Z8051 Series 8-Bit Microcontrollers

Z51F0410

Product Specification

PS029502-0212

PRELIMINARY

www.zilog.com

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8051 is a trademark or registered trademark of Zilog, Inc. All other product or service names are the property of their respective owners.

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

Date	Revision Level	Description	Page
Feb 2012	02	Removed references to SOP, PDIP packages.	All
Jan 2012	01	Original Zilog issue.	All

Table of Contents

List of Figures

List of Tables

Z51F0410

CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH 12-BIT A/D CONVERTER

1. OVERVIEW

1.1 Description

The Z51F0410 MCU is advanced CMOS 8-bit microcontroller with 4K bytes of Flash. This is powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications. This provides the following features : 4K bytes of Flash, 256 bytes of RAM, 256 bytes of Data EEPROM, general purpose I/O, 8/16-bit timer/counter, watchdog timer, watch timer, USART, BUZZER, I2C, on-chip POR, 12-bit A/D converter, analog comparator, 10-bit PWM output, on-chip oscillator and clock circuitry. The Z51F0410 MCU also supports power saving modes to reduce power consumption.

Device Name	Flash	RAM	ADC	I/O PORT	Package
Z51F0410	4K bytes	256 bytes	8 channel	8	10 SSOP

1.2 Features

- **CPU**
  - 8 Bit CISC Core (8051 Compatible,2 clock per cycle)
- **4K Bytes On-chip Flash**
  - Endurance : 10,000 times
  - Retention : 10 years
  - Code Encryption (Super Lock)
- **256 Bytes SRAM**
- **256 Bytes Data EEPROM**
  - mapped in XDATA region
  - Endurance : 100,000 times
  - Retention : 10 years
- **General Purpose I/O**
  - 8 Ports (P0[7:0])
- **One Basic Interval Timer**

  -Reset Release Time - 16ms, 32ms, 64ms Configurable
- **Timer/ Counter**
  - 8Bit×2ch(16Bit×1ch)
  - 16bit x 1ch
- **10-bit PWM (Using Timer1)**
  - 10Bitx1ch
- **Watch Dog Timer**
- **Watch Timer**
- **USART**
- **BUZZER**
- **I2C**
- **12 Bit A/D Converter**
  - 8 Input channels
- **Analog Comparator**
  - On Chip Analog Comparator with ACOUT
- **Interrupt Sources**
  - External (2)

- Pin Change Interrupt(P0) (1)
- USART (2)
- Timer (3)
- I2C (1)
- ADC (1)
- ACOM (1)
- WDT (1)
- WT (1)
- BIT (1)
- Data EEPROM(1)

- **On-Chip RC-Oscillator**
  - 8MHz(±1%) @ 25℃
  - 128KHz(±50%)
- **Power On Reset**
  - 1.4V
- **Programmable Brown-Out Detector**
  - 4 level Selectable
- **Reconfigurable Secondary Pin Function**
- **Minimum Instruction Execution Time**
  - 250ns (@8MHz, NOP Instruction)
- **Power down mode**
  - IDLE, STOP1, STOP2 mode
- **Sub-Active mode**
  - System used external 32.768KHz crystal or system used internal 128KHz oscillator
- **Operating Frequency**
  - 1MHz ~ 8MHz
- **Operating Voltage**
  - 1.8V ~ 5.5V (@ 1~8MHz)
  - 2.2V ~ 5.5V (@ 1~8Mhz, Using Comparator)
- **Operating Temperature : -40 ~ +85**℃
- **Package Type**
  - 10-pin SSOP
  - Pb free package

1.3 Ordering Information

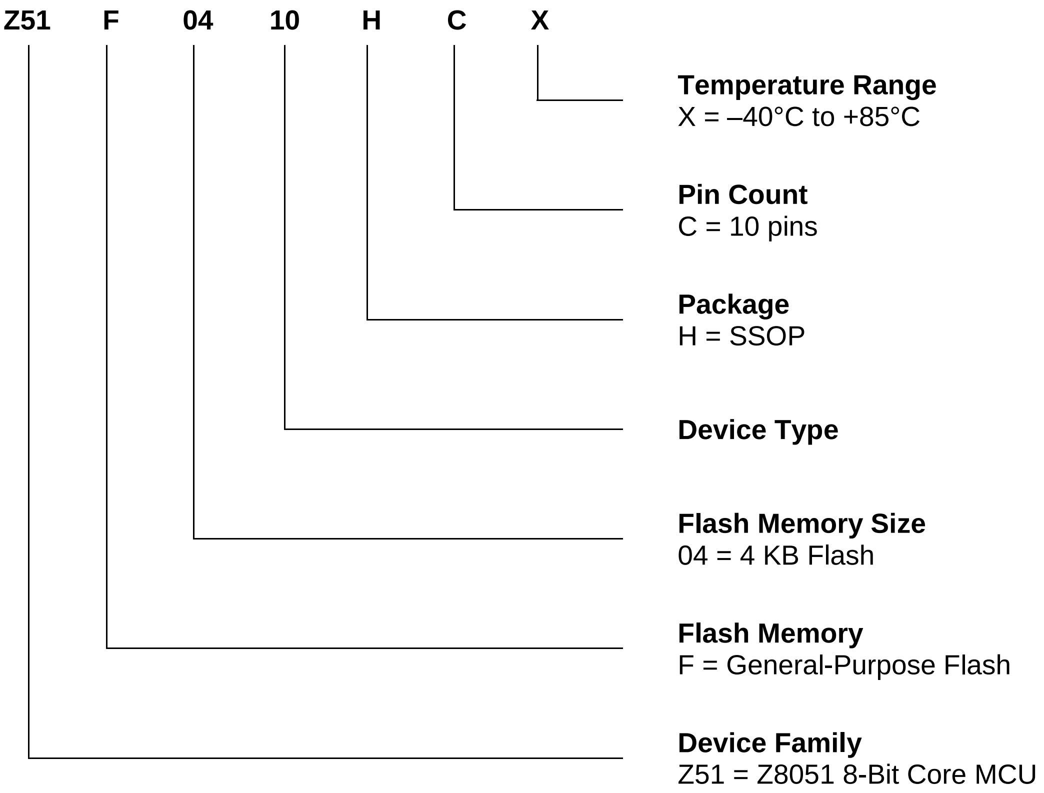
Table 1.1 Ordering Information of the Z51F0410 MCU

Device Name	ROM Size	RAM Size	EEPROM Size	Package
Z51F0410HCX	4K bytes Flash	256 bytes	256 bytes	10SSOP

1.3.1 Part Number Suffix Designation

Zilog part numbers consist of a number of components, as indicated in the following example.

Example: Part number Z51F0410HCX is an 8-bit MCU with 4 KB of Flash memory and 256 bytes of RAM in a 10-pin SSOP package and operating within a –40°C to +85°C temperature range. In accordance with RoHS standards, this device has been built using lead-free solder.



Z51 F 04 10 H C X

Temperature Range
X = –40°C to +85°C

Pin Count
C = 10 pins

Package
H = SSOP

Device Type

Flash Memory Size
04 = 4 KB Flash

Flash Memory
F = General-Purpose Flash

Device Family
Z51 = Z8051 8-Bit Core MCU

1.4 Development Tools

1.4.1 Compiler

Zilog does not provide any compiler for the Z51F0410 MCU. But the CPU core of the Z51F0410 MCU is Mentor 8051, you can use all kinds of third party's standard 8051 compiler like Keil C Compiler, Open Source SDCC (Small Device C Compiler) .. These compilers' output debug information can be integrated with our OCD emulator and debugger. Refer to OCD manual for more details.

1.4.2 OCD Emulator and Debugger

The OCD (On Chip Debug) emulator supports Zilog's 8051 series MCU emulation. The OCD interface uses two wires interfacing between PC and MCU which is attached to user's system. The OCD can read or change the value of MCU's internal memory and I/O peripherals. And also the OCD controls MCU's internal debugging logic, it means OCD controls emulation, step run, monitoring, etc.

The OCD debugger program works on Microsoft-Windows NT, 2000, XP, Vista(32-bit) operating system.

If you want to see details more, please refer to OCD debugger manual. You can download debugger S/W and manual from out web-site.

The connection pins between PC and MCU is as follows:

SCLK (P0[3] of Z51F0410)

SDATA (P0[5] of Z51F0410)

OCD connector diagram: Connect OCD and user system

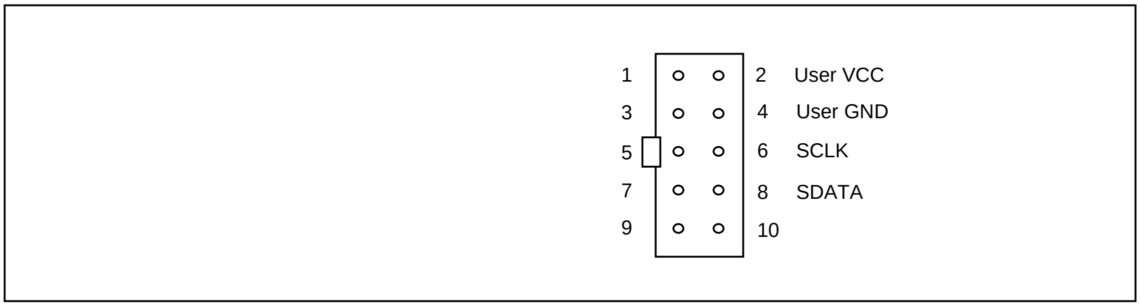


Figure 1.1 OCD Debugger and Pin Configuration

Note: P0[3] is pulled-up by external resistor to avoid malfunction when power is on.

1.5 Block Diagram

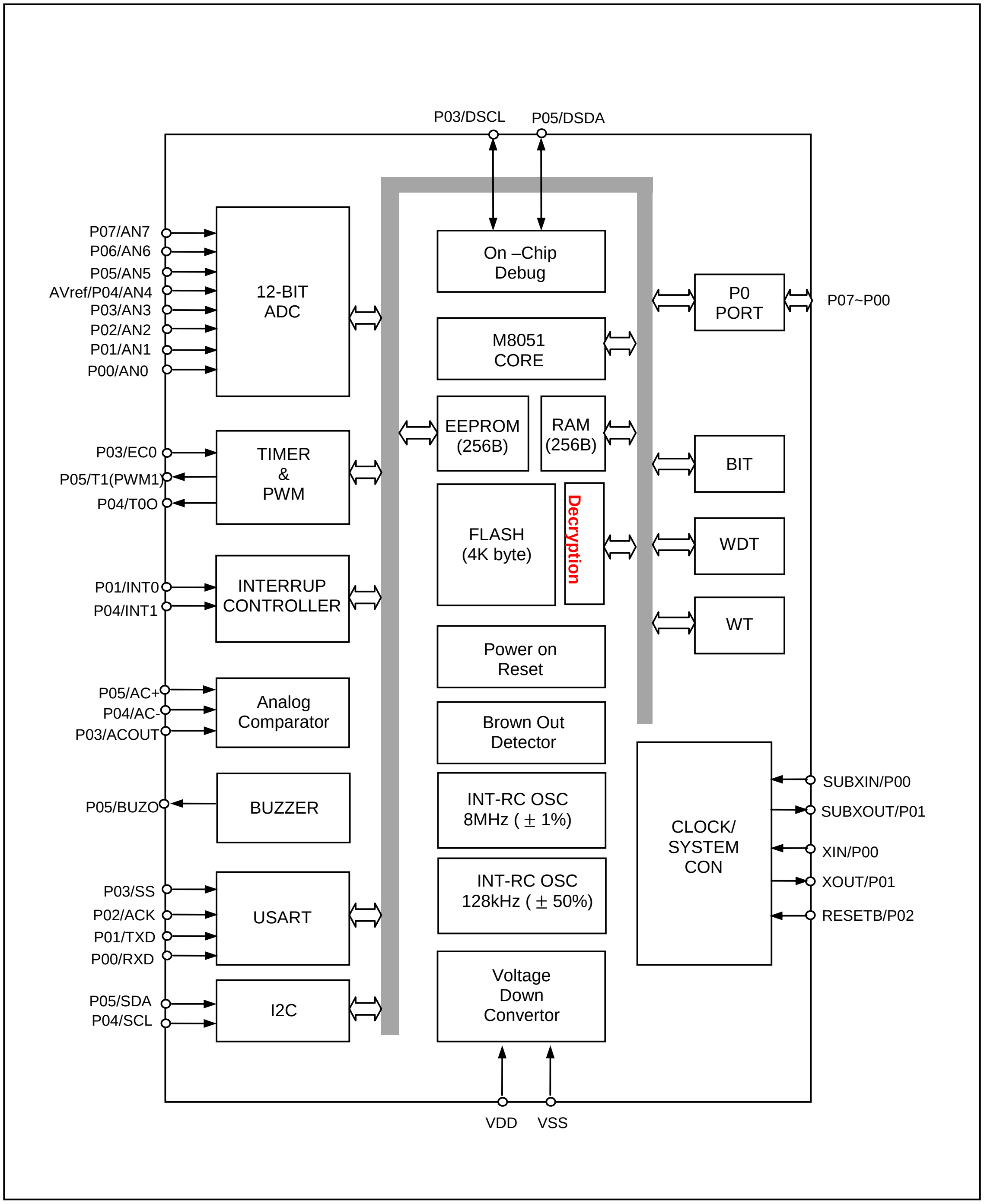


Figure 1.2 Top Abstract Block Diagram

Note) P01, P02, P03 pin are programmable or configure option selectable.

1.6 PIN Assignment

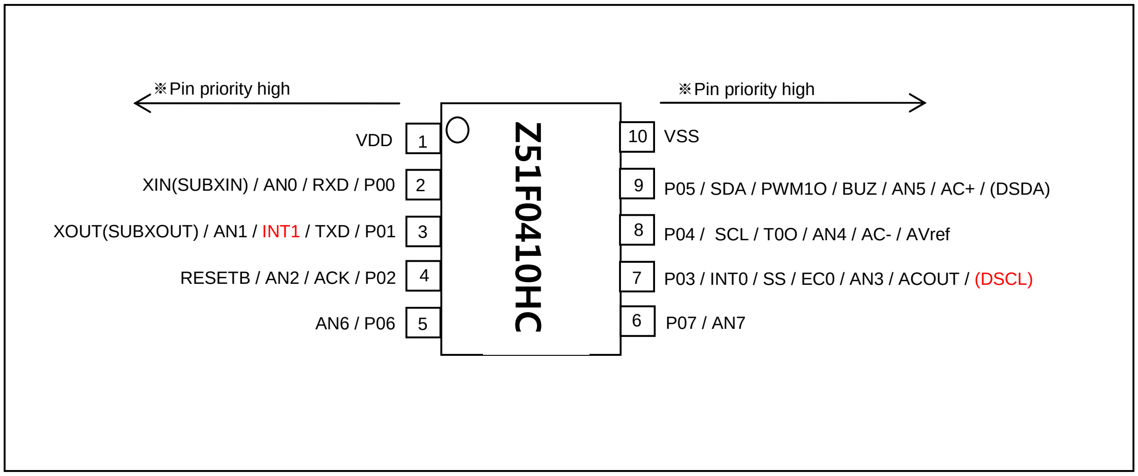

※Pin priority high
※Pin priority high
Z51F0410HC
VDD
1
XIN(SUBXIN) / AN0 / RXD / P00
2
XOUT(SUBXOUT) / AN1 / INT1 / TXD / P01
3
RESETB / AN2 / ACK / P02
4
AN6 / P06
5
10
VSS
9
P05 / SDA / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8
P04 / SCL / T0O / AN4 / AC- / AVref
7
P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6
P07 / AN7

Figure 1.3 10-SSOP PIN Assignment Diagram of Z51F0410HCX

NOTE)

- If 8 PIN PKG, Pin 5 and 6 are removed in 10 PIN PKG.
- On On-Chip Debugging, ISP uses P0[3], P0[5] pin as DSCL, DSDA.
- P00, P01,P02,P06 pin priority is high from left to right. P05, P04,P03,P07 pin priority is high from right to left.

1.7 Package Diagram

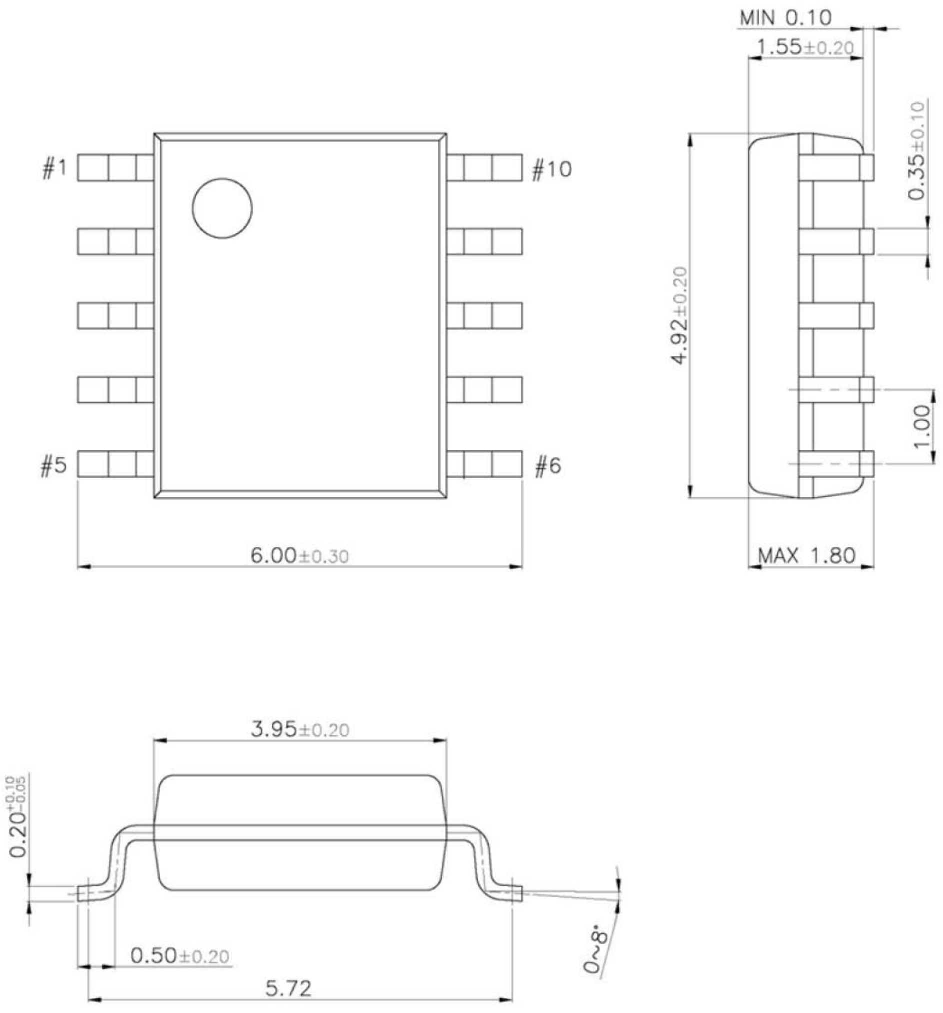


Figure 1.4 10-SSOP Package Diagram

1.8 Reconfigurable Pin Description

1.8.1 USART Pin Location Switch Mode

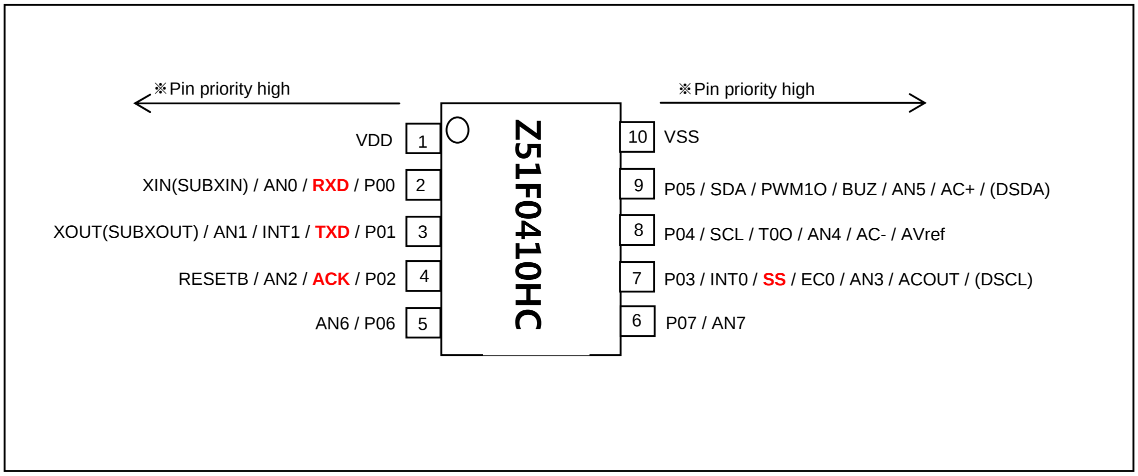

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SUBXIN) / AN0 / RXD / P00 2
XOUT(SUBXOUT) / AN1 / INT1 / TXD / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / P06 5
10 VSS
9 P05 / SDA / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8 P04 / SCL / T0O / AN4 / AC- / AVref
7 P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / AN7

Figure 1.5 USART Pin Location : USART_PINMODE=0

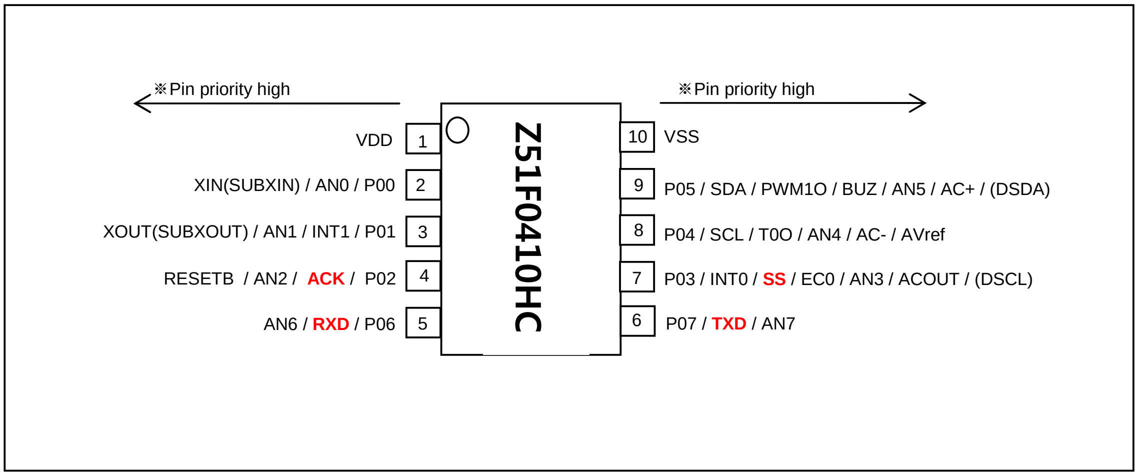

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SUBXIN) / AN0 / P00 2
XOUT(SUBXOUT) / AN1 / INT1 / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / RXD / P06 5
10 VSS
9 P05 / SDA / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8 P04 / SCL / T0O / AN4 / AC- / AVref
7 P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / TXD / AN7

Figure 1.6 USART Pin Location : USART_PINMODE=1

1.8.2 I2C Pin Location Switch Mode

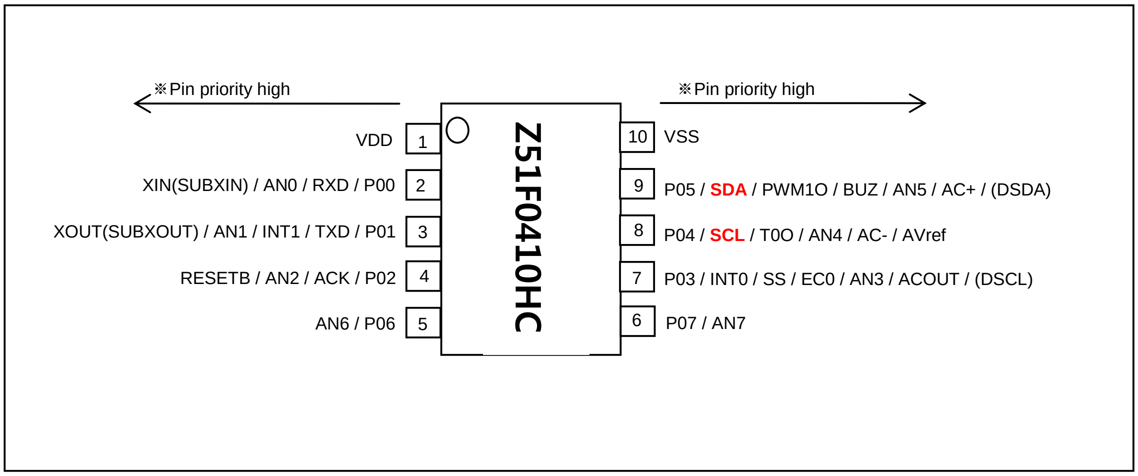

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SUBXIN) / AN0 / RXD / P00 2
XOUT(SUBXOUT) / AN1 / INT1 / TXD / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / P06 5
10 VSS
9 P05 / SDA / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8 P04 / SCL / T0O / AN4 / AC- / AVref
7 P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / AN7

Figure 1.7 I2C Pin Location : I2C_PINMODE=0

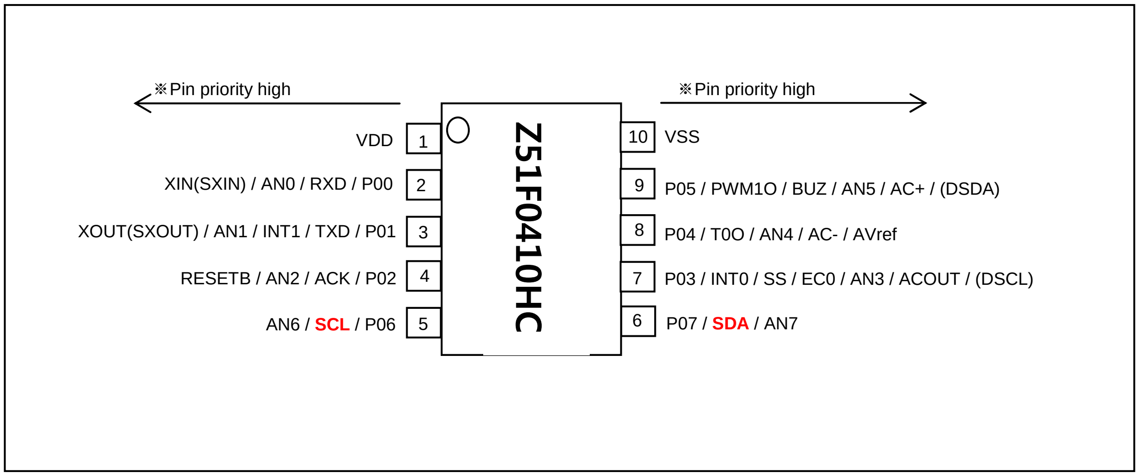

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SXIN) / AN0 / RXD / P00 2
XOUT(SXOUT) / AN1 / INT1 / TXD / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / SCL / P06 5
10 VSS
9 P05 / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8 P04 / T0O / AN4 / AC- / AVref
7 P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / SDA / AN7

Figure 1.8 I2C Pin Location : I2C_PINMODE=1

1.8.3 External Interrupt Pin Location Switch Mode

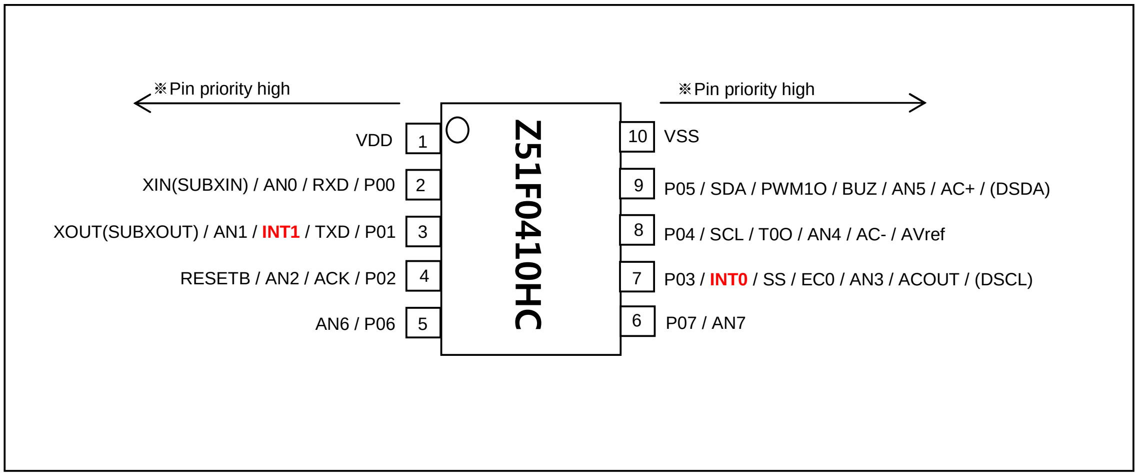

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SUBXIN) / AN0 / RXD / P00 2
XOUT(SUBXOUT) / AN1 / INT1 / TXD / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / P06 5
10 VSS
9 P05 / SDA / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8 P04 / SCL / T0O / AN4 / AC- / AVref
7 P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / AN7

Figure 1.9 External INT Pin Location : EINT_PINMODE=0

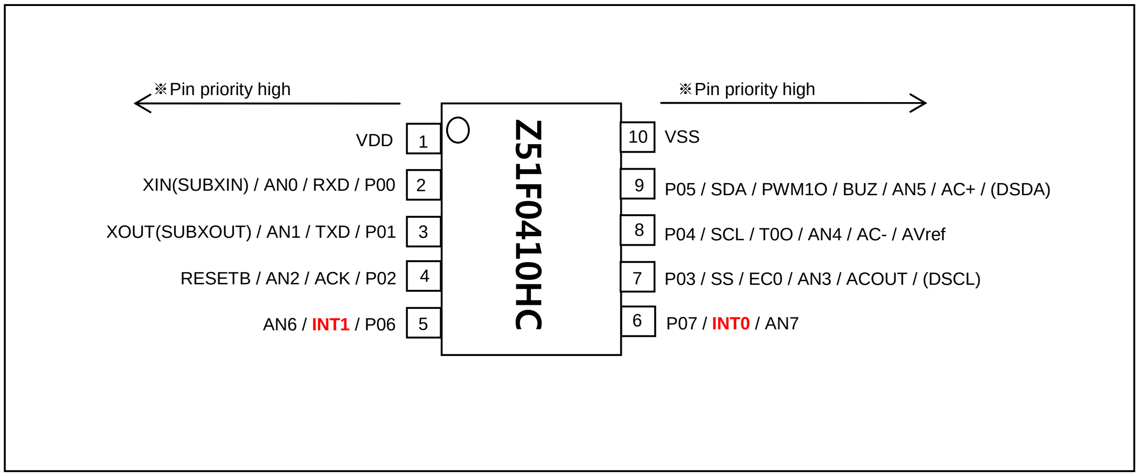

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SUBXIN) / AN0 / RXD / P00 2
XOUT(SUBXOUT) / AN1 / TXD / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / INT1 / P06 5
10 VSS
9 P05 / SDA / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8 P04 / SCL / T0O / AN4 / AC- / AVref
7 P03 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / INT0 / AN7

Figure 1.10 External INT Pin Location : EINT_PINMODE=1

1.8.4 Buzzer Out Pin Location Switch Mode

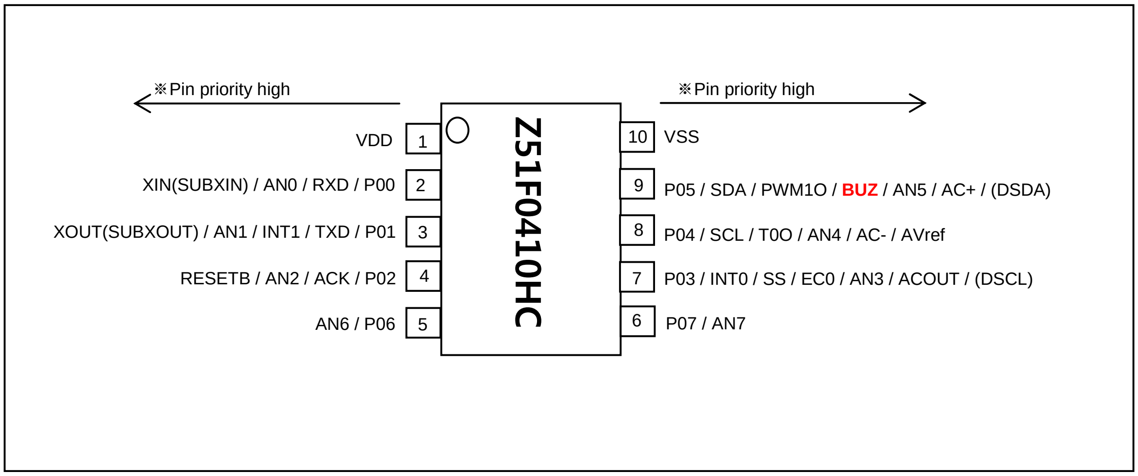

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SUBXIN) / AN0 / RXD / P00 2
XOUT(SUBXOUT) / AN1 / INT1 / TXD / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / P06 5
10 VSS
9 P05 / SDA / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8 P04 / SCL / T0O / AN4 / AC- / AVref
7 P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / AN7

Figure 1.11 Buzzer Out Pin Location : BUZO_PINMODE=0

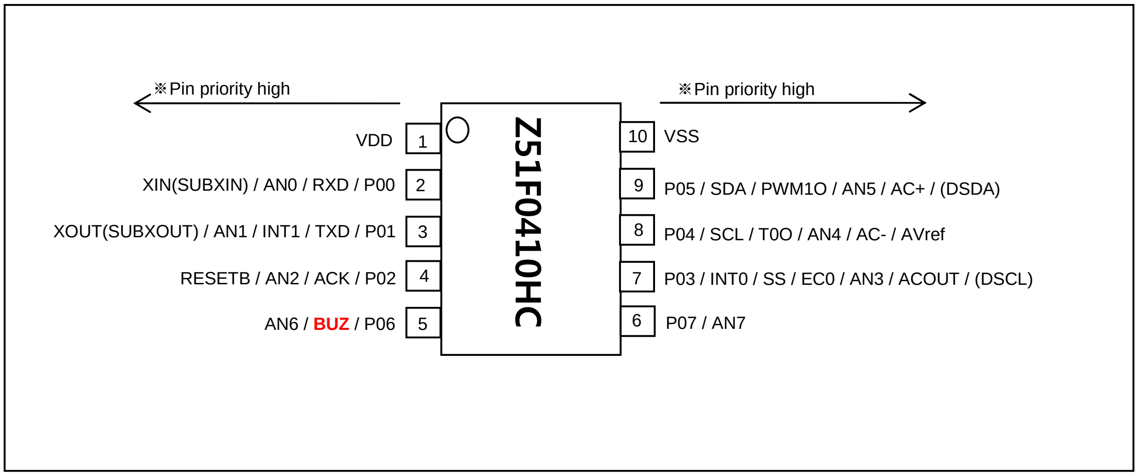

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SUBXIN) / AN0 / RXD / P00 2
XOUT(SUBXOUT) / AN1 / INT1 / TXD / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / BUZ / P06 5
10 VSS
9 P05 / SDA / PWM1O / AN5 / AC+ / (DSDA)
8 P04 / SCL / T0O / AN4 / AC- / AVref
7 P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / AN7

Figure 1.12 Buzzer Out Pin Location : BUZO_PINMODE=1

1.8.5 TIMER Pin Location Switch Mode

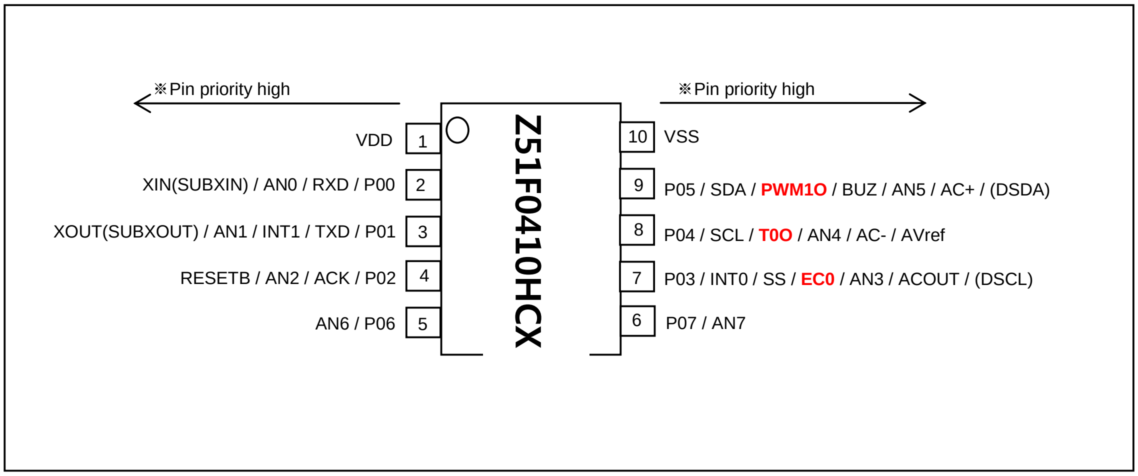

※Pin priority high
※Pin priority high
Z51F0410HCX
VDD 1
XIN(SUBXIN) / AN0 / RXD / P00 2
XOUT(SUBXOUT) / AN1 / INT1 / TXD / P01 3
RESETB / AN2 / ACK / P02 4
AN6 / P06 5
10 VSS
9 P05 / SDA / PWM1O / BUZ / AN5 / AC+ / (DSDA)
8 P04 / SCL / T0O / AN4 / AC- / AVref
7 P03 / INT0 / SS / EC0 / AN3 / ACOUT / (DSCL)
6 P07 / AN7

Figure 1.13 Timer Out Pin Location : TMR_PINMODE=0

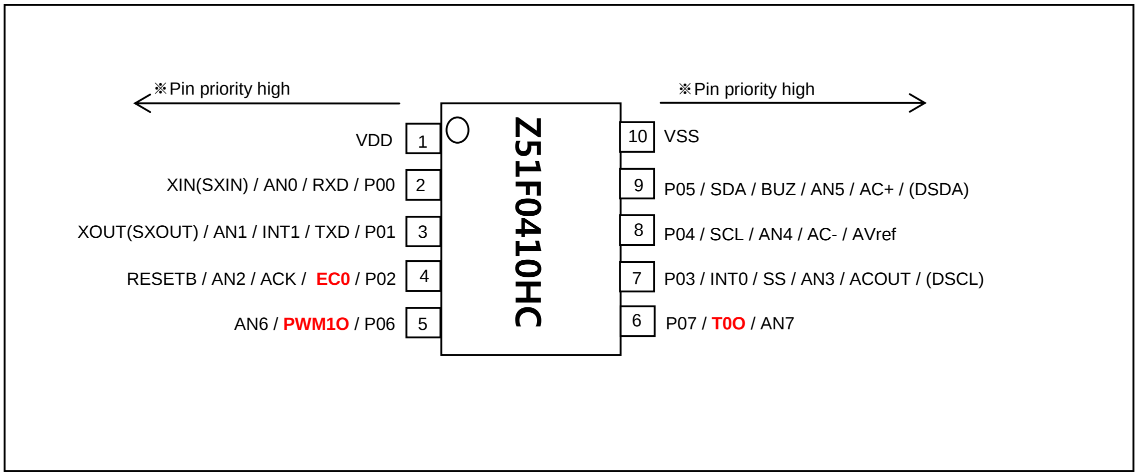

※Pin priority high
※Pin priority high
Z51F0410HC
VDD 1
XIN(SXIN) / AN0 / RXD / P00 2
XOUT(SXOUT) / AN1 / INT1 / TXD / P01 3
RESETB / AN2 / ACK / EC0 / P02 4
AN6 / PWM1O / P06 5
10 VSS
9 P05 / SDA / BUZ / AN5 / AC+ / (DSDA)
8 P04 / SCL / AN4 / AC- / AVref
7 P03 / INT0 / SS / AN3 / ACOUT / (DSCL)
6 P07 / T0O / AN7

Figure 1.14 Timer Out Pin Location : TMR_PINMODE=1

1.9 Code Encryption (Super Lock)

Basically user code data will be programmed with raw data in Flash area. Although code data read mode is protected with lock mode, its contents are vulnerable for several codes hooking method. We provide the code encryption method to secure the user code data. The original user code will be scrambled with user seed key value (private key). The scrambled data is programmed in flash area. The fetched code during CPU operation will be decoded with the user key in configuration area.

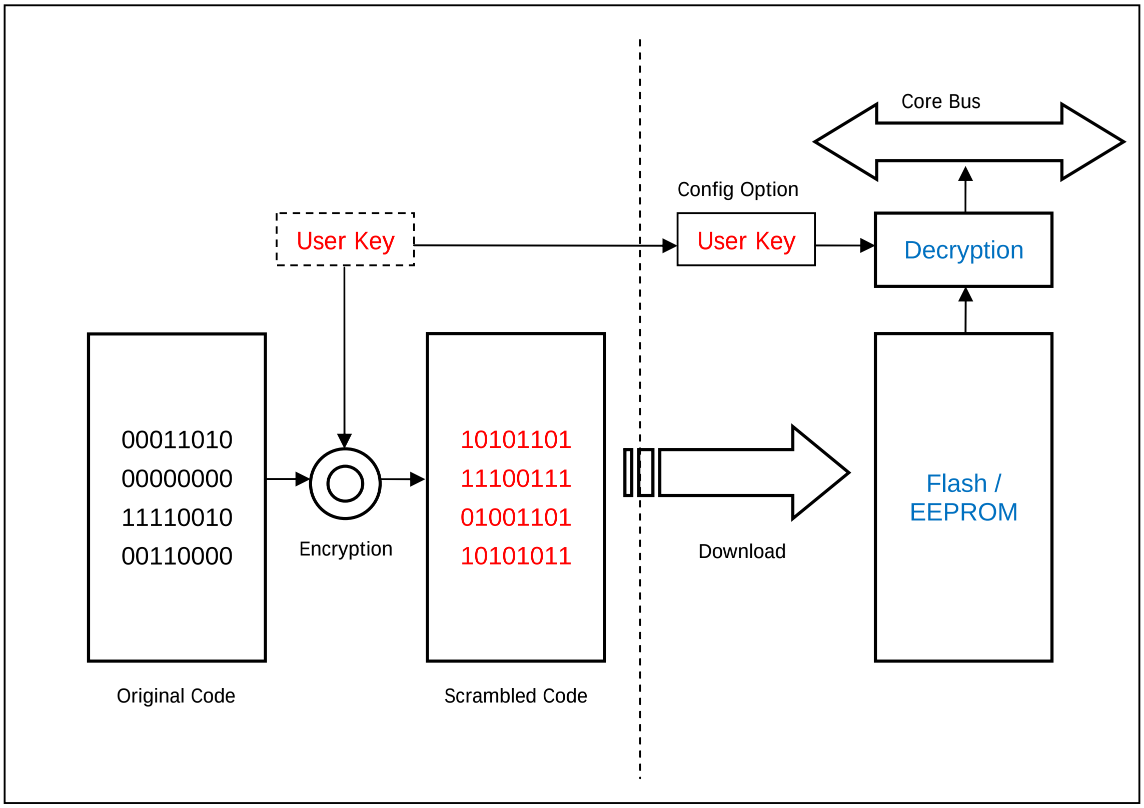


Figure 1.15 Super Lock Enabled Encryption/Decryption Diagram

1.10 Port Structure

1.10.1 General Purpose I/O Port

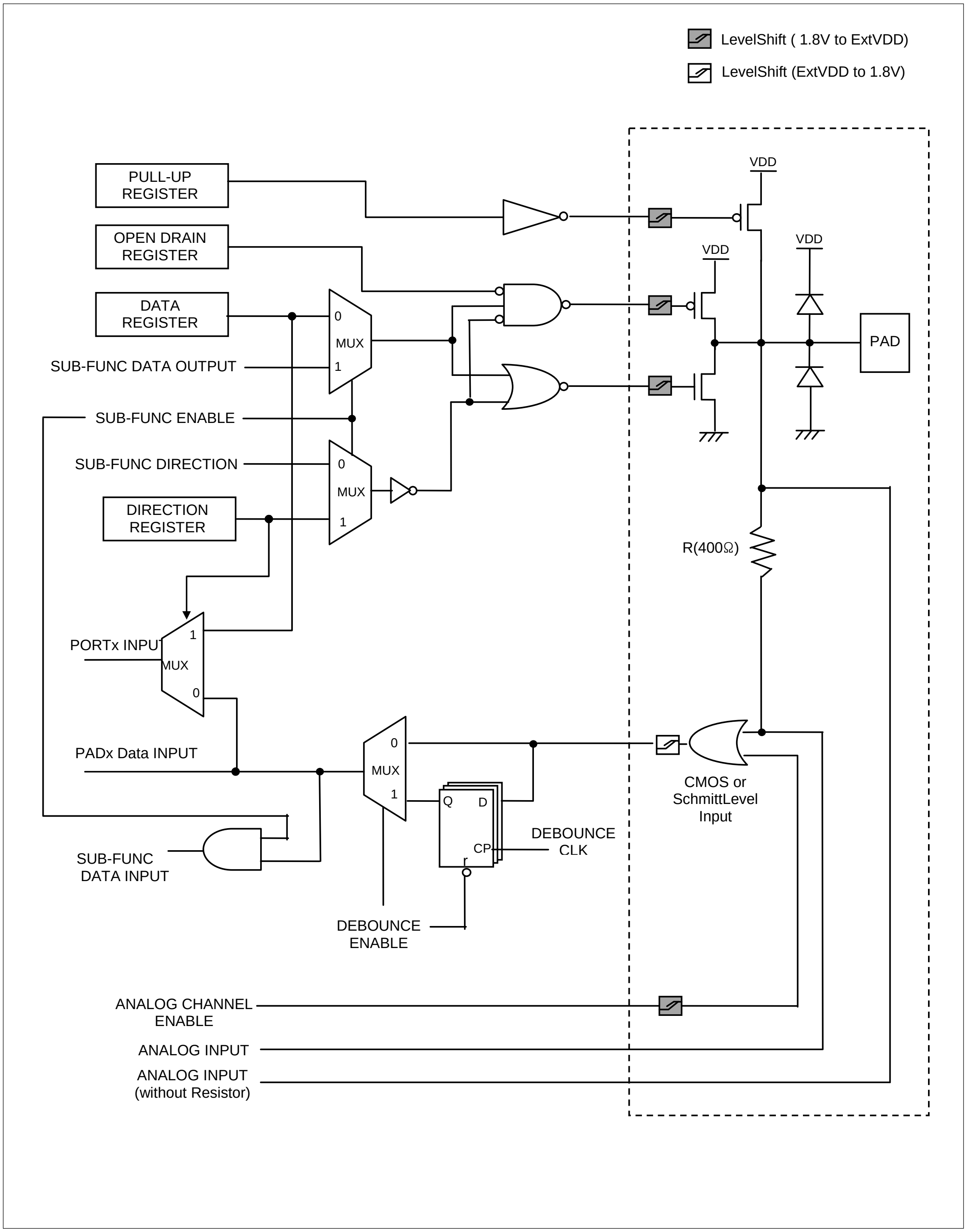


Figure 1.16 General Purpose I/O Port

1.10.2 External Interrupt I/O Port

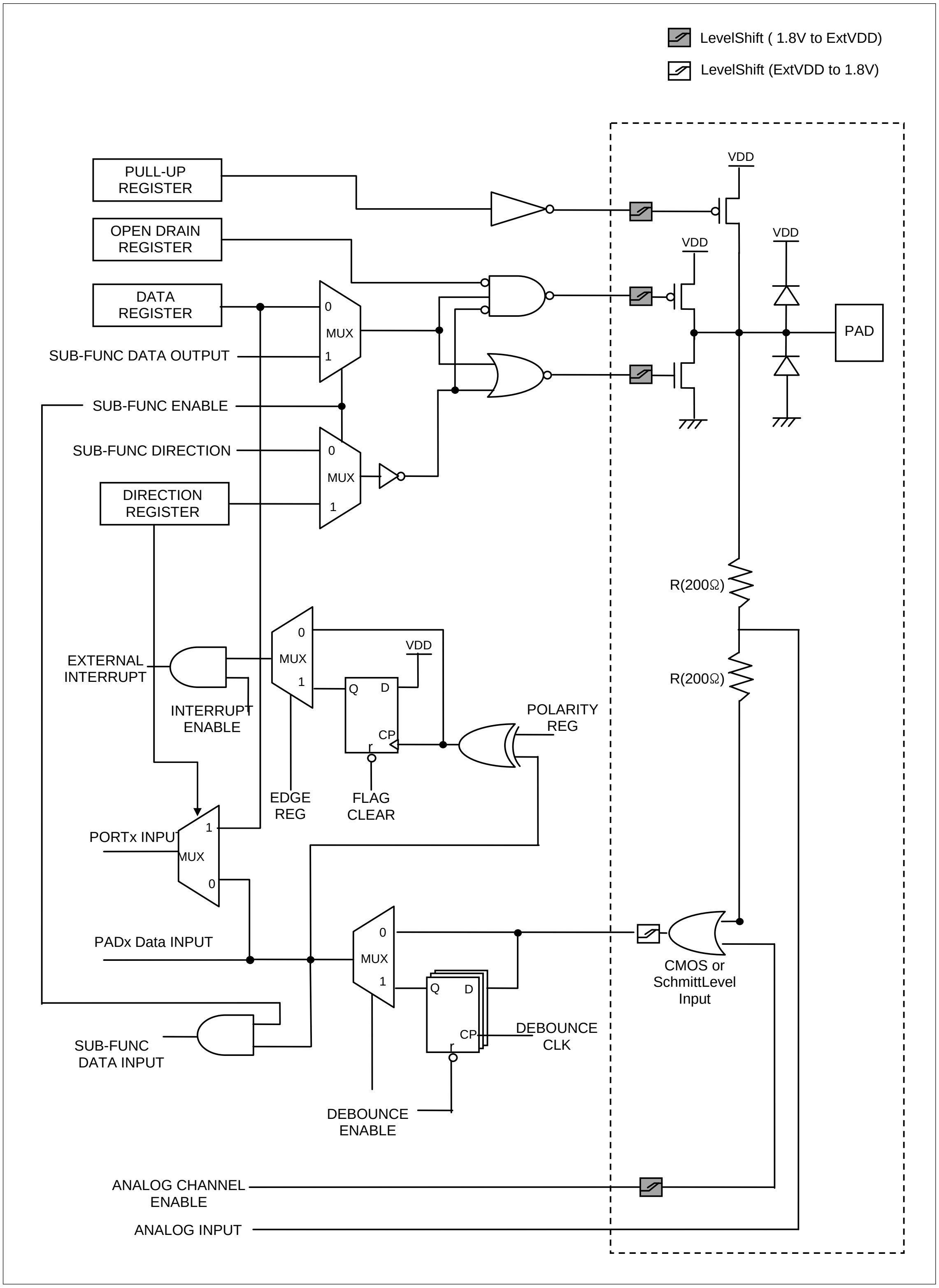


Figure 1.17 External Interrupt I/O Port

1.11 Port Structure Diagram (detail view)

1.11.1 P0[0] Port Structure

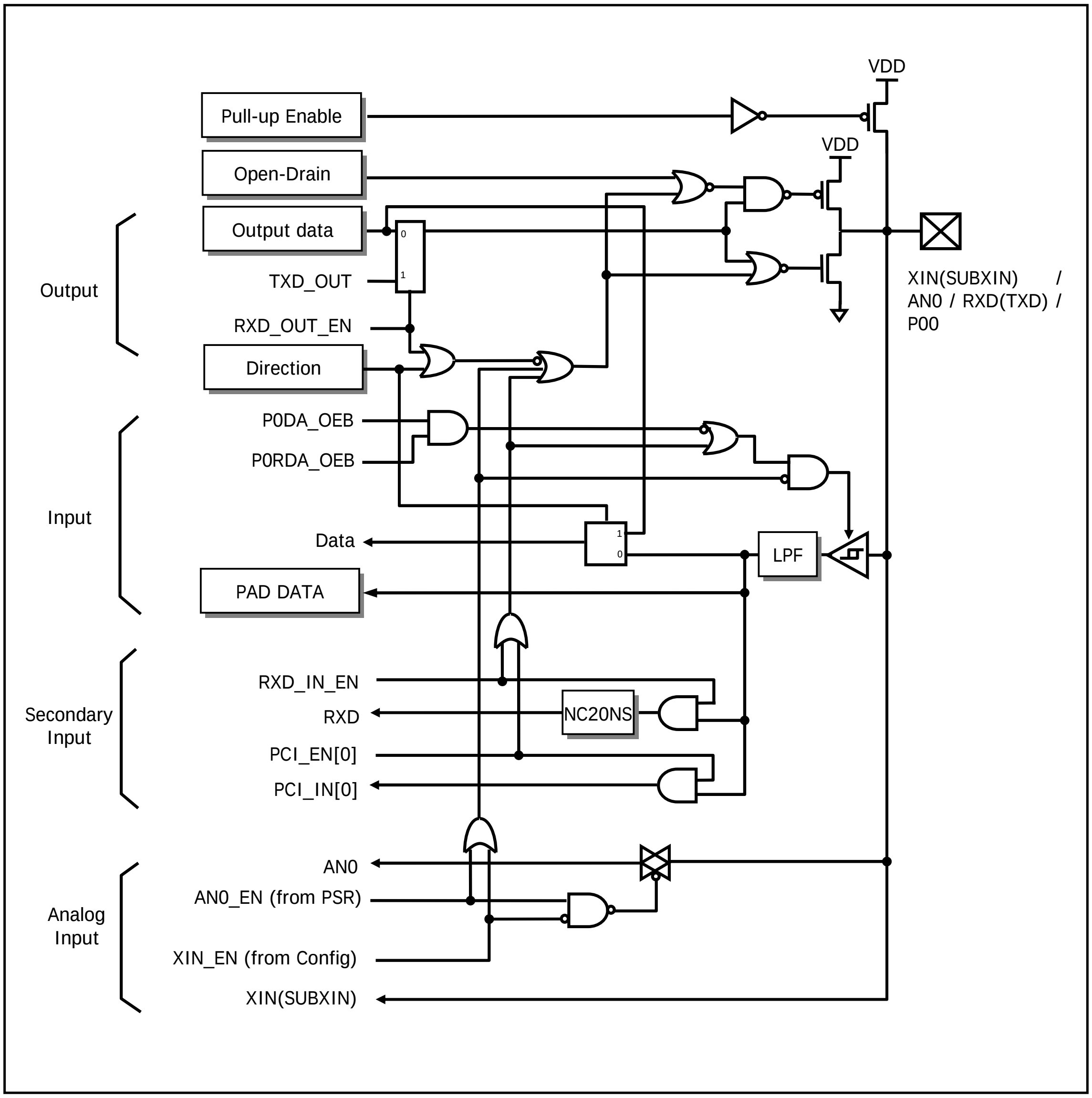


Figure 1.18 XIN(SUBXIN) / AN0 /RXD /P0[0] Port Structure

The pull-up resister is directly controlled by the pull-up register bit regardless of current port direction. The open-drain control is also by open-drain register. On open-drain mode, the push-pull drives just N-MOS. When the direction is output (value 1), the output PAD voltage is controlled by push-pull driver for the current output data. The secondary input or analog channel selection bit disable the output direction regardless of the current direction register. The secondary input RXD_EN, PCI_EN[0] enables the input data path continuously. On normal read mode (non secondary mode), the input data path is only enabled during the CPU OEB (active low). When the analog channel (AN0) is enabled, the first input gate from the PAD is disabled (highest priority) to prevent the input leakage current for the floating voltage status. The XIN function disables all analog channels and secondary input/output. At read operation, the input data is selected by PAD direction register. If its value is ‘1’, it reads the current output register value. Otherwise, it reads the current PAD voltage directly (just during OEB active). In addition, always the current PAD voltage is read by PAD DATA register.

1.11.2 P0[1] Port Structure

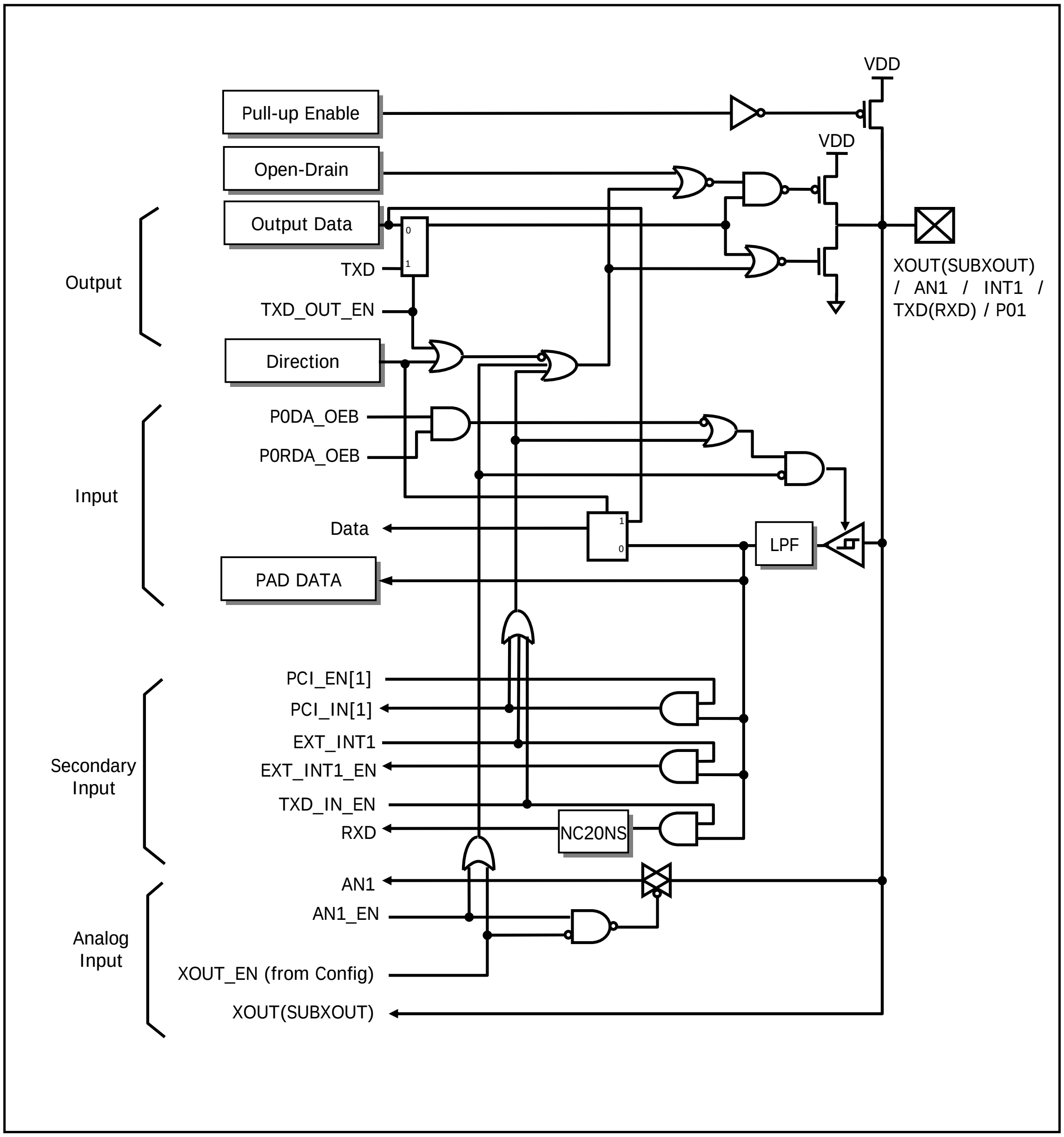


Figure 1.19 XOUT(SUBXOUT) / AN1 / INT1 / TXD / P0[1] Port Structure

The Figure 1.19 shows a brief diagram of P0[1] port structure. The pull-up resister is directly controlled by the pull-up register bit regardless of current port direction. The open-drain control is also by open-drain register. On open-drain mode, the push-pull drives just N-MOS. When the direction is output (value 1), the output PAD voltage is controlled by push-pull driver for the current output data. The secondary input or analog channel selection bit disable the output direction regardless of the current direction register. The secondary input INT1_EN, PCI_EN[1] enables the input data path continuously. On normal read mode (non secondary mode), the input data path is only enabled during the CPU OEB (active low). When the analog channel (AN1) is enabled, the first input gate from the PAD is disabled (highest priority) to prevent the input leakage current for the floating voltage status. The XOUT function disables all analog channels and secondary input/output. At read operation, the input data is selected by PAD direction register. If its value is ‘1’, it reads the current output register value. Otherwise, it reads the current PAD voltage directly (just during OEB active). In addition, always the current PAD voltage is read by PAD DATA register.

1.11.3 P0[2] Port Structure

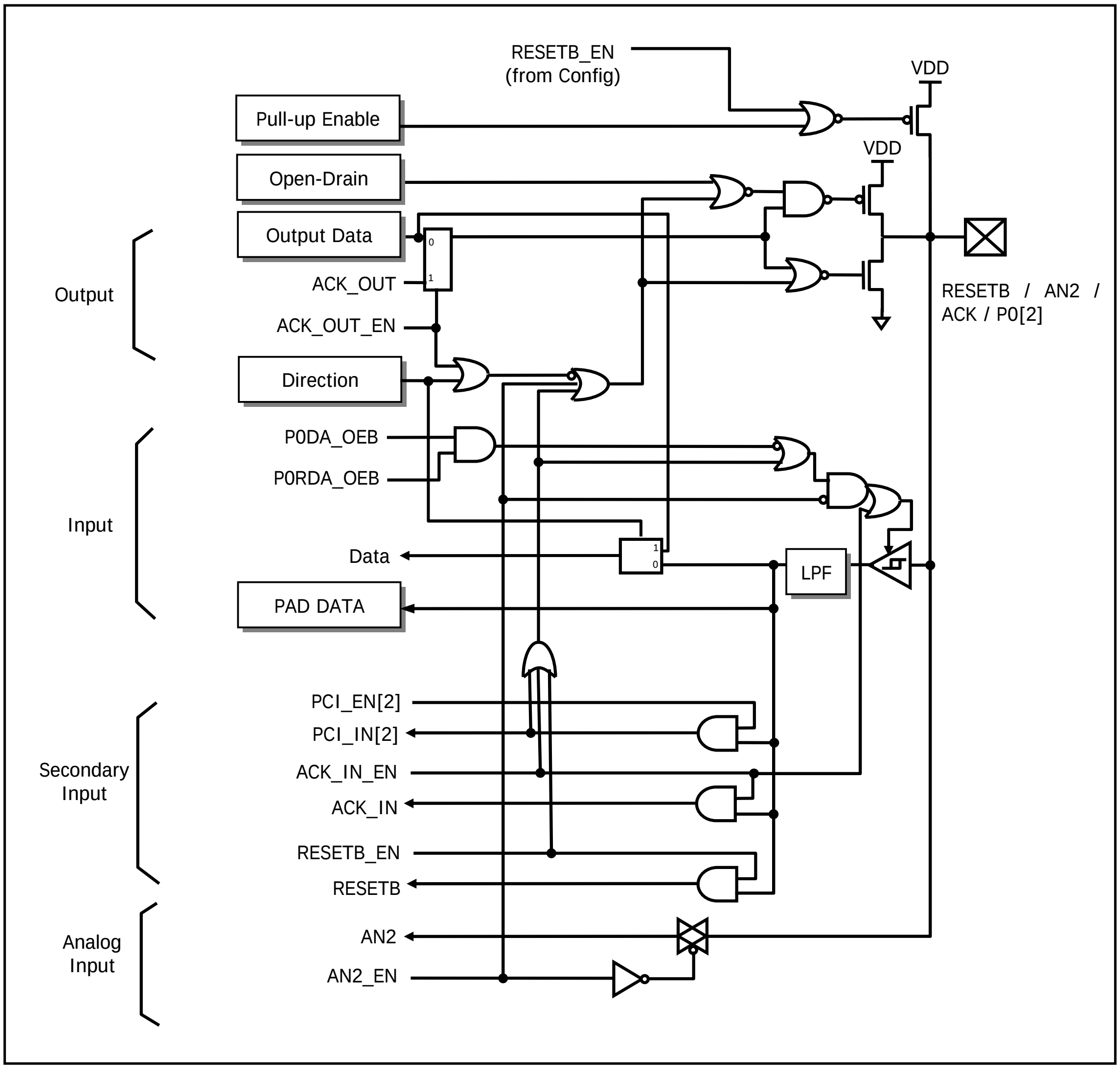


Figure 1.20 RESETB / AN2 / ACK / P0[2] Port Structure

If the RESETB_EN (from config data) is 1, the input secondary data path is enabled with the highest priority level and it automatically enables the pull-up function regardless of pull-up register value.

The analog channel selection bit enables the path of the AN2 and disable normal logic data path to prevent the input gate leakage current. When the direction register value is 0, the input data is always external PAD voltage.

The pull-up resister is directly controlled by the pull-up register bit regardless of current port direction. The open-drain control is also by open-drain register. On open-drain mode, the push-pull drives just N-MOS. When the direction is output (value 1), the output PAD voltage is controlled by push-pull driver for the current output data. The secondary input or analog channel selection bit disable the output direction regardless of the current direction register. The secondary input ACK_IN_EN, PCI_EN[2] enable the input data path continuously. On normal read mode (non secondary mode), the input data path is only enabled during the CPU OEB (active low). When the analog channel (AN2) is enabled, the first input gate from the PAD except of RESET enabled is disabled (highest priority) to prevent the input leakage current for the floating voltage status. At read operation, the input data is selected by PAD direction register. If its value is '1', it reads the current output register value.

Otherwise, it reads the current PAD voltage directly (just during OEB active). In addition, always the current PAD voltage is read by PAD DATA register.

1.11.4 P0[3] Port Structure

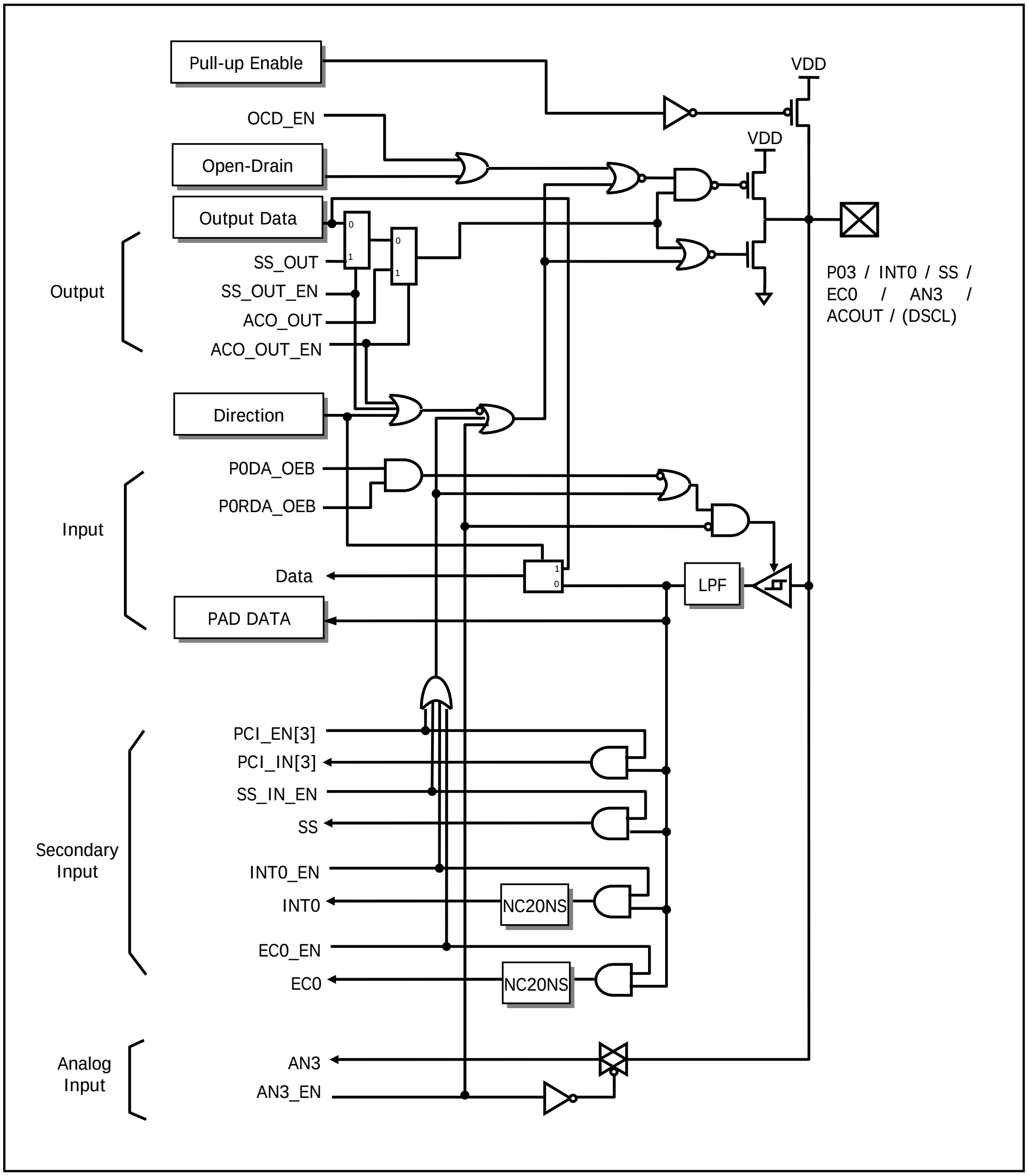


Figure 1.21 P0[3] / INT0 / SS / EC0 / AN3 / ACOUT Port Structure

The pull-up resister is directly controlled by the pull-up register bit regardless of current port direction. The open-drain control is also by open-drain register. On open-drain mode, the push-pull drives just N-MOS. The OCD mode enable the Open-drain Output regardless of the Open-Drain Register value. When the direction is output (value 1), the output PAD voltage is controlled by push-pull driver for the current output data. The secondary input or analog channel selection bit disable the output direction regardless of the current direction register. The secondary input SS_EN, INT0_EN, EC0_EN, PCI_EN[3] enable the input data path continuously. On normal read mode (non secondary mode), the input data path is only enabled during the CPU OEB (active low). When the analog channel (AN3) is enabled, the first input gate from the PAD is disabled (highest priority) to prevent the input leakage current for the floating voltage status. At read operation, the input data is selected by PAD direction register. If its value is ‘1’, it reads the current output register value. Otherwise, it reads the current PAD voltage directly (just during OEB active). In addition, always the current PAD voltage is read by PAD DATA register.

1.11.5 P0[4] Port Structure

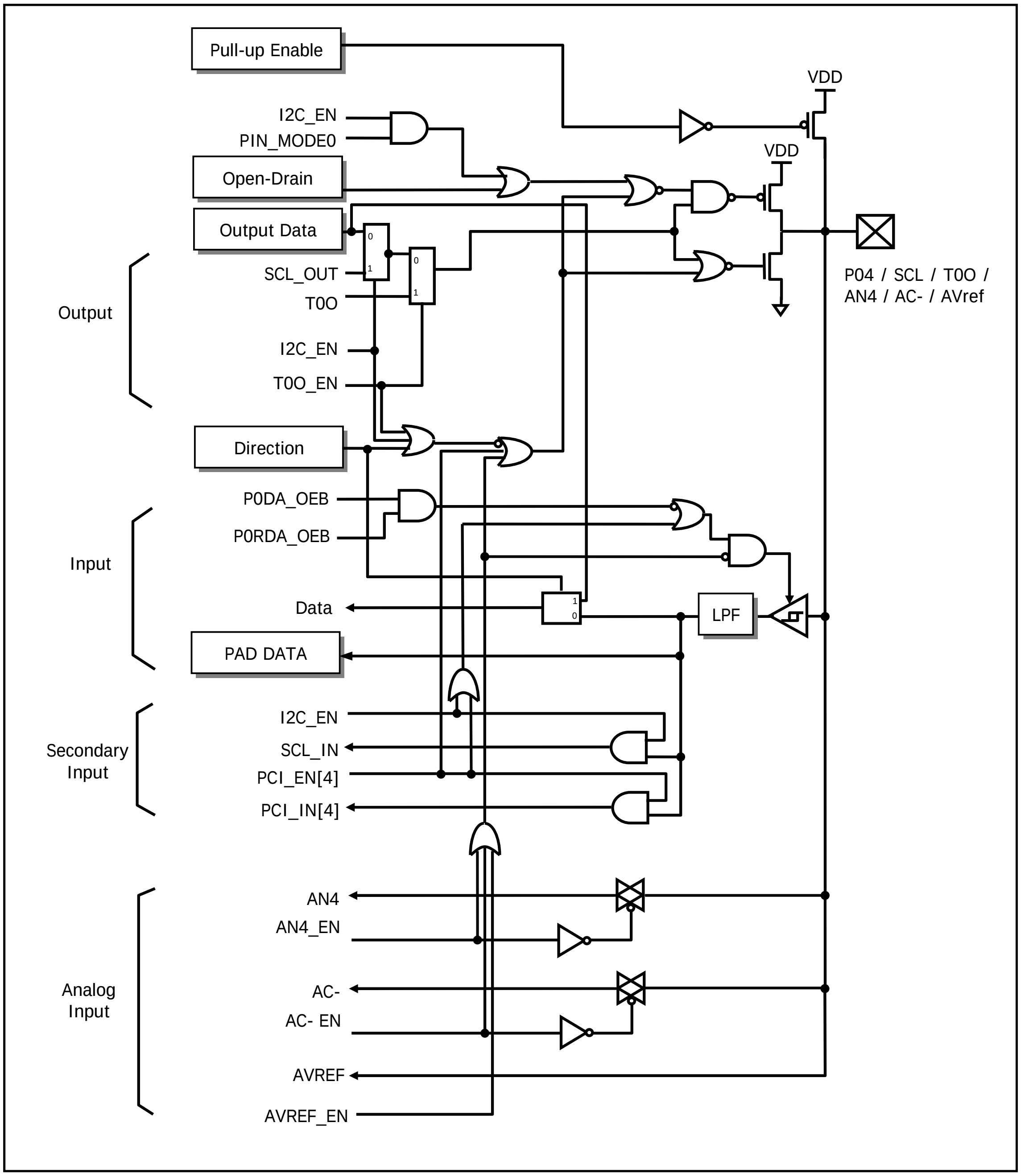


Figure 1.22 P0[4]/SCL/T0O/AN4/AC-/Avref Port Diagram

The pull-up resister is directly controlled by the pull-up register bit regardless of current port direction. The open-drain control is also by open-drain register. On open-drain mode, the push-pull drives just N-MOS. The I2C Mode enable the Open-drain Output regardless of the Open-Drain Register value. When the direction is output (value 1), the output PAD voltage is controlled by push-pull driver for the current output data. The secondary input or analog channel selection bit disable the output direction regardless of the current direction register. The secondary input SCL_IN_EN, PCI_EN[4] enables the input data path continuously. On normal read mode (non secondary mode), the input data path is only enabled during the CPU OEB (active low). When the analog channel (AN4,AC-) is enabled, the first input gate from the PAD is disabled (highest priority) to prevent the input leakage current for the floating voltage status. The AVREF function disables all analog channels and secondary input/output.At read operation, the input data is selected by PAD direction register. If its value is ‘1’, it reads the current output register value. Otherwise, it reads the current PAD voltage directly (just during OEB active). In addition, always the current PAD voltage is read by PAD DATA register.

1.11.6 P0[5] Port Structure

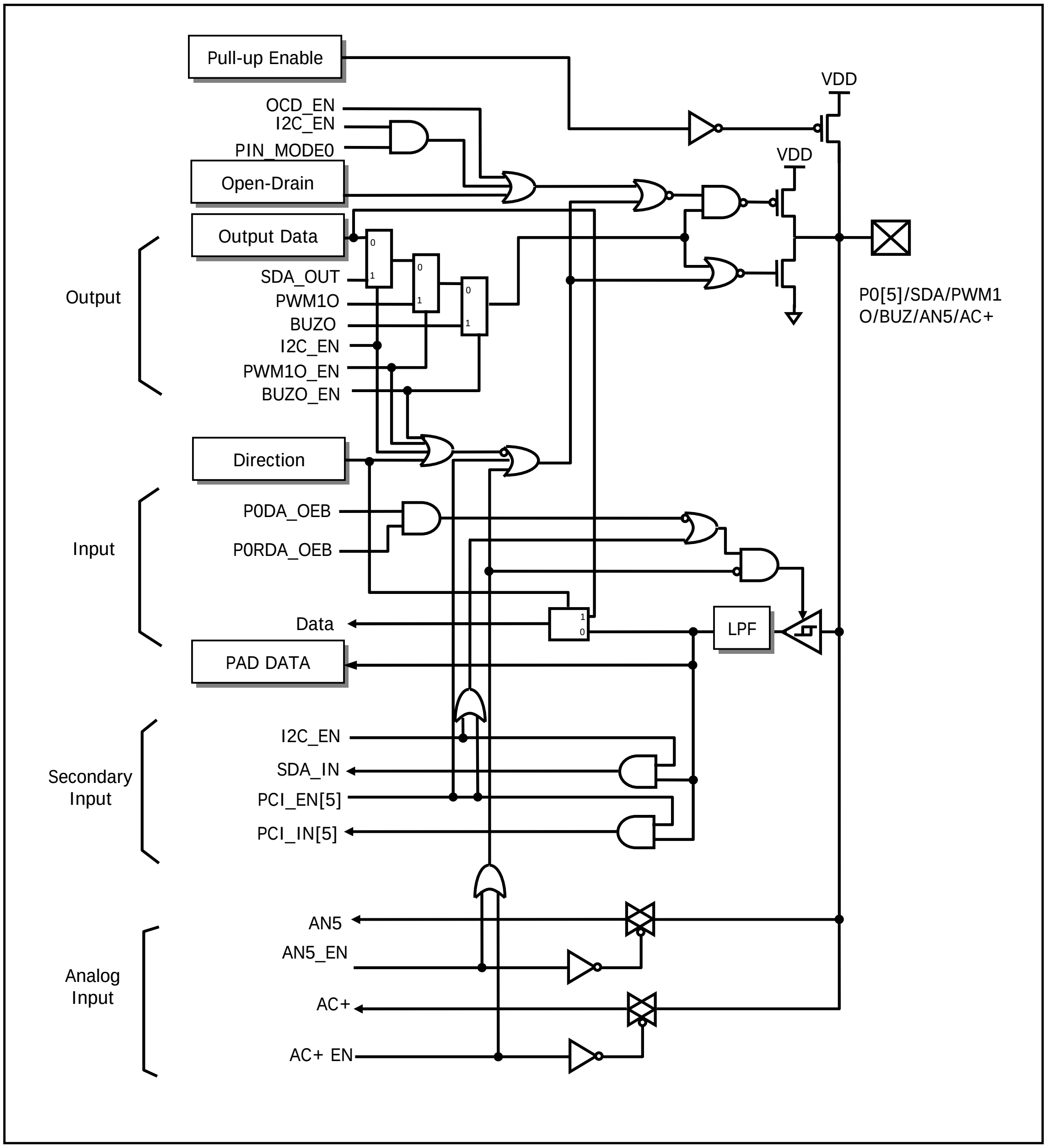


Figure 1.23 P0[5]/SDA/PWM1O/BUZ/AN5/AC+ Port Structure

The pull-up resister is directly controlled by the pull-up register bit regardless of current port direction. The open-drain control is also by open-drain register. On open-drain mode, the push-pull drive just N-MOS. The I2C Mode and OCD Mode enable the Open-drain Output regardless of the Open-Drain Register value. When the direction is output (value 1), the output PAD voltage is controlled by push-pull driver for the current output data. The secondary input or analog channel selection bit disable the output direction regardless of the current direction register. The secondary input SDA_IN_EN, PCI_EN[5] enables the input data path continuously. On normal read mode (non secondary mode), the input data path is only enabled during the CPU OEB (active low). When the analog channel (AN5) is enabled, the first input gate from the PAD is disabled (highest priority) to prevent the input leakage current for the floating voltage status. The AC+ function disables all analog channels and secondary input/output. At read operation, the input data is selected by PAD direction register. If its value is ‘1’, it reads the current output register value. Otherwise, it reads the current PAD voltage directly (just during OEB active). In addition, always the current PAD voltage is read by PAD DATA register.

1.11.7 P0[6]/P0[7] Port Structure

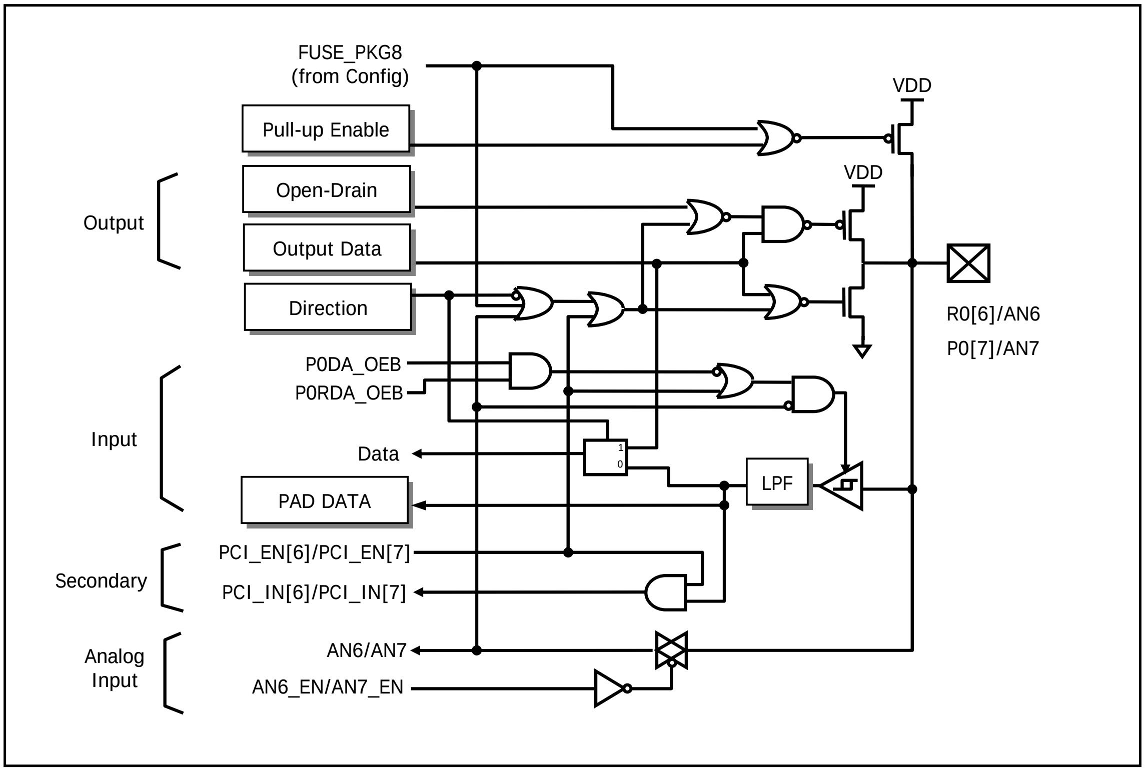


Figure 1.24 P0[6] / AN6, P0[7] / AN7 Port Structure

The analog channel selection bit enables the path of the AN6/AN7 and disable normal logic data path to prevent the input gate leakage current. When the direction register value is 0, the input data is always external PAD voltage.

The pull-up resister is directly controlled by the pull-up register bit regardless of current port direction. The open-drain control is also by open-drain register. On open-drain mode, the push-pull drives just N-MOS. When the direction is output (value 1), the output PAD voltage is controlled by push-pull driver for the current output data. The secondary input PCI_EN[6]/PCI_EN[7] enable the input data path continuously. On normal read mode (non secondary mode), the input data path is only enabled during the CPU OEB (active low). When the analog channel (AN6/AN7) is enabled, the first input gate from the PAD is disabled (highest priority) to prevent the input leakage current for the floating voltage status. At read operation, the input data is selected by PAD direction register. If its value is ‘1’, it reads the current output register value. Otherwise, it reads the current PAD voltage directly (just during OEB active). In addition, always the current PAD voltage is read by PAD DATA register.

1.12 Electrical Characteristics

1.12.1 Absolute Maximum Ratings

Table 1.2 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply Voltage	VDD	-0.3~+6.5	V
	VSS	-0.3~+0.3	V
Normal Voltage Pin	VI	-0.3~VDD+0.3	V
	VO	-0.3~VDD+0.3	V
	IOH	10	mA
	∑IOH	80	mA
	IOL	20	mA
	∑IOL	160	mA
Total Power Dissipation	PT	600	mW
Storage Temperature	TSTG	-45~+125	℃

Note) Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

1.12.2 Recommended Operating Conditions

Table 1.3 Recommended Operation Conditions

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Supply Voltage	VDD	fXIN=1.0~8MHz	1.8	-	5.5	V
		fSUB=32.768KHz				
Operating Temperature	TOPR	VDD=1.8~5.5V	-40	-	85	℃
Operating Frequency	FOPR	fXIN	1	-	8	MHz
		fSUB	-	32.768	-	KHz
		Internal RC-OSC	-	8	-	MHz

1.12.3 A/D Converter Characteristics

(TA=-40℃ ~ +85℃, VDD=AVDD=1.8V ~ 5.5V, VSS=0V)

Table 1.4 A/D Converter Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Resolution		-	-	12	-	bits
Total Accuracy		AVDD=VDD=5.12 V fXIN=4MHz		-	±3(Avref base)	lsb
Integral Linear Error	INL		-	-	±3	lsb
Differential Linearity Error	DLE		-	-	±2	lsb
Zero Offset Error	ZOE		-		±3	lsb
Full Scale Error	FSE		-		±3	lsb

Conversion Time	tCON	12bit conversion Max 3MHz	-	60	-	cycle
Analog Input Voltage	VAN	-	VSS	-	AVDD=VDD	V
Analog Power Voltage	AVDD	-	-	*AVDD=VDD	-	V
Analog Reference Voltage	AVREF	-	2.0	-	AVREF=AVDD	V
Analog Ground Voltage	AVSS	-	-	VSS	-	V
Analog Input Leakage Current		AVDD=VDD=5.12 V	-	-	10	uA
ADC Operating Current	IDD	AVDD=VDD=5.12 V	-	1	3	mA
	SIDD		-	-	1	uA

1.12.4 Voltage Dropout Converter Characteristics

Table 1.5 Voltage Dropout Converter Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	1.8	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
Regulation Voltage		-	1.62	1.8	1.98	V
Drop-out Voltage		-	-	-	0.02	V
Current Drivability		RUN/IDLE	-	10	-	mA
		SUB-ACTIVE	-	1	-	mA
		STOP1	-	50	-	uA
		STOP2	-	10	-	uA
Operating Current	IDD1	RUN/IDLE	-	-	1	mA
	IDD2	SUB-ACTIVE	-	-	0.1	mA
	SIDD1	STOP1	-	-	5	uA
	SIDD2	STOP2	-	-	0.1	uA
Drivability Transition Time	TRAN1	SUB to RUN	-	-	1	uS
	TRAN2	STOP to RUN	-	-		

Note) -STOP1: WDT running - STOP2: WDT disable

1.12.5 Power-On Reset Characteristics

Table 1.6 Power-On Reset Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	VSS	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
RESET Release Level		-	1.3	1.4	1.5	V
Operating Current	IDD	-	-	-	10	uA
	SIDD	-	-	-	1	uA

1.12.6 Brown Out Detector Characteristics

Table 1.7 Brown Out Detector Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Operating Voltage		-	VSS	-	5.5	V
Operating Temperature		-	-40	-	+85	℃
Detection Level		-	4.1	4.3	4.4	V
		-	3.4	3.6	3.7	V
		-	2.4	2.5	2.6	V
		-	1.5	1.6	1.7	V
Hysteresis		-	-	50	-	mV
Operating Current	IDD	-	-	-	50	uA
	SIDD	-	-	-	1	uA

1.12.7 Internal 8Mhz, 128Khz RC Oscillator Characteristics

Table 1.8 Internal 8Mhz RC Oscillator Characteristics

Parameter	Condition	MIN	TYP	MAX	Unit
Operating Voltage		1.62	1.8	3.3	V
Operating Temp.	TBD	-40		85	℃
Clock Freq.	25℃	7.92	8	8.08	MHz
Operating Current	Average Current	140	170	200	uA

Table 1.9 Internal 128Khz RC Oscillator Characteristics

	Condition	MIN	TYP	MAX	Unit
Operating Voltage		1.62	1.8	5.5	V
Operating Temp.	TBD	-40		85	℃
Clock Freq.	TBD	-	128	-	KHz
Operating Current	Average Current	15	20	40	uA

1.12.8 Analog Comparator Characteristics

Table 1.10 DC Electrical Characteristics

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			Temp= -40 °C to 85 °C			
			MIN	TYP	MAX	
IL	Input leakage current	VDDEXT=5V, Vin=1/2VDDEXT	-50	-	50	nA

Voffset	Input offset voltage	VDDEXT=5V, Vin=1/2VDD	10	-	40	±mV
IOP	Operating current	COMP_EN=H		1		mA
IPD	Power down current	COMP_EN=L		1		uA

Table 1.11 AC Characteristics

SYMBOL	PARAMETER	TEST CONDITION	LIMITS			UNIT
			Temp=-40 °C to 85 °C			
			MIN	TYP	MAX	
VRT	Comparator Response time	CL= 50pF, VDDEXT=5V	-	-	500	ns

1.12.9 DC Characteristics

(VDD =1.8~5.5V, VSS =0V, fXIN=10.0MHz, TA=-40~+85℃)

Table 1.12 DC Characteristics

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Input Low Voltage	VIL1	RESETB (External Reset Active) P0	-0.5	-	0.2VDD	V
	VIL2	-	-0.5	-	0.2VDD	V
Input High Voltage	VIH1	RESETB (External Reset Active) P0	0.8VDD	-	VDD+0.5	V
	VIH2	-	0.7VDD	-	VDD+0.5	V
Output Low Voltage	VOL1	P0 (IOL=10mA, VDD=4.5V)	-	-	1	V
Output High Voltage	VOH1	P0 (IOH=-8.57mA, VDD=4.5V)	3.5	-	-	V
Input High Leakage Current	IIH	P0,			1	uA
Input Low Leakage Current	IIL	P0	-1			uA
Pull-Up Resister	RPU	P0(VDD=5.0V)	20	-	62	kΩ
Power Supply Current	IDD1	Run Mode, fXIN=8MHz@5V	-	-	10	mA
	IDD2	Sleep Mode, fXIN=8MHz@5V	-	-	5	mA
	IDD3	Sub Active Mode, fXIN=32.768KHz@5V	-	-	500	uA
	IDD4	STOP1 Mode,WDT Active@5V	-	-	110	uA
	IDD5	STOP2 Mode,WDT Disable@5V	-	-	10	uA

Note) STOP1: WDT running STOP2: WDT disable

1.12.10 AC Characteristics

(VDD=5.0V±10%, VSS=0V, TA=-40~+85℃)

Table 1.13 AC Characteristics

Parameter	Symbol	PIN	MIN	TYP	MAX	Unit
Operating Frequency	fMCP	XIN	1	-	8	MHz
System Clock Cycle Time	tSYS	-	125	-	1000	ns
Oscillation Stabilization Time (8MHz)	tMST1	XIN, XOUT	-	-	10	ms
External Clock “H” or “L” Pulse Width	tCPW	XIN	90	-	-	ns
External Clock Transition Time	tRCP,tFCP	XIN	-	-	10	ns
Interrupt Input Width	tIW	INT0~INT1	2	-	-	tSYS
RESETB Input Pulse “L” Width	tRST	RESETB	8	-	-	tSYS
External Counter Input “H” or “L” Pulse Width	tECW	EC0	2	-	-	tSYS
Event Counter Transition Time	tREC,tFEC	EC0	-	-	20	ns

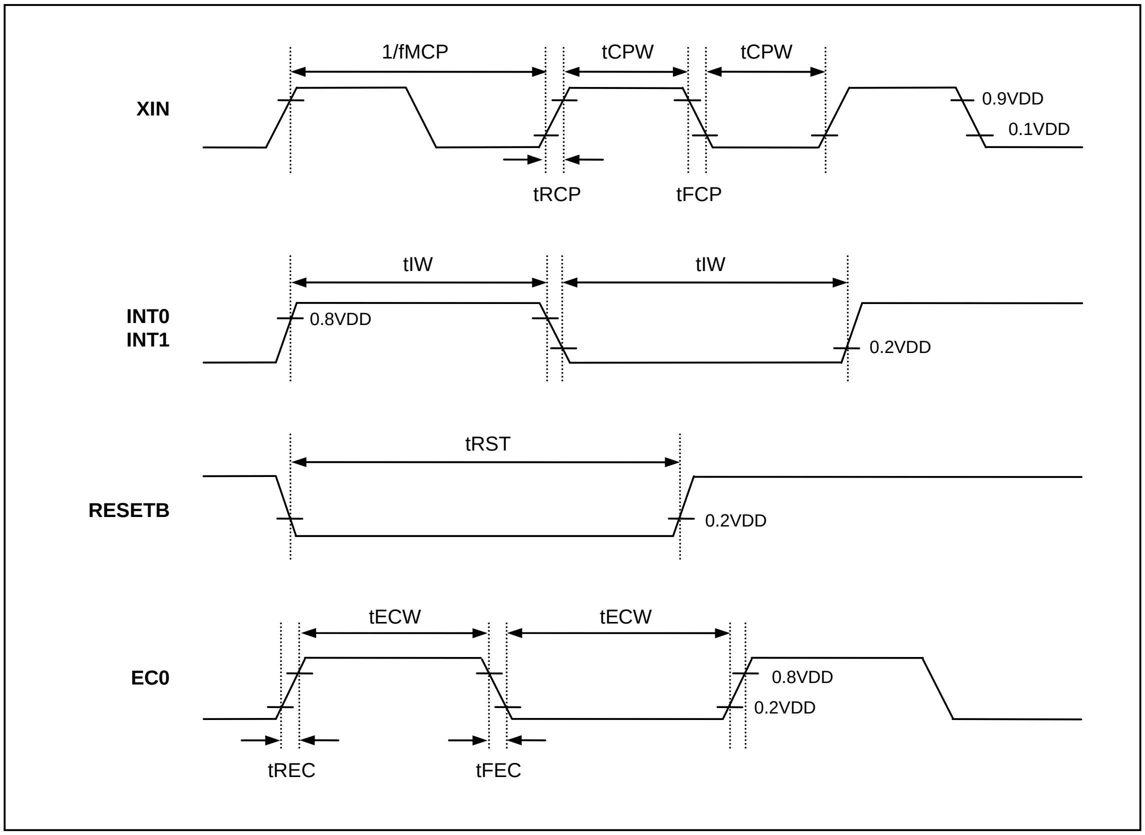


Figure 1.25 AC Timing

1.12.11 Typical Characteristics

These graphs and tables provided in this section are for design guidance only and are not tested or guaranteed. In some graphs or tables the data presented are outside specified operating range (e.g.

outside specified VDD range). This is for information only and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. “Typical” represents the mean of the distribution while “max” or “min” represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

2. Functional Description

2.1 Memory

The Z51F0410 MCU addresses two separate address memory stores: Program memory and Data memory. The logical separation of Program and Data memory allows Data memory to be assessed by 8-bit addresses, which can be more quickly stored and manipulated by 8-bit CPU. Nevertheless, 16-bit Data memory addresses can also be generated through the DPTR register.

Program memory can only be read, not written to. There can be up to 64K bytes of Program memory. In the Z51F0410 Flash version of these devices the 4K bytes of Program memory are provided on-chip. Data memory can be read and written to up to 256 bytes internal memory (DATA) including the stack area.

2.1.1 Program Memory

A 16-bit program counter is capable of addressing up to 64K bytes, but this device has just 4K bytes program memory space.

Figure 2.1 shows a map of the lower part of the program memory. After reset, the CPU begins execution from location 0000H. Each interrupt is assigned a fixed location in program memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External interrupt 0, for example, is assigned to location 000BH. If external interrupt 0 is going to be used, its service routine must begin at location 000BH. If the interrupt is not going to be used, its service location is available as general purpose program memory. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8 byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

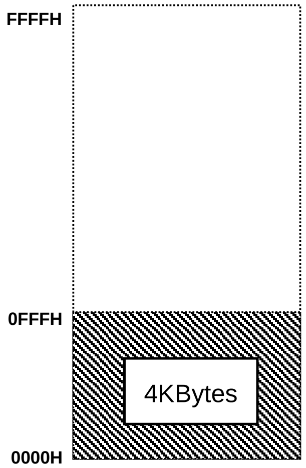


Figure 2.1 Program memory

- Nonvolatile and reprogramming memory: Flash memory based on EEPROM cell

2.1.2 Data Memory

Figure 2.2 shows the internal Data memory space available.

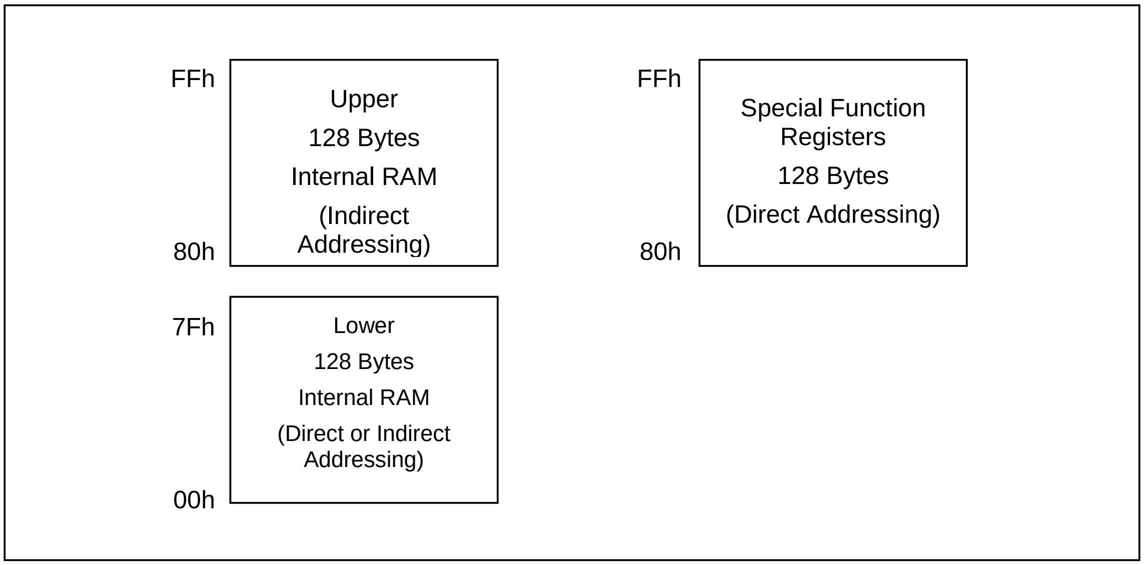


Figure 2.2 Data memory map

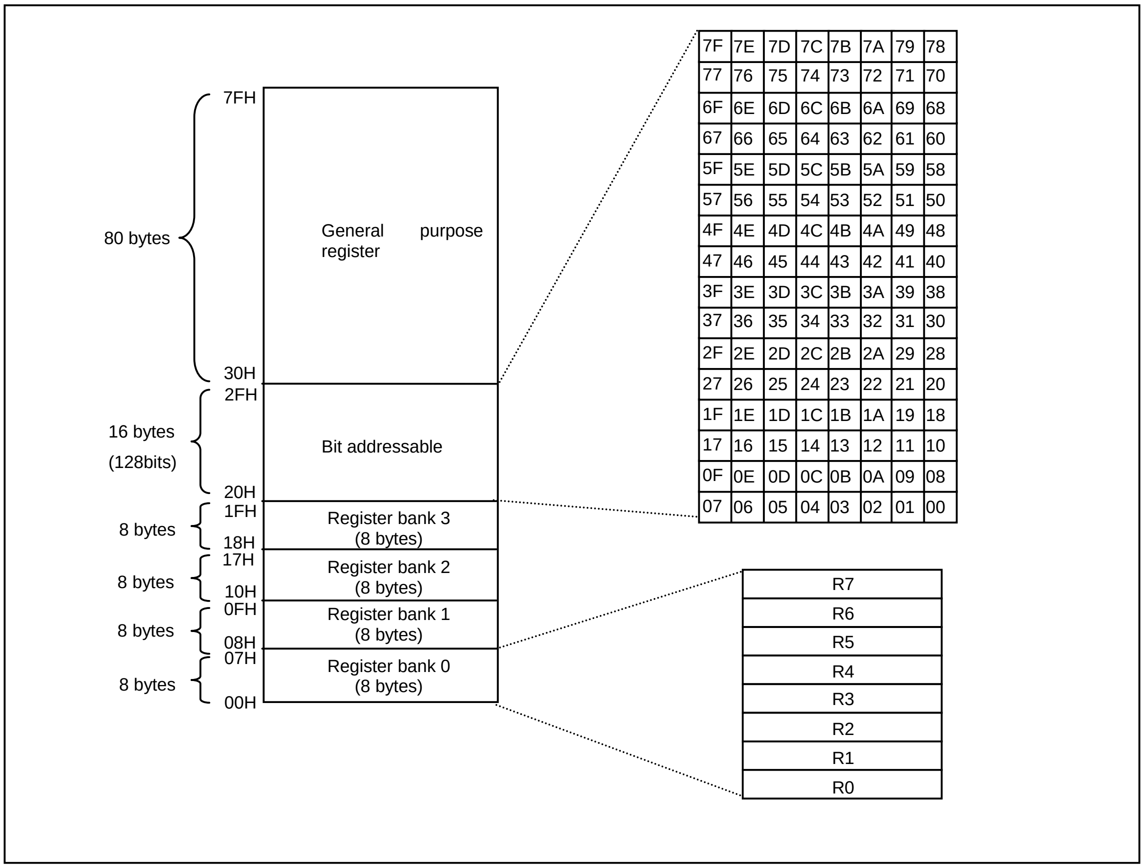
128, upper 128, and SFR space.

Internal Data memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space and indirect addresses higher than 7FH access a different memory space. Thus Figure 2.2 shows the upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128 bytes of RAM are present in all 8051 devices as mapped in Figure 2.3. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word select which register bank is in use. This allows more efficient used of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16 bytes above the register banks form a block of bit-addressable memory space. The 8051 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the lower 128 can be accessed by either direct or indirect addressing. The upper 128 bytes RAM can only be accessed by indirect addressing. These spaces are used for user RAM and stack pointer.



7F	7E	7D	7C	7B	7A	79	78
77	76	75	74	73	72	71	70
6F	6E	6D	6C	6B	6A	69	68
67	66	65	64	63	62	61	60
5F	5E	5D	5C	5B	5A	59	58
57	56	55	54	53	52	51	50
4F	4E	4D	4C	4B	4A	49	48
47	46	45	44	43	42	41	40
3F	3E	3D	3C	3B	3A	39	38
37	36	35	34	33	32	31	30
2F	2E	2D	2C	2B	2A	29	28
27	26	25	24	23	22	21	20
1F	1E	1D	1C	1B	1A	19	18
17	16	15	14	13	12	11	10
0F	0E	0D	0C	0B	0A	09	08
07	06	05	04	03	02	01	00

Figure 2.3 Lower 128 bytes RAM

2.1.3 EEPROM Data Memory

The Z51F0410 MCU features 256 bytes EEPROM Data memory. This area has no relation with RAM/Flash. It can read and write through SFR with 8-bit unit.

For more information about EEPROM Data memory, see EEPROM section

2.2 SFR Map

2.2.1 SFR Map Summary

-	*Reserved*
▨	*M8051 Compatible*

Table 2.1 SFR Map Summary

	0H/8H[1]	1H/9H	2H/AH	3H/BH	4H/CH	5H/DH	6H/EH	7H/FH
F8H	IP1	**FUSE_ CONF2**	**FUSE_ CAL2**	**FUSE_ CAL1**	**FUSE_ CAL0**	**FUSE_ CONF1**	TEST_B	TEST_A
F0H	B	-	FEARL	FEARM	FEARH	FEDR	-	**FUSE_CAL3**
E8H	-	ACCSR	FEMR	FECR	FESR	FETCR	UKEY0	UKEY1
E0H	ACC	-	UCTRL1	UCTRL2	UCTRL3	USTAT	UBAUD	UDATA
D8H	-	-	I2CMR	I2CSR	I2CSCLLR	I2CSCLHR	I2CSDAHR	I2CDR
D0H	PSW	**FKEY0**	**FKEY1-**	**USEED1**	**USEED0**	**SIDA**	**SIDD**	I2CAR
C8H	-	-				**T4H**	**T4CR**	**T4L**
C0H	P0DB	P0PC	-	-	-	-		
B8H	IP	-	-	-	-	-	-	-
B0H	-	-	T0CR	T0	T1CR	T1DR	T1	T1PWHR
A8H	IE	IE1	IE2	IE3	EIFLAG	EIEDGE	EIPOLA	EIENAB
A0H	**PSR1**	P0OD	EO	-	-	-	-	-
98H	P0IO		ADCM	ADCRH	ADCRL	WTMR	WTR	**PSR0**
90H	-	-	**PINMCR**	-	-	-	**BUZCR**	**TFLG**
88H	**P0PD**	P0PU	SCCR	BCCR	BITR	WDTMR	WDTR	**BUZDR**
80H	P0	SP	DPL	DPH			BODR	PCON

Note: 1) These registers are bit-addressable

2.2.2 SFR Map

Table 2.2 SFR Map

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
80H	Port 0 Data Register	P0	R/W	0	0	0	0	0	0	0	0
81H	Stack Pointer	SP	R/W	0	0	0	0	0	1	1	1
82H	Data Pointer Register Low	DPL	R/W	0	0	0	0	0	0	0	0
83H	Data Pointer Register High	DPH	R/W	0	0	0	0	0	0	0	0
84H	Reserved										
84H	Reserved										
86H	BOD Control Register	BODR	R/W	1	0	0	0	0	0	0	1
87H	Power Control Register	PCON	R/W	0	0	0	0	0	0	0	0
88H	Port0 PAD Data Register	P0PD	R	-	-	-	-	-	-	-	-
89H	Port 0 Pull-up Resistor Option Register	P0PU	R/W	0	0	0	0	0	0	0	0
8AH	System Clock Control Register	SCCR	R/W	0	0	0	0	0	1	0	0
8BH	BIT Clock Control Register	BCCR	R/W	0	-	-	-	0	1	0	1
8CH	Basic Interval Timer Register	BITR	R/W	0	0	0	0	0	0	0	0
8DH	Watch Dog Timer Mode Register	WDTMR	R/W	0	0	0	0	0	0	0	0
8EH	Watch Dog Timer Register	WDTR	W	1	1	1	1	1	1	1	1
	Watch Dog Timer Counter Register	WDTCR	R	0	0	0	0	0	0	0	0
8FH	Buzzer Data Register	BUZDR	R/W	1	1	1	1	1	1	1	1
90H	Reserved	-	-	-	-	-	-	-	-	-	-
91H	Reserved	-	-	-	-	-	-	-	-	-	-
92H	Pin Mux Control Register	PINMCR	R/W	-	-	-	0	0	0	0	0
93H	Reserved	-	-	-	-	-	-	-	-	-	-
94H	Reserved	-	-	-	-	-	-	-	-	-	-
95H	Reserved	-	-	-	-	-	-	-	-	-	-
96H	Buzzer Control Register	BUZCR	R/W	-	-	-	-	-	0	0	0
97H	TIMER 0,1,4 Interrupt Flag Register	TFLG	R/W	-	0	0	0	-	-	-	-
98H	Port 0 Direction Register	P0IO	R/W	0	0	0	0	0	0	0	0
99H	Reserved	-	-	-	-	-	-	-	-	-	-
9AH	A/D Converter Mode Register	ADCM	R/W	1	0	0	0	1	1	1	1
9BH	A/D Converter Result High Register	ADCRH	R	X	X	X	X	X	X	X	X
9CH	A/D Converter Result Low Register	ADCRL	R/W	0	1	0	0	-	-	X	X
9DH	Watch Timer Mode Register	WTMR	R/W	0	-	-	0	0	0	0	0
9EH	Watch Timer Register	WTR	W	0	1	1	1	1	1	1	1
	Watch Timer Counter Register	WTCR	R	0	0	0	0	0	0	0	0

9FH	Port Selection Register0	PSR0	R/W	0	0	0	0	0	0	0	0
A0H	Port Selection Register1	PSR1	R/W	0	0	0	0	0	0	0	0
A1H	Port 0 Open Drain Register	P0OD	R/W	0	0	0	0	0	0	0	0
A2H	Reserved	-	-	-	-	-	-	-	-	-	-
A3H	Reserved	-	-	-	-	-	-	-	-	-	-
A4H	Reserved	-	-	-	-	-	-	-	-	-	-
A5H	Reserved	-	-	-	-	-	-	-	-	-	-
A6H	Reserved	-	-	-	-	-	-	-	-	-	-
A7H	Reserved	-	-	-	-	-	-	-	-	-	-
A8H	Interrupt Enable Register 0	IE	R/W	0	0	0	0	0	0	0	0
A9H	Interrupt Enable Register 1	IE1	R/W	0	0	0	0	0	0	0	0
AAH	Interrupt Enable Register 2	IE2	R/W	0	0	0	0	0	0	0	0
ABH	Interrupt Enable Register 3	IE3	R/W	0	0	0	0	0	0	0	0
ACH	External Interrupt Flag Register	EIFLAG	R/W	-	-	-	-	0	0	0	0
ADH	External Interrupt Edge Register	EIEDGE	R/W	-	-	-	-	0	0	0	0
AEH	External Interrupt Polarity Register	EIPOLA	W	-	-	-	-	0	0	0	0
AFH	External Interrupt Enable Register	EIENAB	R/W	-	-	-	-	0	0	0	0
B0H	Reserved	-	-	-	-	-	-	-	-	-	-
B1H	Reserved	-	-	-	-	-	-	-	-	-	-
B2H	Timer 0 Mode Control Register	T0CR	R/W	0	0	0	0	0	0	0	0
B3H	Timer 0 Register	T0	R	0	0	0	0	0	0	0	0
	Timer 0 Data Register	T0DR	W	1	1	1	1	1	1	1	1
	Timer 0 Capture Data Register	CDR0	R	0	0	0	0	0	0	0	0
B4H	Timer 1 Mode Control Register	T1CR	R/W	0	0	0	0	0	0	0	0
B5H	Timer 1 Data Register	T1DR	W	1	1	1	1	1	1	1	1
	Timer 1 PWM Period Register	T1PPR	W	1	1	1	1	1	1	1	1
B6H	Timer 1 Register	T1	R	0	0	0	0	0	0	0	0
	Timer 1 RWM Duty Register	T1PDR	R/W	1	1	1	1	1	1	1	1
	Timer 1 Capture Data Register	CDR1	R	0	0	0	0	0	0	0	0
B7H	Timer 1 PWM Control Register	T1PWHR	R/W	0	0	0	0	0	0	0	0
B8H	Interrupt Priority Control Register 0	IP	R/W	0	0	0	0	0	0	0	0
B9H	Reserved	-	-	-	-	-	-	-	-	-	-
BAH	Reserved	-	-	-	-	-	-	-	-	-	-
BBH	Reserved	-	-	-	-	-	-	-	-	-	-
BCH	Reserved	-	-	-	-	-	-	-	-	-	-
BDH	Reserved	-	-	-	-	-	-	-	-	-	-

BEH	Reserved	-	-	-	-	-	-	-	-	-	-
BFH	Reserved	-	-	-	-	-	-	-	-	-	-
C0H	Port 0 Debounce Register	P0DB	R/W	0	0	0	0	0	0	0	0
C1H	Port 0 Pin Change Interrupt	P0PC	R/W	0	0	0	0	0	0	0	0
C2H	Reserved	-	-	-	-	-	-	-	-	-	-
C3H	Reserved	-	-	-	-	-	-	-	-	-	-
C4H	Reserved	-	-	-	-	-	-	-	-	-	-
C5H	Reserved	-	-	-	-	-	-	-	-	-	-
C6H	Reserved		-	-	-	-	-	-	-	-	-
C7H	Reserved		-	-	-	-	-	-	-	-	-
C8H	Reserved	-	-	-	-	-	-	-	-	-	-
C9H	Reserved	-	-	-	-	-	-	-	-	-	-
CAH	Reserved		-	-	-	-	-	-	-	-	-
CBH	Reserved		-	-	-	-	-	-	-	-	-
CCH	Reserved		-	-	-	-	-	-	-	-	-
CDH	Timer 4 Data High Register	T4H	R	0	0	0	0	0	0	0	0
CEH	Timer 4 Mode Control Register	T4CR	R/W	0	0	0	0	0	0	0	0
CFH	Timer 4 Data Low Register	T4L	R/W	0	0	0	0	0	0	0	0
D0H	Program Status Word Register	PSW	R/W	0	0	0	0	0	0	0	0
D1H	Authetification FAB Key	AUTH_FKEY0	R/W	0	0	0	0	0	0	0	0
D2H	Authetification FAB Key	AUTH_FKEY1	R/W	0	0	0	0	0	0	0	0
D3H	USER SEED0[7:0]	USEED0	R/W	0	0	0	0	0	0	0	0
D4H	USER_SEED1[15:8]	USEED1	R/W	0	0	0	0	0	0	0	0
D5H	SID Access Addresss	SIDA	R	-	-	-	-	-	-	-	-
D6H	Current SID Data Value	SIDD	R	-	-	-	-	-	-	-	-
D7H	I2C Slave Address Register	I2CAR	R/W	0	0	0	0	0	0	0	0
D8H	Reserved	-	-	-	-	-	-	-	-	-	-
D9H	Reserved	-	-	-	-	-	-	-	-	-	-
DAH	I2C Mode Control Register	I2CMR	R/W	0	0	0	0	0	0	0	0
DBH	I2C Status Register	I2CSR	R	0	0	0	0	0	0	0	0
DCH	I2C SCL Low Period Register	I2CSCLLR	R/W	0	0	1	1	1	1	1	1
DDH	I2C SCL High Period Register	I2CSCLHR	R/W	0	0	1	1	1	1	1	1
DEH	I2C SDA Hold Time Register	I2CSDAHR	R/W	0	0	0	0	0	0	1	1
DFH	I2C Data Register	I2CDR	R/W	1	1	1	1	1	1	1	1
E0H	Accumulator Register	ACC	R/W	0	0	0	0	0	0	0	0
E1H	Reserved	-	-	-	-	-	-	-	-	-	-
E2H	USART Control Register 1	UCTRL1	R/W	0	0	0	0	0	0	0	0
E3H	USART Control Register 2	UCTRL2	R/W	0	0	0	0	0	0	0	0
E4H	USART Control Register 3	UCTRL3	R/W	0	0	0	0	0	0	0	0
E5H	USART Status Register	USTAT	R/W	1	0	0	0	-	0	0	0
E6H	USART Baud Rate Generation Register	UBAUD	R/W	1	1	1	1	1	1	1	1
E7H	USART Data Register	UDATA	R/W	1	1	1	1	1	1	1	1

E8H	Reserved	-	-	-	-	-	-	-	-	-	-
E9H	**Analog Comparator Control & Status Register**	**ACCSR**	R/W	0	0	0	0	0	0	0	0
EAH	**Flash and EEPROM Mode Register**	**FEMR**	R/W	0	0	0	0	0	0	0	0
EBH	**Flash and EEPROM Control Register**	**FECR**	R/W	0	0	0	0	0	0	1	1
ECH	**Flash and EEPROM Status Register**	**FESR**	R/W	1	0	0	0	0	0	0	0
EDH	**Flash and EEPROM Timer Control Register**	**FETCR**	R/W	0	0	0	0	0	0	0	0
EEH	**Authetification Key LSB**	**AUTH_UKEY0**	**R/W**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
EFH	**Authetification Key MSB**	**AUTH_UKEY1**	**R/W**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F0H	**B Register**	**B**	R/W	0	0	0	0	0	0	0	0
F1H	Reserved	-	-	-	-	-	-	-	-	-	-
F2H	**Flash and EEPROM Address Low Register**	**FEARL**	W	0	0	0	0	0	0	0	0
F3H	**Flash and EEPROM Address Middle Register**	**FEARM**	W	0	0	0	0	0	0	0	0
F4H	**Flash and EEPROM Address High Register**	**FEARH**	W	0	0	0	0	0	0	0	0
F5H	**Flash and EEPROM Data Register**	**FEDR**	R/W	0	0	0	0	0	0	0	0
F6H	Reserved	-	-	-	-	-	-	-	-	-	-
F7H	**VDC Trimming for RCOSC 128Khz**	**FUSE_CAL3**	R/W	0	0	0	0	0	0	0	0
F8H	**Interrupt Priority Control Register 1**	**IP1**	R/W	0	0	0	0	0	0	0	0
F9H	**Configuration Option 1**	**FUSE_CONF2**	R/W	-	-	-	-	-	-	-	0
FAH	**BGR and BOD Calibration Data**	**FUSE_CAL2**	R/W	0	0	0	0	0	0	0	0
FBH	**INTOSC Calibration Data**	**FUSE_CAL1**	R/W	1	0	0	0	0	0	0	0
FCH	**VDC Trimming for INTOSC 8Mhz**	**FUSE_CAL0**	R/W	0	0	0	0	0	0	0	0
FDH	**Configuration Option 0**	**FUSE_CONF1**	R/W	-	-	-	0	0	0	0	0
FEH	**Function Test Register B**	**TEST_B**	R/W	0	0	0	0	0	0	0	0
FFH	**Function Test Register A**	**TEST_A**	R/W	0	0	0	0	0	0	0	1

2.2.3 Compiler Compatible SFR

ACC (Accumulator) : E0H

7	6	5	4	3	2	1	0
ACC							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

ACC Accumulator

B (B Register) : F0H

7	6	5	4	3	2	1	0
B							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

B B Register

SP (Stack Pointer) : 81H

7	6	5	4	3	2	1	0
SP							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 07H

SP Stack Pointer

DPL (Data Pointer Low Byte) : 82H

7	6	5	4	3	2	1	0
DPL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPL Data Pointer Low Byte

DPH (Data Pointer High Byte) : 83H

7	6	5	4	3	2	1	0
DPH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPH Data Pointer High Byte

PSW (Program Status Word) : D0H

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CY Carry Flag

AC Auxiliary Carry Flag

F0 General Purpose User-Definable Flag

RS1 Register Bank Select bit 1

RS0 Register Bank Select bit 0

OV Overflow Flag

F1 User-Definable Flag

P Parity Flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of ‘1’ bits in the accumulator

2.3 I/O Port

2.3.1 I/O Ports

The Z51F0410 MCU features one I/O ports (P0). Each port can be easily configured by software as I/O pin, internal pull up and open drain pin to meet various system configurations and design requirements. Also P0 includes function that can generate interrupt according to change of state of the pin.

2.3.2 Port Register

2.3.2.1 Data Register (P0)

Data Register is a bidirectional I/O port. If ports are configured as output ports, data can be written to the corresponding bit of the P0. If ports are configured as input ports, the data can be read from the corresponding bit of the P0.

2.3.2.2 Direction Register (P0IO)

Each I/O pin can independently used as an input or an output through the P0IO register. Bits cleared in this read/write register will select the corresponding pin in P0 to become an input, setting a bit sets the pin to output. All bits are cleared by a system reset.

2.3.2.3 Pull-up Resistor Selection Register (P0PU)

The on-chip pull-up resistor can be connected to them in 1-bit units with a pull-up resistor selection register (P0PU). The pull-up register selection controls the pull-up resister enable/disable of each port. When the corresponding bit is 1, the pull-up resister of the pin is enabled. When 0, the pull-up resister is disabled. All bits are cleared by a system reset.

2.3.2.4 Open-drain Selection Register (P0OD)

There is internally open-drain selection register (P0OD) in P0. The open-drain selection register controls the open-drain enable/disable of each port. Ports become push-pull by a system reset. You should connect an external resistor in open-drain output mode.

2.3.2.5 Debounce Enable Register (P0DB)

P0 support debounce function. Debounce time of each ports has 1us, but if P0[2] uses external reset function, it has 3us debounce time. (except P0[2], other port initialization state is OFF)

2.3.2.6 Pin Change Interrupt Enable Register (P0PC)

The P0 can support Pin Change Interrupt function. Pin Change Interrupts PCI will trigger if any enabled P0[7:0] pin toggles. The P0PC Register control which pins contribute to the pin change interrupts.

2.3.2.7 Pin Mux Control Register (PINMCR)

In the 10pin PKG, The secondary pin muxing function is added for pin efficiency.

2.3.2.8 Pin PAD Data Register (P0PD)

It is used to read directly PAD data regardless of port direction.

2.3.2.9 PORT Selection Register0(PSR0, PSR1)

ADC Channel Selection (PSR0), and Comparator Output Selection (PSR1) disables the logic input gate to prevent the leakage current.

2.3.2.10 Register Map

Table 2.3 Register map

Name	Address	Dir	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	98H	R/W	00H	P0 Direction Register
P0PU	89H	R/W	00H	P0 Pull-up Resistor Selection Register
P0OD	A1H	R/W	00H	P0 Open-drain Selection Register
P0DB	C0H	R/W	00H	P0 Debounce Enable Register
P0PC	C1H	R/W	00H	P0 Pin Change Interrupt Enable Register
P0PD	88H	R/W	00H	P0 PAD Data Register
PINMCR	92H	R/W	00H	Pin Mux Control Register

2.3.3 P0 Port

2.3.3.1 P0 Port Description

P0 is 8-bit I/O port. P0 control registers consist of Data register (P0), direction register (P0IO), debounce enable register (P0DB, P2DB), pull-up register selection register (P0PU), open-drain selection register (P0OD).

2.3.3.2 Register description for P0

P0 (P0 Data Register) : 80H

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0[7:0] I/O Data

P0IO (P0 Direction Register) : 98H

7	6	5	4	3	2	1	0
P07IO	P06IO	P05IO	P04IO	P03IO	P02IO	P01IO	P00IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0IO[7:0] P0 data I/O direction.
0 Input
1 Output

P0PU (P0 Pull-up Resistor Selection Register) : 89H

7	6	5	4	3	2	1	0
P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0PU[7:0] Configure pull-up resistor of P0 port
0 disable
1 enable

P0OD (P0 Open-drain Selection Register) : A1H

7	6	5	4	3	2	1	0
P07OD	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0OD[7:0] Configure open-drain of P0 port
0 disable
1 enable

P0DB (P0 Debounce Enable Register) : C0H

7	6	5	4	3	2	1	0
P07DB	P06DB	P05DB	P04DB	P03DB	P02DB	P01DB	P00DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0DB[7:0] Configure debounce of P0 port
0 disable
1 enable

P0PC (P0 Pin Change Interrupt Enable Register) : C1H

7	6	5	4	3	2	1	0
P07PC	P06PC	P05PC	P04PC	P03PC	P02PC	P01PC	P00PC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0PC[7:0]	Configure Pin Change Interrupt of P0 port
	0 disable
	1 enable

PSR0 (ADC Pin Selection Register) : 9FH

7	6	5	4	3	2	1	0
AIN07_EN	AIN06_EN	AIN05_EN	AIN04_EN	AIN03_EN	AIN02_EN	AIN01_EN	AIN00_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PSR0[7:0]	ADC Channel Selection (Disable logic input gate)
	0 disable
	1 enable

PSR1 (Comparator Pin Selection Register) : A0H

7	6	5	4	3	2	1	0
							ACO_EN
							R/W

Initial value : 00H

PSR1[0]	Analog Comparator Output Enable (Disable logic input gate)
	0 disable
	1 enable

P0PD (P0 PAD Data Register) : 88H

7	6	5	4	3	2	1	0
P0PD7	P0PD6	P0PD5	P0PD4	P0PD3	P0PD2	P0PD1	P0PD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0PD[7:0]	PAD input data

PINMCR (Pin Mux Control Register) : 92H

7	6	5	4	3	2	1	0
-	-	-	TMR_CFG	BUZ_CFG	I2C_CFG	USART_CFG	XINT_CFG
-	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

TR_CFG	Timer Pin Control Mode
BUZ_CFG	BUZZER Pin Control Mode
I2C_CFG	I2C Pin Control Mode
USART_CFG	USART Pin Control Mode
XINT_CFG	External Interrupt Pin Control Mode

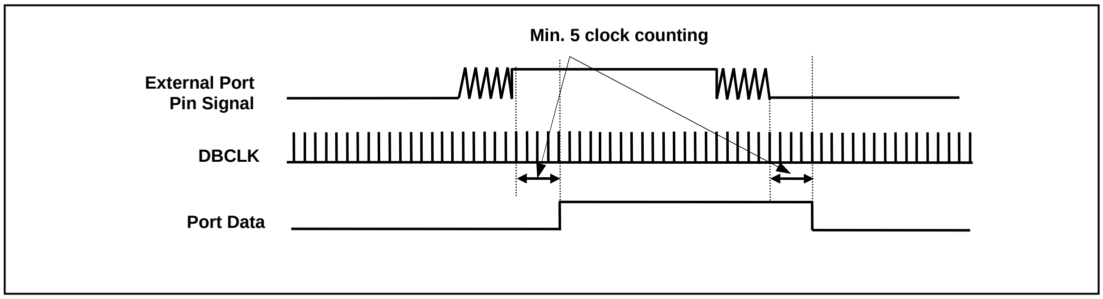


Figure 2.4 Debounce Function

3. Interrupt Controller

3.1 Overview

The Z51F0410 MCU supports up to 15 interrupt sources. The interrupts have separate enable register bits associated with them, allowing software control. They can also have four levels of priority assigned to them. The nonmaskable interrupt source is always enabled with a higher priority than any other interrupt source, and is not controllable by software. The interrupt controller has following features:

- receive the request from 24 interrupt source
- 6 group priority
- 4 priority levels
- Multi Interrupt possibility
- If the requests of different priority levels are received simultaneously, the request of higher priority level is serviced
- Each interrupt source can control by EA bit and each IEx bit
- Interrupt latency: 5–8 machine cycles in single interrupt system

The nonmaskable interrupt is always enabled. The maskable interrupts are enabled through four pair of interrupt enable registers (IE, IE1, IE2, IE3). Bits of IE, IE1, IE2, IE3 register each individually enable/disable a particular interrupt source. Overall control is provided by bit 7 of IE (EA). When EA is set to ‘0’, all interrupts are disabled: when EA is set to ‘1’, interrupts are individually enabled or disabled through the other bits of the interrupt enable registers. The Z51F0410 MCU supports a four-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels by writing to IP or IP1.

External interrupt default mode is level-trigger basically but if needed, it is able to change edge-trigger mode. Table 3.1 shows the Interrupt Group Priority Level that is available for sharing interrupt priority. Priority sets two bit which is to IP and IP1 register about group. Interrupt service routine services higher priority. If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If the request of same or lower priority level is received, that request is not serviced.

Table 3.1 Interrupt Group Priority Level

Interrupt Group	**Highest** →			**Lowest**	
0 (Bit0)	**Interrupt0**	**Interrupt6**	**Interrupt12**	**Interrupt18**	**Highest**
1 (Bit0)	**Interrupt1**	**Interrupt7**	**Interrupt13**	**Interrupt19**	↓
2 (Bit0)	**Interrupt2**	**Interrupt8**	**Interrupt14**	**Interrupt20**	
3 (Bit0)	**Interrupt3**	**Interrupt9**	**Interrupt15**	**Interrupt21**	
4 (Bit0)	**Interrupt4**	**Interrupt10**	**Interrupt16**	**Interrupt22**	
5 (Bit0)	**Interrupt5**	**Interrupt11**	**Interrupt17**	**Interrupt23**	**Lowest**

3.2 External Interrupt

The external interrupt on INT0, INT1 pins receive various interrupt request depending on the edge selection register EIEDGE (External Interrupt Edge register) and EIPOLA (External Interrupt Polarity register) as shown in Figure 3.1. Also each external interrupt source has control setting bits. The

EIFLAG (External interrupt flag register) register provides the status of external interrupts.

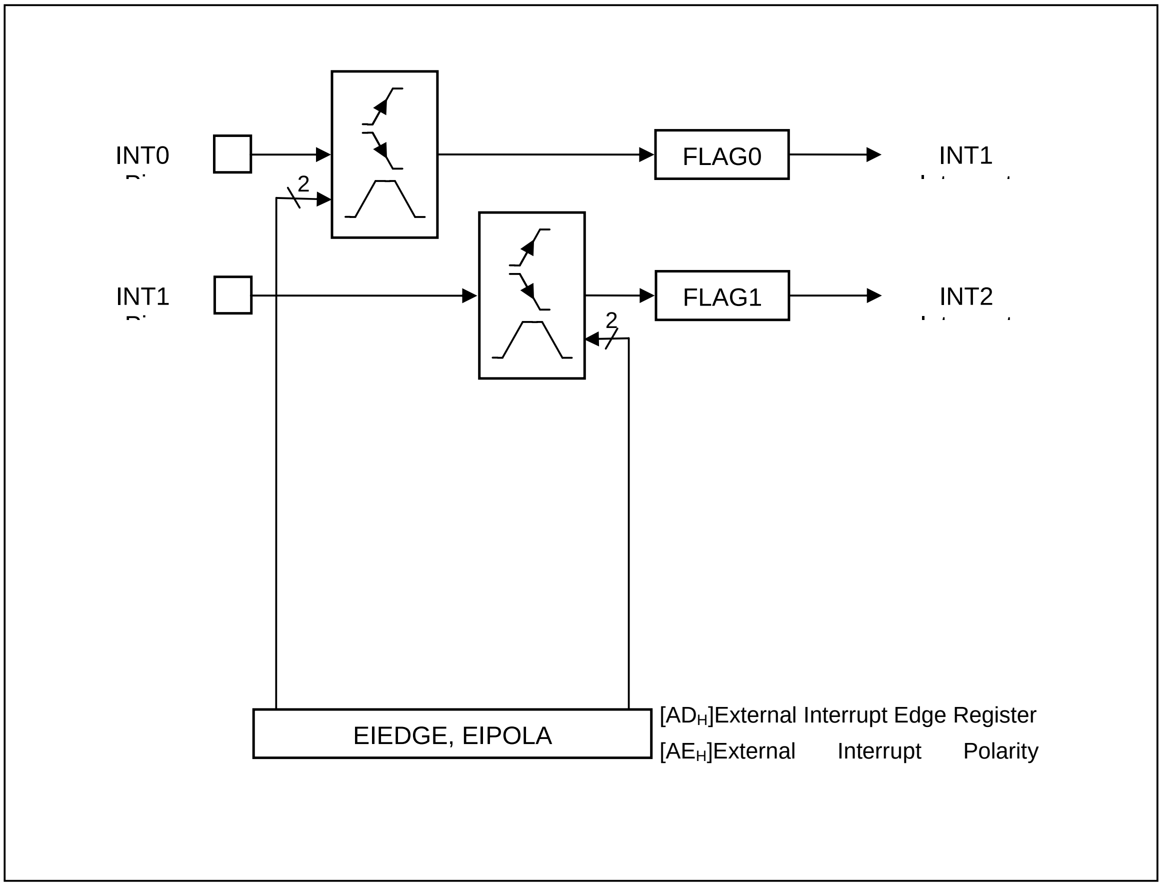

INT0
2
FLAG0
INT1
INT1
FLAG1
2
INT2
EIEDGE, EIPOLA
[AD$_H$]External Interrupt Edge Register
[AE$_H$]External Interrupt Polarity

Figure 3.1 External Interrupt Description

3.3 Block Diagram

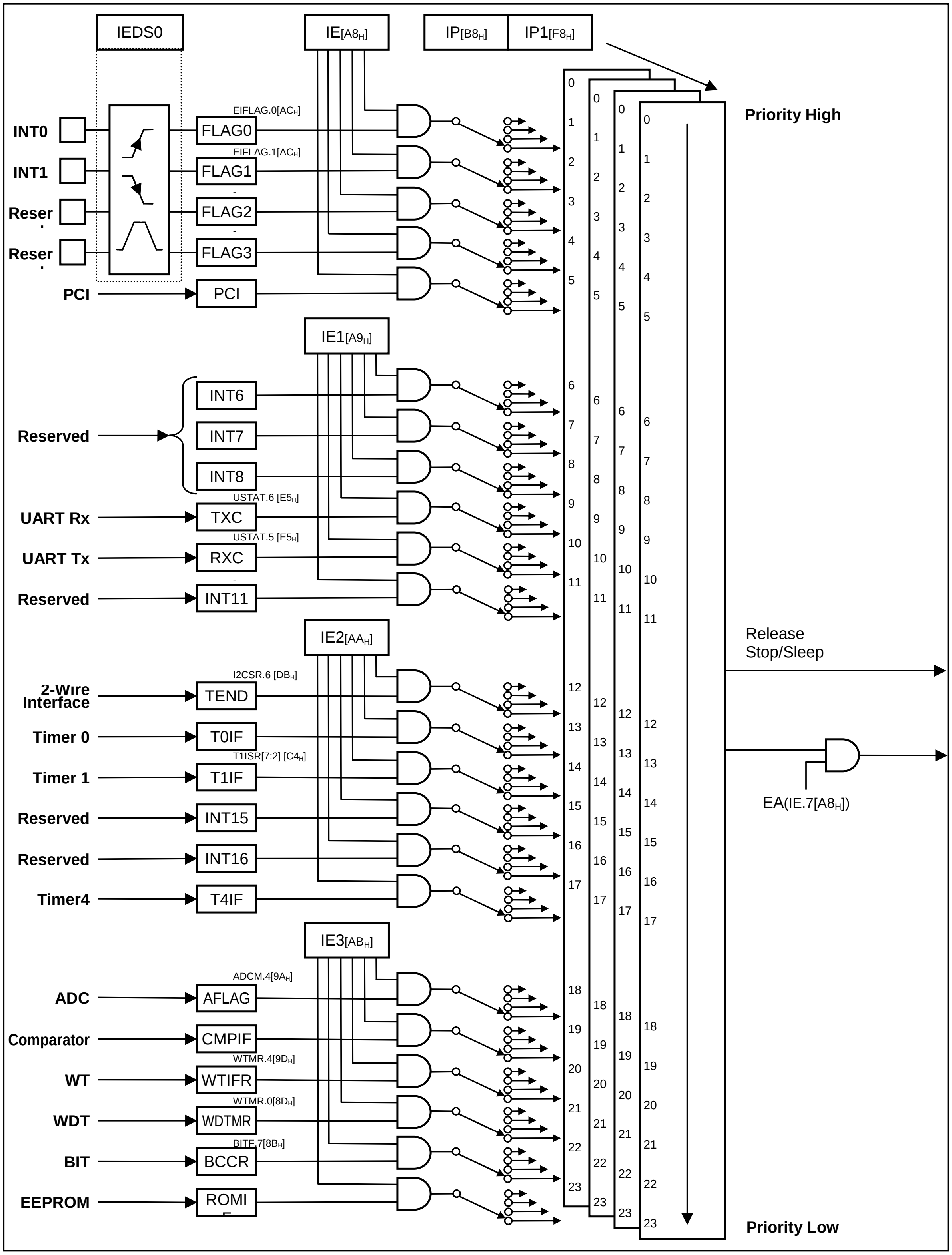


Figure 3.2 Block Diagram of Interrupt

3.4 Interrupt Vector Table

The interrupt controller supports 24 interrupt sources as shown in the Table 3.2 below. When interrupt becomes service, long call instruction (LCALL) is executed in the vector address. Interrupt request 24 has a decided priority order.

Table 3.2 Interrupt Vector Address Table

Interrupt Source	Symbol	Interrupt Enable Bit	Polarity	Mask	Vector Address
Hardware Reset	RESETB	0	0	NonMaskable	0000H
-	INT0	IE0.0	1	Maskable	0003H
External Interrupt 0	INT1	IE0.1	2	Maskable	000BH
External Interrupt 1	INT2	IE0.2	3	Maskable	0013H
-	INT3	IE0.3	4	Maskable	001BH
-	INT4	IE0.4	5	Maskable	0023H
Pin Change Interrupt (P0)	INT5	IE0.5	6	Maskable	002BH
-	INT6	IE1.0	7	Maskable	0033H
-	INT7	IE1.1	8	Maskable	003BH
-	INT8	IE1.2	9	Maskable	0043H
UART Rx	INT9	IE1.3	10	Maskable	004BH
UART Tx	INT10	IE1.4	11	Maskable	0053H
-	INT11	IE1.5	12	Maskable	005BH
I2C	INT12	IE2.0	13	Maskable	0063H
T0	INT13	IE2.1	14	Maskable	006BH
T1	INT14	IE2.2	15	Maskable	0073H
-	INT15	IE2.3	16	Maskable	007BH
-	INT16	IE2.4	17	Maskable	0083H
T4	INT17	IE2.5	18	Maskable	008BH
ADC	INT18	IE3.0	19	Maskable	0093H
Analog Comparator	INT19	IE3.1	20	Maskable	009BH
WT	INT20	IE3.2	21	Maskable	00A3H
WDT	INT21	IE3.3	22	Maskable	00ABH
BIT	INT22	IE3.4	23	Maskable	00B3H
EEPROM	INT23	IE3.5	24	Maskable	00BBH

For maskable interrupt execution, first EA bit must set ‘1’ and specific interrupt source must set ‘1’ by writing a ‘1’ to associated bit in the IEx. If interrupt request is received, specific interrupt request flag set ‘1’. And it remains ‘1’ until CPU accepts interrupt. After that, interrupt request flag will be cleared automatically.

3.5 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to ‘0’ by a reset or an instruction. Interrupt acceptance always generates at last cycle of the instruction. So instead of fetching the current instruction, CPU executes internally LCALL instruction and saves the PC stack. For the interrupt service routine, the interrupt controller gives the address of LJMP instruction to CPU. After finishing the current instruction, at the next instruction to go interrupt service routine needs 3–9 machine cycle and the interrupt service task is terminated upon execution of an interrupt return instruction [RETI]. After generating interrupt, to go to interrupt service routine, the following process is progressed

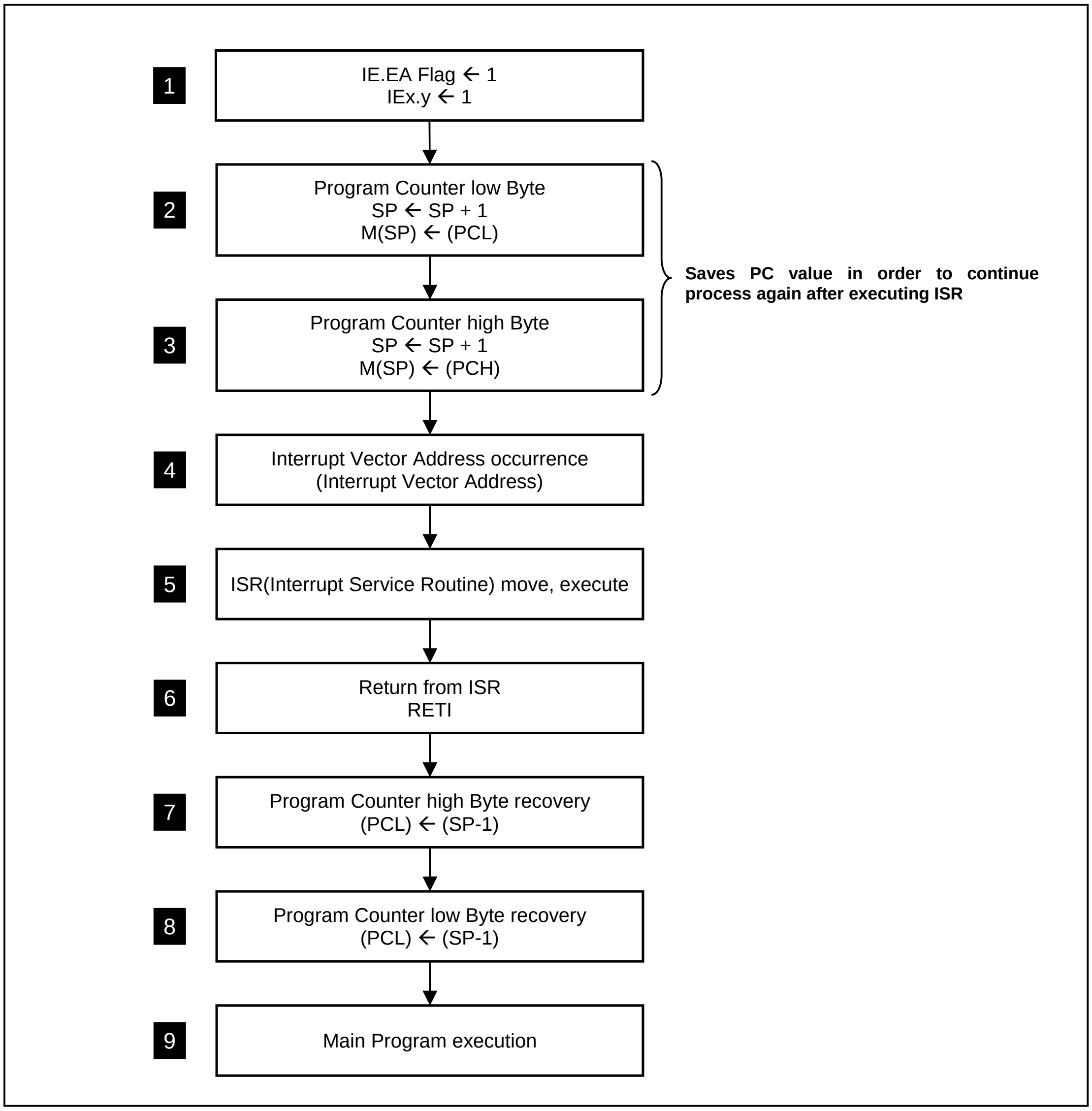

1
IE.EA Flag ← 1
IEx.y ← 1
2
Program Counter low Byte
SP ← SP + 1
M(SP) ← (PCL)
3
Program Counter high Byte
SP ← SP + 1
M(SP) ← (PCH)
Saves PC value in order to continue process again after executing ISR
4
Interrupt Vector Address occurrence
(Interrupt Vector Address)
5
ISR(Interrupt Service Routine) move, execute
6
Return from ISR
RETI
7
Program Counter high Byte recovery
(PCL) ← (SP-1)
8
Program Counter low Byte recovery
(PCL) ← (SP-1)
9
Main Program execution

Figure 3.3 Interrupt Vector Address Table

3.6 Effective Timing after Controlling Interrupt bit

Case a) Control Interrupt Enable Register (IE, IE1, IE2, IE3)

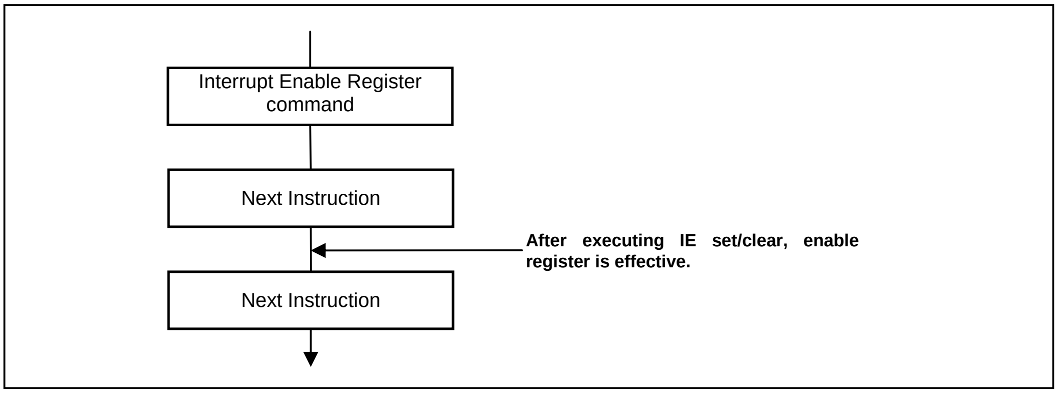

Interrupt Enable Register command
Next Instruction
After executing IE set/clear, enable register is effective.
Next Instruction

Figure 3.4 Effective Timing of Interrupt Enable Register

Case b) Interrupt flag Register

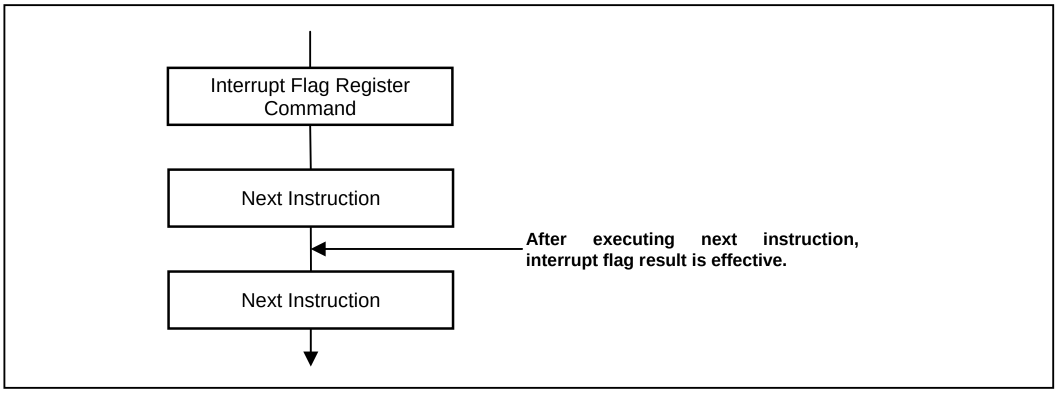

Interrupt Flag Register Command
Next Instruction
After executing next instruction, interrupt flag result is effective.
Next Instruction

Figure 3.5 Effective Timing of Interrupt Flag Register

3.7 Multi Interrupt

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the interrupt are received at the same time simultaneously, an interrupt polling sequence determines by hardware which request is serviced. However, multiple processing through software for special features is possible.

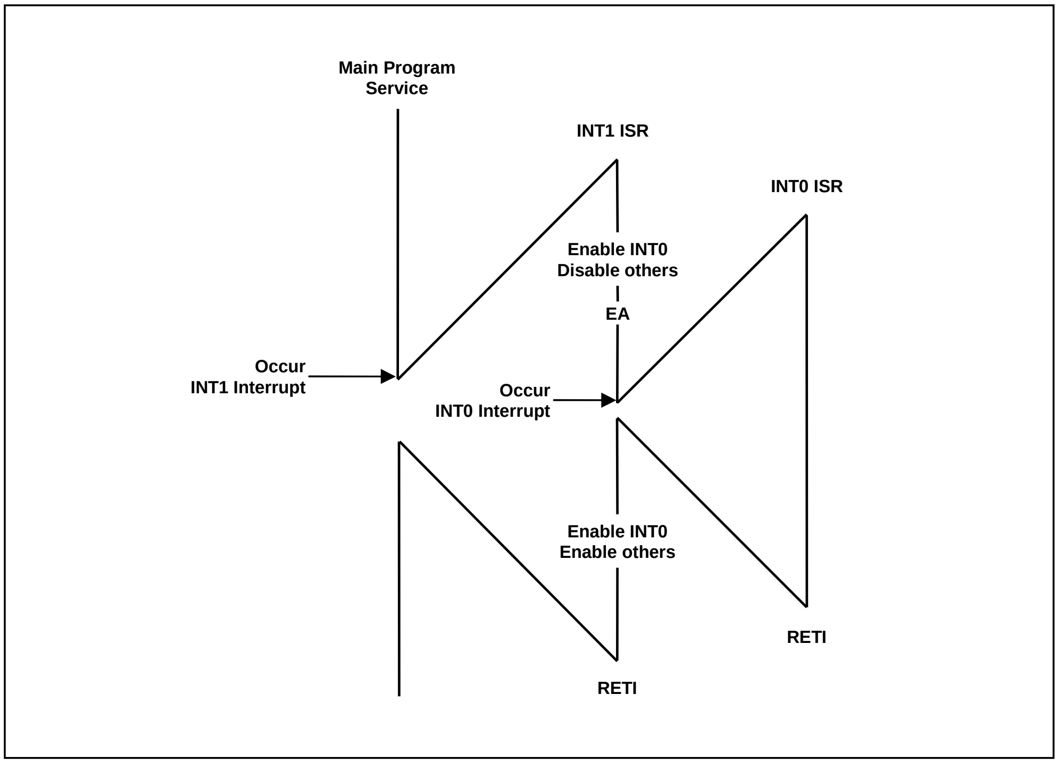


Figure 3.6 Execution of Multi Interrupt

Following example is shown to service INT0 routine during INT1 routine in Figure 3.6. In this example, INT0 interrupt priority is higher than INT1 interrupt priority. If some interrupt is lower than INT1 priority, it can't service its interrupt routine.

Example) Software Multi Interrupt:

```
INT1:    MOV     IE, #81H      ; Enable INT0 only
         MOV     IE1, #00H     ; Disable other
         :
         MOV     IE, #0FFH     ; Enable all Interrupts
         MOV     IE1, #0FFH
         RETI
```

3.8 Interrupt Enable Accept Timing

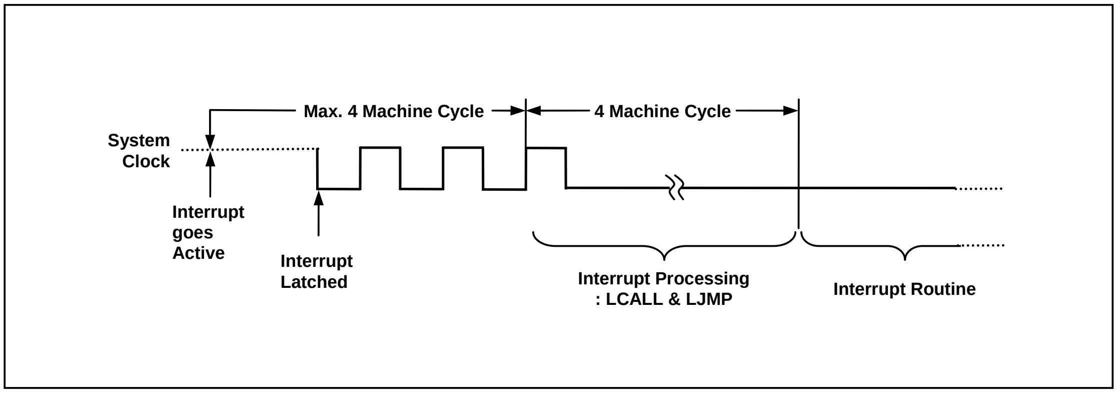


Figure 3.7 Interrupt Response Timing Diagram

3.9 Interrupt Service Routine Address

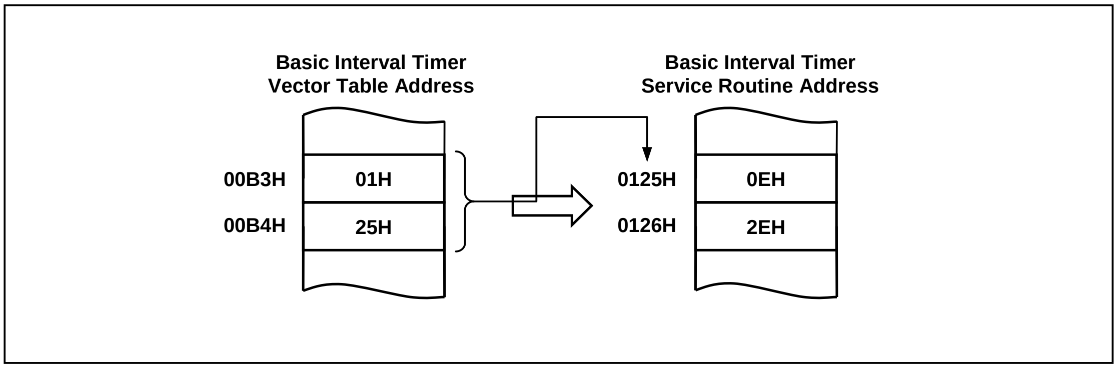


Figure 3.8 Correspondence between vector Table address and the entry address of ISP

3.10 Saving/Restore General-Purpose Registers

```
INTxx : PUSH       PSW
        PUSH       DPL
        PUSH       DPH
        PUSH         B
        PUSH  ACC
        :

Interrupt_Processing:
         :
         :
         POP       ACC
        POP          B
        POP        DPH
        POP        DPL
        POP        PSW
        RETI
```

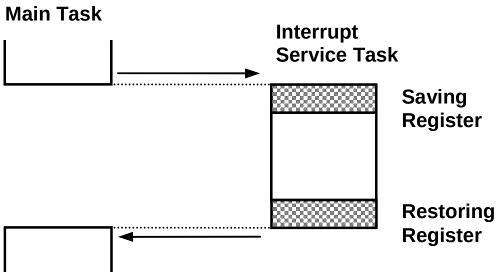


Figure 3.9 Saving/Restore Process Diagram & Sample Source

3.11 Interrupt Timing

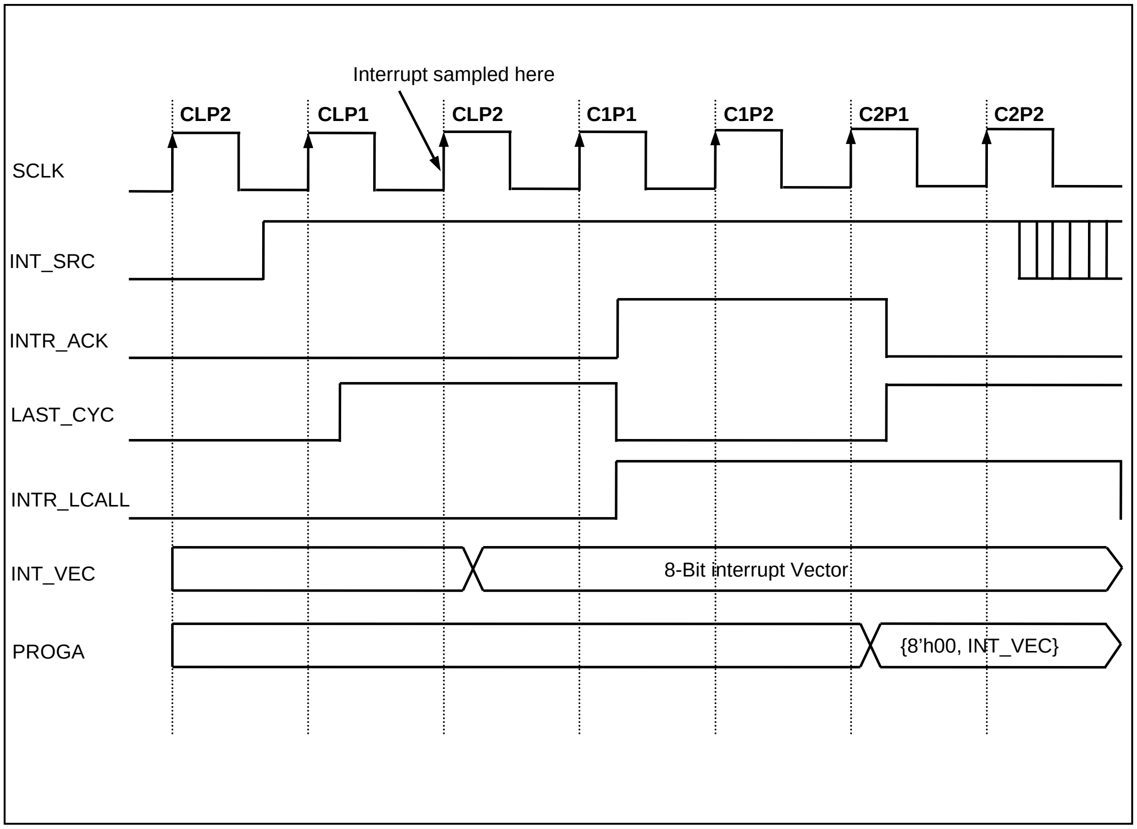


Figure 3.10 Timing chart of Interrupt Acceptance and Interrupt Return Instruction

Interrupt source sampled at last cycle of the command. When sampling interrupt source, it is decided to low 8-bit of interrupt vector. M8051W core makes interrupt acknowledge at first cycle of command, executes long call to jump interrupt routine as INT_VEC.

Note) command cycle C?P?: L=Last cycle, 1=1st cycle or 1st phase, 2=2nd cycle or 2nd phase

3.12 Interrupt Register Overview

3.12.1 Interrupt Enable Register (IE, IE1, IE2, IE3)

Interrupt enable register consists of Global interrupt control bit (EA) and peripheral interrupt control bits. Totally 24 peripheral are able to control interrupt.

3.12.2 Interrupt Priority Register (IP, IP1)

The 24 interrupt divides 6 groups which have each 4 interrupt sources. A group can decide 4 levels interrupt priority using interrupt priority register. Level 3 is the high priority, while level 0 is the low priority. Initially, IP, IP1 reset value is '0'. At that initialization, low interrupt number has a higher priority than high interrupt number. If decided the priority, low interrupt number has a higher priority than high interrupt number in that group.

3.12.3 External Interrupt Flag Register (EIFLAG)

The external interrupt flag register is set to ‘1’ when the external interrupt generating condition is satisfied. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a ‘0’ to it.

3.12.4 External Interrupt Edge Register (EIEDGE)

The External interrupt edge register determines which type of edge or level sensitive interrupt. Initially, default value is level. For level, write ‘0’ to related bit. For edge, write ‘1’ to related bit.

3.12.5 External Interrupt Polarity Register (EIPOLA)

According to EIEDGE register, the external interrupt polarity (EIPOLA) register has a different meaning. If EIEDGE is level type, EIPOLA is able to have Low/High level value. If EIEGDE is edge type, EIPOLA is able to have rising/falling edge value.

3.12.6 External Interrupt Enable Register (EIENAB)

When the external interrupt enable register is written to ‘1’, the corresponding external pin interrupt is enabled. The EIEDGE and EIPOLA register defines whether the external interrupt is activated on rising or falling edge or level sensed.

3.12.7 Register Map

Table 3.3 Register Map

Name	Address	Dir	Default	Description
IE	A8H	R/W	00H	Interrupt Enable Register
IE1	A9H	R/W	00H	Interrupt Enable Register 1
IE2	AAH	R/W	00H	Interrupt Enable Register 2
IE3	ABH	R/W	00H	Interrupt Enable Register 3
IP	B8H	R/W	00H	Interrupt Polarity Register
IP1	F8H	R/W	00H	Interrupt Polarity Register 1
EIFLAG	ACH	R/W	00H	External Interrupt Flag Register
EIEDGE	ADH	R/W	00H	External Interrupt Edge Register
EIPOLA	AEH	R/W	00H	External Interrupt Polarity Register
EIENAB	AFH	R/W	00H	External Interrupt Enable Register

3.13 Interrupt Register Description

The Interrupt Register is used for controlling interrupt functions. Also it has External interrupt control registers. The interrupt register consists of Interrupt Enable Register (IE), Interrupt Enable Register 1 (IE1), Interrupt Enable Register 2 (IE2) and Interrupt Enable Register 3 (IE3). For external interrupt, it consists of External Interrupt Flag Register (EIFLAG), External Interrupt Edge Register (EIEDGE), External Interrupt Polarity Register (EIPOLA) and External Interrupt Enable Register (EIENAB).

3.13.1 Register description for Interrupt

IE (Interrupt Enable Register) : A8H

7	6	5	4	3	2	1	0

EA	-	INT5E	-	-	INT2E	INT1E	-
R/W	R	R/W-	R/	R	R/W	R/W	R/

Initial value : 00H

EA Enable or disable all interrupt bits
0 All Interrupt disable
1 All Interrupt enable

INT5E Enable or disable Pin Change Interrupt
0 disable
1 enable

INT2E Enable or disable External Interrupt 1
0 disable
1 enable

INT1E Enable or disable External Interrupt 0
0 disable
1 enable

IE1 (Interrupt Enable Register 1) : A9H

7	6	5	4	3	2	1	0
-	-	-	INT10E-	INT9E	-	-	-
R	R	R/-	R/W	R/W	R	R	R

Initial value : 00H

INT10E Enable or disable UART Tx Interrupt
0 disable
1 enable

INT9E Enable or disable UART Rx Interrupt
0 disable
1 enable

IE2 (Interrupt Enable Register 2) : AAH

7	6	5	4	3	2	1	0
-	-	INT17E	-	-	INT14E	INT13E	INT12E
R	R	R/W-	R	R	R/W	R/W	R/W

Initial value : 00H

INT17E Enable or disable Timer 4 interrupt
0 disable
1 enable
1 enable

INT14E Enable or disable Timer 1 Interrupt
0 disable
1 enable

INT13E Enable or disable Timer 0 Interrupt
0 disable
1 enable

INT12E Enable or disable I2C Interrupt
0 disable
1 enable

IE3 (Interrupt Enable Register 3) : ABH

7	6	5	4	3	2	1	0
-	-	INT23E	INT22E-	INT21E	INT20E	INT19E	INT18E
R	R	R/W-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

INT23E Enable or disable EEPROM Interrupt

0	disable
1	enable

INT22E Enable or disable BIT Interrupt

0	disable
1	enable

INT21E Enable or disable WDT Interrupt

0	disable
1	enable

INT20E Enable or disable WT Interrupt

0	disable
1	enable

INT19E Enable or disable Analog Comparator Interrupt

0	disable
1	enable

INT18E Enable or disable ADC Interrupt

0	disable
1	enable

IP (Interrupt Priority Register) : B8H

7	6	5	4	3	2	1	0
-	-	IP5	IP4	IP3	IP2	IP1	IP0
R	R	R/W-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP1 (Interrupt Priority Register 1) : F8H

7	6	5	4	3	2	1	0
-	-	IP15	IP14	IP13	IP12	IP11	IP10
R	R	R/W-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP[5:0], IP1[5:0] Select Interrupt Group Priority

IP1x	IPx	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

EIFLAG (External Interrupt Flag Register) : ACH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	FLAG1	FLAG0
R	R	R-	R	R	R	R/W	R/W

Initial value : 00H

FLAG[1:0] If External Interrupt is occurred, the flag becomes ‘1’. The flag can be cleared by writing a ‘0’ to bit

0 External Interrupt not occurred

1 External Interrupt occurred

EIEDGE (External Interrupt Edge Register) : ADH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	EDGE1	EDGE0
R/W-	R/W-	R/W-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

EDGE[1:0] Determines which type of edge or level sensitive interrupt may occ ur.

0 Level (default)

1 Edge

EIPOLA (External Interrupt Polarity Register) : AEH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	POLA1	POLA0
R/W	R/W	R/W-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

POLA[1:0] According to EIEDGE, External interrupt polarity register has a different means. If EIEDGE is level type, external interrupt polarity is able to have Low/High level value. If EIEGDE is edge type, external interrupt polarity is able to have rising/ falling edge value.

Level case:

0 When High level, Interrupt occurred (default)

1 When Low level, Interrupt occurred

Edge case:

0 When Rising edge, Interrupt occurred (default)

1 When Falling edge, Interrupt occurred

EIENAB (External Interrupt Enable Register) : AFH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	ENAB1	ENAB0
R/W-	R/W-	R/W-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

ENAB[1:0] Control External Interrupt

0 disable (default)

1 enable

4. Peripheral Hardware

4.1 Clock Generator

4.1.1 Overview

As shown in Figure 4.1, the clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and the peripheral hardware. It contains main-frequency clock oscillator. The system clock operation can be easily obtained by attaching a crystal between the XIN and XOUT pin, respectively. The system clock can also be obtained from the external oscillator. In this case, it is necessary to put the external clock signal into the XIN pin and open the XOUT pin. The default system clock is INT-RC Oscillator and the default division rate is one. In order to stabilize system internally, use 128 KHz ring-oscillator (±50%) for BIT and WDT.

- Calibrated Internal RC Oscillator (8 MHz / ±1%)
  . INT-RC OSC/1 (Default system clock)
  . INT-RC OSC/2 (4 MHz)
  . INT-RC OSC/4 (2 MHz)
  . INT-RC OSC/8 (1 MHz)
- Crystal Oscillator (1~8 MHz)
- Sub-Clock Crystal Oscillator (32.768 KHz)
- Internal Ring-Oscillator (128 KHz / ±50%)

4.1.2 Block Diagram

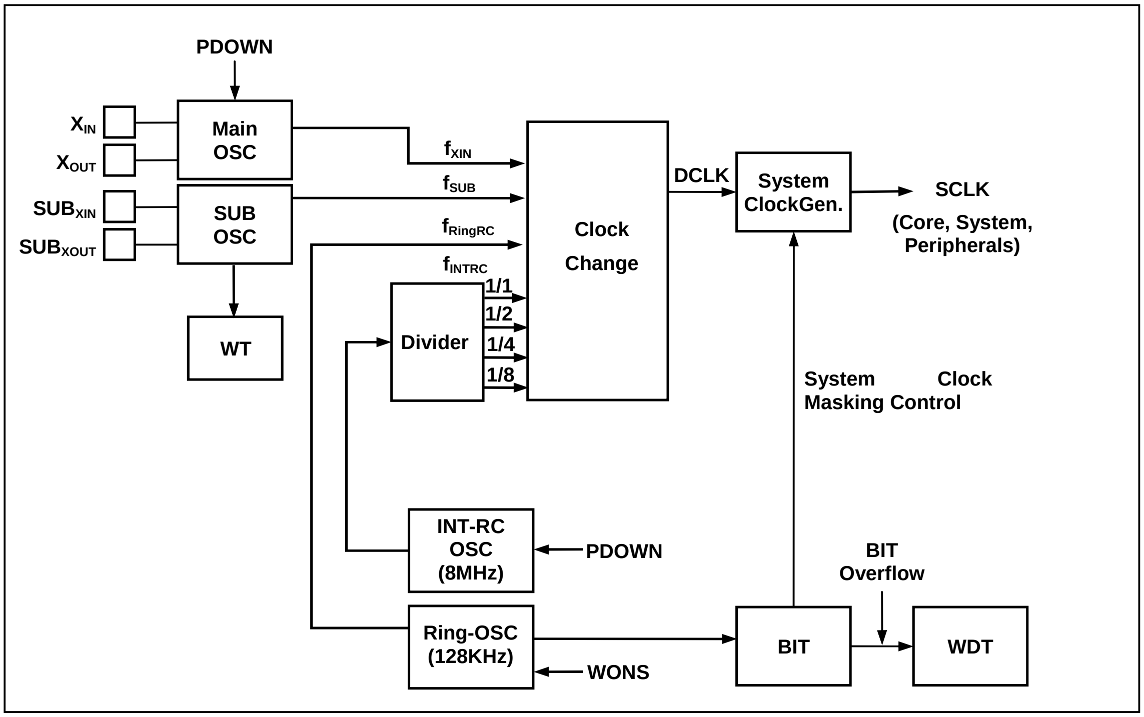


Figure 4.1 Clock Generator Block Diagram

4.1.3 Register Map

Table 4.1 SSCR Register Map

Name	Address	Dir	Default	Description
SCCR	8AH	R/W	04H	System and Clock Control Register

4.1.4 Clock Generator Register description

The Clock Generation Register uses clock control for system operation. The clock generation consists of System and Clock register.

4.1.5 Register description for Clock Generator

SCCR (System and Clock Control Register) : 8AH

7	6	5	4	3	2	1	0
WONS	DIV1	DIV0	CBYS	ISTOP	XSTOP	CS1	CS0
R/W	R/W	R/W-	R/W	R/W	R/W	R/W	R/W

Initial value : 04H

WONS Control the operation of WDT RC-Oscillation during stop mode

0	WDTRC-Oscillator is disabled at stop mode (=STOP2)
1	WDTRC-Oscillator is enabled at stop mode (=STOP1)

DIV[1:0] When using fINTRC as system clock, determine division rate. Note) when using fINTRC as system clock, only division rate come into effect.

Note) To change by software, CBYS set to ‘1’

DIV1	DIV0	description
0	0	fINTRC/1 (8MHz)
0	1	fINTRC/2 (4MHz)
1	0	fINTRC/4 (2MHz)
1	1	fINTRC/8 (1MHz)

CBYS Control the scheme of clock change. If this bit set to ‘0’, clock change is controlled by hardware. But if this set to ‘1’, clock change is controlled by software. Ex) when setting CS[1:0], if CBYS bit set to ‘0’, it is not changed right now, CPU goes to STOP mode and then when wake-up, it applies to clock change.

Note) when clear this bit, keep other bits in SCCR

0	Clock changed by hardware during stop mode (default)
1	Clock changed by software. After clock is changed, it should be cleared for low power.

ISTOP Control the operation of INT-RC Oscillation

Note) when CBYS=’1’, It is applied

0	RC-Oscillation enable (default)
1	RC-Oscillation disable

XSTOP Control the operation of X-Tal Oscillation

Note1) when CBYS=’1’, It is applied

Note2) if XINENA bit in FUSE_CONF to ‘0’, XSTOP is fixed to ‘1’

0	X-Tal Oscillation enable
1	X-Tal Oscillation disable (default)

CS[1:0] Determine System Clock

Note) by CBYS bit, reflection point is decided

CS1 CS0 description

0	0	fINTRC INTRC (8 MHz)
0	1	fXIN External Main Clock (1~8 MHz)
1	0	fSUB (32.768 KHz)
1	1	fRingRC (128KHz, ±50%)

4.2 BIT

4.2.1 Overview

The Z51F0410 MCU features one 8-bit Basic Interval Timer that is free-run and can't stop. Block diagram is shown in Figure 4.2. In addition, the Basic Interval Timer generates the time base for watchdog timer counting. It also provides a Basic interval timer interrupt (BITF).

The Z51F0410 MCU has these Basic Interval Timer (BIT) features:

- During Power On, BIT gives a stable clock generation time
- On exiting Stop mode, BIT gives a stable clock generation time
- As clock function, time interrupt occurrence

4.2.2 Block Diagram

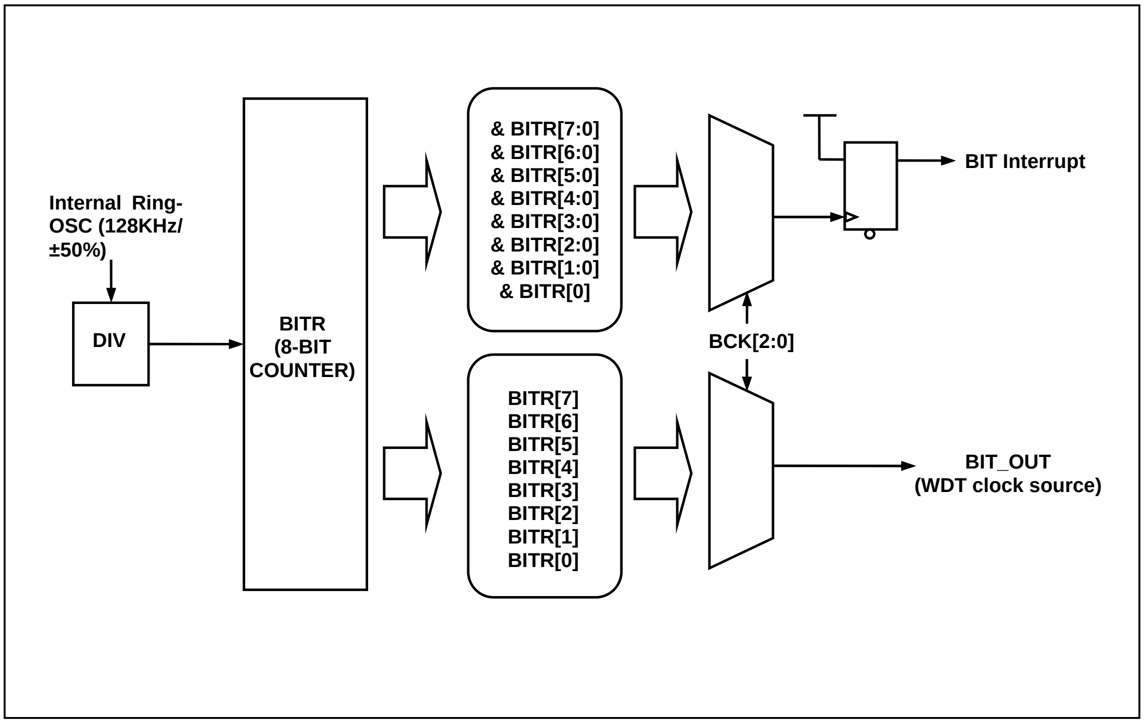


Figure 4.2 BIT Block Diagram

4.2.3 Register Map

Table 4.2 BIT Register Map

Name	Address	Dir	Default	Description
BCCR	8BH	R/W	05H	BIT Clock Control Register
BITR	8CH	R	00H	Basic Interval Timer Register

4.2.4 Bit Interval Timer Register description

The Bit Interval Timer Register consists of BIT Clock control register (BCCR) and Basic Interval Timer register (BITR). If BCLR bit set to '1', BITR becomes '0' and then counts up. After 1 machine cycle, BCLR bit is cleared as '0' automatically.

4.2.5 Register description for Bit Interval Timer

BCCR (BIT Clock Control Register) : 8BH

7	6	5	4	3	2	1	0
BITF	-	-	-	BCLR	BCK2	BCK1	BCK0
R/W	R	R	R	R/W	R/W	R/W	R/W

Initial value : 05H

BITF When BIT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit.

0 no generation

1 generation

BCLR If BCLK Bit is written to '1', BIT Counter is cleared as '0'

0 Free Running

1 Clear Counter

BCK[2:0] Select BIT overflow period (BIT Clock=4 KHz)

BCK2	BCK1	BCK0	
0	0	0	0.5msec (BIT Clock * 2)
0	0	1	1msec
0	1	0	2msec
0	1	1	4msec
1	0	0	8msec
1	0	1	16msec (default)
1	1	0	32msec
1	1	1	64msec

BITR (Basic Interval Timer Register) : 8CH

7	6	5	4	3	2	1	0
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
R	R	R	R	R	R	R	R

Initial value : 00H

BIT[7:0] BIT Counter

4.3 WDT

4.3.1 Overview

The watchdog timer rapidly detects the CPU malfunction such as endless looping caused by noise or the like, and resumes the CPU to the normal state. The watchdog timer signal for detecting malfunction can be selected either a reset CPU or an interrupt request. When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals. It is possible to use free running 8-bit timer mode (WDTRSON=’0’) or watch dog timer mode (WDTRSON=’1’) as setting WDTMR[6] bit. If writing WDTMR[5] to ‘1’, WDT counter value is cleared and counts up. After 1 machine cycle, this bit has ‘0’ automatically. The watchdog timer consists of 8-bit binary counter and the watchdog timer data register. When the value of 8-bit binary counter is equal to the 8 bits of WDTR, the interrupt request flag is generated. This can be used as Watchdog timer interrupt or reset the CPU in accordance with the bit WDTRSON.

The clock source of Watch Dog Timer is BIT overflow output. The interval of watchdog timer interrupt is decided by BIT overflow period and WDTR set value. The equation is as below

WDT Interrupt Interval = (BIT Interrupt Interval) X (WDTR Value+1)

4.3.2 Block Diagram

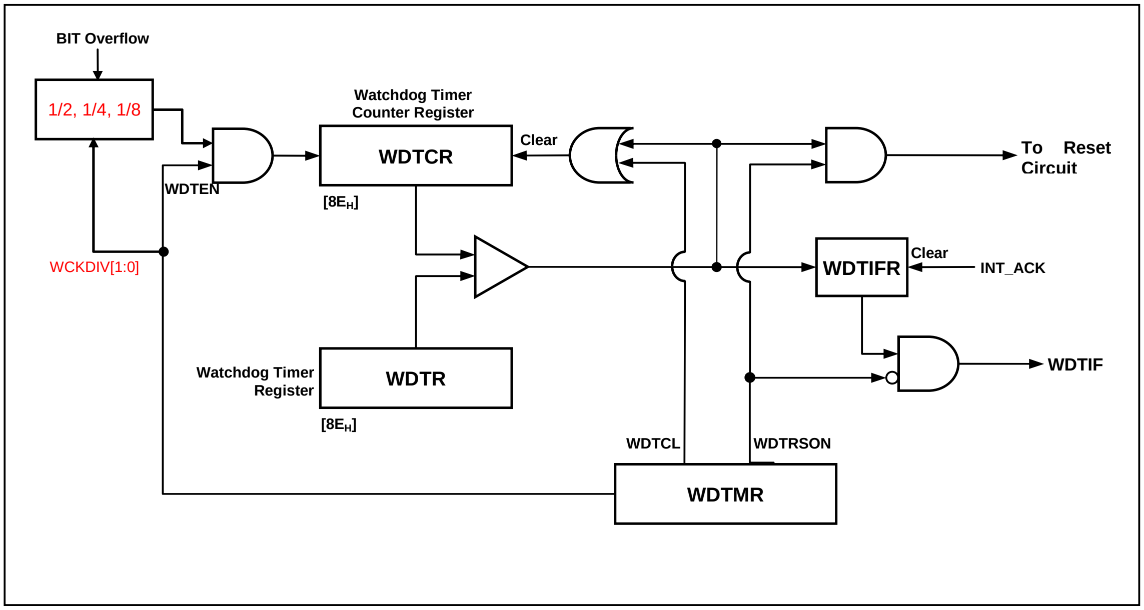


Figure 4.3 WDT Block Diagram

4.3.3 Register Map

Table 4.3 WDT Register Map

Name	Address	Dir	Default	Description
WDTR	8EH	W	FFH	Watch Dog Timer Register
WDTCR	8EH	R	00H	Watch Dog Timer Counter Register
WDTMR	8DH	R/W	00H	Watch Dog Timer Mode Register

4.3.4 Watch Dog Timer Register description

The Watch dog timer (WDT) Register consists of Watch Dog Timer Register (WDTR), Watch Dog Timer Counter Register (WDTCR) and Watch Dog Timer Mode Register (WDTMR).

4.3.5 Register description for Watch Dog Timer

WDTR (Watch Dog Timer Register:Write Case) : 8EH

7	6	5	4	3	2	1	0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0
W	W	W	W	W	W	W	W

Initial value : FFH

WDTR[7:0] Set a period
WDT Interrupt Interval=(BIT Interrupt Interval) x(WDTR Value+1)

Note) To guarantee proper operation, the data should be greater than 01H.

WDTCR (Watch Dog Timer Counter Register:Read Case) : 8EH

7	6	5	4	3	2	1	0
WDTCR7	WDTCR6	WDTCR5	WDTCR4	WDTCR3	WDTCR2	WDTCR1	WDTCR0
R	R	R	R	R	R	R	R

Initial value : 00H

WDTCR[7:0] WDT Counter

WDTMR (Watch Dog Timer Mode Register) : 8DH

7	6	5	4	3	2	1	0
WDTEN	WDTRSON	WDTCL	WCKDIV1	WCKDIV0	-	-	WDTIFR
R/W	R/W	R/W	R/W	R/W	-	-	R/W

Initial value : 00H

WDTEN Control WDT operation
0 disable
1 enable

WDTRSON Control WDT Reset operation
0 Free Running 8-bit timer
1 Watch Dog Timer Reset ON

WDTCL Clear WDT Counter
0 Free Run
1 Clear WDT Counter (auto clear after 1 Cycle)

WCKDIV[1:0] WDT Clock Division Selection
00 Default No Divided
01 WDT Clock = BIT Clock / 2
10 WDT Clock = BIT Clock / 4
11 WDT Clock = BIT Clock / 8

WDTIFR When WDT Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit or auto clear by INT_ACK signal.
0 WDT Interrupt no generation
1 WDT Interrupt generation

4.3.6 WDT Interrupt Timing Waveform

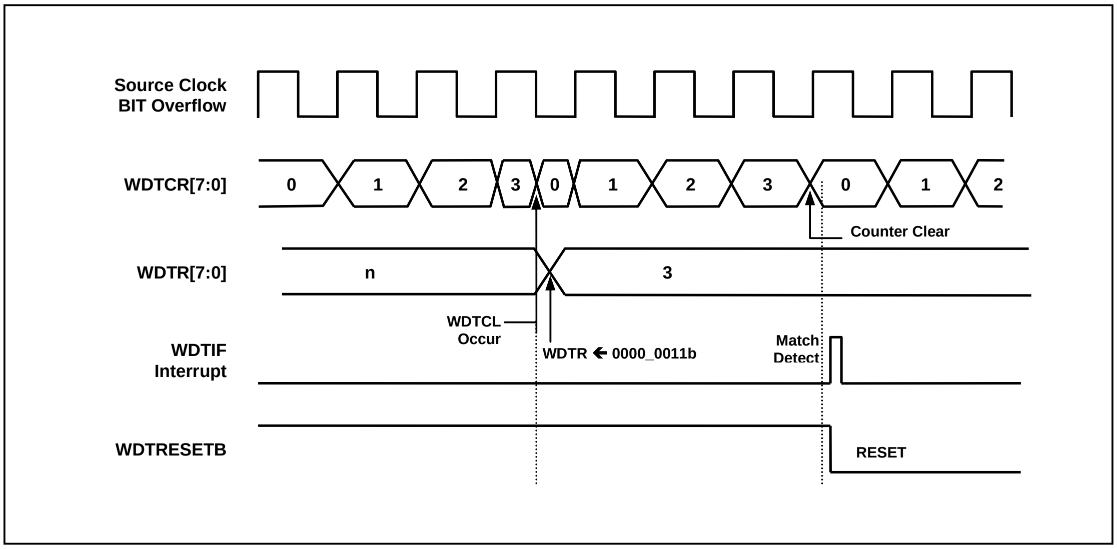


Figure 4.4 WDT Interrupt Timing Waveform

4.4 WT

4.4.1 Overview

The watch timer has the function for RTC (Real Time Clock) operation. It is generally used for RTC design. The internal structure of the watch timer consists of the clock source select circuit, timer counter circuit, output select circuit and watch timer mode register. To operate the watch timer, determine the input clock source, output interval and set WTEN to '1' in watch timer mode register (WTMR). It is able to execute simultaneously or individually. To stop or reset WT, clear the WTEN bit in WTMR register. Even if CPU is STOP mode, sub clock is able to be alive so WT can continue the operation. The watch timer counter circuits may be composed of 21-bit counter which is low 14-bit with binary counter and high 7-bit with auto reload counter in order to raise resolution. In WTR, it can control WT clear and set Interval value at write time, and it can read 7-bit WT counter value at read time.

4.4.2 Block Diagram

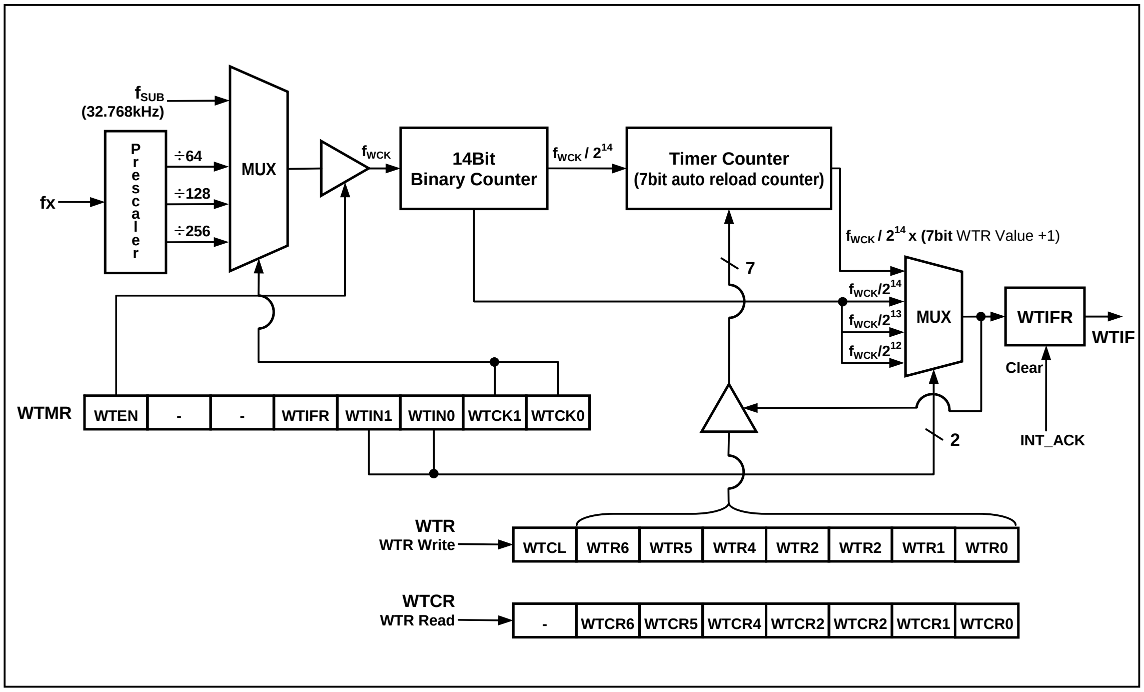


Figure 4.5 Watch Timer Block Diagram

4.4.3 Register Map

Table 4.4 WT Register Map

Name	Address	Dir	Default	Description
WTMR	9DH	R/W	00H	Watch Timer Mode Register
WTR	9EH	W	7FH	Watch Timer Register
WTCR	9EH	R	00H	Watch Timer Counter Register

4.4.4 Watch Timer Register description

The watch timer register (WT) consists of Watch Timer Mode Register (WTMR), Watch Timer Counter Register (WTCR) and Watch Timer Register (WTR). As WTMR is 6-bit writable/readable register, WTMR can control the clock source (WTCK), interrupt interval (WTIN) and function enable/disable (WTEN). Also there is WT interrupt flag bit (WTIFR).

4.4.5 Register description for Watch Timer

WTMR (Watch Timer Mode Register) : 9DH

7	6	5	4	3	2	1	0
WTEN	-	-	WTIFR	WTIN1	WTIN0	WTCK1	WTCK0
R/W	-	-	R/W	R/W	R/W	R/W	R/W

Initial value :0 0H

WTEN Control Watch Timer

0	disable
1	enable

WTIFR When WT Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit or auto clear by INT_ACK signal.

0	WT Interrupt no generation
1	WT Interrupt generation

WTIN[1:0] Determine interrupt interval

WTIN1	WTIN0	description
0	0	fwck/2^11
0	1	fwck/2^13
1	0	fwck/2^14
1	1	fwck/2^14 x (7bit WT Value)

WTCK[1:0] Determine Source Clock

WTCK1	WTCK0	description
0	0	fsub
0	1	fx/256
1	0	fx/128
1	1	fx/64

Remark: fx– Main system clock oscillation frequency
fsub- Sub clock oscillation frequency
fwck- selected Watch Timer clock

WTR (Watch Timer Register:Write Case) : 9EH

7	6	5	4	3	2	1	0
WTCL	WTR6	WTR5	WTR4	WTR3	WTR2	WTR1	WTR0
W	W	W	W	W	W	W	W

Initial value : 7FH

WTCL Clear WT Counter

0	Free Run
1	Clear WT Counter (auto clear after 1 Cycle)

WTR[6:0] Set WT period
WT Interrupt Interval=(fwck/2^14) x(7bit WT Value+1)

Note) To guarantee proper operation, it is greater than 01H to write WTR.

WTCR (Watch Timer Counter Register:Read Case) : 9EH

7	6	5	4	3	2	1	0
	WTCR6	WTCR5	WTCR4	WTCR3	WTCR2	WTCR1	WTCR0

-	R	R	R	R	R	R	R

Initial value : 00H

WTCR[6:0] WT Counter

4.5 Timer/PWM

4.5.1 8-bit Timer/Event Counter 0, 1

4.5.1.1 Overview

Timer 0 and timer 1 can be used either two 8-bit timer/counter or one 16-bit timer/counter with combine them. Each 8-bit timer/event counter module has multiplexer, 8-bit timer data register, 8-bit counter register, mode register, input capture register, comparator. For PWM, it has PWM register (T1PPR, T1PDR, T1PWHR).

It has seven operating modes:

- 8 Bit Timer/Counter Mode
- 8 Bit Capture Mode
- 8 Bit Compare Output Mode
- 16 Bit Timer/Counter Mode
- 16 Bit Capture Mode
- 16 Bit Compare Output Mode
- PWM Mode

The timer/counter can be clocked by an internal or external clock source (external EC0). The clock source is selected by clock select logic which is controlled by the clock select (T0CK[2:0], T1CK[1:0]).

- TIMER0 clock source : fX/2, 4, 16, 64, 256, 1024, 4096, EC0
- TIMER1 clock source : fX/1, 2, 16, T0CK

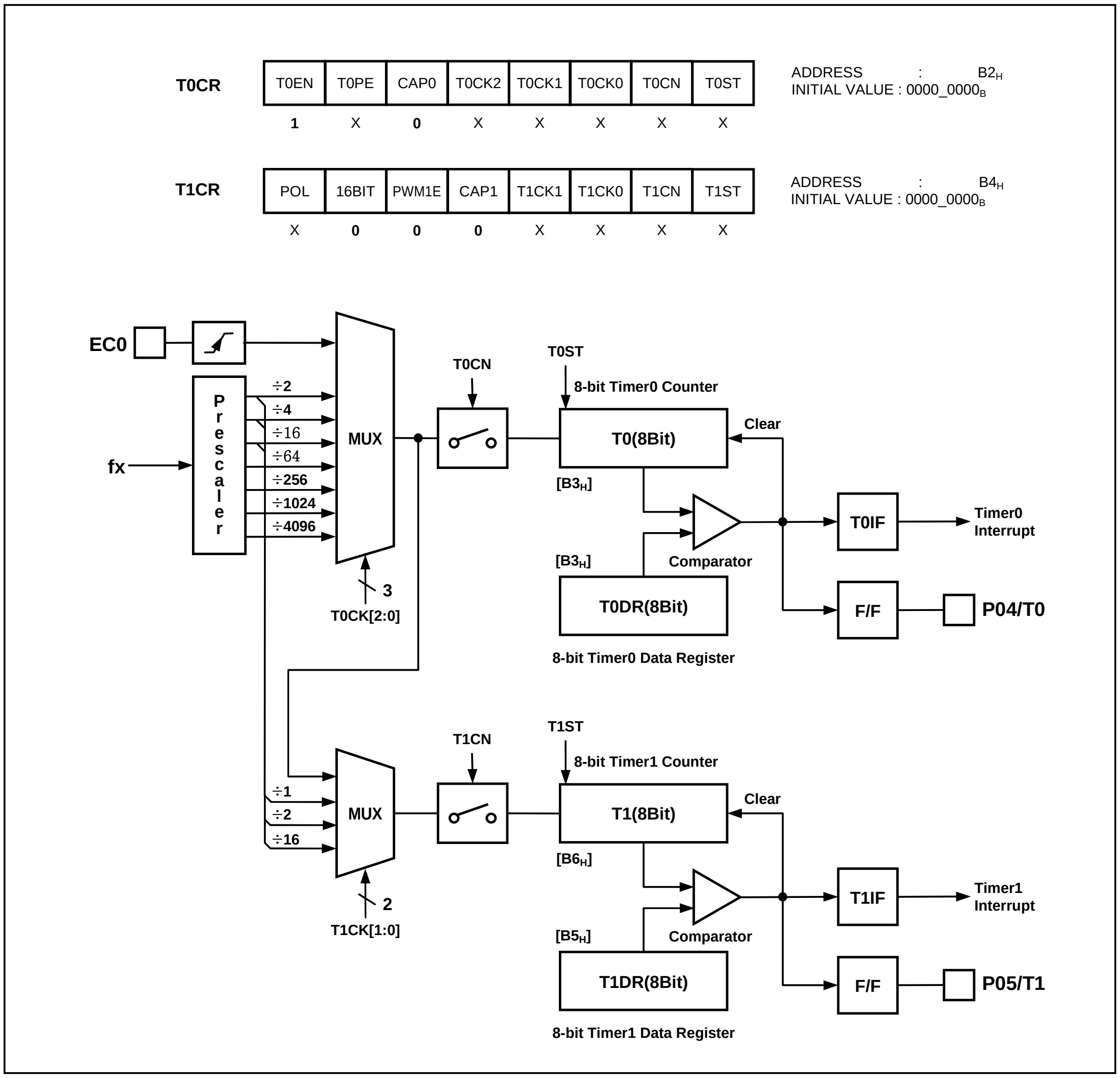
In the capture mode, by INT0, INT1, the data is captured into Input Capture Register. The Timer 0 outputs the compare result to T0 port in 8/16-bit mode. Also the timer 1 outputs the result T1 port in the timer mode and the PWM wave form to PWM1 in the PWM mode.

Table 4.5 Operating Modes of Timer

16 Bit	CAP0	CAP1	PWM1E	T0CK[2:0]	T1CK[1:0]	T0/1_PE	Timer 0	Timer 1
0	0	0	0	XXX	XX	00	8 Bit Timer	8 Bit Timer
0	0	1	0	111	XX	00	8 Bit Event Counter	8 Bit Capture
0	1	0	0	XXX	XX	01	8 Bit Capture	8 Bit Compare Output
0	0	0	1	XXX	XX	11	8 Bit Timer/Counter	10 Bit PWM
1	0	0	0	XXX	11	00	16 Bit Timer	
1	0	0	0	111	11	00	16 Bit Event Counter	
1	1	1	0	XXX	11	00	16 Bit Capture	
1	0	0	0	XXX	11	01	16 Bit Compare Output	

4.5.1.2 8-Bit Timer/Counter Mode

The 8-bit Timer/Counter Mode is selected by control registers as shown in Figure 4.6.



T0CR	T0EN	T0PE	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
	1	X	0	X	X	X	X	X

ADDRESS : B2$_H$
INITIAL VALUE : 0000_0000$_B$

T1CR	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
	X	0	0	0	X	X	X	X

ADDRESS : B4$_H$
INITIAL VALUE : 0000_0000$_B$

Figure 4.6 8 Bit Timer/Event Counter0, 1 Block Diagram

The two 8-bit timers have each counter and data register. The counter register is increased by internal or external clock input. The timer 0 can use the input clock with 2, 4, 16, 64, 256, 1024, 4096 prescaler division rates (T0CK[2:0]). The timer 1 can use the input clock with 1, 2, 16 and timer 0 overflow clock (T1CK[1:0]). When the value of T0, 1 value and the value of T0DR, T1DR are respectively identical in Timer 0, 1, the interrupt of timer 0, 1 occurs. The external clock (EC0) counts up the timer at the rising edge. If EC0 is selected from T0CK[2:0], EC0 port becomes input port. The timer 1 can't use the external EC0 clock.

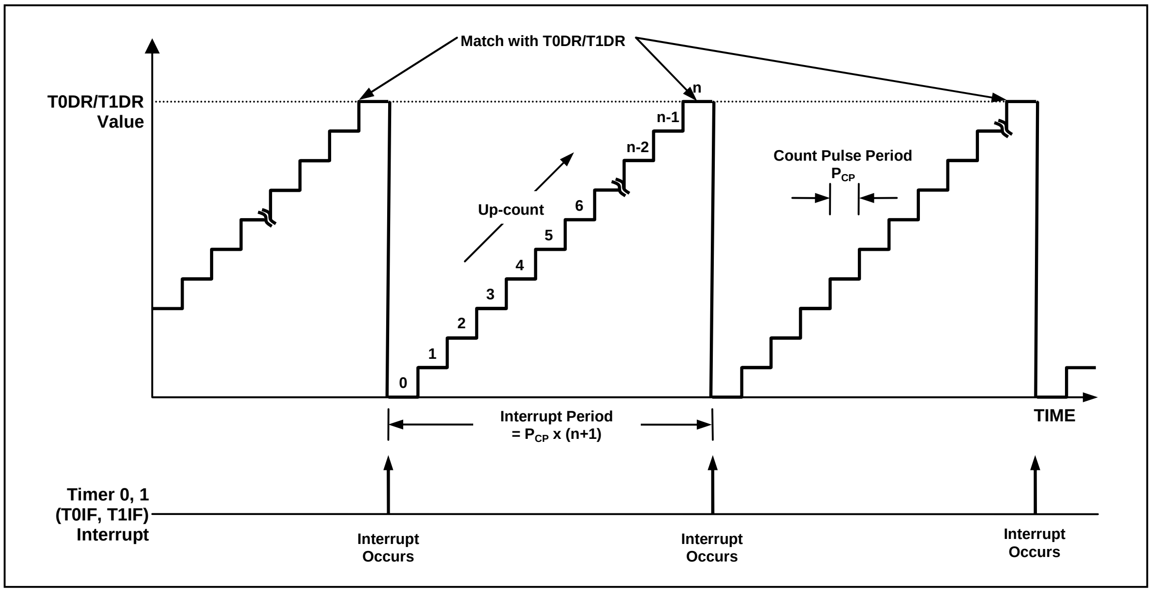


Figure 4.7 Timer/Event Counter0, 1 Example

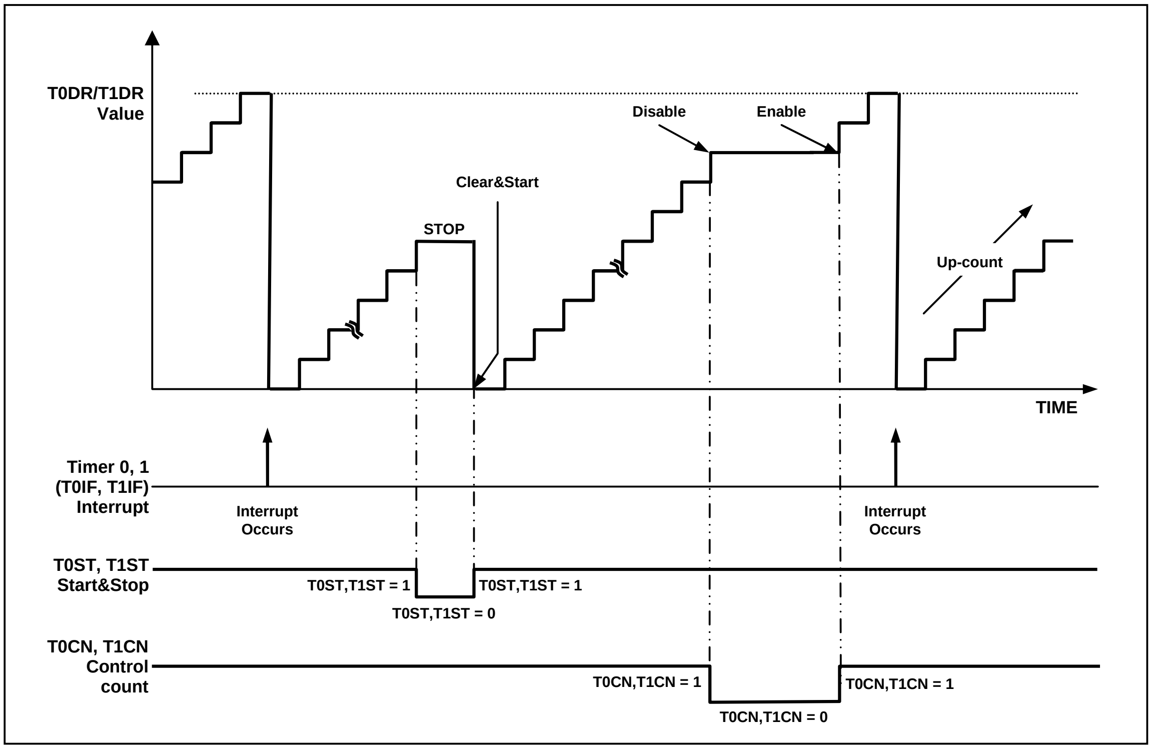


Figure 4.8 Operation Example of Timer/Event Counter0, 1

4.5.1.3 16-Bit Timer/Counter Mode

The timer register is being run with all 16bits. A 16-bit timer/counter register T0, T1 are incremented from 0000H to FFFFH until it matches T0DR, T1DR and then resets to 0000H. The match output generates the Timer 0 interrupt (no timer 1 interrupt). The clock source is selected from T0CK[2:0] and T1CK[1:0] must set 11b and 16BIT bit must set to ‘1’. The timer 0 is LSB 8-bit, the timer 1 is MSB 8-bit. The 16-bit mode setting is shown as Figure 4.9.

T0CR	T0EN	T0PE	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
	1	X	0	X	X	X	X	X

ADDRESS : $B2_H$
INITIAL VALUE : 0000_0000_B

T1CR	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
	X	1	0	0	1	1	X	X

ADDRESS : $B4_H$
INITIAL VALUE : 0000_0000_B

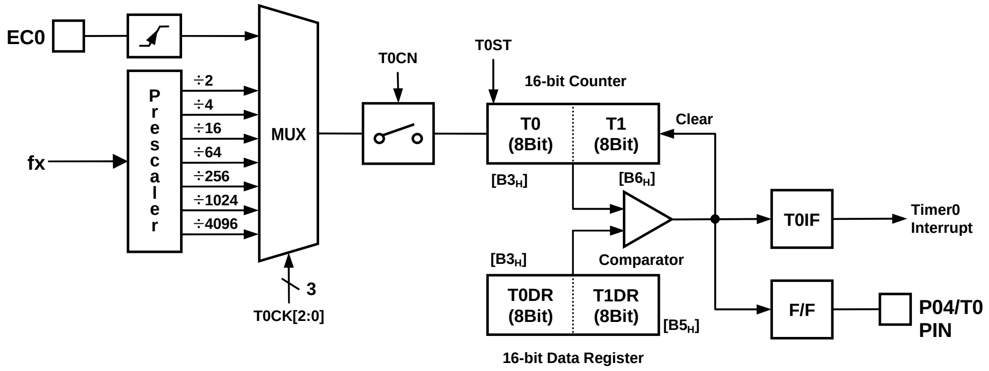


Figure 4.9 16 Bit Timer/Event Counter0, 1 Block Diagram

Note: Do not set T0DR to 0x00 in 16-bit mode. If T0DR is set to 0x00, Timer interrupt or count match occur after T1DR+0x01. If you set T0DR to be 0x00, T1DR must have one fewer number of count than the number of count which you want.

Example: If T1DR=0x01 and T0DR=0x00, counter match occurs when T1=0x02 and T0=0x00.

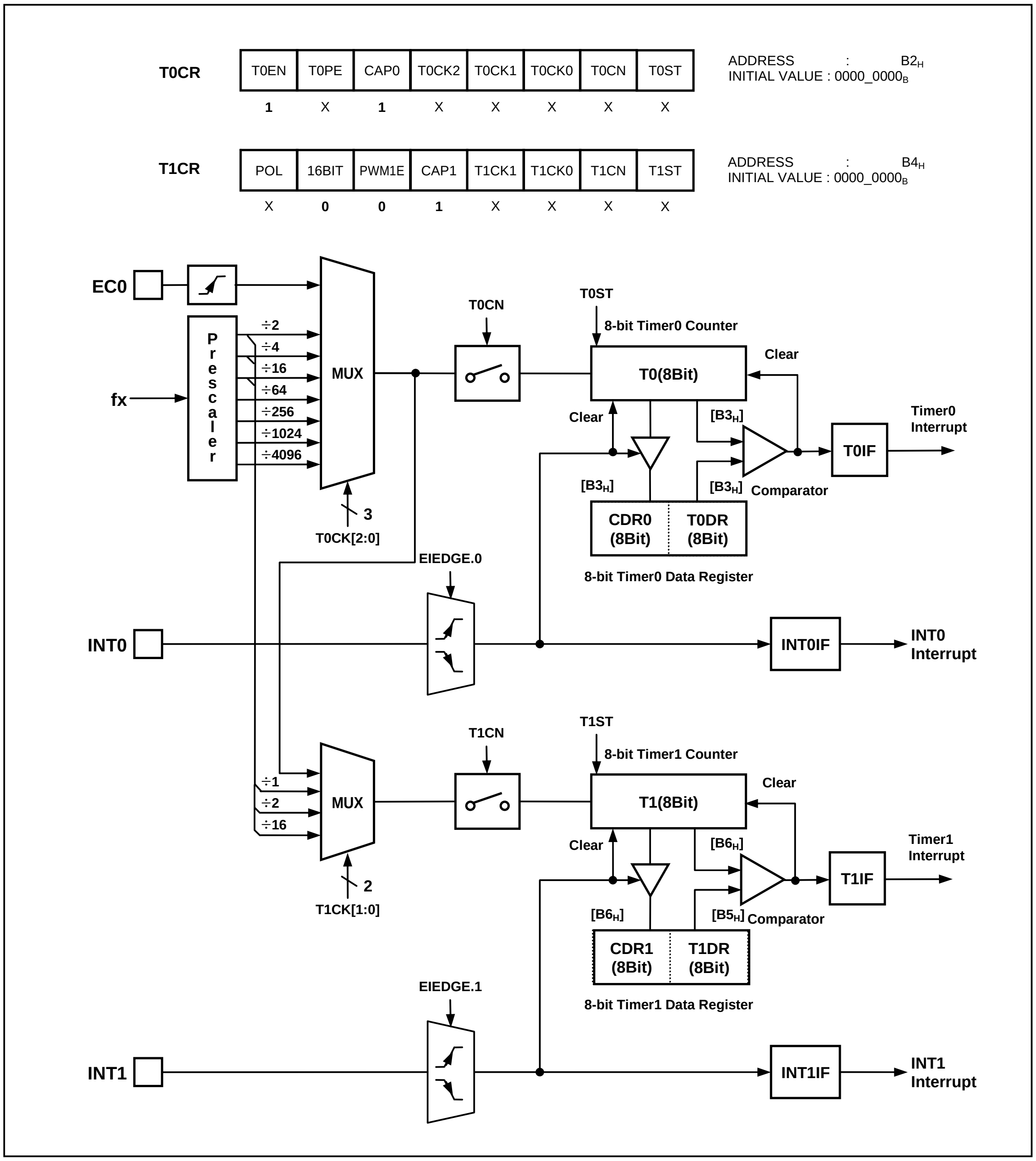
4.5.1.4 8-Bit Capture Mode

The timer 0, 1 capture mode is set by CAP0, CAP1 as ‘1’. The clock source can use the internal/external clock. Basically, it has the same function of the 8-bit timer/counter mode and the interrupt occurs at T0, T1 and T0DR, T1DR matching time, respectively. The capture result is loaded into CDR0, CDR1. The T0, T1 value is automatically cleared by hardware and restarts counter.

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer.

As the EIEDGE and EIPOLA register setting, the external interrupt INT0, INT1 function is chosen.

The CDR0, T0 and T0DR are in same address. In the capture mode, reading operation is read the CDR0, not T0DR because path is opened to the CDR0. The CDR1 has the same function.



T0CR

T0EN	T0PE	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
1	X	1	X	X	X	X	X

ADDRESS : $B2_H$
INITIAL VALUE : 0000_0000_B

T1CR

POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
X	0	0	1	X	X	X	X

ADDRESS : $B4_H$
INITIAL VALUE : 0000_0000_B

Figure 4.10 8-bit Capture Mode for Timer0, 1

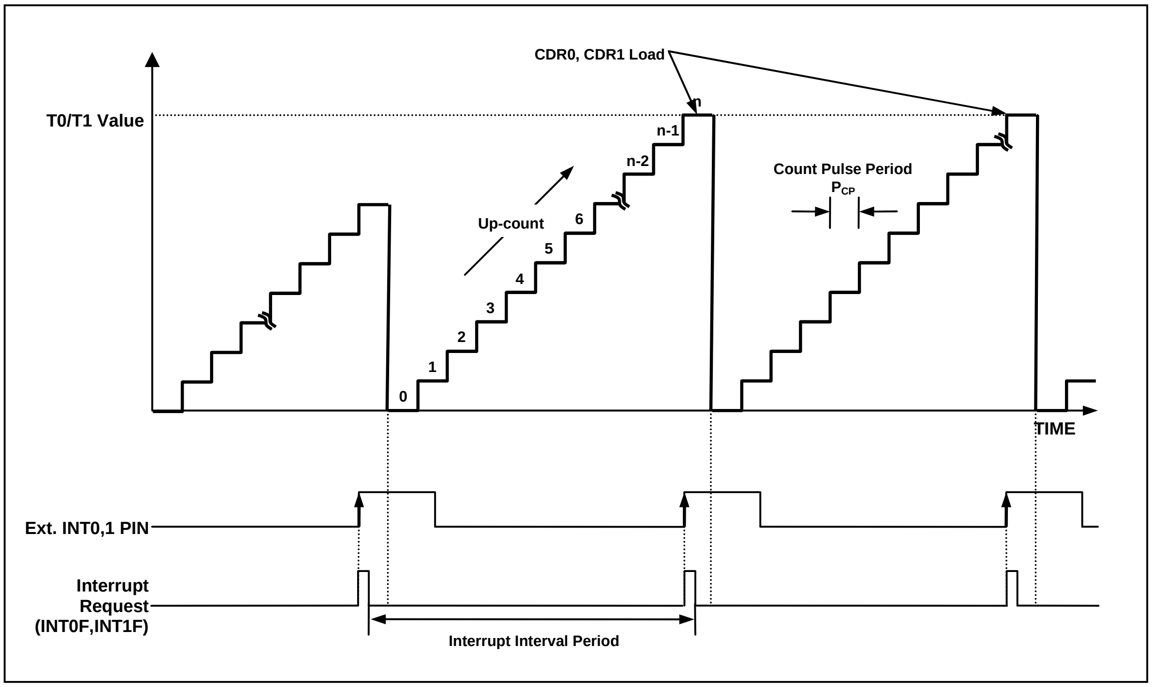


Figure 4.11 Input Capture Mode Operation of Timer 0, 1

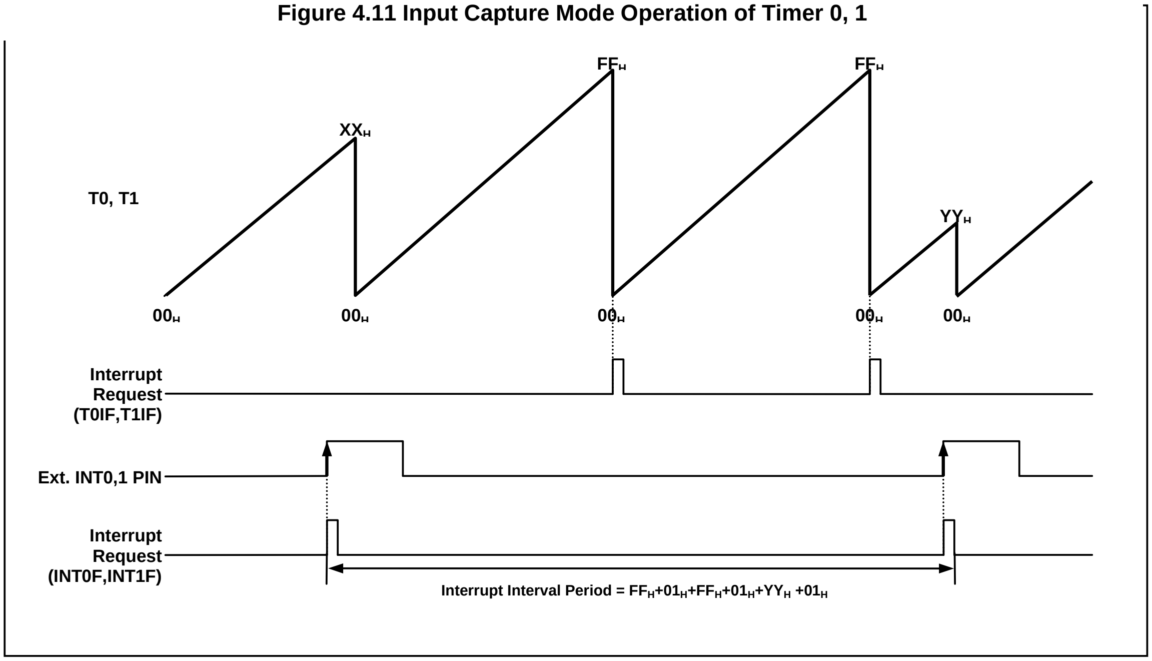

FF_H
FF_H
XX_H
T0, T1
YY_H
00_H
00_H
00_H
00_H
00_H
Interrupt Request (T0IF,T1IF)
Ext. INT0,1 PIN
Interrupt Request (INT0F,INT1F)
Interrupt Interval Period = FF_H+01_H+FF_H+01_H+YY_H +01_H

Figure 4.12 Express Timer Overflow in Capture Mode

4.5.1.5 16-Bit Capture Mode

The 16-bit capture mode is the same operation as 8-bit capture mode, except that the timer register uses 16 bits.

The clock source is selected from T0CK[2:0] and T1CK[1:0] must set 11b and 16BIT0 bit must set to ‘1’. The 16-bit mode setting is shown as Figure 4.13.

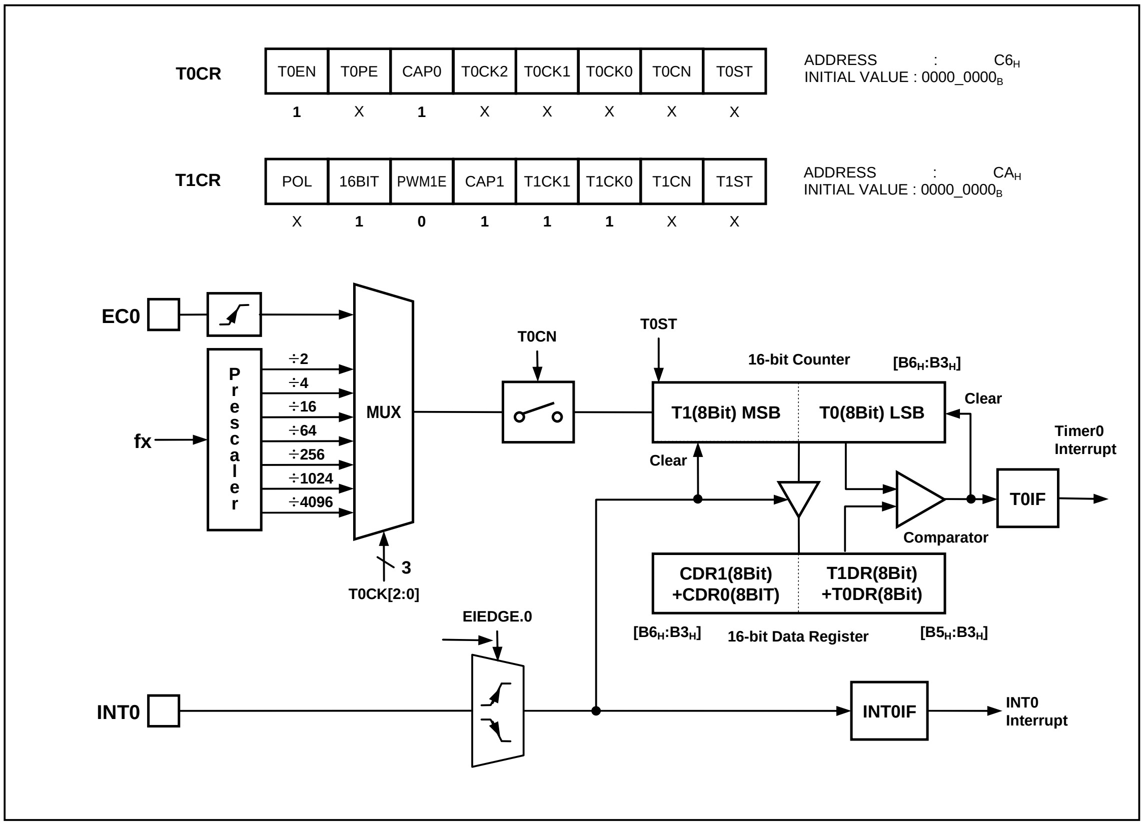


Figure 4.13 16-bit Capture Mode of Timer 0, 1

4.5.1.6 PWM Mode

The timer 1 has a PWM (Pulse Width Modulation) function. In PWM mode, the T1/PWM1 output pin outputs up to 10-bit resolution PWM output. This pin should be configured as a PWM output by set T1_PE to ‘1’. The period of the PWM output is determined by the T1PPR (PWM period register) + T1PWHR[3:2] + T1PWHR[1:0]

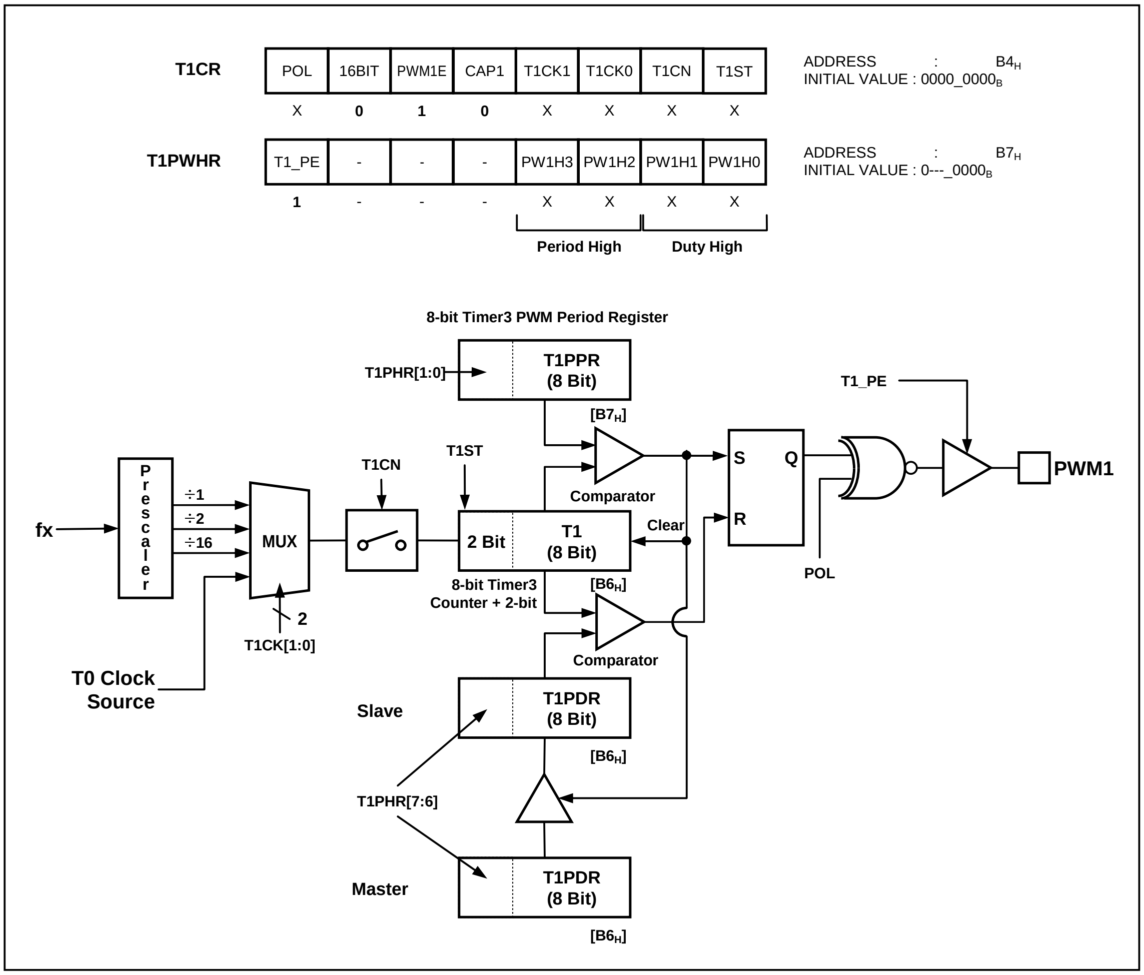
PWM Period = [T1PWHR[3:2]T1PPR] X Source Clock

PWM Duty = [T1PWHR[1:0] T1PDR] X Source Clock

Table 4.6 PWM Frequency vs. Resolution at 8 Mhz

Resolution	Frequency		
	T1CK[1:0]=00 (125ns)	T1CK[1:0]=01 (250ns)	T1CK[1:0]=10 (2us)
10 Bit	7.8KHz	3.9KHz	0.49KHz
9 Bit	15.6KHz	7.8KHz	0.98KHz
8 Bit	31.2KHz	15.6KHz	1.95KHz
7 Bit	62.4KHz	31.2KHz	3.91KHz

The POL bit of T1CR register decides the polarity of duty cycle. If the duty value is set same to the period value, the PWM output is determined by the bit POL (1: High, 0: Low). And if the duty value is set to "00H", the PWM output is determined by the bit POL (1: Low, 0: High).



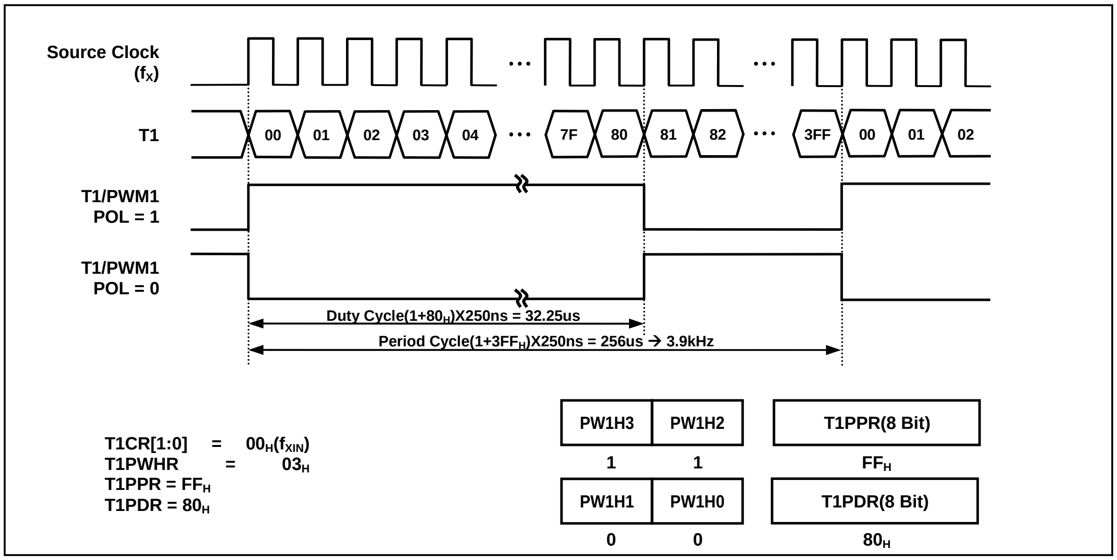
T1CR	POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
	X	0	1	0	X	X	X	X

ADDRESS : $B4_H$
INITIAL VALUE : 0000_0000_B

T1PWHR	T1_PE	-	-	-	PW1H3	PW1H2	PW1H1	PW1H0
	1	-	-	-	X	X	X	X
					Period High		Duty High	

ADDRESS : $B7_H$
INITIAL VALUE : 0---_0000_B

Figure 4.14 PWM Mode



Source Clock
(fX)
T1
00
01
02
03
04
7F
80
81
82
3FF
00
01
02
T1/PWM1
POL = 1
T1/PWM1
POL = 0
Duty Cycle(1+80H)X250ns = 32.25us
Period Cycle(1+3FFH)X250ns = 256us → 3.9kHz

T1CR[1:0] = $00_H(f_{XIN})$
T1PWHR = 03_H
T1PPR = FF_H
T1PDR = 80_H

PW1H3	PW1H2
1	1

T1PPR(8 Bit)
FF_H

PW1H1	PW1H0
0	0

T1PDR(8 Bit)
80_H

Figure 4.15 Example of PWM at 4MHz

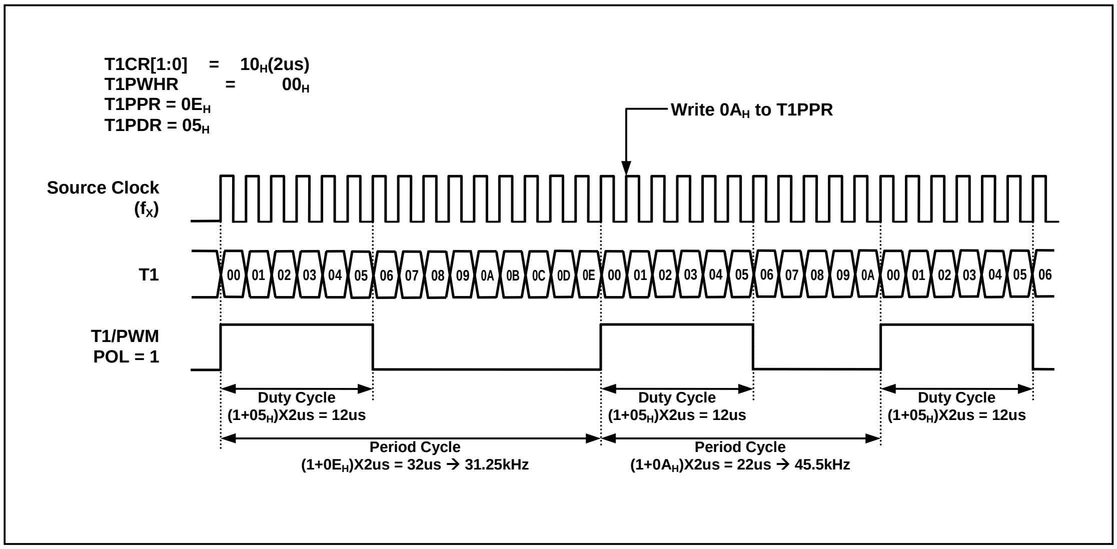

T1CR[1:0] = 10H(2us)
T1PWHR = 00H
T1PPR = 0EH
T1PDR = 05H
Write 0AH to T1PPR
Source Clock (fX)
T1
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 00 01 02 03 04 05 06 07 08 09 0A 00 01 02 03 04 05 06
T1/PWM POL = 1
Duty Cycle (1+05H)X2us = 12us
Duty Cycle (1+05H)X2us = 12us
Duty Cycle (1+05H)X2us = 12us
Period Cycle (1+0EH)X2us = 32us → 31.25kHz
Period Cycle (1+0AH)X2us = 22us → 45.5kHz

Figure 4.16 Example of Changing the Period in Absolute Duty Cycle at 4Mhz

4.5.1.7 8-Bit (16-Bit) Compare Output Mode

If the T1 (T0+T1) value and the T1DR (T0DR+T1DR) value are matched, T1/PWM1 port outputs. The output is 50:50 of duty square wave, the frequency is following

$$f_{COMP} = \frac{\text{Oscillator Frequency}}{2 \times \text{Prescaler Value} \times (TDR + 1)}$$

To export the compare output as T1/PWM1, the T1_PE bit in the T1PWHR register must set to ‘1’.

4.5.1.8 Register Map

Table 4.7 Timer Register Map

Name	Address	Dir	Default	Description
T0CR	B2H	R/W	00H	Timer 0 Mode Control Register
T0	B3H	R	00H	Timer 0 Register
T0DR	B3H	W	FFH	Timer 0 Data Register
CDR0	B3H	R	00H	Capture 0 Data Register
T1CR	B4H	R/W	00H	Timer 1 Mode Control Register
T1DR	B5H	W	FFH	Timer 1 Data Register
T1PPR	B5H	W	FFH	Timer 1 PWM Period Register
T1	B6H	R	00H	Timer 1 Register
T1PDR	B6H	R/W	00H	Timer 1 PWM Duty Register
CDR1	B6H	R	00H	Capture 1 Data Register
T1PWHR	B7H	W	00H	Timer 1 PWM High Register

4.5.1.9 Timer/Counter 0, 1 Register description

The Timer/Counter 0, 1 Register consists of Timer 0 Mode Control Register (T0CR), Timer 0 Register (T0), Timer 0 Data Register (T0DR), Capture 0 Data Register (CDR0), Timer 1 Mode Control Register (T1CR), Timer 1 Data Register (T1DR), Timer 1 PWM Period Register (T1PPR), Timer 1 Register (T1), Timer 1 PWM Duty Register (T1PPR), Capture 1 Data Register (CDR1) and Timer 1 PWM High Register (T1PWHR).

4.5.1.10 Register description for Timer/Counter 0, 1

T0CR (Timer 0 Mode Control Register) : B2H

7	6	5	4	3	2	1	0
T0EN	T0_PE	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T0EN Control Timer 0

0 Timer 0 disable

1 Timer 0 enable

T0_PE Control Timer 0 Output port

0 Timer 0 Output disable

1 Timer 0 Output enable

CAP0 Control Timer 0 operation mode

0 Timer/Counter mode

1 Capture mode

T0CK[2:0] Select Timer 0 clock source. Fx is main system clock frequency

T0CK2	T0CK1	T0CK0	Description
0	0	0	fx/2
0	0	1	fx/2^2
0	1	0	fx/2^4
0	1	1	fx/2^6
1	0	0	fx/2^8
1	0	1	fx/2^10
1	1	0	fx/2^12
1	1	1	External Clock (EC0)

T0CN Control Timer 0 Count pause/continue

0 Temporary count stop

1 Continue count

T0ST Control Timer 0 start/stop

0 Counter stop

1 Clear counter and start

T0 (Timer 0 Register: Read Case) : B3H

7	6	5	4	3	2	1	0
T07	T06	T05	T04	T03	T02	T01	T00
R	R	R	R	R	R	R	R

Initial value : 00H

T0[7:0] T0 Counter data

T0DR (Timer 0 Data Register: Write Case) : B3H

7	6	5	4	3	2	1	0
T0D7	T0D6	T0D5	T0D4	T0D3	T0D2	T0D1	T0D0
W	W	W	W	W	W	W	W

Initial value : FFH

T0D[7:0] T0 Compare data

CDR0 (Capture 0 Data Register: Read Case) : B3H

7	6	5	4	3	2	1	0
CDR07	CDR06	CDR05	CDR04	CDR03	CDR02	CDR01	CDR00
R	R	R	R	R	R	R	R

Initial value : 00H

CDR0[7:0] T0 Capture data

T1CR (Timer 1 Mode Count Register) : B4H

7	6	5	4	3	2	1	0
POL	16BIT	PWM1E	CAP1	T1CK1	T1CK0	T1CN	T1ST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

POL Configure PWM polarity

0 Negative (Duty Match: Clear)

1 Positive (Duty Match: Set)

16BIT Select Timer 1 8/16Bit

0 8 Bit

1 16 Bit

PWM1E Control PWM enable

0 PWM disable

1 PWM enable

CAP1 Control Timer 1 mode

0 Timer/Counter mode

1 Capture mode

T1CK[1:0] Select clock source of Timer 1. Fx is the frequency of main system.

T1CK1	T1CK0	description
0	0	fx
0	1	fx/2
1	0	fx/2^4

1 1 Use Timer 0 Clock

T1CN Control Timer 1 Count pause/continue

0 Temporary count stop

1 Continue count

T1ST Control Timer 1 start/stop

0 Counter stop

1 Clear counter and start

T1DR (Timer 1 Data Register: Write Case) : B5H

7	6	5	4	3	2	1	0
T1D7	T1D6	T1D5	T1D4	T1D3	T1D2	T1D1	T1D0
W	W	W	W	W	W	W	W

Initial value : FFH

T1D[7:0] T1 Compare data

T1PPR (Timer 1 PWM Period Register: Write Case) : B5H

7	6	5	4	3	2	1	0
T1PP7	T1PP6	T1PP5	T1PP4	T1PP3	T1PP2	T1PP1	T1PP0
W	W	W	W	W	W	W	W

Initial value : FFH

T1PP[7:0] T1 PWM Period data

T1 (Timer 1 Register: Read Case) : B6H

7	6	5	4	3	2	1	0
T17	T16	T15	T14	T13	T12	T11	T10
R	R	R	R	R	R	R	R

Initial value : 00H

T1[7:0] T1 Counter Period data

T1PDR (Timer 1 PWM Duty Register) : B6H

7	6	5	4	3	2	1	0
T1PD7	T1PD6	T1PD5	T1PD4	T1PD3	T1PD2	T1PD1	T1PD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T1PD[7:0] T1 PWM Duty data
Note) only write, when PWM1E ‘1’

CDR1 (Capture 1 Data Register: Read Case) : B6H

7	6	5	4	3	2	1	0
CDR17	CDR16	CDR15	CDR14	CDR13	CDR12	CDR11	CDR10
R	R	R	R	R	R	R	R

Initial value : 00H

CDR1[7:0] T1 Capture data

T1PWHR (Timer 1 PWM High Register) : B7H

7	6	5	4	3	2	1	0
T1_PE	-	-	-	PW1H3	PW1H2	PW1H1	PW1H0
W	-	-	-	W	W	W	W

Initial value : 00H

T1_PE Control Timer 1 Output port operation
Note) only writable Bit. Be careful
0 Timer 1 Output disable
1 Timer 1 Output enable

PW1H[3:2] PWM period High value (Bit [9:8])

PW1H[1:0] PWM duty High value (Bit [9:8])

PERIOD:	PW1H3	PW1H2	T1PPR[7:0]
DUTY:	PW1H1	PW1H0	T1PDR[7:0]

TFLG (Timer Interrupt Flag Register) : 97H

7	6	5	4	3	2	1	0
-	-	T1IF	T0IF	-	-	-	-
-	-	R/W	R/W	-	-	-	-

Initial value : 00H

T1IF Timer1 Interrupt Flag

T0IF Timer0 Interrupt Flag

4.5.2 16-Bit Timer 4

4.5.2.1 Overview

The 16-bit timer 4 consists of Multiplexer, Timer Data Register High/Low, Timer Register High/Low, Timer Mode Control Register. It is able to use internal 16-bit timer/ counter without a port output function.

The 16-bit timer 4 is able to use the divided clock of the main clock selected from prescaler output.

4.5.2.2 16 Bit Timer/Counter Mode

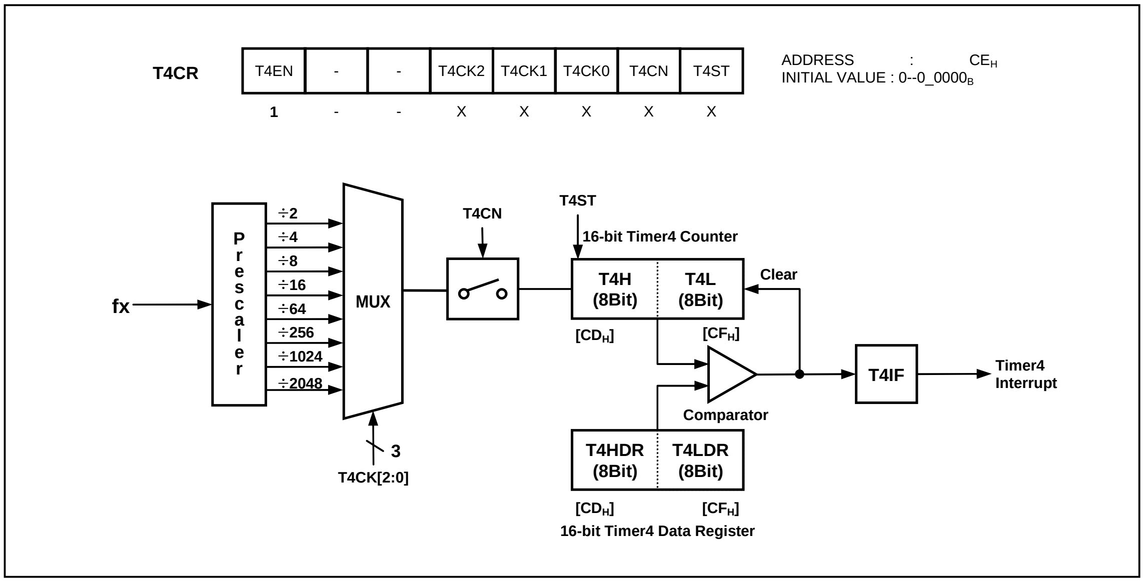


Figure 4.17 Timer4 16-bit Mode Block Diagram

4.5.2.3 Register Map

Table 4-8 Register Map

Name	Address	Dir	Default	Description
T4CR	0xCE	R/W	0H	Timer 4 Mode Control Register
T4L	0xCF	R	0H	Timer 4 Low Register
T4LDR	0xCF	W	FFH	Timer 4 Low Data Register
T4H	0xCD	R	0H	Timer 4 High Register
T4HDR	0xCD	R/W	0H	Timer 4 High Data Register

4.5.2.4 Timer 4 Register description

The timer 4 register consists of Timer 4 Mode Control Register (T4CR), Timer 4 Low Register (T4L), Timer 4 Low Data Register (T4LDR), Timer 4 High Register (T4H), Timer 4 High Data Register (T4HDR).

4.5.2.5 Register description for Timer 4

T4CR (Timer 4 Mode Control Register) : CEH

7	6	5	4	3	2	1	0
T4EN	-	-	T4CK2	T4CK1	T4CK0	T4CN	T4ST

R/W	-	-	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T4EN Control Timer 4 operation

0	Timer 4 disable
1	Timer 4 enable

T4CK[2:0] Select Timer 4 clock source. fx is main system clock frequency

T4CK2	T4CK1	T4CK0	Description
0	0	0	fx/2
0	0	1	fx/4
0	1	0	fx/8
0	1	1	fx/16
1	0	0	fx/64
1	0	1	fx/256
1	1	0	fx/1024
1	1	1	fx/2048

T4CN Control Timer 4 Count pause/continue

0	Temporary count stop
1	Continue count

T4ST Control Timer 4 start/stop

0	Counter stop
1	Clear Counter and start

T4L (Timer 4 Low Register: Read Case) : CFH

7	6	5	4	3	2	1	0
T4L7	T4L6	T4L5	T4L4	T4L3	T4L2	T4L1	T4L0
R	R	R	R	R	R	R	R

Initial value : 00H

T4L[7:0] T4L Counter

T4LDR (Timer 4 Low Data Register: Write Case) : CFH

7	6	5	4	3	2	1	0
T4LD7	T4LD6	T4LD5	T4LD4	T4LD3	T4LD2	T4LD1	T4LD0
W	W	W	W	W	W	W	W

Initial value : FFH

T4LD[7:0] T4L Compare

T4H (Timer 4 High Register: Read Case) : CDH

7	6	5	4	3	2	1	0
T4H7	T4H6	T4H5	T4H4	T4H3	T4H2	T4H1	T4H0
R	R	R	R	R	R	R	R

Initial value : 00H

T4H[7:0] T4H Counter Period

T4HDR (Timer 4 High Data Register: Write Case) : CDH

7	6	5	4	3	2	1	0

T4HD7	T4HD6	T4HD5	T4HD4	T4HD3	T4HD2	T4HD1	T4HD0
W	W	W	W	W	W	W	W

Initial value : FFH

T4HD[7:0] T4H Compare

TFLG (Timer Interrupt Flag Register) : 97H

7	6	5	4	3	2	1	0
-	T4IF	-	-	-	-	-	-
-	R/W	-	-	-	-	-	-

Initial value : 00H

T4IF Timer4 Interrupt Flag

4.6 USART

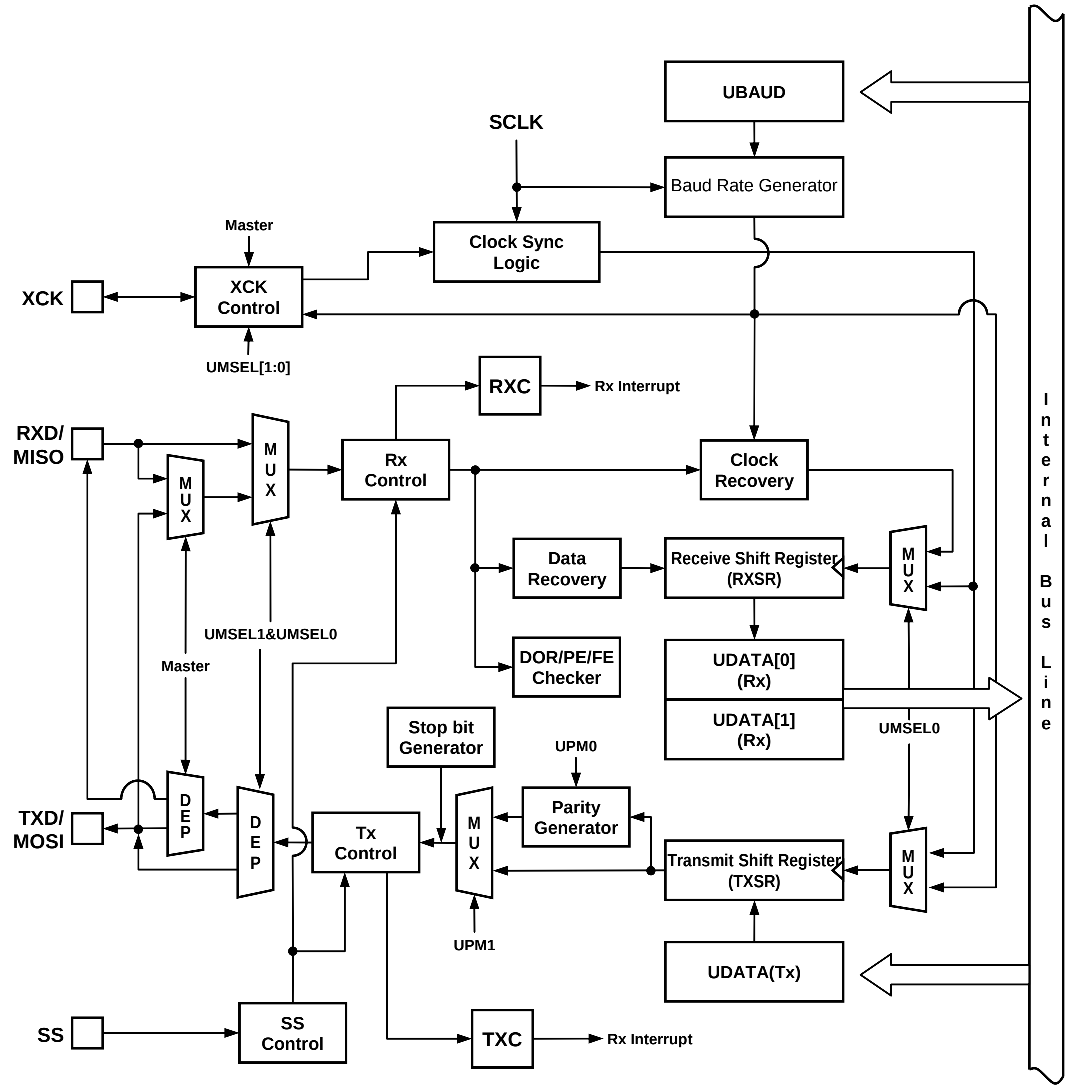
4.6.1 Overview

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are listed below.

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous and SPI Operation
- Supports all four SPI Modes of Operation (Mode 0, 1, 2, 3)
- LSB First or MSB First Data Transfer @SPI mode
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5,6,7,8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Double Speed Asynchronous Communication Mode

USART has three main parts of Clock Generator, Transmitter and Receiver. The Clock Generation logic consists of synchronization logic for external clock input used by synchronous or SPI slave operation, and the baud rate generator for asynchronous or master (synchronous or SPI) operation. The Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows a continuous transfer of data without any delay between frames. The receiver is the most complex part of the USART module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition to the recovery unit, the Receiver includes a parity checker, a shift register, a two level receive FIFO (UDATAn) and control logic. The Receiver supports the same frame formats as the Transmitter and can detect Frame Error, Data OverRun and Parity Errors.

4.6.2 Block Diagram



UCTRL1	UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1	USIZE0	UCPOL	ADDRESS : $E2_H$ INITIAL VALUE : 0000_0000_B
UCTRL2	UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X	ADDRESS : $E3_H$ INITIAL VALUE : 0000_0000_B
UCTRL3	MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8	ADDRESS : $E4_H$ INITIAL VALUE : 0000_-000_B
USTAT	UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE	ADDRESS : $E5_H$ INITIAL VALUE : 1000_0000_B

Figure 4.18 USART Block Diagram

4.6.3 Clock Generation

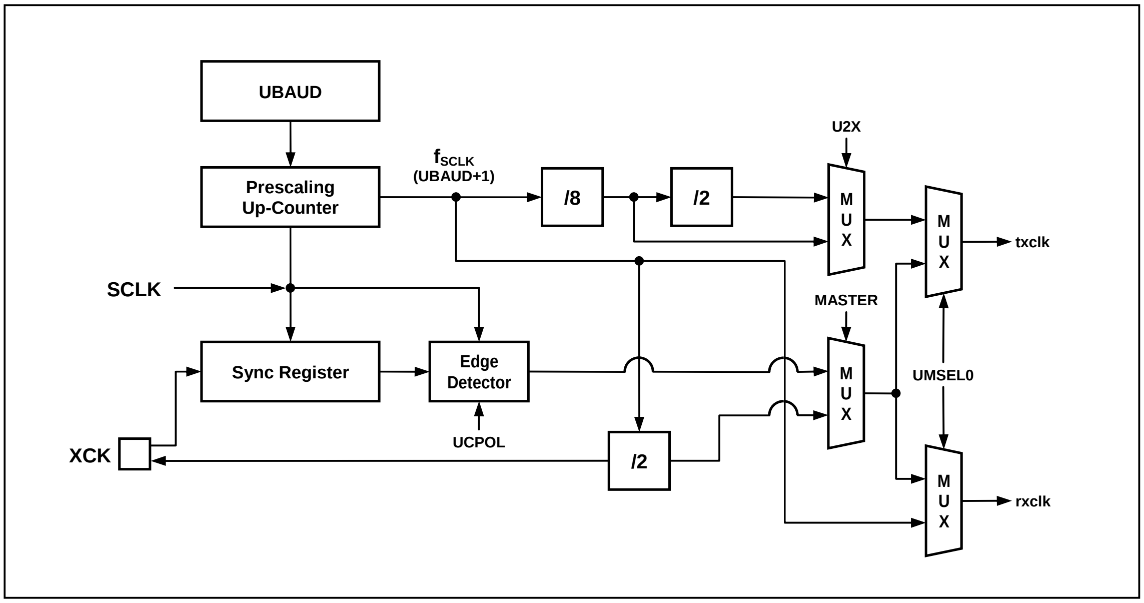


Figure 4.19 Clock Generation Block Diagram

The Clock generation logic generates the base clock for the Transmitter and Receiver. The USART supports four modes of clock operation and those are Normal Asynchronous, Double Speed Asynchronous, Master Synchronous and Slave Synchronous. The clock generation scheme for Master SPI and Slave SPI mode is the same as Master Synchronous and Slave Synchronous operation mode. The UMSELn bit in UCTRL1 register selects between asynchronous and synchronous operation. Asynchronous Double Speed mode is controlled by the U2X bit in the UCTRL2 register. The MASTER bit in UCTRL2 register controls whether the clock source is internal (Master mode, output port) or external (Slave mode, input port). The XCK pin is only active when the USART operates in Synchronous or SPI mode.

Table below contains equations for calculating the baud rate (in bps).

Table 4.9 Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (U2X=0)	$\text{Baud Rate} = \frac{\text{fSCLK}}{16(\text{UBAUD} + 1)}$
Asynchronous Double Speed Mode (U2X=1)	$\text{Baud Rate} = \frac{\text{fSCLK}}{8(\text{UBAUD} + 1)}$
Synchronous or SPI Master Mode	$\text{Baud Rate} = \frac{\text{fSCLK}}{2(\text{UBAUD} + 1)}$

4.6.4 External Clock (XCK)

External clocking is used by the synchronous or spi slave modes of operation.

External clock input from the XCK pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must then pass through an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and therefore the maximum frequency of the external XCK pin is limited by the following equation.

$$\mathrm{fXCK} = \frac{\mathrm{fSCLK}}{4}$$

where fXCK is the frequency of XCK and fSCLK is the frequency of main system clock (SCLK).

4.6.5 Synchronous mode operation

When synchronous or spi mode is used, the XCK pin will be used as either clock input (slave) or clock output (master). The dependency between the clock edges and data sampling or data change is the same. The basic principle is that data input on RXD (MISO in spi mode) pin is sampled at the opposite XCK clock edge of the edge in the data output on TXD (MOSI in spi mode) pin is changed.

The UCPOL bit in UCTRL1 register selects which XCK clock edge is used for data sampling and which is used for data change. As shown in the figure below, when UCPOL is zero the data will be changed at rising XCK edge and sampled at falling XCK edge.

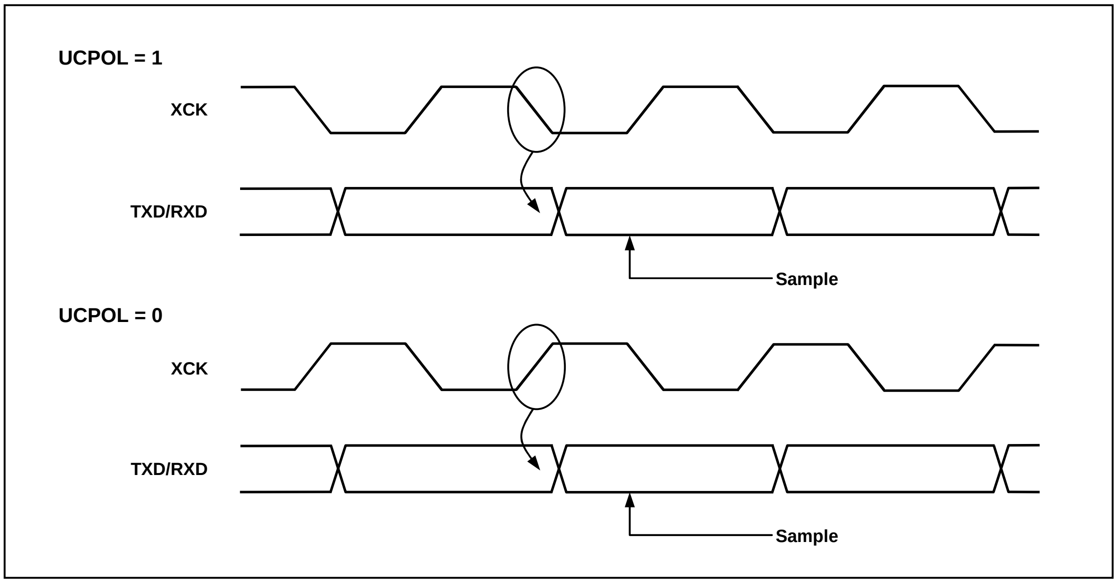


Figure 4.20 Synchronous Mode XCKn Timing.

4.6.6 Data format

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking.

The USART supports all 30 combinations of the following as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit (LSB). Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit (MSB). If enabled the parity bit is inserted after the data bits, before the stop bits. A high to low transition on data pin is considered as start bit. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle means high state of data pin. The next figure shows the possible combinations of the frame formats. Bits inside brackets are optional.

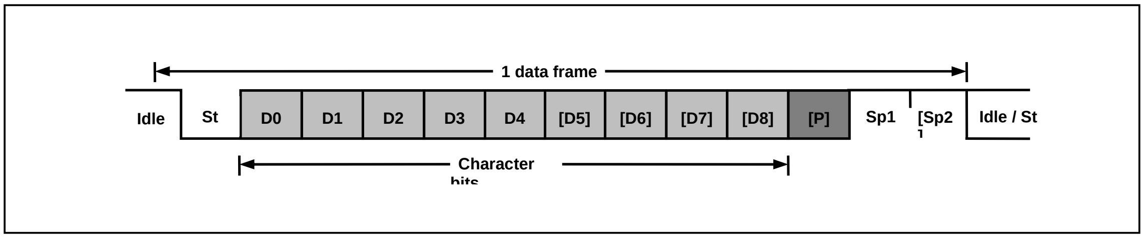


Figure 4.21 frame format

1 data frame consists of the following bits

- Idle No communication on communication line (TxD/RxD)
- St Start bit (Low)
- Dn Data bits (0~8)
- Parity bit ------------ Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

The frame format used by the USART is set by the USIZE[2:0], UPM[1:0] and USBS bits in UCTRL1 register. The Transmitter and Receiver use the same setting.

4.6.7 Priority bit

The parity bit is calculated by doing an exclusive-or of all the data bits. If odd parity is used, the result of the exclusive-or is inverted. The parity bit is located between the MSB and first stop bit of a serial frame.

$P_{even} = D_{n-1} \wedge \dots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$

$P_{odd} = D_{n-1} \wedge \dots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$

P_{even} : Parity bit using even parity

P_{odd} : Parity bit using odd parity

D_n : Data bit n of the character

4.6.8 USART Transmitter

The USART Transmitter is enabled by setting the TXE bit in UCTRL1 register. When the Transmitter is enabled, the normal port operation of the TXD pin is overridden by the serial output pin of USART. The baud-rate, operation mode and frame format must be setup once before doing any transmissions. If synchronous or spi operation is used, the clock on the XCK pin will be overridden and used as transmission clock. If USART operates in spi mode, SS pin is used as SS input pin in slave mode or can be configured as SS output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

Note: In Tx mode, the length of start bit can be shorter than one or two clock of the lengh of the other data bits.

4.6.8.1 Sending Tx data

A data transmission is initiated by loading the transmit buffer (UDATA register I/O location) with the data to be transmitted. The data written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame at the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode (USIZE[2:0]=7), the ninth bit must be written to the TX8 bit in UCTRL3 register before loading transmit buffer (UDATA register).

4.6.8.2 Transmitter flag and interrupt

The USART Transmitter has 2 flags which indicate its state. One is USART Data Register Empty (UDRE) and the other is Transmit Complete (TXC). Both flags can be interrupt sources.

UDRE flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted that has not yet been moved into the shift register. And also this flag can be cleared by writing ‘0’ to this bit position. Writing ‘1’ to this bit position is prevented.

When the Data Register Empty Interrupt Enable (UDRIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART Data Register Empty Interrupt is generated while UDRE flag is set.

The Transmit Complete (TXC) flag bit is set when the entire frame in the transmit shift register has been shifted out and there are no more data in the transmit buffer. The TXC flag is automatically cleared when the Transmit Complete Interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXC bit in UCTRL2 register.

When the Transmit Complete Interrupt Enable (TXCIE) bit in UCTRL2 register is set and the Global Interrupt is enabled, USART Transmit Complete Interrupt is generated while TXC flag is set.

4.6.8.3 Parity Generator

The Parity Generator calculates the parity bit for the sending serial frame data. When parity bit is enabled (UPM[1]=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the sending frame.

4.6.8.4 Disabling Transmitter

Disabling the Transmitter by clearing the TXE bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXD pin is used as normal General Purpose I/O (GPIO) or primary function pin.

4.6.9 USART Receiver

The USART Receiver is enabled by setting the RXE bit in the UCTRL1 register. When the Receiver is enabled, the normal pin operation of the RXD pin is overridden by the USART as the serial input pin of the Receiver. The baud-rate, mode of operation and frame format must be set before serial reception. If synchronous or spi operation is used, the clock on the XCK pin will be used as transfer clock. If USART operates in spi mode, SS pin is used as SS input pin in slave mode or can be configured as SS output pin in master mode. This can be done by setting SPISS bit in UCTRL3 register.

4.6.9.1 Receiving Rx data

When USART is in synchronous or asynchronous operation mode, the Receiver starts data reception when it detects a valid start bit (LOW) on RXD pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of XCK (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's 2nd stop bit in the frame, the 2nd stop bit is ignored by the Receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the UDATA register.

If 9-bit characters are used (USIZE[2:0] = 7) the ninth bit is stored in the RX8 bit position in the UCTRL3 register. The 9th bit must be read from the RX8 bit before reading the low 8 bits from the UDATA register. Likewise, the error flags FE, DOR, PE must be read before reading the data from UDATA register. This is because the error flags are stored in the same FIFO position of the receive buffer.

4.6.9.2 Receiver flag and interrupt

The USART Receiver has one flag that indicates the Receiver state.

The Receive Complete (RXC) flag indicates whether there are unread data present in the receive buffer. This flag is set when there are unread data in the receive buffer and cleared when the receive buffer is empty. If the Receiver is disabled (RXE=0), the receiver buffer is flushed and the RXC flag is cleared.

When the Receive Complete Interrupt Enable (RXCIE) bit in the UCTRL2 register is set and Global Interrupt is enabled, the USART Receiver Complete Interrupt is generated while RXC flag is set.

The USART Receiver has three error flags which are Frame Error (FE), Data OverRun (DOR) and Parity Error (PE). These error flags can be read from the USTAT register. As data received are stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from UDATA register, read the USTAT register first which contains error flags.

The Frame Error (FE) flag indicates the state of the first stop bit. The FE flag is zero when the stop bit was correctly detected as one, and the FE flag is one when the stop bit was incorrect, ie detected as zero. This flag can be used for detecting out-of-sync conditions between data frames.

The Data OverRun (DOR) flag indicates data loss due to a receive buffer full condition. A DOR occurs when the receive buffer is full, and another new data is present in the receive shift register which are to be stored into the receive buffer. After the DOR flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The Parity Error (PE) flag indicates that the frame in the receive buffer had a Parity Error when received. If Parity Check function is not enabled (UPM[1]=0), the PE bit is always read zero.

Note) The error flags related to receive operation are not used when USART is in spi mode.

4.6.9.3 Parity Checker

If Parity Bit is enabled (UPM[1]=1), the Parity Checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame.

4.6.9.4 Disabling Receiver

In contrast to Transmitter, disabling the Receiver by clearing RXE bit makes the Receiver inactive immediately. When the Receiver is disabled the Receiver flushes the receive buffer and the remaining data in the buffer is all reset. The RXD pin is not overridden the function of USART, so RXD pin becomes normal GPIO or primary function pin.

4.6.9.5 Asynchronous Data Reception

To receive asynchronous data frame, the USART includes a clock and data recovery unit. The Clock Recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXD pin.

The Data recovery logic samples and low pass filters the incoming bits, and this removes the noise of RXD pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times the baud-rate for normal mode, and 8 times the baud rate for Double Speed mode (U2X=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the Double Speed mode.

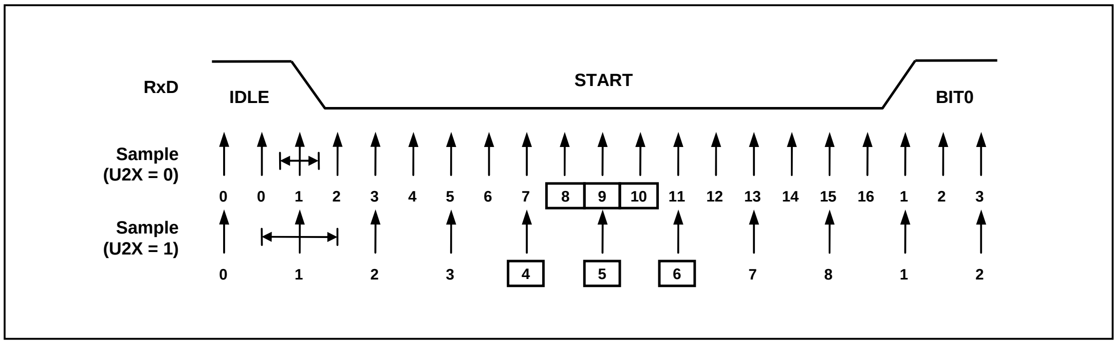


Figure 4.22 Start Bit Sampling

on

on the RXD line, the start bit condition. After detecting high to low transition on RXD line, the clock recovery logic uses samples 8,9, and 10 for Normal mode, and samples 4, 5, and 6 for Double Speed mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the Receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost similar to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for Normal mode and 8 times for Double Speed mode. And uses sample 8, 9, and 10 to decide data value for Normal mode, samples 4, 5, and 6 for Double Speed mode. If more than 2 samples have low levels, the received bit is considered to a logic 0 and more than 2 samples have high levels, the received bit is considered to a logic 1. The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

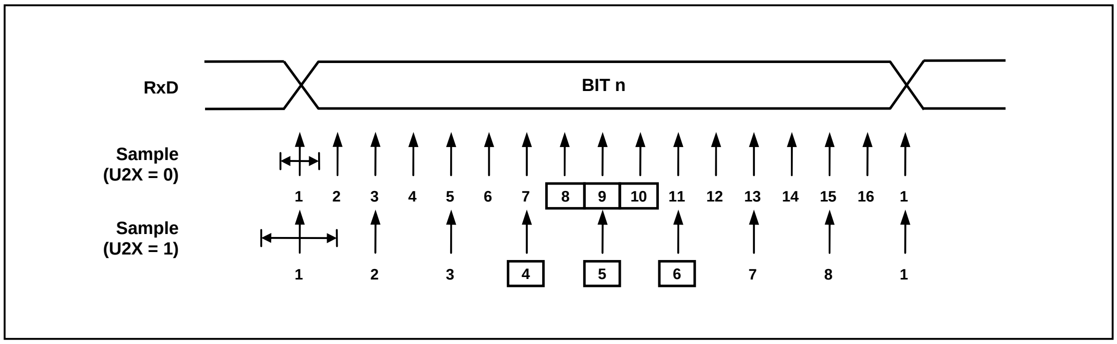


Figure 4.23 Sampling of Data and Parity Bit

ore samples of 3 center values have high level, correct stop bit is detected, else a Frame Error flag is set. After deciding first stop bit whether a valid stop bit is received or not, the Receiver goes idle state and monitors the RXD line to check a valid high to low transition is detected (start bit detection).

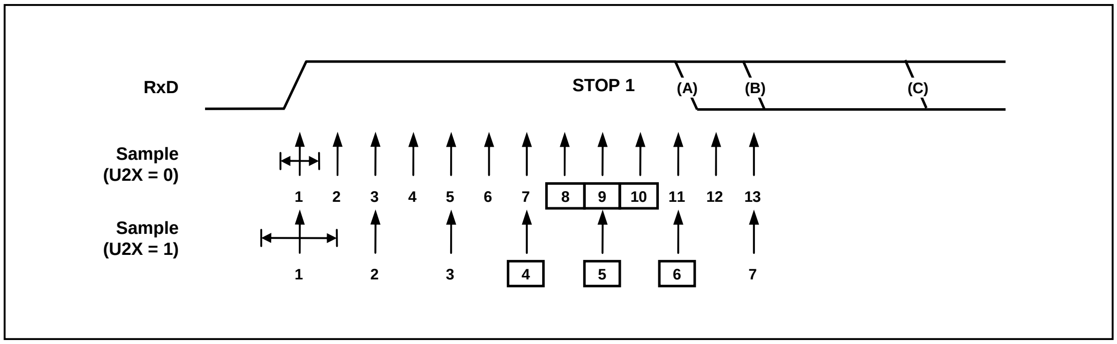


Figure 4.24 Stop Bit Sampling and Next Start Bit Sampling

4.6.10 SPI Mode

The USART can be set to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

- Full duplex, three-wire synchronous data transfer
- Master or Slave operation
- Supports all four SPI modes of operation (mode0, 1, 2, and 3)
- Selectable LSB first or MSB first data transfer
- Double buffered transmit and receive
- Programmable transmit bit rate

When SPI mode is enabled (UMSEL[1:0]=3), the Slave Select (SS) pin becomes active low input in slave mode operation, or can be output in master mode operation if SPISS bit is set.

Note that during SPI mode of operation, the pin RXD is renamed as MISO and TXD is renamed as MOSI for compatibility to other SPI devices.

4.6.10.1 SPI Clock formats and timing

To accommodate a wide variety of synchronous serial peripherals from different manufacturers, the USART has a clock polarity bit (UCPOL) and a clock phase control bit (UCPHA) to select one of four clock formats for data transfers. UCPOL selectively insert an inverter in series with the clock. UCPHA chooses between two different clock phase relationships between the clock and data. Note that

UCPHA and UCPOL bits in UCTRL1 register have different meanings according to the UMSEL[1:0] bits which decides the operating mode of USART.

Table below shows four combinations of UCPOL and UCPHA for SPI mode 0, 1, 2, and 3.

Table 4.10 CPOL Functionality

SPI Mode	UCPOL	UCPHA	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

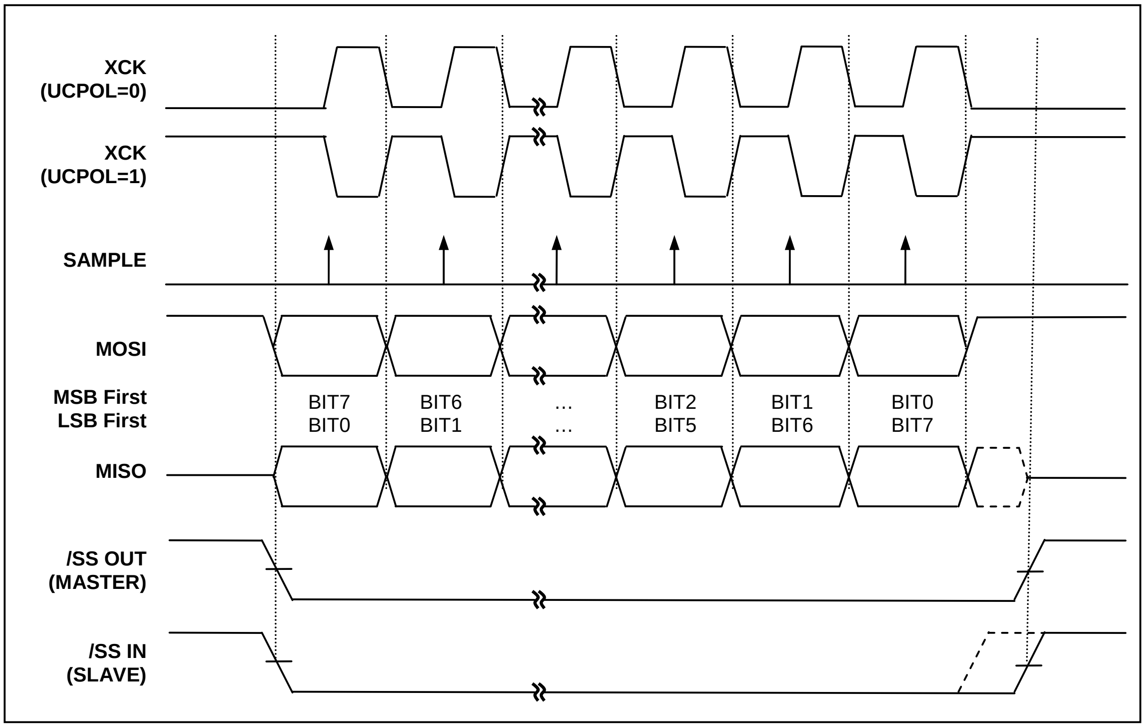


Figure 4.25 SPI Clock Formats when UCPHA=0

to active low. The first XCK edge causes both the master and the slave to sample the data bit value on their MISO and MOSI inputs, respectively. At the second XCK edge, the USART shifts the second data bit value out to the MOSI and MISO outputs of the master and slave, respectively. Unlike the case of UCPHA=1, when UCPHA=0, the slave's SS input must go to its inactive high level between transfers. This is because the slave can prepare the first data bit when it detects falling edge of SS input.

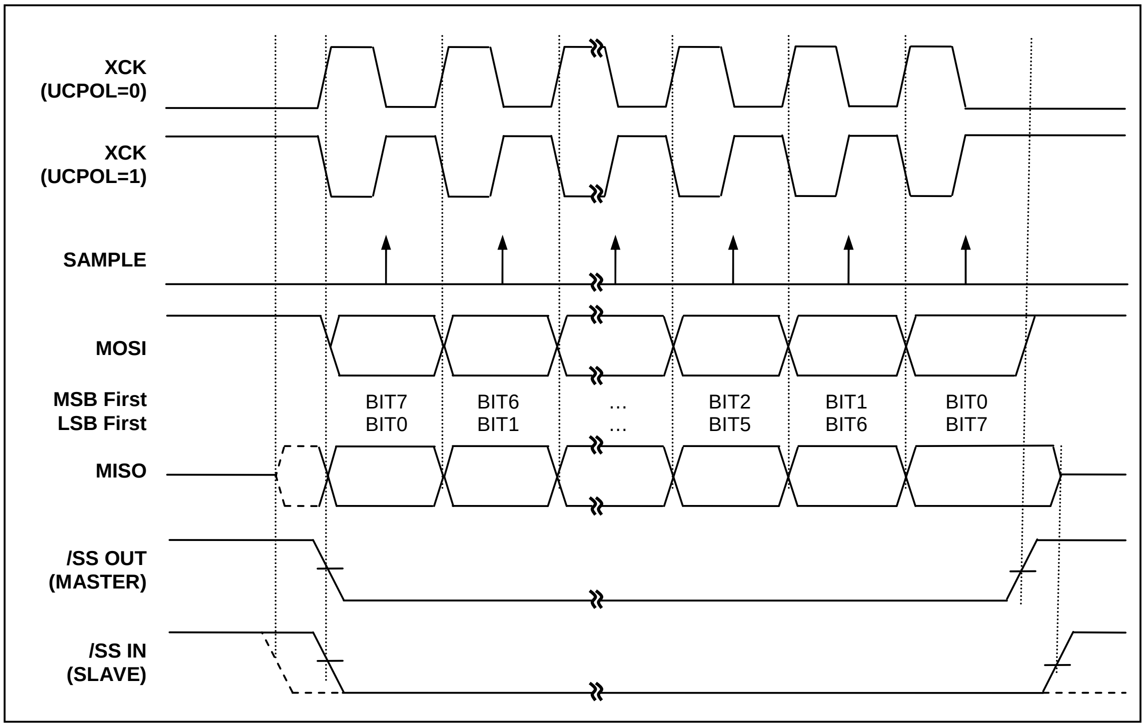


Figure 4.26 SPI Clock Formats when UCPHA=1

When UCPHA=1, the slave begins to drive its MISO output when SS goes active low, but the data is not defined until the first XCK edge. The first XCK edge shifts the first bit of data from the shifter onto the MOSI output of the master and the MISO output of the slave. The next XCK edge causes both the master and slave to sample the data bit value on their MISO and MOSI inputs, respectively. At the third XCK edge, the USART shifts the second data bit value out to the MOSI and MISO output of the master and slave respectively. When UCPHA=1, the slave's SS input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USART resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USART Data Register Empty flag (UDRE=1) and then writing a byte of data to the UDATA Register. In master mode of operation, even if transmission is not enabled (TXE=0), writing data to the UDATA register is necessary because the clock XCK is generated from transmitter block.

4.6.11 Register Map

Table 4.11 USART Register Map

Name	Address	Dir	Default	Description
UCTRL1	E2H	R/W	00H	USART Control 1 Register
UCTRL2	E3H	R/W	00H	USART Control 2 Register
UCTRL3	E4H	R/W	00H	USART Control 3 Register
USTAT	E5H	R	80H	USART Status Register
UBAUD	E6H	R/W	FFH	USART Baud Rate Generation Register
UDATA	E7H	R/W	FFH	USART Data Register

4.6.12 USART Register description

USART module consists of USART Control 1 Register (UCTRL1), USART Control 2 Register (UCTRL2), USART Control 3 Register (UCTRL3), USART Status Register (USTAT), USART Data Register (UDATA), and USART Baud Rate Generation Register (UBAUD).

4.6.13 Register description for USART

UCTRL1 (USART Control 1 Register) : E2H

7	6	5	4	3	2	1	0
UMSEL1	UMSEL0	UPM1	UPM0	USIZE2	USIZE1 UDORD	USIZE0 UCPHA	UCPOL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

UMSEL[1:0] Selects operation mode of USART.

UMSEL1	UMSEL0	Operation Mode
0	0	Asynchronous Mode (Uart)
0	1	Synchronous Mode
1	0	Reserved
1	1	SPI Mode

UPM[1:0] Selects Parity Generation and Check methods

UPM1	UPM0	Parity
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

USIZE[2:0] When in asynchronous or synchronous mode of operation, selects the length of data bits in frame.

USIZE2	USIZE1	USIZE0	Data Length
0	0	0	5 bit
0	0	1	6 bit
0	1	0	7 bit
0	1	1	8 bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9 bit

UDORD This bit is in the same bit position with USIZE1. In SPI mode, when set to one the MSB of the data byte is transmitted first. When set to zero the LSB of the data byte is transmitted first.

0	LSB First
1	MSB First

UCPOL Selects polarity of XCK in synchronous or spi mode

0	TXD change @Rising Edge, RXD change @Falling Edge
1	TXD change @ Falling Edge, RXD change @ Rising Edge

UCPHA This bit is in the same bit position with USIZE0. In SPI mode, along with UCPOL bit, selects one of two clock formats for different kinds of synchronous serial peripherals. Leading edge means first XCK edge and trailing edge means 2nd or last clock edge of XCK in one XCK pulse. And Sample means detecting of incoming receive bit, Setup means preparing transmit data.

UCPOL	UCPHA	Leading Edge	Trailing Edge
0	0	Sample (Rising)	Setup (Falling)
0	1	Setup (Rising)	Sample (Falling)
1	0	Sample (Falling)	Setup (Rising)
1	1	Setup (Falling)	Sample (Rising)

UCTRL2 (USART Control 2 Register) : E3H

7	6	5	4	3	2	1	0
UDRIE	TXCIE	RXCIE	WAKEIE	TXE	RXE	USARTEN	U2X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

UDRIE Interrupt enable bit for USART Data Register Empty.

0 Interrupt from UDRE is inhibited (use polling)

1 When UDRE is set, request an interrupt

TXCIE Interrupt enable bit for Transmit Complete.

0 Interrupt from TXC is inhibited (use polling)

1 When TXC is set, request an interrupt

RXCIE Interrupt enable bit for Receive Complete

0 Interrupt from RXC is inhibited (use polling)

1 When RXC is set, request an interrupt

WAKEIE Interrupt enable bit for Asynchronous Wake in STOP mode. When device is in stop mode, if RXD goes to LOW level an interrupt can be requested to wake-up system.

0 Interrupt from Wake is inhibited

1 When WAKE is set, request an interrupt

TXE Enables the transmitter unit.

0 Transmitter is disabled

1 Transmitter is enabled

RXE Enables the receiver unit.

0 Receiver is disabled

1 Receiver is enabled

USARTEN Activate USART module by supplying clock.

0 USART is disabled (clock is halted)

1 USART is enabled

U2X This bit only has effect for the asynchronous operation and selects receiver sampling rate.

0 Normal asynchronous operation

1 Double Speed asynchronous operation

UCTRL3 (USART Control 3 Register) : E4H

7	6	5	4	3	2	1	0
MASTER	LOOPS	DISXCK	SPISS	-	USBS	TX8	RX8
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

Initial value : 00H

MASTER Selects master or slave in SPI or Synchronous mode operation and controls the direction of XCK pin.

0 Slave mode operation and XCK is input pin.

1 Master mode operation and XCK is output pin

LOOPS Controls the Loop Back mode of USART, for test mode

0 Normal operation

1 Loop Back mode

DISXCK In Synchronous mode of operation, selects the waveform of XCK output.

0 XCK is free-running while USART is enabled in synchronous master mode.

1 XCK is active while any frame is on transferring.

SPISS Controls the functionality of SS pin in master SPI mode.

0 SS pin is normal GPIO or other primary function

1 SS output to other slave device

USBS Selects the length of stop bit in Asynchronous or Synchronous mode of operation.

0 1 Stop Bit

1 2 Stop Bit

TX8 The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Write this bit first before loading the UDATA register.

0 MSB (9th bit) to be transmitted is ‘0’

1 MSB (9th bit) to be transmitted is ‘1’

RX8 The ninth bit of data frame in Asynchronous or Synchronous mode of operation. Read this bit first before reading the receive buffer.

0 MSB (9th bit) received is ‘0’

1 MSB (9th bit) received is ‘1’

USTAT (USART Status Register) : E5H

7	6	5	4	3	2	1	0
UDRE	TXC	RXC	WAKE	SOFTRST	DOR	FE	PE
R/W	R/W	R/W	R/W	R/W	R	R	R

Initial value : 80H

UDRE The UDRE flag indicates if the transmit buffer (UDATA) is ready to receive new data. If UDRE is ‘1’, the buffer is empty and ready to be written. This flag can generate a UDRE interrupt.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXC This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer. This flag is automatically cleared when the interrupt service routine of a TXC interrupt is executed. This flag can generate a TXC interrupt.

0 Transmission is ongoing.

1 Transmit buffer is empty and the data in transmit shift register are shifted out completely.

RXC This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read. The RXC flag can be used to generate a RXC interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKE This flag is set when the RX pin is detected low while the CPU is in stop mode. This flag can be used to generate a WAKE interrupt. This bit is set only when in asynchronous mode of operation.

0 No WAKE interrupt is generated.

1 WAKE interrupt is generated

SOFTRST This is an internal reset and only has effect on USART. Writing ‘1’ to this bit initializes the internal logic of USART and is auto cleared.

0 No operation

1 Reset USART

DOR This bit is set if a Data OverRun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.

0 No Data OverRun

1 Data OverRun detected

FE This bit is set if the first stop bit of next character in the receive buffer is detected as ‘0’. This bit is valid until the receive buffer is read.

0 No Frame Error

1 Frame Error detected

PE This bit is set if the next character in the receive buffer has a Parity Error when received while Parity Checking is enabled. This bit is valid until the receive buffer is read.

0 No Parity Error

1 Parity Error detected

UBAUD (USART Baud-Rate Generation Register) : E6H

7	6	5	4	3	2	1	0
UBAUD7	UBAUD6	UBAUD5	UBAUD4	UBAUD3	UBAUD2	UBAUD1	UBAUD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

UBAUD [7:0] The value in this register is used to generate internal baud rate in asynchronous mode or to generate XCK clock in synchronous or spi mode. To prevent malfunction, do not write ‘0’ in asynchronous mode, and do not write ‘0’ or ‘1’ in synchronous or spi mode.

UDATA (USART Data Register) : E7H

7	6	5	4	3	2	1	0
UDATA7	UDATA6	UDATA5	UDATA4	UDATA3	UDATA2	UDATA1	UDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

UDATA [7:0] The USART Transmit Buffer and Receive Buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the UDATA register. Reading the UDATA register returns the contents of the Receive Buffer.

Write this register only when the UDRE flag is set. In spi or synchronous master mode, write this register even if TX is not enabled to generate clock, XCK.

4.6.14 Baud Rate setting (example)

Table 4.12 Examples of UBAUD Settings for Commonly Used Oscillator Frequencies

Baud Rate	fOSC=1.00MHz				fOSC=1.8432MHz				fOSC=2.00MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4K	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%

19.2K	2	8.5%	6	-7.0%	5	0.0%	11	0.0%	6	-7.0%	12	0.2%
28.8K	1	8.5%	3	8.5%	3	0.0%	7	0.0%	3	8.5%	8	-3.5%
38.4K	1	-18.6%	2	8.5%	2	0.0%	5	0.0%	2	8.5%	6	-7.0%
57.6K	-	-	1	8.5%	1	-25.0%	3	0.0%	1	8.5%	3	8.5%
76.8K	-	-	1	-18.6%	1	0.0%	2	0.0%	1	-18.6%	2	8.5%
115.2K	-	-	-	-	-	-	1	0.0%	-	-	1	8.5%
230.4K	-	-	-	-	-	-	-	-	-	-	-	-

Table 4.13 Examples of UBAUD Settings for Commonly Used Oscillator Frequencies (continued)

Baud Rate	fOSC=3.6864MHz				fOSC=4.00MHz				fOSC=7.3728MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	95	0.0%	191	0.0%	103	0.2%	207	0.2%	191	0.0%	-	-
4800	47	0.0%	95	0.0%	51	0.2%	103	0.2%	95	0.0%	191	0.0%
9600	23	0.0%	47	0.0%	25	0.2%	51	0.2%	47	0.0%	95	0.0%
14.4K	15	0.0%	31	0.0%	16	2.1%	34	-0.8%	31	0.0%	63	0.0%
19.2K	11	0.0%	23	0.0%	12	0.2%	25	0.2%	23	0.0%	47	0.0%
28.8K	7	0.0%	15	0.0%	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
38.4K	5	0.0%	11	0.0%	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
57.6K	3	0.0%	7	0.0%	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
76.8K	2	0.0%	5	0.0%	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
115.2K	1	0.0%	3	0.0%	1	8.5%	3	8.5%	3	0.0%	7	0.0%
230.4K	-	-	1	0.0%	-	-	1	8.5%	1	0.0%	3	0.0%
250K	-	-	1	-7.8%	-	-	1	0.0%	1	-7.8%	3	-7.8%
0.5M	-	-	-	-	-	-	-	-	-	-	1	-7.8%

Table 4.14 Examples of UBAUD Settings for Commonly Used Oscillator Frequencies (continued)

Baud Rate	fOSC=8.00MHz				fOSC=11.0592MHz				fOSC=14.7456MHz			
	U2X=0		U2X=1		U2X=0		U2X=1		U2X=0		U2X=1	
	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR	UBAUD	ERROR
2400	207	0.2%	-	-	-	-	-	-	-	-	-	-
4800	103	0.2%	207	0.2%	143	0.0%	-	-	191	0.0%	-	-
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4K	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2K	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8K	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4K	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6K	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8K	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2K	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4K	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250K	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	-	-	1	0.0%	-	-	2	-7.8%	1	-7.8%	3	-7.8%
1M	-	-	-	-	-	-	-	-	-	-	1	-7.8%

4.7 I2C

4.7.1 Overview

The I2C is one of industrial standard serial communication protocols, and which uses 2 bus lines Serial Data Line (SDA) and Serial Clock Line (SCL) to exchange data. Because both SDA and SCL lines are open-drain output, each line needs pull-up resistor. The features are as shown below.

- Compatible with I2C bus standard
- Multi-master operation
- Up to 400 KHz data transfer speed
- 7 bit address
- Both master and slave operation
- Bus busy detection

4.7.2 Block Diagram

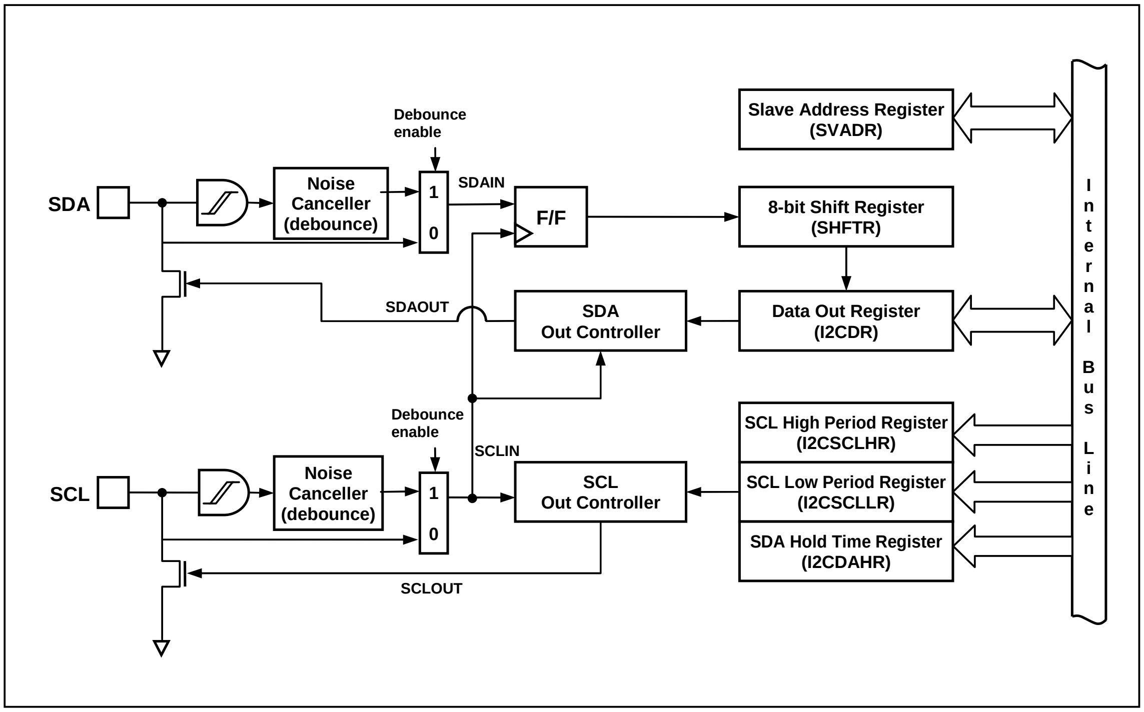

SDA
Debounce enable
Noise Canceller (debounce)
1
0
SDAIN
F/F
SDAOUT
SDA Out Controller
Slave Address Register (SVADR)
8-bit Shift Register (SHFTR)
Data Out Register (I2CDR)
SCL
Debounce enable
Noise Canceller (debounce)
1
0
SCLIN
SCL Out Controller
SCLOUT
SCL High Period Register (I2CSCLHR)
SCL Low Period Register (I2CSCLLR)
SDA Hold Time Register (I2CDAHR)
Internal Bus Line

Figure 4.27 I2C Block Diagram

4.7.3 I2C Bit Transfer

The data on the SDA line must be stable during HIGH period of the clock, SCL. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

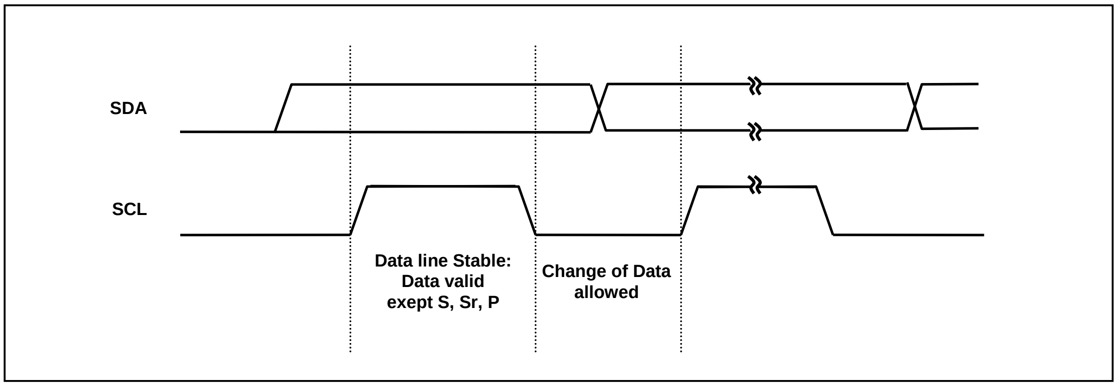


Figure 4.28 Bit Transfer on the I2C-Bus

4.7.4 START / REPEATED START / STOP

One master can issue a START (S) condition to notice other devices connected to the SCL, SDA lines that it will use the bus. A STOP (P) condition is generated by the master to release the bus lines so that other devices can use it.

A high to low transition on the SDA line while SCL is high defines a START (S) condition.

A low to high transition on the SDA line while SCL is high defines a STOP (P) condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after START condition. The bus is considered to be free again after STOP condition, ie, the bus is busy between START and STOP condition. If a repeated START condition (Sr) is generated instead of STOP condition, the bus stays busy. So, the START and repeated START conditions are functionally identical.

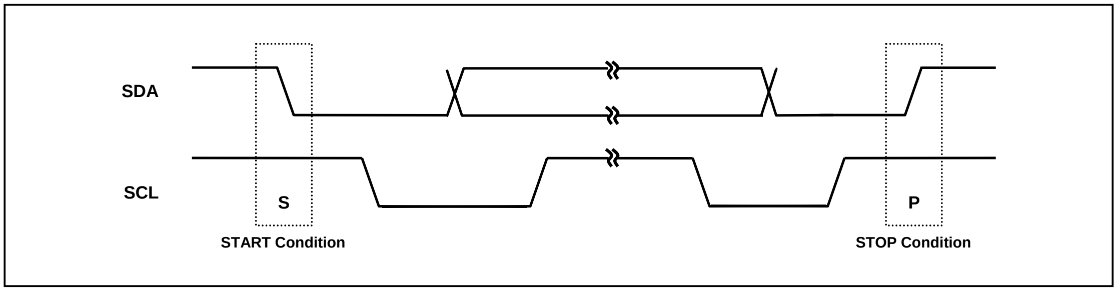


Figure 4.29 START and STOP Condition

4.7.5 DATA TRANSFER

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unlimited. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave can’t receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCL LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL.

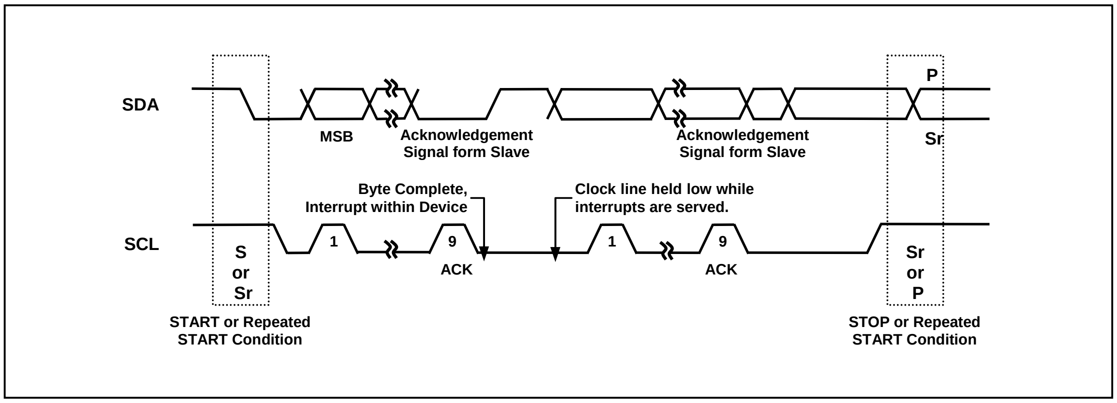

SDA
P
MSB
Acknowledgement Signal form Slave
Acknowledgement Signal form Slave
Sr
Byte Complete, Interrupt within Device
Clock line held low while interrupts are served.
SCL
S or Sr
1
9
ACK
1
9
ACK
Sr or P
START or Repeated START Condition
STOP or Repeated START Condition

Figure 4.30 Data Transfer on the I2C-Bus

4.7.6 ACKNOWLEDGE

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse. When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because it’s performing some real time function, the data line must be left HIGH by the slave. And also, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDA line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

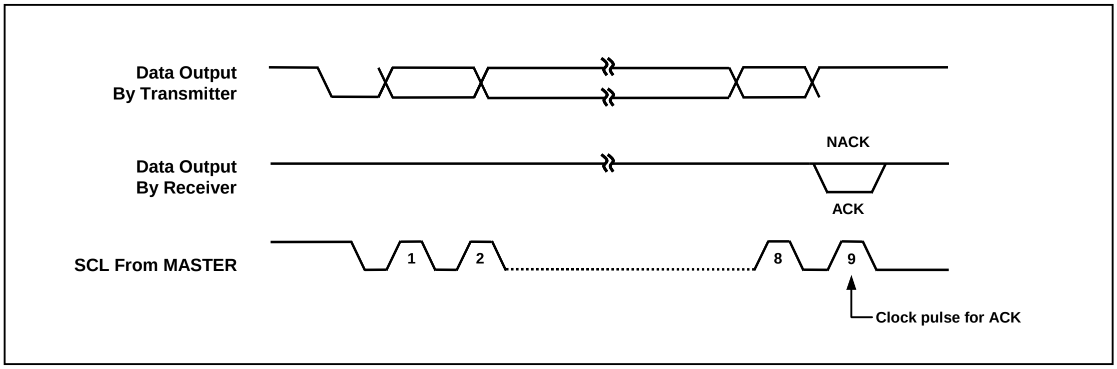


Figure 4.31 Acknowledge on the I2C-Bus

4.7.7 SYNCHRONIZATION / ARBITRATION

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL line. This means that a HIGH to LOW transition on the SCL line will cause the devices concerned to start counting off their LOW period and it will hold the SCL line in that state until the clock HIGH state is reached. However the LOW to HIGH transition of this clock may not change the state of the SCL line if another clock is still within its LOW period. In this way, a synchronized SCL clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition. Arbitration takes place on the SDA line, while the SCL line is at the HIGH level, in such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output state because the level on the bus doesn't correspond to its own level. Arbitration continues for many bits until a winning master gets the ownership of I2C bus. Its first stage is comparison of the address bits.

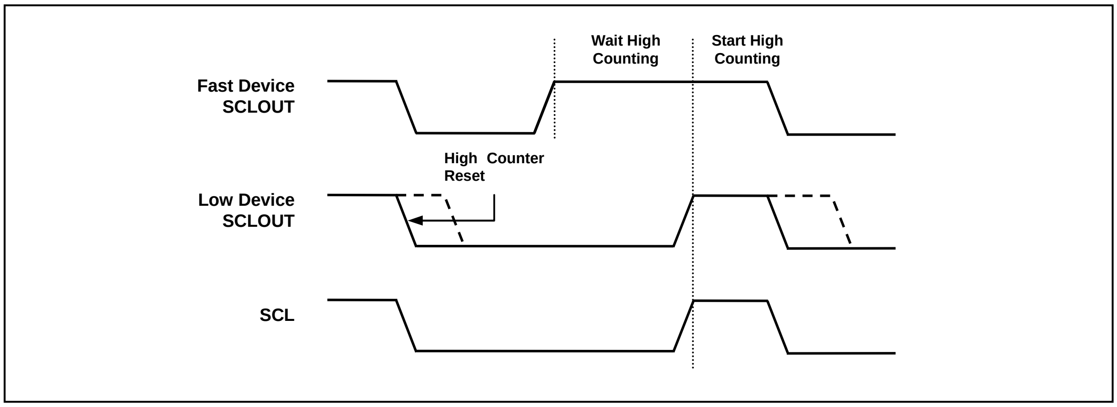


Figure 4.32 Clock Synchronization during Arbitration Procedure

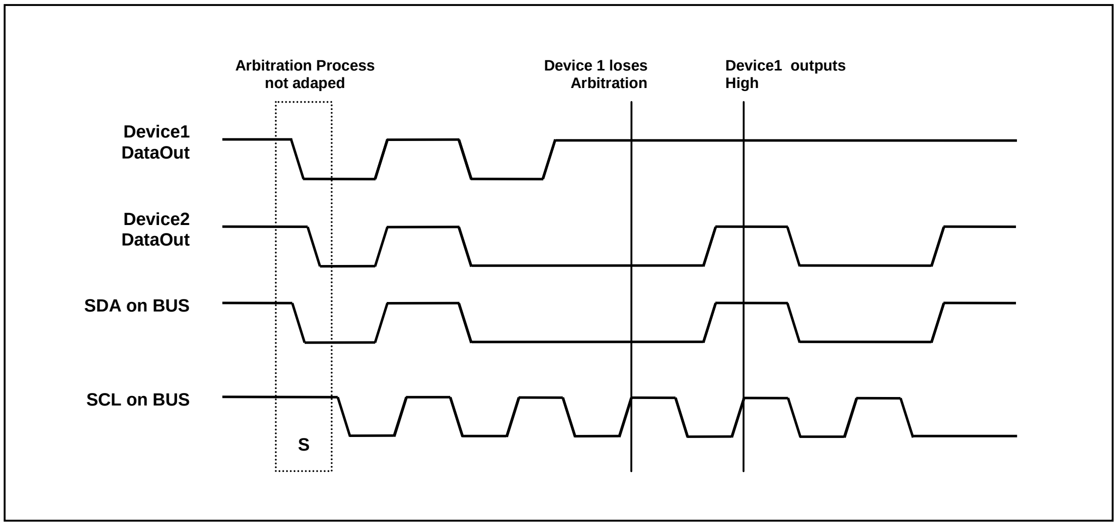


Figure 4.33 Arbitration Procedure of Two Masters

4.7.8 OPERATION

The I2C is byte-oriented and interrupt based. Interrupts are issued after all bus events except for a transmission of a START condition. Because the I2C is interrupt based, the application software is free to carry on other operations during a I2C byte transfer.

Note that when a I2C interrupt is generated, IIF flag in I2CMR register is set, it is cleared by writing an arbitrary value to I2CSR. When I2C interrupt occurs, the SCL line is hold LOW until writing any value to I2CSR. When the IIF flag is set, the I2CSR contains a value indicating the current state of the I2C bus. According to the value in I2CSR, software can decide what to do next.

I2C can operate in 4 modes by configuring master/slave, transmitter/receiver. The operating mode is configured by a winning master. A more detailed explanation follows below.

4.7.8.1 Master Transmitter

To operate I2C in master transmitter, follow the recommended steps below.

1. Enable I2C by setting IICEN bit in I2CMR. This provides main clock to the peripheral.
2. Load SLA+W into the I2CDR where SLA is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is ‘0’. Note that I2CDR is used for both address and data.
3. Configure baud rate by writing desired value to both I2CSCLLR and I2CSCLHR for the Low and High period of SCL line.
4. Configure the I2CSDAHR to decide when SDA changes value from falling edge of SCL. If SDA should change in the middle of SCL LOW period, load half the value of I2CSCLLR to the I2CSDAHR.
5. Set the START bit in I2CMR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the START bit is set, 8-bit data in I2CDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCL. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOST bit in I2CSR is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLSOT bit in I2CSR is set, the ACKEN bit in I2CMR must be set and the received 7-bit address must equal to the SLA bits in I2CSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL LOW. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (Master) can choose one of the following cases regardless of the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to I2CDR.
   2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOP bit in I2CMR.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2CDR and set START bit in I2CMR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR and if transfer direction bit is ‘1’ go to master receiver section.
7. 1-Byte of data is being transmitted. During data transfer, bus arbitration continues.
8. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCL LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOST bit in I2CSR is set. If then, I2C waits in idle state. When the data in I2CDR is transmitted completely, I2C generates TEND interrupt.

   I2C can choose one of the following cases regardless of the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to I2CDR.
   2) Master stops data transfer even if it receives ACK signal from slave. In this case, set the STOP bit in I2CMR.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLA+R/W into the I2CDR and set the START bit in I2CMR.

After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR, and if transfer direction bit is ‘1’ go to master receiver section.

9. This is the final step for master transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

The next figure depicts above process for master transmitter operation of I2C.

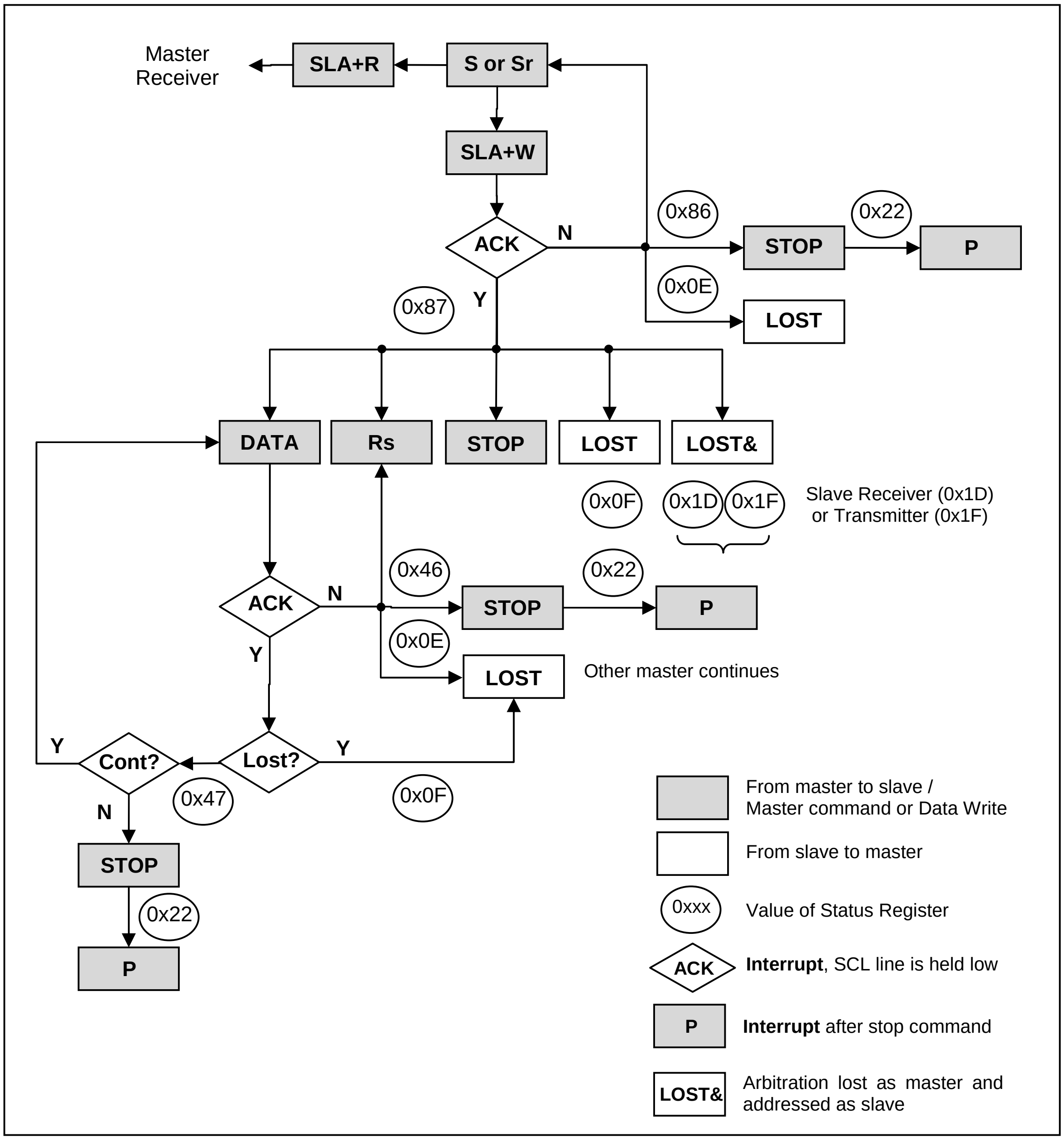


Figure 4.34 Formats and States in the Master Transmitter Mode

4.7.8.2 Master Receiver

To operate I2C in master receiver, follow the recommended steps below.

1. Enable I2C by setting IICEN bit in I2CMR. This provides main clock to the peripheral.
2. Load SLA+R into the I2CDR where SLA is address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is '1'. Note that I2CDR is used for both address and data.
3. Configure baud rate by writing desired value to both I2CSCLLR and I2CSCLHR for the Low and High period of SCL line.
4. Configure the I2CSDAHR to decide when SDA changes value from falling edge of SCL. If SDA should change in the middle of SCL LOW period, load half the value of I2CSCLLR to the I2CSDAHR.
5. Set the START bit in I2CMR. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the START bit is set, 8-bit data in I2CDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCL. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOST bit in I2CSR is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLSOT bit in I2CSR is set, the ACKEN bit in I2CMR must be set and the received 7-bit address must equal to the SLA bits in I2CSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCL LOW. This is because to decide whether I2C continues serial transfer or stops communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can prepare and transmit more data to master. Configure ACKEN bit in I2CMR to decide whether I2C ACKnowledges the next data to be received or not.
   2) Master stops data transfer because it receives no ACK signal from slave. In this case, set the STOP bit in I2CMR.
   3) Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLA+R/W into the I2CDR and set START bit in I2CMR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in I2CDR and if transfer direction bit is '0' go to master transmitter section.

7. 1-Byte of data is being received.
8. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCL LOW. When 1-Byte of data is received completely, I2C generates TEND interrupt.

   I2C can choose one of the following cases according to the RXACK flag in I2CSR.

   1) Master continues receiving data from slave. To do this, set ACKEN bit in I2CMR to ACKnowledge the next data to be received.
   2) Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKEN bit in I2CMR.
   3) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOP bit in I2CMR.
   4) No ACK signal is detected, and master transmits repeated START condition. In this case,

load SLA+R/W into the I2CDR and set the START bit in I2CMR.

After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1) and 2), move to step 7. In case of 3), move to step 9 to handle STOP interrupt. In case of 4), move to step 6 after transmitting the data in I2CDR, and if transfer direction bit is '0' go to master transmitter section.

9. This is the final step for master receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

The processes described above for master receiver operation of I2C can be depicted as the following figure.

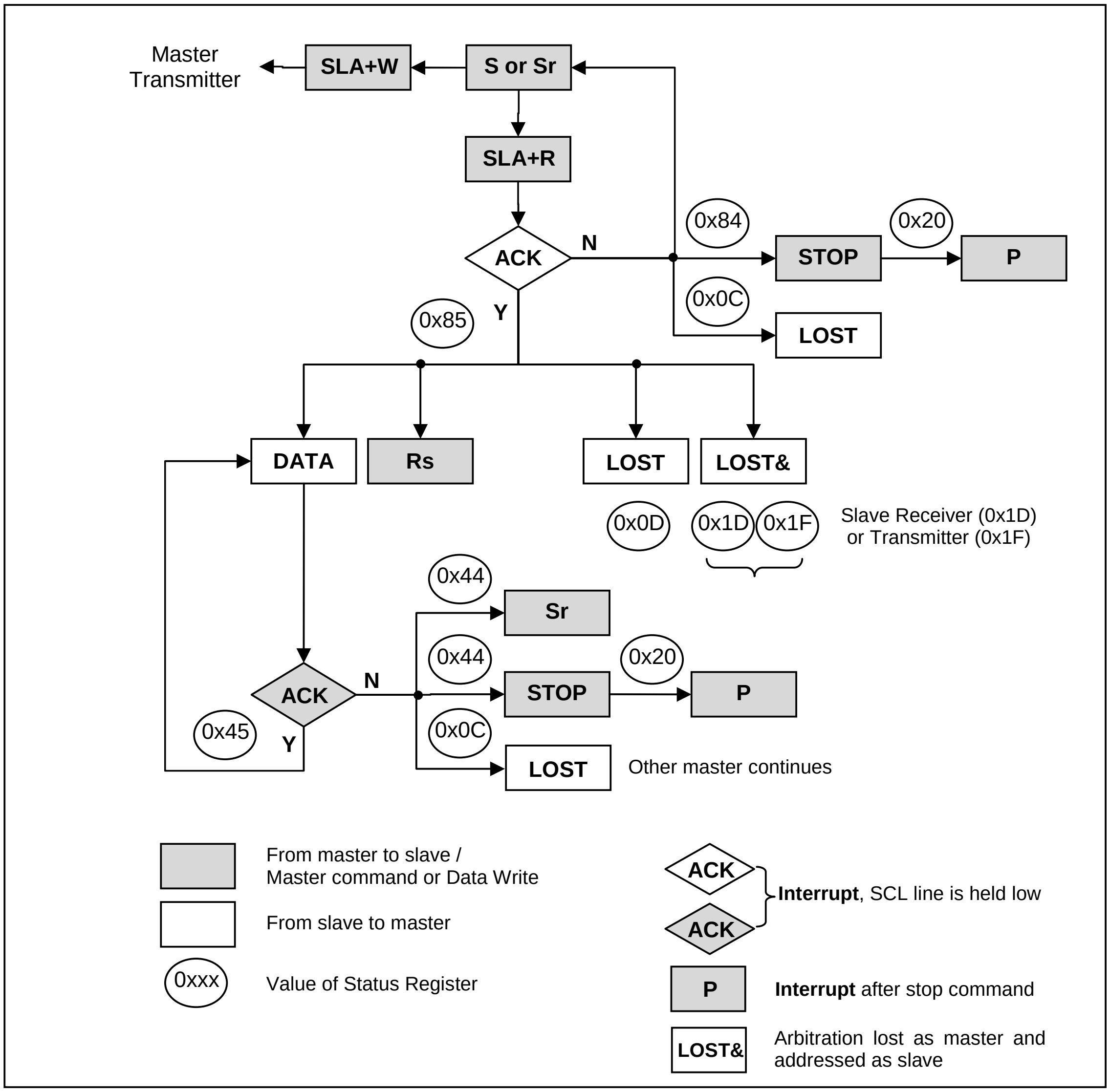


Figure 4.35 Formats and States in the Master Receiver Mode

4.7.8.3 Slave Transmitter

To operate I2C in slave transmitter, follow the recommended steps below.

1. If the main operating clock (SCLK) of the system is slower than that of SCL, load value 0x00 into I2CSDAHR to make SDA change within one system clock period from the falling edge of SCL. Note that the hold time of SDA is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CSDAHR. When the hold time of SDA is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting IICEN bit and INTEN bit in I2CMR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2CSAR. If the GCALLEN bit in I2CSAR is enabled, I2C compares the received data with value 0x00, the general call address.

   Note: General call interrupt can occur as though the received data does not match the general call address. When general call interrupt happens, I2CDR must be checked to match 0x00.
4. If the received address does not equal to SLA bits in I2CSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLA bits and the ACKEN bit is enabled, I2C generates SSEL interrupt and the SCL line is held LOW. Note that even if the address equals to SLA bits, when the ACKEN bit is disabled, I2C enters idle state. When SSEL interrupt occurs, load transmit data to I2CDR and write arbitrary value to I2CSR to release SCL line.
5. 1-Byte of data is being transmitted.
6. In this step, I2C generates TEND interrupt and holds the SCL line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   1) No ACK signal is detected and I2C waits STOP or repeated START condition.
   2) ACK signal from master is detected. Load data to transmit into I2CDR.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

The next figure shows flow chart for handling slave transmitter function of I2C.

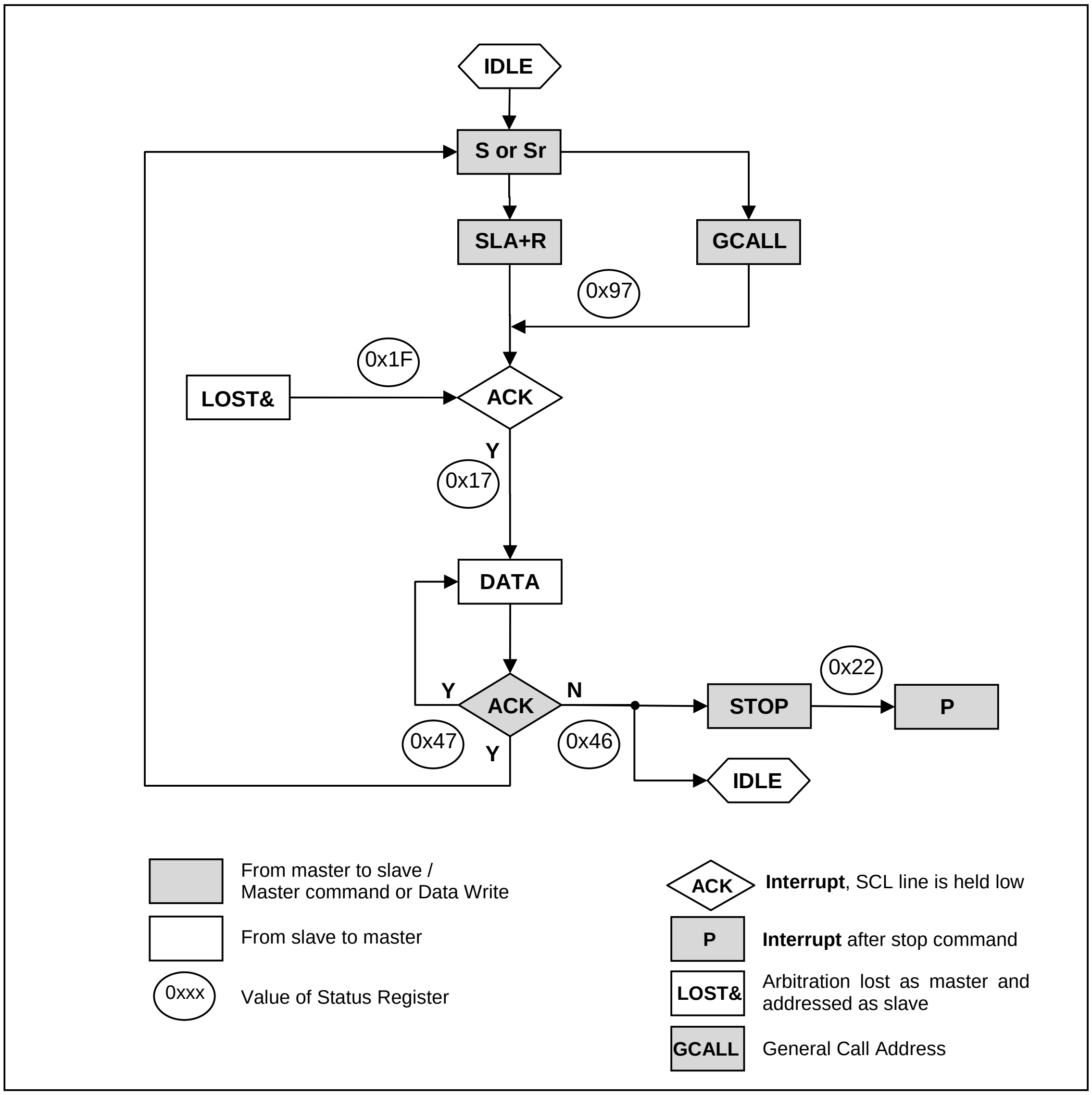


Figure 4.36 Formats and States in the Slave Transmitter Mode

4.7.8.4 Slave Receiver

To operate I2C in slave receiver, follow the recommended steps below.

1. If the main operating clock (SCLK) of the system is slower than that of SCL, load value 0x00 into I2CSDAHR to make SDA change within one system clock period from the falling edge of SCL. Note that the hold time of SDA is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from I2CSDAHR. When the hold time of SDA is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting IICEN bit and INTEN bit in I2CMR. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with SLA bits in I2CSAR. If the GCALLEN bit in I2CSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLA bits in I2CSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLA bits and the ACKEN bit is enabled, I2C generates SSEL interrupt and the SCL line is held LOW. Note that even if the

address equals to SLA bits, when the ACKEN bit is disabled, I2C enters idle state. When SSEL interrupt occurs and I2C is ready to receive data, write arbitrary value to I2CSR to release SCL line.

5. 1-Byte of data is being received.
6. In this step, I2C generates TEND interrupt and holds the SCL line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   1) No ACK signal is detected (ACKEN=0) and I2C waits STOP or repeated START condition.
   2) ACK signal is detected (ACKEN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, write arbitrary value to I2CSR to release SCL line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear I2CSR, write arbitrary value to I2CSR. After this, I2C enters idle state.

The process can be depicted as following figure when I2C operates in slave receiver mode.

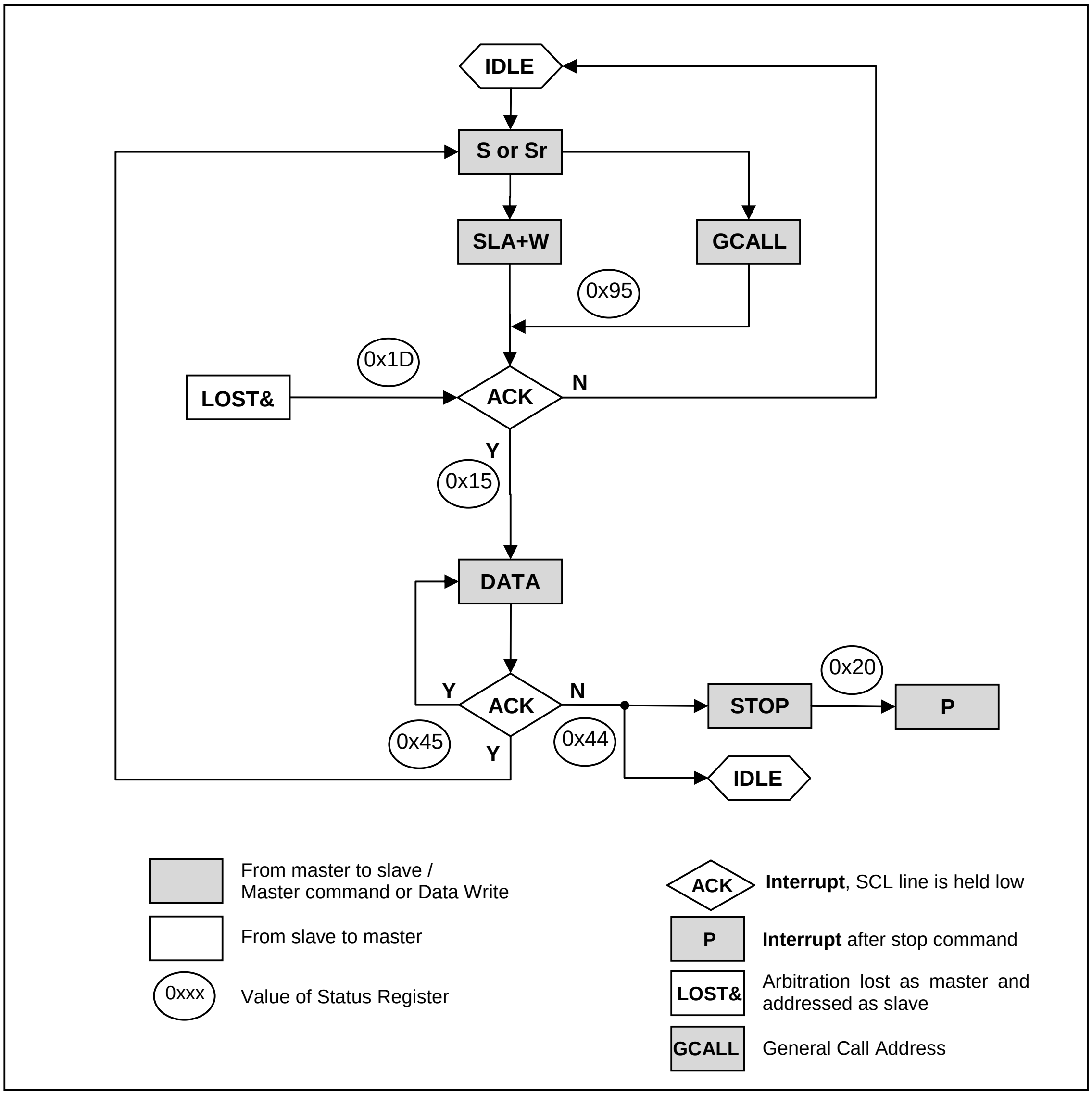


Figure 4.37 Formats and States in the Slave Receiver Mode

4.7.9 Register Map

Table 4.15 I2C Register Map

Name	Address	Dir	Default	Description
I2CMR	DAH	R/W	00H	I2C Mode Control Register
I2CSR	DBH	R	00H	I2C Status Register
I2CSCLLR	DCH	R/W	3FH	SCL Low Period Register
I2CSCLHR	DDH	R/W	3FH	SCL High Period Register
I2CSDAHR	DEH	R/W	01H	SDA Hold Time Register
I2CDR	DFH	R/W	FFH	I2C Data Register
I2CSAR	D7H	R/W	00H	I2C Slave Address Register

4.7.10 I2C Register description

I2C Registers are composed of I2C Mode Control Register (I2CMR), I2C Status Register (I2CSR), SCL Low Period Register (I2CSCLLR), SCL High Period Register (I2CSCLHR), SDA Hold Time Register (I2CSDAHR), I2C Data Register (I2CDR), and I2C Slave Address Register (I2CSAR).

4.7.11 Register description for I2C

I2CMR (I2C Mode Control Register) : DAH

7	6	5	4	3	2	1	0
IIF	IICEN	RESET	INTEN	ACKEN	IMASTERI	STOP	START
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IIF This is interrupt flag bit.

0 No interrupt is generated or interrupt is cleared

1 An interrupt is generated

IICEN Enable I2C Function Block (by providing clock)

0 I2C is inactive

1 I2C is active

RESET Initialize internal registers of I2C.

0 No operation

1 Initialize I2C, auto cleared

INTEN Enable interrupt generation of I2C.

0 Disable interrupt, operates in polling mode

1 Enable interrupt

ACKEN Controls ACK signal generation at ninth SCL period.

Note) ACK signal is output (SDA=0) for the following 3 cases.

When received address packet equals to SLA bits in I2CSAR

When received address packet equals to value 0x00 with GCALL enabled

When I2C operates as a receiver (master or slave)

0 No ACK signal is generated (SDA=1)

1 ACK signal is generated (SDA=0)

MASTER This bit shows whether I2C is in master or slave mode.

0 I2C is in slave mode.

1 I2C is in master mode.

STOP When I2C is master, generates STOP condition.

0 No operation

1 STOP condition is to be generated

START When I2C is master, generates START condition.

0 No operation

1 START or repeated START condition is to be generated

I2CSR (I2C Status Register) : DBH

7	6	5	4	3	2	1	0
GCALL	TEND	STOP	SSEL	MLOST	BUSY	TMODE	RXACK
R	R	R	R	R	R	R	R

Initial value : 00H

GCALL This bit has different meaning depending on whether I2C is master or slave. Note 1)

When I2C is a master, this bit represents whether it received AACK (Address ACK) from slave.

When I2C is a slave, this bit is used to indicate general call.

0 No AACK is received (Master mode)

1 AACK is received (Master mode)

0 Received address is not general call address (Slave mode)

1 General call address is detected (Slave mode)

TEND This bit is set when 1-Byte of data is transferred completely. Note 1)

0 1 byte of data is not completely transferred

1 1 byte of data is completely transferred

STOP This bit is set when STOP condition is detected. Note 1)

0 No STOP condition is detected

1 STOP condition is detected

SSEL This bit is set when I2C is addressed by other master. Note 1)

0 I2C is not selected as slave

1 I2C is addressed by other master and acts as a slave

MLOST This bit represents the result of bus arbitration in master mode. Note 1)

0 I2C maintains bus mastership

1 I2C has lost bus mastership during arbitration process

BUSY This bit reflects bus status.

0 I2C bus is idle, so any master can issue a START condition

1 I2C bus is busy

TMODE This bit is used to indicate whether I2C is transmitter or receiver.

0 I2C is a receiver

1 I2C is a transmitter

RXACK This bit shows the state of ACK signal.

0 No ACK is received

1 ACK is generated at ninth SCL period

Note 1) These bits can be source of interrupt.

When an I2C interrupt occurs except for STOP interrupt, the SCL line is hold LOW. To release SCL, write arbitrary value to I2CSR. When I2CSR is written, the TEND, STOP, SSEL, LOST, RXACK bits are cleared.

I2CSCLLR (SCL Low Period Register) : DCH

7	6	5	4	3	2	1	0
SCLL7	SCLL6	SCLL5	SCLL4	SCLL3	SCLL2	SCLL1	SCLL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3FH

SCLL[7:0] This register defines the LOW period of SCL when I2C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula : t_{SCLK} × (SCLL + 1) where t_{SCLK} is the period of SCLK.

I2CSCLHR (SCL High Period Register) : DDH

7	6	5	4	3	2	1	0
SCLH7	SCLH6	SCLH5	SCLH4	SCLH3	SCLH2	SCLH1	SCLH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3FH

SCLH[7:0] This register defines the HIGH period of SCL when I2C operates in master mode. The base clock is SCLK, the system clock, and the period is calculated by the formula : t_{SCLK} × (SCLH + 3) where t_{SCLK} is the period of SCLK.

So, the operating frequency of I2C in master mode (fI2C) is calculated by the following equation.

$$\mathrm{fI2C} = \frac{1}{\mathrm{tSCLK} \times (\mathrm{SCLL} + \mathrm{SCLH} + 4)}$$

I2CSDAHR (SDA Hold Time Register) : DEH

7	6	5	4	3	2	1	0
SDAH7	SDAH6	SDAH5	SDAH4	SDAH3	SDAH2	SDAH1	SDAH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 01H

SDAH[7:0] This register is used to control SDA output timing from the falling edge of SCL. Note that SDA is changed after t_{SCLK} × SDAH. In master mode, load half the value of SCLL to this register to make SDA change in the middle of SCL. In slave mode, configure this register regarding the frequency of SCL from master. The SDA is changed after t_{SCLK} × (SDAH + 1). So, to insure normal operation in slave mode, the value t_{SCLK} × (SDAH + 1) must be smaller than the period of SCL.

I2CDR (I2C Data Register) : DFH

7	6	5	4	3	2	1	0
ICD7	ICD6	ICD5	ICD4	ICD3	ICD2	ICD1	ICD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

ICD[7:0] When I2C is configured as a transmitter, load this register with data to be transmitted. When I2C is a receiver, the received data is stored into this register.

I2CSAR (I2C Slave Address Register) : D7H

7	6	5	4	3	2	1	0

SLA7	SLA6	SLA5	SLA4	SLA3	SLA2	SLA1	GCALLEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

SLA[7:1] These bits configure the slave address of this I2C module when I2C operates in slave mode.

GCALLEN This bit decides whether I2C allows general call address or not when I2C operates in slave mode.

0	Ignore general call address
1	Allow general call address

4.8 12-Bit A/D Converter

4.8.1 Overview

The analog-to-digital converter (A/D) allows conversion of an analog input signal to a corresponding 12-bit digital value. The A/D module has tenth analog inputs. The output of the multiplex is the input into the converter, which generates the result via successive approximation. The A/D module has four registers which are the control register ADCM (A/D Converter Mode Register), ADCM2 (A/D Converter Mode Register 2) and A/D result register ADCHR (A/D Converter Result High Register) and ADCLR (A/D Converter Result Low Register). It is selected for the corresponding channel to be converted by setting ADSEL[3:0]. To executing A/D conversion, ADST bit sets to ‘1’. The register ADCHR and ADCLR contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADCHR and ADCLR, the A/D conversion status bit AFLAG is set to ‘1’, and the A/D interrupt is set. For processing A/D conversion, AFLAG bit is read as ‘0’. If using STBY (power down) bit, the ADC is disabled. Also internal timer, external generating event, comparator, the trigger of timer1pwm and etc. can start ADC regardless of interrupt occurrence.

ADC Conversion Time = ADCLK * 60 cycles

After STBY bit is reset (ADC power enable) and it is restarted, during some cycle, ADC conversion value may have an inaccurate value.

4.8.2 Block Diagram

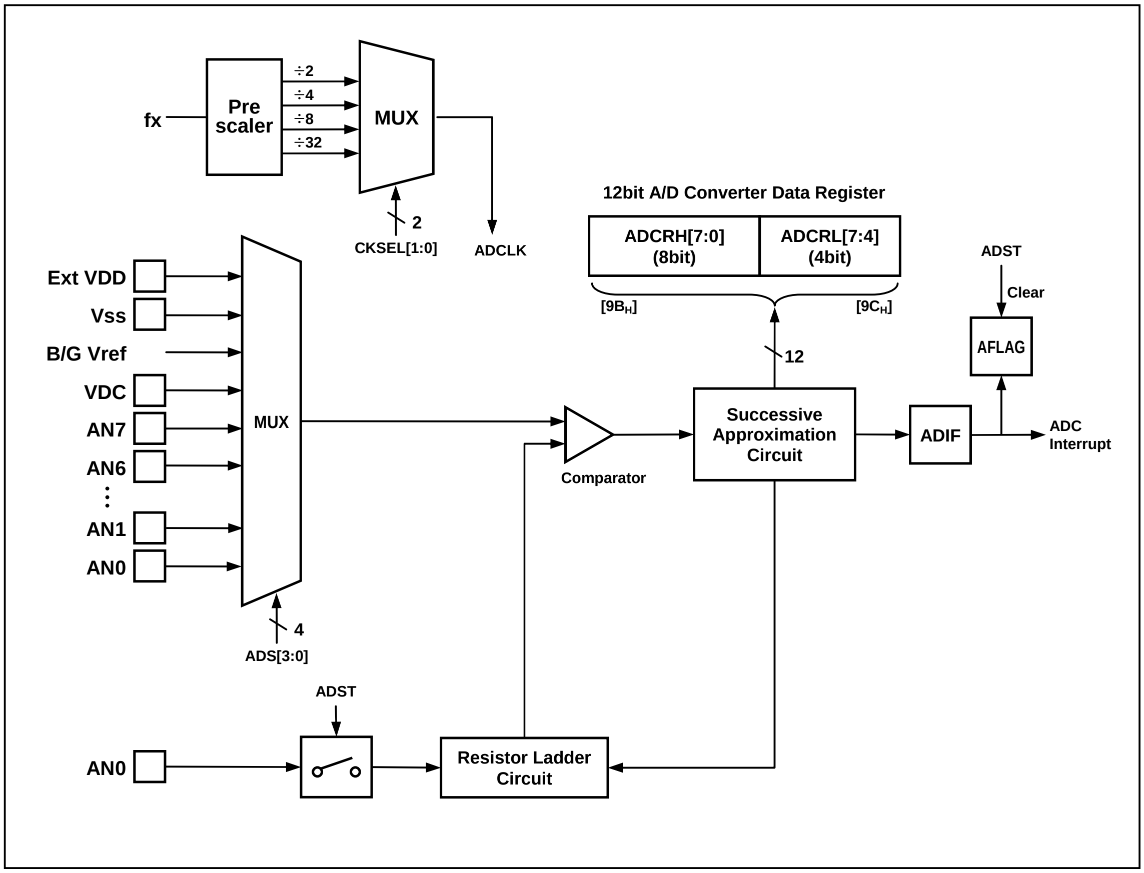


Figure 4.38 ADC Block Diagram

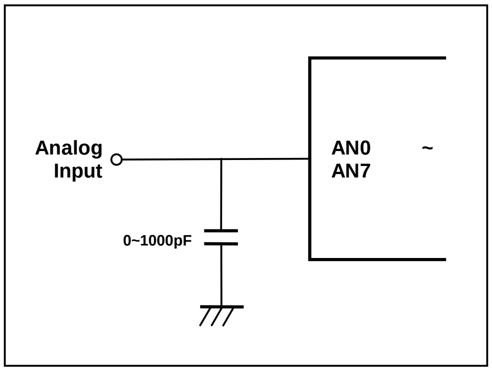


Figure 4.39 A/D Analog Input Pin Connecting Capacitor

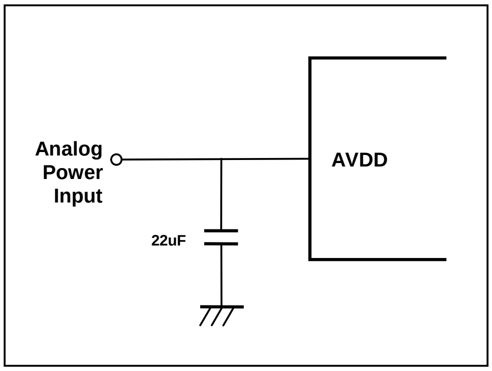


Figure 4.40 A/D Power(AVDD) Pin Connecting Capacitor

4.8.3 ADC Operation

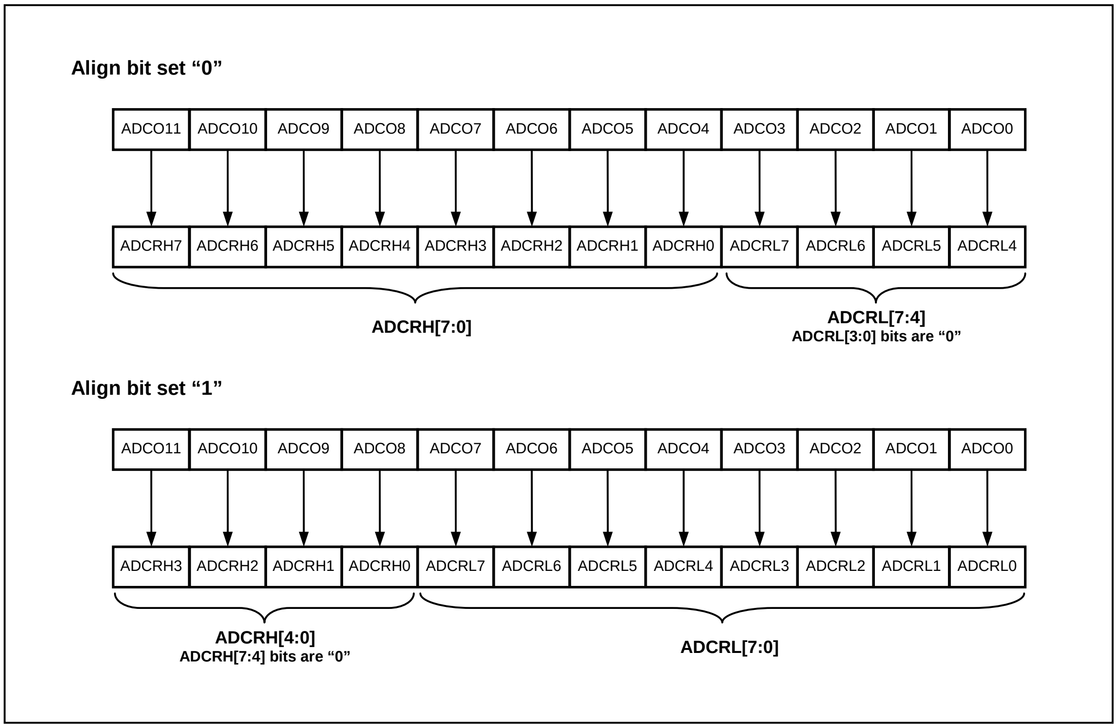


Figure 4.41 ADC Operation for Align bit

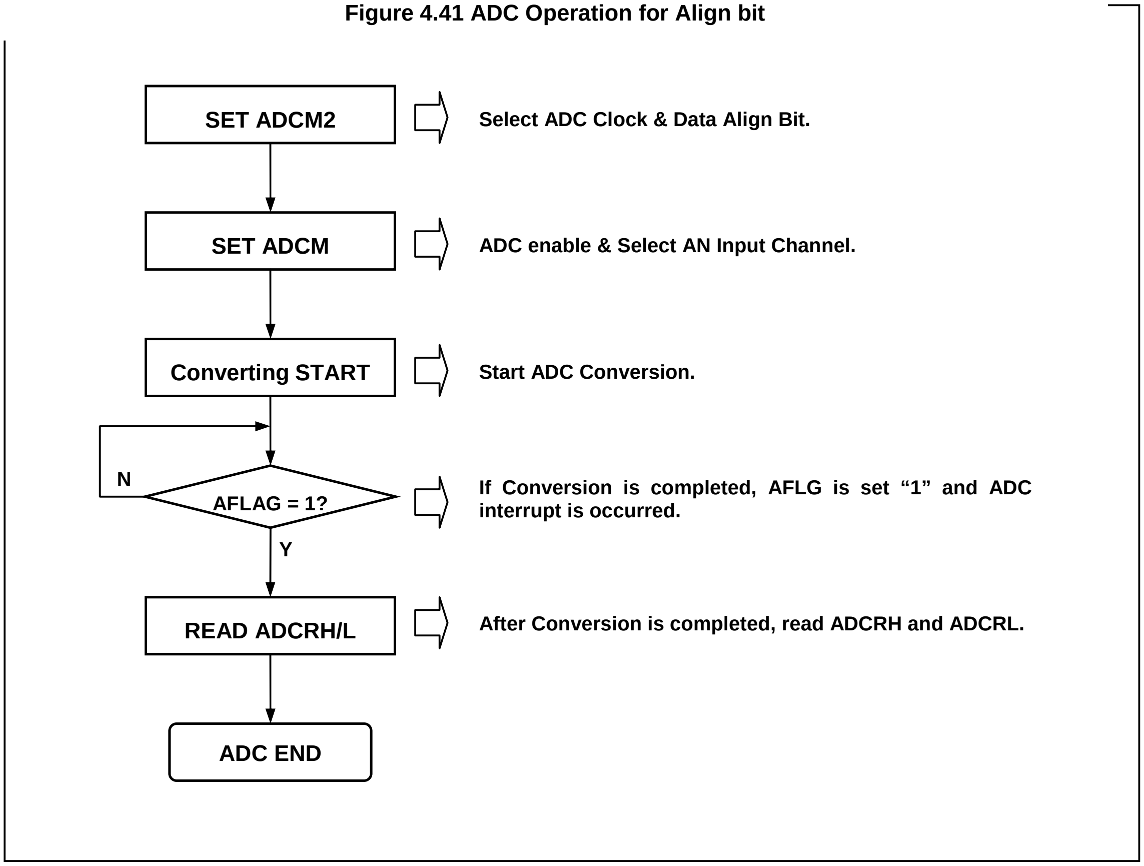

SET ADCM2
Select ADC Clock & Data Align Bit.
SET ADCM
ADC enable & Select AN Input Channel.
Converting START
Start ADC Conversion.
N
AFLAG = 1?
Y
If Conversion is completed, AFLG is set "1" and ADC interrupt is occurred.
READ ADCRH/L
After Conversion is completed, read ADCRH and ADCRL.
ADC END

Figure 4.42 A/D Converter Operation Flow

4.8.4 Register Map

Table 4.16 ADC Register Map

Name	Address	Dir	Default	Description
ADCM	9AH	R/W	8FH	A/D Converter Mode Register
ADCRH	9BH	R	-	A/D Converter Result High Register
ADCRL	9CH	R	-	A/D Converter Result Low Register
ADCM2	9BH	R/W	8FH	A/D Converter Mode 2 Register

4.8.5 ADC Register description

The ADC Register consists of A/D Converter Mode Register (ADCM), A/D Converter Result High Register (ADCRH), A/D Converter Result Low Register (ADCRL), A/D Converter Mode 2 Register (ADCM2).

Note) when STBY bit is set to ‘1’, ADCM2 can be read. If ADC enables, it is possible only to write ADCM2.When reading, ADCRH is read.

4.8.6 Register description for ADC

ADCM (A/D Converter Mode Register) : 9AH

7	6	5	4	3	2	1	0
STBY	ADST	REFSEL	AFLAG	ADSEL3	ADSEL2	ADSEL1	ADSEL0
R/W	R/W	R/W	R/	R/W	R/W	R/W	R/W

Initial value : 8FH

STBY Control operation of A/D standby (power down)
0 ADC module enable
1 ADC module disable (power down)

ADST Control A/D Conversion stop/start.
0 ADC Conversion Stop
1 ADC Conversion Start

REFSEL A/D Converter reference selection
0 Internal Reference (VDD)
1 External Reference(AVREF, AN0 disable)

AFLAG A/D Converter operation state
0 During A/D Conversion
1 A/D Conversion finished

ADSEL[3:0] A/D Converter input selection

ADSEL3	ADSEL2	ADSEL1	ADSEL0	Description
0	0	0	0	Channel0(AN0)
0	0	0	1	Channel1(AN1)
0	0	1	0	Channel2(AN2)
0	0	1	1	Channel3(AN3)
0	1	0	0	Channel4(AN4)
0	1	0	1	Channel5(AN5)
0	1	1	0	Channel6(AN6)
0	1	1	1	Channel7(AN7)
1	0	0	0	Channel8(N/A)
1	0	0	1	Channel9(N/A)

1	0	1	0	Channel10(VDC:1.8V)
1	0	1	1	Channel11(Bandgap Vref:1.17V)
1	1	0	0	Channel12(VSS)
1	1	0	1	Channel13(Ext. VDD)
1	1	1	0	Channel14(N/A)
1	1	1	1	Channel15(N/A)

ADCRH (A/D Converter Result High Register) : 9BH

7	6	5	4	3	2	1	0
ADDM11	ADDM10	ADDM9	ADDM8	ADDM7 ADDL11	ADDM6 ADDL10	ADDM5 ADDL9	ADDM4 ADDL8
R	R	R	R	R	R	R	R

Initial value : xxH

ADDM[11:4] MSB align, A/D Converter High result (8-bit)

ADDL[11:8] LSB align, A/D Converter High result (4-bit)

ADCRL (A/D Converter Result Low Register) : 9CH

7	6	5	4	3	2	1	0
ADDM3 ADDL7	ADDM2 ADDL6	ADDM1 ADDL5	ADDM0 ADDL4	ADDL3	ADDL2	ADDL1	ADDL0
R	R	R	R	R-	R	R	R

Initial value : xxH

ADDM[3:0] MSB align, A/D Converter Low result (4-bit)

ADDL[7:0] LSB align, A/D Converter Low result (8-bit)

ADCM2 (A/D Converter Mode Register) : 9BH

7	6	5	4	3	2	1	0
EXTRG	TSEL2	TSEL1	TSEL0	AMUXEN	ALIGN	CKSEL1	CKSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 8FH

EXTRG A/D external Trigger

0 External Trigger disable

1 External Trigger enable

TSEL[2:0] A/D Trigger Source selection

TSEL2	TSEL1	TSEL0	Description
0	0	0	Ext. Interrupt 0
0	0	1	Analog Comparator Low to High
0	1	0	Analog Comparator High to Low
0	1	1	Timer0 interrupt
1	0	0	Timer1 interrupt
1	0	1	Reserved
1	1	0	Reserved
1	**1**	**1**	Timer4 Interrupt

AMUXEN Control A/D Converter MUX output

0 A/D Converter MUX output disable

1 When STBY=1, A/D Converter MUX output enable

ALIGN A/D Converter data align selection.

0	MSB align (ADCRH[7:0], ADCRL[7:4])
1	LSB align (ADCRH[3:0], ADCRL[7:0])

CKSEL[1:0] A/D Converter Clock selection

CKSEL1	CKSEL0	ADC Clock	ADC VDD
0	0	fx/2	Test Only
0	1	fx/4	3V~5V
1	0	fx/8	2.7V~3V
1	1	fx/32	2.4V~2.7V

Note) 1. fx : system clock

2. ADC clock have to be used 3MHz under.

PSR0 (ADC Pin Selection Register) : 9FH

7	6	5	4	3	2	1	0
AIN07_EN	AIN06_EN	AIN05_EN	AIN04_EN	AIN03_EN	AIN02_EN	AIN01_EN	AIN00_EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

PSR0[7:0] ADC Input Pin Channel Selection (Disable logic input gate)

0	disable
1	enable

4.9 Analog Comparator

4.9.1 Overview

The Analog Comparator compares the input values on the positive pin AN5 and the negative pin AN4. When the voltage on the positive pin AN5 is higher than the voltage on the negative pin AN4, the Analog Comparator output, ACOUT, is set.

4.9.2 Block Diagram

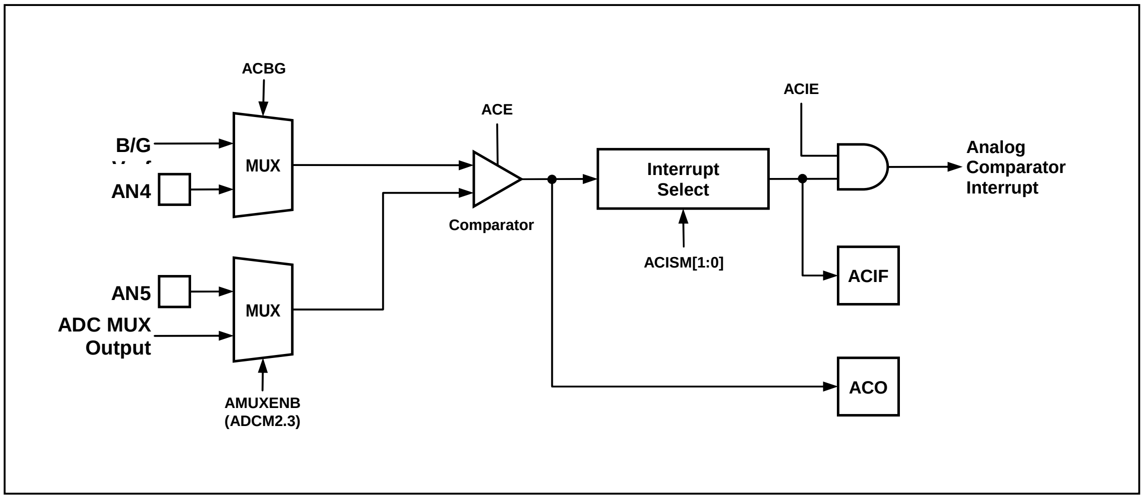


Figure 4.43 Analog Comparator Block Diagram

4.9.3 IN/OUT signal description

ACE : This enables Analog Comparator. When ACE is ‘0’, the output of Comparator goes LOW.

BGR : Band Gap Reference Voltage

ACBG : This selects (-) input source between BGR and AN4. When ACBG is ‘1’, the (-) input to AC is BGR.

AN4 : This can be (-) input to the AC, and comes directly from external analog pad.

AN5 : This can be (+) input to the AC, and comes directly from external analog pad.

AMUXENB : This selects (+) input source between multiplexed output of ADC and AN5. AMUXENB is the inverted signal of AMUXEN bit in ADCM2 register. When AMUXENB is ‘0’, the (+) input to AC comes from ADC module which is selected by ADSEL[3:0], the channel selection bits in ADCM register.

ACOUT : This is the output of Comparator.

4.9.4 Register Map

Table 4.17 Analog Comparator Register Map

Name	Address	Dir	Default	Description
ACCSR	E9H	R/W	00H	Analog Comparator Control & Status Register

4.9.5 Analog Comparator Register description

Analog Comparator Register has one control register, Analog Comparator Control & Status Register (ACCSR). Note that AMUXENB is the inverted signal of AMUXEN bit which comes from ADC's ADCM2 register

4.9.6 Register description for Analog Comparator

ACCSR (Analog Comparator Control & Status Register) : E9H

7	6	5	4	3	2	1	0
ACE	ACBG	ACO	ACIF	ACIE	-	ACISM1	ACISM0
R/W	R/W	R	R	R/W	-	R/W	R/W

Initial value : 00H

ACE Enable Analog Comparator (AC).
0 Disable AC (power down)
1 Enable AC

ACBG Select (-) input source of AC, Band Gap Reference Voltage or AN4.
0 (-) input is from AN4
1 (-) input is from Band Gap Reference Voltage

ACO This bit represents the value of ACOUT (Output of Analog Comparator). ACO bit is sampled by SCLK, system clock, twice. When ACE is '0', this bit is also cleared.
0 Comparator output is LOW
1 Comparator output is HIGH

ACIF This bit is set when an Analog Comparator Interrupt is generated according to the ACISM[1:0] bits. This bit is cleared when Analog Comparator Interrupt is executed or '0' is written to this bit field.
0 No interrupt generated or cleared
1 Interrupt generated

ACIE Enable Analog Comparator Interrupt.
0 Disable Interrupt, Polling mode operation
1 Enable Interrupt

ACISM[1:0] Select Interrupt Mode of Analog Comparator.

ACISM1	ACISM0	Description
0	0	Reserved
0	1	Interrupt on falling edge of ACOUT
1	0	Interrupt on rising edge of ACOUT
1	1	Interrupt on both edge of ACOUT

PSR1 (Comparator Pin Selection Register) : A0H

7	6	5	4	3	2	1	0
							ACO_EN

							R/W

Initial value : 00H

PSR1[0] Analog Comparator Output Enable (Disable logic input gate)

0	disable
1	enable

4.10 Buzzer Driver

4.10.1 Overview

The Buzzer consists of 8 Bit Counter and BUZDR (Buzzer Data Register), BUZCR (Buzzer Control Register). The Square Wave (61.035Hz~125 KHz, @8MHz) gets out of P12/BUZ pin. BUZDR (Buzzer Data Register) controls the Buzzer frequency (look at the following expression). In the BUZCR (Buzzer Control Register), BUCK[1:0] selects source clock divided from prescaler.

$$f_{BUZ}(Hz) = \frac{\text{Oscillator Frequency}}{2 \times \text{Prescaler Ratio} \times (\text{BUZDR} + 1)}$$

BUZDR[7:0]	Buzzer Frequency (kHz)			
	BUZCR[2:1]=00	BUZCR[2:1]=01	BUZCR[2:1]=10	BUZCR[2:1]=11
0000_0000	125kHz	62.5kHz	31.25kHz	15.625kHz
0000_0001	62.5kHz	31.25kHz	15.625kHz	7.812kHz
…	…	…	…	…
1111_1101	492.126Hz	246.063Hz	123.031Hz	61.515Hz
1111_1110	490.196Hz	245.098Hz	122.549Hz	61.274Hz
1111_1111	488.281Hz	244.141Hz	122.07Hz	61.035Hz

Table 11-12 Buzzer Frequency at 8 Mhz

4.10.2 Block Diagram

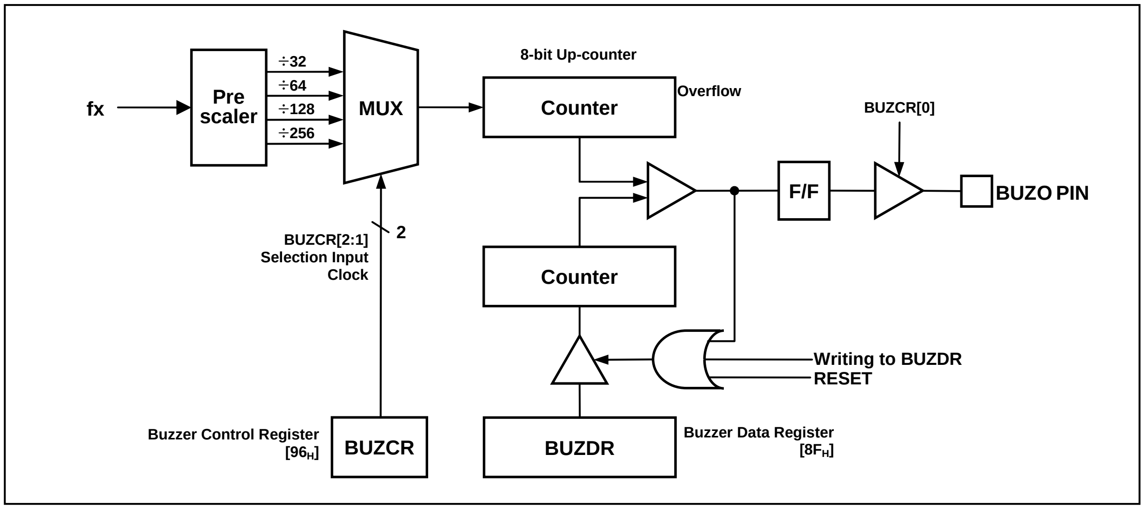


Figure 4-38 Buzzer Driver Block Diagram

4.10.3 Register Map

Name	Address	Dir	Default	Description
BUZDR	8FH	R/W	FFH	Buzzer Data Register
BUZCR	96H	R/W	00H	Buzzer Control Register

Table 11-13 Register Map

4.10.4 Buzzer Driver Register description

Buzzer Driver consists of Buzzer Data Register (BUZDR), Buzzer Control Register (BUZCR).

4.10.5 Register description for Buzzer Driver

BUZDR (Buzzer Data Register) : 8FH

7	6	5	4	3	2	1	0
BUZDR7	BUZDR6	BUZDR5	BUZDR4	BUZDR3	BUZDR2	BUZDR1	BUZDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

BUZDR[7:0] This bits control the Buzzer frequency
Its resolution is 00H ~ FFH

BUZCR (Buzzer Control Register) : 96H

7	6	5	4	3	2	1	0
-	-	-	-	-	BUCK1	BUCK0	BUZEN
-	-	-	-	-	R/W	R/W	R/W

Initial value : 00H

BUCK[1:0] Buzzer Driver Source Clock Selection

BUCK1	BUCK0	Source Clock
0	0	fx/32
0	1	fx/64
1	0	fx/128
1	1	fx/256

BUZEN Buzzer Driver Operation Control

0	Buzzer Driver disable
1	Buzzer Driver enable

Note) fx: Main system clock oscillation frequency

5. Power Down Operation

5.1 Overview

The Z51F0410 MCU features three power-down modes to minimize the power consumption of the device. In power down mode, power consumption is reduced considerably. The device provides three kinds of power saving functions, IDLE, STOP1 and STOP2 mode. In three modes, program is stopped.

5.2 Peripheral Operation In IDLE/STOP Mode

Table 5.1 Peripheral Operation during Power Down Mode.

Peripheral	IDLE Mode	STOP1 Mode	STOP2 Mode
CPU	ALL CPU Operation are Disable	ALL CPU Operation are Disable	ALL CPU Operation are Disable
RAM	Retain	Retain	Retain
Basic Interval Timer	Operates Continuously	Operates Continuously	Stop
Watch Dog Timer	Operates Continuously	Operates Continuously	Stop
Watch Timer	Operates Continuously	Stop (Only operate in sub clock mode)	Stop (Only operate in sub clock mode)
TimerP0~1	Operates Continuously	Halted (Only when the Event Counter Mode is Enable, Timer operates Normally)	Halted (Only when the Event Counter Mode is Enable, Timer operates Normally)
ADC	Operates Continuously	Stop	Stop
BUZ	Operates Continuously	Stop	Stop
SPI/SCI	Operates Continuously	Only operate with external clock	Only operate with external clock
I2C	Operates Continuously	Stop	Stop
Internal OSC (8MHz)	Oscillation	Stop	Stop
Main OSC (1~8MHz)	Oscillation	Stop	Stop
Sub OSC (32.768kHz)	Oscillation	Oscillation	Oscillation
Internal RCOSC (128kHz)	Oscillation	Oscillation	Stop
I/O Port	Retain	Retain	Retain
Control Register	Retain	Retain	Retain
Address Data Bus	Retain	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, Timer Interrupt (EC0), SIO (External clock), External Interrupt, UART by ACK PCI, I2C (slave mode), WT (sub clock),WDT, BIT	By RESET, Timer Interrupt (EC0), SIO (External clock), External Interrupt, UART by ACK PCI, I2C (slave mode), WT (sub clock)

5.3 IDLE mode

The power control register is set to ‘01h’ to enter the IDLE Mode. In this mode, the internal oscillation circuits remain active. Oscillation continues and peripherals are operated normally but CPU stops. It is released by reset or interrupt. To be released by interrupt, interrupt should be enabled before IDLE mode. If using reset, because the device becomes initialized state, the registers have reset value.

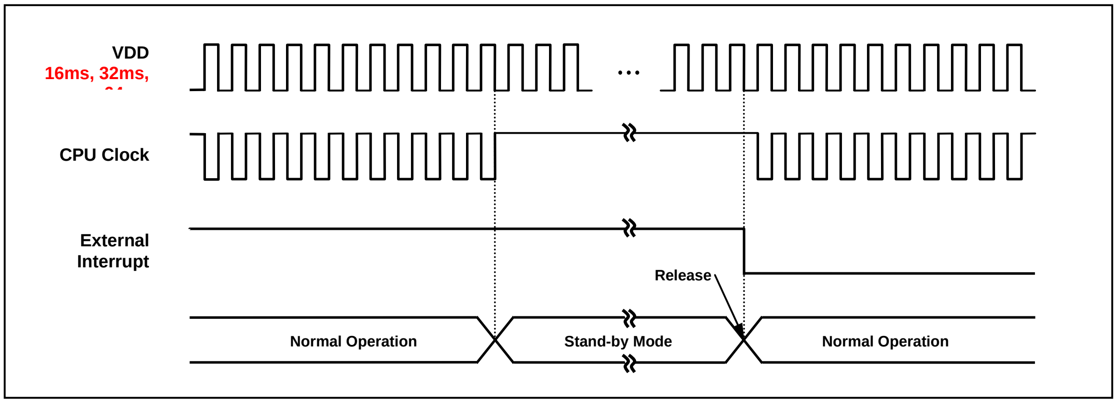


Figure 5.1 IDLE Mode Release Timing by External Interrupt

5.4 STOP mode

The power control register is set to ‘03h’ to enter the STOP Mode. In the stop mode, the main oscillator, system clock and peripheral clock is stopped, but watch timer continue to operate. With the clock frozen, all functions are stopped, but the on-chip RAM and control registers are held.

The source for exit from STOP mode is hardware reset and interrupts. The reset re-defines all the control registers.

When exit from STOP mode, enough oscillation stabilization time is required to normal operation. Figure 5.2 shows the timing diagram. When released from STOP mode, the Basic interval timer is activated on wake-up. Therefore, before STOP instruction, user must be set its relevant prescale divide ratio to have long enough time. this guarantees that oscillator has started and stabilized.

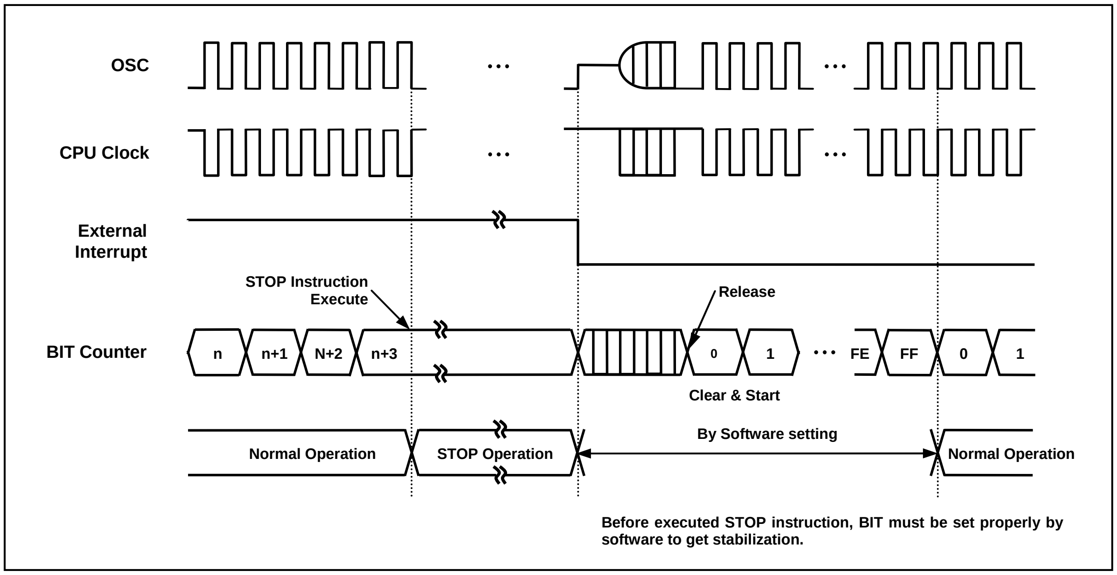


Figure 5.2 STOP Mode Release Timing by External Interrupt

5.5 Release Operation of STOP1, 2 Mode

After STOP1, 2 mode is released, the operation begins according to content of related interrupt register just before STOP1, 2 mode start (Figure 5.3). Interrupt Enable Flag of All (EA) of IE should be set to `1`. Released by only interrupt which each interrupt enable flag = `1`, and jump to the relevant interrupt service routine.

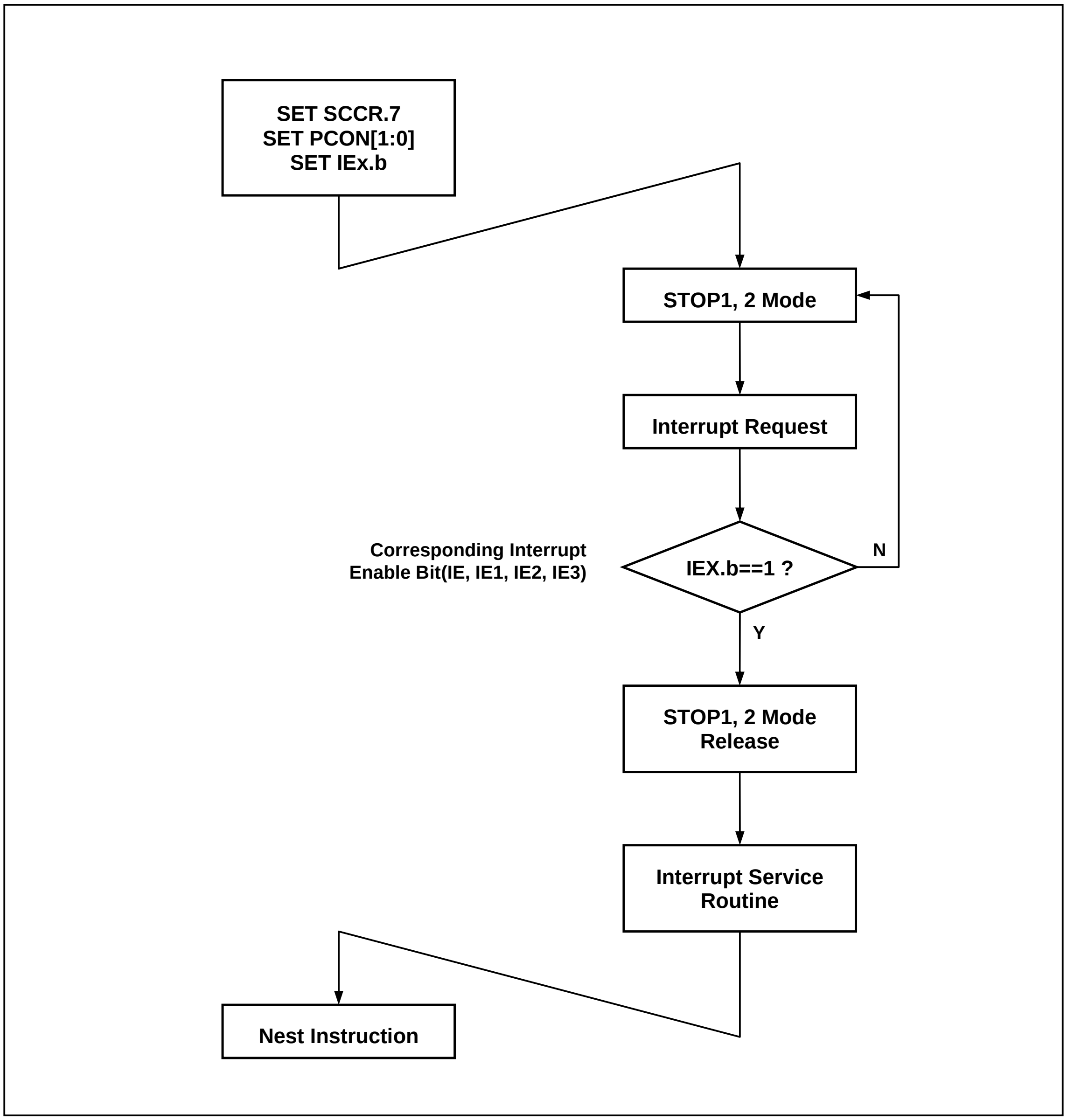


Figure 5.3 STOP1, 2 Mode Release Flow

5.5.1 Register Map

Table 5.2 PCON Register Map

Name	Address	Dir	Default	Description

PCON	87H	R/W	00H	Power Control Register

5.5.2 Power Down Operation Register description

The Power Down Operation Register consists of the Power Control Register (PCON).

5.5.3 Register description for Power Down Operation

PCON (Power Control Register) : 87H

7	6	5	4	3	2	1	0
BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IDLE Mode

01H	IDLE mode enable

STOP1, 2 Mode

03H	STOP1, 2 mode enable

Note)

1. To enter IDLE mode, PCON must be set to ‘01H’.
2. To STOP1,2 mode, PCON must be set to ‘03H’.
   (In STOP1,2 mode, PCON register is cleared automatically by interrupt or reset)
3. When PCON is set to ‘03H’, if SCCR[7] is set to ‘1’, it enters the STOP1 mode. if SCCR[7] is cleared to ‘0’, it enters the STOP2 mode
4. The different thing in STOP 1,2 is only clock operation of internal 128kHz-OSC during STOP mode operating.

6. RESET

6.1 Overview

The Z51F0410 MCU features reset by external RESETB pin. The following is the hardware setting value.

Table 6.1 Reset state

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Peripheral Registers refer
Brown-Out Detector	Enable

6.2 Reset source

The Z51F0410 MCU features five types of reset generation procedures. The following is the reset sources.

- External RESETB
- Power ON RESET (POR)
- WDT Overflow Reset (In the case of WDTEN = `1`)
- BOD Reset (In the case of BODEN = `1 `)
- OCD Reset

6.3 Block Diagram

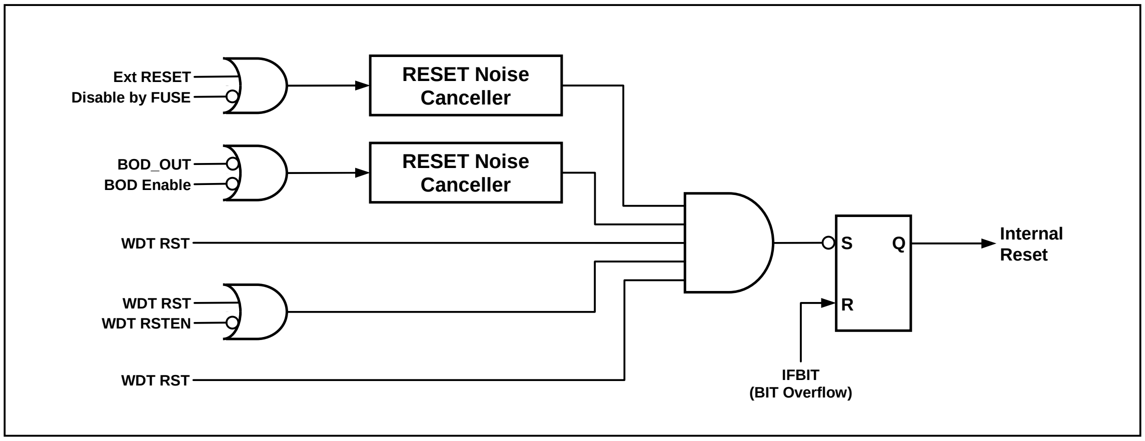


Figure 6.1 RESET Block Diagram

6.4 RESET Noise Canceller

The Figure 6.2 is the noise canceller diagram for noise cancel of RESET. It has the noise cancel value of about 7us (@V_{DD}=5V) to the low input of System Reset.

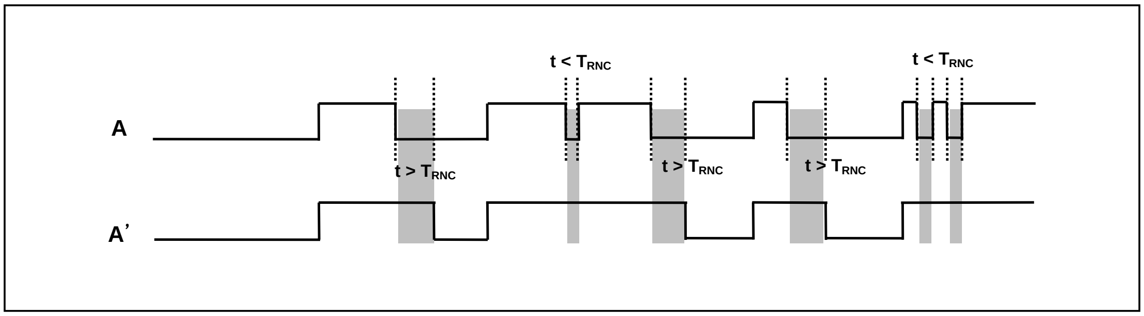


Figure 6.2 Reset noise canceller time diagram

6.5 Power ON RESET

When rising device power, the POR (Power ON Reset) have a function to reset the device. If using POR, it executes the device RESET function instead of the RESET IC or the RESET circuits. And External RESET PIN is able to be used as Normal I/O pin.

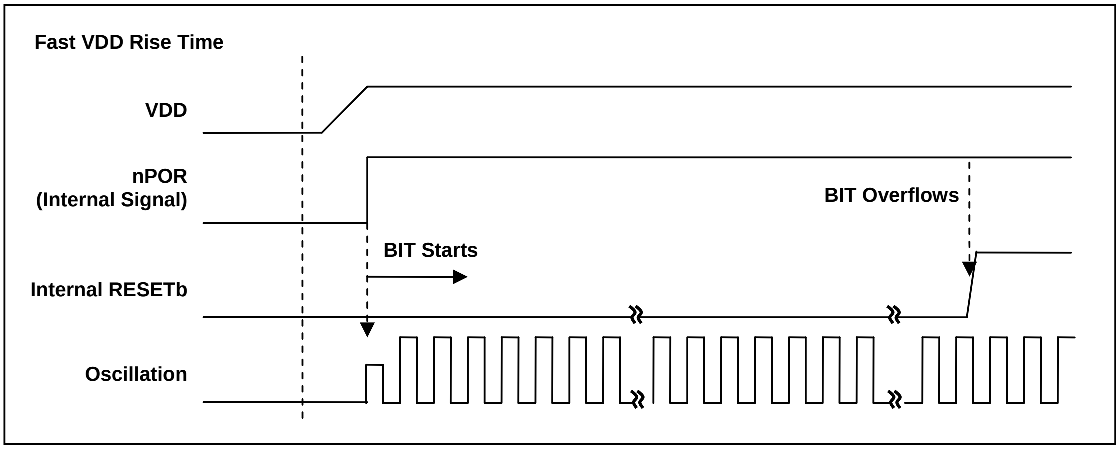


Figure 6.3 Fast VDD rising time

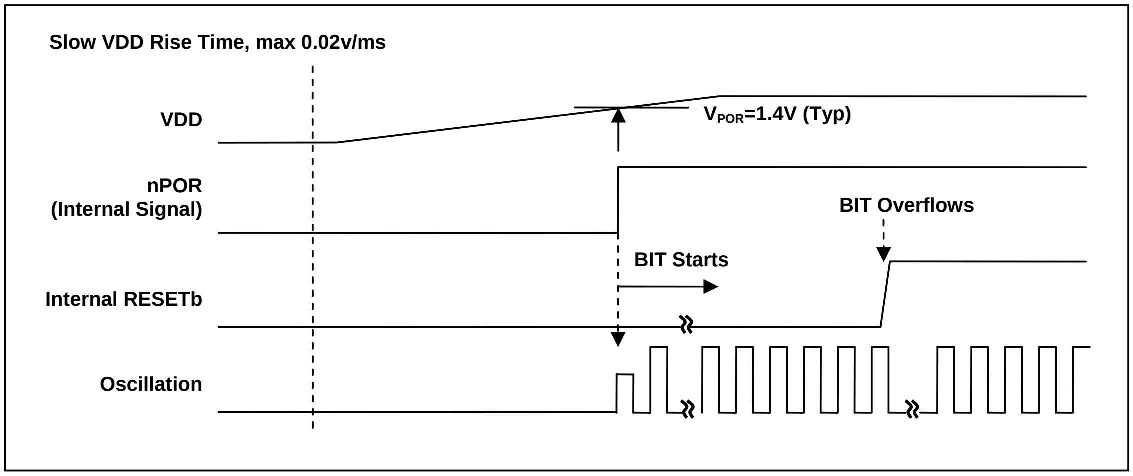
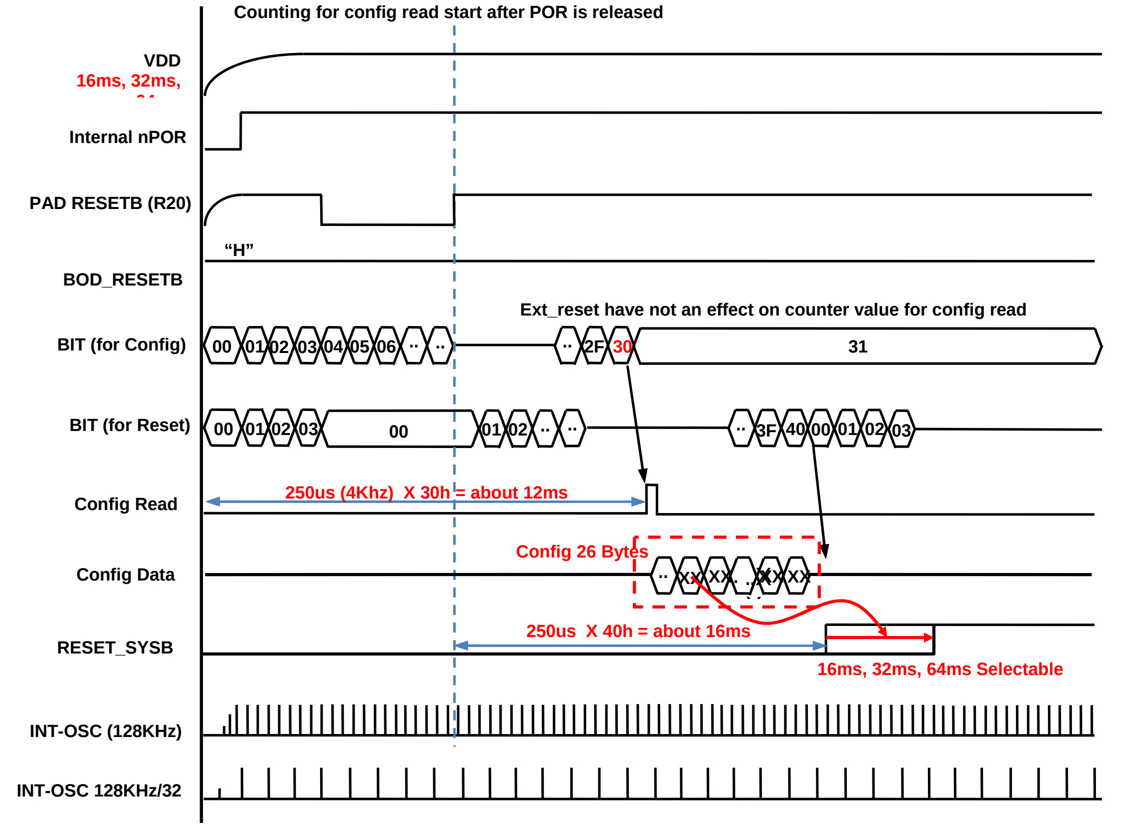


Figure 6.4 Internal RESET Release Timing On Power-Up



INT-OSC 128KHz / 32 = 4KHz (250us)

Figure 6.5 Configuration timing when Power-on

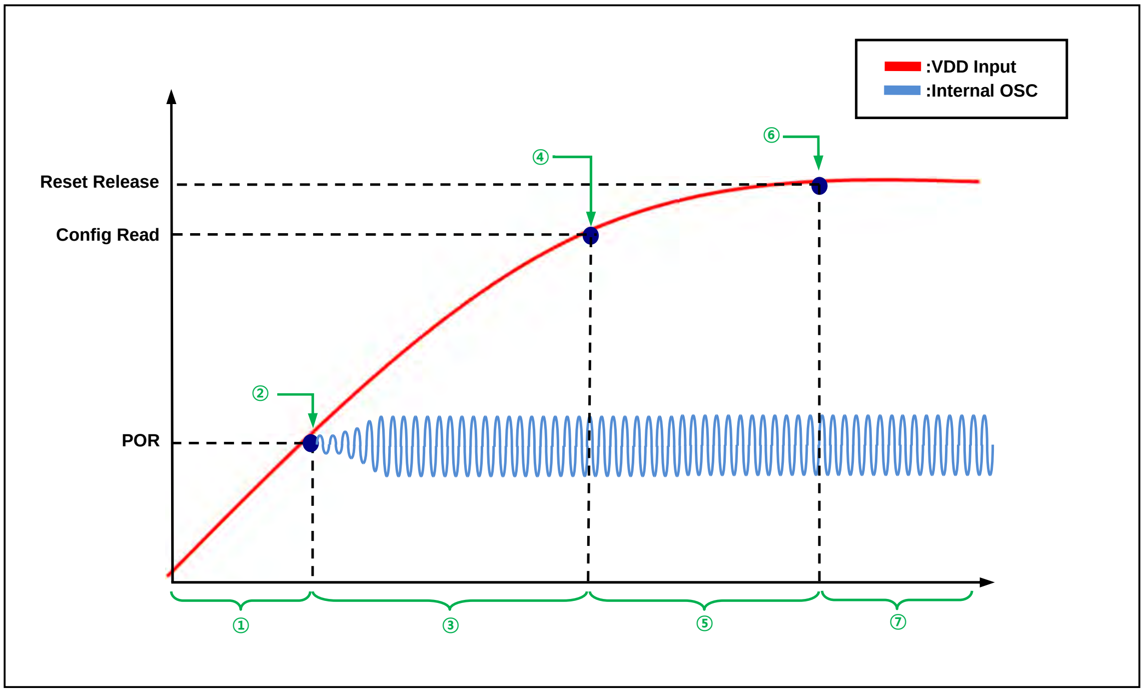


Figure 6.6 Boot Process Wave Form

Table 6.2 Boot Process Description

Process	Description	Remarks
①	-No Operation	
②	-1st POR level Detection -Internal OSC (128KHz) ON	-about 1.4V ~ 1.5V
③	- (INT-OSC128KHz/32)×30h Delay section (=12ms) -VDD input voltage must rise over than flash operating voltage for Config read	-Slew Rate >= 0.025V/ms
④	- Config read point	-about 1.5V ~ 1.6V -Config Value is determined by Writing Option
⑤	- Rising section to Reset Release Level	-16ms point after POR or Ext_reset release
⑥	- Reset Release section (BIT overflow) i) after16ms, after External Reset Release (External reset) ii) 16ms point after POR (POR only)	- BIT is used for Peripheral stability
⑦	-Normal operation	

6.6 External RESETB Input

The External RESETB is the input to a Schmitt trigger. A reset in accomplished by holding the reset pin low for at least 7us over, within the operating voltage range and oscillation stable, it is applied, and the internal state is initialized. After reset state becomes '1', it needs the stabilization time with 16ms and after the stable state, the internal RESET becomes ‘1’. The Reset process step needs 5 oscillator clocks. And the program execution starts at the vector address stored at address 0000H.

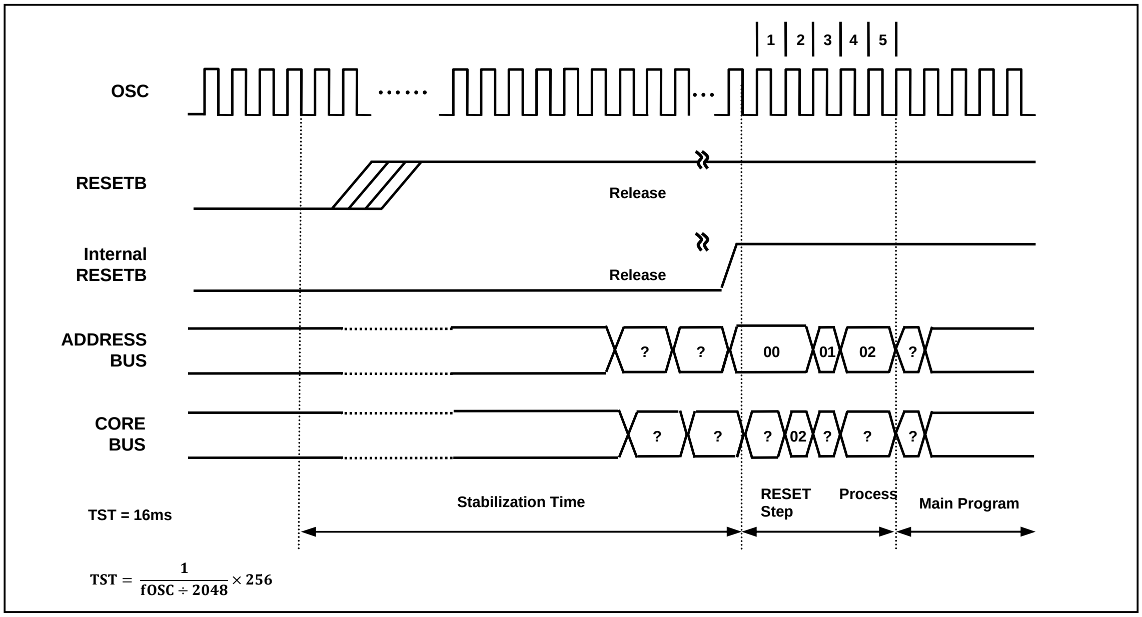


Figure 6.7 Timing Diagram after RESET

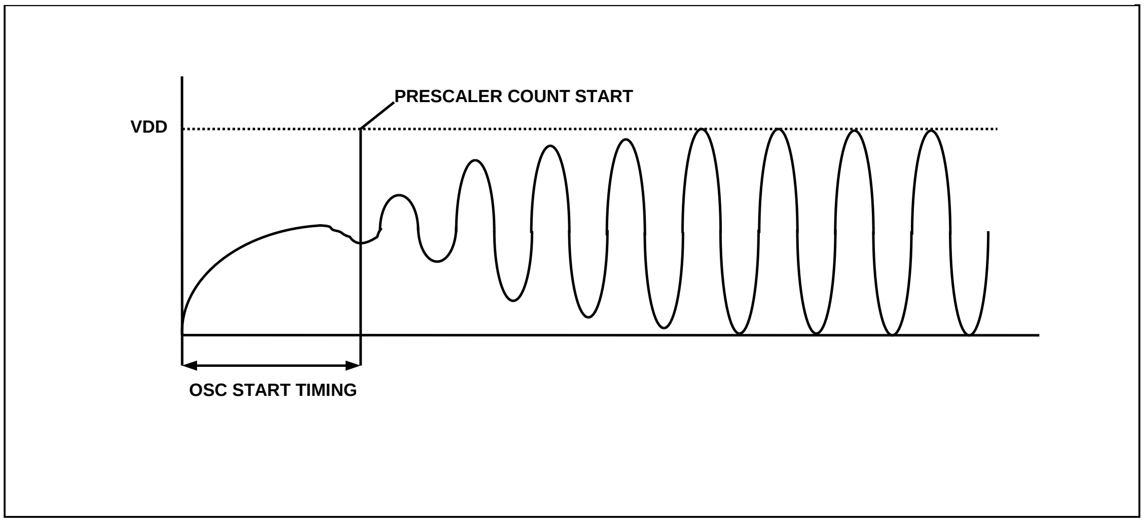


Figure 6.8 Oscillator generating waveform example

6.7 Brown Out Detector Processor

The Z51F0410 MCU features an On-chip Brown-out detection circuit for monitoring the VDD level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by BODLS[1:0] bit to be 1.6V, 2.5V, 3.6V or 4.3V. In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, the BODEN bit is set to off by software.

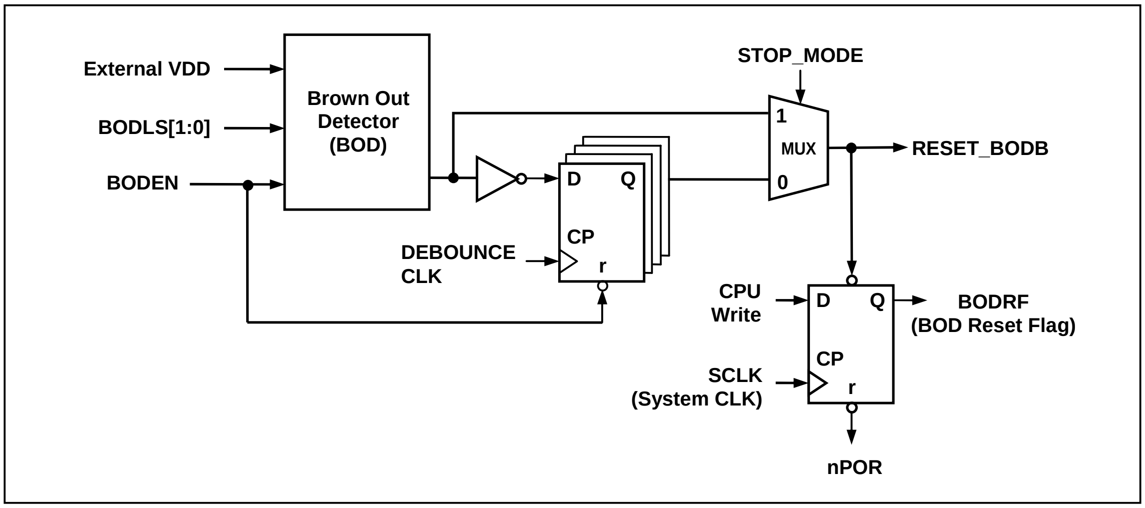


Figure 6.9 Block Diagram of BOD

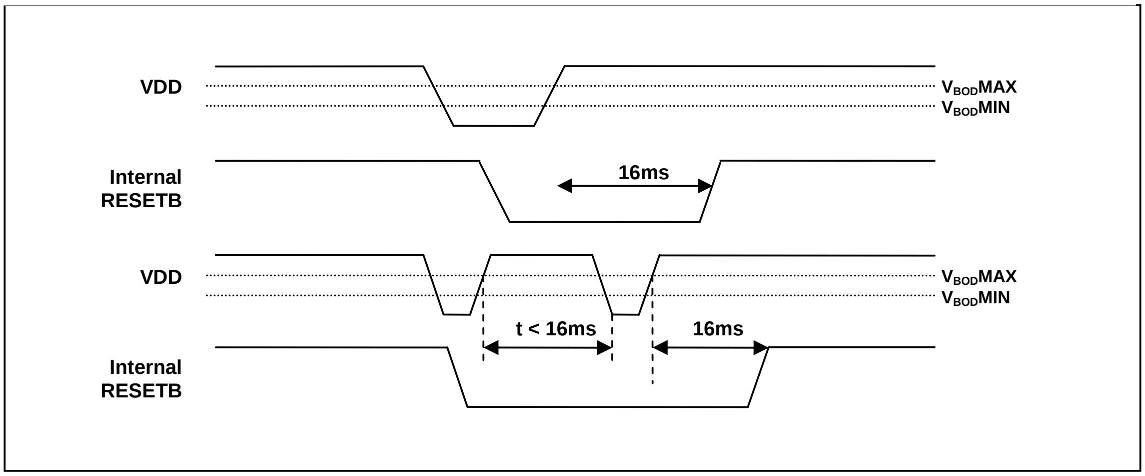


Figure 6.10 Internal Reset at the power fail situation

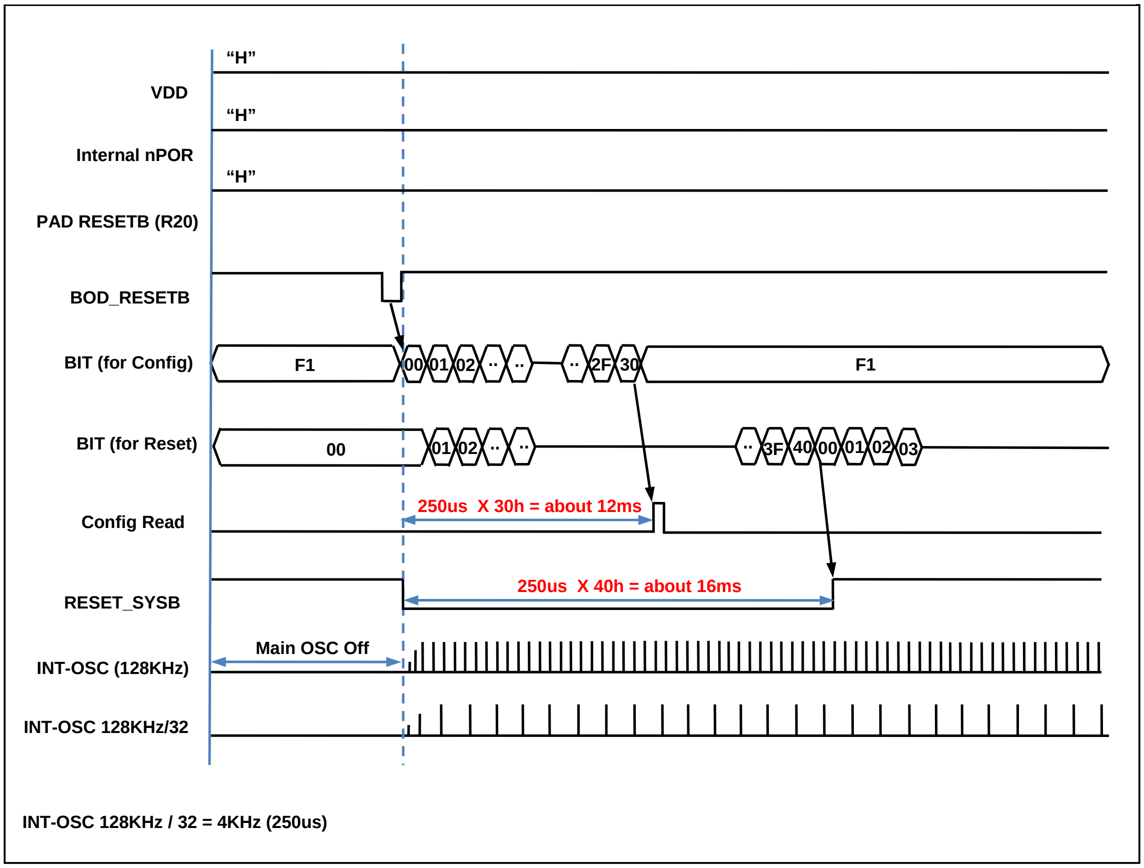


Figure 6.11 Configuration timing when BOD RESET

6.7.1 Register Map

Table 6.3 BOD Register Map

Name	Address	Dir	Default	Description
BODR	0x86	R/W	81H	BOD Control Register

6.7.2 Reset Operation Register description

Reset control Register consists of the BOD Control Register (BODR).

6.7.3 Register description for Reset Operation

BODR (BOD Control Register) : 81H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	OCDRF	BODRF	BODLS[1]	BODLS[0]	BODEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 81H

PORF Power-On Reset flag bit. The bit is reset by writing '0' to this bit.

0 No detection

1 Detection

EXTRF External Reset flag bit. The bit is reset by writing ‘0’ to this bit or by Power ON reset.

0 No detection

1 Detection

WDTRF Watch Dog Reset flag bit. The bit is reset by writing ‘0’ to this bit or by Power ON reset.

0 No detection

1 Detection

OCDRF On-Chip Debug Reset flag bit. The bit is reset by writing ‘0’ to this bit or by Power ON reset.

0 No detection

1 Detection

BODRF Brown-Out Reset flag bit. The bit is reset by writing ‘0’ to this bit or by Power ON reset.

0 No detection

1 Detection

BODLS[1:0] BOD level Voltage

BODLS1	BODLS0	Description
0	0	1.6V
0	1	2.5V
1	0	3.6V
1	1	4.3V

BODEN BOD operation

0 BOD disable

1 BOD enable

7. On-chip Debug System

7.1 Overview

7.1.1 Description

On-chip debug System (OCD) of the Z51F0410 MCU can be used for programming the non-volatile memories and on-chip debugging. Detailed descriptions for programming via the OCD interface can be found in the following chapter.

Figure 7.1 shows a block diagram of the OCD interface and the On-chip Debug system.

7.1.2 Feature

- Two-wire external interface: 1-wire serial clock input, 1-wire bi-directional serial data bus
- Debugger Access to:
  - All Internal Peripheral Units
  - Internal data RAM
  - Program Counter
  - Flash and Data EEPROM Memories
- Extensive On-chip Debug Support for Break Conditions, Including
  - Break Instruction
  - Single Step Break
  - Program Memory Break Points on Single Address
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the two-wire Interface
  - On-chip Debugging Supported by Dr.Choice®
- Operating frequency

  Supports the maximum frequency of the target MCU

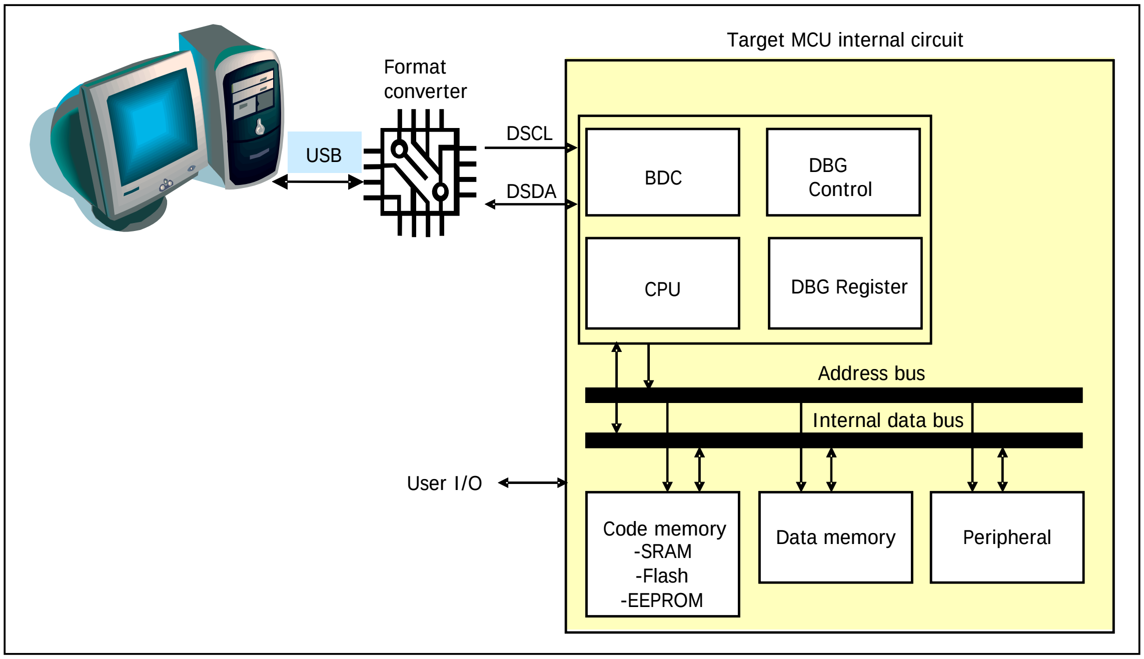


Figure 7.1 Block Diagram of On-chip Debug System

7.2 Two-pin external interface

7.2.1 Basic transmission packet

- 10-bit packet transmission using two-pin interface.
- 1-packet consists of 8-bit data, 1-bit parity and 1-bit acknowledge.
- Parity is even of '1' for 8-bit data in transmitter.
- Receiver generates acknowledge bit as '0' when transmission for 8-bit data and its parity has no error.
- When transmitter has no acknowledge (Acknowledge bit is '1' at tenth clock), error process is executed in transmitter.
- When acknowledge error is generated, host PC makes stop condition and transmits command which has error again.
- Background debugger command is composed of a bundle of packet.
- Star condition and stop condition notify the start and the stop of background debugger command respectively.

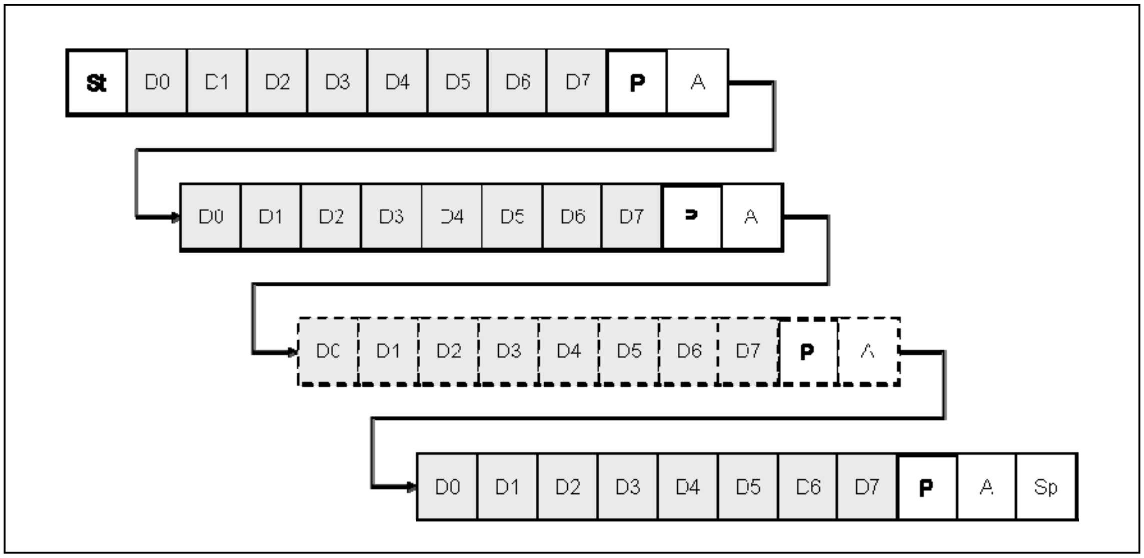


Figure 7.2 10-bit transmission packet

7.2.2 Packet transmission timing

7.2.2.1 Data transfer

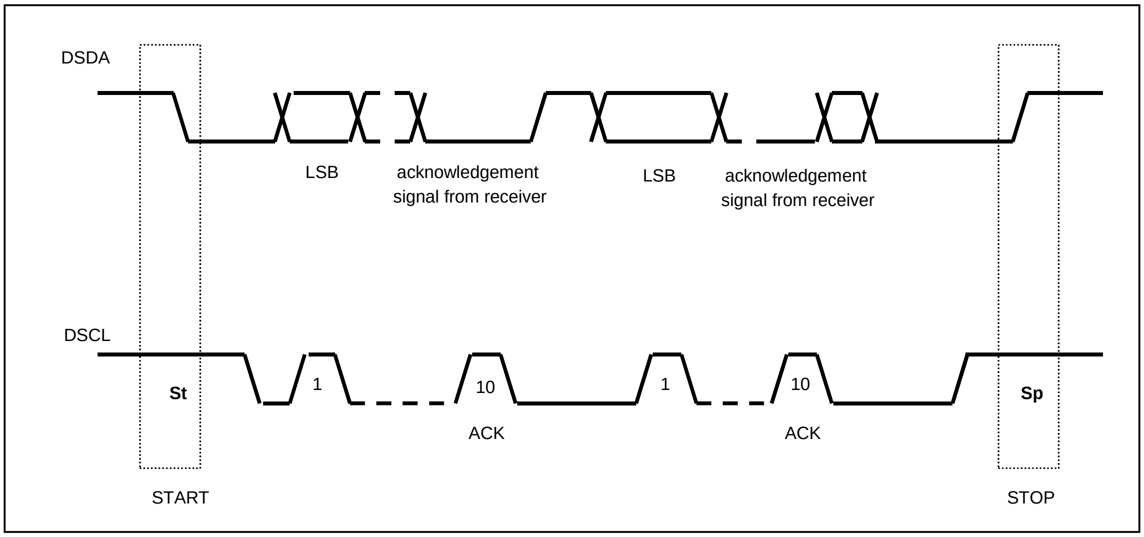

DSDA
LSB
acknowledgement signal from receiver
LSB
acknowledgement signal from receiver
DSCL
St
1
10
ACK
1
10
ACK
Sp
START
STOP

Figure 7.3 Data transfer on the twin bus

7.2.2.2 Bit transfer

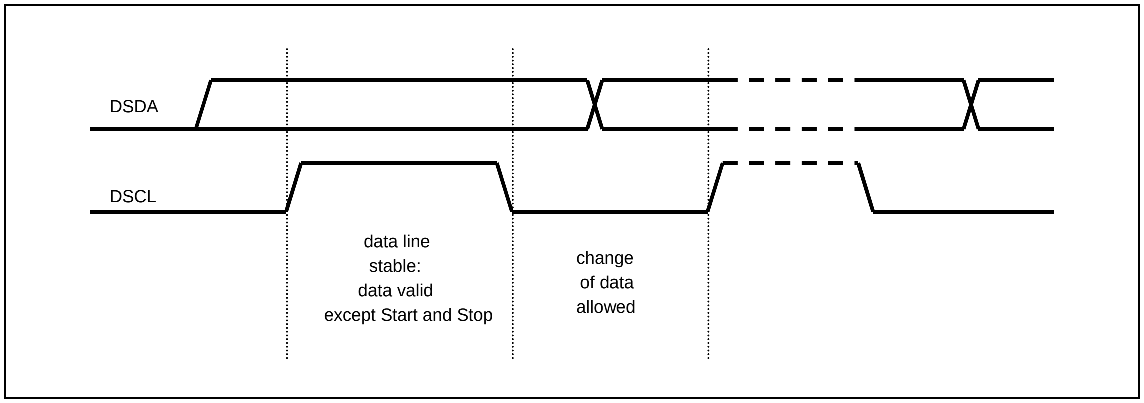


Figure 7.4 Bit transfer on the serial bus

7.2.2.3 Start and stop condition

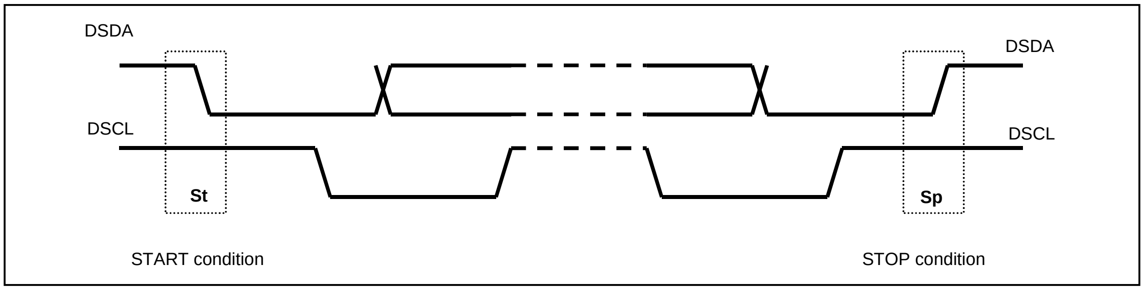


Figure 7.5 Start and stop condition

7.2.2.4 Acknowledge bit

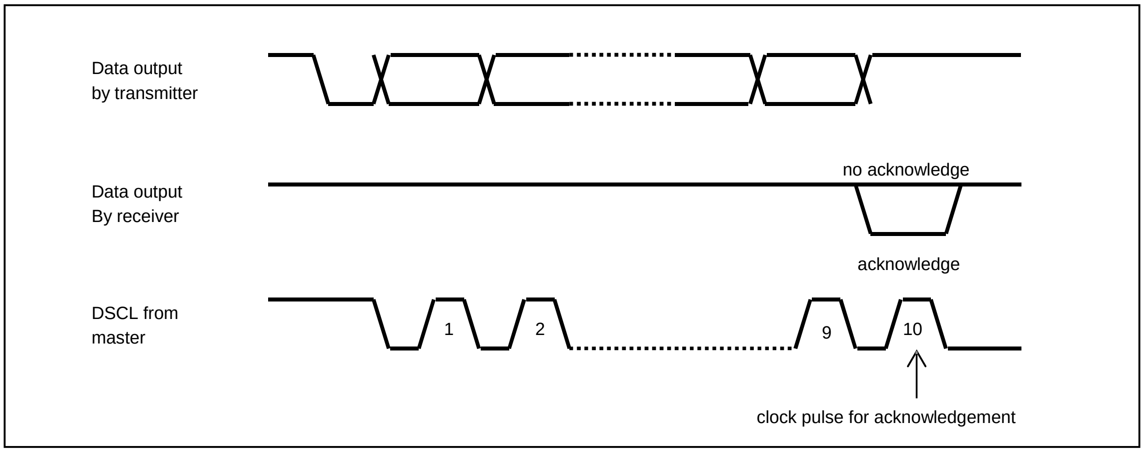


Figure 7.6 Acknowledge on the serial bus

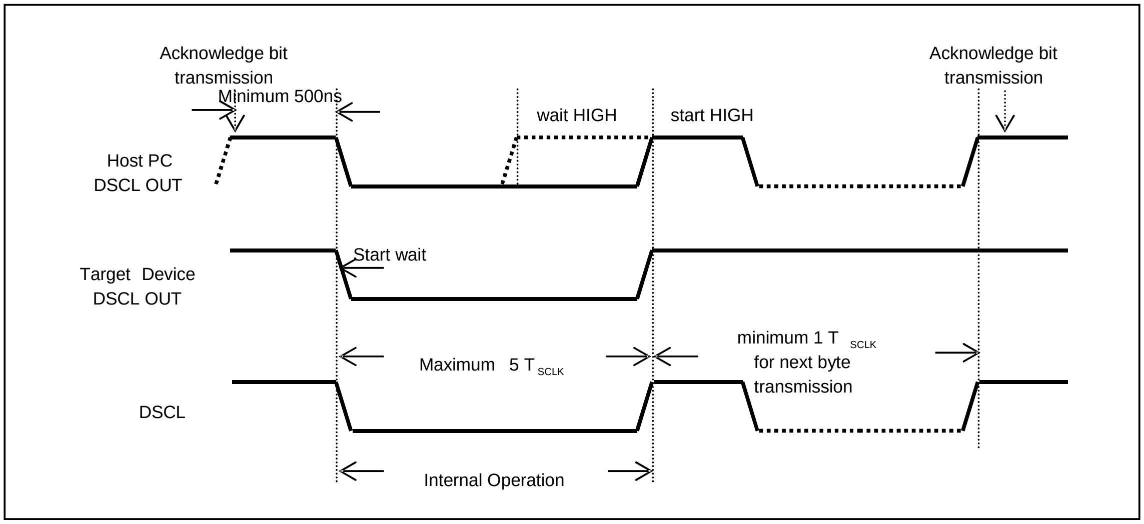


Figure 7.7 Clock synchronization during wait procedure

7.2.3 Connection of transmission

Two-pin interface connection uses open-drain (wire-AND bidirectional I/O).

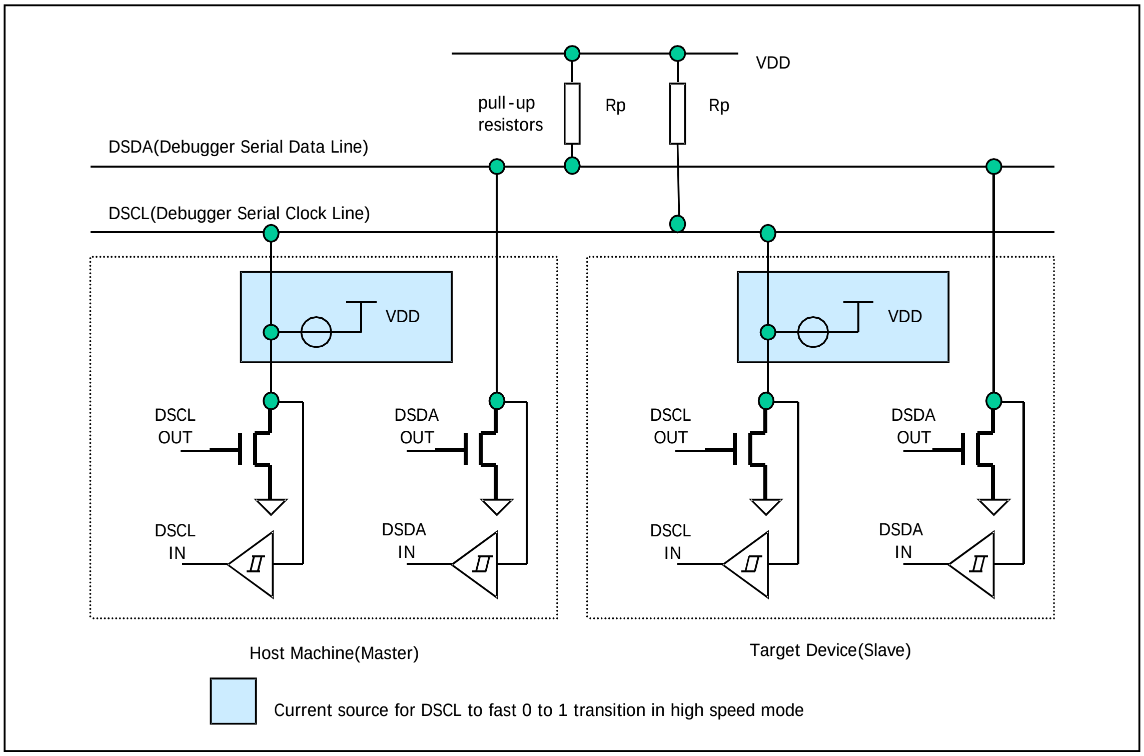

VDD
pull-up
resistors
Rp
Rp
DSDA(Debugger Serial Data Line)
DSCL(Debugger Serial Clock Line)
VDD
VDD
DSCL
OUT
DSDA
OUT
DSCL
OUT
DSDA
OUT
DSCL
IN
DSDA
IN
DSCL
IN
DSDA
IN
Host Machine(Master)
Target Device(Slave)
Current source for DSCL to fast 0 to 1 transition in high speed mode

Figure 7.8 Connection of transmission

8. Memory Programming

8.1 Overview

8.1.1 Description

The Z51F0410 MCU incorporates flash and data EEPROM memory to which a program can be written, erased, and overwritten while mounted on the board. Also, data EEPROM can be programmed or erased in user program.

Serial ISP modes and byte-parallel ROM writer mode are supported.

8.1.2 Features

- Flash Size : 4Kbytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 10,000 program/erase cycles at typical voltage and temperature for flash memory
- Up to 100,000 program/erase cycles at typical voltage and temperature for data EEPROM memory
- Security feature

8.2 Flash and EEPROM Control and status register

Registers to control Flash and Data EEPROM are Mode Register (FEMR), Control Register (FECR), Status Register (FESR), Time Control Register (FETCR), Address Low Register (FEARL), Address Middle Register (FEARM), address High Register (FEARH) and Data Register (FEDR). They are mapped to SFR area and can be accessed only in programming mode.

8.2.1 Register Map

Table 8-1 Flash and EEPROM Register Map

Name	Address	Dir	Default	Description
FEMR	EAH	R/W	00H	Flash and EEPROM Mode Register
FECR	EBH	R/W	03H	Flash and EEPROM Control Register
FESR	ECH	R/W	80H	Flash and EEPROM Status Register
FETCR	EDH	R/W	00H	Flash and EEPROM Time Control Register
FEARL	F2H	R/W	00H	Flash and EEPROM Address Low Register
FEARM	F3H	R/W	00H	Flash and EEPROM Address Middle Register
FEARH	F4H	R/W	00H	Flash and EEPROM Address High Register
FEDR	F5H	R/W	00H	Flash and EEPROM Data Register

8.2.2 Register description for Flash and EEPROM

FEMR (Flash and EEPROM Mode Register) : EAH

7	6	5	4	3	2	1	0
FSEL	ESEL	PGM	ERASE	PBUFF	OTPE	VFY	FEEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FSEL Select flash memory.
0 Deselect flash memory
1 Select flash memory

ESEL Select data EEPROM
0 Deselect data EEPROM
1 Select data EEPROM

PGM Enable program or program verify mode with VFY
0 Disable program or program verify mode
1 Enable program or program verify mode

ERASE Enable erase or erase verify mode with VFY
0 Disable erase or erase verify mode
1 Enable erase or erase verify mode

PBUFF Select page buffer
0 Deselect page buffer
1 Select page buffer

OTPE Select OTP area instead of program memory
0 Deselect OTP area
1 Select OTP area

VFY Set program or erase verify mode with PGM or ERASE
Program Verify: PGM=1, VFY=1
Erase Verify: ERASE=1, VFY=1

FEEN Enable program and erase of Flash and data EEPROM. When inactive, it is possible to read as normal mode
0 Disable program and erase
1 Enable program and erase

FECR (Flash and EEPROM Control Register) : EBH

7	6	5	4	3	2	1	0
AEF	AEE	EXIT1	EXIT0	WRITE	READ	nFERST	nPBRST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 03H

AEF Enable flash bulk erase mode
0 Disable bulk erase mode of Flash memory
1 Enable bulk erase mode of Flash memory

AEE Enable data EEPROM bulk erase mode
0 Disable bulk erase mode of data EEPROM
1 Enable bulk erase mode of data EEPROM

EXIT[1:0] Exit from program mode. It is cleared automatically after 1 clock

EXIT1	EXIT0	Description
0	0	Don’t exit from program mode
0	1	Don’t exit from program mode

1	0	Don’t exit from program mode
1	1	Exit from program mode

WRITE Start to program or erase of Flash and data EEPROM. It is cleared automatically after 1 clock

0	No operation
1	Start to program or erase of Flash and data EEPROM

READ Start auto-verify of Flash or data EEPROM. It is cleared automatically after 1 clock

0	No operation
1	Start auto-verify of Flash or data EEPROM

nFERST Reset Flash or data EEPROM control logic. It is cleared automatically after 1 clock

0	No operation
1	Reset Flash or data EEPROM control logic.

nPBRST Reset page buffer with PBUFF. It is cleared automatically after 1 clock

PBUFF	nPBRST	Description
0	0	Page buffer reset
1	0	Write checksum reset

WRITE and READ bits can be used in program, erase and verify mode with FEAR registers. Read or writes for memory cell or page buffer uses read and write enable signals from memory controller. Indirect address mode with FEAR is only allowed to program, erase and verify

FESR (Flash and EEPROM Status Register) : ECH

7	6	5	4	3	2	1	0
PEVBSY	VFYGOOD	PCRCRD	ENCRYPT	ROMINT	WMODE	EMODE	VMODE
R	R/W	R/W	R/W	R/W	R	R	R

Initial value : 80H

PEVBSY Operation status flag. It is cleared automatically when operation starts. Operations are program, erase or verification

0	Busy (Operation processing)
1	Complete Operation

VFYGOOD Auto-verification result flag.

0	Auto-verification fails
1	Auto-verification successes

PCRCRD CRC Calculation Data Read Control

0	FEARH, FEARM, FEARL represents 24bit Checksum
1	FEARM, FEARL represent 16bit CRC result

ENCRYPT Encryption Mode Control for PGM

0	Normal Data PGM Mode
1	Encryption PGM Mode

ROMINT Flash and Data EEPROM interrupt request flag. Auto-cleared when program/erase/verify starts. Active in program/erase/verify completion

0	No interrupt request.
1	Interrupt request.

WMODE Write mode flag

EMODE Erase mode flag

VMODE Verify mode flag

FEARL (Flash and EEPROM address low Register) : F2H

7	6	5	4	3	2	1	0
ARL7	ARL6	ARL5	ARL4	ARL3	ARL2	ARL1	ARL0
R/W	R.W	R.W	R.W	R.W	R.W	R.W	R.W

Initial value : 00H

ARL[7:0] Flash and EEPROM address low

CHKSUM[7:0] Checksum Result from auto verify mode (PCRCRD==0)

CRC[7:0] CRC Result from auto verify mode (PCRCRD==1)

FEARM (Flash and EEPROM address middle Register) : F3H

7	6	5	4	3	2	1	0
ARM7	ARM6	ARM5	ARM4	ARM3	ARM2	ARM1	ARM0
R/W	R.W	R.W	R.W	R.W	R.W	R.W	R.W

Initial value : 00H

ARM[7:0] Flash and EEPROM address middle

CHKSUM[15:8] Checksum Result from auto verify mode

CRC[15:8] CRC Result from auto verify mode (PCRCRD==1)

FEARH (Flash and EEPROM address high Register) : F4H

7	6	5	4	3	2	1	0
ARH7	ARH6	ARH5	ARH4	ARH3	ARH2	ARH1	ARH0
R/W	R.W	R.W	R.W	R.W	R.W	R.W	R.W

Initial value : 00H

ARH[7:0] Flash and EEPROM address high

CHKSUM[23:16] Checksum Result from auto verify mode

FEAR registers are used for program, erase and auto-verify. In program and erase mode, it is page address and ignored the same least significant bits as the number of bits of page address. In auto-verify mode, address increases automatically by one.

FEARs are write-only register. Reading these registers returns 24-bit checksum result

FEDR (Flash and EEPROM data control Register) : F5H

7	6	5	4	3	2	1	0
FEDR7	FEDR6	FEDR5	FEDR4	FEDR3	FEDR2	FEDR1	FEDR0
W	W	W	W	W	W	W	W

Initial value : 00H

FEDR[7:0] Flash and EEPROM data

Data register. In no program/erase/verify mode, READ/WRITE of FECR read or write data from EEPROM or Flash to this register or from this register to Flash or EEPROM.

The sequence of writing data to this register is used for EEPROM program entry. The mode entrance sequence is to write 0xA5 and 0x5A to it in order.

FETCR (Flash and EEPROM Time control Register) : EDH

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

TCR[7:0] Flash and EEPROM Time control

Program and erase time is controlled by setting FETCR register. Program and erase timer uses 10-bit counter. It increases by one at the edge of 128KHz clock (f_{FETCR}). It is cleared when program or erase starts. Timer stops when 10-bit counter is same to FETCR. PEVBSY is cleared when program, erase or verify start and set when program, erase or verify stop.

Maximum program/erase time: $1/f_{FETCR}$ * 1024 = 8ms ($1/f_{FETCR}$ = 7.8us @128KHz clock)

In the case of 50% of error rate of counter source clock, program or erase time are 4~12ms

* Program/erase time calculation

for page write or erase, Tpe = (TCON+1) * 2 * $1/f_{FETCR}$

for bulk erase, Tbe = (TCON+1) * 4 * $1/f_{FETCR}$

※ Recommended program/erase time (FETCR = FFh)

	Min	Typ	Max	Unit
program/erase time	2	4	6	ms
bulk erase time	4	8	12	ms

8.3 Memory map

8.3.1 Flash Memory Map

Program memory uses 4-Kbyte of Flash memory. It is read by byte and written by byte or page. One page is 32-byte

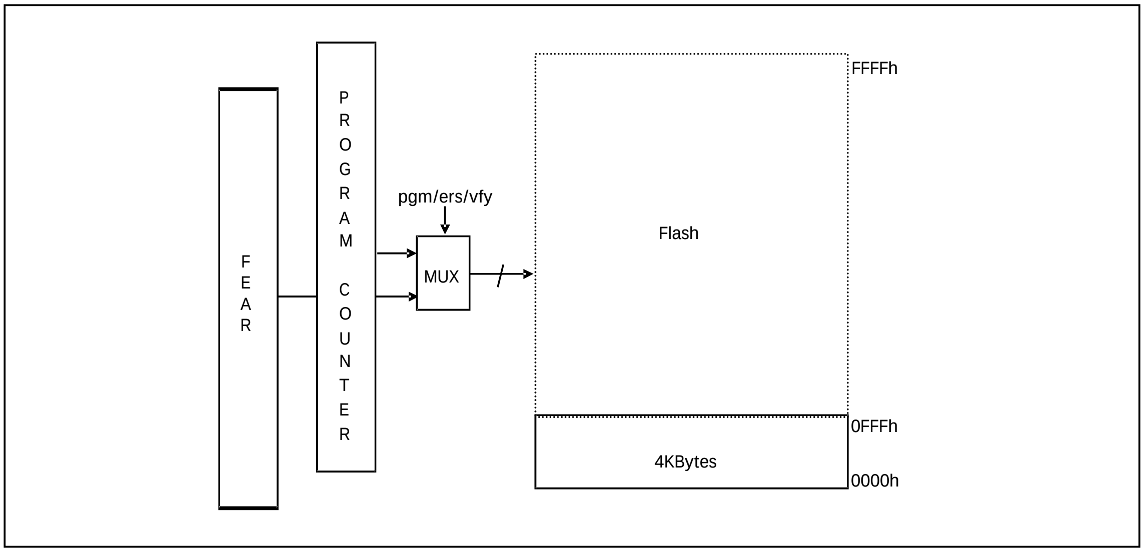


Figure 8.1 Flash Memory Map

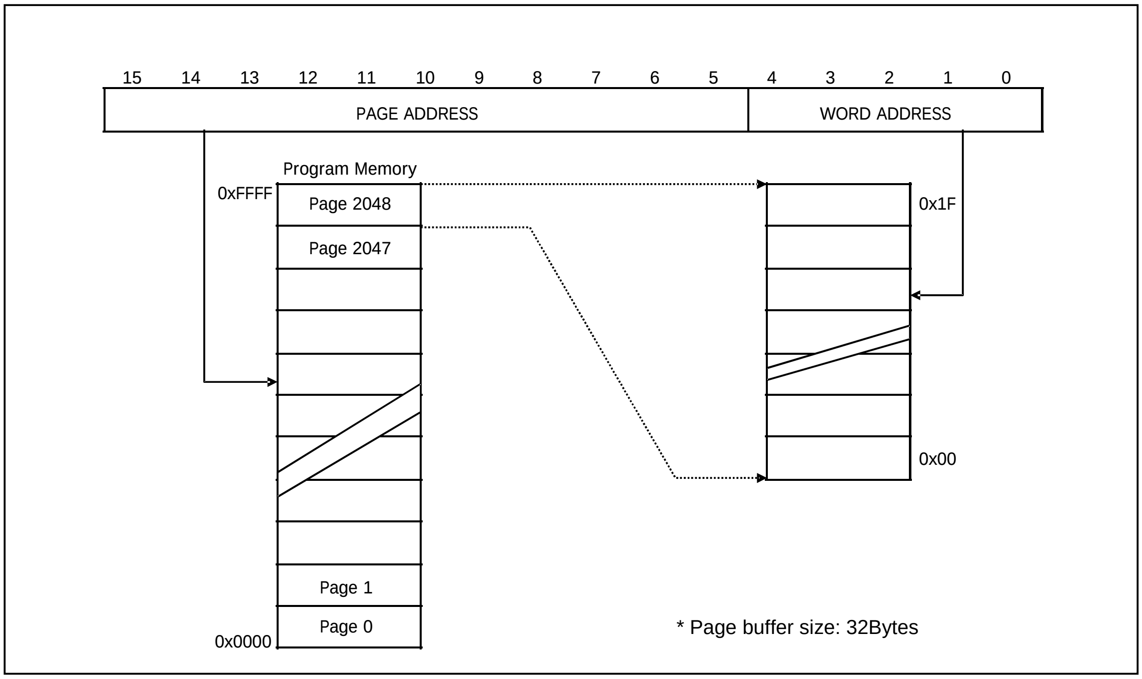


Figure 8.2 Address configuration of Flash memory

8.3.2 Data EEPROM Memory Map

Data EEPROM memory uses 256-byte of EEPROM. It is read by byte and written by byte or page. One page is 16-byte. It is mapped to external data memory of 8051

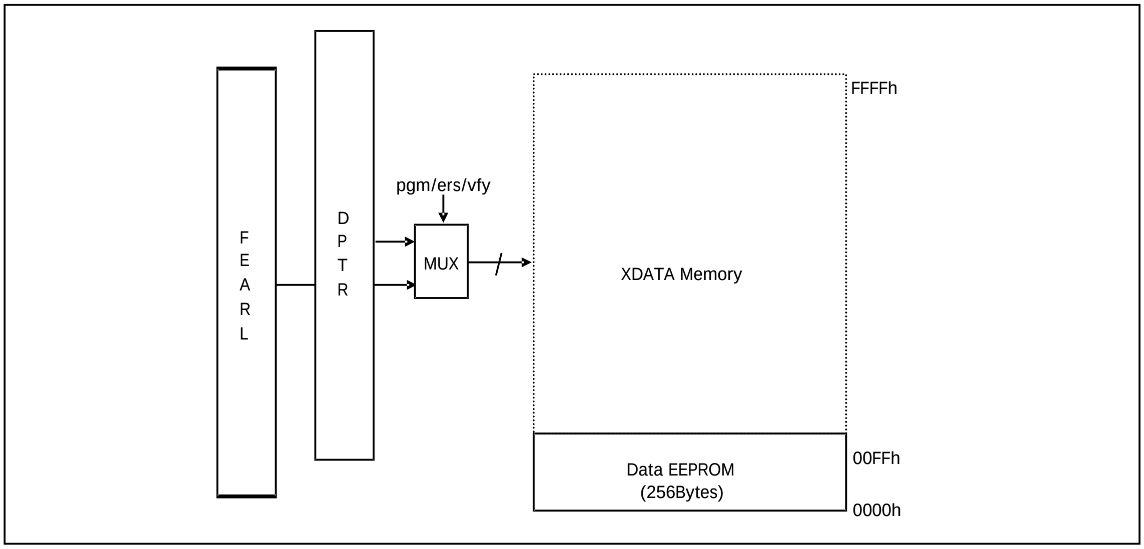


Figure 8.3 Data EEPROM memory map

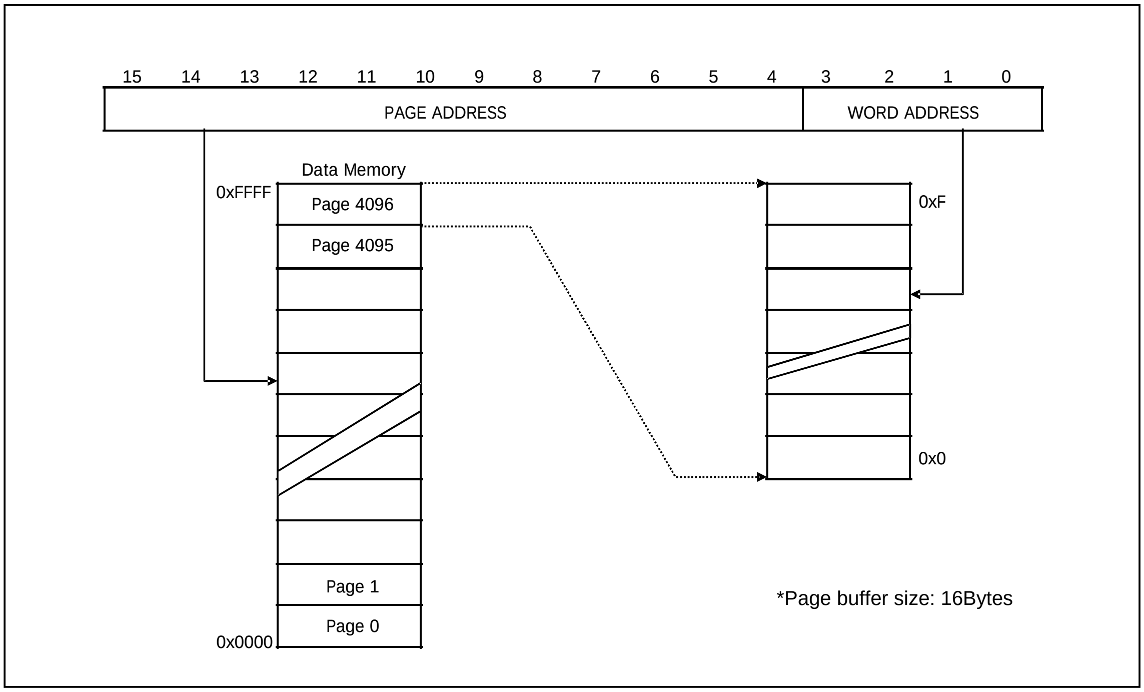


Figure 8.4 Address configuration of data EEPROM

8.4 Serial In-System Program Mode

Serial in-system program uses the interface of debugger which uses two wires. Refer to chapter 14 in details about debugger

8.4.1 Flash operation

Configuration(This Configuration is just used for follow description)

7	6	5	4	3	2	1	0
-	FEMR[4]& [1]	FEMR[5]& [1]	-	-	FEMR[2]	FECR[6]	FECR[7]
-	ERASE&VFY	PGM&VFY	-	-	OTPE	AEE	AEF

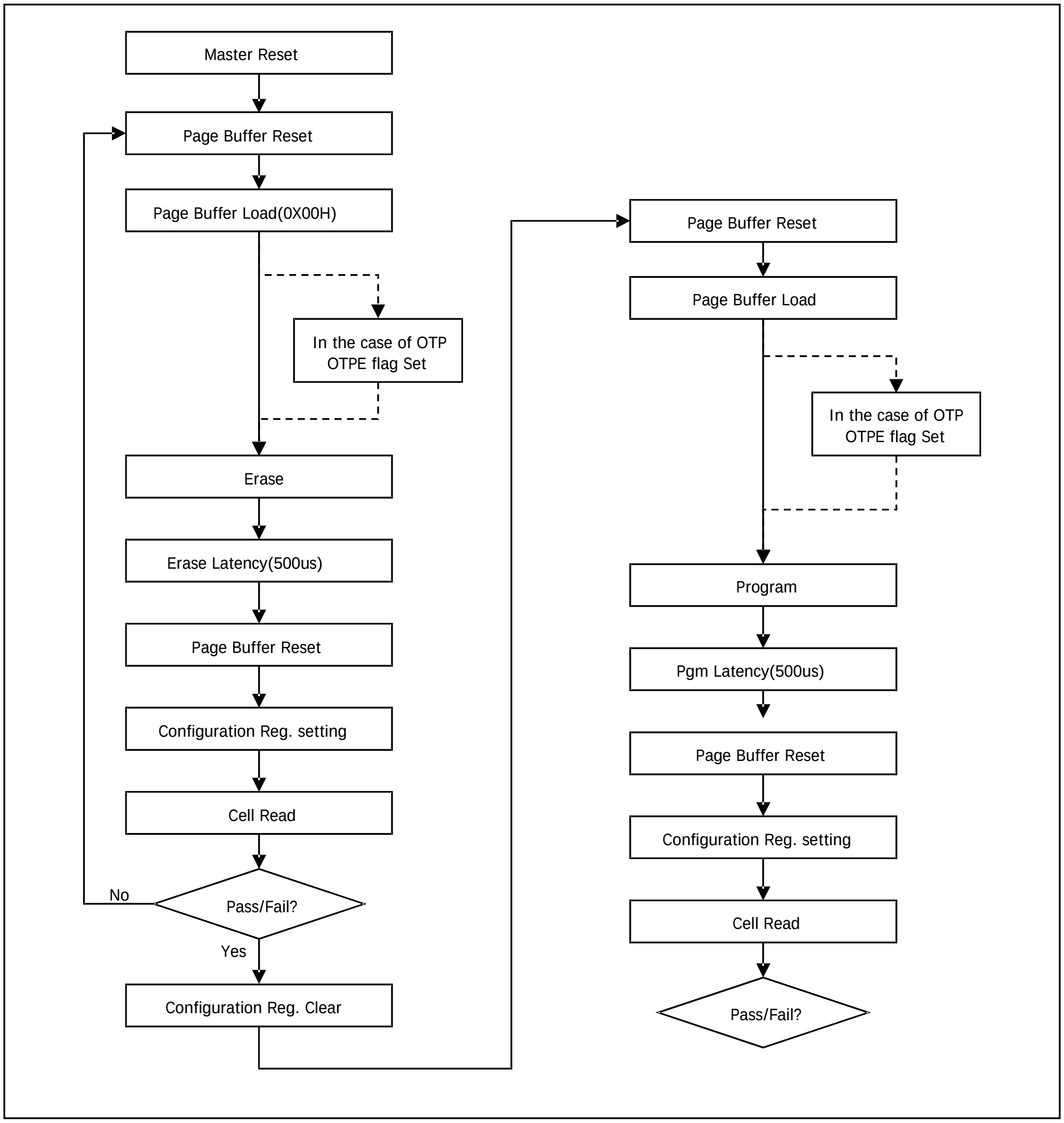


Figure 8.5 The sequence of page program and erase of Flash memory

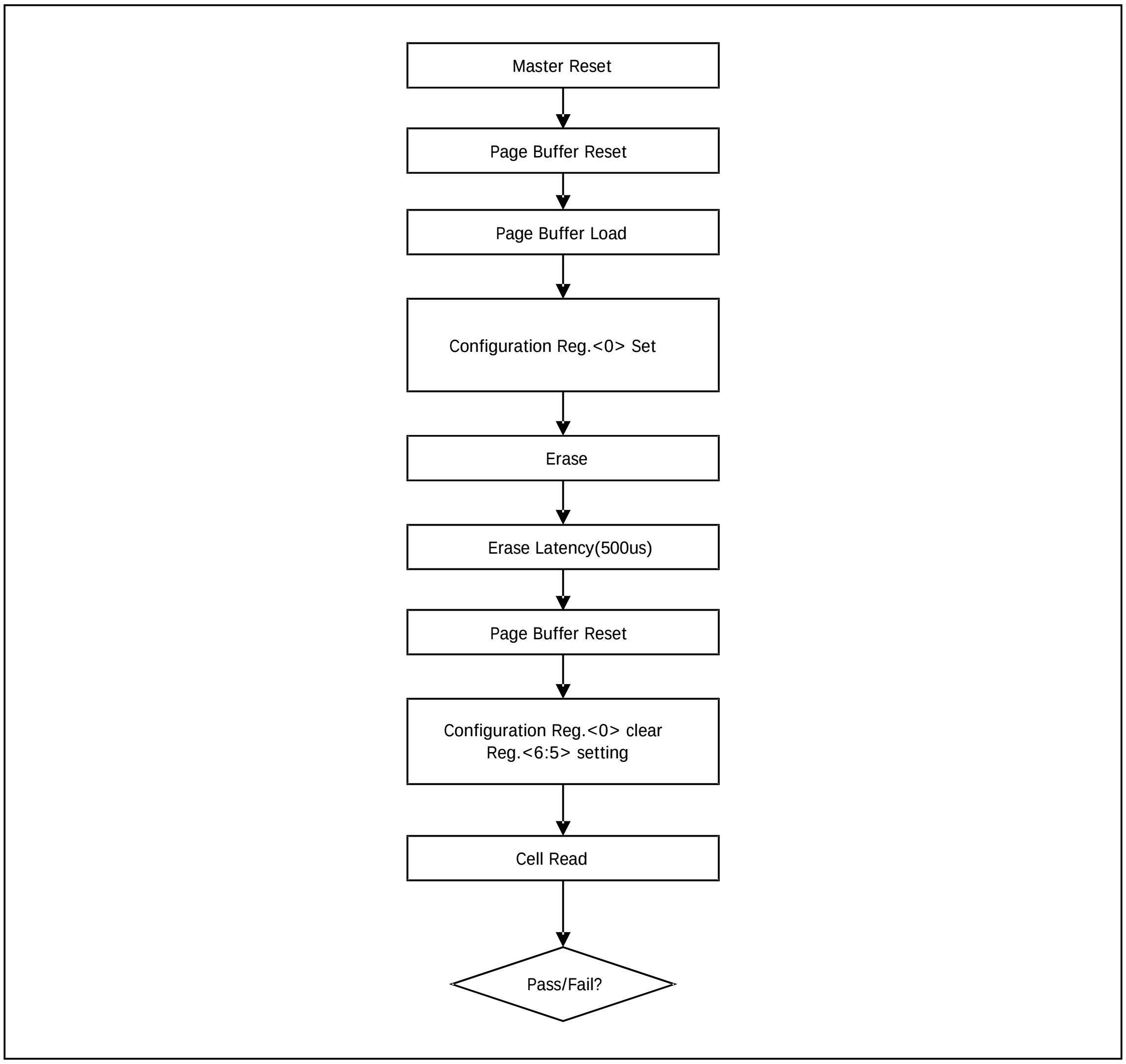


Figure 8.6 The sequence of bulk erase of Flash memory

8.4.1.1 Flash Read

Step 1. Enter OCD(=ISP) mode.
Step 2. Set ENBDM bit of BCR.
Step 3. Enable debug and Request debug mode.
Step 4. Read data from Flash.

8.4.1.2 Enable program mode

Step 1. Enter OCD(=ISP) mode.[1]
Step 2. Set ENBDM bit of BCR.
Step 3. Enable debug and Request debug mode.
Step 4. Enter program/erase mode sequence.[2]
(1) Write 0xAA to 0xF555.
(2) Write 0x55 to 0xFAAA.

(3) Write 0xA5 to 0xF555.

[1] Refer to how to enter ISP mode..

[2] Command sequence to activate Flash write/erase mode. It is composed of sequentially writing data of Flash memory.

8.4.1.3 Flash write mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write data to page buffer.(Address automatically increases by twin.)

Step 5. Set write mode. FEMR:1010_0001

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start program. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

Step 11. Repeat step2 to step 8 until all pages are written.

8.4.1.4 Flash page erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write 'h00 to page buffer. (Data value is not important.)

Step 5. Set erase mode. FEMR:1001_0001

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start erase. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

Step 11. Repeat step2 to step 8 until all pages are erased.

8.4.1.5 Flash bulk erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write 'h00 to page buffer. (Data value is not important.)

Step 5. Set erase mode. FEMR:1001_0001.

(Only main cell area is erased. For bulk erase including OTP area, select OTP area.(set FEMR to 1000_1101.)

Step 6. Set FETCR

Step 7. Start bulk erase. FECR:1000_1011

Step 9. Insert one NOP operation

Step 9. Read FESR until PEVBSY is 1.

8.4.1.6 Flash OTP area read mode

Step 1. Enter OCD(=ISP) mode.

Step 2. Set ENBDM bit of BCR.

Step 3. Enable debug and Request debug mode.

Step 4. Select OTP area. FEMR:1000_0101

Step 5. Read data from Flash.

8.4.1.7 Flash OTP area write mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write data to page buffer.(Address automatically increases by twin.)

Step 5. Set write mode and select OTP area. FEMR:1010_0101

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start program. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

8.4.1.8 Flash OTP area erase mode

Step 1. Enable program mode.

Step 2. Reset page buffer. FEMR: 1000_0001 FECR:0000_0010

Step 3. Select page buffer. FEMR:1000_1001

Step 4. Write 'h00 to page buffer. (Data value is not important.)

Step 5. Set erase mode and select OTP area. FEMR:1001_0101

Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx

Step 7. Set FETCR.

Step 8. Start erase. FECR:0000_1011

Step 9. Insert one NOP operation

Step 10. Read FESR until PEVBSY is 1.

8.4.1.9 Flash program verify mode

Step 1. Enable program mode.

Step 2. Set program verify mode. FEMR:1010_0011

Step 3. Read data from Flash.

8.4.1.10 OTP program verify mode

Step 1. Enable program mode.
Step 2. Set program verify mode. FEMR:1010_0111
Step 3. Read data from Flash.

8.4.1.11 Flash erase verify mode

Step 1. Enable program mode.
Step 2. Set erase verify mode. FEMR:1001_0011
Step 3. Read data from Flash.

8.4.1.12 Flash page buffer read

Step 1. Enable program mode.
Step 2. Select page buffer. FEMR:1000_1001
Step 3. Read data from Flash.

8.4.2 Data EEPROM operation

Program and erase operation of Data EEPROM are executed by direct and indirect address mode.

Direct address mode uses external data area of 8051. Indirect address mode uses address register of SFR area..

8.4.2.1 Data EEPROM Read

Step 1. Enter OCD(=ISP) mode.
Step 2. Set ENBDM bit of BCR.
Step 3. Enable debug and Request debug mode.
Step 4. Read data from Data EEPROM.

8.4.2.2 Enable program mode

Step 1. Enter OCD(=ISP) mode.[1]
Step 2. Set ENBDM bit of BCR.
Step 3. Enable debug and Request debug mode.
Step 4. Enter program/erase mode sequence.[2]
(1) Write 0xA5 to FEDR.
(2) Write 0x5A to FEDR.

[1] Refer to how to enter ISP mode..

[2] Command sequence to activate data EEPROM write/erase mode. It is composed of sequentially writing to data register(FEDR)

8.4.2.3 EEPROM write mode

Step 1. Enable program mode.
Step 2. Reset page buffer. FEMR: 0100_0001 FECR:0000_0010
Step 3. Select page buffer. FEMR:0100_1001
Step 4. Write data to page buffer.(Address automatically increases by twin.)
Step 5. Set write mode. FEMR:0110_0001
Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx
Step 7. Set FETCR.
Step 8. Start program. FECR:0000_1011
Step 9. Insert one NOP operation
Step 10. Read FESR until PEVBSY is 1.
Step 11. Repeat step2 to step 8 until all pages are written.

8.4.2.4 EEPROM page erase mode

Step 1. Enable program mode.
Step 2. Reset page buffer. FEMR: 0100_0001 FECR:0000_0010
Step 3. Select page buffer. FEMR:0100_1001
Step 4. Write 'h00 to page buffer. (Data value is not important.)
Step 5. Set erase mode. FEMR:0101_0001
Step 6. Set page address. FEARH:FEARM:FEARL=20'hx_xxxx
Step 7. Set FETCR.
Step 8. Start erase. FECR:0000_1011
Step 9. Insert one NOP operation
Step 10. Read FESR until PEVBSY is 1.
Step 11. Repeat step2 to step 8 until all pages are erased.

8.4.2.5 EEPROM bulk erase mode

Step 1. Enable program mode.
Step 2. Reset page buffer. FEMR: 0100_0001 FECR:0000_0010
Step 3. Select page buffer. FEMR:0100_1001
Step 4. Write 'h00 to page buffer. (Data value is not important.)
Step 5. Set erase mode. FEMR:0101_0001.
Step 6. Set FETCR
Step 7. Start bulk erase. FECR:0100_1011
Step 8. Insert one NOP operation
Step 9. Read FESR until PEVBSY is 1.

8.4.2.6 Data EEPROM program verify mode

Step 1. Enable program mode.
Step 2. Set program verify mode. FEMR:0110_0011

Step 3. Read data from Flash.

8.4.2.7 Data EEPROM erase verify mode

Step 1. Enable program mode.
Step 2. Set erase verify mode. FEMR:0101_0011
Step 3. Read data from Flash.

8.4.2.8 Data EEPROM page buffer read

Step 1. Enable program mode.
Step 2. Select page buffer. FEMR:0100_1001
Step 3. Read data from Flash.

8.4.3 Summary of Flash and Data EEPROM Program/Erase Mode

Table 8-2 Flash and Data EEPROM Operation Mode

Operation mode		Description
FLASH	Flash read	Read cell by byte.
	Flash write	Write cell by bytes or page.
	Flash page erase	Erase cell by page.
	Flash bulk erase	Erase the whole cells.
	Flash program verify	Read cell in verify mode after programming.
	Flash erase verify	Read cell in verify mode after erase.
	Flash page buffer load	Load data to page buffer.
EEPROM	Data EEPROM read	Read cell by byte.
	Data EEPROM write	Write cell by bytes or page.
	Data EEPROM page erase	Erase cell by page.
	Data EEPROM bulk erase	Erase the whole cells.
	Data EEPROM program verify	Read cell in verify mode after programming.
	Data EEPROM erase verify	Read cell in verify mode after erase.
	Data EEPROM page buffer load	Load data to page buffer.

8.5 Parallel Mode

8.5.1 Overview

Parallel program mode transfers address and data by byte. 3-byte address can be entered by one from the lease significant byte of address. If only LSB is changed, only one byte can be transferred. And if the second byte is changed, the first and second byte can be transferred. Upper 4-bit of the most significant byte selects memory to be accessed. Table 8-1 shows memory type. Address auto-increment is supported when read or write data without address

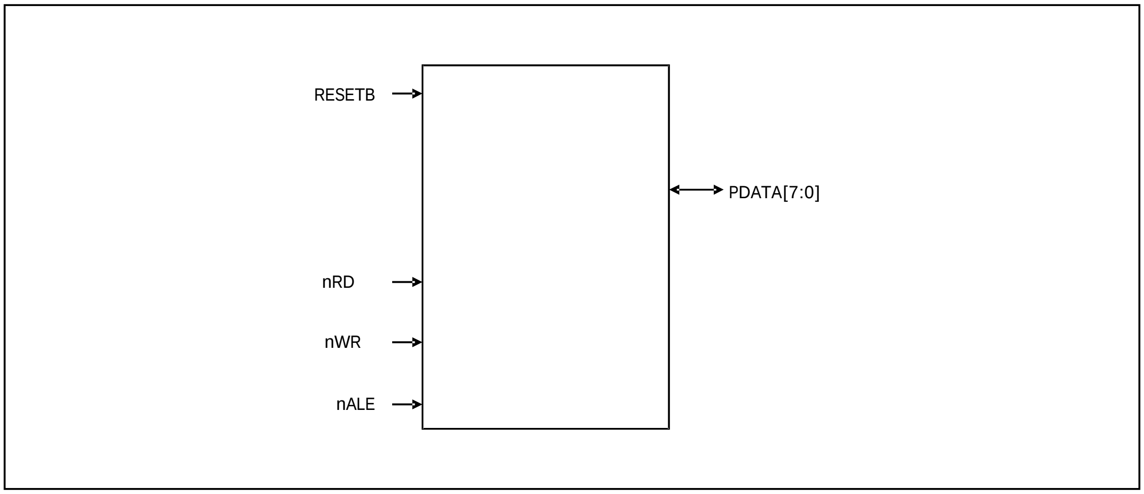


Figure 8.7 Pin diagram for parallel programming

Table 8-3 The selection of memory type by ADDRH[7:4]

ADDRH[7:4]				Memory Type
0	0	0	0	Program Memory
0	0	0	1	External Memory
0	0	1	0	SFR

8.5.2 Parallel Mode instruction format

Table 8-4 Parallel mode instruction format

Instruction	Signal	Instruction Sequence													
n-byte data read with 3-byte address	nALE	L		L		L		H		H		H		H	
	nWR	L	H	L	H	L	H	H	H	H	H	H	H	H	H
	nRD	H	H	H	H	H	H	L	H	L	H	L	H	L	H
	PDATA	ADDRL		ADDRM		ADDRH		DATA0		DATA1		---		DATAn	
n-byte data write with 3-byte address	nALE	L		L		L		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		ADDRM		ADDRH		DATA0		DATA1		---		DATAn	
n-byte data read with 2-byte address	nALE	L		L		H		H		H		H		H	
	nWR	L	H	L	H	H	H	H	H	H	H	H	H	H	H
	nRD	H	H	H	H	L	H	L	H	L	H	L	H	L	H
	PDATA	ADDRL		ADDRM		DATA0		DATA1		DATA2		---		DATAn	
n-byte data write with 2-byte address	nALE	L		L		H		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		ADDRM		DATA0		DATA1		DATA2		---		DATAn	
n-byte data read with 1-byte address	nALE	L		H		H		H		H		H		H	
	nWR	L	H	H	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		DATA0		DATA1		DATA2		DATA3		---		DATAn	
n-byte data write with 1-byte address	nALE	L		H		H		H		H		H		H	
	nWR	L	H	L	H	L	H	L	H	L	H	L	H	L	H
	nRD	H	H	H	H	H	H	H	H	H	H	H	H	H	H
	PDATA	ADDRL		DATA0		DATA1		DATA2		DATA3		---		DATAn	

8.5.3 Parallel Mode timing diagram

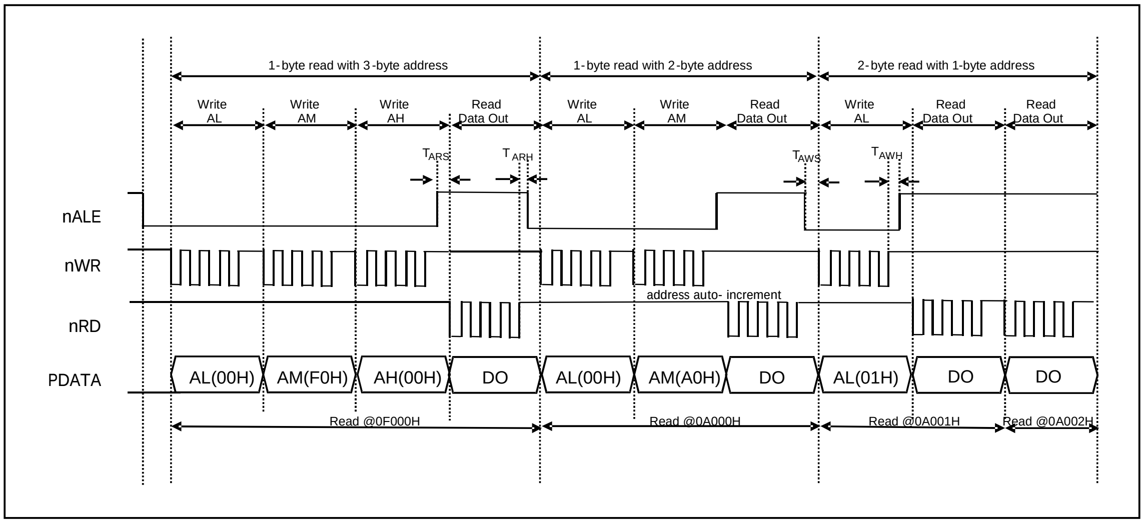


Figure 8.8 Parallel Byte Read Timing of Program Memory

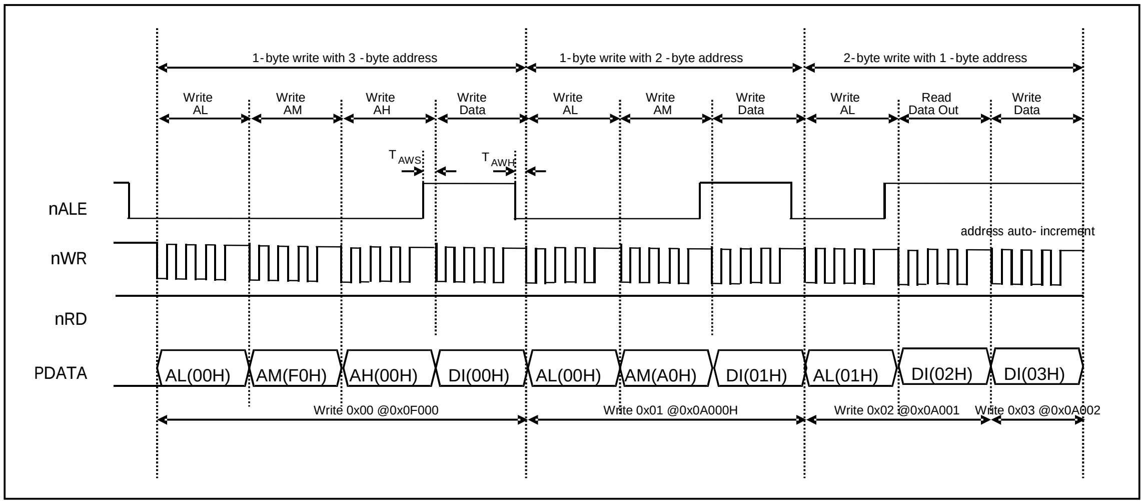


Figure 8.9 Parallel Byte Write Timing of Program Memory

Table 8-5 Control Pin Description

Pin	P01	P02	P03	P04	P05
Function	PDATA[0]	PDATA[1]	nALE	nWR	nRD

8.6 Mode entrance method of ISP and byte-parallel mode

8.6.1 Mode entrance method for ISP

TARGET MODE	R03	R05	R03

OCD(ISP)	‘hC	‘hC	‘hC

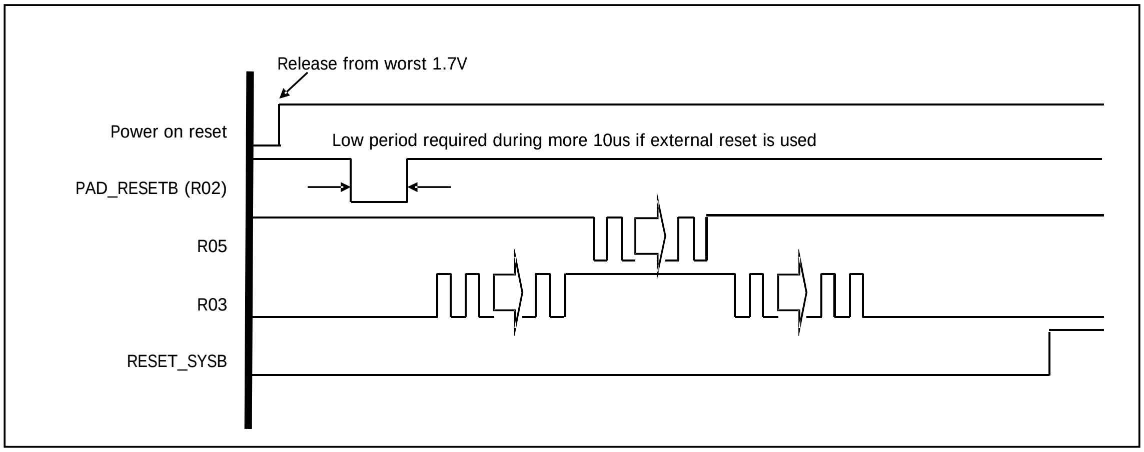


Figure 8.10 ISP mode

8.6.2 Mode entrance of Byte-parallel

TARGET MODE	R0[7:6],R0[3],R0[1]	R0[7:6],R0[3],R0[1]	R0[7:6],R0[3],R0[1]
Byte-Parallel Mode	4‘h5	4‘hA	4‘h5

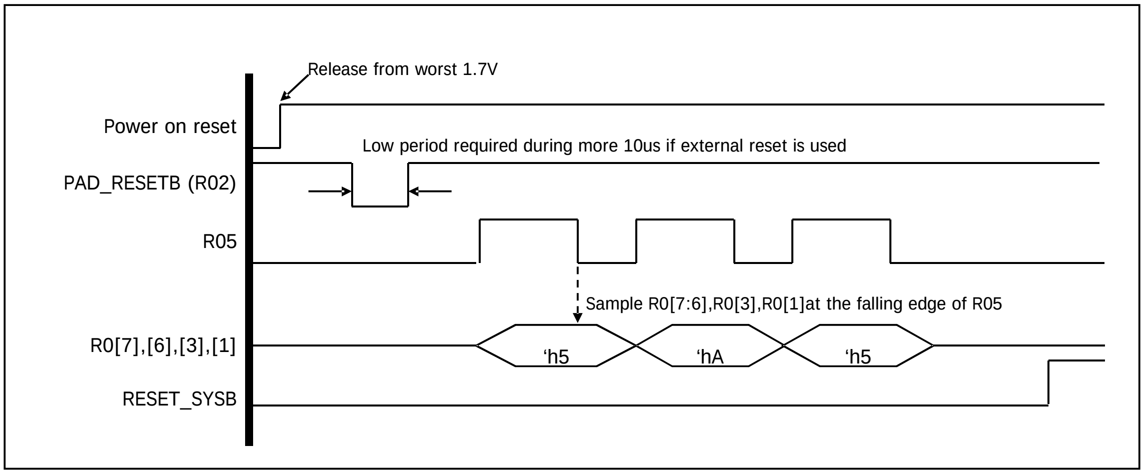

Release from worst 1.7V
Power on reset
Low period required during more 10us if external reset is used
PAD_RESETB (R02)
R05
Sample R0[7:6],R0[3],R0[1]at the falling edge of R05
R0[7],[6],[3],[1]
‘h5
‘hA
‘h5
RESET_SYSB

Figure 8.11 Byte-parallel mode (10pin package only)

8.7 Security

The Z51F0410 MCU provides one Lock bit which can be left unprogrammed (“0”) or can be programmed (“1”) to obtain the additional features listed in Table 8-6. The Lock bit can only be erased to “0” with the bulk erase command

Table 8-6 Security policy using lock-bits

LOCK MODE		USER MODE												ISP/PMODE/BTMODE											
		Flash				DATA EEPROM				OTP				Flash				DATA EEPROM				OTP			
LOCKE	LOCKF	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE	R	W	PE	BE
0	0	X	X	X	X	O	O	O	O	X	X	X	X	O	O	O	O	O	O	O	O	O	O	O	O
0	1	X	X	X	X	O	O	O	O	X	X	X	X	X	X	X	O	O	O	O	O	O	X	X	O
1	0	X	X	X	X	O	O	O	O	X	X	X	X	O	◇	◇	◇	X	X	X	O	O	◇	◇	◇
1	1	X	X	X	X	O	O	O	O	X	X	X	X	X	X	X	◇	X	X	X	O	O	X	X	◇

- LOCKF: Lock bit of Flash memory
- LOCKE: Lock bit of data EEPROM
- R: Read
- W: Write
- PE: Page erase
- BE: Bulk Erase
- O: Operation is possible.
- X: Operation is impossible.
- ◇: When LOCKE is programmed, each operation can be done after data EEPROM is erased with the bulk erase command.

9. Configure option

9.1 Configure option Control Register

FUSE_CONF (Pseudo-Configure Data) : 0x00:FDH

7	6	5	4	3	2	1	0
BITP[1:0]		SXINEN	XINENA	RSTDIS	ENCRYPT	LOCKE	LOCKF
R	R	R	R	R	R	R	R

Initial value : 00H

BITP[1;0] BIT Period Control (Reset Release Time Control)
00 16ms
01 32ms
10 64ms

SXINEN External Sub Oscillator Enable Bit
0 Sub OSC disable (default)
1 Sub OSC enable

XINENA External Main Oscillator Enable Bit
0 Main OSC disable (default)
1 Main OSC Enable

RSTDIS External RESETB disable Bit
0 External RESET enable
1 External RESET disable

ENCRYPT Super Lock Enable Bit (Encryption Mode)
0 Super Lock Disable
1 Super Lock Enable (ROM Data Encrypted)

LOCKE DATA memory LOCK bit
0 LOCK Disable
1 LOCK Enable

LOCKF CODE memory LOCK bit
0 LOCK Disable
1 LOCK Enable

USEED0 (User Lock Private Key) : 0x06:D4H

7	6	5	4	3	2	1	0
USEED0[7:0]							
-	-	-	-	-	-	-	-

Initial value : 00H

USEED0[7:0] User Seed Key for Code Encryption
USEED0 is only available when ENCRYPT is 1

USEED1 (User Lock Private Key) : 0x07:D3H

7	6	5	4	3	2	1	0
USEED1[7:0]							
-	-	-	-	-	-	-	-

Initial value : 00H

USEED1[7:0] User Seed Key for Code Encryption

USEED1 is only available when ENCRYPT is 1

9.2 Serial ID

The Z51F0410 MCU supports 16 bytes of user serial ID for device identification. These ID are mapped to SFR area. User can write SID to 0x0A~0x19 of OTP area and read SID of SFR by indirect adressing mode.

SID is mappet to D6H and SIDA to D5H.

SIDX (Serial ID X) : D5H

7	6	5	4	3	2	1	0
SIDX[7:0]							
-	-	-	-	-	-	-	-

Initial value : 00H

SIDX[7:0] User Serial ID X X is from 0 to 15.

Table 9-1 Summary of SID

	OTP address	*Initial value*
SID0	0x0A	00H
SID1	0x0B	00H
SID2	0x0C	00H
SID3	0x0D	00H
SID4	0x0E	00H
SID5	0x0F	00H
SID6	0x10	00H
SID7	0x11	00H
SID8	0x12	00H
SID9	0x13	00H
SID10	0x14	00H
SID11	0x15	00H
SID12	0x16	00H
SID13	0x17	00H
SID14	0x18	00H
SID15	0x19	00H

10. APPENDIX

10.1 Instruction Table

Instructions are either 1, 2 or 3 bytes long as listed in the 'Bytes' column below.
Each instruction takes either 1, 2 or 4 machine cycles to execute as listed in the following table. 1 machine cycle comprises 2 system clock cycles.

ARITHMETIC				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ADD A,Rn	Add register to A	1	1	28-2F
ADD A,dir	Add direct byte to A	2	1	25
ADD A,@Ri	Add indirect memory to A	1	1	26-27
ADD A,#data	Add immediate to A	2	1	24
ADDC A,Rn	Add register to A with carry	1	1	38-3F
ADDC A,dir	Add direct byte to A with carry	2	1	35
ADDC A,@Ri	Add indirect memory to A with carry	1	1	36-37
ADDC A,#data	Add immediate to A with carry	2	1	34
SUBB A,Rn	Subtract register from A with borrow	1	1	98-9F
SUBB A,dir	Subtract direct byte from A with borrow	2	1	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	1	96-97
SUBB A,#data	Subtract immediate from A with borrow	2	1	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	1	08-0F
INC dir	Increment direct byte	2	1	05
INC @Ri	Increment indirect memory	1	1	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	1	18-1F
DEC dir	Decrement direct byte	2	1	15
DEC @Ri	Decrement indirect memory	1	1	16-17
INC DPTR	Increment data pointer	1	2	A3
MUL AB	Multiply A by B	1	4	A4
DIV AB	Divide A by B	1	4	84
DA A	Decimal Adjust A	1	1	D4

LOGICAL				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ANL A,Rn	AND register to A	1	1	58-5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56-57
ANL A,#data	AND immediate to A	2	1	54
ANL dir,A	AND A to direct byte	2	1	52
ANL dir,#data	AND immediate to direct byte	3	2	53
ORL A,Rn	OR register to A	1	1	48-4F
ORL A,dir	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect memory to A	1	1	46-47
ORL A,#data	OR immediate to A	2	1	44
ORL dir,A	OR A to direct byte	2	1	42
ORL dir,#data	OR immediate to direct byte	3	2	43
XRL A,Rn	Exclusive-OR register to A	1	1	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	1	65
XRL A, @Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	2	1	64
XRL dir,A	Exclusive-OR A to direct byte	2	1	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03

RRC A	Rotate A right through carry	1	1	13

DATA TRANSFER				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOV A,Rn	Move register to A	1	1	E8-EF
MOV A,dir	Move direct byte to A	2	1	E5
MOV A,@Ri	Move indirect memory to A	1	1	E6-E7
MOV A,#data	Move immediate to A	2	1	F8-FF
MOV Rn,A	Move A to register	1	1	A8-AF
MOV Rn,dir	Move direct byte to register	2	2	78-7F
MOV Rn,#data	Move immediate to register	2	1	F5
MOV dir,A	Move A to direct byte	2	1	88-8F
MOV dir,Rn	Move register to direct byte	2	2	85
MOV dir,dir	Move direct byte to direct byte	3	2	86-87
MOV dir,@Ri	Move indirect memory to direct byte	2	2	75
MOV dir,#data	Move immediate to direct byte	3	2	F6-F7
MOV @Ri,A	Move A to indirect memory	1	1	A6-A7
MOV @Ri,dir	Move direct byte to indirect memory	2	2	76-77
MOV @Ri,#data	Move immediate to indirect memory	2	1	90
MOV DPTR,#data	Move immediate to data pointer	3	2	93
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	2	83
MOVC A,@A+PC	Move code byte relative PC to A	1	2	E2-E3
MOVX A,@Ri	Move external data(A8) to A	1	2	F2-F3
MOVX A,@DPTR	Move external data(A16) to A	1	2	F0
MOVX @Ri,A	Move A to external data(A8)	1	2	C0
MOVX @DPTR,A	Move A to external data(A16)	1	2	23
PUSH dir	Push direct byte onto stack	2	2	C0
POP dir	Pop direct byte from stack	2	2	D0
XCH A,Rn	Exchange A and register	1	1	C8-CF
XCH A,dir	Exchange A and direct byte	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	1	1	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	1	D6-D7

BOOLEAN				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	1	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	1	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	1	B2
ANL C,bit	AND direct bit to carry	2	2	82
ANL C,/bit	AND direct bit inverse to carry	2	2	B0
ORL C,bit	OR direct bit to carry	2	2	72
ORL C,/bit	OR direct bit inverse to carry	2	2	A0
MOV C,bit	Move direct bit to carry	2	1	A2
MOV bit,C	Move carry to direct bit	2	2	92

BRANCHING				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ACALL addr 11	Absolute jump to subroutine	2	2	11→F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	2	22
RETI	Return from interrupt	1	2	32
AJMP addr 11	Absolute jump unconditional	2	2	01→E1
LJMP addr 16	Long jump unconditional	3	2	02
SJMP rel	Short jump (relative address)	2	2	80
JC rel	Jump on carry = 1	2	2	40
JNC rel	Jump on carry = 0	2	2	50
JB bit,rel	Jump on direct bit = 1	3	2	20
JNB bit,rel	Jump on direct bit = 0	3	2	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	2	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	2	60

JNZ rel	Jump on accumulator ≠ 0	2	2	70
CJNE A,dir,rel	Compare A,direct jne relative	3	2	B5
CJNE A,#d,rel	Compare A,immediate jne relative	3	2	B4
CJNE Rn,#d,rel	Compare register, immediate jne relative	3	2	B8-BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	2	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	3	2	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5

MISCELLANEOUS				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
NOP	No operation	1	1	00

ADDITIONAL INSTRUCTIONS (selected through EO[7:4])				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOVC @(DPTR++),A	M8051W/M8051EW-specific instruction supporting software download into program memory	1	2	A5
TRAP	Software break command	1	1	A5

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps and calls, with the top 3 bits of the code being used to store the top three bits of the destination address.

The CJNE instructions use the abbreviation #d for immediate data; other instructions use #data.

10.2 Instructions on how to use the input port.

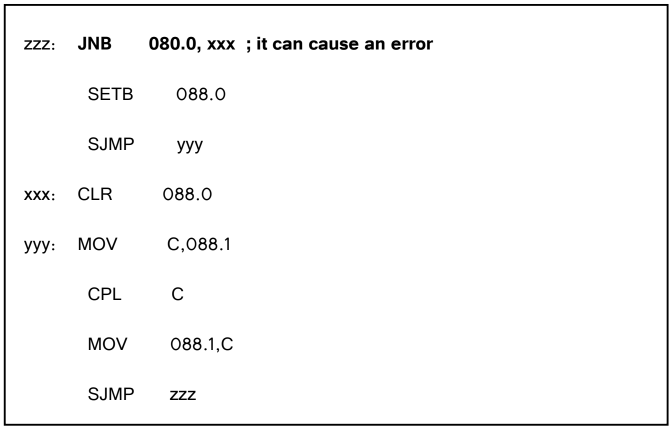
- Error occur status
  - Using compare jump instructions with input port, it could cause error due to the timing conflict inside the MCU.
  - Compare jump Instructions which cause potential error used with input port condition:

```
 JB     bit, rel   ; jump on direct bit=1
JNB     bit, rel   ; jump on direct bit=0
JBC     bit, rel   ; jump on direct bit=1 and clear
CJNE  A, dir, rel          ; compare A, direct jne relative
 DJNZ  dir, rel   ; decrement direct byte, jnz relative
```

  - It is only related with Input port. Internal parameters, SFRs and output bit ports don't cause an y error by using compare jump instructions.
  - If input signal is fixed, there is no error in using compare jump instructions.

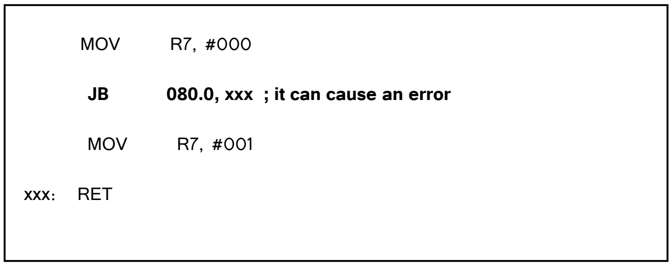
- Error status example

```
while(1){
  if (P00==1){ P10=1; }
  else { P10=0; }
  P11^=1;
}
```



```
zzz:  JNB      080.0, xxx  ; it can cause an error
        SETB     088.0
        SJMP     yyy
xxx:  CLR      088.0
yyy:  MOV      C,088.1
        CPL      C
        MOV      088.1,C
        SJMP     zzz
```

```
unsigned char ret_bit_err(void)
{
  return !P00;
}
```



```
       MOV      R7, #000
        JB       080.0, xxx  ; it can cause an error
        MOV      R7, #001
xxx:  RET
```

- Preventative measures (2 cases)

- Do not use input bit port for bit operation but for byte operation. Using byte operation instead of bit operation will not cause any error in using compare jump instructions for input port.

```
while(1){
 if ((P0&0x01)==0x01){ P10=1; }
  else { P10=0; }
   P11^=1;
}
```

```
zzz:   MOV     A, 080        ; read as byte
        JNB     0E0.0, xxx    ; compare
        SETB     088.0
        SJMP     yyy
xxx:   CLR      088.0
yyy:   MOV      C,088.1
        CPL      C
        MOV      088.1,C
        SJMP     zzz
```

If you use input bit port for compare jump instruction, you have to copy the input port as internal register or carry bit and then use compare jump instruction.

```
bit tt;
while(1){
  tt=P00;
  if (tt==0){ P10=1;}
  else {   P10=0;}
  P11^=1;
}
```

```
zzz:   MOV     C,080.0     ; input port  to C
        MOV     020.0, C     ; C to internal register
        JB       020.0, xxx   ; compare with internal register
        SETB     088.0
        SJMP     yyy
xxx:   CLR      088.0
yyy:   MOV      C,088.1
        CPL      C
        MOV      088.1,C
        SJMP     zzz
```

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base or consider participating in the Zilog Forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.