Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	SAM88RC
Core Size	8-Bit
Speed	12MHz
Connectivity	SIO, UART/USART
Peripherals	LCD, LVR, PWM, WDT
Number of I/O	70
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/zilog/s3f8s7bxzz-tw8b

Table of Contents

1.3 Features

CPU

- SAM88 RC CPU core

Memory

- Program Memory (ROM)
  - 64K × 8 bits program memory
  - Internal Flash memory (program memory)
    - Sector size: 128 bytes
    - 10 years data retention
    - Fast programming time: User program and sector erase available
    - Endurance: 10,000 erase/program cycles
    - External serial programming support
    - Expandable OBP™ (on board program) sector
- Data Memory (RAM)
  - Including LCD display data memory
  - 2,114 × 8 bits data memory

Instruction Set

- 78 instructions
- Idle and stop instructions added for power-down modes

70 I/O Pins

- I/O: 18 pins (Sharing with other signal pins)
- I/O: 52 pins (Sharing with LCD signal outputs)

Interrupts

- 8 interrupt levels and 31 interrupt sources
- Fast interrupt processing feature

8-Bit Basic Timer

- Watchdog timer function
- 4 kinds of clock source

Package Type

- 80-QFP-1420C, 80-TQFP-1212, 80-LQFP

IVC

- Internal Voltage Converter for 5 V operations

Smart Option

- Low Voltage Reset (LVR) level and enable/disable are at your hardwired option (ROM address 3FH)
- ISP related option selectable (ROM address 3EH)

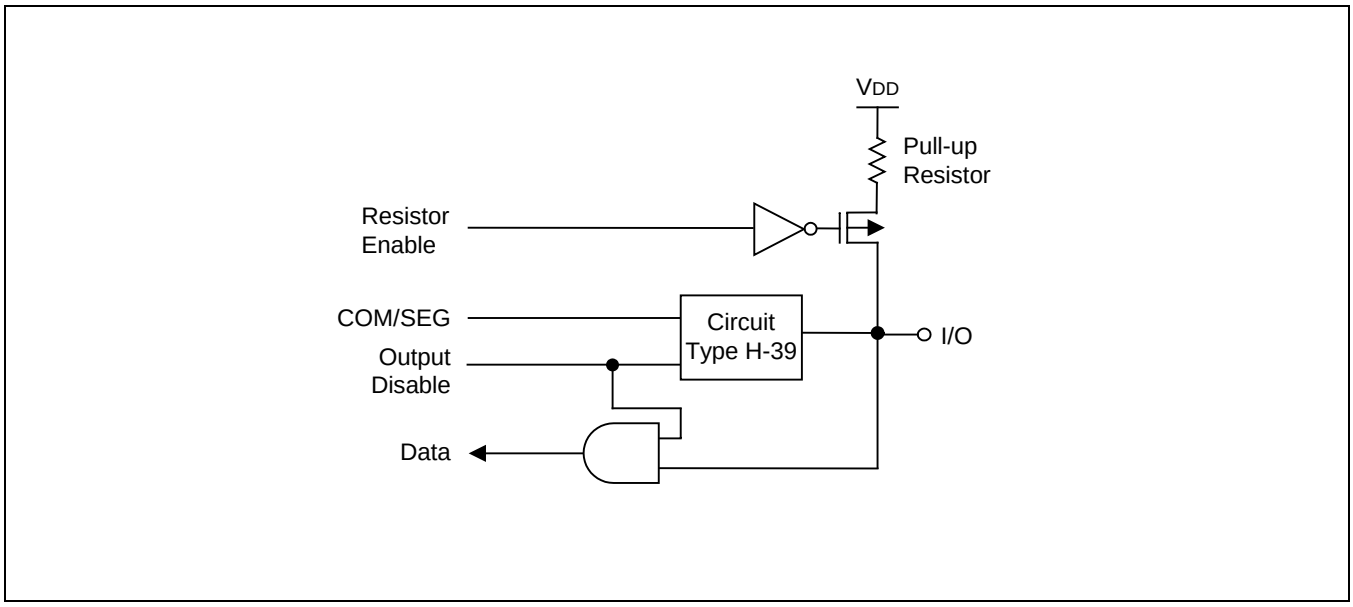


Figure 1-11 Pin Circuit Type H-44 (P0)

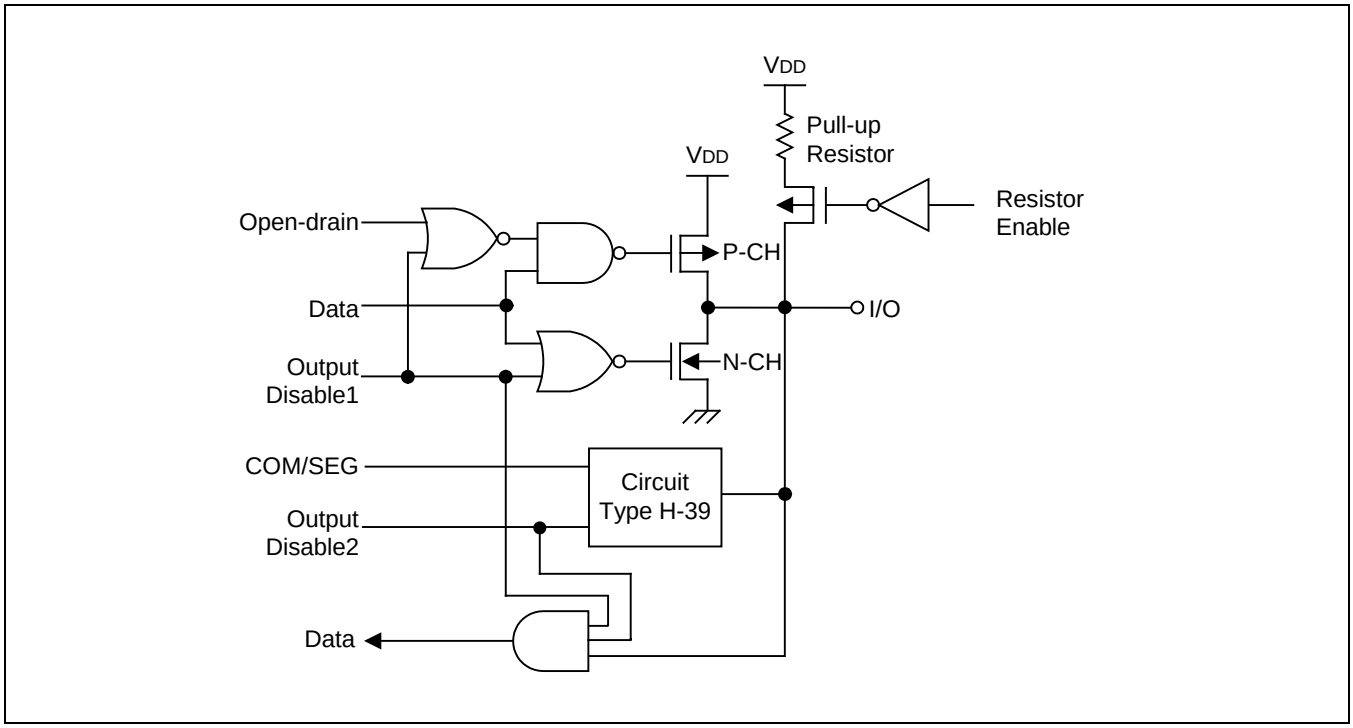


Figure 1-12 Pin Circuit Type H-42 (P1, P3, P6 – P8)

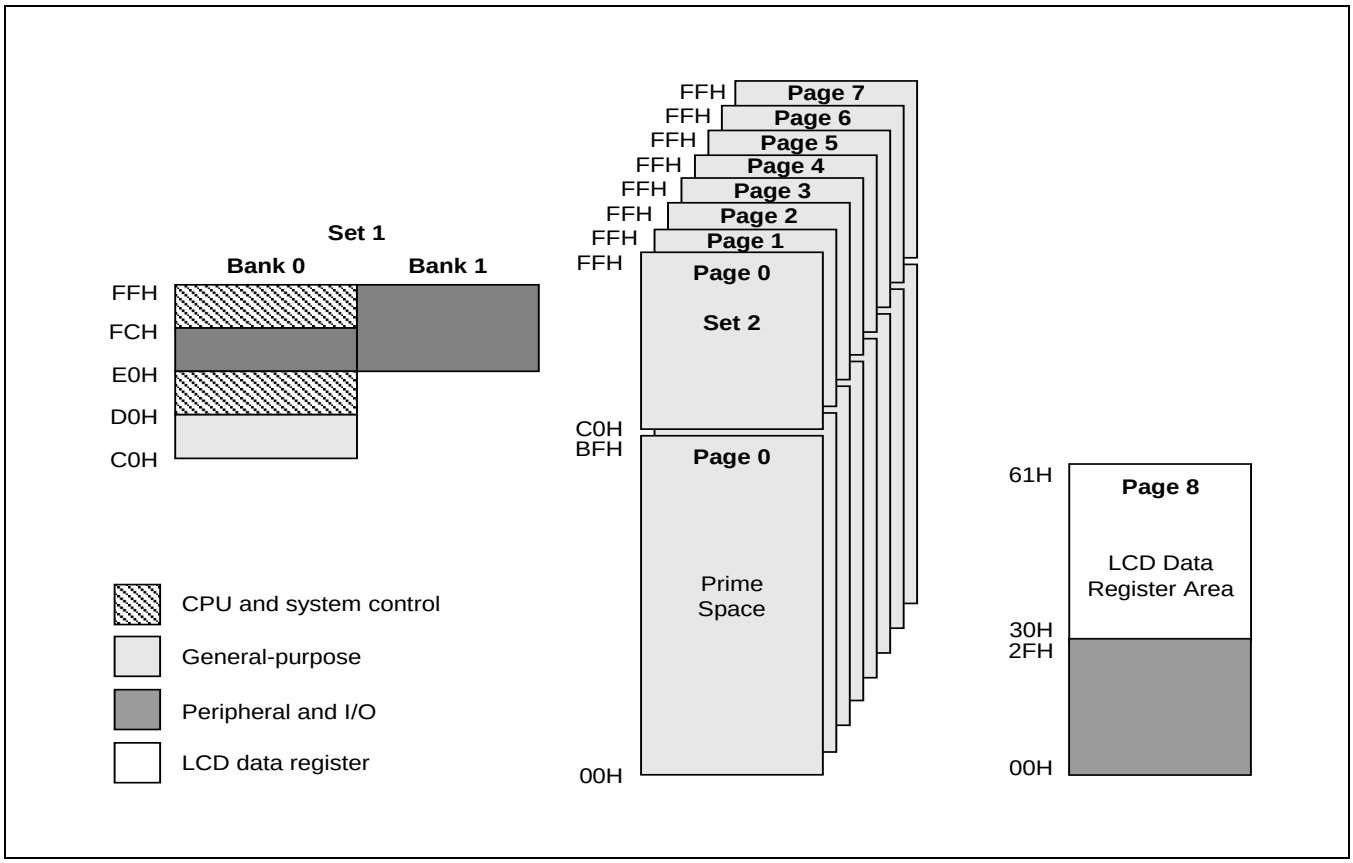


Figure 2-5 Set 1, Set 2, Prime Area Register, and LCD Data Register Map

2.3.5 Working Registers

Instructions can access specific 8-bit registers or 16-bit register pairs using either 4-bit or 8-bit address fields. When 4-bit working register addressing is used, the 256-byte register file can be seen by the programmer as one that consists of 32 8-byte register groups or "slices." Each slice comprises of eight 8-bit registers.

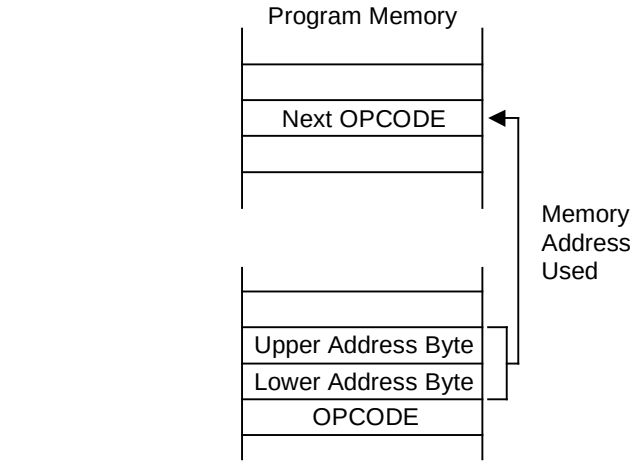
Using the two 8-bit register pointers, RP1 and RP0, two working register slices can be selected at any one time to form a 16-byte working register block. Using the register pointers, you can move this 16-byte register block anywhere in the addressable register file, except the set 2 area.

The terms slice and block are used in this manual to help you visualize the size and relative locations of selected working register spaces:

- One working register slice is 8 bytes (eight 8-bit working registers, R0–R7 or R8–R15)
- One working register block is 16 bytes (sixteen 8-bit working registers, R0–R15)

All the registers in an 8-byte working register slice have the same binary value for their five most significant address bits. This makes it possible for each register pointer to point to one of the 24 slices in the register file. The base addresses for the two selected 8-byte register slices are contained in register pointers RP0 and RP1.

After a reset, RP0 and RP1 always point to the 16-byte common area in set 1 (C0H–CFH).



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

4.1.25 P2CONH: Port 2 Control Register (High Byte) (E6H, Set 1, Bank 1)

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Addressing Mode	Register addressing mode only							

.7–.6 P2.7/INT7/SEG19 Configuration Bits

0	0	Schmitt trigger input mode
0	1	Schmitt trigger input mode, pull-up
1	0	Alternative function (LCD signal)
1	1	Output mode, push-pull

.5-.4 P2.6/INT6/SEG18 Configuration Bits

0	0	Schmitt trigger input mode
0	1	Schmitt trigger input mode, pull-up
1	0	Alternative function (LCD signal)
1	1	Output mode, push-pull

.3–.2 P2.5/INT5/SEG17 Configuration Bits

0	0	Schmitt trigger input mode
0	1	Schmitt trigger input mode, pull-up
1	0	Alternative function (LCD signal)
1	1	Output mode, push-pull

.1–.0 P2.4/INT4/SEG16 Configuration Bits

0	0	Schmitt trigger input mode
0	1	Schmitt trigger input mode, pull-up
1	0	Alternative function (LCD signal)
1	1	Output mode, push-pull

4.1.40 P5CONH: Port 5 Control Register (High Byte) (F0H, Set 1, Bank 1)

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Addressing Mode	Register addressing mode only							

.7–.6 **P5.7/XT_{IN} Configuration Bits**

0	0	Input mode
0	1	Input mode, pull-up
1	0	Alternative function (XT_{IN})
1	1	Output mode, push-pull

.5–.4 **P5.6/XT_{OUT} Configuration Bits**

0	0	Input mode
0	1	Input mode, pull-up
1	0	Alternative function (XT_{OUT})
1	1	Output mode, push-pull

.3–.2 **P5.5/CB (NOTE) Configuration Bits**

0	0	Input mode
0	1	Input mode, pull-up
1	0	Not available
1	1	Output mode, push-pull

.1–.0 **P5.4/CA (NOTE) Configuration Bits**

0	0	Input mode
0	1	Input mode, pull-up
1	0	Not available
1	1	Output mode, push-pull

NOTE: Refer to LCON register in Chapter15.

4.1.67 TD0CON: Timer D0 Control Register (FAH, Set 1, Bank 1)

Bit Identifier	.7	.6	.5	.4	.3	.2	.1	.0
RESET Value	0	0	0	0	0	0	0	0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW
Addressing Mode	Register addressing mode only							

.7–.5 Timer D0 Clock Selection Bits

0	0	0	fxx/1024
0	0	1	fxx/256
0	1	0	fxx/64
0	1	1	fxx/8
1	0	0	fxx/1
1	0	1	External clock (TD0CLK) falling edge
1	1	0	External clock (TD0CLK) rising edge
1	1	1	Counter stop

.4–.3 Timer D0 Operating Mode Selection Bits

0	0	Interval mode (TD0OUT)
0	1	Capture mode (Capture on rising edge, counter running, OVF can occur)
1	0	Capture mode (Capture on falling edge, counter running, OVF can occur)
1	1	PWM mode (OVF and match interrupt can occur)

.2 Timer D0 Counter Clear Bit

0	No effect
1	Clear the timer D0 counter (when write)

.1 Timer D0 Match/Capture Interrupt Enable Bit

0	Disable interrupt
1	Enable interrupt

.0 Timer D0 Overflow Interrupt Enable Bit

0	Disable overflow interrupt
1	Enable overflow interrupt

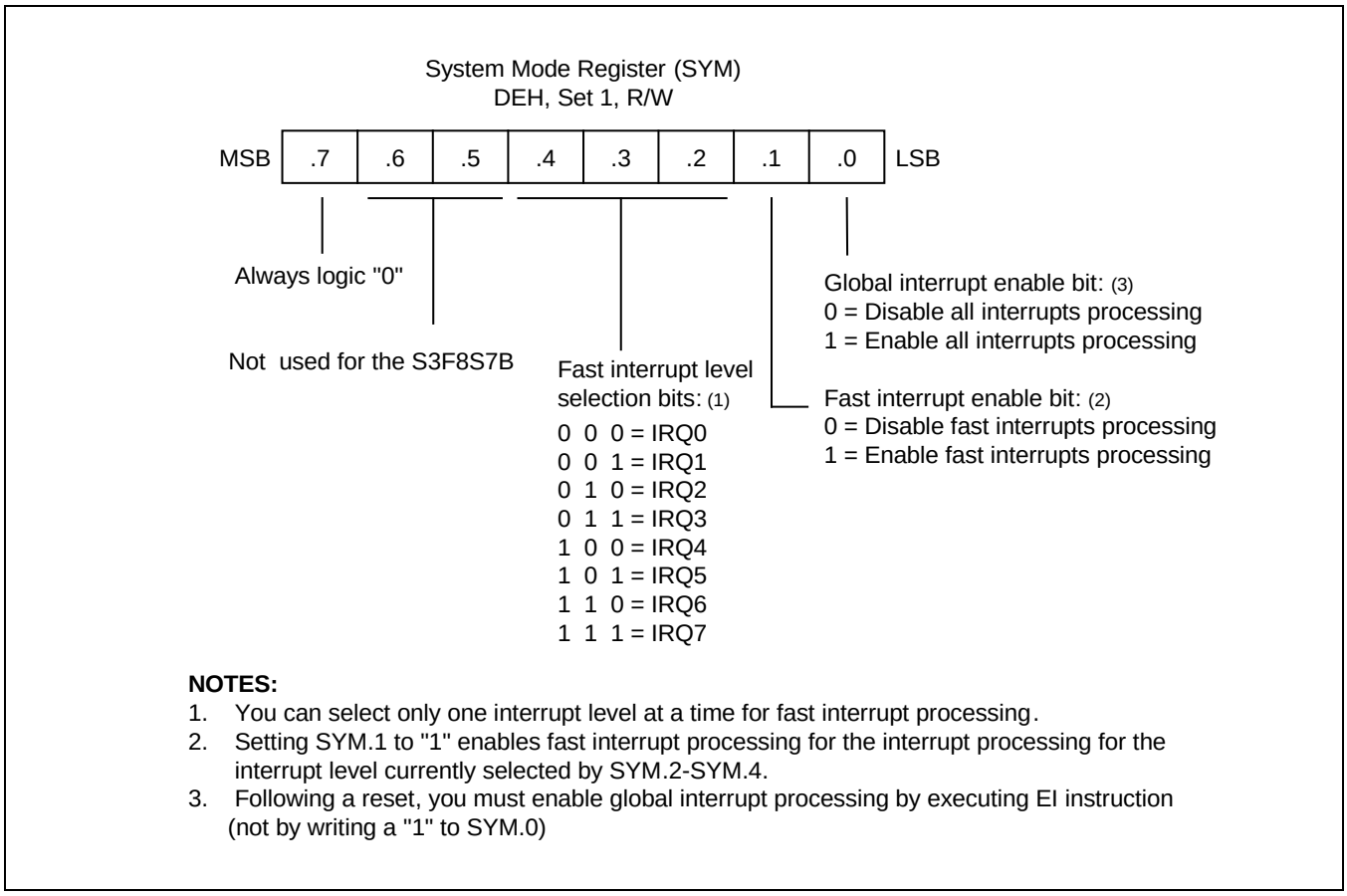
NOTE: The TD0CON.2 value is automatically cleared to "0" after being cleared counter.

5.9 System Mode Register (SYM)

The system mode register, SYM (set 1, DEH), is used to globally enable and disable interrupt processing and to control fast interrupt processing (Refer to *Figure 5-5*).

A reset clears SYM.1, and SYM.0 to "0". The 3-bit value for fast interrupt level selection, SYM.4-SYM.2, is undetermined.

The instructions EI and DI enable and disable global interrupt processing, respectively, by modifying the bit 0 value of the SYM register. In order to enable interrupt processing an Enable Interrupt (EI) instruction must be included in the initialization routine, which follows a reset operation. Although you can manipulate SYM.0 directly to enable and disable interrupts during the normal operation, it is recommended to use the EI and DI instructions for this purpose.



NOTES:

1. You can select only one interrupt level at a time for fast interrupt processing.
2. Setting SYM.1 to "1" enables fast interrupt processing for the interrupt processing for the interrupt level currently selected by SYM.2-SYM.4.
3. Following a reset, you must enable global interrupt processing by executing EI instruction (not by writing a "1" to SYM.0)

Figure 5-5 System Mode Register (SYM)

6.5.17 CP-Compare

CP dst,src

Operation: dst – src

The source operand is compared to (subtracted from) the destination operand, and the appropriate flags are set accordingly. The contents of both operands are unaffected by the comparison.

Flags:
C: Set if a "borrow" occurred (src > dst); cleared otherwise.
Z: Set if the result is "0"; cleared otherwise.
S: Set if the result is negative; cleared otherwise.
V: Set if arithmetic overflow occurred; cleared otherwise.
D: Unaffected.
H: Unaffected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	Addr Mode src
opc | dst | src	2	4	A2	r	r
		6	A3	r	lr
opc | src | dst	3	6	A4	R	R
		6	A5	R	IR
opc | dst | src	3	6	A6	R	IM

Examples: 1. Given: R1 = 02H and R2 = 03H:

```
CP      R1,R2   →       Set the C and S flags
```

Destination working register R1 contains the value 02H and source register R2 contains the value 03H. The statement "CP R1,R2" subtracts the R2 value (source/subtrahend) from the R1 value (destination/minuend). Because a "borrow" occurs and the difference is negative, C and S are "1".

2. Given: R1 = 05H and R2 = 0AH:

```
        CP      R1,R2
        JP      UGE,SKIP
        INC     R1
SKIP    LD      R3,R1
```

In this example, destination working register R1 contains the value 05H which is less than the contents of the source working register R2 (0AH). The statement "CP R1, R2" generates C = "1" and the JP instruction does not jump to the SKIP location. After the statement "LD R3, R1" executes, the value 06H remains in working register R3.

6.5.32 IRET-Interrupt Return

IRET IRET (Normal) IRET (Fast)

Operation:
FLAGS ← @SP PC ↔ IP
SP ← SP + 1 FLAGS ← FLAGS'
PC ← @SP FIS ← 0
SP ← SP + 2
SYM(0) ← 1

This instruction is used at the end of an interrupt service routine. It restores the flag register and the program counter. It also re-enables global interrupts. A "normal IRET" is executed only if the fast interrupt status bit (FIS, bit one of the FLAGS register, 0D5H) is cleared (= "0"). If a fast interrupt occurred, IRET clears the FIS bit that was set at the beginning of the service routine.

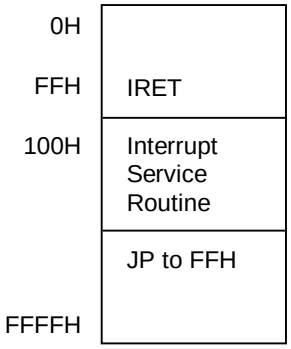
Flags: All flags are restored to their original settings (that is, the settings before the interrupt occurred).

Format:

IRET (Normal)	Bytes	Cycles	Opcode (Hex)
opc	1	10 (internal stack) 12 (internal stack)	BF

IRET (Fast)	Bytes	Cycles	Opcode (Hex)
opc	1	6	BF

Example: In the figure below, the instruction pointer is initially loaded with 100H in the main program before interrupts are enabled. When an interrupt occurs, the program counter and instruction pointer are swapped. This causes the PC to jump to address 100H and the IP to keep the return address. The last instruction in the service routine normally is a jump to IRET at address FFH. This causes the instruction pointer to be loaded with 100H "again" and the program counter to jump back to the main program. Now, the next interrupt can occur and the IP is still correct at 100H.



NOTE: In the fast interrupt example above, if the last instruction is not a jump to IRET, you must pay attention to the order of the last two instructions. The IRET cannot be immediately proceeded by a clearing of the interrupt status (as with a reset of the IPR register).

7.3 Sub Oscillator Circuits

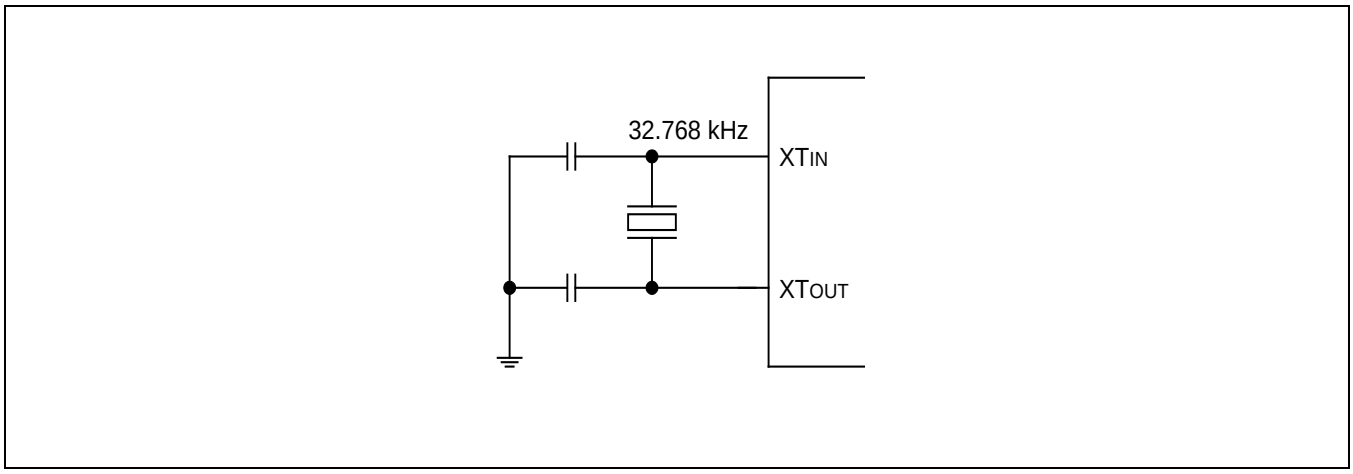


Figure 7-4 Crystal Oscillator (fxt)

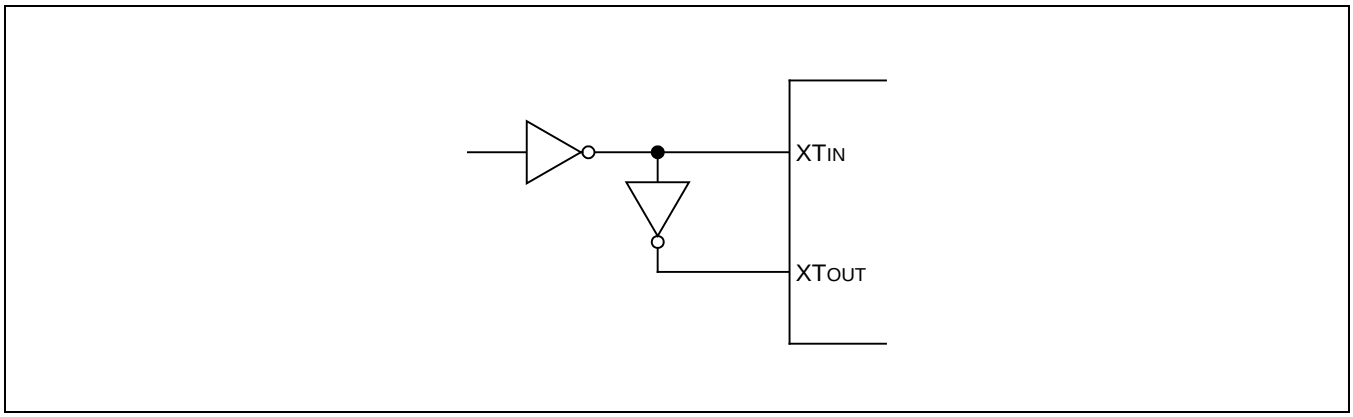


Figure 7-5 External Oscillator (fxt)

9.2.3 Port 2

Port 2 is an 8-bit I/O port with individually configurable pins. Port 2 pins are accessed directly by writing or reading the port 2 data register, P2 at location 02H in page 8. P2.0–P2.7 can serve as inputs (with or without pull-ups), and push pull outputs. And the P2.7–P2.0 can serve as segment pins for LCD or you can configure the following alternative functions:

- Low-byte pins (P2.0–P2.3): INT0-INT3
- High-byte pins (P2.4–P2.7): INT4-INT7

9.2.3.1 Port 2 Control Register (P2CONH, P2CONL)

Port 2 has two 8-bit control registers: P2CONH for P2.4–P2.7 and P2CONL for P2.0–P2.3. A reset clears the P2CONH and P2CONL registers to "00H", configuring all pins to input mode. In input mode, three different selections are available:

- Schmitt trigger input with interrupt generation on falling signal edges.
- Schmitt trigger input with interrupt generation on rising signal edges.
- Schmitt trigger input with interrupt generation on falling/rising signal edges.

When programming the port, please remember that any alternative peripheral I/O function you configure using the port 2 control registers must also be enabled in the associated peripheral module.

9.2.3.2 Port 2 Interrupt Enable and Pending Registers (P2INTH, P2INTL, P2PND)

To process external interrupts at the port 2 pins, the additional control registers are provided: the port 2 interrupt enable register P2INTH (high byte, E8H, set 1, bank 1), P2INTL (Low byte, E9H, set1, bank1) and the port 2 interrupt pending register P2PND (EAH, set 1, bank 1).

The port 2 interrupt pending register P2PND lets you check for interrupt pending conditions and clear the pending condition when the interrupt service routine has been initiated. The application program detects interrupt requests by polling the P2PND register at regular intervals.

When the interrupt enable bit of any port 2 pin is "1", a rising or falling signal edge at that pin will generate an interrupt request. The corresponding P2PND bit is then automatically set to "1" and the IRQ level goes low to signal the CPU that an interrupt request is waiting. When the CPU acknowledges the interrupt request, application software must the clear the pending condition by writing a "0" to the corresponding P2PND bit.

11 8-Bit Timer A/B

11.1 8-Bit Timer A

11.1.1 Overview

The 8-bit timer A is an 8-bit general-purpose timer/counter.

Timer A has three operating modes, one of which you select using the appropriate TACON setting:

- Interval timer mode (Toggle output at TAOUT pin)
- Capture input mode with a rising or falling edge trigger at the TACAP pin
- PWM mode (TAPWM)

Timer A has the following functional components:

- Clock frequency divider (fxx divided by 1024, 256, 64, 8 or 1) with multiplexer
- External clock input pin (TACLK)
- 8-bit counter (TACNT), 8-bit comparator, and 8-bit reference data register (TADATA)
- I/O pins for capture input (TACAP) or PWM or match output (TAPWM, TAOUT)
- Timer A overflow interrupt (IRQ0 vector D0H) and match/capture interrupt (IRQ0 vector CEH) generation
- Timer A control register, TACON (set 1, Bank 0, E2H, read/write)

15 LCD Controller/Driver

15.1 Overview

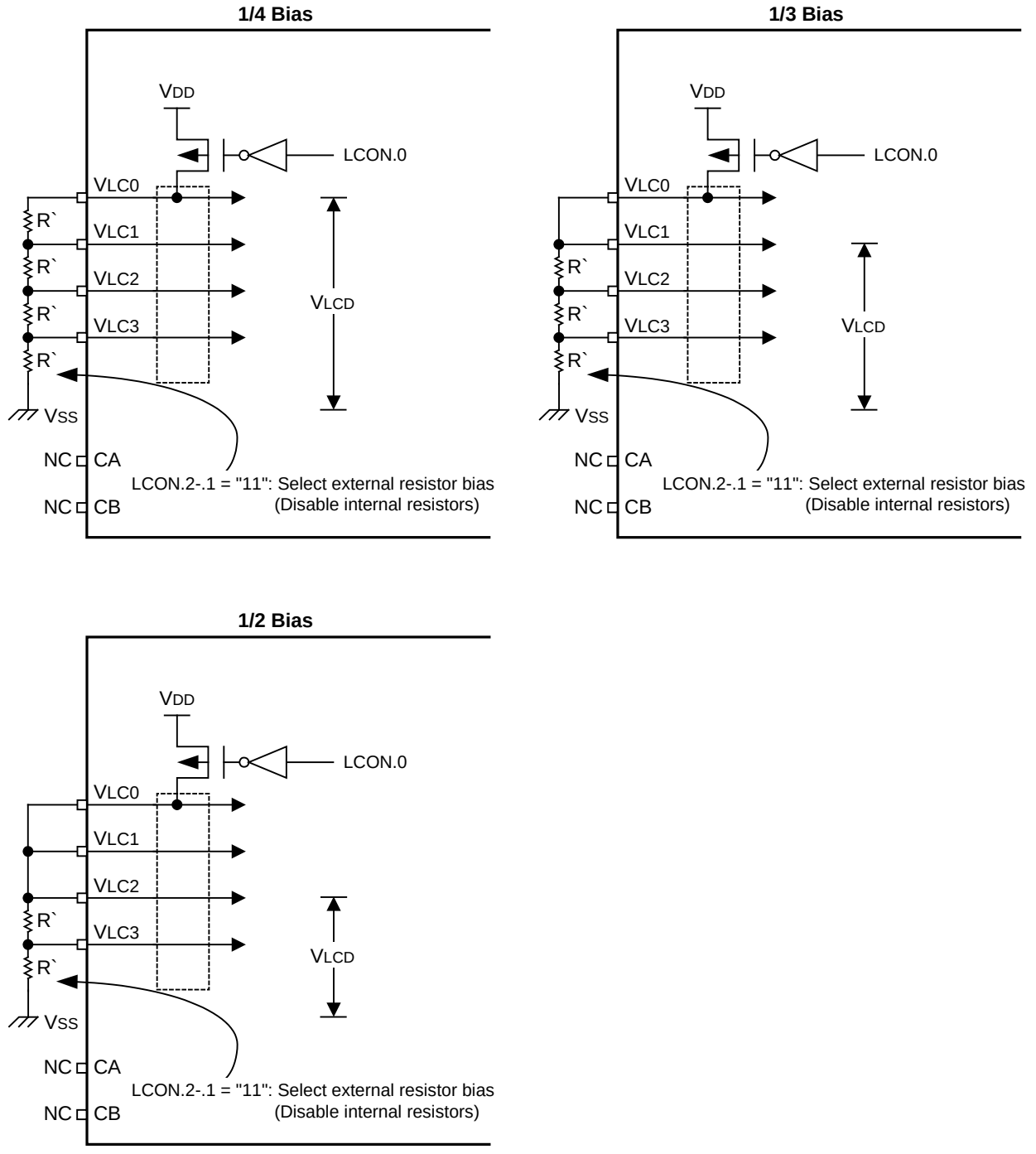
The S3F8S7B microcontroller can directly drive an up-to-352-dot (44 segments x 8 commons) LCD panel. Its LCD block has the following components:

- LCD controller/driver
- Display RAM (30H–61H) for storing display data in page 8
- 6 common/segment output pins (COM2/SEG0–COM7/SEG5)
- 44 segment output pins (SEG6–SEG49)
- 2 common output pins (COM0–COM1)
- Four LCD operating power supply pins (V_{LC0}–V_{LC3})
- LCD bias by voltage booster
- LCD bias by voltage dividing resistors

The LCD control register, LCON, is used to turn the LCD display on and off, select LCD clock frequency, LCD duty and bias, and LCD bias type. The LCD mode control register, LMOD, is used to selects VLCD voltage. Data written to the LCD display RAM can be automatically transferred to the segment signal pins without any program control. When a subsystem clock is selected as the LCD clock source, the LCD display is enabled even in the main clock stop or idle modes.

LCD data stored in the display RAM locations are transferred to the segment signal pins automatically without program control.

15.7 External Resistor Bias Pin Connection



NOTES:

1. The CA and CB pins are not connected when the external resistor bias is selected.
2. V_{LC0} and V_{LC1} should be connected at 1/3 bias.
3. V_{LC0}, V_{LC1}, and V_{LC2} should be connected at 1/2 bias.

Figure 15-7 External Resistor Bias Pin Connection

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

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Supply current (1)	I_{DD1} (2)	Run mode: V_{DD} = 5.0 V Crystal oscillator C1 = C2 = 22 pF	12.0 MHz	–	2.2	4.0	mA
			4.2 MHz	–	1.2	2.0	
		V_{DD} = 3.0 V	4.2 MHz	–	0.8	1.5	
	I_{DD2} (2)	Idle mode: V_{DD} = 5.0 V Crystal oscillator C1 = C2 = 22 pF	12.0 MHz	–	1.3	2.3	
			4.2 MHz	–	0.8	1.5	
		V_{DD} = 3.0V	4.2 MHz	–	0.4	0.8	
	I_{DD3} (3)	Sub Operating mode: V_{DD} = 3.0 V , T_A = 25 °C 32 kHz crystal oscillator		–	80.0	120.0	µA
		Run mode: V_{DD} = 3.0 V, T_A = 25 °C 32 kHz crystal oscillator C = 0.1 µF, No panel load Cap bias LCD on		–	85.0	140.0	
	I_{DD4} (3)	Sub Idle mode: V_{DD} = 3.0 V, T_A = 25 °C 32 kHz crystal oscillator		–	6.0	15.0	
		Idle mode: V_{DD} = 3.0 V, T_A = 25 °C 32 kHz crystal oscillator C = 0.1 µF, No panel load Cap bias LCD on		–	10.0	20.0	
	I_{DD5} (4)	Stop mode: V_{DD} = 5.0 V, T_A = 25 °C		–	0.3	3.0	
		Stop mode: V_{DD} = 5.0 V, T_A = +85 °C		–	3.0	8.0	
		Stop mode: V_{DD} = 5.0 V, T_A = -40 to +85 °C		–	–	8.0	

NOTE: It is middle output voltage when the V_{DD} and V_{LC0} pin are connected.

1. Supply current does not include current drawn through internal pull-up resistors, LCD voltage dividing resistors, the LVR block, and external output current loads.
2. I_{DD1} and I_{DD2} include a power consumption of sub clock oscillation.
3. I_{DD3} and I_{DD4} are the current when the main clock oscillation stops and the sub clock is used.
4. I_{DD5} is the current when the main and sub clock oscillation stops.
5. Every value in this table is measured when bits 4-3 of the system clock control register (CLKCON.4–.3) is set to 11B.

23 Mechanical Data

23.1 Overview

The S3F8S7B microcontroller is currently available in 80-pin QFP and 80-pin TQFP, and 80-pin LQFP packages.

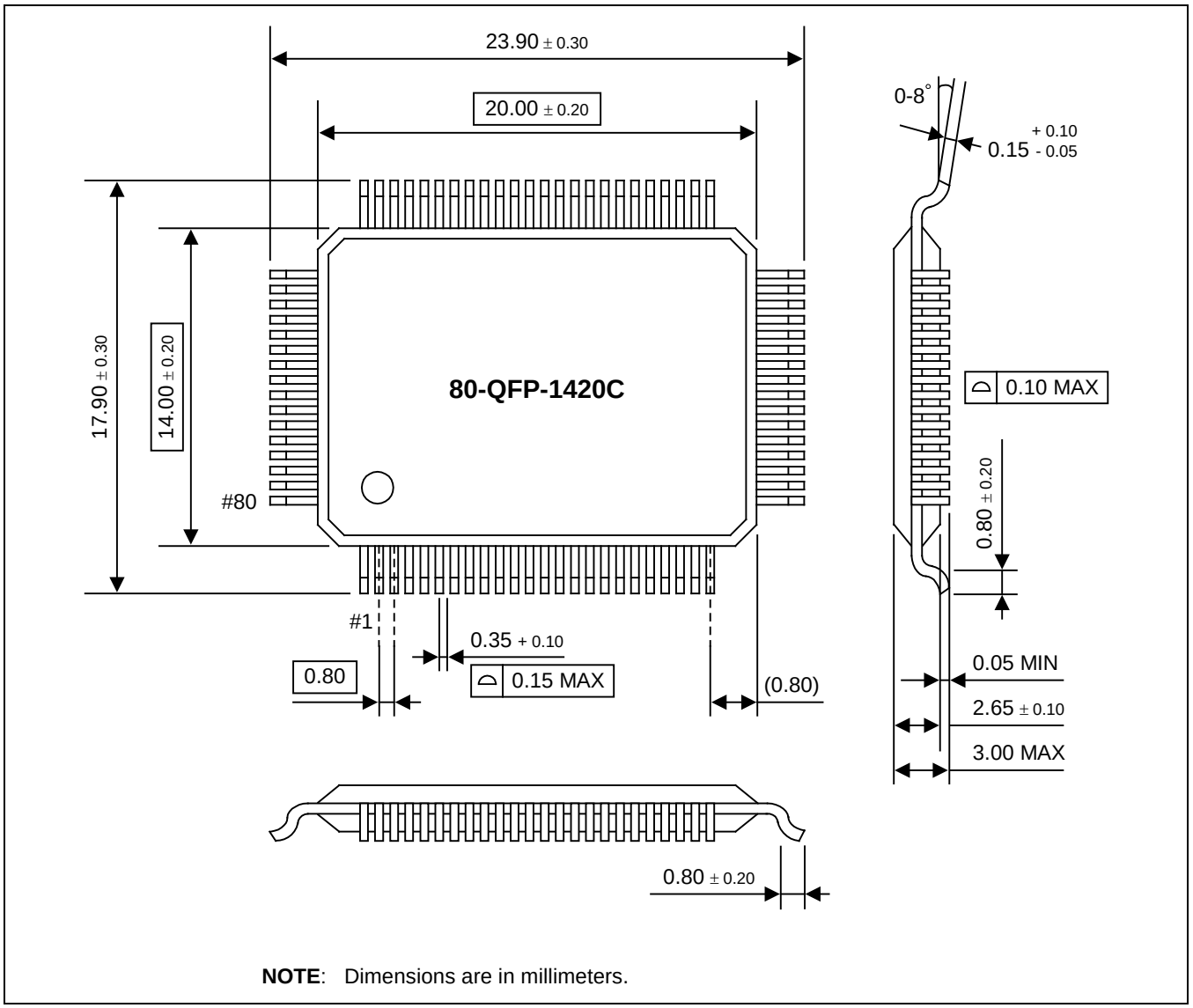


Figure 23-1 Package Dimensions (80-QFP)