1 PRODUCT OVERVIEW

SAM87RI PRODUCT FAMILY

Samsung's SAM87RI family of 8-bit single-chip CMOS microcontrollers offers a fast and efficient CPU, a wide range of integrated peripherals, and various mask-programmable ROM sizes.

A dual address/data bus architecture and a large number of bit- or nibble-configurable I/O ports provide a flexible programming environment for applications with varied memory and I/O requirements. Timer/counters with selectable operating modes are included to support real-time operations. Many SAM87RI microcontrollers have an external interface that provides access to external memory and other peripheral devices.

S3C9644/C9648/P9648 Microcontroller

The S3C9644/C9648/P9648 single-chip 8-bit microcontroller is fabricated using an advanced CMOS process. It is built around the powerful SAM87RI CPU core.

Stop and Idle power-down modes were implemented to reduce power consumption. To increase on-chip register space, the size of the internal register file was logically expanded. The S3C9644 has 4K-bytes of program memory on-chip and S3C9648 has 8K-bytes.

Using the SAM87RI design approach, the following peripherals were integrated with the SAM87RI core:

— Five configurable I/O ports (32 pins)

— 20 bit-programmable pins for external interrupts

— 8-bit timer/counter with three operating modes

— Low speed USB function

The S3C9644/C9648/P9648 is a versatile microcontroller that can be used in a wide range of low speed USB support general purpose applications. It is especially suitable for use as a keyboard controller and is available in a 42-pin SDIP and a 44-pin QFP package.

OTP

The S3C9644/C9648 microcontroller is also available in OTP (One Time Programmable) version, S3P9648. S3P9648 microcontroller has an on-chip 8K-byte one-time-programmable EPROM instead of masked ROM. The S3P9648 is comparable to S3C9644/C9648, both in function and in pin configuration.

FEATURES

CPU

- SAM87RI CPU core

Memory

- 4/8K-byte internal program memory (ROM)
- 208-byte RAM

Instruction Set

- 41 instructions
- IDLE and STOP instructions added for power-down modes

Instruction Execution Time

- 1.0 µs at 6 MHz f_{OSC}

Interrupts

- 25 interrupt sources with one vector, each source has its pending bit
- One level, one vector interrupt structure

Oscillation Circuit

- 6 MHz crystal/ceramic oscillator
- External clock source (6 MHz)

General I/O

- Bit programmable five I/O ports (34 pins total)
  — (D+/PS2, D-/PS2 Included)

Timer/Counter

- One 8-bit basic timer for watchdog function and programmable oscillation stabilization interval generation function
- One 8-bit timer/counter with Compare/Overflow

USB Serial Bus

- Compatible to USB low speed (1.5 Mbps) device 1.0 specification.
- 1 Control endpoint and 2 Data endpoint
- Serial bus interface engine (SIE)
  — Packet decoding/generation
  — CRC generation and checking
  — NRZI encoding/decoding and bit-stuffing
- 8 bytes each receive/transmit USB buffer

Operating Temperature Range

- – 40 _C to + 85 _C

Operating Voltage Range

- 4.0 V to 5.25 V

Package Types

- 42-pin SDIP
- 44-pin QFP

BLOCK DIAGRAM

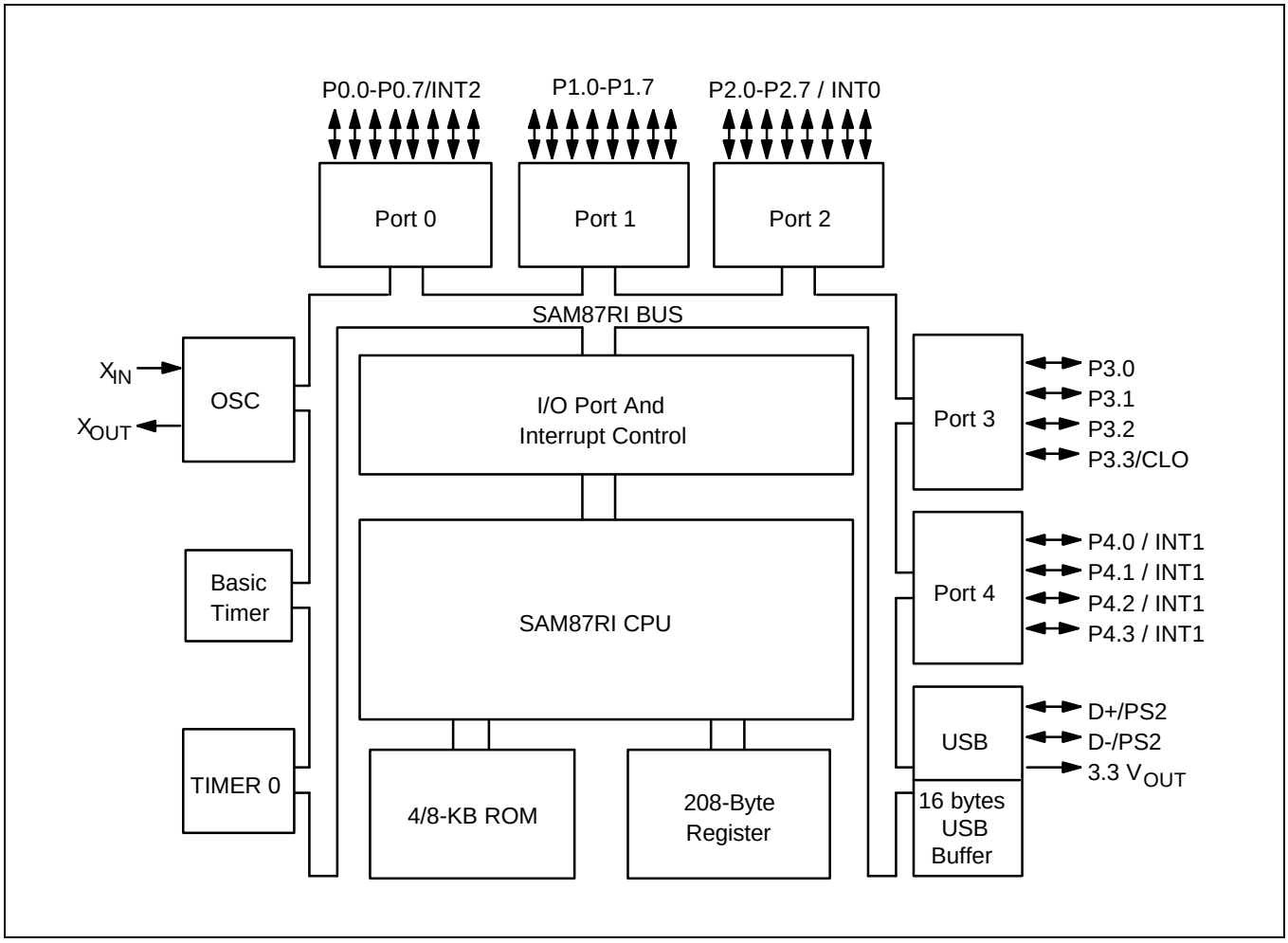


Figure 1-1. Block Diagram

PIN ASSIGNMENTS

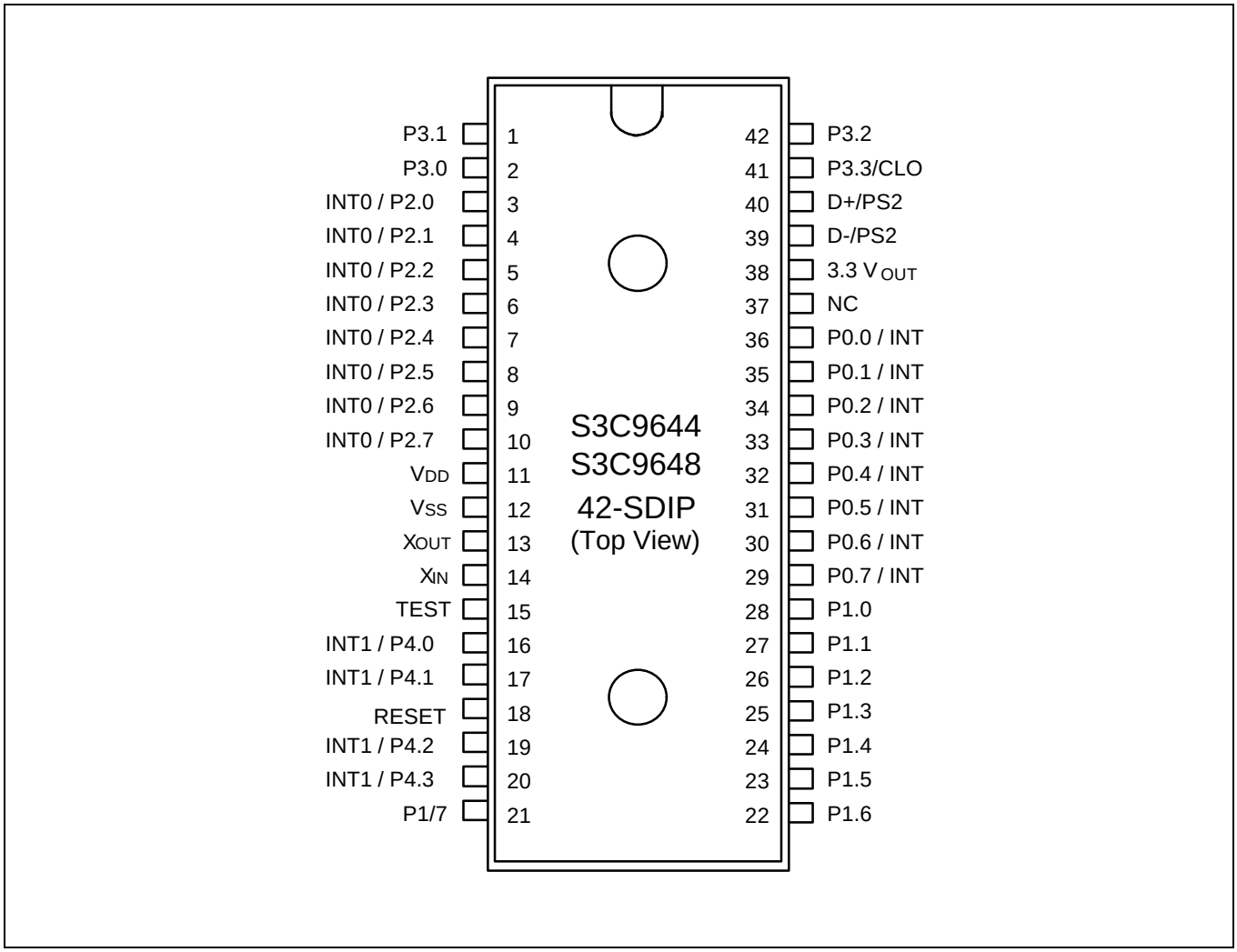


Figure 1-2. Pin Assignment Diagram (42-Pin SDIP Package)

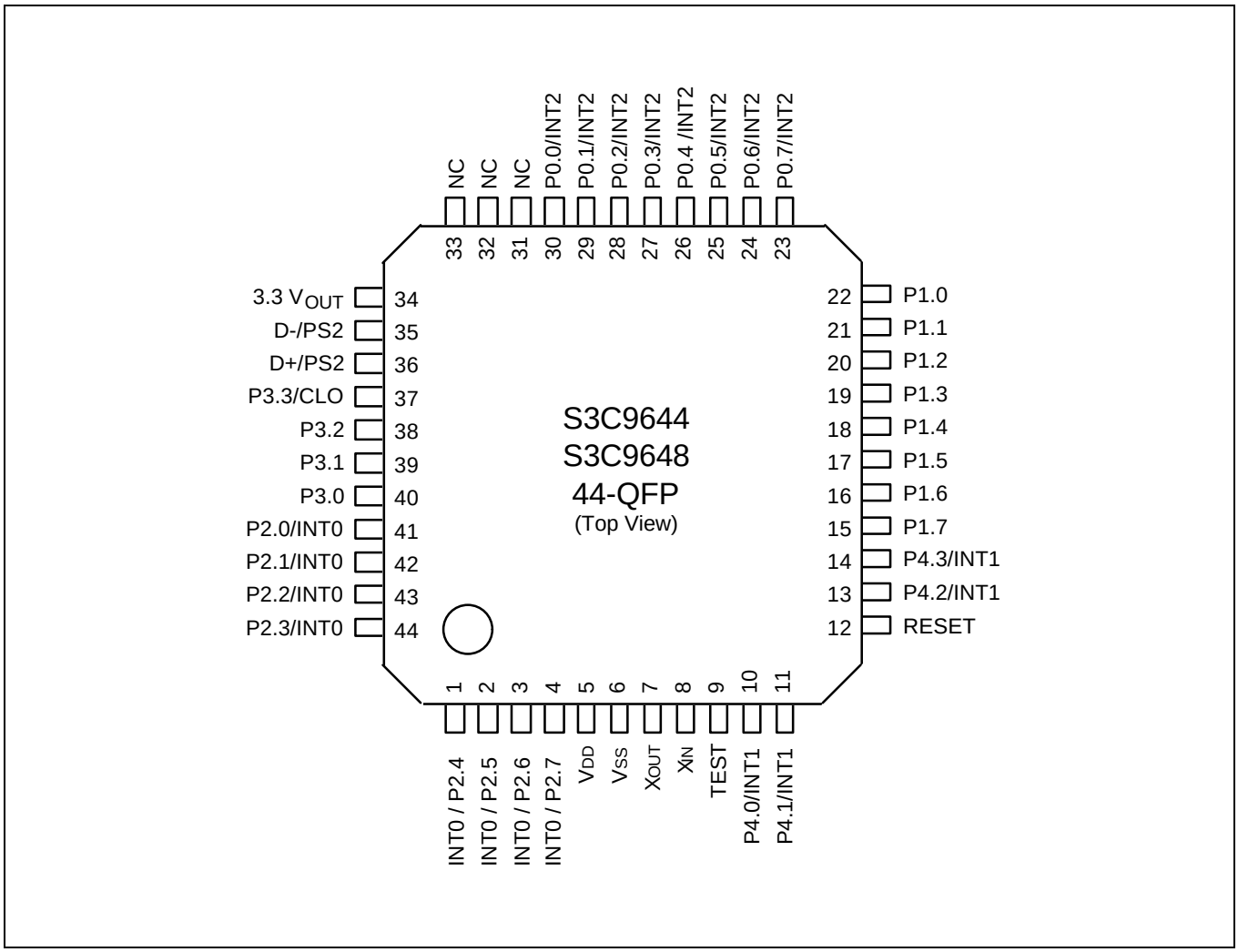


Figure 1-3. Pin Assignment Diagram (44-Pin QFP Package)

PIN DESCRIPTIONS

Table 1-1. S3C9644/C9648/P6408 Pin Descriptions

Pin Names	Pin Type	Pin Description	Circuit Number	Pin Numbers	Share Pins
P0.0-P0.7	I/O	Bit-programmable I/O port for Schmitt trigger input or open-drain output. Port0 can be individually configured as external interrupt inputs. Pull-up resistors are assignable by software.	B	36-29 (30-23)	INT2
P1.0-P1.7	I/O	Bit-programmable I/O port for Schmitt trigger input or open-drain output. Pull-up resistors are assignable by software.	B	28-21 (22-15)	–
P2.0-P2.7	I/O	Bit-programmable I/O port for Schmitt trigger input or open-drain output. Port2 can be individually configured as external interrupt inputs. Pull-up resistors are assignable by software.	B	3-10 (41-44, 1-4)	INT0
P3.0-P3.3	I/O	Bit-programmable I/O port for Schmitt trigger input, open-drain or push-pull output. P3.3 can be used to system clock output(CLO) pin.	C	2, 1, 42, 41 (40-37)	P3.3/CLO
P4.0-P4.3	I/O	Bit-programmable I/O port for Schmitt trigger input or open-drain output or push-pull output. Port4 can be individually configured as external interrupt inputs. In output mode, pull-up resistors are assignable by software. But in input mode, pull-up resistors are fixed.	D	16, 17, 19, 20 (10, 11, 13, 14)	INT1
D+/PS2 D-/PS2	I/O	Programmable port for USB interface or PS2 interface.	–	40-39 (36-35)	–
3.3 V_{OUT}	–	3.3 V output from internal voltage regulator	–	38 (34)	–
X_{IN}, X_{OUT}	–	System clock input and output pin (crystal/ceramic oscillator, or external clock source)	–	14, 13 (8, 7)	–
INT0 INT1 INT2	I	External interrupt for bit-programmable port0, port2 and port4 pins when set to input mode.	–	3-10, 16,17, 19, 20, 29-36 (30-23, 41-44, 1-4, 10, 11, 13, 14)	PORT2/ PORT4/ PORT0
RESET	I	RESET signal input pin. Input with internal pull-up resistor.	A	18 (12)	–
TEST	I	Test signal input pin (for factory use only; connected to V_{SS})	–	15 (9)	–
V_{DD}	–	Power input pin	–	11 (5)	–
V_{SS}	–	Ground input pin	–	12, (6)	–
NC	–	No connection	–	37 (31,32, 33)	–

NOTE: Pin numbers shown in parenthesis '( )' are for the 44-QFP package; others are for the 42-SDIP package.

PIN CIRCUITS

Table 1-2. Pin Circuit Assignments for the S3C9644/C9648/P6408

Circuit Number	Circuit Type	S3C9644/C9648/P6408 Assignments
A	I	RESET signal input
B	I/O	Ports 0, 1, and 2
C	I/O	Port 3
D	I/O	Port 4

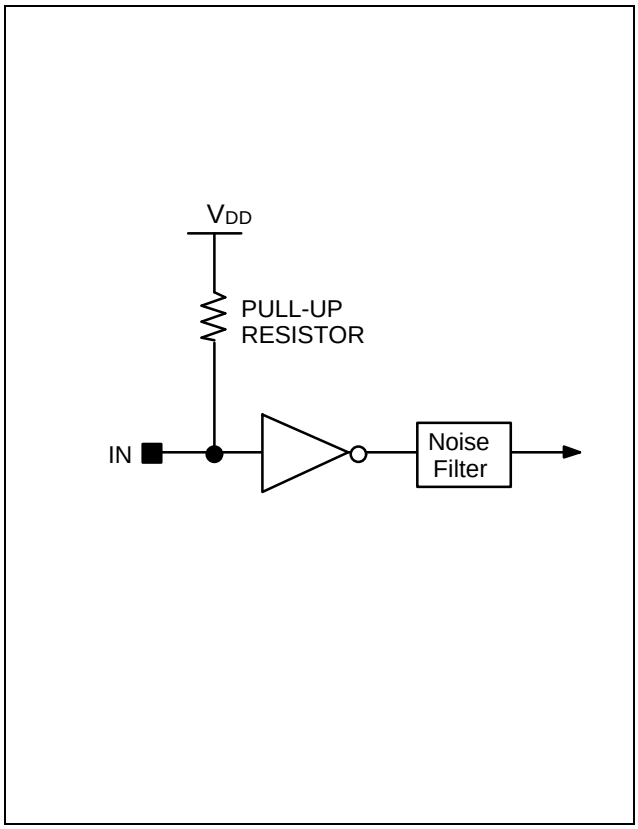


Figure 1-4. Pin Circuit Type A (RESET)

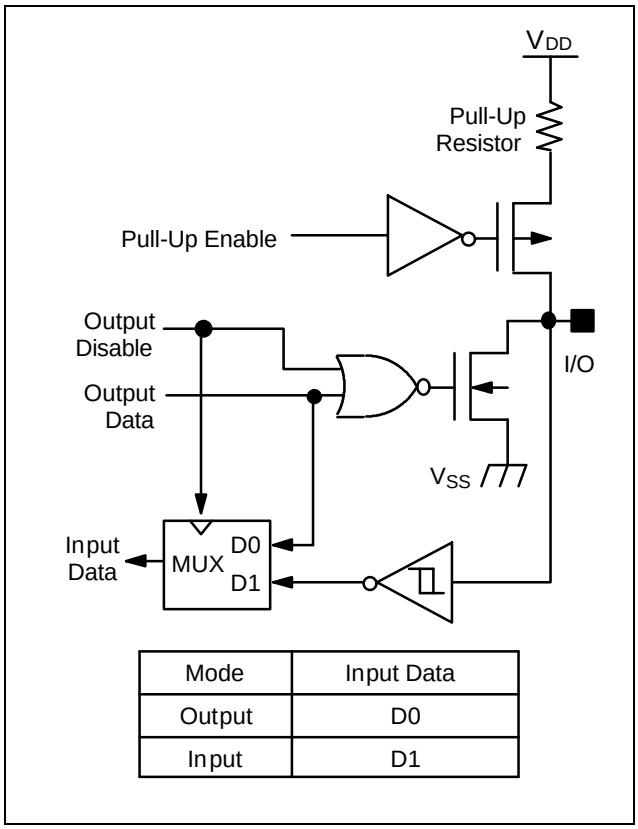
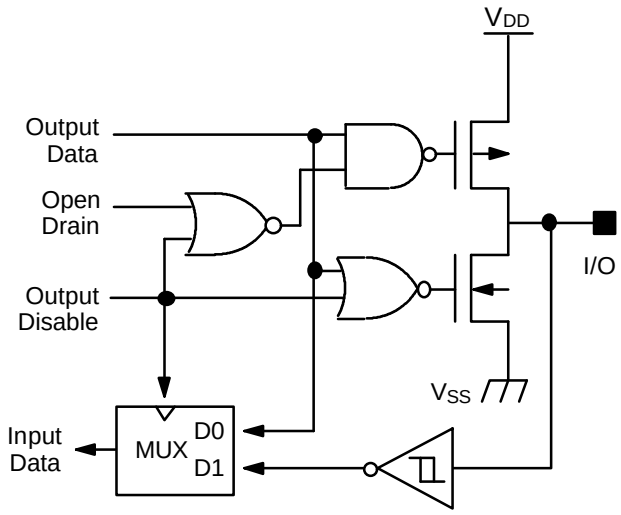
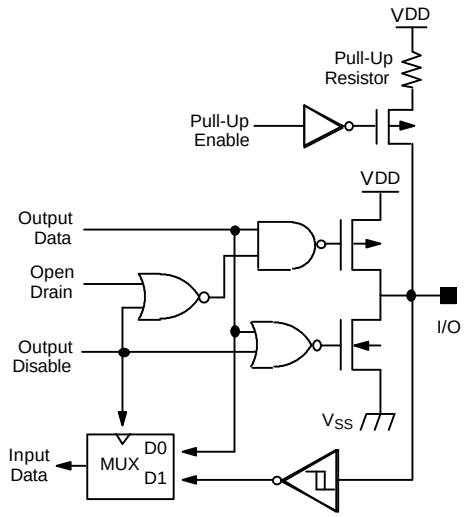


Figure 1-5. Pin Circuit Type B (Ports 0, 1 and 2)



Mode	Input Data
Output	D0
Input	D1

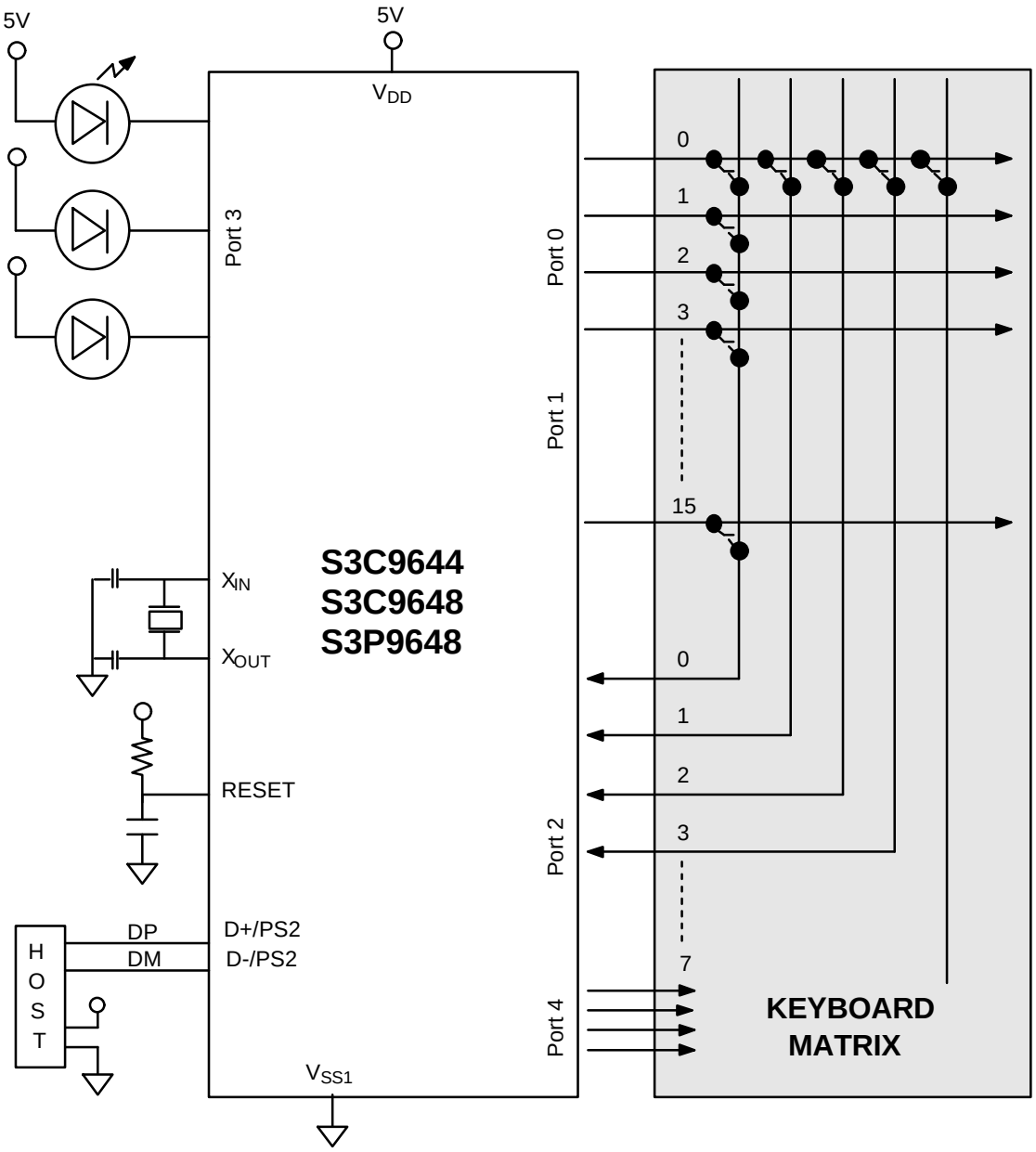
Figure 1-6. Pin Circuit Type C (Port 3)



Mode	Input Data
Output	D0
Input	D1

Figure 1-7. Pin Circuit Type D (Port 4)

APPLICATION CIRCUIT



NOTE: Port4 can use expend keyboard MATRIX.
D+/PS2, D-/PS2 can use PS2 keyboard interface (see PS2CONINT, page 4-25).
Port 4.2, 4.3 can use PS2 mouse interface.
Port 3 can use LED direct drive.

Figure 1-8. Keyboard Application Circuit Diagram

2 ADDRESS SPACES

OVERVIEW

The S3C9644/C9648/P9648 microcontroller has two kinds of address space:

— Program memory (ROM), internal

— Internal register file

A 13-bit address bus supports both program memory. A separate 8-bit register bus carries addresses and data between the CPU and the internal register file.

The S3C9644 has 4K-bytes of mask-programmable program memory on-chip and S3C9648 has 8K-bytes. There is one program memory configuration option:

— Internal ROM mode, in which only the 8K-byte internal program memory is used.

The S3C9644/C9648/P9648 microcontroller has 208 general-purpose registers in its internal register file. Twenty-seven bytes in the register file are mapped for system and peripheral control functions.

PROGRAM MEMORY (ROM)

Normal Operating Mode (Internal ROM)

The S3C9644 has 4K-bytes (locations 0H–0FFFH) of internal mask-programmable program memory. The S3C9648/P9648 has 8K-bytes (locations 0H–1FFFH) of internal mask-programmable program memory.

The first 2 bytes of the ROM (0000H–0001H) are an interrupt vector address.

The program reset address in the ROM is 0100H.

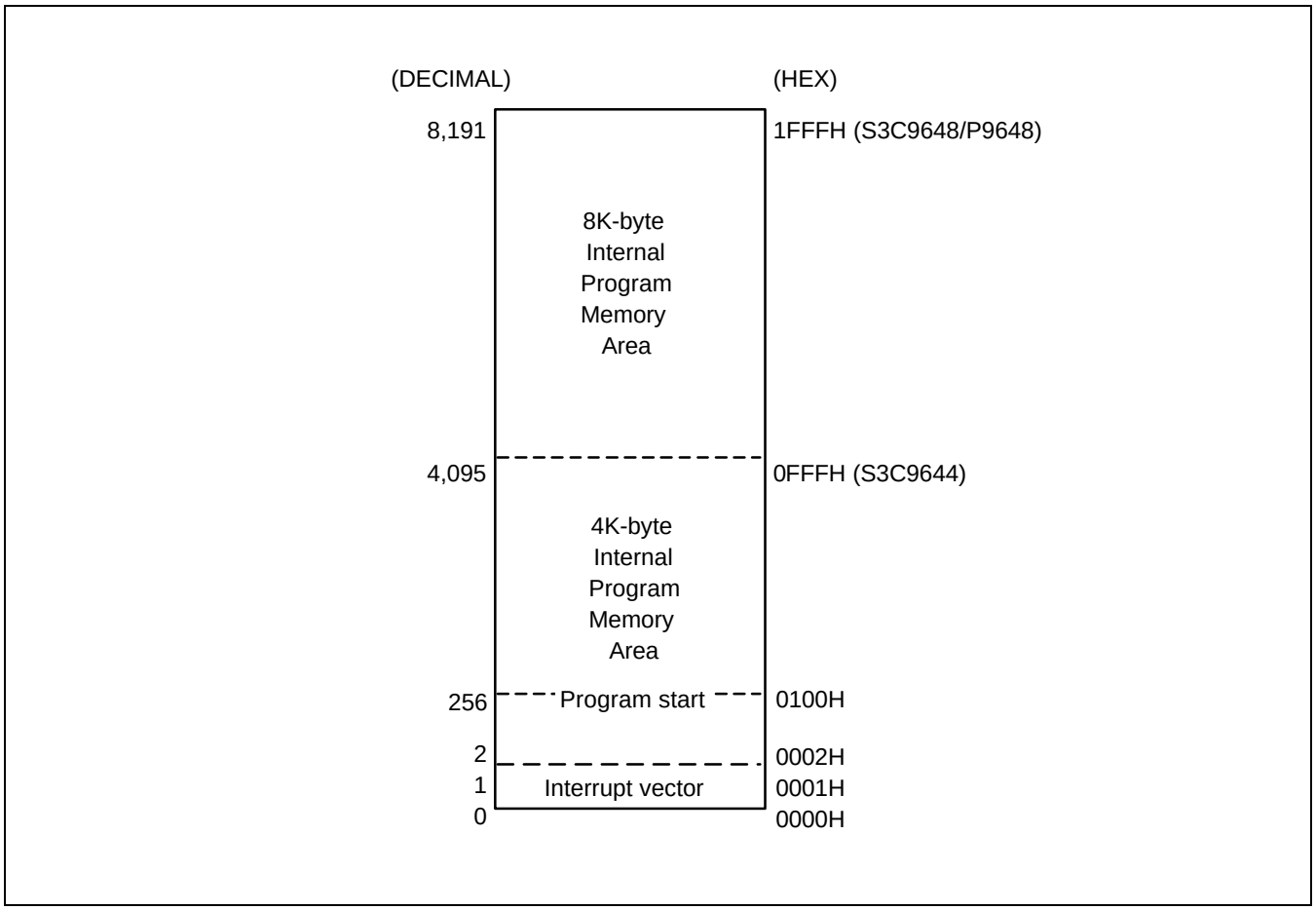


Figure 2-1. Program Memory Address Space

REGISTER ARCHITECTURE

The upper 64 bytes of the S3C9644/C9648/P9648's internal register file are addressed as working registers, system control registers and peripheral control registers. The lower 192 bytes of internal register file (00H–BFH) is called the *general purpose register space*. The total addressable register space is thereby 256 bytes. 233 registers in this space can be accessed.; 208 are available for general-purpose use.

For many SAM87RI microcontrollers, the addressable area of the internal register file is further expanded by the additional of one or more register pages at general purpose register space (00H–BFH). This register file expansion is not implemented in the S3C9644/C9648/P9648, however. Page addressing is controlled by the System Mode Register (SYM.1–SYM.0).

The specific register types and the area (in bytes) that they occupy in the internal register file are summarized in Table 2-1.

Table 2-1. Register Type Summary

Register Type	Number of Bytes
CPU and system control registers	11
Peripheral, I/O, and clock control and data registers	34
General-purpose registers (including the 16-bit common working register area)	208
Total Addressable Bytes	**253**

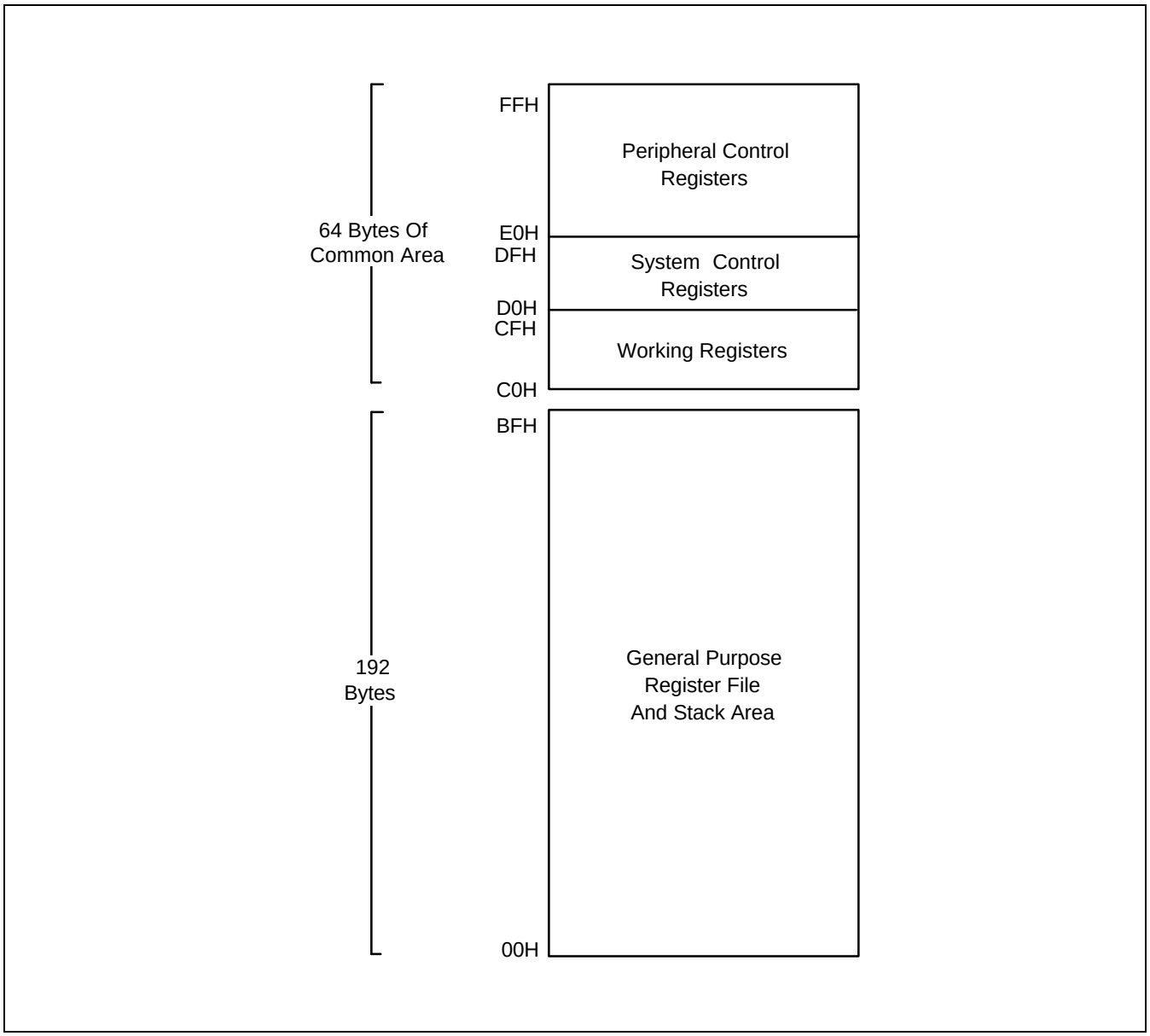


Figure 2-2. Internal Register File Organization

COMMON WORKING REGISTER AREA (C0H–CFH)

The SAM87RI register architecture provides an efficient method of working register addressing that takes full advantage of shorter instruction formats to reduce execution time.

This 16-byte address range is called *common area*. That is, locations in this area can be used as working registers by operations that address any location on any page in the register file. Typically, these working registers serve as temporary buffers for data operations between different pages. However, because the S3C9644/C9648/P9648 uses only page 0, you can use the common area for any internal data operation.

The Register (R) addressing mode can be used to access this area

Registers are addressed either as a single 8-bit register or as a paired 16-bit register. In 16-bit register pairs, the address of the first 8-bit register is always an even number and the address of the next register is an odd number. The most significant byte of the 16-bit data is always stored in the even-numbered register; the least significant byte is always stored in the next (+ 1) odd-numbered register.

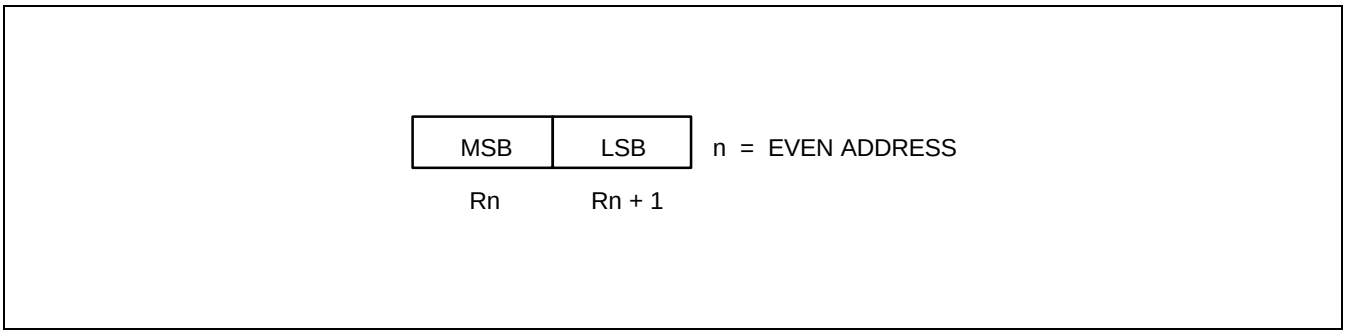


Figure 2-3. 16-Bit Register Pairs

☞ PROGRAMMING TIP — Addressing the Common Working Register Area

As the following examples show, you should access working registers in the common area, locations C0H–CFH, using working register addressing mode only.

Examples:

1. LD 0C2H,40H ; Invalid addressing mode!
   Use working register addressing instead:

   LD R2,40H ; R2 (C2H) ¨ the value in location 40H

2. ADD 0C3H,#45H ; Invalid addressing mode!
   Use working register addressing instead:

   ADD R3,#45H ; R3 (C3H) ¨ R3 + 45H

SYSTEM STACK

S3C9-series microcontrollers use the system stack for subroutine calls and returns and to store data. The PUSH and POP instructions are used to control system stack operations. The S3C9644/C9648/P9648 architecture supports stack operations in the internal register file.

Stack Operations

Return addresses for procedure calls and interrupts and data are stored on the stack. The contents of the PC are saved to stack by a CALL instruction and restored by the RET instruction. When an interrupt occurs, the contents of the PC and the FLAGS register are pushed to the stack. The IRET instruction then pops these values back to their original locations. The stack address is always decremented *before* a push operation and incremented *after* a pop operation. The stack pointer (SP) always points to the stack frame stored on the top of the stack, as shown in Figure 2-4.

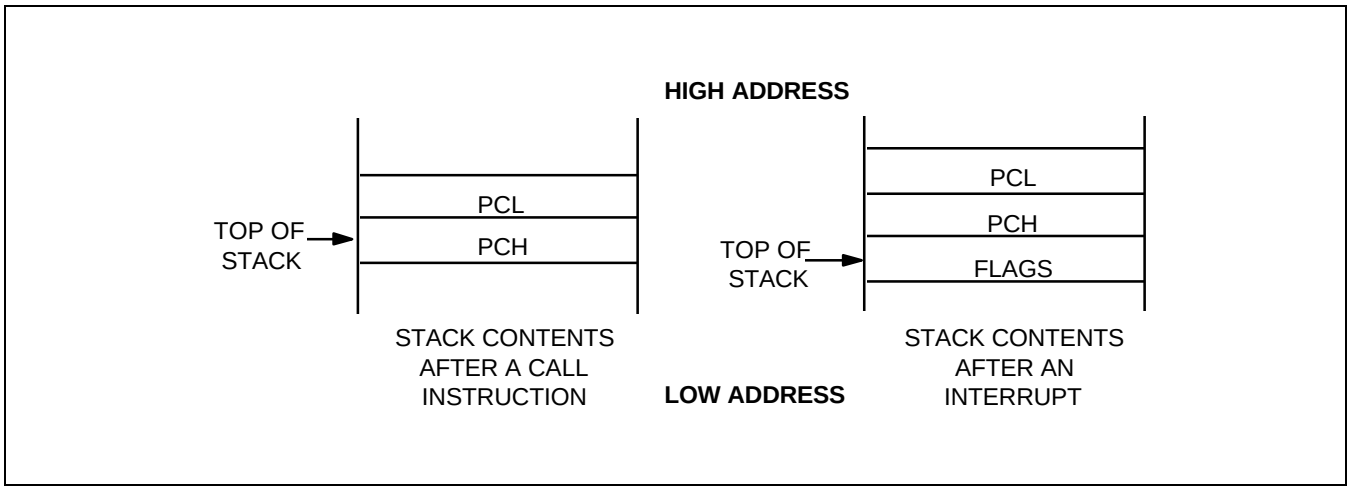


Figure 2-4. Stack Operations

Stack Pointer (SP)

Register location D9H contains the 8-bit stack pointer (SP) that is used for system stack operations. After a reset, the SP value is undetermined.

Because only internal memory space is implemented in the S3C9644/C9648/P9648, the SP must be initialized to an 8-bit value in the range 00H–BFH.

NOTE: In case a Stack Pointer is initialized to 00H, it is decreased to FFH when stack operation starts. This means that a Stack Pointer access invalid stack area.

☞ PROGRAMMING TIP — Standard Stack Operations Using PUSH and POP

The following example shows you how to perform stack operations in the internal register file using PUSH and POP instructions:

```
LD      SP,#0C0H        ;  SP ¨ C0H (Normally, the SP is set to 0C0H by the
                        ;  initialization routine)
•
•
•
PUSH    SYM             ;  Stack address 0BFH ¨ SYM
PUSH    CLKCON          ;  Stack address 0BEH ¨ CLKCON
PUSH    20H             ;  Stack address 0BDH ¨ 20H
PUSH    R3              ;  Stack address 0BCH ¨ R3
•
•
•
POP     R3              ;  R3 ¨ Stack address 0BCH
POP     20H             ;  20H ¨ Stack address 0BDH
POP     CLKCON          ;  CLKCON ¨ Stack address 0BEH
POP     SYM             ;  SYM ¨ Stack address 0BFH
```

3 ADDRESSING MODES

OVERVIEW

Instructions that are stored in program memory are fetched for execution using the program counter. Instructions indicate the operation to be performed and the data to be operated on. Addressing mode is the method used to determine the location of the data operand. The operands specified in SAM87RI instructions may be condition codes, immediate data, or a location in the register file, program memory, or data memory.

The SAM87RI instruction set supports six explicit addressing modes. Not all of these addressing modes are available for each instruction. The addressing modes and their symbols are as follows:

— Register (R)
— Indirect Register (IR)
— Indexed (X)
— Direct Address (DA)
— Relative Address (RA)
— Immediate (IM)

REGISTER ADDRESSING MODE (R)

In Register addressing mode, the operand is the content of a specified register (see Figure 3-1). Working register addressing differs from Register addressing because it uses an 16-byte working register space in the register file and an 4-bit register within that space (see Figure 3-2).

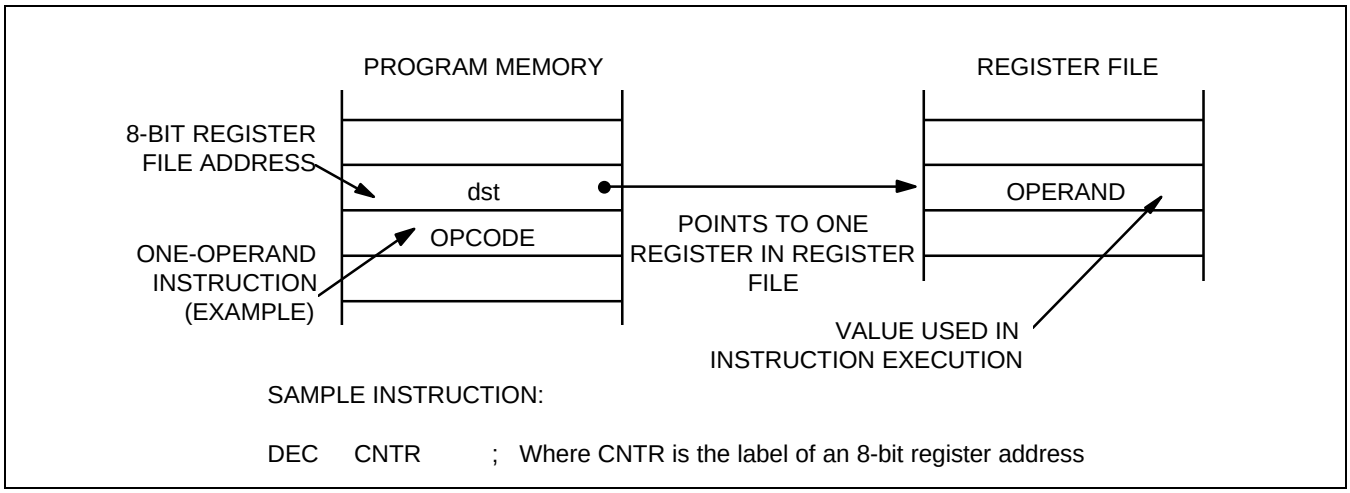


Figure 3-1. Register Addressing

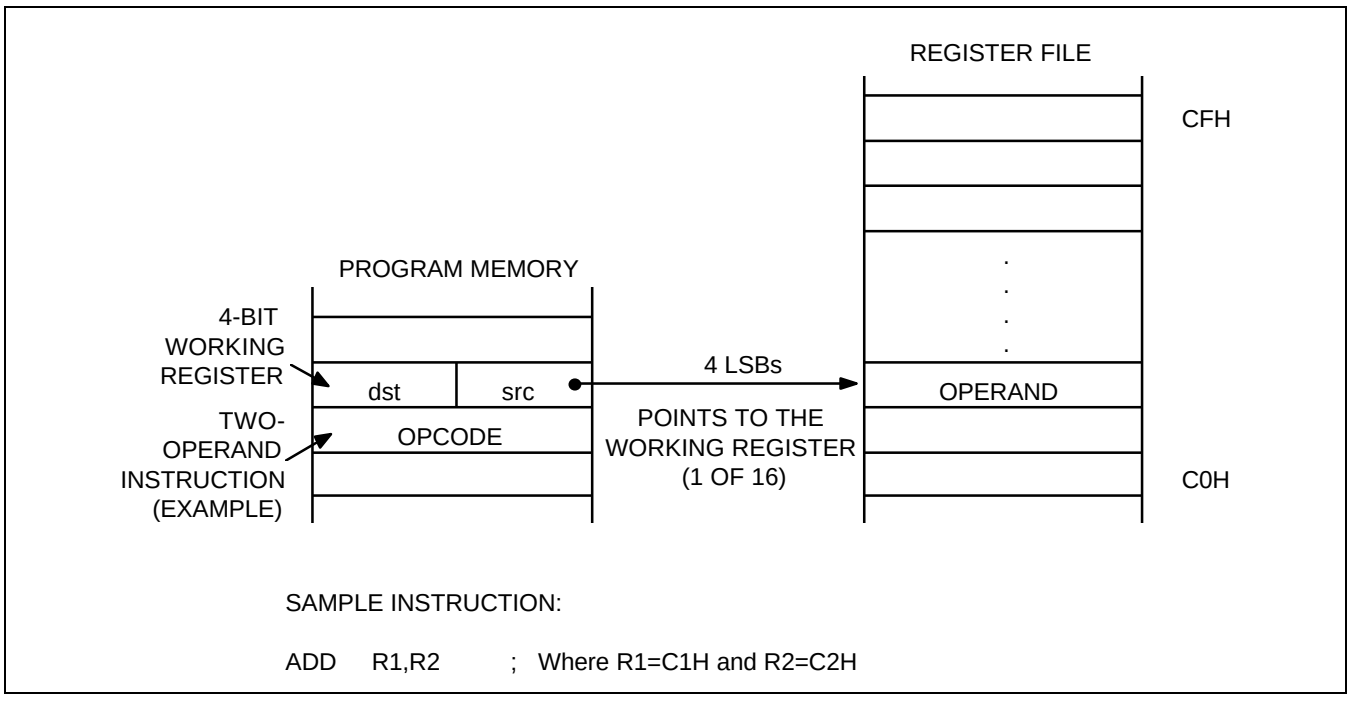


Figure 3-2. Working Register Addressing

INDIRECT REGISTER ADDRESSING MODE (IR)

In Indirect Register (IR) addressing mode, the content of the specified register or register pair is the address of the operand. Depending on the instruction used, the actual address may point to a register in the register file, to program memory (ROM), or to an external memory space (see Figures 3-3 through 3-6).

You can use any 8-bit register to indirectly address another register. Any 16-bit register pair can be used to indirectly address another memory location.

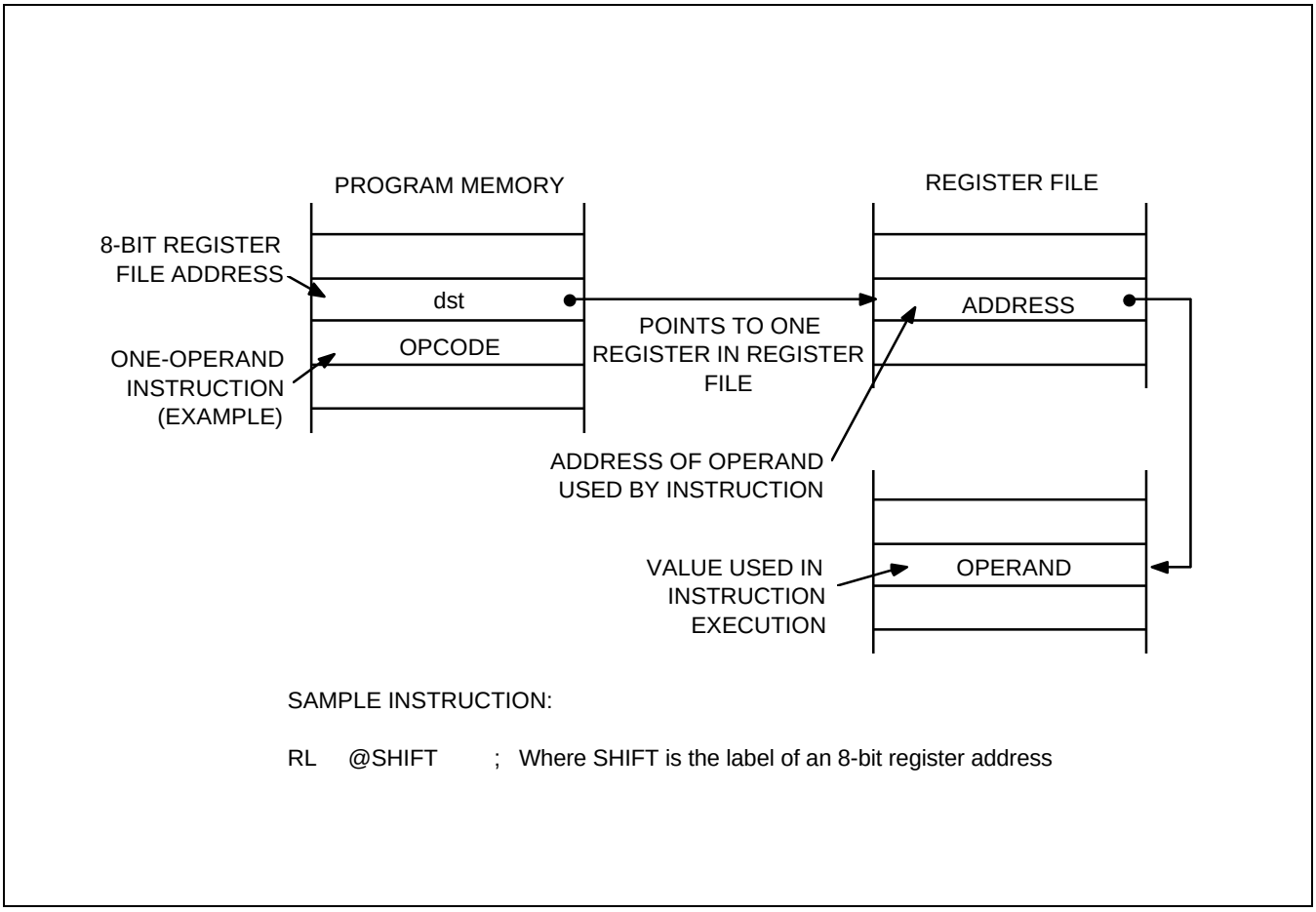


Figure 3-3. Indirect Register Addressing to Register File

INDIRECT REGISTER ADDRESSING MODE (Continued)

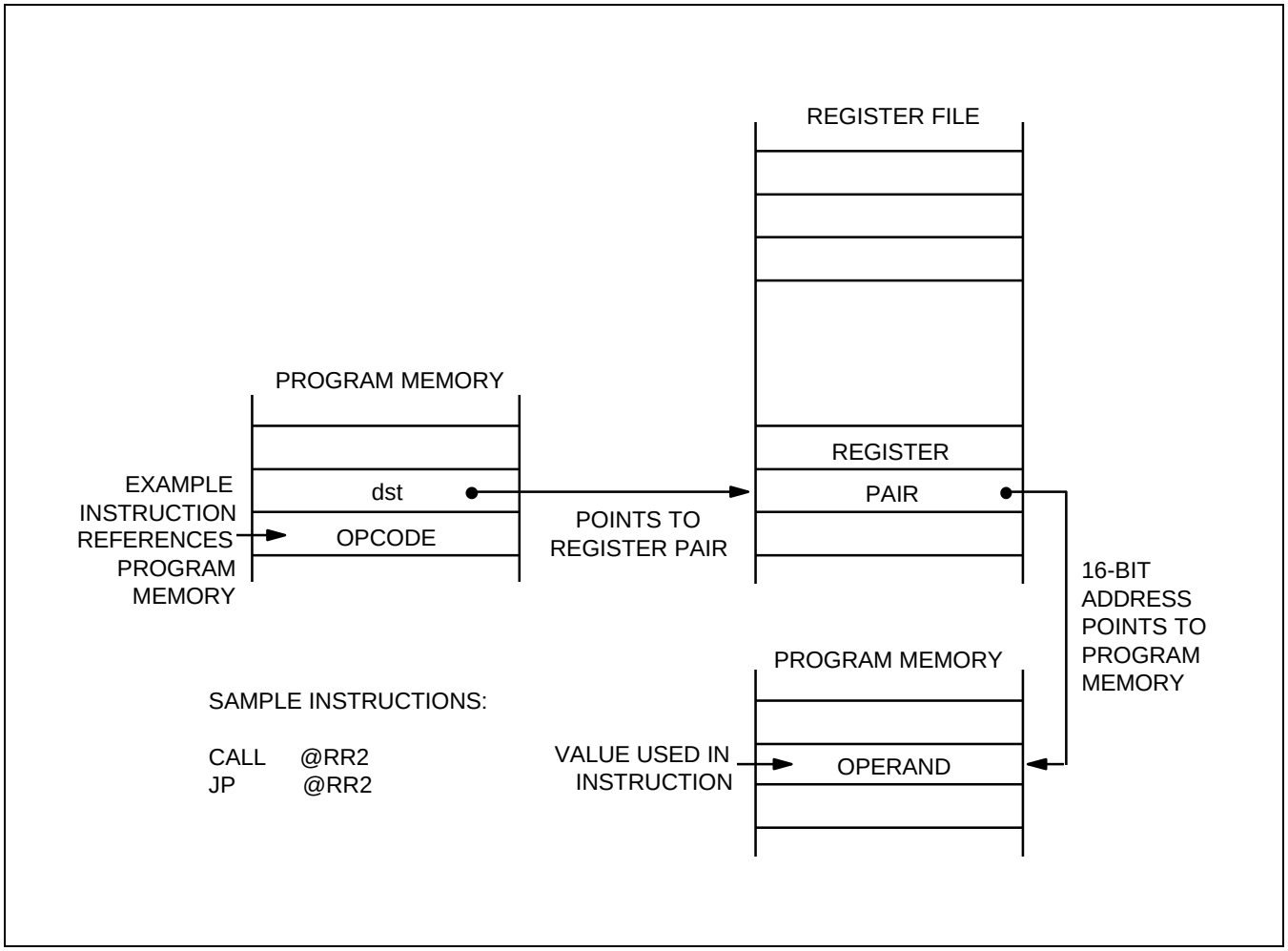


Figure 3-4. Indirect Register Addressing to Program Memory

INDIRECT REGISTER ADDRESSING MODE (Continued)

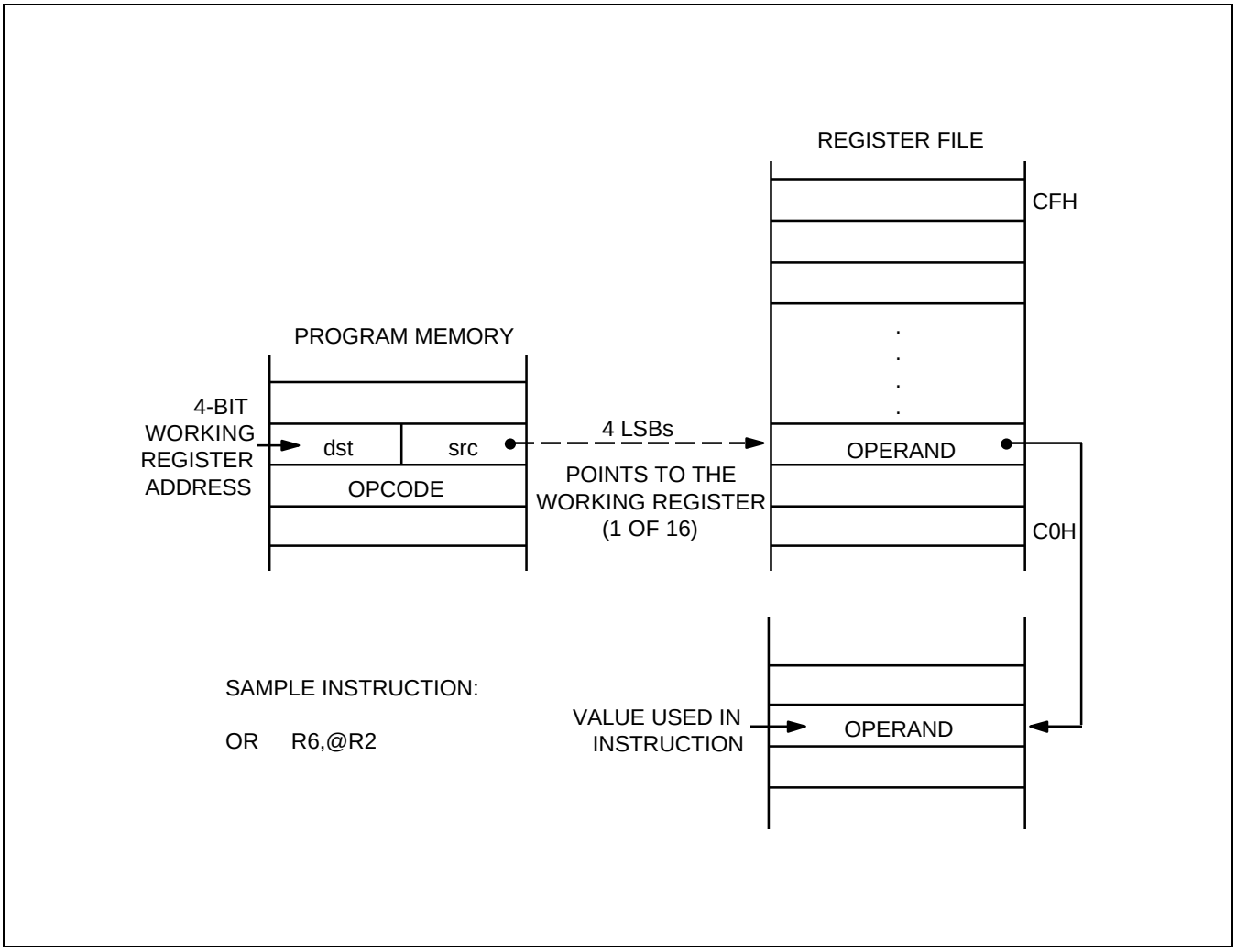


Figure 3-5. Indirect Working Register Addressing to Register File

INDIRECT REGISTER ADDRESSING MODE (Concluded)

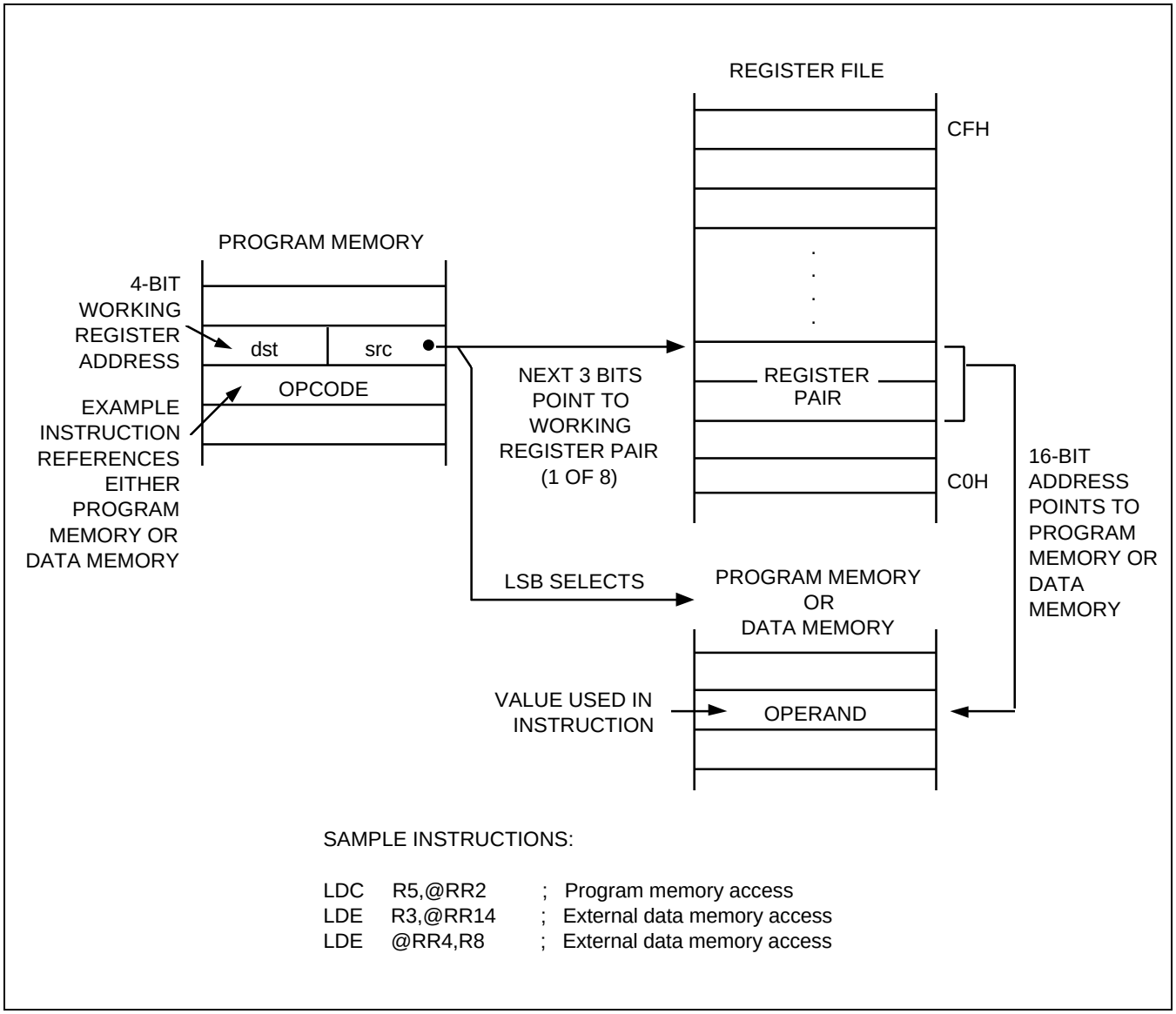


Figure 3-6. Indirect Working Register Addressing to Program or Data Memory

INDEXED ADDRESSING MODE (X)

Indexed (X) addressing mode adds an offset value to a base address during instruction execution in order to calculate the effective operand address (see Figure 3-7). You can use Indexed addressing mode to access locations in the internal register file or in external memory.

In short offset Indexed addressing mode, the 8-bit displacement is treated as a signed integer in the range –128 to +127. This applies to external memory accesses only (see Figure 3-8).

For register file addressing, an 8-bit base address provided by the instruction is added to an 8-bit offset contained in a working register. For external memory accesses, the base address is stored in the working register pair designated in the instruction. The 8-bit or 16-bit offset given in the instruction is then added to the base address (see Figure 3-9).

The only instruction that supports Indexed addressing mode for the internal register file is the Load instruction (LD). The LDC and LDE instructions support Indexed addressing mode for internal program memory, external program memory, and for external data memory, when implemented.

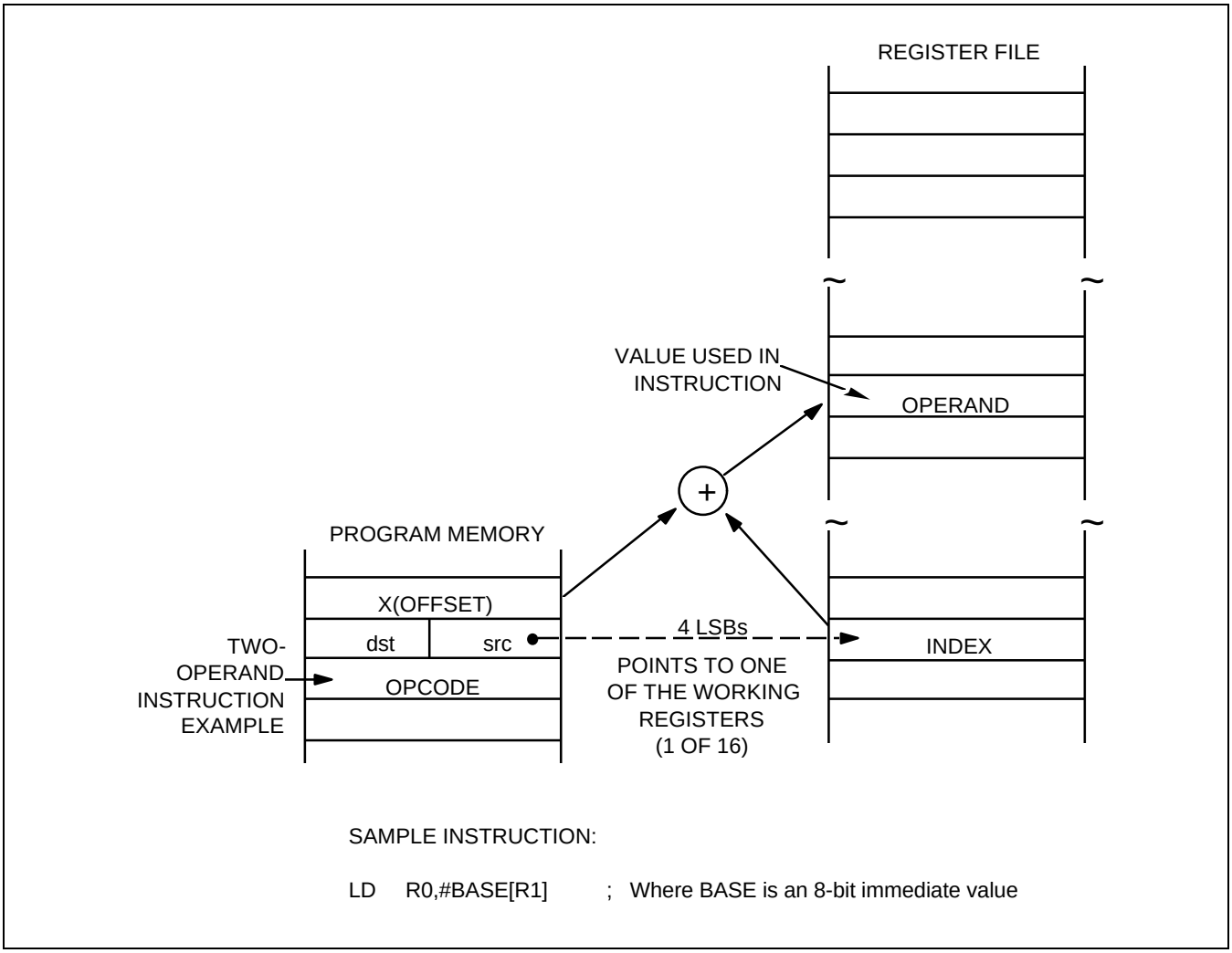
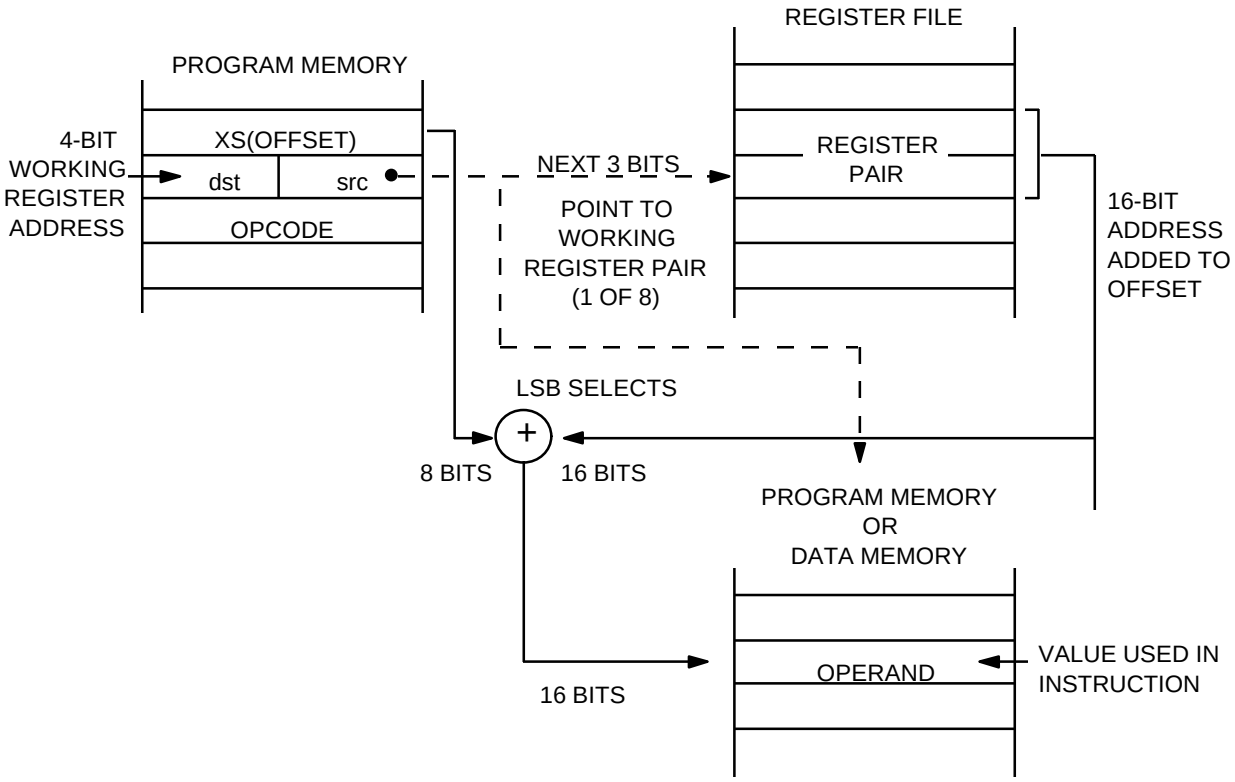


Figure 3-7. Indexed Addressing to Register File

INDEXED ADDRESSING MODE (Continued)



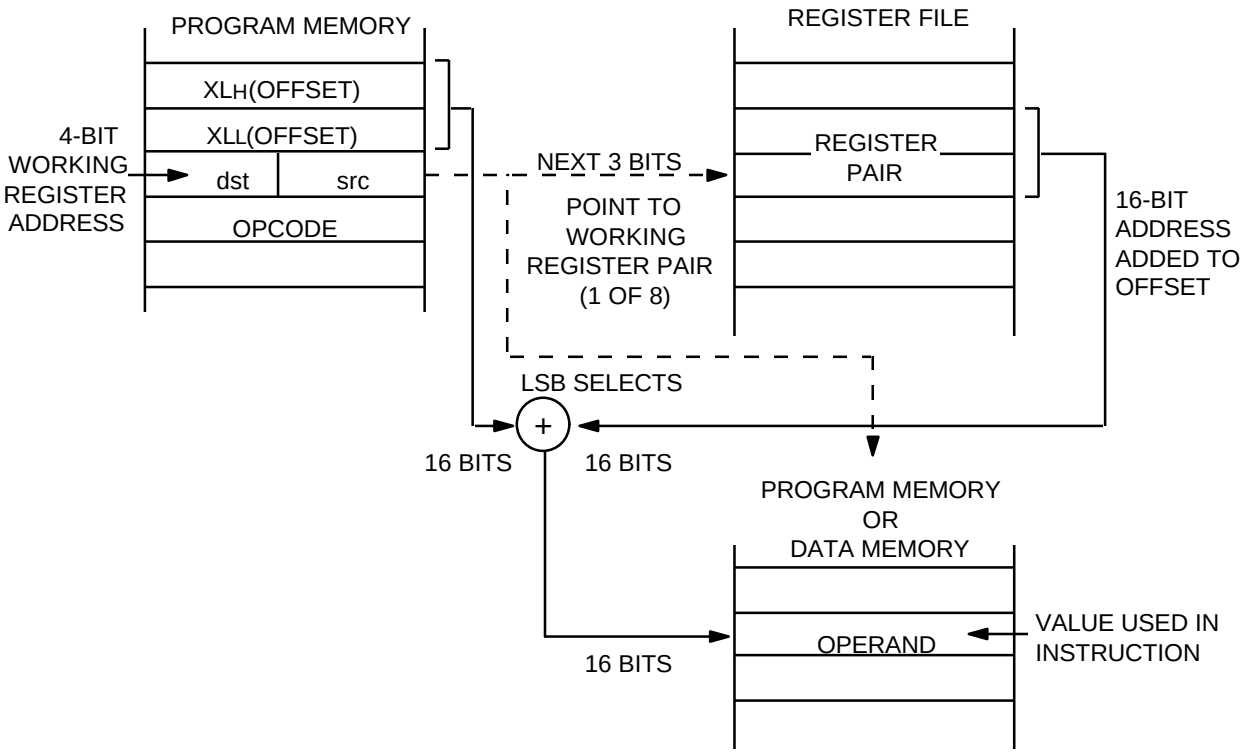
SAMPLE INSTRUCTIONS:

```
LDC     R4,#04H[RR2]    ;  The values in the program address (RR2 + #04H)
                           are loaded into register R4.

LDE     R4,#04H[RR2]    ;  Identical operation to LDC example, except that
                           external program memory is accessed.
```

Figure 3-8. Indexed Addressing to Program or Data Memory with Short Offset

INDEXED ADDRESSING MODE (Concluded)



SAMPLE INSTRUCTIONS:

```
LDC     R4,#1000H[RR2]      ;  The values in the program address (RR2 + #1000H)
                               are loaded into register R4.

LDE     R4,#1000H[RR2]      ;  Identical operation to LDC example, except that
                               external program memory is accessed.
```

Figure 3-9. Indexed Addressing to Program or Data Memory with Long Offset

DIRECT ADDRESS MODE (DA)

In Direct Address (DA) mode, the instruction provides the operand's 16-bit memory address. Jump (JP) and Call (CALL) instructions use this addressing mode to specify the 16-bit destination address that is loaded into the PC whenever a JP or CALL instruction is executed.

The LDC and LDE instructions can use Direct Address mode to specify the source or destination address for Load operations to program memory (LDC) or to external data memory (LDE), if implemented.

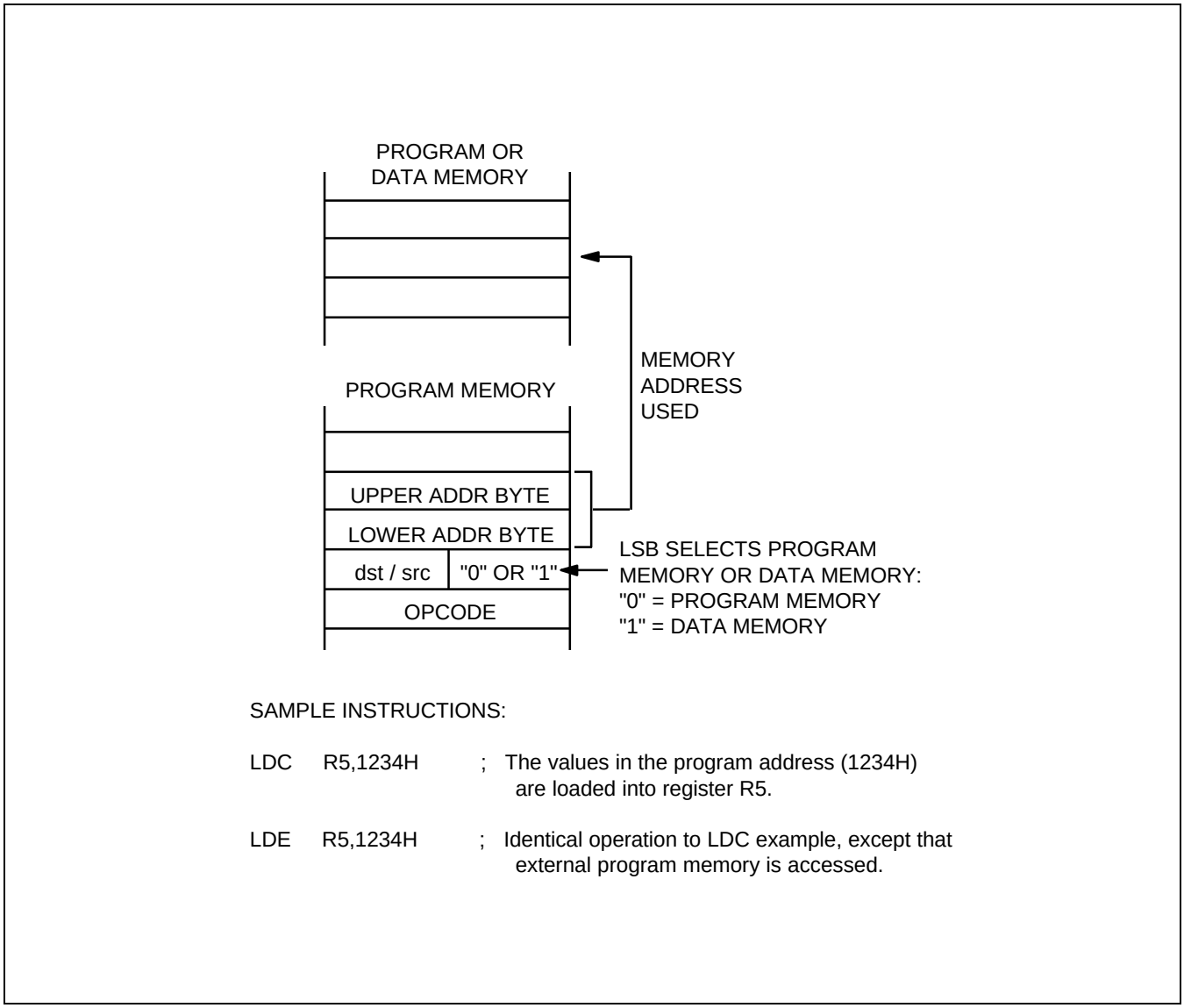


Figure 3-10. Direct Addressing for Load Instructions

DIRECT ADDRESS MODE (Continued)

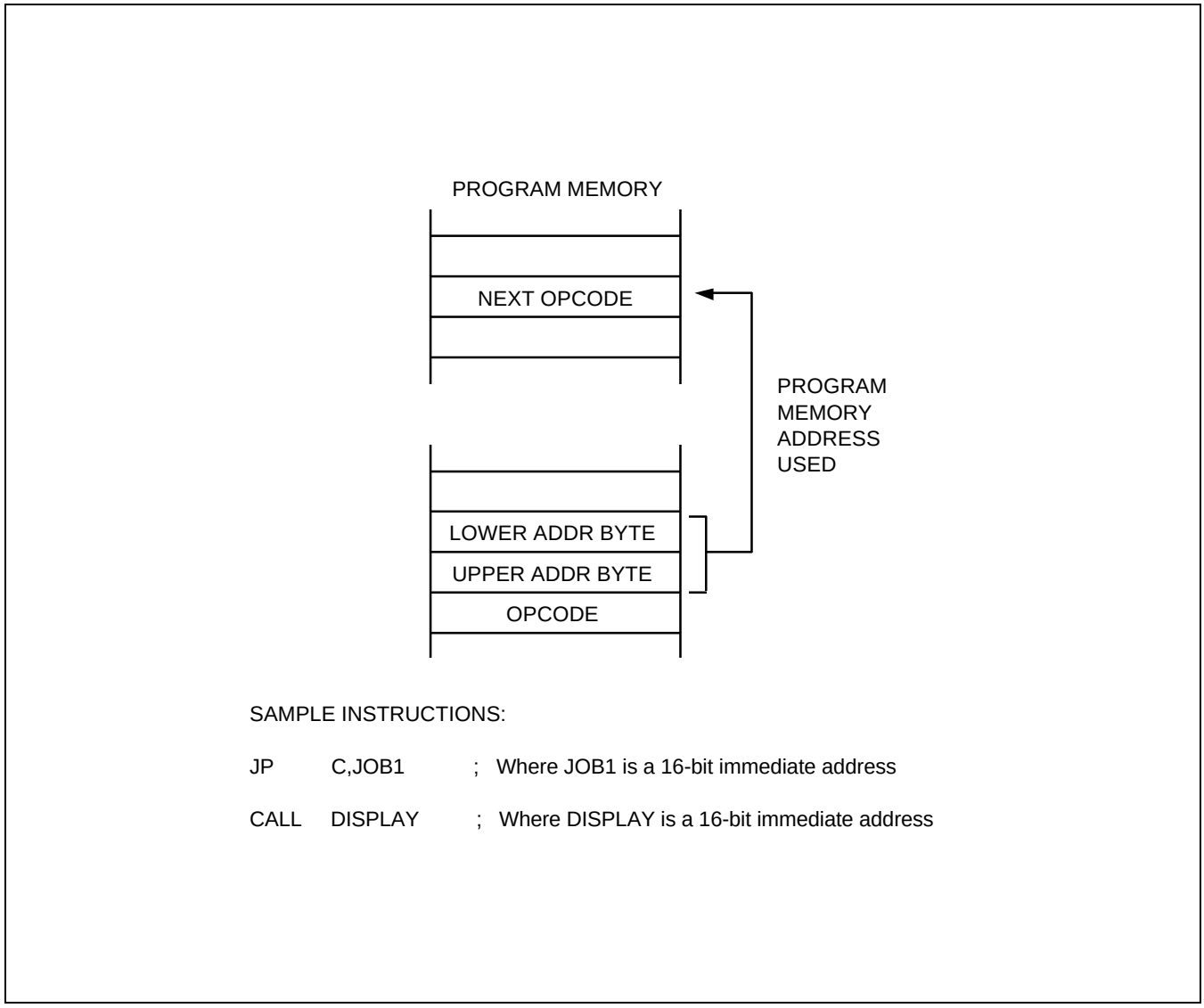


Figure 3-11. Direct Addressing for Call and Jump Instructions

RELATIVE ADDRESS MODE (RA)

In Relative Address (RA) mode, a two's-complement signed displacement between – 128 and + 127 is specified in the instruction. The displacement value is then added to the current PC value. The result is the address of the next instruction to be executed. Before this addition occurs, the PC contains the address of the instruction immediately following the current instruction.

The instructions that support RA addressing is JR.

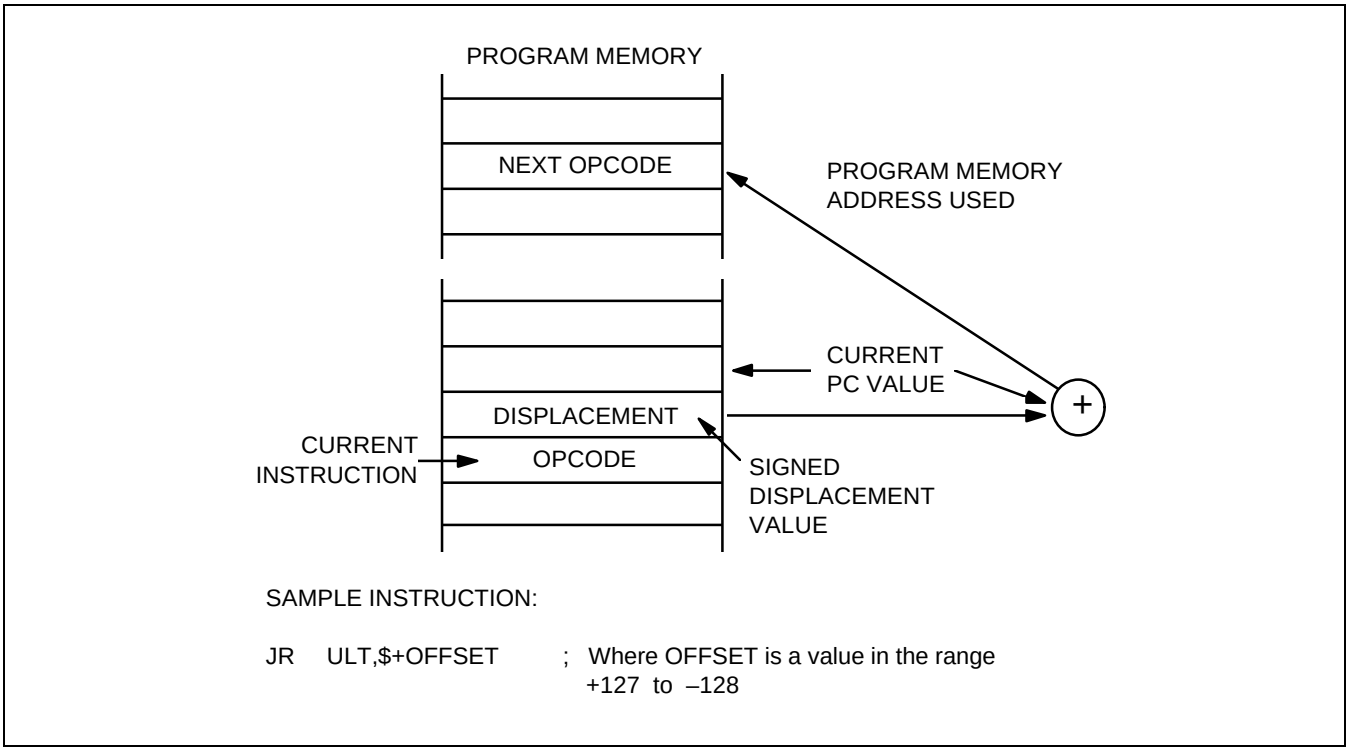


Figure 3-12. Relative Addressing

IMMEDIATE MODE (IM)

In Immediate (IM) addressing mode, the operand value used in the instruction is the value supplied in the operand field itself. Immediate addressing mode is useful for loading constant values into registers.

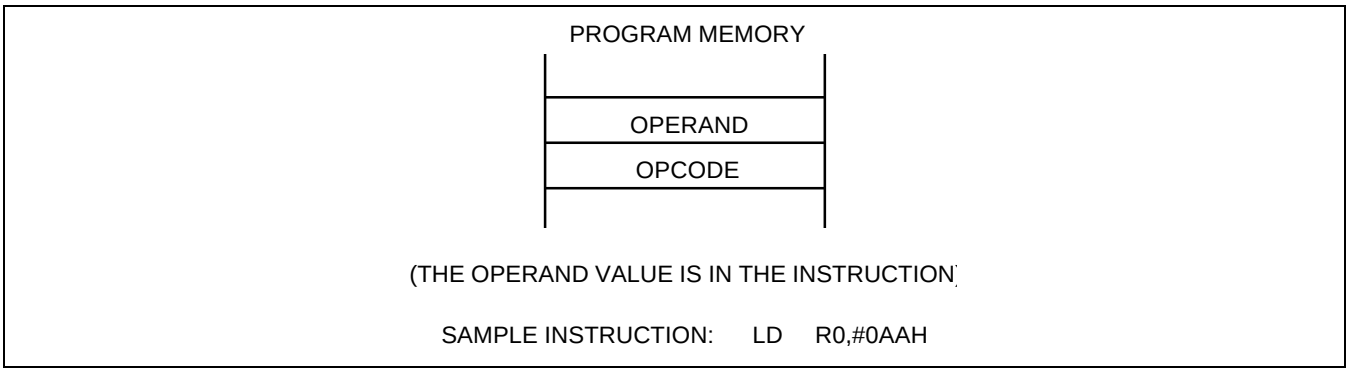


Figure 3-13. Immediate Addressing

4 CONTROL REGISTERS

OVERVIEW

In this section, detailed descriptions of the S3C9644/C9648/P9648 control registers are presented in an easy-to-read format. These descriptions will help familiarize you with the mapped locations in the register file. You can also use them as a quick-reference source when writing application programs.

System and peripheral registers are summarized in Table 4-1. Figure 4-1 illustrates the important features of the standard register description format.

Control register descriptions are arranged in alphabetical order according to register mnemonic. More information about control registers is presented in the context of the various peripheral hardware descriptions in Part II of this manual.

Table 4-1. System and Peripheral control Registers

Register Name	Mnemonic	Decimal	Hex	R/W
Timer 0 counter register	T0CNT	208	D0H	R
Timer 0 data register	T0DATA	209	D1H	R/W
Timer 0 control register	T0CON	210	D2H	R/W
Location D3H is not mapped.				
Clock control register	CLKCON	212	D4H	R/W
System flags register	FLAGS	213	D5H	R/W
Locations D6H-D7H are not mapped.				
Port 0 interrupt control register	P0INT	216	D8H	R/W
Stack pointer	SP	217	D9H	R/W
Port 0 interrupt pending register	P0PND	218	DAH	R/W
Location DBH is not mapped.				
Basic timer control register	BTCON	220	DCH	R/W
Basic timer counter register	BTCNT	221	DDH	R
Location DEH is not mapped.				
System mode register	SYM	223	DFH	R/W
Port 0 data register	P0	224	E0H	R/W
Port 1 data register	P1	225	E1H	R/W
Port 2 data register	P2	226	E2H	R/W
Port 3 data register	P3	227	E3H	R/W
Port 4 data register	P4	228	E4H	R/W
Port 3 control register	P3CON	229	E5H	R/W
Port 0 control register (high byte)	P0CONH	230	E6H	R/W
Port 0 control register (low byte)	P0CONL	231	E7H	R/W
Port 1 control register (high byte)	P1CONH	232	E8H	R/W
Port 1 control register (low byte)	P1CONL	233	E9H	R/W
Port 2 control register (high byte)	P2CONH	234	EAH	R/W
Port 2 control register (low byte)	P2CONL	235	EBH	R/W
Port 2 interrupt control register	P2INT	236	ECH	R/W
Port 2 interrupt pending register	P2PND	237	EDH	R/W
Port 4 control register	P4CON	238	EEH	R/W
Port 4 interrupt enable/pending register	P4INTPND	239	EFH	R/W

Table 4-1. System and Peripheral control Registers (Continued)

Register Name	Mnemonic	Decimal	Hex	R/W
USB function address register	FADDR	240	F0H	R/W
Control endpoint status register	EP0CSR	241	F1H	R/W
Interrupt endpoint 1 control status register	EP1CSR	242	F2H	R/W
Control endpoint byte count register	EP0BCNT	243	F3H	R
Control endpoint FIFO register	EP0FIFO	244	F4H	R/W
Interrupt endpoint 1 FIFO register	EP1FIFO	245	F5H	W
USB interrupt pending register	USBPND	246	F6H	R/W
USB interrupt enable register	USBINT	247	F7H	R
USB power management register	PWRMGR	248	F8H	R/W
Interrupt endpoint 2 control status register	EP2CSR	249	F9H	R/W
Interrupt endpoint 2 FIFO register	EP2FIFO	250	FAH	W
USB/PS2 Mode select register	USBSEL	251	FBH	R/W
D+/PS2, D-/PS2 data register (Only PS2 Mode)	PS2DATA	252	FCH	R/W
PS2 control and interrupt pending register	PS2CONINT	253	FDH	R/W
USB Tranceiver crossover point control register	XCON	254	FEH	R/W
USB reset register	USBRST	255	FFH	R/W

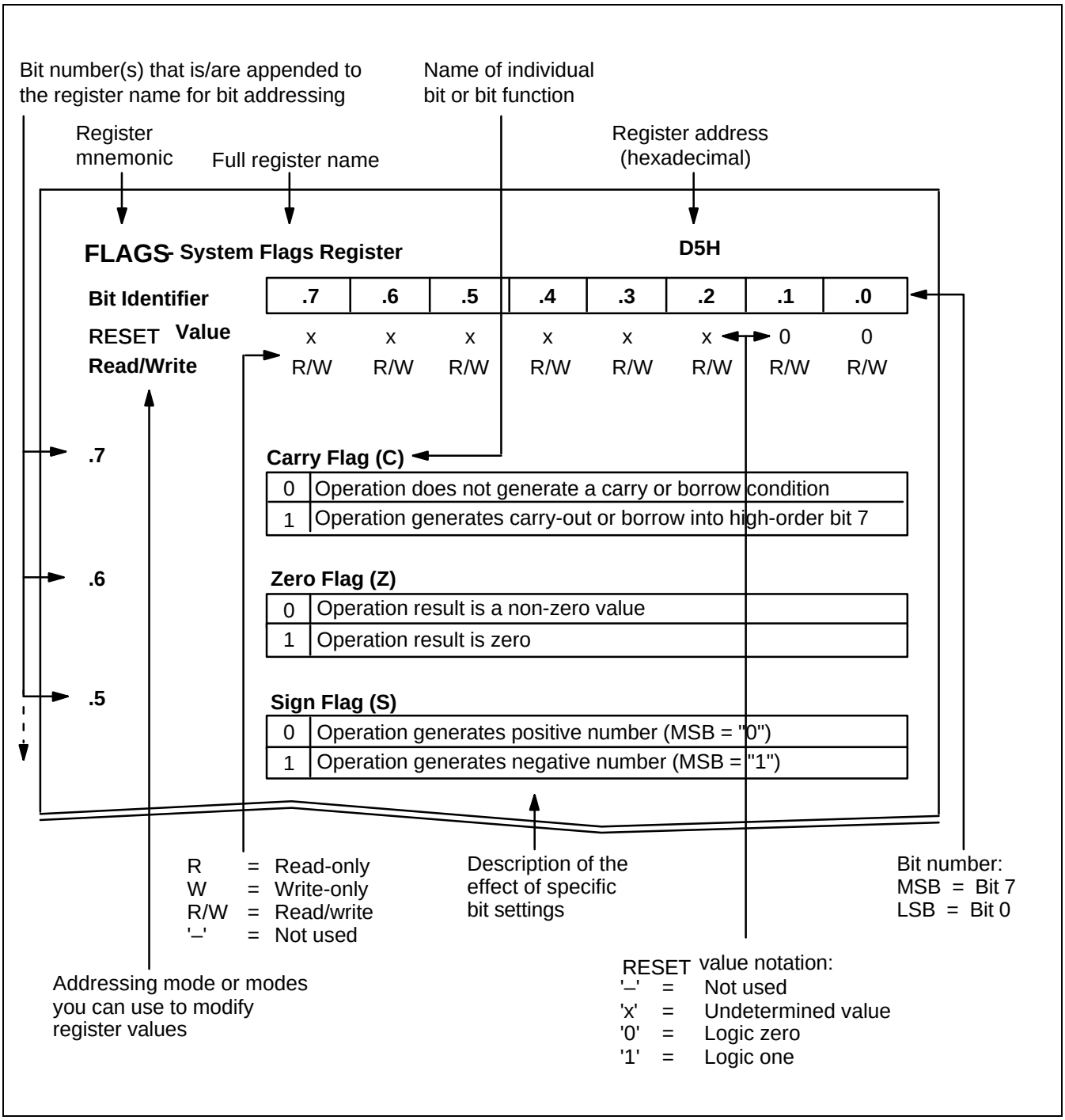


Figure 4-1. Register Description Format

BTCON — Basic Timer Control Register DCH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.4 **Watchdog Timer Enable Bits**

1	0	1	0	Disable watchdog function
Any other value				Enable watchdog function

.3 and .2 **Basic Timer Input Clock Selection Bits**

0	0	$f_{OSC}/4096$
0	1	$f_{OSC}/1024$
1	0	$f_{OSC}/128$
1	1	Invalid setting

.1 **Basic Timer Counter Clear Bit (note)**

0	No effect
1	Clear BTCNT

.0 **Basic Timer Divider Clear Bit (note)**

0	No effect
1	Clear both dividers

NOTE: When you write a "1" to BTCON.0 (or BTCON.1), the basic timer counter (or basic timer divider) is cleared. The bit is then cleared automatically to "0".

CLKCON — System Clock Control Register D4H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 Oscillator IRQ Wake-up Function Bit

0	Enable IRQ for main system oscillator wake-up in power down mode
1	Disable IRQ for main system oscillator wake-up in power down mode

.6 and .5

Not used for S3C9644/C9648/P9648

.4 and .3 CPU Clock (System Clock) Selection Bits (1)

0	0	Divide by 16 (f_{OSC}/16)
0	1	Divide by 8 (f_{OSC}/8)
1	0	Divide by 2 (f_{OSC}/2)
1	1	Non-divided clock (f_{OSC}) (2)

.2-.0

Not used for S3C9644/C9648/P9648

NOTES:

1. After a reset, the slowest clock (divided by 16) is selected as the system clock. To select faster clock speeds, load the appropriate values to CLKCON.3 and CLKCON.4.
2. f_{OSC} means oscillator frequency.

EP0CSR — Control Endpoint 0 Status Register

F1H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 Setup Data End Clear Bit

0	No effect (when write)
1	To clear SETUP_END bit

.6 Out Packet Ready Clear Bit

0	No effect (when write)
1	To clear OUT_PKT_RDY bit

.5 STALL Signal Sending Bit

0	No effect (when write)
1	To send STALL signal

.4 Setup Transfer End Bit

0	No effect (when write)
1	SIE sets this bit when a control transfer ends before DATA_END (bit3) is set

.3 Setup Data End Bit

0	No effect (when write)
1	MCU set this bit after loading or unloading the last packet data into the FIFO

.2 STALL Signal Receive Bit

0	MCU clear this bit to end the STALL condition
1	SIE sets this bit if a control transaction is ended due to a protocol violation

.1 In Packet Ready Bit

0	SIE clear this bit once the packet has been successfully sent to the host
1	MCU sets this bit after writing a packet of data into ENDPOINT0 FIFO

.0 Out Packet Ready Bit

0	No effect (when write)
1	SIE sets this bit once a valid token is written to the FIFO

EP1CSR — Control Endpoint 1 Status Register F2H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 Data Toggle Sequence Clear Bit

0	No effect (when write)
1	MCU sets this bit to clear the data toggle sequence bit. The data toggle is initialized to DATA0.

.6-.3 Maximum Packet Size Bits

0	No effect (when write)
1	These bits indicate the maximum packet size for IN endpoint, and needs to be updated by the MCU before it sets IN_PKT_RDY. Once set, the contents are valid till MCU re-writes them.

.2 FIFO Flush Bit

0	No effect (when write)
1	When MCU writes a one to this register, the FIFO is flushed, and IN_PKT_RDY cleared. The MCU should wait for IN_PKT_RDY to be cleared for the flush to take place.

.1 Force STALL Bit

0	No effect (when write)
1	MCU writes a 1 to this register to issue a STALL handshake to USB. MCU clears this bit, to end the STALL condition.

.0 In Packet Ready Bit

0	SIE clear this bit once the packet has been successfully sent to the host
1	MCU sets this bit, after writing a packet of data into ENDPOINT1 FIFO. USB clears this bit, once the packet has been successfully sent to the host. An interrupt is generated when USB clears this bit, so MCU can load the next packet.

EP2CSR — Control Endpoint 2 Status Register F9H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 Data Toggle Sequence Clear Bit

0	No effect (when write)
1	MCU sets this bit to clear the data toggle sequence bit. The data toggle is initialized to DATA0.

.6-.3 Maximum Packet Size Bits

0	No effect (when write)
1	These bits indicate the maximum packet size for IN endpoint, and needs to be updated by the MCU before it sets IN_PKT_RDY. Once set, the contents are valid till MCU re-writes them.

.2 FIFO Flush Bit

0	No effect (when write)
1	When MCU writes a one to this register, the FIFO is flushed, and IN_PKT_RDY cleared. The MCU should wait for IN_PKT_RDY to be cleared for the flush to take place.

.1 Force STALL Bit

0	No effect (when write)
1	MCU writes a 1 to this register to issue a STALL handshake to USB. MCU clears this bit, to end the STALL condition.

.0 In Packet Ready Bit

0	SIE clear this bit once the packet has been successfully sent to the host
1	MCU sets this bit, after writing a packet of data into ENDPOINT1 FIFO. USB clears this bit, once the packet has been successfully sent to the host. An interrupt is generated when USB clears this bit, so MCU can load the next packet.

FADDR — USB Function Address Register F0H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 Not used for S3C9644/C9648/P9648

.6-.0 FADDR

This register holds the USB address assigned by the host computer. FADDR is located at address F0H and is read/write addressable.

FLAGS — System Flags Register D5H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	–	–	–	–
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 Carry Flag (C)

0	Operation does not generate a carry or borrow condition

.6 Zero Flag (Z)

0	Operation result is a non-zero value
1	Operation result is zero

.5 Sign Flag (S)

0	Operation generates a positive number (MSB = "0")
1	Operation generates a negative number (MSB = "1")

.4 Overflow Flag (V)

0	Operation result is ≤ +127 or _ –128
1	Operation result is _ +127 or ≤ –128

.3-0. Not used for S3C9644/C9648/P9648

P0CONH — Port 0 Control Register (High Byte) E6H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **Port 0, P0.7 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.5 and .4 **Port 0, P0.6 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.3 and .2 **Port 0, P0.5 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.1 and .0 **Port 0, P0.4 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

P0CONL — Port 0 Control Register (Low Byte) E7H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **Port 0, P0.3 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.5 and .4 **Port 0, P0.2 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.3 and .2 **Port 0, P0.1 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.1 and .0 **Port 0, P0.0 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

P0INT — Port 0 Interrupt Control Register D8H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 P0.7 Configuration Bits

0	External interrupt disable
1	External interrupt enable

.6 P0.6 Configuration Bits

0	External interrupt disable
1	External interrupt enable

.5 P0.5 Configuration Bits

0	External interrupt disable
1	External interrupt enable

.4 P0.4 Configuration Bits

0	External interrupt disable
1	External interrupt enable

.3 P0.3 Configuration Bits

0	External interrupt disable
1	External interrupt enable

.2 P0.2 Configuration Bits

0	External interrupt disable
1	External interrupt enable

.1 P0.1 Configuration Bits

0	External interrupt disable
1	External interrupt enable

.0 P0.0 Configuration Bits

0	External interrupt disable
1	External interrupt enable

P0PND — Port 0 Interrupt Pending Register DAH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write (NOTE)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 P0.7 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.6 P0.6 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.5 P0.5 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.4 P0.4 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.3 P0.3 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.2 P0.2 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.1 P0.1 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.0 P0.0 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

P1CONH — Port 1 Control Register (High Byte) E8H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **Port 1, P1.7 Configuration Bits**

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.5 and .4 **Port 1, P1.6 Configuration Bits**

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.3 and .2 **Port 1, P1.5 Configuration Bits**

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.1 and .0 **Port 1, P1.4 Configuration Bits**

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

P1CONL — Port 1 Control Register (Low Byte) E9H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **Port 1, P1.3 Configuration Bits**

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.5 and .4 **Port 1, P1.2 Configuration Bits**

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.3 and .2 **Port 1, P1.1 Configuration Bits**

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.1 and .0 **Port 1, P1.0 Configuration Bits**

0	0	Schmitt trigger input
0	1	Schmitt trigger input with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

P2CONH — Port 2 Control Register (High Byte) EAH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **Port 2, P2.7 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edges external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.5 and .4 **Port 2, P2.6 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edges external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.3 and .2 **Port 2, P2.5 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edges external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.1 and .0 **Port 2, P2.4 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edges external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

P2CONL — Port 2 Control Register (Low Byte) EBH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **Port 2, P2.3 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edges external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.5 and .4 **Port 2, P2.2 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edges external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.3 and .2 **Port 2, P2.1 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edges external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.1 and .0 **Port 2, P2.0 Configuration Bits**

0	0	Schmitt trigger input, rising edge external interrupt
0	1	Schmitt trigger input, falling edges external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

P2INT — Port 2 Interrupt Enable Register ECH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 P2.7 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.6 P2.6 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.5 P2.5 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.4 P2.4 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.3 P2.3 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.2 P2.2 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.1 P2.1 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.0 P2.0 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

P2PND — Port 2 Interrupt Pending Register EDH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write (NOTE)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 P2.7 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.6 P2.6 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.5 P2.5 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.4 P2.4 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.3 P2.3 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.2 P2.2 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.1 P2.1 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

.0 P2.0 Interrupt Pending Bit

0	No pending (when read)/clear pending bit (when write)
1	Pending (when read)/no effect (when write)

NOTE: To clear a port 2 interrupt pending condition, write a "0" to the corresponding P2PND register bit location.

P3CON — Port 3 Control Register E5H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **Port 3, P3.3 Configuration Bits**

0	0	Schmitt trigger input
0	1	System clock output(CLO) mode. CLO comes from system clock circuit.
1	0	Push-pull output
1	1	N-channel open-drain output mode

.5 and .4 **Port 3, P3.2 Configuration Bits**

0	x	Schmitt trigger input
1	0	Push-pull output
1	1	N-channel open-drain output mode

.3 and .2 **Port 3, P3.1 Configuration Bits**

0	x	Schmitt trigger input
1	0	Push-pull output
1	1	N-channel open-drain output mode

.1 and .0 **Port 3, P3.0 Configuration Bits**

0	x	Schmitt trigger input
1	0	Push-pull output
1	1	N-channel open-drain output mode

NOTE: "x" means don't care

P4CON — Port 4 Control Register EEH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **Port 4, P4.3 Configuration Control Bits**

0	0	Schmitt trigger input, falling edge external interrupt with pull-up
0	1	N-CH open drain output mode with pull-up
1	0	N-CH open drain output mode
1	1	Output pull-pull mode

.5 and .4 **Port 4, P4.2 Configuration Control Bits**

0	0	Schmitt trigger input, falling edge external interrupt with pull-up
0	1	N-CH open drain output mode with pull-up
1	0	N-CH open drain output mode
1	1	Output pull-pull mode

.3 and .2 **Port 4, P4.1 Configuration Control Bits**

0	0	Schmitt trigger input, falling edge external interrupt with pull-up
0	1	N-CH open drain output mode with pull-up
1	0	N-CH open drain output mode
1	1	Output pull-pull mode

.1 and .0 **Port 4, P4.0 Configuration Control Bits**

0	0	Schmitt trigger input, falling edge external interrupt with pull-up
0	1	N-CH open drain output mode with pull-up
1	0	N-CH open drain output mode
1	1	Output pull-pull mode

P4INTPND — Port 4 Interrupt Enable and Pending Register EFH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 P4.3 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.6 P4.2 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.5 P4.1 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.4 P4.0 Interrupt Enable Bit

0	External interrupt disable
1	External interrupt enable

.3 P4.3 Interrupt Pending Bit

0	No pending (when bit is read)/clear pending bit (when bit is write)
1	Pending (when bit is read)/no effect (when bit is write)

.2 P4.2 Interrupt Pending Bit

0	No pending (when bit is read)/clear pending bit (when bit is write)
1	Pending (when bit is read)/no effect (when bit is write)

.1 P4.1 Interrupt Pending Bit

0	No pending (when bit is read)/clear pending bit (when bit is write)
1	Pending (when bit is read)/no effect (when bit is write)

.0 P4.0 Interrupt Pending Bit

0	No pending (when bit is read)/clear pending bit (when bit is write)
1	Pending (when bit is read)/no effect (when bit is write)

PS2CONINT — PS2 Control and interrupt pending Register (PS2 Mode only) EEH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 **D+/PS2 Configuration Control Bits**

0	0	Schmitt trigger input, falling edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.5 and .4 **D-/PS2 Configuration Control Bits**

0	0	Schmitt trigger input, falling edge external interrupt
0	1	Schmitt trigger input, falling edge external interrupt with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

.4 **D+/PS2 Interrupt Enable Bit**

0	External interrupt disable
1	External interrupt enable

.3 **D-/PS2 Interrupt Enable Bit**

0	External interrupt disable
1	External interrupt enable

.1 **D+/PS2 Interrupt Pending Bit**

0	No pending (when bit is read)/clear pending bit (when bit is write)
1	Pending (when bit is read)/no effect (when bit is write)

.0 **D-/PS2 Interrupt Pending Bit**

0	No pending (when bit is read)/clear pending bit (when bit is write)
1	Pending (when bit is read)/no effect (when bit is write)

PWRMGR — USB Power Management Register F8H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.2

Not used for S3C9644/C9648/P9648

.1 RESUME Signal Sending Bit

0	RESUME signal is ended
1	While in suspend state, if the MCU wants to initiate a resume, it writes a 1 to this register for 10ms (maximum of 15ms), and clears this register. In suspend mode if this bit is a 1, USB generates resume signaling.

.0 SUSPEND Status Bit

0	Cleared when MCU writes a zero to RESUME signal sending bit or function receives resume signal from the host while in suspend mode
1	This bit is set when SUSPEND interrupt occur

SYM — System Mode Register DFH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	0	0	0
Read/Write	–	–	–	–	–	R/W	R/W	R/W

.7-.3 **Not used for S3C9644/C9648/P9648**

.2 **Global Interrupt Enable Bit (note)**

0	Disable global interrupt processing
1	Enable global interrupt processing

.1 and .0 **Page Selection Bits**

0	0	Addressing page 0 locations for S3C9644/C9648/P9648
Other values		Enable global interrupt processing

NOTE: SYM must be selected bit 1 and 0 into 00 for S3C9644/C9648/P9648.

T0CON — Timer 0 Control Register D2H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7 and .6 T0 Counter Input Clock Selection Bits

0	0	CPU clock/4096
0	1	CPU clock/256
1	0	CPU clock/8
1	1	Invalid selection

.5 and .4 T0 Operating Mode Selection Bits

0	0	Interval timer mode (The counter is automatically cleared whenever T0DATA value equals to T0CNT value)
0	1	Invalid selection
1	0	
1	1	Overflow mode (OVF interrupt can occur)

.3 T0 Counter Clear Bit (T0CLR)

0	No effect when written
1	Clear T0 counter

.2 T0 Overflow Interrupt Enable Bit (T0OVF)

0	Disable T0 overflow interrupt
1	Enable T0 overflow interrupt

.1 T0 Match Interrupt Enable Bit (T0INT)

0	Disable T0 match interrupt
1	Enable T0 match interrupt

.0 T0 Interrupt Pending Bit (T0PND)

0	No interrupt pending/*Clear this pending bit (when write)*
1	Interrupt is pending(when read)/No effect(when write)

NOTE: When you write a "1" to T0CON.3, the timer 0 counter is cleared. The bit is then cleared automatically to "0".

USBPND — USB Interrupt Pending Register F6H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.4

Not used for S3C9644/C9648/P9648

.3 ENDPOINT 2 Interrupt Pending Bit

0	No effect (once read, this bit is cleared automatically)
1	This bit is set, when endpoint2 needs to be serviced

.3 RESUME Interrupt Pending Bit

0	No effect (once read, this bit is cleared automatically)
1	While in suspend mode, if resume signaling is received this bit gets set

.2 SUSPEND Interrupt Pending Bit

0	No effect (once read, this bit is cleared automatically)
1	This bit is set, when suspend signaling is received

.1 ENDPOINT1 Interrupt Pending Bit

0	No effect (once read, this bit is cleared automatically)
1	This bit is set, when endpoint1 needs to be serviced

.0 ENDPOINT0 Interrupt Pending Bit

0	No effect (once read, this bit is cleared automatically)
1	This bit is set, while endpoint 0 needs to serviced. It is set under the following conditions; — OUT_PKT_RDY is set — IN_PKT_RDY get cleared — SENT_STALL gets set — DATA_END gets cleared — SETUP_END gets set

USBINT — USB Interrupt Enable Register

F7H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	1	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.3 Not used for S3C9644/C9648/P9648

.3 ENDPOINT2 Interrupt Pending Bit

0	Disable ENDPOINT 2 interrupt
1	Enable ENDPOINT 2 interrupt

.2 SUSPEND/RESUME Interrupt Enable Bit

0	Disable SUSPEND and RESEME interrupt
1	Enable SUSPEND and RESEME interrupt

.1 ENDPOINT1 Interrupt Pending Bit

0	Disable ENDPOINT 1 interrupt
1	Enable ENDPOINT 1 interrupt

.0 ENDPOINT0 Interrupt Pending Bit

0	Disable ENDPOINT 0 interrupt
1	Enable ENDPOINT 0 interrupt

USBSEL — USB/PS2 Mode select Register FBH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.1 Not used for S3C9644/C9648/P9648

.0 **USB/PS2 Mode select Bit**

0	PS2 Mode
1	USB Mode

XCON — USB Signal Crossover Point Control Register FEH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	0	0	0	0	0	0
Read/Write	–	–	R/W	R/W	R/W	R/W	R/W	R/W

.7-.6 Not used for S3C9644/C9648/P9648

.5-.0 USB Signal Crossover Point Control Bit

Edge delay Control	Bit 5, (2)	Bit 4, (1)	Bit 3, (0)	Delay Value	Delay Unit
RISE edge	0	0	0	0	(about) 2.5nsec
		0	1	1	
		1	0	2	
		1	1	4	
FALL edge	1	0	0	0	
		0	1	1	
		1	0	2	
		1	1	4	

USBRST — USB Reset Register FFH

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	–	–	–	–	–	–	–	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

.7-.1 Not used for S3C9644/C9648/P9648

.0 USB Reset Signal Receive Bit

0	Clear reset signal bit
1	This bit is set when host send USB reset signal

5 INTERRUPT STRUCTURE

OVERVIEW

The SAM87RI interrupt structure has two basic components: a vector, and sources. The number of interrupt sources can be serviced through a interrupt vector which is assigned in ROM address 0000H-0001H.

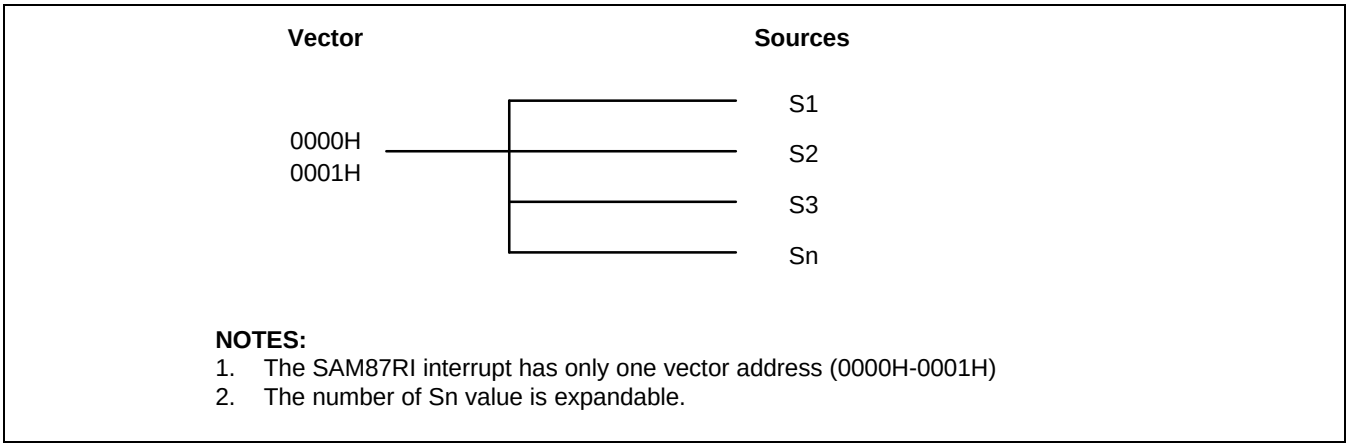


Figure 5-1. S3C9-Series Interrupt Type

INTERRUPT PROCESSING CONTROL POINTS

Interrupt processing can be controlled in two ways: either globally, or by specific interrupt level and source. The system-level control points in the interrupt structure are therefore:

— Global interrupt enable and disable (by EI and DI instructions)

— Interrupt source enable and disable settings in the corresponding peripheral control register(s)

ENABLE/DISABLE INTERRUPT INSTRUCTIONS (EI, DI)

The system mode register, SYM (DFH), is used to enable and disable interrupt processing.

SYM.2 is the enable and disable bit for global interrupt processing respectively, by modifying SYM.2. An Enable Interrupt (EI) instruction must be included in the initialization routine that follows a reset operation in order to enable interrupt processing. Although you can manipulate SYM.2 directly to enable and disable interrupts during normal operation, we recommend that you use the EI and DI instructions for this purpose.

INTERRUPT PENDING FUNCTION TYPES

When the interrupt service routine has executed, the application program's service routine must clear the appropriate pending bit before the return from interrupt subroutine (IRET) occurs.

INTERRUPT PRIORITY

Because there is not a interrupt priority register in SAM87RI, the order of service is determined by a sequence of source which is executed in interrupt service routine.

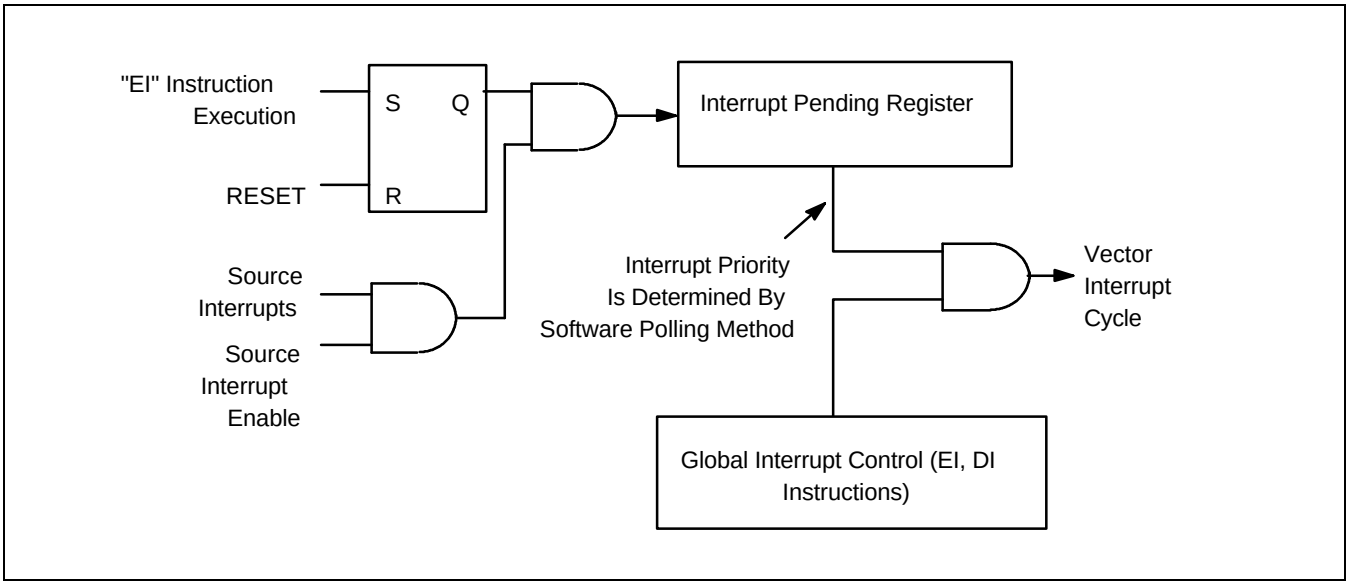


Figure 5-2. Interrupt Function Diagram

INTERRUPT SOURCE SERVICE SEQUENCE

The interrupt request polling and servicing sequence is as follows:

1. A source generates an interrupt request by setting the interrupt request pending bit to "1".
2. The CPU generates an interrupt acknowledge signal.
3. The service routine starts and the source's pending flag is cleared to "0" by software.
4. Interrupt priority must be determined by software polling method.

INTERRUPT SERVICE ROUTINES

Before an interrupt request can be serviced, the following conditions must be met:

— Interrupt processing must be enabled (EI, SYM.2 = "1")

— Interrupt must be enabled at the interrupt's source (peripheral control register)

If all of the above conditions are met, the interrupt request is acknowledged at the end of the instruction cycle. The CPU then initiates an interrupt machine cycle that completes the following processing sequence:

1. Reset (clear to "0") the global interrupt enable bit in the SYM register (DI, SYM.2 = "0") to disable all subsequent interrupts.
2. Save the program counter and status flags to stack.
3. Branch to the interrupt vector to fetch the service routine's address.
4. Pass control to the interrupt service routine.

When the interrupt service routine is completed, an Interrupt Return instruction (IRET) occurs. The IRET restores the PC and status flags and sets SYM.2 to "1"(EI), allowing the CPU to process the next interrupt request.

GENERATING INTERRUPT VECTOR ADDRESSES

The interrupt vector area in the ROM contains the address of the interrupt service routine. Vectored interrupt processing follows this sequence:

1. Push the program counter's low-byte value to stack.
2. Push the program counter's high-byte value to stack.
3. Push the FLAGS register values to stack.
4. Fetch the service routine's high-byte address from the vector address 0000H.
5. Fetch the service routine's low-byte address from the vector address 0001H.
6. Branch to the service routine specified by the 16-bit vector address.

S3C9644/C9648/P9648 INTERRUPT STRUCTURE

The S3C9644/C9648/P9648 microcontroller has fourteen peripheral interrupt sources:

— Timer 0 match interrupt
— Timer 0 overflow interrupt
— Eight external interrupts for port 2, P2.0-P2.7
— Four external interrupts for port 4, P4.0-P4.3

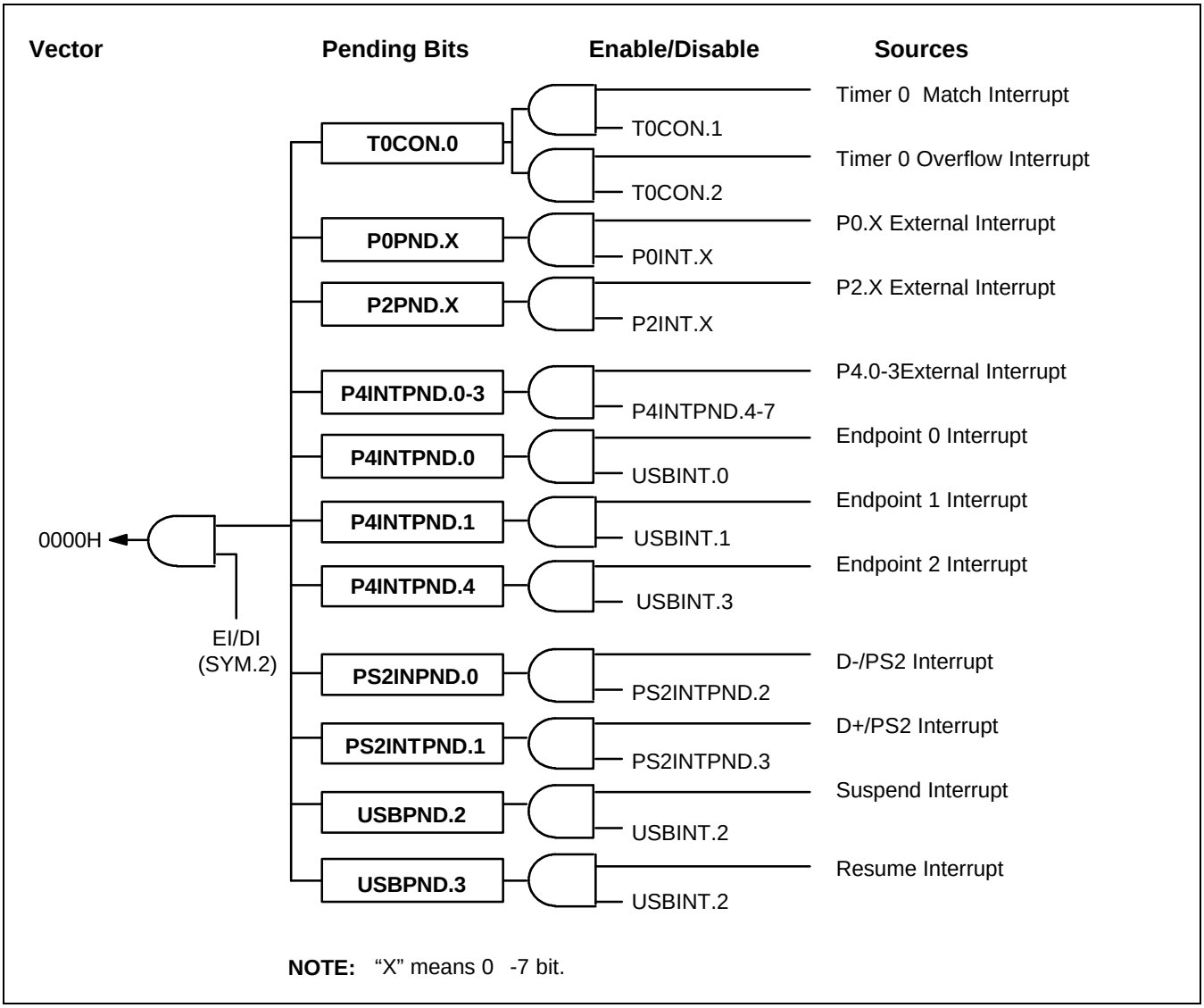


Figure 5-3. S3C9644/C9648/P9648 Interrupt Structure

7 CLOCK CIRCUIT

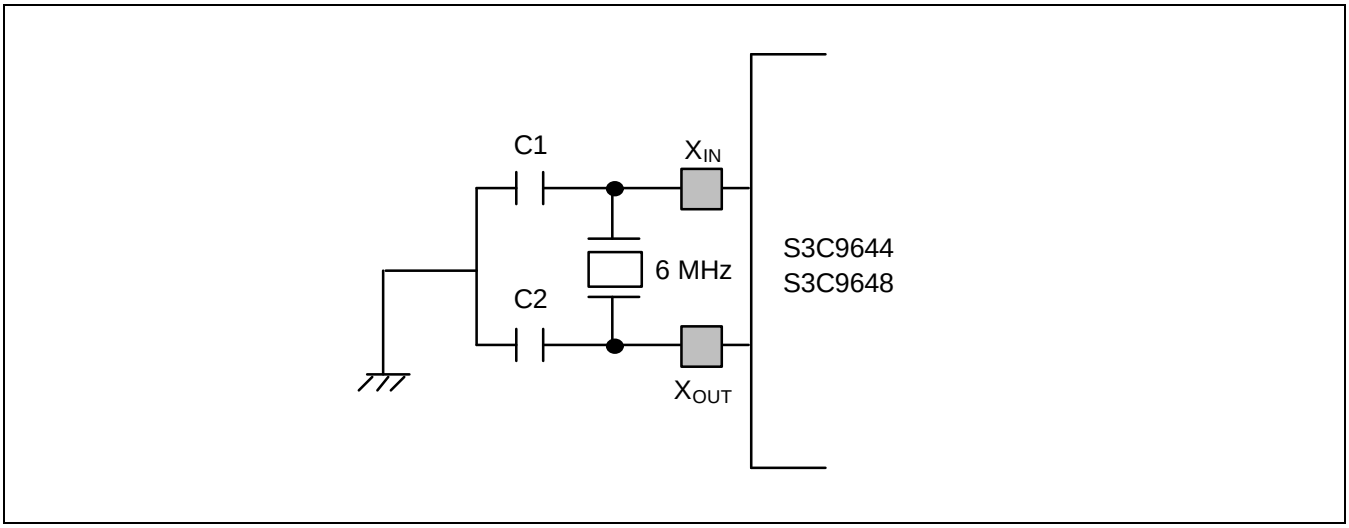


Figure 7-1. Main Oscillator Circuit (Crystal/Ceramic Oscillator)

MAIN OSCILLATOR LOGIC

To increase processing speed and to reduce clock noise, non-divided logic is implemented for the main oscillator circuit. For this reason, very high resolution waveforms (square signal edges) must be generated in order for the CPU to efficiently process logic operations.

CLOCK STATUS DURING POWER-DOWN MODES

The two power-down modes, Stop mode and Idle mode, affect clock oscillation as follows:

— In Stop mode, the main oscillator "freezes", halting the CPU and peripherals. The contents of the register file and current system register values are retained. Stop mode is released, and the oscillator started, by a reset operation or by an external interrupt with RC-delay noise filter (for S3C9644/C9648/P9648, INT0-INT2).

— In Idle mode, the internal clock signal is gated off to the CPU, but not to interrupt control and the timer. The current CPU status is preserved, including stack pointer, program counter, and flags. Data in the register file is retained. Idle mode is released by a reset or by an interrupt (external or internally-generated).

SYSTEM CLOCK CONTROL REGISTER (CLKCON)

The system clock control register, CLKCON, is located in location D4H. It is read/write addressable and has the following functions:

— Oscillator IRQ wake-up function enable/disable (CLKCON.7)

— Oscillator frequency divide-by value: non-divided, 2, 8 or 16 (CLKCON.4 and CLKCON.3)

The CLKCON register controls whether or not an external interrupt can be used to trigger a Stop mode release (This is called the "IRQ wake-up" function). The IRQ wake-up enable bit is CLKCON.7.

After a reset, the external interrupt oscillator wake-up function is enabled, the main oscillator is activated, and the $f_{OSC}/16$ (the slowest clock speed) is selected as the CPU clock. If necessary, you can then increase the CPU clock speed to f_{OSC}, $f_{OSC}/2$ or $f_{OSC}/8$.

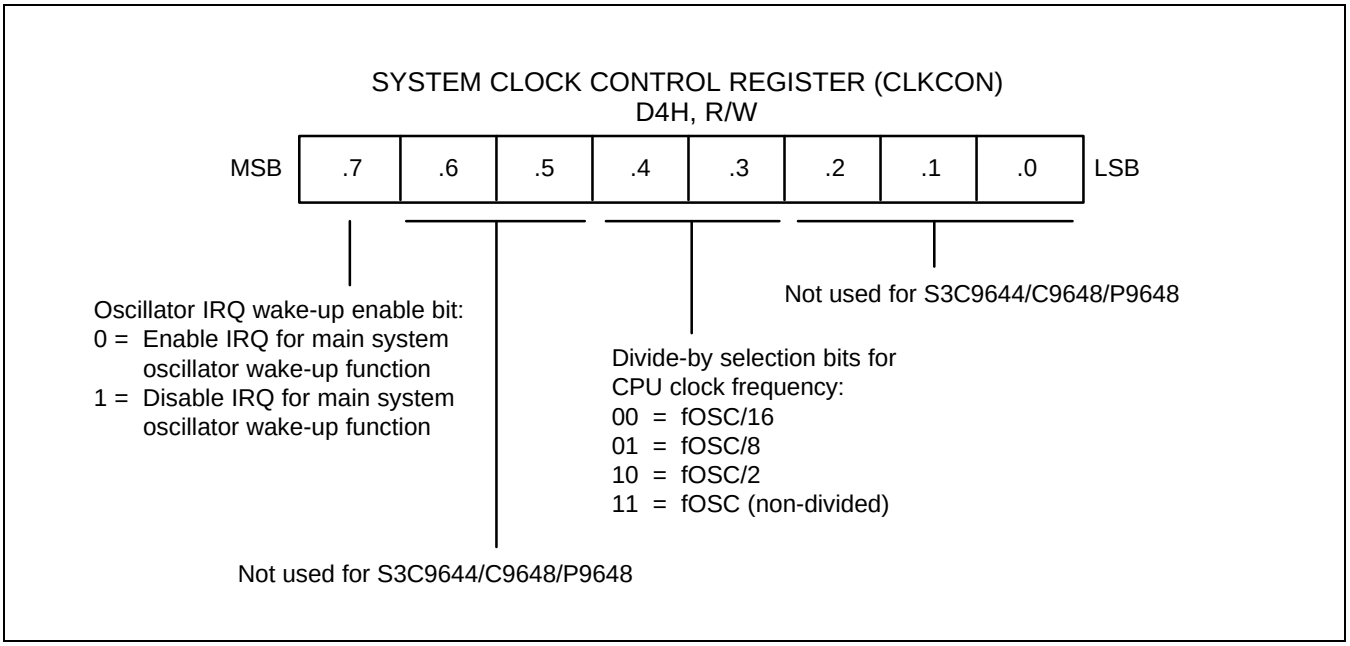


Figure 7-2. System Clock Control Register (CLKCON)

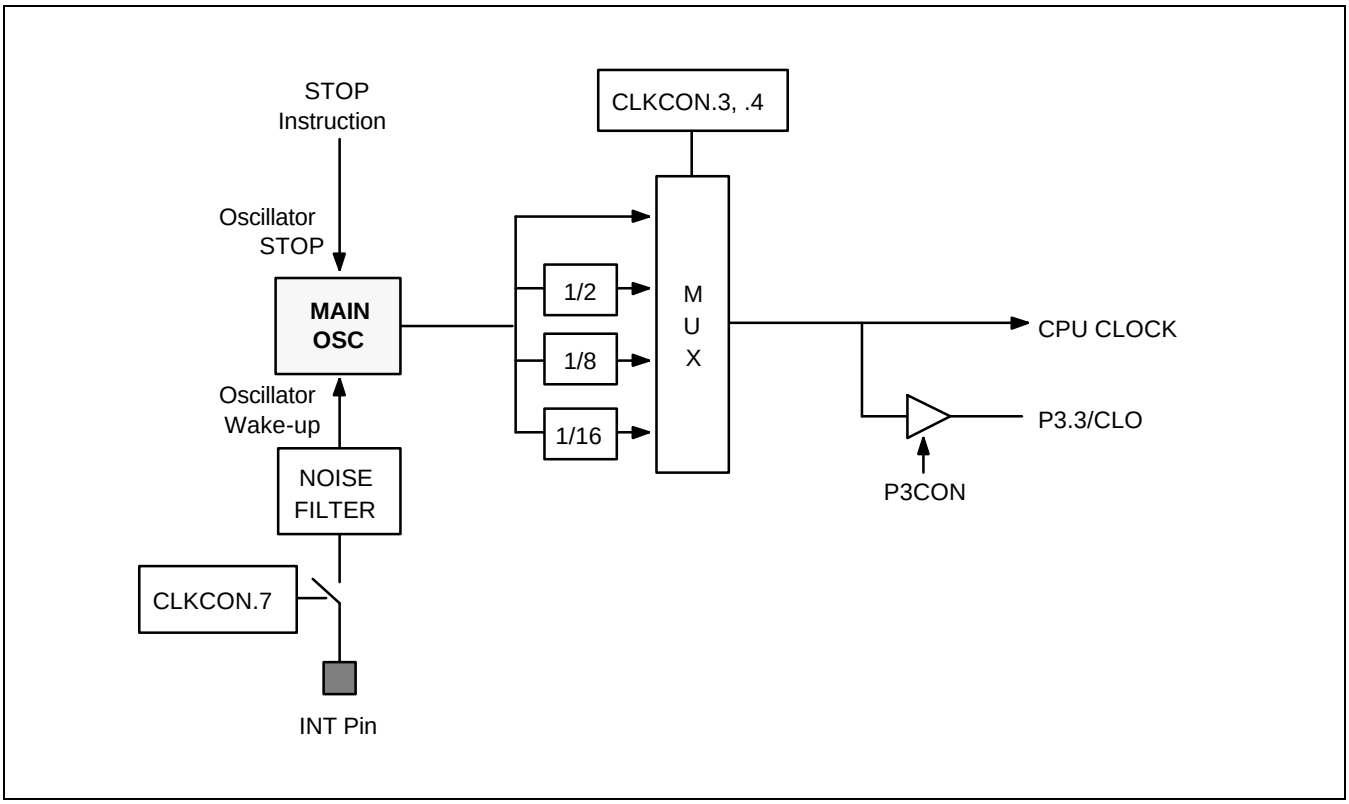


Figure 7-3. System Clock Circuit Diagram

8 RESET AND POWER-DOWN

SYSTEM RESET

OVERVIEW

During a power-on reset, the voltage at V_{DD} is High level and the RESET pin is forced to Low level. The RESET signal is input through a Schmitt trigger circuit where it is then synchronized with the CPU clock. This brings the S3C9644/C9648/P9648 into a known operating status.

The RESET pin must be held to Low level for a minimum time interval after the power supply comes within tolerance in order to allow time for internal CPU clock oscillation to stabilize. The minimum required oscillation stabilization time for a reset is approximately 10ms (@ $2^{16}/f_{OSC}$, f_{OSC} = 6 MHz).

When a reset occurs during normal operation (with both V_{DD} and RESET at High level), the signal at the RESET pin is forced Low and the reset operation starts. All system and peripheral control registers are then set to their default hardware reset values (see Table 8-1).

The following sequence of events occurs during a reset operation:

— All interrupts are disabled.
— The watchdog function (basic timer) is enabled.
— Ports 0-4 are set to Schmitt trigger input mode and all pull-up resistors are disabled.
— Peripheral control and data registers are disabled and reset to their initial values.
— The program counter is loaded with the ROM reset address, 0100H.
— When the programmed oscillation stabilization time interval has elapsed, the address stored in ROM location 0100H (and 0101H) is fetched and executed.

NOTE

To program the duration of the oscillation stabilization interval, you must make the appropriate settings to the basic timer control register, BTCON, before entering Stop mode. Also, if you do not want to use the basic timer watchdog function (which causes a system reset if a basic timer counter overflow occurs), you can disable it by writing '1010B' to the upper nibble of BTCON.

POWER-DOWN MODES

STOP MODE

Stop mode is invoked by the instruction STOP (opcode 7FH). In Stop mode, the operation of the CPU and all peripherals is halted. That is, the on-chip main oscillator stops and the supply current is reduced to less than 300 µA. All system functions are halted when the clock "freezes", but data stored in the internal register file is retained. Stop mode can be released in one of two ways: by a RESET signal or by an external interrupt.

Using RESET to Release Stop Mode

Stop mode is released when the RESET signal is released and returns to High level. All system and peripheral control registers are then reset to their default values and the contents of all data registers are retained. A reset operation automatically selects a slow clock (1/16) because CLKCON.3 and CLKCON.4 are cleared to '00B'. After the oscillation stabilization interval has elapsed, the CPU executes the system initialization routine by fetching the 16-bit address stored in ROM locations 0100H and 0101H.

Using an External Interrupt to Release Stop Mode

Only external interrupts with an RC-delay noise filter circuit can be used to release Stop mode (Clock-related external interrupts cannot be used). External interrupts INT0-INT2 in the S3C9644/C9648/P9648 interrupt structure meet this criteria.

Note that when Stop mode is released by an external interrupt, the current values in system and peripheral control registers are not changed. When you use an interrupt to release Stop mode, the CLKCON.3 and CLKCON.4 register values remain unchanged, and the currently selected clock value is used. If you use an external interrupt for Stop mode release, you can also program the duration of the oscillation stabilization interval. To do this, you must make the appropriate control and clock settings *before* entering Stop mode.

The external interrupt is serviced when the Stop mode release occurs. Following the IRET from the service routine, the instruction immediately following the one that initiated Stop mode is executed.

IDLE MODE

Idle mode is invoked by the instruction IDLE (opcode 6FH). In Idle mode, CPU operations are halted while select peripherals remain active. During Idle mode, the internal clock signal is gated off to the CPU, but not to interrupt logic and timer/counters. Port pins retain the mode (input or output) they had at the time Idle mode was entered.

There are two ways to release Idle mode:

1. Execute a reset. All system and peripheral control registers are reset to their default values and the contents of all data registers are retained. The reset automatically selects a slow clock (1/16) because CLKCON.3 and CLKCON.4 are cleared to '00B'. If interrupts are masked, a reset is the only way to release Idle mode.
2. Activate any enabled interrupt, causing Idle mode to be released. When you use an interrupt to release Idle mode, the CLKCON.3 and CLKCON.4 register values remain unchanged, and the currently selected clock value is used. The interrupt is then serviced. Following the IRET from the service routine, the instruction immediately following the one that initiated Idle mode is executed.

NOTE

Only external interrupts that are not clock-related can be used to release Stop mode. To release Idle mode, however, any type of interrupt (that is, internal or external) can be used.

HARDWARE RESET VALUES

Tables 8-1 through 8-3 list the values for CPU and system registers, peripheral control registers and peripheral data registers following a reset operation in normal operating mode. The following notation is used in these tables to represent specific reset values:

— A "1" or a "0" shows the reset bit value as logic one or logic zero, respectively.

— An 'x' means that the bit value is undefined following a reset.

— A dash ('-') means that the bit is either not used or not mapped.

Table 8-1. Register Values after a Reset

Register Name	Mnemonic	Address		Bit Values after RESET							
		Dec	Hex	7	6	5	4	3	2	1	0
General purpose registers	–	000-191	00H-BFH	x	x	x	x	x	x	x	x
Working registers	R0 - R15	192-207	C0H-CFH	x	x	x	x	x	x	x	x
Timer 0 counter	T0CNT	208	D0H	0	0	0	0	0	0	0	0
Timer 0 data register	T0DATA	209	D1H	1	1	1	1	1	1	1	1
Timer 0 control register	T0CON	210	D2H	0	0	0	0	0	0	0	0
Location D3H is not mapped.											
Clock control register	CLKCON	212	D4H	0	0	0	0	0	0	0	0
System flags register	FLAGS	213	D5H	0	0	0	0	–	–	–	–
Locations D6H - D8H are not mapped.											
Port 0 interrupt control register	P0INT	216	D8H	0	0	0	0	0	0	0	0
Stack pointer	SP	217	D9H	x	x	x	x	x	x	x	x
Port 0 interrupt pending register	P0PND	218	DAH	0	0	0	0	0	0	0	0
Location DBH is not mapped.											
Basic timer control register	BTCON	220	DCH	0	0	0	0	0	0	0	0
Basic timer counter	BTCNT	221	DDH	0	0	0	0	0	0	0	0
Location DEH is not mapped.											
System mode register	SYM	223	DFH	0	–	–	–	–	0	0	0
Port 0 data register	P0	224	E0H	0	0	0	0	0	0	0	0
Port 1 data register	P1	225	E1H	0	0	0	0	0	0	0	0
Port 2 data register	P2	226	E2H	0	0	0	0	0	0	0	0
Port 3 data register	P3	227	E3H	0	0	0	0	0	0	0	0
Port 4 data register	P4	228	E4H	0	0	0	0	0	0	0	0

Table 8-1. Register Values after a Reset (continued)

Bank 0 Register Name	Mnemonic	Address		Bit Values after a Reset							
		Dec	Hex	7	6	5	4	3	2	1	0
Port 3 control register	P3CON	229	E5H	0	0	0	0	0	0	0	0
Port 0 control register (high byte)	P0CONH	230	E6H	0	0	0	0	0	0	0	0
Port 0 control register (low byte)	P0CONL	231	E7H	0	0	0	0	0	0	0	0
Port 1 control register (high byte)	P1CONH	232	E8H	0	0	0	0	0	0	0	0
Port 1 control register (low byte)	P1CONL	233	E9H	0	0	0	0	0	0	0	0
Port 2 control register (high byte)	P2CONH	234	EAH	0	0	0	0	0	0	0	0
Port 2 control register (low byte)	P2CONL	235	EBH	0	0	0	0	0	0	0	0
Port 2 interrupt enable register	P2INT	236	ECH	0	0	0	0	0	0	0	0
Port 2 interrupt pending register	P2PND	237	EDH	0	0	0	0	0	0	0	0
Port 4 control register	P4CON	238	EEH	0	0	0	0	0	0	0	0
Port 4 interrupt enable/pending register	P4INTPND	239	EFH	0	0	0	0	0	0	0	0
USB function address register	FADDR	240	F0H	0	0	0	0	0	0	0	0
Control endpoint status register	EP0CSR	241	F1H	0	0	0	0	0	0	0	0
Interrupt endpoint status register	EP1CSR	242	F2H	0	0	0	0	0	0	0	0
Control endpoint byte count register	EP0BCNT	243	F3H	0	0	0	0	0	0	0	0
Control endpoint FIFO register	EP0FIFO	244	F4H	x	x	x	x	x	x	x	x
Interrupt endpoint FIFO register	EP1FIFO	245	F5H	x	x	x	x	x	x	x	x
USB interrupt pending register	USBPND	246	F6H	0	0	0	0	0	0	0	0
USB interrupt enable register	USBINT	247	F7H	0	0	0	0	0	0	1	1
USB power management register	PWRMGR	248	F8H	0	0	0	0	0	0	0	0
Interrupt endpoint 2 control status register	EP2CSR	249	F9H	0	0	0	0	0	0	0	0
Interrupt endpoint 2 FIFO register	EP2FIFO	250	FAH	x	x	x	x	x	x	x	x
USB/PS2 Mode select register	USBSEL	251	FBH	0	0	0	0	0	0	0	0
D+/PS2, D-/PS2 data register (Only PS2 Mode)	PS2DATA	252	FCH	0	0	0	0	0	0	0	0
PS2 control and interrupt pending register	PS2CONINT	253	FDH	0	0	0	0	0	0	0	0
USB Tranceiver crossover point control register	XCON	254	FEH	–	–	0	0	0	0	0	0
USB reset register	USBRST	255	FFH	x	x	x	x	x	x	x	1

9 I/O PORTS

OVERVIEW

The S3C9644/C9648/P9648 USB Mode has five I/O ports (0-4) with a total of 32 pins.
PS2 Mode has six I/O ports (0-4 and D+/PS2, D-/PS2) with a total of 34 pins.

You can access these ports directly by writing or reading port data register addresses.

For keyboard applications, ports 0, 1 and 2 are usually configured as keyboard matrix input/output. Port 3 can be configured as LED drive. Port 4 is used for host communication or for controlling a mouse or other external device.

Table 9-1. S3C9644/C9648/P9648 Port Configuration Overview

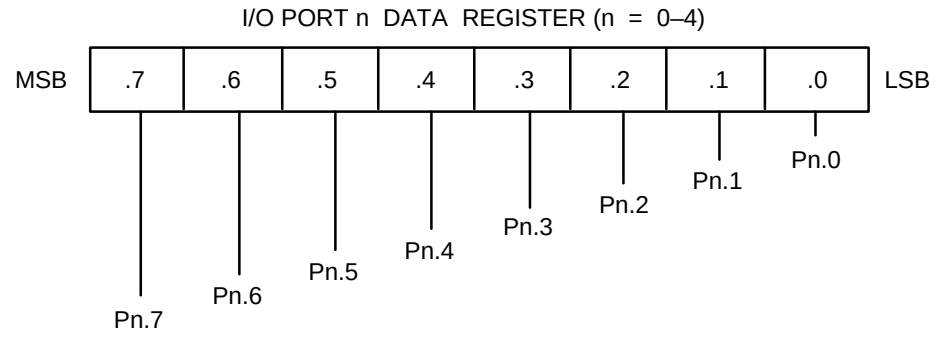
Port	Function Description	Programmability
0	Bit-programmable I/O port for Schmitt trigger input or open-drain output. Port0 can be individually configured as external interrupt inputs. Pull-up resistors are assignable by software.	Bit
1	Bit-programmable I/O port for Schmitt trigger input or open-drain output. Pull-up resistors are assignable by software.	Bit
2	Bit-programmable I/O port for Schmitt trigger input or open-drain output. Port2 can be individually configured as external interrupt inputs. Pull-up resistors are assignable by software.	Bit
3	Bit-programmable I/O port for Schmitt trigger input, open-drain or push-pull output. P3.3 can be used to system clock output (CLO) pin.	Bit
4	Bit-programmable I/O port for Schmitt trigger input or open-drain output or push-pull output. Port4 can be individually configured as external interrupt inputs. In output mode, pull-up resistors are assignable by software. But in input mode, pull-up resistors are fixed.	Bit
D+/PS2 D-/PS2 (PS2 mode Only)	Bit-programmable I/O port for Schmitt trigger input or open-drain output or push-pull output. This port individually configured as external interrupt inputs. In output mode, pull-up resistors are assignable by software. But in input mode, pull-up resistors are fixed.	Bit

PORT DATA REGISTERS

Table 9-2 gives you an overview of the port data register names, locations and addressing characteristics. Data registers for ports 0-4 have the structure shown in Figure 9-1.

Table 9-2. Port Data Register Summary

Register Name	Mnemonic	Decimal	Hex	R/W
Port 0 data register	P0	224	E0H	R/W
Port 1 data register	P1	225	E1H	R/W
Port 2 data register	P2	226	E2H	R/W
Port 3 data register	P3	227	E3H	R/W
Port 4 data register	P4	228	E4H	R/W



NOTE: Because only the four lower-nibble pins of port 3 and port 4 are mapped, data register bits P3.4-P3.7 and P4.4-P4.7 are not used.

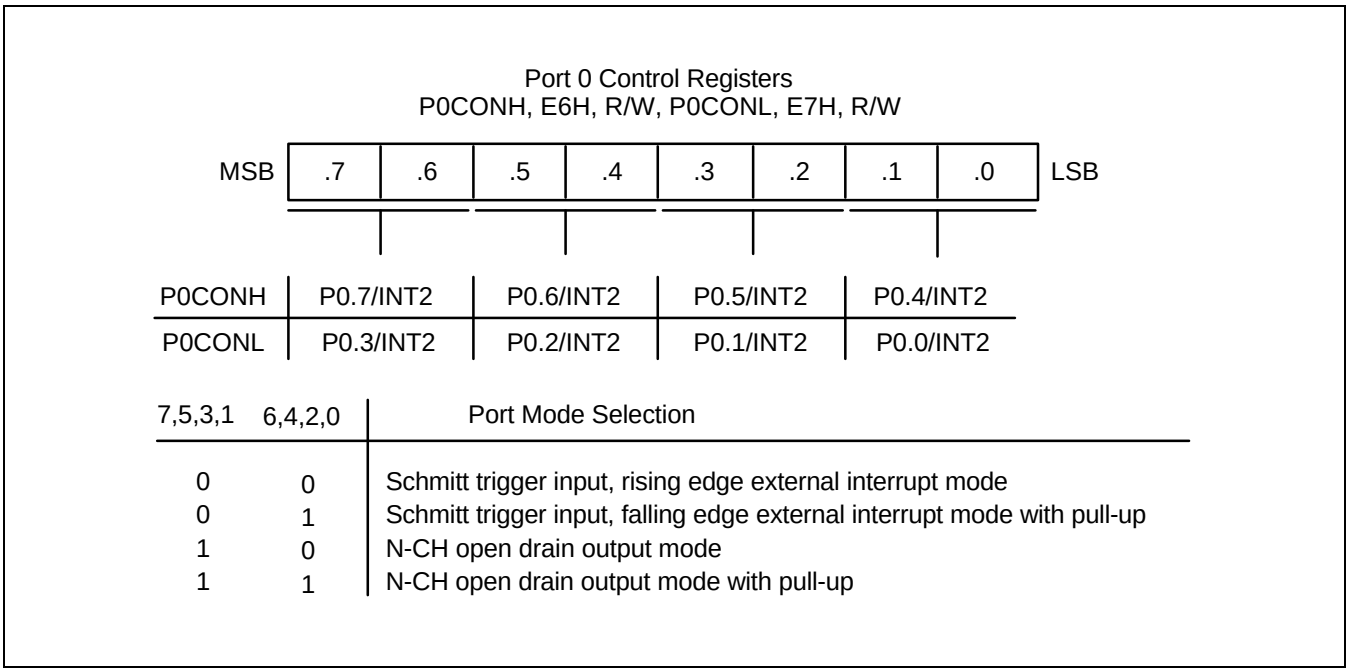
Figure 9-1. Port Data Register Format

PORT 0 AND PORT 1

Ports 0 bit-programmable, general-purpose, I/O ports. You can select Schmitt trigger input mode, N-CH open drain output mode.

You can access ports 0 and 1 directly by writing or reading the corresponding port data registers — P0 (E0H) and P1 (E1H). A reset clears the port control registers P0CONH, P0CONL, P1CONH and P1CONL to '00H', configuring all port 0 and port 1 pins as Schmitt trigger inputs.

In typical keyboard controller applications, the sixteen port 0 and port 1 pins can be used to check pressed key from keyboard matrix by generating keystrobe output signals.



7,5,3,1	6,4,2,0	Port Mode Selection
0	0	Schmitt trigger input, rising edge external interrupt mode
0	1	Schmitt trigger input, falling edge external interrupt mode with pull-up
1	0	N-CH open drain output mode
1	1	N-CH open drain output mode with pull-up

Figure 9-2. Port 0 Control Registers (P0CONH, P0CONL)

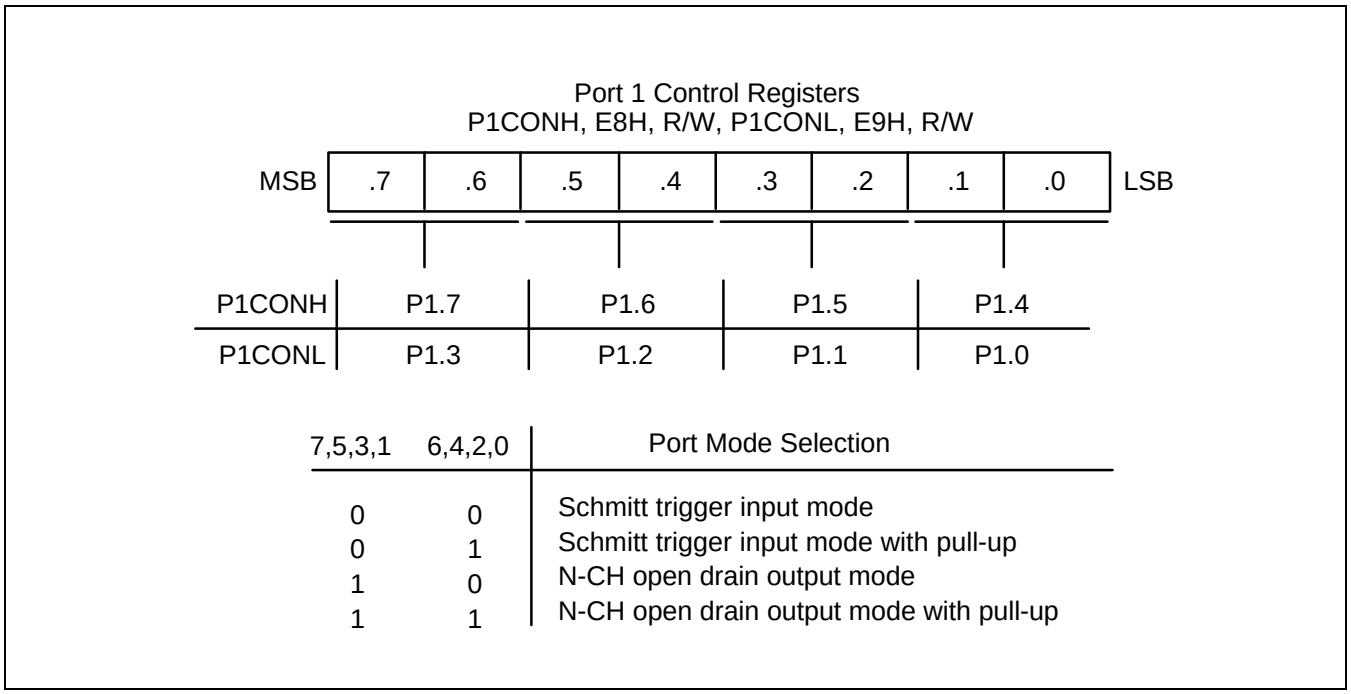


Figure 9-3. Port 1 Control Registers (P1CONH, P1CONL)

PORT 2

Port 2 is an 8-bit I/O port with individually configurable pins. It can be used for general I/O (Schmitt trigger input mode or push-pull output mode). Or, you can use port 2 pins as external interrupt (INT0) inputs. In addition, you can configure a pull-up resistor to individual pins using control register settings. All port 2 pin circuits have noise filters.

In typical keyboard controller applications, the port 2 pins are programmed to receive key input data from the keyboard matrix.

You can address port 2 bits directly by writing or reading the port 2 data register, P2 (E2H). The port 2 high-byte and low-byte control registers, P2CONH and P2CONL, are located at addresses EAH and EBH, respectively.

Two additional registers, are used for interrupt control: P2INT (ECH) and P2PND (EDH). By setting bits in the port 2 interrupt enable register P2INT, you can configure specific port 2 pins to generate interrupt requests when rising or falling signal edges are detected. The application program polls the port 2 interrupt pending register, P2PND, to detect interrupt requests. When an interrupt request is acknowledged, the corresponding pending bit must be cleared by the interrupt service routine.

In case of keyboard applications, the port 2 pins can be used to read key value from key matrix.

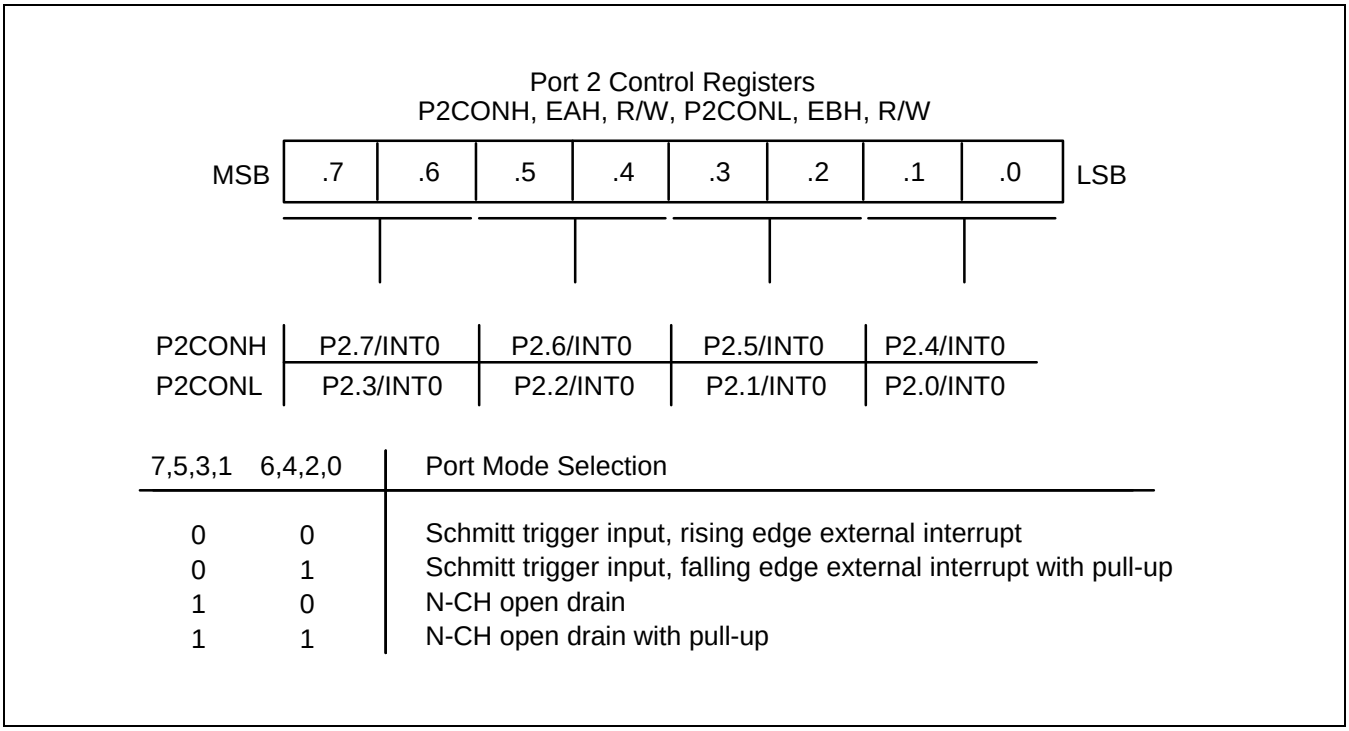


Figure 9-4. Port 2 Control Registers (P2CONH, P2CONL)

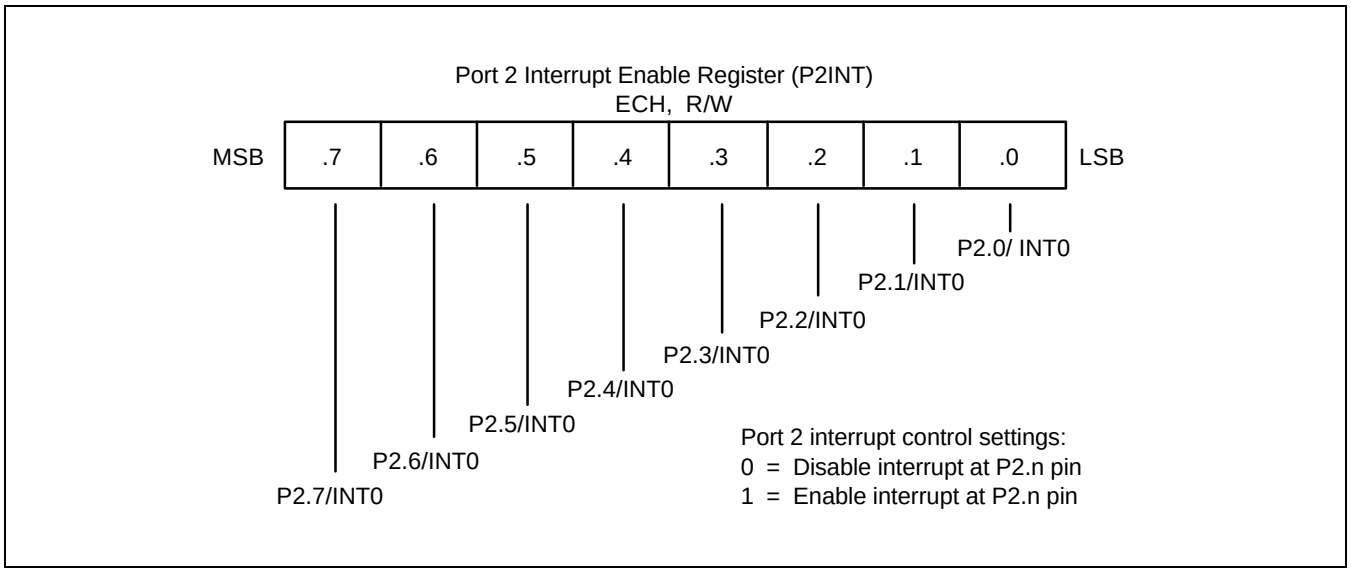


Figure 9-5. Port 2 Interrupt Enable Register (P2INT)

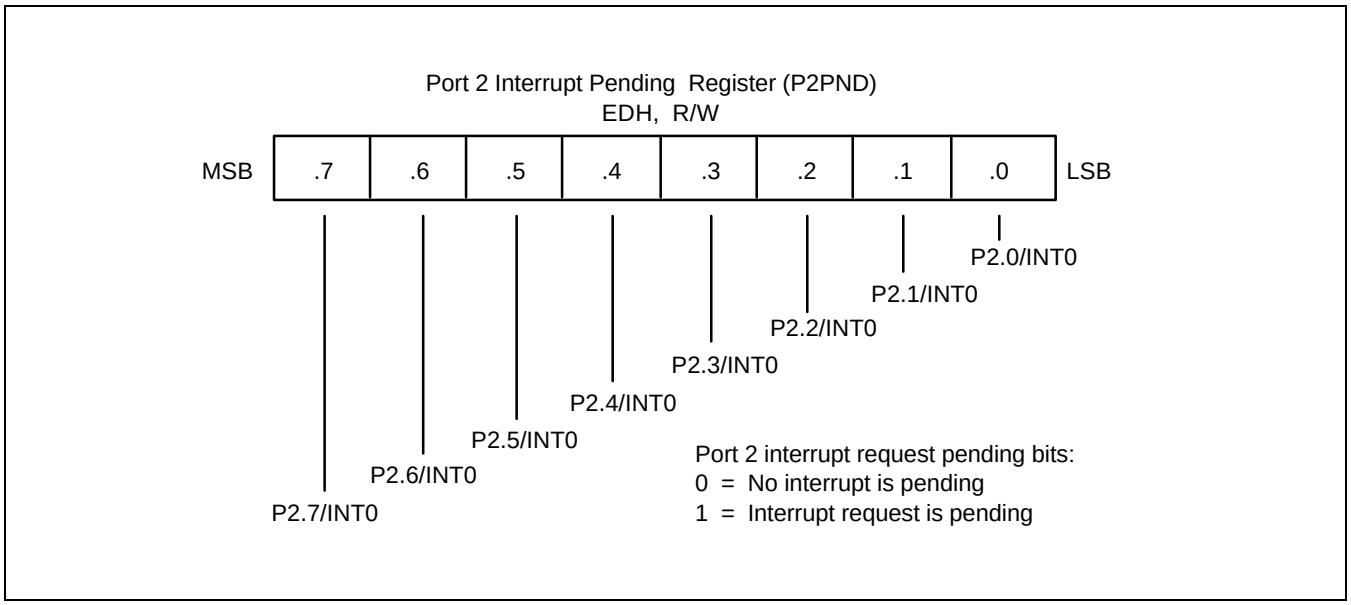
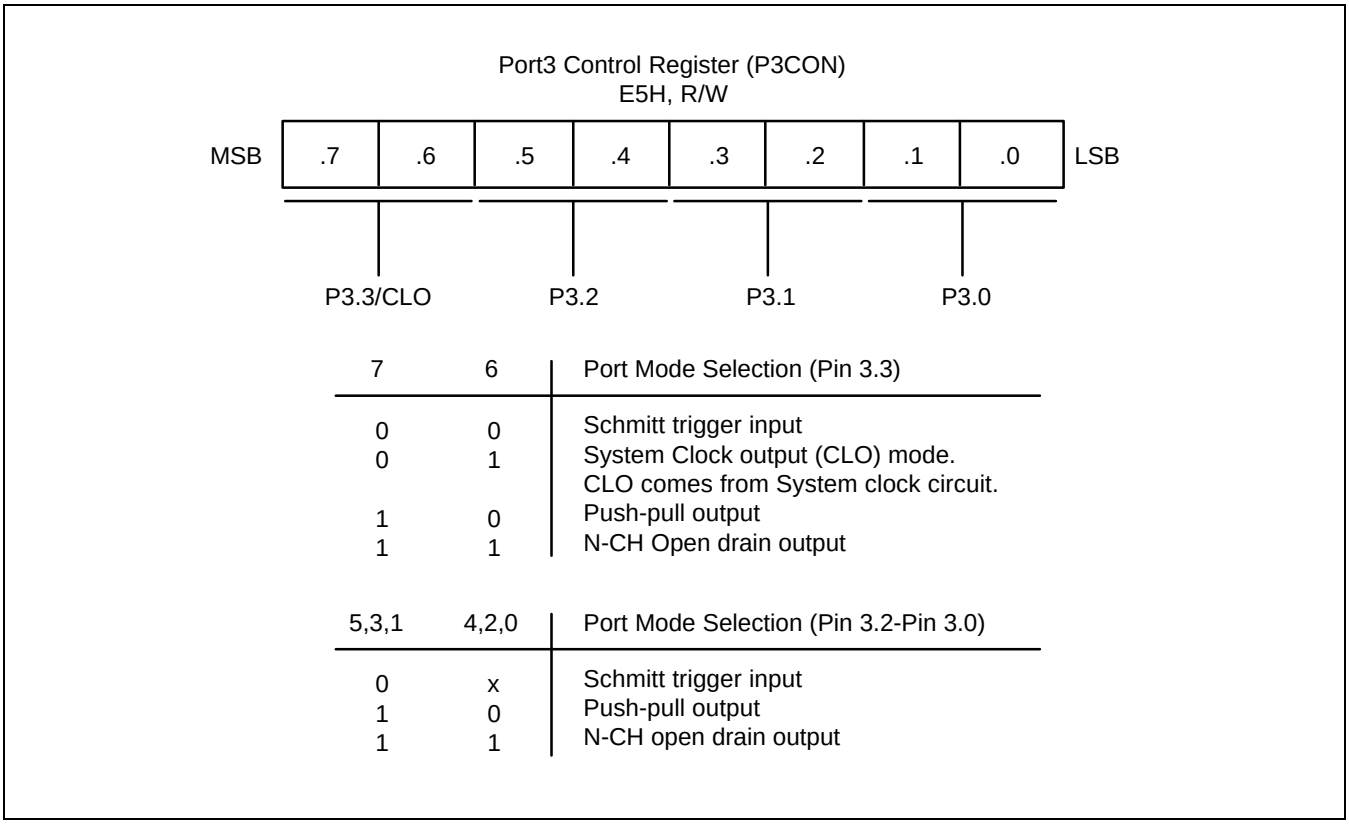


Figure 9-6. Port 2 Interrupt Pending Register (P2PND)

PORT 3

Port 3 is a 4-bit, bit-configurable, general I/O port. It is designed for high-current functions such as LED drive.

A reset configures P3.0-P3.3 to Schmitt trigger input mode. Using the P3CON register (E5H), you can alternatively configure the port 3 pins as n-channel, open-drain outputs. P3.3 can be used to system clock output (CLO) port.



7	6	Port Mode Selection (Pin 3.3)
0	0	Schmitt trigger input
0	1	System Clock output (CLO) mode. CLO comes from System clock circuit.
1	0	Push-pull output
1	1	N-CH Open drain output

5,3,1	4,2,0	Port Mode Selection (Pin 3.2-Pin 3.0)
0	x	Schmitt trigger input
1	0	Push-pull output
1	1	N-CH open drain output

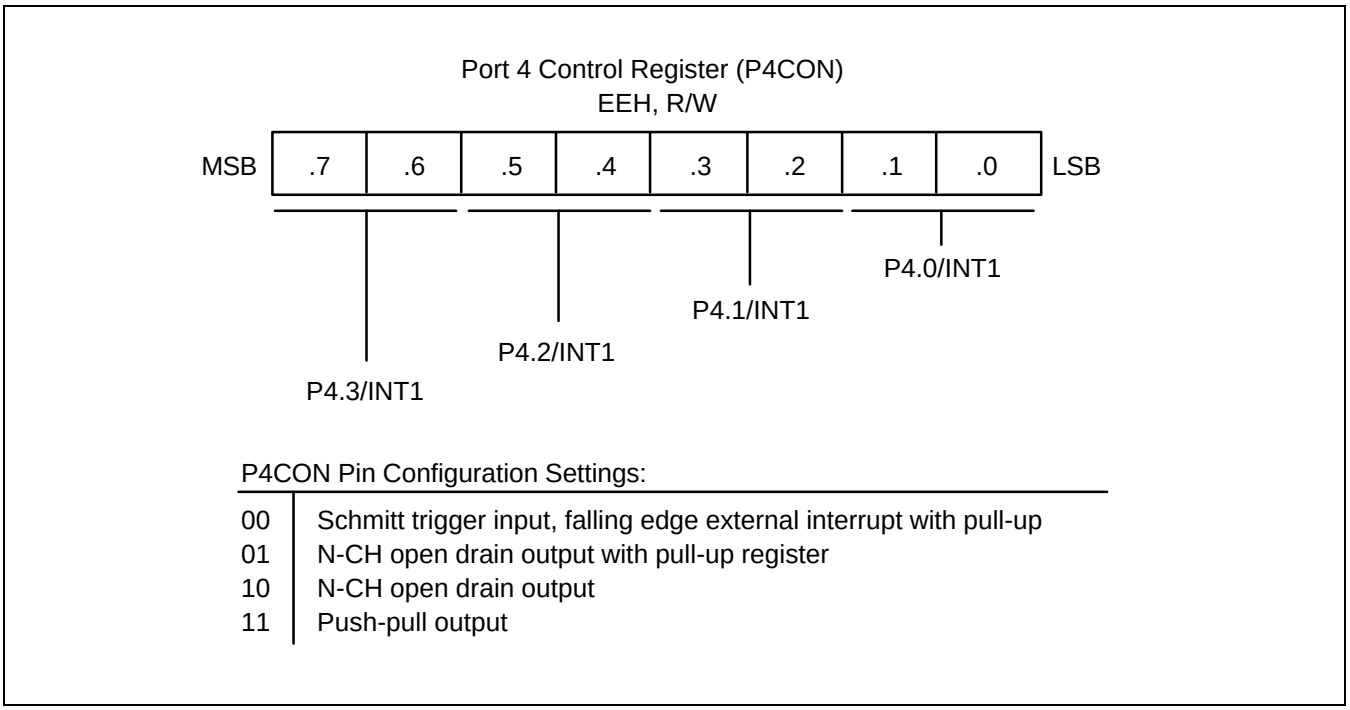
Figure 9-7. Port 3 Control Register (P3CON)

PORT 4

Port 4 is a 4-bit I/O port with individually configurable pins. It can be used for general I/O (Schmitt trigger, N-CH open drain output mode, push-pull output mode). Or, you can use port 4 pins as external interrupt (INT1) inputs. In addition, you can configure a pull-up resistor to individual pins using control register settings. All port 4 pins have noise filters.

A reset configures P4.0-P4.3 to input mode. You address port 4 directly by writing or reading the port 4 data register, P4 (E4H). The port 4 control register, P4CON, is located at EEH.

A additional registers used for interrupt control: P4INTPND (EFH). By setting bits in the port 4 interrupt enable and pending register P4INTPND.7-P4INTPND.4, you can configure specific port 4 pins to generate interrupt requests when falling signal edges are detected. The application program polls the interrupt pending register, P4INTPND.3-P4INTPND.0, to detect interrupt requests. When an interrupt request is acknowledged, the corresponding pending bit must be cleared by the interrupt service routine.



P4CON Pin Configuration Settings:

00	Schmitt trigger input, falling edge external interrupt with pull-up
01	N-CH open drain output with pull-up register
10	N-CH open drain output
11	Push-pull output

Figure 9-8. Port 4 Control Register (P4CON)

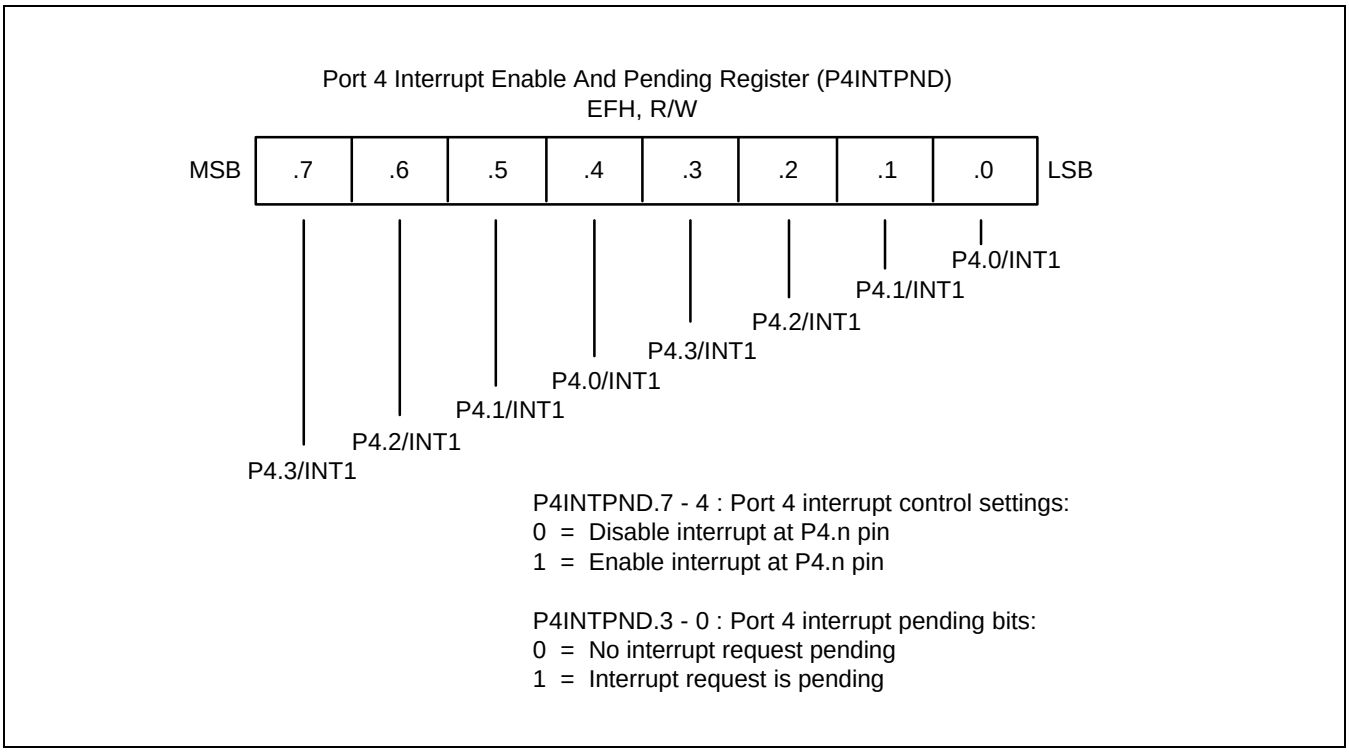
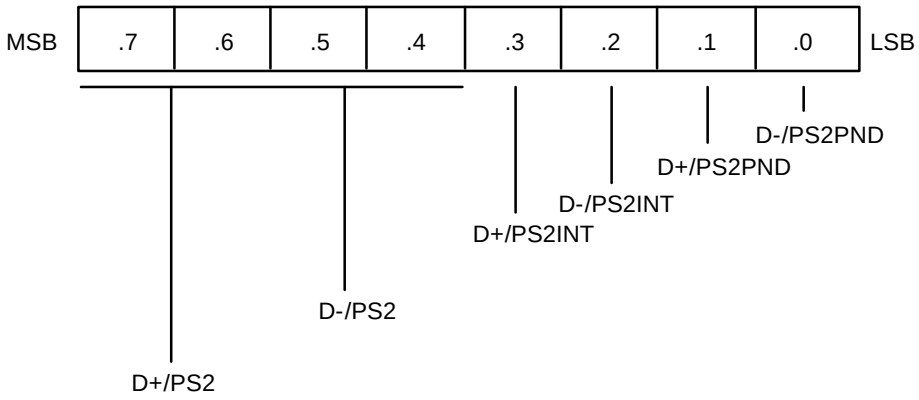


Figure 9-9. Port 4 Interrupt Enable and Pending Register (P4INTPND)

D+/PS2, D-/PS2

PS2 Control And Interrupt And Pending Register
FDH, R/W



PS2CONINT.7-4 Pin Configuration Settings: D+/PS2, D-/PS2

00	Schmitt trigger input, falling edge external interrupt
01	Schmitt trigger input, falling edge external interrupt with pull-up
10	N-CH open drain output
11	N-CH open drain output with pull-up register

PS2CONINT.3-2: Interrupt control setting
0 = Disable interrupt
1 = Enable interrupt

PS2CONINT.3 - 2: Interrupt control setting
0 = No interrupt request pending
1 = Interrupt request is pending

NOTE: Use only PS2MODE

Figure 9-10. PS2CONINT Register (PS2CONINT)

10 BASIC TIMER and TIMER 0

MODULE OVERVIEW

The S3C9644/C9648/P9648 has two default timers: an 8-bit *basic timer* and one 8-bit general-purpose timer/counter. The 8-bit timer/counter is called *timer 0.*

Basic Timer (BT)

You can use the basic timer (BT) in two different ways:

— As a watchdog timer to provide an automatic reset mechanism in the event of a system malfunction.
— To signal the end of the required oscillation stabilization interval after a reset or a Stop mode release.

The functional components of the basic timer block are:

— Clock frequency divider (f_{OSC} divided by 4096, 1024, or 128) with multiplexer
— 8-bit basic timer counter, BTCNT (DDH, read-only)
— Basic timer control register, BTCON (DCH, read/write)

Timer 0

Timer 0 has two operating modes, one of which you select by the appropriate T0CON setting:

— Interval timer mode
— Overflow mode

Timer 0 has the following functional components:

— Clock frequency divider (f_{OSC} divided by 4096, 256, or 8) with multiplexer
— 8-bit counter (T0CNT), 8-bit comparator, and 8-bit reference data register (T0DATA)
— Timer 0 overflow interrupt (T0OVF) and match interrupt (T0INT) generation
— Timer 0 control register, T0CON

BASIC TIMER CONTROL REGISTER (BTCON)

The basic timer control register, BTCON, is used to select the input clock frequency, to clear the basic timer counter and frequency dividers, and to enable or disable the watchdog timer function.

A reset clears BTCON to '00H'. This enables the watchdog function and selects a basic timer clock frequency of $f_{OSC}/4096$. To disable the watchdog function, you must write the signature code '1010B' to the basic timer register control bits BTCON.7-BTCON.4.

The 8-bit basic timer counter, BTCNT, can be cleared at any time during normal operation by writing a "1" to BTCON.1. To clear the frequency dividers for both the basic timer input clock and the timer 0 clock, you write a "1" to BTCON.0.

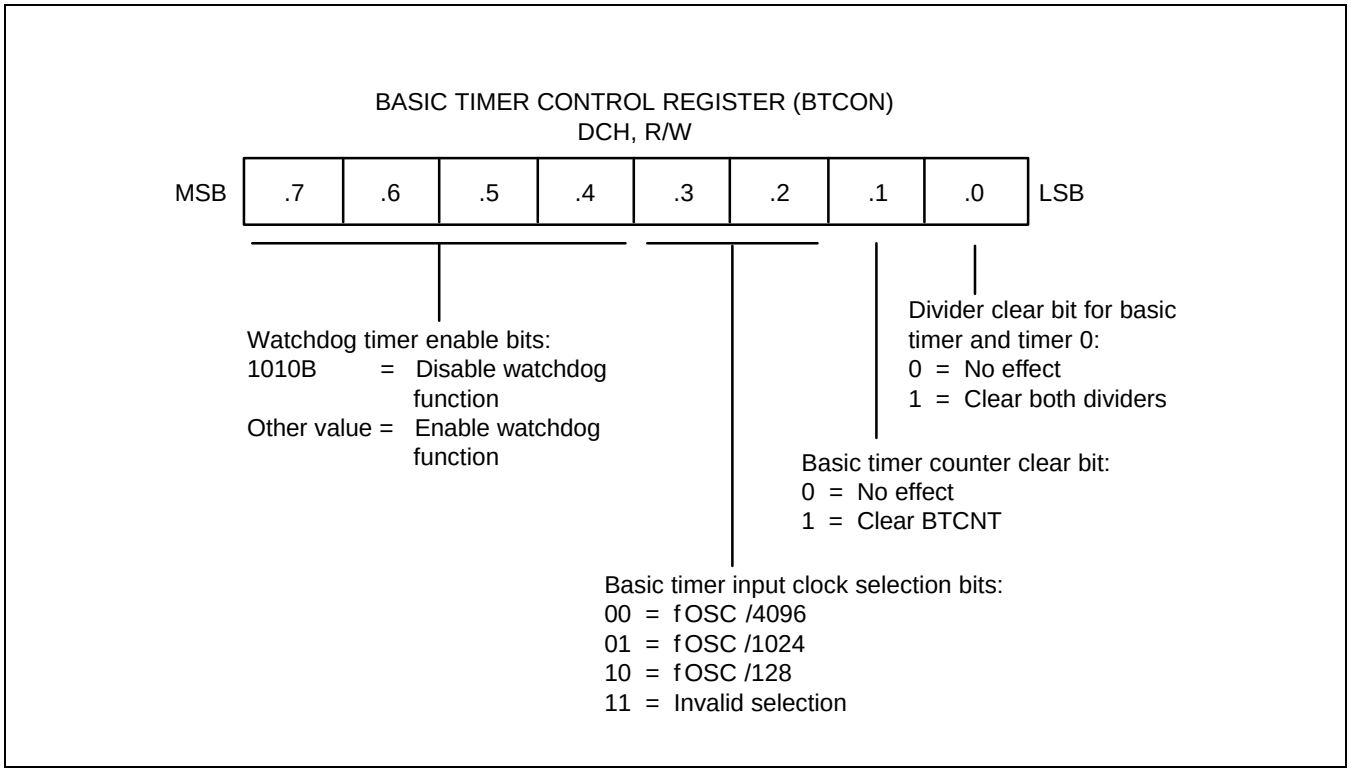


Figure 10-1. Basic Timer Control Register (BTCON)

BASIC TIMER FUNCTION DESCRIPTION

Watchdog Timer Function

You can program the basic timer overflow signal to generate a reset by setting BTCON.7-BTCON.4 to any value other than '1010B' (The '1010B' value disables the watchdog function). A reset clears BTCON to '00H', automatically enabling the watchdog timer function. A reset also selects the CPU clock (as determined by the current CLKCON register setting) divided by 4096 as the BT clock.

A reset whenever a basic timer counter overflow occurs. During normal operation, the application program must prevent the overflow, and the accompanying reset operation, from occurring. To do this, the BTCNT value must be cleared (by writing a "1" to BTCON.1) at regular intervals.

If a system malfunction occurs due to circuit noise or some other error condition, the BT counter clear operation will not be executed and a basic timer overflow will occur, initiating a reset. In other words, during normal operation, the basic timer overflow loop (a bit 7 overflow of the 8-bit basic timer counter, BTCNT) is always broken by a BTCNT clear instruction. If a malfunction does occur, a reset is triggered automatically.

Oscillation Stabilization Interval Timer Function

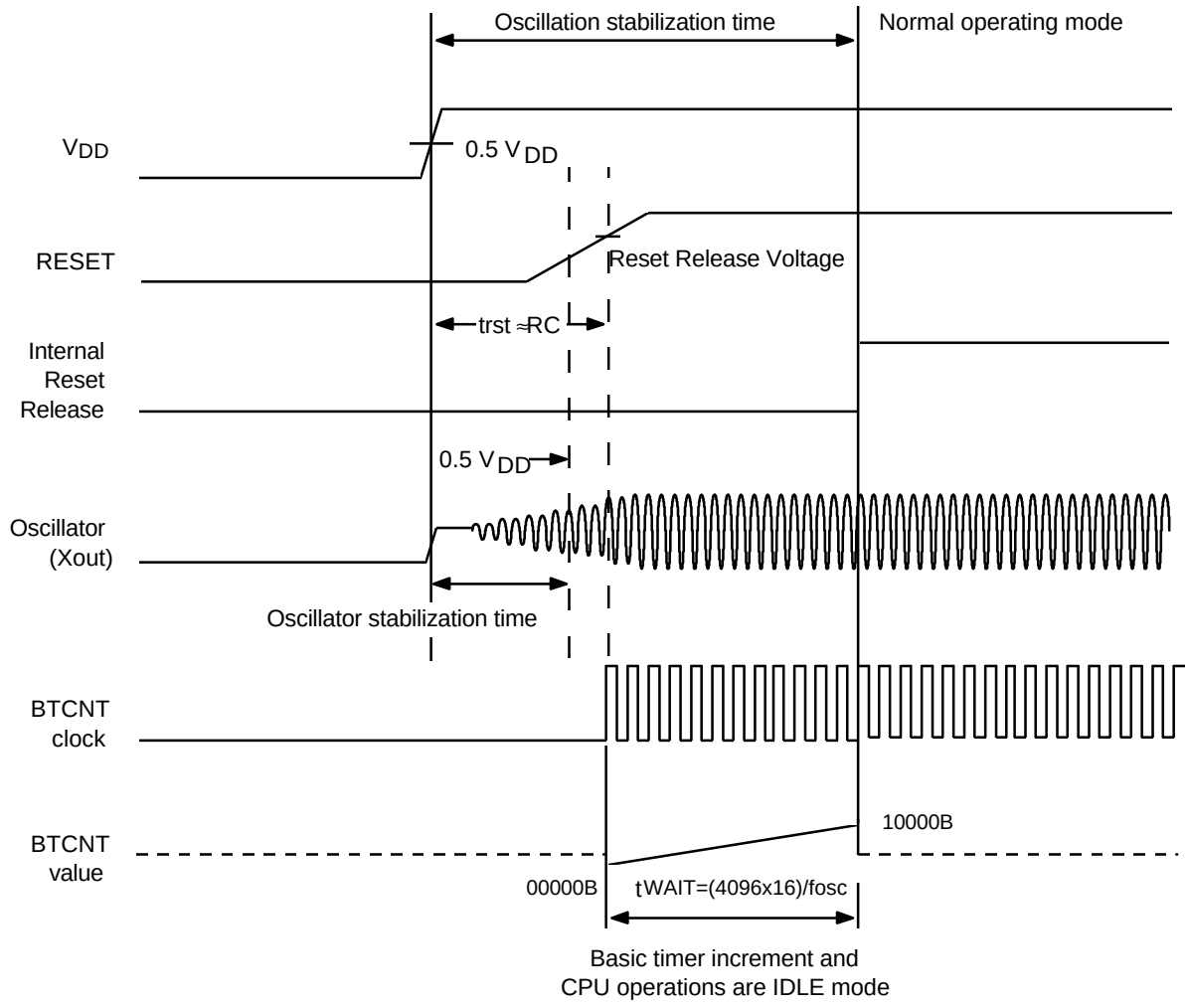
You can also use the basic timer to program a specific oscillation stabilization interval following a reset or when Stop mode has been released by an external interrupt.

In Stop mode, whenever a reset or an external interrupt occurs, the oscillator starts. The BTCNT value then starts increasing at the rate of $f_{OSC}/4096$ (for reset), or at the rate of the preset clock source (for an external interrupt). When BTCNT.4 is set, a signal is generated to indicate that the stabilization interval has elapsed and to gate the clock signal off to the CPU so that it can resume normal operation.

In summary, the following events occur when Stop mode is released:

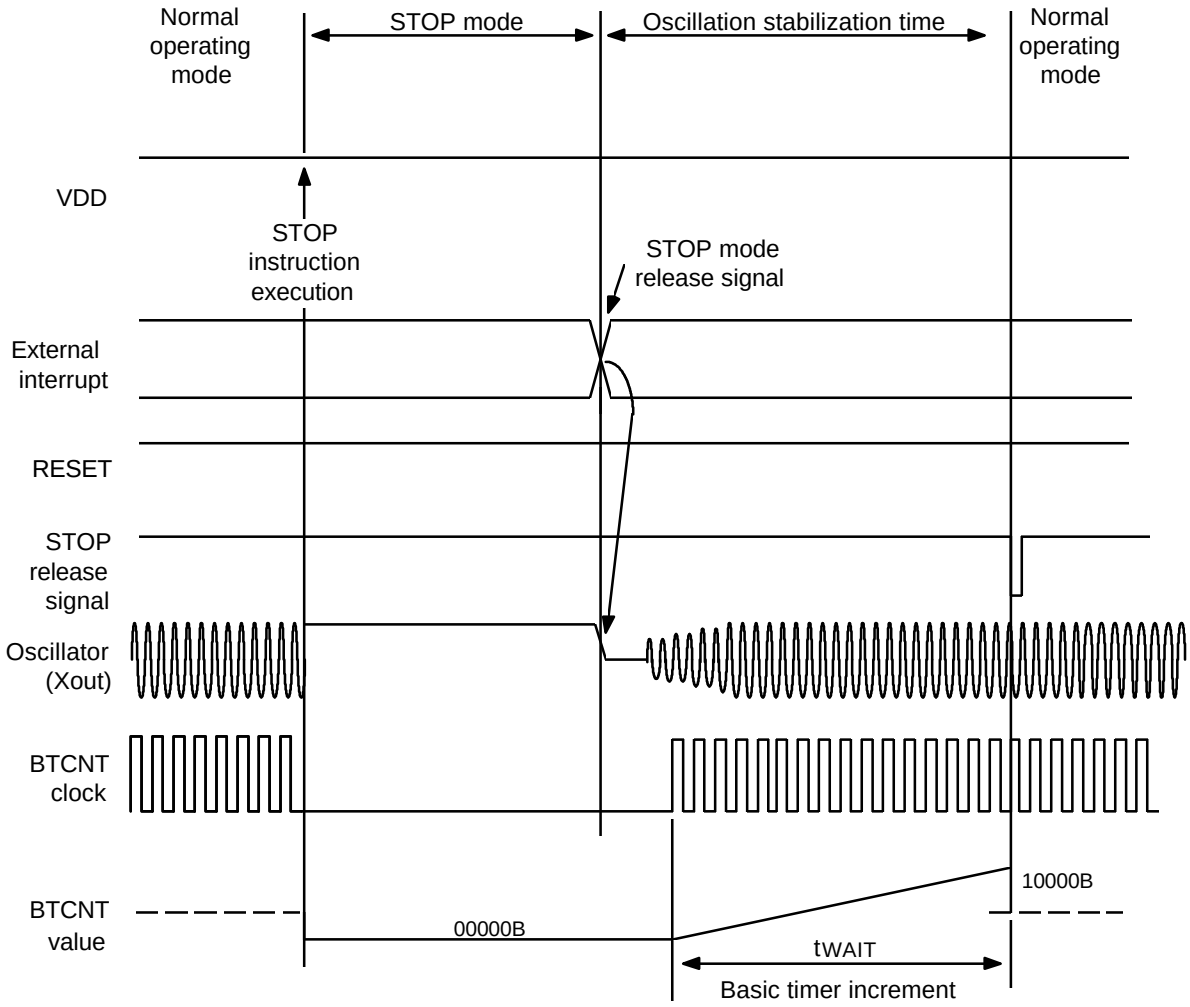
1. During Stop mode, a power-on reset or an external interrupt occurs to trigger the Stop mode release and oscillation starts.
2. If a power-on reset occurred, the basic timer counter will increase at the rate of f_{OSC} /4096. If an external interrupt is used to release Stop mode, the BTCNT value increases at the rate of the preset clock source.
3. Clock oscillation stabilization interval begins and continues until bit 4 of the basic timer counter is set.
4. When a BTCNT.4 is set, normal CPU operation resumes.

Figures 10-2 and 10-3 shows the oscillation stabilization time on RESET and STOP mode release



NOTE: Duration of the oscillator stabilization wait time, tWAIT, when it is released by a Power-on-reset is 4096x16/fosc.
trst ≈RC (R is external resister and C is on chip capacitor)

Figure 10-2. Oscillation Stabilization Time on RESET



NOTE: Duration of the oscillator stabilization wait time, tWAIT, it is released by an interrupt is determined by the setting in basic timer control register, BTCON.

BTCON.3	BTCON.2	tWAIT	tWAIT (When fosc is 6 MHz)
0	0	(4096 x 16) / fosc	10.92 ms
0	1	(1024 x 16) / fosc	2.7 ms
1	0	(128 x 16) / fosc	0.34 ms
1	1	Invalid setting	—

Figure 10-3. Oscillation Stabilization Time on STOP Mode Release

TIMER 0 CONTROL REGISTER (T0CON)

T0CON is located at address D2H, and is read/write addressable.

A reset clears T0CON to '00H'. This sets timer 0 to normal interval match mode, selects an input clock frequency of $f_{OSC}/4096$, and disables the timer 0 overflow interrupt and match interrupt. You can clear the timer 0 counter at any time during normal operation by writing a "1" to T0CON.3.

The timer 0 overflow interrupt can be enabled by writing a "1" to T0CON.2. When a timer 0 overflow interrupt occurs and is serviced by the CPU, the pending condition must be cleared by software by writing a "0" to the timer 0 interrupt pending bit, T0CON.0.

To enable the timer 0 match interrupt, you must write T0CON.1 to "1". To detect an interrupt pending condition, the application program polls T0CON.0. When a "1" is detected, a timer 0 match/ capture interrupt is pending. When the interrupt request has been serviced, the pending condition must be cleared by software by writing a "0" to the timer 0 interrupt pending bit, T0CON.0.

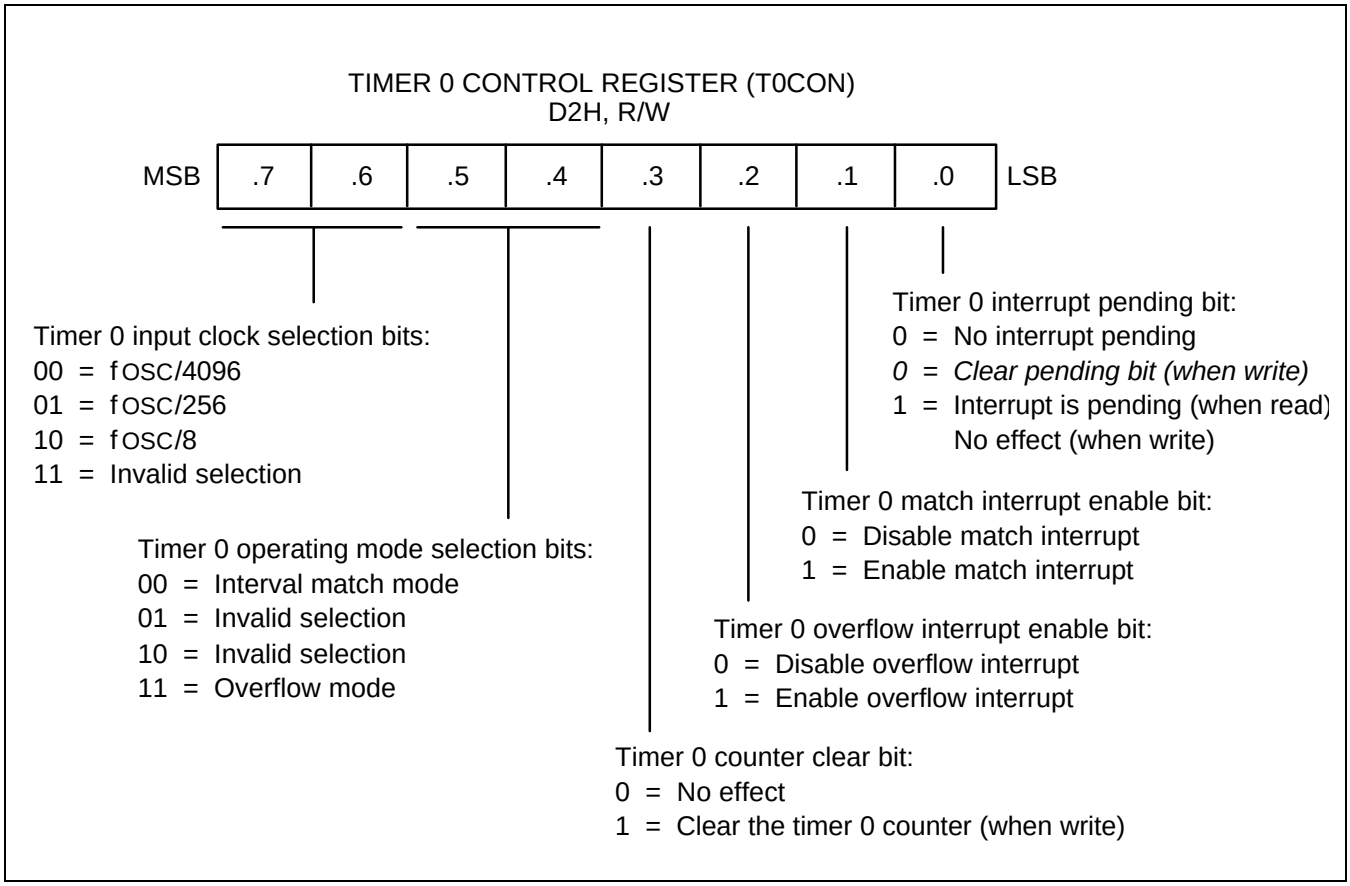


Figure 10-4. Timer 0 Control Register (T0CON)

TIMER 0 FUNCTION DESCRIPTION

Interval Match Mode

In interval match mode, a match signal is generated when the counter value is identical to the value written to the T0 reference data register, T0DATA. The match signal generates a timer 0 match interrupt and then clears the counter. If for example, you write the value '10H' to T0DATA, the counter will increment until it reaches '10H'. At this point, the T0 match interrupt is generated, the counter value is reset and counting resumes.

Overflow Mode

In overflow mode, a overflow signal is generated regardless of the value written to the T0 reference data register when the counter value is overflowed. The overflow signal generates a timer 0 overflow interrupt and then T0 counter is cleared.

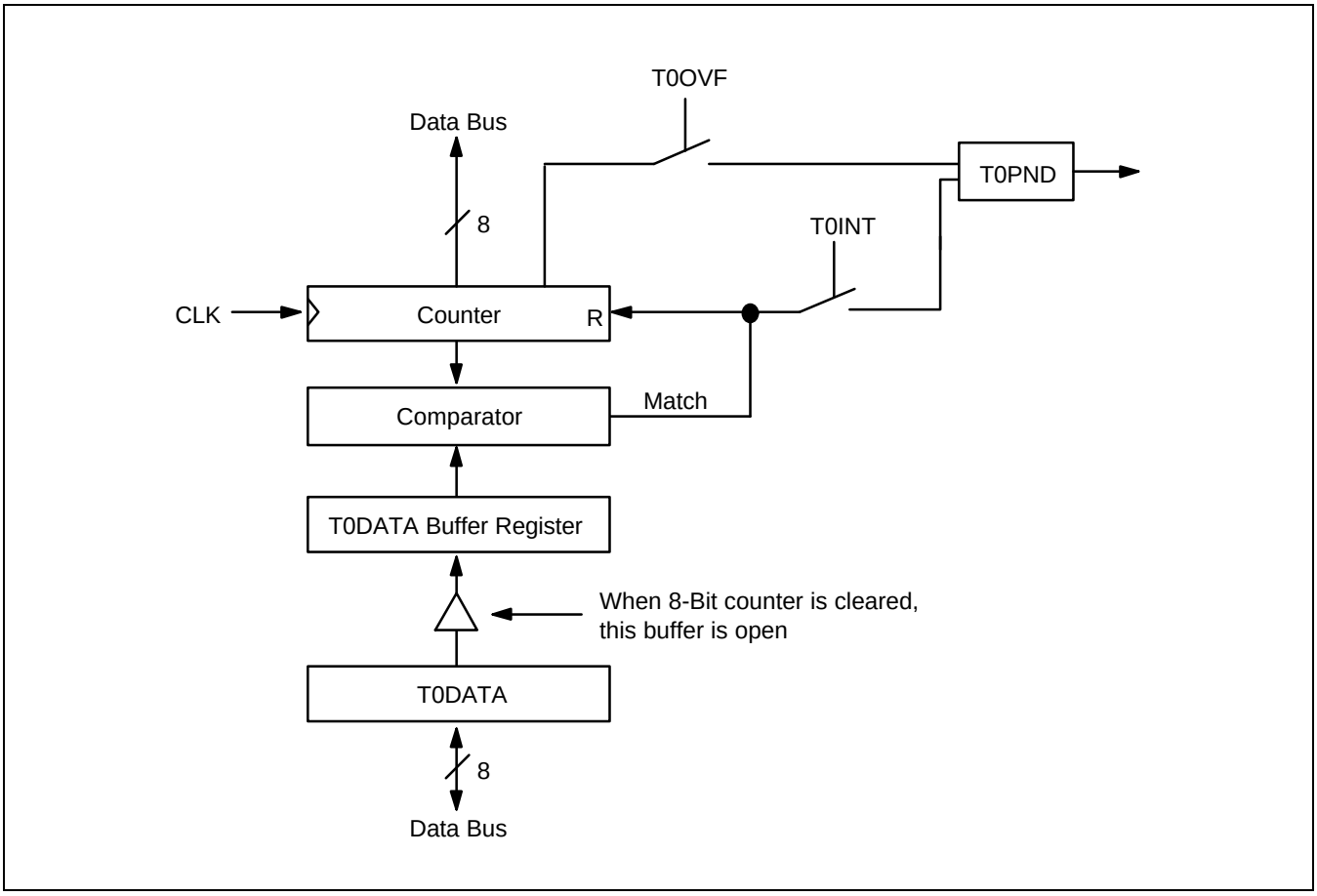


Figure 10-5. Simplified Timer 0 Function Diagram: Interval Timer Mode

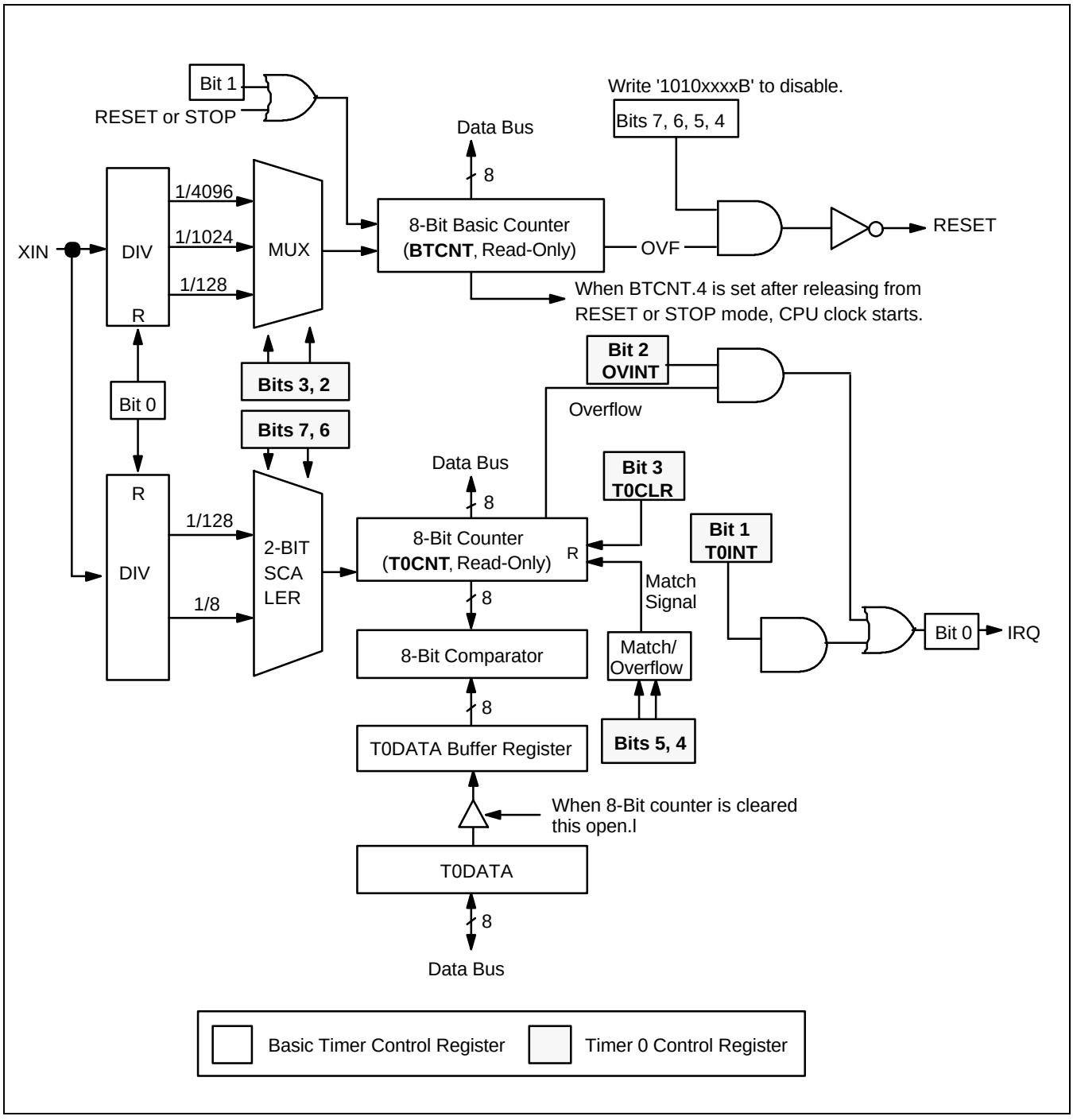


Figure 10-6. Basic Timer and Timer 0 Block Diagram

11 UNIVERSAL SERIAL BUS

OVERVIEW

Universal Serial Bus (USB) is a communication architecture that supports data transfer between a host computer and a wide range of PC peripherals. USB is actually a cable bus in which the peripherals share its bandwidth through a host scheduled token based protocol.

The USB module in S3C9644/C9648/P9648 is designed to serve at a low speed transfer rate (1.5 Mbs) USB device as described in the Universal Serial Bus Specification Revision 1.0. S3C9644/C9648/P9648 can be briefly describe as a microcontroller with SAM 87RI core with an on-chip USB peripheral as can be seen in figure 11-1.

The S3C9644/C9648/P9648 comes equipped with Serial Interface Engine (SIE), which handles the communication protocol of the USB. The S3C9644/C9648/P9648 supports the following control logic: packet decoding/generation, CRC generation/checking, NRZI encoding/decoding, Sync detection, EOP (end of packet) detection and bit stuffing.

S3C9644/C9648/P9648 supports two types of data transfers; control and interrupt. Three endpoints are used in this device; Endpoint 0, Endpoint 1, and Endpoint 2 . Please refer to the USB specification revision 1.0 for detail description of USB.

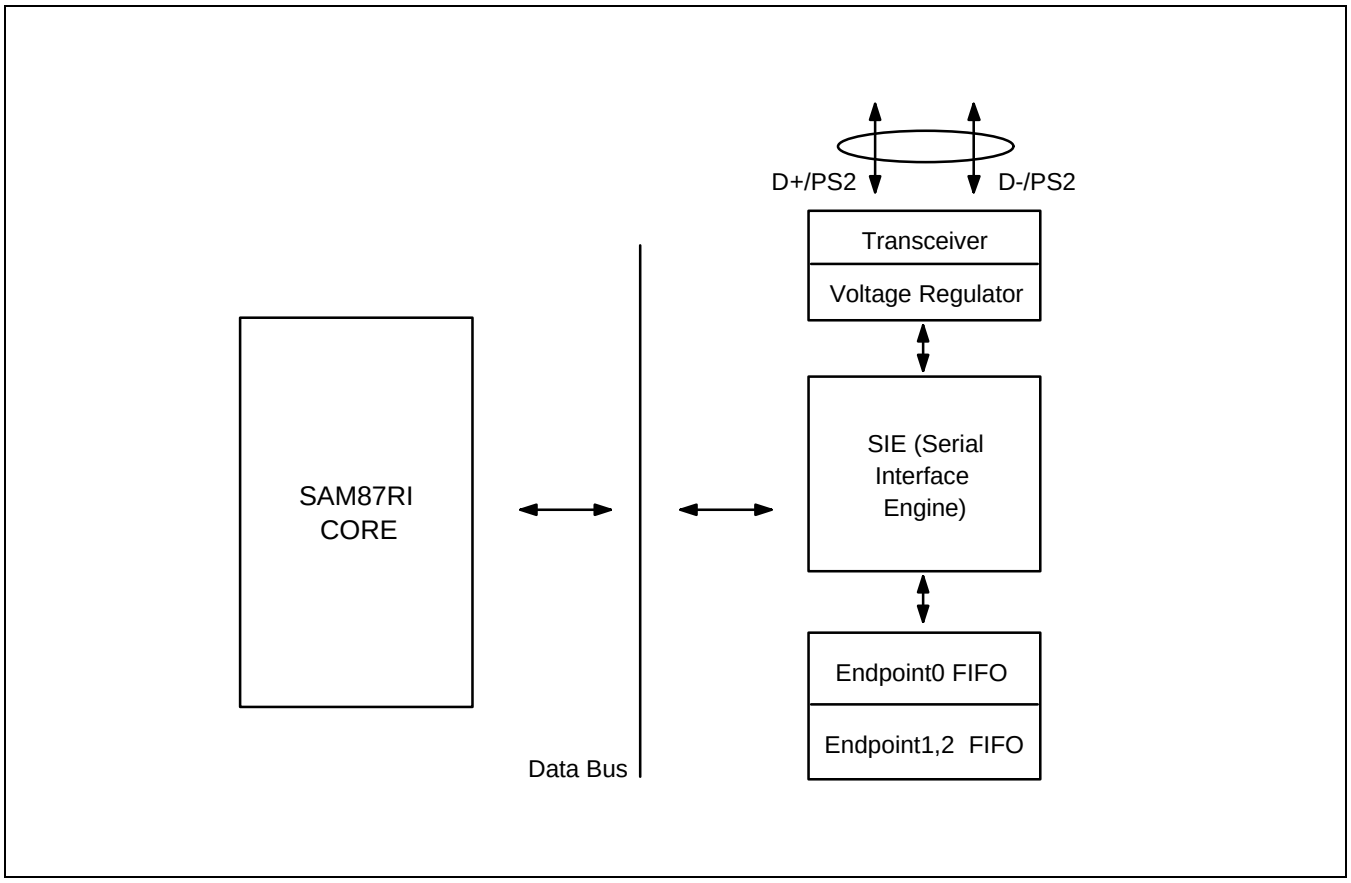


Figure 11-1. USB Peripheral Interface

Serial Bus Interface Engine (SIE)

The Serial Interface Engine interfaces to the USB serial data and handles, deserialization/serialization of data, NRZI encoding/decoding, clock extraction, CRC generation and checking, bit stuffing and other specifications pertaining to the USB protocol such as handling inter packet time out and PID decoding.

Control Logic

The USB control logic manages data movements between the CPU and the transceiver by manipulating the transceiver and the endpoint register. This includes both transmit and receive operations on the USB. The logic contains byte count buffers for transmit operations that load the active transmit endpoint's byte count and use this to determine the number of bytes to transfer. The same buffer is used for receive transactions to count the number of bytes received and transfer that number to the receive endpoint's byte count register at the end of the transaction.

The control logic in S3C9644/C9648/P9648, when transmitting, manages parallel to serial conversion, packet generation, CRC generation, NRZI encoding and bit stuffing.

When receiving, the control logic in S3C9644/C9648/P9648 handles Sync detection, packet decoding, EOP (end of packet) detection, bit stuffing, NRZI decoding, CRC checking and serial to parallel conversion

Bus Protocol

All bus transactions involve the transmission of packets. S3C9644/C9648/P9648 supports three packet types; Token, Data and Handshake. Each transaction starts when the host controller sends a Token Packet to the USB device. The Token packets are generated by the USB host and decoded by the USB device. A Token Packet includes the type description, direction of the transaction, USB device address and the endpoint number.

Data and Handshake packets are both decoded and generated by the USB device. In any transaction, the data is transferred from the host to a device or from a device to the host. The transaction source then sends a Data Packet or indicates that it has no data to transfer. The destination then responds with a Handshake Packet indicating whether the transfer was successful.

Data Transfer Types

USB data transfer occurs between the host software and a specific endpoint on the USB device. An endpoint supports a specific type of data transfer. The S3C9644/C9648/P9648 supports two data transfer endpoints: control and interrupt.

Control transfer configures and assigns an address to the device when detected. Control transfer also supports status transaction, returning status information from device to host.

Interrupt transfer refers to a small, spontaneous data transfer from USB device to host.

Endpoints

Communication flows between the host software and the endpoints on the USB device. Each endpoint on a device has an identifier number. In addition to the endpoint number, each endpoint supports a specific transfer type. S3C9644/C9648/P9648 supports three endpoints: Endpoint 0 supports control transfer, and Endpoint 1 and Endpoint 2 supports interrupt transfer.

USB FUNCTION ADDRESS REGISTER (FADDR)

This register holds the USB address assigned by the host computer. FADDR is located at address F0H and is read/write addressable.

Bit7 Not used

Bit6-0 **FADDR:** MCU updates this register once it decodes a SET_ADDRESS command. MCU must write this register before it clears OUT_PKT_RDY (bit0) and sets DATA_END (bit3) in the EP0CSR register. The function controller use this register's value to decode USB Token packet address. At reset, if the device is not yet configured the value is reset to 0.

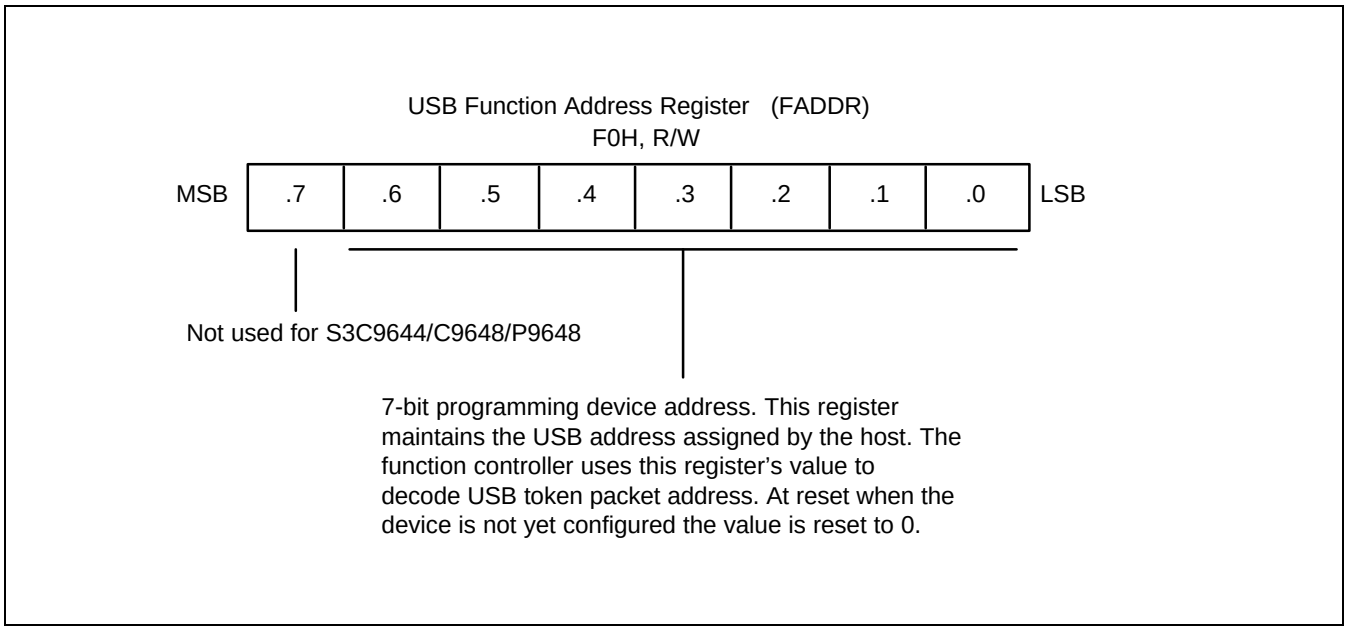


Figure 11-2. USB Function Address Register (FADDR)

CONTROL ENDPOINT CONTROL STATUS REGISTER (EP0CSR)

EP0CSR register controls Endpoint 0 (Control Endpoint), and also holds status bits for Endpoint 0. EP0CSR is located at F1H and is read/write addressable.

Bit7 **CLEAR_SETUP_END:** MCU writes “1” to this bit to clear SETUP_END bit (bit4). This bit is automatically cleared after writing "1" by USB block.

Bit6 **CLEAR_OUT_PKT_RDY:** MCU writes “1” to this bit to clear OUT_PKT_RDY bit (bit0). This bit is automatically cleared after writing "1" by USB block.

Bit5 **SEND_STALL:** MCU writes “1” to this bit to send STALL signal to the Host, at the same time it clears OUT_PKT_RDY (bit0), if it decodes an invalid token. USB issues a STALL Handshake to the current control transfer. This bit gets cleared once a STALL Handshake is issued to the current control transfer.

Bit4 **SETUP_END:** USB sets this bit, when a control transfer ends before DATA_END bit (bit3) is set. MCU clears this bit, by writing a “1” to CLEAR_SETUP_END bit (bit7). When USB sets this bit, an interrupt is generated to MCU. When such condition occurs, USB flushes the FIFO, and invalidates MCU’s access to FIFO.

Bit3 **DATA_END:** MCU sets this bit:

— After loading the last packet of data into the FIFO, and at the same time IN_PKT_RDY bit is set.
— While it clears OUT_PKT_RDY bit after unloading the last packet of data.
— For a zero length data phase, when it clears OUT_PKT_RDY bit, and sets IN_PKT_RDY bit.

Bit2 **SENT_STALL:** USB sets this bit, if a control transaction has ended due to a protocol violation. An interrupt is generated when this bit gets set. MCU clears this bit to end the STALL condition.

Bit1 **IN_PKT_RDY:** MCU sets this bit, after writing a packet of data into Endpoint 0 FIFO. USB clears this bit, once the packet has been successfully sent to the host. An interrupt is generated when USB clears this bit so that MCU can load the next packet. For a zero length data phase, MCU sets IN_PKT_RDY bit and DATA_END bit at the same time.

Bit0 **OUT_PKT_RDY:** USB sets this bit, once a valid token is written to FIFO. An interrupt is generated, when USB sets this bit. MCU clears this bit by writing "1” to CLEAR_OUT_PKT_RDY bit.

In control transfer case, where there is no data phase, MCU after unloading the setup token, sets IN_PKT_RDY, and DATA_END at the same time it clears OUT_PKT_RDY for the setup token.

When SETUP_END bit is set, OUT_PKT_RDY bit may also be set. This happens when the current transfer has ended, and a new control transfer is received before MCU can service the interrupt. In such case, MCU should first clear SETUP_END bit, and then start servicing the new control transfer.

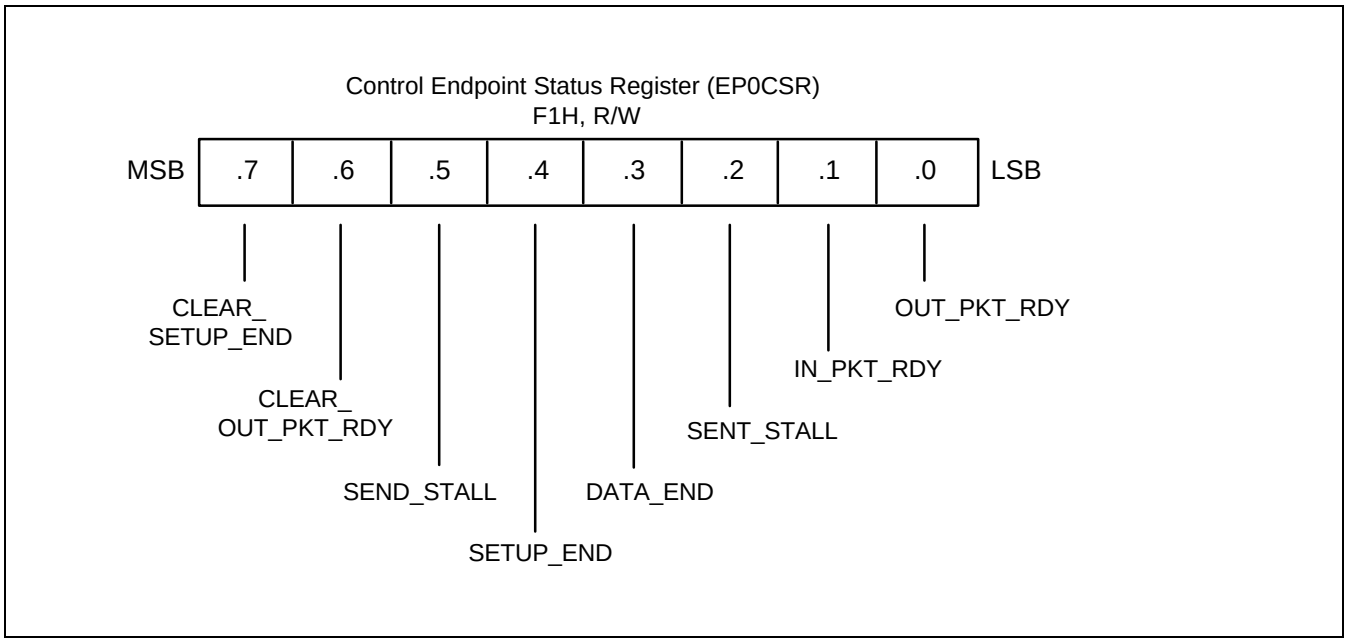


Figure 11-3. Control Endpoint Status Register (EP0CSR)

INTERRUPT ENDPOINT 1 CONTROL STATUS REGISTER (EP1CSR)

EP1CSR is the control register for Endpoint 1, Interrupt Endpoint. This register is located at address F2H and is read/write addressable.

Bit7 **CLEAR_DATA_TOGGLE:** MCU writes “1” to this bit to clear the data toggle sequence bit. When the MCU writes a 1 to this register, the data toggle bit is initialized to DATA0.

Bit6-3 **MAXP:** These bits indicate the maximum packet size for IN endpoint, and needs to be updated by MCU before it sets IN_PKT_RDY. Once set, the contents are valid till MCU re-writes them.

Bit2 **FLUSH_FIFO:** When MCU writes “1” to this register, the FIFO is flushed, and IN_PKT_RDY cleared. The MCU should wait for IN_PKT_RDY to be cleared for the flush to take place.

Bit1 **FORCE_STALL:** MCU writes “1” to this register to issue a STALL Handshake to USB. MCU clears this bit, to end the STALL condition.

Bit0 **IN_PKT_RDY:** MCU sets this bit, after writing a packet of data into Endpoint 1 FIFO. USB clears this bit, once the packet has been successfully sent to the Host. An interrupt is generated when USB clears this bit, so MCU can load the next packet.

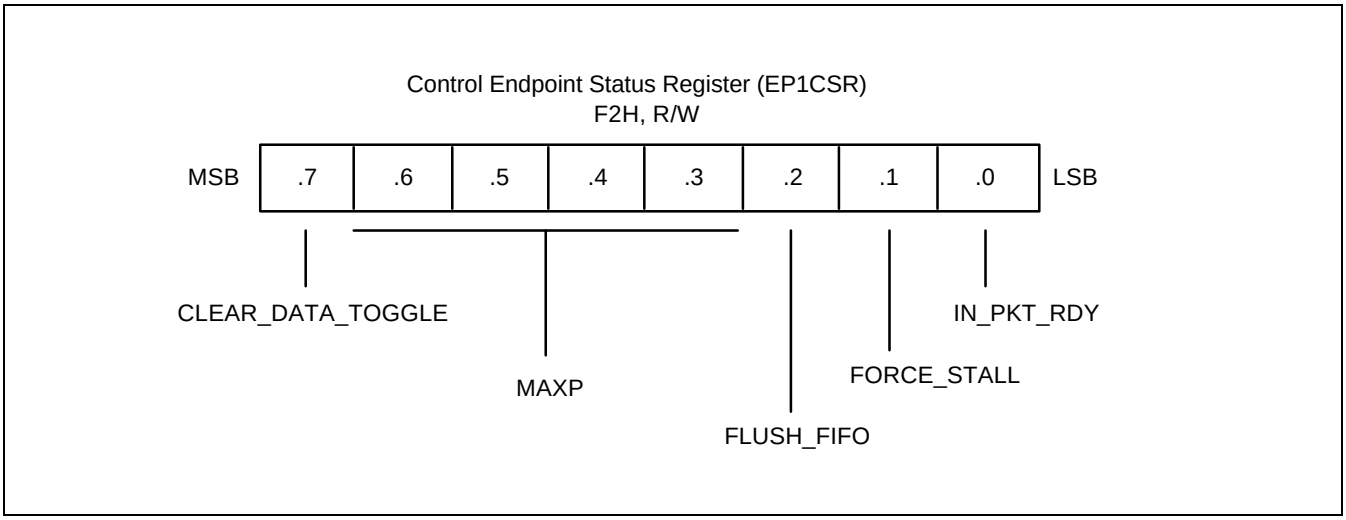


Figure 11-4. 1 Interrupt Endpoint 1 Status Register (EP1CSR)

INTERRUPT ENDPOINT 2 CONTROL STATUS REGISTER (EP2CSR)

EP2CSR is the control register for Endpoint 2, Interrupt Endpoint. This register is located at address F9H and is read/write addressable.

Bit7 **CLEAR_DATA_TOGGLE:** MCU writes “1” to this bit to clear the data toggle sequence bit. When the MCU writes a 1 to this register, the data toggle bit is initialized to DATA0.

Bit6-3 **MAXP:** These bits indicate the maximum packet size for IN endpoint, and needs to be updated by MCU before it sets IN_PKT_RDY. Once set, the contents are valid till MCU re-writes them.

Bit2 **FLUSH_FIFO:** When MCU writes “1” to this register, the FIFO is flushed, and IN_PKT_RDY cleared. The MCU should wait for IN_PKT_RDY to be cleared for the flush to take place.

Bit1 **FORCE_STALL:** MCU writes “1” to this register to issue a STALL Handshake to USB. MCU clears this bit, to end the STALL condition.

Bit0 **IN_PKT_RDY:** MCU sets this bit, after writing a packet of data into Endpoint 1 FIFO. USB clears this bit, once the packet has been successfully sent to the Host. An interrupt is generated when USB clears this bit, so MCU can load the next packet.

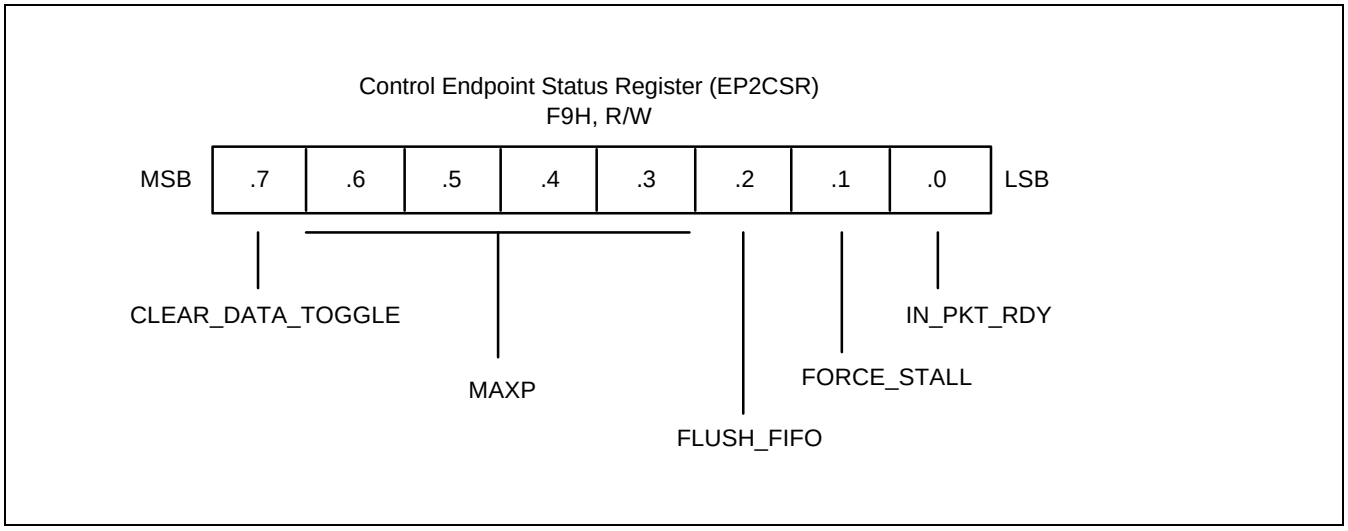


Figure 11-5. 2 Interrupt Endpoint Status Register (EP2CSR)

CONTROL ENDPOINT BYTE COUNT REGISTER (EP0BCNT)

EP0BCNT register has the number of valid bytes in Endpoint 0 FIFO. It is located at address F3H read-only addressable. Once the MCU receives a OUT_PKT_RDY (Bit0 of EP0CSR) for Endpoint 0, then it can read this register to find out the number of bytes to be read from Endpoint 0 FIFO.

CONTROL ENDPOINT FIFO REGISTER (EP0FIFO)

This register is bi-directional, 8-byte depth FIFO used to transfer Control Endpoint data. EP0FIFO is located at address F4H and is read/write addressable.

Initially, the direction of the FIFO, is from the Host to the MCU. After a setup token is received for a control transfer, that is, after MCU unload the setup data packet, and clears OUT_PKT_RDY, the direction of FIFO is changed automatically by the direction bit of data packet.

INTERRUPT ENDPOINT 1 FIFO REGISTER (EP1FIFO)

EP1FIFO is an uni-direction 8-byte depth FIFO used to transfer data from the MCU to the Host. MCU writes data to this register, and when finished set IN_PKT_RDY. This register is located at address F5H.

INTERRUPT ENDPOINT 2 FIFO REGISTER (EP2FIFO)

EP1FIFO is an uni-direction 8-byte depth FIFO used to transfer data from the MCU to the Host. MCU writes data to this register, and when finished set IN_PKT_RDY. This register is located at address FAH.

USB INTERRUPT PENDING REGISTER (USBPND)

USBPND register has the interrupt bits for endpoints and power management. *This register is cleared once read by MCU.* While any one of the bits is set, an interrupt is generated. USBPND is located at address F6H.

Bit7-4 Not used

Bit4 **ENDPT2_PND:** This bit is set, when Endpoint 2 needs to be serviced.

Bit3 **RESUME_PND:** While in suspend mode, if resume signaling is received this bit gets set.

Bit2 **SUSPEND_PND:** This bit is set, when suspend signaling is received.

Bit1 **ENDPT1_PND:** This bit is set, when Endpoint 1 needs to be serviced.

Bit0 **ENDPT0_PND:** This bit is set, when Endpoint 0 needs to be serviced. It is set under any one of the following conditions:

— OUT_PKT_RDY is set.
— IN_PKT_RDY gets cleared.
— SENT_STALL gets set.
— DATA_END gets cleared.
— SETUP_END gets set.

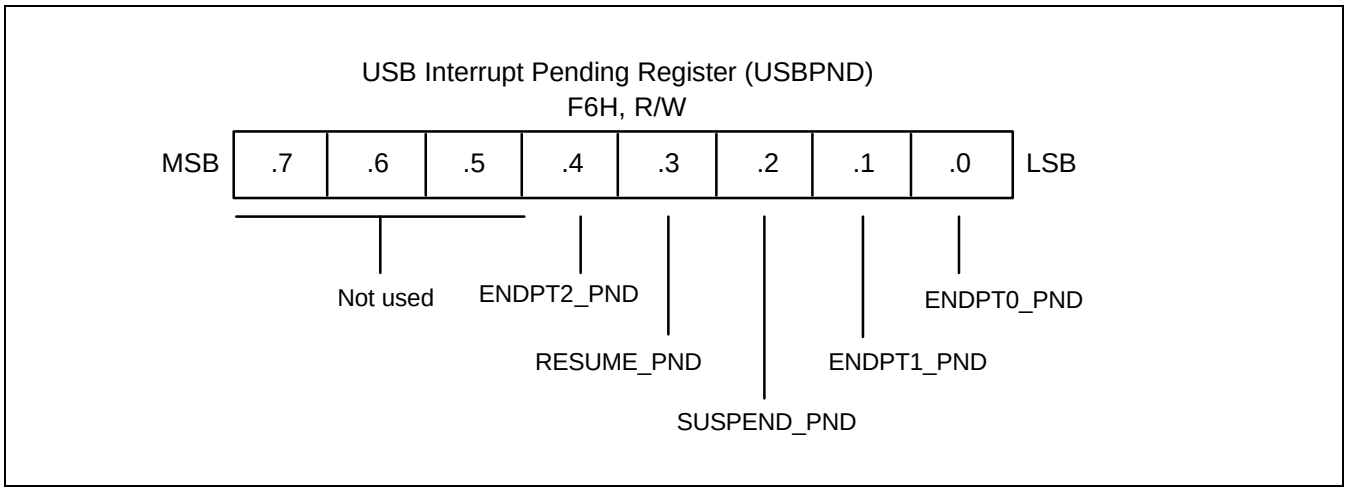


Figure 11-6. USB Interrupt Pending Register (USBPND)

USB INTERRUPT ENABLE REGISTER (USBINT)

USBINT is located at address F7H and is read/write addressable. This register serves as an interrupt mask register. If the corresponding bit = 1 then the respective interrupt is enabled.

By default, all interrupts except suspend interrupt is enabled. Interrupt enables bits for suspend and resume is combined into a single bit (bit 2).

Bit7-3 Not used

Bit3 **ENABLE_ENDPT2_INT:**
1 Enable ENDPOINT 1 INTERRUPT (default)
0 Disable ENDPOINT 1 INTERRUPT

Bit2 **ENABLE_SUSPEND_RESUME_INT:**
1 Enable SUSPEND and RESUME INTERRUPT
0 Disable SUSPEND and RESUME INTERRUPT (default)

Bit1 **ENABLE_ENDPT1_INT:**
1 Enable ENDPOINT 1 INTERRUPT (default)
0 Disable ENDPOINT 1 INTERRUPT

Bit0 **ENABLE_ENDPT0_INT:**
1 Enable ENDPOINT 0 INTERRUPT (default)
0 Disable ENDPOINT 0 INTERRUPT

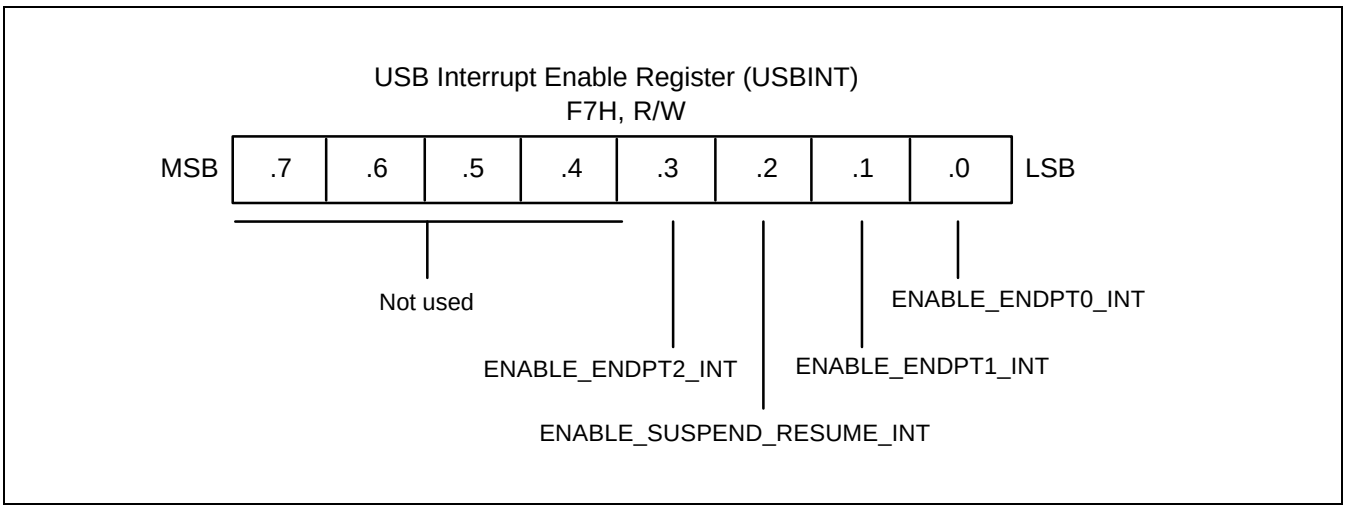


Figure 11-7. USB Interrupt Enable Register (USBINT)

USB POWER MANAGEMENT REGISTER (PWRMGR)

PWRMGR register interacts with the Host's power management system to execute system power events such as SUSPEND or RESUME. This register is located at address F8H and is read/write addressable.

Bit7-2 **RESERVED:** The value read from this bit is zero.

Bit1 **SEND_RESUME:** While in SUSPEND state, if the MCU wants to initiate RESUME, it writes "1" to this register for 10ms (maximum of 15 ms), and clears this register. In SUSPEND mode if this bit reads "1", USB generates RESUME signaling.

Bit0 **SUSPEND_STATE:** Suspend state is set when the MCU sets suspend interrupt. This bit is cleared automatically when:

— MCU writes "0" to SEND_RESUME bit to end the RESUME signaling (after SEND_RESUME is set for 10 ms).
— MCU receives RESUMES signaling from the Host while in SUSPEND mode.

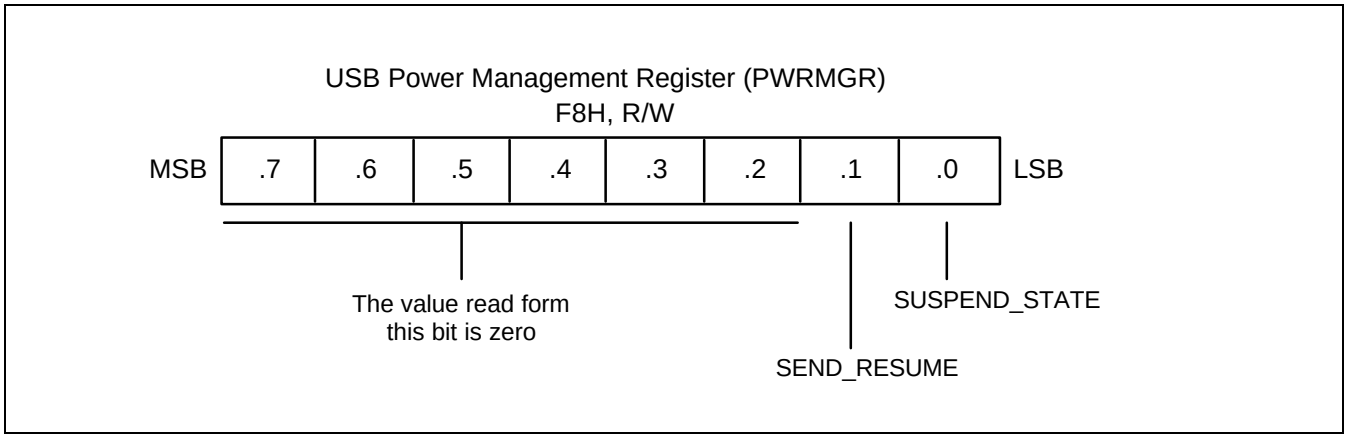


Figure 11-8. USB Power Management Register (PWRMGR)

USB MODE SELECT REGISTER (USBSEL)

USBSEL is located at address FBH and is read/write addressable. This register serves as an USB Mode and PS2 Mode.

Bit7-1 Not used

Bit0 **USBSEL:** 0 = PS2 Mode. (Default)

1 = USB Mode. (This bit is set when the D+/PS2, D-/PS2 port set the D+, D-)

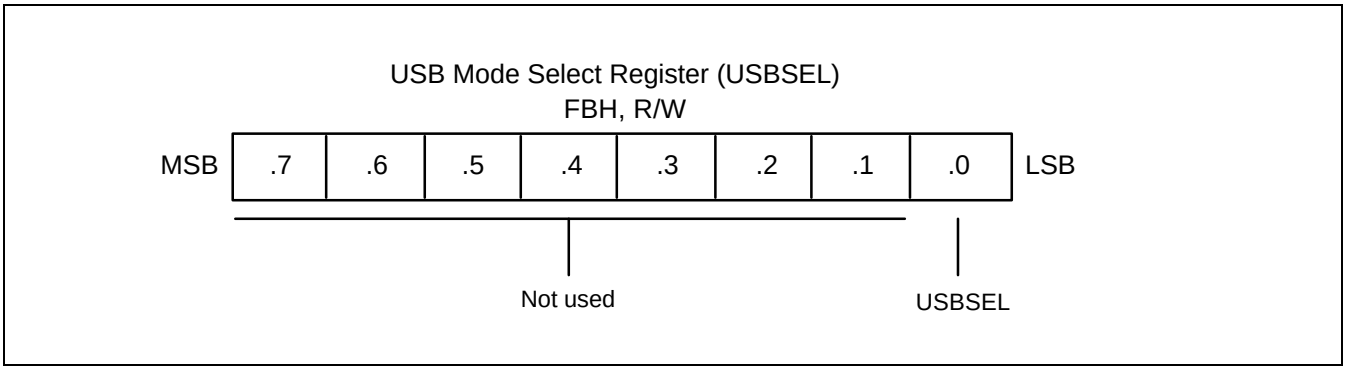


Figure 11-9. USB Mode Select Register (USBSEL)

USB RESET REGISTER (USBRST)

USBRST register receives a reset signal from the Host. This register is located at address FFH and is read/write addressable.

Bit7-1 Not used

Bit0 **USBRST:** This bit is set when the Host issues an USB reset signal.

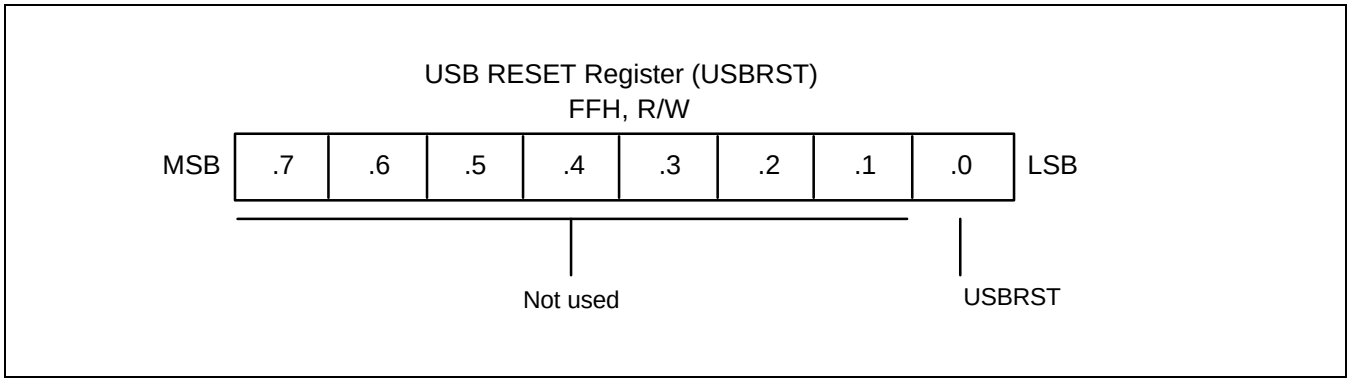


Figure 11-10. USB RESET Register (USBRST)

12 ELECTRICAL DATA

OVERVIEW

In this section, the following S3C9644/C9648/P9648 electrical characteristics are presented in tables and graphs:

— Absolute maximum ratings
— D.C. electrical characteristics
— Input/Output capacitance
— A.C. electrical characteristics
— Input timing for external interrupt (Ports 0, 2 and 4) D+/PS2, D-/PS2 : PS2 Mode Only
— Input timing for RESET
— Oscillator characteristics
— Oscillation stabilization time
— Clock timing measurement points at X_{IN}
— Data retention supply voltage in Stop mode
— Stop mode release timing when initiated by a reset
— Stop mode release timing when initiated by an external interrupt
— Characteristic curves

Table 12-1. Absolute Maximum Ratings

(T_A = 25°C)

Parameter	Symbol	Conditions	Rating	Unit
Supply Voltage	VDD	–	– 0.3 to + 6.5	V
Input Voltage	VIN	All input ports	– 0.3 to VDD + 0.3	V
Output Voltage	VO	All output ports	– 0.3 to VDD + 0.3	V
Output Current High	IOH	One I/O pin active	– 18	mA
		All I/O pins active	– 60	
Output Current Low	IOL	One I/O pin active	+ 30	mA
		Total pin current for ports 3	+ 100	
		Total pin current for ports 0, 1, 2, 4	+ 100	
Operating Temperature	TA	–	– 40 to + 85	°C
Storage Temperature	TSTG	–	– 65 to + 150	°C

Table 12-2. D.C. Electrical Characteristics

(T_A = – 40 °C to + 85 °C, V_{DD} = 4.0 V to 5.25 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Operating Voltage	V_{DD}	f_{OSC} = 6 MHz (instruction clock = 1 MHz)	4.0	5.0	5.25	V
Input High Voltage	V_{IH1}	All input pins except V_{IH2}	0.8 V_{DD}	–	V_{DD}	V
	V_{IH2}	X_{IN}	V_{DD} – 0.5		V_{DD}	
	V_{IH3}	RESET		0.5V_{DD}		
Input Low Voltage	V_{IL1}	All input pins except V_{IL2}	–	–	0.2 V_{DD}	V
	V_{IL2}	X_{IN}			0.4	
	V_{IL2}	RESET		0.5V_{DD}		
Output High Voltage	V_{OH}	I_{OH} = – 200 µA; All output ports except ports 0, 1 and 2, D+, D–	V_{DD} – 1.0	–	–	V
Output Low Voltage	V_{OL}	I_{OL} = 1 mA All output port except D+, D–	–	–	0.4	V
Output Low Current	I_{OL}	V_{OL} = 3V Port 3 only	8	15	23	mA
Input High Leakage Current	I_{LIH1} (3)	V_{IN} = V_{DD} All inputs except I_{LIH2} except D+, D–	–	–	3	µA
	I_{LIH2} (3)	V_{IN} = V_{DD} X_{IN}, X_{OUT}, RESET	–	–	20	µA
Input Low Leakage Current	I_{LIL1} (3)	V_{IN} = 0 V All inputs except I_{LIL2} except D+, D–	–	–	– 3	µA
	I_{LIL2} (3)	V_{IN} = 0 V X_{IN}, X_{OUT}, RESET	–	–	– 20	µA

Table 12-2. D.C. Electrical Characteristics (continued)

(T_A = – 40 °C to + 85 °C, V_{DD} = 4.0 V to 5.25 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Output High Leakage Current	I_{LOH} (1)	$V_{OUT} = V_{DD}$ All I/O pins and output pins except D+, D–	–	–	3	µA
Output Low Leakage Current	ILOL (1)	V_{OUT} = 0 V All I/O pins and output pins except D+, D–	–	–	– 3	µA
Pull-up Resistors	R_{L1}	V_{IN} = 0 V Ports 0, 1, 2, 4.2-3, Reset	25	50	100	kΩ
	R_{L2}	V_{IN} = 0 V; P4.0-1		2.4		
Supply Current (2)	I_{DD1}	Normal operation mode 6 MHz CPU clock	–	5.5	12	mA
	I_{DD2}	Idle mode; 6 MHz oscillator		2.2	5	mA
	I_{DD3}	Stop mode		180	300	µA

NOTES:
1. Except X_{IN} and X_{OUT}.
2. Supply current does not include current drawn through internal pull-up resistors or external output current loads.
3. When USB Mode Only in 4.2 V to 5.25 V, D+ and D– satisfy the USB spec 1.0.

Table 12-3. Input/Output Capacitance

(T_A = – 40 °C to + 85 °C, V_{DD} = 0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Input Capacitance	C_{IN}	f = 1 MHz; Unmeasured pins are connected to V_{SS}	–	–	10	pF
Output Capacitance	C_{OUT}					
I/O Capacitance	C_{IO}					

Table 12-4. A.C. Electrical Characteristics

(T_A = – 40 °C to + 85 °C, V_{DD} = 4.0 V to 5.25 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Interrupt Input High, Low Width	t_{INTH}, t_{INTL}	P0, P2 and P4	–	200	–	ns
RESET Input Low Width	t_{RSL}	RESET	10	–	–	µs

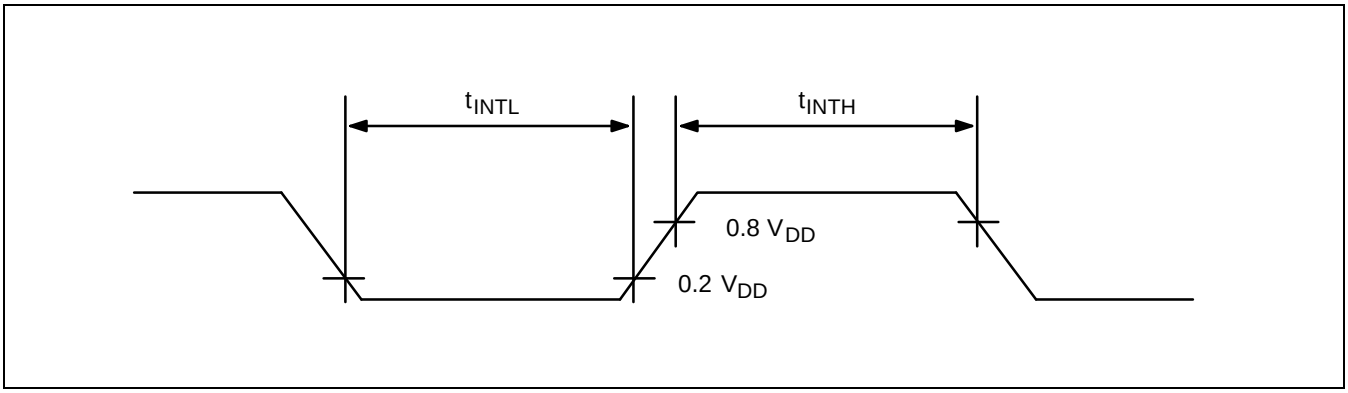


Figure 12-1. Input timing for external interrupt (Ports 0, 2, and 4)

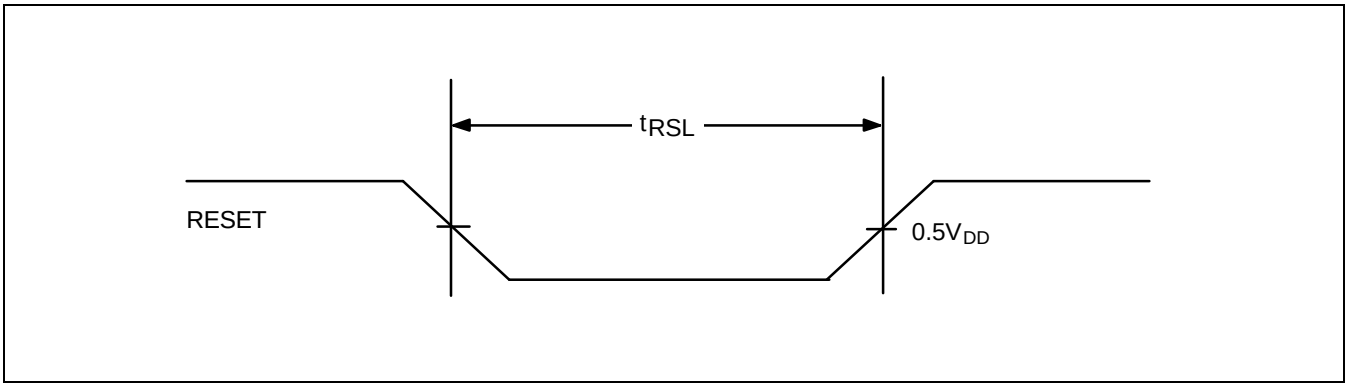


Figure 12-2. Input Timing for RESET

Table 12-5. Oscillator Characteristics

(T_A = – 40°C + 85°C, V_{DD} = 4.0 V to 5.25 V)

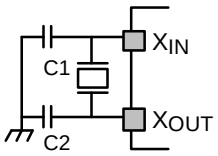
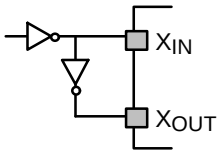
Oscillator	Clock Circuit	Test Condition	Min	Typ	Max	Unit
Main crystal Main ceramic (f_{OSC})		Oscillation frequency	–	6.0	–	MHz
External clock		Oscillation frequency	–	6.0	–	

Table 12-6. Oscillation Stabilization Time

(T_A = – 40°C + 85°C, V_{DD} = 4.0 V to 5.25 V)

Oscillator	Test Condition	Min	Typ	Max	Unit
Main Crystal	f_{OSC} = 6.0 MHz	–	–	10	ms
Main Ceramic	(Oscillation stabilization occurs when V_{DD} is equal to the minimum oscillator voltage range.)				
Oscillator Stabilization Wait Time	t_{WAIT} stop mode release time by a reset	–	$2^{16}/f_{OSC}$	–	
	t_{WAIT} stop mode release time by an interrupt	–	(note)	–	

NOTE: The oscillator stabilization wait time, t_{WAIT}, is determined by the setting in the basic timer control register, BTCON.

Table 12-7. Data Retention Supply Voltage in Stop Mode

(T_A = – 40°C to + 85°C)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Data Retention Supply Voltage	V_{DDDR}	Stop mode	2.0	–	6	V
Data Retention Supply Current	I_{DDDR}	Stop mode; V_{DDDR} = 2.0 V	–	–	300	µA

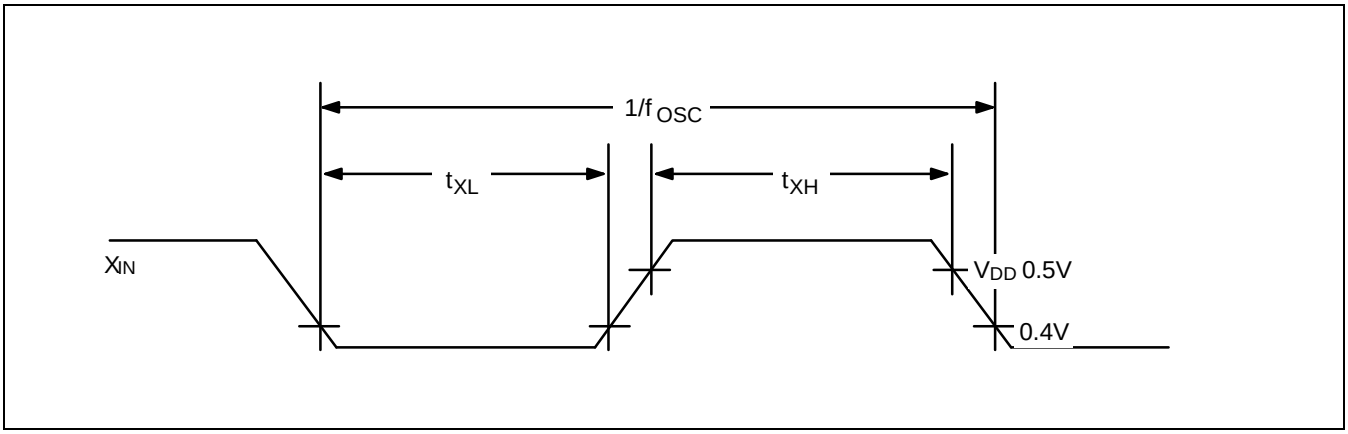


Figure 12-3. Clock Timing Measurement Points at X_{IN}

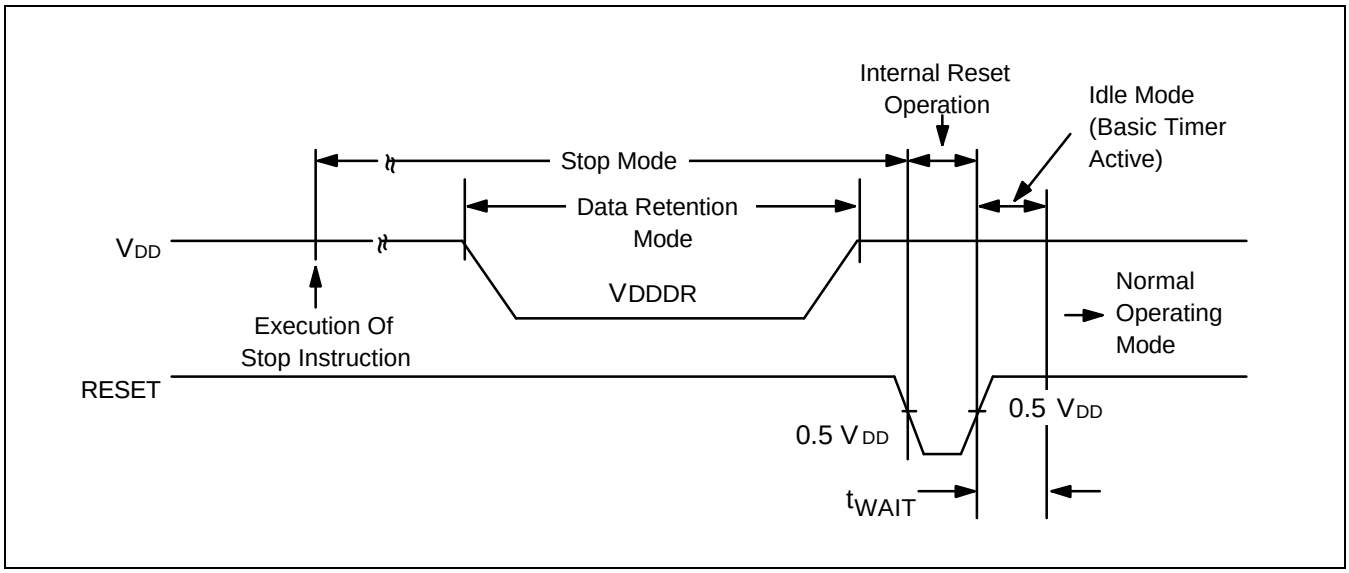


Figure 12-4. Stop Mode Release Timing When Initiated by a Reset

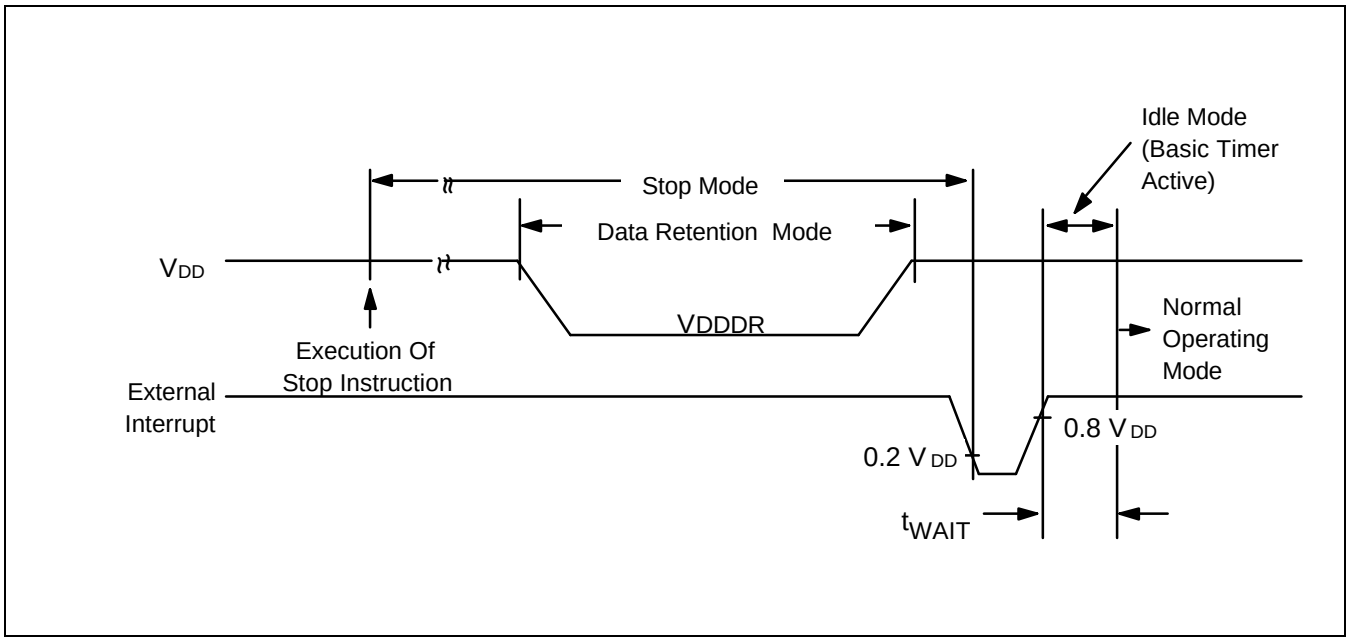


Figure 12-5. Stop Mode Release Timing When Initiated by an External Interrupt

Table 12-8. Low Speed USB Electrical Characteristics

(T_A = – 40°C to + 85°C, Voltage Regulator Output V_{33out} = 2.8 V to 3.5 V, typ 3,3 V)

Parameter	Symbol	Conditions	Min	Max	Unit
Transition Time:					
Rise Time	Tr	CL = 50 pF	75	–	ns
		CL = 350 pF	–	300	
Fall Time	Tf	CL = 50 pF	75	–	
		CL = 350 pF	–	300	
Rise/Fall Time Matching	Trfm	(Tr/Tf) CL = 50 pF	80	120	%
Output Signal Crossover Voltage	Vcrs	CL = 50 pF	1.3	2.0	V
Voltage Regulator Output Voltage	V33OUT	with V33OUT to GND 0.1 µF capacitor	2.8	3.5	V

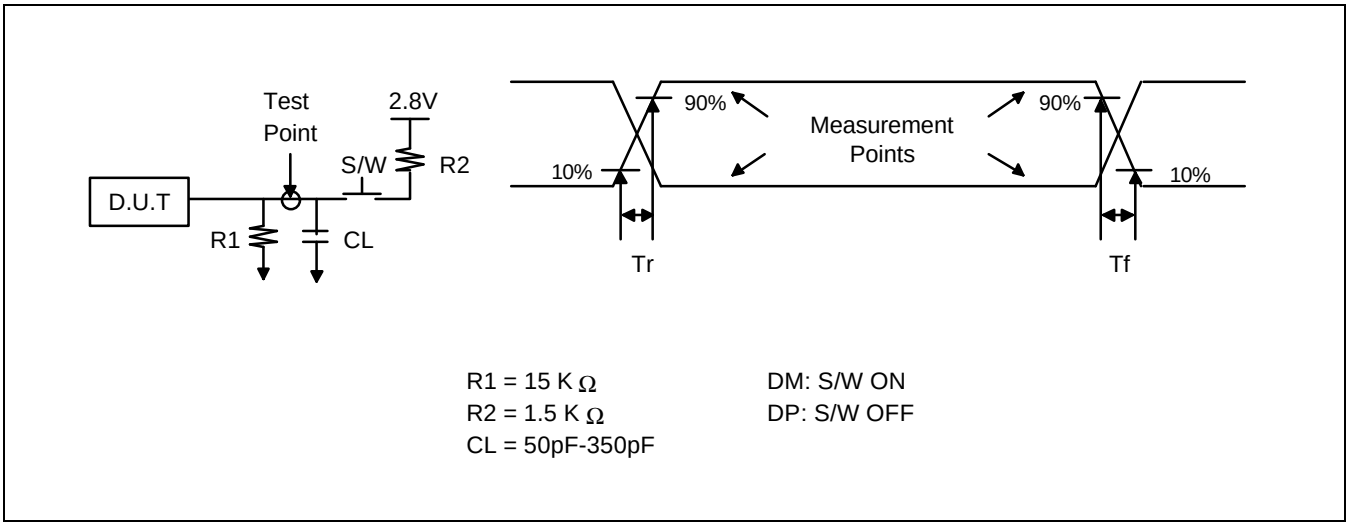


Figure 12-6. USB Data Signal Rise and Fall Time

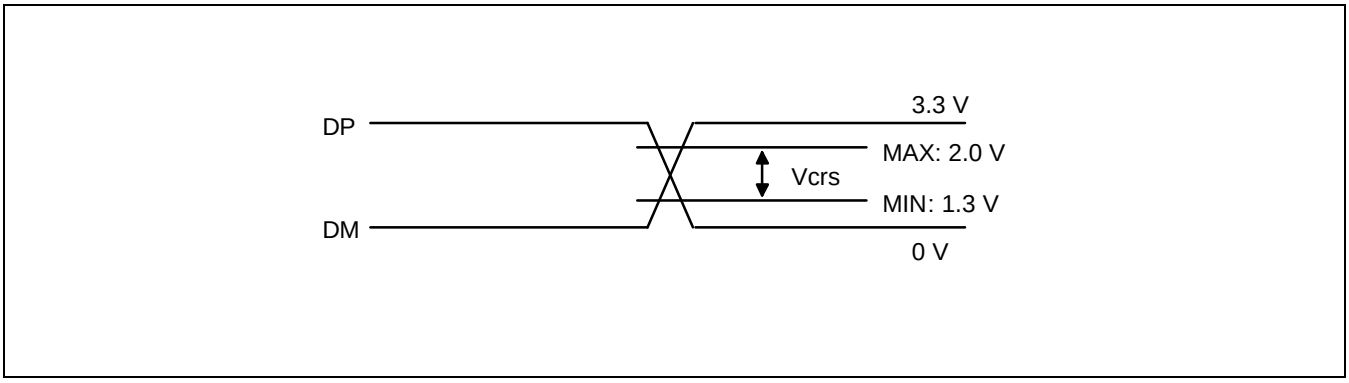


Figure 12-7. USB Output Signal Crossover Point Voltage

13 MECHANICAL DATA

OVERVIEW

The S3C9644/C9648/P9648 is available in a 42-pin SDIP package (Samsung: 42-SDIP-600) and a 44-pin QFP package (44-QFP-1010B). Package dimensions are shown in Figures 13-1 and 13-2.

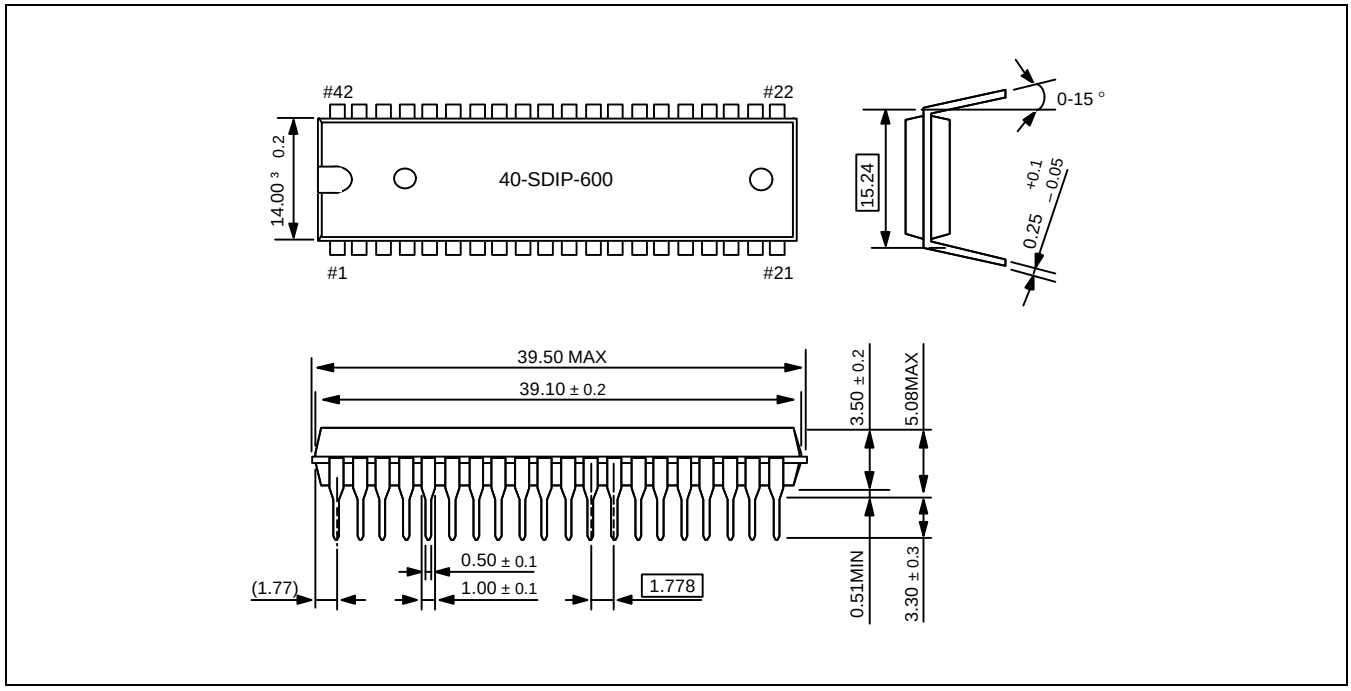
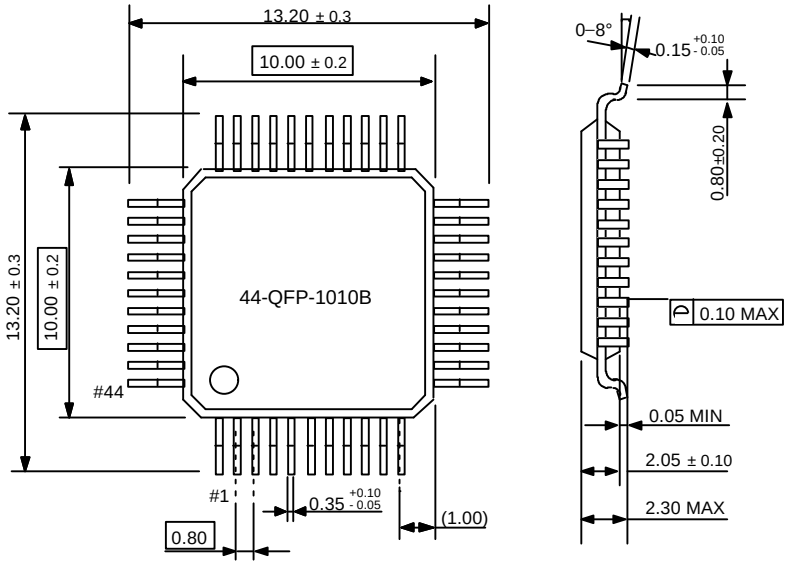


Figure 13-1. 42-Pin SDIP Package Mechanical Data (42-SDIP-600)



NOTE: Dimensions are in millimeters.

Figure 13-2. 44-Pin QFP Package Mechanical Data (44-QFP-1010B)

14 S3P9648 OTP

OVERVIEW

The S3P9648 single-chip CMOS microcontroller is the OTP (One Time Programmable) version of the S3C9644/C9648 microcontroller. It has an on-chip OTP ROM instead of masked ROM. The EPROM is accessed by serial data format.

The S3P9648 is fully compatible with the S3C9644/C9648, both in function and in pin configuration. Because of its simple programming requirements, the S3P9648 is ideal for use as an evaluation chip for the S3C9644/C9648.

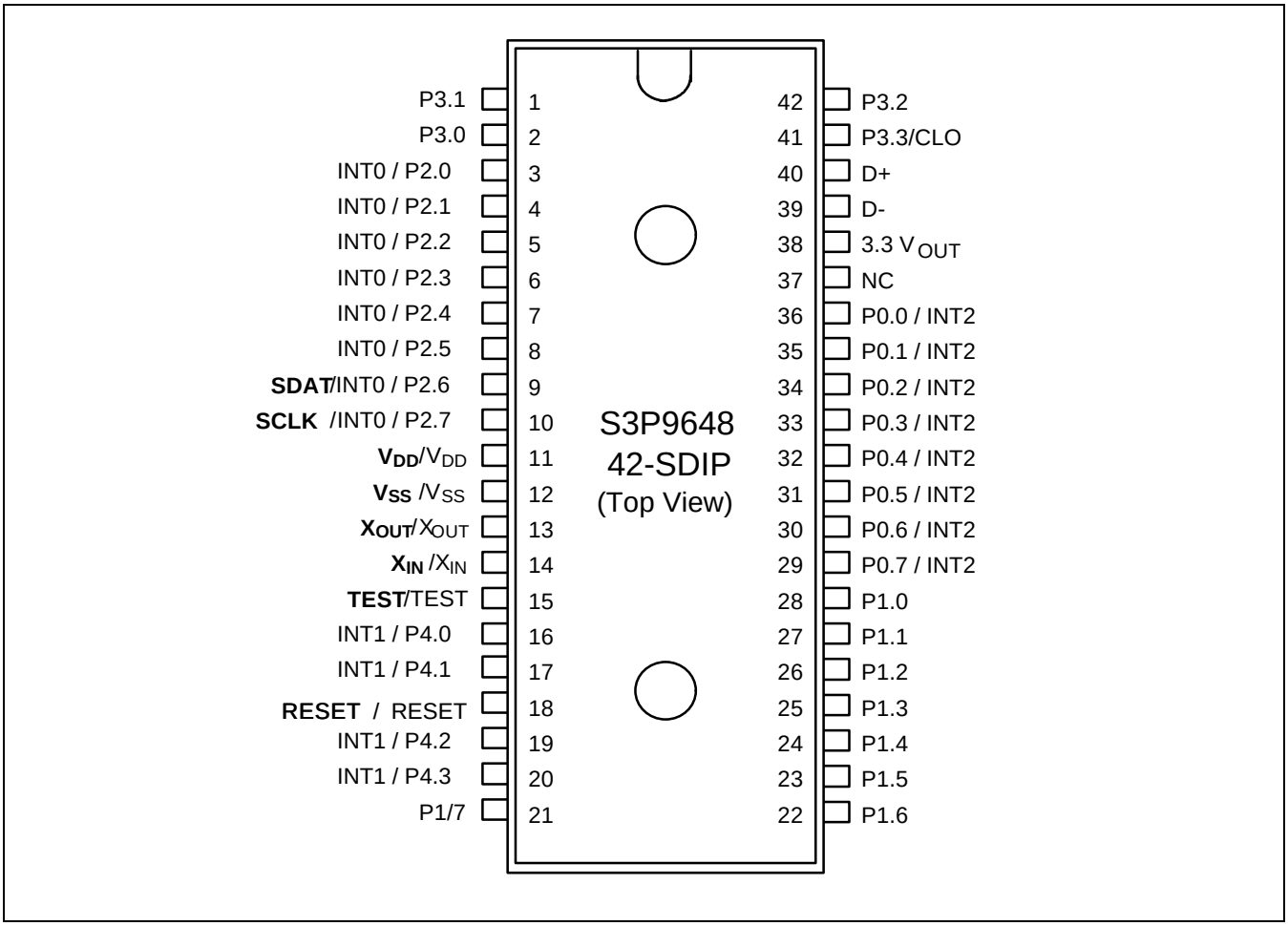


Figure 14-1. S3P9648 Pin Assignments (42-SDIP Package)

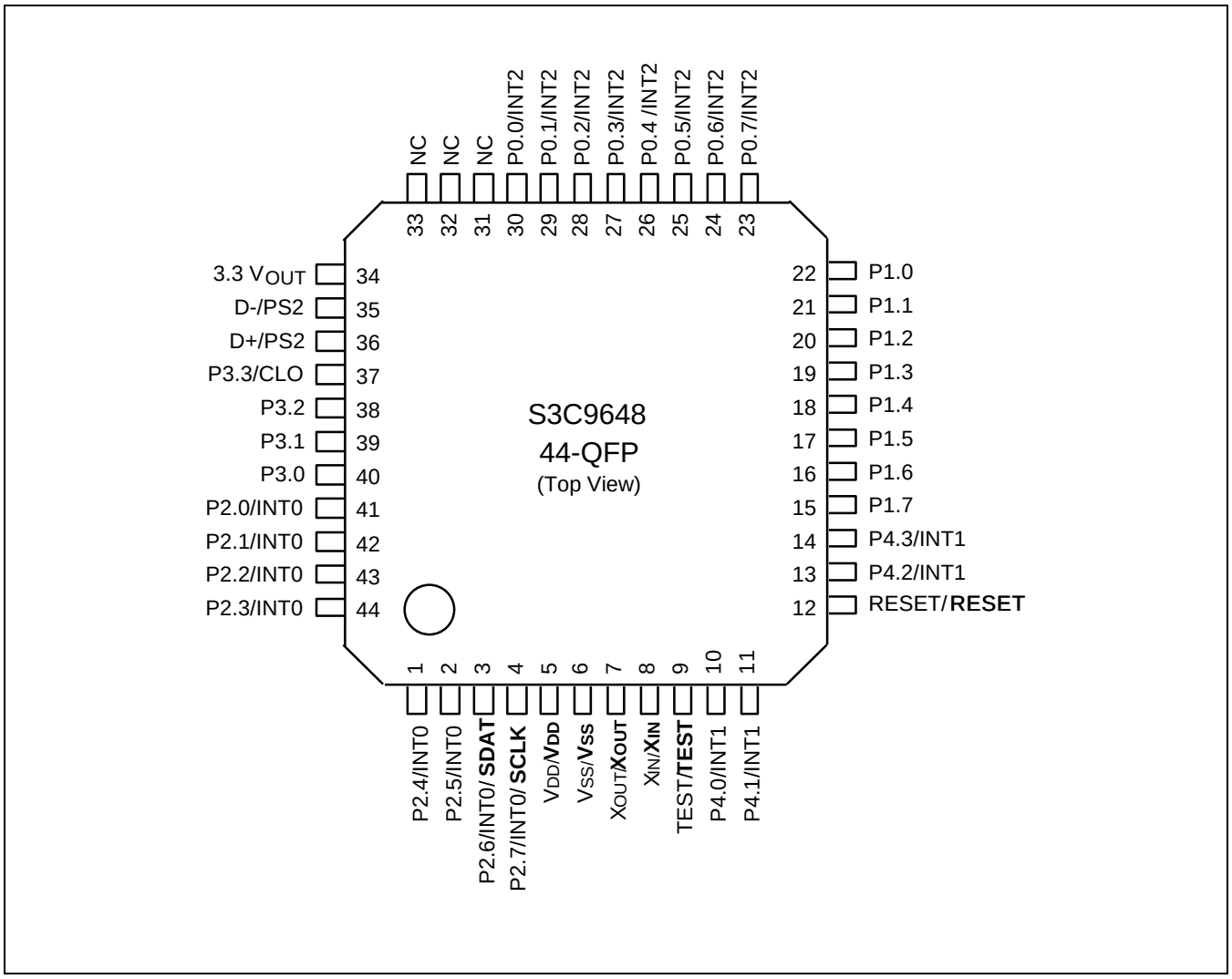


Figure 14-2. S3P9648 Pin Assignments (44-QFP Package)

Table 14-1. Descriptions of Pins Used to Read/Write the EPROM

Main Chip	During Programming			
Pin Name	Pin Name	Pin No.	I/O	Function
P2.6	SDAT	9 (3)	I/O	Serial DATa Pin (Output when reading, Input when writing) Input and Push-pull Output Port can be assigned
P2.7	SCLK	10 (4)	I/O	Serial CLocK Pin (Input Only Pin)
TEST	TEST	15 (9)	I	Chip Initialization and EPROM Cell Writing Power Supply Pin (Indicates OTP Mode Entering) When writing 12.5 V is applied and when reading.
RESET	RESET	18 (12)	I	0 V: OTP write and test mode 5 V: Operating mode
V_{DD} / V_{SS}	V_{DD} / V_{SS}	11(5)/12(6)	–	Logic Power Supply Pin.

NOTE: ( ) means 44 QFP package.

Table 14-2. Comparison of S3P9648 and S3C9644/C9648 Features

Characteristic	S3P9648	S3C9644/C9648
Program Memory	8-Kbyte EPROM	8-Kbyte mask ROM
Operating Voltage (V_{DD})	4.0 V to 5.25 V	4.0 V to 5.25 V
OTP Programming Mode	V_{DD} = 5 V, V_{PP} (RESET) = 12.5 V	
Pin Configuration	42 SDIP/44 QFP	42 SDIP/44 QFP
EPROM Programmability	User Program 1 time	Programmed at the factory

OPERATING MODE CHARACTERISTICS

When 12.5 V is supplied to the V_{PP} (RESET) pin of the S3P9648, the EPROM programming mode is entered. The operating mode (read, write, or read protection) is selected according to the input signals to the pins listed in Table 14-3 below.

Table 14-3. Operating Mode Selection Criteria

V_{DD}	VPP (RESET)	REG/ MEM	ADDRESS (A15-A0)	R/W	MODE
5 V	5 V	0	0000H	1	EPROM read
	12.5 V	0	0000H	0	EPROM program
	12.5 V	0	0000H	1	EPROM verify
	12.5 V	1	0E3FH	0	EPROM read protection

NOTE: "0" means Low level; "1" means High level.

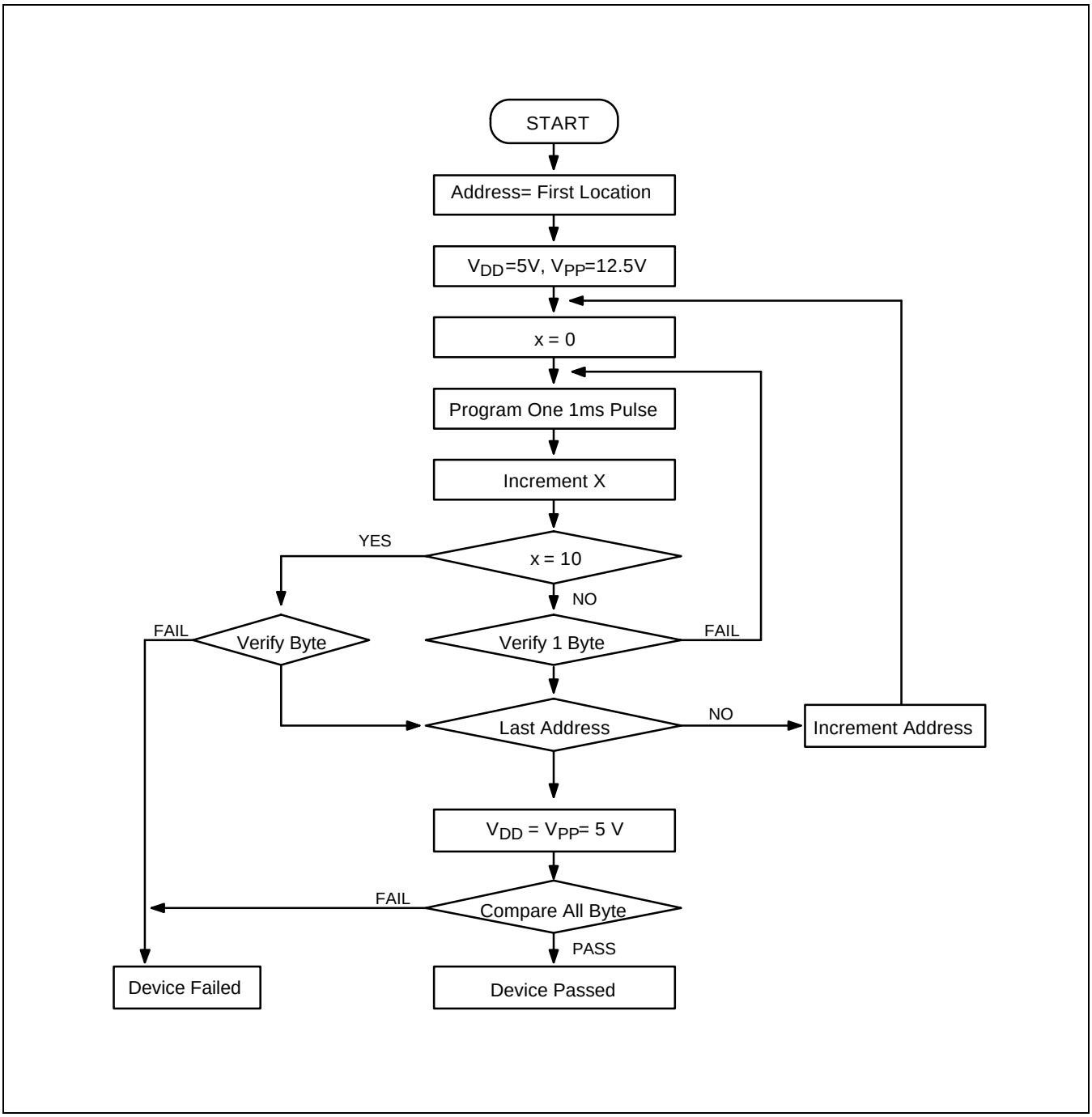


Figure 14-3. OTP Programming Algorithm

Table 14-4. D.C. Electrical Characteristics

(T_A = – 40_C to + 85_C, V_{DD} = 4.0 V to 5.25 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Supply Current (note)	I_{DD1}	Normal mode; 6 MHz CPU clock	–	5.5	12	mA
	I_{DD2}	Idle mode; 6 MHz CPU clock		2.2	5	
	I_{DD3}	Stop mode		180	300	µA

NOTE: Supply current does not include current drawn through internal pull-up resistors or external output current loads.

15 DEVELOPMENT TOOLS

OVERVIEW

Samsung provides a powerful and easy-to-use development support system in turnkey form. The development support system is configured with a host system, debugging tools, and support software. For the host system, any standard computer that operates with MS-DOS as its operating system can be used. One type of debugging tool including hardware and software is provided: the sophisticated and powerful in-circuit emulator, SMDS2+, for S3C7, S3C8, S3C9 families of microcontrollers. The SMDS2+ is a new and improved version of SMDS2. Samsung also offers support software that includes debugger, assembler, and a program for setting options.

SHINE

Samsung Host Interface for in-circuit Emulator, SHINE, is a multi-window based debugger for SMDS2+. SHINE provides pull-down and pop-up menus, mouse support, function/hot keys, and context-sensitive hyper-linked help. It has an advanced, multiple-windowed user interface that emphasizes ease of use. Each window can be sized, moved, scrolled, highlighted, added, or removed completely.

SAMA ASSEMBLER

The Samsung Arrangeable Microcontroller (SAM) Assembler, SAMA, is a universal assembler, and generates object code in standard hexadecimal format. Assembled program code includes the object code that is used for ROM data and required SMDS program control data. To assemble programs, SAMA requires a source file and an auxiliary definition (DEF) file with device specific information.

SASM86

The SASM86 is an relocatable assembler for Samsung's S3C9-series microcontrollers. The SASM86 takes a source file containing assembly language statements and translates into a corresponding source code, object code and comments. The SASM86 supports macros and conditional assembly. It runs on the MS-DOS operating system. It produces the relocatable object code only, so the user should link object file. Object files can be linked with other object files and loaded into memory.

HEX2ROM

HEX2ROM file generates ROM code from HEX file which has been produced by assembler. ROM code must be needed to fabricate a microcontroller which has a mask ROM. When generating the ROM code (.OBJ file) by HEX2ROM, the value "FF" is filled into the unused ROM area upto the maximum ROM size of the target device automatically.

TARGET BOARDS

Target boards are available for all S3C9-series microcontrollers. All required target system cables and adapters are included with the device-specific target board.

OTPs

One times programmable microcontrollers (OTPs) are under development for S3C9644/C9648 microcontroller.

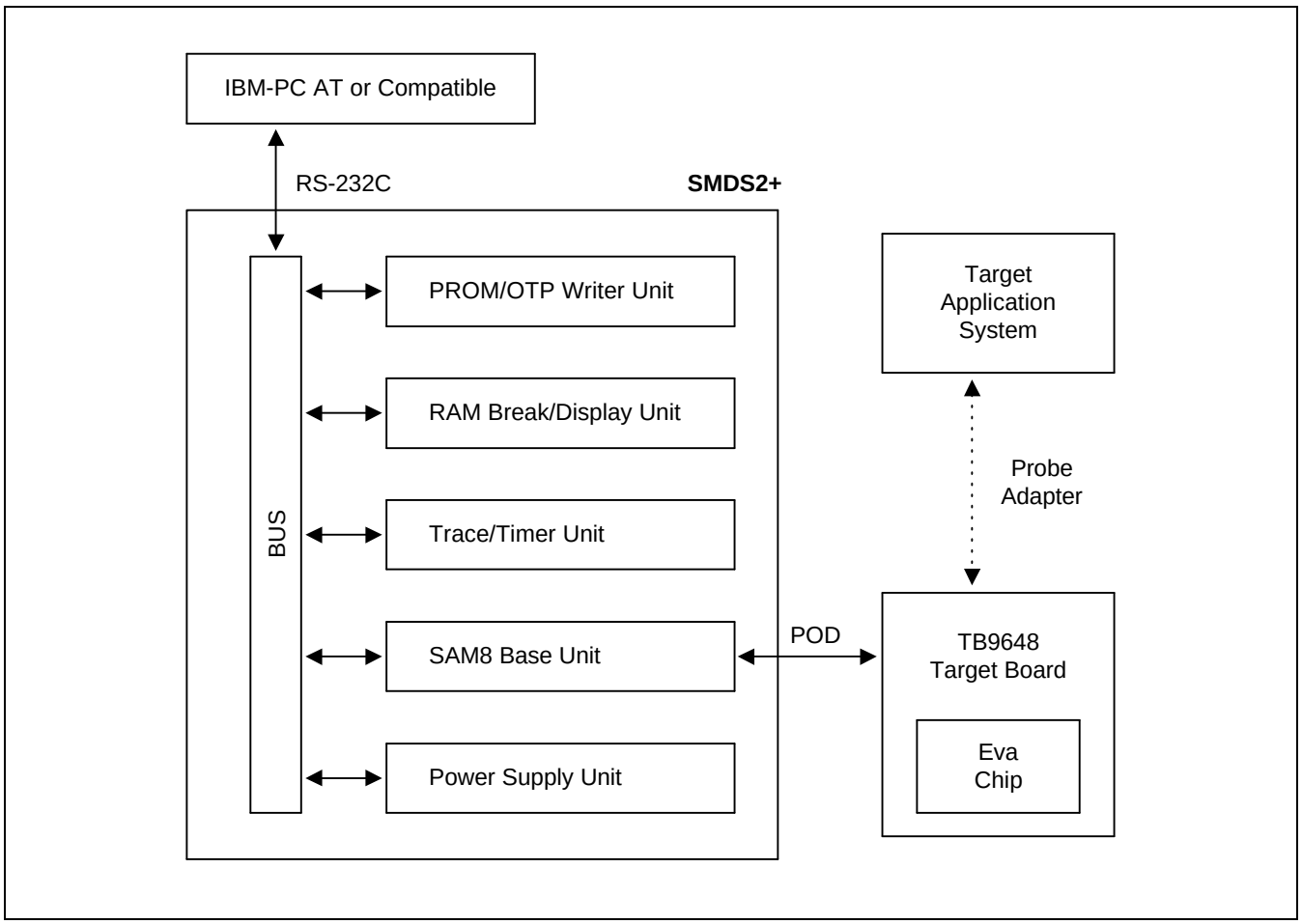


Figure 15-1. SMDS Product Configuration (SMDS2+)

TB9648 TARGET BOARD

The TB9648 target board is used for the S3C9644/C9648 microcontrollers. It is supported by the SMDS2+ development systems. The TB9648 target board can also be used for S3C9644/C9648.

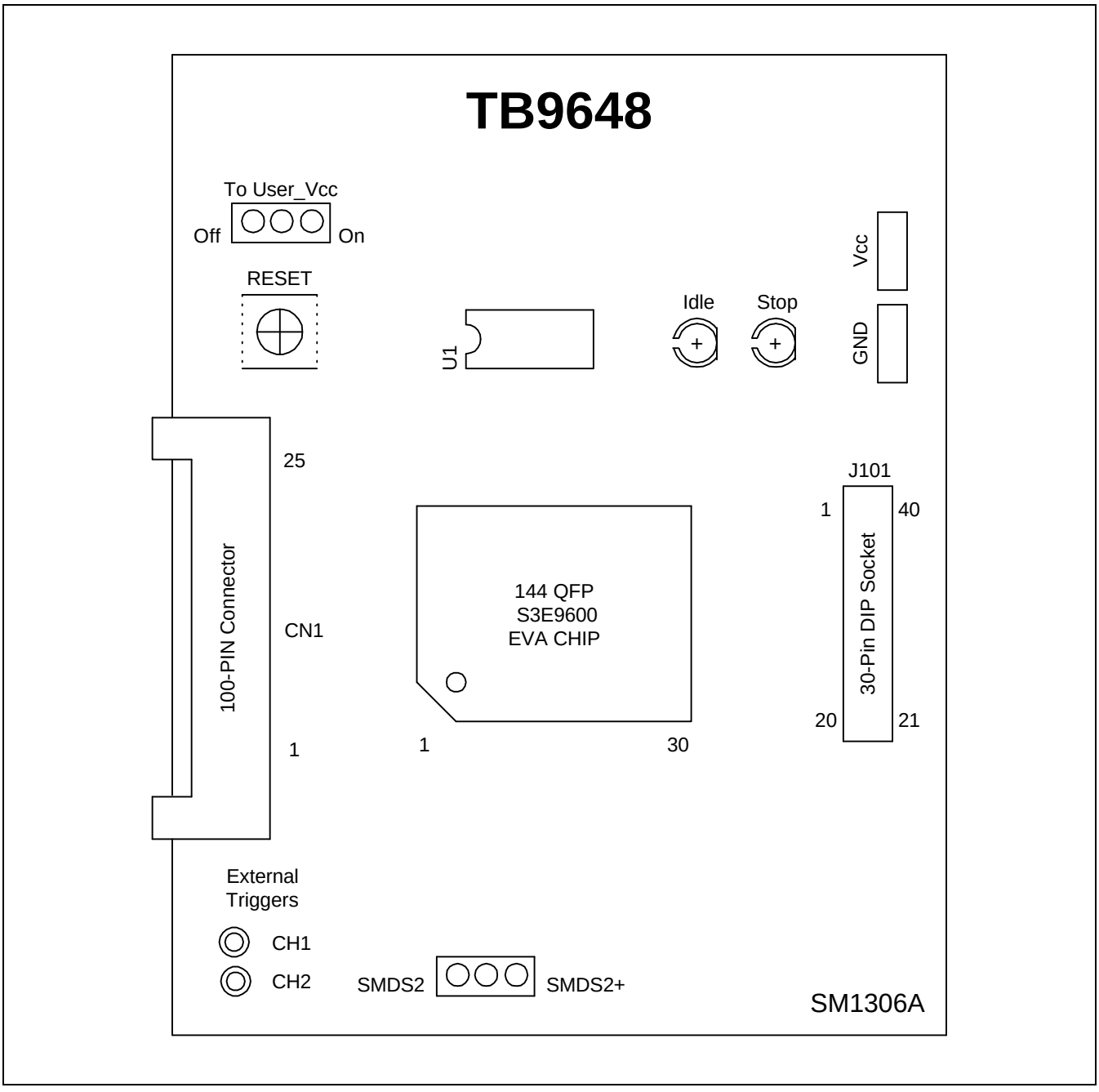
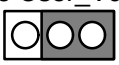
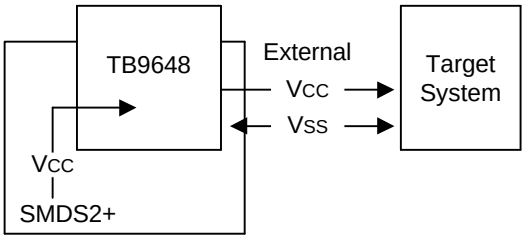
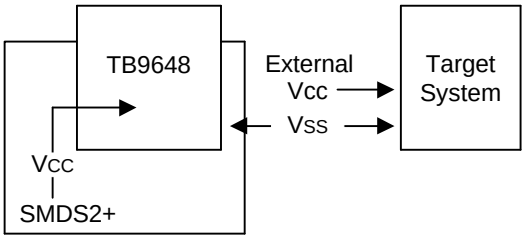
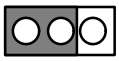


Figure 15-2. TB9648 Target Board Configuration

Table 15-1. Power Selection Settings for TB9648

'To User_Vcc' Settings	Operating Mode	Comments
To User_Vcc OFF  ON		The SMDS2/SMDS2+ supplies V_{CC} to the target board (evaluation chip) and the target system.
To User_Vcc OFF  ON		The SMDS2/SMDS2+ supplies V_{CC} only to the target board (evaluation chip). The target system must have its own power supply.

NOTE: The following symbol in the "To User_V_{CC}" Setting column indicates the electrical short (off) configuration:



SMDS2+ Selection (SAM8)

In order to write data into program memory that is available in SMDS2+, the target board should be selected to be for SMDS2+ through a switch as follows. Otherwise, the program memory writing function is not available.

Table 15-2. The SMDS2+ Tool Selection Setting

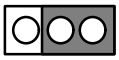
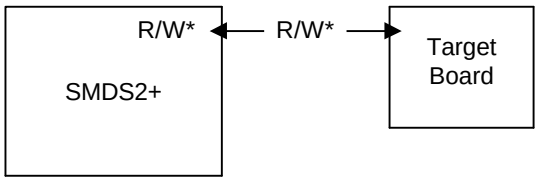
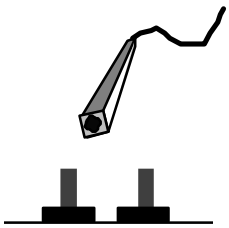
"SW1" Setting	Operating Mode
SMDS2  SMDS2+	

Table 15-3. Using Single Header Pins as the Input Path for External Trigger Sources

Target Board Part	Comments
EXTERNAL TRIGGERS CH1 CH2	 Connector from external trigger sources of the application system You can connect an external trigger source to one of the two external trigger channels (CH1 or CH2) for the SMDS2+ breakpoint and trace functions.

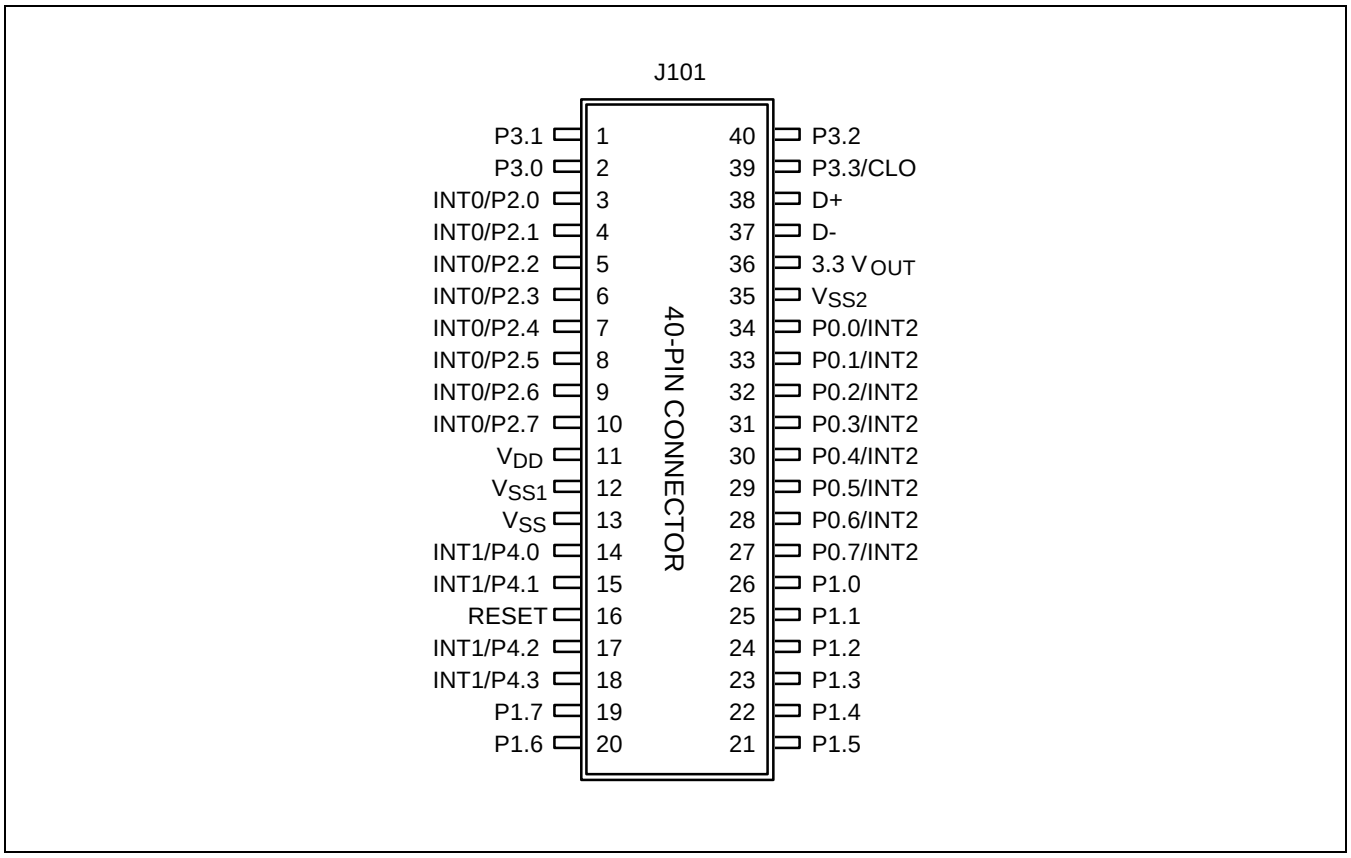


Figure 15-3. 40-Pin Connector for TB9648

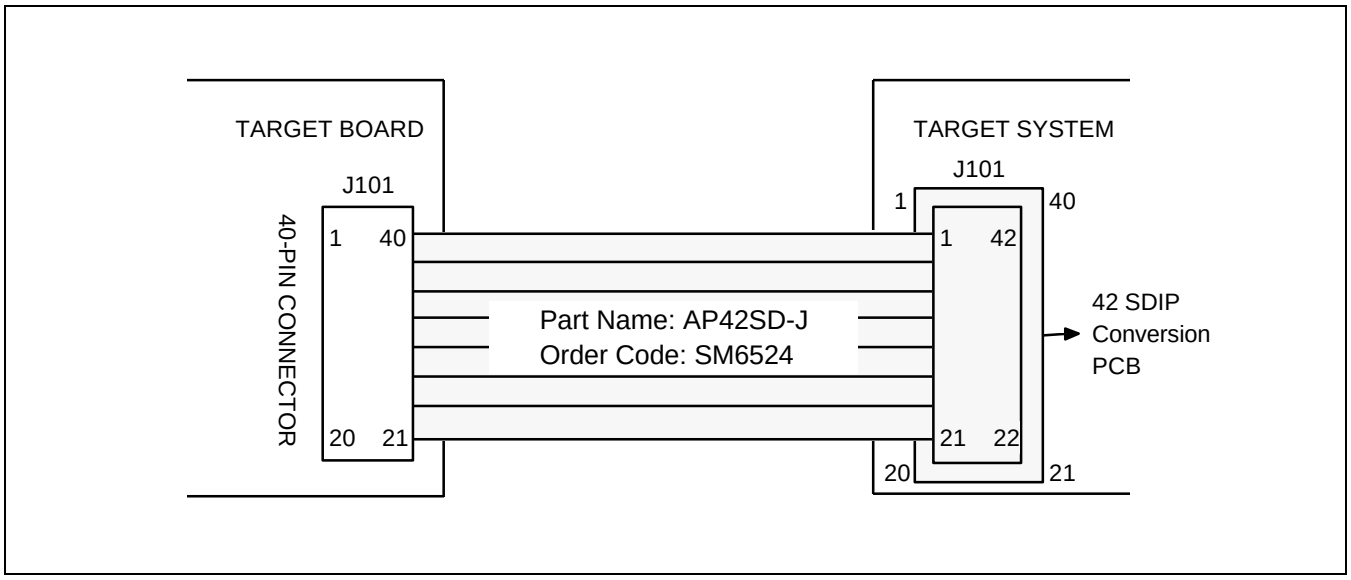


Figure 15-4. S3C9648 Probe Adapter Cable for 42-SDIP Package