ePG3231-EM202

RISC II-2G Series Microcontroller

Product Specification

DOC. VERSION 1.4

ELAN MICROELECTRONICS CORP.

March 2006

Trademark Acknowledgments:
IBM is a registered trademark and PS/2 is a trademark of IBM
Windows is a trademark of Microsoft Corporation
ELAN and ELAN logo are trademarks of ELAN Microelectronics Corporation

The contents of this specification are subject to change without further notice. ELAN Microelectronics assumes no responsibility concerning the accuracy, adequacy, or completeness of this specification. ELAN Microelectronics makes no commitment to update, or to keep current the information and material contained in this specification. Such information and material may change to conform to each confirmed order.

In no event shall ELAN Microelectronics be made responsible for any claims attributed to errors, omissions, or other inaccuracies in the information or material contained in this specification. ELAN Microelectronics shall not be liable for direct, indirect, special incidental, or consequential damages arising from the use of such information or material.

The software (if any) described in this specification is furnished under a license or nondisclosure agreement, and may be used or copied only in accordance with the terms of such agreement.

ELAN Microelectronics products are not intended for use in life support appliances, devices, or systems. Use of ELAN Microelectronics product in such applications is not supported and is prohibited.
NO PART OF THIS SPECIFICATION MAY BE REPRODUCED OR TRANSMITTED IN ANY FORM OR BY ANY MEANS WITHOUT THE EXPRESSED WRITTEN PERMISSION OF ELAN MICROELECTRONICS.

ELAN MICROELECTRONICS CORPORATION

Headquarters:
No. 12, Innovation Road 1
Hsinchu Science Park
Hsinchu, Taiwan 30077
Tel: +886 3 563-9977
Fax: +886 3 563-9966
http://www.emc.com.tw

Hong Kong:
Elan (HK) Microelectronics Corporation, Ltd.
Rm. 1005B, 10/F Empire Centre
68 Mody Road, Tsimshatsui
Kowloon , HONG KONG
Tel: +852 2723-3376
Fax: +852 2723-7780
elanhk@emc.com.hk

USA:
Elan Information Technology Group (U.S.A.)
1821 Saratoga Ave., Suite 250
Saratoga, CA 95070
USA
Tel: +1 408 366-8225
Fax: +1 408 366-8220

Europe:
Elan Microelectronics Corp. (Europe)
Siewerdtstrasse 105
8050 Zurich, SWITZERLAND
Tel: +41 43 299-4060
Fax: +41 43 299-4079
http://www.elan-europe.com

Shenzhen:
Elan Microelectronics Shenzhen, Ltd.
SSMEC Bldg., 3F, Gaoxin S. Ave.
Shenzhen Hi-Tech Industrial Park
Shenzhen, Guandong, CHINA
Tel: +86 755 2601-0565
Fax: +86 755 2601-0500

Shanghai:
Elan Microelectronics Shanghai, Ltd.
23/Bldg. #115 Lane 572, Bibo Road
Zhangjiang Hi-Tech Park
Shanghai, CHINA
Tel: +86 21 5080-3866
Fax: +86 21 5080-4600

Contents

Specification Revision History

Code	Doc. Version	Revision Description	Date
EM002	1.1	Initial version	2/3/2005
EM102	1.2	1. Renamed as ePG3231-EM102. 2. Modified the Initial mode as Fast mode. 3. Modified the Extended MCU mode Port C initial setting. 4. Modified the Internal program as EPG3231_em102.obj 5. Added CPU mode control library subroutine code. 6. Added a Note in the Sleep and Idle mode. 7. Modified the “Download flash data control program” operation condition.	5/10/2005
EM102	1.3	New Specification	7/26/2005
EM202	1.4	1. Renamed as ePG3231-EM202 2. Modified the PLLC capacitor range. 3. Modified the Timer interrupt start address. 4. Modified the A/D converter max. value samlpe rate. 5. Modified the max. value supply current to 10MHz. 6. Modified the A/D conversion operation mode. 7. Added Baud rate select list of Flash download program.	03/08/2006

1 General Description

The ePG3231-EM202 is the Extended MCU mode version of the ePG3231 IC. It is an 8-bit, RISC architecture MCU embedded with a 10-bit SAR A/D converter with touch screen controller, two 8-bit timers and one 16-bit general timer with built-in capture and event counter functions. It is also provided with an IR generator, watchdog timer, SPI, UART, four melody timers, a PWM and a current D/A. In addition, the large size user RAM and program memory capacity, which supports external memory In System Programming (ISP) function makes it ideal for high performance and low cost solution for leisure products and educational learning tools applications.

The ePG3231-EM202 allows users to actualize the concept of using the address and data bus together, along with some commonly used program library for enhanced performance MCU. Integrated functions are on-board programming flash data, external device control library, Flash control library, touch panel control library, melody/speech library, etc. Refer to ePG3231-EM202 User's Guide for details on the development process.

The MCU core is ELAN's second generation RISC based namely RISCII (RII). The core was designed for low power and portable devices. It supports Fast mode, Slow mode, Idle mode and Sleep mode for low power application.

IMPORTANT NOTES

- *Do not use Register BSR (05h) Bit 7 ~ Bit 5.*
- *Do not use Register BSR1 (07h) Bit 7 ~ Bit 5.*
- *Do not use Special Register (2Ah).*
- *Do not use Special Register (4Fh).*
- *Do not use Registers JDNZ & JINZ at FSR1 (09h) special register.*
- *Ports J & K are ideal for keyboard matrix application due to its low output current (see DC Electrical Characteristics in Chapter 9).*
- *Before going into Sleep mode or Idle mode, Ports D, E, & F must be set as Output port and output high. At the same time, Ports G & H must be set as Input port and enabled pull-high.*

1.1 Elan Support Function

- 4-channel Melody or 3-channel Melody + 1-channel Speech
- ePG3231-EM202 User's Guide

NOTE

- *Download the ePG3231-EM202 User's Guide from Elan's Website, www.emc.com.tw*
- *Products → Microcontroller line → RISCII-2G Series → Application Note*

1.2 Applications

- Educational Learning Tools
- Kids PDA and Computer
- Electronics Book
- Dictionary, Data Bank

2 Features

2.1 MCU Features

- 8 bit RISC MCU
- 8×8 multiplier with controllable signed or unsigned operation
- Operating voltage and speed: 16MHz @ 3.0~3.6V, 12MHz @ 2.7~3.0V, 10MHz @ 2.2~2.7V

  External interface access speed formula (Text: Flash or Mask ROM access time)
  Extended MCU mode: FPLL = 2 / [Text + 100ns (> 3V), 120ns (> 2.7V) or 140ns (> 2.4V)]
- One Instruction cycle time = 2 × System clock time
- Supports external PROM/DROM In System Programming (ISP) function
- 128 bytes un-banked RAM including special registers and common registers
- 32×128 bytes banked RAM
- RAM stack can achieve a maximum of 128 stack levels
- Table Look-up function is fast and highly efficient when accompanying a REPEAT instruction
- Register to Register move instruction
- Compare and Branch in one instruction (2 cycles)
- Single Repeat function (max. of 256 repetitions)
- Decimal ADD & SUB instruction
- Full range CALL and JUMP ability (2 cycles)

2.2 Peripheral

- One input port (Port A) and five general I/O ports (Port B ~ Port C, Port I ~ Port K)
- 16-bit timer (Timer 0) with capture and event counter functions
- 8-bit timer (Timer 1) with wake-up function
- 8-bit timer (Timer 2) as beat counter for Melody function

- 8-bit IR generator
- Current D/A for melody and speech application
- 8-bit Watchdog Timer
- 10-bit resolution SAR A/D converter with 6 channels general analog input and 2 channels for touch panel application
- 128 Keys strobe function
- 8/16/24 bits SPI (Serial Peripheral Interface)
- UART (Universal Asynchronous Receiver and Transmitter)
- Melody interrupt & Interrupt priority

  (4-channel Melody/Speech Synthesizer) Disable melody interrupt function

  Interrupt priority (external > capture > speech > Timers 0 ~ 2 > peripheral)

2.3 Internal Specification

- Watchdog Timer with its own on-chip RC oscillator
- MCU modes: Sleep Mode, Idle Mode, Slow Mode, and Fast Mode
- Only Crystal oscillation for system clock
- PLL is turned on during Fast mode, and controlled by PEN bit when the MCU is in Slow mode and Idle mode.
- MCU Wake-up function includes Input wake-up, Timer 1 wake up, Touch Panel wake-up, SPI wake-up, and A/D wake-up.
- MCU interrupt function includes Input Port interrupt, Touch Panel interrupt, Capture interrupt, Speech Timer interrupt, Timer interrupt (Timers 0~2), A/D interrupt, SPI interrupt and UART interrupt.
- MCU reset function includes Power-on reset, RSTB Pin reset, and Watchdog Timer reset

2.4 Internal Supporting Library

- CPU Mode control library
- ELAN, AMD and MXIC flash IC control library
- External device control library
- Touch Panel control library
- Melody/Speech control library
- On board programming flash data

3 ePG3231-EM202 Packaging

3.1 128-Pin QFP Package

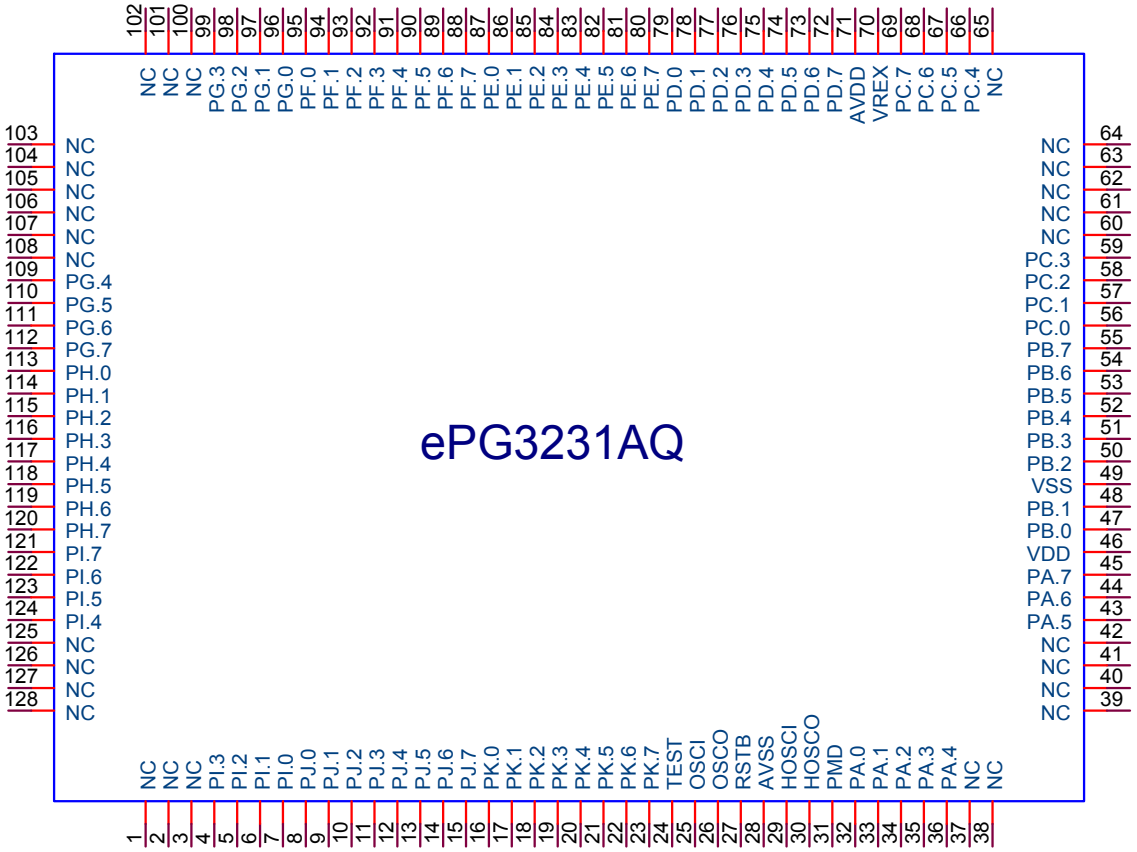


Fig. 3-1 ePG3231 Package: QFP 128 Pins

4 Block Diagram

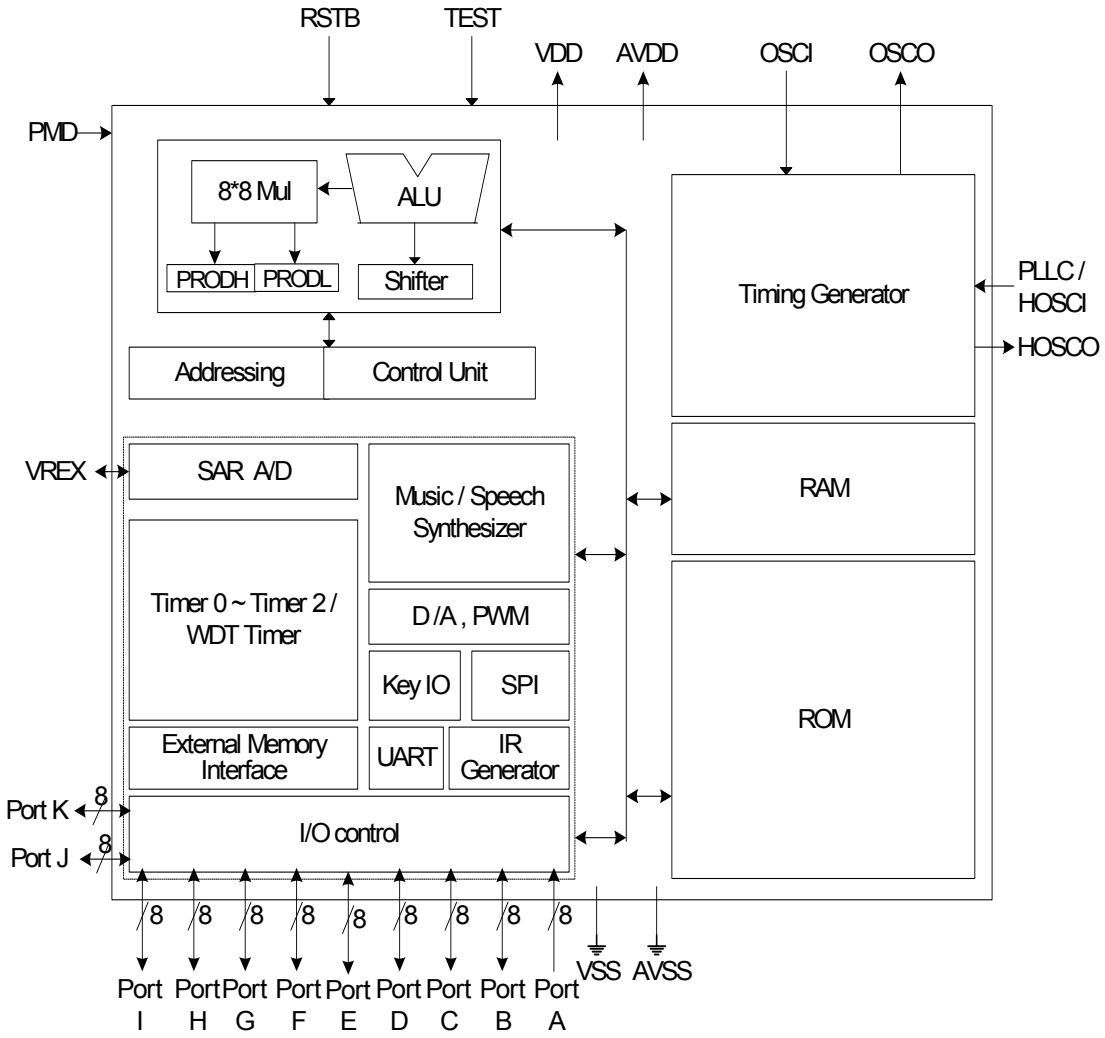
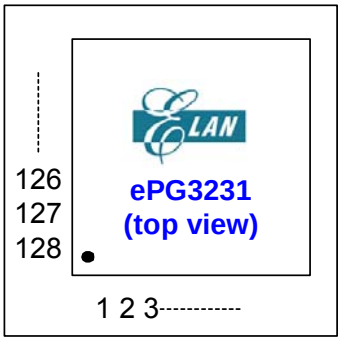


Fig. 4-1 ePG3231 Block Diagram

5 Pin Assignment



No.	Pin Name	No.	Pin Name	No.	Pin Name	No.	Pin Name
1	N.C.	33	PA.1	65	N.C.	97	PG.1 (D1)
2	N.C.	34	PA.2	66	PC.4 (ADIN4/YP)	98	PG.2 (D2)
3	N.C.	35	PA.3	67	PC.5 (ADIN3/XP)	99	PG.3 (D3)
4	PI.3	36	PA.4	68	PC.6 (YN)	100	N.C.
5	PI.2	37	N.C.	69	PC.7 (XN)	101	N.C.
6	PI.1	38	N.C.	70	VREX	102	N.C.
7	PI.0	39	N.C.	71	AVDD	103	N.C.
8	PJ.0 (Strobe 0)	40	N.C.	72	PD.7 (OEB)	104	N.C.
9	PJ.1 (Strobe 1)	41	N.C.	73	PD.6 (WEB)	105	N.C.
10	PJ.2 (Strobe 2)	42	N.C.	74	PD.5 (A21)	106	N.C.
11	PJ.3 (Strobe 3)	43	PA.5	75	PD.4 (A20)	107	N.C.
12	PJ.4 (Strobe 4)	44	PA.6 (Rref)	76	PD.3 (A19)	108	N.C.
13	PJ.5 (Strobe 5)	45	PA.7	77	PD.2 (A18)	109	PG.4 (D4)
14	PJ.6 (Strobe 6)	46	VDD	78	PD.1 (A17)	110	PG.5 (D5)
15	PJ.7 (Strobe 7)	47	PB.0 (VO2)	79	PD.0 (A16)	111	PG.6 (D6)
16	PK.0 (Strobe 8)	48	PB.1 (VO1/DAO)	80	PE.7 (A15)	112	PG.7 (D7)
17	PK.1 (Strobe 9)	49	VSS	81	PE.6 (A14)	113	PH.0 (D8)
18	PK.2 (Strobe 10)	50	PB.2 (IROT)	82	PE.5 (A13)	114	PH.1 (D9)
19	PK.3 (Strobe 11)	51	PB.3	83	PE.4 (A12)	115	PH.2 (D10)
20	PK.4 (Strobe 12)	52	PB.4	84	PE.3 (A11)	116	PH.3 (D11)
21	PK.5 (Strobe 13)	53	PB.5 (EVIN/CPIN)	85	PE.2 (A10)	117	PH.4 (D12)
22	PK.6 (Strobe 14)	54	PB.6 (UTXD)	86	PE.1 (A9)	118	PH.5 (D13)
23	PK.7 (Strobe 15)	55	PB.7 (URXD)	87	PE.0 (A8)	119	PH.6 (D14)
24	TEST	56	PC.0 (ADIN8)	88	PF.7 (A7)	120	PH.7 (D15)
25	OSCI	57	PC.1 (ADIN7)	89	PF.6 (A6)	121	PI.7 (SPISDI)
26	OSCO	58	PC.2 (ADIN6)	90	PF.5 (A5)	122	PI.6 (SPISDO)
27	RSTB	59	PC.3 (ADIN5)	91	PF.4 (A4)	123	PI.5 (SPISCK)
28	AVSS	60	N.C.	92	PF.3 (A3)	124	PI.4 (/SPISS)
29	HOSCI /PLLC	61	N.C.	93	PF.2 (A2)	125	N.C.
30	HOSCO	62	N.C.	94	PF.1 (A1)	126	N.C.
31	PMD	63	N.C.	95	PF.0 (A0)	127	N.C.
32	PA.0	64	N.C.	96	PG.0 (D0)	128	N.C.

6 Pin Description

Name	I/O/P Type	Description
VDD VSS	P	Digital power supply, ranging from 2.2V to 3.6V. Connect to VSS through the capacitors (0.1µF).
AVDD AVSS	P	Analog power supply, ranging from 2.2V to 3.6V. Connect to AVSS through the capacitors (0.1µF).
RSTB	I	System reset input with built-in pull-high resistor (100kΩ typical) Low: RESET asserted High: RESET released
TEST	I	Normally connected to VSS, reserved for testing.
OSCI/RC OSCO	I O	RC or Crystal selection by Code Option 32768Hz oscillator pin, should be connected to VSS through the capacitors (20pF). RC oscillator connector pin, should be connected to VDD through a resistor (2MΩ).
HOSCI/PLLC HOSCO	I O	Crystal or PLL selection by Code Option High frequency crystal oscillator pins, should be connected to VSS through the capacitors (20pF). PLL capacitor connector pin, should be connected to VSS through the capacitors (0.01 ~ 0.047µF).
PMD	I	Fix MCU/Extended MCU mode. Connect to VSS through a resistor (100kΩ).
VREX	I/O	External or internal reference voltage for A/D converter. Connect to VSS through the capacitors (0.1µF).
Port A	I	General input pin Special functions: Wakeup, Interrupt & Key matrix input pin **Bit 7: Control on board programming, connect to VSS** **User Program is Open**
Port B	I or I/O O or I/O I or I/O O O O or I/O O I/O	Bit 7: UART Rx pin or General I/O pin Bit 6: UART Tx pin or General I/O pin Bit 5: Event counter / Capture input or General I/O pin **Bit 4: External device / CS1 control or General Output pin** **Bit 3: External PROM & DROM Flash / CS control** Bit 2: IR generated output or General I/O pin **Bit 1: Current D/A output** Bit 0: General I/O pin
Port C	O O I I I/O I I	**Bit 7: Touch screen X direction negative pin** **Bit 6: Touch screen Y direction negative pin** **Bit 5: Touch screen X direction positive pin & A/D input Channel 3** **Bit 4: Touch screen Y direction positive pin & A/D input Channel 4** Bit 3 ~ 2: General I/O pin Bit 1: External memory interface ready/busy detect input (RY/BY) pin Bit 0: General I/O pin
Port D	O O O O O	**Bit 7: External memory interface output enable (/OE) pin** **Bit 6: External memory interface write enable (/WE) pin** **Bit 5: External device /CS7 control or General Output pin** **Bit 4: External device /CS6 control or General Output pin** **Bit 3~0: Extended MCU mode/Processor mode or External memory interface address [A19:A16]**

Name	I/O/P Type	Description
Port E	O	**Bit 7~0: Extended MCU mode/Processor mode or External memory interface address [A15:A8]**
Port F	O	**Bit 7~0: Extended MCU mode/Processor mode or External memory interface address [A7:A0]**
Port G	I I/O	**Bit 7~0: Extended MCU mode/Processor mode data bus [D7:D0]** **Bit 7~0: External memory interface data [D7:D0]**
Port H	I I/O	**Bit 7~0: Extended MCU mode/Processor mode data bus [D15:D8]** **Bit 7~0: External memory interface data [D15:D8]**
Port I	I or I/O O or I/O I/O I or I/O O O O O	Bit 7: Serial data input pin or General I/O pin Bit 6: Serial data output pin or General I/O pin Bit 5: Serial clock input/output pin or General I/O pin Bit 4: /Slave Select pin or General I/O pin **Bit 3: External device /CS5 control or High drive output pin** **Bit 2: External device /CS4 control or High drive output pin** **Bit 1: External device /CS3 control or High drive output pin** **Bit 0: External device /CS2 control or High drive output pin**
Port J	I/O O	General Output/Input port Bit 7~0: Key strobe 7~0 pins
Port K	I/O O	General Output/Input port Bit 7~0: Key strobe 15~8 pins

NOTE

The Boldfaced descriptions indicate that the I/O has been fixed (By code option setting), cannot be used as a general I/O.

6 Code Option

- Oscillator (OSCSEL): Select “Crystal” oscillator
- Initial mode after reset: Select “Slow” mode
- High freq. system clock (HFSEL): Select “PLLC” mode
- External PROM instruction speed: Select “Fsystem/2”: Instruction cycles per system 2 clock
- Current D/A reference source: Select “D/A internal reference voltage”
- Melody interrupts & Interrupt priority: Select “Mode1”
- UART standard baud rate: Select “PLL frequency is 9.83MHz”
- Port C.7 function select bit: Select “XN for touch panel”
- Port C.6 function select bit: Select “YN for touch panel
- Port C.5 function select bit: Select “XP for touch panel/ADIN3”
- Port C.4 function select bit: Select “YP for touch panel/ADIN4”
- Port C.3 function select bit: Select “General I/O function”
- Port C.2 function select bit: Select “General I/O function”

- Port C.1 function select bit: Select “General I/O function”
- Port C.0 function select bit: Select “ADIN8”
- External Memory Size:

External Memory Size	Ports as Address Bus	Ports as General Output
1MW	Port D[3:0]:Port E: Port F	Port D[5:4]

- DAC and PWM Function Select Bits:

DAC or PWM Function Selection	Port B.0 and Port B.1 Function
DAC is used	Port B.1 is DAO for D/A, Port B.0 is general I/O

7 Function Description

7.1 Reset Function

A Reset can be triggered by:

- Power-on voltage detector reset and power-on reset
- WDT timeout
- RSTB pin pull low

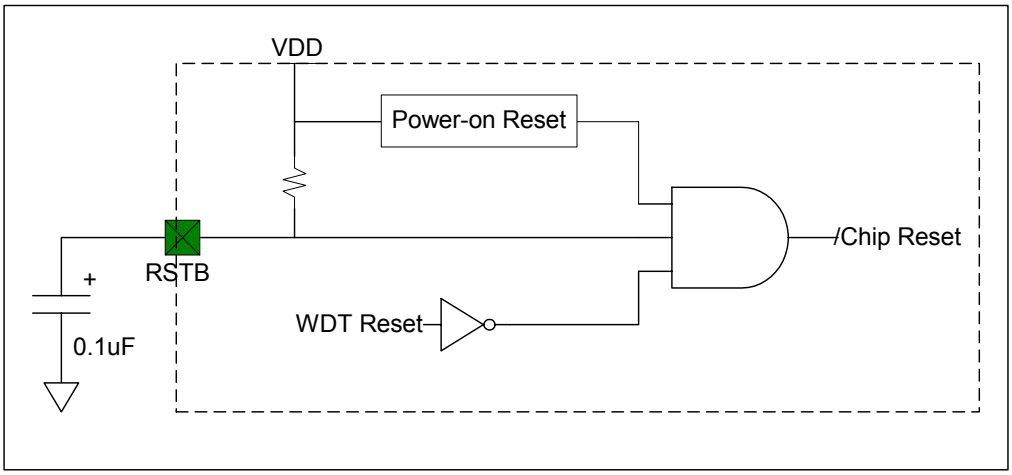


Fig. 7-1 On-chip RESET Circuit

7.1.1 Power–up and Reset Timing

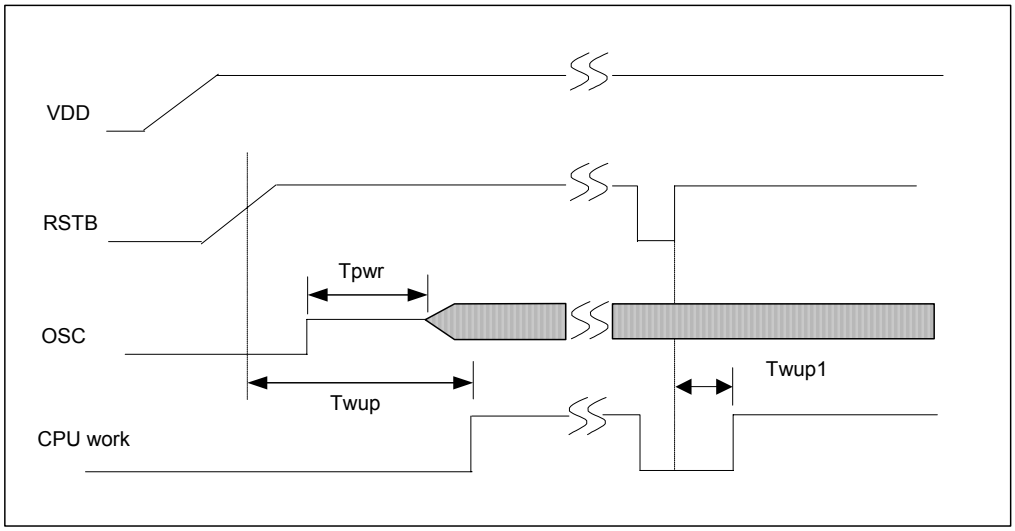


Fig. 7-2 Power-up and Reset Timing

Symbol	Characteristics	Min.	Typ.	Max.	Unit
Tpwr	Oscillator start-up time	100	226	300	ms
Twup	CPU warm-up time	260	340	550	ms
Twup1	CPU reset time	18	22	44	ms

- Status (R0Fh)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
/TO	/PD	SGE	SLE	OV	Z	DC	C

Bit 0 (C): Carry flag or inverse of Borrow flag (B)

When in SUB operation, borrow flag is indicated by the inverse of the carry bit. (B = /C)

Bit 1 (DC): Auxiliary carry flag

Bit 2 (Z): Zero flag

Bit 3 (OV): Overflow flag. Use in signed operation when bit 6 carry into or borrow from signed bit (Bit 7).

Bit 4 (SLE): Computation result is less than or equal to zero (Negative value) after a signed arithmetic. This is only affected by HEX arithmetic instruction.

Bit 5 (SGE): Computation result is greater than or equal to zero (Positive value) after the assigned arithmetic. This is only affected by HEX arithmetic instruction.

NOTE

1. When OV=1 after a signed arithmetic, you can check the SGE and SLE bits to know whether an overflow (carry into sign bit) or underflow (borrow from sign bit) occurs.

OV=1 and SGE=1 → overflow occurs

OV=1 and SLE=1 → underflow occurs

2. When overflow occurs, you should clear the MSB of the Accumulator in order to get the correct value.

When underflow occurs, you should set the MSB of the Accumulator in order to get the correct value.

Example 1: ADD positive value with a positive value, and the ACC signed bit will be affected.

```
MOV     ACC, #60h       ; Signed number +60h
ADD     ACC, #70h       ; +60h ADD WITH +70h
```

Unsigned bit results after execution of the instruction**:**

ACC = 0D0h SGE=1, means the result is greater than or equal to 0 (positive value)

OV=1, means overflow occurred and the result is carried into signed bit (Bit 7)

Signed bit results after execution of the instruction**:**

ACC = 50h (signed bit is cleared)

The actual result = +80h (OV=1) + 50h = +0D0h

Example 2: SUB positive value from negative value, and ACC signed bit will be affected.

```
MOV     ACC, #50h       ; Signed number +50h.
SUB     ACC, #90h       ; +50h SUB from –70h. (Signed number of 90h)
```

Unsigned bit results after execution of the instruction**:**

ACC = 40h SLE=1, means the result is less than or equal to 0 (negative value)

OV=1, Underflow occurred and the result borrowed from signed bit (Bit 7)

Signed bit results after execution of the instruction**:**

ACC = 0C0h (the signed bit is set)

The actual result = –80h (OV=1) + 0C0h (signed number of 0C0h) = 40h.

Bit 6 (/PD): Reset to 0 when entering Sleep mode. Set to 1 by "WDTC" instruction, power-on reset or Reset pin low condition.

Bit 7 (/TO): Reset to 0 when WDT time-out reset. Set to 1 by "WDTC" instruction; enter Sleep Mode, power-on reset or Reset pin low condition.

When a reset occurs, the special function register is reset to its initial value except for the /**TO** & /**PD** bits of the Status register.

Bit 7 (/TO)	Bit 6 (/PD)	Event
0	0	WDT time-out reset from Sleep mode
0	1	WDT time-out reset (not in Sleep mode)
1	0	Reserved
1	1	Power on or RSTB pin low condition

7.1.2 Register Initial Values

Special Registers:

Addr.	Name	Initial Value	Addr.	Name	Initial Value
00h	INDF0	---- ----[1]	10h	TRL2	uuuu uuuu
01h	FSR0	0000 0000	11h	PRODL	uuuu uuuu
02h	PCL	0000 0000	12h	PRODH	uuuu uuuu
03h	PCM	0000 0000	13h	ADOTL	000- -0uu
04h	PCH	---- --00	14h	ADOTH	uuuu uuuu
05h	BSR	---0 0000	15h	UARTTX	xxxx xxxx
06h	STKPTR	0000 0000	16h	UARTRX	xxxx xxxx
07h	BSR1	---0 0000	17h	Port A	xxxx xxxx
08h	INDF1	---- ----[1]	18h	Port B	xxxx xxxx
09h	FSR1	1000 0000	19h	Port C	xxxx xxxx
0Ah	ACC	xxxx xxxx	1Ah	Port D	xxxx xxxx
0Bh	TABPTRL	0000 0000	1Bh	Port E	xxxx xxxx
0Ch	TABPTRM	0000 0000	1Ch	Port F	xxxx xxxx
0Dh	TABPTRH	0000 0000	1Dh	Port G	xxxx xxxx
0Eh	CPUCON	0--0 000c[2]	1Eh	Port H	xxxx xxxx
0Fh	STATUS	cuxx xxxx[3]	1Fh	Port I	xxxx xxxx

Control Registers:

Addr.	Name	Initial Value	Addr.	Name	Initial Value
20h	PFS	0010 0000	38h	DCRHI	0011 0011
21h	STBCON	0100 0000	39h	DCRJK	0011 0011
22h	INTCON	0000 0000	3Ah	PBCON	0000 0000
23h	INTSTA	0000 0000	3Bh	PCCON	0000 0000
24h	TRL0L	uuuu uuuu	3Ch	PLLF	xxxx xxxx
25h	TRL0H	uuuu uuuu	3Dh	T0CL	0000 0000
26h	TRL1	uuuu uuuu	3Eh	T0CH	0000 0000
27h	TR01CON	0000 0000	3Fh	SPICON	0000 0000
28h	TR2CON	0000 0000	40h	SPISTA	0-00 0000
29h	TRLIR	uuuu uuuu	41h	SPRL	xxxx xxxx
2Ah	(Reserved)	---- ----	42h	SPRM	xxxx xxxx
2Bh	POST_ID	1-11 0-00	43h	SPRH	xxxx xxxx
2Ch	ADCON	0101 0000	44h	SFCR	0000 0000
2Dh	PAINTEN	0000 0000	45h	ADDL1~ADDL4	xxxx xxxx
2Eh	PAINTSTA	0000 0000	46h	ADDM1~ADDM4	xxxx xxxx
2Fh	PAWAKE	0000 0000	47h	ADDH1~ADDH4	xxxx xxxx
30h	UARTCON	0000 0010	48h	ENV1~4 / SPHDR	0000 0000 / 0000 0000
31h	UARTSTA	0000 0000	49h	MTCON1~4 / SPHTCON	0000 0000 / 0000 0000
32h	PORTJ	xxxx xxxx	4Ah	MTRL1~4 / SPHTRL	0000 0000 / 0000 0000
33h	PORTK	xxxx xxxx	4Bh	VOCON	0000 0111
34h	DCRB	1111 1111	4Ch	TR1C	1111 1111
35h	DCRC	1111 1111	4Dh	TR2C	1111 1111
36h	DCRDE	0011 0011	4Eh	ADCF	uuuu uuuu
37h	DCRFG	0011 0011	4Fh	(Reserved)	---- ----

Legend: **x** = unknown **–** = unimplemented read as "0"
u = unchanged **c** = value depends on actual condition

[1] Not a physically implemented register

[2] Bit 0 (MS0) of RE (CPUCON) is reloaded from "**INIM**" bit of code option when the MCU is reset.

[3] If it is a power-on reset or RSTB pin is at low condition, the /**TO** bit and /**PD** bit of RF (Status) are set to 1.
If it is a WDT time-out reset, the /**TO** bit is cleared and /**PD** bit is unchanged.

7.2 Oscillator System Function

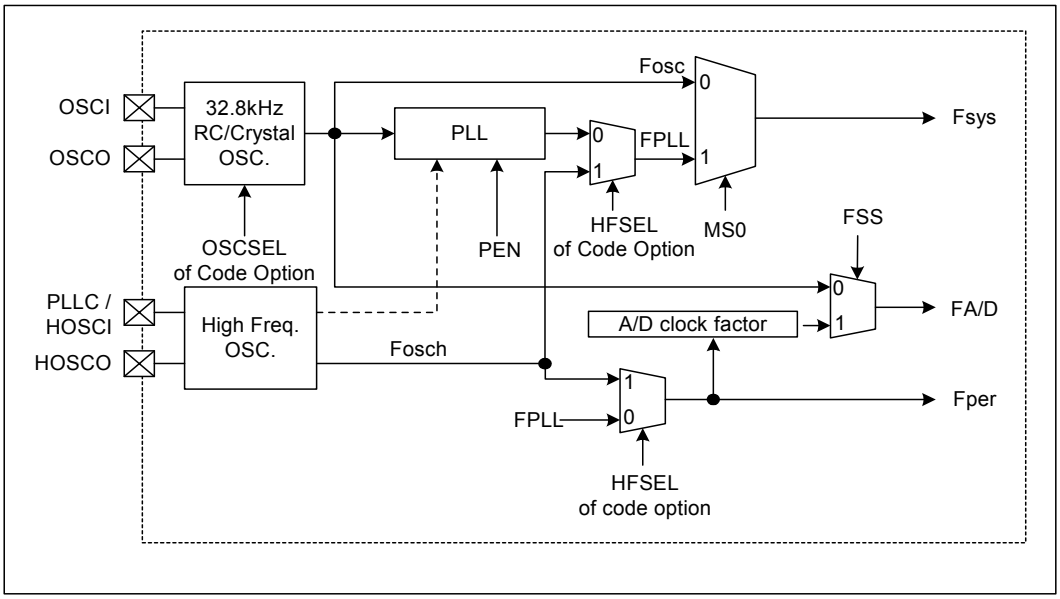


Fig. 7-3 Oscillator System Function Block Diagram

7.2.1 32768Hz Crystal Oscillator

For the 32768Hz Crystal oscillator, connect the crystal between OSCI pin and OSCO pin. Then connect the OSCI and OSCO pins to ground through a 20pF capacitor.

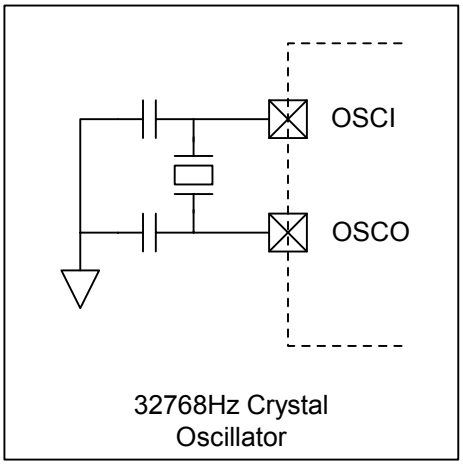


Fig. 7-4 Main Crystal Circuit Diagram

7.2.2 High Frequency Oscillator

High frequency system clock is from the PLL clock. (PLLC/HOSCI pin is used as PLLC pin)

7.2.2.1 Phase Locked Loop (PLL) Oscillator

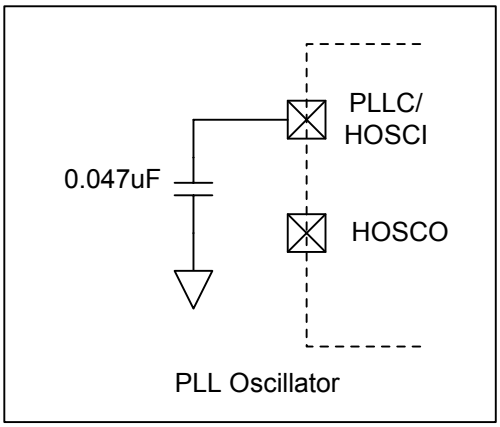


Fig. 7-5 Hi-Frequency Crystal Oscillator Circuit Diagram

- PLLF (R3Ch): Stores the actual PLL frequency value. It is used to verify whether the PLL frequency is stable or not.

$$Factual = 2 \times PLLF \times Fosc$$

- PFS (R20h): Target PLL frequency select register. System clock can be fine tuned from 262kHz to 16MHz. The initial value of PFS register after a chip reset is "20h" (FPLL=2.097 MHz).

$$Ftarget = 2 \times PFS \times Fosc$$

PFS Register	Ftarget (MHz)	PFS Register	Ftarget (MHz)
0~14	N.A. [1]	92	6.029
		107	7.012
15	0.983	122	7.995
31	2.032	137	8.978
46	3.015	150	9.83 [2]
61	3.998	153	10.027
76	4.981	255	16.712

[1] PFS = 0 ~ 14 is not available

[2] When UART is enabled, system clock should be 9.83MHz (PFS=150) or 14.745MHz (PFS=225)

The table is based on 32.768kHz oscillator frequency. The Maximum range of PLL is from 983kHz ~ 16.712MHz.

7.3 MCU Operation Mode

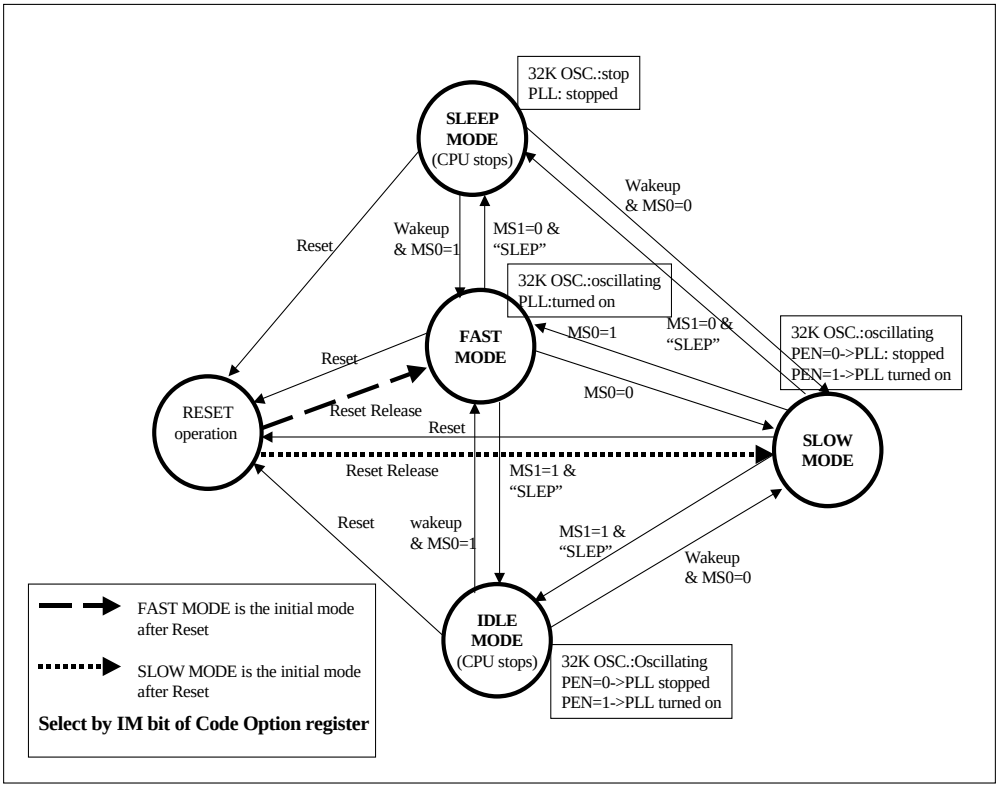


Fig. 7-6 MCU Operation Block Diagram

The following table shows the available functions for each type of MCU Mode:

Device \ Mode	Sleep	Idle	Slow	Fast
OSC (32768Hz)	×	O	O	O
Fsys	×	×	From OSC	From PLL
PLL	×	O	O	O
A/D Conversion	×	O [1]	O	O
Timers 0~2, IR Generator	×	O	O	O
INT	× [2]	× [2]	O	O
SPI	O (Slave)	O (Slave)	O	O
UART	×	×	×	O
Melody Synthesizer	×	×	×	O
PWM, Current D/A	×	×	O	O

Legend: **O** = Function is available if enabled × = Function is Not available

[1] It is recommended to operate the A/D converter in Idle mode to minimize the noise couple from the MCU clock.

[2] Interrupt flag will be recorded but not executed until the MCU wakes up.

■ Timing Diagram

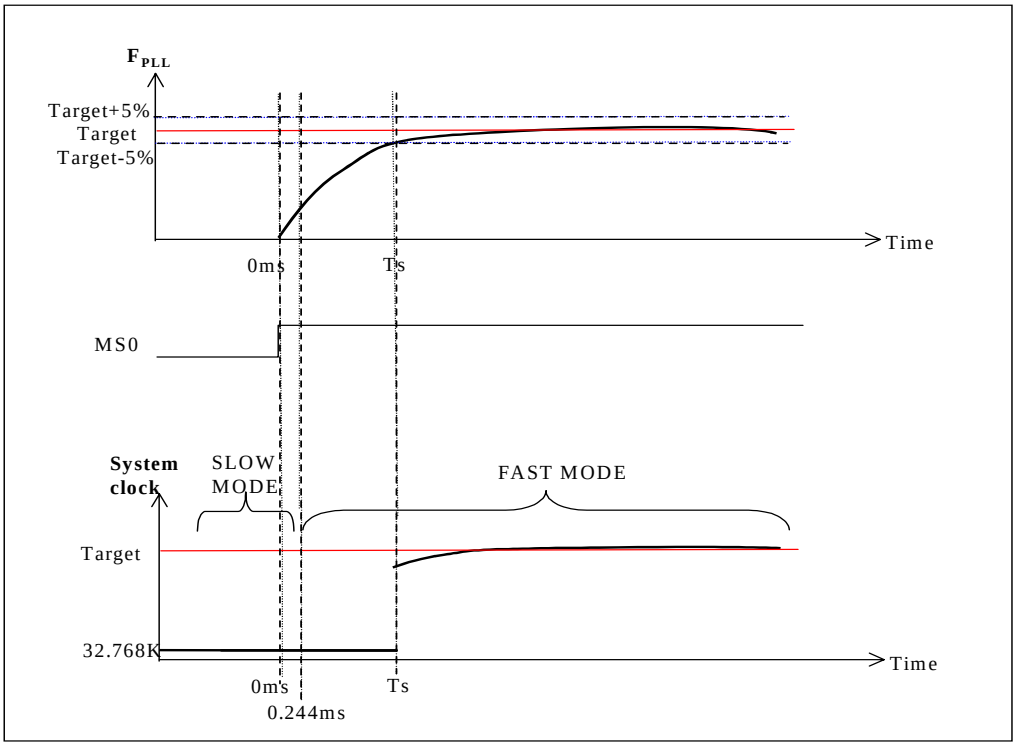


Fig. 7-7 MCU Operation Timing Diagram

NOTE

1. *Switch from Slow mode to Fast mode at Time=0ms*
2. *System clock will switch to FPLL after 8 oscillator clocks, and system clock will then be hundreds of kHz.*
3. *PLL frequency will be stable (±5%) at Time=Ts (2ms~5ms).*

- CPUCON (R0Eh)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	–	–	SMCAND	SMIER	GLINT	MS1	MS0

Sleep Mode: When the **MS1** bit (Bit 1) is set to '0' and "SLEP" instruction is executed, the MCU will enter into Sleep mode.

Idle Mode: When the **MS1** bit (Bit 1) is set to '1' and "SLEP" instruction is executed, the MCU will enter into Idle mode.

Slow Mode: When the **MS0** bit (Bit 0) is set to '0', the MCU will enter into Slow mode.

Fast Mode: When the **MS0** bit (Bit 0) is set to '1', the MCU will into enter Fast mode.

PEN: High Frequency enable bit. This bit is only effective when the MCU is in Idle or Slow mode.

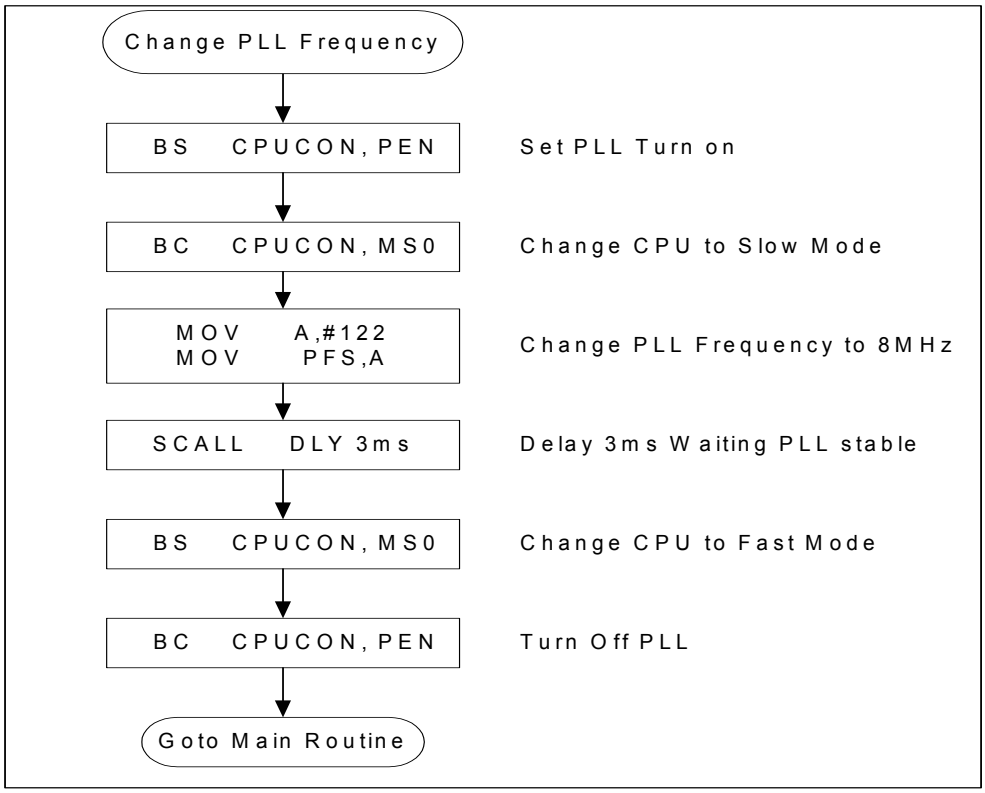
MCU Mode	PEN Bit	PLL On/Off
Sleep	×	Off
Idle/Slow	0	Off
	1	On
Fast	×	On

7.4 Wake-up Function

Mode / Device	Sleep	Idle	Slow	Fast
I/O wake up	O	O	×	×
Touch panel wake up	O	O	×	×
Timer 1 wake up	×	O	×	×
A/D wake up	×	O	×	×
SPI wake up	O (Slave)	O (Slave)	×	×

Legend: **O** = Function available if enabled **×** = Function is Not available

Flowchart



Code Examples

```
Idle & Slow modes change to FAST mode
       BC          CPUCON,MS0
       BS          CPUCON,PE
       MOV         A,#122
                   ;8MHz
       MOV         PFS,A
       Delay 3msec
       BS          CPUCON,MS0
       BC          CPUCON,PE

Fast mode change to Slow mode
       BC          CPUCON,MS0
```

```
Fast & Slow modes change to Idle mode
       CLR         DCRDE
       MOV         A, # 11110000B
       MOV         DCRFG, A
       MOV         A, # 00001111B
       OR          DCRHI, A
       MOV         A, # 11111111B
       MOV         PORTD, A
       MOV         PORTE, A
       MOV         PORTF, A

       BC          CPUCON, MS0
       BS          CPUCON, MS1
       SLEP
       NOP

Fast & Slow modes change to Sleep mode
       CLR         DCRDE
       MOV         A, # 11110000B
       MOV         DCRFG, A
       MOV         A, # 00001111B
       OR          DCRHI, A
       MOV         A, # 11111111B
       MOV         PORTD, A
       MOV         PORTE, A
       MOV         PORTF, A

       BC          CPUCON, MS0
       BC          CPUCON, MS1
       SLEP
       NOP
```

NOTE

Before going into Sleep mode or Idle mode, Ports D, E, & F must be set as Output port and output high. At the same time, Ports G & H must be set as Input port and enable pull-high. Otherwise, higher power consumption problem will result during Sleep and Idle modes.

7.5 Interrupt

When interrupt occurs, the GLINT bit of the CPUCON register is reset to 0, it then disables all interrupts, including Level 1 ~ Level 5. Setting this bit to 1 can enable all un-masked interrupts.

Interrupt Level	Interrupt Source	Start Address	Remark
	RESET	0x10000	
Level 1	Input Port A	0x10002	PAINT, PIRQB
Level 2	Capture	0x10004	CPIF
Level 3	Speech Timer	0x10006	SPHTI
Level 4	Timer0~2	0x10008	TMR0I, TMR1I, TMR2I
Level 5	Peripheral	0x1000A	UERRI, UTXI, URXI, ADIF, SRBFI

Code Example:

```
; ***** Reset program
ResetSEG  CSEG      0X10000
   LJMP  MSTART   ;(0X10000) Initialize
   LJMP  INPTINT  ;(0X10002) Input Port & Touch Panel INT
   LJMP  CAPINT   ;(0X10004) Capture Input INT
   LJMP  SPHINT   ;(0X10006) Speech/Melody Timer INT
   LJMP  TIMERINT ;(0X10008) Timer-0,1,2 INT
   LJMP  PERIPH   ;(0X1000A) Peripheral INT

PgmSEG    CSEG      0X10060
```

```
;---Push interrupt register
PUSH:
   MOV     AccBuf,A
   MOVPR   StatusBuf,Status
   RET

;---POP interrupt register
POP:
   MOVRP   Status,StatusBuf
   MOV     A,AccBuf
   RETI
```

7.5.1 Input Port A and Touch Panel Interrupts

Port A Interrupt (Falling edge trigger): Port A is used as external interrupt/wake up input.

Touch Panel Interrupt (Level trigger): When Port C.7~Port C.4 (X+, X-, Y+ & Y-) connected to touch panel input pins and touch panel is tapped, the PIRQB interrupt occurs.

Code Example:

```
;===Input Port and Touch Panel Interrupt
INPTINT:
      S0CALL     PUSH
      JBC        ADCON,PIRQB,toTPINT
      TEST       PAINTSTA
      JBC        STATUS,F_Z,toPAINT
      SJMP       POP
```

```
; --- Touch Panel Interrupt
toTPINT:
         :
      SJMP       POP
; --- Port A Interrupt
toPAINT:
      CLR        PAINTSTA
        :
      SJMP       POP
```

7.5.2 Capture Input Interrupt

The Capture Input Interrupt is used to capture an input event from rising to falling edge, falling to rising edge, rising to rising edge, or falling to falling edge. When every event input edge is detected, a Capture Interrupt takes place.

Code Example:

```
; === Capture Input Interrupt
CAPINT:
      S0CALL     PUSH
      JBS        INTSTA,CPIF,toCAPINT
      SJMP       POP
```

```
; --- Capture Input Interrupt
toCAPINT:
      BC         INTSTA,CPIF
       :
      SJMP       POP
```

7.5.3 Timer 0, Timer 1, and Timer 2 Interrupts

Timer 0 Interrupt**:** Timer 0 is a 16-bit timer for general time counting. When the counting value is larger than TRL0H **:** TRL0L value, Timer 0 Interrupt takes place.

Timer 1 Interrupt**:** Timer1 is an 8-bit timer for time counting and wake-up function. When Timer 1 counting value underflows, an interrupt occurs and the TRL1 value reloads to counting value.

Timer 2 Interrupt**:** Timer 2 is an 8-bit timer for time counting. When Timer 2 counting value underflows, an interrupt occurs and the TRL2 value reloads to counting value.

Code Example:

```
; === Timer-0, 1, 2 Interrupt
     JBS      INTSTA,TMR0I,toTM0INT
     JBS      INTSTA,TMR1I,toTM1INT
     JBS      INTSTA,TMR2I,toTM2INT
     SJMP     POP
```

```
; --- Timer 0 Interrupt
toTM0INT:
       BC       INTSTA,TMR0I
        :
       SJMP     POP
; --- Timer 1 Interrupt
toTM1INT:
       BC       INTSTA,TMR1I
        :
       SJMP     POP
; --- Timer 2 Interrupt
toTM2INT:
       BC       INTSTA,TMR2I
        :
       SJMP     POP
```

7.5.4 Speech Timer Interrupt (Disable Melody Interrupt Function)

Speech Timer is an 11-bit timer for time counting. When the Speech Timer counting value underflows, an interrupt occurs and the SPHTRL value reloads to counting value.

Code Example:

```
;===Speech Timer Interrupt
SPHINT:
      S0CALL     PUSH
      JBS        SPHTCON,SPHTI,toSPHINT
      SJMP       POP
```

```
;---To Speech Timer Interrupt
toSPHINT:
     BC         SPHTCON,SPHTI
      :
     SJMP       POP
```

7.5.5 Peripheral Interrupt

1. A/D (Analog to Digital converter) Interrupt: The A/D is used to convert analog input signal to digital output bits. When the conversion is completed, an A/D interrupt takes place.

2. UERRI Interrupt: UART receive error interrupt

3. UTXI Interrupt: UART transfer buffer empty interrupt

4. URXI Interrupt: UART receive buffer full interrupt

5. SRBFI Interrupt: SPI read buffer full interrupt

Code Example:

```
; === Peripheral Interrupt
PERIPH:
      S0CALL    PUSH
      JBS       INTSTA,ADIF,toADINT
      JBS       INTSTA,UERRI,toUERRINT
      JBS       INTSTA,UTXI,toUTXINT
      JBS       INTSTA,URXI,toURXINT
      JBS       SPISTA,SRBFI,toSPINT
      SJMP      POP
```

```
; -- A/D Interrupt
toADINT:
     BC         INTSTA,ADIF
      :
     SJMP       POP
; --- UART Receiving Error Interrupt
     toUERRINT:
     BC         INTSTA,UERRI
      :
     SJMP       POP
; --- UART Tx Buffer Full Interrupt
toUTXINT:
     BC         INTSTA,UTXI
      :
     SJMP       POP
; --- UART Rx Buffer Full
Interrupt
toURXINT:
     BC         INTSTA,URXI
      :
     SJMP       POP
; --- SPI interrupt
toSPINT:
     BC         SPISTA,SRBFI
      :
     SJMP       POP
```

7.6 Extended MCU Mode Memory Access

- Extended MCU mode: Both internal PROM / external DROM are available. (PMD=0, add external memory)

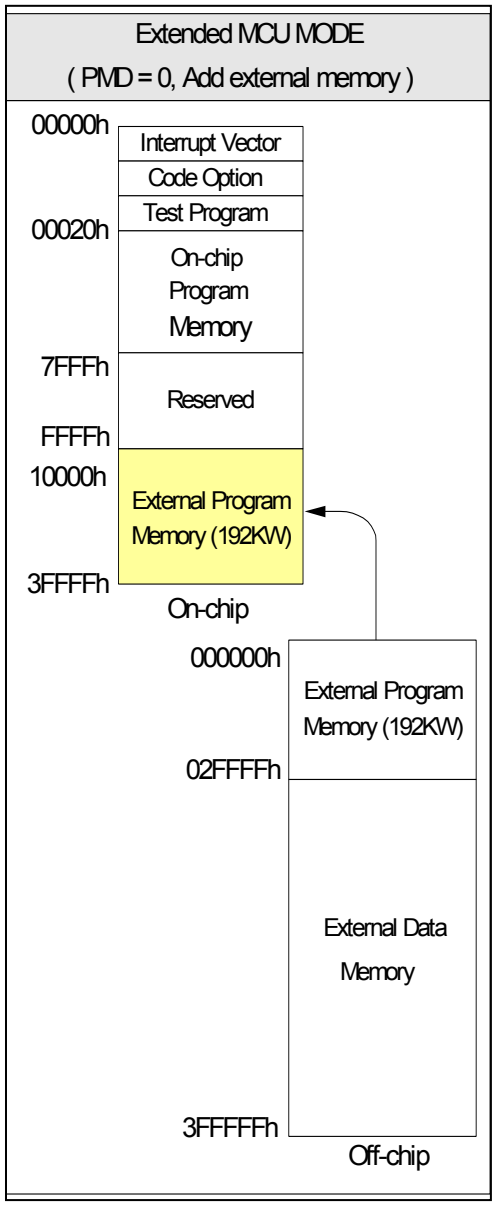


Fig. 7-8 Memory Map for the Extended MCU Mode

7.6.1 MCU External Program Memory Access Timing

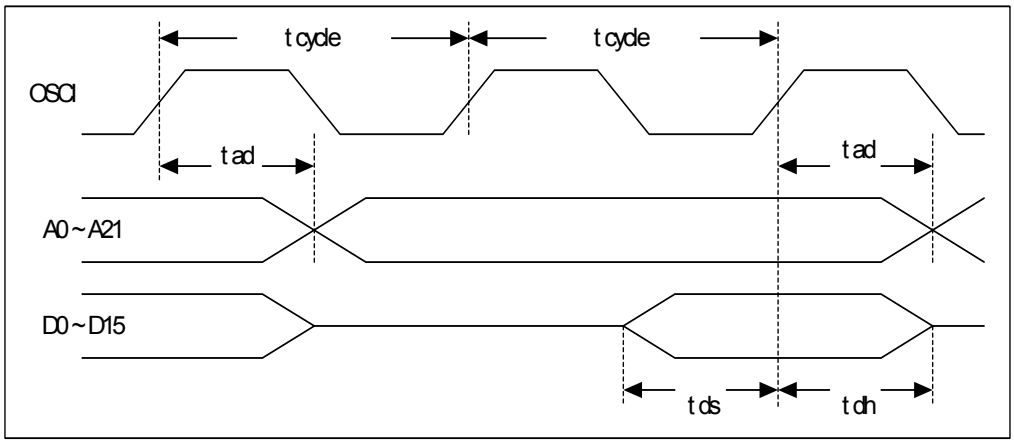


Fig. 7-9 MCU External Program Memory Access Timing Diagram

Symbol	Characteristics	Min.	Typ.	Max.	Unit
t cycle	Clock cycle time	62.5	-	-	ns
t ad	Address delay time	-	-	80	ns
t ds	Data setup up time	10	-	-	ns
t dh	Data hold time	10	-	-	ns

* "Fsys/2" is used to obtain the external PROM instruction speed

External interface access speed formula (Text**:** *Flash or Mask ROM access time*)

Extended MCU mode**:** FPLL = 2 / [Text. + 100ns (> 3V), 120ns (> 2.7V) or 140ns (> 2.4V)]

Example: Flash memory access time**:** 70ns & operator voltage: 3V (Extended MCU mode)

Calculating the PLL maximum speed = 2 / [70ns + 100ns] = 11 MHz

7.6.1.1 MCU and Extended MCU Mode Circuit

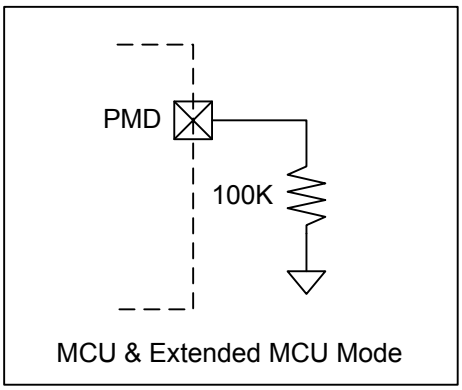


Fig. 7-10 MCU and Extended MCU Mode Circuit Diagram

7.7 Program ROM Map

PMD=0
(Extended MCU mode):
Internal: 8K word x 4 Segments = 32K words
External: 8K word x 24 Segments = 192K words

Addr.	Segment	Addr.	Segment
0000h | 000Bh	Interrupt Vector (12 words)	20000h | 23FFFh	Segment 16 | Segment 17
000Ch | 000Fh	Code Option (4 words)		
0010h | 001Fh	Test Program (16 words)		
0020h | 3FFFh	Segment 0 | Segment 1		
4000h | 7FFFh	Segment 2 | Segment 3	24000h | 27FFFh	Segment 18 | Segment 19
8000h | BFFFh	Not Implemented	28000h | 2BFFFh	Segment 20 | Segment 21
C000h | FFFFh	Not Implemented	2C000h | 2FFFFh	Segment 22 | Segment 23
10000h | 13FFFh	Segment 8 | Segment 9	30000h | 33FFFh	Segment 24 | Segment 25
14000h | 17FFFh	Segment 10 | Segment 11	34000h | 37FFFh	Segment 26 | Segment 27
18000h | 1BFFFh	Segment 12 | Segment 13	38000h | 3BFFFh	Segment 28 | Segment 29
1C000h | 1FFFFh	Segment 14 | Segment 15	3C000h | 3FFFFh	Segment 30 | Segment 31

7.8 RAM Map Registers

(RAM Size: 128 Bytes + 32 Banks × 128 Bytes = 4224 Bytes)

■ Special & Control Registers of RAM:

Legend: R = Readable bit **W** = Writable bit **–** = unimplemented, read as "0"

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	INDF0	R/W							
		Indirect Addressing Pointer 0							
1	FSR0	R/W							
		File Select Register 0 for INDF0							
2	PCL	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
3	PCM	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8
4	PCH							R/W	R/W
		-	-	-	-	-	-	PC17	PC16
5	BSR				R/W				
		-		-	Bank Select Register for INDF0 & General RAM				
6	STKPTR	R/W							
		Stack Pointer							
7	BSR1				R/W				
		-		-	Bank Select Register 1 for INDF1.				
8	INDF1	R/W							
		Indirect Addressing Pointer 1							
9	FSR1	R	R/W						
		1	File Select Register 1 for INDF1						
A	ACC	R/W							
		Accumulator							
B	TABPTRL	R/W							
		Table Pointer Low							
C	TABPTRM	R/W							
		Table Pointer Middle							
D	TABPTRH	R/W							
		Table Pointer High							
E	CPUCON	R/W	-	-	R/W	R/W	R/W	R/W	R/W
		PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0
F	STATUS	R	R	R/W	R/W	R/W	R/W	R/W	R/W
		/TO	/PD	SGE	SLE	OV	Z	DC	C
10	TRL2	R/W							
		Timer 2 Reload Register							
11	PRODL	R/W							
		Multiplier Product Low							
12	PRODH	R/W							
		Multiplier Product High							
13	ADOTL	R/W	R/W	R/W	-	-	R/W	R	R
		WDTEN	EXTMEM	ADWKEN	-	-	FSS	ADOT1	ADOT0
14	ADOTH	R	R	R	R	R	R	R	R
		ADOT9	ADOT8	ADOT7	ADOT6	ADOT5	ADOT4	ADOT3	ADOT2

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
15	UARTTX	W	W	W	W	W	W	W	W
		TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
16	UARTRX	R	R	R	R	R	R	R	R
		RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
17	Port A	R	R	R	R	R	R	R	R
		A.7	A.6	A.5	A.4	A.3	A.2	A.1	A.0
18	Port B	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
19	Port C	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		C.7	C.6	C.5	C.4	C.3	C.2	C.1	C.0
1A	Port D	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.0
1B	Port E	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		E.7	E.6	E.5	E.4	E.3	E.2	E.1	E.0
1C	Port F	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		F.7	F.6	F.5	F.4	F.3	F.2	F.1	F.0
1D	Port G	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		G.7	G.6	G.5	G.4	G.3	G.2	G.1	G.0
1E	Port H	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		H.7	H.6	H.5	H.4	H.3	H.2	H.1	H.0
1F	Port I	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		I.7	I.6	I.5	I.4	I.3	I.2	I.1	I.0
20	PFS	R/W							
		Target PLL Frequency Select Register							
21	STBCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		UINVEN	/REN	BitST	ALL	STB3	STB2	STB1	STB0
22	INTCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE
23	INTSTA	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I
24	TRL0L	R/W							
		Timer 0 Reload Low Byte Register							
25	TRL0H	R/W							
		Timer 0 Reload High Byte Register							
26	TRL1	R/W							
		Timer 1 Reload Register							
27	TR01CON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		T1WKEN	T1EN	T1PSR1	T1PSR0	IREN	T0CS	T0PSR1	T0PSR0
28	TR2CON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		IRPSR1	IRPSR0	T0FNEN1	T0FNEN0	T2EN	T2CS	T2PSR1	T2PSR0
29	TRLIR	R/W							
		IR Reload Register							
2A	(Reserved)	-							
2B	POST_ID	R/W	-	R/W	R/W	R/W	-	R/W	R/W
		EM_ID	-	FSR1_ID	FSR0_ID	EMPE	-	FSR1PE	FSR0PE
2C	ADCON	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
		DET	VRS	ADEN	PIROB	S/DB	CHS2	CHS1	CHS0
2D	PAINTEN	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE
2E	PAINTSTA	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		PA7I	PA6I	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I
2F	PAWAKE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		WKEN7	WKEN6	WKEN5	WKEN4	WKEN3	WKEN2	WKEN1	WKEN0

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
30	UARTCON	W	R/W	R/W	R/W	R/W	R/W	R	R/W
		TB8	UMODE1	UMODE0	BRATE2	BRATE1	BRATE 0	UTBE	TXE
31	UARTSTA	R	R/W	R/W	R/W	R/W	R/W	R	R/W
		RB8	EVEN	PRE	PRERR	OVERR	FMER R	URBF	RXE
32	Port J	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		J.7	J.6	J.5	J.4	J.3	J.2	J.1	J.0
33	Port K	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		K.7	K.6	K.5	K.4	K.3	K.2	K.1	K.0
34	DCRB	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Bit7DC	Bit6DC	Bit5DC	Bit4DC	Bit3DC	Bit2DC	Bit1DC	Bit0DC
35	DCRC	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Bit7DC	Bit6DC	Bit5DC	Bit4DC	Bit3DC	Bit2DC	Bit1DC	Bit0DC
36	DCRDE	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		EHNPU	ELNPU	EHNDC	ELNDC	DHNPU	DLNPU	DHNDC	DLNDC
37	DCRFG	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		GHNPU	GLNPU	GHNDC	GLNDC	FHNPU	FLNPU	FHNDC	FLNDC
38	DCRHI	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		IHNPU	ILNPU	IHNDC	ILNDC	HHNPU	HLNPU	HHNDC	HLNDC
39	DCRJK	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		KHNPU	KLNPU	KHNDC	KLNDC	JHNPU	JLNPU	JHNDC	JLNDC
3A	PBCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Bit7PU	Bit6PU	Bit5PU	Bit4PU	Bit3PU	Bit2PU	Bit1PU	Bit0PU
3B	PCCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Bit7PU	Bit6PU	Bit5PU	Bit4PU	Bit3PU	Bit2PU	Bit1PU	Bit0PU
3C	PLLF	R							
		Actual PLL Frequency Value Register							
3D	TOCL	R							
		Timer 0 Counting Value Low Byte Register							
3E	TOCH	R							
		Timer 0 Counting Value High Byte Register							
3F	SPICON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		TLS1	TLS0	BRS2	BRS1	BRS0	EDS	DORD	SE
40	SPISTA	R/W		R/W	R/W	R/W	R/W	R/W	R
		WEN	-	SRBFIE	SRBFI	SPWKEN	SMP	DCOL	RBF
41	SPRL	R/W							
		Shift Register Low Byte of SPI							
42	SPRM	R/W							
		Shift Register Middle Byte of SPI							
43	SPRH	R/W							
		Shift Register High Byte of SPI							
44	SFCR	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		AGMD2	AGMD1	AGMD0	WDTPRS1	WDTPRS0	SPHSB	CSB1	CSB0
45	ADDL	R/W							
		Melody Channels 1~4 Address Low Byte Register							
46	ADDM	R/w							
		Melody Channels 1~4 Address Middle Byte Register							
47	ADDH	R/w							
		Melody Channels 1~4 Address High Byte Register							
48	ENV / SPHDR	R/w							
		Melody Channels 1~4 Envelope Register/Speech Data Register							
49	MTCON / SPHTCON	R/w							
		Melody Channels 1~4 Control Register/Speech Control Register							

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
4A	MTRL / SPHTRL	R/w							
		Melody Channels 1~4 Reload Register/Speech Reload Register							
4B	VOCON	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		VOEN	DAC	SETR1	SETR0	PWMPSR	VOL2	VOL1	VOL0
4C	TR1C	R							
		Timer 1 Counting Value Register							
4D	TR2C	R							
		Timer 2 Counting Value Register							
4E	ADCF	R/W							
		A/D Clock Factor Register							
4F	(Reserved)	-							

- Reserved for System Use:

Addr	Un-banked
50h | 53h	Don't use these registers

- Other RAM Un-Banked Registers:

Addr	Un-banked
54h | 7Fh	General Purpose RAM

- RAM Banked Register: (selected by BSR)

Addr.	Bank 0	Bank 1	Bank 2	Bank 3	……..	Bank 31
80h | FFh	General Purpose RAM	General Purpose RAM	General Purpose RAM	General Purpose RAM	……..	General Purpose RAM

7.9 Special and Control Registers Description

- STKPTR (R06h): Stock Pointer

The stack level starts from the bottom going up (in a decreasing order), starting from 0FFh of Bank 31.

Stack located at Bank 30 and 31 from address FFh~80h. Initial stack pointer is 00h. Stack pointer address must be even if starting address setting by user.

Bits 0~6 of STKPTR are used as address pointer from 80h~FFh, Bit 7=1 is used to select Bank 31, Bit 7=0 is used to select Bank 30.

Each INT/CALL will stack two bytes address, total capacity is 128 levels.

- PCL, PCM, PCH (R02h, R03h, R04h): Program Counter Register

Bit 17	Bit 16	Bit 15	…….	Bit 8	Bit 7	…….	Bit 0
PCH		PCM			PCL		

This generates up to 256K×16 on-chip ROM addresses at the relative programming instruction codes.

"S0CALL" loads the low 12 bits of the PC (4K*16 ROM).

"SCALL" or "SJUMP" loads the low 13 bits of the PC (8K×16 ROM).

"LCALL" or "LJUMP" loads the full 18 bits of the PC (256K×16 ROM).

"ADD R2, A" or "ADC R2, A" allows a relative address to be added to the current PC. The carry bit of R2 will automatically carry into PCM, PCH.

Code Example:

```
START:  MOV    A,entry
        MOV    number,a  ;number <-- entry
        LCALL  Indirect_JUMP
AAA:    ........
        .......
```

```
Indirect_JUMP:
    MOV      A,number
    ADD      A,ACC       ;A<-- 2*A
    ADD      PCL,A       ;PCL<-- PCL+A
Function_table:
    LJMP  function_address_1  ; number=0
    LJMP  function_address_2  ; number=1
    LJMP  function_address_3  ; number=2
    LJMP  function_address_4  ; number=3
    LJMP  function_address_5  ; number=4
    LJMP  function_address_6  ; number=5
    LJMP  function_address_7  ; number=6
    ........
function_address_1:
    ;Function 1 operation
    ......
    RET   ;PC will return to AAA label
```

- ACC (R0Ah): Accumulator

  Internal data transfer, or instruction operand holding.

- POST_ID (R2Bh): Post Increase / Decrease Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EM_ID	-	FSR1_ID	FSR0_ID	EMPE	-	FSR1PE	FSR0PE

Bit 0 (FSR0PE): Enable FSR0 post increase/decrease function, FSR0 will NOT carry into or borrow from BSR.

Bit 1 (FSR1PE): Enable FSR1 post increase/decrease function, FSR1 will carry into or borrow from BSR1.

Bit 4 (FSR0_ID): Set to 1 means auto increase, reset to 0 means auto decrease FSR0.

Bit 5 (FSR1_ID): Set to 1 means auto increase, reset to 0 means auto decrease FSR1.

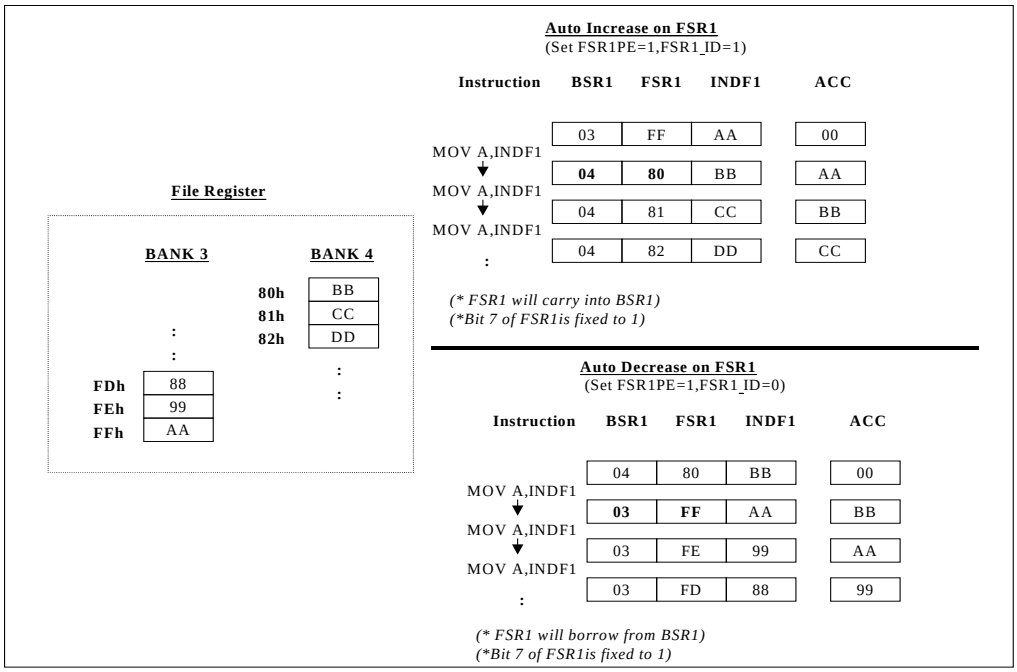
✧ Indirect Addressing Pointer 0

- BSR (R05h) determines which bank is active (working bank) among the 32 banks (Bank 0 ~ bank 31).
- FSR0 (R01h) is an Address Register for INDF0. You can select up to 256 bytes (Address: 00 ~ 0FFh).
- INDF0 (R00h) is not a physically implemented register.

✧ Indirect Addressing Pointer 1

- BSR1 (R07h) is a bank register for INDF1. Cannot determine the working bank for the general register.
- FSR1 (R09h) is an Address Register for INDF1. You can select up to 128 bytes (Address: 80 ~ 0FFh); Bit 7 of FSR1 is fixed to 1.
- INDF1 (R08h) is not a physically implemented register.

The Linear Addressing Capability of INDF1 is shown below:



Code Example:

```
;*****************************************
;*      Const => Working bank setting
;*      REG => Save or Recall register
;*****************************************
; ***** RAM stack macro
; *** Initial RAM stack
IniRAMsk  MACRO     #Const
          MOV       A,#Const
          MOV       BSR1,A
          CLR       FSR1
          BS        POST_ID,FSR1PE
          ENDM
; *** Push RAM stack
PushRAM   MACRO     REG
          BS        POST_ID,FSR1_ID
          MOVRP     INDF1,REG
          ENDM
; *** Pop RAM stack
PopRAM    MACRO     REG
          BC        POST_ID,FSR1_ID
          MOVPR     REG,INDF1
          ENDM
```

```
; *** Main Start Program
Mstart:
       :
       :
       IniRAMsk   #29
       :
       :
MnLoop:
       :
       :
LJMP     MnLoop

; *** Interrupt routine
IntSR:
       PushRAM    ACC
       PushRAM    Status
       :
       :
       :
       :
       PopRAM     Status
       PopRAM     ACC
       RETI
```

Data Transform Bank 0 to Bank 1:

```
     MOV       A,#00110011B      ; Enable FSR0 & FSR1 post increase
     MOV       POST_ID,A
     BANK      #0                ; BSR = 0 working bank
     MOV       A,#1
     MOV       BSR1,A            ; BSR1 = 1 is Bank 1
     MOV       A,#80H
     MOV       FSR0,A            ; FSR0 = 80H
     CLR       FSR1               ; FSR1 = 80H
     MOV       A,#80H
     RPT       ACC
     MOVRP     INDF1,INDF0       ; Move 80H ~ 0FFH data to Bank 1
      :
```

- TABPTRL, TABPTRM, TABPTRH (R0Bh, R0Ch, R0Dh): Table Pointer Register

Bit 23	…….	Bit 16	Bit 15	…….	Bit 8	Bit 7	…….	Bit 0
	TABPTRH			TABPTRM			TABPTRL	

Program ROM or external memory address register

Bit 23: used to select internal/external memory

Bit 22 ~ Bit 1: used to point the memory address

Bit 0: used to select the whether low or high byte of the word pointed to. (See TBRD instruction).

Code Example:

```
;  ***  Program ROM
       :
       :
    TBPTH    #(PROMTabB*2)/10000H
    TBPTM    #(PROMTabB*2)/100H
    TBPTL    #PROMTabB*2
       :
       :
    TBRD     0,ACC    ;no change
    TBRD     1,ACC    ;auto-increase
    TBRD     2,ACC    ;auto-decrease
       :
       :
;***Program ROM Data
PROMTabB:
    DB
    0x00,0x01,0x02,0x03,0x04,0x05
    DB
    0x10,0x11,0x12,0x13,0x14,0x15
    DB
    0x20,0x21,0x22,0x23,0x24,0x25
```

```
;  ***  External Data ROM
INCLUDE    "DROM_E.hdr"   ;to ROM Converter
         :
         :
    TBPTL    #_Data_l
    TBPTM    #_Data_m
    TBPTH    #_Data_h
    BS       TABPTRH,7
    :
    :
    TBRD     0,ACC     ;no change
    TBRD     1,ACC     ;auto-increase
    TBRD     2,ACC     ;auto-decrease
        :
```

- PRODL, PRODH (R11h, R12h): Multiplier Product Low/High

An unsigned or signed 8 × 8 hardware multiplier is included in the microcontroller. The result is stored into the 16 bits product register.

- CPUCON (R0Eh): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0

Bit 3 (SMIER): Signed or unsigned selection bit of the Multiplier (ACC)
"**0**": unsigned Multiplier
"**1**": signed Multiplier

Bit 4 (SMCAND): Signed or unsigned selection bit of the Multiplicand (Constant or Register)
"**0**": unsigned Multiplier
"**1**": signed Multiplier

Code Example:

```
;***Signed Multiplier Operation
;===PRODH:PRODL = A x REG
      BS       CPUCON,SMIER
      BS       CPUCON,SMCAND
      MUL      A,REG
```

```
;***Unsigned Multiplier Operation
;===PRODH:PRODL = A x #k
      BC       CPUCON,SMIER
      BC       CPUCON,SMCAND
      MUL      A,# 88
```

- Port A (R17h): General Input Registers

- STBCON (R21h): Strobe Output Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UINVEN	/REN	BitST	ALL	STB3	STB2	STB1	STB0

Bit 6 (/REN): Port A (7 ~ 0) pull up resistor control bit
"**0**" enable pull-up resistor
"**1**" disable pull-up resistor

- PAINTEN (R2Dh): Port A Interrupt Control Register
"**0**" disable interrupt function
"**1**" enable interrupt function

- PAINTSTA (R2Eh): Port A Interrupt Status Register
Set to "**1**" when pin falling edge is detected
Clear ("**0**") by software

- PAWAKE (R2Fh): Port A Wake-up Control Register
"**0**" disable wake-up function
"**1**" enable wake-up function

- Port B, Port C (R18h, R19h): General I/O Registers

- DCRB, DCRC (R34h, R35h): Port B & Port C Direction Control Registers
Bit 7 ~ Bit 0 (Bit 7DC ~ Bit 0DC)
"**0**" set to output pin
"**1**" set to input pin

- PBCON, PCCON (R3Ah, R3Bh): Port B & Port C Pull-up Resistor Control Registers

  Bit 7 ~ Bit 0 (Bit 7PU ~ Bit 0PU)

  "**0**" disable the pull-up resistor

  "**1**" enable the pull-up resistor

- Port D (R1Ah,): General I/O or External Memory Address Bus and Control Pin

- Port E (R1Bh): External Memory Address Bus Registers

- DCRDE (R36h): Port D & Port E Direction Control & Pull-up Resistor Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EHNPU	ELNPU	EHNDC	ELNDC	DHNPU	DLNPU	DHNDC	DLNDC

  Bit 1, Bit 5 (DHNDC, EHNDC) & Bit 0, Bit 4 (DLNDC, ELNDC): Port D & E high / low nibbles direction control

  "**0**": set as output pin

  "**1**": set as input pin

  Bit 3, Bit 7 (DHNPU, EHNPU) & Bit 2, Bit 6 (DLNPU, ELNPU): Enable Ports D & E high / low nibbles pull-high resistor

  "**0**": disable pull-high resistor

  "**1**": enable pull-high resistor

- Port F (R1Ch): External Memory Address Bus

- Port G (R1Dh): External Memory Data Bus

- DCRFG (R37h): Direction Control & Pull-high Resistor Control of Port F & Port G

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GHNPU	GLNPU	GHNDC	GLNDC	FHNPU	FLNPU	FHNDC	FLNDC

  Bit 1, Bit 5 (FHNDC, GHNDC) & Bit 0, Bit 4 (FLNDC, GLNDC): Port F & G high / low nibbles direction control

  "**0**": set as output pin

  "**1**": set as input pin

  Bit 3, Bit 7 (FHNPU, GHNPU) & Bit 2, Bit 6 (FLNPU, GLNPU): Enable Port F & G high / low nibbles pull-high resistor.

  "**0**": disable pull-high resistor

  "**1**": enable pull-high resistor

- Port H (R1eh): External Memory Data Bus

- Port I (R1fh): General I/O Register

- DCRHI (R38h): Direction Control & Pull-High Resistor Control Of Port H & Port I

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IHNPU	ILNPU	IHNDC	ILNDC	HHNPU	HLNPU	HHNDC	HLNDC

Bit 1, Bit 5 (HHNDC, IHNDC) & Bit 0, Bit 4 (HLNDC, ILNDC): Port H & I high / low nibbles direction control.
"**0**" : set as output pin
"**1**" : set as input pin

Bit 3, Bit 7 (HHNPU, IHNPU) & Bit 2, Bit 6 (HLNPU, ILNPU): Enable Port H & I high / low nibbles pull up resistor.
"**0**": disable pull up resistor
"**1**": enable pull up resistor

- Port J, Port K (R32h, R33h): General I/O Registers

- DCRJK (R39h): Direction Control & Pull-up Resistor Control of Ports J & K

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KHNPU	KLNPU	KHNDC	KLNDC	JHNPU	JLNPU	JHNDC	JLNDC

Bit 1, Bit 5 (JHNDC, KHNDC) & Bit 0, Bit 4 (JLNDC, KLNDC): Port J & K high / low nibbles direction control.
"**0**" : set as output pin
"**1**" : set as input pin

Bit 3, Bit 7 (JHNPU, KHNPU) & Bit 2, Bit 6 (JLNPU, KLNPU): Enable Ports J & K high / low nibbles pull up resistor.
"**0**" : disable pull-high resistor
"**1**" : enable pull-high resistor

Code Example:

```
;***Port A function
;---Port A interrupt
INPTINT:
       PUSH
       MOV     A,PAINTSTA
;---Port A interrupt
       JBS     STATUS,F_Z,Q_PAINT
       MOV     PORTH,A
Q_PAINT:
       POP
       RETI
;---Port H output
       CLR     DCRHI
;---Port A pull-up enable
       BC      STBCON,REN
       CLR     PAINTSTA
       MOV     A,#11111111B
;---Port A interrupt
       MOV     PAINTEN,A
;---Port A wakeup
       MOV      PAWAKE,A
       BS       CPUCON,GLINT
;---Sleep mode
       BC       CPUCON,MS1
       SLEP
       NOP
         :
         :
```

```
;***Output function => 0XAAh to all port
       CLR        DCRB
       CLR        DCRDE
       CLR        DCRJK
       MOV        A,#0XAA
       MOV        PORTB,A
       MOV        PORTD,A
       MOV        PORTE,A
       MOV        PORTJ,A
       MOV        PORTK,A
;
;***Input function => Input all ports
to RAM 80 ~ 85h
       BC         STBCON,REN
       MOV        A,#0XFF
       MOV        DCRC,A
       MOV        PCCON,A
       MOV        DCRFG,A
       MOV        DCRHI,A
       BS           POST_ID,FSR1_ID
       BS           POST_ID,FSR1PE
       CLR          BSR1
       CLR          FSR1
       MOVRP        INDF1,PORTA
       MOVRP        INDF1,PORTC
       MOVRP        INDF1,PORTF
       MOVRP        INDF1,PORTG
       MOVRP        INDF1,PORTH
       MOVRP        INDF1,PORTI
```

8 Peripheral

8.1 Timer 0 (16 Bits Timer with Capture & Event Counter Functions)

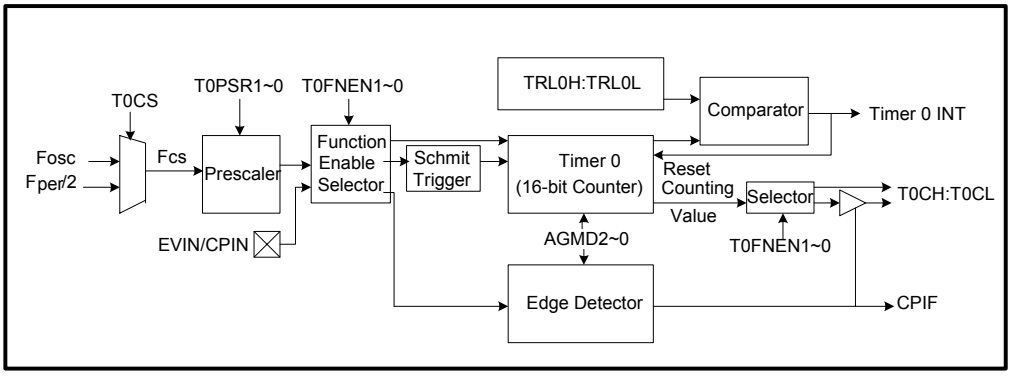


Fig. 8-1 Timer 0 Function Block Diagram

✧ **Timer 0 Mode**

In this mode, Timer 0 is used as a general-purpose 16-bit up counter. There is an interrupt available for user's application.

There is a prescaler for the timer. The T0PSR2~T0PSR0 bits of the TR01CON register determine the prescaler ratio and generate different clock rates for the timer clock source. Counter value will be incremented by one (counting up) according to the timer clock source and stored into the T0CH: T0CL register. The clock source (Fcs) is selected from Fosc or Fper/2 by T0CS and pre-scaled by T0PSR1~0. When the counting value is larger than TRL0H: TRL0L value, Timer 0 interrupt will occur, and the counter value will be reset to zero automatically.

$$T = \frac{1}{Fcs} \times \Pr escaler \times \left(TRL0H : TRL0L + 1\right)$$

Timer 0 Frequency:

Clock Source	Fper / 2	TRL0H:TRL0L	Prescaler	Timer 0 Freq.
Fosc (32.768kHz)	-	FFFFh	1:64	128Hz
Fpll (8MHz)	4MHz	00FFh	1:1	15.6kHz
Fosch (16MHz)	8MHz	00FFh	1:1	31.2kHz

✧ Capture Mode: CPIN (Port B.5) Pin

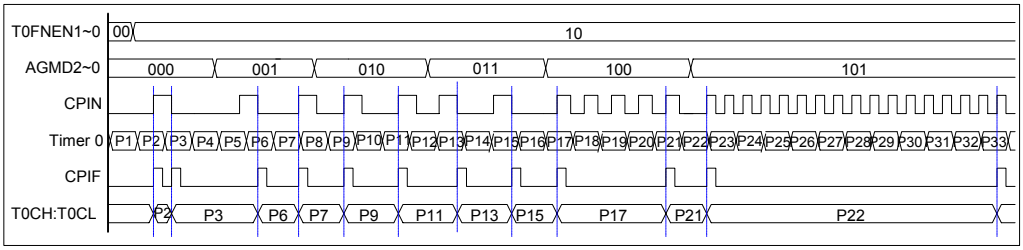
Capture is a function that captures the Timer 0 value when an event occurs on the CPIN pin.

The counter value is captured at: 1st rising edge, 2nd falling edge, etc.; 1st falling edge, 2nd rising edge, etc.; every rising edge or every falling edge selected by AGMD2~0 bit of the SFCR register. When an event edge is detected from the CPIN input pin, the interrupt flag CPIF is set. If a new event edge is detected before the old value in T0CH: T0CL register is read, the old captured value will be lost.

The CPIN pin should be configured in capture function input by setting T0FNEN1~0 bits of TR2CON register.

$$T = \frac{1}{Fcs} \times \Pr escaler \times \left[(T0CH:T0CL)_{NEW} - (T0CH:T0CL)_{OLD} \right]$$

Capture Mode Example:

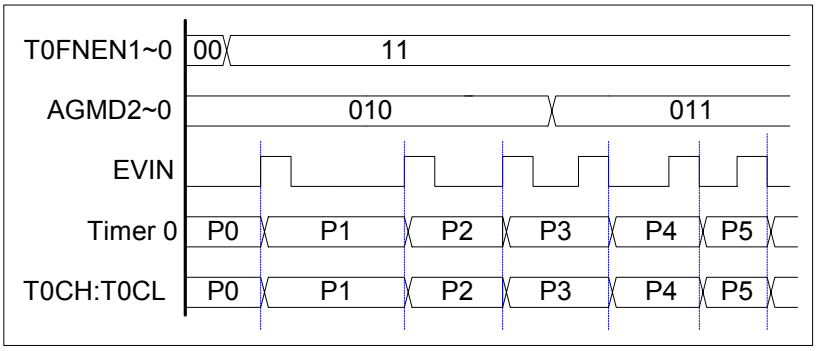


✧ Event Counter Mode: EVIN (Port B.5) Pin

Event counter is a function wherein the 16-bit counter value increments by one when an event occurs on EVIN pin at: every rising edge or every falling edge selected by AGMD2~0 bit of the SFCR register. In other words, the Timer 0 clock source is from an external event (EVIN pin).

The EVIN pin should be configured in event counting function input by setting the T0FNEN1~0 bits of the TR2CON register. The counting value of Timer 0 will be stored in T0CH: T0CL registers.

Event Counter Mode Example:



- TRL0H, TRL0L (R25h, R24h): Used to store the value compared with Timer 0 register.
- T0CH, T0CL (R3Eh, R3Dh): Used to store the Timer 0 counting value in Timer 0 mode and Event counter mode. But in Capture mode, it is used to store the captured value.
- TR01CON (R27h): Timer 0 and Timer 1 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1WKEN	T1EN	T1PSR1	T1PSR0	IREN	T0CS	T0PSR1	T0PSR0

Bit 1 ~ Bit 0 (T0PSR1~T0PSR0): Timer 0 Prescaler Select Bit

T0PSR1: T0PSR0	Prescaler Value
00	1:1
01	1:4
10	1:16
11	1:64

Bit 2 (T0CS): Timer 0 Clock Source Select Bit.

"**0**": Clock source is from Fosc
"**1**": Clock source is from Fper/2

- TR2CON (R28h): Timer 2 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRPSR1	IRPSR0	T0FNEN1	T0FNEN0	T2EN	T2CS	T2PSR1	T2PSR0

Bit 5 ~ Bit 4 (T0FNEN1 ~ T0FNEN0): Timer 0 and Capture, event counter mode select bits.

- SFCR (R44h): Special Function Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGMD2	AGMD1	AGMD0	WDTPSR1	WDTPSR0	SPHSB	CSB1	CSB0

Bit 7 ~ Bit 5 (AGMD2 ~ AGMD0): Capture and Event Counter function edge detector select bits.

T0FNEN 1 ~ 0	Mode	AGMD 2~0	Edge Mode
00	Disable	-	-
01	Timer 0	-	-
10	Capture	000	1st Rising edge, 2nd falling edge, etc.
		001	1st Falling edge, 2nd rising edge, etc.
		010	Every rising edge
		011	Every falling edge
		100	Every 4th rising edge
		101	Every 16th rising edge
11	Event Counter	010	Every rising edge
		011	Every falling edge

Note: 1. In changing from one mode to another, it is necessary to disable the Timer 0.
2. To avoid error, simultaneously setup T0FNEN1 and T0FNEN0. Do not use Bit manipulation instruction. It is recommended to use only the Data Transfer Instruction, particularly MOV A,#k, MOV A,r.

- CPUCON (R0Eh): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0

Bit 2 (GLINT): Global Interrupt Control Bit
"**0**" : Disable all interrupts
"**1**" : Enable all un-masked interrupts

- INTCON (R22h): Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 0 (TMR0IE): Timer 0 Interrupt Control Bit
"**0**" : Disable Timer 0 interrupt
"**1**" : Enable Timer 0 interrupt

Bit 7 (CPIE): Capture Interrupt Control bit
"**0**" : Disable Capture interrupt
"**1**" : Enable Capture interrupt

- INTSTA (R23h): Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 0 (TMR0I): Set to **1** when Timer 0 is larger than TRL0H ~ TRL0L value
Clear to **0** by software or Timer 0

Bit 7 (CPIF): Set to **1** when Capture input edge is detected
Clear to **0** by software or disable Capture

Code Example:

```
; === Timer 0 Interrupt
TIMERINT:
    PUSH
    JBC       INTSTA,TMR0I,Q_Time
    BC        INTSTA,TMR0I
    BTG       PORTC,3
    Q_Time:
    POP
    RETI
; === Timer 0 = (8M/2)/[4x3FFF + 1]
Timer0SR:
     :
    System setting 8MHz
    PC.2 Port H & G setting output port
     :
; --- Fpll & Prescaler 1:4
    MOV       A,#00000101B
    MOV       TR01CON,A
; --- 4ms = (4 x 16383 + 1)/(8M/2)
```

```
; === Capture Input Interrupt
CAPINT:
    PUSH
    JBS       INTSTA,CPIF,Q_ICAP
    BC        INTSTA,CPIF
    BTG       PORTC,3
    BS        INTFLAG,F_ICAP
    Q_ICAP:
    POP
    RETI
    ;
;==1st falling edge, 2nd rising edge, etc.
    CAP_SR:
    System setting 8MHz
    PC.2 Port H & G setting output port
    User setting F_ICAP flag
     :
; --- Count end → 0FFFFH
    MOV       A,#0XFF
```

```
    MOV      A,#0FFH
    MOV       TRL0L,A
    MOV      A,#03FH
    MOV      TRL0H,A
; --- Timer 0 mode
    MOV      A,#00010000B
    MOV      TR2CON,A
; --- Timer 0 interrupt enable
    BS       INTCON,TMR0IE
; --- Clear Timer 0 interrupt status
    BC       INTSTA,TMR0I
; --- Enable global interrupt
    BS       CPUCON,GLINT
    TimeLoop:
; --- Out Timer 0 count to Port H:G
    MOVRP    PORTH,T0CH
    MOVRP    PORTG,T0CL
    SJMP     TimeLoop
```

```
    MOV      TRL0H,A
    MOV      TRL0L,A
; --- PLL/2 & Prescaler 1:1
; --- (8MHz/2)/65536=61Hz
    MOV      A,#00000100B
    MOV      TR01CON,A
; --- 1st Falling - 2nd Rising
    MOV      A,#00100000B
    MOV      SFCR,A
    BS       INTCON,CPIE
; --- 10 → Capture Enable
    MOV      A,#00100000B
    MOV      TR2CON,A
    BC       INTFLAG,F_ICAP
    BS       CPUCON,GLINT
CAP_LOOP:
    JBC      INTFLAG,F_ICAP,

CAP_LOOP
    BC       INTFLAG,F_ICAP
; --- Out capture count to Port H:G
    MOVRP    PORTH,T0CH
    MOVRP    PORTG,T0CL
    SJMP     CAP_LOOP
```

```
;  ===  Every rising edge
EVcntSR:
        :System setting 8MHz
       Ports H & G setting output port
        :
       MOV        A,#0XFF          ; Switch 256 times reload
       MOV        TRL0L,A
       CLR        TRL0H            ; Count start 0000H
       BS         TR01CON,T0CS     ; Fper/2
       MOV        A,#01000000B
       MOV        SFCR,A           ; Rising edge
       MOV        A,#00110000B
       MOV        TR2CON,A         ; 11 → Event count Enable
EV_LOOP:
       MOVRP      PORTH,T0CH       ; Out event count to Port H:G
       MOVRP      PORTG,T0CL
       SJMP       EV_LOOP
```

8.2 Timer 1 (8 Bits)

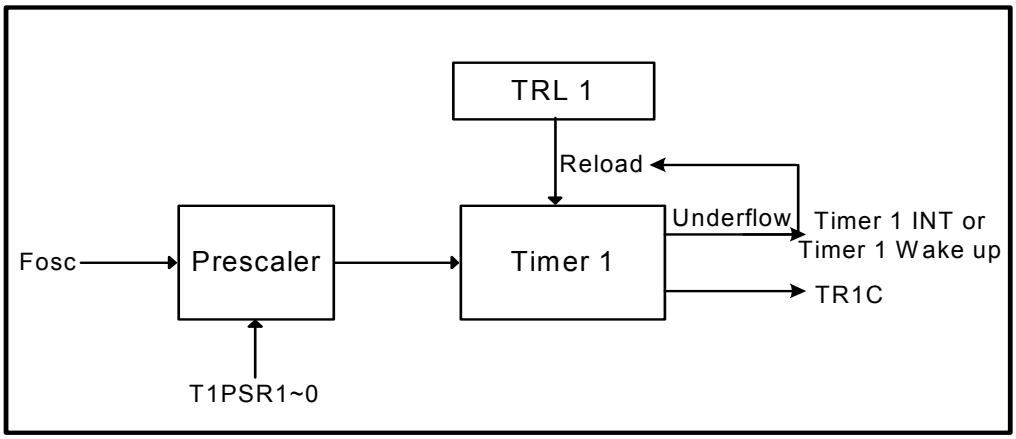


Fig. 8-2 Timer 1 Function Block Diagram

Timer 1 is a general-purpose 8-bit down counter for some applications requiring time counting. There are interrupt and wake up function available for user's application. The clock source is from the oscillator clock.

There is a prescaler for the timer. The **T1PSR1~T1PSR0** bits of the TR01CON register determine the prescaler ratio and generate different clock rates for the timer clock source. Setting **T1WKEN** bit of the TR01CON register to 1 will enable the Timer 1 underflow wake-up function in IDLE MODE.

Counting value will be decremented by one (count down) in accordance with the real timer clock source. When an underflow occurs, the timer interrupt will be triggered if the global interrupt and Timer 1 interrupt are both enabled. At the same time, the TRL1 value will be automatically reloaded into the 8 bits counter.

$$T = \frac{1}{Focs} \times \Pr escaler \times (TRL1 + 1)$$

The Timer 1 frequency range is from 0.5Hz (TRL1 = 0FFh, prescaler = 1: 256) to 8.192kHz (TRL1 = 0h, prescaler = 1:4). The clock source is from the oscillator clock (Fosc).

- TRL1 (R26h): Used to store the auto-reload value of Timer 1. When enabling Timer 1 or an underflow occurs, TRL1 register value will automatically be reloaded into the 8 bits counter.

- TR1C (R4Ch): Used to store the Timer 1 Counting Value

- TR01CON (R27h): Timer 0 and Timer 1 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1WKEN	T1EN	T1PSR1	T1PSR0	IREN	T0CS	T0PSR1	T0PSR0

Bit 5 ~ Bit 4 (T1PSR1~T1PSR0): Timer 1 Prescaler select bit

T1PSR1: T1PSR0	Prescaler Value
00	1:4
01	1:16
10	1:64
11	1:256

Bit 6 (T1EN): Timer 1 Enable Control Bit

"**0**": disable Timer 1 (stop counting)

"**1**": enable Timer 1

Bit 7 (T1WKEN): Enable bit of Timer 1underflow wake up function in Idle Mode.

"**0**": Disable Timer 1 wake up function

"**1**": Enable Timer 1 wake-up function

- CPUCON (R0Eh): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0

Bit 2 (GLINT): Global Interrupt control bit

- INTCON (R22h): Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 1 (TMR1IE): Timer 1 Interrupt Control bit

"**0**": disable Timer 1 interrupt

"**1**": enable Timer 1 interrupt

- INTSTA (R23h): Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 1 (TMR1I): Set to **1** when Timer 1 interrupt occurs. Clear to **0** by software or disable Timer 1.

Code Example:

```
; === Timer 1 interrupt
TIMERINT:
      PUSH
      JBC   INTSTA,TMR1I,Q_Time
      BC    INTSTA,TMR1I
      BTG   PORTC,3
Q_Time:
      POP
      RETI
; === Timer 1 = 32.768K/[256 x 3F + 1]
Timer1SR:
       :
      PC.2 setting output port
       :
      MOV   A,#10110000B
      MOV   TR01CON,A         ; Fosc & Prescaler 1:256 & wakeup
      MOV   A,#03FH
      MOV   TRL1,A            ; 0.5sec = (256 x 63 + 1)/32.768K
      BS    TR01CON,T1EN      ; Timer 1 enable
      BS    INTCON,TMR1IE     ; Timer 1 interrupt enable
      BC    INTSTA,TMR1I      ; Clear Timer 1 interrupt status
      BS    CPUCON,GLINT      ; Enable global interrupt
      BS    CPUCON,MS1        ; Idle mode
T1WLoop:
      SLEP
      NOP
       :
      SJMP  T1Wloop
```

8.3 Timer 2 (8 Bits):

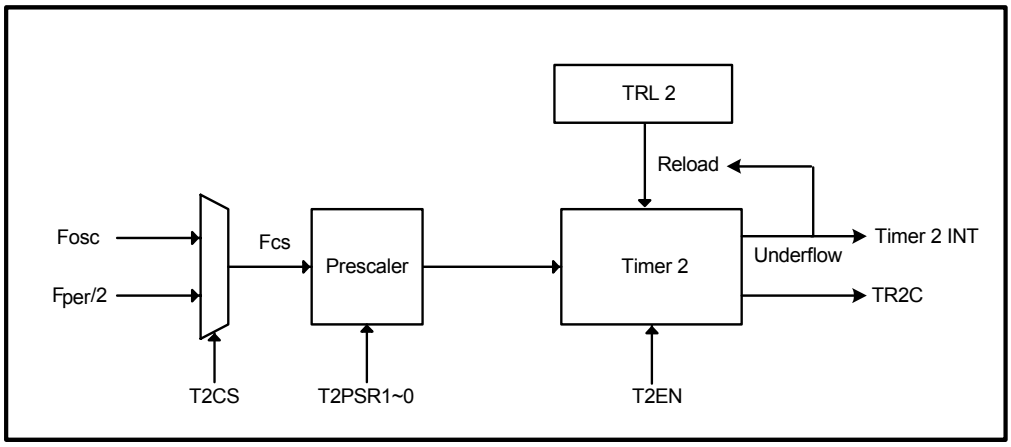


Fig. 8-3 Timer 2 Function Block Diagram

Timer 2 is a general-purpose 8-bit down counter for some applications requiring time counting. There are interrupt functions available for user's application. The clock source (Fcs) is from the oscillator clock or Fper/2.

There is a prescaler for the timer. The T2PSR1~T2PSR0 bits of the TR2CON register determine the prescaler ratio and generate different clock rates for the timer clock source.

Counting value will be decremented by one (counting down) according to the timer clock source. When the counter value underflows, a timer interrupt will occur (if Timer 2 interrupt is enabled).

$$T = \frac{1}{Fcs} \times \Pr escaler \times (TRL2 + 1)$$

Timer 2 Frequency:

Clock Source	Fper / 2	TRL0H:TRL0L	Prescaler	Timer 0 Freq.
Fosc (32.768kHz)	-	FFh	1:8	16Hz
Fpll (8MHz)	4MHz	0Fh	1:1	250kHz
Fosch (16MHz)	8MHz	0Fh	1:1	500kHz

- TRL2 (R10h): used to store the auto-reload value of Timer 2 when Timer 2 is enabled or underflow occurs. TRL2 register will automatically reload into 8 bits counter.

- TR2C (R4Dh): is used to store the Timer 2 counter value

- TR2CON (R28h): Timer 2 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRPSR1	IRPSR0	T0FNEN1	T0FNEN0	T2EN	T2CS	T2PSR1	T2PSR0

Bit 1 ~ Bit 0 (T2PSR1~T2PSR0): Timer 2 Prescaler select bit.

T2PSR1: T2PSR0	Prescaler Value
00	1:1
01	1:2
10	1:4
11	1:8

Bit 2 (T2CS): Timer 2 Clock Source Select Bit

"**0**" : Clock source is from Fosc

"**1**" : Clock source is from Fper/2

Bit 3 (T2EN): Timer 2 Enable Control Bit

"**0**" : Disable Timer 2 (stop counting)

"**1**" : Enable Timer 2

- CPUCON (R0Eh): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0

Bit 2 (GLINT): Global interrupt control bit

- INTCON (R22h): Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 2 (TMR2IE): Timer 2 Interrupt Control bit

"**0**" : Disable Timer 2 interrupt

"**1**" : Enable Timer 2 interrupt

- INTSTA (R23h): Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 2 (TMR2I): Set to **1** when Timer 2 interrupt occurs

Clear to **0** by software or disable Timer 2

Code Example:

```
; === Timer 2 interrupt
TIMERINT:
      PUSH
      JBC   INTSTA,TMR2I,Q_Time
      BC    INTSTA,TMR2I
      BTG   PORTC,3
Q_Time:
      POP
      RETI
; === Timer 2 = (8M/2)/[4 x 3F + 1]
Timer2SR:
       :
      System setting 8MHz
      Port G setting output port
       :
      MOV   A,#00000110B
      MOV   TR2CON,A          ; Fpll & Prescaler 1:4
      MOV   A,#03FH
      MOV   TRL2,A            ; 16us = (4 x 63 + 1)/(8M/2)
      BS    TR2CON,T2EN       ; Timer 2 enable
      BS    INTCON,TMR2IE     ; Timer 2 interrupt enable
      BC    INTSTA,TMR2I      ; Clear Timer 2 interrupt status
      BS    CPUCON,GLINT      ; Enable global interrupt
TMR2Loop:
      MOVRP PORTH,TR2C        ; Out Timer 2 count to Port H
      SJMP  TMR2Loop
```

8.4 IR Generator: IROT (Port B.2) Pin

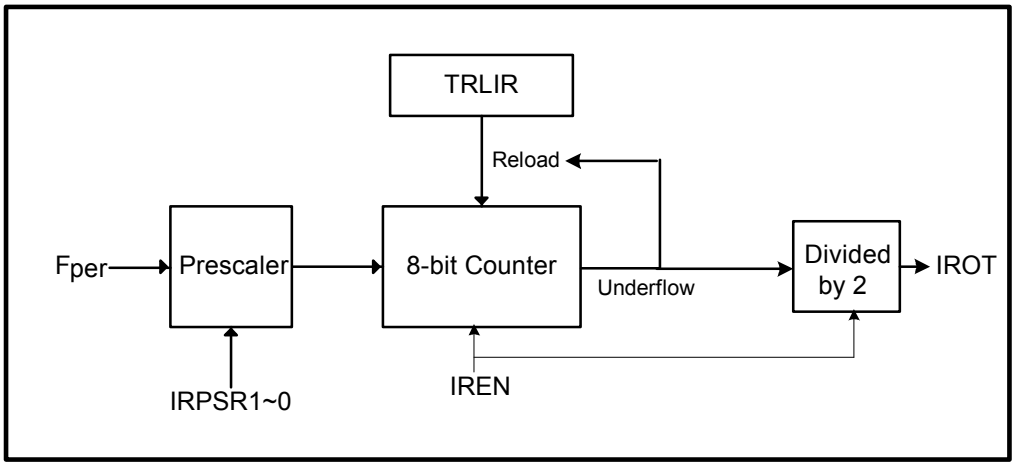


Fig. 8-4 IR Generator Function Block Diagram

IR function is enabled by IREN bit and output on the IROT (Port B.2) pin by a general-purpose 8-bit down counter. When IREN is low, the T-flip-flop should be initialized, as IROT equals zero. The clock source is from the Fper. The IRPSR1 ~ IRPSR0 bits of the TR2CON register determine the prescaler ratio and generate different clock rates for the timer clock source. The counting value will be decremented by one (counting down) according to the clock source. When the counter value underflows, the IR reload register value will be reloaded into the counter.

$$T = \frac{2}{Fper} \times \Pr escale \times (TRLIR + 1)$$

IR Carrier Signal Frequency:

Clock Source	Fper	TRL0H:TRL0L	Prescaler	Timer 0 Freq.
Fpll (8MHz)	8MHz	0Fh	1:1	250kHz
Fosch (16MHz)	16MHz	0Fh	1:1	500kHz

- TRLIR (R29h): Used to store the auto-reload value of the IR generator. When the IR generator is enabled or when an underflow occurs, the TRLIR register value will automatically be reloaded into the 8 bits counter.

- TR01CON (R27h): Timer 0 and Timer 1 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1WKEN	T1EN	T1PSR1	T1PSR0	IREN	T0CS	T0PSR1	T0PSR0

Bit 3 (IREN): IR function enable control bit

"**0**" : Disable IR function and recover IROT pin as a general I/O pin.

"**1**" : Enable IR function and change Port B.2 as IROT output pin.

- TR2CON (R28h): Timer 2 Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRPSR1	IRPSR0	T0FNEN1	T0FNEN0	T2EN	T2CS	T2PSR1	T2PSR0

Bit 7 ~ Bit 6 (IRPSR1~IRPSR0): IR Generator Prescaler Select Bit

IRPSR1: IRPSR0	Prescaler Value
00	1:1
01	1:4
10	1:16
11	1:64

Code Example:

```
; === IR generator 31kHz
      :
      System setting 10MHz
      :
      MOV   A,#10000000B
      MOV   TR2CON,A    ; Prescaler 1: 16
      MOV   A,#9
      MOV   TRLIR,A     ; 10MHz /[ 2 x 16 x ( 9 + 1 ) ] = 31kHz
      BS    TR01CON,IREN
IR_Loop:
      SJMP  IR_Loop
```

8.5 Watchdog Timer (WDT)

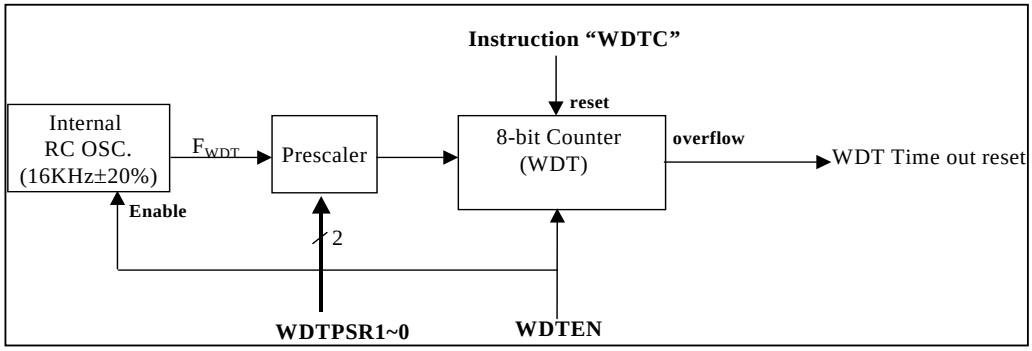


Fig. 8-5 Watchdog Timer Function Block Diagram

The watchdog timer (WDT) clock source is from the on-chip RC oscillator (16kHz ± 20%). The WDT will keep on running even when the oscillator has been turned off (i.e. in Sleep Mode). WDT time-out will cause the MCU to reset (if WDT is enabled). To avoid a WDT reset to occur, user should clear the WDT value by using the "WDTC" instruction before WDT time-out. Setting the WDTEN bit will enable the WDT function. The initial state of WDT or its default condition is WDT disabled. There is also a prescaler to generate different clock rates for the WDT clock source. The prescaler ratio is defined by WDTPSR1 ~ WDTPSR0.

$$T = \frac{1}{F_{WDT}} \times \Pr escaler \times (WDT + 1)$$

The WDT time-out range is 64ms (prescaler=1:4) to 2.048 second (prescaler=1:128).

- ADOTL (R13h): A/D Output Data Low Byte Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTEN	EXTMEM	ADWKEN	-	-	FSS	ADOT1	ADOT0

Bit 7 (WDTEN): Watchdog Timer enable bit

"**0**": disable watchdog timer (stop running)

"**1**": enable watchdog timer

- SFCR (R44h): Special Function Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGMD2	AGMD1	AGMD0	WDTPSR1	WDTPSR0	SPHSB	CSB1	CSB0

Bit 4 ~ Bit 3 (WDTPSR1~WDTPSR0): Watchdog Timer Prescaler select bit

WDTPSR1: WDTPSR0	Prescaler Value
00	1:4
01	1:16
10	1:64
11	1:128

Code Example:

```
; === WDT setting 2.048sec
    :
  Timer 1 (0.5sec wakeup)
    :
  BS    SFCR,WDTPSR0
  BS    SFCR,WDTPSR1 ; Prescaler 1:128
  BC    CPUCON,MS1   ; Change to sleep mode
  WDTC
  SLEP
WDT_Loop:
  SJMP WDT_Loop
```

```
; === Timer 1 interrupt 0.5 sec
TIMERINT:
    PUSH
    JBC     INTSTA,TMR1I,Q_Time
    BC      INTSTA,TMR1I
    WDTC
     :
     :
Q_Time:
    POP
    RETI
```

8.6 Universal Asynchronous Receiver Transmitter (UART)

- RS232C compatible
- Mode selectable (7/8/9 bit) with/without parity bit
- Baud rate selection
- Error detect function
- Interrupt available for Tx buffer empty, Rx buffer full and receiver error
- TXD and RXD port inverse output control

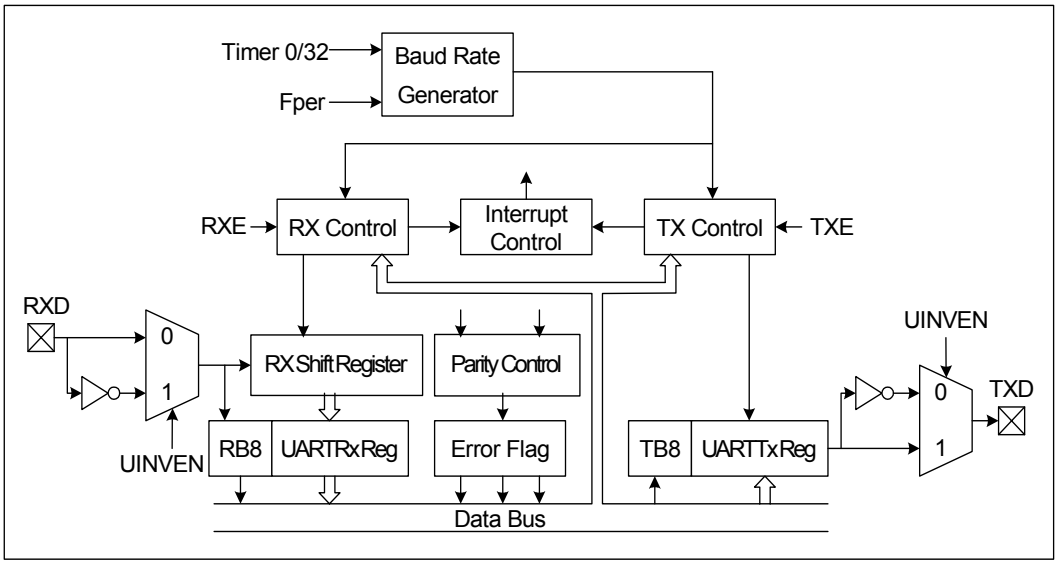


Fig. 8-6 UART Function Block Diagram

In Universal Asynchronous Receiver Transmitter (UART), each transmitted or received character is individually synchronized by framing it with a start bit and stop bit.

Full duplex data transfer is possible because the UART has independent transmit and receive sections. Double buffering in both sections enable the UART to be programmed for continuous data transfer.

The figure below shows the general format of one character sent or received. The communication channel is normally held in the marked state (high). Character transmission or reception starts with a transition to the space state (low).

The first bit transmitted or received is the start bit (low). It is followed by the data bits, in which the least significant bit (LSB) comes first. The data bits are followed by the parity bit. If present, then the stop bit or bits (high) confirm the end of the frame.

In receiving, the UART synchronizes on the falling edge of the start bit. When two or more “0”s are detected during three sampling, it is recognized as normal start bit and receiving operation is started.

8.6.1 Data Format in UART

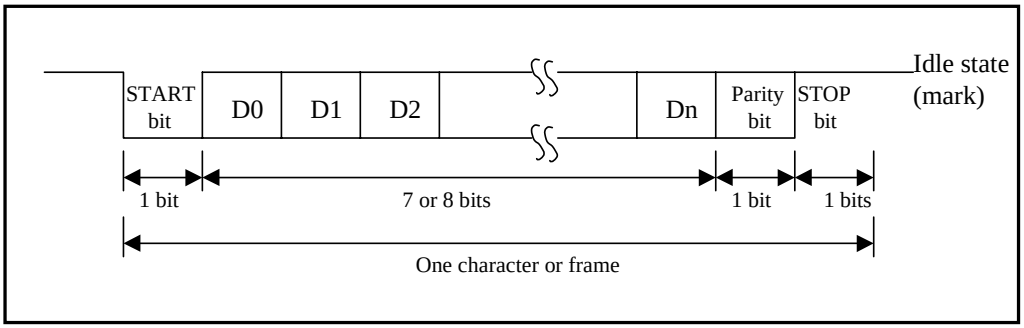


Fig. 8-7 UART Data Format Diagram

8.6.2 UART Modes

There are three modes in UART. Mode 1 (7 bits data) and Mode 2 (8 bits data) allow the addition of a parity bit. The parity bit addition is not available in Mode 3. The Figure below shows the data format in each mode.

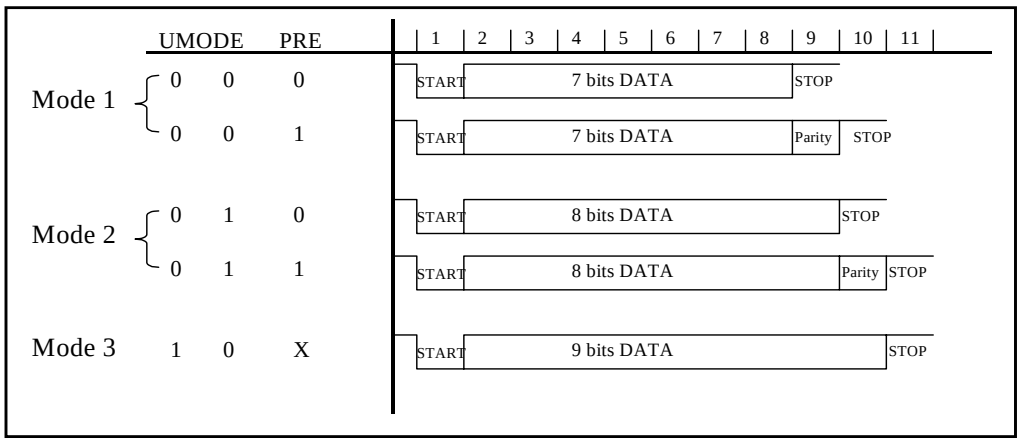


Fig. 8-8 Data Format in UART Modes

8.6.3 UART Transmit Data Function

In transmitting serial data, the UART operates as follows**:**

1. Sets the **TXE** bit of UARTCON register to enable UART transmission function.
2. Writes data into UARTTX register and the **UTBE** bit of UARTCON register is cleared by hardware to allow start of data transmission.
3. Serial transmit data are transmitted in the following order from the TXD pin.
   a) Start bit**:** one “0” bit is output
   b) Transmit data**:** 7, 8, or 9 bits data are output from LSB to MSB
   c) Parity bit**:** one parity bit (odd or even selectable) is output
   d) Stop bit**:** one “1” bit (stop bit) is output
   e) Mark state**:** output “1” continues until the start bit of the next transmit data
4. After transmitting the stop bit, the UART generates a **UTXI** interrupt (if enabled)

8.6.4 UART Receive Data Function

1. Sets the **RXE** bit of the UARTCON register to enable the UART receiving function.
2. The UART monitors the RXD pin and synchronizes internally when it detects a start bit.
3. Received data is shifted into the UARTRX register in LSB to MSB sequence.
4. The parity bit and the stop bit are received. After one character is received, the UART generates a **URXI** interrupt (if enabled). And the **URBF** bit of the UARTSTA register is set to 1.
5. The UART makes the following checks:
   a) Parity check**:** The number of "1" in the received data must match with the even or odd parity setting of the **EVEN** bit in the UARTSTA register.
   b) Frame check**:** The start bit must be "0" and the stop bit must be "1."
   c) Overrun check: the **URBF** bit of the UARTCON register must be cleared (i.e., the UARTRX register should be read out) before the next received data are loaded into the UARTRX register.

   If any checks failed, the UERRI interrupt will be generated (if enabled). And the error flag is indicated in **PRERR, OVERR** or **FMERR** bit. The error flag should be cleared by software, otherwise, a UERRI interrupt will occur when the next byte is received.
6. Read the received data from the UARTRX register. The **URBF** bit will be cleared by hardware.

8.6.5 UART Baud Rate Generator

- The baud rate generator comprises of a circuit that generates a clock pulse which determines the transfer speed for transmitting/receiving data in the UART.
- The input clock of the baud rate generator is derived from the system clock divided by 64 or from Timer 0 divided by 32.
- The system clock should be at 9.830 MHz (PFS=150) or 14.745 MHz (PFS=225) when UART is enabled.
- The **BRATE2 ~ BRATE0** bits of the UARTCON register can determine the desired baud rate.

8.6.6 UART Registers

- UARTCON (R30h): UART Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TB8	UMODE1	UMODE0	BRATE2	BRATE1	BRATE0	UTBE	TXE

Bit 0 (TXE): Enable transmission

Bit 1 (UTBE): UART transfer buffer empty flag. Set to 1 when transfer buffer is empty. Reset to 0 automatically when writing into the UARTTX register.

NOTE

When transmit data function is enabled, the UTBE (read-only) bit will be cleared by hardware. Hence, writing to the UARTTX register is required when user wants to start transmitting data.

Bit 4 ~ Bit 2 (BRATE 2 ~ 0): Baud Rate Selector

SELBR3 (for Code Option)		0: Fper = Fpll		1: Fper = Fpll x 2/3
BRATE2 ~ 0	Fper (PFS = 4 ~ 255)	Fper = 9.83MHz (PFS = 150)	Fper = 14.745MHz (PFS = 225)	Fper = 14.745MHz (PFS = 225)
000	Timer 0/32	Timer 0/32	Timer 0/32	Timer 0/32
001	Fper/4096 baud	2400 baud	3600 baud	2400 baud
010	Fper/2048 baud	4800 baud	7200 baud	4800 baud
011	Fper/1024 baud	9600 baud	14400 baud	9600 baud
100	Fper/512 baud	19200 baud	28800 baud	19200 baud
101	Fper/256 baud	38400 baud	57600 baud	38400 baud
110	Fper/128 baud	76800 baud	115200 baud	76800 baud
111	Fper/64 baud	153600 baud	230400 baud	153600 baud

Bit 6 ~ Bit 5 (UMODE 1 ~ 0): UART Mode

UMODE 1: UMODE 0	UART Mode
00	Mode 1: 7-bit data
01	Mode 2: 8-bit data
10	Mode 3: 9-bit data
11	Reserved

Bit 7 (TB8): Bit 8 Transmission Data

- UARTSTA (R31h): UART STATUS Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RB8	EVEN	PRE	PRERR	OVERR	FMERR	URBF	RXE

Bit 0 (RXE): Enable receive data function

Bit 1 (URBF): UART read buffer full flag. Set to 1 when one character is received. Reset to 0 automatically when read from the UARTRX register.

NOTE

When receive data function is enabled, URBF (read-only) bit will be cleared by hardware. Hence, reading from the UARTRX register is required to avoid overrun error.

Bit 2 (FMERR): Framing error flag. Set to **1** when framing error occurs
Clear to **0** by software

Bit 3 (OVERR): Overrun error flag. Set to **1** when overrun error occurs
Clear to **0** by software

Bit 4 (PRERR): Parity error flag. Set to **1** when parity error occurs
Clear to **0** by software

Bit 5 (PRE): Enable parity addition
"**0**" : Disable
"**1**" : Enable

Bit 6 (EVEN): Select parity check
"**0**" : Odd parity
"**1**" : Even parity

Bit 7 (RB8): Receiving Data Bit 8

- UARTTX (R15h): UART Transfer Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0

Bit 7 ~ Bit 0 (TB7 ~ TB0): Transmission data register. UARTTX register is write-only.

- UARTRX (R16h): UART Receiver Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

Bit 7 ~ Bit 0 (RB7 ~ RB0): Receiving data register. UARTRX register is read-only.

- STBCON (R21h): Strobe Output Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UINVEN	/REN	BitST	ALL	STB3	STB2	STB1	STB0

Bit 7 (UINVEN): Enable UART TXD and RXD port inverse output.

"**0**" : disable TXD and RXD port inverse output
"**1**" : enable TXD and RXD port inverse output

- CPUCON (R0Eh): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0

Bit 2 (GLINT): Global interrupt control bit

"**0**": disable all interrupt
"**1**": enable all un-masked interrupt

- INTCON (R22h): Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 3 (UERRIE): UART receiving error interrupt control bit

"**0**": disable
"**1**": enable

Bit 4 (UTXIE): UART Transfer buffer empty interrupt control bit

"**0**": disable
"**1**": enable

Bit 5 (URXIE): UART Receiver buffer full interrupt control bit

"**0**": disable
"**1**": enable

- INTSTA (R23h): Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 3 (UERRI): Set to **1** when UART receiving error occurs
Clear to **0** by software or disable UART

Bit 4 (UTXI): Set to **1** when UART transfer buffer empty occurs
Clear to **0** by software or disable UARTTX (TXE=0)

Bit 5 (URXI): Set to **1** when UART receiver buffer full occurs
Clear to **0** by software or disable UARTRX (RXE=0)

Transmission Counter Timing:

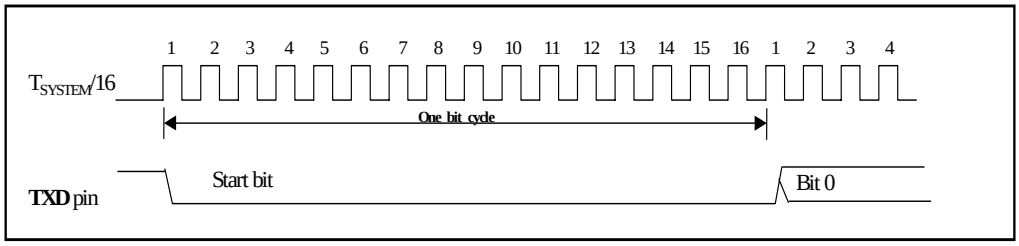


Fig. 8-9 UART Transmit Counter Timing Diagram

UART Transmit Operation (8 bits data with parity bit):

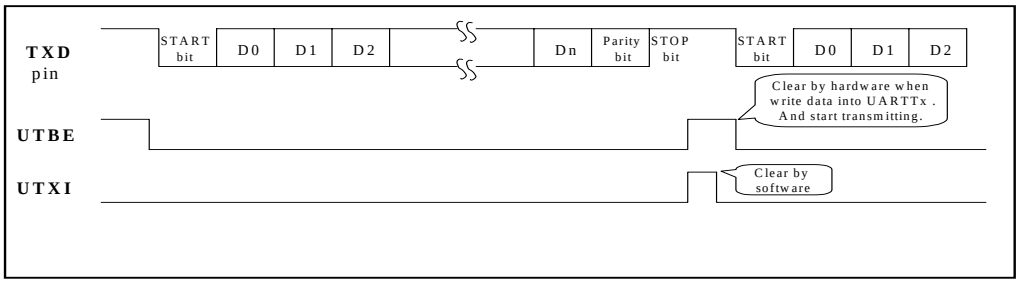


Fig. 8-10 UART Transmit Operation

Code Example:

```
; === UART Transfer Buffer Empty Interrupt
PERIPH:
          PUSH
          JBC       INTSTA,UTXI,Q_UTXINT
          BC        INTSTA,UTXI
          MOV       A,UTX_NO
          COMA      ACC
          MOV       UTX_NO,A
          MOV       UARTTX,A              ;Tx data 55,AA,55,AA
Q_UTXINT: POP
          RETI
; === UART 38400 baud 8-bit inverse
UTX_SR:
           :
          System setting 9.83MHz
           :
          BS        STBCON,UINVEN         ; TXD & RXD inverse
          MOV       A,#00110101B          ; Enable Tx
          MOV       UARTCON,A             ; 8bit, 38400baud
          MOV       A,#01100000B          ; Disable Rx
          MOV       UARTSTA,A             ; Even Parity
          BC        INTSTA,UTXI           ; TX buffer empty occurs
          BS        INTCON,UTXIE          ; En. TX interrupt
          BS        CPUCON,GLINT          ; Global interrupt
          MOV       A,#0X55
          MOV       UTX_NO,A
          MOV       UARTTX,A              ; Tx data 55
TX_loop:
          SJMP      TX_loop
```

Receiving Counter Timing:

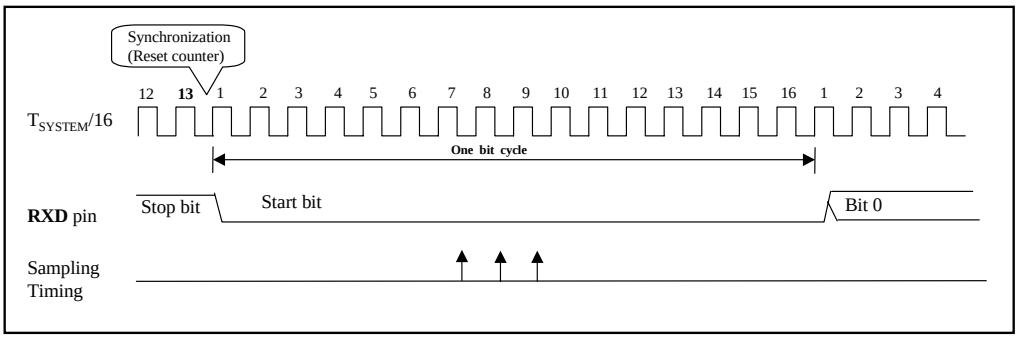


Fig. 8-11 UART Receive Counter Timing Diagram

UART Receive Operation (8 bits data with parity and stop bit):

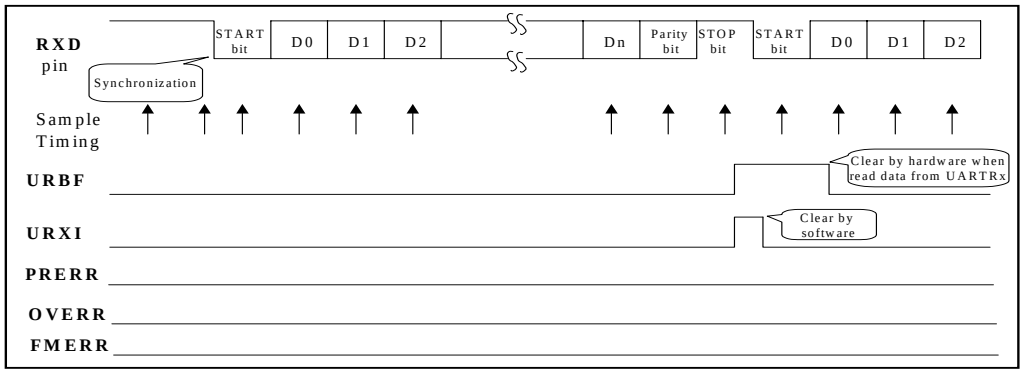


Fig. 8-12 UART Receive Operation

Code Example:

```
;===UART Receiver buffer full interrupt
PERIPH:
     PUSH
     JBC      INTSTA,URXI,UERRINT
     BC       INTSTA,URXI
     MOVPR    URX_NO,UARTRX
     SJMP     Q_RXINT
;
;===UART error interrupt
UERRINT:
     JBC      INTSTA,UERRI,Q_RXINT
     BC       INTSTA,UERRI
;---Framing error flag
;---Over run error flag
;---Parity error flag
     MOV      A,UARTSTA
     AND      A,#00011100B
     MOV      PORTH,A
     BC       UARTSTA,FMERR
     BC       UARTSTA,OVERR
     BC       UARTSTA,PRERR
Q_RXINT:
     POP
     RETI
```

```
;===UART 38400 baud 8bit inverse
URX_SR:
      :
   System setting 9.83MHz
   Port H & G setting output port
     :
;---TXD & RXD inverse
   BS       STBCON,UINVEN
;---Disable Tx, 8bit, 38400baud
   MOV      A,#00110100B
   MOV      UARTCON,A
;---Enable Rx, Even Parity
   MOV      A,#01100001B
   MOV      UARTSTA,A
;---UART RX buffer empty interrupt
   BS       INTSTA,URXI
   BS       INTCON,URXIE
;---UART RX error interrupt
   BS       INTSTA,UERRI
   BS       INTCON,UERRIE
;---Global interrupt
   BS       CPUCON,GLINT
RX_loop:
   MOVRP    PORTG,URX_NO
   SJMP     RX_loop
```

8.7 A/D Converter

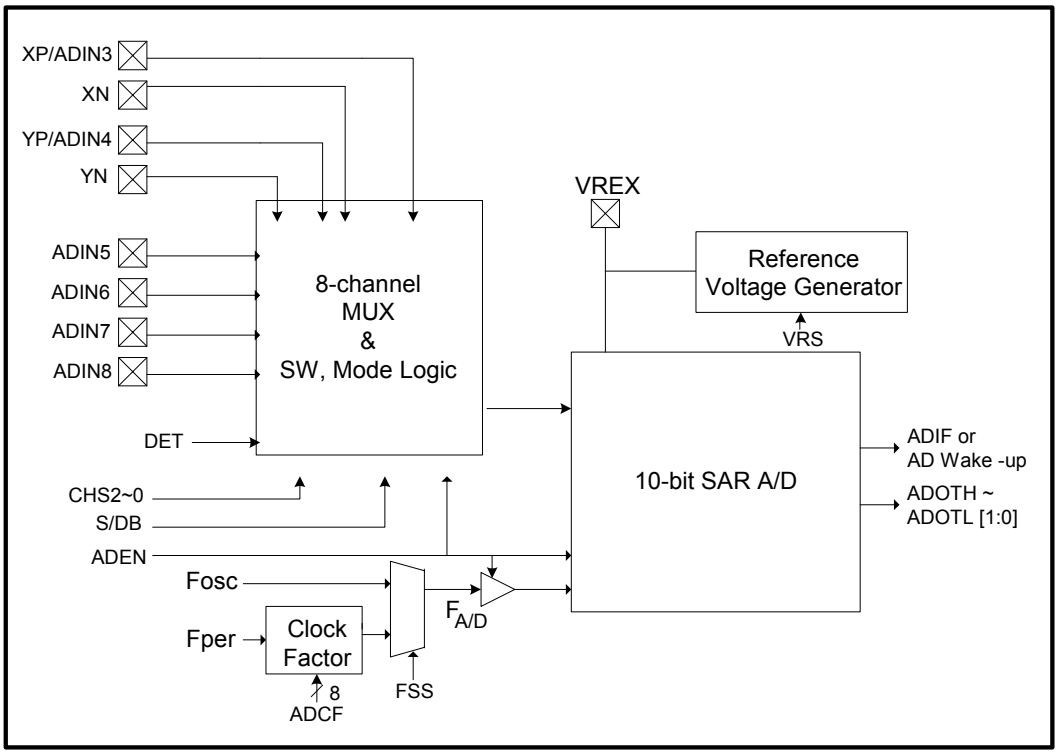


Fig. 8-8 A/D Converter Function Block Diagram

This A/D converter is an 8-channel 10 bit resolution. When the MCU is in Slow or Fast mode and ADEN=1, the A/D conversion will run immediately. The two channels; XP and YP with low on-resistance switches, are used for driving touch screen application. The remaining 6 channels are for general application.

The A/D operation for touch panel application is described as follows:

Step 1: Pen down detection

If the panel is not touched, the PIRQB is high. When the touch panel is pressed, the PIRQB is low and PIRQB interrupt occur (if INT is enabled).

Step 2: Measure the X position

If the PIRQB is low for quite some time, clear the DET bit, the PIRQB returns to high. Then measure the X position.

Step 3: Measure the Y position

Measure the Y position immediately after Step 2.

Step 4: Back to Step 1

VREX: Reference voltage input pin when VRS=1; Reference voltage output pin when VRS=0

XN (Port C.7): X negative position input

YN (Port C.6): Y negative position input

XP/ADIN3 (Port C.5): X positive position input or A/D input Channel 3

YP/ADIN4 (Port C.4): Y positive position input or A/D input Channel 4

ADIN5 (Port C.3): A/D input Channel 5

ADIN6 (Port C.2): A/D input Channel 6

ADIN7 (Port C.1): A/D input Channel 7

ADIN8 (Port C.0): A/D input Channel 8

8.7.1 A/D Converter Registers

- ADCON (R2Ch): A/D Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DET	VRS	ADEN	PIRQB	S/DB	CHS2	CHS1	CHS0

Bit 2 ~ Bit 0 (CHS2 ~ CHS0): 2-channel touch screen & 4-channel A/D input selection.

Bit 3 (S/DB): Reference mode control bit
"**0**" : Differential reference mode
"**1**" : Single-ended reference mode

Bit 4 (PIQRB): Touch screen status bit. It is a read bit
"**0**" : Touch screen is tapped
"**1**" : Touch screen is not tapped

Bit 5 (ADEN): A/D enable control bit. Automatically clears to "0" when ADIF occurs.
"**0**" : A/D disable
"**1**" : A/D enable

Bit 6 (VRS): A/D input reference voltage selection and enable/disable internal reference generator bit
"**0**" : Enable the internal reference generator and the reference voltage is from the internal reference voltage generator
"**1**" : Disable the internal reference generator and the reference voltage is from the external VREX pin

Bit 7 (DET): Touch panel pen down detection mode control bit. Enables/disables PIRQB interrupt and wake-up functions
"**0**" : Disable the detection mode. S switches are off for interrupts and wake-up functions
"**1**" : Enable the detection mode. S switches are on for interrupts and wake-up functions

ADEN	DET	CHS [2:0]	Vin	VRS	Mode
0	0	-	-	1	Standby mode
0	1	-	-	1	Pen-down detection
1	0	000	YP	1	Measure X position. (Touch panel)
1	0	001	XP	1	Measure Y position. (Touch panel)
1	0	010	ADIN3	0/1	Measure ADIN3
1	0	011	ADIN4	0/1	Measure ADIN4
1	0	100	ADIN5	0/1	Measure ADIN5
1	0	101	ADIN6	0/1	Measure ADIN6
1	0	110	ADIN7	0/1	Measure ADIN7
1	0	111	ADIN8	0/1	Measure ADIN8

- ADOTH (R14h): A/D Output High Byte Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADOT9	ADOT8	ADOT7	ADOT6	ADOT5	ADOT4	ADOT3	ADOT2

- ADOTL (R13h): A/D Output Low Byte Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTEN	EXTMEM	ADWKEN	-	-	FSS	ADOT1	ADOT0

Bit 7H ~ Bit 0H ~ Bit 1L ~ Bit 0L (ADOT9 ~ ADOT0): 10-bit resolution A/D output data.

Bit 2 (FSS): A/D clock source select bit

"**0**" : A/D clock source is from Fosc

"**1**" : A/D clock source is from Fper/2

NOTE

When MCU is in FAST mode, the A/D clock source must be from PLL (FSS = 1, PEN = 1). Sourcing A/D clock from Oscillator (FSS = 0, PEN = 0) is prohibited.

Bit 5 (ADWKEN): A/D wake up control bit

"**0**" : Disable A/D wake-up function

"**1**" : Enable A/D wake-up function

- ADCF (R4Eh): A/D Clock Factor Register

The ADCF is used as a clock factor, such as**:**

$$F_{A/D} = \frac{F_{Per}}{2(ADCF + 1)}$$

A/D Throughput rate = $F_{A/D}$/12

	Fper=2.03M (PFS=31)	Fper=3.99M (PFS=61)	Fper=7.99M (PFS=122)	Fper=9.83M (PFS=150)	Fper=11.99M (PFS=183)	Fper=14.02M (PFS=214)	Fper=16.7M (PFS=255)
ADCF=3	FA/D=254k	FA/D=499k	FA/D=999k	FA/D=1229k	FA/D=1499k	FA/D=1753k	FA/D=2089k
ADCF=7	FA/D=127k	FA/D=250k	FA/D=499k	FA/D=614k	FA/D=749k	FA/D=876k	FA/D=1044k
ADCF=15	FA/D=63k	FA/D=125k	FA/D=250k	FA/D=307k	FA/D=374k	FA/D=438k	FA/D=522k
ADCF=31	FA/D=31k	FA/D=62k	FA/D=125k	FA/D=154k	FA/D=187k	FA/D=219k	FA/D=261k
ADCF=63	FA/D=15k	FA/D=31k	FA/D=62k	FA/D=77k	FA/D=93k	FA/D=109k	FA/D=130k
ADCF=95	FA/D=11k	FA/D=21k	FA/D=42k	FA/D=60k	FA/D=73k	FA/D=86k	FA/D=102k
ADCF=127	FA/D=10k	FA/D=21k	FA/D=31k	FA/D=51k	FA/D=62k	FA/D=73k	FA/D=87k
ADCF=159	FA/D=6k	FA/D=12k	FA/D=25k	FA/D=31k	FA/D=37k	FA/D=44k	FA/D=52k
ADCF=191	FA/D=5k	FA/D=10k	FA/D=21k	FA/D=25k	FA/D=31k	FA/D=37k	FA/D=44k
ADCF=223	FA/D=4.5k	FA/D=8.9k	FA/D=17.8k	FA/D=21.9k	FA/D=26.8k	FA/D=31.3k	FA/D=37.3k
ADCF=255	FA/D=3.9k	FA/D=7.8k	FA/D=15.6k	FA/D=19.2k	FA/D=23.4k	FA/D=27.3k	FA/D=32.6k

Note: FA/D value greater than 1.4 MHz is invalid.

- CPUCON (R0Eh): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0

Bit 0 (MS0): CPU Fast/Slow mode setting
"**0**" : Slow mode
"**1**" : Fast mode

Bit 1 (MS1): CPU Sleep & Idle mode setting
"**0**" : Sleep mode
"**1**" : Idle mode

Bit 2 (GLINT): Global interrupt control bit
"**0**" : Disable all interrupts
"**1**" : Enable all un-mask interrupts

Bit 7 (PEN): PLL enable (only effective when the MCU is in Idle or Slow mode)
"**0**" : Disable PLL
"**1**" : Enable PLL

- INTCON (R22h): Interrupt Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIE	ADIE	URXIE	UTXIE	UERRIE	TMR2IE	TMR1IE	TMR0IE

Bit 6 (ADIE): A/D interrupt control bit
"**0**" : Disable
"**1**" : Enable

- INTSTA (R23h): Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPIF	ADIF	URXI	UTXI	UERRI	TMR2I	TMR1I	TMR0I

Bit 6 (ADIF): Set to **1** when A/D output data is ready to be read
Clear to "**0**" by software or disable A/D

Timing Diagram of General A/D Converter Application

CHS [2:0] = 010 ~ 111

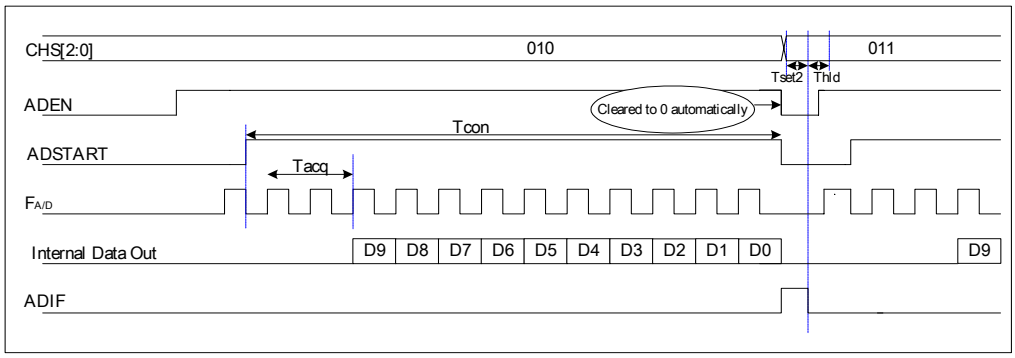


Fig. 8-14 A/D Converter General Application Timing Diagram

Correlation between A/D Converter and MCU Mode

MCU in Slow or Fast Mode

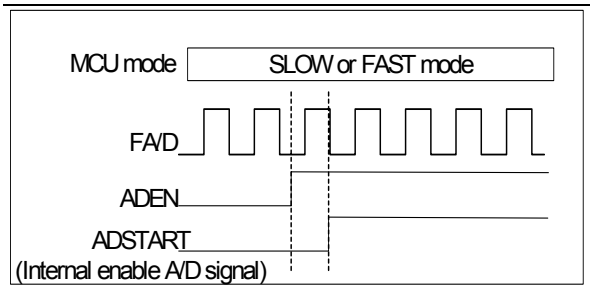


Fig. 8-15 A/D Converter vs. MCU Mode

Code Example:

```
; === A/D interrupt
PERIPH:
   PUSH
   JBC        INTSTA,ADIF,Q_ADINT
   BC         INTSTA,ADIF
   BS         INTFLAG,F_IAD
Q_ADINT:
   POP
   RETI
; === Fpll=8MHz & ADCF=7 => FA/D=499kHz
AD_SR:
     :
  System setting 8MHz
  Port H & G setting output port
     :
; --- PLL enable
   BS         CPUCON,PEN
; --- Clock source is PLL
   BS         ADOTL,FSS
; --- FA/D=499kHz
   MOV        A,#7
   MOV        ADCF,A
; --- VRIN, Differential, ADIN3
   MOV        A,#00000010B
   MOV        ADCON,A
; --- AD interrupt enable
   BS         INTCON,ADIE
   BC         INTSTA,ADIF
   BS         CPUCON,GLINT
```

```
; === Fast mode: MCU in fast mode
    BS      CPUCON,MS1
; --- Repeat detect A/D 3 times
    MOV     A,#3
AD3times:
; --- AD enable
    BS      ADCON,ADEN
Chk_AD:
    JBC     INTFLAG,F_IAD,Chk_AD
    BC      INTFLAG,F_IAD
    JDNZ    ACC,AD3times
```

```
; === Slow mode: MCU in slow mode
    BC      CPUCON,MS0
; --- Repeat detect A/D 3 times
    MOV     A,#3
AD3times:
;---AD enable
    BS      ADCON,ADEN
Chk_AD:
    JBC     INTFLAG,F_IAD,Chk_AD
    BC      INTFLAG,F_IAD
    JDNZ    ACC,AD3times
```

```
; --- Out AD to Port H : G
    MOVRP   PORTH,ADOTH
    MOV     A,ADOTL
    AND     A,#00000011B
    MOV     PORTG,A
     :
```

8.7.4 A/D Converter Flowchart

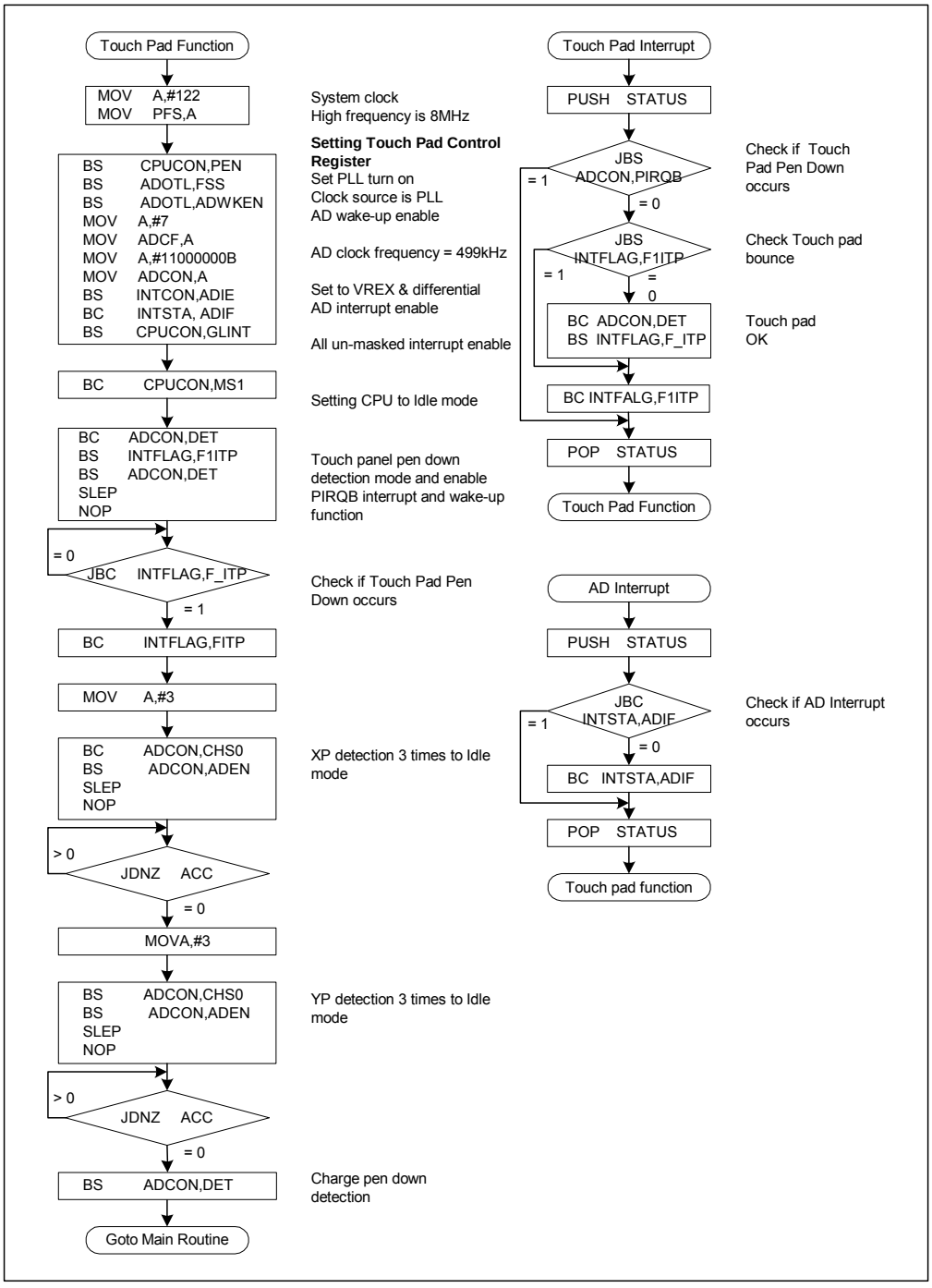


Fig. 8-16 A/D Converter Flow Chart

Code Example

```
;; *** Touch panel Interrupt
INPTINT:
       PUSH
       JBS      ADCON,PIRQB,Q_TPINT      ; Touch screen status bit
       JBS      INTFLAG,F1ITP,TPINT1
       BC       ADCON,DET                ; Pen down detection disable
       BS       INTFLAG,F_ITP            ; Pen down ok flag

TPINT1:
       BC       INTFLAG,F1ITP            ; Pen down detect 2 times
Q_TPINT:
       POP
       RETI
; === A/D interrupt
PERIPH:
       PUSH
       JBC      INTSTA,ADIF,Q_ADINT
       BC       INTSTA,ADIF
Q_ADINT:
       POP
       RETI
; === Touch panel routine
TP_SR:
        :
       System setting 8MHz
       Port H & G setting output port
        :
       BS       CPUCON,PEN               ; PLL enable
       BS       ADOTL,FSS                ; Clock source is PLL
       BS       ADOTL,ADWKEN             ; AD wake-up
       MOV      A,#7
       MOV      ADCF,A                   ; FA/D=499kHz
       MOV      A,#11000000B
       MOV      ADCON,A                  ; VREX, Differential
       BS       INTCON,ADIE              ; AD interrupt enable
       BC       INTSTA,ADIF
       BS       CPUCON,GLINT
TPILoop:
       BC       ADCON,DET                ; Pen down detection disable
       BS       INTFLAG,F1ITP
       BS       ADCON,DET                ; Pen down detection enable
TPILp1:
       JBS      INTFLAG,F_ITP,TPILp1
; --- Repeat YP detect A/D 3 times
       MOV      A,#3
YP3times:
       BS       ADCON,CHS0               ; YP detection
       BS       ADCON,ADEN               ; AD enable
WaitYAD:
       JBS      ADCON,ADEN,WaitYAD
       JDNZ     ACC,YP3times
       MOVRP    PORTG,ADOTH
; --- Repeat XP detect A/D 3 times
       MOV      A,#3
XP3times:
       BC       ADCON,CHS0               ; XP detection
       BS       ADCON,ADEN               ; AD enable
WaitXAD:
       JBS      ADCON,ADEN,WaitXAD
       JDNZ     ACC,XP3times
       MOVRP    PORTH,ADOTH
       BS       ADCON,DET
        :
       SJMP     TPILoop
```

8.8 Input/Output Key

The 8-pin key input (Port A) and 16-pin key strobe (Port J and Port K) can achieve a maximum 128-key matrix. The 16 to 1 key strobe multiplex pins are controlled by **STB3 ~ STB0** bits of STBCON register. Interrupt is enabled when Port A falling edge is detected. Similarly, Wake-up is enabled when a key input falling edge is detected.

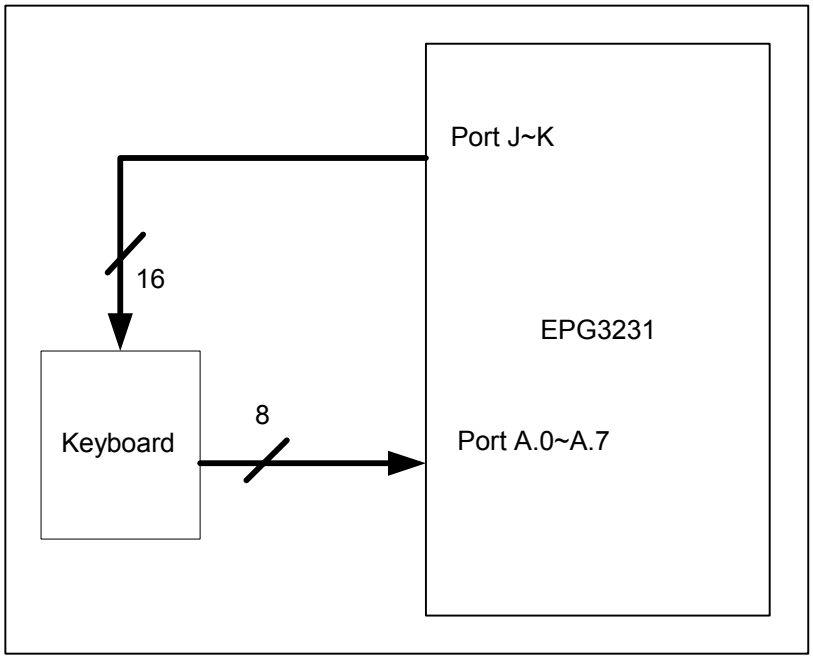


Fig. 8-17 Keyboard Function Block Diagram

Port A.0~Port A.7 are input ports with controllable pull-high resistor.

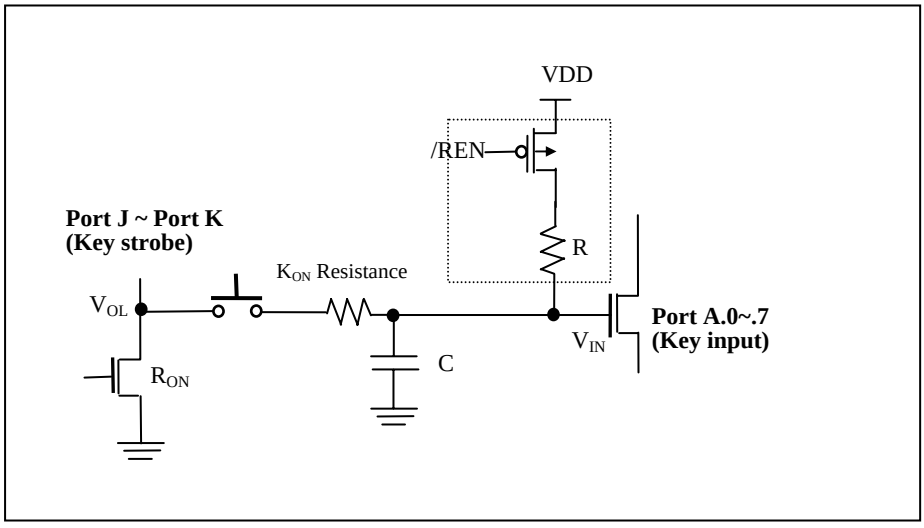


Fig. 8-18 Key Circuit Diagram

Port J ~ Port K are designed for 16 bits key strobe or general I/O. When BitST of STBCON register is set, Port J and Port K become key strobe pins. 16 to 1 multiplexing of Port J and Port K are selected by STB3~STB0 bits of STBCON register.

Only selected pins will output low but the other 15 pins will remain at a high level. When ALL bits of STBCON register is set, all strobe pins will output low.

The detailed function is summarized in the following table:

STBCON			Key Strobe (Port J & Port K)															
BitS	ALL	STB3~	PJ.0	PJ.1	PJ.2	PJ.3	PJ.4	PJ.5	PJ.6	PJ.7	PK.0	PK.1	PK.2	PK.3	PK.4	PK.5	PK.6	PK.7
0	x	x	General I/O Port															
1	0	0000	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0001	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		0010	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
		0011	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
		0100	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
		0101	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
		0110	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
		0111	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
		1000	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
		1001	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
		1010	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
		1011	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
		1100	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
		1101	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
		1110	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
		1111	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	1	xxxx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.8.1 Input/Output Key Registers

- Port A (R17h): Port A Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 ~ Bit 0: Key input. Input falling edge interrupt or wake-up pin.

- PAINTEN (R2Dh): Port A Interrupt Enable Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7IE	PA6IE	PA5IE	PA4IE	PA3IE	PA2IE	PA1IE	PA0IE

Bit 7 ~ Bit 0 (PA7IE ~ PA0IE): Interrupt Control Bit
"**0**": disable
"**1**": enable

- PAINTSTA (R2Eh): Port A Interrupt Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7I	PA6I	PA5I	PA4I	PA3I	PA2I	PA1I	PA0I

Bit 7 ~ Bit 0 (PA7I ~ PA0I): Port A interrupt INT status. Set to **1** when pin falling edge is detected. Clear to **0** by software.

- PAWAKE (R2Fh): Port A Wake-up Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WKEN7	WKEN6	WKEN5	WKEN4	WKEN3	WKEN2	WKEN1	WKEN0

Bit 7 ~ Bit 0 (WKEN7 ~ WKEN0): Port A wake-up function control bit

"**0** ": disable wake-up function
"**1**" : enable wake-up function

- STBCON (R21h): Strobe Output Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UINVEN	/REN	BitST	ALL	STB3	STB2	STB1	STB0

Bit 3 ~ Bit 0 (STB0 ~ 3): Strobe output select bit

Bit 4 (All): Set All strobe

Bit 5 (Bit ST): Enable Bit strobe

"**0**" : Port J and Port K are general I/O ports
"**1**" : Port J and Port K are key strobe pins. Strobe signal as defined by STB3~0.

Bit 6 (/REN): Port A.0~Port A.7 Pull-high resistor control bit

"**0**" : enable pull-high resistor
"**1**" : disable pull-high resistor

- Port J, Port K (R32h, R33h): General I/O Registers
- DCRJK (R39h): Direction control & Pull-high resistor control of Port J & Port K

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KHNPU	KLNPU	KHNDC	KLNDC	JHNPU	JLNPU	JHNDC	JLNDC

Bit 1, Bit 5 (JHNDC, KHNDC) & Bit 0, Bit 4 (JLNDC, KLNDC): Ports J & K high / low nibbles direction control.

"**0**": set as output pin
"**1**": set as input pin

Bit 3, Bit 7 (JHNPU, KHNPU) & Bit 2, Bit 6 (JLNPU, KLNPU): Enable Ports J & K high / low nibbles pull-high resistor

"**0**" : disable pull-high resistor
"**1**" : enable pull-high resistor

Code Example:

```
; Key matrix 1 (Port A and Ground):
; *** Interrupt Port A data
INPTINT:
       PUSH
       MOVRP    PORTH,PAINTSTA
       CLR      PAINTSTA
       POP
       RETI
; === Sleep mode
PAIN_SR:
         :
; --- PortA wakeup
       MOV      A,#11111111B
       MOV      PAWAKE,A
         :
         :
; --- /REN Pull-up enable
       BC       STBCON,REN
; --- Port A interrupt enable.
       MOV      A,#11111111B
       MOV      PAINTEN,A
       CLR      PAINTSTA
       BS       CPUCON,GLINT
; --- Sleep MODE
       BC       CPUCON,MS1
PAINloop:
       SLEP
       NOP
         :
       SJMP     PAINloop
```

```
; Key matrix 2 (Port A and Port J - Port K):
; *** Key scan function
       CLR      DCRJK                   ; Set Ports J & K as output
       MOV      A,#0XFF
       MOV      PAWAKE,A                ; Set Port A wakeup function
       BC       STBCON,REN              ; Port A pull-high enable
       BS       STBCON,ALL              ; All strobe enable
       BC       CPUCON,MS1              ; Sleep mode
KeySleep:
       SLEP
       NOP
       MOV      A,Port A                ; Port A input data
       JE       A,#0XFF,KeySleep
       CLR      STBCON
       BS       STBCON,BitST            ; Bit strobe enable
       MOV      A,#0FEH
KeyLoop:
       JGE      A,PortA,KeyScan         ; If A >= Port A Goto KeyScan
       INC      STBCON
       SJMP     KeyLoop
KeyScan:
       SWAPA    STBCON
       AND      A,#11110000B
       MOV      Key_No,A                ; Key_No:XXXX 0000
       MOV      A,PortA
       BC       STBCON,BitST            ; Bit strobe disable
; --- Check key number
ChKeyNo:
       RRC      ACC
       JBC      STATUS,F_C,KeyScanOk
       INC      Key_No
       SJMP     ChKeyNo
; --- Key Scan is finished
KeyScanOk:
       SAWP     Key_No                  ; Key_No:0XXX XXXX
         :
```

8.9 Serial Peripheral Interface (SPI)

Operation in either Master mode or Slave mode

Three-wire or Four-wire full duplex synchronous communication

Programmable Shift Register Length (24/16/8 bits)

Programmable bit rates of communication

Programmable clock polarity

Programmable shift direction

Programmable sample phase

Interrupt flag available for the read buffer full

Up to 4MHz (system clock at 16MHz) bit frequency

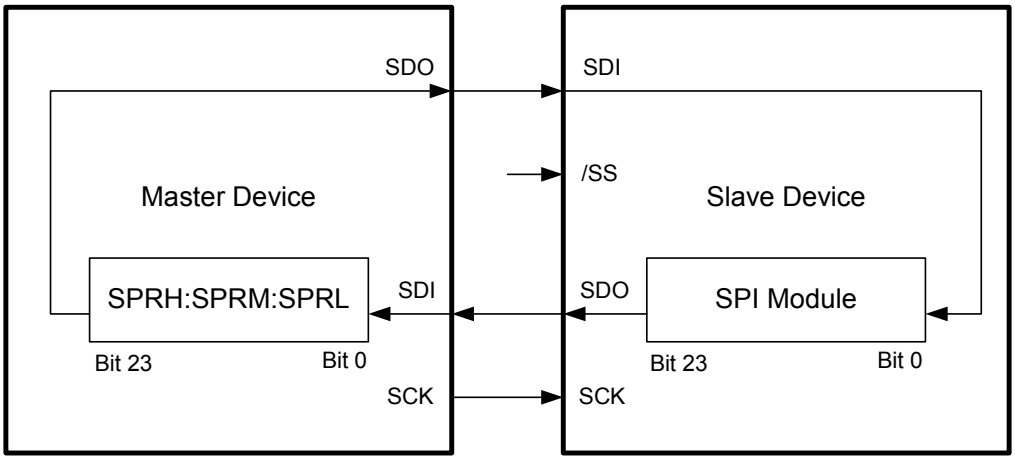


Fig. 8-19a Single SPI Master/Slave Communication

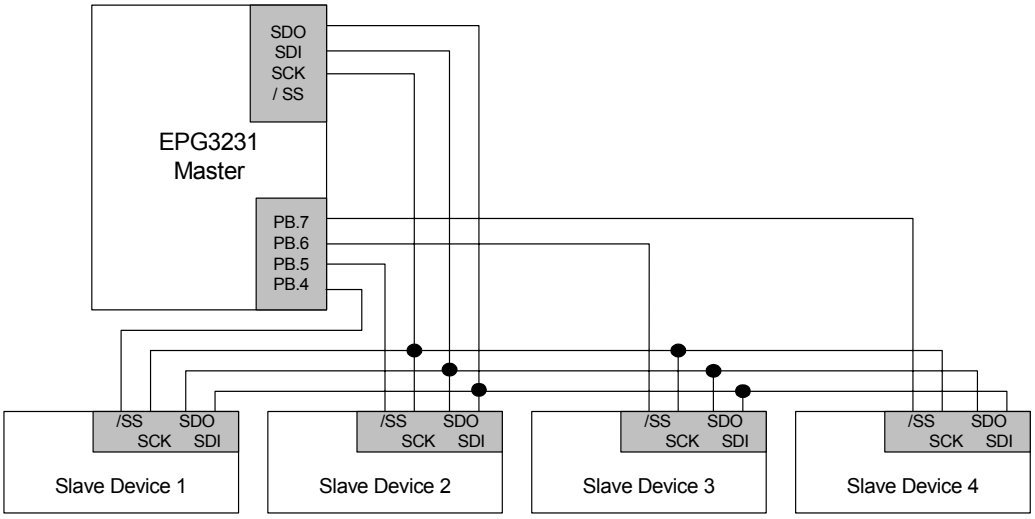


Fig. 8-19b SPI Configuration Example of Single-Master and Multi-Slaves

The MCU communicates with other devices through SPI module. If the MCU is defined as the master controller, it sends clock through the SCK pin. An 8-bit data is transmitted and received at the same time. If the MCU, however, is defined as a slave, its SCK pin could be programmed as an input pin. Data will continue to be shifted at selected clock rate and selected edge.

- Setup the **TLS1 ~ TLS0** bits of SPICON register to select the shift register length of SPI and enable/disable SPI function.
- Setup the **BRS2 ~ BRS0** bit of SPICON register to select the SPI mode (master/slave) and Bit Rate. When in Master mode, the clock source can be selected from system clock or half of timer 0 interval. When in Slave mode, the **/SS** pin can be enabled or disabled.
- Setup the **DORD** bit of SPICON register to determine the shift direction.
- Setup the **EDS** bit of SPICON register to select the raising edge or falling edge latch of the data.
- Setup the **SMP** bit of SPISTA register to select the sample phase at the middle or the end of the data output time.

✧ Master Mode

In Master mode, the SCK pin functions as clock output pin.

If a 24-bit shift register length is selected, SPRH, SPRM, and SPRL registers are the high, middle, and low bytes of the shift register respectively. Likewise, if 8-bit shift register length is selected; SPRL register becomes the content of the shift register.

When writing data to SPRH, SPRM, and SPRL registers, it is only after writing data into SPRL register that the SE bit of SPICON register is set by hardware automatically and starts shifting. After the shift buffer becomes empty, **SE** bit is cleared by hardware and clock output is stopped from the **SCK** pin.

The receiver is active during SPI transfer. When the receive buffer is full, **RBF** flag is set and interrupt occurs (if enabled). During a read out of the shift register contents, it is only after the SPRL register has been read out that the hardware will automatically clear the **RBF** flag. If SPRL register has not been read out, RBF bit will remain set and data collision will take place at the next clock input.

✧ Slave Mode

In slave mode, input clock is from the Master device. **SCK** pin is a clock input pin. The **SE** bit is not used to control the starting shift in this mode. But it is a Transfer buffer empty status bit.

The same as with the Master Mode, you can select the shift register length. Transfer data are written to SPRH, SPRM, SPRL registers. After writing data into the SPRL register, **SE** bit of SPICON register will be set by hardware. But the start shifting is controlled by the MASTER device clock input.

While the shift buffer is empty the **SE** bit will be cleared. At the same time, when the receive buffer is full, **RBF** flag will be set and an interrupt occurs (if enabled). The received data is at SPRH, SPRM and SPRL register; you should read them out before the next clock input. Otherwise, data collision will happen and **DCOL** bit of the SPISTA register will be set.

SDI (I): Serial Data Input. (Receive serially)

SDO (O): Serial Data Output. (Transmit serially, in slave mode, defined as high-impedance, if not selected.)

SCK (I/O): Serial Clock input/output. (When in Master mode, sends clock through the SCK pin. However, if defined as a slave, its **SCK** pin is programmed as an input pin).

/SS* (I):** /Slave Select. (This pin ***/SS will be active when /SS function is enabled. (BRS=110) else ***/SS*** pin is a general purpose I/O.

Master device remains low for the ***/SS*** pin to signify the slave(s) for transmit/receive data. Ignore the data on the **SDI** and **SDO** pins while /SS is high, because the SDO is no longer driven.

SPRH; SPRM; SPRL (R41h; R42h; R43h): SPI shift buffer for 24/16/8 bits length.

The buffer will ignore any write until the shifting is completed. If you select 24 bits shift buffer, SPRH: SPRM: SPRL register is the contents of the 24 bits shift buffer. Else if 8 bits shift buffer is selected, SPRL register is the contents of the shift register.

When writing data into the SPRL register, **SE** bit of the SPICON register will be set by hardware and starts shifting. While the shift buffer is empty, at the same time, receive buffer is full, the received data is shifted in SPRH, SPRM and SPRL registers. After SPRL register has been read out, hardware will clear the **RBF** flag automatically.

- SPICON (R3Fh): SPI Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TLS1	TLS0	BRS2	BRS1	BRS0	EDS	DORD	SE

Bit 0 (SE): Shift enable. Set to "**1**" automatically when writing data into the SPRL register and shifting starts. Reset to "**0**" when a transfer buffer empty is detected.

NOTE

The SE bit is read-only and is cleared by hardware when SPI is enabled. Hence, writing to the SPRL register is necessary when user wants to start shifting the data.

Bit 1 (DORD): Data transmission order
"**0**" : Shift left (MSB first)
"**1**" : Shift right (LSB first)

Bit 2 (EDS): Select the rising / falling edge latch by programming the EDS bit

"**0**" : Falling edge

"**1**" : Rising edge

Bit 5 ~ Bit 3 (BRS2 ~ BRS0): Bit rate select. Programs the clock frequency/rates and sources**:**

000**:** Master, TMR0/2

001**:** Master, (FPLL/2) /4

010**:** Master, (FPLL/2) /16

011**:** Master, (FPLL/2) /64

100**:** Master, (FPLL/2) /256

101**:** Master, (FPLL/2) /1024

110**:** Slave, /SS enable

111**:** Slave, /SS disable

SPI bit rate table

Prescaler		FPLL / 2			
BRS2:0	Bit Rate	16MHz	10MHz	4MHz	32.768kHz
001	(FPLL/2) /4	4000000	2500000	1000000	8196
010	(FPLL/2) /16	1000000	625000	250000	2048
011	(FPLL/2) /64	250000	156250	62500	512
100	(FPLL/2) /256	62500	39063	15625	128
101	(FPLL/2) /1024	15625	9766	3096	32

Bit 7 ~ Bit 6 (TLS1 ~ TLS0): Shift buffer length select. The Shift buffer length is programmable.

00: SPI disable

01: Enable SPI and shift buffer length = 24 bits

10: Enable SPI and shift buffer length = 16 bits

11: Enable SPI and shift buffer length = 8 bits

- SPISTA (R40h): SPI Status Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WEN	-	SRBFIE	SRBFI	SPWKEN	SMP	DCOL	RBF

Bit 0 (RBF): Set to "1" by Buffer Full Detector, and automatically cleared to "0" when data are read from the SPRL register.

NOTE

The RBF bit (read-only) is cleared by hardware when SPI is enabled. Hence, reading the SPRL register is necessary to avoid data collision (DCOL) condition.

Bit 1 (DCOL): SPI Data collision

Bit 2 (SMP): SPI data input sample phase
"**0**" : Input data sampled at the middle of data output time
"**1**" : Input data sampled at the end of data output time

NOTE
In Slave mode, data input sample is fixed at the middle of data output time.

Bit 3 (SPWKEN): SPI wake-up enable control bit
"**0**" : Disable SPI (Slave mode) read buffer full wakeup
"**1**" : Enable SPI (Slave mode) read buffer full wakeup

Bit 4 (SRBFI): Set to "**1**" when an SPI read buffer full occurs. Cleared to "**0**" by software or disable SPI.
"**0**" : Data collision does not occur
"**1**" : Data collision occurs. Should be cleared by software

Bit 5 (SRBFIE): Control bit of SPI read buffer full interrupt
"**0**" : Disable interrupt function
"**1**" : Enable interrupt function

- CPUCON (R0Eh): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0

Bit 2 (GLINT): Global interrupt control bit
"**0**" : Disable all interrupts
"**1**" : Enable all un-mask interrupts

Master Mode: (Shift buffer Length = 24bits)

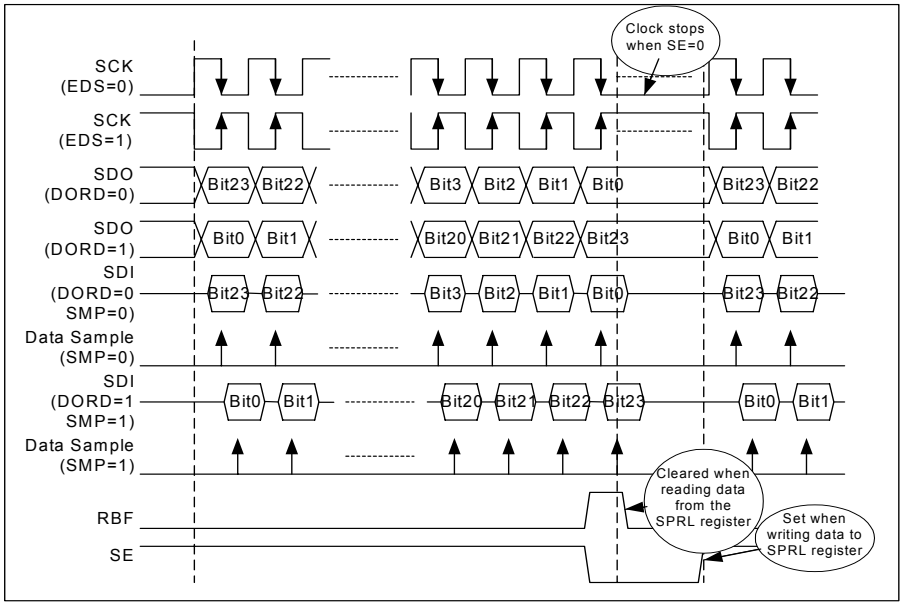


Fig. 8-20 SPI Master Mode Timing Diagram

Code Example: Master Mode (8bit)

```
;*** Interrupt SPI
PERIPH:
    PUSH
    COMA    DATACNT
;--- SPI read buffer full
    JBC     SPISTA,SRBFI,Q_SPINT
    BC      SPISTA,SRBFI
    BS      INTFLAG,F_SPI
;--- SPI Data collision
    JBC     SPISTA,DCOL,Q_SPINT
    MOV     A,#0XFF
Q_SPINT:
    MOV     DATACNT,A
    POP
    RETI
;=== 8MHz/4 = 2000000 bit rate
SPIM_SR:
       :
    System setting 8MHz
    Port G setting output port
       :
;--- 8bit, Fsystem/2, Rising edge & MSB
    MOV     A,#11001100B
    MOV     SPICON,A
;--- SPI full interrupt
    MOV     A,#00100000B
    MOV     SPISTA,A
;--- Global interrupt
    BS      CPUCON,GLINT
;--- SPI data output => 55
    MOV     A,#0X55
    MOV     DATACNT,A
SPI8LOOP:
    MOV     A,DATACNT
    MOV     SPRL,A
;--- SPI Data collision
    JBC     SPISTA,DCOL,SPI8LP1
    BC      SPISTA,DCOL
;--- SPI data output resend => 55
    MOV     A,#0X55
    MOV     DATACNT,A
SPI8LP1:
    JBC     INTFLAG,F_SPI,SPI8LP1
SPI8LP2:
    BC      INTFLAG,F_SPI
    MOVRP   PORTG,SPRL
    SJMP    SPI8LOOP
```

Slave Mode: (Shift Buffer Length = 8 bits, /SS enable)

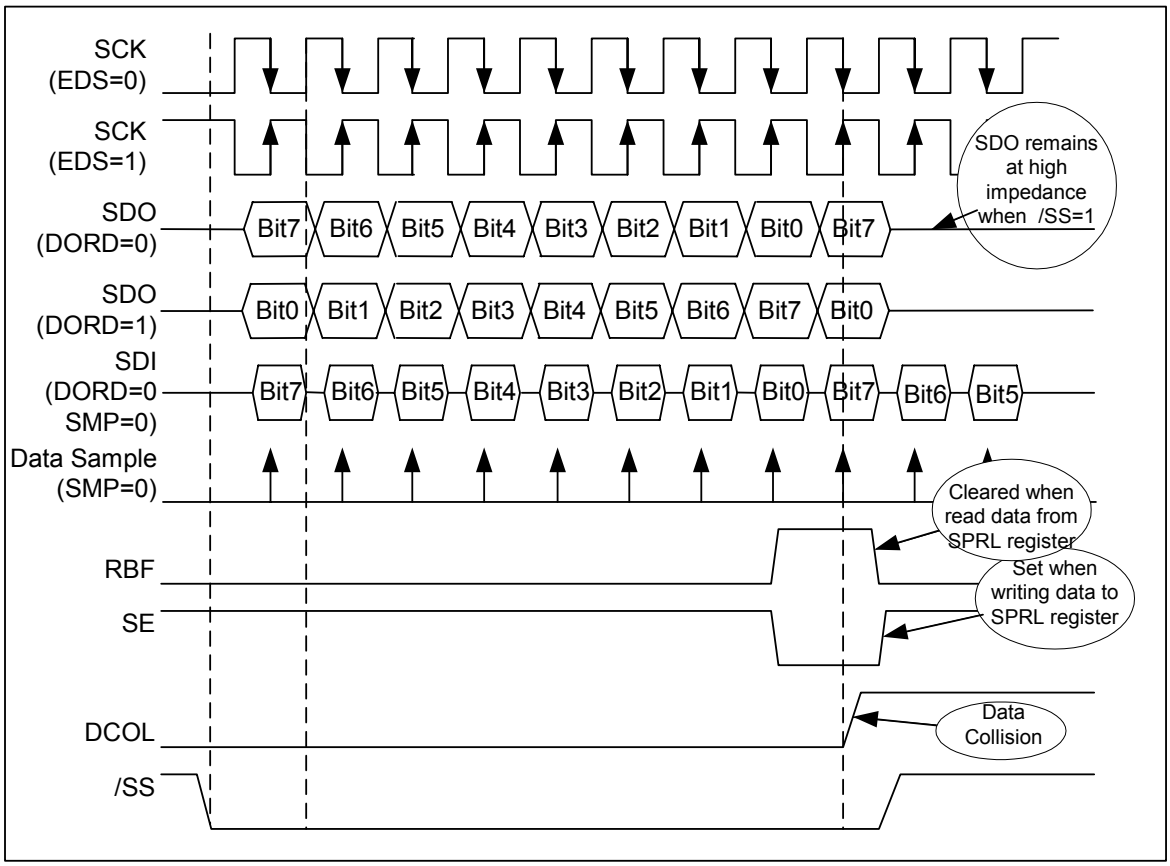


Fig. 8-21 SPI Slave Mode Timing Diagram

Code Example: Slave Mode (8 bit)

```
; *** Interrupt SPI
PERIPH:
       PUSH
       JBC      SPISTA,SRBFI,Q_SPINT
       BC       SPISTA,SRBFI
       BS       INTFLAG,F_SPI
Q_SPINT:
       POP
       RETI
; *** SPI slave mode
         :
       System setting 8MHz
       Port G setting output port
         :
; === SPI 8bit & Sleep mode
SPIS_SR:
; --- 8bit, Slave /SS enable, Rising edge & LSB
       MOV      A,#11110100B
       MOV      SPICON,A
```

```
; --- SPI Wake-up & SPI Full Interrupt
       MOV      A,#00101000B
       MOV      SPISTA,A
; ---  Global interrupt
       BS       CPUCON,GLINT
; --- Sleep mode
       BC       CPUCON,MS1
SPIS8Lp:
       SLEP
       NOP
       MOVRP    PORTG,SPRL
       BC       INTFLAG,F_SPI
; --- SPI Data collision
       JBC      SPISTA,DCOL,SPIS8Lp
       MOV      A,#0XFF
       MOV      SPRL,A
       SJMP     SPIS8Lp
```

8.10 Melody/Speech Synthesizer

The EPG3231-EM202 MCU provides four channels for melody/speech function. Channels 1~3 are destined for melody channel, and Channel 4 can be a melody or a speech channel determined by the SPHSB bit (bit 2 of R44). The Melody Channels 1 ~ 4 / Speech Channel can be controlled by R45 ~ R4A of the corresponding control register Banks 0 ~ 3. Bit 0 ~ Bit 2 of R44 are used to select the current control register bank.

	Melody Channel 1	Melody Channel 2	Melody Channel 3	Melody Channel 4 / Speech Channel
R44h	xxxx x000	xxxx x001	xxxx x010	xxxx x011 / xxxx x1xx
R45h	ADDL	ADDL	ADDL	ADDL / -
R46h	ADDM	ADDM	ADDM	ADDM / -
R47h	ADDH	ADDH	ADDH	ADDH / -
R48h	ENV	ENV	ENV	ENV / SPHDR
R49h	MTCON	MTCON	MTCON	MTCON / SPHTCON
R4Ah	MTRL	MTRL	MTRL	MTRL / SRHTRL

- SFCR (R44h): Special Function Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGMD2	AGMD1	AGMD0	WDTPSR1	WDTPSR0	SPHSB	CSB1	CSB0

Bits 0 ~ 1 (CSB0 ~ CSB1): Channel select bits

Bit 2 (SPHSB): Speech Channel/Melody Channel 4 select bit

"**0**" : Melody Channel 4 enabled, Speech channel disabled

"**1**" : Melody Channel 4 disabled, Speech channel enabled

SFCR [2: 0]	Channel Selection	Control Register Bank
000	Melody Channel 1	Bank 0
001	Melody Channel 2	Bank 1
010	Melody Channel 3	Bank 2
011	Melody Channel 4	Bank 3
1xx	Speech Channel	Bank 3

8.10.1 Melody Function

The MCU melody function can effectively manage the instrument waveform address setting, instrument synthesis frequency control, and envelope control. It is embedded with four melody channels and with built-in large data ROM size for melody waveform data storage. Its melody timer clock source is from Fper/2. Three melody timers (MT1, MT2, & MT3) provide interrupt function while the fourth one (MT4) is without interrupt. The Melody timer is an 11 bits timer for time counting. When the melody timer counting value underflows, interrupt occurs and the MTRL value is automatically reloaded to counting value. To synthesize the instrument melody, user should write the starting address of the waveform to R45 ~ R47, setup the envelope value, and then enable the melody timer. The control register interrupt rate is calculated as follows:

$$Interrupt\_rate = \frac{Fper}{MTPSR \times MTRL[10:0]+1}$$

■ **Melody Interrupt and Interrupt Priority (for Code Option)**

"Mode 1" : Disable the melody interrupt function and interrupt priority
(external > capture > speech > Timers 0 ~ 2 > peripheral)

"Mode 2" : Enable the melody interrupt function and interrupt priority
(Timer 0 > capture > speech/melody > external > Timer 1, 2 > peripheral)

The control registers are listed as follows**:**

- ADDH, ADDM, ADDL (R47h ~ R45h): Address Registers (Write-only Register)

These registers, i.e., ADDL, ADDM, and ADDH are treated as instrument waveform address. Each melody channel has its own waveform data address pointer that points to the waveform start address in data ROM. The address values are written by program and its total length is 24 bits.

- ENV (R48h): Envelope Register

The envelope register stores the envelope value for the current melody channel. The user's program should calculate the proper envelope value to obtain a suitable ADSR (Attack-Decay-Sustain-Release) for different instruments. The tone generator will process the waveform data with the envelope automatically and then synthesize the final instrument melody to the mixer of the PWM and D/A converter.

The data written to the envelope register should be a 7-bit unsigned value and located in Bits 0~6 (the corresponding envelope value must be 0 to 127), which means the envelope resolution is of 128 steps. The reset initial value is "0".

- MTRL (R4Ah): Melody Timer Auto-reload Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MTRL7	MTRL6	MTRL5	MTRL4	MTRL3	MTRL2	MTRL1	MTRL0

The melody timer is an 11-bit down counter for melody applications. The frequency generated by the melody timer is determined by the value of 11-bit melody timer auto-reload register (including MTRL and MTRLH0~2 of MTCON). When the counter value underflows, the timer will be auto-reloaded. To obtain the correct frequency, refer to the frequency reference table and look for the correct value for MTRL and MTRLH0~2 of MTCON.

- MTCON (R49h): Melody Timer Control Register

The MTCON is used to determine the three MSB's of the 11-bit auto-reload register and enable/disable the melody timer of the current melody channel. Once the melody timer is enabled, it will fetch the waveform data (pointed to by the address registers) from data ROM, process the data with the envelope, and then feed the data to the DAC or PWM mixer automatically.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MTPSR1 Mode 2 only	MTPSR0 Mode 2 only	MTI Mode 2 only	MTIE Mode 2 only	MTEN	MTRLH2	MTRLH1	MTRLH0

Bits 2 ~ 0 (MTRLH0 ~ MTRLH2): Bits 8 ~10 of the melody timer auto-reload register.

Bits 4 ~ 3 (MTIE, MTEN): Melody Timer Enable Control Bits

MTIE	MTEN	Melody Timer Function
×	0	Melody Timer Disable
0	1	Melody Timer Enable and read melody data without interrupt
1	1	Melody Timer interrupt enable without fetching the melody data. (Mode 2 only)

Bit 5 (MTI): Melody timer interrupt flag. Set to 1 when a melody timer interrupt occurs. Clear to "0" by software or disable melody timers (Mode 2 only).

Bit 7 ~ 6 (MTPSR1 ~ MTPSR0): Melody timer pre-scale control bits (Mode 2 only).

MTPSR1: MTPSR0	Prescaler Value
00	1:1
01	1:4
10	1:16
11	1:64

8.10.2 Speech Function

The 11-bit melody timer (MT4) for Channel 4 is shared with speech timer. The clock source for speech timer is from Fper/2. When R44 [2:0] = "1xx," the control register bank will change to speech channel. An interrupt function is available for user's application. The interrupt rate is calculated as follows**:**

$$Interrupt\_rate = \frac{Fper}{SPHTPSR \times SPHTRL[10:0] + 1}$$

The control registers are listed as follows:

- SPHDR (R48h): Speech Data Register

In speech function control, SPHDR acts as an output window to the PWM and D/A converter mixer. The program should write the synthesized data to SPHDR, and the data will be fed into the mixer at the next speech timer underflow. For correct mixing operation, the value to be written to SPHDR must be an 8-bit signed data. The reset initial value is "0."

- SPHTRL (R4Ah): Low Byte of Speech Timer Auto-reload Register

Speech timer is an 11-bit down counter for speech applications. The frequency generated by speech timer is determined by the value of the 11-bit auto-reload register, including SPHTRL and SPHTRLH0 ~ SPHTRLH2 of SPHTCON. When the counter value underflows, a timer interrupt will occur and auto-reload from the 11-bit auto-reload register.

- SPHTCON (R49h): Speech Timer Control Register

SPHTCON is used to determine the three MSB of the 11-bit auto-reload register and enable/disable the speech timer.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPHTPSR1 Mode2 only	SPHTPSR0 Mode2 only	SPHTI	SPHTIE	SPHTEN	SPHTRLH2	SPHTRLH1	SPHTRLH0

Bits 2~0 (SPHTRLH2~ SPHTRLH0): Bits 8~10 of the 11-bit auto-reload register

Bit 4 ~ 3 (SPHTIE, PSHTEN): Speech Timer Enable Control Bits

SPHTIE	SPHTEN	Speech Timer Function
×	0	Speech Timer disabled
0	1	Speech Timer enabled and read speech data without interrupt.
1	1	Speech Timer interrupt enabled without fetching speech data.

Bit 5 (SPHTI): Speech timer interrupt flag. Set to 1 when a speech timer interrupt occurs. Cleared to 0 by software or disable the speech timers.

Bits 7 ~ 6 (SPHTPSR1 ~ SPHTPSR0): Speech Timer Prescaler Control Bits (Mode 2 only)

SPHTPSR1: SPHTPSR0	Prescaler Value
00	1:2
01	1:8
10	1:32
11	1:128

- CPUCON (R0Eh): MCU Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEN	-	-	SMCAND	SMIER	GLINT	MS1	MS0

Bit 2 (GLINT): Global interrupt control bit

"**0**" : Disable all interrupts

"**1**" : Enable all un-masked interrupts

8.11 PWM / DAC Function

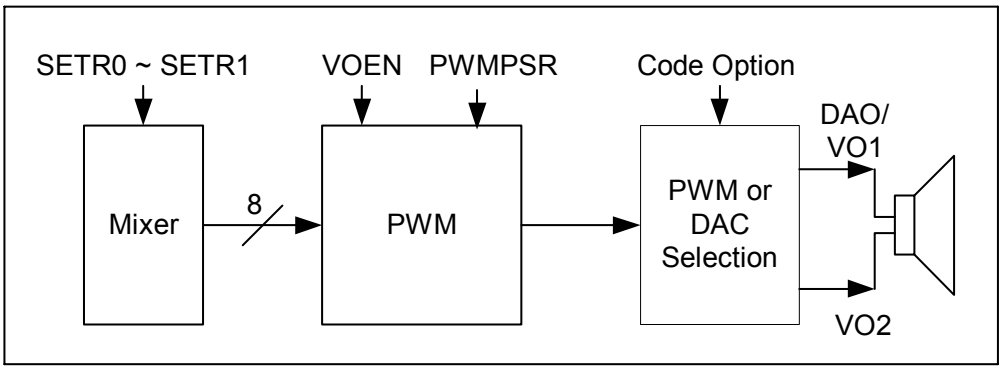
The ePG3231-EM202 is embedded with two choices of melody/speech outputs, i.e., PWM and D/A converter.

When the PWM function is enabled, the voice output uses PWM to drive the speaker directly. The 8-bit PWM function block diagram is shown in the following figure. The PWD register is double buffered for glitch free operation.

When Bit 7 of PWD is "1" and the PWM timer counter is equal to the PWM value (Bits 0 ~ 6 of PWD), the VO1 transfers to low until the PWM timer is reset or overflows. The VO2 is always kept at "0" in this case.

When Bit 7 of PWD is "0" and the PWM timer counter is equal to the inverse of Bits 0 ~ 6 of PWD, the VO2 transfers to low until the PWM timer is reset or overflows. The VO1 is always kept at "0" in this case.

$$T_{period} = \frac{128}{F_{per}} \times \Pr escale;\ T_{duty} = \frac{1}{F_{per}} \times \Pr escale \times (PWD + 1)$$



8.11.1 PWM Function Block Diagram

Fig. 8-22 PWM Function Block Diagram

8.11.2 DAC Function Block Diagram:

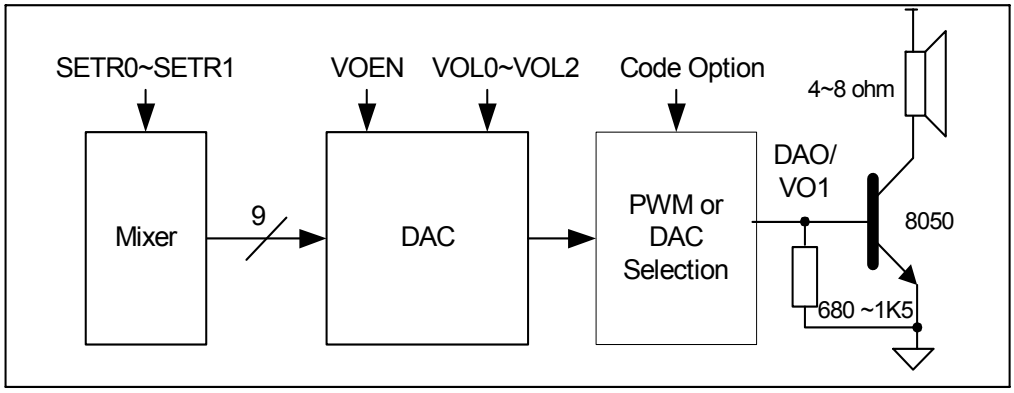


Fig. 8-23 DAC Function Block Diagram

If both SPHSB and VOEN bits are set to “1” and SPHTEN bit is cleared to “0”, the data of the speech data register will be output immediately through the D/A converter or PWM when the register is changed.

8.11.3 Current D/A Converter Reference Source (for Code Option)

"D/A Converter Internal Reference Voltage": D/A converter output varies in accordance with temperature and process variations.

"A/D Vrex and Port A.6 Rref": A/D converter external reference voltage & D/A converter reference resistor.

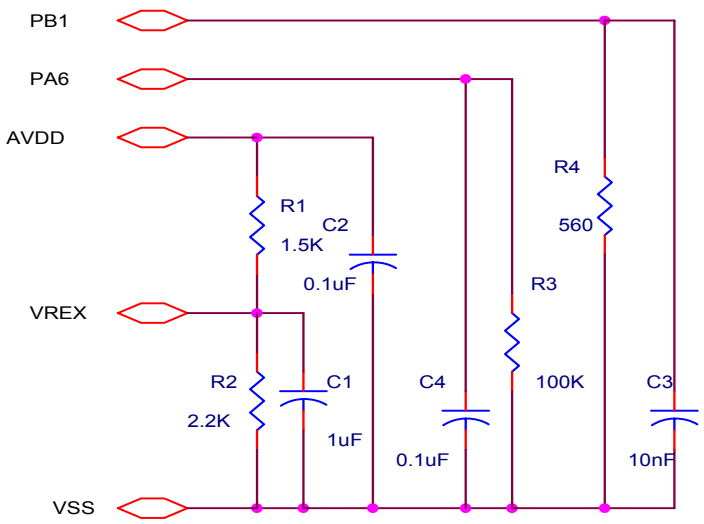


Fig. 8-24 External A/D & D/A Converters Reference Circuits

8.11.4 PWM / DAC Function Registers

- VOCON (R4Bh): Voice Output Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VOEN	DAC	SETR1	SETR0	PWMPSR	VOL2	VOL1	VOL0

Bits 0 ~ 2 (VOL0 ~ VOL2): DAC Volume control

VOL2 ~ VOL0	Volume
000	1 (min.)
001	2
010	3
011	4
100	5
101	6
110	7
111	8 max.

Bit 3 (PWMPSR): PWM Timer pre-scale select bit

"**0**" : Pre-scale 1:1

"**1**" : Pre-scale 1:2

Bits 5 ~ 4 (SETR1 ~ SETR0): Set dynamic range

While mixing, the accumulation result of mixer may have large dynamic range (up to 11 bits), but the DAC has only 9-bit resolution and PWM has only 8-bit. You can define suitable output data range to prevent saturation condition.

SETR1~SETR0	Output Data Fed to PWM/DAC
10	Take Bits 3~10 of mixer accumulation result of PWM Take Bits 2~10 of mixer accumulation result of DAC
01	Take Bits 2~9 of mixer accumulation result of PWM Take Bits 1~9 of mixer accumulation result of DAC
00 or 11	Take Bits 1~8 of mixer accumulation result of PWM Take Bits 0~8 of mixer accumulation result of DAC

Bit 6 (DAC): D/A output control bit (for Code option, Current D/A reference source)

"**0**" : D/A output control by melody timer and speech timer
"**1**" : D/A output control through SPHDR register

Bit 7 (VOEN): Voice Output Control Bit

"**0**": DAC/PWM disable
"**1**" : DAC/PWM enable

- ADCON (R2Ch): A/D Control Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DET	VRS	ADEN	PIRQB	S/DB	CHS2	CHS1	CHS0

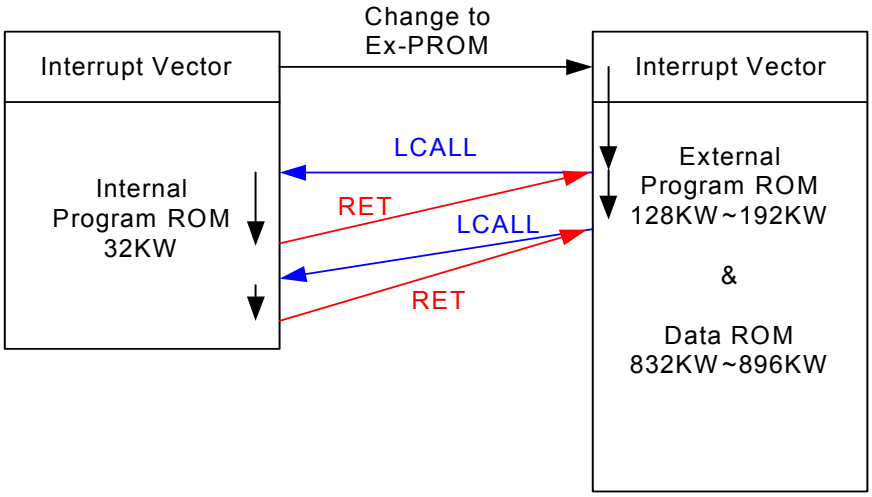
Bit 6 (VRS): A/D input reference voltage select and enable/disable internal reference generator bit.

"**0**": Enable internal reference generator and the reference voltage is from the internal reference voltage generator.
"**1**": Disable internal reference generator and the reference voltage is from the external VREX pin.

Code Example: Refer to Melody & Speech application note.

9 Program Library

9.1 Program Library Control Block



9.2 CPU Mode Control Library

RAM: Unbanked RAM 0×50 ~ 0×52, provides Interrupt backup ACC, STATUS & BSR.

MACRO List:

- PUSH: Interrupt routine pushes ACC, STATUS & BSR data
- POP: Interrupt routine pops up ACC, STATUS & BSR data.
- JNE: Jump to address if REG1 <> REG2.
- InAddr: Set address to ADD (H, M, L) RAM
- InData: Set data to I_DATA (H, L) RAM
- In_CS: Set CS control data to CS_BUF RAM

Subroutine List:

- IOset_Before_Sleep: Set the I/O pin function (Ports D, E, F, G, and H) before going into Sleep mode and Idle mode. Not setting the I/O port will cause a large current problem in Sleep and Idle modes.
- ENTER_SLEEP: Controls the CPU to enter Sleep mode, and the flash IC /CS pin to a high level (disable), and auto execute "IOset_Before_Sleep" library.
- ENTER_IDLE: Controls the CPU to enter Idle mode, and the flash IC /CS pin to a high level (disable), and auto execute "IOset_Before_Sleep" library.
- ENTER_PFS: Changing CPU fast mode frequency, must wait for 5ms to change to Low frequency from 12MHz~16MHz. No need to wait when changing to other frequency.
- PFS_Stable: In changing the CPU fast mode frequency, wait for 5ms since the system must work mainly in a stable frequency.

9.3 ELAN, AMD & MXIC Flash IC Control Library

RAM: Unbanked RAM 0×53 is flash control flag, RAM 0×60 ~ 0×67 control flash library.

Subroutine List:

- ExMemIni: External memory control I/O initial
- ExFReset: Control Flash IC S/W reset
- ExFReadID: Read Flash ID code
- ExFSecErase: Execute Flash sector erase function
- ExFBlkErase: Execute Flash Block erase function **(only for EM39LV800)**
- ExFAllErase: Execute Flash All chip erase function
- ExFWrite: Execute Flash first write data
- ExFSeqWr: Execute Flash sequence write data
- ExMemRead: Execute Flash first read data
- ExMemSeqRd: Execute Flash sequence read data

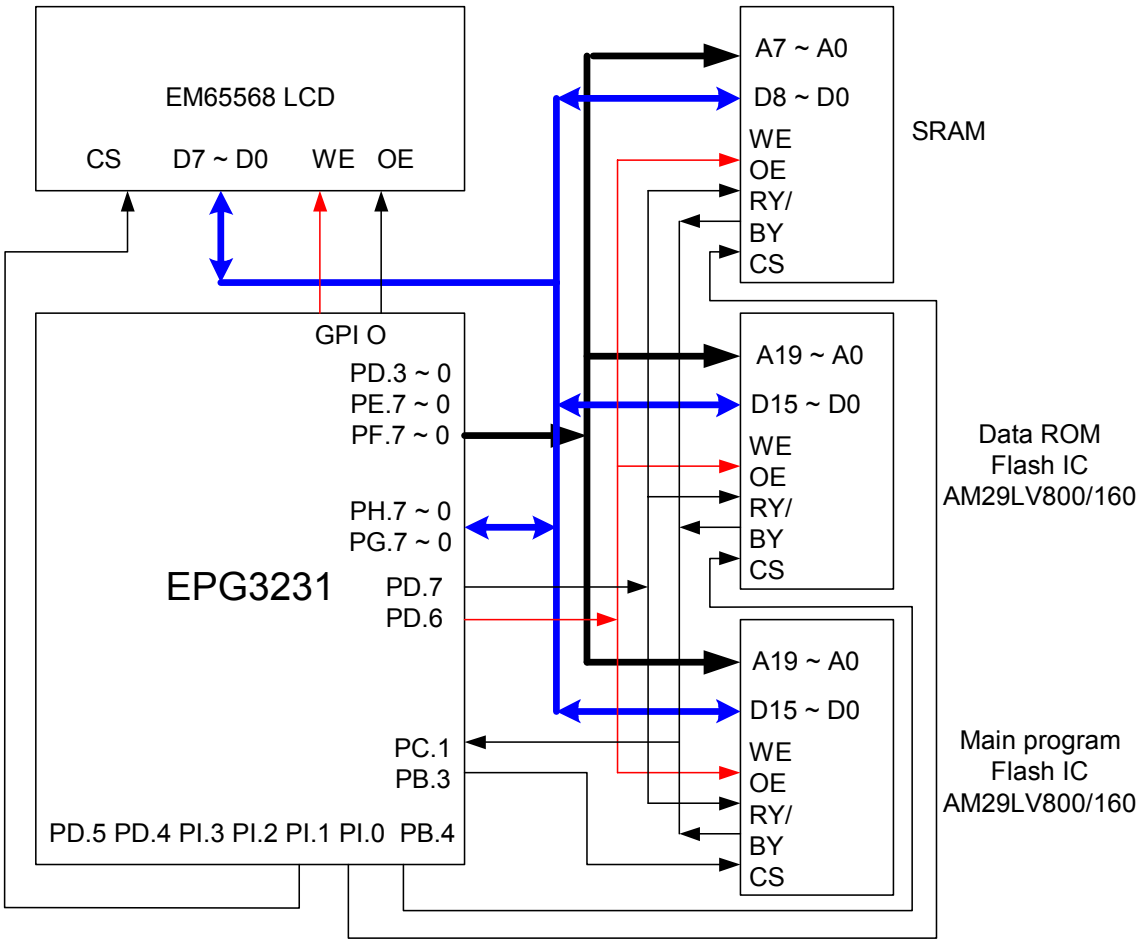
9.4 External Device Control Library

RAM: Unbanked RAM 0×53 is external device interface control flag, RAM 0×60 ~ 0×67 control external memory interface library.

Subroutine List:

- ExMemIni: External device control I/O initial
- ExMemWrite: Execute External device first write data
- ExMemSeqWr: Execute External device sequence Write data
- ExMDWrite: Execute External device only data write
- ExMemRead: Execute External device first read data
- ExMemSeqRd: Execute External device sequence Read data
- ExMDRead: Execute External device only data read

External Memory Interface Circuit:



9.5 Touch Panel Control Library

RAM: Unbanked RAM 0x53 control touch panel flag, RAM 0x60 ~ 0x67 control touch panel library.

MACRO List:

- InDIV16: Set Dividend 16 bits & Divisor value
- InDIV8: Set Dividend 8 bits & Divisor value

Subroutine List:

- InitialTP: Set touch panel A/D initial
- DetPenDown: Detect touch panel pen down routine
- GetTPXY: Get touch panel X & Y point values
- TPHold: Detect key release for touch panel control
- UDIV1608: Unsigned 16 bits / 8 bits subroutine
- UDIV8_8: Unsigned 8 bits / 8 bits subroutine

9.6 Melody & Speech Library

RAM: Unbanked RAM 0×53, 0x54 control flag, RAM Bank 0 0×BC ~ 0×FF melody & speech library control RAM.

MACRO List:

- InExVDP: Set Voice data point high & middle byte
- InMidNo: Set melody number
- InMelody: Set melody code point
- InSphNo: Set Speech number
- InSpeech: Set Speech data point
- InTone: Set Tone frequency
- WaitMid: Wait Melody/Speech play ending

Subroutine List:

- MelodyInitial: Melody & Speech initial
- VO_ON: Turn-on Voice output
- VO_OFF: Turn-off Voice output
- PlayMelodyNo: Call play melody number
- MelodyResume: Call resume melody play
- MelodyPause: Call melody play pause
- MelodyOff: Call melody turn off
- PlayMelody: Call play melody
- PlaySpeechNo: Call play Speech number
- SpeechResume: Call resume speech play
- SpeechPause: Call speech play pause
- SpeechOff: Call Speech turn off
- PlaySpeech: Call play Speech
- VoicePause: Call melody & speech play pause
- VoiceResume: Call resume melody & speech play
- PlayOneTone: Call play single tone

10 Download Flash Data Control Program: (PC RS232)

10.1 Operation Conditions Setting

S/W setting: PLL=14.745MHz (9.83MHz if Baud rate used 38400), Baud rate = 230400, (115200, 57600, 38400,) even parity check, UART mode = 8 bit

H/W setting: Connect Port A.7 to VSS, connect the PMD pin to VSS

Bit No. of Port A				Select Baud Rate
7	6	5	4 ~ 0	
0	1	1	11111	230400
0	0	1	11111	38400
0	1	0	11111	115200
0	0	0	11111	57600

10.2 Control Flash Type

AMD flash IC: AM29LV800T/B, AM29LV160 T/B, AM29LV320 T/B, AM29LV640 T/B

MXIC flash IC: MX29LV800 T/B, MX29LV160 T/B, MX29LV320 T/B, MX29LV640 T/B

Elan flash IC: EM39LV800

10.3 Main Function

1. Read flash ID
2. Sector erase
3. All chip erase
4. All chip programming
5. Partial programming
6. Change partial programming size. (32KW, 64KW, 128KW & 256KW)
7. Change flash IC type (ELAN or AMD or MXIC)
8. No defined command
9. Read flash data. (2KW; Next continue 2KW & All data)

10.4 Command & Output Message Table

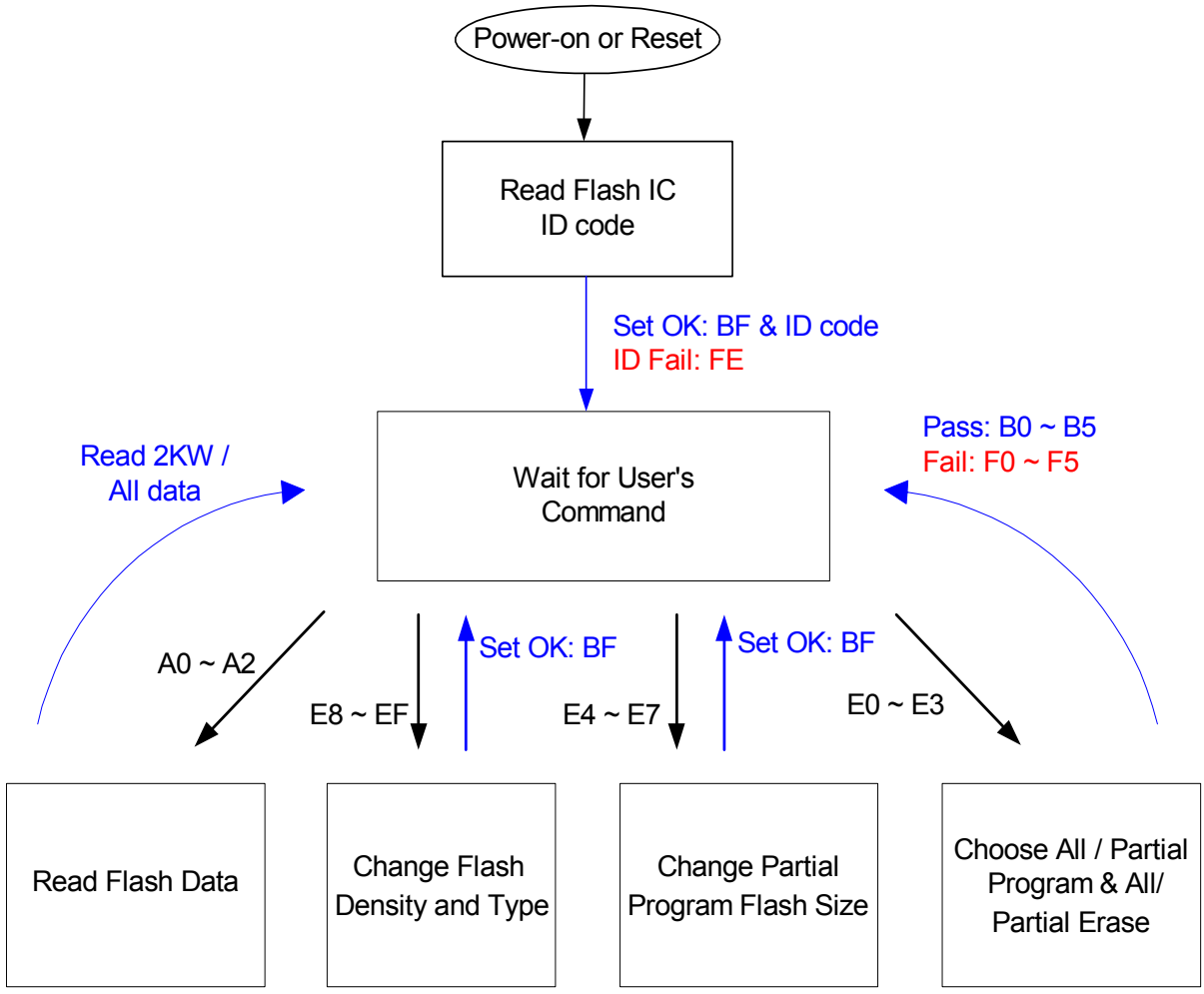
Command	Function Description
A0	Read 2KW flash data
A1	Read next 2KW flash data
A2	Read all flash data
D8	Elan flash EM39LV800
E0	All chip erase
E1	All chip program
E2	Sector erase
E3	Partial program
E4	Partial program size (32KW)
E5	Partial program size (64KW)
E6	Partial program size (128KW)
E7	Partial program size (256KW)
E8	IC change--TOP/512KW
E9	IC change--TOP/1MW
EA	IC change--TOP/2MW
EB	IC change--TOP/4MW
EC	IC change--BOT/512KW
ED	IC change--BOT/1MW
EE	IC change--BOT/2MW
EF	IC change--BOT/4MW

Output	Function Description
B0	All chip erase/verify pass
B1	All chip program/verify pass
B2	Sector erase/verify pass
B3	Partial erase/verify pass
B4	Partial program/verify pass
BE	Command ACK
BF	Setting OK
F0	All chip erase/verify fail
F1	All chip program/verify fail
F2	Sector erase/verify fail
F3	Partial erase/verify fail
F4	Partial program/verify fail
FA	Framing error
FB	Overrun error
FC	Parity error
FE	Read ID fail
FF	Command error

10.5 Down Load Flowchart

Condition:

H/W setting: Connect Port A.7 to VSS, then connect the PMD pin to VSS.



11 Electrical Characteristic

■ Absolute Maximum Ratings

Items	Symbol	Condition	Limits	Unit
Supply voltage	VDD		-0.3 to +3.6	V
Input voltage (general input port)	VIN		-0.5 to VDD +0.5	V
Power Consumption (Topr=70°C)	PD		400	mW
Soldering Temperature (time)	Tsld		350 (3sec)	°C
Operating temperature range	Topr		-10 to +70	°C
Storage temperature range	Tstr		-55 to +125	°C

■ Recommended Operating Conditions

Items	Symbol	Condition	Limits	Unit
Supply voltage	VDD		2.2 to 3.6	V
	AVDD		2.4 to 3.6	
Input voltage	VIH		VDD x 0.9 to VDD	V
	VIL		0 to VDD x 0.1	V
A/D full-Scale input span	ADRG	Positive input-negative input	0 to VREX	V
Operating temperature	Topr		-10 to +70	°C

■ DC Electrical Characteristics

(Condition: Ta=-10~+70°C, VDD= 3.0 ± 0.3V)

Parameter	Symbol	Condition		Min.	Typ.	Max.	Unit
CLOCK	Fmain	Main Clock Frequency		1	-	16	MHz
	Fsub	Sub-clock frequency	RC OSC	24.6	32.768	41	kHz
			Crystal OSC	-	32.768	-	
Supply current	Idd1	Sleep mode	VDD=3V, no load	-	-	1	μA
	Idd2	Idle mode	VDD=3V RC OSC	-	8	12	
	Idd3		VDD=3V, Crystal OSC	-	5	8	
	Idd4	Slow mode	VDD=3V, RC / Crystal OSC, No load	-	20	30	
	Idd5	Fast mode	VDD=3V, Fmain=4MHz, No load	-	900	1200	
	Idd6		VDD=3V, Fmain=10MHz, No load	-	2000	3000	
			VDD=3V, Fmain=15MHz, No load	-	3000	4000	
Input voltage	VIH1	PA[0:7], PB[0:7], PC[0:7], PD[0:7], PE[0:7], PF[0:7], PG[0:7], PH[0:7], PI[7:0], PJ[7:0], PK[7:0] (as general input port)		VDD×0.7	-	VDD	V
	VIL1			0	-	VDD×0.3	
Input threshold Voltage (Schmitt)	VT+	RSTB, PB.5 as EVIN or CPIN;		0.5×VDD	-	0.75×VDD	V
	VT-			0.2×VDD	-	0.4×VDD	
Output current	IOH1	PB[0:7], PC[0:7], PI[7:4], PG[0:7], PH[0:7] (general output port)	VDD=3V , VOH=2.4V	-1.1	-2.2	-3.3	mA
	IOL1		VDD=3V , VOL=0.2V	+1.1	+2.2	+3.3	
	IOH2	PB[1] (D/A output)	VDD=3.0V,VOH=0.7V	-2.5	-3.5	-4.5	
	IOH3	PB[1:0] (PWM output)	VDD=3.0V,VOH=2.5V	-70	-100	-150	
	IOL3		VDD=3.0V,VOL=0.5V	+70	+100	+150	
	IOH4	PB[3:4]	VDD=3.0V,VOH=1.0V	-3	-6	-9	

Parameter	Symbol	Condition		Min.	Typ.	Max.	Unit
	IOL4		VDD=3.0V,VOL=0.5V	+1.5	+3	+4.5	
	IOH5	PJ[7:0]~PK[7:0] (key strobe)	VDD=3.0V,VOH=2.4V	-5	-10	-15	μA
	IOL5		VDD=3.0V,VOL=0.2V	+0.5	+1	+1.5	mA
	IOH6	PD[7:0]~PF[0:7]	VDD=3.0V,VOH=2.6V	-1.1	-2.2	-3.3	mA
	IOL6		VDD=3.0V,VOL=0.4V	+1.1	+2.2	+3.3	mA
	IOH7	PB[2](IR output), PI[3:0]	VDD=3.0V,VOH=2.1V	-5	-10	-15	mA
	IOL7		VDD=3.0V,VOL=0.9V	+5	+10	+15	mA
Input leakage current	IIL	All Input port (without pull-high/low resistor), Vin= VDD or GND		-	-	±1	μA
Large Pull-high resistance	RPU1	PA[7:0]~PK[7:0]	Vin=GND	500	1000	1500	KΩ
	RPU2	RSTB	Vin=GND	250	500	750	
Small Pull-high resistance	RPU3	PA[7:0]~PK[7:0]	Vin=2.0V	50	100	200	KΩ
	RPU4	RSTB	Vin=2.0V	50	100	200	
Large Pull-low resistance	RPD1	TEST	Vin=VDD	250	500	750	KΩ
Small Pull-low resistance	RPD2	TEST	Vin=1.0V	1.0	2.5	4.0	KΩ
Touch Panel Pull-low resistance	RPD3	DET=1, Xn pin	Vin=VDD	25	50	100	KΩ
Data retention voltage	Vret			1.6	-	-	V
Power-on reset voltage	Vpor			1.4	1.5	1.6	V
A/D Conversion (VDD=3.0V, AVDD=3.0V, Ta=-10~+70°C , Fclk=12*Fsample)							
Analog Input							
Mux leakage current	Imux	On/off leakage current, Vin=0 or VDD		-	0.1	1	μA
System Performance							
Resolution				-	10	-	Bits
Integral nonlinearity	INL			-2	-	+2	LSB
Differential nonlinearity	DNL			-2	-	+2	LSB
Offset error	OErr			-4	-	+4	LSB
Gain error	GErr			-4	-	+4	LSB
No Missing code	MC			No missing code			bit
AVDD Supply current	Ivdd3	AVDD=3.0V,VDD=3.0V, Fsample=20kHz, ADEN=1, VRS=1		-	0.5	0.7	mA
	Ivdd4	ADEN=0, VRS=1		-	-	1	uA
Driver current	IOH	Xp, Yp (VDD= 2.9 ± 0.3V) (Voh=VDD-0.2V)		-20	-30	-45	mA
Sink current	IOL	Xn, Yn (VDD= 2.9 +/- 0.3V) (Vol=0.2V)		+20	+30	+45	mA
Reference voltage							
Internal reference voltage	VRIN	AVDD=3.0 ± 0.3V		1.8	2.0	2.2	V
Internal reference supply current	Ivrin	VDD=3.0V, AVDD=3.0V, VRS=0, Voh=0.2V		400	500	-	μA
VREX input current	Iref1	ADEN=1, VRS=1		-	300	500	μA
	Iref2	ADEN=0, VRS=1		-	-	1	μA

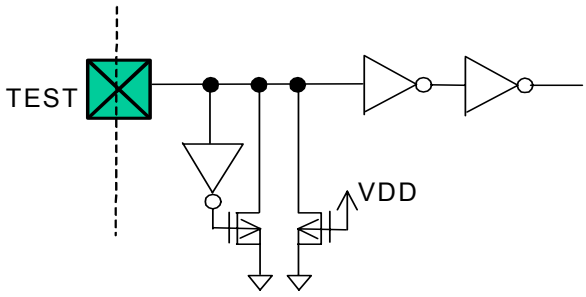
■ **AC Electrical Characteristics** (Condition: Ta=-10~+70°C, VDD= 3.0 ± 0.3V)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Instruction cycle time	Tcycle	Fmain=1MHz	-	2[1]	-	μs
		Fmain=4MHz	-	0.5[1]	-	
		Fmain=16MHz	-	0.34[1]	-	
A/D Conversion (VDD=3.0V, AVDD=3.0V, Ta=-10~+70°C)						
Throughput rate	THP1	VDD=3.0V, AVDD=3.0V	-	-	80	ksps
	THP2	VDD=2.4V, AVDD=2.4V	-	-	60	
Power supply rejection Ratio	PSRR1+		37	40	-	dB
	PSRR1-		43	46	-	
Signal to noise ratio	SNR		51	54	-	dB

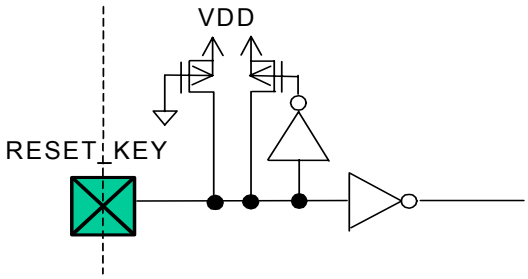
Note [1]: Instruction cycle time= 2 x (System clock time);
[2]: T=time duration between ADEN=1 to ADSTART=1

12 Pin Type Circuit Diagrams

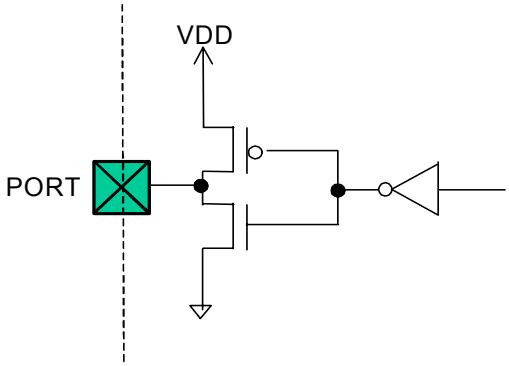
TEST Pin Type



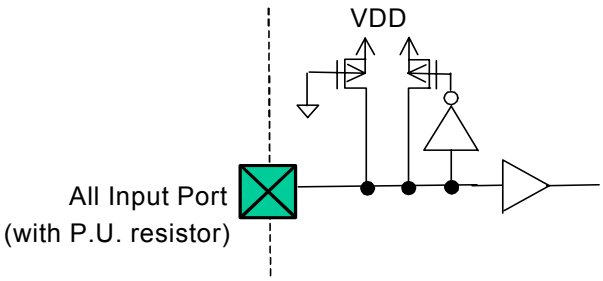
Reset Pin Type



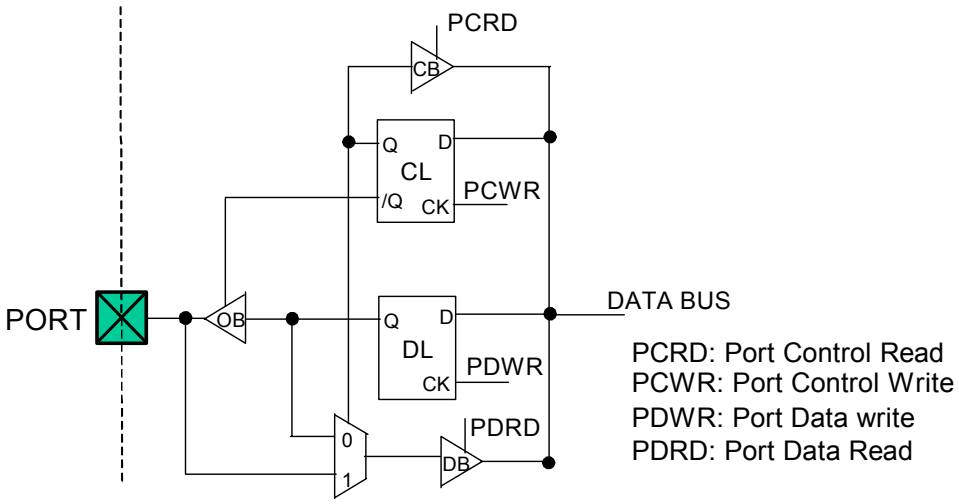
Output Pin Type



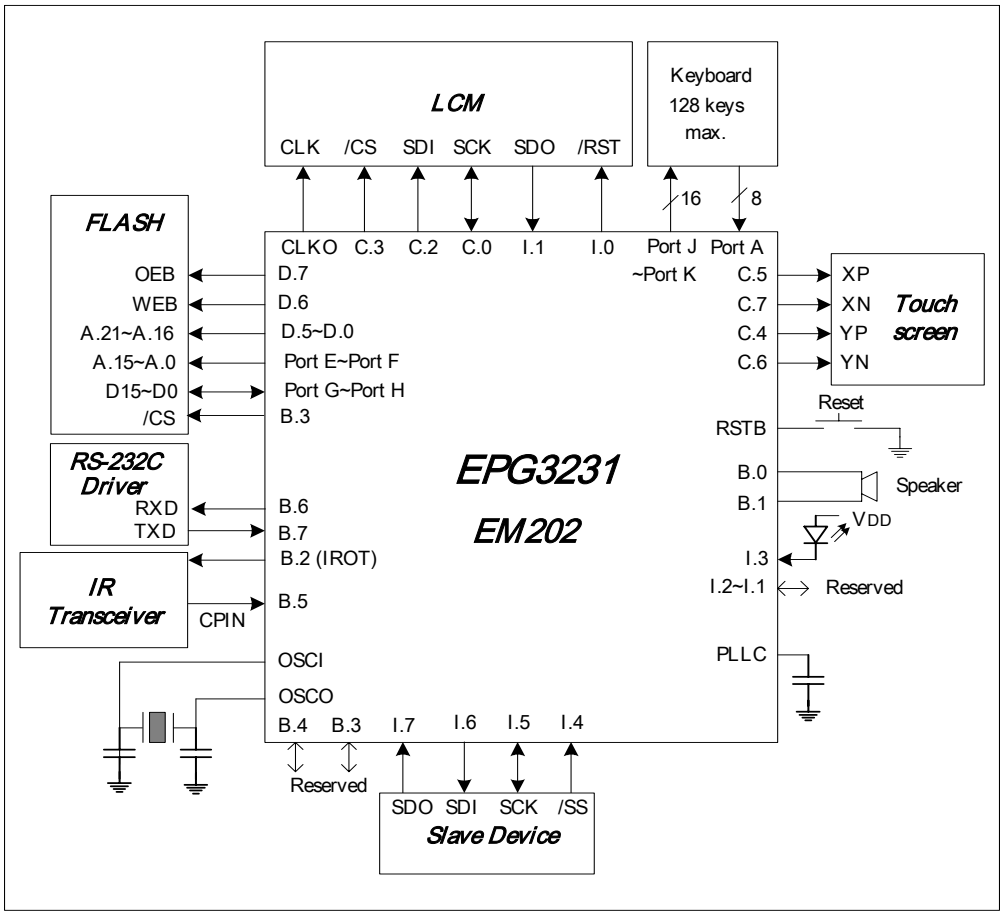
Input Pin Type



General I/O Pin Type



13 Application Circuit



14 Instruction Set

Legend: **addr:** address **i:** Table pointer control **p:** special file register (0h~1Fh)
b: bit **k:** constant **r:** File Register

Type	Instruction Binary	Mnemonic	Operation	Status Affected	Cycle
System Control	0000 0000 0000 0000	NOP	No operation	None	1
	0000 0000 0000 0001	WDTC	WDT ← 0; /TO←1; /PD←1.	None	1
	0000 0000 0000 0010	SLEP	Enter Idle Mode if MS1=1. Enter Sleep Mode if MS1=0.	None	1
	0010 0111 rrrr rrrr	RPT r	Single repeat *(r) times on next instruction. *(r) is the content of register r.	None	1
	0100 0011 kkkk kkkk	BANK #k	BSR ← k.	None	1
Rom Table Look Up	0100 0000 kkkk kkkk	TBPTL #k	TABPTRL ← k.	None	1
	0100 0001 kkkk kkkk	TBPTM #k	TABPTRM ← k.	None	1
	0100 0010 kkkk kkkk	TBPTH #k	TABPTRH ← k.	None	1
	0010 10 i i rrrr rrrr	TBRD i,r	r←ROM[(TABPTR)]. (*1) (*2)	None	2
	0010 1011 rrrr rrrr	TBRD A,r	r←ROM[(TABPTR+ACC)] (*2)	None	2
Data Transfer	0010 0100 rrrr rrrr	CLR r	r ← 0.	Z	1
	0100 1110 kkkk kkkk	MOV A,#k	A ← k.	None	1
	0010 0000 rrrr rrrr	MOV A,r	A ← r.	Z	1
	0010 0001 rrrr rrrr	MOV r,A	r ← A.	None	1
	100p pppp rrrr rrrr	MOVRP p,r	Register p ← Register r.	None	1
	101p pppp rrrr rrrr	MOVPR r,p	Register r ← Register p.	None	1
Exchange	0000 1111 rrrr rrrr	SWAP r	r(0:3)←→r(4:7)	None	1
	0000 1110 rrrr rrrr	SWAPA r	r(0:3)→A(4:7);r(4:7)→A(0:3)	None	1
Bit Manipulation	0110 1bbb rrrr rrrr	BC r,b	r(b) ← 0	None	1
	0111 0bbb rrrr rrrr	BS r,b	r(b) ← 1	None	1
	0111 1bbb rrrr rrrr	BTG r,b	r(b) ← /r(b)	None	1
Arithmetic Operation	0001 1100 rrrr rrrr	INCA r	A ← r+1.	C,Z	1
	0001 1101 rrrr rrrr	INC r	r ← r+1	C,Z	1
	0001 0000 rrrr rrrr	ADD A,r	A ← A+r	C,DC,Z,OV,SGE,SLE	1
	0001 0001 rrrr rrrr	ADD r,A	r ← r+A (*4)	C,DC,Z,OV,SGE,SLE	1
	0100 1010 kkkk kkkk	ADD A,#k	A ← A+k	C,DC,Z,OV,SGE,SLE	1
	0001 0010 rrrr rrrr	ADC A,r	A ← A+r+C	C,DC,Z,OV,SGE,SLE	1
	0001 0011 rrrr rrrr	ADC r,A	r ← r+A+C	C,DC,Z,OV,SGE,SLE	1
	0100 1011 kkkk kkkk	ADC A,#k	A ← A+k+C	C,DC,Z,OV,SGE,SLE	1
	0001 1110 rrrr rrrr	DECA r	A ← r-1	C,Z	1
	0001 1111 rrrr rrrr	DEC r	r← r-1	C,Z	1
	0001 0110 rrrr rrrr	SUB A,r	A ← r-A (*6)	C,DC,Z,OV,SGE,SLE	1
	0001 0111 rrrr rrrr	SUB r,A	r ← r-A (*6)	C,DC,Z,OV,SGE,SLE	1
	0100 1100 kkkk kkkk	SUB A,#k	A ← k-A (*6)	C,DC,Z,OV,SGE,SLE	1
	0001 1000 rrrr rrrr	SUBB A,r	A ← r-A-/C (*6)	C,DC,Z,OV,SGE,SLE	1
	0001 1001 rrrr rrrr	SUBB r,A	r ← r-A-/C (*6)	C,DC,Z,OV,SGE,SLE	1
	0100 1101 kkkk kkkk	SUBB A,#k	A ← k-A-/C (*6)	C,DC,Z,OV,SGE,SLE	1

Legend: **addr:** address **i:** Table pointer control **p:** special file register (0h~1Fh)
b: bit **k:** constant **r:** File Register

Type	Instruction Binary	Mnemonic	Operation	Status Affected	Cycle
Arithmetic Operation	0010 0110 rrrr rrrr	MUL A,r	PRODH:PRODL ← A ***(r)**	None	1
	0100 1111 kkkk kkkk	MUL A,#k	PRODH:PRODL ← A ***(k)**	None	1
	0001 0100 rrrr rrrr	ADDDC A,r	A ← (Decimal ADD) A+r+C	C, DC, Z	1
	0001 0101 rrrr rrrr	ADDDC r,A	r ← (Decimal ADD) r+A+C	C, DC, Z	1
	0001 1010 rrrr rrrr	SUBDB A,r	A ← (Decimal SUB) r-A-/C	C, DC, Z	1
	0001 1011 rrrr rrrr	SUBDB r,A	r ← (Decimal SUB) r-A-/C	C, DC, Z	1
Logic Operation	0000 0010 rrrr rrrr	OR A,r	A ← A .or. r.	Z	1
	0000 0011 rrrr rrrr	OR r,A	r ← r .or. A.	Z	1
	0100 0100 kkkk kkkk	OR A,#k	A ← A .or. k.	Z	1
	0000 0100 rrrr rrrr	AND A,r	A ← A .and. r.	Z	1
	0000 0101 rrrr rrrr	AND r,A	r ← r .and. A.	Z	1
	0100 0101 kkkk kkkk	AND A,#k	A ← A .and. k.	Z	1
	0000 0110 rrrr rrrr	XOR A,r	A ← A .xor. r.	Z	1
	0000 0111 rrrr rrrr	XOR r,A	r ← r .xor. A.	Z	1
	0100 0110 kkkk kkkk	XOR A,#k	A ← A .xor. k.	Z	1
	0000 1000 rrrr rrrr	COMA r	A ← /r.	Z	1
	0000 1001 rrrr rrrr	COM r	r ← /r.	Z	1
Rotate	0000 1010 rrrr rrrr	RRCA r	A(n-1) ←r(n);C ←r(0);A(7) ←C	C	1
	0000 1011 rrrr rrrr	RRC r	r(n-1) ←r(n);C ←r(0);r(7) ←C	C	1
	0000 1100 rrrr rrrr	RLCA r	A(n+1) ←r(n);C ←r(7);A(0) ←C	C	1
	0000 1101 rrrr rrrr	RLC r	r(n+1) ←r(n);C ←r(7);r(0) ←C	C	1
Shift	0010 0010 rrrr rrrr	SHRA r	A(n-1)←r(n); A(7)←C	None	1
	0010 0011 rrrr rrrr	SHLA r	A(n+1)←r(n); A(0)←C	None	1
Bit Compare & Jump	0101 1bbb rrrr rrrr aaaa aaaa aaaa aaaa	JBC r,b,addr	If r(b)=0,jump to addr; PC[15:0] ← addr. **(*3)**	None	2
	0110 0bbb rrrr rrrr aaaa aaaa aaaa aaaa	JBS r,b,addr	If r(b)=1,jump to addr; PC[15:0] ← addr. **(*3)**	None	2
Compare	0010 0101 rrrr rrrr	TEST r	Z ← 0 if r<>0;Z ← 1 if r=0.	Z	1
Compare & Jump	0101 0000 rrrr rrrr aaaa aaaa aaaa aaaa	JDNZ A,r,addr	A ← r-1, jump to addr if not zero; PC [15:0] ← addr. **(*3)**	None	2
	0101 0001 rrrr rrrr aaaa aaaa aaaa aaaa	JDNZ r,addr	r ← r-1, jump to addr if not zero; PC [15:0] ← addr. **(*3)**	None	2
	0101 0010 rrrr rrrr aaaa aaaa aaaa aaaa	JINZ A,r,addr	A ← r+1,jump to addr if not zero; PC[15:0] ← addr. **(*3)**	None	2
	0101 0011 rrrr rrrr aaaa aaaa aaaa aaaa	JINZ r,addr	r ← r+1,jump to addr if not zero; PC[15:0] ← addr. **(*3)**	None	2
	0100 0111 kkkk kkkk aaaa aaaa aaaa aaaa	JGE A,#k,addr	Jump to addr if A≧k; PC [15:0] ← addr. **(*3)**	None	2
	0100 1000 kkkk kkkk aaaa aaaa aaaa aaaa	JLE A,#k,addr	Jump to addr if A≦k; PC [15:0] ← addr. **(*3)**	None	2
	0100 1001 kkkk kkkk aaaa aaaa aaaa aaaa	JE A,#k,addr	Jump to addr if A=k; PC[15:0] ← addr. **(*3)**	None	2

Legend: **addr:** address **i:** Table pointer control **p:** special file register (0h~1Fh)
b: bit **k:** constant **r:** File Register

Type	Instruction Binary	Mnemonic	Operation	Status Affected	Cycle
Compare & Jump	0101 0101 rrrr rrrr aaaa aaaa aaaa aaaa	JGE A,r,addr	Jump to addr if A≧r; PC[15:0] ← addr. (*3)	None	2
	0101 0110 rrrr rrrr aaaa aaaa aaaa aaaa	JLE A,r,addr	Jump to addr if A≦r; PC[15:0] ← addr. (*3)	None	2
	0101 0111 rrrr rrrr aaaa aaaa aaaa aaaa	JE A,r,addr	Jump to addr if A=r; PC[15:0] ← addr. (*3)	None	2
Jump	110a aaaa aaaa aaaa	SJMP addr	PC ← addr; PC [13..16] unchanged	None	1
	0000 0000 0010 aaaa aaaa aaaa aaaa aaaa	LJMP addr (2 words)	PC ← addr.	None	2
Subroutine	0011 aaaa aaaa aaaa	S0CALL addr	Top of Stack] ← PC+1; PC [11:0] ← addr; PC [12:16] ← 00000 (*5)	None	1
	111a aaaa aaaa aaaa	SCALL addr	[Top of Stack] ← PC+1; PC [12:0] ← addr; PC [13:16] unchanged	None	1
	0000 0000 0011 aaaa aaaa aaaa aaaa aaaa	LCALL addr (2 words)	[Top of Stack] ← PC+1; PC ← addr.	None	2
	0010 1011 1111 1110	RET	PC ← (Top of Stack)	None	1
	0010 1011 1111 1111	RETI	PC ← (Top of Stack); Enable Interrupt	None	1

(*1) TBRD i, r:

r ← ROM [(TABPTR)];

i=00: TABPTR not changed

i=01: TABPTR ← TABPTR+1

i=10: TABPTR ← TABPTR-1

(*2) TABPTR = (TABPTRH: TABPTRM: TABPTRL).

*Bit 0 of TABPTRL is used to select low byte or high byte of the pointed ROM data.

Bit 0=0: Low byte of the pointed ROM data

Bit 0=1: High byte of pointed ROM data

*The maximum look-up table space is 8Mbytes.

(*3) The maximum jump range is 64K absolute address; meaning it can jump only within 64K range.

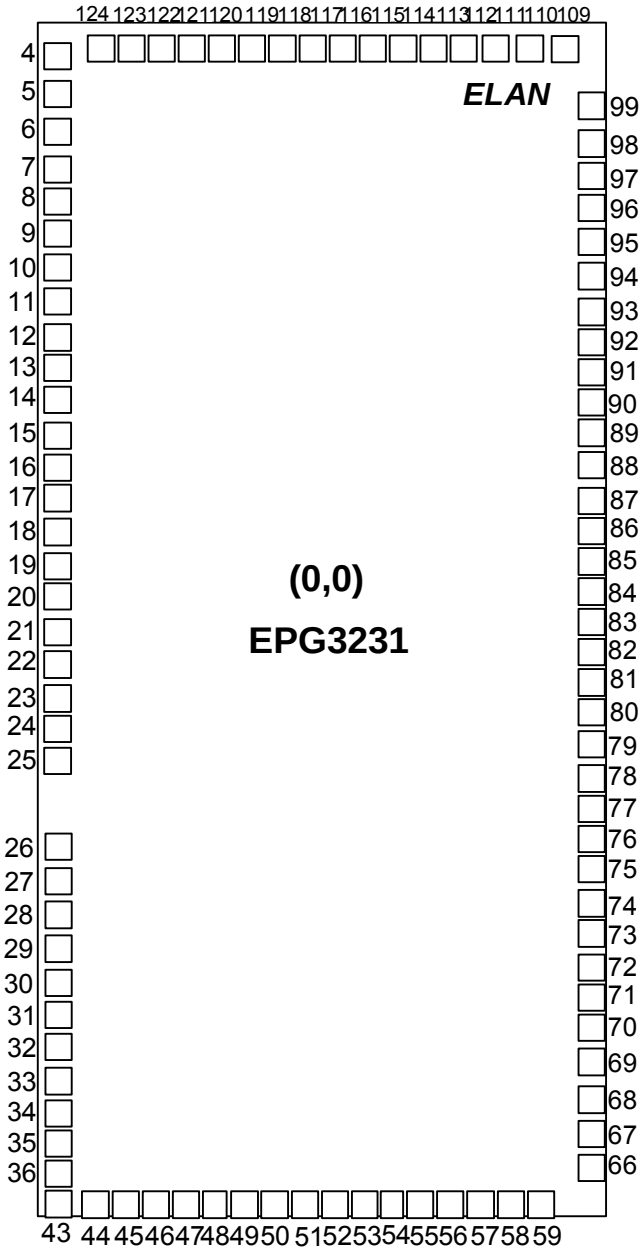
(*4) Carry bit of ADD PCL, A or ADD TABPTRL, A will automatically carry into PCM or TABPTRM.

The Instruction cycle of writing to the PC (program counter) takes 2 cycles.

(*5) S0CALL addressing ability is from 0×000 to 0×FFF (4K space).

(*6) When in SUB operation, borrow flag is indicated by the inverse of the carry bit. That is B=/C.

15 Pad Diagram



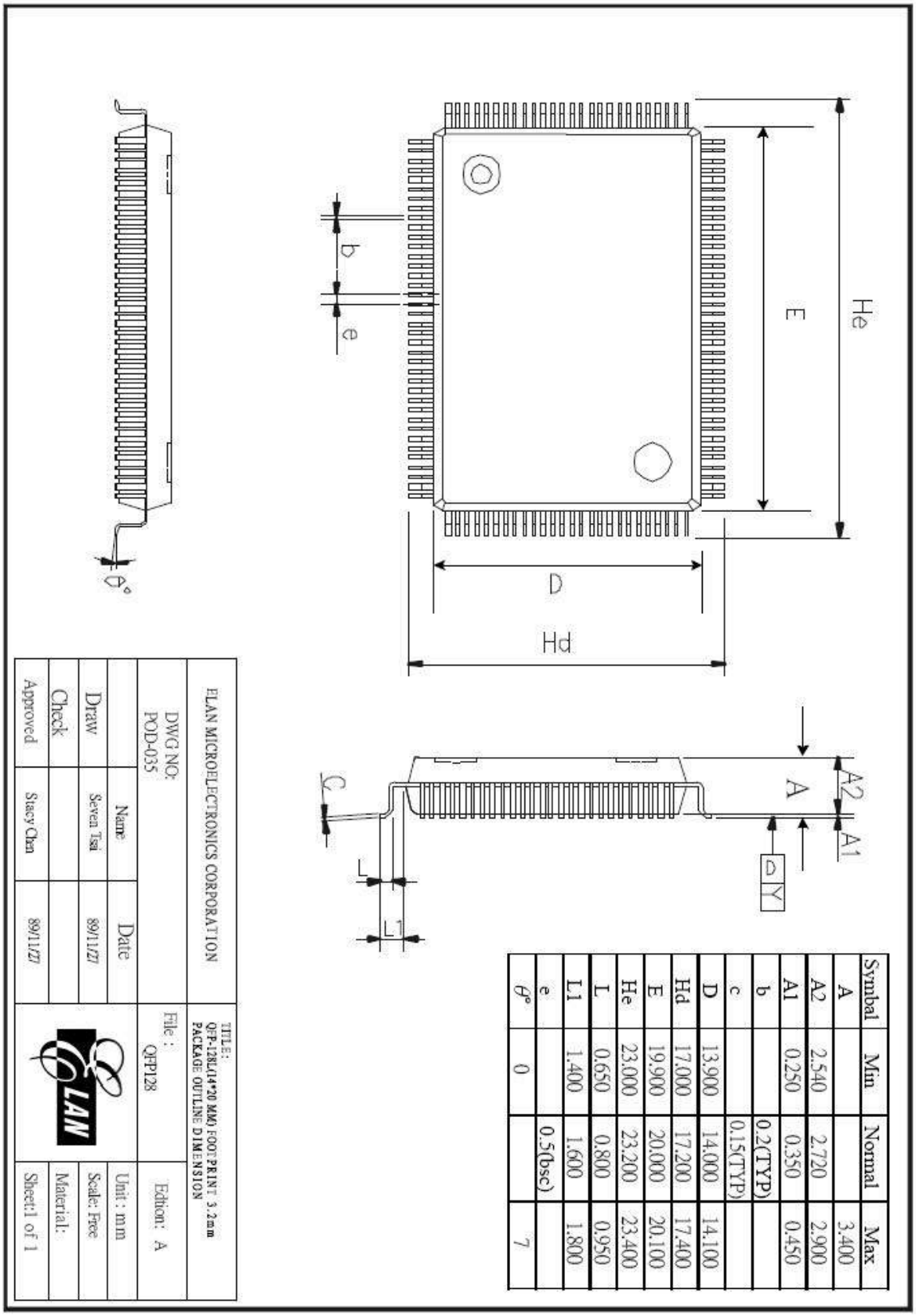
Chip size: 2050 *4180 μm^2

Pin No.	Symbol	X	Y	Pin No.	Symbol	X	Y
1	NC			65	NC		
2	NC			66	PC_4	895.0	-1838.8
3	NC			67	PC_5	895.0	-1718.8
4	PI_3	-895.0	1960.0	68	PC_6	895.0	-1598.8
5	PI_2	-895.0	1830.0	69	PC_7	895.0	-1483.8
6	PI_1	-895.0	1710.0	70	VREX	895.0	-1368.8
7	PI_0	-895.0	1590.0	71	AVDD	895.0	-1258.8
8	PJ_0	-895.0	1472.5	72	PD_7	895.0	-1148.8
9	PJ_1	-895.0	1355.0	73	PD_6	895.0	-1038.8
10	PJ_2	-895.0	1240.0	74	PD_5	895.0	-933.8
11	PJ_3	-895.0	1125.0	75	PD_4	895.0	-828.8
12	PJ_4	-895.0	1010.0	76	PD_3	895.0	-723.8
13	PJ_5	-895.0	895.0	77	PD_2	895.0	-618.8
14	PJ_6	-895.0	782.5	78	PD_1	895.0	-513.8
15	PJ_7	-895.0	670.0	79	PD_0	895.0	-408.8
16	PK_0	-895.0	557.5	80	PE_7	895.0	-303.8
17	PK_1	-895.0	447.5	81	PE_6	895.0	-198.8
18	PK_2	-895.0	337.5	82	PE_5	895.0	-93.8
19	PK_3	-895.0	230.7	83	PE_4	895.0	11.2
20	PK_4	-895.0	125.7	84	PE_3	895.0	116.2
21	PK_5	-895.0	20.7	85	PE_2	895.0	221.2
22	PK_6	-895.0	-84.2	86	PE_1	895.0	326.2
23	PK_7	-895.0	-189.3	87	PE_0	895.0	431.2
24	TEST	-895.0	-294.3	88	PF_7	895.0	536.2
25	OSCI	-895.0	-399.2	89	PF_6	895.0	641.2
26	OSCO	-895.0	-710.2	90	PF_5	895.0	746.2
27	RSTB	-895.0	-815.2	91	PF_4	895.0	851.2
28	AVSS	-895.0	-920.2	92	PF_3	895.0	956.2
29	HOSCI	-895.0	-1030.2	93	PF_2	895.0	1066.2
30	HOSCO	-895.0	-1142.7	94	PF_1	895.0	1176.2
31	PMD	-895.0	-1257.7	95	PF_0	895.0	1286.2
32	PA_0	-895.0	-1372.7	96	PG_0	895.0	1401.2
33	PA_1	-895.0	-1487.7	97	PG_1	895.0	1516.2
34	PA_2	-895.0	-1602.7	98	PG_2	895.0	1636.2
35	PA_3	-895.0	-1717.7	99	PG_3	895.0	1756.2

Pin No.	Symbol	X	Y	Pin No.	Symbol	X	Y
36	PA_4	-895.0	-1835.2	100	NC		
37	NC			101	NC		
38	NC			102	NC		
39	NC			103	NC		
40	NC			104	NC		
41	NC			105	NC		
42	NC			106	NC		
43	PA_5	-895.0	-1960.0	107	NC		
44	PA_6	-768.0	-1960.0	108	NC		
45	PA_7	-648.0	-1960.0	109	PG_4	788.9	1960.0
46	VDD	-533.0	-1960.0	110	PG_5	683.5	1960.0
47	PB_0	-420.5	-1960.0	111	PG_6	579.5	1960.0
48	PB_1	-310.5	-1960.0	112	PG_7	475.5	1960.0
49	GND	-200.5	-1960.0	113	PH_0	371.5	1960.0
50	PB_2	-100.5	-1960.0	114	PH_1	267.5	1960.0
51	PB_3	-0.5	-1960.0	115	PH_2	163.5	1960.0
52	PB_4	99.5	-1960.0	116	PH_3	59.5	1960.0
53	PB_5	199.5	-1960.0	117	PH_4	-44.5	1960.0
54	PB_6	299.5	-1960.0	118	PH_5	-148.5	1960.0
55	PB_7	399.5	-1960.0	119	PH_6	-252.5	1960.0
56	PC_0	499.5	-1960.0	120	PH_7	-356.5	1960.0
57	PC_1	605.5	-1960.0	121	PI_7	-460.5	1960.0
58	PC_2	711.5	-1960.0	122	PI_6	-564.5	1960.0
59	PC_3	817.5	-1960.0	123	PI_5	-668.5	1960.0
60	NC			124	PI_4	-772.5	1960.0
61	NC			125	NC		
62	NC			126	NC		
63	NC			127	NC		
64	NC			128	NC		

For PCB layout, the IC substrate must be connected to VSS.

16 Package



Symbol	Min	Normal	Max
A			3.400
A2	2.540	2.720	2.900
A1	0.250	0.350	0.450
b		0.2(TYP)	
c		0.15(TYP)	
D	13.900	14.000	14.100
E	19.900	20.000	20.100
Hd	17.000	17.200	17.400
He	23.000	23.200	23.400
L	0.650	0.800	0.950
L1	1.400	1.600	1.800
e		0.5(bsc)	
θ°	0		7

ELAN MICROELECTRONICS CORPORATION			TITLE: QFP-128L(14*20 MM) FOOTPRINT 3.2mm PACKAGE OUTLINE DIMENSION
DWG NO: POD-035			File : QFP128
	Name	Date	
Draw	Seven Tsai	89/11/27	
Check			
Approved	Stacy Chan	89/11/27	
			Edition: A
			Unit : mm
			Scale: Free
			Material:
			Sheet:1 of 1