



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





PSD834F2V

Flash PSD, 3.3V Supply, for 8-bit MCUs
2 Mbit + 256 Kbit Dual Flash Memories and 64 Kbit SRAM

PRELIMINARY DATA

FEATURES SUMMARY

- FLASH IN-SYSTEM PROGRAMMABLE (ISP) PERIPHERAL FOR 8-BIT MCUs
- 3.3 V \pm 10% SINGLE SUPPLY VOLTAGE
- 2 MBIT OF PRIMARY FLASH MEMORY (8 UNIFORM SECTORS, 32K x 8)
- 256 KBIT SECONDARY FLASH MEMORY (4 UNIFORM SECTORS)
- 64 KBIT OF BATTERY-BACKED SRAM
- OVER 3,000 GATES OF PLD: DPLD and CPLD
- 27 RECONFIGURABLE I/O PORTS
- ENHANCED JTAG SERIAL PORT
- PROGRAMMABLE POWER MANAGEMENT
- HIGH ENDURANCE:
 - 100,000 Erase/WRITE Cycles of Flash Memory
 - 1,000 Erase/WRITE Cycles of PLD

Figure 1. Packages

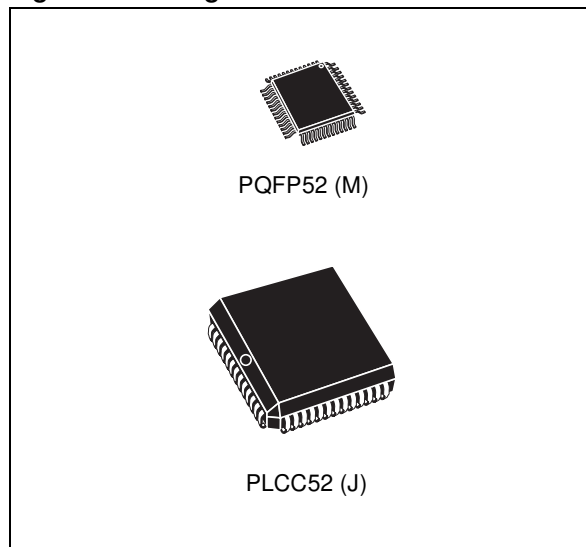


TABLE OF CONTENTS

SUMMARY DESCRIPTION 5

KEY FEATURES 6

PSD ARCHITECTURAL OVERVIEW 8

Memory 8

Page Register 8

PLDs 8

I/O Ports 8

MCU Bus Interface 8

JTAG Port 9

In-System Programming (ISP) 9

Power Management Unit (PMU) 9

DEVELOPMENT SYSTEM 10

PIN DESCRIPTION 11

PSD REGISTER DESCRIPTION AND ADDRESS OFFSET 13

DETAILED OPERATION 15

MEMORY BLOCKS 15

Primary Flash Memory and Secondary Flash memory Description 15

Memory Block Select Signals 15

INSTRUCTIONS 17

Power-down Instruction and Power-up Mode 17

READ 17

Programming Flash Memory 19

Erasing Flash Memory 21

Specific Features 22

SRAM 23

Sector Select and SRAM Select 23

Page Register 26

PLDS	27
The Turbo Bit in PSD	27
Decode PLD (DPLD)	29
Complex PLD (CPLD)	30
Output Macrocell (OMC)	32
Product Term Allocator	33
Input Macrocells (IMC)	35
MCU BUS INTERFACE	38
I/O PORTS	46
General Port Architecture	46
Port Operating Modes	47
MCU I/O Mode	47
PLD I/O Mode	47
Address Out Mode	47
Address In Mode	49
Data Port Mode	49
Peripheral I/O Mode	49
JTAG In-System Programming (ISP)	49
Port Configuration Registers (PCR)	50
Port Data Registers	51
Ports A and B – Functionality and Structure	52
Port D Structure	54
Port D – Functionality and Structure	54
Port D Structure	54
POWER MANAGEMENT	56
PLD Power Management	59
PSD Chip Select Input (CSI, PD2)	60
Input Clock	60
Input Control Signals	60

RESET TIMING AND DEVICE STATUS AT RESET	61
Warm Reset	61
I/O Pin, Register and PLD Status at Reset	61
Reset of Flash Memory Erase and Program Cycles.	61
PROGRAMMING IN-CIRCUIT USING THE JTAG SERIAL INTERFACE	63
Standard JTAG Signals.	63
JTAG Extensions.	64
Security and Flash memory Protection.	64
INITIAL DELIVERY STATE.	64
AC/DC PARAMETERS	65
MAXIMUM RATING.	68
DC AND AC PARAMETERS.	69
PACKAGE MECHANICAL	86
PART NUMBERING	91
PQFQ52 PIN ASSIGNMENTS	92
PLCC52 PIN ASSIGNMENTS.	93
REVISION HISTORY.	94

SUMMARY 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.

The CPLD in the PSD devices features an optimized macrocell logic architecture. The PSD 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 device includes a JTAG Serial Programming interface, to allow In-System Programming (ISP) of the *entire device*. This feature reduces development time, simplifies the manufacturing flow, and dramatically lowers the cost of field upgrades. Using ST's special Fast-JTAG programming, a design can be rapidly programmed into the PSD in as little as seven seconds.

The innovative PSD family solves key problems faced by designers when managing discrete Flash memory devices, such as:

- First-time In-System Programming (ISP)
- Complex address decoding
- Simultaneous READ and WRITE to the device.

The JTAG Serial Interface block allows In-System Programming (ISP), and eliminates the need for an external Boot EPROM, or an external programmer. To simplify Flash memory updates, program execution is performed from a secondary Flash memory while the primary Flash memory is being updated. This solution avoids the complicated hardware and software overhead necessary to implement IAP.

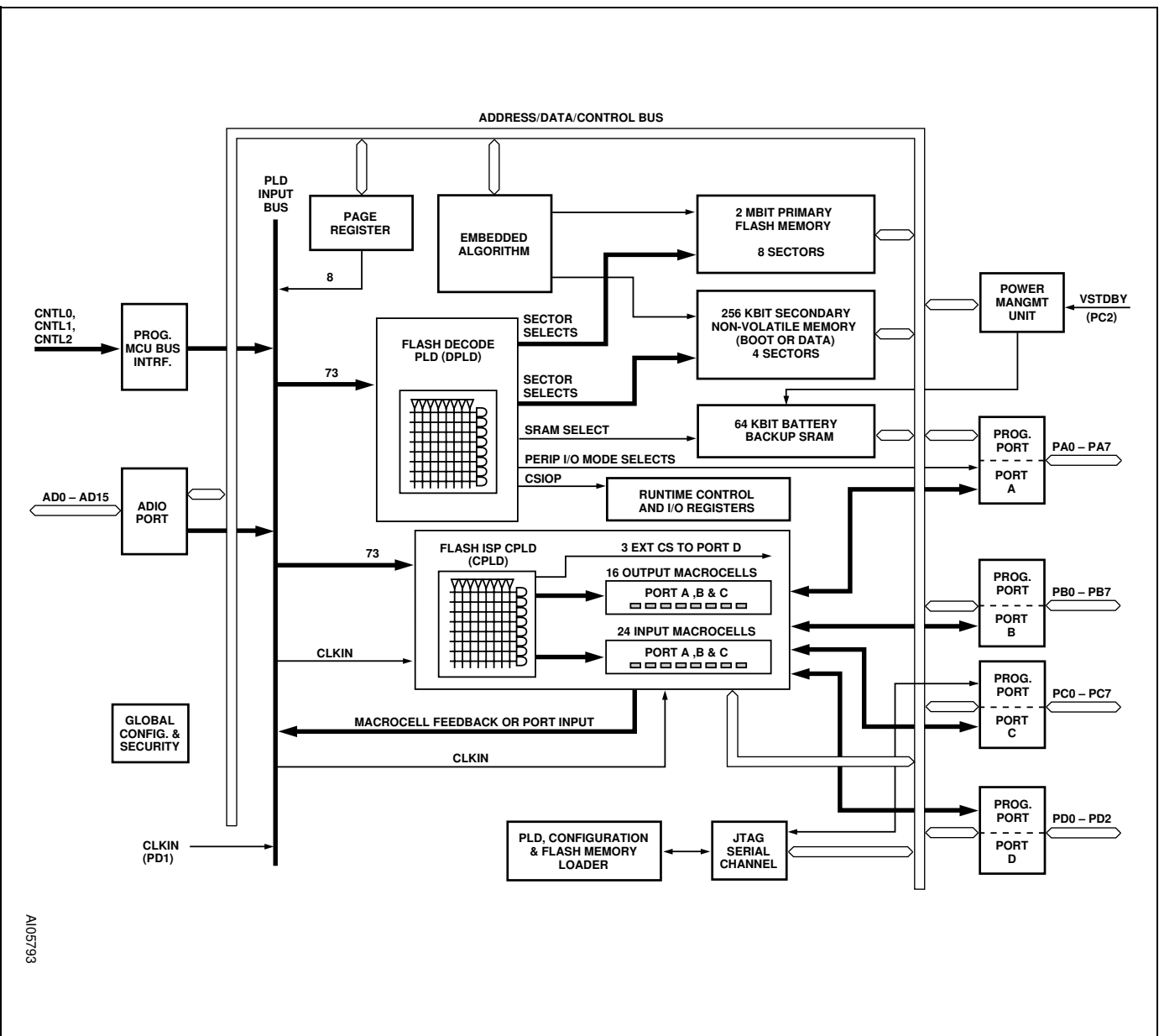
ST makes available a software development tool, PSDsoft Express, that generates ANSI-C compliant code for use with your target MCU. This code allows you to manipulate the non-volatile memory (NVM) within the PSD. Code examples are also provided for:

- Flash memory IAP via the UART of the host MCU
- Memory paging to execute code across several PSD memory pages
- Loading, reading, and manipulation of PSD macrocells by the MCU.

KEY FEATURES

- A simple interface to 8-bit microcontrollers that use either multiplexed or non-multiplexed busses. The bus interface logic uses the control signals generated by the microcontroller automatically when the address is decoded and a READ or WRITE is performed. A partial list of the MCU families supported include:
 - Intel 8031, 80196, 80186, 80C251, and 80386EX
 - Motorola 68HC11, 68HC16, 68HC12, and 683XX
 - Philips 8031 and 8051XA
 - Zilog Z80 and Z8
- Internal 2 Mbit Flash memory. This is the main Flash memory. It is divided into 8 equal-sized blocks that can be accessed with user-specified addresses.
- Internal secondary 256 Kbit Flash boot memory. It is divided into 4 equal-sized blocks that can be accessed with user-specified addresses. This secondary memory brings the ability to execute code and update the main Flash *concurrently*.
- Internal 64 Kbit SRAM. The SRAM's contents can be protected from a power failure by connecting an external battery.
- CPLD with 16 Output macrocells (OMCs) and 24 Input macrocells (IMCs). The CPLD may be used to efficiently implement a variety of logic functions for internal and external control. Examples include state machines, loadable shift registers, and loadable counters.
- Decode PLD (DPLD) that decodes address for selection of internal memory blocks.
- 27 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.
 - 16 of the I/O ports may be configured as open-drain outputs.
- Standby current as low as 25µA.
- Built-in JTAG compliant serial port allows full-chip In-System Programmability (ISP). With it, you can program a blank device or reprogram a device in the factory or the field.
- Internal page register that can be used to expand the microcontroller address space by a factor of 256.
- Internal programmable Power Management Unit (PMU) that supports a low power mode called Power Down Mode. The PMU can automatically detect a lack of microcontroller activity and put the PSD into Power-down mode.
- Erase/WRITE cycles:
 - Flash memory – 100,000 minimum
 - PLD – 1,000 minimum
 - Data Retention: 15 year minimum (for Main Flash memory, Boot, PLD and Configuration bits)

Figure 2. PSD Block Diagram



A105793

PSD ARCHITECTURAL OVERVIEW

PSD devices contain several major functional blocks. Figure 2 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.

Memory

Each of the memory blocks is briefly discussed in the following paragraphs. A more detailed discussion can be found in the section entitled “MEMORY BLOCKS” on page 15.

The 2 Mbit (256K x 8) Flash memory is the primary memory of the PSD. It is divided into 8 equally-sized sectors that are individually selectable.

The 256 Kbit (32K x 8) secondary Flash memory is divided into 4 equally-sized sectors. Each sector is individually selectable.

The 64 Kbit SRAM is intended for use as a scratch-pad memory or as an extension to the MCU SRAM. If an external battery is connected to Voltage Stand-by (VSTBY, PC2), data is retained in the event of power failure.

Each sector of memory 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.

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 sectors of the Flash memories into different memory spaces for IAP.

PLDs

The device contains two PLDs, the Decode PLD (DPLD) and the Complex PLD (CPLD), as shown in Table 1, each optimized for a different function. The functional partitioning of the PLDs reduces power consumption, optimizes cost/performance, and eases design entry.

Table 1. PLD I/O

Name	Inputs	Outputs	Product Terms
Decode PLD (DPLD)	73	17	42
Complex PLD (CPLD)	73	19	140

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. The CPLD has 16 Output Macrocells (OMC) and 3 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 the PMMR2. These registers are set by the MCU at run-time. There is a slight penalty to PLD propagation time when invoking the power management features.

I/O Ports

The PSD has 27 individually configurable I/O pins distributed over the four ports (Port A, B, C, and D). 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 C for In-System Programming (ISP).

Ports A and B can also be configured as a data port for a non-multiplexed bus.

MCU Bus Interface

PSD interfaces easily with most 8-bit MCUs that have either multiplexed or non-multiplexed address/data buses. The device is configured to respond to the MCU's control signals, which are also used as inputs to the PLDs. For examples, please see the section entitled “MCU Bus Interface Examples” on page 41.

JTAG Port

In-System Programming (ISP) can be performed through the JTAG signals on Port C. This serial interface allows complete programming of the entire PSD device. A blank device can be completely programmed. The JTAG signals (TMS, TCK, TSTAT, T $\overline{\text{ERR}}$, TDI, TDO) can be multiplexed with other functions on Port C. Table 2 indicates the JTAG pin assignments.

In-System Programming (ISP)

Using the JTAG signals on Port C, the entire PSD device can be programmed or erased without the use of the MCU. The primary Flash memory can also be programmed in-system by the MCU executing the programming algorithms out of the secondary memory, or SRAM. The secondary memory can be programmed the same way by executing out of the primary Flash memory. The PLD or other PSD Configuration blocks can be programmed through the JTAG port or a device programmer. Table 3 indicates which programming methods can program different functional blocks of the PSD.

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 sleep 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. Please see the section entitled "POWER MANAGEMENT" on page 56 for more details.

Table 2. JTAG Signals on Port C

Port C Pins	JTAG Signal
PC0	TMS
PC1	TCK
PC3	TSTAT
PC4	T $\overline{\text{ERR}}$
PC5	TDI
PC6	TDO

Table 3. Methods of Programming Different Functional Blocks of the PSD

Functional Block	JTAG Programming	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

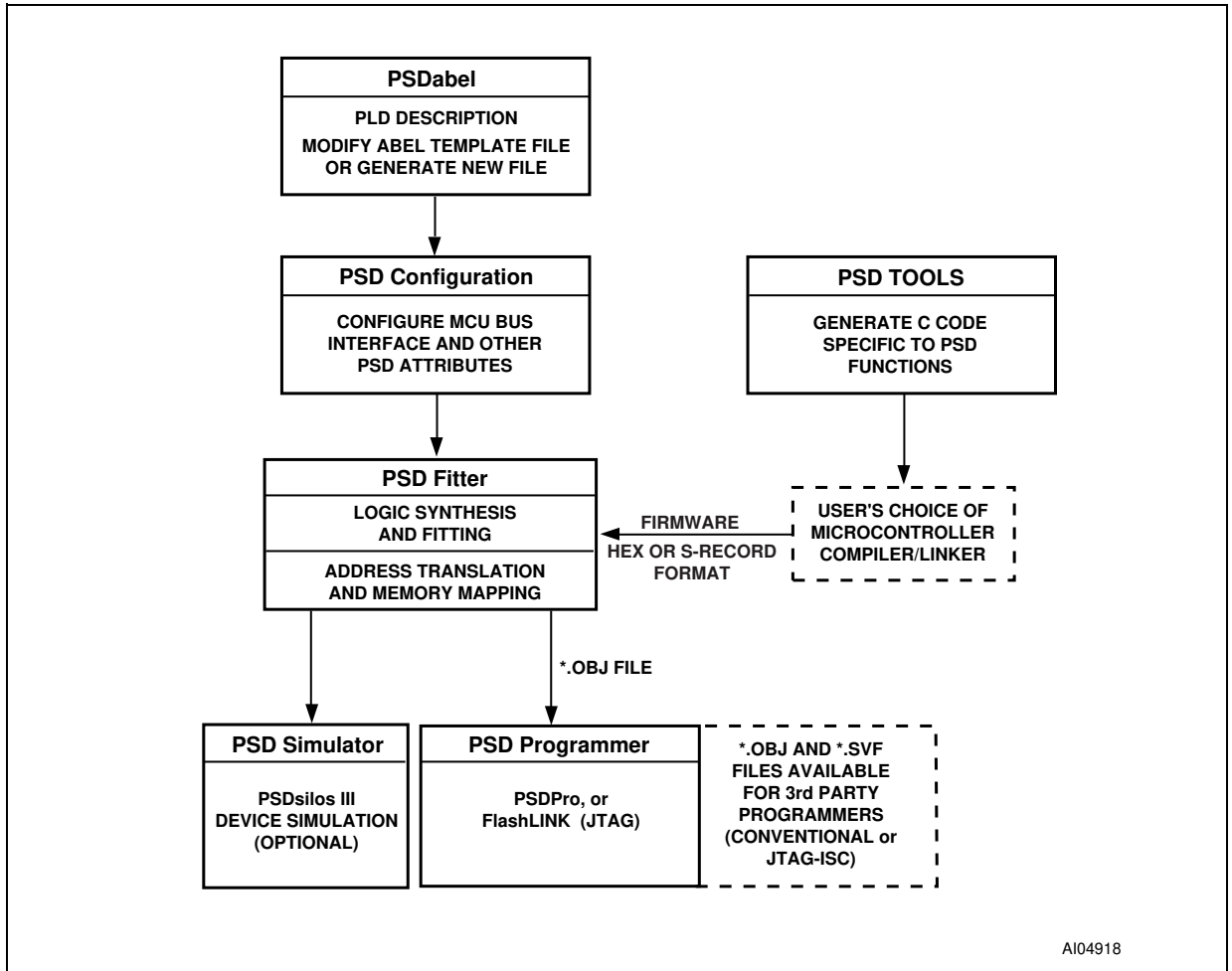
DEVELOPMENT SYSTEM

The PSD family is supported by PSDsoft Express, a Windows-based software development tool. 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 3. PSDsoft Express is available from our web site (the ad-

dress is given on the back page of this data sheet) or other distribution channels.

PSDsoft Express directly supports two low cost device programmers from 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 3. PSDsoft Express Development Tool



AI04918

PIN DESCRIPTION

Table 4 describes the signal names and signal functions of the PSD.

Table 4. Pin Description (for the PLCC52 package - Note 1)

Pin Name	Pin	Type	Description
ADIO0-7	30-37	I/O	This is the lower Address/Data port. Connect your MCU address or address/data bus according to the following rules: 1. 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. 2. If your MCU does not have a multiplexed address/data bus, or you are using an 80C251 in page mode, connect A0-A7 to this port. 3. If you are using an 80C51XA in burst mode, connect A4/D0 through A11/D7 to this port. 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 was selected. The addresses on this port are passed to the PLDs.
ADIO8-15	39-46	I/O	This is the upper Address/Data port. Connect your MCU address or address/data bus according to the following rules: 1. If your MCU has a multiplexed address/data bus where the data is multiplexed with the lower address bits, connect A8-A15 to this port. 2. If your MCU does not have a multiplexed address/data bus, connect A8-A15 to this port. 3. If you are using an 80C251 in page mode, connect AD8-AD15 to this port. 4. If you are using an 80C51XA in burst mode, connect A12/D8 through A19/D15 to this port. 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 was selected. The addresses on this port are passed to the PLDs.
CNTL0	47	I	The following control signals can be connected to this port, based on your MCU: 1. \overline{WR} – active Low Write Strobe input. 2. R_W – active High READ/active Low WRITE input. This port is connected to the PLDs. Therefore, these signals can be used in decode and other logic equations.
CNTL1	50	I	The following control signals can be connected to this port, based on your MCU: 1. \overline{RD} – active Low Read Strobe input. 2. \overline{E} – E clock input. 3. \overline{DS} – active Low Data Strobe input. 4. \overline{PSEN} – connect \overline{PSEN} to this port when it is being used as an active Low READ signal. For example, when the 80C251 outputs more than 16 address bits, \overline{PSEN} is actually the READ signal. This port is connected to the PLDs. Therefore, these signals can be used in decode and other logic equations.
CNTL2	49	I	This port can be used to input the \overline{PSEN} (Program Select Enable) signal from any MCU that uses this signal for code exclusively. If your MCU does not output a Program Select Enable signal, this port can be used as a generic input. This port is connected to the PLDs.
\overline{Reset}	48	I	Resets I/O Ports, PLD macrocells and some of the Configuration Registers. Must be Low at Power-up.

Pin Name	Pin	Type	Description
PA0 PA1 PA2 PA3 PA4 PA5 PA6 PA7	29 28 27 25 24 23 22 21	I/O	<p>These pins make up Port A. These port pins are configurable and can have the following functions:</p> <ol style="list-style-type: none"> 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellAB0-7) outputs. 3. Inputs to the PLDs. 4. Latched address outputs (see Table 5). 5. Address inputs. For example, PA0-3 could be used for A0-A3 when using an 80C51XA in burst mode. 6. As the data bus inputs D0-D7 for non-multiplexed address/data bus MCUs. 7. D0/A16-D3/A19 in M37702M2 mode. 8. Peripheral I/O mode. <p>Note: PA0-PA3 can only output CMOS signals with an option for high slew rate. However, PA4-PA7 can be configured as CMOS or Open Drain Outputs.</p>
PB0 PB1 PB2 PB3 PB4 PB5 PB6 PB7	7 6 5 4 3 2 52 51	I/O	<p>These pins make up Port B. These port pins are configurable and can have the following functions:</p> <ol style="list-style-type: none"> 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellAB0-7 or McellBC0-7) outputs. 3. Inputs to the PLDs. 4. Latched address outputs (see Table 5). <p>Note: PB0-PB3 can only output CMOS signals with an option for high slew rate. However, PB4-PB7 can be configured as CMOS or Open Drain Outputs.</p>
PC0	20	I/O	<p>PC0 pin of Port C. This port pin can be configured to have the following functions:</p> <ol style="list-style-type: none"> 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellBC0) output. 3. Input to the PLDs. 4. TMS Input² for the JTAG Serial Interface. <p>This pin can be configured as a CMOS or Open Drain output.</p>
PC1	19	I/O	<p>PC1 pin of Port C. This port pin can be configured to have the following functions:</p> <ol style="list-style-type: none"> 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellBC1) output. 3. Input to the PLDs. 4. TCK Input² for the JTAG Serial Interface. <p>This pin can be configured as a CMOS or Open Drain output.</p>
PC2	18	I/O	<p>PC2 pin of Port C. This port pin can be configured to have the following functions:</p> <ol style="list-style-type: none"> 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellBC2) output. 3. Input to the PLDs. 4. VSTBY – SRAM stand-by voltage input for SRAM battery backup. <p>This pin can be configured as a CMOS or Open Drain output.</p>
PC3	17	I/O	<p>PC3 pin of Port C. This port pin can be configured to have the following functions:</p> <ol style="list-style-type: none"> 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellBC3) output. 3. Input to the PLDs. 4. $\overline{\text{TSTAT}}$ output² for the JTAG Serial Interface. 5. Ready/Busy output for parallel In-System Programming (ISP). <p>This pin can be configured as a CMOS or Open Drain output.</p>
PC4	14	I/O	<p>PC4 pin of Port C. This port pin can be configured to have the following functions:</p> <ol style="list-style-type: none"> 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellBC4) output. 3. Input to the PLDs. 4. $\overline{\text{TERR}}$ output² for the JTAG Serial Interface. 5. Battery-on Indicator (VBATON). Goes High when power is being drawn from the external battery. <p>This pin can be configured as a CMOS or Open Drain output.</p>

Pin Name	Pin	Type	Description
PC5	13	I/O	PC5 pin of Port C. This port pin can be configured to have the following functions: 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellBC5) output. 3. Input to the PLDs. 4. TDI input ² for the JTAG Serial Interface. This pin can be configured as a CMOS or Open Drain output.
PC6	12	I/O	PC6 pin of Port C. This port pin can be configured to have the following functions: 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellBC6) output. 3. Input to the PLDs. 4. TDO output ² for the JTAG Serial Interface. This pin can be configured as a CMOS or Open Drain output.
PC7	11	I/O	PC7 pin of Port C. This port pin can be configured to have the following functions: 1. MCU I/O – write to or read from a standard output or input port. 2. CPLD macrocell (McellBC7) output. 3. Input to the PLDs. 4. DBE – active Low Data Byte Enable input from 68HC912 type MCUs. This pin can be configured as a CMOS or Open Drain output.
PD0	10	I/O	PD0 pin of Port D. This port pin can be configured to have the following functions: 1. ALE/AS input latches address output from the MCU. 2. MCU I/O – write or read from a standard output or input port. 3. Input to the PLDs. 4. CPLD output (External Chip Select).
PD1	9	I/O	PD1 pin of Port D. This port pin can be configured to have the following functions: 1. MCU I/O – write to or read from a standard output or input port. 2. Input to the PLDs. 3. CPLD output (External Chip Select). 4. CLKIN – clock input to the CPLD macrocells, the APD Unit's Power-down counter, and the CPLD AND Array.
PD2	8	I/O	PD2 pin of Port D. This port pin can be configured to have the following functions: 1. MCU I/O – write to or read from a standard output or input port. 2. Input to the PLDs. 3. CPLD output (External Chip Select). 4. PSD Chip Select Input (CSI). When Low, the MCU can access the PSD memory and I/O. When High, the PSD memory blocks are disabled to conserve power.
Vcc	15, 38		Supply Voltage
GND	1, 16, 26		Ground pins

Note: 1. The pin numbers in this table are for the PLCC package only. See the package information, on page 90 onwards, for pin numbers on other package types.

2. These functions can be multiplexed with other functions.

PSD REGISTER DESCRIPTION AND ADDRESS OFFSET

Table 6 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 section gives a more detailed description.

Table 5. I/O Port Latched Address Output Assignments (Note 1)

MCU	Port A		Port B	
	Port A (3:0)	Port A (7:4)	Port B (3:0)	Port B (7:4)
8051XA (8-bit)	N/A	Address a7-a4	Address a11-a8	N/A
80C251 (page mode)	N/A	N/A	Address a11-a8	Address a15-a12
All other 8-bit multiplexed	Address a3-a0	Address a7-a4	Address a3-a0	Address a7-a4
8-bit non-multiplexed bus	N/A	N/A	Address a3-a0	Address a7-a4

Note: 1. See the section entitled "I/O PORTS", on page 46, on how to enable the Latched Address Output function.
2. N/A = Not Applicable

Table 6. Register Address Offset

Register Name	Port A	Port B	Port C	Port D	Other ¹	Description
Data In	00	01	10	11		Reads Port pin as input, MCU I/O input mode
Control	02	03				Selects mode between MCU I/O or Address Out
Data Out	04	05	12	13		Stores data for output to Port pins, MCU I/O output mode
Direction	06	07	14	15		Configures Port pin as input or output
Drive Select	08	09	16	17		Configures Port pins as either CMOS or Open Drain on some pins, while selecting high slew rate on other pins.
Input Macrocell	0A	0B	18			Reads Input Macrocells
Enable Out	0C	0D	1A	1B		Reads the status of the output enable to the I/O Port driver
Output Macrocells AB	20	20				READ – reads output of macrocells AB WRITE – loads macrocell flip-flops
Output Macrocells BC		21	21			READ – reads output of macrocells BC WRITE – loads macrocell flip-flops
Mask Macrocells AB	22	22				Blocks writing to the Output Macrocells AB
Mask Macrocells BC		23	23			Blocks writing to the Output Macrocells BC
Primary Flash Protection					C0	Read only – Primary Flash Sector Protection
Secondary Flash memory 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.

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

DETAILED OPERATION

As shown in Figure 2, the PSD consists of six major types of functional blocks:

- Memory Blocks
- PLD Blocks
- MCU Bus Interface
- I/O Ports
- Power Management Unit (PMU)
- JTAG Interface

The functions of each block are described in the following sections. Many of the blocks perform multiple functions, and are user configurable.

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 Express.

Primary Flash Memory and Secondary Flash memory Description

The primary Flash memory is divided evenly into eight equal sectors. The secondary Flash memory is divided into four equal sectors. 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. 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 Ready/Busy (PC3). This pin is set up using PSDsoft Express Configuration.

Memory Block Select Signals

The DPLD generates the Select signals for all the internal memory blocks (see the section entitled “PLDS”, on page 27). Each of the eight sectors of

the primary Flash memory has a Select signal (FS0-FS7) which can contain up to three product terms. Each of the four 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, these flexible Select signals allow dynamic re-mapping of sectors from one memory space to the other.

Ready/Busy (PC3). This signal can be used to output the Ready/Busy status of the PSD. The output on Ready/Busy (PC3) is a 0 (Busy) when Flash memory is being written to, or when Flash memory is being erased. The output is a 1 (Ready) when no WRITE or Erase cycle is in progress.

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 7.

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

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

Table 7. Instructions

Instruction	FS0-FS7 or CSBOOT0-CSBOOT3	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7
READ ⁵	1	"Read" RD @ RA						
Read Main Flash ID ⁶	1	AAh@ X555h	55h@ XAAAh	90h@ X555h	Read identifier (A6,A1,A0 = 0,0,1)			
Read Sector Protection ^{6,8,13}	1	AAh@ X555h	55h@ XAAAh	90h@ X555h	Read identifier (A6,A1,A0 = 0,1,0)			
Program a Flash Byte ¹³	1	AAh@ X555h	55h@ XAAAh	A0h@ X555h	PD@ PA			
Flash Sector Erase ^{7,13}	1	AAh@ X555h	55h@ XAAAh	80h@ X555h	AAh@ XAAAh	55h@ XAAAh	30h@ SA	30h ⁷ @ next SA
Flash Bulk Erase ¹³	1	AAh@ X555h	55h@ XAAAh	80h@ X555h	AAh@ XAAAh	55h@ XAAAh	10h@ X555h	
Suspend Sector Erase ¹¹	1	B0h@ XXXXh						
Resume Sector Erase ¹²	1	30h@ XXXXh						
Reset ⁶	1	F0h@ XXXXh						
Unlock Bypass	1	AAh@ X555h	55h@ XAAAh	20h@ X555h				
Unlock Bypass Program ⁹	1	A0h@ XXXXh	PD@ PA					
Unlock Bypass Reset ¹⁰	1	90h@ XXXXh	00h@ XXXXh					

- Note: 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-FS7 or CSBOOT0-CSBOOT3) of the sector to be erased, or verified, must be Active (High).
3. Sector Select (FS0 to FS7 or CSBOOT0 to CSBOOT3) signals are active High, and are defined in PSDsoft Express.
4. Only address bits A11-A0 are used in instruction decoding.
5. No Unlock or instruction cycles are required when the device is in the READ Mode
6. 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 (DQ5/DQ13) Bit goes High.
7. Additional sectors to be erased must be written at the end of the Sector Erase instruction within 80 μ s.
8. 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)
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.
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 MCU cannot invoke these instructions while executing code from the same Flash memory as that for which the instruction is intended. The MCU must fetch, for example, the code from the secondary Flash memory when reading the Sector Protection Status of the primary Flash memory.

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 time-out period. Some instructions are structured to include READ operations after the initial WRITE operations.

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

The PSD supports the instructions summarized in Table 7:

Flash memory:

- Erase memory by chip or sector
- Suspend or resume sector erase
- Program a Byte
- Reset to READ Mode
- Read primary Flash Identifier value
- Read Sector Protection Status
- Bypass

These instructions are detailed in Table 7. For efficient decoding of the instructions, the first two bytes of an instruction are the coded cycles and are followed by an instruction byte or confirmation byte. The coded cycles consist of writing the data AAh to address X555h during the first cycle and data 55h to address XAAAh during the second cycle. Address signals A15-A12 are Don't Care during the instruction WRITE cycles. However, the appropriate Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) must be selected.

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

Power-down Instruction and Power-up Mode

Power-up Mode. The PSD internal logic is reset upon Power-up to the READ Mode. Sector Select (FS0-FS7 and CSBOOT0-CSBOOT3) must be

held Low, and Write Strobe (\overline{WR} , CNTL0) High, during Power-up for maximum security of the data contents and to remove the possibility of a byte being written on the first edge of Write Strobe (\overline{WR} , CNTL0). Any WRITE cycle initiation is locked when V_{CC} is below V_{LKO} .

READ

Under typical conditions, the MCU may read the primary Flash memory or the 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.

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 7). 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.

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 7). During the READ operation, address Bits A6, A1, and A0 must be '0,' '0,' and '1,' respectively, and the appropriate Sector Select (FS0-FS7) must be High. The identifier for the device is E7h.

Read Memory Sector Protection Status. The primary Flash memory Sector Protection Status is read with an instruction composed of 4 operations: 3 specific WRITE operations and a READ operation (see Table 7). During the READ operation, address Bits A6, A1, and A0 must be '0,' '1,' and '0,' respectively, while Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) designates the Flash memory sector whose protection has to be verified. 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 also be read by the MCU accessing the Flash Protection registers in PSD I/O space. See the section entitled "Flash Memory Sector Protect", on page 22, for register definitions.

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 8. The status bits can be read as many times as needed. 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 the section entitled “Programming Flash Memory”, on page 19, for details.

Data Polling Flag (DQ7). When erasing or programming in Flash memory, the Data Polling Flag (DQ7) Bit outputs the complement of the bit being entered for programming/writing on the DQ7 Bit. Once the Program instruction or the WRITE operation is completed, the true logic value is read on the Data Polling Flag (DQ7) Bit (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 Flag (DQ7) Bit outputs a '0.' After completion of the cycle, the Data Polling Flag (DQ7) Bit outputs the last bit programmed (it is a '1' after erasing).
- If the byte 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 Flag (DQ7) Bit is reset to '0' for about 100µs, and then returns to the previous addressed byte. No erasure is performed.

Toggle Flag (DQ6). The PSD offers another way for determining when the Flash memory Program cycle is completed. During the internal WRITE operation and when either the FS0-FS7 or CSBOOT0-CSBOOT3 is true, the Toggle Flag (DQ6) Bit toggles from '0' to '1' and '1' to '0' on subsequent attempts to read any byte of the memory.

When the internal cycle is complete, the toggling stops and the data READ on the Data Bus D0-D7 is the addressed memory byte. 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 (DQ6) Bit is effective after the fourth WRITE pulse (for a Program instruction) or after the sixth WRITE pulse (for an Erase instruction).
- If the byte 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 (DQ6) Bit toggles to '0' for about 100µs and then returns to the previous addressed byte.

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

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

In case of an error in a Flash memory Sector Erase or Byte Program cycle, the Flash memory sector in which the error occurred or to which the programmed byte belongs must no longer be used. Other Flash memory sectors may still be used. The Error Flag (DQ5) Bit is reset after a Reset Flash instruction.

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

Table 8. Status Bit

Functional Block	FS0-FS7/CSBOOT0-CSBOOT3	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
Flash Memory	V _{IH}	Data Polling	Toggle Flag	Error Flag	X	Erase Time-out	X	X	X

Note: 1. X = Not guaranteed value, can be read either 1 or 0.
 2. DQ7-DQ0 represent the Data Bus Bits, D7-D0.
 3. FS0-FS7 and CSBOOT0-CSBOOT3 are active High.



Programming Flash Memory

Flash memory must be erased prior to being programmed. A byte of Flash memory is erased to all 1s (FFh), and is programmed by setting selected bits to '0.' The MCU may erase Flash memory all at once or by-sector, but not byte-by-byte. However, the MCU may program Flash memory byte-by-byte.

The primary and secondary Flash memories require the MCU to send an instruction to program a byte or to erase sectors (see Table 7).

Once the MCU issues a Flash memory Program or Erase instruction, it must check for 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/Busy (PC3).

Data Polling. Polling on the Data Polling Flag (DQ7) Bit is a method of checking whether a Program or Erase cycle is in progress or has completed. Figure 4 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 byte to be programmed in Flash memory to check status. The Data Polling Flag (DQ7) Bit of this location becomes the complement of b7 of the original data byte to be programmed. The MCU continues to poll this location, comparing the Data Polling Flag (DQ7) Bit and monitoring the Error Flag (DQ5) Bit. When the Data Polling Flag (DQ7) Bit matches b7 of the original data, and the Error Flag (DQ5) Bit remains '0,' the embedded algorithm is complete. If the Error Flag (DQ5) Bit is '1,' the MCU should test the Data Polling Flag (DQ7) Bit again since the Data Polling Flag (DQ7) Bit may have changed simultaneously with the Error Flag (DQ5) Bit (see Figure 4).

The Error Flag (DQ5) Bit is set if either an internal time-out occurred while the embedded algorithm attempted to program the byte or if the MCU attempted to program a '1' to a bit that was not erased (not erased is logic 0).

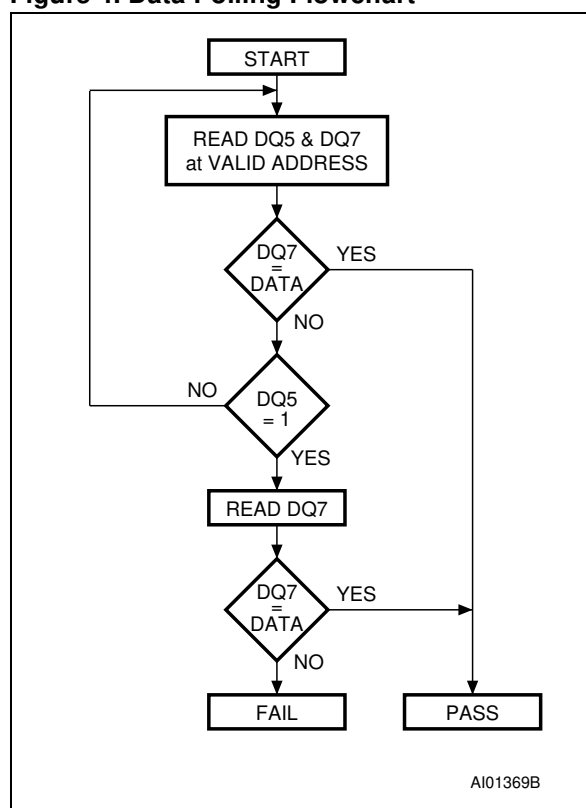
It is suggested (as with all Flash memories) to read the location again after the embedded program-

ming algorithm has completed, to compare the byte that was written to the Flash memory with the byte that was intended to be written.

When using the Data Polling method during an Erase cycle, Figure 4 still applies. However, the Data Polling Flag (DQ7) Bit is '0' until the Erase cycle is complete. A '1' on the Error Flag (DQ5) Bit indicates a time-out condition on the Erase cycle; a '0' indicates no error. The MCU can read any location within the sector being erased to get the Data Polling Flag (DQ7) Bit and the Error Flag (DQ5) Bit.

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

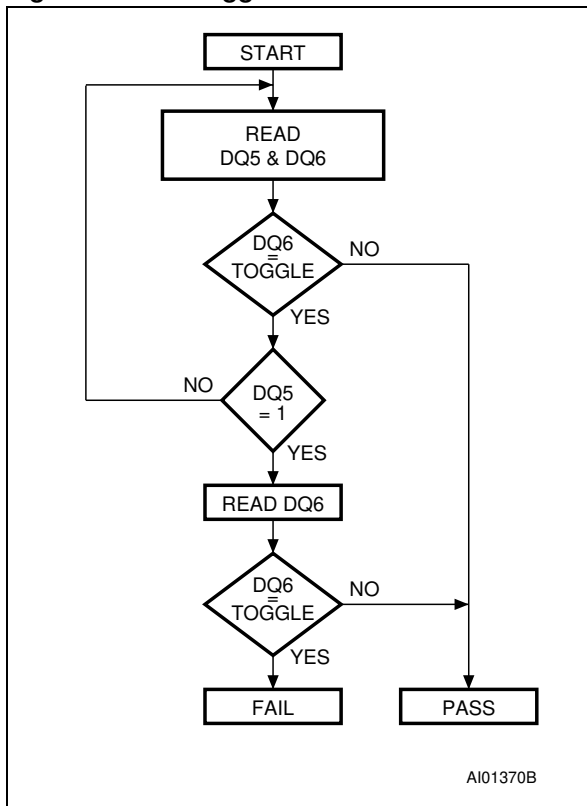
Figure 4. Data Polling Flowchart



Data Toggle. Checking the Toggle Flag (DQ6) Bit is a method of determining whether a Program or Erase cycle is in progress or has completed. Figure 5 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 of the byte to be programmed in Flash memory to check status. The Toggle Flag (DQ6) Bit of this location 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 (DQ6) Bit and monitoring the Error Flag (DQ5) Bit. When the Toggle Flag (DQ6) Bit stops toggling (two consecutive READs yield the same value), and the Error Flag (DQ5) Bit remains '0,' the embedded algorithm is complete. If the Error Flag (DQ5) Bit is '1,' the MCU should test the Toggle Flag (DQ6) Bit again, since the Toggle Flag (DQ6) Bit may have changed simultaneously with the Error Flag (DQ5) Bit (see Figure 5).

Figure 5. Data Toggle Flowchart



The Error Flag (DQ5) Bit is set if either an internal time-out occurred while the embedded algorithm attempted to program the byte, 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 byte that was written to Flash memory with the byte that was intended to be written.

When using the Data Toggle method after an Erase cycle, Figure 5 still applies. the Toggle Flag (DQ6) Bit toggles until the Erase cycle is complete. A 1 on the Error Flag (DQ5) Bit indicates a time-out condition on the Erase cycle; a 0 indicates no error. The MCU can read any location within the sector being erased to get the Toggle Flag (DQ6) Bit and the Error Flag (DQ5) Bit.

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

Unlock Bypass. The Unlock Bypass instructions allow the system to program bytes 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 code, 20h (as shown in Table 7).

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 code, A0h. The second cycle contains the program address and data. Additional data is programmed in the same manner. These instructions dispense with the initial two Unlock cycles required in the standard Program instruction, resulting in faster total Flash memory programming.

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

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

Erasing Flash Memory

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 7. If any byte of the Bulk Erase instruction is wrong, the Bulk Erase instruction aborts and the device is reset to the Read Flash memory status.

During a Bulk Erase, the memory status may be checked by reading the Error Flag (DQ5) Bit, the Toggle Flag (DQ6) Bit, and the Data Polling Flag (DQ7) Bit, as detailed in the section entitled “Programming Flash Memory”, on page 19. The Error Flag (DQ5) Bit 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.

Flash Sector Erase. The Sector Erase instruction uses six WRITE operations, as described in Table 7. Additional Flash Sector Erase codes and Flash memory sector addresses can be written subsequently to erase other Flash memory sectors in parallel, without further coded cycles, if the additional bytes are transmitted in a shorter time than the time-out period of about 100 μ s. The input of a new Sector Erase code restarts the time-out period.

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

During a Sector Erase, the memory status may be checked by reading the Error Flag (DQ5) Bit, the Toggle Flag (DQ6) Bit, and the Data Polling Flag

(DQ7) Bit, as detailed in the section entitled “Programming Flash Memory”, on page 19.

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

Suspend Sector Erase. When a Sector Erase cycle is in progress, the Suspend Sector Erase instruction can be used to suspend the cycle by writing 0B0h to any address when an appropriate Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) is High. (See Table 7). This allows reading of data from another Flash memory sector after the Erase cycle has been suspended. Suspend Sector Erase is accepted only during an Erase cycle and defaults to READ Mode. A Suspend Sector Erase instruction executed during an Erase time-out period, in addition to suspending the Erase cycle, terminates the time out period.

The Toggle Flag (DQ6) Bit 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 (DQ6) Bit stops toggling between 0.1 μ s and 15 μ s 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 sector that was *not* being erased is valid.
- The Flash memory *cannot* be programmed, and only responds to Resume Sector Erase and Reset Flash instructions (READ is an operation and is allowed).
- If a Reset Flash instruction is received, data in the Flash memory sector that was being erased is invalid.

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 address while an appropriate Sector Select (FS0-FS7 or CSBOOT0-CSBOOT3) is High. (See Table 7.)

Specific Features

Flash Memory Sector Protect. Each primary and secondary Flash memory sector 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 through the JTAG Port or a Device Programmer.

Sector protection can be selected for each sector using the PSDsoft Express Configuration 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 PSD/EE protection registers (in the CSIOP block). See Table 9 and Table 10.

Table 9. Sector Protection/Security Bit Definition – Flash Protection Register

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

Note: 1. Bit Definitions:

Sec<i>_Prot 1 = Primary Flash memory or secondary Flash memory Sector <i> is write protected.

Sec<i>_Prot 0 = Primary Flash memory or secondary Flash memory Sector <i> is not write protected.

Table 10. Sector Protection/Security Bit Definition – PSD/EE Protection Register

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

Note: 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.

1 = Security Bit in device has been set.

Reset Flash. The Reset Flash instruction consists of one WRITE cycle (see Table 7). It can also be optionally preceded by the standard two WRITE decoding cycles (writing AAh to 55h and 55h to AAh). It 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) to '1') during a Flash memory Program or Erase cycle.

The Reset Flash instruction puts the Flash memory back into normal READ Mode. If an Error condition has occurred (and the device has set the Error Flag Bit (DQ5) to '1') the Flash memory is put back into normal READ Mode within 25 μs of the Reset Flash instruction having been issued. The Reset Flash instruction is ignored when it is issued dur-

ing a Program or Bulk Erase cycle of the Flash memory. The Reset Flash instruction aborts any on-going Sector Erase cycle, and returns the Flash memory to the normal READ Mode within 25 μs.

Reset (RESET) Signal. A pulse on Reset (\overline{RESET}) 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 (RESET) pulse (except for Power On Reset, as described on page 61) be at least 25 μs so that the Flash memory is always ready for the MCU to fetch the bootstrap instructions after the Reset cycle is complete.

SRAM

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

The SRAM can be backed up using an external battery. The external battery should be connected to Voltage Stand-by (VSTBY, PC2). If you have an external battery connected to the PSD, the contents of the SRAM are retained in the event of a power loss. The contents of the SRAM are retained so long as the battery voltage remains at 2 V or greater. If the supply voltage falls below the battery voltage, an internal power switch-over to the battery occurs.

PC4 can be configured as an output that indicates when power is being drawn from the external battery. Battery-on Indicator (VBATON, PC4) is High with the supply voltage falls below the battery voltage and the battery on Voltage Stand-by (VSTBY, PC2) is supplying power to the internal SRAM.

SRAM Select (RS0), Voltage Stand-by (VSTBY, PC2) and Battery-on Indicator (VBATON, PC4) are all configured using PSDsoft Express Configuration.

Sector Select and SRAM Select

Sector Select (FS0-FS7, CSBOOT0-CSBOOT3) and SRAM Select (RS0) are all outputs of the DPLD. They are setup by writing equations for them in PSDlabel. The following rules apply to the equations for these signals:

1. Primary Flash memory and secondary Flash memory Sector Select signals must *not* be larger than the physical sector size.
2. Any primary Flash memory sector must *not* be mapped in the same memory space as another Flash memory sector.
3. A secondary Flash memory sector must *not* be mapped in the same memory space as another secondary Flash memory sector.
4. SRAM, I/O, and Peripheral I/O spaces must *not* overlap.
5. A secondary Flash memory sector *may* overlap a primary Flash memory sector. In case of

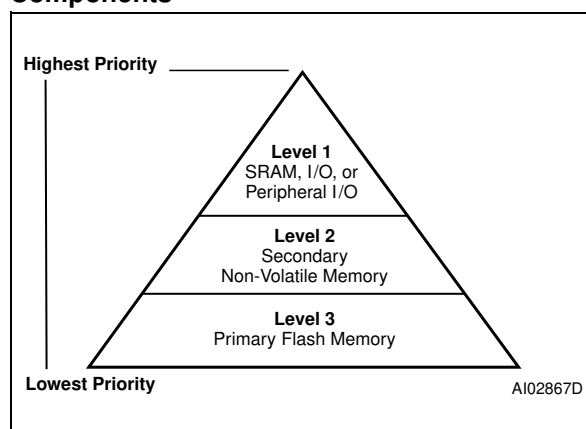
overlap, priority is given to the secondary Flash memory sector.

6. 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.

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 6 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 one has the highest priority and level 3 has the lowest.

Figure 6. Priority Level of Memory and I/O Components



Memory Select Configuration for MCUs with Separate Program and Data Spaces. The 8031 and compatible family of MCUs, which includes the 80C51, 80C151, 80C251, and 80C51XA, have separate address spaces for Program memory (selected using Program Select Enable (PSEN, CNTL2)) and Data memory (selected using Read Strobe (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 Express 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 primary and secondary Flash memories. This is easily done with the VM register by using PSDsoft Express Configuration to configure it for Boot-up and having the MCU change it when desired.

Table 11 describes the VM Register.

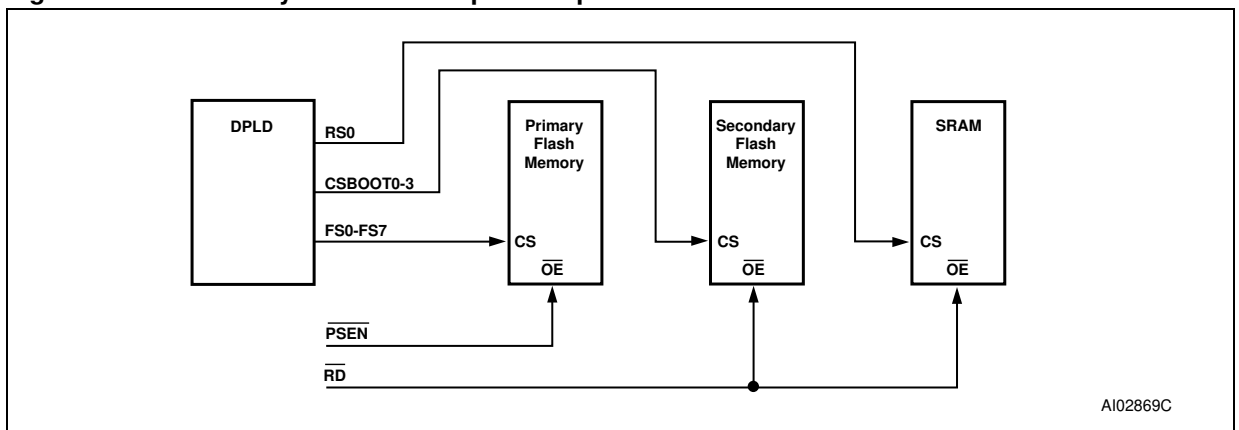
Table 11. VM Register

Bit 7 PIO_EN	Bit 6	Bit 5	Bit 4 Primary FL_Data	Bit 3 Secondary EE_Data	Bit 2 Primary FL_Code	Bit 1 Secondary EE_Code	Bit 0 SRAM_Code
0 = disable PIO mode	not used	not used	0 = \overline{RD} can't access Flash memory	0 = \overline{RD} can't access Secondary Flash memory	0 = \overline{PSEN} can't access Flash memory	0 = \overline{PSEN} can't access Secondary Flash memory	0 = \overline{PSEN} can't access SRAM
1 = enable PIO mode	not used	not used	1 = \overline{RD} access Flash memory	1 = \overline{RD} access Secondary Flash memory	1 = \overline{PSEN} access Flash memory	1 = \overline{PSEN} access Secondary Flash memory	1 = \overline{PSEN} access SRAM

Configuration Modes for MCUs with Separate Program and Data Spaces. Separate Space Modes. Program space is separated from Data space. For example, Program Select Enable (PSEN, CNTL2) is used to access the program

code from the primary Flash memory, while Read Strobe (\overline{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 7).

Figure 7. 8031 Memory Modules – Separate Space



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 (PSEN, CNTL2)

or Read Strobe (\overline{RD} , CNTL1). For example, to configure the primary Flash memory in Combined space, Bits b2 and b4 of the VM register are set to 1 (see Figure 8).

Figure 8. 8031 Memory Modules – Combined Space

