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 address/data bus architecture and a large number of bit-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.

S3C9424/C9428/P9428 MICROCONTROLLER

The S3C9424/C9428/P9428 single-chip 8-bit microcontroller is fabricated using an advanced CMOS process. It is built around the powerful SAM87Ri CPU core. The S3C9424/C9428/P9428 is a versatile microcontroller, with its A/D converter, SIO, IIC and a zero-crossing detection capability it can be used in a wide range of general purpose applications.

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 S3C9424/C9428/P9428 have 4K-byte or 8K-byte of program memory on-chip (ROM) and 208-bytes of general purpose register area RAM.

Using the SAM87Ri design approach, the following peripherals were integrated with the SAM87Ri core:

- Four configurable I/O ports (24 pins)
- Nine interrupt sources with one vector and one interrupt level
- Two 8-bit timer/counter with various operating modes
- Analog to digital converter with 12 input channels and 10-bit resolution
- One synchronous SIO module
- One IIC module
- Two 12-bit PWM output

The S3C9424/C9428/P9428 microcontroller is ideal for use in a wide range of electronic applications requiring simple timer/counter, PWM, ADC, SIO, IIC, ZCD and capture functions. S3C9424/C9428/P9428 is available in a 28/32-pin SOP and a 30-pin SDIP package.

OTP

The S3P9428 is an OTP (One Time Programmable) version of the S3C9424/C9428 microcontroller. The S3P9428 has on-chip 8-K-byte one-time-programmable EPROM instead of masked ROM. The S3P9428 is fully compatible with the S3C9424/C9428, in function, in D.C. electrical characteristics and in pin configuration.

FEATURES

CPU

- SAM87RI CPU core

Memory

- 208-byte general purpose register area (RAM)
- 4K/8K byte internal program memory (ROM)

Instruction Set

- 41 instructions
- The SAM87RI core provides all the SAM87 core instruction except the word-oriented instruction, multiplication, division, and some one-byte instruction

Instruction Execution Time

- 375 ns at 16 MHz fosc(minimum)

Interrupts

- 9 interrupt sources and 1 vector
- One interrupt level

General I/O

- Four I/O ports (total 24pins)
- Bit programmable ports

Serial I/O

- One synchronous serial I/O module
- Selectable transmit and receive rates

Multi-Master IIC-Bus

- Serial peripheral interface

Zero-Crossing Detection Circuit

- Zero crossing detection circuit that generates a digital signal in synchronism with an AC signal input

Built-in reset Circuit (LVD)

- Low voltage detector for safe reset

Timer/Counters

- One 8-bit basic timer for watchdog function
- One 8-bit timer/counter with three operating mode
- One 8-bit timer/counter

PWM module

- 12-bit PWM 2-ch (Max: 250KHz)
- 6-bit base + 6-bit extension frame
- One 8-bit timer/counter

A/D Converter

- 12 analog input pins
- 10-bit conversion resolution

Buzzer Frequency Range

- 200 Hz to 20 kHz signal can be generated

Oscillator Freqeuncy

- 1-MHz to 16-MHz external crystal oscillator Maximum 16-MHz CPU clock
- RC: 4MHz(typ)

Operating Temperature Range

- – 40°C to + 85°C

Operating Voltage Range

- 3.0 V to 5.5 V (LVD)
- 1.8 V to 5.5 V (No LVD)

OTP Interface Protocol Spec

- Serial OTP

Package Types

- S3C9424/C9428
  32-pin SOP-450 (3V LVD)
  30-pin SDIP-400 (3V LVD)
  28-pin SOP-375

BLOCK DIAGRAM

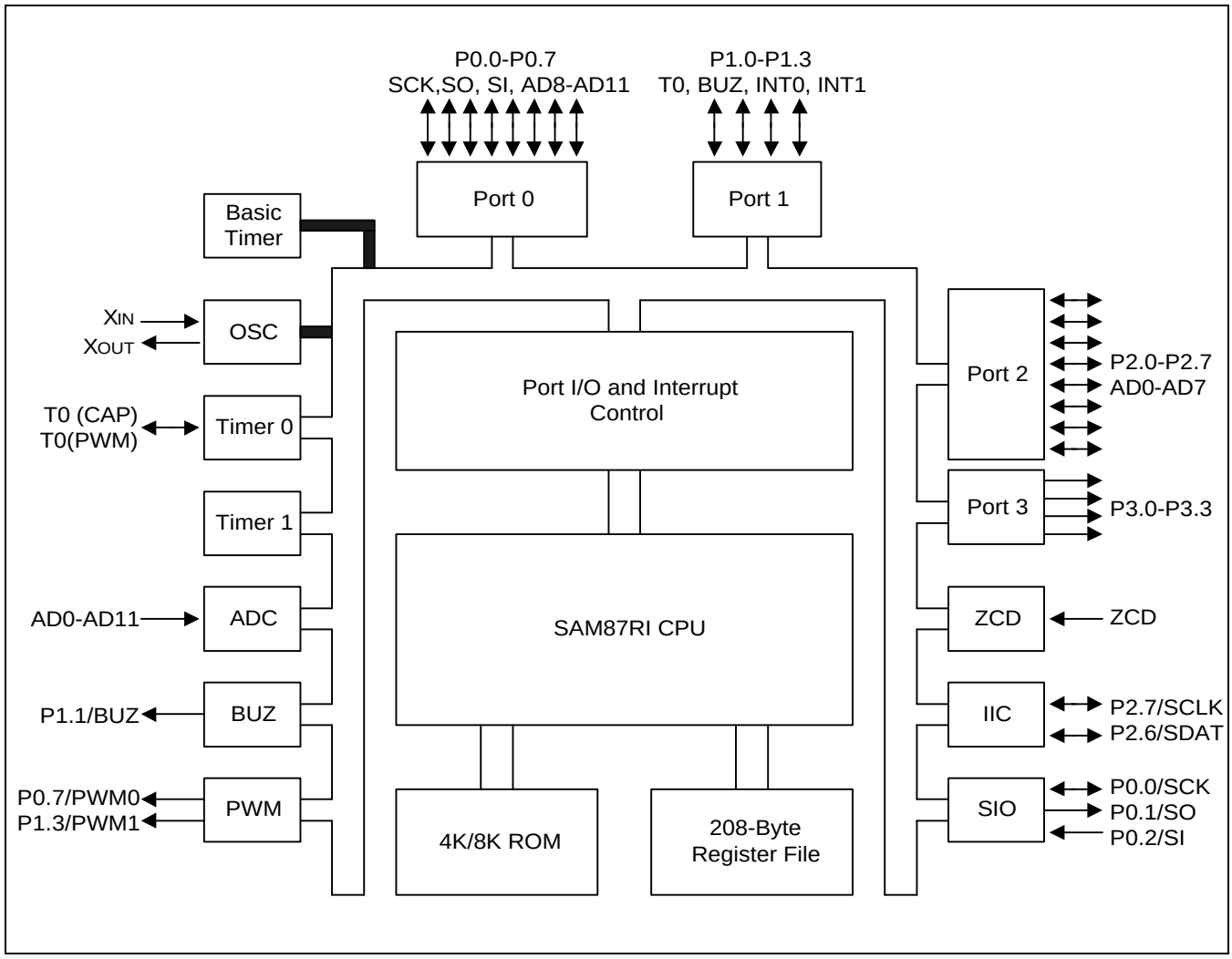


Figure 1-1. Block Diagram

PIN ASSIGNMENTS

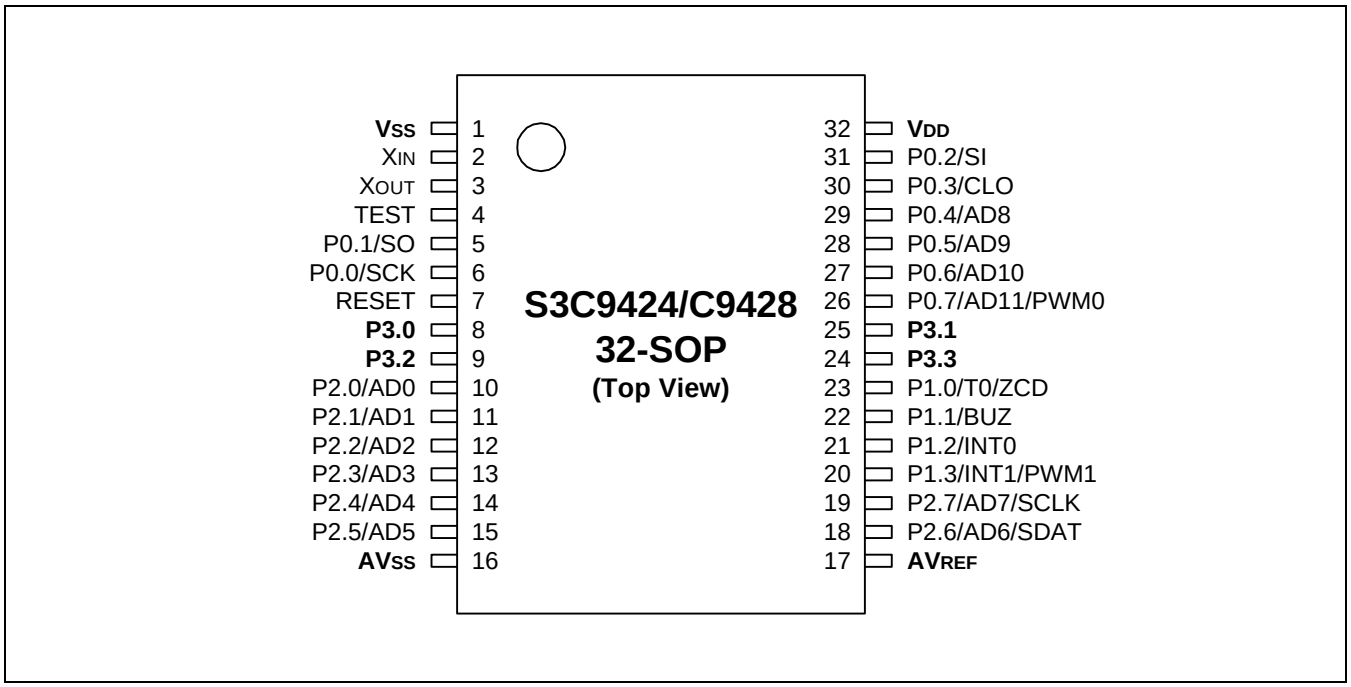


Figure 1-2. Pin Assignment Diagram (32-Pin SOP Package)

PIN ASSIGNMENTS (Continued)

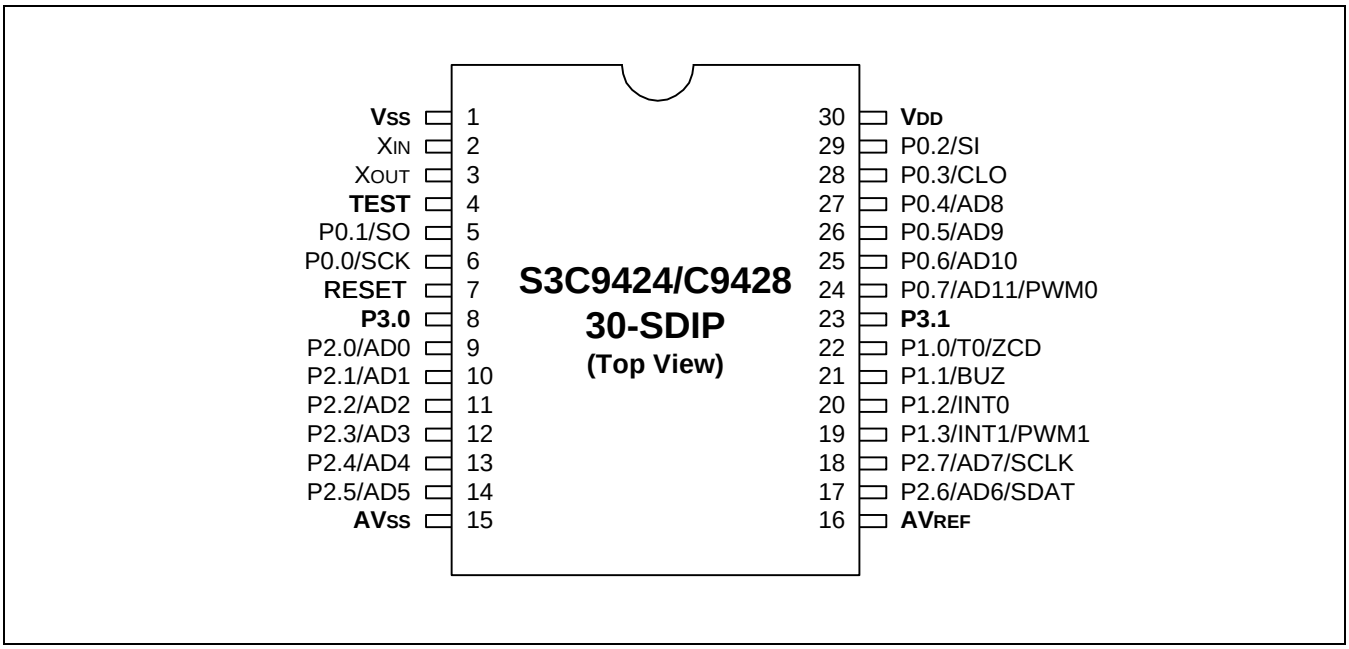


Figure 1-3. Pin Assignment Diagram (30-Pin SDIP Package)

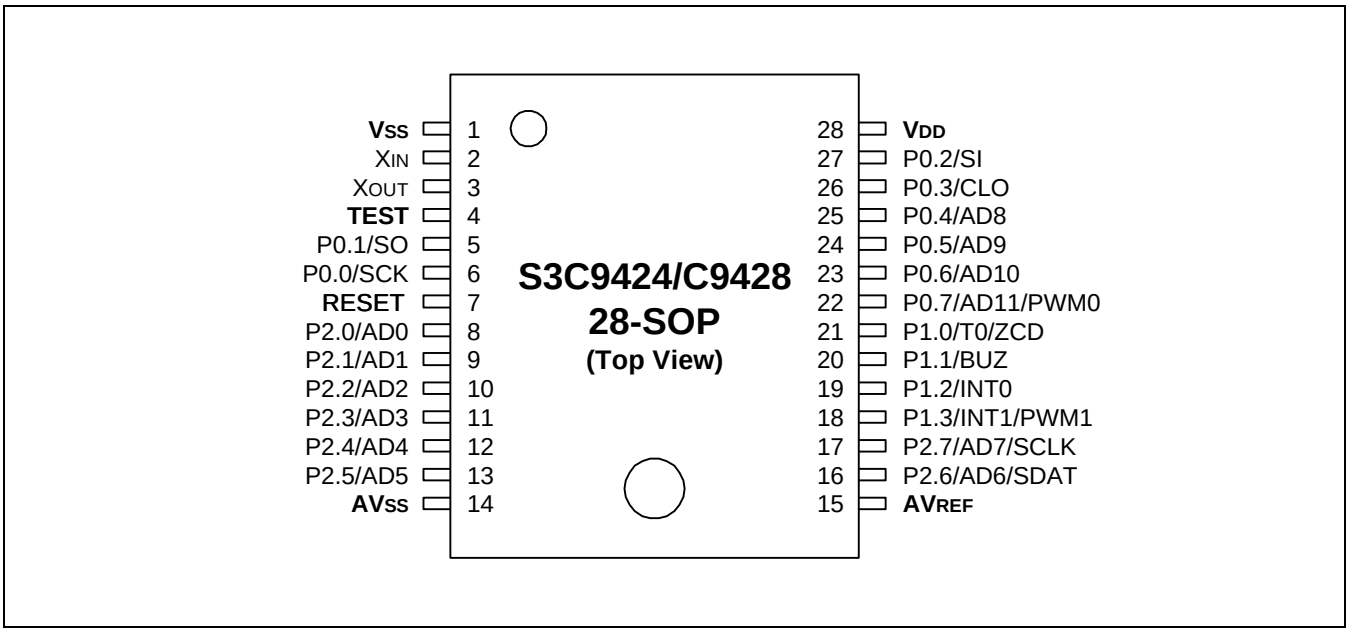


Figure 1-4. Pin Assignment Diagram (28-Pin SOP Package)

PIN DESCRIPTIONS

Table 1-1. S3C9424/C9428/P9428 Pin Descriptions

Pin Names	Pin Type	Pin Description	Pin Type	Share Pins
P0.0-P0.7	I/O	Bit-programmable I/O port for Schmitt trigger input or push-pull, open-drain output. Pull-up resistors are assignable by software.	E E-1	SCK,SO,SI , CLO, AD8-AD11
P1.0-P1.3	I/O	Bit-programmable I/O port for Schmitt trigger input or push-pull output. Pull-up resistors are assignable by software. Port 1 pins can also be used as alternative functions.	D	T0/ZCD BUZ INT0 INT1
P2.0-P2.7	I/O	Bit-programmable I/O port for Schmitt trigger input or push-pull, open drain output. Pull up resistors are assignable by software. Port 2 can also be used as external interrupt, A/D input.	E-1	AD0-AD7
P3.0-P3.3	O	Push-pull or open-drain output port. Pull-up resistors are assignable by software.	E-2	–
X_{IN}, X_{OUT}	–	Crystal/ceramic, or RC oscillator signal for system clock.	–	–
RESET	I	System RESET signal input pin.	B	–
TEST	I	Test signal input pin (for factory use only: must be connected to V_{SS})	–	–
AV_{REF}, AV_{SS}	–	A/D converter reference voltage input and ground	–	–
V_{DD}, V_{SS}	–	Voltage input pin and ground	–	–
SCK	I/O	Serial interface clock input or output	E	P0.0
SO	O	Serial data output	E	P0.1
SI	I	Serial data output	E	P0.2
CLO	O	System clock output port	E	P0.3
SCLK SDAT	I/O	IIC CLOCK IIC DATA	E-1	P2.7 P2.6
BUZ	O	200 Hz-20 kHz frequency output for buzzer sound.	D	P1.1
ZCD	I	Zero crossing detector input	D	P1.0
T0	I/O	Timer 0 capture input or 10-bit PWM output	D	P1.0
INT0 INT1	I	External interrupt input	D	P1.2 P1.3
PWM0 PWM1	O	12-bit PWM output	E-1 D	P0.7 P1.3
AD0-AD11	I	A/D converter input	E-1	P2.0-P2.7 P0.4-P0.7

PIN CIRCUITS

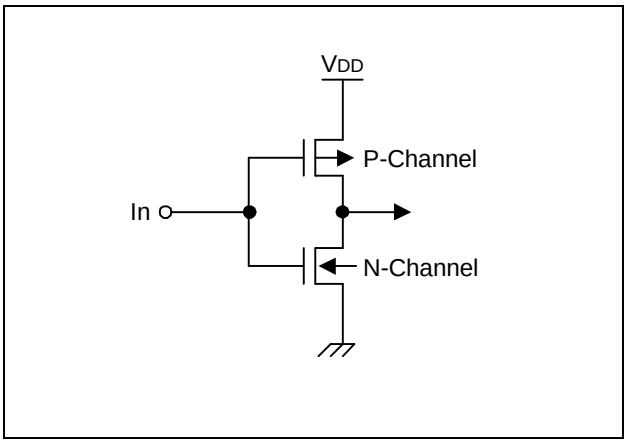


Figure 1-5. Pin Circuit Type A

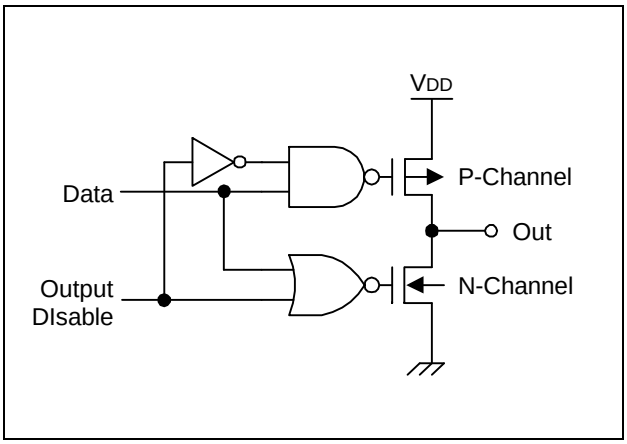


Figure 1-7. Pin Circuit Type C

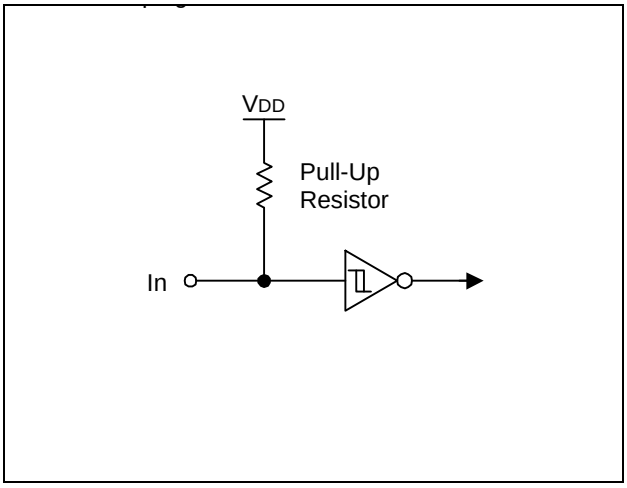


Figure 1-6. Pin Circuit Type B

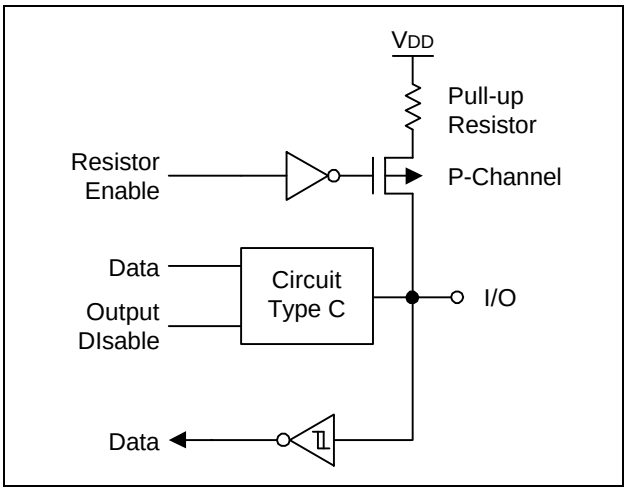


Figure 1-8. Pin Circuit Type D

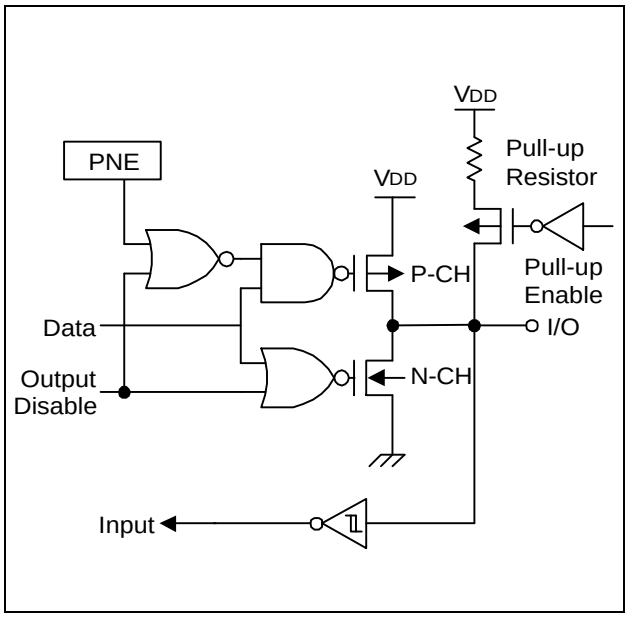


Figure 1-9. Pin Circuit Type E

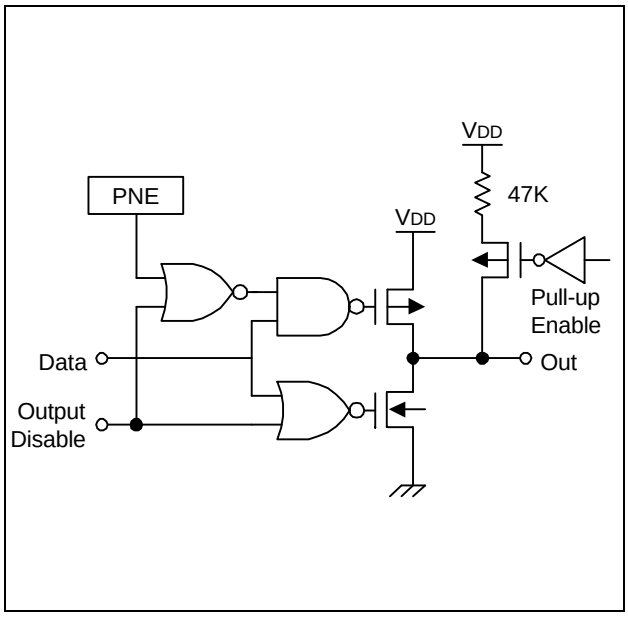


Figure 1-11. Pin Circuit Type E-2

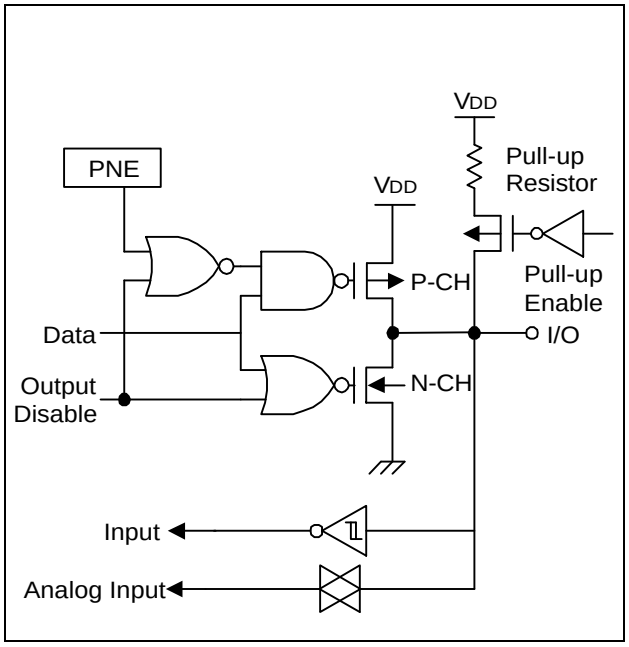


Figure 1-10. Pin Circuit Type E-1

2 ADDRESS SPACES

OVERVIEW

The S3C9424/C9428/P9428 microcontroller has two kinds of address space:

— Internal program memory (ROM)
— Internal register file

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

The S3C9424/C9428/P9428 have 4K-bytes or 8K-bytes of mask-programmable on-chip program memory: which is configured as the Internal ROM mode, all of the 4K-byte internal program memory is used.

The S3C9424/C9428/P9428 microcontroller has 208 general-purpose registers in its internal register file. Forty-four bytes in the register file are mapped for system and peripheral control functions.

PROGRAM MEMORY (ROM)

Normal Operating Mode

The S3C9424/C9428/P9428 have 4K-bytes (locations 0H-0FFFH) or 8K-bytes (locations 0H-1FFFH) of internal mask-programmable program memory.

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

Unused locations (0002H–00FFH) can be used as normal program memory.

The program reset address in the ROM is 0100H.

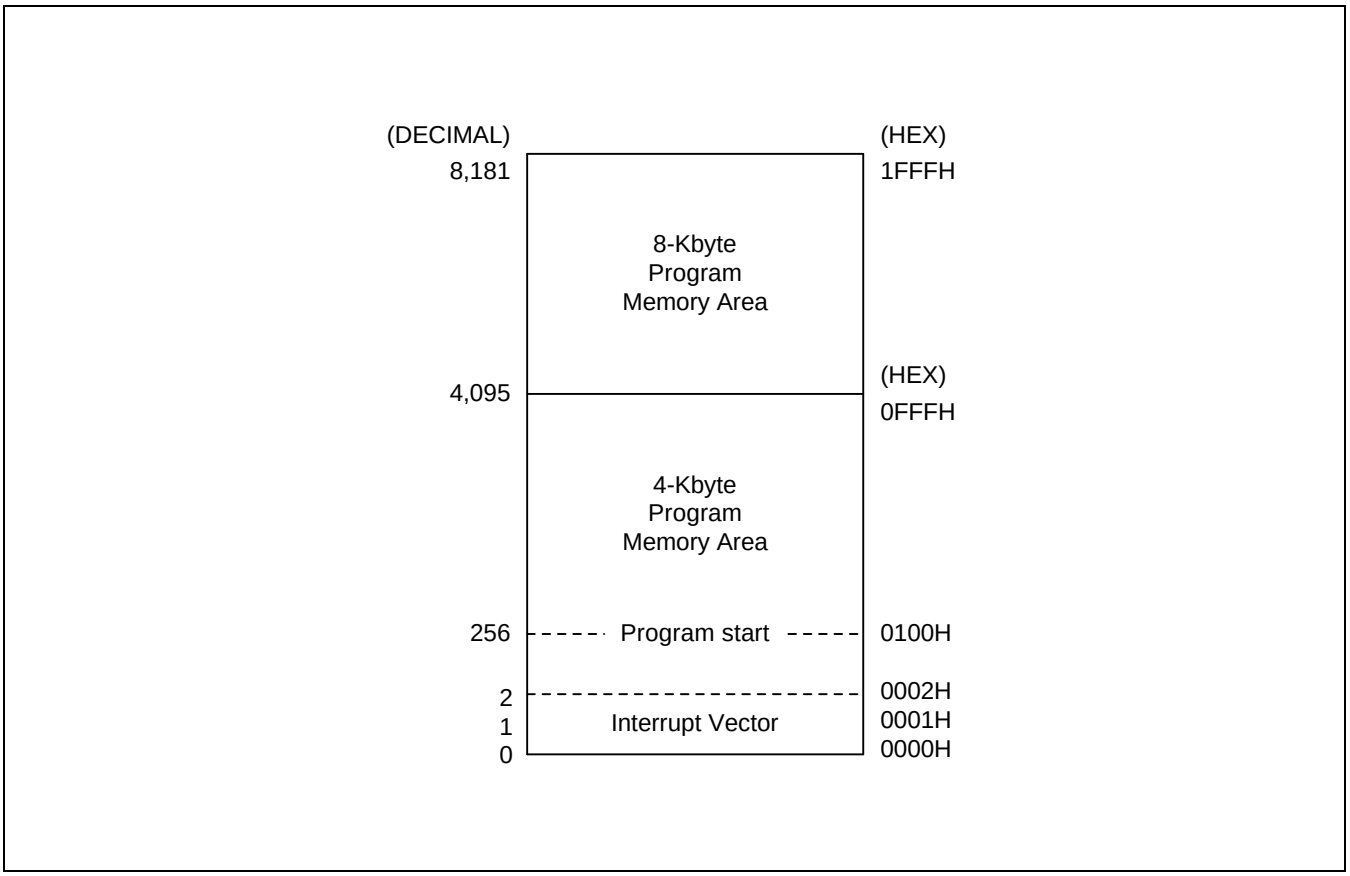


Figure 2-1. Program Memory Address Space

REGISTER ARCHITECTURE

The upper 64-bytes of the S3C9424/C9428/P9428'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. 252registers 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 additional register pages at the general purpose register space (00H-BFH). This register file expansion is not implemented in the S3C9424/C9428/P9428, however.

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	12
Peripheral, I/O, and clock control and data registers	32
General-purpose registers (including the 16-bit common working register area)	208
Total Addressable Bytes	**252**

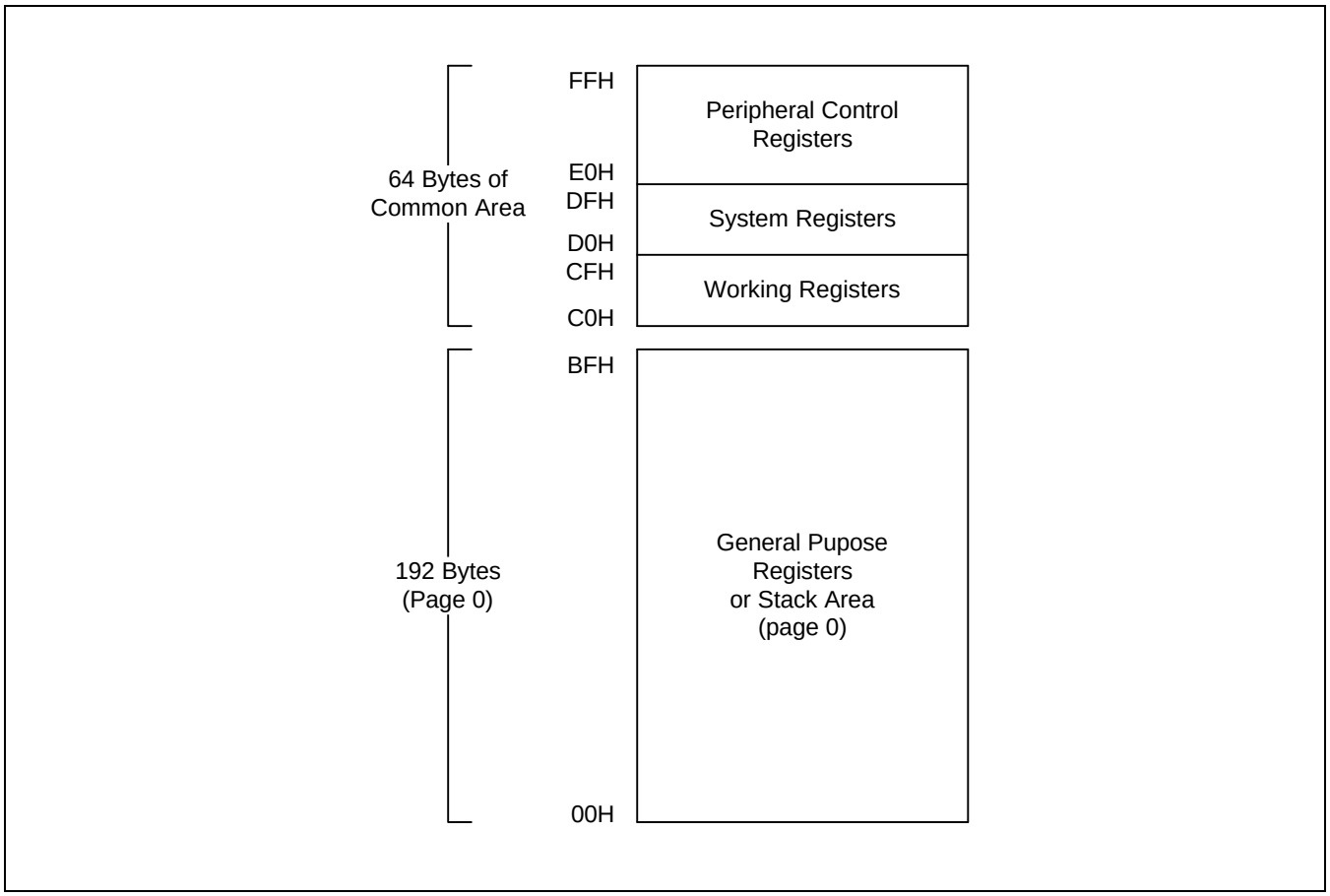


Figure 2-2. Internal Register File Organization

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 S3C9424/C9428/P9428 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.

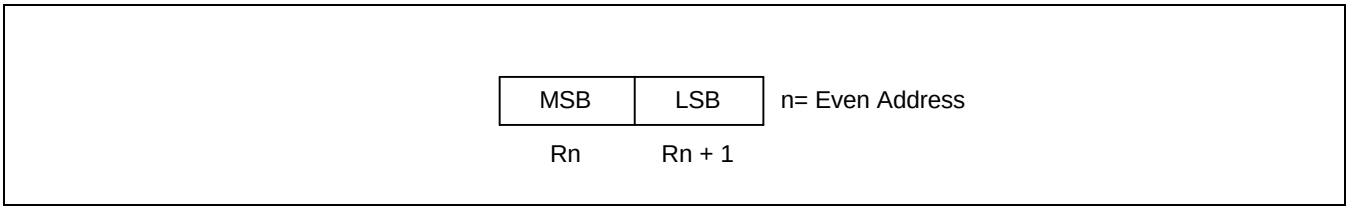


Figure 2-3. 16-Bit Register Pairs

☞ 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 S3C9424/C9428/P9428 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.

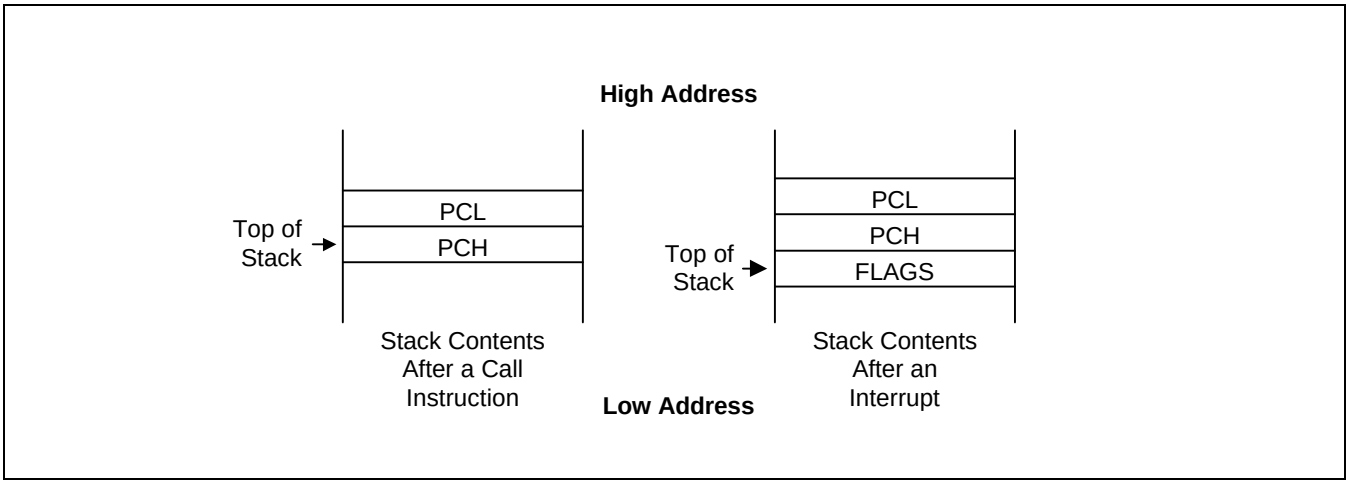


Figure 2-4. Stack Operations

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 S3C9424/C9428/P9428, the SP must be initialized to an 8-bit value in the range 00H-0C0H.

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    R15             ;  Stack address 0BEH ← R15
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     R15             ;  R15 ← Stack address 0BEH
POP     SYM             ;  SYM ← Stack address 0BFH
```

3 ADDRESSING MODES

OVERVIEW

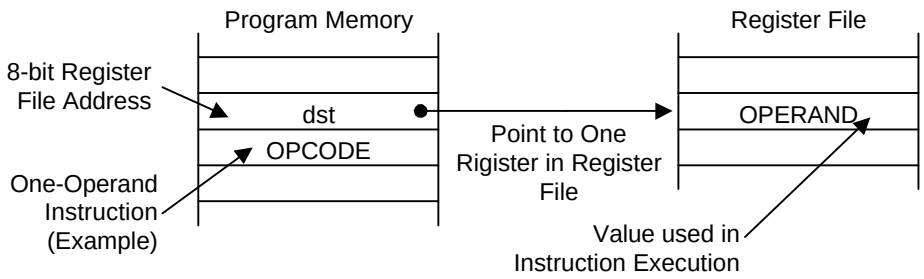
Instructions that are stored in program memory are fetched for execution using the program counter. Instructions indicate the operation to be performed and the data to be operated on. *Addressing mode* is the method used to determine the location of the data operand. The operands specified in SAM87Ri 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)

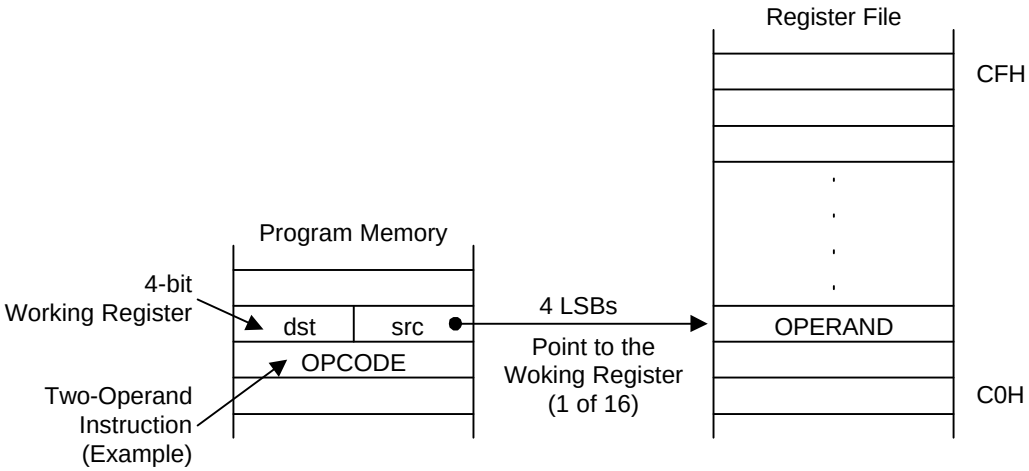
In Register addressing mode, the operand is the content of a specified register (see Figure 3-1). Working register addressing differs from Register addressing because it uses an 16-byte working register space in the register file and an 4-bit register within that space (see Figure 3-2).



Sample Instruction:

```
DEC     CNTR            ;  Where CNTR is the label of an 8-bit register address
```

Figure 3-1. Register Addressing



Sample Instruction:

```
ADD     R1, R2          ;  Where R1 = C1H and R2 = C2H
```

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.

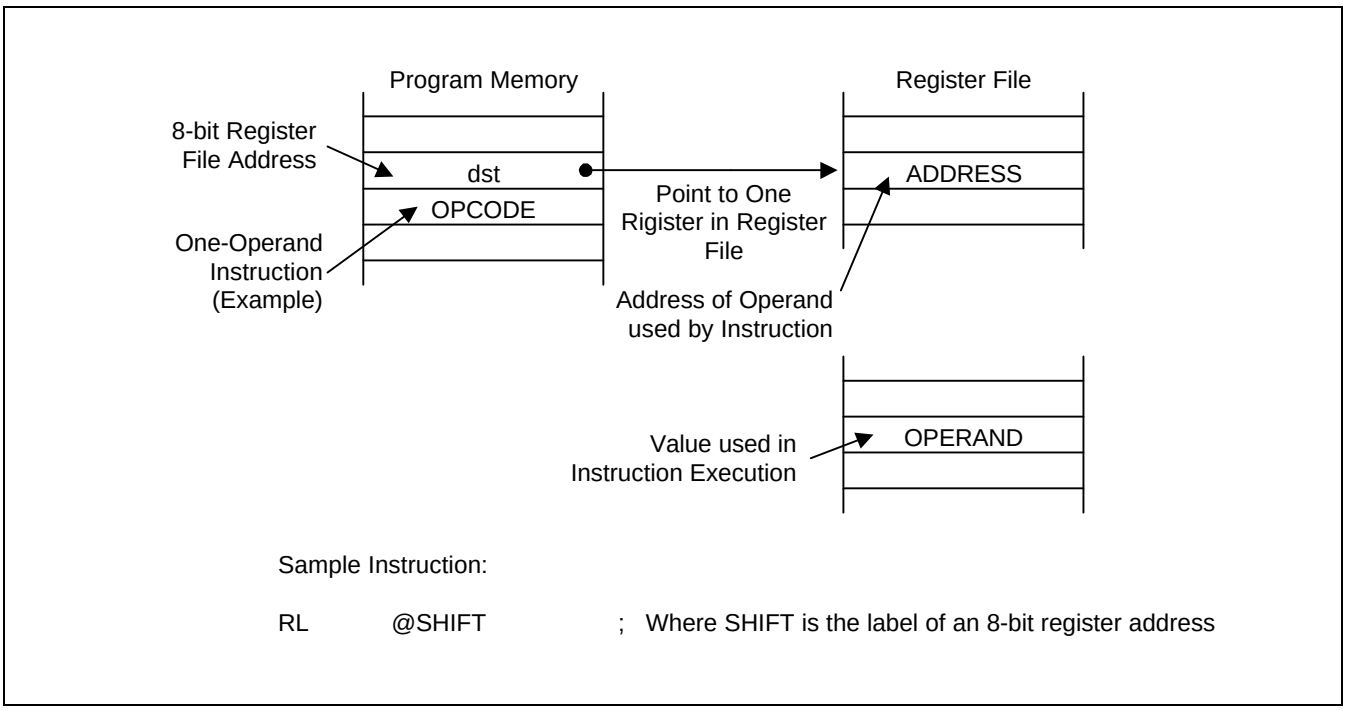


Figure 3-3. Indirect Register Addressing to Register File

INDIRECT REGISTER ADDRESSING MODE (Continued)

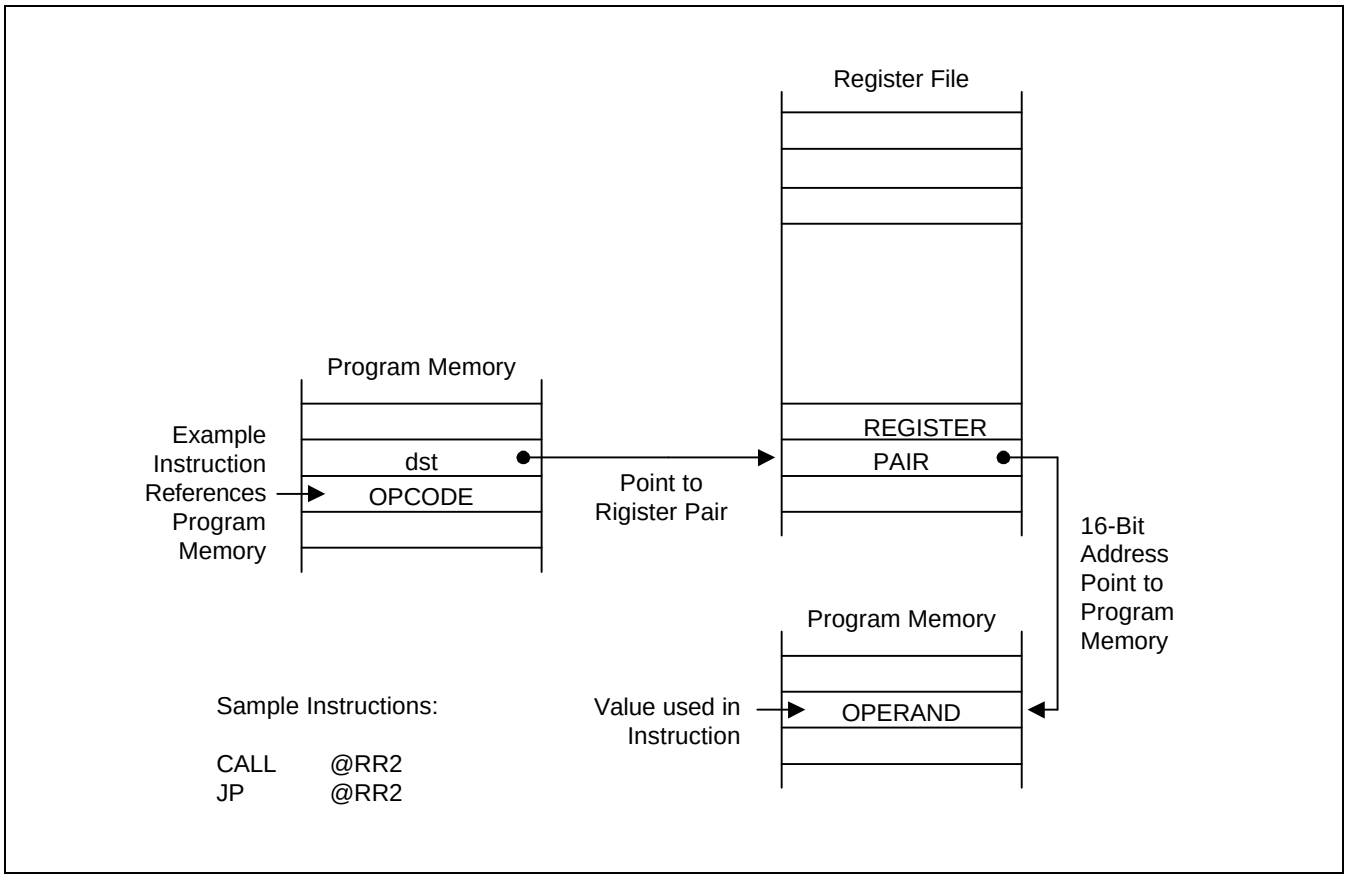


Figure 3-4. Indirect Register Addressing to Program Memory

INDIRECT REGISTER ADDRESSING MODE (Continued)

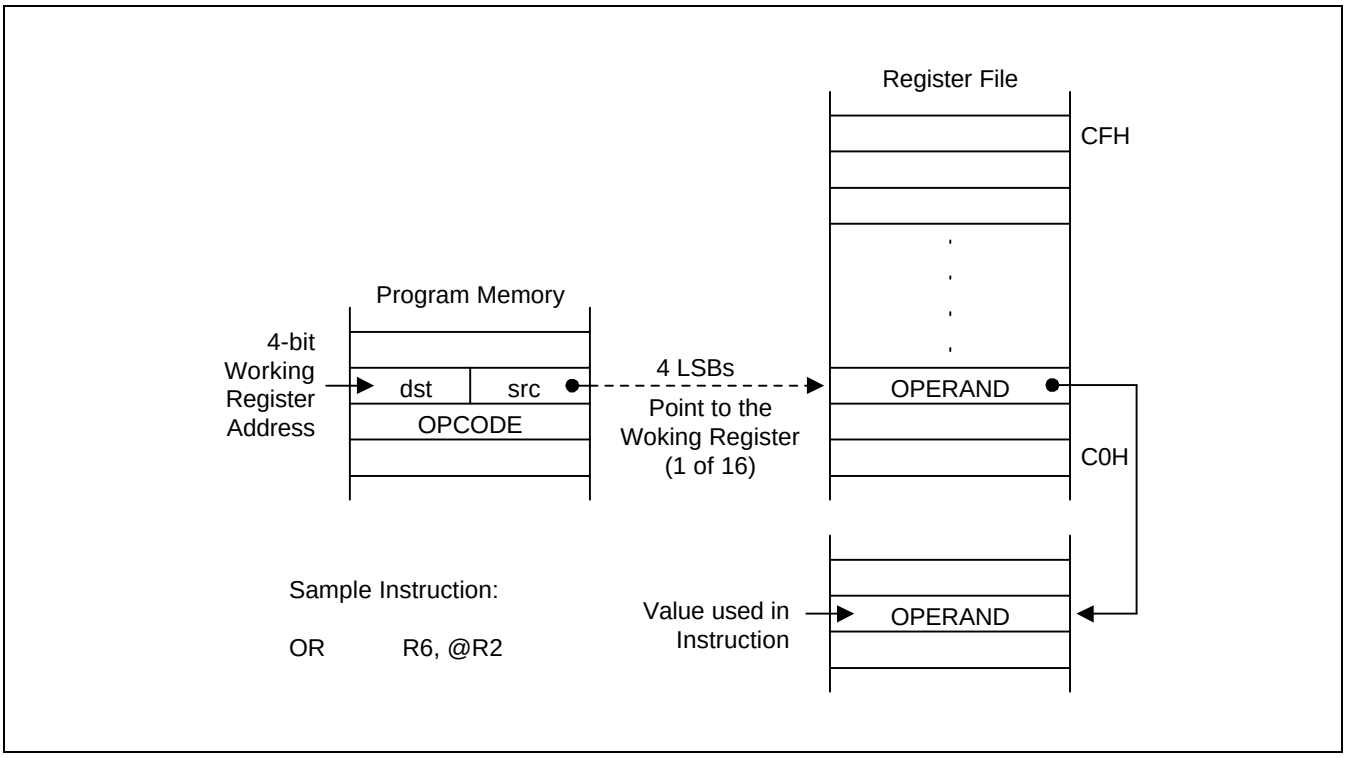
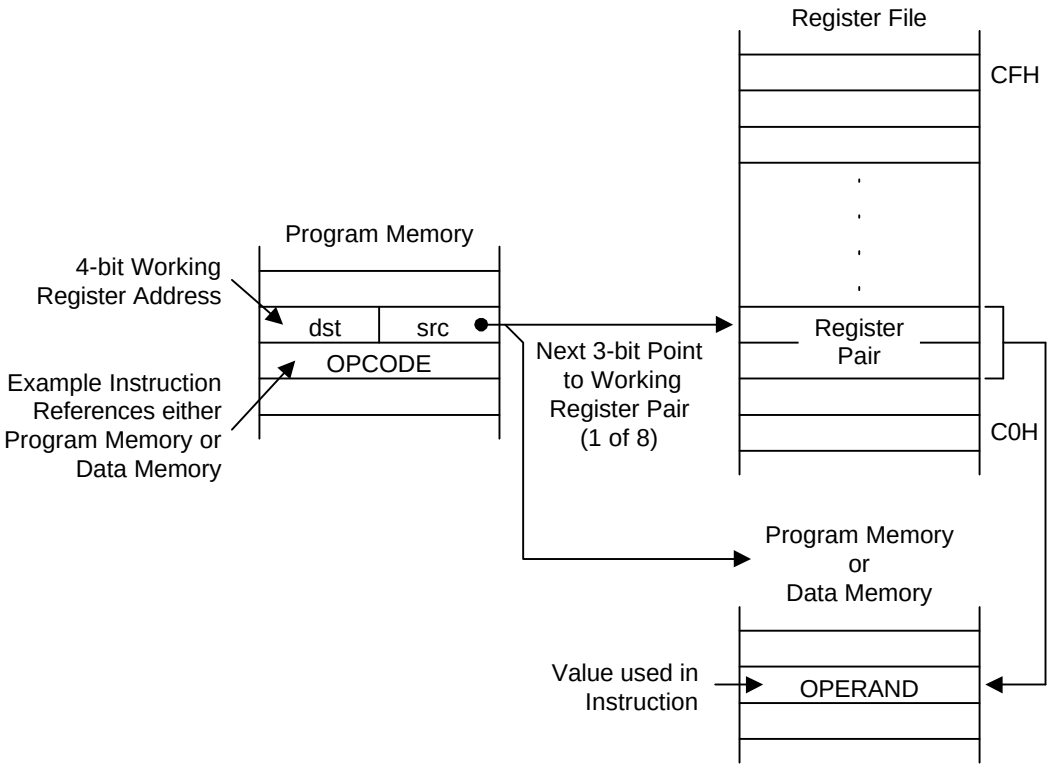


Figure 3-5. Indirect Working Register Addressing to Register File

INDIRECT REGISTER ADDRESSING MODE (Concluded)



Sample Instructions:

```
LCD     R5,@RR2       ;  Program memory access
LDE     R3,@RR14      ;  External data memory access
LDE     @RR4, R8      ;  External data memory access
```

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.

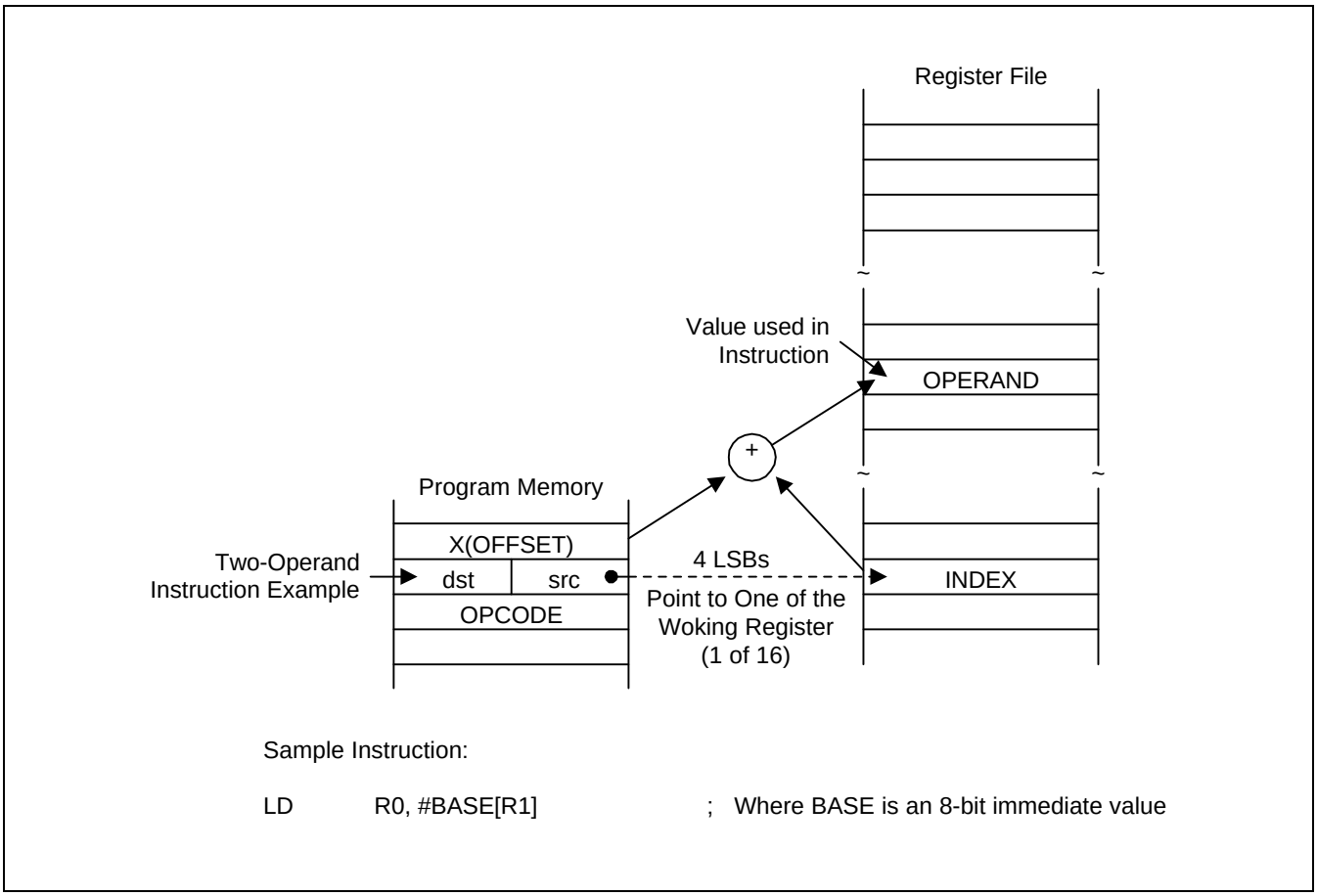
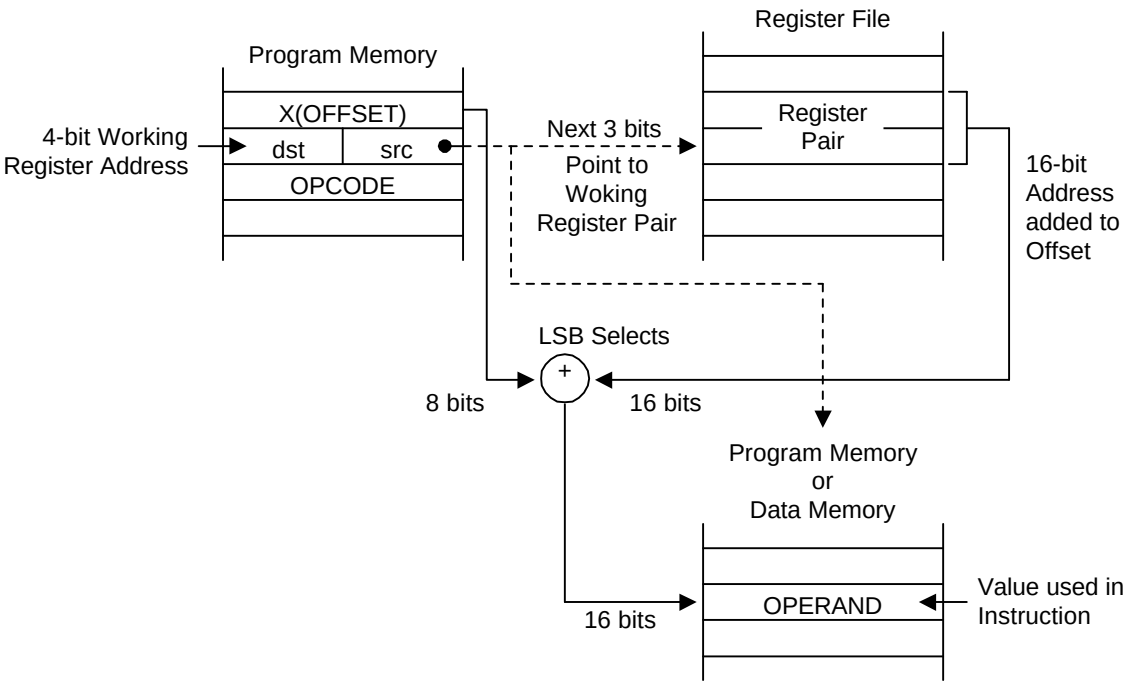


Figure 3-7. Indexed Addressing to Register File

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)

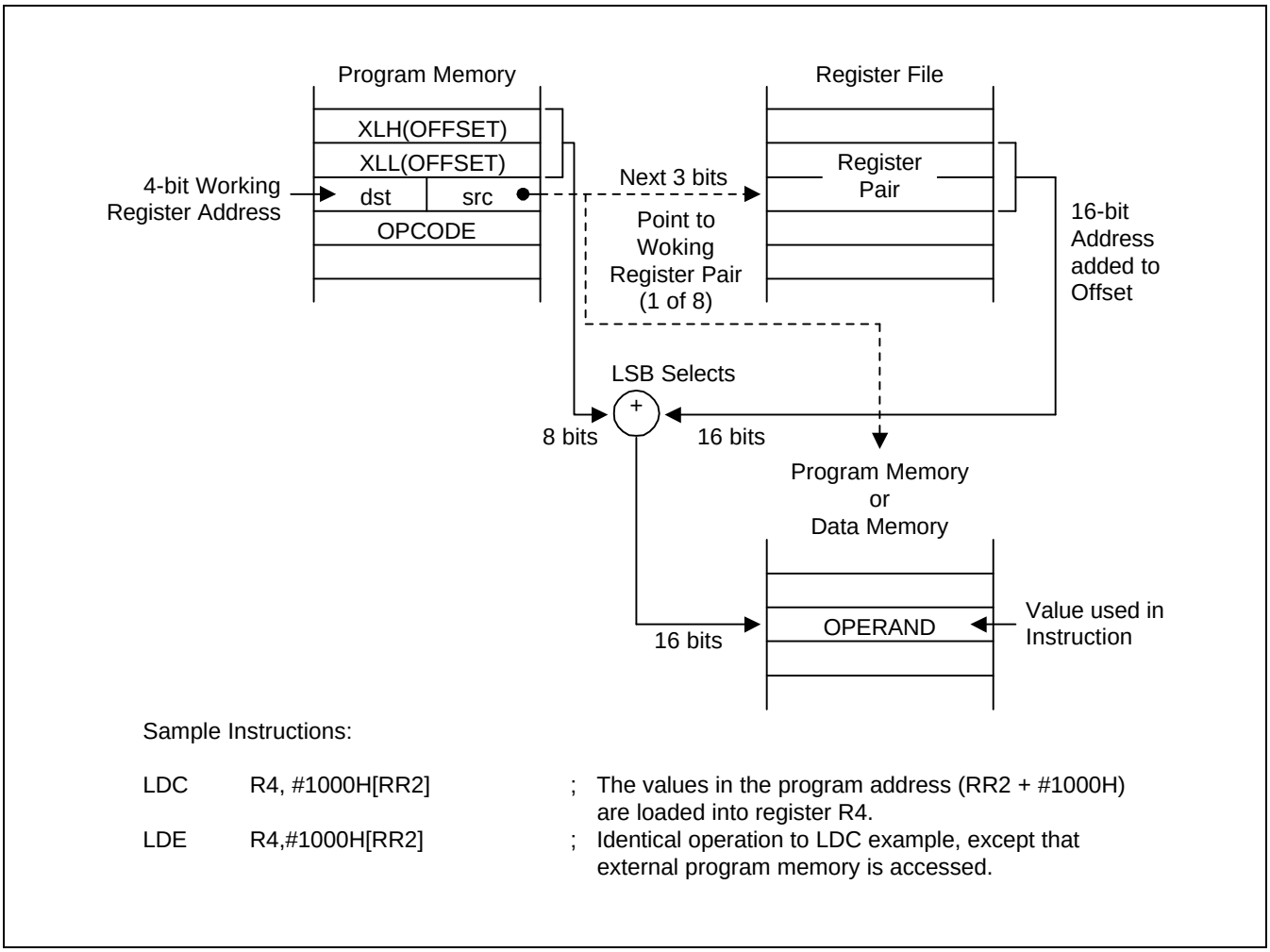
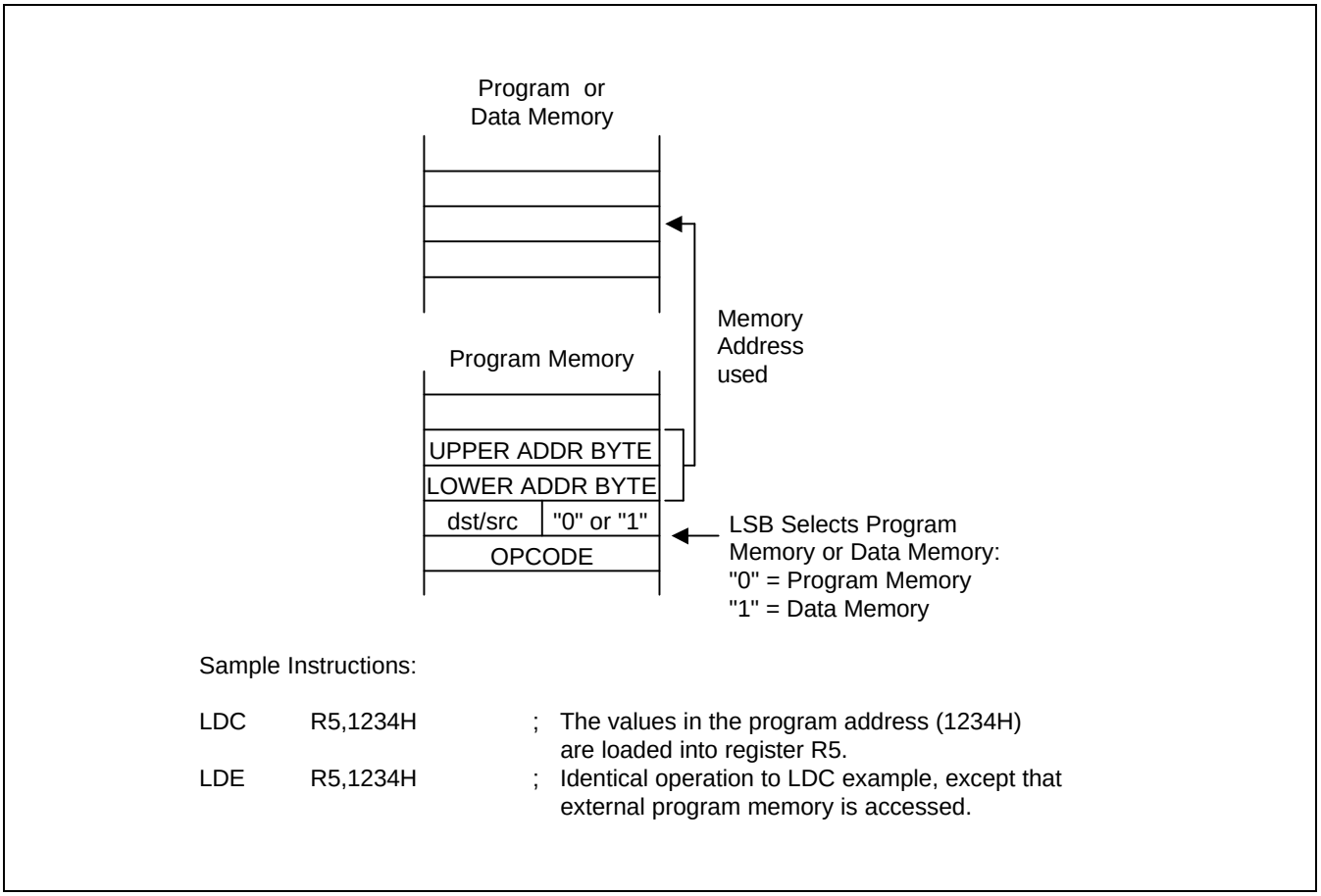


Figure 3-9. Indexed Addressing to Program or Data Memory with Long Offset

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.

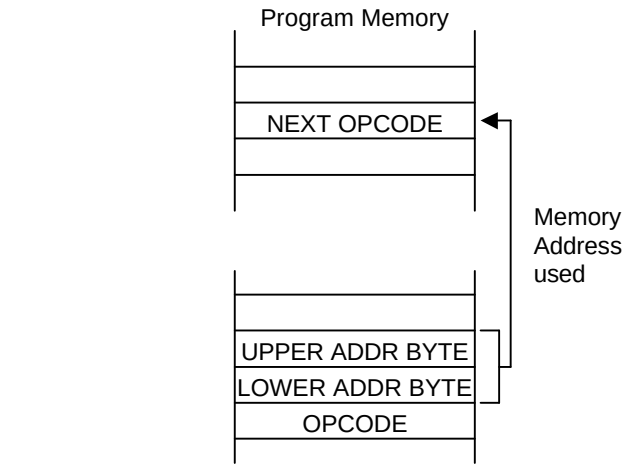


Sample Instructions:

```
LDC     R5,1234H        ;  The values in the program address (1234H)
                           are loaded into register R5.
LDE     R5,1234H        ;  Identical operation to LDC example, except that
                           external program memory is accessed.
```

Figure 3-10. Direct Addressing for Load Instructions

DIRECT ADDRESS MODE (Continued)



Sample Instructions:

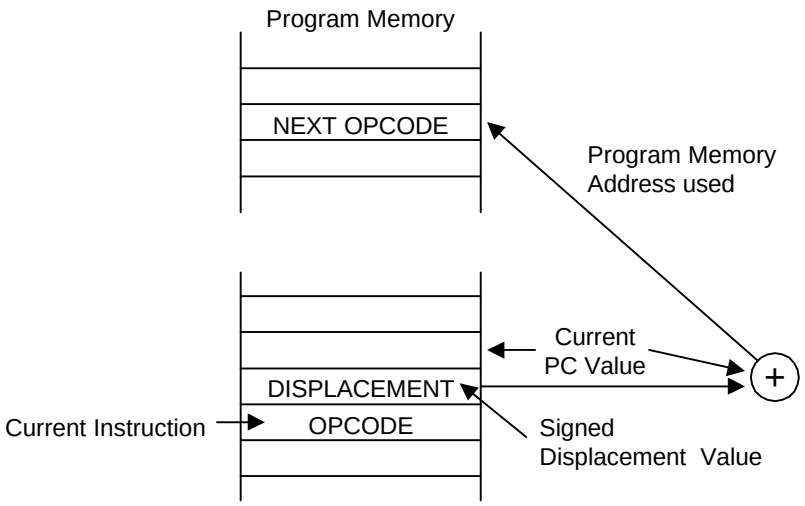
```
JP      C,JOB1      ;  Where JOB1 is a 16-bit immediate address
CALL    DISPLAY     ;  Where DISPLAY is a 16-bit immediate address
```

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.



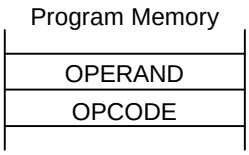
Sample Instructions:

```
JR      ULT,$+OFFSET      ;  Where OFFSET is a value in the range +127 to -128
```

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.



(The Operand value is in the instruction)

Sample Instruction:

```
LD      R0,#0AAH
```

Figure 3-13. Immediate Addressing

4 CONTROL REGISTERS

OVERVIEW

In this section, detailed descriptions of the S3C9424/C9428/P9428 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. Register Map and RESET Status

Register Name	Mnemonic	Address & Location		RESET Value (bit)							
		Address	R/W	7	6	5	4	3	2	1	0
Timer 0 counter register	T0CNT	D0H	R	0	0	0	0	0	0	0	0
Timer 0 data register	T0DATA	D1H	R/W	1	1	1	1	1	1	1	1
Timer 0 control register (high)	T0CONH	D2H	R/W	–	–	–	–	–	–	–	0
Timer 0 control register (low)	T0CONL	D3H	R/W	0	0	0	0	0	0	0	0
Clock control register	CLKCON	D4H	R/W	0	0	0	0	0	0	0	0
System flags register	FLAGS	D5H	R/W	x	x	x	x	–	–	–	–
Locations D6H-D8H are not mapped.											
Stack pointer register	SP	D9H	R/W	x	x	x	x	x	x	x	x
Locations DAH is reserved.											
MDS special register	MDSREG	DBH	R/W	0	0	0	0	0	0	0	0
Basic timer control register	BTCON	DCH	R/W	0	0	0	0	0	0	0	0
Basic timer counter	BTCNT	DDH	R	0	0	0	0	0	0	0	0
Test mode control register	FTSTCON	DEH	W	–	–	0	0	0	0	0	0
System mode register	SYM	DFH	R/W	–	–	–	–	–	0	0	0

NOTE: '–' is Not mapped, 'x'is Undefined

Table 4-1. Register Map and RESET Status (Continued)

Register Name	Mnemonic	Address Hex	R/W	RESET Value (bit)							
				7	6	5	4	3	2	1	0
Port 0 data register	P0	E0H	R/W	0	0	0	0	0	0	0	0
Port 1 data register	P1	E1H	R/W	–	–	–	–	0	0	0	0
Port 2 data register	P2	E2H	R/W	0	0	0	0	0	0	0	0
Port 3 data register	P3	E3H	R/W	–	–	–	–	0	0	0	0
Timer 1 control register	T1CON	E4H	R/W	0	0	0	0	0	0	0	0
Timer 1 data register	T1DATA	E5H	R/W	1	1	1	1	1	1	1	1
Port 0 control register (high)	P0CONH	E6H	R/W	0	0	0	0	0	0	0	0
Port 0 control register (low)	P0CONL	E7H	R/W	0	0	0	0	0	0	0	0
Port 0 pull-up resistor enable register	P0PUR	E8H	R/W	0	0	0	0	0	0	0	0
Port 1 control register	P1CON	E9H	R/W	0	0	0	0	0	0	0	0
Port 1 pull-up, pending register	P1PND	EAH	R/W	0	0	0	0	0	0	0	0
Port 2 control register (high)	P2CONH	EBH	R/W	0	0	0	0	0	0	0	0
Port 2 control register (low)	P2CONL	ECH	R/W	0	0	0	0	0	0	0	0
Port 2 pull-up resistor enable register	P2PUR	EDH	R/W	0	0	0	0	0	0	0	0
Port 3 control register	P3CON	EEH	R/W	0	0	0	0	0	0	0	0
SIO data register	SIODATA	EFH	R/W	0	0	0	0	0	0	0	0
SIO control register	SIOCON	F0H	R/W	0	0	0	0	0	0	0	0
SIO prescaler	SIOPS	F1H	R/W	0	0	0	0	0	0	0	0
IIC-bus clock control register	ICCR	F2H	R/W	0	0	0	0	0	0	0	0
IIC-bus clock/status register	ICSR	F3H	R/W	0	0	0	0	0	0	0	0
IIC-bus address register	IAR	F4H	R/W	x	x	x	x	x	x	x	–
IIC-bus Tx/Rx data shift register	IDSR	F5H	R/W	x	x	x	x	x	x	x	x
8-bit prescaler for buzzer output	BUZPS	F6H	R/W	0	0	0	0	0	0	0	0
A/D control register	ADCON	F7H	R/W	0	0	0	0	0	0	0	0
A/D converter data register (high)	ADDATAH	F8H	R	x	x	x	x	x	x	x	x
A/D converter data register (low)	ADDATAL	F9H	R	0	0	0	0	0	0	x	x
PWM 0 data register	PWM0	FAH	R/W	–	–	0	0	0	0	0	0
PWM 0 extension data register	PWM0EX	FBH	R/W	0	0	0	0	0	0	–	–
PWM 1 data register	PWM1	FCH	R/W	–	–	0	0	0	0	0	0
PWM 1 extension data register	PWM1EX	FDH	R/W	0	0	0	0	0	0	–	–
PWM control register	PWMCON	FEH	R/W	0	0	0	0	0	0	0	0
Zero crossing detection control register	ZCMOD	FFH	R/W	–	–	–	0	0	0	0	0

NOTE: '–' is Not mapped, 'x'is Undefined

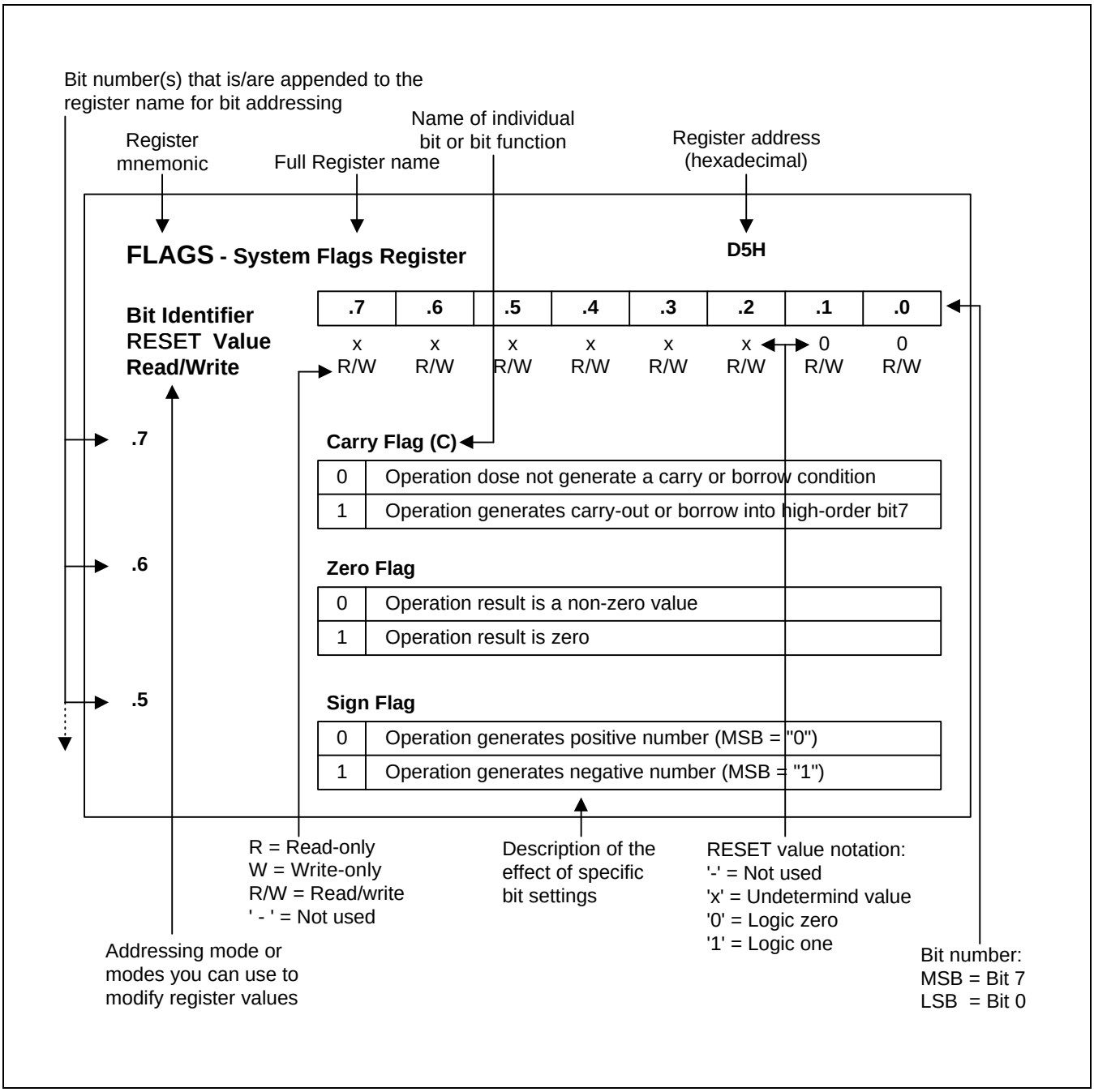


Figure 4-1. Register Description Format

ADCON — A/D Converter Control Register F7H

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 A/D Converter Input Pin Selection Bits

0	0	0	0	AD0 (P2.0)
0	0	0	1	AD1 (P2.1)
0	0	1	0	AD2 (P2.2)
0	0	1	1	AD3 (P2.3)
0	1	0	0	AD4 (P2.4)
0	1	0	1	AD5 (P2.5)
0	1	1	0	AD6 (P2.6)
0	1	1	1	AD7 (P2.7)
1	0	0	0	AD8 (P0.4)
1	0	0	1	AD9 (P0.5)
1	0	1	0	AD10 (P0.6)
1	0	1	1	AD11 (P0.7)
1	1	0	0	Internally connected to GND
1	1	0	1	Internally connected to GND
1	1	1	0	Internally connected to GND
1	1	1	1	Internally connected to AV_{REF}

.3 End-of-Conversion Status Bit

0	A/D conversion is in progress
1	A/D conversion complete

.2–.1 Clock Source Selection Bit

0	0	fosc/16
0	1	fosc/8
1	0	fosc/4
1	1	fosc/1

.0 Conversion Start Bit

0	No meaning
1	A/D conversion start

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 Function Enable Bit**

1	0	1	0	Disable watchdog timer function
Others				Enable watchdog timer function

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

0	0	fosc/4096
0	1	fosc/1024
1	0	fosc/128
1	1	Invalid setting

.1 **Basic Timer 8-bit Counter Clear Bit** (note)

0	No effect
1	Clear basic timer counter value

.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".

BUZPS — 6-Bit Prescaler for Buzzer Output 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 Buzzer Output Enable Bit

0	Disable buzzer output (buzzer off)
1	Enable buzzer output (buzzer on)

.6 Buzzer Clock Selection Bit

0	Divided by 256 (fx/256)
1	Divided by 64 (fx/64)

.5–.0 6-Bit Prescaler

0	0	0	0	0	0	divided by 2 [fx/(256 or 64)]
0	0	0	0	0	1	divided by 4 [fx/(256 or 64)]
0	0	0	0	1	0	divided by 6 [fx/(256 or 64)]
0	0	0	0	1	1	divided by 8 [fx/(256 or 64)]
• • •	• • •	• • •	• • •	• • •	• • •	divided by 2x(n+1) [fx/(256 or 64)]
1	1	1	1	1	1	divided by 128 [fx/(256 or 64)]

CLKCON — System Clock Control Register D4H

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

.7 Oscillator IRQ Wake-up Function Enable Bit

0	Enable IRQ for main system oscillator wake-up function
1	Disable IRQ for main system oscillator wake-up function

.6 and .5 Not used for S3C9424/C9428/P9428

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

0	0	Divide by 16 (fosc/16)
0	1	Divide by 8 (fosc/8)
1	0	Divide by 2 (fosc/2)
1	1	Non-divided clock (fosc) (2)

.2–.0 Not used for S3C9424/C9428/P9428

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. fosc means oscillator frequency

FLAGS — System Flags Register D5H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	x	x	x	x	x	x	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
1	Operation generates a carry-out or borrow into high-order bit 7

.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 $\leq +127$ or ≥ -128
1	Operation result is $> +127$ or < -128

.3–.0 Not used for S3C9424/C9428/P9428

ICCR — Multi-master IIC-Bus Clock Control 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 Acknowledgement Enable Bit

0	Acknowledgement disable mode
1	Acknowledgement enable mode

.6 Tx Clock Selection Bit

0	fosc/16
1	fosc/512

.5 Multi-master IIC-Bus Tx/Rx Interrupt Enable Bit

0	Disable interrupt
1	Enable interrupt

.4 Multi-master IIC-Bus Tx/Rx Interrupt pending Bit

0	When write “0” to this bit or when ICSR.4 is “0”
1	When 1-byte transmit/receive is terminated, general call or slave address match occurred, or arbitration lost

.3–.0 ICCR.3-0: Transmit Clock 4-Bit Prescaler Bits

SCL clock = IICLK/CCR<3:0> + 1
where, IICLK = fosc/16 when IICR.6 is “0”, IICLK = fosc/512 when ICCR.6 is “1”

ICSR — Multi-master IIC-Bus Control/Status Register F3H

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–.6 Master/Slave Tx/Rx Mode Selection Bits

0	0	Slave receiver mode (Default mode)
0	1	Slave transmitter mode
1	0	Master receiver mode
1	1	Master transmitter mode

.5 IIC-Bus Busy Bit

0	IIC-bus is not busy
0	Stop condition generation
1	IIC-bus is busy (when read)
1	Stop condition generation (when write)

.4 IIC-bus Interface Module Enable Bit

0	Disable IIC-bus data transmit/receive
1	Enable IIC-bus data transmit/receive

.3 Arbitration Lost Bit

This bit is set by H/W when the serial I/O interface, in master transmit mode, loses a bus arbitration procedure. In slave mode this flag is set to “1” when ICCR.5 is “1” and ICSR.2 is “0”

.2 Address Match Bit

0	When Start or Stop or Reset
1	When received slave address matches to IAR register or general call

.1 General Call Bit

0	When Start/Stop condition is generated
1	When received slave address is “00000000” (general call)

.0 Received Acknowledge Bit

0	ACK is received
1	ACK is not received

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–.6 **Port 0, P0.7/AD11/PWM0 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD11) ; Schmitt trigger input off
1	0	Push-pull output
1	1	Alternative function (PWM0 Output)

.5–.4 **Port 0, P0.6/AD10 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD10) ; Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

.3–.2 **Port 0, P0.5/AD9 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD9) ; Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

.1–.0 **Port 0, P0.4/AD10 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD8) ; Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

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–.6 **Port 0, P0.3/CLO Configuration Bits**

0	0	Schmitt trigger input
0	1	Alternative function; CLO ouput
1	0	Push-pull output
1	1	Open-drain output

.5–.4 **Port 0, P0.2/SI Configuration Bits**

0	0	Schmitt trigger input; SI input
0	1	Schmitt trigger input; SI input
1	0	Push-pull output
1	1	Open-drain output

.3–.2 **Port 0, P0.1/SO Configuration Bits**

0	0	Schmitt trigger input
0	1	Alternative function; SO output
1	0	Push-pull output
1	1	Open-drain output

.1–.0 **Port 0, P0.0/SCK Configuration Bits**

0	0	Schmitt trigger input; SCK input
0	1	Alternative function; SCK output
1	0	Push-pull output
1	1	Open-drain output

P0PUR — Port 0 Pull-up Resistor Enable Register 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 Port 0, P0.7 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.6 Port 0, P0.6 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.5 Port 0, P0.5 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.4 Port 0, P0.4 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.3 Port 0, P0.3 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.2 Port 0, P0.2 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.1 Port 0, P0.1 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.0 Port 0, P0.0 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

P1CON — Port 1 Control Register 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/INT1/PWM1 Configuration Bits**

0	0	Schmitt trigger input; INT1 interrupt disabled
0	1	Schmitt trigger input; Interrupt on falling edge
1	0	Push-pull output
1	1	Alternative function (PWM1 output)

.5 and .4 **Port 1, P1.2/INT0 Configuration Bits**

0	0	Schmitt trigger input; INT0 interrupt disabled
0	1	Schmitt trigger input; Interrupt on falling edge
1	0	Push-pull output
1	1	Schmitt trigger input; Interrupt on rising edge

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

0	0	Schmitt trigger
0	1	Schmitt trigger input
1	0	Push-pull output
1	1	Alternative function (BUZ output)

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

0	0	Schmitt trigger input (or T0 Capture input)
0	1	ZCD input; ZCD enable
1	0	Push-pull output
1	1	Alternative function (T0 output; match or PWM)

P1PND — Port 1 Interrupt Pending Register 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 Port 1, P1.3/INT1/PWM1 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.6 Port 1, P1.2/INT0 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.5 Port 1, P1.1/BUZ Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.4 Port 1, P1.0/T0/ZCD Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.3 Port 1,P1.3/INT1/PWM1 Open-drain Enable Bit

0	Push pull output mode
1	Open-drain output

.2 Port 1, P1.1/BUZ Open-drain Enable Bit

0	Push pull output mode
1	Open-drain output

.1 Port 1, P1.3/INT1 Interrupt Pending Bit

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

.0 Port 1, P1.2/INT0 Interrupt Pending Bit

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

P2CONH — Port 2 Control Register (High 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, P2.7/AD7/SCLK Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD7); Schmitt trigger input off
1	0	Push-pull output
1	1	Alternative function (IIC Clock pin); Open-drain type

.5 and .4 **Port 2, P2.6/AD6/SDAT Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD6); Schmitt trigger input off
1	0	Push-pull output
1	1	Alternative function (IIC Data pin); Open-drain type

.3 and .2 **Port 2, P2.5/AD5 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD5); Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

.1 and .0 **Port 2, P2.4/AD4 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD4); Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

P2CONL — Port 2 Control Register (Low Byte) 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 and .6 **Port 2, P2.3/AD3 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD3); Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

.5 and .4 **Port 2, P2.2/AD2 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD2); Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

.3 and .2 **Port 2, P2.1/AD1Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD1); Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

.1 and .0 **Port 2, P2.0/AD0 Configuration Bits**

0	0	Schmitt trigger input
0	1	A/D converter input (AD0); Schmitt trigger input off
1	0	Push-pull output
1	1	Open-drain output

P2PUR — Port 2 Pull-up Resistor Enable Register EDH

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 Port 2.7 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.6 Port 2.6 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.5 Port 2.5 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.4 Port 2.4 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.3 Port 2.3 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.2 Port 2.2 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.1 Port 2.1 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

.0 Port 2.0 Pull-up Resistor Enable Bit

0	Disable pull-up
1	Enable pull-up

P3CON — Port 3 Control Register EEH

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

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

0	0	Push-pull output
0	1	Push-pull output
1	0	Open-drain output; Pull-up resistor disable
1	1	Open-drain output; Pull-up resistor enable

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

0	0	Push-pull output
0	1	Push-pull output
1	0	Open-drain output; Pull-up resistor disable
1	1	Open-drain output; Pull-up resistor enable

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

0	0	Push-pull output
0	1	Push-pull output
1	0	Open-drain output; Pull-up resistor disable
1	1	Open-drain output; Pull-up resistor enable

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

0	0	Push-pull output
0	1	Push-pull output
1	0	Open-drain output; Pull-up resistor disable
1	1	Open-drain output; Pull-up resistor enable

PWMCON — PWM Control Register FEH

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–.6 PWM Input Clock Slection Bit

0	0	fosc/256
0	1	fosc/64
1	0	fosc/8
1	1	fosc/1

.5 PWM 1 Data Reload Interval Selection Bit

0	Reload from 12-bit up counter overflow
1	Reload from 6-bit up counter overflow

.4 PWM 0 Data Reload Interval Selection Bit

0	Reload from 12-bit up counter overflow
1	Reload from 6-bit up counter overflow

.3 PWM Counter Clear Bit

0	No effect
1	Clear 12-bit up counter (when write)

.2 PWM Counter Enable Bit

0	Stop counter
1	Start (Resume counting)

.1 PWM Overflow Interrupt Enable bit (12-bit Counter Overflow)

0	Disable interrupt
1	Enable interrupt

.0 PWM 12-Bit Counter Overflow Interrupt Pending Bit

0	No interrupt pending
0	*Clear pending bit (when write)*
1	Interrupt is pending

SIOCON — Serial I/O Module Control Registers F0H

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 SIO Shift Clock Selection Bit

0	Interval clock (P.S Clock)
1	External clock (SCK)

.6 Data Direction Control Bit

0	MSB-first mode
1	LSB-first mode

.5 SIO Mode Selection Bit

0	Receive-only mode
1	Transmit/Receive mode

.4 Shift Clock Edge Selection Bit

0	Tx at falling edges, Rx at rising edges.
1	Tx at rising edges, Rx at falling edges.

.3 SIO Counter Clear and Shift Start Bit

0	No action
1	Clear 3-bit counter and start shifting

.2 SIO Shift Operation Enable Bit

0	Disable shift and clock counter
1	Enable shift and clock counter

.1 SIO Interrupt Enable Bit

0	Disable SIO interrupt
1	Enable SIO interrupt

.0 SIO Interrupt Pending Bit

0	No interrupt pending
0	*Clear pending condition (when write)*
1	Interrupt is pending

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 S3C9424/C9428/P9428**

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

0	Disable all interrupt (DI instruction)
1	Enable all interrupt (EI Instruction)

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

0	0	page 0
0	1	page 1 **(Not used for S3C9424/C9428/P9428)**
1	0	page 2 **(Not used for S3C9424/C9428/P9428)**
1	1	page 3 **(Not used for S3C9424/C9428/P9428)**

NOTE: Following a reset, you enable global interrupt processing by executing an EI instruction (not by writing a “1” to SYM.2).

T0CONH — TIMER 0 Control Register (High Byte) D2H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0 (8)
RESET Value	–	–	–	–	–	–	–	0
Read/Write	–	–	–	–	–	–	–	R/W

.7–.1 **Not used for S3C9424/C9428/P9428**

.0 (8) **Timer 0 Overflow Interrupt Pending Bit (overflow interrupt)**

0	No interrupt pending *(when read)*
0	Clear Pending bit *(when write)*
1	Interrupt is pending *(when read)*

T0CONL — TIMER 0 Control Register (Low Byte) D3H

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 **Timer 0 Input Clock Selection Bits**

0	0	fosc/4096
0	1	fosc/256
1	0	fosc/8
1	1	fosc/1

.5 and .4 **Timer 0 Operating Mode Selection Bits**

0	0	Interval mode
0	1	Capture mode (capture on rising edge, counter running, OVF)
1	0	Capture mode (capture on falling edge, counter running, OVF)
1	1	PWM mode (OVF interrupt can occur)

.3 **Timer 0 Counter Clear Bit**

0	No effect
1	Clear the timer 0 counter (when write)

.2 **Timer 0 Overflow Interrupt Enable Bit**

0	Disable overflow interrupt
1	Enable overflow interrupt

.1 **Timer 0 Interrupt Enable Bit**

0	Disable interrupt
1	Enable interrupt

.0 **Timer 0 Interrupt Pending Bit (Capture or Match Interrupt)**

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

NOTE: When you write a “1” to T0CONL.3 the timer 0 counter is cleared. The bit is then cleared automatically to “0”.

T1CON — Timer 1 Control Register E4H

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 Timer 1 Counter Control Bit

0	Disable operation
1	Enable counter operation

.6 and .4 Timer 1 Input Clock Selection Bits

0	0	0	fosc/4096
0	0	1	fosc/1024
0	1	0	fosc/512
0	1	1	fosc/256
1	0	0	fosc/128
1	0	1	fosc/32

.3 Timer 1 Counter Automatic Clear Enable Bit

0	Disable
1	Enable ZCD clear signal to clear the timer 1 counter

.2 Timer 1 Counter Clear Enable Bit

0	No effect
1	Clear the timer 1 counter (when write)

.1 Timer 1 Interrupt Enable Bit

0	Disable interrupt
1	Enable interrupt

.0 Timer 1 Interrupt Pending Bit

0	No interrupt pending
0	Clear pending bit *(when write)*
1	Interrupt is pending

ZCMOD — Zero Crossing Detection Control Register FFH

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

.7-.5 **Not used for S3C9424/C9428/P9428**

.4 **ZCD Operation Enable Bit**

0	Disable operation
1	Enable operation

.3 and .2 **Interrupt Mode Selection Bits**

0	0	Interrupt on falling edge
0	1	Interrupt on rising edge
1	0	Interrupt on both edge
1	1	Not used

.1 **ZCD Interrupt Enable Bit**

0	Disable interrupt
1	Enable interrupt

.0 **ZCD Interrupt Pending Bit**

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

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 an interrupt vector which is assigned in ROM address 0000H.

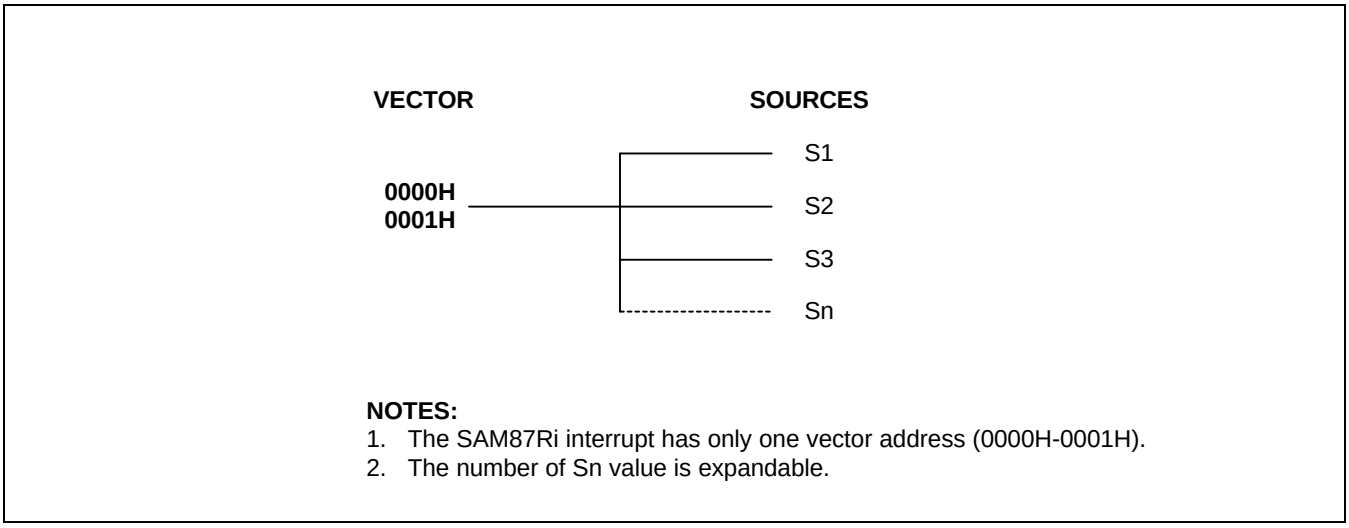


Figure 5-1. S3C9-Series Interrupt Type

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.

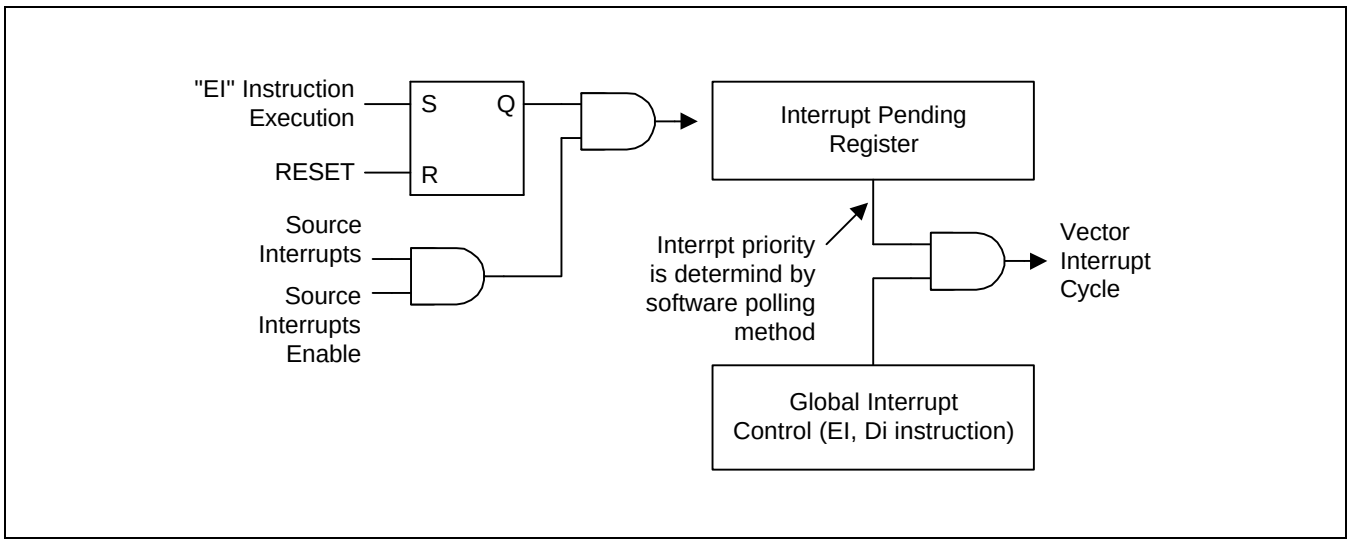


Figure 5-2. Interrupt Function Diagram

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.

S3C9424/C9428/P9428 INTERRUPT STRUCTURE

The S3C9424/C9428/P9428 microcontroller has nine peripheral interrupt sources:

— Timer 0 match/capture interrupt
— Timer 0 overflow interrupt
— Timer 1 match interrupt
— Zero-cross detection
— Two external interrupts for port 1, P1.2-P1.3
— SIO interrupt
— PWM overflow interrupt
— IIC-bus Tx/Rx interrupt

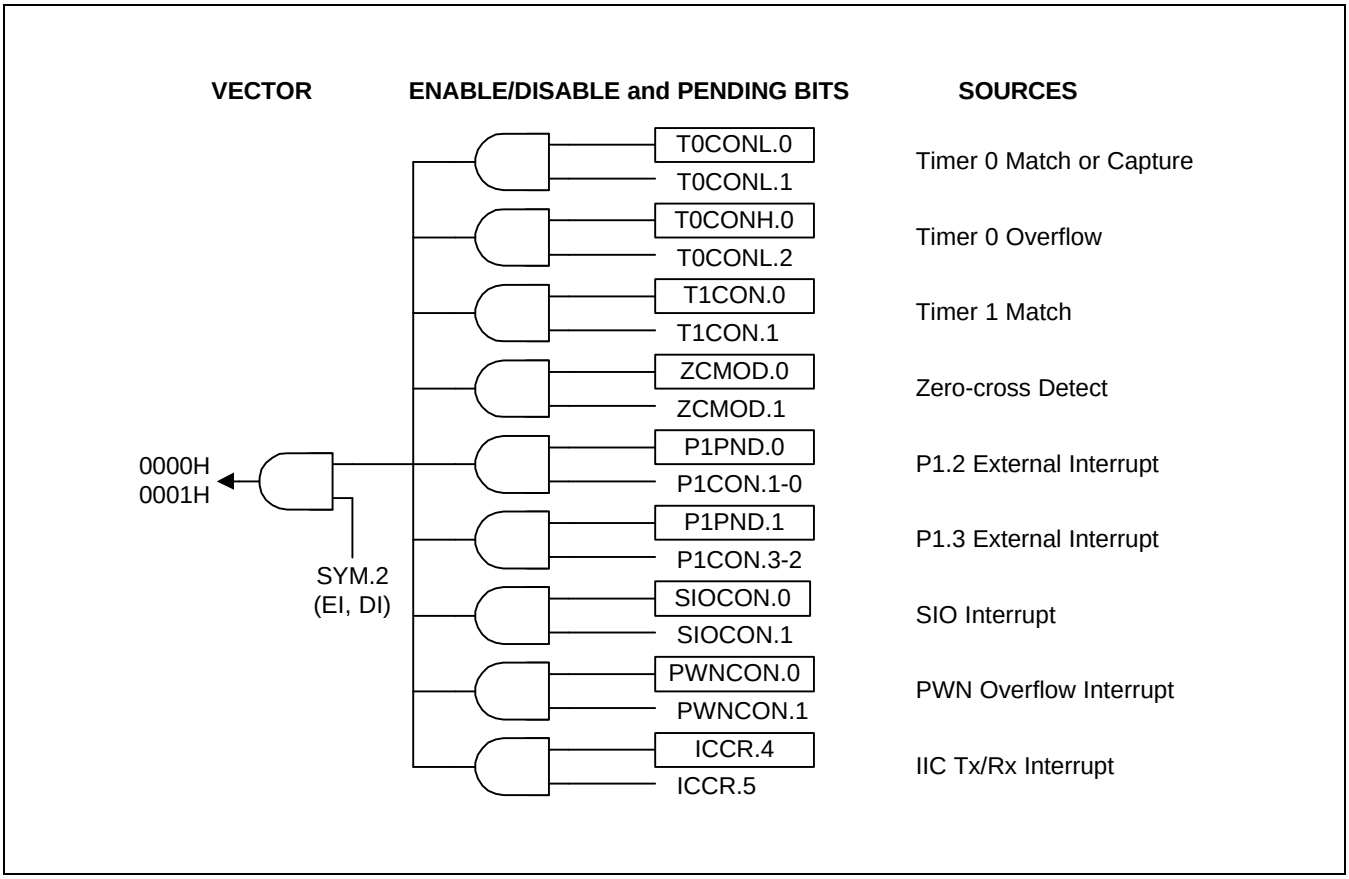


Figure 5-3. S3C9424/C9428/P9428 Interrupt Structure

7 CLOCK CIRCUIT

OVERVIEW

An RC oscillation source provides a typical 4 MHz clock for S3C9424/C9428/P9428. An internal capacitor supports the RC oscillator circuit. An external crystal or ceramic oscillation source provides a maximum 16 MHz clock. The X_{IN} and X_{OUT} pins connect the oscillation source to the on-chip clock circuit. Simplified RC oscillator and crystal/ceramic oscillator circuits are shown in Figures 7-1 and 7-2.

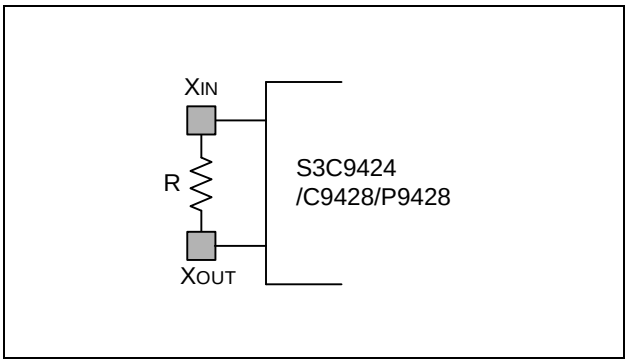


Figure 7-1. Main Oscillator Circuit (RC Oscillator with Internal Capacitor)

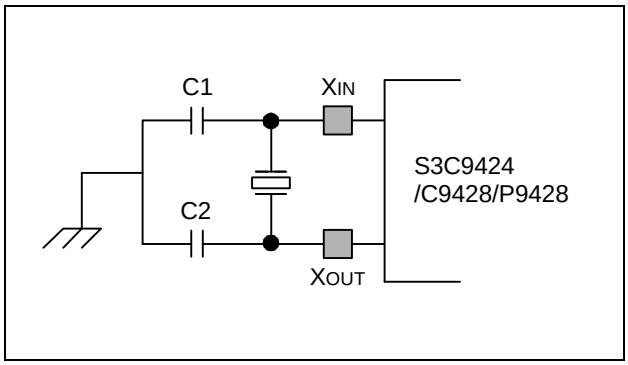


Figure 7-2. Main Oscillator Circuit (Crystal/Ceramic Oscillator)

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 S3C9424/C9428/P9428, INT0-INT1).

— 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 fosc, fosc/2 or fosc/8.

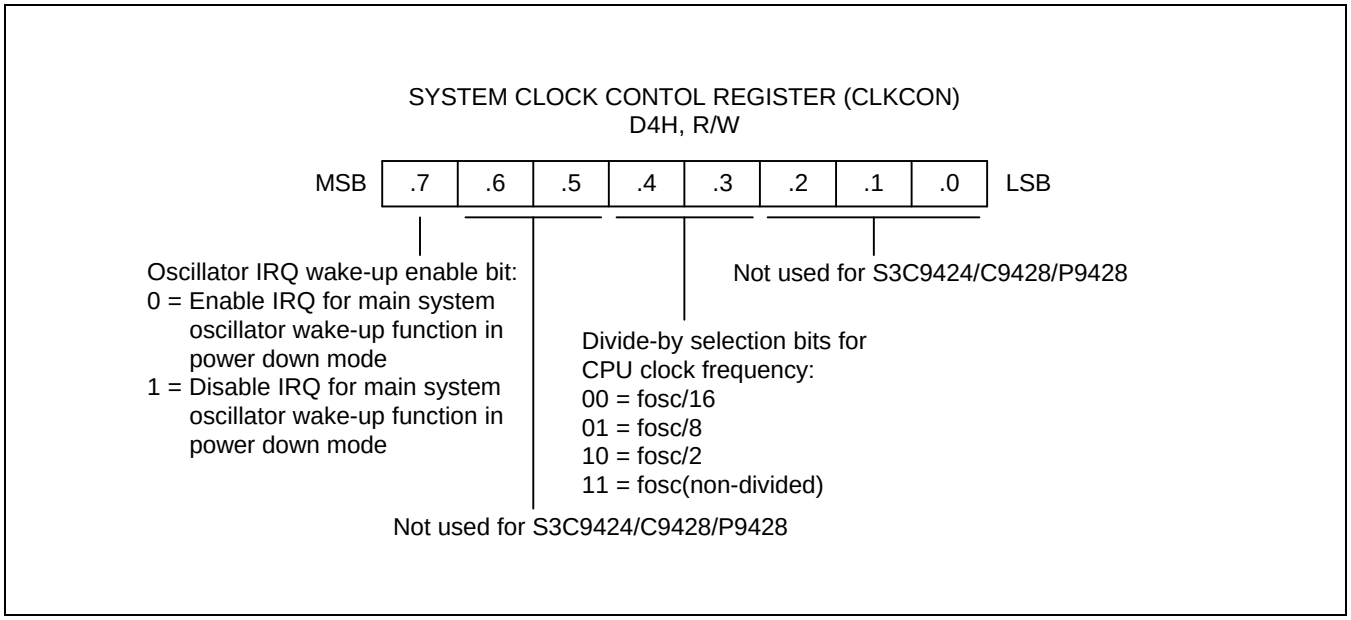
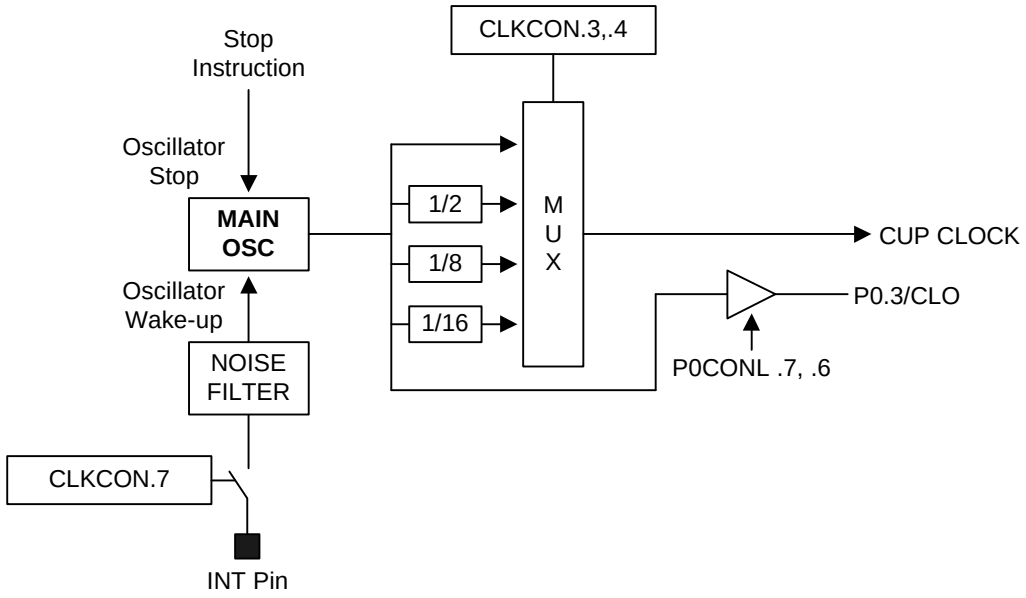


Figure 7-3. System Clock Control Register (CLKCON)



NOTE: An external interrupt with an RC-delay nosie fillter can be used to release Stop mode and "wake-up" the main ascillator . In the S3C9424/C9428/P9428, the INT0-INT1 external interrupts and ZCD interrupt are of this type.

Figure 7-4. System Clock Circuit Diagram

8 RESET and POWER-DOWN

SYSTEM RESET

OVERVIEW

The S3C9424/C9428/P9428 can be RESET in four ways:

— by power-on reset
— by the external reset input pin pulled low
— by the digital watchdog peripheral timing out
— by Low Voltage Detection (LVD)

During a power-on reset, the voltage at V_{DD} is High level 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 bring the S3C9424/C9428/P9428 into a known operating status. To ensure correct start-up, the user should take care that reset signal is not released before the V_{DD} level is sufficient to allow MCU operation at the chosen frequency.

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 6.55 ms ($\cong 2^{16}$/fosc, fosc = 10 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 MCU provides a watchdog timer function in order to ensure graceful recovery from software malfunction. If watchdog timer is not refreshed before an end-of-counter condition (overflow) is reached, the internal reset will be activated.

The on-chip Low Voltage Detector, features static Reset when supply voltage is below a reference value (Typ. 2.6 V). Thanks to this feature, external reset circuit can be removed while keeping the application safety. As long as the supply voltage is below the reference value, there is a internal and static RESET. The MCU can start only when the supply voltage rises over the reference value.

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.

MCU Initialization Sequence

The following sequence of events occurs during a reset operation:

— All interrupts are disabled.
— The watchdog function (basic timer) is enabled.
— Ports 0-1 are set to input mode and all pull-up resistors are disabled.
— Peripheral control and data registers are disabled and reset to their initial values (See Table 8-1).
— 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.

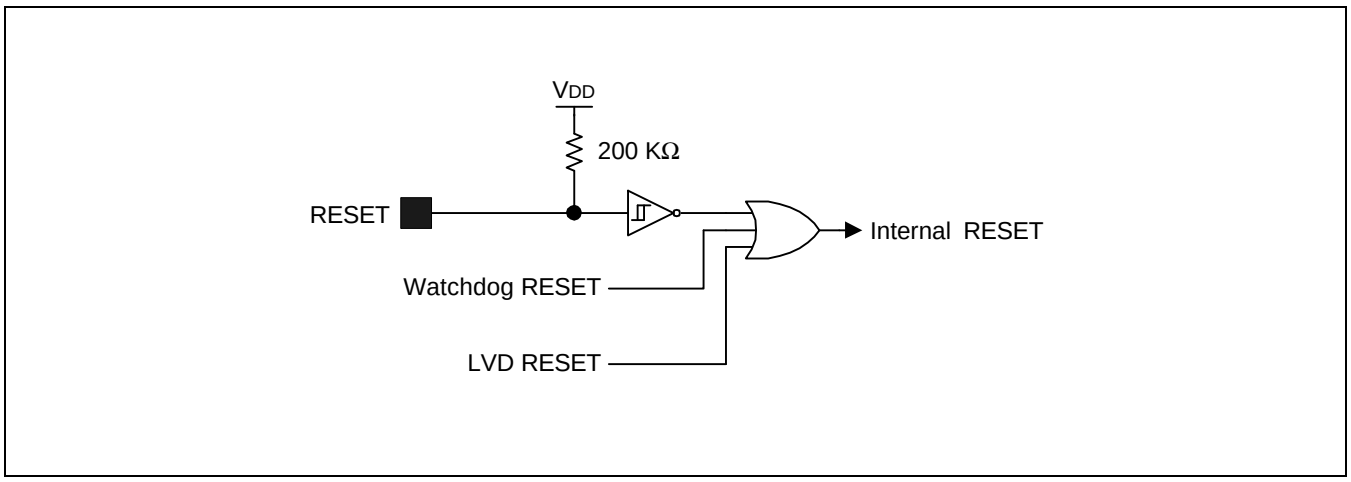


Figure 8-1. Reset Block Diagram

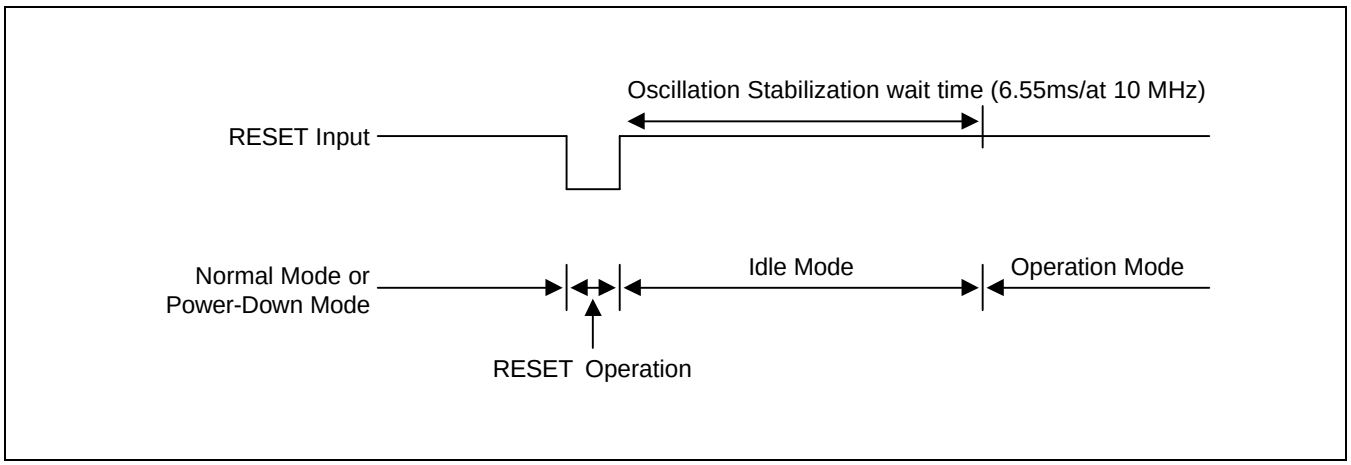


Figure 8-2. Timing for Oscillation Stabilization after RESET

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 5 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 (fosc/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-INT1 in the S3C9424/C9428/P9428 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 (fosc/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.

NOTES

1. 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.
2. Before enter the STOP or IDLE mode, the ZCD (P1CON) and ADC (P0CONH, P2CONH, P2CONL) must be disabled. Otherwise, the STOP or IDLE current will be increased significantly.

HARDWARE RESET VALUES

Table 8-1 lists the values for CPU and system registers, peripheral control registers, and peripheral data registers following a reset operation in normal operating mode.

— 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 Map and RESET Status

Register Name	Mnemonic	Address & Location		RESET Value (bit)							
		Address	R/W	7	6	5	4	3	2	1	0
Timer 0 counter register	T0CNT	D0H	R	0	0	0	0	0	0	0	0
Timer 0 data register	T0DATA	D1H	R/W	1	1	1	1	1	1	1	1
Timer 0 control register (high)	T0CONH	D2H	R/W	–	–	–	–	–	–	–	0
Timer 0 control register (low)	T0CONL	D3H	R/W	0	0	0	0	0	0	0	0
Clock control register	CLKCON	D4H	R/W	0	0	0	0	0	0	0	0
System flags register	FLAGS	D5H	R/W	x	x	x	x	–	–	–	–
Locations D6H-D8H are not mapped.											
Stack pointer register	SP	D9H	R/W	x	x	x	x	x	x	x	x
Locations DAH is reserved.											
MDS special register	MDSREG	DBH	R/W	0	0	0	0	0	0	0	0
Basic timer control register	BTCON	DCH	R/W	0	0	0	0	0	0	0	0
Basic timer counter	BTCNT	DDH	R	0	0	0	0	0	0	0	0
Test mode control register	FTSTCON	DEH	W	–	–	0	0	0	0	0	0
System mode register	SYM	DFH	R/W	–	–	–	–	–	0	0	0

NOTE: '–' is Not mapped, 'x'is Undefined

Table 8-1. Register Map and RESET Status (Continued)

Register Name	Mnemonic	Address	R/W	RESET Value (bit)							
		Hex		7	6	5	4	3	2	1	0
Port 0 data register	P0	E0H	R/W	0	0	0	0	0	0	0	0
Port 1 data register	P1	E1H	R/W	–	–	–	–	0	0	0	0
Port 2 data register	P2	E2H	R/W	0	0	0	0	0	0	0	0
Port 3 data register	P3	E3H	R/W	–	–	–	–	0	0	0	0
Timer 1 control register	T1CON	E4H	R/W	0	0	0	0	0	0	0	0
Timer 1 data register	T1DATA	E5H	R/W	1	1	1	1	1	1	1	1
Port 0 control register (high)	P0CONH	E6H	R/W	0	0	0	0	0	0	0	0
Port 0 control register (low)	P0CONL	E7H	R/W	0	0	0	0	0	0	0	0
Port 0 pull-up resistor enable register	P0PUR	E8H	R/W	0	0	0	0	0	0	0	0
Port 1 control register	P1CON	E9H	R/W	0	0	0	0	0	0	0	0
Port 1 pull-up, pending register	P1PND	EAH	R/W	0	0	0	0	0	0	0	0
Port 2 control register (high)	P2CONH	EBH	R/W	0	0	0	0	0	0	0	0
Port 2 control register (low)	P2CONL	ECH	R/W	0	0	0	0	0	0	0	0
Port 2 pull-up resistor enable register	P2PUR	EDH	R/W	0	0	0	0	0	0	0	0
Port 3 control register	P3CON	EEH	R/W	0	0	0	0	0	0	0	0
SIO data register	SIODATA	EFH	R/W	0	0	0	0	0	0	0	0
SIO control register	SIOCON	F0H	R/W	0	0	0	0	0	0	0	0
SIO prescaler	SIOPS	F1H	R/W	0	0	0	0	0	0	0	0
IIC-bus clock control register	ICCR	F2H	R/W	0	0	0	0	0	0	0	0
IIC-bus clock/status register	ICSR	F3H	R/W	0	0	0	0	0	0	0	0
IIC-bus address register	IAR	F4H	R/W	x	x	x	x	x	x	x	–
IIC-bus Tx/Rx data shift register	IDSR	F5H	R/W	x	x	x	x	x	x	x	x
8-bit prescaler for buzzer output	BUZPS	F6H	R/W	0	0	0	0	0	0	0	0
A/D control register	ADCON	F7H	R/W	0	0	0	0	0	0	0	0
A/D converter data register (high)	ADDATAH	F8H	R	x	x	x	x	x	x	x	x
A/D converter data register (low)	ADDATAL	F9H	R	0	0	0	0	0	0	x	x
PWM 0 data register	PWM0	FAH	R/W	–	–	0	0	0	0	0	0
PWM 0 extension data register	PWM0EX	FBH	R/W	0	0	0	0	0	0	–	–
PWM 1 data register	PWM1	FCH	R/W	–	–	0	0	0	0	0	0
PWM 1 extension data register	PWM1EX	FDH	R/W	0	0	0	0	0	0	–	–
PWM control register	PWMCON	FEH	R/W	0	0	0	0	0	0	0	0
Zero crossing detection control register	ZCMOD	FFH	R/W	–	–	–	0	0	0	0	0

NOTE: '–' is Not mapped, 'x'is Undefined

☞ PROGRAMMING TIP — Sample S3C9424/C9428/P9428 Initialization Routine

The following sample program suggests how to program the initial program settings for

```
;--------------<< Interrupt Vector Address >>

            ORG         0000H

            VECTOR      00H,INT_4208        ;  S3C9428 has only one interrupt vector

            ORG         0100H
INITIAL:

            LD          SYM, #00H           ;  Global/Fast interrupt disable  -> SYM
            LD          BTCON, #10100010B   ;  Watch-dog disable
            LD          CLKCON, #00011000B  ;  non-divided CPU clock
            LD          SP, #0C0H           ;  9428 → 00~BF (After decrease, push data)

;--------------<< Port Initialization >>

            LD          P0CONH, #0          ;  input
            LD          P0CONL, #0          ;  input
            LD          P0PUR, #0FFH        ;  pull-up enable
            LD          P1CON, #50H         ;  input, EXT.INT enable
            LD          P1PND, #0F0H        ;  pull-up enable
            LD          P2CONH, #0          ;  input
            LD          P2CONL, #0          ;  input
            LD          P2PUR, #0FFH        ;  pull-up enable
            LD          P3CON, #0           ;  push-pull output

;--------------<< Timer 0 Setting >>

            LD          T0DATA, #41H        ;  interrupt interval  --1.667msec (10MHz base)
            LD          T0CONL, #01000010B  ;  Timer 0 match output

;--------------<< RAM Area Clear >>

            LD          R0, #0              ;  RAM clear area setting

RAM_CLR:

            CLR         @R0
            INC         R0
            CP          R0, #0BFH           ;  general register area →  00H~BFH
            JR          ULE, RAM_CLR
```

☞ **PROGRAMMING TIP — Sample S3C9424/C9428/P9428 Initialization Routine (Continued)**

```
;--------------<< Initialize Other Register >>

                •
                •
                •
                EI

;--------------<< Main Loop >>

MAIN:

                NOP
                LD          R0, #33H
                •
                •
                •
                CALL        SUB_ROUTINE0        ;   subroutine call
                •
                •
                •
                CALL        SUB_ROUTINE1        ;   subroutine call
                •
                •
                •
                JP          MAIN

;--------------<< Subroutine >>

SUB_ROUTINE0:

                NOP
                •
                •
                •
                RET

SUB_ROUTINE1:

                NOP
                •
                •
                •
                RET
```

☞ PROGRAMMING TIP — Sample S3C9424/C9428/P9428 Initialization Routine (Continued)

```
;--------------<< Interrupt Service Routine >>

INT_4208:

            LD      R0, T0CONL          ;  S3C9428 has just one interrupt vector
            AND     R0, #00000011B      ;  only Timer 0 match interrupt enable
            CP      R0, #00000011B
            JP      EQ, INT_TIMER0      ;  T0CON's pending bit & INT. enable bit check

INT_TIMER0:

            AND     T0CONL, #11111110B  ;  pending clear
            •
            •
            •
            IRET
            •
            •
            •
            END
```

9 I/O PORTS

OVERVIEW

The S3C9424/C9428/P9428 has four I/O ports (0–3): 32-SOP type, with 24 I/O pins total and 28-SOP type, with 20 I/O pins total. You access these ports directly by writing or reading port data register addresses.

All Ports can be configured as LED drive. (High current output: typical 10 mA)

Table 9-1. S3C9424/C9428/P9428 Port Configuration Overview

Port	Function Description	Programmability
0	Bit-programmable I/O port for Schmitt trigger input or push-pull, open-drain out put. Pull-up resistors are assignable by software.	Bit
1	Bit-programable I/O port for Schmitt trigger input or push-pull output. Pull-up resistors are assignable by software. Port1 pins can also be used as alternative function.	Bit
2	Bit-programmable I/O port for Schmitt trigger input or push-pull, open drain output. Pull-up resistors are assignable by software. Port2 pins can also be used as A/D converter input.	Bit
3	Push-pull or open-drain output port. Pull-up resistors are assignable by software.	Bit

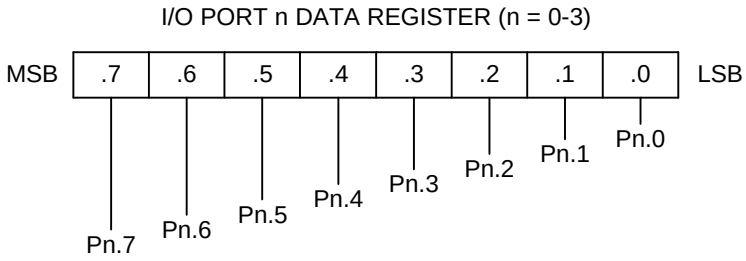
PORT DATA REGISTERS

Table 9-2 gives you an overview of the port data register names, locations, and addressing characteristics. Data registers for ports 0–3 have the structure shown in Figure 9-1.

Table 9-2. Port Data Register Summary

Register Name	Mnemonic	Hex	R/W
Port 0 data register	P0	E0H	R/W
Port 1 data register	P1	E1H	R/W
Port 2 data register	P2	E2H	R/W
Port 3 data register	P3	E3H	R/W

NOTE: A reset operation clears the P0-P3 data register to "00H".



NOTES:
1. Eight bits of the data register are used in the port 0 and port 2.
2. Only lower four bits of the data register are used in the port 1.
3. Only lower four bits of the data register are used in the port 3 (32-SOP type).

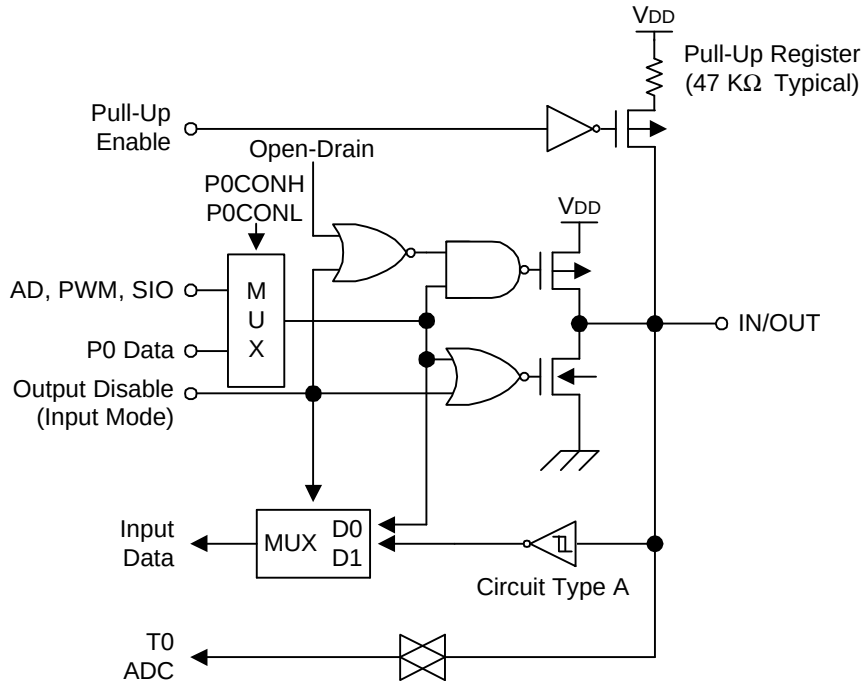
Figure 9-1. Port Data Register Format

PORT 0

Port 0 is a bit-programmable, general-purpose, I/O ports. You can select normal input or push-pull, open drain output mode. In addition, you can configure a pull-up resistor to individual pins using control register settings.

You access port 0 directly by writing or reading the corresponding port data register, P0 (E0H). A reset clears the port control register, P0CONH and P0CONL, to "00H" configuring port 0 pins as normal inputs.

One addition register is used to control Port 0: P0PUR (E8H).



NOTE: I/O pins have protection diodes through and V_{DD} and V_{SS}.

Mode	Input Data
Output	D0
Input	D1

Figure 9-2. Port 0 Circuit Diagram

PORT 0 CONTROL REGISTERS (HIGH BYTE)
E6H, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7-.6] Port 0, P0.7/AD11/PWM0 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD11); Schmitt trigger input off
10 = Push-pull output
11 = Alternative function (PWM0 Output)

[.5-.4] Port 0, P0.6/AD10 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD10); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

[.3-.2] Port0, P0.5/AD9 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD9); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

[.1-.0] Port0, P0.4/AD8 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD8); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

Figure 9-3. Port 0 High-Byte Control Register (P0CONH)

PORT 0 CONTROL REGISTERS (LOW BYTE)
E7H, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7-.6] Port 0, P0.3/CLO Configuration Bits
00 = Schmitt trigger input
01 = Alternative function; CLO output
10 = Push-pull output
11 = Open-drain output

[.5-.4] Port 0, P0.2/SI Configuration Bits
00 = Schmitt trigger input; SI Input
01 = Schmitt trigger input; SI Input
10 = Push-pull output
11 = Open-drain output

[.3-.2] Port0, P0.1/SO Configuration Bits
00 = Schmitt trigger input
01 = Alternative function; SO Output
10 = Push-pull output
11 = Open-drain output

[.1-.0] Port0, P0.0/SCK Configuration Bits
00 = Schmitt trigger input; SCK Input
01 = Alternative function; SCK Output
10 = Push-pull output
11 = Open-drain output

Figure 9-4. Port 0 Low-Byte Control Register (P0CONL)

PORT 0 PULL-UP RESISTOR ENABLE REGISTERS
E8H, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7] Port 0.7 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.6] Port 0.6 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.5] Port 0.5 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.4] Port 0.4 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.3] Port 0.3 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.2] Port 0.2 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.1] Port 0.1 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.0] Port 0.0 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

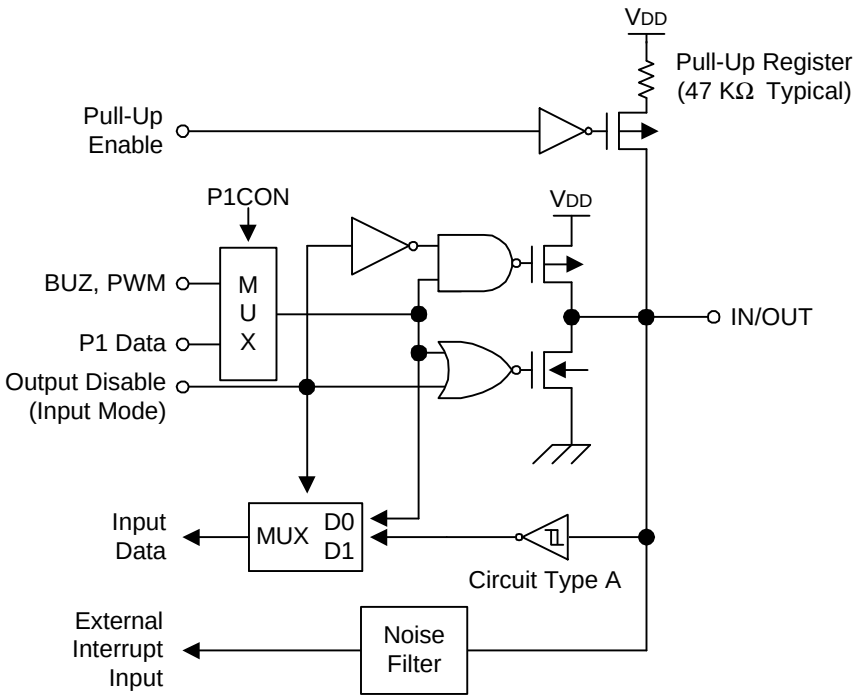
Figure 9-5. Port 0 Pull-up Enable Control Registers (P0PUR)

PORT 1

Port 1, is a 4-bit I/O port with individually configurable pins. It can be used for general I/O port (Schmitt trigger input mode or push-pull output mode). You can also use port1 as special input (ZCD) or output (BUZ, PWM). In addition, you can configure a pull-up resistor to individual pin using control register settings.

In normal operating mode, a reset clears P1CON to "00H", configuring P1.0-P1.3 as normal Schmitt trigger inputs, but you can also configure P1CON to "0FFH" for alternative functions.

You address port 1 bits directly by writing or reading the port 1 data register, P1 (E1H). The port 1 control register, P1CON is located at addresses E9H.



NOTE: I/O pins have protection diodes through and V_{DD} and V_{SS}.

Mode	Input Data
Output	D0
Input	D1

Figure 9-6. Port 1 Circuit Diagram

PORT 1 CONTROL REGISTERS
E9H, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7-.6] Port 1, P1.3/PWM1/INT1 Configuration Bits
00 = Schmitt trigger input; INT1 interrupt disabled
01 = Schmitt trigger input; Interrupt on falling edge
10 = Push-pull output
11 = Alternative function (PWM1 Output)

[.5-.4] Port 1, P1.2/INT0 Configuration Bits
00 = Schmitt trigger input; INT0 interrupt disabled
01 = Schmitt trigger input; Interrupt on falling edge
10 = Push-pull output
11 = Schmitt trigger input; Interrupt on rising edge

[.3-.2] Port1, P1.1/BUZ Configuration Bits
00 = Schmitt trigger input;
01 = Schmitt trigger input;
10 = Push-pull output
11 = Alternative function (BUZ Output)

[.1-.0] Port1, P1.0/T0/ZCD Configuration Bits
00 = Schmitt trigger input (or T0 Capture input)
01 = ZCD input; ZCD Enable
10 = Push-pull output
11 = Alternative function (T0 Output; match or PWM)

Figure 9-7. Port 1 Control Registers (P1CON)

PORT 1 INTERRUPT PENDING REGISTERS
EAH, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7] Port 1.3/INT1/PWM1, Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.6] Port 1.2/INT0, Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.5] Port 1.1/BUZ, Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.4] Port 1.0/T0/ZCD, Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.3] Port 1.3/INT1/PWM1, Open-drain Enable Bit
0 = Push pull output mode
1 = Open-drain output mode

[.2] Port 1.1/BUZ, Open-drain Enable Bit
0 = Push pull output mode
1 = Open-drain output mode

[.1] Port 1.3/INT1, Interrupt Pending Bit
0 = No interrupt pending (when read)
0 = Clear pending bit (when write)
1 = Interrupt is pending (when read)/No effect (when write)

[.0] Port 1.2/INT0 Interrupt Pending Bit
0 = No interrupt pending (when read)
0 = Clear pending bit (when write)
1 = Interrupt is pending (when read)/No effect (when write)

Figure 9-8. Port 1 Interrupt Pending Registers (P1PND)

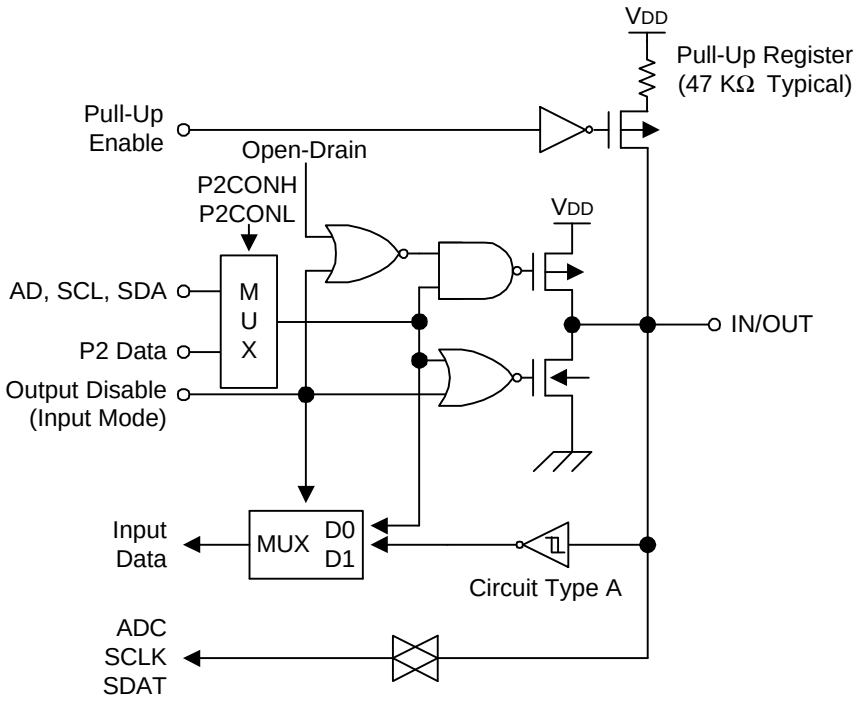
PORT 2

Port 2 is a 8-bit I/O port with individually configurable pins. It can be used for general I/O port (Schmitt trigger input mode or push-pull output mode or N-channel open-drain output mode). You can also use port 2 pins as A/D inputs. In addition, you can configure a pull-up resistor to individual pins using control register settings.

In normal operating mode, a reset clears P2CONH and P2CONL to "00H", configuring P2.0-P2.7 as normal Schmitt trigger inputs.

You address port 2 bits directly by writing or reading the port 2 data register, P2 (E2H). The port 2 control register, P2CONH is located at addresses EBH and P2CONL at ECH.

One additional register is used to control Port 2: P2PUR (EDH). By setting port 2 open-drain and pull-up resistor enable register, P2PUR, you can configure specific pins as open-drain or push-pull output.



NOTE: I/O pins have protection diodes through and V_{DD} and V_{SS}.

Mode	Input Data
Output	D0
Input	D1

Figure 9-9. Port 2 Circuit Diagram

PORT 2 CONTROL REGISTERS (HIGH BYTE)
EBH, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7-.6] Port2, P2.7/AD7/SCLK Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD7); Schmitt trigger input off
10 = Push-pull output
11 = Alternative function (IIC Clock pin):open-drain type

[.5-.4] Port 2, P2.6/AD6/SDAT Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD6); Schmitt trigger input off
10 = Push-pull output
11 = Alternative function (IIC Data pin):open-drain type

[.3-.2] Port2, P2.5/AD5 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD5); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

[.1-.0] Port2, P2.4/AD4 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD4); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

Figure 9-10. Port 2 High-Byte Control Registers (P2CONH)

PORT 2 CONTROL REGISTERS (LOW BYTE)
ECH, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7-.6] Port2, P2.3/AD3 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD3); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

[.5-.4] Port 2, P2.2/AD2 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD2); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

[.3-.2] Port2, P2.1/AD1 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD1); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

[.1-.0] Port2, P2.0/AD0 Configuration Bits
00 = Schmitt trigger input
01 = A/D converter input (AD0); Schmitt trigger input off
10 = Push-pull output
11 = Open-drain output

Figure 9-11. Port 2 Low-Byte Control Register (P2CONL)

PORT 2 PULL-UP RESISTOR ENABLE REGISTERS
EDH, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7] Port 2.7 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.6] Port 2.6 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.5] Port 2.5 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.4] Port 2.4 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.3] Port 2.3 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.2] Port 2.2 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

[.1] Port 2.1 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

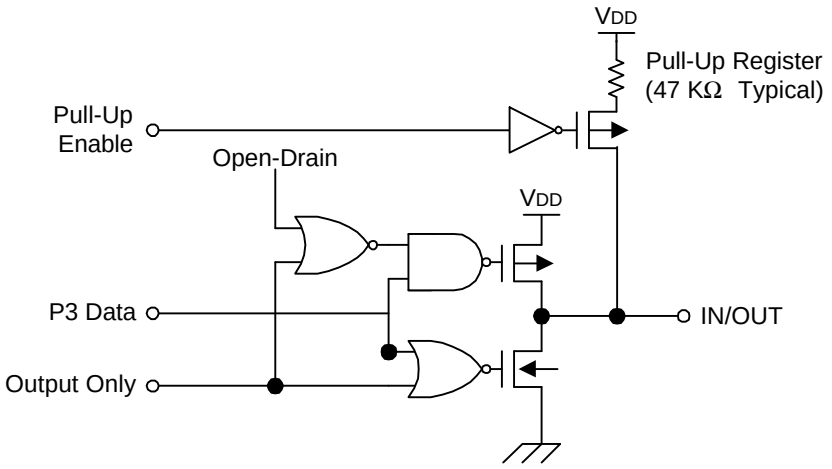
[.0] Port 2.0 Pull-up Resistor Enable Bit
0 = Disable pull-up
1 = Enable pull-up

Figure 9-12. Port 2 Pull-up Resistor Enable Register (P2PUR)

PORT 3

Port 3 is a 4-bit I/O port with individually configurable pins. It can be used only output port. In addition, you can configure a pull-up register to individual pins using control register settings.

In normal operating mode, reset clears P3CON to "00H", configures P3.0-P3.3 push-pull output mode. Using the P3CON register (EEH), you can alternatively configure the port 3 pins as push-pull output, or as open-drain output.



NOTE: I/O pins have protection diodes through and V_{DD} and V_{SS}.

Figure 9-13. Port 3 Circuit Diagram

PORT 3 CONTROL REGISTERS
EEH, R/W

MSB	.7	.6	.5	.4	.3	.2	.1	.0	LSB

[.7-.6] Port 3, P3.3 Configuration Bits
00 = Push-pull output
01 = Push-pull output
10 = Open-drain output: pull-up resistor disable
11 = Open-drain output: pull-up resistor enable

[.5-.4] Port 3, P3.2 Configuration Bits
00 = Push-pull output
01 = Push-pull output
10 = Open-drain output: pull-up resistor disable
11 = Open-drain output: pull-up resistor enable

[.3-.2] Port3, P3.1 Configuration Bits
00 = Push-pull output
01 = Push-pull output
10 = Open-drain output: pull-up resistor disable
11 = Open-drain output: pull-up resistor enable

[.1-.0] Port3, P3.0 Configuration Bits
00 = Push-pull output
01 = Push-pull output
10 = Open-drain output: pull-up resistor disable
11 = Open-drain output: pull-up resistor enable

Figure 9-14. Port 3 Control Registers (P3CON)

☞ PROGRAMMING TIP – Configuring I/O Port Pins to Specification

The following sample program shows how to configure the S3C9424/C9428/P9428 I/O ports to specification

Program comments show the effect of the settings:

```
•
•
•
LD      P0CONH, #01101010B    ;  0.7 - AD input
                              ;  0.6–0.4 - push-pull output
LD      P0CONL, #10010101B    ;  0.3 - push-pull output
                                 0.2–0.0 - SIO setting
LD      P1CON, #10101010B     ;  1.3–1.0 - push-pull output
LD      P2CONH, #11111010B    ;  2.7, 2.6 - IIC setting
                              ;  2.5, 2.4 - push-pull output
LD      P2CONL, #10101010B    ;  2.0–2.3 - push-pull output
LD      P3CON, #00000000B     ;  3.3–3.0 - push-pull output
•
•
•
```

10 BASIC TIMER and TIMERS

MODULE OVERVIEW

The S3C9424/C9428/P9428 has three default timers: an 8-bit *basic timer,* one 8-bit general-purpose timer/counter, called *timer 0,* and one 8-bit timer/counter for the zero-crossing detection circuit called *timer 1*.

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 three operating modes, one of which you select by an appropriate T0CONL setting:

— Interval timer mode
— Capture input mode
— 8-bit PWM output mode

Timer 0 has the following functional components:

— Clock frequency divider (fosc divided by 4096, 256, 8, or 1) with multiplexer
— 8-bit counter (T0CNT), 8-bit comparator, and 8-bit data register (T0DATA)
— I/O pin (P1.0, T0 match) for timer 0 match/PWM output or capture input
— Timer 0 overflow interrupt (T0OVF) and match interrupt (T0INT) generation
— Timer 0 control registers, T0CONH and T0CONL (D2H and D3H respectively)

Timer 1

Timer 1 has one operating mode, interval timer mode. You can clear the timer 1 counter by appropriate setting of T1CON register. If T1CON.3 is set to “1”, T1CNT is cleared by the ZCD edge detection.

Timer 1 has the following components:

— Clock frequency divider (f_{OSC} divided by 4096, 1024, 512, 256, 128, or 32)
— 8-bit counter (T1CNT), 8-bit comparator, and a 8-bit data register (T1DATA)
— Timer 1 control register, T1CON

BASIC TIMER (BT)

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 fosc/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 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.

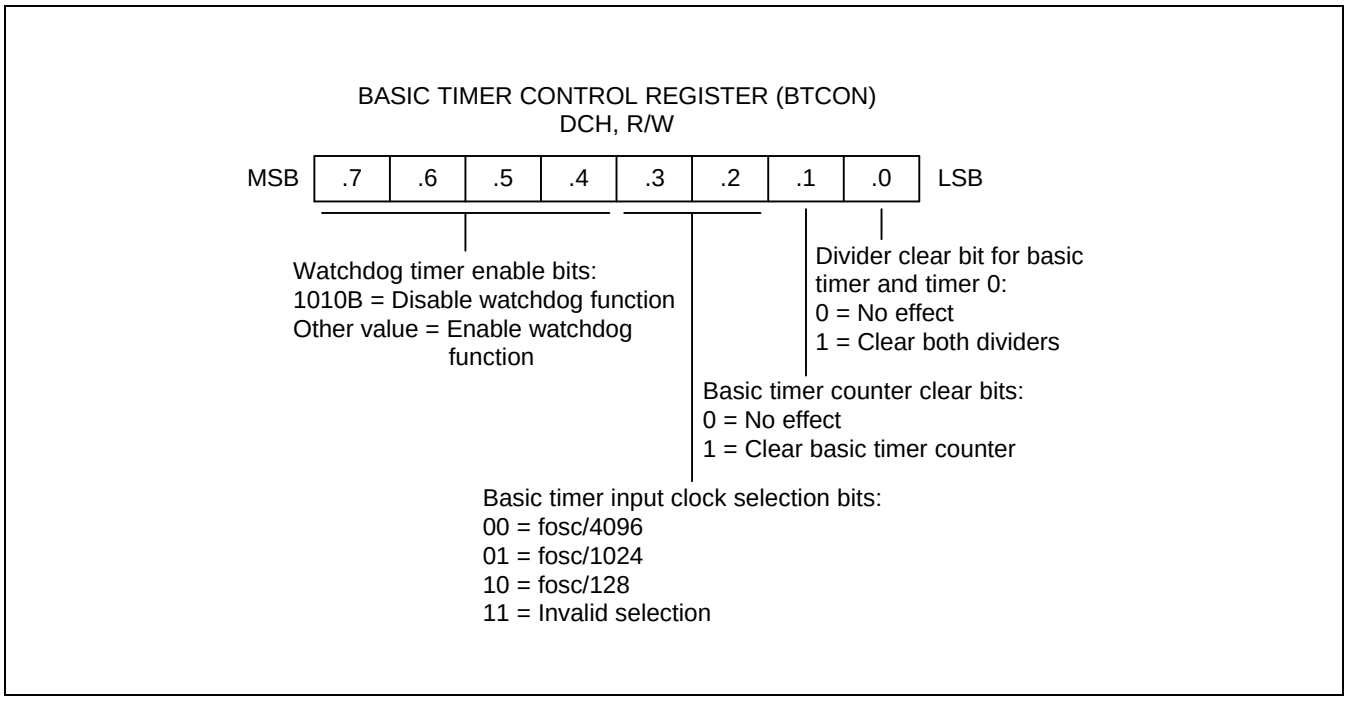


Figure 10-1. Basic Timer Control Register (BTCON)

BASIC TIMER FUNCTION DESCRIPTION

Watchdog Timer Function

You can program the basic timer overflow signal (BTOVF) 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 overflows 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

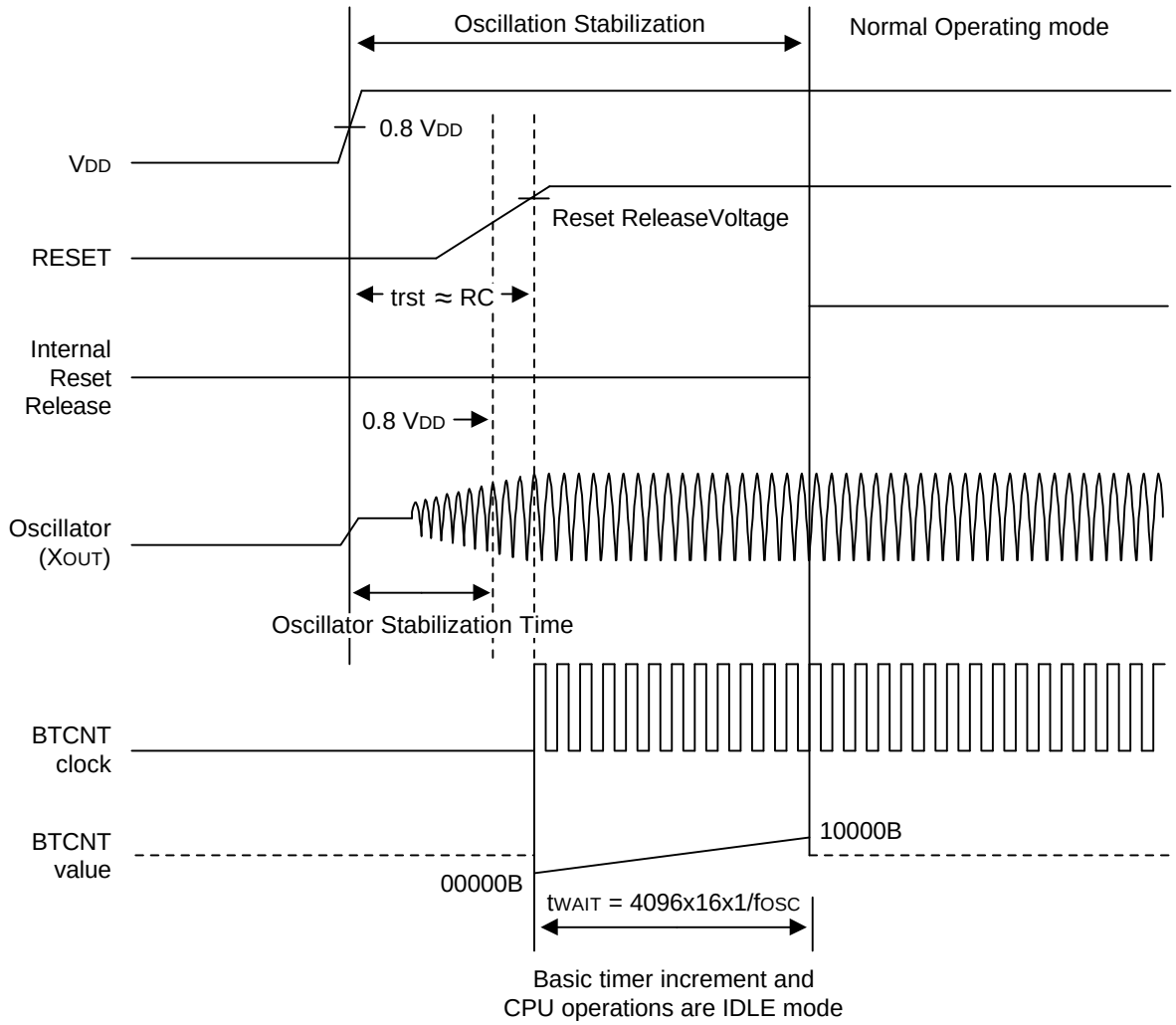
You can also use the basic timer to program a specific oscillation stabilization interval following a reset or when Stop mode has been released by an external interrupt.

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:

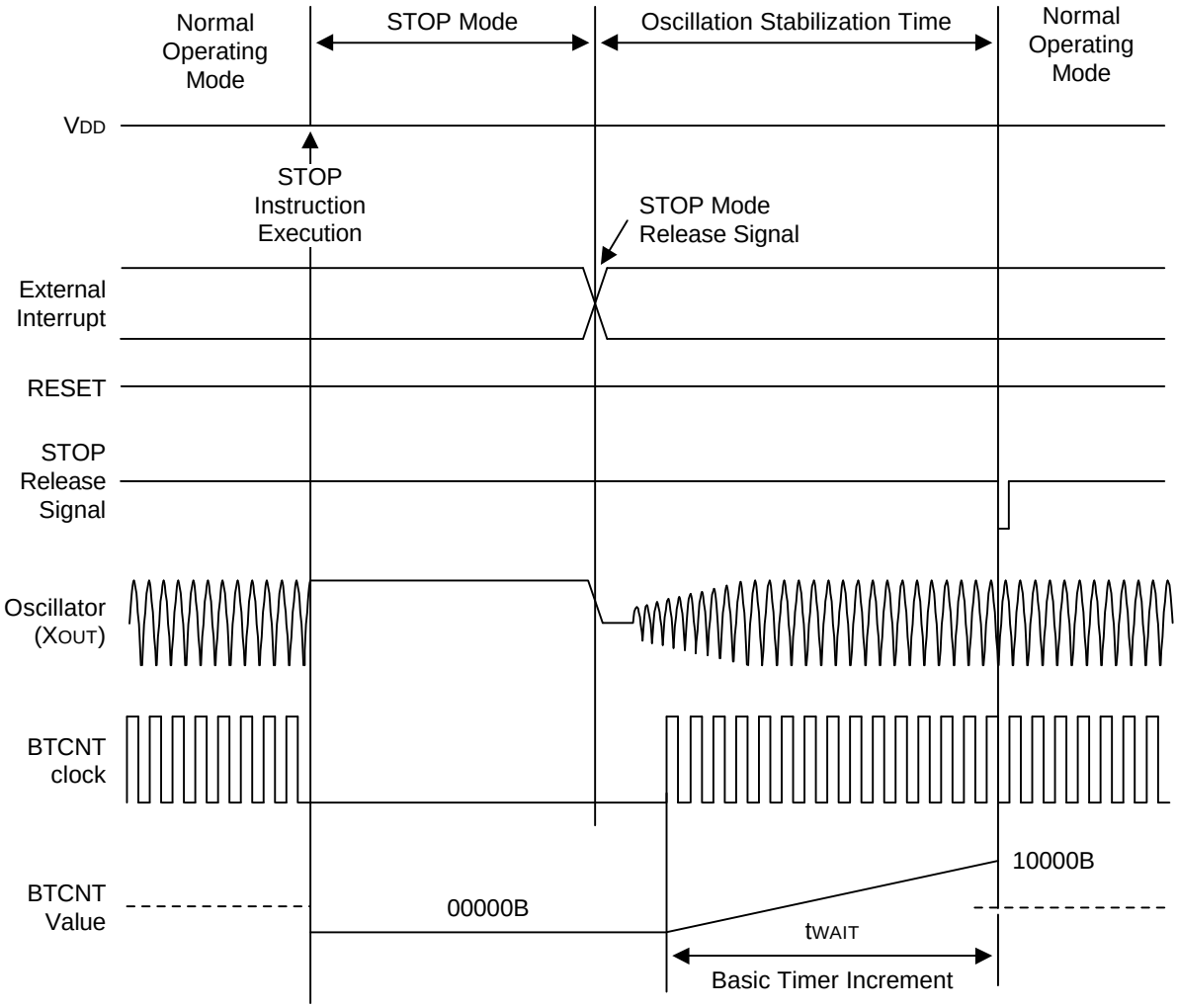
1. During Stop mode, a power-on reset or an external interrupt occurs to trigger the Stop mode release and oscillation starts.
2. If a power-on reset occurred, the basic timer counter will increase at the rate of $f_{OSC}/4096$. If an external interrupt is used to release Stop mode, the BTCNT value increases 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.

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



NOTE: During of the oscillator stabilization wait time, tWAIT, when it is released by a Power-on-reset is 4096x16/fosc.

Figure 10-2. Oscillation Stabilization Time on RESET



NOTE: Duration of the oscillator stabilzation 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 10 MHz)
0	0	4096 x 16/fosc	6.55 ms
0	1	1024 x 16/fosc	1.64 ms
1	0	128 x 16/fosc	0.2 ms
1	0	Invalid setting	—

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

☞ PROGRAMMING TIP — Configuring the Basic Timer

This example shows how to configure the basic timer to sample specification

```
;----------<<Initialize system and peripherals>>

            ORG     0100H                   ;  Reset start address
RESET       DI                              ;  disable interrupt
            LD      BTCON,#10100010B        ;  disable watchdog function
                                            ;  clock source: fosc/4096 (104 ms overflow at 10 MHz)
            LD      CLKCON,#00011000B       ;  CPU clock source select (non-divided)
            LD      SP,#0C0H                ;  S3C9424/C9428/P9428 Stack pointer initial
                    •
                    •
                    •
                    •
             EI                             ;  enable interrupt

;----------<< Main loop >>

MAIN
                    •
                    •
                    •
            LD      BTCON,#02H              ;  enable watchdog function
                                            ;  clear basic timer counter (BTCON) before overflow
                                               occurs
                    •
                    •
                    •
            JP      T,MAIN
                    •
                    •
                    •
```

TIMER 0

TIMER 0 CONTROL REGISTERS (T0CONH and T0CONL)

The timer 0 control register low byte, T0CONL, is used to select the timer 0 operating mode (interval timer, capture mode, or PWM mode) and input clock frequency, to clear the timer 0 counter, and to enable the T0 overflow interrupt and T0 match/capture interrupt. It also contains a pending bit for T0 match/capture interrupts.

Timer 0 control register high byte, T0CONH, contains a pending bit for T0 overflow interrupt. Only one bit in T0CONH register is used, T0CONH.0.

A reset clears T0CONL to "00H". This sets timer 0 to normal interval timer mode, selects an input clock frequency of f_{OSC} /4096, and disables the T0 overflow interrupt and match/capture interrupts. The T0 counter can be cleared at any time during normal operation by writing a "1" to T0CONL.3.

The T0 overflow interrupt, T0OVF, is IRQ0 with vector 00H. When a T0 overflow interrupt occurs and is serviced by the CPU, the pending condition is cleared manually set by writing “0” to T0CONH.0. To enable the T0 match/capture interrupt (T0INT, IRQ0, vector 00H), you must set T0CONL.1 to "1". The interrupt service routine must clear the pending condition by writing a "0" to the T0 interrupt pending bit, T0CONL.0.

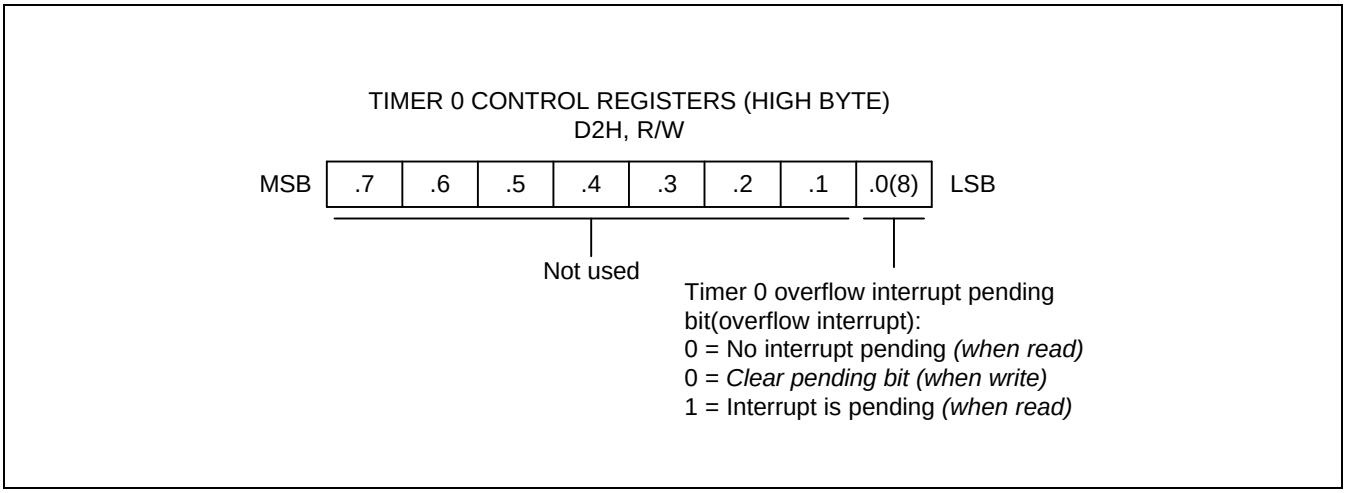


Figure 10-4. Timer 0 High-Byte Control Registers (T0CONH)

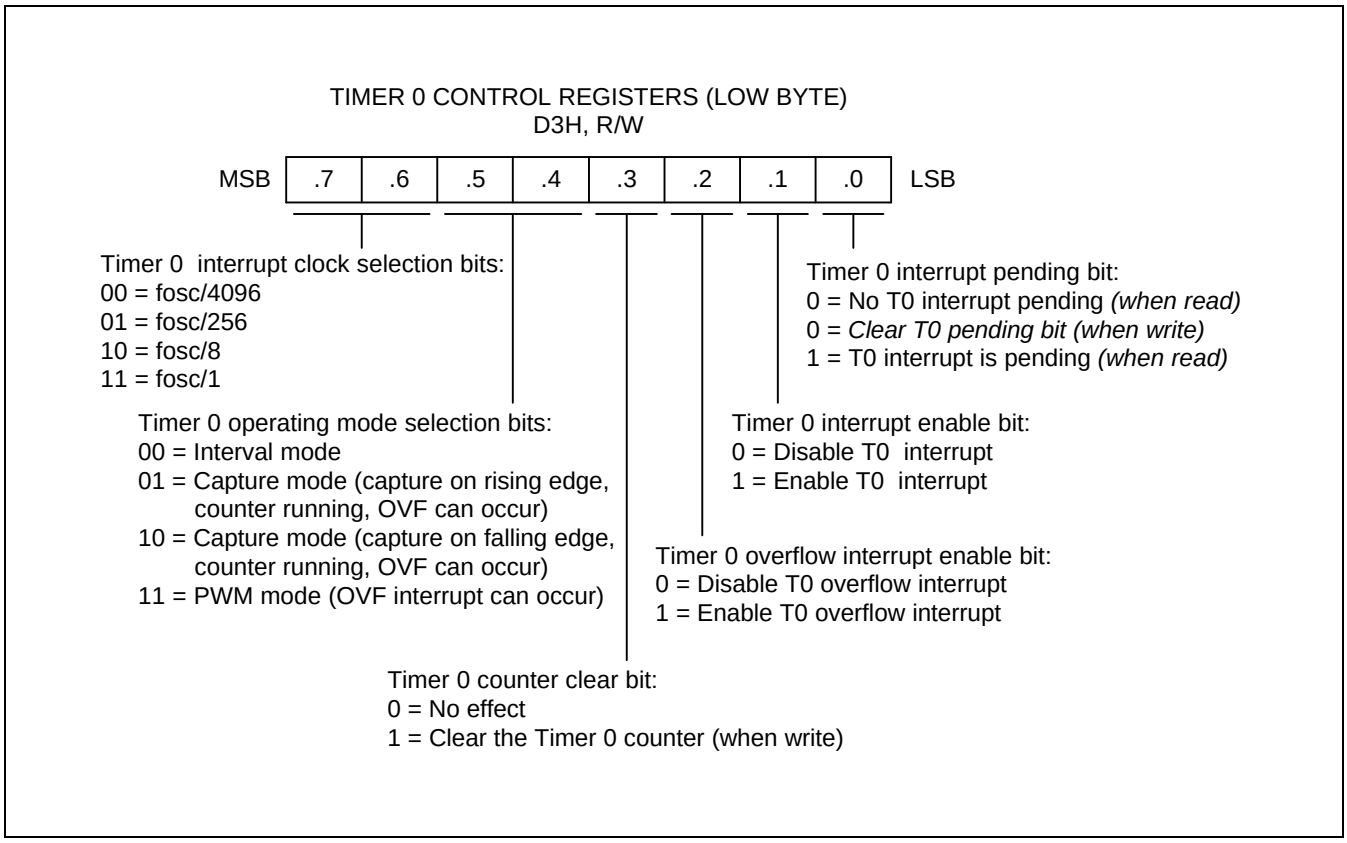


Figure 10-5. Timer 0 Low-Byte Control Registers (T0CONL)

TIMER 0 FUNCTION DESCRIPTION

Timer 0 Interrupts (IRQ0, Vectors 00H)

The Timer 0 module can generate two interrupts; the timer 0 overflow interrupt (T0OVF), and the timer 0 match/capture interrupt (T0INT). T0OVF is interrupt level IRQ0, vector 00H; T0INT is also level IRQ0, vector 00H. The T0OVF interrupt pending condition is cleared by setting the T0CONH.0 pending bit to “0”. The T0INT pending condition must be cleared by software by writing a "0" to the T0CONL.0 pending bit.

INTERVAL TIMER MODE

In interval timer mode, a match signal is generated when the counter value is identical to the value written to the Timer 0 reference data register, T0DATA. The match signal generates a Timer 0 match interrupt (T0INT, vector 00H) 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 Timer 0 interrupt request is generated, the counter value is reset and counting resumes. With each match, the level of the signal at the Timer 0 output pin is inverted.

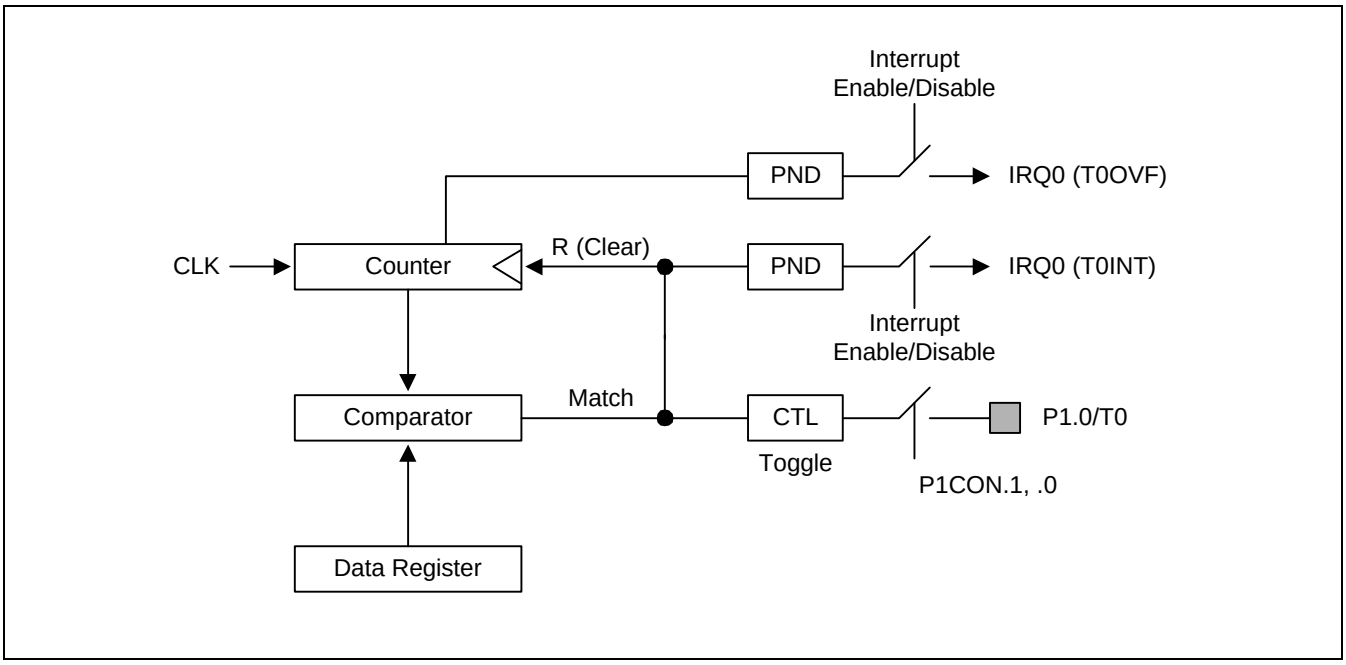


Figure 10-6. Simplified Timer 0 Function Diagram (Interval Timer Mode)

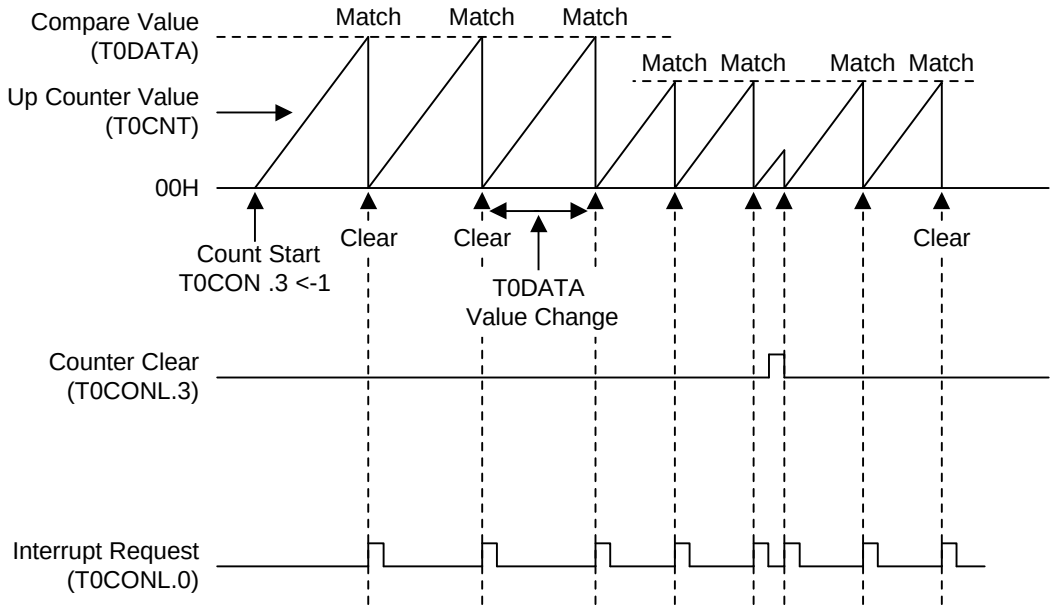
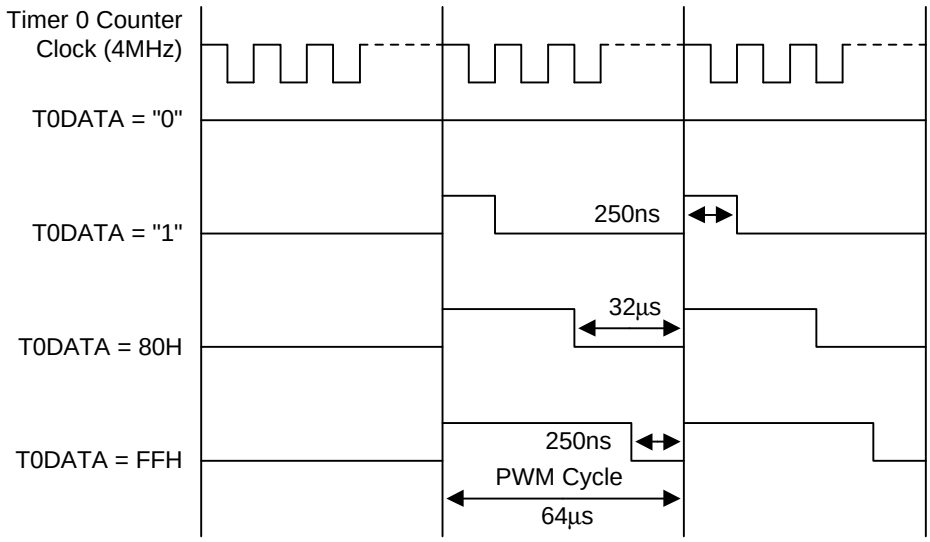


Figure 10-7. Timer 0 Timing Diagram

PULSE WIDTH MODULATION MODE

Pulse width modulation (PWM) mode lets you program the width (duration) of the pulse that is output at the T0 pin. As in interval timer mode, a match signal is generated when the counter value is identical to the value written to the T0 data register (T0DATA). In PWM mode, however, the match signal does not clear the counter (it runs continuously, overflowing at "FFH", and continues incrementing from "00H").

Although it is possible to use the match signal to generate a T0INT interrupt, an interrupt is typically not used in PWM-type applications. Instead, the pulse at the T0 pin is held to High level as long as the data register (T0DATA) value is *greater than* the counter (T0CNT) value for 8-bit PWM operation.



NOTES:
1. A system clock frequency of 4MHz is assumed.
2. The input clock of timer 0 count(T0CNT) is assumed in non-divided (fosc/1).

Figure 10-8. Simplified Timer 0 Function Diagram (PWM Mode)

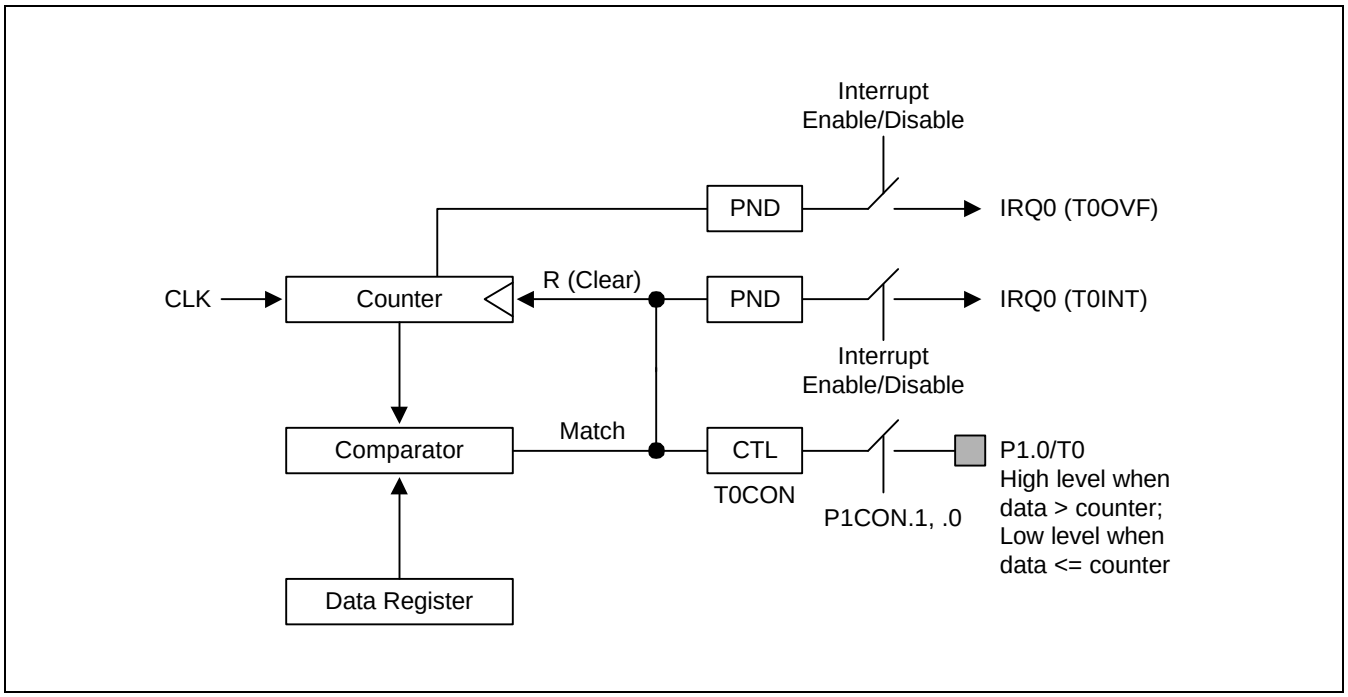


Figure 10-9. PWM Block Function Diagram

CAPTURE MODE

In capture mode, a signal edge that is detected at the T0 pin opens a gate and loads the current counter value into the T0 data register. Rising edges or falling edges can be selected to trigger this operation. Both kinds of T0 interrupts can be used in capture mode: T0OVF is generated when a counter overflow occurs, and T0INT is generated when the counter value is loaded into the data register. By reading the captured data value in T0DATA, and assuming a specific value for t_{CLK}, you can determine the pulse width (duration) of the signal being input at the T0 pin. (See Figure 10-10.)

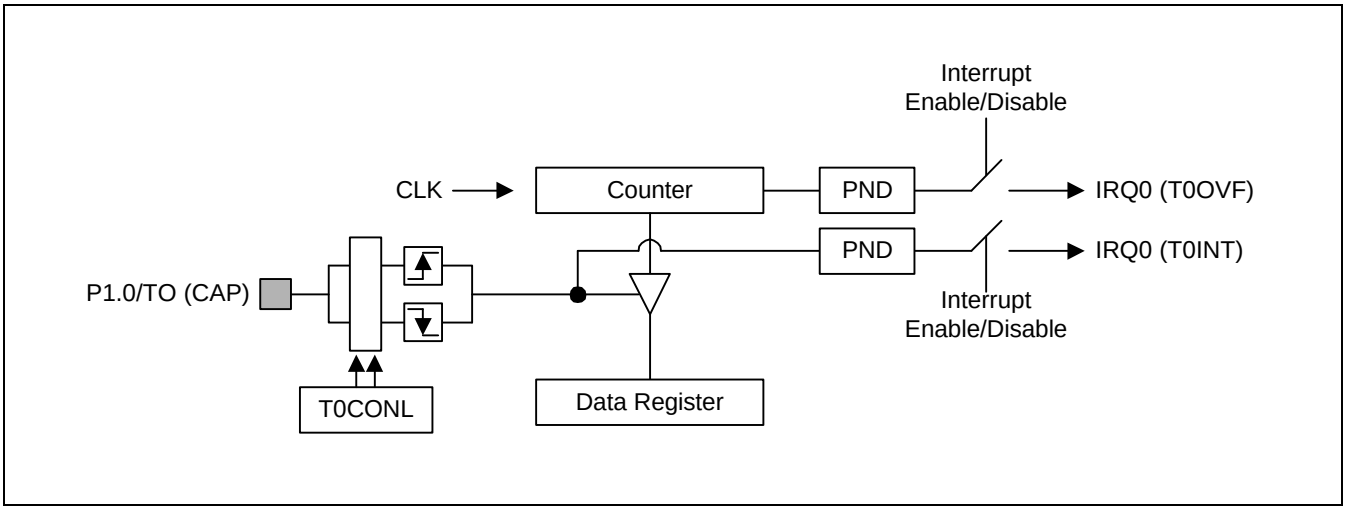
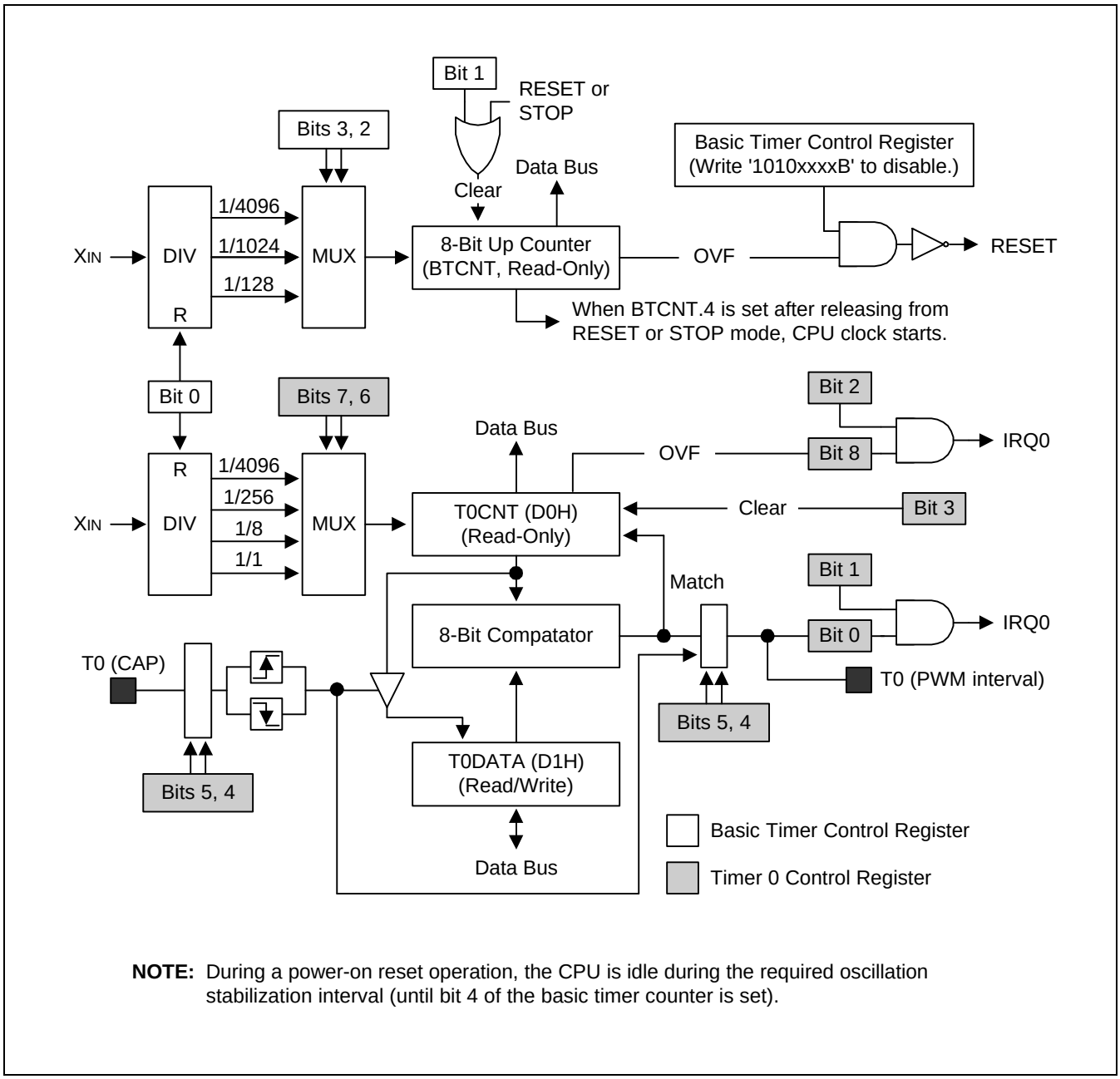


Figure 10-10. Simplified Timer 0 Function Diagram (Capture Mode)



NOTE: During a power-on reset operation, the CPU is idle during the required oscillation stabilization interval (until bit 4 of the basic timer counter is set).

Figure 10-11. Basic Timer and Timer 0 Block Diagram

☞ PROGRAMMING TIP – Configuring Timer 0 (Interval Mode)

The following sample program sets Timer 0 to interval timer mode.

```
            ORG         0000H

            VECTOR      00H, INT_4208           ;  S3C9428 has only one interrupt vector

            ORG         0100H
INITIAL:

            LD          SYM, #00H               ;  Global/Fast interrupt disable -> SYM
            LD          BTCON, #10100010B       ;  Watch-dog disable
            LD          CLKCON, #00011000B      ;  non-divided CPU clock
            LD          SP, #0C0H               ;  9428 → 00~BF (After decrease, push data)

            •
            •
            LD          T0DATA, #41H            ;  interrupt interval – 1.69msec (10MHz base)
            LD          T0CONL, #01000010B      ;  Timer 0 match interrupt enable
            •
            •
            •
            EI
MAIN:

            •
            •
            •
            CALL        SUB_ROUTINE
            •
            •
            •
            JP          MAIN
SUB_ROUTINE:

            NOP
            •
            •
            •
            RET
```

☞ PROGRAMMING TIP – Configuring Timer 0 (Interval Mode) (Continued)

```
INT_4208:                                ;  S3C9428 has just one interrupt vector

          LD        R0, T0CONL           ;  only Timer 0 match interrupt enable
          AND       R0, #00000011B
          CP        R0, #00000011B
          JP        EQ, INT_Timer 0      ;  T0CON'spending bit & INT. enable bit check

INT_Timer 0:

          AND       T0CONL, #11110110B   ;  pending clear
          •
          •
          •
          IRET
```

☞ PROGRAMMING TIP – Configuring Timer 0 (PWM Mode)

The following sample program sets Timer 0 to 8-bit PWM mode.

```
        ORG     0100H

INITIAL:

        LD      SYM, #00H               ;  Global/Fast interrupt disable -> SYM
        LD      BTCON, #10100010B       ;  Watch-dog disable
        LD      CLKCON, #00011000B      ;  non-divided CPU clock
        LD      SP, #0C0H               ;  9428 → 00~BF (After decrease, push data)
        •
        •
        LD      P1CON, #00000011B       ;  P1.0 → T0 (PWM) output
        LD      T0DATA, #80H            ;  half duty
        LD      T0CONL, #01111000B      ;  Timer 0 PWM mode
        •
        •

MAIN:

        •
        •
        •
        •
        •
        JP      MAIN
```

TIMER 1

TIMER 1 CONTROL REGISTER (T1CON)

The timer 1 control register,T1CON, located at E4H operates in interval timer mode. By setting the appropriate bits in T1CON you can select the input clock frequency and enable the Timer 1 interrupt. T1CON also contains a pending bit for Timer 1 interrupt.

A reset clears T1CON to "00H". This sets timer 1 to normal interval mode and selects an input clock frequency of fosc/4096 and disables the Timer 1 interrupt.

You may clear the timer 1 counter by either setting T1CON.2 to “1” or enable ZCD clear signal to clear the timer 1 counter by setting T1CON.3 to “1”.

To enable Timer 1 match interrupt (IRQ0, vector 00H) you must set T1CON.1 to “1”. The interrupt service routine must clear the pending condition by writing a “0” to the Timer 1 interrupt pending bit, T1CON.0.

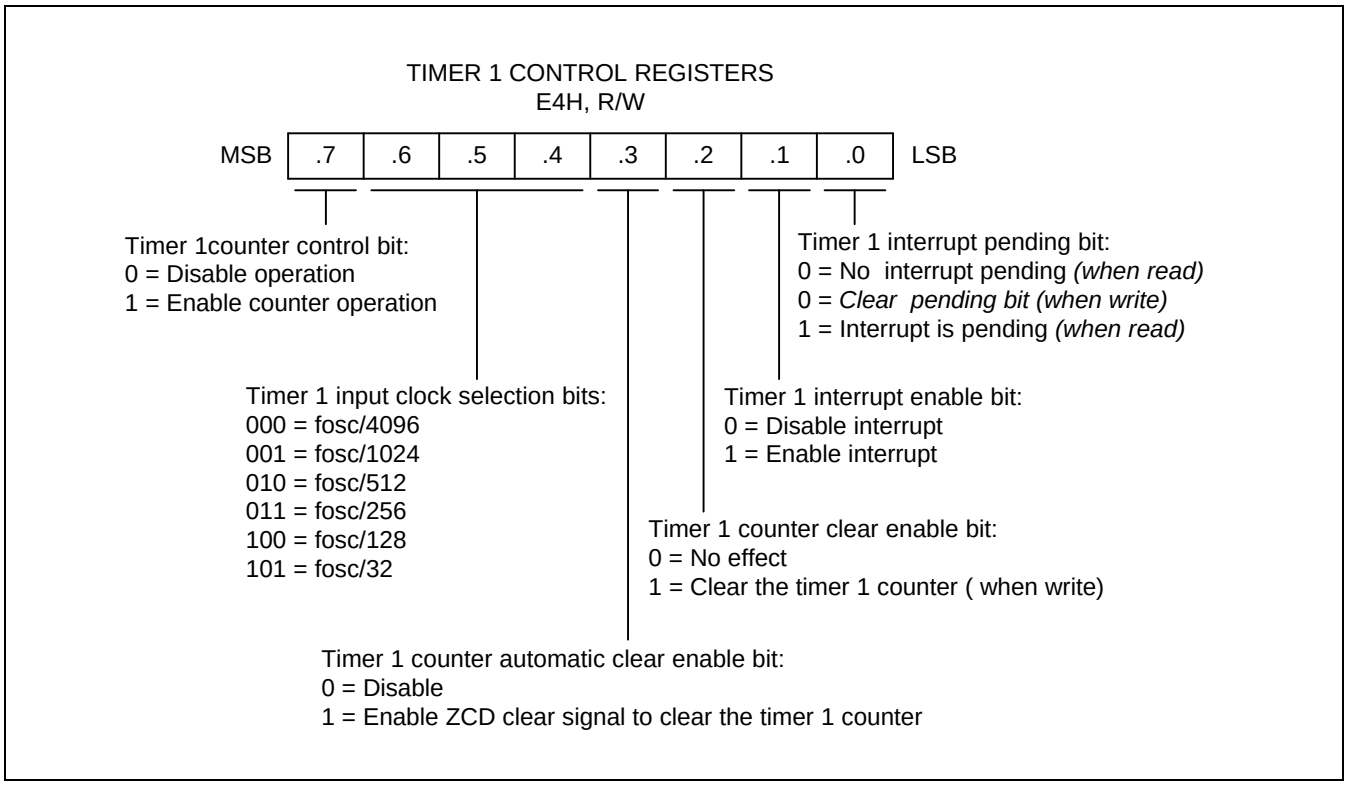
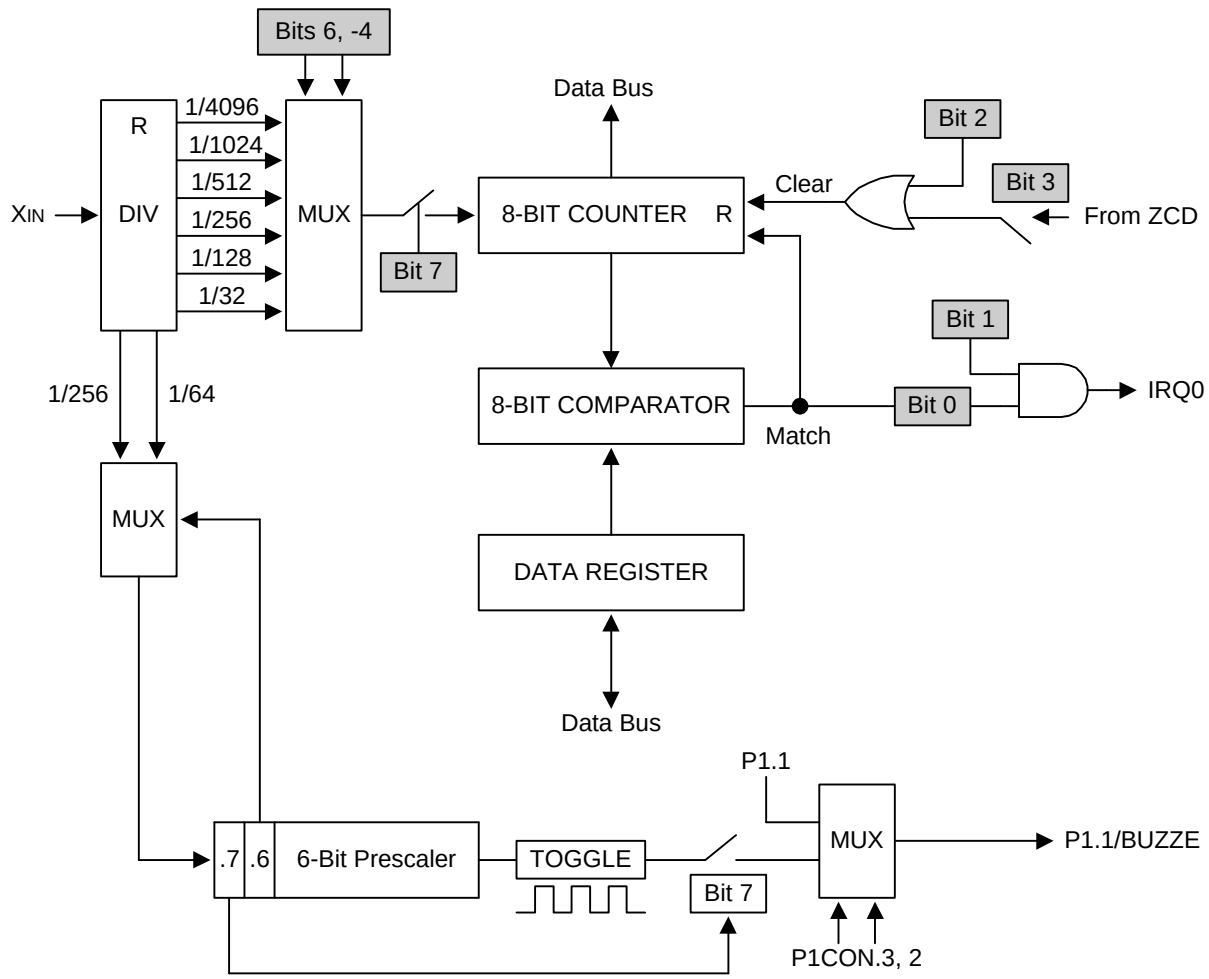


Figure 10-12. Timer 1 Control Register (T1CON)



NOTE: When P1.1/BUZZER is used as BUZZER output pin, the initial value is "0" (Low Level). (When the bit7 of BUZPS is set to "0", the output of P1.1 is Low.)

Figure 10-13. Timer 1 Block Diagram

BUZZER OUTPUT CONTROL REGISTER (BUZPS)

Buzzer output control register is used to select the frequency from 200 Hz to 20 KHz. And these various frequency can be used to generate the melody signal. By selecting the clock source (bit of BUZPS) and the value of prescaler, the desire frequency can be obtained. The BUZPS.7 can be used to control the buzzer output when P1.1 is set to buzzer output mode (configure P1CON).

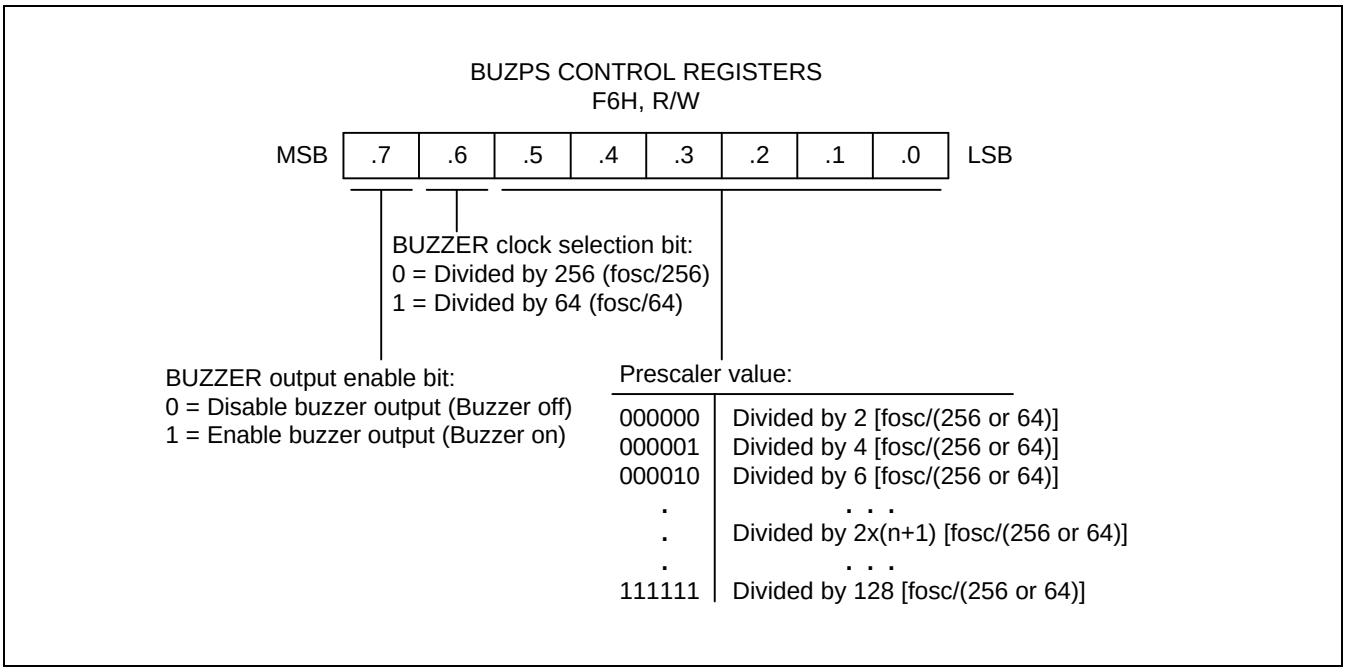


Figure 10-14. Buzzer Output Control Register (BUZPS)

☞ PROGRAMMING TIP – Configuring Timer 1

The following sample program sets Timer 1 to interval timer mode.

```
        ORG     0000H

        VECTOR  00H, INT_4208       ;  S3C9428 has only one interrupt vector

        ORG     0100H

INITIAL:

        LD      SYM, #00H           ;  Global/Fast interrupt disable -> SYM
        LD      BTCON, #10100010B   ;  Watch-dog disable
        LD      CLKCON, #00011000B  ;  non-divided CPU clock
        LD      SP, #0C0H           ;  9428 → 00~BF (After decrease, push data)
        •
        •
        LD      T1DATA, #41H        ;  interrupt interval – 6.76msec (10MHz base)
        LD      T1CON, #10010110B   ;  Timer1 match interrupt enable
        •
        •
        •
        EI

MAIN:

        •
        •
        •
        CALL    SUB_ROUTINE
        •
        •
        •
        JP      MAIN

SUB_ROUTINE:

        NOP
        •
        •
        •
        RET
```

☞PROGRAMMING TIP – Configuring Timer 1 (Continued)

```
INT_4208:                                   ;  S3C9428 has just one interrupt vector

          LD        R0, T1CON
          AND       R0, #00000011B
          CP        R0, #00000011B
          JP        EQ, INT_T1              ;  Timer 0 interrupt routine
          •
          •
          •
INT_T1:

          AND       T1CON, #11111010B       ;  pending bit clear
          •
          •
          •
          IRET
```

☞PROGRAMMING TIP – Configuring Buzzer

The following sample program sets Buzzer output.

```
        ORG     0000H

        VECTOR  00H, INT_4208       ;  S3C9428 has only one interrupt vector

        ORG     0100H

INITIAL:

        LD      SYM, #00H           ;  Global/Fast interrupt disable -> SYM
        LD      BTCON, #10100010B   ;  Watch-dog disable
        LD      CLKCON, #00011000B  ;  non-divided CPU clock
        LD      SP, #0C0H           ;  9428 → 00~BF (After decrease, push data)
        •
        •
        LD      P1CON, #00001100B   ;  P1.1 → Buzzer output
        LD      BUZPS, #10000000B   ;  fosc/512 Buzzer wave output
        •
        •
        •
        EI

MAIN:

        •
        •
        •
        CALL    SUB_ROUTINE
        •
        •
        •
        JP      MAIN

SUB_ROUTINE:

        NOP
        •
        •
        •
        RET
```

11 A/D CONVERTER

OVERVIEW

The A/D converter (ADC) module uses successive approximation logic to convert analog levels at one of the twelve input channels to equivalent 10-bit digital values. The analog input level must lie between the AV_{REF} and AV_{SS} values. The A/D converter has the following components:

— Twelve multiplexed analog input pins (AD0-AD11)

— Analog comparator with successive approximation logic

— 10-bit A/D conversion data output registers (ADDATAH, ADDATAL)

— ADC control register (ADCON)

An analog-to-digital conversion procedure is initiated when the CPU writes a value to the ADCON register at address F7H to select one of the twelve available input pins. You select the desired input channel by setting the appropriate bits in the ADCON register.

The S3C9424/C9428/P9428 microcontroller performs 10-bit conversions for only one input channel at a time. You can dynamically select different analog input channels during program execution by manipulating selection bits in the ADCON register.

During a normal conversion, ADC logic initially sets the successive approximation register to 200H (the approximate half-way point of an 10-bit register). This register is then updated automatically during each conversion step. The successive approximation block performs 10-bit conversions for one input channel at a time. You can dynamically select different channels by mainpulating the channel selection bit value (ADCON.7-4) in the ADCON register. To start the A/D conversion, you should set a the enable bit, ADCON.0. When a conversion is completed, ADCON.3, the end-of-conversion (EOC) bit is automatically set to 1 and the result is dumped into the ADDATA register where it can be read. The A/D converter then enters an idle state. Remember to read the contents of ADDATA before another conversion starts. Otherwise, the previous result will be overwritten by the next conversion result.

NOTE

Because the ADC does not use sample-and-hold circuitry, it is important that any fluctuations in the analog level at the AD0-AD11 input pins during a conversion procedure be kept to an absolute minimum. Any change in the input level, perhaps due to circuit noise, will invalidate the result.

INTERNAL REFERENCE VOLTAGE LEVELS

In the ADC function block, the analog input voltage level is compared to the reference voltage. The analog input level must remain within the range AV_{SS} to AV_{REF} (usually, AV_{REF} = V_{DD}).

Different reference voltage levels are generated internally along the resistor tree during the analog conversion process for each conversion step. The reference voltage level for the first bit conversion is always 1/2 AV_{REF}.

USING A/D PINS FOR STANDARD DIGITAL INPUT

The ADC module's input pins are alternatively used as digital input in port 0 and port 2. The AD0-AD7 share pin names are P2.0-P2.7 and AD8-AD11 share pin names are P0.4-P0.7, respectively

A/D CONVERTER CONTROL REGISTER (ADCON)

The A/D converter control register, ADCON, is located at address F7H. ADCON has four functions:

— Bits 7–4 select an analog input pin (AD0-AD11).

— Bit 3 indicates the status of the A/D conversion.

— Bit2-1 select clock source.

— Bit 0 starts the A/D conversion.

Only one analog input channel can be selected at a time. You can dynamically select any one of the eight analog input pins (AD0-AD11) by manipulating the 4-bit value for ADCON.7-ADCON.4

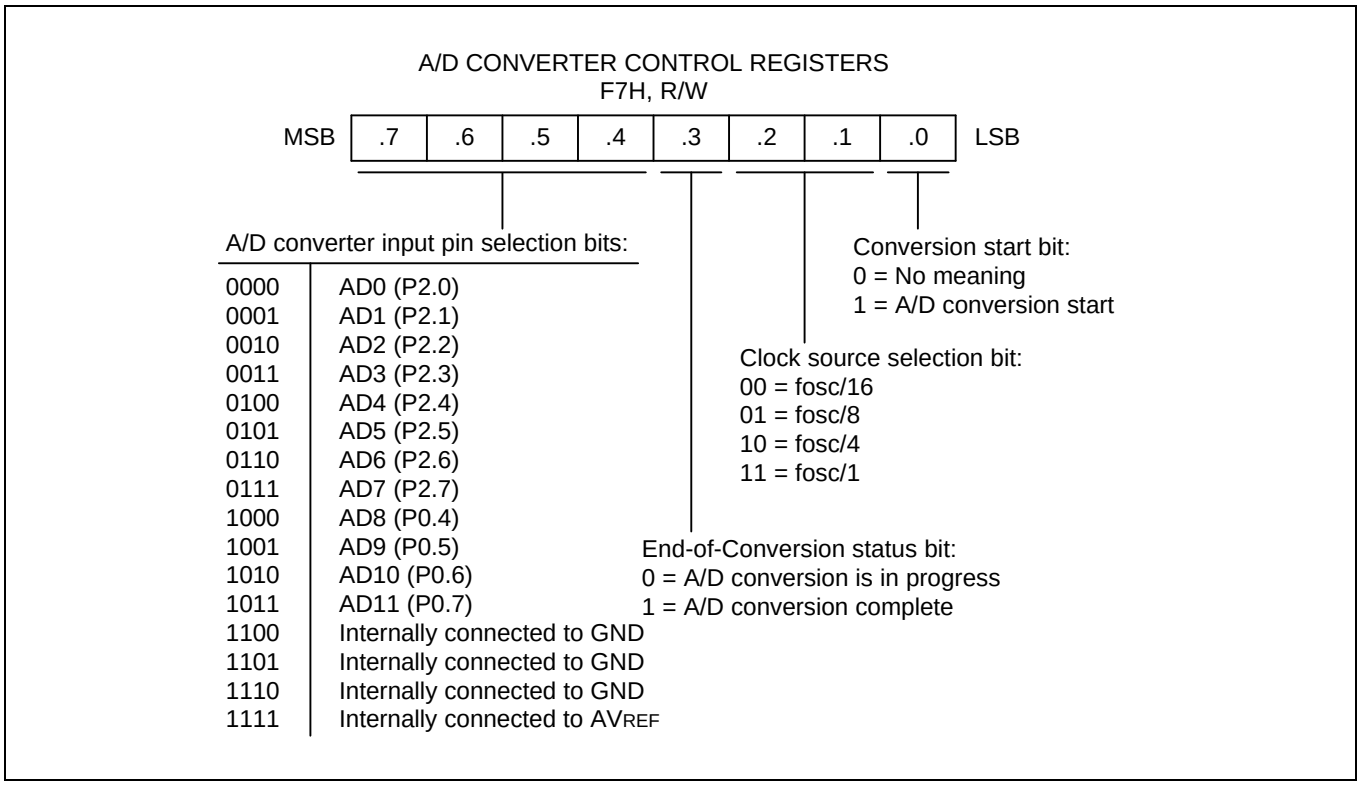


Figure 11-1. A/D Converter Control Register (ADCON)

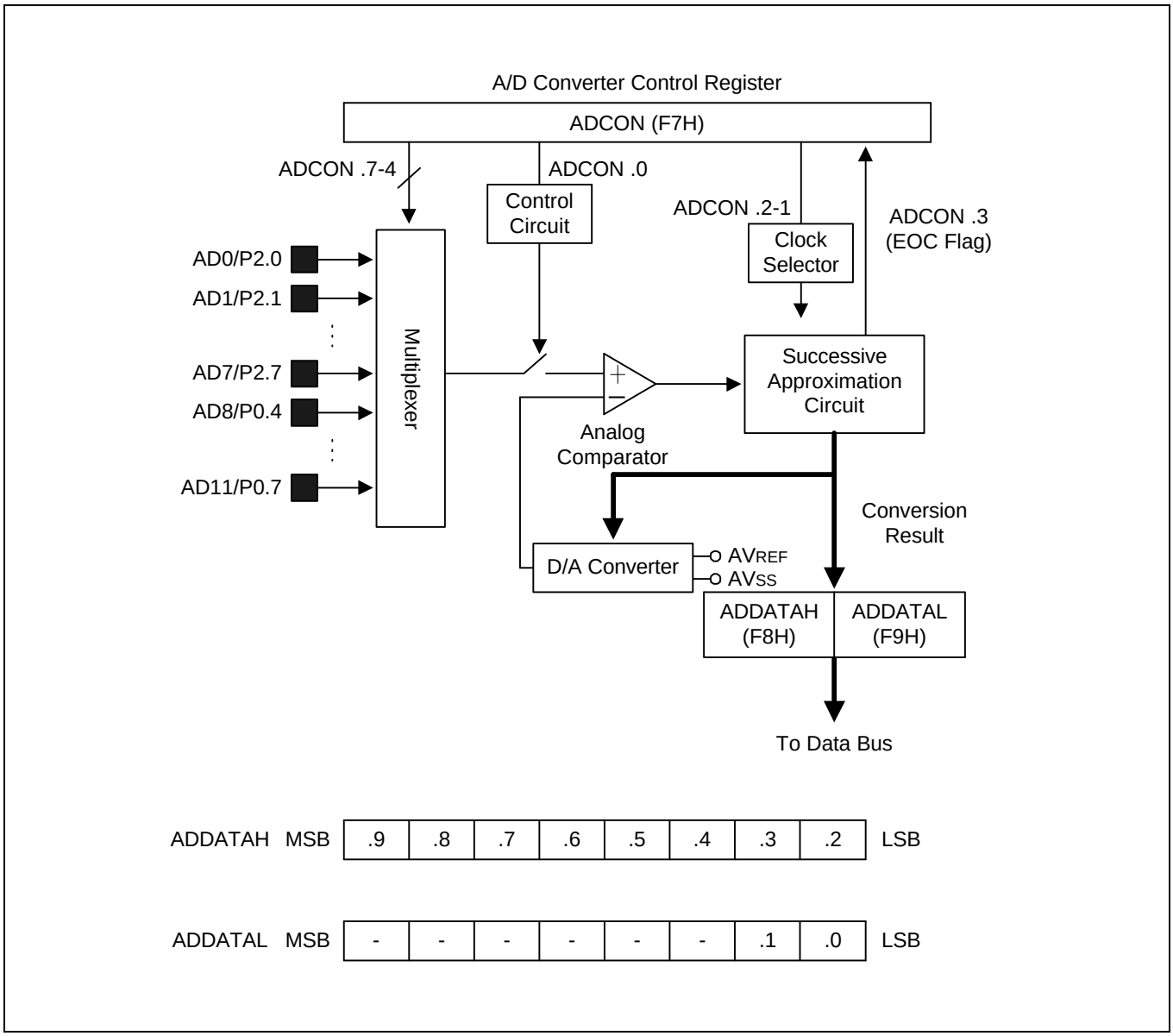


Figure 11-2. A/D Converter Circuit Diagram

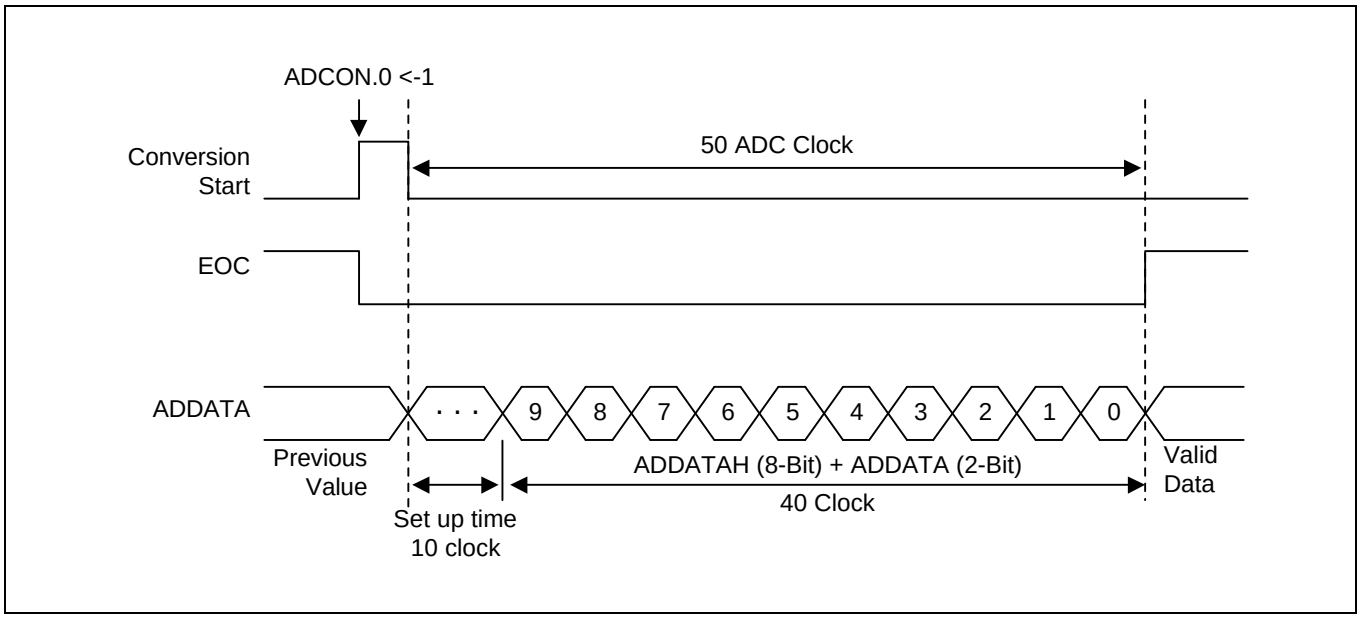


Figure 11-3. A/D Converter Timing Diagram

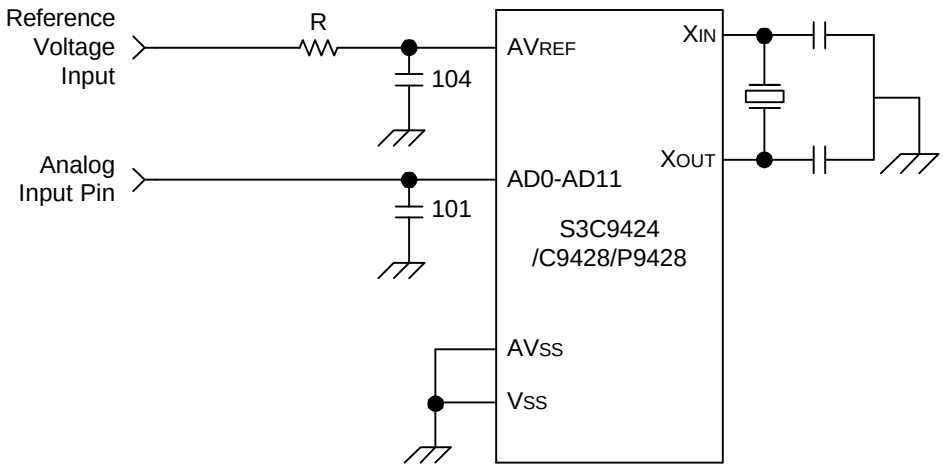
CONVERSION TIMING

The A/D conversion process requires 4 steps (4 clock edges) to convert each bit and 10 clocks to step-up A/D conversion. Therefore, total of 50 clocks are required to complete an 10-bit conversion: With an 10MHz CPU clock frequency, one clock cycle is 400 ns (4/fosc). If each bit conversion requires 4 clocks, the conversion rate is calculated as follows:

4 clocks/bit x 10-bits + set-up time (10 clock) = 50 clocks
50 clock x 400 ns = 20 μs at 10MHz, 1 clock time = 4/fosc (assuming ADCON.2-.1 = 10)

INTERNAL A/D CONVERSION PROCEDURE

1. Analog input must remain between the voltage range of AV_{SS} and AV_{REF}.
2. Configure the analog input pins to input mode by making the appropriate settings in P2CONH, P2CONL and P0CONH registers.
3. Before the conversion operation starts, you must first select one of the twelve input pins (AD0–AD11) by writing the appropriate value to the ADCON register.
4. When conversion has been completed, (40 CPU clocks have elapsed), the EOC flag is set to “1”, so that a check can be made to verify that the conversion was successful.
5. The converted digital value is loaded to the output register, ADDATAH (High 8-bit) and ADDATAL (Low 2-bit), then the ADC module enters an idle state.
6. The digital conversion result can now be read from the ADDATAH and ADDATA registers.



NOTES:
1. The symbol 'R' signifies an offset resistor with a value of from 50 to 100 Ohms.
2. It is recommended that GND of the oscillator and GND of the AVSS/VSS must be connected separately with GND of the power.

Figure 11-4. Recommended A/D Converter Circuit for Highest Absolute Accuracy

☞ PROGRAMMING TIP PROGRAMMING TIP – Configuring 10-bit A/D Converter

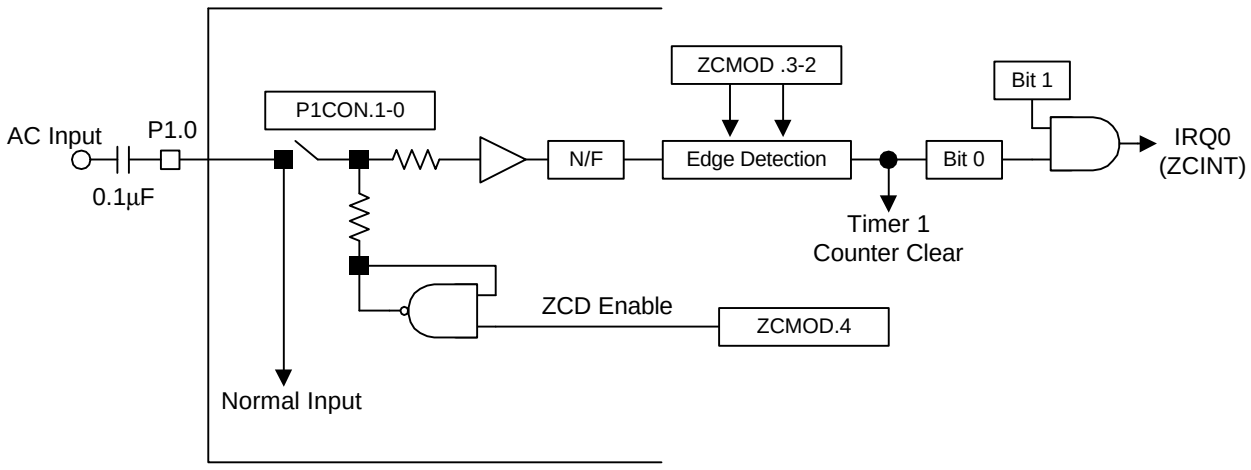
```
                ORG     0000H

                VECTOR  00H, INT_4208       ;  S3C9428 has only one interrupt vector

                ORG     0100H

INITIAL:

                LD      SYM, #00H           ;  Global/Fast interrupt disable -> SYM
                LD      BTCON, #10100010B   ;  Watch-dog disable
                LD      CLKCON, #00011000B  ;  non-divided CPU clock
                LD      SP, #0C0H           ;  9428 → 00–BF (After decrease, push data)
                •
                •
                •
                EI

MAIN:

                •
                •
                •
                CALL    Sub_ADC
                •
                •
                •
                JP      MAIN

Sub_ADC:

                LD      P0CONH, #55H        ;  P0.7~0.4: AD input enable
                LD      P0PUR, #00H         ;  pull-up disable
                LD      ADCON, #10000001B   ;  Select P0.4, Conversion start

CONV_LOOP:

                TM      ADCON, #00001000B   ;  Check E0C bit
                JP      Z, CONV_LOOP
                                            ;  Conversion is completed
                LD      R8, ADDATAH         ;  High 8-bit of result
                LD      R9, ADDATAL         ;  Low 2-bit of result
                •
                •
                •
                RET
```

12 ZERO-CROSSING DETECTION CIRCUIT

OVERVIEW

Zero-crossing detection circuit in Samsung's S3C9424/C9428/P9428, generates a digital signal in synchronism with an AC signal input. It provides the timing signal for operations which are synchronized with the AC line. The zero crossing detection circuit digitizes the AC signal it receives from the power supply.

By setting bits 1 and 0 in port 1 control register (P1CON), you can enable zero-crossing detection. Zero-crossing detector is shown in Figure 12-1.



NOTE: N/F is the abbreviation of noise filter.

Figure 12-1. Zero-Crossing Detector Diagram

ZERO-CROSSING DETECTOR CONTROL REGISTER

The zero crossing detector control register, ZCMOD, is used to select interrupt mode (interrupt on falling edge, rising edge or both).

Reset clears ZCMOD to '00H', and configures interrupt selection mode to falling edge and disables ZCD interrupt. The interrupt pending bit must be cleared by writing "0" to ZCMOD.0

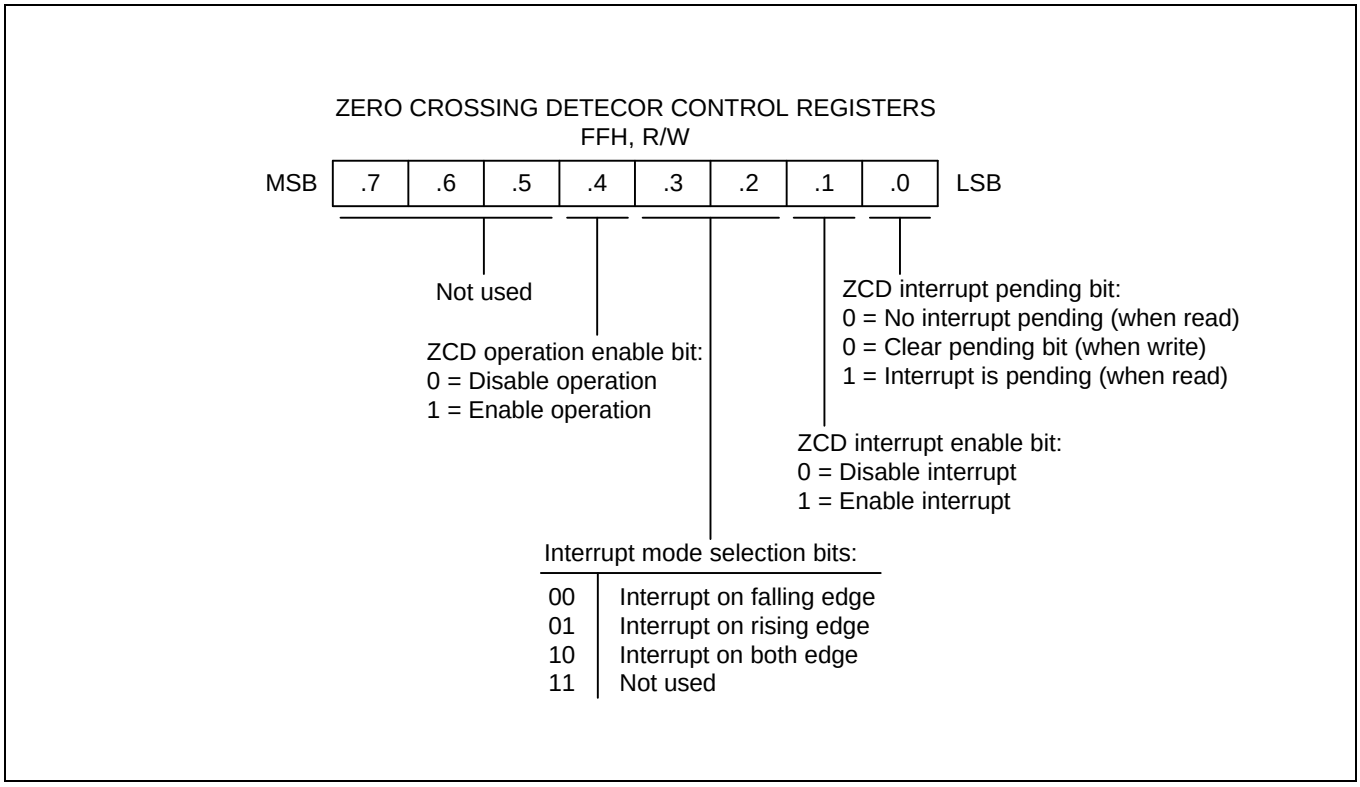


Figure 12-2. Zero-Crossing Detector Control Register (ZCMOD)

ZERO CROSS DETECTOR

ZCD circuit detects the zero-cross point of the AC waveform. Three types of detection can be selected, the point from positive to negative, the point from negative to positive, and both.

The zero cross detection circuit has the noise filter circuit in it. The detected zero cross point can be used to clear the timer 1 counter (T1CON.3 = 1).

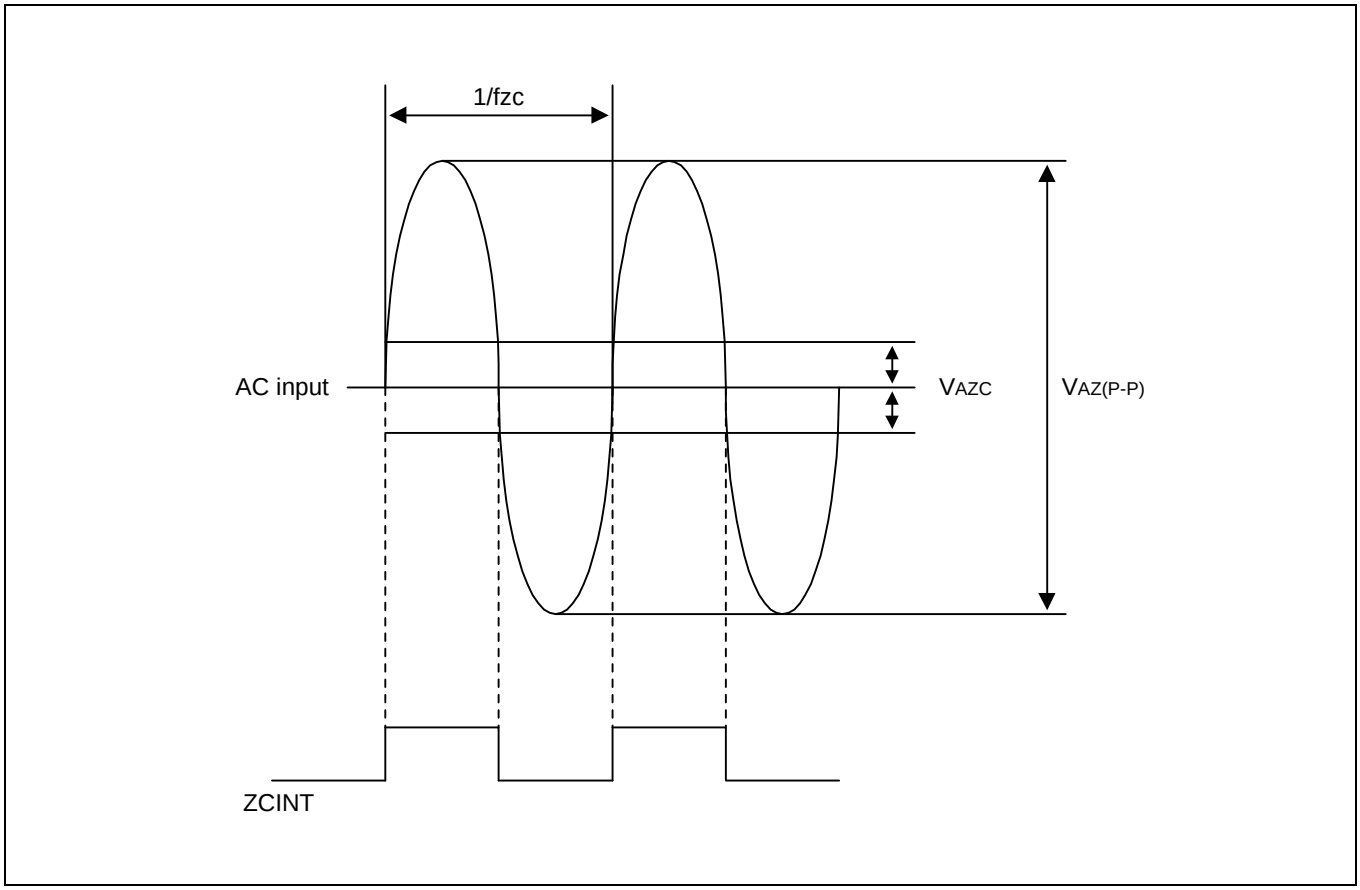


Figure 12-3. Zero-Crossing Waveform Diagram

☞ PROGRAMMING TIP – Configuring ZCD

```
        ORG     0000H

        VECTOR  00H, INT_4208       ;  S3C9428 has only one interrupt vector

        ORG     0100H

INITIAL:

        LD      SYM, #00H           ;  Global/Fast interrupt disable -> SYM
        LD      BTCON, #10100010B   ;  Watch-dog disable
        LD      CLKCON, #00011000B  ;  non-divided CPU clock
        LD      SP, #0C0H           ;  9428 → 00–BF (After decrease, push data)
        •
        •
        LD      P1CON, #00000001B   ;  P1.0 → ZCD input enable
                                    ;  Input sine wave to P1.0
        LD      ZCMOD, #00010010B   ;  ZCD operation/interrupt enable
        •
        •
        •
        EI

MAIN:

        •
        •
        •
        CALL    SUB_ROUTINE
        •
        •
        •
        JP      MAIN

SUB_ROUTINE:

        NOP
        •
        •
        •
        RET
```

☞ PROGRAMMING TIP1 – Configuring ZCD (Continued)

```
INT_4208:

        LD      R0, ZCMOD
        AND     R0, #00000011B
        CP      R0, #00000011B
        JP      EQ, INT_ZCD             ;  ZCD interrupt routine
        •
        •
        •

INT_ZCD:

        AND     ZCMOD, #11111110B       ;  pending clear
        •
        •
        •
        IRET
```

13 12-BIT PWM (PULSE WIDTH MODULATION)

OVERVIEW

This microcontroller has the 12-bit PWM circuit. The operation of all PWM circuit is controlled by a single control register, PWMCON.

The PWM counter is a 12-bit incrementing counter. It is used by the 12-bit PWM circuits. To start the counter and enable the PWM circuits, you set PWMCON.2 to "1". If the counter is stopped, it retains its current count value; when re-started, it resumes counting from the retained count value. When there is a need to clear the counter you set PWMCON.3 to "1".

You can select a clock for the PWM counter by set PWMCON.6-.7. Clocks which you can select are Fosc/256, Fosc/64, Fosc/8, Fosc/1.

FUNCTION DESCRIPTION

PWM

The 12-bit PWM circuits have the following components:

— 6-bit comparator and extension cycle circuit
— 6-bit reference data registers (PWM0, PWM1)
— 6-bit extension data registers (PWM0EX, PWM1EX)
— PWM output pins (P0.7/PWM0, P1.3/PWM1)

PWM counter

The PWM counter is a 12-bit incrementing counter comprised of a lower 6-bit counter and an upper 6-bit counter.

To determine the PWM module's base operating frequency, the lower byte counter is compared to the PWM data register value. In order to achieve higher resolutions, the six bits of the upper counter can be used to modulate the "stretch" cycle. To control the "stretching" of the PWM output duty cycle at specific intervals, the 6-bit extended counter value is compared with the 6-bit value (bits 7-2) that you write to the module's extension register.

PWM data and extension registers

PWM (duty) data registers, located in FAH and FCH, determine the output value generated by each 12-bit PWM circuit. These registers, PWM is read/write addressable.

— 8-bit data register PWM0 and PWM1, of which only bits 5-0 are used.

— 8-bit extension registers PWM0EX (FBH) and PWM1EX (FDH), of which only bits 7-2 are used

To program the required PWM output, you load the appropriate initialization values into the 6-bit data registers (PWM0, PWM1) and the 6-bit extension registers (PWM0EX, PWM1EX). To start the PWM counter, or to resume counting, you set PWMCON.2 to "1".

A reset operation disables all PWM output. The current counter value is retained when the counter stops. When the counter starts, counting resumes at the retained value.

PWM clock rate

The timing characteristics of both 12-bit output channels are identical, and are based on the Fosc clock frequency. The counter clock value is determined by the setting of PWMCON.6-.7.

Table 13-1. PWM Control and Data Registers

Register Name	Mnemonic	Address	Function
PWM data registers	PWM0, PWM1	FAH, FCH	6-bit PWM basic cycle frame value
	PWM0EX, PWM1EX	FBH, FDH	6-bit extension ("stretch") value
PWM control registers	PWMCON	FEH	PWM counter stop/start (resume), and Fosc clock settings

PWM function Description

The PWM output signal toggles to Low level whenever the lower 6-bit counter matches the reference value stored in the module's data register (PWM0, PWM1). If the value in the PWM0 or PWM1 register is not zero, an overflow of the lower counter causes the PWM output to toggle to High level. In this way, the reference value written to the data register determines the module's base duty cycle.

The value in the 6-bit extension counter is compared with the extension settings in the 6-bit extension data registers (PWM0EX, PWM1EX). This 6-bit extension counter value, together with extension logic and the PWM module's extension register , is then used to "stretch" the duty cycle of the PWM output. The "stretch" value is one extra clock period at specific intervals, or cycles (see Table 13-2).

If, for example, the value in the extension register is '04H', the 32nd cycle will be one pulse longer than the other 63 cycles. If the base duty cycle is 50 %, the duty of the 32nd cycle will therefore be "stretched" to approximately 51% duty. For example, if you write 80H to the extension register, all odd-numbered pulses will be one cycle longer. If you write FCH to the extension register, all pulses will be stretched by one cycle except the 64th pulse. PWM output goes to an output buffer and then to the corresponding PWM output pin. In this way, you can obtain high output resolution at high frequencies.

Table 13-2. PWM output "stretch" Values for Extension Registers PWM0EX

PWM0EX Bit	"Stretched" Cycle Number
7	1, 3, 5, 7, 9, . . . , 55, 57, 59, 61, 63
6	2, 6, 10, 14, . . . , 50, 54, 58, 62
5	4, 12, 20, . . . , 44, 52, 60
4	8, 24, 40, 56
3	16, 48
2	32
1	Not used
0	Not used

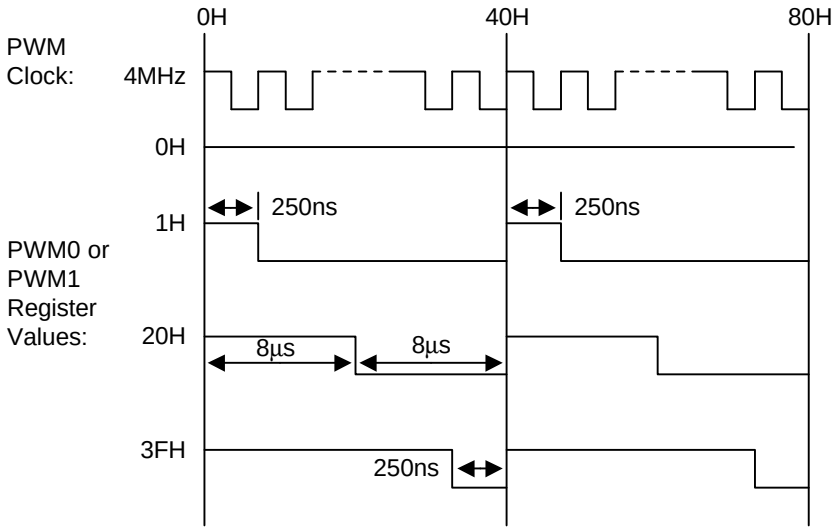


Figure 13-1. 12-Bit PWM Basic Waveform

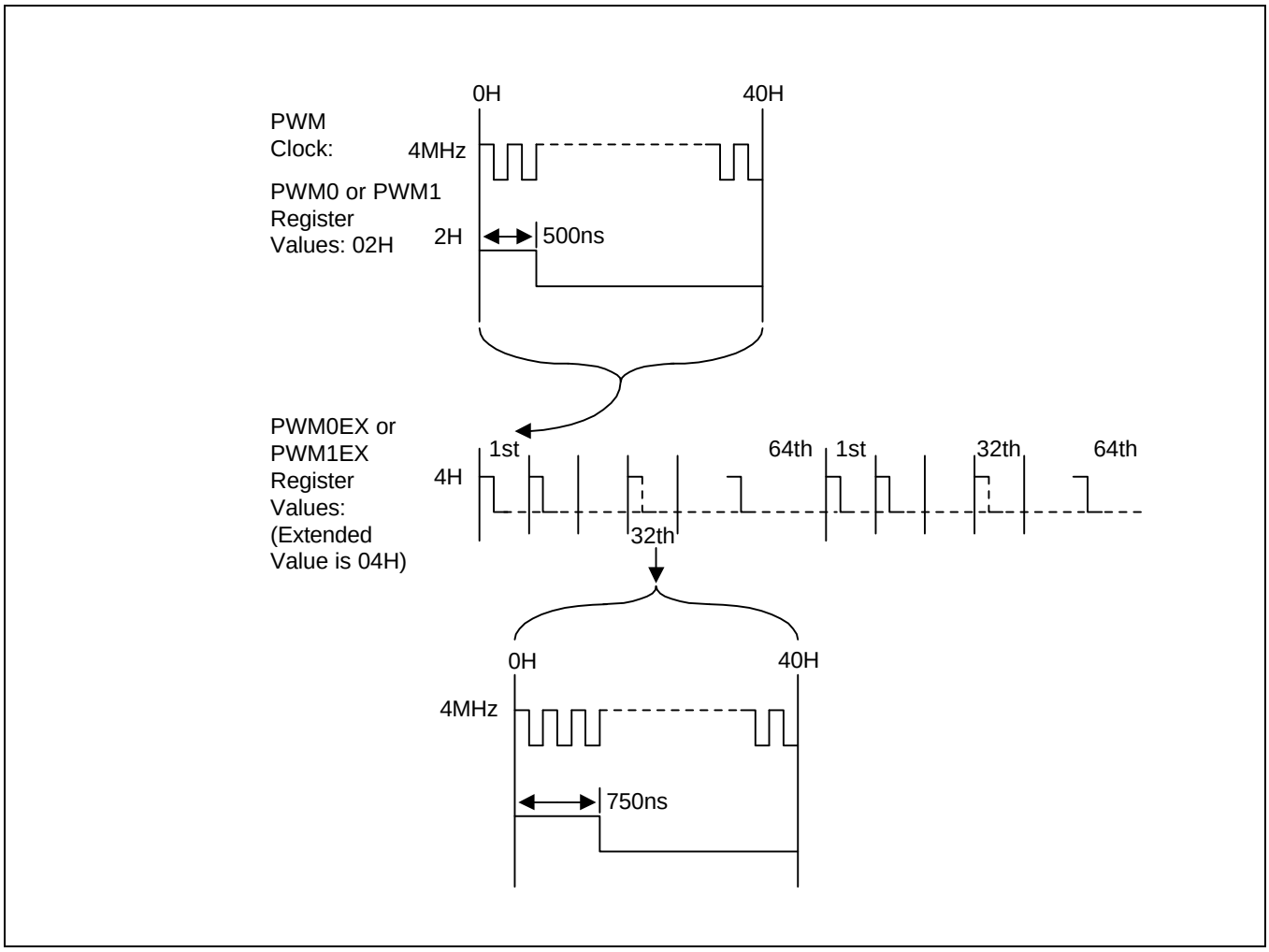


Figure 13-2. 12-Bit Extended PWM Waveform

PWM CONTROL REGISTER (PWMCON)

The control register for the PWM module, PWMCON, is located at register address FEH. PWMCON is used the 12-bit PWM modules. Bit settings in the PWMCON register control the following functions:

— PWM counter clock selection
— PWM data reload interval selection
— PWM counter clear
— PWM counter stop/start (or resume) operation
— PWM counter overflow (upper 6-bit counter overflow) interrupt control

A reset clears all PWMCON bits to logic zero, disabling the entire PWM module.

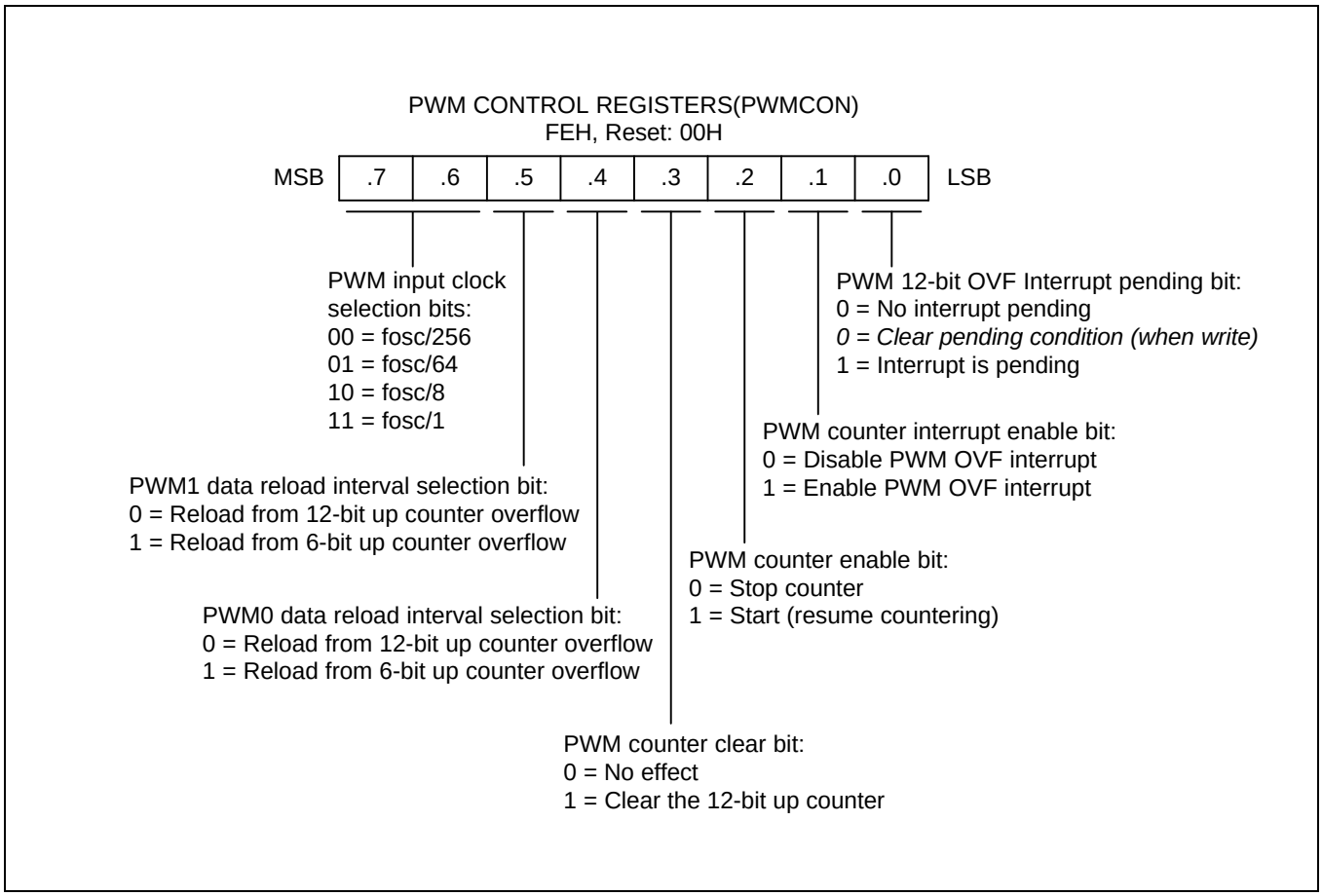


Figure 13-3. PWM/Capture Module Control Register (PWMCON)

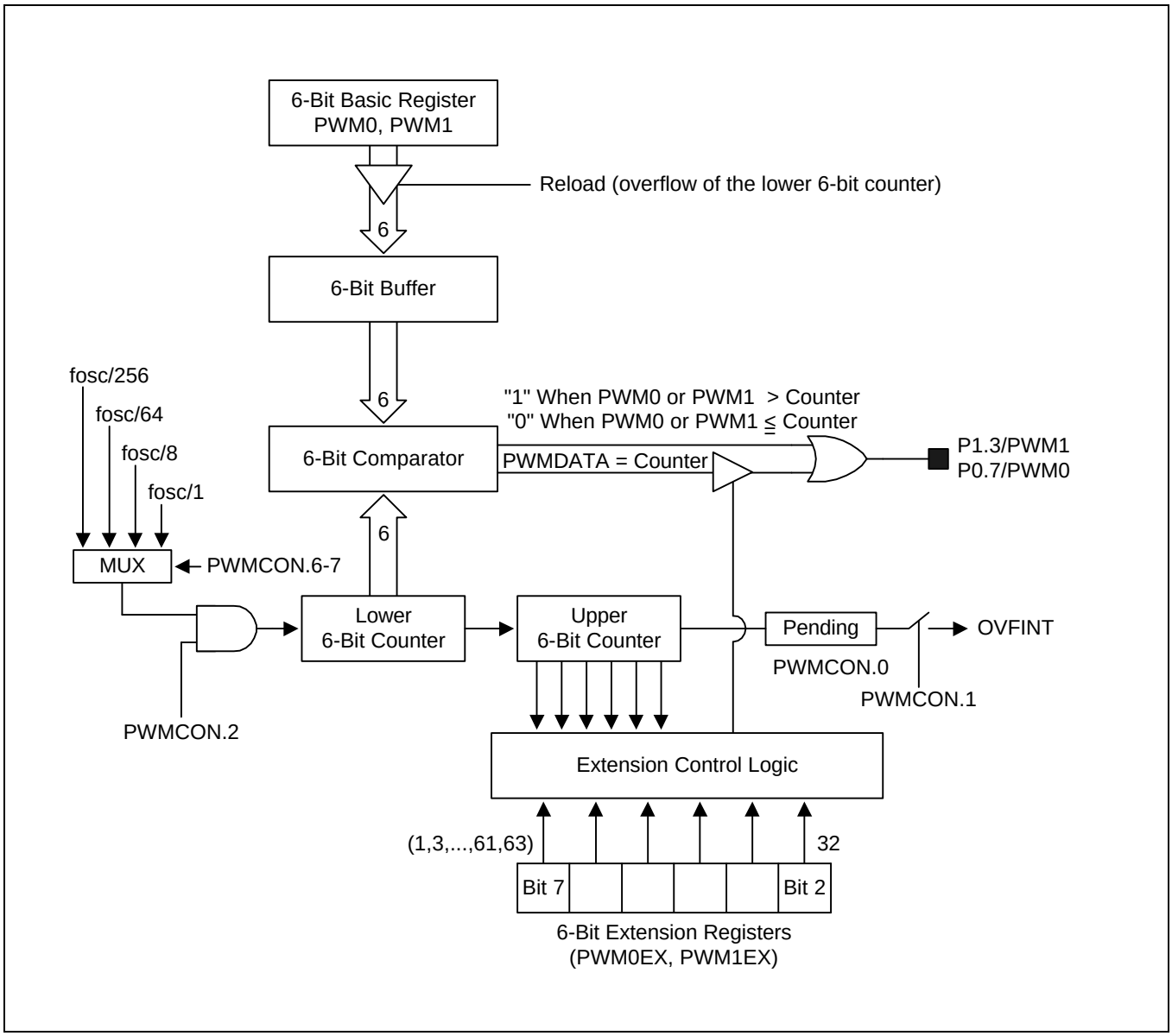


Figure 13-4. PWM/Capture Module Functional Block Diagram

☞ PROGRAMMING TIP — Programming the PWM Module to Sample Specifications

```
        ORG     0000H

        VECTOR  00H, INT_4208       ;  S3C9428 has only one interrupt vector

        ORG     0100H
INITIAL:

        LD      SYM, #00H           ;  Global/Fast interrupt disable -> SYM
        LD      BTCON, #10100010B   ;  Watch-dog disable
        LD      CLKCON, #00011000B  ;  non-divided CPU clock
        LD      SP, #0C0H           ;  9428 → 00–BF (After decrease, push data)

        •
        •
        LD      P0CONH, #0C0H       ;  P0.7 PWM0 output

        LD      PWM0EX, #0          ;  Extension register setting
        LD      PWM0, #20H          ;  Data register setting
        LD      PWMCON, #00000100B  ;  Start counting
                                    ;  Half dutyPWM wave out to P0.7
        •
        •
        •
        EI

MAIN:

        •
        •
        •
        CALL    SUB_ROUTINE
        •
        •
        •
        JP      MAIN
SUB_ROUTINE:

        NOP
        •
        •
        •
        RET
```

14 SERIAL I/O INTERFACE

OVERVIEW

Serial I/O module, SIO can interface with various types of external devices that require serial data transfer. The components of each SIO function block are:

— 8-bit control register (SIOCON)
— Clock selection logic
— 8-bit data buffer (SIODATA)
— 8-bit prescaler (SIOPS)
— 3-bit serial clock counter
— Serial data I/O pins (SI, SO)
— External clock input pin (SCK)

SIO module can transmit or receive 8-bit serial data at a frequency determined by its corresponding control register settings. To ensure flexible data transmission rates, you can select an internal or external clock source.

PROGRAMMING PROCEDURE

To program the SIO module, follow these basic steps:

1. Configure the I/O pins at port 0 (SO, SCK, SI) by loading the appropriate value to the P1CONL Register.
2. Load an 8-bit value to the SIOCON control register to properly configure the serial I/O module. In this operation, SIOCON.2 must be set to "1" to enable the data shifter.
3. For interrupt generation, set the serial I/O interrupt enable bit (SIOCON.1) to "1".
4. When you the transmit data to the serial buffer, write data to SIODATA and set SIOCON.3 to 1, the shift operation starts.
5. When the shift operation (transmit/receive) is completed, the SIO pending bit (SIOCON.0) is set to "1" and an SIO interrupt request is generated.

SERIAL I/O CONTROL REGISTERS (SIOCON)

The control registers for serial I/O interface, SIOCON, is located at F0H. It has the control settings for SIO module.

— Clock source selection (internal or external) for shift clock
— Interrupt enable
— Edge selection for shift operation
— Clear 3-bit counter and start shift operation
— Shift operation (transmit) enable
— Mode selection (transmit/receive or receive-only)
— Data direction selection (MSB first or LSB first)

A reset clears the SIOCON value to "00H". This configures the corresponding module with an internal clock source at the SCK, selects receive-only operating mode, and clears the 3-bit counter. The data shift operation and the interrupt are disabled. The selected data direction is MSB-first.

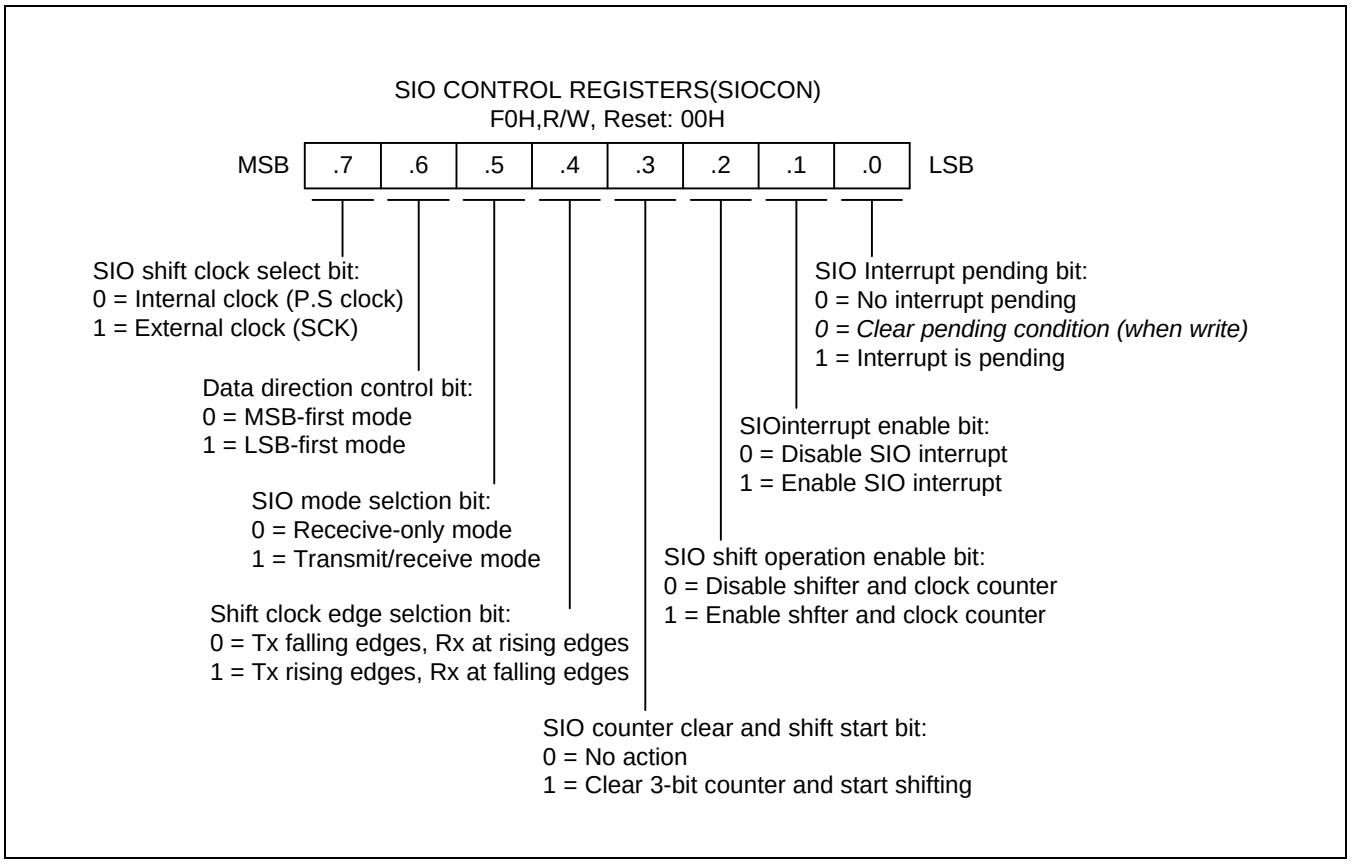


Figure 14-1. Serial I/O Interface Control Register (SIOCON)

SIO PRESCALER REGISTER (SIOPS)

The control register for serial I/O interface module, SIOPS is located at F1H.
The value stored in the SIO prescaler registers, SIOPS, lets you determine the SIO clock rate (baud rate) as follows:

Baud rate = Input clock(Xin/2) / 2(pre-scaler value + 1), or external SCK input clock

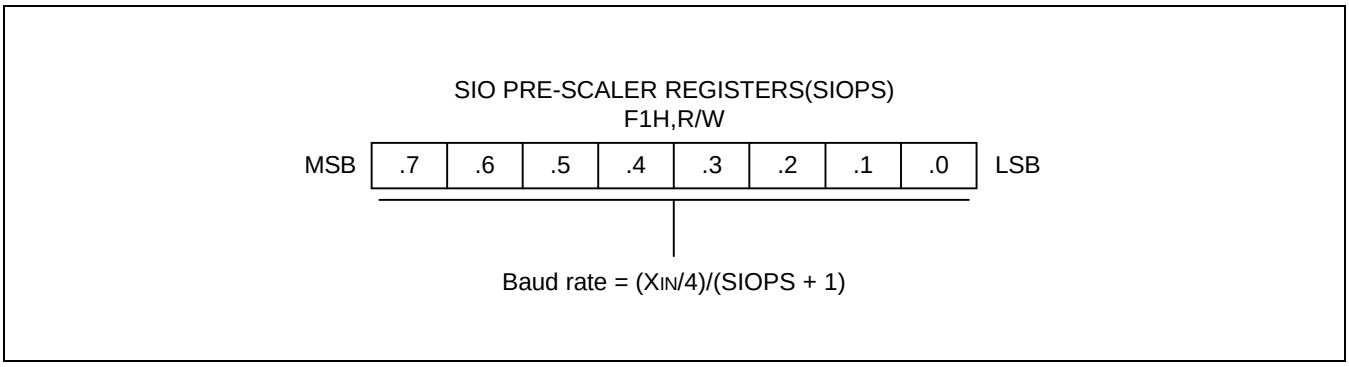


Figure 14-2. SIO Pre-scaler Register (SIOPS)

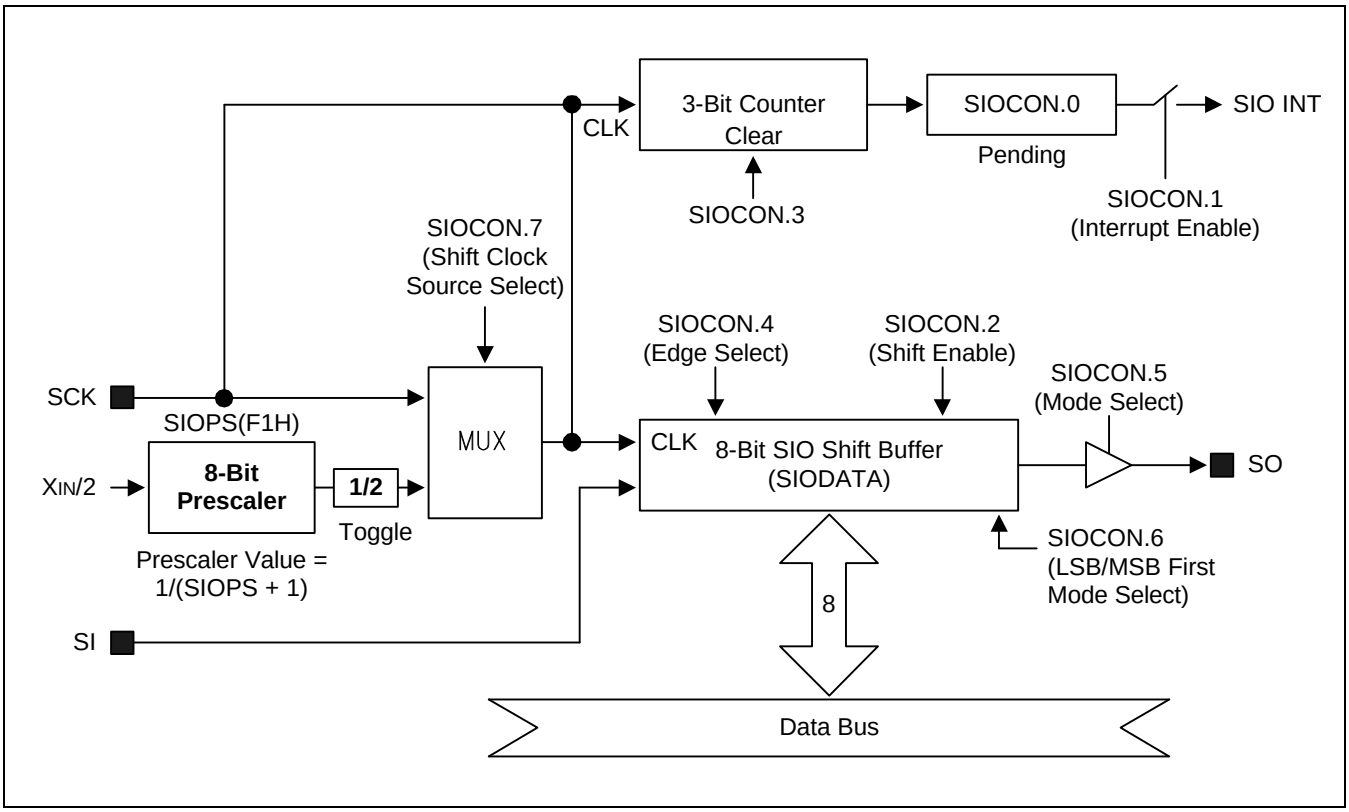


Figure 14-3. SIO Functional Block Diagram

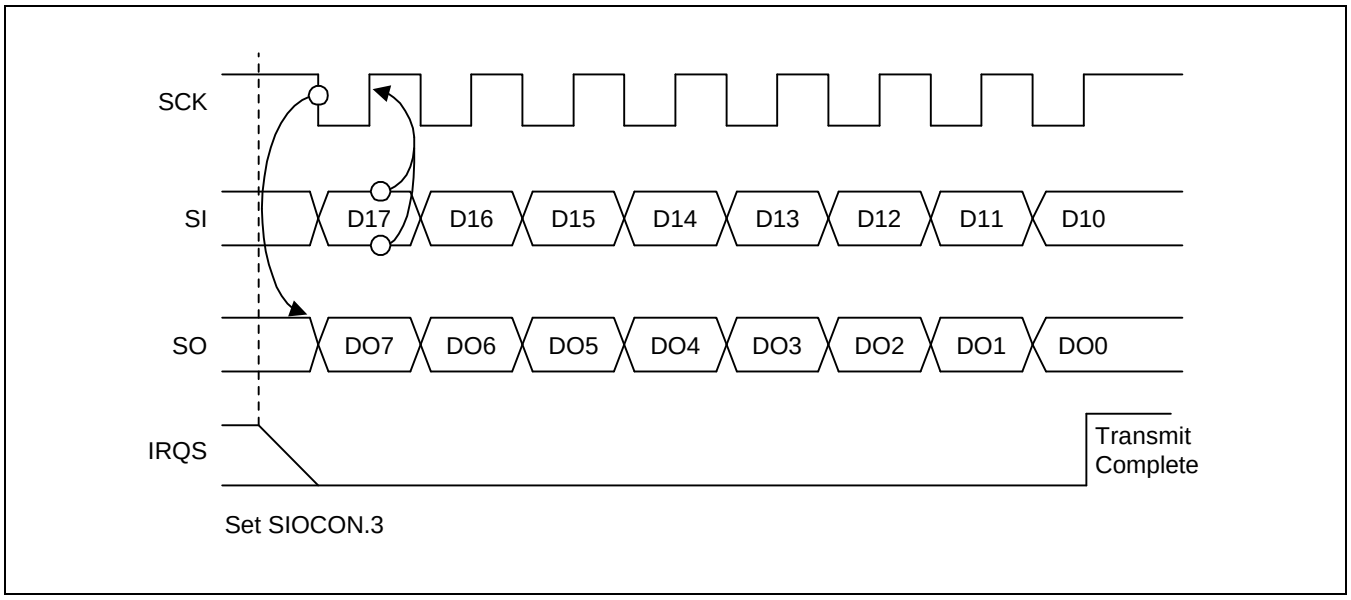


Figure 13-4. Serial I/O Timing in Transmit-Receive Mode (Tx at falling, SIOCON.4 = 0)

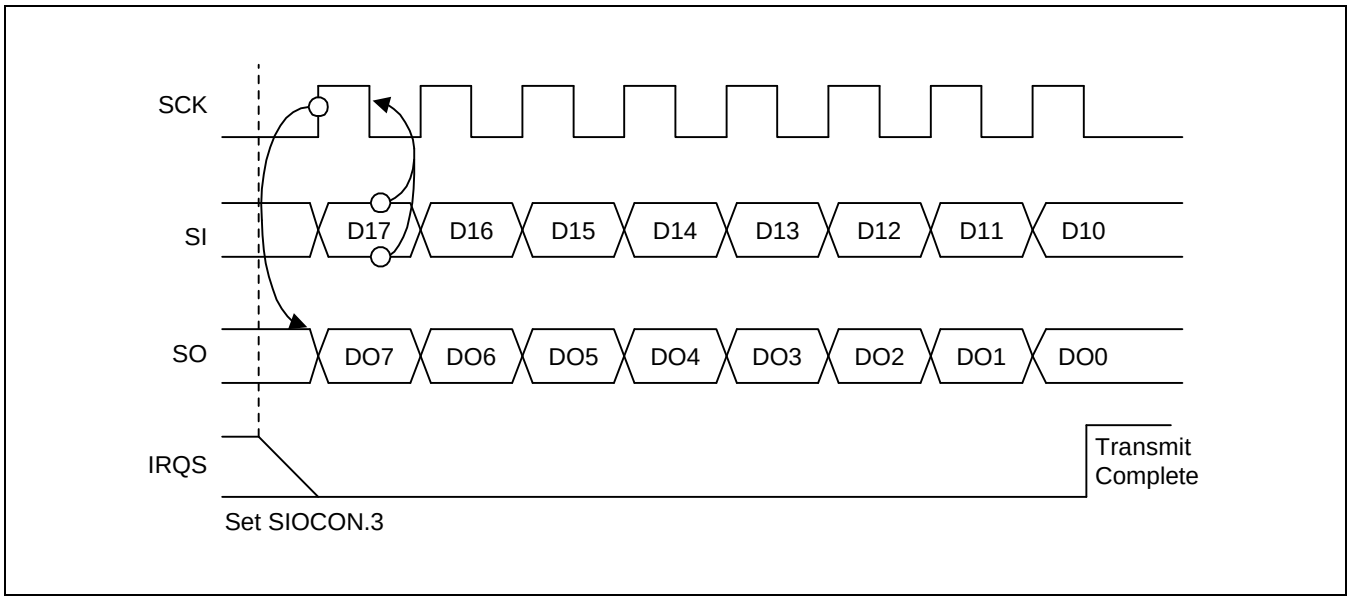


Figure 14-5. Serial I/O Timing in Transmit-Receive Mode (Tx at rising, SIOCON.4 = 1)

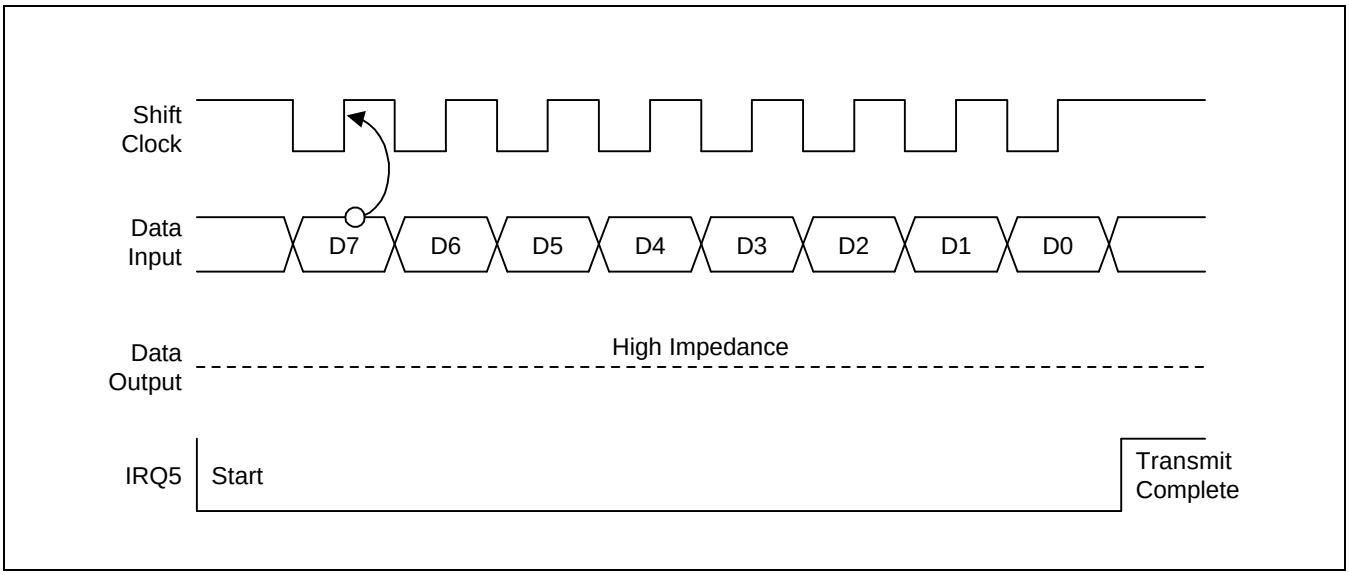


Figure 14-6. Serial I/O Timing in Receive-Only Mode

☞ PROGRAMMING TIP — SIO

```
                ORG     0000H

                VECTOR  00H, INT_4208       ;  S3C9428 has only one interrupt vector

                ORG     0100H
INITIAL:

                LD      SYM, #00H           ;  Global/Fast interrupt disable -> SYM
                LD      BTCON, #10100010B   ;  Watch-dog disable
                LD      CLKCON, #00011000B  ;  non-divided CPU clock
                LD      SP, #0C0H           ;  9428 → 00~BF (After decrease, push data)

                •
                •
                LD      P0CONL, #10010101B  ;  0.2~0.0 – SIO setting
                •
                •
                LD      SIOCON, #00100110B  ;  Enable SIO/Interrupt
                LD      SIOPS, #20          ;  setting baud rate
                •
                •
                EI
```

☞ PROGRAMMING TIP — SIO (Continued)

```
MAIN:
            •
            •
            •
            CALL    SUB_SIO             ;  Data transmit routine
            •
            •
            •
            JP      MAIN
SUB_SIO:
            LD      SIODATA, TRANSBUF   ;  1-byte transmission
            OR      SIOCON, #00001000B  ;  Shift start (8-bit transmit)
            •
            •
            RET
INT_4208:                               ;  S3C9428 has just one interrupt vector
            LD      R0, SIOCON
            AND     R0, #00000011B
            CP      R0, #00000011B
            JP      EQ, INT_SIO         ;  SIOCON's pending bit & INT. enable bit check
INT_SIO:
            AND     SIOCON, #11111110   ;  Pending bit clear
            •
            •
            •
            IRET
```

15 IIC-BUS INTERFACE

OVERVIEW

The S3C9424/C9428/P9428 microcontrollers support a multi-master IIC-bus serial interface. A dedicated serial data line (SDAT) and a serial clock line (SCLK) carry information between bus masters and peripheral devices which are connected to the IIC-bus. The SDAT and SCLK lines are bi-directional.

In multi-master IIC-bus mode, multiple S3C9424/C9428/P9428 microcontrollers can receive or transmit serial data to or from slave devices. The master S3C9424/C9428/P9428 which initiates a data transfer over the IIC-bus is responsible for terminating the transfer. Standard bus arbitration functions are supported.

To control multi-master IIC-bus operations, you write values to the following registers:

— IIC-bus control register, ICCR

— IIC-bus control/status register, ICSR

— IIC-bus Tx/Rx data shift register, IDSR

— IIC-bus address register, IAR

When the IIC-bus is free, the SDAT and SCLK lines are both at High level. A High-to-Low transition of SDAT initiates a Start condition. A Low-to-High transition of SDAT while SCLK remains steady at High level initiates a Stop condition.

Start and Stop conditions are always generated by the bus master. A 7-bit address value in the first data byte that is put onto the bus after the Start condition is initiated determines which slave device the bus master selects. The 8th bit determines the direction of the transfer (read or write).

Every data byte that is put onto the SDAT line must total eight bits. The number of bytes which can be sent or received per bus transfer operation is unlimited. Data is always sent most-significant bit (MSB) first and every byte must be immediately followed by an acknowledge (ACK) bit.

MULTI-MASTER IIC-BUS CONTROL REGISTER (ICCR)

The multi-master IIC-bus control register, ICCR, is located at address F2H. It is read/write addressable. ICCR settings control the following IIC-bus functions:

— CPU acknowledge signal (ACK) enable or suppress

— IIC-bus clock source selection (fosc/16 or fosc/512)

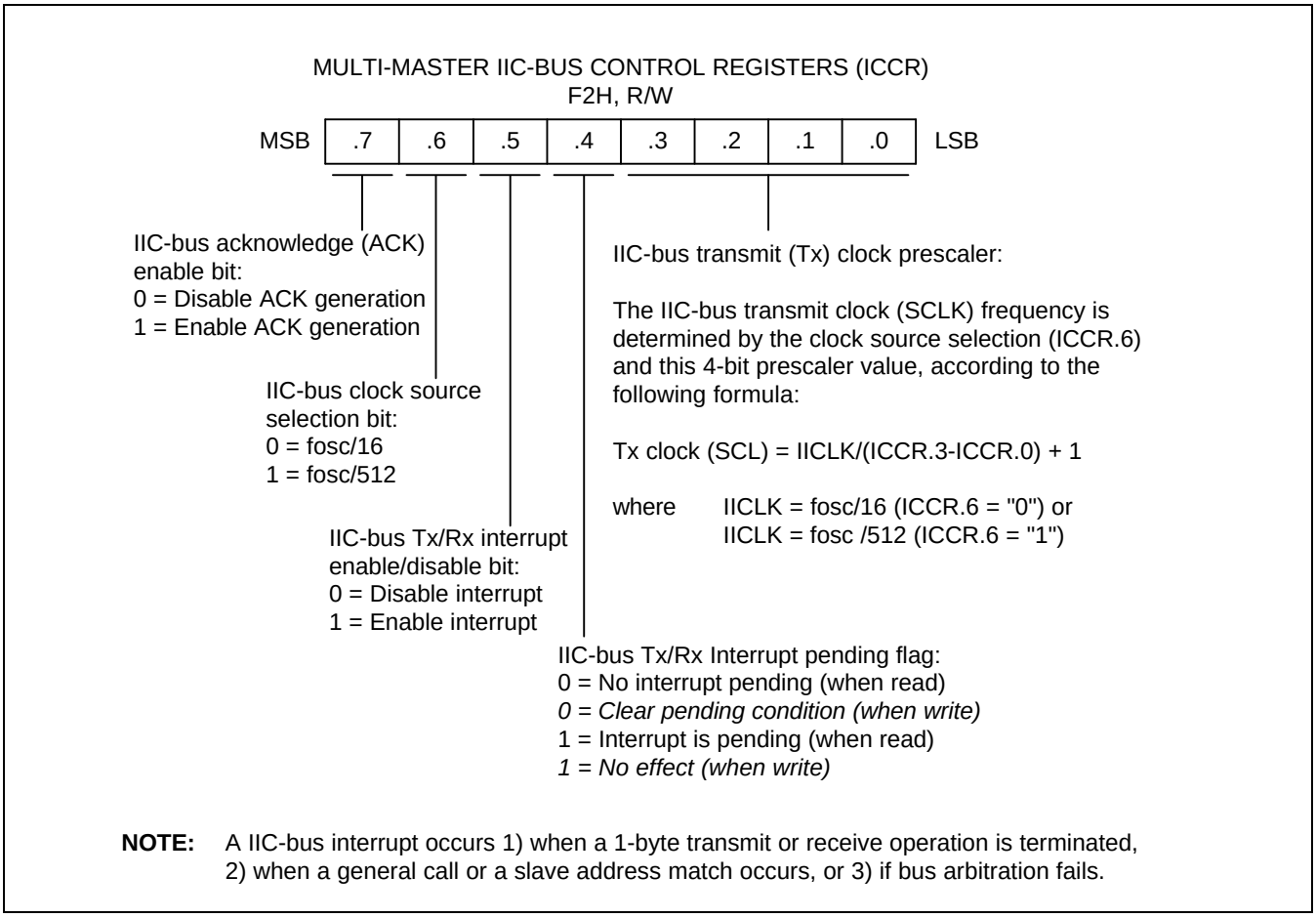
— Transmit/receive interrupt enable or disable

— Transmit/receive interrupt pending control

— 4-bit prescaler for the serial transmit clock (SCLK)

In the S3C9424/C9428/P9428 interrupt structure, the IIC-bus Tx/Rx interrupt is assigned level IRQ2, vector F8H. To enable this interrupt, you set ICCR.5 to “1”. Program software can then poll the IIC-bus Tx/Rx interrupt pending bit (ICCR.4) to detect IIC-bus receive or transmit requests. When the CPU acknowledges the interrupt request from the IIC-bus, the interrupt service routine must clear the interrupt pending condition by writing a “0” to ICCR.4.

The SCL frequency is determined by the IIC-bus clock source selection (fosc/16 or fosc/512) and the 4-bit prescaler value in the ICCR register (see Figure 15-1).



NOTE: A IIC-bus interrupt occurs 1) when a 1-byte transmit or receive operation is terminated, 2) when a general call or a slave address match occurs, or 3) if bus arbitration fails.

Figure 15-1. Multi-Master IIC-Bus Control Register (ICCR)

Table 15-1. Sample Timing Calculations for the IIC-Bus Transmit Clock (SCL)

ICCR.3-ICCR.0 Value	IICLK (ICCR.3-ICCR.0 Settings + 1)	(fosc = 8 MHz) ICCR.6 = 0 (fosc/16) IICLK = 500 kHz	(fosc = 8 MHz) ICCR.6 = 1 (fosc/512) IICLK = 15.625 kHz
0000	IICLK/1	400 kHz (note)	15.625 kHz
0001	IICLK/2	250 kHz	7.1825 kHz
0010	IICLK/3	116.7 kHz	5.2038 kHz
0011	IICLK/4	125 kHz	3.9063 kHz
0100	IICLK/5	100 kHz	3.1250 kHz
0101	IICLK/6	83.3 kHz	2.6042 kHz
0110	IICLK/7	71.4 kHz	2.2321 kHz
0111	IICLK/8	62.5 kHz	1.9531 kHz
1000	IICLK/9	55.6 kHz	1.7361 kHz
1001	IICLK/10	50 kHz	1.5625 kHz
1010	IICLK/11	45.5 kHz	1.4205 kHz
1011	IICLK/12	41.7 kHz	1.3021 kHz
1100	IICLK/13	38.5 kHz	1.2019 kHz
1101	IICLK/14	35.7 kHz	1.1160 kHz
1110	IICLK/15	33.3 kHz	1.0417 kHz
1111	IICLK/16	31.25 kHz	0.9766 kHz

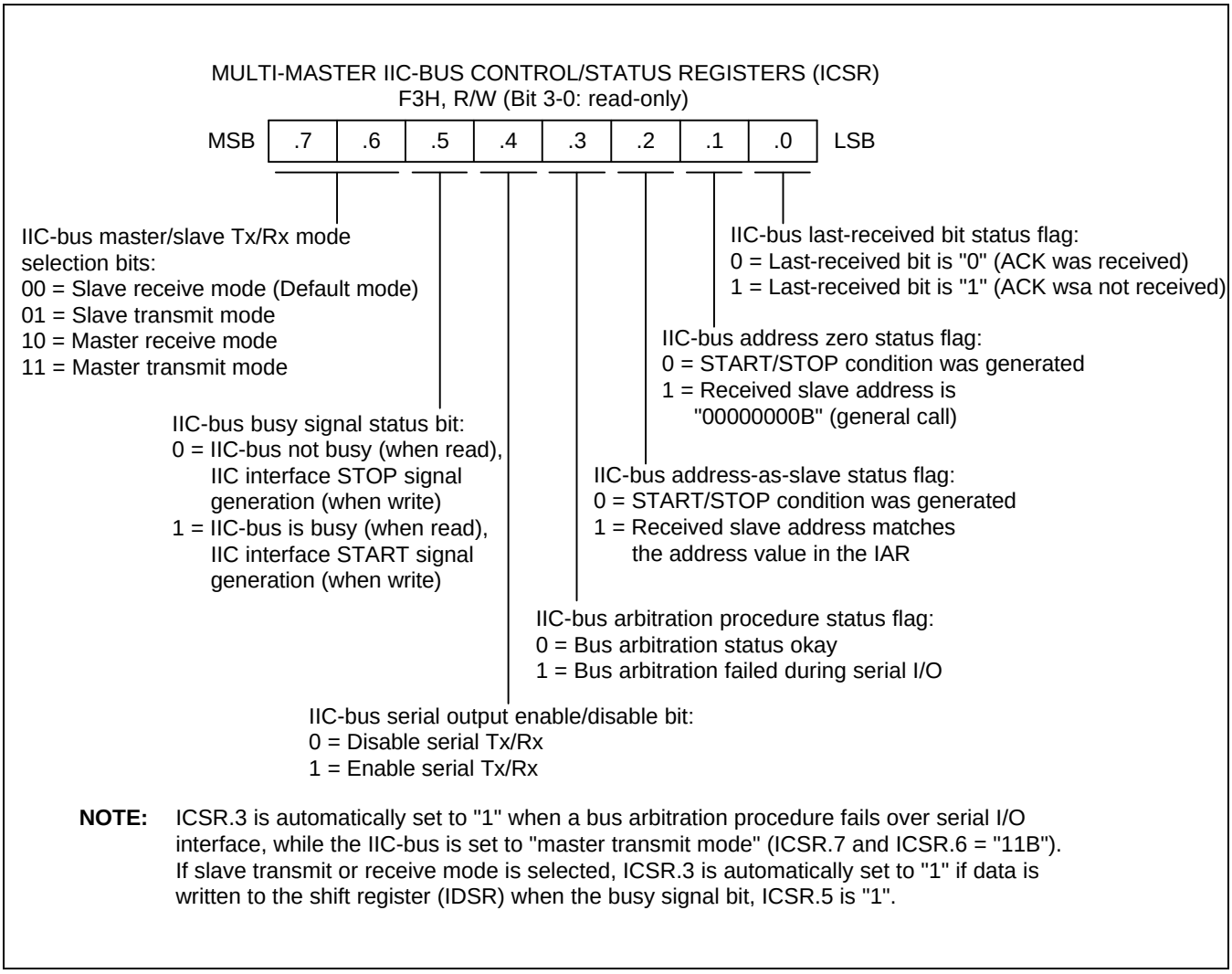
NOTE: Max. IICLK = 400 kHz.

MULTI-MASTER IIC-BUS CONTROL/STATUS REGISTER (ICSR)

The multi-master IIC-bus control/status register, ICSR, is located at address F3H. Four bits in this register, ICSR.3-ICSR.0, are read-only status flags.

ICSR register settings are used to control or monitor the following IIC-bus functions (see Figure 15-2):

— Master/slave transmit or receive mode selection
— IIC-bus busy status flag
— Serial output enable/disable
— Failed bus arbitration procedure status flag
— Slave address/address register match or general call received status flag
— Slave address 00000000B (general call) received status flag
— Last received bit status flag (not ACK = “1”, ACK = “0”)



NOTE: ICSR.3 is automatically set to "1" when a bus arbitration procedure fails over serial I/O interface, while the IIC-bus is set to "master transmit mode" (ICSR.7 and ICSR.6 = "11B"). If slave transmit or receive mode is selected, ICSR.3 is automatically set to "1" if data is written to the shift register (IDSR) when the busy signal bit, ICSR.5 is "1".

Figure 15-2. Multi-Master IIC-Bus Control/Status Register (ICSR)

MULTI-MASTER IIC-BUS TRANSMIT/RECEIVE DATA SHIFT REGISTER (IDSR)

The IIC-bus data shift register, IDSR, is located at address F5H. In a transmit operation, data that is written to the IDSR is transmitted serially, MSB first. (For receive operations, the input data is written into the IDSR register LSB first.)

The ICSR.4 setting enables or disables serial transmit/receive operations. When ICSR.4 = “1”, data can be written to the shift register. The IIC-bus shift register can, however, be read at any time, regardless of the current ICSR.4 setting.

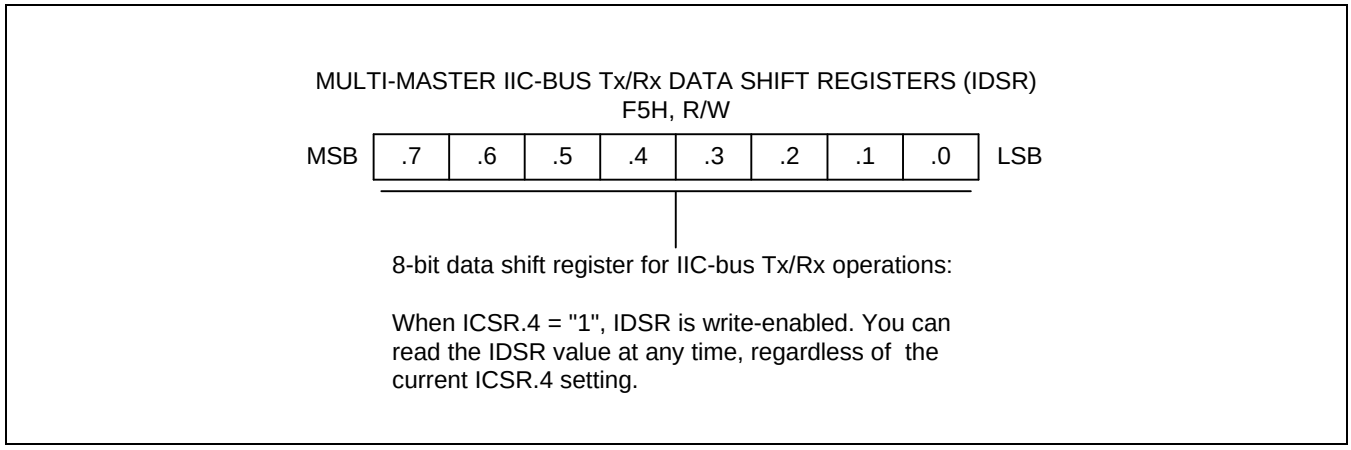


Figure 15-3. Multi-Master IIC-Bus Tx/Rx Data Shift Register (IDSR)

MULTI-MASTER IIC-BUS ADDRESS REGISTER (IAR)

The address register for the IIC-bus interface, IAR, is located at address F4H. It is used to store a latched 7-bit slave address. This address is mapped to IAR.7-IAR.1; bit 0 is not used (see Figure 15-4).

The latched slave address is compared to the next received slave address. If a match condition is detected, and if the latched value is 00000000B, a general call status is detected.

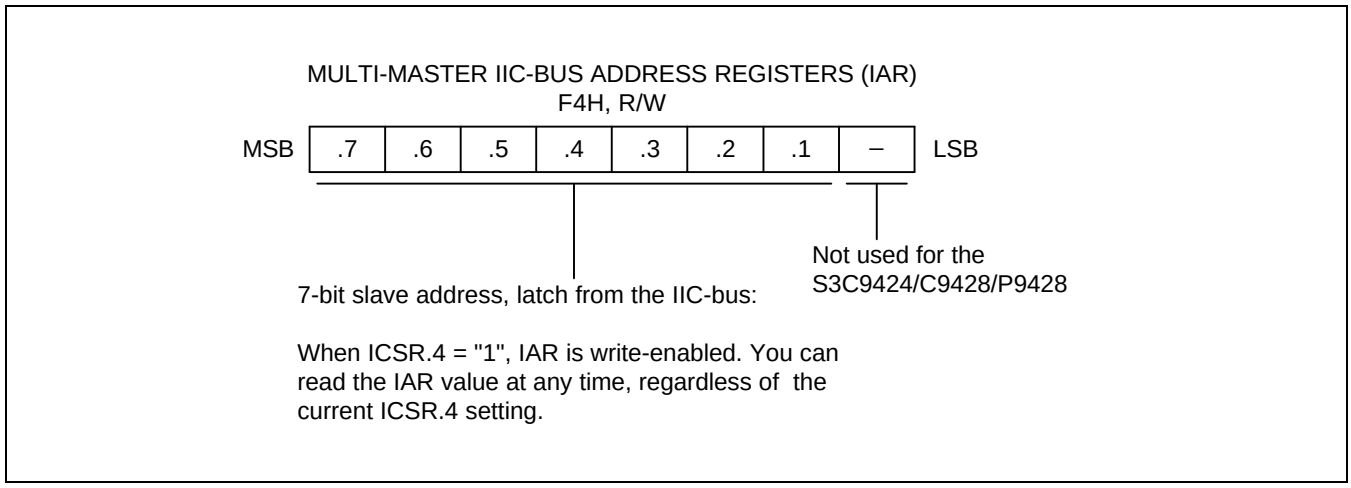
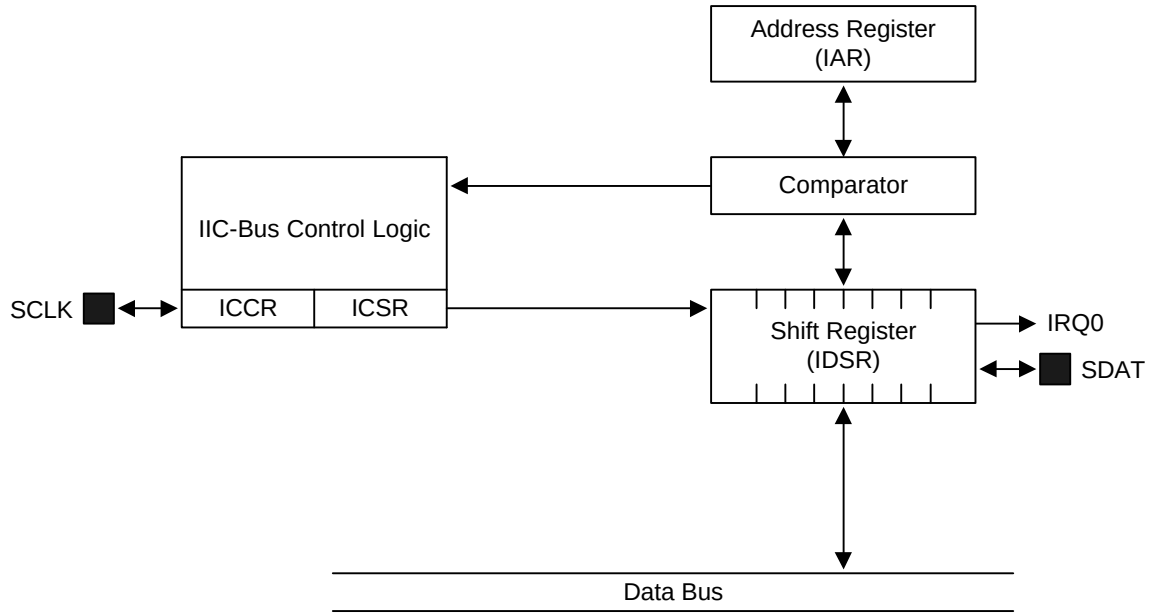


Figure 15-4. Multi-Master IIC-Bus Address Register (IAR)



NOTE: The IIC-bus interrupt (IRQ0) is generated when a 1-byte receive or transmit operation is terminated before the shift operation has been completed.

Figure 15-5. IIC-Bus Block Diagram

THE IIC-BUS INTERFACE

The S3C9424/C9428/P9428 IIC-bus interface has four operating modes:

— Master transmitter mode
— Master receive mode
— Slave transmitter mode
— Slave receive mode

Functional relationships between these operating modes are described below.

START AND STOP CONDITIONS

When the IIC-bus interface is inactive, it is in slave mode. The interface is therefore always in slave mode when a start condition is detected on the SDAT line. (A start condition is a High-to-Low transition of the SDAT line while the clock signal, SCLK, is High level.) When the interface enters master mode, it initiates a data transfer and generates the SCLK signal.

A start condition initiates a one-byte serial data transfer over the SDAT line and a stop condition ends the transfer. (A stop condition is a Low-to-High transition of the SDAT line while SCLK is High level.) Start and stop conditions are always generated by the master. The IIC-bus is “busy” when a start condition is generated. A few clocks after a stop condition is generated, the IIC-bus is again “free”.

When a master initiates a start condition, it sends its slave address onto the bus. The address byte consists of a 7-bit address and a 1-bit transfer direction indicator (that is, write or read). If bit 8 is “0”, a transmit operation (write) is indicated; if bit 8 is “1”, a request for data (read) is indicated.

The master ends the indicated transfer operation by transmitting a stop condition. If the master wants to continue sending data over the bus, it can the generate another slave address and another start condition. In this way, read-write operations can be performed in various formats.

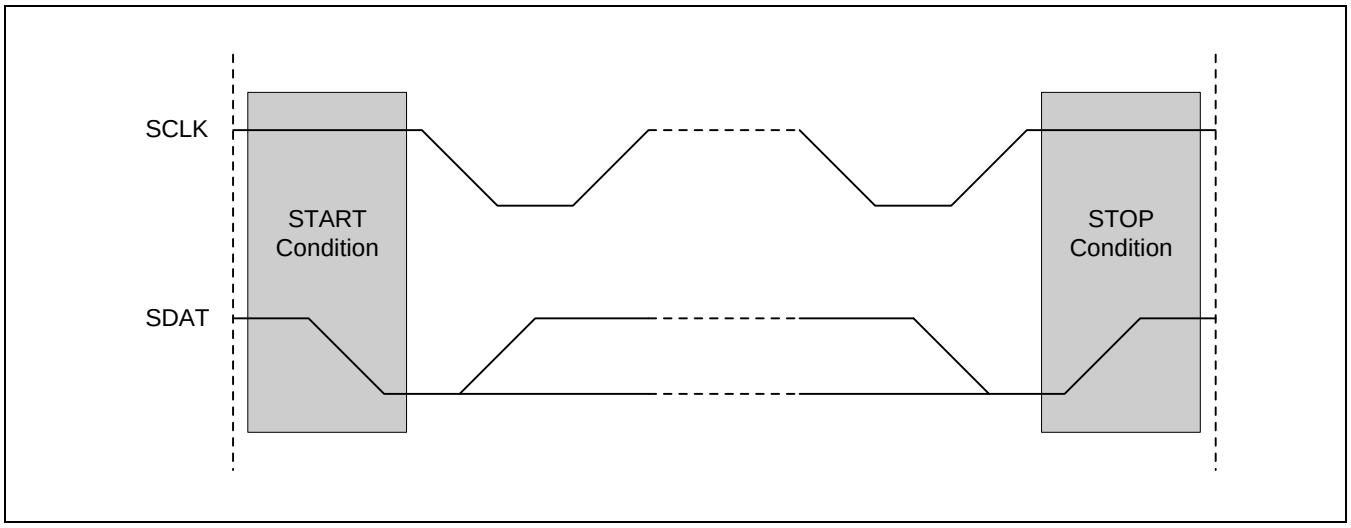


Figure 15-6. Start and Stop Conditions

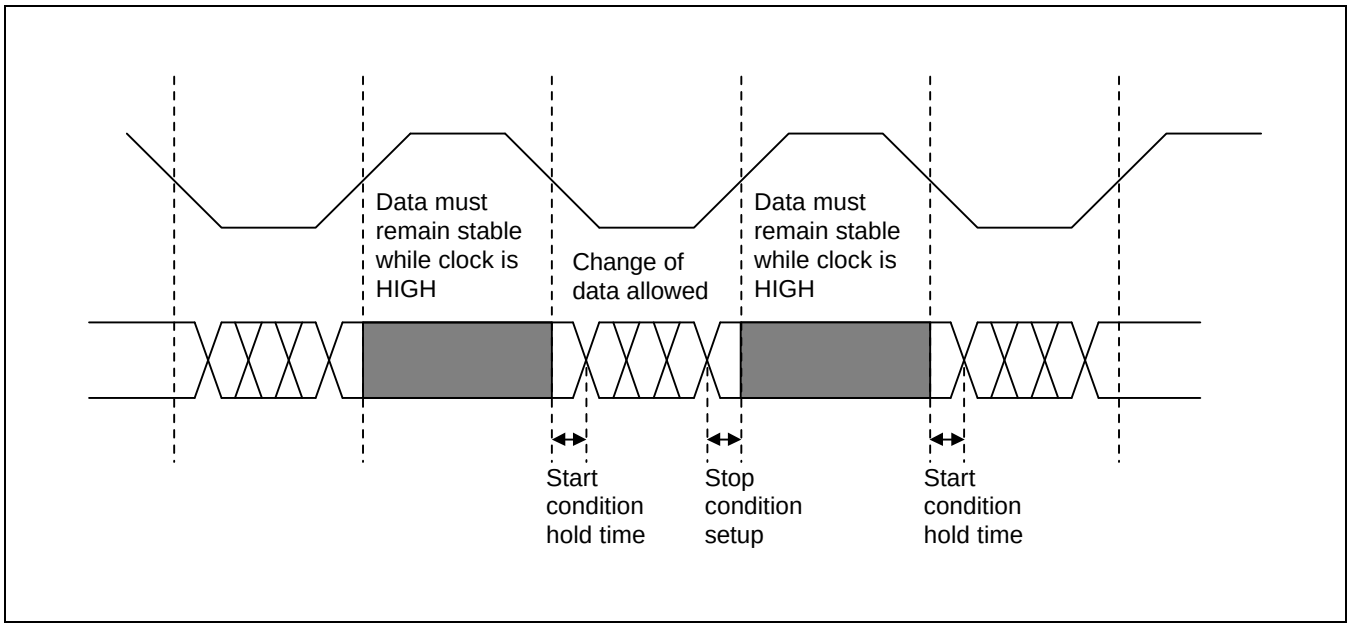
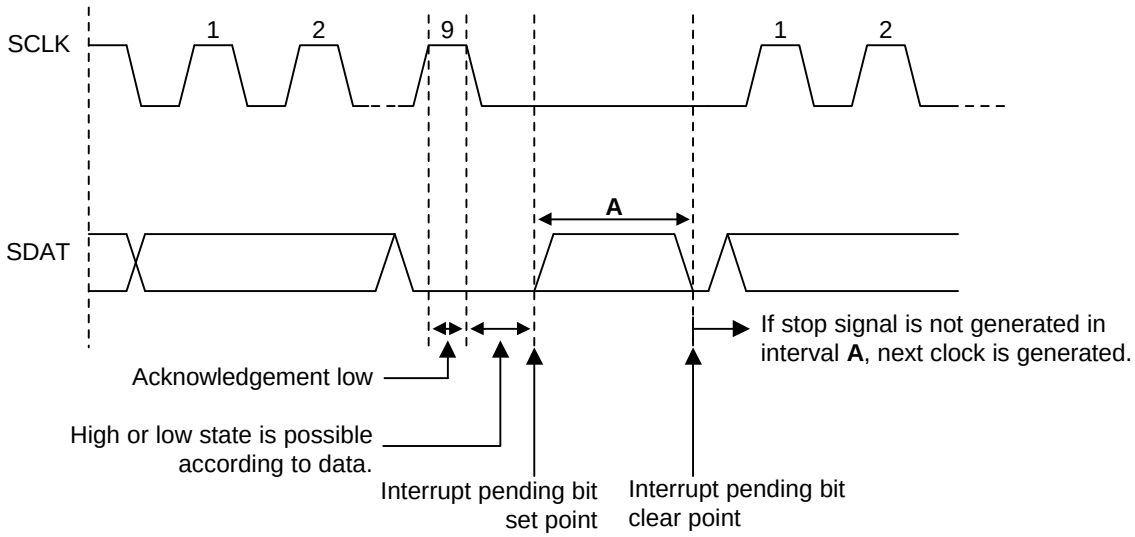


Figure 15-7. Input Data Protocol

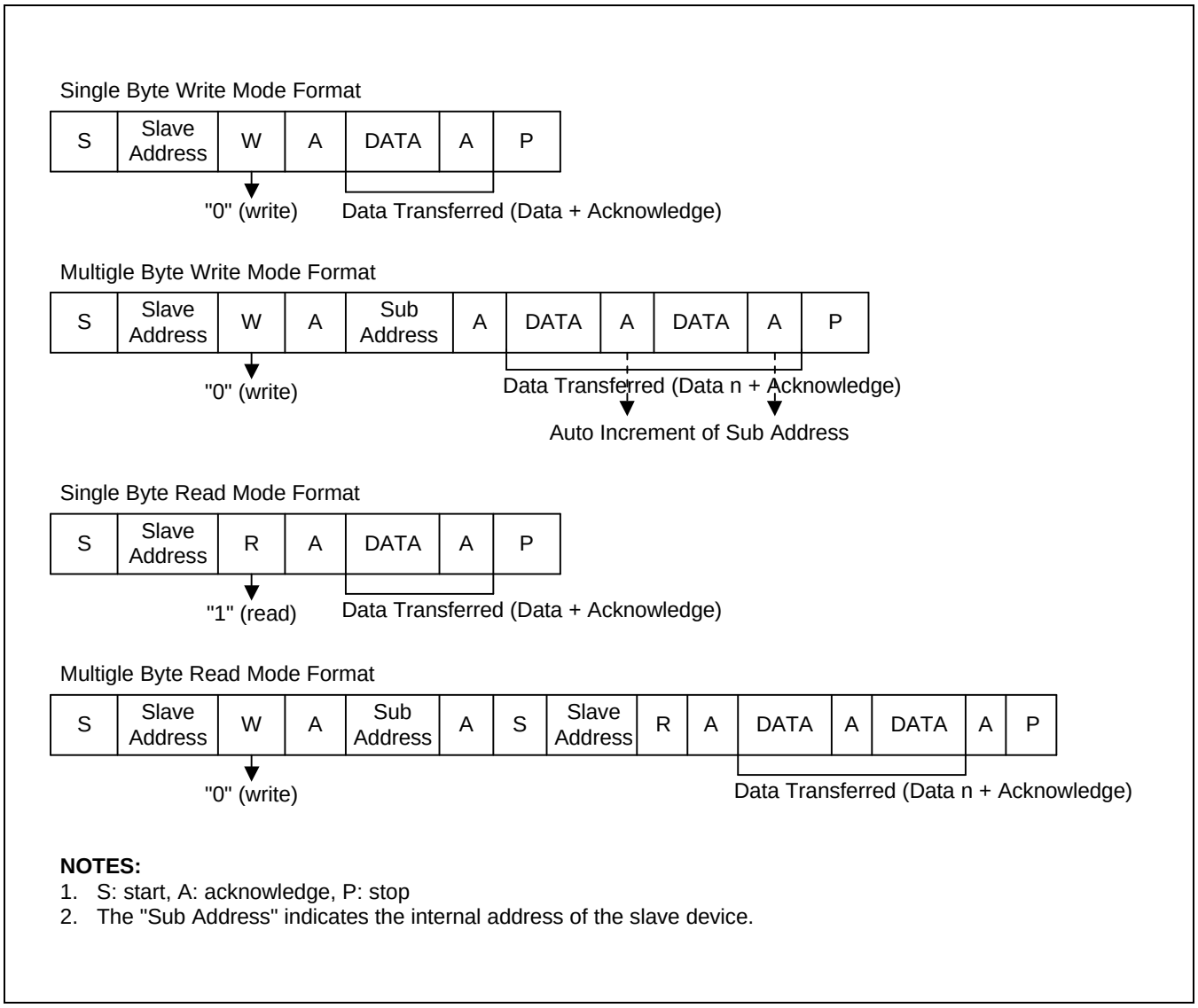


NOTE: In IIC operation, interrupt pending information is indispensable, therefore it is not possible to control IIC operation only in Main routine without interrupt.

Figure 15-8. Interrupt Pending information

DATA TRANSFER FORMATS

Every byte put on the SDAT line must be eight bits in length. The number of bytes which can be transmitted per transfer is unlimited. The first byte following a start condition is the address byte. This address byte is transmitted by the master when the IIC-bus is operating in master mode. Each byte must be followed by an acknowledge (ACK) bit. Serial data and addresses are always sent MSB first.



NOTES:
1. S: start, A: acknowledge, P: stop
2. The "Sub Address" indicates the internal address of the slave device.

Figure 15-9. IIC-Bus Interface Data Formats

ACK SIGNAL TRANSMISSION

To complete a one-byte transfer operation, the receiver must send an ACK bit to the transmitter. The ACK pulse occurs at the ninth clock of the SCLK line (eight clocks are required to complete the one-byte transfer). The clock pulse required for the transmission of the ACK bit is always generated by the master.

The transmitter releases the SDAT line (that is, it sends the SDAT line High) when the ACK clock pulse is received. The receiver must drive the SDAT line Low during the ACK clock pulse so that SDAT is Low during the High period of the ninth SCLK pulse.

The ACK bit transmit function can be enabled and disabled by software (ICCR.7). However, the ACK pulse on the ninth clock of SCLK is required to complete a one-byte data transfer operation.

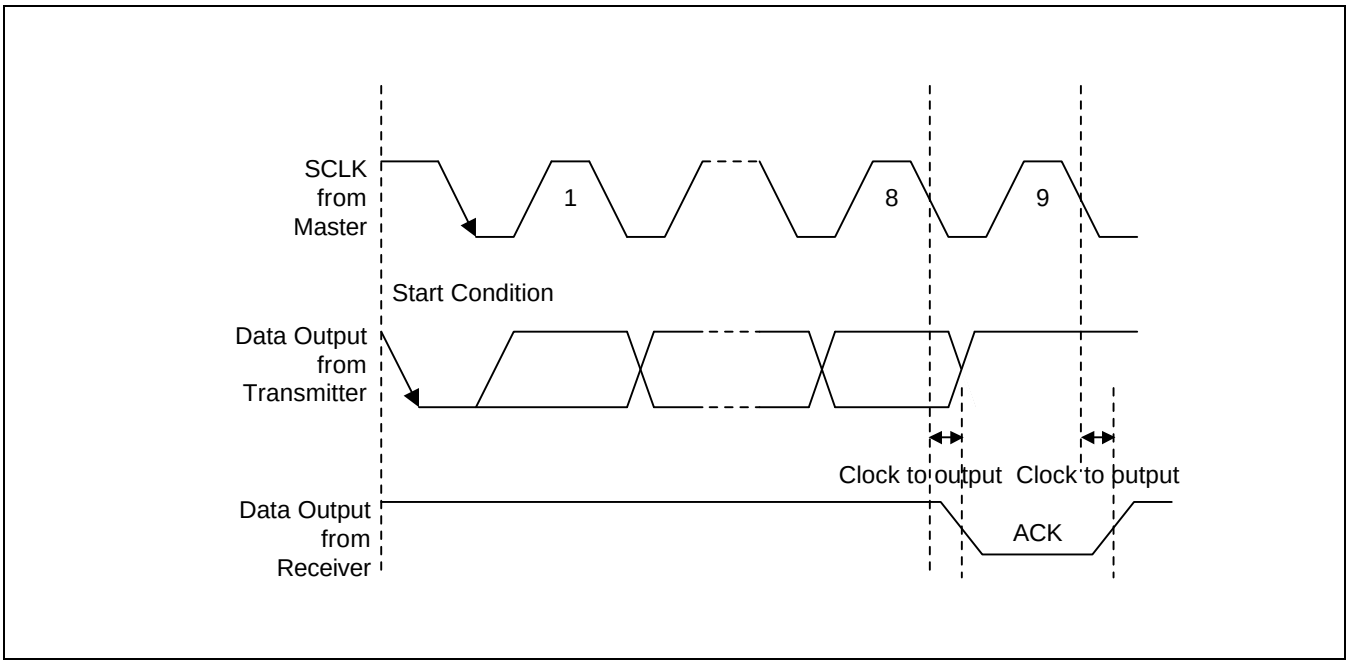


Figure 15-10. Acknowledge Response from Receiver

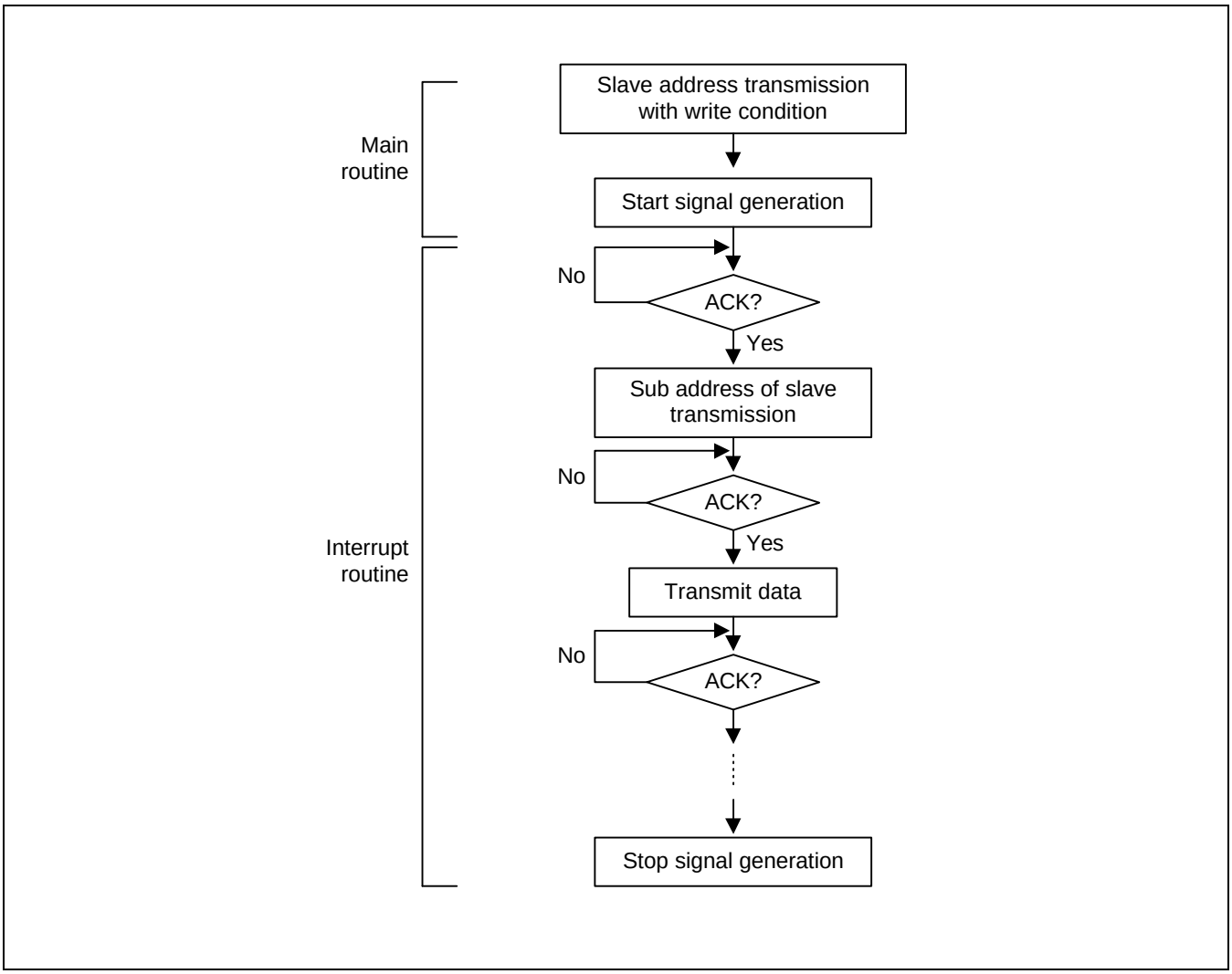


Figure 15-11. Write Operation Sequence

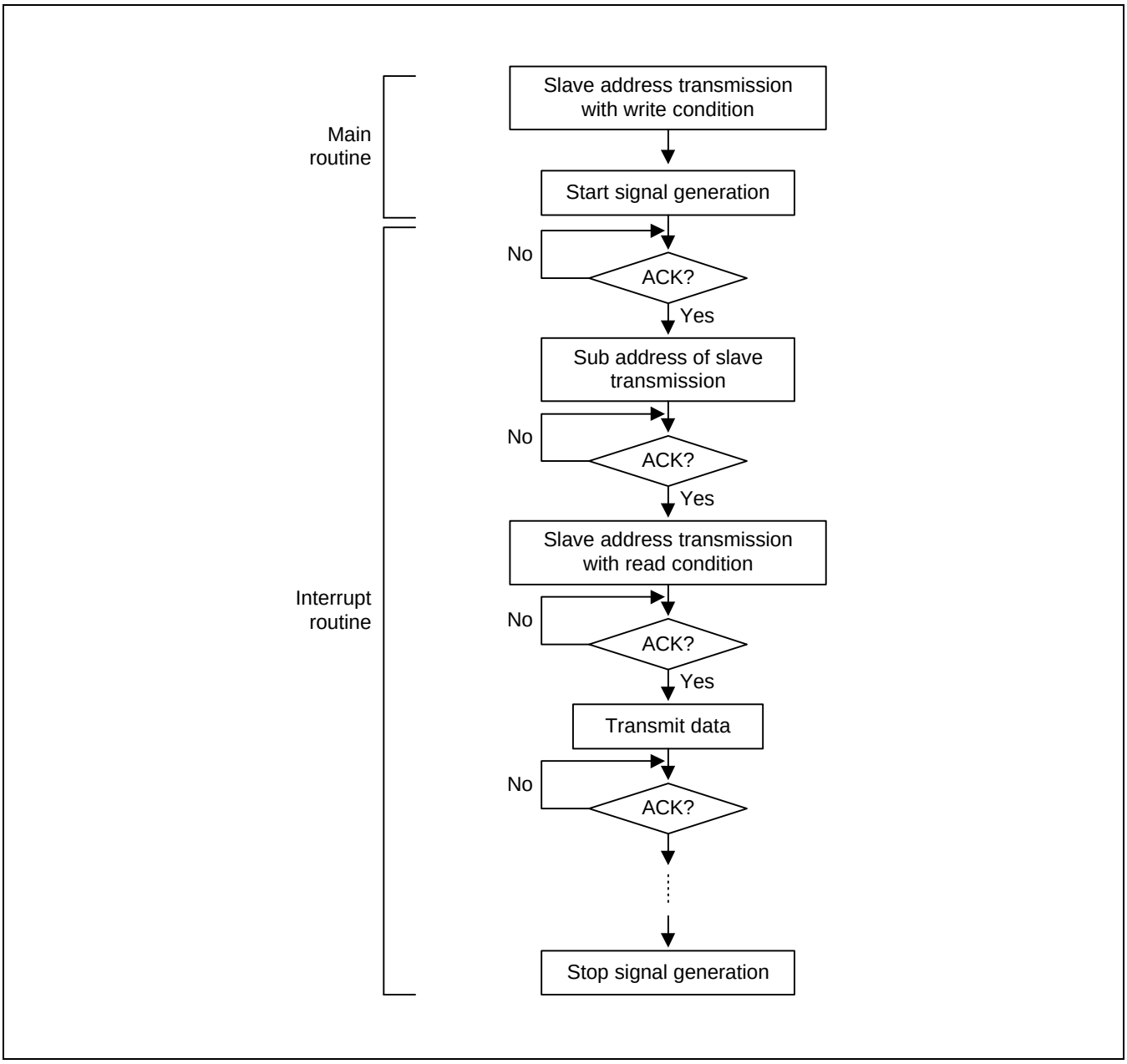


Figure 15-12. Read Operation Sequence

READ-WRITE OPERATIONS

When operating in transmitter mode, the IIC-bus interface interrupt routine waits for the master (the S3C9424/C9428/P9428) to write a data byte into the IIC-bus data shift register (IDSR). To do this, it holds the SCL line Low prior to transmission.

In receive mode, the IIC-bus interface waits for the master to read the byte from the IIC-bus data shift register (IDSR). It does this by holding the SCLK line Low following the complete reception of a data byte.

BUS ARBITRATION PROCEDURES

Arbitration takes place on the SDAT line to prevent contention on the bus between two masters. If a master with a SDAT High level detects another master with an SDAT active Low level, it will not initiate a data transfer because the current level on the bus does not correspond to its own. The master which loses the arbitration can generate SCLK pulses only until the end of the last-transmitted data byte. The arbitration procedure can continue while data continues to be transferred over the bus.

The first stage of arbitration is the comparison of address bits. If a master loses the arbitration during the addressing stage of a data transfer, it is possible that the master which won the arbitration is attempting to address the master which lost. In this case, the losing master must immediately switch to slave receiver mode.

ABORT CONDITIONS

If a slave receiver does not acknowledge the slave address, it must hold the level of the SDAT line High. This signals the master to generate a stop condition and to abort the transfer.

If a master receiver is involved in the aborted transfer, it must also signal the end of the slave transmit operation. It does this by not generating an ACK after the last data byte received from the slave. The slave transmitter must then release the SDAT to allow a master to generate a stop condition.

CONFIGURING THE IIC-BUS

To control the frequency of the serial clock (SCLK), you program the 4-bit prescaler value in the ICCR register. The IIC-bus interface address is stored in IIC-bus address register, IAR. (By default, the IIC-bus interface address is an unknown value.)

☞ PROGRAMMING TIP — Programming the IIC-Bus Interface

```
                ORG         0000H

                VECTOR      00H, INT_4208       ;  S3C9428 has only one interrupt vector


IICcounter      equ         32H                 ;  Counter the total number of reading/writing
IICFlag         equ         33H                 ;  To check read or write mode
IICFinish       equ         34H                 ;  IIC is completed?
IICBufAddr      equ         35H                 ;  To store Read data (point base address of IIC)
SlaveAddr       equ         36H                 ;  EEPROM's identifier
SubAddr         equ         37H                 ;  Internal memory address of EEPROM to read or
DataNum         equ         38H                 ;  How many data will be read or written?
ReadOrWrite     equ         39H                 ;  Which operation will be executed?
IICBuf          equ         3AH                 ;  (3AH–3FH)


                ORG         0100H
INITIAL:

                LD          SYM, #00H           ;  Global/Fast interrupt disable -> SYM
                LD          BTCON, #10100010B   ;  Watch-dog disable
                LD          CLKCON, #00011000B  ;  non-divided CPU clock
                LD          SP, #0C0H           ;  9428 →00–BF (After decrease, push data)
                •
                •
                •
                EI
MAIN:

                •
                •
                •
                CALL        Sub_IIC
                •
                •
                •
                JP          MAIN
```

☞ PROGRAMMING TIP — Programming the IIC-Bus Interface (Continued)

```
;--------------<< IIC Mode >>

Sub_IIC:

            LD      P2CONH, #0F0H           ;  P2.7/2.6 → IIC setting
            LD      ICCR, #11100000B        ;  ACK enable/INT. enable/prescaler set

            TM      ReadOrWrite, #00000001B;  .0 bit → 1 Read operation
            JP      NZ, Read_Operation

Write_Operation:                            ;  .0 bit → 0 write operation

            LD      SlaveAddr, #0A0H        ;  First (U3) EEPROM address/write
            INC     DataNum                 ;  Add 1 for sub address (SubAddr)

            CALL    Bus_Busy                ;  IIC bus busy?

            LD      IICBufAddr, #IICBuf     ;  buffer address load

            LD      IDSR, SlaveAddr
            LD      ICSR, #11110000B        ;  Master trans/Start signal/Enable IIC

            CALL    Finish_Wait             ;  Wait until operation is completed

            JP      IIC_md_end

Read_Operation:

            LD      SlaveAddr, #0A1H        ;  First (U3) EEPROM address/read
            OR      IICFlag, #00000100B     ;  IIC_Flag.2 → indicate read operation

            CALL    Bus_Busy                ;  IIC bus busy?

            LD      IDSR, #0A0H             ;  First (U3) EEPROM address with write condition
            LD      ICSR, #11110000B        ;  Master trans/Start signal/Enable IIC

            CALL    Finish_Wait             ;  Wait until operation is completed.

IIC_md_end:

            RET
```

☞ PROGRAMMING TIP — Programming the IIC-Bus Interface (Continued)

```
;--------------<< Check IIC Bus >>

Bus_Busy                                    ;  Called by IIC test routine

            LD      R9, ICSR
            TM      R9, #00100000B          ;  bus check
            JP      NZ, Bus_Busy
            RET

Finish_Wait:                                ;  Called by IIC test routine

            NOP
            CP      IICFinish, #0FFH        ;  If IIC operation is finished
            JP      NE, Finish_Wait         ;  IICFinish is set FFH in INT. routine
            CLR     IICFinish               ;  IIC operation completed

            RET

INT_4208:                                   ;  S3C9428 has only one interrupt vector

            •
            •
            LD      R0, ICCR
            AND     R0, #00110000B
            CP      R0, #00110000B
            JP      EQ,INT_IIC              ;  IIC interrupt routine
            •
            •
            •

;--------------<< IIC interrupt service routine >>

INT_IIC:

            TM      IICcounter, #10000000B
            JP      NZ, Read_Byte
            LD      R1, ICSR
            TM      R1, #00000001B          ;  ACK check
            JP      Z, ACK_OK
            OR      IICFlag, #00000010B     ;  → ACK fail

Stop_Condition:

            AND     ICSR, #11011111B        ;  stop condition generation
            LD      IICFlag, #0FFH          ;  → operation end: check it in main routine
            JP      INT_IIC_End
```

☞ **PROGRAMMING TIP — Programming the IIC-Bus Interface (Continued)**

```
ACK_OK:

        CLR     IICFinish
        TM      IICFlag, #00001000B     ;  Test if in read mode
        JP      NZ, Read_Modd

        INC     IICcounter              ;  Update data trans procedure
        CP      IICcounter, #02H
        JP      uge, Data_Load
        LD      IDSR, SubAddr           ;  counter = 1 → sub address (memory in EEPROM)
                                        ;  Slave address was transmitted in main
        JP      INT_IIC_End

Data_Load:

        TM      IICFlag, #00000100B     ; read mode
        JP      NZ, Read_Mod
        LD      R0, IICBufAddr
        CP      IICcounter, #03H
        JP      ult, Next_Byte
        CP      DataNum, IICcounter
        JP      ult, Stop_Condition
        LD      R0, IICBufAddr
        INC     R0                      ;  indirect addressing
        LD      IICBufAddr, R0          ;  update IICBufAddr value

Next_Byte:

        LD      IDSR,@R0
        JP      INT_IIC_End

Read_Mod:

        LD      IDSR, SlaveAddr         ;  slave address with read condition
        LD      ICSR, #10110000B        ;  change to master receiver mode
        AND     IICFlag, #11111011B     ;  clear read flag

        LD      IICBufAddr, #IICBuf     ;  To store received data
        OR      IICFlag, #00001000B     ;  first data read flag set
        JP      INT_IIC_End

Read_Modd:                              ;  After trans slave address with read

        AND     IICFlag, #11110111B     ;  clear first data read flag
        CLR     IICcounter
        OR      IICcounter, #80H        ;  jump to Read_Modd routine without ACK check
        JP      INT_IIC_End
```

☞ PROGRAMMING TIP — Programming the IIC-Bus Interface (Concluded)

```
Read_Byte:                                  ;  data read

        INC     IICcounter
        LD      R1, IICcounter              ;  .7 bit of IICcounter is important
        AND     R1, #01111111B
        CP      R1, DataNum
        JP      ugt, Stop_Condition

        LD      R1, IICBufAddr              ;  Next data read
        LD      @R1, IDSR                   ;  store read data
        INC     IICBufAddr

INT_IIC_End:

        AND     ICCR, #11101111B            ;  pending clear

        IRET
```

16 ELECTRICAL DATA

OVERVIEW

In this section, the following S3C9424/C9428/P9428 electrical characteristics are presented in tables and graphs:

- Absolute maximum ratings
- D.C. electrical characteristics
- A.C. electrical characteristics
- Operating Voltage Range
- Schmitt trigger input characteristics
- Oscillator characteristics
- Oscillation stabilization time
- Data retention supply voltage in Stop mode
- Stop mode release timing when initiated by a RESET
- Power-on RESET circuit characteristics
- A/D converter electrical characteristics
- Zero-crossing detector
- Zero Crossing Waveform Diagram

Table 16-1. Absolute Maximum Ratings

(T_A = 25°C)

Parameter	Symbol	Conditions	Rating	Unit
Supply voltage	V_{DD}	–	– 0.3 to + 6.5	V
Input voltage	V_I	All input ports	– 0.3 to V_{DD} + 0.3	V
Output voltage	V_O	All output ports	– 0.3 to V_{DD} + 0.3	V
Output current high	I_{OH}	One I/O pin active	– 25	mA
		All I/O pins active	– 80	
Output current low	I_{OL}	One I/O pin active	+ 30	mA
		Total pin current for ports 1, 2, 3	+ 100	
		Total pin current for ports 0	+ 200	
Operating temperature	T_A	–	– 40 to + 85	°C
Storage temperature	T_{STG}	–	– 65 to + 150	°C

Table 16-2. D.C. Electrical Characteristics(30SDIP, 32SOP)

(T_A = – 40°C to + 85°C, V_{DD} = 3.0 V to 5.5 V)

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Input high voltage	V_{IH1}	Ports 0, 1, 2 and RESET	V_{DD}= 3.0 to 5.5 V	0.8 V_{DD}	–	V_{DD}	V
	V_{IH3}	X_{IN} and X_{OUT}		V_{DD} – 0.1			
Input low voltage	V_{IL1}	Ports 0, 1, 2 and RESET	V_{DD}= 3.0 to 5.5 V	–	–	0.2 V_{DD}	V
	V_{IL2}	X_{IN} and X_{OUT}				0.1	
Output high voltage	V_{OH}	I_{OH} = – 10 mA ports 0-3	V_{DD}= 4.5 to 5.5 V	V_{DD} – 1.5	V_{DD} – 0.4	–	V
Output low voltage	V_{OL}	I_{OL} = 25 mA port 0-3	V_{DD}= 4.5 to 5.5 V	–	0.4	2.0	V
Input high leakage current	I_{LIH1}	All input pins except I_{LIH2}	V_{IN} = V_{DD}	–	–	1	µA
	I_{LIH2}	X_{IN}, X_{OUT}	V_{IN} = V_{DD}			20	
Input low leakage current	I_{LIL1}	All input pins except I_{LIL2} and RESET	V_{IN} = 0 V	–	–	– 1	µA
	I_{LIL2}	X_{IN}, X_{OUT}	V_{IN} = 0 V			– 20	
Output high leakage current	I_{LOH}	All output pins	V_{OUT} = V_{DD}	–	–	2	µA
Output low leakage current	I_{LOL}	All output pins	V_{OUT} = 0 V	–	–	– 2	µA
Pull-up resistor	R_P	V_{IN} = 0 V Port 0-2	V_{DD} = 5 V	30	47	70	KΩ
		RESET	V_{DD} = 5 V	100	200	350	
Supply current	I_{DD1}	RUN mode 16-MHz CPU clock	V_{DD} = 4.5 to 5.5 V	–	11	20	mA
		4-MHz CPU clock	V_{DD} = 3 V		1.5	4	
	I_{DD2}	Idle mode 16-MHz CPU clock	V_{DD} = 4.5 to 5.5 V	–	3	8	
		4-MHz CPU clock	V_{DD} = 3.3 V		0.5	2	
	I_{DD3}	Stop mode	V_{DD} = 4.5 to 5.5 V	–	65	100	µA
			V_{DD} = 3.3 V		45	80	

NOTE: D.C. electrical values for Supply current (I_{DD1} to I_{DD3}) do not include current drawn through internal pull-up resisters, output port drive current, ZCD and ADC.

Table 16-3. D.C. Electrical Characteristics (28SOP)

(T_A = – 40°C to + 85°C, V_{DD} = 1.8 V to 5.5 V)

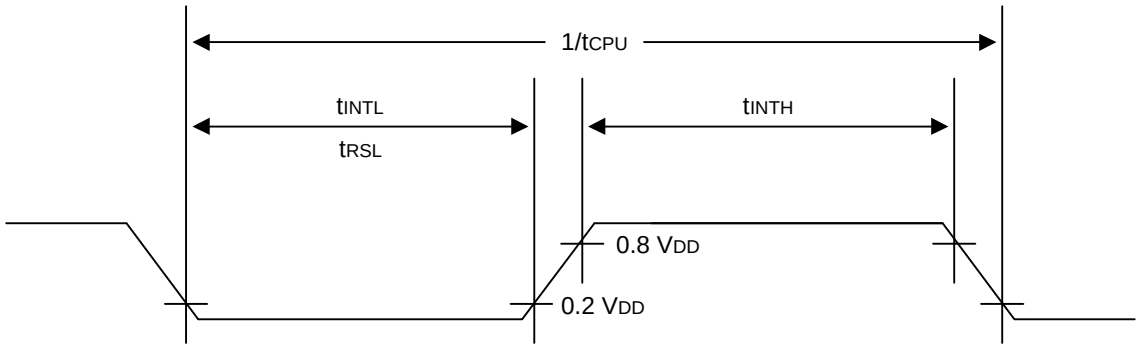
Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Input high voltage	V_{IH1}	Ports 0, 1, 2 and RESET	V_{DD}= 1.8 to 5.5 V	0.8 V_{DD}	–	V_{DD}	V
	V_{IH3}	X_{IN} and X_{OUT}		V_{DD} – 0.1			
Input low voltage	V_{IL1}	Ports 0, 1, 2 and RESET	V_{DD}= 1.8 to 5.5 V	–	–	0.2 V_{DD}	V
	V_{IL2}	X_{IN} and X_{OUT}				0.1	
Output high voltage	V_{OH}	I_{OH} = – 10 mA ports 0-3	V_{DD}= 4.5 to 5.5 V	V_{DD} – 1.0	V_{DD} – 0.4	–	V
Output low voltage	V_{OL}	I_{OL} = 25 mA port 0-3	V_{DD}= 4.5 to 5.5 V	–	0.4	2.0	V
Input high leakage current	I_{LIH1}	All input pins except I_{LIH2}	V_{IN} = V_{DD}	–	–	1	µA
	I_{LIH2}	X_{IN}, X_{OUT}	V_{IN} = V_{DD}			20	
Input low leakage current	I_{LIL1}	All input pins except I_{LIL2} and RESET	V_{IN} = 0 V	–	–	– 1	µA
	I_{LIL2}	X_{IN}, X_{OUT}	V_{IN} = 0 V			– 20	
Output high leakage current	I_{LOH}	All output pins	V_{OUT} = V_{DD}	–	–	2	µA
Output low leakage current	I_{LOL}	All output pins	V_{OUT} = 0 V	–	–	– 2	µA
Pull-up resistor	R_P	V_{IN} = 0 V Port 0-2	V_{DD} = 5 V	30	47	70	KΩ
		RESET	V_{DD} = 5 V	100	200	350	
Supply current	I_{DD1}	RUN mode 16-MHz CPU clock	V_{DD} = 4.5 to 5.5 V	–	11	20	mA
		3-MHz CPU clock	V_{DD} = 1.8 to 2.2 V		1	3	
	I_{DD2}	Idle mode 16-MHz CPU clock	V_{DD} = 4.5 to 5.5 V	–	3	9	
		3-MHz CPU clock	V_{DD} = 1.8 to 2.2 V		0.3	1.0	
	I_{DD3}	Stop mode	V_{DD} = 4.5 to 5.5 V	–	0.1	5	µA
			V_{DD} = 3 V				
			V_{DD} = 1.8 to 2.2 V				

NOTE: D.C. electrical values for Supply current (I_{DD1} to I_{DD3}) do not include current drawn through internal pull-up resisters, output port drive current, ZCD and ADC.

Table 16-4. A.C. Electrical Characteristics

(T_A = –40°C to + 85°C, V_{DD} = 1.8 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Interrupt input high, low width	t_{INTH}, t_{INTL}	Port 1v(INT0, INT1) V_{DD} = 5V ± 10%	–	200	–	ns
RESET input low width	t_{RSL} –	Input V_{DD} = 5V ± 10%	–	1	–	us



NOTE: The unit tcpu means one CPU clock period.

Figure 16-1. Input Timing Measurement Points

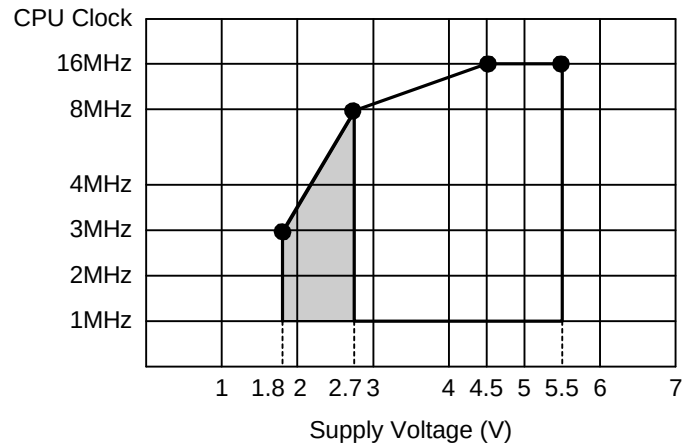


Figure 16-2. Operating Voltage Range (KS86C4204/C4208)

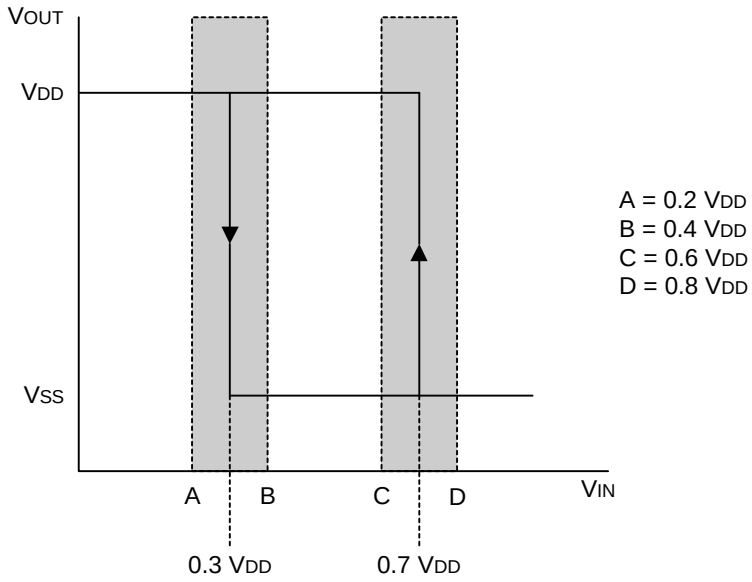


Figure 16-3. Schimtt Trigger Input Characteristic Diagram

Table 16-5. Oscillator Characteristics (30SDIP, 32SOP)

(T_A = – 40°C to + 85°C)

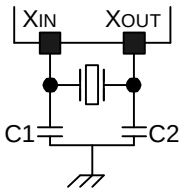
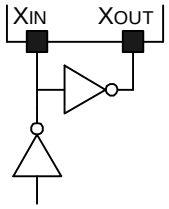
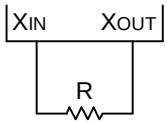
Oscillator	Clock Circuit	Test Condition	Min	Typ	Max	Unit
Main crystal or ceramic		V_{DD} = 4.5 to 5.5 V V_{DD} = 3.0 to 4.5 V	1 1	– –	16 8	MHz
External clock (Main system)		V_{DD} = 4.5 to 5.5 V V_{DD} = 3.0 to 4.5 V	1 1	– –	16 8	
RC oscillator		V_{DD} = 4.75 to 5.25 V Tolerance: 10%	–	4	–	

Table 16-6. Oscillation Stabilization Time (28SOP)

(T_A = – 40°C to + 85°C)

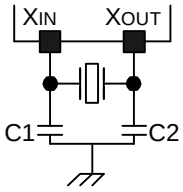
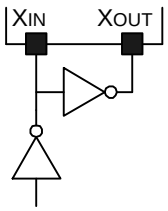
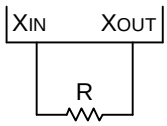
Oscillator	Clock Circuit	Test Condition	Min	Typ	Max	Unit
Main crystal or ceramic		V_{DD} = 4.5 to 5.5 V V_{DD} = 2.7 to 4.5 V V_{DD} = 1.8 to 2.7 V	1 1 1	– – –	16 8 3	MHz
External clock (Main system)		V_{DD} = 4.5 to 5.5 V V_{DD} = 2.7 to 4.5 V VDD = 1.8 to 2.7 V	1 1 1	– – –	16 8 3	
RC oscillator		V_{DD} = 4.75 to 5.25 V Tolerance: 10%	–	4	–	

Table 16-7. Oscillation Stabilization Time

(T_A = – 40°C to + 85°C, V_{DD} = 1.8 V to 5.5 V)

Oscillator	Test Condition	Min	Typ	Max	Unit
Main crystal	fosc > 1.0 MHz	–	–	20	ms
Main ceramic	Oscillation stabilization occurs when V_{DD} is equal to the minimum oscillator voltage range.	–	–	10	
External clock (main system)	X_{IN} input high and low width (t_{XH}, t_{XL})	25	–	500	ns
Oscillator stabilization wait time	t_{WAIT} when released by a reset (1)	–	2^{16}/fosc	–	ms
	t_{WAIT} when released by an interrupt (2)	–	–	–	

NOTES:

1. fosc is the oscillator frequency.
2. The duration of the oscillator stabilization wait time, t_{WAIT}, when it is released by an interrupt is determined by the setting in the basic timer control register, BTCON.

Table 16-8. Data Retention Supply Voltage in Stop Mode

(T_A = – 40°C to + 85°C, V_{DD} = 1.8 V to 5.5V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Data retention supply voltage	V_{DDDR}	Stop mode	1.8	–	5.5	V
Data retention supply current	I_{DDDR}	Stop mode; V_{DDDR} = 1.8 V	–	0.1	5	µA

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

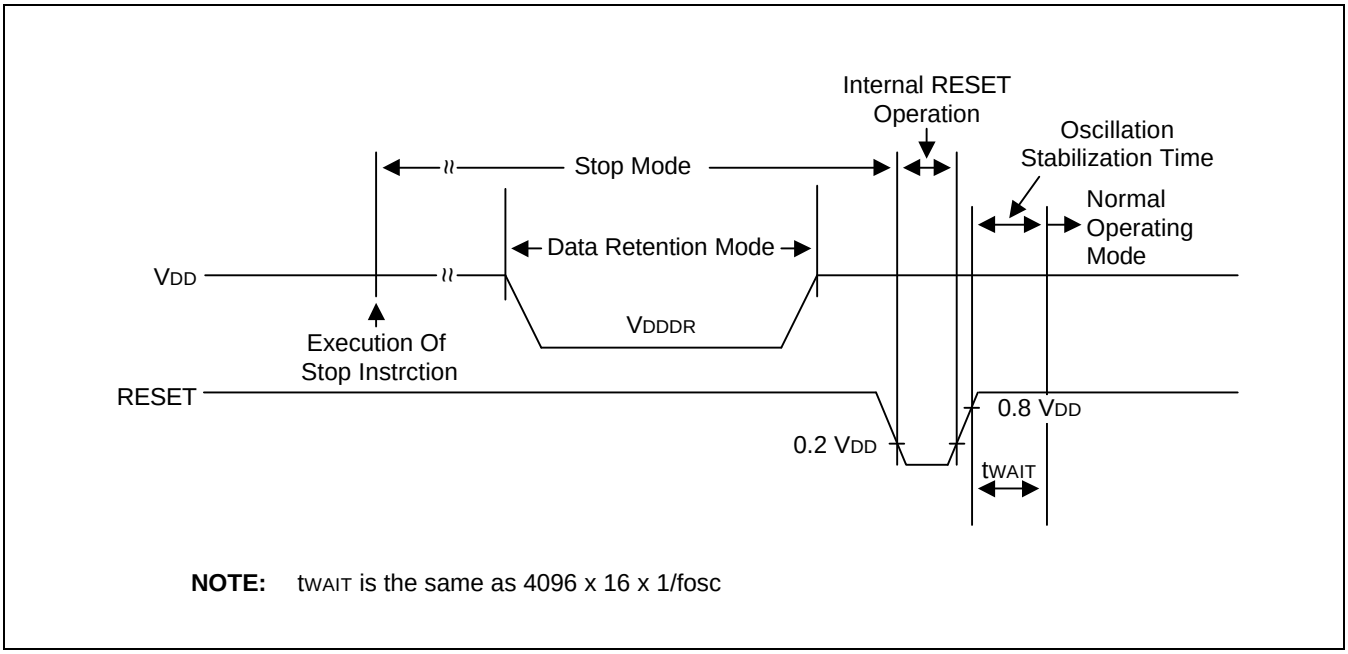


Figure 16-4. Stop Mode Release Timing When Initiated by a RESET

Table 16-9. Power-on RESET Circuit Characteristics

(T_A = – 40 °C to + 85 °C, V_{DD} = 3.0 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Power-on reset voltage high	V_{DDH}		3.0	–	5.5	V
Power-on reset voltage low	V_{DDL}		0	2.6	3.0	V
Power supply voltage rise time	t_r		10		(1)	us
Power supply voltage off time	t_{off}		0.5			s
Power-on reset circuit cunsumption current (2)	I_{DDPR}	V_{DD} = 5 V ± 10%		65	100	µA
		V_{DD} = 3.3 V		45	80	

NOTES:

1. 216/fx (= 6.55 ms at fx = 10 MHz)
2. Current consumed when power-on reset circuit is provided internally.

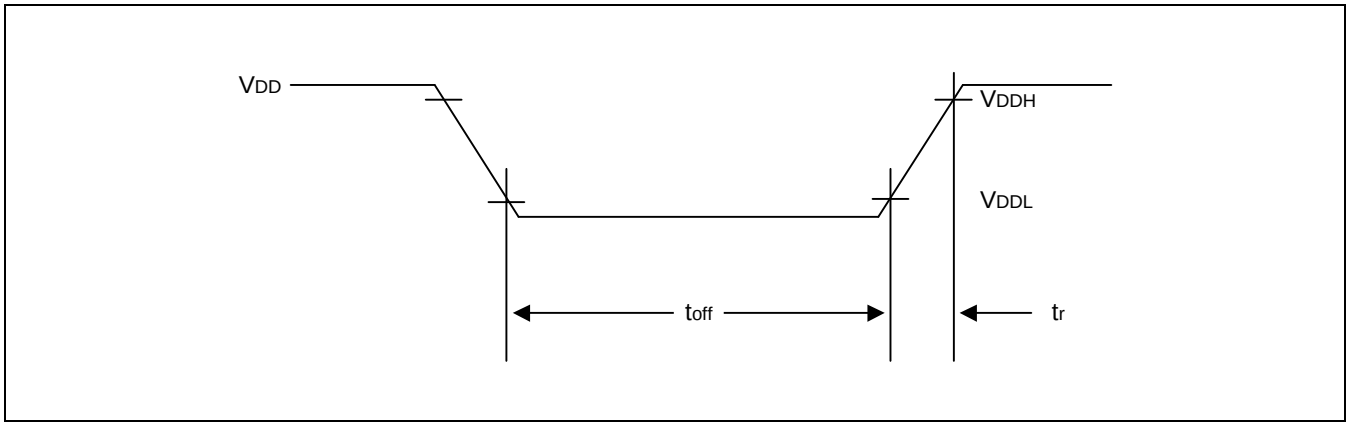


Figure16-5. Power-on RESET Timing

Table 16-10. A/D Converter Electrical Characteristics

(T_A = – 40°C to + 85°C, V_{DD} = 1.8/3.0 V to 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Total accuracy		V_{DD} = 5.12 V CPU clock = 10 MHz AV_{REF} = 5.12 V AV_{SS} = 0 V	–	–	± 3	LSB
Integral linearity error	ILE	"	–	–	± 2	LSB
Differential linearity error	DLE	"	–	–	± 1	
Offset error of top	EOT	"	–	±1	±3	
Offset error of bottom	EOB	"	–	±1	± 2	
Conversion time(1)	t_{CON}	fosc = 10 MHz	20	–	–	µs
Analog input voltage	V_{IAN}	–	AV_{SS}	–	AV_{REF}	V
Analog input impedance	R_{AN}	–	2	–	–	MΩ
ADC reference voltage	AV_{REF}	–	2.5	–	V_{DD}	V
ADC reference ground	AV_{SS}	–	V_{SS}	–	V_{SS} + 0.3	V
Analog input current	I_{ADIN}	AV_{REF} = V_{DD} = 5 V	–	–	10	µA
ADC block current (2)	I_{ADC}	AV_{REF} = V_{DD} = 5 V	–	1	3	mA
		AV_{REF} = V_{DD} = 3 V		0.5	1.5	
		AV_{REF} = V_{DD} = 5 V Power down mode	–	100	500	nA

NOTES:

1. 'Conversion time' is the time required from the moment a conversion operation starts until it ends.
2. I_{ADC} is operating current during A/D conversion.

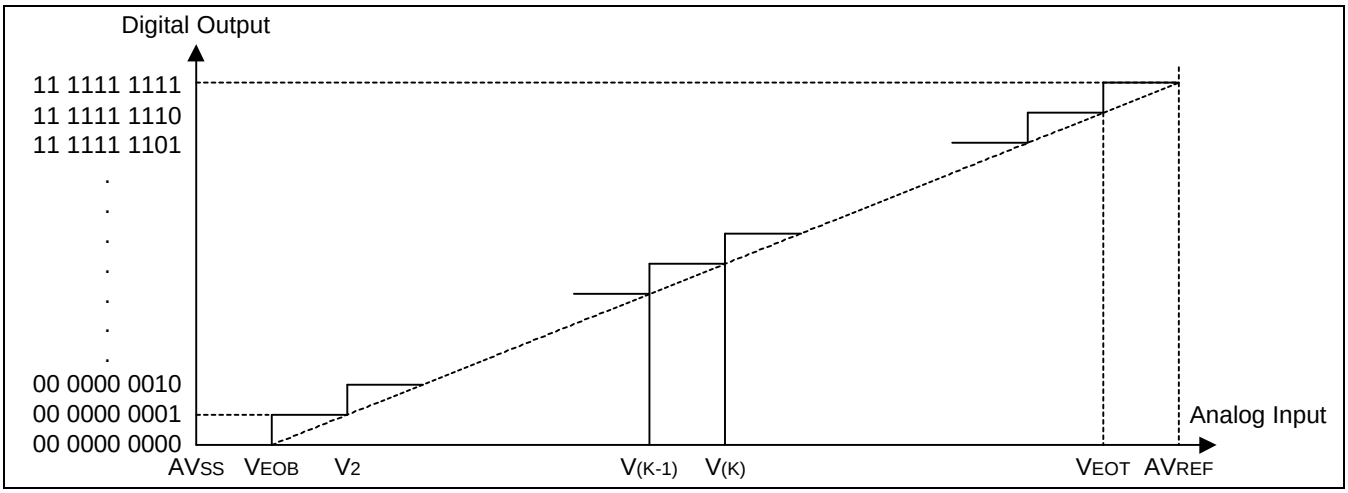


Figure 16-6. Definition of DLE and ILE

Table 16-11. Zero Crossing Detector

(T_A = – 40°C to + 85°C, V_{DD} = 4.5 V to 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Zero-crossing detection input voltage	V_{ZC}	AC connection c = 0.1 µF	1.0	–	3.0	Vp-p
Zero-crossing detection accuracy	V_{AZC}	f_{ZC} = 60 Hz (sine wave) V_{DD} = 5 V f_{OSC} = 10 MHz	–	–	± 150	mV
Zero-crossing detection input frequency	f_{ZC}	–	40	–	200	Hz

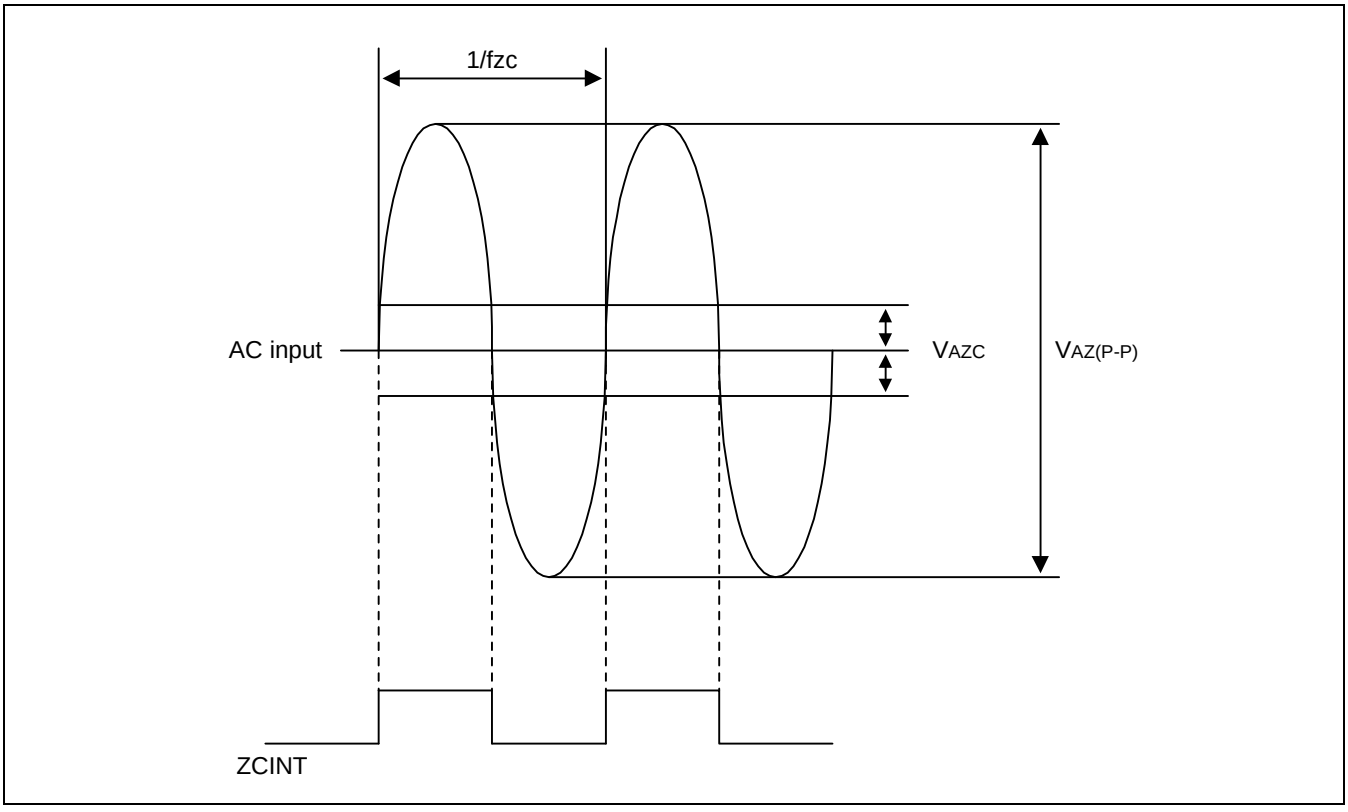
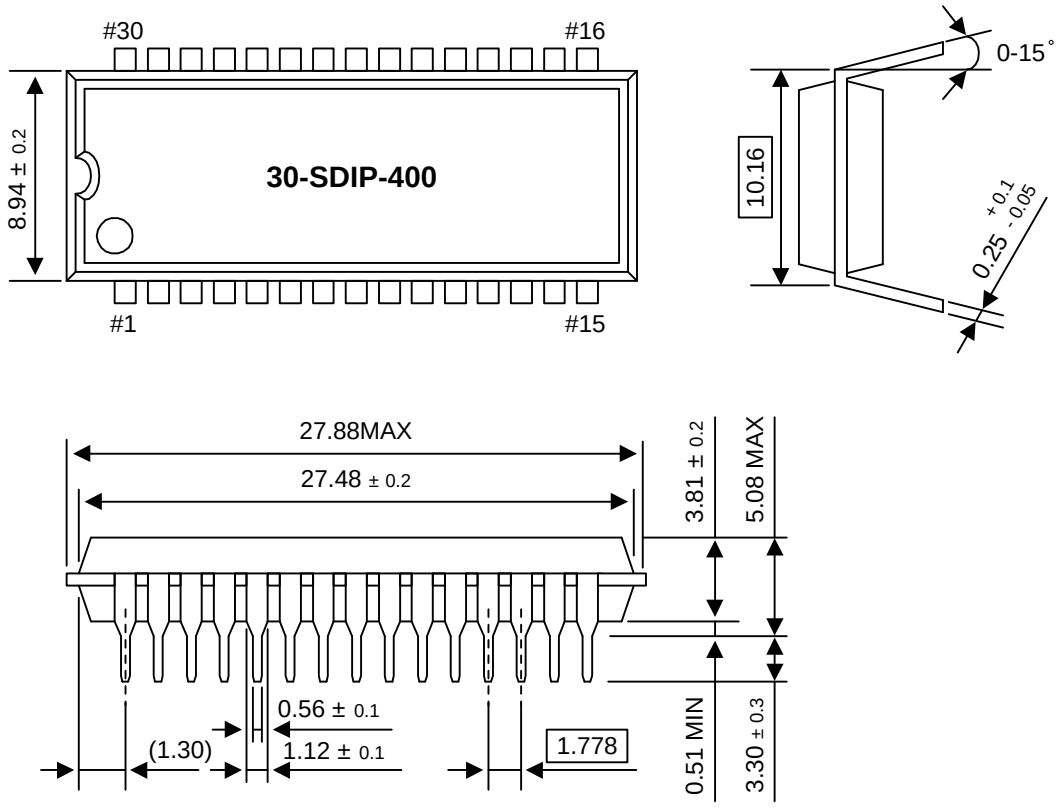


Figure 16-7. Zero Crossing Waveform Diagram

17 MECHANICAL DATA

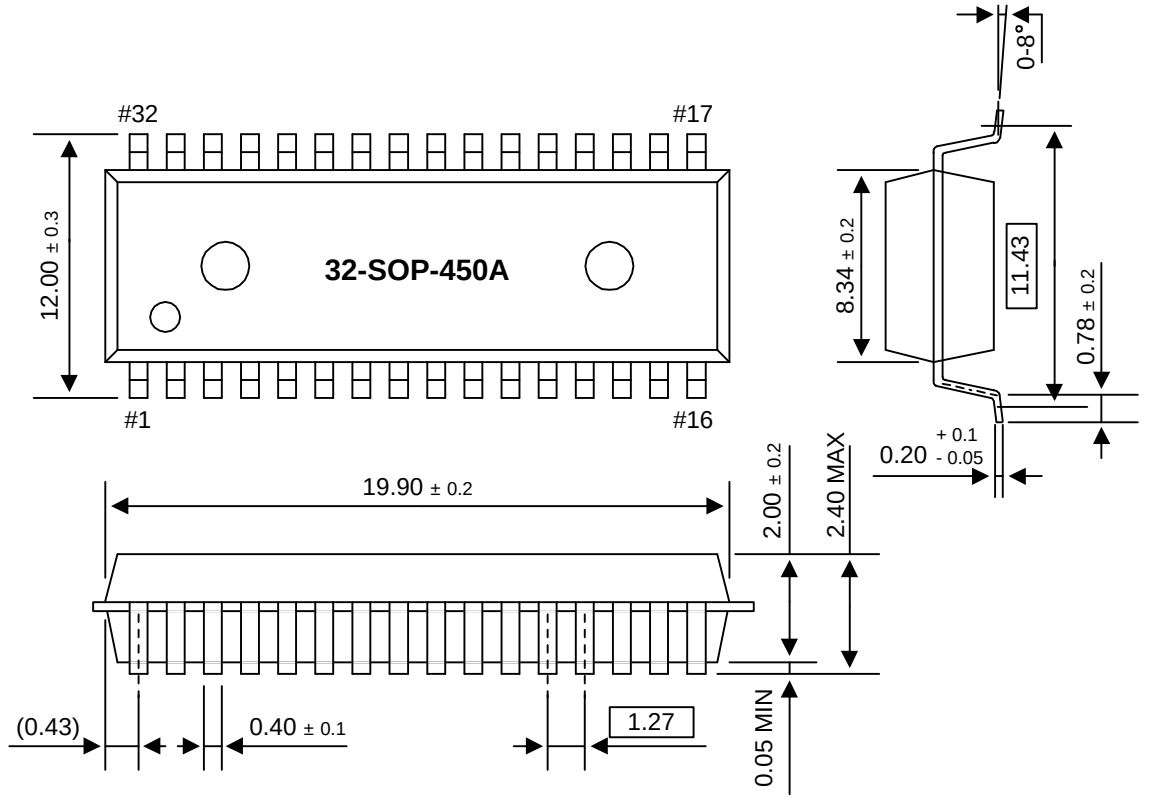
OVERVIEW

The S3C9424/C9428 is available in a 30-pin SDIP package (Samsung: 30-SDIP-400) and a 32-pin SOP package (32-SOP-450A) and a 28-pin SOP package (28-SOP-375). Package dimensions are shown in Figures 17-1, 17-2, and 17-3



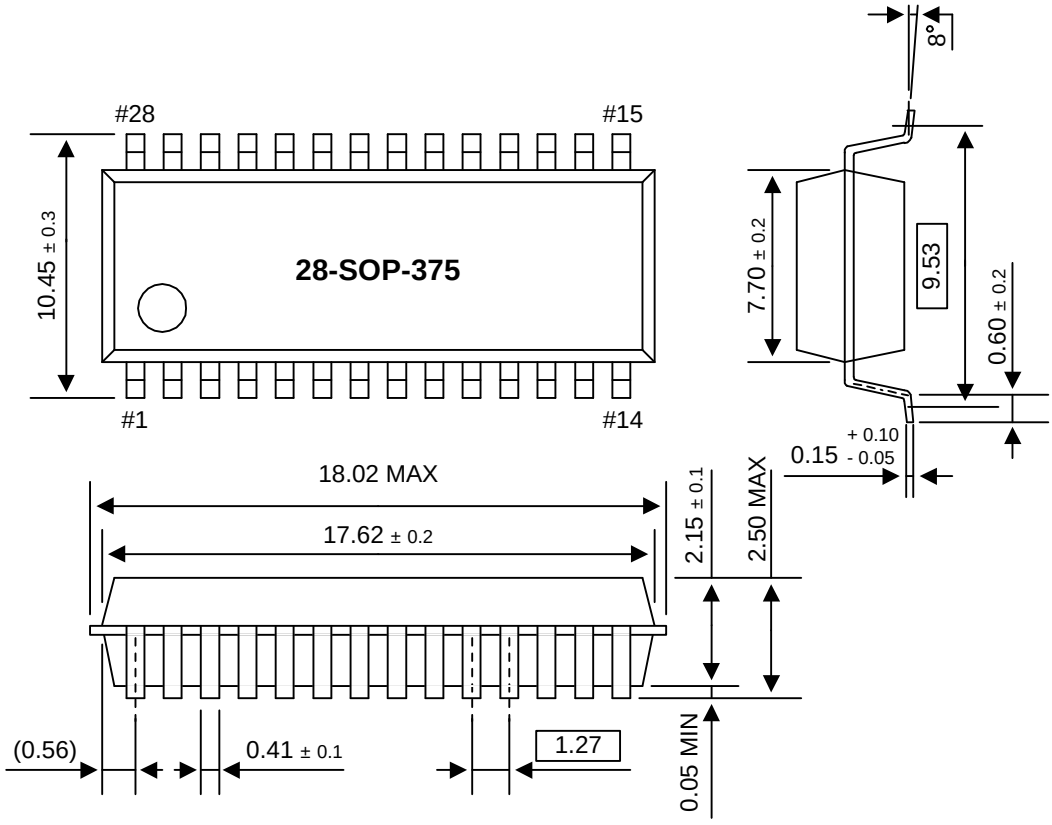
NOTE: Dimensions are in millimeters.

Figure 17-1. 30-Pin SDIP Package Dimensions



NOTE: Dimensions are in millimeters

Figure 17-2. 32-SOP-450A Package Dimensions



NOTE: Dimensions are in millimeters

Figure 17-3. 28-SOP-375 Package Dimensions

18 S3P9428 OTP

OVERVIEW

The S3P9428 single-chip CMOS microcontroller is the OTP (One Time Programmable) version of the S3C9424/C9428 microcontroller. It has an on-chip OTP ROM instead of masked ROM. The EPROM is accessed by serial data format.

The S3P9428 is fully compatible with the S3C9424/C9428, both in function and in pin configuration. Because of its simple programming requirements, the S3P9428 is ideal for use as an evaluation chip for the S3C9424/C9428.

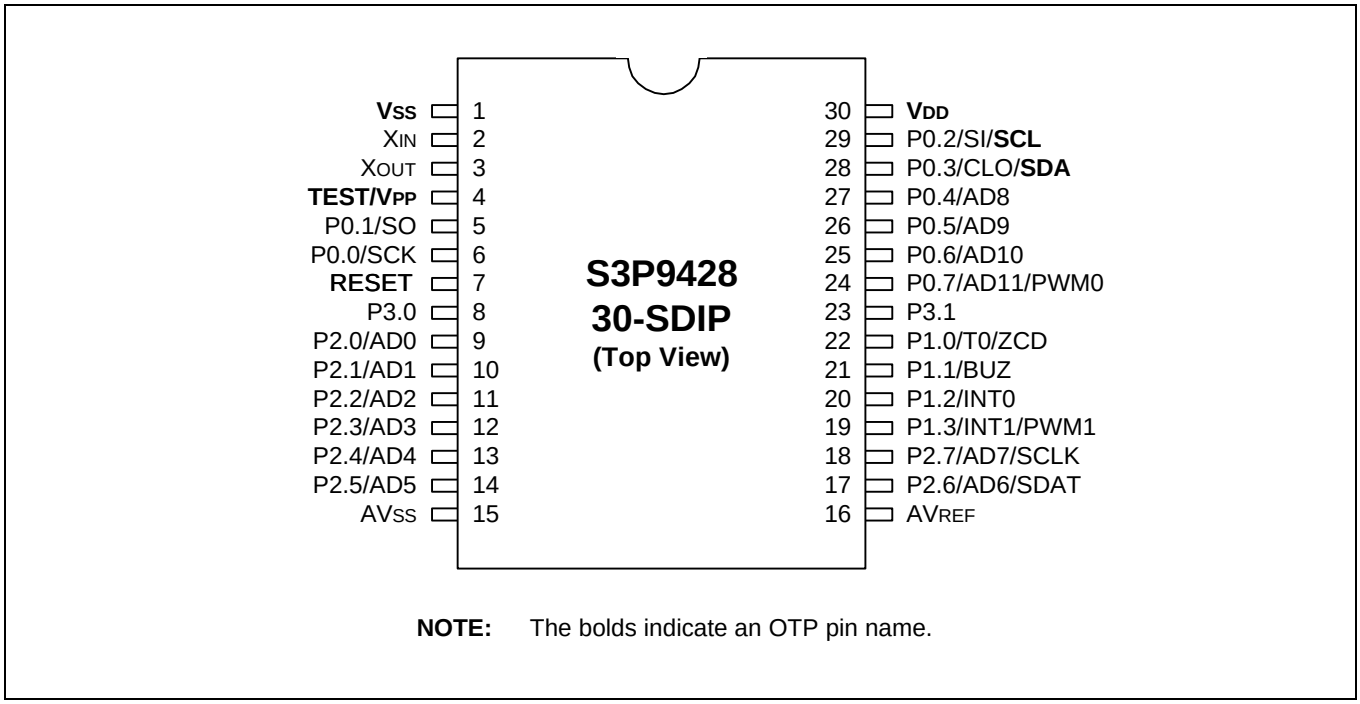
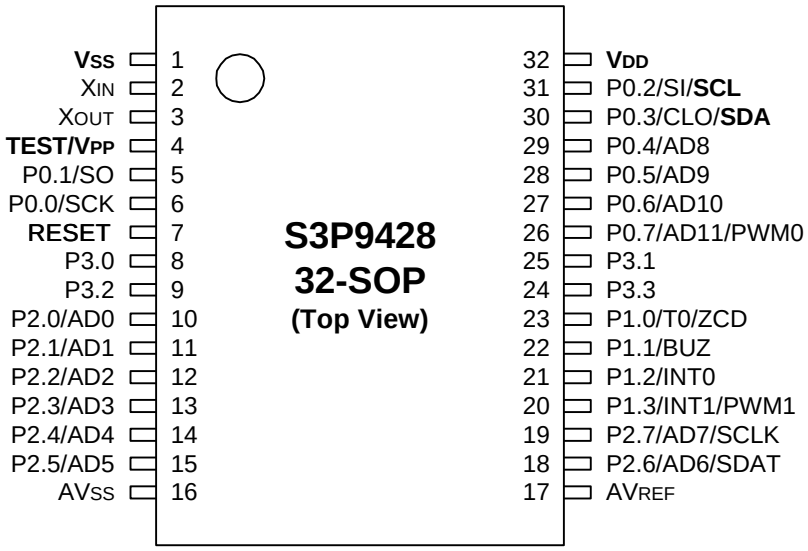
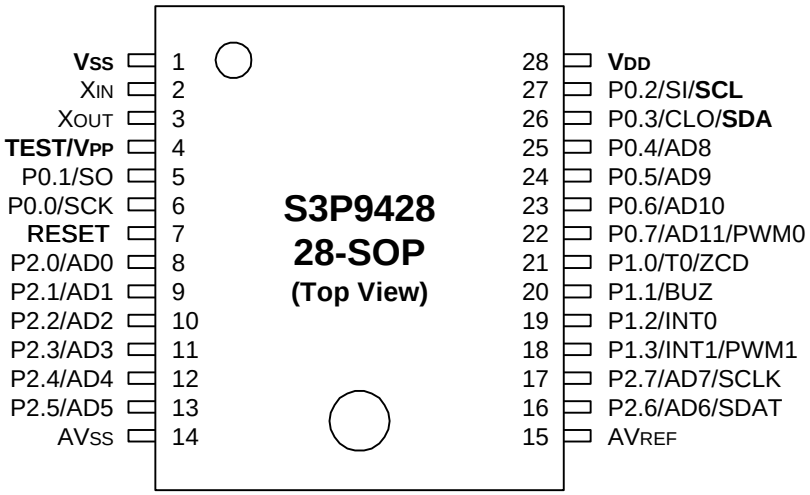


Figure 18-1. Pin Assignment Diagram (30-Pin SDIP Package)



NOTE: The bolds indicate an OTP pin name.

Figure 18-2. Pin Assignment Diagram (32-Pin SOP Package)



NOTE: The bolds indicate an OTP pin name.

Figure 18-3. Pin Assignment Diagram (28-Pin SOP Package)

Table 18-1. Descriptions of Pins Used to Read/Write the EPROM

Main Chip	During Programming			
Pin Name	Pin Name	Pin No.	I/O	Function
P0.3	SDAT	S3P9428 - 30 SDIP: 28 - 32 SOP: 30	I/O	Serial data pin (output when reading, Input when writing) Input and push-pull output port can be assigned
P0.2	SCLK	S3P9428 - 30 SDIP: 29 - 32 SOP: 31	I	Serial clock pin (input only pin)
TEST	V_{PP} (TEST)	4	I	Power supply pin for EPROM cell writing (indicates that OTP enters into the writing mode). When 12.5 V is applied, OTP is in writing mode and when 5 V is applied, OTP is in reading mode. (Option)
RESET	RESET	7	I	Chip Initialization
V_{DD}/V_{SS}	V_{DD}/V_{SS}	S3P9428 - 30 SDIP: 30/1 - 32 SOP: 32/1	I	Logic power supply pin.

Table 18-2. Comparison of S3P9428 and S3C9424/C9428 Features

Characteristic	S3P9428	S3C9424/C9428
Program Memory	8-Kbyte EPROM	4/8-Kbyte mask ROM
Operating Voltage (V_{DD})	3.0 V to 5.5 V (28 SOP: 1.8 V to 5.5)	3.0 V to 5.5 V (28 SOP: 1.8 V to 5.5)
OTP Programming Mode	V_{DD} = 5 V, V_{PP} (TEST) = 12.5 V	
Pin Configuration	30 SDIP/32 SOP/28SOP	
EPROM Programmability	User Program 1 time	Programmed at the factory

OPERATING MODE CHARACTERISTICS

When 12.5 V is supplied to the V_{PP} (TEST) pin of the S3P9428, 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 18-3 below.

Table 18-3. Operating Mode Selection Criteria

V_{DD}	V_{pp} (TEST)	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.

19 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) for S3C9424/C9428 microcontroller already have been developed.

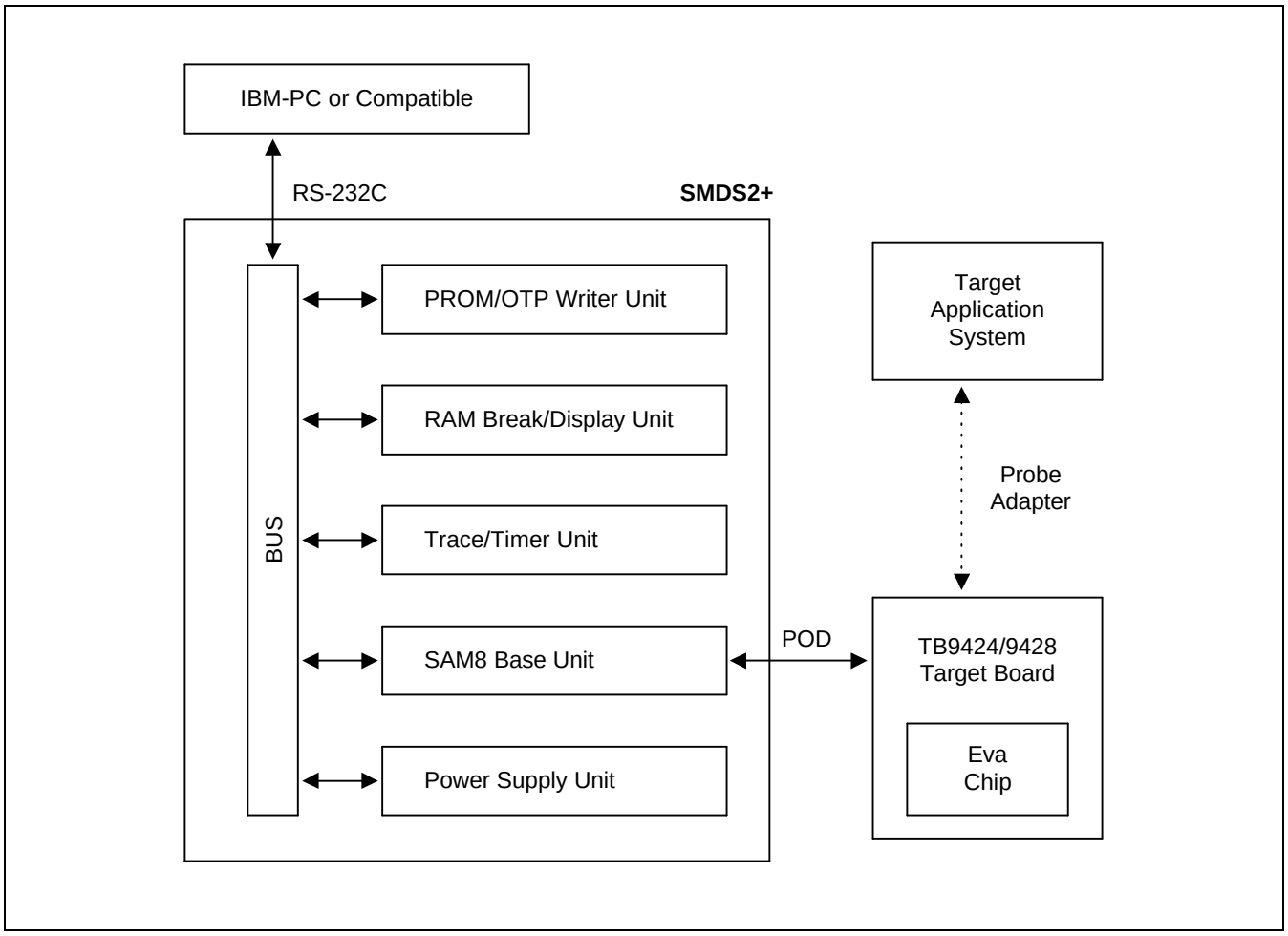


Figure 19-1. SMDS Product Configuration (SMDS2+)

TB9424/9428 TARGET BOARD

The TB9424/9428 target board is used for the S3P9428 microcontrollers. It is supported by the SMDS2+ development systems.

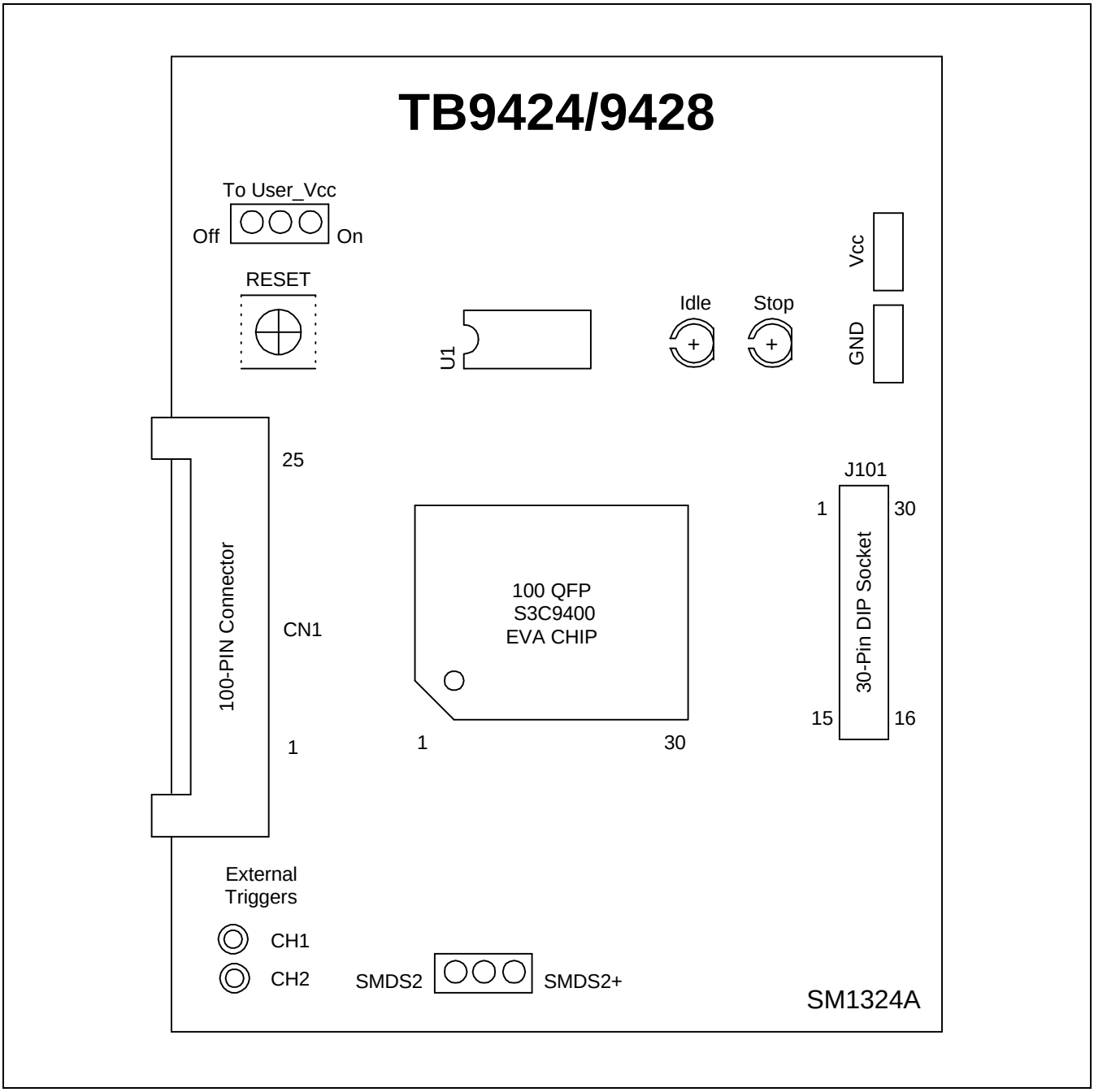
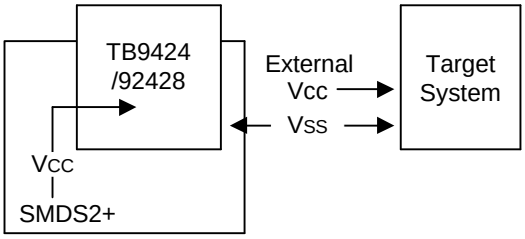
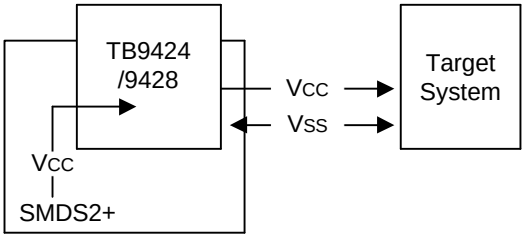


Figure 19-2. TB9424/9428 Target Board Configuration

Table 19-1. Power Selection Settings for TB864204A/08A

"To User_Vcc" Settings	Operating Mode	Comments
To User_Vcc Off  On		The SMDS2+ main board supplies V_{CC} to the target board (evaluation chip) and the target system.
To User_Vcc Off  On		The SMDS2+ main board 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_Vcc" 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 19-2. The SMDS2+ Tool Selection Setting

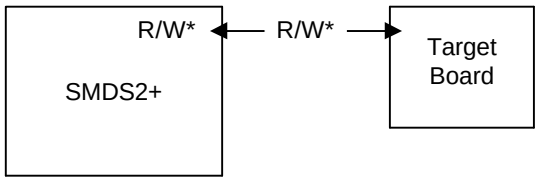
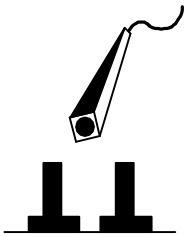
"SW1" Setting	Operating Mode
SMDS2  SMDS2+	

Table 19-3. 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.

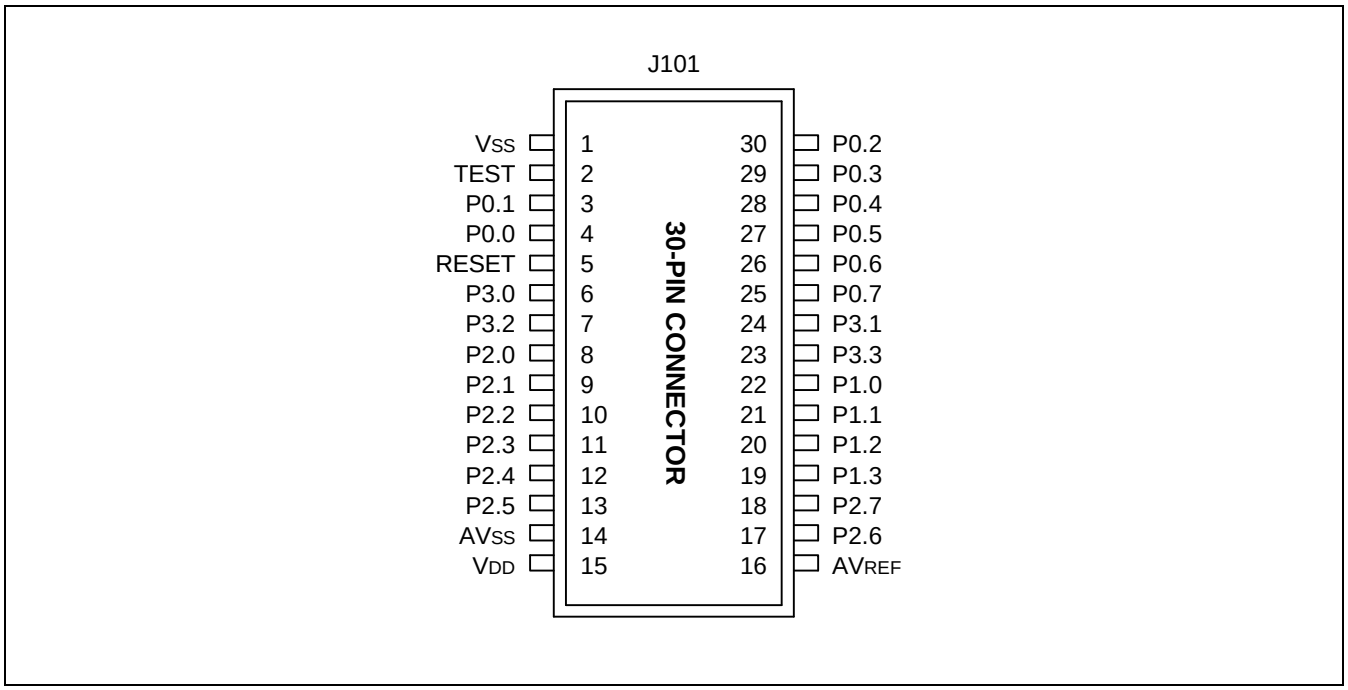


Figure 19-3. 30-Pin Connector for TB9424/9428

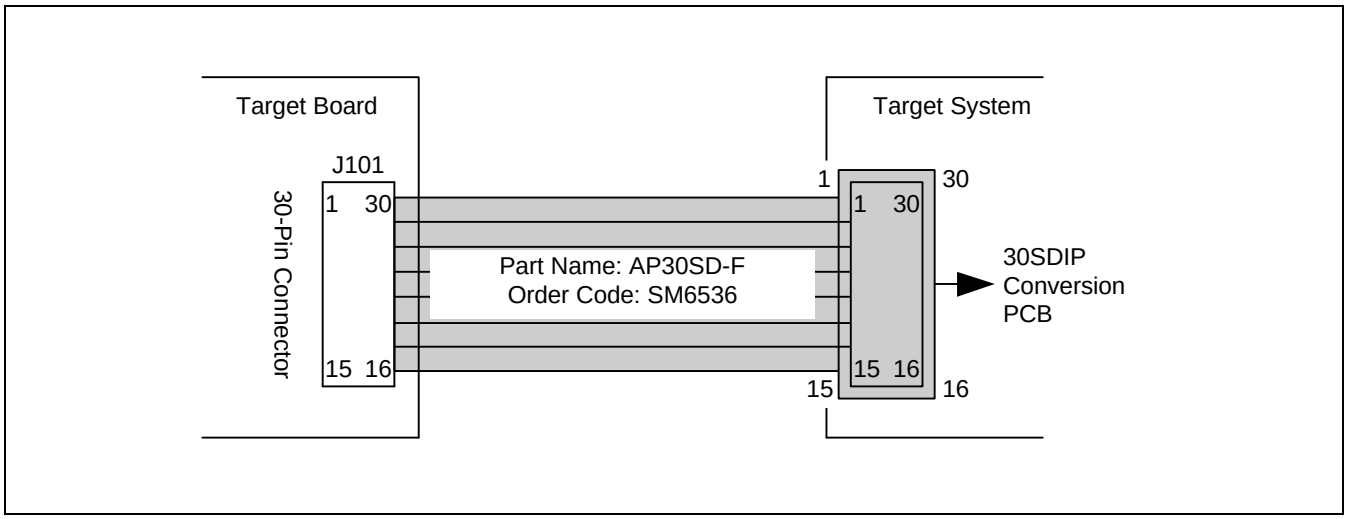


Figure 19-4. S3P9428 Probe Adapter for 30-SDIP Package