ePG3231

RISC II-2G Series Microcontroller

Product Specification

DOC. VERSION 2.5

ELAN MICROELECTRONICS CORP.

November 2005

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

Copyright © 2005 by ELAN Microelectronics Corporation
All Rights Reserved
Printed in Taiwan

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
1821 Saratoga Ave., Suite 250
Saratoga, CA 95070
USA
Tel: +1 408 366-8223
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 Corporation, Ltd.
23/Bldg. #115 Lane 572, Bibo Road
Zhangjiang Hi-Tech Park
Shanghai, CHINA
Tel: +86 021 5080-3866
Fax: +86 021 5080-4600

Contents

Specification Revision History

Doc. Version	Revision Description	Date
2.0	1. Added Code option: melody/speech interrupt and prescaler 2. Added Code option: interrupt priority change 3. Added Code option: current D/A reference option 4. Added function: melody/speech interrupt and prescaler 5. Modified the affected INC and DEC Status. 6. Added 128-pin QFP package Diagram. 7. Added Power Consumption & Soldering Temperature Electrical Characteristic. 8. Modified the Operating temperature range: –10 ~ +70°C. 9. Added the EXTMEM bit description.	2003/12/26
2.1	Modified the Current D/A reference source, Melody interrupt & Interrupt priority description.	2004/02/23
2.2	1. Modified the EPG3231 remark PM board version for code option. 2. Provided selection for UART standard baud rate: 9.83MHz or 14.745MHz.	2004/05/17
2.3	1. Added CPU change operation mode example. 2. Added External ROM speed setting example & formula. 3. Modified the SPI clock source from PLL/2 only. 4. Added I/O pin type circuit diagrams.	2004/07/19
2.4	Added 64-pin 7x7mm LQFP package diagram.	2004/12/13
2.5	1. Added a Note about Sleep and Idle mode. 2. Modified PLLC capacitor range. 3. Modified A/D converter samlpe rate max value 4. Modified 10MHz supply current max value 5. Modified the A/D conversion operation mode.	2005/11/08

1 General Description

The e**PG3231** is an 8-bit RISC MCU which has an embedded 10 bits SAR A/D converter with touch screen controller, two 8-bit timers and one 16-bit general timer with capture and event counter functions, IR generator, watchdog timer, SPI, UART, four melody timers, a PWM and a current D/A. Furthermore, the EPG3231 has an embedded large size user RAM and program memory with supporting external memory in system programming (ISP) function. It is suitable for educational learning tools application requiring high performance and low cost solution.

The MCU core is ELAN's second generation RISC based MCU, 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 applications.

IMPORTANT NOTES

- *Do not use Register BSR (05h) Bit7 ~ Bit5.*
- *Do not use Register BSR1 (07h) Bit7 ~ Bit5.*
- *Do not use Special Register (2Ah).*
- *Don not use Special Register (4Fh).*
- *Do not use Registers JDNZ & JINZ at FSR1 (09h) special register.*
- *Port 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 enable pull-high.*

1.1 ELAN Software Support

Tool for 4-channel Melody or 3-channel Melody + 1-channel Speech.

1.2 Applications

- PDA and computer for kids
- Electronic book
- Dictionary, Data Bank
- Other Educational Learning Aids

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 times)
  Processor mode: FPLL = 2 / [Text: + 80ns (> 3V), 100ns (> 2.7V) or 120ns (> 2.4V)],
  Extended MCU mode: FPLL = 2 / [Text: + 100ns (> 3V), 120ns (> 2.7V) or 140ns (> 2.4V)]
- One Instruction cycle time = 2 × System clock time
- Processor mode, MCU mode and Extended MCU mode selection through the PMD pin
- External PROM/DROM In System Programming (ISP) function supported
- 32K word internal ROM for MCU mode
- Maximum of 256K word external ROM for Processor mode
- 128 bytes un-banked RAM including special and common registers
- 32 ×128 bytes banked RAM
- RAM stack has a maximum of 128 stack levels
- Look-up Table function is fast and highly efficient when teamed up with REPEAT instruction
- Register to Register instruction transfer
- Compare and Branch in one instruction (2 cycles)
- Single Repeat function (max. of 256 repetitions)
- Decimal ADD and SUB instructions
- Full range CALL and JUMP ability (2 cycles)

2.2 Peripheral

- One input port (Port A) and ten general I/O ports. (Port B ~ 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) use as beat counter for Melody function
- 8-bit IR generator
- 8-bit PWM and a 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
- Parallel External memory interface
- 8/16/24-bit SPI. (Serial Peripheral Interface)
- UART (Universal Asynchronous Receiver and Transmitter)
- Melody interrupt & Interrupt priority (by code option)

  "Mode1": (4-channel Melody/Speech Synthesizer) Disable melody interrupt function and Interrupt priority (External > Capture > Speech > Timers 0 ~ 2 > Peripheral).

  "Mode2": (4-channel Melody/Speech Synthesizer) Enable 11-bit melody/speech timer interrupt and Interrupt priority (Timer 0 > Capture > Speech / Melody > External >Timers 1, 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
- Support RC and crystal oscillations as system clock
- PLL is turned on during FAST mode, and controlled by PEN bit when MCU is in SLOW or 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.

3 Block Diagram

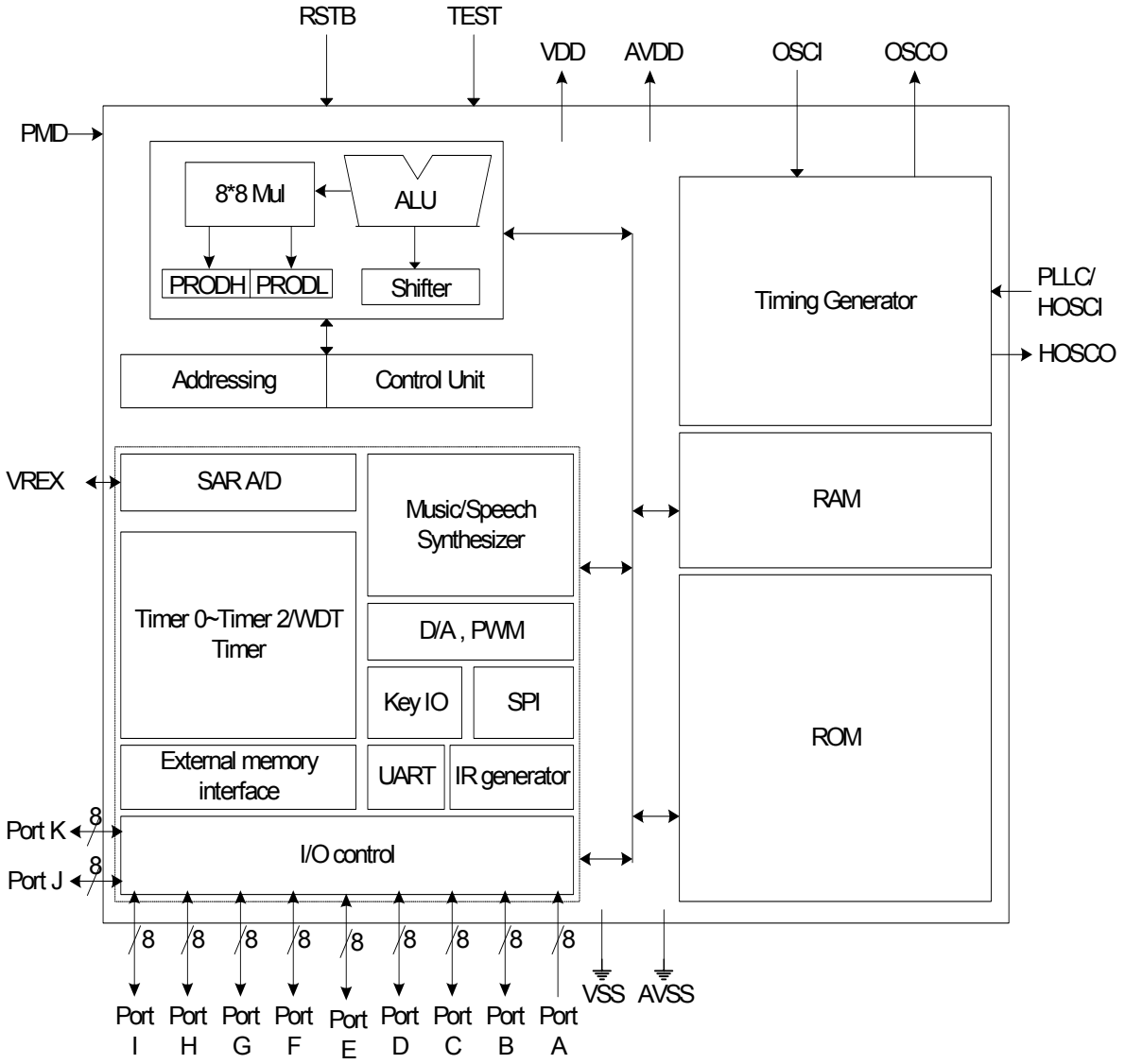


Fig. 3-1 ePG3231 Block Diagram

4 ePG3231 Packaging and Pin Assignment

4.1 128-Pin QFP Package

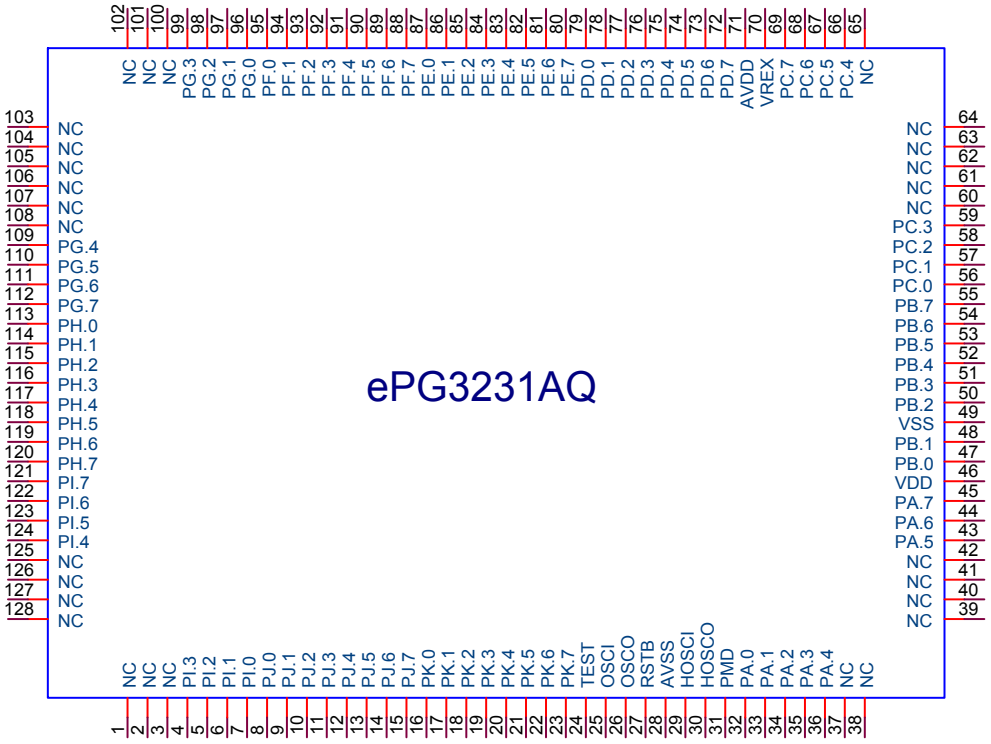


Fig. 4-1 ePG3231 Package: QFP 128 Pins

64-Pin (7 x 7 mm) LQFP Package

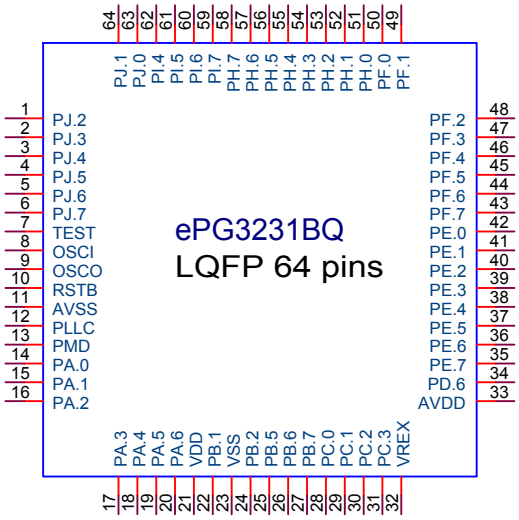
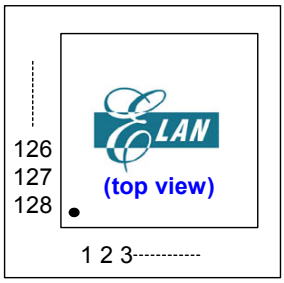


Fig. 4-2 ePG3231 Package: LQFP 64 Pins

4.2 Pin Assignment



AQ No.	BQ No.	ePG3231 Pin Name	AQ No.	BQ No.	ePG3231 Pin Name	AQ No.	BQ No.	ePG3231 Pin Name	AQ No.	BQ No.	ePG3231 Pin Name
1	-	N.C.	33	15	PA.1	65	-	N.C.	97	-	PG.1 (D1)
2	-	N.C.	34	16	PA.2	66	-	PC.4 (ADIN4/YP)	98	-	PG.2 (D2)
3	-	N.C.	35	17	PA.3	67	-	PC.5 (ADIN3/XP)	99	-	PG.3 (D3)
4	-	PI.3	36	18	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	32	VREX	102	-	N.C.
7	-	PI.0	39	-	N.C.	71	33	AVDD	103	-	N.C.
8	63	PJ.0 (Strobe 0)	40	-	N.C.	72	-	PD.7 (OEB)	104	-	N.C.
9	64	PJ.1 (Strobe 1)	41	-	N.C.	73	34	PD.6 (WEB)	105	-	N.C.
10	1	PJ.2 (Strobe 2)	42	-	N.C.	74	-	PD.5 (A21)	106	-	N.C.
11	2	PJ.3 (Strobe 3)	43	19	PA.5	75	-	PD.4 (A20)	107	-	N.C.
12	3	PJ.4 (Strobe 4)	44	20	PA.6 (Rref)	76	-	PD.3 (A19)	108	-	N.C.
13	4	PJ.5 (Strobe 5)	45	-	PA.7	77	-	PD.2 (A18)	109	-	PG.4 (D4)
14	5	PJ.6 (Strobe 6)	46	21	VDD	78	-	PD.1 (A17)	110	-	PG.5 (D5)
15	6	PJ.7 (Strobe 7)	47	-	PB.0 (VO2)	79	-	PD.0 (A16)	111	-	PG.6 (D6)
16	-	PK.0 (Strobe 8)	48	22	PB.1 (VO1/DAO)	80	35	PE.7 (A15)	112	-	PG.7 (D7)
17	-	PK.1 (Strobe 9)	49	23	VSS	81	36	PE.6 (A14)	113	51	PH.0 (D8)
18	-	PK.2 (Strobe 10)	50	24	PB.2 (IROT)	82	37	PE.5 (A13)	114	52	PH.1 (D9)
19	-	PK.3 (Strobe 11)	51	-	PB.3	83	38	PE.4 (A12)	115	53	PH.2 (D10)
20	-	PK.4 (Strobe 12)	52	-	PB.4	84	39	PE.3 (A11)	116	54	PH.3 (D11)
21	-	PK.5 (Strobe 13)	53	25	PB.5 (EVIN/CPIN)	85	40	PE.2 (A10)	117	55	PH.4 (D12)
22	-	PK.6 (Strobe 14)	54	26	PB.6 (UTXD)	86	41	PE.1 (A9)	118	56	PH.5 (D13)
23	-	PK.7 (Strobe 15)	55	27	PB.7 (URXD)	87	42	PE.0 (A8)	119	57	PH.6 (D14)
24	7	TEST	56	28	PC.0 (ADIN8)	88	43	PF.7 (A7)	120	58	PH.7 (D15)
25	8	OSCI	57	29	PC.1 (ADIN7)	89	44	PF.6 (A6)	121	59	PI.7 (SPISDI)
26	9	OSCO	58	30	PC.2 (ADIN6)	90	45	PF.5 (A5)	122	60	PI.6 (SPISDO)
27	10	RSTB	59	31	PC.3 (ADIN5)	91	46	PF.4 (A4)	123	61	PI.5 (SPISCK)
28	11	AVSS	60	-	N.C.	92	47	PF.3 (A3)	124	62	PI.4 (/SPISS)
29	12	HOSCI /PLLC	61	-	N.C.	93	48	PF.2 (A2)	125	-	N.C.
30	-	HOSCO	62	-	N.C.	94	49	PF.1 (A1)	126	-	N.C.
31	13	PMD	63	-	N.C.	95	50	PF.0 (A0)	127	-	N.C.
32	14	PA.0	64	-	N.C.	96	-	PG.0 (D0)	128	-	N.C.

5 Pin Description

Name	I/O/Power Function	Description
VDD VSS	P	Digital power supply. The range is from 2.2V to 3.6V. Connect VSS through a capacitor (0.1μF).
AVDD AVSS	P	Analog power supply. The range is from 2.2V to 3.6V. Connect AVSS through a capacitor (0.1μF).
RSTB	I	System reset. Input with built-in pull up resistor (100K ohms typical). Low: RESET asserted High: RESET released
TEST	I	Normally connected to VSS. Reserved for testing purposes.
OSCI/RC OSCO	I O I/O	RC or Crystal oscillator selection is by Code Option. 32768 Hz oscillator pin. Should be connected to VSS through a capacitor (20pF). RC oscillator connector pin. Should be connected to VDD through resistor (2MΩ).
HOSCI/PLLC HOSCO	I O	Crystal or PLL selection is by Code Option. PLL capacitor connector pin. Should be connected to VSS through 0.01~0.047μF capacitor. High frequency Chrystal oscillator pin. Should be connected to VSS through a capacitor (20pF).
PMD	I	Processor mode and MCU/Extended MCU mode selection pin. Connect to VDD or VSS through a resistor (100KΩ). 0: MCU mode / Extended MCU mode 1: Processor mode
VREX	I/O	External or internal reference voltage for A/D converter. Connect to VSS through a 0.1μF capacitor.
Port A	I I	General input port for special functions, i.e., Wake-up, Interrupt, & Key matrix input. Bit 6: D/A reference resistor
Port B	I/O I O I O O O	General Input/Output port Bit 7: UART Rx pin Bit 6: UART Tx pin Bit 5: Event counter/Capture input pin Bit 2: IR output pin Bit 1: PWM or Current D/A output pin Bit 0: PWM output pin
Port C	I/O O O I I I I I I	General Input/Output port 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: A/D input Channel 5 Bit 2: A/D input Channel 6 Bit 1: A/D input Channel 7 Bit 0: A/D input Channel 8
Port D	I/O O I/O O	General Input/Output port Bit 7: Extended PROM/DROM low active output enable Bit 6: External memory interface write enable pin Bit 5~0: Processor mode or External memory interface address [A21:A16]
Port E	O	General Input/Output port Bit 7~0: Extended MCU mode/Processor mode or External memory interface address [A15:A8].
Port F	O	General Input/Output port. Bit 7~0: Extended MCU mode/Processor mode or External memory interface address [A7:A0].

Name	I/O/Power Function	Description
Port G	I I/O	General Input/Output port 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	General Input/Output port Bit 7~0: Extended MCU mode/Processor mode data bus [D15:D8] Bit 7~0: External memory interface data [D15:D8]
Port I	I/O I O I/O I O	General Input/Output port Bit 7: Serial data input pin Bit 6: Serial data output pin Bit 5: Serial clock input/output pin Bit 4: /Slave Select pin Bit 3~0: High drive output pins
Port J	I/O O	General Input/Output port Bit 7~0: Key Strobe 7 ~ 0 pins
Port K	I/O O	General Input/Output port Bit 7~0: Key Strobe 15 ~ 8 pins

6 Code Option

- Oscillator (OSCSEL): Select “RC” oscillator or “Crystal” oscillator
- Initial mode after reset: Select “Slow” mode or “Fast” mode
- High frequency system clock (HFSEL):
  - Select “PLLC” mode, or
  - “Crystal” mode

  NOTE
  Applicable to PMEPG32A or PMEPG32B processor module only.
- External PROM instruction speed:
  - “Fsystem/2”: Instruction cycles per 2 system clocks
  - “Fsystem/4”: Instruction cycles per 4 system clocks
  - “Fsystem/6”: Instruction cycles per 6 system clocks

  NOTE
  Applicable to PMEPG32A or PMEPG32B processor module only.
  - “Fsystem/8”: Instruction cycles per system 8 clock.
- Current D/A reference source:
  - Select “D/A internal reference voltage,” or
  - “A/D Vrex and Port A.6 Rref”

  NOTE
  Applicable to PMEPG32A or PMEPG32B processor module only.

- Melody interrupt & Interrupt priority:
  - "**Mode 1**": Disable melody interrupt function, and Interrupt priority (External > Capture > Speech > Timers 0 ~ 2 > Peripheral).
  - "**Mode 2**": Enable melody interrupt function & Interrupt priority (Timer 0 > Capture > Speech/melody > External > Timers 1, 2 > peripheral)

NOTE
Applicable to PMEPG32A or PMEPG32B processor module only.

- UART standard baud rate:
  - Select "PLL frequency is 9.83MHz," or
  - "PLL frequency is 14.745MHz"

NOTE
Applicable to PMEPG32A or PMEPG32B processor module only.

- Port C.7 function selection bit: Select "XN for touch panel" or "General I/O function"
- Port C.6 function selection bit: Select "YN for touch panel" or "General I/O function"
- Port C.5 function selection bit: Select "XP for touch panel/ADIN3" or "General I/O function"
- Port C.4 function selection bit: Select "YP for touch panel/ADIN4" or "General I/O function"
- Port C.3 function selection bit: Select "ADIN5" or "General I/O function"
- Port C.2 function selection bit: Select "ADIN6" or "General I/O function"
- Port C.1 function selection bit: Select "ADIN7" or "General I/O function"
- Port C.0 function selection bit: Select "ADIN8" or "General I/O function"
- External memory size:

External Memory Size	Ports as Address Bus	Ports as General I/O
64KW	Port E:Port F	Port D[5:0]
128KW	Port D[0]:Port E:Port F	Port D[5:1]
256KW	Port D[1:0]:Port E:Port F	Port D[5:2]
512KW	Port D[2:0]:Port E:PortF	Port D[5:3]
1MW	Port D[3:0]:Port E:Port F	Port D[5:4]
2MW	Port D[4:0]:Port E:Port F	Port D[5]
4MW	Por t D[5:0]:Port E:Port F	×

- DAC and PWM function selection bits:

DAC or PWM Function Selection	Port B.0 and Port B.1 Function
"DAC is admitted to be used"	Port B.1 is DAO for D/A; Port B.0 is general I/O
"PWM is admitted to be used"	Port B.1 is VO1 and Port B.0 is VO2 for PWM
"DAC and PWM are prohibited", not for use	General I/O

7 Function Description

7.1 Reset Function

A RESET can be triggered by:

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

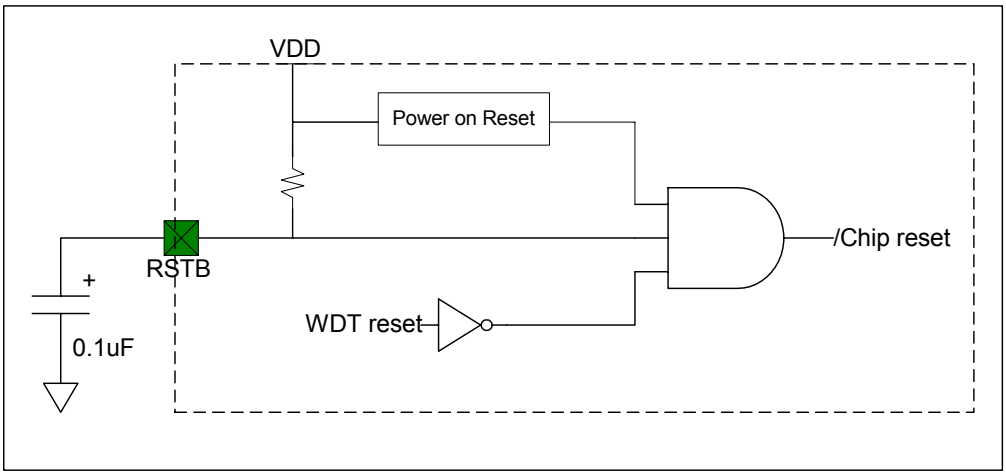


Fig. 7-1 On-chip RESET Circuit

7.1.1 Power–up and Reset Timing

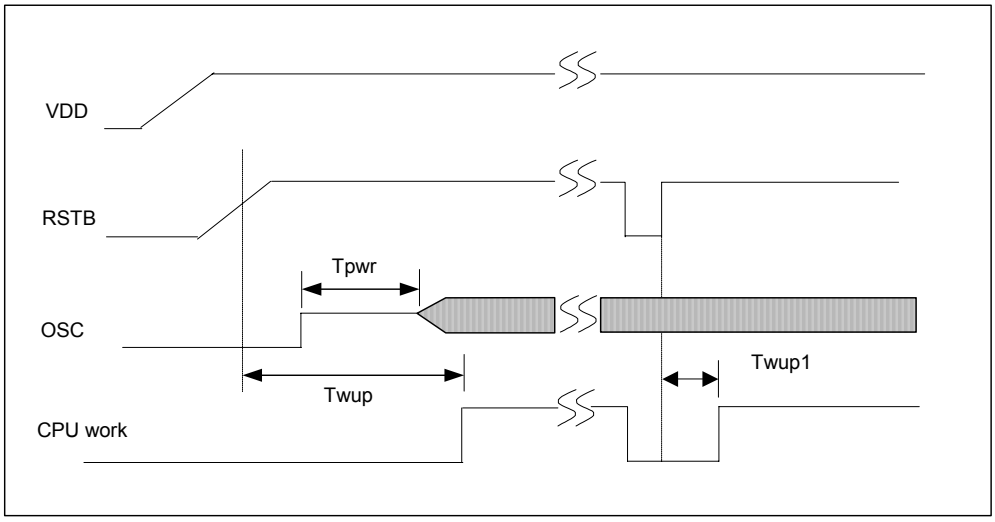


Fig. 7-2 Power-up and RESET Timing

Symbol	Characteristics	Min.	Typical	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

7.1.2 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)
Under SUB operation, borrow flag is indicated by the inverse of 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 (Bit7).

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

Bit 5 (SGE): Computation result is greater than or equal to zero (positive value) after signed arithmetic. Only affected by HEX arithmetic instruction.

NOTE

1. *When OV=1 after signed arithmetic, you can check SGE bit and SLE bit to verify whether overflow (carry into sign bit) or underflow (borrow from sign bit) occurred.*
   If OV=1 and SGE=1 → overflow occurred.
   If OV=1 and SLE=1 → underflow occurred.
2. *When overflow took place, you should clear the MSB of Accumulator to get the correct value.*
   When underflow took place, you should set the MSB of accumulator 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 enter 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 reset occurs, special function register is reset to initial value except for the /TO and /PD bits of 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 SLEEP mode)
1	0	Reserved.
1	1	Power up or RSTB pin low condition

7.1.3 Register Initial Values

Special Register:

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	cu×× ××××[3]	1Fh	PORT I	xxxx xxxx

Control Register:

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 physical register

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

[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 remains unchanged.

7.2 Oscillator System Function

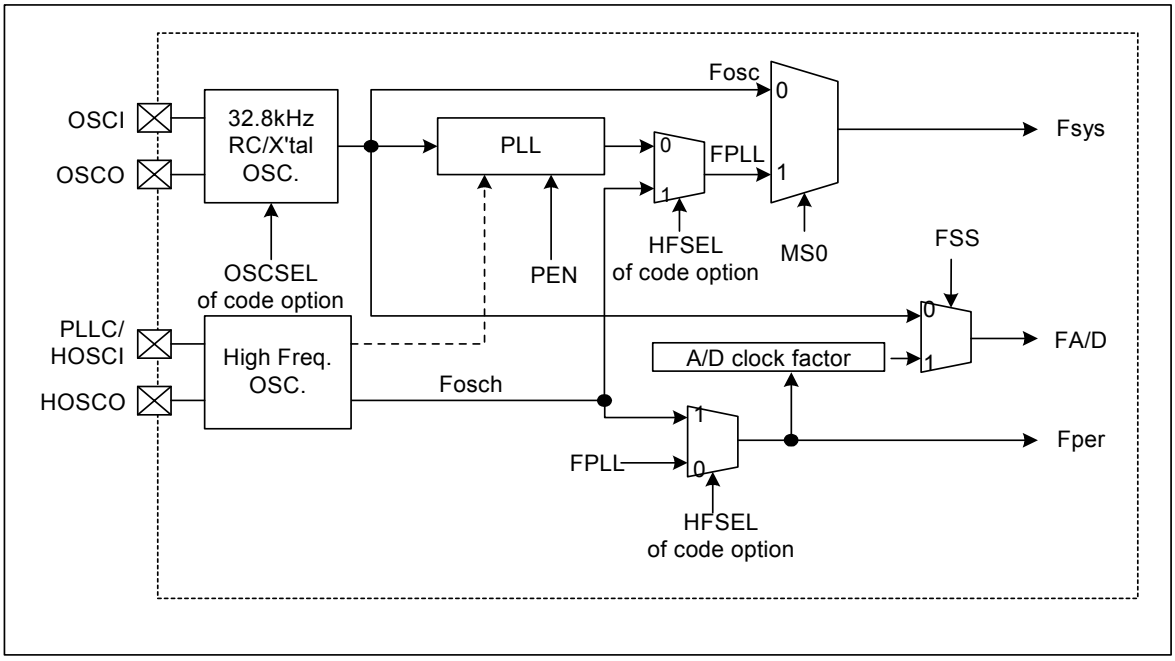


Fig. 7-3 Oscillator System Function Block Diagram

7.2.1 32.768kHz Crystal or RC Oscillator

With a 32.768kHz RC oscillator, a pull-up resistor (2MΩ) must be connected to OSCI pin and the OSCO pin should be floating.

With a 32.768kHz Crystal oscillator, a crystal should be connected between OSCI pin and OSCO pin. The OSCI and OSCO pins are then connected to ground through a 20pF capacitor respectively.

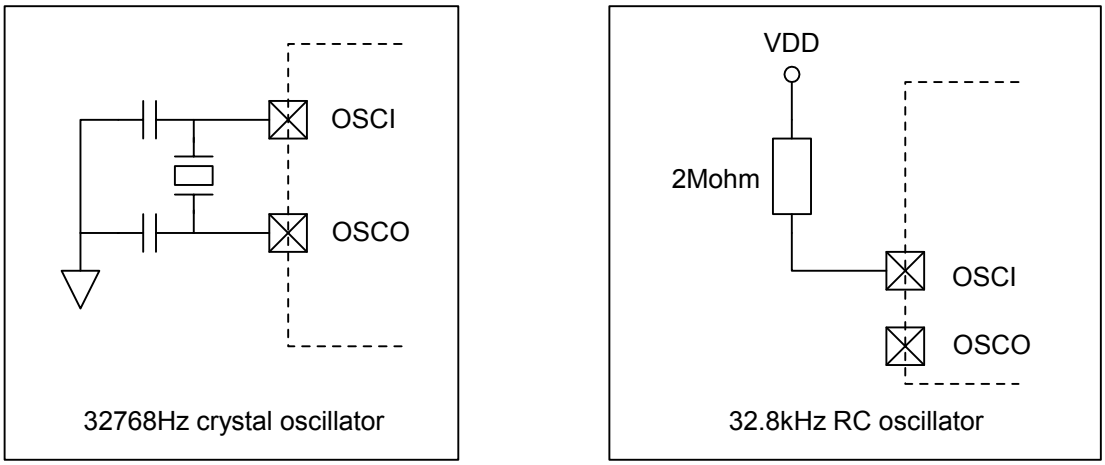


Fig. 7-3 Main Crystal and RC Oscillator Circuit Diagram

7.2.2 High Frequency Oscillator

"HFSEL" code option is available in the high frequency oscillator system. The "HFSEL" code option selects the PLL or crystal oscillator**:**

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

1: High frequency system clock is from HOSCI pin (PLLC/HOSCI pin is used as HOSCI pin).

7.2.2.1 Crystal Oscillator

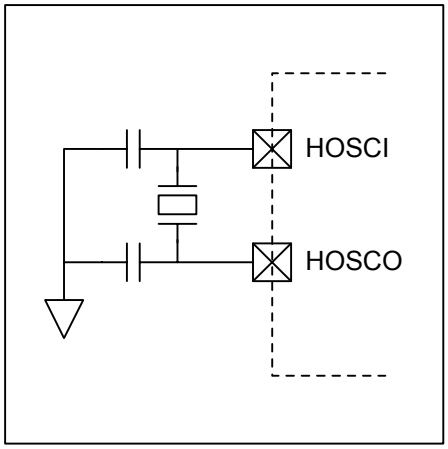


Fig. 7-4 Hi-Frequency Crystal Oscillator Circuit Diagram

- **PLLF (R3Ch):** Reserved
- **PFS (R20h):** System clock can be selectable by PFS register. The initial value of PFS register after a chip reset is "××××0000B" (FPLL = Fosch).

PFS Register	Factual (Hz)
0	Fosch
1 ~ 15	Fosch/2 / PFS
16 ~ 255	Reserved

7.2.2.2 Phase Locked Loop (PLL)

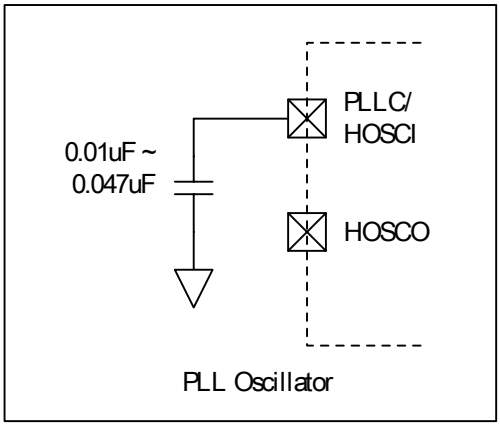


Fig. 7-5 Hi-Frequency PLL Oscillator Circuit Diagram

- **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 Chip reset is "20h" (FPLL=2.097 MHz).

$$Ft\arg et = 2 \times PFS \times Fosc$$

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

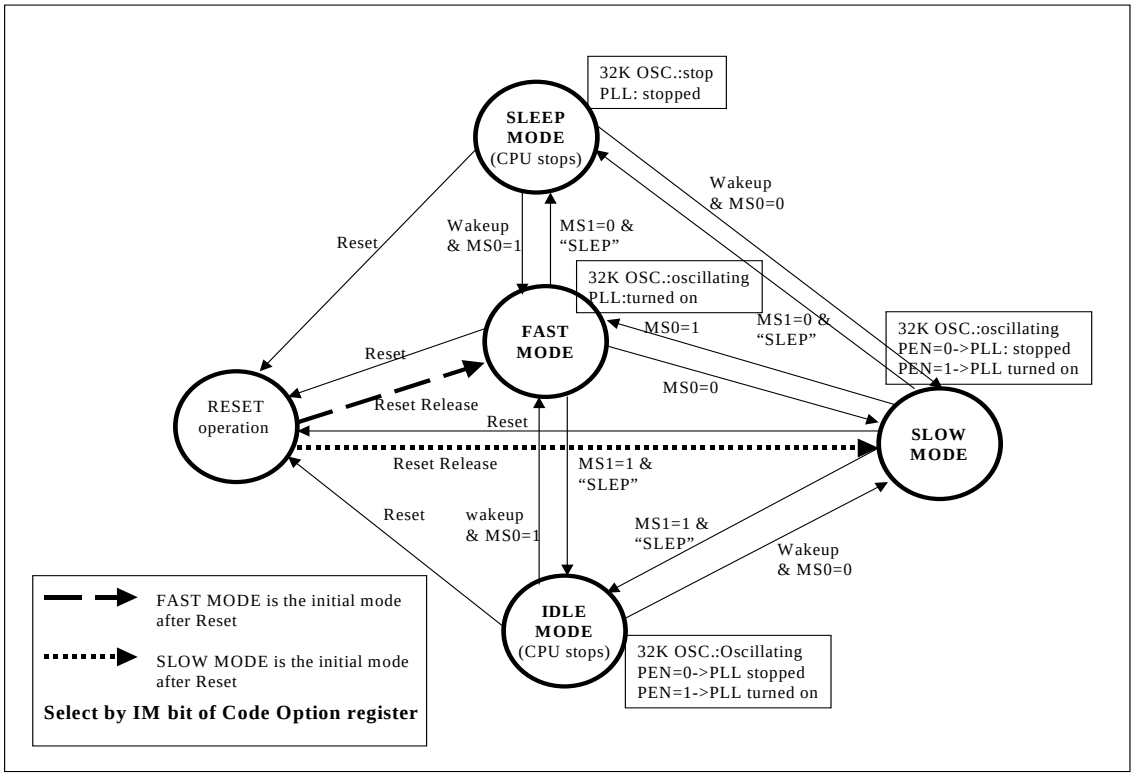


Fig. 7-6 MCU Operation Block Diagram

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

Mode / Device	SLEEP	IDLE	SLOW	FAST
OSC (32768Hz)	×	O	√	√
Fsys	×	×	From OSC	From PLL
PLL	×	√	√	√
A/D conversion	×	√ [2]	√	√
Timers 0~2, IR generator	×	√	√	√
INT	× [1]	× [1]	√	√
SPI	√ (slave)	√ (slave)	√	√
UART	×	×	×	√
Melody Synthesizer	×	×	×	√
PWM, current D/A	×	×	√	√

Legend: " √ " = function is available if enabled " × " = function is Not available

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

[2] It is recommended to operate the A/D converter in IDLE mode to lower the noise couple from the MCU clock.

7.3.1 Timing Diagram

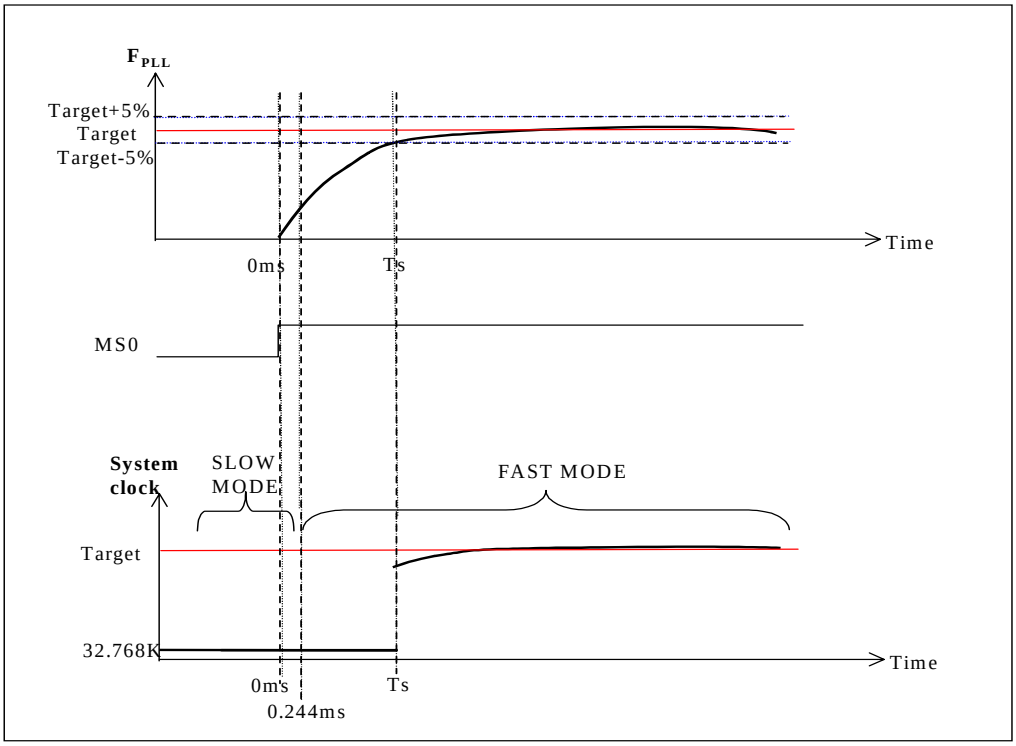


Fig. 7-7 MCU Operation Timing Diagram

NOTE

- *Slow mode will switch to Fast mode at Time=0ms*
- *The System clock will switch into FPLL after 8 oscillations clocks, and frequency will then increase to about hundreds of kHz.*
- *The PLL frequency will be stable (±5%) at Time=Ts (around 2ms~5ms).*

7.3.2 MCU Operation CPUCON (R0Eh)

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

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.

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.

PEN: High Frequency enable bit. This bit is only effective when MCU is in IDLE or SLOW mode.

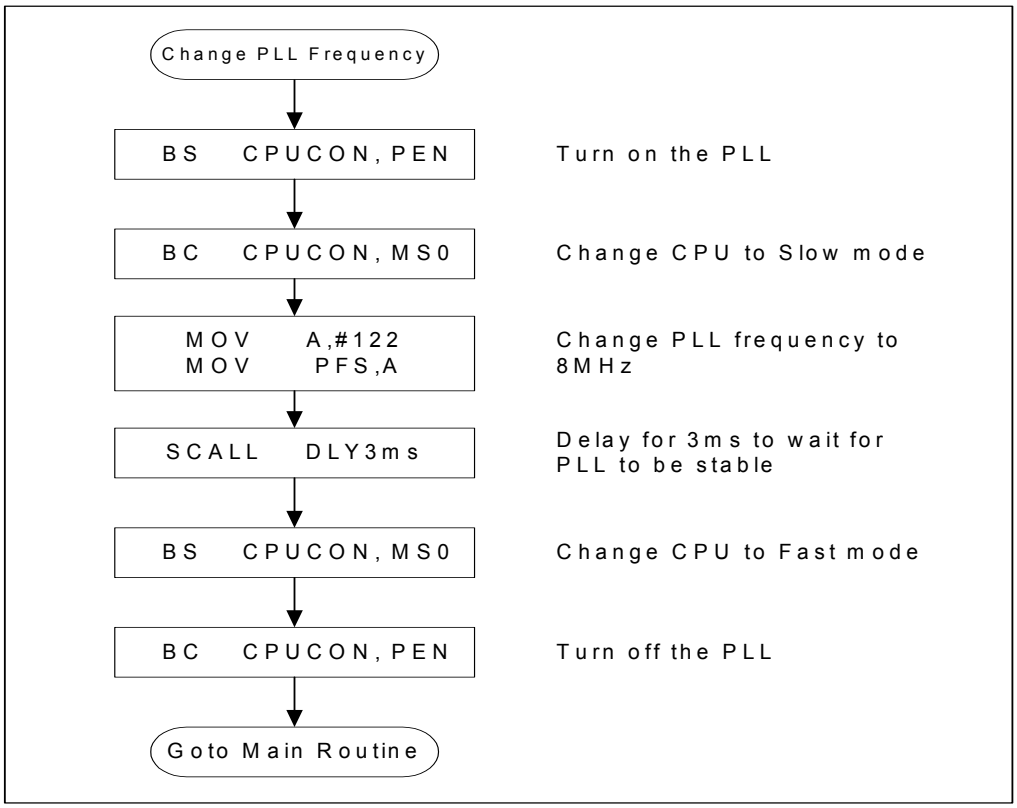
MCU Mode	PEN Bit	PLL On/Off
SLEEP	×	Off
IDLE/SLOW	0	Off
	1	On
FAST	×	On

7.4 Wake-up Function:

Mode / Device	SLEEP	IDLE	SLOW	FAST
I/O wake up	√	√	×	×
Touch panel wake up	√	√	×	×
Timer 1 wake up	×	√	×	×
A/D wake up	×	√	×	×
SPI wake up	√ (Slave)	√ (Slave)	×	×

Legend: √ = Function available if enabled ×: Function NOT available.

7.4.1 Flowchart



7.4.2 Code Examples

Idle & Slow mode change to FAST mode

```
BC      CPUCON,MS0
BS      CPUCON,PE
MOV     ,#122       ;8MHz
MOV     PFS,A
Delay 3msec
BS      CPUCON,MS0
BC      CPUCON,PE
```

Fast mode change to Slow mode

```
BC      CPUCON,MS0
```

Fast & Slow mode 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 mode 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
```

IMPORTANT !

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 (Bit 2) of CPUCON register is reset to 0. It disables all interrupts, including LEVEL1 ~ LEVEL5 (~ LEVEL6 for Mode 2). Setting this bit to '1' will enable all un-mask interrupts.

The interrupt priority has two different sequences (Melody interrupts and Interrupt priority) that are selected by code option.

"Mode 1": Disable melody interrupt function and Interrupt priority (External > Capture > Speech > Timers 0 ~ 2 > Peripheral).

Interrupt Level	Interrupt Source	Start Address	Remarks
	RESET	0x00000	
Level 1	Input Port A	0x00002	PAINT, PIRQB
Level 2	Capture	0x00004	CPIF
Level 3	Speech Timer	0x00006	SPHTI
Level 4	Timers 0~2	0x00008	TMR0I, TMR1I, TMR2I
Level 5	Peripheral	0x0000A	UERRI, UTXI, URXI, ADIF, SRBFI

"Mode 2": Enable melody interrupt function and Interrupt priority (Timer 0 > Capture > Speech/Melody > External > Timers 1, 2 > Peripheral).

Interrupt Level	Interrupt Source	Start Address	Remarks
	RESET	0x00000	
Level 1	Timer 0	0x00008	TMR0I
Level 2	Capture	0x00004	CPIF
Level 3	Speech & melody Timer	0x00006	SPHTI (MTI)
Level 4	Input Port A	0x00002	PAINT, PIRQB
Level 5	Timer 1, 2	0x00008	TMR1I, TMR2I
Level 6	Peripheral	0x0000A	UERRI, UTXI, URXI, ADIF, SRBFI

Code Example:

```
; *****     Reset program
ResetSEG  CSEG   0X00
          LJMP   MSTART     ;(0X00) Initialize
          LJMP   INPTINT    ;(0X02) Input Port & Touch Panel Interrupt
          LJMP   CAPINT     ;(0X04) Capture Input Interrupt
          LJMP   SPHINT     ;(0X06) Speech/Melody Timer Interrupt
          LJMP   TIMERINT   ;(0X08) Timer-0,1,2 Interrupt
          LJMP   PERIPH     ;(0X0A) Peripheral Interrupt

PgmSEG    CSEG   0X20
```

```
; --- 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-) connect 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
; --- PortA 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, the Capture Interrupt takes place.

Code Example:

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

```
; --- Capture input interrupt
toCAPINT:
        BS      INTSTA,CPIF
          :
        SJMP    POP
```

7.5.3 Speech / Melody Timer Interrupt (Enable Melody Interrupt Function)

Speech Timer (or SPHTI) and the three Melody Timers (MT1, MT2, MT3) are quipped with interrupt function, while MT4 has no such function. Speech/Melody Timer is an 11 bits timer for time count. When the Speech/Melody Timer count value underflows, interrupt occurs, and the SPHTRL/MTRL value reloads to count value.

Code Example:

```
; === Speech/melody Timer Interrupt
SPHINT:
        S0CALL  PUSH
; --- Melody Bank 0
        MOV     A,#11111000B
        AND     SFCR,A
        JBS     MTCON,MTI,toMINT0
; --- Melody Bank 1
        INC     SFCR
        JBS     MTCON,MTI,toMINT1
; --- Melody Bank 2
        INC     SFCR
        JBS     MTCON,MTI,toMINT2
; --- Speech bank
        BS      SFCR,SPHSB
        JBS     SPHTCON,SPHTI,toSPHINT
        SJMP    POP
```

```
; --- To melody timer interrupt
toMINT0:
          :
        SJMP    Q_MINT
toMINT1:
          :
        SJMP    Q_MINT
toMINT2:
          :
Q_MINT:
        BC      MTCON,MTI
        SJMP    POP
; --- To speech/melody timer interrupt
toSPHINT:
        BC      SPHTCON,SPHTI
          :
        SJMP    POP
```

7.5.4 Speech Timer Interrupt (Disable Melody Interrupt Function):

Speech Timer is an 11 bits timer for time count. When the Speech Timer counts the value underflows, the interrupt occurs and the SPHTRL value reloads to count 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 Timer 0, Timer 1, and Timer 2 Interrupts

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

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

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

Code Example:

```
; === Timer-0, 1, 2 Interrupt
TIMERINT:
        S0CALL  PUSH
        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
```

7.5.6 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, 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 Processor Mode Memory Access

The EPG3231 can operate in different modes where the memory is either on chip or off chip. The three different modes are:

- MCU mode: Only internal PROM is available (PMD=0, No external memory)
- Extended MCU mode: Both internal PROM / external DROM are available (PMD=0, add an external memory)
- Processor mode: Both external PROM & DROM are available.

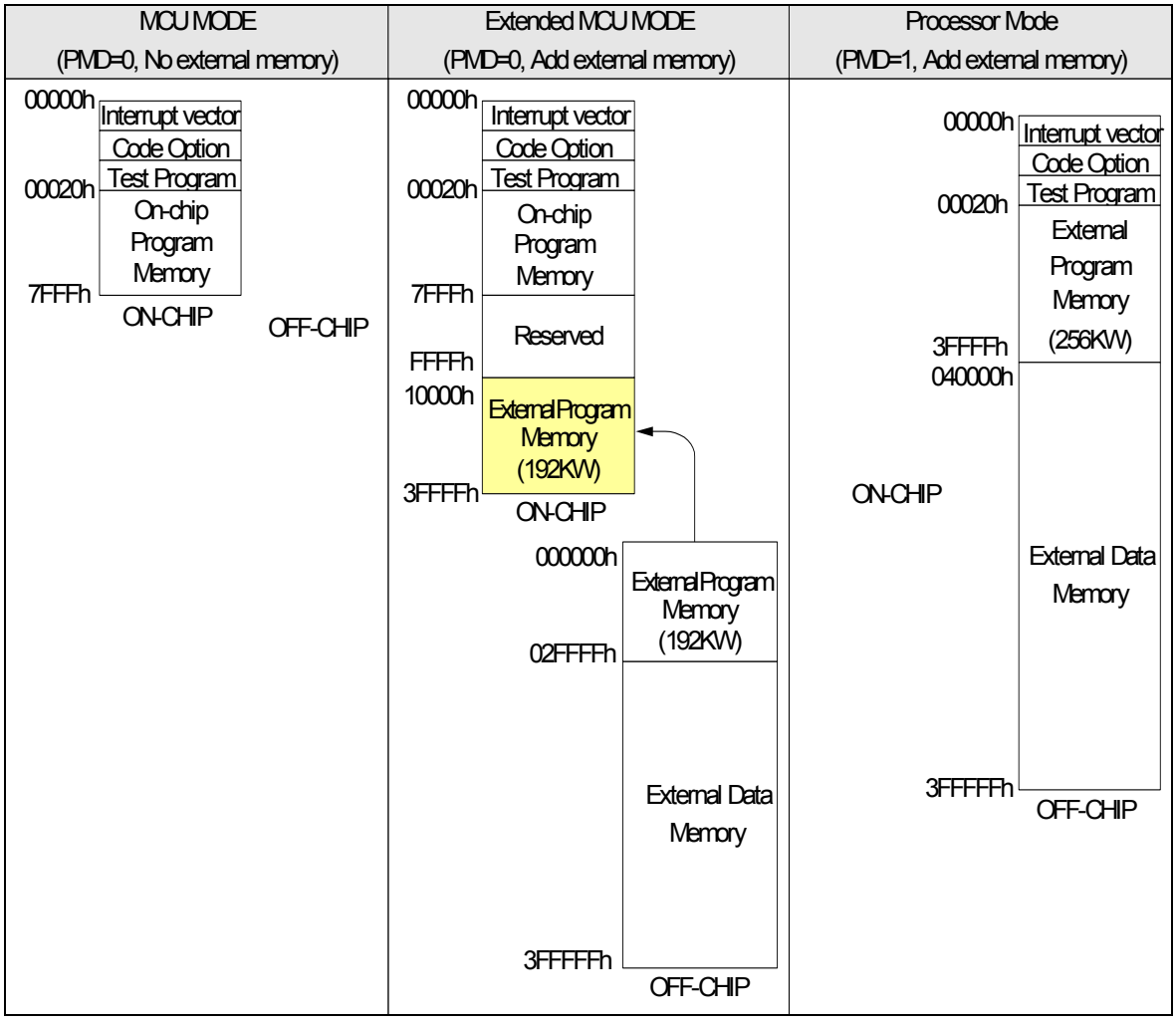


Fig. 7-8 Memory Map in Processor Mode

7.6.1 MCU External Program Memory Access Timing

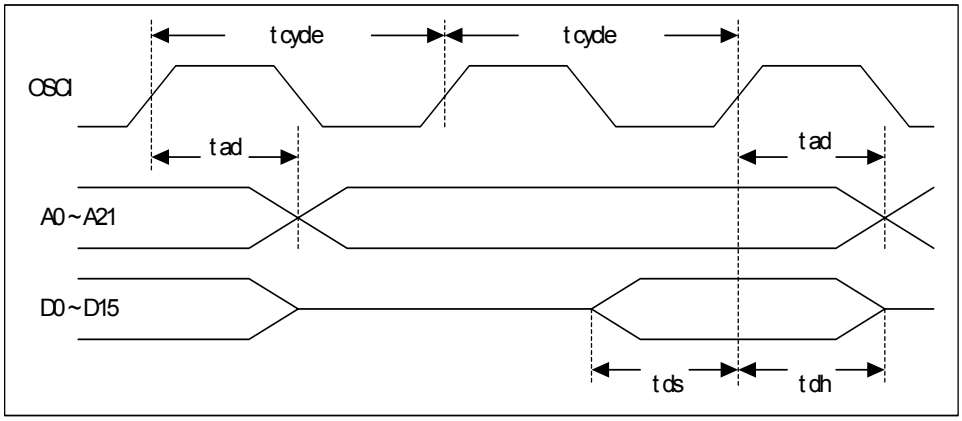


Fig. 7-9 MCU External Program Memory Access Timing Diagram

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*)

Processor mode: FPLL = 2 / [Text. + 80ns (> 3V), 100ns (> 2.7V) or 120ns (> 2.4V)]

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

Example 1: Flash memory access time: 70ns & operating voltage: 3V (Processor mode)
Calculating the PLL maximum speed = 2 / [70ns + 80ns] = 13 MHz.

Example 2: Mask ROM access time: 40ns & operating voltage: 3V (Processor mode)
Calculating the PLL maximum speed = 2 / [40ns + 80ns] = 16 MHz.

Example 3: Flash memory access time: 70ns & operating voltage: 3V (Extended MCU mode)
Calculating the PLL maximum speed = 2 / [70ns + 100ns] = 11 MHz.

7.6.1.1 Processor Mode Circuit

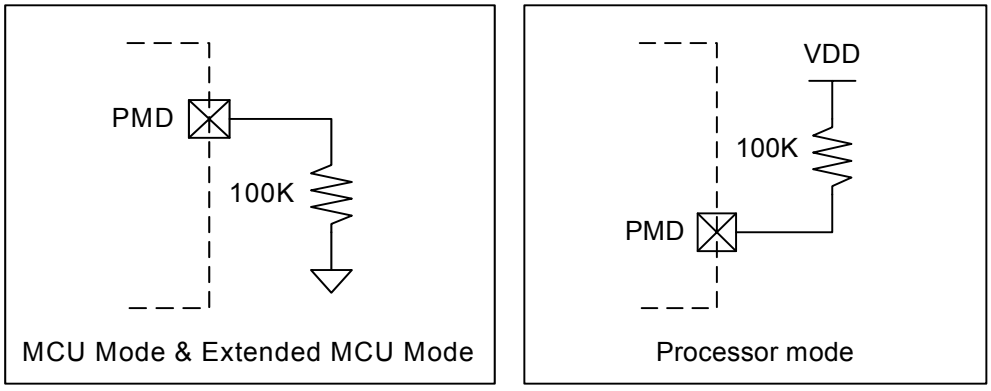


Fig. 7-10 MCU Mode, Extended MCU Mode and Processor Mode Circuit Diagrams

7.7 Program ROM Map

PMD=0 (MCU mode): 8K words x 4 Segments = 32K words (Extended MCU mode): Internal: 8K word x 4 Segments = 32K words External: 8K word x 24 Segments = 192K words				PMD=1 (Processor mode): 8K words x 32 Segments=256K words			
Addr.	**Segment**	**Addr.**	**Segment**	**Addr.**	**Segment**	**Addr.**	**Segment**
0000h | 000Bh	Interrupt Vector (12 words)	20000h | 23FFFh	Segment 16 | Segment 17	0000h | 000Bh	Interrupt Vector (12 words)	20000h | 23FFFh	Segment 16 | Segment 17
000Ch | 000Fh	Code Option (4 words)			000Ch | 000Fh	Code Option (4 words)		
0010h | 001Fh	Test Program (16 words)			0010h | 001Fh	Test Program (16 words)		
0020h | 3FFFh	Segment 0 | Segment 1			0020h | 3FFFh	Segment 0 | Segment 1		
4000h | 7FFFh	Segment 2 | Segment 3	24000h | 27FFFh	Segment 18 | Segment 19	4000h | 7FFFh	Segment 2 | Segment 3	24000h | 27FFFh	Segment 18 | Segment 19
8000h | BFFFh	Not Implement	28000h | 2BFFFh	Segment 20 | Segment 21	8000h | BFFFh	Segment 4 | Segment 5	28000h | 2BFFFh	Segment 20 | Segment 21
C000h | FFFFh	Not Implement	2C000h | 2FFFFh	Segment 22 | Segment 23	C000h | FFFFh	Segment 6 | Segment 7	2C000h | 2FFFFh	Segment 22 | Segment 23
10000h | 13FFFh	Segment 8 | Segment 9	30000h | 33FFFh	Segment 24 | Segment 25	10000h | 13FFFh	Segment 8 | Segment 9	30000h | 33FFFh	Segment 24 | Segment 25
14000h | 17FFFh	Segment 10 | Segment 11	34000h | 37FFFh	Segment 26 | Segment 27	14000h | 17FFFh	Segment 10 | Segment 11	34000h | 37FFFh	Segment 26 | Segment 27
18000h | 1BFFFh	Segment 12 | Segment 13	38000h | 3BFFFh	Segment 28 | Segment 29	18000h | 1BFFFh	Segment 12 | Segment 13	38000h | 3BFFFh	Segment 28 | Segment 29
1C000h | 1FFFFh	Segment 14 | Segment 15	3C000h | 3FFFFh	Segment 30 | Segment 31	1C000h | 1FFFFh	Segment 14 | Segment 15	3C000h | 3FFFFh	Segment 30 | Segment 31

7.8 RAM Map for Special and Control Registers (RAM Size:128 Bytes+ 32 Banks × 128 Bytes=4224 Bytes)

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 [1]				R/W				
		-	-	-	Bank Select Register for INDF0 & General RAM				
6	STKPTR	R/W							
		Stack Pointer							
7	BSR1[1]				R/W				
		-	-	-	Bank Select Register 1 for INDF1				
8	INDF1	R/W							
		Indirect Addressing Pointer 1							
9	FSR1 [2]	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

[1] Do not use BSR (05h) Bit 7 ~ Bit 5 and BSR 1 (07h) Bit 7 ~ Bit 5

[2] Do not use JDNZ & JINZ at FSR1 (09h) special register

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	PORTI	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 Selection 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) [3]								
		-							
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

[3] Do not use special register (2Ah)

Addr	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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
30	UARTCON	W	R/W	R/W	R/W	R/W	R/W	R	R/W
		TB8	UMODE1	UMODE0	BRATE2	BRATE1	BRATE0	UTBE	TXE
31	UARTSTA	R	R/W	R/W	R/W	R/W	R/W	R	R/W
		RB8	EVEN	PRE	PRERR	OVERR	FMERR	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

Addr	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
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							
4A	MTRL/SPHTRL								
		Melody Channels1~4 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 Count Value Register							
4D	TR2C	R							
		Timer 2 Count Value Register							
4E	ADCF	R/W							
		A/D Clock Factor Register							
4F	(Reserved) [4]								
		-							

[4] Do not use special register (4Fh)

■ **Other RAM Un-Banked Registers**

Address	Un-Banked
50h | 7Fh	General Purpose RAM

■ **RAM Banked Registers (Selected by BSR)**

Address	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 (Partial) Detailed Description

7.9.1 STKPTR (R06h): Stock Pointer

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

Stack is located in BANKs 30 and 31 from address FFh ~ 80h. And the initial stack pointer is 00h.

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

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

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

Bit17	Bit16	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCH					PCM								PCL				

Generates up to 256K×16 on chip ROM addresses relative to 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 into the current PC. The carry bit of R2 will automatically carry into PCM and 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
```

7.9.3 ACC (R0Ah): Accumulator

Internal data transfer, or instruction operand holding.

7.9.4 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 nor 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 of FSR0.

Bit 5 (FSR1_ID): Set to '1' means auto increase. Reset to '0' means auto decrease of FSR1.

- **BSR, FSR0, INDF0 (R05h, R01h, R00h):** Indirect Address Pointer 0

  BSR (R05h): Determines which bank should be active (working bank) among the 32 banks (Bank 0 ~ Bank 31).

  FSR0 (R01h): Address register for INDF0. User can select up to 256 bytes (Address: 00 ~ 0FFh).

  INDF0 (R00h): Not a physically implemented register.

- **BSR1, FSR1, INDF1 (R07h, R09h, R08h):** Indirect Address Pointer 1

  BSR1 (R07h): Bank register for INDF1. It cannot determine the working bank for general register.

  FSR1 (R09h): Address register for INDF1. User can select up to 128 bytes (Address: 80 ~ 0FFh). Bit 7 of FSR1 is fixed to '1'.

  INDF1 (R08h): Not a physically implemented register.

The linear address capabilities of INDF1 are as shown in the following diagram:

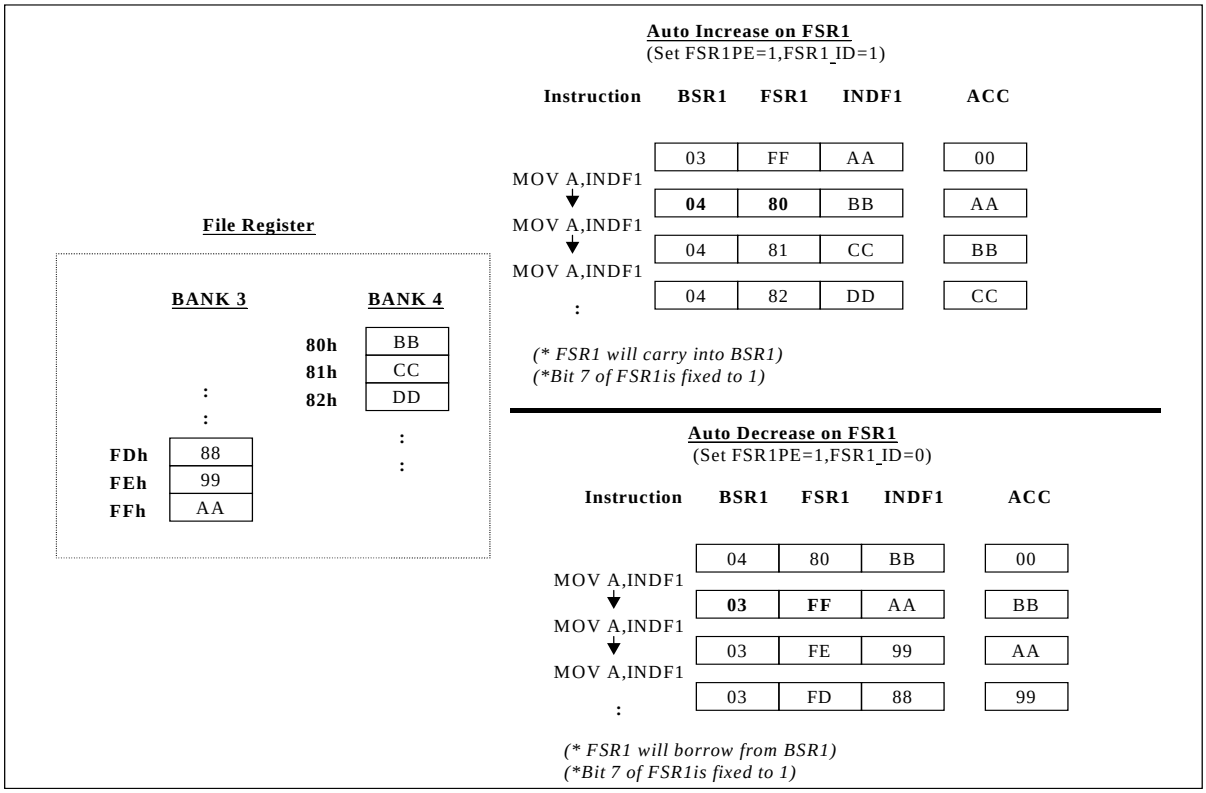


Fig. 7-11 INDF1 Linear Address Capabilities Diagram

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
```

```
Transform data from 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 Bank1
         :
```

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

Bit 23 ~ Bit 16	Bit15 ~ Bit 8	Bit 7 ~ Bit 0
TABPTRH	TABPTRM	TABPTRL

Program ROM or external memory address register.

Bit 23 is used to select internal/external memory.

Bit 22 ~ Bit 1 are used to point the memory address.

Bit 0 is used to select the low byte or high byte of the pointed word (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 ROMConverter
        :
        :
      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
:
```

7.9.6 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.

7.9.7 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" : Multiplier is unsigned
"1" : Multiplier is signed

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

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
```

7.9.8 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 to "0" by software

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

7.9.9 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

7.9.10 Port D, Port E (R1Ah, R1Bh): General I/O or External Memory Address Bus & Control Pin, and 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 & Port E high / low nibble direction control
"0" : Set to output pin
"1" : Set to input pin

Bit 3, Bit 7 (DHNPU, EHNPU) & Bit 2, Bit 6 (DLNPU, ELNPU):
Enable Port D & Port E high / low nibble pull-up resistor
"0" : Disable the pull-up resistor
"1" : Enable the pull up resistor

7.9.11 Port F, Port G (R1Ch, R1Dh): External Memory Address Bus and External Memory Data Bus Registers

- **DCRFG (R37h):** Port F & Port G Direction Control & Pull-up Resistor Control

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 & Port G high / low nibble direction control
"0" : Set to output pin
"1" : Set to input pin

Bit 3, Bit 7 (FHNPU, GHNPU) & Bit 2, Bit 6 (FLNPU, GLNPU):
Enable Port F & Port G high / low nibble pull up resistor
"0" : Disable the pull-up resistor
"1" : Enable the pull-up resistor

7.9.12 Port H, Port I (R1Eh, R38h): External Memory Data Bus and General I/O Registers

■ **DCRHI (R38h):** Port H & Port I Direction Control & Pull-up Resistor Control

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 & Port I high / low nibble direction control.

"0" : Set to output pin

"1" : Set to input pin

Bit 3, Bit 7 (HHNPU, IHNPU) & Bit 2, Bit 6 (HLNPU, ILNPU):

Enable Port H & Port I high / low nibble pull up resistor

"0" : Disable the pull-up resistor

"1" : Enable the pull-up resistor

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

■ **DCRJK (R39h):** Port J & Port K Direction Control & Pull-up Resistor Control

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 & Port K high / low nibble direction control

"0" : Set to output pin

"1" : Set to input pin

Bit 3, Bit 7 (JHNPU, KHNPU) & Bit 2, Bit 6 (JLNPU, KLNPU):

Enable Port J & Port K high / low nibble pull up resistor

"0" : Disable the pull up resistor

"1" : Enable the pull up 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 port 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 Peripherals

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

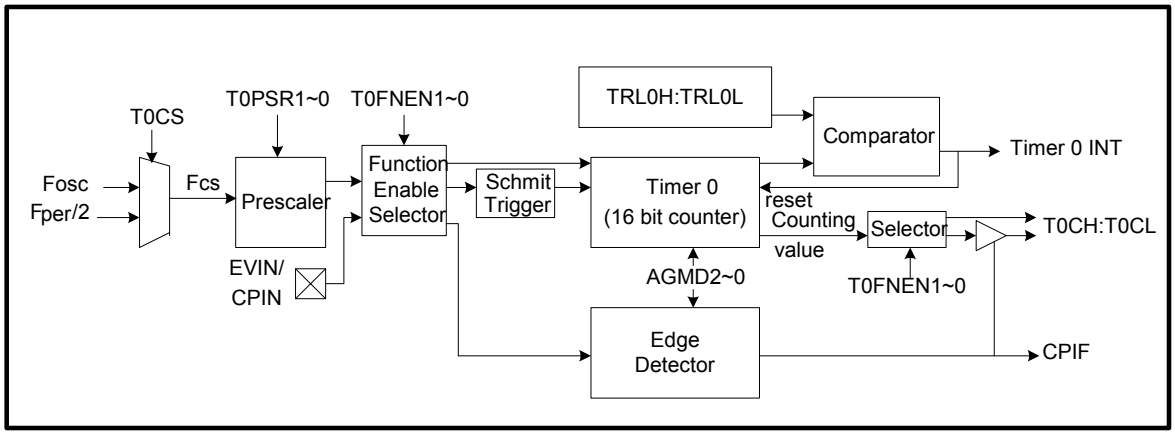


Fig. 8-1 Timer 0 Function Block Diagram

8.1.1 Timer 0 Mode

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

The timer is also equipped with a prescaler. The T0PSR2 ~ 0PSR0 bit of TR01CON register determines the prescaler ratio and generate different clock rates as clock source for the timer. Counter value is increased by one (count up) in accordance with the actual type of timer clock source in use and is 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 counter value is larger than TRL0H: TRL0L value, the Timer0 interrupt will occur, and the counter value is automatically reset to zero.

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

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

8.1.2 Capture Mode: CPIN (Port B.5) Pin

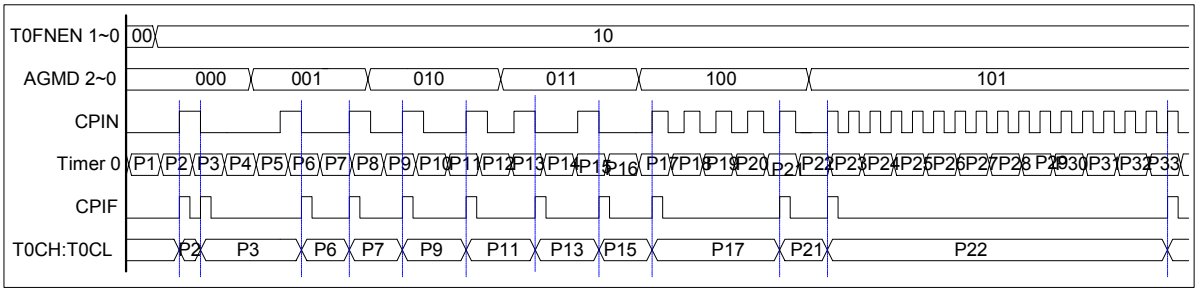
Capture is a function that captures the Timer 0 value when an event occurs on CPIN pin.

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

The CPIN pin should be configured in capture function input by setting the 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:

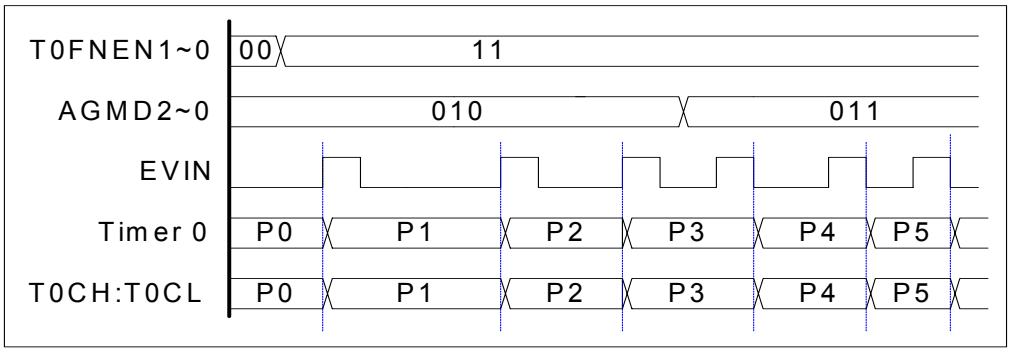


8.1.3 Event Counter Mode: EVIN (Port B.5) Pin

Event counter is a function that allows the 16-bit counter value to increase by one when an event occurs on EVIN pin at every rising edge or every falling edge as selected by AGMD 2~0 bit of SFCR register. In other words, the clock source of Timer 0 is from external event (EVIN pin).

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

Event Counter Mode Example:



8.1.4 Timer 0 Applicable Registers

- **TRL0H, TRL0L (R25h, R24h):**
  Used to store the values compared with Timer 0 register.

- **T0CH, T0CL (R3Eh, R3Dh):**
  Used to store the Timer 0 counter value in Timer 0 mode and Event counter mode. Note that under Capture mode, it is used to store the captured value.

- **TR01CON (R27h):** Timer 0 and Timer 1 Control Registers

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, Capture, and event counter modes selection 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 functions edge detector selection bits

T0FNEN 1 ~ 0	Mode	AGMD 2~0	Edge Mode
00	Disable	-	-
01	Timer 0	-	-
10	Capture	000	1st Rising edge, 2nd falling edge and so on
		001	1st Falling edge, 2nd rising edge and so on
		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. Disable the Timer 0 before changing from one mode to another.
2. To avoid error, setup T0FNEN1 and T0FNEN0 simultaneously.

■ **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-mask 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 0 (TMR0IE): Control bit of Timer 0 interrupt.
"0" : Disable Timer 0 interrupt
"1" : Enable Timer 0 interrupt.

Bit 7 (CPIE): Control bit of Capture interrupt
"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 disable Timer 0

Bit 7 (CPIF): Sets 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) / [4 x 3FFF + 1]
Timer0SR:
          :
        System setting 8MHz
        PC.2 Port H & G setting output port
          :
; --- Fpll & Prescale 1:4
        MOV     A,#00000101B
        MOV     TR01CON,A
; --- 4ms = (4 x 16383 + 1) / (8M/2)
        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
```

```
; === 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 and so on
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     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
          Port 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)

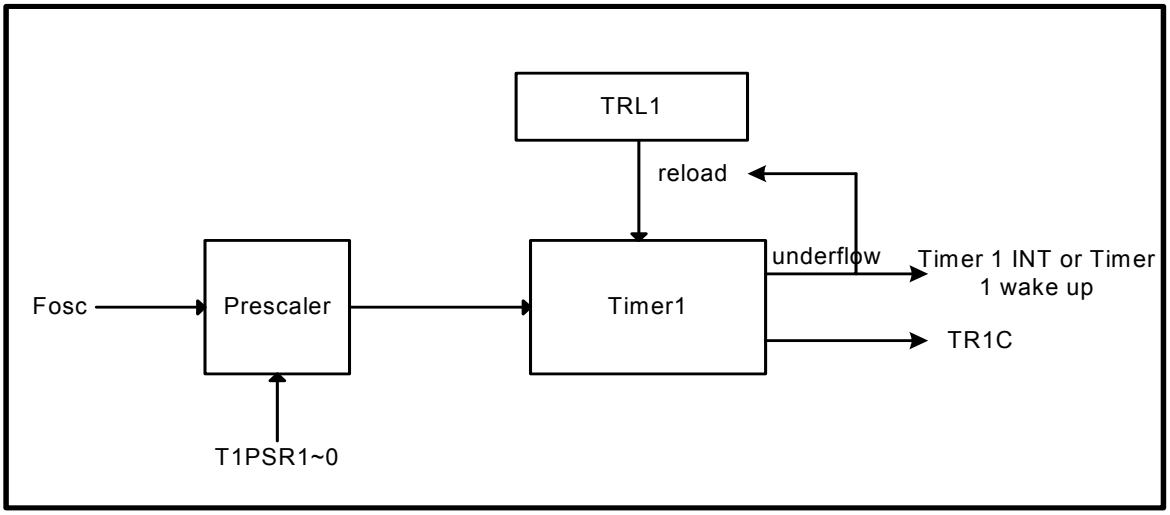


Fig. 8-2 Timer 1 Function Block Diagram

Timer1 is a general-purpose 8-bit countdown counter for applications that require time count. Furthermore, Timer 1 offers interrupt and wake-up functions for user's convenience. The clock source is from the oscillator clock (Fosc).

The Timer also offers a prescaler, where the T1PSR1~T1PSR0 bits of TR01CON register determine the prescaler ratio while generating different clock rates as clock source for the timer. Setting T1WKEN bit of TR01CON register to "1," will enable Timer 1 to underflow wake up function under IDLE MODE.

Counter value will decrease by one (countdown) according to the frequency of the timer clock source. When the counter value underflows, the timer interrupt will trigger if the global interrupt and Timer1 interrupt are both enabled. At the same time, TRL1 will automatically reload into 8 bits counter.

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

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

8.2.1 Timer 1 Applicable Registers

- **TRL1 (R26h):** Used to store the auto-reload value of Timer 1. When enabling Timer 1 or when underflow occurs, TRL1 register will automatically reload into 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 for Timer 1 underflow 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): Control bit of Timer 1 interrupt
"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	Bit 1	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
; === Timer1 = 32.768K / [256 x 3F + 1]
Timer1SR:
         :
        PC.2 setting output port
         :
        MOV     A,#10110000B
        MOV     TR01CON,A       ; Fosc & Prescale 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)

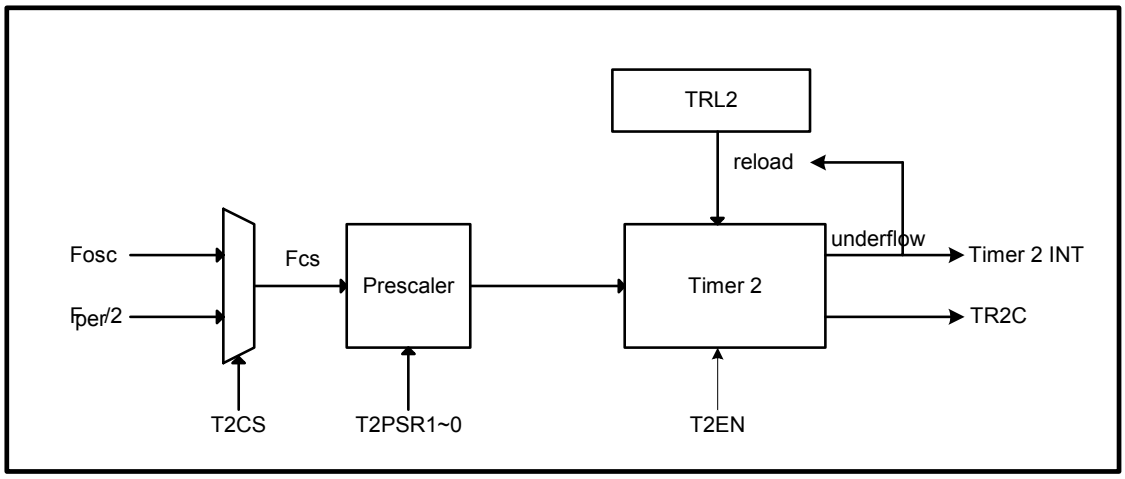


Fig. 8-3 Timer 2 Function Block Diagram

Timer 2 is a general-purpose 8 bits countdown counter for some applications which require time count. Interrupt function is also available for user's application. The clock source (Fcs) is from the oscillator clock or Fper/2.

Timer 2 is also equipped with a prescaler. The T2PSR1~T2PSR0 bit of TR2CON register determine the prescaler ratio while generating different clock rates as the clock source for the timer.

Counter value will decrease by one (counting from countdown) according to the actual type of timer clock source in use. When counter value underflows, the timer interrupt will occur if Timer 2 interrupt is enabled.

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

8.3.1 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

8.3.2 Timer 2 Applicable Registers

- **TRL2 (R10h):** is 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): Control bit of Timer 2 interrupt
"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
; === Timer2 = (8M/2) / [4 x 3F + 1]
Timer2SR:
            :
            System setting 8MHz
            Port G setting output port
            :
            MOV         A,#00000110B
            MOV         TR2CON,A        ; Fpll & Prescale 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

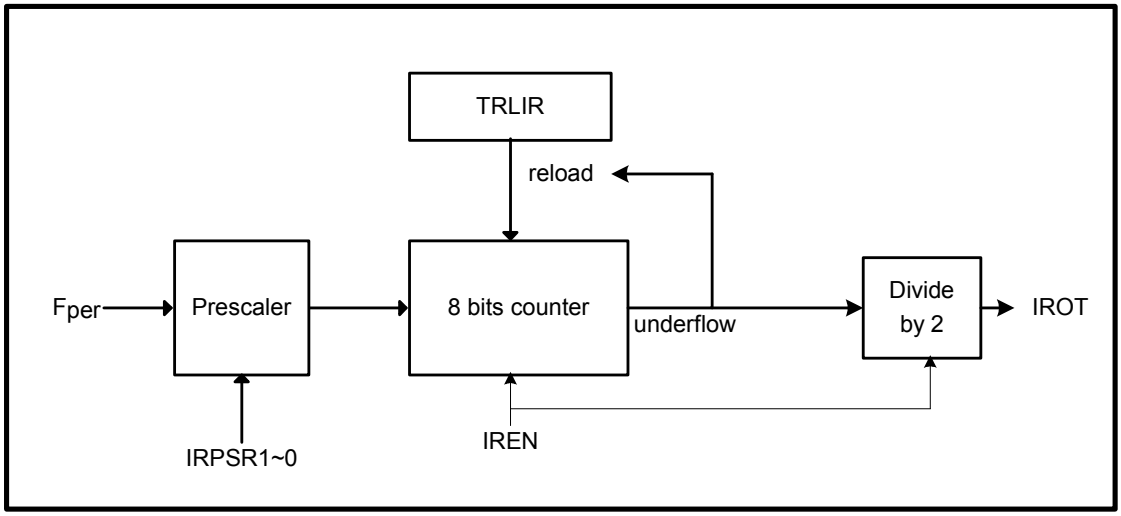


Fig. 8-3 IR Generator Function Block Diagram

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

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

8.4.1 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

8.4.2 IR Generator Applicable Registers

- **TRLIR (R29h):** Used to store the auto-reload value of IR generator. When IR generator is enabled or underflow occurs, the TRLIR register will automatically reload into 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 the IR function and recover the IROT pin as a general I/O pin
"1" : Enable the 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)

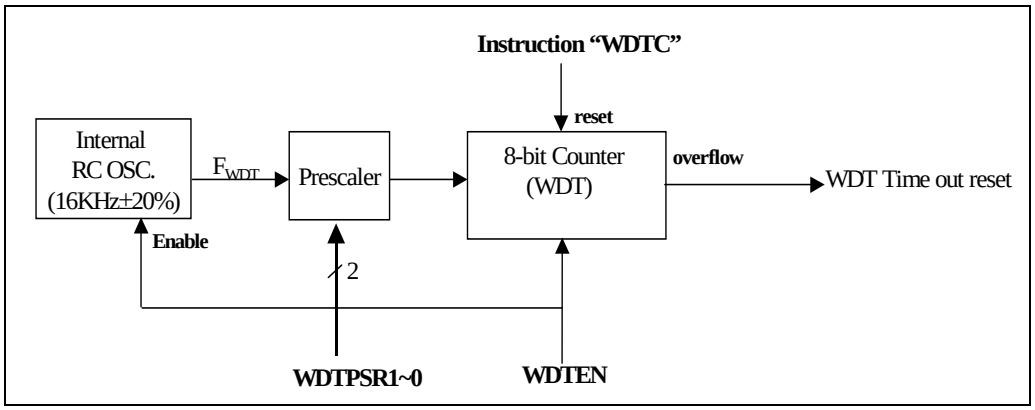


Fig. 8-5 Watchdog Timer Function Block Diagram

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 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. Disable is the WDT default condition. There is also a prescaler to generate different clock rates for the WDT clock source. The prescaler ratio is defined by WDTPSR1 ~ WDTPSR0.

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

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

8.5.1 Watchdog Timer Applicable Registers

■ **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 the watchdog timer (stop running)

"1" : Enable the 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 selectable
- Error detects function
- Interrupt available for Tx buffer empty, Rx buffer full, and receiver error
- TXD and RXD ports inverse output control

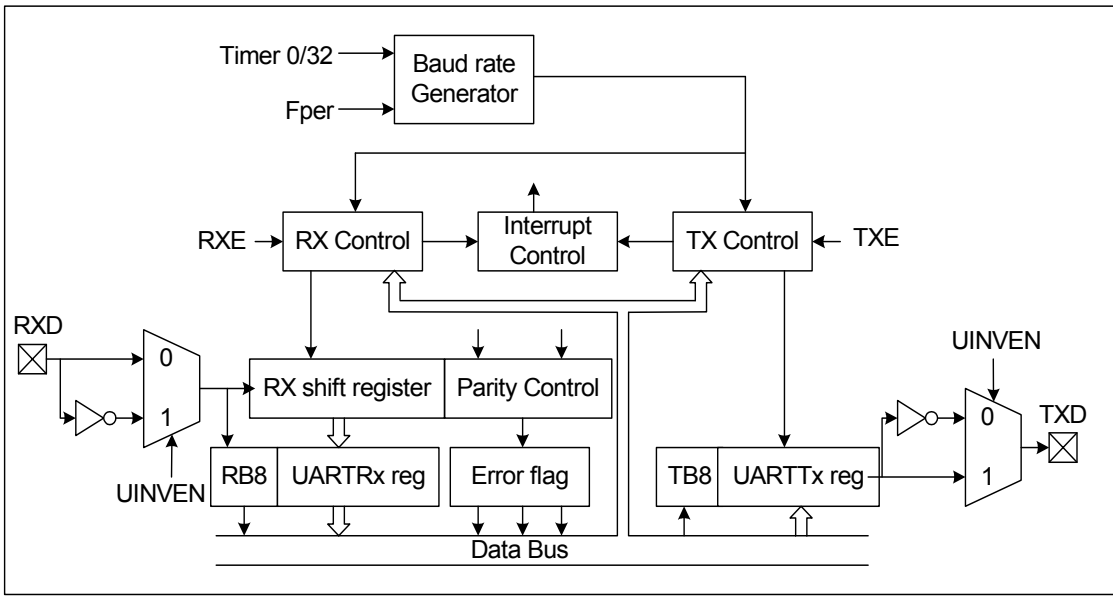


Fig. 8-6 UART Function Block Diagram

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

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 a character sent or received. The communication channel is normally held in the mark 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 then followed by the parity bit which will cause the stop bit to be high to confirm the end of the frame.

In receiving, the UART synchronizes on the falling edge of the start bit. When two or more “0” are detected during 3 samples, it is recognized as normal start bit and the receiving operation is started.

8.6.1 Data Format in UART

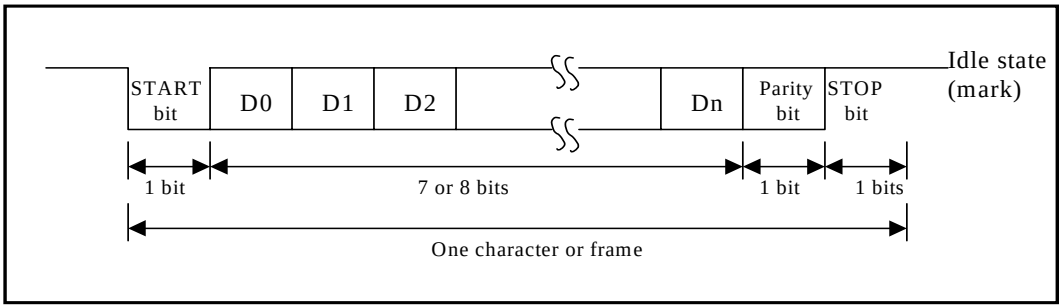


Fig. 8-7 UART Data Format Diagram

8.6.2 UART Modes

There are three modes in the UART. Mode 1 (7 bits data) and Mode 2 (8 bits data) allow the addition of a parity bit. Mode 3 (9 bits data) however, does not allow the parity bit addition. The figure below shows the data format in each mode.

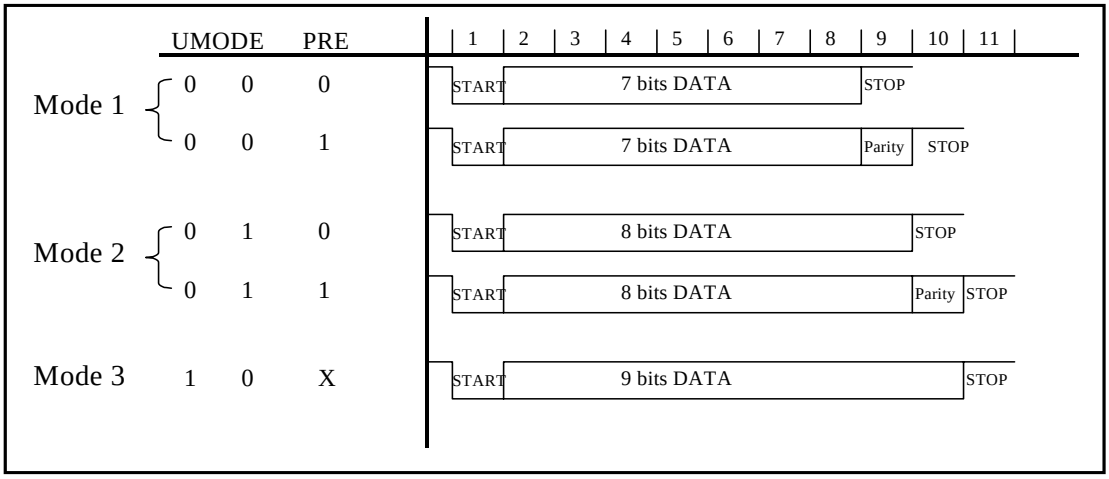


Fig. 8-8 UART Modes Data Format

8.6.3 UART Transmit Data

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 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

In receiving, the UART operates as follows:

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 then checks the following:
   a) Parity check: The number of "1" in the receive 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: URBF bit of UARTCON register must be cleared (i.e., UARTRX register should be read out) before the next received data are loaded into the UARTRX register.

   If any of the above 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 else the UERRI interrupt will occur when the next byte is received.
6. Reads the received data from UARTRX register and the URBF bit is 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 of the transmitted/received 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 9.830MHz (PFS = 150) or 14.745MHz (PFS = 225) when UART is enabled.
- The BRATE2 ~ BRATE0 bits of the UARTCON register can determine the desired baud rate.

8.6.6 UART Applicable 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 the transmit data function

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

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

Bit 4 ~ Bit 2 (BRATE2 ~ 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	Timer0/32	Timer0/32	Timer0/32	Timer0/32
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 (UMODE1 ~ 0): UART mode

UMODE1: UMODE0	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): Transmit data Bit 8

■ **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 is enabled, URBF (read-only) bit will be cleared by hardware. Hence, read-only 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): Receive 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): Transmit 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 ~ Bit0 (RB7 ~ RB0): Receive 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 interrupts
"1" : Enable all un-mask 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 3 (UERRIE): Control bit of UART receive error interrupt
"0" : Disable
"1" : Enable

Bit 4 (UTXIE): Control bit of UART Transfer buffer empty interrupt
"0" : Disable
"1" : Enable

Bit 5 (URXIE): Control bit of UART Receiver buffer full interrupt
"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 receive 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 (i.e., TXE = 0)

Bit 5 (URXI): Set to "1" when UART receive buffer full occurs
Clear to "0" by software or disable UARTRX (i.e., RXE = 0)

8.6.7 Transmit Counter Timing

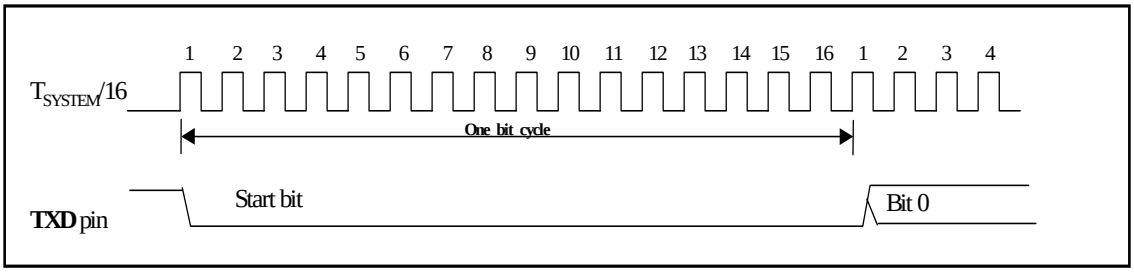


Fig. 8-9 UART Transmit Counter Timing

8.6.8 UART Transmit Operation (8-Bit Data with Parity Bit)

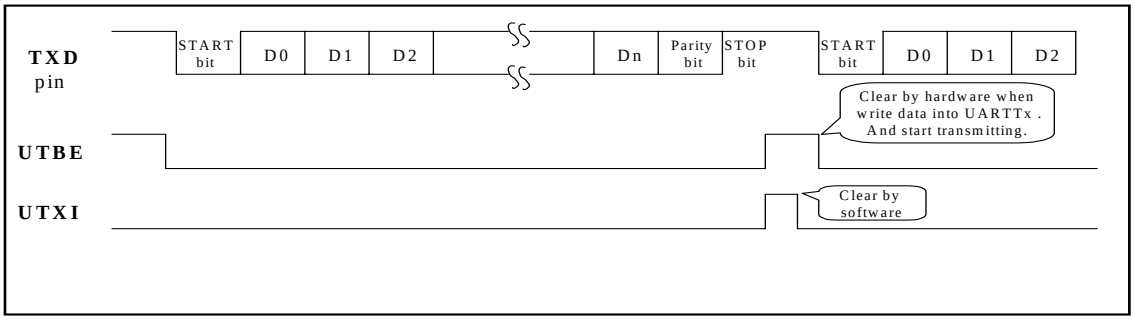


Fig. 8-10 UART Transmit Operation

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 8bit 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
```

8.6.9 Receive Counter Timing

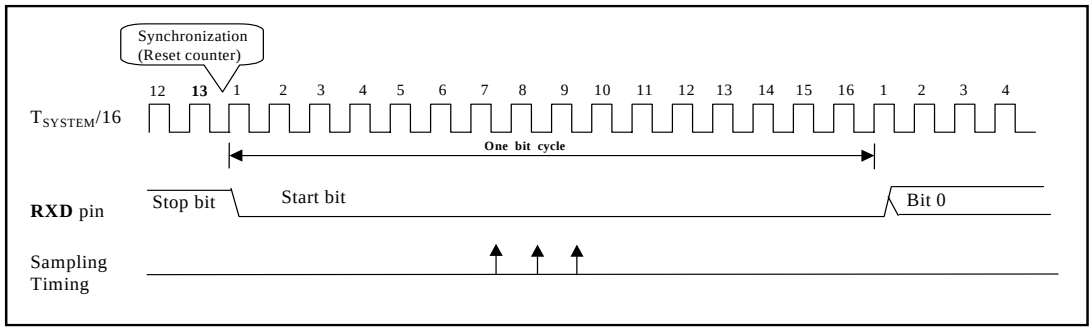


Fig. 8-11 UART Receive Counter Timing

8.6.10 UART Receive Operation (8-Bit Data with Parity and Stop Bit)

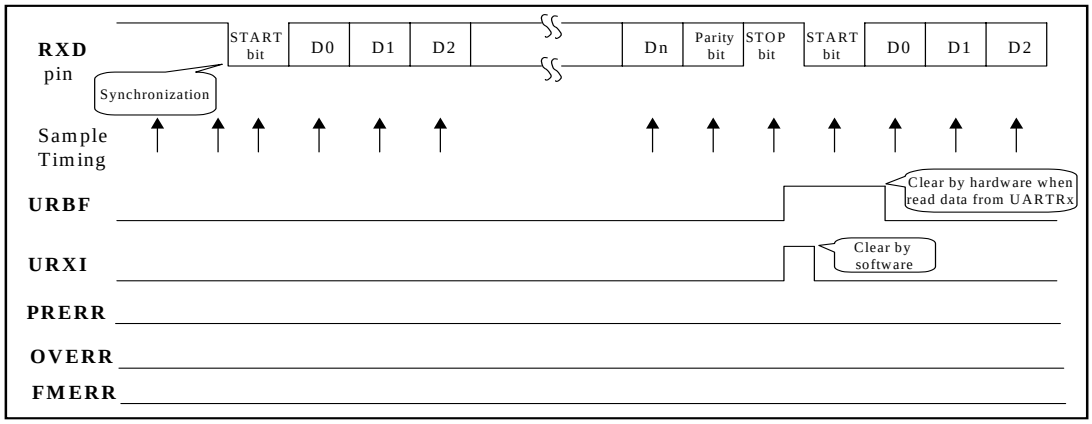


Fig. 8-12 UART Receive Operation

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

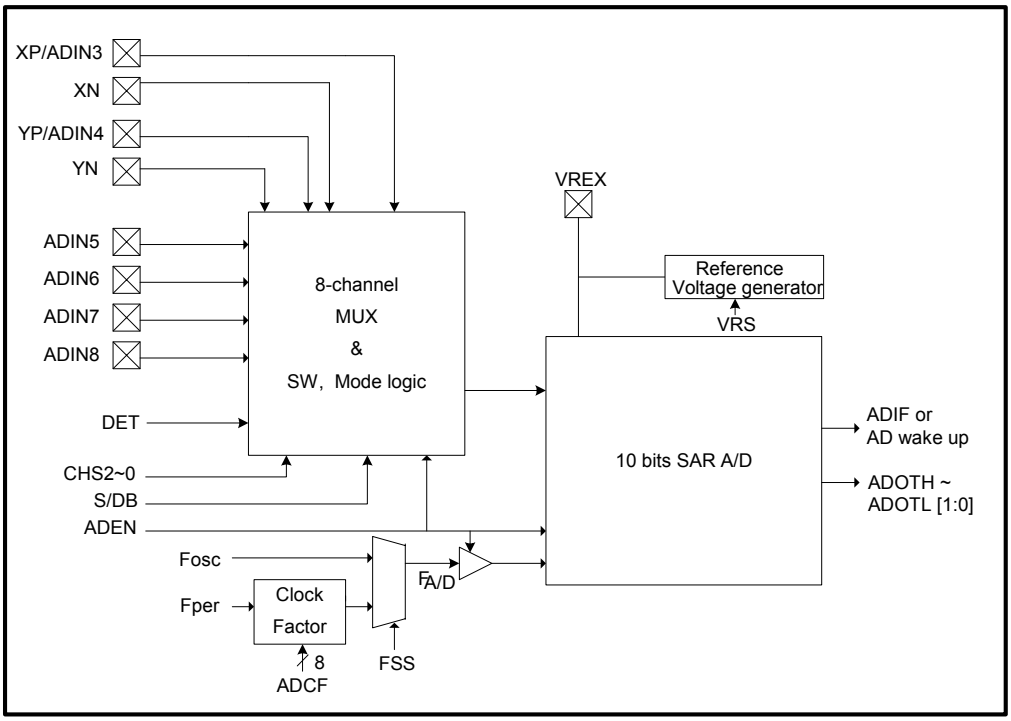


Fig. 8-13 A/D Converter Function Block Diagram

VREX: Reference voltage I/O pin
When VRS=1; it is input pin
When VRS=0, it is output pin

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 converter input Channel 5

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

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

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

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 converter operation for touch panel application is as follows:

Step 1: Pen down detection
If the panel is not tapped, the PIRQB is high. When the touch panel is tapped, the PIRQB is low and PIRQB interrupt occurs (if INT is enabled).

Step 2: Measure the X position
If the PIRQB remains low and steady for awhile, the DET bit is cleared, then the PIRQB returns to high and the X position is measured.

Step 3: Measure the Y position
Y position is immediately measured after Step 2

Step 4: Back to Step 1

8.7.1 A/D Converter Applicable 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 & 6-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 disabled
"1" : A/D enabled

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 off 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

Bit 7 ~ Bit 0 (ADOT9 ~ ADOT2: 10-bit resolution A/D output data for Registers ADOT9 ~ 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 1 ~ Bit 0 (ADOT1 ~ ADOT0): 10-bit resolution A/D output data for ADOT1 ~ ADOT0

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 the 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:

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

A/D Throughput rate=FA/D/12

ADCF Value	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

Any FA/D value greater than 1.4MHz 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): High frequency 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

8.7.2 A/D Converter General Applicable Timing

CHS [2:0] = 010 ~ 111

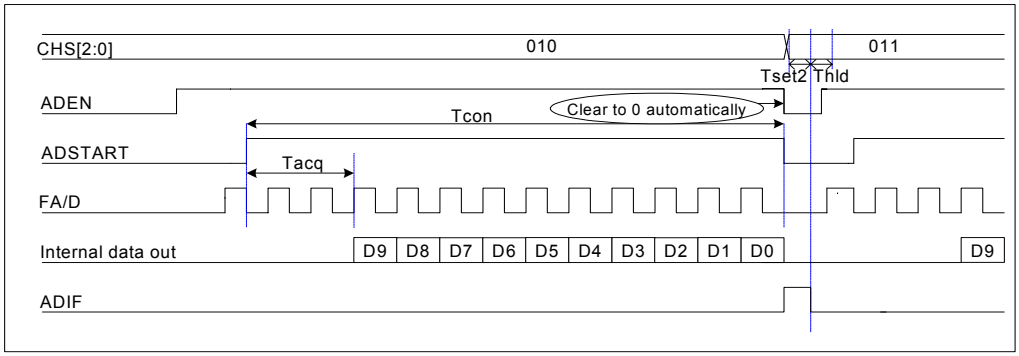


Fig. 8-14 A/D Converter Gen. Application Timing Diagram

8.7.3 Correlation between A/D Converter and MCU Mode

When MCU is in FAST mode. When MCU is in SLOW mode.

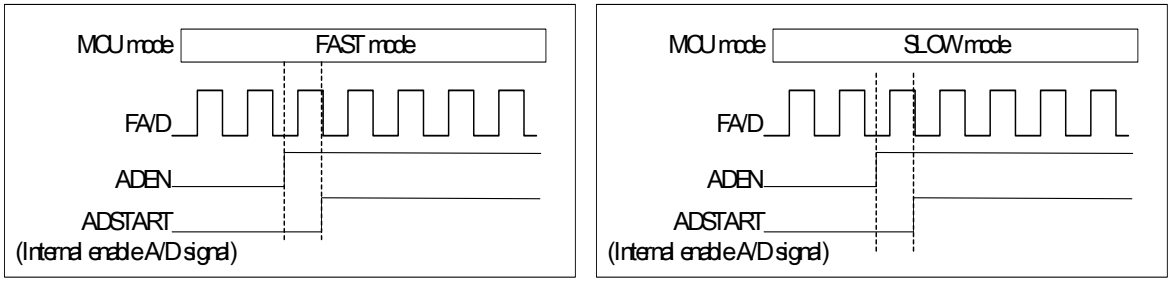


Fig. 8-15 A/D Converter vs. MCU Mode Correlation

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

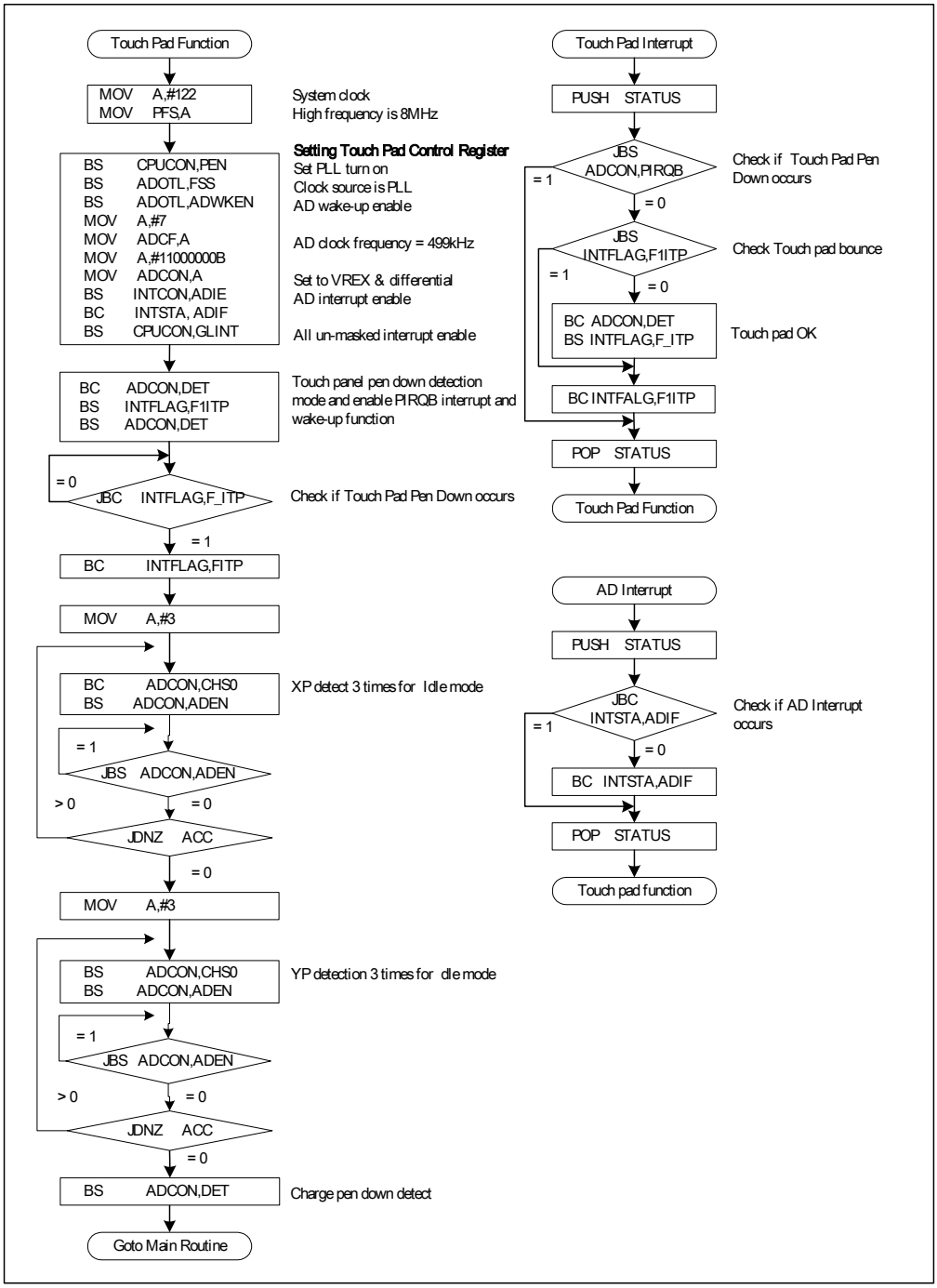


Fig. 8-16 A/D Converter Flow Chart

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 detection 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 wakeup
        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:
        JBC     INTFLAG,F_ITP,TPILp1
        BC      INTFLAG,F_ITP
; --- 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 of 128-key matrix. The 16 to 1 key strobe multiplex pins are controlled by STB3 ~ STB0 bits of the STBCON register. Interrupt is enabled when Port A falling edge is detected. Similarly, Wake-up is enabled when key input falling edge is detected.

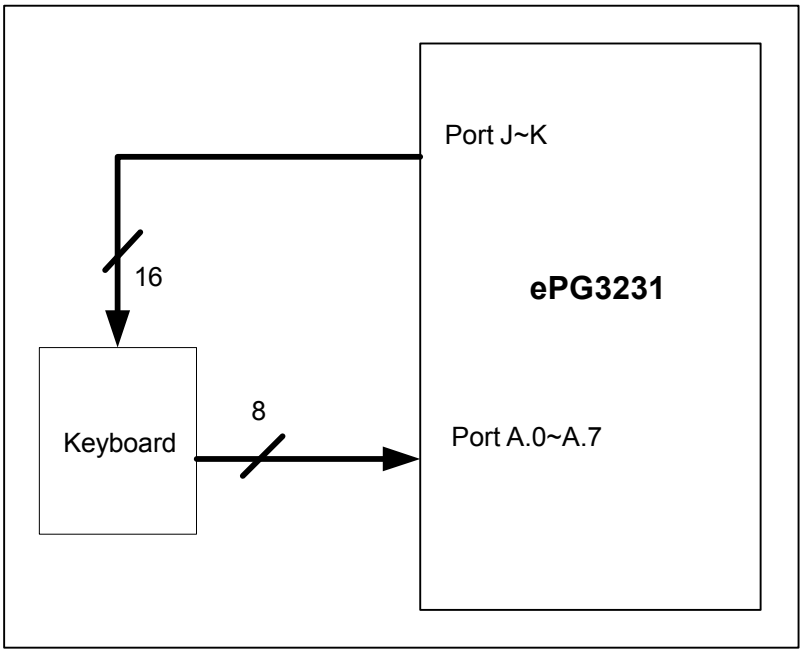


Fig. 8-17 Keyboard Function Block Diagram

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

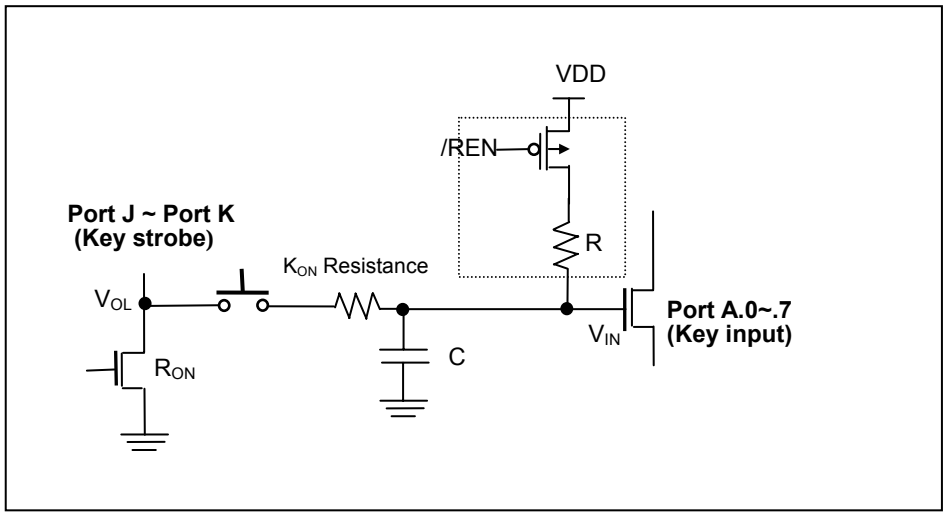


Fig. 8-18 Key Circuit Diagram

Port J ~ Port K are designed for 16-bit key strobe or general I/O. When the STBCON register is set to "BitST" bit, Ports J and K become key strobe pin. The 16 to 1 Port J and Port K multiplex are selected by STB3 ~ STB0 bits of the STBCON register. Only the selected pin will output low and the other 15 pins will remain high. Likewise, when the STBCON register is set to "ALL" bit, all strobe pins will output low.

The detailed function is summarized in the following table:

STBCON			Key Strobe (Port J & Port K)															
BitS	ALL	STB3~0	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	×	×	Use as 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	××××	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.8.1 Input/Output Key Applicable Registers

■ **Port A (R17h):** Port A Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

Bit 7 ~ Bit 0: Input key. Input falling edges 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 Wakeup 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 wakeup 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 selector bit

Bit 4 (ALL): Set All strobe

Bit 5 (BitST): Enable Bit strobe
"0" : Port J and Port K function as general I/O port
"1" : Port J and Port K function as key strobe pin. The strobe signal is defined by STB3 ~ 0 registers.

Bit 6 (/REN): Port A.0 ~ Port A.7 Pull up resistor control bit
"0" : Enable pull-up resistor
"1" : Disable pull-up resistor

■ **Port J, Port K (R32h, R33h):** General I/O registers

■ **DCRJK (R39h):** Port J & Port K Direction and Pull-up Resistor Control

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 to output pin
"1" : Set to 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-up resistor
"1" : Enable pull-up 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:
             :
;  ---  Port A 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                   ; Ports J & K output setting
            MOV         A,#0XFF
            MOV         PAWAKE,A                ; Port A wake-up function setting
            BC          STBCON,REN              ; Port A pull-up enable
            BS          STBCON,ALL              ; All strobe enable
            BC          CPUCON,MS1              ; Sleep mode
KeySleep:
            SLEP
            NOP
            MOV         A,PORTA                 ; 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 >= PORTA 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 External Memory Interface:

The EPG3231 provides a parallel memory interface for external memory devices, such as Flash memory, MASK ROM, SRAM, etc. There are 22 bits address bus and 16 bits data I/O bus. Address auto increase/decrease function is provided to achieve sequential memory access function. The data bus is 8 bits or 16 bits.

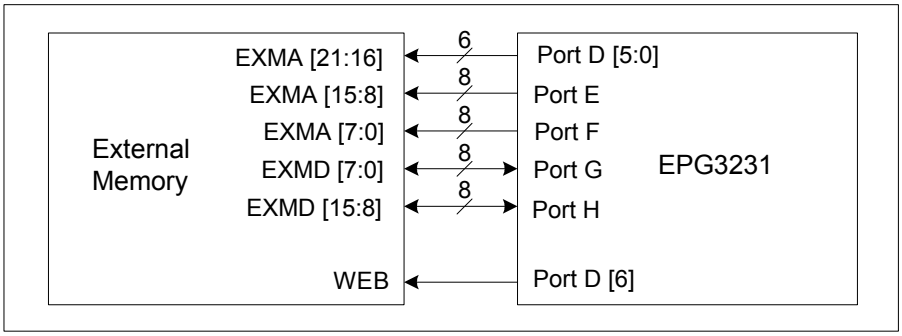


Fig. 8-19 Flash Memory and SRAM Interface

Port D.5~0, Port E and Port F are designed for external memory address bus, while Port G and Port H are for external memory data bus. Set the **EMPE** bit of POST_ID register to 1 to enable the external memory address post increase or post decrease function. Then set up the **EM_ID** bit of POST_ID register to determine whether post increase or post decrease. After reading data from Port G (low byte) register, the Port D [5:0]: Port E: Port F registers will auto increase or decrease. This function can achieve sequential memory access without changing the address data. A negative pulse will be generated from Port D.6 (WEB pin) when writing to Port G register (refer to WEB signal timing). This signal can be used as WEB for external memory.

■ **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 6 (EXTMEM): Set as external memory bus or as I/O port
"0" : Ports D, E, F, G & H are used as I/O or memory bus depending on the Program Counter location
"1" : Ports D, E, F, G & H are used as memory bus

■ **POST_ID (R2Bh):** Post Increase/ Decrease 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 7 (EM_ID): Setting this bit means auto increase Port F: Port E: Port D [6:0] registers, resetting this bit means auto decrease.

Bit 3 (EMPE): Enable external memory address post increase/decrease function when reading data from Port G.

(Only reading data from Port G register can activate this function)

■ **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 7 (WEN): Enable Port D.6 as external memory data output WEB signal.
(Refer to the timing diagram)
"0" : Disable
"1" : Enable

(WEB signal output is only available when writing to Port G register).

■ **Port D (R1Ah):** Bit 5 ~ Bit 0 (D.5~D.0): High byte of external memory address bus

Bit 6 (WEB): External memory interface write enable pin

Bit 7 (/OE): Extended PROM/DROM low active output enable

■ **Port E (R1Bh):** Middle byte of external memory address bus. Port E has the capability to carry into/borrow from Port D register.

■ **DCRDE (R36h):** Direction Control & Pull-up resistor Control of Port D & Port E

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): Ports D & E high / low nibbles direction control
"0" : Set to output pin
"1" : Set to input pin

Bit 3, Bit 7 (DHNPU, EHNPU) & Bit 2, Bit 6 (DLNPU, ELNPU): Enable Ports D & E high / low nibble pull-up resistor
"0" : Disable pull-up resistor
"1" : Enable pull-up resistor

■ **PORTF (R1Ch):** Low byte of the external memory address bus. Port F has ability to carry into/borrow from Port E register

■ **PORTG (R1Dh):** Low byte register of the external memory data bus

■ **DCRFG (R37h):** Direction control & Pull up resistor control of Ports 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): Ports F & G high / low nibbles direction control.
"0" : Set to output pin
"1" : Set to input pin

Bit 3, Bit 7 (FHNPU, GHNPU) & Bit 2, Bit 6 (FLNPU, GLNPU): Enable Ports F & G high / low nibble pull-up resistor
"0" : Disable pull-up resistor
"1" : Enable pull-up resistor

- **PORTH (R1Eh):** High byte registers of external memory data bus

- **DCRHI (R38h):** Direction control & Pull-up resistor control of Port H

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 0 (HHNDC & HLNDC): Port H high & low nibbles direction control
"0" : Set to output pin
"1" : Set to input pin

Bit 3 & Bit 2 (HHNPU & HLNPU): Enable Port H high & low nibbles pull up resistor
"0" : Disable pull-up resistor
"1" : Enable pull-up resistor

8.9.1 8-Bit Data Bus: Read Mode Timing

EMPE=1 (enable post increase/decrease function), EM_ID=1 (set to auto increase)

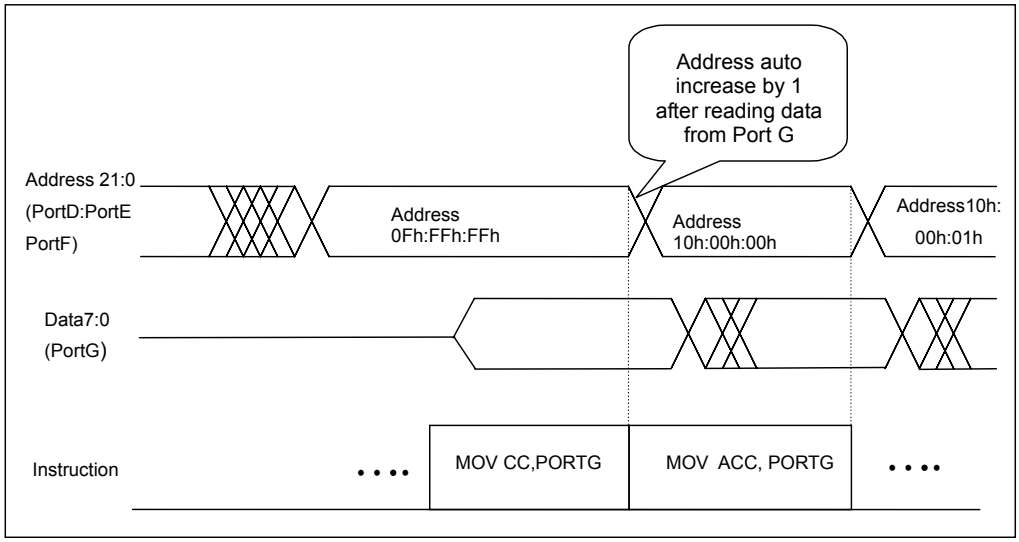


Fig. 8-20 Flash Memory and SRAM Interface

EMPE=1 (enable post increase/decrease function), EM_ID=0 (set to address auto decrease)

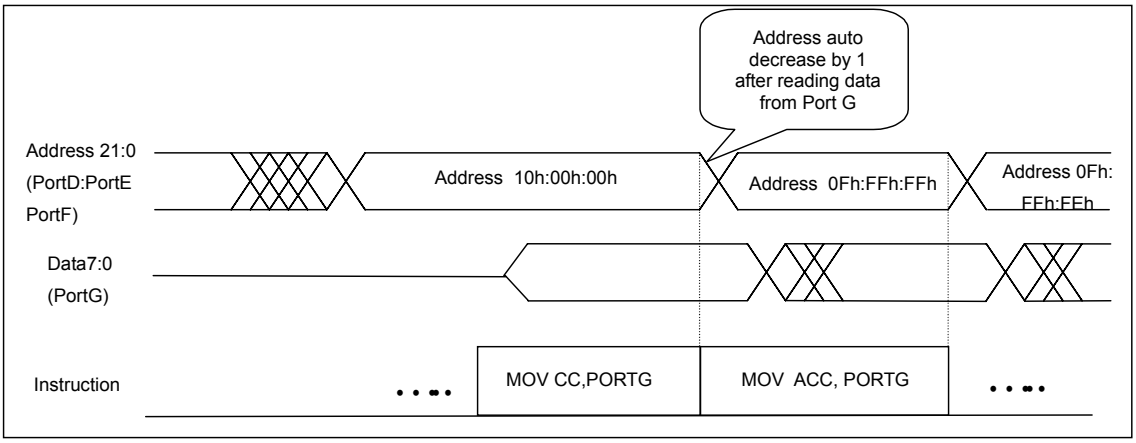


Fig. 8-21 Flash Memory and SRAM Interface

8.9.2 8-Bit Data Bus: Write Mode (WEB) Timing

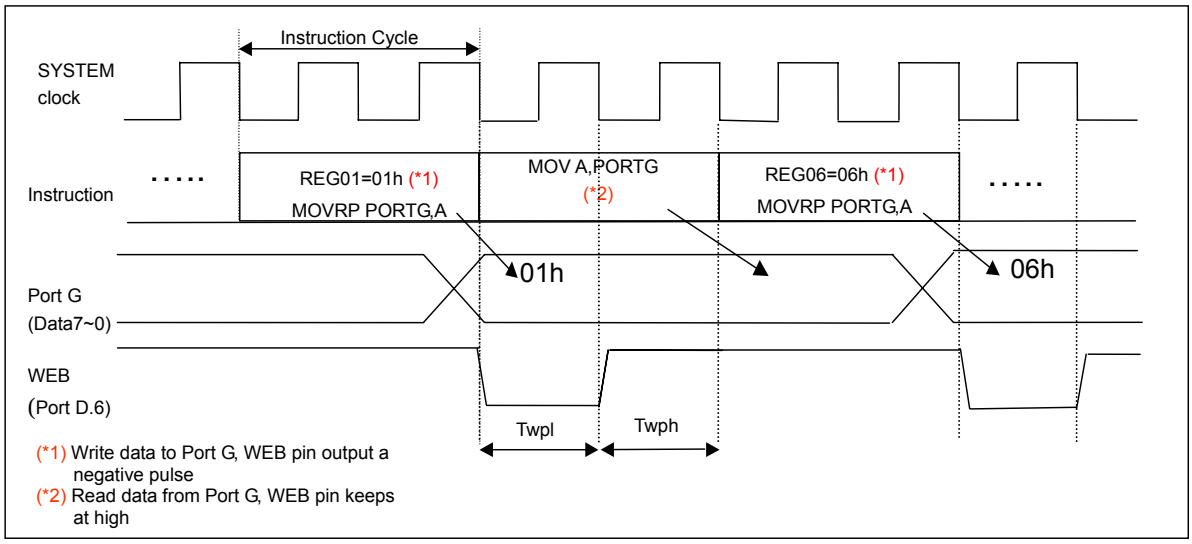


Fig. 8-22 WEB Signal Output Timing Diagram

8.9.3 16-Bit Data Bus: Read Mode Timing

EMPE=1 (enable post increase/decrease function), EM_ID=1 (set to auto increase)

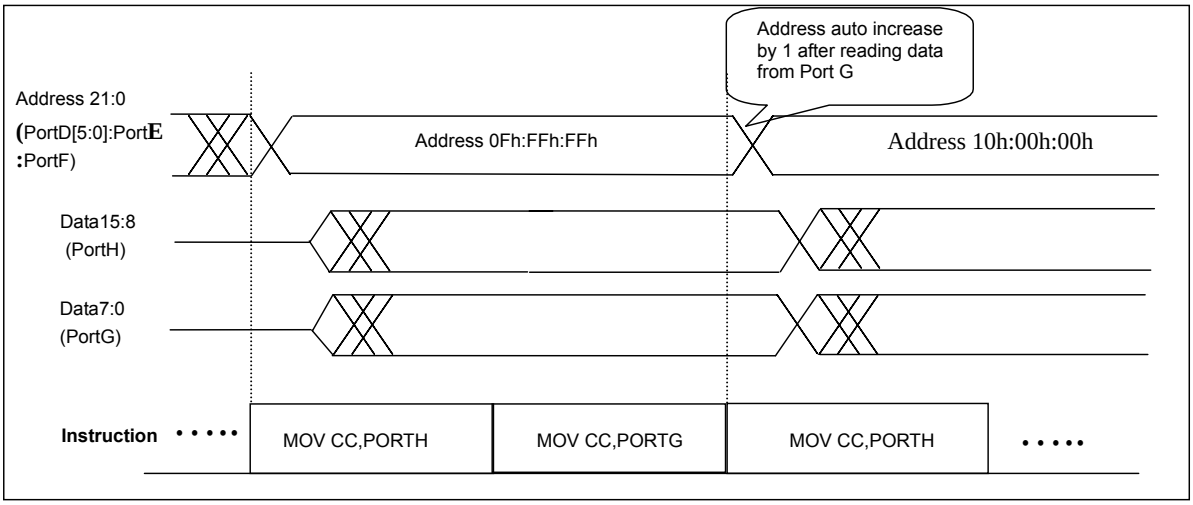


Fig. 8-23 Flash Memory and SRAM Interface

8.9.4 16-Bit Data Bus: Read Mode Timing

EMPE=1 (enable post increase/decrease function), EM_ID=0 (set to address auto decrease)

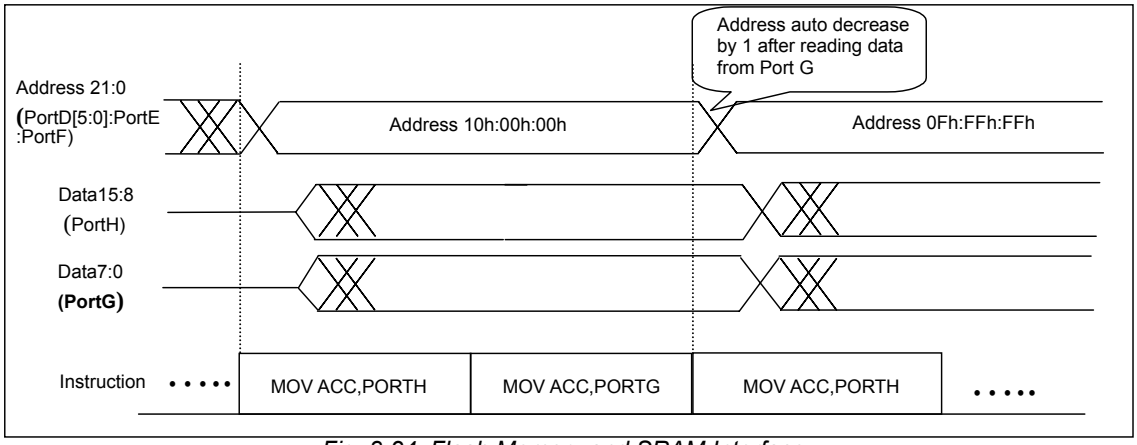


Fig. 8-24 Flash Memory and SRAM Interface

8.9.5 16-Bit Data Bus: Write Mode (WEB) Timing

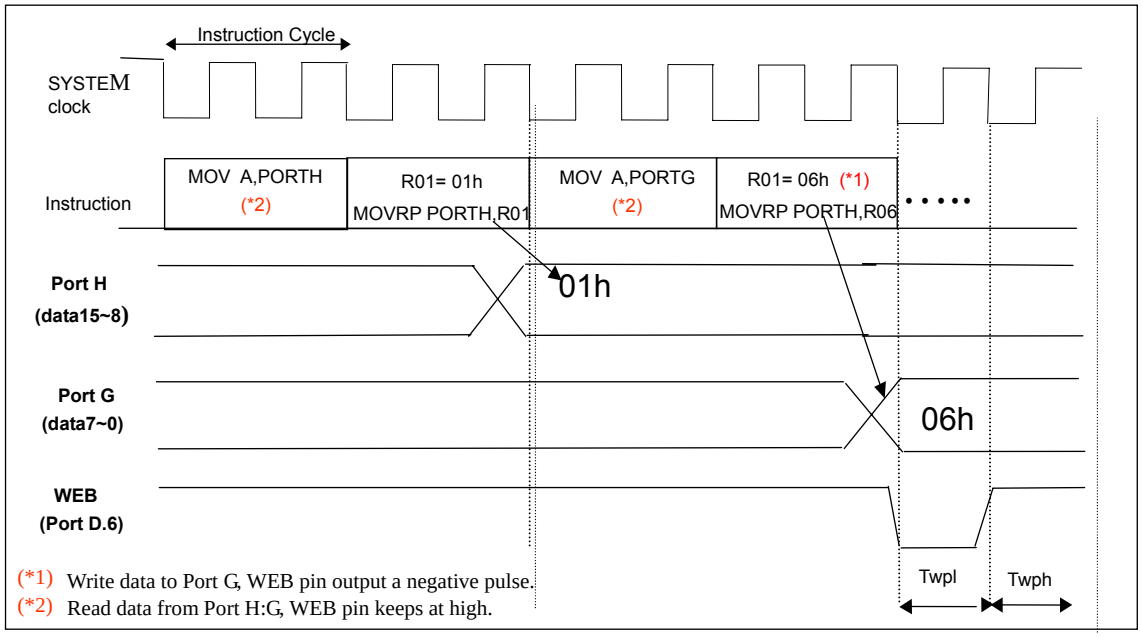


Fig. 8-25 WEB Signal Output Timing Diagram

Code Example:

```
INCLUDE   "ExDATA.hdr"
;***External memory control
;--- Address D:E:F & Data output G:H
      CLR           DCRDE
      CLR           DCRFG
      CLR           DCRHI
;---External memory enable auto INC. & WEN signal
      BS            POST_ID,EM_ID
      BS            POST_ID,EMPE
      BS            SPISTA,WEN
;---RAM enable auto INC.
      BS            POST_ID,FSR1_ID
      BS            POST_ID,FSR1PE
;---Bank 1 address 80
      MOV           A,#1
      MOV           BSR1,A
      CLR           FSR1
;---Address 10000
      CLR           PORTF
      CLR           PORTE
      MOV           A,#11000001B
      MOV           PORTD,A
;---Memory IC CE active
      BC            PORTD,CE
;---Bank 1 address 80 ~ 9Fh data out to 10000 ~ 1000Fh
      MOV           A,#16
DataOut:
      MOVPR         PORTH,INDF1
      MOVPR         PORTG,INDF1
      JDNZ          ACC,DataOut
;---Memory IC CE pin disable
      BS            PORTD,CE
       :
       :
```

```
;---Read external memory data
;---Table pointer setting
      TBPTH     #T_DATA1_h
      TBPTM     #T_DATA1_m
      TBPTL     #T_DATA1_l
;---External data ROM
      BS        TABPTRH,7
;---Memory IC CE active
      BC            PORTD,CE
;---Bank 1 address 80
      MOV       A,#1
      MOV       BSR1,A
      CLR       FSR1
;---Read data 16 word to bank 2 address 80 ~ 9Fh
      MOV       A,#32
      RPT       ACC
      TBRD      1,INDF1
       :
       :

;*****ExDATA.hdr file  *****
_T_DATA1_       EQU         0x000000
;Segment size = 0x10000, 65536 Bytes
_T_DATA1_l      EQU         0x00
_T_DATA1_m      EQU         0x00
_T_DATA1_h      EQU         0x00

_T_DATA2_       EQU         0x010000
;Segment size = 0x41560, 267616 Bytes
_T_DATA2_l      EQU         0x00
_T_DATA2_m      EQU         0x00
_T_DATA2_h      EQU         0x01
       :
       :
```

8.10 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 communication bit rates
- 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

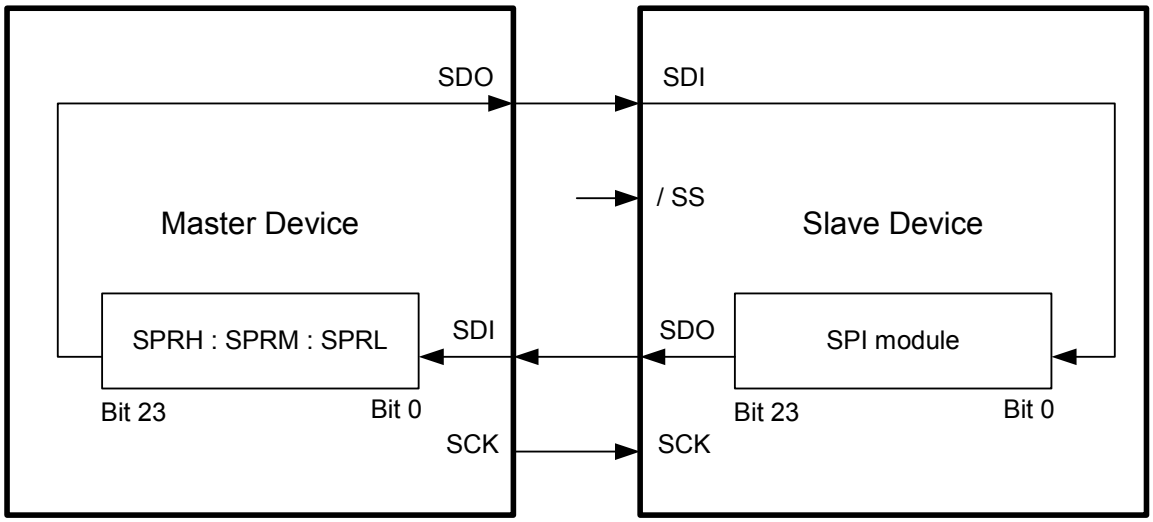


Fig. 8-26a Single SPI Master/Slave Communication

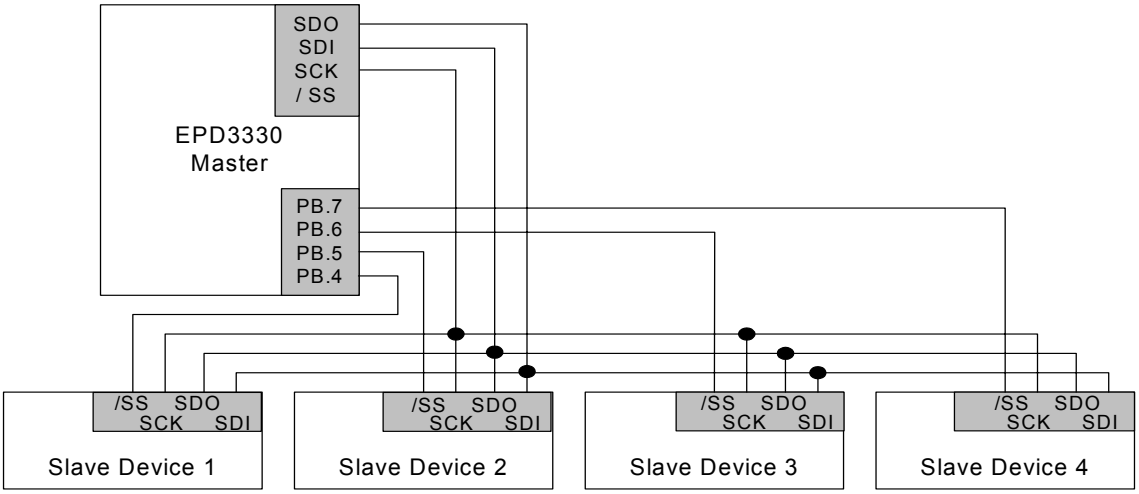


Fig. 8-26b SPI Configuration Example of Single-Master and Multi-Slaves

The MCU communicates with other devices through an 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.

8.10.1 SPI Pin Description

SDI (I): Serial Data Input pin. Receives data serially

SDO (O): Serial Data Output pin. Transmit data serially. Under Slave mode, defined as high-impedance, if not selected.

SCK (I/O): Serial Clock input/output pin. When in Master mode, sends clock through the SCK pin. However, under Slave mode, SCK pin is programmed as an input pin).

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

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

8.10.2 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 an 8-bit shift register length is selected; the SPRL register becomes the contents 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 SCK pin.

Receiver is active during SPI transfer. When the receive buffer is full, RBF flag is set and interrupt occurs (if enable). During read out of the shift register content, it is only after 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 in.

8.10.3 Slave Mode

Under Slave mode, input clock is sourced from MASTER device. SCK pin functions as clock input pin. The SE bit is not used to control the starting shift in this mode; but is used as Transfer buffer empty status bit.

As in Master mode, you can select shift register length and write transfer 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. But the start shifting is controlled by the MASTER device clock input.

When the shift buffer is empty, the SE bit will be cleared. At the same time, the receive buffer becomes full, RBF flag is set, and interrupt occurs (if enabled). The data are received by SPRH, SPRM, and SPRL registers. You should read them out before the next clock-in takes place. Otherwise data collision will result and DCOL bit of SPISTA register will be set.

8.10.4 SPI Applicable Registers

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

The buffer will deny any write attempt until shifting is completed. If a 24-bit shift buffer is selected, it will include the SPRH, SPRM, and SPRL registers. Else, if 8-bit shift buffer is selected, only the SPRL register is included.

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

■ **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. Therefore, writing to the SPRL register is necessary when you want 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 edges 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

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. Shift buffer length is programmable as follows:
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 is cleared by hardware when SPI is enabled and this bit becomes read-only. 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

8.10.5 SPI Timing Diagrams

■ **Master Mode (Shift Buffer Length = 24Bits)**

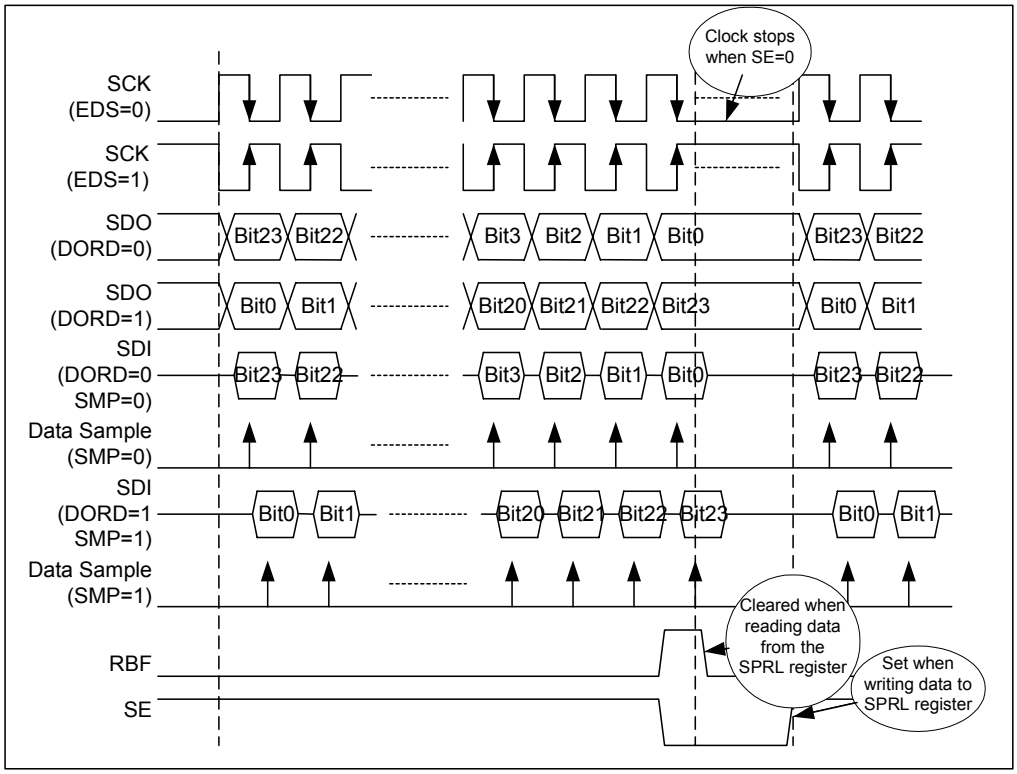


Fig. 8-27 SPI Master Mode Timing Diagram

■ **Slave Mode (Shift Buffer Length = 8Bits, /SS Enabled)**

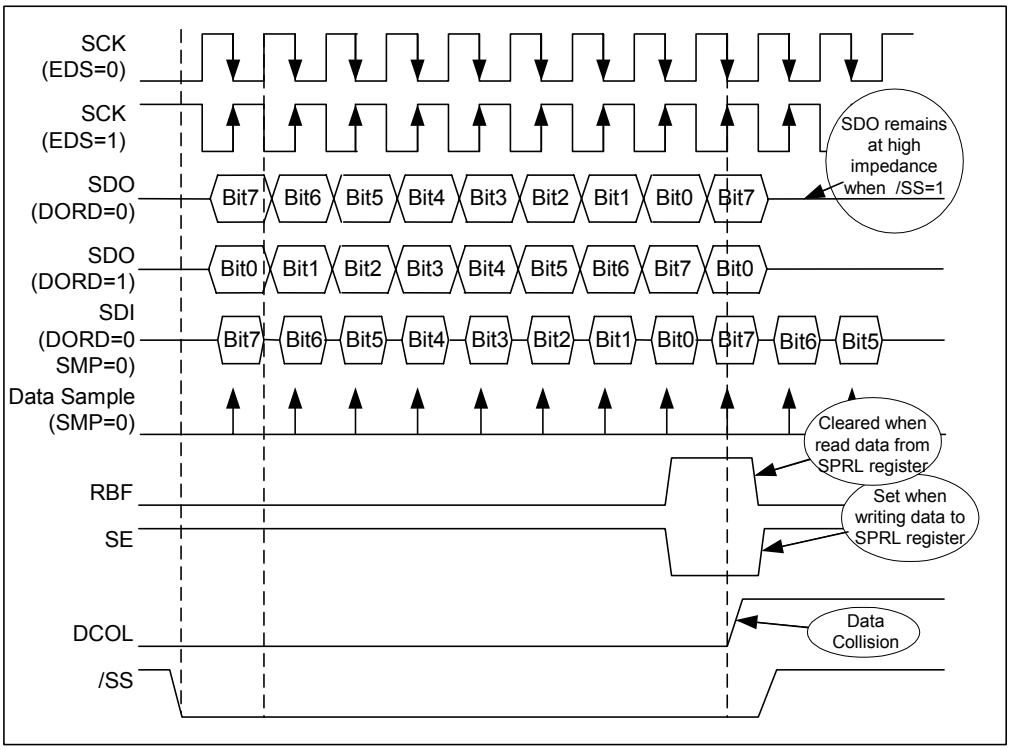


Fig. 8-28 SPI Slave Mode Timing Diagram

8.10.6 SPI Code Examples

■ **Master Mode (8-bit) Code Example:**

```
; *** 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 = 2000000bit rate
SPIM_SR:
          :
        System setting 8MHz
        Port G setting output port
          :
; --- 8bit, (FPLL/2) / 4, 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 (8-bit) Code Example:**

```
; *** 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 Wakeup & 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
```

8.11 Melody/Speech Synthesizer

The ePG3231 MCU provides four channels for melody/speech function. Channels 1~3 are destined for melody channel, and Channel 4 can be either melody or speech channel as determined by SPHSB bit (Bit 2 of R44). The Channels 1 ~ 4 are controlled by R45 ~ R4A of the corresponding control register Banks 0 ~ 3. And the Bits 0 ~ 2 of R44 are used to select the current control register bank.

	Melody Channel 1	Melody Channel 2	Melody Channel 3	Melody/Speech Channel 4
R44h	×××× ×000	×××× ×001	×××× ×010	×××× ×011 / ×××× ×1××
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

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

Bit 2 (SPHSB): Speech/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
1××	Speech Channel	Bank 3

8.11.1 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 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-bit timer for time counting. When the melody timer count value underflows, interrupt occurs and the MTRL value is auto-reloaded to count 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 registers are listed as follows**:**

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

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

"Mode1"**:** Disable melody interrupt function
Interrupt priority (External > Capture > Speech > Timers 0 ~ 2 > Peripheral)

"Mode2": Enable melody interrupt function
Interrupt priority (Timer 0 > Capture > Speech/melody > External > Timers 1, 2 > Peripheral).

- **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 the data ROM. The address values are written by the 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

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

Pitch	Index No.	Pitch	Index No.	Pitch	Index No.
A2	0X00	D4	0X11	G5	0X22
A#2	0X01	D#4	0X12	G#5	0X23
B2	0X02	E4	0X13	A5	0X24
C3	0X03	F4	0X14	A#5	0X25
C#3	0X04	F#4	0X15	B5	0X26
D3	0X05	G4	0X16	C6	0X27
D#3	0X06	G#4	0X17	C#6	0X28
E3	0X07	A4	0X18	D6	0X29
F3	0X08	A#4	0X19	D#6	0X2A
F#3	0X09	B4	0X1A	E6	0X2B
G3	0X0A	C5	0X1B	F6	0X2C
G#3	0X0B	C#5	0X1C	F#6	0X2D
A3	0X0C	D5	0X1D	G6	0X2E
A#3	0X0D	D#5	0X1E	G#6	0X2F
B3	0X0E	E5	0X1F	A6	0X30
C4	0X0F	F5	0X20	A#6	0X31
C#4	0X10	F#5	0X21		

■ **MTCON (R49h):** Melody Timer Control Register

The MTCON is used to determine the three MSB's of the 11-bit auto-reload register and to 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 (Mode2 only)	MTPSR0 (Mode2 only)	MTI (Mode2 only)	MTIE (Mode2 only)	MTEN	MTRLH2	MTRLH1	MTRLH0

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

Bits 4 ~ 3 (MTIE, MTEN): Melody timer enable control bits

MTIE	MTEN	Melody Timer Function
X	0	Melody Timer Disabled
0	1	Melody Timer Enabled and read melody data without interrupt
1	1	Melody Timer interrupt enabled without fetch melody data (Mode2 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 (Mode2 only).

Bit 7 ~ 6 (MTPSR1 ~ MTPSR0): Melody timer prescaler control bits (Mode2 only).

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

8.11.2 Speech Function

The 11-bit speech timer is shared with a melody timer (MT4) for Channel 4. 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 control registers are listed as follows**:**

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

- **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 is 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

The Speech timer is an 11-bit down counter for speech applications. The frequency generated by the 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, timer interrupt will occur and auto-reloads 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 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

Bits 4 ~ 3 (SPHTIE, PSHTEN): Speech timer enable control bits

SPHTIE	SPHTEN	Speech Timer Function
X	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 the speech timers interrupt occurred. Clear ("0") by software or speech timers disabled.

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

8.12 PWM / DAC Function

The ePG3231 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 the Bit 7 of PWD is "1" and the PWM timer counter equals to PWM value (Bits 0 ~ 6 of PWD), the VO1 transfers to low until the PWM timer is reset or overflowed. The VO2 is always kept at "0" in this case.

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

$$T_{period} = \frac{128}{F_{Per}} \times \Pr escaler; \quad T_{duty} = \frac{1}{F_{Per}} \times \Pr escaler \times (PWD + 1)$$

8.12.1 PWM Function Block Diagram

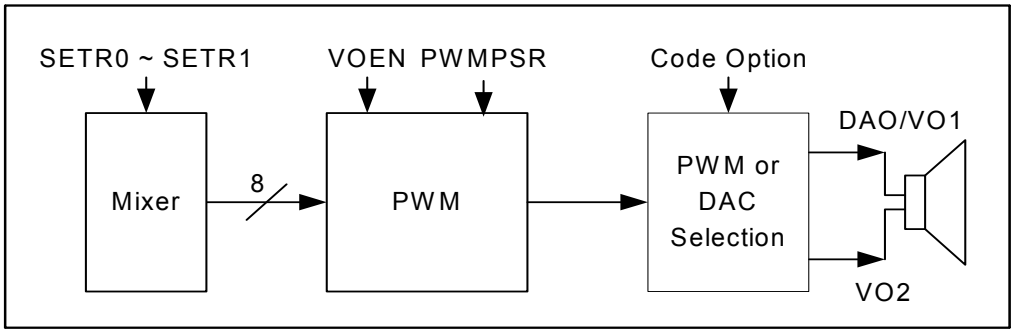


Fig. 8-29 PWM Function Block Diagram

8.12.2 DAC Function Block Diagram

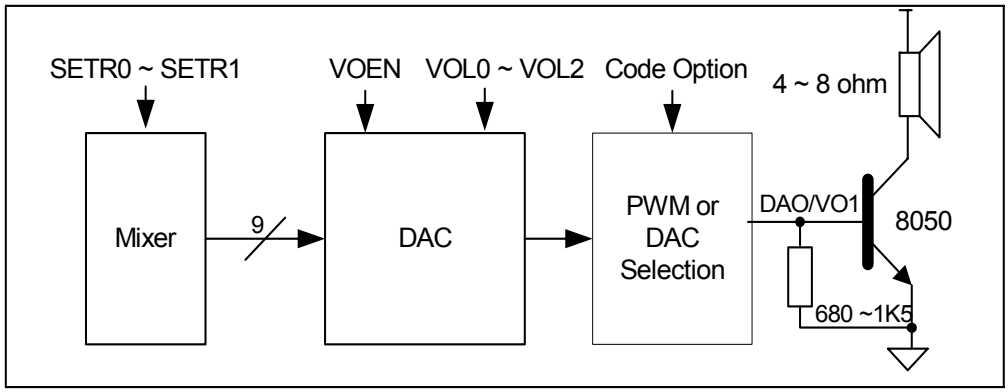


Fig. 8-30 DAC Function Block Diagram

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

8.12.3 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.

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

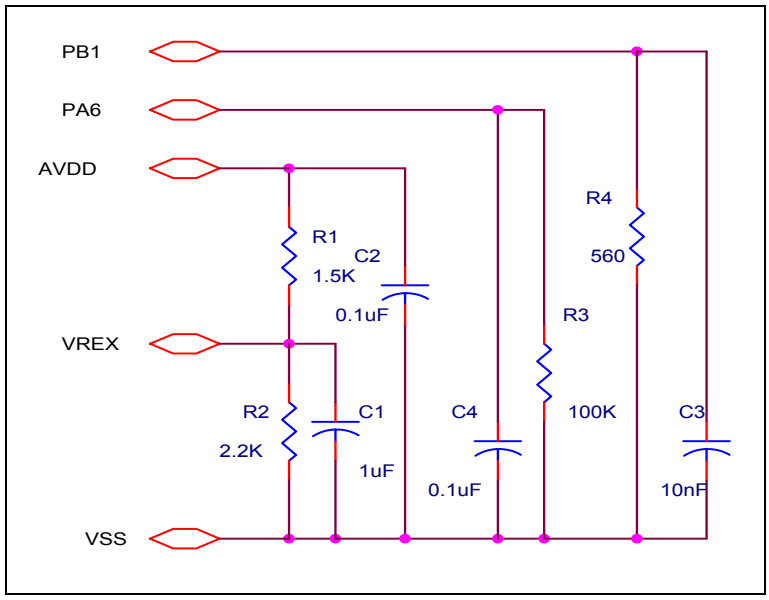


Fig. 8-31 External A/D & D/A Converters Reference Circuits

8.12.4 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

Bit 0 ~ 2 (VOL0 ~ VOL2): Volume control of DAC.

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 prescaler select bit

"0" : Prescaler 1:1

"1" : Prescaler 1:2

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

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

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

Bit 6 (DAC): D/A converter output control bit (for Code option Current D/A converter reference source)
"0" : D/A converter output control is by melody timer or speech timer
"1" : D/A output control is through SPHDR register

Bit 7 (VOEN): Voice output control bit
"0" : DAC/PWM disabled
"1" : DAC/PWM enabled

■ **ADCON (R2Ch):** A/D Converter 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
"1" : Disable internal reference generator and the reference voltage is sourced from external VREX pin
"0" : Enable internal reference generator and the reference voltage is sourced from internal reference voltage generator

Code Example:

Refer to the Melody & Speech Application Note (available upon request).

9 Electrical Characteristics

■ Absolute Maximum Ratings

Items	Sym.	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 Dissipation (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	Sym.	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	Sym.	Condition		Min	Typ	Max	Unit
CLOCK	Fmain	Main-clock frequency		1	-	16	MHz
	Fsub	Sub-clock frequency	RC OSC	24.6	32.8	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	

Parameter	Sym.	Condition		Min	Typ	Max	Unit
Output Current	IOH1	PB[0:7], PC[0:7], PI[7:4], PG[0:7], PH[0:7] (as 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] (as D/A output)	VDD=3.0V, VOH=0.7V	-2.5	-3.5	-4.5	
	IOH3	PB[1:0] (as 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] (as EL chop and CK output)	VDD=3.0V, VOH=1.0V	-3	-6	-9	
	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] (as 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 ports (without pull up/down resistor) Vin=VDD or GND		-	-	±1	μA
Large Pull-Up Resistance	RPU1	PA[7:0]~PK[7:0]	Vin=GND	500	1000	1500	KΩ
	RPU2	RSTB	Vin=GND	250	500	750	
Small Pull-Up Resistance	RPU3	PA[7:0]~PK[7:0]	Vin=2V	50	100	200	KΩ
	RPU4	RSTB	Vin=2V	50	100	200	
Large Pull-Down Resistance	RPD1	TEST	Vin=VDD	250	500	750	KΩ
Small Pull-Down Resistance	RPD2	TEST	Vin=1V	1.0	2.5	4.0	KΩ
Touch Panel Pull- Down 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

Parameter	Sym.	Condition	Min	Typ	Max	Unit
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	Sym.	Condition	Min	Typ	Max	Unit
Instruction Cycle Time	Tcycle	Fmain=1MHz	-	2*	-	μs
		Fmain=4MHz	-	0.5*	-	
		Fmain=16MHz	-	0.34*	-	
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

* Instruction cycle time = 2 × (System clock time)

10 Pin Type Circuit Diagrams

■ **TEST Pin Type**

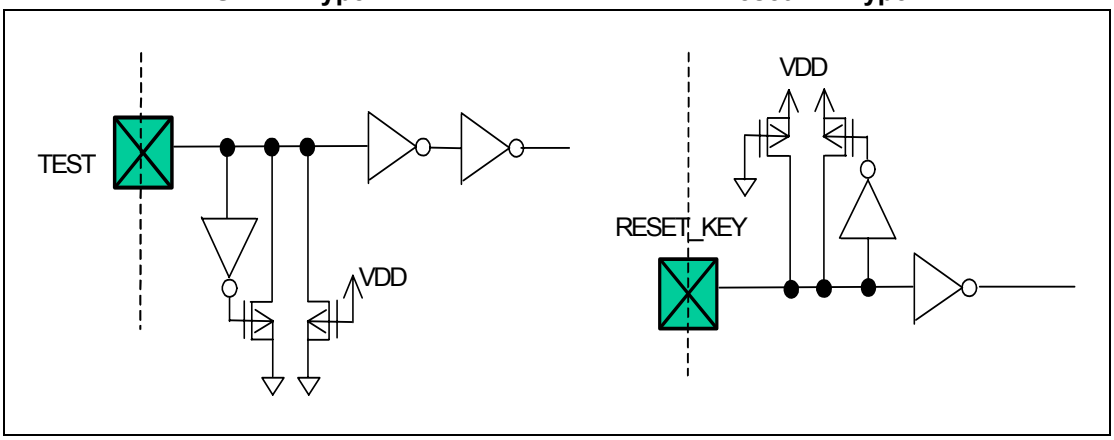


Fig. 8-32a TEST Pin Type Circuit Diagram

■ **Reset Pin Type**

Fig. 8-32b Reset Pin Type Circuit Diagram

■ **Output Pin Type**

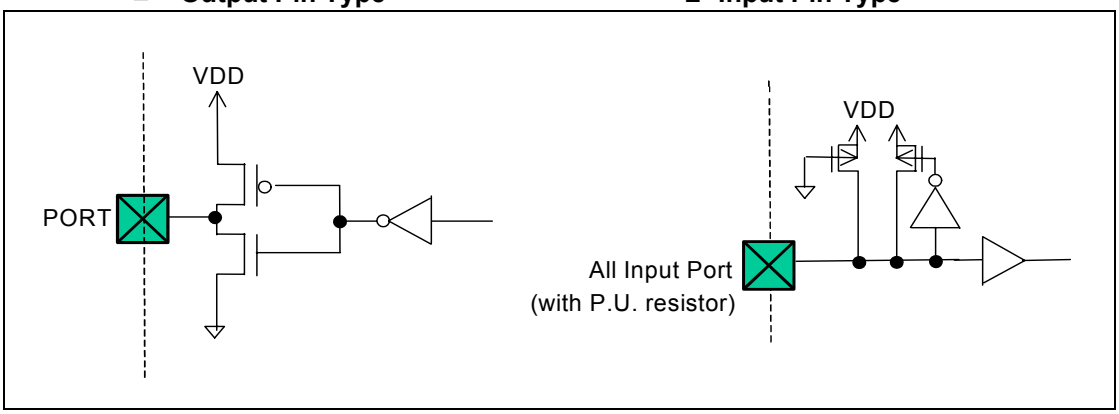


Fig. 8-32c Output Pin Type Circuit Diagram

■ **Input Pin Type**

Fig. 8-32d Input Pin Type Circuit Diagram

■ **General I/O Pin Type**

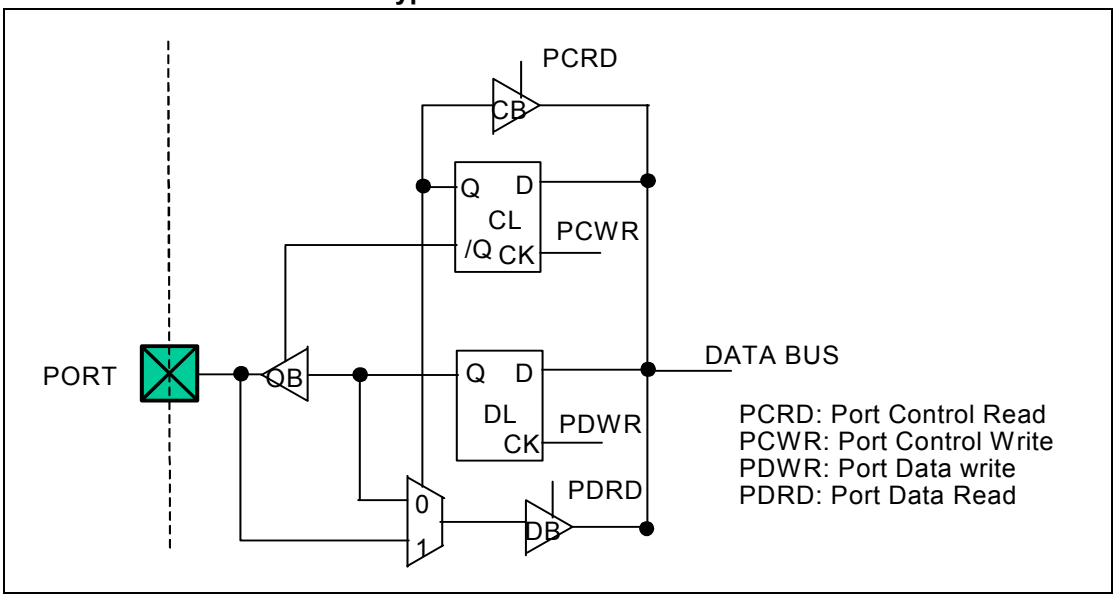


Fig. 8-32e General I/O Pin Type Circuit Diagram

11 Application Circuit

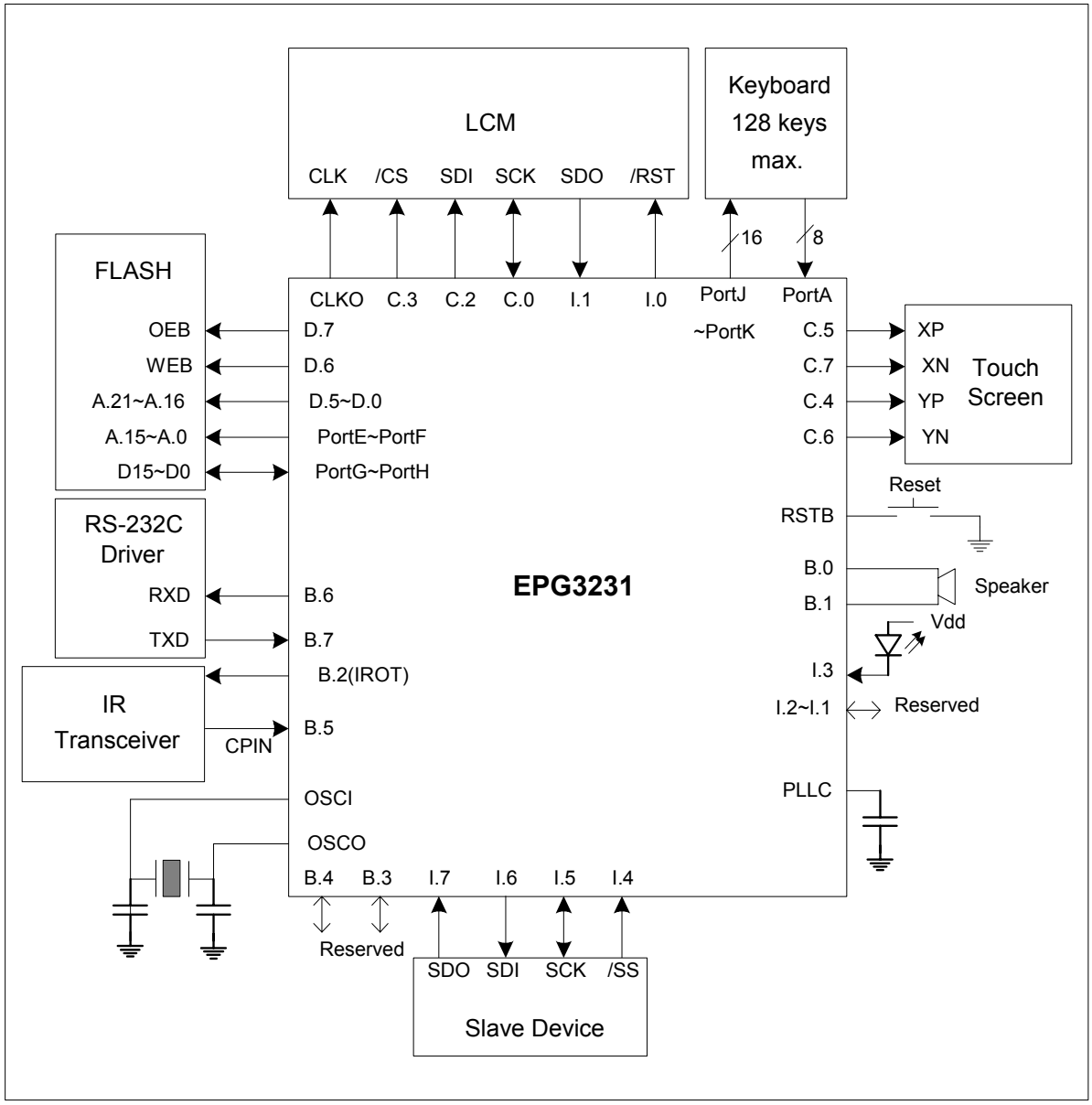


Fig. 8-33 Application Circuit Diagram

12 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 (*1)	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 11 i i rrrr rrrr	TBRD i,r	r←ROM[(TABPTR)]. (*2) (*3)	None	2
	0010 1111 rrrr rrrr	TBRD A,r	r←ROM[(TABPTR+ACC)]. (*3)	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 (*5)	C,DC,Z,OV,SGE,SLE	1
	0001 0111 rrrr rrrr	SUB r,A	r ← r-A (*5)	C,DC,Z,OV,SGE,SLE	1
	0100 1100 kkkk kkkk	SUB A,#k	A ← k-A (*5)	C,DC,Z,OV,SGE,SLE	1
	0001 1000 rrrr rrrr	SUBB A,r	A ← r-A-/C (*5)	C,DC,Z,OV,SGE,SLE	1
	0001 1001 rrrr rrrr	SUBB r,A	r ← r-A-/C (*5)	C,DC,Z,OV,SGE,SLE	1
	0100 1101 kkkk kkkk	SUBB A,#k	A ← k-A-/C (*5)	C,DC,Z,OV,SGE,SLE	1

Type	Instruction Binary	Mnemonic	Operation	Status Affected	Cycle
Arithmetic Operation (con't)	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. **(*6)**	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. **(*6)**	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. **(*6) (*7)**	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. (*6) (*7)	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. **(*6) (*7)**	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. **(*6) (*7)**	None	2
	0100 0111 kkkk kkkk aaaa aaaa aaaa aaaa	JGE A,#k,addr	Jump to addr if A□k; PC [15:0] ← addr. **(*6)**	None	2
	0100 1000 kkkk kkkk aaaa aaaa aaaa aaaa	JLE A,#k,addr	Jump to addr if A□k; PC [15:0] ← addr. **(*6)**	None	2
	0100 1001 kkkk kkkk aaaa aaaa aaaa aaaa	JE A,#k,addr	Jump to addr if A=k; PC[15:0] ← addr. **(*6)**	None	2

Type	Instruction Binary	Mnemonic	Operation	Status Affected	Cycles
Compare & Jump (con't)	0101 0101 rrrr rrrr aaaa aaaa aaaa aaaa	JGE A,r,addr	Jump to addr if A≧r; PC[15:0] ← addr. (*6)	None	2
	0101 0110 rrrr rrrr aaaa aaaa aaaa aaaa	JLE A,r,addr	Jump to addr if A≦r; PC[15:0] ← addr. (*6)	None	2
	0101 0111 rrrr rrrr aaaa aaaa aaaa aaaa	JE A,r,addr	Jump to addr if A=r; PC[15:0] ← addr. (*6)	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 (*8)	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) "**r**" is the repeat value of Register r which determines the number of times the next instruction has to be repeated.

(*2) TBRD i, r:
r← ROM [(TABPTR)];
i=00: TABPTR no change
i=01: TABPTR←TABPTR+1
i=10: TABPTR←TABPTR-1

(*3) TABPTR=(TABPTRH: TABPTRM: TABPTRL)
Bit0=0**:** Low byte of the pointed ROM data
Bit0=1**:** High byte of the pointed ROM data

NOTE

- *Bit 0 of TABPTRL is used to select low byte or high byte of the pointed ROM data.*
- *The maximum table look up space is internal 8Mbytes.*

(*4) Carry bit of ADD PCL, A or ADD TABPTRL, A will automatically carry into PCM or TABPTRM. The Instruction cycle of write to PC (program counter) takes 2 cycles.

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

(*6) The maximum jump range is 64K absolute address; means only can jump within the same 64K range.

(*7) Do not use JDNZ & JINZ at FSR1 (09B) special register.

(*8) S0CALL address ability is from 0x000 to 0xFFF (4K space).

13 Pad Diagram and Locations

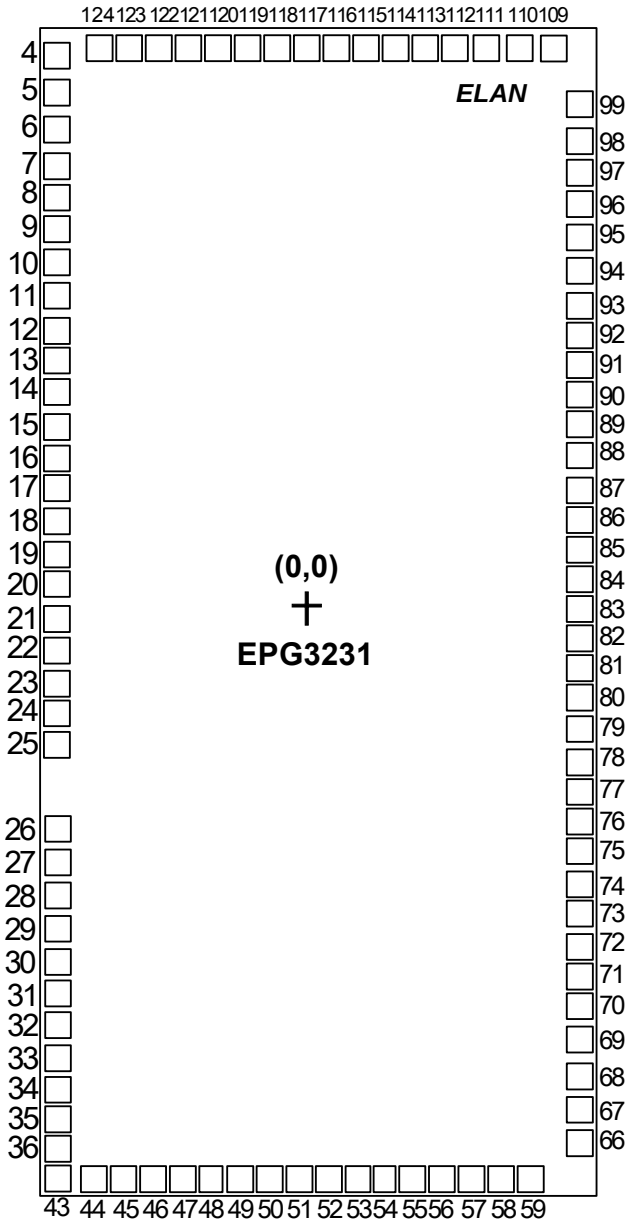


Fig. 8-34 ePG3231 Pad Locations

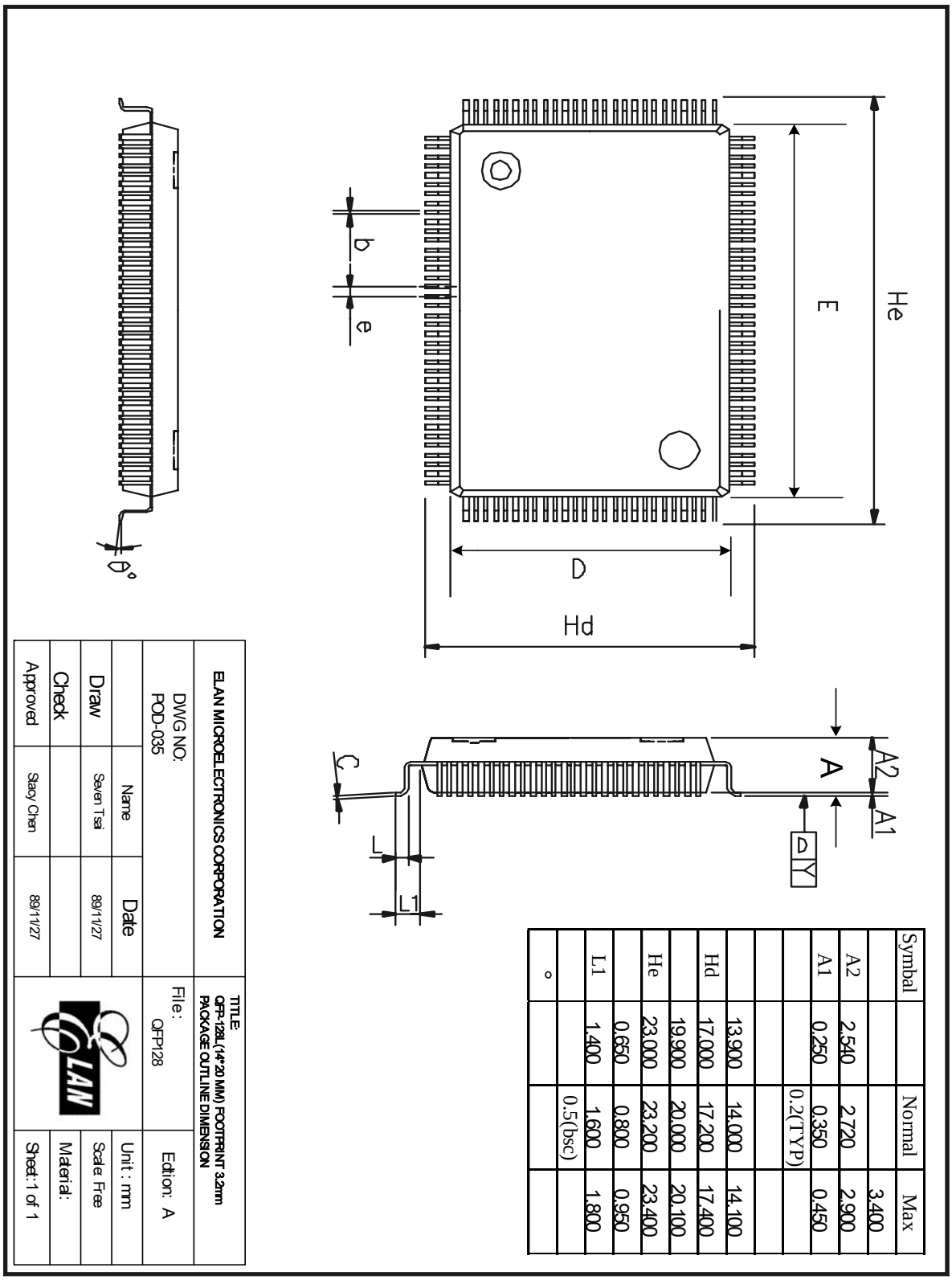
Chip Size: 2050 * 4180 μm^2

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

Pin No.	Symbol	X	Y	Pin No.	Symbol	X	Y
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
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		

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

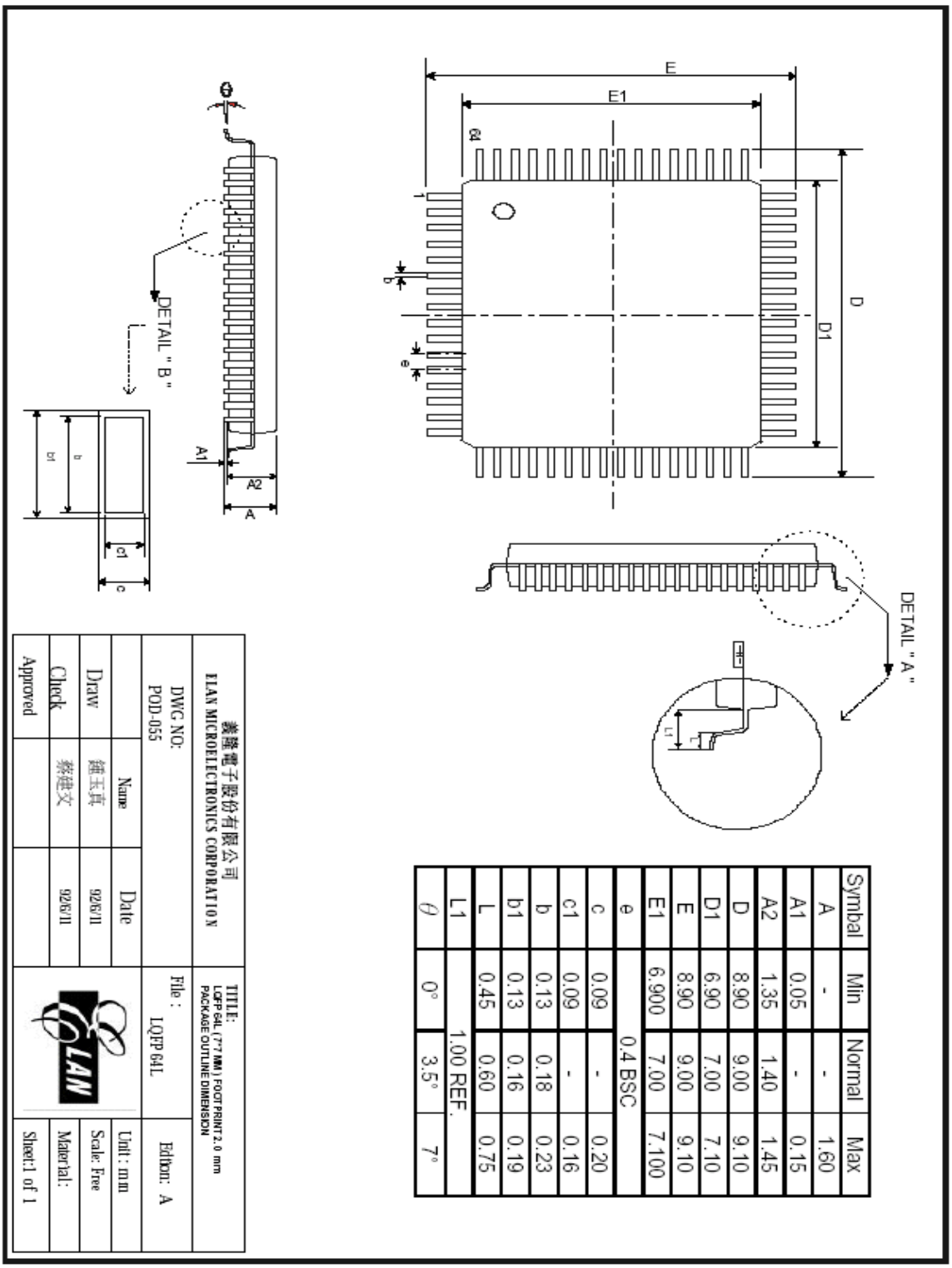
14 Package



Symbal		Normal	Max
			3.400
A2	2.540	2.720	2.900
A1	0.250	0.350	0.450
		0.2(TYP)	
	13.900	14.000	14.100
Hd	17.000	17.200	17.400
	19.900	20.000	20.100
He	23.000	23.200	23.400
	0.650	0.800	0.950
L1	1.400	1.600	1.800
		0.5(bsc)	
°			

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

Fig. 8-35 ePG3231 Package QFP 128 pin Outline Dimension



Symbal	Min	Normal	Max
A	-	-	1.60
A1	0.05	-	0.15
A2	1.35	1.40	1.45
D	8.90	9.00	9.10
D1	6.90	7.00	7.10
E	8.90	9.00	9.10
E1	6.900	7.00	7.100
e	0.4 BSC		
c	0.09	-	0.20
c1	0.09	-	0.16
b	0.13	0.18	0.23
b1	0.13	0.16	0.19
L	0.45	0.60	0.75
L1	1.00 REF.		
θ	0°	3.5°	7°

義隆電子股份有限公司
ELAN MICROELECTRONICS CORPORATION

TITLE: LQFP 64L (7*7 MM) FOOT PRINT 2.0 mm PACKAGE OUTLINE DIMENSION

DWG NO: POD-055

File : LQFP 64L

Edition: A

	Name	Date
Draw	鍾玉真	9/26/11
Check	蔡建文	9/26/11
Approved		

Unit : mm

Scale: Free

Material:

Sheet:1 of 1

Fig. 8-36 ePG3231 Package LQFP 64 pin Outline Dimension