Welcome to E-XFL.COM

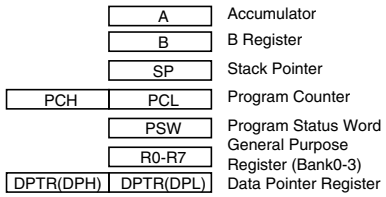
What is "Embedded - Microcontrollers"?

"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, UART/USART
Peripherals	LVR, POR, PWM, WDT
Number of I/O	46
Program Memory Size	160KB (160K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 4x8b
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/upsd3233b-40u6

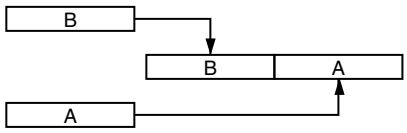
Figure 5. 8032 MCU registers



2.2.1 Accumulator

The Accumulator is the 8-bit general purpose register, used for data operation such as transfer, temporary saving, and conditional tests. The Accumulator can be used as a 16-bit register with B Register as shown below.

Figure 6. Configuration of BA 16-bit registers



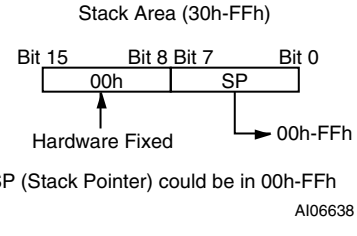
2.2.2 B register

The B Register is the 8-bit general purpose register, used for an arithmetic operation such as multiply, division with Accumulator.

2.2.3 Stack pointer

The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07h after reset. This causes the stack to begin at location 08h.

Figure 7. Stack pointer



2.3 Program memory

The program memory consists of two Flash memories: the main Flash memory (1 or 2 Mbit) and the Secondary Flash memory (256 Kbit). The Flash memory can be mapped to any address space as defined by the user in the PSDsoft Tool. It can also be mapped to Data memory space during Flash memory update or programming.

After reset, the CPU begins execution from location 0000h. As shown in *Figure 9*, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003h. If External Interrupt 0 is going to be used, its service routine must begin at location 0003h. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

The interrupt service locations are spaced at 8-byte intervals: 0003h for External Interrupt 0, 000Bh for Timer 0, 0013h for External Interrupt 1, 001Bh for Timer 1 and so forth. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

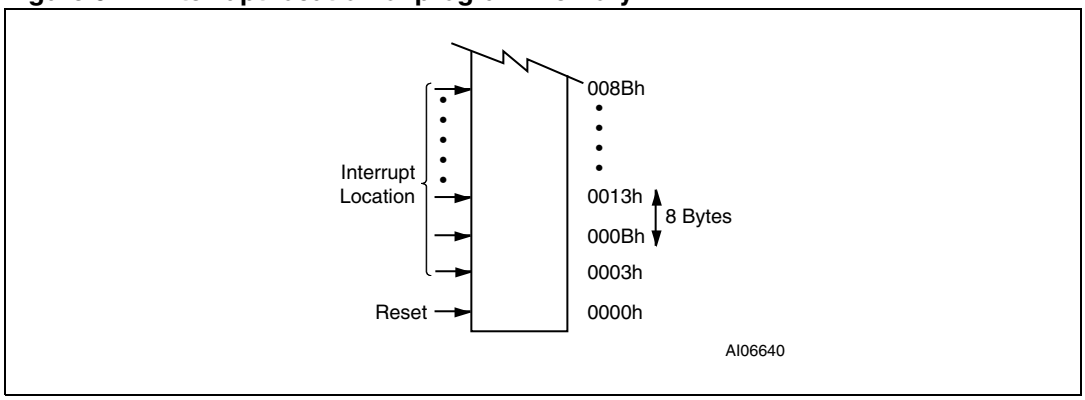
2.4 Data memory

The internal data memory is divided into four physically separated blocks: 256 bytes of internal RAM, 128 bytes of Special Function Registers (SFRs) areas, 256 bytes of external RAM (XRAM-DDC) and 8 Kbytes (XRAM-PSD) in the PSD module.

2.5 RAM

Four register banks, each 8 registers wide, occupy locations 0 through 31 in the lower RAM area. Only one of these banks may be enabled at a time. The next 16 bytes, locations 32 through 47, contain 128 directly addressable bit locations. The stack depth is only limited by the available internal RAM space of 256 bytes.

Figure 9. Interrupt location of program memory



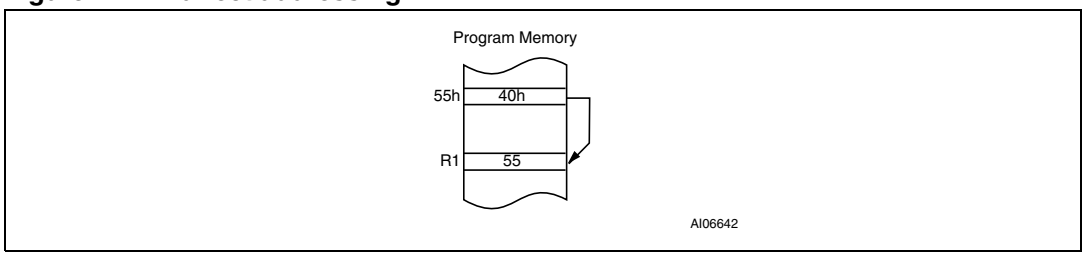
2.9.2 Indirect addressing

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed. The address register for 8-bit addresses can be R0 or R1 of the selected register bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit “data pointer” register, DPTR.

Example:

```
mov @R1, #40 H ;[R1] <-----40H
```

Figure 11. Indirect addressing



2.9.3 Register addressing

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of four banks is selected at execution time by the two bank select bits in the PSW.

Example:

```
mov PSW, #0001000B ; select Bank0
mov A, #30H
mov R1, A
```

2.9.4 Register-specific addressing

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point it. The opcode itself does that.

2.9.5 Immediate constants addressing

The value of a constant can follow the opcode in Program memory.

Example:

```
mov A, #10H.
```

2.9.6 Indexed addressing

Only Program memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program memory. A 16-bit base register (either DPTR or PC) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program memory is formed by adding the Accumulator data to the base pointer.

5 Interrupt system

There are interrupt requests from 10 sources as follows.

- INT0 external interrupt
- 2nd USART interrupt
- Timer 0 interrupt
- I^2C interrupt
- INT1 external interrupt (or ADC interrupt)
- DDC interrupt
- Timer 1 interrupt
- USB interrupt
- USART interrupt
- Timer 2 interrupt

5.1 External Int0 interrupt

- The INT0 can be either level-active or transition-active depending on Bit IT0 in register TCON. The flag that actually generates this interrupt is Bit IE0 in TCON.
- When an external interrupt is generated, the corresponding request flag is cleared by the hardware when the service routine is vectored to only if the interrupt was transition activated.
- If the interrupt was level activated then the interrupt request flag remains set until the requested interrupt is actually generated. Then it has to deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

5.2 Timer 0 and 1 interrupts

- Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1 which are set by an overflow of their respective Timer/Counter registers (except for Timer 0 in Mode 3).
- These flags are cleared by the internal hardware when the interrupt is serviced.

5.3 Timer 2 interrupt

- Timer 2 Interrupt is generated by TF2 which is set by an overflow of Timer 2. This flag has to be cleared by the software - not by hardware.
- It is also generated by the T2EX signal (Timer 2 External Interrupt P1.1) which is controlled by EXEN2 and EXF2 Bits in the T2CON register.

5.4 I^2C interrupt

- The interrupt of the I^2C is generated by Bit INTR in the register S2STA.
- This flag is cleared by hardware.

Figure 15. Interrupt system

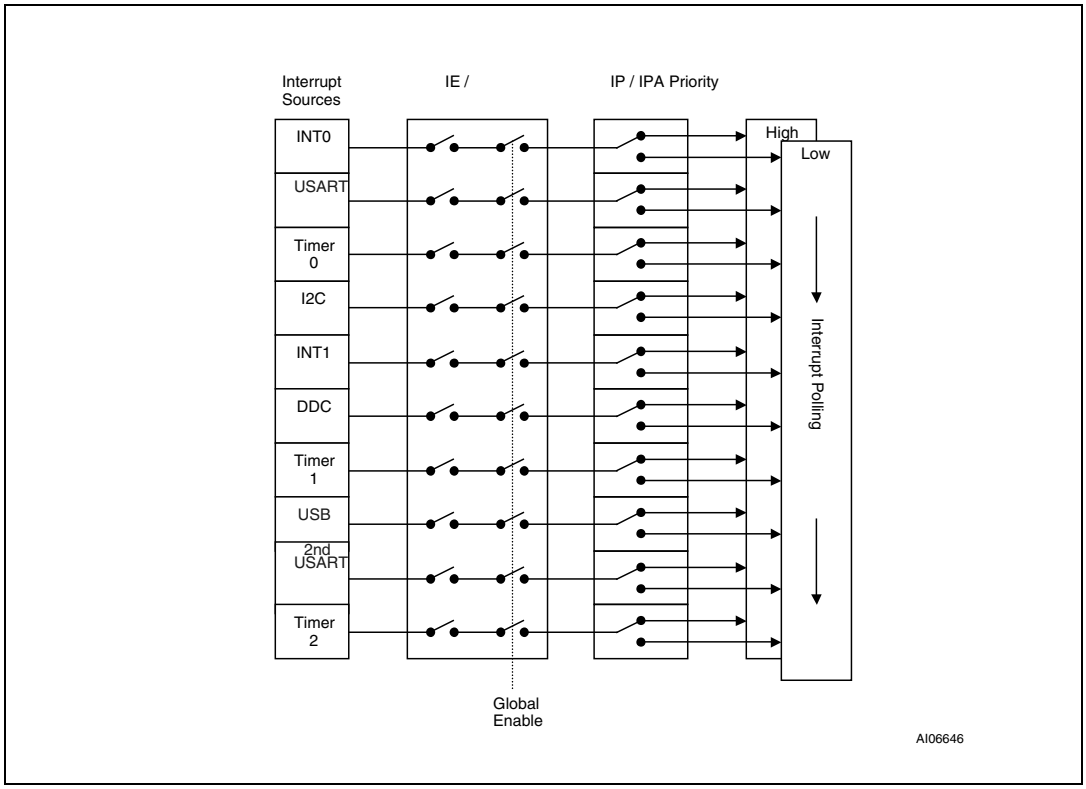


Table 18. SFR register description

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
A7	IEA	EDDC	—	—	ES2	—	—	EI^{2}C	EUSB	00	Interrupt Enable (2nd)
A8	IE	EA	—	ET2	ES	ET1	EX1	ET0	EX0	00	Interrupt Enable
B7	IPA	PDDC	—	—	PS2	—	—	PI^{2}C	PUSB	00	Interrupt Priority (2nd)
B8	IP	—	—	PT2	PS	PT1	PX1	PT0	PX0	00	Interrupt Priority

5.9 Interrupt priority structure

Each interrupt source can be assigned one of two priority levels. Interrupt priority levels are defined by the interrupt priority special function register IP and IPA.

0 = low priority

1 = high priority

Table 27. Description of the PCON bits

Bit	Symbol	Function
7	SMOD	Double baud data rate bit UART
6	SMOD1	Double baud data rate bit 2nd UART
5	LVREN	LVR disable bit (active High)
4	ADSFINT	Enable ADC Interrupt
3	RCLK1(1)	Received clock flag (UART 2)
2	TCLK1(1)	Transmit clock flag (UART 2)
1	PD	Activate Power-down mode (High enable)
0	IDL	Activate Idle mode (High enable)

1. See the T2CON register for details of the flag description

6.4 Idle mode

The instruction that sets PCON.0 is the last instruction executed in the normal operating mode before Idle mode is activated. Once in the Idle mode, the CPU status is preserved in its entirety: Stack pointer, Program counter, Program status word, Accumulator, RAM and All other registers maintain their data during Idle mode.

There are three ways to terminate the Idle mode.

1. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware terminating Idle mode. The interrupt is serviced, and following return from interrupt instruction RETI, the next instruction to be executed will be the one which follows the instruction that wrote a logic '1' to PCON.0.
2. External hardware reset: the hardware reset is required to be active for two machine cycle to complete the RESET operation.
3. Internal reset: the microcontroller restarts after 3 machine cycles in all cases.

6.5 Power-down mode

The instruction that sets PCON.1 is the last executed prior to going into the Power-down mode. Once in Power-down mode, the oscillator is stopped. The contents of the on-chip RAM and the Special Function Register are preserved.

The Power-down mode can be terminated by an external RESET.

16.1.1 DDCDAT register

DDC1 DATA register for transmission (DDCDAT: 0D5h)

- 8-bit READ and WRITE register
- Indicates DATA BYTE to be transmitted in DDC1 protocol

16.1.2 DDCADR register

Address pointer for DDC interface (DDCADR: 0D6h)

- 8-bit READ and WRITE register.
- Address pointer with the capability of the post increment. After each access to RAMBUF register (either by software or by hardware DDC1 interface), the content of this register will be increased by one. It's available both in DDC1, DDC2 (DDC2B, DDC2B+, and DDC2AB) and system operation.

Table 60. DDC SFR memory map

SFR Addr	Reg Name	Bit Register Name								Reset Value	Comments
		7	6	5	4	3	2	1	0		
D4	RAMBUF									XX	DDC Ram Buffer
D5	DDCDAT									00	DDC Data xmit register
D6	DDCADR									00	Addr pointer register
D7	DDCCON	—	EX_DAT	SWENB	DDC_AX	DDCINT	DDC1EN	SWHINT	M0	00	DDC Control Register

Table 61. Description of the DDCON register bits

Bit	Symbol	Function
7	—	Reserved
6	EX_DAT	0 = The SRAM has 128 bytes (Default) 1 = The SRAM has 256 bytes
5	SWENB	**Note:** This bit is valid for DDC1 & DDC2b modes 0 = Data is automatically read from SRAM at the current location of DDCADR and sent out via current DDC protocol. (Default) 1 = MCU is interrupted during the current data byte transmission period to load the next byte of data to send out.
4	DDC_AX	**Note:** This bit is valid for DDC1 & DDC2b modes 0 = Data is automatically read from SRAM at the current location of DDCADR and sent out via current DDC protocol. (Default) 1 = MCU is interrupted during the current data byte transmission period to load the next byte of data to send out. This bit only affects DDC2b mode Operation: 0 = DDC2b I2C Address is A0/A1 (default) 1 = DDC2b I2C Address is AX. Least 3 significant address bits are ignored.

18 PSD module

The PSD module provides configurable Program and Data memories to the 8032 CPU core (MCU). In addition, it has its own set of I/O ports and a PLD with 16 macrocells for general logic implementation.

Ports A,B,C, and D are general purpose programmable I/O ports that have a port architecture which is different from the I/O ports in the MCU module.

The PSD module communicates with the MCU module through the internal address, data bus (A0-A15, D0-D7) and control signals ($\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE, $\overline{RESET}$). The user defines the Decoding PLD in the PSDsoft Development Tool and can map the resources in the PSD module to any program or data address space. *Figure 49* shows the functional blocks in the PSD module.

18.1 Functional overview

- 1 or 2 Mbit Flash memory. This is the main Flash memory. It is divided into eight equal-sized blocks that can be accessed with user-specified addresses.
- Secondary 256 Kbit Flash boot memory. It is divided into four equal-sized blocks thatat can be accessed with user-specified addresses. This secondary memory brings the ability to execute code and update the main Flash *concurrently.*
- 64 Kbit SRAM.
- CPLD with 16 Output Micro Cells (OMCs} and 20 Input Micro Cells (IMCs). The CPLD may be used to efficiently implement a variety of logic functions for internal and external control. Examples include state machines, loadable shift registers, and loadable counters.
- Decode PLD (DPLD) that decodes address for selection of memory blocks in the PSD module.
- Configurable I/O ports (Port A,B,C and D) that can be used for the following functions:
  - MCU I/Os
  - PLD I/Os
  - Latched MCU address output
  - Special function I/Os
  - I/O ports may be configured as open-drain outputs
- Built-in JTAG compliant serial port allows full-chip In-System Programmability (ISP). With it, you can program a blank device or reprogram a device in the factory or the field.
- Internal page register that can be used to expand the 8032 MCU module address space by a factor of 256.
- Internal programmable Power Management Unit (PMU) that supports a low-power mode called Power-down mode. The PMU can automatically detect a lack of the 8032 CPU core activity and put the PSD module into Power-down mode.
- Erase/WRITE cycles:

  Flash memory - 100,000 minimum

  PLD - 1,000 minimum

  Data Retention: 15 year minimum (for Main Flash memory, Boot, PLD and Configuration bits)

Typically, the MCU can read Flash memory using READ operations, just as it would read a ROM device. However, Flash memory can only be altered using specific Erase and Program instructions. For example, the MCU cannot write a single byte directly to Flash memory as it would write a byte to RAM. To program a byte into Flash memory, the MCU must execute a Program instruction, then test the status of the Program cycle. This status test is achieved by a READ operation or polling Ready/$\overline{\text{Busy}}$ (PC3).

22.3 Instructions

An instruction consists of a sequence of specific operations. Each received byte is sequentially decoded by the PSD module and not executed as a standard WRITE operation. The instruction is executed when the correct number of bytes are properly received and the time between two consecutive bytes is shorter than the time-out period. Some instructions are structured to include READ operations after the initial WRITE operations.

The instruction must be followed exactly. Any invalid combination of instruction bytes or time-out between two consecutive bytes while addressing Flash memory resets the device logic into READ mode (Flash memory is read like a ROM device).

The Flash memory supports the instructions summarized in *Table 85*:

- Erase memory by chip or sector
- Suspend or resume sector erase
- Program a Byte
- RESET to READ mode
- Read Sector Protection Status
- Bypass

These instructions are detailed in *Table 85*. For efficient decoding of the instructions, the first two bytes of an instruction are the coded cycles and are followed by an instruction byte or confirmation byte. The coded cycles consist of writing the data AAh to address X555h during the first cycle and data 55h to address XAAAh during the second cycle. Address signals A15-A12 are Don't Care during the instruction WRITE cycles. However, the appropriate Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) must be selected.

The primary and secondary Flash memories have the same instruction set (except for Read Primary Flash Identifier). The Sector Select signals determine which Flash memory is to receive and execute the instruction. The primary Flash memory is selected if any one of Sector Select (FS0-FS7) is High, and the secondary Flash memory is selected if any one of Sector Select (CSBOOT0-CSBOOT3) is High.

Table 85. Instructions

Instruction	FS0-FS7 or CSBOOT0-CSBOOT3	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7
READ(5)	1	"Read" RD @ RA						
READ Sector Protection(6,8,13)	1	AAh@ X555h	55h@ XAAAh	90h@ X555h	Read status @ XX02h			

To exit the Unlock Bypass mode, the system must issue the two-cycle Unlock Bypass Reset Flash instruction. The first cycle must contain the data 90h; the second cycle the data 00h. Addresses are Don't Care for both cycles. The Flash memory then returns to READ mode.

22.7 Erasing Flash memory

22.7.1 Flash Bulk Erase

The Flash Bulk Erase instruction uses six WRITE operations followed by a READ operation of the status register, as described in *Table 85*. If any byte of the Bulk Erase instruction is wrong, the Bulk Erase instruction aborts and the device is reset to the READ Flash memory status.

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), as detailed in *Section 22.6: Programming Flash memory*. The Error Flag bit (DQ5) returns a '1' if there has been an Erase Failure (maximum number of Erase cycles have been executed).

It is not necessary to program the memory with 00h because the PSD module automatically does this before erasing to 0FFh.

During execution of the Bulk Erase instruction, the Flash memory does not accept any instructions.

22.7.2 Flash Sector Erase

The Sector Erase instruction uses six WRITE operations, as described in *Table 85*. Additional Flash Sector Erase codes and Flash memory sector addresses can be written subsequently to erase other Flash memory sectors in parallel, without further coded cycles, if the additional bytes are transmitted in a shorter time than the time-out period of about 100µs. The input of a new Sector Erase code restarts the time-out period.

The status of the internal timer can be monitored through the level of the Erase Time-out Flag bit (DQ3). If the Erase Time-out Flag bit (DQ3) is '0,' the Sector Erase instruction has been received and the time-out period is counting. If the Erase Time-out Flag bit (DQ3) is '1,' the time-out period has expired and the embedded algorithm is busy erasing the Flash memory sector(s). Before and during Erase time-out, any instruction other than Suspend Sector Erase and Resume Sector Erase instructions abort the cycle that is currently in progress, and reset the device to READ mode.

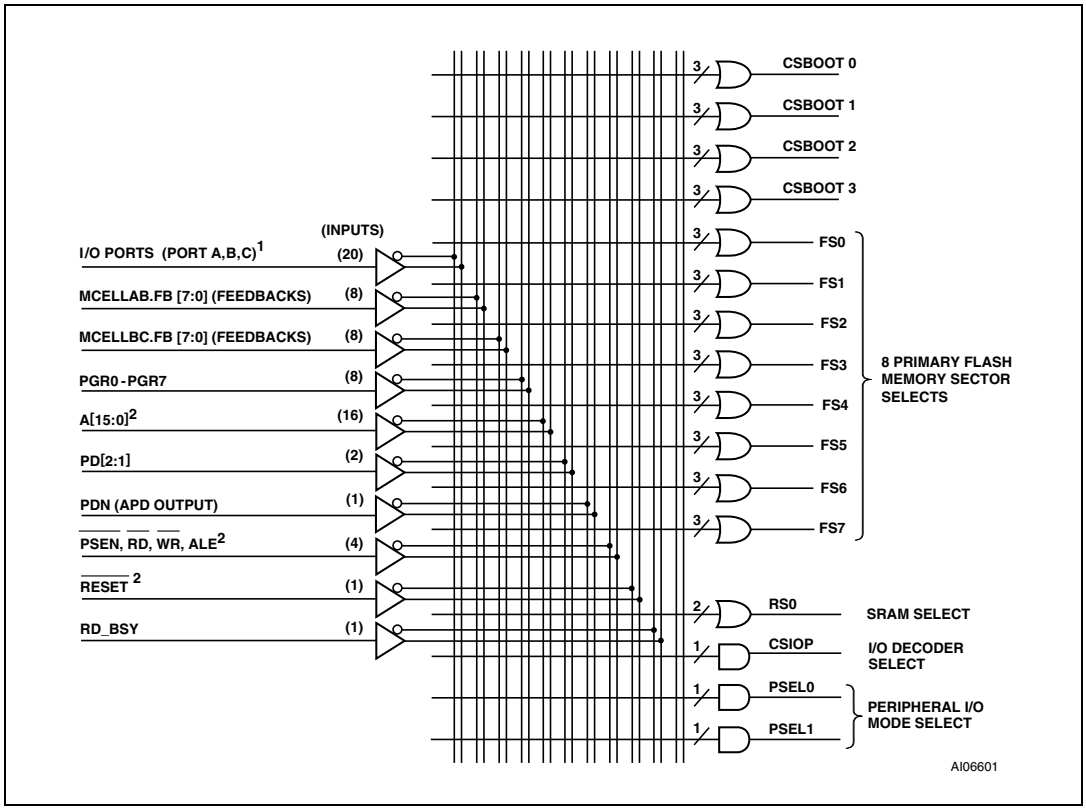
During a Sector Erase, the memory status may be checked by reading the Error Flag bit (DQ5), the Toggle Flag bit (DQ6), and the Data Polling Flag bit (DQ7), as detailed in *Section 22.6: Programming Flash memory*.

During execution of the Erase cycle, the Flash memory accepts only RESET and Suspend Sector Erase instructions. Erasure of one Flash memory sector may be suspended, in order to read data from another Flash memory sector, and then resumed.

22.7.3 Suspend Sector Erase

When a Sector Erase cycle is in progress, the Suspend Sector Erase instruction can be used to suspend the cycle by writing 0B0h to any address when an appropriate Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) is High. (See *Table 85*). This allows reading of data from another Flash memory sector after the Erase cycle has been suspended.

Table 91. DPLD logic array



1. Port A inputs are not available in the 52-pin package
2. Inputs from the MCU module

23.3 Complex PLD (CPLD)

The CPLD can be used to implement system logic functions, such as loadable counters and shift registers, system mailboxes, handshaking protocols, state machines, and random logic. The CPLD can also be used to generate External Chip Select (ECS1-ECS2), routed to Port D.

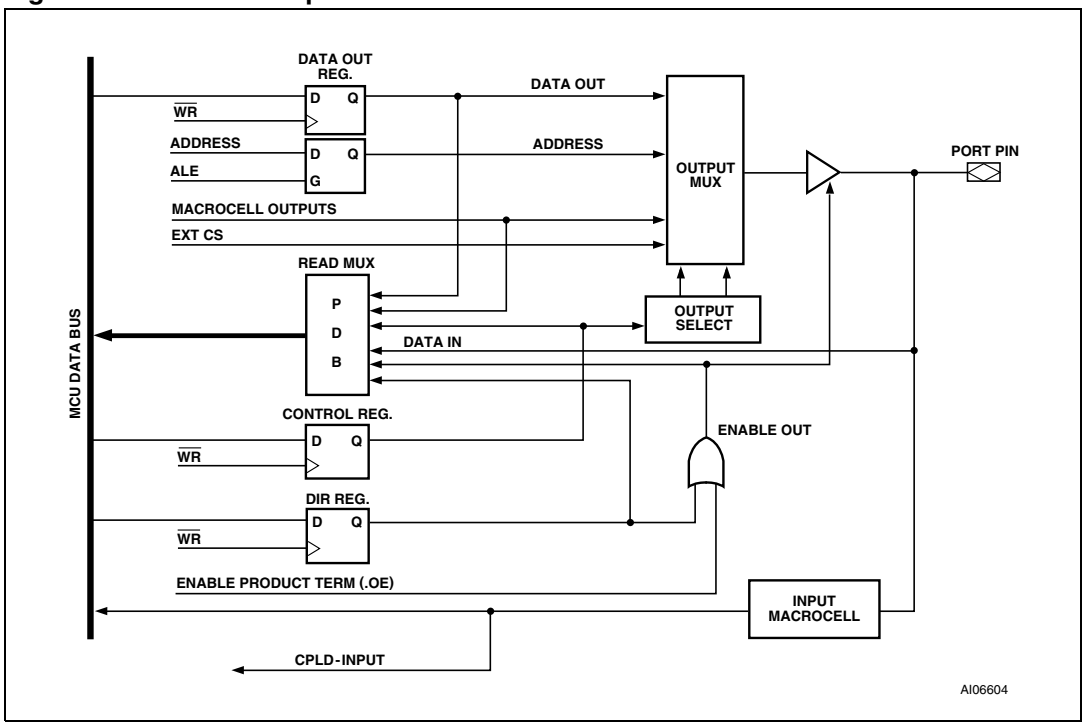
Although External Chip Select (ECS1-ECS2) can be produced by any Output Macrocell (OMC), these External Chip Select (ECS1-ECS2) on Port D do not consume any Output Macrocells (OMC).

As shown in *Figure 58*, the CPLD has the following blocks:

- 20 Input Macrocells (IMC)
- 16 Output Macrocells (OMC)
- Macrocell Allocator
- Product Term Allocator
- AND Array capable of generating up to 137 product terms
- Four I/O Ports.

Each of the blocks are described in the sections that follow.

Figure 61. General I/O port architecture



The Port pin's tri-state output driver enable is controlled by a two input OR gate whose inputs come from the CPLD AND Array enable product term and the Direction Register. If the enable product term of any of the Array outputs are not defined and that port pin is not defined as a CPLD output in the PSDsoft, then the Direction Register has sole control of the buffer that drives the port pin.

The contents of these registers can be altered by the MCU. The Port Data Buffer (PDB) feedback path allows the MCU to check the contents of the registers.

Ports A, B, and C have embedded Input Macrocells (IMC). The Input Macrocells (IMC) can be configured as latches, registers, or direct inputs to the PLDs. The latches and registers are clocked by Address Strobe (ALE) or a product term from the PLD AND Array. The outputs from the Input Macrocells (IMC) drive the PLD input bus and can be read by the MCU. See *Figure 60*.

24.2 Port operating modes

The I/O Ports have several modes of operation. Some modes can be defined using PSDsoft, some by the MCU writing to the Control Registers in CSIOP space, and some by both. The modes that can only be defined using PSDsoft must be programmed into the device and cannot be changed unless the device is reprogrammed. The modes that can be changed by the MCU can be done so dynamically at run-time. The PLD I/O, Data Port, Address Input, and Peripheral I/O modes are the only modes that must be defined before programming the device. All other modes can be changed by the MCU at run-time. See Application Note *AN1171* for more detail.

Table 93 summarizes which modes are available on each port. *Table 96* shows how and where the different modes are configured. Each of the port operating modes are described in the following sections.

Figure 62. Peripheral I/O mode

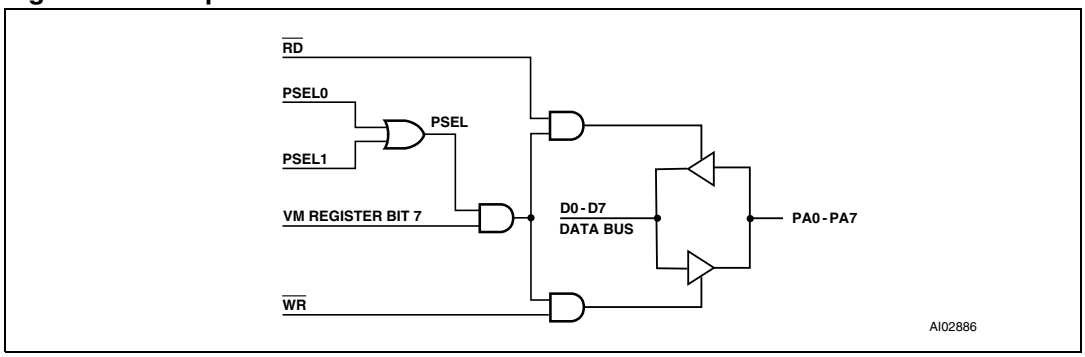


Table 93. Port operating modes

Port mode	Port A(1)	Port B	Port C	Port D
MCU I/O	Yes	Yes	Yes	Yes
PLD I/O				
McellAB Outputs	Yes	Yes	No	No
McellBC Outputs	No	Yes	Yes(2)	No
Additional Ext. CS Outputs	No	No	No	Yes
PLD Inputs	Yes	Yes	Yes	Yes
Address Out	Yes (A7 – 0)	Yes (A7 – 0)	No	No
Peripheral I/O	Yes	No	No	No
JTAG ISP	No	No	Yes(3)	No

1. Port A is not available in the 52-pin package.
2. On pins PC2, PC3, PC4, and PC7 only.
3. JTAG pins (TMS, TCK, TDI, TDO) are dedicated pins.

Table 94. Port operating mode settings

Mode	Defined in PSDsoft	Control Register Setting(1)	Direction Register Setting(1)	VM Register Setting(1)
MCU I/O	Declare pins only	0	1 = output, 0 = input (Note 2)	N/A
PLD I/O	Logic equations	N/A	(Note 2)	N/A
Address Out (Port A,B)	Declare pins only	1	1 (Note 2)	N/A
Peripheral I/O (Port A)	Logic equations (PSEL0 & 1)	N/A	N/A	PIO Bit = 1

1. N/A = Not Applicable
2. The direction of the Port A,B,C, and D pins are controlled by the Direction Register ORed with the individual output enable product term (.oe) from the CPLD AND Array.

Table 95. I/O port latched address output assignments

Port A (PA3-PA0)	Port A (PA7-PA4)	Port B (PB3-PB0)	Port B (PB7-PB4)
Address a3-a0	Address a7-a4	Address a3-a0	Address a7-a4

are not set, writing to the macrocell loads data to the macrocell flip-flops. See *Section 23: PLDs*.

24.9.4 OMC mask register

Each OMC Mask Register Bit corresponds to an Output Macrocell (OMC) flip-flop. When the OMC Mask Register Bit is set to a '1,' loading data into the Output Macrocell (OMC) flip-flop is blocked. The default value is '0' or unblocked.

24.9.5 Input macrocells (IMC)

The Input Macrocells (IMC) can be used to latch or store external inputs. The outputs of the Input Macrocells (IMC) are routed to the PLD input bus, and can be read by the MCU. See *Section 23: PLDs*.

24.9.6 Enable out

The Enable Out register can be read by the MCU. It contains the output enable values for a given port. A '1' indicates the driver is in output mode. A '0' indicates the driver is in tri-state and the pin is in input mode.

Table 100. Drive register pin assignment

Drive Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Port A	Open Drain	Open Drain	Open Drain	Open Drain	Slew Rate	Slew Rate	Slew Rate	Slew Rate
Port B	Open Drain	Open Drain	Open Drain	Open Drain	Slew Rate	Slew Rate	Slew Rate	Slew Rate
Port C	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain	Open Drain
Port D	NA(1)	NA(1)	NA(1)	NA(1)	NA(1)	Slew Rate	Slew Rate	NA(1)

1. NA = Not Applicable.

Table 101. Port data registers

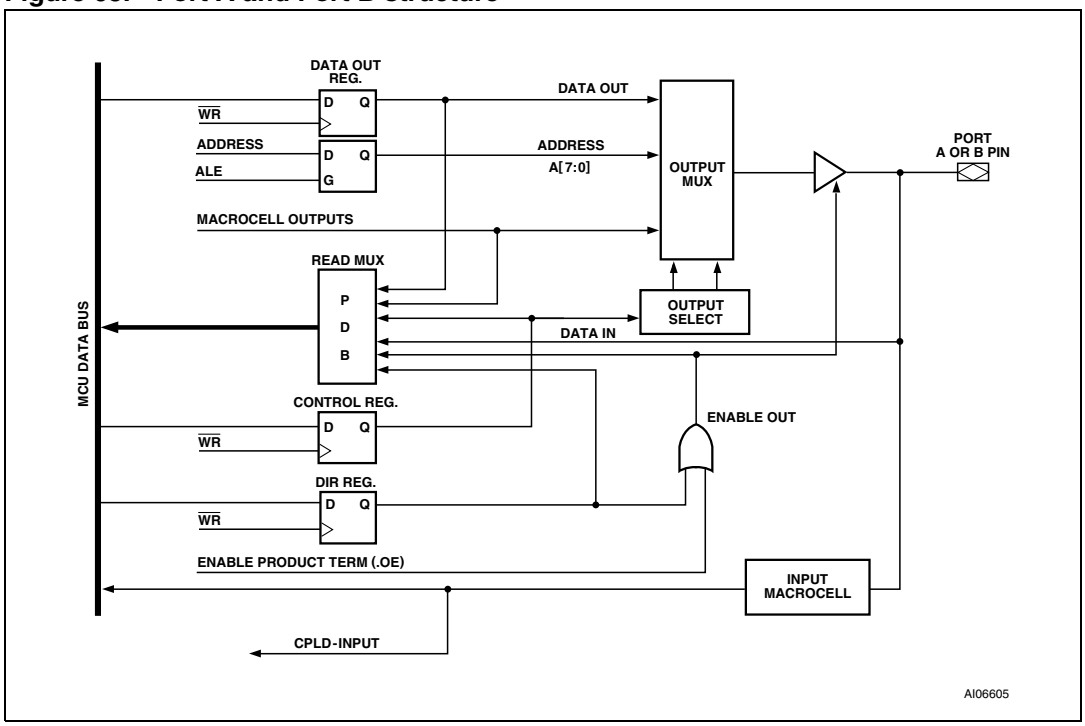
Register Name	Port	MCU Access
Data In	A,B,C,D	READ – input on pin
Data Out	A,B,C,D	WRITE/READ
Output Macrocell	A,B,C	READ – outputs of macrocells WRITE – loading macrocells flip-flop
Mask Macrocell	A,B,C	WRITE/READ – prevents loading into a given macrocell
Input Macrocell	A,B,C	READ – outputs of the Input Macrocells
Enable Out	A,B,C	READ – the output enable control of the port driver

24.10 Ports A and B – functionality and structure

Ports A and B have similar functionality and structure, as shown in *Figure 63*. The two ports can be configured to perform one or more of the following functions:

- MCU I/O mode
- CPLD Output – Macrocells McellAB7-McellAB0 can be connected to Port A or Port B. McellBC7-McellBC0 can be connected to Port B or Port C.
- CPLD Input – Via the Input Macrocells (IMC).
- Latched Address output – Provide latched address output as per *Table 95*.
- Open Drain/Slew Rate – pins PA3-PA0 and PB3-PB0 can be configured to fast slew rate, pins PA7-PA4 and PB7-PB4 can be configured to Open Drain mode.
- Peripheral mode – Port A only (80-pin package)

Figure 63. Port A and Port B structure



24.11 Port C – functionality and structure

Port C can be configured to perform one or more of the following functions (see *Figure 64*):

- MCU I/O mode
- CPLD Output – McellBC7-McellBC0 outputs can be connected to Port B or Port C.
- CPLD Input – via the Input Macrocells (IMC)
- In-System Programming (ISP) – JTAG pins (TMS, TCK, TDI, TDO) are dedicated pins for device programming. (See *Section 27: Programming in-circuit using the JTAG serial interface*, for more information on JTAG programming.)
- Open Drain – Port C pins can be configured in Open Drain mode

Port C does not support Address Out mode, and therefore no Control Register is required.

29 AC/DC parameters

These tables describe the AD and DC parameters of the UPSD323xx 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 70* and *Figure 71* 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 70. PLD I_{CC}/frequency consumption (5 V range)

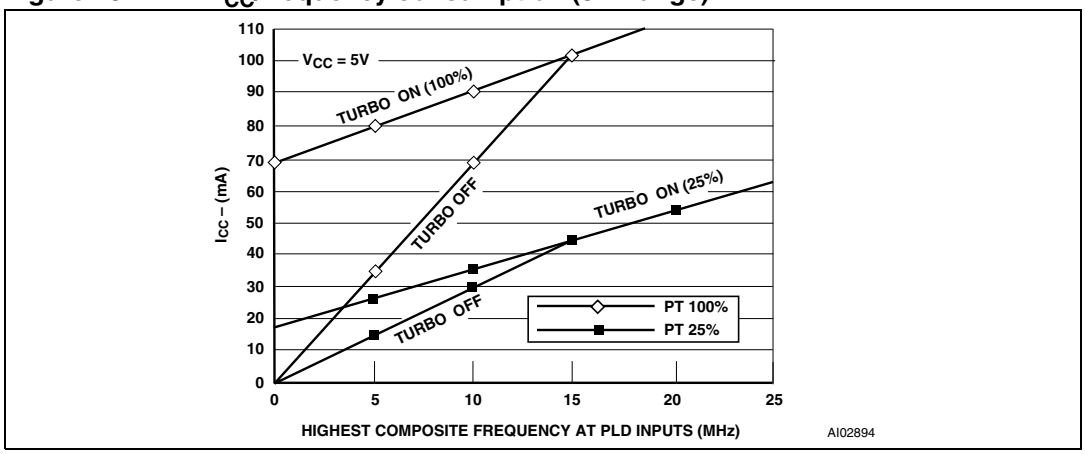


Figure 77. Synchronous clock mode timing – PLD

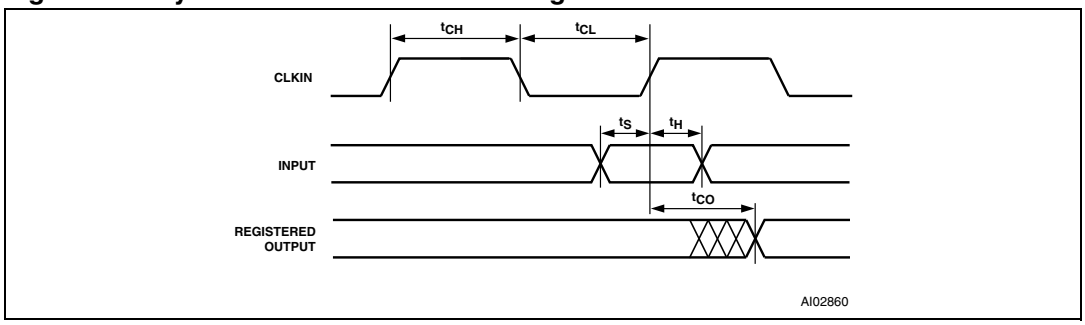


Table 129. CPLD macrocell synchronous clock mode timing (5 V devices)

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) $t_{CLCL} = t_{CH} + t_{CL}$.

Table 131. CPLD macrocell asynchronous clock mode timing (5 V devices)

Symbol	Parameter	Conditions	Min	Max	PT aloc	Turbo off	Slew rate	Unit
f_{MAXA}	Maximum frequency external feedback	$1/(t_{SA}+t_{COA})$		38.4				MHz
	Maximum frequency internal feedback (f_{CNTA})	$1/(t_{SA}+t_{COA}-10)$		62.5				MHz
	Maximum frequency pipelined data	$1/(t_{CHA}+t_{CLA})$		71.4				MHz
t_{SA}	Input setup time		7		+ 2	+ 10		ns
t_{HA}	Input hold time		8					ns
t_{CHA}	Clock input high time		9			+ 10		ns
t_{CLA}	Clock input low time		9			+ 10		ns
t_{COA}	Clock to output delay			21		+ 10	– 2	ns
t_{ARDA}	CPLD array delay	Any macrocell		11	+ 2			ns
t_{MINA}	Minimum clock period	$1/f_{CNTA}$	16					ns

Table 132. CPLD macrocell asynchronous clock mode timing (3 V devices)

Symbol	Parameter	Conditions	Min.	Max.	PT aloc	Turbo off	Slew rate	Unit
f_{MAXA}	Maximum frequency external feedback	$1/(t_{SA}+t_{COA})$		21.7				MHz
	Maximum frequency internal feedback (f_{CNTA})	$1/(t_{SA}+t_{COA}-10)$		27.8				MHz
	Maximum frequency pipelined data	$1/(t_{CHA}+t_{CLA})$		33.3				MHz
t_{SA}	Input setup time		10		+ 4	+ 20		ns
t_{HA}	Input hold time		12					ns
t_{CHA}	Clock input high time		17			+ 20		ns
t_{CLA}	Clock input low time		13			+ 20		ns
t_{COA}	Clock to output delay			36		+ 20	– 6	ns
t_{ARD}	CPLD array delay	Any macrocell		25	+ 4			ns
t_{MINA}	Minimum clock period	$1/f_{CNTA}$	36					ns

Figure 81. Peripheral I/O Read timing

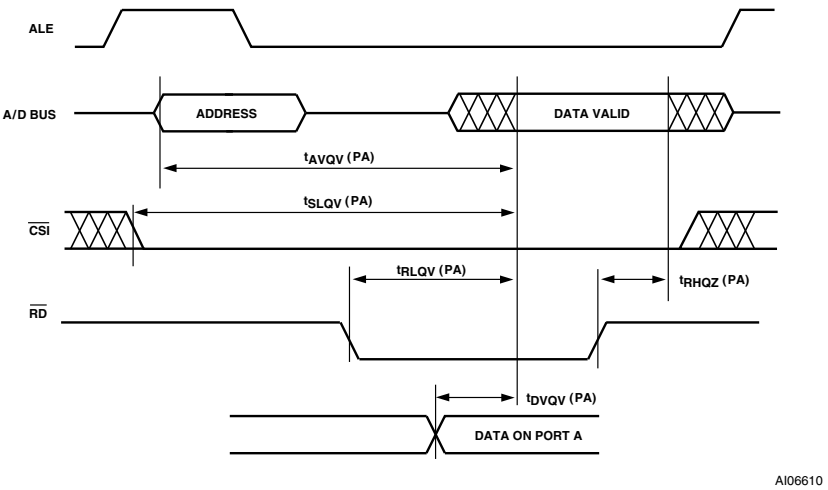


Table 137. Port A peripheral data mode Read timing (5 V devices)

Symbol	Parameter	Conditions	Min.	Max.	Turbo off	Unit
$t_{AVQV-PA}$	Address valid to data valid	(Note 1)		37	+ 10	ns
$t_{SLQV-PA}$	$\overline{CSI}$ valid to data valid			27	+ 10	ns
$t_{RLQV-PA}$	$\overline{RD}$ to data valid	(Note 2)		32		ns
$t_{DVQV-PA}$	Data in to data out valid			22		ns
$t_{RHQZ-PA}$	$\overline{RD}$ to data high-Z			23		ns

1. Any input used to select Port A Data Peripheral mode.
2. Data is already stable on Port A.

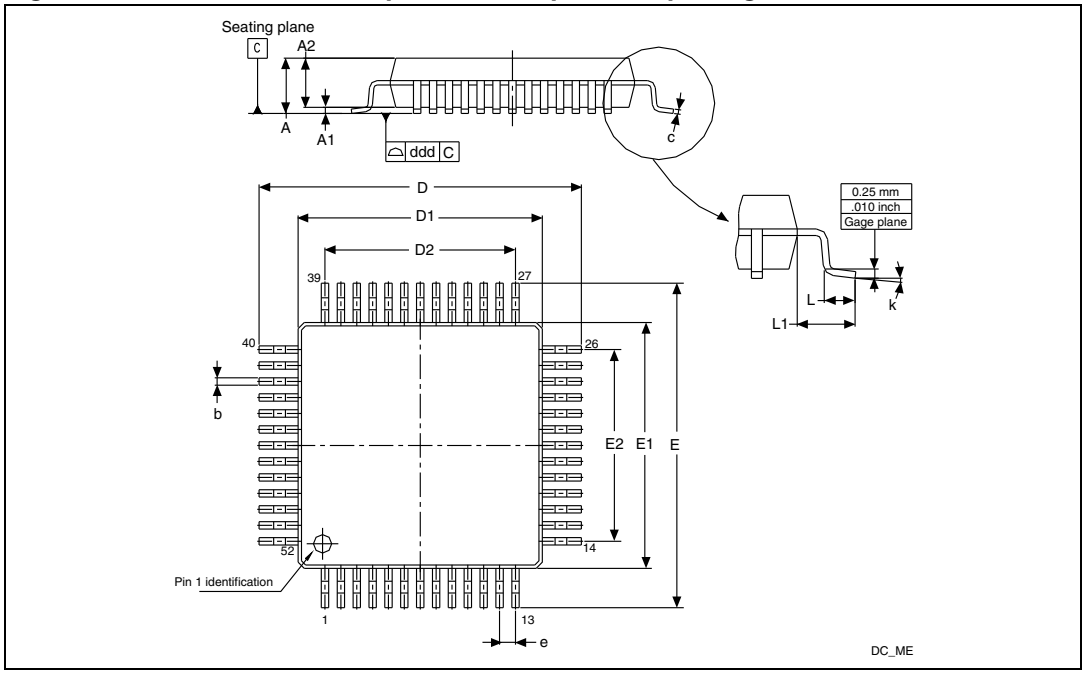
Table 138. Port A peripheral data mode Read timing (3 V devices)

Symbol	Parameter	Conditions	Min.	Max.	Turbo off	Unit
$t_{AVQV-PA}$	Address valid to data valid	(Note 1)		50	+ 20	ns
$t_{SLQV-PA}$	$\overline{CSI}$ valid to data valid			37	+ 20	ns
$t_{RLQV-PA}$	$\overline{RD}$ to data valid	(Note 2)		45		ns
$t_{DVQV-PA}$	Data in to data out valid			38		ns
$t_{RHQZ-PA}$	$\overline{RD}$ to data high-Z			36		ns

1. Any input used to select Port A Data Peripheral mode.
2. Data is already stable on Port A.

33 Package mechanical information

Figure 91. LQFP52 – 52-lead plastic thin, quad, flat package outline



1. Drawing is not to scale.

Table 146. LQFP52 – 52-lead plastic thin, quad, flat package mechanical data

Symbol	millimeters			inches[1]		
	Typ	Min	Max	Typ	Min	Max
A			1.60			0.063
A1		0.05	0.15		0.002	0.0059
A2		1.35	1.45		0.0531	0.0571
b		0.22	0.38		0.0087	0.015
C		0.09	0.2		0.0035	0.0079
D	12			0.4724		
D1	10			0.3937		
D2	7.8			0.3071		
E	12			0.4724		
E1	10			0.3937		
E2	7.8			0.3071		
e	0.65			0.0256		
L		0.45	0.75		0.0177	0.0295
L1	1			0.0394		
k		0°	7°		0°	7°
ddd	0.100			0.0039		

1. Values in inches are converted from mm and rounded to 4 decimal digits.