on a higher level can overlap and has priority over any memory on a lower level. Memories on the same level must not overlap.

Example: FS0 is valid when the 8032 produces an address in the range of 8000h to BFFFh. CSBOOT0 is valid from 8000h to 9FFFh. RS0 is valid from 8000h to 87FFh. Any address from the 8032 in the range of RS0 always accesses the SRAM. Any address in the range of CSBOOT0 greater than 87FFh (and less than 9FFFh) automatically addresses Secondary Flash memory. Any address greater than 9FFFh accesses Main Flash memory. One-half of the Main Flash memory segment and one-fourth of the Secondary Flash memory segment cannot be accessed by the 8032.

Figure 68. PSD module memory priority



28.5.20 Reset Flash

The Reset Flash instruction sequence resets the embedded algorithm running on the state machine in the targeted Flash memory (Main or Secondary) and the memory goes into Read Array mode. The Reset Flash instruction consists of one bus WRITE cycle as shown in *Table 163 on page 209*, and it must be executed after any error condition that has occurred during a Flash memory Program or Erase operation.

It may take the Flash memory up to 25µs to complete the Reset cycle. The Reset Flash instruction sequence is ignored when it is issued during a Program or Bulk Erase operation. The Reset Flash instruction sequence aborts any on-going Sector Erase operation and returns the Flash memory to Read Array mode within 25µs.

28.5.21 Reset signal applied to Flash memory

Whenever the PSD module receives a reset signal from the MCU module, any operation that is occurring in either Flash memory array will be aborted and the array(s) will go to Read Array mode. It may take up to 25µs to abort an operation and achieve Read Array mode.

A reset from the MCU module will result from any of these events: an active signal on the UPSD34xx $\overline{\text{RESET\_IN}}$ input pin, a watchdog timer time-out, detection of low V_{CC}, or a JTAG debug channel reset event.

28.5.22 Flash memory sector protection

Each Flash memory sector can be separately protected against program and erase operations. This mode can be activated (or deactivated) by selecting this feature in PSDsoft Express and then programming through the JTAG Port. Sector protection can be selected for individual sectors, and the 8032 cannot override the protection during run-time. The 8032 can read, but not change, sector protection.

Any attempt to program or erase a protected Flash memory sector is ignored. The 8032 may read the contents of a Flash sector even when a sector is protected.

Sector protection status is not read using Flash memory instruction sequences, but instead this status is read by the 8032 reading two registers within csiop address space shown in Table *165* and Table *166*.

28.5.23 Flash memory protection during power-up

Flash memory WRITE operations are automatically prevented while V_{DD} is ramping up until it rises above V_{LKO} voltage threshold at which time Flash memory WRITE operations are allowed.

28.5.24 PSD module security bit

A programmable security bit in the PSD module protects its contents from unauthorized viewing and copying. The security bit is set using PSDsoft Express and programmed into the PSD module with JTAG. When set, the security bit will block access of JTAG programming equipment from reading or modifying the PSD module Flash memory and PLD configuration. The security bit also blocks JTAG access to the MCU module for debugging. The only way to defeat the security bit is to erase the entire PSD module using JTAG (erase is the only JTAG operation allowed while security bit is set), after which the device is blank and may be used again. The 8032 MCU will always have access to Flash

28.5.29 Output macrocell

The GPLD has 16 OMCs. Architecture of one individual OMC is shown in *Figure 76*. OMCs can be used for internal node feedback (buried registers to build shift registers, etc.), or their outputs may be routed to external port pins. The user can choose any mixture of OMCs used for buried functions and OMCs used to drive port pins.

Referring to *Figure 76*, for each OMC there are native product terms available from the AND-OR Array to form logic, and also borrowed product terms are available (if unused) from other OMCs. The polarity of the final product term output is controlled by the XOR gate. Each OMC can implement sequential logic using the flip-flop element, or combinatorial logic when bypassing the flip-flop as selected by the output multiplexer. An OMC output can drive a port pin through the OMC Allocator, it can also drive the 8032 data bus, and also it can drive a feedback path to the AND-OR Array inputs, all at the same time.

The flip-flop in each OMC can be synthesized as a D, T, JK, or SR type in PSDsoft Express. OMC flip-flops are specified using PSDsoft Express in the “User Defined Nodes” section of the Design Assistant. Each flip-flop’s clock, preset, and clear inputs may be driven individually from a product term of the AND-OR Array, defined by equations in PSDsoft Express for signals *. c, *.pr, and *.re respectively. The preset and clear inputs on the flip-flops are level activated, active-high logic signals. The clock inputs on the flip-flops are rising-edge logic signals.

Optionally, the signal CLKIN (pin PD1) can be used for a common clock source to all OMC flip-flops. Each flip-flop is clocked on the rising edge. A common clock is specified in PSDsoft Express by assigning the function “Common Clock Input” for pin PD1 in the Pin Definition section, and then choosing the signal CLKIN when specifying the clock input (*.c) for individual flip-flops in the “User Defined Nodes” section.

Table 168. OMC port and data bit assignments (continued)

OMC	Port assignment[1],[2]	Native product terms from AND-OR array	Maximum borrowed product terms	Data bit on 8032 data bus for loading or reading OMC
MCELLBC3	Port B3 or C3	4	5	D3
MCELLBC4	Port B4 or C4	4	6	D4
MCELLBC5	Port B5	4	6	D5
MCELLBC6	Port B6	4	6	D6
MCELLBC7	Port B7 orC7	4	6	D7

1. MCELLAB0-MCELLAB7 can be output to Port A pins only on 80-pin devices. Port A is not available on 52-pin devices.
2. Port pins PC0, PC1, PC5, and PC6 are dedicated JTAG pins and are not available as outputs for MCELLBC 0, 1, 5, or 6.

28.5.32 Loading and reading OMCs

Each of the two OMC groups (eight OMCs each) occupies a byte in csiop space, named MCELLAB and MCELLBC (see *Table 169* and *Table 170*). When the 8032 writes or reads these two OMC registers in csiop it is accessing each of the OMCs through its 8-bit data bus, with the bit assignment shown in *Table 168 on page 227*. Sometimes it is important to know the bit assignment when the user builds GPLD logic that is accessed by the 8032. For example, the user may create a 4-bit counter that must be loaded and read by the 8032, so the user must know which nibble in the corresponding csiop OMC register the firmware must access. The fitter report generated by PSDsoft Express will indicate how it assigned the OMCs and data bus bits to the logic. The user can optionally force PSDsoft Express to assign logic to specific OMCs and data bus bits if desired by using the ‘PROPERTY’ statement in PSDsoft Express. Please see the PSDsoft Express User’s Manual for more information on OMC assignments.

Loading the OMC flip-flops with data from the 8032 takes priority over the PLD logic functions. As such, the preset, clear, and clock inputs to the flip-flop can be asynchronously overridden when the 8032 writes to the csiop registers to load the individual OMCs.

Table 169. Output macrocell MCELLAB (address = csiop + offset 20h)[1]

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

1. All bits clear to logic ’0’ at power-on reset, but do not clear after warm reset conditions (non-power-on reset).

Table 170. Output macrocell MCELLBC (address = csiop + offset 21h)[1]

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

1. All bits clear to logic ’0’ at power-on reset, but do not clear after warm reset conditions (non-power-on reset).

28.5.50 Port C structure

Port C supports the following operating modes on pins PC2, PC3, PC4, PC7:

- MCU I/O Mode
- GPLD Output Mode from Output Macrocells MCELLBC2, MCELLBC3, MCELLBC4, MCELLBC7
- GPLD Input Mode to Input Macrocells IMCC2, IMCC3, IMCC4, IMCC7

See *Figure 86 on page 246* for detail.

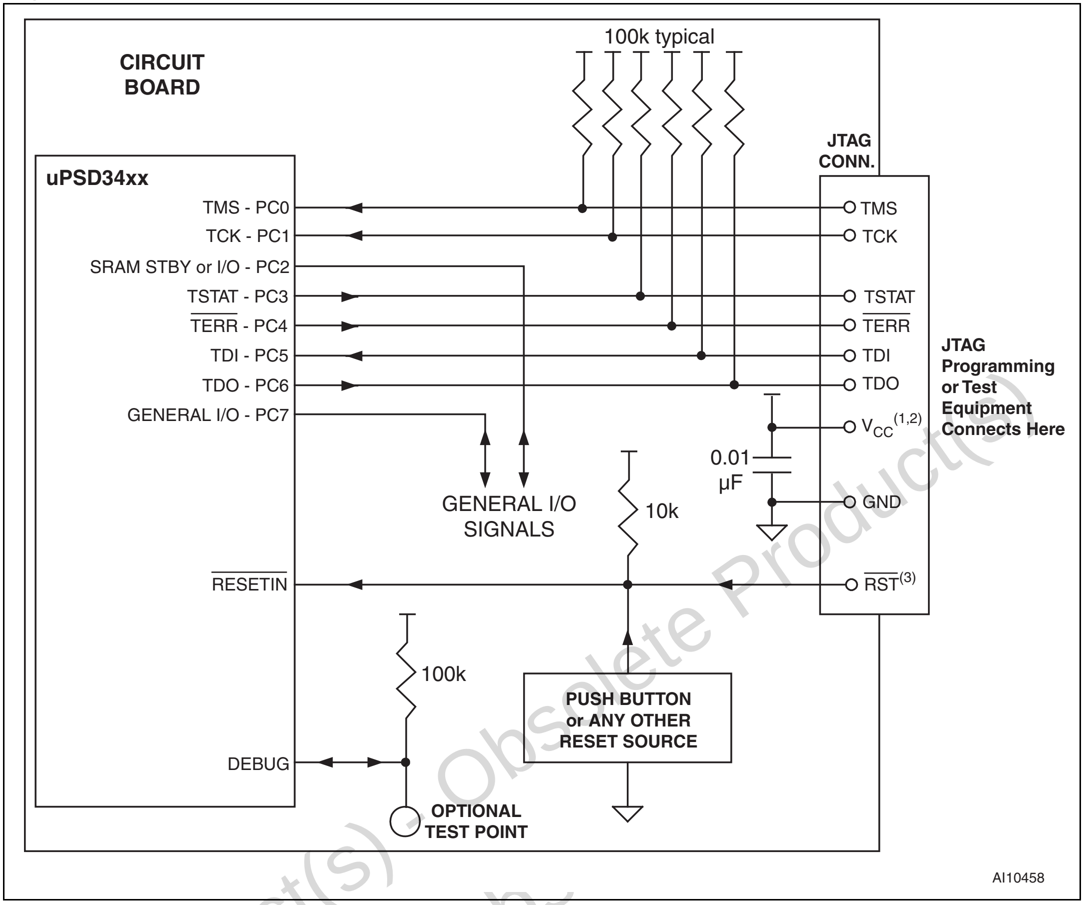
Port C pins can also be configured in PSDsoft for other dedicated functions:

- Pins PC3 and PC4 support TSTAT and $\overline{\text{TERR}}$ status indicators, to reduce the amount of time required for JTAG ISP programming. These two pins must be used together for this function, adding to the four standard JTAG signals. When TSTAT and $\overline{\text{TERR}}$ are used, it is referred to as “6-pin JTAG”. PC3 and PC4 cannot be used for other functions if they are used for 6-pin JTAG. See *Section 28.6.1: JTAG ISP and JTAG debug on page 257* for details.
- PC3 can be used as an output to indicate when a Flash memory program or erase operation has completed. This is specified in PSDsoft Express as *Section 28.5.13: Ready/Busy (PC3) on page 215*.

The remaining four pins (TDI, TDO, TCK, TMS) on Port C are dedicated to the JTAG function and cannot be used for any other function. See *Section 28.6.1: JTAG ISP and JTAG debug on page 257*.

Port C also supports the Open Drain output drive type options on pins PC2, PC3, PC4, and PC7 using the csiop Drive Select registers.

Figure 92. Recommended 6-pin JTAG connections



1. For 5 V UPSD34xx devices, pull-up resistors and V_{CC} pin on the JTAG connector should be connected to 5 V system V_{DD}.
2. For 3.3 V UPSD34xx devices, pull-up resistors and V_{CC} pin on the JTAG connector should be connected to 3.3 V system V_{CC}.
3. This signal is driven by an Open-Drain output in the JTAG equipment, allowing more than one source to activate RESET_IN.

28.6.6 Recommended JTAG connector

There is no industry standard JTAG connector. STMicroelectronics recommends a specific JTAG connector and pinout for uPSD3xxx so programming and debug equipment will easily connect to the circuit board. The user does not have to use this connector if there is a different connection scheme.

The recommended connector scheme can accept a standard 14-pin ribbon cable connector (2 rows of 7 pins on 0.1” centers, 0.025” square posts, standard keying) as shown in *Figure 93*. See the STMicroelectronics “FlashLINK, FL-101 User Manual” for more information.

Figure 101. Synchronous Clock mode timing – PLD

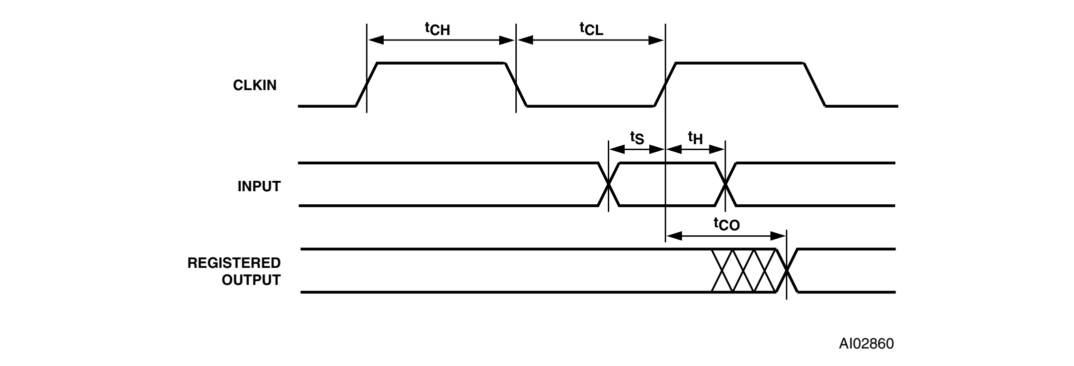


Table 222. CPLD macrocell synchronous clock mode timing (5 V PSD module)

Symbol	Parameter	Conditions	Min	Max	PT Aloc	Turbo Off	Slew rate[(1)]	Unit
f_{MAX}	Maximum frequency external feedback	$1/(t_S+t_{CO})$		40.0				MHz
	Maximum frequency internal feedback (f_{CNT})	$1/(t_S+t_{CO}-10)$		66.6				MHz
	Maximum frequency pipelined data	$1/(t_{CH}+t_{CL})$		83.3				MHz
t_S	Input setup time		12		+ 2	+ 10		ns
t_H	Input hold time		0					ns
t_{CH}	Clock high time	Clock input	6					ns
t_{CL}	Clock low time	Clock input	6					ns
t_{CO}	Clock to output delay	Clock input		13			– 2	ns
t_{ARD}	CPLD array delay	Any macrocell		11	+ 2			ns
t_{MIN}	Minimum clock period[(2)]	$t_{CH}+t_{CL}$	12					ns

1. Fast slew rate output available on PA3-PA0, PB3-PB0, and PD2-PD1. Decrement times by given amount.
2. CLKIN (PD1) $t_{CLCL} = t_{CH} + t_{CL}$.

33 Part numbering

Table 239. Ordering information scheme

Example: UPSD 34 3 4 E V – 40 U 6 T

Device type

uPSD = Microcontroller PSD

Family

34 = Turbo Plus core

SRAM size

2 = 4 Kbyte

3 = 8 Kbyte

5 = 32 Kbyte

Main Flash memory size

2 = 64 Kbyte

3 = 128 Kbyte

4 = 256 Kbyte

IP Mix

E = IP Mix: USB, I^{2}C, SPI, UART (2), IrDA, ADC, Supervisor, PCA

Operating voltage

blank: V_{CC} = 3.0 to 3.6 V, V_{DD} = 4.5 to 5.5 V

V: V_{CC} = V_{DD} = 3.0 to 3.6 V

Revision

"-" = Revision A

B = Revision B

Speed

–40 = 40 MHz

Package

T = 52-pin LQFP ECOPACK-compliant package

U = 80-pin LQFP ECOPACK-compliant package

Temperature range

6 = –40 to 85 °C

Shipping option

Tape & reel packing = T

For other options, or for more information on any aspect of this device, please contact the ST sales office nearest you.