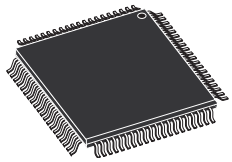
# PSD4256G6V

## Flash in-system programmable (ISP) peripherals for 8-bit or 16-bit MCUs

## Features

- Dual bank Flash memories
  - 8 Mbits of primary Flash memory (16 uniform sectors, 64 Kbytes)
  - 512 Kbits of secondary Flash memory with 4 sectors
  - Concurrent operation: read from one memory while erasing and writing the other
  - 256 kbits of SRAM
  - PLD with macrocells
  - Over 3000 gates of PLD: CPLD and DPLD
  - CPLD with 16 output macrocells (OMCs) and 24 input macrocells (IMCs)
  - DPLD - user defined internal chip select decoding
- Seven I/O ports with 52 I/O pins
  52 individually configurable I/O port pins 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
- In-system programming (ISP) with JTAG
  - Built-in JTAG compliant serial port allows full-chip in-system programmability
  - Efficient manufacturing allow easy product testing and programming
  - Use low cost FlashLINK™ cable with PC
- Page register
  - Internal page register that can be used to expand the microcontroller address space by a factor of 256



LQFP80 (U)
80-lead Thin, Quad Flat

- Programmable power management
- High endurance
  - 100,000 erase/write cycles of Flash memory
  - 1,000 erase/write cycles of PLD
  - 15 year data retention
- Single supply voltage
  - 3 V (+20%/–10%)
- Memory speed
  - 100 ns Flash memory and SRAM access time for $V_{CC}$ = 3 V (+20%/–10%)
  - 90 ns Flash memory and SRAM access time for $V_{CC}$ = 3.3 V (+/–10%)
- Packages are ECOPACK®

# Contents

# List of tables

# List of figures

# 1 Description

The PSD family of memory systems for microcontrollers (MCUs) brings in-system-programmability (ISP) to Flash memory and programmable logic. The result is a simple and flexible solution for embedded designs. PSD devices combine many of the peripheral functions found in MCU-based applications (8-bit or 16-bit), such as configurable memories, PLD logic, and I/O.

PSD devices integrate an optimized macrocell logic architecture. The macrocell was created to address the unique requirements of embedded system designs. It allows direct connection between the system address/data bus, and the internal PSD registers, to simplify communication between the MCU and other supporting devices.

The PSD family offers two methods to program the PSD Flash memory while the PSD is soldered to the circuit board: in-system programming (ISP) via JTAG, and in-application programming (IAP).

## 1.1 In-system programming (ISP) via JTAG

An IEEE 1149.1 compliant JTAG in-system programming (ISP) interface is included on the PSD enabling the entire device (Flash memories, PLD, configuration) to be rapidly programmed while soldered to the circuit board. This requires no MCU participation, which means the PSD can be programmed anytime, even when completely blank.

The innovative JTAG interface to Flash memories is an industry first, solving key problems faced by designers and manufacturing houses, such as:

### 1.1.1 First time programming

How do I get firmware into the Flash memory the very first time? JTAG is the answer. Program the blank PSD with no MCU involvement.

### 1.1.2 Inventory build-up of pre-programmed devices

How do I maintain an accurate count of pre-programmed Flash memory and PLD devices based on customer demand? How many and what version? JTAG is the answer. Build your hardware with blank PSDs soldered directly to the board and then custom program just before they are shipped to the customer. No more labels on chips, and no more wasted inventory.

### 1.1.3 Expensive sockets

How do I eliminate the need for expensive and unreliable sockets? JTAG is the answer. Solder the PSD directly to the circuit board. Program first time and subsequent times with JTAG. No need to handle devices and bend the fragile leads.

## 1.2 In-application programming

Two independent Flash memory arrays are included so that the MCU can execute code from one while erasing and programming the other. Robust product firmware updates in the filed are possible over any communication channel (e.g., CAN, Ethernet, UART, J1850) using this unique architecture. Designers are relieved of these problems.

### 1.2.1 Simultaneous read and write to Flash memory

How can the MCU program the same memory from which it executing code? It cannot. The PSD allows the MCU to operate the two Flash memory blocks concurrently, reading code from one while erasing and programming the other during IAP.

### 1.2.2 Complex memory mapping

How can I map these two memories efficiently? A programmable decode PLD (DPLD) is embedded in the PSD module. The concurrent PSD memories can be mapped anywhere in MCU address space, segment by segment with extremely high address resolution. As an option, the secondary Flash memory can be swapped out of the system memory map when IAP is complete. A built-in page register breaks the MCU address limit.

### 1.2.3 Separate program and data space

How can I write to Flash memory while it resides in program space during field firmware updates? My 80C51XA will not allow it. The PSD provides means to reclassify Flash memory as data space during IAP, then back to program space when complete.

## 1.3 PSDsoft™

PSDsoft, a software development tool from ST, guides you through the design process step-by-step making it possible to complete an embedded MCU design capable of ISP/IAP in just hours. Select your MCU and PSDsoft takes you through the remainder of the design with point and click entry, covering PSD selection, pin definitions, programmable logic inputs and outputs, MCU memory map definition, ANSI-C code generation for your MCU, and merging your MCU firmware with the PSD design. When complete, two different device programmers are supported directly from PSDsoft: FlashLINK (JTAG) and PSDpro.

**Figure 1. Logic diagram**

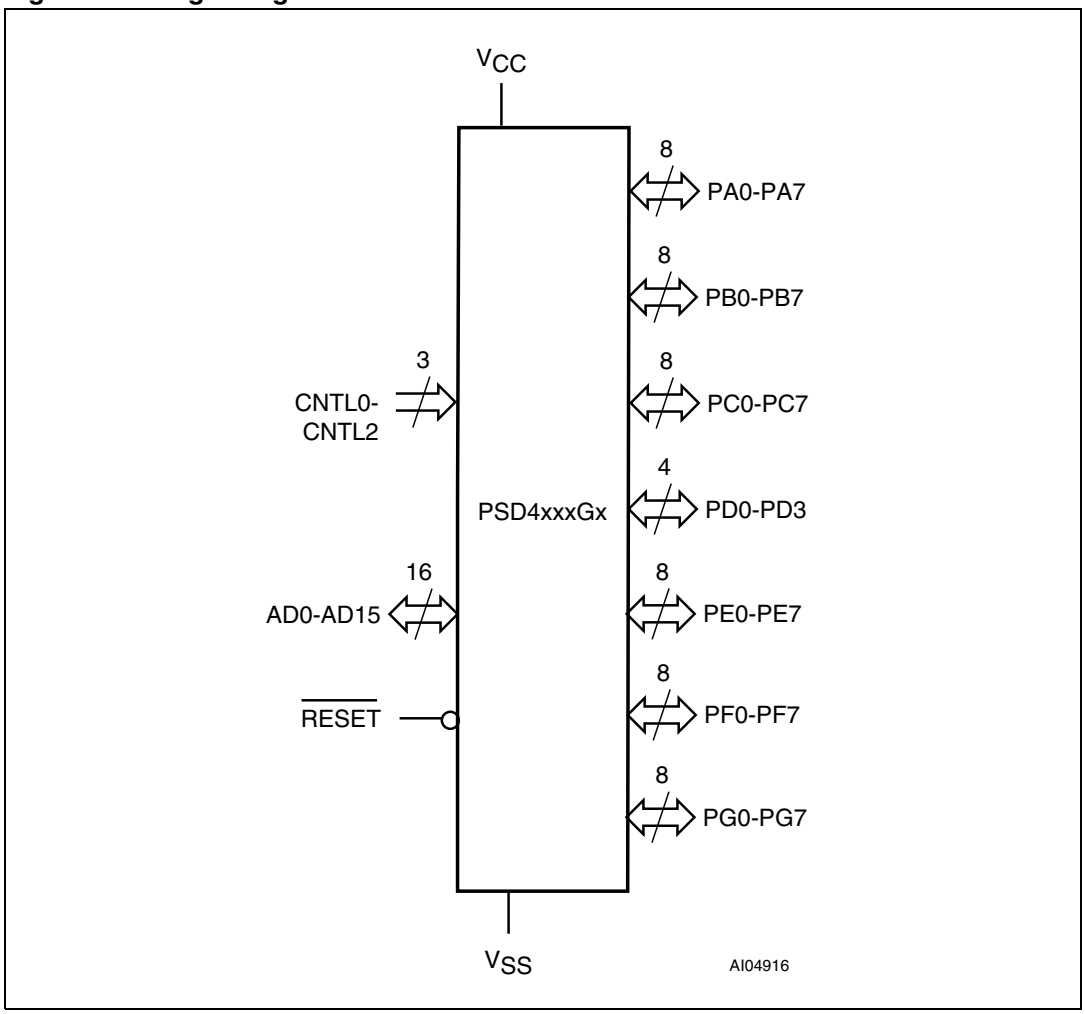



**Table 1. Pin names**

| Signal names | Function |
|---|---|
| PA0-PA7 | Port-A |
| PB0-PB7 | Port-B |
| PC0-PC7 | Port-C |
| PD0-PD3 | Port-D |
| PE0-PE7 | Port-E |
| PF0-PF7 | Port-F |
| PG0-PG7 | Port-G |
| AD0-AD15 | Address/Data |
| CNTL0-CNTL2 | Control |
| $\overline{\text{RESET}}$ | Reset |

**Table 1. Pin names (continued)**

| Signal names | Function |
|---|---|
| $V_{CC}$ | Supply voltage |
| $V_{SS}$ | Ground |

**Figure 2. LQFP80 connections**

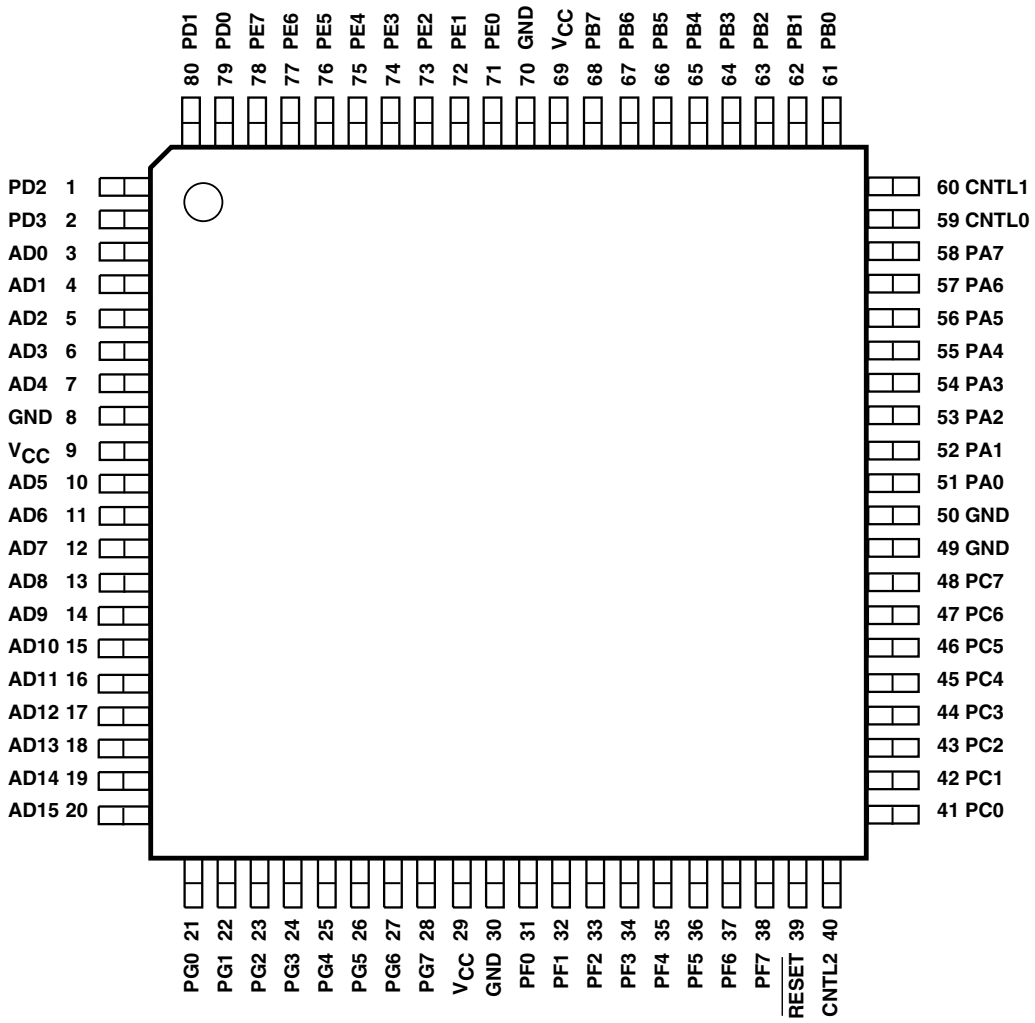


**Table 2. LQFP80 pin description**

| Pin name(1) | Pin | Type | Description |
|---|---|---|---|
| ADIO0-ADIO7 | 3-7<br>10-12 | I/O | This is the lower Address/Data port. Connect your MCU address or address/data bus according to the following rules:<br>– If your MCU has a multiplexed address/data bus where the data is multiplexed with the lower address bits, connect AD0-AD7 to this port;<br>– If your MCU does not have a multiplexed address/data bus, connect A0-A7 to this port; and<br>– If you are using an 80C51XA in burst mode, connect A4/D0 through A11/D7 to this port.<br>ALE or AS latches the address. The PSD drives data out only if the READ signal is active and one of the PSD functional blocks has been selected. The addresses on this port are passed to the PLDs. |
| ADIO8-ADIO15 | 13-20 | I/O | This is the upper Address/Data port. Connect your MCU address or address/data bus according to the following rules:<br>– If your MCU has a multiplexed address/data bus where the data is multiplexed with the address bits, connect A8-A15 or AD8-AD15 to this port;<br>– If your MCU does not have a multiplexed address/data bus, connect A8-A15 to this port; and<br>– If you are using an 80C51XA in burst mode, connect A12/D8 through A19/D15 to this port.<br>ALE or AS latches the address. The PSD drives data out only if the READ signal is active and one of the PSD functional blocks has been selected. The addresses on this port are passed to the PLDs. |
| CNTL0 | 59 | I | The following control signals can be connected to this pin, based on your MCU:<br>– $\overline{WR}$ – active Low, WRITE Strobe input<br>– R_$\overline{W}$ – active High, READ/active Low WRITE input<br>– $\overline{WRL}$ – active Low, WRITE to Low-byte<br>This pin is connected to the PLDs. Therefore, these signals can be used in decode and other logic equations. |
| CNTL1 | 60 | I | The following control signals can be connected to this pin, based on your MCU:<br>– 1$\overline{RD}$ – active Low, READ Strobe input<br>– E – E clock input<br>– $\overline{DS}$ – active Low, Data Strobe input<br>– $\overline{LDS}$ – active Low, Strobe for low data byte<br>This pin is connected to the PLDs. Therefore, these signals can be used in decode and other logic equations. |
| CNTL2 | 40 | I | READ or other Control input pin, with multiple configurations. Depending on the MCU interface selected, this pin can be:<br>– $\overline{PSEN}$ – Program Select Enable, active Low in code retrieve bus cycle (80C51XA mode);<br>– BHE – High-byte enable, 16-bit data bus;<br>– $\overline{UDS}$ – active Low, Strobe for high data byte, 16-bit data bus mode;<br>– SIZ0 – Byte enable input; or<br>– LSTRB – Low Strobe input.<br>This pin is also connected to the PLDs. |

**Table 2. LQFP80 pin description (continued)**

| Pin name(1) | Pin | Type | Description |
|---|---|---|---|
| RESET | 39 | I | Active Low input. Resets I/O ports, PLD macrocells and some of the Configuration registers and JTAG registers. Must be Low at Power-up. $\overline{\text{RESET}}$ also aborts any Flash memory program or erase cycle that is currently in progress. |
| PA0-PA7 | 51-58 | I/O CMOS or Open Drain | These pins make up Port A. These port pins are configurable and can have the following functions:<br>– MCU I/O – standard output or input port;<br>– CPLD macrocell (McellA0-McellA7) outputs; and<br>– Latched, transparent or registered PLD inputs (can also be PLD input for address A16 and above) |
| PB0-PB7 | 61-68 | I/O CMOS or Open Drain | These pins make up Port B. These port pins are configurable and can have the following functions:<br>– MCU I/O – standard output or input port;<br>– CPLD macrocell (McellB0-McellB7) outputs; and<br>– Latched, transparent or registered PLD inputs (can also be PLD input for address A16 and above). |
| PC0-PC7 | 41-48 | I/O CMOS | These pins make up Port C. These port pins are configurable and can have the following functions:<br>– MCU I/O – standard output or input port;<br>– External Chip Select (ECS0-ECS7) outputs; and<br>– Latched, transparent or registered PLD inputs (can also be PLD input for address A16 and above). |
| PD0 | 79 | I/O CMOS or Open Drain | PD0 pin of Port D. This port pin can be configured to have the following functions:<br>– ALE/AS input – latches address on ADIO0-ADIO15;<br>– $\overline{\text{AS}}$ input – latches address on ADIO0-ADIO15 on the rising edge;<br>– MCU I/O – standard output or input port; and<br>– Transparent PLD input (can also be PLD input for address A16 and above). |
| PD1 | 80 | I/O CMOS or Open Drain | PD1 pin of Port D. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Transparent PLD input (can also be PLD input for address A16 and above); and<br>– CLKIN – clock input to the CPLD macrocell, the APD Unit's Power-down counter, and the CPLD AND Array. |
| PD2 | 1 | I/O CMOS or Open Drain | PD2 pin of Port D. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Transparent PLD input (can also be PLD input for address A16 and above); and<br>– PSD Chip Select Input ($\overline{\text{CSI}}$). When Low, the MCU can access the PSD memory and I/O. When High, the PSD memory blocks are disabled to conserve power. The falling edge of this signal can be used to get the device out of Power-down mode. |
| PD3 | 2 | I/O CMOS or Open Drain | PD3 pin of Port D. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Transparent PLD input (can also be PLD input for address A16 and above); and<br>– $\overline{\text{WRH}}$ – for 16-bit data bus, WRITE to high byte, active low. |

**Table 2. LQFP80 pin description (continued)**

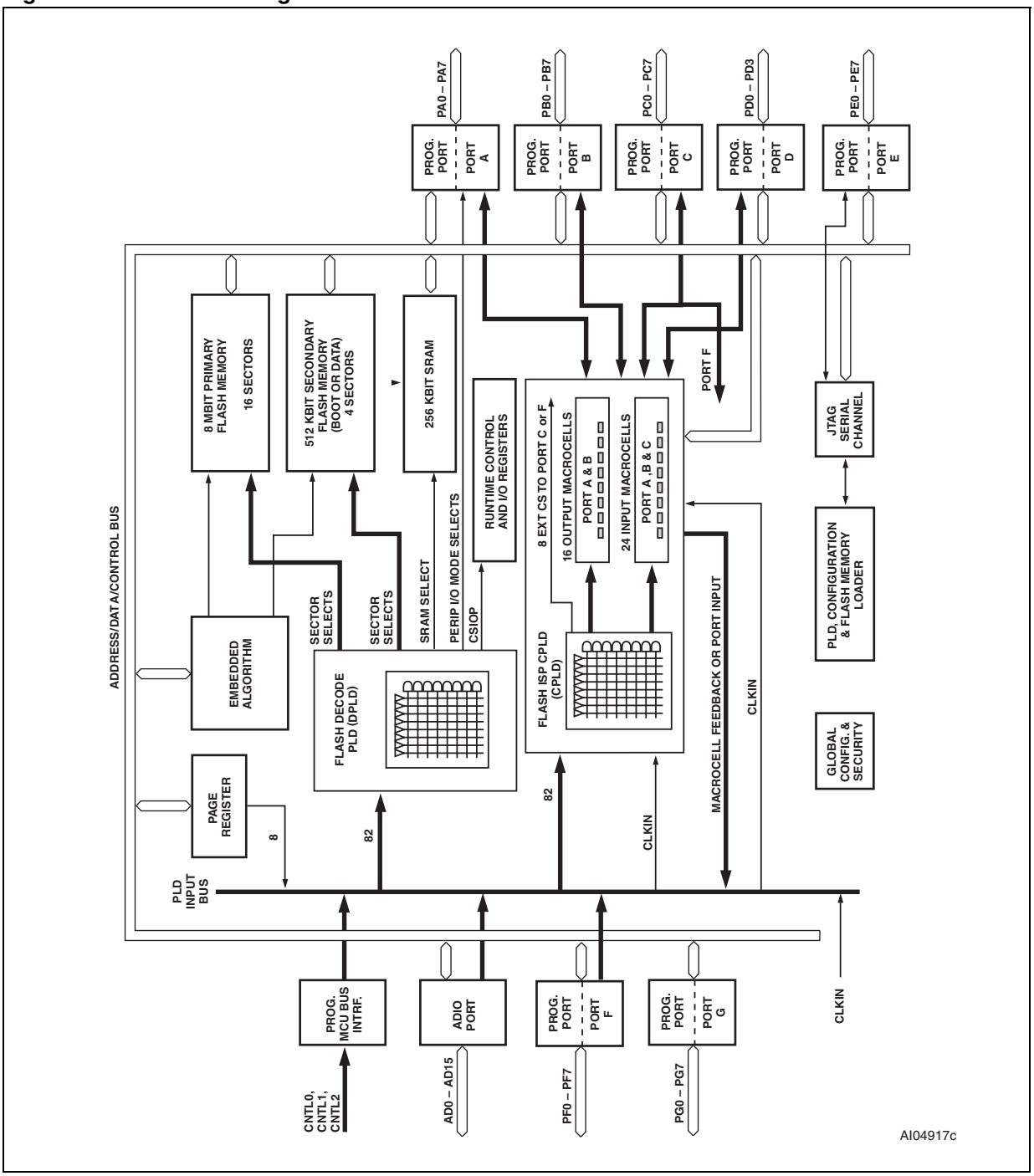
| Pin name(1) | Pin | Type | Description |
|---|---|---|---|
| PE0 | 71 | I/O CMOS or Open Drain | PE0 pin of Port E. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address output; and<br>– TMS Input for the JTAG Serial Interface. |
| PE1 | 72 | I/O CMOS or Open Drain | PE1 pin of Port E. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address output; and<br>– TCK Input for the JTAG Serial Interface. |
| PE2 | 73 | I/O CMOS or Open Drain | PE2 pin of Port E. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address output; and<br>– TDI input for the JTAG Serial Interface. |
| PE3 | 74 | I/O CMOS or Open Drain | PE3 pin of Port E. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address output; and<br>– TDO output for the JTAG Serial Interface. |
| PE4 | 75 | I/O CMOS or Open Drain | PE4 pin of Port E. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address output:<br>– TSTAT output for the JTAG Serial Interface; and<br>– Ready/$\overline{\text{Busy}}$ output for parallel in-system programming (ISP). |
| PE5 | 76 | I/O CMOS or Open Drain | PE5 pin of Port E. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address output; and<br>– $\overline{\text{TERR}}$ active Low output for the JTAG Serial Interface. |
| PE6 | 77 | I/O CMOS or Open Drain | PE6 pin of Port E. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address output. |
| PE7 | 78 | I/O CMOS or Open Drain | PE7 pin of Port E. This port pin can be configured to have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address output. |

**Table 2. LQFP80 pin description (continued)**

| Pin name(1) | Pin | Type | Description |
|---|---|---|---|
| PF0-PF7 | 31-38 | I/O CMOS or Open Drain | These pins make up Port F. These port pins are configurable and can have the following functions:<br>– MCU I/O – standard output or input port;<br>– External Chip Select (ECS0-ECS7) outputs, or inputs to CPLD;<br>– Latched address outputs;<br>– Address A1-A3 inputs in 80C51XA mode (PF0 is grounded);<br>– Data bus port (D0-D7) in a non-multiplexed bus configuration;<br>– Peripheral I/O mode; and<br>– MCU RESET mode. |
| PG0-PG7 | 21-28 | I/O CMOS or Open Drain | These pins make up Port G. These port pins are configurable and can have the following functions:<br>– MCU I/O – standard output or input port;<br>– Latched address outputs;<br>– Data bus port (D8-D15) in a non-multiplexed, 16-bit bus configuration; and<br>– MCU RESET mode. |
| $V_{CC}$ | 9, 29, 69 | | Supply voltage |
| GND | 8, 30, 49, 50, 70 | | Ground pins |

1. Signal names that have multiple names or functions are defined using PSDsoft.

**Figure 3. PSD block diagram**




1. Additional address lines can be brought in to the device via Port A, B, C, D, or F.

# 2 PSD architectural overview

PSD devices contain several major functional blocks. *Figure 3: PSD block diagram* shows the architecture of the PSD device family. The functions of each block are described briefly in the following sections. Many of the blocks perform multiple functions and are user configurable.

## 2.1 Memory

Each of the memory blocks is briefly discussed in the following paragraphs. A more detailed discussion can be found in *Section 6.1: Memory blocks*.

The 8 Mbit primary Flash memory is the main memory of the PSD. It is divided into 16 equally-sized sectors that are individually selectable.

The 512 Kbit secondary Flash memory is divided into 4 sectors. Each sector is individually selectable.

The 256 Kbit SRAM is intended for use as a scratch-pad memory or as an extension to the MCU SRAM.

Each memory block can be located in a different address space as defined by the user. The access times for all memory types includes the address latching and DPLD decoding time.

## 2.2 PLDs

The device contains two PLD blocks, the Decode PLD (DPLD) and the Complex PLD (CPLD), as shown in *Figure 3: PSD block diagram*, each optimized for a different function. The functional partitioning of the PLDs reduces power consumption, optimizes cost/performance, and eases design entry.

The DPLD is used to decode addresses and to generate Sector Select signals for the PSD internal memory and registers. The DPLD has combinatorial outputs, while the CPLD can implement more general user-defined logic functions. The CPLD has 16 output macrocells (OMC) and 8 combinatorial outputs. The PSD also has 24 input macrocells (IMC) that can be configured as inputs to the PLDs. The PLDs receive their inputs from the PLD Input Bus and are differentiated by their output destinations, number of product terms, and macrocells.

The PLDs consume minimal power. The speed and power consumption of the PLD is controlled by the Turbo bit in PMMR0 and other bits in PMMR2. These registers are set by the MCU at run-time. There is a slight penalty to PLD propagation time when not in the Turbo mode.

## 2.3 I/O ports

The PSD has 52 I/O pins divided among seven ports (Port A, B, C, D, E, F, and G). Each I/O pin can be individually configured for different functions. Ports can be configured as standard MCU I/O ports, PLD I/O, or latched address outputs for MCUs using multiplexed address/data buses.

The JTAG pins can be enabled on Port E for in-system programming (ISP).

## 2.4 MCU bus interface

The PSD easily interfaces with most 8-bit or 16-bit MCUs, either with multiplexed or non-multiplexed address/data buses. The device is configured to respond to the MCU's control pins, which are also used as inputs to the PLDs.

## 2.5 ISP via JTAG port

In-system programming (ISP) can be performed through the JTAG signals on Port E. This serial interface allows complete programming of the entire PSD module device. A blank device can be completely programmed. The JTAG signals (TMS, TCK, TSTAT, $\overline{\text{TERR}}$, TDI, TDO) can be multiplexed with other functions on Port E. *Table 4: JTAG signals on port E* indicates the JTAG pin assignments.

**Table 3. PLD I/O**

| Name | Inputs | Outputs | Product terms |
|---|---|---|---|
| Decode PLD (DPLD) | 82 | 17 | 43 |
| Complex PLD (CPLD) | 82 | 24 | 150 |

**Table 4. JTAG signals on port E**

| Port E pins | JTAG signal |
|---|---|
| PE0 | TMS |
| PE1 | TCK |
| PE2 | TDI |
| PE3 | TDO |
| PE4 | TSTAT |
| PE5 | TERR |

## 2.6 In-system programming (ISP)

Using the JTAG signals on Port E, the entire PSD device (memory, logic, configuration) can be programmed or erased without the use of the MCU.

## 2.7 In-application programming (IAP)

The primary Flash memory can also be programmed, or re-programmed, in-system by the MCU executing the programming algorithms out of the secondary Flash memory, or SRAM. The secondary Flash memory can be programmed the same way by executing out of the primary Flash memory. *Table 5: Methods of programming different functional blocks of the PSD* indicates which programming methods can program different functional blocks of the PSD.

## 2.8 Page register

The 8-bit Page register expands the address range of the MCU by up to 256 times. The paged address can be used as part of the address space to access external memory and peripherals, or internal memory and I/O. The Page register can also be used to change the address mapping of the Flash memory blocks into different memory spaces for IAP.

## 2.9 Power Management Unit (PMU)

The Power Management Unit (PMU) gives the user control of the power consumption on selected functional blocks based on system requirements. The PMU includes an Automatic power-down (APD) Unit that turns off device functions during MCU inactivity. The APD Unit has a Power-down mode that helps reduce power consumption.

The PSD also has some bits that are configured at run-time by the MCU to reduce power consumption of the CPLD. The Turbo bit in PMMR0 can be reset to '0' and the CPLD latches its outputs and goes to Standby mode until the next transition on its inputs.

Additionally, bits in PMMR2 can be set by the MCU to block signals from entering the CPLD to reduce power consumption. See *Section 20: Power management* for more details.

**Table 5. Methods of programming different functional blocks of the PSD**

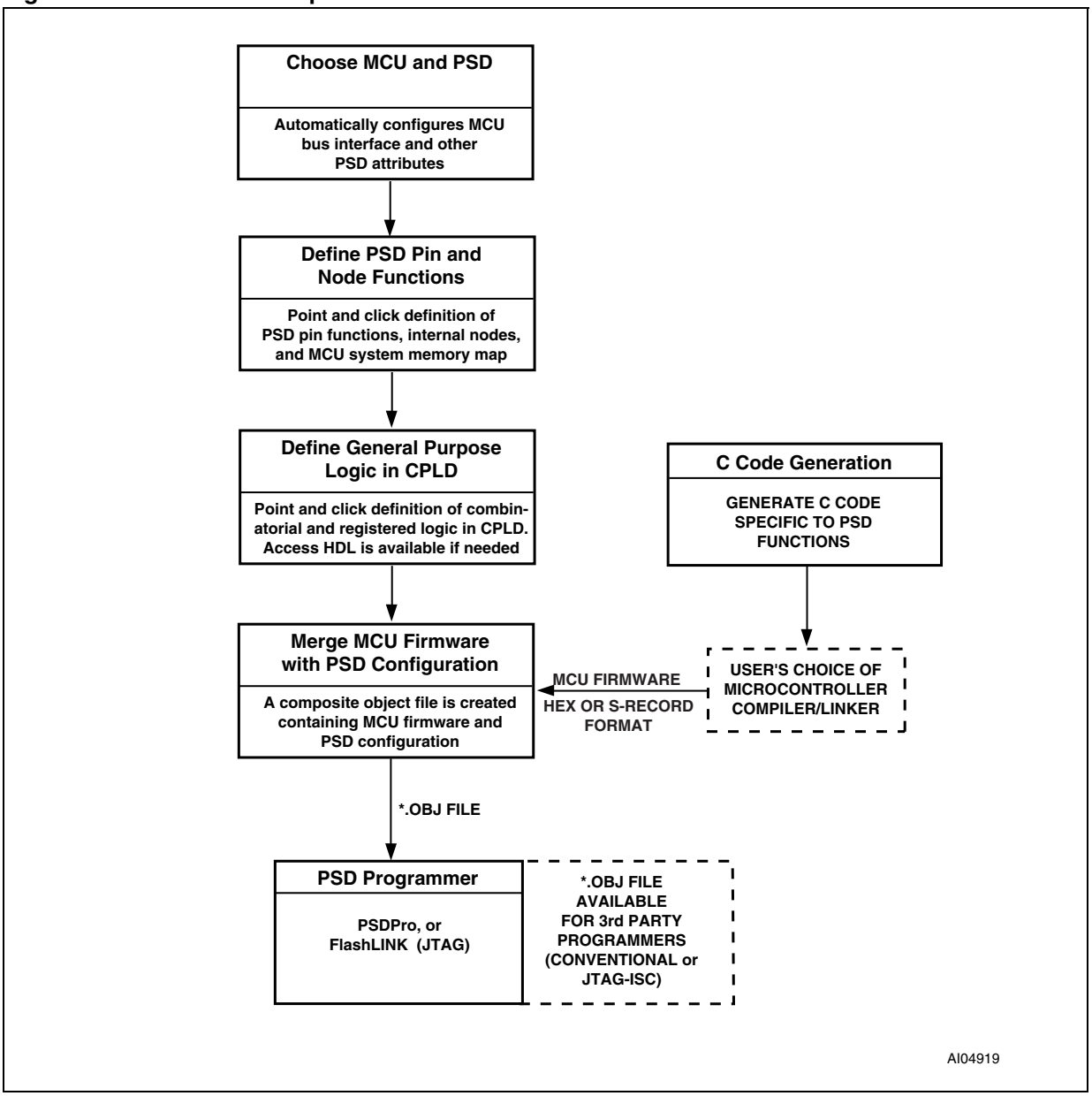
| Functional block | JTAG-ISP | Device programmer | IAP |
|---|---|---|---|
| Primary Flash memory | Yes | Yes | Yes |
| Secondary Flash memory | Yes | Yes | Yes |
| PLD array (DPLD and CPLD) | Yes | Yes | No |
| PSD configuration | Yes | Yes | No |

# 3 Development system

The PSD family is supported by PSDsoft, a Windows-based software development tool (Windows-95, Windows-98, Windows-NT). A PSD design is quickly and easily produced in a point and click environment. The designer does not need to enter Hardware Description Language (HDL) equations, unless desired, to define PSD pin functions and memory map information. The general design flow is shown in *Figure 4* PSDsoft is available from our web site (the address is given on the back page of this data sheet) or other distribution channels.

PSDsoft directly supports two low cost device programmers form ST: PSDpro and FlashLINK (JTAG). Both of these programmers may be purchased through your local distributor/representative, or directly from our web site using a credit card. The PSD is also supported by third party device programmers. See our web site for the current list.

**Figure 4. PSDsoft development tool**



# 4 PSD register description and address offsets

*Table 6: Register address offset* shows the offset addresses to the PSD registers relative to the CSIOP base address. The CSIOP space is the 256 bytes of address that is allocated by the user to the internal PSD registers. *Table 6* provides brief descriptions of the registers in CSIOP space. The following sections give a more detailed description.

**Table 6. Register address offset**

| Register name | Port A | Port B | Port C | Port D | Port E | Port F | Port G | Other(1) | Description |
|---|---|---|---|---|---|---|---|---|---|
| Data In | 00 | 01 | 10 | 11 | 30 | 40 | 41 | | Reads port pin as input, MCU I/O input mode |
| Control | | | | | 32 | 42 | 43 | | Selects mode between MCU I/O or Address Out |
| Data Out | 04 | 05 | 14 | 15 | 34 | 44 | 45 | | Stores data for output to port pins, MCU I/O output mode |
| Direction | 06 | 07 | 16 | 17 | 36 | 46 | 47 | | Configures port pin as input or output |
| Drive Select | 08 | 09 | | 19 | 38 | | 49 | | Configures port pins as either CMOS or Open Drain |
| Input Macrocell | 0A | 0B | | 1A | | | | | Reads input macrocells |
| Enable Out | 0C | 0D | 1C | | | 4C | | | Reads the status of the output enable to the I/O port driver |
| Output Macrocells A | 20 | | | | | | | | READ – reads output of macrocell A<br>WRITE – loads macrocell Flip-flops |
| Output Macrocells B | | 21 | | | | | | | READ – reads output of macrocell B<br>WRITE – loads macrocell Flip-flops |
| Mask Macrocell A | 22 | | | | | | | | Blocks writing to the output macrocells A |
| Mask Macrocell B | | 23 | | | | | | | Blocks writing to the output macrocells B |
| Flash Memory Protection 1 | | | | | | | | C0 | Read-only – Primary Flash Sector Protection |
| Flash Memory Protection 2 | | | | | | | | C1 | Read-only – Primary Flash Sector Protection |
| Flash Boot Protection | | | | | | | | C2 | Read-only – PSD Security and secondary Flash memory Sector Protection |
| JTAG Enable | | | | | | | | C7 | Enables JTAG port |
| PMMR0 | | | | | | | | B0 | Power Management register 0 |
| PMMR2 | | | | | | | | B4 | Power Management register 2 |
| Page | | | | | | | | E0 | Page register |
| VM | | | | | | | | E2 | Places PSD memory areas in Program and/or data space on an individual basis. |

**Table 6. Register address offset (continued)**

| Register name | Port A | Port B | Port C | Port D | Port E | Port F | Port G | Other(1) | Description |
|---|---|---|---|---|---|---|---|---|---|
| Memory_ID0 | | | | | | | | F0 | Read-only – SRAM and primary memory size |
| Memory_ID1 | | | | | | | | F1 | Read-only – Secondary memory type and size |

1. Other registers that are not part of the I/O ports.

# 5 Register bit definition

All the registers of the PSD are included here for reference. Detailed descriptions of these registers can be found in the following sections.

**Table 7. Data-In registers - Ports A, B, C, D, E, F, and G(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Port pin 7 | Port pin 6 | Port pin 5 | Port pin 4 | Port pin 3 | Port pin 2 | Port pin 1 | Port pin 0 |

1. Bit definitions (read-only registers): READ port pin status when port is in MCU I/O input mode.

**Table 8. Data-Out registers - Ports A, B, C, D, E, F, and G(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Port pin 7 | Port pin 6 | Port pin 5 | Port pin 4 | Port pin 3 | Port pin 2 | Port pin 1 | Port pin 0 |

1. Bit definitions: Latched data for output to port pin when pin is configured in MCU I/O output mode.

**Table 9. Direction registers - Ports A, B, C, D, E, F, and G(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Port pin 7 | Port pin 6 | Port pin 5 | Port pin 4 | Port pin 3 | Port pin 2 | Port pin 1 | Port pin 0 |

1. Bit definitions:
**Portpin <i>** 0 = Port pin <i> is configured in Input mode (default).
**Portpin <i>** 1 = Port pin <i> is configured in Output mode.

**Table 10. Control registers - Ports E, F, and G(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Port pin 7 | Port pin 6 | Port pin 5 | Port pin 4 | Port pin 3 | Port pin 2 | Port pin 1 | Port pin 0 |

1. Bit definitions:
Portpin <i> 0 = Port pin <i> is configured in MCU I/O mode (default).
Portpin <i> 1 = Port pin <i> is configured in Latched Address Out mode.

**Table 11. Drive registers - Ports A, B, D, E, and G(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Port pin 7 | Port pin 6 | Port pin 5 | Port pin 4 | Port pin 3 | Port pin 2 | Port pin 1 | Port pin 0 |

1. Bit definitions:
Portpin <i> 0 = Port pin <i> is configured for CMOS Output driver (default).
Portpin <i> 1 = Port pin <i> is configured for Open Drain output driver.

**Table 12. Enable-Out registers - Ports A, B, C, and F(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Port pin 7 | Port pin 6 | Port pin 5 | Port pin 4 | Port pin 3 | Port pin 2 | Port pin 1 | Port pin 0 |

1. Bit definitions (read-only registers):
Portpin <i> 0 = Port pin <i> is in tri-state driver (default).
Portpin <i> 1 = Port pin <i> is enabled.

**Table 13. Input Macrocells - Ports A, B, and C(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| IMcell 7 | IMcell 6 | IMcell 5 | IMcell 4 | IMcell 3 | IMcell 2 | IMcell 1 | IMcell 0 |

1. Bit definitions (read-only registers):
READ input macrocell (IMC7-IMC0) status on Ports A, B, and C.

**Table 14. Output Macrocells A register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Mcella 7 | Mcella 6 | Mcella 5 | Mcella 4 | Mcella 3 | Mcella 2 | Mcella 1 | Mcella 0 |

1. Bit definitions:
WRITE register: Load MCellA7-MCellA0 with '0' or '1.'
READ register: Read MCellA7-MCellA0 output status.

**Table 15. Out Macrocell B register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Mcellb 7 | Mcellb 6 | Mcellb 5 | Mcellb 4 | Mcellb 3 | Mcellb 2 | Mcellb 1 | Mcellb 0 |

1. Bit definitions:
WRITE register: Load MCellB7-MCellB0 with '0' or '1.'
READ register: Read MCellB7-MCellB0 output status.

**Table 16. Mask Macrocell A register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Mcella 7 | Mcella 6 | Mcella 5 | Mcella 4 | Mcella 3 | Mcella 2 | Mcella 1 | Mcella 0 |

1. Bit definitions:
McellA<i>_Prot 0 = Allow MCellA<i> flip-flop to be loaded by MCU (default).
McellA<i>_Prot 1 = Prevent MCellA<i> flip-flop from being loaded by MCU.

**Table 17. Mask Macrocell B register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Mcellb 7 | Mcellb 6 | Mcellb 5 | Mcellb 4 | Mcellb 3 | Mcellb 2 | Mcellb 1 | Mcellb 0 |

1. Bit definitions:
McellB<i>_Prot 0 = Allow MCellB<i> flip-flop to be loaded by MCU (default).
McellB<i>_Prot 1 = Prevent MCellB<i> flip-flop from being loaded by MCU.

**Table 18. Flash Memory Protection register 1(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Sec7_Prot | Sec6_Prot | Sec5_Prot | Sec4_Prot | Sec3_Prot | Sec2_Prot | Sec1_Prot | Sec0_Prot |

1. Bit definitions (read-only register):
Sec<i>_Prot 1 = Primary Flash memory Sector <i> is write protected.
Sec<i>_Prot 0 = Primary Flash memory Sector <i> is not write protected.

**Table 19. Flash Memory Protections register 2(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Sec15_ Prot | Sec14_ Prot | Sec13_ Prot | Sec12_ Prot | Sec11_ Prot | Sec10_ Prot | Sec9_Prot | Sec8_Prot |

1. Bit definitions (read-only register):
Sec<i>_Prot 1 = Primary Flash memory Sector <i> is write protected.
Sec<i>_Prot 0 = Primary Flash memory Sector <i> is not write protected.

**Table 20. Flash Boot Protection register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Security_ Bit | not used | not used | not used | Sec3_Prot | Sec2_Prot | Sec1_Prot | Sec0_Prot |

1. Bit definitions:
Sec<i>_Prot 1 = Secondary Flash memory Sector <i> is write protected.
Sec<i>_Prot 0 = Secondary Flash memory Sector <i> is not write protected.
Security_Bit 0 = Security Bit in device has not been set.
Security_Bit 1 = Security Bit in device has been set.

**Table 21. JTAG Enable register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| not used | not used | not used | not used | not used | not used | not used | JTAGEnable |

1. Bit definitions:
JTAGEnable 1 = JTAG port is enabled.
JTAGEnable 0 = JTAG port is disabled.

**Table 22. Page register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| PGR 7 | PGR 6 | PGR 5 | PGR 4 | PGR 3 | PGR 2 | PGR 1 | PGR 0 |

1. Bit definitions: configure Page input to PLD. Default is PGR7-PGR0 = '0.'

**Table 23. PMMR0 register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| not used (set to 0) | not used (set to 0) | PLD MCells CLK | PLD Array CLK | PLD Turbo | not used (set to 0) | APD Enable | not used (set to 0) |

1. The bits of this register are cleared to zero following power-up. Subsequent Reset ($\overline{\text{RESET}}$) pulses do not clear the registers.
Bit definitions:
APD Enable
0 = Automatic power-down (APD) is disabled.
1 = Automatic power-down (APD) is enabled.
PLD Turbo
0 = PLD Turbo is on.
1 = PLD Turbo is off, saving power.
PLD Array CLK
0 = CLKIN to the PLD AND array is connected. Every CLKIN change powers up the PLD when Turbo Bit is off.
1 = CLKIN to the PLD AND array is disconnected, saving power.
PLD MCells CLK
0 = CLKIN to the PLD macrocell is connected.
1 = CLKIN to the PLD macrocell is disconnected, saving power.

**Table 24. PMMR2 register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| not used (set to 0) | PLD Array WRH | PLD Array ALE | PLD Array CNTL2 | PLD Array CNTL1 | PLD Array CNTL0 | not used (set to 0) | PLD Array Addr |

1. For Bit 4, Bit 3, and Bit 2: see *Table 34* and *Table 35* for the signals that are blocked on pins CNTL0-CNTL2.
Bit Definitions:
PLD Array Addr
0 = Address A7-A0 are connected to the PLD array.
1 = Address A7-A0 are blocked from the PLD array, saving power.
Note: In X A mode, A3-A0 come from PF3-PF0, and A7-A4 come from ADIO7-ADIO4.
PLD Array CNTL2
0 = CNTL2 input to the PLD AND array is connected.
1 = CNTL2 input to the PLD AND array is disconnected, saving power.
PLD Array CNTL1
0 = CNTL1 input to the PLD AND array is connected.
1 = CNTL1 input to the PLD AND array is disconnected, saving power.
PLD Array CNTL0
0 = CNTL0 input to the PLD AND array is connected.
1 = CNTL0 input to the PLD AND array is disconnected, saving power.
PLD Array ALE
0 = ALE input to the PLD AND array is connected.
1 = ALE input to the PLD AND array is disconnected, saving power.
PLD Array WRH
0 = WRH/DBE input to the PLD AND array is connected.
1 = WRH/DBE input to the PLD AND array is disconnected, saving power.

**Table 25. VM register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Peripheral mode | not used (set to 0) | not used (set to 0) | FL_data | Boot_data | FL_code | Boot_code | SR_code |

1. On $\overline{\text{RESET}}$, bits 1-4 are loaded to configurations that are selected by the user in PSDsoft. Bit 0 and Bit 7 are always cleared on $\overline{\text{RESET}}$. Bit 0-4 are active only when the device is configured in 8051 mode.
Bit definitions:
SR_code
0 = PSEN cannot access SRAM in 80C51XA modes.
1 = PSEN can access SRAM in 80C51XA modes.
Boot_Code
0 = PSEN cannot access secondary NVM in 80C51XA modes.
1 = PSEN can access secondary NVM in 80C51XA modes.
FL_Code
0 = PSEN cannot access primary Flash memory in 80C51XA modes.
1 = PSEN can access primary Flash memory in 80C51XA modes.
Boot_data
0 = RD cannot access secondary NVM in 80C51XA modes.
1 = RD can access secondary NVM in 80C51XA modes.
FL_data
0 = RD cannot access primary Flash memory in 80C51XA modes.
1 = RD can access primary Flash memory in 80C51XA modes.
Peripheral mode
0 = Peripheral mode of Port F is disabled.
1 = Peripheral mode of Port F is enabled.

**Table 26. Memory_ID0 register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| S_size 3 | S_size 2 | S_size 1 | S_size 0 | F_size 3 | F_size 2 | F_size 1 | F_size 0 |

1. Bit definitions:
F_size[3:0]
0h = There is no primary Flash memory
1h = Primary Flash memory size is 256 Kbit
2h = Primary Flash memory size is 512 Kbit
3h = Primary Flash memory size is 1 Mbit
4h = Primary Flash memory size is 2 Mbit
5h = Primary Flash memory size is 4 Mbit
6h = Primary Flash memory size is 8 Mbit
S_size[3:0] 0h = There is no SRAM
1h = SRAM size is 16 Kbit
2h = SRAM size is 32 Kbit
3h = SRAM size is 64 Kbit
4h = SRAM size is 128 Kbit
5h = SRAM size is 256 Kbit

**Table 27. Memory_ID1 register(1)**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| not used (set to 0) | not used (set to 0) | B_type 1 | B_type 0 | B_size 3 | B_size 2 | B_size 1 | B_size 0 |

1. Bit Definitions:
F_size[3:0]
0h = There is no secondary NVM
1h = Secondary NVM size is 128 Kbit
2h = Secondary NVM size is 256 Kbit
3h = Secondary NVM size is 512 Kbit
S_size[3:0]
0h = Secondary NVM is Flash memory
1h = Secondary NVM is EEPROM

# 6 Detailed operation

As shown in *Figure 3: PSD block diagram*, the PSD consists of six major types of functional blocks:

- Memory blocks
- MCU Bus Interface
- I/O ports
- Power Management Unit (PMU)
- JTAG-ISP interface
- The functions of each block are described in the following sections. Many of the blocks perform multiple functions, and are user configurable.

## 6.1 Memory blocks

The PSD has the following memory blocks:

- Primary Flash memory
- Secondary Flash memory
- SRAM

The Memory Select signals for these blocks originate from the Decode PLD (DPLD) and are user-defined in PSDsoft.

*Table 28* summarizes the sizes and organizations of the memory blocks.

**Table 28. Memory block size and organization**

| Sector number | Primary Flash memory | | Secondary Flash memory | | SRAM | |
|---|---|---|---|---|---|---|
| | Sector size (Kbytes) | Sector Select signal | Sector size (Kbytes) | Sector Select signal | SRAM size (bytes) | SRAM Select signal |
| 0 | 64 | FS0 | 16 | CSBOOT0 | 32 | RS0 |
| 1 | 64 | FS1 | 8 | CSBOOT1 | | |
| 2 | 64 | FS2 | 8 | CSBOOT2 | | |
| 3 | 64 | FS3 | 32 | CSBOOT3 | | |
| 4 | 64 | FS4 | | | | |
| 5 | 64 | FS5 | | | | |
| 6 | 64 | FS6 | | | | |
| 7 | 64 | FS7 | | | | |
| 8 | 64 | FS8 | | | | |
| 9 | 64 | FS9 | | | | |
| 10 | 64 | FS10 | | | | |
| 11 | 64 | FS11 | | | | |
| 12 | 64 | FS12 | | | | |

Table 28. Memory block size and organization (continued)

| Sector number | Primary Flash memory | | Secondary Flash memory | | SRAM | |
|---|---|---|---|---|---|---|
| | Sector size (Kbytes) | Sector Select signal | Sector size (Kbytes) | Sector Select signal | SRAM size (bytes) | SRAM Select signal |
| 13 | 64 | FS13 | | | | |
| 14 | 64 | FS14 | | | | |
| 15 | 64 | FS15 | | | | |
| Total | 1024 | 16 Sectors | 64 | 4 Sectors | 32 | |

## 6.2 Primary Flash memory and secondary Flash memory description

The primary Flash memory is divided evenly into 8 sectors. The secondary Flash memory is divided into 4 sectors of different size. Each sector of either memory block can be separately protected from program and erase cycles.

Flash memory may be erased on a sector-by-sector basis, and programmed word-by-word. Flash sector erasure may be suspended while data is read from other sectors of the block and then resumed after reading.

During a program or erase cycle in Flash memory, the status can be output on the Ready/$\overline{\text{Busy}}$ pin (PE4). This pin is set up using PSDsoft.

### 6.2.1 Memory Block Select signals

The DPLD generates the Select signals for all the internal memory blocks (see *Section 15: PLDs*). Each of the sectors of the primary Flash memory has a Select signal (FS0-FS15) which can contain up to three product terms. Each of the sectors of the secondary Flash memory has a Select signal (CSBOOT0-CSBOOT3) which can contain up to three product terms. Having three product terms for each Select signal allows a given sector to be mapped in different areas of system memory. When using a MCU with separate program and data space (80C51XA), these flexible Select signals allow dynamic re-mapping of sectors from one memory space to the other before and after IAP. The SRAM block has a single Select signal (RS0).

### 6.2.2 Ready/$\overline{\text{Busy}}$ (PE4)

This signal can be used to output the Ready/$\overline{\text{Busy}}$ status of the PSD. The output is a '0' (Busy) when a Flash memory block is being written to, *or* when a Flash memory block is being erased. The output is a '1' (Ready) when no WRITE or Erase cycle is in progress.

### 6.2.3 Memory operation

The primary Flash memory and secondary Flash memory are addressed through the MCU Bus Interface. The MCU can access these memories in one of two ways:

- The MCU can execute a typical bus WRITE or READ operation just as it would if accessing a RAM or ROM device using standard bus cycles.
- The MCU can execute a specific instruction that consists of several WRITE and READ operations. This involves writing specific data patterns to special addresses within the Flash memory to invoke an embedded algorithm. These instructions are summarized in *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*.

Typically, the MCU can read Flash memory using READ operations, just as it would read a ROM device. However, Flash memory can only be erased and programmed using specific instructions. For example, the MCU cannot write a single byte directly to Flash memory as one would write a byte to RAM. To program a word into Flash memory, the MCU must execute a program instruction, then test the status of the programming event. This status test is achieved by a READ operation or polling Ready/$\overline{\text{Busy}}$ (PE4).

Flash memory can also be read by using special instructions to retrieve particular Flash device information (sector protect status and ID).

**Table 29. 16-bit instructions(1)(2)(3)(4)**

| Instruction(5) | FS0-FS15 or CSBOOT0-CSBOOT3 | Cycle 1 | Cycle 2 | Cycle 3 | Cycle 4 | Cycle 5 | Cycle 6 | Cycle 7 |
|---|---|---|---|---|---|---|---|---|
| READ(6) | 1 | “Read” RD @ RA | | | | | | |
| READ Main Flash ID(7)(8) | 1 | AAh@ XAAAh | 55h@ X554h | 90h@ XAAAh | Read ID @ XX02h | | | |
| READ Sector Protection(7)(8)(9) | 1 | AAh@ XAAAh | 55h@ X554h | 90h@ XAAAh | Read 00h or 01h @ XX04h | | | |
| Program a Flash Word(8) | 1 | AAh@ XAAAh | 55h@ X554h | A0h@ XAAAh | PD@ PA | | | |
| Flash Sector Erase(8)(10) | 1 | AAh@ XAAAh | 55h@ X554h | 80h@ XAAAh | AAh@ XAAAh | 55h@ X554h | 30h@ SA | 30h(7)@ next SA |
| Flash Bulk Erase(8) | 1 | AAh@ XAAAh | 55h@ X554h | 80h@ XAAAh | AAh@ XAAAh | 55h@ X554h | 10h@ XAAAh | |
| Suspend Sector Erase(11) | 1 | B0h@ XXXXh | | | | | | |
| Resume Sector Erase(12) | 1 | 30h@ XXXXh | | | | | | |
| $\overline{\text{RESET}}$(7) | 1 | F0h@ XXXXh | | | | | | |
| Unlock Bypass | 1 | AAh@ XAAAh | 55h@ X554h | 20h@ XAAAh | | | | |

**Table 29. 16-bit instructions(1)(2)(3)(4)**

| Instruction(5) | FS0-FS15 or CSBOOT0-CSBOOT3 | Cycle 1 | Cycle 2 | Cycle 3 | Cycle 4 | Cycle 5 | Cycle 6 | Cycle 7 |
|---|---|---|---|---|---|---|---|---|
| Unlock Bypass Program(13) | 1 | A0h@ XXXXh | PD@ PA | | | | | |
| Unlock Bypass Reset(14) | 1 | 90h@ XXXXh | 00h@ XXXXh | | | | | |

1. All bus cycles are WRITE bus cycles, except the ones with the “Read” label.
2. All values are in hexadecimal:
   X = “Don’t care.” Addresses of the form XXXXh, in this table, must be even addresses
   RA = Address of the memory location to be read
   RD = Data read from location RA during the READ cycle
   PA = Address of the memory location to be programmed. Addresses are latched on the falling edge of WRITE Strobe ($\overline{WR}$, CNTL0).
   PA is an even address for PSD in word programming mode.
   PD = Data word to be programmed at location PA. Data is latched on the rising edge of WRITE Strobe ($\overline{WR}$, CNTL0)
   SA = Address of the sector to be erased or verified. The Sector Select (FS0-FS15 or CSBOOT0-CSBOOT3) of the sector to be erased, or verified, must be Active (High).
3. Sector Select (FS0 to FS15 or CSBOOT0 to CSBOOT3) signals are active High, and are defined in PSDsoft.
4. Only address bits A11-A0 are used in instruction decoding.
5. All WRITE bus cycles in an instruction are byte-WRITE to an even address (XAAAh or X554h). A Flash memory program bus cycle writes a word to an even address.
6. No Unlock or instruction cycles are required when the device is in the Read mode.
7. The RESET instruction is required to return to the Read mode after reading the Flash ID, or after reading the Sector Protection Status, or if the Error Flag bit (DQ5/DQ13) goes High.
8. The MCU cannot invoke these instructions while executing code from the same Flash memory as that for which the instruction is intended. The MCU must retrieve, for example, the code from the secondary Flash memory when reading the Sector Protection Status of the primary Flash memory.
9. The data is 00h for an unprotected sector, and 01h for a protected sector. In the fourth cycle, the Sector Select is active, and (A1,A0) = (1,0).
10. Additional sectors to be erased must be written at the end of the Sector Erase instruction within 80µs.
11. The system may perform read and program cycles in non-erasing sectors, read the Flash ID or read the Sector Protection Status when in the Suspend Sector Erase mode. The Suspend Sector Erase instruction is valid only during a Sector Erase cycle.
12. The Resume Sector Erase instruction is valid only during the Suspend Sector Erase mode.
13. The Unlock Bypass instruction is required prior to the Unlock Bypass Program instruction.
14. The Unlock Bypass Reset Flash instruction is required to return to reading memory data when the device is in the Unlock Bypass mode.

**Table 30. 8-bit instructions**

| Instruction[1] | FS0-FS7 or CSBOOT0-CSBOOT3 | Cycle 1 | Cycle 2 | Cycle 3 | Cycle 4 | Cycle 5 | Cycle 6 | Cycle 7 |
|---|---|---|---|---|---|---|---|---|
| READ[2] | 1 | "Read" RD @ RA | | | | | | |
| READ Main Flash ID[3][4] | 1 | AAh@ 555h | 55h@ AAAh | 90h@ 555h | READ ID@ 01h | | | |
| READ Sector Protection[3][4][5] | 1 | AAh@ 555h | 55h@ AAAh | 90h@ 555h | Read 00h or 01h @ 02h | | | |
| Program a Flash Word[4] | 1 | AAh@ 555h | 55h@ AAAh | A0h@ 555h | PD@ PA | | | |
| Flash Sector Erase[4][6] | 1 | AAh@ 555h | 55h@ AAAh | 80h@ 555h | AAh@ 555h | 55h@ AAAh | 30h@ SA | 30h[7]@ next SA |
| Flash Bulk Erase[4] | 1 | AAh@ 555h | 55h@ AAAh | 80h@ 555h | AAh@ 555h | 55h@ AAAh | 10h@ 555h | |
| Suspend Sector Erase[7] | 1 | B0h@ XXXh | | | | | | |
| Resume Sector Erase[8] | 1 | 30h@ XXXh | | | | | | |
| $\overline{\text{RESET}}$[3] | 1 | F0h@ XXXh | | | | | | |
| Unlock Bypass | 1 | AAh@ 555h | 55h@ AAAh | 20h@ 555h | | | | |
| Unlock Bypass Program[9] | 1 | A0h@ XXXh | PD@ PA | | | | | |
| Unlock Bypass Reset[10] | 1 | 90h@ XXXh | 00h@ XXXh | | | | | |

1. All WRITE bus cycles in an instruction are byte-WRITE to an even address (555h or AAAh). A Flash memory Program bus cycle writes a word to an even address.
2. No Unlock or instruction cycles are required when the device is in the Read mode.
3. The RESET instruction is required to return to the Read mode after reading the Flash ID, or after reading the Sector Protection Status, or if the Error Flag bit (DQ5/DQ13) goes High.
4. The MCU cannot invoke these instructions while executing code from the same Flash memory as that for which the instruction is intended. The MCU must retrieve, for example, the code from the secondary Flash memory when reading the Sector Protection Status of the primary Flash memory.
5. The data is 00h for an unprotected sector, and 01h for a protected sector. In the fourth cycle, the Sector Select is active, and (A1,A0) = (1,0).
6. Additional sectors to be erased must be written at the end of the Sector Erase instruction within 80 µs.
7. The system may perform READ and Program cycles in non-erasing sectors, read the Flash ID or read the Sector Protection Status when in the Suspend Sector Erase mode. The Suspend Sector Erase instruction is valid only during a Sector Erase cycle.
8. The Resume Sector Erase instruction is valid only during the Suspend Sector Erase mode.
9. The Unlock Bypass instruction is required prior to the Unlock Bypass program instruction.
10. The Unlock Bypass Reset Flash instruction is required to return to reading memory data when the device is in the Unlock Bypass mode.

# 7 Instructions

An instruction consists of a sequence of specific operations. Each received byte is sequentially decoded by the PSD 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 timeout 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 timeout between two consecutive bytes while addressing Flash memory resets the device logic into Read mode (Flash memory is read like a ROM device).

The PSD supports the instructions summarized in *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*:

- Erase memory by chip or sector
- Suspend or resume sector erase
- Program a Word
- $\overline{\text{RESET}}$ to Read mode
- READ Primary Flash Identifier value
- READ Sector Protection Status
- Bypass

These instructions are detailed in *Table 29* and *Table 30*. 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 XAAAh during the first cycle and data 55h to address X554h during the second cycle (unless the Bypass instruction feature is used, as described later). Address signals A15-A12 are “Don’t care” during the instruction WRITE cycles. However, the appropriate Sector Select signal (FS0-FS15, 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 its Sector Select signals (FS0-FS15) is High, and the secondary Flash memory is selected if any one of its Sector Select signals (CSBOOT0-CSBOOT3) is High.

## 7.1 Power-up condition

The PSD internal logic is reset upon Power-up to the Read mode. Sector Select (FS0-FS15 and CSBOOT0-CSBOOT3) must be held Low, and WRITE Strobe ($\overline{\text{WR}}$/$\overline{\text{WRL}}$, CNTL0) High, during Power-up for maximum security of the data contents and to remove the possibility of data being written on the first edge of WRITE Strobe ($\overline{\text{WR}}$/$\overline{\text{WRL}}$, CNTL0). Any WRITE cycle initiation is locked when $V_{CC}$ is below $V_{LKO}$.

## 7.2 READ

Under typical conditions, the MCU may read the primary Flash memory, or secondary Flash memory, using READ operations just as it would a ROM or RAM device. Alternately, the MCU may use READ operations to obtain status information about a Program or Erase cycle that is currently in progress. Lastly, the MCU may use instructions to read special data from these memory blocks. The following sections describe these READ functions.

## 7.3 READ Memory Contents

Primary Flash memory and secondary Flash memory are placed in the Read mode after Power-up, chip reset, or a Reset Flash instruction (see *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*). The MCU can read the memory contents of the primary Flash memory, or the secondary Flash memory by using READ operations any time the READ operation is not part of an instruction.

## 7.4 READ Primary Flash Identifier

The primary Flash memory identifier is read with an instruction composed of 4 operations: 3 specific WRITE operations and a READ operation (see *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*). The identifier for the primary Flash memory is E9h. The secondary Flash memory does not support this instruction.

## 7.5 READ Memory Sector Protection Status

The Flash memory Sector Protection Status is read with an instruction composed of four operations: three specific WRITE operations and a READ operation (see *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*). The READ operation produces 01h if the Flash memory sector is protected, or 00h if the sector is not protected.

The sector protection status for all NVM blocks (primary Flash memory, or secondary Flash memory) can be read by the MCU accessing the Flash Protection and Flash Boot Protection registers in PSD I/O space. See *Section 10.1: Flash Memory Sector Protect* for register definitions.

## 7.6 Reading the Erase/Program Status bits

The PSD provides several status bits to be used by the MCU to confirm the completion of an Erase or Program cycle of Flash memory. These status bits minimize the time that the MCU spends performing these tasks and are defined in *Table 31: Status bits*. The status byte resides in an even location, and can be read as many times as needed. Also note that DQ15-DQ8 is an even byte for Motorola MCUs with a 16-bit data bus.

For Flash memory, the MCU can perform a READ operation to obtain these status bits while an Erase or Program instruction is being executed by the embedded algorithm. See *Section 8: Programming Flash memory* for details.

**Table 31. Status bits**

| DQ7 | DQ6 | DQ5 | DQ4 | DQ3 | DQ2 | DQ1 | DQ0 |
|---|---|---|---|---|---|---|---|
| Data Polling | Toggle Flag | Error Flag | X | Erase Time-out | X | X | X |

**Status bits for Motorola 16-bit MCU(1)**

| DQ15 | DQ14 | DQ13 | DQ12 | DQ11 | DQ10 | DQ9 | DQ8 |
|---|---|---|---|---|---|---|---|
| Data Polling | Toggle Flag | Error Flag | X | Erase Time-out | X | X | X |

1. X = Not guaranteed value, can be read either '1' or '0.'
DQ15-DQ0 represent the Data Bus bits, D15-D0.
FS0-FS15/CSBOOT0-CSBOOT3 are active High.

## 7.7 Data Polling (DQ7) - DQ15 for Motorola

When erasing or programming in Flash memory, the Data Polling bit (DQ7/DQ15) outputs the complement of the bit being entered for programming/writing on the DQ7/DQ15 bit. Once the Program instruction or the WRITE operation is completed, the true logic value is read on the Data Polling bit (DQ7/DQ15) (in a READ operation).

- Data polling is effective after the fourth WRITE pulse (for a Program instruction) or after the sixth WRITE pulse (for an Erase instruction). It must be performed at the address being programmed or at an address within the Flash memory sector being erased.
- During an Erase cycle, the Data Polling bit (DQ7/DQ15) outputs a '0.' After completion of the cycle, the Data Polling bit (DQ7/DQ15) outputs the last bit programmed (it is a '1' after erasing).
- If the location to be programmed is in a protected Flash memory sector, the instruction is ignored.
- If all the Flash memory sectors to be erased are protected, the Data Polling bit (DQ7/DQ15) is reset to '0' for about 100 µs, and then returns to the value from the previously addressed location. No erasure is performed.

## 7.8 Toggle Flag (DQ6) – DQ14 for Motorola

The PSD offers another way for determining when the Flash memory Program cycle is completed. During the internal WRITE operation and when either FS0-FS15 or CSBOOT0-CSBOOT3 is true, the Toggle Flag bit (DQ6/DQ14) toggles from 0 to '1' and '1' to '0' on subsequent attempts to read any word of the memory.

When the internal cycle is complete, the toggling stops and the data read on the Data Bus D0-D7 is the value from the addressed memory location. The device is now accessible for a new READ or WRITE operation.

The cycle is finished when two successive READs yield the same output data.

- The Toggle Flag bit (DQ6/DQ14) is effective after the fourth WRITE pulse (for a Program instruction) or after the sixth WRITE pulse (for an Erase instruction).
- If the location to be programmed belongs to a protected Flash memory sector, the instruction is ignored.
- If all the Flash memory sectors selected for erasure are protected, the Toggle Flag bit (DQ6/DQ14) toggles to '0' for about 100 µs and then returns to the value from the previously addressed location.

## 7.9 Error Flag (DQ5) – DQ13 for Motorola

During a normal Program or Erase cycle, the Error Flag bit (DQ5/DQ13) is reset to '0.' This bit is set to '1' when there is a failure during a Flash memory Program, Sector Erase, or Bulk Erase cycle.

In the case of Flash memory programming, the Error Flag bit (DQ5/DQ13) indicates the attempt to program a Flash memory bit, or bits, from the programmed state, 0, to the erased state, '1,' which is not a valid operation. The Error Flag bit (DQ5/DQ13) may also indicate a Time-out condition while attempting to program a word.

In case of an error in a Flash memory Sector Erase or Word Program cycle, the Flash memory sector in which the error occurred or to which the programmed location belongs must no longer be used. Other Flash memory sectors may still be used. The Error Flag bit (DQ5/DQ13) is reset after a $\overline{\text{RESET}}$ instruction. A $\overline{\text{RESET}}$ instruction is required after detecting an error on the Error Flag bit (DQ5/DQ13).

## 7.10 Erase Time-out Flag (DQ3) – DQ11 for Motorola

The Erase Time-out Flag bit (DQ3/DQ11) reflects the timeout period allowed between two consecutive Sector Erase instructions. The Erase Time-out Flag bit (DQ3/DQ11) is reset to '0' after a Sector Erase cycle for a period of 100 µs + 20% unless an additional Sector Erase instruction is decoded. After this period, or when the additional Sector Erase instruction is decoded, the Erase Time-out Flag bit (DQ3/DQ11) is set to '1.'

# 8 Programming Flash memory

Flash memory must be erased prior to being programmed. The MCU may erase Flash memory all at once or by-sector. Although erasing Flash memory occurs on a sector or device basis, programming Flash memory occurs on a word basis.

The primary and secondary Flash memories require the MCU to send an instruction to program a word or to erase sectors (see *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*).

Once the MCU issues a Flash memory Program or Erase instruction, it must check the status bits for completion. The embedded algorithms that are invoked inside the PSD support several means to provide status to the MCU. Status may be checked using any of three methods: data polling, data toggle, or Ready/$\overline{\text{Busy}}$ (PE4) signal.

## 8.1 Data polling

Polling on the Data Polling bit (DQ7/DQ15) is a method of checking whether a Program or Erase cycle is in progress or has completed. *Figure 5* shows the data polling algorithm.

When the MCU issues a Program instruction, the embedded algorithm within the PSD begins. The MCU then reads the location of the word to be programmed in Flash memory to check the status. The Data Polling bit (DQ7/DQ15) becomes the complement of the corresponding bit of the original data word to be programmed. The MCU continues to poll this location, comparing data and monitoring the Error Flag bit (DQ5/DQ13). When the Data Polling bit (DQ7/DQ15) matches the corresponding bit of the original data, and the Error Flag bit (DQ5/DQ13) remains '0,' the embedded algorithm is complete. If the Error Flag bit (DQ5/DQ13) is '1,' the MCU should test the Data Polling bit (DQ7/DQ15) again since the Data Polling bit (DQ7/DQ15) may have changed simultaneously with the Error Flag bit (DQ5/DQ13) (see *Figure 5*).

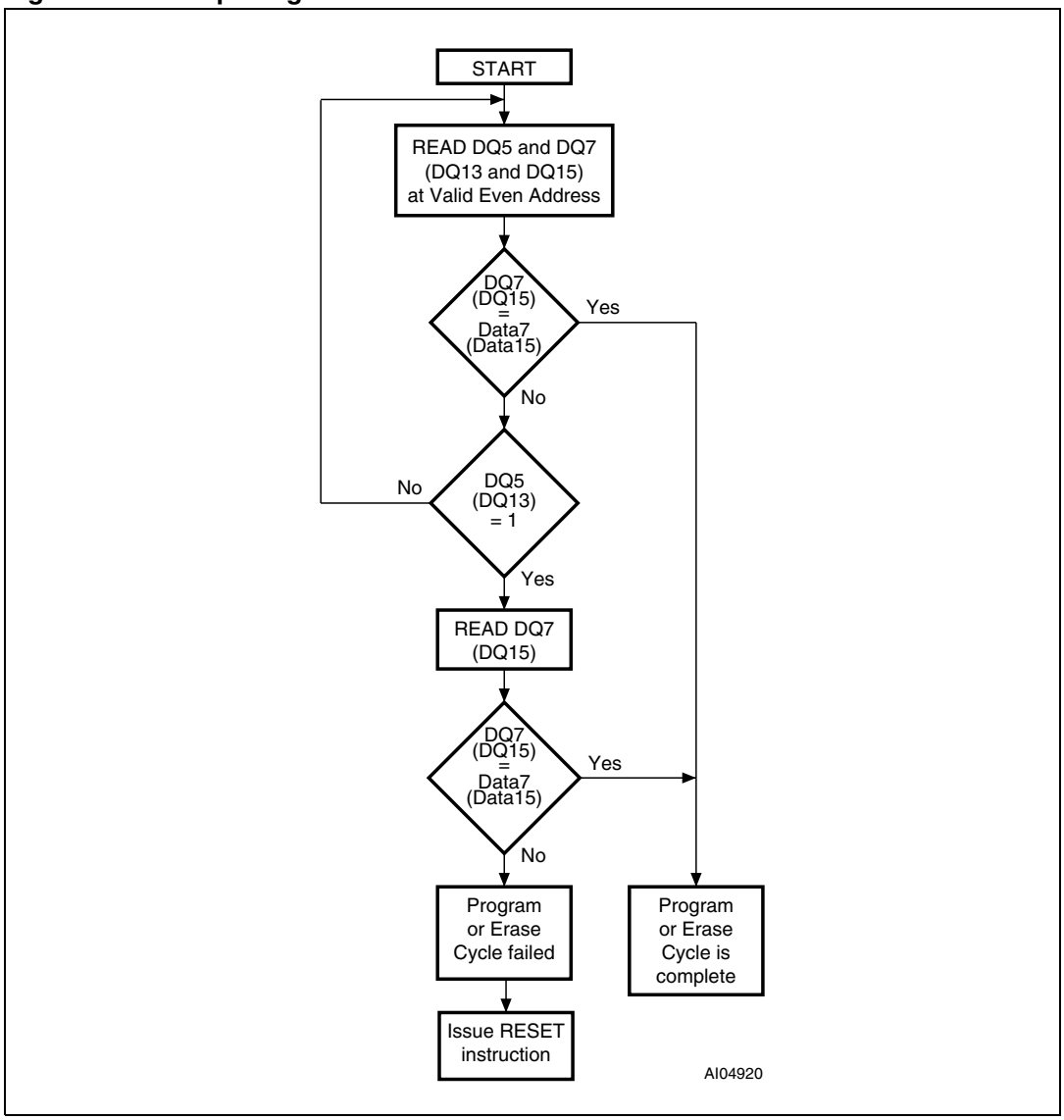
The Error Flag bit (DQ5/DQ13) is set if either an internal timeout occurred while the embedded algorithm attempted to program the location or if the MCU attempted to program a '1' to a bit that was not erased (not erased is logic '0').

It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed, to compare the word that was written to the Flash memory with the word that was intended to be written.

When using the data polling method during an Erase cycle, *Figure 5* still applies. However, the Data Polling bit (DQ7/DQ15) is '0' until the Erase cycle is complete. A '1' on the Error Flag bit (DQ5/DQ13) indicates a timeout condition on the Erase cycle, a 0 indicates no error. The MCU can read any even location within the sector being erased to get the Data Polling bit (DQ7/DQ15) and the Error Flag bit (DQ5/DQ13).

PSDsoft generates ANSI C code functions that implement these Data polling algorithms.

**Figure 5. Data polling flowchart**




## 8.2 Data toggle

Checking the Toggle Flag bit (DQ6/DQ14) is another method of determining whether a Program or Erase cycle is in progress or has completed. *Figure 6* shows the data toggle algorithm.

When the MCU issues a Program instruction, the embedded algorithm within the PSD begins. The MCU then reads the location to be programmed in Flash memory to check the status. The Toggle Flag bit (DQ6/DQ14) toggles each time the MCU reads this location until the embedded algorithm is complete. The MCU continues to read this location, checking the Toggle Flag bit (DQ6/DQ14) and monitoring the Error Flag bit (DQ5/DQ13). When the Toggle Flag bit (DQ6/DQ14) stops toggling (two consecutive READs yield the same value), and the Error Flag bit (DQ5/DQ13) remains '0,' the embedded algorithm is complete. If the Error Flag bit (DQ5/DQ13) is '1,' the MCU should test the Toggle Flag bit (DQ6/DQ14) again,

since the Toggle Flag bit (DQ6/DQ14) may have changed simultaneously with the Error Flag bit (DQ5/DQ13) (see *Figure 6*).

The Error Flag bit (DQ5/DQ13) is set if either an internal timeout occurred while the embedded algorithm attempted to program, or if the MCU attempted to program a '1' to a bit that was not erased (not erased is logic '0').

It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed, to compare the word that was written to Flash memory with the word that was intended to be written.

When using the data toggle method after an Erase cycle, *Figure 6* still applies. the Toggle Flag bit (DQ6/DQ14) toggles until the Erase cycle is complete. A '1' on the Error Flag bit (DQ5/DQ13) indicates a timeout condition on the Erase cycle, a '0' indicates no error. The MCU can read any even location within the sector being erased to get the Toggle Flag bit (DQ6/DQ14) and the Error Flag bit (DQ5/DQ13).

PSDsoft generates ANSI C code functions which implement these Data Toggling algorithms.

## 8.3 Unlock Bypass

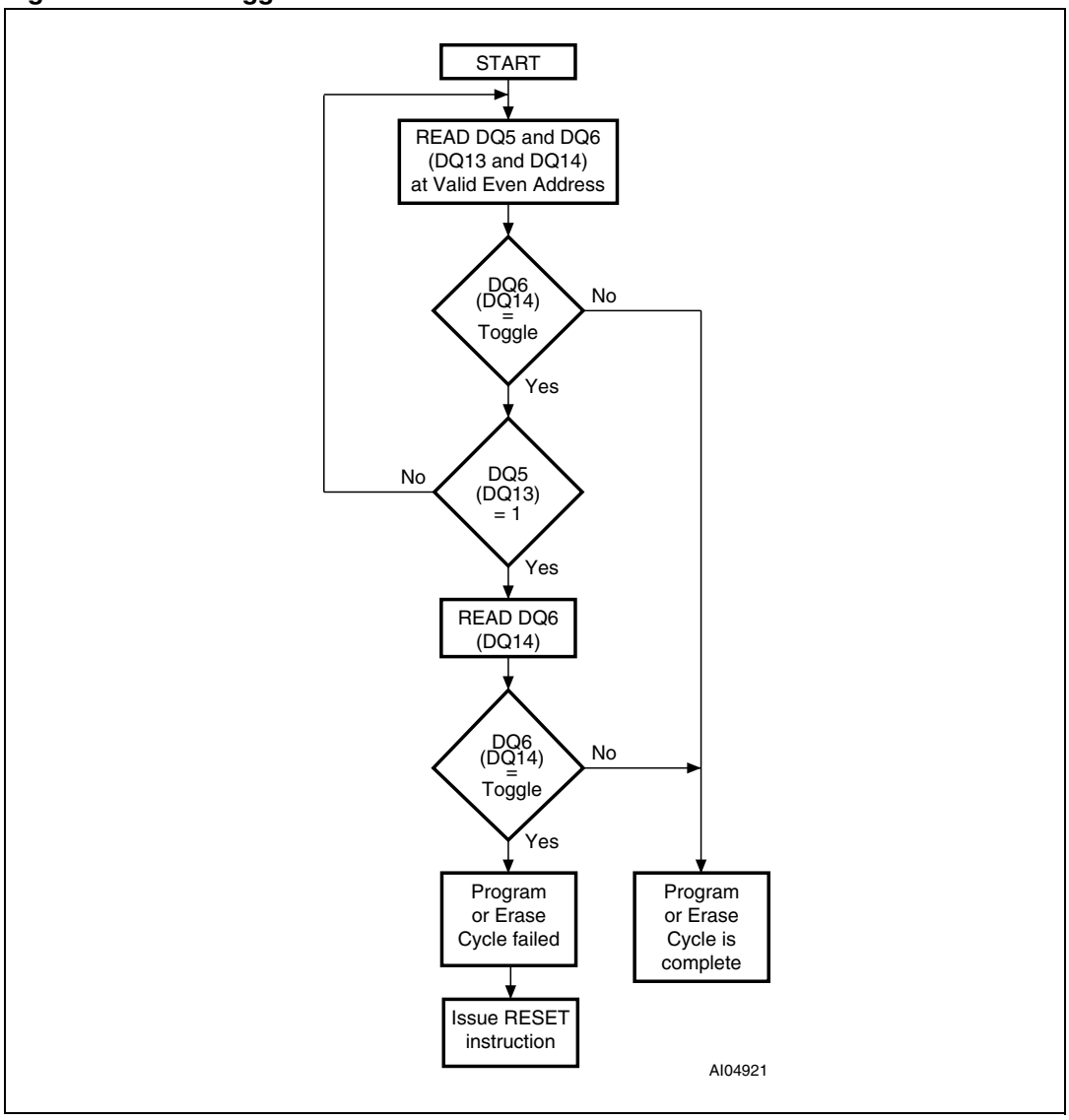
The Unlock Bypass instruction allows the system to program words to the Flash memories faster than using the standard Program instruction. The Unlock Bypass mode is entered by first initiating two Unlock cycles. This is followed by a third WRITE cycle containing the Unlock Bypass command, 20h (as shown in *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*). The Flash memory then enters the Unlock Bypass mode.

A two-cycle Unlock Bypass Program instruction is all that is required to program in this mode. The first cycle in this instruction contains the Unlock Bypass Program command, A0h. The second cycle contains the program address and data. Additional data is programmed in the same manner. This mode dispense with the initial two Unlock cycles required in the standard Program instruction, resulting in faster total programming time.

During the unlock bypass mode, only the Unlock Bypass Program and Unlock Bypass Reset instructions are valid.

To exit the Unlock Bypass mode, the system must issue the two-cycle Unlock Bypass Reset 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.

**Figure 6. Data toggle flowchart**



# 9 Erasing Flash memory

## 9.1 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 29: 16-bit instructions* and *Table 30: 8-bit instructions*. If any byte of the Bulk Erase instruction is wrong, the Bulk Erase instruction aborts and the device is reset to the READ memory mode.

During a Bulk Erase, the memory status may be checked by reading the Error Flag bit (DQ5/DQ13), the Toggle Flag bitbit (DQ6/DQ14), and the Data Polling bit (DQ7/DQ15), as detailed in *Section 8: Programming Flash memory*. The Error Flag bit (DQ5/DQ13) 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 automatically does this before erasing to 0FFh.

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

## 9.2 Flash Sector Erase

The Sector Erase instruction uses six WRITE operations, as described in *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*. Additional Flash Sector Erase confirm commands and Flash memory sector addresses can be written subsequently to erase other Flash memory sectors in parallel, without further coded cycles, if the additional commands are transmitted in a shorter time than the timeout period of about 100 µs. The input of a new Sector Erase command restarts the timeout period.

The status of the internal timer can be monitored through the level of the Erase Time-out Flag bit (DQ3/DQ11). If the Erase Time-out Flag bit (DQ3/DQ11) is '0,' the Sector Erase instruction has been received and the timeout period is counting. If the Erase Time-out Flag bit (DQ3/DQ11) is '1,' the timeout period has expired and the PSD is busy erasing the Flash memory sector(s). Before and during Erase timeout, any instruction other than Suspend Sector Erase and Resume Sector Erase, abort the cycle that is currently in progress, and reset the device to Read mode. It is not necessary to program the Flash memory sector with 00h as the PSD does this automatically before erasing.

During a Sector Erase, the memory status may be checked by reading the Error Flag bit (DQ5/DQ13), the Toggle Flag bit (DQ6/DQ14), and the Data Polling bit (DQ7/DQ15), as detailed in *Section 8: Programming Flash memory*.

During execution of the Erase cycle, the Flash memory accepts only $\overline{\text{RESET}}$ 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.

## 9.3 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 even address when an appropriate Sector Select (FS0-FS15 or CSBOOT0-CSBOOT3) is High. (See *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*). This allows reading of data from another Flash memory sector after the Erase cycle has been suspended. Suspend Sector Erase is accepted only during the Flash Sector Erase instruction execution and defaults to Read mode. A Suspend Sector Erase instruction executed during an Erase timeout period, in addition to suspending the Erase cycle, terminates the time out period.

The Toggle Flag bit (DQ6/DQ14) stops toggling when the PSD internal logic is suspended. The status of this bit must be monitored at an address within the Flash memory sector being erased. The Toggle Flag bit (DQ6/DQ14) stops toggling between 0.1 µs and 15 µs after the Suspend Sector Erase instruction has been executed. The PSD is then automatically set to Read mode.

If an Suspend Sector Erase instruction 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 $\overline{\text{RESET}}$ instructions (READ is an operation and is allowed).
- If a $\overline{\text{RESET}}$ instruction is received, data in the Flash memory sector that was being erased is invalid.

## 9.4 Resume Sector Erase

If a Suspend Sector Erase instruction was previously executed, the Erase cycle may be resumed with this instruction. The Resume Sector Erase instruction consists of writing 030h to any even address while an appropriate Sector Select (FS0-FS15 or CSBOOT0-CSBOOT3) is High. See *Table 29: 16-bit instructions* and *Table 30: 8-bit instructions*.

# 10 Specific features

## 10.1 Flash Memory Sector Protect

Each sector of primary or secondary Flash memory can be separately protected against Program and Erase cycles. Sector Protection provides additional data security because it disables all Program or Erase cycles. This mode can be activated (or deactivated) through the JTAG-ISP port or a device programmer.

Sector protection can be selected for each sector using the PSDsoft program. This automatically protects selected sectors when the device is programmed through the JTAG port or a device programmer. Flash memory sectors can be unprotected to allow updating of their contents using the JTAG port or a device programmer. The MCU can read (but cannot change) the sector protection bits.

Any attempt to program or erase a protected Flash memory sector is ignored by the device. The Verify operation results in a READ of the protected data. This allows a guarantee of the retention of the Protection status.

The sector protection status can be read by the MCU through the Flash memory protection and secondary Flash memory protection registers (in the CSIOP block) or use the READ Sector Protection instruction. See *Table 18* to *Table 20*.

## 10.2 $\overline{\text{RESET}}$ instruction

The $\overline{\text{RESET}}$ instruction consists of one WRITE cycle (see *Table 29* and *Table 30*). It can also be optionally preceded by the standard two WRITE decoding cycles (writing AAh to AAAh, and 55h to 554h).

The $\overline{\text{RESET}}$ instruction must be executed after:

- Reading the Flash Protection Status or Flash ID
- An Error condition has occurred (and the device has set the Error Flag bit (DQ5/DQ13) to '1') during a Flash memory Program or Erase cycle.

The $\overline{\text{RESET}}$ instruction immediately puts the Flash memory back into normal Read mode. However, if there is an error condition (with the Error Flag bit (DQ5/DQ13) set to '1') the Flash memory will return to the Read mode in 25 µs after the $\overline{\text{RESET}}$ instruction is issued.

The $\overline{\text{RESET}}$ instruction is ignored when it is issued during a Program or Bulk Erase cycle of the Flash memory. The $\overline{\text{RESET}}$ instruction aborts any on-going Sector Erase cycle, and returns the Flash memory to the normal Read mode in 25 µs.

## 10.3 Reset ($\overline{\text{RESET}}$) pin

A pulse on the Reset ($\overline{\text{RESET}}$) pin aborts any cycle that is in progress, and resets the Flash memory to the Read mode. When the reset occurs during a Program or Erase cycle, the Flash memory takes up to 25 µs to return to the Read mode. It is recommended that the Reset ($\overline{\text{RESET}}$) pulse (except for Power On Reset, as described in *Section 21.1: Power-on RESET*) be at least 25 µs so that the Flash memory is always ready for the MCU to retrieve the bootstrap instructions after the $\overline{\text{RESET}}$ cycle is complete.

# 11 SRAM

The SRAM is enabled when SRAM Select (RS0) from the DPLD is High. SRAM Select (RS0) can contain up to three product terms, allowing flexible memory mapping.

SRAM Select (RS0) is configured using PSDsoft.

# 12 Memory select signals

The primary Flash Memory Sector Select (FS0-FS15), secondary Flash Memory Sector Select (CSBOOT0-CSBOOT3) and SRAM Select (RS0) signals are all outputs of the DPLD. They are defined using PSDsoft. The following rules apply to the equations for these signals:

- Primary Flash memory and secondary Flash memory Sector Select signals must *not* be larger than the physical sector size.
- Any primary Flash memory sector must *not* be mapped in the same memory space as another Flash memory sector.
- A secondary Flash memory sector must *not* be mapped in the same memory space as another secondary Flash memory sector.
- SRAM, I/O, and Peripheral I/O spaces must *not* overlap.
- A secondary Flash memory sector *may* overlap a primary Flash memory sector. In case of overlap, priority is given to the secondary Flash memory sector.
- SRAM, I/O, and Peripheral I/O spaces *may* overlap any other memory sector. Priority is given to the SRAM, I/O, or Peripheral I/O.

## 12.1 Example

FS0 is valid when the address is in the range of 8000h to BFFFh, CSBOOT0 is valid from 8000h to 9FFFh, and RS0 is valid from 8000h to 87FFh. Any address 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 segment 0. Any address greater than 9FFFh accesses the primary Flash memory segment 0. You can see that half of the primary Flash memory segment 0 and one-fourth of secondary Flash memory segment 0 cannot be accessed in this example. Also note that an equation that defined FS1 to anywhere in the range of 8000h to BFFFh would *not* be valid.

*Figure 7* shows the priority levels for all memory components. Any component on a higher level can overlap and has priority over any component on a lower level. Components on the same level must *not* overlap. Level 1 has the highest priority and level 3 has the lowest.

## 12.2 Memory select configuration for MCUs with separate program and data spaces

The 80C31 and compatible family of MCUs can be configured to have separate address spaces for Program memory (selected using Program Select Enable ($\overline{\text{PSEN}}$, CNTL2)) and Data memory (selected using READ Strobe ($\overline{\text{RD}}$, CNTL1)). Any of the memories within the PSD can reside in either space or both spaces. This is controlled through manipulation of the VM register that resides in the CSIOP space.

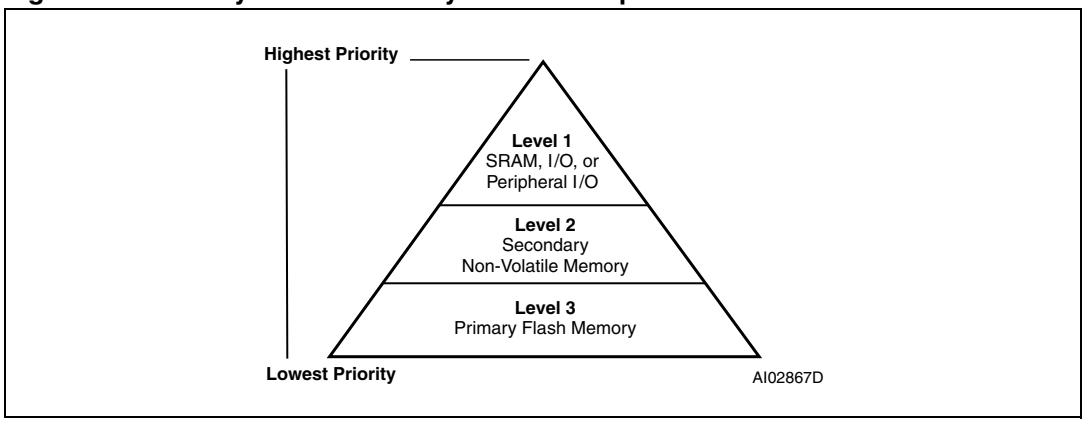
The VM register is set using PSDsoft to have an initial value. It can subsequently be changed by the MCU so that memory mapping can be changed on-the-fly.

For example, you may wish to have SRAM and primary Flash memory in the data space at Boot-up, and secondary Flash memory in the program space at Boot-up, and later swap the secondary Flash memory and primary Flash memory. This is easily done with the VM

register by using PSDsoft to configure it for Boot-up and having the MCU change it when desired.

*Table 25* describes the VM register.

**Figure 7. Priority level of memory and I/O components**




## 12.3 Configuration modes for MCUs with separate program and data spaces

### 12.3.1 Separate space modes

Program space is separated from data space. For example, Program Select Enable ($\overline{\text{PSEN}}$, CNTL2) is used to access the program code from the primary Flash memory, while READ Strobe ($\overline{\text{RD}}$, CNTL1) is used to access data from the secondary Flash memory, SRAM and I/O port blocks. This configuration requires the VM register to be set to 0Ch (see *Figure 8: 8031 memory modules – separate space*).

### 12.3.2 Combined space modes

The Program and data spaces are combined into one memory space that allows the primary Flash memory, secondary Flash memory, and SRAM to be accessed by either Program Select Enable ($\overline{\text{PSEN}}$, CNTL2) or READ Strobe ($\overline{\text{RD}}$, CNTL1). For example, to configure the primary Flash memory in Combined space, bits 2 and 4 of the VM register are set to 1 (see *Figure 9: 8031 memory modules – combined space*).

## 12.4 80C31 memory map example

See the application notes for examples.

**Figure 8. 8031 memory modules – separate space**

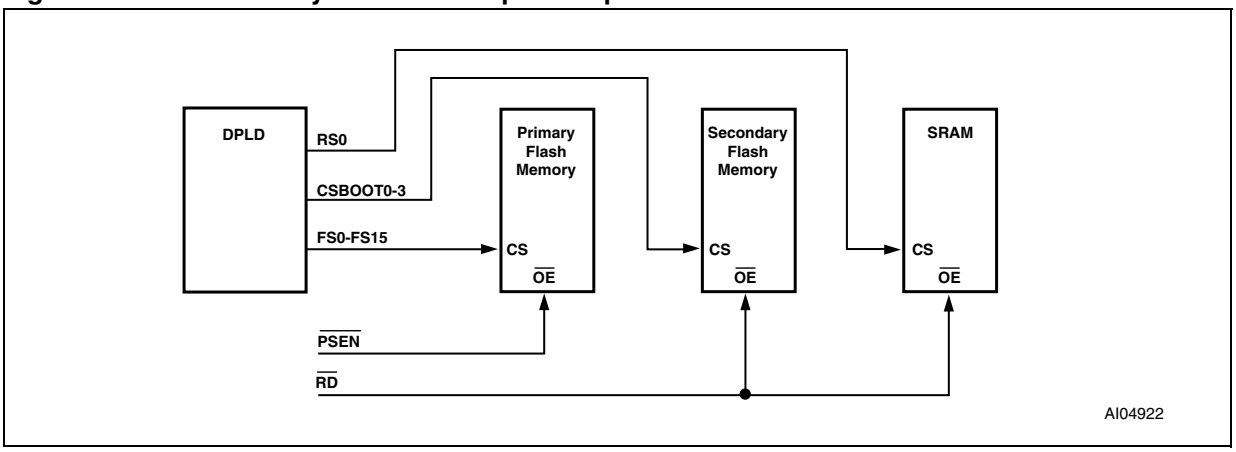
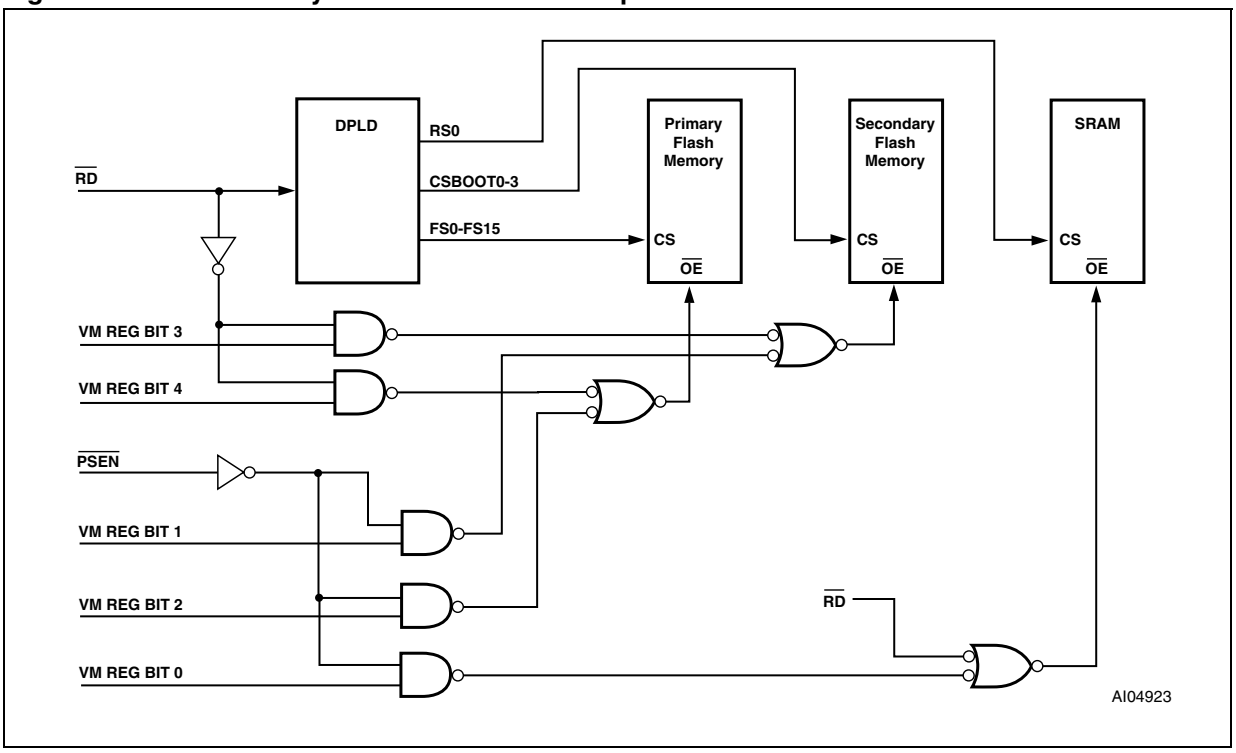



**Figure 9. 8031 memory modules – combined space**



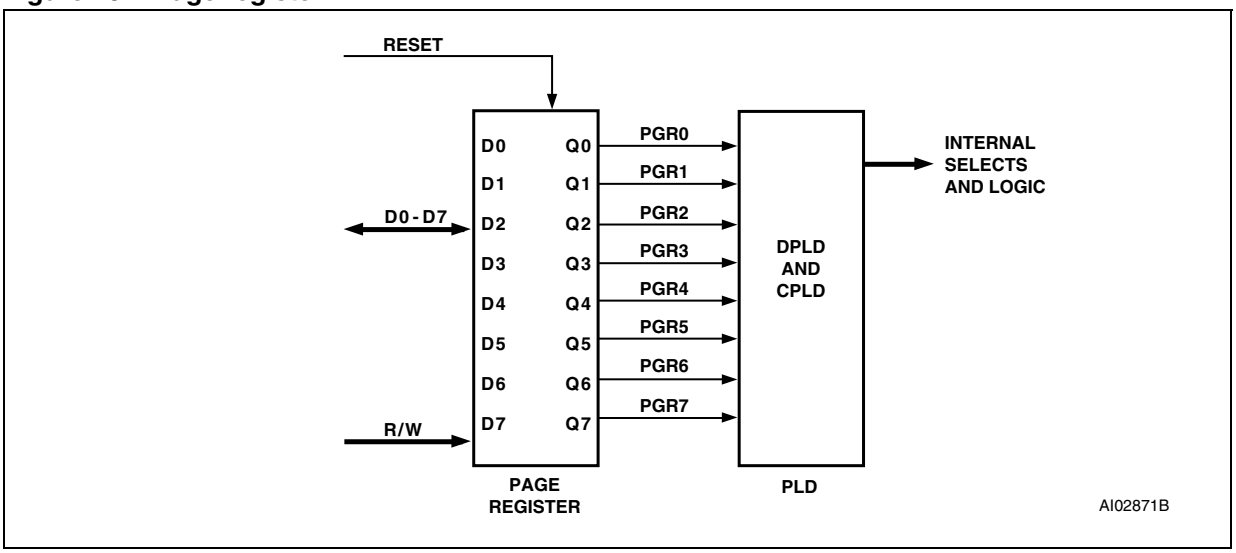
# 13 Page register

The 8-bit Page register increases the addressing capability of the MCU by a factor of up to 256. The contents of the register can also be read by the MCU. The outputs of the Page register (PGR0-PGR7) are inputs to the DPLD decoder and can be included in the Sector Select (FS0-FS15, CSBOOT0-CSBOOT3), and SRAM Select (RS0) equations.

If memory paging is not needed, or if not all eight page register bits are needed for memory paging, these bits may be used in the CPLD for general logic. See application note AN1154.

*Table 22* and *Figure 10* show the Page register. The eight flip-flops in the register are connected to the internal data bus (D0-D7). The MCU can write to or read from the Page register. The Page register can be accessed at address location CSIOP + E0h.

**Figure 10. Page register**



# 14 Memory ID registers

The 8-bit “read-only” memory status registers are included in the CSIOP space. The user can determine the memory configuration of the PSD device by reading the Memory ID0 and Memory ID1 registers. The content of the registers is defined as shown in *Table 26: Memory_ID0 register* and *Table 27: Memory_ID1 register*.

# 15 PLDs

The PLDs bring programmable logic functionality to the PSD. After specifying the logic for the PLDs using PSDsoft, the logic is programmed into the device and available upon Power-up.

The PSD contains two PLDs: the Decode PLD (DPLD), and the Complex PLD (CPLD). The PLDs are briefly discussed in the next few paragraphs, and in more detail in the following sections. *Figure 11: PLD diagram* shows the configuration of the PLDs.

The DPLD performs address decoding for internal components, such as memory, registers, and I/O ports Select signals.

The CPLD can be used for logic functions, such as loadable counters and shift registers, state machines, and encoding and decoding logic. These logic functions can be constructed using the 16 output macrocells (OMC), 24 input macrocells (IMC), and the AND Array. The CPLD can also be used to generate External Chip Select (ECS0-ECS2) signals.

The AND Array is used to form product terms. These product terms are specified using PSDsoft. An Input Bus consisting of 82 signals is connected to the PLDs. The signals are shown in *Table 32: DPLD and CPLD Inputs*

### The Turbo bit in PSD

The PLDs in the µPSD3200 Family can minimize power consumption by switching to standby when inputs remain unchanged for an extended time of about 70ns. Resetting the Turbo bit to '0' (bit 3 of the PMMR0 register) automatically places the PLDs into standby if no inputs are changing. Turning the Turbo mode off increases propagation delays while reducing power consumption. See *Section 20: Power management*, on how to set the Turbo bit.

Additionally, five bits are available in the PMMR2 register to block MCU control signals from entering the PLDs. This reduces power consumption and can be used only when these MCU control signals are not used in PLD logic equations.

Each of the two PLDs has unique characteristics suited for its applications. They are described in the following sections.

**Table 32. DPLD and CPLD Inputs**

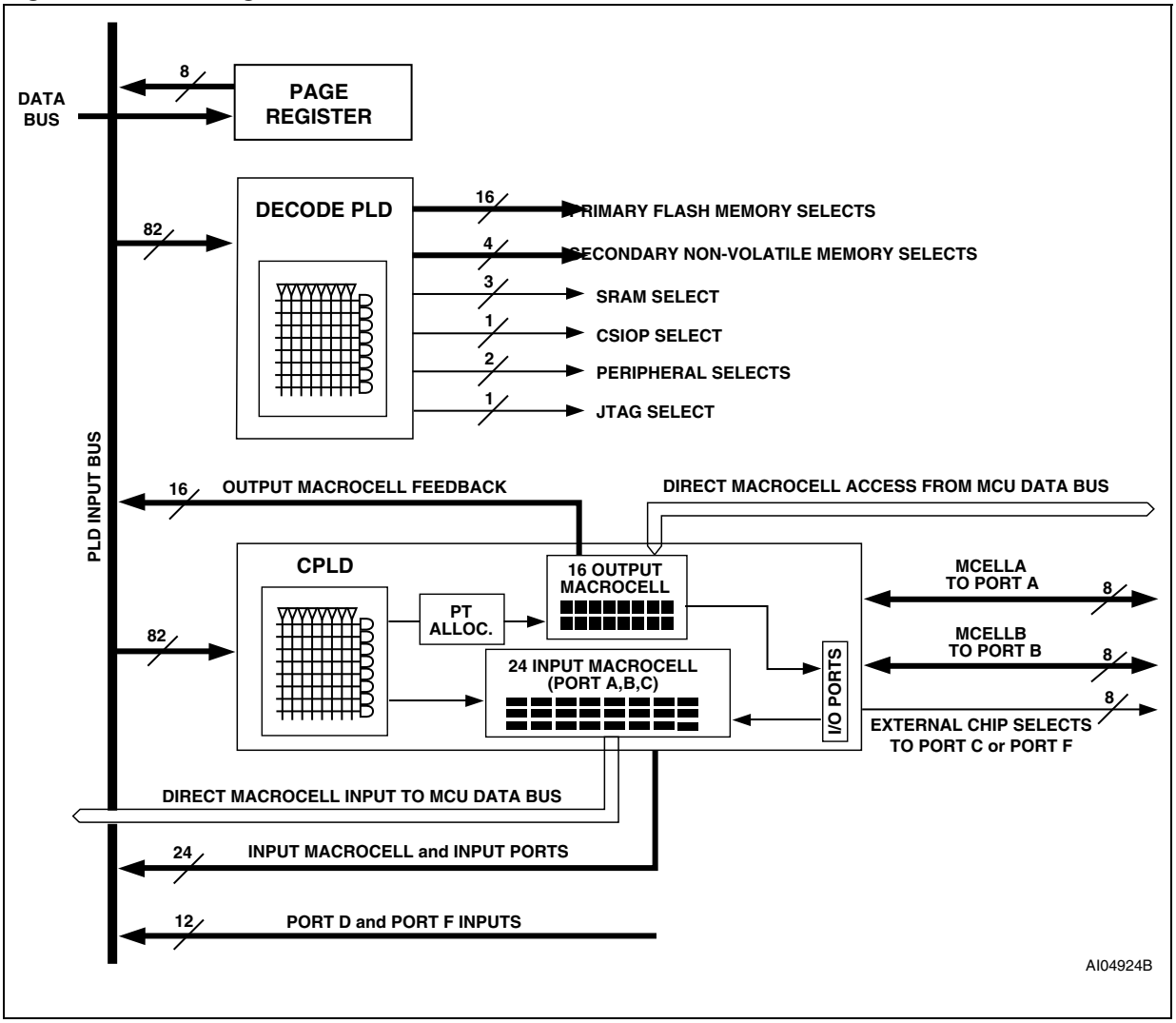
| Input source | Input name | Number of signals |
|---|---|---|
| MCU address bus(1) | A15-A0 | 16 |
| MCU control signals | CNTL0-CNTL2 | 3 |
| Reset | RST | 1 |
| Power-down | PDN | 1 |
| Port A input macrocells | PA7-PA0 | 8 |
| Port B input macrocells | PB7-PB0 | 8 |
| Port C input macrocells | PC7-PC0 | 8 |
| Port D inputs | PD3-PD0 | 4 |
| Port F inputs | PF7-PF0 | 8 |

**Table 32. DPLD and CPLD Inputs (continued)**

| Input source | Input name | Number of signals |
|---|---|---|
| Page register | PGR7-PGR0 | 8 |
| Macrocell A feedback | MCELLA.FB7-FB0 | 8 |
| Macrocell B feedback | MCELLB.FB7-FB0 | 8 |
| Flash memory Program Status bit | Ready/$\overline{\text{Busy}}$ | 1 |

1. The address inputs are A19-A4 in 80C51XA mode.

**Figure 11. PLD diagram**

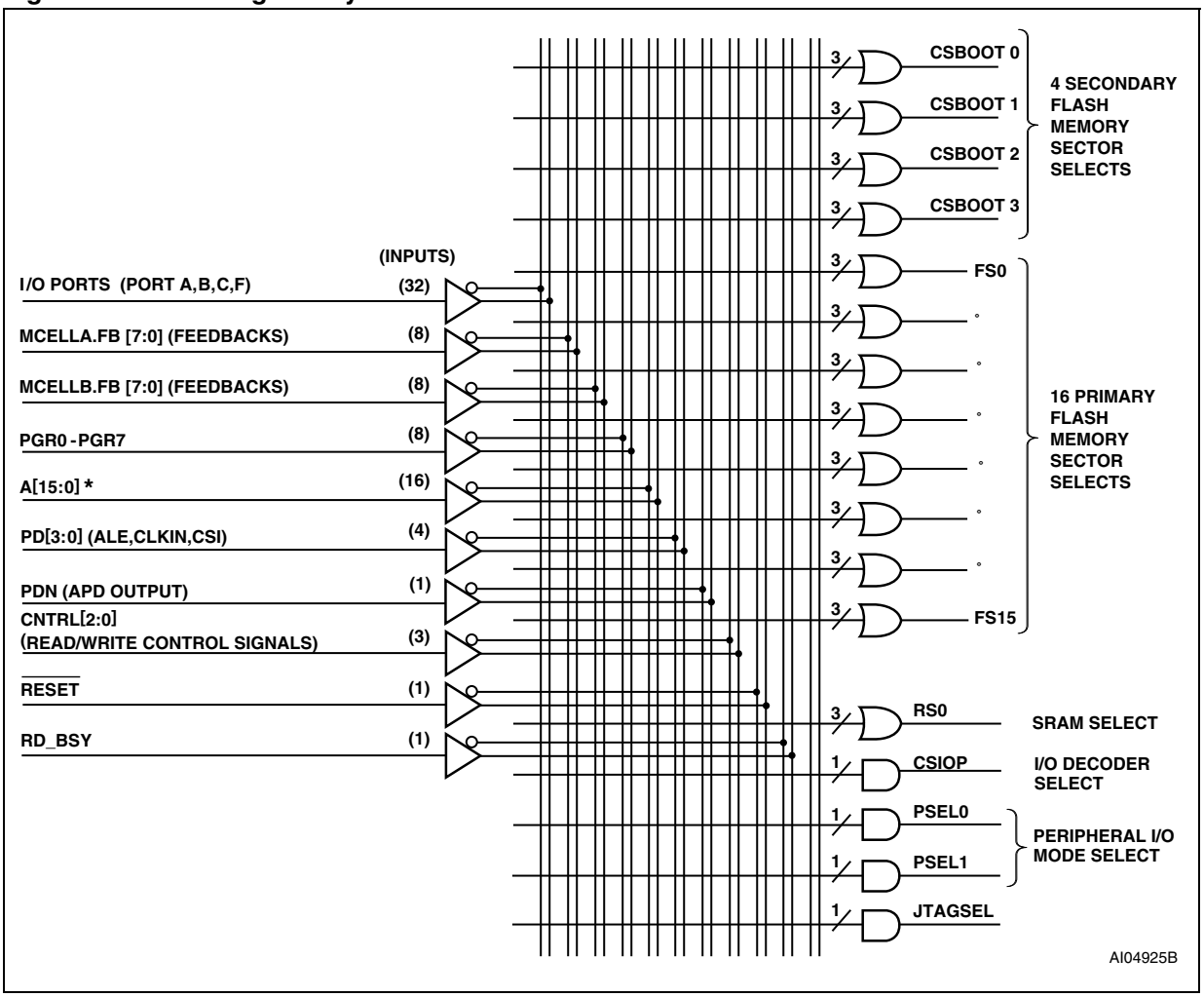


# 16 Decode PLD (DPLD)

The DPLD, shown in *Figure 12: DPLD logic array*, is used for decoding the address for internal and external components. The DPLD can be used to generate the following decode signals:

- 8 Sector Select (FS0-FS15) signals for the primary Flash memory (three product terms each)
- 4 Sector Select (CSBOOT0-CSBOOT3) signals for the secondary Flash memory (three product terms each)
- 1 internal SRAM Select (RS0) signal (three product terms)
- 1 internal CSIOP Select (PSD Configuration register) signal
- 1 JTAG Select signal (enables JTAG-ISP on Port E)
- 2 internal Peripheral Select signals (Peripheral I/O mode).

**Figure 12. DPLD logic array**




1. The address inputs are A19-A4 when in 80C51XA mode.
2. Additional address lines can be brought in the PSD via Port A, B, C, D, or F.

# 17 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 eight External Chip Select (ECS0-ECS7), routed to Port C or Port F.

Although External Chip Select (ECS0-ECS7) can be produced by any output macrocell (OMC), these eight External Chip Select (ECS0-ECS7) on Port C or Port F do not consume any output macrocells (OMC).

As shown in *Figure 13: Macrocell and I/O port*, the CPLD has the following blocks:

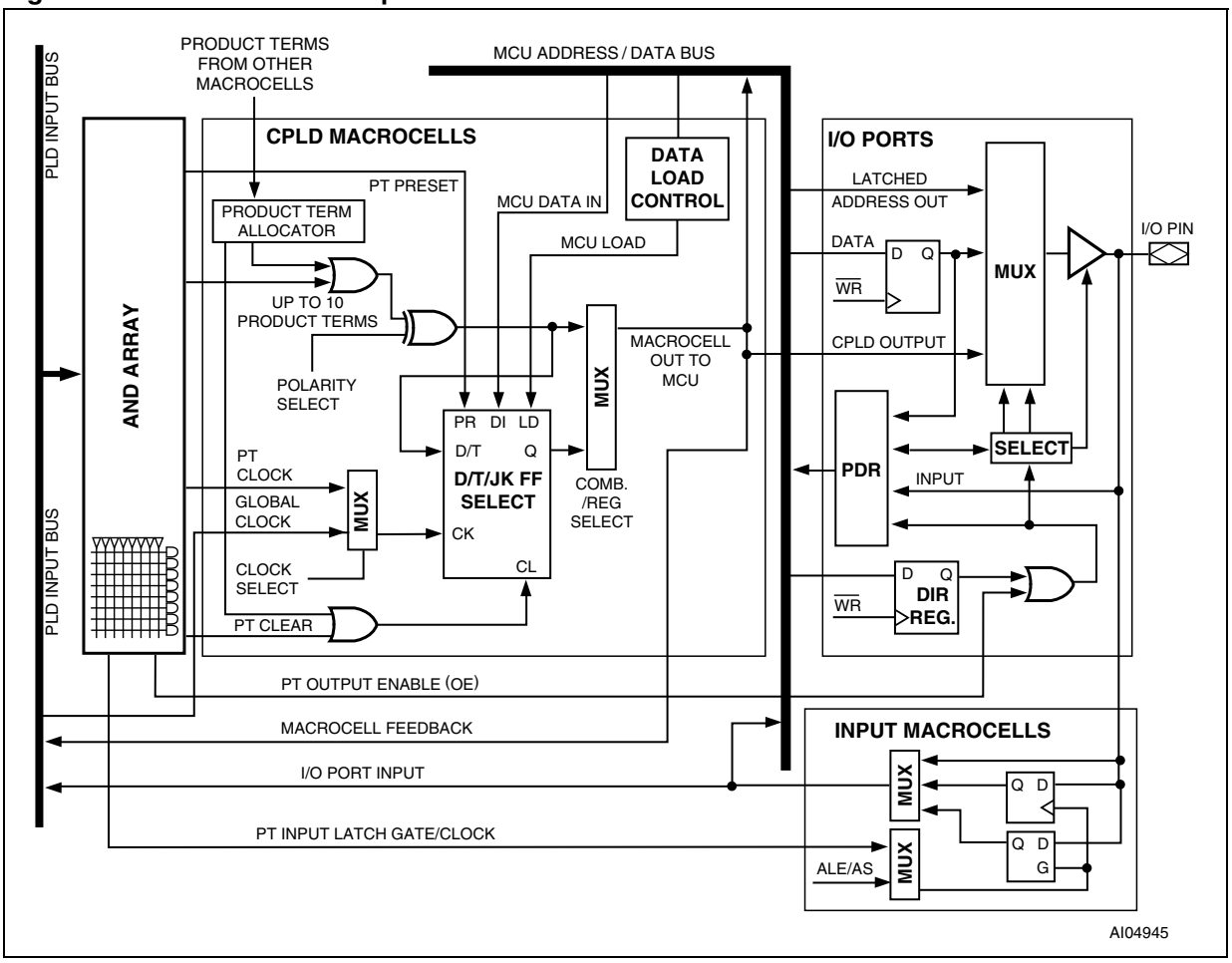
- 24 input macrocells (IMC)
- 16 output macrocells (OMC)
- Product Term Allocator
- AND Array capable of generating up to 196 product terms
- Four I/O Ports.

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

The input macrocells (IMC) and output macrocells (OMC) are connected to the PSD internal data bus and can be directly accessed by the MCU. This enables the MCU software to load data into the output macrocells (OMC) or read data from both the Input and output macrocells (IMC and OMC).

This feature allows efficient implementation of system logic and eliminates the need to connect the data bus to the AND Array as required in most standard PLD macrocell architectures.

**Figure 13. Macrocell and I/O port**




## 17.1 Output macrocell (OMC)

Eight of the output macrocells (OMC) are connected to Ports A pins and are named as McellA0-McellA7. The other eight macrocells are connected to Ports B pins and are named as McellB0-McellB7.

The output macrocells (OMC) architecture is shown in *Figure 14: CPLD output macrocell*. As shown in the figure, there are native product terms available from the AND Array, and borrowed product terms available (if unused) from other output macrocells (OMC). The polarity of the product term is controlled by the XOR gate. The output macrocells (OMC) can implement either sequential logic, using the flip-flop element, or combinatorial logic. The multiplexer selects between the sequential or combinatorial logic outputs. The multiplexer output can drive a port pin and has a feedback path to the AND Array inputs.

The flip-flop in the output macrocells (OMC) block can be configured as a D, T, JK, or SR type in the PSDsoft program. The flip-flop's clock, preset, and clear inputs may be driven from a product term of the AND Array. Alternatively, the external CLKIN (PD1) signal can be used for the clock input to the flip-flop. The flip-flop is clocked on the rising edge of CLKIN (PD1). The preset and clear are active High inputs. Each clear input can use up to two product terms.

**Table 33. Output macrocell port and data bit assignments**

| Output macrocell | Port assignment | Native product terms | Maximum borrowed product terms | 16-bit MCU loading or reading(1) | Motorola 16-bit MCU for loading or reading |
|---|---|---|---|---|---|
| McellA0 | Port A0 | 3 | 6 | D0 | D8 |
| McellA1 | Port A1 | 3 | 6 | D1 | D9 |
| McellA2 | Port A2 | 3 | 6 | D2 | D10 |
| McellA3 | Port A3 | 3 | 6 | D3 | D11 |
| McellA4 | Port A4 | 3 | 6 | D4 | D12 |
| McellA5 | Port A5 | 3 | 6 | D5 | D13 |
| McellA6 | Port A6 | 3 | 6 | D6 | D14 |
| McellA7 | Port A7 | 3 | 6 | D7 | D15 |
| McellB0 | Port B0 | 4 | 5 | D8 | D0 |
| McellB1 | Port B1 | 4 | 5 | D9 | D1 |
| McellB2 | Port B2 | 4 | 5 | D10 | D2 |
| McellB3 | Port B3 | 4 | 5 | D11 | D3 |
| McellB4 | Port B4 | 4 | 6 | D12 | D4 |
| McellB5 | Port B5 | 4 | 6 | D13 | D5 |
| McellB6 | Port B6 | 4 | 6 | D14 | D6 |
| McellB7 | Port B7 | 4 | 6 | D15 | D7 |

1. D7-D0 are used for loading or reading in 8-bit mode.

## 17.2 Product Term Allocator

The CPLD has a Product Term Allocator. PSDsoft, uses the Product Term Allocator to borrow and place product terms from one macrocell to another. The following list summarizes how product terms are allocated:

- McellA0-McellA7 all have three native product terms and may borrow up to six more
- McellB0-McellB3 all have four native product terms and may borrow up to five more
- McellB4-McellB7 all have four native product terms and may borrow up to six more.

Each macrocell may only borrow product terms from certain other macrocells. Product terms already in use by one macrocell are not available for another macrocell.

If an equation requires more product terms than are available to it, then “external” product terms are required, which consume other output macrocells (OMC). If external product terms are used, extra delay is added for the equation that required the extra product terms. This is called product term expansion. PSDsoft performs this expansion as needed.

## 17.3 Loading and reading the output macrocells (OMC)

The output macrocells (OMC) block occupies a memory location in the MCU address space, as defined by the CSIOP (see *Figure 20* to *Figure 30* for examples of the basic connections between the PSD and some popular MCUs). The PSD Control input pins are labeled as to the MCU function for which they are configured. The MCU bus interface is specified using the PSDsoft Express Configuration. The flip-flops in each of the 16 output macrocells (OMC) can be loaded from the data bus by a MCU. Loading the output macrocells (OMC) with data from the MCU takes priority over internal functions. As such, the preset, clear, and clock inputs to the flip-flop can be overridden by the MCU. The ability to load the flip-flops and read them back is useful in such applications as loadable counters and shift registers, mailboxes, and handshaking protocols.

Data is loaded to the output macrocells (OMC) on the trailing edge of WRITE Strobe ($\overline{WR}$/$\overline{WRL}$, CNTL0).

## 17.4 The OMC Mask register

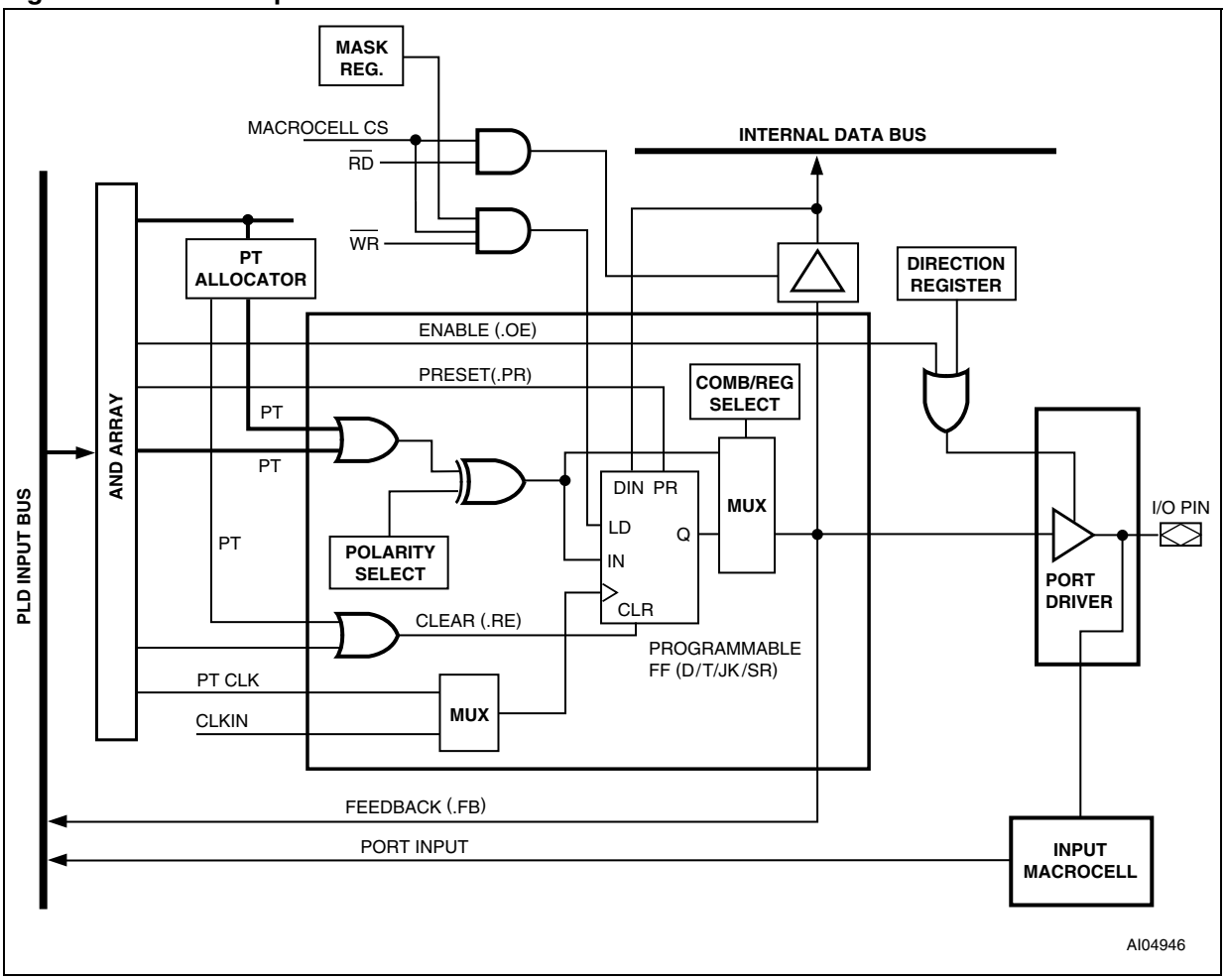
There is one Mask register for each of the two groups of eight output macrocells (OMC). The Mask registers can be used to block the loading of data to individual output macrocells (OMC). The default value for the Mask registers is 00h, which allows loading of the output macrocells (OMC). When a given bit in a Mask register is set to a 1, the MCU is blocked from writing to the associated output macrocells (OMC). For example, suppose McellA0-McellA3 are being used for a state machine. You would not want a MCU WRITE to McellA to overwrite the state machine registers. Therefore, you would want to load the Mask register for McellA (Mask Macrocell A) with the value 0Fh.

## 17.5 The Output Enable of the OMC

The output macrocells (OMC) can be connected to an I/O port pin as a PLD output. The output enable of each port pin driver is controlled by a single product term from the AND Array, ORed with the Direction register output. The pin is enabled upon Power-up if no output enable equation is defined and if the pin is declared as a PLD output in PSDsoft.

If the output macrocell (OMC) output is declared as an internal node and not as a port pin output in the PSDabel file, then the port pin can be used for other I/O functions. The internal node feedback can be routed as an input to the AND array.

**Figure 14. CPLD output macrocell**




## 17.6 Input macrocells (IMC)

The CPLD has 24 input macrocells (IMC), one for each pin on Ports A, B, and C. The architecture of the input macrocells (IMC) is shown in *Figure 15: Input macrocell*. The input macrocells (IMC) are individually configurable, and can be used as a latch, register, or to pass incoming port signals prior to driving them onto the PLD input bus. The outputs of the input macrocells (IMC) can be read by the MCU through the internal data bus.

The enable for the latch and clock for the register are driven by a multiplexer whose inputs are a product term from the CPLD AND Array or the MCU Address Strobe (ALE/AS). Each product term output is used to latch or clock four input macrocells (IMC). Port inputs 3-0 can be controlled by one product term and 7-4 by another.

Configurations for the input macrocells (IMC) are specified by PSDsoft (see Application Note *AN1171*). Outputs of the input macrocells (IMC) can be read by the MCU via the IMC buffer. See *Figure 20* to *Figure 30* for examples of the basic connections between the PSD and some popular MCUs. The PSD Control input pins are labeled as to the MCU function for which they are configured. The MCU bus interface is specified using the I/O ports. (See *Section 2.3: I/O ports*.)

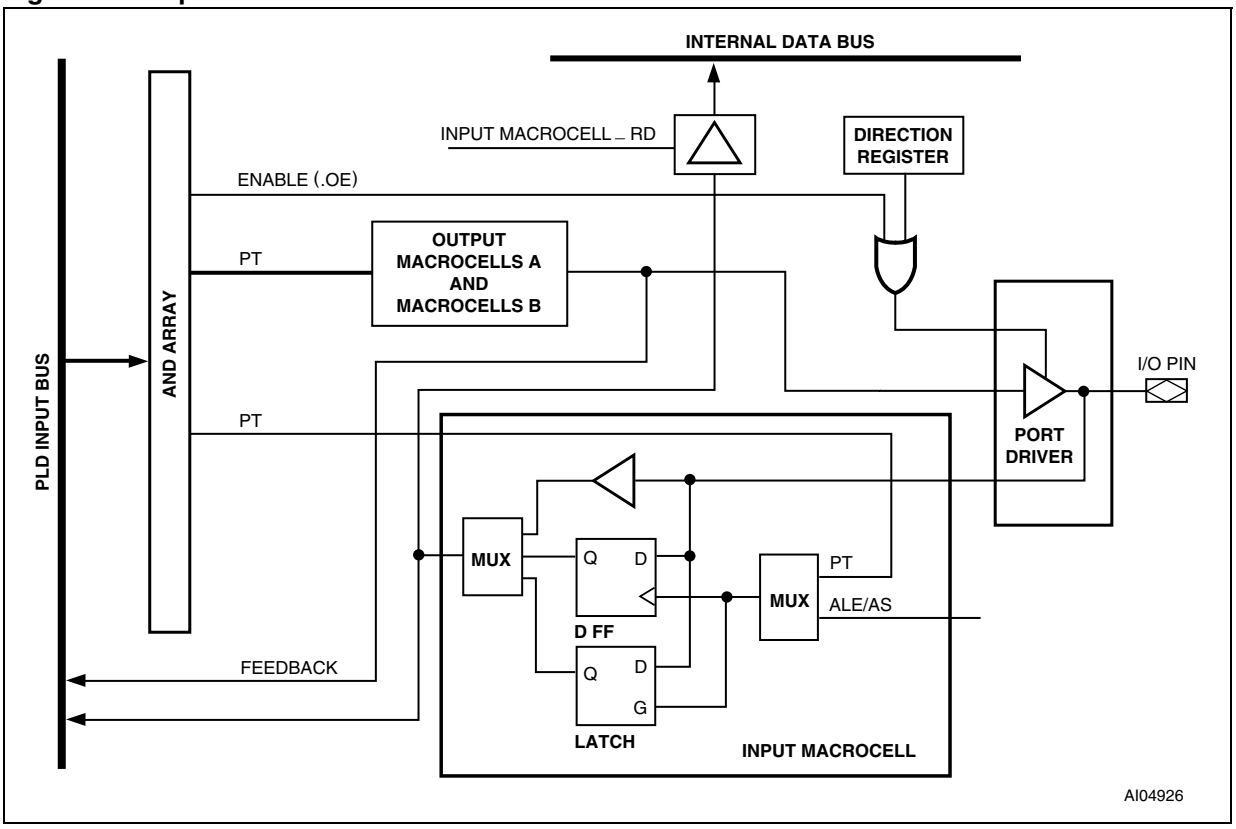
Input macrocells (IMC) can use Address Strobe (ALE/AS, PD0) to latch address bits higher than A15. Any latched addresses are routed to the PLDs as inputs.

Input macrocells (IMC) are particularly useful with handshaking communication applications where two processors pass data back and forth through a common mailbox. *Figure 17: Handshaking communication using input macrocells* shows a typical configuration where the Master MCU writes to the Port A Data Out register. This, in turn, can be read by the Slave MCU via the activation of the “Slave-READ” output enable product term.

The Slave can also write to the Port A input macrocells (IMC) and the Master can then read the input macrocells (IMC) directly.

Note that the “Slave-READ” and “Slave-Wr” signals are product terms that are derived from the Slave MCU inputs READ Strobe ($\overline{RD}$, CNTL1), WRITE Strobe ($\overline{WR}$/$\overline{WRL}$, CNTL0), and Slave_CS.

**Figure 15. Input macrocell**




## 17.7 External Chip Select

The CPLD also provides eight External Chip Select (ECS0-ECS7) outputs that can be used to select external devices. Each External Chip Select (ECS0-ECS7) consists of one product term that can be configured active High or Low.

The output enable of the pin is controlled by either the output enable product term or the Direction register. (See *Figure 16: External Chip Select signal*.)

**Figure 16. External Chip Select signal**

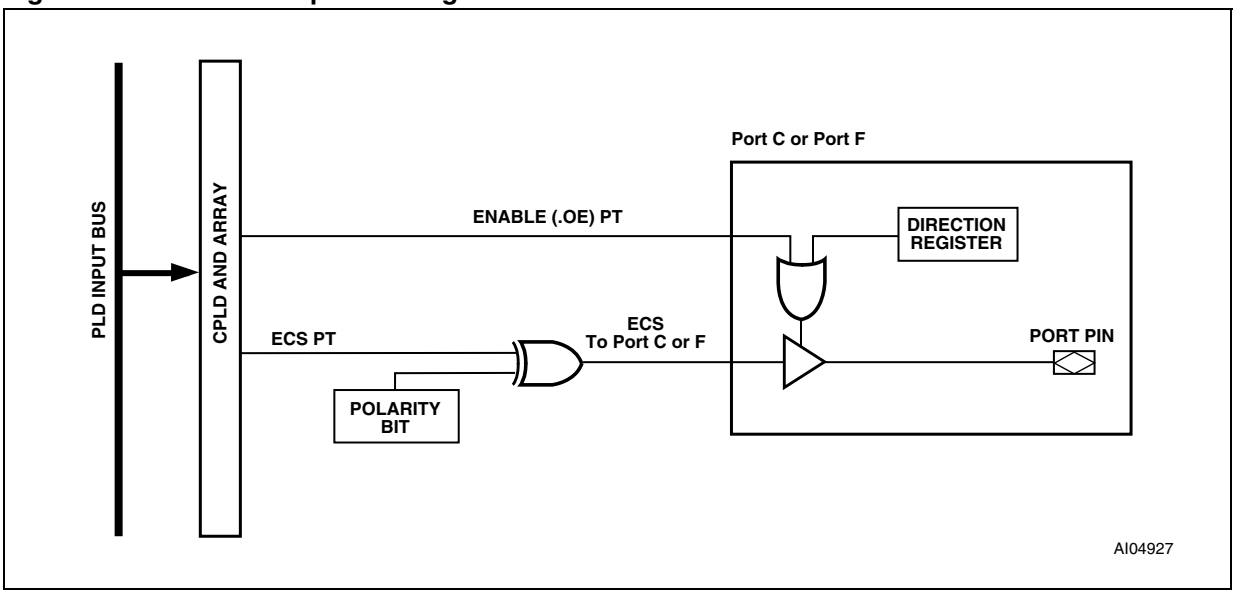
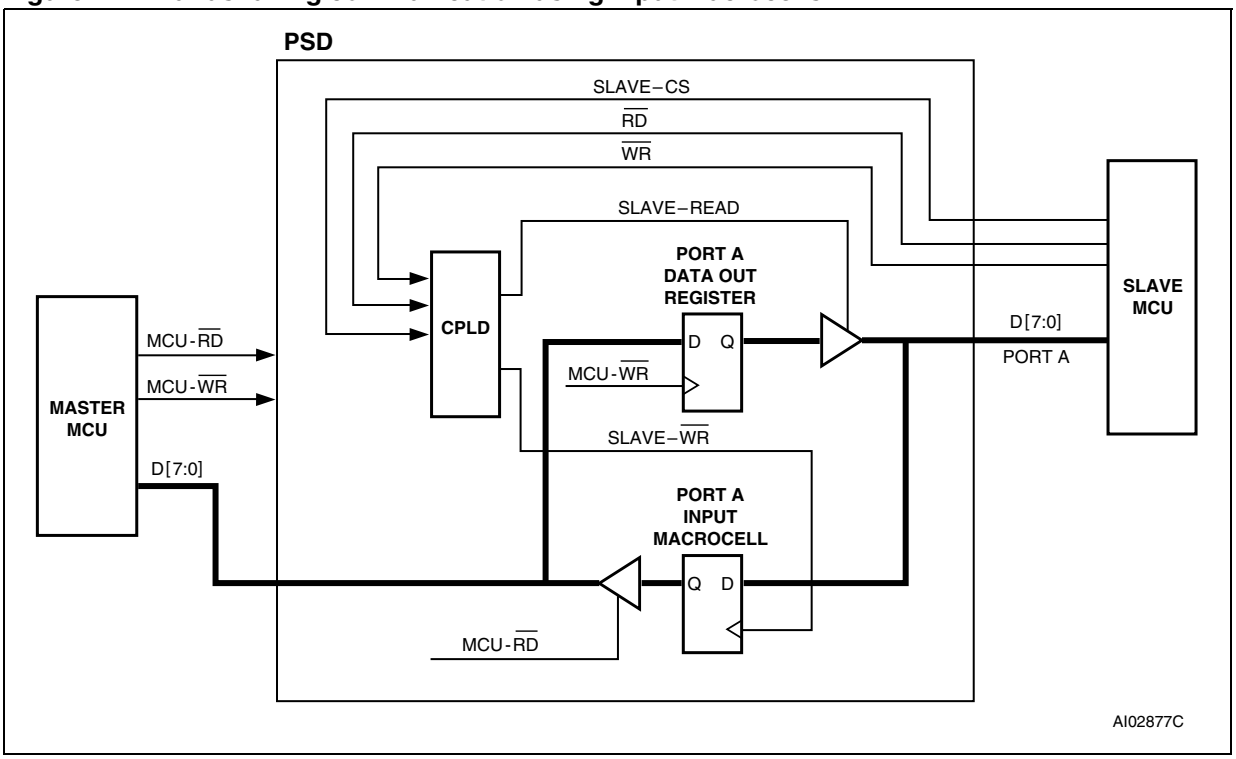



**Figure 17. Handshaking communication using input macrocells**



# 18 MCU bus interface

The “no-glue logic” MCU bus interface block can be directly connected to most popular 8-bit and 16-bit MCUs and their control signals. Key MCUs, with their bus types and control signals, are shown in *Table 34* and *Table 35*. The MCU interface type is specified using the PSDsoft.

**Table 34. 16-bit MCUs and their control signals**

| MCU | CNTL0 | CNTL1 | CNTL2 | PD3 | PD0(1) | ADIO0 | PF3-PF0 |
|---|---|---|---|---|---|---|---|
| 68302, 68306, MMC2001 | R/$\overline{W}$ | LDS | UDS | (2) | AS | — | (2) |
| 68330, 68331, 68332, 68340 | R/$\overline{W}$ | DS | SIZ0 | (2) | AS | A0 | (2) |
| 68LC302, MMC2001 | WEL | OE | — | WEH | AS | — | (2) |
| 68HC16 | R/$\overline{W}$ | DS | SIZ0 | (2) | AS | A0 | (2) |
| 68HC912 | R/$\overline{W}$ | E | LSTRB | DBE | E | A0 | (2) |
| 68HC812 (3) | R/$\overline{W}$ | E | LSTRB | (2) | (2) | A0 | (2) |
| 80196 | WR | RD | BHE | (2) | ALE | A0 | (2) |
| 80196SP | WRL | RD | (2) | WRH | ALE | A0 | (2) |
| 80186 | WR | RD | BHE | (2) | ALE | A0 | (2) |
| 80C161, 80C164-80C167 | WR | RD | BHE | (2) | ALE | A0 | (2) |
| 80C51XA | WRL | RD | PSEN | WRH | ALE | A4/D0 | A3-A1 |
| H8/300 | WRL | RD | (2) | WRH | AS | A0 | — |

1. ALE/AS input is optional for MCUs with a non-multiplexed bus.
2. Unused CNTL2 pin can be configured as CPLD input. Other unused pins (PD3-PD0, PF3-PF0) can be configured for other I/O functions.
3. This configuration is for MC68HC812A4_EC at 5MHz, 3 V only.

**Table 35. 8-bit MCUs and their control signals**

| MCU | CNTL0 | CNTL1 | CNTL2 | PD7 | PD0(1) | ADIO0 | PF3-PF0 |
|---|---|---|---|---|---|---|---|
| 8031/8051 | WR | RD | PSEN | (2) | ALE | A0 | (2) |
| 80C51XA | WR | RD | PSEN | (2) | ALE | A4 | A3-A0 |
| 80C251 | WR | PSEN | (2) | (2) | ALE | A0 | (2) |
| 80C251 | WR | RD | PSEN | (2) | ALE | A0 | (2) |
| 80198 | WR | RD | (2) | (2) | ALE | A0 | (2) |
| 68HC11 | R/$\overline{W}$ | E | (2) | (2) | AS | A0 | (2) |
| 68HC05C0 | WR | RD | (2) | (2) | AS | A0 | (2) |
| 68HC912 | R/W | E | (2) | DBE | AS | A0 | (2) |
| Z80 | WR | RD | (2) | (2) | (2) | A0 | (2) |
| Z8 | R/W | DS | (2) | (2) | AS | A0 | (2) |
| 68330 | R/W | DS | (2) | (2) | AS | A0 | A3-A1 |

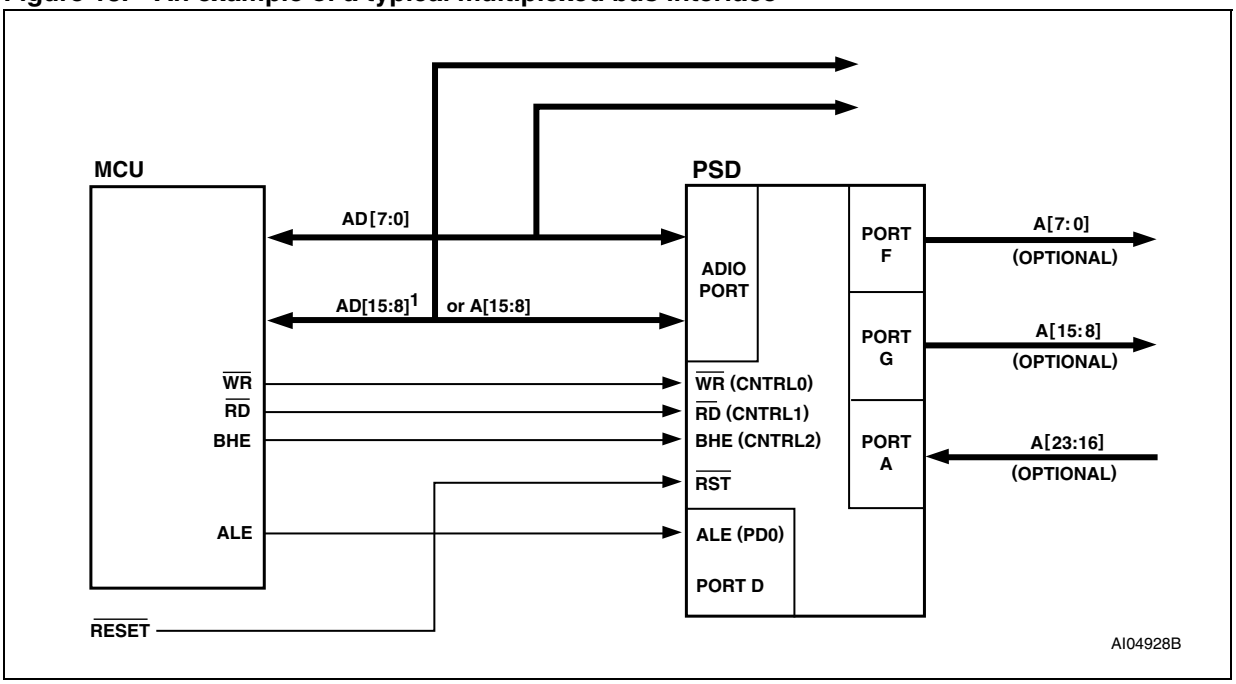
1. ALE/AS input is optional for MCUs with a non-multiplexed bus.

2. Unused CNTL2 pin can be configured as CPLD input. Other unused pins (PD3-PD0, PF3-PF0) can be configured for other I/O functions.

## 18.1 PSD interface to a multiplexed bus

*Figure 18* shows an example of a system using an MCU with a multiplexed bus and a PSD4256G6V. The ADIO port on the PSD is connected directly to the MCU address/data bus. Address Strobe (ALE/AS, PD0) latches the address signals internally. Latched addresses can be brought out to Port E, F or G. The PSD drives the ADIO data bus only when one of its internal resources is accessed and READ Strobe ($\overline{RD}$, CNTL1) is active. Should the system address bus exceed sixteen bits, Ports A, B, C, or F may be used as additional address inputs.

**Figure 18. An example of a typical multiplexed bus interface**

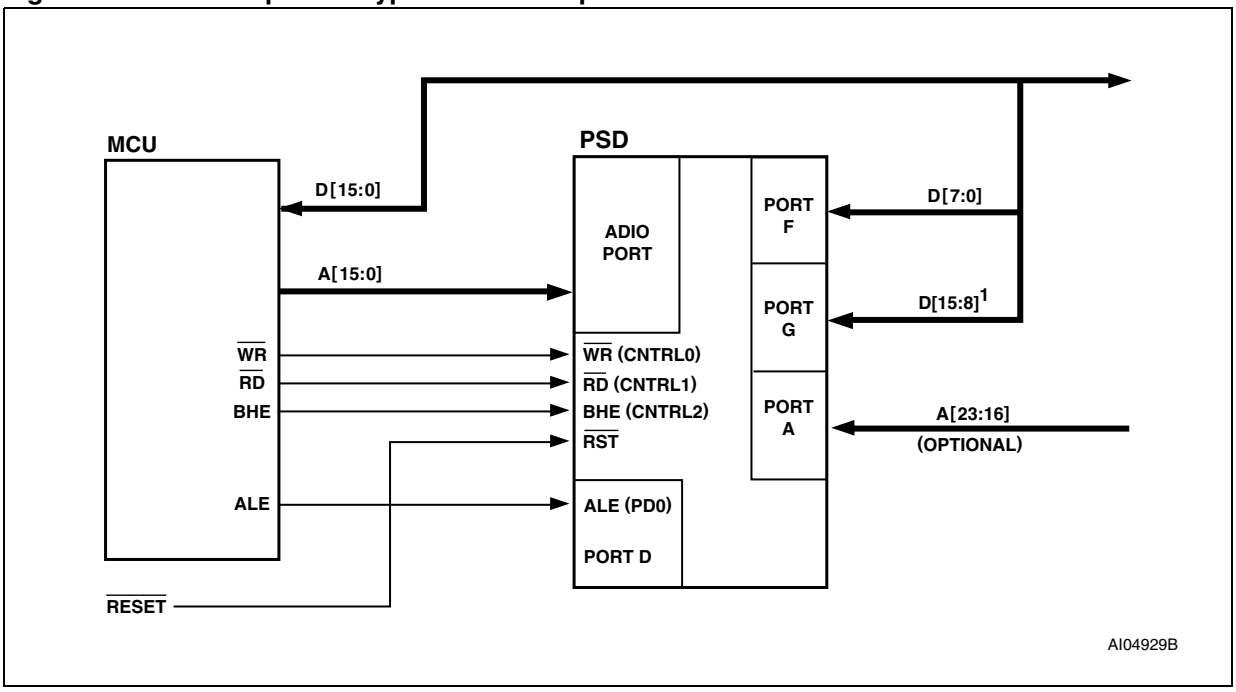



1. AD[15:8] is for 16-bit MCU

## 18.2 PSD interface to a non-multiplexed, 16-bit bus

*Figure 19* shows an example of a system using an MCU with a 16-bit, non-multiplexed bus and a PSD4256G6V. The address bus is connected to the ADIO port, and the data bus is connected to Ports F and G. Ports F and G are in tri-state mode when the PSD is not accessed by the MCU. Should the system address bus exceed sixteen bit, Ports A, B, or C may be used for additional address inputs.

**Figure 19. An example of a typical non-multiplexed bus interface**




1. D[15:8] is for 16-bit MCU

## 18.3 Data byte enable reference for a 16-bit bus

MCUs have different data byte orientations. *Table 36* to *Table 39* show how the µPSD3200 Family interprets byte/word operations in different bus WRITE configurations. Even-byte refers to locations with address A0 equal to 0, and odd byte as locations with A0 equal to 1.

**Table 36. 16-bit data bus with BHE**

| BHE | A0 | D15-D8 | D7-D0 |
|---|---|---|---|
| 0 | 0 | Odd byte | Even byte |
| 0 | 1 | Odd byte | — |
| 1 | 0 | — | Even byte |

## 18.4 16-bit MCU bus interface examples

*Figure 20* to *Figure 25* show examples of the basic connections between the µPSD3200 Family and some popular MCUs. The µPSD3200 Family Control input pins are labeled as to the MCU function for which they are configured. The MCU bus interface is specified using PSDsoft.

**Table 37. 16-bit data bus with WRH and WRL**

| WRH | WRL | D15-D8 | D7-D0 |
|---|---|---|---|
| 0 | 0 | Odd byte | Even byte |
| 0 | 1 | Odd byte | — |
| 1 | 0 | — | Even byte |

**Table 38. 16-bit data bus with SIZ0, A0 (Motorola MCU)**

| SIZ0 | A0 | D15-D8 | D7-D0 |
|---|---|---|---|
| 0 | 0 | Even byte | Odd byte |
| 1 | 0 | Even byte | — |
| 1 | 1 | — | Odd byte |

**Table 39. 16-bit data bus with LDS, UDS (Motorola MCU)**

| WRH | WRL | D15-D8 | D7-D0 |
|---|---|---|---|
| 0 | 0 | Even byte | Odd byte |
| 1 | 0 | Even byte | — |
| 0 | 1 | — | Odd byte |

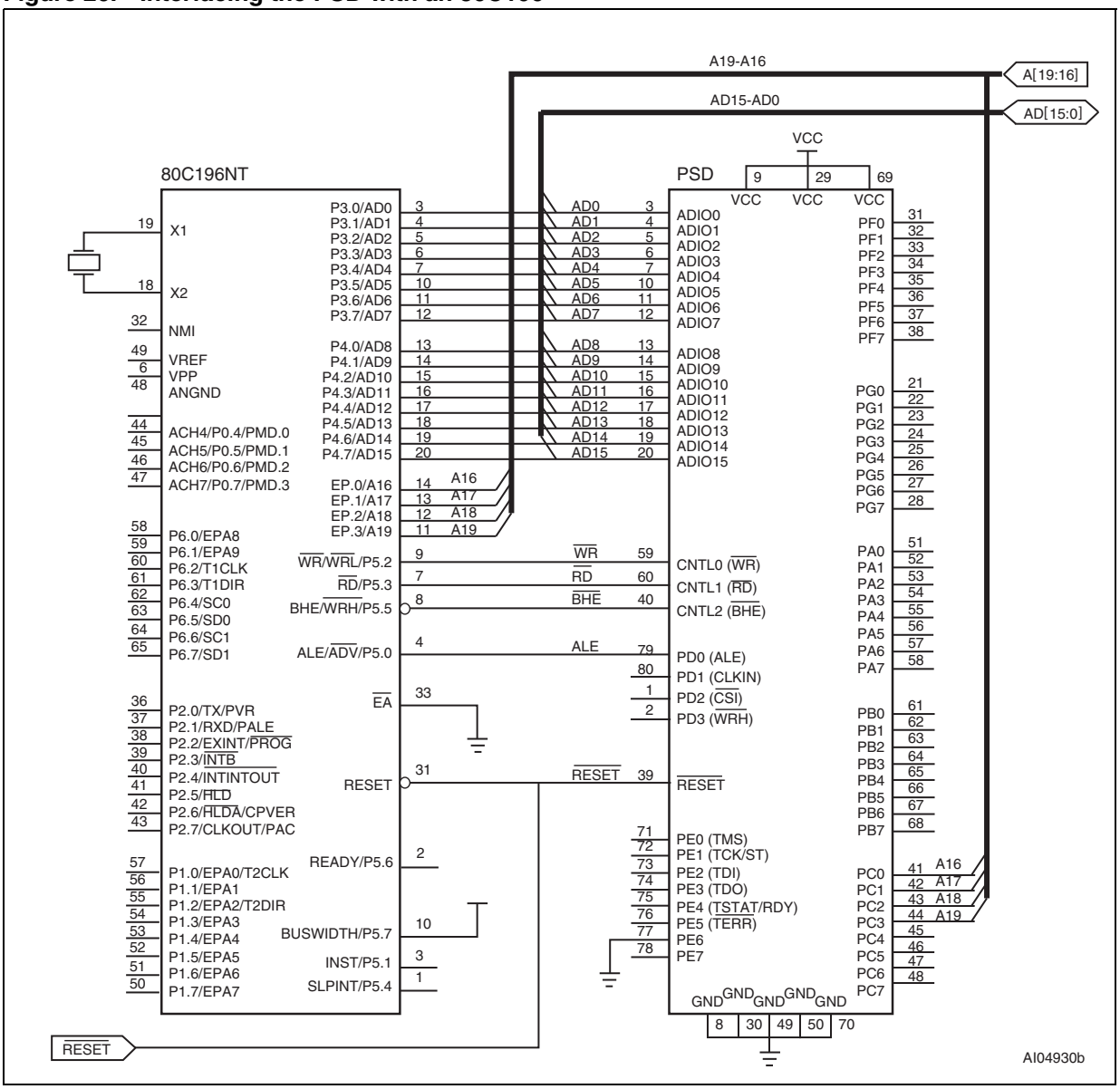
## 18.5 80C196 and 80C186

In *Figure 20*, the Intel 80C196 MCU, which has a 16-bit multiplexed address/data bus, is shown connected to a µPSD3200 Family. The READ Strobe ($\overline{RD}$, CNTL1), and WRITE Strobe ($\overline{WR}$/$\overline{WRL}$, CNTL0) signals are connected to the CNTL pins.

When BHE is not used, the PSD can be configured to receive WRL and WRITE Enable High-byte ($\overline{WRH}$/DBE, PD3) from the MCU. Higher address inputs (A16-A19) can be routed to Ports A, B, or C as input to the PLD.

The AMD 80186 family has the same bus connection to the PSD as the 80C196.

**Figure 20. Interfacing the PSD with an 80C196**



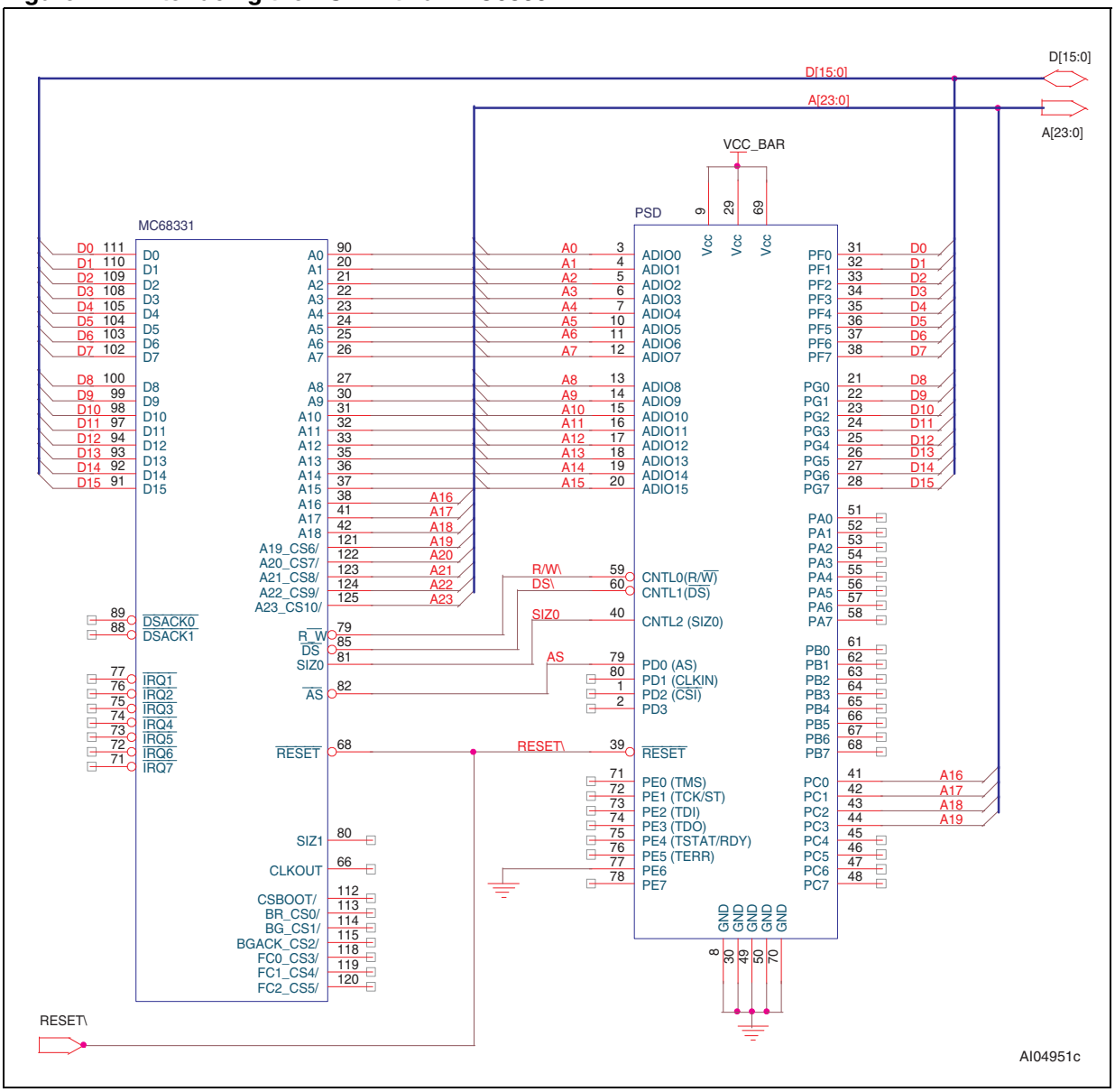
## 18.6 MC683xx and MC68HC16

*Figure 21* shows a MC68331 with a 16-bit non-multiplexed data bus and 24-bit address bus. The data bus from the MC68331 is connected to Port F (D0-D7) and Port G (D8-D15). The SIZ0 and A0 inputs determine the high/low byte selection.

The R/W, DS and SIZ0 signals are connected to the CNTL0-CNTL2 pins.

The MC68HC16, and other members of the MC683xx family, has the same bus connection to the PSD as the MC68331 shown in *Figure 21*

**Figure 21. Interfacing the PSD with an MC68331**



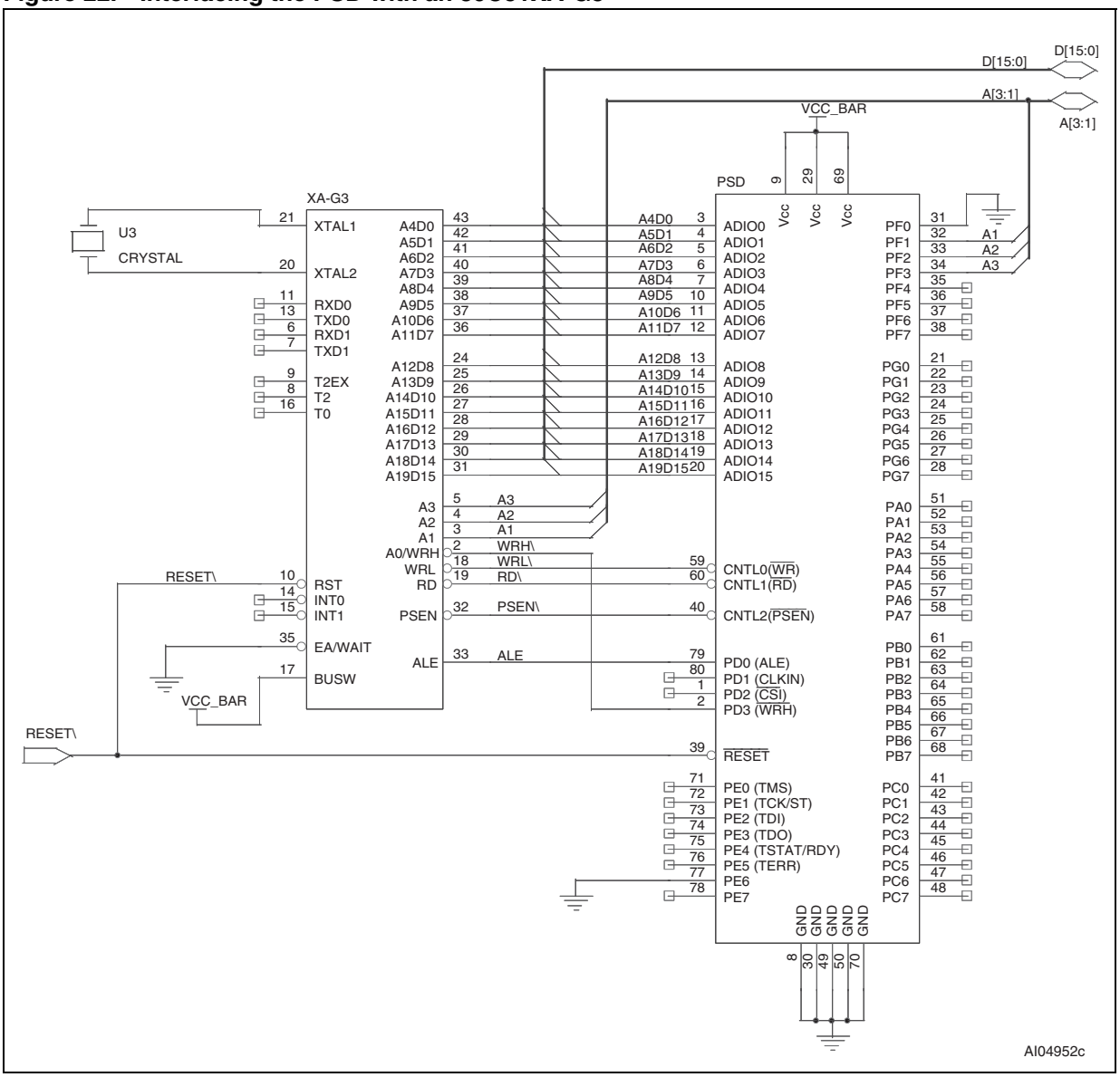
## 18.7 80C51XA

The Philips 80C51XA MCU has a 16-bit multiplexed bus with burst cycles. Address bits (A3-A1) are not multiplexed, while (A19-A4) are multiplexed with data bits (D15-D0).

The µPSD3200 Family supports the 80C51XA burst mode. The $\overline{WRH}$ signal is connected to PD3, and $\overline{WHL}$ is connected to CNTL0. The $\overline{RD}$ and $\overline{PSEN}$ signals are connected to the CNTL1 and CNTL2 pins. *Figure 22* shows the schematic diagram.

The 80C51XA improves bus throughput and performance by issuing burst cycles to retrieve codes from memory. In burst cycles, address A19-A4 are latched internally by the PSD, while the 80C51XA drives the A3-A1 signals to retrieve sequentially up to 16 bytes of code. The PSD access time is then measured from address A3-A1 valid to data in valid. The PSD bus timing requirement in a burst cycle is identical to the normal bus cycle, except the address setup and hold time with respect to Address Strobe (ALE/AS, PD0) is not required.

**Figure 22. Interfacing the PSD with an 80C51XA-G3**

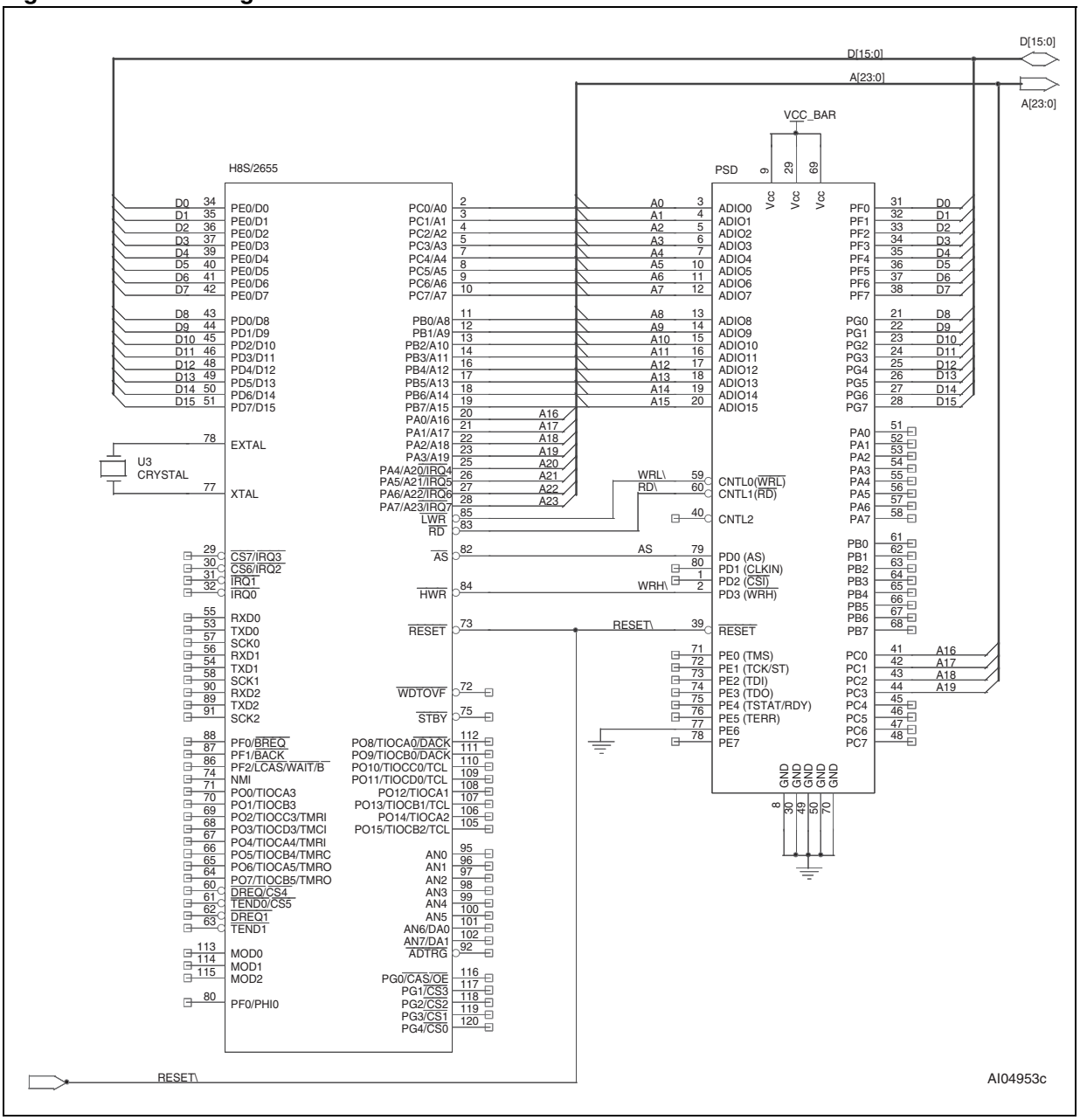


## 18.8 H8/300

*Figure 23* shows an Hitachi H8/2350 with a 16-bit non-multiplexed data bus, and a 24-bit address bus. The H8 data bus is connected to Port F (D0-D7) and Port G (D8-D15).

The $\overline{WRH}$ signal is connected to PD3, and $\overline{WHL}$ is connected to CNTL0. The $\overline{RD}$ signal is connected to CNTL1. The connection to the Address Strobe (AS) signal is optional, and is required if the addresses are to be latched.

**Figure 23. Interfacing the PSD with an H83/2350**



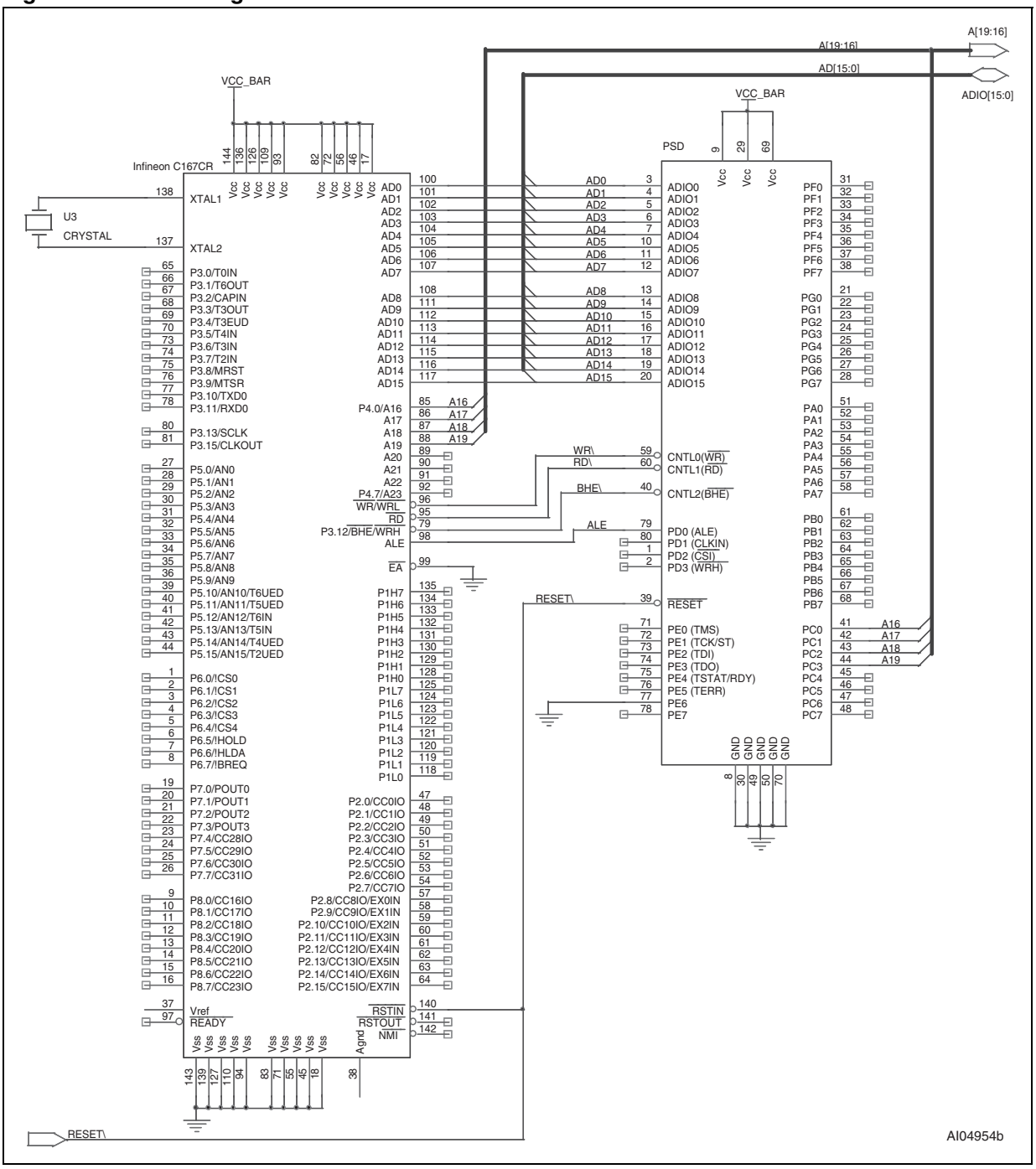
## 18.9 MMC2001

The Motorola MCORE MMC2001 MCU has a MOD input pin that selects internal or external boot ROM. The PSD can be configured as the external Flash boot ROM or as extension to the internal ROM (see *Figure 24*).

The MMC2001 has a 16-bit external data bus and 20 address lines with external chip select signals. The Chip Select Control registers allow the user to customize the bus interface and timing to fit the individual system requirement. A typical interface configuration to the PSD is shown in *Figure 24*. The MMC2001's R/$\overline{W}$ signal is connected to the CNTL0 pin, while $\overline{EB0}$ and $\overline{EB1}$ (enable byte-0 and enable byte-1) are connected to the CNTL1 ($\overline{UDS}$) and CNTL2 ($\overline{LDS}$) pins. The WEN bit in the Chip Select Control register should be set to 1 to terminate the $\overline{EB0}$-$\overline{EB1}$ earlier to provide the write data hold time for the PSD. The WSC and WWS bits in the Control register are set to wait states that meet the PSD access time requirement.

Another option is to configure the $\overline{EB0}$ and $\overline{EB1}$ as $\overline{WRL}$ and $\overline{WRH}$ signals. In this case, the PSD control setting will be: $\overline{OE}$, $\overline{WRL}$, $\overline{WRH}$ where $\overline{OE}$ is the READ signal for the MMC2001.

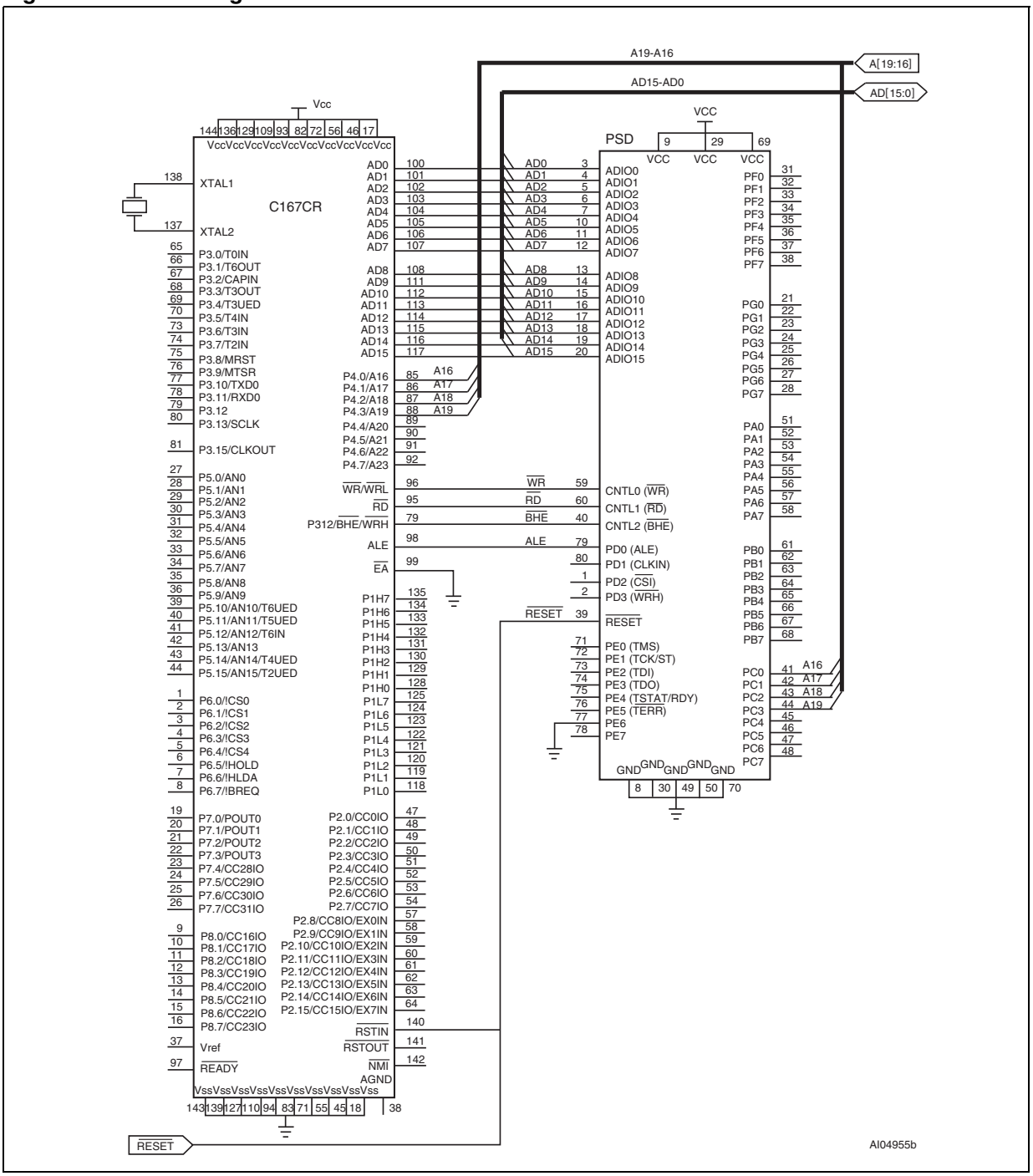
## Figure 24. Interfacing the PSD with an MMC2001



## 18.10 C16x family

The PSD supports Infineon's C16X family of MCUs (C161-C167) in both the multiplexed and non-multiplexed bus configuration. In *Figure 25*, the C167CR is shown connected to the PSD in a multiplexed bus configuration. The control signals from the MCU are $\overline{WR}$, $\overline{RD}$, $\overline{BHE}$ and ALE, and are routed to the corresponding PSD pins. The C167 has another control signal setting ($\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$, ALE) which is also supported by the PSD.

**Figure 25. Interfacing the PSD with a C167CR**



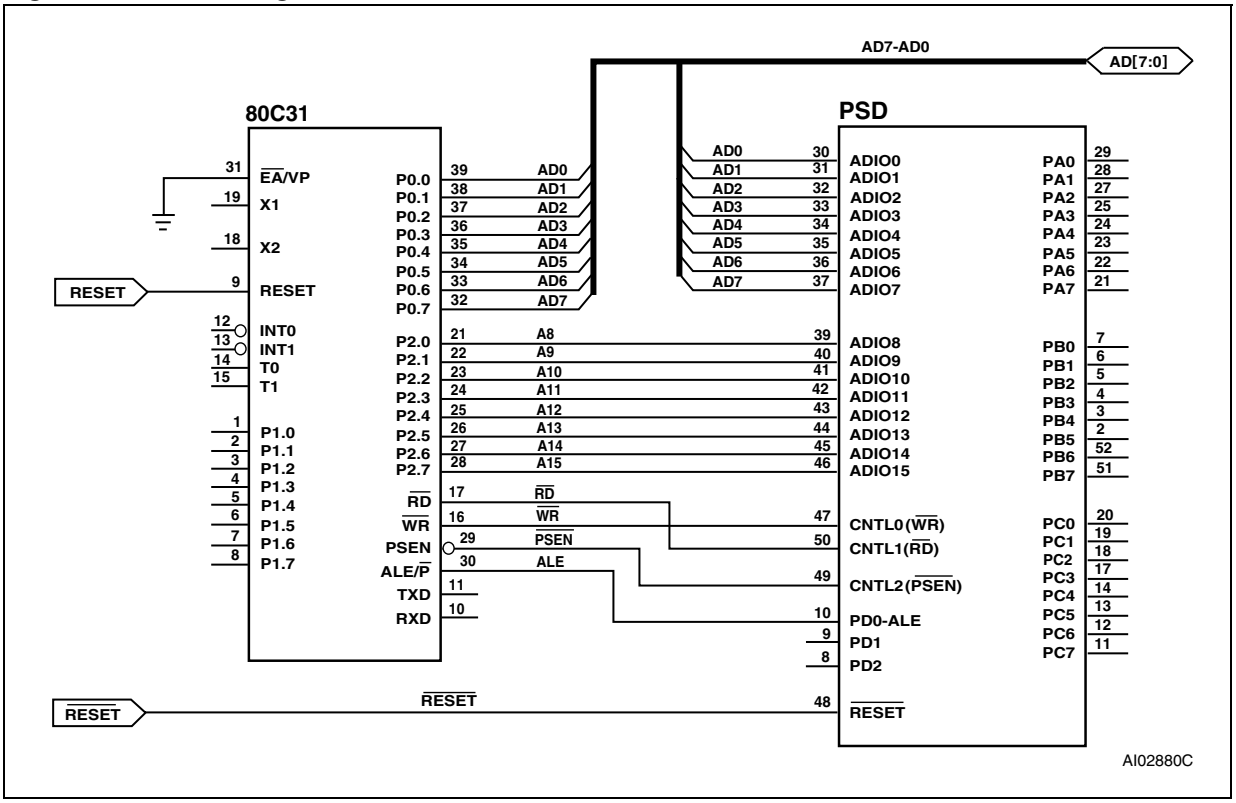
## 18.11 8-bit MCU bus interface examples

*Figure 26* to *Figure 30* show examples of the basic connections between the PSD and some popular MCUs. The PSD Control input pins are labeled as to the MCU function for which they are configured. The MCU bus interface is specified using the PSDsoft Express Configuration.

## 18.12 80C31

*Figure 26* shows the bus interface for the 80C31 which has an 8-bit, multiplexed address/data bus. The lower address byte is multiplexed with the data bus. The MCU control signals Program Select Enable ($\overline{PSEN}$, CNTL2), READ Strobe ($\overline{RD}$, CNTL1), and WRITE Strobe ($\overline{WR}$, CNTL0) may be used for accessing the internal memory and I/O ports blocks. Address Strobe (ALE/AS, PD0) latches the address.

**Figure 26. Interfacing the PSD with an 80C31**



## 18.13 80C251

The Intel 80C251 MCU features a user-configurable bus interface with four possible bus configurations, as shown in *Table 40*. The first configuration is 80C31-compatible, and the bus interface to the PSD is identical to that shown in *Figure 26*. The second and third configurations have the same bus connection as shown in *Figure 27*. There is only one READ Strobe ($\overline{PSEN}$) connected to CNTL1 on the PSD. The A16 connection to PA0 allows for a larger address input to the PSD. The fourth configuration is shown in *Figure 28*. READ Strobe ($\overline{RD}$) is connected to CNTL1 and Program Select Enable ($\overline{PSEN}$) is connected to CNTL2.

The 80C251 has two major operating modes:

### 18.13.1 Page mode

Data (D7-D0) is multiplexed with address (A15-A8). In a bus cycle where there is a Page hit, Address Strobe (ALE/AS, PD0) is not active and only addresses (A7-A0) are changing.

### 18.13.2 Non-Page mode

The data is multiplexed with the lower address byte and Address Strobe (ALE/AS, PD0) is active in every bus cycle.

The PSD supports both modes. In Page mode, the PSD bus timing is identical to Non-Page mode, except the address hold time and setup time with respect to Address Strobe (ALE/AS, PD0) is not required. The PSD access time is measured from address (A7-A0) valid to data invalid.

**Table 40. 80C251 configurations**

| Configuration | 80C251 READ/WRITE pins | Connecting to PSD pins | Page mode |
|---|---|---|---|
| 1 | WR<br>RD<br>PSEN | CNTL0<br>CNTL1<br>CNTL2 | Non-Page mode, 80C31 compatible A7-A0 multiplex with D7-D0 |
| 2 | WR<br>PSEN only | CNTL0<br>CNTL1 | Non-Page mode<br>A7-A0 multiplex with D7-D0 |
| 3 | WR<br>PSEN only | CNTL0<br>CNTL1 | Page mode<br>A15-A8 multiplex with D7-D0 |
| 4 | WR<br>RD<br>PSEN | CNTL0<br>CNTL1<br>CNTL2 | Page mode<br>A15-A8 multiplex with D7-D0 |

**Figure 27. Interfacing the PSD with the 80C251, with one READ Input**

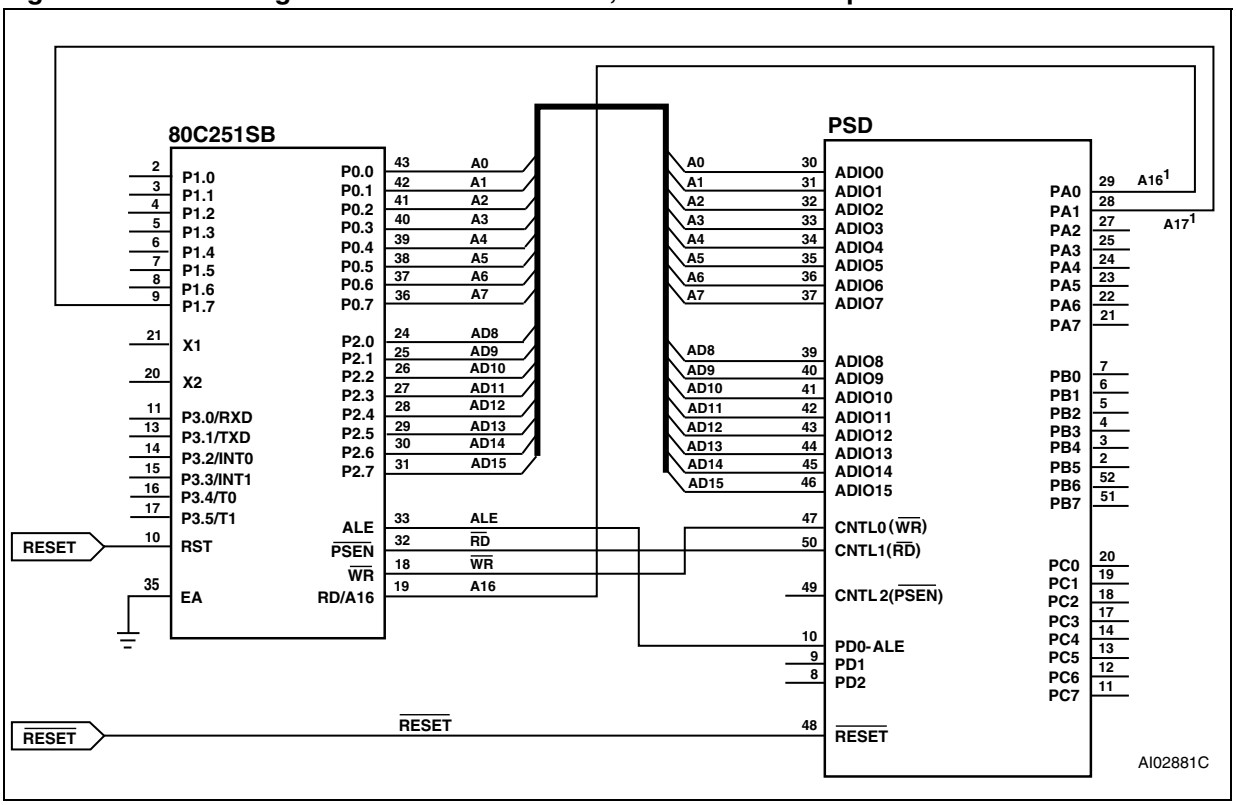
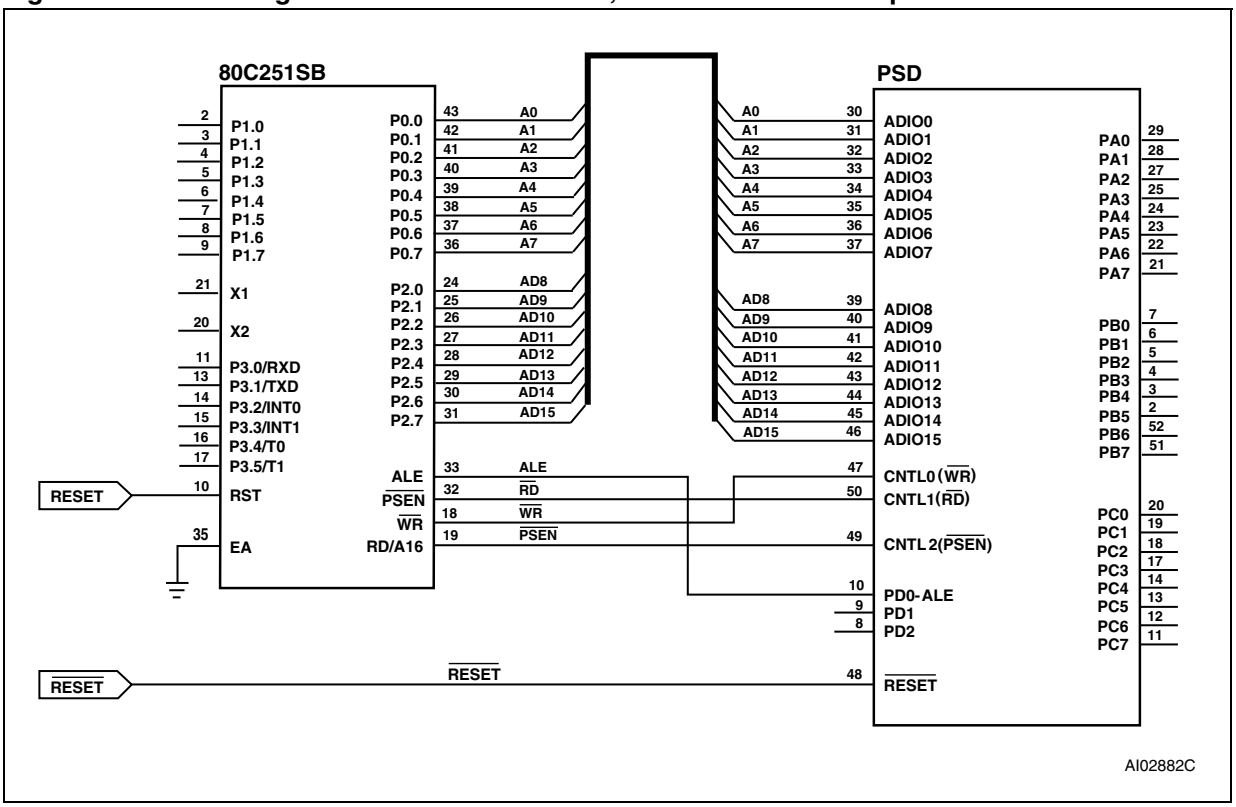



**Figure 28. Interfacing the PSD with the 80C251, with $\overline{RD}$ and $\overline{PSEN}$ Inputs**



## 18.14 80C51XA

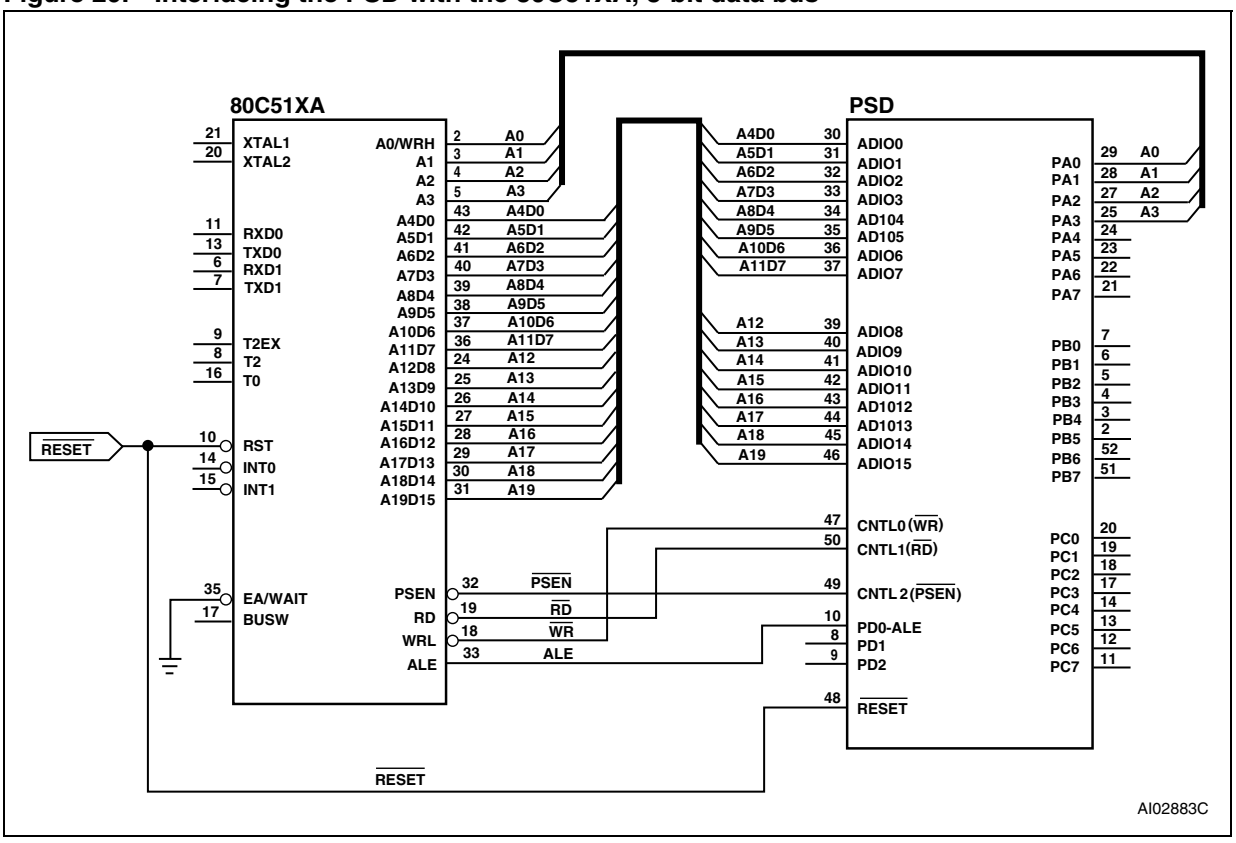
The Philips 80C51XA MCU family supports an 8- or 16-bit, multiplexed bus that can have burst cycles. Address bits (A3-A0) are not multiplexed, while (A19-A4) are multiplexed with data bits (D15-D0) in 16-bit mode. In 8-bit mode, (A11-A4) are multiplexed with data bits (D7-D0).

The 80C51XA can be configured to operate in 8-bit data mode (see *Figure 29*).

The 80C51XA improves bus throughput and performance be executing burst cycles for code retrievals.

In Burst mode, address (A19-A4) are latched internally by the PSD, while the 80C51XA changes the A3-A0 signals to retrieve up to 16 bytes of code. The PSD access time is them measured from address A3-A0 valid to data invalid. The PSD bus timing requirement in Burst mode is identical to the normal bus cycle, except the address setup and hold time with respect to Address Strobe (ALE/AS, PD0) does not apply.

**Figure 29. Interfacing the PSD with the 80C51XA, 8-bit data bus**

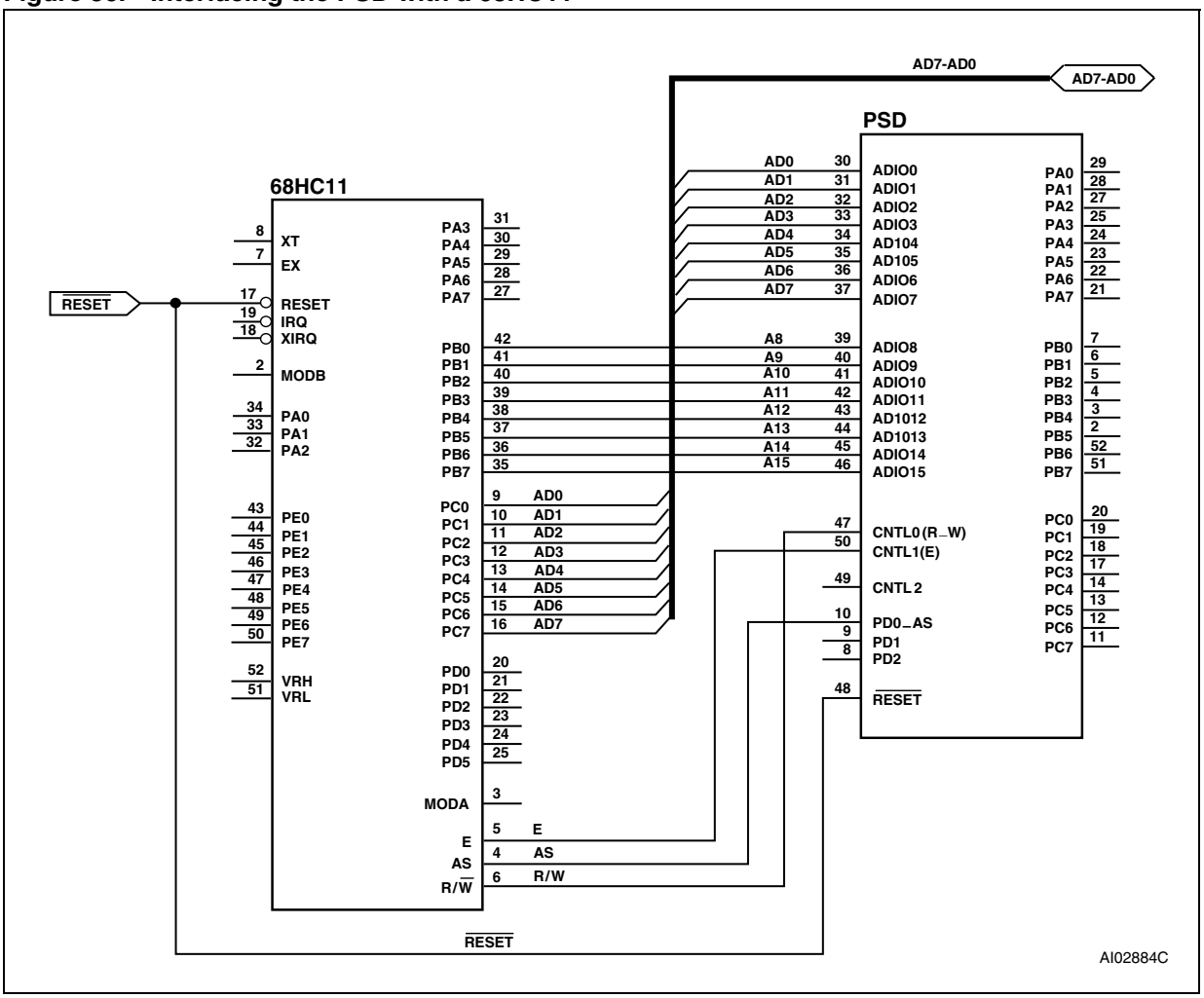


## 18.15 68HC11

Figure *Figure 30* shows a bus interface to a 68HC11 where the PSD is configured in 8-bit, multiplexed mode with E and R/$\overline{W}$ settings.

The DPLD can be used to generate the READ and $\overline{WR}$ signals for external devices.

**Figure 30. Interfacing the PSD with a 68HC11**



# 19 I/O ports

There are seven programmable I/O ports: Ports A, B, C, D, E, F and G. Each port pin is individually user configurable, thus allowing multiple functions per port. The ports are configured using PSDsoft or by the MCU writing to on-chip registers in the CSIOP space.

The topics discussed in this section are:

- General port architecture
- Port operating modes
- Port Configuration registers (PCR)
- Port data registers
- Individual port functionality.

## 19.1 General port architecture

The general architecture of the I/O port block is shown in *Figure 31*. Individual port architectures are shown in *Figure 33*, *Figure 34*, and *Figure 35*. In general, once the purpose for a port pin has been defined, that pin is no longer available for other purposes. Exceptions are noted.

As shown in *Figure 31*, the ports contain an output multiplexer whose select signals are driven by the configuration bits in the Control registers (Ports E, F and G only) and PSDsoft Configuration. Inputs to the multiplexer include the following:

- Output data from the Data Out register
- Latched address outputs
- CPLD macrocell output
- External Chip Select from the CPLD.

The Port Data Buffer (PDB) is a tri-state buffer that allows only one source at a time to be read. The Port Data Buffer (PDB) is connected to the Internal Data Bus for feedback and can be read by the MCU. The Data Out and macrocell outputs, Direction register and Control register, and port pin input are all connected to the Port Data Buffer (PDB).

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 PSDabel file, 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.

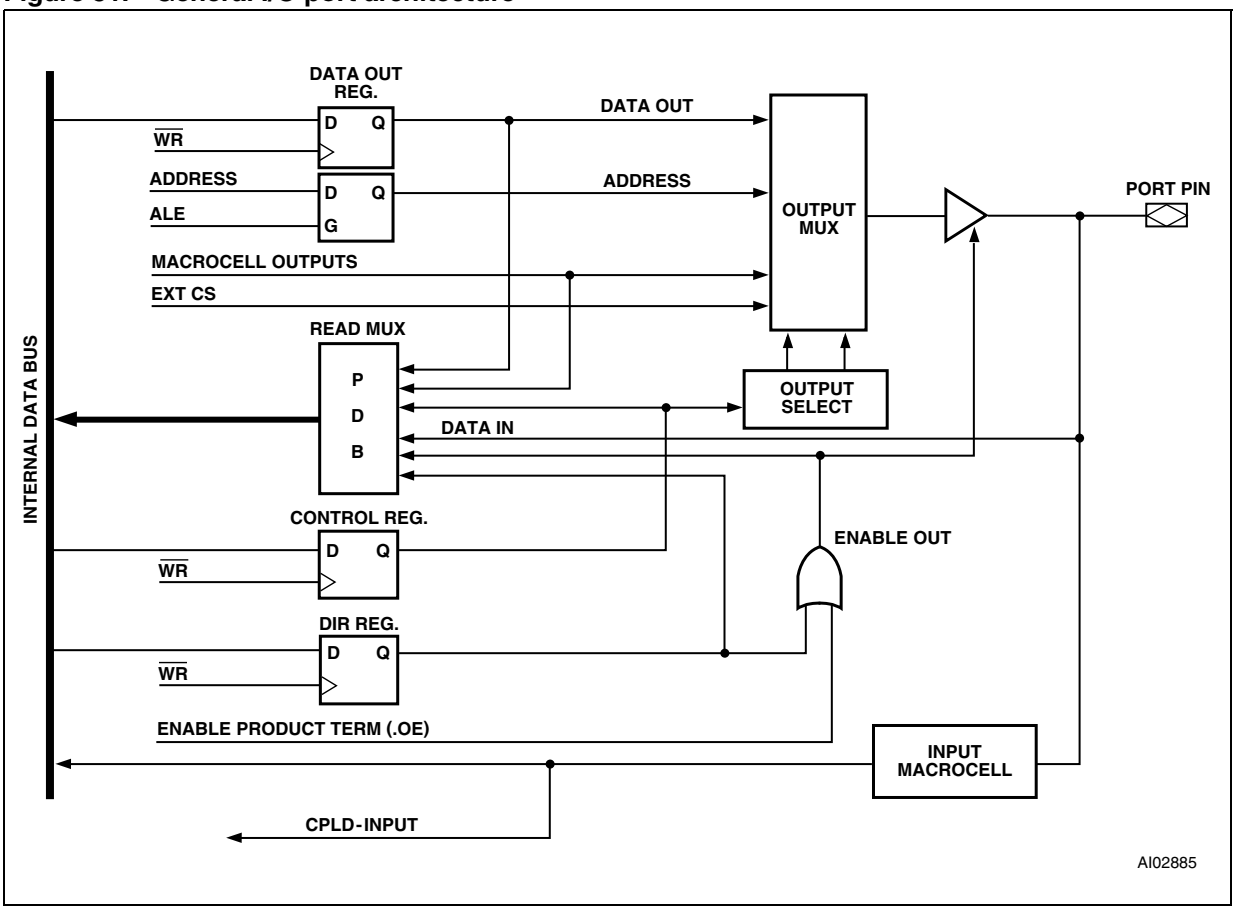
Ports A, B, and C have embedded input macrocells (IMC). The Input macrocells (IMC) can be configured as latches, registers, or direct inputs to the PLDs. The latches and registers are clocked by Address Strobe (ALE/AS, PD0) or a product term from the PLD AND Array. The outputs from the input macrocells (IMC) drive the PLD input bus and can be read by the MCU. See *Section 17.6: Input macrocells (IMC)*.

## 19.2 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 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, Peripheral I/O and MCU $\overline{\text{RESET}}$ 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 41* summarizes which modes are available on each port. *Table 42* shows how and where the different modes are configured. Each of the port operating modes are described in the following sections.

**Figure 31. General I/O port architecture**



## 19.3 MCU I/O mode

In the MCU I/O mode, the MCU uses the PSD ports to expand its own I/O ports. By setting up the CSIOP space, the ports on the PSD are mapped into the MCU address space. The addresses of the ports are listed in *Table 6*.

A port pin can be put into MCU I/O mode by writing a 0 to the corresponding bit in the Control register (for Ports E, F and G). The MCU I/O direction may be changed by writing to the corresponding bit in the Direction register, or by the output enable product term. See *Section 19.2: Port operating modes*. When the pin is configured as an output, the content of the Data Out register drives the pin. When configured as an input, the MCU can read the port input through the Data In buffer. See *Figure 31*.

Ports A, B and C do not have Control registers, and are in MCU I/O mode by default. They can be used for PLD I/O if they are specified in PSDsoft.

## 19.4 PLD I/O mode

The PLD I/O mode uses a port as an input to the CPLD's input macrocells (IMC), and/or as an output from the CPLD's output macrocells (OMC). The output can be tri-stated with a control signal.

This output enable control signal can be defined by a product term from the PLD, or by resetting the corresponding bit in the Direction register to 0. The corresponding bit in the Direction register must not be set to 1 if the pin is defined for a PLD input signal in PSDsoft. The PLD I/O mode is specified in PSDsoft by declaring the port pins, and then specifying an equation in PSDsoft.

## 19.5 Address Out mode

For MCUs with a multiplexed address/data bus, Address Out mode can be used to drive latched addresses onto the port pins. These port pins can, in turn, drive external devices. Either the output enable or the corresponding bits of both the Direction register and Control register must be set to a 1 for pins to use Address Out mode. This must be done by the MCU at run-time. See *Table 43* for the address output pin assignments on Ports E, F and G for various MCUs.

Note: Do not drive address signals with Address Out mode to an external memory device if it is intended for the MCU to Boot from the external device. The MCU must first Boot from PSD memory so the Direction and Control register bits can be set.

**Table 41. Port operating modes**

| Port mode | Port A | Port B | Port C | Port D | Port E | Port F | Port G |
|---|---|---|---|---|---|---|---|
| MCU I/O | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PLD I/O | | | | | | | |
| McellA output | Yes | Yes | No | No | No | No | No |
| McellB outputs | No | Yes | No | No | No | No | No |
| Additional Ext. CS outputs | No | No | Yes | No | No | Yes | No |
| PLD Inputs | Yes | Yes | Yes | Yes | No | Yes | No |
| Address Out | No | No | No | No | Yes (A7 – 0) | Yes (A7 – 0) | Yes (A7 – 0) or (A15 – 8) |
| Address In | Yes | Yes | Yes | Yes | No | Yes | No |
| Data Port | No | No | No | No | No | Yes | Yes |
| Peripheral I/O | Yes | No | No | Yes | No | Yes | No |
| JTAG ISP | No | No | No | No | Yes(1) | No | No |
| MCU $\overline{\text{RESET}}$ mode(2) | No | No | No | No | No | Yes | Yes |

1. Can be multiplexed with other I/O functions.
2. Available to Motorola 16-bit 683xx and HC16 families of MCUs.

**Table 42. Port operating mode settings(1)**

| Mode | Defined in PSDsoft | Control register setting | Direction register setting | VM register setting | JTAG enable |
|---|---|---|---|---|---|
| MCU I/O | Declare pins only | 0(2) | 1 = output, 0 = input(3) | N/A | N/A |
| PLD I/O | Declare pins and Logic equations | N/A | (3) | N/A | N/A |
| Data Port (Port F, G) | Selected for MCU with non-multiplexed bus | N/A | N/A | N/A | N/A |
| Address Out (Port E, F, G) | Declare pins only | 1 | 1(3) | N/A | N/A |
| Address In (Port A, B, C, D, F) | Declare pins or Logic equation for input macrocells | N/A | N/A | N/A | N/A |
| Peripheral I/O (Port F) | Logic equations (PSEL0 and PSEL1) | N/A | N/A | PIO bit = 1 | N/A |
| JTAG ISP(4) | Declare pins only | N/A | N/A | N/A | JTAG_Enable |
| MCU $\overline{\text{RESET}}$ mode | Specific pin logic level | N/A | N/A | N/A | N/A |

1. N/A = Not Applicable
2. Control register setting is not applicable to Ports A, B and C.
3. The direction of the Port A,B,C, and F pins are controlled by the Direction register ORed with the individual output enable product term (.oe) from the CPLD AND Array.
4. Any of these three methods enables the JTAG pins on Port E.

**Table 43. I/O port latched address output assignments(1)**

| MCU | Port E (PE3-PE0) | Port E (PE7-PE4) | Port F (PF3-PF0) | Port F (PF7-PF4) | Port G (PG3-PG0) | Port G (PG7-PG4) |
|---|---|---|---|---|---|---|
| 80C51XA | N/A(1) | Address a7-a4 | N/A | Address a7-a4 | Address a11-a8 | Address a15-a12 |
| 80C251 | N/A | N/A | N/A | N/A | Address a11-a8 | Address a15-a12 |
| All Other MCUs with Multiplexed Bus | Address a3-a0 | Address a7-a4 | Address a3-a0 | Address a7-a4 | Address a11-a8 (a3-a0 for 8-bit MCU) | Address a15-a12 (a7-a4 for 8-bit MCU) |

1. N/A = Not Applicable.

## 19.6 Address In mode

For MCUs that have more than 16 address signals, the higher addresses can be connected to Port A, B, C, D or F, and are routed as inputs to the PLDs. The address input can be latched in the input macrocell (IMC) by Address Strobe (ALE/AS, PD0). Any input that is included in the DPLD equations for the primary Flash memory, secondary Flash memory or SRAM is considered to be an address input.

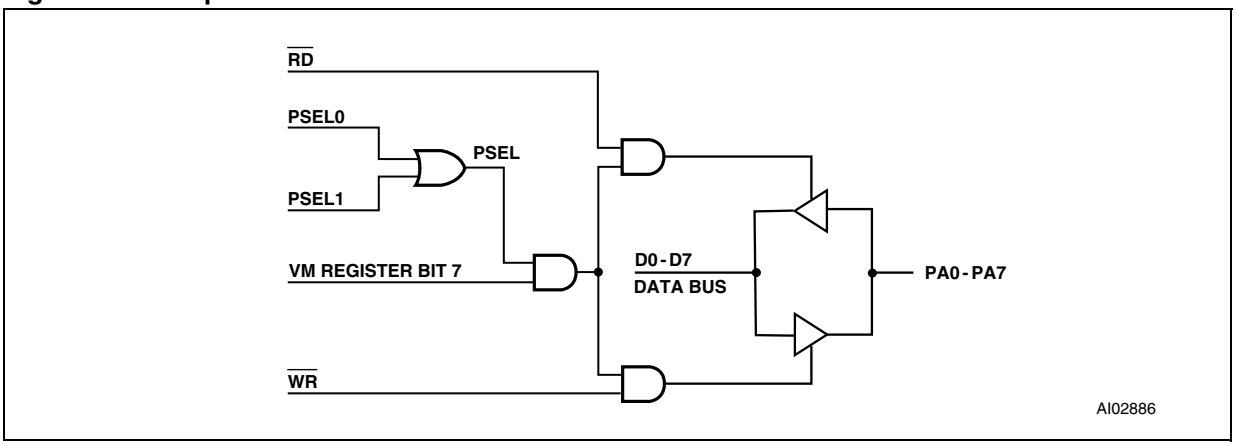
## 19.7 Data Port mode

Ports F and G can be used as a data bus port for a MCU with a non-multiplexed address/data bus. The Data Port is connected to the data bus of the MCU. The general I/O functions are disabled in Ports F and G if the ports are configured as a Data Port. Data Port mode is automatically configured in PSDsoft when a non-multiplexed bus MCU is selected.

## 19.8 Peripheral I/O mode

Peripheral I/O mode can be used to interface with external 8-bit peripherals. In this mode, all of Port F serves as a tri-state, bi-directional data buffer for the MCU. Peripheral I/O mode is enabled by setting bit 7 of the VM register to a 1. *Figure 32* shows how Port A acts as a bi-directional buffer for the MCU data bus if Peripheral I/O mode is enabled. An equation for PSEL0 and/or PSEL1 must be specified in PSDsoft. The buffer is tri-stated when PSEL0 or PSEL1 is not active.

**Figure 32. Peripheral I/O mode**




## 19.9 JTAG in-system programming (ISP)

Port E is JTAG compliant, and can be used for in-system programming (ISP). You can multiplex JTAG operations with other functions on Port E because in-system programming (ISP) is not performed during normal system operation. For more information on the JTAG Port, see *Section 2.5: ISP via JTAG port*.

## 19.10 MCU $\overline{\text{RESET}}$ mode

Ports F and G can be configured to operate in MCU $\overline{\text{RESET}}$ mode. This mode is available when PSD is configured for the Motorola 16-bit 683xx and HC16 family and is active only during reset.

At the rising edge of the $\overline{\text{RESET}}$ input, the MCU reads the logic level on the data bus (D15-D0) pins. The MCU then configures some of its I/O pin functions according to the logic level input on the data bus lines. Two dedicated buffers are usually enabled during $\overline{\text{RESET}}$ to drive the data bus lines to the desired logic level.

The PSD can replace the two buffers by configuring Ports F and G to operate in MCU $\overline{\text{RESET}}$ mode. In this mode, the PSD will drive the pre-defined logic level or data pattern on to the MCU data bus when $\overline{\text{RESET}}$ is active and there is no ongoing bus cycle. After $\overline{\text{RESET}}$, Ports F and G return to the normal Data Port mode.

The MCU $\overline{\text{RESET}}$ mode is enabled and configured in PSDsoft. The user defines the logic level (data pattern) that will be drive out from Ports F and G during $\overline{\text{RESET}}$.

## 19.11 Port Configuration registers (PCR)

Each port has a set of Port Configuration registers (PCR) used for configuration. The contents of the registers can be accessed by the MCU through normal READ/WRITE bus cycles at the addresses given in *Table 6*. The addresses in *Table 6* are the offsets in hexadecimal from the base of the CSIOP register.

The pins of a port are individually configurable and each bit in the register controls its respective pin. For example, bit 0 in a register refers to bit 0 of its port. The three Port

Configuration registers (PCR), shown in *Table 44*, are used for setting the port configurations. The default Power-up state for each register in *Table 44* is 00h.

## 19.12 Control register

Any bit reset to '0' in the Control register sets the corresponding port pin to MCU I/O mode, and a 1 sets it to Address Out mode. The default mode is MCU I/O. Only Ports E, F and G have an associated control register.

**Table 44. Port configuration registers (PCR)**

| Register name | Port | MCU access |
|---|---|---|
| Control | E, F, G | WRITE/READ |
| Direction | A, B, C, D, E, F, G | WRITE/READ |
| Drive Select(1) | A, B, D, E, G | WRITE/READ |

1. See *Table 48* for Drive register bit definition.

## 19.13 Direction register

The Direction register controls the direction of data flow in the I/O ports. Any bit set to 1 in the Direction register causes the corresponding pin to be an output, and any bit set to 0 causes it to be an input. The default mode for all port pins is input.

*Figure 33*, *Figure 34* and *Figure 35* show the port Architecture diagrams for Ports A/B/C, D, and E/F/G, respectively. The direction of data flow for Ports A, B, C and F are controlled not only by the direction register, but also by the output enable product term from the PLD AND Array. If the output enable product term is not active, the Direction register has sole control of a given pin's direction.

An example of a configuration for a port with the three least significant bits set to output and the remainder set to input is shown in *Table 47*. Since Port D only contains four pins, the Direction register for Port D has only the four least significant bits active.

## 19.14 Drive Select register

The Drive Select register configures the pin driver as Open Drain or CMOS. An external pull-up resistor should be used for pins configured as Open Drain.

A pin can be configured as Open Drain if its corresponding bit in the Drive Select register is set to a 1. The default pin drive is CMOS.

*Table 48* shows the Drive register for Ports A, B, D, E and G. It summarizes which pins can be configured as Open Drain outputs.

**Table 45. Port pin Direction Control, Output Enable P.T. not defined**

| Direction register bit | Port pin mode |
|---|---|
| 0 | Input |
| 1 | Output |

**Table 46. Port pin Direction Control, Output Enable P.T. Defined**

| Direction register bit | Output Enable P.T. | Port pin mode |
|---|---|---|
| 0 | 0 | Input |
| 0 | 1 | Output |
| 1 | 0 | Output |
| 1 | 1 | Output |

**Table 47. Port direction assignment example**

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

**Table 48. Drive register pin assignment(1)**

| 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 | Open Drain | Open Drain | Open Drain | Open Drain |
| Port B | 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) | Open Drain | Open Drain | Open Drain | Open Drain |
| Port E | Open Drain | Open Drain | Open Drain | Open Drain | Open Drain | Open Drain | Open Drain | Open Drain |
| Port G | Open Drain | Open Drain | Open Drain | Open Drain | Open Drain | Open Drain | Open Drain | Open Drain |

1. NA = Not Applicable.

## 19.15 Port Data registers

The Port Data registers, shown in *Table 49*, are used by the MCU to write data to or read data from the ports. *Table 49* shows the register name, the ports having each register type, and MCU access for each register type. The registers are described next.

## 19.16 Data In

Port pins are connected directly to the Data In buffer. In MCU I/O Input mode, the pin input is read through the Data In buffer.

## 19.17 Data Out register

Stores output data written by the MCU in the MCU I/O output mode. The contents of the register are driven out to the pins if the Direction register or the output enable product term is set to 1. The contents of the register can also be read back by the MCU.

## 19.18 Output macrocells (OMC)

The CPLD output macrocells (OMC) occupy a location in the MCU's address space. The MCU can read the output of the output macrocells (OMC). If the Mask Macrocell register bits are not set, writing to the macrocell loads data to the macrocell flip-flops. See *Section 19: I/O ports*.

## 19.19 Mask Macrocell register

Each Mask Macrocell register bit corresponds to an output macrocell (OMC) flip-flop. When the Mask Macrocell 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.

## 19.20 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 17.6: Input macrocells (IMC)*.

**Table 49. Port data registers**

| Register name | Port | MCU access |
|---|---|---|
| Data In | A, B, C, D, E, F, G | READ – input on pin |
| Data Out | A, B, C, D, E, F, G | WRITE/READ |
| Output macrocell | A, B | READ – outputs of macrocells<br>WRITE – loading macrocells flip-flop |
| Mask macrocell | A, B | WRITE/READ – prevents loading into a given Macrocell |
| Input macrocell | A, B, C | READ – outputs of the input macrocells |
| Enable Out | A, B, C, F | READ – the output enable control of the port driver |

## 19.21 Enable Out

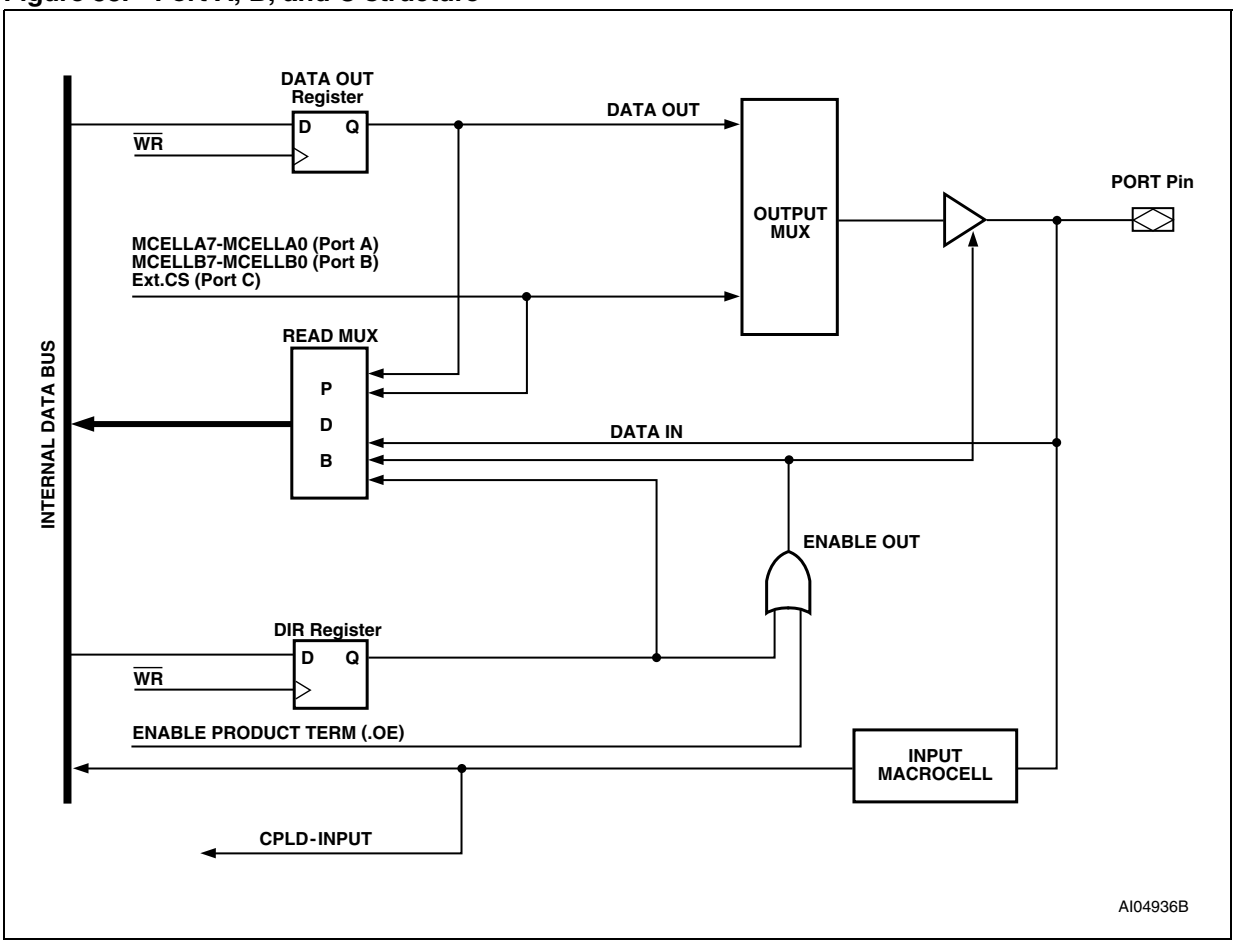
The Enable Out register can be read by the MCU. It contains the output enable values for a given port. A 1 indicates the driver is in output mode. A 0 indicates the driver is in tri-state and the pin is in input mode.

## 19.22 Ports A, B and C – Functionality and Structure

Ports A, B, and C have similar functionality and structure, as shown in *Figure 33*. The ports can be configured to perform one or more of the following functions:

- MCU I/O mode
- CPLD Output – macrocells McellA7-McellA0 can be connected to Port A. McellB7-McellB0 can be connected to Port B. External Chip Select (ECS7-ECS0) can be connected to Port C or Port F.
- CPLD Input – Via the input macrocells (IMC).
- Address In – Additional high address inputs using the input macrocells (IMC).
- Open Drain – pins PA7-PA0 can be configured to Open Drain mode.

**Figure 33. Port A, B, and C structure**

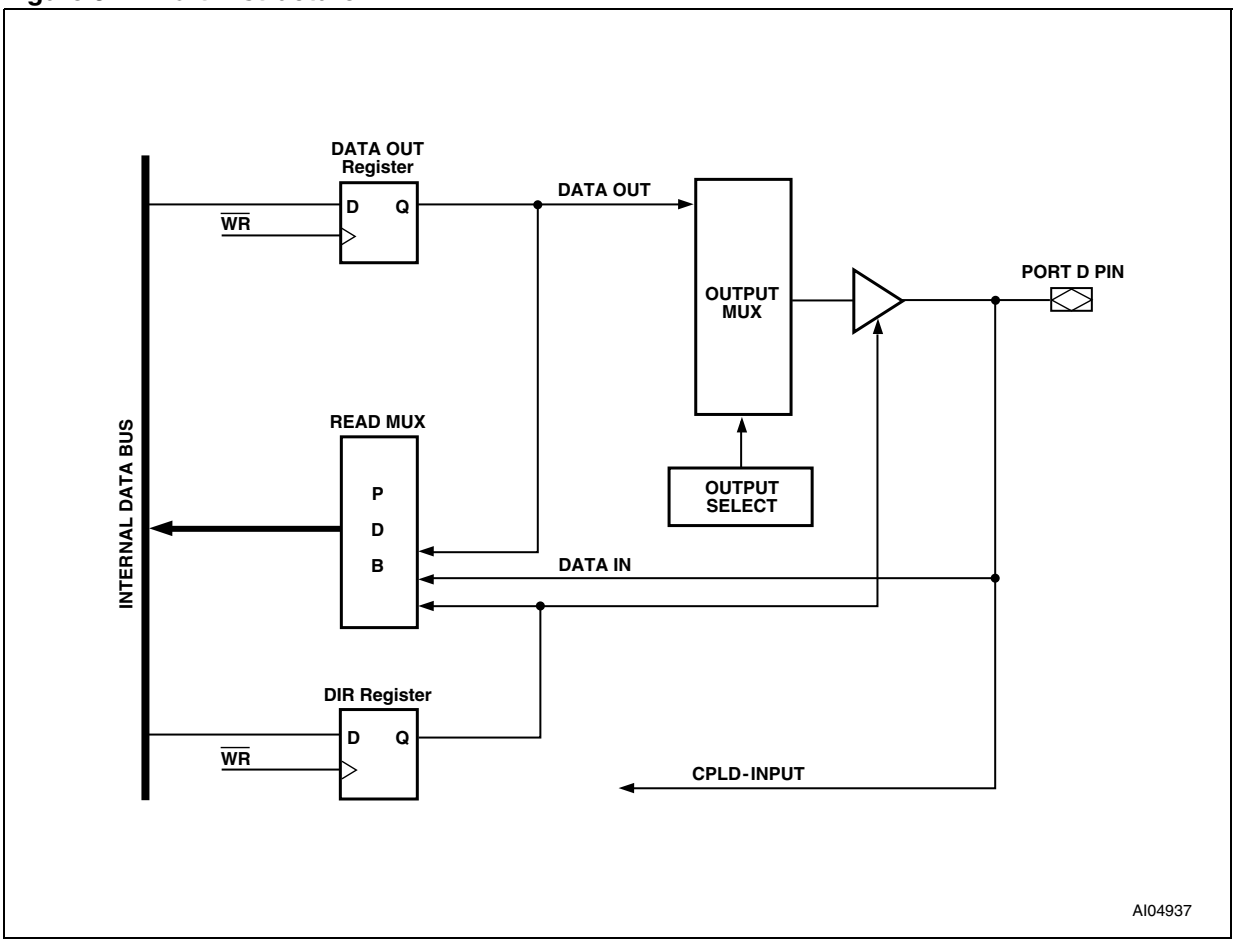


## 19.23 Port D – functionality and structure

Port D has four I/O pins. See *Figure 34*. Port D can be configured to perform one or more of the following functions:

- MCU I/O mode
- CPLD Input – direct input to the CPLD, no input macrocells (IMC)
- Port D pins can be configured in PSDsoft as input pins for other dedicated functions:
- Address Strobe (ALE/AS, PD0)
- CLKIN (PD1) as input to the macrocells Flip-flops and APD counter
- PSD Chip Select Input ($\overline{CSI}$, PD2). Driving this signal High disables the Flash memory, SRAM and CSIOP.
- WRITE-Enable High-byte ($\overline{WRH}$, PD3) input, or as DBE input from a MC68HC912.

**Figure 34. Port D structure**



## 19.24 Port E – functionality and structure

Port E can be configured to perform one or more of the following functions (see *Figure 35*):

- MCU I/O mode
- In-system programming (ISP) – JTAG port can be enabled for programming/erase of the PSD device. (See *Section 22: In-circuit programming using the serial interface* for more information on JTAG programming.)
- Open Drain – pins can be configured in Open Drain mode
- Latched Address output – Provide latched address output.

## 19.25 Port F – functionality and structure

Port F can be configured to perform one or more of the following functions:

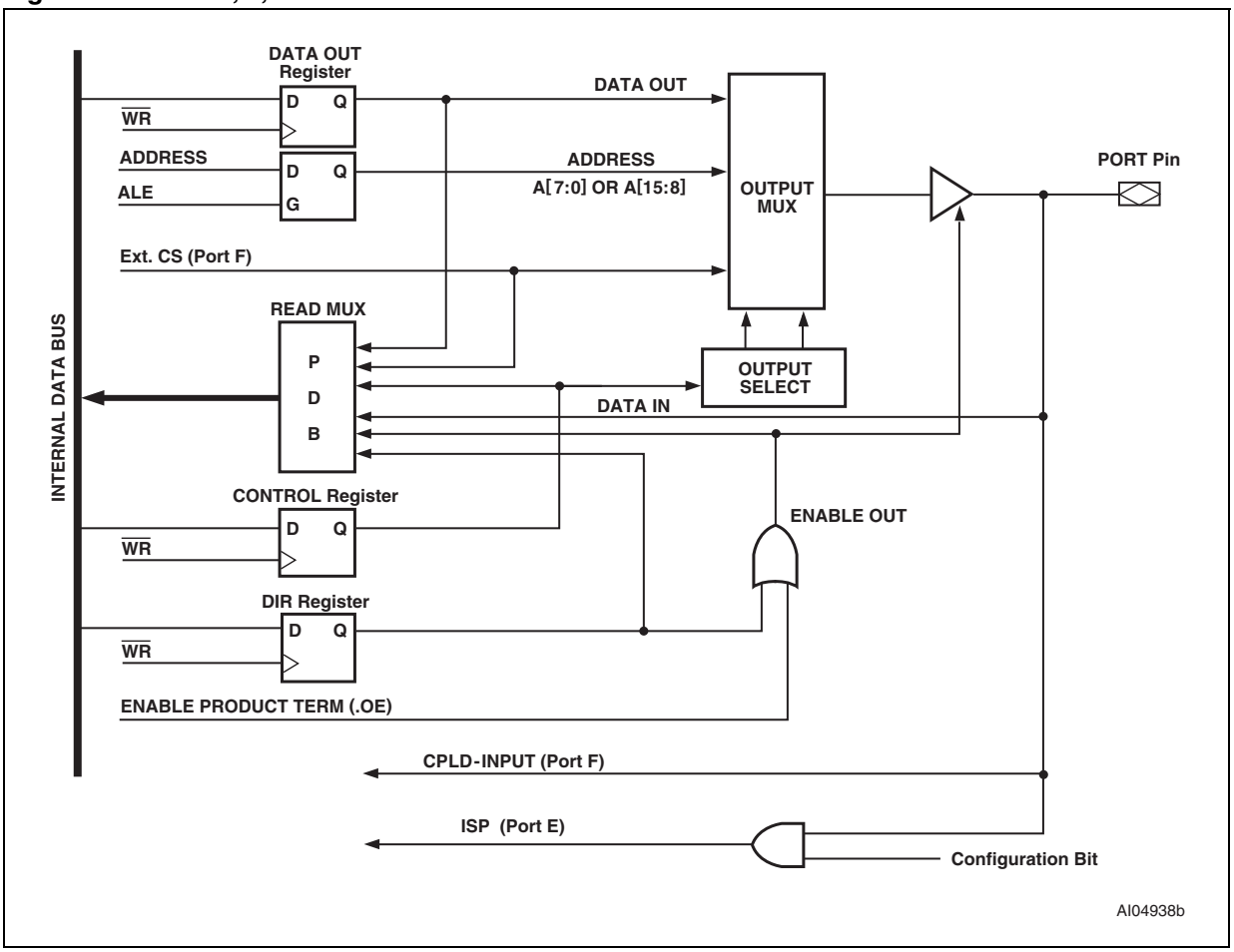
- MCU I/O mode
- CPLD Output – External Chip Select (ECS7-ECS0) can be connected to Port F or Port C.
- CPLD Input – direct input to the CPLD, no input macrocells (IMC)
- Latched Address output – Provide latched address output as per *Table 43*.
- Data Port – connected to D7-D0 when Port F is configured as Data Port for a non-multiplexed bus
- Peripheral mode
- MCU $\overline{\text{RESET}}$ mode – for 16-bit Motorola 683xx and HC16 MCUs

## 19.26 Port G – Functionality and Structure

Port G can be configured to perform one or more of the following functions:

- MCU I/O mode
- Latched Address output – Provide latched address output as per *Table 43*.
- Open Drain – pins can be configured in Open Drain mode
- Data Port – connected to D15-D8 when Port G is configured as Data Port for a non-multiplexed bus
- MCU $\overline{\text{RESET}}$ mode – for 16-bit Motorola 683xx and HC16 MCUs

**Figure 35. Port E, F, and G structure**



# 20 Power management

The PSD device offers configurable power saving options. These options may be used individually or in combinations, as follows:

- All memory blocks in a PSD (primary Flash memory, secondary Flash memory, and SRAM) are built with power management technology. In addition to using special silicon design methodology, power management technology puts the memories into Standby mode when address/data inputs are not changing (zero DC current). As soon as a transition occurs on an input, the affected memory “wakes up”, changes and latches its outputs, then goes back to standby. The designer does *not* have to do anything special to achieve memory Standby mode when no inputs are changing—it happens automatically.
  The PLD sections can also achieve Standby mode when its inputs are not changing, as described for the Power Management mode registers (PMMR), later.
- The Automatic power-down (APD) block allows the PSD to reduce to standby current automatically. The APD Unit also blocks MCU address/data signals from reaching the memories and PLDs. This feature is available on all PSD devices. The APD Unit is described in more detail in *Section 20.1: Automatic power-down (APD) Unit and Power-down mode*.
  Built in logic monitors the Address Strobe of the MCU for activity. If there is no activity for a certain period (the MCU is asleep), the APD Unit initiates Power-down mode (if enabled). Once in Power-down mode, all address/data signals are blocked from reaching the PSD memories and PLDs, and the memories are deselected internally. This allows the memories and PLDs to remain in Standby mode even if the address/data signals are changing state externally (noise, other devices on the MCU bus, etc.). Keep in mind that any unblocked PLD input signals that are changing states keeps the PLD out of Standby mode, but not the memories.
- PSD Chip Select Input ($\overline{\text{CSI}}$, PD2) can be used to disable the internal memories, placing them in Standby mode even if inputs are changing. This feature does not block any internal signals or disable the PLDs. This is a good alternative to using the APD Unit, especially if your MCU has a chip select output. There is a slight penalty in memory access time when PSD Chip Select Input ($\overline{\text{CSI}}$, PD2) makes its initial transition from deselected to selected.
- The Power Management mode registers (PMMR) can be written by the MCU at run-time to manage power. All PSD devices support “blocking bits” in these registers that are set to block designated signals from reaching both PLDs. Current consumption of the PLDs is directly related to the composite frequency of the changes on their inputs (see *Figure 39*).
  Significant power savings can be achieved by blocking signals that are not used in DPLD or CPLD logic equations at run-time. PSDsoft creates a fuse map that automatically blocks the low address byte (A7-A0) or the control signals (CNTL0-CNTL2, ALE and WRITE-Enable High-byte ($\overline{\text{WRH}}$/DBE, PD3)) if none of these signals are used in PLD logic equations.
  PSD devices have a Turbo bit in PMMR0. This bit can be set to turn the Turbo mode off (the default is with Turbo mode turned on). While Turbo mode is off, the PLDs can achieve standby current when no PLD inputs are changing (zero DC current). Even when inputs do change, significant power can be saved at lower frequencies (AC current), compared to when Turbo mode is on. When the Turbo mode is on, there is a significant DC current component, and the AC component is higher.

## 20.1 Automatic power-down (APD) Unit and Power-down mode

The APD Unit, shown in *Figure 36*, puts the PSD into Power-down mode by monitoring the activity of Address Strobe (ALE/AS, PD0). If the APD Unit is enabled, as soon as activity on Address Strobe (ALE/AS, PD0) stops, a four bit counter starts counting. If Address Strobe (ALE/AS, PD0) remains inactive for fifteen clock periods of CLKIN (PD1), Power-down (PDN) goes High, and the PSD enters Power-down mode, as discussed next.

## 20.2 Power-down mode

By default, if you enable the APD Unit, Power-down mode is automatically enabled. The device enters Power-down mode if Address Strobe (ALE/AS, PD0) remains inactive for fifteen periods of CLKIN (PD1).

The following should be kept in mind when the PSD is in Power-down mode:

- If Address Strobe (ALE/AS, PD0) starts pulsing again, the PSD returns to normal operation. The PSD also returns to normal operation if either PSD Chip Select Input ($\overline{\text{CSI}}$, PD2) is Low or the Reset ($\overline{\text{RESET}}$) input is High.
- The MCU address/data bus is blocked from all memory and PLDs.
- Various signals can be blocked (prior to Power-down mode) from entering the PLDs by setting the appropriate bits in the Power Management mode registers (PMMR). The blocked signals include MCU control signals and the common CLKIN (PD1). Note that blocking CLKIN (PD1) from the PLDs does not block CLKIN (PD1) from the APD Unit.
- All PSD memories enter Standby mode and are drawing standby current. However, the PLDs and I/O ports blocks do *not* go into Standby mode because you do not want to have to wait for the logic and I/O to “wake-up” before their outputs can change. See *Table 50* for Power-down mode effects on PSD ports.
- Typical Standby current is or the order of µA. This standby current value assumes that there are no transitions on any PLD input.

**Table 50. Effect of Power-down mode on ports**

| Port function | Pin level |
|---|---|
| MCU I/O | No change |
| PLD Out | No change |
| Address Out | Undefined |
| Data Port | Tri-state |
| Peripheral I/O | Tri-state |

**Figure 36. APD unit**

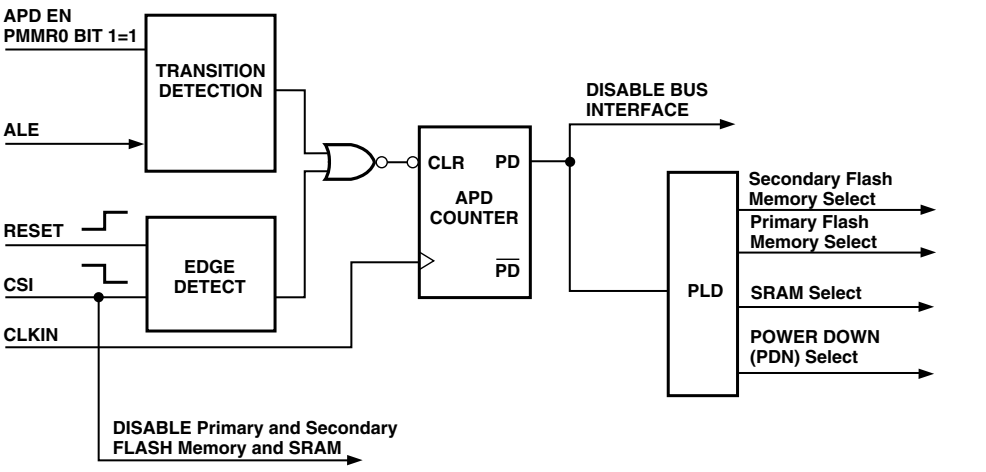



**Table 51. PSD timing and standby current during Power-down mode**

| Mode | PLD propagation delay | Memory access time | Access recovery time to normal access | Typical standby current |
|---|---|---|---|---|
| Power-down | Normal $t_{PD}$ (1) | No Access | $t_{LVDV}$ | 50 µA (2) |

1. Power-down does not affect the operation of the PLD. The PLD operation in this mode is based only on the Turbo bit.
2. Typical current consumption assuming no PLD inputs are changing state and the PLD Turbo bit is 0.

## 20.3 Other power saving options

The PSD offers other reduced power saving options that are independent of the Power-down mode. Except for PSD Chip Select Input ($\overline{CSI}$, PD2) features, they are enabled by setting bits in PMMR0 and PMMR2 (as summarized in *Table 23* and *Table 24*).

## 20.4 PLD power management

The power and speed of the PLDs are controlled by the Turbo bit (bit 3) in PMMR0. By setting the bit to 1, the Turbo mode is off and the PLDs consume the specified standby current when the inputs are not switching for an extended time of 70ns. The propagation delay time is increased after the Turbo bit is set to 1 (turned off) when the inputs change at a composite frequency of less than 15MHz. When the Turbo bit is reset to '0' (turned on), the PLDs run at full power and speed. The Turbo bit affects the PLD's DC power, AC power, and propagation delay. See the AC and DC characteristics tables in *Section 25: DC and AC parameters* for PLD timing values.

Blocking MCU control signals with the PMMR2 bits can further reduce PLD AC power consumption.

## 20.5 PSD Chip Select Input ($\overline{CSI}$, PD2)

PD2 of Port D can be configured in PSDsoft as PSD Chip Select Input ($\overline{CSI}$). When Low, the signal selects and enables the internal primary Flash memory, secondary Flash memory, SRAM, and I/O blocks for READ or WRITE operations involving the PSD. A High on PSD Chip Select Input ($\overline{CSI}$, PD2) disables the primary Flash memory, secondary Flash memory, and SRAM, and reduces the PSD power consumption. However, the PLD and I/O signals remain operational when PSD Chip Select Input ($\overline{CSI}$, PD2) is High.

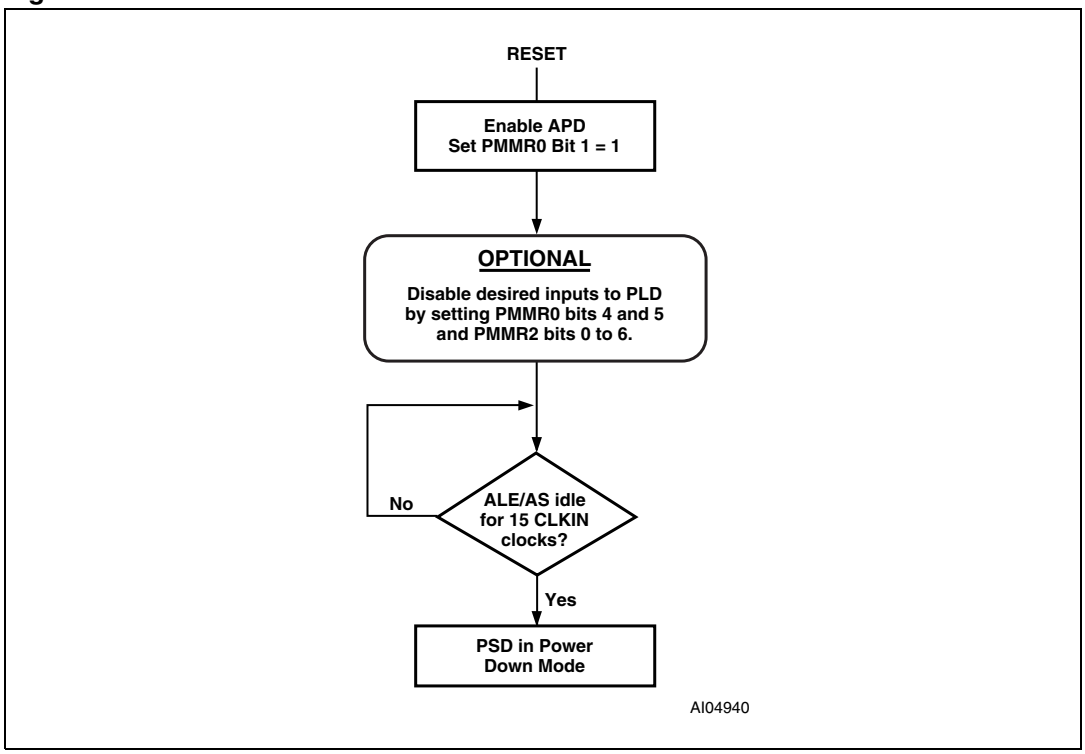
There may be a timing penalty when using PSD Chip Select Input ($\overline{CSI}$, PD2) depending on the speed grade of the PSD that you are using. See the timing parameter $t_{SLQV}$ in *Table 70*.

## 20.6 Input clock

The PSD provides the option to turn off CLKIN (PD1) to the PLD to save AC power consumption. CLKIN (PD1) is an input to the PLD AND Array and the output macrocells (OMC).

During Power-down mode, or, if CLKIN (PD1) is not being used as part of the PLD logic equation, the clock should be disabled to save AC power. CLKIN (PD1) is disconnected from the PLD AND Array or the macrocells block by setting bits 4 or 5 to a 1 in PMMR0.

**Figure 37. Enable Power-down flowchart**



## 20.7 Input control signals

The PSD provides the option to turn off the address input (A7-A0) and input control signals (CNTL0, CNTL1, CNTL2, Address Strobe (ALE/AS, PD0) and WRITE-Enable High-byte ($\overline{WRH}$/DBE, PD3)) to the PLD to save AC power consumption. These signals are inputs to the PLD AND Array. During Power-down mode, or, if any of them are not being used as part of the PLD logic equation, these control signals should be disabled to save AC power. They are disconnected from the PLD AND Array by setting bits 0, 2, 3, 4, 5 and 6 to a 1 in PMMR2.

**Table 52. ADP counter operation**

| APD Enable bit | ALE PD polarity | ALE level | APD counter |
|---|---|---|---|
| 0 | X | X | Not counting |
| 1 | X | Pulsing | Not counting |
| 1 | 1 | 1 | Counting (generates PDN after 15 clock cycles) |
| 1 | 0 | 0 | Counting (generates PDN after 15 clock cycles) |

# 21 RESET timing and device status at RESET

## 21.1 Power-on RESET

Upon Power-up, the PSD requires a Reset (RESET) pulse of duration $t_{NLNH-PO}$ (minimum 1ms) after $V_{CC}$ is steady. During this period, the device loads internal configurations, clears some of the registers and sets the Flash memory into Operating mode. After the rising edge of Reset (RESET), the PSD remains in the RESET mode for an additional period, $t_{OPR}$ (maximum 120 ns), before the first memory access is allowed.

The PSD Flash memory is reset to the Read mode upon Power-up. Sector Select (FS0-FS15 and CSBOOT0-CSBOOT3) must all be Low, WRITE Strobe (WR/WRL, CNTL0) High, during Power-on RESET for maximum security of the data contents and to remove the possibility of data being written on the first edge of WRITE Strobe (WR/WRL, CNTL0). Any Flash memory WRITE cycle initiation is prevented automatically when $V_{CC}$ is below $V_{LKO}$.

## 21.2 Warm RESET

Once the device is up and running, the device can be reset with a pulse of a much shorter duration, $t_{NLNH}$ (minimum 150 ns). The same $t_{OPR}$ period is needed before the device is operational after Warm RESET. *Figure 38* shows the timing of the Power-up and Warm RESET.

## 21.3 I/O pin, register and PLD Status at RESET

*Table 53* shows the I/O pin, register and PLD status during Power-on RESET, Warm RESET and Power-down mode. PLD outputs are always valid during Warm RESET, and they are valid in Power-on RESET once the internal PSD Configuration bits are loaded. This loading of PSD is completed typically long before the $V_{CC}$ ramps up to operating level. Once the PLD is active, the state of the outputs are determined by equations specified in PSDsoft.

## 21.4 RESET of Flash memory Erase and Program cycles

An external Reset (RESET) also resets the internal Flash memory state machine. During a Flash memory Program or Erase cycle, Reset (RESET) terminates the cycle and returns the Flash memory to the Read mode within a period of $t_{NLNH-A}$ (minimum 25 µs).

**Figure 38. Reset (RESET) timing**

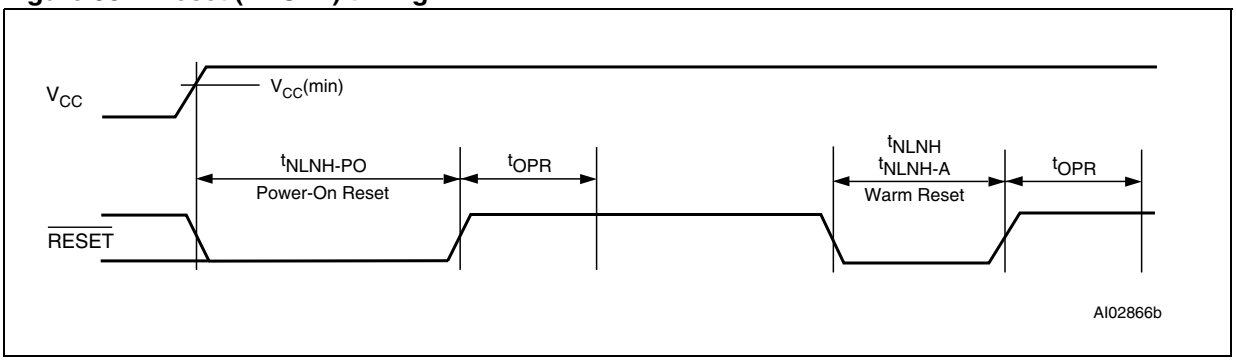


**Table 53. Status during Power-on $\overline{\text{RESET}}$, Warm $\overline{\text{RESET}}$, and Power-down mode**

| Port configuration | Power-on $\overline{\text{RESET}}$ | Warm Reset | Power-down mode |
|---|---|---|---|
| MCU I/O | Input mode | Input mode | Unchanged |
| PLD output | Valid after internal PSD configuration bits are loaded | Valid | Depends on inputs to PLD (addresses are blocked in PD mode) |
| Address Out | Tri-stated | Tri-stated | Not defined |
| Data Port | Tri-stated | Tri-stated | Tri-stated |
| Peripheral I/O | Tri-stated | Tri-stated | Tri-stated |

| Register | Power-On Reset | Warm Reset | Power-down mode |
|---|---|---|---|
| PMMR0 and PMMR2 | Cleared to 0 | Unchanged | Unchanged |
| Macrocells Flip-flop status | Cleared to 0 by internal Power-On Reset | Depends on .re and .pr equations | Depends on .re and .pr equations |
| VM register(1) | Initialized, based on the selection in PSDsoft configuration menu | Initialized, based on the selection in PSDsoft configuration menu | Unchanged |
| All other registers | Cleared to 0 | Cleared to 0 | Unchanged |

1. The SR_code and Peripheral mode bits in the VM register are always cleared to '0' on Power-on $\overline{\text{RESET}}$ or Warm $\overline{\text{RESET}}$.

# 22 In-circuit programming using the serial interface

The JTAG serial interface on the PSD can be enabled on Port E (see *Table 54*). All memory blocks (primary Flash memory and secondary Flash memory), PLD logic, and PSD Configuration bits may be programmed through the JTAG-ISC Serial Interface. A blank device can be mounted on a printed circuit board and programmed using JTAG in-system programming (ISP).

The standard JTAG signals (IEEE 1149.1) are TMS, TCK, TDI, and TDO. Two additional signals, TSTAT and $\overline{\text{TERR}}$, are optional JTAG extensions used to speed up Program and Erase cycles.

By default, on a blank PSD (as shipped from the factory, or after erasure), four pins on Port E are enabled for the basic JTAG signals TMS, TCK, TDI, and TDO.

See application note AN1153 for more details on JTAG in-system programming (ISP).

## 22.1 Standard JTAG signals

The standard JTAG signals (TMS, TCK, TDI, and TDO) can be enabled by any of three different conditions that are logically ORed. When enabled, TDI, TDO, TCK, and TMS are inputs, waiting for a serial command from an external JTAG controller device (such as FlashLINK or Automated Test Equipment). When the enabling command is received from the external JTAG controller device, TDO becomes an output and the JTAG channel is fully functional inside the PSD. The same command that enables the JTAG channel may optionally enable the two additional JTAG pins, TSTAT and $\overline{\text{TERR}}$.

The following symbolic logic equation specifies the conditions enabling the four basic JTAG pins (TMS, TCK, TDI, and TDO) on their respective Port E pins. For purposes of discussion, the logic label JTAG_ON is used. When JTAG_ON is true, the four pins are enabled for JTAG. When JTAG_ON is false, the four pins can be used for general PSD I/O.

```
JTAG_ON = PSDsoft_enabled +
/* An NVM configuration bit inside the PSD is set by the designer in
the PSDsoft Configuration utility. This dedicates the pins for JTAG
at all times (compliant with IEEE 1149.1 */

Microcontroller_enabled +
/* The microcontroller can set a bit at run-time by writing to the
PSD register, JTAG Enable. This register is located at address CSIOP
+ offset C7h. Setting the JTAG_ENABLE bit in this register will
enable the pins for JTAG use. This bit is cleared by a PSD reset or
the microcontroller. See Table 21 for bit definition. */

PSD_product_term_enabled;
/* A dedicated product term (PT) inside the PSD can be used to
enable the JTAG pins. This PT has the reserved name JTAGSEL. Once
defined as a node in PSDabel, the designer can write an equation for
JTAGSEL. This method is used when the Port E JTAG pins are
multiplexed with other I/O signals. It is recommended to tie
logically the node JTAGSEL to the JEN\ signal on the Flashlink cable
when multiplexing JTAG signals. See Application Note 1153 for
details. */
```

The state of the PSD Reset ($\overline{\text{RESET}}$) signal does not interrupt (or prevent) JTAG operations if the JTAG pins are dedicated by an NVM configuration bit (via PSDsoft). However, Reset ($\overline{\text{RESET}}$) will prevent or interrupt JTAG operations if the JTAG Enable register (as shown in *Table 21*) is used to enable the JTAG pins.

The PSD supports JTAG In-System-Programmability (ISP) commands, but not Boundary Scan. ST's PSDsoft software tool and FlashLINK JTAG programming cable implement the JTAG In-System-Programmability (ISP) commands.

## 22.2 JTAG extensions

TSTAT and $\overline{\text{TERR}}$ are two JTAG extension signals enabled by a JTAG command received over the four standard JTAG pins (TMS, TCK, TDI, and TDO). They are used to speed Program and Erase cycles by indicating status on PSD pins instead of having to scan the status out serially using the standard JTAG channel. See application note AN1153.

$\overline{\text{TERR}}$ indicates if an error has occurred when erasing a sector or programming in Flash memory. This signal goes Low (active) when an Error condition occurs, and stays Low until a specific JTAG command is executed or a Reset ($\overline{\text{RESET}}$) pulse is received after an "ISC_DISABLE" command.

TSTAT behaves the same as Ready/$\overline{\text{Busy}}$ (PE4) described in *Section 6.2.2: Ready/Busy (PE4)*. TSTAT is High when the µPSD3200 Family device is in Read mode (primary Flash memory and secondary Flash memory contents can be read). TSTAT is Low when Flash memory Program or Erase cycles are in progress, and also when data is being written to the secondary Flash memory.

TSTAT and $\overline{\text{TERR}}$ can be configured as open-drain type signals with a JTAG command.

*Note: The state of Reset ($\overline{\text{Reset}}$) does not interrupt (or prevent) JTAG operations if the JTAG signals are dedicated by an NVM Configuration bit (via PSDsoft). However, Reset ($\overline{\text{Reset}}$) prevents or interrupts JTAG operations if the JTAG Enable register (as shown in Table 21) is used to enable the JTAG signals.*

## 22.3 Security and Flash memory protection

When the security bit is set, the device cannot be read on a device programmer or through the JTAG port. When using the JTAG port, only a Full Chip Erase command is allowed.

All other Program, Erase and Verify commands are blocked. Full Chip Erase returns the device to a non-secured blank state. The Security bit can be set in PSDsoft.

All primary Flash memory and secondary Flash memory sectors can individually be sector protected against erasure. The sector protect bits can be set in PSDsoft.

**Table 54. JTAG port signals**

| Port E pin | JTAG signals | Description |
|---|---|---|
| PE0 | TMS | mode Select |
| PE1 | TCK | Clock |
| PE2 | TDI | Serial Data In |
| PE3 | TDO | Serial Data Out |
| PE4 | TSTAT | Status |
| PE5 | TERR | Error Flag |

# 23 Initial delivery state

When delivered from ST, the PSD device has all bits in the memory and PLDs set to 1. The PSD Configuration register bits are set to 0. The code, configuration, and PLD logic are loaded using the programming procedure. Information for programming the device is available directly from ST. Please contact your local sales representative.

# 24 Maximum rating

Stressing the device above the rating listed in the Absolute Maximum Ratings" table may cause permanent damage to the device. These are stress ratings only and operation of the device at these or any other conditions above those indicated in the Operating sections of this specification is not implied. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability. Refer also to the STMicroelectronics SURE Program and other relevant quality documents.

**Table 55. Absolute maximum ratings**

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $T_{STG}$ | Storage temperature | –65 | 150 | °C |
| $T_{LEAD}$ | Lead temperature during Soldering (20 seconds max.)(1) | | 235 | °C |
| $V_{IO}$ | Input and output voltage (Q = $V_{OH}$ or Hi-Z) | –0.6 | 4.0 | V |
| $V_{CC}$ | Supply voltage | –0.6 | 4.0 | V |
| $V_{PP}$ | Device programmer supply voltage | –0.6 | 13.5 | V |
| $V_{ESD}$ | Electrostatic discharge voltage (human body model)(2) | –2000 | 2000 | V |

1. IPC/JEDEC J-STD-020A.
2. JEDEC Std JESD22-A114A (C1=100 pF, R1=1500 Ω, R2=500 Ω).

# 25 DC and AC parameters

This section summarizes the operating and measurement conditions, and the DC and AC characteristics of the device:

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

The parameters in the DC and AC Characteristic tables that follow are derived from tests performed under the Measurement Conditions summarized in the relevant tables. Designers should check that the operating conditions in their circuit match the measurement conditions when relying on the quoted parameters.

The following are issues concerning the parameters presented:

- In the DC specification the supply current is given for different modes of operation. Before calculating the total power consumption, determine the percentage of time that the PSD is in each mode. Also, the supply power is considerably different if the Turbo bit is 0.
- The AC power component gives the PLD, Flash memory, and SRAM mA/MHz specification. *Figure 39* shows 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 39. PLD $I_{CC}$ / frequency consumption**

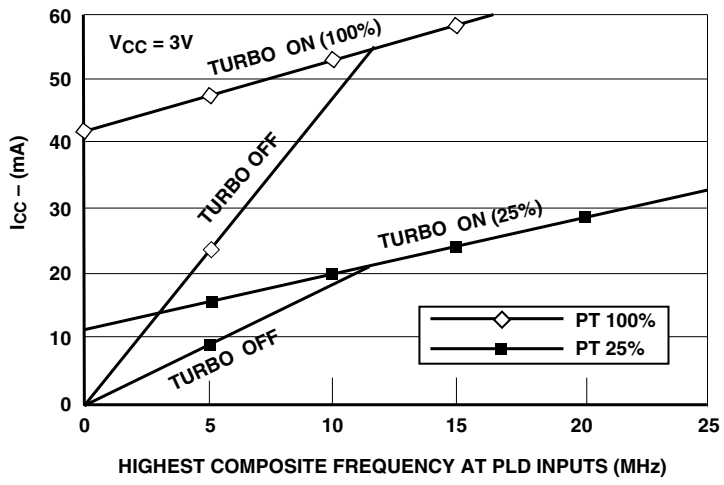


**Table 56. Example of PSD typical power calculation at $V_{CC}$ = 3.0 V (with Turbo mode On)**

| Conditions | | | |
|---|---|---|---|
| Highest composite PLD input frequency | | | |
| | (Freq PLD) | = 8MHz | |
| MCU ALE frequency (Freq ALE) | | = 4MHz | |
| | % Flash memory Access | = 80% | |
| | % SRAM access | = 15% | |
| | % I/O access | = 5% (no additional power above base) | |
| Operational modes | | | |
| | % Normal | = 10% | |
| | % Power-down mode | = 90% | |
| Number of product terms used | | | |
| | (from fitter report) | = 54 PT | |
| | % of total product terms | = 54/217 = 25% | |
| Turbo mode | | = ON | |
| **Calculation (using typical values)** | | | |
| | $I_{CC}$ total | = Ipwrdown x %pwrdown + %normal x ($I_{CC}$ (ac) + $I_{CC}$ (dc)) | |
| | | = Ipwrdown x %pwrdown + % normal x (%Flash x 1.2 mA/MHz x Freq ALE | |
| | | | + %SRAM x 0.8 mA/MHz x Freq ALE |
| | | | + % PLD x 1.1 mA/MHz x Freq PLD |
| | | | + #PT x 200 µA/PT) |
| | | = 50 µA x 0.90 + 0.1 x (0.8 x 1.2 mA/MHz x 4MHz | |
| | | | + 0.15 x 0.8 mA/MHz x 4MHz |
| | | | + 1.1 mA/MHz x 8MHz |
| | | | + 54 x 0.2 mA/PT) |
| | | = 45 µA + 0.1 x (3.84 + 0.48 + 8.8 + 10.8 mA) | |
| | | = 45 µA + 0.1 x 23.92 | |
| | | = 45 µA + 2.39 mA | |
| | | = 2.43 mA | |
| This is the operating power with no Flash memory Program or Erase cycles in progress. Calculation is based on $I_{OUT}$ = 0 mA. | | | |

**Table 57. Example of PSD typical power calculation at $V_{CC}$ = 3.0V (with Turbo mode Off)**

| Conditions | | | |
|---|---|---|---|
| Highest composite PLD input frequency | | | |
| | (Freq PLD) | = 8MHz | |
| MCU ALE frequency (Freq ALE) | | = 4MHz | |
| | % Flash memory Access | = 80% | |
| | % SRAM access | = 15% | |
| | % I/O access | = 5% (no additional power above base) | |
| Operational modes | | | |
| | % Normal | = 10% | |
| | % Power-down mode | = 90% | |
| Number of product terms used | | | |
| | (from fitter report) | = 54 PT | |
| | % of total product terms | = 54/217 = 25% | |
| | Turbo mode | = Off | |
| **Calculation (using typical values)** | | | |
| | $I_{CC}$ total | = Ipwrdown x %pwrdown + %normal x ($I_{CC}$ (ac) + $I_{CC}$ (dc)) | |
| | | = Ipwrdown x %pwrdown + % normal x (%Flash x 1.2 mA/MHz x Freq ALE | |
| | | | + %SRAM x 0.8 mA/MHz x Freq ALE |
| | | | + % PLD x (from graph using Freq PLD)) |
| | | = 50 µA x 0.90 + 0.1 x (0.8 x 1.2 mA/MHz x 4MHz | |
| | | | + 0.15 x 0.8 mA/MHz x 4MHz |
| | | | + 15 mA) |
| | | = 45 µA + 0.1 x (3.84 + 0.48 + 15) | |
| | | = 45 µA + 0.1 x 18.84 | |
| | | = 45µA + 1.94 mA | |
| | | = 1.98 mA | |
| This is the operating power with no Flash memory Program or Erase cycles in progress. Calculation is based on $I_{OUT}$ = 0 mA. | | | |

**Table 58. Operating conditions**

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | 2.7 | 3.6 | V |
| $T_A$ | Ambient operating temperature (industrial) | –40 | 85 | °C |
| | Ambient operating temperature (commercial) | 0 | 70 | °C |

**Table 59. AC signal letters for PLD timing(1)**

| Symbol | Signal |
|---|---|
| A | Address input |
| C | CEout output |
| D | Input data |
| E | E input |
| I | Interrupt input |
| L | ALE input |
| N | $\overline{\text{RESET}}$ input or output |
| P | Port Signal output |
| R | $\overline{\text{UDS}}$, $\overline{\text{LDS}}$, $\overline{\text{DS}}$, $\overline{\text{RD}}$, $\overline{\text{PSEN}}$ inputs |
| S | Chip Select input |
| T | R/$\overline{\text{W}}$ input |
| W | WR input |
| B | $V_{STBY}$ output |
| M | Output macrocell |

1. Example: $t_{AVLX}$ = Time from Address Valid to ALE Invalid.

**Table 60. AC signal behavior symbols for PLD timing (1)**

| Symbol | Signal behavior |
|---|---|
| t | Time |
| L | Logic level low or ALE |
| H | Logic level high |
| V | Valid |
| X | No longer a valid logic level |
| Z | Float |
| PW | Pulse width |

1. Example: $t_{AVLX}$ = Time from Address Valid to ALE Invalid.

**Table 61. AC measurement conditions(1)**

| Symbol | Parameter | Min. | Max. | Unit |
|---|---|---|---|---|
| $C_L$ | Load capacitance | 30 | | pF |

1. Output Hi-Z is defined as the point where data out is no longer driven.

**Table 62. Capacitance(1)**

| Symbol | Parameter | Test condition | Typ.(2) | Max. | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Input capacitance (for input pins) | $V_{IN}$ = 0 V | 4 | 6 | pF |
| $C_{OUT}$ | Output capacitance (for input/output pins) | $V_{OUT}$ = 0 V | 8 | 12 | pF |
| $C_{VPP}$ | Capacitance (for CNTL2/$V_{PP}$) | $V_{PP}$ = 0 V | 18 | 25 | pF |

1. Sampled only, not 100% tested.
2. Typical values are for $T_A$ = 25°C and nominal supply voltages.

**Figure 40. AC measurement I/O waveform**

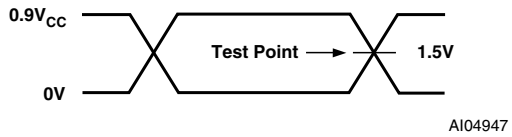



**Figure 41. AC measurement load circuit**

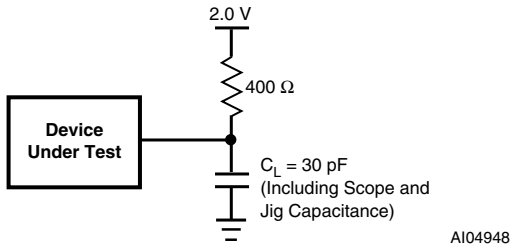



**Figure 42. Switching waveforms - key**

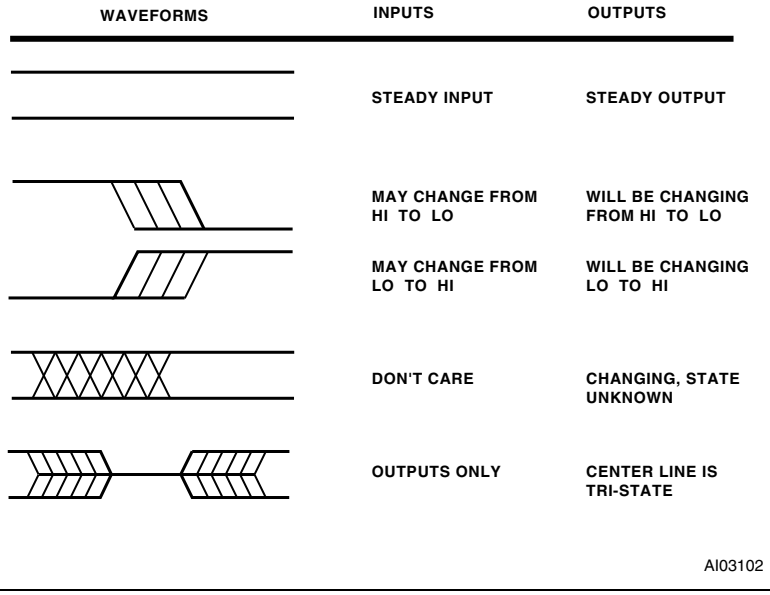


**Table 63. DC characteristics**

| Symbol | Parameter | | Conditions[1] | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $V_{IH}$ | High level input voltage | | 2.7 V < $V_{CC}$ < 3.6 V | 0.7$V_{CC}$ | | $V_{CC}$ +0.5 | V |
| $V_{IL}$ | Low level input voltage | | 2.7 V < $V_{CC}$ < 3.6 V | –0.5 | | 0.8 | V |
| $V_{IH1}$ | $\overline{RESET}$ high level input voltage | | (2) | 0.8$V_{CC}$ | | $V_{CC}$ +0.5 | V |
| $V_{IL1}$ | $\overline{RESET}$ low level Input Voltage | | (2) | –0.5 | | 0.2$V_{CC}$ –0.1 | V |
| $V_{HYS}$ | $\overline{RESET}$ pin hysteresis | | | 0.3 | | | V |
| $V_{LKO}$ | $V_{CC}$ (min) for Flash Erase and Program | | | 1.5 | | 2.3 | V |
| $V_{OL}$ | Output low voltage | | $I_{OL}$ = 20 µA, $V_{CC}$ = 2.7 V | | 0.01 | 0.1 | V |
| | | | $I_{OL}$ = 4 mA, $V_{CC}$ = 2.7 V | | 0.15 | 0.45 | V |
| $V_{OH}$ | Output high voltage | | $I_{OH}$ = –20 µA, $V_{CC}$ = 2.7 V | 2.6 | 2.69 | | V |
| | | | $I_{OH}$ = –1 mA, $V_{CC}$ = 2.7 V | 2.3 | 2.4 | | V |
| $I_{SB}$ | Standby supply current for Power-down mode | | $\overline{CSI}$ >$V_{CC}$ –0.3 V[3][4] | | 50 | 100 | µA |
| $I_{LI}$ | Input leakage current | | $V_{SS}$ < $V_{IN}$ < $V_{CC}$ | –1 | ±0.1 | 1 | µA |
| $I_{LO}$ | Output leakage current | | 0.45 < $V_{IN}$ < $V_{CC}$ | –10 | ±5 | 10 | µA |
| $I_{CC}$ (DC)[5] | Operating supply current | PLD only | PLD_TURBO = Off, f = 0 MHz[4] | | 0 | | µA/PT |
| | | | PLD_TURBO = On, f = 0 MHz | | 200 | 400 | µA/PT |
| | | Flash memory | During Flash memory Write/Erase only | | 10 | 25 | mA |
| | | | Read-only, f = 0 MHz | | 0 | 0 | mA |
| | | SRAM | f = 0 MHz | | 0 | 0 | mA |
| $I_{CC}$ (AC)[5] | PLD AC Adder | | | | (6) | | |
| | Flash memory AC adder | | | | 1.2 | 1.8 | mA/MHz |
| | SRAM AC adder | | | | 0.8 | 1.5 | mA/MHz |

1. Conditions (in addition to those given in *Table 58*, $V_{CC}$ = 4.5 to 5.5 V): $V_{SS}$ = 0 V; $C_L$ for Port 0, ALE and PSEN output is 100 pF; $C_L$ for other outputs is 80 pF
2. Reset ($\overline{RESET}$) has hysteresis. $V_{IL1}$ is valid at or below 0.2$V_{CC}$ –0.1. $V_{IH1}$ is valid at or above 0.8$V_{CC}$.
3. $\overline{CSI}$ deselected or internal PD is active.
4. PLD is in non-Turbo mode, and none of the inputs are switching.
5. $I_{OUT}$ = 0 mA
6. Please see *Figure 39* for the PLD current calculation.

**Figure 43. Input to Output Disable / Enable**

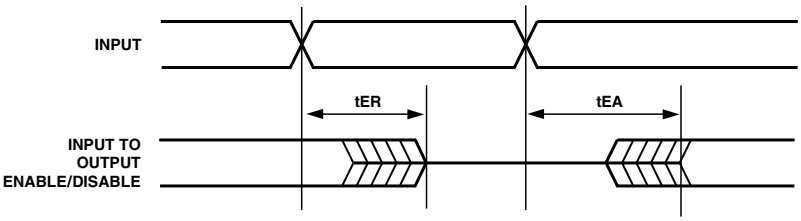



**Figure 44. Asynchronous $\overline{\text{RESET}}$ / Preset**

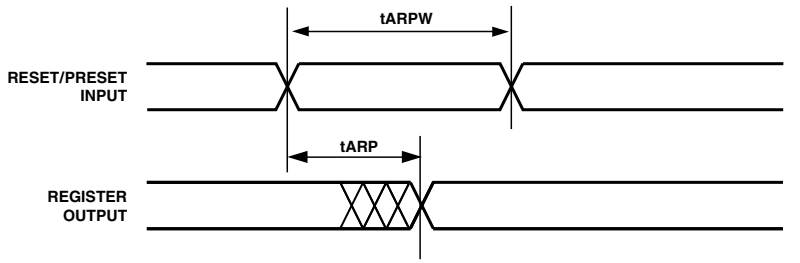



**Table 64. CPLD combinatorial timing**

| Symbol | Parameter | Conditions | -10 | | PT aloc | Turbo off | Unit |
|---|---|---|---|---|---|---|---|
| | | | Min | Max | | | |
| $t_{PD}$ | CPLD input pin/feedback to CPLD combinatorial output | | | 38 | + 4 | + 20 | ns |
| $t_{EA}$ | CPLD input to CPLD Output Enable | | | 43 | | + 20 | ns |
| $t_{ER}$ | CPLD input to CPLD Output Disable | | | 43 | | + 20 | ns |
| $t_{ARP}$ | CPLD register Clear or Preset delay | | | 38 | | + 20 | ns |
| $t_{ARPW}$ | CPLD register Clear or Preset pulse width | | 28 | | | + 20 | ns |
| $t_{ARD}$ | CPLD array delay | Any macrocell | | 23 | + 4 | | ns |

**Figure 45. Synchronous Clock mode timing – PLD**

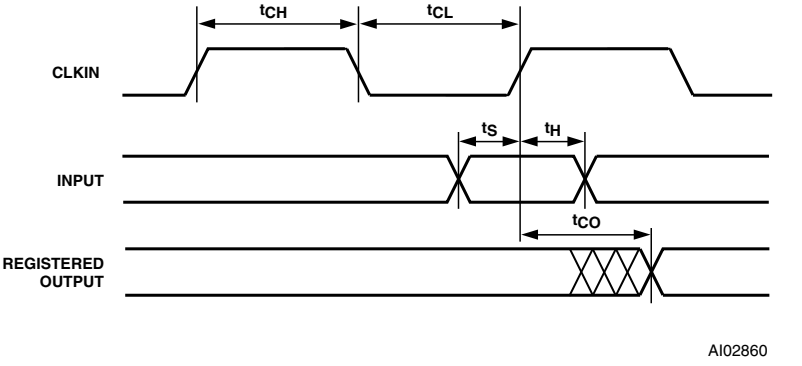


**Table 65. CPLD macrocell synchronous Clock mode timing**

| Symbol | Parameter | Conditions | -10 | | PT aloc | Turbo off | Unit |
|---|---|---|---|---|---|---|---|
| | | | Min | Max | | | |
| $f_{MAX}$ | Maximum frequency external feedback | $1/(t_S+t_{CO})$ | | 22.7 | | | MHz |
| | Maximum frequency internal feedback ($f_{CNT}$) | $1/(t_S+t_{CO}-10)$ | | 29.4 | | | MHz |
| | Maximum frequency pipelined data | $1/(t_{CH}+t_{CL})$ | | 45.0 | | | MHz |
| $t_S$ | Input setup time | | 18 | | + 4 | + 20 | ns |
| $t_H$ | Input hold time | | 0 | | | | ns |
| $t_{CH}$ | Clock high time | Clock input | 11 | | | | ns |
| $t_{CL}$ | Clock low time | Clock input | 11 | | | | ns |
| $t_{CO}$ | Clock to output delay | Clock Input | | 26 | | | ns |
| $t_{ARD}$ | CPLD array delay | Any macrocell | | 23 | + 4 | | ns |
| $t_{MIN}$ | Minimum clock period(1) | $t_{CH}+t_{CL}$ | 22 | | | | ns |

1. CLKIN (PD1) $t_{CLCL} = t_{CH} + t_{CL}$.

**Figure 46. Asynchronous Clock mode timing (Product Term Clock)**

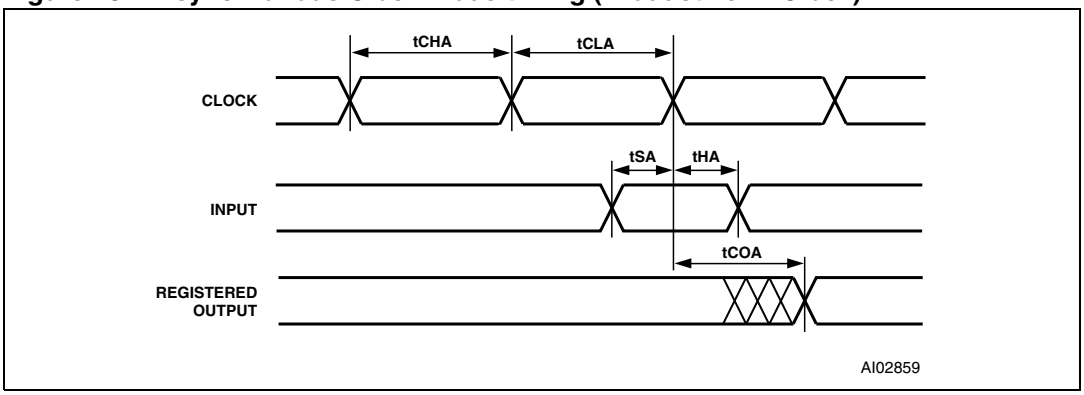



**Table 66. CPLD macrocell asynchronous Clock mode timing**

| Symbol | Parameter | Conditions | -10 | | PT aloc | Turbo off | Unit |
|---|---|---|---|---|---|---|---|
| | | | Min | Max | | | |
| $f_{MAXA}$ | Maximum frequency external feedback | $1/(t_{SA}+t_{COA})$ | | 23.8 | | | MHz |
| | Maximum frequency internal feedback ($f_{CNTA}$) | $1/(t_{SA}+t_{COA}-10)$ | | 31.25 | | | MHz |
| | Maximum frequency pipelined data | $1/(t_{CHA}+t_{CLA})$ | | 38.4 | | | MHz |
| $t_{SA}$ | Input setup time | | 8 | | + 4 | + 20 | ns |
| $t_{HA}$ | Input hold time | | 10 | | | | ns |
| $t_{CHA}$ | Clock high time | | 15 | | | + 20 | ns |
| $t_{CLA}$ | Clock low time | | 12 | | | + 20 | ns |

**Table 66. CPLD macrocell asynchronous Clock mode timing (continued)**

| Symbol | Parameter | Conditions | -10 | | PT aloc | Turbo off | Unit |
|---|---|---|---|---|---|---|---|
| | | | Min | Max | | | |
| $t_{COA}$ | Clock to output delay | | | 34 | | + 20 | ns |
| $t_{ARD}$ | CPLD array delay | Any macrocell | | 23 | + 4 | | ns |
| $t_{MINA}$ | Minimum clock period | $1/f_{CNTA}$ | 32 | | | | ns |

**Figure 47. Input macrocell timing (Product Term Clock)**

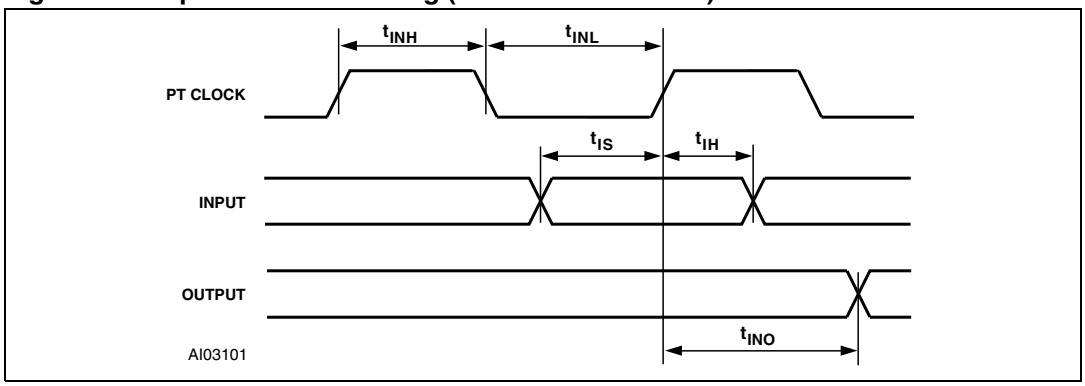
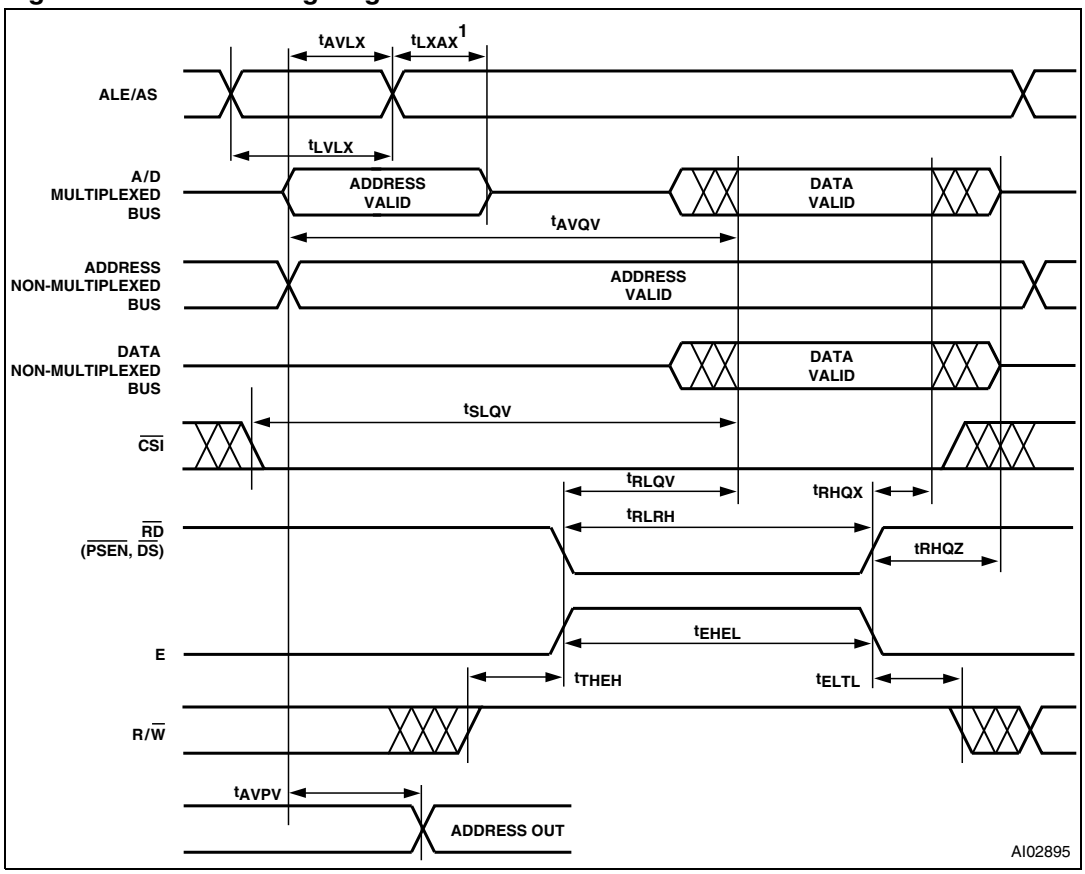



**Table 67. Input macrocell timing**

| Symbol | Parameter | Conditions | -10 | | PT aloc | Turbo off | Unit |
|---|---|---|---|---|---|---|---|
| | | | Min | Max | | | |
| $t_{IS}$ | Input setup time | (1) | 0 | | | | ns |
| $t_{IH}$ | Input hold time | (1) | 25 | | | + 20 | ns |
| $t_{INH}$ | NIB input high time | (1) | 13 | | | | ns |
| $t_{INL}$ | NIB input low time | (1) | 12 | | | | ns |
| $t_{INO}$ | NIB input to combinatorial delay | (1) | | 55 | + 4 | + 20 | ns |

1. Inputs from Port A, B, and C relative to register/latch clock from the PLD. ALE latch timings refer to $t_{AVLX}$ and $t_{LXAX}$.

**Figure 48. READ timing diagram**




1. $t_{AVLX}$ and $t_{LXAX}$ are not required for 80C251 in Page mode or 80C51XA in Burst mode.

**Table 68. READ timing**

| Symbol | Parameter | Conditions | -10 | | Turbo off | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Max | | |
| $t_{LVLX}$ | ALE or AS pulse width | | 22 | | | ns |
| $t_{AVLX}$ | Address setup time | (1) | 7 | | | ns |
| $t_{LXAX}$ | Address hold time | (1) | 8 | | | ns |
| $t_{AVQV}$ | Address valid to data valid | 2.7 V < $V_{CC}$ < 3.6V (1) | | 100 | + 20 | ns |
| | Address valid to data valid, 8031, 80251 | 3.0 V < $V_{CC}$ < 3.6 V(1) | | 90 | + 20 | ns |
| $t_{SLQV}$ | CS valid to data valid | | | 100 | | ns |
| $t_{RLQV}$ | $\overline{RD}$ to data valid 8-bit bus | (2) | | 35 | | ns |
| | $\overline{RD}$ or $\overline{PSEN}$ to data valid 8-bit bus, 8031, 80251 | (3) | | 45 | | ns |
| $t_{RHQX}$ | $\overline{RD}$ data hold time | (4) | 0 | | | ns |
| $t_{RLRH}$ | $\overline{RD}$ pulse width | | 36 | | | ns |
| $t_{RHQZ}$ | $\overline{RD}$ to data High-Z | (4) | | 38 | | ns |
| $t_{EHEL}$ | E pulse width | | 38 | | | ns |

**Table 68. READ timing (continued)**

| Symbol | Parameter | Conditions | -10 | | Turbo off | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Max | | |
| $t_{THEH}$ | R/$\overline{W}$ setup time to enable | | 10 | | | ns |
| $t_{ELTL}$ | R/$\overline{W}$ hold time after enable | | 0 | | | ns |
| $t_{AVPV}$ | Address input valid to address output delay | (5) | | 35 | | ns |

1. Any input used to select an internal PSD function.
2. $\overline{RD}$ timing has the same timing as $\overline{DS}$, $\overline{LDS}$, and $\overline{UDS}$ signals.
3. $\overline{RD}$ and $\overline{PSEN}$ have the same timing for 8031.
4. $\overline{RD}$ timing has the same timing as $\overline{DS}$, $\overline{LDS}$, $\overline{UDS}$, and $\overline{PSEN}$ signals.
5. In multiplexed mode latched address generated from ADIO delay to address output on any port.

**Figure 49. WRITE timing diagram**

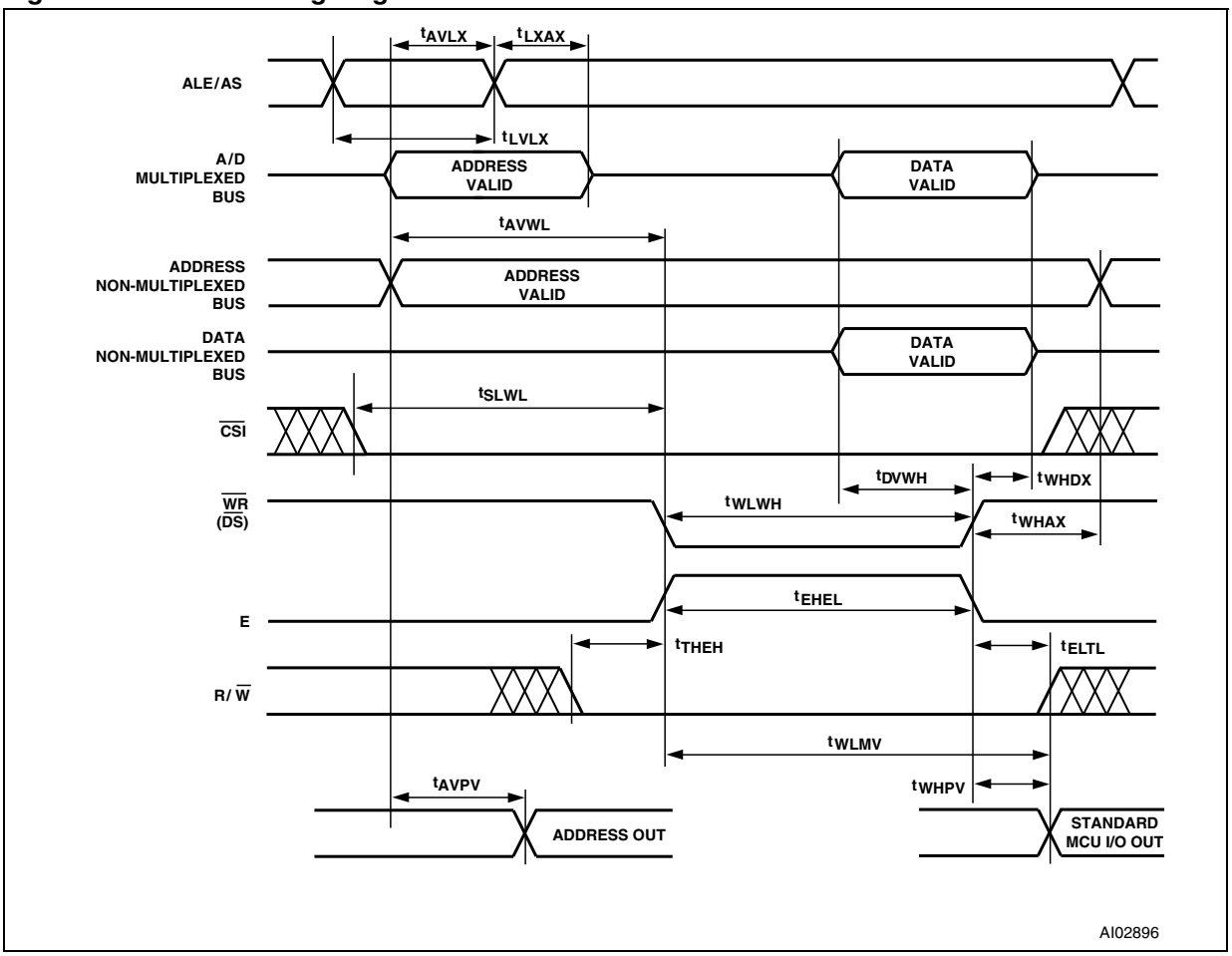



**Table 69. WRITE timing**

| Symbol | Parameter | Conditions | -10 | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| $t_{LVLX}$ | ALE or AS pulse width | | 22 | | |
| $t_{AVLX}$ | Address setup time | (1) | 7 | | ns |
| $t_{LXAX}$ | Address hold time | (1) | 8 | | ns |
| $t_{AVWL}$ | Address valid to leading edge of $\overline{WR}$ | (1)(2) | 15 | | ns |
| $t_{SLWL}$ | $\overline{CS}$ valid to leading edge of $\overline{WR}$ | (2) | 15 | | ns |
| $t_{DVWH}$ | $\overline{WR}$ data setup time | (2) | 40 | | ns |
| $t_{WHDX}$ | $\overline{WR}$ data hold time | (2)(3) | 5 | | ns |
| $t_{WLWH}$ | $\overline{WR}$ pulse width | (2) | 40 | | ns |
| $t_{WHAX1}$ | Trailing edge of $\overline{WR}$ to address invalid | (2) | 8 | | ns |
| $t_{WHAX2}$ | Trailing edge of $\overline{WR}$ to DPLD address invalid | (2)(4) | 0 | | ns |
| $t_{WHPV}$ | Trailing edge of $\overline{WR}$ to port output valid using I/O port data register | (2) | | 45 | ns |

**Table 69. WRITE timing (continued)**

| Symbol | Parameter | Conditions | -10 | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| $t_{DVMV}$ | Data valid to port output valid using macrocell register preset/clear | (2)(5) | | 65 | ns |
| $t_{AVPV}$ | Address input valid to address output delay | (6) | | 35 | ns |
| $t_{WLMV}$ | $\overline{WR}$ valid to port output valid using macrocell register preset/clear | (2)(7) | | 65 | ns |

1. Any input used to select an internal PSD function.
2. $\overline{WR}$ has the same timing as E, $\overline{LDS}$, $\overline{UDS}$, $\overline{WRL}$, and $\overline{WRH}$ signals.
3. tWHAX is 11 ns when writing to the output macrocell registers.
4. tWHAX2 is the address hold time for DPLD inputs that are used to generate Sector Select signals for internal PSD memory.
5. Assuming WRITE is active before data becomes valid.
6. In multiplexed mode, latched address generated from ADIO delay to address output on any port.
7. Assuming data is stable before active WRITE signal.

**Figure 50. Peripheral I/O READ timing diagram**

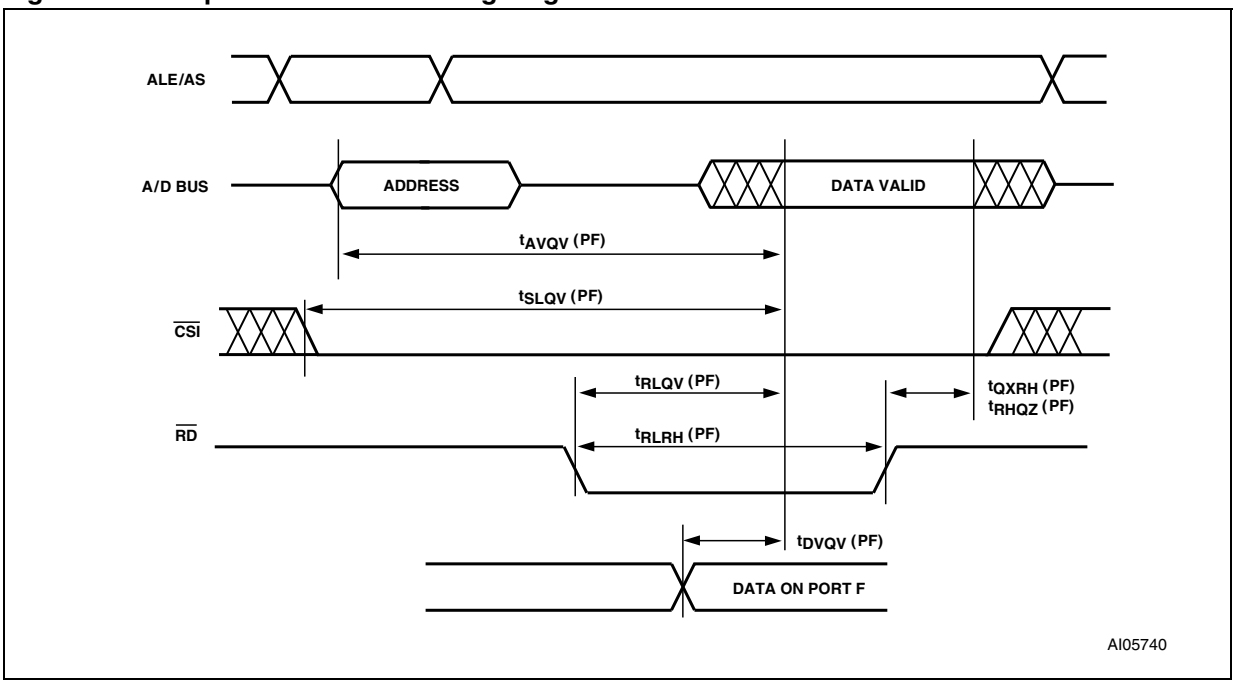



**Table 70. Port F Peripheral data mode READ timing**

| Symbol | Parameter | Conditions | -10 | | Turbo off | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Max | | |
| $t_{AVQV-PF}$ | Address valid to data valid | (1) | | 50 | + 20 | ns |
| $t_{SLQV-PF}$ | $\overline{CSI}$ valid to data valid | | | 50 | + 20 | ns |
| $t_{RLQV-PF}$ | $\overline{RD}$ to data valid | (2)(3) | | 35 | | ns |
| | $\overline{RD}$ to data valid 8031 mode | | | 45 | | ns |
| $t_{DVQV-PF}$ | Data in to data out valid | | | 34 | | ns |
| $t_{QXRH-PF}$ | $\overline{RD}$ data hold time | | 0 | | | ns |
| $t_{RLRH-PF}$ | $\overline{RD}$ pulse width | (2) | 35 | | | ns |
| $t_{RHQZ-PF}$ | $\overline{RD}$ to data High-Z | (2) | | 38 | | ns |

1. Any input used to select Port F Data Peripheral mode.
2. $\overline{RD}$ has the same timing as $\overline{DS}$, $\overline{LDS}$, $\overline{UDS}$, and $\overline{PSEN}$ (in 8031 combined mode).
3. Data is already stable on Port F.

**Figure 51. Peripheral I/O WRITE timing diagram**

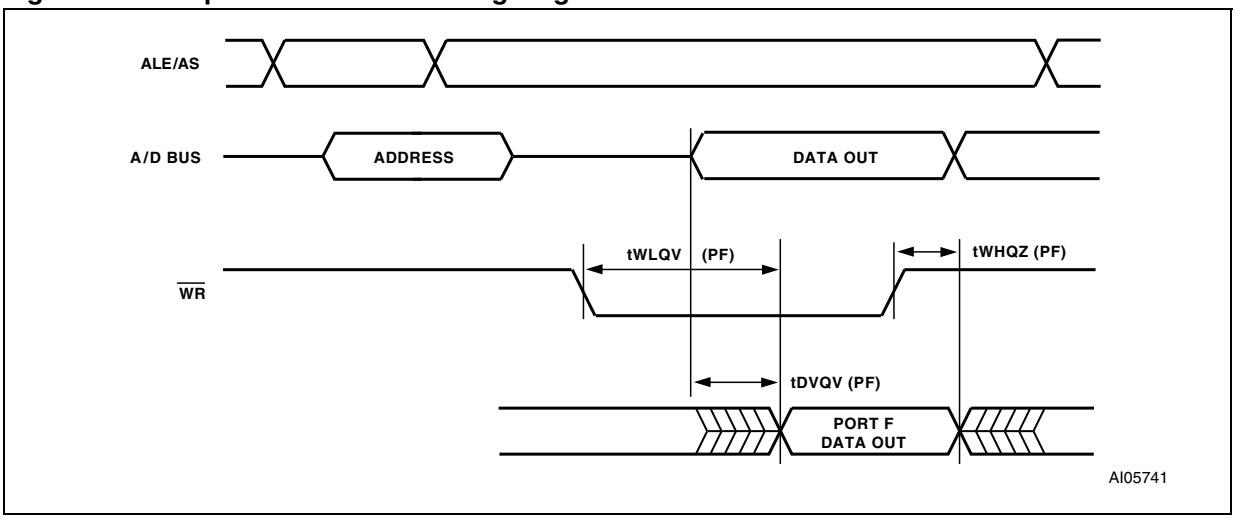



**Table 71. Port F peripheral data mode WRITE timing**

| Symbol | Parameter | Conditions | -10 | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| $t_{WLQV-PF}$ | $\overline{WR}$ to Data propagation delay | (1) | | 40 | ns |
| $t_{DVQV-PF}$ | Data to Port A Data propagation delay | (2) | | 35 | ns |
| $t_{WHQZ-PF}$ | $\overline{WR}$ invalid to Port A tri-state | (1) | | 33 | ns |

1. $\overline{WR}$ has the same timing as the E, $\overline{LDS}$, $\overline{UDS}$, $\overline{WRL}$, and $\overline{WRH}$ signals.
2. Data stable on ADIO pins to data on Port F.

**Table 72. Program, WRITE and Erase times**

| Symbol | Parameter | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| | Flash Program | | 8.5 | | s |
| | Flash Bulk Erase(1) (pre-programmed) | | 3 | 30 | s |
| | Flash Bulk Erase (not pre-programmed) | | 10 | | s |
| $t_{WHQV3}$ | Sector Erase (pre-programmed) | | 1 | 30 | s |
| $t_{WHQV2}$ | Sector Erase (not pre-programmed) | | 2.2 | | s |
| $t_{WHQV1}$ | Byte Program | | 14 | 1200 | µs |
| | Program / Erase cycles (per sector) | 100,000 | | | cycles |
| $t_{WHWLO}$ | Sector Erase timeout | | 100 | | µs |
| $t_{Q7VQV}$ | DQ7 valid to output (DQ7-DQ0) valid (data polling)(2)(3) | | | 30 | ns |

1. Programmed to all zero before erase.
2. The polling status, DQ7, is valid tQ7VQV time units before the data byte, DQ0-DQ7, is valid for reading.
3. DQ7 is DQ15 for Motorola MCU with 16-bit data bus.

**Figure 52. Reset ($\overline{\text{RESET}}$) timing diagram**

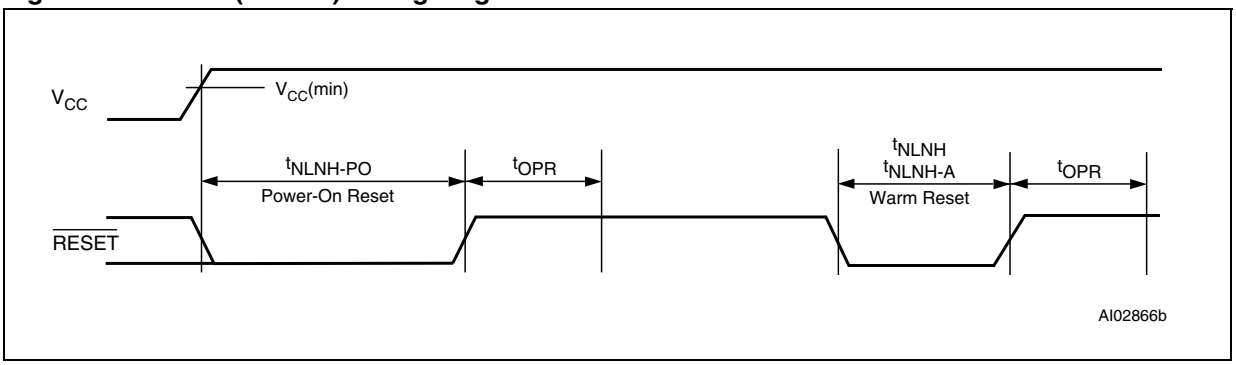



**Table 73. Reset ($\overline{\text{RESET}}$) timing**

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $t_{NLNH}$ | $\overline{\text{RESET}}$ active low time(1) | | 300 | | ns |
| $t_{NLNH-PO}$ | Power-on $\overline{\text{RESET}}$ active low time | | 1 | | ms |
| $t_{NLNH-A}$ | Warm $\overline{\text{RESET}}$ active low time(2) | | 25 | | µs |
| $t_{OPR}$ | $\overline{\text{RESET}}$ high to operational device | | | 300 | ns |

1. Reset ($\overline{\text{RESET}}$) does not reset Flash memory Program or Erase cycles.
2. Warm $\overline{\text{RESET}}$ aborts Flash memory Program or Erase cycles, and puts the device in Read mode.

**Table 74. Power-down timing**

| Symbol | Parameter | Conditions | -10 | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| $t_{LVDV}$ | ALE access time from Power-down | | | 128 | ns |
| $t_{CLWH}$ | Maximum delay from APD Enable to internal PDN valid signal | Using CLKIN (PD1) | 15 * $t_{CLCL}$ | | µs |

**Figure 53. ISC timing diagram**

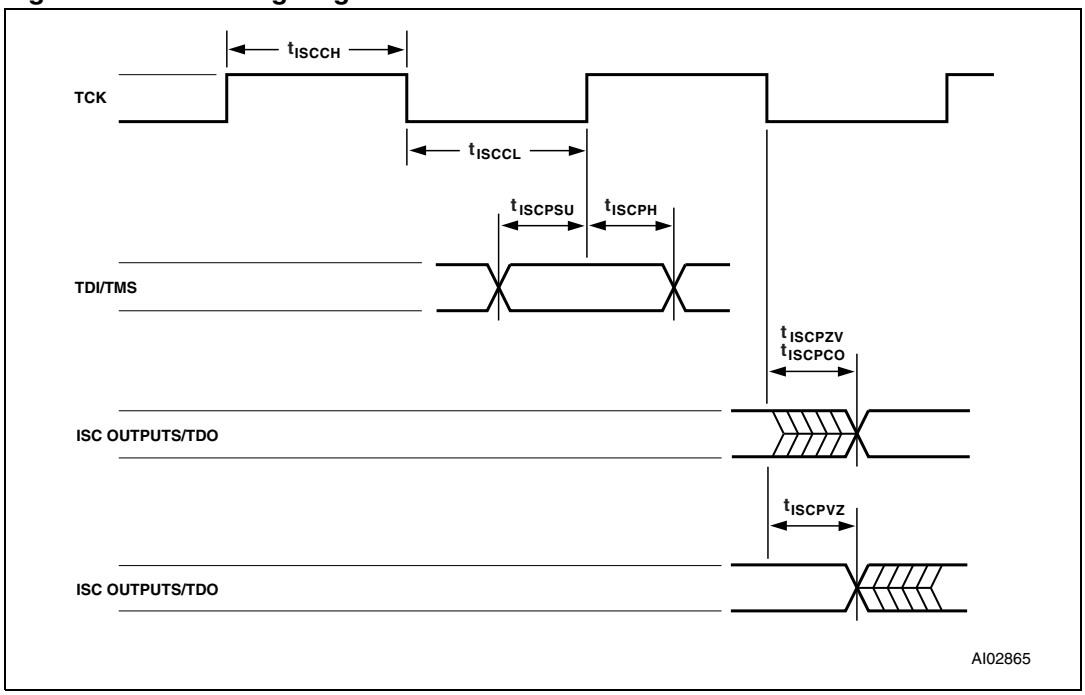



**Table 75. ISC timing**

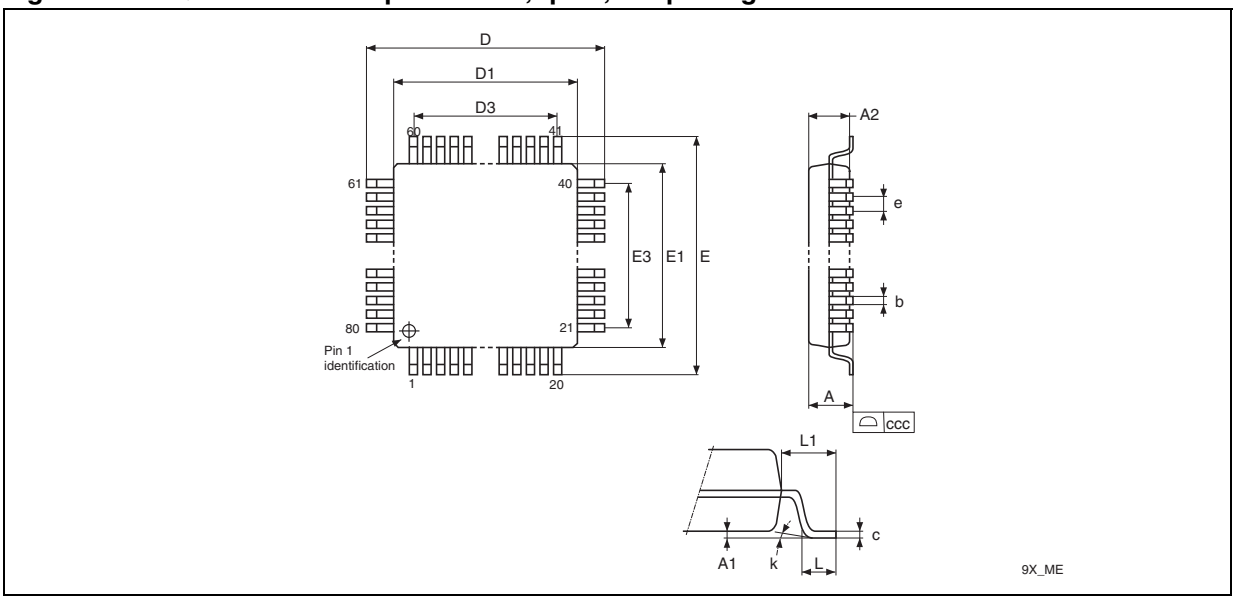
| Symbol | Parameter | Conditions | -10 | | Unit |
|---|---|---|---|---|---|
| | | | Min | Max | |
| $t_{ISCCF}$ | Clock (TCK, PC1) frequency (except for PLD) | (1) | | 15 | MHz |
| $t_{ISCCH}$ | Clock (TCK, PC1) high time (except for PLD) | (1) | 30 | | ns |
| $t_{ISCCL}$ | Clock (TCK, PC1) low time (except for PLD) | (1) | 30 | | ns |
| $t_{ISCCFP}$ | Clock (TCK, PC1) frequency (PLD only) | (2) | | 2 | MHz |
| $t_{ISCCHP}$ | Clock (TCK, PC1) high time (PLD only) | (2) | 240 | | ns |
| $t_{ISCCLP}$ | Clock (TCK, PC1) Low Time (PLD only) | (2) | 240 | | ns |
| $t_{ISCPSU}$ | ISC Port setup time | | 11 | | ns |
| $t_{ISCPH}$ | ISC Port hold up time | | 5 | | ns |
| $t_{ISCPCO}$ | ISC Port clock to output | | | 26 | ns |
| $t_{ISCPZV}$ | ISC Port high-Impedance to valid output | | | 26 | ns |
| $t_{ISCPVZ}$ | ISC Port valid output to high-Impedance | | | 26 | ns |

1. For non-PLD Programming, Erase or in ISC by-pass mode.
2. For Program or Erase PLD only.

# 26 Package mechanical information

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: www.st.com. ECOPACK® is an ST trademark.

**Figure 54. LQFP80 – 80-lead plastic thin, quad, flat package outline**




1. Drawing is not to scale.

**Table 76. LQFP80 – 80-lead plastic thin, quad, flat package mechanical data(1)**

| Symb | mm | | | inches | | |
|---|---|---|---|---|---|---|
| | Typ | Min | Max | Typ | Min | Max |
| A | – | – | 1.600 | – | – | 0.0630 |
| A1 | – | 0.050 | 0.150 | – | 0.0020 | 0.0060 |
| A2 | 1.400 | 1.350 | 1.450 | 0.0550 | 0.0530 | 0.0570 |
| b | 0.220 | 0.170 | 0.270 | 0.0090 | 0.0070 | 0.0110 |
| c | – | 0.090 | 0.200 | – | 0.0040 | 0.0080 |
| D | 14.000 | – | – | 0.5510 | – | – |
| D1 | 12.000 | – | – | 0.4720 | – | – |
| D2 | 9.500 | – | – | 0.3740 | – | – |
| E | 14.000 | – | – | 0.5510 | – | – |
| E1 | 12.000 | – | – | 0.4720 | – | – |
| E2 | 9.500 | – | – | 0.3740 | – | – |
| e | 0.500 | – | – | 0.0200 | – | – |

**Table 76. LQFP80 – 80-lead plastic thin, quad, flat package mechanical data(1) (continued)**

| Symb | mm | | | inches | | |
|---|---|---|---|---|---|---|
| | Typ | Min | Max | Typ | Min | Max |
| L | 0.600 | 0.45 | 0.750 | 0.0240 | 0.0180 | 0.0300 |
| L1 | 1.000 | – | – | 0.0390 | – | – |
| a | 3.5 | 0° | 7° | 3.5 | 0° | 7° |
| n | 80 | | | 80 | | |
| Nd | 20 | | | 20 | | |
| Ne | 20 | | | 20 | | |
| CP | – | – | 0.080 | – | – | 0.0030 |

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

# 27 Part numbering

**Table 77. Ordering information scheme**

Example: PSD42 5 6 G 6 V – 10 U I T

**Device Type**

PSD42 = Flash PSD with CPLD

**SRAM Size**

5 = 256 Kbit

**Flash memory Size**

6 = 8 Mbit

**I/O Count**

G = 52 I/O

**2nd Non-Volatile memory**

6 = 512 Kbit Flash memory

**Operating Voltage**

V = $V_{CC}$ = 2.7 to 3.6 V

**Speed**

10 = 100 ns

**Package**

U = ECOPACK LQFP80

**Temperature Range**

I = –40 to 85°C (Industrial)

**Option**

T = Tape & Reel Packing

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.

# Appendix A PIn assignments

**Table 78. LQFP80 pin connections**

| Pin num. | Pin assignments | Pin num. | Pin assignments | Pin num. | Pin assignments | Pin num. | Pin assignments |
|---|---|---|---|---|---|---|---|
| 1 | PD2 | 21 | PG0 | 41 | PC0 | 61 | PB0 |
| 2 | PD3 | 22 | PG1 | 42 | PC1 | 62 | PB1 |
| 3 | AD0 | 23 | PG2 | 43 | PC2 | 63 | PB2 |
| 4 | AD1 | 24 | PG3 | 44 | PC3 | 64 | PB3 |
| 5 | AD2 | 25 | PG4 | 45 | PC4 | 65 | PB4 |
| 6 | AD3 | 26 | PG5 | 46 | PC5 | 66 | PB5 |
| 7 | AD4 | 27 | PG6 | 47 | PC6 | 67 | PB6 |
| 8 | GND | 28 | PG7 | 48 | PC7 | 68 | PB7 |
| 9 | $V_{CC}$ | 29 | $V_{CC}$ | 49 | GND | 69 | $V_{CC}$ |
| 10 | AD5 | 30 | GND | 50 | GND | 70 | GND |
| 11 | AD6 | 31 | PF0 | 51 | PA0 | 71 | PE0 |
| 12 | AD7 | 32 | PF1 | 52 | PA1 | 72 | PE1 |
| 13 | AD8 | 33 | PF2 | 53 | PA2 | 73 | PE2 |
| 14 | AD9 | 34 | PF3 | 54 | PA3 | 74 | PE3 |
| 15 | AD10 | 35 | PF4 | 55 | PA4 | 75 | PE4 |
| 16 | AD11 | 36 | PF5 | 56 | PA5 | 76 | PE5 |
| 17 | AD12 | 37 | PF6 | 57 | PA6 | 77 | PE6 |
| 18 | AD13 | 38 | PF7 | 58 | PA7 | 78 | PE7 |
| 19 | AD14 | 39 | RESET | 59 | CNTL0 | 79 | PD0 |
| 20 | AD15 | 40 | CNTL2 | 60 | CNTL1 | 80 | PD1 |

# 28 Revision history

**Table 79. Document revision history**

| Date | Revision | Changes |
|---|---|---|
| 06-Aug-2001 | 1.0 | Document written |
| 13-Sep-01 | 1.1 | Package mechanical data updated |
| 14-Dec-01 | 1.2 | Added 100 ns specification; removed 90 and 120 ns specifications. Updated AC specification and Port C and F functions |
| 12-Aug-02 | 1.3 | Added 8-bit MCU Interface |
| 14-Nov-02 | 1.4 | Update characteristics and features ( *Table 68*, *Table 77*) |
| 17-Nov-03 | 1.5 | Modify Instructions, correct part numbering ( *Table 29*, *Table 77*) |
| 09-Mar-04 | 2.0 | Reformatted; corrected mechanical dimensions, ordering information ( *Table 76*, *Table 77*) |
| 25-Jun-04 | 3.0 | Reformatted; corrected typing error (READ Primary Flash Identifier) |
| 09-Jan-2009 | 4.0 | Updated status to 'not for new design'.<br>Document reformatted. Updated disclaimer text.<br>Change TQFP to LQFP package.<br>Added ECOPACK text in cover page, and updated package information in section *Section 26: Package mechanical information*.<br>Backup battery feature removed: updated *Features*, *Table 2* (pins PE6 and PE7), *Figure 3: PSD block diagram*, *Section 2.1: Memory*, *Section 11: SRAM*, *Section 19.24: Port E – functionality and structure*, and *Figure 35: Port E, F, and G structure*. Removed SRAM standby mode in *Section 20: Power management*. Removed $V_{STBY}$, $I_{STBY}$, $V_{STBYON}$, $V_{OH1}$, $V_{DF}$, and $I_{IDLE}$ from *Table 63*. Removed $V_{STBYON}$ timing table.<br>Updated *Figure 29*.<br>Regrouped sections AC/DC parameters, and DC and AC parameters. |
| 13-Feb-2009 | 5 | Corrected datasheet status.<br>Update title on cover page.<br>Updated *Figure 3: PSD block diagram* to remove power management unit and PE6.<br>Updated PE6 and PE7 in *Figure 20*, *Figure 21*, *Figure 22*, *Figure 23*, *Figure 24*, and *Figure 25*. |

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.



STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

**www.st.com**