Welcome to E-XFL.COM

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
Peripherals	LVD, POR, PWM, WDT
Number of I/O	45
Program Memory Size	288KB (288K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
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/upsd3354dv-40u6

4.1 Internal memory (MCU, standard 8032 memory: DATA, IDATA, SFR)

4.1.1 DATA memory

The first 128 bytes of internal SRAM ranging from address 0x0000 to 0x007F are called DATA, which can be accessed using 8032 **direct or indirect** addressing schemes and are typically used to store variables and stack.

Four register banks, each with 8 registers (R0 – R7), occupy addresses 0x0000 to 0x001F. Only one of these four banks may be enabled at a time. The next 16 locations at 0x0020 to 0x002F contain 128 directly addressable bit locations that can be used as software flags. SRAM locations 0x0030 and above may be used for variables and stack.

4.1.2 IDATA memory

The next 128 bytes of internal SRAM are named IDATA and range from address 0x0080 to 0x00FF. IDATA can be accessed only through 8032 **indirect addressing** and is typically used to hold the MCU stack as well as data variables. The stack can reside in both DATA and IDATA memories and reach a size limited only by the available space in the combined 256 bytes of these two memories (since stack accesses are always done using indirect addressing, the boundary between DATA and IDATA does not exist with regard to the stack).

4.1.3 SFR memory

Special function registers (*Table 5 on page 41*) occupy a separate physical memory, but they logically overlap the same 128 bytes as IDATA, ranging from address 0x0080 to 0x00FF. SFRs are accessed only using **direct addressing**. There 86 active registers used for many functions: changing the operating mode of the 8032 MCU core, controlling 8032 peripherals, controlling I/O, and managing interrupt functions. The remaining unused SFRs are reserved and should not be accessed.

16 of the SFRs are both byte- and bit-addressable. Bit-addressable SFRs are those whose address ends in “0” or “8” hex.

4.2 External memory (PSD module: program memory, data memory)

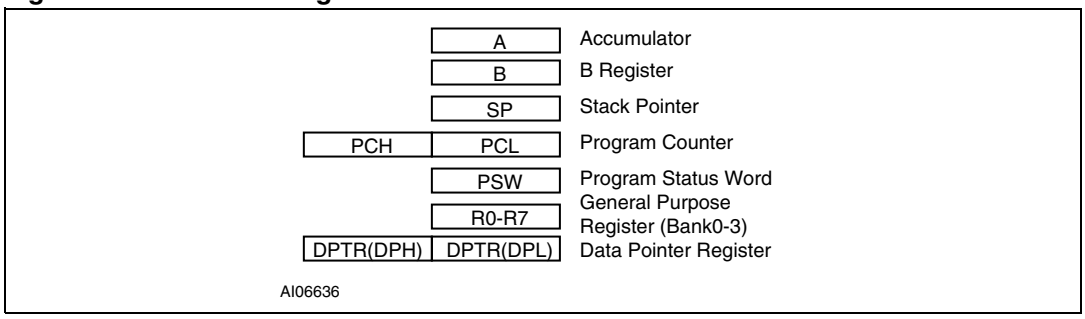
The PSD module has four memories: main Flash, secondary Flash, SRAM, and CSIOP. *Section 27: PSD module on page 164* for more detailed information on these memories.

Memory mapping in the PSD module is implemented with the Decode PLD (DPLD) and optionally the Page register. The user specifies decode equations for individual segments of each of the memories using the software tool PSDsoft Express. This is a very easy point-and-click process allowing total flexibility in mapping memories. Additionally, each of the memories may be placed in various combinations of 8032 program address space or 8032 data address space by using the software tool PSDsoft Express.

7 8032 MCU registers

The UPSD33xx has the following 8032 MCU core registers, also shown in *Figure 10*.

Figure 10. 8032 MCU registers



7.1 Stack Pointer (SP)

The SP is an 8-bit register which holds the current location of the top of the stack. It is incremented before a value is pushed onto the stack, and decremented after a value is popped off the stack. The SP is initialized to 07h after reset. This causes the stack to begin at location 08h (top of stack). To avoid overlapping conflicts, the user must initialize the top of the stack to 20h if all four banks of registers R0 - R7 are used, and the user must initialize the top of stack to 30h if all of the 8032 bit memory locations are used.

7.2 Data Pointer (DPTR)

DPTR is a 16-bit register consisting of two 8-bit registers, DPL and DPH. The DPTR register is used as a base register to create an address for indirect jumps, table look-up operations, and for external data transfers (XDATA). When not used for addressing, the DPTR register can be used as a general purpose 16-bit data register.

Very frequently, the DPTR register is used to access XDATA using the External Direct addressing mode. The UPSD33xx has a special set of SFR registers (DPTC, DPTM) to control a secondary DPTR register to speed memory-to-memory XDATA transfers. Having dual DPTR registers allows rapid switching between source and destination addresses (see details in *Section 11: Dual data pointers on page 56*).

7.3 Program Counter (PC)

The PC is a 16-bit register consisting of two 8-bit registers, PCL and PCH. This counter indicates the address of the next instruction in program memory to be fetched and executed. A reset forces the PC to location 0000h, which is where the reset jump vector is stored.

Table 5. SFR memory map with direct address and reset value (continued)

SFR addr (hex)	SFR name	Bit name and <Bit Address>								Reset value (hex)	Reg. descr. with link
		7	6	5	4	3	2	1	0		
D4	SPITDR	SPITDR[7:0]								00	Table 91 on page 149
D5	SPIRDR	SPIRDR[7:0]								00	
D6	SPICON0	–	TE	RE	SPIEN	SSEL	FLSB	SPO	–	00	Table 85 on page 147
D7	SPICON1	–	–	–	–	TEIE	RORIE	TIE	RIE	00	Table 87 on page 148
D8(1)	SCON1	SM0 <DF	SM1 <DE>	SM2 <DD>	REN <DC>	TB8 <DB>	RB8 <DA>	TI <D9>	RI <D8>	00	Table 65 on page 109
D9	SBUF1	SBUF1[7:0]								00	Figure 24 on page 104
DA	RESERVED										
DB	S1SETUP	SS_EN	SMPL_SET[6:0]							00	Table 80 on page 132
DC	S1CON	CR2	EN1	STA	STO	ADDR	AA	CR1	CR0	00	Table 71 on page 128
DD	S1STA	GC	STOP	INTR	TX_MD	B_BUSY	B_LOST	$\overline{\text{ACK\_R}}$	SLV	00	Table 74 on page 130
DE	S1DAT	S1DAT[7:0]								00	Table 76 on page 131
DF	S1ADR	S1ADR[7:0]								00	Table 78 on page 131
E0(1)	A	A[7:0] <bit addresses: E7h, E6h, E5h, E4h, E3h, E2h, E1h, E0h>								00	Section 7.4 on page 38
E1 to EF	RESERVED										
F0(1)	B	B[7:0] <bit addresses: F7h, F6h, F5h, F4h, F3h, F2h, F1h, F0h>								00	Section 7.5 on page 38
F1	RESERVED										
F2	RESERVED										
F3	RESERVED										
F4	RESERVED										
F5	RESERVED										
F6	RESERVED										
F7	RESERVED										
F8	RESERVED										
F9	CCON0	–	–	–	DBGCE	CPU_AR	CPUPS[2:0]			10	Table 27 on page 69
FA	RESERVED										
FB	CCON2	–	–	–	PCA0CE	PCA0PS[3:0]				10	Table 99 on page 154
FC	CCON3	–	–	–	PCA1CE	PCA1PS[3:0]				10	Table 101 on page 155
FD	RESERVED										

9 8032 addressing modes

The 8032 MCU uses 11 different addressing modes listed below:

- Register
- Direct
- Register Indirect
- Immediate
- External Direct
- External Indirect
- Indexed
- Relative
- Absolute
- Long
- Bit

9.1 Register addressing

This mode uses the contents of one of the registers R0 - R7 (selected by the last three bits in the instruction opcode) as the operand source or destination. This mode is very efficient since an additional instruction byte is not needed to identify the operand. For example:

```
MOV A, R7                    ; Move contents of R7 to accumulator
```

9.2 Direct addressing

This mode uses an 8-bit address, which is contained in the second byte of the instruction, to directly address an operand which resides in either 8032 DATA SRAM (internal address range 00h-07Fh) or resides in 8032 SFR (internal address range 80h-FFh). This mode is quite fast since the range limit is 256 bytes of internal 8032 SRAM. For example:

```
MOV A, 40h                   ; Move contents of DATA SRAM
                             ; at location 40h into the accumulator
```

9.3 Register indirect addressing

This mode uses an 8-bit address contained in either register R0 or R1 to indirectly address an operand which resides in 8032 IDATA SRAM (internal address range 80h-FFh). Although 8032 SFR registers also occupy the same physical address range as IDATA, SFRs will not be accessed by register Indirect mode. SFRs may only be accesses using Direct address mode. For example:

Table 7. Logical instruction set (continued)

Mnemonic[1] and use		Description	Length/cycles
XRL	A, @Ri	Exclusive-OR indirect SRAM to ACC	1 byte/1 cycle
XRL	A, #data	Exclusive-OR immediate data to ACC	2 byte/1 cycle
XRL	direct, A	Exclusive-OR ACC to direct byte	2 byte/1 cycle
XRL	direct, #data	Exclusive-OR immediate data to direct byte	3 byte/2 cycle
CLR	A	Clear ACC	1 byte/1 cycle
CPL	A	Compliment ACC	1 byte/1 cycle
RL	A	Rotate ACC left	1 byte/1 cycle
RLC	A	Rotate ACC left through the carry	1 byte/1 cycle
RR	A	Rotate ACC right	1 byte/1 cycle
RRC	A	Rotate ACC right through the carry	1 byte/1 cycle

1. All mnemonics copyrighted ©Intel Corporation 1980.

Table 8. Data transfer instruction set

Mnemonic[1] and use		Description	Length/cycles
MOV	A, Rn	Move register to ACC	1 byte/1 cycle
MOV	A, direct	Move direct byte to ACC	2 byte/1 cycle
MOV	A, @Ri	Move indirect SRAM to ACC	1 byte/1 cycle
MOV	A, #data	Move immediate data to ACC	2 byte/1 cycle
MOV	Rn, A	Move ACC to register	1 byte/1 cycle
MOV	Rn, direct	Move direct byte to register	2 byte/2 cycle
MOV	Rn, #data	Move immediate data to register	2 byte/1 cycle
MOV	direct, A	Move ACC to direct byte	2 byte/1 cycle
MOV	direct, Rn	Move register to direct byte	2 byte/2 cycle
MOV	direct, direct	Move direct byte to direct	3 byte/2 cycle
MOV	direct, @Ri	Move indirect SRAM to direct byte	2 byte/2 cycle
MOV	direct, #data	Move immediate data to direct byte	3 byte/2 cycle
MOV	@Ri, A	Move ACC to indirect SRAM	1 byte/1 cycle

16 Oscillator and external components

The oscillator circuit of UPSD33xx devices is a single stage, inverting amplifier in a Pierce oscillator configuration. The internal circuitry between pins XTAL1 and XTAL2 is basically an inverter biased to the transfer point. Either an external quartz crystal or ceramic resonator can be used as the feedback element to complete the oscillator circuit. Both are operated in parallel resonance. Ceramic resonators are lower cost, but typically have a wider frequency tolerance than quartz crystals. Alternatively, an external clock source from an oscillator or other active device may drive the UPSD33xx oscillator circuit input directly, instead of using a crystal or resonator.

The minimum frequency of the quartz crystal, ceramic resonator, or external clock source is 1 MHz if the I^2C interface is not used. The minimum is 8 MHz if I^2C is used. The maximum is 40 MHz in all cases. This frequency is f_{OSC}, which can be divided internally as described in *Section 14: MCU clock generation on page 68*.

The pin XTAL1 is the high gain amplifier input, and XTAL2 is the output. To drive the UPSD33xx device externally from an oscillator or other active device, XTAL1 is driven and XTAL2 is left open-circuit. This external source should drive a logic low at the voltage level of 0.3 V_{CC} or below, and logic high at 0.7 V_{CC} or above, up to 5.5 V_{CC}. The XTAL1 input is 5 V tolerant.

Most of the quartz crystals in the range of 25 MHz to 40 MHz operate in the third overtone frequency mode. An external LC tank circuit at the XTAL2 output of the oscillator circuit is needed to achieve the third overtone frequency, as shown in *Figure 14 on page 75*. Without this LC circuit, the crystal will oscillate at a fundamental frequency mode that is about 1/3 of the desired overtone frequency.

Note: In Figure 14 on page 75 crystals which are specified to operate in fundamental mode (not overtone mode) do not need the LC circuit components. Since quartz crystals and ceramic resonators have their own characteristics based on their manufacturer, it is wise to also consult the manufacturer's recommended values for external components.

Table 73. Selection of the SCL frequency in Master mode based on f_{OSC} examples

CR2	CR1	CR0	f_{OSC} divided by:	Bit rate (kHz) @ f_{OSC}			
				12 MHz f_{OSC}	24 MHz f_{OSC}	36 MHz f_{OSC}	40 MHz f_{OSC}
0	0	0	32	375	750	X(1)	X(1)
0	0	1	48	250	500	750	833
0	1	0	60	200	400	600	666
0	1	1	120	100	200	300	333
1	0	0	240	50	100	150	166
1	0	1	480	25	50	75	83
1	1	0	960	12.5	25	37.5	41
1	1	1	1920	6.25	12.5	18.75	20

1. These values are beyond the bit rate supported by UPSD33xx.

23.9 I²C Interface Status register (S1STA)

The S1STA register provides status regarding immediate activity and the current state of operation on the I²C bus. All bits in this register are read-only except bit 5, INTR, which is the interrupt flag.

23.9.1 Interrupt conditions

If the I²C interrupt is enabled (EI²C = 1 in SFR named IEA, and EA =1 in SFR named IE), and the SIOE is initialized, then an interrupt is automatically generated when any one of the following five events occur:

- When the SIOE receives an address that matches the contents of the SFR, S1ADR. Requirements: SIOE is in Slave mode, and bit AA = 1 in the SFR S1CON.
- When the SIOE receives General Call address. Requirements: SIOE is in Slave mode, bit AA = 1 in the SFR S1CON
- When a complete data byte has been received or transmitted by the SIOE while in Master mode. The interrupt will occur even if the Master looses arbitration.
- When a complete data byte has been received or transmitted by the SIOE while in selected Slave mode.
- A STOP condition on the bus has been recognized by the SIOE while in selected Slave mode.

Selected Slave mode means the device address sent by the Master device at the beginning of the current data transfer matched the address stored in the S1ADR register.

If the I²C interrupt is not enabled, the MCU may poll the INTR flag in S1STA.

configuration (which means the user does not have to use the $\overline{\text{SPISEL}}$ signal from UPSD33xx in this case).

The SPI specification does not specify high-level protocol for data exchange, only low-level bit-serial transfers are defined.

24.2 Full-duplex operation

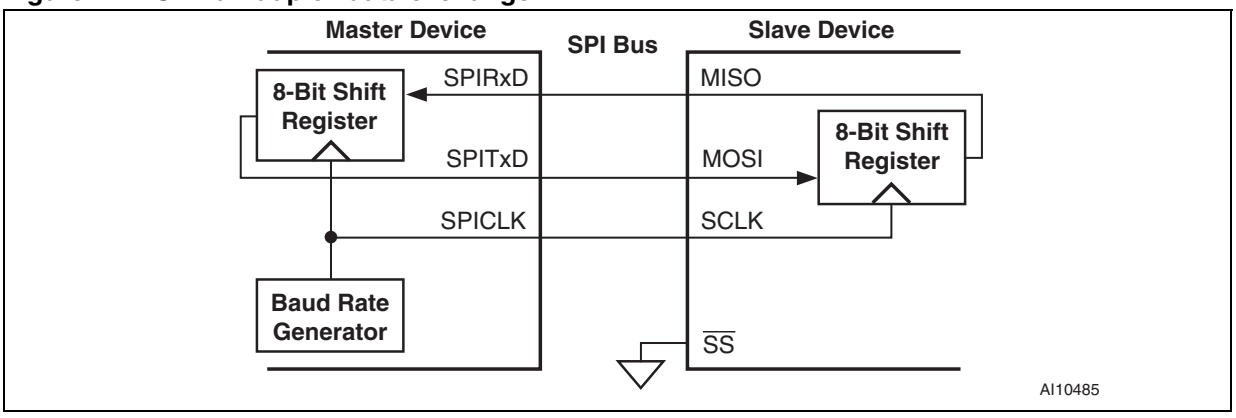
When an SPI transfer occurs, 8 bits of data are shifted out on one pin while a different 8 bits of data are simultaneously shifted in on a second pin. Another way to view this transfer is that an 8-bit shift register in the Master and another 8-bit shift register in the Slave are connected as a circular 16-bit shift register. When a transfer occurs, this distributed shift register is shifted 8 bit positions; thus, the data in the Master and Slave devices are effectively exchanged (see *Figure 41 on page 143*).

24.3 Bus-level activity

Figure 42 on page 144 details an SPI receive operation (with respect to bus Master) and *Figure 43 on page 144* details an SPI transmit operation. Also shown are internal flags available to firmware to manage data flow. These flags are accessed through a number of SFRs.

Note: The UPSD33xx SPI interface SFRs allow the choice of transmitting the most significant bit (MSB) of a byte first, or the least significant bit (LSB) first. The same bit-order applies to data reception. Figure 42 and Figure 43 illustrate shifting the LSB first.

Figure 41. SPI full-duplex data exchange



26 Programmable counter array (PCA) with PWM

There are two programmable counter array blocks (PCA0 and PCA1) in the UPSD33xx. A PCA block consists of a 16-bit up-counter, which is shared by three TCM (Timer Counter module). A TCM can be programmed to perform one of the following four functions:

1. Capture mode: capture counter values by external input signals
2. Timer mode
3. Toggle Output mode
4. PWM mode: fixed frequency (8-bit or 16-bit), programmable frequency (8-bit only)

26.1 PCA block

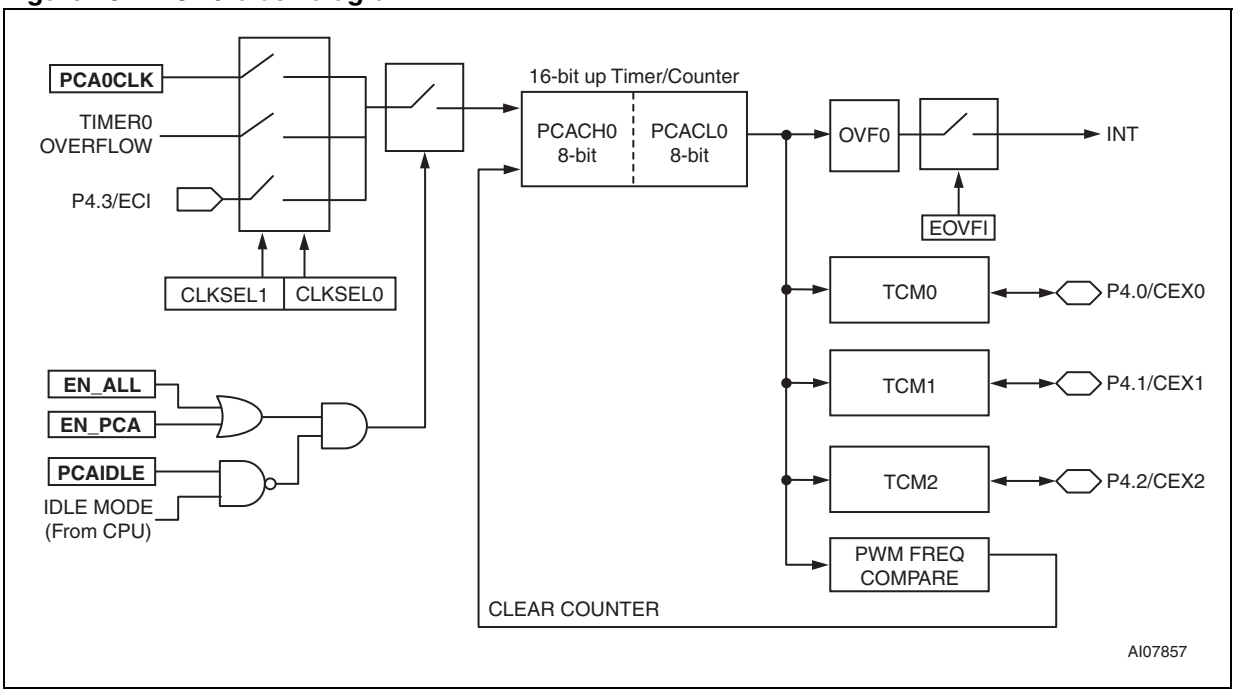
The 16-bit Up-Counter in the PCA block is a free-running counter (except in PWM mode with programmable frequency). The Counter has a choice of clock input: from an external pin, Timer 0 Overflow, or PCA Clock.

A PCA block has 3 Timer Counter modules (TCM) which share the 16-bit Counter output. The TCM can be configured to capture or compare counter value, generate a toggling output, or PWM functions. Except for the PWM function, the other TCM functions can generate an interrupt when an event occurs.

Every TCM is connected to a port pin in Port 4; the TCM pin can be configured as an event input, a PWMs, a Toggle Output, or as External Clock Input. The pins are general I/O pins when not assigned to the TCM.

The TCM operation is configured by Control registers and Capture/Compare registers. *Table 98 on page 154* lists the SFR registers in the PCA blocks.

Figure 46. PCA0 block diagram



27.1.14 I/O ports

For 80-pin UPSD33xx devices, the PSD module has 22 individually configurable I/O pins distributed over four ports (these I/O are in addition to I/O on MCU module). For 52-pin UPSD33xx devices, the PSD module has 13 individually configurable I/O pins distributed over three ports. See *Figure 73 on page 219* for I/O port pin availability on these two packages.

I/O port pins on the PSD module (Ports A, B, C, and D) are completely separate from the port pins on the MCU module (Ports 1, 3, and 4). They even have different electrical characteristics. I/O port pins on the PSD module are accessed by csiop registers, or they are controlled by PLD equations. Conversely, I/O Port pins on the MCU module are controlled by the 8032 SFR registers.

Table 113. General I/O pins on PSD module

Pkg	Port A	Port B	Port D	Port D	Total
52-pin	0	8	4	1	13
80-pin	8	8	4	2	22

Note: Four pins on Port C are dedicated to JTAG, leaving four pins for general I/O.

Each I/O pin on the PSD module can be individually configured for different functions on a pin-by-pin basis (*Figure 68 on page 208*). Following are the available functions on PSD module I/O pins.

- **MCU I/O:** 8032 controls the output state of each port pin or it reads input state of each port pin, by accessing csiop registers at run-time. The direction (in or out) of each pin is also controlled by csiop registers at run-time.
- **PLD I/O:** PSDsoft Express logic equations and pin configuration selections determine if pins are connected to OMC outputs or IMC inputs. This is a static and non-volatile configuration. Port pins connected to PLD outputs can no longer be driven by the 8032 using MCU I/O output mode.
- **Latched MCU Address Output:** Port A or Port B can output de-multiplexed 8032 address signals A0 - A7 on a pin-by-pin basis as specified in csiop registers at run-time. In addition, Port B can also be configured to output de-multiplexed A8-A15 in PSDsoft Express.
- **Data Bus Repeater:** Port A can bi-directionally buffer the 8032 data bus (de-multiplexed) for a specified address range in PSDsoft Express. This is referred to as ***Peripheral I/O mode*** in this document.
- **Open Drain Outputs:** Some port pins can function as open-drain as specified in csiop registers at run-time.
- Pins on Port D can be used for **external chip-select** outputs originating from the DPLD, without consuming OMC resources within the GPLD.

on the PSD module as an input to both PLDs (without routing a signal externally on PC board) and it's signal name is "rd_bsy". The Ready/Busy output can be probed during lab development to check the timing of Flash memory programming in the system at run-time.

27.4.13 Bypassed Unlock sequence

The Bypass Unlock mode allows the 8032 to program bytes in the Flash memories faster than using the standard Flash program instruction sequences because the typical AAh, 55h unlock bus cycles are bypassed for each byte that is programmed. Bypassing the unlock sequence is typically used when the 8032 is intentionally programming a large number of bytes (such as during IAP). After intentional programming is complete, typically the Bypass mode would be disabled, and full protection is back in place to prevent unwanted WRITEs to Flash memory.

The Bypass Unlock mode is entered by first initiating two Unlock bus cycles. This is followed by a third WRITE operation containing the Bypass Unlock command, 20h (as shown in *Table 117 on page 184*). The Flash memory array that received that sequence then enters the Bypass Unlock mode. After this, a two bus cycle program operation is all that is required to program a byte in this mode. The first bus cycle in this shortened program instruction sequence contains the Bypassed Unlocked Program command, A0h, to any valid address within the unlocked Flash array. The second bus cycle contains the address and data of the byte to be programmed. Programming status is checked using toggle, polling, or Ready/Busy just as before. Additional data bytes are programmed the same way until this Bypass Unlock mode is exited.

To exit Bypass Unlock mode, the system must issue the Reset Bypass Unlock instruction sequence. The first bus cycle of this instruction must write 90h to any valid address within the unlocked Flash Array; the second bus cycle must write 00h to any valid address within the unlocked Flash Array. After this sequence the Flash returns to Read Array mode.

During Bypass Unlock mode, only the Bypassed Unlock Program instruction, or the Reset Bypass Unlock instruction is valid, other instruction will be ignored.

27.4.14 Erasing Flash memory

Flash memory may be erased sector-by-sector, or an entire Flash memory array may be erased with one command (bulk).

27.4.15 Flash bulk Erase

The Flash Bulk Erase instruction sequence uses six WRITE operations followed by a READ operation of the status register, as described in *Table 117 on page 184*. If any byte of the Bulk Erase instruction sequence is wrong, the Bulk Erase instruction sequence aborts and the device is reset to the Read Array mode. The address provided by the 8032 during the Flash Bulk Erase command sequence may select any one of the eight Flash memory sector select signals FSx or one of the four signals CSBOOTx. An erase of the entire Flash memory array will occur in a particular array even though a command was sent to just one of the individual Flash memory sectors within that array.

During a Bulk Erase, the memory status may be checked by reading the Error Flag Bit (DQ5), the Toggle Flag Bit (DQ6), and the Data Polling Flag Bit (DQ7). The Error Flag Bit (DQ5) returns a '1' if there has been an erase failure. Details of acquiring the status of the Bulk Erase operation are detailed in *Section 27.4.9: Programming Flash memory on page 186*.

27.4.18 Resume sector erase

If a Suspend Sector Erase instruction sequence was previously executed, the erase cycle may be resumed with this instruction sequence. The Resume Sector Erase instruction sequence consists of writing the command 30h to any valid address within the Flash array that was suspended as shown in *Table 117 on page 184*.

27.4.19 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 117 on page 184*, 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.

27.4.20 Reset signal applied to Flash memory

Whenever the PSD module receives a reset signal from the MCU module, any operation 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 UPSD33xx $\overline{\text{RESET\_IN}}$ input pin, a watchdog timer timeout, detection of low V_{CC}, or a JTAG debug channel reset event.

27.4.21 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 119* and *Table 120 on page 193*.

27.4.22 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.

27.4.23 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

Table 132. MCU I/O Mode Port A Data In register (address = csiop + offset 00h)(1)(2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

1. Port A not available on 52-pin UPSD33xx devices
2. For each bit, 1 = current state of input pin is logic '1,' 0 = current state is logic ’0’

Table 133. MCU I/O Mode Port B Data In register (address = csiop + offset 01h)(1)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0

1. For each bit, 1 = current state of input pin is logic '1,' 0 = current state is logic ’0’

Table 134. MCU I/O Mode Port C Data In register (address = csiop + offset 10h)(1)(2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC7	X	X	PC4	PC3	PC2	X	X

1. X = Not guaranteed value, can be read either '1' or '0.'
2. For each bit, 1 = current state of input pin is logic '1,' 0 = current state is logic ’0’

Table 135. MCU I/O Mode Port D Data Inregister (address = csiop + offset 11h)(1)(2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	X	X	X	X	PD2(3)	PD1	X

1. X = Not guaranteed value, can be read either '1' or '0.'
2. For each bit, 1 = current state of input pin is logic '1,' 0 = current state is logic ’0’
3. Not available on 52-pin UPSD33xx devices

Table 136. MCU I/O Mode Port A Data Out register (address =csiop+offset 04h)(1)(2)(3)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

1. Port A not available on 52-pin UPSD33xx devices
2. For each bit, 1 = drive port pin to logic '1,' 0 = drive port pin to logic ’0’
3. Default state of register is 00h after reset or power-up

Table 137. MCU I/O Mode Port B Data Out register (address = csiop + offset 05h)(1)(2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0

1. For each bit, 1 = drive port pin to logic '1,' 0 = drive port pin to logic ’0’
2. Default state of register is 00h after reset or power-up

27.4.54 Chip Select Input ($\overline{CSI}$)

Pin PD2 of Port D can optionally be configured in PSDsoft Express as the PSD module Chip Select Input, $\overline{CSI}$, which is an active-low logic input. By default, pin PD2 does not have the $\overline{CSI}$ function.

When the $\overline{CSI}$ function is specified in PSDsoft Express, the $\overline{CSI}$ signal is automatically included in DPLD chip select equations for FSx, CSBOOTx, RS0, and CSIOP. When the $\overline{CSI}$ pin is driven to logic ’0’ from an external device, all of these memories will be available for READ and WRITE operations. When $\overline{CSI}$ is driven to logic '1,' none of these memories are available for selection, regardless of the address activity from the 8032, reducing power consumption. The state of the PLD and port I/O pins are not changed when $\overline{CSI}$ goes to logic ’1’ (disabled).

27.4.55 PLD non-turbo mode

The power consumption and speed of the PLDs are controlled by the Turbo Bit (Bit 3) in the csiop PMMR0 register. By setting this bit to logic '1,' the Turbo mode is turned off and both PLDs consume only standby current when ALL PLD inputs have no transitions for an extended time (65ns for 5 V devices, 100ns for 3.3 V devices), significantly reducing current consumption. The PLDs will latch their outputs and go to standby, drawing very little current. When Turbo mode is off, PLD propagation delay time is increased as shown in the AC specifications for the PSD module. Since this additional propagation delay also effects the DPLD, the response time of the memories on the PSD module is also lengthened by that same amount of time. If Turbo mode is off, the user should add an additional wait state to the 8032 BUSCON SFR register if the 8032 clock frequency is higher that a particular value. Please refer to *Table 49 on page 88* in the MCU module section.

The default state of the Turbo Bit is logic '0,' meaning Turbo mode is on by default (after power-up and reset conditions) until it is turned off by the 8032 writing to PMMR0.

27.4.56 PLD current consumption

Figure 84 and *Figure 85 on page 243* (5 V and 3.3 V devices respectively) show the relationship between PLD current consumption and the composite frequency of all the transitions on PLD inputs, indicating that a higher input frequency results in higher current consumption.

Current consumption of the PLDs have a DC component and an AC component. Both need to be considered when calculating current consumption for a specific PLD design. When Turbo mode is on, there is a linear relationship between current and frequency, and there is a substantial DC current component consumed by the PSD module when there are no transitions on PLD inputs (composite frequency is zero). The magnitude of this DC current component is directly proportional to how many product terms are used in the equations of both PLDs. PSDsoft Express generates a “fitter” report that specifies how many product terms were used in a design out of a total of 186 available product terms. *Figure 84* and *Figure 85 on page 243* both give two examples, one with 100% of the 186 product terms used, and another with 25% of the 186 product terms used.

27.4.57 Turbo mode current consumption

To determine the AC current component of the specific PLD design with Turbo mode on, the user will have to interpolate from the graph, given the number of product terms specified in the fitter report, and the estimated composite frequency of PLD input signal transitions. For

the DC component (y-axis crossing), the user can calculate the number by multiplying the number of product terms used (from fitter report) times the DC current per product term specified in the DC specifications for the PSD module. The total PLD current usage is the sum of its AC and DC components.

27.4.58 Non-turbo mode current consumption

Notice in *Figure 84* and *Figure 85 on page 243* that when Turbo mode is off, the DC current consumption is “zero” (just standby current) when the composite frequency of PLD input transitions is zero (no input transitions). Now moving up the frequency axis to consider the AC current component, current consumption remains considerably less than Turbo mode until PLD input transitions happen so rapidly that the PLDs do not have time to latch their outputs and go to standby between the transitions anymore. This is where the lines converge on the graphs, and current consumption becomes the same for PLD input transitions at this frequency and higher regardless if Turbo mode is on or off. To determine the current consumption of the PLDs with Turbo mode off, extrapolate the AC component from the graph based on number of product terms and input frequency. The only DC component in non-Turbo mode is the PSD module standby current.

The key to reducing PLD current consumption is to reduce the composite frequency of transitions on the PLD input bus, moving down the frequency scale on the graphs. One way to do this is to carefully select which signals are entering PLD inputs, not selecting high frequency signals if they are not used in PLD equations. Another way is to use PLD “Blocking Bits” to block certain signals from entering the PLD input bus.

27.4.59 PLD blocking bits

Blocking specific signals from entering the PLDs using bits of the csiop PMMR registers can further reduce PLD AC current consumption by lowering the effective composite frequency of inputs to the PLDs.

27.4.60 Blocking 8032 bus control signals

When the 8032 is active on the MCU module, four bus control signals ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, and ALE) are constantly transitioning to manage 8032 bus traffic. Each time one of these signals has a transition from logic ’1’ to '0,' or 0 to '1,' it will wake up the PLDs if operating in non-Turbo mode, or when in Turbo mode it will cause the affected PLD gates to draw current. If equations in the DPLD or GPLD do not use the signals $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, or ALE then these signals can be blocked which will reduce the AC current component substantially. These bus control signals are rarely used in DPLD equations because they are routed in silicon directly to the memory arrays of the PSD module, bypassing the PLDs. For example, it is NOT necessary to qualify a memory chip select signal with an MCU write strobe, such as “fs0 = address range & !WR_”. Only “fs0 = address range” is needed.

Each of the 8032 bus control signals may be blocked individually by writing to Bits 2, 3, 4, and 5 of the PMMR2 register shown in *Table 155 on page 226*. Blocking any of these four bus control signals only prevents them from reaching the PLDs, but they will always go to the memories directly.

However, sometimes it is necessary to use these 8032 bus control signals in the GPLD when creating interface signals to external I/O peripherals. But it is still possible to save power by dynamically unblocking the bus signals before reading/writing the external device, then blocking the signals after the communication is complete.

The user can also block an input signal coming from pin PC7 to the PLD input bus if desired by writing to Bit 6 of PMMR2.

27.4.61 Blocking common clock, CLKIN

The input CLKIN (from pin PD1) can be blocked to reduce current consumption. CLKIN is used as a common clock input to all OMC flip-flips, it is a general input to the PLD input bus, and it is used to clock the APD counter. In PSDsoft Express, the function of pin PD1 must be specified as “Common Clock Input, CLKIN” before programming the device with JTAG to get the CLKIN function.

Bit 4 of PMMR0 can be set to logic ’1’ to block CLKIN from reaching the PLD input bus, but CLKIN will still reach the APD counter.

Bit 5 of PMMR0 can be set to logic ’1’ to block CLKIN from reaching the OMC flip-flops only, but CLKIN is still available to the PLD input bus and the APD counter.

See *Table 154 on page 225* for details.

27.5 PSD module reset conditions

The PSD module receives a reset signal from the MCU module. This reset signal is referred to as the “$\overline{\text{RST}}$” input in PSD module documentation, and it is active-low when asserted. The character of the $\overline{\text{RST}}$ signal generated from the MCU module is described in *Section 19: Supervisory functions on page 89*.

Upon power-up, and while $\overline{\text{RST}}$ is asserted, the PSD module immediately loads its configuration from non-volatile bits to configure the PLDs and other items. PLD logic is operational and ready for use well before $\overline{\text{RST}}$ is de-asserted. The state of PLD outputs are determined by equations specified in PSDsoft Express.

The Flash memories are reset to Read Array mode after any assertion of $\overline{\text{RST}}$ (even if a program or erase operation is occurring).

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

Once the UPSD33xx is up and running, any subsequent reset operation is referred to as a warm reset, until power is turned off again. Some PSD module functions are reset in different ways depending if the reset condition was caused from a power-up reset or a warm reset. *Table 158 on page 233* summarizes how PSD module functions are affected by power-up and warm resets, as well as the affect of PSD module power-down mode (from APD).

The I/O pins of PSD module Ports A, B, C, and D do not have weak internal pull-ups.

In MCU I/O mode, Latched Address Out mode, and Peripheral I/O mode, the pins of Ports A, B, C, and D become standard CMOS inputs during a reset condition. If no external devices are driving these pins during reset, then these inputs may float and draw excessive current. If low power consumption is critical during reset, then these floating inputs should be pulled up externally to V_{DD} with a weak (100KΩ minimum) resistor.

In PLD I/O mode, pins of Ports A, B, C, and D may also float during reset if no external device is driving them, and if there is no equation specified for the DPLD or GPLD to make

28 AC/DC parameters

These tables describe the AD and DC parameters of the UPSD33xx Devices:

- DC electrical specification
- AC timing specification
- PLD timing
  - Combinatorial timing
  - Synchronous Clock mode
  - Asynchronous Clock mode
  - Input macrocell timing
- MCU module timing
  - READ timing
  - WRITE timing
  - Power-down and $\overline{\text{RESET}}$ timing

The following are issues concerning the parameters presented:

- In the DC specification the supply current is given for different modes of operation.
- The AC power component gives the PLD, Flash memory, and SRAM mA/MHz specification. *Figure 84* and *Figure 85 on page 243* show the PLD mA/MHz as a function of the number of Product Terms (PT) used.
- In the PLD timing parameters, add the required delay when Turbo Bit is '0.'

Figure 84. PLD I_{CC} /frequency consumption (5 V range)

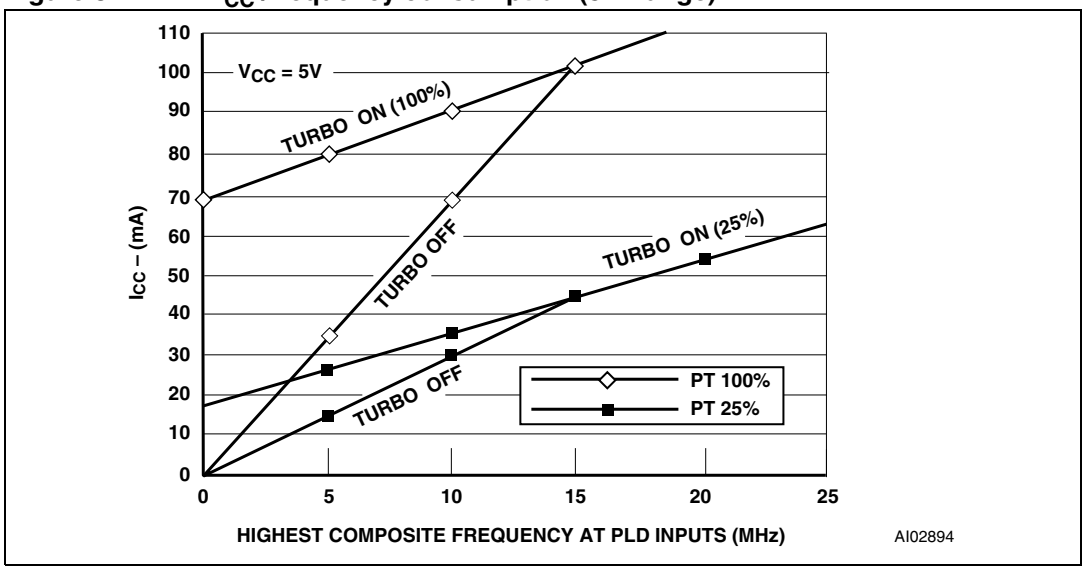


Table 170. n, m, and x, y values

# of PFQCLK in BUSCON register	PSEN (code) cycle		READ cycle		WRITE cycle	
	n	m	n	m	x	y
3	1	2	-	-	-	-
4	2	3	2	3	2	1
5	3	4	3	4	3	2
6	4	5	4	5	4	3
7	-	-	5	6	5	4

Figure 88. External WRITE cycle (80-pin device only)

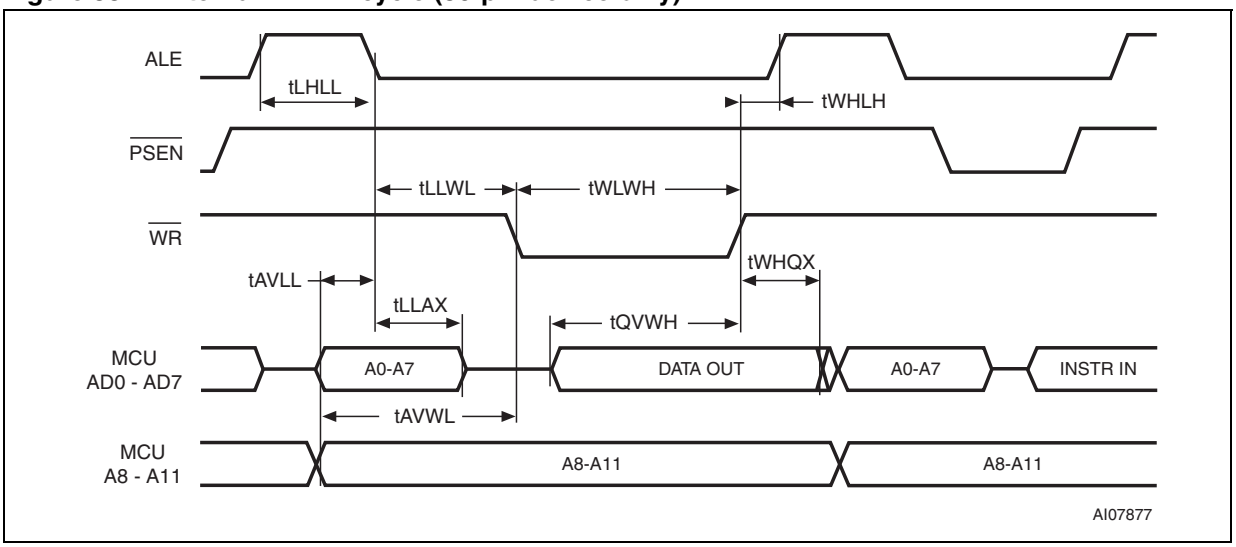


Table 171. External WRITE cycle AC characteristics (3 V or 5 V device)

Symbol	Parameter	40 MHz oscillator(1)		Variable oscillator $1/t_{CLCL}$ = 8 to 40 MHz		Unit
		Min	Max	Min	Max	
t_{LHLL}	ALE pulse width	17		$t_{CLCL} - 8$		ns
t_{AVLL}	Address Setup to ALE	13		$t_{CLCL} - 12$		ns
t_{LLAX}	Address hold after ALE	7.5		$0.5t_{CLCL} - 5$		ns
t_{WLWH}	WR pulse width(2)	40		$xt_{CLCL} - 10$		ns
t_{LLWL}	ALE to $\overline{WR}$	7.5		$0.5t_{CLCL} - 5$		ns
t_{AVWL}	Address valid to $\overline{WR}$	27.5		$1.5t_{CLCL} - 10$		ns
t_{WHLH}	$\overline{WR}$ high to ALE high	6.5	14.5	$0.5t_{CLCL} - 6$	$0.5t_{CLCL} + 2$	ns
t_{QVWH}	Data setup before $\overline{WR}$(y)	20		$yt_{CLCL} - 5$		ns
t_{WHQX}	Data hold after $\overline{WR}$	6.5	14.5	$0.5t_{CLCL} - 6$	$0.5t_{CLCL} + 2$	ns

1. BUSCON register is configured for 4 PFQCLK.
2. Refer to *Table 172* for “n” and “m” values.

Figure 95. Peripheral I/O WRITE timing

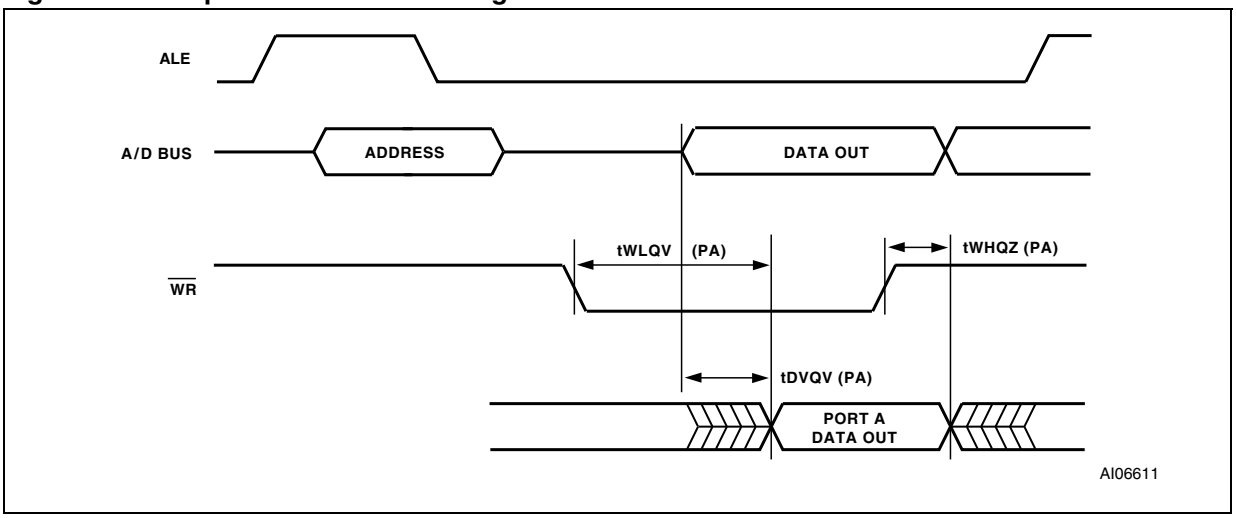


Table 185. Port A peripheral data mode WRITE Timing (5 V PSD module)

Symbol	Parameter	Conditions	Min	Max	Unit
$t_{WLQV-PA}$	$\overline{WR}$ to data propagation delay			25	ns
$t_{DVQV-PA}$	Data to Port A data propagation delay	(1)		22	ns
$t_{WHQZ-PA}$	$\overline{WR}$ Invalid to Port A Tri-state			20	ns

1. Data stable on Port 0 pins to data on Port A.

Table 186. Port A peripheral data mode WRITE Timing (3 V PSD module)

Symbol	Parameter	Conditions	Min	Max	Unit
$t_{WLQV-PA}$	$\overline{WR}$ to data propagation delay			42	ns
$t_{DVQV-PA}$	Data to Port A data propagation delay	(1)		38	ns
$t_{WHQZ-PA}$	$\overline{WR}$ Invalid to Port A Tri-state			33	ns

1. Data stable on Port 0 pins to data on Port A.

Table 187. Supervisor Reset and LVD.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{RST\_LO\_IN}$	Reset input duration		1(1)			µs
t_{RST_ACTV}	Generated Reset duration	f_{OSC} = 40 MHz	10(2)			ms
t_{RST_FIL}	Reset input spike filter			1		µs
V_{RST_HYS}	Reset input hysteresis	V_{CC} = 3.3 V		0.1		V
V_{RST_THRESH}	LVD trip threshold	V_{CC} = 3.3 V	2.4	2.6	2.8	V

1. 25µs minimum to abort a Flash memory program or erase cycle in progress.
2. As F_{OSC} decreases, t_{RST_ACTV} increases. Example: t_{RST_ACTV} = 50ms when F_{OSC} = 8 MHz

32 Part numbering

Table 193. Ordering information scheme

Example: UPSD 33 3 4 D V – 40 U 6 T

Device Type

UPSD = Microcontroller PSD

Family

33 = Turbo core

SRAM Size

1 = 2 Kbyte

3 = 8 Kbyte

5 = 32 Kbyte

Main Flash memory Size

2 = 64 Kbyte

3 = 128 Kbyte

4 = 256 Kbyte

IP Mix

D = IP Mix: I^2C, SPI, UART(2), IrDA, ADC, Supervisor, PCA

Operating voltage

blank = V_{CC} = 4.5 to 5.5 V

V = V_{CC} = 3.0 to 3.6V

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 a list of available options (e.g., Speed, Package) or for further information on any aspect of this device, please contact the ST Sales Office nearest to you.