ABOV SEMICONDUCTOR
8-BIT SINGLE-CHIP MICROCONTROLLERS

MC81F4104

MC81F4104 M/B/S

User's Manual (Ver. 1.39)

SEMICONDUCTOR

Version 1.39
Published by FAE Team

Additional information of this manual may be served by ABOV Semiconductor offices in Korea or Distributors.

ABOV Semiconductor reserves the right to make changes to any information here in at any time without notice.

The information, diagrams and other data in this manual are correct and reliable; however, ABOV Semiconductor is in no way responsible for any violations of patents or other rights of the third party generated by the use of this manual.

REVISION HISTORY

VERSION 1.40 (April 24, 2012) This book

Add the chapter ' 7.9 POR Electrical CHARACTERISTICS'.

VERSION 1.38 (December 28, 2010) This book

Change '5.5v' to "VDD level" in DC Electrical Characteristics description(page 22).

Change '5.0v' to "External RC Condition" in DC Electrical Characteristics description(page 27).

Change '5.0v' to "Internal RC Condition" in DC Electrical Characteristics description(page 28).

VERSION 1.37 (December 23, 2009) This book

Update the chapter ' 20.3 Reset circuit'.

Add Reset pin information.

VERSION 1.36 (December 15, 2009) This book

Update the chapter ' 20.3 Reset circuit'.

VERSION 1.35 (October 19, 2009) This book

Change EVA.board picture. (the board's color is changed from blue to green)

VERSION 1.34 (September 30, 2009)

Add more tools at "1.3 Development Tools".

VERSION 1.33 (September 18, 2009)

Remove rising/falling time at LVR electrical characteristics.

Change '1.83v' to "POR level" in POR description.

Add POR level at "DC CHARACTERISTICS".

Add ROM option read timing information.

Add "Typical Characteristics".

VERSION 1.22 (July 7, 2009)

"23.3 Hardware Conditions to Enter the ISP Mode" is updated.

Note of R03 port control register is updated.

VERSION 1.21 (June 29, 2009)

8 SOP ordering name is changed from "MC81F4104D" to "MC81F4104M".

VERSION 1.2 (June 29, 2009)

Remove 'WDT' at "Stop release" description. 'WDT' is not a release source of STOP mode.

VERSION 1.1 (June 17, 2009)

Add rom writing endurance at features.

VERSION 1.0 (June 15, 2009)

Remove “preliminary”.

Some errata are fixed.

Remove “(or 16Bit *1ch)” at timer clause of the feature page.

VERSION 0.9 Preliminary (April 16, 2009)

Add a sub-chapter ‘Changing the stabilizing time’ at the chapter ‘Power down operation’.

Add a note for R00/R01 ports after R0CONH description.

One of BIT’s clock source ‘2048’ is changed to ‘1024’.

VERSION 0.8 Preliminary (April 8, 2009)

Description of ISP chapter is updated.

Operation range is changed.(2.0v~5.5v > 2.2v~5.5v)

VERSION 0.7 Preliminary (April 1, 2009)

Chapter ‘7.ELECTRICAL CHARACTERISTICS’ is updated.

VERSION 0.6 Preliminary (March 5, 2009)

Correct pin map diagram in the chapter ’22.EMULATOR’.

Move the SCLK pin for ISP is moved to R04 port.

Note for ADC recommended circuit is changed.

VERSION 0.5 Preliminary (February 12, 2009)

Update the chapter ‘6. PORT STRUCTURE’.

Update the chapter ‘7. ELECTRICAL CHARACTERISTICS’.

Update the chapter ’23. IN SYSTEM PROGRAMMING’.

VERSION 0.4 Preliminary (December 19, 2008)

Block diagrams of Timer 2/3 and PWM are corrected.

VERSION 0.3 Preliminary (December 8, 2008)

Operating Voltage Changed (2.2V~5.5V → 2.0V~5.5V)

VERSION 0.2 Preliminary (November 17, 2008)

Some errata are corrected.

VERSION 0.1 Preliminary (November 14, 2008)

Some errata are corrected.

VERSION 0.0 Preliminary (November 12, 2008)

TABLE OF CONTENTS

MC81F4104

8 bit MCU with 12-bit A/D Converter

1. OVERVIEW

1.1 Description

MC81F4104 is a CMOS 8 bit MCU which provides a 4K bytes FLASH-ROM and 192 bytes RAM.
It has following major features,

12 bit ADC : It has 7(5) ch A/D Converter which can be used to measure minute electronic voltage and currents.

810 Core : Same with ABOV's 800 Core but twice faster. 800 Core use a divided system clock but 810 Core use the system clock directly

1.2 Features

ROM(FLASH) : 4K Bytes
(Endurance: 100 cycle)

SRAM **:** 192 Bytes

Minimum instruction execution time
166n sec at 12MHz (NOP instruction)

12-bit A/D converter : 7 ch

General Purpose I/O(GPIO)
10-pin PKG: 8
8-pin PKG: 6

Timer/Counter
8Bit x 2ch

PWM
10 bit High Speed PWM * 1ch

Watchdog timer(WDT) **:** 8Bit x 1 ch

Basic Interval Timer(BIT) : 8Bit x 1ch

Interrupt Source : 9 ch
External Interrupts : 3 ch
Timer 2/3 Match/Overflow
WDT, BIT

Power On Reset (POR)

Low Voltage Reset (LVR)
4 level detector (2.4/2.7/3.0/4.0V)

Power Down Mode
Stop mode
Sleep mode

Operating Voltage & Frequency
4.0V – 5.5V (at 1.0 – 12.0MHz)
2.7V – 5.5V (at 1.0 – 8.0MHz)
2.2V – 5.5V (at 1.0 – 4.2MHz)

Operating Temperature
- 40°C ~ 85°C

Oscillator Type
Crystal, Ceramic, RC for main clock
Internal Oscillator (8MHz/4MHz/2MHz/1MHz)

Package
10 SSOP, 8 PDIP/SOP
Available Pb free package

1.3 Development Tools

The MC81F4104 is supported by a full-featured macro assembler, C-Compiler, an in-circuit emulator CHOICE-Dr.^{TM} , FALSH programmers and ISP tools. There are two different type of programmers such as single type and gang type. For more detail, Macro assembler operates under the MS-Windows 95 and up versioned Windows OS. And HMS800C compiler only operates under the MS-Windows 2000 and up versioned Windows OS.

Please contact sales part of ABOV semiconductor. And you can see more information at (http://www.abov.co.kr)

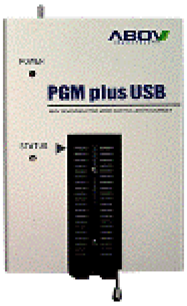


Figure 1-1 PGMplusUSB (Single Writer)

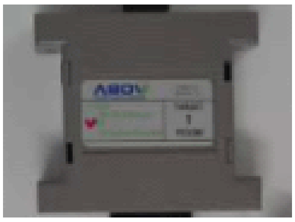


Figure 1-2 SIO ISP (In System Programmer)

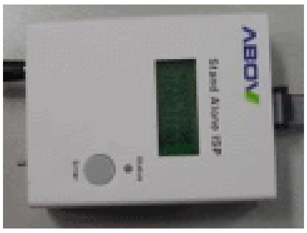


Figure 1-3 StandAlone ISP

(VDD power is not supplied)

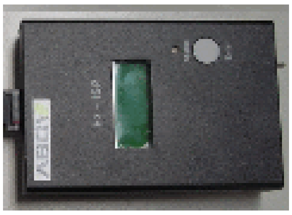


Figure 1-4 Ez-ISP

(VDD supplied Standalone type ISP)

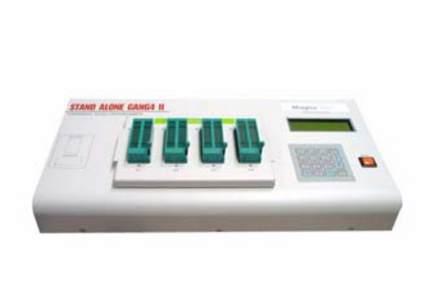


Figure 1-5 StandAlone Gang4 (for Mass Production)

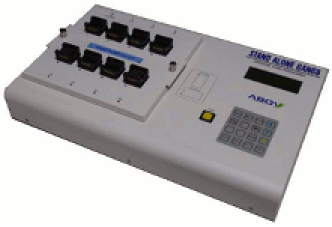


Figure 1-6 StandAlone Gang8 (for Mass Production)

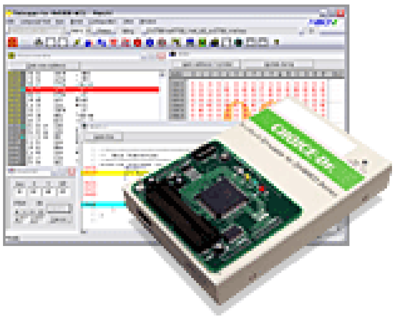


Figure 1-7 Choice-Dr (Emulator)

1.4 Ordering Information

Device Name	FLASH ROM	RAM	Package
MC81F4104M	4K Bytes	192 Bytes	8_SOP
MC81F4104B			8_PDIP
MC81F4104S			10_SSOP

2. BLOCK DIAGRAM

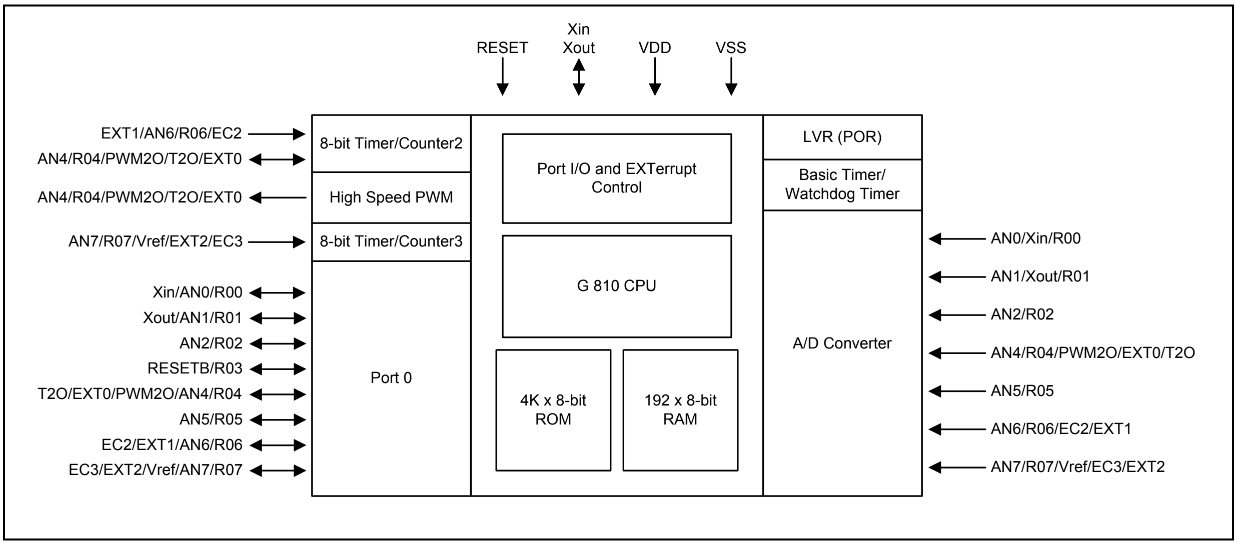
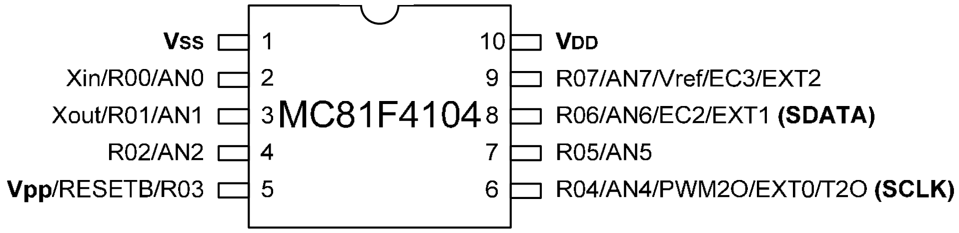


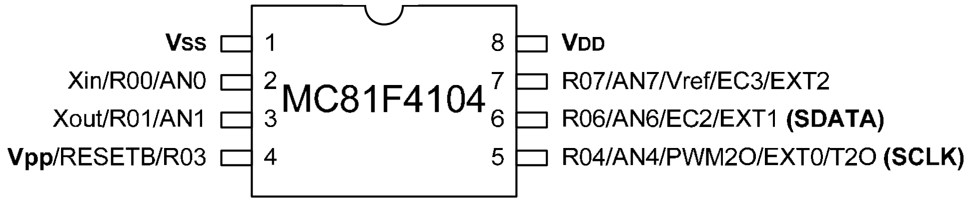
Figure 2-1 System Block Diagram

3. PIN ASSIGNMENT

3.1 10 pin- SSOP



3.2 8 pin- PDIP/SOP



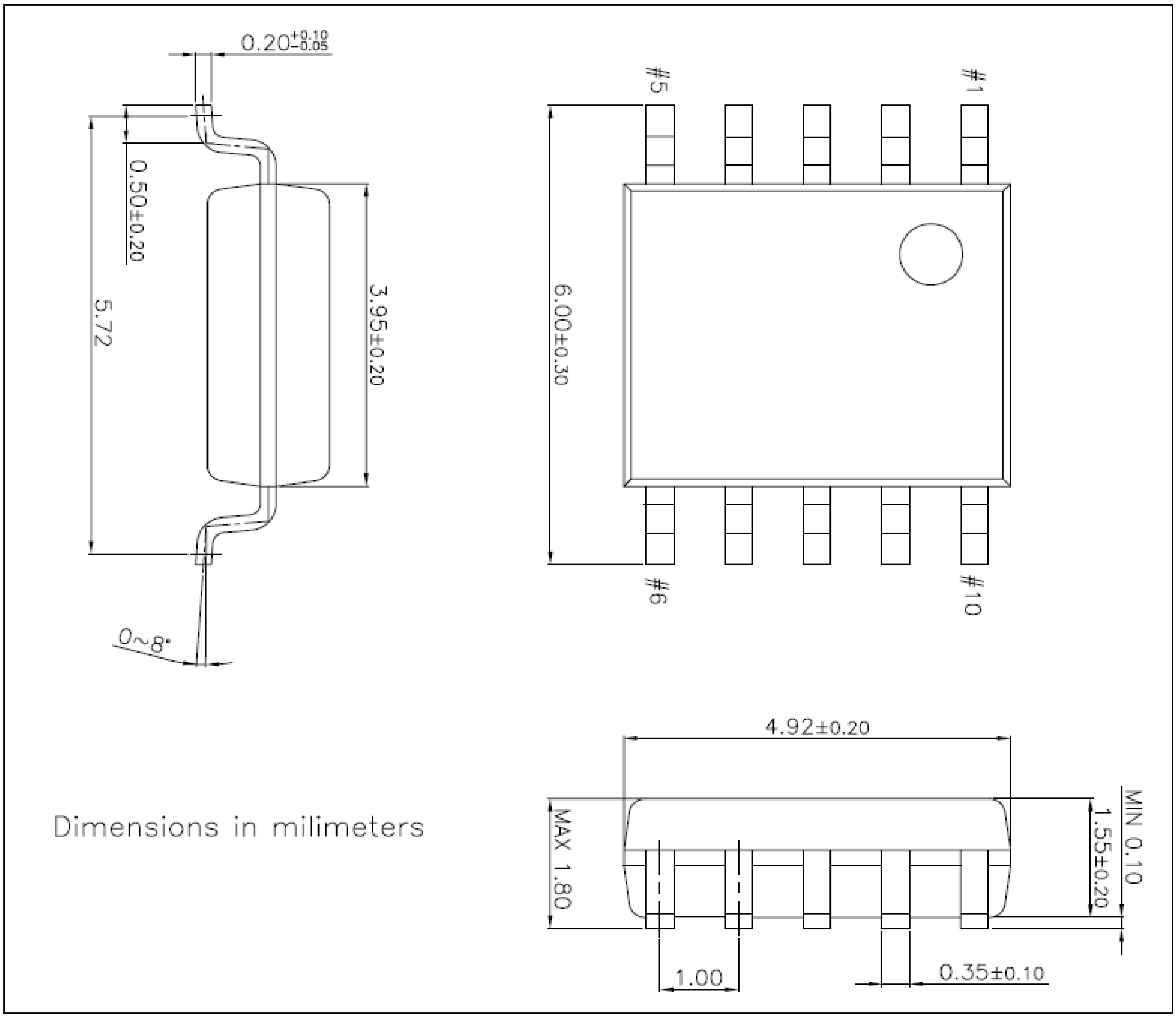
3.3 Summary

	alternative functions	Pin number		Pin status at RESET
		10pin	8pin	
R00	AN0/Xin	2	2	input
R01	AN1/Xout	3	3	input
R02	AN2	4	x	Open-drain output
R03	Vpp/RESETB	5	4	input
R04	AN4/EXT0/PWM2O/T2O	6	5	input
R05	AN5	7	x	Open-drain output
R06	AN6/EXT1/EC2	8	6	input
R07	AN7/EXT2/Vref/EC3	9	7	input
VDD	-	10	8	-
VSS	-	1	1	-

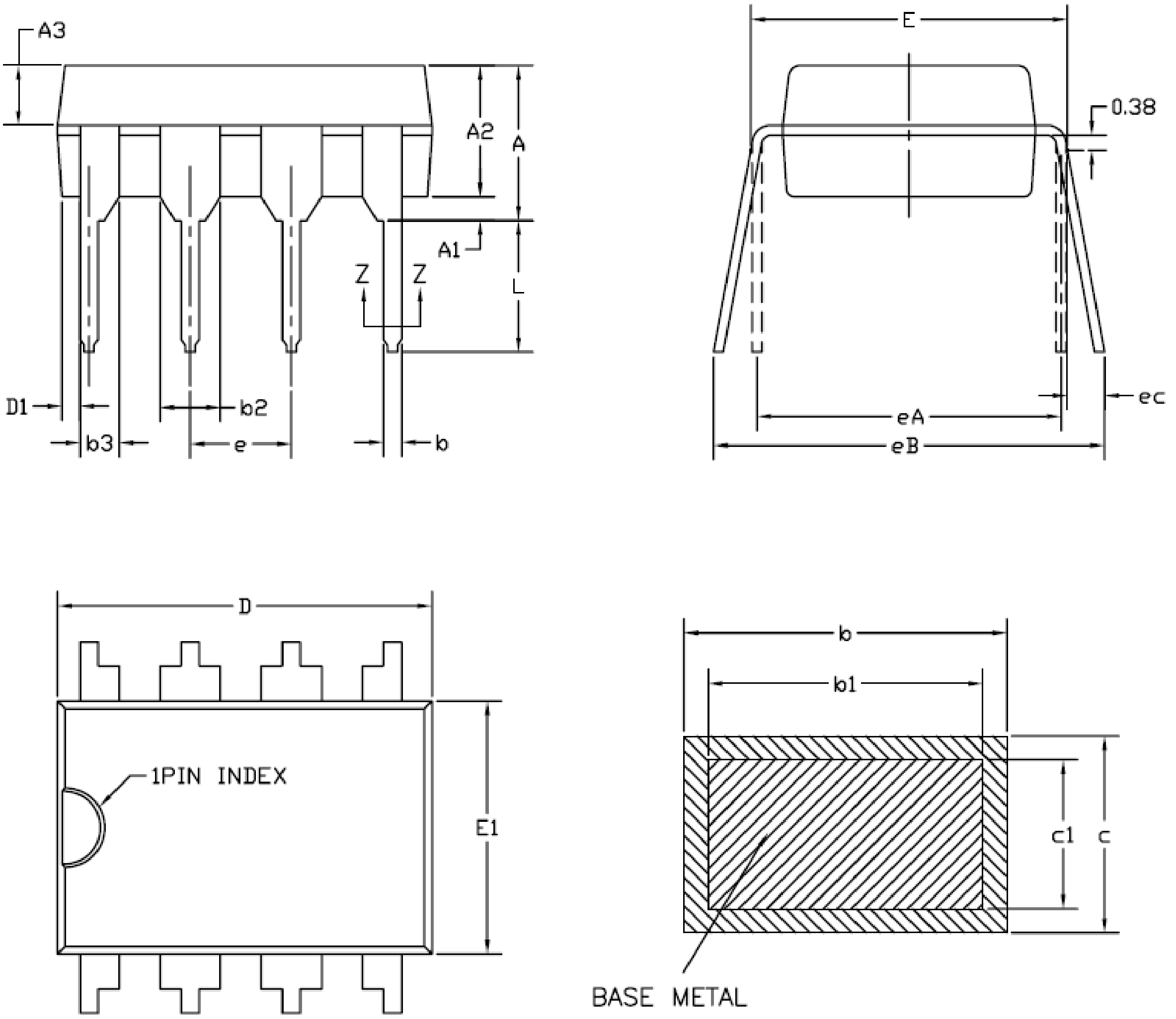
Note :
Some pins are initialized by open-drain output mode, when the device is reset. Because the pins are hided in 8 pin package and it is stable that hided pins are be in open-drain-output mode.

4. PACKAGE DIAGRAM

4.1 10 SSOP - MC81F4104S



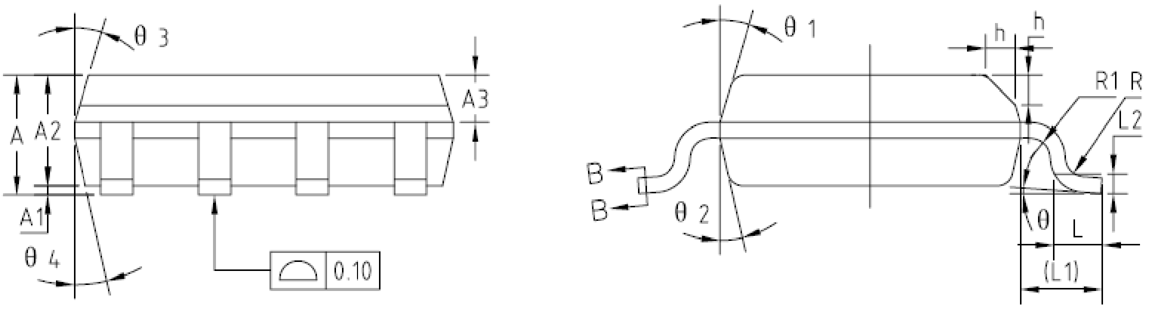
4.2 8 PDIP - MC81F4104B

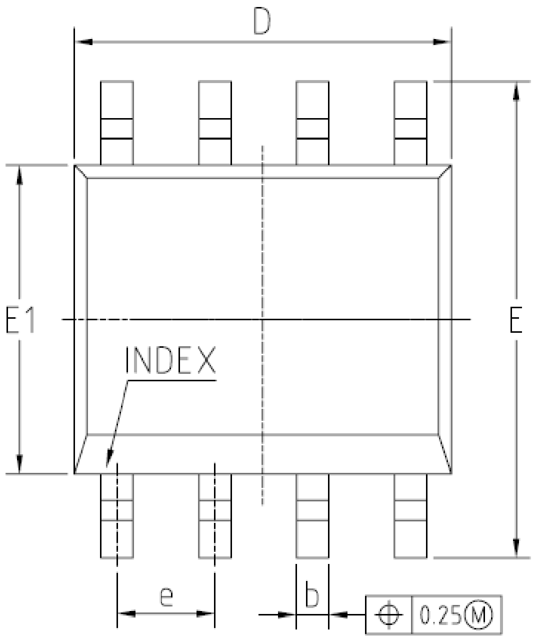


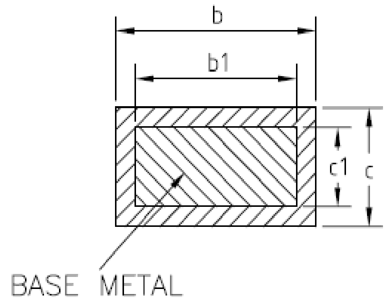
COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	4.80
A1	0.50	–	–
A2	3.10	3.30	3.50
A3	1.40	1.50	1.60
b	0.38	–	0.55
b1	0.38	0.46	0.51
b2	1.47	1.52	1.57
b3	0.89	0.99	1.09
c	0.21	–	0.35
c1	0.20	0.25	0.30
D	9.10	9.20	9.30
D1	0.13	–	–
E	7.62	7.87	8.25
E1	6.25	6.35	6.45
e	2.54BSC		
eA	7.62BSC		
eB	7.62	8.80	10.90
ec	0	–	1.52
L	2.92	3.30	3.81

4.3 8 SOP - MC81F4104M







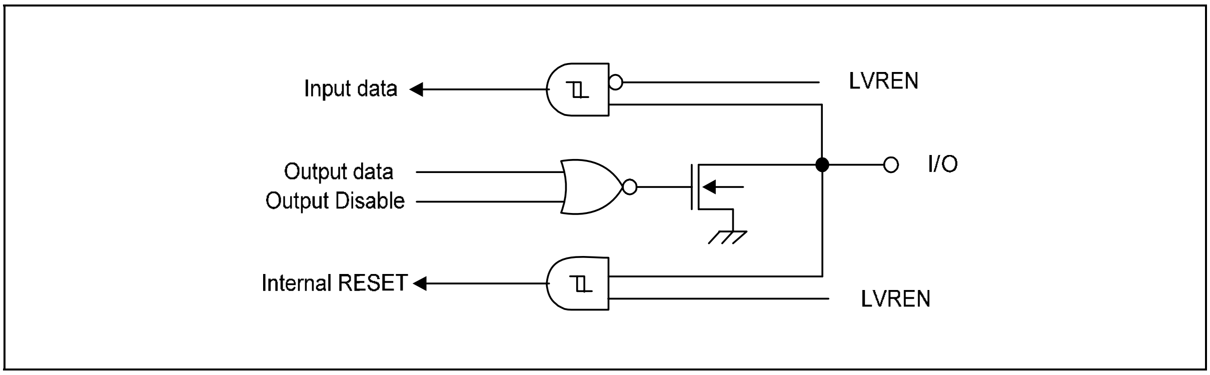
COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	1.35	1.55	1.75
A1	0.10	0.15	0.25
A2	1.25	1.40	1.65
A3	0.50	0.60	0.70
b	0.38	–	0.51
b1	0.37	0.42	0.47
c	0.17	–	0.25
c1	0.17	0.20	0.23
D	4.80	4.90	5.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	1.27BSC		
L	0.45	0.60	0.80
L1	1.04REF		
L2	0.25BSC		
R	0.07	–	–
R1	0.07	–	–
h	0.30	0.40	0.50
θ	0°	–	8°
θ 1	15°	17°	19°
θ 2	11°	13°	15°
θ 3	15°	17°	19°
θ 4	11°	13°	15°

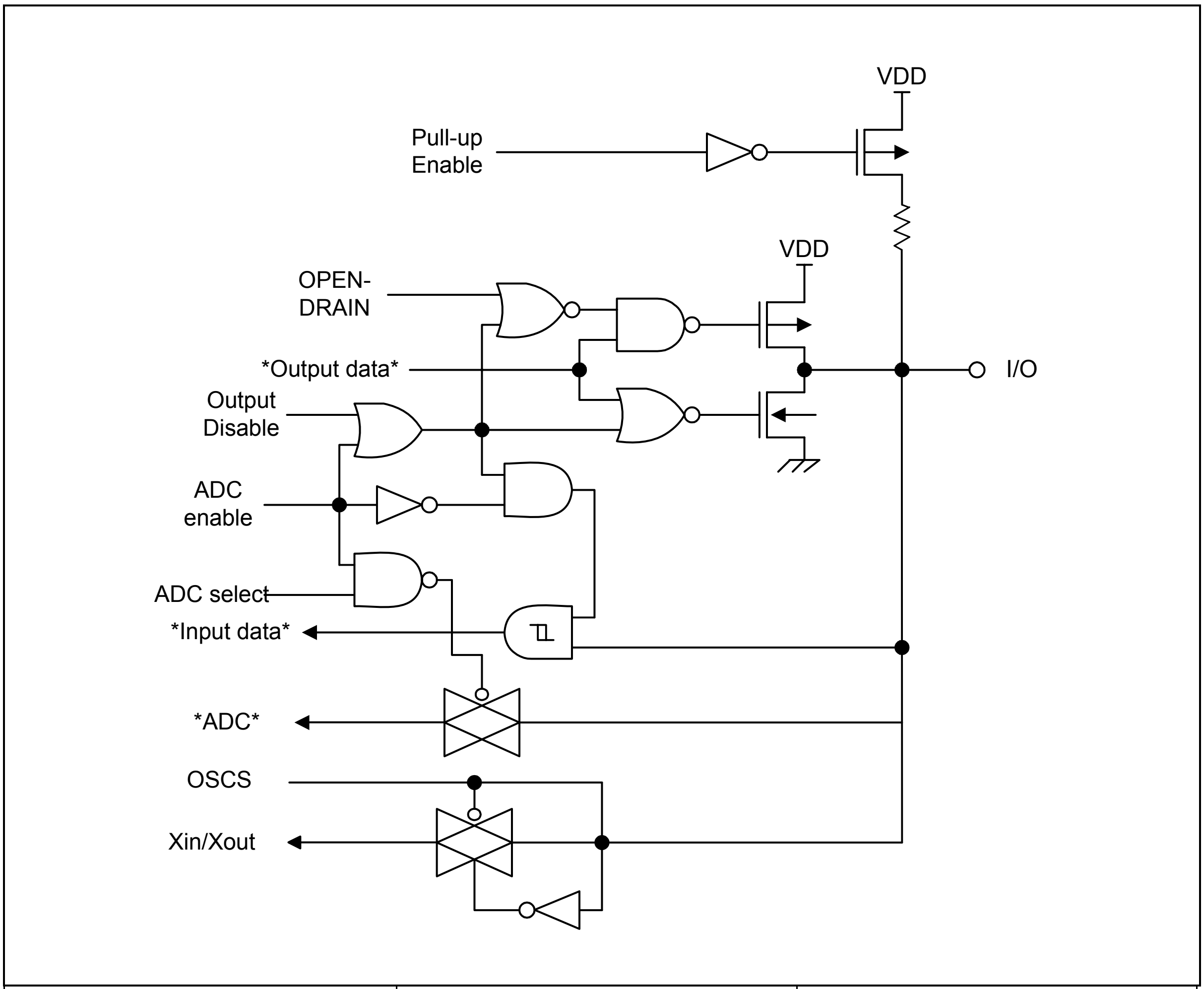
5. PIN DESCRIPTION

Pin Names	I/O	Pin Description	Alternative Functions
R00 R01 R02 R03 R04 R05 R06 R07	I/O	This port is a 1-bit programmable I/O pin. Schmitt trigger input, Push-pull, or Open-drain output port. When used as an input port, a Pull-up resistor can be specified in 1-bit.	Xin/AN0 Xout/AN1 AN2 RESETB PWM2O/T2O/AN4/EXT0 AN5 EC2/AN6/EXT1 EC3/Vref/AN7/EXT2
EXT0	I/O	External interrupt input/Timer 2 capture input	R04/PWM2O/T2O/AN4
EXT1	I/O	External interrupt input	R06/EC2/AN6/
EXT2	I/O	External interrupt input/Timer 3 capture input	R07/EC3/Vref/AN7
T2O	I/O	Timer 2 clock output	R04/PWM2O/AN4/EXT0
EC2	I/O	Timer 2 event count input	R06/AN6/EXT1
PWM2O	I/O	PWM 2 clock output	R04/T2O/AN4/EXT0
EC3	I/O	Timer 3 event count input	R07/Vref/AN7/EXT2
AN0 AN1 AN2 AN4 AN5 AN6 AN7	I/O	ADC input pins	R00/Xin R01/Xout R02 R04/PWM2O/T2O/EXT0 R05 R06/EC2/EXT1 R07/EC3/Vref/EXT2
RESETB	I	System reset pin	R03
XIN XOUT	–	Main oscillator pins	R00/AN0, R01/AN1
VDD VSS	–	Power input pins	– –
VREF	–	A/D converter reference voltage	R07/AN7/EC3/EXT2

6. PORT STUCTURE

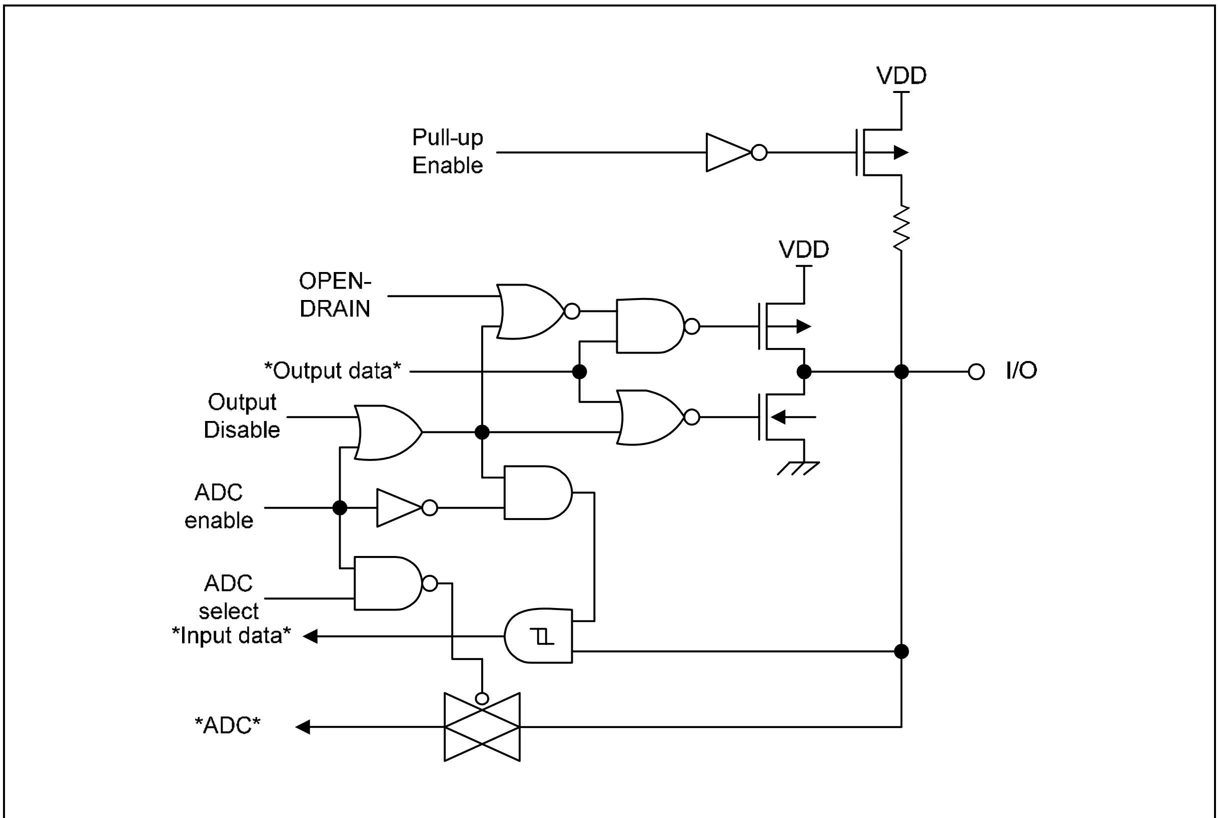


R03/RESETB



Input/Output data	Clock	ADC
R00	Xin	AN0
R01	Xout	AN1

R00/Xin, R01/Xout



Input/Output data	Input data	Output data	ADC
R02	-	-	AN2
R04	EXT0	PWM2O/T2O	AN4
R05	-	-	AN5
R06	EXT1/EC2	-	AN6
R07	EXT2/EC3	-	AN7/Vref

7. ELECTRICAL CHARACTERISTICS

7.1 Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit	Note
Supply Voltage	VDD	-0.3 – +6.0	V	–
Normal Voltage Pin	VI	-0.3 – VDD+0.3	V	Voltage on any pin with respect to Vss
	VO	-0.3 – VDD+0.3	V	
	IOH	-10	mA	Maximum current output sourced by (IOH per I/O pin)
	ΣIOH	-80	mA	Maximum current (ΣIOH)
	IOL	20	mA	Maximum current sunk by (IOL per I/O pin)
	ΣIOL	160	mA	Maximum current (ΣIOL)
Total Power Dissipation	fXIN	600	mW	–
Storage Temperature	TSTG	-65 – +150	°C	–

Note : Stresses above 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 above 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.

7.2 Recommended Operating Conditions

(T_A = - 40 °C to + 85°C)

Parameter	Symbol	Conditions	Min	Max	Units
		fx = 1.0 – 4.2MHz	2.2	5.5	
		fx = 1.0 – 8.0MHz	2.7	5.5	
		fx = 1.0 – 12.0MHz	4.0	5.5	
Operating Temperature	TOPR	VDD = 2.2 – 5.5V	-40	85	°C

7.3 A/D Converter Characteristics

(T$_A$ = - 40 °C to + 85°C, V$_{ref}$ = 2.7 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
A/D converting Resolution	–	–	–	12	–	bits
Integral Linearity Error	ILE	Vref = 5.12V, V$_{SS}$ = 0V, T$_A$ = + 25 °C	–	–	± 3	LSB
Differential Linearity Error	DLE		–	–	± 2	
Offset Error of Top	EOT		–	± 1	± 3	
Offset Error of Bottom	EOB		–	± 1	± 3	
Overall Accuracy	–		–	± 3	± 5	
Conversion time	t$_{CONV}$	–	25	–	–	µs
Analog input voltage	VAIN	–	V$_{SS}$	–	Vref	V
Analog Reference Voltage	Vref	–	2.7	–	5.5	V
Analog input current	I$_{AIN}$	VDD = Vref = 5V	–	–	10	µA
Analog block current	IAVDD	VDD = Vref = 5V VDD = Vref = 3V	– –	1 0.5	3 1.5	mA
		VDD = Vref = 5V Power down mode	–	100	500	nA
BGR	-	VDD = 5v, T$_A$ = + 25 °C	-	1.67	-	V
	-	VDD = 4v, T$_A$ = + 25 °C	-	1.63	-	
	-	VDD = 3v, T$_A$ = + 25 °C	-	1.62	-	

7.4 DC Electrical Characteristics

(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 – 5.5V, Vss=0V, f_{XIN}=12MHz)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Input High Voltage	VIH1	R0, V_{DD} = 4.5V – 5.5V	0.8VDD	–	VDD+0.3	V
	VIH2	Xin, Xout V_{DD} = 4.5V – 5.5V	0.8VDD	–	VDD+0.3	
Input Low Voltage	VIL1	R0, V_{DD} = 4.5V – 5.5V	– 0.3	–	0.2VDD	V
	VIL2	Xin, Xout V_{DD} = 4.5V – 5.5V	– 0.3	–	0.2VDD	
Output High Voltage	VOH	All output ports IOH = – 2mA V_{DD} = 4.5V – 5.5V	VDD-1.0	–	–	V
Output Low Voltage	VOL	All output ports IOL=15mA V_{DD} = 4.5V – 5.5V	–	–	2.0	V
Input high leakage current	IIH	R0x – R3x, Vin=VDD	–	–	1	uA
Input low leakage current	IIL	R0x – R3x, Vin=Vss	- 1	–	–	uA
Pull-up resistor	RPU	VI=0V, TA=25°C, R0 except R03 VDD=5V	25	50	100	kΩ
		VI=0V, TA=25°C, R0 except R03 VDD=3V	50	100	200	
OSC feedback resistor	RX	Xin=VDD, Xout=VSS TA=25°C, VDD=5V	350	700	1500	MΩ
Supply current	IDD1	Active mode, fx=12MHz, VDD=5V±10% Crystal oscillator	–	8.0	15.0	mA
		fx=8MHz, VDD=3V±10%	–	3.0	6.0	
	ISLEEP	Sleep mode, fx=12MHz, VDD=5V±10% Crystal oscillator	–	2.0	4.0	mA
		fx=8MHz, VDD=3V±10%	–	1.0	2.0	

	ISTOP	Stop mode VDD=5.5V, TA=25°C	–	0.5	5.0	uA
POR level			1.82		2.1	v

7.5 Input/Output Capacitance

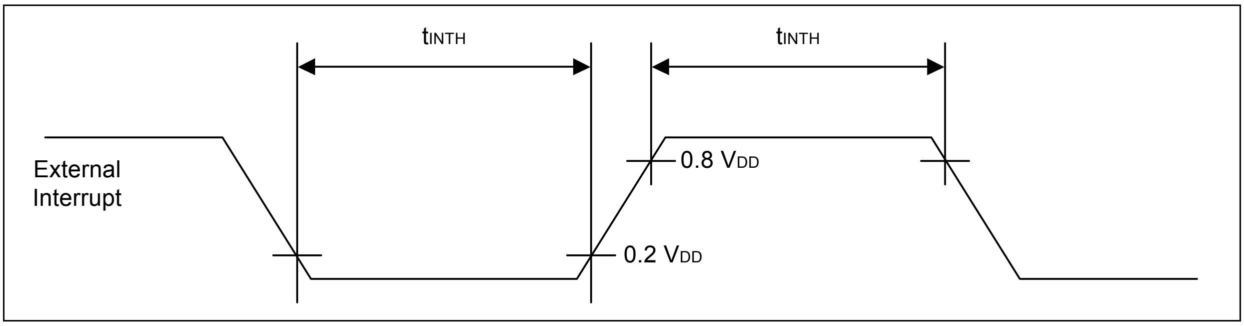
(T_A = - 40 °C to + 85°C, V_{DD} = 0 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Input Capacitance	CIN	f=1MHz Unmeasured pins are connected Vss	–	–	10	pF
Output Capacitance	COUT					
I/O Capacitance	CIO					

7.6 Serial Electric Characteristics

(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Interrupt input, high, low width	t_{INTH}, t_{INTL}	All interrupt, V_{DD} = 5 V	200	–	–	nS
RESETB input low width	t_{RSL}	Input, V_{DD} = 5 V	10	–	–	uS



7-1 Input Timing for External Interrupt

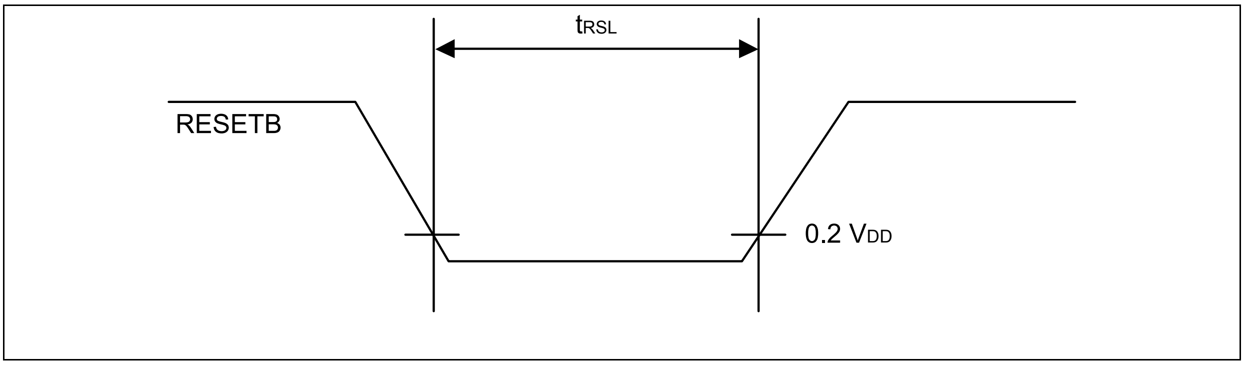
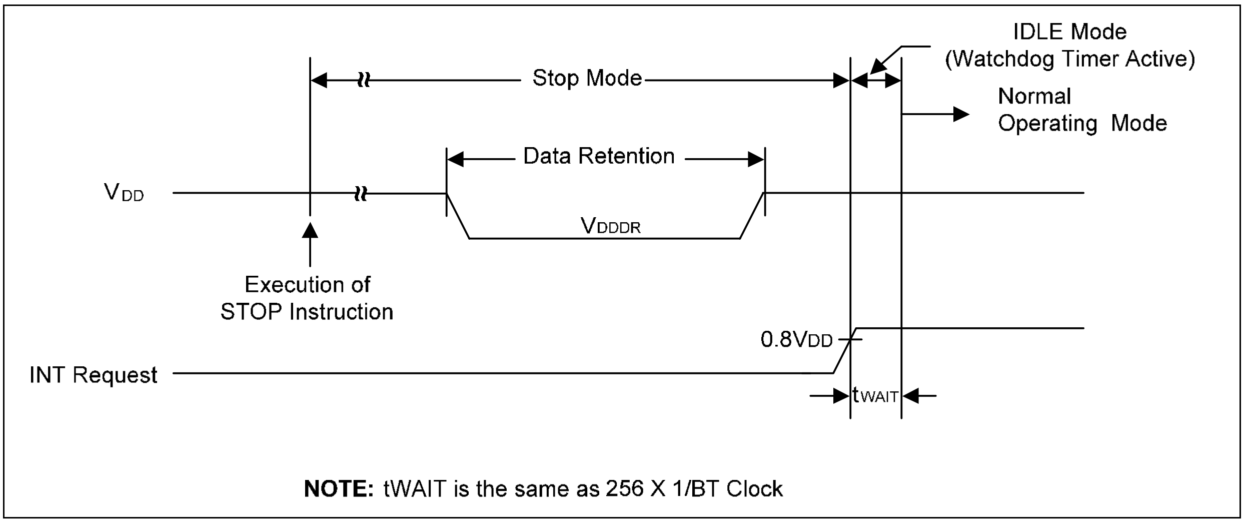


Figure 7-2 Input Timing for RESETB

7.7 Data Retention Voltage in Stop Mode

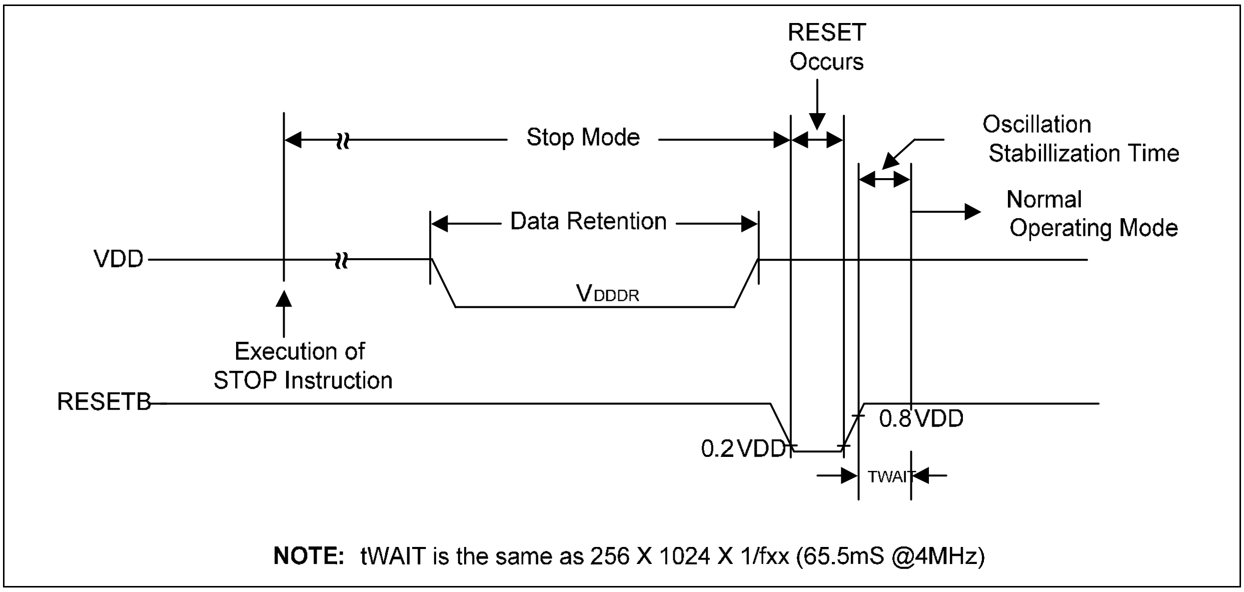
(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
Data retention supply voltage	V_{DDDR}	–	2.2	–	5.5	V
Data retention supply current	I_{DDDR}	V_{DDDR} = 2.2V (T_A = 25 °C), Stop mode	–	–	1	uA



NOTE: tWAIT is the same as 256 X 1/BT Clock

Figure 7-3 Stop Mode Release Timing When Initiated by an Interrupt



NOTE: tWAIT is the same as 256 X 1024 X 1/fxx (65.5mS @4MHz)

Figure 7-4 Stop Mode Release Timing When Initiated by RESETB

7.8 LVR (Low Voltage Reset) Electrical Characteristics

(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
LVR voltage	VLVR	–	2.2	2.4	2.6	V
			2.5	2.7	2.9	
			2.7	3.0	3.3	
			3.6	4.0	4.4	
Hysteresis voltage of LVR	∆V	–	–	10	100	mV
Current consumption	ILVR	VDD = 3V	–	45	80	uA

NOTES:

1. The current of LVR circuit is consumed when LVR is enabled by “ROM Option”.
2. 2^{16}/fx (= 6.55 ms at fx = 10 MHz)

7.9 POR (Power on Reset) Electrical Characteristics

(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 V to 5.5 V)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
POR voltage	VPOR	–	1.4	1.7	2.0	V
VDD Voltage Rising Time	tR	–	0.05	–	16.7	V/mS
Current consumption	IPOR	VDD = 3V	–	45	80	uA

7.10 Main clock Oscillator Characteristics

(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 V to 5.5 V)

Oscillator	Parameter	Conditions	Min	Typ.	Max	Units
Crystal	Main oscillation frequency	2.2 V – 5.5 V	1.0	–	4.2	MHz
		2.7 V – 5.5 V	1.0	–	8.0	
		4.0 V – 5.5 V	1.0	–	12.0	
Ceramic Oscillator	Main oscillation frequency	2.2 V – 5.5 V	1.0	–	4.2	MHz
		2.7 V – 5.5 V	1.0	–	8.0	
		4.0 V – 5.5 V	1.0	–	12.0	

External Clock	X_{IN} input frequency	2.2 V – 5.5 V	1.0	–	4.2	MHz
		2.7 V – 5.5 V	1.0	–	8.0	
		4.0 V – 5.5 V	1.0	–	12.0	

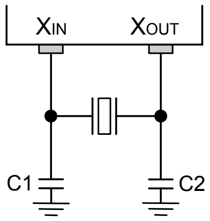


Figure 7-5 Crystal/Ceramic Oscillator

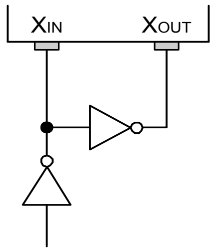


Figure 7-6 External Clock

7.11 External RC Oscillation Characteristics

(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 V to 5.0 V)

Parameter	Symbol	Conditions	Min	Typ.	Max	Units
RC oscillator frequency Range [1]	fERC	T_A = 25 °C	1	–	8	MHz
Accuracy of RC Oscillation [2]	ACCERC	V_{DD} =5.0V, T_A = 25 °C	– 6	–	+ 6	%
		V_{DD} =5.0V, T_A = – 10 °C to + 70 °C	– 12	–	+ 12	
RC oscillator setup time [3]	tSUERC	T_A = 25 °C	–	–	10	mS

NOTES:

1. The external resistor is connected between V_{DD} and X_{IN} pin and the 270pF capacitor is connected between X_{IN} and V_{SS} pin. (X_{OUT} pin can be used as a normal port). The frequency is adjusted by external resistor.
2. The min/max frequencies are within the range of RC OSC frequency (1MHz to 8MHz)
3. Data based on characterization results, not tested in production

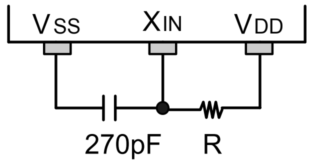


Figure 7-7 External Clock

7.12 Internal RC Oscillation Characteristics

(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 V to 5.0 V)

Parameter	Symbol	Conditions	Min	Typ.	Max	Units
RC oscillator frequency (1)	fIRC	V_{DD} =5.0V, T_A = 25 °C	-4%	8.0	4%	MHz
		V_{DD} =5.0V, T_A = – 40 °C to + 85 °C	-20%	8.0	20%	
Clock duty ratio	TOD	–	40	50	60	%
RC oscillator setup time (2)	tSUIRC	T_A = 25 °C	–	–	10	mS

NOTES:

1. Data based on characterization results, not tested in production
2. X_{IN} and X_{OUT} pins can be used as I/O ports.

7.13 Main Oscillation Stabilization Time

(T_A = - 40 °C to + 85°C, V_{DD} = 2.2 V to 5.5 V)

Oscillator	Conditions	Min	Typ.	Max	Units
Crystal	fx > 1 MHz Oscillation stabilization occurs when V_{DD} is equal to the minimum oscillator voltage range.	–	–	60	mS
Ceramic		–	–	10	mS
External Clock	X_{IN} input high and low width (t_{XH}, t_{XL})	40.0	–	480	nS

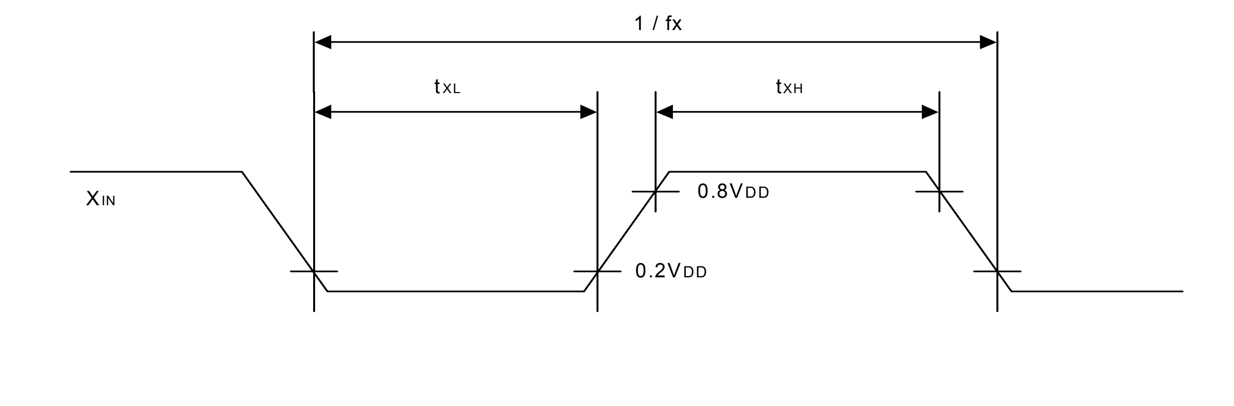


Figure 7-8 Clock Timing Measurement at XIN

7.14 Operating Voltage Range

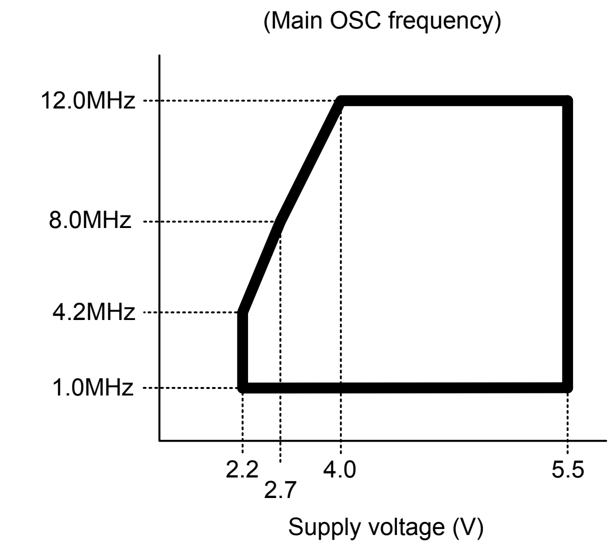


Figure 7-9 Operating Voltage Range

7.15 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.

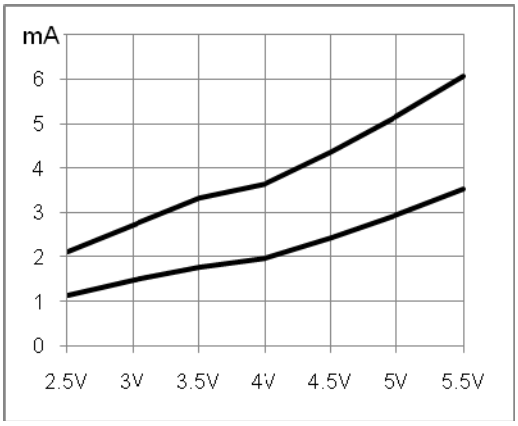


Figure 7-10 I_{DD} – V_{DD} in Normal Mode

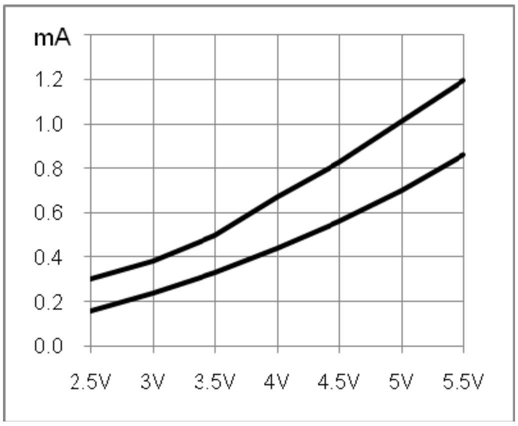


Figure 7-11 I_{SLEEP} – V_{DD} in Sleep Mode

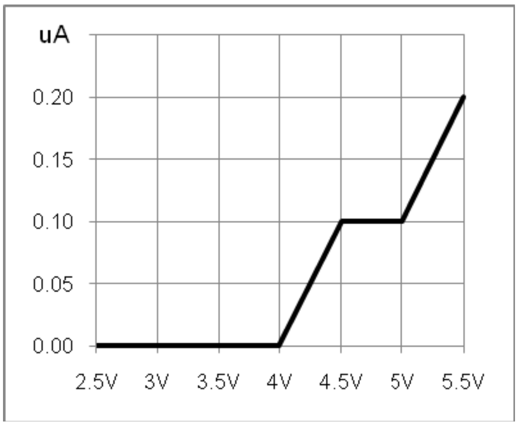


Figure 7-12 I_{STOP} – V_{DD} in STOP Mode

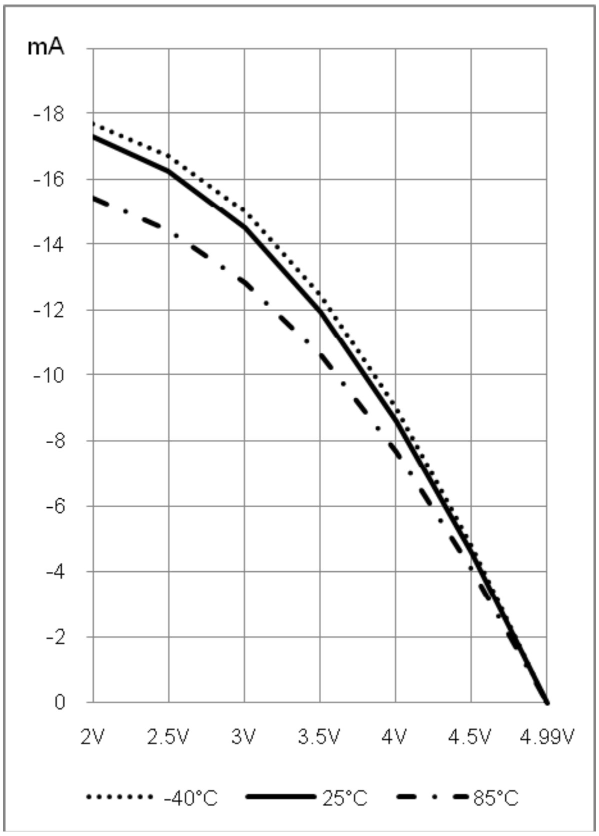


Figure 7-13 I_{OH} - V_{OH} at V_{DD}=5v

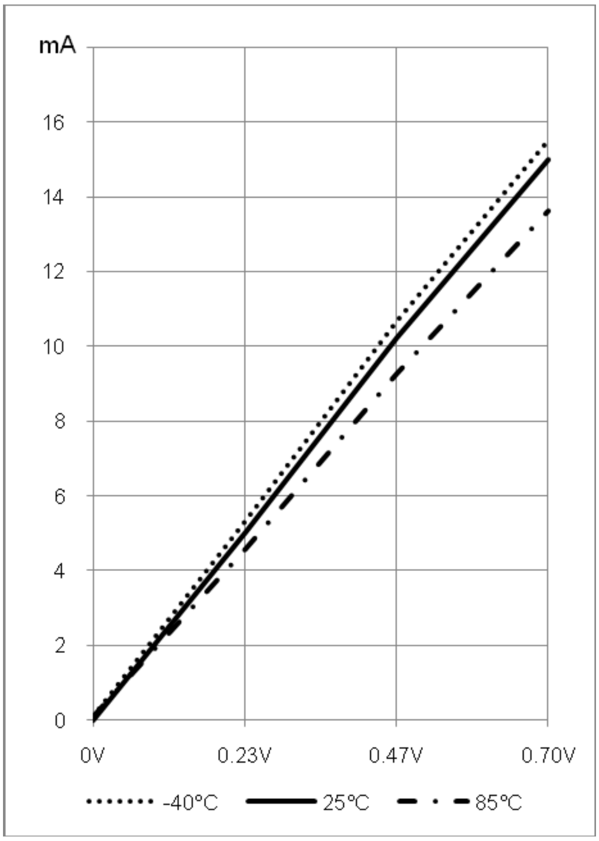


Figure 7-14 I_{OL} - V_{OL} at V_{DD}=5v

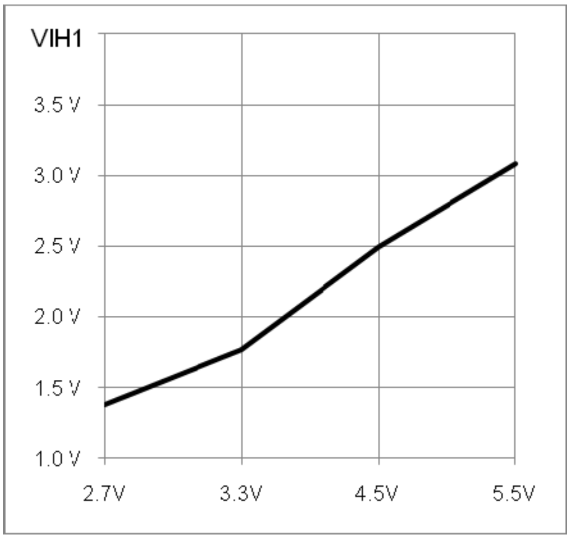


Figure 7-15 V_{IH1} - V_{DD}

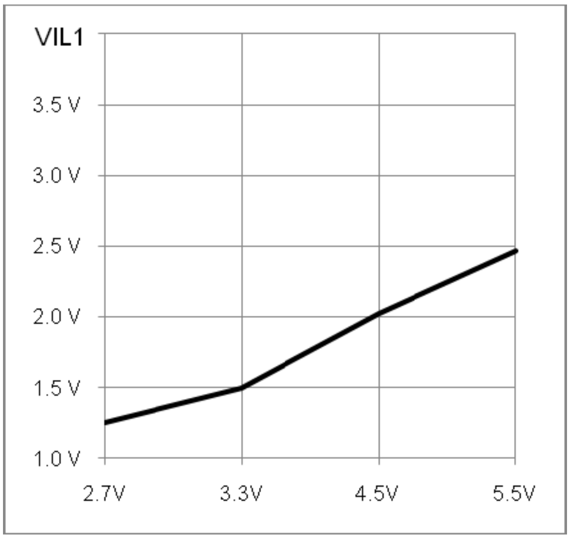


Figure 7-16 V_{IL1} - V_{DD}

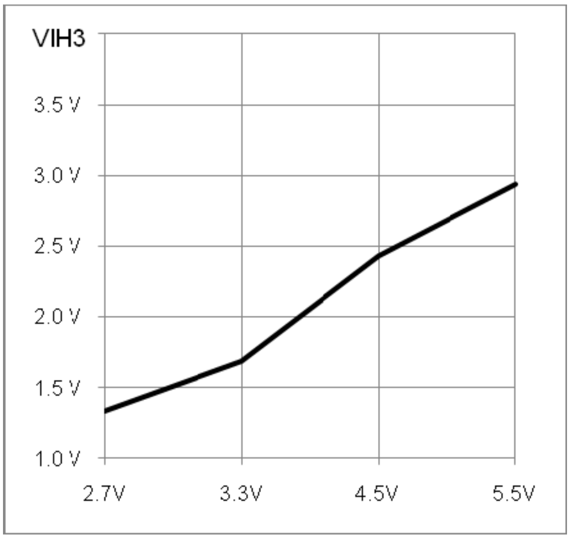


Figure 7-17 V_{IH2} - V_{DD}

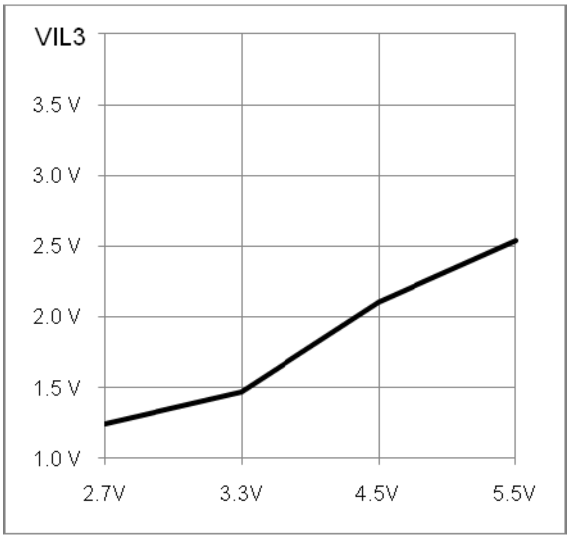


Figure 7-18 V_{IL2} - V_{DD}

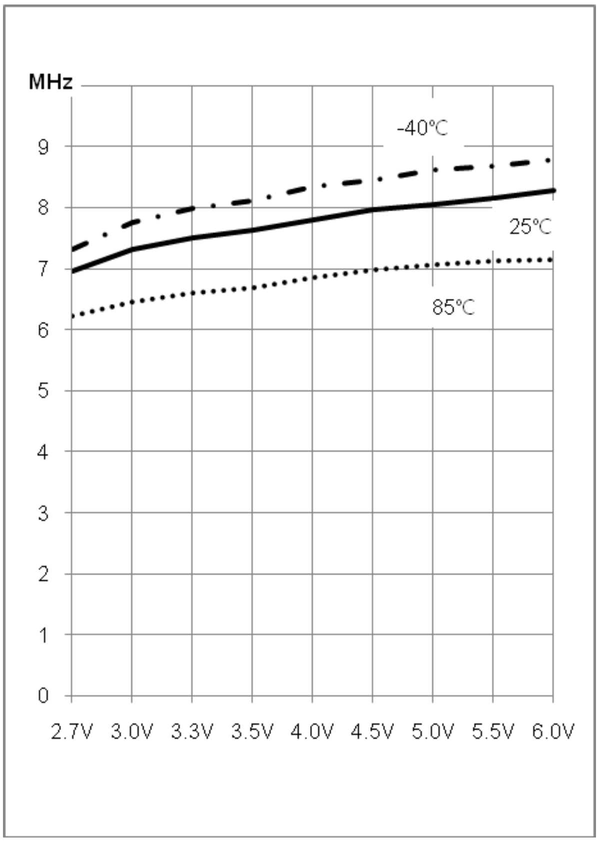


Figure 7-19 8MHz Internal OSC Freq. - V_{DD}

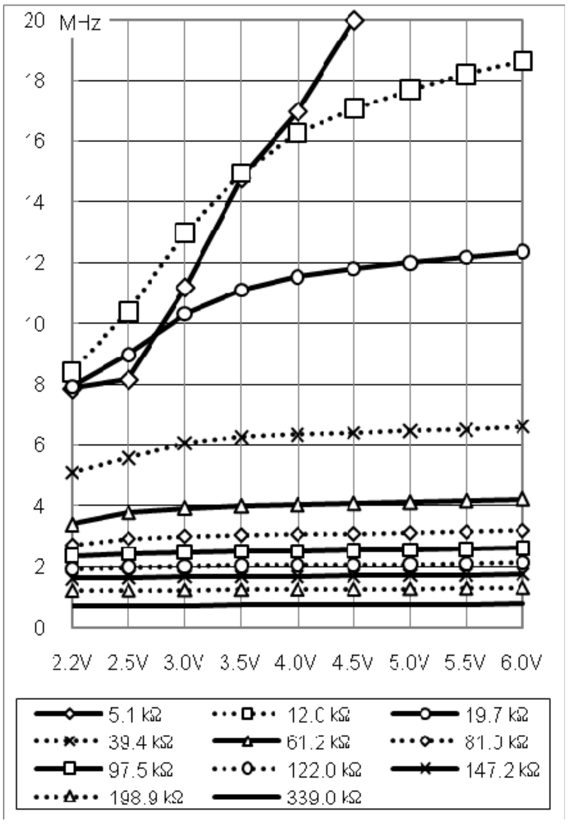


Figure 7-20 Ext. R/C OSC Freq. - V_{DD} at 25℃

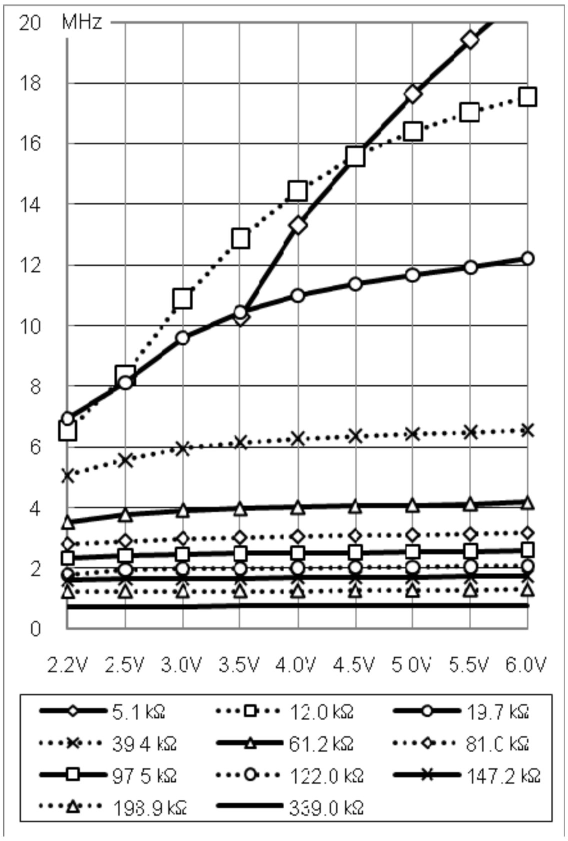


Figure 7-21 Ext. R/C OSC Freq. - V_{DD} at 85℃

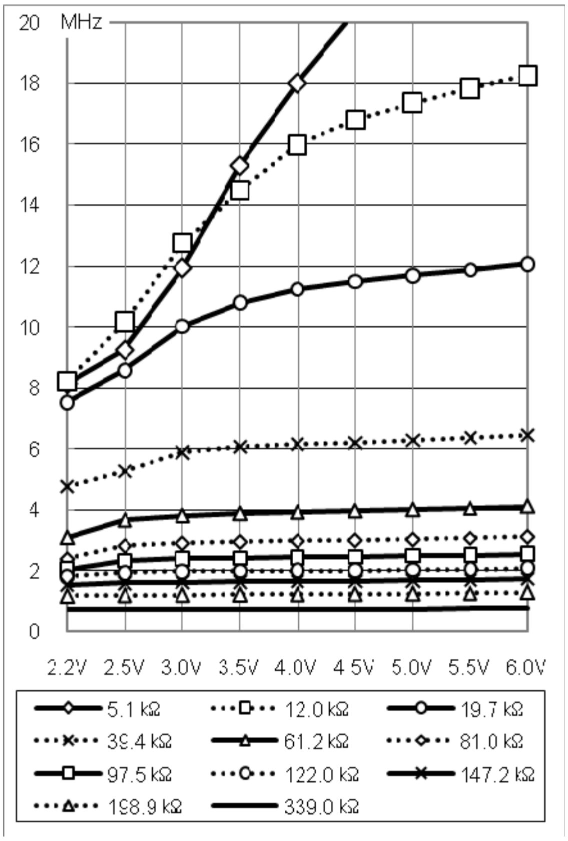


Figure 7-22 Ext.l R/C OSC Freq. - V_{DD} at -40℃

8. ROM OPTION

The ROM Option is a start-condition byte of the chip. The default ROM Option value is 00H (LVR enable and External RC is selected). It can be changed by appropriate writing tools such as PGMPlusUSB, ISP, etc.

8.1 Rom Option

	7	6	5	4	3	2	1	0
ROM OPTION	LVREN	LVRS		–	–	OSCS		

LVREN	LVR Enable/Disable bit	0: Enable (R03) 1: Disable (RESETB)
LVRS	LVR Level Selection bits	00: 2.4V 01: 2.7V 10: 3.0V 11: 4.0V
–	bit4 – bit3	Not used MC81F4104
OSCS	Oscillator Selection bits	000: External RC 001: Internal RC; 4MHz 010: Internal RC; 2MHz 011: Internal RC; 1MHz 100: Internal RC; 8MHz 101: Not available (Note 4) 110: Not available (Note 5) 111: Crystal/ceramic oscillator

Note :

1. When LVR is enabled, LVR level should be set to appropriate value, not default value.
2. When you select the Crystal/ceramic oscillator, R33 and R34 pins are automatically selected for XIN and XOUT mode.
3. When you select the external RC, R34 pin is automatically selected for XIN mode.
4. If OSCS is set by ‘101’, Oscillator works as ‘Internal RC; 4MHz’ mode.
5. If OSCS is set by ‘110’, Oscillator works as ‘Internal RC; 2MHz’ mode.

8.2 Read Timing

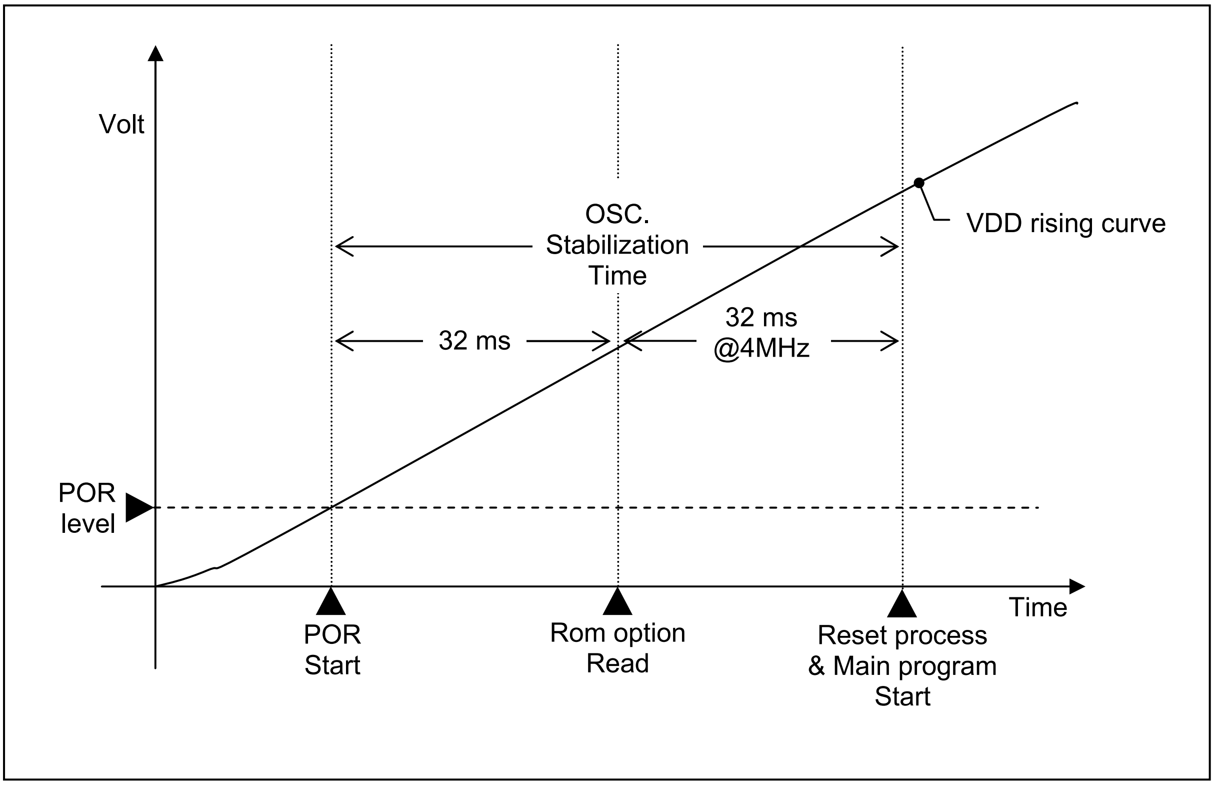


Figure 8-1 ROM option read timing diagram

Rom option is affected 32 mili-second (typically) after VDD cross the POR level. More precisely saying, the 32 mili-second is the time for 1/2 counting of 1024 divided BIT with 4 MHz internal OSC. After the ROM option is affected, system clock source is changed based on the ROM option. And then, rest 1/2 counting is continued with changed clock source. So, hole stabilization time is variable depend on the clock source.

	Before read ROM option	After read ROM option	OSC Stabilization Time
Formula	250ns x 128(BTCR) x 1024(divider)	Period x 128(BTCR) x 1024(divider)	Before + After
Int-RC 4MHz	32 ms	32 ms	64 ms
Int-RC 8MHz	32 ms	16 ms	48 ms
X-tal 12 MHz	32 ms	10.7 ms	42.7 ms
X-tal 16 Mhz	32 ms	8 ms	40 ms

Table 8-1 examples of OSC stabilization time

Note that ROM option is affected in OSC stabilization time. So even you change the ROM option by ISP. It is not affected until system is reset. In other words, you must reset the system after change the ROM option.

9. MEMORY ORGANIZATION

This MCU has separated address spaces for the *program memory* and the *data Memory*.

The program memory is a ROM which stores a program code. It is not possible to write a data at the program memory while the MCU is running.

The Data Memory is a REM which is used by MCU at running time.

9.1 Registers

There are few registers which are used for MCU operating.

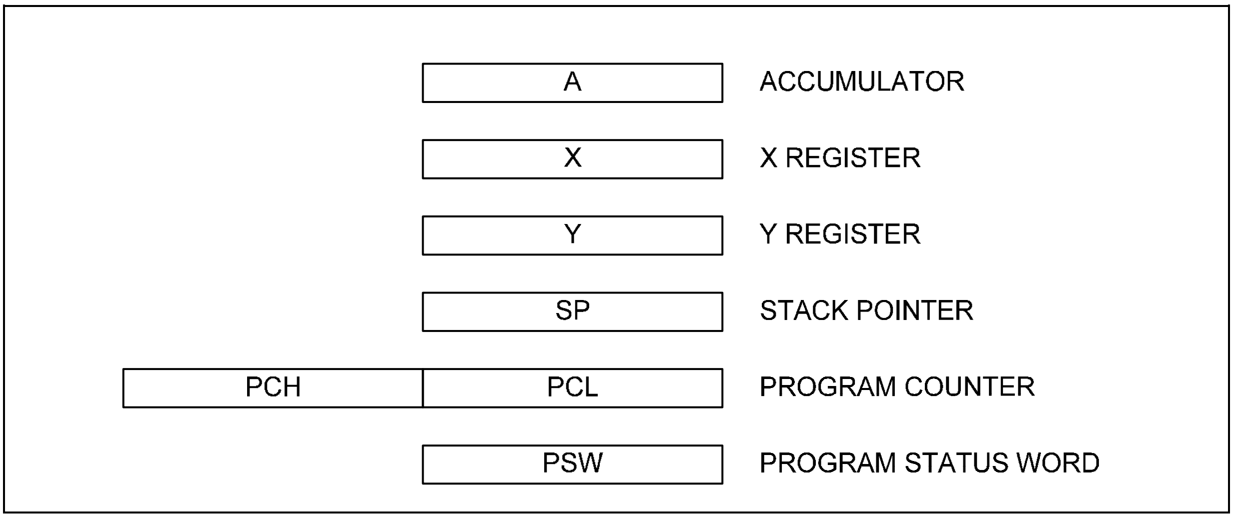
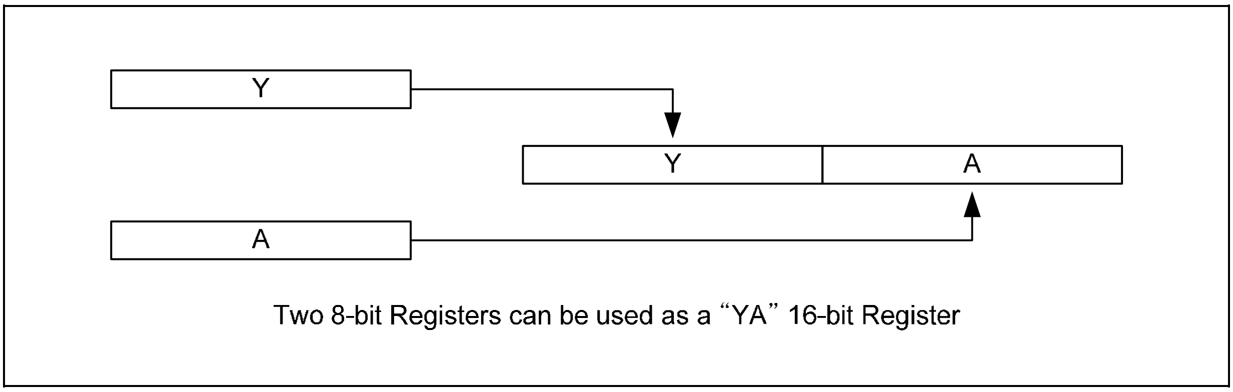


Figure 9-1 Configuration of Registers

Accumulator(A Register) : Accumulator is a 8-bit general purpose register, which is used for accumulating and some data operations such as transfer, temporary saving, and conditional judgment , etc.

And it can be used as a part of 16-bit register with Y Register as shown below.



Two 8-bit Registers can be used as a “YA” 16-bit Register

Figure 9-2 Configuration of YA 16-bit Registers

X, Y Registers: In the addressing mode, those are used as a index register. It makes it possible to access at Xth or Yth memory from specific address. It is extremely effective for referencing a subroutine table and a memory table.

These registers also have increment, decrement, comparison and data transfer functions, and they can be used as a simple accumulator.

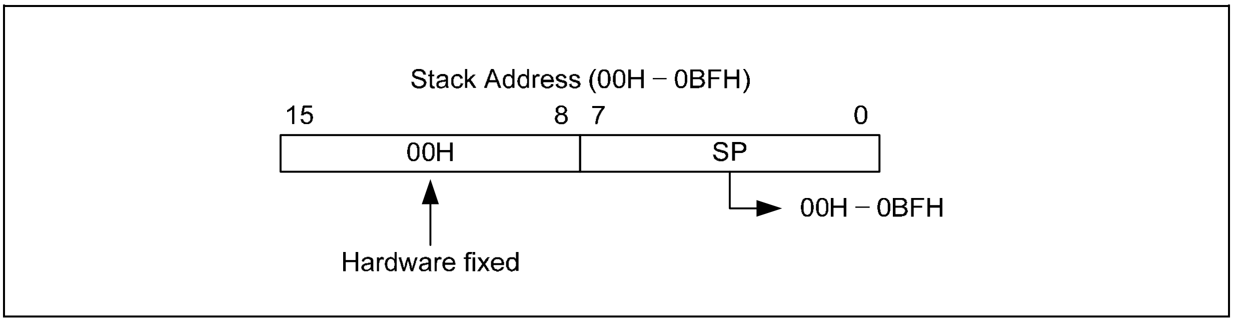


Figure 9-3 Stack Pointer

Stack Pointer: Stack Pointer is an 8-bit register which indicates the current 'push' point in the stack area. It is used to push and pop when interrupts or general function call is occurred. Stack Pointer identifies the location in the stack to be accessed (save or restore).

Generally, SP is automatically updated when a subroutine call is executed or an interrupt is accepted. However, if it is used in excess of the stack area permitted by the data memory allocating configuration, the user-processed data may be lost.

The stack can be located at any position within 00H to 0BFH of the internal data memory. The SP is not initialized by hardware, requiring to write the initial value (the location with which the use of the stack starts) by using the initialization routine. Normally, the initial value of "BFH" is used.

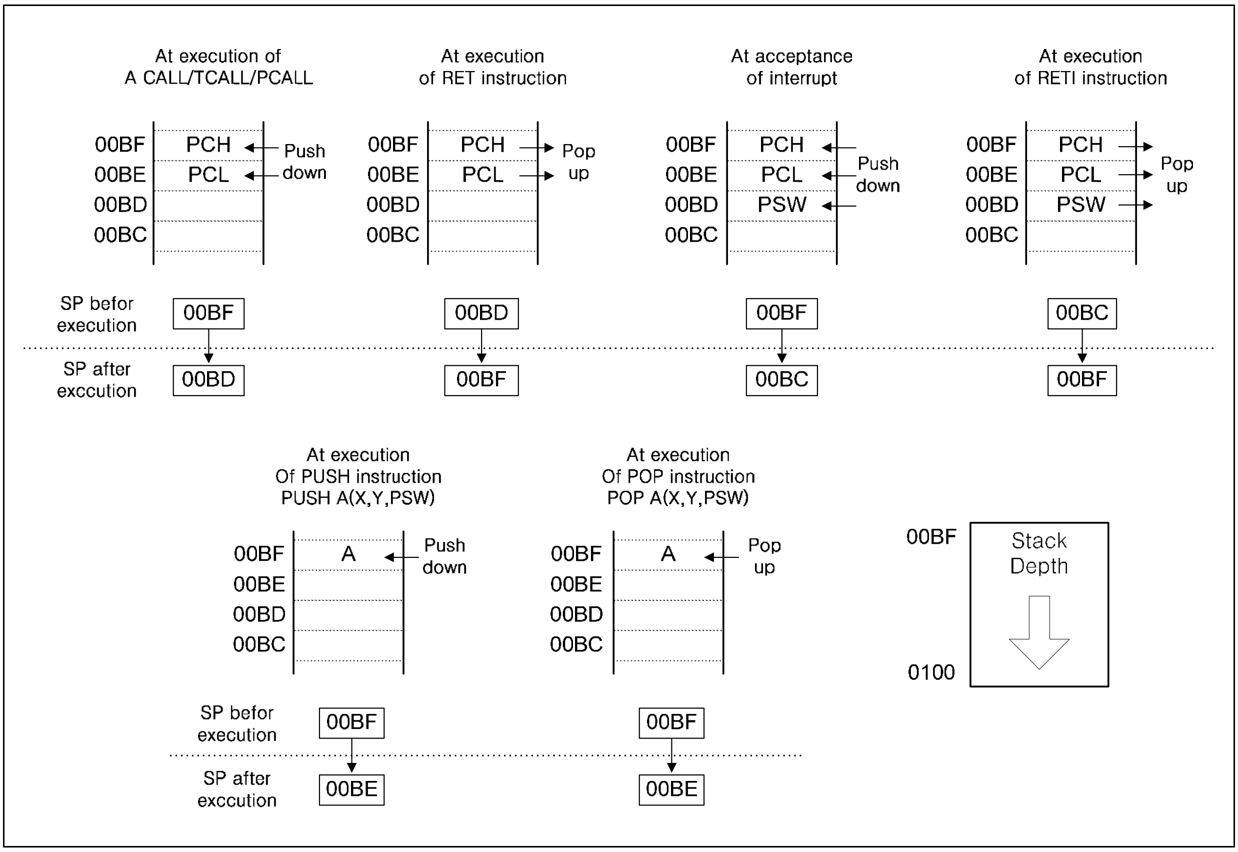


Figure 9-4 Stack Operation

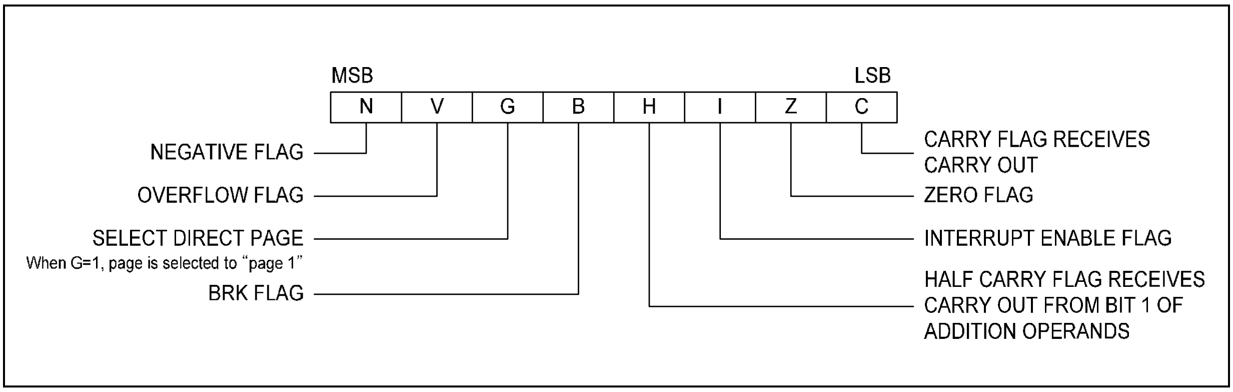

MSB
LSB
N
V
G
B
H
I
Z
C
NEGATIVE FLAG
OVERFLOW FLAG
SELECT DIRECT PAGE
When G=1, page is selected to "page 1"
BRK FLAG
CARRY FLAG RECEIVES CARRY OUT
ZERO FLAG
INTERRUPT ENABLE FLAG
HALF CARRY FLAG RECEIVES CARRY OUT FROM BIT 1 OF ADDITION OPERANDS

Figure 9-5 PSW (Program Status Word) Registers

Program Status Word: Program Status Word (PSW)contains several bits that reflect the current state of the CPU. It contains the Negative flag, the Overflow flag, the Break flag the Half Carry (for BCD operation), the Interrupt enable flag, the Zero flag, and the Carry flag.

[Carry flag C]

This flag stores any carry or borrow from the ALU of CPU after an arithmetic operation and is also changed by the Shift Instruction or Rotate Instruction.

[Zero flag Z]

This flag is set when the result of an arithmetic operation or data transfer is “0” and is cleared by any other result.

[Interrupt disable flag I]

This flag enables/disables all interrupts except interrupt caused by Reset or software BRK instruction. All interrupts are disabled when cleared to “0”. This flag immediately becomes “0” when an interrupt is served. It is set by the EI instruction and cleared by the DI instruction.

[Half carry flag H]

After operation, this is set when there is a carry from bit 3 of ALU or there is no borrow from bit 4 of ALU. This bit can not be set or cleared except CLRV instruction with Overflow flag (V).

[Break flag B]

This flag is set by software BRK instruction to distinguish BRK from TCALL instruction with the same vector address.

[Direct page flag G]

This flag assigns RAM page for direct addressing mode. In the direct addressing mode, addressing area is from zero page 00H to 0FFH when this flag is "0". If it is set to "1", addressing area is assigned 100H to 1FFH. It is set by SETG instruction and cleared by CLRG.

[Overflow flag V]

This flag is set to “1” when an overflow occurs as the result of an arithmetic operation involving signs. An overflow occurs when the result of an addition or subtraction exceeds +127(7FH) or -128(80H). The CLRV instruction clears the overflow flag. There is no set instruction. When the BIT instruction is executed, bit 6 of memory is copied to this flag.

[Negative flag N]

This flag is set to match the sign bit (bit 7) status of the result of a data or arithmetic operation. When the BIT instruction is executed, bit 7 of memory is copied to this flag.

9.2 Program Memory

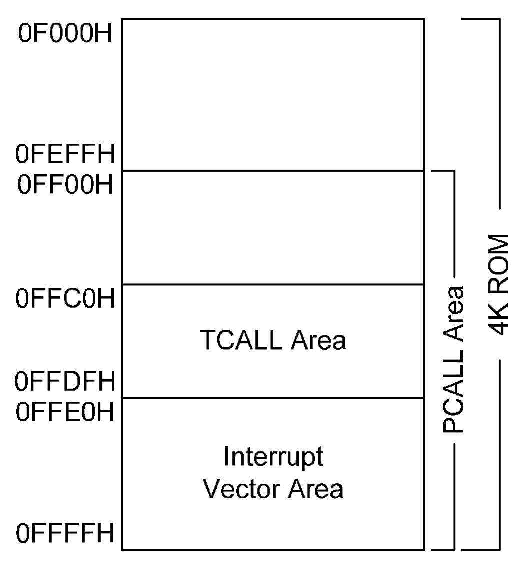


Figure 9-6 Program Memory Map

A 16-bit program counter is capable of addressing up to 64K bytes, but this device has 4k bytes program memory space only physically implemented. Accessing a location above FFFFH will cause a wrap-around to 0000H.

Figure 9-6 shows a map of Program Memory. After reset, the CPU begins execution from reset vector which is stored in address FFFEH and FFFFH. As shown in Figure 9-6, each area is assigned a fixed location in Program Memory.

Program memory area contains the user program Page Call (PCALL) area contains subroutine program to reduce program byte length by using 2 bytes PCALL instead of 3 bytes CALL instruction. If it is frequently called, it is more useful to save program byte length.

Table Call (TCALL) causes the CPU to jump to each TCALL address, where it commences the execution of the service routine. The Table Call service area spaces 2-byte for every TCALL: 0FFC0H for TCALL15, 0FFC2H for TCALL14, etc., as shown in Figure 9-7.

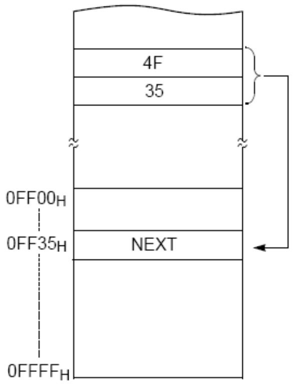
The interrupt causes the CPU to jump to specific location where it commences the execution of the service routine. The interrupt service locations spaces 2-byte interval. The External interrupt 0, for Example, is assigned to location 0FFFCH.

Any area from 0FF00H to 0FFFFH, if it is not going to be used, its service location is available as general purpose Program Memory.

Example :

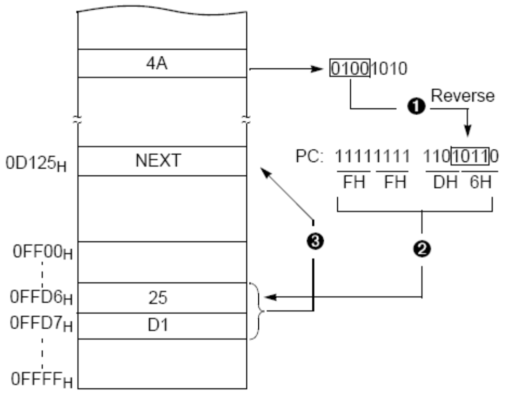
PCALL→ rel

```
4F35     PCALL  35H
```



TCALL→ n

```
4A       TCALL  4
```



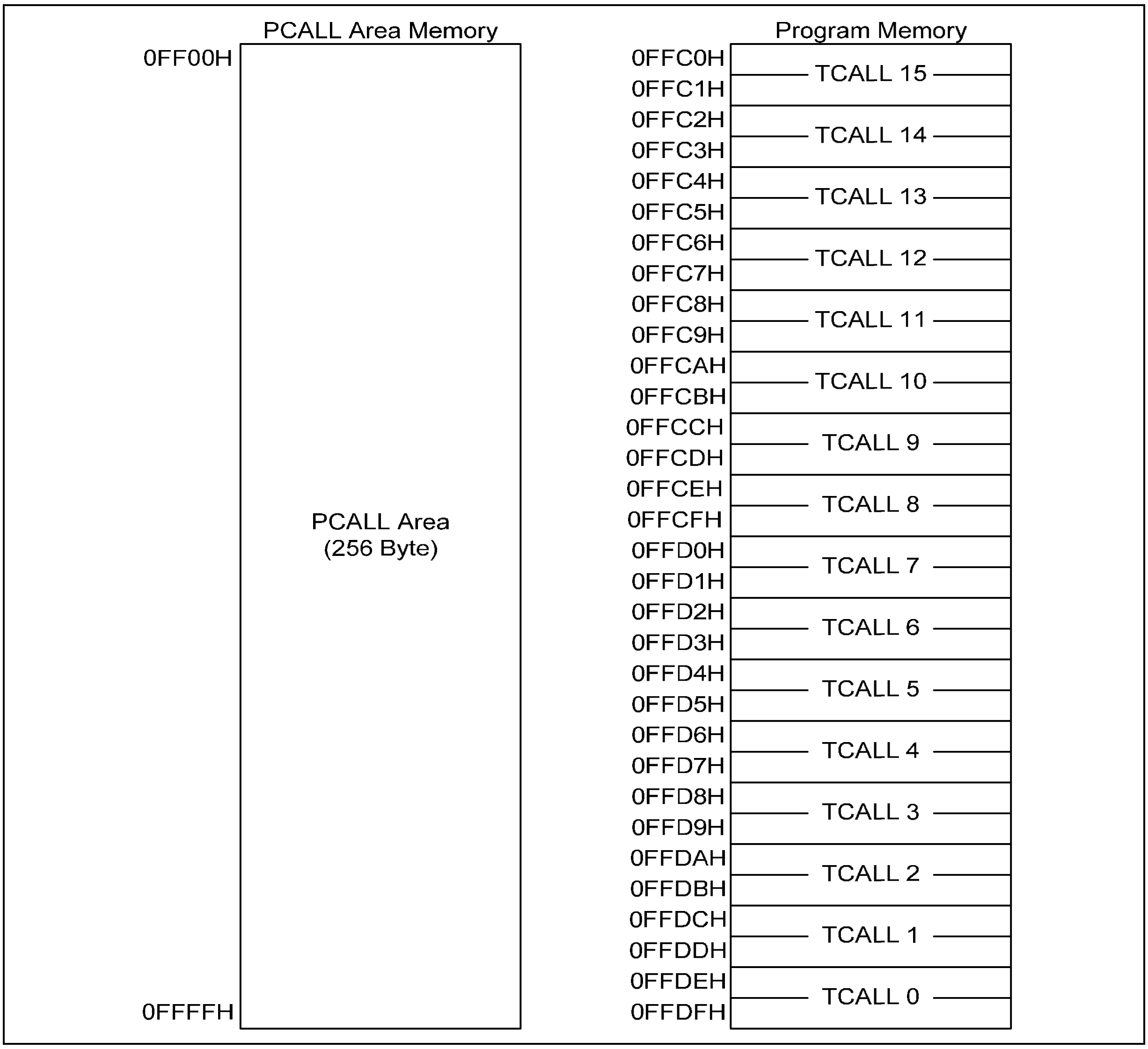


Figure 9-7 PCALL and TCALL Memory Area

Example : Usage of TCALL

```
LDA #5
TCALL 0FH               ;1BYTE INSTRUCTION
 :                      ;INSTEAD OF 3 BYTES
 :                      ;NORMAL CALL

;TABLE CALL ROUTINE

FUNC_A :  LDA LRG0
RET

FUNC_B : LDA LRG1
RET

;TABLE CALL ADD. AREA

ORG 0FFC0H               ;TCALL ADDRESS AREA
DW FUNC_A
DW FUNC_B
```

9.3 Data Memory

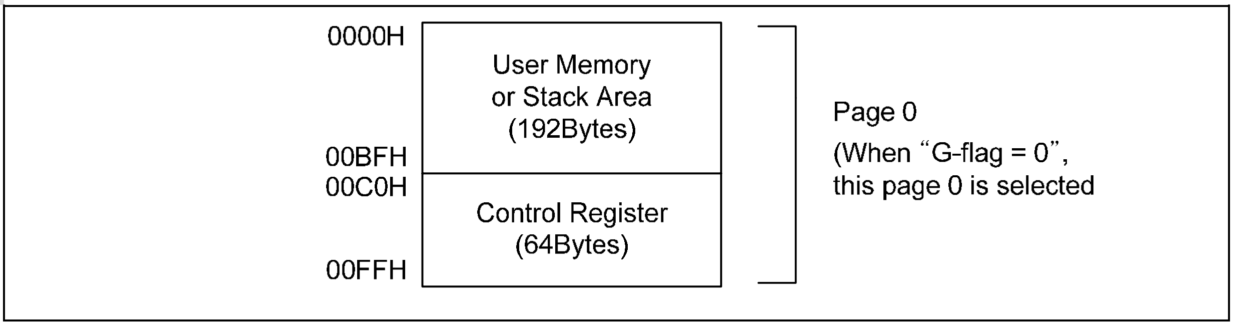


Figure 9-8 Data Memory Map

Figure 9-8 shows the internal Data Memory space available. Data Memory is divided into two groups, a user RAM/Stack memory and Control registers.

9.4 User Memory

The MC81F4104 has a 192 bytes user memory (RAM) including stack area. So it has only one memory page (page0).

9.5 Stack Area

The stack provides the area where the return address is saved before a jump is performed during the processing routine at the execution of a subroutine call instruction or the acceptance of an interrupt.

When returning from the processing routine, executing the subroutine return instruction [RET] restores the contents of the program counter from the stack; executing the interrupt return instruction [RETI] restores the contents of the program counter and flags.

The save/restore locations in the stack are determined by the stack pointed (SP). The SP is automatically decreased after the saving, and increased before the restoring. This means the value of the SP indicates the stack location number for the next save. Refer to Figure 9-4. .

9.6 Control Registers (SFR)

The control registers are used by the CPU and Peripheral function blocks for controlling the desired operation of the device. Therefore these registers contain control and status bits for the interrupt system, the timer/ counters, analog to digital converters and I/O ports. The control registers are in address range of 0C0H to 0FFH. It also be called by SFR(Special Function Registers).

Note that unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

More detailed information of each registers are explained in each peripheral section.

Note : Write only registers can not be accessed by bit manipulation instruction. Do not use read-modify-write instruction. Use byte manipulation instruction, for Example "LDM".

Example : To write at CKCTLR

```
LDM CKCTLR,#0AH ;Divide ratio(÷32)
```

Address	Register Name	Mnemonic	R/W	Initial value							
Hex											
00C0H	R0 Port Data Register	R0	R/W	0	0	1	0	0	1	0	0
00C6H	R0 Port Control Register High Byte	R0CONH	R/W	0	0	0	–	0	0	0	1

00C7H	R0 Port Control Register Middle Byte	R0CONM	R/W	0	0	0	–	–	–	0	0
00C8H	R0 Port Control Register Low Byte	R0CONL	R/W	–	–	0	1	0	0	0	0
00C9H	R0 Port Pull-up Resistor Enable Register	PUR0	R/W	0	0	0	0	–	0	0	0
00CAH	R0 Port External Interrupt Register	EINT0	R/W	–	–	0	0	0	0	0	0
00CCH	R0 Port External Interrupt Request Register	ERQ0	R/W	–	–	–	–	–	0	0	0
00D0H	Timer 2 Status And Control Register	T2SCR	R/W	–	–	0	0	0	0	0	0
00D1H	Timer 2 Data Register	T2DR	R/W	1	1	1	1	1	1	1	1
00D2H	Timer 2 Counter Register	T2CR	R	0	0	0	0	0	0	0	0
00D3H	Timer 3 Status And Control Register	T3SCR	R/W	–	–	0	0	0	0	0	0
00D4H	Timer 3 Data Register	T3DR	R/W	1	1	1	1	1	1	1	1
00D5H	Timer 3 Counter Register	T3CR	R	0	0	0	0	0	0	0	0
00DDH	A/D Mode Register	ADMR	R/W	0	0	0	0	0	0	0	0
00DEH	A/D Converter Data Register High Byte	ADDRH	R	X	X	X	X	X	X	X	X
00DFH	A/D Converter Data Register Low Byte	ADDRL	R	X	X	X	X	–	–	–	–
00E2H	PWM Status And Control Register	PWMSCR	R/W	0	0	–	–	–	–	–	–
00E3H	PWM Period And Duty Register	PWMPDR	R/W	–	–	–	–	1	1	1	1
00E6H	PWM2 Data Register	PWM2DR	R/W	1	1	1	1	1	1	1	1
00EAH	Interrupt Enable Register High Byte	IENH	R/W	–	–	–	–	0	0	0	0
00EBH	Interrupt Enable Register Low Byte	IENL	R/W	–	–	–	–	–	0	–	0
00ECH	Interrupt Request Register High Byte	IRQH	R/W	–	–	–	–	0	0	0	0
00EDH	Interrupt Request Register Low Byte	IRQL	R/W	–	–	–	–	–	0	–	0
00F1H	Basic Timer Counter Register	BTCR	R	X	X	X	X	X	X	X	X
00F2H	Clock control Register	CKCTLR	R/W	–	–	–	1	0	1	1	1
00F3H	Power On Reset Control Register	PORC	R/W	0	0	0	0	0	0	0	0
00F4H	Watchdog Timer Register	WDTR	R/W	0	1	1	1	1	1	1	1
00F5H	Stop & Sleep Mode Control Register	SSCR	R/W	0	0	0	0	0	0	0	0
00F6H	Watchdog Timer Status Register	WDTSR	R/W	0	0	0	0	0	0	0	0
00F7H	Watchdog Timer Counter Register	WDTCR	R	X	X	X	X	X	X	X	X

Table 9-1 Control Register 1/2

Mnemonic	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Hex								
R0	00C0H	R0 Port Data Register							
R0CONH	00C6H	R07			–	R06		R05	
R0CONM	00C7H	R04			–	–	–	R03	
R0CONL	00C8H	–	–	R02		R01		R00	
PUR0	00C9H	PUR07	PUR06	PUR05	PUR04	–	PUR02	PUR01	PUR00
EINT0	00CAH	–	–	EXT2IE		EXT1IE		EXT0IE	

ERQ0	00CCH	–	–	–	–	–	EXT2IR	EXT1IR	EXT0IR
T2SCR	00D0H	–	–	T2MS	T2CC	T2CS			
T2DR	00D1H	Timer 2 Data Register							
T2CR	00D2H	Timer 2 Counter Register							
T3SCR	00D3H	–	–	T3MS	T3CC	T3CS			
T3DR	00D4H	Timer 3 Data Register							
T3CR	00D5H	Timer 3 Counter Register							
ADMR	00DDH	SSBIT	EOC	ADCLK		ADCH			
ADDRH	00DEH	A/D Converter Data Register High Byte							
ADDRL	00DFH	A/D Converter Data Register Low Byte							
PWMSCR	00E2H	POL2	PWMS	–	–	–	–	–	–
PWMPDR	00E3H	–	–	–	–	P2DH	P2DL	PPH	PPL
PWM2DR	00E6H	PWM 2 Data Register							
IENH	00EAH	–	–	–	–	T2MIE	T2OVIE	T3MIE	T3OVIE
IENL	00EBH	–	–	–	–	–	WDTIE	–	BTIE
IRQH	00ECH	–	–	–	–	T2MIR	T2OVIR	T3MIR	T3OVIR
IRQL	00EDH	–	–	–	–	–	WDTIR	–	BTIR
BTCR	00F1H	Basic Timer Counter Register							
CKCTLR	00F2H	–	–	–	WDTON	BTCL	BTS		
PORC	00F3H	POREN							
WDTR	00F4H	WDTCL	WDTCMP						
SSCR	00F5H	Stop and Sleep Control Register							
WDTSR	00F6H	Watchdog Timer Status Register							
WDTCR	00F7H	Watchdog Timer Counter Register							

Table 9-2 Control Register 2/2

9.7 Addressing modes

The MC81Fxxx series MCU uses six addressing modes;

- Register Addressing
- Immediate Addressing
- Direct Page Addressing
- Absolute Addressing
- Indexed Addressing
- Indirect Addressing

Register Addressing

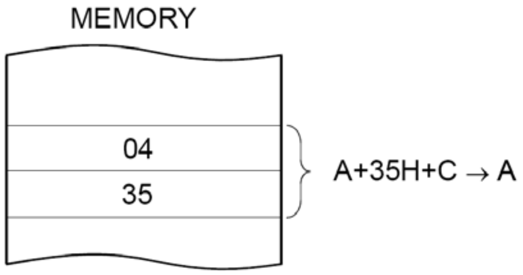
Register addressing means to access to the data of the A, X, Y, C and PSW registers. For Example ‘ASL (Arithmetic Shift Left)’ only accesses the A register.

Immediate Addressing

In this mode, second byte (operand) is accessed as a data immediately.

Example :

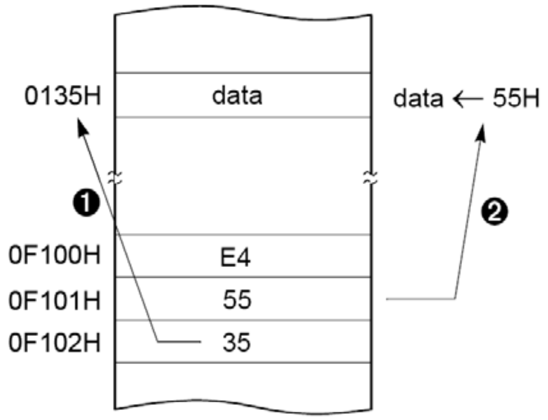
```
  :
ADC #35h   ;op code is 04h
  :
  :
```



When G-flag is 1, then RAM address is defined by 16-bit address which is composed of 8-bit RAM paging register (RPR) and 8-bit immediate data.

Example :

```
  :              ;When G = 1, RPR = 1
LDM #35h,#55h   ;op code is 0E4h
  :
  :
```

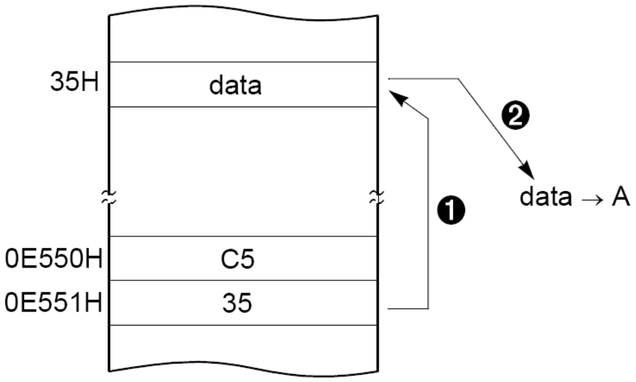


Direct Page Addressing -> dp

In this mode, an address is specified within direct page. Current accessed page is selected by RPR(RAM Page select Register). And dp(Direct Page) is an one byte data which indicates the target address in the current accessed page.

Example :

```
  :        ;When G = 0
LDA 35h    ;A = [35h]
  :        ;op code is 0C5h
  :
```



Absolute Addressing

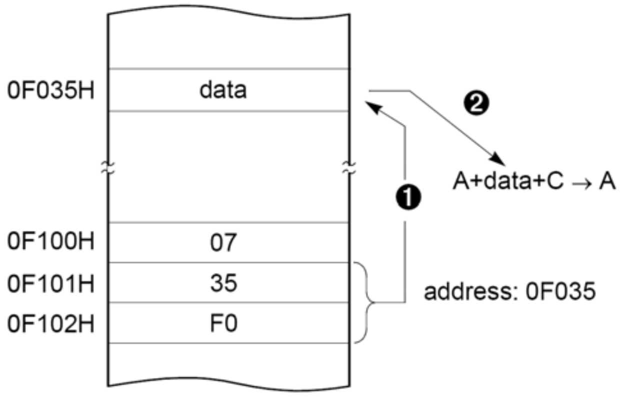
Absolute addressing sets corresponding memory data to Data, i.e. second byte (Operand I) of command becomes lower level address and third byte (Operand II) becomes upper level address. With 3 bytes command, it is possible to access to whole memory area.

ADC, AND, CMP, CMPX, CMPY, EOR, LDA, LDX,LDY, OR, SBC, STA, STX, STY

The operation within data memory (RAM) : ASL, BIT, DEC, INC, LSR, ROL, ROR

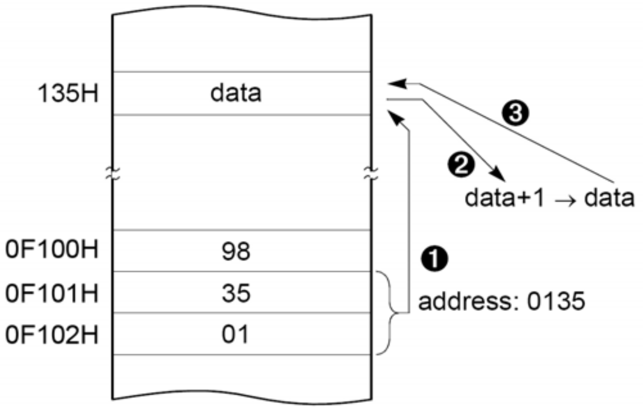
Example :

```
  :         ;When G = 0
ADC !0F035h ;A = A + C + ROM[0F035h]
  :         ;op code is 07h
  :
```



Example : Addressing accesses the address 0135H regardless of G-flag.

```
  :         ;When G = 0
INC !0135h  ;increase ROM[135h]
  :         ;op code is 98h
  :
```



Indexed Addressing

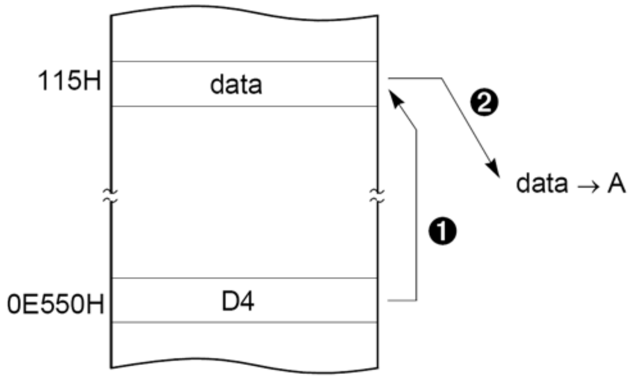
X indexed direct page (no offset) → {X}

In this mode, an address is specified by the X register.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA, XMA

Example :

```
    :          ;When G = 1, X = 15h
  LDA {X}      ;A = ROM[(RPR<<8) + X]
    :          ;op code is 0D4h
    :
```



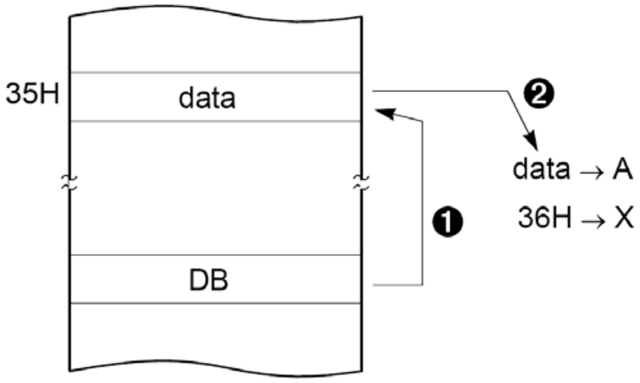
X indexed direct page, auto increment→ {X}+

In this mode, a address is specified within direct page by the X register and the content of X is increased by 1.

LDA, STA

Example:

```
    :          ;When G = 0, X = 35h
LDA {X}+    ;A = ROM[(RPR<<8) + X]
 :          ; and X = X + 1
    :          ;op code is 0DBh
    :
```



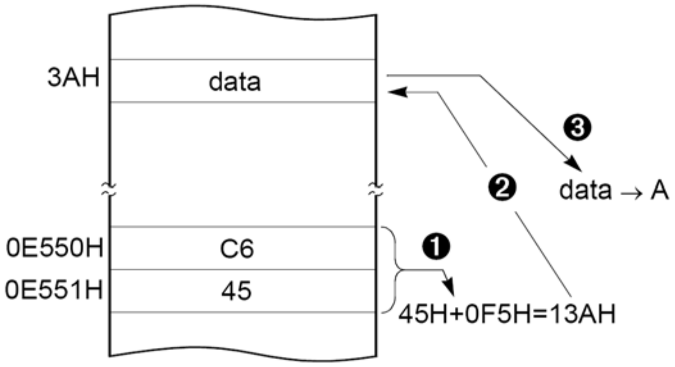
X indexed direct page (8 bit offset) → dp+X

This address value is the second byte (Operand) of command plus the data of X-register. And it assigns the memory in direct page.

ADC, AND, CMP, EOR, LDA, LDY, OR, SBC, STA,STY, XMA, ASL, DEC, INC, LSR, ROL, ROR

Example :

```
     :          ;When G = 0, X = 0F5h
LDA 45h + X ;op code is 0C6h
 :          ;
     :          ;
     :
```



Y indexed direct page (8 bit offset) → dp+Y

This address value is the second byte (Operand) of command plus the data of Y-register, which assigns Memory in Direct page.

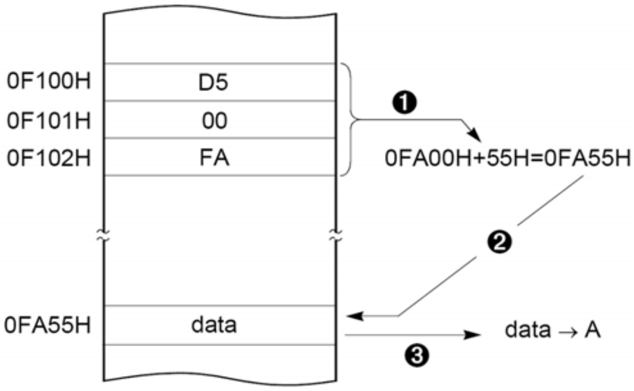
This is same with above ‘X indexed direct page’. Use Y register instead of X.

Y indexed absolute → !abs+Y

Accessing the value of 16-bit absolute address plus Y-register value. This addressing mode can specify memory in whole area.

Example :

```
         :              ;when Y = 55h
LDA !0FA00H+Y       ;op code is D5h
     :
```



Indirect Addressing

Direct page indirect → [dp]

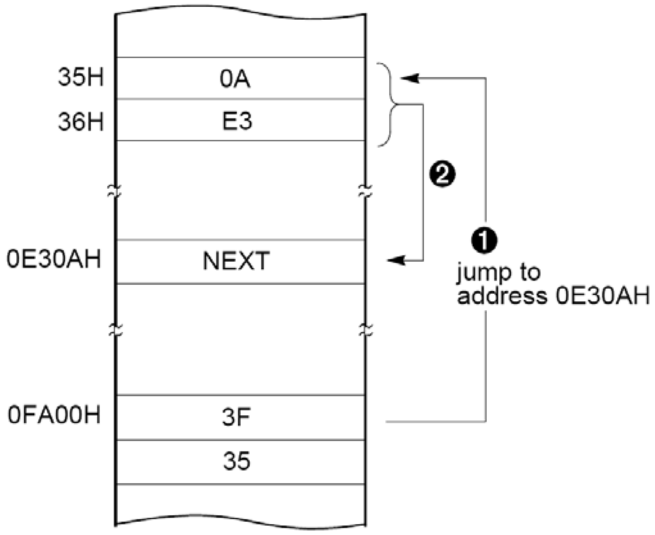
Assigns data address to use for accomplishing command which sets memory data (or pair memory) by Operand.

Also index can be used with Index register X,Y.

JMP, CALL

Example :

```
     :              ;when G = 0
JMP [35h]       ;op code is 3Fh
 :
```



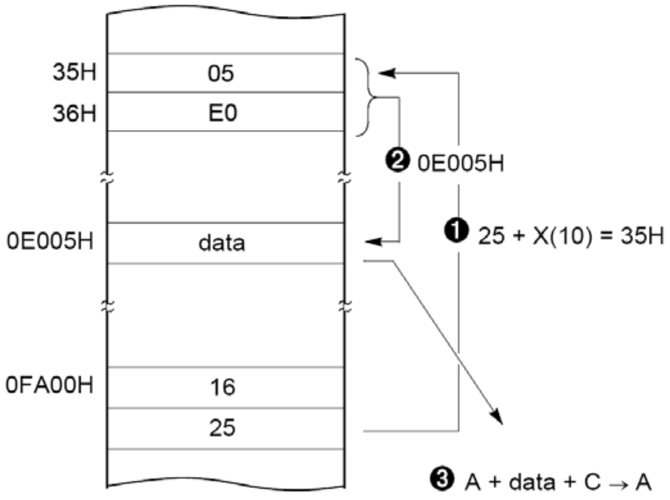
X indexed indirect → [dp+X]

Processes memory data as Data, assigned by 16-bit pair memory which i s determined by pair data [dp+X+1][dp+X] Operand plus X-register data in Direct page.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA

Example :

```
     :              ;when G = 0
     :              ;      X = 10h
ADC [25h + X]  ;op code is 16h
 :
```



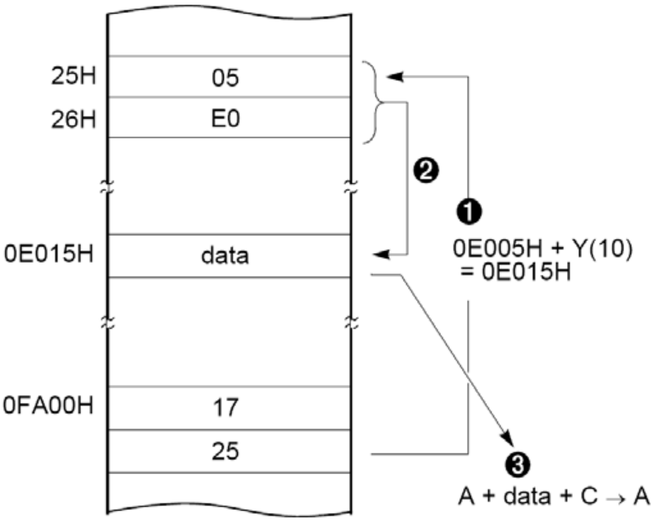
Y indexed indirect → [dp]+Y

Processes memory data as Data, assigned by the data [dp+1][dp] of 16-bit pair memory paired by Operand in Direct page plus Y-register data.

ADC, AND, CMP, EOR, LDA, OR, SBC, STA

Example :

```
     :              ;when G = 0
     :              ;      Y = 10h
ADC [25h + Y]  ;op code is 17h
 :
```



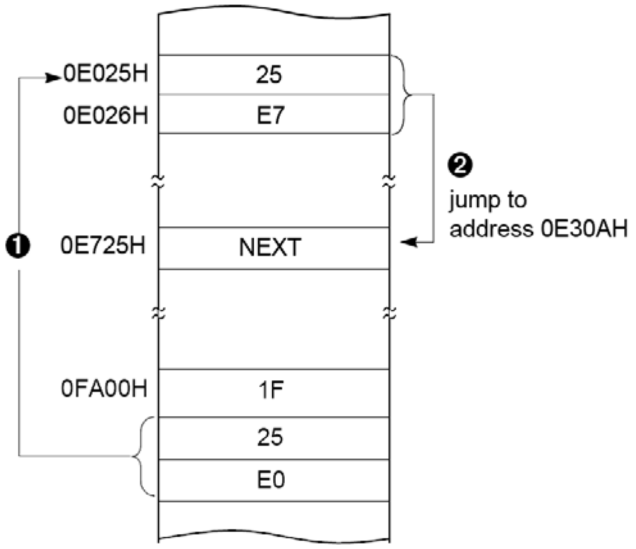
Absolute indirect → [!abs]

The program jumps to address specified by 16-bit absolute address.

JMP

Example :

```
     :              ;when G = 0
JMP [0E025h]    ;op code is 1Fh
 :
```



10. I/O PORTS

The MC81F4104 microcontroller has one I/O port, P0. The CPU accesses ports by writing or reading port register directly.

10.1 R0 Port Registers

R0CONH – R05~07

R0 PORT CONTROL HIGH REGISTER **00C6H**

When programming the port, please remember that any alternative peripheral I/O function that defined by the R0CONH register must also be enabled in the associated peripheral module.

	7	6	5	4	3	2	1	0	
R0CONH	R07			-	R06		R05		Reset value:
	R/W	R/W	R/W	R/W	R/W	R/W	RW	R/W	000-_0001b

R07	R07/AN7/Vref/EC3/EXT2	000: Schmitt trigger input mode (EC3/EXT2) 001: Output mode, open-drain 010: Alternative function (AN7) 011: Alternative function (Vref) 1xx: Output mode, push-pull
–	bit4	Not used for MC81F4104
R06	R06/AN6/EC2/EXT1	00: Schmitt trigger input mode (EC2/EXT1) 01: Output mode, open-drain 10: Alternative function (AN6) 11: Output mode, push-pull
R05	R05/AN5	00: Schmitt trigger input mode 01: Output mode, open-drain 10: Alternative function (AN5) 11: Output mode, push-pull

R0CONM – R03~04

R0 PORT CONTROL MIDDLE REGISTER **00C7H**

When programming the port, please remember that any alternative peripheral I/O function that defined by the R0CONM register must also be enabled in the associated peripheral module.

	7	6	5	4	3	2	1	0	
R0CONM	R04			-	-	-	R03		Reset value:
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	000-_--00b

R04	R04/AN4/PWM2O/T2O/EXT0	000: Schmitt trigger input mode (EXT0) 001: Output mode, open-drain 010: Alternative function (AN4) 011: Alternative function (PWM2O/T2O) 1xx: Output mode, push-pull
–	bit4 – bit2	Not used for MC81F4104
R03	R03/RESETB (*note*)	00: Schmitt trigger input mode 01: Output mode, open-drain 10: Not available 11: Not available

Note :

If you want to use RESETB, the LVREN (ROM OPTION [7]) must select to LVR disable mode (‘1’). If you want to use R35, the LVREN (ROM OPTION [7]) must select to LVR enable mode (‘0’).

Even you are in case of using emulator you must select the ROM OPTION switch properly to use those R03 ports.

R0CONL – R00~02

R0 PORT CONTROL LOW REGISTER **00C8H**

When programming the port, please remember that any alternative peripheral I/O function that defined by the R0CONL register must also be enabled in the associated peripheral module.

	7	6	5	4	3	2	1	0	
R0CONL	–	–	R02		R01		R00		Reset value:
	–	–	R/W	R/W	R/W	R/W	R/W	R/W	--01_0000b

–	bit7 – bit6	Not used for MC81F4104
R02	R02/AN2	00: Schmitt trigger input mode 01: Output mode, open-drain 10: Alternative function (AN2) 11: Output mode, push-pull
R01	R01/Xout/AN1 (*note*)	00: Schmitt trigger input mode 01: Output mode, open-drain 10: Alternative function (AN1) 11: Output mode, push-pull
R00	R00/Xin/AN0 (*note*)	00: Schmitt trigger input mode 01: Output mode, open-drain 10: Alternative function (AN0) 11: Output mode, push-pull

Note :

If you want to use X_{IN} and X_{OUT}, the OSCS (ROM OPTION [2:0]) must select to Crystal/ceramic oscillator mode (111b). If you want to use R00 and R01, the OSCS (ROM OPTION [2:0]) must select to Internal RC mode (001b, 010b, 011b, 100b).

Even you are in case of using emulator, you must select the OSC option as an internal RC mode to use R00 and R01 ports as general I/O ports.

PUR0

R0 PORT PULL-UP ENABLE REGISTER **00C9H**

Using the PUR0 register, you can configure pull-up resistors to individual R07-R00 pins.

	7	6	5	4	3	2	1	0	
PUR0	PUR07	PUR06	PUR05	PUR04	-	PUR02	PUR01	PUR00	Reset value: 00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

PUR07	R07 Pull-up Resistor Enable Bit	0: Disable pull-up resistor 1: Enable pull-up resistor
PUR06	R06 Pull-up Resistor Enable Bit	0: Disable pull-up resistor 1: Enable pull-up resistor
PUR05	R05 Pull-up Resistor Enable Bit	0: Disable pull-up resistor 1: Enable pull-up resistor
PUR04	R04 Pull-up Resistor Enable Bit	0: Disable pull-up resistor 1: Enable pull-up resistor
-	bit 3	Not used for MC81F4104
PUR02	R02 Pull-up Resistor Enable Bit	0: Disable pull-up resistor 1: Enable pull-up resistor
PUR01	R01 Pull-up Resistor Enable Bit	0: Disable pull-up resistor 1: Enable pull-up resistor
PUR00	R00 Pull-up Resistor Enable Bit	0: Disable pull-up resistor 1: Enable pull-up resistor

R0

R0 PORT DATA REGISTER **00C0H**

	7	6	5	4	3	2	1	0	
R0	R07	R06	R05	R04	R03	R02	R01	R00	Reset value: 00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

In input mode, it represents the R0 port status. In output mode, R0 port represents it.	1: High 0 : Low

11. INTERRUTP CONTROLLER

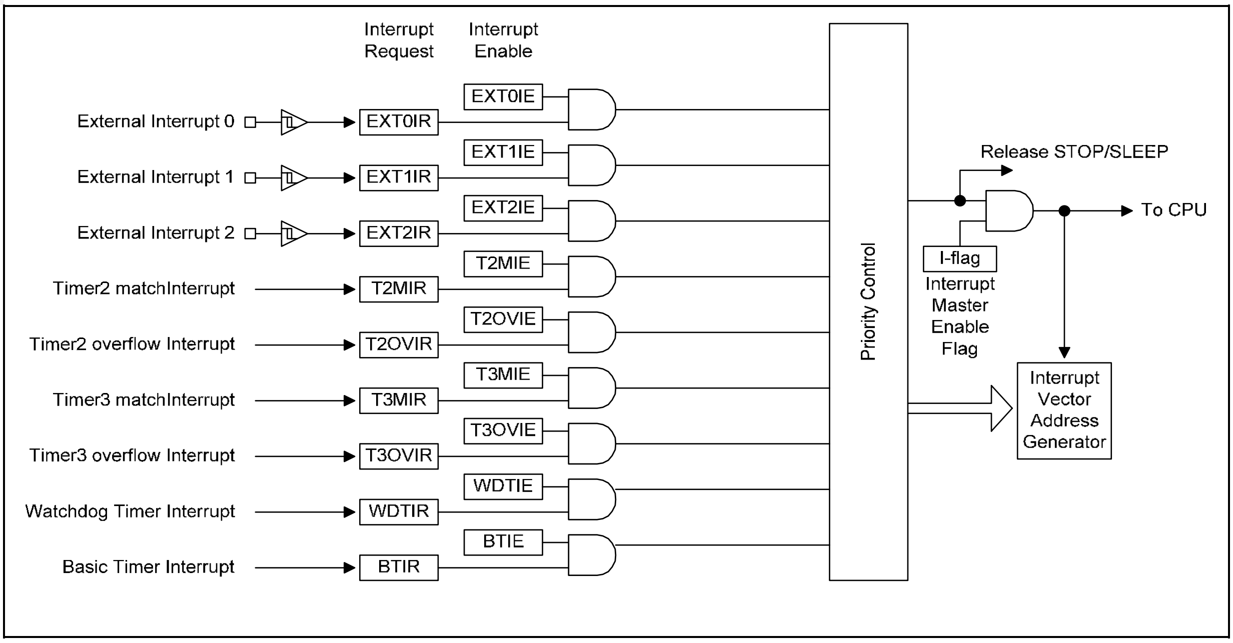


Figure 11-1 Block Diagram of Interrupt

The MC81F4104 interrupt circuits consist of Interrupt enable register (IENH, IENL), Interrupt request flags of IRQH, IRQL, Priority circuit, and Master enable flag (“I” flag of PSW). And 9 interrupt sources are provided.

The interrupt vector addresses are shown in ‘11.5 Interrupt Vector & Priority Table’ on page 65. Interrupt enable registers are shown in next paragraph. These registers are composed of interrupt enable flags of each interrupt source and these flags determine whether an interrupt will be accepted or not. When the enable flag is “0”, a corresponding interrupt source is disabled.

Note that PSW contains also a master enable bit, I-flag, which disables all interrupts at once.

11.1 Registers

IENH

INTERRUPT ENABLE HIGH REGISTER **00EAH**

	7	6	5	4	3	2	1	0	
IENH	-	-	-	-	T2MIE	T2OVIE	T2MIE	T3OVIE	Reset value:
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	----_0000b

–	bit7 – bit4	Not used for MC81F4104
T2MIE	Timer 2 Match Interrupt Enable Bit	0: Disable interrupt 1: Enable interrupt
T2OVIE	Timer 2 Overflow Interrupt Enable Bit	0: Disable interrupt 1: Enable interrupt
T3MIE	Timer 3 Match Interrupt Enable Bit	0: Disable interrupt 1: Enable interrupt
T3OVIE	Timer 3 Overflow Interrupt Enable Bit	0: Disable interrupt 1: Enable interrupt

IENL

INTERRUPT ENABLE LOW REGISTER **00EBH**

	7	6	5	4	3	2	1	0	
IENL	-	-	-	-	-	WDTIE	–	BITIE	Reset value: 00H
	R/W	R/W	R/W	R/W	R/W	R/W	–	R/W	

–	bit7 – bit3	Not used for MC81F4104
WDTIE	Watchdog Timer Interrupt Enable Bit	0: Disable interrupt 1: Enable interrupt
–	bit1	Not used for MC81F4104
BTIE	Basic Timer Interrupt Enable Bit	0: Disable interrupt 1: Enable interrupt

IRQH

INTERRUPT REQUSEST HIGH REGISTER 00ECH

	7	6	5	4	3	2	1	0	
IQRH	-	-	-	-	T2MIR	T2OVIR	T3MIR	T3OVIR	Reset value:
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	----_0000b

–	bit7 – bit4	Not used for MC81F4104
T2MIR	Timer 2 Match Interrupt Request Flag	0: Interrupt request flag is not pending, request flag bit clear 1: Interrupt request flag is pending
T2OVIR	Timer 2 Overflow Interrupt Request Flag	0: Interrupt request flag is not pending, request flag bit clear 1: Interrupt request flag is pending
T3MIR	Timer 3 Match Interrupt Request Flag	0: Interrupt request flag is not pending, request flag bit clear 1: Interrupt request flag is pending
T3OVIR	Timer 3 Overflow Interrupt Request Flag	0: Interrupt request flag is not pending, request flag bit clear 1: Interrupt request flag is pending

IRQL

INTERRUPT REQUSEST LOW REGISTER 00EDH

	7	6	5	4	3	2	1	0	
IRQL	-	-		-	-	WDTIR	–	BITIR	Reset value: 00H
	R/W	R/W	R/W	R/W	R/W	R/W	–	R/W	

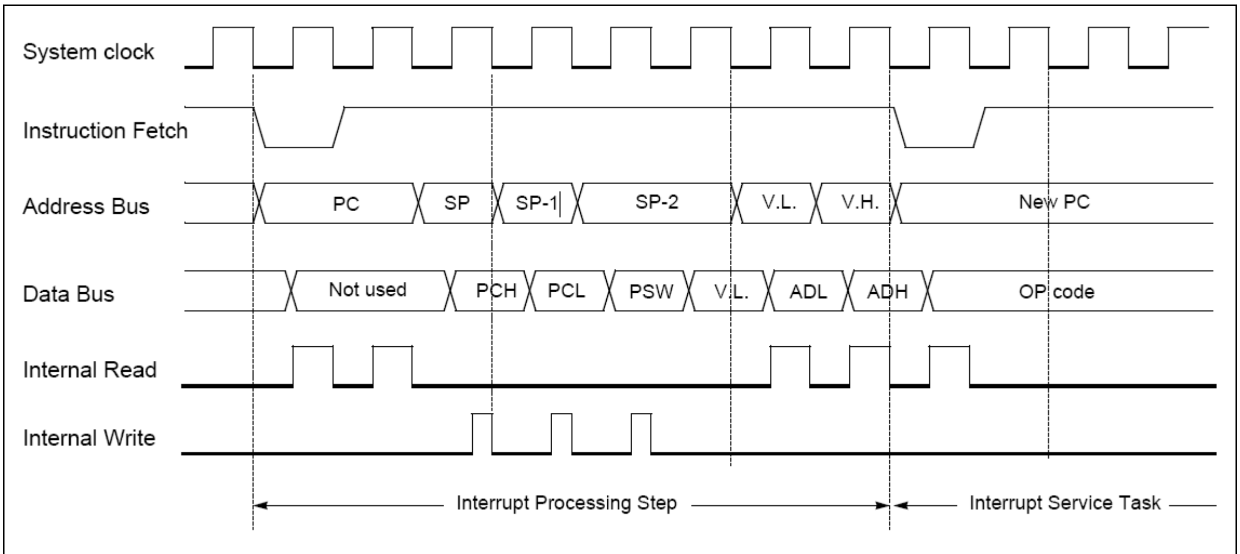
–	bit7 – bit4	Not used for MC81F4104
WDTIR	Watchdog Timer Interrupt Request Flag	0: Interrupt request flag is not pending, request flag bit clear 1: Interrupt request flag is pending
–	bit1	Not used for MC81F4104
BTIR	Basic Timer Interrupt Request Flag	0: Interrupt request flag is not pending, request flag bit clear 1: Interrupt request flag is pending

11.2 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 sequence requires 8 cycles of fXIN (1μs at fXIN= 4MHz) after the completion of the current instruction execution. The interrupt service task is terminated upon execution of an interrupt return instruction [RETI].

Interrupt acceptance

1. The interrupt master enable flag (I-flag) is cleared to “0” to temporarily disable the acceptance of any following maskable interrupts. When a non-maskable interrupt is accepted, the acceptance of any following interrupts is temporarily disabled.
2. Interrupt request flag for the interrupt source accepted is cleared to “0”.
3. The contents of the program counter (return address) and the program status word are saved (pushed) onto the stack area. The stack pointer decreases 3 times.
4. The entry address of the interrupt service program is read from the vector table address and the entry address is loaded to the program counter.
5. The instruction stored at the entry address of the interrupt service program is executed.



V.L. and V.H. are vector addresses.
ADL and ADH are start addresses of interrupt service routine as vector contents.

Figure 11-2 Timing chart of Interrupt Acceptance and Interrupt Return Instruction

A interrupt request is not accepted until the I-flag is set to “1” even if a requested interrupt has higher priority than that of the current interrupt being serviced. When nested interrupt service is required, the I-flag should be set to “1” by “EI” instruction in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

Saving/Restoring General-purpose Register

the program status word are automatically saved on the stack, but accumulator and other registers are not saved itself. These registers are saved by the software if necessary. Also, when multiple interrupt services are nested, it is necessary to avoid using the same data memory area for saving registers.

The following method is used to save/restore the general-purpose registers.

Example: Register save using push and pop instructions.

```
INTxx :
PUSH A
PUSH X
PUSH Y
;SAVE ACC.
;SAVE X REG.
;SAVE Y REG.

;; interrupt processing ;;

POP Y
POP X
POP A
RETI
;RESTORE Y REG.
;RESTORE X REG.
;RESTORE ACC.
;RETURN
```

General-purpose register save/restore using push and pop instructions;

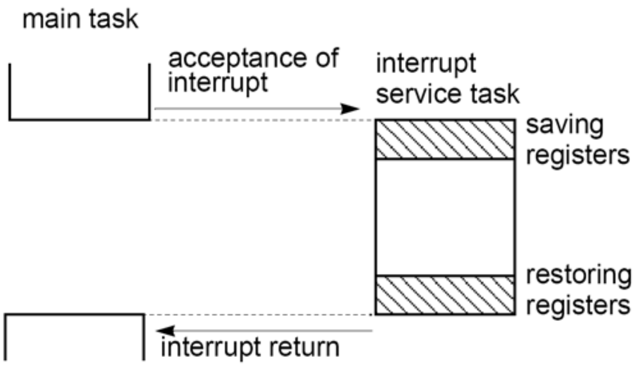
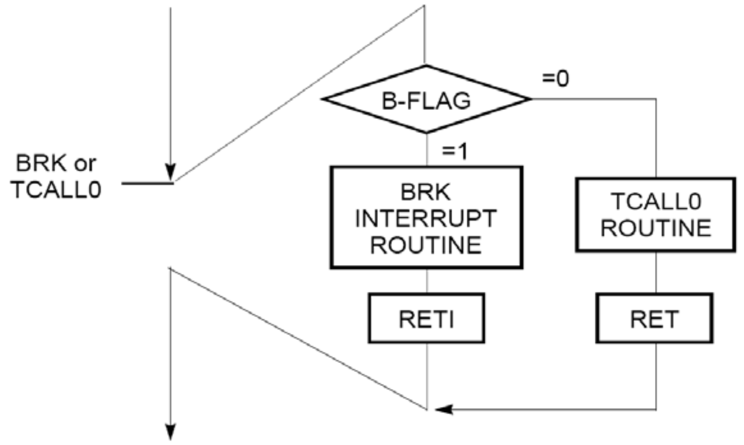


Figure 11-3 Saving/Restoring in Interrupt Routine

11.3 BRK Interrupt

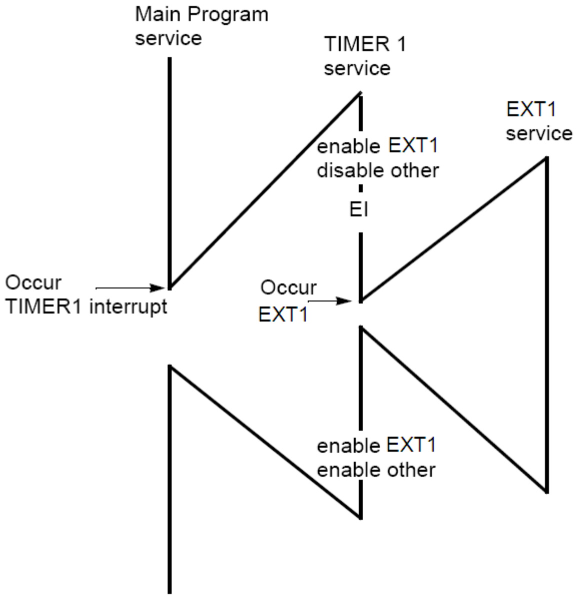
Software interrupt can be invoked by BRK instruction, which has the lowest priority order. Interrupt vector address of BRK is shared with the vector of TCALL 0 (Refer to Program Memory Section). When BRK interrupt is generated, B-flag of PSW is set to distinguish BRK from TCALL 0.

Each processing step is determined by B-flag as shown in Figure



11.4 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 internal polling sequence determines by hardware which request is serviced. However, multiple processing through software for special features is possible. Generally when an interrupt is accepted, the I-flag is cleared to disable any further interrupt. But as user sets I-flag in interrupt routine, some further interrupt can be serviced even if certain interrupt is in progress.



In this example, the EXT1 interrupt can be serviced without any pending, even TIMER1 is in progress.

Because of re-setting the interrupt enable registers IENH,IENL and master enable "EI" in the TIMER1 routine.

Figure 11-4 Execution of Multi Interrupt

11.5 Interrupt Vector & Priority Table

Address	Interrupt	INT number	Priority
0FFE0H	Basic Interval Timer	INT0	15 (lowest priority)
0FFE2H	Watchdog Timer	INT1	14
0FFE4H	Timer 3 overflow	INT2	13
0FFE6H	Timer 3 match	INT3	12
0FFE8H	Timer 2 overflow	INT4	11
0FFEAH	Timer 2 match	INT5	10
0FFECH	-	-	9
0FFEEH	-	-	8
0FFF0H	-	-	7
0FFF2H	-	-	6
0FFF4H	-	-	5
0FFF6H	-	-	4
0FFF8H	External 2	INT12	3
0FFFAH	External 1	INT13	2
0FFFCH	External 0	INT14	1
0FFFEH	RESET	INT15	0 (highest priority)

Table 11-1 Interrupt Vector & Priority

12. EXTERNAL INTERRUPTS

The external interrupt pins are edge triggered depending on the ‘external interrupt registers’.

The edge detection of external interrupt has three transition activated mode: rising edge, falling edge, and both edge.

12.1 Registers

EINT0 – EXT 2~0 / R04~R07

R0 PORT EXTERNAL INTERRUPT ENABLE HIGH REGISTER **00CAH**

You can use EINT0H register setting to select Disable interrupt or Enable interrupt (by falling, rising, or both falling and rising edge).

	7	6	5	4	3	2	1	0	
EINT0	-		EXT2IE		EXT1IE		EXT0IE		Reset value:
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	--00_0000b

-	bit 7 – bit 6	Not used for MC81F4104
EXT2IE	**R07/EXT2 External Interrupt Enable Bits**	00: Disable Interrupt 01: Enable Interrupt by falling edge 10: Enable Interrupt by rising edge 11: Enable Interrupt by both falling and rising edge
EXT1IE	**R06/EXT1 External Interrupt Enable Bits**	
EXT0IE	**R04/EXT0 External Interrupt Enable Bits**	

ERQ0 – EXT 10,11,0~5 / R00~R07

R0 PORT EXTERNAL INTERRUPT REQUEST REGISTER **00CCH**

When an interrupt is generated, the bit of ERQ0 that generated it is cleared by the hardware when the service routine is vectored to only if the interrupt was transition-activated.

	7	6	5	4	3	2	1	0	
ERQ0	-	-	-	-	-	EXT2IR	EXT1IR	EXT0IR	Reset value: 00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

-	bit 7 – bit 3	Not used for MC81F4104
EXT2IR	R07/EXT2 External Interrupt Request Flag	0: Interrupt request flag is not pending, request flag bit clear 1: Interrupt request flag is pending
EXT1IR	R06/EXT1 External Interrupt Request Flag	
EXT0IR	R04/EXT0 External Interrupt Request Flag	

12.2 Procedure

To generate external interrupt, following steps are required,

1. Prepare external interrupt sub-routine(function).
2. Set external interrupt pins to input mode. (use RnCONH/M/L registers).
3. Enable the external interrupt and select the edge mode. (use EINT0 register).
4. Make sure global interrupt is enabled. (use 'EI' instruction).

After finish above steps, the external interrupt sub-routine is calling, when the edge is detected.

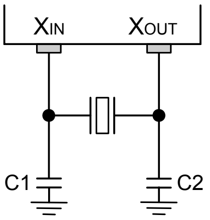
When the generated external interrupt is one of the external interrupt groups, the EINTF register is used to recognize which external interrupt is generated.

13. OSCILLATION CIRCUITS

There are few example circuits for main oscillators.

Oscillation circuit is designed to be used either with a ceramic resonator or crystal oscillator. Since each crystal and ceramic resonator have their own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

13.1 Main Oscillation Circuits



C1, C2 = 10 ~ 30 pF

* The example load capacitor value(C1, C2) is common value but may not be appropriate for some crystal or ceramic resonator.

Figure 13-1 Crystal/Ceramic Oscillator

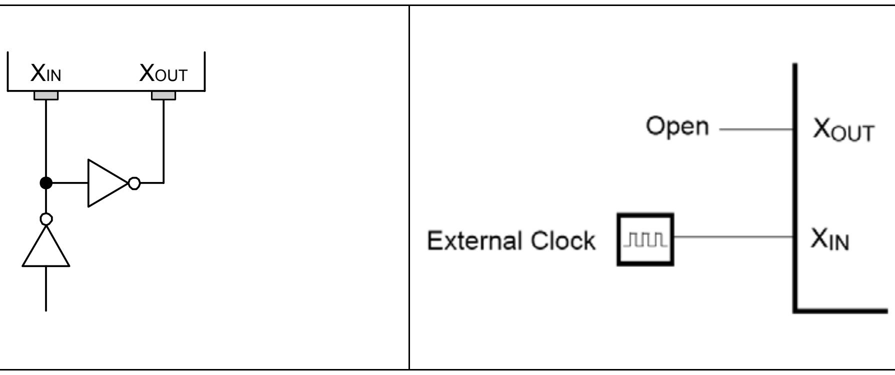
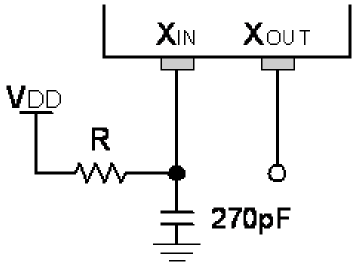
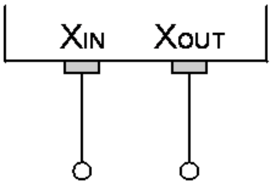


Figure 13-2 External Clock



Xout pin can be used as a normal pin.

Figure 13-3 External RC Oscillator



Xout and Xin pins can be used as normal pins

Figure 13-4 Internal RC Oscillator

13.2 PCB Layout

For reference, here is an example layout for oscillator circuit.

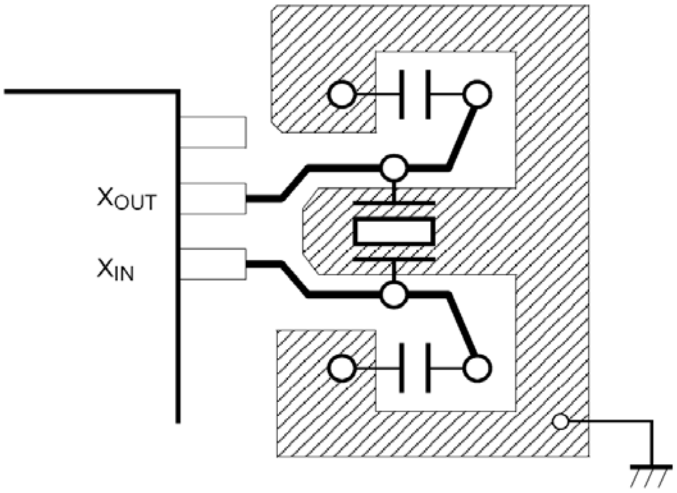


Figure 13-5 Layout of Oscillator PCB circuit

Note : Minimize the wiring length. Do not allow the wiring to intersect with other signal conductors. Do not allow the wiring to come near changing high current. Set the potential of the grounding position of the oscillator capacitor to that of V_{SS}. Do not ground it to any ground pattern where high current is present. Do not fetch signals from the oscillator.

14. BASIC INTERVAL TIMER

The MC81F4104 has one 8-bit Basic Interval Timer that is free-run and can not be stopped except when peripheral clock is stopped.

The Basic Interval Timer generates the time base for watchdog timer counting. It also provides a Basic interval timer interrupt.

The 8-bit Basic interval timer register (BTCR) is increased every internal count pulse which is divided by prescaler. Since prescaler has divided ratio by 8 to 1024, the count rate is 1/8 to 1/1024 of the oscillator frequency.

As the count overflow from FFH to 00H, this overflow causes the interrupt to be generated. The Basic Interval Timer is controlled by the clock control register (CKCTLR).

When write "1" to bit BTCL of CKCTLR, BTCR register is cleared to "0" and restart to count-up. The bit BTCL becomes "0" after one machine cycle by hardware.

The bit WDTON decides Watchdog Timer or the normal 7-bit timer.

Source clock can be selected by lower 3 bits of CKCTLR.

14.1 Registers

CKCTLR

CLOCK CONTROL REGISTER **00F2H**

	7	6	5	4	3	2	1	0	
CKCTLR	–	–	–	WDTON	BTCL	BTS			Reset value: 17H
	–	–	–	R/W	R/W	R/W	R/W	R/W	

–	bit7 – bit5	Not used for MC81F4104
WDTON	Watchdog Timer Enable Bit	0: Operate as 7-bit timer 1: Enable Watchdog timer
BTCL	Basic Timer Clear Bit	0: Normal operation (free-run) 1: Clear 8-bit counter (BITR) to “0”, This bit becomes 0 automatically after one machine cycle, and starts counting.
BTS	Basic Interval Timer Source Clock Selection Bits	000: fxin/8 001: fxin/16 010: fxin/32 011: fxin/64 100: fxin/128 101: fxin/256 110: fxin/512 111: fxin/1024

CKCTLR[2:0]	Source clock	Interrupt(overflow) period (ms) @ fxin = 8MHz
000	fxin/8	0.256
001	fxin/16	0.512
010	fxin/32	1.024
011	fxin/64	2.048
100	fxin/128	4.096
101	fxin/256	8.192
110	fxin/512	16.384
111	fxin/1024	32.768

Figure 14-1 Basic Interval Timer Interrupt Period

BTCR

BASIC TIMER COUNTER REGISTER **00F1H**

	7	6	5	4	3	2	1	0	
BTCR	One byte register								Reset value: XXH
	R	R	R	R	R	R	R	R	

A 8 bit count register for the basic interval timer.

15. WATCH DOG TIMER

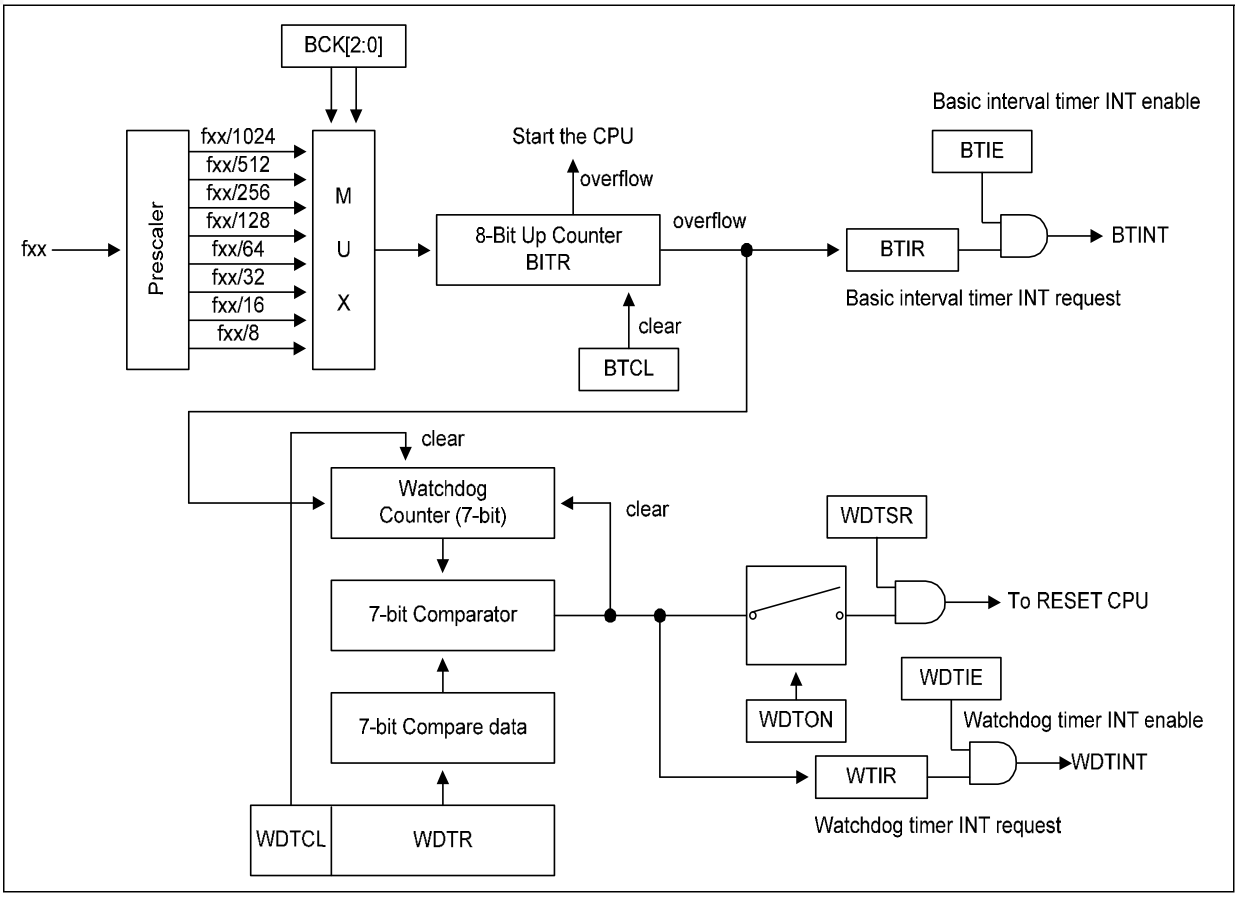


Figure 15-1 Block diagram of Basic Interval Timer/Watchdog Timer

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 a 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.

The watchdog timer uses the Basic Interval Timer as a clock source.

The watchdog timer consists of 7-bit binary counter and the watchdog timer data register. When the value of 7-bit binary counter is equal to the lower 7 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 WDTON.

Watchdog reset feature is disabled when the watchdog timer status register(WDTSR) value is '0A5h'. Note that, WDTSR's reset value is '00h'. And reset value of WDTON is '1'. So watchdog timer reset is enabled at reset time.

15.1 Registers

WDTR

WATCHDOG TIMER REGISTER **00F4H**

	7	6	5	4	3	2	1	0	
WDTR	WDTCL	WDTCMP							Reset value: 7FH
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

WDTCL	Watchdog Timer Clear Bit	0: Free-run count 1: When the WDTCL is set to “1”, binary counter is cleared to “0”. And the WDTCL becomes “0” automatically after one machine cycle. Counter count up again.
WDTCMP	bit6 – bit0	7-bit compare data

WDTSR

WATCHDOG TIMER STATUS REGISTER **00F6H**

	7	6	5	4	3	2	1	0	
WDTSR	One byte register								Reset value: 00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Watchdog Timer Function Disable Code (for System Reset)	10100101: Disable watchdog timer function Others: Enable watchdog timer function

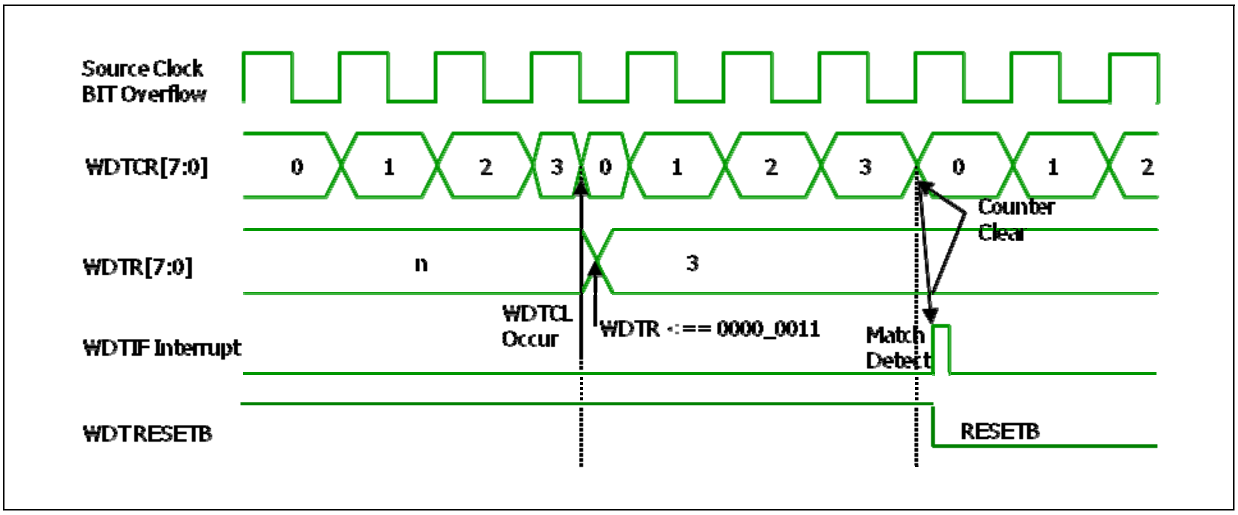


Figure 15-2 Watchdog Timer Timing

16. Timer 2

The 8-bit timer 2 is an 8-bit general-purpose timer. Timer 2 have two operating modes, you can select one of them using the appropriate T2SCR setting:

- Interval timer mode (Toggle output at T2O pin)
- Capture input mode with a rising or falling edge trigger at EXT0 pin

16.1 Registers

T2DR

TIMER 2 DATA REGISTER **00D1H**

	7	6	5	4	3	2	1	0	
T2DR	One byte register								Reset value: FFH
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

A 8-bit compare value register for the timer 2 match interrupt.

T2CR

TIMER 2 COUNTER REGISTER **00D2H**

	7	6	5	4	3	2	1	0	
T2CR	One byte register								Reset value: 00H
	R	R	R	R	R	R	R	R	

A 8-bit count register for the timer 2

T2SCR

TIMER 2 STATUS AND CONTROL REGISTER (T2SCR) 00D0H

To enable the timer 2 match interrupt, you must set “1” to T2MIE.
When the timer 2 match interrupt sub-routine is serviced, the timer 1 match interrupt request flag bit, T2MIR, is cleared automatically.

To enable the timer 2 overflow interrupt, you must set “1” to T2OVIE.
When the timer 2 overflow interrupt sub-routine is serviced, the timer 2 overflow interrupt request flag bit, T2OVIR is cleared automatically.

	7	6	5	4	3	2	1	0	
T2SCR	-	–	T2MS	T2CC	T2CS				Reset value:
	-	–	R/W	R/W	R/W	R/W	R/W	R/W	--00_0000b

–	bit7 - bit6	Not used for MC81F4104
T2MS	Timer 2 Mode Selection Bit	0: Interval mode (T2O) 1: Capture mode (OVF can occur)
T2CC	Timer 2 Counter Clear Bit	0: No effect 1: Clear the Timer 2 counter (When write, automatically cleared “0” after being cleared counter)
T2CS	Timer 2 Clock Selection Bits	0000: Counter stop 0001: Not available 0010: Not available 0011: Not available 0100: Not available 0101: External clock (EC2) rising edge 0110: External clock (EC2) falling edge 0111: Not available 1000: fxx/1 1001: fxx/2 1010: fxx/4 1011: fxx/8 1100: fxx/16 1101: fxx/64 1110: fxx/256 1111: fxx/1024

Note :

You must set the T2CC(T2SCR.4) bit after set T2DR register. The timer 2 counter value is compared with timer 2 buffer register instead of T2DR. And T2DR value is copied to timer 2 buffer.

16.2 Timer 2 8-Bit Mode

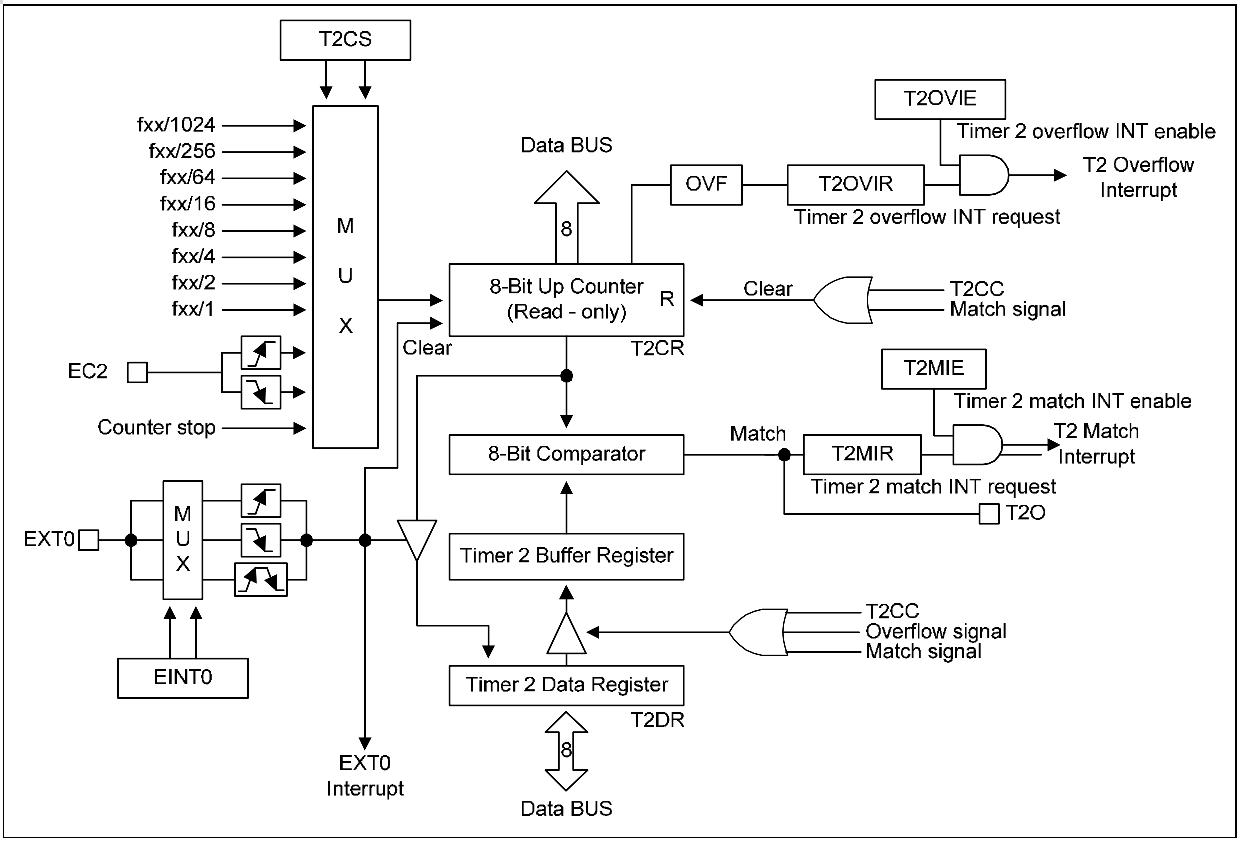


Figure 16-1 8-bit Timer 2 Block Diagram

Timer 2 has the following functional components:

- Clock frequency divider (fxx divided by 1024, 256, 64, 16, 8, 4, 2, 1, fxt) with multiplexer
- External clock input pin, EC2 (R06)
- I/O pins for capture input, EXT0 (R04) or match output T2O (R04)
- 8-bit counter (T2CR), 8-bit comparator, and 8-bit reference data register (T2DR)
- Timer 2 status and control register (T2SCR)
- Timer 2 overflow interrupt and match interrupt generation

Function Description

Interval Timer Mode

A match signal is generated and T2O pins are toggled when the T2CR register value equals the T2DR register value. The match signal generates a timer match interrupt and clears the T2CR register.

Capture Mode

In capture mode, you have to set EXT0 interrupt. When the EXT0 interrupt is occurred, the T2CR register value is loaded into the T2DR register and the T2CR register is cleared.

And the timer 2 overflow interrupt is generated whenever the T2CR value is overflowed.

So, If you count how many overflow is occurred and read the T2DR value in EXT0 interrupt routine, it is possible to measure the time between two EXT0 interrupts. Or it is possible to measure the time from the T2 initial time to the EXT0 interrupt occurred time.

The time = (256 * tCLK) * overflow_count + (tCLK * T2DR)

Note

‘t_{CLK}’ is the period time of the timer-counter’s clock source

You must set the T2DR value before set the T2SCR register. Because T2DR value is fetched when the count is started(the T2CC bit is set) or match/overflow event is occurred.

17. Timer 3

The 8-bit timer 3 is an 8-bit general-purpose timer. Timer 3 have two operating modes, you can select one of them using the appropriate T3SCR setting:

- Interval timer mode (Toggle output at T3O pin)
- Capture input mode with a rising or falling edge trigger at EXT2 pin

17.1 Registers

T3DR

TIMER 3 DATA REGISTER **00D4H**

	7	6	5	4	3	2	1	0	
T3DR	One byte register								Reset value: FFH
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

A 8-bit compare value register for the timer 3 match interrupt.

T3CR

TIMER 3 COUNTER REGISTER **00D5H**

	7	6	5	4	3	2	1	0	
T3CR	One byte register								Reset value: 00H
	R	R	R	R	R	R	R	R	

A 8-bit count register for the timer 3

T3SCR

TIMER 3 STATUS AND CONTROL REGISTER **00D3H**

To enable the timer 3 match interrupt, you must set “1” to T3MIE.
When the timer 3 match interrupt sub-routine is serviced, the timer 1 match interrupt request flag bit, T3MIR, is cleared automatically.

To enable the timer 3 overflow interrupt, you must set “1” to T3OVIE.
When the timer 3 overflow interrupt sub-routine is serviced, the timer 3 overflow interrupt request flag bit, T3OVIR, is cleared automatically.

	7	6	5	4	3	2	1	0	
T3SCR	–	–	T3MS	T3CC	T3CS				Reset value: 00H
	–	–	R/W	R/W	R/W	R/W	R/W	R/W	

–	bit7 – bit6	Not used for MC81F4104
T3MS	Timer 3 Mode Selection Bit	0: Interval mode 1: Capture mode (OVF can occur)
T3CC	Timer 3 Counter Clear Bit	0: No effect 1: Clear the Timer 3 counter (When write, automatically cleared “0” after being cleared counter)
T3CS	Timer 3 Clock Selection Bits	0000: Counter stop 0001: Not available 0010: Not available 0011: Not available 0100: Not available 0101: External clock (EC3) rising edge 0110: External clock (EC3) falling edge 0111: Not available 1000: fxx/2 1001: fxx/4 1010: fxx/8 1011: fxx/16 1100: fxx/32 1101: fxx/128 1110: fxx/512 1111: fxx/2048

Note :

You must set the T3CC(T3SCR.4) bit after set T3DR register. The timer 3 counter value is compared with timer 3 buffer register instead of T3DR. And T3DR value is copied to timer 3 buffer.

17.2 Timer 3 8-Bit Mode

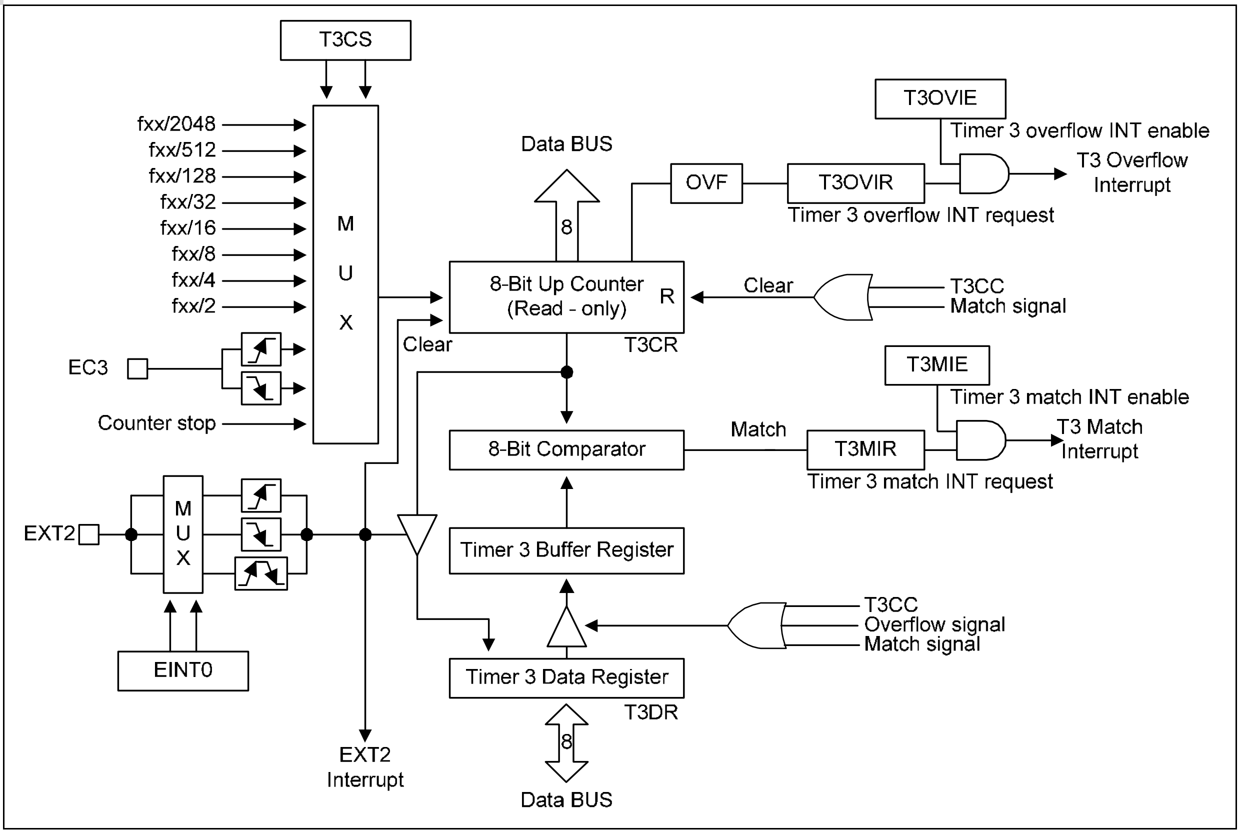


Figure 17-1 8-bit Timer 3 Block Diagram

Timer 3 has the following functional components:

- Clock frequency divider (fxx divided by 2048, 512, 128, 32, 16, 8, 4, 2) with multiplexer
- External clock input pin, EC3 (R07)
- I/O pins for capture input, EXT2 (R07)
- 8-bit counter (T3CR), 8-bit comparator, and 8-bit reference data register (T3DR)
- Timer 3 status and control register (T3SCR)
- Timer 3 overflow interrupt and match interrupt generation

Function Description

Interval Timer Mode

A match signal is generated and T3O pins are toggled when the T3CR register value equals the T3DR register value. The match signal generates a timer match interrupt and clears the T3CR register.

Capture Mode

In capture mode, you have to set EXT2 interrupt. When the EXT2 interrupt is occurred, the T3CR register value is loaded into the T3DR register and the T3CR register is cleared.

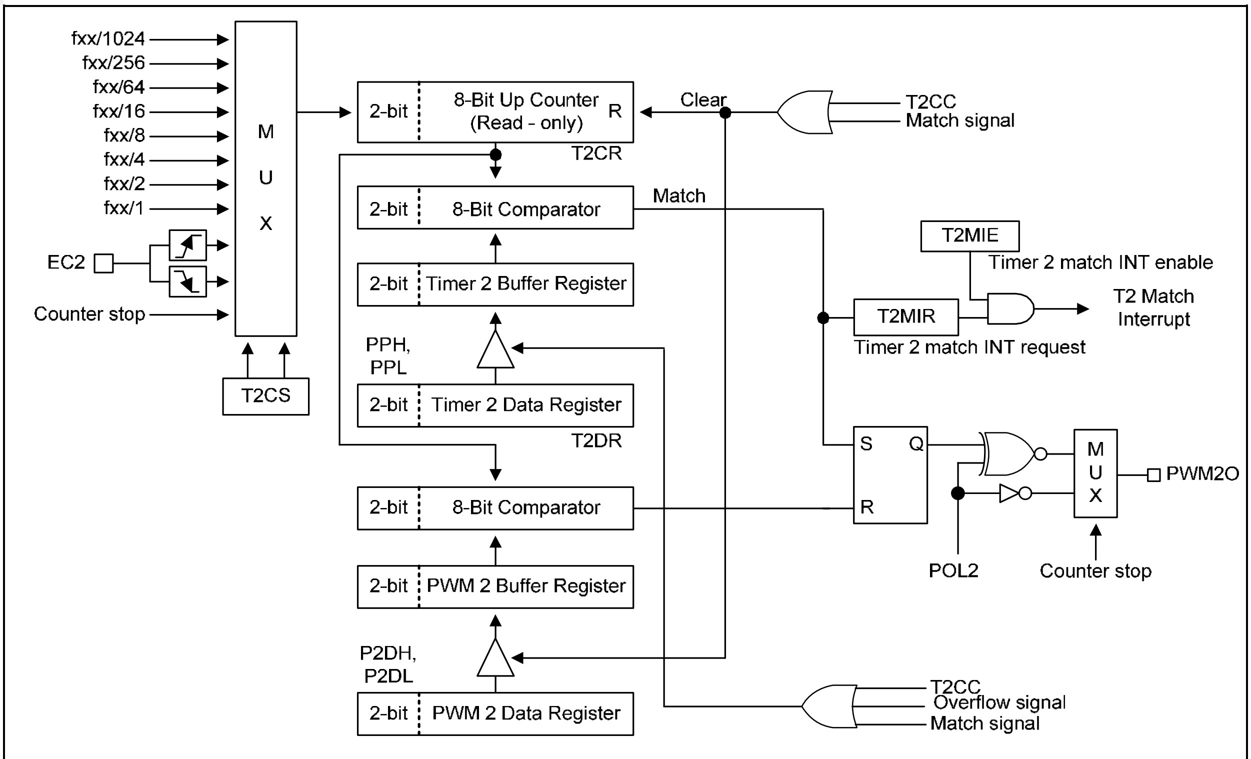
And the timer 3 overflow interrupt is generated whenever the T3CR value is overflowed.

So, If you count how many overflow is occurred and read the T3DR value in EXT2 interrupt routine, it is possible to measure the time between two EXT2 interrupts. Or it is possible to measure the time from the T3 initial time to the EXT2 interrupt occurred time.

The time = (256 * tCLK) * overflow_count + (tCLK * T3DR)

Note
'tCLK' is the period time of the timer-counter's clock source
You must set the T3DR value before set the T3SCR register. Because T3DR value is fetched when the count is started(the T3CC bit is set) or match/overflow event is occurred.

18. High Speed PWM



NOTE:

1. When you cleared the POL2 and counter stop, PWM2O is high status.
2. When you set the POL2 and counter stop, PWM2O is low status.

Figure 18-1 High Speed PWM Block Diagram

The MC81F4104 has one high speed PWM (Pulse Width Modulation) function which shared with Timer2.

In PWM mode, the R04/PWM2O pin operates as a 10-bit resolution PWM output port. For this mode, the R04 of R0CONM should be set to alternative function mode.

The period of the PWM output is determined by the T2DR (T2 data Register) and PWMPDR[1:0] (PWM Period Duty Register) and the duty of the PWM output is determined by the PWM2DR(PWM 2 Data Register) and PWMPDR[3:2] (PWM Period Duty Register).

User can use PWM data by writing the lower 8-bit period value to the T2DR and the higher 2-bit period value to the PWMPDR[1:0]. And the duty value can be used with the PWM2DR and the PWMPDR[3:2] in the same way.

The bit POL2 of PWMSCR decides the polarity of duty cycle. The duty value can be changed when the PWM outputs. However the changed duty value is output after the current period is over. And it can be maintained the duty value at present output when changed only period value shown as Example of PWM2. As it were, the absolute duty time is not changed in varying frequency.

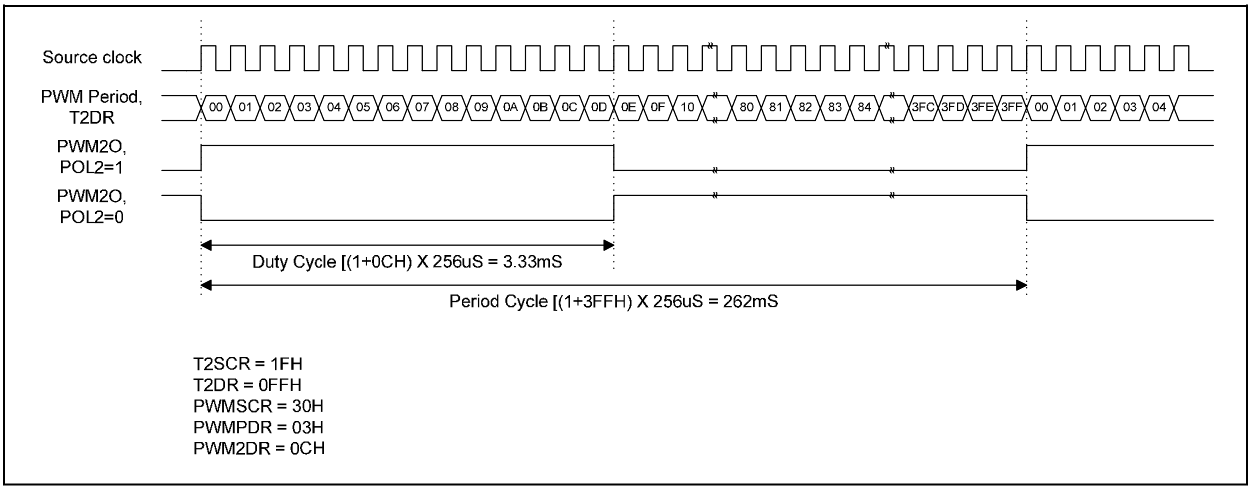
Note :
When user need to change mode from the Timer2 mode to the PWM mode, the Timer2 should be stopped firstly, and then set period and duty register value. If user writes register values and changes mode to PWM mode while Timer2 is in operation, the PWM data would be different from expected data in the beginning.

PWM Period = [PWMPDR[1:0]T2DR+1] X Source Clock

PWM2 Duty = [PWMPDR[3:2]PWM2DR+1] X Source Clock

If it needed more higher frequency of PWM, it should be reduced resolution.

Note :
If the duty value and the period value are same, 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). The period value must be same or more than the duty value, and 00H cannot be used as the period value.


Source clock
PWM Period, T2DR
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 80 81 82 83 84 3FC 3FD 3FE 3FF 00 01 02 03 04
PWM2O, POL2=1
PWM2O, POL2=0
Duty Cycle [(1+0CH) X 256uS = 3.33mS
Period Cycle [(1+3FFH) X 256uS = 262mS

T2SCR = 1FH
T2DR = 0FFH
PWMSCR = 30H
PWMPDR = 03H
PWM2DR = 0CH

Figure 18-2 Example of PWM2 at 8MHz

18.1 Registers

PWMSCR

PWM STATUS AND CONTROL REGISTER (PWMSCR) **00E2H**

	7	6	5	4	3	2	1	0	
PWMSCR	POL2	PWMS	-	-	–	–	–	–	Reset value:
	R/W	R/W	R/W	R/W	–	–	–	–	00--_----b

POL2	PWM 2 Polarity Selection Bit	0: PWM 2 duty active low 1: PWM 2 duty active high
PWMS	PWM Selection Bit	0: Timer 2 mode (interval or capture) 1: PWM mode (PWM2O, PWM3O, PWM4O)
–	bit5 – bit0	Not used for MC81F4104

PWMPDR

PWM PERIOD DUTY REGISTER **00E3H**

	7	6	5	4	3	2	1	0	
PWMPDR	-	-			P2DH	P2DL	PPH	PPL	Reset value:-0H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

-	bit 7 – bit 4	Not used for MC81F4104
P2DH	PWM 2 Duty High Bit	PWM2 duty value (9,8th bits)
P2DL	PWM 2 Duty Low Bit	
PPH	PWM Period High Bit	Period value (9/8th bits)
PPL	PWM Period Low Bit	

PWM2DR

PWM 2 DATA REGISTER **00E6H**

	7	6	5	4	3	2	1	0	
PWM2DR	One byte register								Reset value: FFH
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

A 8-bit data register for lower bits of 10-bit PWM 2 duty value.

19. 12-BIT ADC

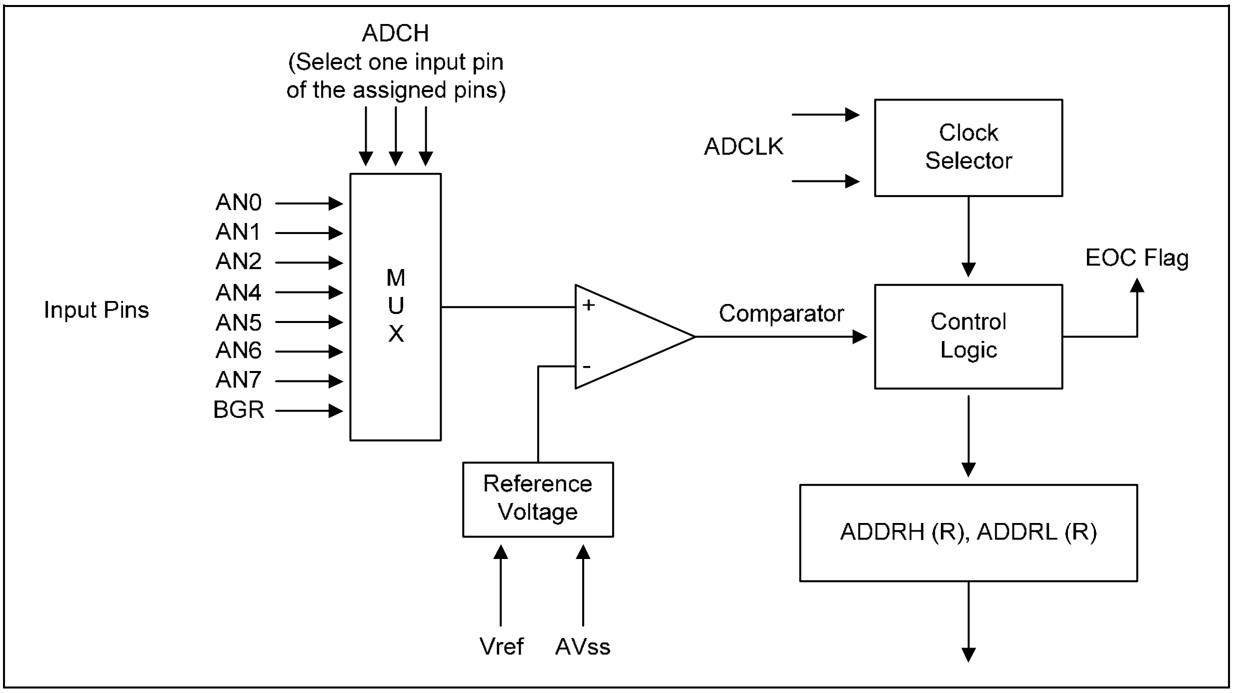


Figure 19-1 A/D Converter Block Diagram

The 12-bit A/D converter (ADC) module uses successive approximation logic to convert analog levels entering at one of the 1` input channels to equivalent 12-bit digital values. The analog input level must lie between the V_{REF} and V_{SS} values. The A/D converter has the analog comparator with successive approximation logic, D/A converter logic (resistor string type), A/D mode register (ADMR), 8 multiplexed analog data input pins (AD0-AD2,AD4-AD7,BGR), and 12-bit A/D conversion data output register (ADDRH/ADDRL).

19.1 Registers

ADMR

A/D MODE REGISTER **00DDH**

	7	6	5	4	3	2	1	0	
ADMR	SSBIT	EOC	ADCLK		ADCH				Reset value: 00H
	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	

After reset, the start/stop bit is turned off. You can select only one analog input channel at a time. Other analog input (AD0-AD2, AD4-AD7,BGR) can be selected dynamically by manipulating the ADCH(ADMR[4:0]). And the pins not used for analog input can be used for normal I/O function.

SSBIT	Start or Stop bit	0: Stop operation 1: Start operation	
EOC	End of Conversion	0: Conversion not complete 1: Conversion complete	
ADCLK	A/D Clock Selection	00: fxx/1 01: fxx/2	10: fxx/4 11: fxx/8
ADCH	A/D Input Pin Selection	0000: AN0 0001: AN1 0010: AN2 0011: Not available 0100: AN4 0101: AN5 0110: AN6 0111: AN7	1000: available 1001: Not available 1010: Not available 1011: Not available 1100: Not available 1101: Not available 1110: AN14 1111: BGR

ADDRH

A/D CONVERTER DATA HIGH REGISTER **00DEH**

	7	6	5	4	3	2	1	0	
ADDRH	.11	.10	.9	.8	.7	.6	.5	.4	Reset value: XXH
	R	R	R	R	R	R	R	R	

A 8-bit data register for higher 8-bits of the 12-bit ADC result.

ADDRL

A/D CONVERTER DATA LOW REGISTER **00DFH**

	7	6	5	4	3	2	1	0	
ADDRL	.3	.2	.1	.0	-	-	-	-	Reset value: X-H
	R	R	R	R	R	R	R	R	

A 8-bit data register for lower 4-bits of the 12-bit ADC result.

19.2 Procedure

To do the A/D converting, follow these basic steps:

1. Set the ADC pins as the alternative mode.
2. Set the ADMR register for
   - setting ADC channel
   - setting Clock
   - clearing the ‘End of Conversion’ bit
   - starting ADC
3. Wait until ADC is finished (check the ‘End of Conversion’ bit)
   When ADC is finished, EOC bit is set and SSBIT is cleared automatically.
4. Read the ADCRH and ADCRL register

To initiate an analog-to-digital conversion procedure, at first you must set ADC pins to alternative function (ADC analog input) mode. And you write the channel selection data in the A/D mode register (ADMR) to select one of analog input channels and set the conversion start/stop bit, SSBIT. The pins not used for ADC can be used for normal I/O.

To start the A/D conversion, you should set the start/stop bit, SSBIT. When a conversion is completed, the end-of-conversion bit, EOC is automatically set to 1 and the result is dumped into the ADDRH/ADDRL register. Then the A/D converter enters an idle state. The EOC bit is cleared when SSBIT is set.

Note that, ADC interrupt is not provided.

Note : Because the A/D converter has no sample-and-hold circuitry, it is very important that fluctuation of the analog level at the AD0-AD2,AD4-AD7 input pins during a conversion procedure be kept to an absolute minimum. Any change in the input level, perhaps due to noise, will invalidate the result.
If the chip enters to STOP or IDLE mode in conversion process, there will be a leakage current path in A/D block. You must use STOP or IDLE mode after ADC operation is finished.

19.3 Conversion Timing

The A/D conversion process requires 4 steps (4 clock edges) to convert each bit and 10 clocks to set-up A/D conversion. Therefore, total of 66 clocks are required to complete a 12-bit conversion: When fxx/8 is selected for conversion clock with a 12 MHz fxx clock frequency, one clock cycle is 0.66 µs. Each bit conversion requires 4 clocks, the conversion rate is calculated as follows:

4 clocks/bit × 14 bits + set-up time = 66 clocks, 66 clock × 0.66 µs = 44.0 µs at 1.5 MHz (12 MHz/8)

Note : The A/D converter needs at least 25 µs for conversion time. So you must set the conversion time slower than 25 µs.

19.4 Internal Reference Voltage Levels

In the ADC function block, the analog input voltage level is compared to the reference voltage. The analog input level must be remained within the range V_{SS} to V_{REF}.

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

19.5 Recommended Circuit

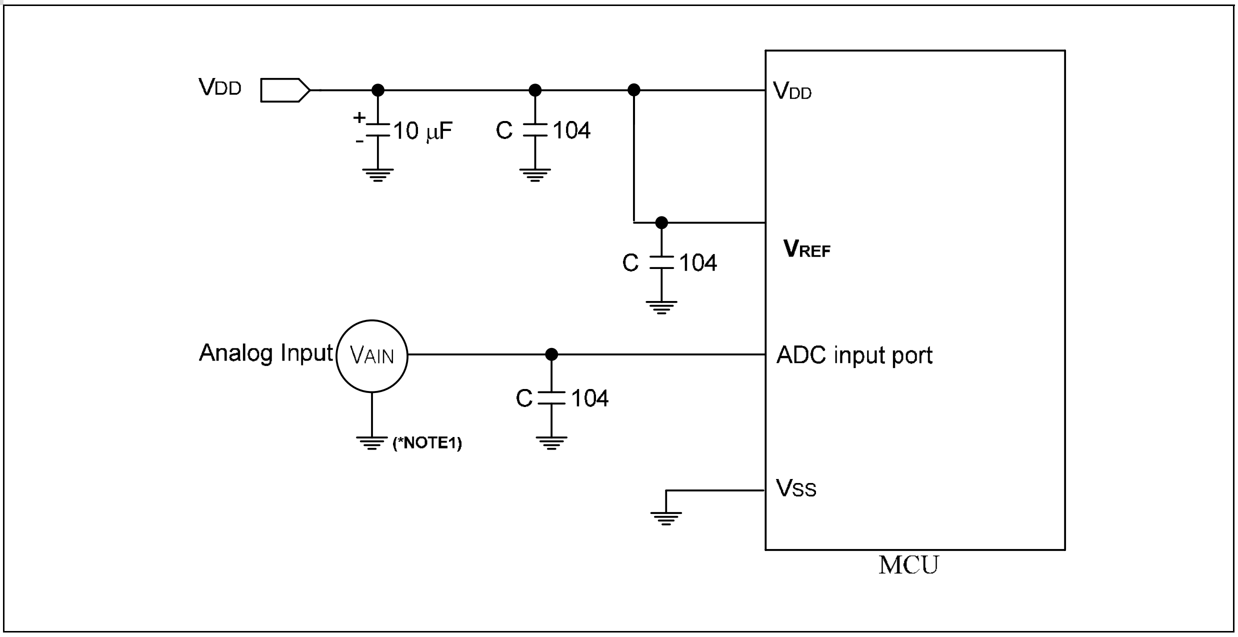


Figure 19-2 Recommended A/D Converter Circuit

Note : Lay out the GND of V_{AIN} as close as possible to the power source.

20. RESET

20.1 Reset Process

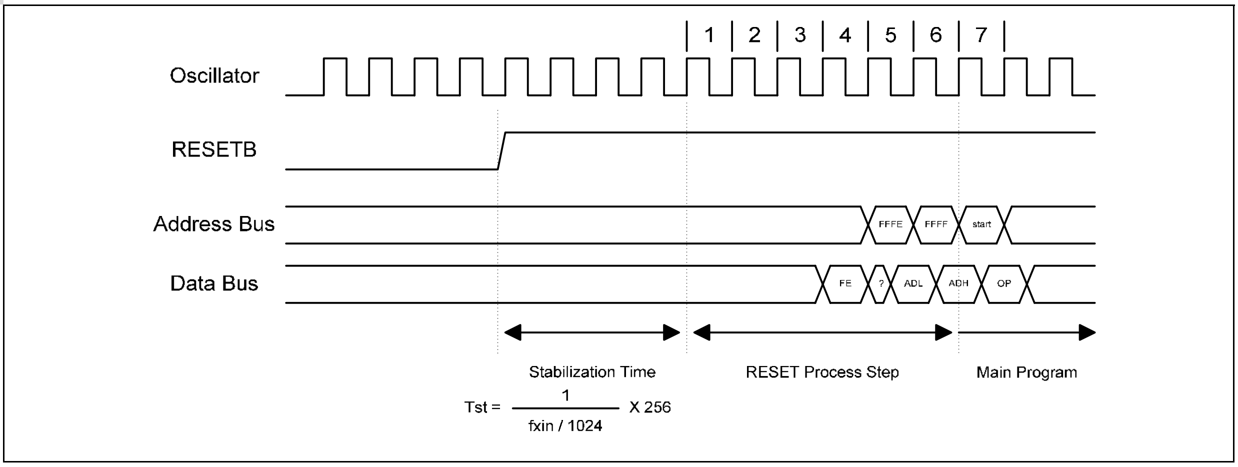


Figure 20-1 Timing Diagram After Reset

When the reset event is occurred, there is a 'stabilization time' at the beginning. This time is counted from 00h to FFh by BIT. So it takes 1/(fxin/1024) * 256 second.

After that, the 'reset process step' is started. It takes 6 system clock time. At this time, following statuses are initialized.

On- chip Hardware	Initial Value
Program Counter (PC)	high byte = a byte at FFFFh low byte = a byte at FFFEh FFFFh and FFFEh stores the reset vector.
RAM Page Register (PRP)	0
G-flag (G)	0
Operation Mode	OSCS setting of Rom option
Control registers	Initialized by reset values (See '9.6 Control Registers (SFR)' on page 44)
Low Voltage Reset	LVREN setting of Rom option

Table 20-1 Initializing Status by Reset

After that, the main program execution is started from the reset vector address which is stored at FFFFh and FFFFEh.

20.2 Reset Sources

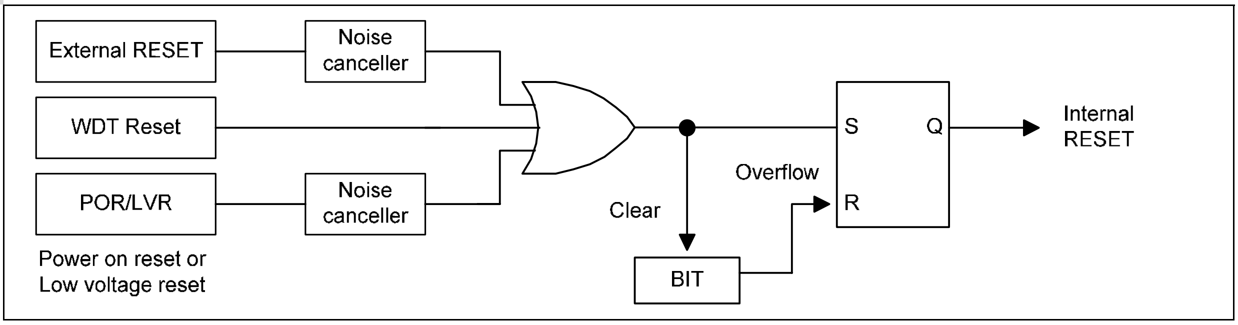


Figure 20-2 Reset Sources Diagram

There are four reset sources in MC81F4104. Those are external reset, watch dog timer reset, power on reset and low voltage reset.

20.3 Reset Circuit

When the external reset is enabled and the input signal of RESET pin is going to low for a while and going to high, the external reset is occurred.(See '7.6 Serial Electric Characteristics' on page23 for more timing information.)

The Reset Pin should be pulled up to VDD with 75kohm resistor, if reset pin voltage is higher than VDD over 2V gap, MCU process self test procedure

It is possible to use a Reset pin like like Figure 20-3.

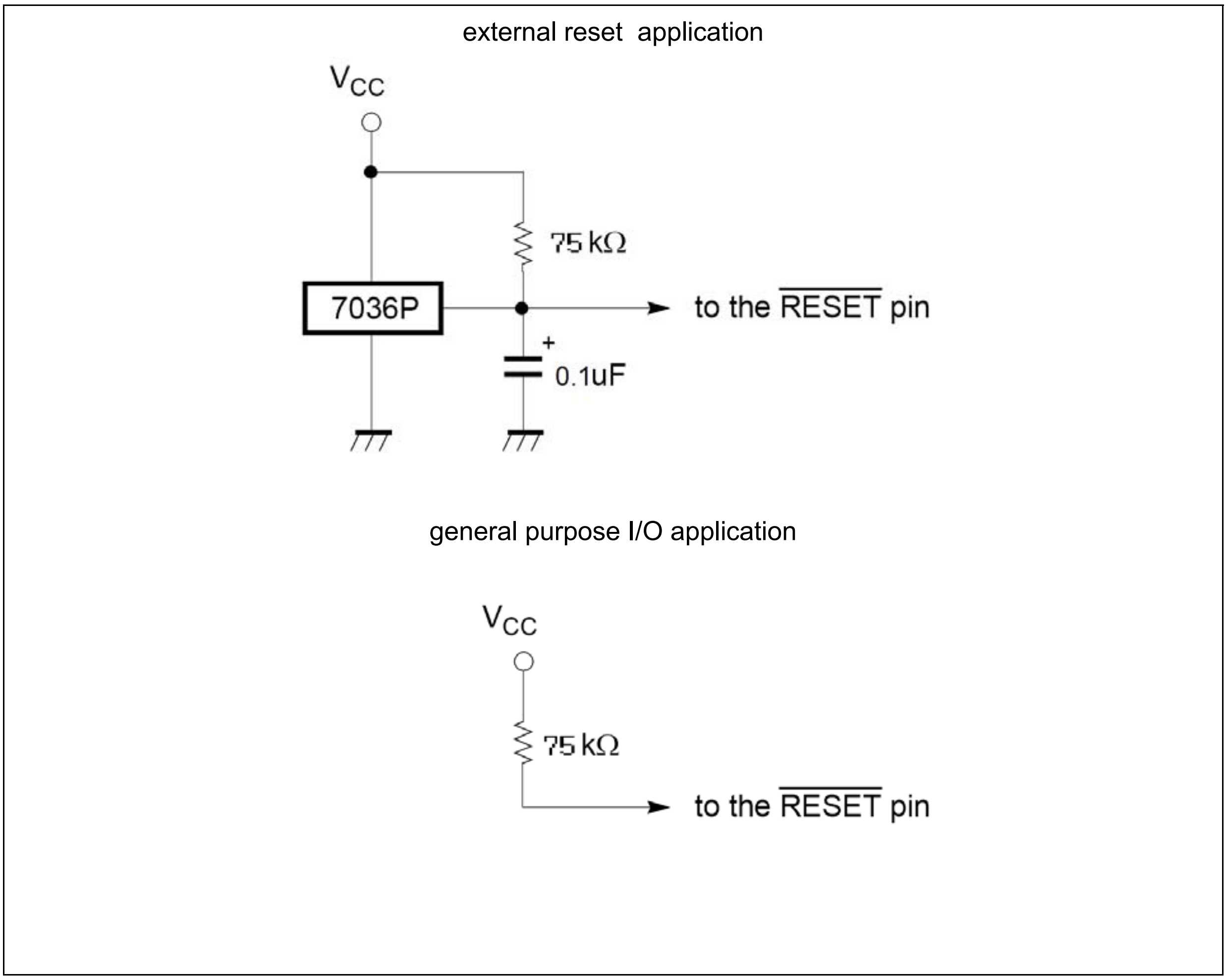

external reset application
V_{CC}
75 kΩ
7036P
to the $\overline{\text{RESET}}$ pin
+
0.1uF
general purpose I/O application
V_{CC}
75 kΩ
to the $\overline{\text{RESET}}$ pin

Figure 20-3 Reset circuit Example

20.4 Watch Dog Timer Reset

See ‘15. WATCH DOG TIMER’ on page 73.

20.5 Power On Reset

There is a internal power on reset circuit internally. We simply call it POR. POR occurs the reset event when VDD is rising over the POR level.

Note that, POR can be enabled and disabled by the PORC register. And default setting is ‘POR enable’. So at the first time power is supplied, POR is working always even external reset is enabled.

PORC

POWER ON RESET CONTROL REGISTER **(00F3H)**

	7	6	5	4	3	2	1	0	
PORC	One byte register								Reset value:00H

POR Enable/Disable	01011010: POR disable Others: POR enable

> **Note :** It is recommended to disable the POR. When POR is enabled, current consumption is increased and, the LVR(Low Voltage Reset) is ignored even the LVR is enabled by the ‘ROM OPTION’.

20.6 Low Voltage Reset

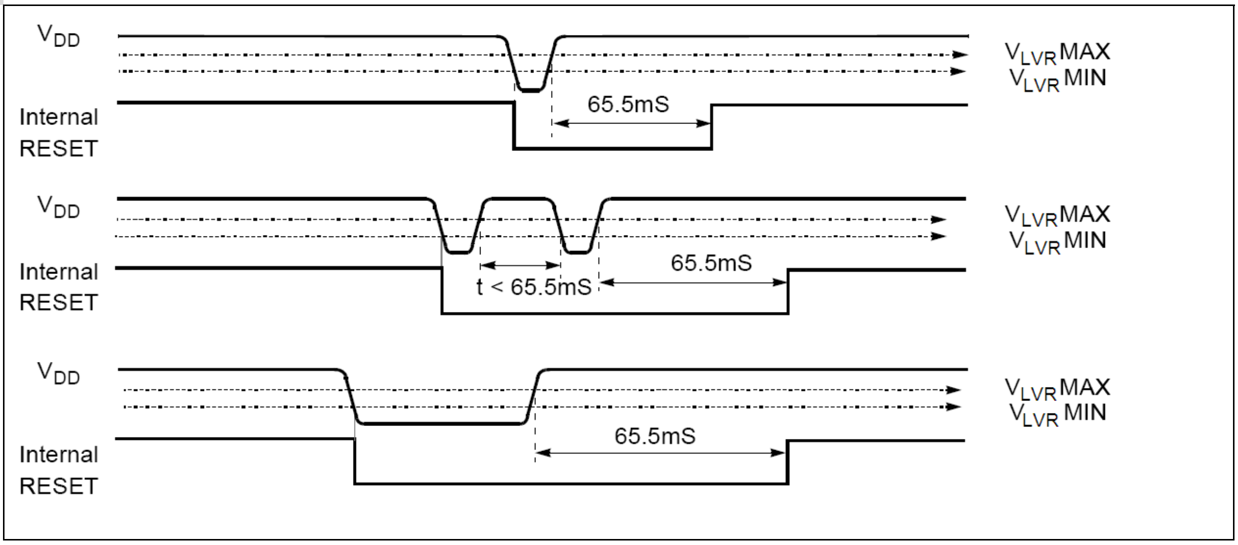


Figure 20-4 LVR Timing Diagram at 4MHz system clock

The low voltage reset occurs the reset event when current VDD is going down under the LVR level.

It is configurable by the rom-option. (See ‘8. ROM OPTION’ on page 34)

If you want to know more detail timing information, see ‘7.8 LVR (Low Voltage Reset) Electrical Characteristics’ on page 26.

21. POWER DOWN OPERATION

In the power-down modes, power consumption is reduced considerably. For applications where power consumption is a critical factor, device provides two kinds of power saving functions, STOP mode and SLEEP mode. Table 21-1 on page 101 shows the status of each Power Saving Mode. SLEEP mode is entered by the SSCR register to “0Fh”. and STOP mode is entered by STOP instruction after the SSCR register to “5Ah”.

21.1 Sleep Mode

In this mode, the internal oscillation circuits remain active. Oscillation continues and peripherals are operated normally but CPU stops. Movement of all peripherals is shown in Table 21-1 on page 101. SLEEP mode is entered by setting the SSCR register to “0Fh”. It is released by Reset or interrupt. To be released by interrupt, interrupt should be enabled before SLEEP mode.

SSCR

STOP AND SLEEP CONTROL REGISTER **00F5H**

	7	6	5	4	3	2	1	0	
SSCR	One byte register								Reset value: 00H
	W	W	W	W	W	W	W	W	

It is used to set the stop or sleep mode.	5Ah : STOP 0Fh : SLEEP

Note: To get into STOP mode, **SSCR must be set to 5AH** just before STOP instruction execution. At STOP mode, Stop & Sleep Control Register (SSCR) value is cleared automatically when released.

To get into SLEEP mode, **SSCR must be set to 0FH**.

Release the SLEEP mode

The exit from SLEEP mode is hardware reset or all interrupts. Reset re-defines all the Control registers but does not change the on-chip RAM.(Be careful, If the code is compiled with RAM clear option, RAM is cleared after reset by ram clear routine. It is possible to disable the RAM clear option by option menu). Interrupts allow both on-chip RAM and Control registers to retain their values. If I-flag = 1, the normal interrupt response takes place. If I-flag = 0, the chip will resume execution starting with the instruction following the SLEEP instruction. It will not vector to interrupt service routine. (refer to Figure 21-3)

When exit from SLEEP mode by reset, enough oscillation stabilization time is required to normal operation. Figure 21-2 shows the timing diagram. When released from the SLEEP mode, the Basic interval timer is activated on wake-up. It is increased from 00H until FFH. The count overflow is set to start normal operation.

Note :

After SLEEP mode, at least one or more NOP instruction for data bus pre-charge time should be written.

```
LDM SSCR,#0FH
NOP                  ;for data bus pre-charge time
NOP                  ;for data bus pre-charge time
```

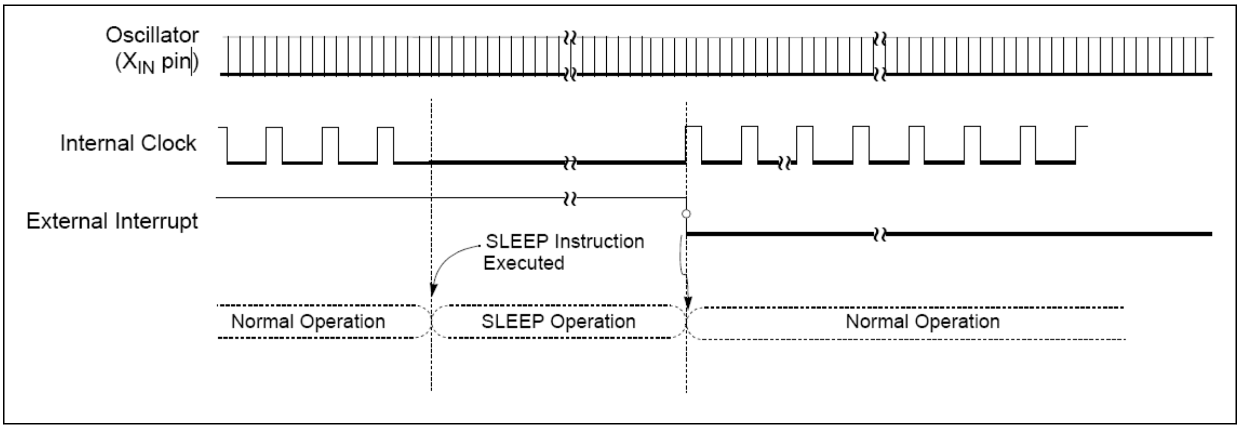


Figure 21-1 SLEEP Mode Release Timing by External Interrupt

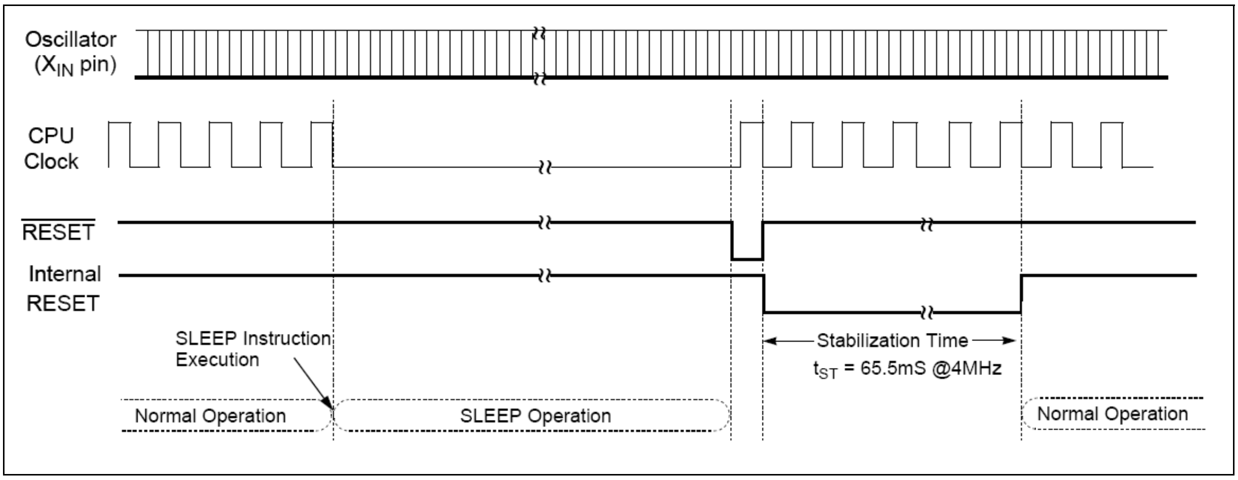


Figure 21-2 Timing of SLEEP Mode Release by Reset

21.2 Stop Mode

In the Stop mode, the main oscillator, system clock and peripheral clock is stopped. With the clock frozen, all functions are stopped, but the on-chip RAM and Control registers are held. The port pins out the values held by their respective port data register, port direction registers. Oscillator stops and the systems internal operations are all held up.

The states of the RAM, registers, and latches valid immediately before the system is put in the STOP state are all held.

The program counter stop the address of the instruction to be executed after the instruction "STOP" which starts the STOP operating mode.

Note :
The Stop mode is activated by execution of STOP instruction after setting the SSCR to “5AH”. (This register should be written by byte operation. If this register is set by bit manipulation instruction, for example "set1" or "clr1" instruction, it may be undesired operation)

In the Stop mode of operation, V_{DD} can be reduced to minimize power consumption. Care must be taken, however, to ensure that V_{DD} is not reduced before the Stop mode is invoked, and that V_{DD} is restored to its normal operating level, before the Stop mode is terminated.

The reset should not be activated before V_{DD} is restored to its normal operating level, and must be held active long enough to allow the oscillator to restart and stabilize.

Note :
After STOP instruction, at least two or more NOP instruction should be written.
Ex)

```
        LDM CKCTLR,#0FH             ;more than 20ms
        LDM SSCR,#5AH
        STOP
        NOP                         ;for stabilization time
        NOP                         ;for stabilization time
```

In the STOP operation, the dissipation of the power associated with the oscillator and the internal hardware is lowered; however, the power dissipation associated with the pin interface (depending on the external circuitry and program) is not directly determined by the hardware operation of the STOP feature. This point should be little current flows when the input level is stable at the power voltage level (V_{DD}/V_{SS}); however, when the input level gets higher than the power voltage level (by approximately 0.3 to 0.5V), a current begins to flow. Therefore, if cutting off the output transistor at an I/O port puts the pin signal into the high-impedance state, a current flow across the ports input transistor, requiring to fix the level by pull-up or other means.

Release the STOP mode

The source for exit from STOP mode is hardware reset, external interrupt, Timer(EC2,3). Reset re-defines all the Control registers but does not change the on-chip RAM. External interrupts allow both on-chip RAM and Control registers to retain their values.

If I-flag = 1, the normal interrupt response takes place. If I-flag = 0, the chip will resume execution starting with the instruction following the STOP instruction. It will not vector to interrupt service routine.

(refer to Figure 21-3) When exit from Stop mode by external interrupt, enough oscillation stabilization time is required to normal operation. Figure 21-4 shows the timing diagram. When released from the Stop mode, the Basic interval timer is activated on wake-up. It is increased from 00H until FFH. The count overflow is set to start normal operation. Therefore, before STOP instruction, user must be set its relevant prescaler divide ratio to have long enough time (more than 20msec). This guarantees that oscillator has started and stabilized. By reset, exit from Stop mode is shown in Figure 21-5.

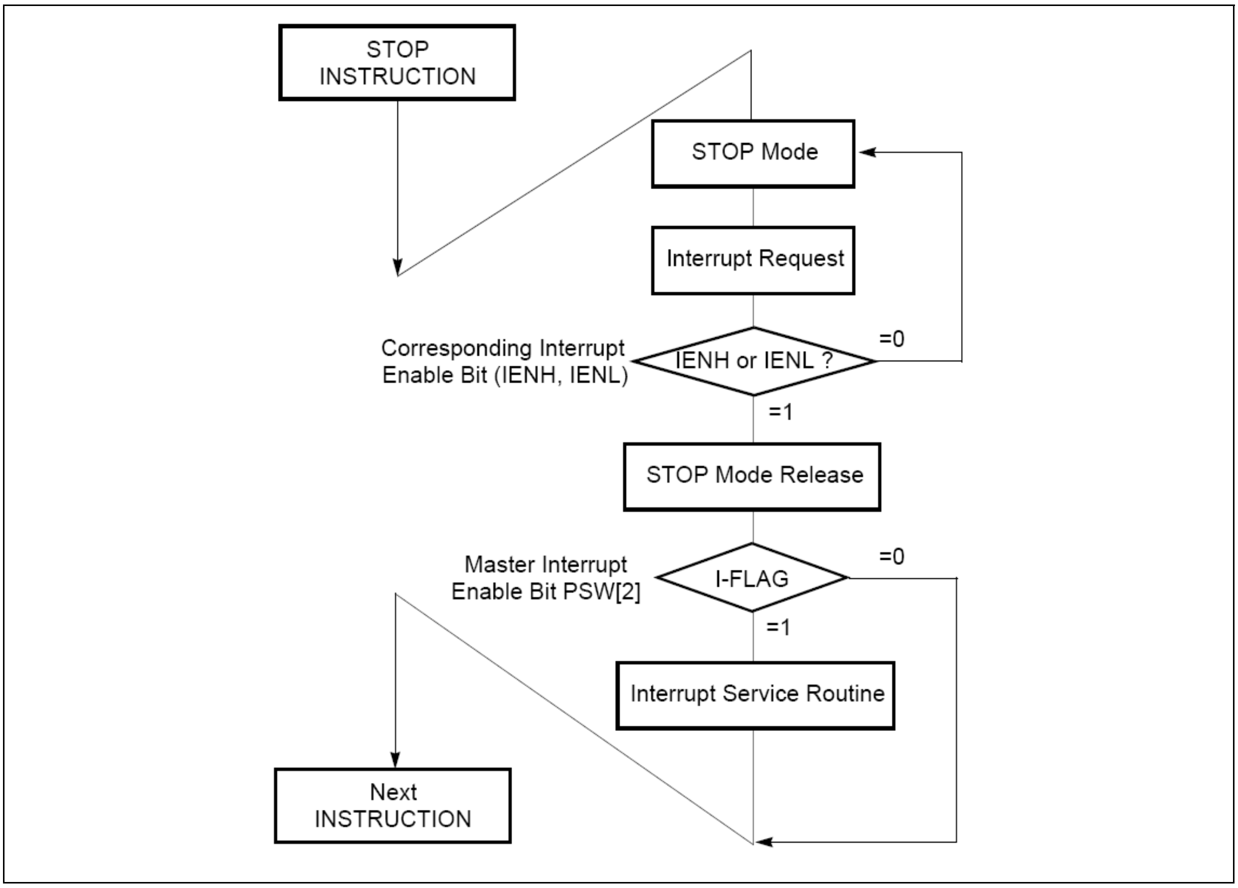
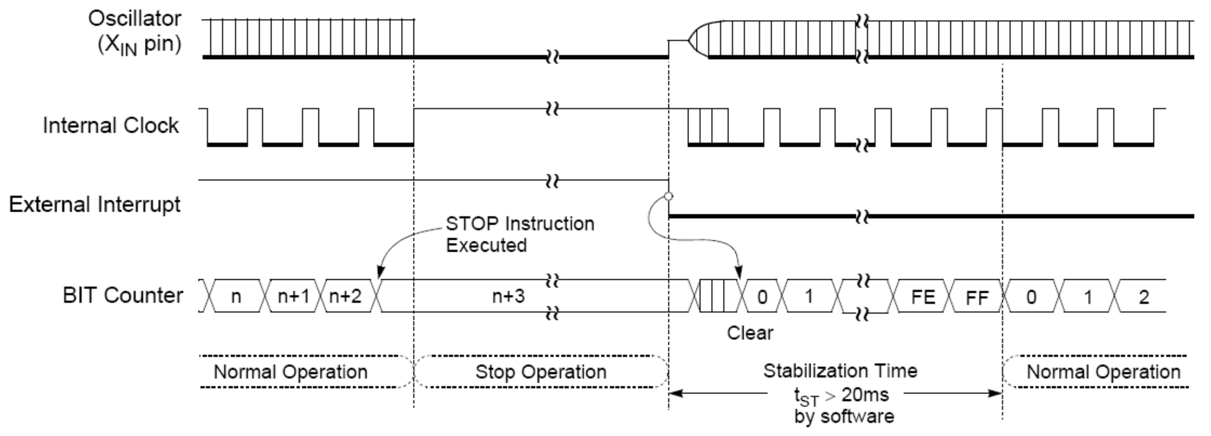

STOP INSTRUCTION
STOP Mode
Interrupt Request
Corresponding Interrupt Enable Bit (IENH, IENL)
IENH or IENL ?
=0
=1
STOP Mode Release
Master Interrupt Enable Bit PSW[2]
I-FLAG
=0
=1
Interrupt Service Routine
Next INSTRUCTION

Figure 21-3 STOP Releasing Flow by Interrupts



Before executing Stop instruction, Basic Interval Timer must be set properly by software to get stabilization time which is longer than 20ms.

Figure 21-4 STOP Mode Release Timing by External Interrupt

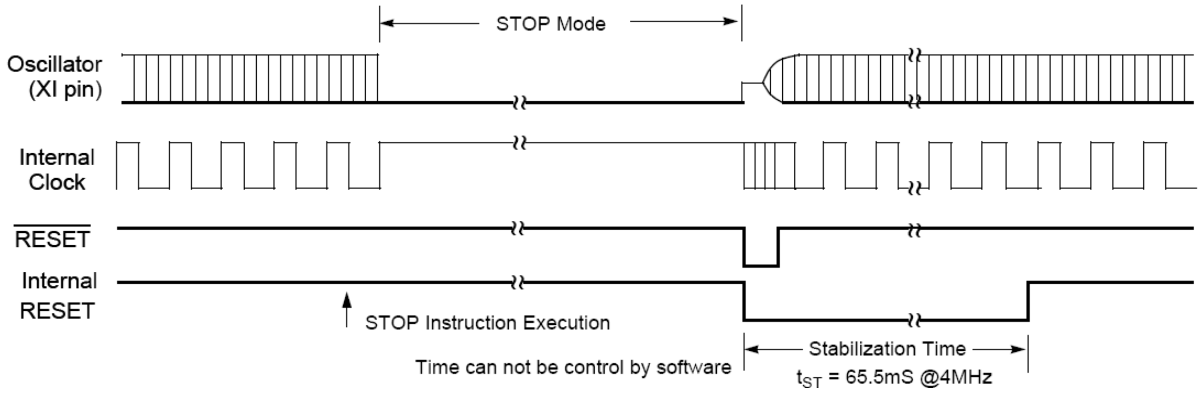


Figure 21-5 Timing of STOP Mode Release by Reset

21.3 Sleep vs Stop

Peripheral	STOP Mode	SLEEP Mode
CPU	Stop	Stop
RAM	Retain	Retain
Basic Interval Timer	Stop	Operates Continuously
Watchdog Timer	Stop	Operates Continuously
Timer/Counter	Stop (The event counter can operate normally)	Operates Continuously
Buzzer, ADC	Stop	Operates Continuously
Main Oscillator	Stop	Oscillation
I/O Ports	Retain	Retain
Control Registers	Retain	Retain
Prescaler	Retain	Retain
Address Data Bus	Retain	Retain
Release Source	Reset, Timer(EC2/3) , External Interrupt	Reset, All Interrupts

Table 21-1 Peripheral Operation During Power Saving Mode

21.4 Changing the stabilizing time

After reset or wake up from the stop/sleep mode, there is a stabilizing time to make sure the system oscillation is stabilized. Actually the stabilizing time is the basic interval timer's one cycle time.

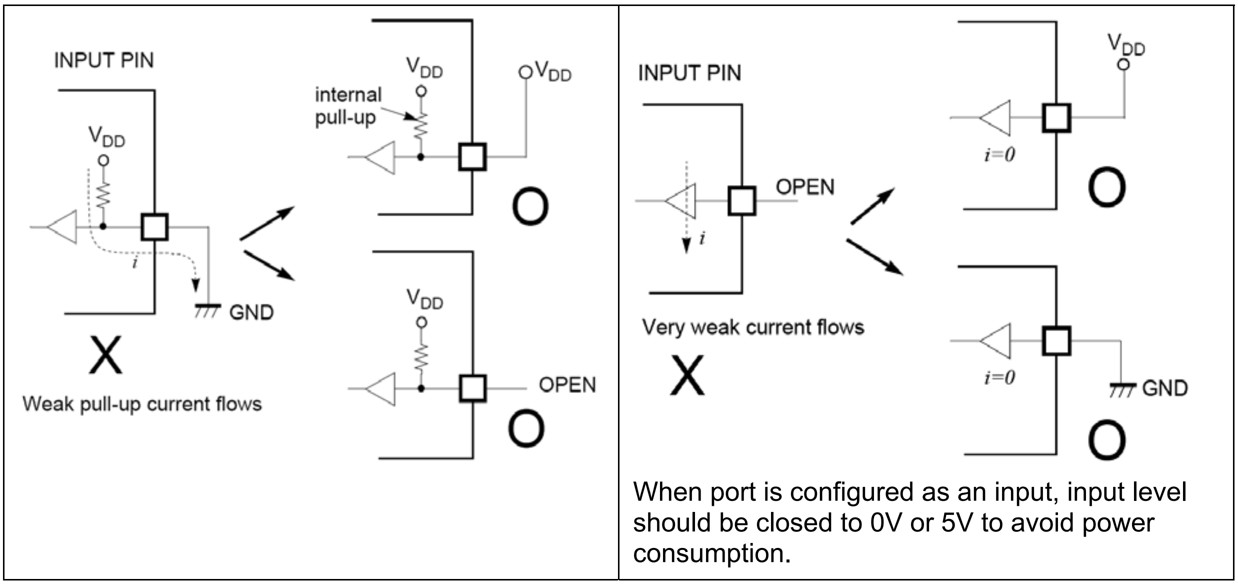
So it is adjustable by changing the basic interval timer's clock division.(See chapter '14.BASIC INTERVAL TIMER' at page 70 to know how to change the basic interval timer's clock division.)

It is useful to reduce the power consumption in battery operation with stop/sleep mode. In the battery operation, reducing normal operation time is the key-point to reducing the power consumption.

Note that, it is not possible after reset. Because after reset, the control registers are initialized.

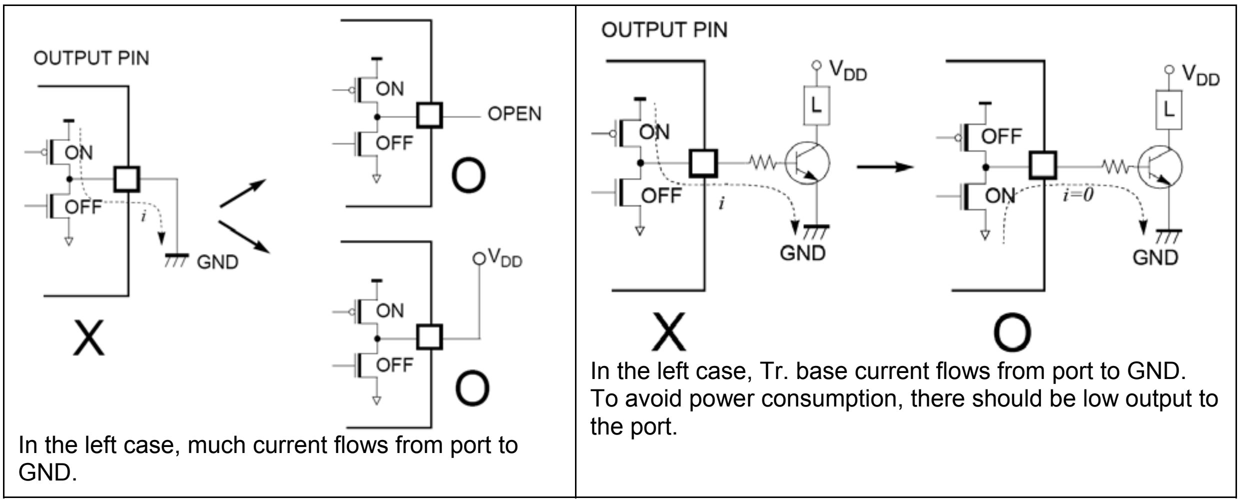
21.5 Minimizing Current Consumption

The Stop mode is designed to reduce power consumption. To minimize current drawn during Stop mode, the user should turnoff output drivers that are sourcing or sinking current, if it is practical.



When port is configured as an input, input level should be closed to 0V or 5V to avoid power consumption.

Figure 21-6 Application Example of Unused Input Port



In the left case, much current flows from port to GND.

In the left case, Tr. base current flows from port to GND. To avoid power consumption, there should be low output to the port.

Figure 21-7 Application Example of Unused Output Port

Note :

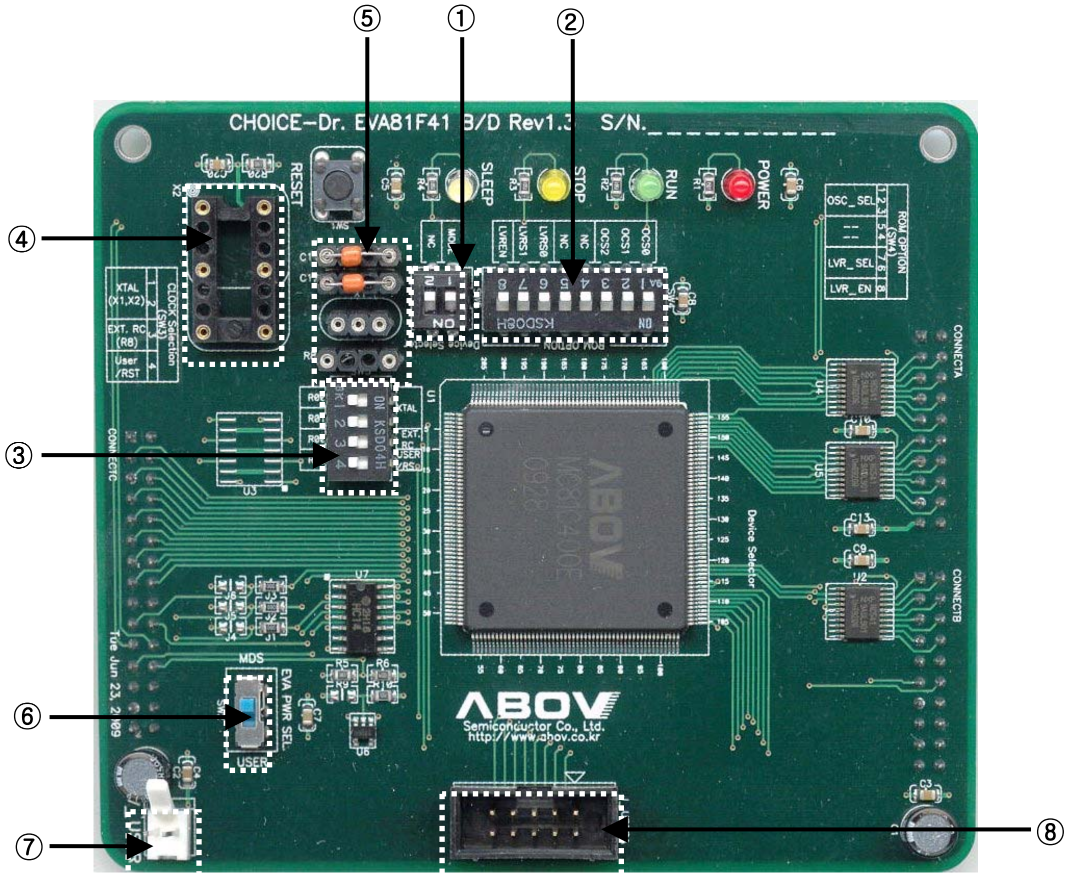
In the STOP operation, the power dissipation associated with the oscillator and the internal hardware is lowered; however, the power dissipation associated with the pin interface (depending on the external circuitry and program) is not directly determined by the hardware operation of the STOP feature. This point should be little current flows when the input level is stable at the power voltage level (V_{DD}/V_{SS}); however, when the input level becomes higher than the power voltage level (by approximately 0.3V), a current begins to flow. Therefore, if cutting off the output transistor at an I/O port puts the pin signal into the high impedance state, a current flow across the ports input transistor, requiring it to fix the level by pull-up or other means.

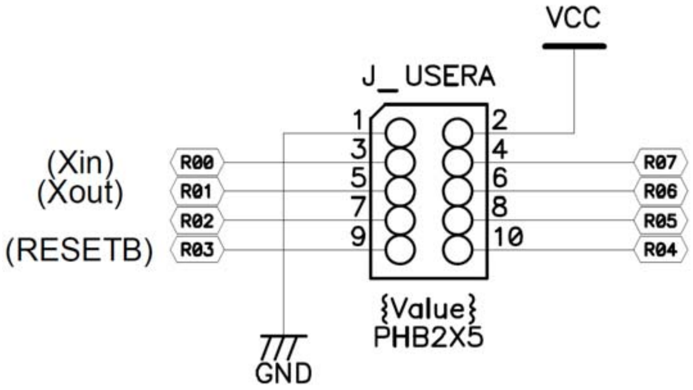
It should be set properly in order that current flow through port doesn't exist.

First consider the port setting to input mode. Be sure that there is circuit. In input mode, the pin impedance viewing from external MCU is very high that the current doesn’t flow. But input voltage level should be V_{SS} or V_{DD}. Be careful that if unspecified voltage, i.e. if uncertain voltage level (not V_{SS} or V_{DD}) is applied to input pin, there can be little current (max. 1mA at around 2V) flow.

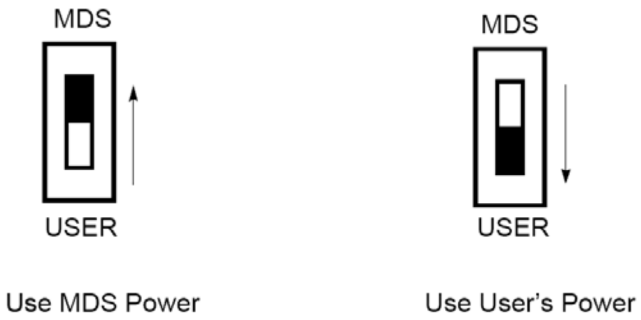
If it is not appropriate to set as an input mode, then set to output mode considering there is no current flow. The port setting to High or Low is decided by considering its relationship with external circuit. For example, if there is external pull-up resistor then it is set to output mode, i.e. to High, and if there is external pull-down register, it is set to low.

22. EMULATOR


⑤
①
②
④
③
⑥
⑦
⑧
CHOICE-Dr. EVA81F41 B/D Rev1.3 S/N.
RESET
SLEEP
STOP
RUN
POWER
MDS
EVA PWR SEL
USER
ABOV
Semiconductor Co., Ltd.
http://www.abov.co.kr


VCC
J_USERA
1
2
3
4
5
6
7
8
9
10
(Xin)
(Xout)
(RESETB)
R00
R01
R02
R03
R07
R06
R05
R04
{Value}
PHB2X5
GND

Mark	Name	Description
①	SW5.3 - MODE	It is used for developing emulator. So, user must turn it off always.
	SW5.4	Not Connected
②	SW4.1 – OSCS.0	Rom Option bit 0~2 : OSC Selection bits (On : 1, Off : 0) 000: External RC 001: Internal RC; 4MHz 010: Internal RC; 2MHz 011: Internal RC; 1MHz 100: Internal RC; 8MHz 101: Not available 110: Not available 111: Crystal/ceramic oscillator
	SW4.2 – OSCS.1	
	SW4.3 – OSCS.2	
	SW4.4	Not Connected
	SW4.5	Not Connected
	SW4.6 – LVRS.0	Rom Option bit 5~6 : Low Voltage Reset Level Selection bit (On: 1, Off : 0) 00: 2.4V 10: 3.0V 01: 2.7V 11: 4.0V
	SW4.7 – LVRS.1	
	SW4.8 – LVREN	Rom Option bit 7 : Low Voltage Reset Enable bit On : (1) Disable (RESETB) Off : (0) Enable (R35)
③	SW3.1 – R00	On : Connect the XTAL to R00/XIN pin Off : Disconnect
	SW3.2 – R01	On : Connect the XTAL to R01/XOUT pin Off : Disconnect
	SW3.3 – R00	On : Connect the EXT.RC to R00/XIN pin Off : Disconnect
	SW3.4 – R03	On : Connect the Reset to R03/Reset pin Off : Disconnect
④	X2	A Oscillator socket
⑤	X1	A Crystal/Resonator socket
	C11	A capacitor socket for crystal
	C12	A capacitor socket for crystal
	R8	Register socket for External RC Oscillator

Mark	Name	Description
⑥	SW2 – EVA PWR SEL	Eva.Board power source selection switch  MDS USER MDS USER Use MDS Power Use User's Power User's power source is supplied from the connector V_USER(⑦) which is described below.
⑦	V_USER	A connector for power source which can be used for Eva.Board.
⑧	J_USERA	A connecter for target system.

Note :
Only GND is connected between Eva.Board and target system. VDD is not connected.
So, the target system is required it's own power source.

Using 'V_USER' is not recommended. It's own power source is more stable. Besides, Choice-Dr can change the VDD level it self. There is a switch which changes the VDD level at the bottom of the Choice-Dr hardware.(But old version of Choice-Dr hardware dose not support it)

23. IN SYSTEM PROGRAMMING

23.1 Getting Started

The In-System Programming (ISP) is an ability to program the code into the MCU while it is installed in a complete system.

USB_SIO_ISP uses both USB to communicate with PC and SIO to communicate with MCU. That is why we call it as ‘USB_SIO_ISP’. In fact there are another ISP types. So remember that all MC81F4xxx series use ‘USB_SIO_ISP’.

Here is a procedure to use ISP.

1. Power off the target system.

   If you use the RESET/Vpp pin as an output mode, power on timing is very important.
   So you must read ‘Entering ISP mode at power on time’ and strictly obey the procedure.

2. Install the USB_SIO_ISP software. (It is required at only first time)

   1) Download the ISP software from http://www.abov.co.kr

   2) Unzip the downloaded file and connect the USB_SIO_ISP board.

   3) Install the driver for USB_SIO_ISP. (There is a driver file in the zip file.)

3. Make sure the hardware condition is satisfied. And connect the ISP cable.

   See ‘23.3 Hardware Conditions to Enter the ISP Mode’ page 110,

4. Run the software and select a device.

   All commands are enabled after select the device.

5. Power on the target system.

   If you use the RESET/Vpp pin as an input mode, power on timing is not that important.
   But make sure the power is turned-on before execute the ISP commands.

6. Execute ISP commands as you want.

   If you want to write a code into your MCU, it is recommendable to do following step.

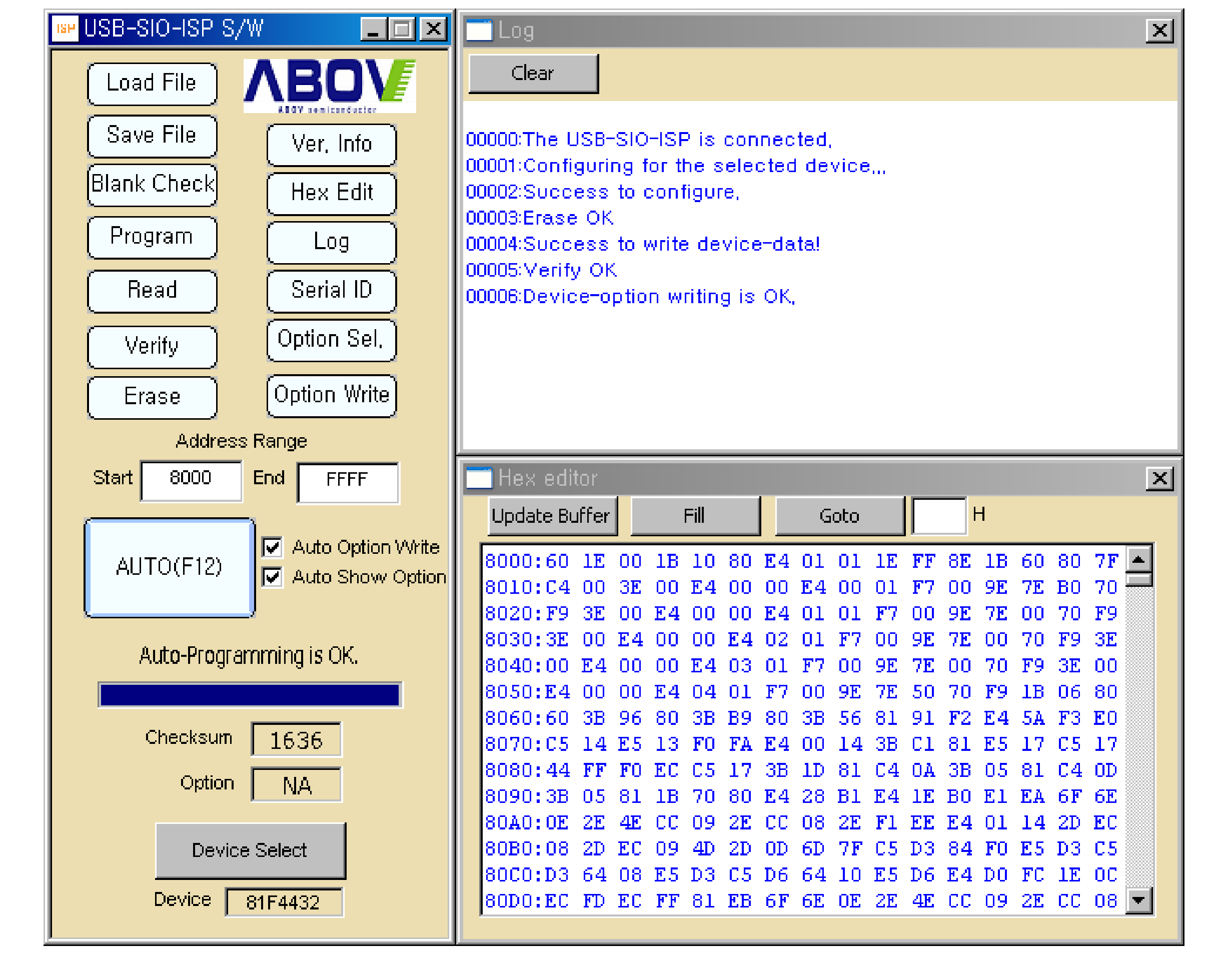
   ‘Load File’ -> ‘Auto’(while ‘Auto Option Write’ and ‘Auto Show Option’ options are enabled).

After finish an ISP command is executed, the MCU enters to normal operation mode automatically. So you can see the system is working right after the ISP command is finished. (‘Auto’ is assumed as one command’)

In fact, it is possible to repeat the step-6 until the hardware condition is changed. But in case of RESET/Vpp pin is used as an output mode, do not repeat step-6. In that case, you must follow the procedure. See ‘Entering ISP mode at power on time’ for more information.

After you change the ‘Rom Option’, you must do power-off and power-on to reflect the changed ‘Rom Option’, even you can repeat the step-6 and see the changed code’s operation without doing it. The MCU reads the ‘Rom option’ when only the ‘power on reset time’.

23.2 Basic ISP S/W Information



Address Range

Start 8000 End FFFF

☑ Auto Option Write

☑ Auto Show Option

Auto-Programming is OK.

Checksum 1636

Option NA

Device 81F4432

```
00000:The USB-SIO-ISP is connected.
00001:Configuring for the selected device...
00002:Success to configure.
00003:Erase OK
00004:Success to write device-data!
00005:Verify OK
00006:Device-option writing is OK.
```

```
8000:60 1E 00 1B 10 80 E4 01 01 1E FF 8E 1B 60 80 7F
8010:C4 00 3E 00 E4 00 00 E4 00 01 F7 00 9E 7E B0 70
8020:F9 3E 00 E4 00 00 E4 01 01 F7 00 9E 7E 00 70 F9
8030:3E 00 E4 00 00 E4 02 01 F7 00 9E 7E 00 70 F9 3E
8040:00 E4 00 00 E4 03 01 F7 00 9E 7E 00 70 F9 3E 00
8050:E4 00 00 E4 04 01 F7 00 9E 7E 50 70 F9 1B 06 80
8060:60 3B 96 80 3B B9 80 3B 56 81 91 F2 E4 5A F3 E0
8070:C5 14 E5 13 F0 FA E4 00 14 3B C1 81 E5 17 C5 17
8080:44 FF F0 EC C5 17 3B 1D 81 C4 0A 3B 05 81 C4 0D
8090:3B 05 81 1B 70 80 E4 28 B1 E4 1E B0 E1 EA 6F 6E
80A0:0E 2E 4E CC 09 2E CC 08 2E F1 EE E4 01 14 2D EC
80B0:08 2D EC 09 4D 2D 0D 6D 7F C5 D3 84 F0 E5 D3 C5
80C0:D3 64 08 E5 D3 C5 D6 64 10 E5 D6 E4 D0 FC 1E 0C
80D0:EC FD EC FF 81 EB 6F 6E 0E 2E 4E CC 09 2E CC 08
```

Figure 23-1 ISP Software

The Figure 23-1 is the USB_SIO_ISP software based on MS-Windows. This software supports only SIO_ISP type devices.

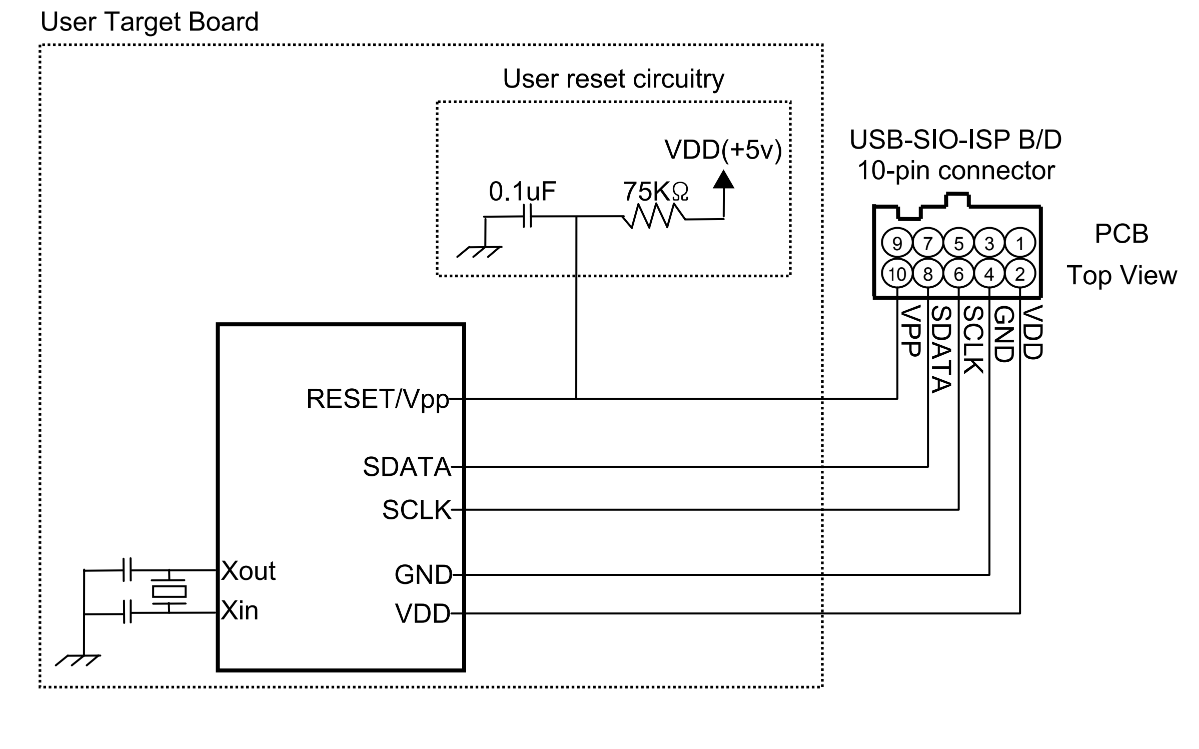
Function	Description
Load File	Load the data from the selected file storage into the memory buffer.
Save File	Save the current data in your memory buffer to a disk storage by using the Intel Motorola HEX format.
Blank Check	Verify whether or not a device is in an erased or unprogrammed state. Program This button enables you to place new data from the memory buffer into the target device.
Program	Write the current data into the MCU.
Read	Read the data in the target MCU into the buffer for examination. The checksum will be displayed on the checksum box.
Verify	Assures that data in the device matches data in the memory buffer. If your device is secured, a verification error is detected.

Erase	Erase the data in your target MCU before programming it.
Option Selection	Set the configuration data of target MCU. The security locking is set with this button.
Option Write	Progam the configuration data of target MCU. The security locking is performed with this button.
AUTO	Following sequence is performed ; 1.Erase 2.Program 3.Verify 4.Option Write
Auto Option Write	Enable the option writing when the 'AUTO' sequence is executing.
Auto Show Option	Enable showing the option window when 'AUTO' button is pressed.
Ver. Info	It shows the version information.
Log	It shows/hides the log windows
Hex Edit	It shows/hides 'Hex editor'. In 'Hex editor' you can modify the currently loaded data.
Fill	Buffer Fill the selected area with a data.
Goto	Display the selected page.
Start ______	Starting address
End ______	End address
Checksum	Display the check sum(Hex decimal) after reading the target device.
Option	It shows currently selected option code in hexadecimal.
Device Select	It is used to select a target device.
Device	It shows currently selected device.

Note: MCU Configuration value is erased after erase operation. It must be configured to match with user target board. Otherwise, it is failed to enter ISP mode, or its operation is not desirable.

23.3 Hardware Conditions to Enter the ISP Mode

Anytime RESET/ Vpp pin goes +9V, the MCU entering an ISP mode except RESET/Vpp pin is output mode(See note1).



1. If other signals affect SIO communication in ISP mode, disconnect these pins by using a jumper or a switch.

Figure 23-2 Hardware Conditions to Enter the ISP Mode

Note:

1) Using RESET/Vpp pin as an output mode is not recommended even it is possible. Anytime RESET/Vpp pin goes +9v, the MCU entering an ISP mode except RESET/Vpp pin is output mode. If it is output mode, +9v signal is clashing with the output voltage.

So if RESET/Vpp pin is used as an output mode, do not try to execute any ISP commands when MCU is in normal operation mode. It is allowable when only power on time. See 'Entering ISP mode at power on time' for more information.

2) There is a 10KΩ pull-down register at VPP pin in the ISP Board. That is why 75KΩ register is suggested for R/C reset circuit. So those two register makes a voltage divider circuit when ISP board is connected. So the VPP level can't go down to low level status if the register of reset circuit value is too small. Otherwise, if the register value is too large the capacitor value also changed and the reset circuit's characteristics also changed.

23.4 Entering ISP mode at power on time

Basically anytime +9v signal is forced to RESET/Vpp pin, the MCU is entering into ISP mode. But it makes trouble when the RESET/Vpp pin is output mode. Because the +9v signal is clashing with the port's output voltage.

But it is possible to enter the ISP mode at the power on time even RESET/Vpp pin is used as an output mode. There is an oscillator stabilizing time when power is turn on. While in the time RESET/Vpp pin is in input mode even it is used as an output mode in operation time.

A proper procedure is required to make sure that ISP board catch the oscillator stabilizing time to enter the ISP mode. See following procedure.

1. Power off the target system.
2. Configure the target system as ISP mode.
3. Attach a ISP B/D into the target system.
4. Run the ISP S/W
5. Select the target device.
6. Power on the target system.
7. Execute ISP commands as you want.

Note :

Power on the target system after select the target device is essential. Because when target device is selected, ISP board is getting ready to catch the proper timing to rise the Vpp(+9v) signal.

23.5 USB-SIO-ISP Board

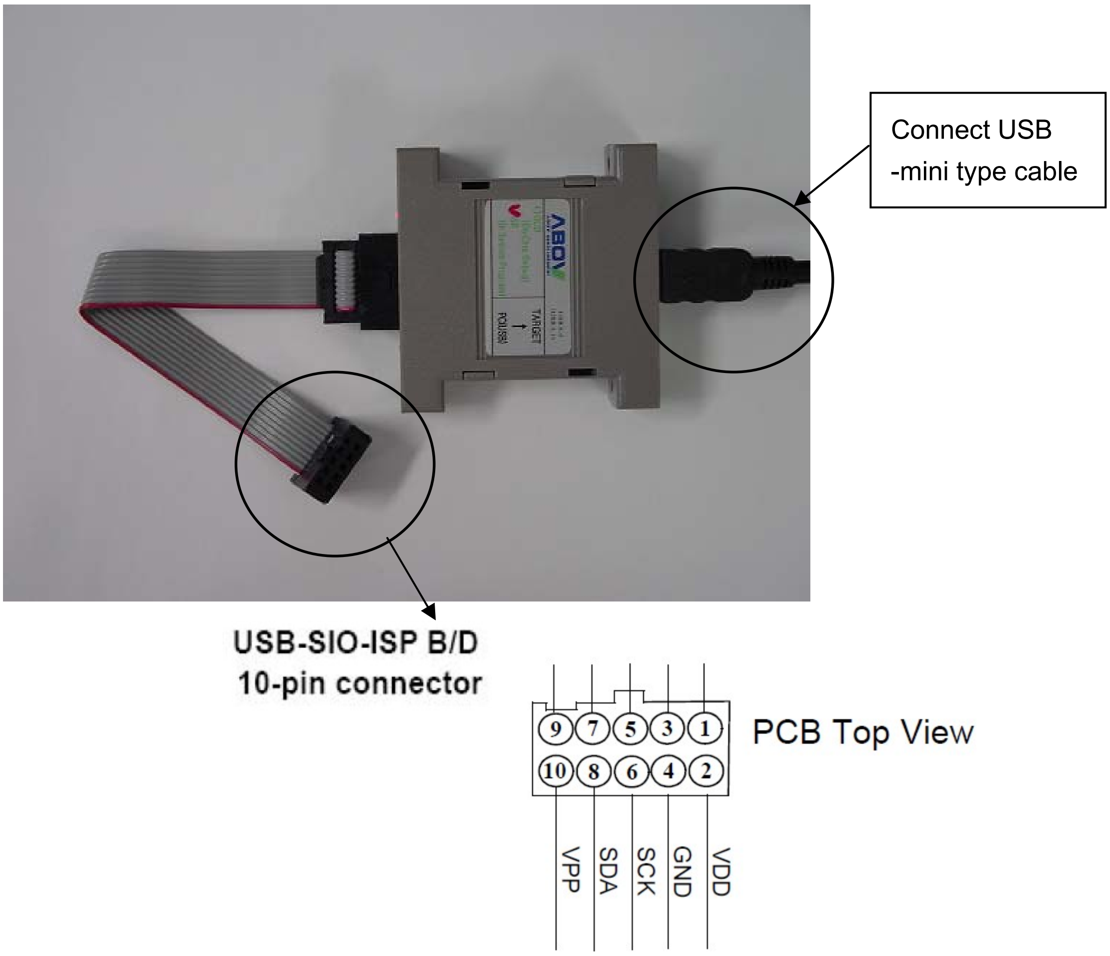
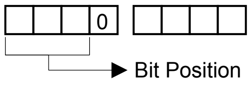
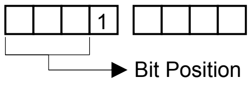


Figure 23-3 USB-SIO-ISP Board

24. INSTRUCTION SET

24.1 Terminology List

A	Accumulator
X	X - register
Y	Y - register
PSW	Program Status Word
#imm	8-bit Immediate data
dp	Direct Page Offset Address
!abs	Absolute Address
[ ]	Indirect expression
{ }	Register Indirect expression
{ }+	Register Indirect expression, after that, Register auto-increment
.bit	Bit Position
A.bit	Bit Position of Accumulator
dp.bit	Bit Position of Direct Page Memory
M.bit	Bit Position of Memory Data (000H~0FFFFH)
rel	Relative Addressing Data
upage	U-page (0FF00H~0FFFFH) Offset Address
n	Table CALL Number (0~15)
+	Addition
x	Upper Nibble Expression in Opcode when it is even number (bit7~bit5, bit4=0) 
y	Upper Nibble Expression in Opcode when it is odd number (bit7~bit5, bit4=1) 
–	Subtraction
×	Multiplication
/	Division
( )	Contents Expression
∧	AND
∨	OR
⊕	Exclusive OR
~	NOT
←	Assignment / Transfer / Shift Left

→	Shift Right
↔	Exchange
=	Equal
≠	Not Equal

24.2 Instruction Map

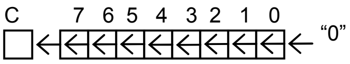
LOW HIGH	00000 00	00001 01	00010 02	00011 03	00100 04	00101 05	00110 06	00111 07	01000 08	01001 09	01010 0A	01011 0B	01100 0C	01101 0D	01110 0E	01111 0F
000	-	SET1 dp.bit	BBS A.bit,rel	BBS dp.bit,rel	ADC #imm	ADC dp	ADC dp+X	ADC !abs	ASL A	ASL dp	TCALL 0	SETA1 .bit	BIT dp	POP A	PUSH A	BRK
001	CLRC	″	″	″	SBC #imm	SBC dp	SBC dp+X	SBC !abs	ROL A	ROL dp	TCALL 2	CLRA1 .bit	COM dp	POP X	PUSH X	BRA rel
010	CLRG	″	″	″	CMP #imm	CMP dp	CMP dp+X	CMP !abs	LSR A	LSR dp	TCALL 4	NOT1 M.bit	TST dp	POP Y	PUSH Y	PCALL Upage
011	DI	″	″	″	OR #imm	OR dp	OR dp+X	OR !abs	ROR A	ROR dp	TCALL 6	OR1 OR1B	CMPX dp	POP PSW	PUSH PSW	RET
100	CLRV	″	″	″	AND #imm	AND dp	AND dp+X	AND !abs	INC A	INC dp	TCALL 8	AND1 AND1B	CMPY dp	CBNE dp+X	TXSP	INC X
101	SETC	″	″	″	EOR #imm	EOR dp	EOR dp+X	EOR !abs	DEC A	DEC dp	TCALL 10	EOR1 EOR1B	DBNE dp	XMA dp+X	TSPX	DEC X
110	SETG	″	″	″	LDA #imm	LDA dp	LDA dp+X	LDA !abs	TXA	LDY dp	TCALL 12	LDC LDCB	LDX dp	LDX dp+Y	XCN	DAS (N/A)
111	EI	″	″	″	LDM dp,#imm	STA dp	STA dp+X	STA !abs	TAX	STY dp	TCALL 14	STC M.bit	STX dp	STX dp+Y	XAX	STOP

LOW HIGH	10000 10	10001 11	10010 12	10011 13	10100 14	10101 15	10110 16	10111 17	11000 18	11001 19	11010 1A	11011 1B	11100 1C	11101 1D	11110 1E	11111 1F
000	BPL rel	CLR1 dp.bit	BBC A.bit,rel	BBC dp.bit,rel	ADC {X}	ADC !abs+Y	ADC [dp+X]	ADC [dp]+Y	ASL !abs	ASL dp+X	TCALL 1	JMP !abs	BIT !abs	ADDW dp	LDX #imm	JMP [!abs]
001	BVC rel	″	″	″	SBC {X}	SBC !abs+Y	SBC [dp+X]	SBC [dp]+Y	ROL !abs	ROL dp+X	TCALL 3	CALL !abs	TEST !abs	SUBW dp	LDY #imm	JMP [dp]
010	BCC rel	″	″	″	CMP {X}	CMP !abs+Y	CMP [dp+X]	CMP [dp]+Y	LSR !abs	LSR dp+X	TCALL 5	MUL	TCLR1 !abs	CMPW dp	CMPX #imm	CALL [dp]
011	BNE rel	″	″	″	OR {X}	OR !abs+Y	OR [dp+X]	OR [dp]+Y	ROR !abs	ROR dp+X	TCALL 7	DBNE Y	CMPX !abs	LDYA dp	CMPY #imm	RETI
100	BMI rel	″	″	″	AND {X}	AND !abs+Y	AND [dp+X]	AND [dp]+Y	INC !abs	INC dp+X	TCALL 9	DIV	CMPY !abs	INCW dp	INC Y	TAY
101	BVS rel	″	″	″	EOR {X}	EOR !abs+Y	EOR [dp+X]	EOR [dp]+Y	DEC !abs	DEC dp+X	TCALL 11	XMA {X}	XMA dp	DECW dp	DEC Y	TYA
110	BCS rel	″	″	″	LDA {X}	LDA !abs+Y	LDA [dp+X]	LDA [dp]+Y	LDY !abs	LDY dp+X	TCALL 13	LDA {X}+	LDX !abs	STYA dp	XAY	DAA (N/A)
111	BEQ	″	″	″	STA	STA	STA	STA	STY	STY	TCALL	STA	STX	CBNE	XYX	NOP

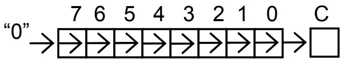
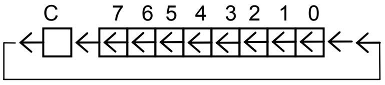
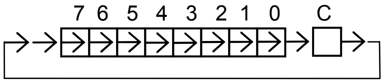
	rel				{X}	!abs+Y	[dp+X]	[dp]+Y	!abs	dp+X	15	{X}+	!abs	dp		

24.3 Instruction Set

Arithmetic / Logic

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
1	ADC #imm	04	2	2	Add with carry. A ← (A) + (M) + C	NV--H-ZC
2	ADC dp	05	2	3		
3	ADC dp + X	06	2	4		
4	ADC !abs	07	3	4		
5	ADC !abs + Y	15	3	5		
6	ADC [dp + X]	16	2	6		
7	ADC [dp] + Y	17	2	6		
8	ADC { X }	14	1	3		
9	AND #imm	84	2	2	Logical AND A ← (A) ∧ (M)	N-----Z-
10	AND dp	85	2	3		
11	AND dp + X	86	2	4		
12	AND !abs	87	3	4		
13	AND !abs + Y	95	3	5		
14	AND [dp + X]	96	2	6		
15	AND [dp] + Y	97	2	6		
16	AND { X }	94	1	3		
17	ASL A	08	1	2	Arithmetic shift left 	N-----ZC
18	ASL dp	09	2	4		
19	ASL dp + X	19	2	5		
20	ASL !abs	18	3	5		
21	CMP #imm	44	2	2	Compare accumulator contents with memory contents (A) - (M)	N-----ZC
22	CMP dp	45	2	3		
23	CMP dp + X	46	2	4		
24	CMP !abs	47	3	4		
25	CMP !abs + Y	55	3	5		

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
26	CMP [dp + X]	56	2	6		
27	CMP [dp] + Y	57	2	6		
28	CMP { X }	54	1	3		
29	CMPX #imm	5E	2	2	Compare X contents with memory contents (X) - (M)	N-----ZC
30	CMPX dp	6C	2	3		
31	CMPX !abs	7C	3	4		
32	CMPY #imm	7E	2	2	Compare Y contents with memory contents (Y) - (M)	N-----ZC
33	CMPY dp	8C	2	3		
34	CMPY !abs	9C	3	4		
35	COM dp	2C	2	4	1's Complement : (dp) ← ~(dp)	N-----Z-
36	DAA	-	-	-	Unsupported	-
37	DAS	-	-	-	Unsupported	-
38	DEC A	A8	1	2	Decrement M ← (M) - 1	N-----Z-
39	DEC dp	A9	2	4		
40	DEC dp + X	B9	2	5		
41	DEC !abs	B8	3	5		
42	DEC X	AF	1	2		
43	DEC Y	BE	1	2		
44	DIV	9B	1	12	Divide : YA/X Q:A, R:Y	NV--H-Z-
45	EOR #imm	A4	2	2	Exclusive OR A ← (A) ⊕ (M)	N-----Z-
46	EOR dp	A5	2	3		
47	EOR dp + X	A6	2	4		
48	EOR !abs	A7	3	4		
49	EOR !abs + Y	B5	3	5		
50	EOR [dp + X]	B6	2	6		
51	EOR [dp] + Y	B7	2	6		
52	EOR { X }	B4	1	3		
53	INC A	88	1	2	Increment	N-----Z-

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
54	INC dp	89	2	4	M ← (M) + 1	
55	INC dp + X	99	2	5		
56	INC !abs	98	3	5		
57	INC X	8F	1	2		
58	INC Y	9E	1	2		
59	LSR A	48	1	2	Arithmetic shift left 	N-----ZC
60	LSR dp	49	2	4		
61	LSR dp + X	59	2	5		
62	LSR !abs	58	3	5		
63	MUL	5B	1	9	Multiply : YA ← Y × A	N-----Z-
64	OR #imm	64	2	2	Logical OR A ← (A) ∨ (M)	N-----Z-
65	OR dp	65	2	3		
66	OR dp + X	66	2	4		
67	OR !abs	67	3	4		
68	OR !abs + Y	75	3	5		
69	OR [dp + X]	76	2	6		
70	OR [dp] + Y	77	2	6		
71	OR { X }	74	1	3		
72	ROL A	28	1	2	Rotate left through carry 	N-----ZC
73	ROL dp	29	2	4		
74	ROL dp + X	39	2	5		
75	ROL !abs	38	3	5		
76	ROR A	68	1	2	Rotate right through carry 	N-----ZC
77	ROR dp	69	2	4		
78	ROR dp + X	79	2	5		
79	ROR !abs	78	3	5		
80	SBC #imm	24	2	2	Subtract with carry A ← (A) - (M) - ~(C)	NV--HZC
81	SBC dp	25	2	3		

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
82	SBC dp + X	26	2	4		
83	SBC !abs	27	3	4		
84	SBC !abs + Y	35	3	5		
85	SBC [dp + X]	36	2	6		
86	SBC [dp] + Y	37	2	6		
87	SBC { X }	34	1	3		
88	TST dlp	4C	2	3	Test memory contents for negative or zero (dp) – 00H	N-----Z-
89	XCN	CE	1	5	Exchange nibbles within the accumulator A7~A4 ↔ A3~A0	N-----Z-

Register / Memory Operation

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
1	LDA #imm	C4	2	2	Load accumulator A ← (M)	N-----Z-
2	LDA dp	C5	2	3		
3	LDA dp + X	C6	2	4		
4	LDA !abs	C7	3	4		
5	LDA !abs + Y	D5	3	5		
6	LDA [dp + X]	D6	2	6		
7	LDA [dp] + Y	D7	2	6		
8	LDA { X }	D4	1	3		
9	LDA { X }+	DB	1	4	X-register auto-increment : A ← (M), X ← X + 1	
10	LDM dp, #imm	E4	3	5	Load memory with immediate data : (M) ← imm	--------
11	LDX #imm	1E	2	2	Load X-register X ← (M)	N-----Z-
12	LDX dp	CC	2	3		
13	LDX dp + Y	CD	2	4		
14	LDX !abs	DC	3	4		
15	LDY #imm	3E	2	2	Load Y-register Y ← (M)	N-----Z-
16	LDY dp	C9	2	3		
17	LDY dp + Y	D9	2	4		
18	LDY !abs	D8	3	4		
19	STA dp	E5	2	4	Store accumulator contents in memory (M) ← A	--------
20	STA dp + X	E6	2	5		
21	STA !abs	E7	3	5		
22	STA !abs + Y	F5	3	6		
23	STA [dp + X]	F6	2	7		
24	STA [dp] + Y	F7	2	7		
25	STA { X }	F4	1	4		

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
26	STA { X }+	FB	1	4	X-register auto-increment : (M) ← A, X ← X + 1	
27	STX dp	EC	2	4	Store X-register contents in memory (M) ← X	--------
28	STX dp + Y	ED	2	5		
29	STX !abs	FC	3	5		
30	STY dp	E9	2	4	Store Y-register contents in memory (M) ← Y	--------
31	STY dp + X	F9	2	5		
32	STY !abs	F8	3	5		
33	TAX	E8	1	2	Transfer accumulator contents to X-register : X ← A	N-----Z-
34	TAY	9F	1	2	Transfer accumulator contents to Y-register : Y ← A	N-----Z-
35	TSPX	AE	1	2	Transfer stack-pointer contents to X-register : X ← sp	N-----Z-
36	TXA	C8	1	2	Transfer X-register contents to accumulator : A ← X	N-----Z-
37	TXSP	8E	1	2	Transfer X-register contents to stack-pointer : sp ← X	N-----Z-
38	TYA	BF	1	2	Transfer Y-register contents to accumulator : A ← Y	N-----Z-
39	XAX	EE	1	4	Exchange X-register contents with accumulator : X ↔ A	--------
40	XAY	DE	1	4	Exchange Y-register contents with accumulator : Y ↔ A	--------
41	XMA dp	BC	2	5	Exchange memory contents with accumulator : (M) ↔ A	N-----Z-
42	XMA dp + X	AD	2	6		
43	XMA {X}	BB	1	5		
44	XYX	FE	1	4	Exchange X-register contents with Y-register : X ↔ Y	--------

16 BIT manipulation

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
1	ADDW dp	1D	2	5	16-bits add without carry YA ← (YA) + (dp + 1) (dp)	NV--H-ZC
2	CMPW dp	5D	2	4	Compare YA contents with memory pair contents : (YA) - (dp + 1) (dp)	N-----ZC
3	DECW dp	BD	2	6	Decrement memory pair (dp + 1) (dp) ← (dp + 1) (dp) – 1	N-----Z-
4	INCW dp	9D	2	6	Increment memory pair (dp + 1) (dp) ← (dp + 1) (dp) + 1	N-----Z-
5	LDYA dp	7D	2	5	Load YA YA ← (dp + 1) (dp)	N-----Z-
6	STYA dp	DD	2	5	Store YA (dp + 1) (dp) ← YA	--------
7	SUBW dp	3D	2	5	16-bits subtract without carry YA ← (YA) - (dp + 1) (dp)	NV--H-ZC

BIT manipulation

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
1	AND1 M.bit	8B	3	4	Bit AND C-flag : C ← (C) ∧ (M.bit)	-------C
2	AND1B M.bit	8B	3	4	Bit AND C-flag and NOT : C ← (C) ∧ ~(M.bit)	-------C
3	BIT dp	0C	2	4	Bit test A with memory : Z ← (A) ∧ (M), N ← (M7), V ← (M6)	MM----Z-
4	BIT !abs	1C	3	5		
5	CLR1 dp.bit	y1	2	4	Clear bit : (M.bit) ← “0”	--------
6	CLRA1 A.bit	2B	2	2	Clear A bit : (A.bit) ← “0”	--------
7	CLRC	20	1	2	Clear C-flag : C ← “0”	-------0
8	CLRG	40	1	2	Clear G-flag : G ← “0”	--0-----
9	CLRV	80	1	2	Clear V-flag : V ← “0”	-0--0---
10	EOR1 M.bit	AB	3	5	Bit exclusive-OR C-flag : C ← (C) ⊕ (M.bit)	-------C
11	EOR1B M.bit	AB	3	5	Bit exclusive-OR C-flag and NOT : C ← (C) ⊕ ~(M.bit)	-------C
12	LDC M.bit	CB	3	4	Load C-flag : C ← (M.bit)	-------C
13	LDCB M.bit	CB	3	4	Load C-flag with NOT : C ← ~(M.bit)	-------C
14	NOT1 M.bit	4B	3	5	Bit complement : (M.bit) ← ~(M.bit)	--------
15	OR1 M.bit	6B	3	5	Bit OR C-flag : C ← C ∨ (M.bit)	-------C
16	OR1B M.bit	6B	3	5	Bit OR C-flag and NOT : C ← C ∨ ~ (M.bit)	-------C
17	SET1 dp.bit	x1	2	4	Set bit : (M.bit) ← “1”	--------
18	SETA1 A.bit	0B	2	2	Set A bit : (A.bit) ← “1”	--------
19	SETC	A0	1	2	Set C-flag : C ← “1”	-------1
20	SETG	C0	1	2	Set G-flag : G ← “1”	--1-----
21	STC M.bit	EB	3	6	Store C-flag : (M.bit) ← C	--------
22	TCLR1 !abs	5C	3	6	Test and clear bits with A : A – (M), (M) ← (M) ∧ ~(A)	N-----Z-
23	TSET1 !abs	3C	3	6	Test and set bits with A : A – (M), (M) ← (M) ∨ (A)	N-----Z-

Branch / Jump

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
1	BBC A.bit, rel	y2	2	4/6	Branch if bit clear : If (bit) = 0, then pc ← (pc) + rel	--------
2	BBC dp.bit, rel	y3	3	5/7		
3	BBS A.bit, rel	x2	2	4/6	Branch if bit set : If (bit) = 1, then pc ← (pc) + rel	--------
4	BBS dp.bit, rel	x3	3	5/7		
5	BCC rel	50	2	2/4	Branch if carry bit clear : If (C) = 0, then pc ← (pc) + rel	--------
6	BCS rel	D0	2	2/4	Branch if carry bit set : If (C) = 1, then pc ← (pc) + rel	--------
7	BEQ rel	F0	2	2/4	Branch if equal : If (Z) = 1, then pc ← (pc) + rel	--------
8	BMI rel	90	2	2/4	Branch if minus : If (N) = 1, then pc ← (pc) + rel	--------
9	BNE rel	70	2	2/4	Branch if not equal : If (Z) = 0, then pc ← (pc) + rel	--------
10	BPL rel	10	2	2/4	Branch if plus : If (N) = 0, then pc ← (pc) + rel	--------
11	BRA rel	2F	2	4	Branch always : pc ← (pc) + rel	--------
12	BVC rel	30	2	2/4	Branch if overflow bit clear : If (V) = 0, then pc ← (pc) + rel	--------
13	BVS rel	B0	2	2/4	Branch if overflow bit set : If (V) = 1, then pc ← (pc) + rel	--------
14	CALL !abs	3B	3	8	Subroutine call M(sp) ← (pcH), sp ← sp – 1, M(sp) ← (pcL), sp ← sp – 1, If !abs, pc ← abs ; if [dp], pcL ← (dp), pcH ← (dp + 1)	--------
15	CALL [dp]	5F	2	8		
16	CBNE dp, rel	FD	3	5/7	Compare and branch if not equal : if (A) ≠ (M), then pc ← (pc) + rel	--------
17	CBNE dp+X, rel	8D	3	6/8		
18	DBNE dp, rel	AC	3	5/7	Decrement and branch if not equal : if (M) ≠ 0, then pc ← (pc) + rel	--------
19	DBNE Y, rel	7B	2	4/6		
20	JMP !abs	1B	3	3	Unconditional jump : pc ← jump address	--------
21	JMP [!abs]	1F	3	5		
22	JMP [dp]	3F	2	4		

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
23	PCALL upage	4F	2	6	U-page call M(sp) ← (pcH), sp ← sp – 1, M(sp) ← (pcL), sp ← sp – 1, pcL ← (upage), pcH ← "0FFH"	--------
24	TCALL n	nA	1	8	Table call M(sp) ← (pcH), sp ← sp – 1, M(sp) ← (pcL), sp ← sp – 1, pcL ← (Table vector L), pcH ← (Table vector H)	--------

Control Operation / Etc

NO.	MNEMONIC	OP CODE	BYTE NO	CYCLE NO	OPERATION	FLAG NVGBHIZC
1	BRK	0F	1	8	Software interrupt : B ← "1", M(sp) ← (pcH), sp ← sp – 1, M(sp) ← (pcL), sp ← sp – 1, M(sp) ← (PSW), sp ← sp – 1, pcL ← (0FFDEH), pcH ← (0FFDFH)	---1-0--
2	DI	60	1	3	Disable interrupt : I ← "0"	-----0--
3	EI	E0	1	3	Enable interrupt : I ← "1"	-----1--
4	NOP	FF	1	2	No operation	--------
5	POP A	0D	1	4	sp ← sp + 1, A ← M(sp) sp ← sp + 1, X ← M(sp) sp ← sp + 1, Y ← M(sp) sp ← sp + 1, PSW ← M(sp)	--------
6	POP X	2D	1	4		
7	POP Y	4D	1	4		
8	POP PSW	6D	1	4		restored
9	PUSH A	0E	1	4	M(sp) ← A, sp ← sp - 1 M(sp) ← X, sp ← sp - 1 M(sp) ← Y, sp ← sp - 1 M(sp) ← PSW, sp ← sp - 1	--------
10	PUSH X	2E	1	4		
11	PUSH Y	4E	1	4		
12	PUSH PSW	6E	1	4		
13	RET	6F	1	5	Return from subroutine sp ← sp + 1, pcL ← M(sp), sp ← sp + 1, pcH ← M(sp)	--------
14	RETI	7F	1	6	Return from interrupt sp ← sp + 1, PSW ← M(sp), sp ← sp + 1, pcL ← M(sp), sp ← sp + 1, pcH ← M(sp)	restored
15	STOP	EF	1	3	Stop mode (halt CPU, stop oscillator)	--------