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.

S3C9432/C9434 MICROCONTROLLER

The S3C9432/C9434 single-chip 8-bit microcontroller is fabricated using an advanced CMOS process. It is built around the powerful SAM87Ri CPU core. The S3C9432/C9434 is a versatile microcontroller, with its A/D converter, timer, PWM, and SIO 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 S3C9432/C9434 have 2K-bytes or 4K-bytes of program memory on-chip (ROM) and 112-bytes of general purpose register area RAM.

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

— Three configurable I/O ports (13 pins)
— Five interrupt sources with one vector and one interrupt level
— One 8-bit timer/counter with time interval mode
— Analog to digital converter with five input channels and 10-bit resolution
— One synchronous SIO module
— One 12-bit PWM output

The S3C9432/C9434 microcontroller is ideal for use in a wide range of electronic applications requiring simple timer/counter, PWM, ADC, and SIO. S3C9432/C9434 is available in a 20/18/16-pin DIP and a 20-pin SOP package.

OTP

The S3P9434 is an OTP (One Time Programmable) version of the S3C9432/C9434 microcontroller. The S3P9434 has on-chip 4K-byte one-time-programmable EPROM instead of masked ROM. The S3P9434 is fully compatible with the S3C9432/C9434, in function, in D.C. electrical characteristics and in pin configuration.

FEATURES

CPU

- SAM87RI CPU core

Memory

- 2/4K-byte internal program memory (ROM)
- 112-byte general purpose register area (RAM)

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

- 600 ns at 10 MHz f_{OSC} (minimum cycles)
- 375 ns at 16 MHz f_{OSC} (minimum cycles)

Interrupts

- 5 interrupt sources with one vector and one level interrupt structure

General I/O

- Two I/O ports (Toatal 13 pins)
- One output only port (port 2)
- Bit programmable ports

Serial I/O

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

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 for the time interval mode

PWM Module

- 12-bit PWM 1-ch (Max: 250 kHz)
- 6-bit base + 6-bit extension frame

A/D Converter

- Five analog input pins
- 10-bit conversion resolution

Buzzer Frequency Range

- 200 Hz to 20 kHz signal can be generated

Oscillation Frequency

- 1 MHz to 16 MHz external crystal oscillator
- Maximum 16 MHz CPU clock
- 4 MHz RC oscillator

Operating Temperature Range

- - 40°C to + 85°C

Operating Voltage Range

- 3.0 V to 5.5 V

OTP Interface Protocol Spec

- Serial OTP

Package Types

- 20-pin DIP-300
- 20-pin SOP-375
- 18-pin DIP-300
- 16-pin DIP-300

BLOCK DIAGRAM

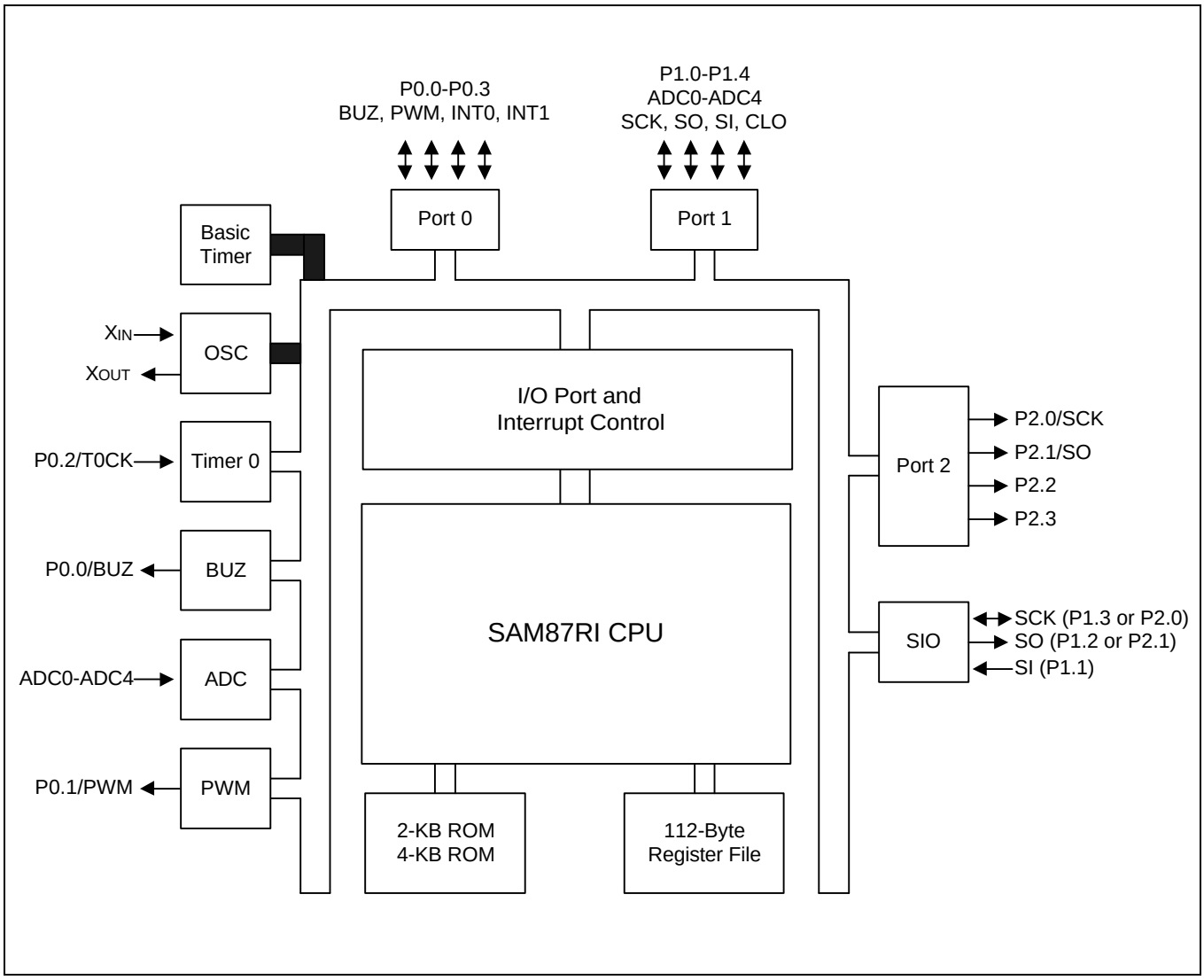


Figure 1-1. Block Diagram

PIN ASSIGNMENTS

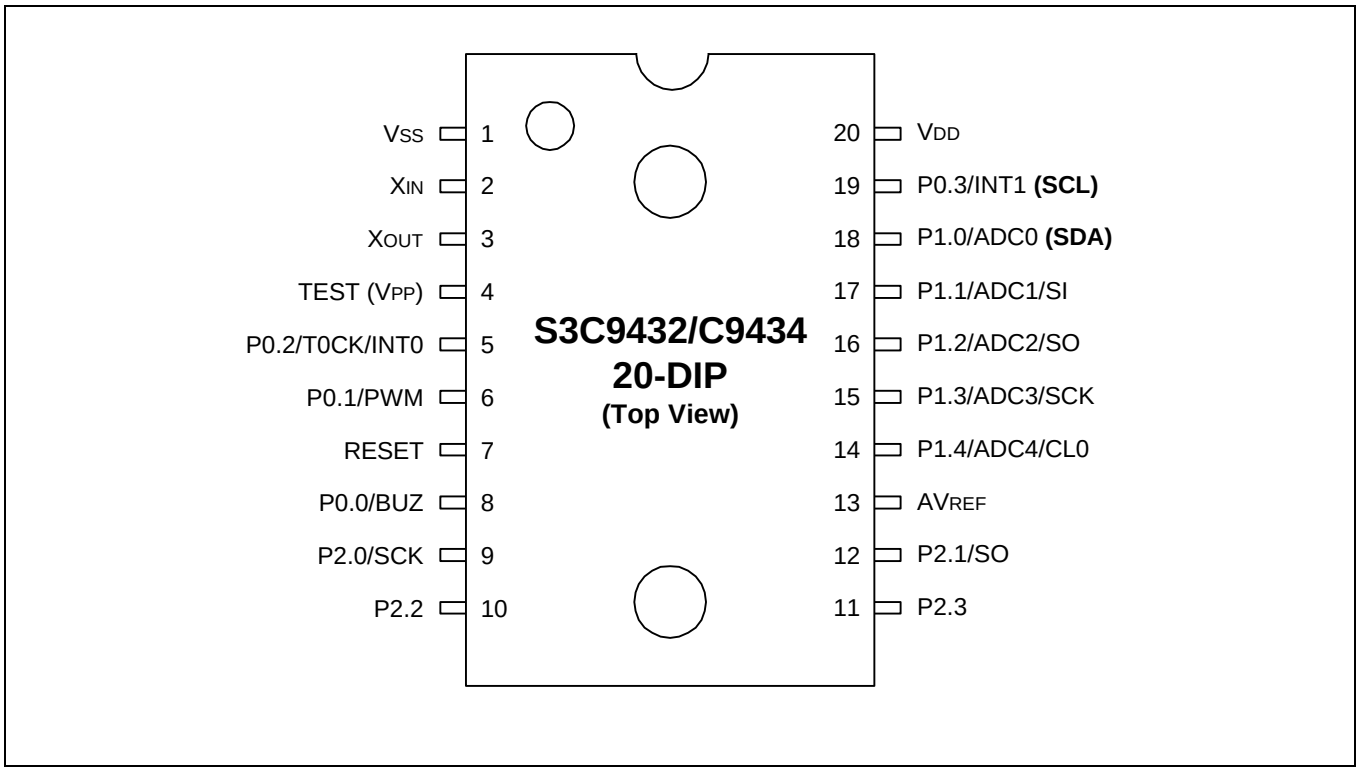


Figure 1-2. Pin Assignment Diagram (20-Pin DIP Package)

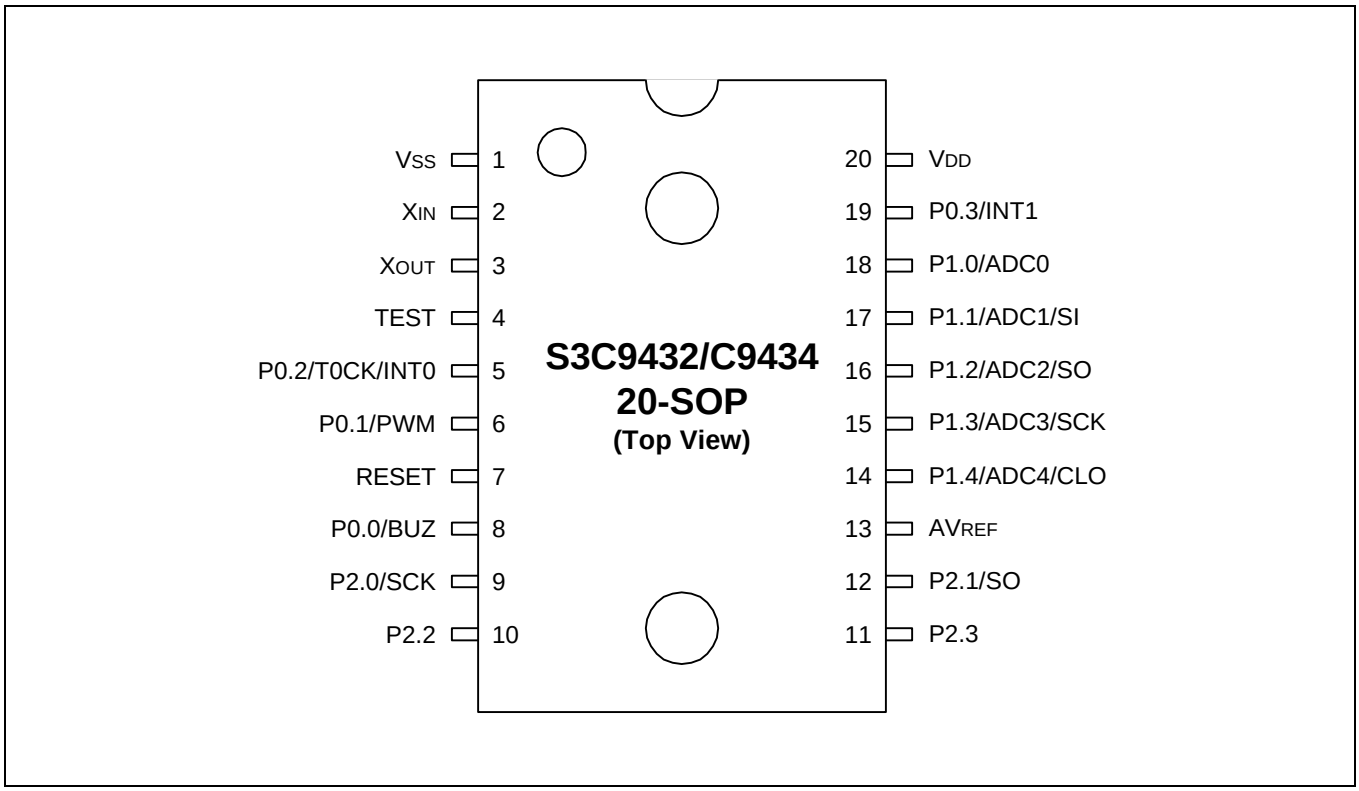


Figure 1-3. Pin Assignment Diagram (20-Pin SOP Package)

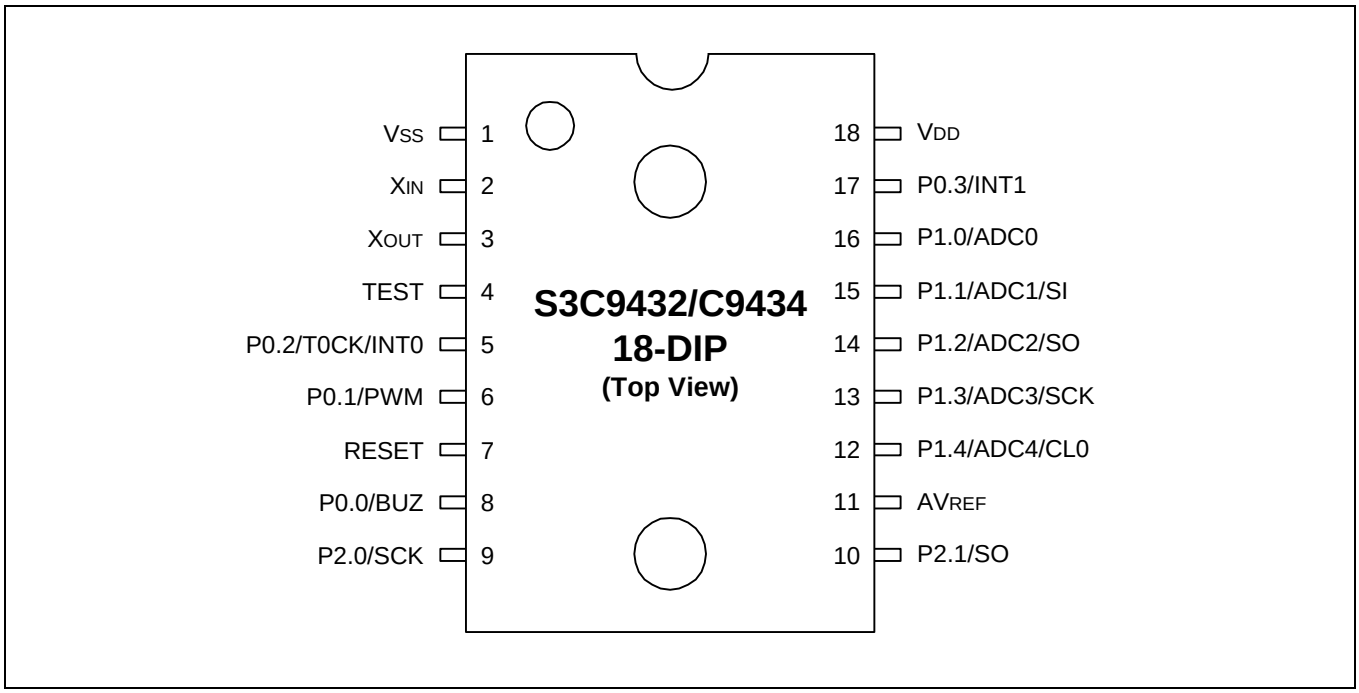


Figure 1-4. Pin Assignment Diagram (18-Pin DIP Package)

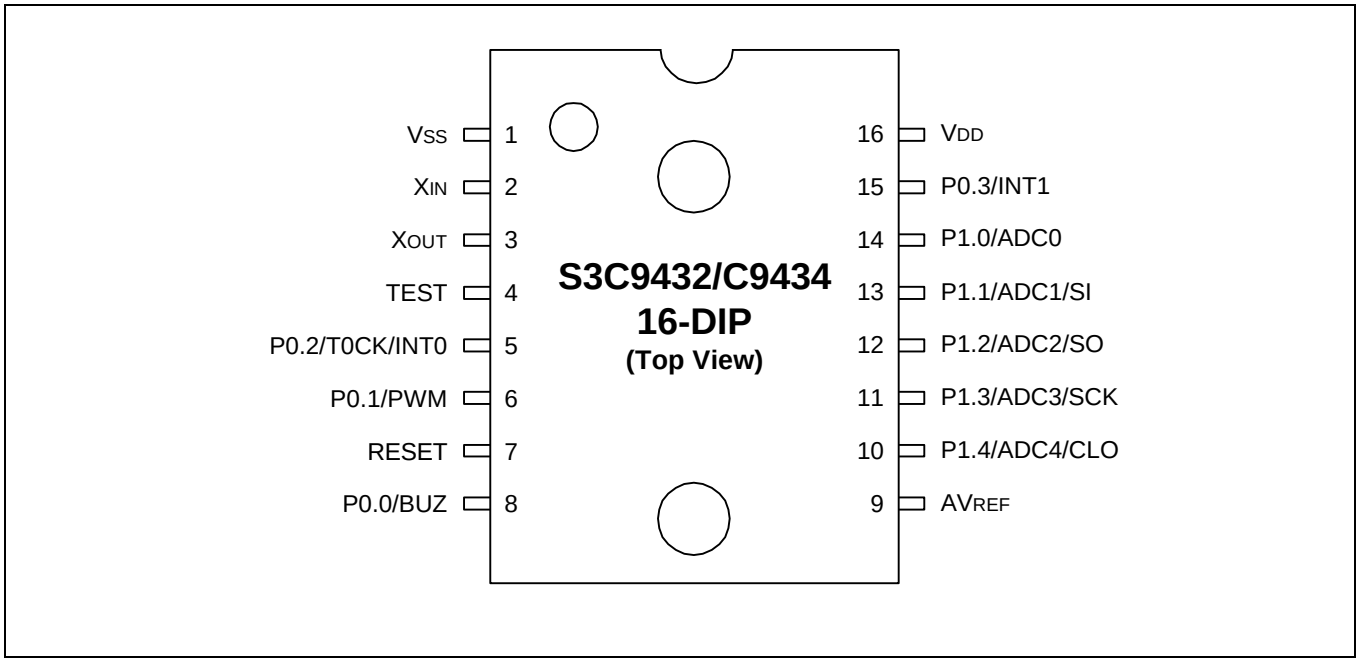


Figure 1-5. Pin Assignment Diagram (16-Pin DIP Package)

PIN DESCRIPTIONS

Table 1-1. S3C9432/C9434 Pin Descriptions

Pin Names	Pin Type	Pin Description	Circuit Type	Share Pins
P0.0-P0.3	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 0 pins can also be used as alternative function.	E	BUZ PWM INT0/T0CK INT1
P1.0-P1.4	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 1 pins can also be used as alternative function.	E-1	ADC0-ADC4 SI SO SCK CLO
P2.0-P2.3	O	Push-pull or open-drain output port. Pull up resistors are assignable by software. Port 2.0-2.1 pins can also be used as alternative function.	E-2	SCK SO
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})	–	–
V_{DD}, V_{SS}	–	Voltage input pin and ground	–	–
AV_{REF}	–	A/D converter reference voltage input and ground	–	–
AV_{SS}		Bonded to V_{SS} internally		
SCK	I/O	Serial interface clock I/O	E-1 E-2	P1.3 or P2.0
SO	O	Serial data output	E-1 E-2	P1.2 or P2.1
SI	I	Serial data input	E-1	P1.1
CLO	O	System clock output port	E-1	P1.4
BUZ	O	200 Hz- 20 kHz frequency output for buzzer sound	E	P0.0
PWM	O	12-bit PWM output	E	P0.1
INT0-INT1	I	External interrupt input port	E	P0.2 P0.3
T0CK	I	Timer 0 external clock input	E	P0.2
ADC0-ADC4	I	A/D converter input	E-1	P1.0-P1.4

PIN CIRCUITS

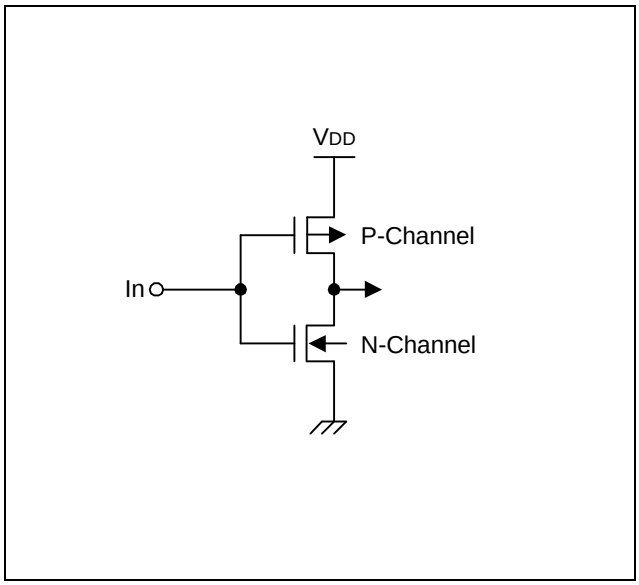


Figure 1-6. Pin Circuit Type A

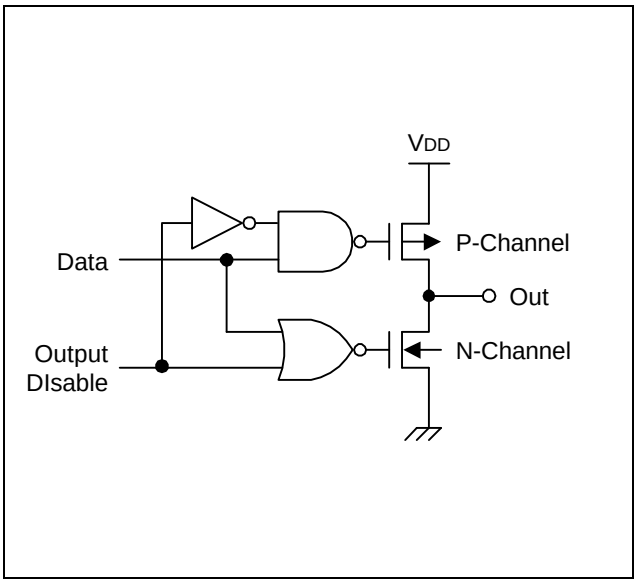


Figure 1-8. Pin Circuit Type C

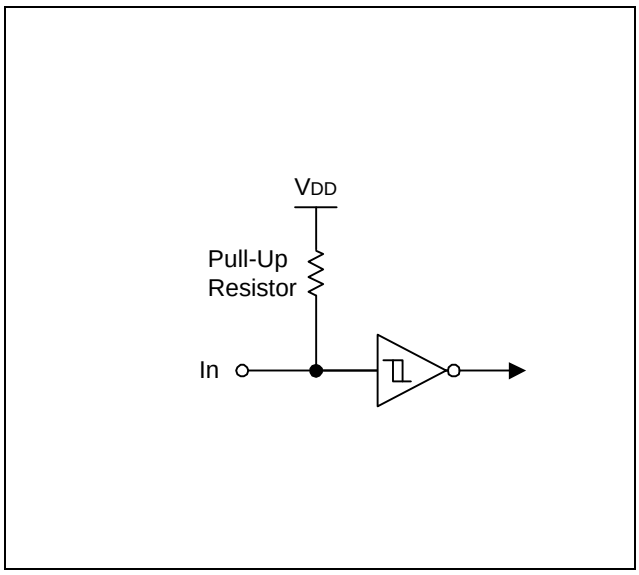


Figure 1-7. Pin Circuit Type B

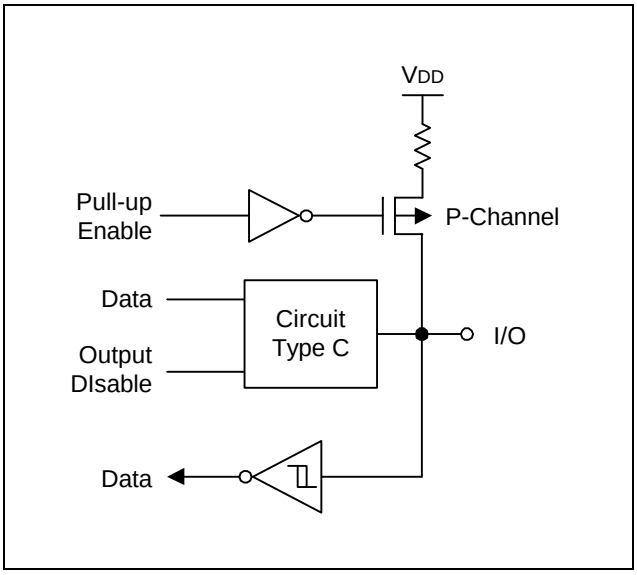


Figure 1-9. Pin Circuit Type D

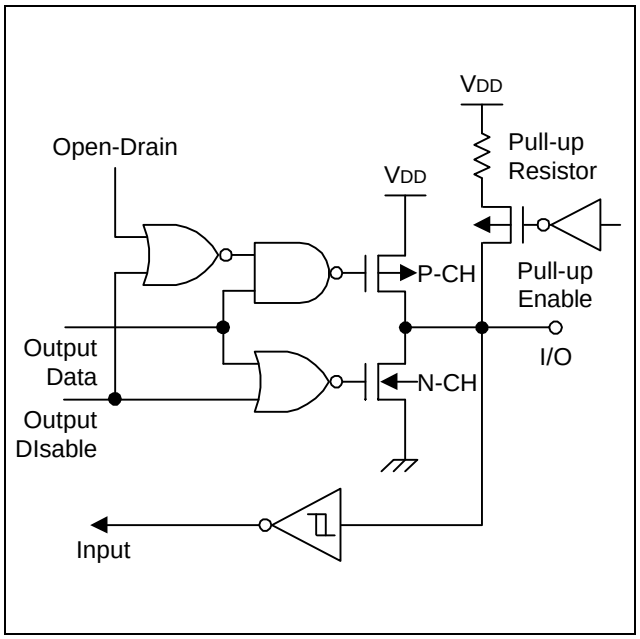


Figure 1-10. Pin Circuit Type E

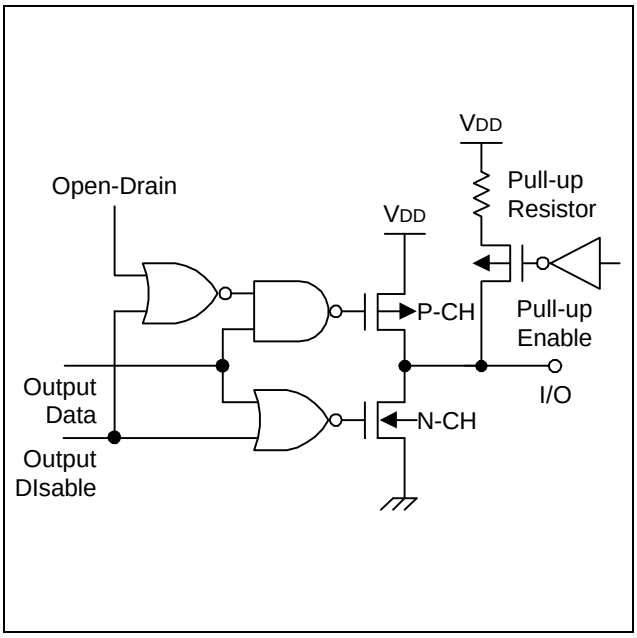


Figure 1-12. Pin Circuit Type E-2

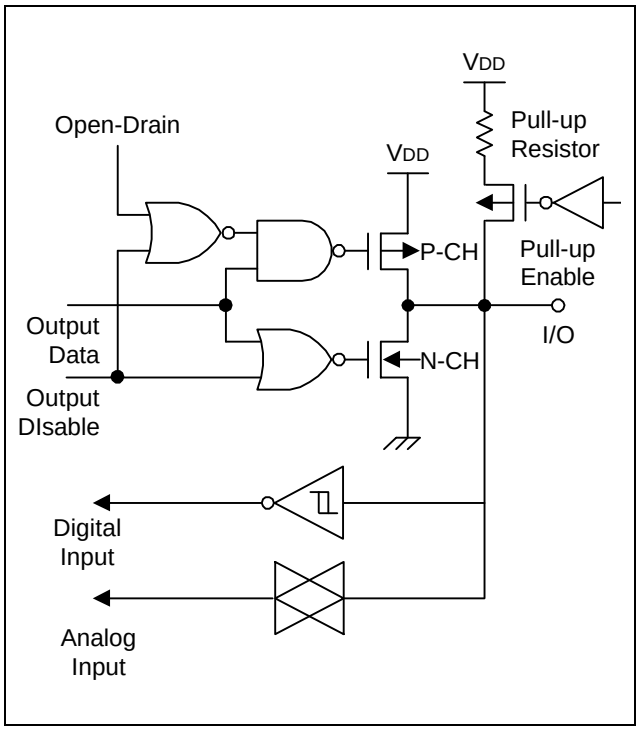


Figure 1-11. Pin Circuit Type E-1

2 ADDRESS SPACES

OVERVIEW

The S3C9432/C9434 microcontroller has two kinds of address space:

— Internal program memory (ROM)
— Internal register file

A 12-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 S3C9432/C9434 have 2K-bytes or 4K-bytes of mask-programmable on-chip program memory: which is configured as the Internal ROM mode, all of the 4-Kbyte internal program memory is used.

The S3C9432/C9434 microcontroller has 112 general-purpose registers in its internal register file. Thirty-one bytes in the register file are mapped for system and peripheral control functions.

PROGRAM MEMORY (ROM)

Normal Operating Mode

The S3C9432/C9434 have 2K-bytes (locations 0H-07FFH) or 4K-bytes (locations 0H-0FFFH) 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.

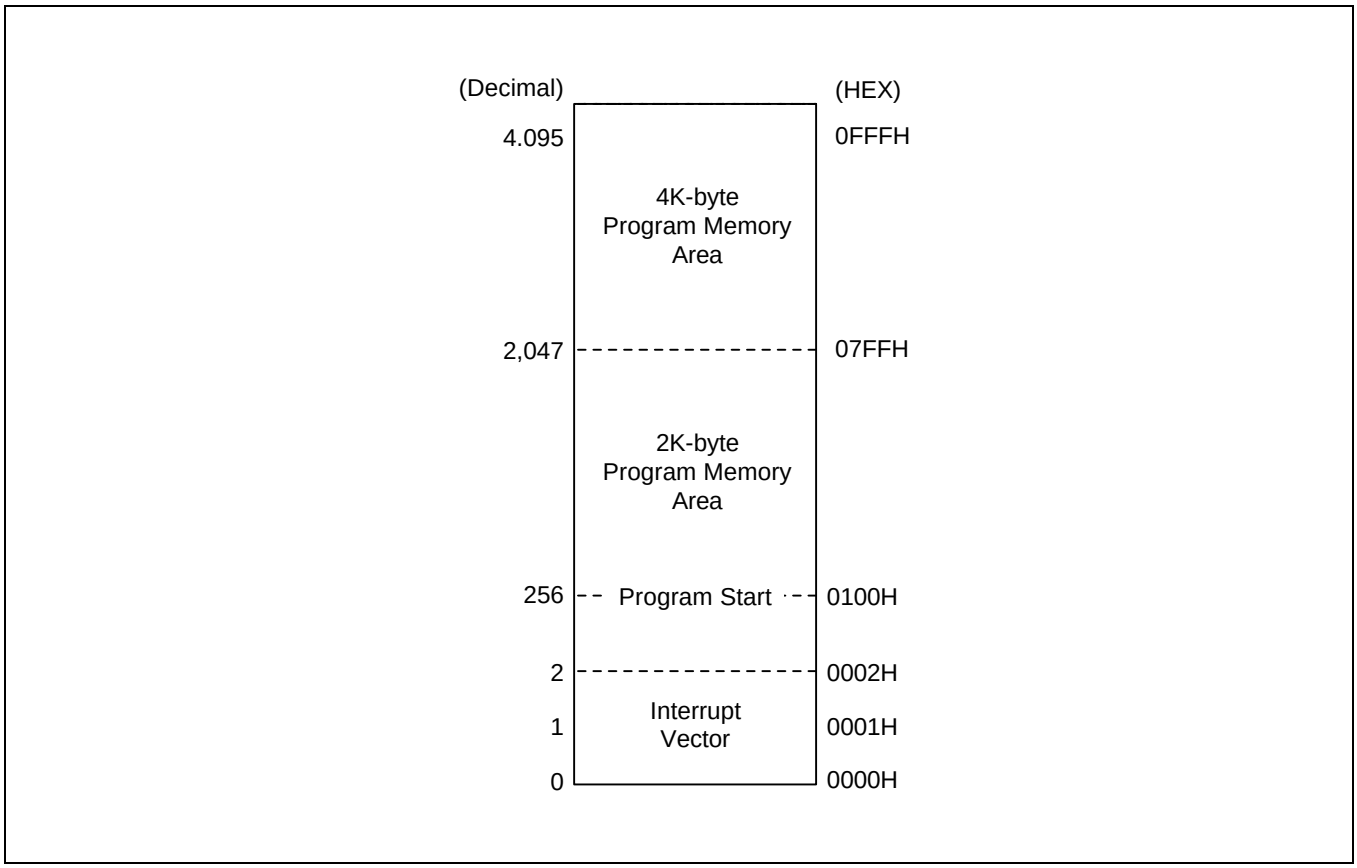


Figure 2-1. Program Memory Address Space

REGISTER ARCHITECTURE

The upper 64-bytes of the S3C9432/C9434's internal register file are addressed as working registers, system control registers and peripheral control registers. The lower 96-bytes of internal register file(00H-5FH) is called the *general purpose register space*. The total addressable register space is thereby 160-bytes. 143 registers in this space can be accessed; 112 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: page0). This register file expansion is not implemented in the S3C9432/C9434 , 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	11
Peripheral, I/O, and clock control and data registers	20
General-purpose registers (including the 16-bit common working register area)	112
Total Addressable Bytes	143

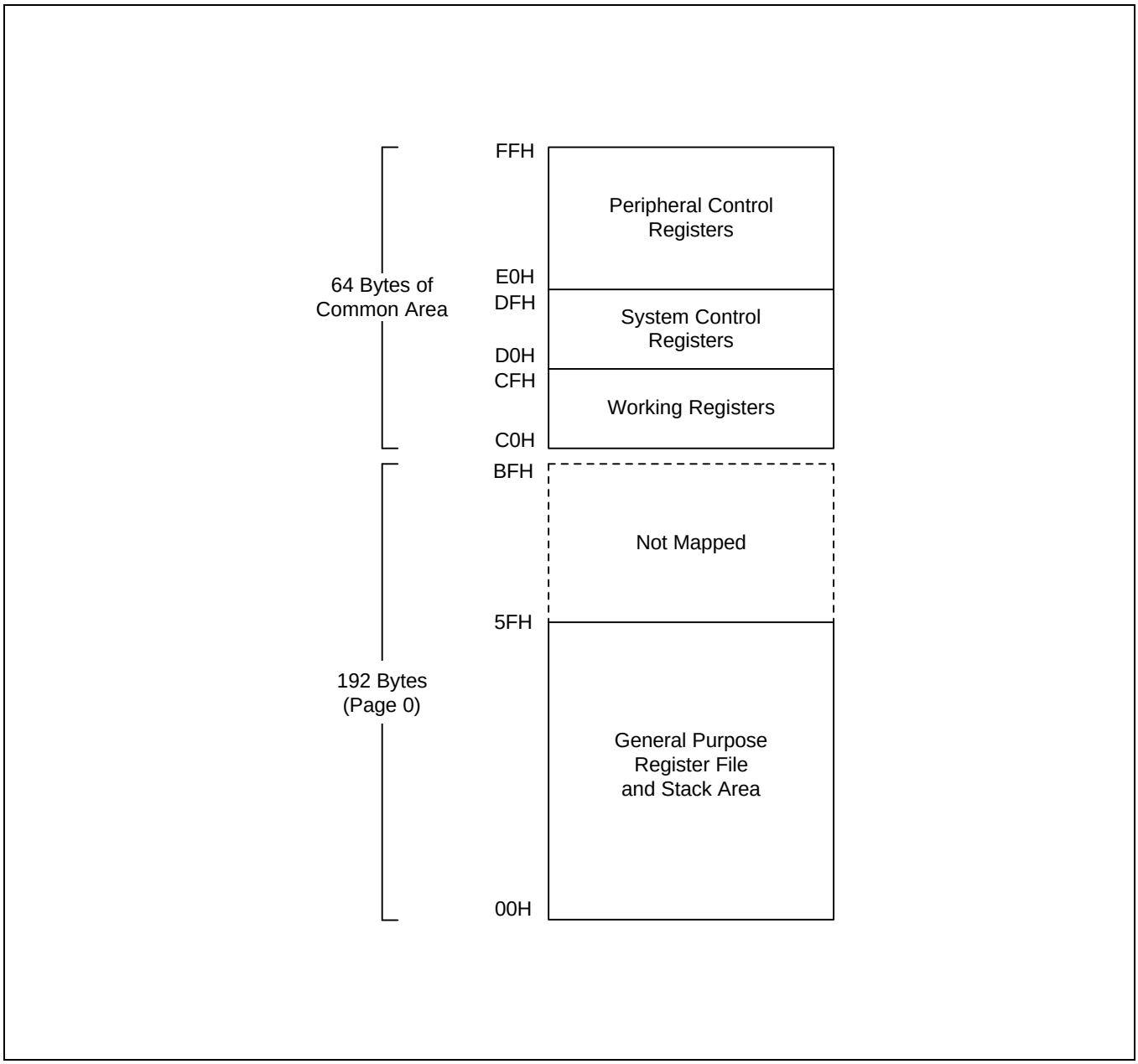


Figure 2-2. Internal Register File Organization

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.

This16-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 S3C9432/C9434 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.

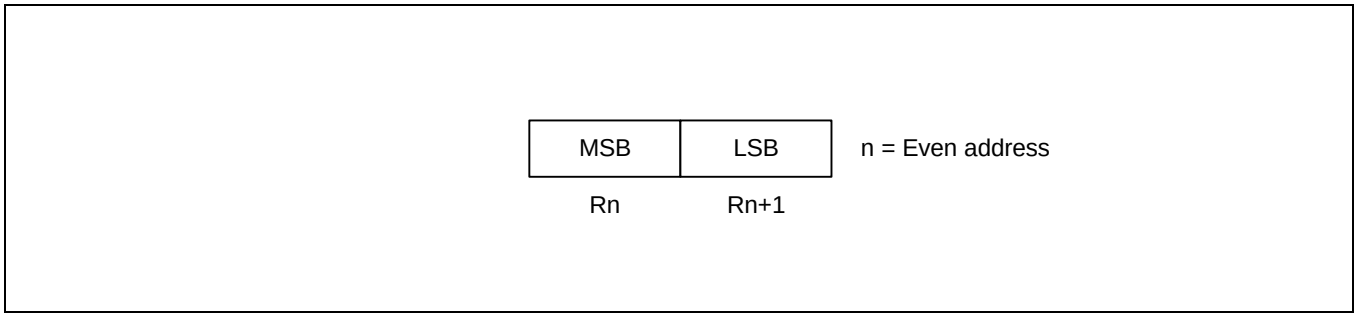


Figure 2-3. 16-Bit Register Pairs

☞ 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 S3C9432/C9434 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.

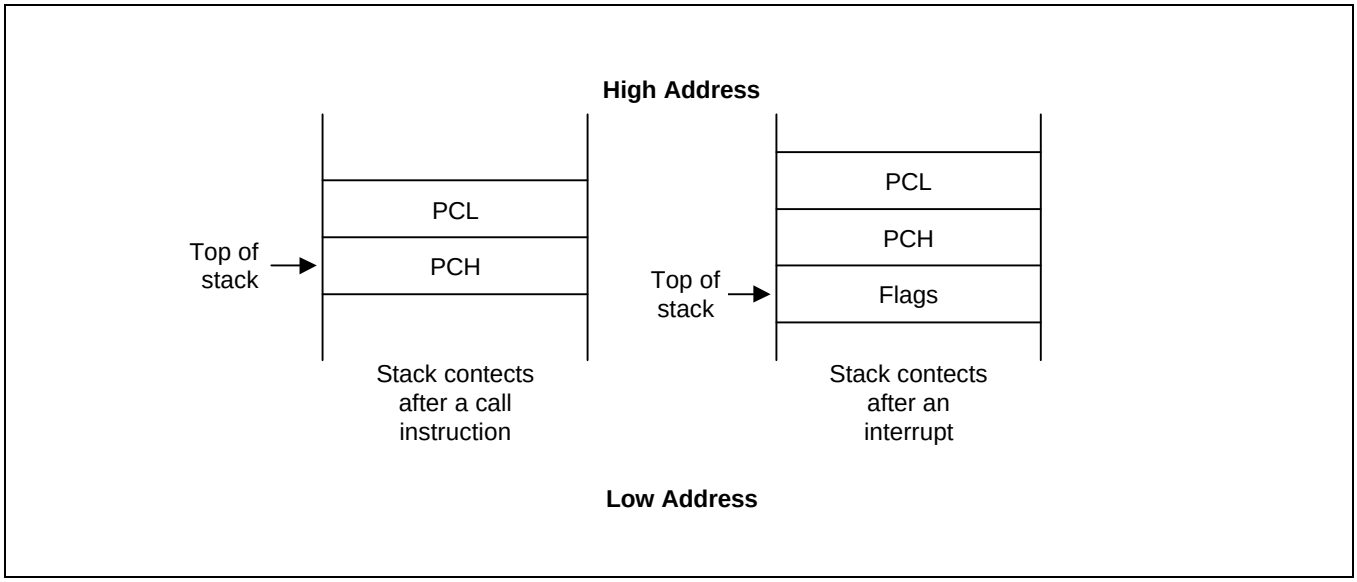


Figure 2-4. Stack Operations

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 S3C9432/C9434 , the SP must be initialized to an 8-bit value in the range 00H–060H.

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,#60H      ;  SP ← 60H (Normally, the SP is set to 60H 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

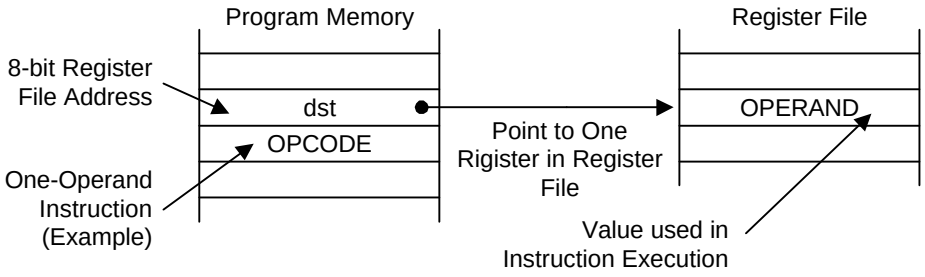
Instructions that are stored in program memory are fetched for execution using the program counter. Instructions indicate the operation to be performed and the data to be operated on. *Addressing mode* is the method used to determine the location of the data operand. The operands specified in SAM87Ri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

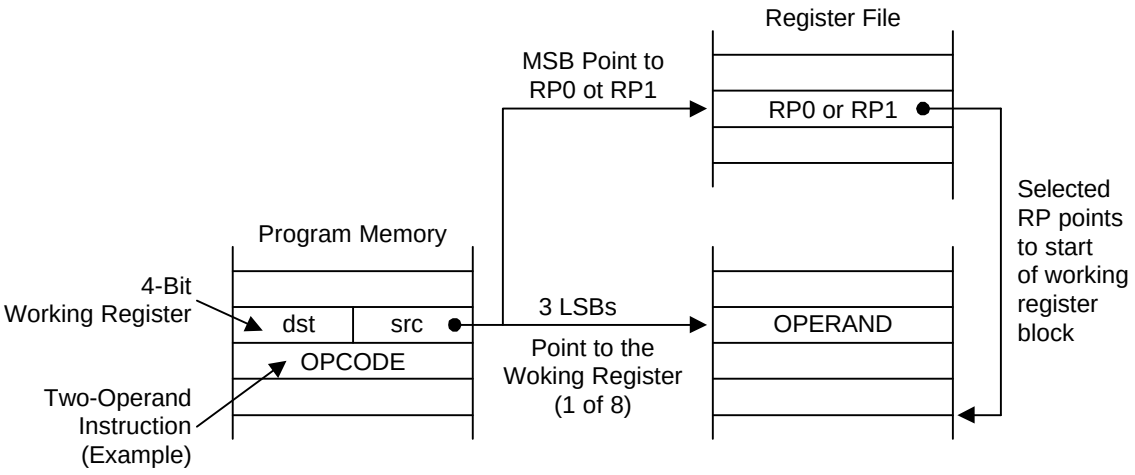
In Register addressing mode, the operand is the content of a specified register (see Figure 3-1). Working register addressing differs from Register addressing because it uses an 16-byte working register space in the register file and an 4-bit register within that space (see Figure 3-2).



Sample Instruction:

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

Figure 3-1. Register Addressing



Sample Instruction:

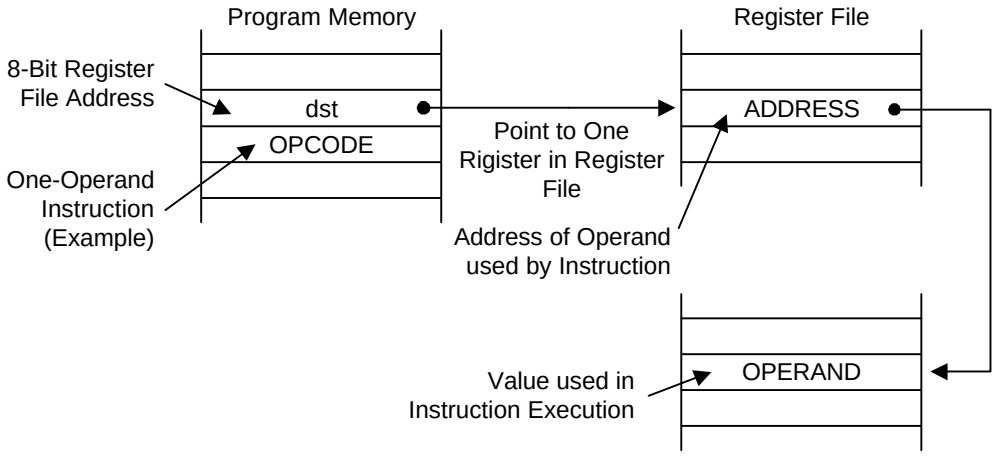
```
ADD     R1, R2          ;   Where R1 and R2 are registers in the curruntly
                            selected working register area.
```

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.



Sample Instruction:

```
RL      @SHIFT          ;   Where SHIFT is the label of an 8-Bit register address
```

Figure 3-3. Indirect Register Addressing to Register File

INDIRECT REGISTER ADDRESSING MODE (Continued)

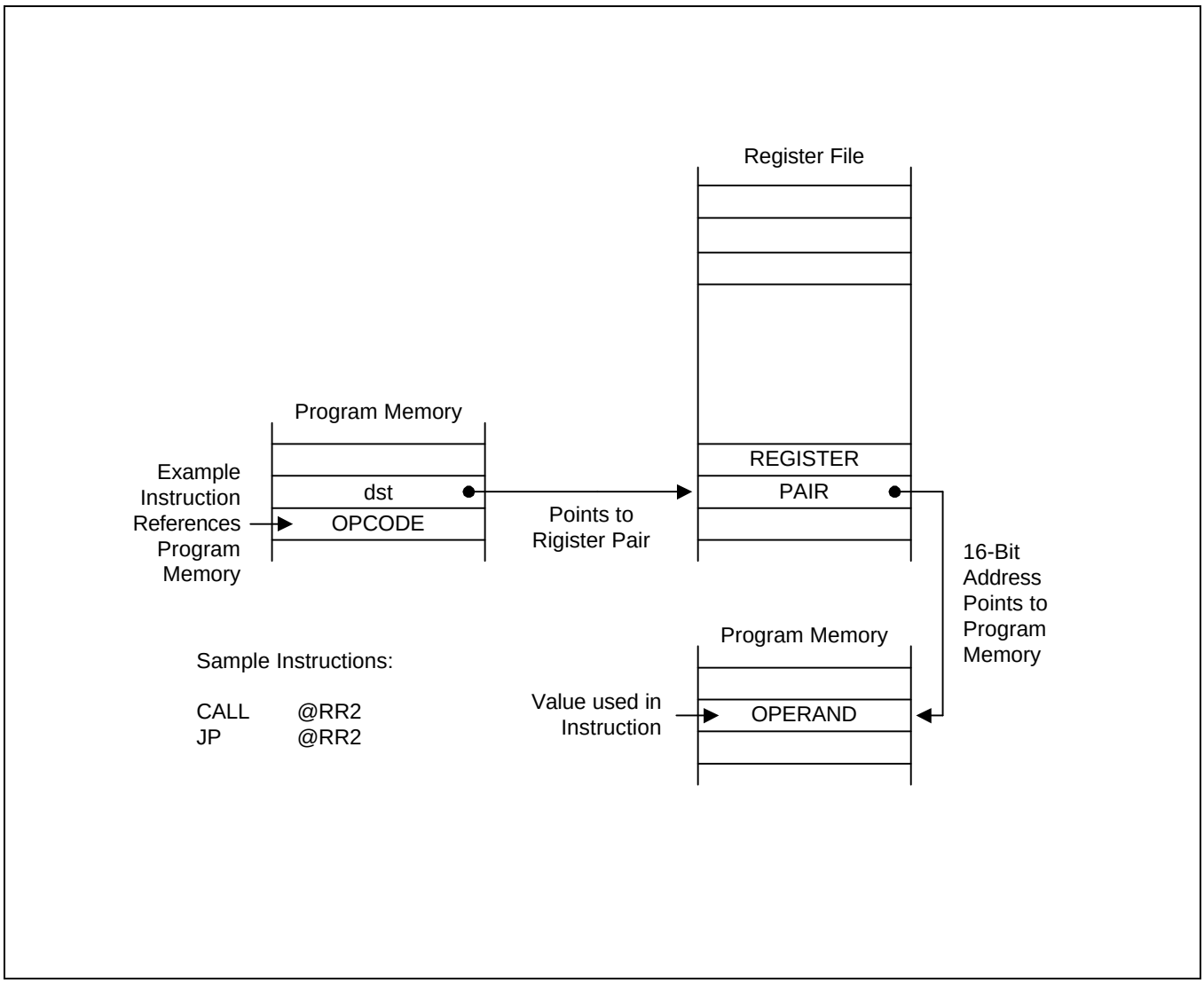


Figure 3-4. Indirect Register Addressing to Program Memory

INDIRECT REGISTER ADDRESSING MODE (Continued)

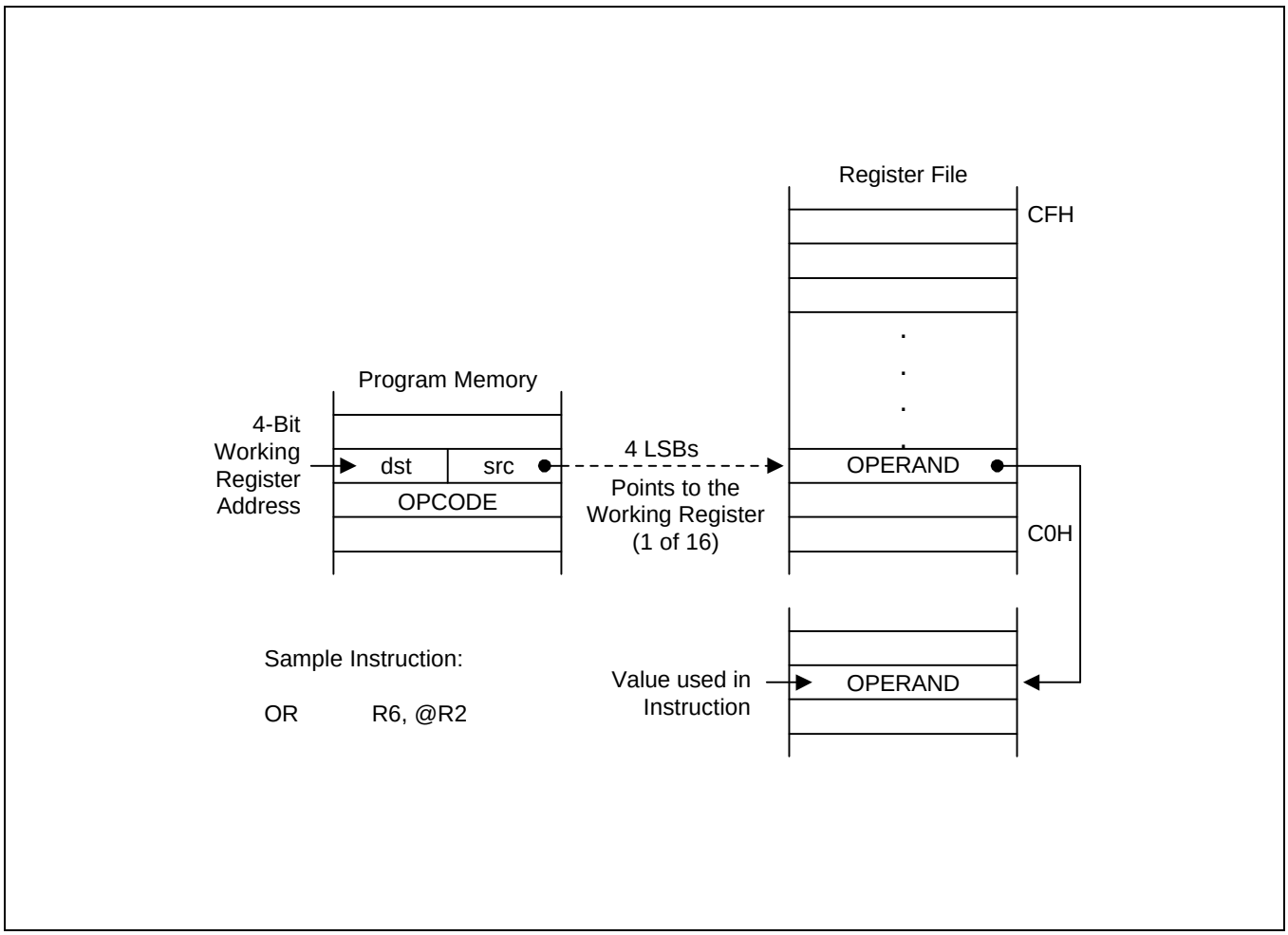


Figure 3-5. Indirect Working Register Addressing to Register File

INDIRECT REGISTER ADDRESSING MODE (Concluded)

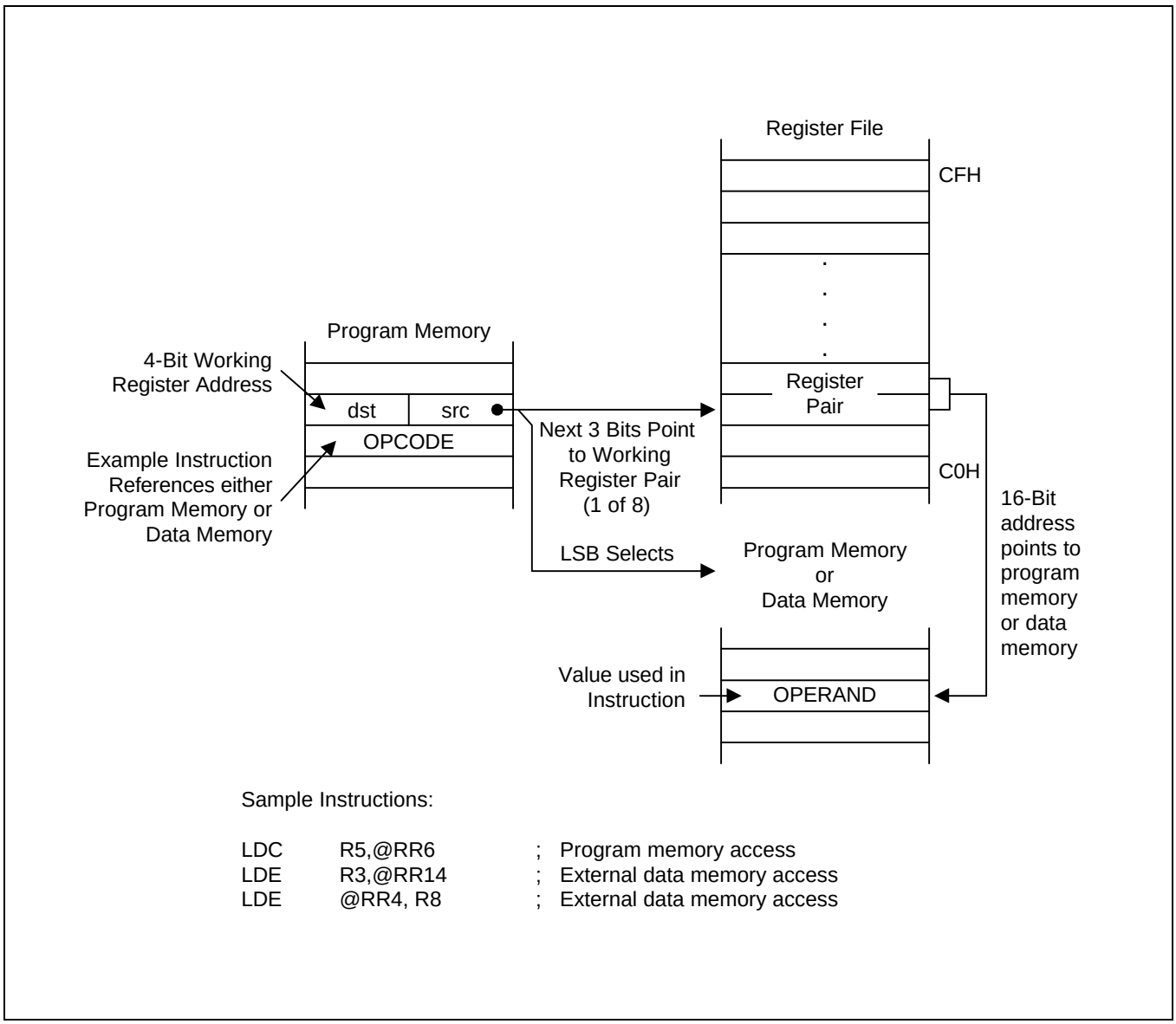


Figure 3-6. Indirect Working Register Addressing to Program or Data Memory

INDEXED ADDRESSING MODE (X)

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

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

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

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

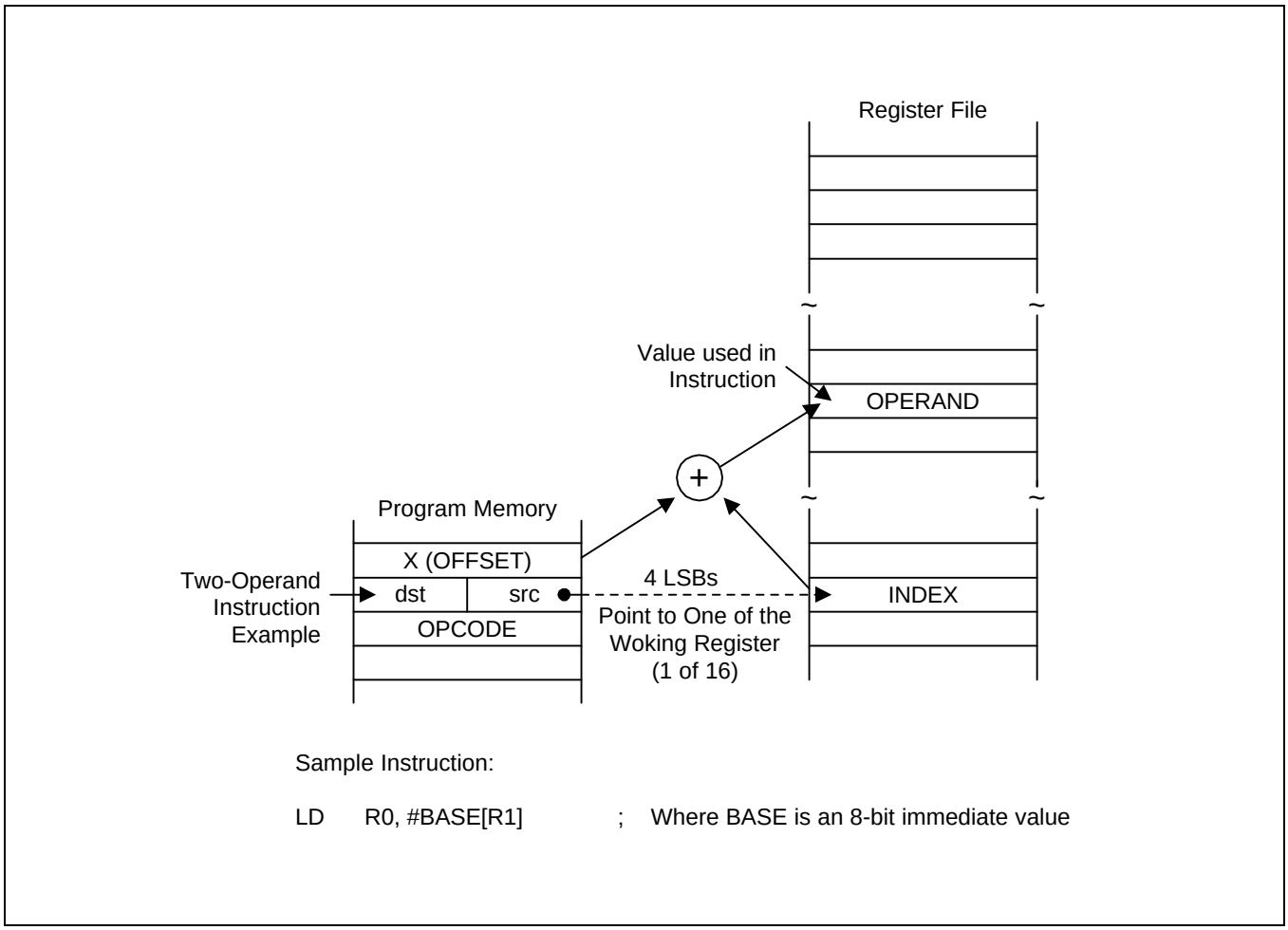
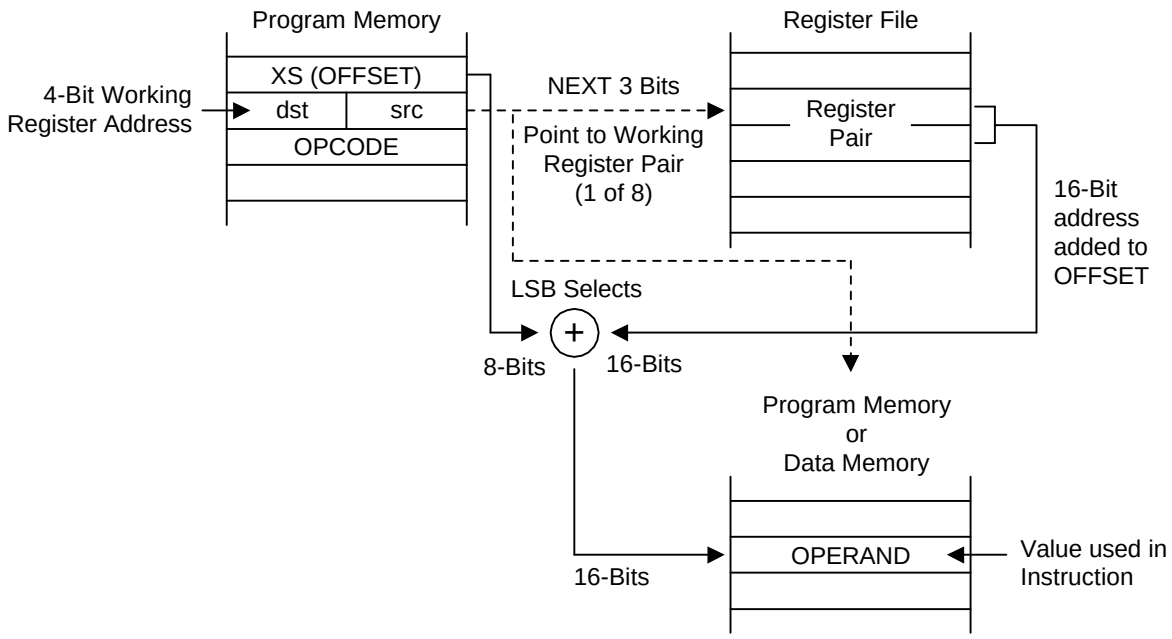


Figure 3-7. Indexed Addressing to Register File

INDEXED ADDRESSING MODE (Continued)

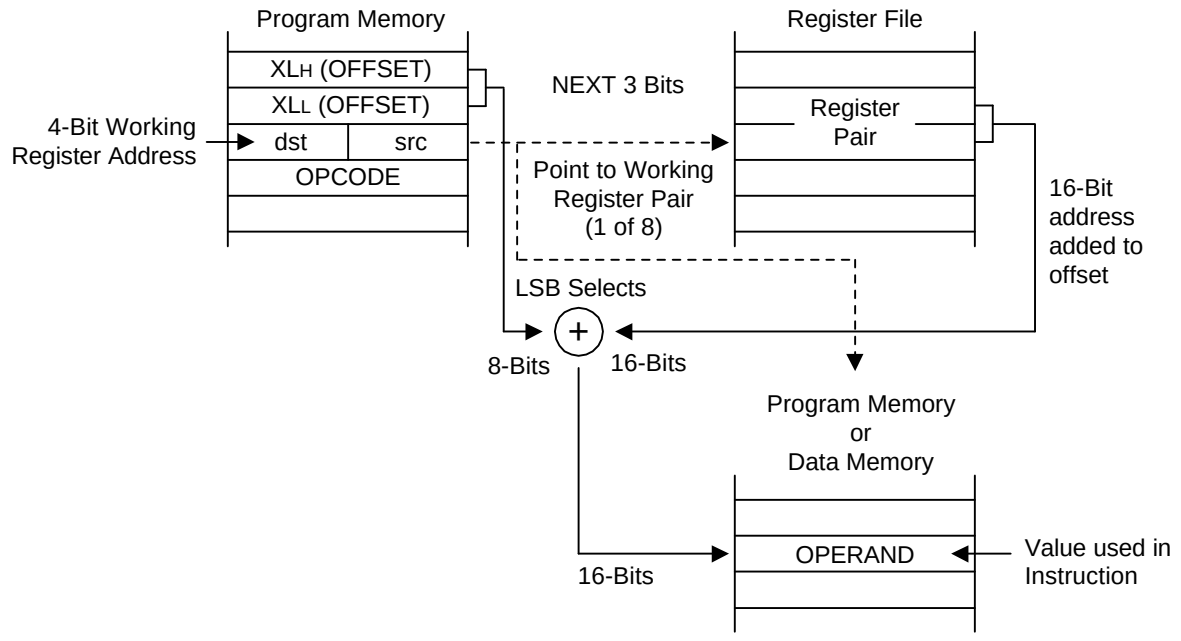


Sample Instructions:

```
LDC     R4, #04H[RR2]     ; The values in the program address (RR2 + #04H)
                          ; are loaded into register R4.
LDE     R4,#04H[RR2]      ; Identical operation to LDC example, except that
                          ; external program memory is accessed.
```

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

INDEXED ADDRESSING MODE (Concluded)



Sample Instructions:

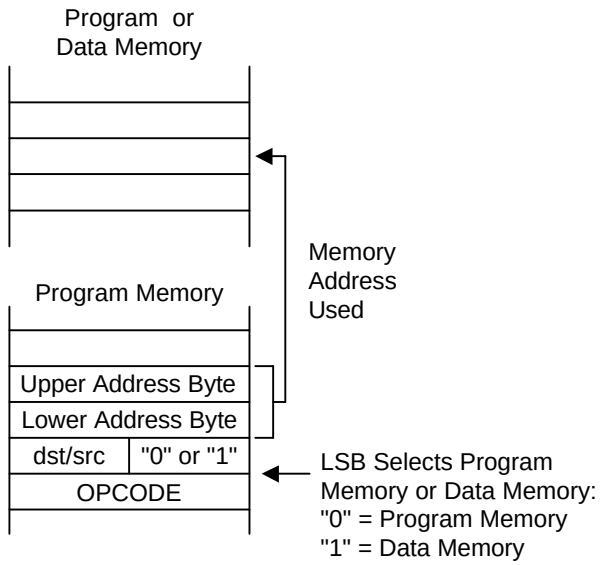
```
LDC     R4, #1000H[RR2]     ;  The values in the program address (RR2 + #1000H)
                               are loaded into register R4.
LDE     R4, #1000H[RR2]     ;  Identical operation to LDC example, except that
                               external program memory is accessed.
```

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

DIRECT ADDRESS MODE (DA)

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

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

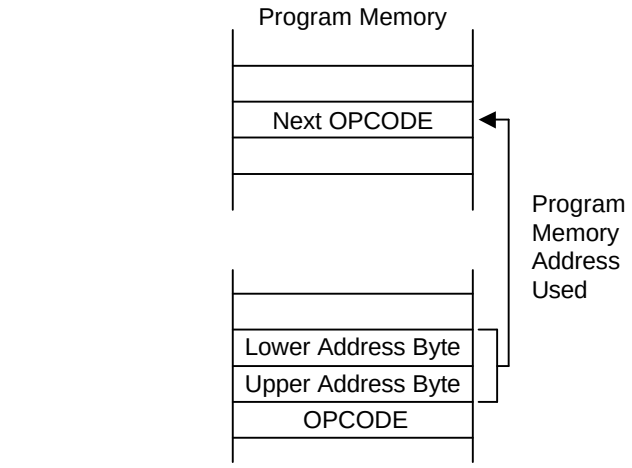


Sample Instructions:

```
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:

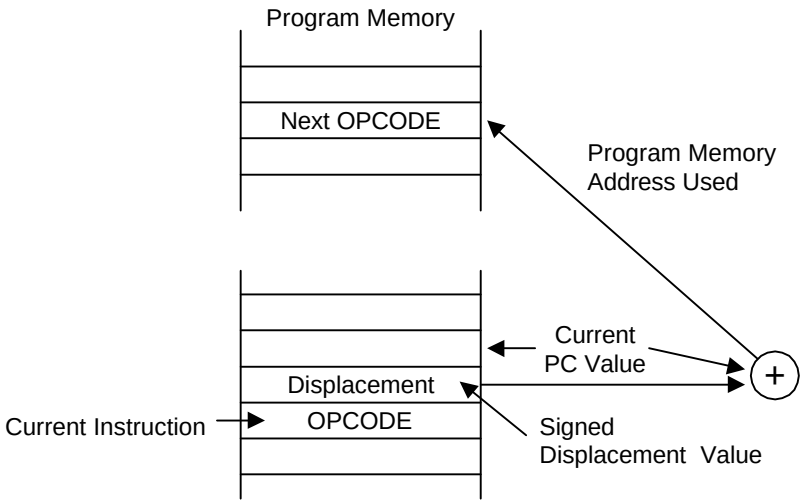
```
JP      C,JOB1      ;  Where JOB1 is a 16-Bit immediate address
CALL    DISPLAY     ;  Where DISPLAY is a 16-Bit immediate address
```

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.



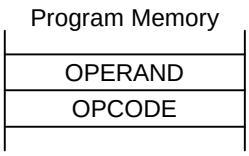
Sample Instructions:

```
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 S3C9432/C9434 control registers are presented in an easy-to-read format. These descriptions will help familiarize you with the mapped locations in the register file. You can also use them as a quick-reference source when writing application programs.

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

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

Table 4-1. System and Peripheral control Registers

Register Name	Mnemonic	Hex	R/W
Timer 0 counter register	T0CNT	D0H	R
Timer 0 data register	T0DATA	D1H	R/W
Timer 0 control register	T0CON	D2H	R/W
Location D3H is not mapped.			
Clock control register	CLKCON	D4H	R/W
System flags register	FLAGS	D5H	R/W
Locations D6H–D8H are not mapped.			
Stack pointer register	SP	D9H	R/W
Location DAH is reserved.			
MDS special register	MDSREG	DBH	R/W
Basic timer control register	BTCON	DCH	R/W
Basic timer counter	BTCNT	DDH	R
Test mode control register	FTSTCON	DEH	W
System mode register	SYM	DFH	R/W

NOTE: The factory test mode register, FTSTCON, is for factory use only. Its value should always be '00H' during the normal operation.

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

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
Locations E3H-E5H are not mapped.			
Port 0 control register	P0CON	E6H	R/W
Port 0 open-drain & pull-up control register	P0DPUR	E7H	R/W
Port 0 interrupt pending register	P0PND	E8H	R/W
Port 1 control register (high byte)	P1CONH	E9H	R/W
Port 1 control register (low byte)	P1CONL	EAH	R/W
Port 1 open-drain & pull-up control register	P1DPUR	EBH	R/W
Port 2 control register	P2CON	ECH	R/W
Locations EDH-EEH are not mapped.			
SIO data register	SIODATA	EFH	R/W
SIO control register	SIOCON	F0H	R/W
SIO prescaler	SIOPS	F1H	R/W
PWM data register	PWMDATA	F2H	R/W
PWM extension data register	PWMEX	F3H	R/W
PWM control register	PWMCON	F4H	R/W
Location F5H is not mapped.			
Prescaler for buzzer output	BUZPS	F6H	R/W
A/D control register	ADCON	F7H	R/W
A/D converter data register (high byte)	ADDATAH	F8H	R
A/D converter data register (low byte)	ADDATAL	F9H	R
Locations FAH-FFH are not mapped.			

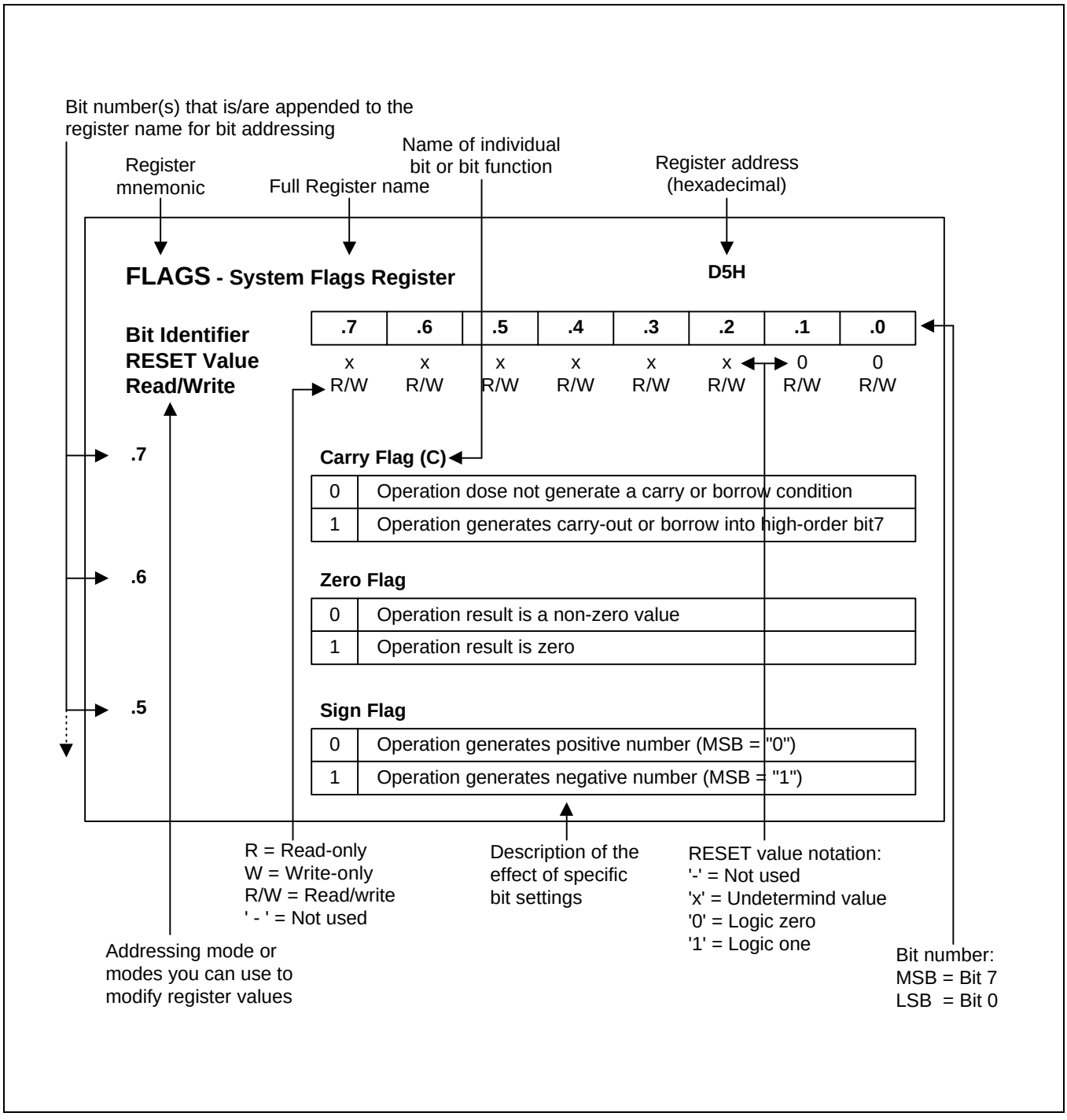


Figure 4-1. Register Description Format

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	ADC0 (P1.0)
0	0	0	1	ADC1 (P1.1)
0	0	1	0	ADC2 (P1.2)
0	0	1	1	ADC3 (P1.3)
0	1	0	0	ADC4 (P1.4)
0	1	0	1	Internally connected with GND
• • •				Internally connected with GND
1	1	1	0	Internally connected with GND
1	1	1	1	Internally connected with AVref

.3 End-of-Conversion Status Bit

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

.2-.1 Clock Source Selection Bit (note)

0	0	f_{OSC} /16 ($f_{OSC} \leq$ 16 MHz)
0	1	f_{OSC} /8 ($f_{OSC} \leq$ 16 MHz)
1	0	f_{OSC} /4 ($f_{OSC} \leq$ 10 MHz)
1	1	f_{OSC} /1 ($f_{OSC} \leq$ 2.5 MHz)

.0 Conversion Start Bit

0	No meaning
1	A/D conversion start

NOTE: Maximum ADC clock input = 2.5 MHz.

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	$f_{OSC}/4096$
0	1	$f_{OSC}/1024$
1	0	$f_{OSC}/128$
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 divider (or basic timer counter) 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)]

NOTE: When P0.0/Buzzer is used as Buzzer output pin, the initial value is 0. When the bit 7 of BUZPS is set to 0, the output of P0.0 is Low.

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-.5 Not used for S3C9432/C9434/P9434

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

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

.2-.0 Not used for S3C9432/C9434/P9434

NOTES:

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

FLAGS — System Flags Register D5H

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	x	x	x	x	–	–	–	–
Read/Write	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 S3C9432/C9434/P9434

P0CON — Port 0 Control Register 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, P0.3/INT1 Configuration Bits**

0	0	Schmitt trigger input; INT1 interrupt disable
0	1	Schmitt trigger input; interrupt on falling edge
1	0	Push-pull output
1	1	Schmitt trigger input; interrupt on rising edge

.5-.4 **Port 0, P0.2/T0CK/INT0 Configuration Bits**

0	0	Schmitt trigger input (T0CK input); INT0 interrupt disable
0	1	Schmitt trigger input (T0CK input); interrupt on falling edge
1	0	Push-pull output
1	1	Schmitt trigger input (T0CK input); interrupt on rising edge

.3-.2 **Port 0, P0.1/PWM Configuration Bits**

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

.1-.0 **Port 0, P0.0/BUZ Configuration Bits**

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

P0DPUR — Port 0 Pull-up Resistor Enable Register 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 Port 0.3/INT1 N-Channel Open-Drain Enable Bit

0	Configure P0.3 as a push-pull output
1	Configure P0.3 as a n-channel open-drain output

.6 Port 0.2/T0CK/INT0 N-Channel Open-Drain Enable Bit

0	Configure P0.2 as a push-pull output
1	Configure P0.2 as a n-channel open-drain output

.5 Port 0.1/PWM N-Channel Open-Drain Enable Bit

0	Configure P0.1 as a push-pull output
1	Configure P0.1 as a n-channel open-drain output

.4 Port 0.0/BUZ N-Channel Open-Drain Enable Bit

0	Configure P0.0 as a push-pull output
1	Configure P0.0 as a n-channel open-drain output

.3 Port 0.3/INT1 Pull-up Resistor Enable Bit

0	Disable pull-up resistor
1	Enable pull-up resistor

.2 Port 0.2/T0CK/INT0 Pull-up Resistor Enable Bit

0	Disable pull-up resistor
1	Enable pull-up resistor

.1 Port 0.1/PWM Pull-up Resistor Enable Bit

0	Disable pull-up resistor
1	Enable pull-up resistor

.0 Port 0.0/BUZ Pull-up Resistor Enable Bit

0	Disable pull-up resistor
1	Enable pull-up resistor

NOTE: In order to use the open-drain output mode, the push-pull output bit in the P0CON should be set first.

P0PND — Port 0 Interrupt Pending Register E8H

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

.7-.2

Not used for the S3C9432/C9434/P9434

.1 Port 0.3/INT1 Interrupt Pending Bit

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

.0 Port 0.2/INT0 Interrupt Pending Bit

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

NOTES:

1. To clear an interrupt pending condition at a port0 pin, you must write a "0" to the corresponding P0PND bit location.
2. To avoid programming errors, we recommend using load instructions when manipulating P0PND values.

P1CONH — Port 1 Control Register (High Byte) E9H

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 the S3C9432/C9434/P9434

.2-.0 **Port 1, P1.4/ADC4/CLO Configuration Bits**

0	0	x	Schmitt trigger input
0	1	0	Schmitt trigger input; pull-up enable
0	1	1	A/D converter input (ADC4); Schmitt trigger input off
1	0	0	Push-pull output
1	0	1	Open-drain output
1	1	0	Open-drain output; pull-up enable
1	1	1	Alternative function; CLO output

P1CONL — Port 1 Control Register (Low Byte) EAH

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

.7-.6 Port 1, P1.3/ADC3/SCK Configuration Bits

0	0	Schmitt trigger input; SCK input
0	1	Alternative function; SCK output
1	0	Push-pull output
1	1	A/D converter input (ADC3); Schmitt trigger input off

.5-.4 Port 1, P1.2/ADC2/SO Configuration Bits

0	0	Schmitt trigger input
0	1	Alternative function; SO output
1	0	Push-pull output
1	1	A/D converter input (ADC2); Schmitt trigger input off

.3-.2 Port 1, P1.1/ADC1/SI Configuration Bits

0	0	Schmitt trigger input; SI input
0	1	Schmitt trigger input; SI input
1	0	Push-pull output
1	1	A/D converter input (ADC1); Schmitt trigger input off

.1-.0 Port 1, P1.0/ADC0 Configuration Bits

0	0	Schmitt trigger input
0	1	Schmitt trigger input
1	0	Push-pull output
1	1	A/D converter input (ADC0); Schmitt trigger input off

P1DPUR — Port 1 Open-drain & Pull-up Control Register 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 Port 1.3 N-channel Open-drain Enable Bit

0	Configure P1.3 as a push-pull output
1	Configure P1.3 as a n-channel open drain output

.6 Port 1.2 N-channel Open-drain Enable Bit

0	Configure P1.2 as a push-pull output
1	Configure P1.2 as a n-channel open drain output

.5 Port 1.1 N-channel Open-drain Enable Bit

0	Configure P1.1 as a push-pull output
1	Configure P1.1 as a n-channel open drain output

.4 Port 1.0 N-channel Open-drain Enable Bit

0	Configure P1.0 as a push-pull output
1	Configure P1.0 as a n-channel open drain output

.3 Port 1.3 Pull-up Resistor Enable Bit

0	Disable pull-up resistor
1	Enable pull-up resistor

.2 Port 1.2 Pull-up Resistor Enable Bit

0	Disable pull-up resistor
1	Enable pull-up resistor

.1 Port 1.1 Pull-up Resistor Enable Bit

0	Disable pull-up resistor
1	Enable pull-up resistor

.0 Port 1.0 Pull-up Resistor Enable Bit

0	Disable pull-up resistor
1	Enable pull-up resistor

NOTE: In order to use the open-drain output mode, the push-pull output bit in the P1CONL should be set first.

P2CON — Port 2 Control Register ECH

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

.7-.6 Port 2, P2.3 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-.4 Port 2, P2.2 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-.2 Port 2, P2.1/SO Configuration Bits

0	0	Push-pull output
0	1	Alternative function; SO output
1	0	Open-drain output; pull-up resistor disable
1	1	Open-drain output; pull-up resistor enable

.1-.0 Port 2, P2.0/SCK Configuration Bits

0	0	Push-pull output
0	1	Alternative function; SCK output
1	0	Open-drain output; pull-up resistor disable
1	1	Open-drain output; pull-up resistor enable

NOTE: P2.0/SCK can be used only as the output mode, so it can not be used as the external clock input.

PWMCON — PWM Control Register F4H

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

.7-.6 PWM Input Clock Selection Bits

0	0	$f_{OSC}/256$
0	1	$f_{OSC}/64$
1	0	$f_{OSC}/8$
1	1	$f_{OSC}/1$

.5-.4

Not used for the S3C9432/C9434/P9434

.3 PWM Counter Clear Bit

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

.2 PWM Counter Enable Bit

0	Stop counter
1	Start (Resume Counting)

.1 PWM Overflow Interrupt Enable Bit (12-bit overflow)

0	Disable interrupt
1	Enable interrupt

.0 PWM Overflow Interrupt Pending Bit

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

SIOCON — Serial I/O Module Control Register F0H

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

.7 SIO Shift Clock Selection Bit

0	Internal 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 shifter and clock counter
1	Enable shifter and clock counter

.1 SIO Interrupt Enable Bit

0	Disable SIO interrupt
0	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 S3C9432/C9434/P9434

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

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

.1-.0 **Page Selection Bits**

0	0	page 0
0	1	page 1 (not used for S3C9432/C9434/P9434)
1	0	page 2 (not used for S3C9432/C9434/P9434)
1	1	page 3 (not used for S3C9432/C9434/P9434)

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

T0CON — TIMER 0 Control Register D3H

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

.7-.6 Timer 0 Input Clock Selection Bits

0	0	$f_{OSC}/4096$
0	1	$f_{OSC}/256$
1	0	$f_{OSC}/8$
1	1	T0CK

.5-.4

Note used for the S3C9432/C9434/P9434

.3 Timer 0 Counter Clear Bit

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

.2

Note used for the S3C9432/C9434/P9434

.1 Timer 0 Interrupt Enable Bit

0	Disable interrupt
1	Enable interrupt

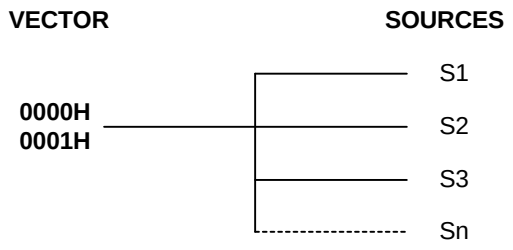
.0 Timer 0 Interrupt Pending Bit (Match Interrupt)

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.



NOTES:
1. The SAM87Ri interrupt has only one vector address (0000H-0001H).
2. The number of Sn value is expandable.

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.

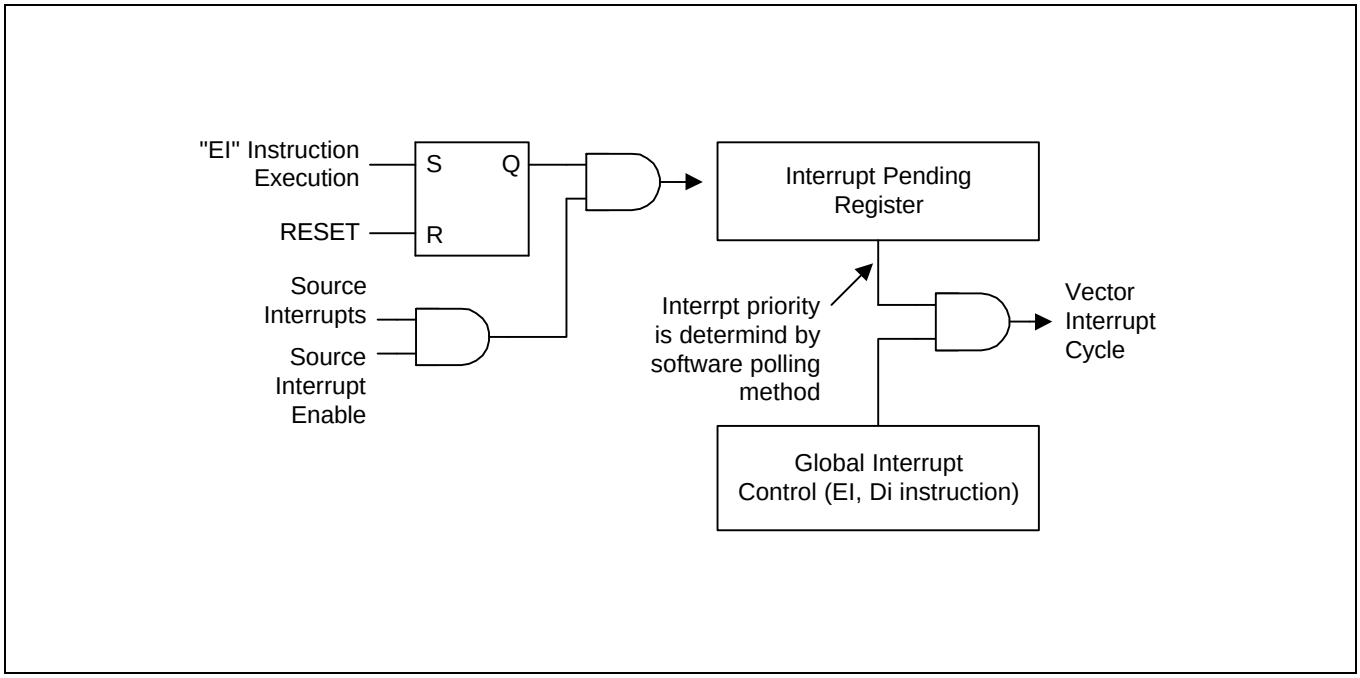


Figure 5-2. Interrupt Function Diagram

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.

S3C9432/C9434 INTERRUPT STRUCTURE

The S3C9432/C9434 microcontroller has five peripheral interrupt sources:

— PWM overflow
— Timer 0 match
— SIO interrupt
— P0.2 external interrupt
— P0.3 external interrupt

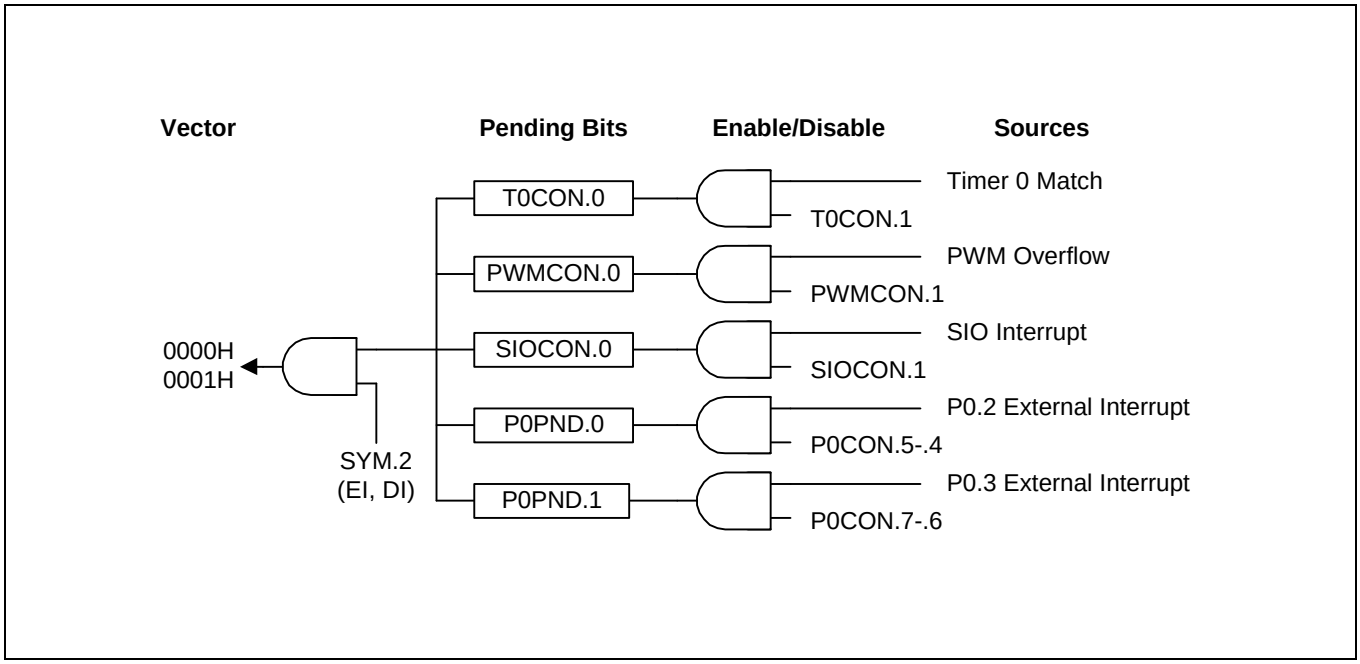


Figure 5-3. S3C9432/C9434 Interrupt Structure

7 CLOCK CIRCUIT

OVERVIEW

An RC oscillation source provides a typical 4 MHz clock for S3C9432/C9434. 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.

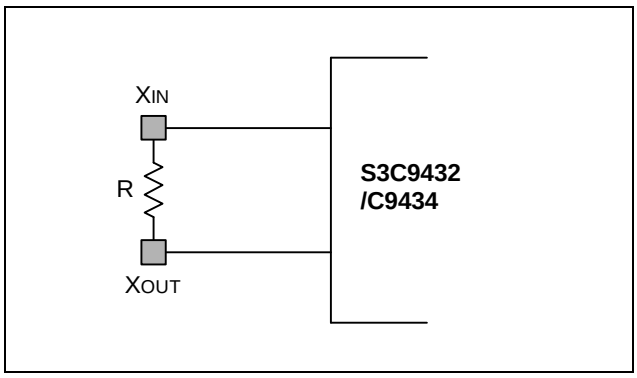


Figure 7-1. Main Oscillator Circuit (RC Oscillator with Internal Capacitor)

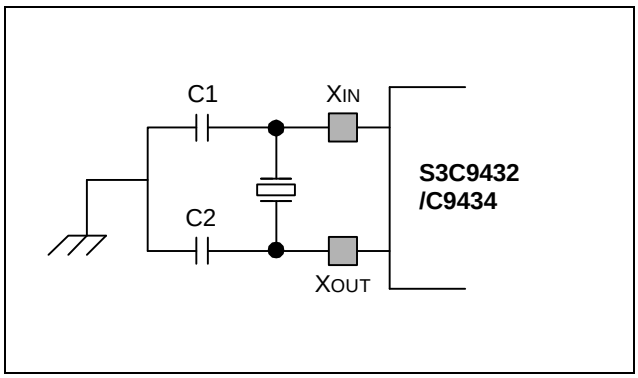


Figure 7-2. Main Oscillator Circuit (Crystal/Ceramic Oscillator)

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 S3C9432/C9434, 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 f_{OSC}, $f_{OSC}/2$ or $f_{OSC}/8$.

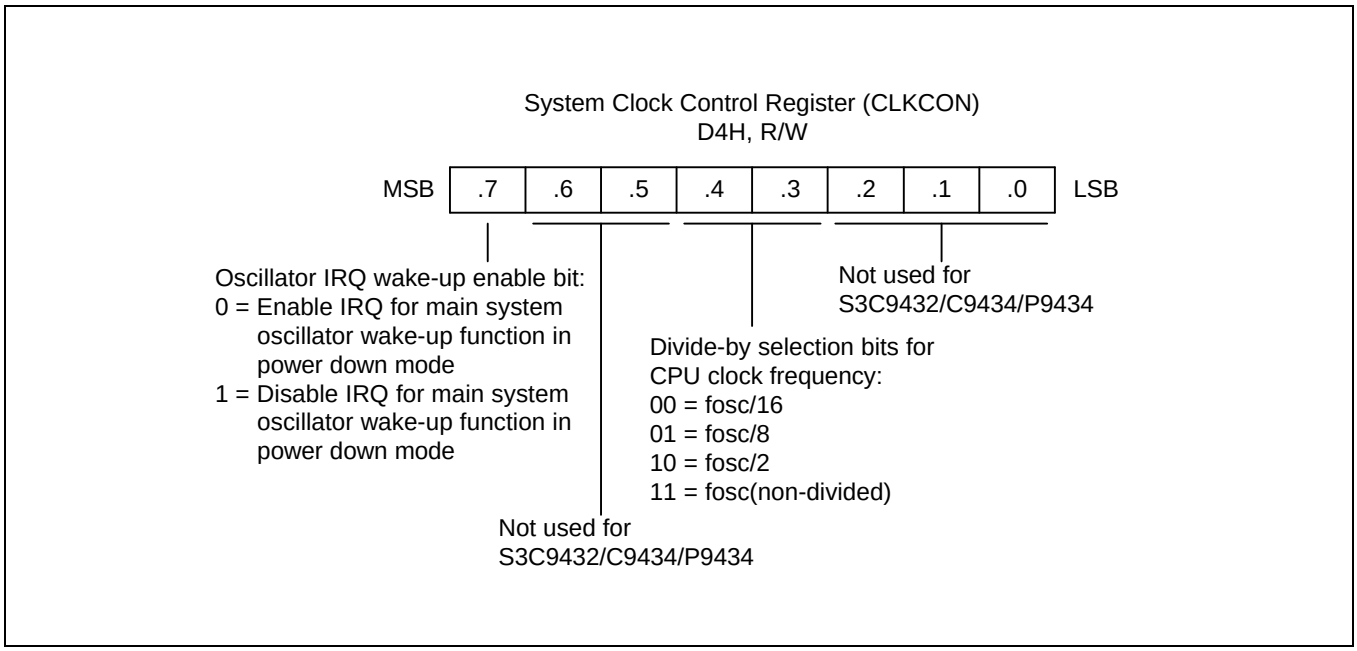
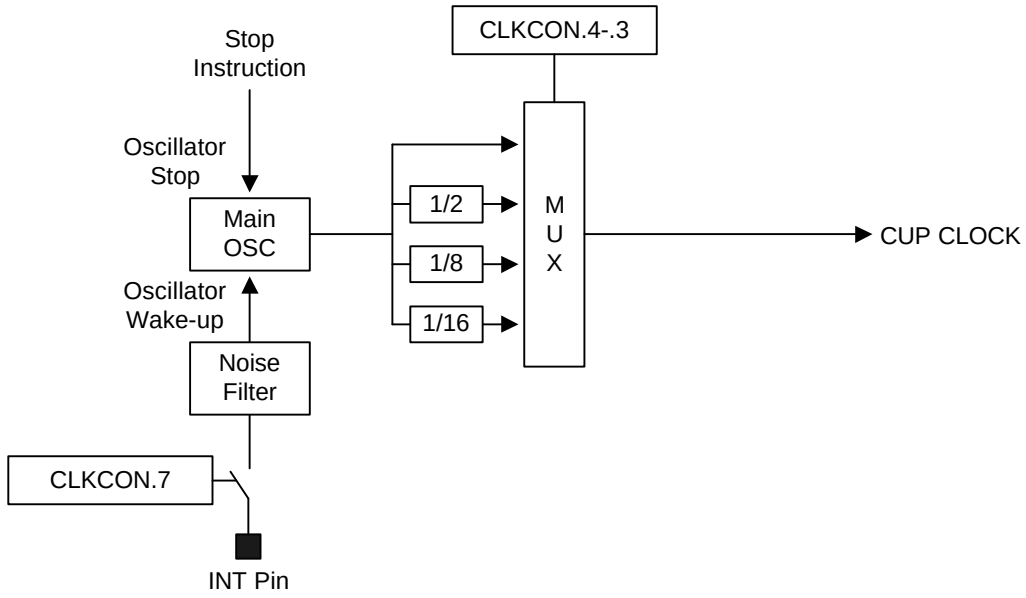


Figure 7-3. System Clock Control Register (CLKCON)



NOTE: An external interrupt with an RC-delay nosie fillter can be used to release Stop mode and "wake-up" the main oscillator . In the S3C9432/C9434, the INT0-INT1 external interrupts and SIO interrupt are of this type.

Figure 7-4. System Clock Circuit Diagram

8 RESET and POWER-DOWN

SYSTEM RESET

OVERVIEW

The S3C9432/C9434 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 and the RESET pin is forced to Low level. The RESET signal is input through a Schmitt trigger circuit where it is then synchronized with the CPU clock. This brings the S3C9432/C9434 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 ($\cong 2^{16}/f_{OSC}$, f_{OSC} = 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. 2.8 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.

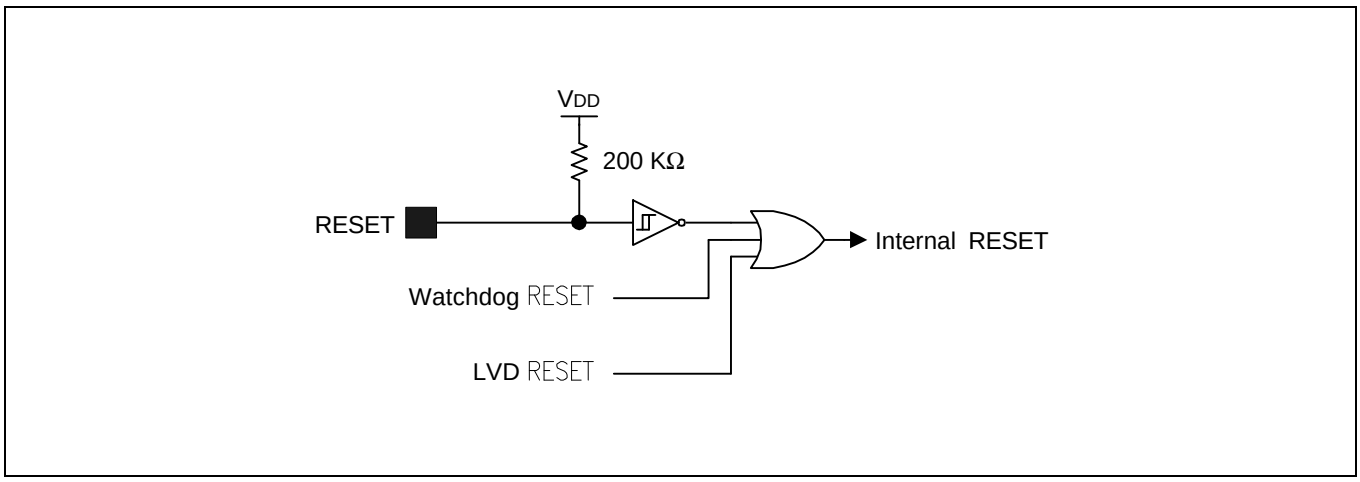


Figure 8-1. Reset Block Diagram

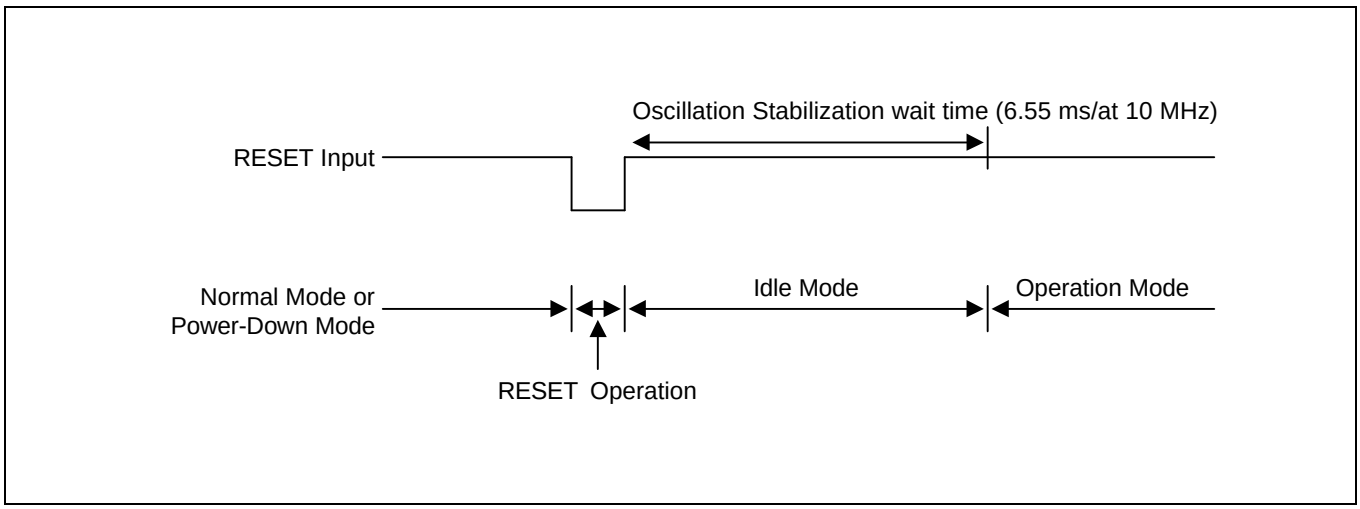


Figure 8-2. Timing for Oscillation Stabilization after RESET

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 100 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 (f_{OSC}/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 S3C9432/C9434 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 (f_{OSC}/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 ADC 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 Values after a Reset

Register Name	Mnemonic	Address	Bit Values After RESET							
			7	6	5	4	3	2	1	0
Timer 0 counter register	T0CNT	D0H	0	0	0	0	0	0	0	0
Timer 0 data register	T0DATA	D1H	1	1	1	1	1	1	1	1
Timer 0 control register	T0CON	D2H	0	0	0	0	0	0	0	0
Location D3H is not mapped.										
Clock control register	CLKCON	D4H	0	0	0	0	0	0	0	0
System flags register	FLAGS	D5H	x	x	x	x	–	–	–	–
Locations D6H-D8H are not mapped.										
Stack pointer register	SP	D9H	x	x	x	x	x	x	x	x
Location DAH is not mapped.										
MDS special register	MDSREG	DBH	0	0	0	0	0	0	0	0
Basic timer control register	BTCON	DCH	0	0	0	0	0	0	0	0
Basic timer counter	BTCNT	DDH	0	0	0	0	0	0	0	0
Test mode control register	FTSTCON	DEH	–	–	0	0	0	0	0	0
System mode register	SYM	DFH	–	–	–	–	–	0	0	0

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

Bank 0 Register Name	Mnemonic	Address	Bit Values After a Reset							
			7	6	5	4	3	2	1	0
Port 0 data register	P0	E0H	–	–	–	–	0	0	0	0
Port 1 data register	P1	E1H	–	–	–	0	0	0	0	0
Port 2 data register	P2	E2H	–	–	–	–	0	0	0	0
Locations E3H-E5H are not mapped.										
Port 0 control register	P0CON	E6H	0	0	0	0	0	0	0	0
Port 0 open-drain & pull-up control register	P0DPUR	E7H	0	0	0	0	0	0	0	0
Port 0 interrupt pending register	P0PND	E8H	–	–	–	–	–	–	0	0
Port 1 control register (high byte)	P1CONH	E9H	–	–	–	–	–	0	0	0
Port 1 control register (low byte)	P1CONL	EAH	0	0	0	0	0	0	0	0
Port 1 open-drain & pull-up control register	P1DPUR	EBH	0	0	0	0	0	0	0	0
Port 2 interrupt pending register	P2CON	ECH	0	0	0	0	0	0	0	0
Locations EDH-EEH are not mapped.										
SIO data register	SIODATA	EFH	0	0	0	0	0	0	0	0
SIO control register	SIOCON	F0H	0	0	0	0	0	0	0	0
SIO prescaler	SIOPS	F1H	0	0	0	0	0	0	0	0
PWM data register	PWMDATA	F2H	–	–	0	0	0	0	0	0
PWM extension data register	PWMEX	F3H	0	0	0	0	0	0	–	–
PWM control register	PWMCON	F4H	0	0	–	–	0	0	0	0
Location F5H is not mapped.										
Prescaler for buzzer output	BUZPS	F6H	0	0	0	0	0	0	0	0
A/D control register	ADCON	F7H	–	0	0	0	0	–	–	0
A/D converter data register (high byte)	ADDATAH	F8H	x	x	x	x	x	x	x	x
A/D converter data register (low byte)	ADDATAL	F9H	0	0	0	0	0	0	x	x
Locations FAH-FFH are not mapped.										

NOTE: "–" means not mapped, "x" means undefined.

☞ PROGRAMMING TIP — Sample S3C9434 Initialization Routin

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

            ORG       0000H
            VECTOR    00H,INT_4304          ;  S3C9434 has only one interrupt vector

;--------------<< Initialize System and Peripherals >>

            ORG       0100H
RESET:      DI                              ;  disable interrupt
            LD        BTCON,#10100011B      ;  Watch-dog disable
            LD        CLKCON,#00011000B     ;  Select non-divided cpu clock
            LD        SP,#60H               ;  Stack pointer must be set

            LD        P0CON,#01011010B      ;  p0.3/0.2 input // p0.1/0.0 output
            LD        P0DPUR,#00001111B     ;  p0.0-0.3 pull-up enable
            LD        P1CONH,#00000100B     ;  p1.4 push-pull output
            LD        P1CONL,#00000000B     ;  p1.0-p1.3 schmitt trigger input
            LD        P1DPUR,#00001111B     ;  p1.0-p1.3 pull-up resistor enable
            LD        P2CON,#00000000B      ;  p2.0-p2.3 push-pull output

;--------------<< Timer 0 settings >>

            LD        T0DATA,#50H           ;  CPU = 11.0592Mhz, interrupt interval = 2 msec
            LD        T0CON,#01001010B      ;  Fosc/256, Timer0 interrupt enable

;--------------<< Clear all data registers from 00h to 5Fh >>

            LD        R0,#0                 ;  RAM clear
RAM_CLR:    CLR       @R0                   ;
            INC       R0                    ;
            CP        R0,#5FH               ;
            JP        ULE,RAM_CLR

;--------------<< Initialize other registers >>

            •
            •
            •
            EI                              ;  Enable interrupt
```

☞ PROGRAMMING TIP — Sample S3C9434 Initialization Routine (Continued)

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

MAIN:       NOP                         ;  Start main loop
            LD          BTCON,#02H      ;  Enable watchdog function
                                        ;  Basic counter (BTCNT) clear
            •
            •
            CALL        KEY_SCAN        ;  Sub-block module
            •
            •
            •
            CALL        LED_DISPLAY     ;  Sub-block module
            •
            •
            •
            CALL        JOB             ;  Sub-block module
            •
            •
            •
            JR          T,MAIN          ;

;--------------<< Subroutines >>

KEY_SCAN:   NOP                         ;  Sub-block module
            •
            •
            •
            RET

LED_DISPLAY: NOP                        ;  Sub-block module
            •
            •
            •
            RET

JOB:        NOP                         ;  Sub-block module
            •
            •
            •
            RET
```

☞ PROGRAMMING TIP — Sample S3C9434 Initialization Routine (Continued)

```
;--------------<< Interrupt Service Routines >>          ;  Interrupt enable bit and pending bit check

INT_4304:     TM      T0CON,#00000010B    ;  Timer0 interrupt enable check
              JR      Z,NEXT_CHK1         ;
              TM      T0CON,#00000001B    ;  If timer0 interrupt was occurred,
              JP      NZ,INT_TIMER0       ;  T0CON.0 bit would be set.

NEXT_CHK1:    TM      SIOCON,#00000010B   ;  SIO interrupt enable check
              JR      Z,NEXT_CHK2         ;
              TM      SIOCON,#00000001B   ;
              JP      NZ,SIO_INT          ;

NEXT_CHK2:    TM      PWMCON,#00000010B   ;  PWM overflow interrupt enable check
              JR      Z,NEXT_CHK3         ;
              TM      PWMCON,#00000001B   ;
              JP      NZ,PWMOVF_INT       ;

NEXT_CHK3:    TM      P0PND,#00000001B    ;
              JP      NZ,INT0_INT         ;
              TM      P0PND,#00000010B    ;
              JP      NZ,INT1_INT         ;
              IRET                        ;  Interrupt return

;--------------< Timer0 interrupt service routine >

INT_TIMER0:   •                           ;
              •
              AND     T0CON,#11110110B    ;  Pending bit clear
              IRET                        ;  Interrupt return

;--------------< SIO interrupt service routine >

SIO_INT:      •
              •
              AND     SIOCON,#11111110B   ;  Pending bit clear
              IRET                        ;  Interrupt return

;--------------< PWM overflow interrupt service routine >

PWMOVF_INT:       •
                  •
              AND     PWMCON,#11111110B   ;  Pending bit clear
              IRET                        ;  Interrupt return
```

☞ PROGRAMMING TIP — Sample S3C9434 Initialization Routine (Continued)

```
;--------------< External interrupt0 service routine >

INT0_INT:   •
            •
            LD          P0PND,#00000010B    ;  INT0 Pending bit clear
            IRET                            ;  Interrupt return

;--------------< External interrupt1 service routine >

INT0_INT:   •
            •
            LD          P0PND,#00000001B    ;  INT1 Pending bit clear
            IRET                            ;  Interrupt return
            •
            •
            END                             ;
```

9 I/O PORTS

OVERVIEW

The S3C9432/C9434 has two I/O ports and one output port: with 13 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. S3C9432/C9434 Port Configuration Overview

Port	Function Description	Programmability
0	Bit-programmable I/O port for schmitt trigger input or push-pull, open-drain output. Pull-up resistors are assignable by software. Port0 pins can also be used as alternative function.	Bit
1	Bit-programmable I/O port for schmitt trigger input or push-pull, open-drain output. Pull-up resistors are assignable by software. Port1 pins can also be used as A/D converter input or alternative function.	Bit
2	Push-pull or open drain only output port. Pull-up resistors are assignable by software. Port2 can also be used as alternative function.	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 0-2 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

NOTE: A reset operation clears the P0-P2 data register to "00H".

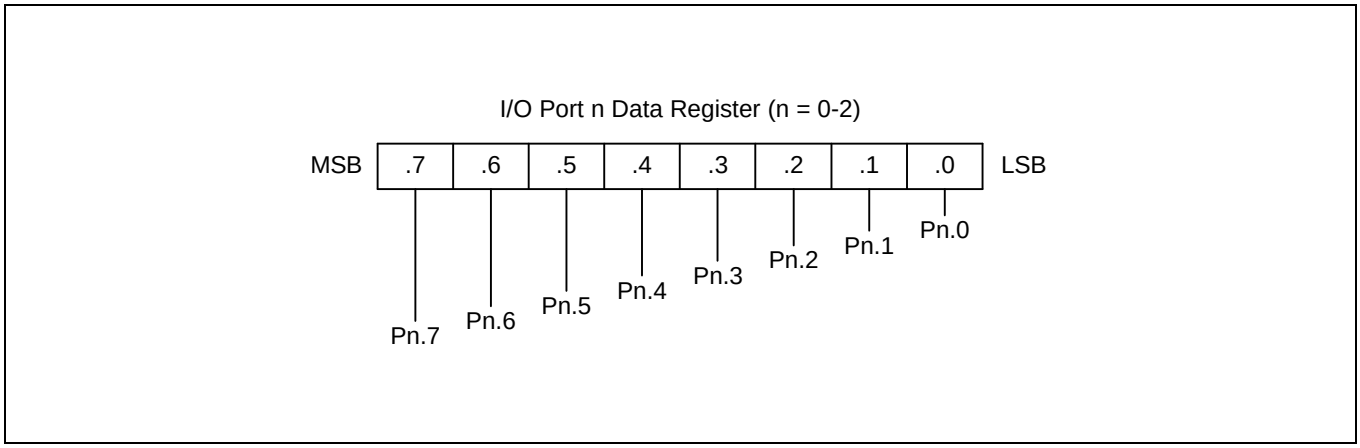


Figure 9-1. Port Data Register Format

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. It is designed for high-current functions such as LED direct drive.

You access port 0 directly by writing or reading the corresponding port data register, P0 (E0H). A reset clears the port control register, P0CON, to "00H" configuring port 0 pins as normal inputs.

Two addition resisters are used to control Port 0: P0DPUR (E7H) and P0PND (E8H). By setting bits in the Port 0 open-drain enable register P0DPUR, you can configure specific pin as a open-drain output.

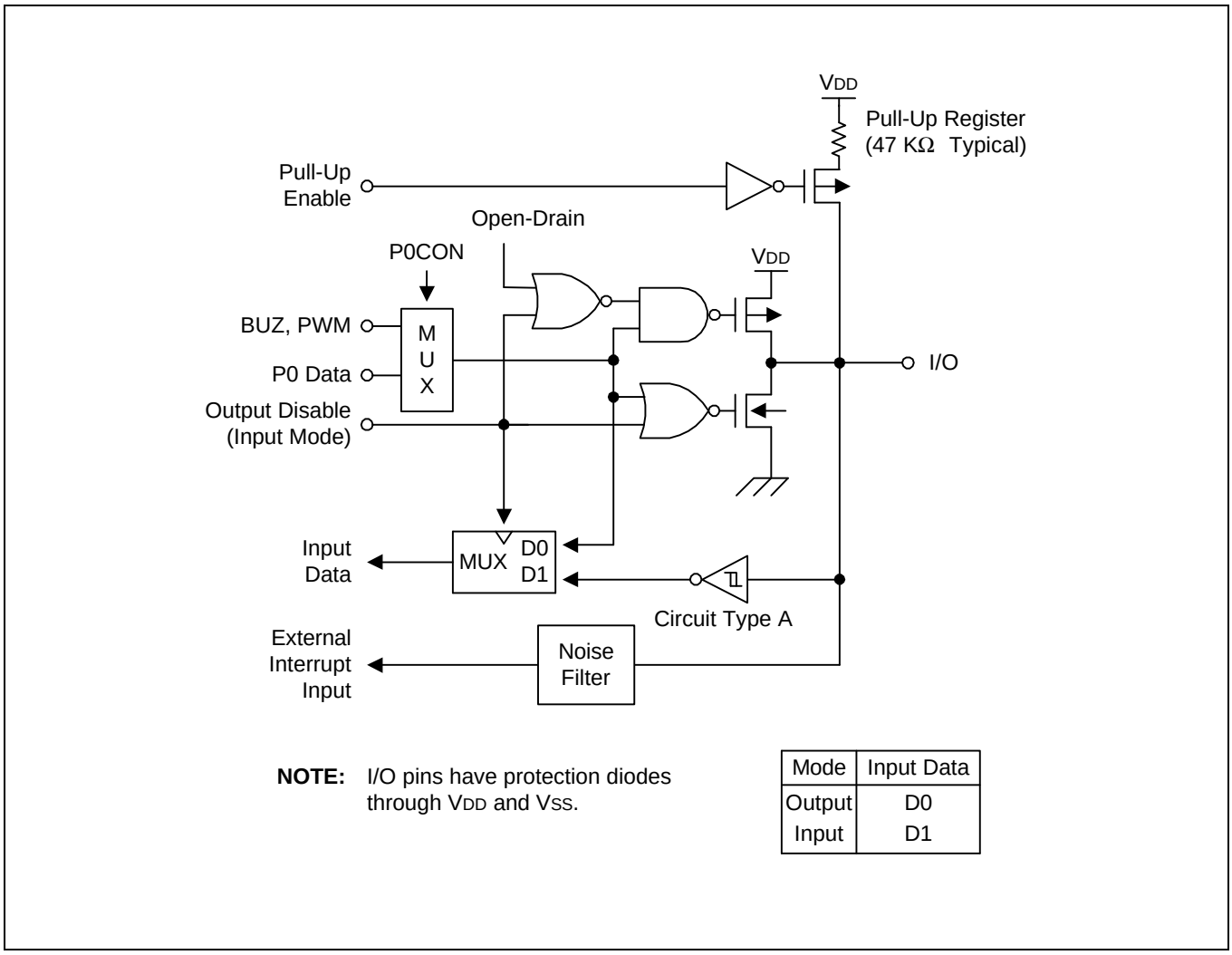


Figure 9-2. Port 0 Circuit Diagram

Port 0 Control Registers
E6H, R/W

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

[.7-.6] Port 0, P0.3/INT1 Configuration Bits
00 = Schmitt trigger input; INT1 interrupt disable
01 = Schmitt trigger input; interrupt on falling edge
10 = Push-pull output
11 = Schmitt trigger input; interrupt on rising edge

[.5-.4] Port 0, P0.2/INT0 Configuration Bits
00 = Schmitt trigger input (T0CK input); INT0 interrupt disable
01 = Schmitt trigger input (T0CK input); interrupt on falling edge
10 = Push-pull output
11 = Schmitt trigger input (T0CK input); interrupt on rising edge

[.3-.2] Port0, P0.1/PWM Configuration Bits
00 = Schmitt trigger input
01 = Schmitt trigger input
10 = Push-pull output
11 = Alternative function (PWM Output)

[.1-.0] Port0, P0.0/BUZ Configuration Bits
00 = Schmitt trigger input
01 = Schmitt trigger input
10 = Push-pull output
11 = Alternative function (BUZ output)

Figure 9-3. Port 0 Control Register (P0CON)

Port 0 N-Channel Open-drain Enable Register
E7H, R/W

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

[.7] Port 0.3/INT1 N-Channel Open-drain Enable Bit
0 = Configure P0.3 as a push-pull output
1 = Configure P0.3 as a N-channel open-drain output

[.6] Port 0.2/T0CK/INT0 N-Channel Open-drain Enable Bit
0 = Configure P0.2 as a push-pull output
1 = Configure P0.2 as a N-channel open-drain output

[.5] Port 0.1/PWM N-Channel Open-drain Enable Bit
0 = Configure P0.1 as a push-pull output
1 = Configure P0.1 as a N-channel open-drain output

[.4] Port 0.0/BUZ N-Channel Open-drain Enable Bit
0 = Configure P0.0 as a push-pull output
1 = Configure P0.0 as a N-channel open-drain output

[.3] Port 0.3/INT1 Pull-up Resistor Enable Bit
0 = Disable pull-up resistor
1 = Enable pull-up resistor

[.2] Port 0.2/T0CK/INT0 Pull-up Resistor Enable Bit
0 = Disable pull-up resistor
1 = Enable pull-up resistor

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

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

Figure 9-4. Port 0 N-channel Open-drain Enable Register (P0DPUR)

Port 0 Interrupt Pending Register
E8H, R/W

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

[.7-.2] Not used for S3C9432/C9434/P9434

[.1] Port 0.3/INT1, Interrupt Pending Bit
0 = No interrupt pending (when read)
0 = Pending bit clear (when write)
1 = Interrupt is pending (when read)

[.1] Port 0.2/INT0, Interrupt Pending Bit
0 = No interrupt pending (when read)
0 = Pending bit clear (when write)
1 = Interrupt is pending (when read)

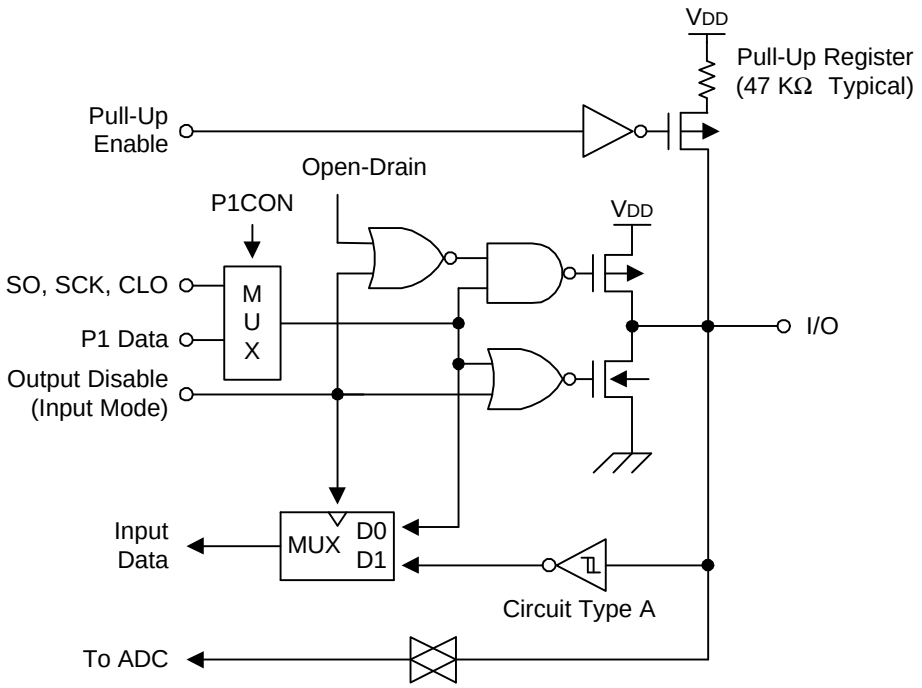
Figure 9-5. Port 0 Interrupt Pending Registers (P0PND)

PORT 1

Port 1, is a 5-bit I/O port with individually configurable pins. It can be used for general I/O port (Schmitt trigger input mode, push-pull output mode or n-channel open-drain output mode). You can also use port1 as special input (ADC, SI or SCK) or output (SO, SCK, CLO). In addition, you can configure a pull-up resistor to individual pin using control register settings. It is designed for high-current functions such as LED direct drive.

In normal operating mode, a reset clears P1CONH and P1CONL to "00H", configuring P1.0-P1.4 as normal Schmitt trigger inputs.

You address port 1 bits directly by writing or reading the port 1 data register, P1 (E1H).



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

Mode	Input Data
Output	D0
Input	D1

Figure 9-6. Port 1 Circuit Diagram

Port 1 Control Register (High-Byte)
E9H, R/W

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

[.7-.3] Not used for S3C9432/C9434/P9434

[.2-.0] Port 1, P1.4/ADC4/CLO Configuration Bits
00x = Schmitt trigger input
010 = Schmitt trigger input; pull-up enable
011 = A/D converter input (ADC4); schmitt trigger input off
100 = Push-pull output
101 = Open-drain output
110 = Open-drain output; pull-up enable
111 = Alternative function; CLO output

Figure 9-7. Port 1 High-Byte Control Register (P1CONH)

Port 1 Control Register (Low-Byte)
EAH, R/W

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

[.7-.6] Port 1, P1.3/ADC3/SCK Configuration Bits
00 = Schmitt trigger input; SCK input
01 = Alternative function; SCK output
10 = Push-pull output
11 = A/D converter input (ADC3); schmitt trigger input off

[.5-.4] Port 1, P1.2/ADC2/SO Configuration Bits
00 = Schmitt trigger input
01 = Alternative function; SO output
10 = Push-pull output
11 = A/D converter input (ADC2); schmitt trigger input off

[.3-.2] Port 1, P1.1/ADC1/SI Configuration Bits
00 = Schmitt trigger input; SI input
01 = Schmitt trigger input; SI input
10 = Push-pull output
11 = A/D converter input (ADC1); schmitt trigger input off

[.1-.0] Port 1, P1.0/ADC0 Configuration Bits
00 = Schmitt trigger input
01 = Schmitt trigger input
10 = Push-pull output
11 = A/D converter input (ADC0); schmitt trigger input off

Figure 9-8. Port 1 Low-Byte Control Register (P1CONL)

Port 1 Open-drain Pull-up Control Register
EBH, R/W

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

[.7] Port 1.3, N-Channel Open-drain Enable Bit
0 = Configure P1.3 as a push-pull output
1 = Configure P1.3 as a N-channel open-drain output

[.6] Port 1.2, N-Channel Open-drain Enable Bit
0 = Configure P1.2 as a push-pull output
1 = Configure P1.2 as a N-channel open-drain output

[.5] Port 1.1, N-Channel Open-drain Enable Bit
0 = Configure P1.1 as a push-pull output
1 = Configure P1.1 as a N-channel open-drain output

[.4] Port 1.0, N-Channel Open-drain Enable Bit
0 = Configure P1.0 as a push-pull output
1 = Configure P1.0 as a N-channel open-drain output

[.3] Port 1.3, Pull-up Resistor Enable Bit
0 = Disable pull-up resistor
1 = Enable pull-up resistor

[.2] Port 1.2, Pull-up Resistor Enable Bit
0 = Disable pull-up resistor
1 = Enable pull-up resistor

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

[.0] Port 1.0, Pull-up Resistor Enable Bit
0 = Disable pull-up resistor
1 = Enable pull-up resistor

Figure 9-9. Port 1 Open-drain & Pull-up Control Register (P1DPUR)

PORT 2

Port 2 is a 4-bit output port with individually configurable pins. It can be used for general output port (push-pull output mode or N-channel open-drain output mode). You can also use port 2 pins as serial clock output or serial data output pin. In addition, you can configure a pull-up resistor to individual pins using control register settings. It is designed for high-current functions such as LED direct drive.

In normal operating mode, a reset clears P2CON to "00H", configuring P2.0-P2.3 as push-pull output.

You address port 2 bits directly by writing or reading the port 2 data register, P2 (E2H). The port 2 control register, P2CON is located at addresses ECH.

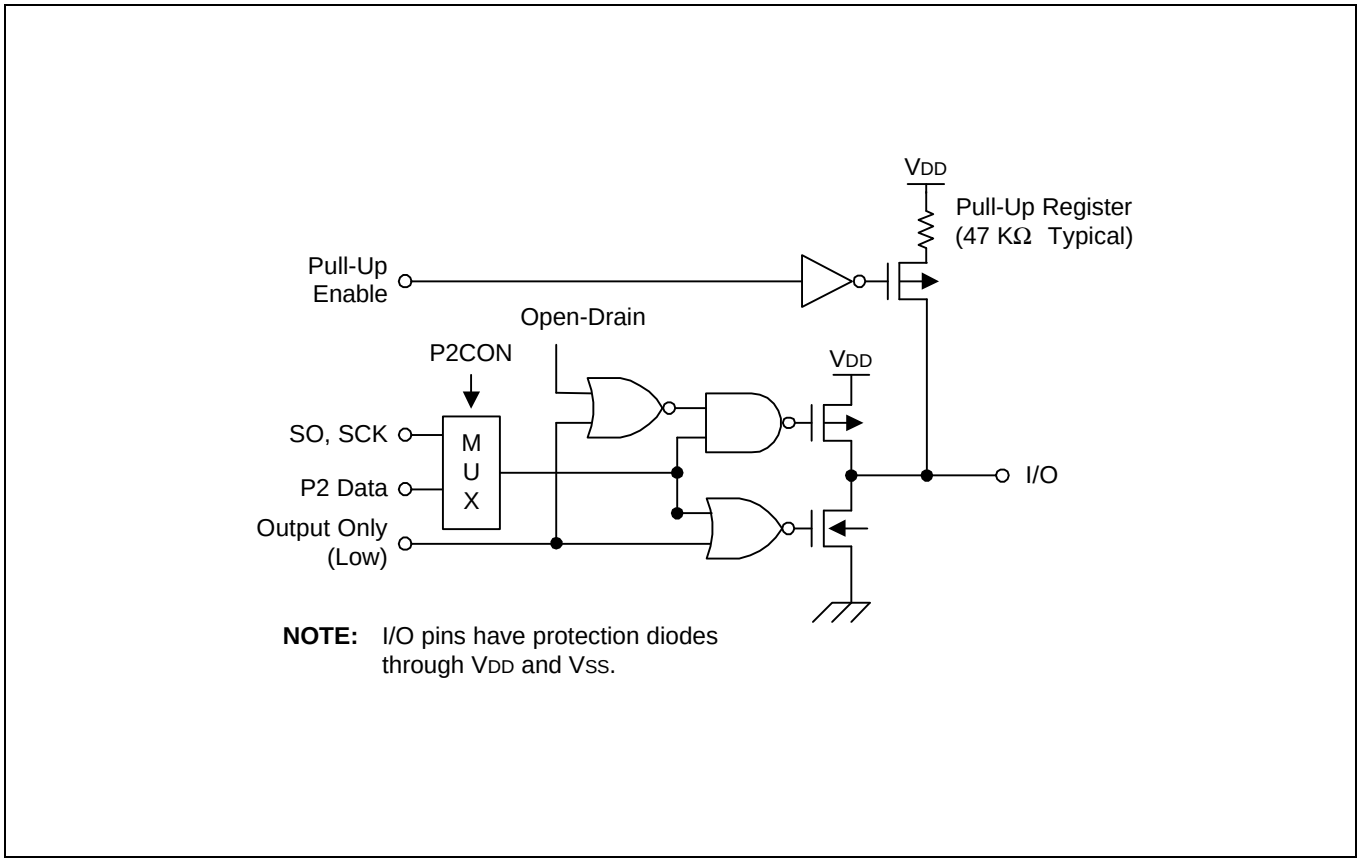


Figure 9-10. Port 2 Circuit Diagram

Port 2 Control Register
ECH, R/W

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

[.7-.6] Port 2, P2.3 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 2, P2.2 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] Port 2, P2.1/SO Configuration Bits
00 = Push-pull output
01 = Alternative function; SO output
10 = Open-drain output; pull-up resistor disable
11 = Open-drain output; pull-up resistor enable

[.1-.0] Port 2, P2.0/SCK Configuration Bits
00 = Push-pull output
01 = Alternative function; SCK output
10 = Open-drain output; pull-up resistor disable
11 = Open-drain output; pull-up resistor enable

Figure 9-11. Port 2 Control Registers (P2CON)

10 BASIC TIMER and TIMER 0

MODULE OVERVIEW

The S3C9432/C9434 has two default timers: an 8-bit *basic timer,* one 8-bit general-purpose timer/counter, called *timer 0*.

Basic Timer (BT)

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

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

The functional components of the basic timer block are:

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

Timer 0

Timer 0 has the following functional components:

— Clock frequency divider (f_{OSC} divided by 4096, 256, 8, or T0CK) with multiplexer
— 8-bit counter (T0CNT), 8-bit comparator, and 8-bit data register (T0DATA)
— Timer 0 control register (T0CON)

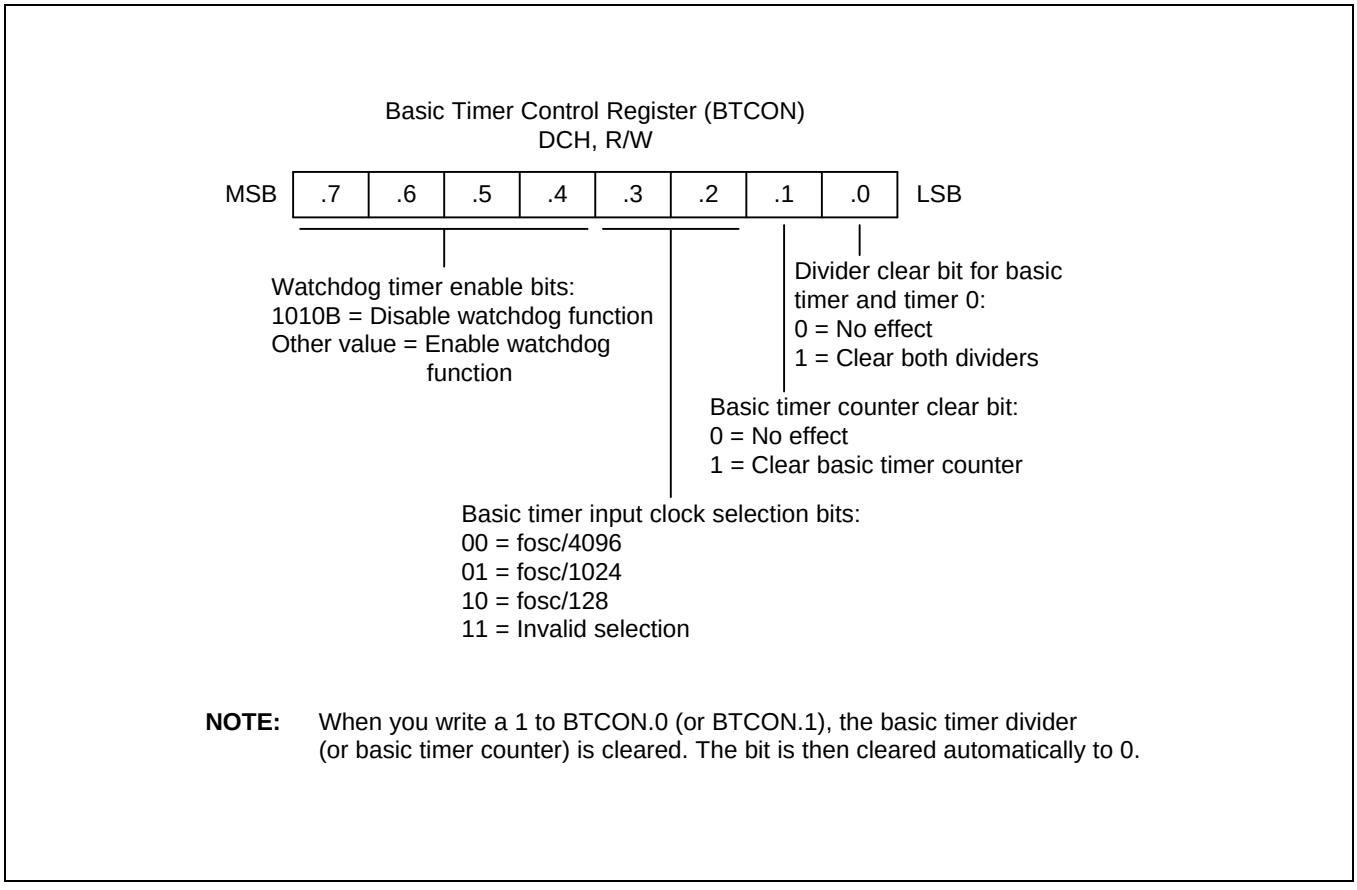
BASIC TIMER (BT)

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 f_{OSC}/4096. To disable the watchdog function, you must write the signature code "1010B" to the basic timer register control bits BTCON.7-BTCON.4.

The 8-bit basic timer counter, BTCNT, can be cleared 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.



NOTE: When you write a 1 to BTCON.0 (or BTCON.1), the basic timer divider (or basic timer counter) is cleared. The bit is then cleared automatically to 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 CCON register setting) divided by 4096 as the BT clock.

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

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

Oscillation Stabilization Interval Timer Function

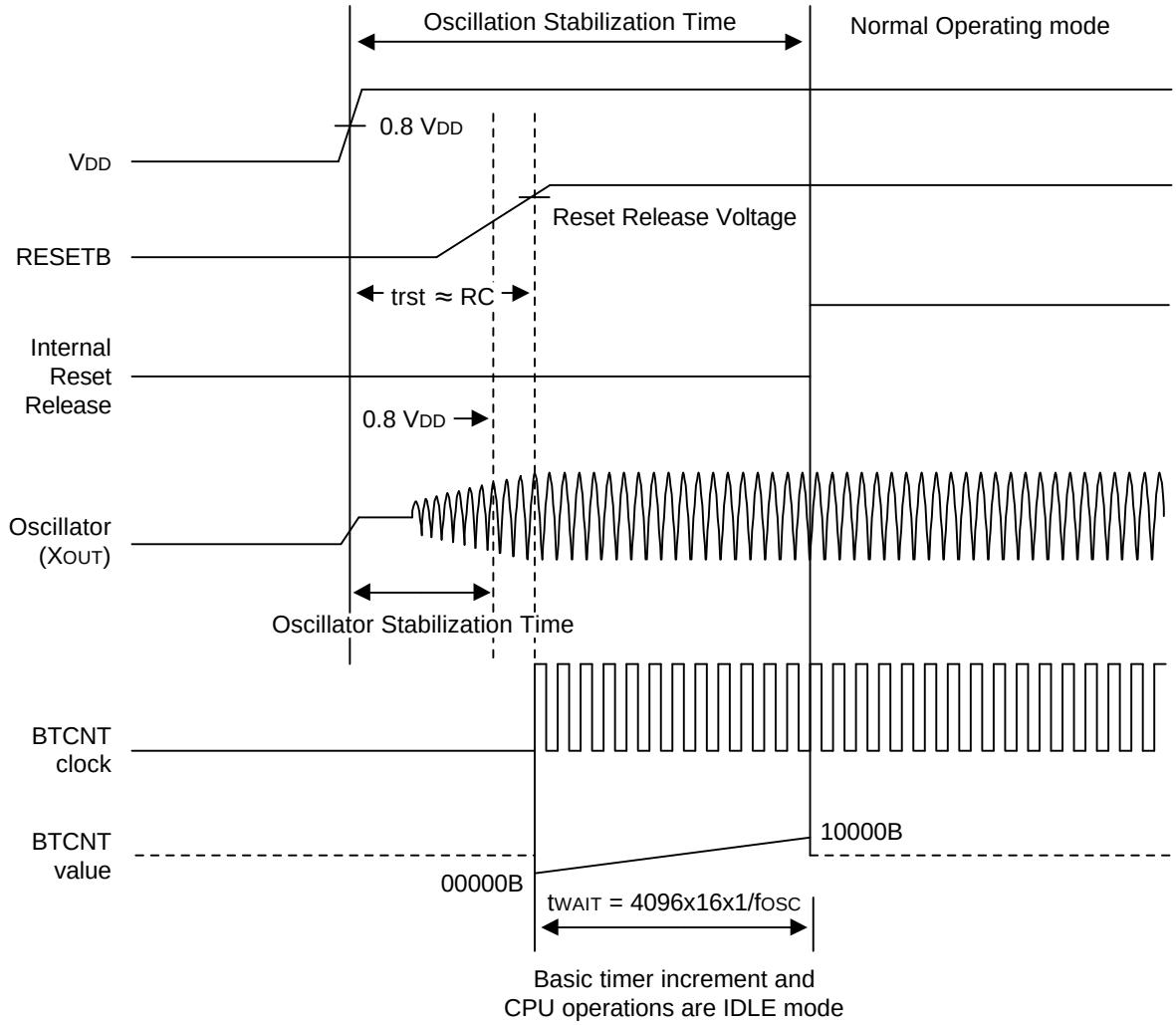
You can also use the basic timer to program a specific oscillation stabilization interval following a reset or when Stop mode has been released by an external interrupt.

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

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

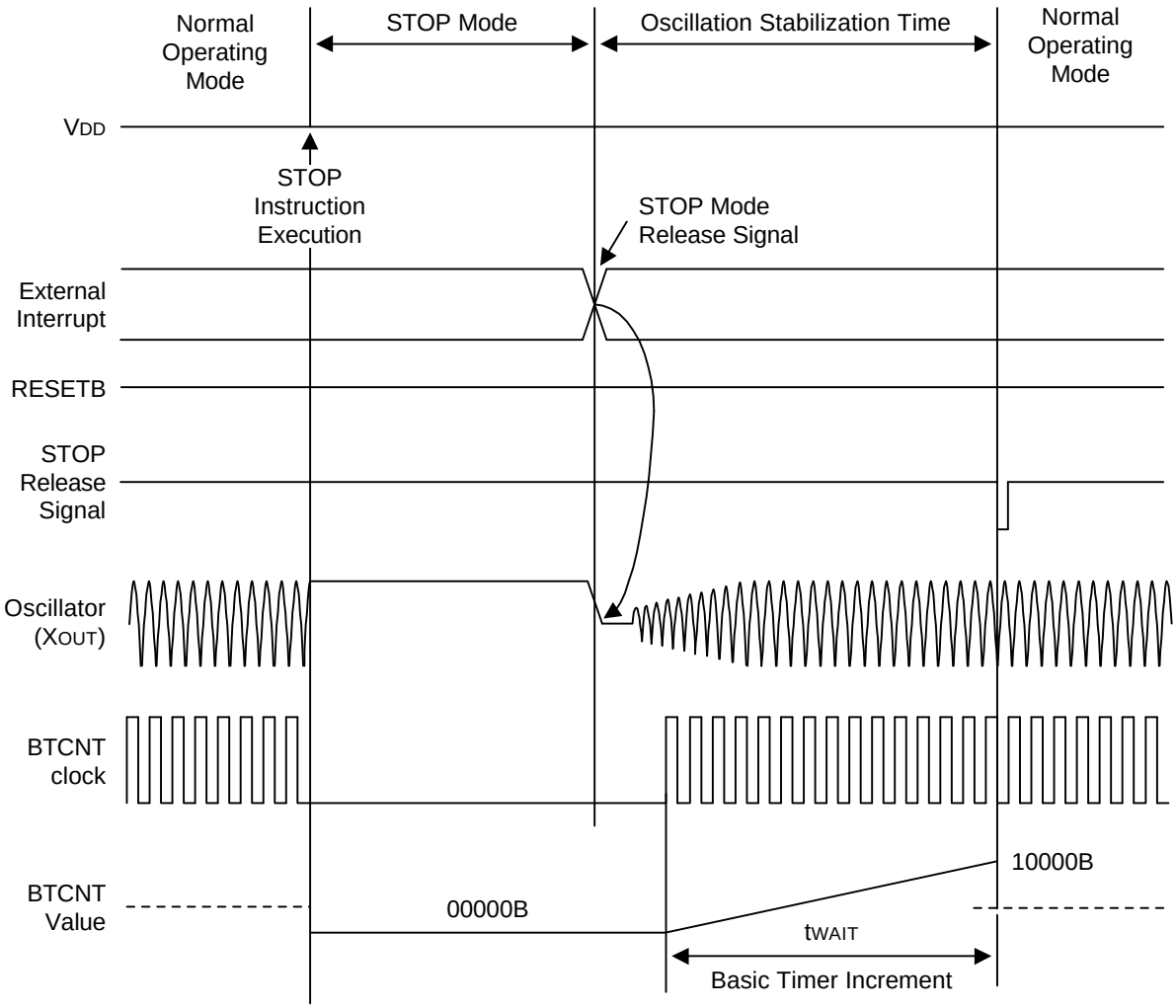
1. During Stop mode, a power-on reset or an external interrupt occurs to trigger the Stop mode release and oscillation starts.
2. If a power-on reset occurred, the basic timer counter will increase at the rate of $f_{OSC}/4096$. If an external interrupt is used to release Stop mode, the BTCNT value increases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

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



NOTE: Duration of the oscillator stabilization wait time, t_{WAIT}, when it is released by a Power-on-reset is 4096 x 16/f_{OSC}.
$t_{RST} \approx$ RC (R is external resistor and C is on chip capacitor)

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	1	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.

```
                ORG         0000H
                VECTOR      00H,INT_4304        ;  S3C9434 has only one interrupt vector

;--------------<< Initialize System and Peripherals >>


                ORG         0100H

RESET:          DI                              ;  Disable interrupt
                LD          CLKCON,#00011000B   ;  Select non-divided cpu clock
                LD          SP,#60H             ;  Stack pointer must be set
                •
                •
                LD          BTCON,#02H          ;  Enable watchdog function
                                                ;  Basic timer clock: Fosc/4096
                                                ;  Basic counter (BTCNT) clear
                •
                •
                •
                EI                              ;  Enable interrupt

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

MAIN:           •
                LD          BTCON,#02H          ;  Enable watchdog function
                                                ;  Basic counter (BTCNT) clear
                •
                •
                •
                JR          T,MAIN              ;

;--------------<< Interrupt Service Routines >>

INT_4304:       •                               ;  Interrupt enable bit and pending bit check
                •                               ;
                •                               ;  Pending bit clear
                IRET                            ;
                •
                •
                END                             ;
```

TIMER 0

TIMER 0 CONTROL REGISTERS (T0CON)

The timer 0 control register, T0CON, is used to select the timer 0 operating mode (interval timer) and input clock frequency, to clear the timer 0 counter, and to enable the T0 match interrupt. It also contains a pending bit for T0 match interrupts.

A reset clears T0CON to "00H". This sets timer 0 to normal interval timer mode, selects an input clock frequency of f_{OSC} /4096, and disables the T0 match interrupts. The T0 counter can be cleared at any time during normal operation by writing a "1" to T0CON.3.

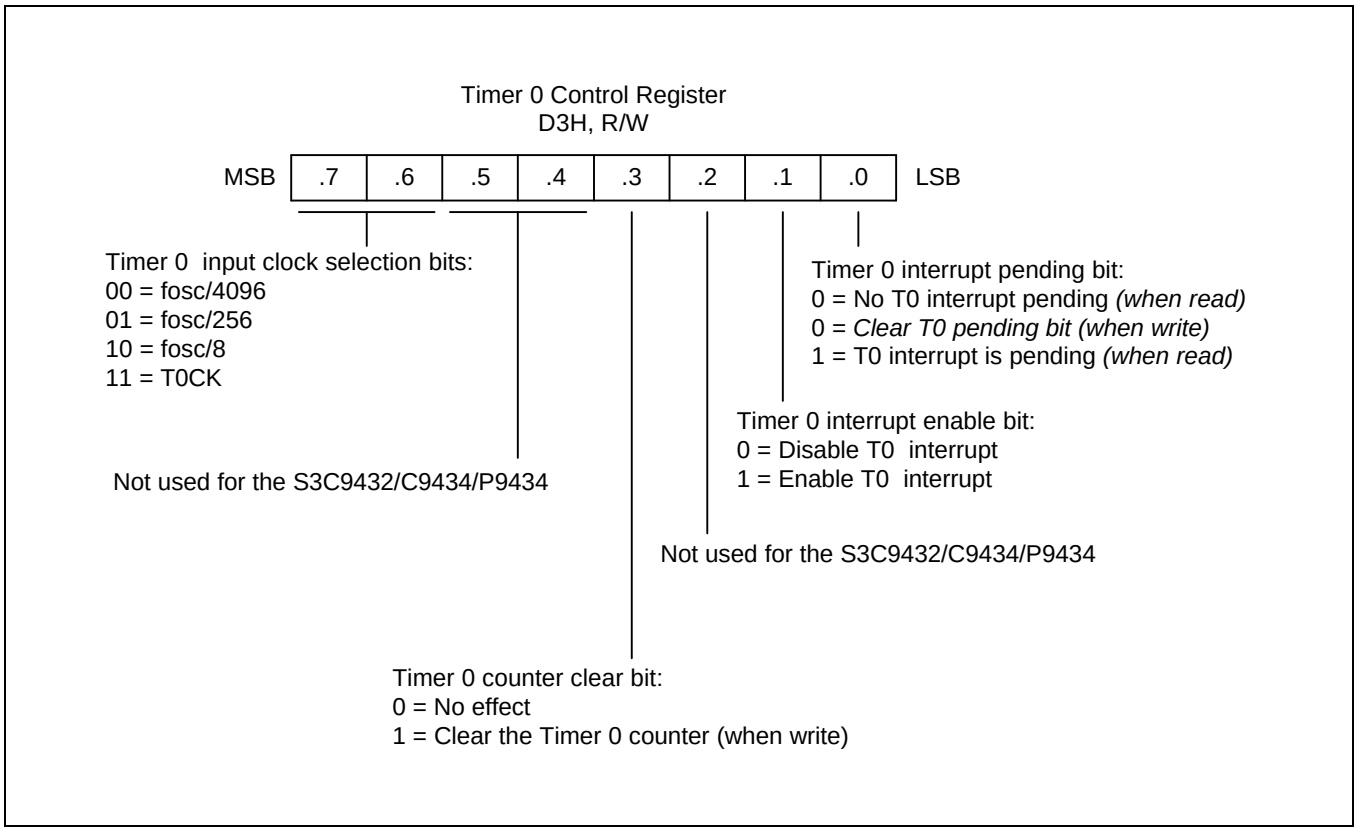


Figure 10-4. Timer 0 Control Registers (T0CON)

TIMER 0 FUNCTION DESCRIPTION

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.

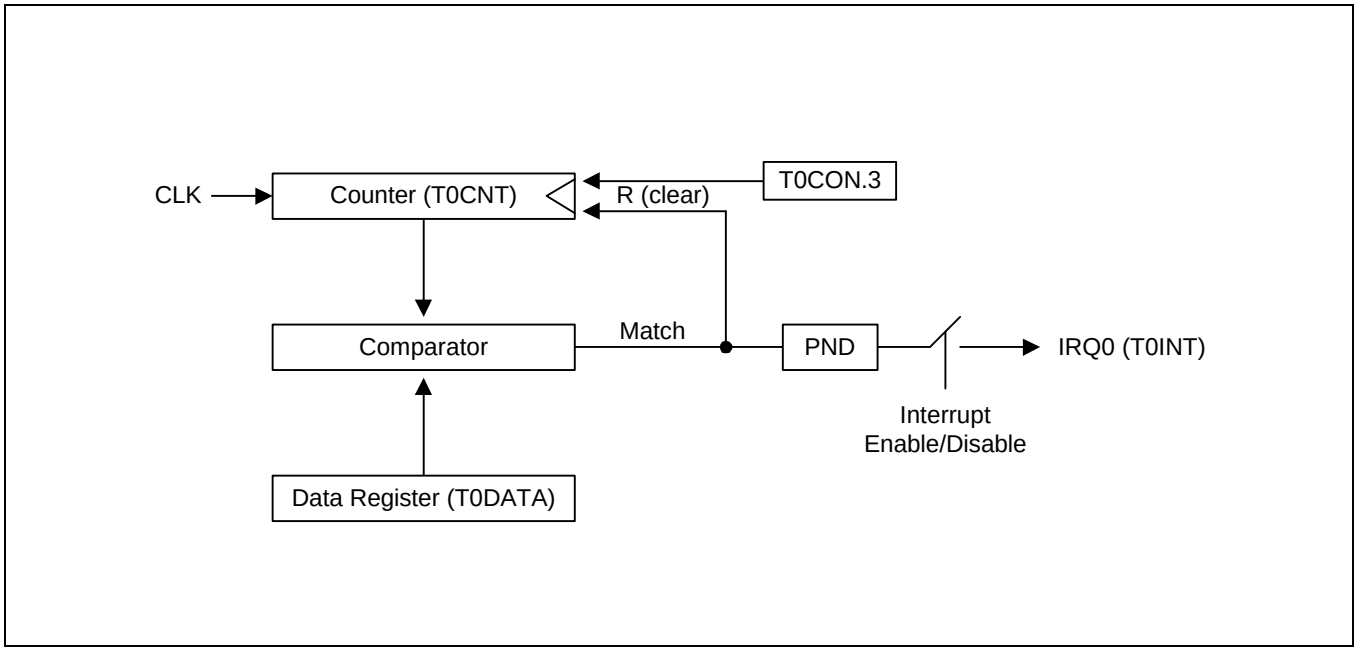


Figure 10-5. Simplified Timer 0 Function Diagram (Interval Timer Mode)

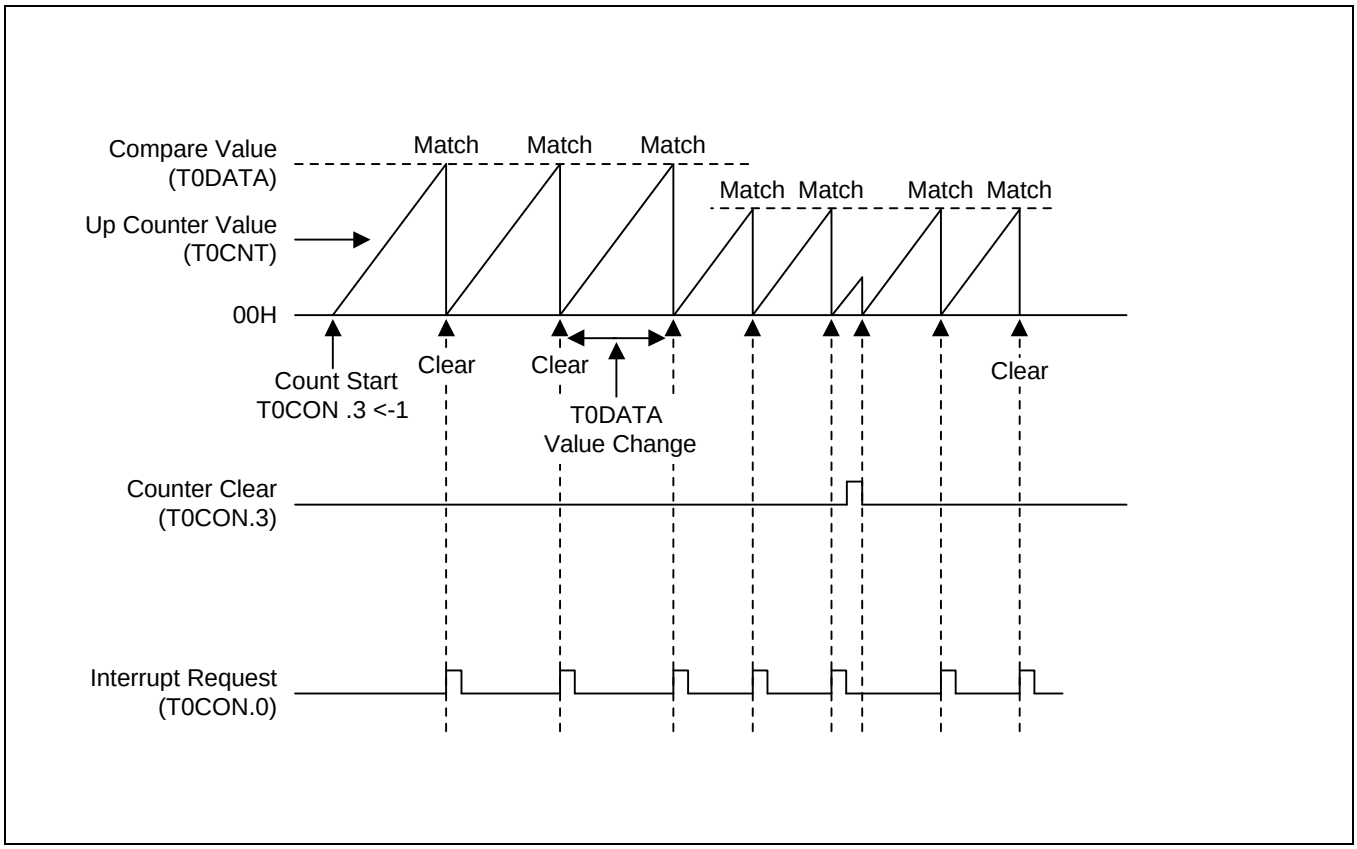
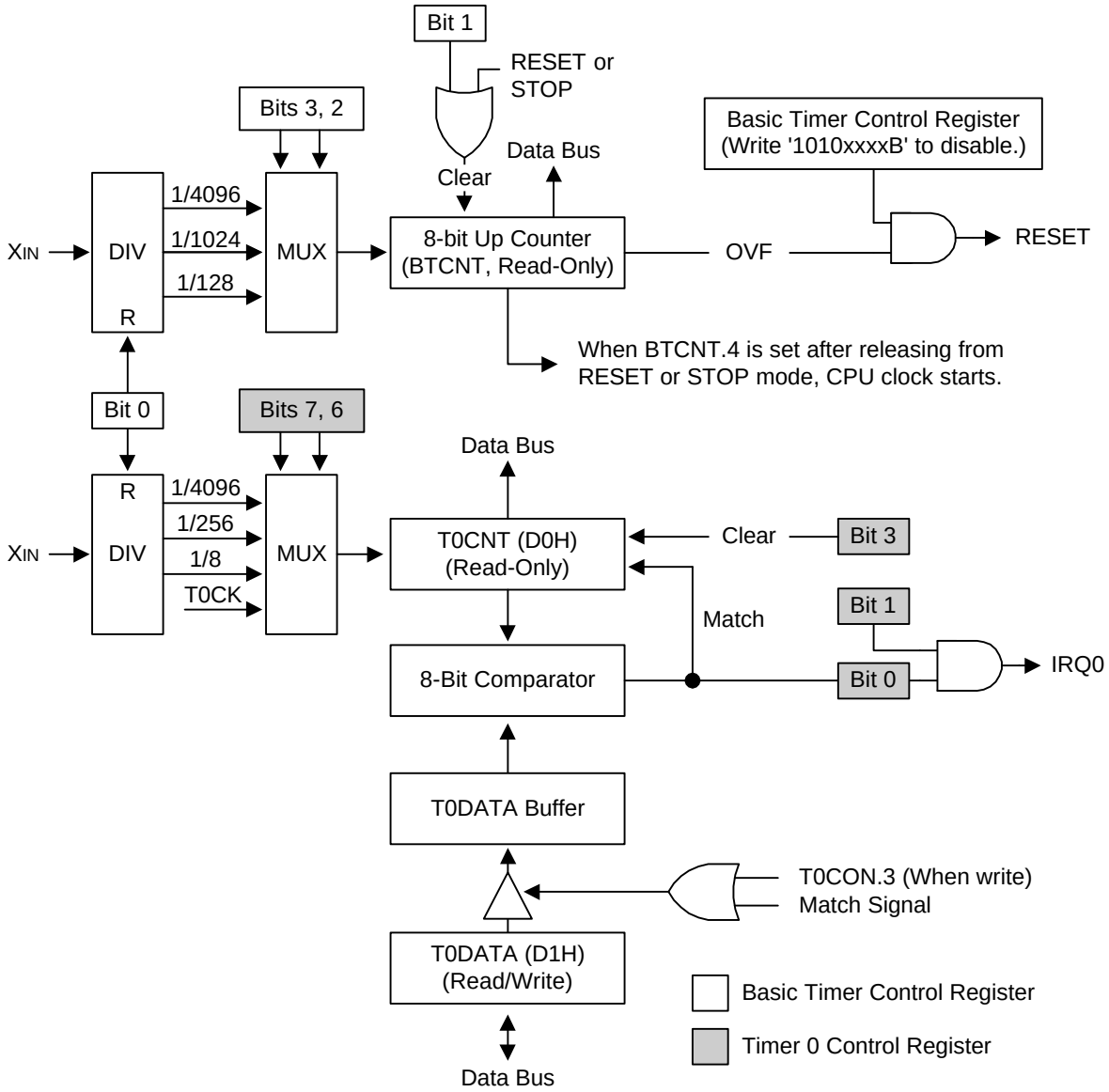


Figure 10-6. Timer 0 Timing Diagram



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 10-7. Basic Timer and Timer 0 Block Diagram

☞ PROGRAMMING TIP1 -- Configuring Timer 0 (Interval Mode)

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

```
                ORG         0000H
                VECTOR      00H,INT_4304        ;  S3C9434 has only one interrupt vector
                ORG         0100H
RESET:          DI                              ;  Disable interrupt
                LD          BTCON,#10100011B    ;  Watch-dog disable
                LD          CLKCON,#00011000B   ;  Select non-divided CPU clock
                LD          SP,#60H             ;  Set stack pointer
                LD          P0CON,#01011010B    ;  P0.3/0.2 input // P0.1/0.0 output
                LD          P0DPUR,#00001111B   ;  P0.0 - 0.3 pull-up enable
                LD          P1CONH,#00000100B   ;  P1.4 push-pull output
                LD          P1CONL,#00000000B   ;  P1.0 - P1.3 schmitt trigger input
                LD          P1DPUR,#00001111B   ;  P1.0 - P1.3 pull-up resistor enable
                LD          P2CON,#00000000B    ;  P2.0 - P2.3 push-pull output

;--------------<< Timer 0 settings >>

                LD          T0DATA,#50H         ;  CPU = 11.0592Mhz, interrupt interval = 2 msec
                LD          T0CON,#01001010B    ;  Fosc/256, Timer0 interrupt enable
                •
                •
                •
                EI                              ;  Enable interrupt

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

MAIN:           NOP                             ;  Start main loop
                •
                •
                •
                CALL        LED_DISPLAY         ;  Sub-block module
                •
                •
                •
                CALL        JOB                 ;  Sub-block module
                •
                •
                •
                JR          T,MAIN              ;
```

☞ PROGRAMMING TIP1 -- Configuring Timer 0 (Interval Mode) (Continued)

```
LED_DISPLAY:    NOP                             ;
                •                               ;
                •                               ;
                •                               ;
                RET                             ;

JOB:            NOP                             ;
                •                               ;
                •                               ;
                •                               ;
                RET                             ;

;--------------<< Interrupt Service Routines >>

INT_4304:       TM      T0CON,#00000010B        ;  Interrupt enable check
                JR      Z,NEXT_CHK1             ;

                TM      T0CON,#00000001B        ;  If timer0 interrupt was occurred,
                JP      NZ,INT_TIMER0           ;  T0CON.0 bit would be set.

NEXT_CHK1:      •                               ;  Interrupt enable bit and pending bit check
                •                               ;
                •                               ;
                IRET                            ;

INT_TIMER0:                                     ;  Timer0 interrupt service routine
                •
                •
                •
                AND     T0CON,#11110110B        ;   Pending bit clear
                IRET                            ;
                •
                •
                END                             ;
```

11 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 $f_{OSC}/256$, $f_{OSC}/64$, $f_{OSC}/8$, $f_{OSC}/1$.

FUNCTION DESCRIPTION

PWM

The 12-bit PWM circuits have the following components:

— 6-bit comparator and extension cycle circuit

— 6-bit reference data registers (PWMDATA)

— 6-bit extension data registers (PWMEX)

— PWM output pins (P0.1/PWM)

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 F2H, determine the output value generated by each 12-bit PWM circuit. These registers, PWM is read/write addressable.

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

— 8-bit extension registers PWMEX (F3H), 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 (PWMDATA) and the 6-bit extension registers (PWMEX). 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 f_{OSC} clock frequency. The counter clock value is determined by the setting of PWMCON.6-.7.

Table 11-1. PWM Control and Data Registers

Register Name	Mnemonic	Address	Function
PWM data registers	PWMDATA	F2H	6-bit PWM basic cycle frame value
	PWMEX	F3H	6-bit extension ("stretch") value
PWM control registers	PWMCON	F4H	PWM counter stop/start (resume), and F_{OSC} 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 (PWMDATA). If the value in the PWMDATA 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 register (PWMEX). 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 11-2).

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 11-2. 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

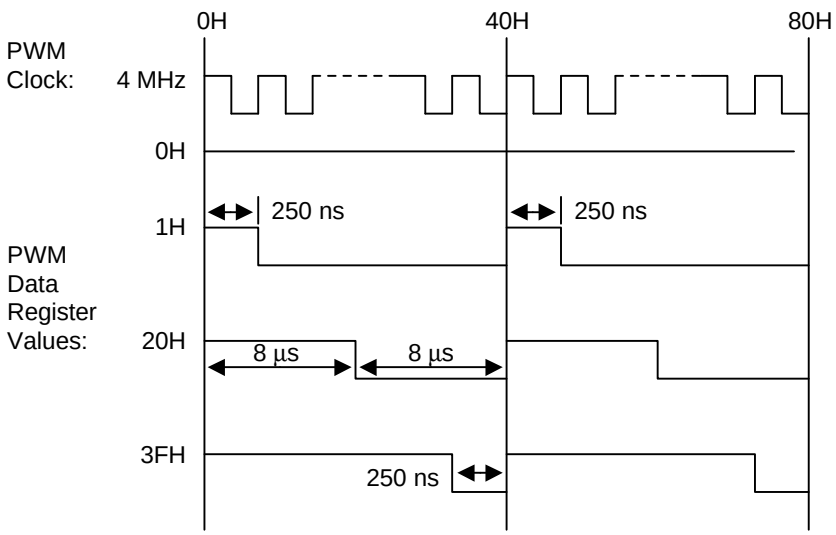


Figure 11-1. 12-Bit PWM Basic Waveform

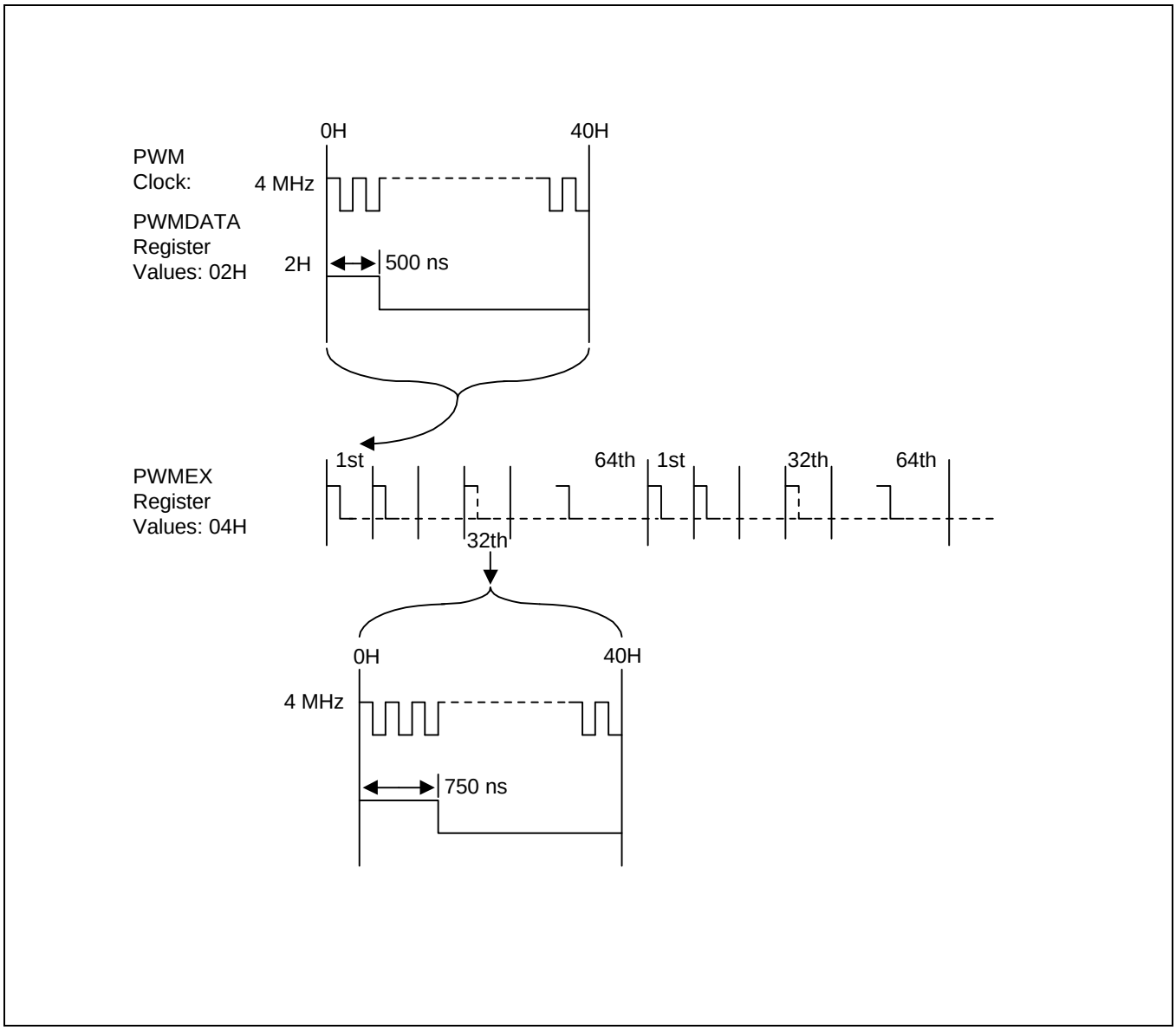


Figure 11-2. 12-Bit Extended PWM Waveform

PWM CONTROL REGISTER (PWMCON)

The control register for the PWM module, PWMCON, is located at register address F4H. 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.

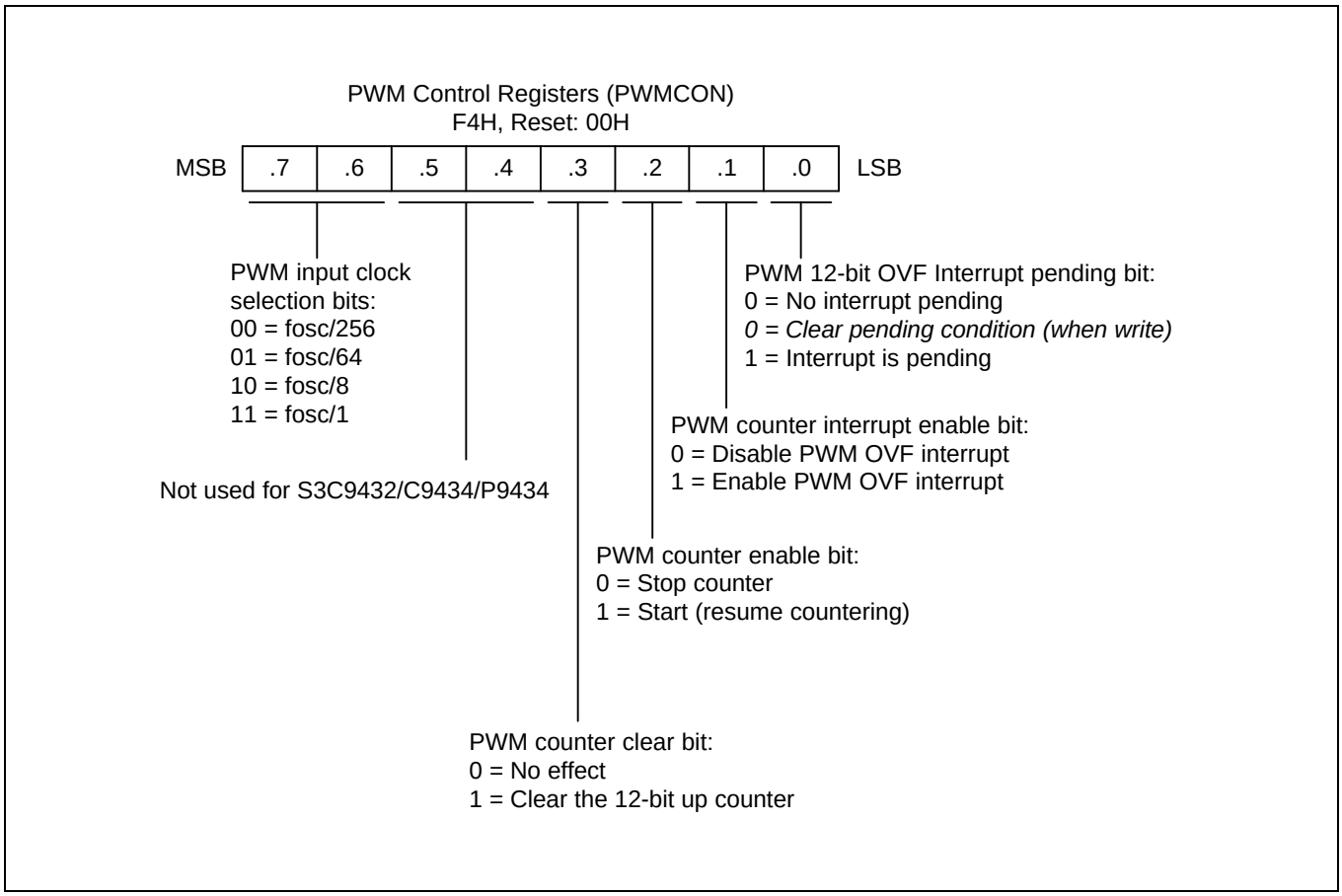


Figure 11-3. PWM/Capture Module Control Register (PWMCON)

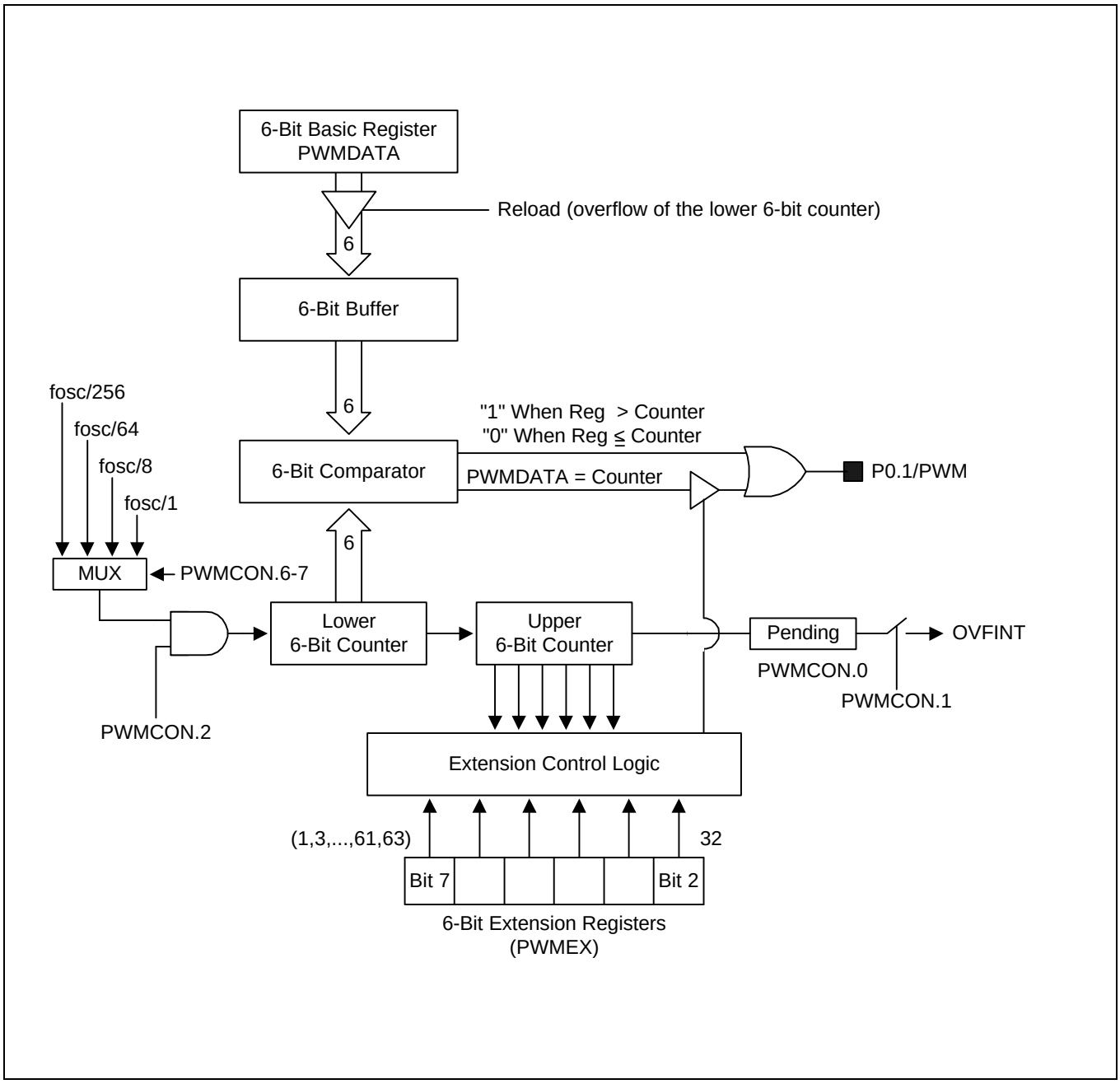


Figure 11-4. PWM/Capture Module Functional Block Diagram

☞ PROGRAMMING TIP — Programming the PWM Module to Sample Specifications

```
                ORG     0000H
                VECTOR  00H,INT_4304            ;  S3C9434 has only one interrupt vector

;--------------<< Initialize System and Peripherals >>

                ORG     0100H
RESET:          DI                              ;  disable interrupt
                LD      BTCON,#10100011B        ;  Watch-dog disable
                •
                •
                LD      P0CON,#00001100B        ;  Configure P0.1 PWM output
                LD      PWMCON,#00000110B       ;  Fosc/256, counter/interrupt enable
                LD      PWMDATA,#10H            ;  Set value between 00H - 3FH
                LD      PWMEX,#80H              ;  .1/.0 bit are not used
                •
                •
                EI                              ;  Enable interrupt

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

MAIN:                                           ;
                •                               ;
                •                               ;
                •                               ;
                •                               ;
                JR      t,MAIN                  ;

;--------------<< Interrupt Service Routines >>

INT_4304:       •                               ;  Interrupt enable bit and pending bit check
                TM      PWMCON,#00000010B       ;  Interrupt enable check
                JR      Z,NEXT_CHK1             ;
                TM      PWMCON,#00000001B       ;  Interrupt pending bit check
                JP      NZ,INT_PWM              ;  PWMCON's pending bit set --> PWM interrupt

NEXT_CHK1:      •
                •
                •
                IRET                            ;
```

☞ **PROGRAMMING TIP — Programming the PWM Module to Sample Specifications (Continued)**

```
INT_PWM:                                ;  PWM interrupt service routine
          •
          •
          •
          AND       PWMCON,#11110110B   ;  pending bit clear
          IRET                          ;
          •
          •
          END
```

12 A/D CONVERTER

OVERVIEW

The 10-bit A/D converter (ADC) module uses successive approximation logic to convert analog levels entering at one of the five 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:

— Analog comparator with successive approximation logic

— D/A converter logic

— ADC control register (ADCON)

— Five multiplexed analog data input pins (ADC0-ADC4)

— 10-bit A/D conversion data output register (ADDATAH/L):

— AV_{REF} and AV_{SS} pins

To initiate an analog-to-digital conversion procedure, you write the channel selection data in the A/D converter control register ADCON to select one of the five analog input pins (ADCn, n = 0-4) and set the conversion start or enable bit, ADCON.0. The read-write ADCON register is located at address F7H.

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 manipulating 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, ACON.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 ADC0-ADC4 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.

USING A/D PINS FOR STANDARD DIGITAL INPUT

The ADC module's input pins are alternatively used as digital input in port 1. The ADC0-ADC4 share pin names are P1.0-P1.4.

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 (ADC0-ADC4).

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

— Bits 2-1 select a conversion speed.

— 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 five analog input pins (ADC0-ADC4) by manipulating the 4-bit value for ADCON.7-ADCON.4.

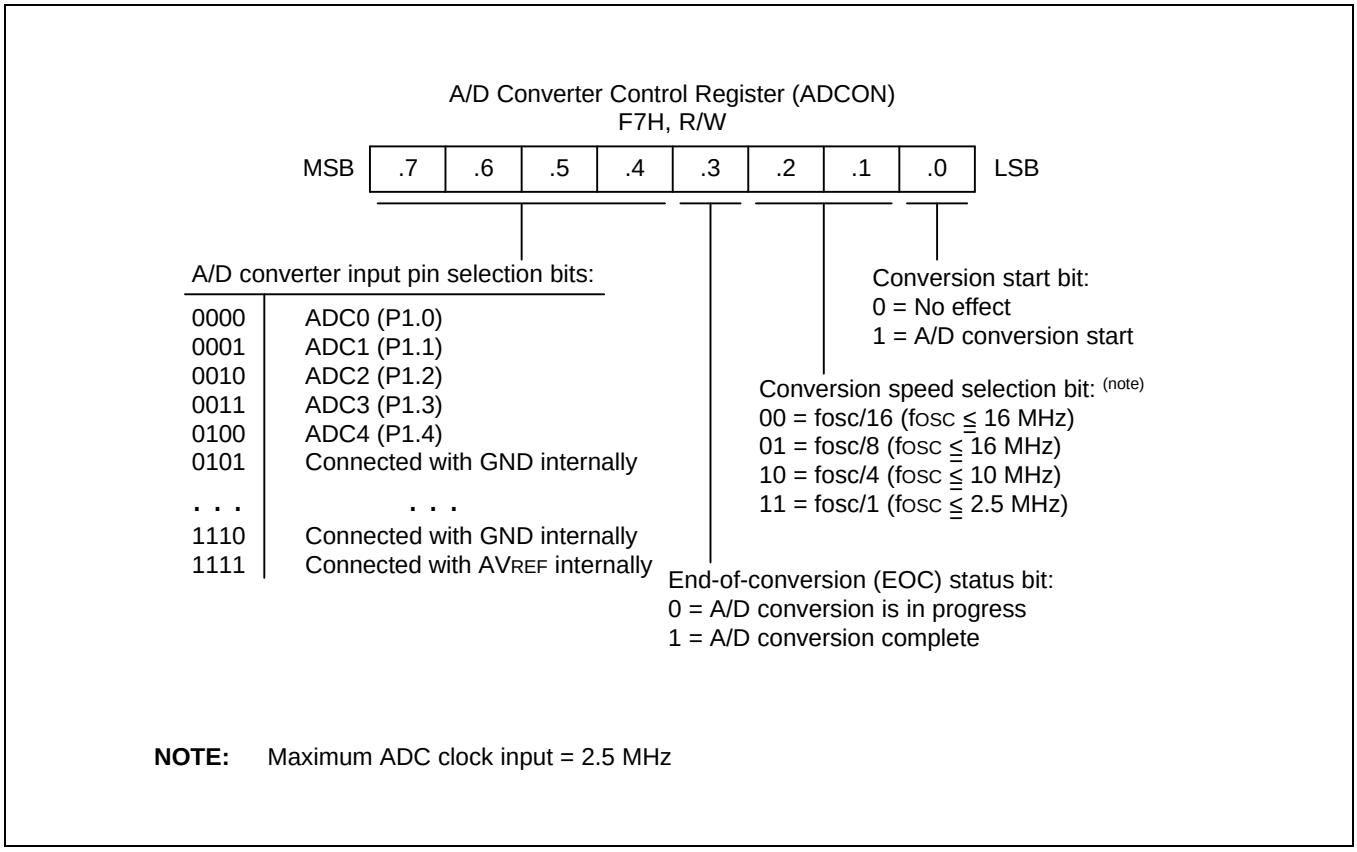


Figure 12-1. A/D Converter Control Register (ADCON)

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}.

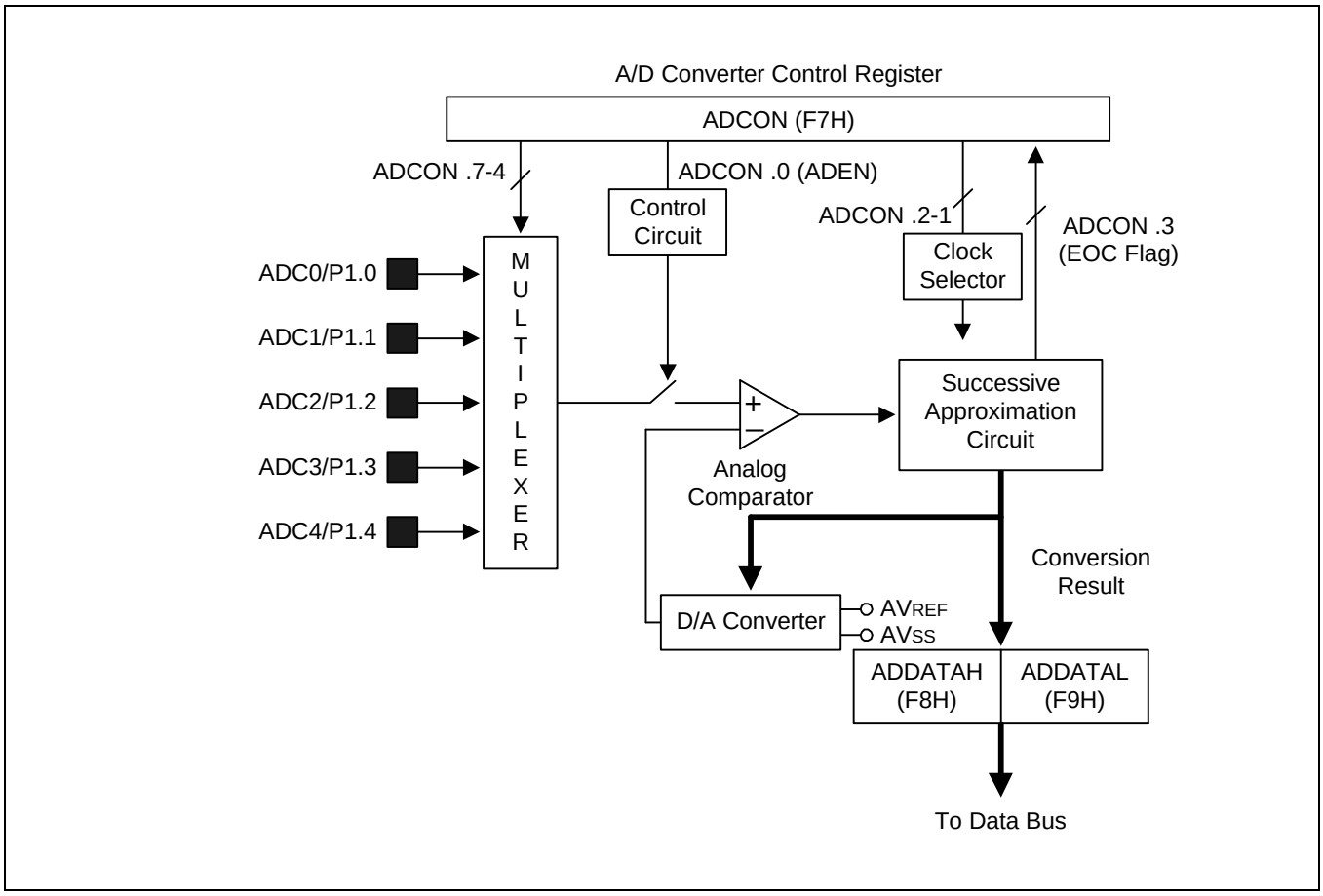


Figure 12-2. A/D Converter Circuit Diagram

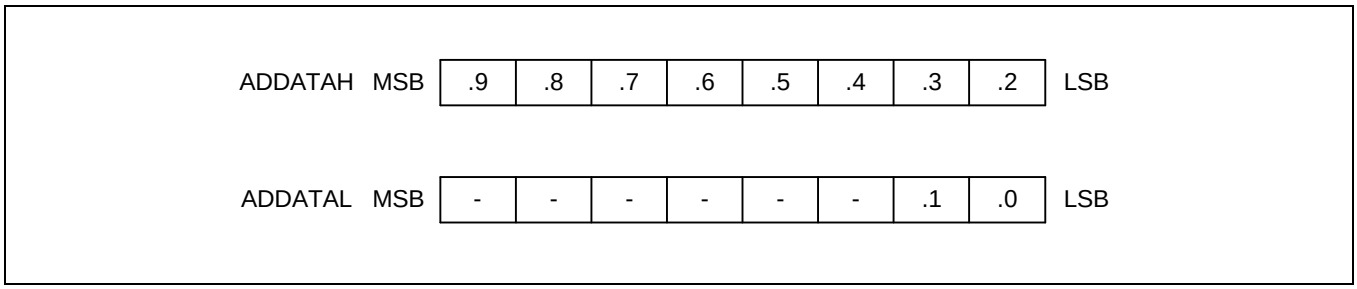


Figure 12-3. A/D Converter Data Register (ADDATAH/L)

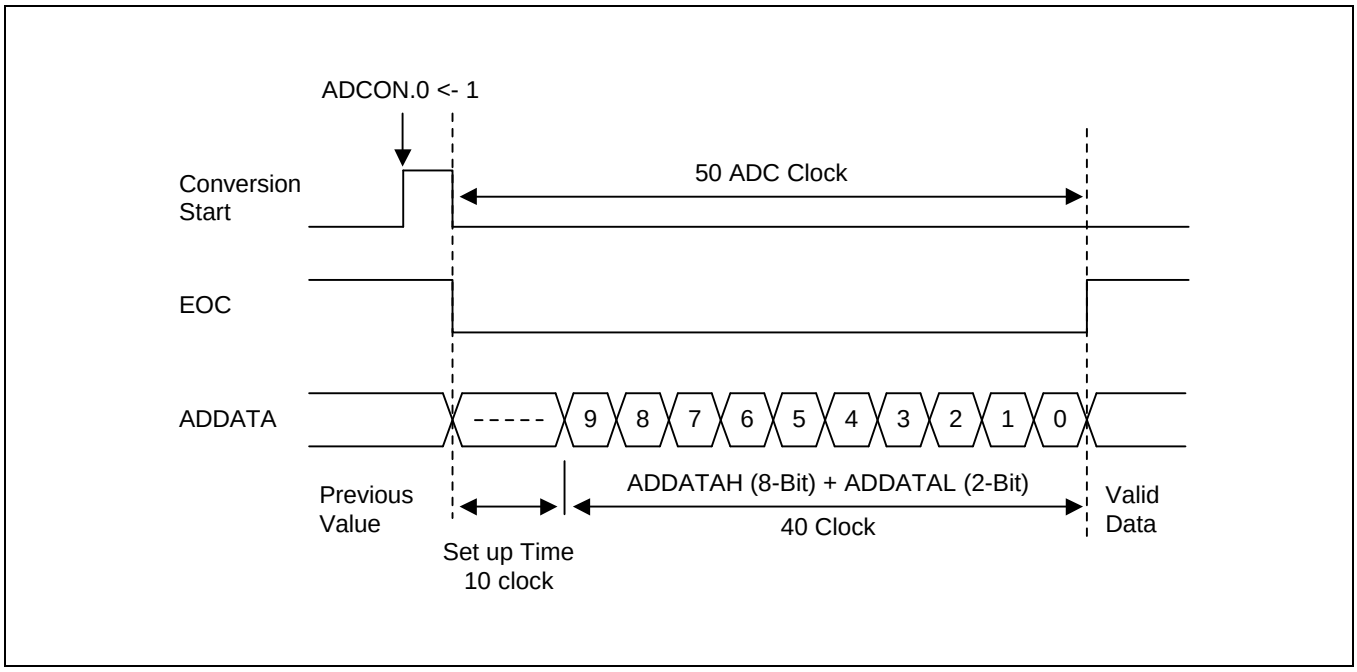


Figure 12-4. A/D Converter Timing Diagram

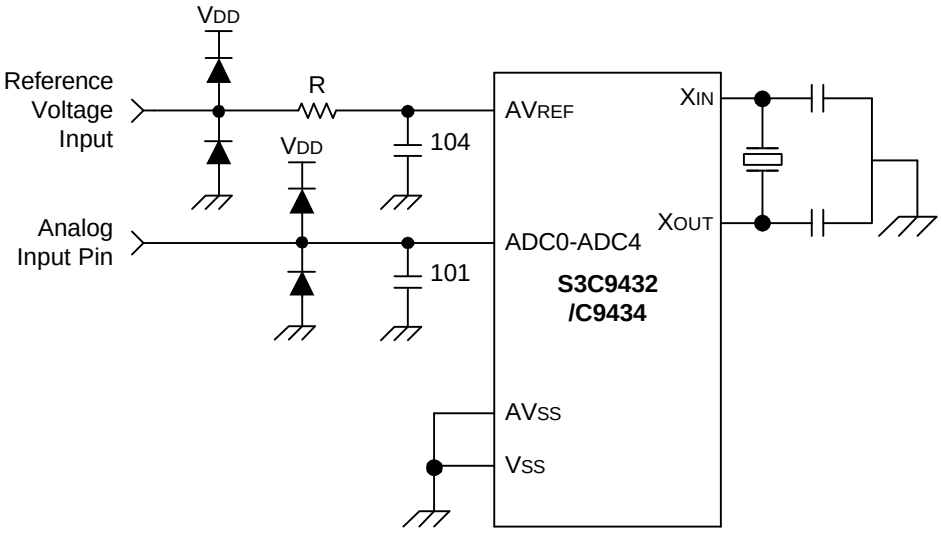
CONVERSION TIMING

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 10 MHz 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 + step-up time (10 clock) = 50 clocks
50 clock x 400 ns = 20 µs at 10 MHz, 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 P1CONH and P1CONL registers.
3. Before the conversion operation starts, you must first select one of the five input pins (ADC0-ADC4) by writing the appropriate value to the ADCON register.
4. When conversion has been completed, (50 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 (8-bit) and ADDATAL (2-bit), then the ADC module enters an idle state.
6. The digital conversion result can now be read from the ADDATAH and ADDATAL register.



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 12-5. Recommended A/D Converter Circuit for Highest Absolute Accuracy

☞ PROGRAMMING TIP– Configuring A/D Converter

```
                ORG     0000H
                VECTOR  00H,INT_4304        ;  S3C9434 has only one interrupt vector

                ORG     0100H
RESET:          DI                          ;  disable interrupt
                LD      BTCON,#10100011B    ;  Watch-dog disable
                •
                •
                •
                LD      P1CONH,#00000011B   ;  Configure P1.4 AD input
                LD      P1CONL,#11111111B   ;  Configure P1.0-1.3 AD input
                •
                •
                EI                          ;  Enable interrupt

;--------------<< Main loop >>
MAIN:           •
                •
                •
                CALL    AD_CONV             ;  Subroutine for AD conversion
                •
                •
                •
                JR      t,MAIN              ;

AD_CONV:        LD      ADCON,#00000001B    ;  Select analog input channel --> P1.0
                                            ;  Select conversion speed --> Fosc/16
                                            ;  Set coversion start bit

CONV_LOOP:      TM      ADCON,#00001000B    ;  Check EOC flag
                JR      Z,CONV_LOOP         ;  If EOC flag=0, jump to CONV_LOOP until EOC flag=1
                LD      R0,ADDATAH          ;  High 8 bits of conversion result are stored
                                            ;  to addatah register
                LD      R1,ADDATAL          ;  Low 2 bits of conversion result are stored
                                            ;  to addatal register
                •
                •
                •
                RET                         ;
INT_4304:       •                           ;  Interrupt enable bit and pending bit check
                •                           ;
                •                           ;  Pending bit clear
                IRET                        ;
                •
                •
                END
```

13 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 3 (SO, SCK, SI) by loading the appropriate value to the P1CONL and P2CON Register if necessary.
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.

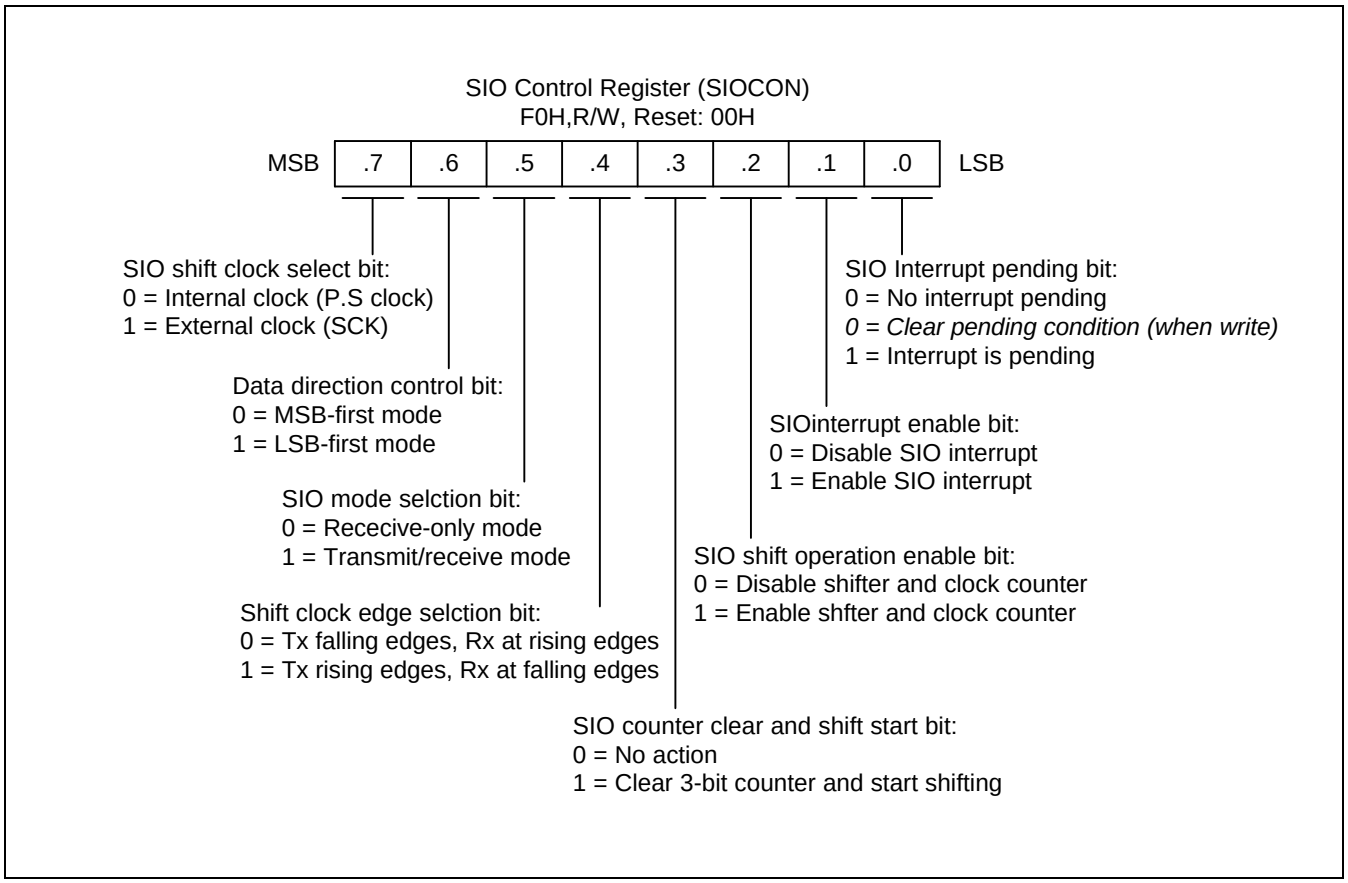


Figure 13-1. Serial I/O Interface Control Register (SIOCON)

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

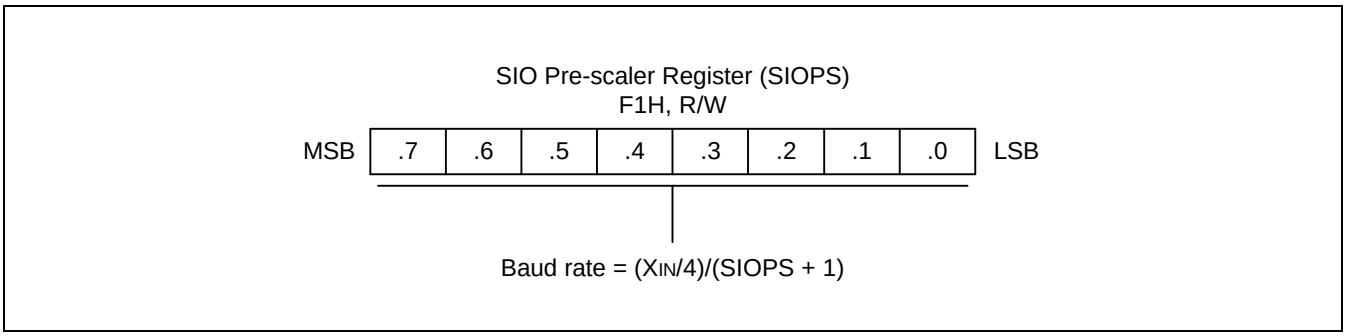


Figure 13-2. SIO Pre-scaler Register (SIOPS)

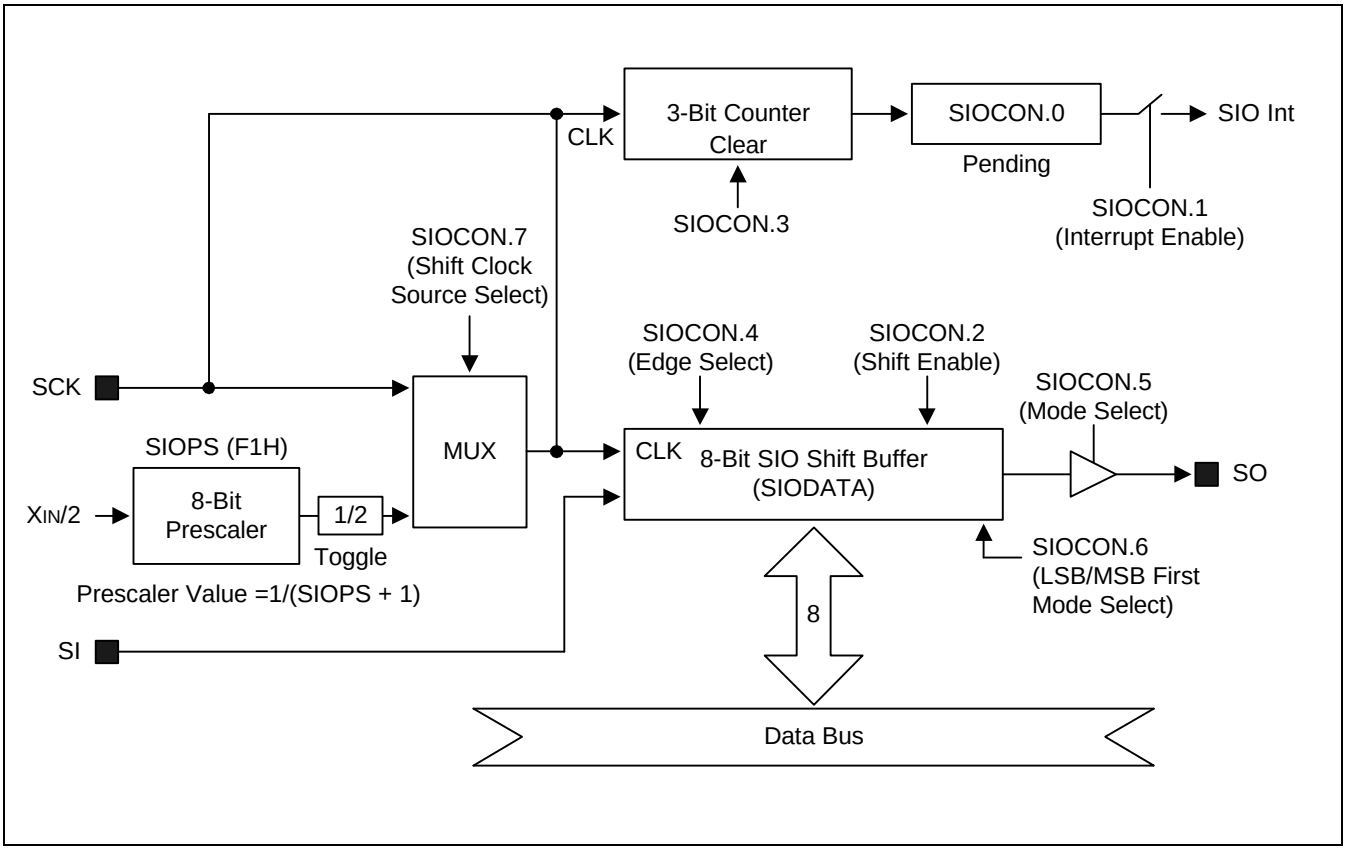


Figure 13-3. SIO Functional Block Diagram

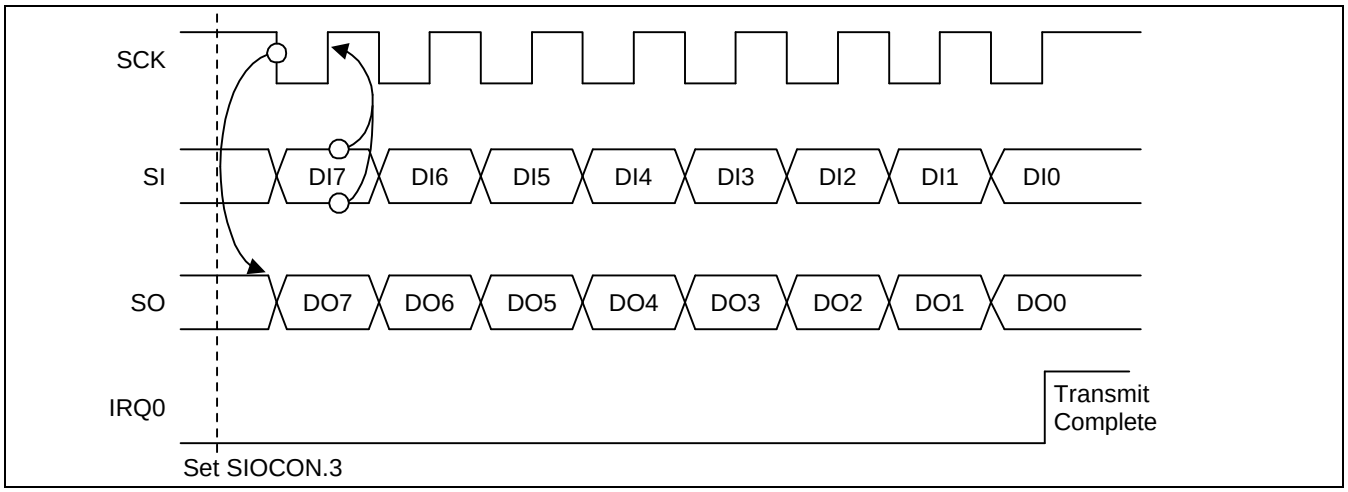


Figure 13-4. Serial I/O Timing in Transmit-Receive Mode (Tx at falling, SIOCON.4 = 0)

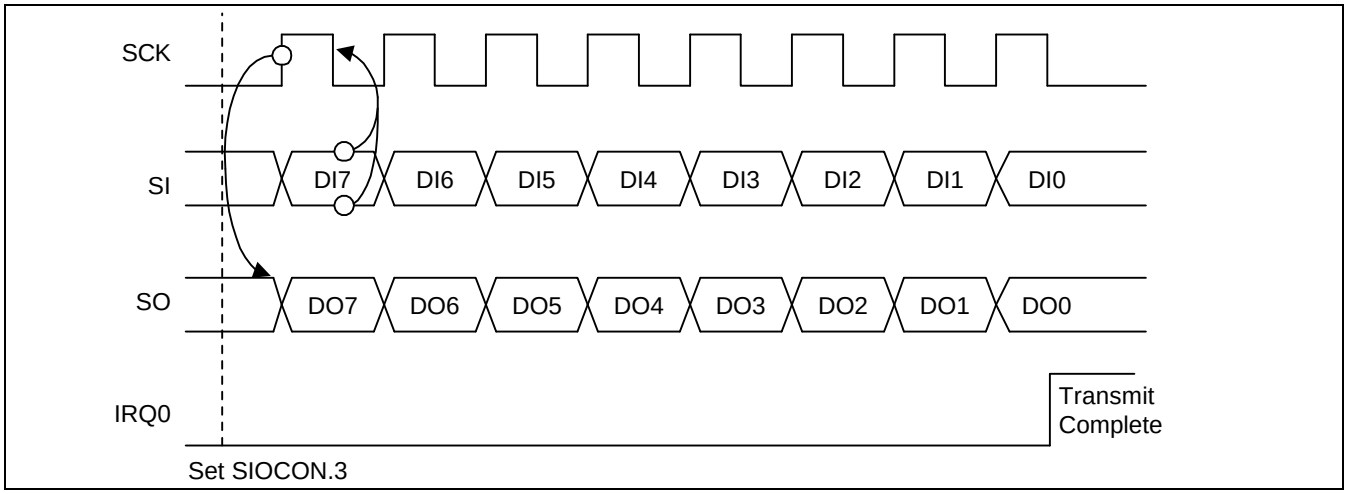


Figure 13-5. Serial I/O Timing in Transmit-Receive Mode (Tx at rising, SIOCON.4 = 1)

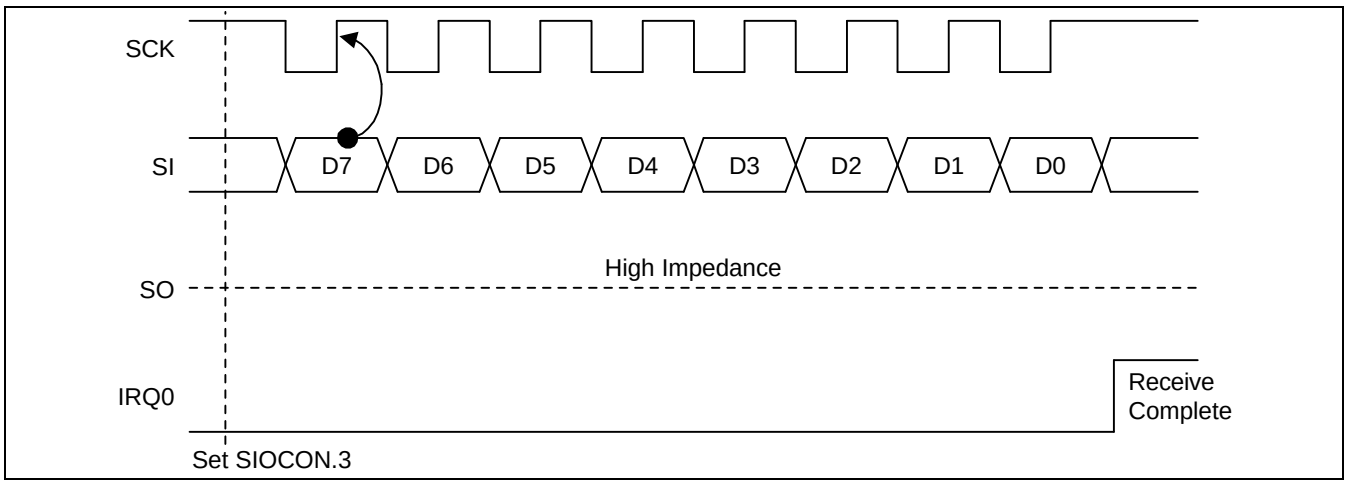


Figure 13-6. Serial I/O Timing in Receive-Only Mode

☞ PROGRAMMING TIP — SIO

```
                ORG     0000H
                VECTOR  00H,INT_4304        ;  S3C9434 has only one interrupt vector

SIO_RX_BUF      EQU     40H
SIO_TX_BUF      EQU     41H

;--------------<< Initialize System and Peripherals >>

                ORG     0100H
RESET:          DI                          ;  disable interrupt
                LD      BTCON,#10100011B    ;  Watch-dog disable
                •
                •
                LD      P2CON,#00000101B    ;  Configure SO (P2.1), SCK (P2.0)
                LD      P1CONL,#00100000B   ;  Configure SI (P1.1)
                LD      SIOCON,#00100110B   ;  Enable SIO, transmit/receive mode
                LD      SIOPS,#9            ;  Baud rate = (input clock)/40
                •
                •
                EI                          ;  Enable interrupt

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

MAIN:           •
                •
                CALL    SIO_MODE            ;  To transmit data
                •
                •
                JR      t,MAIN              ;

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

SIO_MODE:       •
                •
                LD      SIODATA,SIO_TX_BUF  ;  Load transmission data
                OR      SIOCON,#00001000B   ;  After data loading SIOCON.3 bit must be set
                •
                •
                RET                         ;
```

☞ PROGRAMMING TIP — SIO (Continued)

```
;--------------<< Interrupt Service Routines >>

INT_4304:   •                                ;  Interrupt enable bit and pending bit check
            TM      SIOCON,#00000010B        ;  Interrupt enable bit check
            JR      Z,NEXT_CHK1              ;
            TM      SIOCON,#00000001B        ;  Interrupt pending bit check
            JP      NZ,INT_SIO               ;  SIO interruopt

NEXT_CHK1:  •
            •
            •
            IRET                             ;

INT_SIO:                                     ;  SIO interrupt service routine
            •
            •
            •
            LD      SIO_RX_BUF,SIODATA       ;  Save received data
            AND     SIOCON,#11110110B        ;  Pending bit clear
            IRET                             ;  Interrupt return
            •
            •
            END                              ;
```

14 ELECTRICAL DATA

OVERVIEW

In this section, the following S3C9432/C9434 electrical characteristics are presented in tables and graphs:

— Absolute maximum ratings
— D.C. electrical characteristics
— A.C. electrical characteristics
— Input Timing Measurement Points
— Oscillator characteristics
— Oscillation stabilization time
— Operating Voltage Range
— Schmitt trigger input characteristics
— Data retention supply voltage in Stop mode
— Stop mode release timing when initiated by a RESET
— A/D converter electrical characteristics
— LVD circuit characteristics
— LVD reset Timing
— Serial I/O timing characteristics
— Serial data transfer timing

Table 14-1. 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
		All I/O pins active	+ 150	
Operating temperature	T_A	–	- 40 to + 85	°C
Storage temperature	T_{STG}	–	- 65 to + 150	°C

Table 14-2. DC Electrical Characteristics

(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, and RESET	V_{DD}= 3.0 to 5.5 V	0.8 V_{DD}	–	V_{DD}	V
	V_{IH2}	X_{IN} and X_{OUT}		V_{DD} - 0.1			
Input low voltage	V_{IL1}	Ports 0, 1, 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, 1, 2	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, 1, and 2	V_{DD}= 4.5 to 5.5 V	–	0.4	2.0	V

Table 14-2. DC Electrical Characteristics (Continued)

(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 leakage current	I_{LIH1}	All inputs except I_{LIH2}	$V_{IN} = V_{DD}$	–	–	1	uA
	I_{LIH2}	X_{IN}, X_{OUT}	$V_{IN} = V_{DD}$			20	
Input low leakage current	I_{LIL1}	All inputs except I_{LIL2} and RESET	V_{IN} = 0 V	–	–	-1	uA
	I_{LIL2}	X_{IN}, X_{OUT}	V_{IN} = 0 V			-20	
Output high leakage current	I_{LOH}	All outputs	$V_{OUT} = V_{DD}$	–	–	2	uA
Output low leakage current	I_{LOL}	All outputs	V_{OUT} = 0 V	–	–	-2	uA
Pull-up resistors	R_P	V_{IN} = 0 V Ports 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} = 5V ± 10%	–	11	20	mA
		8 MHz CPU clock	V_{DD} = 3.3 V		3	6	
	I_{DD2}	Idle mode 16 MHz CPU clock	V_{DD} = 5V ± 10%	–	5	8	
		8 MHz CPU clock	V_{DD} = 3.3 V		0.7	2.5	
	I_{DD3}	Stop mode	V_{DD} = 5V ± 10%	–	65	100	uA
			V_{DD} = 3.3 V		45	80	

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

Table 14-3. AC Electrical Characteristics

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

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

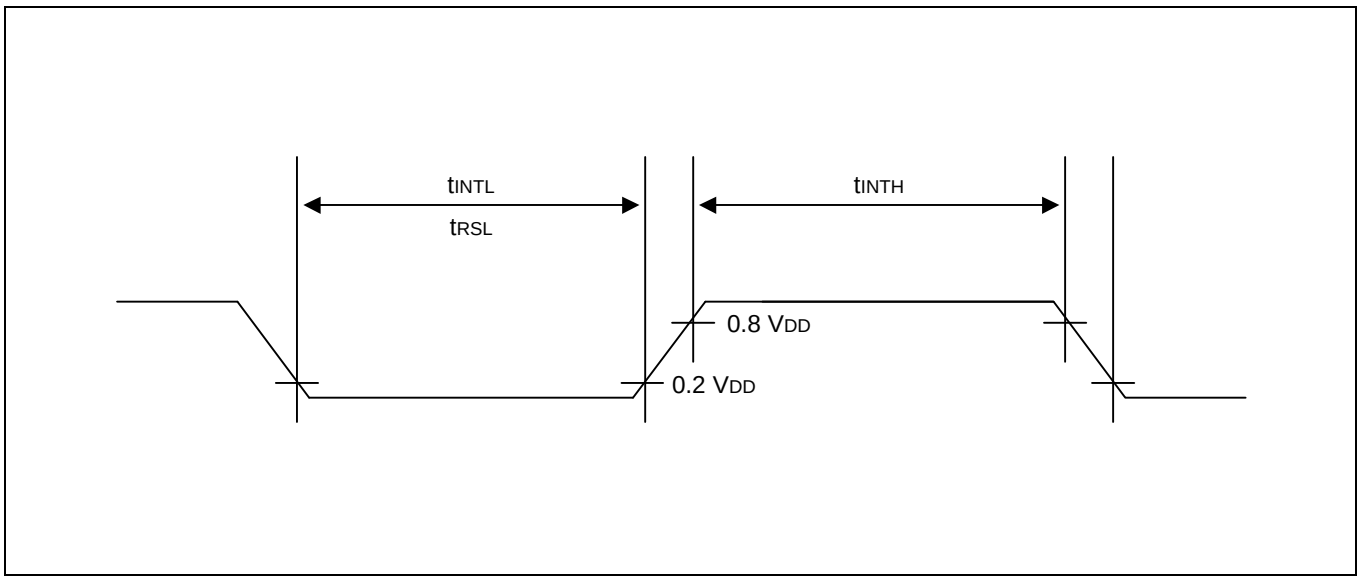


Figure 14-1. Input Timing Measurement Points

Table 14-4. Oscillator Characteristics

(T_A = - 40°C to + 85°C)

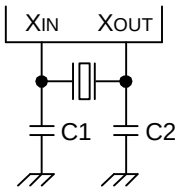
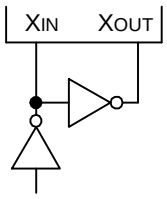
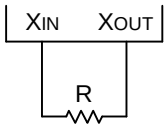
Oscillator	Clock Circuit	Test Condition	Min	Typ	Max	Unit
Main crystal or ceramic		V_{DD} = 4.5 to 5.5 V V_{DD} = 3.0 to 4.5 V	1 1	– –	16 8	MHz
External clock		V_{DD} = 4.5 to 5.5 V V_{DD} = 3.0 to 4.5 V	1 1	– –	16 8	
RC oscillator		V_{DD} = 5 V, R = 10 KΩ V_{DD} = 3 V, R = 22 KΩ	– –	4 2	– –	

Table 14-5. Oscillation Stabilization Time

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

Oscillator	Test Condition	Min	Typ	Max	Unit
Main crystal	f_{OSC} > 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)	–	$2^{16}/f_{OSC}$	–	ms
	t_{WAIT} when released by an interrupt (2)	–	–	–	

NOTES:

1. f_{OSC} 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 settings in the basic timer control register, BTCON.

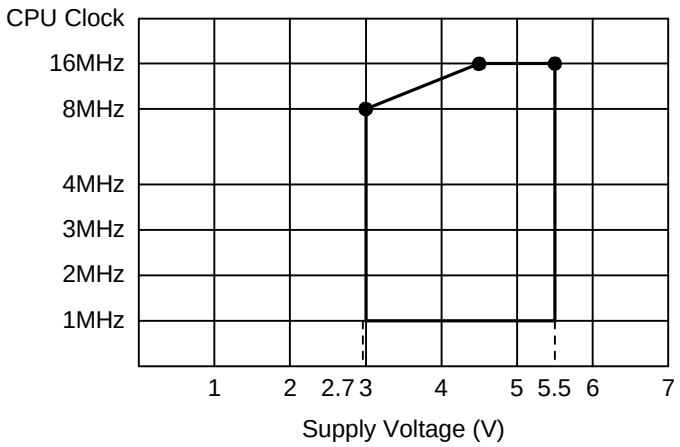


Figure 14-2. Operating Voltage Range

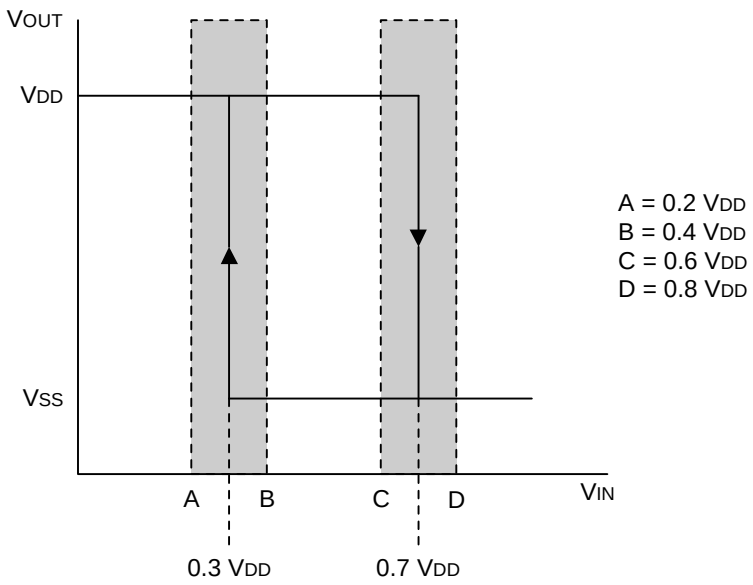


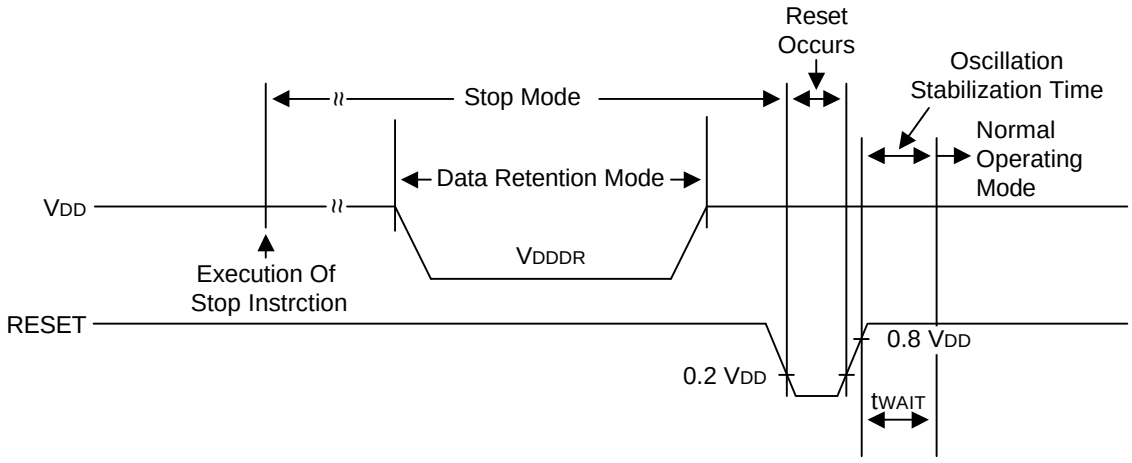
Figure 14-3. Schmitt Trigger Input Characteristics Diagram

Table 14-6. Data Retention Supply Voltage in Stop Mode

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

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Data retention supply voltage	V_{DDDR}	Stop mode	2.0	–	5.5	V
Data retention supply current	I_{DDDR}	Stop mode; V_{DDDR} = 2.0 V	–	0.1	5	uA

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



NOTE: tWAIT is the same as 4096 x 16 x 1/fosc

Figure 14-4. Stop Mode Release Timing When Initiated by a RESET

Table 14-7. A/D Converter Electrical Characteristics

(T_A = - 40°C to + 85°C, V_{DD} = 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	
Differential linearity error	DLE	–	–	–	± 1	
Offset error of top	EOT	–	–	± 1	± 3	
Offset error of bottom	EOB	–	–	± 1	± 2	
Conversion time (1)	t_{CON}	f_{OSC} = 10 MHz	–	50x4/ f_{OSC}	–	µs
Analog input voltage	V_{IAN}	–	AV_{SS}	–	AV_{REF}	V
Analog input impedance	R_{AN}	–	2	–	–	MΩ
ADC reference voltage	AV_{REF}	–	3.0	–	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
Analog block current (2)	I_{ADC}	AV_{REF} = V_{DD} = 5 V conversion time = 20 µs		1	3	mA
		AV_{REF} = V_{DD} = 3 V conversion time = 20 µs		0.5	1.5	mA
		AV_{REF} = V_{DD} = 5 V when 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.

Table 14-8. LVD Circuit Characteristics

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

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		(note)	us
Power supply voltage off time	t_{off}		0.5			sec
Power-on reset circuit consumption current	I_{DDPR}	V_{DD} = 5 V ± 10 %		65	100	uA
		V_{DD} = 3 V		45	80	uA

NOTE: Oscillation stabilization time = 2^{16}/fx (= 6.55 ms at fx = 10 MHz)

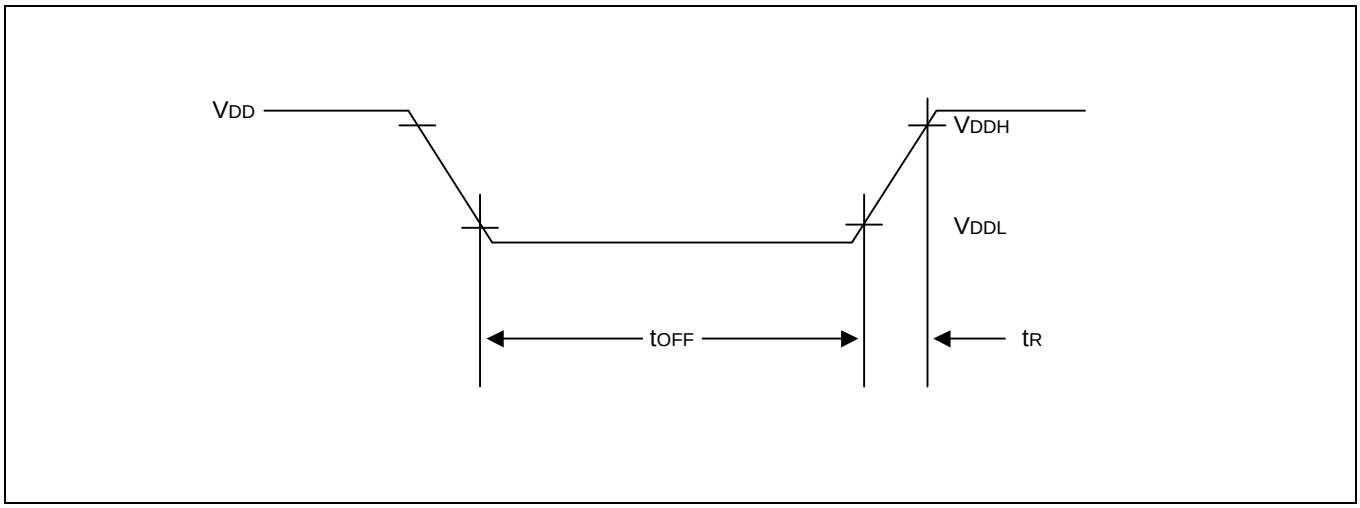


Figure 14-5. LVD Reset Timing

Table 14-9. Serial I/O Timing Characteristics

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

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
SCK Cycle Time	t_{CKY}	External SCK source	1000	–	–	ns
		Internal SCK source	1000			
SCK High, Low Width	t_{KH}, t_{KL}	External SCK source	500	–	–	
		Internal SCK source	t_{KCY}/2 – 50			
SI Setup Time to SCK Low	t_{SIK}	External SCK source	250	–	–	
		Internal SCK source	250			
SI Hold Time to SCK High	t_{KSI}	External SCK source	400	–	–	
		Internal SCK source	400			
Output Delay for SCK to SO	t_{KSO}	External SCK source	–	–	300	
		Internal SCK source			250	

NOTE: "SCK" means serial I/O clock frequency, "SI" means serial data input, and "SO" means serial data output.

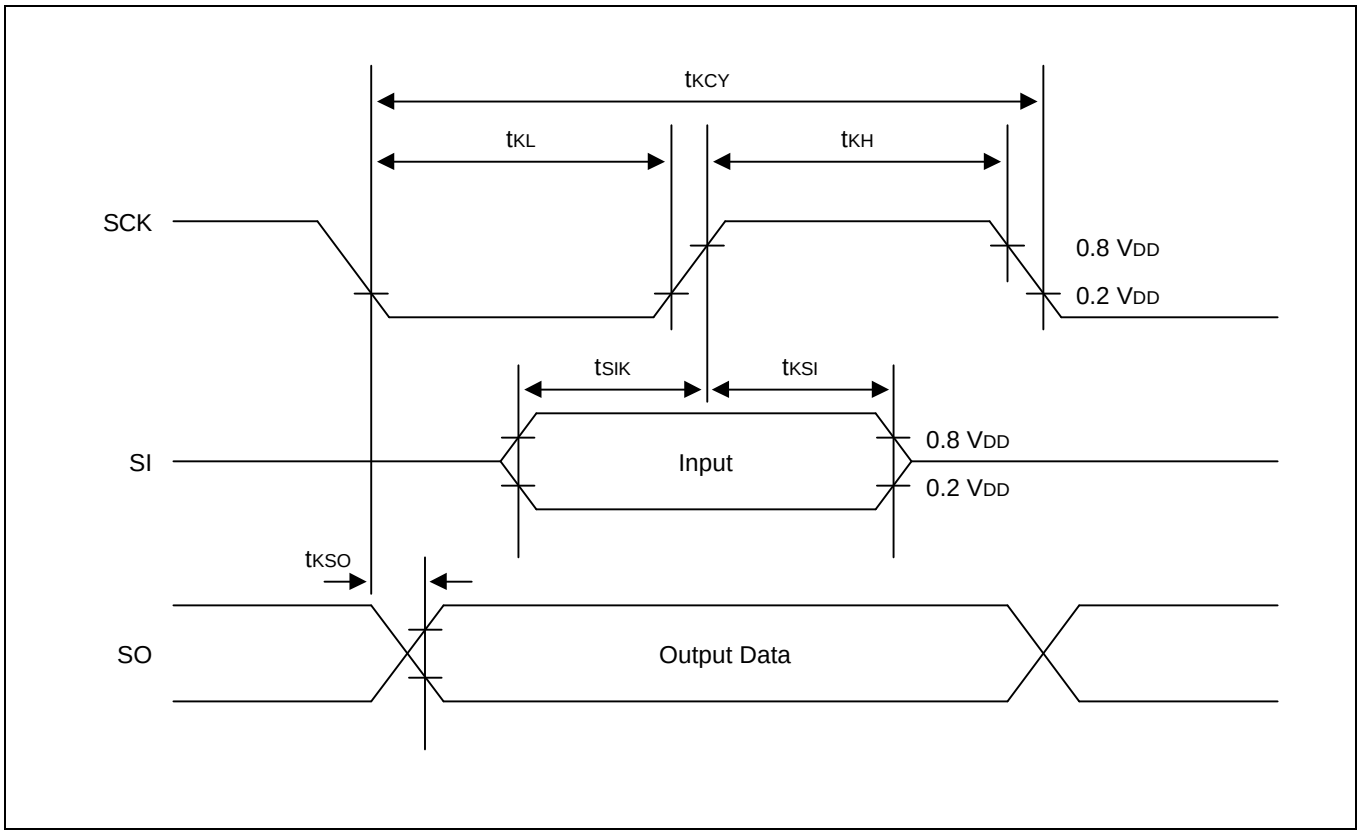


Figure 14-6. Serial Data Transfer Timing

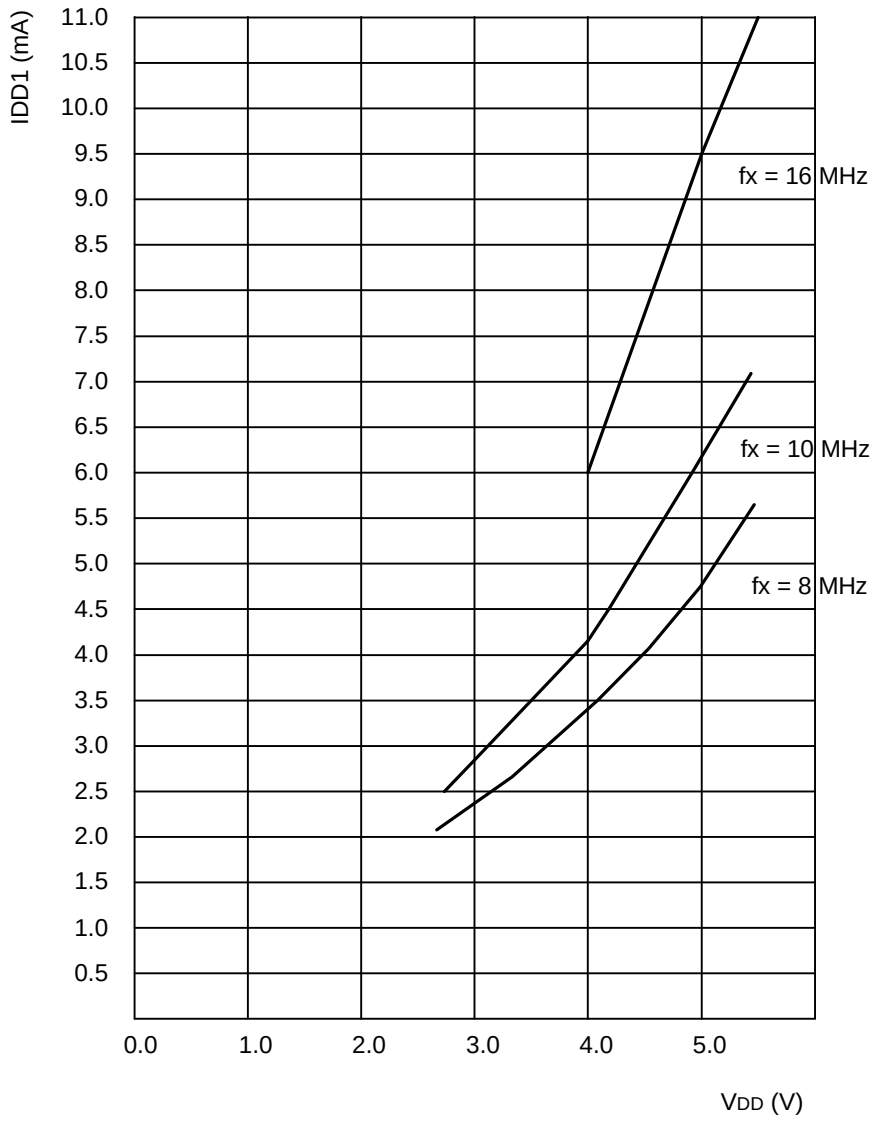


Figure 14-7. I_{DD1} vs V_{DD}

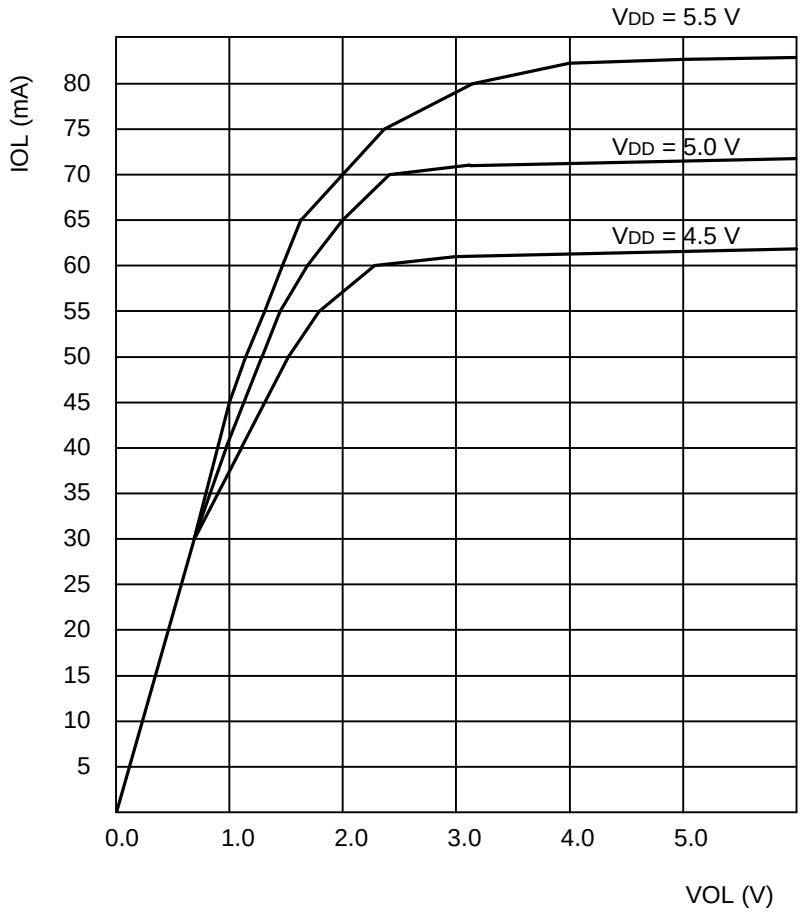


Figure 14-8. I_{OL} vs V_{OL}

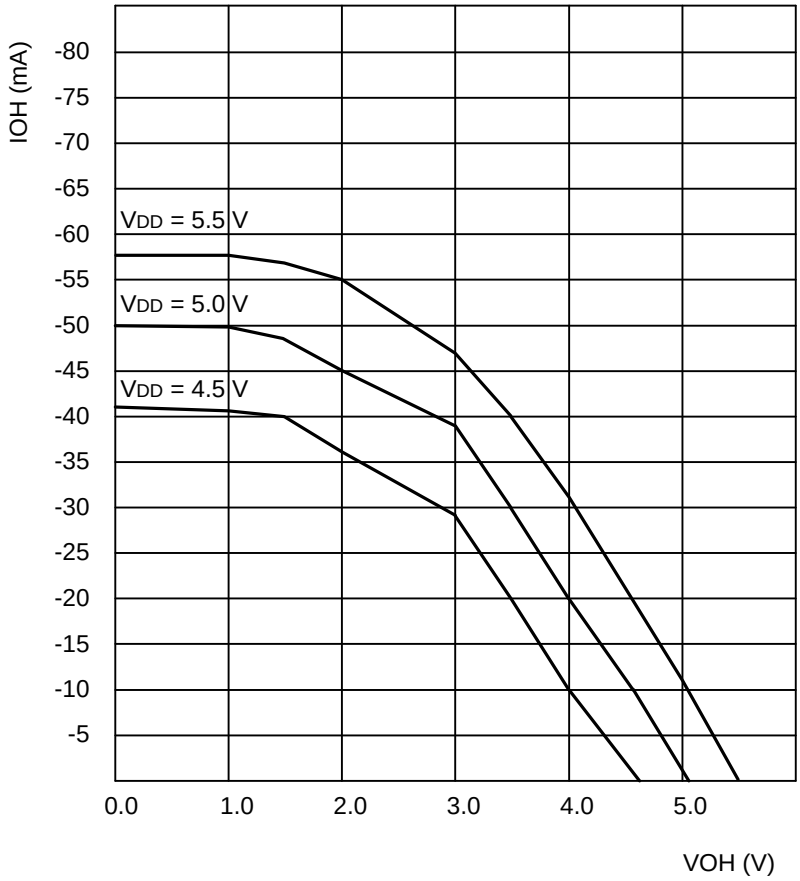
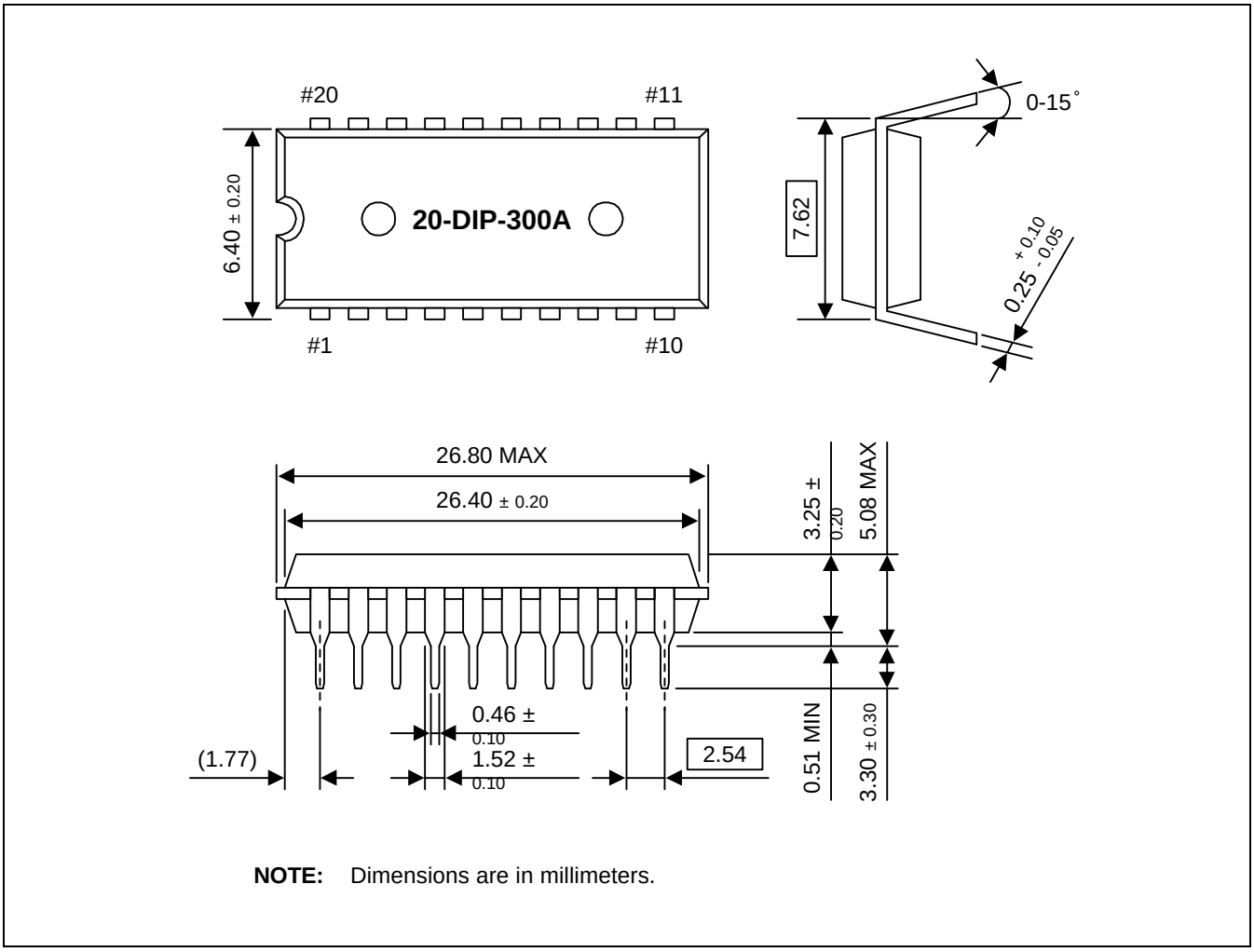


Figure 14-9. I_{OH} vs V_{OH}

15 MECHANICAL DATA

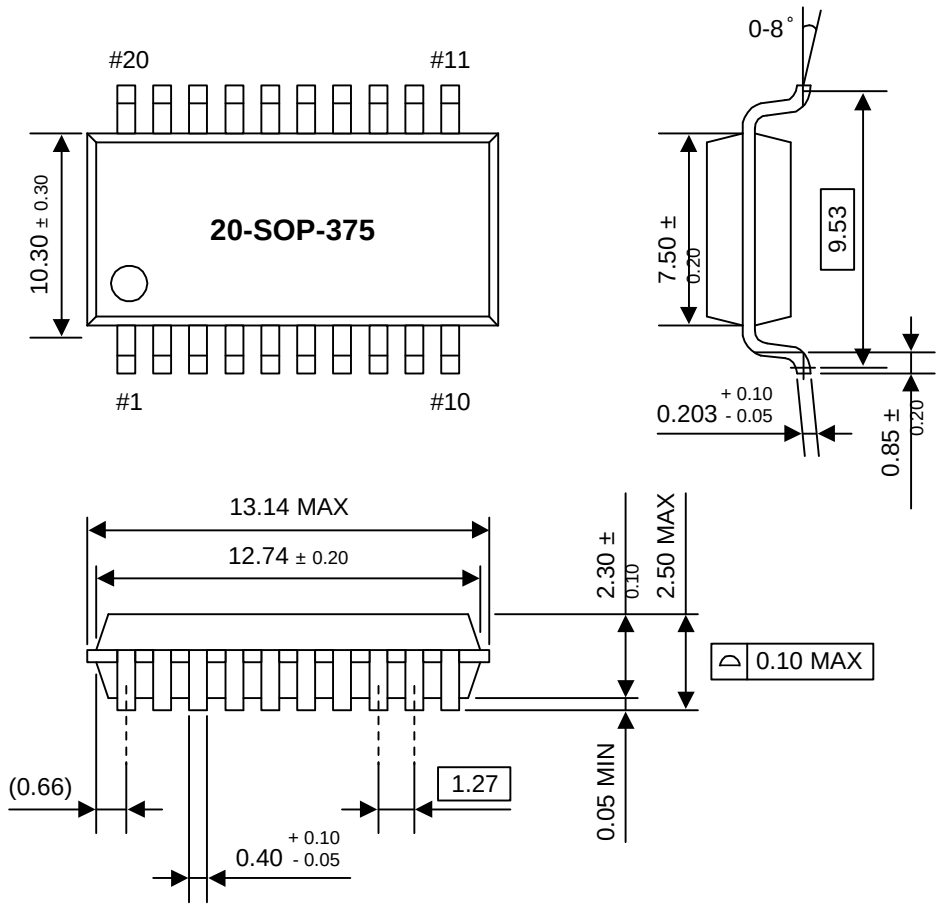
OVERVIEW

The S3C9432/C9434 is available in a 20-pin SDIP package (Samsung: 20-DIP-300A), a 20-pin SOP package (Samsung: 20-SOP-375), a 18-pin DIP package (Samsung: 18-DIP-300A). Package dimensions are shown in Figure 15-1, 15-2, and 15-3.



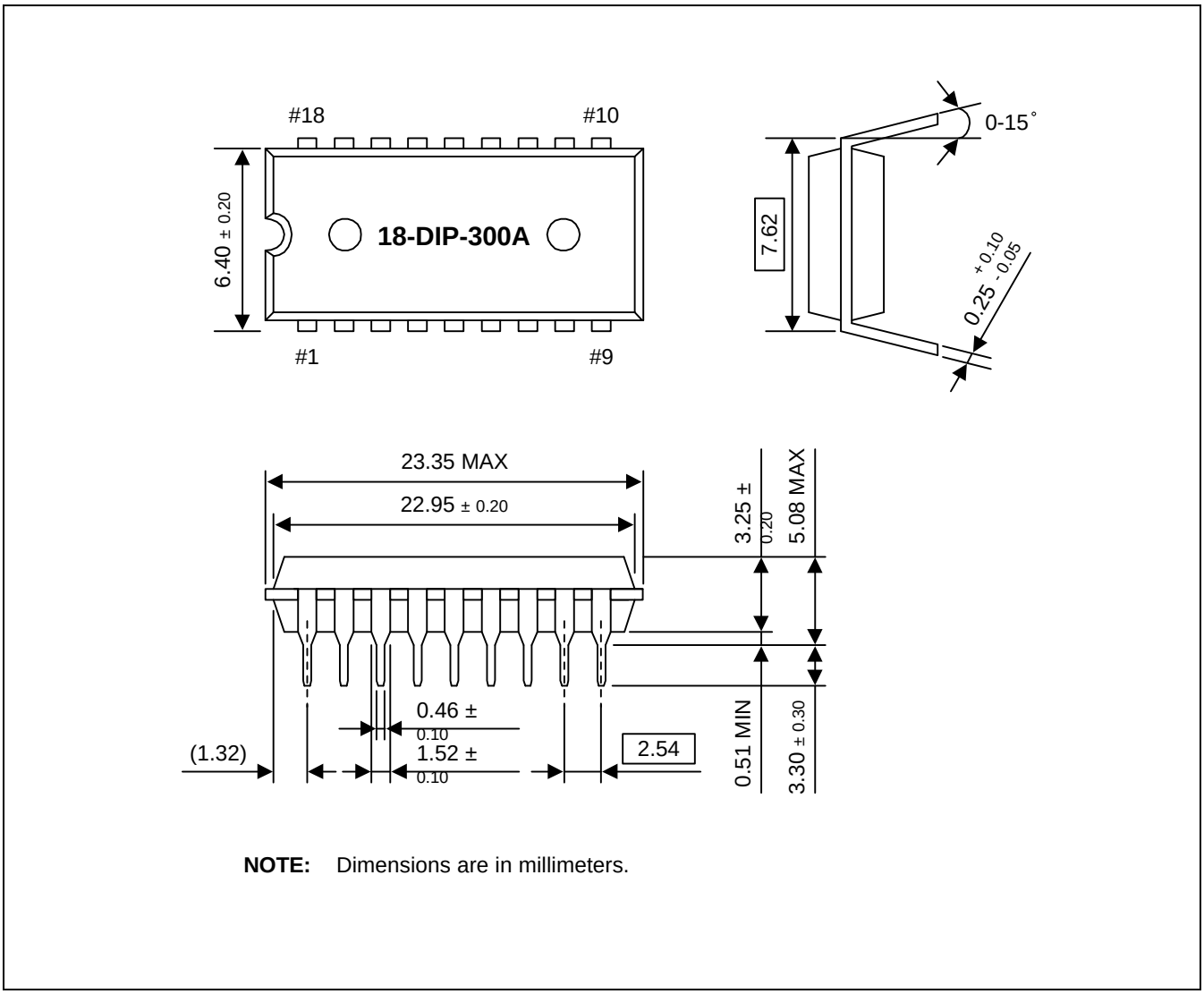
NOTE: Dimensions are in millimeters.

Figure 15-1. 20-DIP-300A Package Dimensions



NOTE: Dimensions are in millimeters.

Figure 15-2. 20-SOP-375 Package Dimensions



NOTE: Dimensions are in millimeters.

Figure 15-3. 18-DIP-300A Package Dimensions

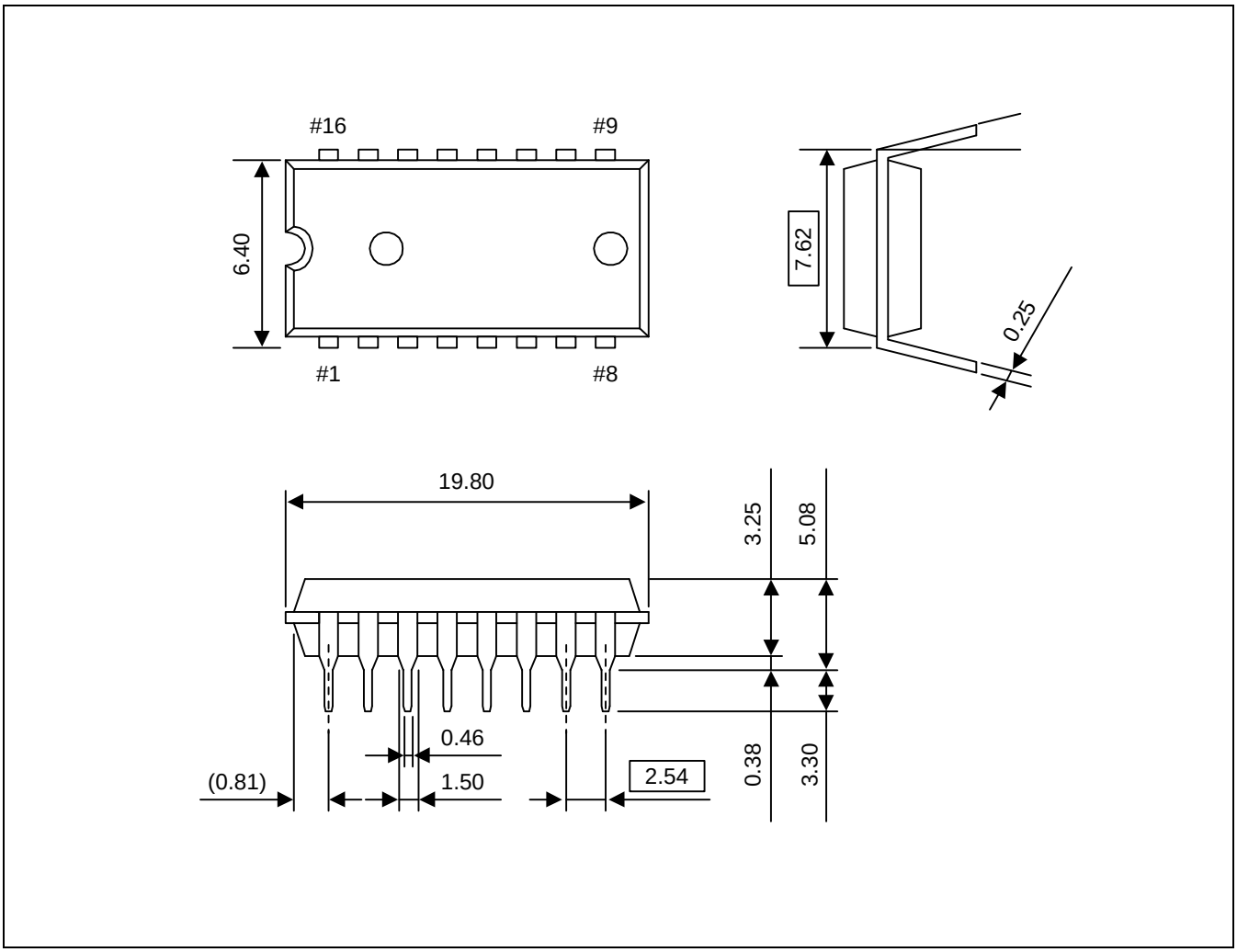


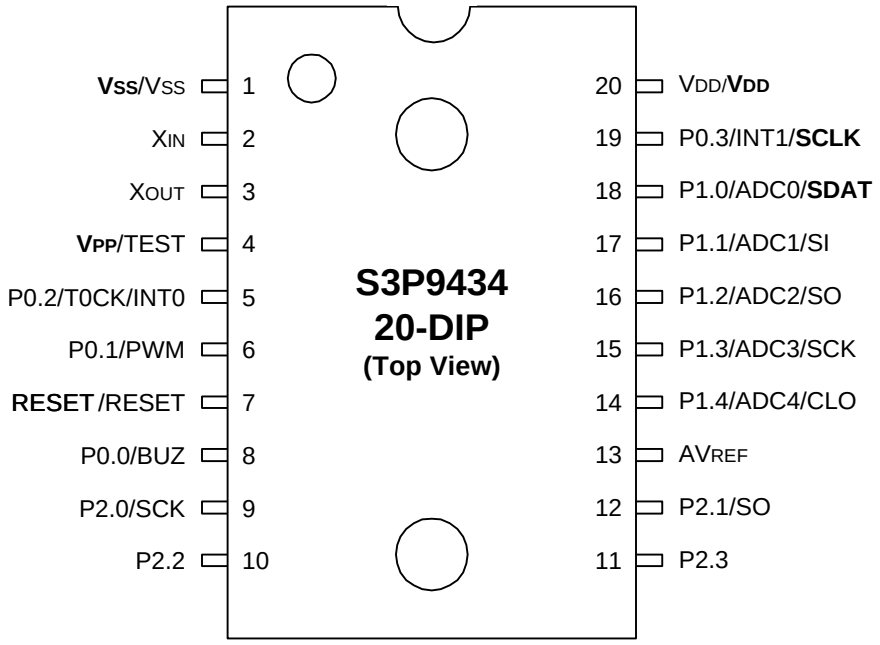
Figure 15-4. 16-DIP-300A Package Dimensions

16 S3P9434 OTP

OVERVIEW

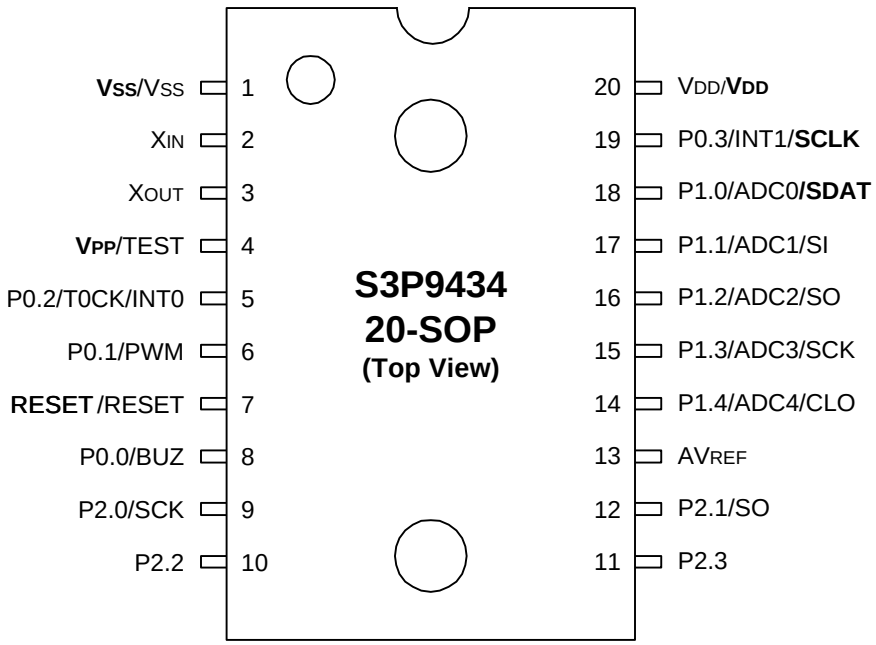
The S3P9434 single-chip CMOS microcontroller is the OTP (One Time Programmable) version of the S3C9432/C9434 microcontroller. It has an on-chip OTP ROM instead of masked ROM. The EPROM is accessed by serial data format.

The S3P9434 is fully compatible with the S3C9432/C9434, in function, in D.C. electrical characteristics, and in pin configuration. Because of its simple programming requirements, the S3P9434 is ideal for use as an evaluation chip for the S3C9432/C9434.



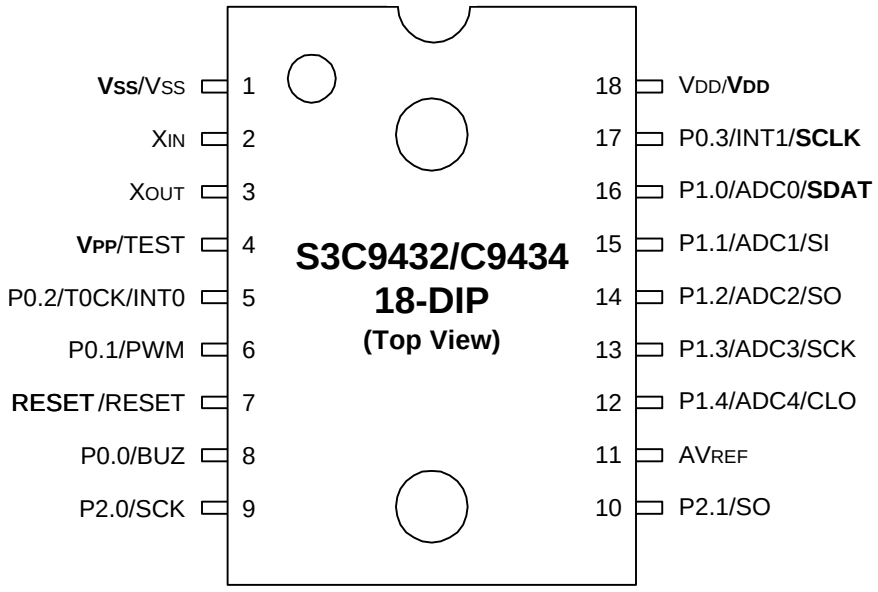
NOTE: The bolds indicate an OTP pin name.

Figure 16-1. Pin Assignment Diagram (20-Pin DIP Package)



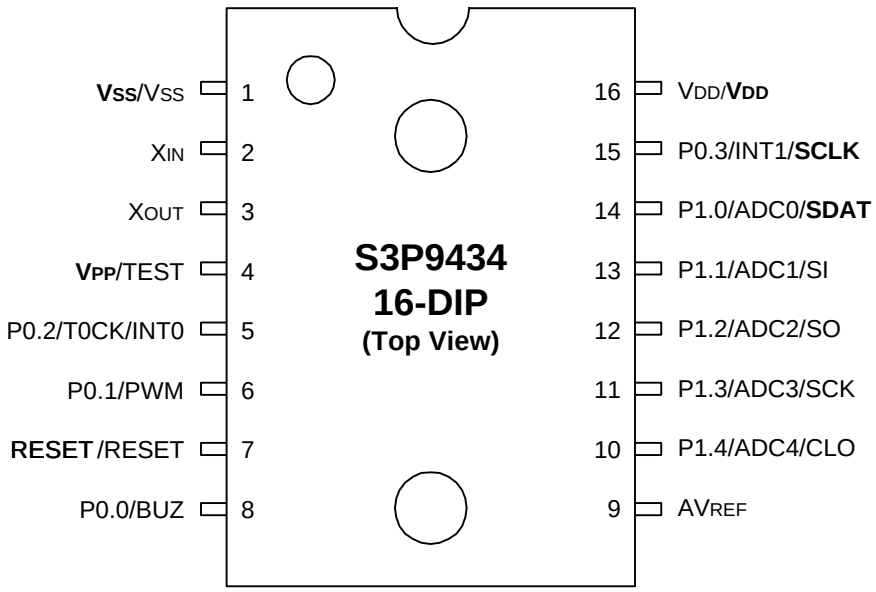
NOTE: The bolds indicate an OTP pin name.

Figure 16-2. Pin Assignment Diagram (20-Pin SOP Package)



NOTE: The bolds indicate an OTP pin name.

Figure 16-3. Pin Assignment Diagram (18-Pin DIP Package)



NOTE: The bolds indicate an OTP pin name.

Figure 16-4. Pin Assingment Diagram (16-Pin DIP Package)

Table 16-1. Descriptions of Pins Used to Read/Write the EPROM

Main Chip Pin Name	During Programming			
	Pin Name	Pin No.	I/O	Function
P0.3	SDAT	18 (20-pin) 16 (18-pin)	I/O	Serial data pin (output when reading, Input when writing) Input and push-pull output port can be assigned
P0.2	SCLK	19 (20-pin) 17 (18-pin)	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}	20 (20-pin), 18 (18-pin) 1 (20-pin), 1 (18-pin)	I	Logic power supply pin.

NOTE: ( ) means the SOP OTP pin number.

Table 16-2. Comparison of S3P9434 and S3C9432/C9434 Features

Characteristic	S3P9434	S3C9432/C9434
Program Memory	4 Kbyte EPROM	2K/4K byte mask ROM
Operating Voltage (V_{DD})	3.0 V to 5.5 V	3.0 V to 5.5 V
OTP Programming Mode	V_{DD} = 5 V, V_{PP} (TEST) = 12.5 V	
Pin Configuration	20 DIP/20 SOP/18 DIP	
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 S3P9434, 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 16-3 below.

Table 16-3. Operating Mode Selection Criteria

V_{DD}	VPP (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.

17 DEVELOPMENT TOOLS

OVERVIEW

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

SHINE

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

SAMA ASSEMBLER

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

SASM86

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

HEX2ROM

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

TARGET BOARDS

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

OTPs

One times programmable microcontrollers (OTPs) are under development for S3C9432/C9434 microcontroller.

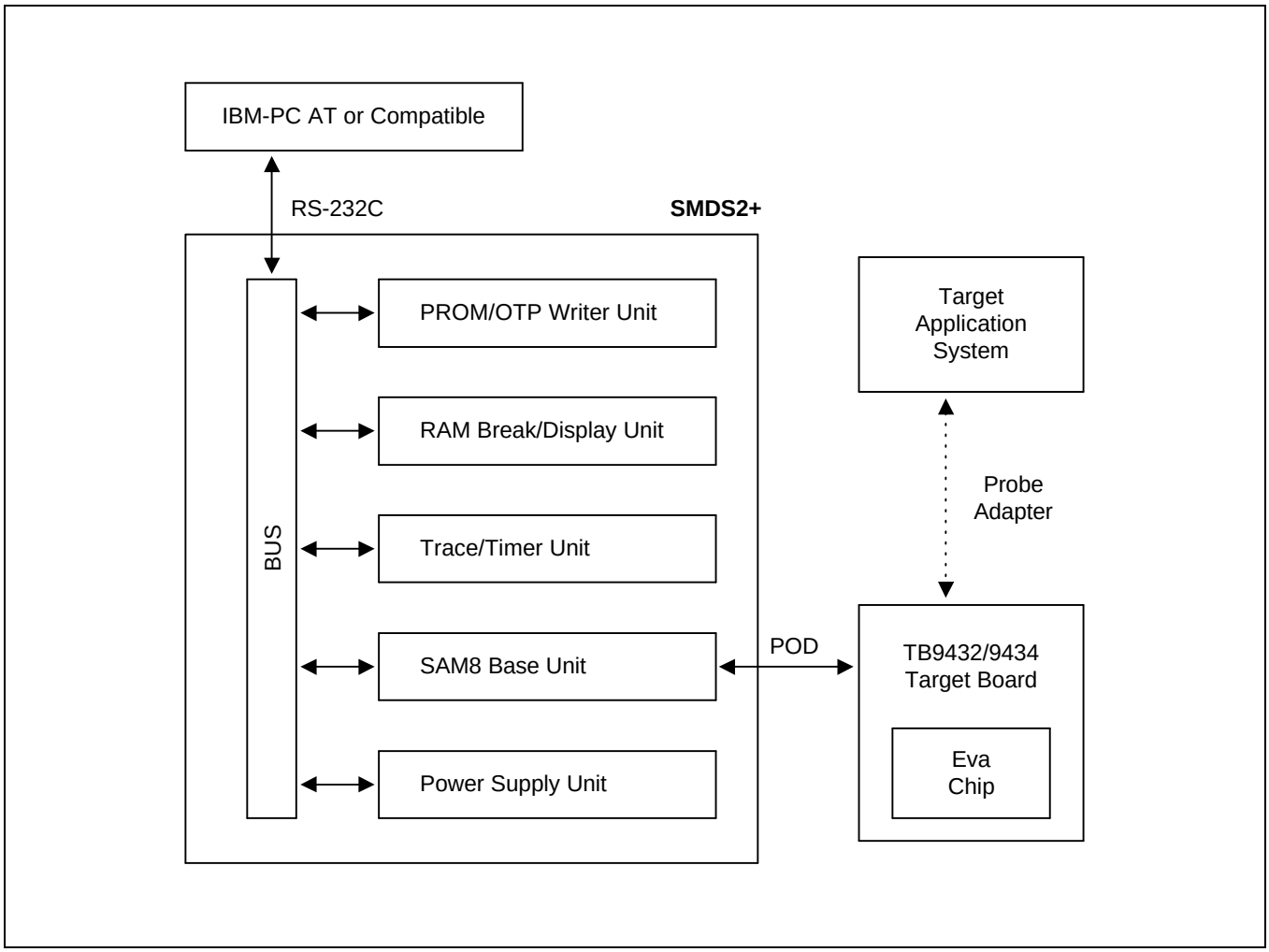


Figure 17-1. SMDS Product Configuration (SMDS2+)

TB9432/9434 TARGET BOARD

The TB9432/9434 target board is used for the S3C9432/C9434 microcontrollers. It is supported by the SMDS2+ development systems. The TB9432/9434 target board can also be used for S3C9432/C9434.

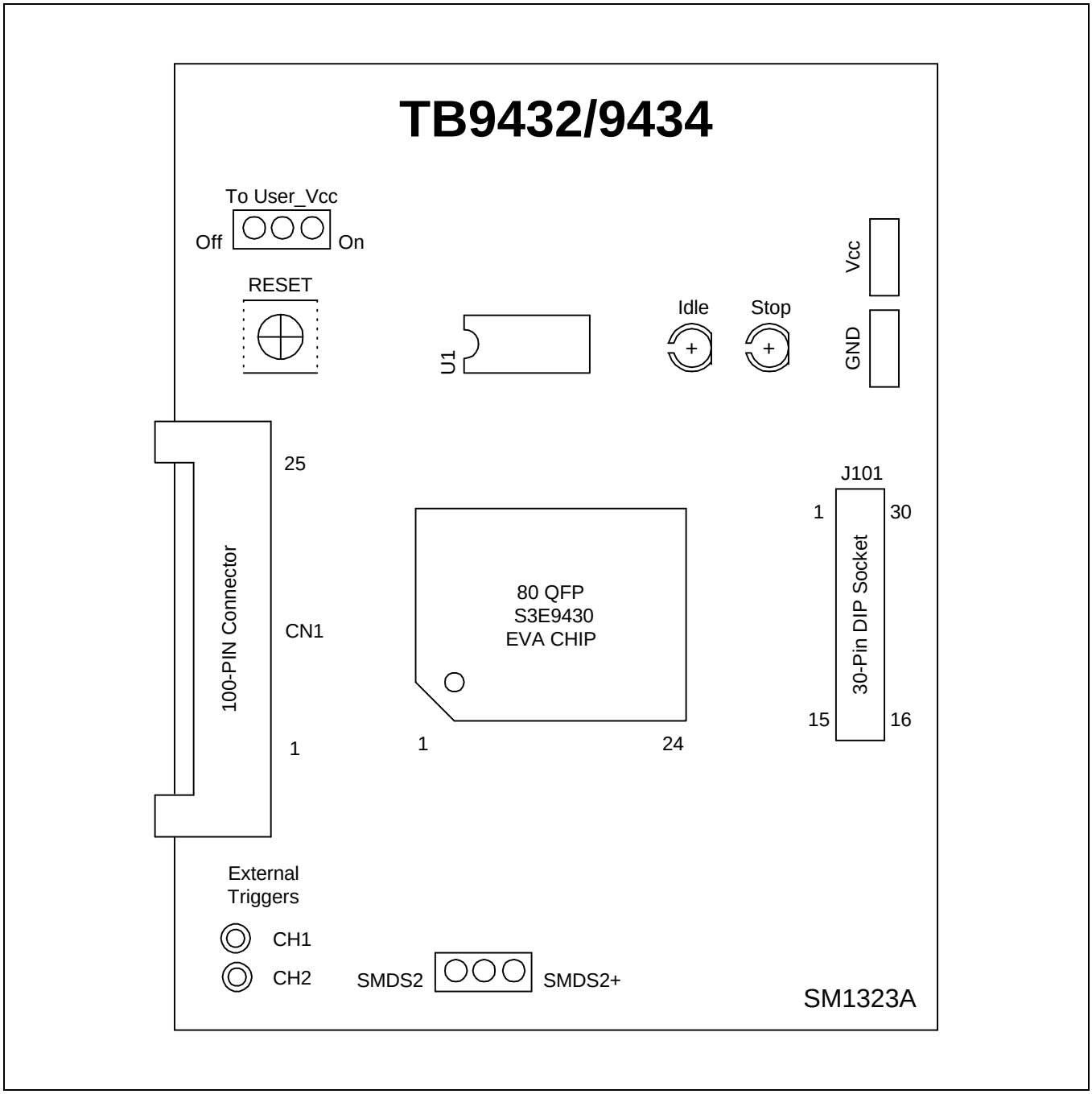
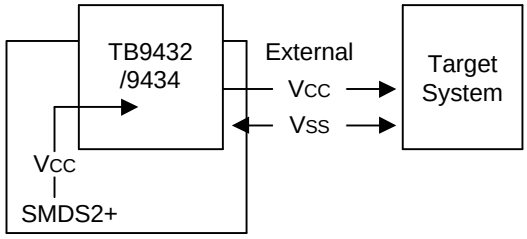
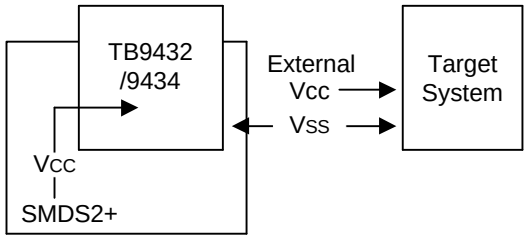


Figure 17-2. TB9432/9434 Target Board Configuration

Table 17-1. Power Selection Settings for TB9432/9434

"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 17-2. The SMDS2+ Tool Selection Setting

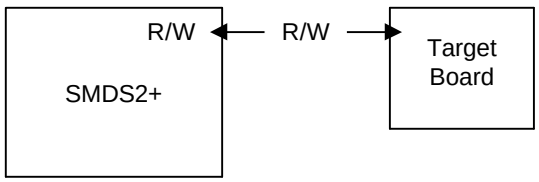
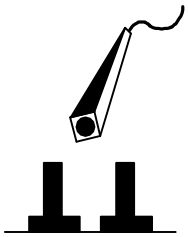
"SW1" Setting	Operating Mode
SMDS ○ ● ● SMDS2+	

Table 17-3. 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.

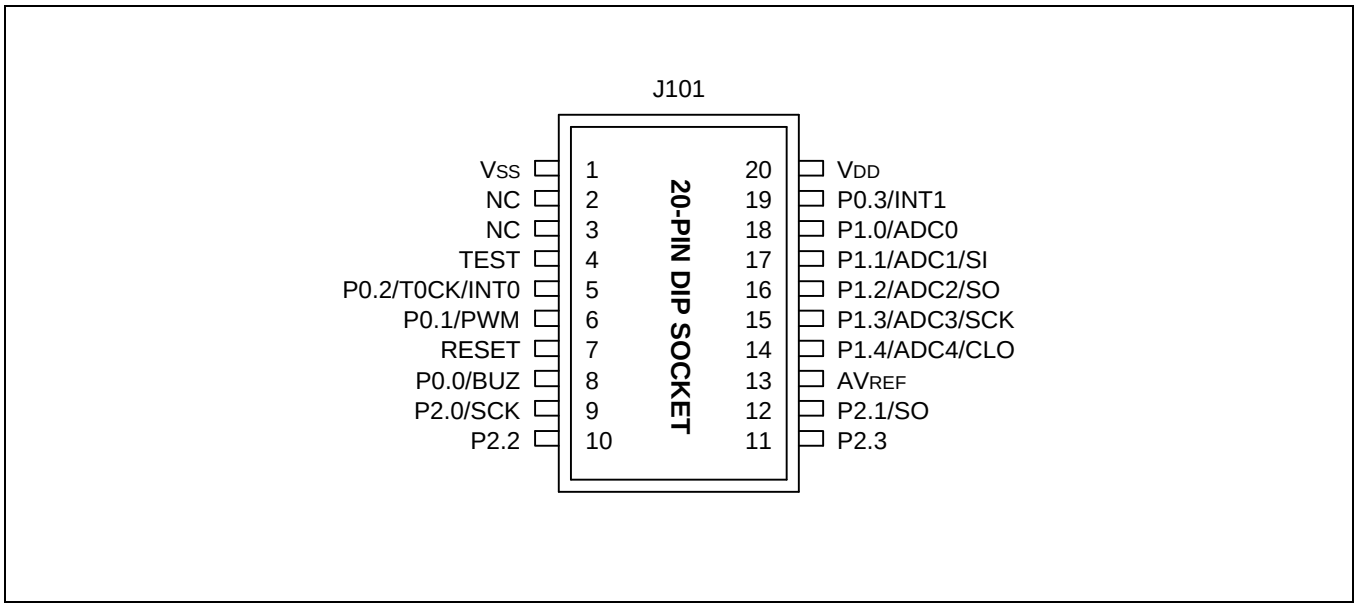


Figure 17-3. 20-Pin Connector for TB9432/9434

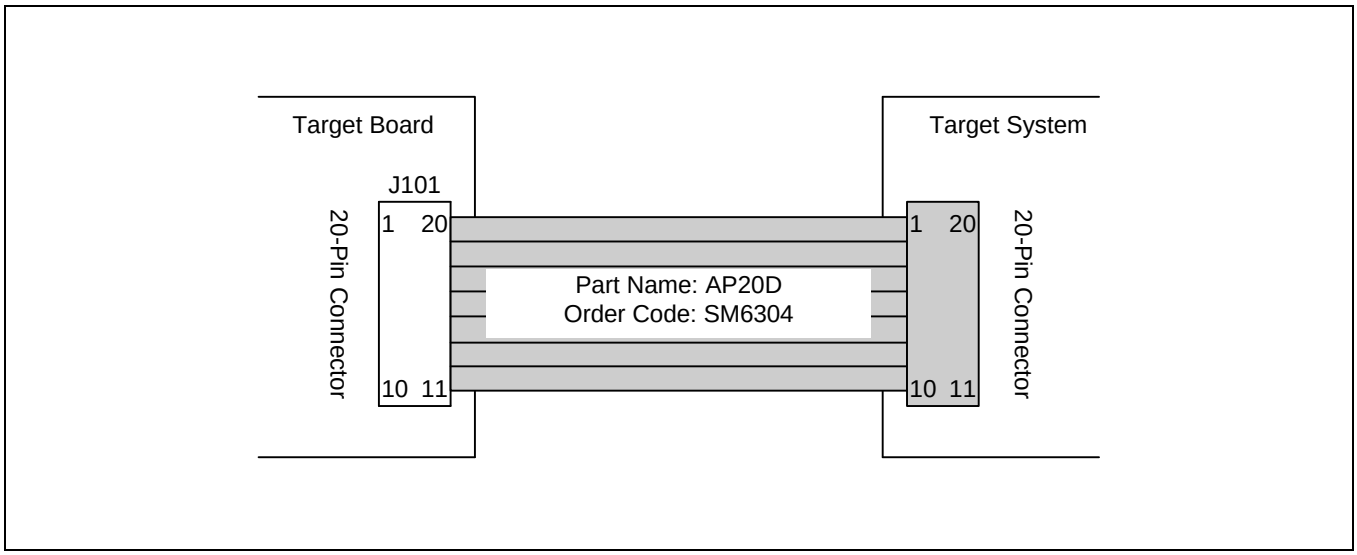


Figure 17-4. S3C9432/C9434 Probe Adapter for 20-DIP Package