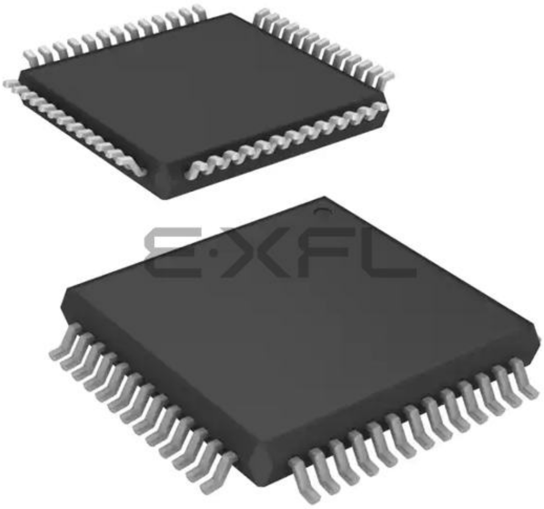


Welcome to E-XFL.COM

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, USB
Peripherals	LVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	288KB (288K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K 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	52-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/upsd3454eb40t6

11.2 Data pointer mode register, DPTM (86h)

The two “background” data pointers, DPTR0 and DPTR1, can be configured to automatically increment, decrement, or stay the same after a MOVX instruction accesses the DPTR register. Only the currently selected pointer will be affected by the increment or decrement. This feature is controlled by the DPTM register defined in *Table 15*.

The automatic increment or decrement function is effective only for the MOVX instruction, and not MOVC or any other instruction that uses the DTPR register.

11.2.1 Firmware example

The 8051 assembly code illustrated in *Table* shows how to transfer a block of data bytes from one XDATA address region to another XDATA address region. Auto-address incrementing and auto-pointer toggling will be used.

Table 15. DPTM: data pointer mode register (SFR 86h, reset value 00h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	–	MD11	MD10	MD01	MD00

Table 16. DPTM register bit definition

Bit	Symbol	R/W	Definition
7-4	–	–	Reserved
3-2	MD[11:10]	R,W	DPTR1 mode bits 00: DPTR1 No change 01: Reserved 10: Auto Increment 11: Auto Decrement
1-0	MD[01:00]	R,W	DPTR0 mode bits 00: DPTR0 No change 01: Reserved 10: Auto Increment 11: Auto Decrement

Figure 13. Clock generation logic

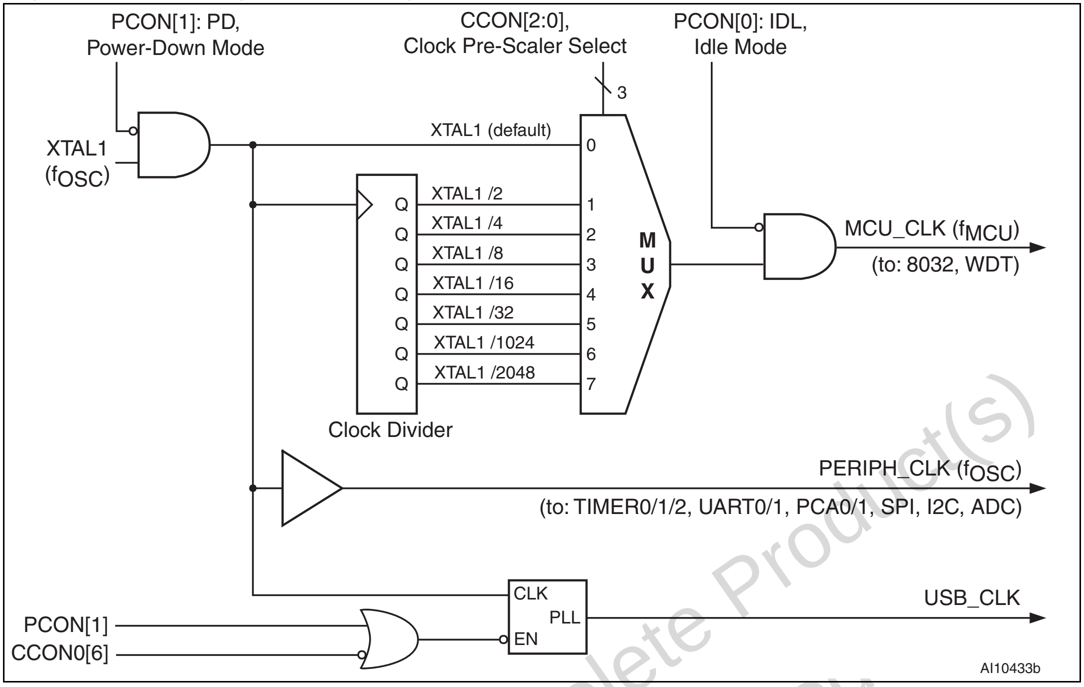


Table 27. CCON0: clock control register (SFR F9h, reset value 50h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLLM[4]	PLLEN	UPLLCE	DBGCE	CPUAR	CPUPS[2:0]		

Table 28. CCON0 register bit definition

Bit	Symbol	R/W	Definition
7	PLLM[4]	R,W	Upper bit of the 5-bit PLLM[4:0] Multiplier (Default: '0' for PLLM = 00h)
6	PLLEN	R,W	PLL Enable 0 = Disable PLL operation 1 = Enable PLL operation (Default condition after reset)
5	UPLLCE	R,W	USB Clock Enable 0 = USB clock is disabled (Default condition after reset) 1 = USB clock is enabled
4	DBGCE	R,W	Debug Unit Breakpoint Comparator Enable 0 = JTAG Debug Unit comparators are disabled 1 = JTAG Debug Unit comparators are enabled (Default condition after reset)

17 I/O ports of mcu module

The MCU module has three 8-bit I/O ports: Port 1, Port 3, and Port 4. The PSD module has four other I/O ports: Port A, B, C, and D. This section describes only the I/O ports on the MCU module.

I/O ports will function as bidirectional general-purpose I/O (GPIO), but the port pins can have alternate functions assigned at run-time by writing to specific SFRs. The default operating mode (during and after reset) for all three ports is GPIO input mode. Port pins that have no external connection will not float because each pin has an internal weak pull-up (~150 kΩ) to V_{CC}.

I/O ports 3 and 4 are 5 V tolerant, meaning they can be driven/pulled externally up to 5.5 V without damage. The pins on Port 4 have a higher current capability than the pins on Ports 1 and 3.

Three additional MCU ports (only on 80-pin UPSD34xx devices) are dedicated to bring out the 8032 MCU address, data, and control signals to external pins. One port, named MCUAD[7:0], has eight multiplexed address/data bidirectional signals. The third port has MCU bus control outputs: read, write, program fetch, and address latch. These ports are typically used to connect external parallel peripherals and memory devices, but they may NOT be used as GPIO. Notice that the eight upper address signals do not come out to pins on the port. If high-order address signals are required on external pins (MCU addresses A[15:8]), then these address signals can be brought out as needed to PLD output pins or to the Address Out mode pins on PSD module ports. See PSD module section, "*Section 28.5.40: Latched address output mode on page 238* for details.

Figure 15 on page 80 represents the flexibility of pin function routing controlled by the SFRs. Each of the 24 pins on three ports, P1, P3, and P4, may be individually routed on a pin-by-pin basis to a desired function.

17.1 MCU port operating modes

MCU port pins can operate as GPIO or as alternate functions (see *Figure 16 on page 80* through *Figure 18 on page 81*).

Depending on the selected pin function, a particular pin operating mode will automatically be used:

- GPIO - Quasi-bidirectional mode
- UART0, UART1 - Quasi-bidirectional mode
- SPI - Quasi-bidirectional mode
- I2C - Open drain mode
- ADC - Analog input mode
- PCA output - Push-Pull mode
- PCA input - Input only (Quasi-bidirectional)
- Timer 0,1,2 - Input only (Quasi-bidirectional)

Table 69. Commonly used baud rates generated from timer 1 (continued)

UART mode	f_{OSC} MHz	Desired baud rate	Resultant baud rate	Baud rate deviation	SMOD bit in PCON	Timer 1		
						$C/\overline{T}$ Bit in TMOD	Timer mode in TMOD	TH1 reload value (hex)
Modes 1 or 3	3.6864	19200	19200	0	1	0	2	FF
Modes 1 or 3	3.6864	9600	9600	0	1	0	2	FE
Modes 1 or 3	1.8432	9600	9600	0	1	0	2	FF
Modes 1 or 3	1.8432	4800	4800	0	1	0	2	FE

21.4 More about UART mode 0

Refer to the block diagram in *Figure 30 on page 113*, and timing diagram in *Figure 31 on page 113*.

Transmission is initiated by any instruction which writes to the SFR named SBUF. At the end of a write operation to SBUF, a 1 is loaded into the 9th position of the transmit shift register and tells the TX Control unit to begin a transmission. Transmission begins on the following MCU machine cycle, when the “SEND” signal is active in *Figure 31*.

SEND enables the output of the shift register to the alternate function on the port containing pin RxD, and also enables the SHIFT CLOCK signal to the alternate function on the port containing the pin, TxD. At the end of each SHIFT CLOCK in which SEND is active, the contents of the transmit shift register are shifted to the right one position.

As data bits shift out to the right, zeros come in from the left. When the MSB of the data byte is at the output position of the shift register, then the '1' that was initially loaded into the 9th position, is just to the left of the MSB, and all positions to the left of that contain zeros. This condition flags the TX Control unit to do one last shift, then deactivate SEND, and then set the interrupt flag TI. Both of these actions occur at S1P1.

Reception is initiated by the condition REN = 1 and RI = 0. At the end of the next MCU machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register, and in the next clock phase activates RECEIVE. RECEIVE enables the SHIFT CLOCK signal to the alternate function on the port containing the pin, TxD. Each pulse of SHIFT CLOCK moves the contents of the receive shift register one position to the left while RECEIVE is active. The value that comes in from the right is the value that was sampled at the RxD pin. As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the right-most position arrives at the left-most position in the shift register, it flags the RX Control unit to do one last shift, and then it loads SBUF. After this, RECEIVE is cleared, and the receive interrupt flag RI is set.

Figure 34. UART mode 2, block diagram

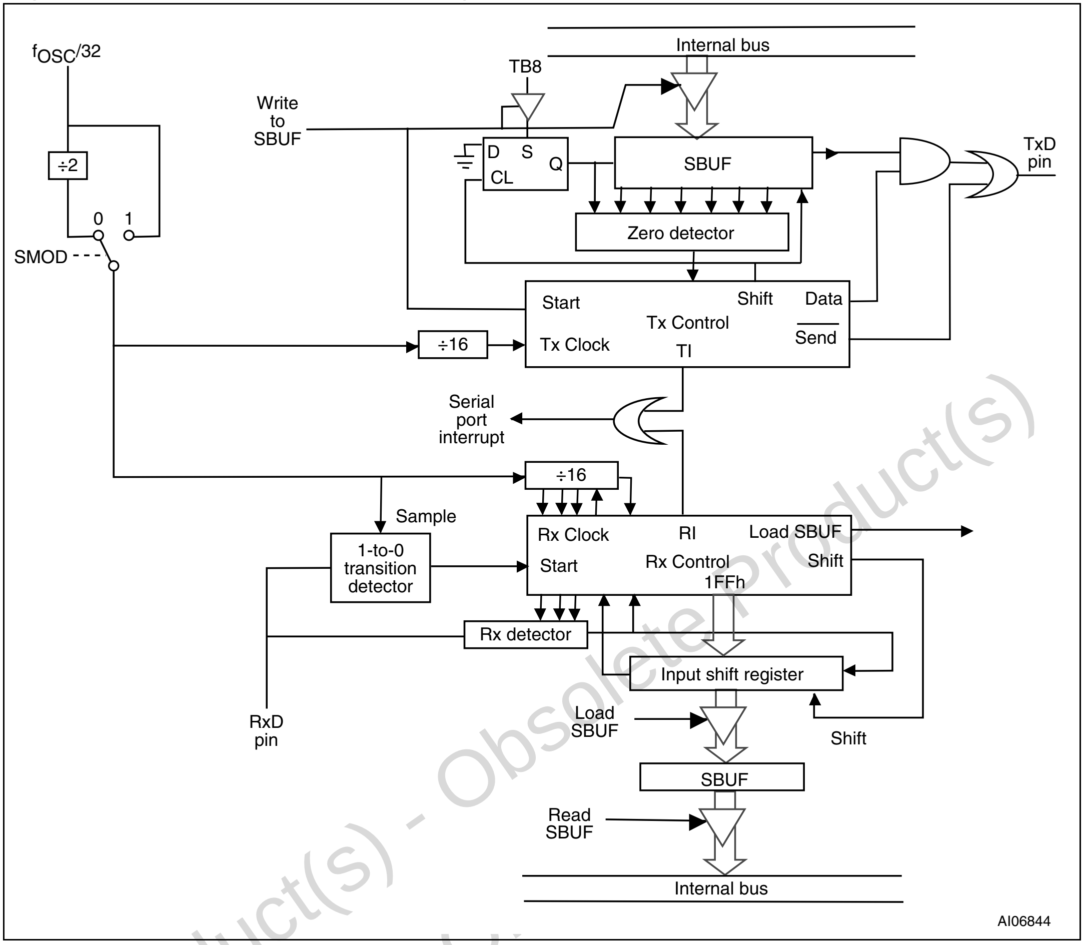
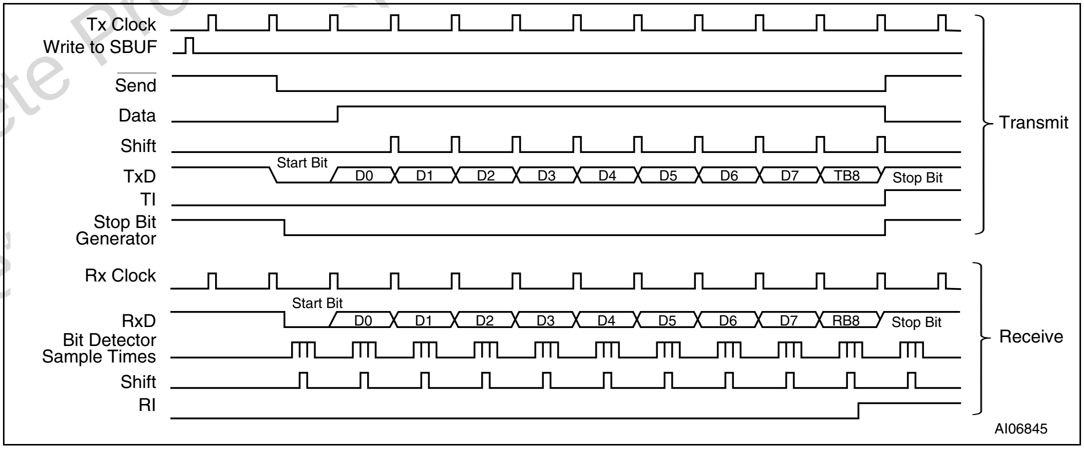


Figure 35. UART mode 2, timing diagram



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 $1/f_{OSC}$.

Table 139. ACON register bit definition (continued)

Bit	Symbol	Function
1	ADST	ADC start bit 0 = Force to zero 1 = Start ADC, then after one cycle, the bit is cleared to '0.'
0	ADSF	ADC Status Bit 0 = ADC conversion is not completed 1 = ADC conversion is completed. The bit can also be cleared with software.

Table 140. ADCPS register details (SFR 94h, Reset Value 00h)

Bit	Symbol	Function
7:4	–	Reserved
3	ADCCE	ADC conversion reference clock enable 0 = ADC reference clock is disabled (default) 1 = ADC reference clock is enabled
2:0	ADCPS[2:0]	ADC reference clock prescaler Only three Prescaler values are allowed: ADCPS[2:0] = 0, for f_{OSC} frequency 16MHz or less. Resulting ADC clock is f_{OSC}. ADCPS[2:0] = 1, for f_{OSC} frequency 32MHz or less. Resulting ADC clock is $f_{OSC}/2$. ADCPS[2:0] = 2, for f_{OSC} frequency 32MHz > 40MHz. Resulting ADC clock is $f_{OSC}/4$.

Table 141. ADAT0 register (SFR 95h, reset value 00h)

Bit	Symbol	Function
7:0	–	Store ADC output, Bit 7 - 0

Table 142. ADAT1 register (SFR 96h, reset value 00h)

Bit	Symbol	Function
7:2	–	Reserved
1.. 0	–	Store ADC output, Bit 9, 8

27 Programmable counter array (PCA) with PWM

There are two Programmable Counter Array blocks (PCA0 and PCA1) in the UPSD34xx. 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)

27.1 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 143 on page 182* lists the SFR registers in the PCA blocks.

Figure 57. PCA0 block diagram

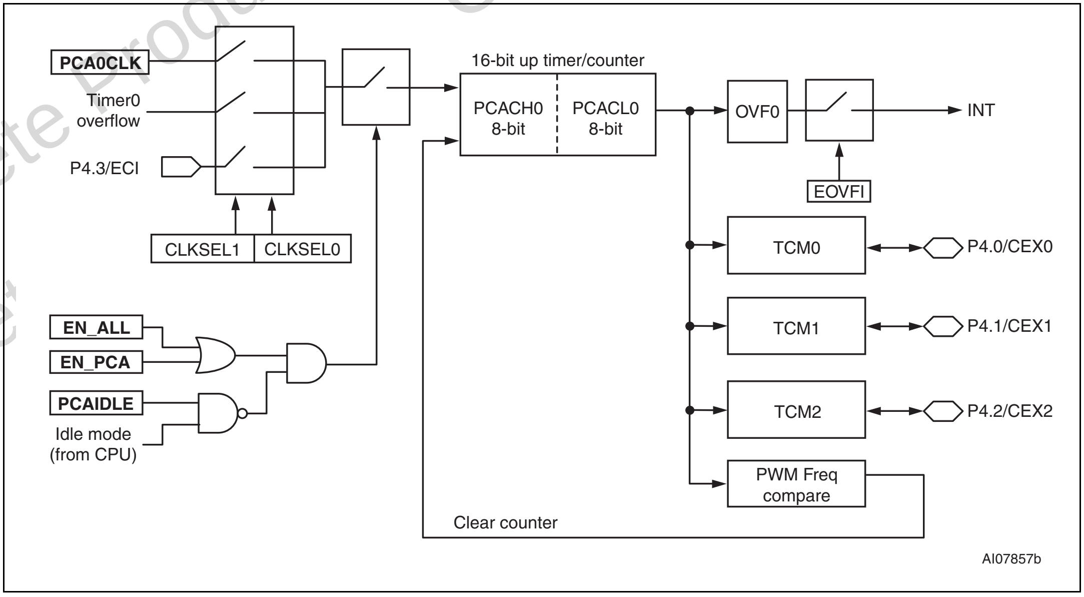


Table 144. CCON2 register bit definition (SFR 0FBh, reset value 10h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	PCA0CE	PCA0PS3	PCA0PS2	PCA0PS1	PCA0PS0

Table 145. CCON2 register bit definition

Bit	Symbol	R/W	Definition
4	PCA0CE	R/W	PCA0 Clock Enable 0 = PCA0CLK is disabled 1 = PCA0CLK is enabled (default)
3:0	PCA0PS [3:0]	R/W	PCA0 Prescaler $f_{PCA0CLK} = f_{OSC}$ / (2 ^ PCA0PS[3:0]) Divisor range: 1, 2, 4, 8, 16... 16384, 32768

Table 146. CCON3 register bit definition (SFR 0FCh, reset value 10h)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	–	PCA1CE	PCA1PS3	PCA1PS2	PCA1PS1	PCA1PS0

Table 147. CCON3 register bit definition

Bit	Symbol	R/W	Definition
4	PCA1CE	R/W	PCA1 Clock Enable 0 = PCA1CLK is disabled 1 = PCA1CLK is enabled (default)
3:0	PCA1PS [3:0]	R/W	PCA1 Prescaler $f_{PCA1CLK} = f_{OSC}$ / (2 ^ PCA1PS[3:0]) Divisor range: 1, 2, 4, 8, 16... 16384, 32768

27.3 Operation of TCM modes

Each of the TCM in a PCA block supports four modes of operation. However, an exception is when the TCM is configured in PWM Mode with programmable frequency. In this mode, all TCM in a PCA block must be configured in the same mode or left to be not used.

27.4 Capture mode

The CAPCOM registers in the TCM are loaded with the counter values when an external pin input changes state. The user can configure the counter value to be loaded by positive edge, negative edge or any transition of the input signal. At loading, the TCM can generate an interrupt if it is enabled.

27.5 Timer mode

The TCM modules can be configured as software timers by enable the comparator. The user writes a value to the CAPCOM registers, which is then compared with the 16-bit counter. If there is a match, an interrupt can be generated to CPU.

28.1.4 Secondary Flash memory

The smaller Secondary Flash memory is also divided into equal sized sectors that are individually selectable by the Decode PLD signals, named CSBOOTx, one signal for each Secondary Flash memory sector. Each sector can be located at any address within 8032 program address space (accessed with $\overline{PSEN}$) or XDATA space (accessed with $\overline{RD}$ or $\overline{WR}$) as defined with PSDsoft Express. The user only has to specify an address range for each segment, and specify if Secondary Flash memory will reside in 8032 data or program address space, and then $\overline{PSEN}$, $\overline{RD}$, or $\overline{WR}$ are automatically activated for the specified range. 8032 firmware is easily programmed into Secondary Flash memory using PSDsoft Express and others. See *Table 157 on page 193* for Secondary Flash sector sizes.

28.1.5 SRAM

The SRAM is selected by a single signal, named RS0, from the Decode PLD. SRAM may be located at any address within 8032 XDATA space (accessed with $\overline{RD}$ or $\overline{WR}$). These choices are specified using PSDSoft Express, where the user specifies an SRAM address range. See *Table 157 on page 193* for SRAM sizes.

Table 157. UPSD34xx memory configuration

Device	Main Flash memory			Secondary Flash memory			SRAM
	Total Flash size (bytes)	Individual sector size (bytes)	Number of sectors (Sector Select signal)	Total Flash size (bytes)	Individual sector size (bytes)	Number of sectors (Sector Select signal)	SRAM size (bytes)
UPSD3422	64 KB	16 KB	4 (FS0-3)	32 KB	8 KB	4 (CSBOOT0-3)	4 KB
UPSD3433	128 KB	16 KB	8 (FS0-7)	32 KB	8 KB	4 (CSBOOT0-3)	8 KB
UPSD3434	256 KB	32 KB	8 (FS0-7)	32 KB	8 KB	4 (CSBOOT0-3)	8 KB
UPSD3454	256 KB	32 KB	8 (FS0-7)	32 KB	8 KB	4 (CSBOOT0-3)	32 KB

28.1.6 Runtime control registers, csiop

A block of 256 bytes is decoded inside the PSD module for module control and status (see *Table 162 on page 205*). The base address of these 256 locations is referred to in this data sheet as csiop (Chip Select I/O Port), and is selected by the Decode PLD output signal, $\overline{CSIOP}$. The csiop registers are always viewed by the 8032 as XDATA, and are accessed with $\overline{RD}$ and $\overline{WR}$ signals. The address range of csiop is specified using PSDsoft Express where the user only has to specify an address range of 256 bytes, and then the $\overline{RD}$ or $\overline{WR}$ signals are automatically activated for the specified range. Individual registers within this block are accessed with an offset from the specified csiop base address. 39 registers are used out of the 256 locations to control the output state of I/O pins, to read I/O pins, to set the memory page, to control 8032 program and data address space, to control power management, to READ/WRITE macrocells inside the General PLD, and other functions during runtime. Unused locations within csiop are reserved and should not be accessed.

28.1.7 Memory page register

8032 MCU architecture has an inherent size limit of 64K bytes in either program address space or XDATA space. Some UPSD34xx devices have much more memory that 64K, so special logic such as this page register is needed to access the extra memory. This 8-bit

Table 158. General I/O pins on PSD module

Pkg	Port A	Port B	Port C	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 79 on page 232*). 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 bidirectionally 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.

28.1.15 JTAG port

In-System Programming (ISP) can be performed through the JTAG signals on Port C. This serial interface allows programming of the entire PSD module device or subsections of the PSD module (for example, only Flash memory but not the PLDs) without the participation of the 8032. A blank UPSD34xx device soldered to a circuit board can be completely programmed in 10 to 25 seconds. The four basic JTAG signals on Port C; TMS, TCK, TDI, and TDO form the IEEE-1149.1 interface. The PSD module does not implement the IEEE-1149.1 Boundary Scan functions, but uses the JTAG interface for ISP an 8032 debug. The PSD module can reside in a standard JTAG chain with other JTAG devices and it will remain in BYPASS mode when other devices perform JTAG functions.

ISP programming time can be reduced as much as 30% by using two optional JTAG signals on Port C, TSTAT and $\overline{\text{TERR}}$, in addition to TMS, TCK, TDI and TDO, and this is referred to as "6-pin JTAG". The FlashLINK JTAG programming cable is available from STMicroelectronics and PSDsoft Express software is available at no charge from www.st.com/psm. More JTAG ISP information maybe found in *Section 28.6.1: JTAG ISP and JTAG debug on page 257*.

The MCU module is also included in the JTAG chain within the UPSD34xx device for 8032 debugging and emulation. While debugging, the PSD module is in BYPASS mode. Conversely, during ISP, the MCU module is in BYPASS mode.

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

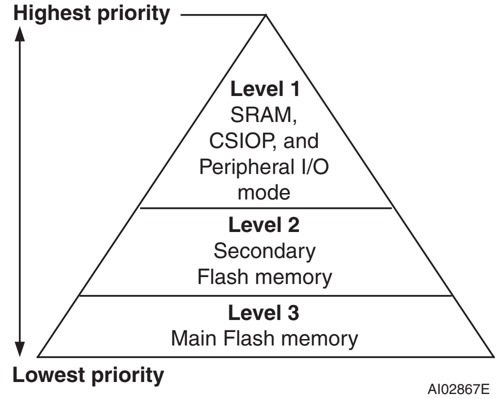


Table 162. CSIOP registers and their offsets (in hexadecimal) (continued)

Register name	Port A (80-pin)	Port B	Port C	Port D	Other	Description	Link
PMMR0					B0h	Power management register 0. WRITE and READ.	*Table 200 on page 249*
PMMR2					B4h	Power management register 2. WRITE and READ.	*Table 201 on page 249*
PMMR3					C7h	Power management register 3. WRITE and READ. However, Bit 1 can be cleared only by a reset condition.	*Table 202 on page 250*
Page					E0h	Memory page register. WRITE and READ.	*Table 62 on page 194*
VM (Virtual Memory)					E2h	Places PSD module memories into 8032 Program Address Space and/or 8032 XDATA Address Space. (VM overrides initial non-volatile setting that was specified in PSDsoft Express. Reset restores initial setting)	*Table 160 on page 203*

28.5 PSD module detailed operation

Specific details are given here for the following key functional areas on the PSD module:

- Flash Memories
- PLDs (DPLD and GPLD)
- I/O Ports
- Power Management
- JTAG ISP and Debug Interface

28.5.1 Flash memory operation

The Flash memories are accessed through the 8032 Address, Data, and Control Bus interfaces. Flash memories (and SRAM) cannot be accessed by any other bus master other than the 8032 MCU (these are not dual-port memories).

The 8032 cannot write to Flash memory as it would an SRAM (supply address, supply data, supply $\overline{\text{WR}}$ strobe, assume the data was correctly written to memory). Flash memory must first be “unlocked” with a special instruction sequence of byte WRITE operations to invoke an internal algorithm inside either Flash memory array, then a single data byte is written (programmed) to the Flash memory array, then programming status is checked by a byte READ operation or by checking the Ready/$\overline{\text{Busy}}$ pin (PC3). *Table 163 on page 209* lists all of the special instruction sequences to program a byte to either of the Flash memory arrays, erase the arrays, and check for different types of status from the arrays.

Table 163. Flash memory instruction sequences(1)(2)

Instr. Seq.	Bus Cycle 1	Bus Cycle 2	Bus Cycle 3	Bus Cycle 4	Bus Cycle 5	Bus Cycle 6	Bus Cycle 7	Link
Read Memory Contents (Read Array mode)	Read byte from any valid Flash memory addr							*Read memory contents on page 210*
Program (write) a byte to Flash Memory	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write A0h to X555h (*command*)	Write data byte to address				*Programming Flash memory on page 212*
Bypass Unlock	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 20h to X555h (*command*)					*Bypassed unlock sequence on page 215*
Program a byte to Flash Memory with Bypassed Unlock	Write A0h to XXXXh (*command*)	Write data byte to address						*Bypassed unlock sequence on page 215*
Reset Bypass Unlock	Write 90h to XXXXh (*command*)	Write 00h to XXXXh (*command*)						*Bypassed unlock sequence on page 215*
Flash Bulk Erase(3)	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 80h to X555h (*command*)	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 10h to X555h (*command*)		*Flash bulk erase on page 216*
Flash Sector Erase	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 80h to X555h (*command*)	Write AAh to X555h (*unlock*)	Write 55h to XAAAh (*unlock*)	Write 30h to desired Sector (*command*)	Write 30h to another Sector (*command*)	*Flash sector erase on page 216*
Suspend Sector Erase	Write B0h to address that activates FSx or CSBOOTx where erase is in progress (*command*)							*Suspend sector erase on page 217*

The Erase Time-out Flag Bit (DQ3) reflects the time-out period allowed between two consecutive sector erase instruction sequence bytes. If multiple sector erase commands are desired, the additional sector erase commands (30h) must be sent by the 8032 to another sector within 80µs after the previous sector erase command. DQ3 is 0 before this time period has expired, indicating it is OK to issue additional sector erase commands. DQ3 will go to logic ’1’ if the time has been longer than 80µs since the previous sector erase command (time has expired), indicating that is not OK to send another sector erase command. In this case, the 8032 must start a new sector erase instruction sequence (unlock and command), beginning again after the current sector erase operation has completed.

During a Sector Erase operation, 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), as detailed in *Section 28.5.5: Reading the erase/program status bits on page 210*.

During a Sector Erase operation, a Flash memory accepts only Reset Flash and Suspend Sector Erase instruction sequences. Erasure of one Flash memory sector may be suspended, in order to read data from another Flash memory sector, and then resumed.

The address provided with the initial Flash Sector Erase command sequence (*Table 163 on page 209*) must select the first desired sector (FSx or CSBOOTx) to erase. Subsequent sector erase commands that are appended within the time-out period must be addressed to other desired segments within the same Flash memory array.

28.5.18 Suspend sector erase

When a Sector Erase operation is in progress, the Suspend Sector Erase instruction sequence can be used to suspend the operation by writing B0h to any valid address within the Flash array that currently is undergoing an erase operation. This allows reading of data from a different Flash memory sector within the same array after the Erase operation has been suspended. Suspend Sector Erase is accepted only during an Erase operation.

There is up to 15µs delay after the Suspend Sector Erase command is accepted and the array goes to Read Array mode. The 8032 will monitor the Toggle Flag Bit (DQ6) to determine when the erase operation has halted and Read Array mode is active.

If a Suspend Sector Erase instruction sequence was executed, the following rules apply:

- Attempting to read from a Flash memory sector that was being erased outputs invalid data.
- Reading from a Flash memory sector that was *not* being erased is valid.
- The Flash memory *cannot* be programmed, and only responds to Resume Sector Erase and Reset Flash instruction sequences.
- If a Reset Flash instruction sequence is received, data in the Flash memory sector that was being erased is invalid.

28.5.19 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 163 on page 209*.

Table 189. MCU I/O mode port D direction register (address = csiop + offset 15h)[1][2]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
N/A	N/A	N/A	N/A	N/A	PD2[3]	PD1	N/A

1. For each bit, 1 = out from UPSD34xx port pin1, 0 = in to PSD34xx port pin.
2. Default state for register is 00h after reset or power-up.
3. Not available on 52-pin uPSD34xx devices.

28.5.39 PLD I/O mode

Pins on Ports A, B, C, and D can serve as inputs to either the DPLD or the GPLD. Inputs to these PLDs from Ports A, B, and C are routed through IMCs before reaching the PLD input bus. Inputs to the PLDs from Port D do not pass through IMCs, but route directly to the PLD input bus.

Pins on Ports A, B, and C can serve as outputs from GPLD OMCs, and Port D pins can be outputs from the DPLD (external chip-selects) which do not consume OMCs.

Whenever a pin is specified to be a PLD output, it cannot be used for MCU I/O mode, or other pin modes. If a pin is specified to be a PLD input, it is still possible to read the pin using MCU I/O input mode with the csiop register Data In. Also, the csiop Direction register can still affect a pin which is used for a PLD input. The csiop Data Out register has no effect on a PLD output pin.

Each pin on Ports A, B, C, and D have a tri-state buffer at the final output stage. The Output Enable signal for this buffer is driven by the logical OR of two signals. One signal is an Output Enable signal generated by the AND-OR array (from an .oe equation specified in PSDsoft), and the other signal is the output of the csiop Direction register. This logic is shown in *Figure 79 on page 232*. At power-on, all port pins default to high-impedance input (Direction registers default to 00h). However, if an equation is written for the Output Enable that is active at power-on, then the pin will behave as an output.

PLD I/O equations are specified in PSDsoft Express and programmed into the uPSD using JTAG. *Figure 80* shows a very simple combinatorial logic example which is implemented on pins of Port B.

To give a general idea of how PLD logic is implemented using PSDsoft Express, *Figure 81 on page 237* illustrates the pin declaration window of PSDsoft Express, showing the PLD output at pin PB0 declared as “Combinatorial” in the “PLD Output” section, and a signal name, “pld_out”, is specified. The other three signals on pins PB1, PB2, and PB3 would be declared as “Logic or Address” in the “PLD Input” section, and given signal names.

In the “Design Assistant” window of PSDsoft Express shown in *Figure 82 on page 238*, the user simply enters the logic equation for the signal “pld_out” as shown. The user can either type in the logic statements or enter them using a point-and-click method, selecting various signal names and logic operators available in the window.

After PSDsoft Express has accepted and realized the logic from the equations, it synthesizes the logic statement:

pld_out = (pld_in_1 # pld_in_2) & !pld_in_3;

to be programmed into the GPLD. See the PSDsoft User’s Manual for all the steps.

28.5.47 Individual port structures

Ports A, B, C, and D have some differences. The structure of each individual port is described in the next sections.

28.5.48 Port A structure

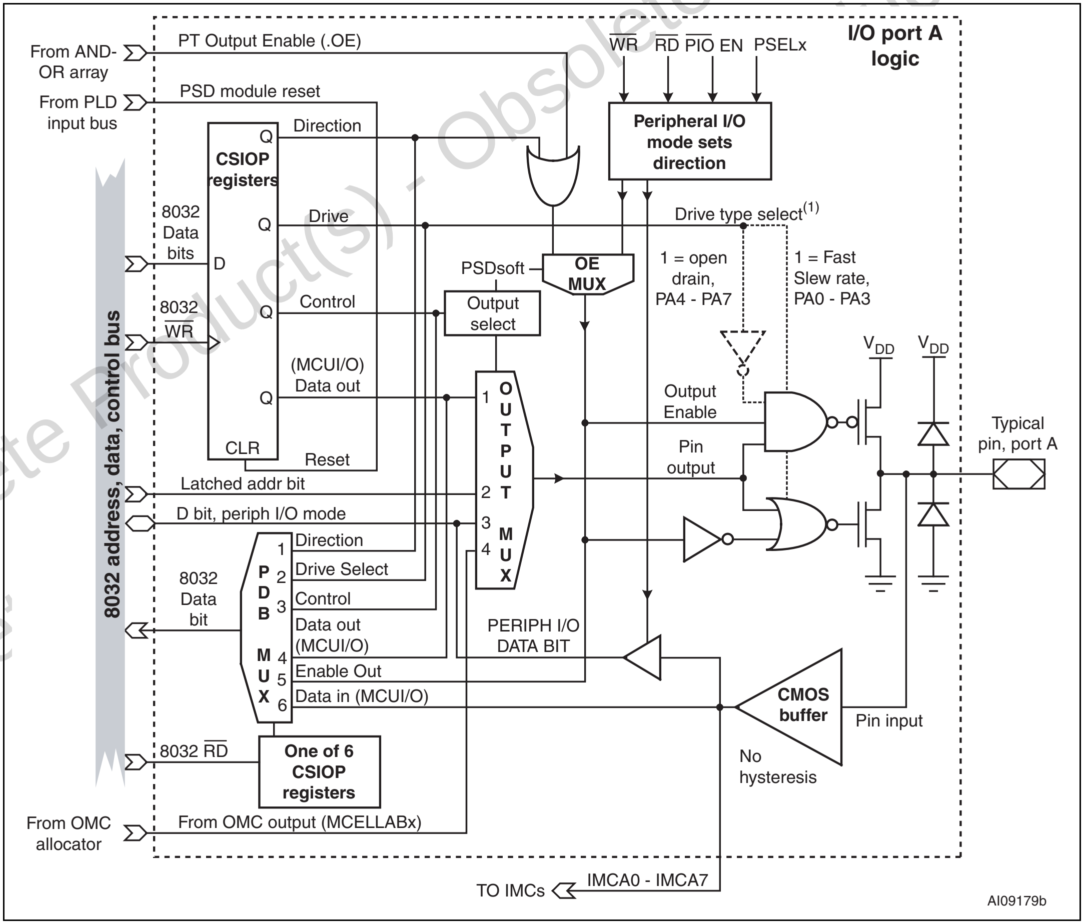
Port A supports the following operating modes:

- MCU I/O Mode
- GPLD Output Mode from Output Macrocells MCELLABx
- GPLD Input Mode to Input Macrocells IMCAx
- Latched Address Output Mode
- Peripheral I/O Mode

Port A also supports Open Drain/Slew Rate output drive type options using csiop Drive Select registers. Pins PA0-PA3 can be configured to fast slew rate, pins PA4-PA7 can be configured to Open Drain Mode.

See *Figure 84* for details.

Figure 84. Port A structure



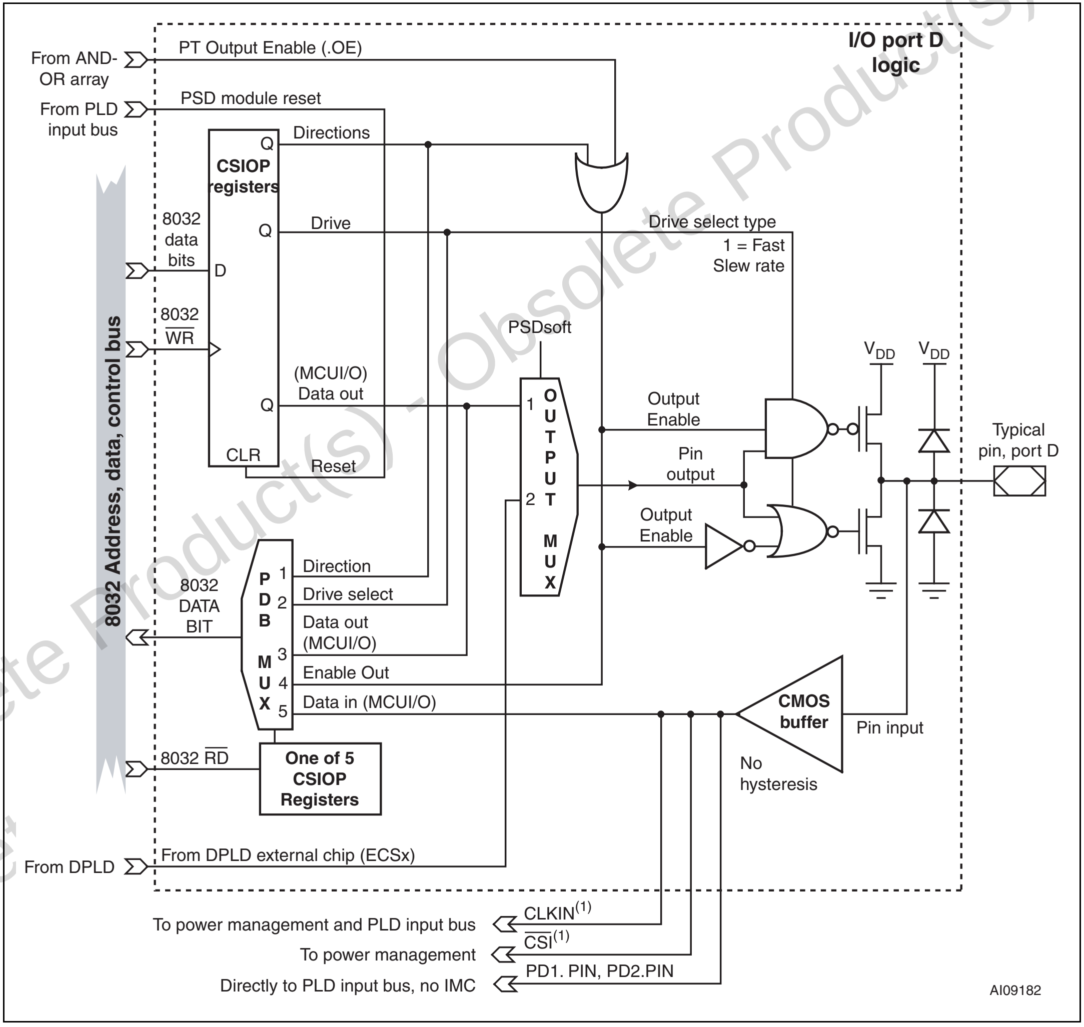
Note: 1 Port pins PA0-PA3 are capable of Fast Slew Rate output drive option. Port pins PA4-PA7 are capable of Open Drain output option.

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

- PD1 can be used as a common clock input to all 16 OMC Flip-flops (see *Section 28.1.11: OMCs on page 195*) and also the *Section 28.5.53: Automatic power-down (APD) on page 250*.
- PD2 can be used as a common chip select signal ($\overline{CSI}$) for the Flash and SRAM memories on the PSD module (see *Section 28.5.55: Chip select input (CSI) on page 253*). If driven to logic ’1’ by an external source, $\overline{CSI}$ will force the Flash memory into standby mode regardless of what other internal memory select signals are doing on the PSD module. This is specified in PSDsoft as “PSD Chip Select Input, $\overline{CSI}$”.

Port D also supports the Fast Slew Rate output drive type option using the csiop Drive Select registers.

Figure 87. Port D structure



Note: 1 Optional function on a specific Port D pin.