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	Obsolete
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-BQFP
Supplier Device Package	80-QFP
Purchase URL	https://www.e-xfl.com/product-detail/zilog/s3f8s7bxzz-qw8b

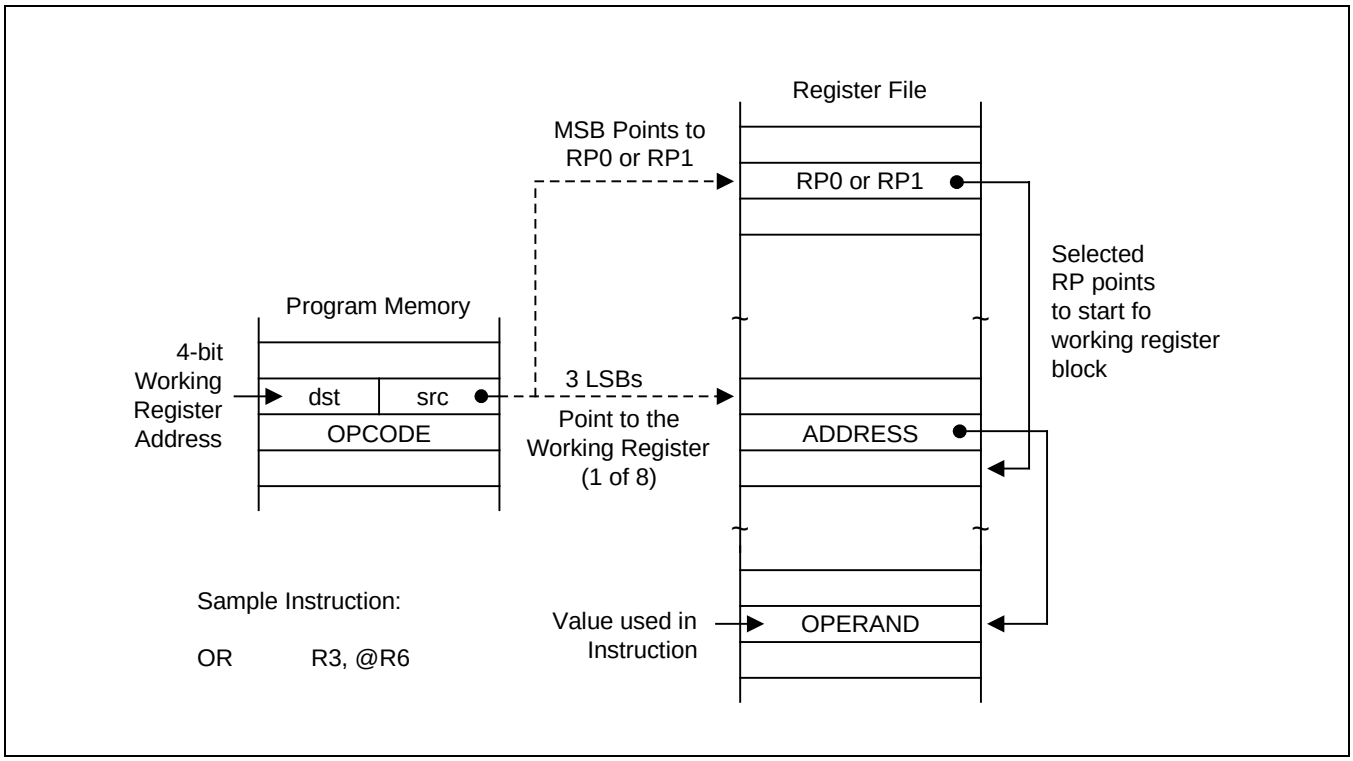


Figure 3-5 Indirect Working Register Addressing to Register File

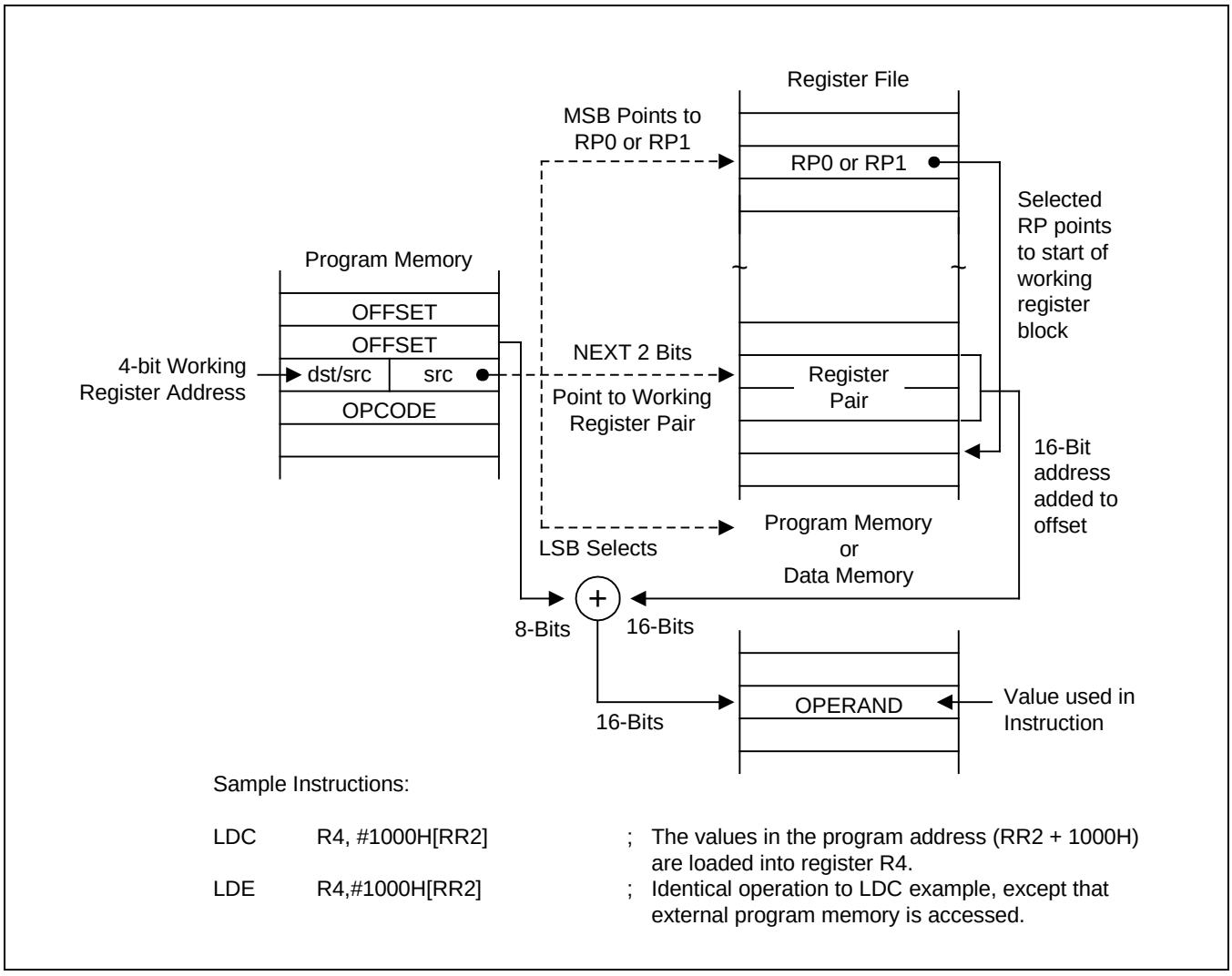


Figure 3-9 Indexed Addressing to Program or Data Memory

4.1.12 IPH: Instruction Pointer (High Byte) (DAH, Set 1)

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

.7–.0 Instruction Pointer Address (High Byte)

The high-byte instruction pointer value is the upper eight bits of the 16-bit instruction pointer address (IP15–IP8). The lower byte of the IP address is located in the IPL register (DBH).

4.1.13 IPL: Instruction Pointer (Low Byte) (DBH, Set 1)

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

.7–.0 Instruction Pointer Address (Low Byte)

The low-byte instruction pointer value is the lower eight bits of the 16-bit instruction pointer address (IP7–IP0). The upper byte of the IP address is located in the IPH register (DAH).

4.1.31 P3CONM: Port 3 Control Register (Middle Byte) (ECH, 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 P3.4/TD0CLK/PG4/SEG24 Configuration Bits

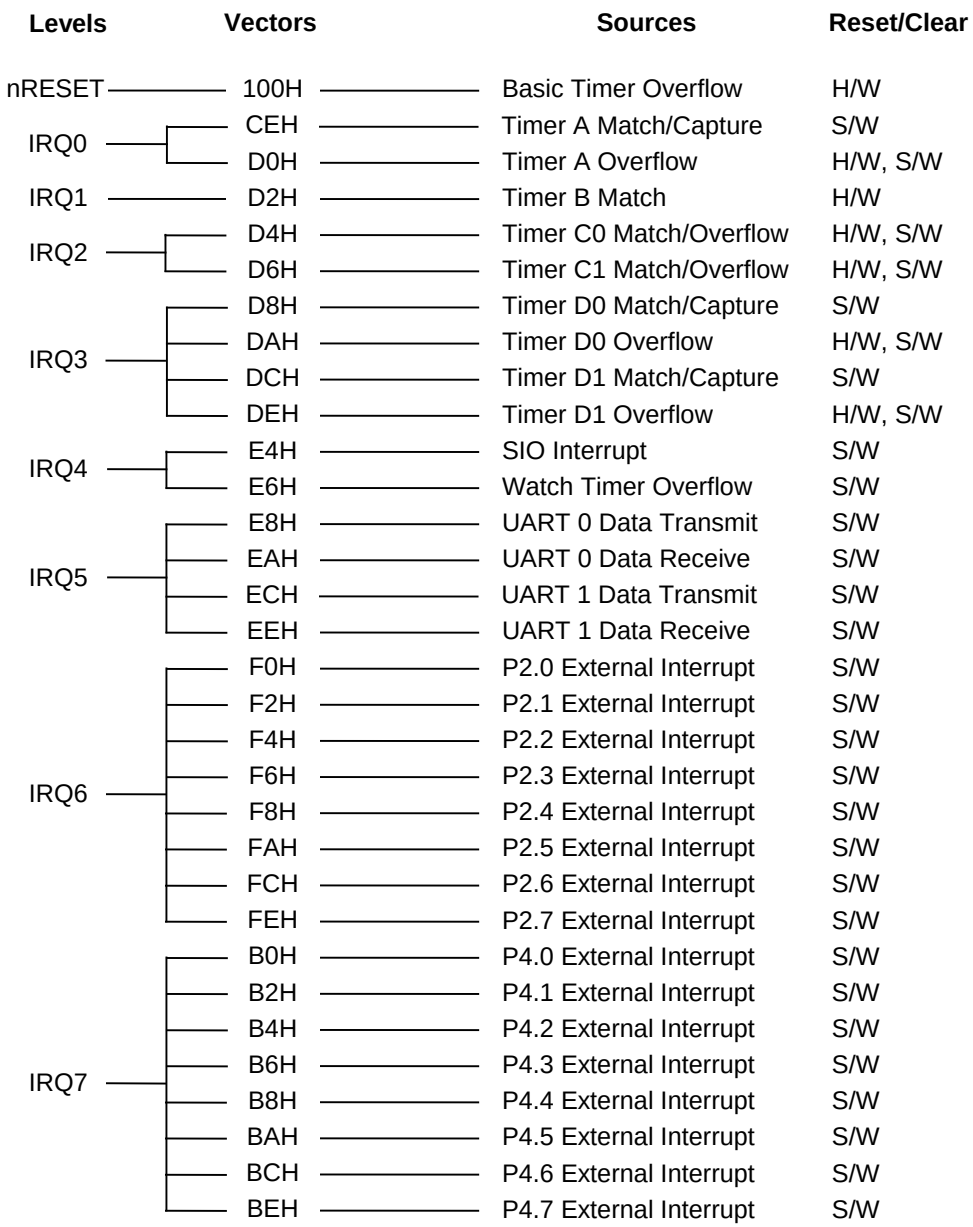
0	0	Input mode (TD0CLK)
0	1	Alternative function (PG4)
1	0	Alternative function (LCD signal)
1	1	Output mode

.5–.3 P3.3/TD0OUT/TD0PWM/TD0CAP/PG3/SEG23 Configuration Bits

0	0	0	Input mode (TD0CAP)
0	0	1	Alternative function (PG3)
0	1	0	Alternative function (LCD signal)
0	1	1	Output mode
1	x	x	Alternative function (TD0OUT/TD0PWM)

.2–.0 P3.2/TC1OUT/TC1PWM/PG2/SEG22 Configuration Bits

0	0	0	Input mode
0	0	1	Alternative function (PG2)
0	1	0	Alternative function (LCD signal)
0	1	1	Output mode
1	x	x	Alternative function (TC1OUT/TC1PWM)



NOTES:

1. Within a given interrupt level, the low vector address has high priority. For example, CEH has higher priority than D0H within the level IRQ0 the priorities within each level are set at the factory.
2. External interrupts are triggered by a rising or falling edge, depending on the corresponding control register setting.

Figure 5-2 S3F8S7B Interrupt Structure

6.5.29 IDLE-Idle Operation

IDLE

Operation:

The IDLE instruction stops the CPU clock while allowing system clock oscillation to continue. Idle mode can be released by an interrupt request (IRQ) or an external reset operation.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst	Addr Mode src
opc	1	4	6F	–	–

Example: The instruction

```
IDLE
```

stops the CPU clock but not the system clock.

6.5.55 RL-Rotate Left

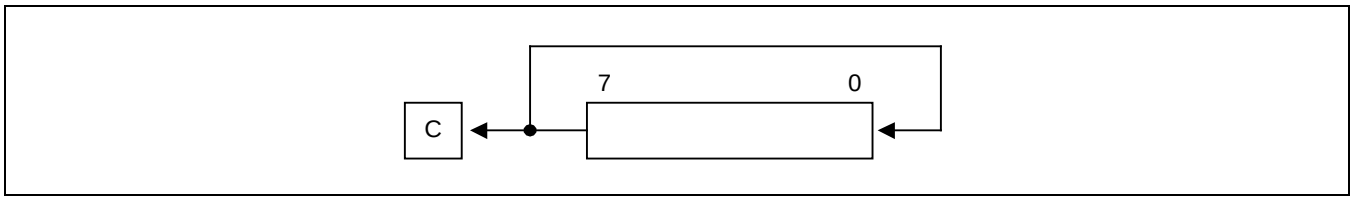
RL dst

Operation: C ← dst (7)

dst (0) ← dst (7)

dst (n + 1) ← dst (n), n = 0–6

The contents of the destination operand are rotated left one bit position. The initial value of bit 7 is moved to the bit zero (LSB) position and also replaces the carry flag.



Flags:

C: Set if the bit rotated from the most significant bit position (bit 7) was "1".

Z: Set if the result is "0"; cleared otherwise.

S: Set if the result bit 7 is set; cleared otherwise.

V: Set if arithmetic overflow occurred; cleared otherwise.

D: Unaffected.

H: Unaffected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode dst
opc | dst	2	4	90	R
		4	91	IR

Examples: Given: Register 00H = 0AAH, register 01H = 02H and register 02H = 17H:

```
RL      00H     →       Register 00H = 55H, C = "1"
RL      @01H    →       Register 01H = 02H, register 02H = 2EH, C = "0"
```

In the first example, if general register 00H contains the value 0AAH (10101010B), the statement "RL 00H" rotates the 0AAH value left one bit position, leaving the new value 55H (01010101B) and setting the carry and overflow flags.

6.5.64 SRP/SRP0/SRP1-Set Register Pointer

SRP src

SRP0 src

SRP1 src

Operation:

If src (1) = 1 and src (0) = 0 then: RP0 (3–7) ←□ src (3–7)
If src (1) = 0 and src (0) = 1 then: RP1 (3–7) ←□ src (3–7)
If src (1) = 0 and src (0) = 0 then: RP0 (4–7) ←□ src (4–7),

RP0 (3) □ ← 0
RP1 (4–7) ←□ src (4–7),
RP1 (3) □ ← 1

The source data bits one and zero (LSB) determine whether to write one or both of the register pointers, RP0 and RP1. Bits 3–7 of the selected register pointer are written unless both register pointers are selected. RP0.3 is then cleared to logic zero and RP1.3 is set to logic one.

Flags: No flags are affected.

Format:

	Bytes	Cycles	Opcode (Hex)	Addr Mode src
opc | src	2	4	31	IM

Examples: The statement

```
SRP #40H
```

sets register pointer 0 (RP0) at location 0D6H to 40H and register pointer 1 (RP1) at location 0D7H to 48H.

The statement "SRP0 #50H" sets RP0 to 50H, and the statement "SRP1 #68H" sets RP1 to 68H.

7.3 Sub Oscillator Circuits

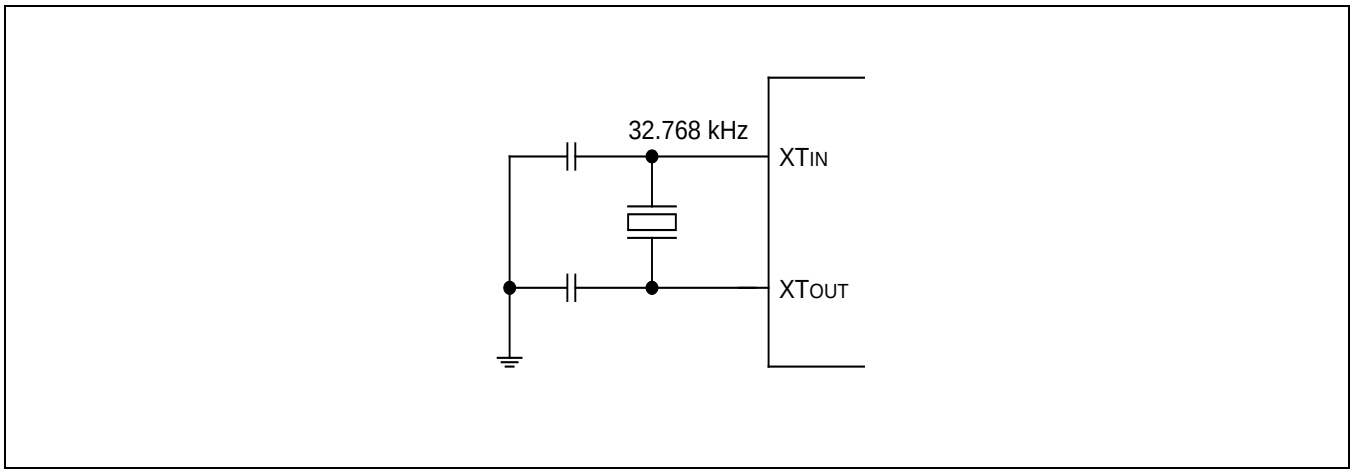


Figure 7-4 Crystal Oscillator (fxt)

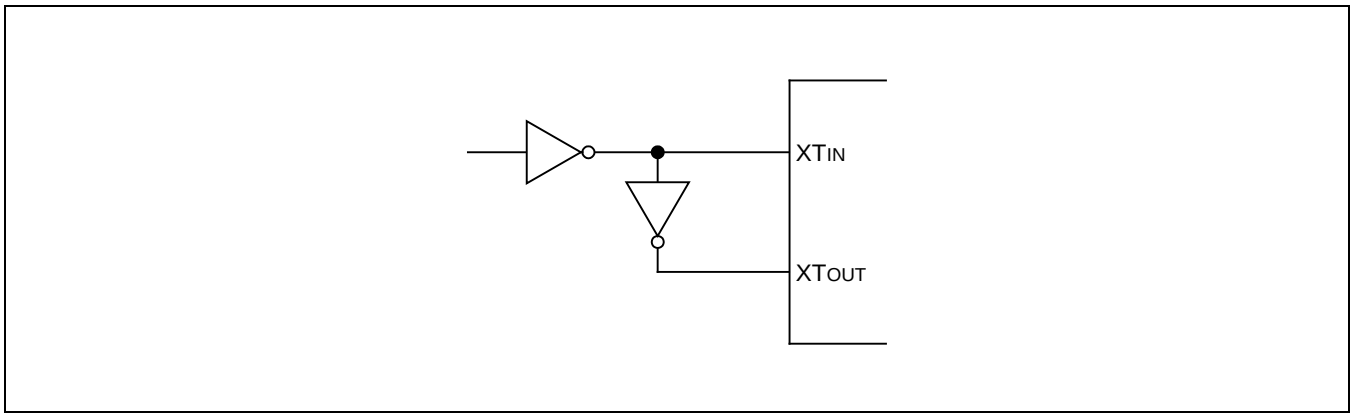


Figure 7-5 External Oscillator (fxt)

```
LD      STPCON,#10100101B
STOP
NOP
NOP
NOP
```

9 I/O Ports

9.1 Overview

The S3F8S7B microcontroller has nine bit-programmable I/O ports, P0–P8. The port 8 is a 6-bit port and the others are 8-bit ports. This gives a total of 70 I/O pins. Each port can be flexibly configured to meet application design requirements. The CPU accesses ports by directly writing or reading port registers. No special I/O instructions are required.

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)

11.1.2 Timer A Control Register (TACON)

You use the timer A control register, TACON, to

- Select the timer A operating mode (interval timer, capture mode, or PWM mode)
- Select the timer A input clock frequency
- Clear the timer A counter, TACNT
- Enable the timer A overflow interrupt or timer A match/capture interrupt

TACON is located in set 1, Bank 0 at address E2H, and is read/write addressable using Register addressing mode.

A reset clears TACON to '00H'. This sets timer A to normal interval timer mode, selects an input clock frequency of fxx/1024, and disables all timer A interrupts. You can clear the timer A counter at any time during normal operation by writing a "1" to TACON.2.

The timer A overflow interrupt (TAOVF) is interrupt level IRQ0 and has the vector address D0H. When a timer A overflow interrupt occurs and is serviced by the CPU, the pending condition is cleared automatically by hardware or must be cleared by software.

To enable the timer A match/capture interrupt (IRQ0, vector CEH), you must write TACON.1 to "1". To detect a match/capture interrupt pending condition, the application program polls INTPND.1. When a "1" is detected, a timer A match or capture interrupt is pending. When the interrupt request has been serviced, the pending condition must be cleared by software by writing a "0" to the timer A match/capture interrupt pending bit, INTPND.1.

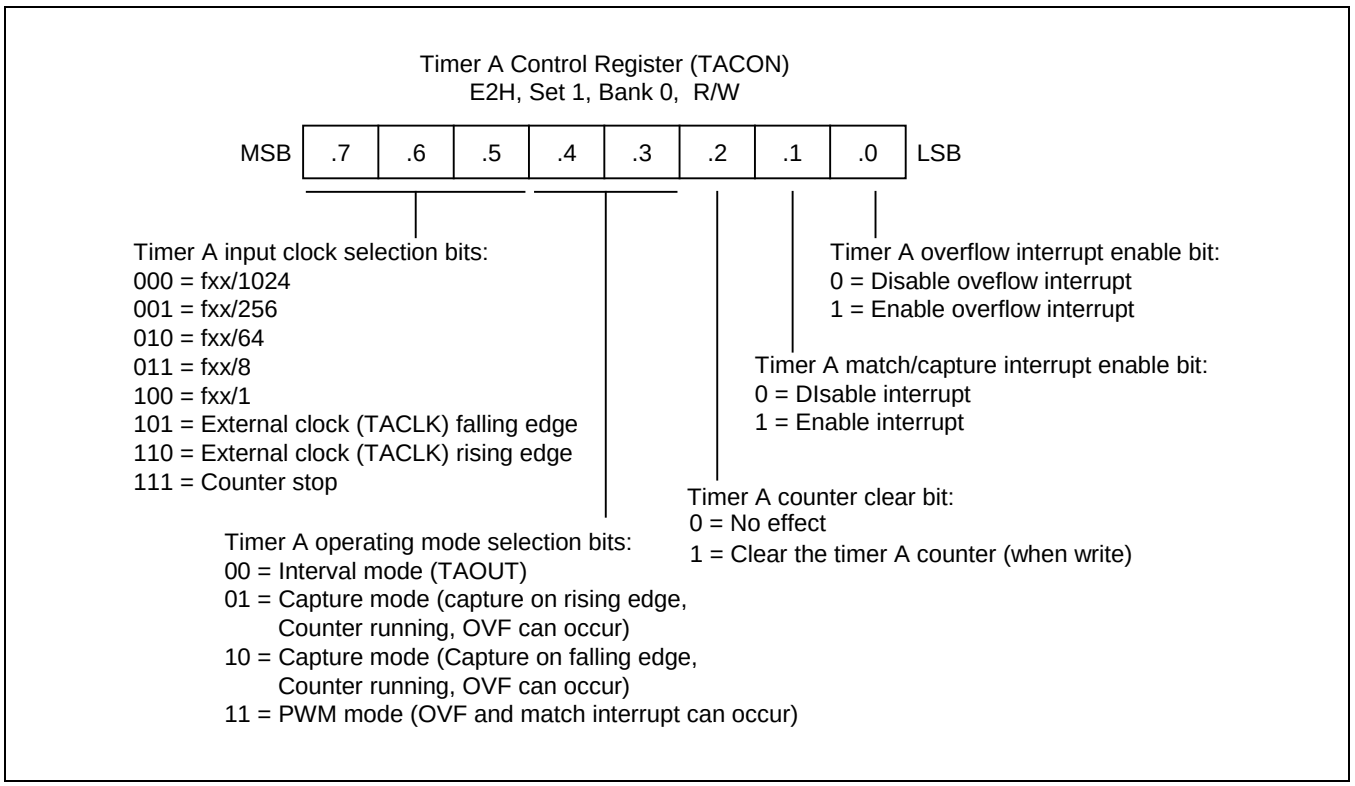


Figure 11-1 Timer A Control Register (TACON)

11.1.3.3 Pulse Width Modulation Mode

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

Although you can use the match signal to generate a timer A overflow interrupt, interrupts are not typically used in PWM-type applications. Instead, the pulse at the TAPWM pin is held to Low level as long as the reference data value is less than or equal to (≤) the counter value and then the pulse is held to High level for as long as the data value is greater than (>) the counter value. One pulse width is equal to $t_{CLK} \times 256$ (Refer to *Figure 11-3*).

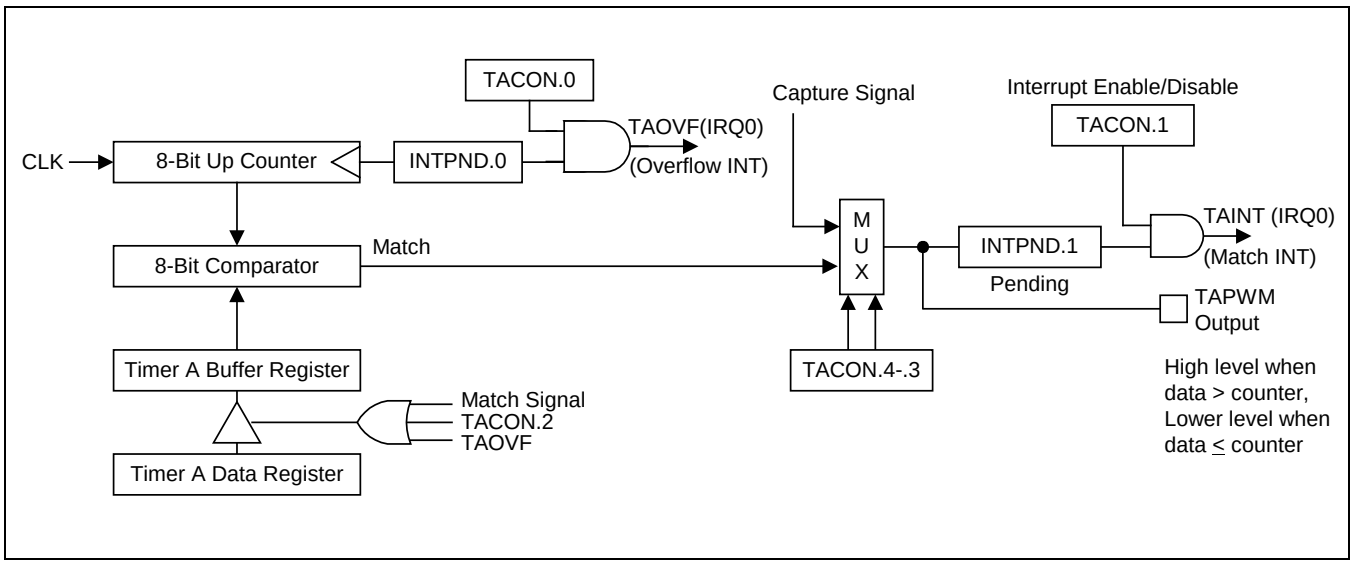


Figure 11-3 Simplified Timer A Function Diagram: PWM Mode

13.1.2 Timer D0 Control Register (TD0CON)

You use the timer D0 control register, TD0CON, to

- Select the timer D0 operating mode (interval timer, capture mode, or PWM mode)
- Select the timer D0 input clock frequency
- Clear the timer D0 counter, TD0CNTH/T D0CNTL
- Enable the timer D0 overflow interrupt or timer D0 match/capture interrupt

TD0CON is located in set 1 and bank 1 at address FAH, and is read/write addressable using Register addressing mode.

A reset clears TD0CON to "00H". This sets timer D0 to normal interval timer mode, selects an input clock frequency of fxx/1024, and disables all timer D0 interrupts. To disable the counter operation, please set TD0CON.7–.5 to 111B. You can clear the timer D0 counter at any time during normal operation by writing a "1" to TD0CON.2.
The timer D0 overflow interrupt (TD0OVF) is interrupt level IRQ3 and has the vector address DAH. When a timer D0 overflow interrupt occurs and is serviced interrupt (IRQ3, vector DAH), you must write TD0CON.0 to "1". When a timer D0 overflow interrupt occurs and is serviced by the CPU, the pending condition is cleared automatically by hardware or must be cleared by software.

To enable the timer D0 match/capture interrupt (IRQ3, vector D8H), you must write TD0CON.1 to "1". To detect a match/capture interrupt pending condition, the application program polls INTPND.3. When a "1" is detected, a timer D0 match or capture interrupt is pending. When the interrupt request has been serviced, the pending condition must be cleared by software by writing a "0" to the timer D0 match/capture interrupt pending bit, INTPND.3.

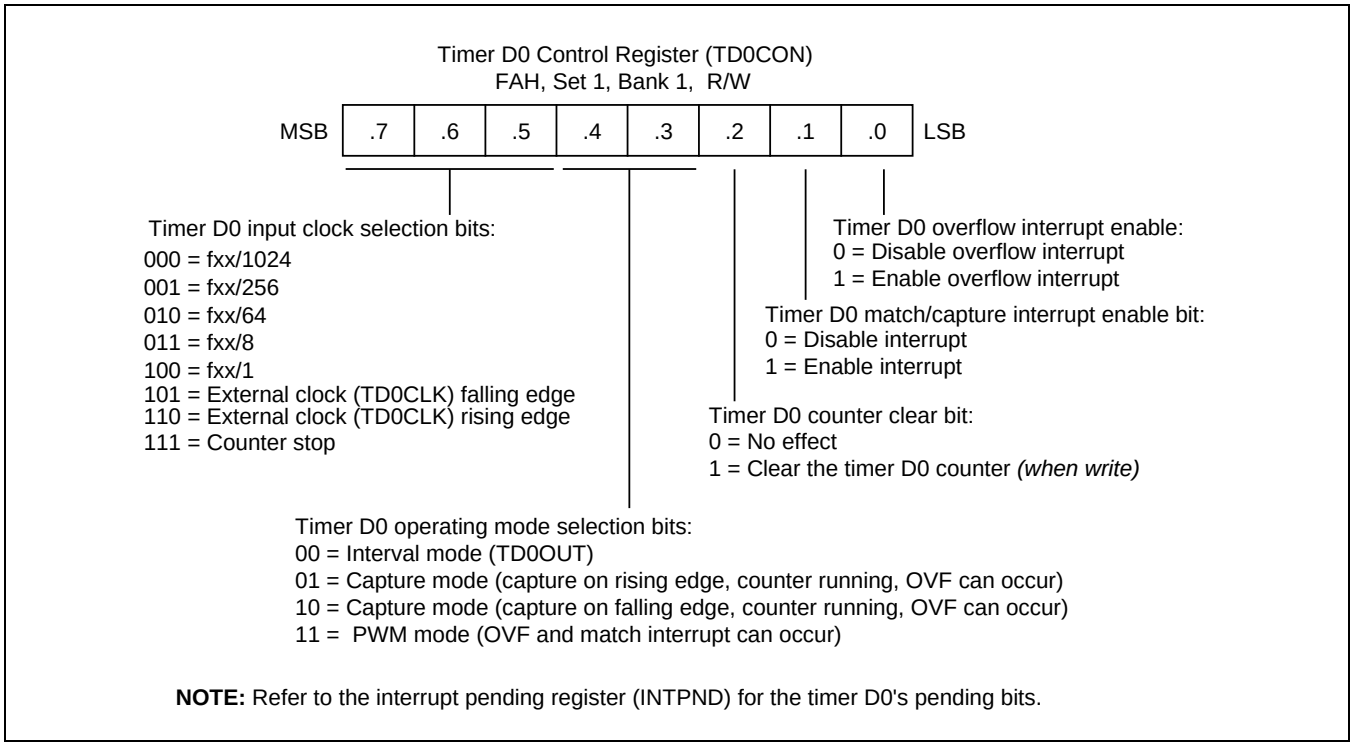


Figure 13-1 Timer D0 Control Register (TD0CON)

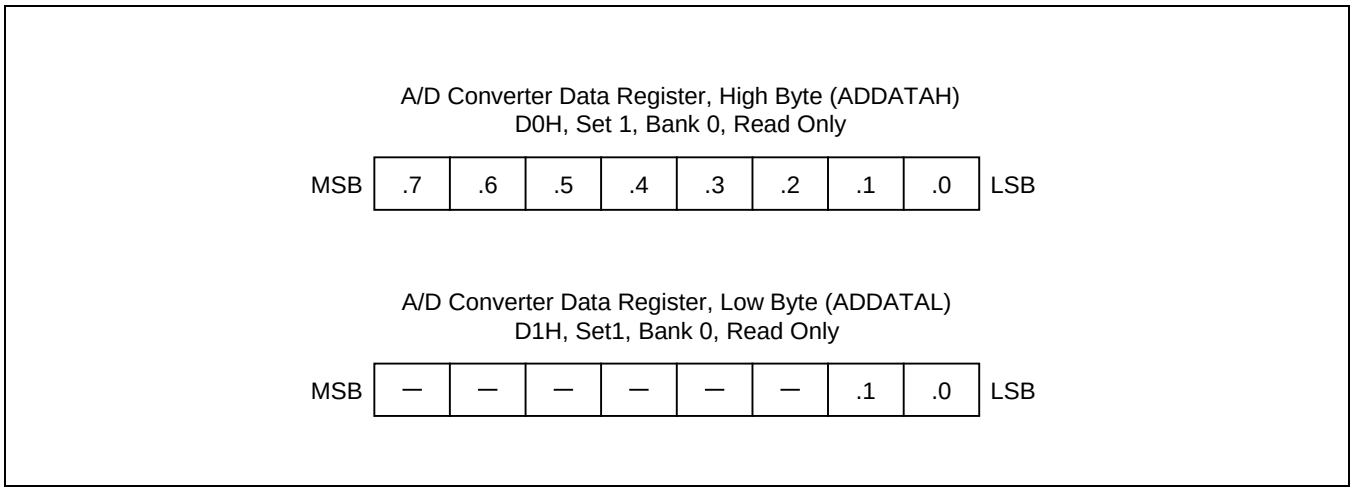


Figure 16-2 A/D Converter Data Register (ADDATAH/L)

19.13.2 Mode 3 Receive Procedure

1. Select the UART 0 clock, UART0CONL.3 and .2.
2. Select the UART 0 transmit parity-bit autogeneration enable or disable (UART0CONL.7).
3. Select mode 3 and set the RE (Receive Enable) bit in the UART0CONH register to "1".
4. The receive operation will be started when the signal at the RxD0 (P1.2) pin goes to low level.

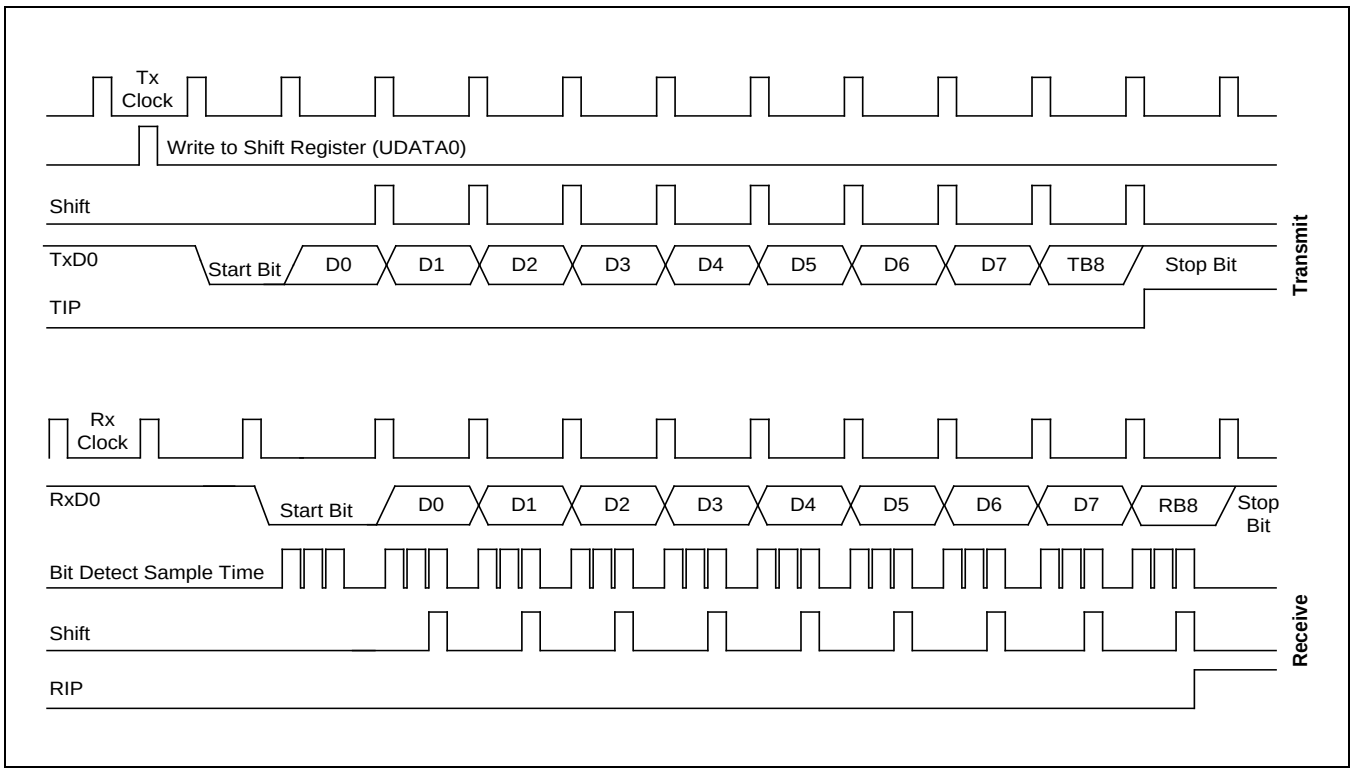


Figure 19-9 Timing Diagram for Serial Port Mode 3 Operation

19.14 Serial Communication for Multiprocessor Configurations

The S3F8-series multiprocessor communication features lets a "master" S3F8S7B send a multiple-frame serial message to a "slave" device in a multi- S3F8S7B configuration. It does this without interrupting other slave devices that may be on the same serial line.

This feature can be used only in UART Modes 2 or 3. In these modes 2 and 3, 9 data bits are received. The 9th bit value is written to RB8 (UART0CONH.2). The data receive operation is concluded with a stop bit. You can program this function so that when the stop bit is received, the serial interrupt will be generated only if RB8 = "1".

To enable this feature, you set the MCE bit in the UART0CONH register. When the MCE bit is "1", serial data frames that are received with the 9th bit = "0" do not generate an interrupt. In this case, the 9th bit simply separates the address from the serial data.

20 Pattern Generation Module

20.1 Overview

20.1.1 Pattern Generation Flow

You can output up to 8-bit through P3.0–P3.7 by tracing the following sequence. First of all, you have to change the PGDATA into what you want to output. And then you have to set the PGCON to enable the pattern generation module and select the triggering signal. From now, bits of PGDATA are on the P3.0–P3.7 whenever the selected triggering signal happens.

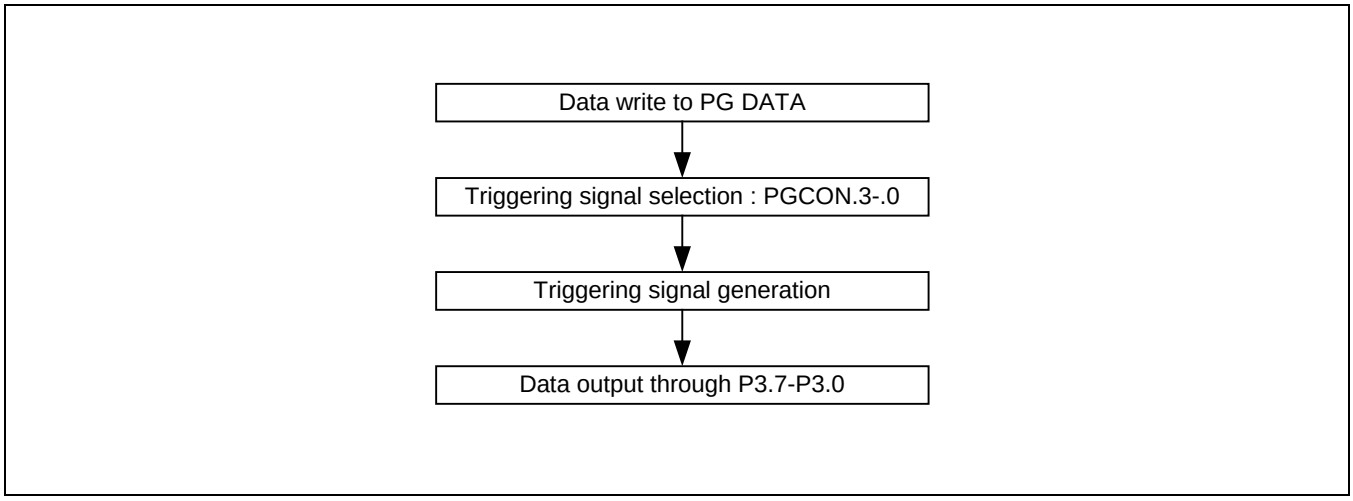
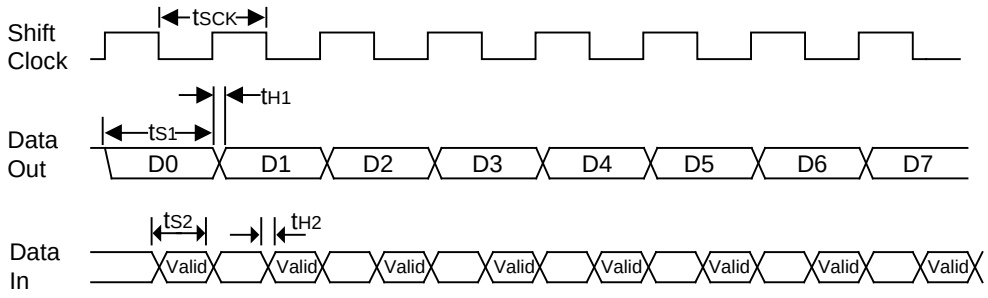


Figure 20-1 Pattern Generation Flow



NOTE: The symbols shown in this diagram are defined as follows:

f_{SCK}	Serial port clock cycle time
t_{S1}	Output data setup to clock rising edge
t_{S2}	Clock rising edge to input data valid
t_{H1}	Output data hold after clock rising edge
t_{H2}	Input data hold after clock rising edge

Figure 22-8 Timing Waveform for the UART Module

25.3 Third Parties for Development Tools

Zilog provides a complete line of development tools that support the S3 Family of Microcontrollers. With long experience in developing MCU systems, these third party firms are bonafide leaders in MCU development tool technology.

25.3.1 In-Circuit Emulators

- OPENice-i500/2000
- SK-1200 SmartKit

25.3.2 OTP/MTP Programmers

- GW-Uni2
- AS-Pro2
- Elnec programmers

To obtain the S3 Family development tools that will satisfy your S3F80QB development objectives, contact your local Zilog Sales Office, or visit Zilog's Third Party Tools page to review our list of third party tool suppliers.