

自带12位A/D转换器及 LCD 驱动 8位单片机



功能特性

- 高速 8051 CPU内核8位单片机
- MCU 基本功能
 - 32K字节闪存代码存储器
 - 1024字节SRAM
- 内置模拟功能
 - 上电复位和低电压检测复位
 - 内部 16MHz RC 振荡器 ($\pm 1.5\%$, $T_A = 0 \sim +50^\circ\text{C}$)
 - 看门狗定时器 RC 振荡器 (5kHz)
- 外围设备
 - 12-bit AD转换器(16inputs)
 - USI (USART + SPI + I2C) 2组
 - LCD 驱动 (21seg x 8com)
- I/O 和封装
 - 多达 42 可编程 I/O (44 MQFP-1010)
 - 32 LQFP
 - 32/28 SOP
- 工作条件
 - 1.8V - 5.5V 电压范围
 - -40°C - 85°C 温度范围
- 应用
 - 小家电
 - 无刷直流电机控制

MC96F6432A

MC96F6332A

用户手册

V 1.7

修订记录

Version	Date	Revision list
0.0	2015.09.18	Published this book.
0.1	2015.12.16	Change symbol name from ILE, DLE, FSE, t_{CON} , V_{AN} , I_{AN} to INL, DNL, TOE, t_{CONV} , V_{AIN} , I_{AIN} in 7.3 A/D Converter Characteristics Fix typos.
1.0	2016.02.03	Add a "MC96F6432AU" 48 QFN package. Add IRC Max 16MHz in 7.6 High Internal RC Oscillator Characteristics. Add a note about IRC frequency in 11.1 Clock Generator.
1.1	2016.03.25	Modify 48-Pin QFN Package Diagram
1.2	2016.04.08	Add a "Full-flash erase mode method" in different table. Remove the note about "direct bit test and branch instructions", Modify the program tips in Chapter 15. Flash Memory. Add an appendix about "Flash Protection for invalid Erase/Write"
1.3	2016.05.26	Change '8.5MHz (Max)' to "Main X-TAL Operating Frequency" in Electrical Characteristics.
1.4	2016.05.30	Add a "MC96F6332AL" 32LQFP package.
1.5	2016.07.06	Remove the 28TSSOP package. Change Gang programmer from StandAlone GANG8 to E-GANG4/E-GANG6. Fixed typos of I2C Status Register.
1.6	2017.02.09	Revised this book. Updated Package diagrams in Chapter 4. Package Diagram. Added the note on the flash memory erase and write in Chapter 15. Flash Memory. Updated OCD dongle image in Chapter 1.3 Development tools.
1.7	2018.06.22	Revised this book. Fixed typos. Updated All Package Diagram.

版本1.7

由 FAE 团队发行

版权所有 2018 ABOV半导体公司.

本手册的附加信息可能会提供给 ABOV 办事处及分销商。

ABOV半导体公司有权修改任何信息在任何时间，恕不另行通知。

本手册中的信息，图表和其他数据是正确而且可靠的；

但是，在使用本手册过程中任何违反专利或第三方权利的纠纷，ABOV 公司概不负责。

1 综述

1.1. 说明

MC96F6432A 是一款自带32K 字节闪存的8位 CMOS 单片机. 可以为许多嵌入式控制应用提供高效灵活低成本解决方案。具有以下特点: 32K 字节 FLASH, 256 字节 IRAM, 768 字节 XRAM, 通用 I/O, 基本间隔定时器, 看门狗定时器 8/16-bit 定时器/计数器, 16-bit PPG 输出, 8-bit PWM 输出, 10-bit PWM 输出, 实时定时器, 蜂鸣器驱动口, SPI, USI, 12-bit A/D 转换器, LCD 驱动, 片内 POR, LVR, LVI, 片内振荡器和时钟电路. MC96F6432A 具有节能省电模式以减少电能消耗.

型号	FLASH	XRAM	IRAM	ADC	I/O 口	封装
MC96F6432AU	32Kbytes	768bytes	256bytes	16 inputs	42	48 QFN
MC96F6432AQ				16 inputs	42	44 MQFP
MC96F6332AL				12 inputs	30	32 LQFP
MC96F6332AD				12 inputs	30	32 SOP
MC96F6332AM				11 inputs	26	28 SOP

Table 1.1 MC96F6432A选型信息

1.2 性能

- CPU
 - 8-bit CISC 内核 (M8051, 2 时钟/周期)
- ROM (FLASH) 容量
 - 自读写32K字节 Flash
 - ISP (In-System Programming)
 - 擦写次数 : 100,000次(Sector 0~507)
100,000次(Sector 508~511)
 - 保存: 10年
- 256bytes IRAM
- 768bytes XRAM
 - (27bytes 包含 LCD 显示 RAM)
- 通用 I/O (GPIO)
 - 普通 I/O : 9个 (P0[2:0], P5[5:0])
 - LCD 共用 I/O : 33个 (P0[7:3], P1, P2, P3, P4)
- 定时器/计数器
 - Basic Interval Timer (BIT) 8-bit × 1-ch
 - Watch Dog Timer (WDT) 8-bit × 1-ch
 - 看门狗使用 5kHz 内部 RC振荡器
 - 8-bit × 1-ch (T0), 16-bit × 2-ch (T1/T2)
 - 8-bit × 2-ch (T3/T4) or 16-bit × 1-ch (T3)
- 可编程脉冲发生器
 - 脉冲发生器 (by T1/T2)
 - 8-bit PWM (by T0)
 - 6-ch 10-bit 马达 PWM (by T4)
- Watch Timer (WT)
 - 内部32.768kHz 3.91ms/0.25s/0.5s/1s /1min间隔
- 蜂鸣器
 - 8-bit × 1-ch
- SPI 2
 - 8-bit × 1-ch
- USI0/1 (UART + SPI + I2C)
 - 8-bit UART × 2-ch, 8-bit SPI × 2-ch和I2C × 2-ch
- 12-bit A/D 转换器
 - 16 路输入通道
- LCD 驱动
 - 21seg x 8com
 - 内部或外部偏压
 - 两组内部电阻可选
 - 1/2, 1/3, 1/4, 1/5, 1/6 和1/8 duty 可选
 - 电阻偏压和 16-step 可调
- 上电复位
 - 复位电平(1.4V)
- 低电压复位
 - 12 种电平可选
(1.85/ 2.20/ 2.32/ 2.44/ 2.59/ 2.75/ 2.93/ 3.14/
3.38/ 3.67/ 4.00/ 4.40V)
- 低电压检测
 - 11 levels detect
(2.20/ 2.32/ 2.44/ 2.59/ 2.75/ 2.93/ 3.14/ 3.38/
3.67/ 4.00/ 4.40V)
- 中断源
 - 外部中断
 - (EINT0~7, EINT8, EINT10, EINT11, EINT12) (12)
 - Timer(0/1/2/3/4) (5)
 - WDT (1)
 - BIT (1)
 - WT (1)
 - SPI 2 (1)
 - USI0/1 (6)
 - ADC (1)
- 内部 RC 振荡器
 - 内部 RC 频率: 16MHz ±1.5% (TA= 0 ~ +50°C)
- 省电模式
 - STOP, IDLE 模式
- 工作电压 和 频率
 - 2.2V~ 5.5V (@32 ~ 38kHz with Crystal)
 - 2.2V~ 5.5V (@0.4 ~ 4.2MHz with Crystal)
 - 2.7V~ 5.5V (@0.4 ~ 8.5MHz with Crystal)
 - 2.2V~ 5.5V (@0.5 ~ 16.0MHz with Internal RC)
 - Voltage dropout converter included for core
- 最小指令执行时间
 - 235ns (@8.5MHz main clock)
 - 61us (@ 32.768kHz sub clock)
- 工作温度
 - -40 ~ +85°C
- 振荡器类型
 - 0.4 - 8.5MHz 晶体或陶振作为主时钟
 - 32.768kHz 晶体作为副时钟
- 封装类型
 - 48 QFN
 - 44 MQFP-1010
 - 32 LQFP, 32 SOP
 - 28 SOP
 - Pb-free package

1.3 开发工具

1.3.1 编译器

ABOV 不提供编译器.建议咨询编译器提供商

MC96F6432A 是曼托 (Mentor) 8051内核, ROM 大小低于64K字节.因此, 开发人员可以使用 8051 标志的其他提供商的编译器.

1.3.2 OCD(On-chip debugger) 调试器

OCD (On Chip Debug) 仿真器支持 ABOV 公司 8051 系列 MCU 仿真. OCD 接口通过两根线连接 PC 和用户系统所用 MCU. OCD 可以读到或更改 MCU 内部存储器和外围 I/O. 同时 OCD 也控制 MCU 内部仿真逻辑, 包括控制仿真, 单步运行, 监测等.

OCD 调试程序安装 Microsoft-Windows NT, 2000, XP, Vista (32-bit) 操作系统.如果想要了解更详细的信息, 请参考 OCD 调试手册。 可以从我们的网站下载仿真器 SW 和手册 (<http://www.abov.co.kr>).

连接:

- DSCL (MC96F6432 P01 口)
- DSDA (MC96F6432 P00 口)

NOTE)

1. MC96F6432A 不支持 OCD 调试功能. MC96F6432 用来作为调试仿真使用.
2. 请勿直接 MC96F6432 中的位指令和分支指令, 来访问 P0, P1, P2, P3, P4, P5 和 EIFLAG0 寄存器. 请参考 "MC96F6432 用户手册".

OCD 连接图: 连接 OCD到用户系统

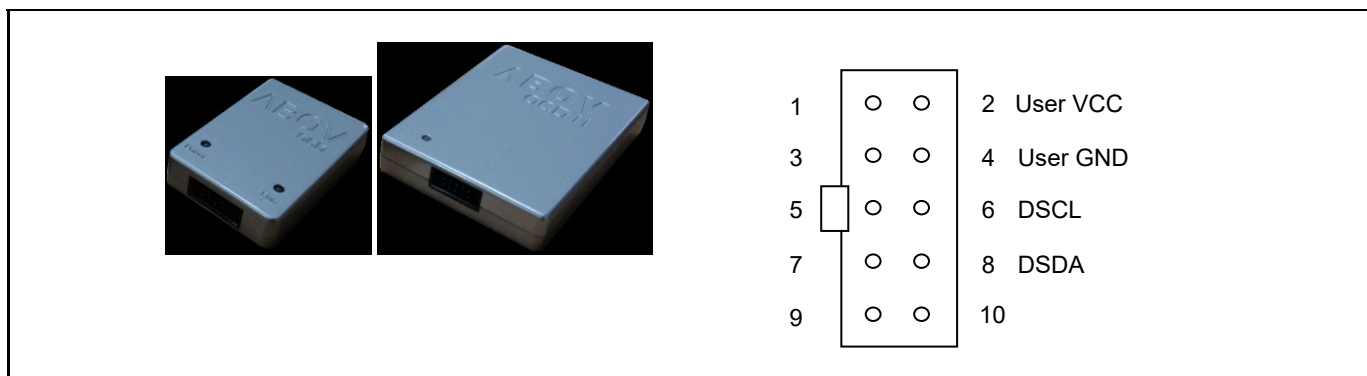


Figure 1.1 调试工具(OCD1/OCD2)连接说明

Subject	MC96F6432A	MC96F6432S	MC96F6432
On Chip Debugger (OCD)	Not supported	Not supported	Supported
Internal RC frequency (IRC)	Tolerance TA= 0°Cto+50°C : ±1.5% TA= -20°Cto+85°C : ±2.5% TA= -40°Cto+85°C : ±3.5%	Tolerance TA= 0°Cto+50°C : ±1.5% TA= -20°Cto+85°C : ±2.5% TA= -40°Cto+85°C : ±3.5%	Tolerance TA= 0°Cto+50°C : ±1.0% TA= -20°Cto+85°C : ±2.0% TA= -40°Cto+85°C : ±3.0%
x-tal filter selectable (XTFLSR Register)	x-tal filter selection register for noise immunity - 4.2MHz < x-tal ≤ 8.5MHz - x-tal ≤ 4.2MHz	Not supported	Not supported
LVR/LVI	12 Level Selectable - 1.85V, 2.2V,,,,, 4.4V	14 Level Selectable - 1.6V, 2.0V, 2.1V,,,,, 4.4V	14 Level Selectable - 1.6V, 2.0V, 2.1V,,,,, 4.4V
	11 Level Selectable - 2.2V,,,,, 4.4V	13 Level Selectable - 2.0V, 2.1V,,,,, 4.4V	13 Level Selectable - 2.0V, 2.1V,,,,, 4.4V
	LVR and LVI Current Both : 14.0(TYP)/24.0(MAX) uA One: 10.0(TYP)/18.0(MAX) uA	LVR and LVI Current Both : 14.0(TYP)/24.0(MAX) uA One: 10.0(TYP)/18.0(MAX) uA	LVR and LVI Current Both : 10.0(TYP)/15.0(MAX) uA One: 8.0(TYP)/12.0(MAX) uA
LCD	Internal LCD Bias Dividing Resistor Select - RLCD1 = 60kΩ - RLCD2 = 120kΩ	Internal LCD Bias Dividing Resistor Select - RLCD1 = 60kΩ - RLCD2 = 120kΩ	Internal LCD Bias Dividing Resistor - RLCD = 60kΩ
	RLCD1 - 40(MIN)/60(TYP)/80(MAX) kΩ RLCD2 - 80(MIN)/120(TYP)/160(MAX) kΩ	RLCD1 - 40(MIN)/60(TYP)/80(MAX) kΩ RLCD2 - 80(MIN)/120(TYP)/160(MAX) kΩ	RLCD - 40(MIN)/60(TYP)/80(MAX) kΩ
Vector Area Protection	En/Disable Vector Area(00H – FFH) Protection	Not supported	Not supported
Specific Area for Write Protection	8 kinds of protection size selectable - Address 0100H – 03FFH - Address 0100H – 07FFH - Address 0100H – 0BFFH - Address 0100H – 0FFFH - Address 0100H – 77FFH - Address 0100H – 7BFFH - Address 0100H – 7DFFH - Address 0100H – 7EFFH	4 kinds of protection size selectable - Address 0100H – 0FFFH - Address 0100H – 07FFH - Address 0100H – 77FFH - Address 0100H – 03FFH - Address 0100H – 01FFH	4 kinds of protection size selectable - Address 0100H – 0FFFH - Address 0100H – 07FFH - Address 0100H – 03FFH - Address 0100H – 01FFH
ADC	A/D Converter INL : ±6LSB DNL : ±1 LSB TOE : ±5 LSB ZOE : ±5 LSB	A/D Converter INL: ±6LSB DNL: ±1 LSB TOE: ±5 LSB ZOE: ±5 LSB	A/D Converter INL: ±4LSB DNL: ±1 LSB TOE: ±3 LSB ZOE: ±3 LSB
Supply Current IDD3 (Sub operation)	Typ/Max : 90/180 [uA] - 32.768kHz, VDD=3V±10%, TA=25°C	Typ/Max : 60/90 [uA] - 32.768kHz, VDD=3V±10%, TA=25°C	Typ/Max : 50/80 [uA] - 32.768kHz, VDD=3V±10%, TA=25°C
Operating Voltage and Frequency	VDD: 2.2V to 5.5V, Freq.: Up to 16MHz - 2.2V to 5.5V @ 32 to 38kHz with x-tal - 2.2V to 5.5V @ 0.4 to 4.2MHz with x-tal - 2.7V to 5.5V @ 0.4 to 8.5MHz with x-tal - 2.2V to 5.5V @ 0.5 to 16MHz with IRC	VDD: 1.8V to 5.5V, Freq.: Up to 16MHz - 1.8V to 5.5V @ 32 to 38kHz with x-tal - 1.8V to 5.5V @ 0.4 to 4.2MHz with x-tal - 2.7V to 5.5V @ 0.4 to 10MHz with x-tal - 3.0V to 5.5V @ 0.4 to 12MHz with x-tal - 1.8V to 5.5V @ 0.5 to 8MHz with IRC - 2.0V to 5.5V @ 0.5 to 16MHz with IRC	VDD: 1.8V to 5.5V, Freq.: Up to 16MHz - 1.8V to 5.5V @ 32 to 38kHz with x-tal - 1.8V to 5.5V @ 0.4 to 4.2MHz with x-tal - 2.7V to 5.5V @ 0.4 to 10MHz with x-tal - 3.0V to 5.5V @ 0.4 to 12MHz with x-tal - 1.8V to 5.5V @ 0.5 to 8MHz with IRC - 2.0V to 5.5V @ 0.5 to 16MHz with IRC
Power-On Reset Characteristics	VDD Voltage Rising Time Min/Max : 0.05/30.0[V/ms]	VDD Voltage Rising Time Min/Max : 0.05/30.0[V/ms]	VDD Voltage Rising Time Min/Max : 0.05/5.0[V/ms]
Full-flash erase mode method	Sector erase mode	Sector erase mode	Sector and byte erase mode

Table 1.2 MC96F6432A, MC96F6432A and MC96F6432 之间的差异

1.3.3 烧写器

单一烧写器:

E-PGM+: 可以直接烧写 MCU.

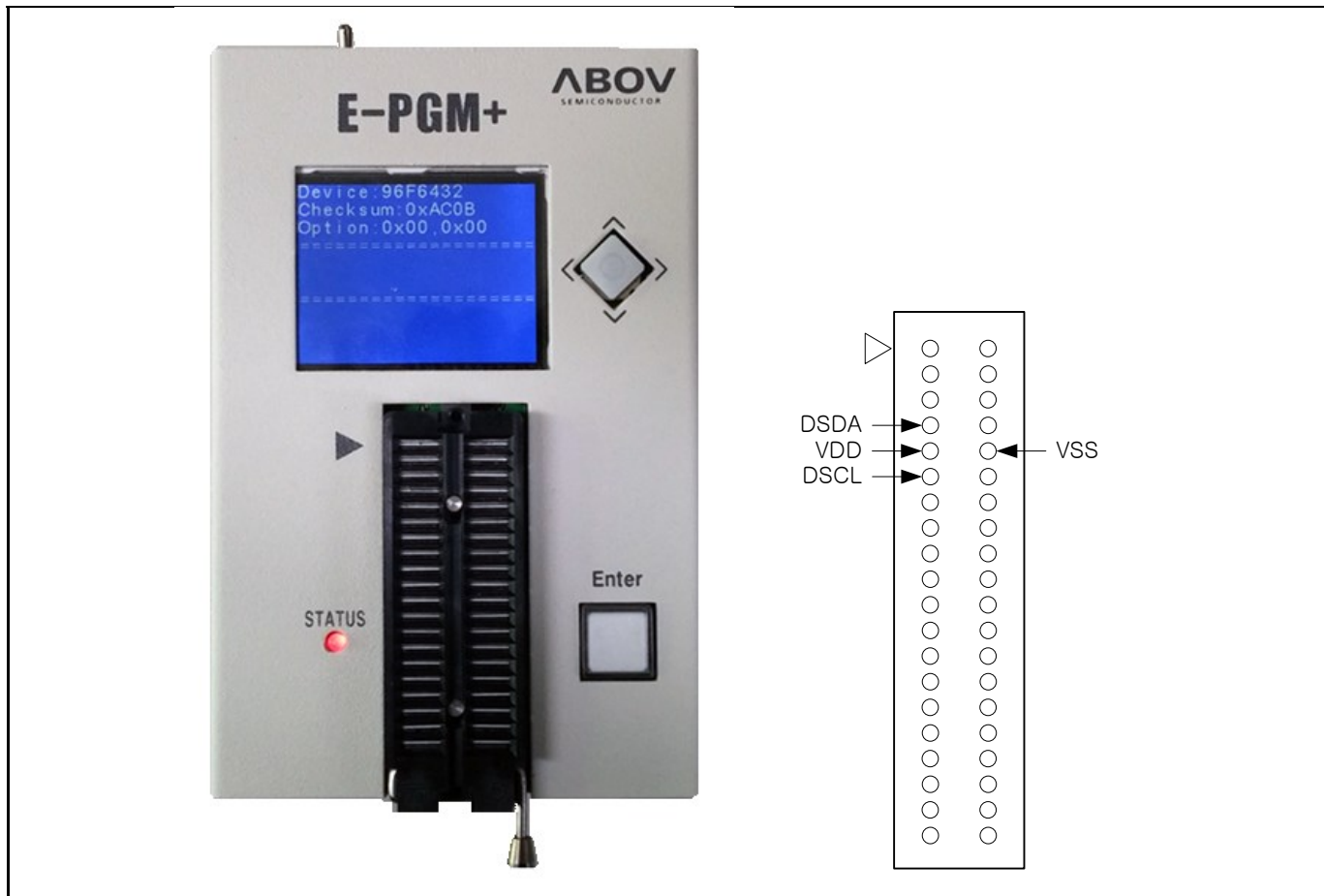


Figure 1.2 E-PGM+ (单一烧写器)

群烧写器: E-GANG4 和 E-GANG6

- 它能连接电脑运行。
- 它可以脱机烧写，在代码加载到烧写器之后无需和电脑连接。
- 支持USB接口。
- 非常容易连接硬件。



Figure 1.3 E-GANG4 and E-GANG6 (用于大量生产)

1.4 MTP 烧写

1.4.1 概述

MC96F6432A 程序存储器属于 MTP 类型. 这种存储器采用串行数据格式存储. 需要四根连接线 (DSCL, DSDA, VDD, 和 VSS) 来烧写和读取.

Pin name	Main chip pin name	在烧写过程中	
		I/O	说明
DSCL	P01	I	串行时钟脚. 仅作为输入口.
DSDA	P00	I/O	串行数据脚. 在读取过程中输出口, 在烧写过程中输入口. 可以指定为输入/上拉 说出口.
VDD, VSS	VDD, VSS	-	提供电源.

Table 1.3 用于烧写/读取 Flash 时端口说明

1.4.2 在线烧录

MC96F6432A 只需要串行协议四根线包括VDD 和 VSS 烧写 FLASH. 所以如果PCB板设计时已经设计好烧写信号线是可以在线烧录的.

1.4.2.1 电路设计指导

在 FLASH 烧写过程中, 烧录工具需要4根信号线 DSCL, DSDA, VDD, 和 VSS. 在设计PCB 电路时, 你需要考虑这些信号线在线烧录时的使用.

请务必注意设计这些信号线相关的电路, 因为DSCL 和 DSDA 的时序对烧录非常重要.

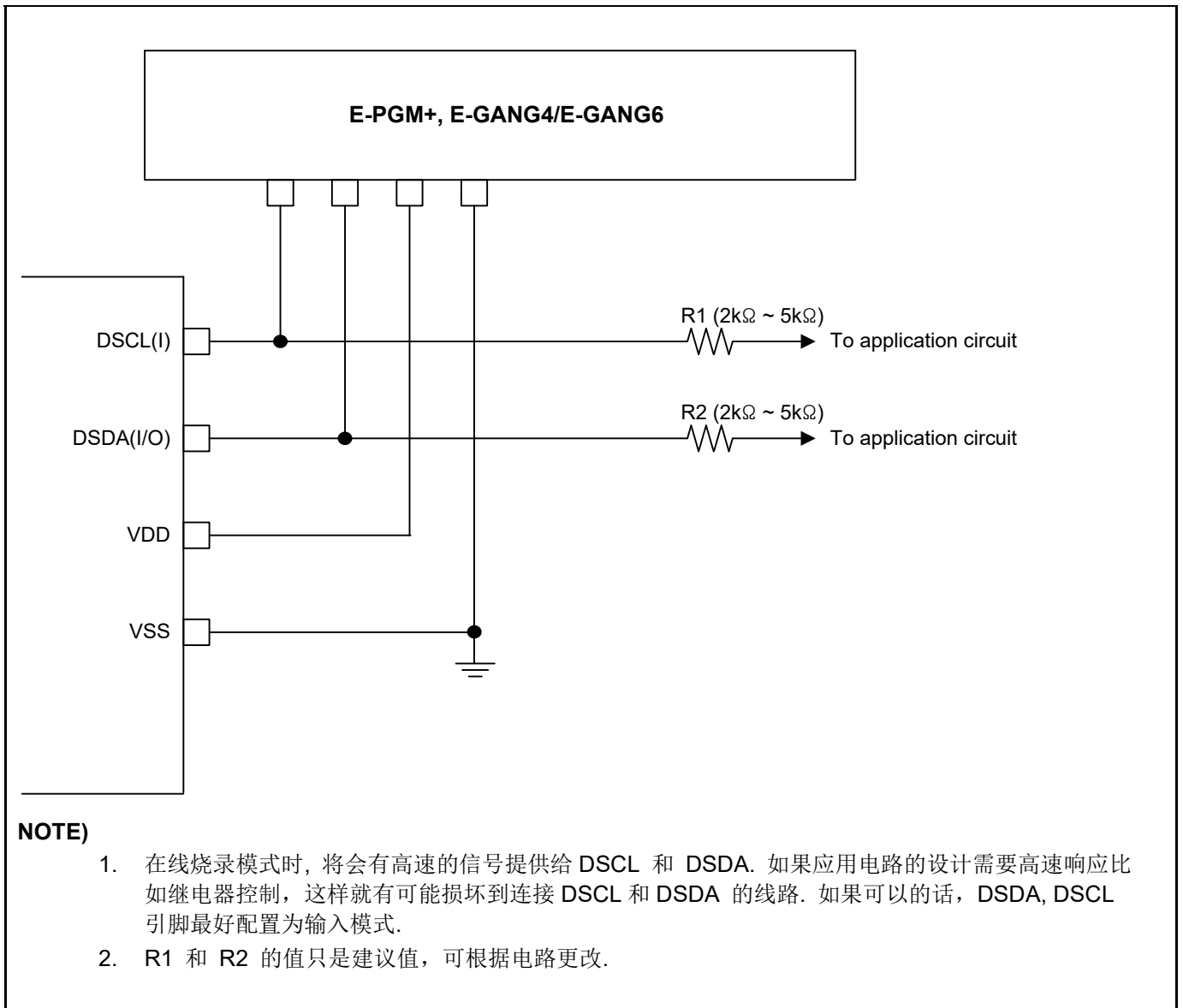


Figure 1.4 在线烧录 PCB 设计指导

2 方框图

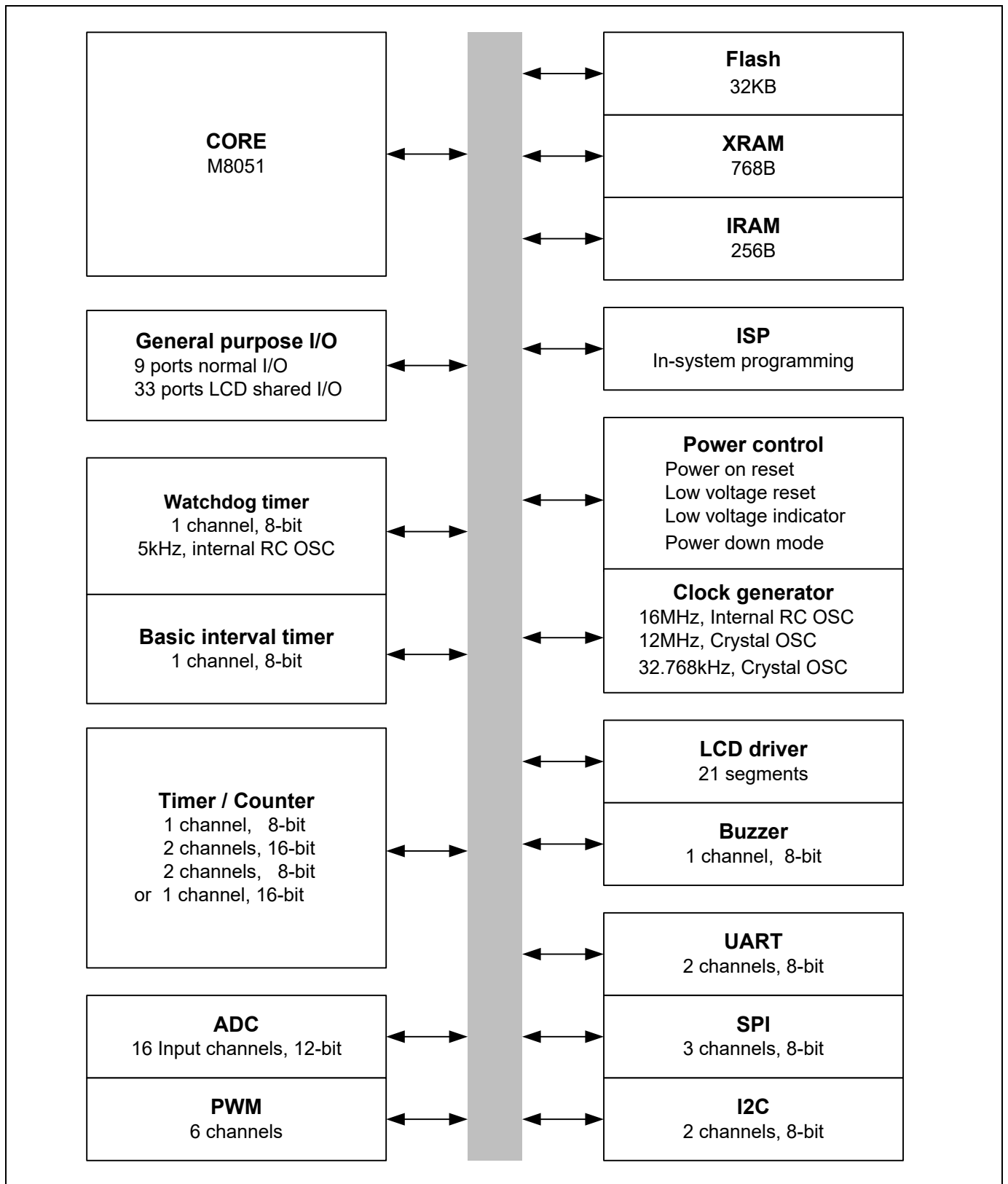
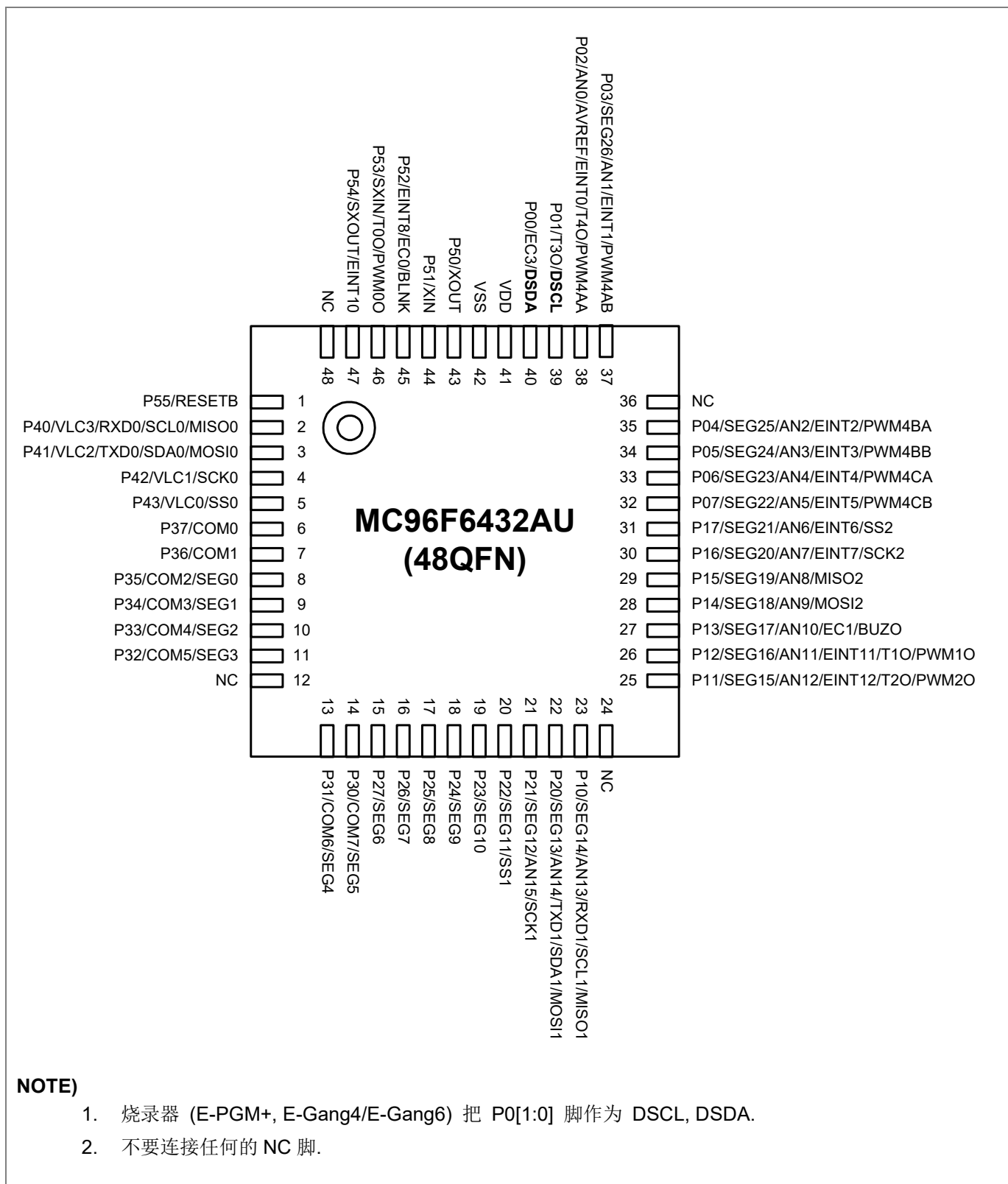


Figure 2.1 MC96F6432A 方框图

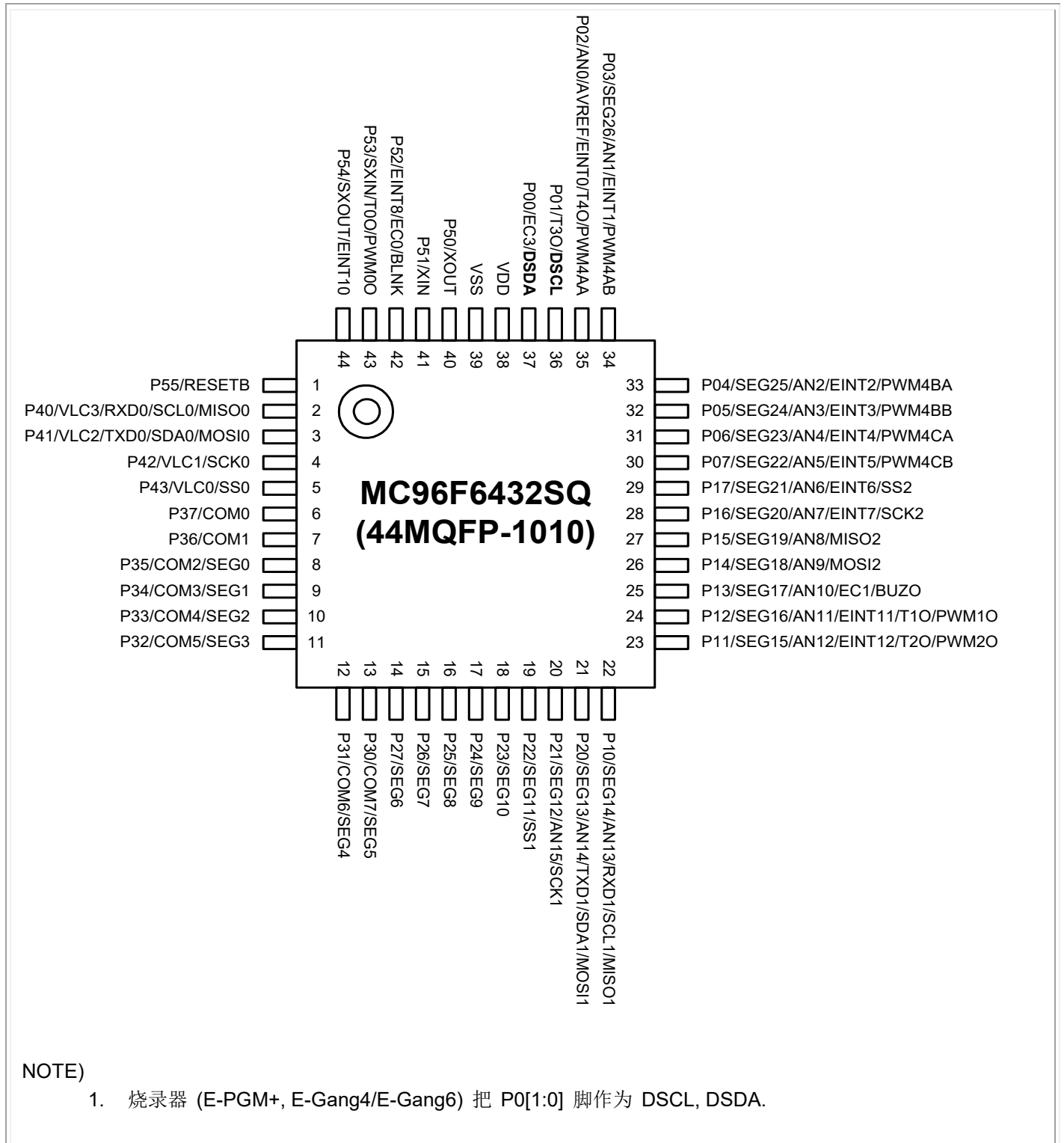
3 引脚分配



NOTE)

1. 烧录器 (E-PGM+, E-Gang4/E-Gang6) 把 P0[1:0] 脚作为 DSCL, DSDA.
2. 不要连接任何的 NC 脚.

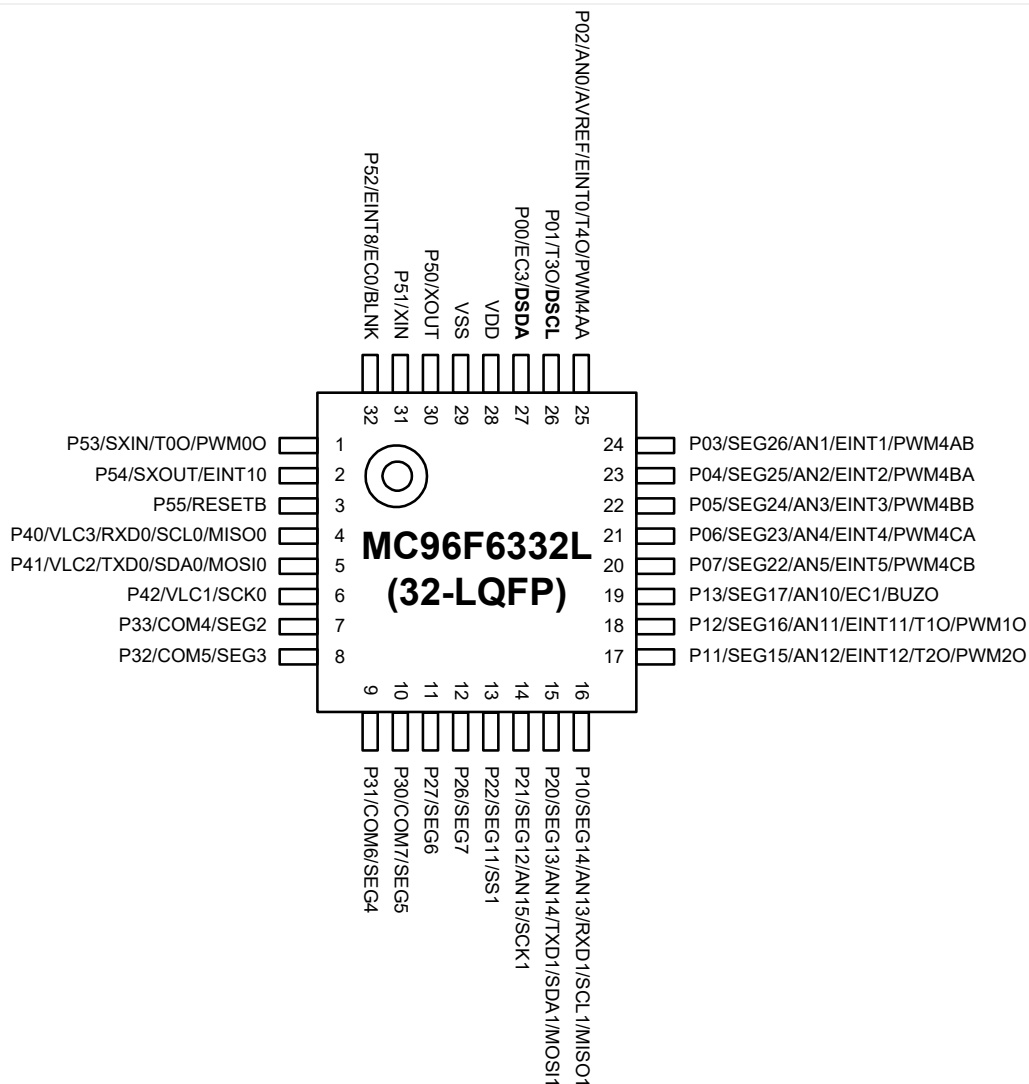
Figure 3.1 MC96F6432AU 48QFN 引脚分配



NOTE)

1. 烧录器 (E-PGM+, E-Gang4/E-Gang6) 把 P0[1:0] 脚作为 DSCL, DSDA.

Figure 3.2 MC96F6432AQ 44MQFP-1010 引脚分配



NOTE)

1. 烧录器 (E-PGM+, E-Gang4/E-Gang6) 把 P0[1:0] 脚作为 DSCL, DSDA.
2. 使用 32 脚封装时 P14-P17, P23-P25, P34-P37 和 P43 应该通过软件控制选择为 push-pull 输出或上拉输入.

Figure 3.3 MC96F6332AL 32LQFP-0707 引脚分配

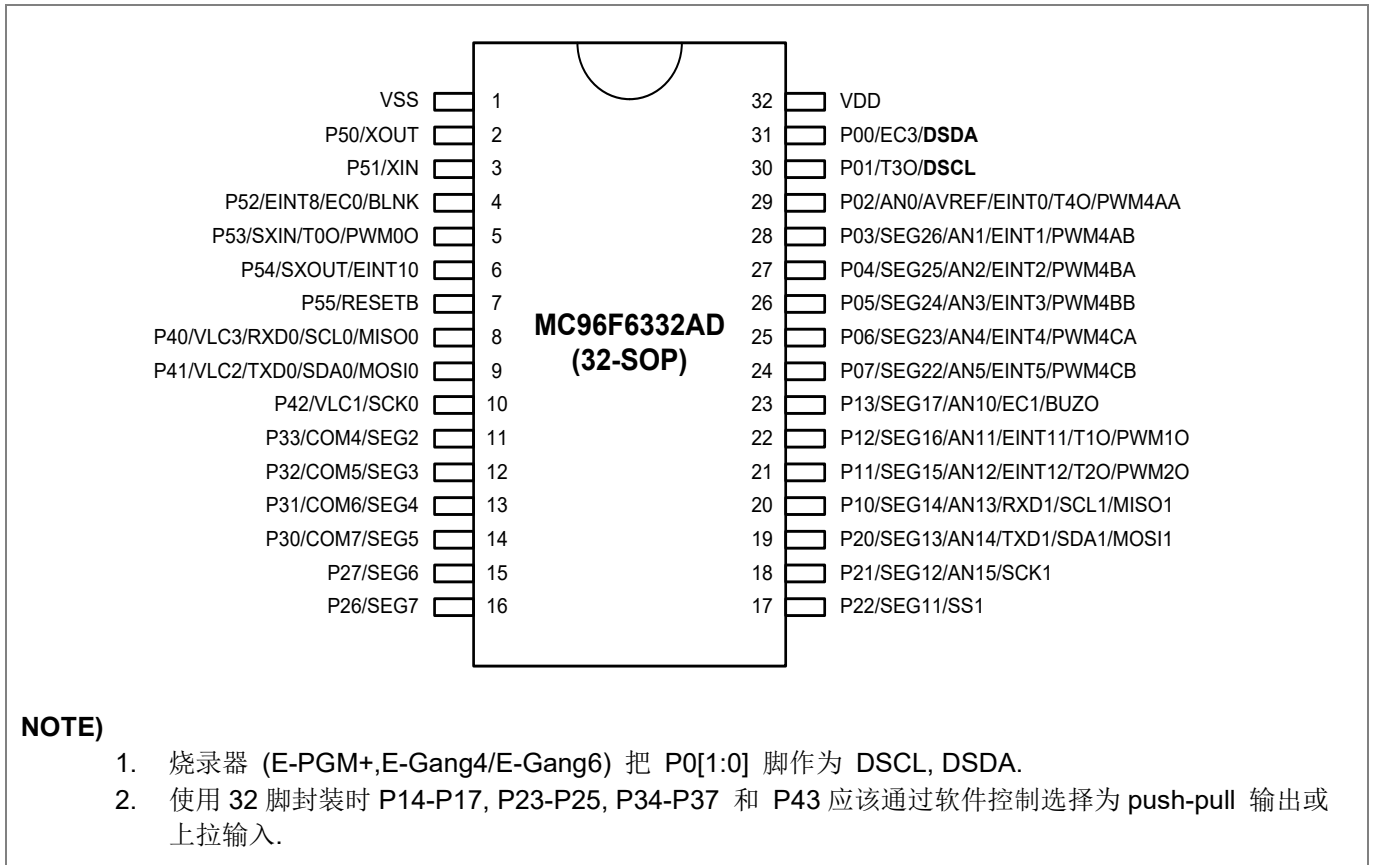


Figure 3.4 MC96F6332AD 32SOP 引脚分配

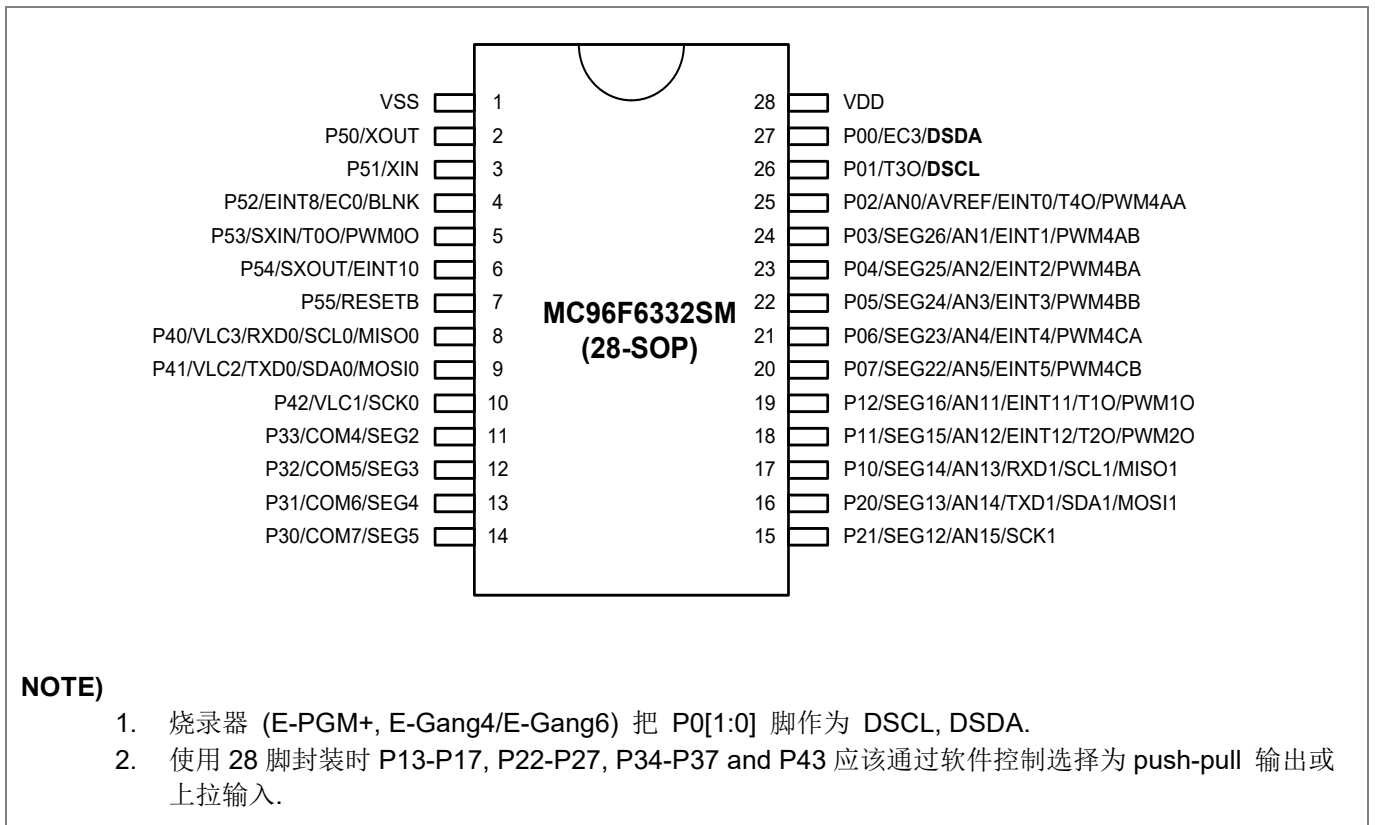


Figure 3.5 MC96F6332AM 28SOP 引脚分配

4 封装图

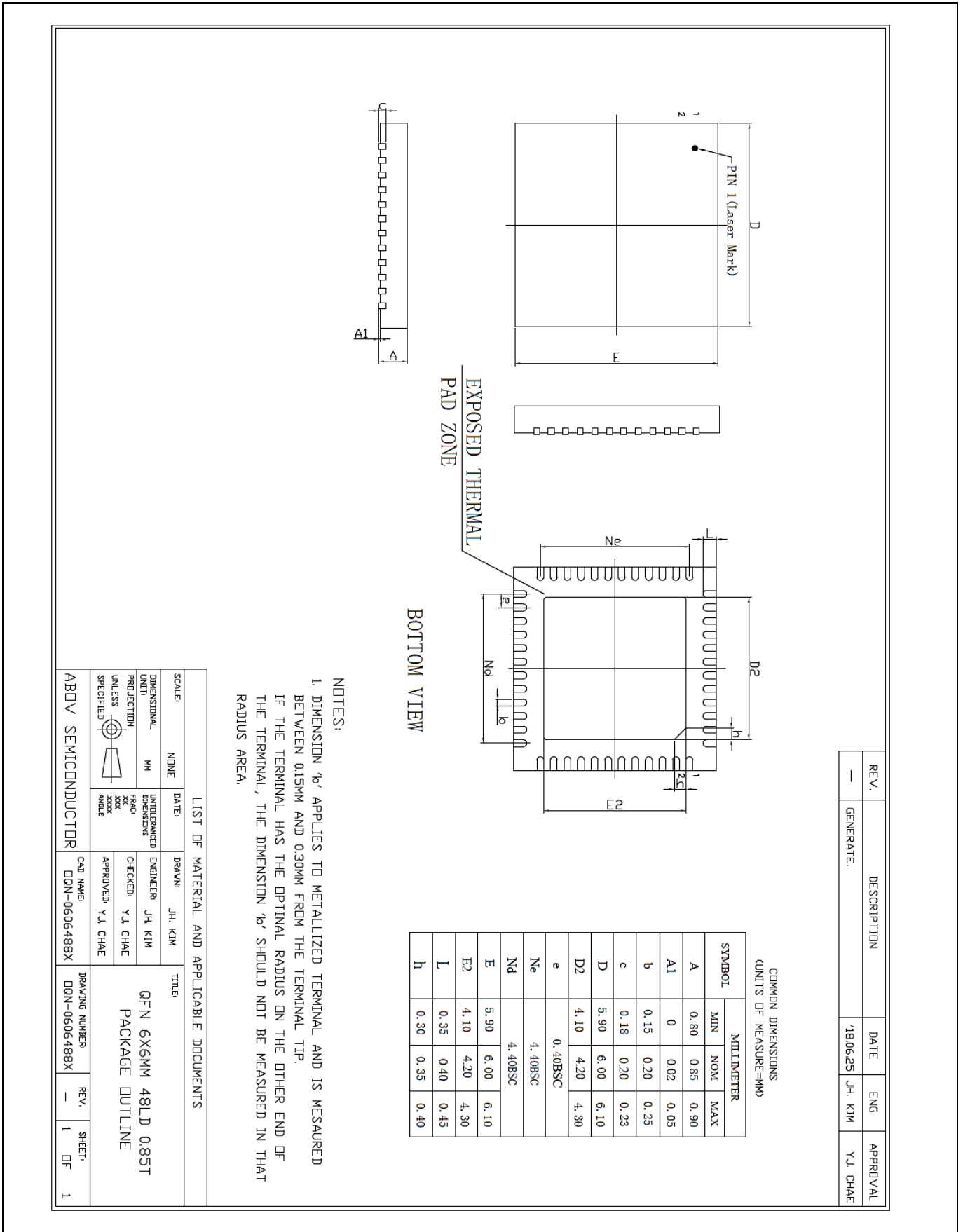


Figure 4.1 48-Pin QFN 封装

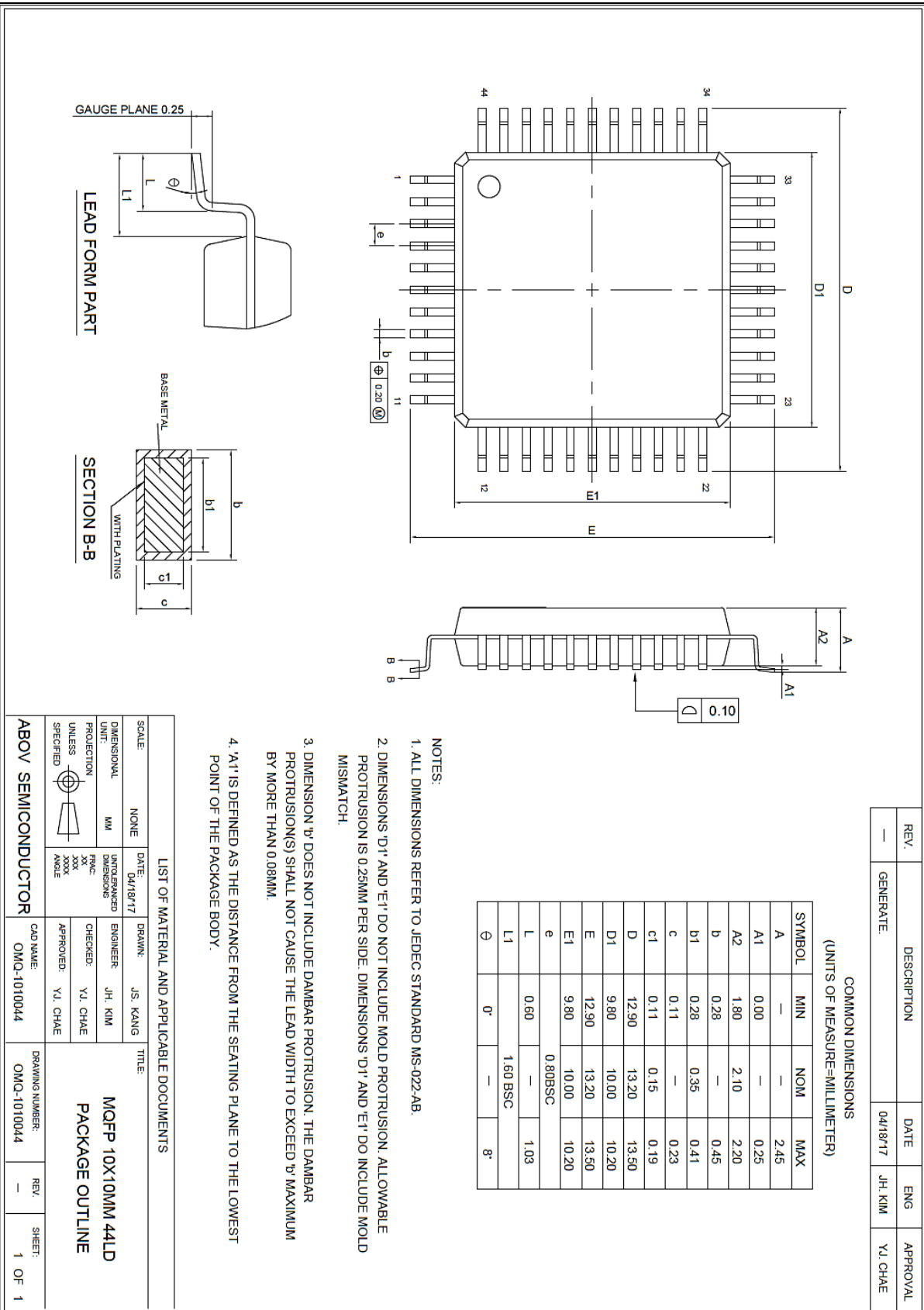


Figure 4.2 44-Pin MQFP 封装

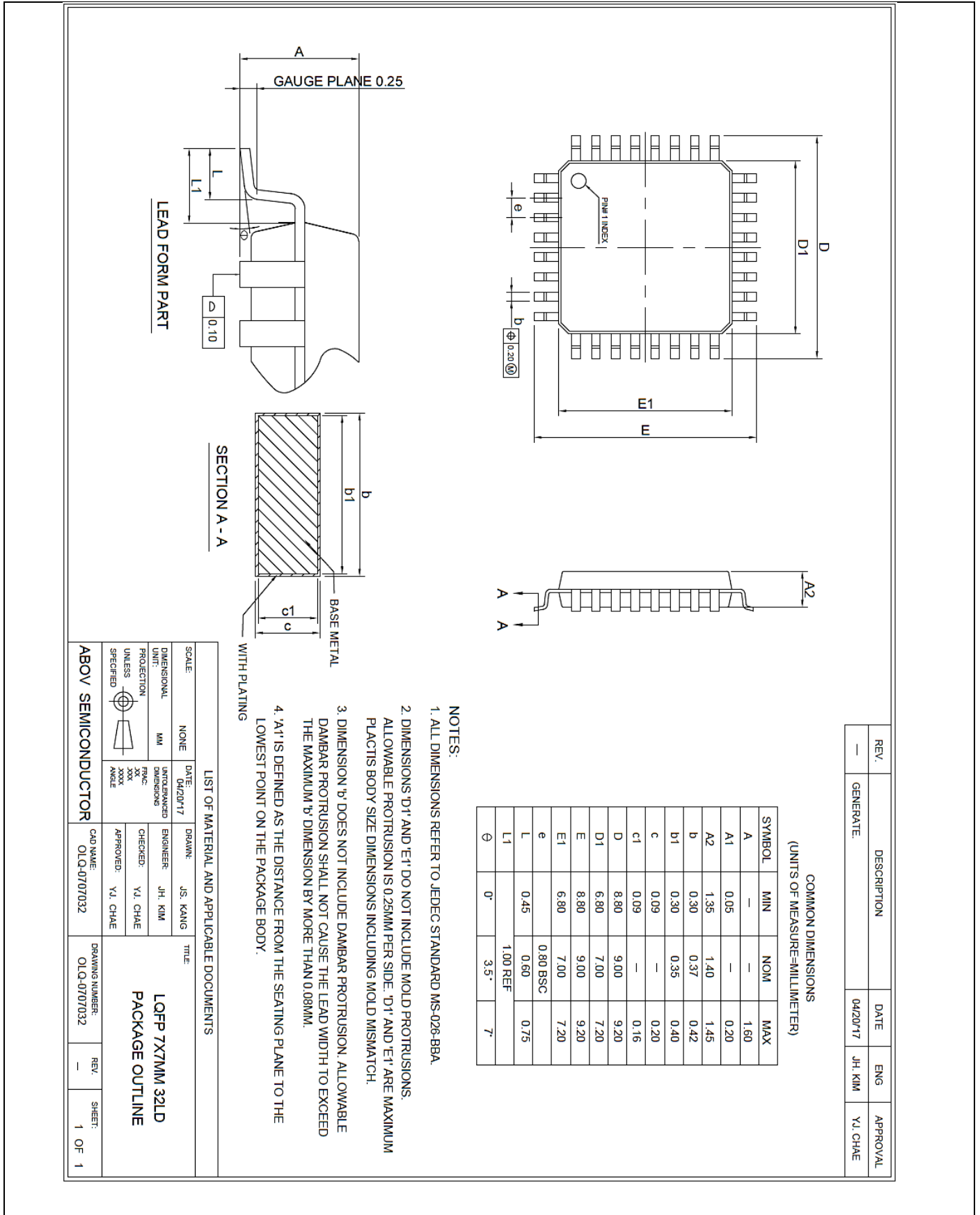


Figure 4.3 32-Pin LQFP Package

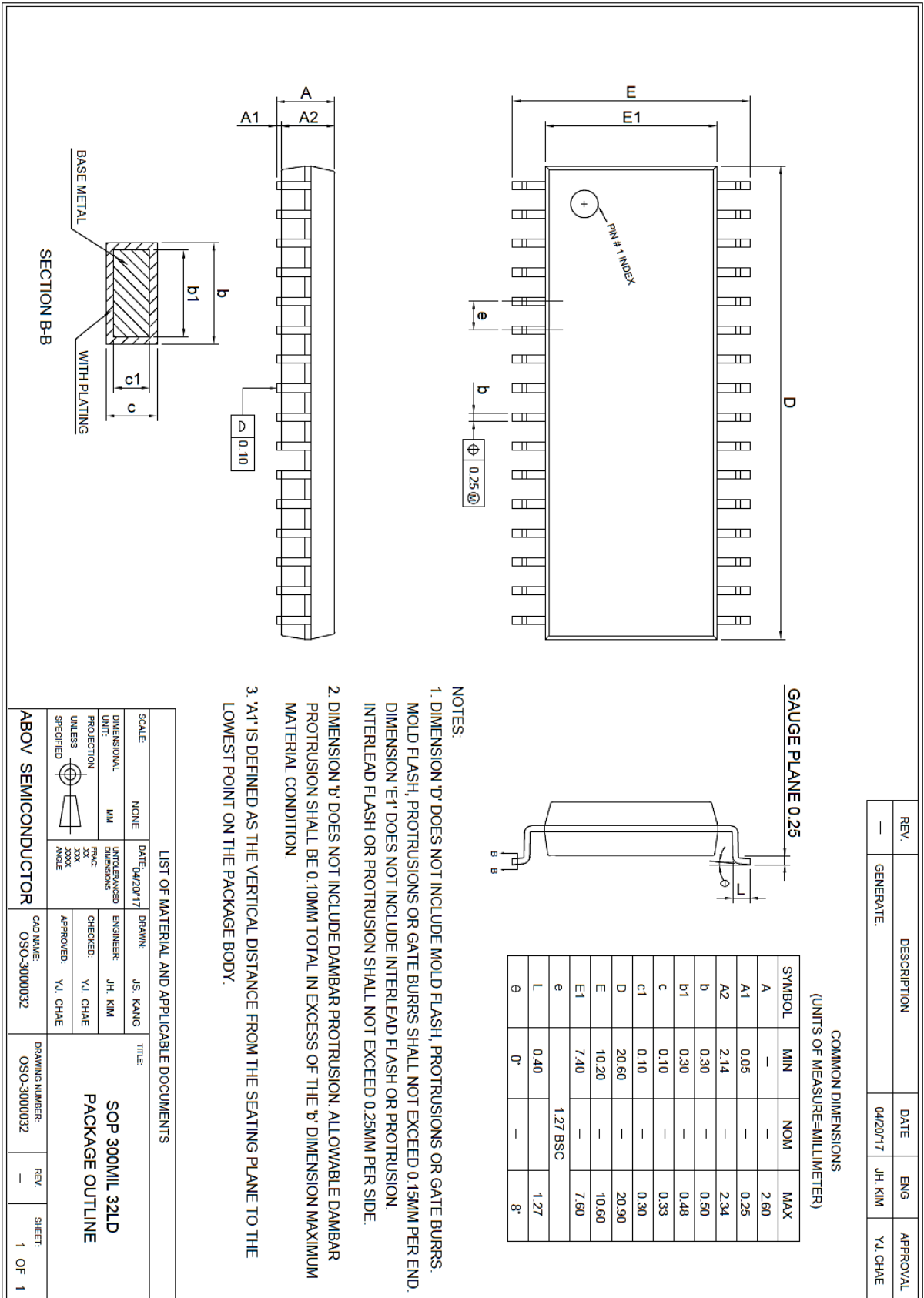


Figure 4.4 32-Pin SOP 封装

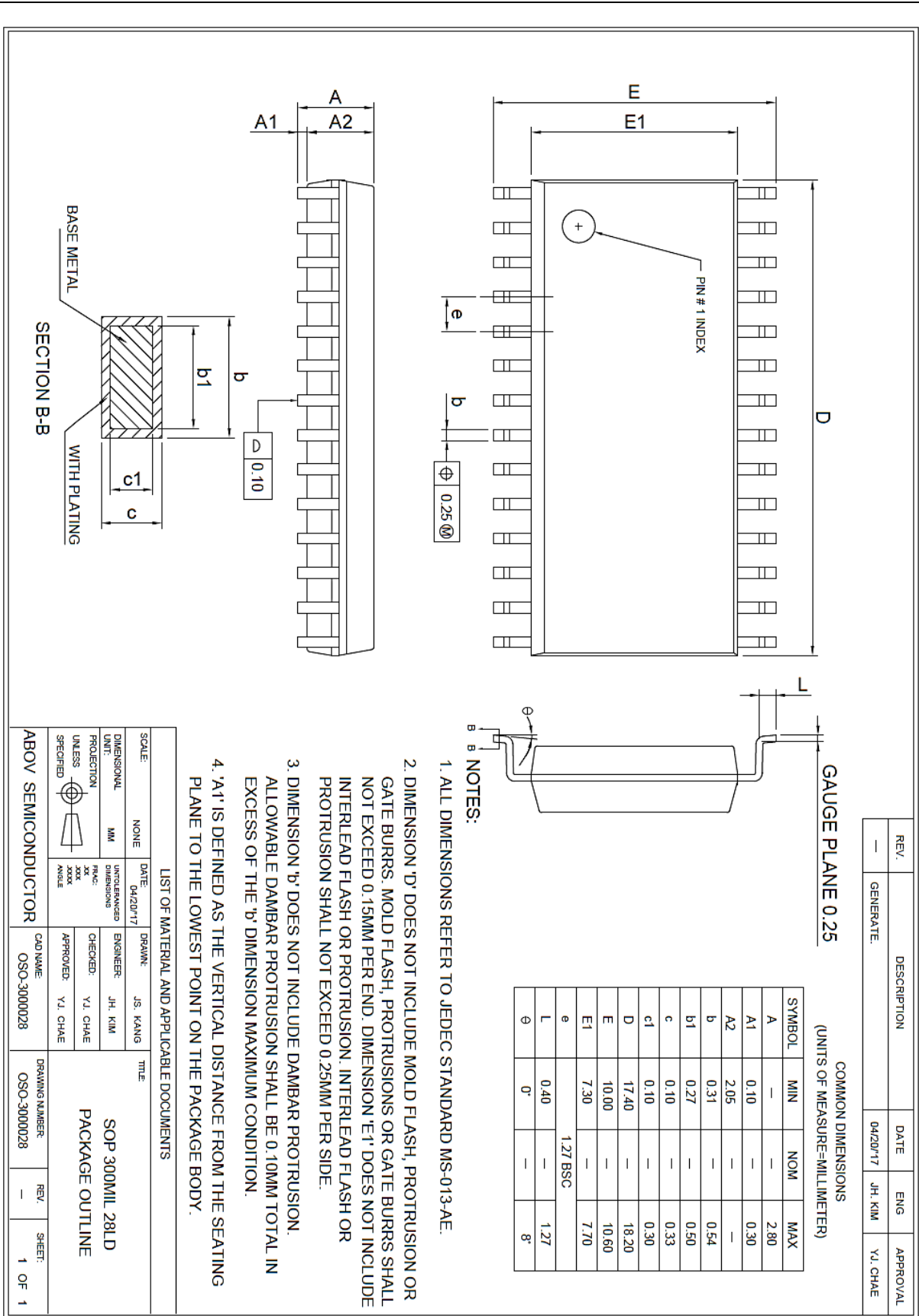


Figure 4.5 28-Pin SOP 封装

5 管脚说明

PIN Name	I/O	Function	@RESET	复用
P00	I/O	P0 端口可作为施密特触发输入，推-拉输出，或者漏极开路输出I/O口。 上拉电阻寄存器可指定到每一位。	Input	EC3/DSDA
P01				T30/DSCL
P02				AN0/AVREF/EINT0/T4O/PWM4AA
P03				SEG26/AN1/EINT1/PWM4AB
P04				SEG25/AN2/EINT2/PWM4BA
P05				SEG24/AN3/EINT3/PWM4BB
P06				SEG23/AN4/EINT4/PWM4CA
P07				SEG22/AN5/EINT5/PWM4CB
P10	I/O	P1 端口可作为施密特触发输入，推-拉输出，或者漏极开路输出I/O口。 上拉电阻寄存器可指定到每一位。 P14 – P17 在32脚封装中不存在。 P13 – P17 在28脚封装中不存在。	Input	SEG14/AN13/RXD1/SCL1/MISO1
P11				SEG15/AN12/EINT12/T2O/PWM2O
P12				SEG16/AN11/EINT11/T1O/PWM1O
P13				SEG17/AN10/EC1/BUZO
P14				SEG18/AN9/MOSI2
P15				SEG19/AN8/MISO2
P16				SEG20/AN7/EINT7/SCK2
P17				SEG21/AN6/EINT6/SS2
P20	I/O	P2 端口可作为施密特触发输入，推-拉输出，或者漏极开路输出I/O口。 上拉电阻寄存器可指定到每一位。 P23 – P25 在32脚封装中不存在。 P22 – P27 在28脚封装中不存在。	Input	SEG13/AN14/TXD1/SDA1/MOSI1
P21				SEG12/AN15/SCK1
P22				SEG11/SS1
P23				SEG10
P24				SEG9
P25				SEG8
P26				SEG7
P27				SEG6
P30	I/O	P3 端口可作为施密特触发输入，推-拉输出，或者漏极开路输出I/O口。 上拉电阻寄存器可指定到每一位。 P34 – P37 只存在于 44-pin 封装中。	Input	COM7/SEG5
P31				COM6/SEG4
P32				COM5/SEG3
P33				COM4/SEG2
P34				COM3/SEG1
P35				COM2/SEG0
P36				COM1
P37				COM0
P40	I/O	P4 端口可作为施密特触发输入，推-拉输出，或者漏极开路输出I/O口。 上拉电阻寄存器可指定到每一位。 P43 只存在于 44-pin 封装。	Input	VLC3/RXD0/SCL0/MISO0
P41				VLC2/TXD0/SDA0/MOSI0
P42				VLC1/SCK0
P43				VLC0/SS0

Table 5.1 普通口说明

PIN Name	I/O	Function	@RESET	复用
P50	I/O	P5 端口可作为施密特触发输入，推-拉输出，或者漏极开路输出I/O口。上拉电阻寄存器可指定到每一位。	Input	XOUT
P51				XIN
P52				EINT8/EC0/BLNK
P53				SXIN/T00/PWM00
P54				SXOUT/EINT10
P55				RESETB
EINT0	I/O	外部中断输入和定时器3捕捉输入	Input	P02/AN0/AVREF/T40/PWM4AA
EINT1	I/O	外部中断输入和定时器4捕捉输入	Input	P03/SEG26/AN1/PWM4AB
EINT2	I/O	外部中断输入	Input	P04/SEG25/AN2/PWM4BA
EINT3				P05/SEG24/AN3/PWM4BB
EINT4				P06/SEG23/AN4/PWM4CA
EINT5				P07/SEG22/AN5/PWM4CB
EINT6				P17/SEG21/AN6/SS2
EINT7				P16/SEG20/AN7/SCK2
EINT8				P52/EC0/BLNK
EINT10	I/O	外部中断输入和定时器0捕捉输入	Input	P54/SXOUT
EINT11	I/O	外部中断输入和定时器1捕捉输入	Input	P12/SEG16/AN11/T10/PWM10
EINT12	I/O	外部中断输入和定时器2捕捉输入	Input	P11/SEG15/AN12/T20/PWM20
T00	I/O	Timer 0间隔输出	Input	P53/SXIN/PWM00
T10	I/O	Timer 1间隔输出	Input	P12/SEG16/AN11/EINT11/PWM10
T20	I/O	Timer 2间隔输出	Input	P11/SEG15/AN12/EINT12/PWM20
T30	I/O	Timer 3间隔输出	Input	P01/DSCL
T40	I/O	Timer 4间隔输出	Input	P02/AN0/AVREF/EINT0/PWM4AA
PWM00	I/O	Timer 0 PWM 输出	Input	P53/SXIN/T00
PWM10	I/O	Timer 1 PWM 输出	Input	P12/SEG16/AN11/EINT11/T10
PWM20	I/O	Timer 2 PWM 输出	Input	P11/SEG15/AN12/EINT12/T20
PWM4AA	I/O	Timer 4 PWM 输出	Input	P02/AN0/AVREF/EINT0/T40
PWM4AB				P03/SEG26/AN1/EINT1
PWM4BA				P04/SEG25/AN2/EINT2
PWM4BB				P05/SEG24/AN3/EINT3
PWM4CA				P06/SEG23/AN4/EINT4
PWM4CB				P07/SEG22/AN5/EINT5
BLNK	I/O	6-ch PWM 外部同步信号输入	Input	P52/EINT8/EC0
EC0	I/O	Timer 0 EC 输入	Input	P52/EINT8/BLNK
EC1	I/O	Timer 1 EC 输入	Input	P13/SEG17/AN10
EC3	I/O	Timer 3 EC 输入	Input	P00/DSDA

Table 5.2 普通口说明

PIN Name	I/O	Function	@RESET	Shared with
BUZO	I/O	蜂鸣器信号输出	Input	P13/SEG17/AN10/EC1
SCK0	I/O	串行时钟0输入/输出	Input	P42/VLC1
SCK1	I/O	串行时钟1输入/输出	Input	P21/SEG12/AN15
SCK2	I/O	串行时钟2输入/输出	Input	P16/SEG20/AN7/EINT7
MOSI0	I/O	SPI 0 主控输出, 从动输入	Input	P41/VLC2/TXD0/SDA0
MOSI1	I/O	SPI 1 主控输出, 从动输入	Input	P20/SEG13/AN14/TXD1/SDA1
MOSI2	I/O	SPI 2 主控输出, 从动输入	Input	P14/SEG18/AN9
MISO0	I/O	SPI 0 主控输入, 从动输出	Input	P40/VLC3/RXD0/SCL0
MISO1	I/O	SPI 1 主控输入, 从动输出	Input	P10/SEG14/AN13/RXD1/SCL1
MISO2	I/O	SPI 2 主控输入, 从动输出	Input	P15/SEG19/AN8
SS0	I/O	SPI 0 从动选择输入	Input	P43/VLC0
SS1	I/O	SPI 1 从动选择输入	Input	P22/SEG11
SS2	I/O	SPI 2 从动选择输入	Input	P17/SEG21/AN6/EINT6
TXD0	I/O	UART 0 数据输出	Input	P41/VLC2/SDA0/MOSI0
TXD1	I/O	UART 1 数据输出	Input	P20/SEG13/AN14/SDA1/MOSI1
RXD0	I/O	UART 0 数据输入	Input	P40/VLC3/SCL0/MISO0
RXD1	I/O	UART 1 数据输入	Input	P10/SEG14/AN13/SCL1/MISO1
SCL0	I/O	I2C 0 时钟输入/输出	Input	P40/VLC3/RXD0/MISO0
SCL1	I/O	I2C 1 时钟输入/输出	Input	P10/SEG14/AN13/RXD1/MISO1
SDA0	I/O	I2C 0 数据输入/输出	Input	P41/VLC2/TXD0/MOSI0
SDA1	I/O	I2C 1 数据输入/输出	Input	P20/SEG13/AN14/TXD1/MOSI1
AVREF	I/O	A/D转换参考电压	Input	P02/AN0/EINT0/T4O/PWM4AA
AN0	I/O	A/D 转换器模拟信号输入通道	Input	P02/AVREF/EINT0/T4O/PWM4AA
AN1				P03/SEG26/EINT1/PWM4AB
AN2				P04/SEG25/EINT2/PWM4BA
AN3				P05/SEG24/EINT3/PWM4BB
AN4				P06/SEG23/EINT4/PWM4CA
AN5				P07/SEG22/EINT5/PWM4CB
AN6				P17/SEG21/EINT6/SS2
AN7				P16/SEG20/EINT7/SCK2
AN8				P15/SEG19/MISO2
AN9				P14/SEG18/MOSI2
AN10				P13/SEG17/EC1
AN11				P12/SEG16/EINT11/T10/PWM10
AN12				P11/SEG15/EINT12/T20/PWM20
AN13				P10/SEG14/RXD1/SCL1/MISO1
AN14				P20/SEG13/TXD1/SDA1/MOSI1
AN15				P21/SEG12/SCK1

Table 5.3 普通口说明

PIN Name	I/O	Function	@RESET	复用
VLC0	I/O	LCD 偏置电压引脚	Input	P43/SS0
VLC1				P42/SCK0
VLC2				P41/TXD0/SDA0/MOSI0
VLC3				P40/RXD0/SCL0/MISO0
COM0 COM1	I/O	LCD 共用信号输出	Input	P37-P36
COM2-COM3				P35-P34/SEG0-SEG1
COM4-COM7				P33-P30/SEG2-SEG5
SEG0-SEG1	I/O	LCD 段选信号输出	Input	P35-P34/COM2-COM3
SEG2-SEG5				P33-P30/COM4-COM7
SEG6-SEG10				P27-P23
SEG11				P22/SS1
SEG12				P21/SCK1/AN15
SEG13				P20/AN14/TXD1/SDA1/MOSI1
SEG14				P10/AN13/RXD1/SCL1/MISO1
SEG15				P11/AN12/EINT12/T2O/PWM2O
SEG16				P12/AN11/EINT11/T1O/PWM1O
SEG17				P13/AN10/EC1
SEG18				P14/AN9/MOSI2
SEG19				P15/AN8/MISO2
SEG20				P16/AN7/EINT7/SCK2
SEG21				P17/AN6/EINT6/SS2
SEG22				P07/AN5/EINT5/PWM4CB
SEG23				P06/AN4/EINT4/PWM4CA
SEG24				P05/AN3/EINT3/PWM4BB
SEG25				P04/AN2/EINT2/PWM4BA
SEG26				P03/AN1/EINT1/PWM4AB

Table 5.4 普通口说明

PIN Name	I/O	Function	@RESET	复用
RESETB	I/O	作为复位脚使用时需要通过CONFIGURE OPTION将系统复位脚设置上拉	Input	P55
DSDA	I/O	片内调试数据 输入/输出	Input	P00/EC3
DSCL	I/O	片内调试时钟 输入/输出	Input	P01/T3O
XIN	I/O	主振荡器引脚	Input	P51
XOUT				P50
SXIN	I/O	副振荡器引脚	Input	P53/T0O/PWM0O
SXOUT				P54/EINT10
VDD, VSS	-	电源	-	-

Table 5.5 普通口说明

注意)

1. P14–P17, P23–P25, P34–P37, 和 P43 在 32 脚封装中不存在.
2. P13–P17, P22–P27, P34–P37, 和 P43 在 28 脚封装中不存在.
3. P55/RESETB 通过“CONFIGURE OPTION”将其配置为 P55 和 RESETB 脚
4. P00/EC3/DSDA 和 P01/T3O/DSCL 在上电复位时连接了仿真器, 这个脚则自动的配置为仿真脚.
5. P00/EC3/DSDA 和 P01/T3O/DSCL 只有在复位或上电复位时被配置为内部上拉输入.
6. P50/XOUT, P51/XIN, P53/SXIN/T0O/PWM0O, 和 P54/SXOUT/EINT10 通过软件控制设置为复用功能.

6 端口结构

6.1 普通 I/O 口

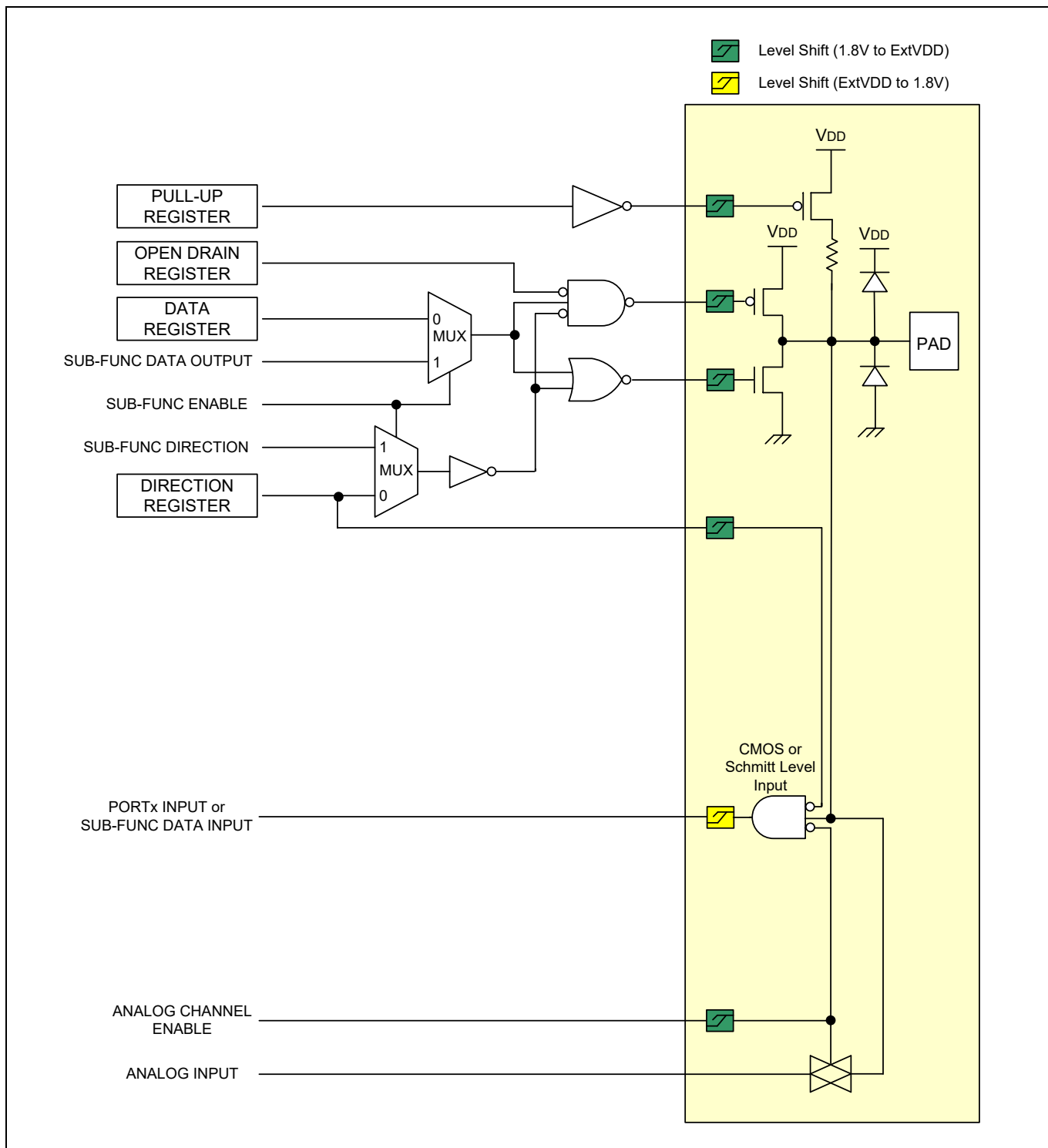


Figure 6.1 通用 I/O 口

6.2 外部中断 I/O 口

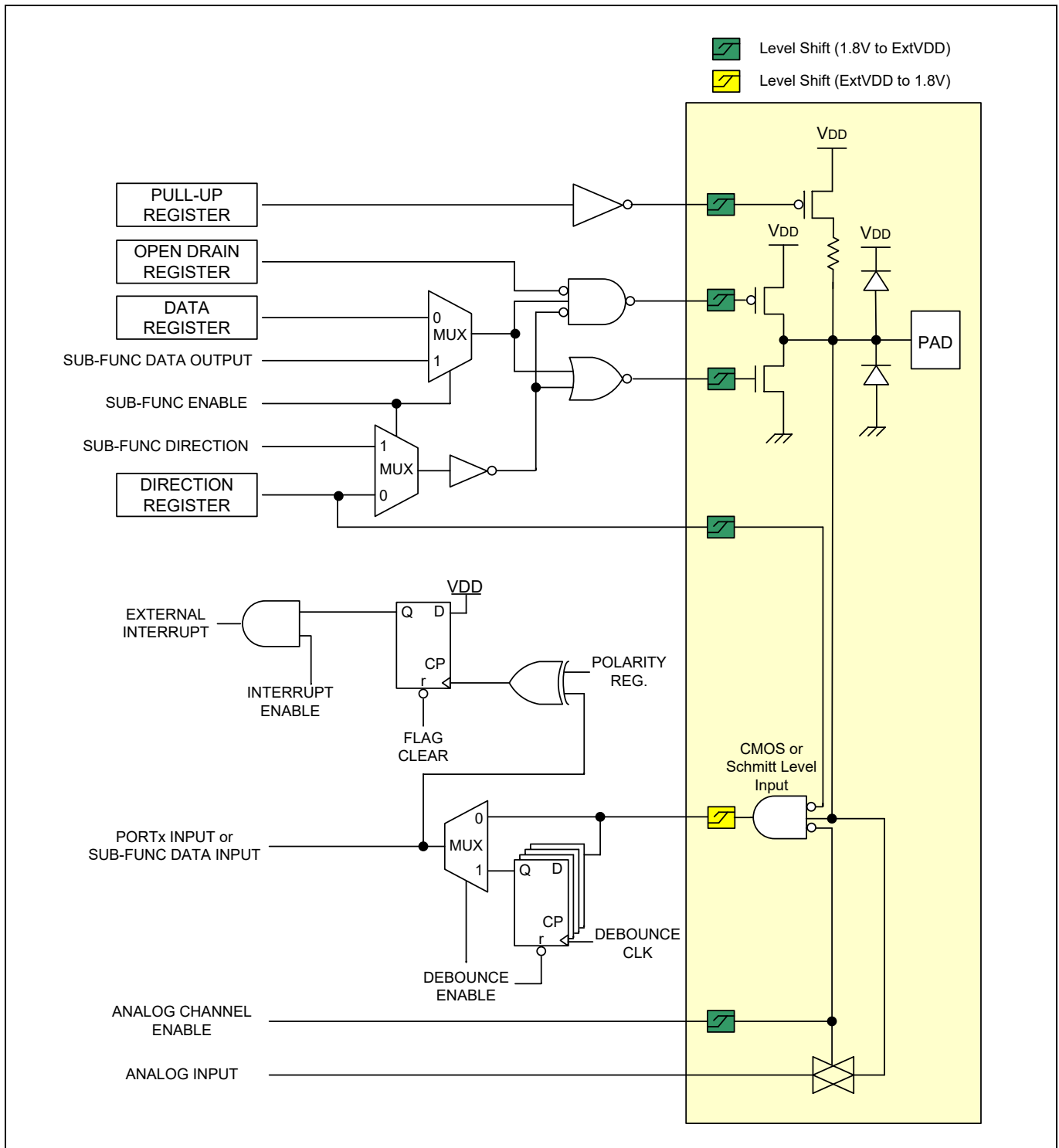


Figure 6.2 外部中断 I/O 口

7 电气特性

7.1 额定参数范围

Parameter	Symbol	Rating	Unit	Note
电源电压	VDD	-0.3~+6.5	V	-
正常电压	V _I	-0.3~VDD+0.3	V	VDD到VSS电压
	V _O	-0.3~VDD+0.3	V	
	I _{OH}	10	mA	最大输出电流 (I _{OH} per I/O pin)
	∑I _{OH}	80	mA	最大电流 (∑I _{OH})
	I _{OL}	60	mA	最大灌电流 (I _{OL} per I/O pin)
	∑I _{OL}	120	mA	最大电流 (∑I _{OL})
总功耗	P _T	600	mW	-
储存温度	T _{STG}	-65~+150	°C	-

Table 7.1 极限参数

NOTE)

1. 芯片使用时超过极限参数可能会对芯片造成永久性损坏. 这是一个额定值, 在任何情况下如果对芯片的有效操作参数超过规格书规定值都是不允许的. 长时间超过极限参数工作会影响器件的可靠性.

7.2 推荐操作参数

(TA=-40°C ~ +85°C)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit	
工作电压	VDD	f _x = 32 ~ 38kHz	Sub Crystal	2.2	-	5.5	V
		f _x = 0.4 ~ 4.2MHz	Main Crystal	2.2	-	5.5	
		f _x = 0.4 ~8.5MHz		2.7	-	5.5	
		f _x = 0.5 ~ 16MHz	Internal RC	2.2	-	5.5	
工作温度	T _{OPR}	VDD=2.2~5.5V	-40	-	85	°C	

Table 7.2 推荐工作参数

7.3 A/D 转换特性

(T_A=-40°C ~ +85°C, VDD=2.2V ~ 5.5V, VSS=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
分辨率	-	-	-	12	-	bit
积分非线性误差	INL	AVREF= 2.7V – 5.5V fx= 8MHz	-	-	±6	LSB
微分非线性误差	DNL		-	-	±1	
零点偏移误差	TOE		-	-	±5	
满量程误差	ZOE		-	-	±5	
转换时间	t _{CONV}	12-bit resolution, 8MHz	20	-	-	us
模拟输入电压	V _{AIN}	-	VSS	-	AVREF	V
模拟参考电压	AVREF	*Note 3	2.2	-	VDD	
模拟输入漏电流	I _{AIN}	AVREF=5.12V	-	-	2	uA
ADC 工作电流	I _{ADC}	Enable	-	1	2	mA
		Disable	-	-	0.1	uA

Table 7.3 A/D 转换特性

注意)

1. 零点偏移误差是指 000000000000 和转换输出的零点输入电压(VSS).
2. 满量程误差是指 111111111111 和转换输出的满量程输入电压(AVREF).
3. 当 AVREF 电压低于 2.7V,ADC 分辨率很差.

7.4 上电复位特性

(T_A=-40°C ~ +85°C, VDD=2.2V ~ 5.5V, VSS=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESET 电平	V _{POR}	-	-	1.4	-	V
VDD 电压上升时间	t _R	-	0.05	-	30.0	V/ms
POR 电流	I _{POR}	-	-	0.2	-	uA

Table 7.4 上电复位特性

7.5 低电压复位和低电压检测特性

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit	
选择电平	V _{LVR} V _{LVI}	LVR 可以选择所有电平但是LVI 只能选择除了1.85V以外的其他电平.	-	1.85	2.15	V	
			2.05	2.20	2.35		
			2.17	2.32	2.47		
			2.29	2.44	2.59		
			2.39	2.59	2.79		
			2.55	2.75	2.95		
			2.73	2.93	3.13		
			2.94	3.14	3.34		
			3.18	3.38	3.58		
			3.37	3.67	3.97		
			3.70	4.00	4.30		
4.10	4.40	4.70					
滞后	ΔV	-	-	50	150	mV	
最小脉冲宽度	t _{LOW}	-	100	-	-	us	
LVR 和 LVI 电流	I _{BL}	Enable (Both)	VDD= 3V, RUN Mode	-	14.0	24.0	uA
		Enable (One of two)		-	10.0	18.0	
		Disable (Both)	VDD= 3V	-	-	0.1	

Table 7.5 LVR 和 LVI 特性

7.6 内部高速 RC 振荡器特性

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

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
频率	f _{IRC}	VDD = 2.2 – 5.5V	-	16	-	MHz
误差	-	TA = 0°C to +50°C	-	-	±1.5	%
		TA = -20°C to +85°C			±2.5	
		TA = -40°C to +85°C			±3.5	
时钟占空比	TOD	-	40	50	60	%
稳定时间	T _{HFS}	-	-	-	100	us
IRC 电流	I _{IRC}	Enable	-	0.2	-	mA
		Disable	-	-	0.1	uA

Table 7.6 内部高速 RC 振荡器特性

注意)

1. VDD 和 VSS 之间需要连接一个 0.1uF 旁路电容.

7.7 内部看门狗定时器 RC 振荡器特性

(T_A=-40°C ~ +85°C, VDD=2.2V ~ 5.5V, VSS=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
频率	f _{WDTRC}	–	2	5	10	kHz
稳定时间	t _{WDTS}	–	–	–	1	ms
WDTRC 电流	I _{WDTRC}	Enable	–	1	–	uA
		Disable	–	–	0.1	

Table 7.7 内部 WDTRC 振荡器特性

7.8 LCD 电压特性

(T_A=-40°C ~ +85°C, VDD=2.2V ~ 5.5V, VSS=0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit	
LCD 电压	V _{LC0}	LCD contrast disabled, 1/4 bias	Typx0.95	VDD	Typx1.05	V	
		LCD contrast enabled, 1/4 bias, RLCD1, No panel load	LCDCR=00H	Typx0.9	VDDx16/31	Typx1.1	V
			LCDCR=01H		VDDx16/30		
			LCDCR=02H		VDDx16/29		
			LCDCR=03H		VDDx16/28		
			LCDCR=04H		VDDx16/27		
			LCDCR=05H		VDDx16/26		
			LCDCR=06H		VDDx16/25		
			LCDCR=07H		VDDx16/24		
			LCDCR=08H		VDDx16/23		
			LCDCR=09H		VDDx16/22		
			LCDCR=0AH		VDDx16/21		
			LCDCR=0BH		VDDx16/20		
			LCDCR=0CH		VDDx16/19		
			LCDCR=0DH		VDDx16/18		
LCDCR=0EH	VDDx16/17						
LCDCR=0FH	VDDx16/16						
LCD Mid Bias Voltage(note)	V _{LC1}	VDD=2.7V to 5.5V, LCD clock = 0Hz, 1/4 bias, No panel load	Typx0.9	3/4xVLC0	Typx1.1	V	
	V _{LC2}		Typx0.9	2/4xVLC0	Typx1.1		
	V _{LC3}		Typx0.9	1/4xVLC0	Typx1.1		
LCD Driver Output Impedance	R _{LO}	VLCD=3V, ILOAD=±10uA	–	5	10	kΩ	
LCD Bias Dividing Resistor	R _{LCD1}	T _A = 25°C	40	60	80		
	R _{LCD2}		80	120	160		

Table 7.8 LCD 电压特性

注意)

1. 当 VDD 和 V_{LC0} 节点连接时, 输出为中间电压.

7.9 DC 特性

(T_A = -40°C ~ +85°C, VDD = 2.2V ~ 5.5V, VSS = 0V, f_{XIN} = 8.0MHz)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit	
Input High Voltage	V _{IH1}	P0, P1, P5, RESETB	0.8VDD	–	VDD	V	
	V _{IH2}	All input pins except V _{IH1}	0.7VDD	–	VDD	V	
Input Low Voltage	V _{IL1}	P0, P1, P5, RESETB	–	–	0.2VDD	V	
	V _{IL2}	All input pins except V _{IL1}	–	–	0.3VDD	V	
Output High Voltage	V _{OH}	VDD=4.5V, I _{OH} =-2mA, All output ports;	VDD-1.0	–	–	V	
Output Low Voltage	V _{OL1}	VDD=4.5V, I _{OL} = 10mA; All output ports except V _{OL2}	–	–	1.0		
	V _{OL2}	VDD=4.5V, I _{OL} = 15mA; P1	–	–	1.0	V	
Input High Leakage Current	I _{IH}	All input ports	–	–	1	uA	
Input Low Leakage Current	I _{IL}	All input ports	-1	–	–	uA	
Pull-Up Resistor	R _{PU1}	VI=0V, T _A = 25°C All Input ports	VDD=5.0V	25	50	100	kΩ
			VDD=3.0V	50	100	200	
	R _{PU2}	VI=0V, T _A = 25°C RESETB	VDD=5.0V	150	250	400	kΩ
			VDD=3.0V	300	500	700	
OSC feedback resistor	R _{X1}	XIN= VDD, XOUT= VSS T _A = 25°C, VDD= 5V	600	1200	2000	kΩ	
	R _{X2}	SXIN=VDD, SXOUT=VSS T _A = 25°C, VDD=5V	2500	5000	10000		
Supply Current	I _{DD1} (RUN)	f _{XIN} =8MHz	VDD= 5V±10%	–	2.6	5.2	mA
		f _{XIN} =4MHz	VDD= 3V±10%	–	1.2	2.4	
		f _{IRC} =16MHz	VDD= 5V±10%	–	3.0	6.0	
	I _{DD2} (IDLE)	f _{XIN} =8MHz	VDD= 5V±10%	–	1.8	3.6	mA
		f _{XIN} =4MHz	VDD= 3V±10%	–	0.8	1.6	
		f _{IRC} =16MHz	VDD= 5V±10%	–	1.5	3.0	
	I _{DD3}	f _{XIN} =32.768kHz VDD= 3V±10% T _A = 25°C	Sub RUN	–	90.0	180.0	uA
	I _{DD4}		Sub IDLE	–	8.0	16.0	uA
I _{DD5}	STOP, VDD= 5V±10%, T _A = 25°C		–	0.5	3.0	uA	

Table 7.9 DC 特性

注意)

1. f_{XIN} 是外部主时钟, f_{SUB} 是外部副时钟, f_{IRC} 是内部 RC, f_X 是选择系统时钟..
2. 所有的电流项不包括内部 WDTRC 振荡器和外围设备.
3. 所有的电流项包括上电复位模块 (POR).

7.10 AC 特性

(TA= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESETB 输入低脉冲宽度	t_{RSL}	Input, VDD= 5V	10	-	-	us
中断输入高低脉冲宽度	t_{iWH} , t_{iWL}	All interrupt, VDD= 5V	200	-	-	ns
外部计数器输入高低脉冲宽度	t_{ECWH} , t_{ECWL}	ECn, VDD = 5V (n= 0, 1, 3)	200	-	-	
外部计数器转换时间	t_{REC} , t_{FEC}	ECn, VDD = 5V (n= 0, 1, 3)	20	-	-	

Table 7.10 AC 特性

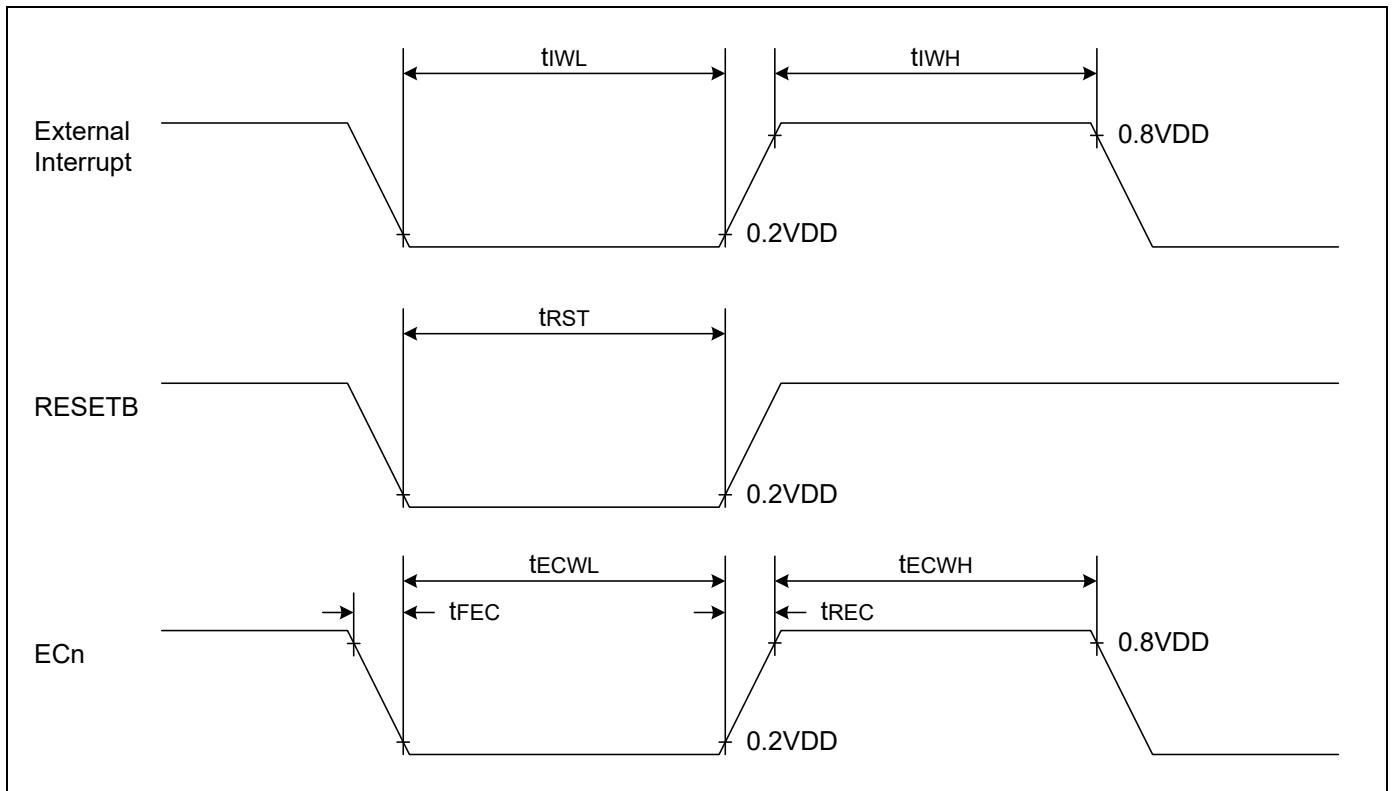


Figure 7.1 AC 时序

7.11 SPI0/1/2 特性

($T_A = -40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD} = 2.2\text{V} \sim 5.5\text{V}$)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
输出时钟脉冲周期	tSCK	Internal SCK source	200	-	-	ns
输入时钟脉冲周期		External SCK source	200	-	-	
输出时钟高低脉冲宽度	tSCKH,	Internal SCK source	70	-	-	
输入时钟高低脉冲宽度	tSCKL	External SCK source	70	-	-	
首个输出时钟延迟时间	tFOD	Internal/External SCK source	100	-	-	
输出时钟延迟时间	tDS	-	-	-	50	
输入设置时间	tDIS	-	100	-	-	
输入维持时间	tDIH	-	150	-	-	

Table 7.11 SPI0/1/2 特性

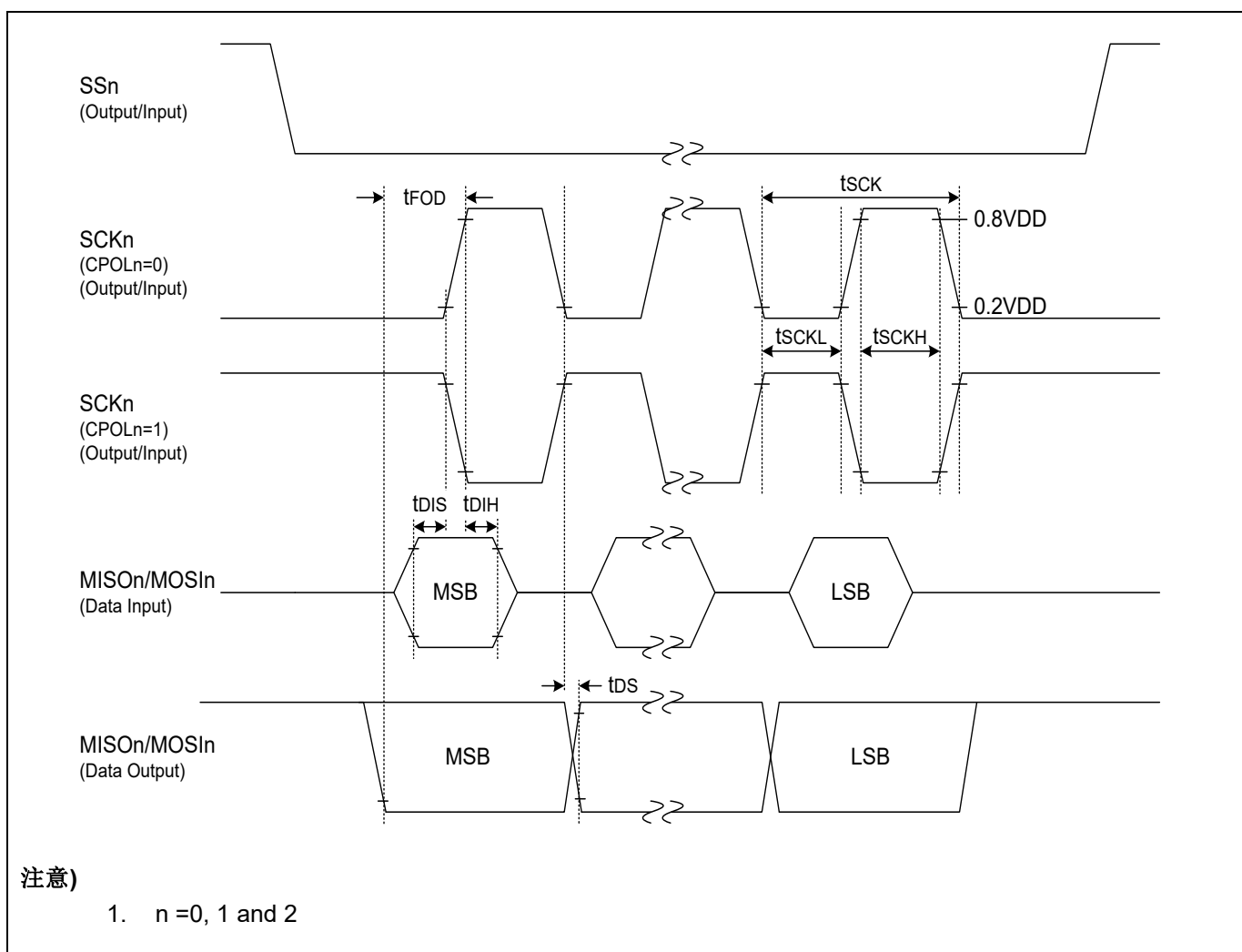


Figure 7.2 SPI0/1/2 时序

7.12 UART0/1 特性

(TA=-40°C ~ +85°C, VDD=1.8V ~ 5.5V, fXIN=11.1MHz)

Parameter	Symbol	MIN	TYP	MAX	Unit
串口时钟周期	t_{SCK}	1800	$t_{CPU} \times 16$	2200	ns
输出数据设置到上升沿时间	t_{S1}	810	$t_{CPU} \times 13$	-	
时钟上升沿到输入有效数据时间	t_{S2}	-	-	590	
时钟上升沿后输出数据维持时间	t_{H1}	$t_{CPU} - 50$	t_{CPU}	-	
时钟上升沿后输入数据维持时间	t_{H2}	0	-	-	
串口时钟高低点评宽度	t_{HIGH}, t_{LOW}	720	$t_{CPU} \times 8$	1280	

Table 7.12 UART0/1 特性

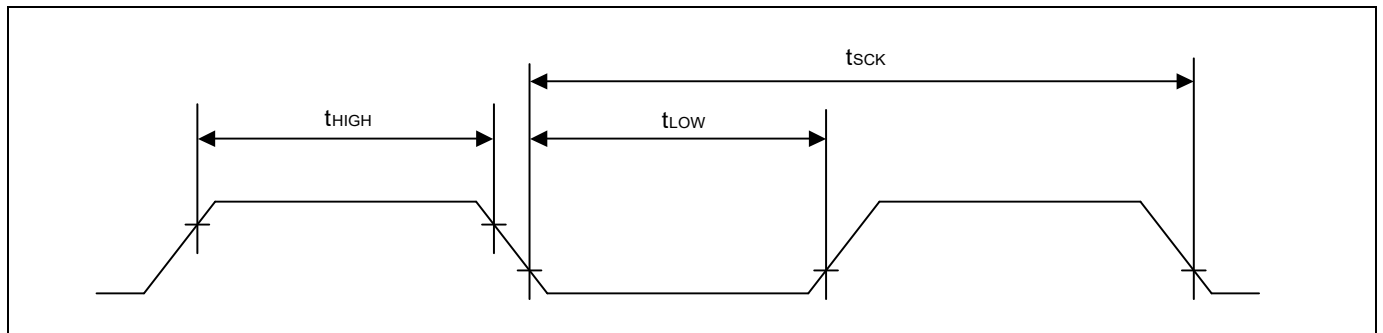


Figure 7.3 UART0/1 时序特性波形图

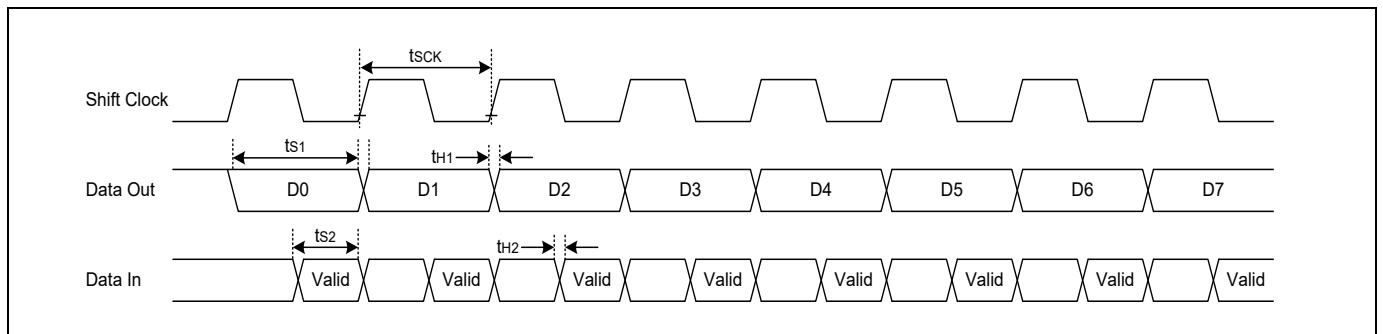


Figure 7.4 UART0/1 模块时序波形图

7.13 I2C0/1 Characteristics

(TA=-40°C ~ +85°C, VDD=2.2V ~ 5.5V)

Parameter	Symbol	Standard Mode		High-Speed Mode		Unit
		MIN	MAX	MIN	MAX	
时钟频率	tSCL	0	100	0	400	kHz us
时钟高脉冲宽度	tSCLH	4.0	-	0.6	-	
时钟低脉冲宽度	tSCLL	4.7	-	1.3	-	
总线空闲时间	tBF	4.7	-	1.3	-	
开始命令设置时间	tSTSU	4.7	-	0.6	-	
开始命令维持时间	tSTHD	4.0	-	0.6	-	
停止命令设置时间	tSPSU	4.0	-	0.6	-	
停止命令维持时间	tSPHD	4.0	-	0.6	-	
时钟有效输出	tVD	0	-	0	-	
数据输入维持时间	tDIH	0	-	0	1.0	
数据输入设置时间	tDIS	250	-	100	-	

Table 7.13 I2C0/1 特性

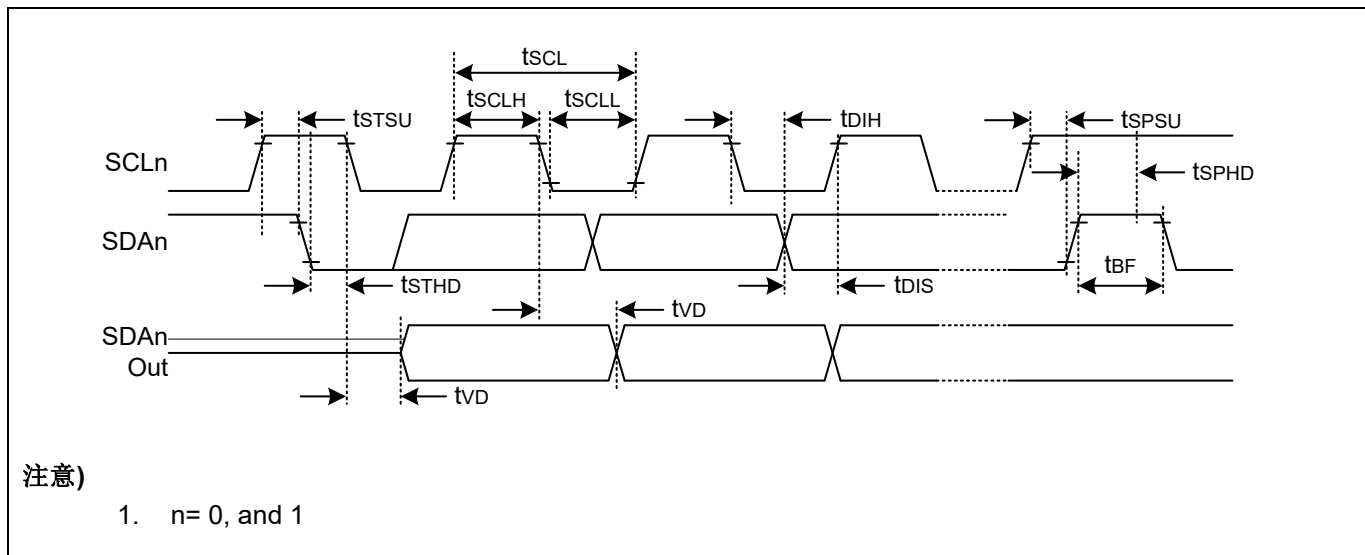


Figure 7.5 I2C0/1 时序

7.14 Stop 模式下保存数据所需电压

($T_A = -40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD} = 1.8\text{V} \sim 5.5\text{V}$)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
保存数据电压	V_{DDDR}	—	2.2	—	5.5	V
保持数据电流	I_{DDDR}	$V_{DDDR} = 2.2\text{V}$, ($T_A = 25^{\circ}\text{C}$), Stop mode	—	—	1	μA

Table 7.14 Stop 模式保存数据所需电压

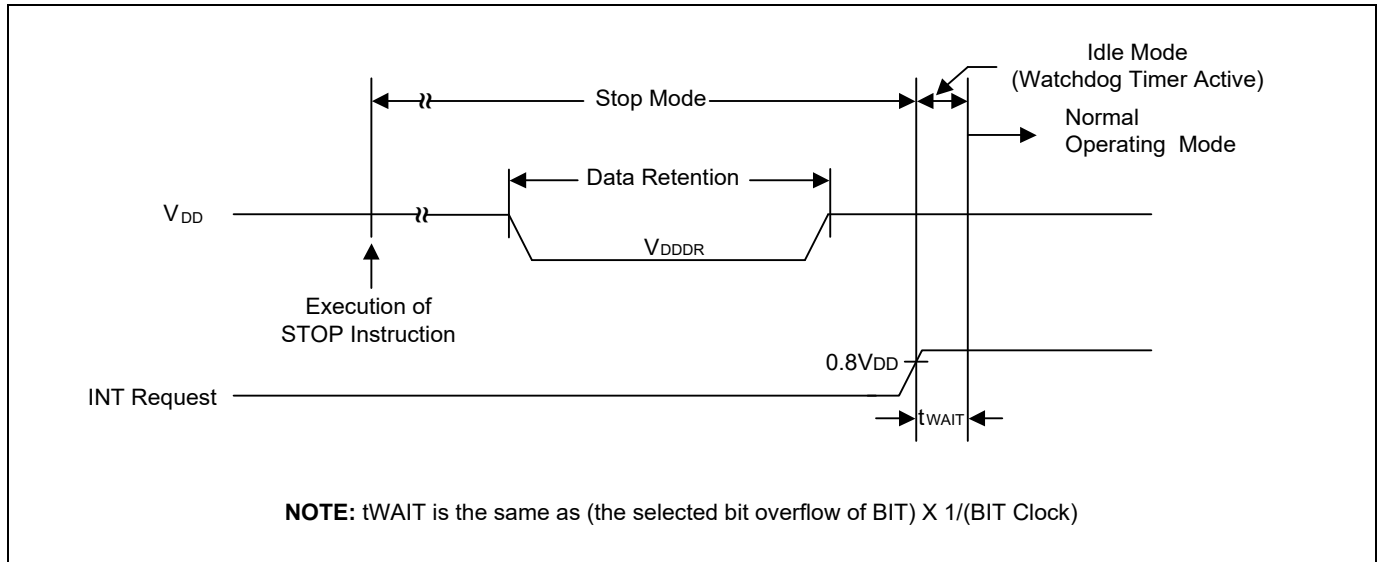


Figure 7.6 Stop 模式被中断唤醒时序

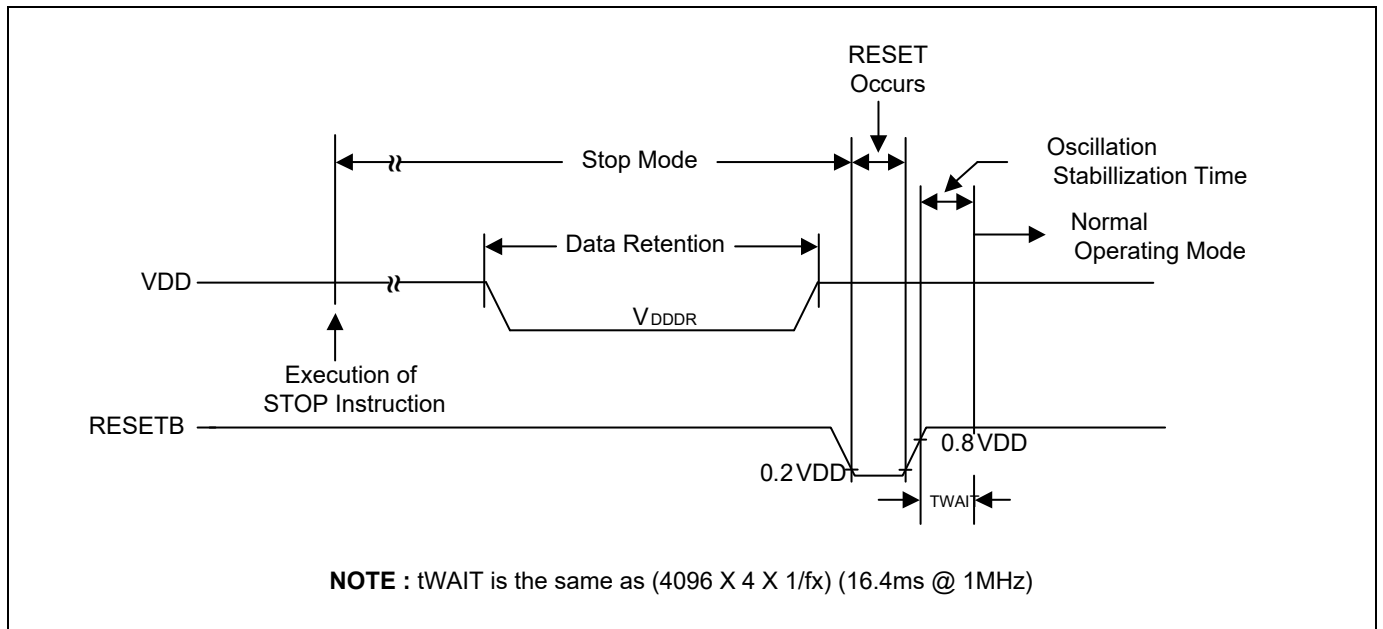


Figure 7.7 Stop 复位唤醒时序

7.15 内部 Flash Rom 特性

(T_A=-40°C ~ +85°C, VDD=1.8V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Sector Write Time	t _{FSW}	–	–	2.5	2.7	ms
Sector Erase Time	t _{FSE}	–	–	2.5	2.7	
Code Write Protection Time	t _{FHL}	–	–	2.5	2.7	
Page Buffer Reset Time	t _{FBR}	–	–	–	5	us
Flash Programming Frequency	f _{PGM}	–	0.4	–	–	MHz
Endurance of Write/Erase	NF _{WE}	Sector 0 to 507	–	–	10,000	times
		Sector 508 to 511(256 bytes)	–	–	100,000	
Flash Data Retention Time	t _{RT}	–	10	–	–	years

Table 7.15 内部 Flash Rom 特性

注意)

1. 在 flash 操作时, SCCR 的 SCLK[1:0] 位必须设置为“00” 或 “01” (INT-RC OSC 或 Main X-TAL).

7.16 输入/输出 特性

(T_A=-40°C ~ +85°C, VDD=0V)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Input 电容	C _{IN}	f _x = 1MHz Unmeasured pins are connected to VSS	–	–	10	pF
Output 电容	C _{OUT}					
I/O 电容	C _{IO}					

Table 7.16 Input/Output 特性

7.17 主时钟振荡特性

($T_A = -40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD} = 2.2\text{V} \sim 5.5\text{V}$)

Oscillator	Parameter	Condition	MIN	TYP	MAX	Unit
Crystal	Main oscillation frequency	2.2V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	8.5	
Ceramic Oscillator	Main oscillation frequency	2.2V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	8.5	
External Clock	XIN input frequency	2.2V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	8.5	

Table 7.17 主时钟振荡特性

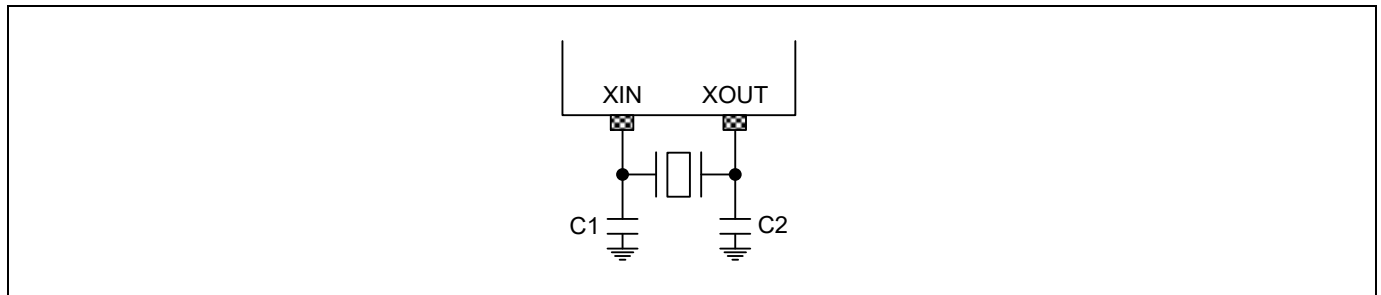


Figure 7.8 晶体/陶振

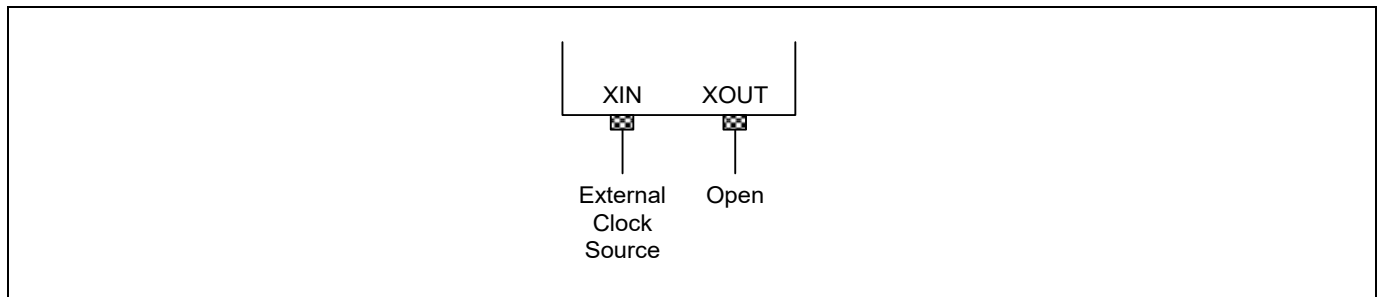


Figure 7.9 外部时钟

7.18 副时钟振荡特性

($T_A = -40^{\circ}\text{C} \sim +85^{\circ}\text{C}$, $V_{DD} = 2.2\text{V} \sim 5.5\text{V}$)

Oscillator	Parameter	Condition	MIN	TYP	MAX	Unit
晶体	Sub oscillation frequency	2.2V – 5.5V	32	32.768	38	kHz
外部时钟	SXIN input frequency		32	–	100	kHz

Table 7.18 副时钟振荡特性

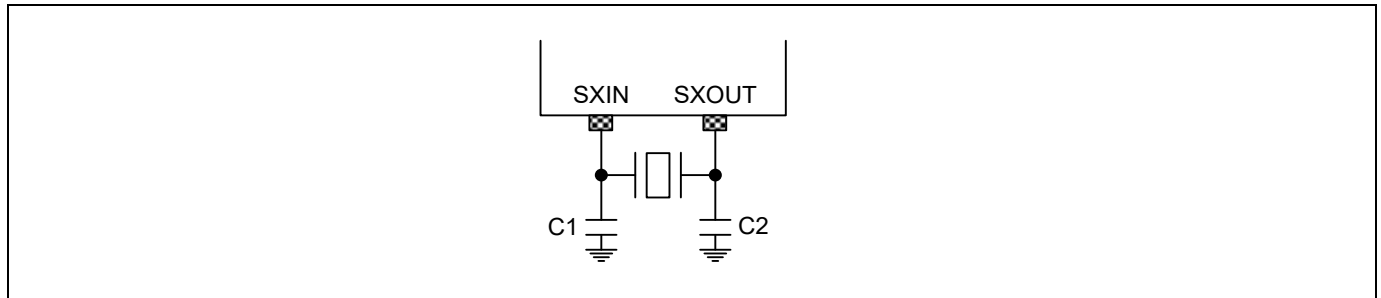


Figure 7.10 晶体振荡器

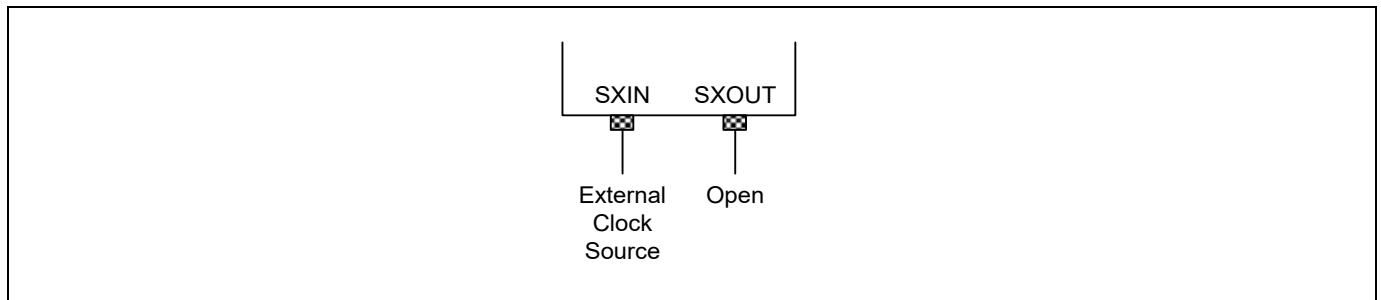


Figure 7.11 外部时钟

7.19 主时钟振荡稳定特性

(T_A=-40°C ~ +85°C, VDD=2.2V ~ 5.5V)

Oscillator	Parameter	MIN	TYP	MAX	Unit
Crystal	f _x > 4MHz, VDD = 2.7V ~ 5.5V,	-	-	15	ms
	f _x > 1MHz, VDD = 2.2V, T _A =-40°C			60	
Ceramic	-	-	-	10	ms
External Clock	f _{XIN} = 0.4 to 8.5MHz XIN input high and low width (t _{xH} , t _{xL})	58	-	1250	ns

Table 7.19 主时钟振荡稳定特性

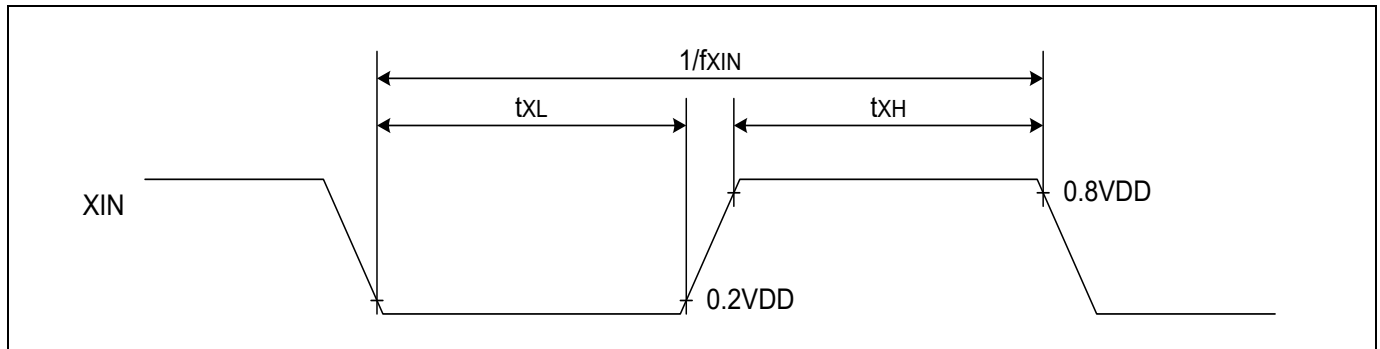


Figure 7.12 XIN 脚时钟测量时序

7.20 副时钟特性

(T_A=-40°C ~ +85°C, VDD=2.2V ~ 5.5V)

Oscillator	Parameter	MIN	TYP	MAX	Unit
Crystal	-	-	-	10	s
External Clock	SXIN input high and low width (t _{xH} , t _{xL})	5	-	15	us

Table 7.20 副时钟振荡稳定特性

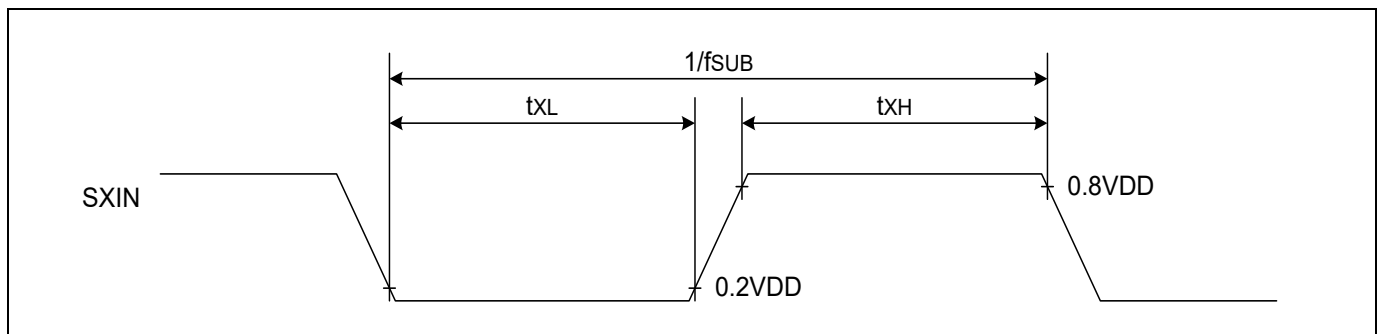


Figure 7.13 SXIN 脚时钟测量时序

7.21 工作电压范围

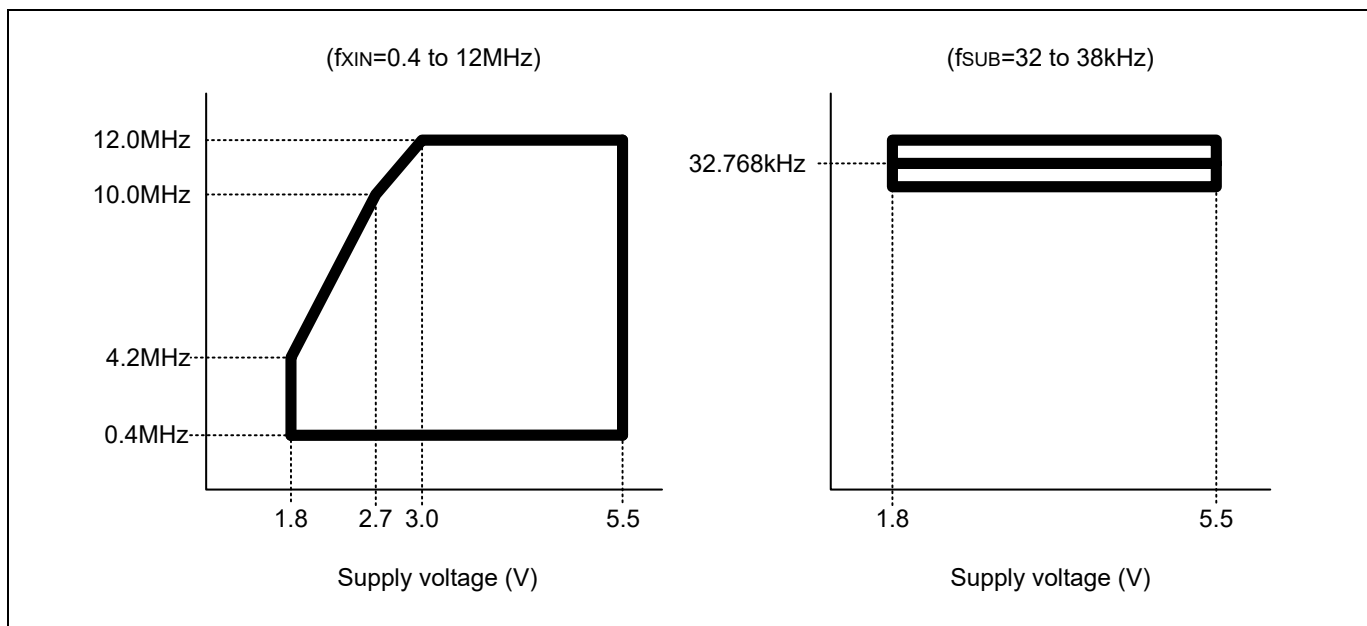


Figure 7.14 工作电压范围

7.22 推荐电路和布板

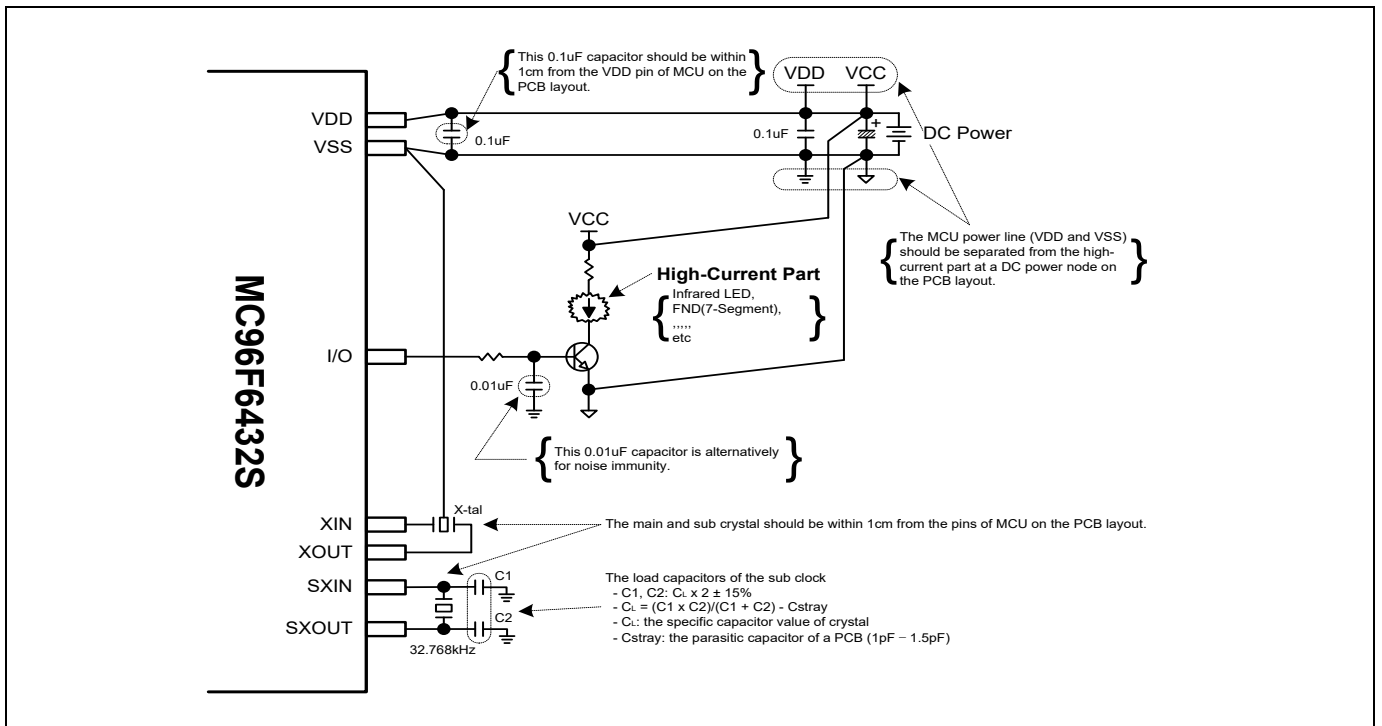


Figure 7.15 推荐电压范围

7.23 推荐开关电源的电路布局

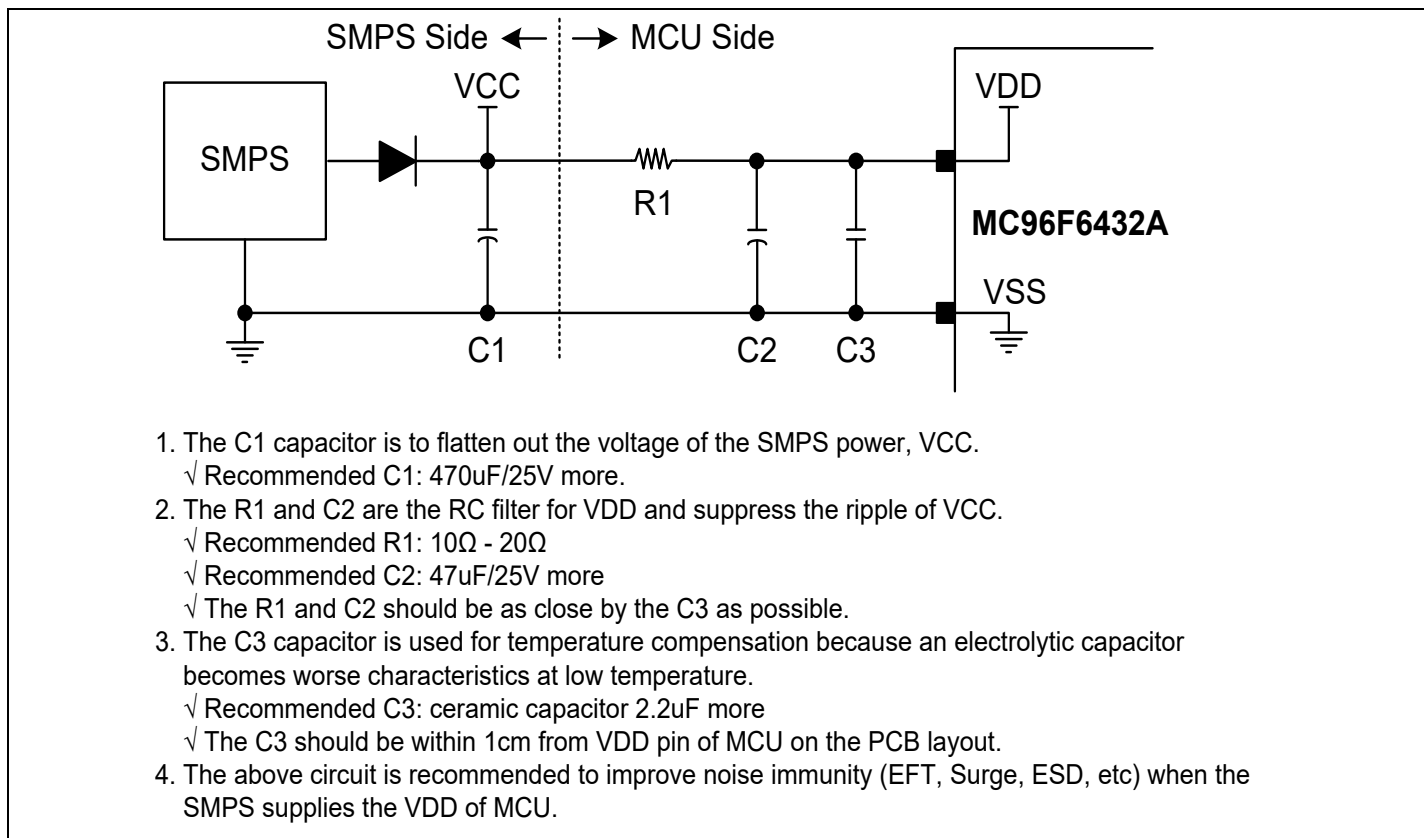


Figure 7.16 推荐开关电源的电路布局

7.24 典型特性

本节的这些图表只是设计指导并没有测试担保. 图表中的一些数据超出了指定的操作范围, 只是作为操作参数内的参考.

本节的这些数据只是数据单元采集的概要, 在最大 (+ 3 σ)和最小 (-3 σ)之间的一些典型参数的误差标准.

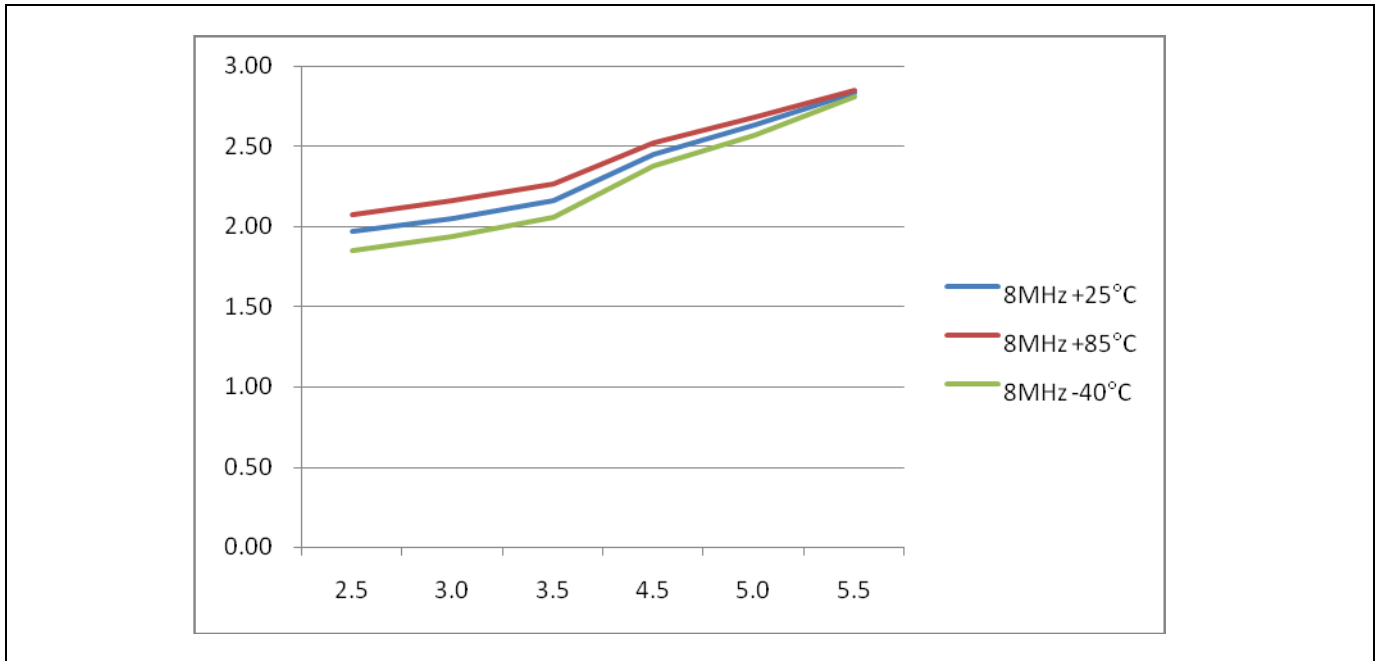


Figure 7.17 RUN (IDD1) 电流

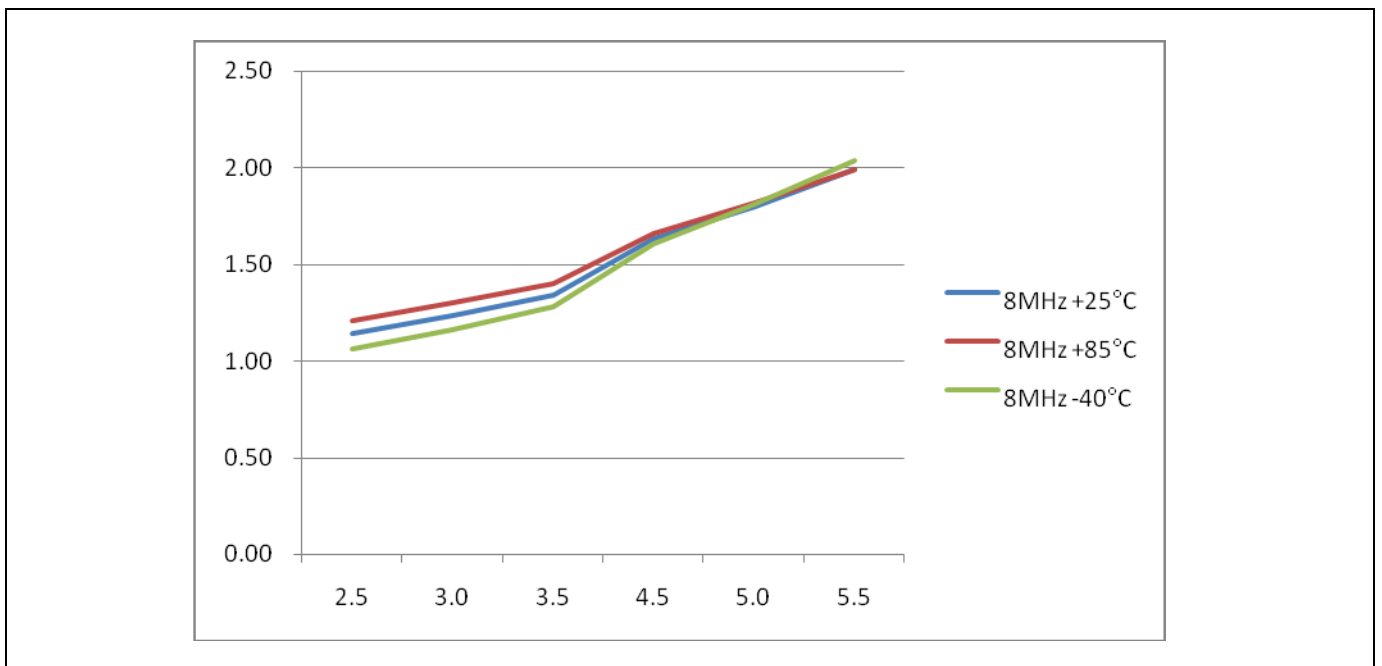


Figure 7.18 IDLE (IDD2) 电流

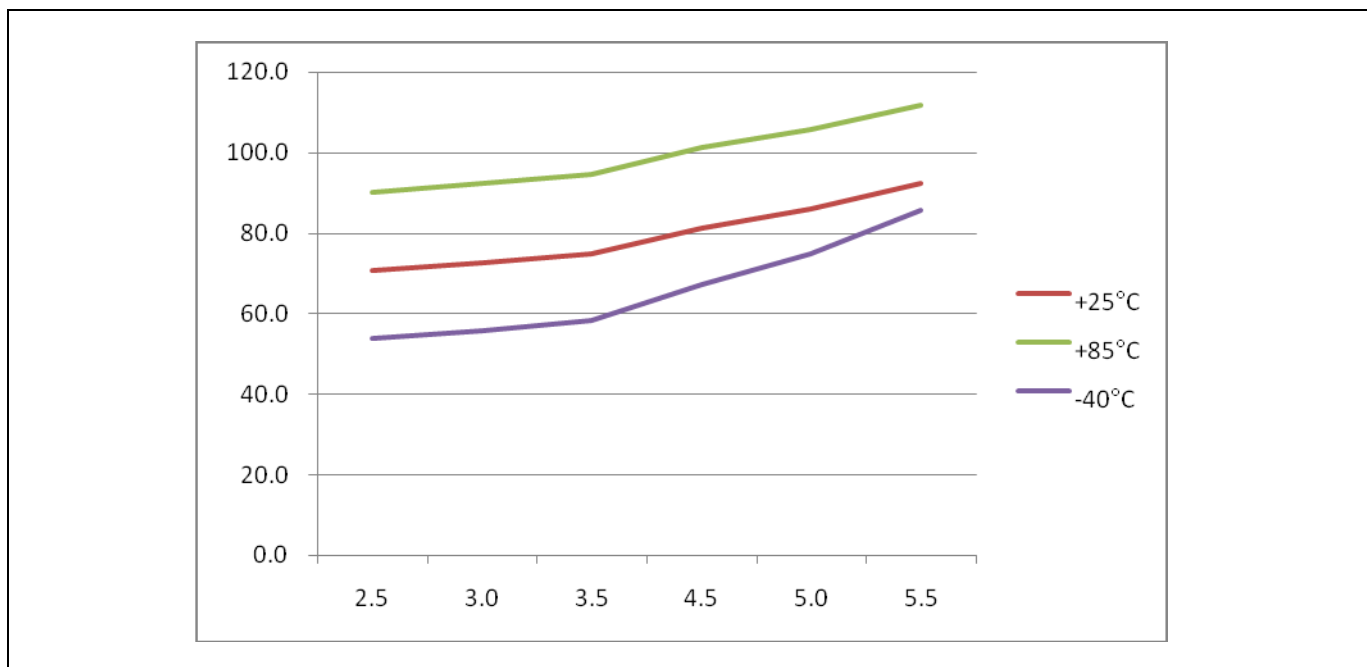


Figure 7.19 SUB RUN (IDD3) 电流

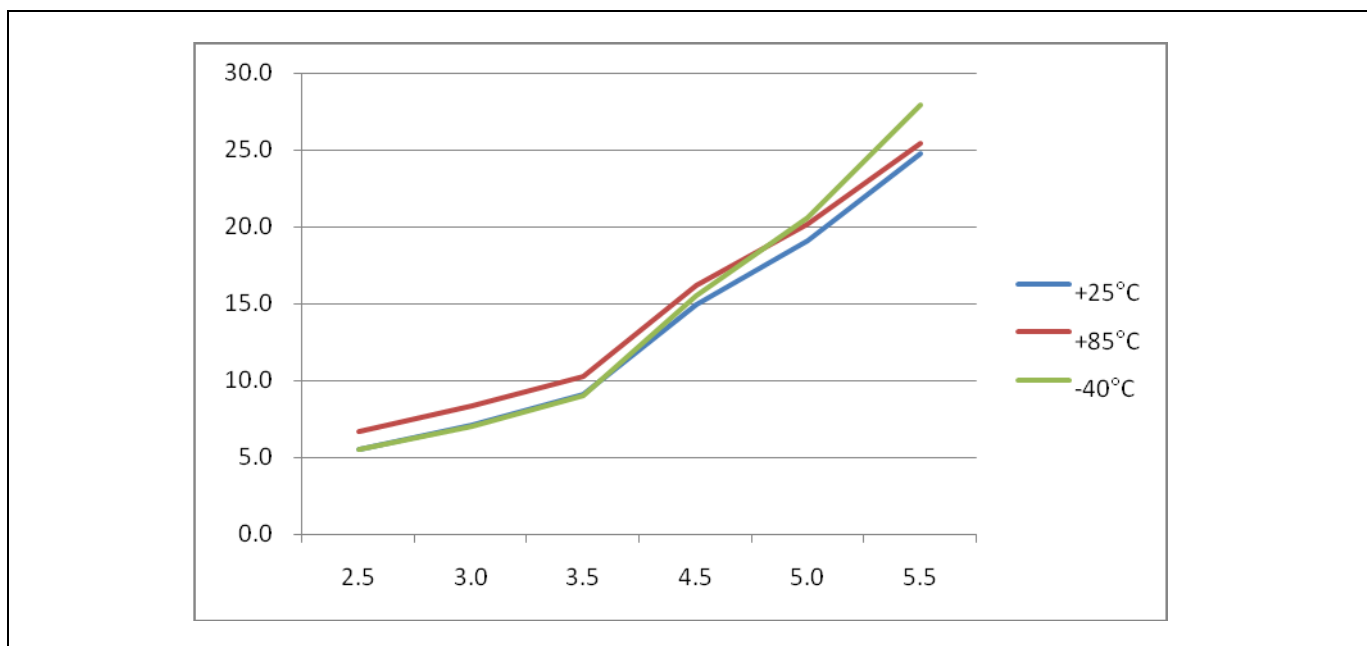


Figure 7.20 SUB IDLE (IDD4) 电流

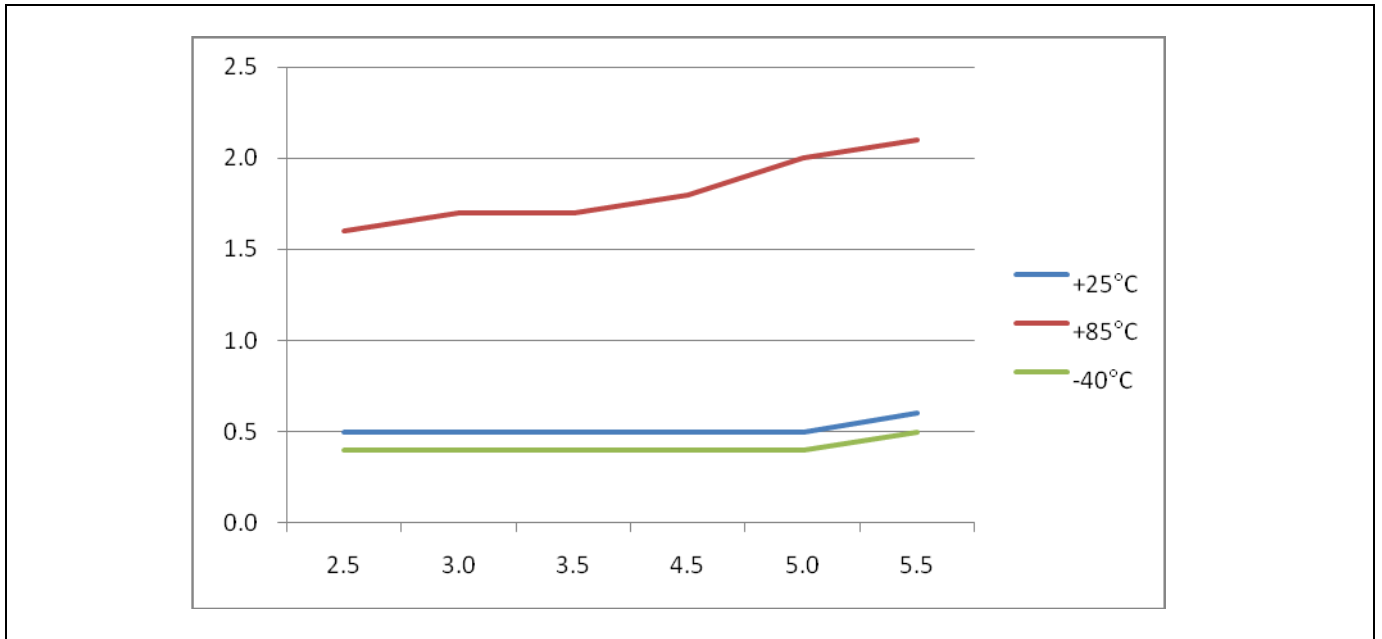


Figure 7.21 STOP (IDD5) 电流

8 存储器

MC96F6432A 具有两个独立的地址存储空间:程序存储器和数据存储器. 合理的分配程序存储器和允许8位存储的数据存储器,这样就使得8位CPU进入数据存储器更加迅速. 不过, 16位数据存储地址同样可以通过DPTR寄存器进行存储.

MC96F6432A 具备片内 32k 字节 ISP 类型的可读可写 flash 存储, IRAM 是 256 bytes 包括堆栈区. XRAM 是 768 字节包括27个字节LCD显示RAM.

8.1 程序存储器

一个 16-bit 程序计数器寻址最大可达64k bytes, 但是此芯片只有 32k bytes 程序存储空间。

图 8.1 所示为程序存储器较低区域. 复位后, CPU 从0000H开始执行. 每个中断在程序存储器中都分配有固定的地址.

中断使 CPU 跳到该地址开始执行服务程序.例如外部中断 11,被分配地址为 000BH. 如果使用外部中断11, 它的服务程序必须从 000BH位置开始, 如果没有用到该中断, 其服务地址作为普通程序存储地址使用. 如果中断小于8个字节, 不需要加跳转指令. 更长的程序需要使用跳转指令来跳过随后的中断位置.

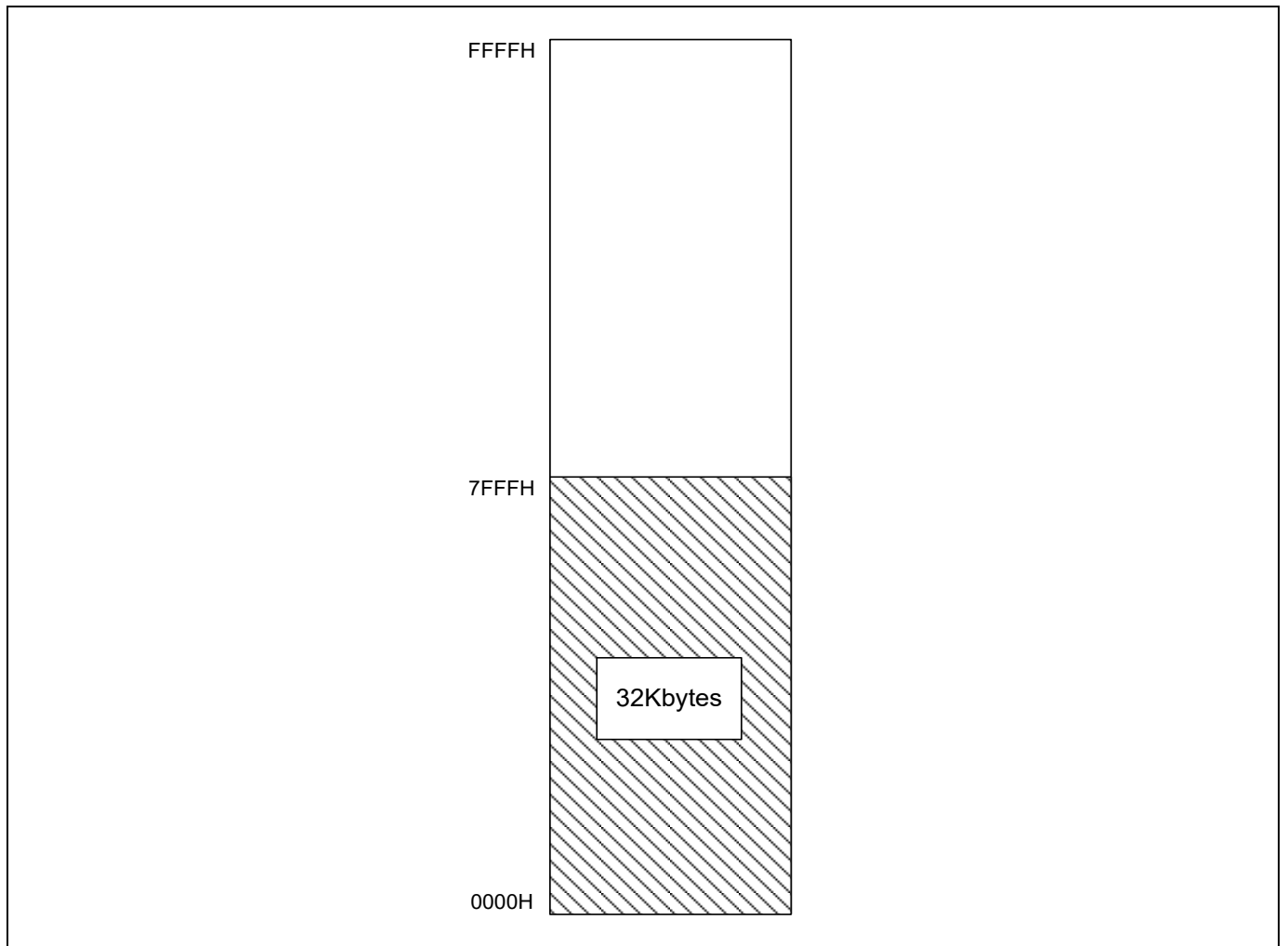


Figure 8.1 程序存储器

注意)

1. 32K 字节包括中断向量地址

8.2 数据存储

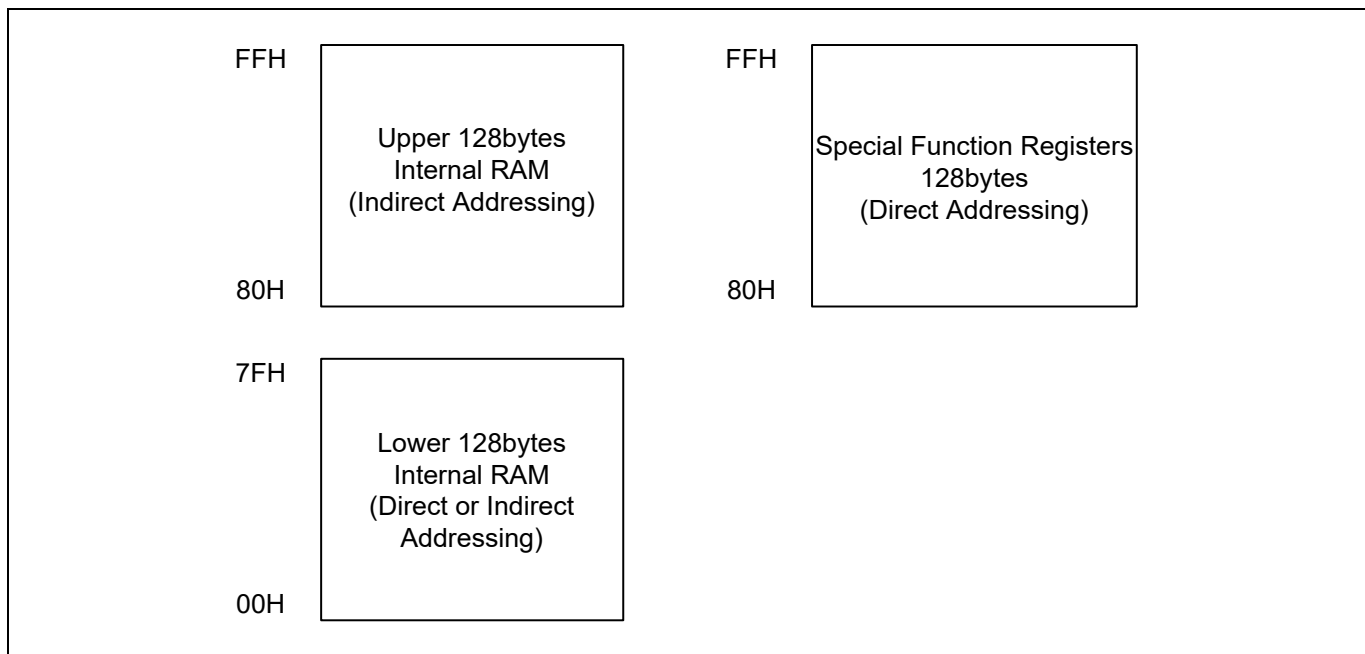


Figure 8.2 数据存储

内部数据存储分为3个部分，低 128 bytes, 高 128 bytes, SFR.

内部数据存储地址是一个字节宽度，这就意味着地址空间只有 256 bytes. 但是，事实上通过内部RAM寻址方式调节可提供384 bytes.直接寻址 高于 7FH 的存储空间和间接寻址 高于7FH 的存储空间进入到不同的存储空间. 图 8.2 所示高128 bytes 和SFR 占用相同的区域, 80H 到 FFH, 但他们本身却是独立的.

图 8.3 显示的是所有8051系列的低128字节RAM. 最低的 32 字节组成了4个寄存器组. 程序指令可以调用从 R0 到 R7的寄存器. 程序状态字中的两位决定选用哪个寄存器组. 这样可以更有效的使用代码空间, 因为寄存器指令要比直接寻址指令更短.

寄存器组后边的 16 bytes组成了可位寻址的存储空间. 8051 指令设置包括了位指令宽度的操作, 但是这个区域的128 bits 可以通过这些指令直接寻址. 该区域的地址是00H 到 7FH.

低128字节的所有字节均可以直接访问或间接访问. 高128字节只能间接寻址访问. 这些区域用作数据RAM和堆栈.

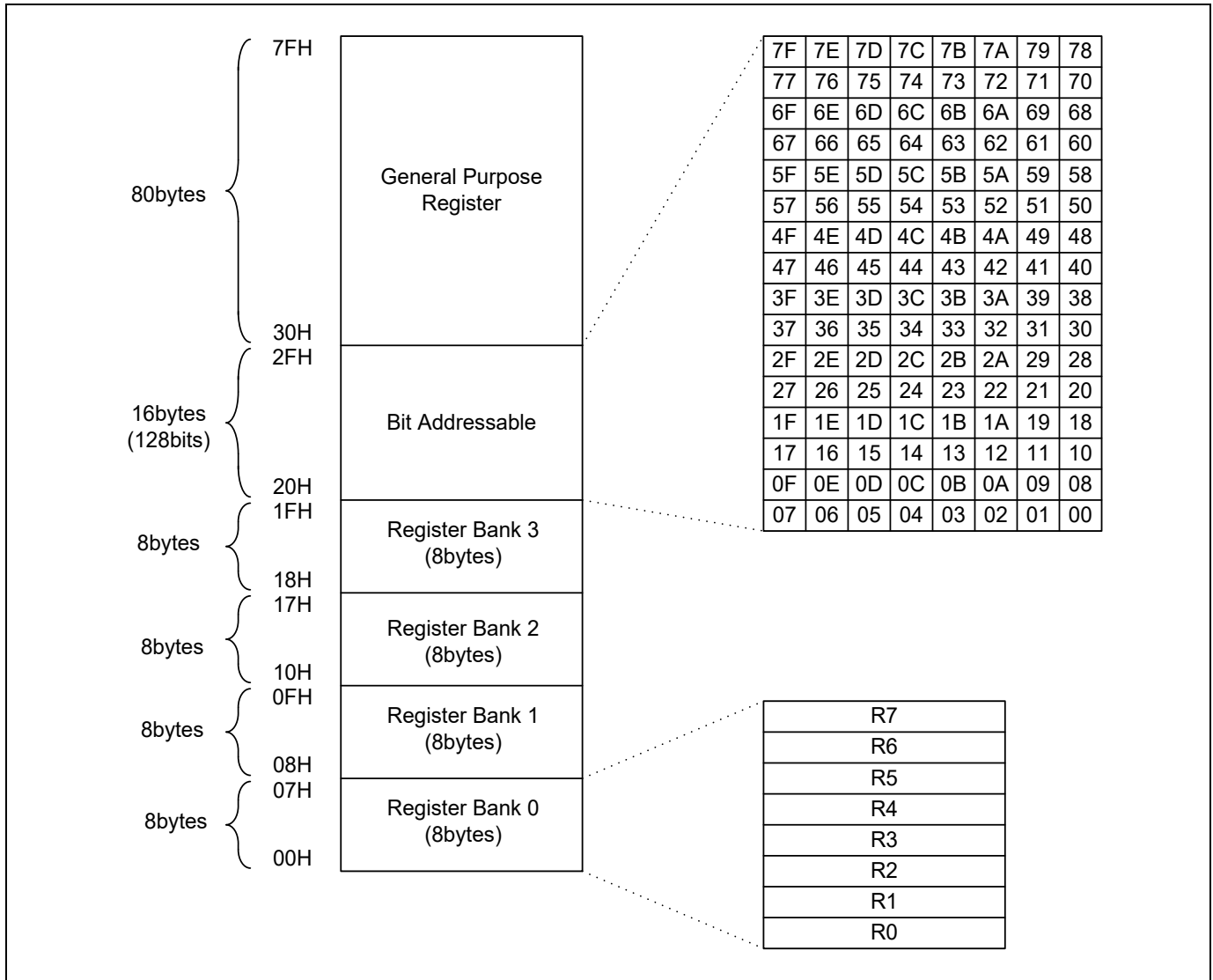


Figure 8.3 低 128字节 RAM

8.3 外部数据存储器

MC96F6432A 有 768 字节 XRAM 和 XSFR. 该区域与 RAM/FLASH没有联系. 可通过8位的SFR进行读写.

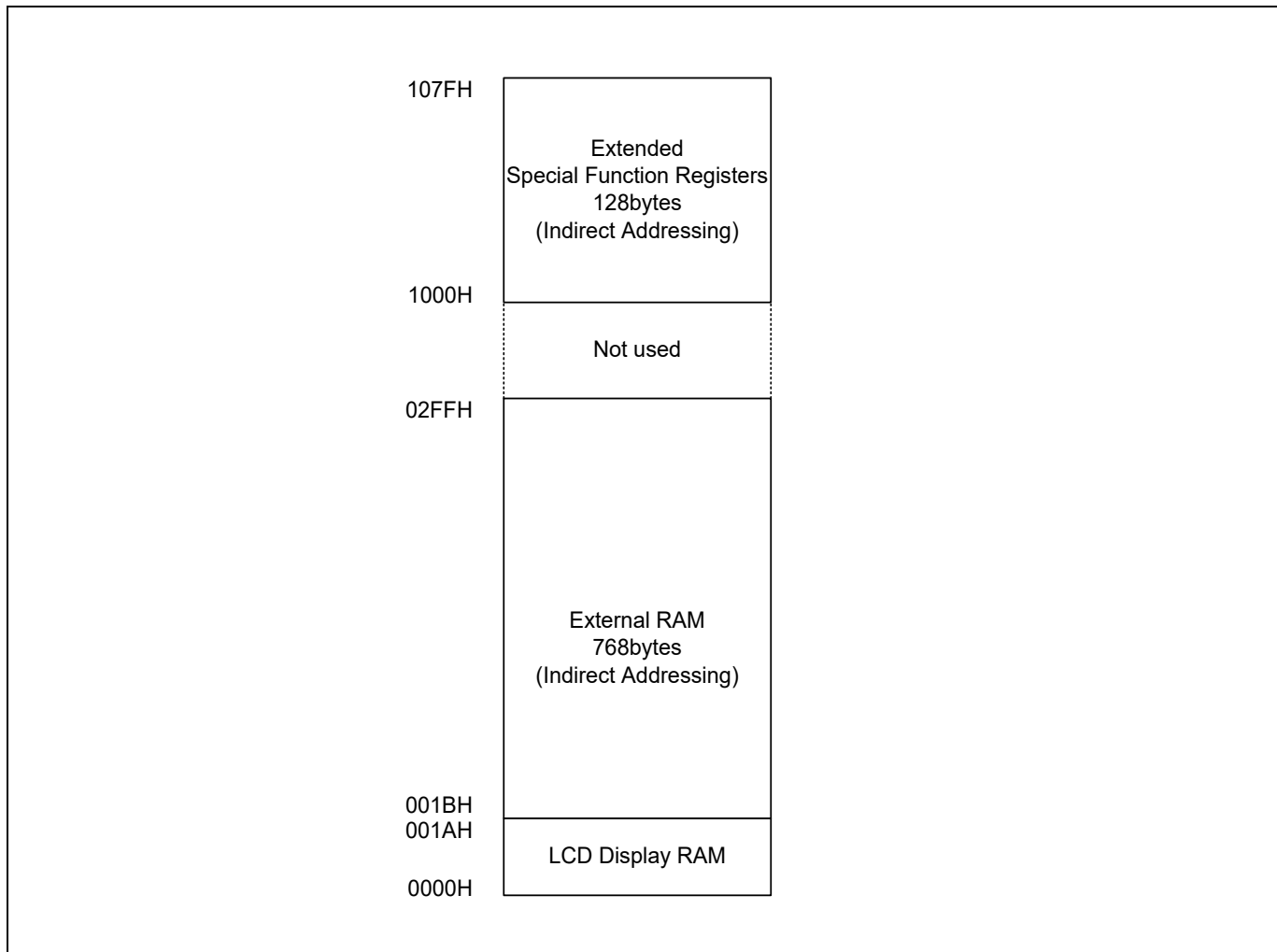


Figure 8.4 XDATA 存储区域

8.4 SFR

8.4.1 SFR 图表摘要

-	Reserved
	M8051 compatible

	00H/8H ⁽¹⁾	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
0F8H	IP1	-	FSADRH	FSADRM	FSADRL	FIDR	FMCR	P5FSR
0F0H	B	USI1ST1	USI1ST2	USI1BD	USI1SDHR	USI1DR	USI1SCLR	USI1SCHR
0E8H	RSTFR	USI1CR1	USI1CR2	USI1CR3	USI1CR4	USI1SAR	P3FSR	P4FSR
0E0H	ACC	USI0ST1	USI0ST2	USI0BD	USI0SDHR	USI0DR	USI0SCLR	USI0SCHR
0D8H	LVRCR	USI0CR1	USI0CR2	USI0CR3	USI0CR4	USI0SAR	P0DB	P15DB
0D0H	PSW	P5IO	P0FSRL	P0FSRH	P1FSRL	P1FSRH	P2FSRL	P2FSRH
0C8H	OSCCR	P4IO	-	-	-	-	-	-
0C0H	EIFLAG0	P3IO	T2CRL	T2CRH	T2ADRL	T2ADRH	T2BDRL	T2BDRH
0B8H	IP	P2IO	T1CRL	T1CRH	T1ADRL	T1ADRH	T1BDRL	T1BDRH
0B0H	P5	P1IO	T0CR	T0CNT	T0DR/ T0CDR	SPICR	SPIDR	SPISR
0A8H	IE	IE1	IE2	IE3	P0PU	P1PU	P2PU	P3PU
0A0H	P4	P0IO	EO	P4PU	EIPOL0L	EIPOL0H	EIFLAG1	EIPOL1
98H	P3	LCDCRL	LCDCRH	LCDCCR	ADCCRL	ADCCRH	ADCRL	ADCRH
90H	P2	P0OD	P1OD	P2OD	P4OD	P5PU	WTCR	BUZCR
88H	P1	WTDR/ WTCNT	SCCR	BITCR	BITCNT	WDTCR	WDTCR/ WTCNT	BUZDR
80H	P0	SP	DPL	DPH	DPL1	DPH1	LVICR	PCON

Table 8.1 SFR 图表摘要

注意)

1. 00H/8H, 这一列的寄存器支持位操作

	00H/8H ⁽¹⁾	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
1078H	-	-	-	-	-	-	-	-
1070H	-	-	-	-	-	-	-	-
1068H	-	-	-	-	-	-	-	-
1060H	-	-	-	-	-	-	-	-
1058H	-	-	-	-	-	-	-	-
1050H	-	-	-	-	-	-	-	-
1048H	-	-	-	-	-	-	-	-
1040H	-	-	-	-	-	-	-	-
1038H	XTFLSR	-	-	-	-	-	-	-
1030H	-	-	-	-	-	-	-	-
1028H	-	-	-	-	-	-	-	-
1020H	-	-	-	-	-	-	-	-
1018H	-	-	-	-	-	-	-	-
1010H	T4DLYA	T4DLYB	T4DLYC	T4DR	T4CAPR	T4CNT	-	-
1008H	T4PPRL	T4PPRH	T4ADRL	T4ADRH	T4BDRL	T4BDRH	T4CDRL	T4CDRH
1000H	T3CR	T3CNT/ T3DR/ T3CAPR	T4CR	T4PCR1	T4PCR2	T4PCR3	T4ISR	T4IMSK

Table 8.2 XSFR 图表摘要

8.4.2 SFR

Address	Function	Symbol	R/W	@Reset								
				7	6	5	4	3	2	1	0	
80H	P0 Data Register	P0	R/W	0	0	0	0	0	0	0	0	0
81H	Stack Pointer	SP	R/W	0	0	0	0	0	1	1	1	1
82H	Data Pointer Register Low	DPL	R/W	0	0	0	0	0	0	0	0	0
83H	Data Pointer Register High	DPH	R/W	0	0	0	0	0	0	0	0	0
84H	Data Pointer Register Low 1	DPL1	R/W	0	0	0	0	0	0	0	0	0
85H	Data Pointer Register High 1	DPH1	R/W	0	0	0	0	0	0	0	0	0
86H	Low Voltage Indicator Control Register	LVICR	R/W	–	–	0	0	0	0	0	0	0
87H	Power Control Register	PCON	R/W	0	–	–	–	0	0	0	0	0
88H	P1 Data Register	P1	R/W	0	0	0	0	0	0	0	0	0
89H	Watch Timer Data Register	WTDR	W	0	1	1	1	1	1	1	1	1
	Watch Timer Counter Register	WTCNT	R	–	0	0	0	0	0	0	0	0
8AH	System and Clock Control Register	SCCR	R/W	–	–	–	–	–	–	0	0	0
8BH	BIT Control Register	BITCR	R/W	0	0	0	–	0	0	0	0	1
8CH	Basic Interval Timer Counter Register	BITCNT	R	0	0	0	0	0	0	0	0	0
8DH	Watch Dog Timer Control Register	WDTCR	R/W	0	0	0	–	–	–	0	0	0
8EH	Watch Dog Timer Data Register	WDTDR	W	1	1	1	1	1	1	1	1	1
	Watch Dog Timer Counter Register	WDTCNT	R	0	0	0	0	0	0	0	0	0
8FH	BUZZER Data Register	BUZZDR	R/W	1	1	1	1	1	1	1	1	1
90H	P2 Data Register	P2	R/W	0	0	0	0	0	0	0	0	0
91H	P0 Open-drain Selection Register	P0OD	R/W	0	0	0	0	0	0	0	0	0
92H	P1 Open-drain Selection Register	P1OD	R/W	0	0	0	0	0	0	0	0	0
93H	P2 Open-drain Selection Register	P2OD	R/W	0	0	0	0	0	0	0	0	0
94H	P4 Open-drain Selection Register	P4OD	R/W	–	–	–	–	0	0	0	0	0
95H	P5 Pull-up Resistor Selection Register	P5PU	R/W	–	–	0	0	0	0	0	0	0
96H	Watch Timer Control Register	WTCR	R/W	0	–	–	0	0	0	0	0	0
97H	BUZZER Control Register	BUZZCR	R/W	–	–	–	–	–	0	0	0	0
98H	P3 Data Register	P3	R/W	0	0	0	0	0	0	0	0	0
99H	LCD Driver Control Low Register	LCDCRL	R/W	–	–	0	0	0	0	0	0	0
9AH	LCD Driver Control High Register	LCDCRH	R/W	–	–	–	0	–	–	0	0	0
9BH	LCD Contrast Control Register	LCDCCR	R/W	0	–	–	–	0	0	0	0	0
9CH	A/D Converter Control Low Register	ADCCRL	R/W	0	0	0	0	0	0	0	0	0
9DH	A/D Converter Control High Register	ADCCRH	R/W	0	–	0	0	0	0	0	0	0
9EH	A/D Converter Data Low Register	ADCDRL	R	x	x	x	x	x	x	x	x	x
9FH	A/D Converter Data High Register	ADCDRH	R	x	x	x	x	x	x	x	x	x

Table 8.3 SFR Map

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
A0H	P4 Data Register	P4	R/W	-	-	-	-	0	0	0	0
A1H	P0 Direction Register	P0IO	R/W	0	0	0	0	0	0	0	0
A2H	Extended Operation Register	EO	R/W	-	-	-	0	-	0	0	0
A3H	P4 Pull-up Resistor Selection Register	P4PU	R/W	-	-	-	-	0	0	0	0
A4H	External Interrupt Polarity 0 Low Register	EIPOL0L	R/W	0	0	0	0	0	0	0	0
A5H	External Interrupt Polarity 0 High Register	EIPOL0H	R/W	0	0	0	0	0	0	0	0
A6H	External Interrupt Flag 1 Register	EIFLAG1	R/W	0	0	0	0	0	0	0	0
A7H	External Interrupt Polarity 1 Register	EIPOL1	R/W	0	0	0	0	0	0	0	0
A8H	Interrupt Enable Register	IE	R/W	0	-	0	0	0	0	0	0
A9H	Interrupt Enable Register 1	IE1	R/W	-	-	0	0	0	0	-	0
AAH	Interrupt Enable Register 2	IE2	R/W	-	-	0	0	0	0	0	0
ABH	Interrupt Enable Register 3	IE3	R/W	-	-	0	0	0	0	0	0
ACH	P0 Pull-up Resistor Selection Register	P0PU	R/W	0	0	0	0	0	0	0	0
ADH	P1 Pull-up Resistor Selection Register	P1PU	R/W	0	0	0	0	0	0	0	0
AEH	P2 Pull-up Resistor Selection Register	P2PU	R/W	0	0	0	0	0	0	0	0
AFH	P3 Pull-up Resistor Selection Register	P3PU	R/W	0	0	0	0	0	0	0	0
B0H	P5 Data Register	P5	R/W	-	-	0	0	0	0	0	0
B1H	P1 Direction Register	P1IO	R/W	0	0	0	0	0	0	0	0
B2H	Timer 0 Control Register	T0CR	R/W	0	-	0	0	0	0	0	0
B3H	Timer 0 Counter Register	T0CNT	R	0	0	0	0	0	0	0	0
B4H	Timer 0 Data Register	T0DR	R/W	1	1	1	1	1	1	1	1
	Timer 0 Capture Data Register	T0CDR	R	0	0	0	0	0	0	0	0
B5H	SPI 2 Control Register	SPICR	R/W	0	0	0	0	0	0	0	0
B6H	SPI 2 Data Register	SPIDR	R/W	0	0	0	0	0	0	0	0
B7H	SPI 2 Status Register	SPISR	R/W	0	0	0	-	0	0	-	-
B8H	Interrupt Priority Register	IP	R/W	-	-	0	0	0	0	0	0
B9H	P2 Direction Register	P2IO	R/W	0	0	0	0	0	0	0	0
BAH	Timer 1 Control Low Register	T1CRL	R/W	0	0	0	0	-	0	0	0
BBH	Timer 1 Conter High Register	T1CRH	R/W	0	-	0	0	-	-	-	0
BCH	Timer 1 A Data Low Register	T1ADRL	R/W	1	1	1	1	1	1	1	1
BDH	Timer 1 A Data High Register	T1ADRH	R/W	1	1	1	1	1	1	1	1
BEH	Timer 1 B Data Low Register	T1BDRL	R/W	1	1	1	1	1	1	1	1
BFH	Timer 1 B Data High Register	T1BDRH	R/W	1	1	1	1	1	1	1	1

Table 8.4 SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset								
				7	6	5	4	3	2	1	0	
C0H	External Interrupt Flag 0 Register	EIFLAG0	R/W	0	0	0	0	0	0	0	0	0
C1H	P3 Direction Register	P3IO	R/W	0	0	0	0	0	0	0	0	0
C2H	Timer 2 Control Low Register	T2CRL	R/W	0	0	0	0	–	0	–	0	0
C3H	Timer 2 Control High Register	T2CRH	R/W	0	–	0	0	–	–	–	–	0
C4H	Timer 2 A Data Low Register	T2ADRL	R/W	1	1	1	1	1	1	1	1	1
C5H	Timer 2 A Data High Register	T2ADRH	R/W	1	1	1	1	1	1	1	1	1
C6H	Timer 2 B Data Low Register	T2BDRL	R/W	1	1	1	1	1	1	1	1	1
C7H	Timer 2 B Data High Register	T2BDRH	R/W	1	1	1	1	1	1	1	1	1
C8H	Oscillator Control Register	OSCCR	R/W	–	–	0	0	1	0	0	0	0
C9H	P4 Direction Register	P4IO	R/W	–	–	–	–	0	0	0	0	0
CAH	Reserved	–	–	–								
CBH	Reserved	–	–	–								
CCH	Reserved	–	–	–								
CDH	Reserved	–	–	–								
CEH	Reserved	–	–	–								
CFH	Reserved	–	–	–								
D0H	Program Status Word Register	PSW	R/W	0	0	0	0	0	0	0	0	0
D1H	P5 Direction Register	P5IO	R/W	–	–	0	0	0	0	0	0	0
D2H	P0 Function Selection Low Register	P0FSRL	R/W	–	0	0	0	0	0	0	0	0
D3H	P0 Function Selection High Register	P0FSRH	R/W	–	–	0	0	0	0	0	0	0
D4H	P1 Function Selection Low Register	P1FSRL	R/W	0	0	0	0	0	0	0	0	0
D5H	P1 Function Selection High Register	P1FSRH	R/W	0	0	0	0	0	0	0	0	0
D6H	P2 Function Selection Low Register	P2FSRL	R/W	–	–	0	0	0	0	0	0	0
D7H	P2 Function Selection High Register	P2FSRH	R/W	–	–	–	–	0	0	0	0	0
D8H	Low Voltage Reset Control Register	LVRCCR	R/W	0	–	–	0	0	0	0	0	0
D9H	USI0 Control Register 1	USI0CR1	R/W	0	0	0	0	0	0	0	0	0
DAH	USI0 Control Register 2	USI0CR2	R/W	0	0	0	0	0	0	0	0	0
DBH	USI0 Control Register 3	USI0CR3	R/W	0	0	0	0	0	0	0	0	0
DCH	USI0 Control Register 4	USI0CR4	R/W	0	–	–	0	0	–	0	0	0
DDH	USI0 Slave Address Register	USI0SAR	R/W	0	0	0	0	0	0	0	0	0
DEH	P0 De-Debounce Enable Register	P0DB	R/W	0	0	0	0	0	0	0	0	0
DFH	P1/P5 De-Debounce Enable Register	P15DB	R/W	–	–	0	0	0	0	0	0	0

Table 8.5 SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset								
				7	6	5	4	3	2	1	0	
E0H	Accumulator Register	ACC	R/W	0	0	0	0	0	0	0	0	0
E1H	USI0 Status Register 1	USI0ST1	R/W	1	0	0	0	–	0	0	0	0
E2H	USI0 Status Register 2	USI0ST2	R	0	0	0	0	0	0	0	0	0
E3H	USI0 Baud Rate Generation Register	USI0BD	R/W	1	1	1	1	1	1	1	1	1
E4H	USI0 SDA Hold Time Register	USI0SDHR	R/W	0	0	0	0	0	0	0	0	1
E5H	USI0 Data Register	USI0DR	R/W	0	0	0	0	0	0	0	0	0
E6H	USI0 SCL Low Period Register	USI0SCLR	R/W	0	0	1	1	1	1	1	1	1
E7H	USI0 SCL High Period Register	USI0SCHR	R/W	0	0	1	1	1	1	1	1	1
E8H	Reset Flag Register	RSTFR	R/W	1	x	0	0	x	–	–	–	–
E9H	USI1 Control Register 1	USI1CR1	R/W	0	0	0	0	0	0	0	0	0
EAH	USI1 Control Register 2	USI1CR2	R/W	0	0	0	0	0	0	0	0	0
EBH	USI1 Control Register 3	USI1CR3	R/W	0	0	0	0	0	0	0	0	0
ECH	USI1 Control Register 4	USI1CR4	R/W	0	–	–	0	0	–	0	0	0
EDH	USI1 Slave Address Register	USI1SAR	R/W	0	0	0	0	0	0	0	0	0
EEH	P3 Function Selection Register	P3FSR	R/W	0	0	0	0	0	0	0	0	0
EFH	P4 Function Selection Register	P4FSR	R/W	–	0	0	0	0	0	0	0	0
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0	0
F1H	USI1 Status Register 1	USI1ST1	R/W	1	0	0	0	–	0	0	0	0
F2H	USI1 Status Register 2	USI1ST2	R	0	0	0	0	0	0	0	0	0
F3H	USI1 Baud Rate Generation Register	USI1BD	R/W	1	1	1	1	1	1	1	1	1
F4H	USI1 SDA Hold Time Register	USI1SDHR	R/W	0	0	0	0	0	0	0	0	1
F5H	USI1 Data Register	USI1DR	R/W	0	0	0	0	0	0	0	0	0
F6H	USI1 SCL Low Period Register	USI1SCLR	R/W	0	0	1	1	1	1	1	1	1
F7H	USI1 SCL High Period Register	USI1SCHR	R/W	0	0	1	1	1	1	1	1	1
F8H	Interrupt Priority Register 1	IP1	R/W	–	–	0	0	0	0	0	0	0
F9H	Reserved	–	–	–								
FAH	Flash Sector Address High Register	FSADRH	R/W	–	–	–	–	0	0	0	0	0
FBH	Flash Sector Address Middle Register	FSADRM	R/W	0	0	0	0	0	0	0	0	0
FCH	Flash Sector Address Low Register	FSADRL	R/W	0	0	0	0	0	0	0	0	0
FDH	Flash Identification Register	FIDR	R/W	0	0	0	0	0	0	0	0	0
FEH	Flash Mode Control Register	FMCR	R/W	0	–	–	–	–	0	0	0	0
FFH	P5 Function Selection Register	P5FSR	R/W	–	–	0	0	0	0	0	0	0

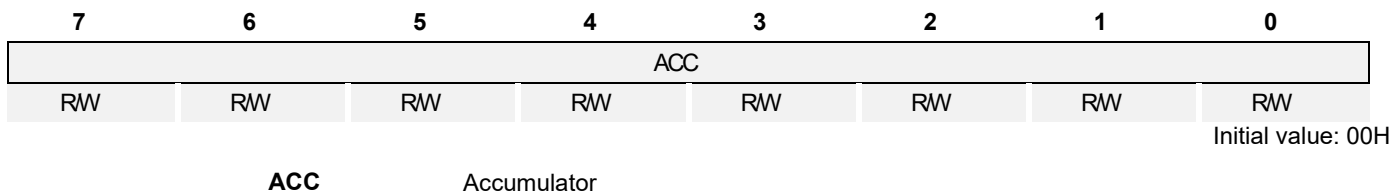
Table 8.6 SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset								
				7	6	5	4	3	2	1	0	
1000H	Timer 3 Control Register	T3CR	R/W	0	–	0	0	0	0	0	0	0
1001H	Timer 3 Counter Register	T3CNT	R	0	0	0	0	0	0	0	0	0
	Timer 3 Data Register	T3DR	W	1	1	1	1	1	1	1	1	1
	Timer 3 Capture Data Register	T3CAPR	R	0	0	0	0	0	0	0	0	0
1002H	Timer 4 Control Register	T4CR	R/W	0	0	0	0	0	0	0	0	0
1003H	Timer 4 PWM Control Register 1	T4PCR1	R/W	0	0	0	0	0	0	0	0	0
1004H	Timer 4 PWM Control Register 2	T4PCR2	R/W	0	0	0	0	0	0	0	0	0
1005H	Timer 4 PWM Control Register 3	T4PCR3	R/W	–	0	0	0	–	–	–	–	–
1006H	Timer 4 Interrupt Status Register	T4ISR	R/W	0	0	0	0	0	–	–	–	–
1007H	Timer 4 Interrupt Mask Register	T4MSK	R/W	0	0	0	0	0	–	–	–	–
1008H	Timer 4 PWM Period Low Register	T4PPRL	R/W	1	1	1	1	1	1	1	1	1
1009H	Timer 4 PWM Period High Register	T4PPRH	R/W	–	–	–	–	–	–	–	0	0
100AH	Timer 4 PWM A Duty Low Register	T4ADRL	R/W	0	1	1	1	1	1	1	1	1
100BH	Timer 4 PWM A Duty High Register	T4ADRH	R/W	–	–	–	–	–	–	–	0	0
100CH	Timer 4 PWM B Duty Low Register	T4BDRL	R/W	0	1	1	1	1	1	1	1	1
100DH	Timer 4 PWM B Duty High Register	T4BDRH	R/W	–	–	–	–	–	–	–	0	0
100EH	Timer 4 PWM C Duty Low Register	T4CDRL	R/W	0	1	1	1	1	1	1	1	1
100FH	Timer 4 PWM C Duty High Register	T4CDRH	R/W	–	–	–	–	–	–	–	0	0
1010H	Timer 4 PWM A Delay Register	T4DLYA	R/W	0	0	0	0	0	0	0	0	0
1011H	Timer 4 PWM B Delay Register	T4DLYB	R/W	0	0	0	0	0	0	0	0	0
1012H	Timer 4 PWM C Delay Register	T4DLYC	R/W	0	0	0	0	0	0	0	0	0
1013H	Timer 4 Data Register	T4DR	R/W	1	1	1	1	1	1	1	1	1
1014H	Timer 4 Capture Data Register	T4CAPR	R	0	0	0	0	0	0	0	0	0
1015H	Timer 4 Counter Register	T4CNT	R	0	0	0	0	0	0	0	0	0
.....												
1038H	X-tal Filter Selection Register	XTFLSR	R/W	0	0	0	0	0	0	–	–	0
.....												
107FH	Reserved	–	–	–								

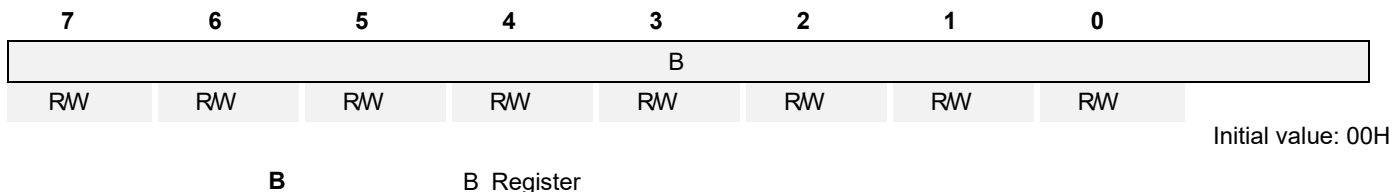
Table 8.7 XSFR Map (Continued)

8.4.3 SFR Map

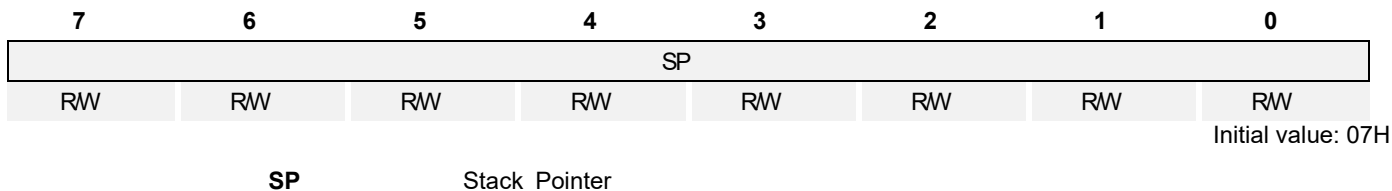
ACC (累加器): E0H



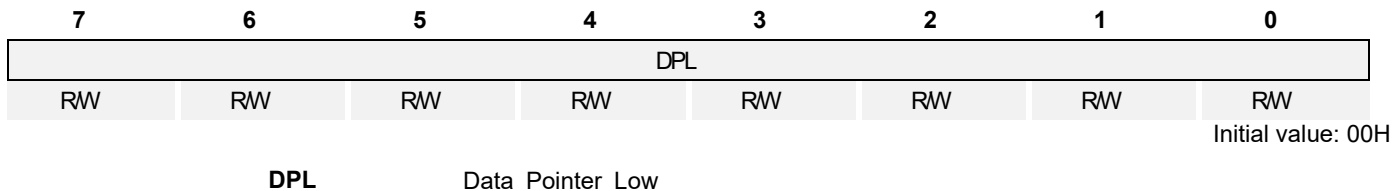
B (B Register): F0H



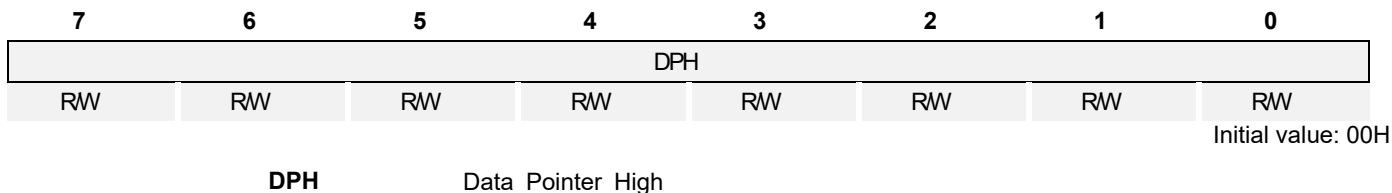
SP (Stack Pointer): 81H



DPL (Data Pointer Register Low): 82H



DPH (Data Pointer Register High): 83H



DPL1 (Data Pointer Register Low 1): 84H

7	6	5	4	3	2	1	0
DPL1							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

DPL1 Data Pointer Low 1

DPH1 (Data Pointer Register High 1): 85H

7	6	5	4	3	2	1	0
DPH1							
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

DPH1 Data Pointer High 1

PSW (Program Status Word Register): D0H

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

Initial value: 00H

CY 进位标志
AC 辅助进位标志
F0 通用用户可定义标志
RS1 寄存器组选择位 1
RS0 寄存器组选择位 0
OV 溢出标志位
F1 用户可定义标志位
P 奇偶标志. 每个指令周期通过硬件设置/清除来表示累加器中1的数量的奇偶

EO (Extended Operation Register): A2H

7	6	5	4	3	2	1	0
–	–	–	TRAP_EN	–	DPSEL2	DPSEL1	DPSEL0
–	–	–	RW	–	RW	RW	RW

Initial value: 00H

TRAP_EN 选择指令 (Keep always '0').
0 Select MOV @ (DPTR++), A
1 Select Software TRAP Instruction

DPSEL[2:0] 选择数据指针寄存器组

DPSEL2	DPSEL1	DPSEL0	Description
0	0	0	DPTR0
0	0	1	DPTR1
Reserved			

9 I/O 口

9.1 I/O 口

The MC96F6432A有6组 I/O 口 (P0 ~ P5). 通过软件设置可以轻松的把每个口配置为I/O 口, 内部上拉和漏极开路以匹配不同的系统结构和设计要求. P0具有根据引脚状态变化引起中断的功能.

9.2 端口寄存器

9.2.1 数据寄存器 (Px)

数据寄存器是双向 I/O 口. 如果作为输出口使用, 数据可以写入到Px的相应位. 如果设置为输入口, 数据可以Px的相应位读到.

9.2.2 方向寄存器 (PxIO)

每个 I/O 口可以通过PxIO 寄存器单独的作为输入或输出来使用, 清除相应的位将会使其变为输入模式. 置位相应的位就会使其变为输出口. 系统复位后可以清除大部分位, 但是有一些不能被清除.

9.2.3 上拉电阻选择寄存器 (PxPU)

片内上拉寄存器可以单独的将其和上拉寄存器连接(PxPU). 上拉寄存器选择控制每个口的上拉寄存器的使能和关闭. 相应的位置 1, 这个脚的上拉寄存器就打开. 如果清零, 上拉就关闭. 所有的位复位后清零.

9.2.4 漏极开路选择寄存器(PxOD)

P0 ~ P4和P5具有内部漏极开路选择寄存器. 控制着每个口漏极开路的开关. 复位后大部分端口变为推-拉状态, 但是有一些口变为漏极开路状态.

9.2.5 消抖使能寄存器 (PxDB)

P0[7:2], P1[2:1], P1[7:6], P52 和 P54具有去抖功能. 去抖时钟是 $fx/1$, $fx/4$, 和 $fx/4096$

9.2.6 端口功能选择寄存器 (PxFSR)

这些寄存器是用来定义端口的复用功能，请记得将这些寄存器设置到相应的功能。复位后PxFSR 被清除到 '00H'，使所有口变为普通 I/O。

9.2.7 寄存器图

Name	Address	Direction	Default	Description
P0	80H	R/W	00H	P0 数据寄存器
P0IO	A1H	R/W	00H	P0 方向寄存器
P0PU	ACH	R/W	00H	P0 上拉电阻选择寄存器
P0OD	91H	R/W	00H	P0 漏极开路选择寄存器
P0DB	DEH	R/W	00H	P0 消抖使能寄存器
P0FSRH	D3H	R/W	00H	P0 功能选择高位寄存器
P0FSRL	D2H	R/W	00H	P0 功能选择低位寄存器
P1	88H	R/W	00H	P1 数据寄存器
P1IO	B1H	R/W	00H	P1 方向寄存器
P1PU	ADH	R/W	00H	P1 上拉电阻选择寄存器
P1OD	92H	R/W	00H	P1 漏极开路选择寄存器
P15DB	DFH	R/W	00H	P1/P5 消抖使能寄存器
P1FSRH	D5H	R/W	00H	P1 功能选择高位寄存器
P1FSRL	D4H	R/W	00H	P1 功能选择低位寄存器
P2	90H	R/W	00H	P2 数据寄存器
P2IO	B9H	R/W	00H	P2 方向寄存器
P2PU	AEH	R/W	00H	P2 上拉电阻选择寄存器
P2OD	93H	R/W	00H	P2 漏极开路选择寄存器
P2FSRH	D7H	R/W	00H	P2 功能选择高位寄存器
P2FSRL	D6H	R/W	00H	P2 功能选择低位寄存器
P3	98H	R/W	00H	P3 数据寄存器
P3IO	C1H	R/W	00H	P3 方向寄存器
P3PU	AFH	R/W	00H	P3 上拉电阻选择寄存器
P3FSR	EEH	R/W	00H	P3 功能选择寄存器
P4	A0H	R/W	00H	P4 数据寄存器
P4IO	C9H	R/W	00H	P4 方向寄存器
P4PU	A3H	R/W	00H	P4 上拉电阻选择寄存器
P4OD	94H	R/W	00H	P4 漏极开路选择寄存器
P4FSR	EFH	R/W	00H	P4 功能选择寄存器
P5	B0H	R/W	00H	P5 数据寄存器
P5IO	D1H	R/W	00H	P5 方向寄存器
P5PU	95H	R/W	00H	P5 上拉电阻选择寄存器
P5FSR	FFH	R/W	00H	P5 功能选择寄存器

Table 9.1 Port 寄存器图

9.3 P0 口

9.3.1 P0 端口说明

P0 是 8-bit I/O 口. P0 控制寄存器由P0 数据寄存器 (P0), P0方向寄存器 (P0IO), 防抖使能寄存器(P0DB), P0 上拉电阻选择寄存器 (P0PU), 和P0 漏极开路寄存器(P0OD)组成. 具体请参考端口功能选择寄存器.

9.3.2 P0寄存器说明

P0 (P0 数据寄存器): 80H

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

Initial value : 00H

P0[7:0] I/O 数据

P0IO (P0 方向寄存器): A1H

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

Initial value: 00H

P0IO[7:0] P0 数据 I/O 方向.
 0 输入
 1 输出

NOTE)

1. EC3/EINT0 ~ EINT5 功能在输入状态时才有

P0PU (P0 上拉电阻选择寄存器): ACH

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

Initial value: 00H

P0PU[7:0] 配置 P0 上拉电阻
 0 禁止
 1 使能

P0OD (P0 漏极开路选择寄存器): 91H

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

Initial value: 00H

P0OD[7:0] 配置 P0 口漏极开路

0	推-拉输出
1	漏极开路输出

P0DB (P0 消抖使能寄存器): DEH

7	6	5	4	3	2	1	0
DBCLK1	DBCLK0	P07DB	P06DB	P05DB	P04DB	P03DB	P02DB
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

DBCLK[1:0] 配置端口消抖时钟

DBCLK1	DBCLK0	Description
0	0	fx/1
0	1	fx/4
1	0	fx/4096
1	1	Reserved

P07DB 配置 P07 口消抖

0	禁止
1	使能

P06DB 配置 P06 口消抖

0	禁止
1	使能

P05DB 配置 P05 口消抖

0	禁止
1	使能

P04DB 配置 P04 口消抖

0	禁止
1	使能

P03DB 配置 P03 口消抖

0	禁止
1	使能

P02DB 配置 P02 口消抖

0	禁止
1	使能

NOTE)

1. 如果在使能引脚上时钟采样时相同的电平没有被连续三到四次检测到, 那么这个信号就被作为干扰淘汰掉.
2. 一个输入脉冲在整个过程中应该被检测到 3 个以上的时钟才被认为是有效.
3. 在 STOP 模式端口消抖被自动的关闭 STOP 唤醒之后又重新恢复.

P0FSRH (P0 功能选择高位寄存器): D3H

7	6	5	4	3	2	1	0
–	–	P0FSRH5	P0FSRH4	P0FSRH3	P0FSRH2	P0FSRH1	P0FSRH0
–	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

P0FSRH[5:4]	P07 功能选择		
	P0FSRH5	P0FSRH4	Description
	0	0	I/O Port (EINT5功能只有在输入状态时能使用)
	0	1	SEG22
	1	0	AN5
	1	1	PWM4CB
P0FSRH[3:2]	P06 功能选择		
	P0FSRH3	P0FSRH2	Description
	0	0	I/O Port (EINT4功能只有在输入状态时能使用)
	0	1	SEG23
	1	0	AN4
	1	1	PWM4CA
P0FSRH[1:0]	P05 功能选择		
	P0FSRH1	P0FSRH0	Description
	0	0	I/O Port (EINT3功能只有在输入状态时能使用)
	0	1	SEG24
	1	0	AN3
	1	1	PWM4BB

P0FSRL (P0 功能选择低位寄存器): D2H

7	6	5	4	3	2	1	0
–	P0FSRL6	P0FSRL5	P0FSRL4	P0FSRL3	P0FSRL2	P0FSRL1	P0FSRL0
–	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P0FSRL[6:5] P04 功能选择

P0FSRL6	P0FSRL5	Description
0	0	I/O Port(EINT2功能只有在输入状态时能使用)
0	1	SEG25
1	0	AN2
1	1	PWM4BA

P0FSRL[4:3] P03 功能选择

P0FSRL4	P0FSRL3	Description
0	0	I/O Port(EINT1功能只有在输入状态时能使用)
0	1	SEG26
1	0	AN1
1	1	PWM4AB

P0FSRL[2:1] P02 功能选择

P0FSRL2	P0FSRL1	Description
0	0	I/O Port (EINT0功能只有在输入状态时能使用)
0	1	AVREF
1	0	AN0
1	1	T4O/PWM4A

P0FSRL0 P01 功能选择

0	I/O Port
1	T3O 功能

9.4 P1 口

9.4.1 P1 端口说明

P1 是 8-bit I/O 口。P1 控制寄存器由 P1 数据寄存器 (P1), P1 方向寄存器 (P1IO), 消抖使能寄存器 (P15DB), P1 上拉电阻选择寄存器 (P1PU), 和 P1 漏极开路 (P1OD)。具体请参考端口功能寄存器。

9.4.2 Register description for P1

P1 (P1 数据寄存器): 88H

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P1[7:0] I/O 数据

P1IO (P1 方向寄存器): B1H

7	6	5	4	3	2	1	0
P17IO	P16IO	P15IO	P14IO	P13IO	P12IO	P11IO	P10IO
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P1IO[7:0] P1 数据 I/O 方向

0 输入

1 输出

NOTE)

1. EINT6/ENINT7/EINT11/EINT12/SS2/EC1 功能在输入状态时才有

P1PU (P1 上拉电阻选择寄存器): ADH

7	6	5	4	3	2	1	0
P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P1PU[7:0] 配置 P1 上拉电阻

0 禁止

1 使能

P10D (P1 漏极开路选择寄存器): 92H

7	6	5	4	3	2	1	0
P17OD	P16OD	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 08H

P10D[7:0]	配置 P1 口漏极开路
0	推-拉输出
1	漏极开路输出

P15DB (P1/P5 消抖使能寄存器): DFH

7	6	5	4	3	2	1	0
-	-	P54DB	P52DB	P17DB	P16DB	P12DB	P11DB
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

P54DB	配置 P54 口消抖
0	禁止
1	使能
P52DB	配置 P52 口消抖
0	禁止
1	使能
P17DB	配置 P17 口消抖
0	禁止
1	使能
P16DB	配置 P16 口消抖
0	禁止
1	使能
P12DB	配置 P12 口消抖
0	禁止
1	使能
P11DB	配置 P11 口消抖
0	禁止
1	使能

NOTE)

1. 如果在使能引脚上时钟采样时相同的电平没有被连续三到四次检测到，那么这个信号就被作为干扰淘汰掉。
2. 一个输入脉冲在整个过程中应该被检测到 3 个以上的时钟才被认为是有效。
3. 在 STOP 模式端口消抖被自动的关闭 STOP 唤醒之后又重新恢复。
4. 请参考 P1 和 P5 口消抖时钟。

P1FSRH (P1 功能选择高位寄存器): D5H

7	6	5	4	3	2	1	0
P1FSRH7	P1FSRH6	P1FSRH5	P1FSRH4	P1FSRH3	P1FSRH2	P1FSRH1	P1FSRH0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P1FSRH[7:6]	P17 功能选择		
	P1FSRH7	P1FSRH6	Description
	0	0	I/O Port(EINT6/SS2功能只有在输入状态时才能使用)
	0	1	SEG21
	1	0	AN6
	1	1	Not used
P1FSRH[5:4]	P16 功能选择		
	P1FSRH5	P1FSRH4	Description
	0	0	I/O Port (EINT7功能只有在输入状态时才能使用)
	0	1	SEG20
	1	0	AN7
	1	1	SCK2
P1FSRH[3:2]	P15 功能选择		
	P1FSRH3	P1FSRH2	Description
	0	0	I/O Port
	0	1	SEG19
	1	0	AN8
	1	1	MISO2
P1FSRH[1:0]	P14 功能选择		
	P1FSRH1	P0FSRH0	Description
	0	0	I/O Port
	0	1	SEG18
	1	0	AN9
	1	1	MOSI2

P1FSRL (P1 功能选择低位寄存器): D4H

7	6	5	4	3	2	1	0
P1FSRL7	P1FSRL6	P1FSRL5	P1FSRL4	P1FSRL3	P1FSRL2	P1FSRL1	P1FSRL0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P1FSRL[7:6] P13 功能选择

P1FSRL7	P1FSRL6	Description
0	0	I/O Port(EC1功能只有在输入状态时才能使用)
0	1	SEG17
1	0	AN10
1	1	BUZO

P1FSRL[5:4] P12功能选择

P1FSRL5	P1FSRL4	Description
0	0	I/O Port(EINT11功能只有在输入状态时才能使用)
0	1	SEG16
1	0	AN11
1	1	T10/PWM10

P1FSRL[3:2] P11 功能选择

P1FSRL3	P1FSRL2	Description
0	0	I/O Port(EINT12功能只有在输入状态时才能使用)
0	1	SEG15
1	0	AN12
1	1	T20/PWM20

P1FSRL[1:0] P10 功能选择

P1FSRL1	P1FSRL0	Description
0	0	I/O Port
0	1	SEG14
1	0	AN13
1	1	RXD1/SCL1/MISO1

9.5 P2 口

9.5.1 P2 端口说明

P2 是 8-bit I/O 口。P2 控制寄存器由 P2 数据寄存器 (P2), P2 方向寄存器(P2IO), P2上拉电阻选择寄存器(P2PU) 和 P2 漏极开路选择寄存器 (P2OD)。具体请参考端口功能寄存器。

9.5.2 P2 寄存器说明

P2 (P2 数据寄存器): 90H

7	6	5	4	3	2	1	0
P27	P26	P25	P24	P23	P22	P21	P20
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P2[7:0] I/O 数据

P2IO (P2 方向寄存器): B9H

7	6	5	4	3	2	1	0
P27IO	P26IO	P25IO	P24IO	P23IO	P22IO	P21IO	P20IO
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P2IO[7:0] P2 数据 I/O 方向

0 输入

1 输出

NOTE)

1. SS1 功能只有在输入状态时才能使用

P2PU (P2 上拉电阻选择寄存器): AEH

7	6	5	4	3	2	1	0
P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P2PU[7:0] 配置 P2 上拉电阻

0 禁止

1 使能

P2OD (P2 漏极开路选择寄存器): 93H

7	6	5	4	3	2	1	0
P27OD	P26OD	P25OD	P24OD	P23OD	P22OD	P21OD	P20OD
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P2OD[7:0] 配置 P2 口漏极开路

0 推-拉输出

1 漏极开路输出

P2FSRH (P2 功能选择高位寄存器): D7H

7	6	5	4	3	2	1	0
-	-	-	-	P2FSRH3	P2FSRH2	P2FSRH1	P2FSRH0
-	-	-	-	RW	RW	RW	RW

Initial value: 00H

- P2FSRH3** P27 功能选择
 - 0 I/O Port
 - 1 SEG6
- P2FSRH2** P26 功能选择
 - 0 I/O Port
 - 1 SEG7
- P2FSRH1** P25 功能选择
 - 0 I/O Port
 - 1 SEG8
- P2FSRH0** P24 功能选择
 - 0 I/O Port
 - 1 SEG9

P2FSRL (P2 功能选择低位寄存器): D6H

7	6	5	4	3	2	1	0
-	-	P2FSRL5	P2FSRL4	P2FSRL3	P2FSRL2	P2FSRL1	P2FSRL0
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

- P2FSRL5** P23 功能选择
 - 0 I/O Port
 - 1 SEG10
- P2FSRL4** P22功能选择
 - 0 I/O Port(SS1功能只有在输入状态时才能使用)
 - 1 SEG11
- P2FSRL[3:2]** P21 功能选择

P2FSRL3	P2FSRL2	Description
0	0	I/O Port
0	1	SEG12
1	0	AN15
1	1	SCK1
- P2FSRL[1:0]** P20 功能选择

P2FSRL1	P1FSRL0	Description
0	0	I/O Port
0	1	SEG13
1	0	AN14
1	1	TXD1/SDA1/MOSI1

9.6 P3 口

9.6.1 P3 端口说明

P3 是 8-bit I/O 口。P3 控制寄存器由 P3 数据寄存器 (P3), P3方向寄存器(P3IO), P3上拉电阻选择寄存器(P3PU)组成。具体请参考端口功能寄存器。

9.6.2 P3 寄存器说明

P3 (P3 数据寄存器): 98H

7	6	5	4	3	2	1	0
P37	P36	P35	P34	P33	P32	P31	P30
RW	RW	RW	RW	RW	RW	RW	RW

Initial value : 00H

P3[7:0] I/O 数据

P3IO (P3 方向寄存器): C1H

7	6	5	4	3	2	1	0
P37IO	P36IO	P35IO	P34IO	P33IO	P32IO	P31IO	P30IO
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P3IO[7:0] P3 数据 I/O 方向
 0 输入
 1 输出

P3PU (P3 上拉电阻选择寄存器): AFH

7	6	5	4	3	2	1	0
P37PU	P36PU	P35PU	P34PU	P33PU	P32PU	P31PU	P30PU
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P3PU[7:0] 配置 P3 上拉电阻
 0 禁止
 1 使能

P3FSR (P3 功能选择寄存器): EEH

7	6	5	4	3	2	1	0
P3FSR7	P3FSR6	P3FSR5	P3FSR4	P3FSR3	P3FSR2	P3FSR1	P3FSR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P3FSR7	P37 功能选择
	0 I/O Port
	1 COM0
P3FSR6	P36 功能选择
	0 I/O Port
	1 COM1
P3FSR5	P35 功能选择
	0 I/O Port
	1 COM2/SEG0
P3FSR4	P34 功能选择
	0 I/O Port
	1 COM3/SEG1
P3FSR3	P33 功能选择
	0 I/O Port
	1 COM4/SEG2或COM0
P3FSR2	P32 功能选择
	0 I/O Port
	1 COM5/SEG3或COM1
P3FSR1	P31 功能选择
	0 I/O Port
	1 COM6/SEG4或COM2/SEG4
P3FSR0	P30 功能选择
	0 I/O Port
	1 COM7/SEG5或COM3/SEG5

注意)

1. 当 P30-P35 被用作 com/seg 复用功能时，芯片会根据 LCDCTRL 寄存器的值自动将其配置为 com 或 seg.
2. COM0-COM3 信号通过 P33-P30 脚输出. 涉及到 LCD 驱动控制高位寄存器(LCDCRH).

9.7 P4 口

9.7.1 P4 端口说明

P4 是8-bit I/O口。P4控制寄存器由 P4数据寄存器 (P4), P4方向寄存器(P4IO), P4上拉电阻选择寄存器(P4PU) 和 P4漏极开路选择寄存器 (P4OD)。具体请参考端口功能寄存器。

9.7.2 P4 寄存器说明

P4 (P4 数据寄存器): A0H

7	6	5	4	3	2	1	0
-	-	-	-	P43	P42	P41	P40
-	-	-	-	RW	RW	RW	RW

Initial value : 00H

P4[3:0] I/O 数据

P4IO (P4 方向寄存器): C9H

7	6	5	4	3	2	1	0
-	-	-	-	P43IO	P42IO	P41IO	P40IO
-	-	-	-	RW	RW	RW	RW

Initial value: 00H

P4IO[3:0] P4 数据 I/O 方向
 0 输入
 1 输出

NOTE)

1. SS0 功能在输入状态时才有

P4PU (P4 上拉电阻选择寄存器): A3H

7	6	5	4	3	2	1	0
-	-	-	-	P43PU	P42PU	P41PU	P40PU
-	-	-	-	RW	RW	RW	RW

Initial value: 00H

P4PU[3:0] 配置P4上拉电阻
 0 禁止
 1 使能

P4OD (P4 漏极开路选择寄存器): 94H

7	6	5	4	3	2	1	0
-	-	-	-	P43OD	P42OD	P41OD	P40OD
-	-	-	-	RW	RW	RW	RW

Initial value: 00H

P4OD[3:0] 配置 P4 口漏极开路
 0 推-拉输出
 1 漏极开路输出

P4FSR (P4 功能选择寄存器): EFH

7	6	5	4	3	2	1	0
-	P4FSR6	P4FSR5	P4FSR4	P4FSR3	P4FSR2	P4FSR1	P4FSR0
-	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

P4FSR6 P43 功能选择
 0 I/O Port(SS0 功能在输入状态时才有)
 1 VLC0

P4FSR[5:4] P42 功能选择

P4FSR5	P4FSR4	Description
0	0	I/O Port
0	1	VLC1
1	0	SCK0
1	1	Not used

P4FSR[3:2] P41 功能选择

P4FSR3	P4FSR2	Description
0	0	I/O Port
0	1	VLC2
1	0	TXD0/SDA0/MOSI0
1	1	Not used

P4FSR6[1:0] P40 功能选择

P4FSR1	P4FSR0	Description
0	0	I/O Port
0	1	VLC3
1	0	RXD0/SCL0/MISO0
1	1	Not used

9.8 P5 口

9.8.1 P5 端口说明

P5 是6-bit I/O 口。P5 控制寄存器由 P3 数据寄存器 (P5), P3方向寄存器(P5IO), P5上拉电阻选择寄存器(P5PU) 和 P5漏极开路选择寄存器 (P5OD)。具体请参考端口功能寄存器。

9.8.2 P5 寄存器说明

P5 (P5 数据寄存器): B0H

7	6	5	4	3	2	1	0
–	–	P55	P54	P53	P52	P51	P50
–	–	RW	RW	RW	RW	RW	RW

Initial value : 00H

P5[5:0] I/O 数据

P5IO (P5 方向寄存器): D1H

7	6	5	4	3	2	1	0
–	–	P55IO	P54IO	P53IO	P52IO	P51IO	P50IO
–	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

P5IO[5:0] P5 数据 I/O 方向
 0 输入
 1 输出

NOTE)

1. EC0/EINT8/EINT10/BLNK 功能在输入状态时才有

P5PU (P5 上拉电阻选择寄存器): 95H

7	6	5	4	3	2	1	0
–	–	P55PU	P54PU	P53PU	P52PU	P51PU	P50PU
–	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

P5PU[5:0] 配置P5上拉电阻
 0 禁止
 1 使能

P5FSR (P5 功能选择寄存器): FFH

7	6	5	4	3	2	1	0
-	-	P5FSR5	P5FSR4	P5FSR3	P5FSR2	P5FSR1	P5FSR0
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

P5FSR5	P54 功能选择		
	0	I/O Port(EINT10 功能在输入状态时才有)	
	1	SXOUT	
P5FSR[4:3]	P53 功能选择		
	P5FSR4	P5FSR3	Description
	0	0	I/O Port
	0	1	SXIN
	1	0	T00/PWM00
	1	1	Not used
P5FSR2	P51 功能选择		
	0	I/O Port	
	1	XIN	
P5FSR[1:0]	P50 功能选择		
	P5FSR1	P5FSR0	Description
	0	0	I/O Port
	0	1	XOUT
	1	0	Not used
	1	1	Not used

NOTE)

1. 注意 P55/RESETB 选项的配置.

10 中断控制

10.1 概述

MC96F6432A有 23个中断源. 各中断分别具有各种独立的使能寄存器位, 允许软件控制. 分配为4个中断优先级, 非可屏蔽中断源通常比其他使能中断源具有更高的优先级, 且不可通过软件控制。中断控制器有以下特性:

- 23个中断源接收中断
- 6优先级组
- 4个优先等级
- 多中断复用
- 如果同时接收到不同中断优先级的中断请求, 首先响应较高中断优先级的请求.
- 各中断位可通过EA 位和 IEx 位控制
- 中断响应时间: 单个中断3~9 个机器周期

可屏蔽中断通过四对中断使能寄存器 (IE, IE1, IE2, IE3)使能. IE, IE1, IE2, IE3的每一位分别单独的使能/禁止相应的中断源. 总中断控制位是 IE中的7位 (EA)。EA 置 '0', 所有中断禁止, EA置 '1', 中断通过中断使能寄存器中的其他位分别对其使能和禁止. EA位通常被清零后立即跳到中断服务程序, 置1则执行[RETI]指令. MC96F6432A支持4个优先级. 每个可屏蔽中断通过IP 和 IP1来分配中断优先等级.

中断默认的是电平模式触发, 如果有需要的话可以改变电平触发模式. 表10.1所示为有共同中断优先级的中断组..优先级通过中断优先寄存器的两位来设置 (一位是 IP, 另一位是 IP1).较高优先级的中断服务程序首先执行, 如果同时接收到两个不同优先级的中断请求, 那么优先级高的比优先级低的先被响应.

Interrupt Group	Highest Lowest				
	→				
0 (Bit0)	Interrupt 0	Interrupt 6	Interrupt 12	Interrupt 18	Highest Lowest
1 (Bit1)	Interrupt 1	Interrupt 7	Interrupt 13	Interrupt 19	
2 (Bit2)	Interrupt 2	Interrupt 8	Interrupt 14	Interrupt 20	
3 (Bit3)	Interrupt 3	Interrupt 9	Interrupt 15	Interrupt 21	
4 (Bit4)	Interrupt 4	Interrupt 10	Interrupt 16	Interrupt 22	
5 (Bit5)	Interrupt 5	Interrupt 11	Interrupt 17	Interrupt 23	

Table 10.1 中断优先等级

10.2 外部中断

在INT0, INT1, INT5, INT6 和 INT11引脚上的外部中断通过外部中断边沿触发器 (EIPOL0H/L) 和(EIPOL1)来接收不同的中断请求, 图10.1所示为外部中断边沿触发器1 (EIPOL1) 和外部中断边沿触发器2(EIPOL2).每个外部中断源可以通过位操作开关. 外部中断标志寄存器0 (EIFLAG0) 外部中断标志寄存器1 (EIFLAG1)提供外部中断的状态.

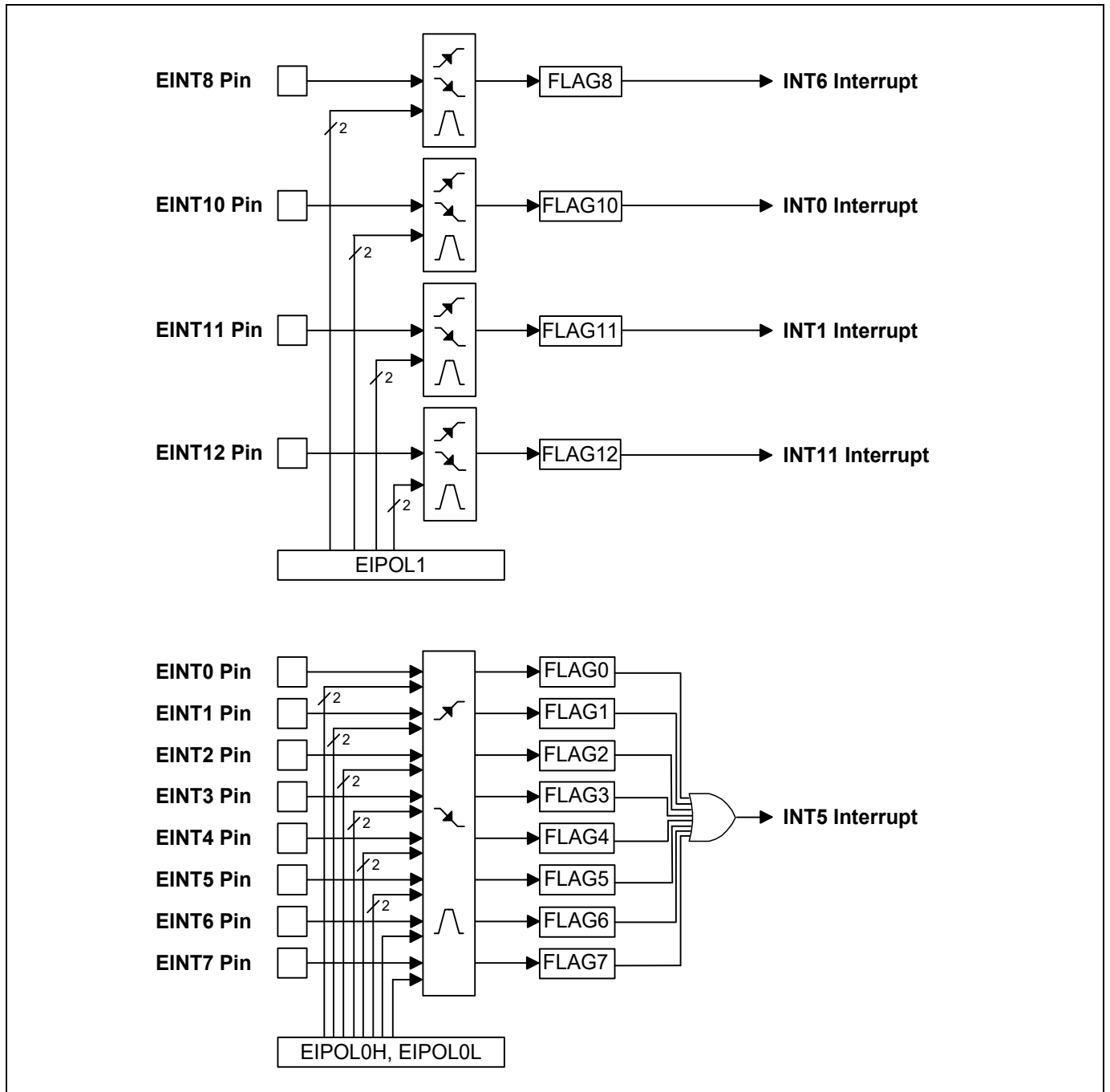


Figure 10.1 外部中断说明

10.3 方框图

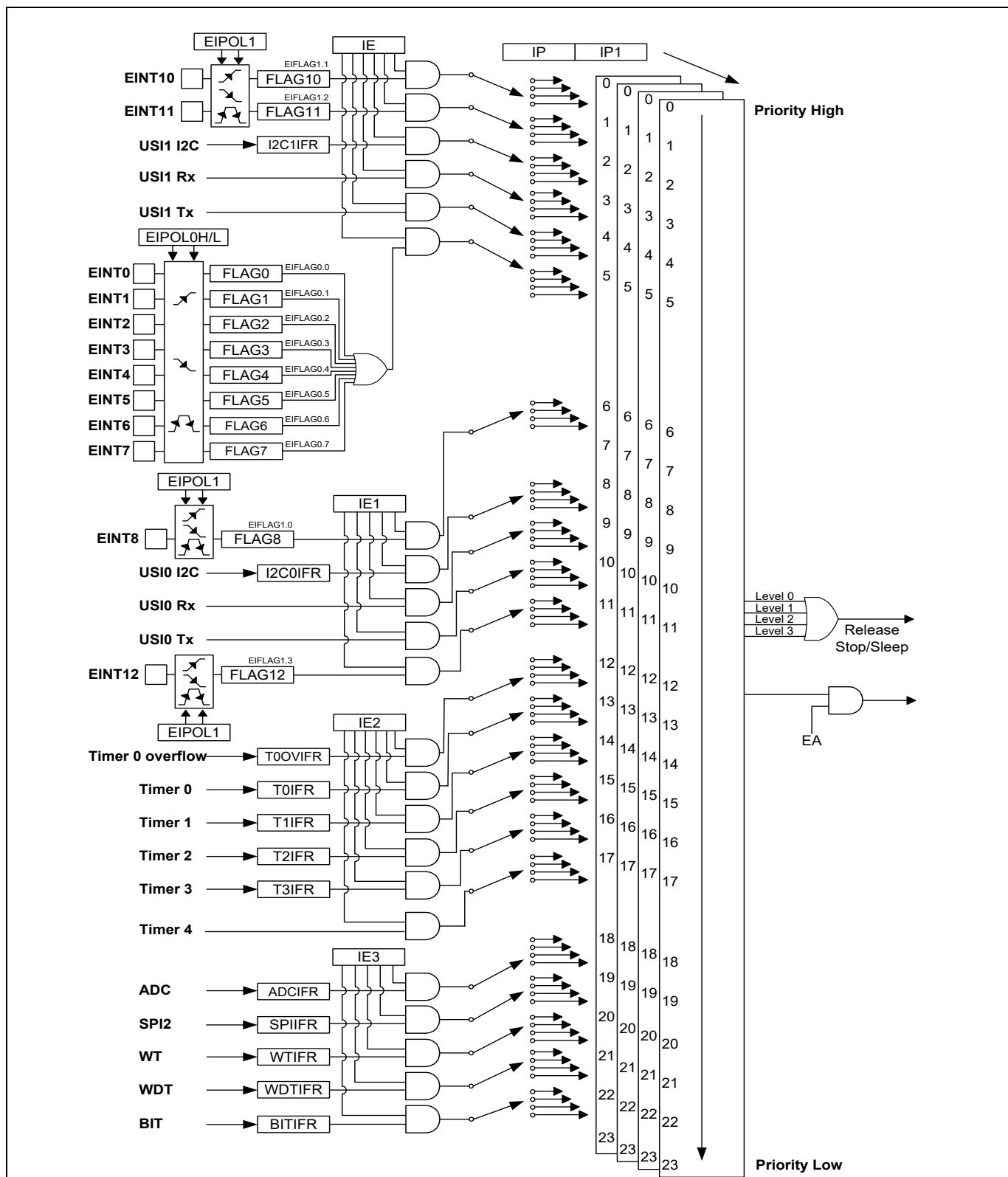


Figure 10.2 中断方框图

NOTE)

1. stop/idle 模式的复位信号可以由所有使能的中断产生，和中断优先级无关。
2. 当数据写入到 IE, IE1, IE2, IE3, IP, IP1, 和 PCON 寄存器时中断请求被延迟。

10.4 中断向量表

图表10.2中断控制器支持 24 个中断源. 当中断被响应, 执行长调用指令(LCALL)同时程序计数器跳到向量地址. 所有的中断请求都有明确的优先等级.

Interrupt Source	Symbol	Interrupt Enable Bit	Polarity	Mask	Vector Address
Hardware Reset	RESETB	-	0	Non-Maskable	0000H
External Interrupt 10	INT0	IE.0	1	Maskable	0003H
External Interrupt 11	INT1	IE.1	2	Maskable	000BH
USI1 I2C Interrupt	INT2	IE.2	3	Maskable	0013H
USI1 Rx Interrupt	INT3	IE.3	4	Maskable	001BH
USI1 Tx Interrupt	INT4	IE.4	5	Maskable	0023H
External Interrupt 0 - 7	INT5	IE.5	6	Maskable	002BH
External Interrupt 8	INT6	IE1.0	7	Maskable	0033H
-	INT7	IE1.1	8	Maskable	003BH
USI0 I2C Interrupt	INT8	IE1.2	9	Maskable	0043H
USI0 Rx Interrupt	INT9	IE1.3	10	Maskable	004BH
USI0 Tx Interrupt	INT10	IE1.4	11	Maskable	0053H
External Interrupt 12	INT11	IE1.5	12	Maskable	005BH
T0 Overflow Interrupt	INT12	IE2.0	13	Maskable	0063H
T0 Match Interrupt	INT13	IE2.1	14	Maskable	006BH
T1 Match Interrupt	INT14	IE2.2	15	Maskable	0073H
T2 Match Interrupt	INT15	IE2.3	16	Maskable	007BH
T3 Match Interrupt	INT16	IE2.4	17	Maskable	0083H
T4 Interrupt	INT17	IE2.5	18	Maskable	008BH
ADC Interrupt	INT18	IE3.0	19	Maskable	0093H
SPI 2 Interrupt	INT19	IE3.1	20	Maskable	009BH
WT Interrupt	INT20	IE3.2	21	Maskable	00A3H
WDT Interrupt	INT21	IE3.3	22	Maskable	00ABH
BIT Interrupt	INT22	IE3.4	23	Maskable	00B3H
-	INT23	IE3.5	24	Maskable	00BBH

Table 10.2 中断向量表地址

执行可屏蔽中断, EA位必须置 '1'且IEx相应的位必须置 1 来使能, 如果接收到中断请求, 相应的中断标志位被置 '1'. '1'将会一直被保存到 CPU 接受中断. 之后, 中断请求标志位将会被自动清零.

10.5 中断顺序

中断请求会一直保存，直到接受中断或者中断锁存器被复位或指令清零。接受中断通常在最后的指令周期产生，而不是当前的指令，CPU执行内部 LCALL 指令并且保存 PC 堆栈。关于中断服务程序，中断控制器将 LJMP 指令送到 CPU。完成当前指令之后，需要3~9个机器周期进入中断服务程序。中断服务被中断返回指令[RETI]终止。整个中断服务流程，如图所示。

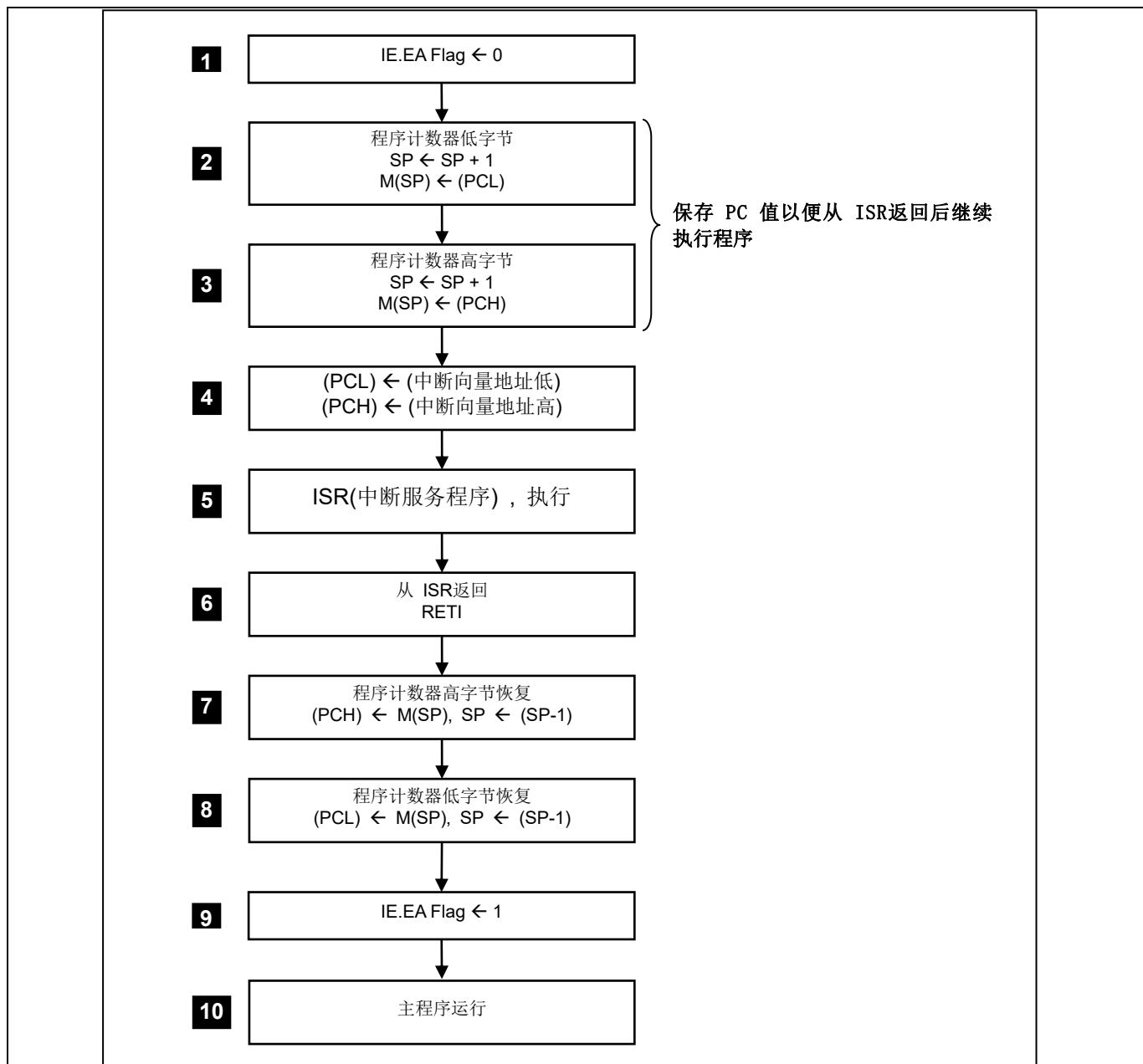


Figure 10.3 中断向量地址表

10.6 中断控制位之后的作用时序

Case a) 中断控制使能寄存器 (IE, IE1, IE2, IE3)

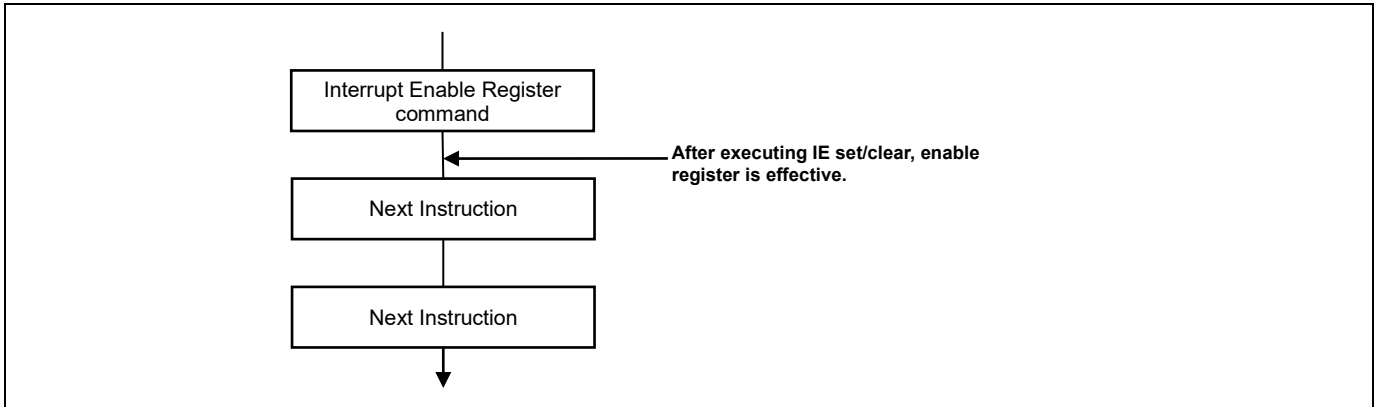


Figure 10.4 Effective 中断使能寄存器响应时序

Case b) 中断标志寄存器

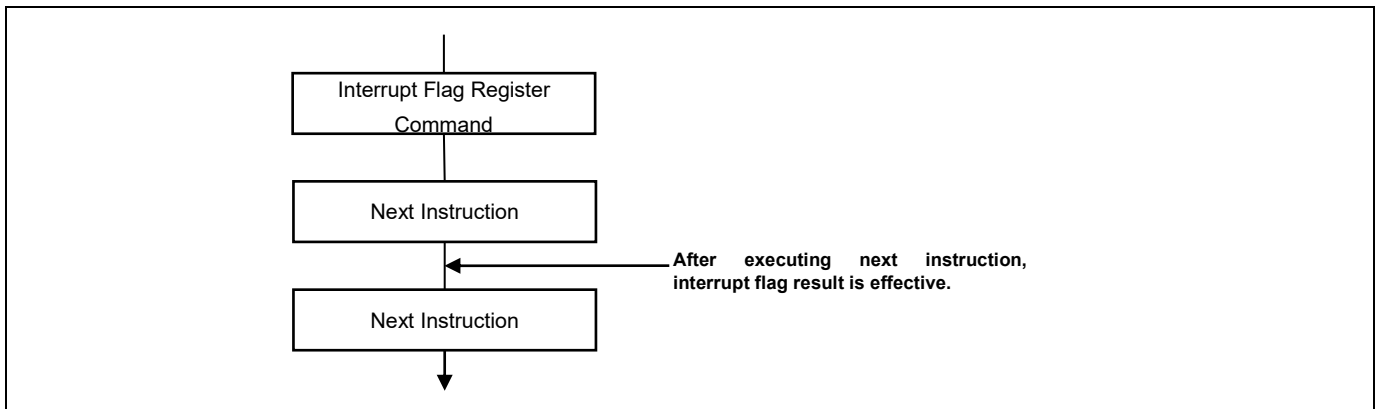


Figure 10.5 中断标志寄存器响应时序

10.7 多中断复用

如果同时收到不同优先级的中断请求，首先响应优先级较高的请求。如果同时接收到同一优先级的多个中断，则由硬件查询序列决定响应哪一中断。不过也可通过软件实现多中断请求。

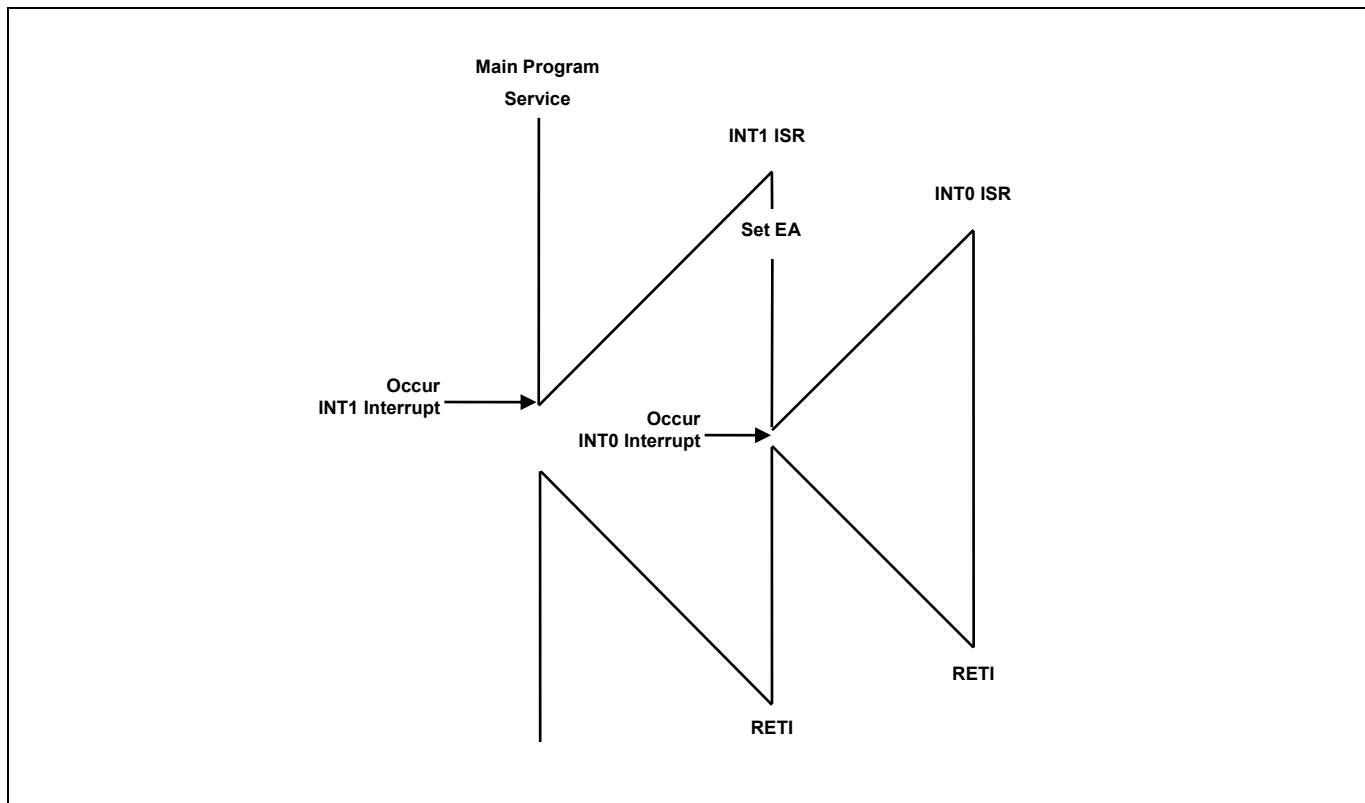


Figure 10.6 中断作用时序

图 10.6 所示为多中断复用的例子，执行INT1 时，有更高优先级的INT0发生，会立即执行INT0并且保存INT1的服务子程序。如果INT0和INT1优先级相同或低于INT1 ，那么INT0在INT1执行完成之后才能执行。

中断服务程序只能被比它更高优先级的中断，如果两个不同优先级的中断同时发生，首先执行优先级较高的中断。中断不能被低于或相同优先级的中断进行中断。如果两个优先级相同的中断同时发生，中断次序由扫描顺序决定。

10.8 中断使能接受时序

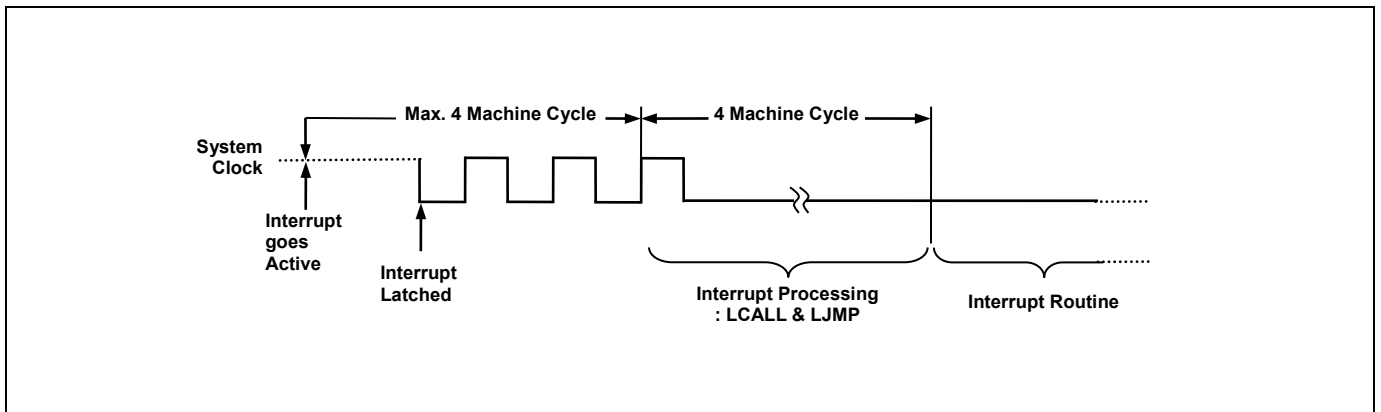


Figure 10.7 中断响应时序图

10.9 中断服务程序地址

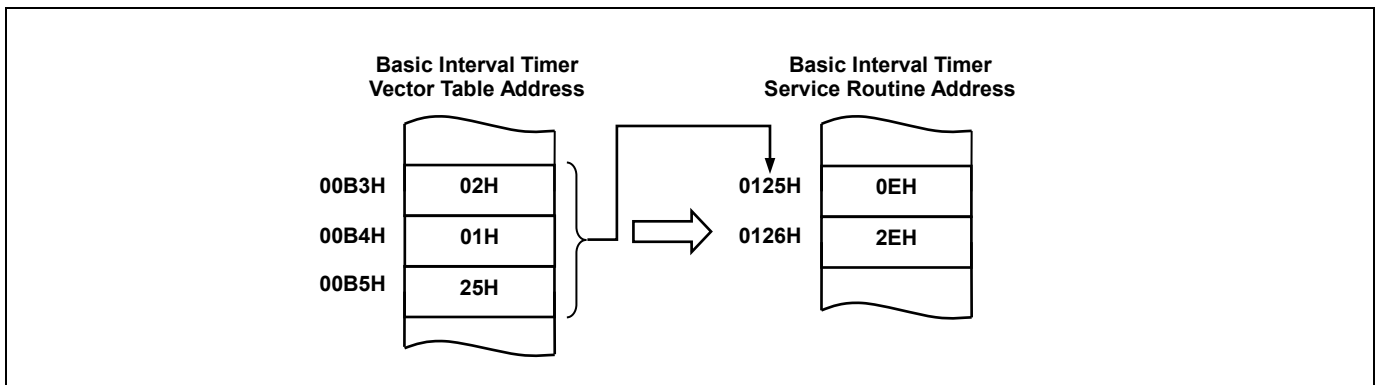


Figure 10.8 向量表地址和对应的 ISP 入口地址

10.10 通用寄存器的设置和复位

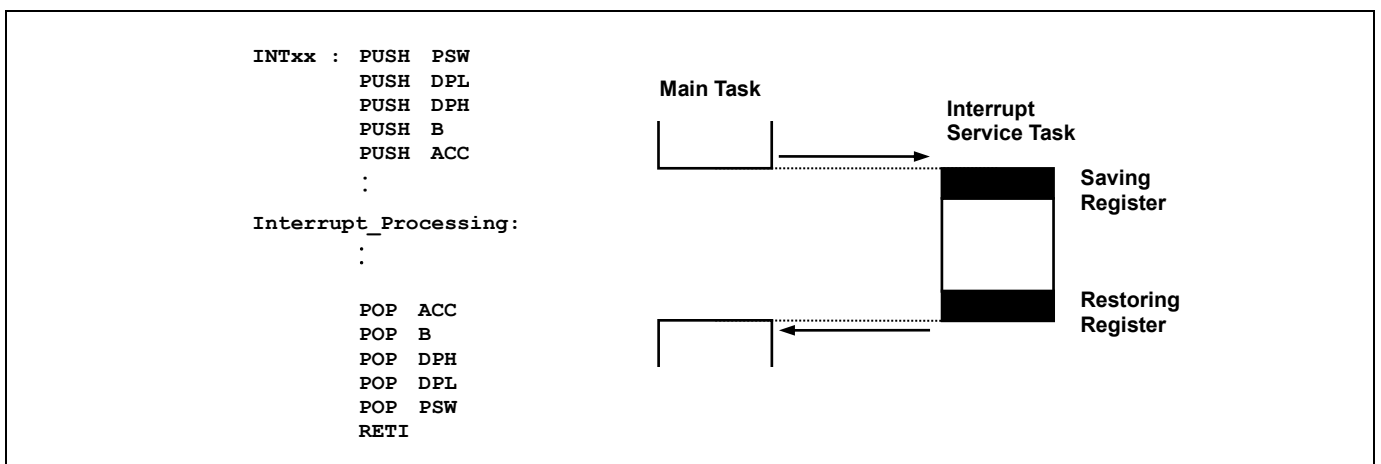


Figure 10.9 设置举例

10.11 中断时序

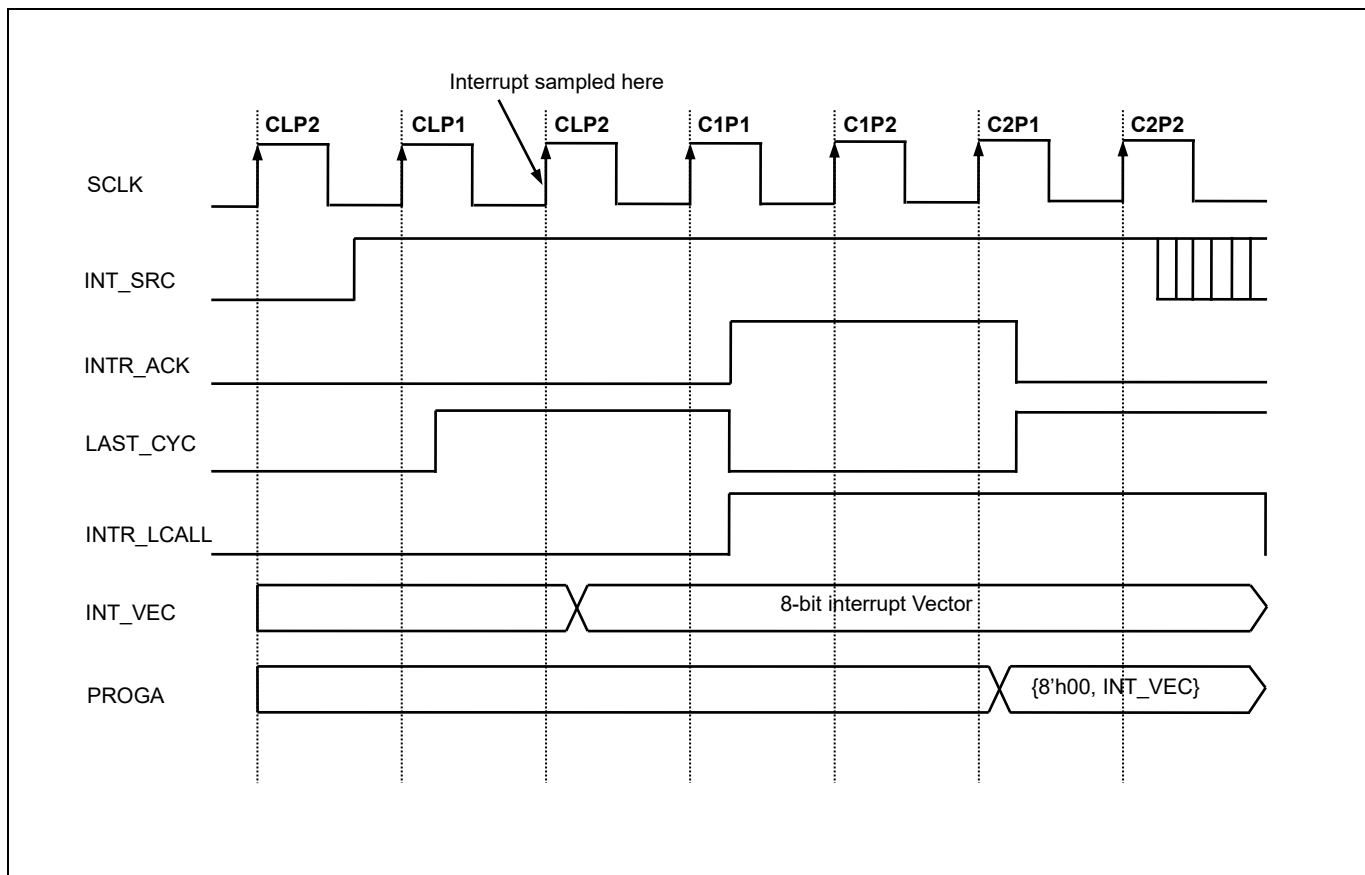


Figure 10.10 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

中断源在命令的最后一个周期取样。如果检测到中断向量(INT_VEC)的低8位被确认 is decided. M8051W 内核在命令的第一个周期产生应答, 执行长调用指令到中断服务程序..

NOTE)

1. command cycle CLPx: L=Last cycle, 1=1st cycle or 1st phase, 2=2nd cycle or 2nd phase

10.12 中断寄存器概述

10.12.1 中断使能寄存器 (IE, IE1, IE2, IE3)

中断使能寄存器由总中断位 (EA)和外围各中断位组成。共有 24 个可控制中断。

10.12.2 中断优先级寄存器 (IP, IP1)

24 个中断被分为6组每组有4个中断源。每组中断可以通过中断优先级寄存器配置为4个中断优先级. Level 3 优先级最高, level 0 优先级最低。复位后 IP 和IP1 被清除为 '00H'. 同一组中, 低序号中断有优先权。

10.12.3 外部中断标志位寄存器 (EIFLAG0, EIFLAG1)

当外部中断条件满足时外部中断标志位 0 寄存器 (EIFLAG0) 外部中断标志位 1 寄存器(EIFLAG1) 置 '1'. 中断服务程序被执行标志位清零, 或者写入0进行手动清除.

10.12.4 外部中断优先级寄存器 (EIPOL0L, EIPOL0H, EIPOL1)

外部中断高低边沿触发寄存器0 (EIPOL0H/L), 外部中断高低边沿触发寄存器1(EIPOL1) 外部中断高低边沿触发寄存器 2 (EIPOL2) 决定了上边沿, 下边沿, 或双边沿触发中断. 默认值是任何边沿没有中断.

10.12.5 中断寄存器图

Name	Address	Direction	Default	Description
IE	A8H	R/W	00H	中断使能寄存器
IE1	A9H	R/W	00H	中断使能寄存器 1
IE2	AAH	R/W	00H	中断使能寄存器 2
IE3	ABH	R/W	00H	中断使能寄存器 3
IP	B8H	R/W	00H	中断优先级寄存器
IP1	F8H	R/W	00H	中断优先级寄存器 1
EIFLAG0	C0H	R/W	00H	外部中断标志位 0 寄存器
EIPOL0L	A4H	R/W	00H	外部中断触发 0 低位寄存器
EIPOL0H	A5H	R/W	00H	外部中断触发 0 高位寄存器
EIFLAG1	A6H	R/W	00H	外部中断标志位 1 寄存器
EIPOL1	A7H	R/W	00H	外部中断触发 1 寄存器

Table 10.3 中断寄存器图

10.12.6 中断寄存器说明

中断寄存器用于控制中断功能。也有外部中断控制寄存器。中断寄存器由中断使能寄存器 (IE, IE1, IE2, IE3)组成。外部中断由外部中断标志位0 (EIFLAG0), 外部中断高低边沿触发寄存器0(EIPOL0H/L), 外部中断标志位1 (EIFLAG1), 外部中断边沿寄存器1 (EIPOL1)组成。

10.12.7 中断寄存器说明

IE (中断使能寄存器): A8H

7	6	5	4	3	2	1	0
EA	–	INT5E	INT4E	INT3E	INT2E	INT1E	INT0E
RW	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

- EA** 打开或关闭所有中断
 - 0 All Interrupt disable
 - 1 All Interrupt enable
- INT5E** 使能或禁止外部中断 0 ~ 7 (EINT0 ~ EINT7)
 - 0 禁止
 - 1 使能
- INT4E** 使能或禁止 USI1 Tx 中断
 - 0 禁止
 - 1 使能
- INT3E** 使能或禁止 USI1 Rx 中断
 - 0 禁止
 - 1 使能
- INT2E** 使能或禁止 USI1 I2C 中断
 - 0 禁止
 - 1 使能
- INT1E** 使能或禁止外部中断 11(EINT11)
 - 0 禁止
 - 1 使能
- INT0E** 使能或禁止外部中断 10 (EINT10)
 - 0 禁止
 - 1 使能

IE1 (中断使能寄存器 1): A9H

7	6	5	4	3	2	1	0
–	–	INT11E	INT10E	INT9E	INT8E	–	INT6E
–	–	RW	RW	RW	RW	–	RW

Initial value: 00H

INT11E 使能或禁止外部中断 12 (EINT12)

0 禁止

1 使能

INT10E 使能或禁止 USIO Tx 中断

0 禁止

1 使能

INT9E 使能或禁止 USIO Rx 中断

0 禁止

1 使能

INT8E 使能或禁止 USIO I2C 中断

0 禁止

1 使能

INT6E 使能或禁止外部中断 8 (EINT8)

0 禁止

1 使能

IE2 (中断使能寄存器 2): AAH

7	6	5	4	3	2	1	0
—	—	INT17E	INT16E	INT15E	INT14E	INT13E	INT12E
—	—	RW	RW	RW	RW	RW	RW

Initial value: 00H

- INT17E** 使能或禁止 Timer 4 中断
 0 禁止
 1 使能
- INT16E** 使能或禁止 Timer 3 匹配中断
 0 禁止
 1 使能
- INT15E** 使能或禁止 Timer 2 匹配中断
 0 禁止
 1 使能
- INT14E** 使能或禁止 Timer 1 匹配中断
 0 禁止
 1 使能
- INT13E** 使能或禁止 Timer 0 匹配中断
 0 禁止
 1 使能
- INT12E** 使能或禁止 Timer 0 溢出中断
 0 禁止
 1 使能

IE3 (中断使能寄存器 3): ABH

7	6	5	4	3	2	1	0
—	—	—	INT22E	INT21E	INT20E	INT19E	INT18E
—	—	—	RW	RW	RW	RW	RW

Initial value :00H

- INT22E** 使能或禁止 BIT 中断
 0 禁止
 1 使能
- INT21E** 使能或禁止 WDT 中断
 0 禁止
 1 使能
- INT20E** 使能或禁止 WT 中断
 0 禁止
 1 使能
- INT19E** 使能或禁止 SPI 2 中断
 0 禁止
 1 使能
- INT18E** 使能或禁止 ADC 中断
 0 禁止
 1 使能

IP (中断优先级寄存器): B8H

7	6	5	4	3	2	1	0
–	–	IP5	IP4	IP3	IP2	IP1	IP0
–	–	RW	RW	RW	RW	RW	RW

Initial value :00H

IP1 (中断优先级寄存器 1): F8H

7	6	5	4	3	2	1	0
–	–	IP15	IP14	IP13	IP12	IP11	IP10
–	–	RW	RW	RW	RW	RW	RW

Initial value :00H

**IP[5:0],
IP1[5:0]**

选择中断优先级组

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

EIFLAG0 (外部中断标志位 0 寄存器): C0H

7	6	5	4	3	2	1	0
FLAG7	FLAG6	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value :00H

EIFLAG0[7:0] 当外部中断 0-7产生时该位置位. 该位只有通过写入0进行清除, 所以该位需软件清零.
 0 没有外部中断 0 ~ 7 发生
 1 外部中断 0 ~ 7 发生

EIPOL0H (外部中断触发 0 高位寄存器): A5H

7	6	5	4	3	2	1	0
POL7		POL6		POL5		POL4	
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

EIPOL0H[7:0] 外部中断 (EINT7, EINT6, EINT5, EINT4) 触发边沿选择

POLn[1:0]	Description
0 0	任何边沿不发生中断
0 1	上升沿触发
1 0	下降沿触发
1 1	双边沿触发

Where n =4, 5, 6 and 7

EIPOL0L (外部中断触发 0 低位寄存器): A4H

7	6	5	4	3	2	1	0
POL3		POL2		POL1		POL0	
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

EIPOL0L[7:0] 外部中断 (EINT0, EINT1, EINT2, EINT3) 触发边沿选择

POLn[1:0]	Description
0 0	任何边沿不发生中断
0 1	上升沿触发
1 0	下降沿触发
1 1	双边沿触发

Where n =0, 1, 2 and 3

EIFLAG1 (外部中断标志位 1 寄存器): A6H

7	6	5	4	3	2	1	0
T0OVIFR	T0IFR	T3IFR	–	FLAG12	FLAG11	FLAG10	FLAG8
RW	RW	RW	–	RW	RW	RW	RW

Initial value :00H

- T0OVIFR** 当 T0 溢出中断产生时该位置位，写入0清除或通过 INT_ACK 信号自动进行清除..
- 0 没有 T0 溢出中断发生
- 1 T0 溢出中断发生
- T0IFR** 当 T0 中断产生时该位置位，写入0清除或通过 INT_ACK 信号自动进行清除..
- 0 没有 T0 中断发生
- 1 T0 中断发生
- T3IFR** 当 T3 中断产生时该位置位，写入0清除或通过 INT_ACK 信号自动进行清除..
- 0 没有 T3 中断发生
- 1 T3 中断发生
- EIFLAG1[3:0]** 当外部中断 (EINT8, EINT10-EINT12)产生时该位置位，写入0清除或通过 INT_ACK 信号自动进行清除.
- 0 没有外部中断发生
- 1 外部中断发生

EIPOL1 (外部中断触发 1 寄存器): A7H

7	6	5	4	3	2	1	0
POL12		POL11		POL10		POL8	
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

- EIPOL1[7:0]** 外部中断 (EINT8,EINT10,EINT11,EINT12) 触发边沿选择
- | POLn[1:0] | Description |
|-----------|-------------|
| 0 0 | 任何边沿不发生中断 |
| 0 1 | 上升沿触发 |
| 1 0 | 下降沿触发 |
| 1 1 | 双边沿触发 |
- Where n =8, 10, 11 and 12

11 外围硬件

11.1 时钟发生器

11.1.1 概述

图 11.1所示, 时钟发生器为CPU和外围设备提供基本时钟脉冲, 它包括主/副时钟振荡器。主/副时钟可以通过在XIN/SXIN 和 XOUT/SXOUT 引脚上分别连接晶体获得. 也可以通过外部时钟获得, 如果这样的话, 需要输入时钟信号到 XIN /SXIN 并且打开 XOUT/SXOUT脚。默认的系统时钟是1MHz INT-RC默认8分频. 为了内部时钟的稳定, POR 使用 1 MHz INT-RC.

- 调校内部 RC Oscillator (16MHz)
 - INT-RC OSC/1 (16MHz)
 - INT-RC OSC/2 (8MHz)
 - INT-RC OSC/4 (4MHz)
 - INT-RC OSC/8 (2MHz)
 - INT-RC OSC/16 (1MHz,默认)
 - INT-RC OSC/32 (0.5MHz)
- 主时钟 (0.4~8.5MHz)
- 副时钟 (32.768kHz)
- 内部 WDTRC 振荡器 (5kHz)

11.1.2 方框图

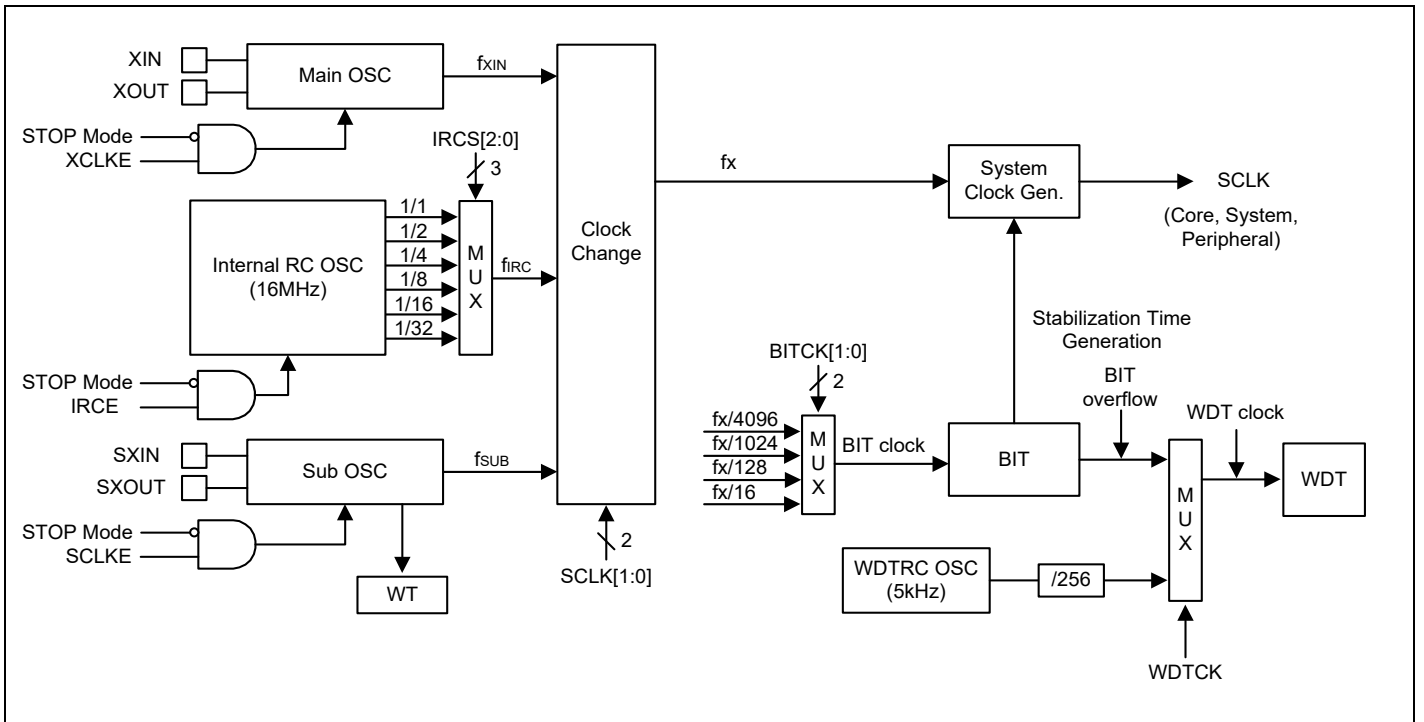


Figure 11.1 时钟发生器框图

11.1.3 寄存器图

Name	Address	Direction	Default	Description
SCCR	8AH	R/W	00H	系统时钟控制寄存器
OSCCR	C8H	R/W	20H	振荡器控制寄存器
XTFLSR	1038H	R/W	00H	X-tal 滤波器选择寄存器

Table 11.1 时钟发生器寄存器图

11.1.4 时钟发生器寄存器说明

时钟发生寄存器使用时钟控制系统操作。时钟发生器包含系统时钟控制寄存器和振荡控制寄存器组成。

11.1.5 时钟发生器寄存器说明

SCCR (系统时钟控制寄存器): 8AH

7	6	5	4	3	2	1	0
-	-	-	-	-	-	SCLK1	SCLK0
-	-	-	-	-	-	R/W	R/W

Initial value: 00H

SCLK [1:0] 选择系统时钟

SCLK1	SCLK0	Description
0	0	外部副时钟作为系统时钟 (f _{SUB})
0	1	外部主时钟作为系统时钟 (f _{XIN})
1	0	外部副时钟作为系统时钟 (f _{SUB})
1	1	Not used

OSCCR (振荡器控制寄存器): C8H

7	6	5	4	3	2	1	0
–	–	IRCS2	IRCS1	IRCS0	IRCE	XCLKE	SCLKE
–	–	RW	RW	RW	RW	RW	RW

Initial value: 08H

IRCS[2:0]	内部 RC 分频选择			
	IRCS2	IRCS1	IRCS0	Description
	0	0	0	INT-RC/32 (0.5MHz)
	0	0	1	INT-RC/16 (1MHz)
	0	1	0	INT-RC/8 (2MHz)
	0	1	1	INT-RC/4 (4MHz)
	1	0	0	INT-RC/2 (8MHz)
	1	0	1	INT-RC/1 (16MHz)
	Other values			Not used
IRCE	控制内部 RC 振荡器			
	0	使能内部 RC OSC		
	1	禁止内部 RC OSC		
XCLKE	控制外部主时钟运行			
	0	关闭外部主时钟 X-TAL		
	1	使能外部主时钟 X-TAL		
SCLKE	控制外部副时钟运行			
	0	关闭外部副时钟 SX-TAL		
	1	使能外部副时钟 SX-TAL		

NOTE)

1. 有噪音环境中建议使用 IRC 8MHz 或更低的频率

XTFLSR (X-tal 滤波选择寄存器): 1038H (ESFR)

7	6	5	4	3	2	1	0
WTP4	WTP3	WTP2	WTP1	WTP0	-	-	XRNS
W	W	W	W	W	-	-	R/W

Initial value: 00H

WTP[4:0] 写识别位. 这些位会在XTFLSR写入后, 立刻清除为“00000b”. 读取为0x00.
 10101b Write 0x15 to these bits with valid XRNS[2:0]
 Other values Write is ignored.

XRNS 外部主晶振范围选择. 当选择fXIN为系统时钟源时, 这些位有效.

XRNS	Description
0	4.2MHz < x-tal ≤ 8.5MHz
1	x-tal ≤ 4.2MHz

NOTE)

1. 当系统时钟是 IRC, 应更改主晶振范围 (XRNS).

11.2 BIT

11.2.1 概述

MC96F6432A有一个独立运行不能停止的 8 位基本间隔定时器. 如图11.2方框图所示. 另外它还为看门狗计数器提供扫描基线, 提供一个基本间隔定时器中断 (BITIFR).

MC96F6432A基本间隔定时器的特性如下:

- 上电过程中, BIT 提供稳定的时钟脉冲时间
- 退出Stop 模式, BIT 提供稳定的时钟脉冲时间
- 作为定时器使用, 产生定时器中断

11.2.2 方框图

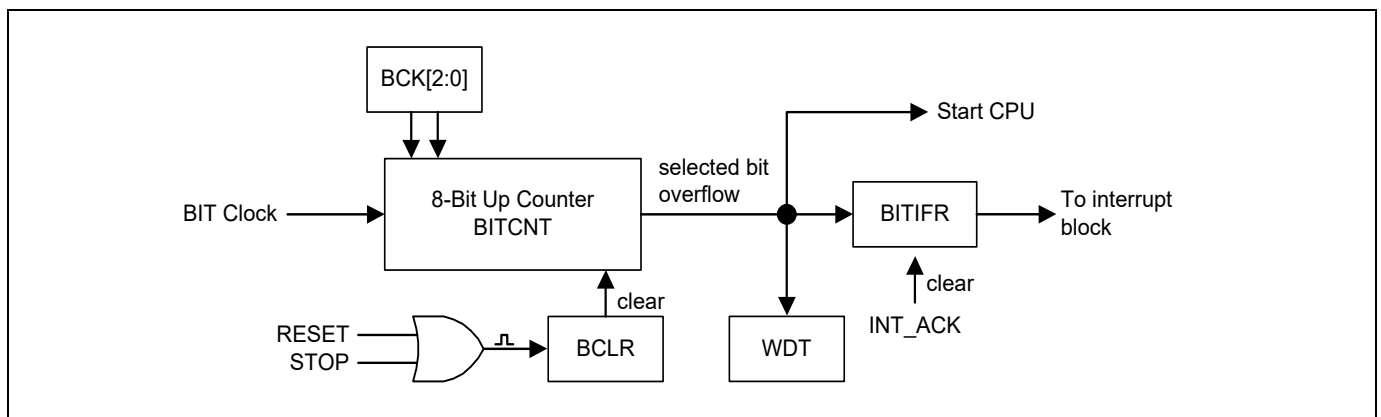


Figure 11.2 BIT方框图

11.2.3 寄存器图

Name	Address	Direction	Default	Description
BITCNT	8CH	R	00H	Basic Interval Timer 计数寄存器
BITCR	8BH	R/W	01H	Basic Interval Timer 控制寄存器

Table 11.2 BIT 寄存器

11.2.4 BIT 寄存器说明

基本间隔寄存器由 BITCNT 和BITCR组成. 如果 BCLR 位设置为 ‘1’, BITCNT 变为 ‘0’ 并开始计数。一个机器周期后, BCLR 位自动被清零.

11.2.5 BIT寄存器

BITCNT (BIT 计数寄存器): 8CH

7	6	5	4	3	2	1	0
BITCNT7	BITCNT6	BITCNT5	BITCNT4	BITCNT3	BITCNT2	BITCNT1	BITCNT0
R	R	R	R	R	R	R	R

Initial value: 00H

BITCNT[7:0] BIT 计数

BITCR (BIT 控制寄存器): 8BH

7	6	5	4	3	2	1	0
BITIFR	BITCK1	BITCK0	-	BCLR	BCK2	BCK1	BCK0
RW	RW	RW	-	RW	RW	RW	RW

Initial value: 01H

BITIFR 当 BIT 中断产生时该位置 ‘1’. 写入0进行清零或者由INT_ACK 信号自动清零.

- 0 没有中断产生BIT
- 1 有中断产生BIT

BITCK[1:0] 选择 BIT 时钟

BITCK1	BITCK0	Description
0	0	fx/4096
0	1	fx/1024
1	0	fx/128
1	1	fx/16

BCLR 如果该位写入 ‘1’, BIT 计数器清零

- 0 Free Running
- 1 Clear Counter

BCK[2:0] 选择 BIT 溢出周期

BCK2	BCK1	BCK0	Description (fx=1MHz, fx/4096)
0	0	0	Bit 0 overflow (BIT Clock * 2)
0	0	1	Bit 1 overflow (BIT Clock * 4) (default)
0	1	0	Bit 2 overflow (BIT Clock * 8)
0	1	1	Bit 3 overflow (BIT Clock * 16)
1	0	0	Bit 4 overflow (BIT Clock * 32)
1	0	1	Bit 5 overflow (BIT Clock * 64)
1	1	0	Bit 6 overflow (BIT Clock * 128)
1	1	1	Bit 7 overflow (BIT Clock * 256)

11.3 WDT

11.3.1 概述

看门狗可以迅速的发现CPU 因干扰或类似的问题引起的死循环, 并将其恢复到正常状态. 该故障检测信号也可以作为 CPU 复位或者中断复位. 不作为故障检测使用时, 可用做固定时间产生中断的定时器. 也可以通过设置WDTCCR[6]位作为独立运行的8位定时器模式 (WDTRSON='0') 或者看门狗定时器模式 (WDTRSON='1'). 如果 WDTCCR[5] 被写入 '1', WDT 计数器的值被清除并开始计数. 一个机器周期后, 该位被自动清零. 它由8位二进制计数器和看门狗数据寄存器组成. 当8位二进制计数器的值等于8 位 WDTCNT值时, 产生中断请求标志位. 可以根据WDTRSON位的设置用于看门狗中断或者复位.

它的时钟输入源来自于 BIT 溢出. 内部看门狗中断由BIT溢出周期和WDTDR 设定的值来决定. 计算公式如下

$$\text{WDT Interrupt Interval} = (\text{BIT Interrupt Interval}) \times (\text{WDTDR Value} + 1)$$

11.3.2 WDT 中断时序波形图

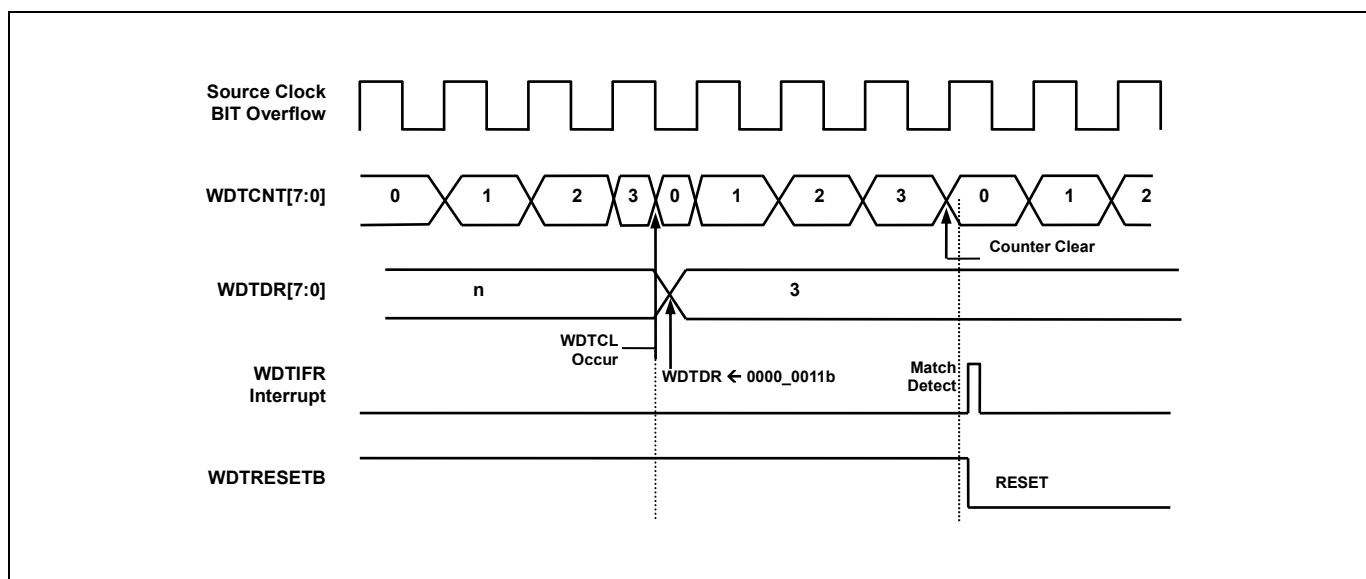


Figure 11.3 看门狗定时器中断时序波形

11.3.3 方框图

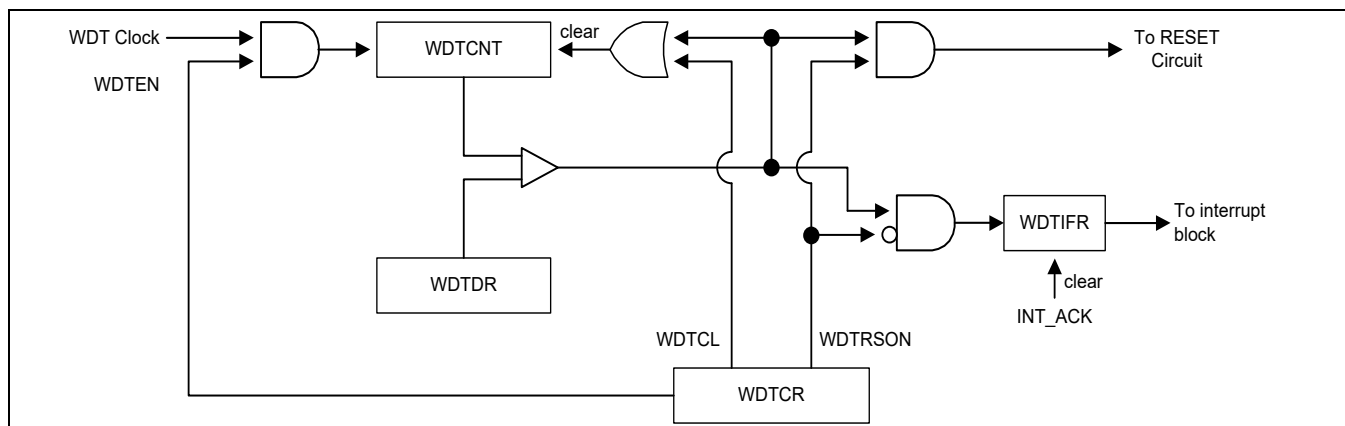


Figure 11.4 WDT方框图

11.3.4 寄存器图

Name	Address	Direction	Default	Description
WDCNT	8EH	R	00H	Watch Dog Timer 计数寄存器
WDTDR	8EH	W	FFH	Watch Dog Timer 数据寄存器
WDTCLR	8DH	R/W	00H	Watch Dog Timer 控制寄存器

Table 11.3 WDT寄存器图

11.3.5 WDT寄存器说明

看门狗定时器寄存器主要由WDCNT, WDTDR 和WDTCLR组成.

11.3.6 WDT寄存器

WDTCNT (WDT 计数寄存器: 只读): 8EH

7	6	5	4	3	2	1	0
WDTCNT7	WDTCNT6	WDTCNT5	WDTCNT4	WDTCNT3	WDTCNT2	WDTCNT1	WDTCNT0
R	R	R	R	R	R	R	R

Initial value: 00H

WDTCNT[7:0] WDT 计数

WDTDR (WDT 数据寄存器: 只写): 8EH

7	6	5	4	3	2	1	0
WDTDR7	WDTDR6	WDTDR5	WDTDR4	WDTDR3	WDTDR2	WDTDR1	WDTDR0
W	W	W	W	W	W	W	W

Initial value: FFH

WDTDR[7:0] 设置周期

WDT中断=(BIT 中断) x(WDTDR 值+1)

NOTE)

1. 不要写 0.

WDTCR (WDT控制寄存器): 8DH

7	6	5	4	3	2	1	0
WDTEN	WDRSON	WDTCL	–	–	–	WDTCK	WDTIFR
RW	RW	RW	–	–	–	RW	RW

Initial value: 00H

WDTEN WDT控制操作

0 禁止

1 使能

WDRSON WDT RESET 控制

0 8-bit 独立运行定时器

1 WDT RESET 打开

WDTCL 清除 WDT 计数器

0 自由运行

1 清除 WDT 计数器 (一个周期后自动清除)

WDTCK 控制 WDT 时钟

0 BIT 溢出为 WDT 提供时钟 (WDTRC 关闭)

1 WDTRC 为 WDT 提供时钟 (WDTRC 使能)

WDTIFR 当WDT中断产生时置 '1'. 写入0清除或者被INT_ACK 信号自动清除

0 WDT中断不产生

1 WDT中断产生

11.4 WT

11.4.1 概述

WT有 RTC (Real Time Clock) 操作功能. 通常用于 RTC 操作. WT的内部结构由时钟选择电路, 定时器计数电路, 输出选择电路和控制寄存器组成. WTCR寄存器用来控制输入时钟, 输出间隔, 和设置WTEN 位. 可以同时执行也可有单独执行. 清除WTEN位来停止或复位WT. 即使是在 STOP 模式, 副时钟仍存在则WT可以进行运行. 为了提高分辨率 WT计数器可以组成21位计数器, 包括低14位和高7位二进制计数器. WTR寄存器在写入时可以用来清除和设置WT的间隔值, 在读时序时可以读7位计数器的值.

WT为LCD驱动提供时钟频率 (f_{LCD}). 所以, 如果WT禁止的话, LCD驱动是不能工作的.

11.4.2 方框图

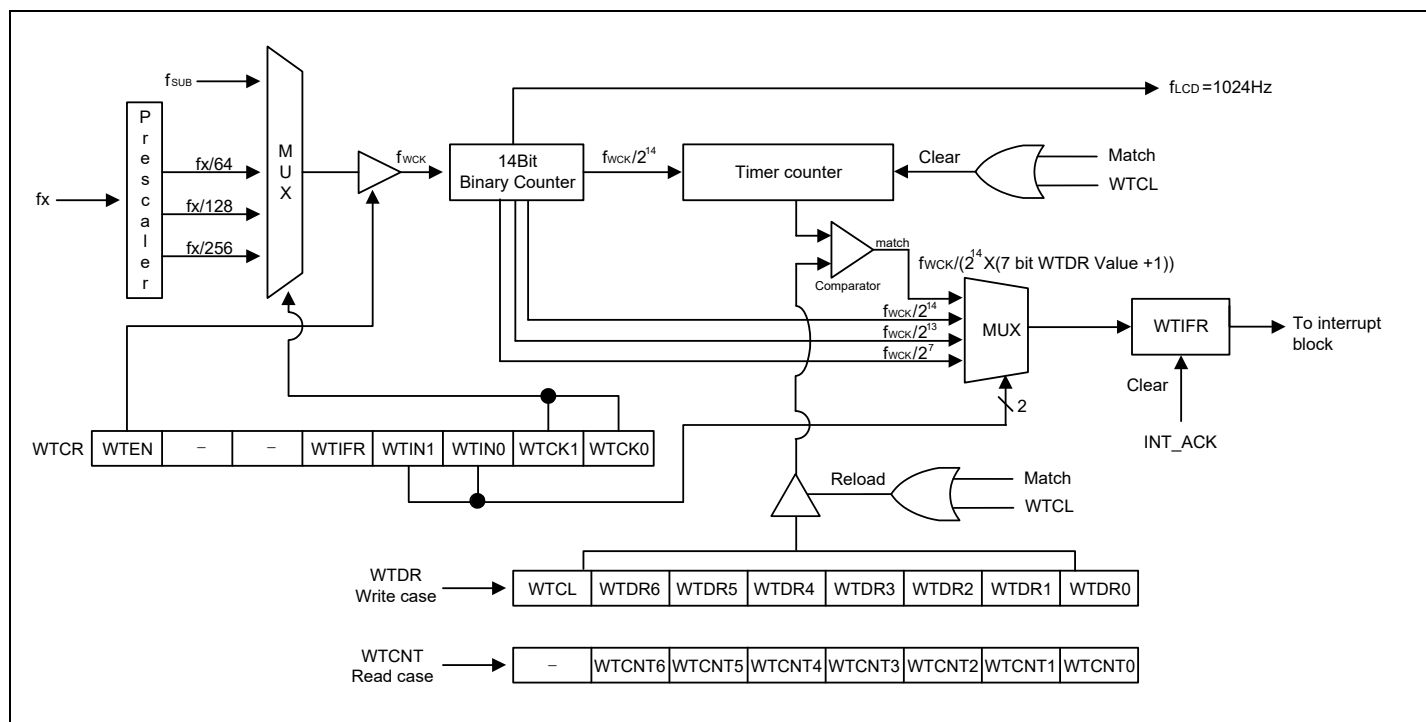


Figure 11.5 WT 框图

11.4.3 寄存器图

Name	Address	Direction	Default	Description
WTCNT	89H	R	00H	Watch Timer 计数寄存器
WTDR	89H	W	7FH	Watch Timer 数据寄存器
WTCR	96H	R/W	00H	Watch Timer 控制寄存器

Table 11.4 WT寄存器

11.4.4 WT说明

WT寄存器由WTCNT，WTDR，WTCR组成。WTCR是6位可读、可写寄存器，WTCR可以控制时钟(WTCK[1:0])，中断间隔(WTIN[1:0])，和开关(WTEN)。WT有一个中断标志位(WTIFR)。

11.4.5 WT寄存器

WTCNT (WT 计数器寄存器 : 只读): 89H

7	6	5	4	3	2	1	0
-	WTCNT6	WTCNT5	WTCNT4	WTCNT3	WTCNT2	WTCNT1	WTCNT0
-	R	R	R	R	R	R	R

Initial value: 00H

WTCNT[6:0] WT 计数

WTDR (WT 数据寄存器 : 只写): 89H

7	6	5	4	3	2	1	0
WTCL	WTDR6	WTDR5	WTDR4	WTDR3	WTDR2	WTDR1	WTDR0
RW	W	W	W	W	W	W	W

Initial value: 7FH

WTCL 清除 WT 计数器
 0 Free Run
 1 清除 WT 计数 (一个周期后自动清除)

WTDR[6:0] S设置 WT 周期
 WT 中断间隔= $fwck/(2^{14} \times (7\text{bit WTDR Value}+1))$

NOTE)

1. 不要写入“0”。

WTCR (WT控制寄存器): 96H

7	6	5	4	3	2	1	0
WTEN	-	-	WTIFR	WTIN1	WTIN0	WTCK1	WTCK0
RW	-	-	RW	RW	RW	RW	RW

Initial value: 00H

WTEN	WT控制操作		
	0	禁止	
	1	使能	
WTIFR	当 WT 中断产生时置 '1'. 写入0进行清除或通过 INT_ACK 信号自动清零.		
	0	WT 中断不产生	
	1	WT 中断产生	
WTIN[1:0]	决定中断间隔		
	WTIN1	WTIN0	Description
	0	0	$f_{wck}/2^7$
	0	1	$f_{wck}/2^{13}$
	1	0	$f_{wck}/2^{14}$
	1	1	$f_{wck}/(2^{14} \times (7\text{bit WTDR Value}+1))$
WTCK[1:0]	选择时钟		
	WTCK1	WTCK0	Description
	0	0	f_{SUB}
	0	1	$f_x/256$
	1	0	$f_x/128$
	1	1	$f_x/64$

NOTE)

1. f_x —系统时钟 (Where $f_x= 4.19\text{MHz}$)
2. f_{SUB} — 副时钟频率(32.768kHz)
3. f_{wck} —所选 WT 时钟
4. f_{LCD} – LCD 频率(Where $f_x= 4.19\text{MHz}$, $WTCK[1:0]='10'$; $f_{LCD}= 1024\text{Hz}$)

11.5 定时器 0

11.5.1 概述

8位定时器0由多路转换器，定时器0 计数器寄存器，数据寄存器，捕捉数据寄存器和定时器0控制寄存器组成 (T0CNT, T0DR, T0CDR, T0CR)

有3个操作模式:

- 8-bit 定时器/计数器 模式
- 8-bit PWM 输出模式
- 8-bit 捕捉模式

定时器/计数器 0 可以通过内部时钟或外部时钟控制(EC0). 时钟选择通过T0CK[2:0]位来控制.

- TIMER0 时钟源: $f_x/(2, 4, 8, 32, 128, 512, 2048)$, EC0

捕捉模式中，通过 EINT10 将数据捕捉到捕捉数据寄存器中 (T0CDR). 定时器/计数器模式时，当计算值等于 T0DR时，T0O 口打开。PWM模式下可以通过PWM0O口输出PWM波形.

T0EN	T0MS[1:0]	T0CK[2:0]	Timer 0
1	00	XXX	8-bit定时器/计数器 模式
1	01	XXX	8-bit PWM 模式
1	1X	XXX	8-bit 捕捉模式

Table 11.5 Timer 0 模式

11.5.2 8-bit 定时器/计数器 模式

图11.6所示为 8-bit 定时器/计数器 模式控制选择

8位定时器具有计数器和数据寄存器，计数寄存器通过内部或外部时钟输入来累加。Timer 0 可用的时钟预分频状态有2, 4, 8, 32, 128, 512 和 2048 (T0CK[2:0]). 当T0CNT 和T0DR 值相同时，会产生匹配信号同时产生定时器中断0. T0CNT 值自动被匹配信号清除. 也可通过软件清除 (T0CC).

外部时钟 (EC0) 在上升沿开始累加，如果时钟通过T0CK[2:0]选择, EC0 口应该设置为输入状态 (P52IO 位)

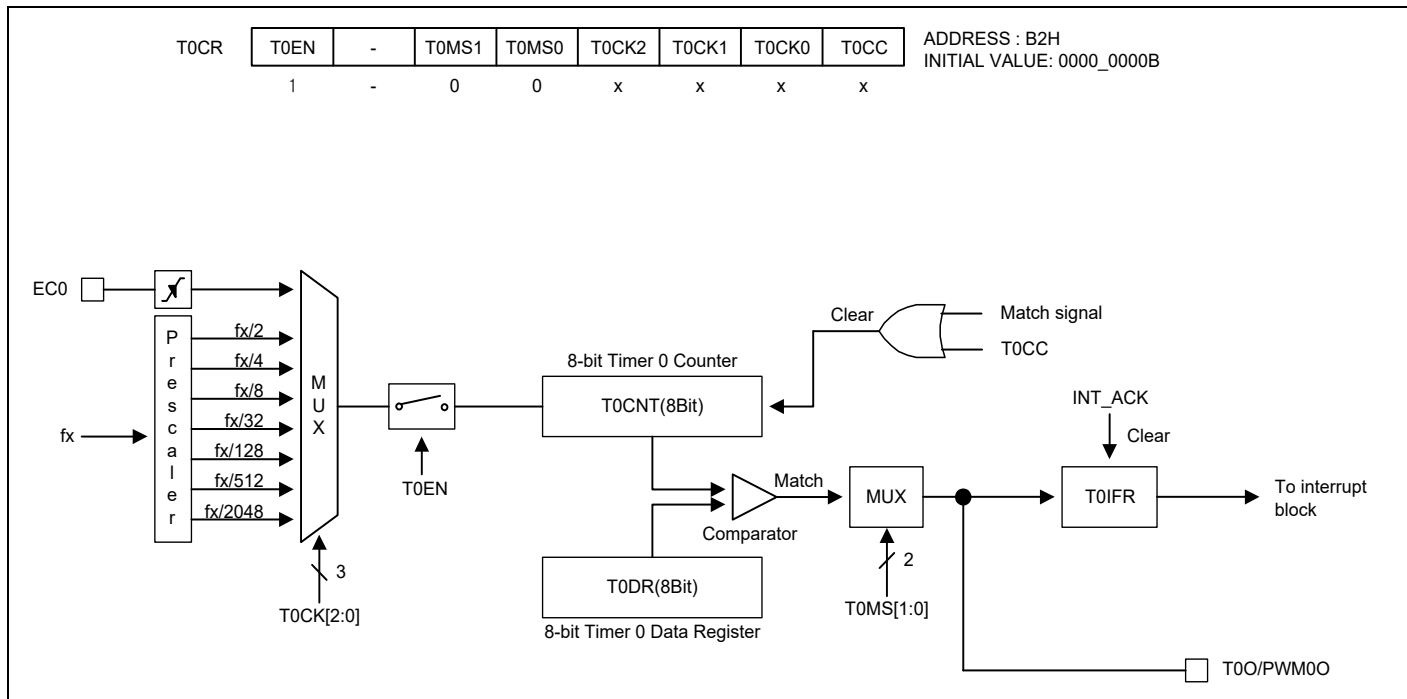


Figure 11.6 定时器0 8-bit 定时器/计数器模式

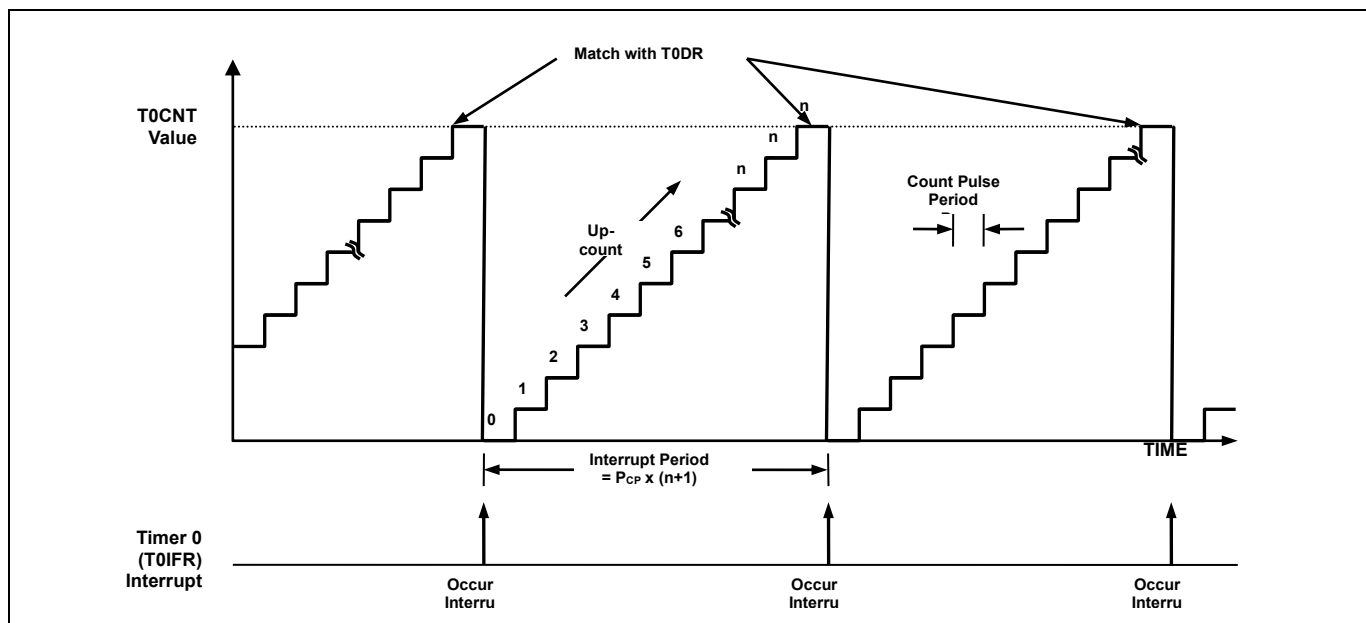


Figure 11.7 8-bit 定时器/计数器 0 举例

11.5.3 8-bit PWM 模式

定时器0具有高速PWM功能。在PWM模式中，T0O/PWM0O 引脚输出8位分辨率的PWM。该脚需要通过设置P5FSR [4:3]位，将其配置为PWM输出口。在8位定时器/计数器模式时，当计数值与T0DR值相同时会产生一个匹配信号。在定时器0模式下，当T0CNT 和T0DR 相同时，产生匹配信号同时产生定时器中断。在PWM模式时，会一直运行直到“FFH”时溢出，然后继续从“00H”开始。当计数溢出时定时器0溢出中断产生。T0CNT 的值通过软件T0CC 位清除。

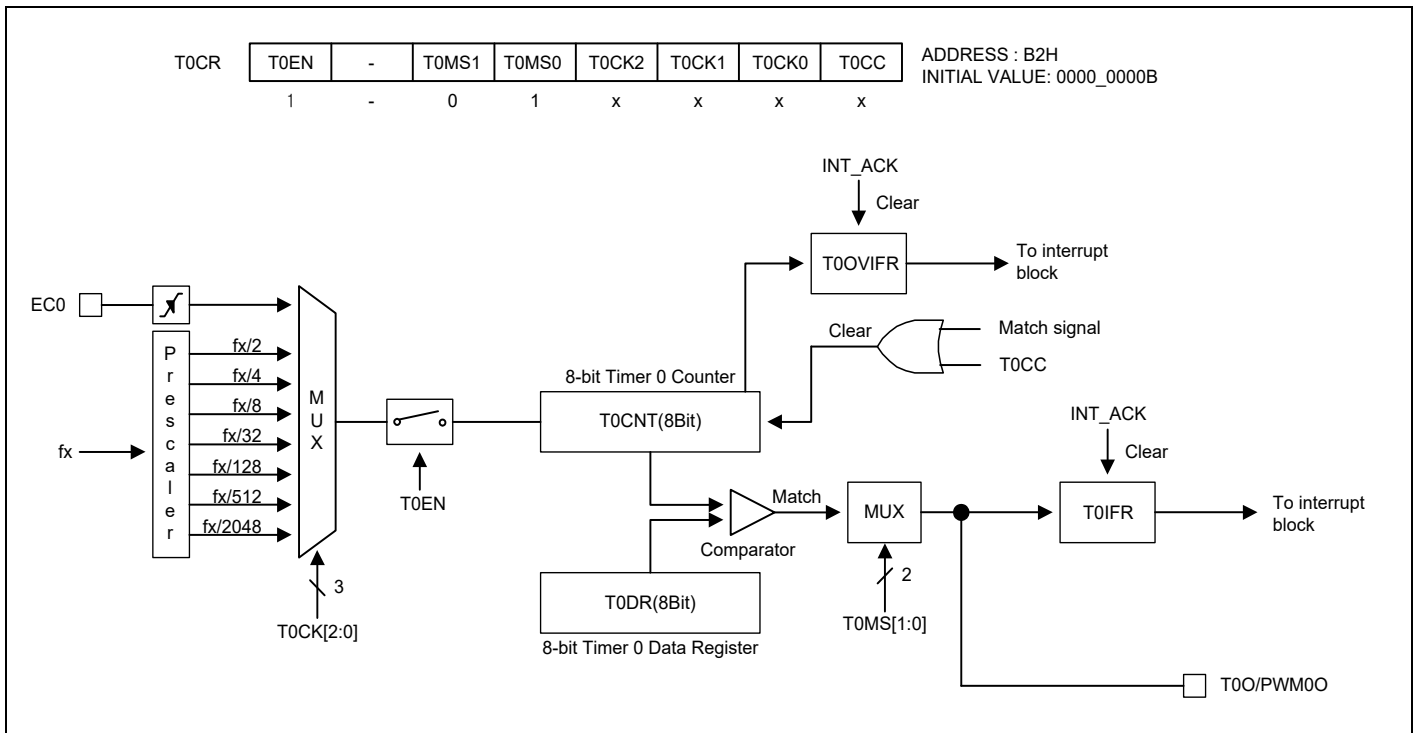


Figure 11.8 定时器0 8-bit PWM 模式

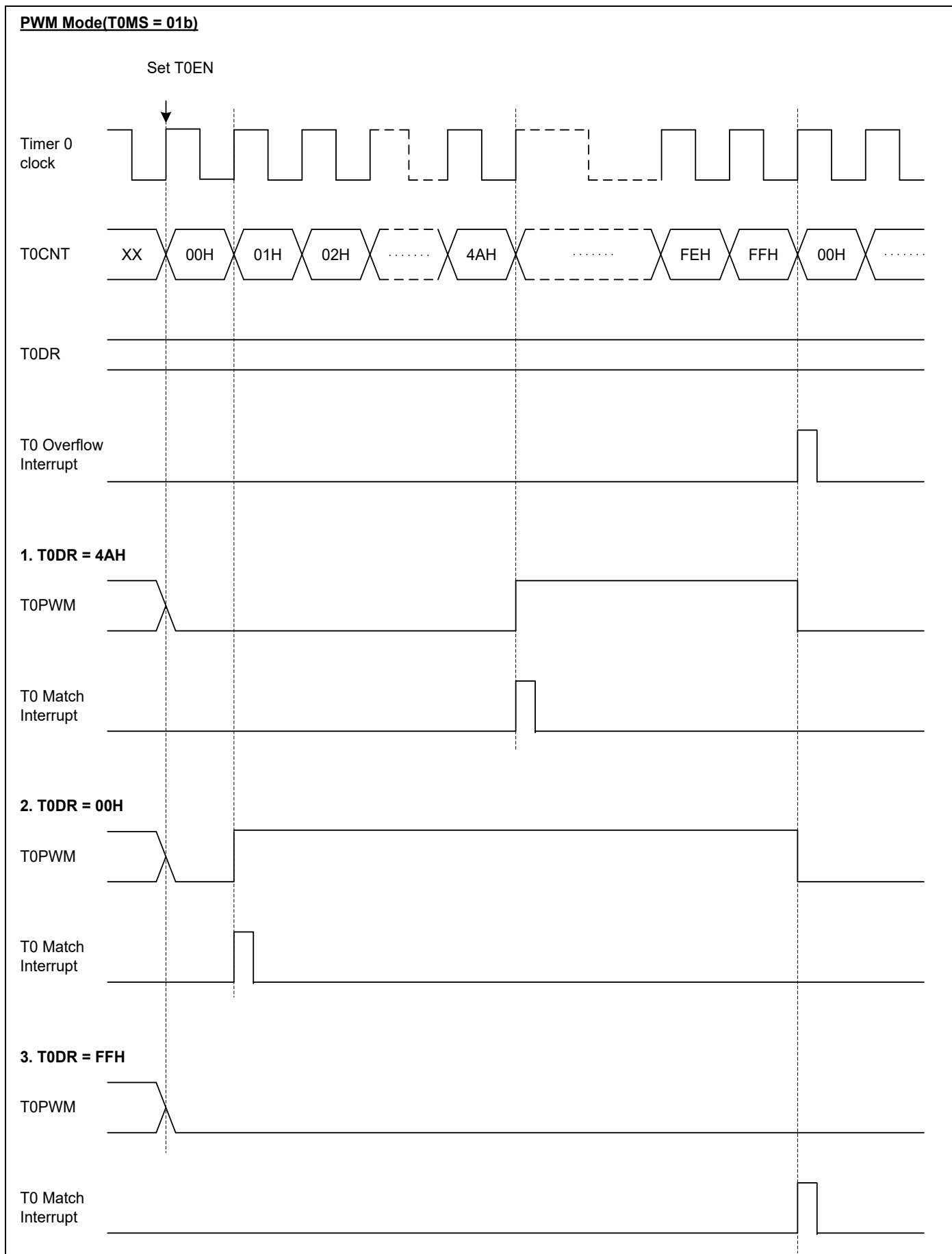


Figure 11.9 定时器0 在PWM 模式时输出波形

11.5.4 8-bit 捕捉模式

设置 T0MS[1:0] 位为 '1x' 把定时器0设为捕捉模式. 时钟可以使用内部和外部时钟。当T0CNT 的值和T0DR相同时，它基本上和定时器/计数器模式以及中断的发生具有相同的功能。 T0CNT 值被匹配信号自动清零或者通过软件(T0CC)清零..

当捕捉的脉冲信号宽度大于定时器周期最大值时，捕捉模式下的定时器中断是非常有用的

捕捉的结果装载T0CDR. 在定时器0捕捉模式下，(T0O) 波形输出是无法使用的. 根据 EIPOL1 寄存器设置，外部中断触发功能EINT10可以选择. 当然， EINT10 引脚必须设置为输入口

T0CDR 和 T0DR 使用相同的地址. 在捕捉模式时，读取时读T0CDR 而不是T0DR，但是写入时是T0DR.

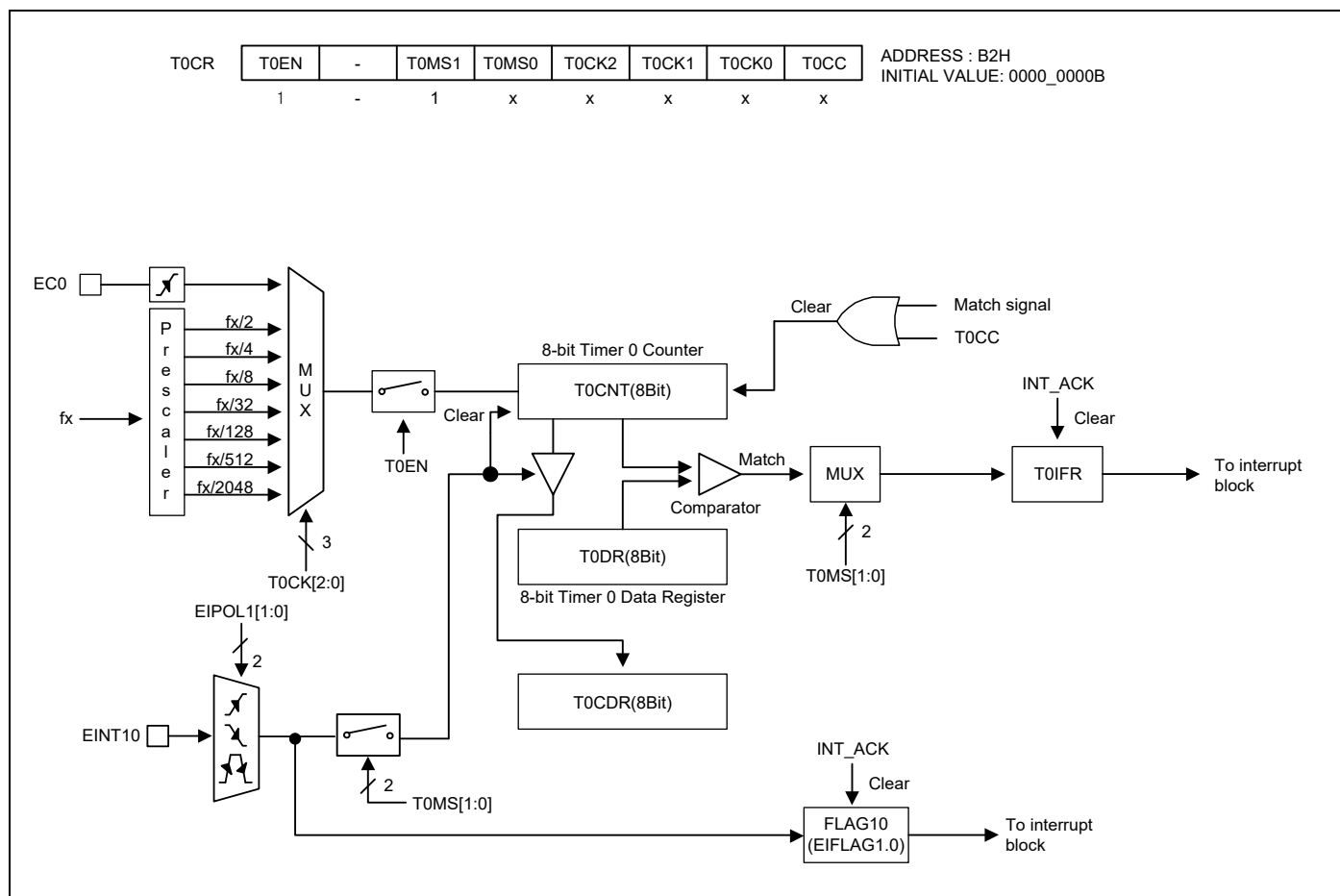


Figure 11.10 定时器0 8-bit 捕捉模式

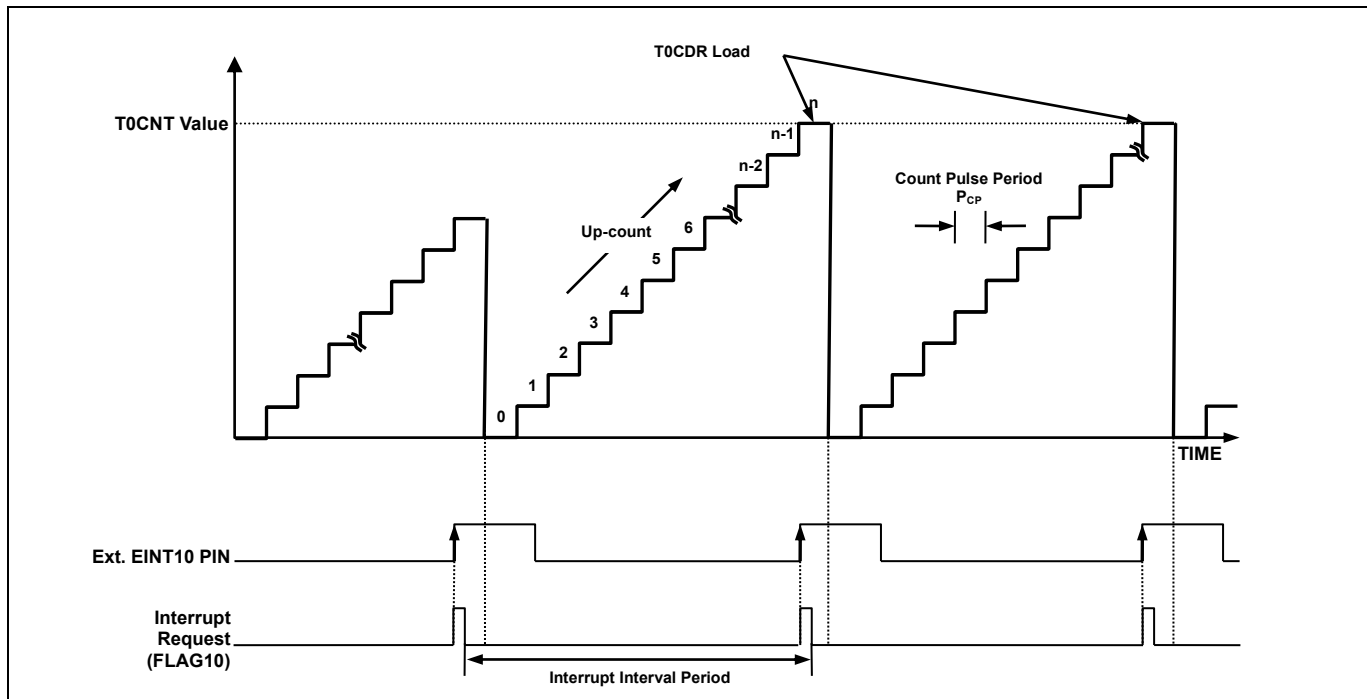


Figure 11.11 定时器0 输入捕捉模式

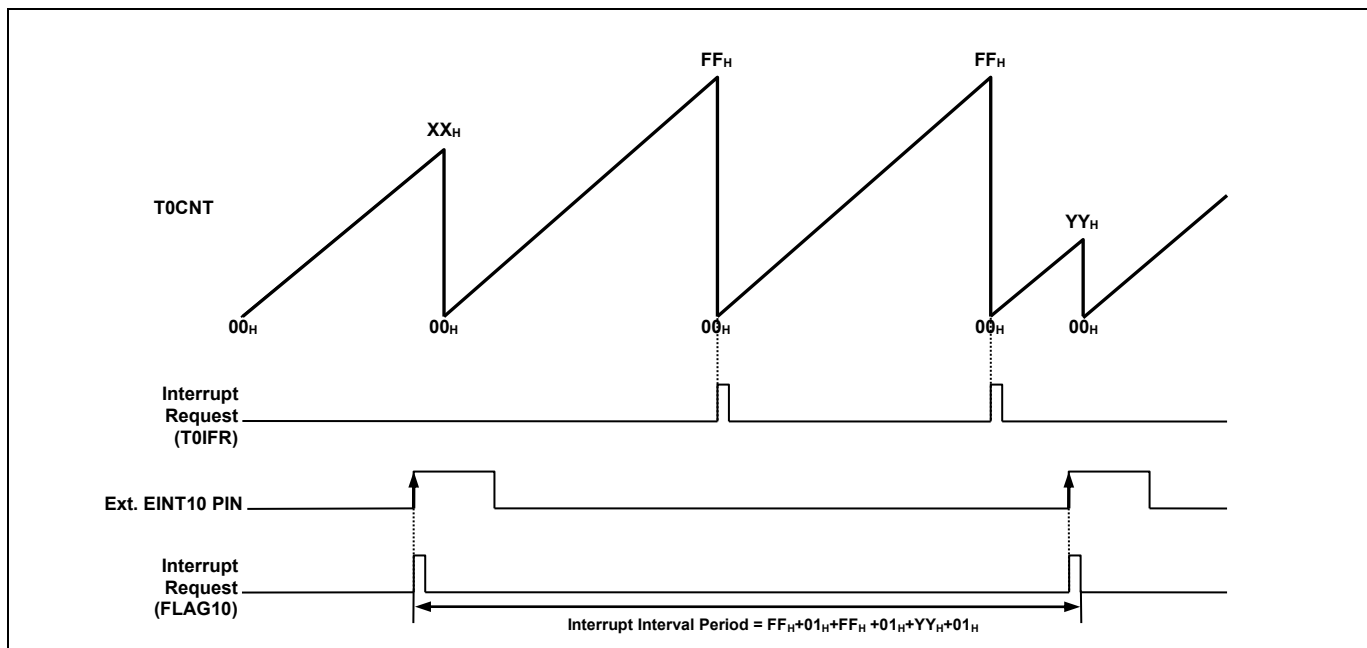


Figure 11.12 捕捉模式中的定时器溢出

11.5.5 方框图

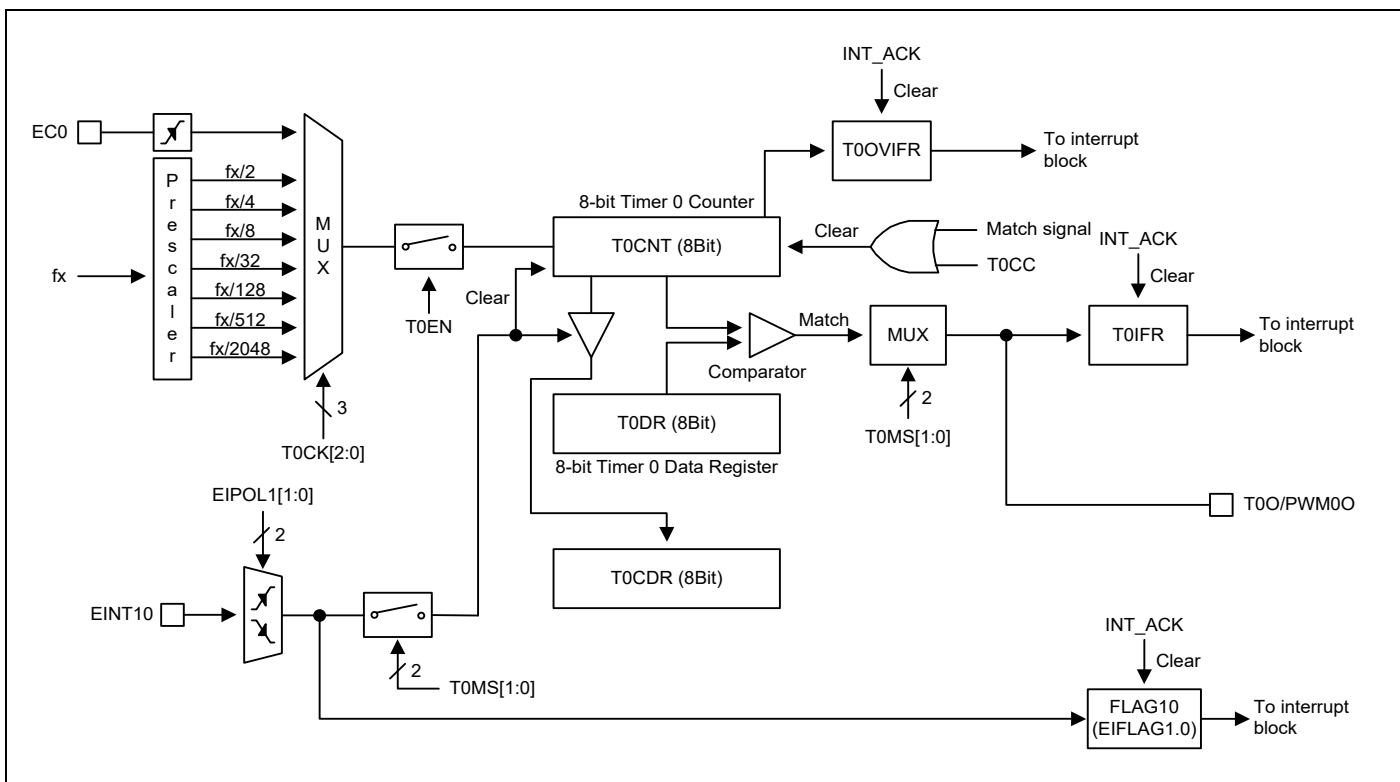


Figure 11.13 8-bit 定时器0方框图

11.5.6 寄存器图

Name	Address	Direction	Default	Description
T0CNT	B3H	R	00H	Timer 0 计数寄存器
T0DR	B4H	R/W	FFH	Timer 0 数据寄存器
T0CDR	B4H	R	00H	Timer 0 捕捉数据寄存器
T0CR	B2H	R/W	00H	Timer 0 控制寄存器

Table 11.6 定时器0 寄存器图

11.5.7 定时器/计数器0 寄存器说明

定时器/计数器 0寄存器 由T0CNT, T0DR, T0CDR, T0CR. 以及外部中断标志位寄存器(EIFLAG1)中的T0IFR 和 T0OVIFR 位组成.

11.5.8 定时器/计数器0 寄存器

TOCNT (定时器 0 计数寄存器): B3H

7	6	5	4	3	2	1	0
TOCNT7	TOCNT6	TOCNT5	TOCNT4	TOCNT3	TOCNT2	TOCNT1	TOCNT0
R	R	R	R	R	R	R	R

Initial value: 00H

TOCNT[7:0] T0 计数

TODR (定时器 0 数据寄存器): B4H

7	6	5	4	3	2	1	0
TODR7	TODR6	TODR5	TODR4	TODR3	TODR2	TODR1	TODR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: FFH

TODR[7:0] T0 数据

TOCDR (定时器 0 捕捉数据寄存器: 只读, 只用在捕捉模式): B4H

7	6	5	4	3	2	1	0
TOCDR7	TOCDR6	TOCDR5	TOCDR4	TOCDR3	TOCDR2	TOCDR1	TOCDR0
R	R	R	R	R	R	R	R

Initial value: 00H

TOCDR[7:0] T0 捕捉数据

T0CR (定时器 0 控制寄存器): B2H

7	6	5	4	3	2	1	0
TOEN	–	TOMS1	TOMS0	T0CK2	T0CK1	T0CK0	T0CC
RW	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

T0EN 控制 Timer 0

0 Timer 0 禁止

1 Timer 0 使能

TOMS[1:0] 控制 Timer 0 工作模式

TOMS1 TOMS0 Description

0 0 定时器/计数器 模式

0 1 PWM 模式

1 x 捕捉模式

T0CK[2:0] 选择时钟. fx 是系统时钟频率

T0CK2 T0CK1 T0CK0 Description

0 0 0 fx/2

0 0 1 fx/4

0 1 0 fx/8

0 1 1 fx/32

1 0 0 fx/128

1 0 1 fx/512

1 1 0 fx/2048

1 1 1 External Clock (EC0)

T0CC Clear timer 0 Counter

0 No effect

1 清除 Timer 0 计数器 (When write, automatically cleared "0" after being cleared counter)

NOTE)

1. 在捕捉模式使用匹配中断.
2. 参考外部中断标志位寄存器 1 (EIFLAG1).

11.6 定时器 1

11.6.1 概述

16位定时器1 由 定时器1A数据寄存器（高/低），定时器1B数据寄存器（高/低）和控制寄存器（高/低）组成 (T1ADRH, T1ADRL, T1BDRH, T1BDRL, T1CRH, T1CRL).

有4个操作模式:

- 16-bit 定时器/计数器 模式
- 16-bit 捕捉模式
- 16-bit PPG 输出模式 (单次)
- 16-bit PPG 输出模式 (重复)

定时器/计数器 1 可以由内部或外部时钟源(EC1)提供时钟. 时钟选择是通过时钟控制选择位(T1CK[2:0])进行选择..

- TIMER 1 时钟源: $f_x/(1, 2, 4, 8, 64, 512, 2048)$, EC1

捕捉模式下, 数据由EINT11捕捉到捕捉数据寄存器(T1BDRH/T1BDRL). 定时器/计数器模式时 Timer 1 通过T1O输出计数器和数据寄存器的比较结果. PPG模式下也可以通过PWM1O 口输出 PWM 波形.

T1EN	P1FSRL[5:4]	T1MS[1:0]	T1CK[2:0]	Timer 1
1	11	00	XXX	16-bit 定时器/计数器 模式
1	00	01	XXX	16-bit 捕捉模式
1	11	10	XXX	16-bit PPG 模式 (单次)
1	11	11	XXX	16-bit PPG 模式 (重复)

Table 11.7 定时器 1工作模式

11.6.2 16-bit 定时器/计数器 模式

图11.14 所示为通过寄存器选择16位定时器/计数器模式.

它有计数寄存器和数据寄存器. 计数寄存器由内部或外部时钟输入进行累加. Timer 1可以使用的输入时钟分频有 1, 2, 4, 8, 64, 512 和 2048. 当 T1CNTH, T1CNTL 和 T1ADRH, T1ADRL 值分别相同时, 会产生一个匹配信号同时产生定时器1 中断. T1CNTH, T1CNTL 值被匹配信号自动清零. 也可以通过软件(T1CC)进行清零.

外部时钟 (EC1) 在上升沿开始计数. 如果EC1 通过T1CK[2:0]选择时钟源, EC1口需要被P13IO位设置为输入状态.

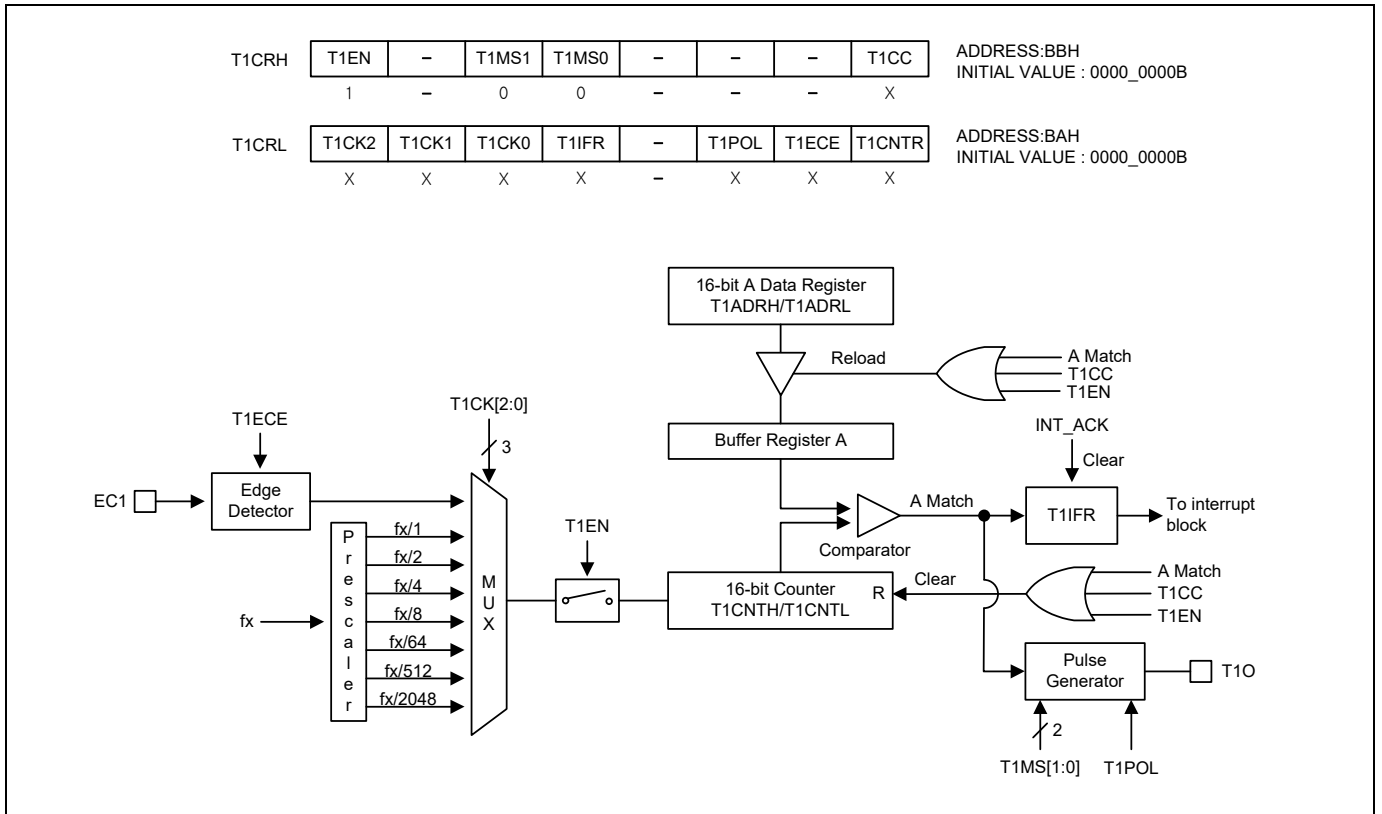


Figure 11.14 定时器1 16-bit 定时器/计数器 模式

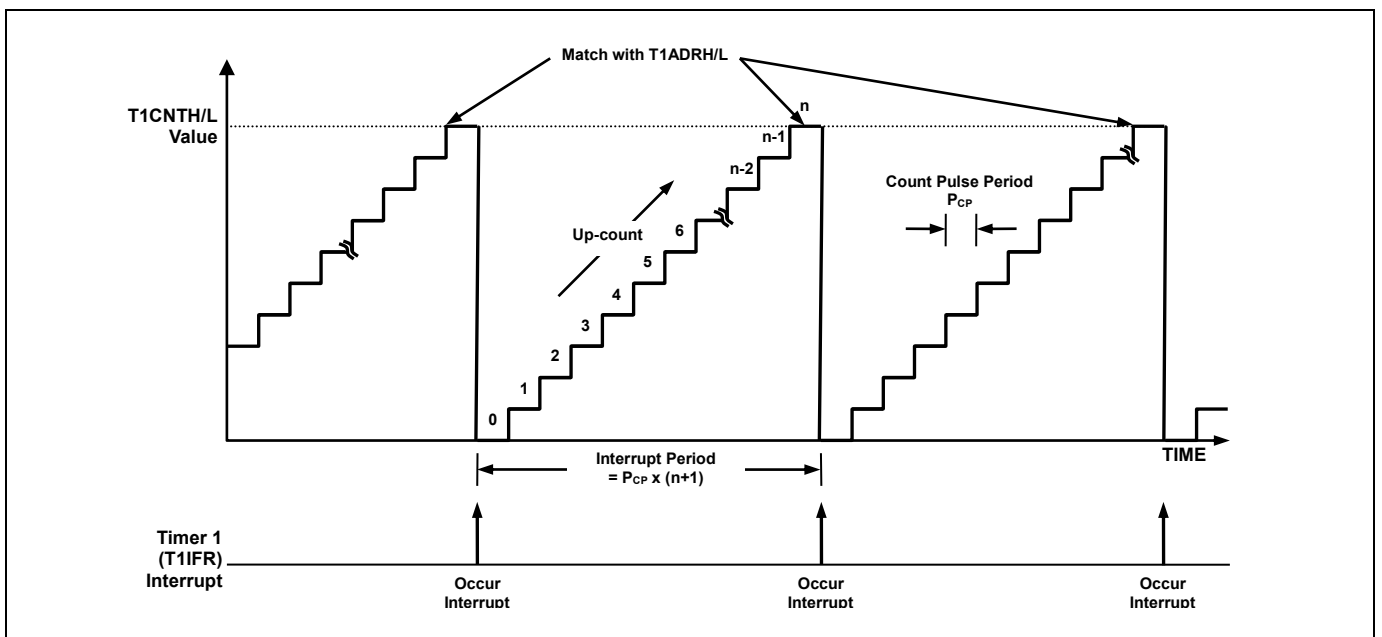


Figure 11.15 16-bit 定时器/计数器 1 举例

11.6.3 16-bit 捕捉模式

定时器1 通过把T1MS[1:0] 设置为 '01'将其设为16位捕捉模式. 时钟可以使用内部和外部时钟. 当T1CNTH/T1CNTL 的值和T1ADRH/T1ADRL相同时 , 它基本上和定时器/计数器模式以及中断的发生具有相同的功能. T1CNTH, T1CNT L 值被匹配信号自动清零或者通过软件(T1CC)清零.

当捕捉的脉冲信号宽度大于定时器周期最大值时, 捕捉模式下的定时器中断是非常有用的

捕捉的结果装载到T1BDRH/T1BDRL.

根据 EIPOL1 寄存器设置, 选择外部中断EINT11触发边沿 , 当然, EINT11 引脚必须设为输入状态.

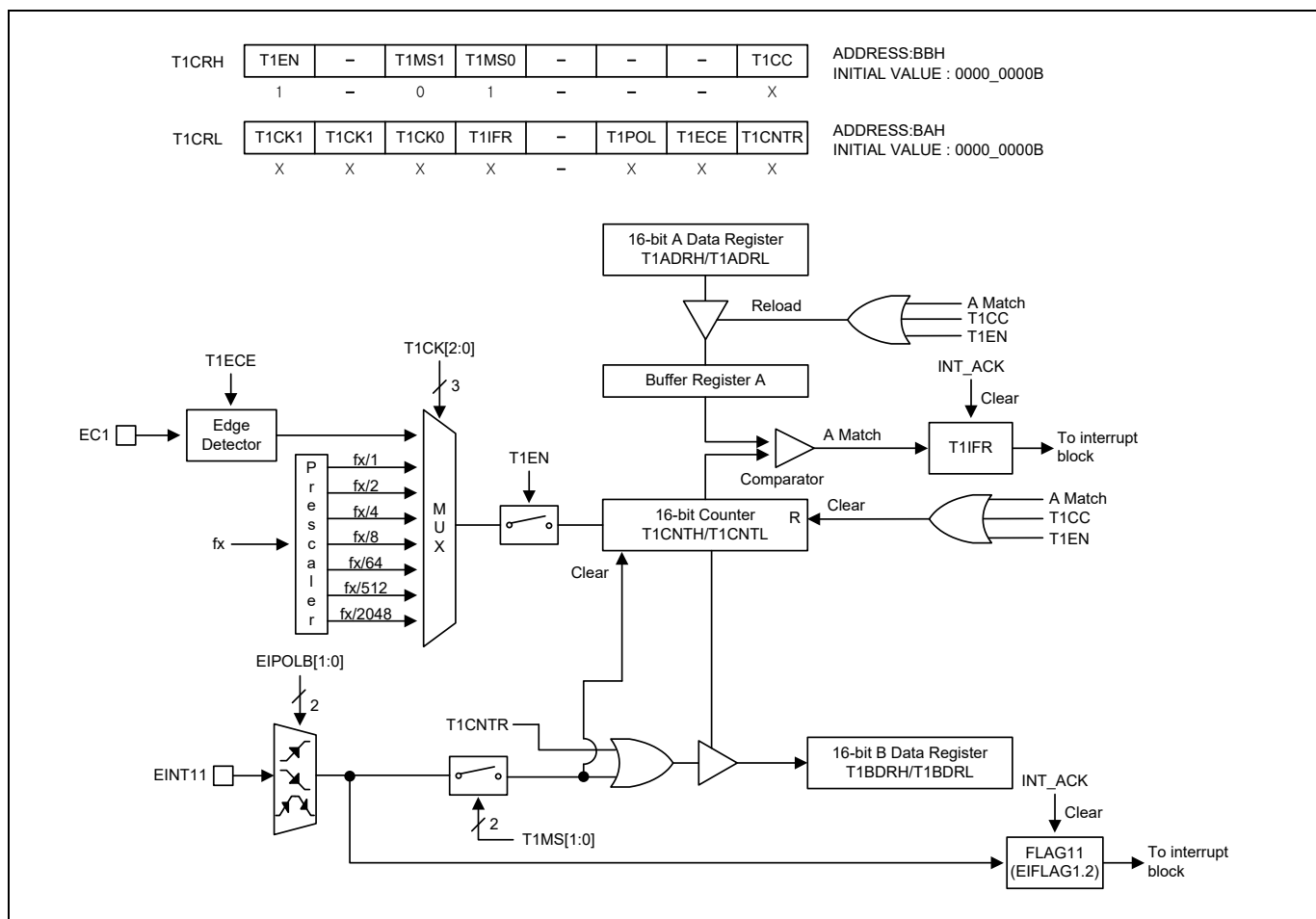


Figure 11.16 定时器1 16-bit 捕捉模式1

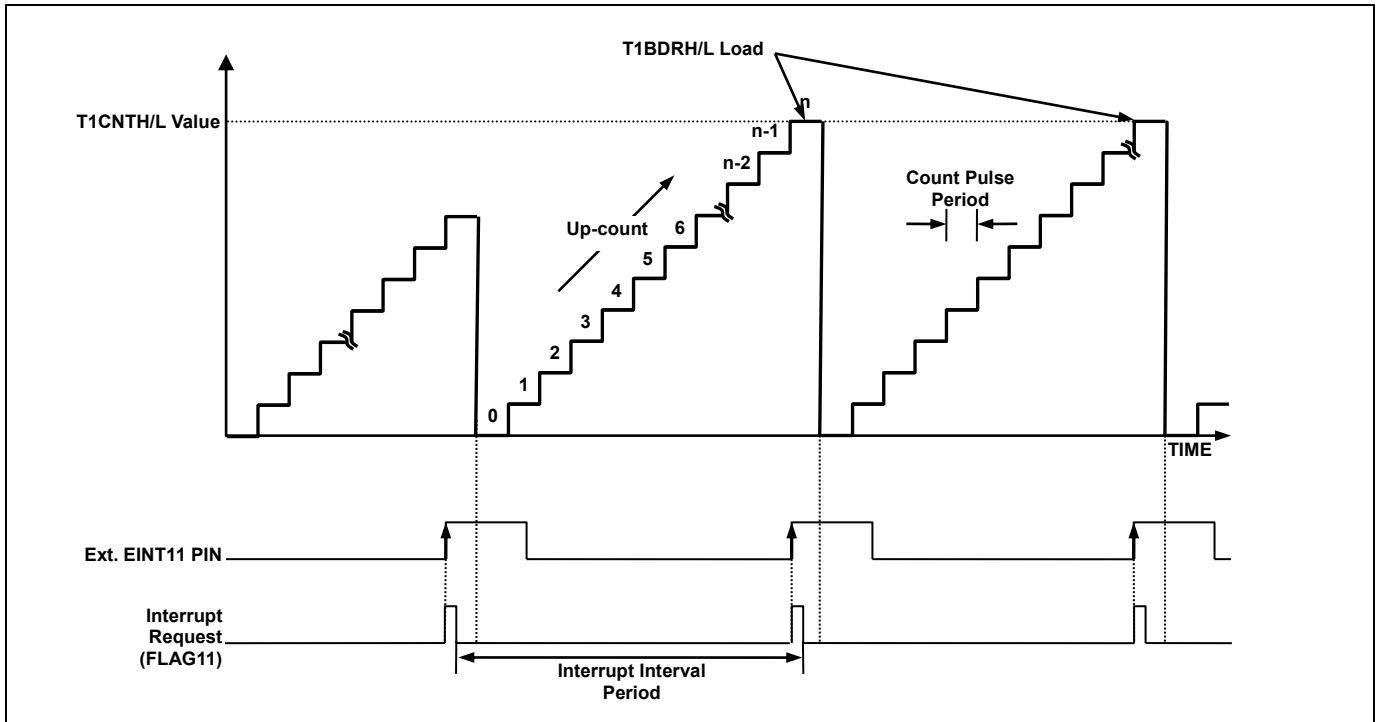


Figure 11.17 定时器1 输入捕捉模式

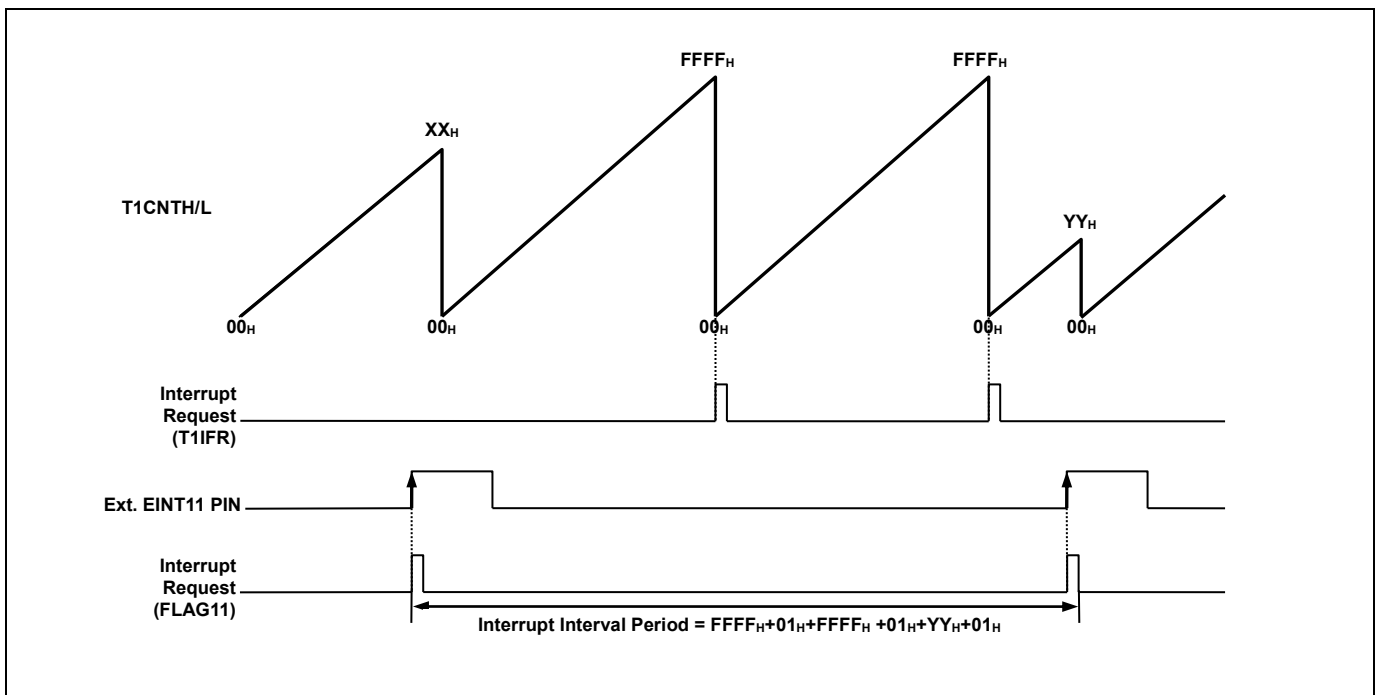


Figure 11.18 捕捉模式定时器溢出

11.6.4 16-bit PPG 模式

定时器1有PPG (可编程脉冲发生) 功能。 PPG 模式中, T1O/PWM1O 引脚输出取决于16位分辨率PWM输出. 通过将P1FSRL[5:4] 设置为 '11'将该引脚配置为PWM输出. PWM的周期由T1ADRH/T1ADRL的值决定. PWM占空比输出由T1BDRH/T1BDRL的值决定.

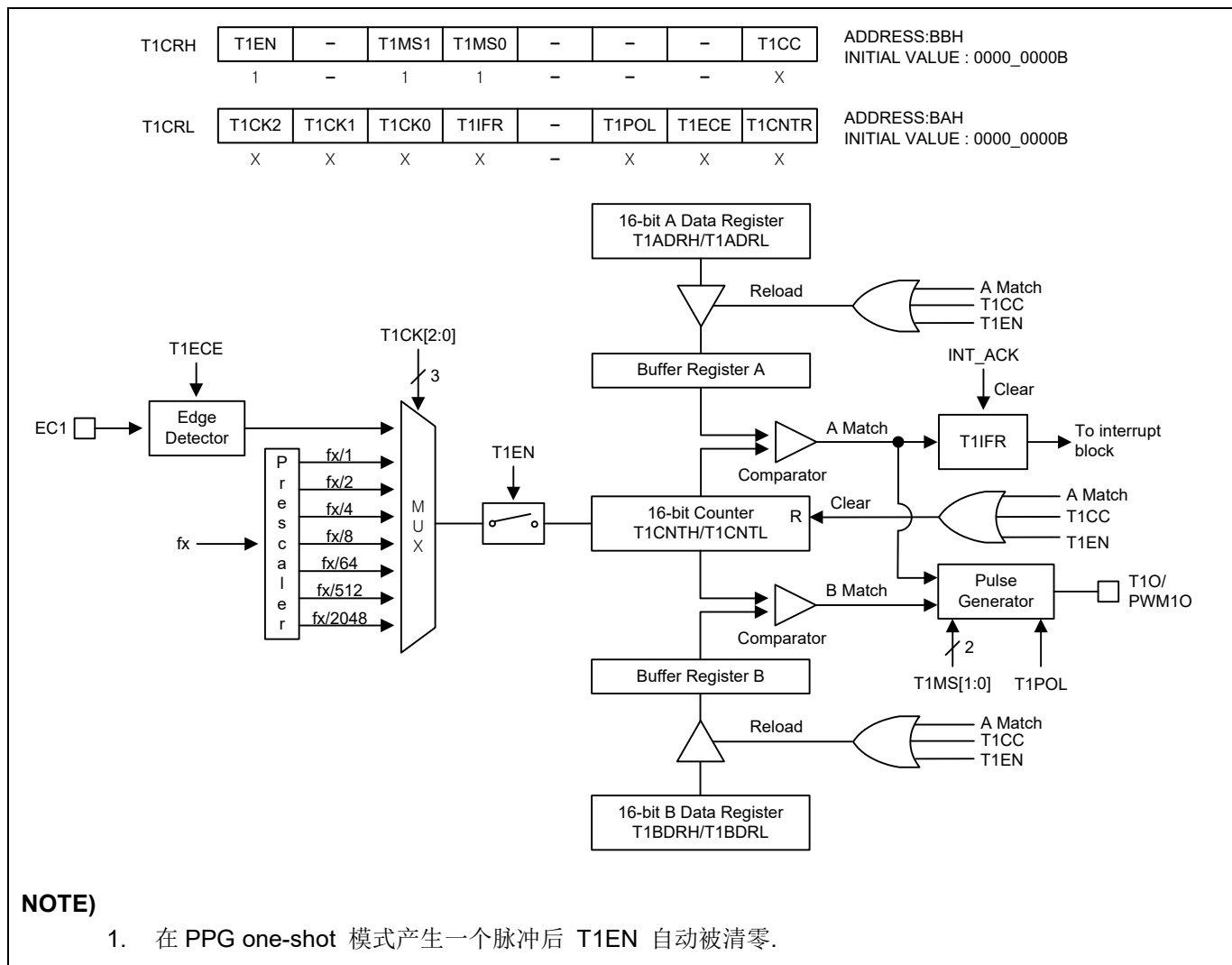


Figure 11.19 定时器1 16-bit PPG 模式

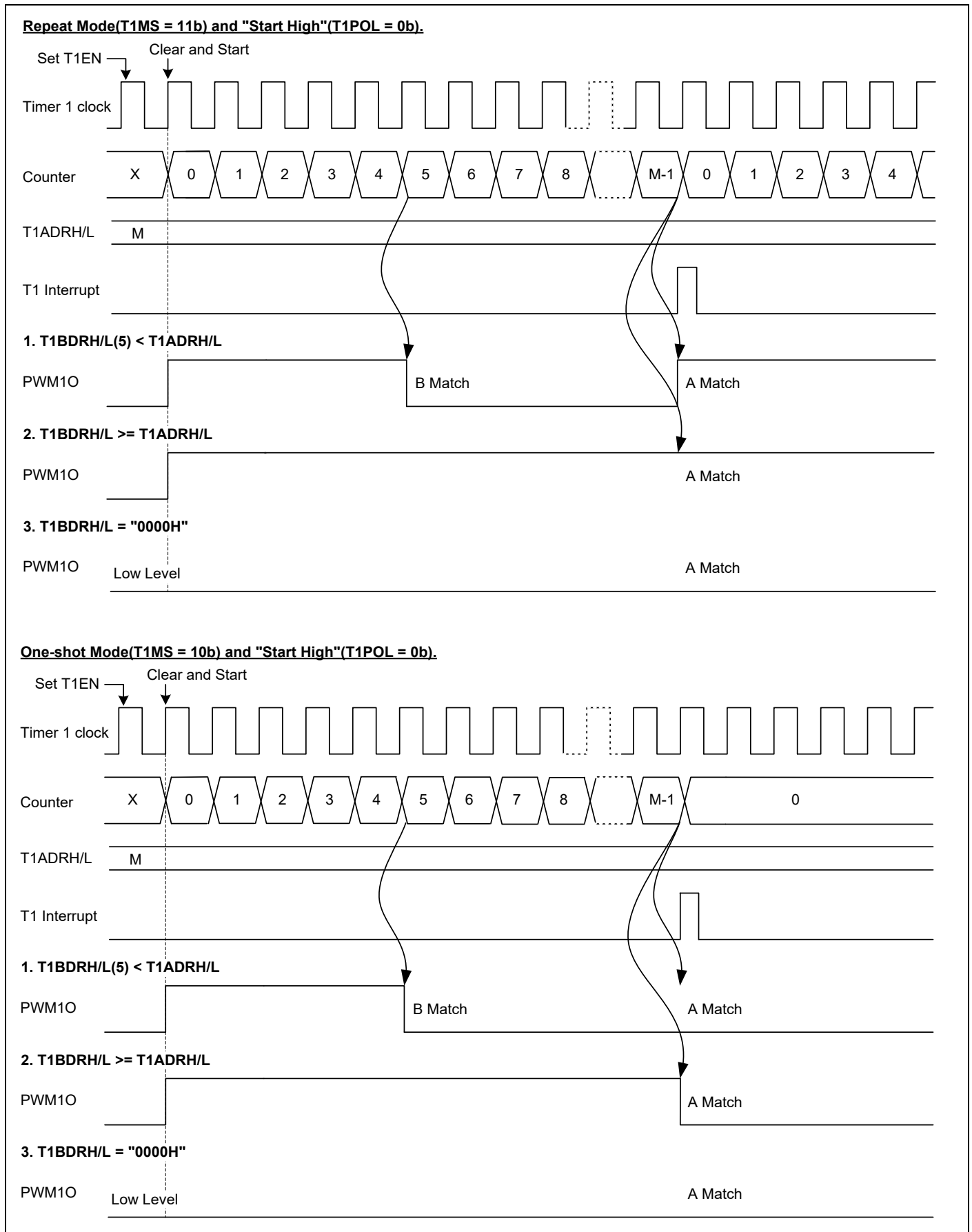


Figure 11.20 定时器1 16-bit PPG 模式时序图

11.6.5 方框图

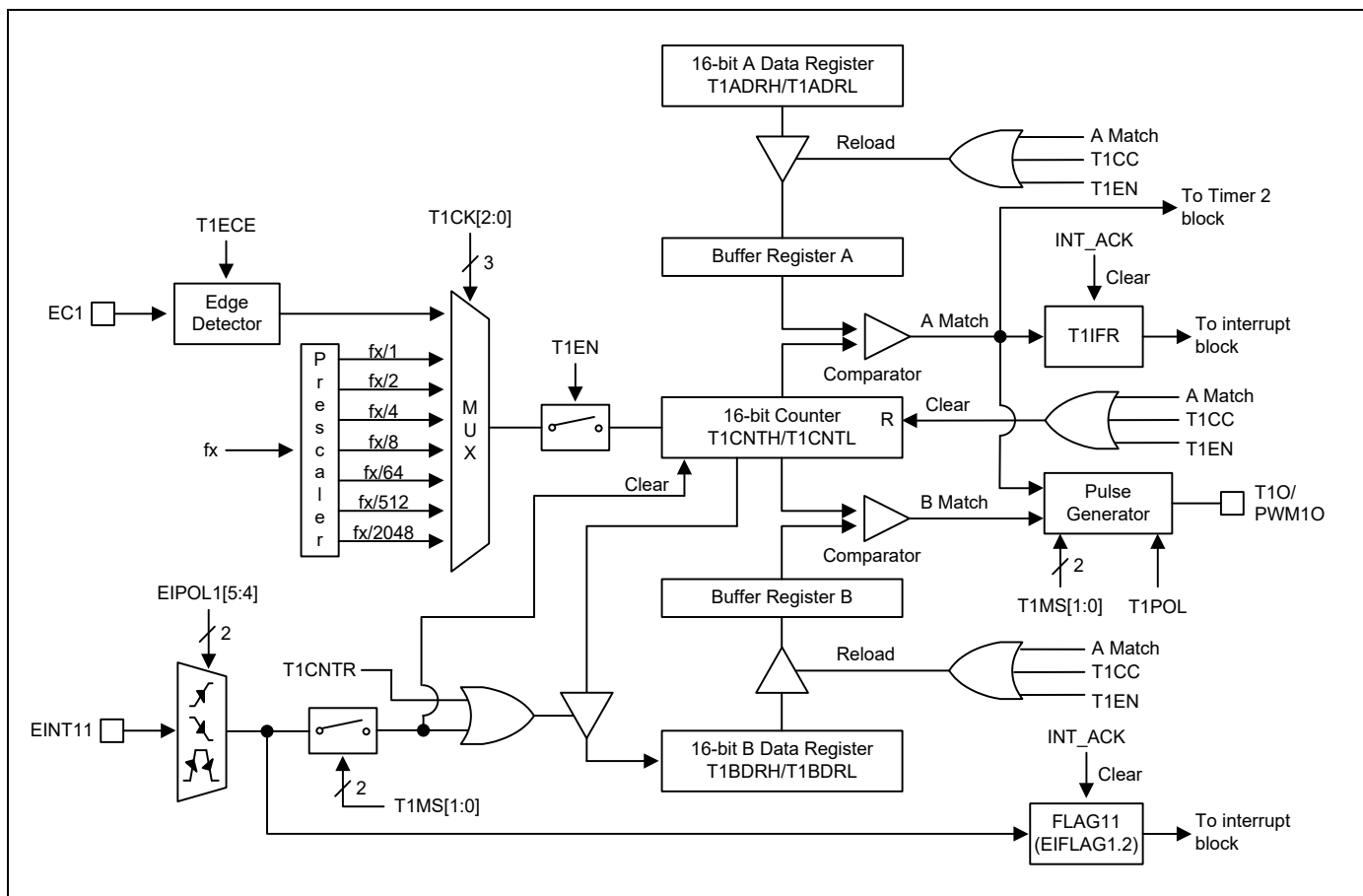


Figure 11.21 16-bit 定时器 1 框图

11.6.6 寄存器图

Name	Address	Direction	Default	Description
T1ADRH	BDH	R/W	FFH	Timer 1 A 数据高位寄存器
T1ADRL	BCH	R/W	FFH	Timer 1 A 数据低位寄存器
T1BDRH	BFH	R/W	FFH	Timer 1 B 数据高位寄存器
T1BDRL	BEH	R/W	FFH	Timer 1 B 数据低位寄存器
T1CRH	BBH	R/W	00H	Timer 1 控制高位寄存器
T1CRL	BAH	R/W	00H	Timer 1 控制低位寄存器

Table 11.8 定时器 1 寄存器

11.6.7 定时器/计数器 1 寄存器说明

定时器/计数器 1 由T1ADRH, T1ADRL, T1BDRH, T1BDRL, T1CRH 和T1CRL组成.

11.6.8 定时器/计数器 1 寄存器

T1ADRH (定时器 1 A 数据高位寄存器): BDH

7	6	5	4	3	2	1	0
T1ADRH7	T1ADRH6	T1ADRH5	T1ADRH4	T1ADRH3	T1ADRH2	T1ADRH1	T1ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T1ADRH[7:0] T1 A 数据高字节

T1ADRL (定时器 1 A 数据低位寄存器): BCH

7	6	5	4	3	2	1	0
T1ADRL7	T1ADRL6	T1ADRL5	T1ADRL4	T1ADRL3	T1ADRL2	T1ADRL1	T1ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T1ADRL[7:0] T1 A 数据低字节

NOTE)

1. PPG 模式时 T1ADRH/T1ADRL 不要写入“0000H”

T1BDRH (定时器 1 B 数据高位寄存器): BFH

7	6	5	4	3	2	1	0
T1BDRH7	T1BDRH6	T1BDRH5	T1BDRH4	T1BDRH3	T1BDRH2	T1BDRH1	T1BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T1BDRH[7:0] T1 B 数据高字节

T1BDRL (定时器 1 B 数据低位寄存器): BEH

7	6	5	4	3	2	1	0
T1BDRL7	T1BDRL6	T1BDRL5	T1BDRL4	T1BDRL3	T1BDRL2	T1BDRL1	T1BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T1BDRL[7:0] T1 B 数据低字节

T1CRH (定时器 1 高位控制寄存器): BBH

7	6	5	4	3	2	1	0
T1EN	-	T1MS1	T1MS0	-	-	-	T1CC
RW	-	R/W	RW	-	-	-	RW

Initial value: 00H

- T1EN** 控制 Timer 1
 - 0 Timer 1 禁止
 - 1 Timer 1 使能 (计数清零然后开始计数)
- T1MS[1:0]** 控制 Timer 1 工作模式

T1MS1	T1MS0	Description
0	0	定时器/计数器 模式 (T1O: toggle at A match)
0	1	捕捉模式 (The A match interrupt can occur)
1	0	PPG 单次模式 (PWM1O)
1	1	PPG 重复模式 (PWM1O)
- T1CC** 清除 1 计数
 - 0 No effect
 - 1 清除Timer 1 计数 (写入时, 自动清零)

T1CRL (定时器 1 低位控制寄存器): BAH

7	6	5	4	3	2	1	0
T1CK2	T1CK1	T1CK0	T1IFR	–	T1POL	T1ECE	T1CNTR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W

Initial value: 00H

T1CK[2:0] 选择 Timer 1 时钟. fx 是系统主时钟频率

T1CK2	T1CK1	T1CK0	Description
0	0	0	fx/2048
0	0	1	fx/512
0	1	0	fx/64
0	1	1	fx/8
1	0	0	fx/4
1	0	1	fx/2
1	1	0	fx/1
1	1	1	External clock (EC1)

T1IFR T1 中断产生时置'1'. 在该位写入0进行清零或通过 INT_ACK 信号自动清零..

0	T1 中断不产生
1	T1 中断产生

T1POL T1O/PWM1O输出极性选择

0	高电平有效 (低电平时 T1O/PWM1O 禁止)
1	低电平有效 (高电平时 T1O/PWM1O 禁止)

T1ECE Timer 1 外部时钟脉冲边沿选择

0	下降沿
1	上升沿

T1CNTR Timer 1 计数器读取控制

0	No effect
1	把计数器的值加载到 B 数据寄存器 (写入时加载后自动清零)

11.7 定时器 2

11.7.1 概述

16位定时器 2 由多路转换器, T2ADRH, T2ADRL, T2BDRH, T2BDRL, T2CRH, T2CRL组成.

有4个操作模式:

- 16-bit 定时器/计数器 模式
- 16-bit 捕捉模式
- 16-bit PPG 输出模式 (一次)
- 16-bit PPG 输出模式 (重复)

定时器 2 可以由内部、外部时钟源或者T1 A Match提供时钟.通过控制设置(T2CK[2:0]) 位来选取时钟.

- TIMER 2 时钟源: $f_x/(1, 2, 4, 8, 32, 128, 512)$, T1 A Match

捕捉模式下, 数据通过EINT12 将数据捕捉到捕捉数据寄存器 (T2BDRH/T2BDRL). 在定时器/计数器 模式下, 当计数值等于T2ADRH/L时, T2O 口打开. 在PPG模式时PWM2O 口也可以输出PWM波形.

T2EN	P1FSRL[3:2]	T2MS[1:0]	T2CK[2:0]	Timer 2
1	11	00	XXX	16-bit 定时器/计数器 模式
1	00	01	XXX	16-bit 捕捉模式
1	11	10	XXX	16-bit PPG 模式 (单次)
1	11	11	XXX	16-bit PPG 模式 (重复)

Table 11.9 Timer 2 工作模式选择

11.7.2 16-bit 定时器/计数器 模式

图11.22 所示为定时器/计数器 模式控制图。

它由计数寄存器和数据寄存器。计数寄存器通过内部时钟或timer 1 A Match时钟计数。定时器2 可以使用的时钟分频有1, 2, 4, 8, 32,128, 512 和 T1 A Match (T2CK[2:0])。当 T2CNTH/T2CNTL 和T2ADRH/T2ADRL 的值相等时，产生匹配信号同时产生定时器2中断。T2CNTH/T2CNTL 值被匹配信号自动清零也可以通过软件清零(T2CC)。

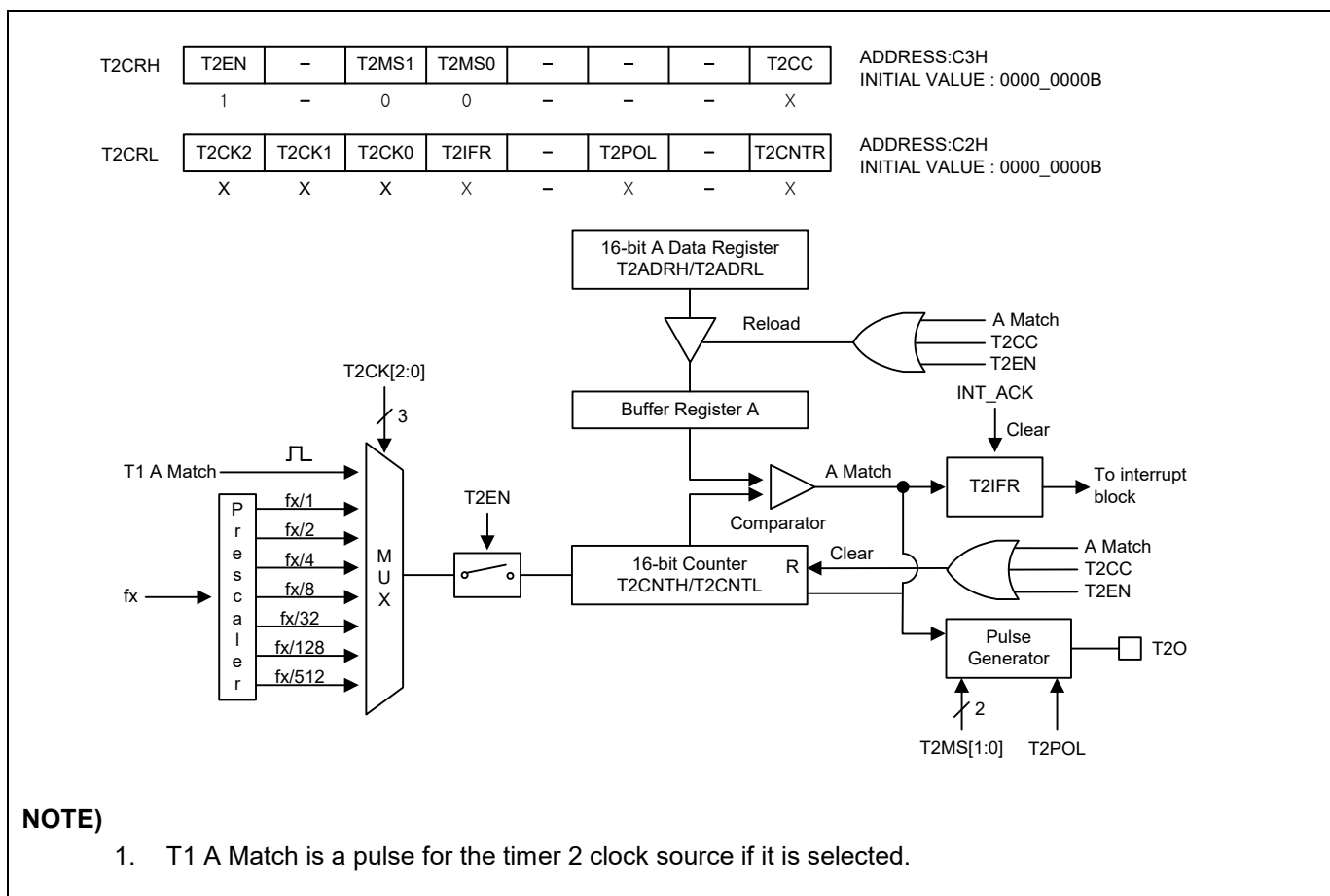


Figure 11.22 定时器2 16-bit 定时器/计数器 模式

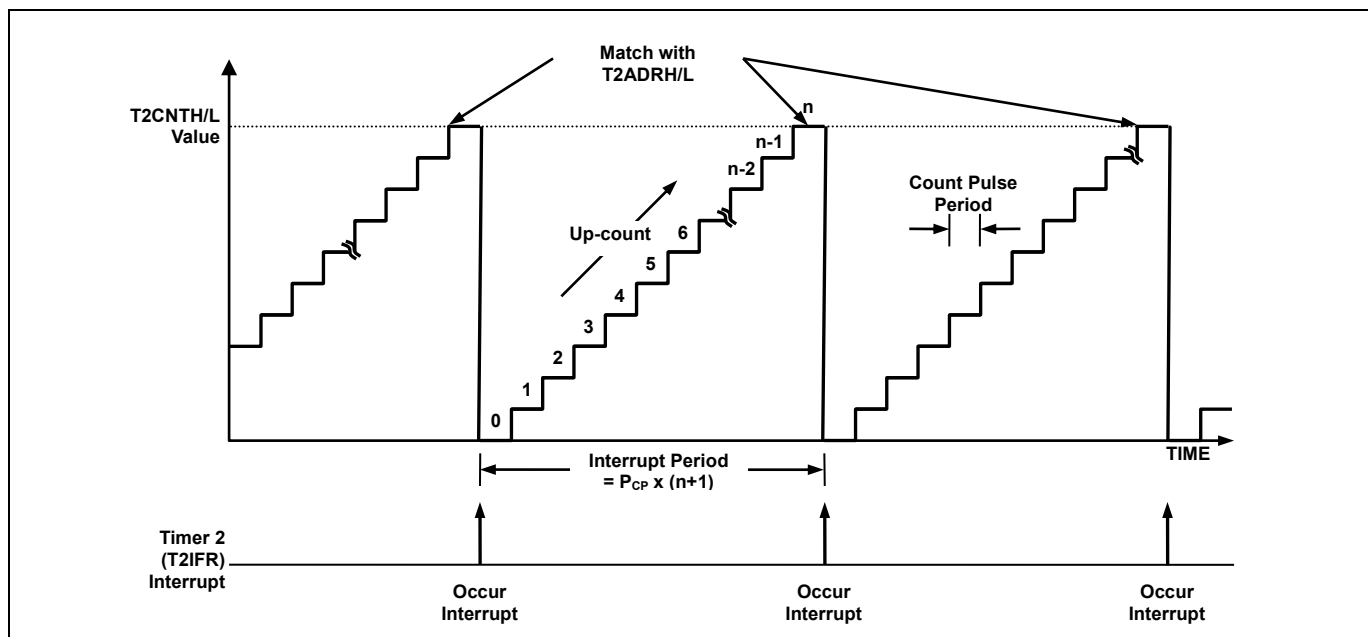


Figure 11.23 16-bit 定时器/计数器 2 举例

11.7.3 16-bit 捕捉模式

设置 T2MS[1:0] 为 '01'，将定时器 2 配置为捕捉模式. 时钟源可采用内部时钟. 当T2CNTH/T2CNTL 的值和T2ADRH/T2ADRL相同时基本上和定时器/计数器模式以及中断的发生具有相同的功能. T2CNTH, T2CNTL 值被匹配信号自动清零或者通过软件(T2CC)清零..

当捕捉的脉冲信号宽度大于定时器周期最大值时，捕捉模式下的定时器中断是非常有用的.

捕捉结果加载到 T2BDRH/T2BDRL. 在定时器 2 捕捉模式, (T2O)波形输出是不可用的.

根据 EIPOL1 寄存器设置, 选择EINT12 触发边沿. 当然, EINT12 引脚需要设置位输入状态.

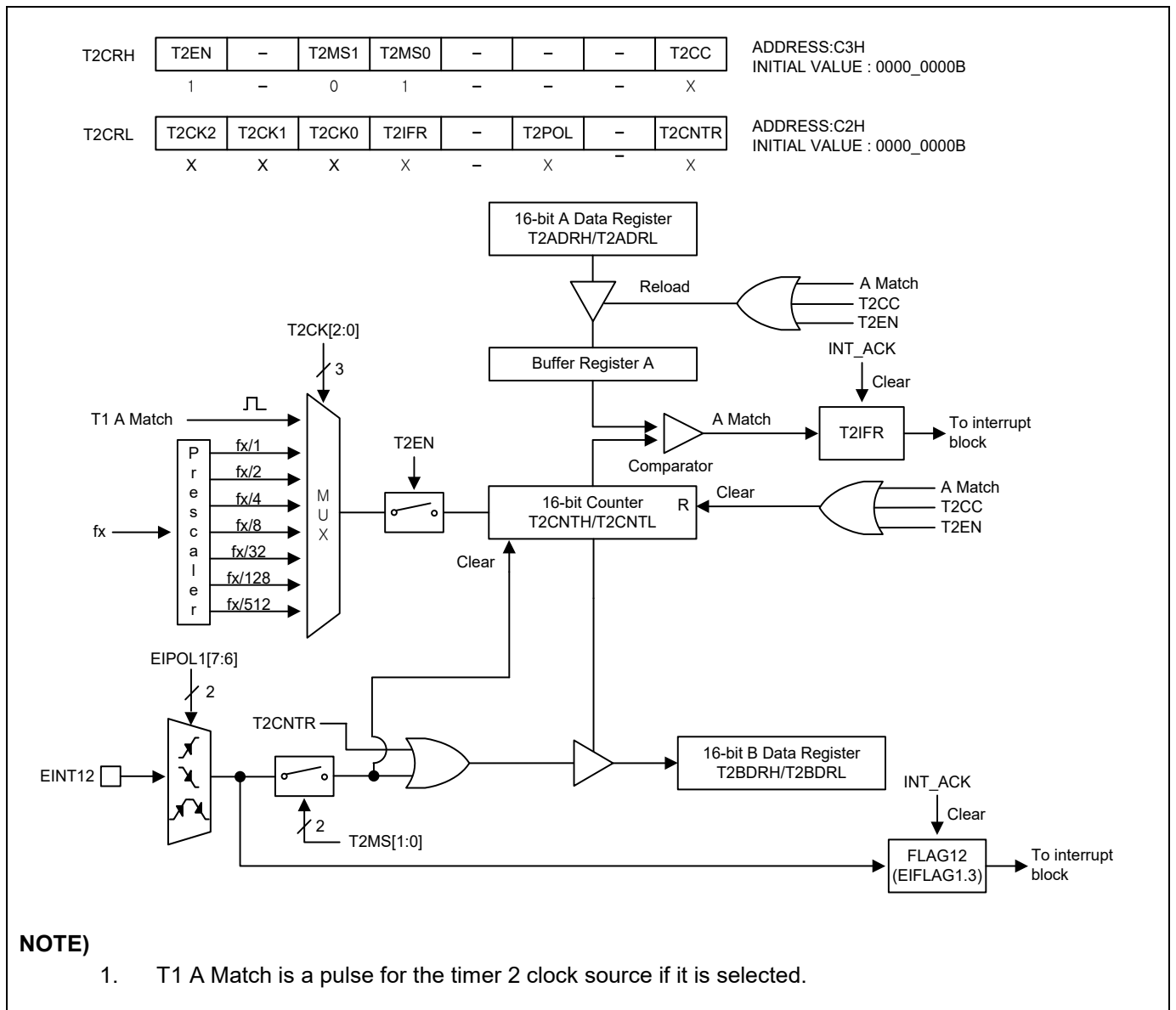


Figure 11.24 定时器2 16-bit 捕捉模式

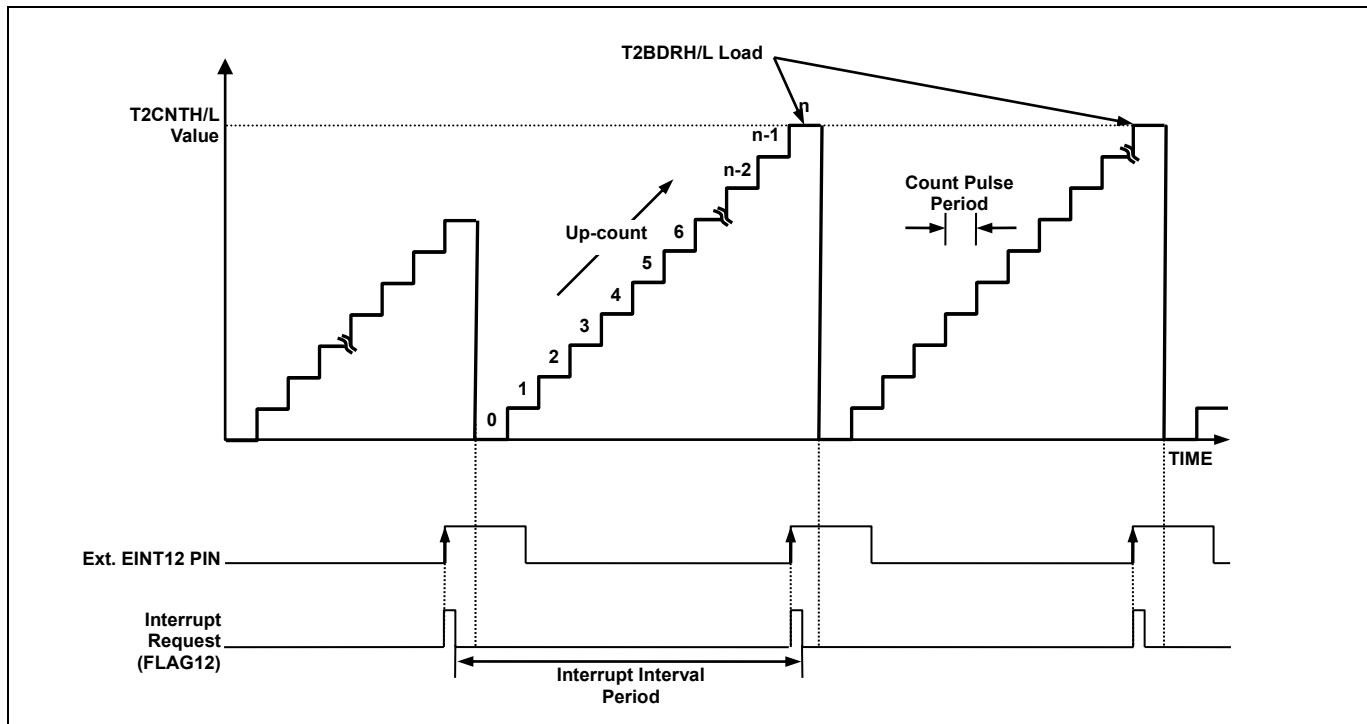


Figure 11.25 定时器 2 输入捕捉模式操作

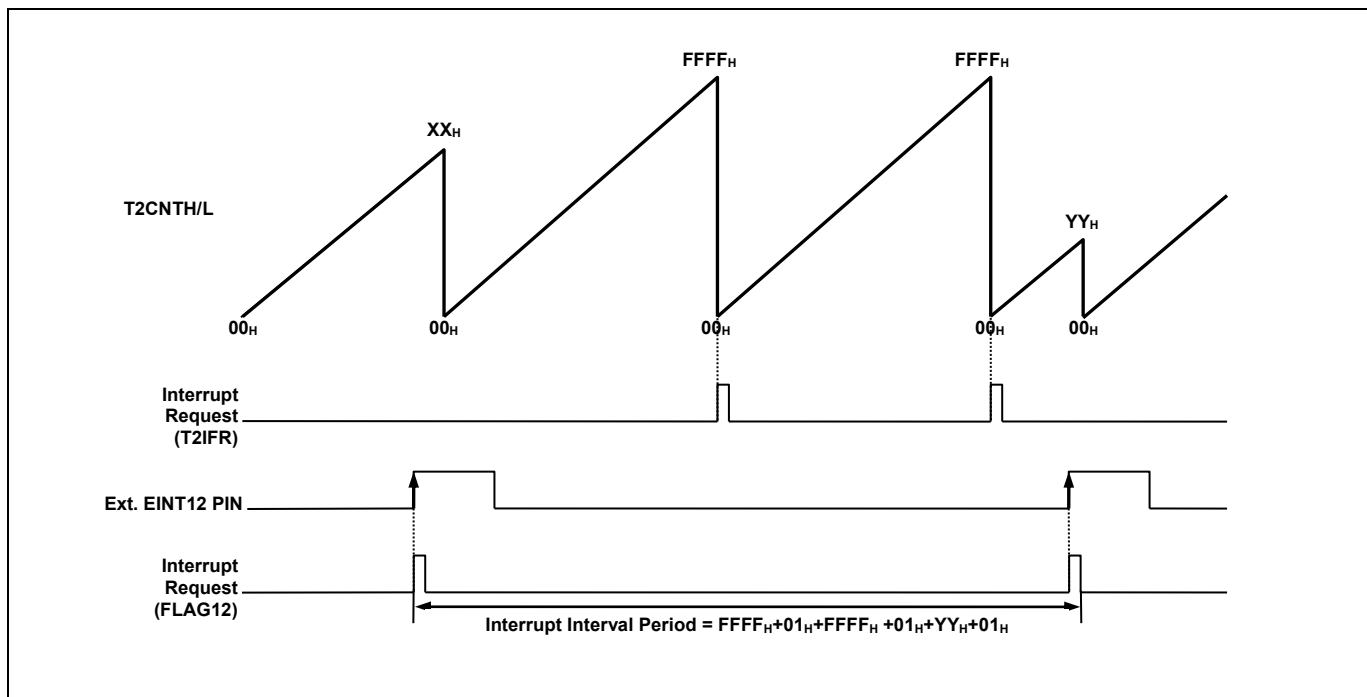


Figure 11.26 捕捉模式定时器溢出

11.7.4 16-bit PPG 模式

定时器2有 PPG (可编程脉冲发生) 功能. PPG 模式时, T2O/PWM2O 引脚输出16位分辨率PWM. 该引脚应该通过设置P1FSRL[3:2] 为 '11'将其配置为PWM模式. PWM 输出周期由T2ADRH/T2ADRL决定. 占空比由T2BDRH/T2BDRL 决定.

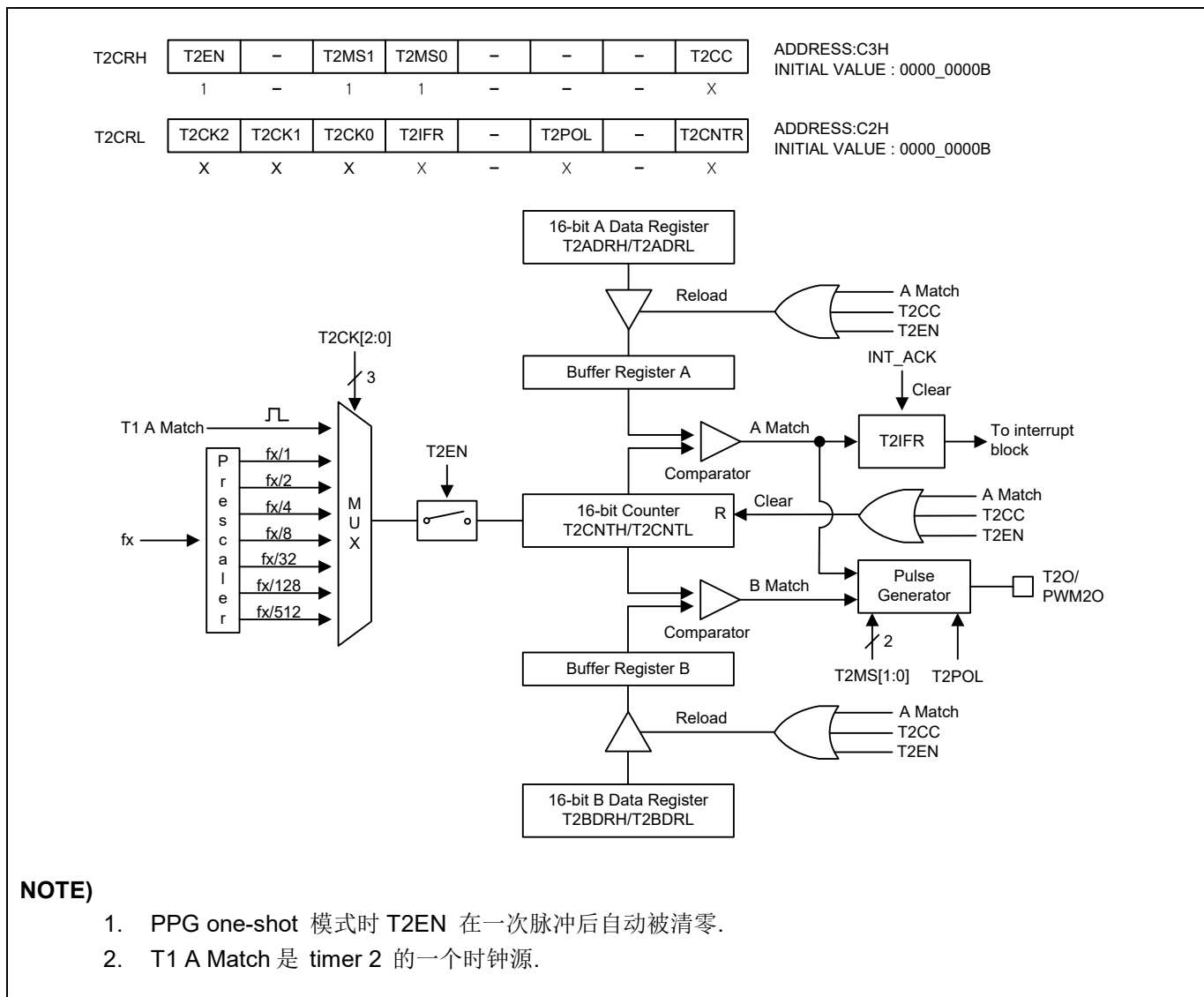


Figure 11.27 定时器2 16-bit PPG 模式

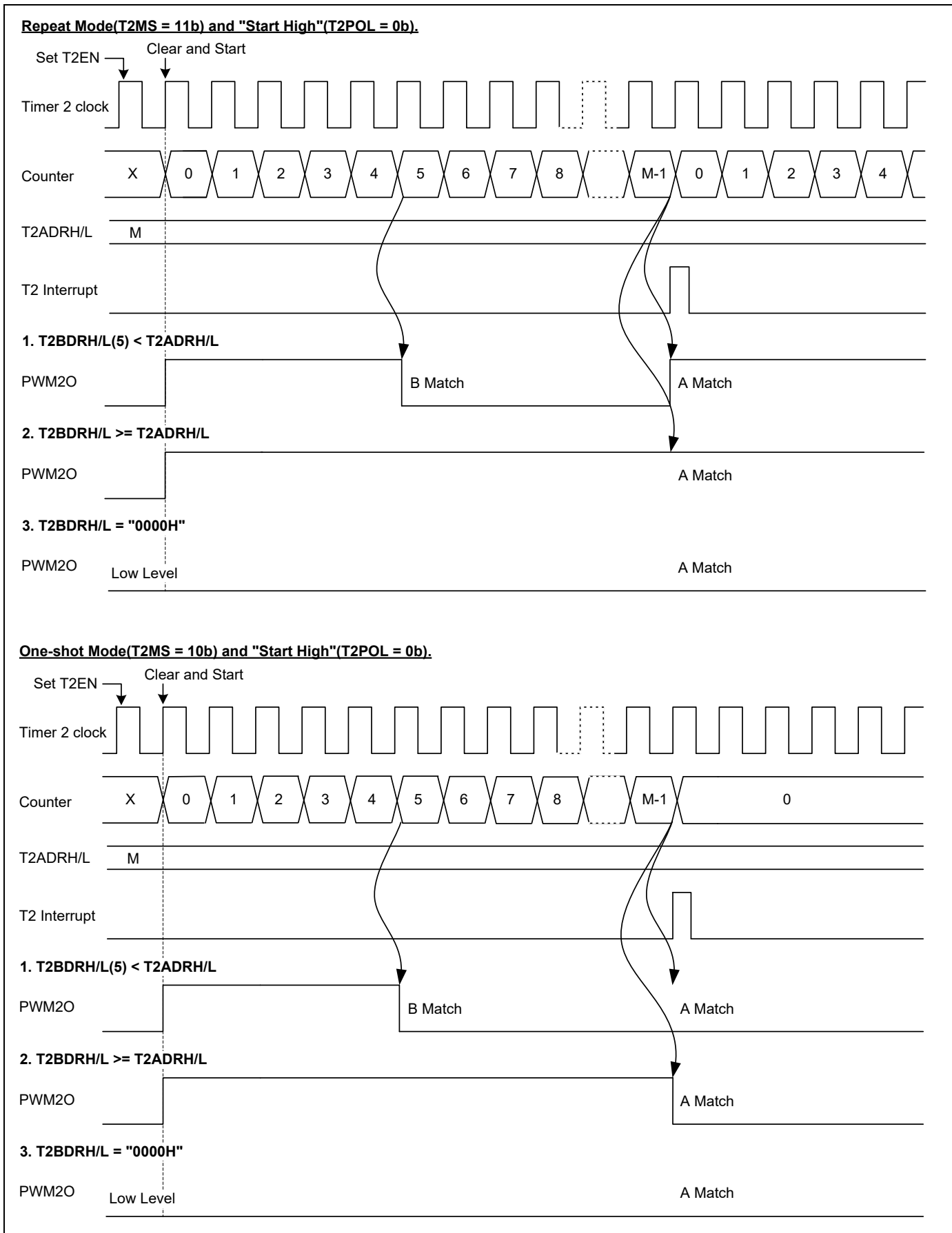


Figure 11.28 定时器2 16-bit PPG 模式时序图

11.7.5 方框图

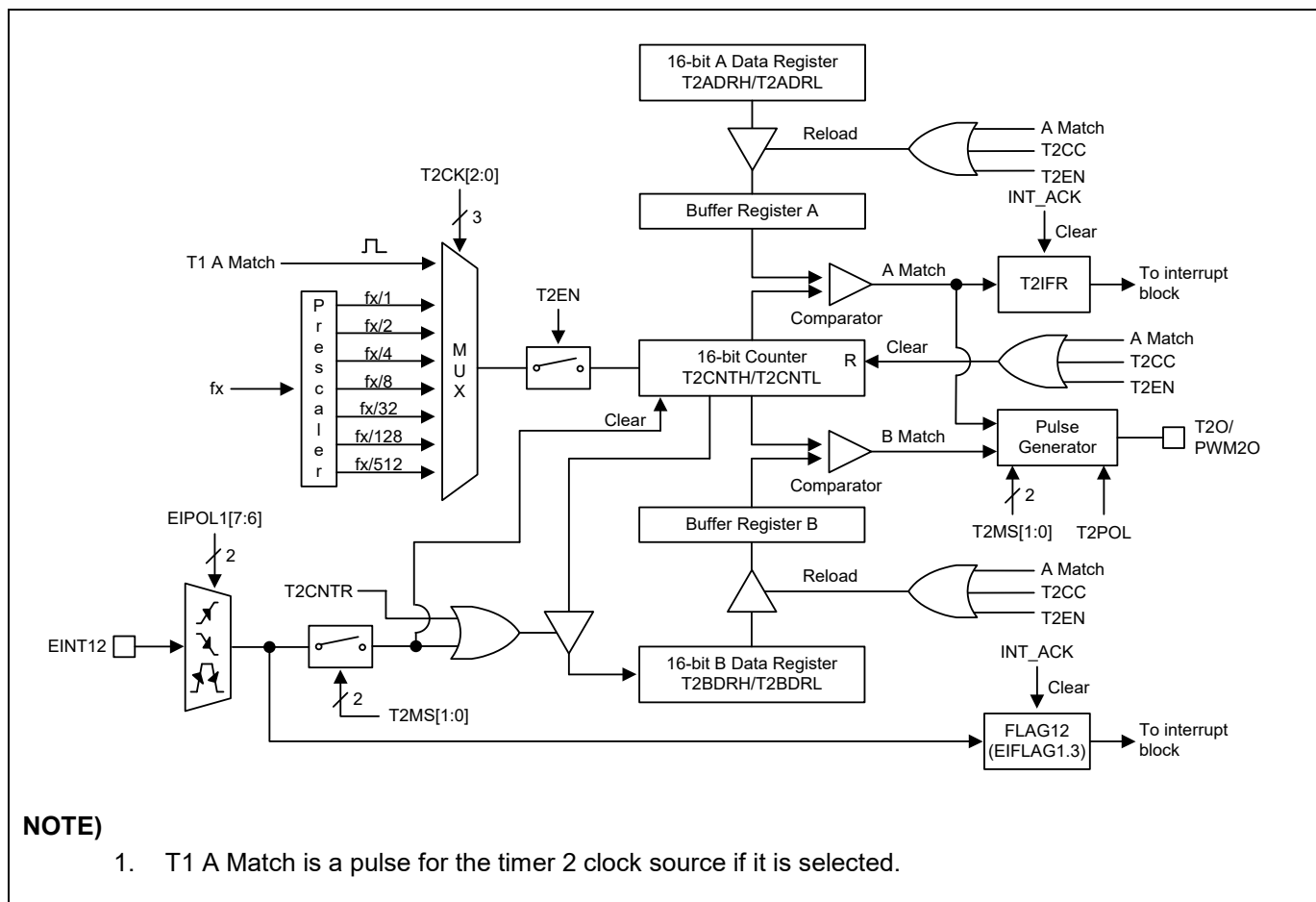


Figure 11.29 16-bit 定时器 2 方框图

11.7.6 寄存器图

Name	Address	Direction	Default	Description
T2ADRH	C5H	R/W	FFH	Timer 2 A 数据高位寄存器
T2ADRL	C4H	R/W	FFH	Timer 2 A 数据低位寄存器
T2BDRH	C7H	R/W	FFH	Timer 2 B 数据高位寄存器
T2BDRL	C6H	R/W	FFH	Timer 2 B 数据低位寄存器
T2CRH	C3H	R/W	00H	Timer 2 控制高位寄存器
T2CRL	C2H	R/W	00H	Timer 2 控制低位寄存器

Table 11.10 Timer 2 寄存器图

11.7.7 定时器/计数器 2 寄存器说明

定时器/计数器 2 寄存器由T2ADRH, T2ADRL, T2BDRH , T2BDRL, T2CRH, T2CRL组成.

11.7.8 定时器/计数器 2 寄存器

T2ADRH (定时器 2 A 高位数据寄存器): C5H

7	6	5	4	3	2	1	0
T2ADRH7	T2ADRH6	T2ADRH5	T2ADRH4	T2ADRH3	T2ADRH2	T2ADRH1	T2ADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T2ADRH[7:0] T2 A 数据高字节

T2ADRL (定时器 2 A 低位数据寄存器): C4H

7	6	5	4	3	2	1	0
T2ADRL7	T2ADRL6	T2ADRL5	T2ADRL4	T2ADRL3	T2ADRL2	T2ADRL1	T2ADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T2ADRL[7:0] T2 A 数据低字节

NOTE)

1. PPG 模式时 T2ADRH/T2ADRL 寄存器不要写入“0000H”.

T2BDRH (定时器2 B 高位数据寄存器): C7H

7	6	5	4	3	2	1	0
T2BDRH7	T2BDRH6	T2BDRH5	T2BDRH4	T2BDRH3	T2BDRH2	T2BDRH1	T2BDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T2BDRH[7:0] T2 B 数据高字节

T2BDRL (定时器 2 B 低位数据寄存器): C6H

7	6	5	4	3	2	1	0
T2BDRL7	T2BDRL6	T2BDRL5	T2BDRL4	T2BDRL3	T2BDRL2	T2BDRL1	T2BDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: FFH

T2BDRL[7:0] T2 B 数据低字节

T2CRH (定时器 2 高位控制寄存器): C3H

7	6	5	4	3	2	1	0
T2EN	–	T2MS1	T2MS0	–	–	–	T2CC
RW	–	R/W	R/W	–	–	–	RW

Initial value: 00H

T2EN	控制 Timer 2	
0	Timer 2 禁止	
1	Timer 2 使能 (Counter clear and start)	
T2MS[1:0]	控制 Timer 2 工作模式	
	T2MS1	T2MS0 Description
0	0	定时器/计数器 模式 (T2O: toggle at A match)
0	1	捕捉模式 (The A match interrupt can occur)
1	0	PPG one-shot 模式 (PWM2O)
1	1	PPG repeat 模式 (PWM2O)
T2CC	清除定时器 2 计数器	
0	No effect	
1	清除 Timer 2 计数 (写入时 清除计数后自动清零)	

T2CRL (定时器 2 低位控制寄存器): CAH

7	6	5	4	3	2	1	0
T2CK2	T2CK1	T2CK0	T2IFR	-	T2POL	-	T2CNTR
R/W	R/W	R/W	R/W	-	RW	-	RW

Initial value: 00H

T2CK[2:0] 选择的定时器 2 时钟. fx 是系统主时钟频率

T2CK2	T2CK1	T2CK0	Description
0	0	0	fx/512
0	0	1	fx/128
0	1	0	fx/32
0	1	1	fx/8
1	0	0	fx/4
1	0	1	fx/2
1	1	0	fx/1
1	1	1	T1 A Match

T2IFR T2 匹配中断产生时置 ‘1’. 写入0进行清零或者通过INT_ACK信号自动清零..

0	没有T2中断产生
1	T2中断产生

T2POL T2O/PWM2O 输出极性选择

0	高电平有效 (低电平 T2O/PWM2O 禁止)
1	低电平有效 (低电平 T2O/PWM2O 禁止)

T2CNTR Timer 2 计数器读取控制

0	No effect
1	加载计数值到 B 数据寄存器(写入时, 加载后自动清零)

11.8 定时器 3, 4

11.8.1 概述

定时器 3 和定时器4 可以作为8位定时器/计数器 或者 结合在一起用作16位定时器/计数器。每个8位定时器/计数器模块由多路转换器，8位数据寄存器，8位计数寄存器，控制寄存器和捕捉数据寄存器组成(T3CNT, T3DR, T3CAPR, T3CR, T4CNT, T4DR, T4CAPR, T4CR)。作为PWM模式，有 PWM 寄存器 (T4PPRL, T4PPRH, T4ADRL, T4ADRH, T4BDRL, T4BDRH, T4CDRL, T4CDRH, T4DLYA, T4DLYB, T4DLYC)。

有以下5个特性:

- 8-bit 定时器/计数器模式
- 8-bit 捕捉模式
- 16-bit 定时器/计数器模式
- 16-bit 捕捉模式
- 10-bit PWM 模式

定时器/计数器3 和 4 可以通过内部或外部(EC3)时钟计时。时钟源通过时钟选择位(T3CK[2:0], T4CK[3:0])来选择。定时器4可以选用的时钟要比定时器3多。

- TIMER 3 时钟源: $f_x/(2, 4, 8, 32, 128, 512, 2048)$, EC3
- TIMER 4 时钟源: $f_x/(1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384)$, T3 时钟

捕捉模式时，通过 EINT0/EINT1，数据被捕捉到数据寄存器 (T3CAPR, T4CAPR)。

8位定时器/计数器模式时，当计数值等于T3DR/T4DR，T3O/T4O口打开。16位定时器/计数器模式时，定时器 3 由T3O输出计数器和数据寄存器的比较结果。PWM 波形有PWMAA, PWMAB, PWMBB, PWMCA, PWMCB 口输出 (6-channel)。

16BIT	T3MS	T4MS	PWM4E	T3CK[2:0]	T4CK[3:0]	Timer 3	Timer 4
0	0	0	0	XXX	XXXX	8-bit 定时器/计数器模式	8-bit 定时器/计数器模式
0	1	1	0	XXX	XXXX	8-bit 捕捉模式	8-bit 捕捉模式
1	0	0	0	XXX	XXXX	16-bit 定时器/计数器模式	
1	1	1	0	XXX	XXXX	16-bit 捕捉模式	
0	X	X	1	XXX	XXXX	10-bit PWM 模式	

Table 11.11 定时器 3, 4 工作模式

11.8.2 8-bit 定时器/计数器 3, 4 模式

图11.30所示为8-bit 定时器/计数器模式控制。

两个8位定时器又各自的计数寄存器和数据寄存器。计数寄存器通过外部时钟输入或内部时钟开始增加。定时器3可以使用的内部时钟分频有2, 4, 8, 32, 128, 512, 2048 和EC3。定时器4可以使用的输入时钟分频有1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 和定时器3 时钟分频。当T3CNT, T4CNT 和 T3DR, T4DR 的值分别相等时, 定时器3,4产生中断

外部时钟 (EC3) 在上升沿开始计数。如果EC3 通过T3CK[2:0]来选择时钟源, EC3 口应设为输入口。定时器4 不能使用外部时钟EC3。

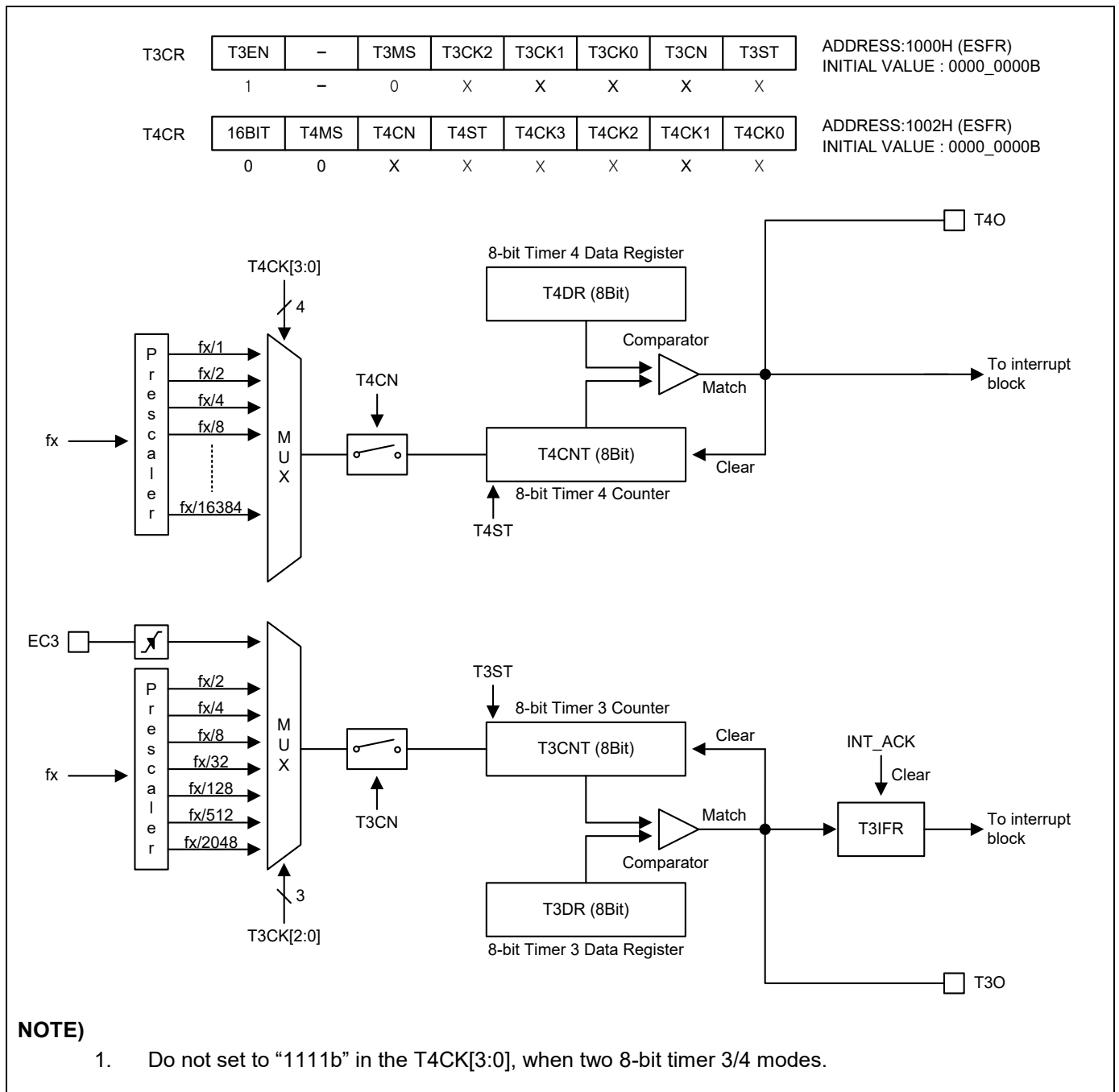


Figure 11.30 定时器3,4 8-bit 定时器/计数器 模式

11.8.3 16-bit 定时器/计数器 3 模式

图11.31所示为16-bit 定时器/计数器模式控制.

16-bit 定时器有计数寄存器和数据寄存器. 计数寄存器通过内部或外部时钟输入开始计数, 定时器 3 可以使用的输入时钟分频有2, 4, 8, 32, 128, 512 和 2048.

T3CNT, T4CNT 从0000H 到 FFFFH 计数直到与T3DR, T4DR 的值相匹配然后清除到0000H. 匹配信号发出产生定时器3中断(没有定时器 4 中断).通过 T3CK[2:0] 选择时钟源, 16BIT bit 必须设置为 '1'. Timer 3是低位优先 LSB 8-bit, 定时器 4是高位优先 MSB 8-bit.

外部时钟 (EC3) 在上升沿开始计数. 如果 EC3 通过 T3CK[2:0]时钟, EC3口需要设置为输入口.

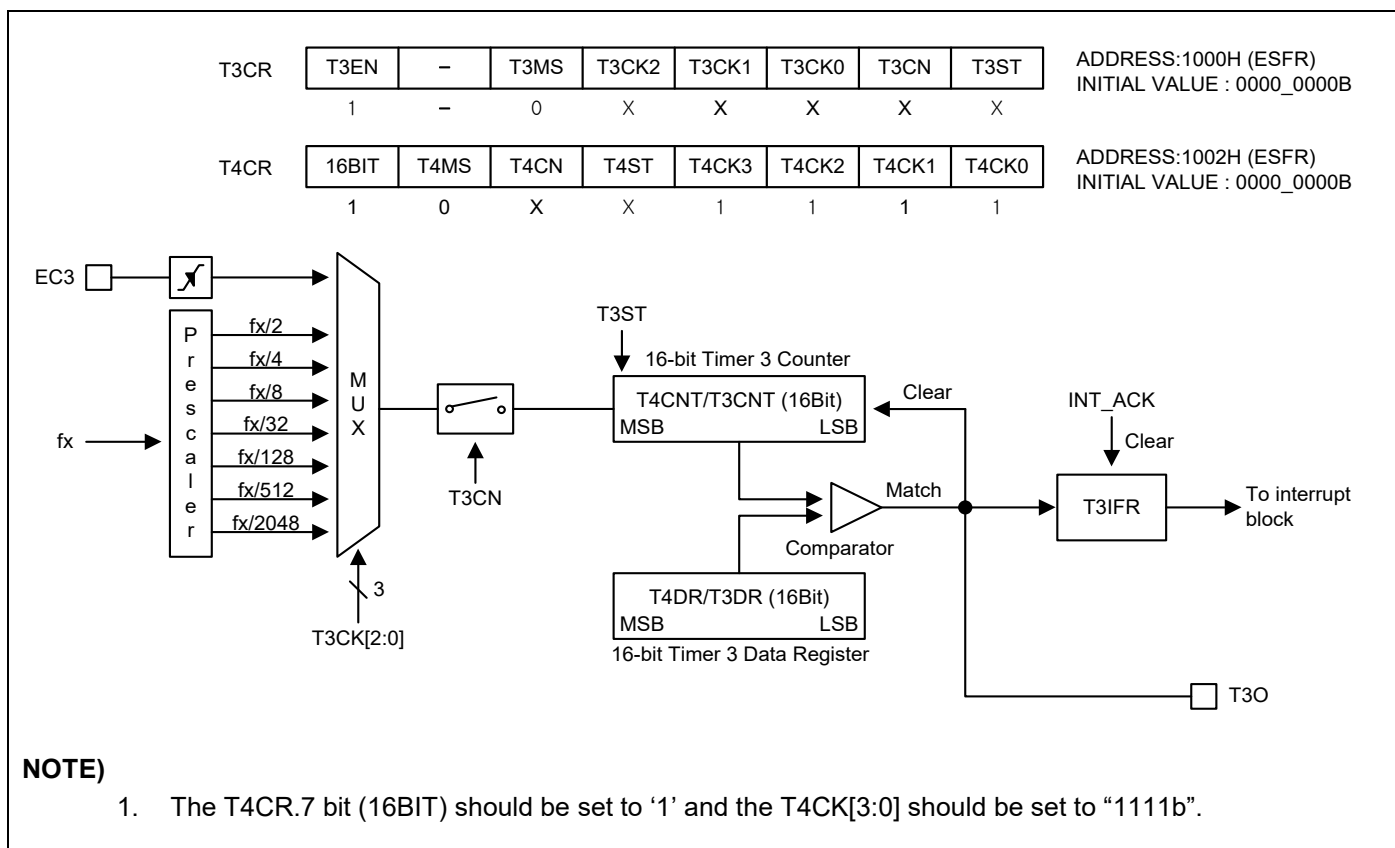


Figure 11.31 定时器3 16-bit 定时器/计数器 模式

11.8.4 8-bit 定时器 3, 4 捕捉模式

8-bit 捕捉模式通过下图所示进行控制寄存器的选择.

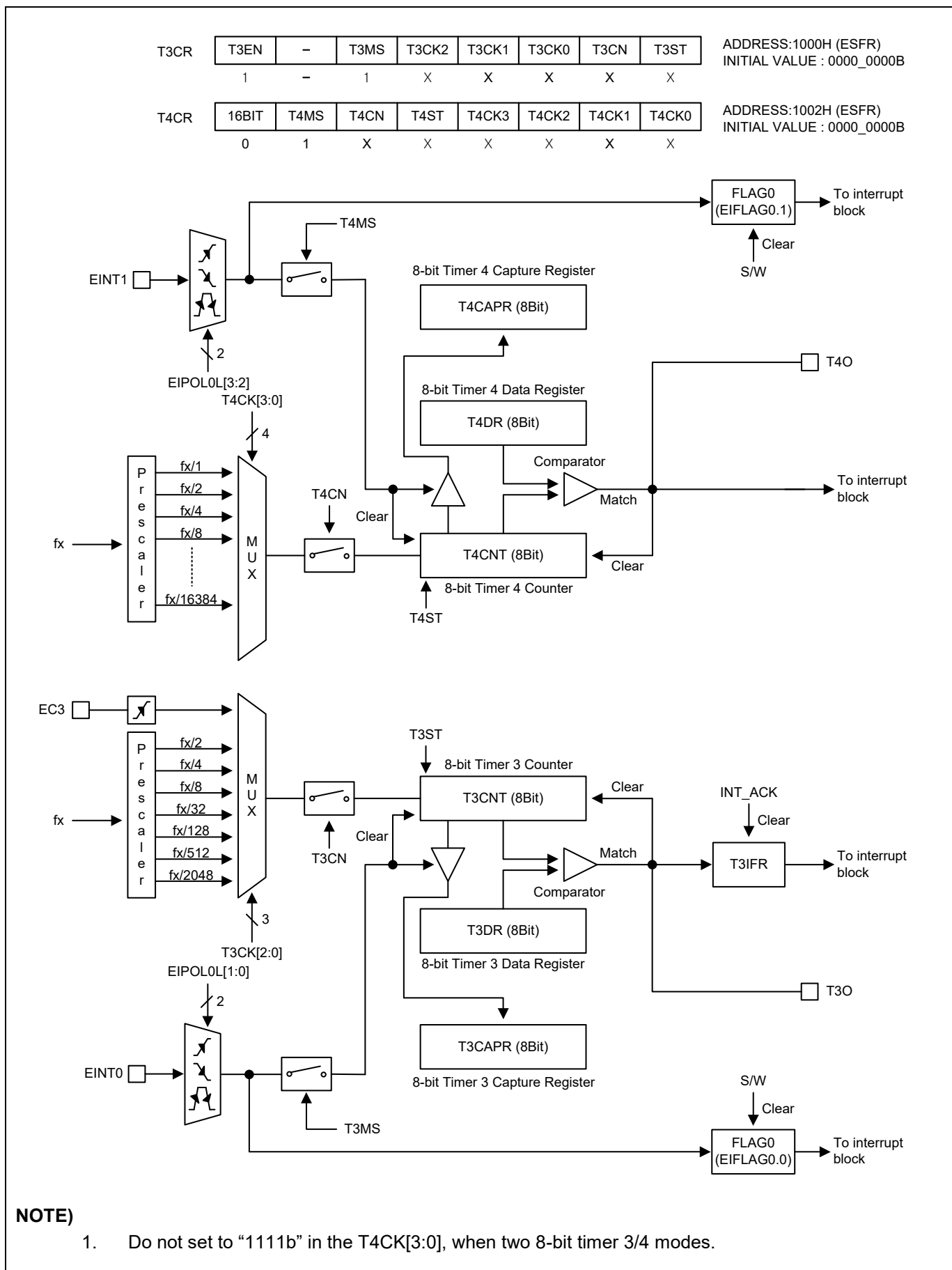
定时器3, 4 捕捉模式通过置位 T3MS, T4MS. 时钟源可用内部或外部时钟. Basically, 当T3CNT, T4CNT的值等于 T3DR, T4DR时, 它基本上和8位定时器/计数器功能相同. T3CNT, T4CNT 的值被匹配信号自动清零.

当捕捉的脉冲信号宽度大于定时器周期最大值时, 捕捉模式下的定时器中断是非常有用的.

捕捉结果加载到T3CAPR, T4CAPR. 在定时器 3, 4 捕捉模式时, 定时器 3, 4 (T3O, T4O) 不能输出波形.

根据 EIPOL0L 寄存器的设置, 选择外部中断EINT0 和 EINT1 功能. 当然, EINT0和 EINT1 脚必须设置为输入口.

T3CAPR 和 T3DR 在相同的地址. 在捕捉模式时, 读取时读T3CAPR 而不是T3DR, 但是写入时是T3DR, T4CAPR 有相同的功能.



NOTE)

1. Do not set to "1111b" in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.32 定时器3,4 8-bit 捕捉模式

11.8.5 定时器3 16-bit 捕捉模式

图11.33所示为16-bit 捕捉模式控制选择.

16-bit 捕捉模式和8-bit 捕捉模式操作时一样的, 除了定时器寄存器使用16 bits. 16-bit定时器 3 捕捉模式需要设置T3MS, T4MS 为 '1'.通过T3CK[2:0] 选择时钟, 16BIT位必须设置为'1'. 定时器 3 是低位优先 LSB 8-bit, 定时器 4 是高位优先 MSB 8-bit.

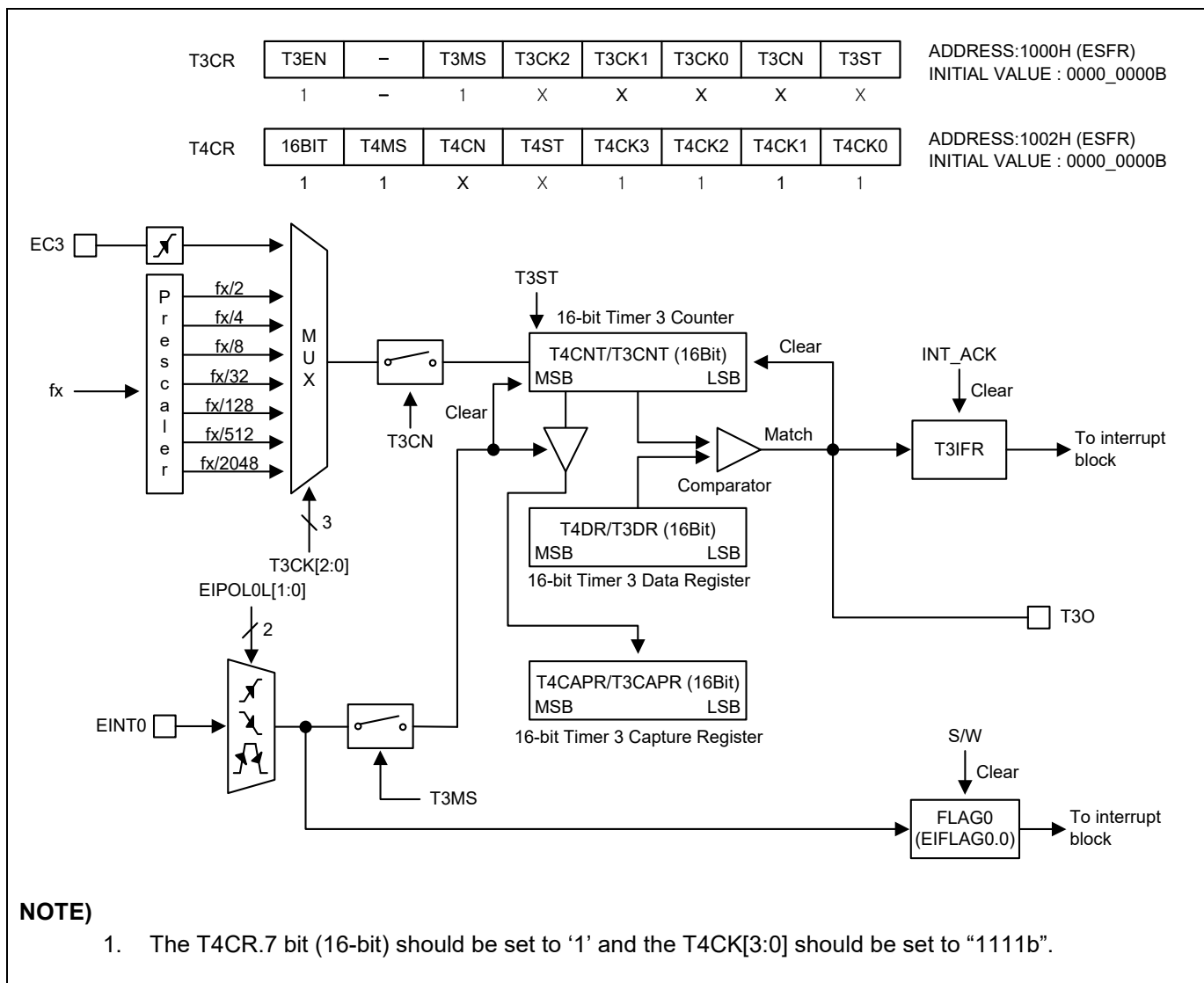


Figure 11.33 定时器3 16-bit 捕捉模式

11.8.6 定时器4 10-bit PWM 模式

定时器 4 有一个高速PWM (Pulse Width Modulation). 在 PWM 模式下, 6个通道引脚输出10位分辨率的PWM. 这些脚需要通过设置PWM4E 为'1'来配置为PWM. 当2bit +T4CNT 和T4PPRH/L 的值相同时, 产生一个周期匹配信号同时产生定时器4中断. 10-bit PWM 模式时,当 10-bit 计数值和T4xDRH/L 的值相同时A, B, C, 匹配信号产生. PWM 的输出周期由T4PPRH/L (PWM period register), T4xDRH/L (each channel PWM duty register)决定.

PWM Period= [T4PPRH/T4PPRL] X Source Clock

PWM Duty(A-ch) = [T4ADRH/T4ADRL] X Source Clock

Resolution	Frequency		
	T4CK[3:0]=0001 (250ns)	T4CK[3:0]=0010 (500ns)	T4CK[3:0]=0100 (2us)
10-bit	3.9kHz	1.95kHz	0.49kHz
9-bit	7.8kHz	3.9kHz	0.98kHz
8-bit	15.6kHz	7.8kHz	1.95kHz
7-bit	31.2kHz	15.6kHz	3.91kHz

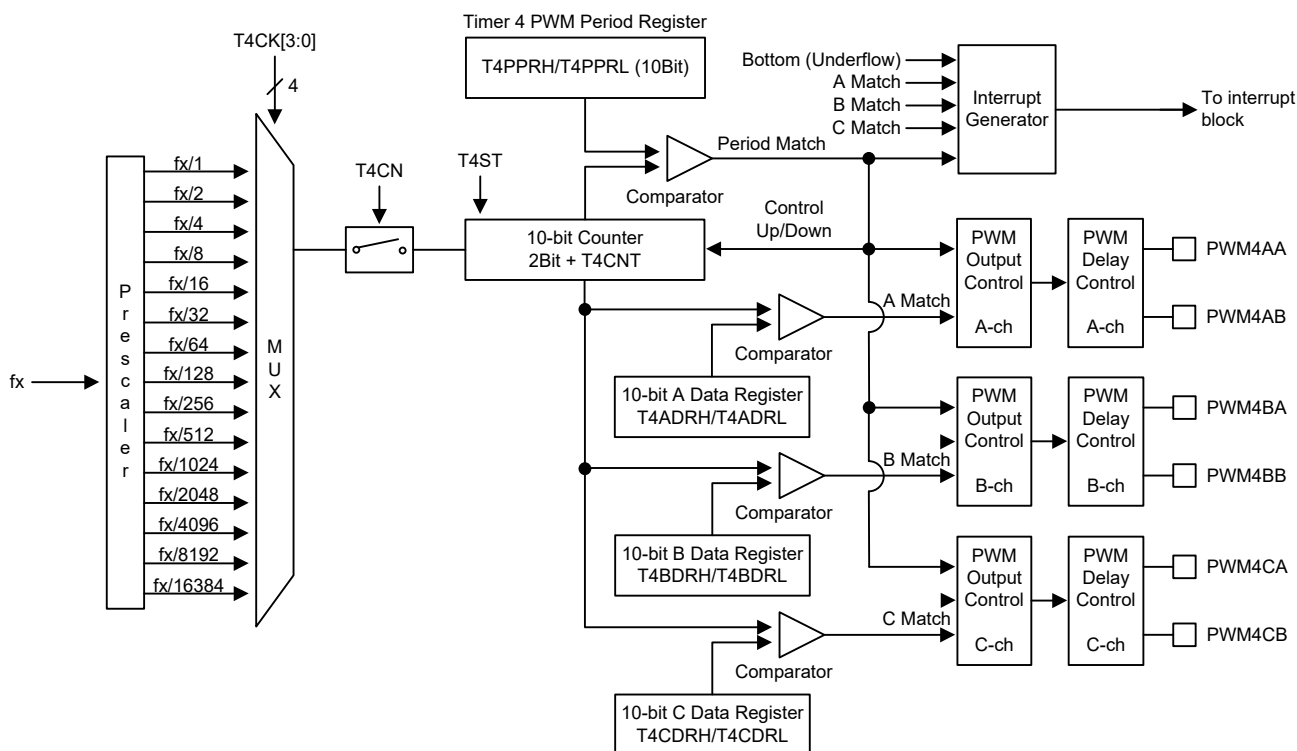
Table 11.12 PWM 周期 vs. 分辨率 (8MHz)

T4PCR3 的 POLxA 位决定占空比极性, 如果占空比值和周期值设置相同, PWM 输出由POLxA (1: High, 0: Low)决定. 如果占空比值设置为"00H", PWM 输出由POLxA (1: Low, 0: High)决定.

PHLT:PxxOE	POLxA	POLBO	POLxB	PWM4xA Pin Output	PWM4xB Pin Output	
0x, x0, 00	0	0	0	Low-level	Low-level	
			1	Low-level	High-level	
		1	x	Low-level	Low-level	
	1	0	0	0	High-level	High-level
				1	High-level	Low-level
		1	x	High-level	High-level	
11	0	x	0	Positive-phase	Positive-Phase	
			1	Positive-phase	Negative-Phase	
	1	x	0	Negative-Phase	Negative-Phase	
			1	Negative-Phase	Positive-phase	

Table 11.13 PWM 通道极性

T4CR	16BIT	T4MS	T4CN	T4ST	T4CK3	T4CK2	T4CK1	T4CK0	ADDRESS:1002H (ESFR) INITIAL VALUE : 0000_0000B
	0	X	X	X	X	X	X	X	
T4PCR1	PWM4E	ESYNC	BMOD	PHLT	UPDT	UALL	NOPS1	NOPS0	ADDRESS:1003H (ESFR) INITIAL VALUE : 0000_0000B
	1	X	X	X	X	X	X	X	
T4PCR2	FORCA	-	PAAOE	PABOE	PBAOE	PBBOE	PCAOE	PCBOE	ADDRESS:1004H (ESFR) INITIAL VALUE : 0000_0000B
	0	-	X	X	X	X	X	X	
T4PCR3	HZCLR	POLBO	POLAA	POLAB	POLBA	POLBB	POLCA	POLCB	ADDRESS:1005H (ESFR) INITIAL VALUE : 0000_0000B
	X	X	X	X	X	X	X	X	

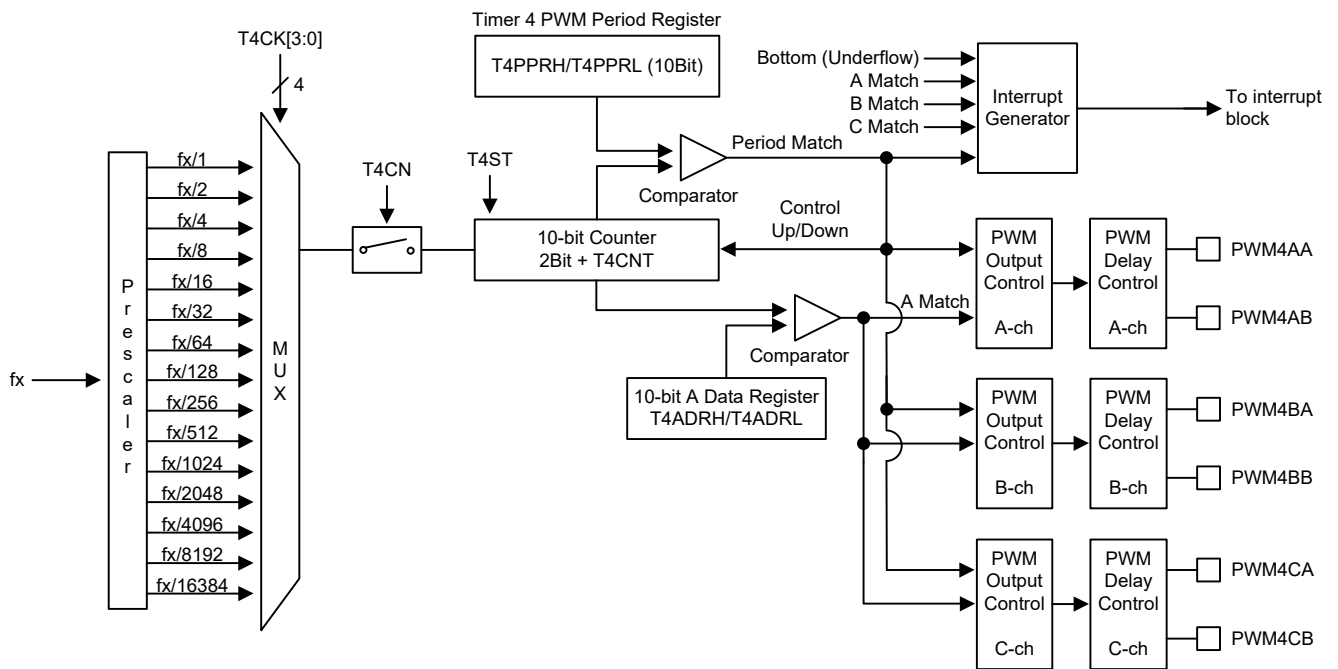


NOTE)

1. Do not set to "1111b" in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.34 10-bit PWM模式 (Force 6-ch)

T4CR	16BIT	T4MS	T4CN	T4ST	T4CK3	T4CK2	T4CK1	T4CK0	ADDRESS:1002H (ESFR)
	0	X	X	X	X	X	X	X	INITIAL VALUE : 0000_0000B
T4PCR1	PWM4E	ESYNC	BMOD	PHLT	UPDT	UALL	NOPS1	NOPS0	ADDRESS:1003H (ESFR)
	1	X	X	X	X	X	X	X	INITIAL VALUE : 0000_0000B
T4PCR2	FORCA	-	PAAOE	PABOE	PBAOE	PBBOE	PCAOE	PCBOE	ADDRESS:1004H (ESFR)
	1	-	X	X	X	X	X	X	INITIAL VALUE : 0000_0000B
T4PCR3	HZCLR	POLBO	POLAA	POLAB	POLBA	POLBB	POLCA	POLCB	ADDRESS:1005H (ESFR)
	X	X	X	X	X	X	X	X	INITIAL VALUE : 0000_0000B



NOTE)

1. Do not set to "1111b" in the T4CK[3:0], when two 8-bit timer 3/4 modes.

Figure 11.35 10-bit PWM 模式 (Force All-ch)

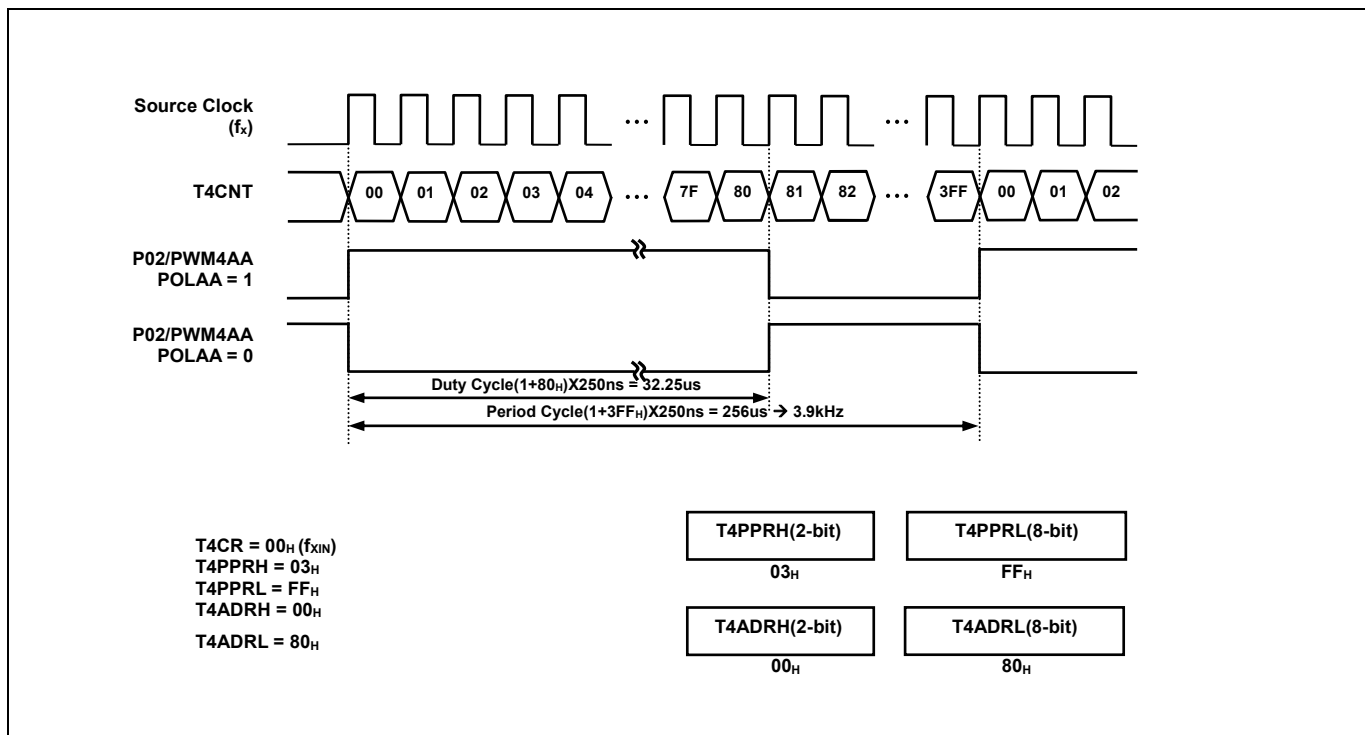


Figure 11.36 PWM 举例 (4 MHz)

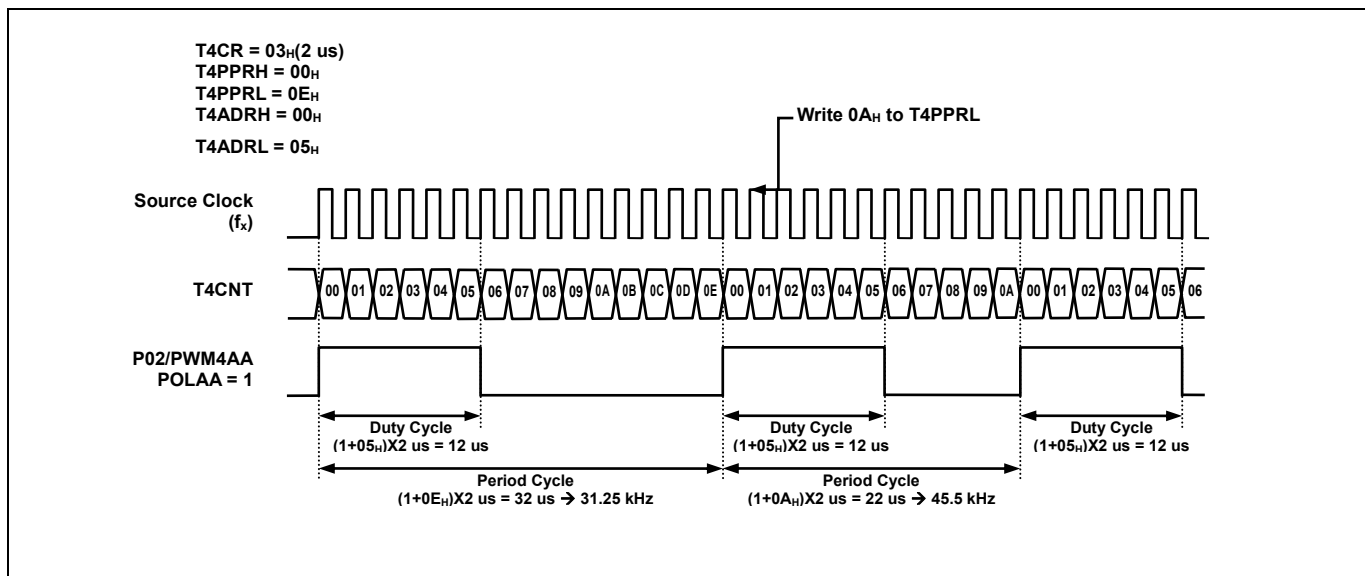


Figure 11.37 绝对占空比条件下改变周期举例 (4 MHz)

同时更新周期 & 占空比寄存器值

当周期匹配时 PWM 的周期和占空比从暂存器转移到 T4PPRH/L (PWM Period Register) 和 T4ADRH/L/T4BDRH/L/T4CDRH/L (PWM Duty Register). 如果你想要周期和占空比立即改变, UPDT bit in the T4PCR1 寄存器的 UPDT 位必须置 '1'. 需要注意的内部时钟时, 数据的传送需要3个时钟周期. 所以更新的数据需要提前3个周期被写入来得到正确的波形.

相位校正& 频率修正

操作PWM时, 通过T4PCR1 寄存器的BMOD(back-to-back 模式) 位会改变相位和频率

在 back-to-back 模式时, PWM 计数器重复高/低计数. 事实上, 有效的占空比和周期变为了寄存器设定值的两倍

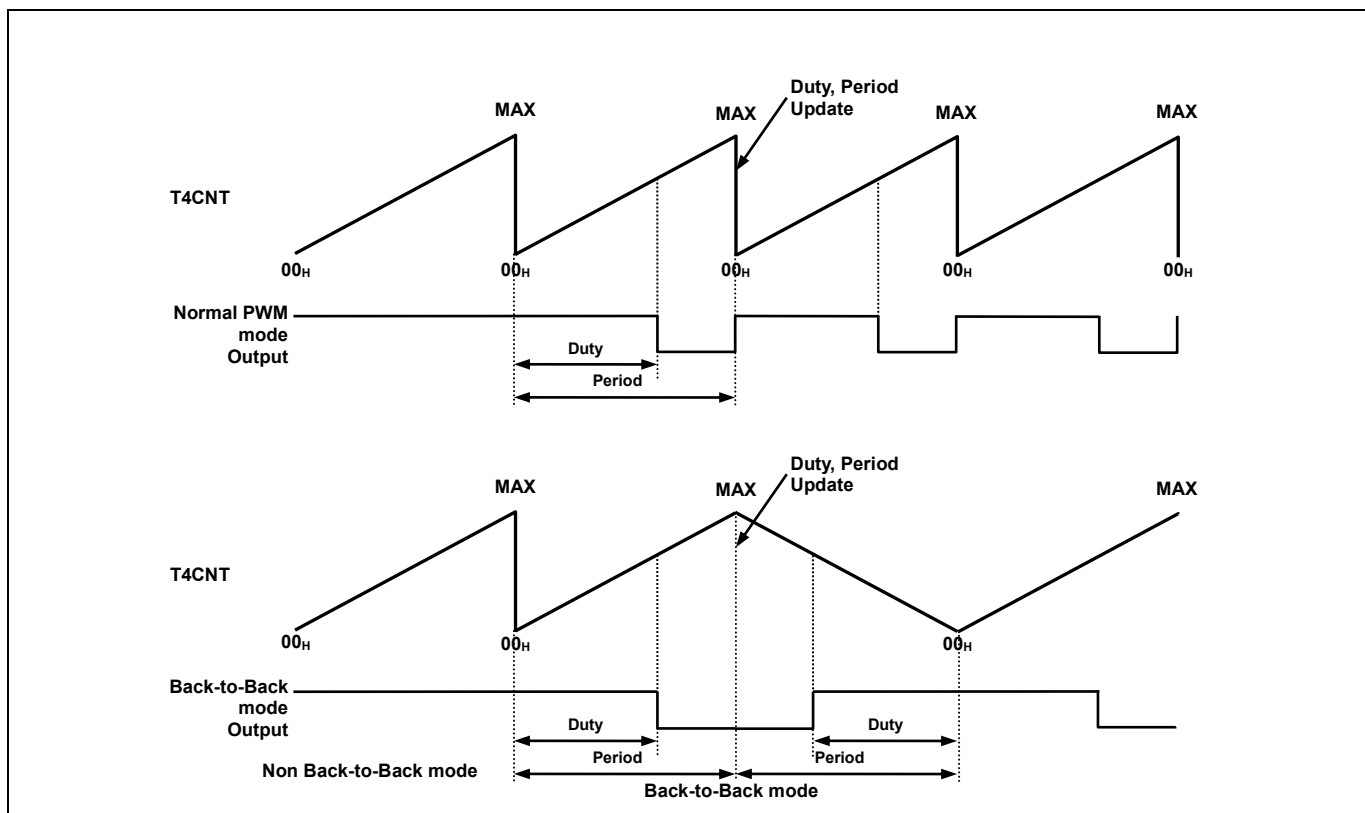


Figure 11.38 PWM 输出波形举例

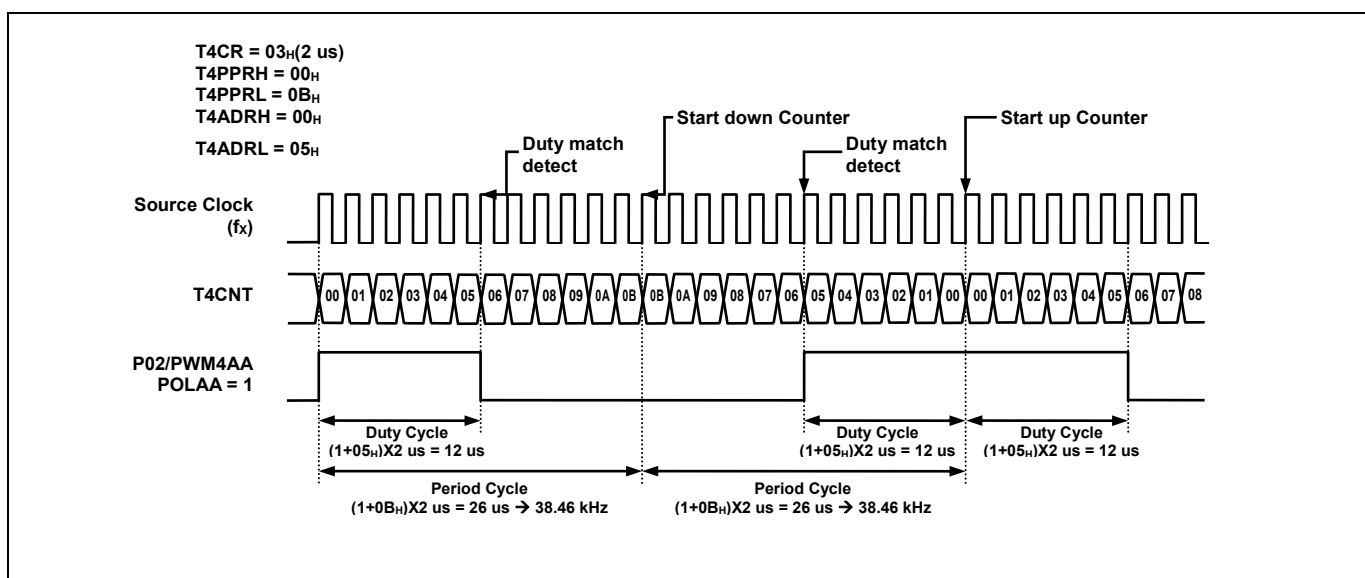


Figure 11.39 PWM 波形举例 (Back-to-Back模式 4 MHz)

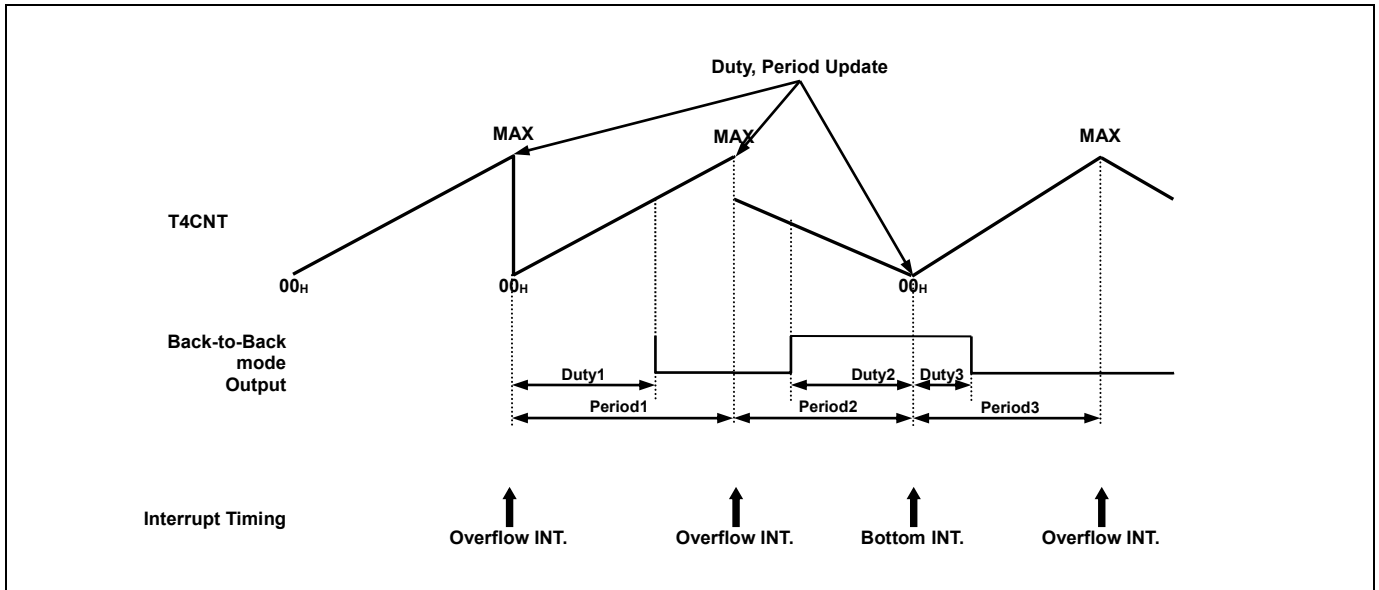


Figure 11.40 PWM相位校正和频率修正

External Sync

如果使用 T4PCR1 寄存器的ESYNC 位, 可以使PWM输出和外部信号同步.

如果 ESYNC 位是 ‘1’, 外部信号通过BLNK引脚转移到PWM 模块. BLNK 信号是低, PWM 输出立即变为复位值, 并且内部计数器开始复位. 如果 BLNK 信号变为 ‘1’, 计数器开始并且PWM 正常输出. (Figure 11.41 referred)

PWM Halt

若使用T4PCR1 寄存器的PHLT 位, 可通过软件停止PWM 工作. 在PHLT 位变为 ‘1’时, PWM 输出变为复位值, 内部计数器复位到 0. 不改变PWM 设置, 暂时停止PWM. 假如 T4CNT, 当停止计数器时, PWM 输出脚在声明前仍保持状态. 但是如果PHLT 位置 ‘1’, PWM 输出脚也复位.

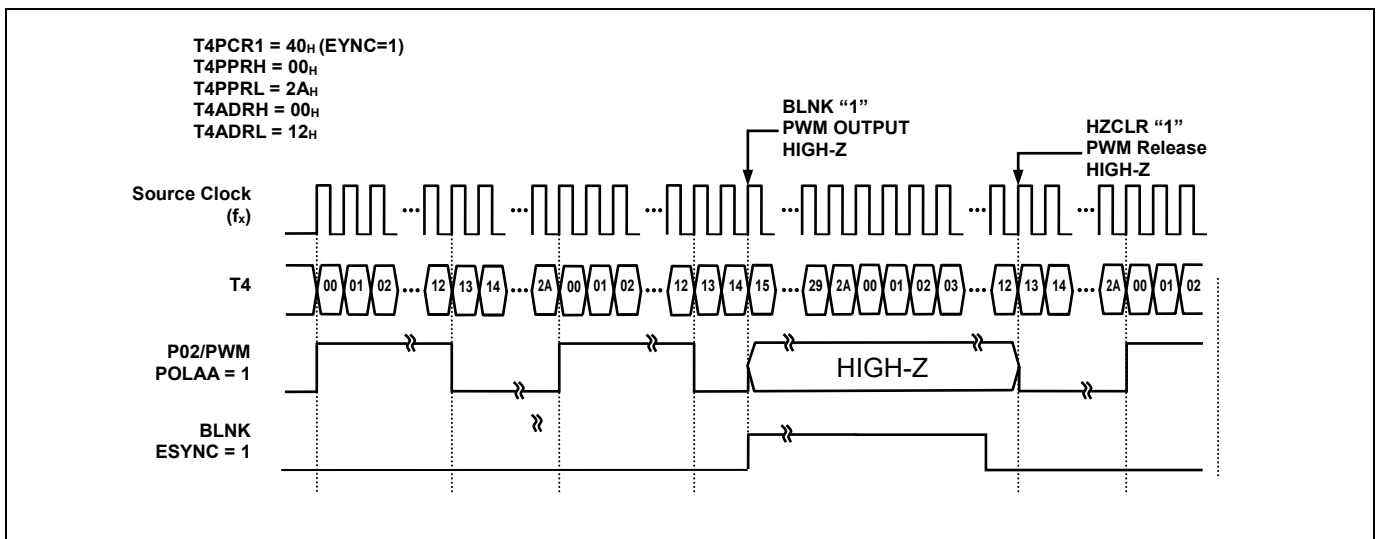


Figure 11.41 具有 BLNK 输入的 PWM 外部同步示例

FORCE Drive ALL Channel with A-ch mode

如果FORCA 位置 '1',通过 A-ch 通道PWM的输出可以打开或禁止所有的PWM输出脚. 需要注意的是A, B, C通道的反向输出和A-ch 输出波形相同.根据 POLAA/BB/CC, 可以控制反向输出.

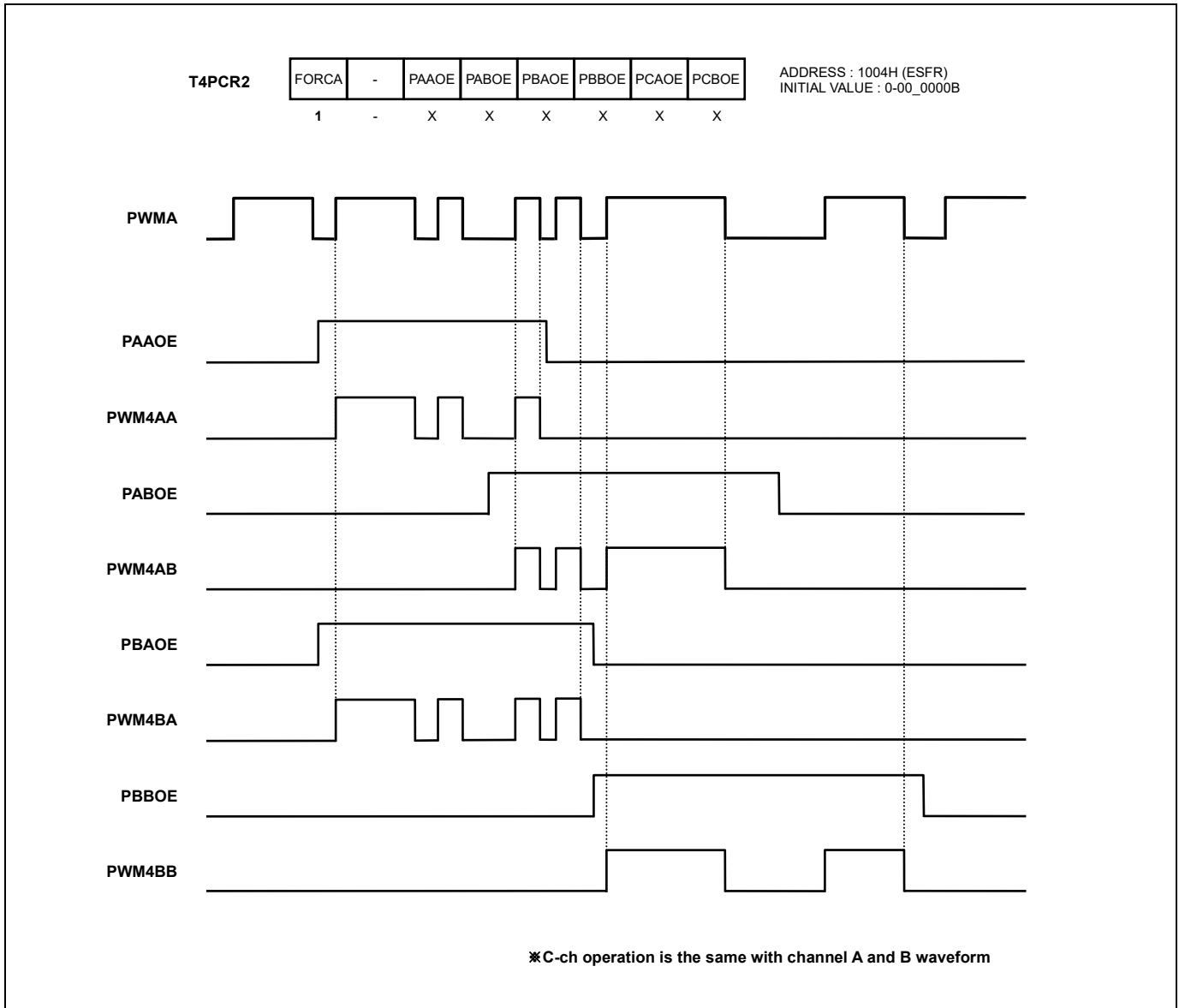


Figure 11.42 强制使用 A-ch 驱动所有通道的示例

FORCE 6-Ch Drive

如果 FORCA 位是 '0', 可以打开或停止PWM 输出脚, 每个通道占空比计数器产生反向输出. 反向输出和PWM输出是反向的. A AA/AB , BA/BB, CA/CB 分别对其进行控制. 如果UALL位是 '1', 当A-channel占空比寄存器被写入值时, 同时更新 B/C 通道的占空比值.

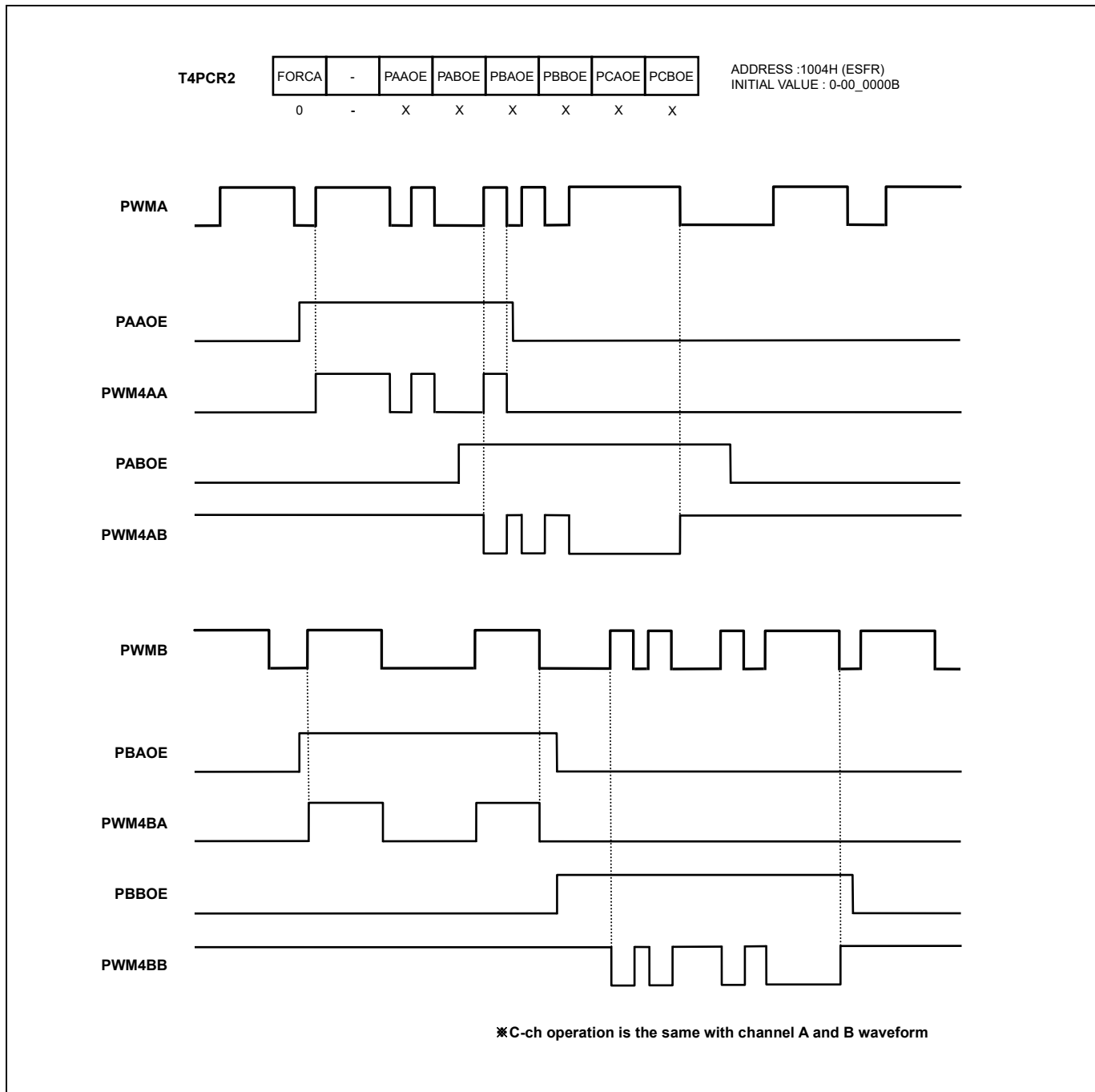


Figure 11.43 Force Drive 6-ch 模式举例

PWM output Delay

如果使用 T4DLYA, T4DLYB, T4DLYC 寄存器, 根据上升沿可以延时PWM 输出. 那时, 不改变下降沿, 所以占空比在延时是减小的. POLAA/BA/CA 位为 '0', 延时被用于下降沿. POLAA/BA/CA 为 '1', 延时用于上升沿. 会产生一对不重叠的时钟. 每个通道可以有4-bit 延时. 最大选择1/8分频使用 NOPS[1:0] 最大延时 128 时钟周期.

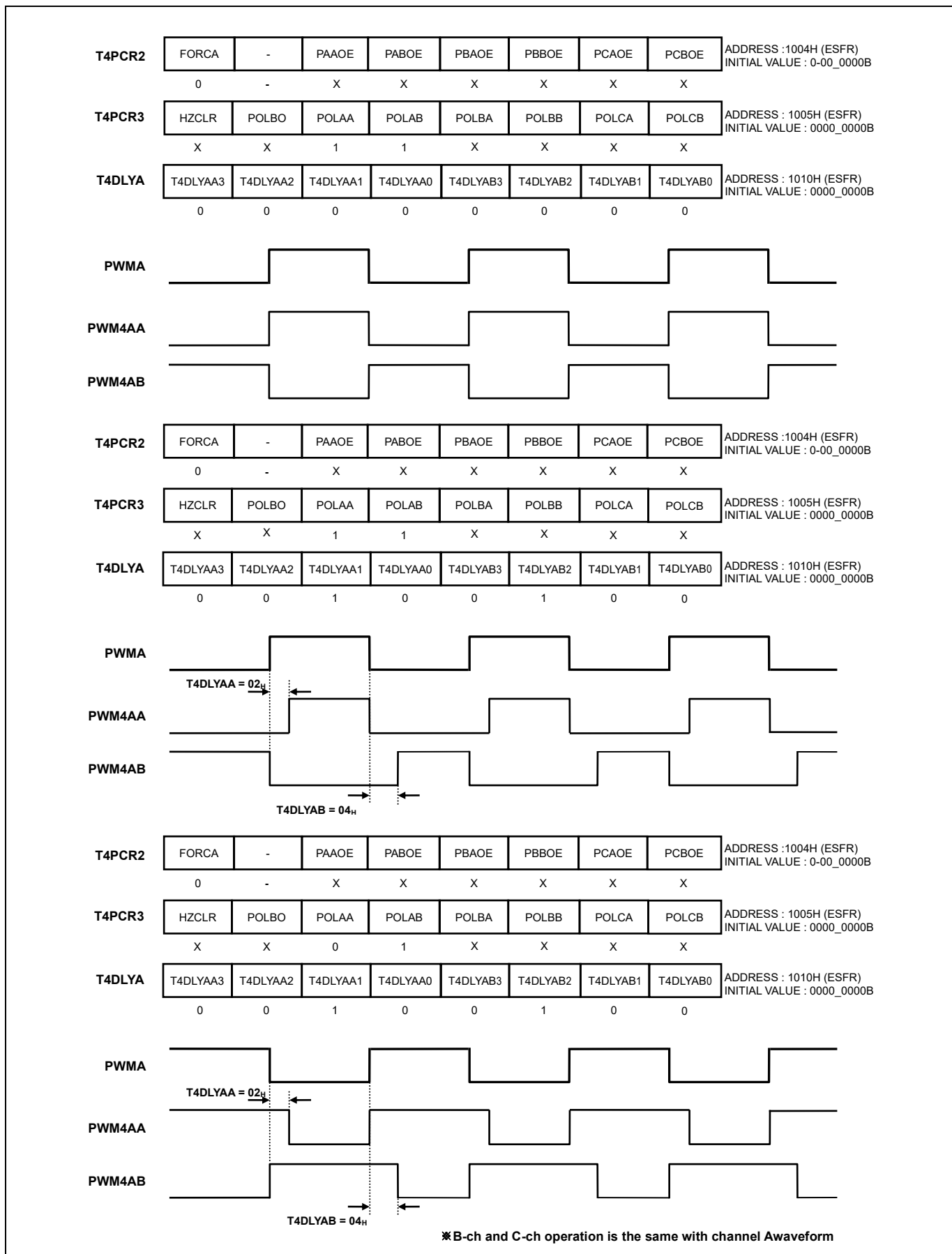


Figure 11.44 PWM Delay 举例

11.8.7 方框图

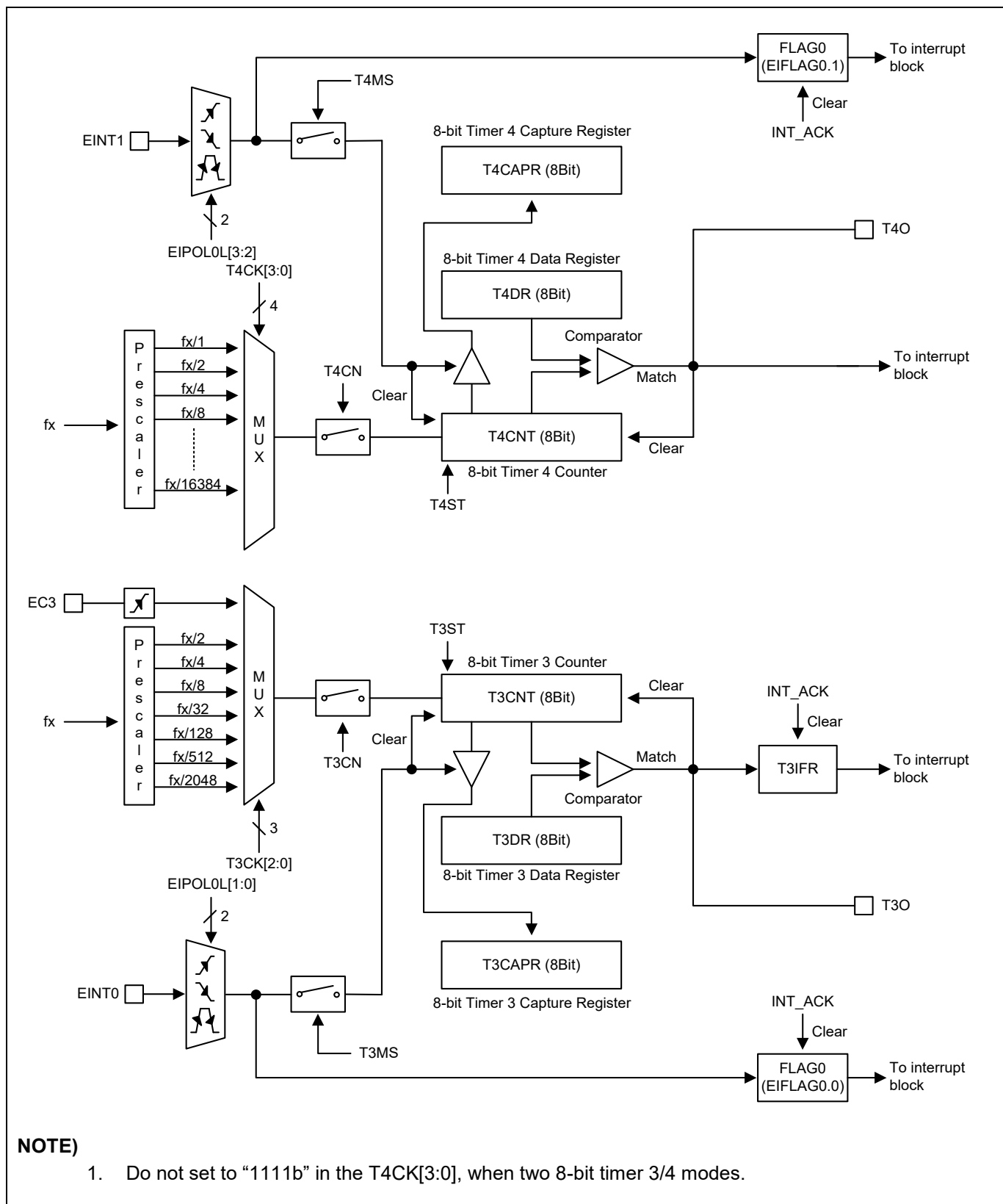


Figure 11.45 两组8-bit定时器3,4 框图

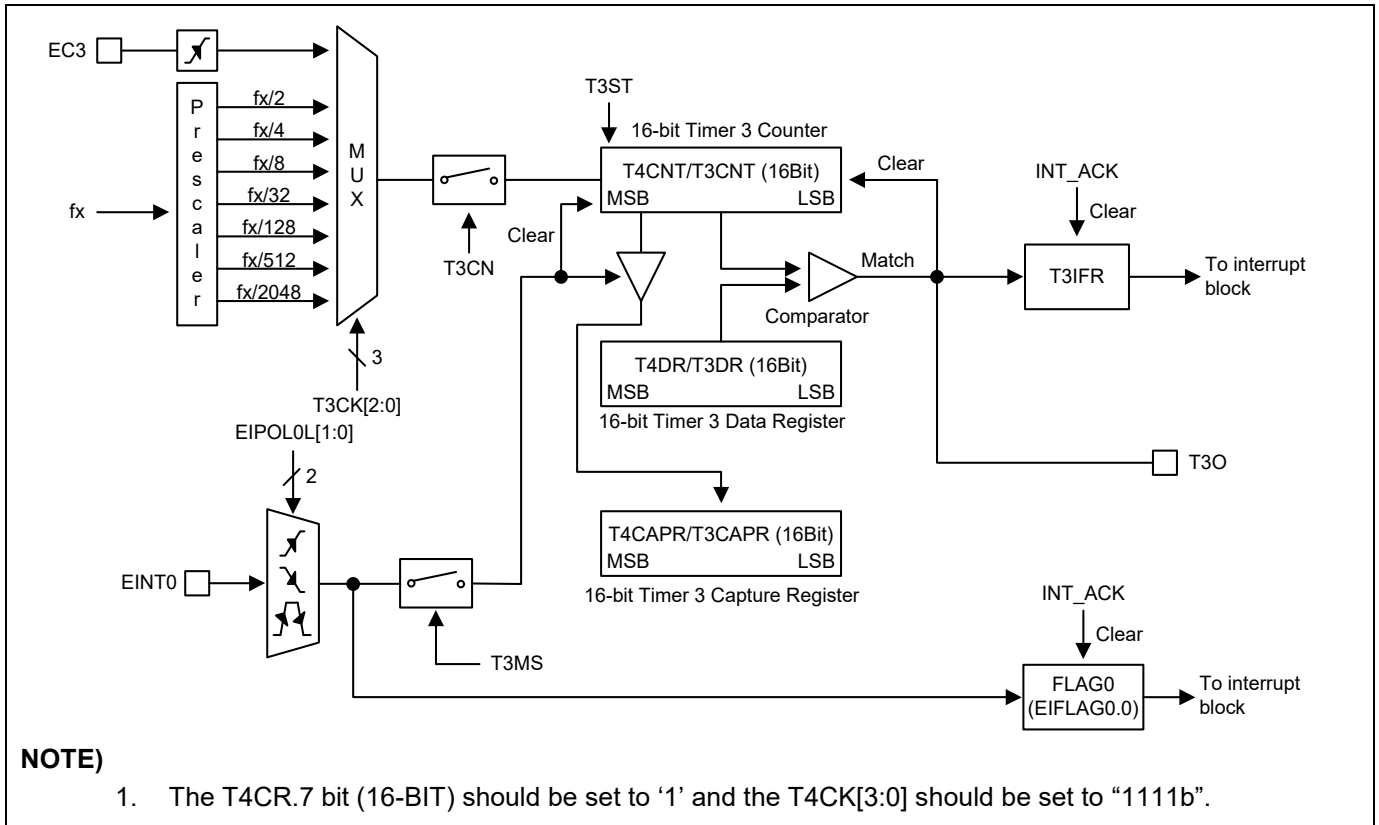


Figure 11.46 16-bit 定时器 3 方框图

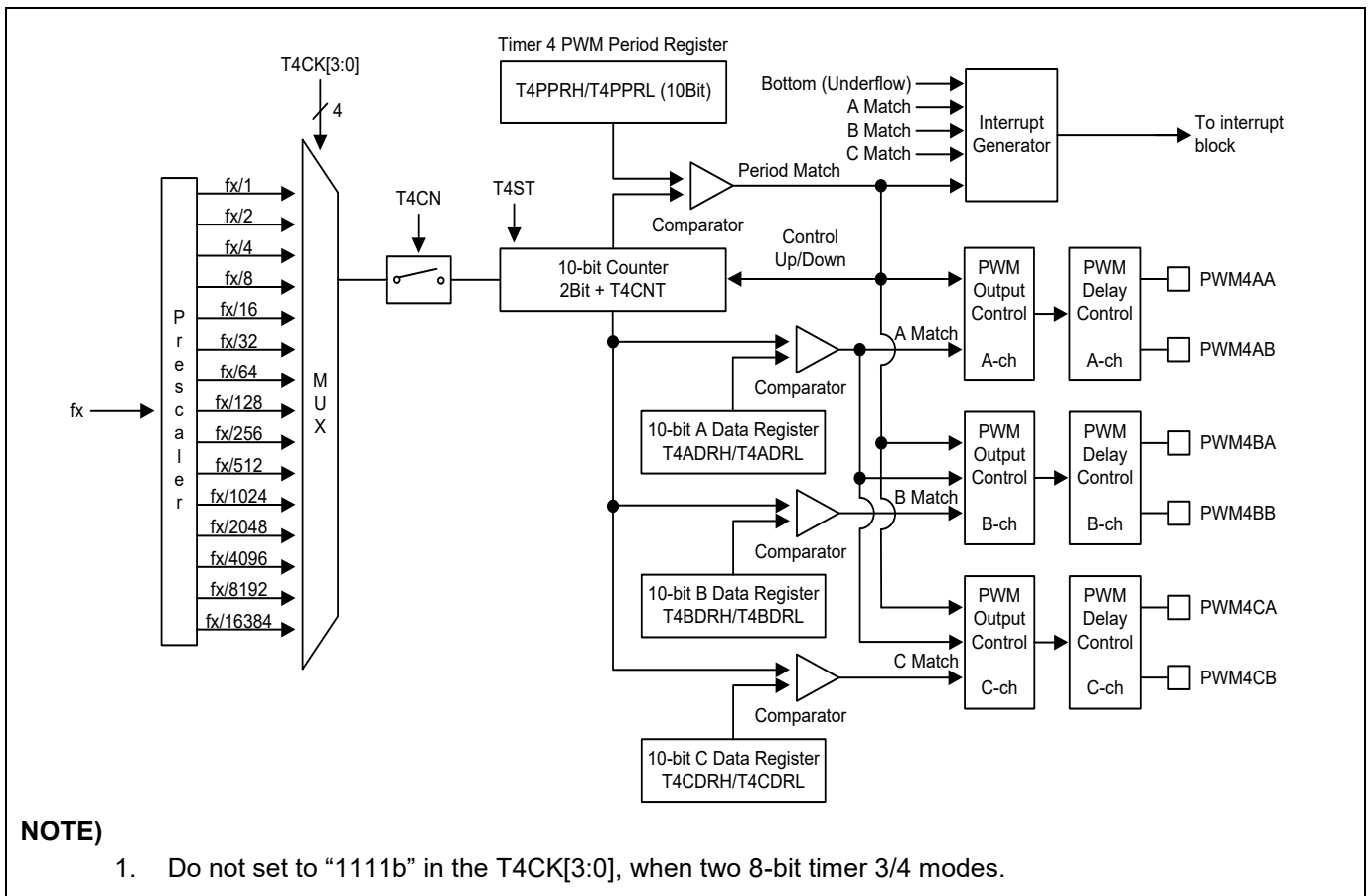


Figure 11.47 定时器4 10-bit PWM 框图

11.8.8 寄存器图

Name	Address	Direction	Default	Description
T3CNT	1001H (ESFR)	R	00H	Timer 3 计数寄存器
T3DR	1001H (ESFR)	W	FFH	Timer 3 数据寄存器
T3CAPR	1001H (ESFR)	R	00H	Timer 3 捕捉数据寄存器
T3CR	1000H (ESFR)	R/W	00H	Timer 3 控制寄存器
T4PPRH	1009H (ESFR)	R/W	00H	Timer 4 PWM 周期高位寄存器
T4PPRL	1008H (ESFR)	R/W	FFH	Timer 4 PWM 周期低位寄存器
T4ADRH	100BH (ESFR)	R/W	00H	Timer 4 PWMA 占空比高位寄存器
T4ADRL	100AH (ESFR)	R/W	7FH	Timer 4 PWMA 占空比低位寄存器
T4BDRH	100DH (ESFR)	R/W	00H	Timer 4 PWM B 占空比高位寄存器
T4BDRL	100CH (ESFR)	R/W	7FH	Timer 4 PWM B 占空比低位寄存器
T4CDRH	100FH (ESFR)	R/W	00H	Timer 4 PWM C 占空比高位寄存器
T4CDRL	100EH (ESFR)	R/W	7FH	Timer 4 PWM C 占空比低位寄存器
T4DLYA	1010H (ESFR)	R/W	00H	Timer 4 PWMA 延时寄存器
T4DLYB	1011H (ESFR)	R/W	00H	Timer 4 PWM B 延时寄存器
T4DLYC	1012H (ESFR)	R/W	00H	Timer 4 PWM C 延时寄存器
T4DR	1013H (ESFR)	R/W	FFH	Timer 4 数据寄存器
T4CAPR	1014H (ESFR)	R	00H	Timer 4 捕捉数据寄存器
T4CNT	1015H (ESFR)	R	00H	Timer 4 计数寄存器
T4CR	1002H (ESFR)	R/W	00H	Timer 4 控制寄存器
T4PCR1	1003H (ESFR)	R/W	00H	Timer 4 PWM 控制寄存器 1
T4PCR2	1004H (ESFR)	R/W	00H	Timer 4 PWM 控制寄存器 2
T4PCR3	1005H (ESFR)	R/W	00H	Timer 4 PWM 控制寄存器 3
T4ISR	1006H (ESFR)	R/W	00H	Timer 4 中断状态寄存器
T4MSK	1007H (ESFR)	R/W	00H	Timer 4 中断屏蔽寄存器

Table 11.14 定时器 3, 4寄存器图

11.8.9 定时器/计数器 3 寄存器说明

定时器3寄存器由 T3CNT, T3DR, T3CAPR, T3CR组成.

11.8.10 定时器/计数器 3 寄存器

T3CNT (定时器 3 计数寄存器: 只读, 只用于定时器模式): 1001H (ESFR)

7	6	5	4	3	2	1	0
T3CNT7	T3CNT6	T3CNT5	T3CNT4	T3CNT3	T3CNT2	T3CNT1	T3CNT0
R	R	R	R	R	R	R	R

Initial value: 00H

T3CNT[7:0] T3 计数

T3DR (定时器 3 数据寄存器: 只写): 1001H (ESFR)

7	6	5	4	3	2	1	0
T3DR7	T3DR6	T3DR5	T3DR4	T3DR3	T3DR2	T3DR1	T3DR0
W	W	W	W	W	W	W	W

Initial value: FFH

T3DR[7:0] T3 数据

T3CAPR (定时器 3 捕捉数据寄存器: 只读, 只用于捕捉模式): 1001H (ESFR)

7	6	5	4	3	2	1	0
T3CAPR7	T3CAPR6	T3CAPR5	T3CAPR4	T3CAPR3	T3CAPR2	T3CAPR1	T3CAPR0
R	R	R	R	R	R	R	R

Initial value: 00H

T3CAPR[7:0] T3 捕获数据

T3CR (定时器 3 控制寄存器): 1000H (ESFR)

7	6	5	4	3	2	1	0
T3EN	–	T3MS	T3CK2	T3CK1	T3CK0	T3CN	T3ST
RW	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

T3EN	控制 Timer 3			
	0	Timer 3 禁止		
	1	Timer 3 使能		
T3MS	控制 Timer 3 工作模式			
	0	定时器/计数器 模式 (T3O: toggle at match)		
	1	捕捉模式 (the match interrupt can occur)		
T3CK[2:0]	Select Timer 3 clock source. fx is main system clock frequency			
	T3CK2	T3CK1	T3CK0	Description
	0	0	0	fx/2
	0	0	1	fx/4
	0	1	0	fx/8
	0	1	1	fx/32
	1	0	0	fx/128
	1	0	1	fx/512
	1	1	0	fx/2048
	1	1	1	外部时钟 (EC3)
T3CN	控制 Timer 3 计数暂停/继续			
	0	暂停计数		
	1	继续计数		
T3ST	控制 Timer 3 启动/停止			
	0	计数器停止		
	1	清除计数器并启动		

NOTE)

1. T3 中断标志位涉及到外部中断标志位寄存器 1 (EIFLAG1).

11.8.11 定时器/计数器 4 计数器说明

定时器/计数器4由T4PPRH/L, T4ADRH/L, T4BDRH/L, T4CDRH/L, T4DLYA, T4DLYB, T4DLYC, T4DR, T4CAPR, T4CNT, T4CR, T4PCR1, T4PCR2, T4PCR3, T4ISR 和 T4MSK 组成

11.8.12 定时器/计数器 4 寄存器

T4PPRH (定时器 4 PWM 周期高位寄存器: 只用于6-ch PWM模式): 1009H (ESFR)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	T4PPRH1	T4PPRH0
-	-	-	-	-	-	RW	RW

Initial value: 00H

T4PPRH[1:0] T4 PWM 周期数据高字节

T4PPRL (定时器 4 PWM 周期低位寄存器: 只用于6-ch PWM模式): 1008H (ESFR)

7	6	5	4	3	2	1	0
T4PPRL7	T4PPRL6	T4PPRL5	T4PPRL4	T4PPRL3	T4PPRL2	T4PPRL1	T4PPRL0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: FFH

T4PPRL[7:0] T4 PWM 周期数据低字节

T4ADRH (定时器 4 PWM A 占空比高位寄存器: 只用于6-ch PWM模式): 100BH (ESFR)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	T4ADRH1	T4ADRH0
-	-	-	-	-	-	RW	RW

Initial value: 00H

T4ADRH[1:0] T4 PWM A 占空比数据高字节

T4ADRL (定时器 4 PWM A 占空比低位寄存器: 只用于6-ch PWM模式): 100AH (ESFR)

7	6	5	4	3	2	1	0
T4ADRL7	T4ADRL6	T4ADRL5	T4ADRL4	T4ADRL3	T4ADRL2	T4ADRL1	T4ADRL0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 7FH

T4ADRL[7:0] T4 PWM A 占空比数据低字节

T4BDRH (定时器 4 PWM B 占空比高位寄存器: 只用于6-ch PWM模式): 100DH (ESFR)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	T4BDRH1	T4BDRH0
–	–	–	–	–	–	RW	RW

Initial value: 00H

T4BDRH[1:0] T4 PWM B 占空比数据高字节

T4BDRL (定时器 4 PWM B 占空比低位寄存器: 只用于6-ch PWM模式): 100CH (ESFR)

7	6	5	4	3	2	1	0
T4BDRL7	T4BDRL6	T4BDRL5	T4BDRL4	T4BDRL3	T4BDRL2	T4BDRL1	T4BDRL0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 7FH

T4BDRL[7:0] T4 PWM B 占空比数据低字节

T4CDRH (定时器 4 PWM C 占空比高位寄存器: 只用于6-ch PWM模式): 100FH (ESFR)

7	6	5	4	3	2	1	0
–	–	–	–	–	–	T4CDRH1	T4CDRH0
–	–	–	–	–	–	RW	RW

Initial value: 00H

T4CDRH[1:0] T4 PWM C 占空比数据高字节

T4CDRL (定时器 4 PWM C 占空比低位寄存器: 只用于6-ch PWM模式): 100EH (ESFR)

7	6	5	4	3	2	1	0
T4CDRL7	T4CDRL6	T4CDRL5	T4CDRL4	T4CDRL3	T4CDRL2	T4CDRL1	T4CDRL0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 7FH

T4CDRL[7:0] T4 PWM C 占空比数据低字节

T4DLYA (定时器 4 PWM A 延时寄存器: 只用于6-ch PWM模式): 1010H (ESFR)

7	6	5	4	3	2	1	0
T4DLYA3	T4DLYA2	T4DLYA1	T4DLYA0	T4DLYAB3	T4DLYAB2	T4DLYAB1	T4DLYAB0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

T4DLYAA[3:0] PWM4AA 延时数据 (仅上升沿)

T4DLYAB[3:0] PWM4AB 延时数据 (仅上升沿)

T4DLYB (定时器 4 PWM B 延时寄存器: 只用于6-ch PWM模式): 1011H (ESFR)

7	6	5	4	3	2	1	0
T4DLYBA3	T4DLYBA2	T4DLYBA1	T4DLYBA0	T4DLYBB3	T4DLYBB2	T4DLYBB1	T4DLYBB0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

T4DLYBA[3:0] PWM4BA 延时数据 (仅上升沿)

T4DLYBB[3:0] PWM4BB 延时数据 (仅上升沿)

T4DLYC (定时器 4 PWM C 延时寄存器 : 只用于6-ch PWM模式): 1012H (ESFR)

7	6	5	4	3	2	1	0
T4DLYCA3	T4DLYCA2	T4DLYCA1	T4DLYCA0	T4DLYCB3	T4DLYCB2	T4DLYCB1	T4DLYCB0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

T4DLYCA[3:0] PWM4CA 延时数据 (仅上升沿)

T4DLYCB[3:0] PWM4CB 延时数据 (仅上升沿)

T4DR (定时器 4 数据寄存器: 用于定时器和捕捉模式): 1013H (ESFR)

7	6	5	4	3	2	1	0
T4DR7	T4DR6	T4DR5	T4DR4	T4DR3	T4DR2	T4DR1	T4DR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: FFH

T4DR[7:0] T4 数据

T4CAPR (定时器 4 捕捉数据寄存器: 只读, 只用于捕捉模式): 1014H (ESFR)

7	6	5	4	3	2	1	0
T4CAPR7	T4CAPR6	T4CAPR5	T4CAPR4	T4CAPR3	T4CAPR2	T4CAPR1	T4CAPR0
R	R	R	R	R	R	R	R

Initial value: 00H

T4CAPR[7:0] T4 捕捉数据

T4CNT (定时器 4 计数器寄存器: 只读, 只用于定时器模式): 1015H (ESFR)

7	6	5	4	3	2	1	0
T4CNT7	T4CNT6	T4CNT5	T4CNT4	T4CNT3	T4CNT2	T4CNT1	T4CNT0
R	R	R	R	R	R	R	R

Initial value: 00H

T4CNT[7:0] T4 计数

T4CR (定时器 4 控制寄存器): 1002H (ESFR)

7	6	5	4	3	2	1	0
16BIT	T4MS	T4CN	T4ST	T4CK3	T4CK2	T4CK1	T4CK0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

16BIT	选择两组 8-bit 或一组 16-bit 模式			
0	两组 8-bit Timer 3/4			
1	一组 16-bit Timer 3			
T4MS	控制 Timer 4 工作模式			
0	定时器/计数器 模式 (T4O: toggle at match)			
1	捕捉模式(the match interrupt can occur)			
T4CN	控制 Timer 4 计数暂停/继续			
0	暂停计数			
1	继续计数			
T4ST	控制 Timer 4 开始/停止			
0	计数停止			
1	清除计数, 重新开始			
T4CK[3:0]	选择 Timer 4 时钟源. fx 是系统主时钟			
T4CK3	T4CK2	T4CK1	T4CK0	Description
0	0	0	0	fx/1
0	0	0	1	fx/2
0	0	1	0	fx/4
0	0	1	1	fx/8
0	1	0	0	fx/16
0	1	0	1	fx/32
0	1	1	0	fx/64
0	1	1	1	fx/128
1	0	0	0	fx/256
1	0	0	1	fx/512
1	0	1	0	fx/1024
1	0	1	1	fx/2048
1	1	0	0	fx/4096
1	1	0	1	fx/8192
1	1	1	0	fx/16384
1	1	1	1	Timer 3 时钟 (only 16-bit Timer 3)

T4PCR1 (定时器 4 PWM 控制寄存器 1): 1003H (ESFR)

7	6	5	4	3	2	1	0
PWM4E	ESYNC	BMOD	PHLT	UPDT	UALL	NOPS1	NOPS0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

- PWM4E** 控制 Timer 4 模式
 - 0 选择 Timer 4 的定时器/计数器或捕捉模式
 - 1 选择 Timer 4 的10-bit PWM 模式
- ESYNC** 用 BLNK 引脚选择外部同步操作
 - 0 禁用外部同步操作
 - 1 启用外部同步操作

(The all PWM4xA/PWM4xB pins are high-impedance outputs on rising edge of the BLNK input pin. Where x= A, B and C)
- BMOD** 控制 Back-to-Back 模式操作
 - 0 禁用 back-to-back 模式 (up count only)
 - 1 启用 back-to-back 模式 (up/down count only)
- PHLT** 控制 Timer 4 PWM 操作
 - 0 运行 10-bit PWM
 - 1 停止 10-bit PWM (计数器保持和输出禁止)
- UPDT** 选择 T4PPR/T4ADR/T4BDR/T4CDR 更新定时器
 - 0 在 T4CNT 和 T4PPR 的周期匹配中更新
 - 1 写入时随时更新
- UALL** 控制更新所有占空比寄存器 (T4ADR/T4BDR/T4CDR)
 - 0 分别写入占空比寄存器
 - 1 通过 Timer 4 PWM 占空比寄存器写入所有占空比寄存器 (T4ADR)
- NOPS[1:0]** 选择重置可预分频

NOPS1	NOPS0	Description
0	0	$f_{PWM}/1$
0	1	$f_{PWM}/2$
1	0	$f_{PWM}/4$
1	1	$f_{PWM}/8$

NOTE)

1. f_{PWM} 是 Timer 4 PWM 的时钟频率.

T4PCR2 (定时器 4 PWM 控制寄存器 2): 1004H (ESFR)

7	6	5	4	3	2	1	0
FORCA	–	PAAOE	PABOE	PBAOE	PBBOE	PCAOE	PCBOE
RW	–	RW	RW	RW	RW	RW	RW

Initial value: 00H

FORCA	控制 PWM 输出模式
0	6-channel 模式 (PWM4xA/PWM4xB 引脚输出分别按照 T4xDR 寄存器. 其中 x = A, B 和 C)
1	强制 A-channel 模式 (所有 PWM4xA/PWM4xB 引脚输出按照只有 T4ADR 寄存器. 其中 x = A, B 和 C)
PAAOE	选择通道 PWM4AA 操作
0	禁用 PWM4AA 输出
1	启用 PWM4AA 输出
PABOE	选择通道 PWM4AB 操作
0	禁用 PWM4AB 输出
1	启用 PWM4AB 输出
PBAOE	选择通道 PWM4BA 操作
0	禁用 PWM4BA 输出
1	启用 PWM4BA 输出
PBBOE	选择通道 PWM4BB 操作
0	禁用 PWM4BB 输出
1	启用 PWM4BB 输出
PCAOE	选择通道 PWM4CA 操作
0	禁用 PWM4CA 输出
1	启用 PWM4CA 输出
PCBOE	选择通道 PWM4CB 操作
0	禁用 PWM4CB 输出
1	启用 PWM4CB 输出

T4PCR3 (定时器 4 PWM 控制寄存器 3): 1005H (ESFR)

7	6	5	4	3	2	1	0
HZCLR	POLBO	POLAA	POLAB	POLBA	POLBB	POLCA	POLCB
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

- HZCLR** 高阻抗输出清除位
 - 0 No effect
 - 1 清除高阻输出
(PWM4xA/PWM4xB引脚返回到输出, 该位自动清零为逻辑'0'. 其中x = A, B 和 C)
- POLBO** 当这些引脚禁用时, 配置 PWM4AB/PWM4BB/PWMCB 通道极性
 - 0 禁止时, 根据极性设置输出这些引脚 (POLAB/POLBB/POLCB 位)
 - 1 这些引脚与 PWM4xA 引脚电平相同, 无论禁用时的极性设置如何 (POLAB/POLBB/POLCB 位, 其中 x = A, B 和 C)
- POLAA** 配置 PWM4AA 通道极性
 - 0 高电平开始 (禁止时该引脚为低电平)
 - 1 低电平启动 (禁止时该引脚为高电平)
- POLAB** 配置 PWM4AB 通道极性
 - 0 PWM4AA 引脚的非反转信号
 - 1 PWM4AA 引脚的反转信号
- POLBA** 配置 PWM4BA 通道极性
 - 0 高电平开始 (禁止时该引脚为低电平)
 - 1 低电平启动 (禁止时该引脚为高电平)
- POLBB** 配置 PWM4BB 通道极性
 - 0 PWM4BA 引脚的非反转信号
 - 1 PWM4BA 引脚的反转信号
- POLCA** 配置 PWM4CA 通道极性
 - 0 高电平开始 (禁止时该引脚为低电平)
 - 1 低电平启动 (禁止时该引脚为高电平)
- POLCB** 配置 PWM4CB 通道极性
 - 0 PWM4CA 引脚的非反转信号
 - 1 PWM4CA 引脚的反转信号

T4ISR (定时器 4 中断状态寄存器): 1006H (ESFR)

7	6	5	4	3	2	1	0
IOVR	IBTM	ICMA	ICMB	ICMC	–	–	–
RW	RW	RW	RW	RW	–	–	–

Initial value: 00H

- IOVR** Timer 4 比较匹配或 Timer 4 溢出中断状态, 向该位写 '0' 以清除
 0 比较匹配或溢出没有发生
 1 比较匹配或溢出发生
- IBTM** Timer 4 底部中断状态, 向该位写 '0' 以清除
 (在 Back-to-Back 模式)
 0 底部没有发生
 1 底部发生
- ICMA** Timer 4 PWM A-ch 匹配中断状态, 向该位写 '0' 以清除
 0 PWM A-ch 匹配没有发生
 1 PWM A-ch 匹配发生
- ICMB** Timer 4 PWM B-ch 匹配中断状态, 向该位写 '0' 以清除
 0 PWM B-ch 匹配没有发生
 1 PWM B-ch 匹配发生
- ICMC** Timer 4 PWM C-ch 匹配中断状态, 向该位写 '0' 以清除
 0 PWM C-ch 匹配没有发生
 1 PWM C-ch 匹配发生

T4MSK (定时器 4 中断屏蔽寄存器): 1007H (ESFR)

7	6	5	4	3	2	1	0
OVRMSK	BTMMSK	CMAMSK	CMBMSK	CMCMSK	–	–	–
RW	RW	RW	RW	RW	–	–	–

Initial value: 00H

- OVRMSK** 控制 Timer 4 比较匹配或溢出中断
 0 禁止比较匹配或溢出中断
 1 启用比较匹配或溢出中断
- BTMMSK** 控制 Timer 4 底部中断
 0 禁用底部中断
 1 启用底部中断
- CMAMSK** 控制 Timer 4 PWM A-ch 匹配中断
 0 禁用 PWM A-ch 匹配中断
 1 启用 PWM A-ch 匹配中断
- CMBMSK** 控制 Timer 4 PWM B-ch 匹配中断
 0 禁用 PWM B-ch 匹配中断
 1 启用 PWM B-ch 匹配中断
- CMCMSK** 控制 Timer 4 PWM C-ch 匹配中断
 0 禁用 PWM C-ch 匹配中断
 1 启用 PWM C-ch 匹配中断

11.9 蜂鸣器驱动

11.9.1 概述

蜂鸣器包括8位计数器, 蜂鸣器数据寄存器(BUZDR), 和蜂鸣器控制寄存器(BUZCR). 方波(61.035Hz~125.0 kHz @8MHz) 从P13/SEG17/AN10/EC1/BUZO 引脚输出. 蜂鸣器数据寄存器 (BUZDR) 控制蜂鸣器频率(如下表所示). 在蜂鸣器控制寄存器(BUZCR)中, BUCK[1:0] 通过分频器选择时钟.

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

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

Table 11.15 蜂鸣器频率 (8 MHz)

11.9.2 方框图

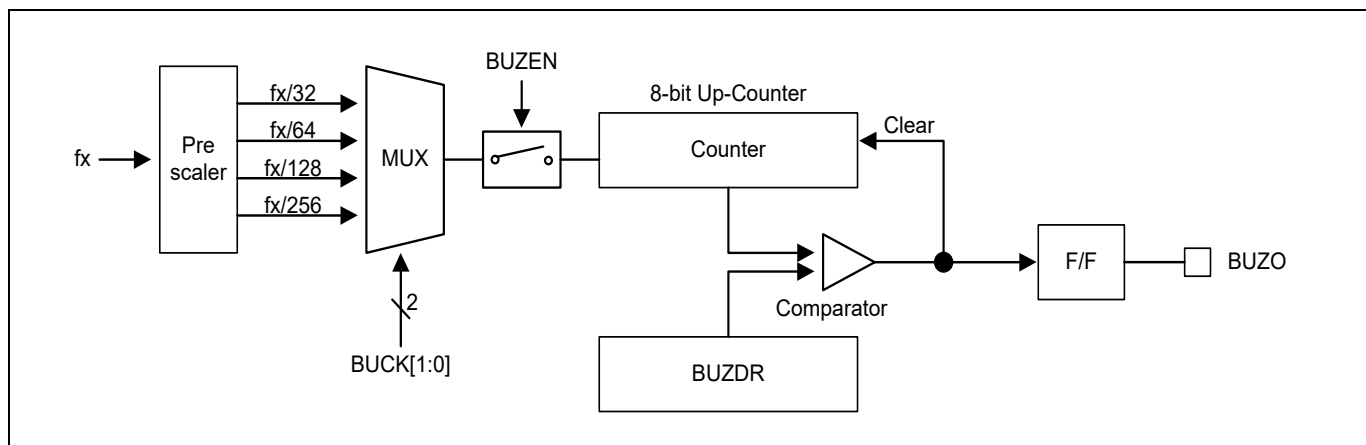


Figure 11.48 蜂鸣器驱动方框图

11.9.3 寄存器图

Name	Address	Direction	Default	Description
BUZDR	8FH	R/W	FFH	蜂鸣器数据寄存器
BUZCR	97H	R/W	00H	蜂鸣器控制寄存器

Table 11.16 蜂鸣器驱动寄存器

11.9.4 蜂鸣器驱动寄存器说明

蜂鸣器驱动由(BUZDR) 和(BUZCR)组成.

11.9.5 蜂鸣器驱动寄存器

BUZDR (蜂鸣器数据寄存器): 8FH

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

Initial value: FFH

BUZDR[7:0] 该位控制蜂鸣器的频率
分辨率为 00H ~ FFH

BUZCR (蜂鸣器控制寄存器): 97H

7	6	5	4	3	2	1	0
-	-	-	-	-	BUCK1	BUCK0	BUZEN
-	-	-	-	-	RW	RW	RW

Initial value: 00H

BUCK[1:0] 蜂鸣器驱动程序源时钟选择

BUCK1	BUCK0	Description
0	0	fx/32
0	1	fx/64
1	0	fx/128
1	1	fx/256

BUZEN 蜂鸣器驱动器操作控制

0	蜂鸣器驱动禁用
1	蜂鸣器驱动启用

NOTE)

1. fx: 系统时钟频率.

11.10 SPI 2

11.10.1 概述

MC96F6432A 配置有串行外围接口(SPI 2). SPI 允许外围串行设备之间同步串行数据传送。通过4跟线(MOSI2, MISO2, SCK2, SS2)可以全双工通信, 支持主机和从机模式, 可选择串行时钟极性, 相位和低位数据优先传送和高位数据优先传送。

11.10.2 方框图

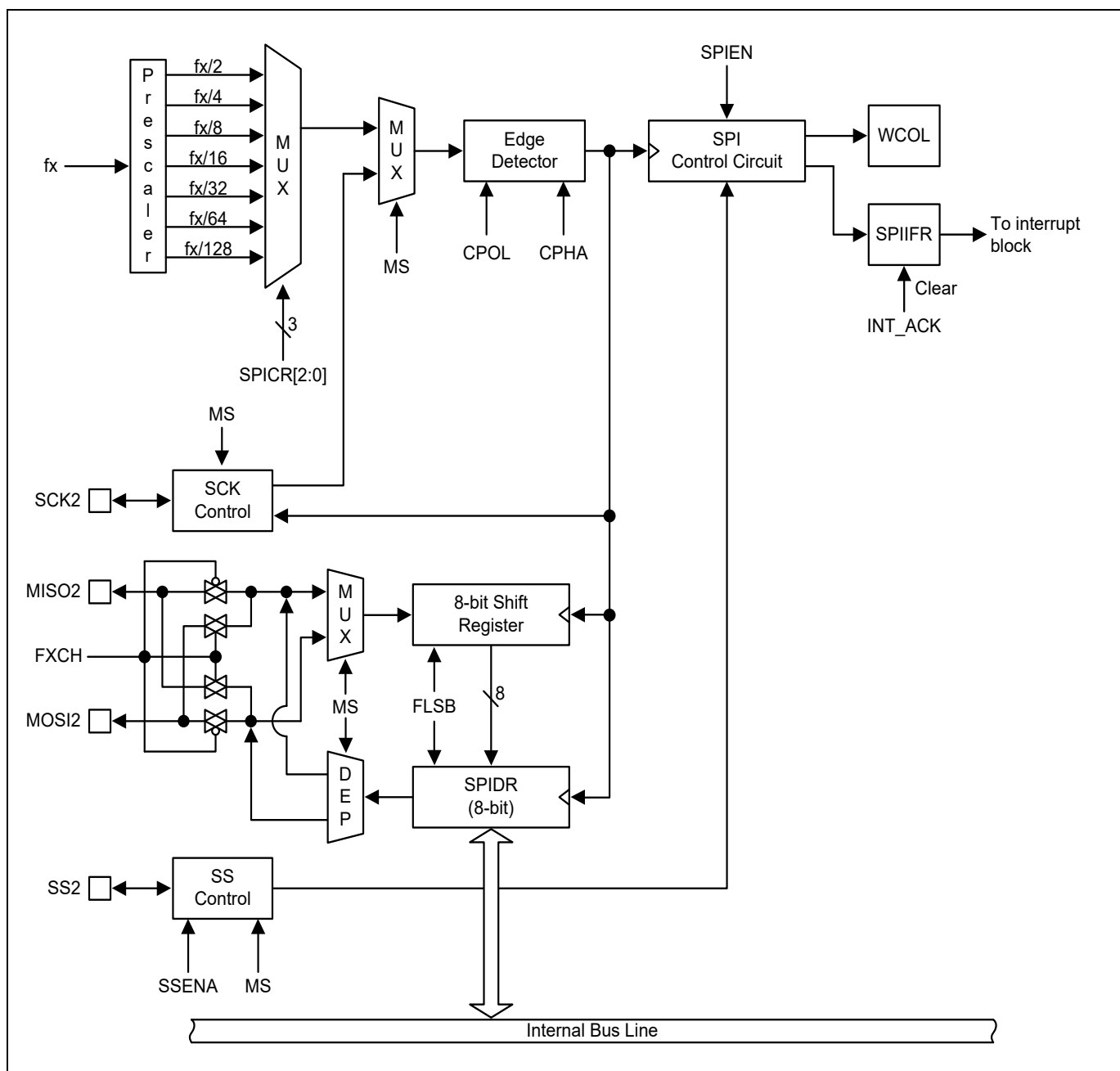


Figure 11.49 SPI 2 方框图

11.10.3 数据发送/接收

用户可以通过以下步骤使用SPI2串口数据发送

1. 通过SPICR控制寄存器选择 SPI2 操作模式 (主动/从动, 极性, 相位).
2. 当 SPI2 配置为主机模式时, 通过SS2信号选择从机(低电平有效)
当 SPI2 配置为从机模式时, 被从主机过来的SS2信号选择
3. 用户写一个字节到数据寄存器SPIDR 时, SPI2 将开始一个操作
4. 这次, 如果SPI2配置为主机, 串行时钟将由SCK2引脚产生. 主机发送8位到从机 (发送), 同时从机发送8位到主机 (接收). 如果是从机模式, 串行时钟将有SCK2引脚产生, 从机发送8位到主机 (发送), 同时主机发送8位到从机 (接收).
5. 当执行发送/接收操作时, SPIIFR 位将会置位. 如果SPI 2中断被使能, 产生一个中断请求. 执行相应的中断时SPIIFR 位通过硬件清零. 如果SPI 2中断禁止, 当用户读取SPISR 状态寄存器时SPIIFR 位被清零, 然后进入 (读/写) 数据寄存器SPIDR.

11.10.4 SS2 引脚功能

1. 当 SPI 2设置为从机模式时, SS2引脚通常是输入状态. 如果低电平信号进入到SS2引脚, SPI 2逻辑运行. 如果高电平信号进入到SS2引脚, SPI2逻辑停止. 这时, SPI2逻辑将会复位, 接收不到数据.
2. 当 SPI 2 设置为主机模式时, 用户可以通过P17IO 选择SS2引脚方向. 如果SS2引脚配置为输出, 用户可以将其设为输出模式. 如果SS2设置为输入模式, 高电平信号必须进入到SS引脚以保障 主机操作. 如果低电平信号进入到SS2引脚, SPI2 逻辑认为另一主控器选择作为从动, 且开始向它发送数据. 为避免总线争用, SPICR 中的MSB位必须清除同时SPI 2变为从机, SPISR中的SPIIFR位置位, 如果SPI 2中断被使能, 将会产生一个中断请求.

注意)

1. 当 SS2 引脚配置为主机模式时, SS2 引脚的输出值由用户软件(P17IO)来规定. 在 SPICR 设置前, SS2 引脚的方向必须先确定
2. 如果你不需要用 SS2 引脚, 清除 SPISR 中的 SSENA 位. 所以, 你可以通过 P17IO 自由使用其开关. 此时, SS2 信号由内部高低电平驱动, 从机时低电平
3. 当 SS 2 引脚作为输入时, 如果高电平信号进入到 SS2 引脚, SS_HIGH 标志位将会置位. 你可以手动写入 0 进行清除.

11.10.5 SPI 2 时序图

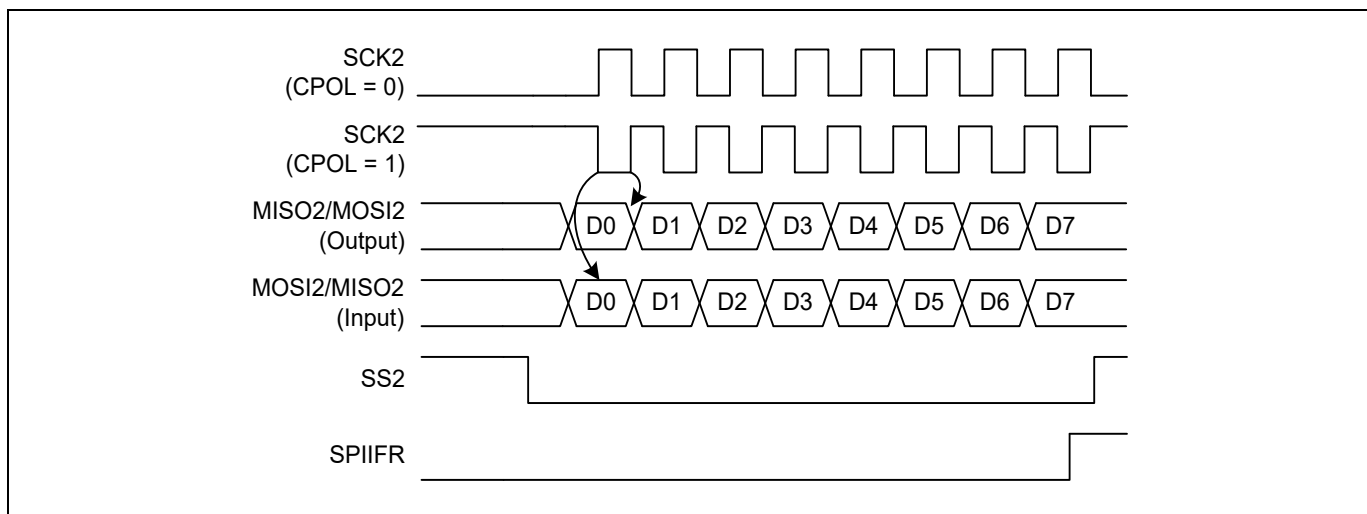


Figure 11.50 SPI 2 发送/接收时序图在CPHA = 0时

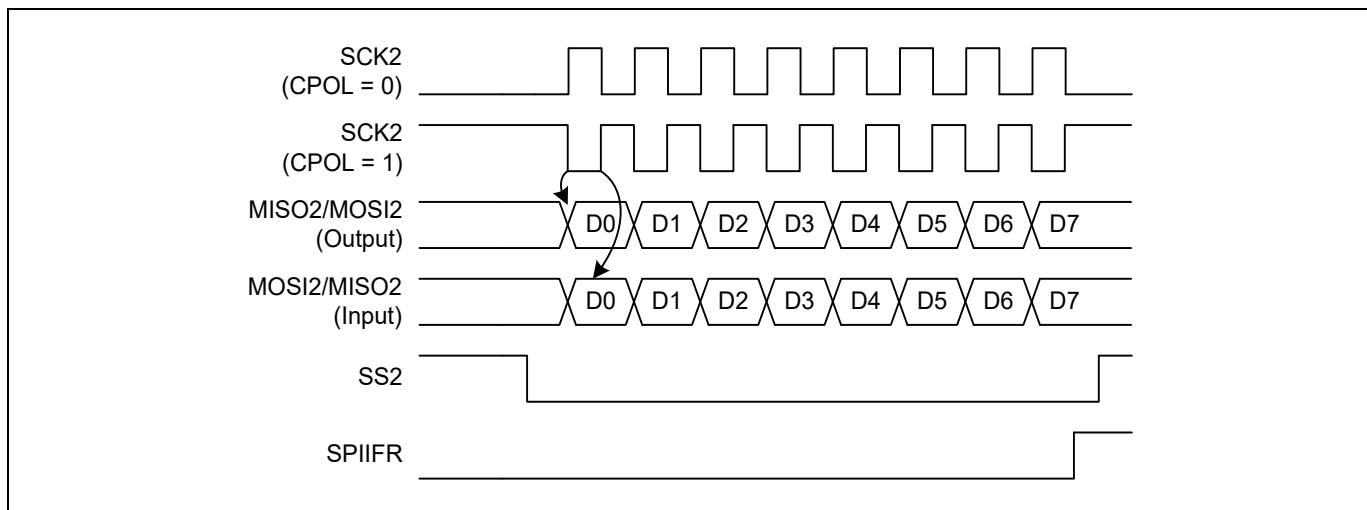


Figure 11.51 SPI 2 发送/接收时序图在CPHA = 1时

11.10.6 寄存器图

Name	Address	Direction	Default	Description
SPISR	B7H	R/W	00H	SPI 2 状态寄存器
SPIDR	B6H	R/W	00H	SPI 2 数据寄存器
SPICR	B5H	R/W	00H	SPI 2 控制寄存器

Table 11.17 SPI 2寄存器图

11.10.7 SPI 2 寄存器说明

SPI 2 寄存器由SPICR, SPISR和SPIDR组成

11.10.8 SPI 2 寄存器

SPIDR (SPI 2 数据寄存器): B6H

7	6	5	4	3	2	1	0
SPIDR7	SPIDR6	SPIDR5	SPIDR4	SPIDR3	SPIDR2	SPIDR1	SPIDR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

SPIDR [7:0]

SPI 2 数据

写入一个字节到这个数据寄存器, SPI 2 将启动一个操作.

SPI SR (SPI 2 状态寄存器): B7H

7	6	5	4	3	2	1	0
SPIIFR	WCOL	SS_HIGH	–	FXCH	SSENA	–	–
RW	R	RW	–	RW	RW	–	–

Initial value: 00H

- SPIIFR** 当 SPI 2 中断产生时, 该位置 ‘1’. 如果 SPI 2 中断使能, 该位被 INT_ACK 信号自动清除. 如果 SPI 2 中断禁止, 在读取SPI SR 状态寄存器时清除该位, 然后开始执行 (读/写). 写入 “1” 无效.

 - 0 SPI 2 中断没有发生
 - 1 SPI 2 中断发生
- WCOL** 在发送时如果有数据写入到数据寄存器SPIDR, 该位置位. 当状态寄存器SPI SR 读取时该位清除, 然后开始操作(读/写) SPIDR

 - 0 No collision
 - 1 Collision
- SS_HIGH** 当 SS2 脚配置为输入状态时, 如果 “HIGH” 脉冲信号输入, 该标志位置位.

 - 0 写入 ‘0’ 时清零
 - 1 写入 ‘1’ 时无效
- FXCH** SPI 2 口功能改变改变控制位

 - 0 No effect
 - 1 更换 MOSI2 和 MISO2 功能
- SSENA** 该位控制 SS2 引脚工作

 - 0 禁止
 - 1 使能 (P17 将作为普通输入)

SPICR (SPI 2控制寄存器): B5H

7	6	5	4	3	2	1	0
SPIEN	FLSB	MS	CPOL	CPHA	DSCR	SCR1	SCR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

SPIEN	该位控制 SPI 2 操作		
	0	禁止 SPI 2 operation	
	1	使能 SPI 2 operation	
FLSB	选择数据传输顺序		
	0	MSB first	
	1	LSB first	
MS	该位选择主控 从动模式		
	0	Slave mode	
	1	Master mode	
CPOL	选择串口时钟 (SCK2) 模式.		
CPHA	在idle模式 CPOL 位决定 SCK2's 值.		
	CPHA 位决定SCK2 在开始还是结束时采样数据.		
	CPOL	CPHA	Leading edge Trailing edge
	0	0	Sample (Rising) Setup (Falling)
	0	1	Setup (Rising) Sample (Falling)
	1	0	Sample (Falling) Setup (Rising)
	1	1	Setup (Falling) Sample (Rising)
DSCR	These three bits select the SCK2 rate of the device configured as a master. Wh		
SCR[2:0]	en DSCR bit is written one, SCK2 will be doubled in master mode.		
	DSCR	SCR1	SCR0 SCK2 frequency
	0	0	0 fx/4
	0	0	1 fx/16
	0	1	0 fx/64
	0	1	1 fx/128
	1	0	0 fx/2
	1	0	1 fx/8
	1	1	0 fx/32
	1	1	1 fx/64

11.11 12-bit A/D 转换器

11.11.1 概述

A/D 可以将一个输入的模拟信号转换为相应的 12-bit 数字信号。A/D 模块有8个模拟输入通道。复用器的输出作为输入，通过逐次逼近法得到近似值。A/D 模块有AD控制高位寄存器(ADCCRH), AD控制低位寄存器(ADCCRL), AD高位数字寄存器(ADCDRH), AD低位数字寄存器(ADC DRL)。ADSEL[3:0]的设置用来选择转换通道。执行AD操作时, TRIG[2:0]位须设置为 'xxx'。ADCDRH 和 ADCDRL 包含AD的转换结果。AD转换完成, 转换结果保存到ADCDRH 和 ADCDRL, AD转换状态位置 '1', AD中断产生。在 AD 转换中, AFLAG 位被读作 '0'

11.11.2 转换时序

AD转换需要4步 (4 个时钟边沿) 对每一位进行转换, 10个时钟设置AD转换器。所以完成AD转换需要58个时钟: 在8MHz 时钟频率下选择 fxx/8 分频时, 一个时钟周期是1 μ s。每位转换需要4个时钟, 转换率计算如下:

$4 \text{ clocks/bit} \times 12 \text{ bits} + \text{set-up time} = 58 \text{ clocks}$,

$58 \text{ clock} \times 1\mu\text{s} = 58\mu\text{s at } 1\text{MHz (} 8\text{MHz}/8)$

注意)

1. A/D 转换时间至少需要 20 μ s。所以你必须设置转换时间大于 20 μ s。

11.11.3 方框图

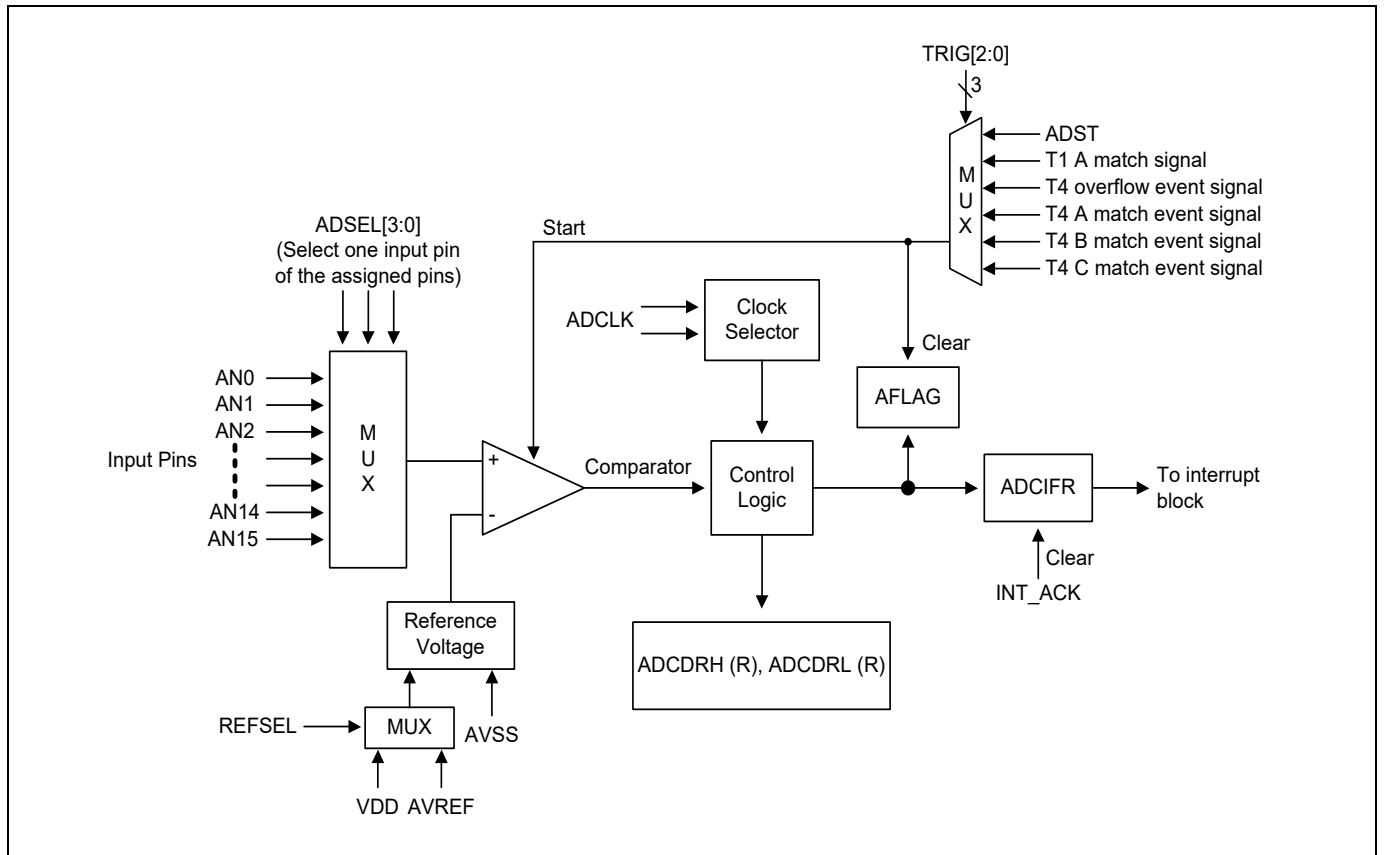


Figure 11.52 12-bit ADC 方框图

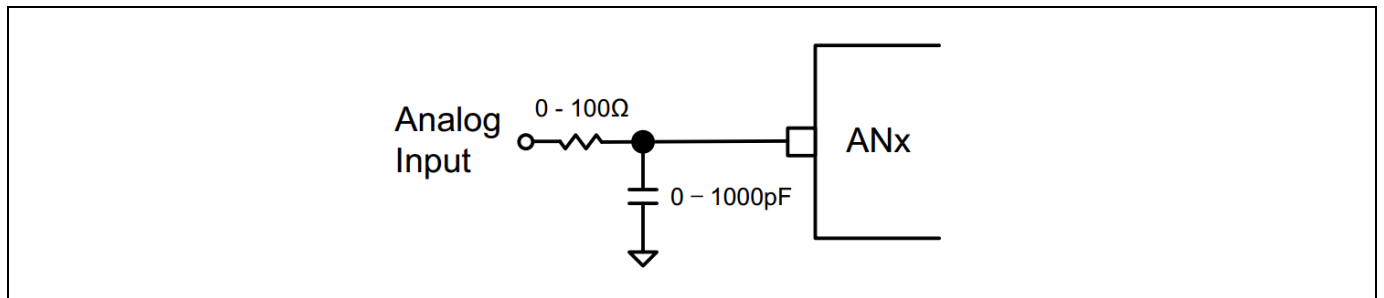


Figure 11.53 A/D模拟输入引脚电容

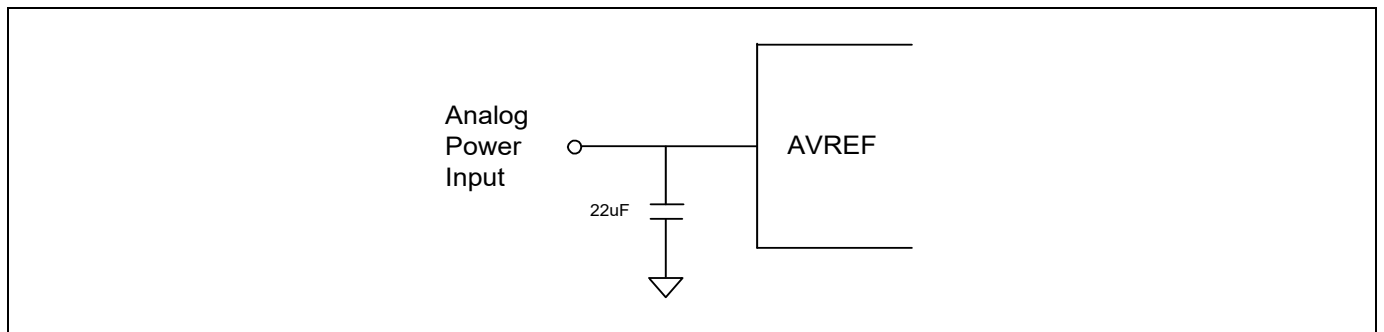


Figure 11.54 A/D Power (AVREF) 引脚电容

11.11.4 ADC 操作

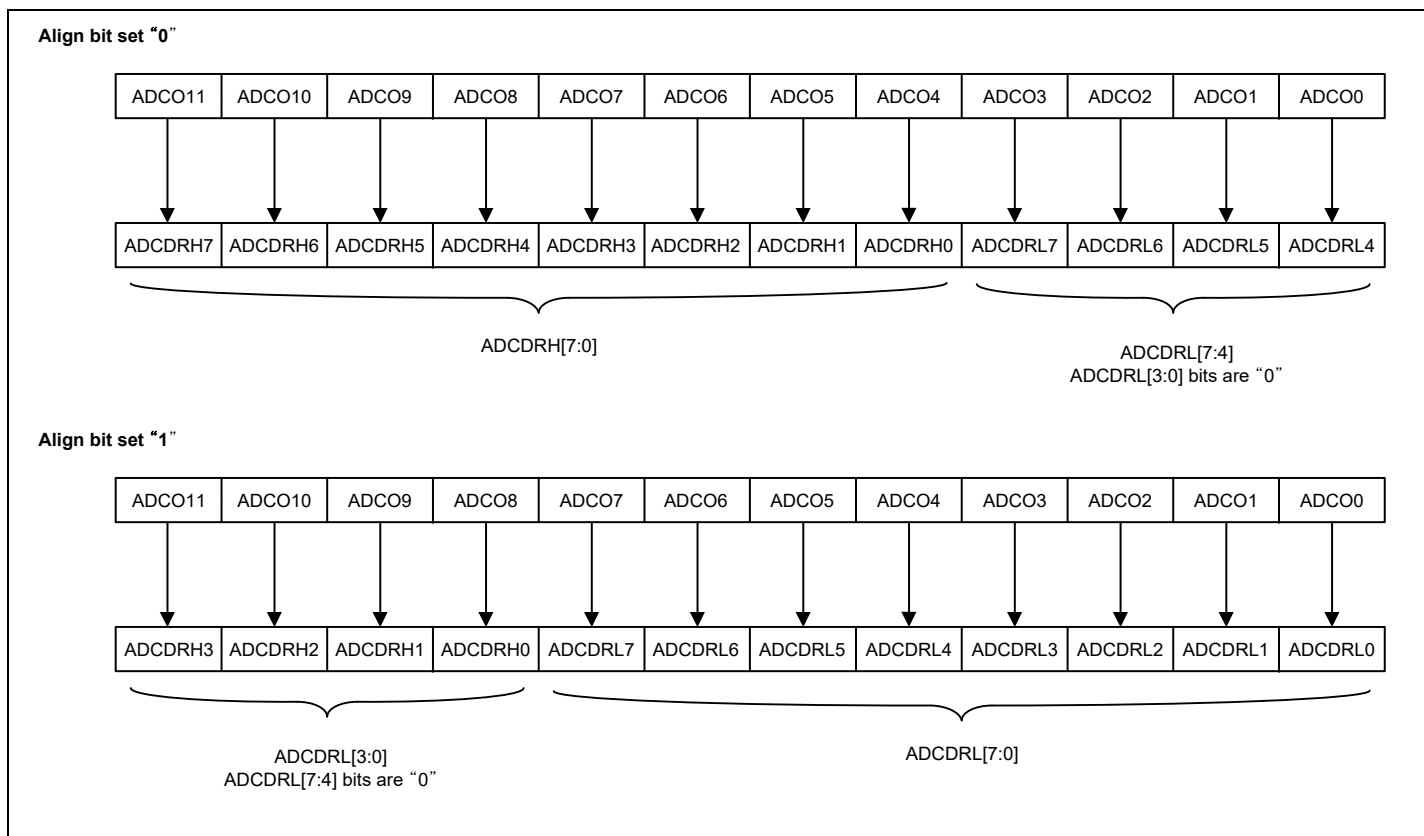


Figure 11.55 用于对齐位的 ADC 操作

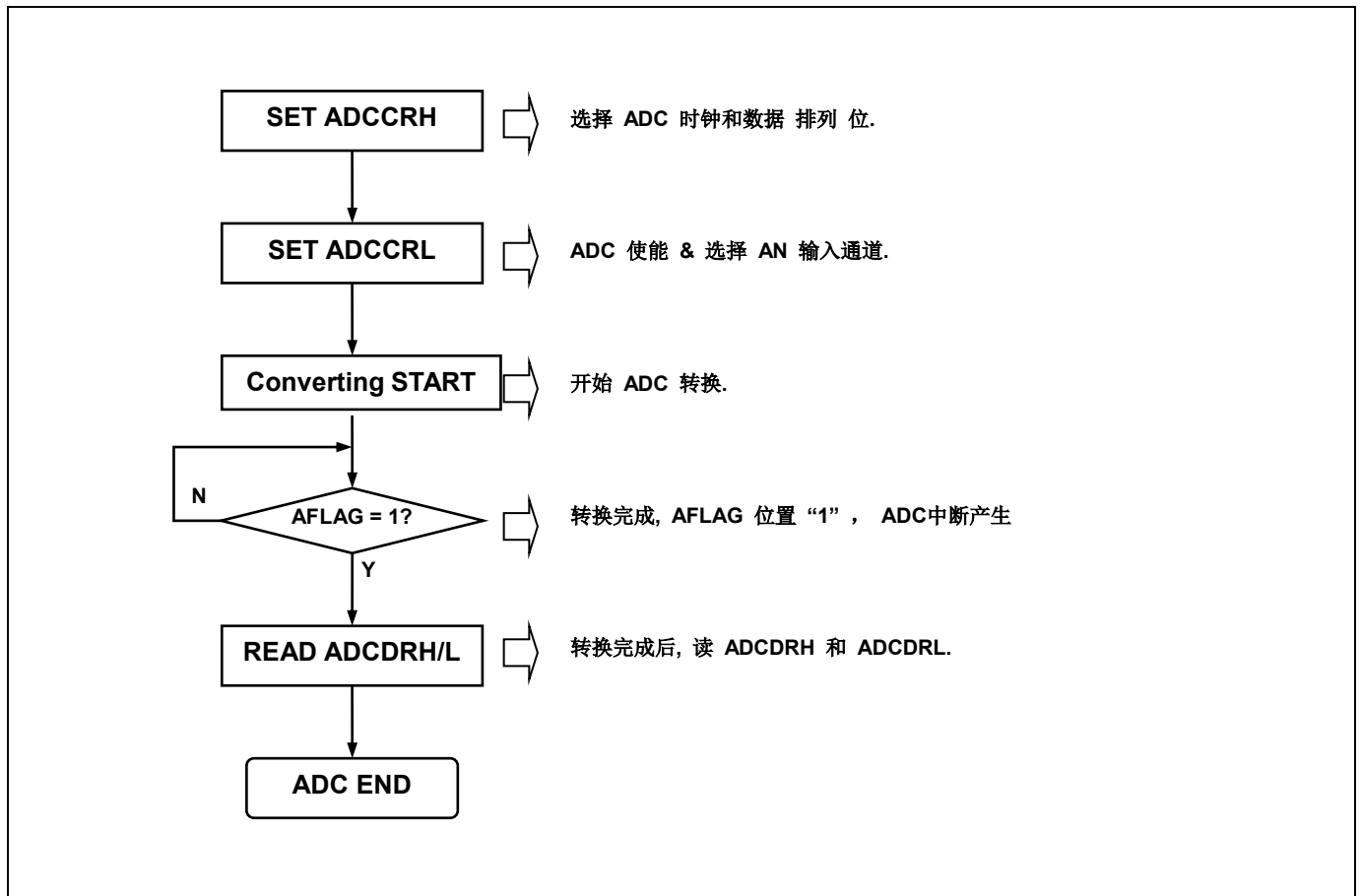


Figure 11.56 A/D 转换操作流程

11.11.5 寄存器图

Name	Address	Direction	Default	Description
ADCDRH	9FH	R	xxH	A/D 换数据高位寄存器
ADCDRL	9EH	R	xxH	A/D 转换数据低位寄存器
ADCCRH	9DH	R/W	00H	A/D 转换控制高位寄存器
ADCCRL	9CH	R/W	00H	A/D 转换控制低位寄存器

Table 11.18 ADC 寄存器

11.11.6 ADC 寄存器说明

ADC 寄存器由ADCDRH, ADCDRL, ADCCRH和ADCCRL组成.

11.11.7 ADC寄存器

ADCDRH (A/D 转换数据高位寄存器): 9FH

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

Initial value: xxH

ADDM[11:4] MSB 对齐, A/D 转换高数据 (8-bit)

ADDL[11:8] LSB 对齐, A/D 转换高数据 (4-bit)

ADCDRL (A/D 转换数据低位寄存器): 9EH

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

Initial value: xxH

ADDM[3:0] MSB 对齐, A/D 转换低数据 (4-bit)

ADDL[7:0] LSB 对齐, A/D 转换低数据 (8-bit)

ADCCRH (A/D 转换控制高位寄存器): 9DH

7	6	5	4	3	2	1	0
ADCIFR	-	TRIG2	TRIG1	TRIG0	ALIGN	CKSEL1	CKSEL0
RW	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

ADCIFR ADC 中断产生时, 该位置'1'. 写入0进行清零或由INT_ACK信号自动清零. 写入“1”无效.

0 没有 ADC 中断产生

1 有 ADC 中断产生

TRIG[2:0] A/D 触发信号选择

TRIG2	TRIG1	TRIG0	Description
0	0	0	ADST
0	0	1	Timer 1 A 匹配信号
0	1	0	Timer 4 溢出事件信号
0	1	1	Timer 4 A 匹配事件信号
1	0	0	Timer 4 B 匹配事件信号
1	0	1	Timer 4 C 匹配事件信号
其他值			未用

ALIGN A/D 转换数据排列选择.

0 MSB 对齐 (ADCDRH[7:0], ADCDRL[7:4])

1 LSB 对齐 (ADCDRH[3:0], ADCDRL[7:0])

CKSEL[1:0] A/D 转换时钟选择

CKSEL1	CKSEL0	Description
0	0	fx/1
0	1	fx/2
1	0	fx/4
1	1	fx/8

ADCCRL (A/D 转换控制低位寄存器): 9CH

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

Initial value: 00H

STBY	控制操作 A/D (睡眠模式时ADC模块自动禁止)				
	0	ADC 模块禁止			
	1	ADC 模块使能			
ADST	控制 A/D 转换 停止/开始.				
	0	No effect			
	1	转换开始触发信号			
REFSEL	A/D 转换参考电压选择				
	0	内部参考电压 (VDD)			
	1	外部参考电压 (AVREF)			
AFLAG	A/D 转换操作状态 (STBY位置 '0' 或CPU 在 STOP 模式时该位清零)				
	0	A/D 转换中			
	1	A/D 转换完成			
ADSEL[3:0]	A/D 转换输入选择				
	ADSEL3	ADSEL2	ADSEL1	ADSEL0	Description
	0	0	0	0	AN0
	0	0	0	1	AN1
	0	0	1	0	AN2
	0	0	1	1	AN3
	0	1	0	0	AN4
	0	1	0	1	AN5
	0	1	1	0	AN6
	0	1	1	1	AN7
	1	0	0	0	AN8
	1	0	0	1	AN9
	1	0	1	0	AN10
	1	0	1	1	AN11
	1	1	0	0	AN12
	1	1	0	1	AN13
	1	1	1	0	AN14
	1	1	1	1	AN15

11.12 USI (USART + SPI + I2C)

11.12.1 Overview

USI 是 USART, SPI 和 I2C的缩写, MC96F6432A 有两组USI 功能模块, USI0 和 USI1 功能完全相同. 每个 USI 由USI 控制寄存器1/2/3/4, USI 状态寄存器 1/2, USI波特率发生寄存器, USI 数据寄存器, USI SDA 维持时间寄存器, USI SCL 高电平周期寄存器, USI SCL 低电平周期寄存器, 和 USI 从动地址寄存器 (USInCR1, USInCR2, USInCR3, USInCR4, USInST1, USInST2, USInBD, USInDR, USInSDHR, USInSCHR, USInSCLR, USInSAR) 组成 n 是 0 或 1.

工作模式通过 USIn 选择位(USIxMS[1:0])选择.

有四个工作模式:

- 异步模式 (UART)
- 同步模式 (USART)
- SPI 模式
- I2C 模式

11.12.2 USIn UART 模式

USART 是高度灵活的串口通讯设备. 主要特性列举如下.

- 同时双向操作 (独立串行接收和发送寄存器)
- 同步或异步操作
- 波特率发生器
- 支持串口帧数为5, 6, 7, 8, 或 9 的数据位和 1 或2 个停止位
- 硬件支持奇偶校验的产生和奇偶检查
- 数据超出检测
- 帧差错检测
- 三个单独的中断, TX 完成, TX 数据寄存器空和 RX 完成
- 双速异步通讯模式

USIn 主要有时钟发生器, 发送器和接收器. 时钟发生逻辑包括外部时钟输入同步逻辑或SPI从动操作, 和波特率发生器或主机操作. 发送器包括写信号缓存器, 一个串口移位寄存器, 奇偶发生器和操作不同帧格式的控制逻辑.

写入暂存器允许两帧数据之间没有任何延时的连续发送. 接收器的时钟和数据接收装置是UART模块最复杂的部分, 接收装置用作异步数据接收. 除了接收装置以外, 接收器还包括一个奇偶校验器, 一个移位寄存器, 一个two-level receive FIFO (USIxDR)和控制逻辑. 接收器支持和发送器时相同的帧格式, 并可以检测到帧差错, 数据超出和奇偶错误.

11.12.3 USn UART 框图

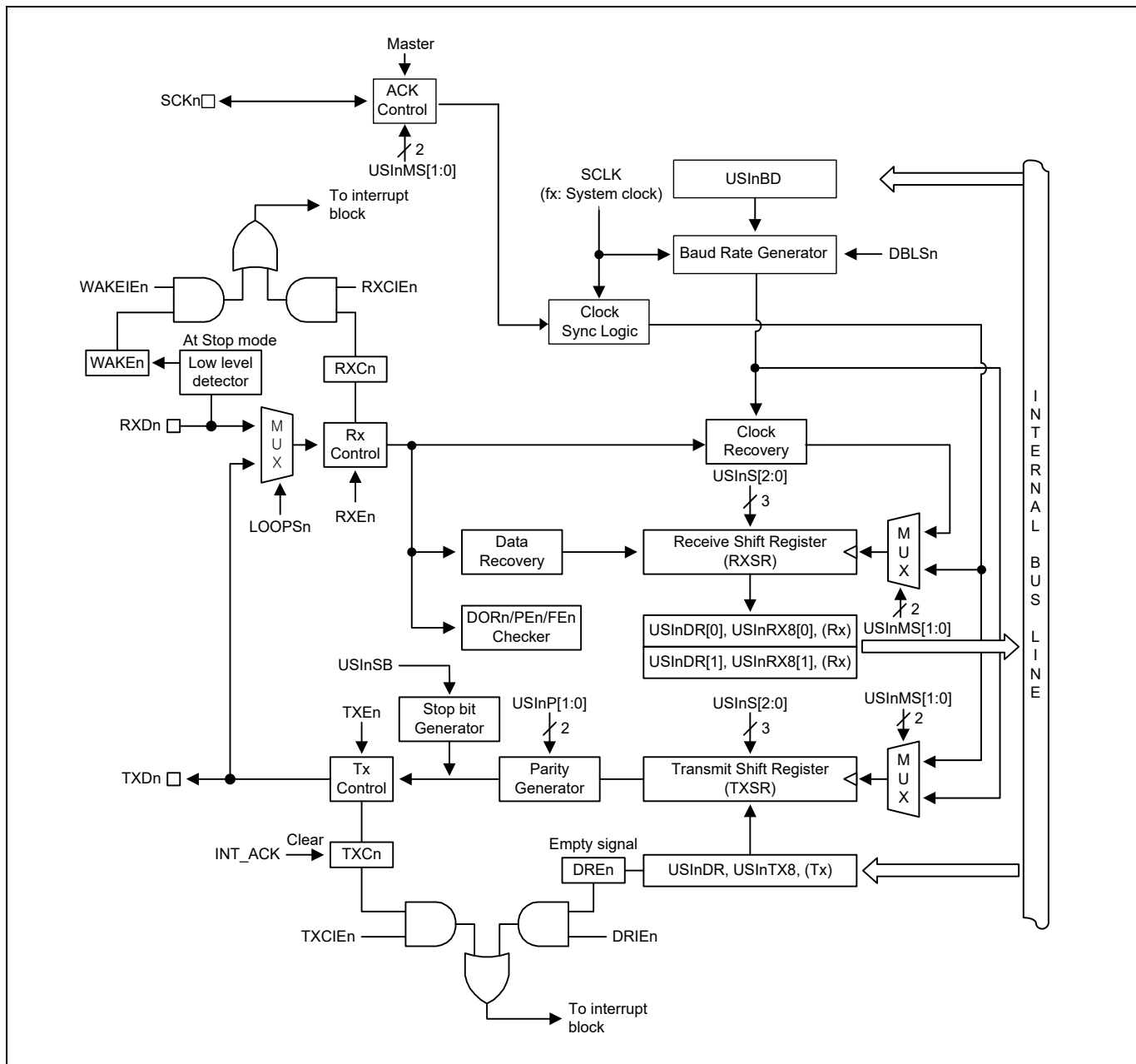


Figure 11.57 USn USART 框图

11.12.4 USIn 时钟发生器

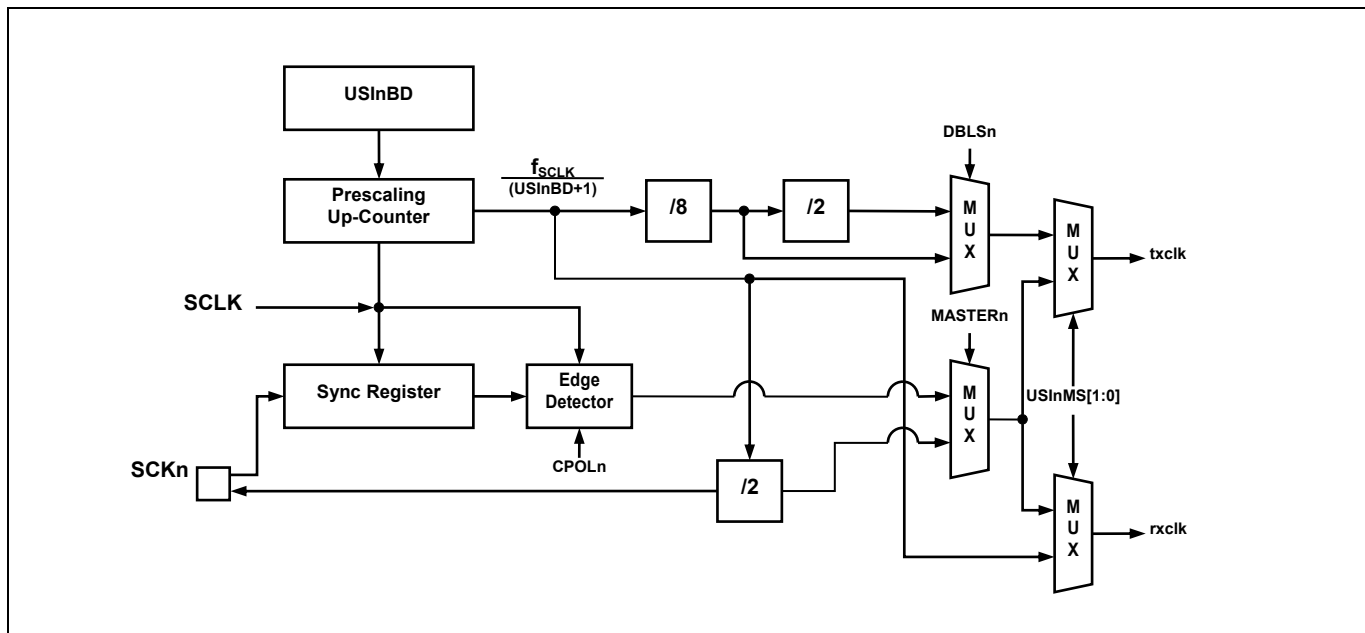


Figure 11.58 时钟生成方框图 (USIn)

时钟发生器逻辑为发送器和接收器产生基本时钟. USIn 支持4个时钟发生模式: 普通异步, 双速异步, 主机同步模式, 从机同步模式. USInCR1寄存器的USInMS[1:0] 位选择同步还是异步操作. 异步双速模式通过USInCR2 寄存器的DBLS0位控制. USInCR3 寄存器的MASTER0位控制时钟源是内部(主机模式, 输出脚)或者外部(从机, 输入脚). 只有当USIn 工作在同步或 SPI 模式时SCK0 引脚是可用的.

下表所示为波特率的计算公式(in bps).

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (DBLS0=0)	$\text{Baud Rate} = \frac{f_x}{16(\text{USInBD} + 1)}$
Asynchronous Double Speed Mode (DBLS0=1)	$\text{Baud Rate} = \frac{f_x}{8(\text{USInBD} + 1)}$
Synchronous or SPI Master Mode	$\text{Baud Rate} = \frac{f_x}{2(\text{USInBD} + 1)}$

Table 11.19 波特率设置计算公式

11.12.5 USIn 外部时钟 (SCK0)

外部时钟使用同步方式.

SCK0脚外部时钟输入通过同步逻辑采样来达到稳定, 在被发生器或接受器使用之前同步逻辑输出必须穿过边沿检测器。该流程介绍了两个CPU时钟周期延迟, 最大频率达到1MHz.

11.12.6 USIn 同步模式

使用同步或SPI 模式时, SCK0 脚将被用作时钟输入(slave)或时钟输出(master). SCK0数据采样和发送发生在时钟的不同边沿, 例如, 如果数据输入脚RXD0 (MISO0 in SPI mode)在 SCK0 的上升沿采样, 数据输出脚TXD0 (MOSI0 in SPI mode) 在下降沿改变.

USInCR1 寄存器的CPOLn 用于选择数据采样和数据更改的SCKn 时钟沿. 如下表所示, 当 CPOLn 为0时, 数据在SCKn 上升沿进行改变在下降沿进行采样.

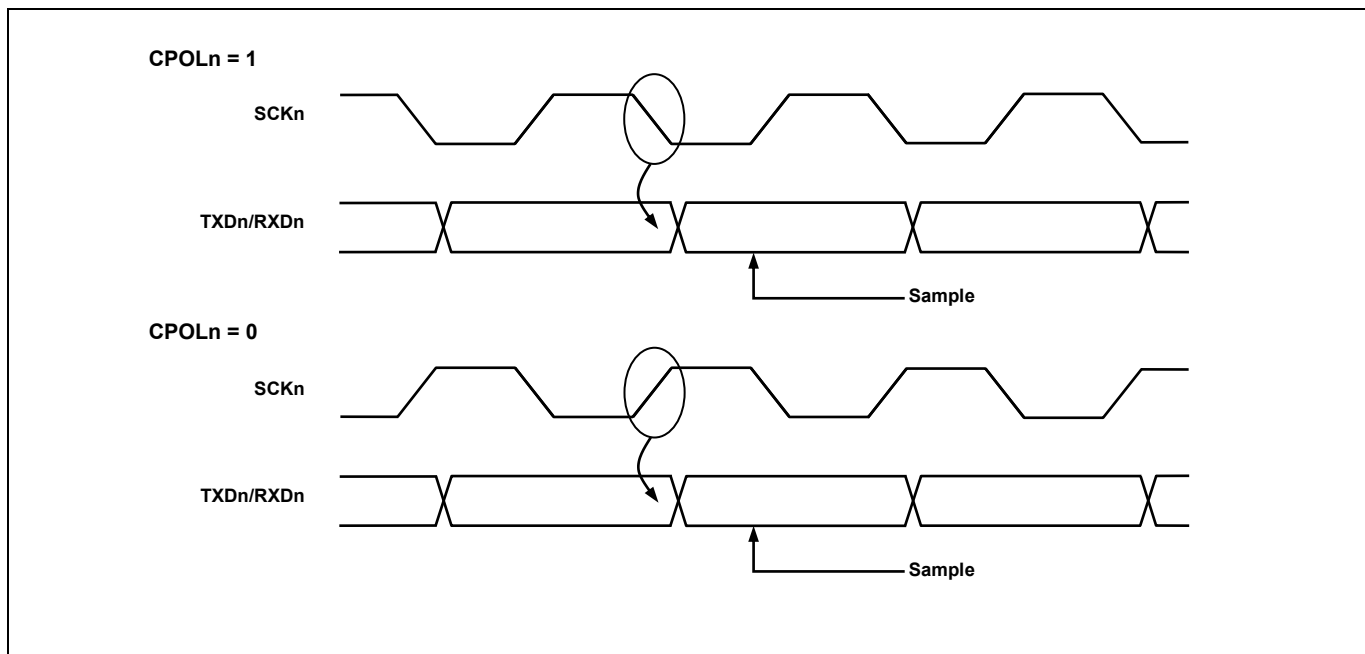


Figure 11.59 同步模式 SCKn 时序 (USIn)

11.12.7 USIn UART 数据格式

串行格式定义为同步位数据位中的字符，和随机的奇偶错误校验。

USAR支持以下全部30个数据组合中的任何一个有效的格式。

- 1 个开始位
- 5, 6, 7, 8 或 9 个数据位
- 无, 奇数或偶数校验位
- 1 或 2 停止位

一帧数据开始于紧接着最低有效位的开始位(LSB). 然后是下一个数据位，直到第9位，以最高有效位结束。如果校验功能使能，校验位插入到最后一个数据位和停止位之间。数据引脚上从低到高的传送被视为开始位。当一帧完整的数据传送完成，下一帧数据会直接跟随其后。或者通信数据线可以设置为休眠状态。休眠意思是数据脚高电平。下图所示为可能用到的数据格式。里边的位数可以选择。

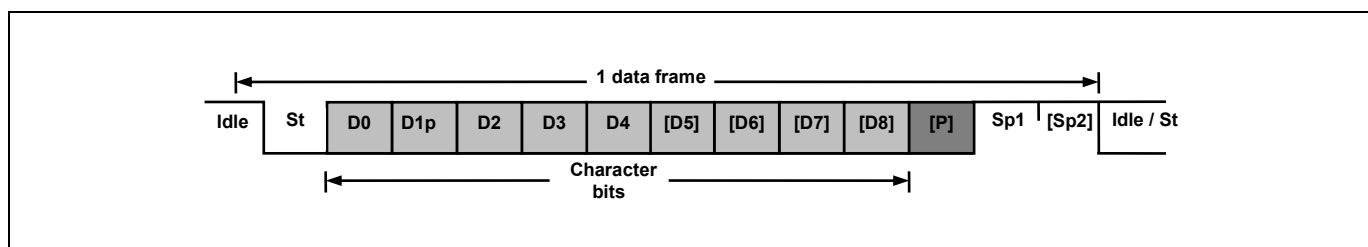


Figure 11.60 帧格式(USIn)

1 帧数据包括以下位

- Idle 在通讯线上没有通讯(TXD0/RXD0)
- St 开始位(Low)
- Dn 数据位(0~8)
- Parity bit -----奇、偶校验，无校验
- Stop bit(s) ----- 1 bit 或 2 bits

UART 帧格式通过USInCR1 中的USInS[2:0], USInPM[1:0]位和 USInCR3 寄存器中的USInSB 位来设置。发送和接收使用相同的设置。

11.12.8 USIn UART 校验位

校验位是所有数据位通过异或计算得到. 如果使用奇校验, 异或的结果是反向的. 校验位插入到ST位 和 第一个停止位.

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

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

P_{even} : 校验位使用偶校验

P_{odd} : 校验位采用奇校验

D_n : 数据位

11.12.9 USIn UART 发送

UART发送的使能是通过设置USInCR2寄存器中的TXE0 位. 当发送使能时, TXD0 引脚应该通过P4FSR[3:2]设置为TXD0 功能, 波特率, 操作模式和帧格式必须在发送之前一次设置完. 在同步工作模式时, SCK0 引脚用于传送时钟, 所以需要通过P4FSR[5:4]设置为SCK0 .

11.12.9.1 USIn UART 发送 TX 数据

数据传输通过加载所要发送的数据到发送缓存器(USInDR)开始. 当移位寄存器准备好发送新的一帧数据时, 写入到发送缓存器中的数据移位到移位寄存器中. 移位寄存器在空闲状态或在发送的最后一个停止位后立即加载新的数据. 当移位寄存器加载新的数据时, 会根据控制寄存器设置发送一帧完整的数据. 如果这9位已经被用, 第九位必须在它加载到发送缓存器(USInDR)之前写到USInCR3寄存器中的USInTX8位.

11.12.9.2 USIn UART 发送标志位和中断

UART 发送器有2个标志位, 它表明了其状态. 一个是UART 数据寄存器空标志(DREn) 另一个是传送完成标志位(TXCn). 俩位都可以作为中断源.

DREn表示传送暂存器已经准备去接收新的数据. 当发送暂存器清空之后该位置位, 当发送暂存器包含被发送但是还没有移位到移位寄存器的数据时该位清零. 该标志位也可以在该位置写入0进行清除. 写入1是不允许的→prohibited.

当USInCR2寄存器中的数据寄存器清空中断(DRIEn)使能, 同时总中断也打开, DREn置位时 USInST1 状态寄存器清空中断会产生中断

当发送移位寄存器中的整帧数据已经移出并且没有数据在缓存器中时TXCn标志位置位. TXCn标志位自动清除当发送完成中断服务程序执行时, 或是你可以在USInST1寄存器的TXCn位写入0进行清零.

当USInCR2寄存器中的发送完成中断使能位(TXCIEn) 置位并且总中断打开, 当TXCn 位置位时UART 发送完成中断会产生中断.

11.12.9.3 USIn UART 奇偶校验

奇偶校验发生器对发出的串行数据进行奇偶位计算。当奇偶校验使能时(USInPM1=1), 发送器控制逻辑将其插入到MSB和帧数据的第一个停止位之间。

11.12.9.4 USIn UART 禁止发送

通过清空TXE 位禁止发送不会生效直到正在发送的数据发送完整。当发送禁止时, TXDn引脚可以作为普通I/O口(GPIO)。

11.12.10 USIn UART 接收

UART 通过设置USInCR2 寄存器的RXEn位来使能。当其使能时, RXDn 脚应该设为输入状态RXDn功能口。波特率, 操作模式和帧格式必须在接收之前设置完成。在同步或者SPI 工作模式时, SCKn用于传送时钟, 所以需要通过P4FSR [5:4]设置为SCKn 功能。在SPI 工作模式时, SSn 引脚在从机模式或配置为SSn输出口(master)。可以通过USInCR 3 寄存器的USInSSEN 位来设置。

11.12.10.1 USIn UART 接收 RX 数据

当UART为同步或异步模式下, 当RXDn上检测到开始位(LOW)时接收器开始数据接收。开始位后的每一位由预定义的波特率和移位到移位寄存器开始采样直到接收到第一位停止位。即使一帧中有第二个停止位, 第二个停止位也被接收器忽略。也就是说, 接收到第一个停止位意味着一帧完整的数据存在于接收移位寄存器, 同时移位寄存器的内容转移到接收暂存器。接收暂存器通过USInDR 寄存器读取。

如果9位字符被用 (USInS[2:0] = "111"), 第九位存储到USInCR3寄存器USInRX8的位置。第9位必须在从USInDR读取低8位之前通过USInRX8读取。同样的, 错误标志位FEn, DORn, PEn 必须在从USInDR读取数据之前读取。因为错误标志位是存储在接收暂存器中和FIFO相同的位置。

11.12.10.2 USIn UART 接收标志位和中断

UART 接收器有一个标志位用来表明接收器的状态。

接收完成 (RXCn) 标志位表示在接收缓存器中是否有未读取的数据。当在接收缓存器有未读取的数据置位，当接收缓存器清空时该位清零。如果接收器禁止(RXEn=0),接收缓存器被清空并且RXCn清零。

当USInCR2寄存器中的接收完成中断使能位(RXCIEn)置位同时总中断打开，RXCn标志位置位时UART 接收完成中断产生中断

UART 接收器有三个错误标志位，分别是帧错误(FEn)，数据超出(DORn) 和奇偶校验错误 (PEn)。这些错误标志位可以通过USInST1 寄存器读取。接收数据存储于2级接收暂存器，这些标志位也存储在接收缓冲区相同的位置。所以，在从USInDR 寄存器读取接收数据前，先读取包含错误标志位的USInST1 寄存器..

帧格式错误(FEn) 标志位表示第一个停止位的状态。当停止位正确的检测到‘1’时，FEn 是 ‘0’。当停止位错误，也就是检测到0时，FE 是 ‘1’。该位可以用检测帧与帧之间数据失去同步。

数据超出 (DORn)标志位表明由于接收缓冲区装满而引起的数据丢失。当接收缓存器装满，同时另一个已经存储到接收缓存器的新数据又存在于接收移位寄存器时，DORn 产生。当DORn标志位置位后，所有进来的数据都会丢失。避免数据的丢失或清除该位，读取接收缓存器。

奇偶校验错误 (PEn) 标志位表明在接收暂存器接收的数据中已经有一个校验错误。如果奇偶校验检测功能没有使能(USInPM1=0), PE0位通常是读 ‘0’。

11.12.10.3 USIn UART 奇偶校验

如果奇偶校验位使能(USInPM1=1)，奇偶校验检查器计算进来的数据的奇偶和接收到串口帧数据进行比较。

11.12.10.4 USIn UART 禁止接收

与发送形成对比，通过清零RXEn位来禁止接收器使接收器立即执行。当接收器禁止时，接收器清空缓存器，在缓存器中剩下的数据全部复位，同时RXDn 引脚可以作为普通I/O口 (GPIO)。

11.12.10.5 USIn 异步数据接收

接收外部数据结构时，UART 包括一个时钟和数据恢复装置。时钟恢复逻辑用于将内部产生的波特率时钟和从RXDn引脚引入的异步连续数据结构相同步。

据恢复逻辑对引入位进行采样和低通滤波，这样可以消除RXDn脚上的干扰。

下面举例说明对一个引入结构的开始位进行采样的过程。采样率为波特率正常模式时16次，双速模式(DBLSn=1)时8次，水平箭头表示的是由于异步采样引起的同步变化，注意多数时间显示的都是双速模式。

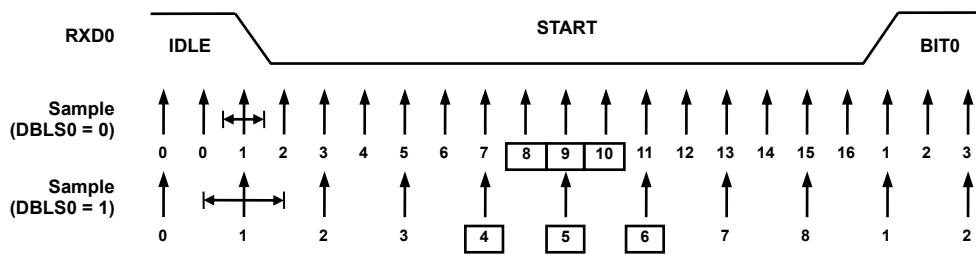


Figure 11.61 异步开始采样 (USIn)

当接收使能时(RXEn=1), 时钟提取逻辑尝试在RXDn上查找一个由高到低的过度变化, 这是开始位的条件。在RXDn上检测到由高到低的变化时, 常规模式中时钟提取逻辑采用采样的8, 9和10位以用于检测是否接收到有效的开始位。如果采样到多于两个的逻辑低电平, 就认为是有效地开始信号, 同时内部产生的时钟同步进入的数据。数据覆盖开始, 每个开始位同步过程都被重复。

综上所述, 当接收时钟和开始位同步时, 数据传送开始。数据获取的过程和时钟获取的过程是类似的。常规模式时每个引入的位数据覆盖逻辑采样16次, 双速模式时采样8次。常规模式时采样的8,9,10决定数据值。如果采样到多于2个的低电平, 接收位认为是逻辑 '0' 如果采样到多于2个的高电平, 则认为是逻辑'1'。数据覆盖过程一直被重复直到包含一个停止位的完整的数据接收完成。决定位的值按顺序存储到接收移位寄存器。注意接收器只使用第一个停止位。在内部接收到第一个停止位后, 接收器进入空闲状态等待开始位。

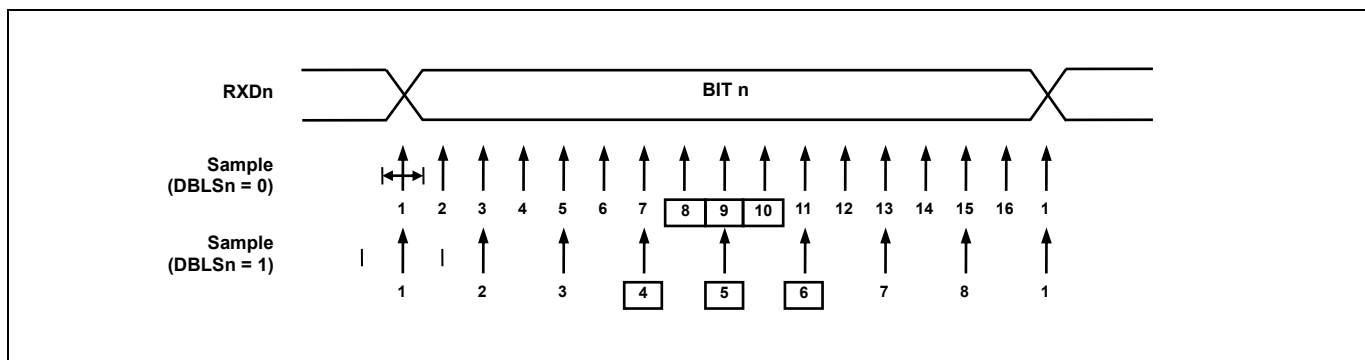


Figure 11.62 异步数据采样和校验位(USIn)

检测停止位的过程和时钟、数据的获取过程类似。也就是, 如果采样到多于2个或3个中间值是高电平, 就认为检测到停止位, 否则, FE标志位置位。在确定第一个停止位有效后, 接收器进入空闲模式等待下一个开始位。

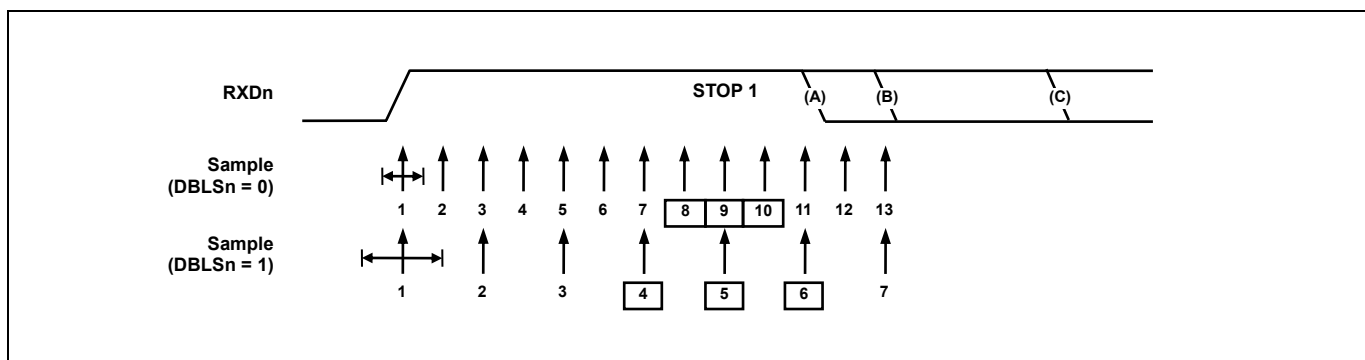


Figure 11.63 停止位采样和下一个开始信号采样(USIn)

11.12.11 USIn SPI 模式

USIn 可以设置为工业标准SPI 模式. SPI 模式有以下特性.

- 全双工，三相同步数据传送
- 主机和从机模式
- 支持所有SPIn 模式中的所有模式(mode 0, 1, 2, and 3)
- 可选择数据传送顺序低位优先还是高位优先
- 双缓存发送和接收
- 可编程传送位速率

SPI 模式使能时(USInMS[1:0]="11"), 在从动模式中 SSn脚低电平输入有效, 或者在USInSSEN 为0时设置为输出（主机模式）.

注意在SPI 模式时, RXDn 更名为MISON , TXDn 为 MOSIn.

11.12.12 USIn SPI 时钟格式和时序

为适应不同的设备, USIn 有一个时钟极性位(CPOLn)和相位控制位 (CPHAn) 去选择数据传送的时钟格式.

CPOLn 选择性的嵌入一个与时钟串联一起的逆变器. CPHAn 选择时钟和数据的不同相位. 注意根据USInMS[1:0]位决定USIn 工作模式USInCR1寄存器的CPHAn和CPOLn位有不同的含义.

下表为SPI的四种模式.

SPI Mode	CPOLn	CPHAn	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

Table 11.20 CPOLn 功能

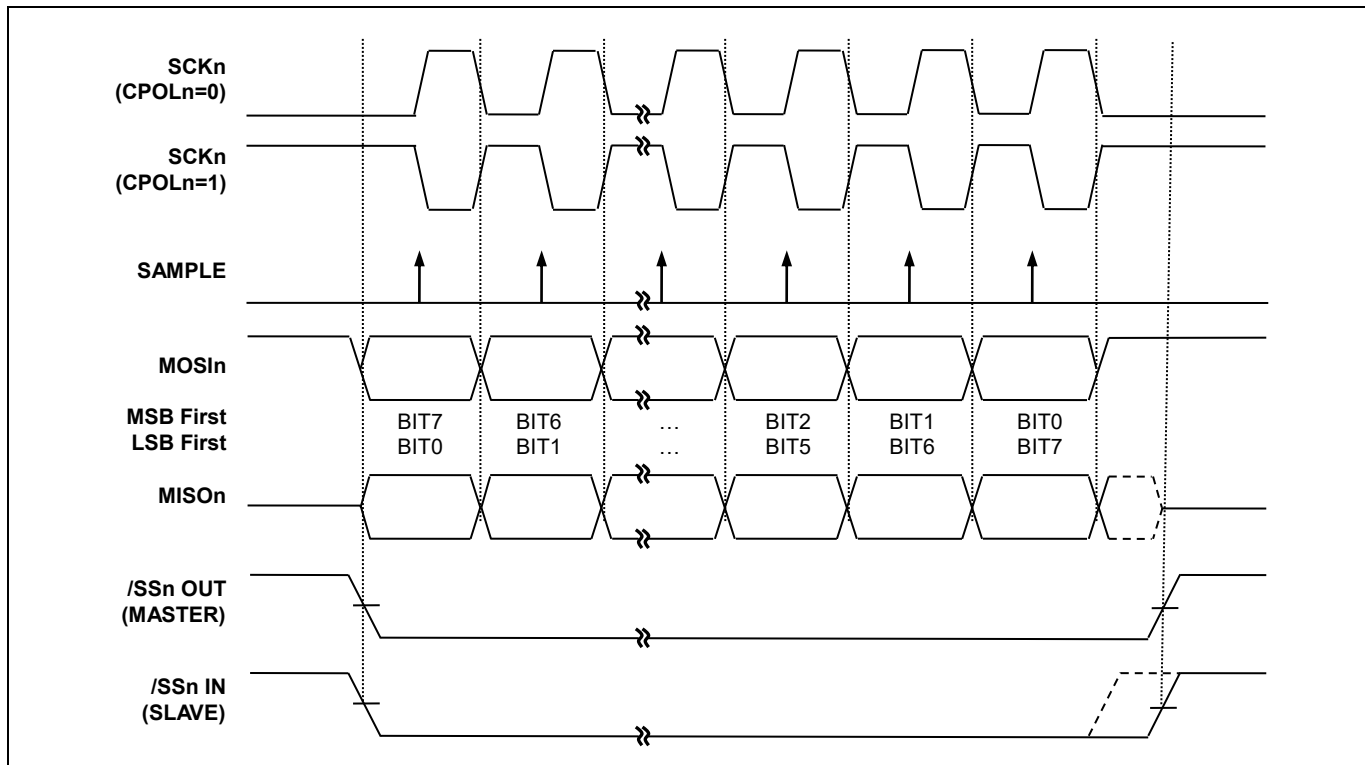


Figure 11.64 USn SPI Clock Formats when CPHAn=0

CPHAn=0时, 当SSn低电平有效时从机开始驱动MISOIn 输出第一个数据位的值. 第一个 SCKn 边沿引起主机和从机MISOIn 和 MOSIn分别进行数据值采集. 第二个SCKn 边沿, USIn 转移第二个数据位的值. 和CPHAn=1时不同, CPHAn=0时, 在传送过程中从机的SSn 输入必须定位到高电平. 这是因为当它检测到SSn输入的下沿时从机可以准备第一位数据.

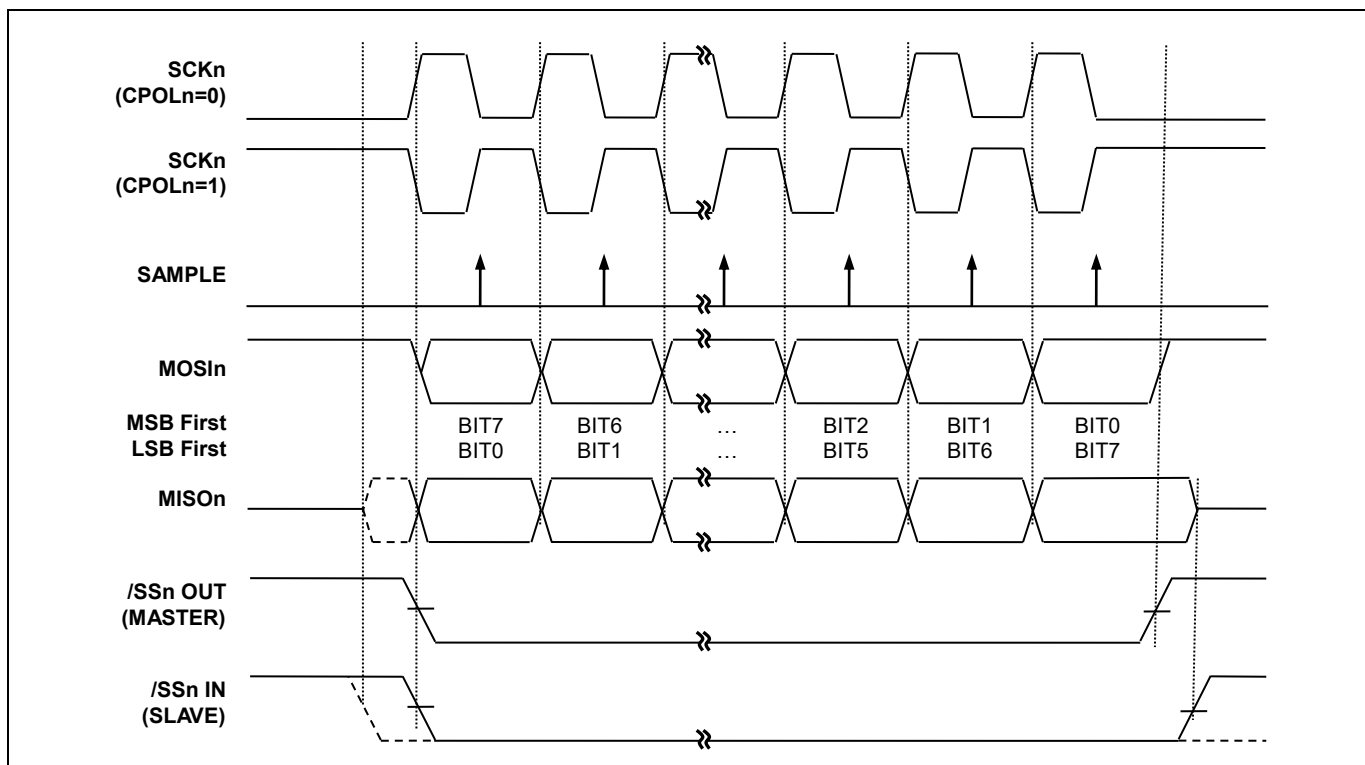


Figure 11.65 USIn SPI Clock Formats when CPHA=1

CPHA=1, SSn低电平有效时从机开始驱动MISOIn 输出，但是数据不确定直到第一个SCKn 边沿. 第一个 SCKn边沿从移位器转移数据的第一位到 MOSIn (master)和MISOIn (slave). 下一个SCKn 边沿引起主机和从机MISOIn 和 MOSIn 分别进行数据采样. 第三个SCKn 边沿, USIn 转移数据第二位值到MOSIn 和 MISOIn. CPHA=1, 从机的 S Sn 输入不需要维持高电平.

因为 SPI 再使用USIn资源, SPI 模式和同步或异步模式类似. SPI 传送通过检测到USIn 数据寄存器清空标志位(DRE n=1)开始, 然后写一个字节到 USInDR. 主机模式时, 即使发送装置没有使能(TXEn=0), 也必须写入数据到USInDR , 因为SCKn 是从发送模块产生的.

11.12.13 USIn SPI 框图

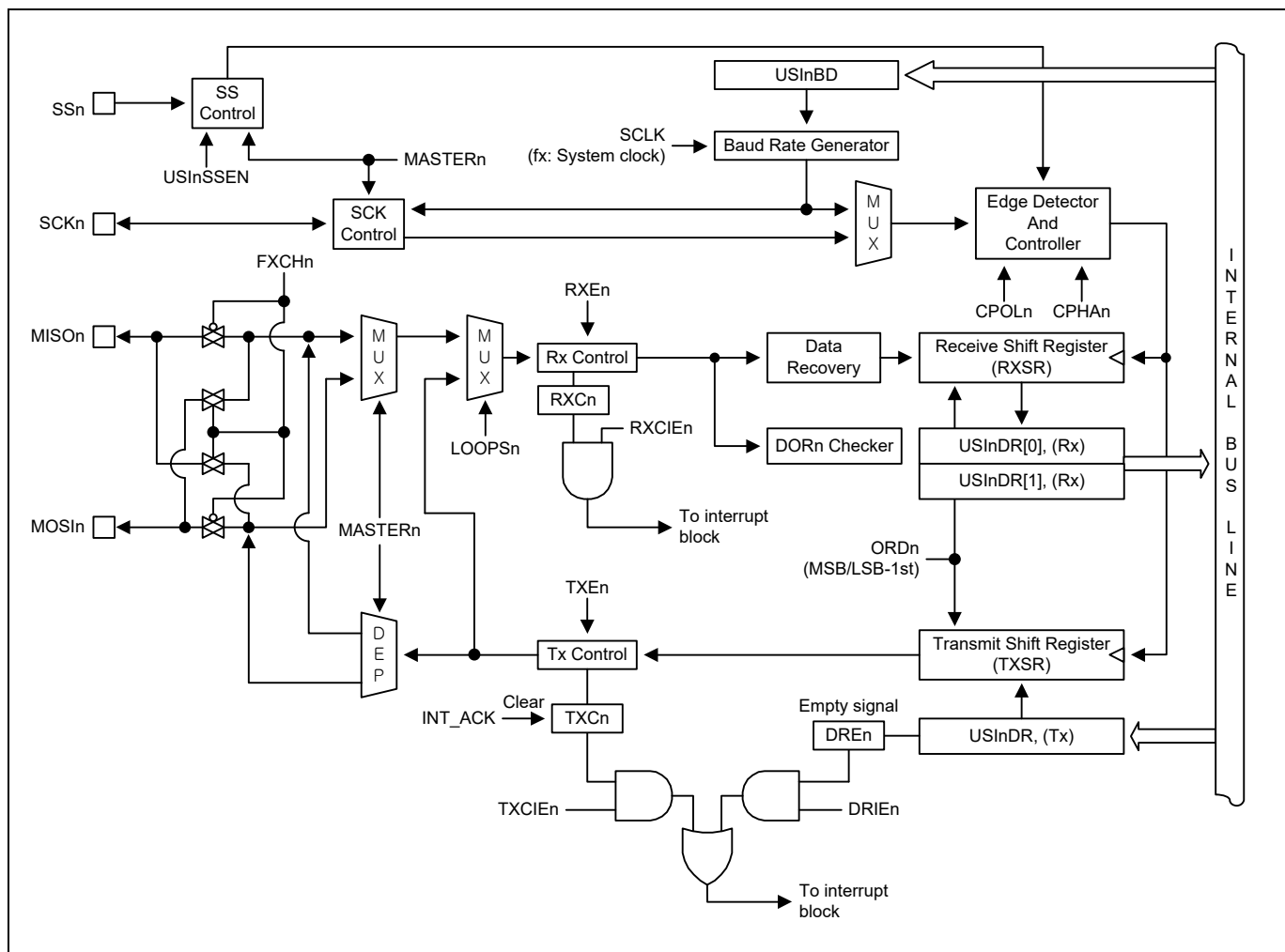


Figure 11.66 USIn SPI 框图

11.12.14 USIn I2C 模式

USIn 能够用作工业标准串口通讯协议模式. 它使用两根主线 (SDA) 和(SCL) 交换数据. 因为SDA 和 SCL都是漏极开路输出, 每根线都需要上拉电阻. 特性如下.

- 兼容 I2C 总线标准
- 多主控操作
- 高达400kHz 数据传输速率
- 7 位地址
- 主控和从动操作
- 总线忙时检测

11.12.15 USIn I2C 位传送

当时钟高电平周期时, SDA 总线上的数据必须保持稳定, SCL 线上的时钟信号是低时, 数据线上的高或低状态才可以改变. START(S), repeated START(Sr) 和 STOP(P) 情况除外, 这些情况下数据线改变发生在时钟线高电平时.

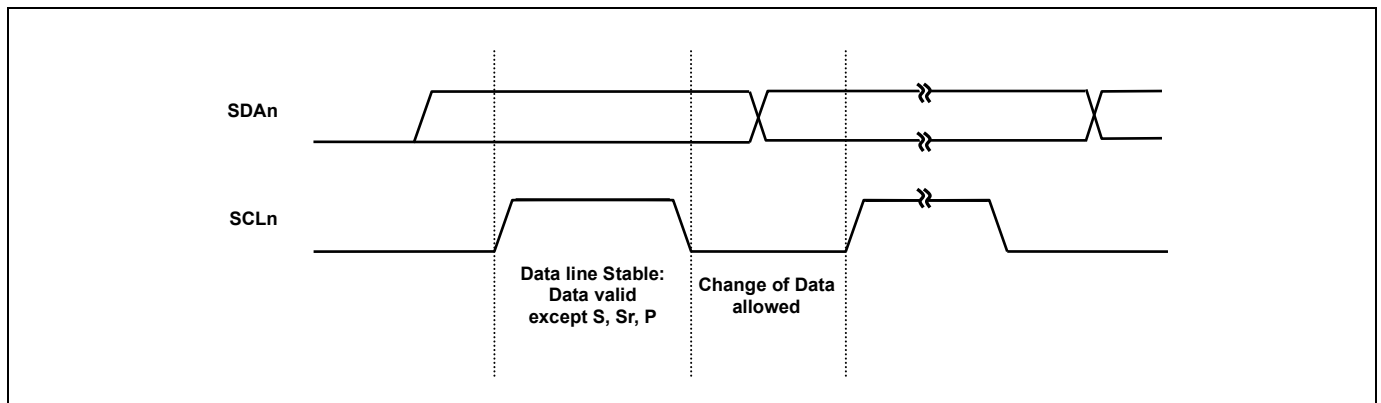


Figure 11.67 I2C-Bus 总线位传输(USIn)

11.12.16 USIn I2C 开始 / 重复开始/ 停止信号

主控器会发出一个START (S) 信号通知其他设备连接他们将会用到的SCL, SDA总线. 一个STOP (P) 信号由主控器产生去释放总线以便其他设备可以使用.

SCL高电平时, SDA上由高到低的转变被规定是一个 START (S)信号.

SCL高电平时, SDA上由低到高的转变被规定是一个STOP (P) 信号.

START 和 STOP 信号通常由主机产生. START 信号后总线通常被认为是忙碌. STOP 信号后总线被认为释放, ie, 总线在 START 和 STOP 信号之间是忙碌状态. 如果STOP信号被重复 START 信号 (Sr)取代, 总线保持忙碌状态. 所以, START 和 重复 START 信号功能上是一样的.

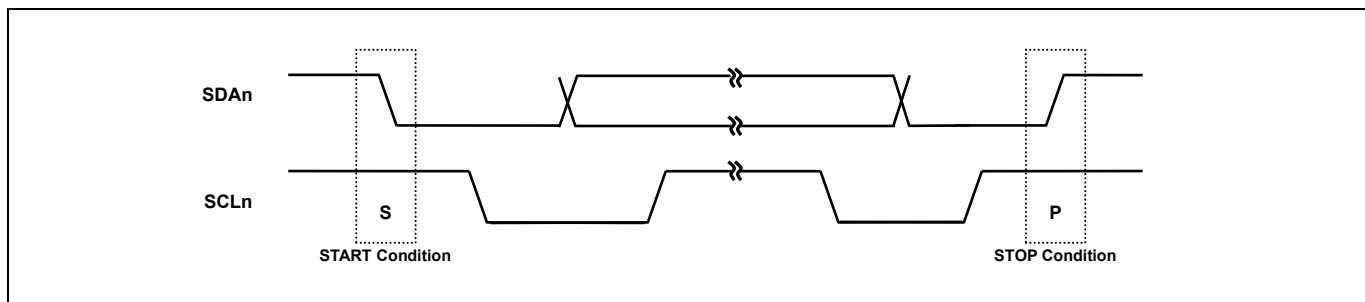


Figure 11.68 START 和 STOP 条件 (USIn)

11.12.17 USIn I2C 数据传送

SDA 线上的每一个字节必须是8位长度. 每次传输的字节数是没有限制的. 每个字节必须跟随一个应答位. 数据传送的方式是最高有效位优先 (MSB). 从机不能接收或发送另一个完整数据字节除非它完成一些其他的功能, 他可以保持SCL线上低电平强迫主机进入等待状态. 当从机为另一字节数据做准备时释放SCL线, 数据传送可以继续.

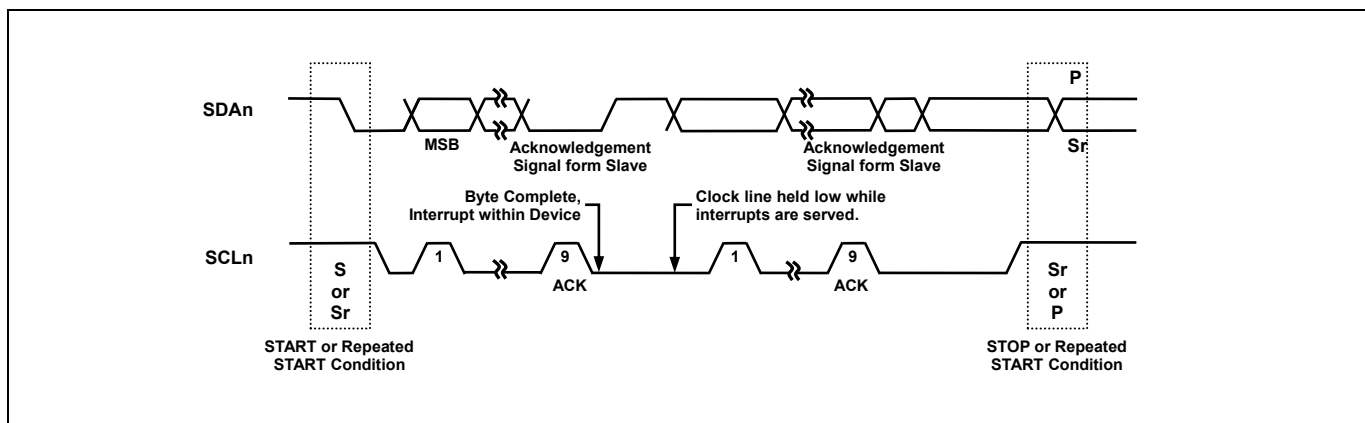


Figure 11.69 I2C-Bus 数据传输(USIn)

11.12.18 USIn I2C 应答

应答和主机产生的时钟脉冲宽度有关。在应答时钟脉冲中发送器释放SDA (HIGH). 接收器必须拉低SDA 线以便在时钟脉冲高时保持稳定的低电平. 当从机被主机编址后, 如果它因执行实时功能而不能接收或发送, 数据线必须被从机拉高, 以及, 从机被主机编址后, 不能接收更多数据位, 从机接收器必须释放SDA线. 主机可以产生一个STOP 信号退出发送, 或者重复开始信号开始新的传送.

如果主机接收器涉及到发送, 必须数据的结尾给从机发生器发送信号在输出主控器锁定的最后字节不产生应答. 从机发生器必须释放数据线以允许主机产生STOP 或重复START 信号.

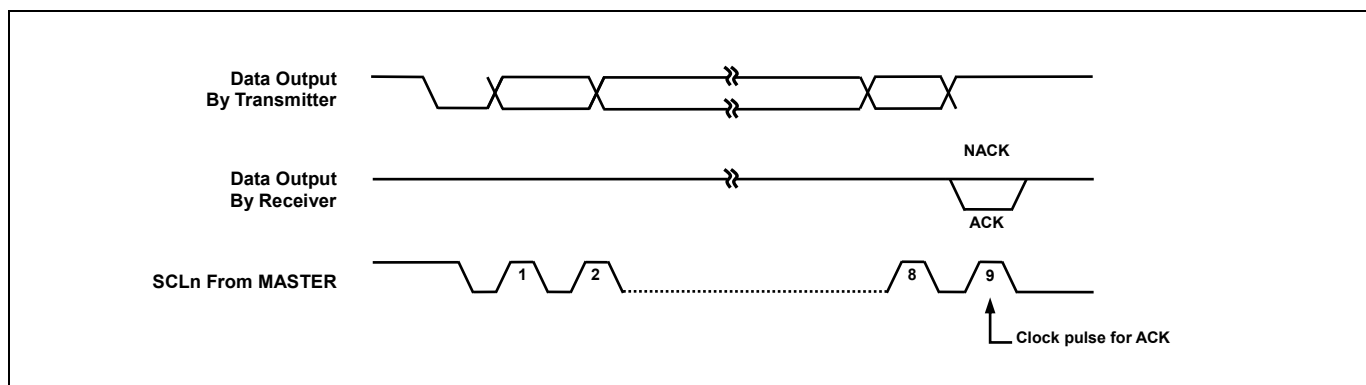


Figure 11.70 I2C-Bus 应答 (USIn)

11.12.19 USIn I2C 同步 / 仲裁

时钟同步是由I2C和SCL线与连接完成。意思就是SCL上一个由高到低到转变将会引起设备开始对低电平脉冲进行计数并保持 SCL 现在的状态直到时钟高电平被接收到。然而如果另一个时钟仍然维持在低电平的话, 时钟由低到高的变化不能改变SCL线的状态。这样, 随着最长的低电平脉冲产生一个同步SCL时钟, 高电平由最短的高脉冲决定.

如果总线空闲时主机可以开始进行发送. 两个或更多个主机可以产生START信号. 仲裁在SDA线上产生, 当SCL高电平时, 通过主机发送一个高电平, 当另一个主机在发送低电平时会关闭数据输出状态, 因为总线电平不等于它自身的电平. 仲裁连续很多位直到一个主机赢得I2C总线. 第一阶段是对照地址位.

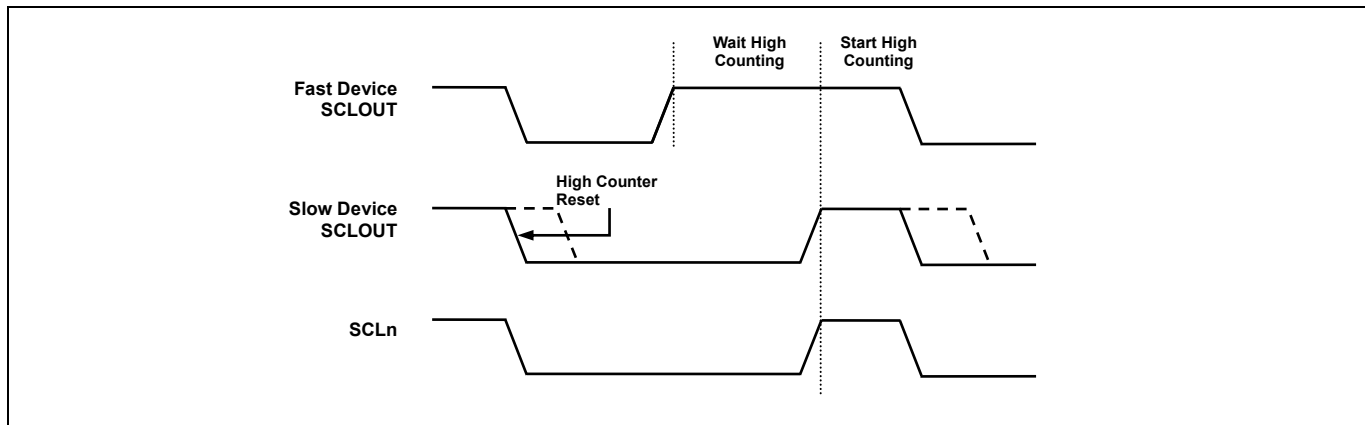


Figure 11.71 仲裁过程中的时钟同步 (USIn)

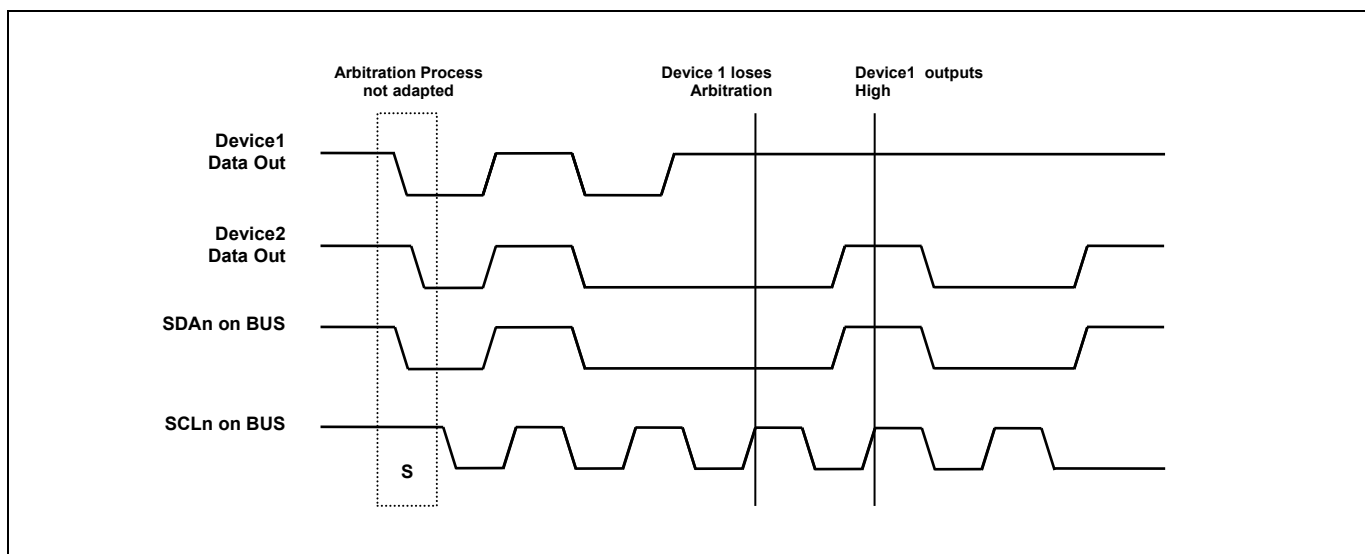


Figure 11.72 双主机仲裁过程 (USIn)

11.12.20 USIn I2C 操作

I2C 基于字节导向可产生中断。除了开始信号之外所有的总线活动结束以后中断被发出。由于I2C基于中断，在I2C字节传送时，应用软件可以任意进行其他的操作。

请注意，当 I2C 产生中断时，USInCR4 寄存器中的 IICnIFR 标志位置1，当 USInST2 寄存器中的所有中断源清零时清零。当 I2C 中断发生时， SCLn 线保持低电平，直到 USInST2 寄存器清零。当IICnIFR 标志位置1，USInST2 包括I2C 总线当前状态。 根据USInST2 值，软件可以决定下一步动作。

I2C 通过设置主动/从动，发送/接收可以有4个操作模式。操作模式由赢得的主机来配置。以下是详细的操作说明。

11.12.20.1 USIn I2C 主机发送

I2C 作为主机发送时，以下是推荐的步骤.

1. 设置USInCR1寄存器的USInMS[1:0]位和USInCR2的USInEN位使能 I2C. 这里向外围提供主时钟.
2. 加载 SLA+W 到USInDR , 这里 SLAn 是从机的地址, W 是基于主机角度的传送方向. 主机发送时, W 是 '0'. 注意USInDR 被地址和数据共用.
3. 在SCL由低到高跳变时通过写入期望值到USInSCLR 和 USInSCHR设置波特率.
4. 设置 USInSDHR决定从SCL的下降沿开始何时SDAn改变数值 . 如果SDA应该在SCL低脉冲的中间改变, 加载USInSCLR 一半的值得到 USInSDHR.
5. 设置USInCR4的 STARTCn位. 发送一个 START 信号. 并且设置如何处理中断和ACK信号. 当STARTC 位置位时, USInDR 中的8位数据根据波特率发送出来.
6. 这是主控机发送地址信息包的ACK处理信号. 7位地址和1位传输方向被发送到目标从机时, 主机会知道在SCLn高电平的第九位从机是否应答. 如果主机赢得总线, I2C 会不管来自从机的ACK应答而产生 GCALL中断.. 如果在仲裁过程中 I2C 丢失总线控制权, USInST2 的MLOSTn 会置位, I2C会以空闲状态等待或者作为一个可寻址从机来操作. 当USInST2的MLOSTn位置位时作为从机来操作, USInCR4中的 ACKnEN位必须置位同时接收到的7位地址必须等于USInSAR的USInSLA[6:0]位. 假如这样的话I2C作为从机发送或从机接收来操作. 在这个阶段, I2C 保持SCL 低电平. 这是由于决定I2C是否继续传送还是停止通讯. 假如I2C在第一个数据传送时没有丢失总线主控权继续下一步.

I2C (Master) 可以忽略从机的ACK应答信号而选择一下的某一个情况.

- 1) 主机从从机接收ACK信号, 由于从机可以从主机接收更多的数据所以继续数据传送. 这样的话, 加载数据到USInDR.
- 2) 主机停止数据传送即使它接收从机的ACK信号. 此时USInCR4的STOPCn位置位.
- 3) 主机不检测ACK信号而发送重复START信号. 此时, 加载 SLA+R/W 到 USInDR 同时USInCR4的STARTCn位置位.

在做了以上的这些动作以后, 将 USInST2 中的所有的中断源清零, 释放SCLn线. 在第一种情况时1), 转移到 7.如果是第二种情况 2), 转移到 9 去处理 STOP 中断. 如果是第三种情况 3), 发送完USInDR中的数据后转移到6, 同时如果传送方向位是'1' 转到主机接收部分.

7. 发送一个字节的的数据, 在传送过程中, 总线仲裁继续.

8. 这是主控制器传送信息数据包的ACK信号处理环节。I2C保持SCLn低电平。在传送数据仲裁其他主控制器I2C丢失总线控制权时，USInST2的MLOSTn置位，此时，I2C在空闲状态等待。USInDR中的数据发送完成，I2C产生TENDn 中断。

I2C 可以忽略从机的ACK信号选择以下的一种情况

- 1) 主机从从机接收ACK信号，由于从机可以从主机接收更多的数据所以继续数据传送。这样的话，加载数据到USInDR。
- 2) 主机停止数据传送即使它接收从机的ACK信号。此时USInCR4的STOPCn位置位。
- 3) 主机不检测ACK信号而发送重复START信号。此时，加载 SLA+R/W 到 USInDR 同时USInCR4的STAR TCn位置位。

在做了以上的这些动作以后，将 USInST2 中的所有的中断源清零，去释放SCLn线。在第一种情况时1)，转移到7。如果是第二种情况 2)，转移到 9 去处理 STOP 中断。如果是第三种情况 3)，发送完USInDR中的数据后转移到6，同时如果传送方向位是‘1’ 转到主机接收部分。

9. 这是I2C传送功能的最后一步，执行STOP 中断，STOP 位表示主机和从机之间的数据传送已经完成。写入‘0’到USInST2，清除USInST2，然后I2C 进入空闲状态。

11.12.20.2 USIn I2C 主机接收

I2C 作为主机接收时，以下是操作步骤。

1. 设置USInCR1的USInMS[1:0]位和USInCR2的USInEN位使能 I2C. 这里向外围提供主时钟
2. 加载 SLA+R 到USInDR ，这里 SLA 是从机的地址， R 是基于主机角度的传送方向. 主机接收时， R 是 '1'. 注意USInDR 被地址和数据共用.
3. 在SCLn由低到高跳变时通过写入期望值到USInSCLR和 USInSCHR设置波特率.
4. 设置 USInSDHR决定从SCL的下降沿开始何时SDA改变数值 . 如果SDA应该在SCL低脉冲的中间改变, 加载 USInSCLR 一半的值到 USInSDHR.
5. 设置USInCR4的 STARTCn位. 发送一个 START 信号. 并且设置如何处理中断和ACK信号. 当STARTC 位置位时， USInDR 中的8位数据根据波特率发送出来.
6. 这是主控机发送地址信息包的ACK处理信号。7位地址和1位传输方向被发送到目标从机时，主机会知道在SCL高电平第九位从机是否应答。如果主机赢得总线，I2C 会忽略来自从机的ACK应答而产生 GCALL中断。如果在仲裁过程中 I2C 丢失总线控制权， USInST2 的MLOSTn 会置位，I2C会以空闲状态等待或者作为一个可寻址从机来操作。当USInST2的MLOSTn位置位而作为从机来操作， USInCR4中的 ACKnEN位必须置位同时接收到的7位地址必须等于USInSAR的USInSLA[6:0]位。假如这样的话I2C作为从机发送或从机接收来操作。在这个阶段，I2C 保持SCL 低电平。这是由于决定I2C是否继续传送还是停止通讯。假如I2C在第一个数据传送时没有丢失总线主控权继续下一步。

I2C (Master) 可以根据从机的ACK应答信号而选择以下的某一种情况。

- 1) 主机从从机接收ACK信号，由于从机可以从主机接收更多的数据所以继续数据传送。设置USInCR4的ACKnEN位决定I2C是否应答下一个接收的数据。
- 2) 主机停止数据传送因为没有接收到从机的ACK信号。此时USInCR4的STOPCn位置位。
- 3) 主机由于没有检测到从机的ACK信号而发送重复START信号。此时，加载 SLA+R/W 到 USInDR 同时USInCR4的STARTnC位置位。

在做了以上的某个动作以后，将 USInST2 中的所有的中断源清零，去释放SCLn线。在第一种情况时1)，转移到7.如果是第二种情况 2)，转移到 9 去处理 STOP 中断。如果是第三种情况 3)，发送完USInDR中的数据后转移到6，同时如果传送方向位是'0' 转到主机发送部分。

7. 接收一个字节的的数据.
8. 这是主控器传送信息数据包的ACK信号处理环节。I2C保持SCLn低电平。当一个字节数据接收完成，I2C产生TENDn中断。I2Cn可以根据USInST2的RXACKn标志位而选择以下的某一种情况。
 - 1) 主机继续从从机接收数据。要这样做须设置USInCR4 的ACKnEN 去应答下一个要接收的数据
 - 2) 如果没有接收到下个数据的ACK信号主机想要终止数据传送，可以通过清除USInCR4的ACKnEN来这样做。
 - 3) 由于没有检测到ACK信号，主机终止数据传送，此时，USInCR4的 STOPCn置位。
 - 4) 没有检测到 ACK 信号，主机发送重复START 信号。在这种情况下，加载 SLA+R/W 到USInDR ，USInCR4的STARTCn置位。

在做了以上的某个动作以后，将 USInST2 中的所有的中断源清零，去释放SCLn线。在第一种情况或第二种情况时)，转移到7。如果是第三种情况 3)，转移到 9 去处理 STOP 中断，如果是第四种情况，发送完USInDR 中的数据后转移到6，同时如果传送方向位是‘0’ 转到主机发送部分。

9. 这是I2C传送功能的最后一步，执行STOP 中断，STOP 位表示主机和从机之间的数据传送已经完成。写入任意值到USInST2 清除USInST2, 然后I2C 进入空闲状态。

11.12.20.3 USIn I2C 从机发送

I2C 作为从机发送时，以下是推荐的步骤.

1. 如果系统主操作时钟慢于 SCLn, 加载0x00 到USInSDHR, 使 SDA_n从SCLn的下降沿伴随着一个系统脉冲开始改变. 注意 SDA_n 的维持时间由SCLK 的SDAH 周期计算, 这里SDAH 是来自USInSDHR 的数. 当 SDA 的维持时间长于SCLK 周期时, I2C (slave) 不适合发送串行数据..
2. 设置USInCR1 的USInMS[1:0]位, USInCR4的IICnIE 位和USInCR2的USInEN位使能 I2C. 这里向外围提供系统主时钟.
3. 当检测到一个 START 信号, I2C接收一个字节的的数据和USInSAR 的USInSLA[6:0] 位比较. 如果USInSAR 的GCALLn位使能, I2C 将接收到的数据和0x00进行比较, 一般调用地址.
4. 如果接收地址不等于USInSAR 的USInSLA[6:0], I2C 进入空闲状态等待另一个开始信号. 否则如果地址等于 USInSLA[6:0]位 并且ACKnEN 位是打开的, I2C 产生 SSEL中断并且 SCL线保持低电平. 注意即使地址等于USInSLA[6:0] , 当ACKnEN 位禁止时, I2C 仍进入空闲状态. SSELn 中断发生时, 加载传送数据到USInDR, 将USInST2 中的所有中断源清零, 来释放SCLn线.
5. 一个数据发送完成.
6. 在这一过程中, I2C 产生TENDn 中断并且维持SCLn低电平而忽略来自主机的ACK应答信号. 从机可以选择以下某一种情况.

1) 没有检测到 ACK 信号, I2C 等待 STOP 或重复的START 信号.

2) 检测到来自主机的ACK 信号. 加载数据到 USInDR.

在做了以上的某个动作以后, 将 USInST2 中的所有中断源清零, 释放SCL线. 在第一种情况), 转移到7终止通讯. 如果是第2种情况), 转移到 5, 无论发生哪种情况, 只要检测到重复开始信号, 就转到4

7. 这是I2C从机发送功能的最后一步, 执行STOP中断. STOPC位表示主机和从机之间的数据传送已经完成. USInST2 清零, 写'0' 到 USInST2, 然后I2C 进入空闲状态.

11.12.20.4 USIn I2C 从机接收

I2C 作为从机接收时，以下是推荐的步骤。

1. 如果系统主操作时钟慢于 SCL，加载0x00 到 USInSDHR，使 SDA从SCL的下降沿伴随着一个系统脉冲开始改变。注意 SDA 的维持时间由SCLK 的SDAH 周期计算，这里SDAH 是来自USInSDHR 的数。当SDA 的维持时间长于SCLK 周期时，I2C (slave) 不适合发送串行数据。
2. 设置USInCR1的USInMS[1:0]位 和USInCR4的IICnIE位及USInCR2的USInEN位使能 I2C。这里向外围提供系统主时钟。
3. 当检测到一个 START 信号，I2C接收一个字节的的数据和USInSAR 的USInSLA[6:0]位比较。如果USInSAR的 GCALLn位使能，I2C 将接收到的数据和0x00进行比较，一般调用地址。
4. 如果接收地址不等于USInSAR的SLAn，I2C 进入空闲状态等待另一个开始信号。否则如果地址等于SLA位 并且ACKnEN 位是打开的，I2C 产生 SSELn中断并且 SCL线保持低电平。注意即使地址等于SLAn位，当ACK nEN 位禁止时，I2C 仍进入空闲状态。SSEL 中断发生时，I2C准备接收数据，将 USInST2 中的所有中断源清零，来释放SCLn线。
5. 传输一个字节的的数据。
6. 在这一过程中，I2C 产生TENDn 中断并且维持SCLn 低电平而忽略来自主机的ACK应答信号。从机可以选择以下某一种情况。

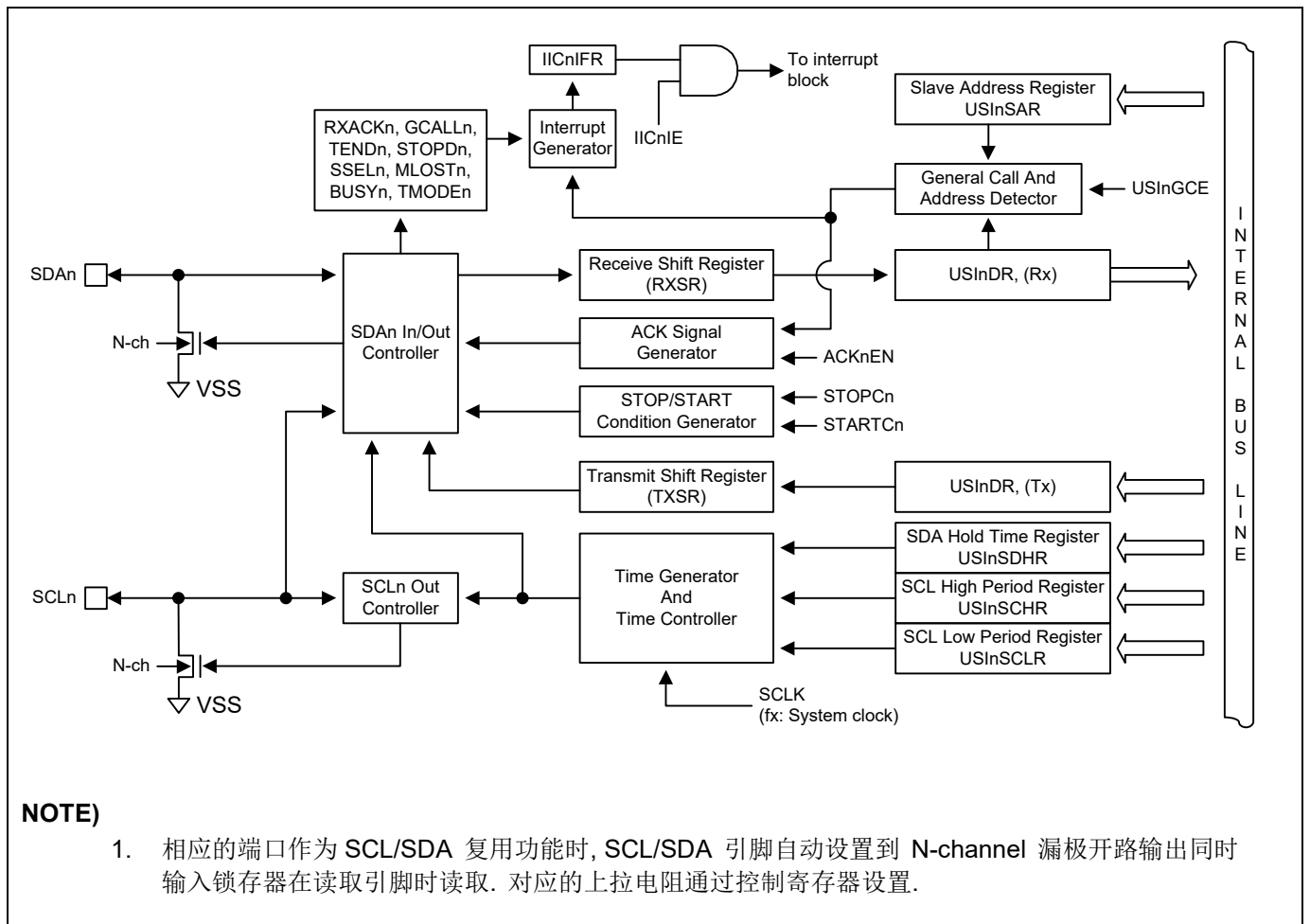
1) 没有检测到 ACK 信号(ACKnEN=0)，I2C 等待 STOP 或重复的START 信号。

2) 检测到来自主机的ACK 信号(ACKnEN=1)。I2C可继续从主机接收数据ACK。

在做了以上的某个动作以后，将 USInST2 中的所有中断源清零，来释放SCLn线。在第一种情况)，转移到7终止通讯。如果是第2种情况)，转移到 5，无论发生哪种情况，只要检测到重复开始信号，就转到4。

7. 这是I2C从机接收功能的最后一步，执行STOP中断。STOPCn位表示主机和从机之间的数据传送已经完成。USInST2清零，写'0' 到USInST2，然后I2C 进入空闲状态。

11.12.21 USIn I2C 方框图



NOTE)

1. 相应的端口作为 SCL/SDA 复用功能时, SCL/SDA 引脚自动设置到 N-channel 漏极开路输出同时输入锁存器在读取引脚时读取. 对应的上拉电阻通过控制寄存器设置.

Figure 11.73 USIn I2C 框图

11.12.22 寄存器图

Name	Address	Direction	Default	Description
USI0BD	E3H	R/W	FFH	USI0 波特率发生寄存器
USI0DR	E5H	R/W	00H	USI0 数据寄存器
USI0SDHR	E4H	R/W	01H	USI0 SDA 维持时间寄存器
USI0SCHR	E7H	R/W	3FH	USI0 SCL 高脉冲寄存器
USI0SCLR	E6H	R/W	3FH	USI0 SCL 低脉冲寄存器
USI0SAR	DDH	R/W	00H	USI0 从机地址寄存器
USI0CR1	D9H	R/W	00H	USI0 控制寄存器 1
USI0CR2	DAH	R/W	00H	USI0 控制寄存器 2
USI0CR3	DBH	R/W	00H	USI0 控制寄存器 3
USI0CR4	DCH	R/W	00H	USI0 控制寄存器 4
USI0ST1	E1H	R/W	80H	USI0 状态寄存器 1
USI0ST2	E2H	R	00H	USI0 状态寄存器 2
USI1BD	F3H	R/W	FFH	USI1 波特率发生寄存器
USI1DR	F5H	R/W	00H	USI1 数据寄存器
USI1SDHR	F4H	R/W	01H	USI1 SDA 维持时间寄存器
USI1SCHR	F7H	R/W	3FH	USI1 SCL 高脉冲寄存器
USI1SCLR	F6H	R/W	3FH	USI1 SCL 低脉冲寄存器
USI1SAR	EDH	R/W	00H	USI1 从机地址寄存器
USI1CR1	E9H	R/W	00H	USI1 控制寄存器 1
USI1CR2	EAH	R/W	00H	USI1 控制寄存器 2
USI1CR3	EBH	R/W	00H	USI1 控制寄存器 3
USI1CR4	ECH	R/W	00H	USI1 控制寄存器 4
USI1ST1	F1H	R/W	80H	USI1 状态寄存器 1
USI1ST2	F2H	R	00H	USI1 状态寄存器 2

Table 11.21 USI 寄存器

11.12.23 USIn 寄存器说明

USIn 模块由 USInBD, USInDR, USInSDHR, USInSCHR, USInSCLR, USInSAR, USInCR1/2/3/4, USInST1/2 组成.

11.12.24 USIn 寄存器说明

USInBD (USIn 波特率发生寄存器: UART 和 SPI模式): E3H

7	6	5	4	3	2	1	0
USInBD7	USInBD6	USInBD5	USInBD4	USInBD3	USInBD2	USInBD1	USInBD0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: FFH

USInBD[7:0] 寄存器的值用于产生内部波特率或在SPI模式产生SCKn 时钟。为预防故障，异步模式不要写 '0'，SPI模式不要写 '0' 或 '1'。

NOTE)

1. 和 USInSAR 寄存器一样，USInBD 寄存器在 I2C 模式中被用作从机地址。

USInDR (USIn 数据寄存器: UART, SPI, 和 I2C 模式): E5H

7	6	5	4	3	2	1	0
USInDR7	USInDR6	USInDR5	USInDR4	USInDR3	USInDR2	USInDR1	USInDR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

USInDR[7:0] USIn 发送缓存、接收缓存和数据寄存器共用相同的I/O 地址。发送数据缓存是要被写入到USInDR寄存器的。读USInDR寄存器返回接收缓存的内容。
只有当DREn置位时才可以写该寄存器。在SPI主控模式中，当数据写入到寄存器时SCK 时钟产生。

USInSDHR (USInSDA 维持时间寄存器: 用于I2C 模式): E4H

7	6	5	4	3	2	1	0
USInSDHR7	USInSDHR6	USInSDHR5	USInSDHR4	USInSDHR3	USInSDHR2	USInSDHR1	USInSDHR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 01H

USInSDHR[7:0] 该寄存器用于控制I2C模式中从SCL下降沿的SDAn输出时序.

注意)

1. SDAn 在 $t_{SCLK} \times (USInSDHR+2)$ 之后改变, 在 主控模式中 SDA n 在 SCLn 中间位置改变.
2. 在从动模式中, 该寄存器的配置和主机的 SCLn 频率有关.
3. 主控模式中 SDAn 在 $t_{SCLK} \times (USInSDHR+2)$ 之后改变. 所以, 为了 确保在从动模式下的运行, $t_{SCLK} \times (USInSDHR +2)$ 的值必须小于 SCL 的周期.

USInSCHR (USInSCL 高脉冲寄存器: 用于 I2C 模式): E7H

7	6	5	4	3	2	1	0
USInSCHR7	USInSCHR6	USInSCHR5	USInSCHR4	USInSCHR3	USInSCHR2	USInSCHR1	USInSCHR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 3FH

USInSCHR[7:0] 该寄存器定义了I2C 模式中SCLn的高脉冲.

基础时钟是SCLK, 系统时钟, 周期是通过下面方式计算出来的:

$$t_{SCLK} \times (4 \times USInSCHR + 2)$$

t_{SCLK} 是 SCLK 的周期.

所以, 主控模式下周期计算公式如下.

$$f_{I2C} = \frac{1}{t_{SCLK} \times (4 \times (USIO SCLR + USIO SCHR) + 4)}$$

USInSCLR (USInSCL 低脉冲寄存器: 用于 I2C 模式): E6H

7	6	5	4	3	2	1	0
USInSCLR7	USInSCLR6	USInSCLR5	USInSCLR4	USInSCLR3	USInSCLR2	USInSCLR1	USInSCLR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 3FH

USInSCLR[7:0] 该寄存器定义了I2C 模式中SCLn的高脉冲。
基础时钟是SCLK, 系统时钟, 周期是通过下面方式计算出来的:
 $t_{SCLK} \times (4 \times USInSCLR + 2)$
 t_{SCLK} 是 SCLK 的周期.

USInSAR (USIn 从机地址寄存器: 用于I2C 模式): DDH

7	6	5	4	3	2	1	0
USInSLA6	USInSLA5	USInSLA4	USInSLA3	USInSLA2	USInSLA1	USInSLA0	USInGCE
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

USInSLA[6:0] 配置I2C的从机地址

USInGCE 从动模式中该位决定 I2C是否允许general call address.

0 不允许 general call address

1 允许 general call address

USInCR1 (USIn 控制寄存器 1: 用于 UART, SPI, 和 I2C 模式): D9H

7	6	5	4	3	2	1	0
USInMS1	USInMSn	USInPM1	USInPMn	USInS2	USInS1 ORDn	USInS0 CPHAn	CPOLn
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

USInMS[1:0]	选择 USIn 工作模式			
	USInMS1	USInMS0	Operation mode	
	0	0	异步模式 (UART)	
	0	1	同步模式	
	1	0	I2C 模式	
	1	1	SPI 模式	
USInPM[1:0]	选择奇偶发生和校验方式(只用于UART 模式)			
	USInPM1	USInPM0	Parity	
	0	0	No Parity	
	0	1	Reserved	
	1	0	偶数	
	1	1	奇数	
USInS[2:0]	同步或异步模式下, 选择数据格式长度			
	USInS2	USInS1	USInS0	Data Length
	0	0	0	5 bit
	0	0	1	6 bit
	0	1	0	7 bit
	0	1	1	8 bit
	1	0	0	Reserved
	1	0	1	Reserved
	1	1	0	Reserved
	1	1	1	9 bit
ORDn	该位和USInS1在同一位位置. 设置为1时MSB首先发送. 为0时LSB首先发送(仅用于SPI 模式)			
	0	LSB-first		
	1	MSB-first		
CPHAn	该位和USInS0在同一位位置. 该位决定在SCKn的开始还是结束边沿进行数据采样(仅用于SPI 模式).			
	CPOLn	CPHAn	Leading edge	Trailing edge
	0	0	Sample (Rising)	Setup (Falling)
	0	1	Setup (Rising)	Sample (Falling)
	1	0	Sample (Falling)	Setup (Rising)
	1	1	Setup (Falling)	Sample (Rising)
CPOLn	该位决定ACK时钟极性在同步和SPI模式.			
	0	TXD change @Rising Edge, RXD change @Falling Edge		
	1	TXD change @Falling Edge, RXD change @Rising Edge		

USInCR2 (USIn 控制寄存器 2: 用于 UART, SPI, 和 I2C 模式): DAH

7	6	5	4	3	2	1	0
DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn	USInEN	DBLSn
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

DRIEn	中断使能位当寄存器空时(只用于 UART 和 SPI 模式). 0 DREn中断被抑制 (use polling) 1 当 DREn 置位时, 请求中断
TXCIEn	中断使能当发送完成时(只用于 UART 和 SPI 模式). 0 TXCn中断被抑制 (use polling) 1 当TXCn 置位时, 请求中断
RXCIEn	中断使能当接收完成时(只用于 UART 和 SPI 模式). 0 RXCn中断被抑制 (use polling) 1 当 RXCn 置位时, 请求中断
WAKEIEn	STOP模式异步唤醒时中断使能. 在睡眠模式时, RXDn 变为低电平时中断请求唤醒系统. (UART). 同时DRIEn 和 USInST1寄存器值应该被分别设置为'0b'和 "00H". 0 中断唤醒被抑制 1 当WAKEn置位时, 请求中断
TXEn	使能发送功能(UART 和 SPI). 0 发送禁止 1 发送使能
RXEn	使能接收功能(UART 和 SPI 模式). 0 接收禁止 1 接收使能
USInEN	激活 USIn 功能. n USIn禁止 1 USIn使能
DBLSn	选择接收采样率(only UART). 0 正常异步操作 1 双倍速异步操作

USInCR3 (USIn 控制寄存器 3: 用于UART, SPI, 和I2C): DBH

7	6	5	4	3	2	1	0
MASTERn	LOOPSn	DISSCKn	USInSSEN	FXCHn	USInSB	USInTX8	USInRX8
RW	RW	RW	RW	RW	RW	RW	R

Initial value: 00H

- MASTERn** 在SPI和同步模式操作中选择主控或从动，并且控制SCKn引脚方向
 - 0 从动模式 (外部时钟作为 SCKn).
 - 1 主控模式(内部时钟作为 SCKn).
- LOOPSn** 控制USIn循环模式用于测试(UART 和 SPI)
 - 0 正常模式
 - 1 循环模式
- DISSCKn** 在同步操作模式中，选择SCKn输出波形
 - 0 当UART在同步主控模式中使能时 ACK自由运行。
 - 1 有数据传输时ACK是激活状态
- USInSSEN** 控制 SSn 引脚(只用于SPI)
 - 0 禁止
 - 1 使能 (SSn 需设置为普通输入口)
- FXCHn** 改变 SPI 口功能(只用于SPI)
 - 0 No effect
 - 1 改变 MOSIn 和 MISO_n 功能
- USInSB** 在同步或异步模式下选择停止位长度.
 - 0 1 Stop Bit
 - 1 2 Stop Bit
- USInTX8** 在同步或异步模式下每帧数据的第九位. 加载USInDR寄存器之前首先写该位
 - 0 MSB (9th bit)发送 ‘0’
 - 1 MSB (9th bit)发送 ‘1’
- USInRX8** 在同步或异步模式下每帧数据的第九位. 读取接收缓存之前首先读该位 (只用于UART).
 - 0 MSB (9th bit) 接收 ‘0’
 - 1 MSB (9th bit) 接收 ‘1’

USInCR4 (USIn 控制寄存器 4: 用于I2C): DCH

7	6	5	4	3	2	1	0
IICnIFR	–	TXDLYENBn	IICnIE	ACKnEN	IMASTERn	STOPCn	STARTCn
R	–	RW	RW	RW	R	RW	RW

Initial value: 00H

IICnIFR	I2C中断标志位. 有中断产生时, 该位变为'1'. 将 USInST2 寄存器中所有中断源清零. 写入“1”无效. 0 没有I2C 中断产生 1 有I2C 中断产生
TXDLYENBn	USInSDHR 寄存器控制位 0 使能 USInSDHR 寄存器 1 禁止 USInSDHR 寄存器
IICnIE	I2C 模式中断使能位 0 I2C 中断被抑制 (use polling) 1 使能 I2C 中断
ACKnEN	控制 ACK 信号产生在 SCLn 第九个周期. 0 没有 ACK 信号产生(SDAn =1) 1 产生 ACK 信号(SDAn =0) NOTE) ACK 信号输出的 3 种情况 (SDA =0). 1. 当收到地址等于 USInSAR 的 USInSLA 位. 2. 在 GCALLn 使能情况下, 当收到地址等于 0x00. 3. 当 I2C 作为接收时(主控或从动)
IMASTERn	I2C模式 0 I2C 主控模式 1 I2C 从动模式
STOPCn	当 I2C 是主控模式时, STOP 命令产生 0 No effect 1 STOP命令产生
STARTCn	当 I2C是主控模式时, START 命令产生 0 No effect 1 START 或重复开始命令产生

USInST1 (USIn 状态寄存器 1: UART 和 SPI 模式): E1H

7	6	5	4	3	2	1	0
DREn	TXCn	RXCn	WAKEn	USInRST	DORn	FEn	PEn
RW	RW	R	RW	RW	R	RW	RW

Initial value: 80H

DREn	DREn 标志位表示发送缓存 (USInDR) 是否已经准备接收新数据. 如果 DREn 是 '1', 缓存器是空的并且准备被写入. 该标志位可产生DREn 中断. 0 发送缓存不是空的. 1 发送缓存是空的.
TXCn	当发送移位寄存器中的全部数据被移出并且没有新的当前数据存在于发送缓存器时该位置位. 当TXCn中断服务程序执行时该位自动清零. 会产生TXCn 中断. 该位会被自动清除. 0 发送进行中. 1 发送缓存是空的并且在发送移位寄存器中的数据被完全移出.
RXCn	当有未读数据在接收缓存器时置位并且在接收缓存器中的所有数据被读取时该位清除. RXCn 标志位用于产生 RXCn 中断. 0 没有未读的数据在接收缓存器 1 至少有一个数据在接收缓存器
WAKEn	当CPU 在STOP 模式下 RXDn 检测到低电平时该位置位. 该位用于产生WAKEn 中断. 该位只有在异步模式时置位. 需要通过软件进行清除(只有于UART) 0 没有 WAKE 中断产生. 1 有WAKE 中断产生
USInRST	这是一个内部复位并且只对USIn有效. 写入'1' 对USIn内部逻辑进行初始化并且自动清零. 0 No operation 1 Reset USIn
DORn	当数据超出时该位置位. 该位置位时, 输入数据被忽略, 直到接收缓存被读取该位一直有效. 0 No Data Overrun 1 Data Overrun detected
FEn	如果接收缓存器中下一个字符的第一个停止位被检测到是0时, 该位置位. 直到接收缓存被读取该位一直有效. (仅用于 UART) 0 No Frame Error 1 Frame Error detected
PEn	奇偶校验使能情况下如果接收缓存器的下一个字符有校验错误, 该位置位. 直到接收缓存器被读取该位一直有效. (只有于 UART) 0 No Parity Error 1 Parity Error detected

USInST2 (USIn 状态寄存器2: 用于 I2C): E2H

7	6	5	4	3	2	1	0
GCALLn	TENDn	STOPDn	SSELn	MLOSTn	BUSYn	TMODEn	RXACKn
R	RW	RW	RW	RW	RW	R	RW

Initial value: 00H

GCALLn	根据I2C主控还是被动模式该位有不同的含义. 主控模式时, 该位代表是否从从机接收 A ACK (address ACK) . 0 No AACK is received (主控模式) 1 AACK is received (主控模式) 当 I2C 是从动模式时, 该位用于指明general call. 0 General call address is not detected (从动模式) 1 General call address is detected (从动模式)
TENDn	当一个字节数据发送完成时该位置位 0 1 byte 数据没有发送完 1 1 byte 数据发送完成
STOPDn	当检测到STOP 命令时该位置位. 0 没有检测到 STOP 命令 1 检测到 STOP 命令
SSELn	I2C 被其他主机编址时该位置位. 0 I2C 不被选为从机 1 I2C 被其他主机编址并担任从机
MLOSTn	该位表示在主动模式时总线仲裁的结果. 0 I2C 维持主控权 1 I2C 在仲裁过程中维持主控权
BUSYn	该位反应了总线的状态 0 I2C 总线空闲, 所以主机可以发送开始命令 1 I2C 总线繁忙
TMODEn	该位用于表示 I2C 是发送还是接收. 0 I2C 接收 1 I2C 发送
RXACKn	该位显示 ACK 信号状态 0 No ACK is received 1 ACK is received at ninth SCL period

NOTE)

1. GCALLn, TENDn, STOPDn, SSELn, and MLOSTn 位可作为中断源.
2. 除了 STOP 模式外如果有 I2C 中断产生时, SCLn 维持在低电平. 要释放 SCLn, 写任意值到 USInST2. 在 USInST2 被写入时, TENDn, STOPDn, SSELn, MLOSTn, 和 RXACKn 位被清除.
3. 写"0"到相应的位时, GCALLn, TENDn, STOPDn, SSELn, and MLOSTn 位将清零.

11.12.25 波特率设置(举例)

Baud Rate (bps)	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR
2400	25	0.2%	47	0.0%	51	0.2%
4800	12	0.2%	23	0.0%	25	0.2%
9600	6	-7.0%	11	0.0%	12	0.2%
14.4k	3	8.5%	7	0.0%	8	-3.5%
19.2k	2	8.5%	5	0.0%	6	-7.0%
28.8k	1	8.5%	3	0.0%	3	8.5%
38.4k	1	-18.6%	2	0.0%	2	8.5%
57.6k	-	-	1	-25.0%	1	8.5%
76.8k	-	-	1	0.0%	1	-18.6%
115.2k	-	-	-	-	-	-
230.4k	-	-	-	-	-	-

Baud Rate (bps)	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR
2400	25	0.2%	47	0.0%	51	0.2%
4800	12	0.2%	23	0.0%	25	0.2%
9600	6	-7.0%	11	0.0%	12	0.2%
14.4k	3	8.5%	7	0.0%	8	-3.5%
19.2k	2	8.5%	5	0.0%	6	-7.0%
28.8k	1	8.5%	3	0.0%	3	8.5%
38.4k	1	-18.6%	2	0.0%	2	8.5%
57.6k	-	-	1	-25.0%	1	8.5%
76.8k	-	-	1	0.0%	1	-18.6%
115.2k	-	-	-	-	-	-
230.4k	-	-	-	-	-	-

Baud Rate (bps)	fx=8.00MHz	
	USI0BD/USI1BD	ERROR
2400	207	0.2%
4800	103	0.2%
9600	51	0.2%
14.4k	34	-0.8%
19.2k	25	0.2%
28.8k	16	2.1%
38.4k	12	0.2%
57.6k	8	-3.5%
76.8k	6	-7.0%
115.2k	3	8.5%
230.4k	1	8.5%
250k	1	0.0%
0.5M	-	-
1M	-	-

Table 11.22 Examples of USI0BD and USI1BD Settings for Commonly Used Oscillator Frequencies

11.13 LCD 驱动

11.13.1 概述

LCD 驱动通过LCD 控制寄存器 (LCDCRH/L)控制。 LCLK[1:0] 决定COM信号扫描每个段选输出的频率。复位将LCD CRH 和 LCDCRL的值清除到 '0'。

如果子时钟被用作系统时钟时LCD 显示在IDLE 和 STOP 模式时可继续运行

11.13.2 LCD Display RAM Organization

显示数据存储在外部数据存储器的显示区域。

存储在外部数据区域的显示数据被自动读取，并且通过团建发送给LCD 驱动。 LCD 驱动产生和显示数据及驱动函数一样的段选信号和共用端信号。因此，显示只能被外部数据显示区域的覆盖数据改变

图 11-99 所示为外部数据区域和 COM/SEG 脚之间的通讯。当显示数据时1时，LCD打开，是0时关闭。

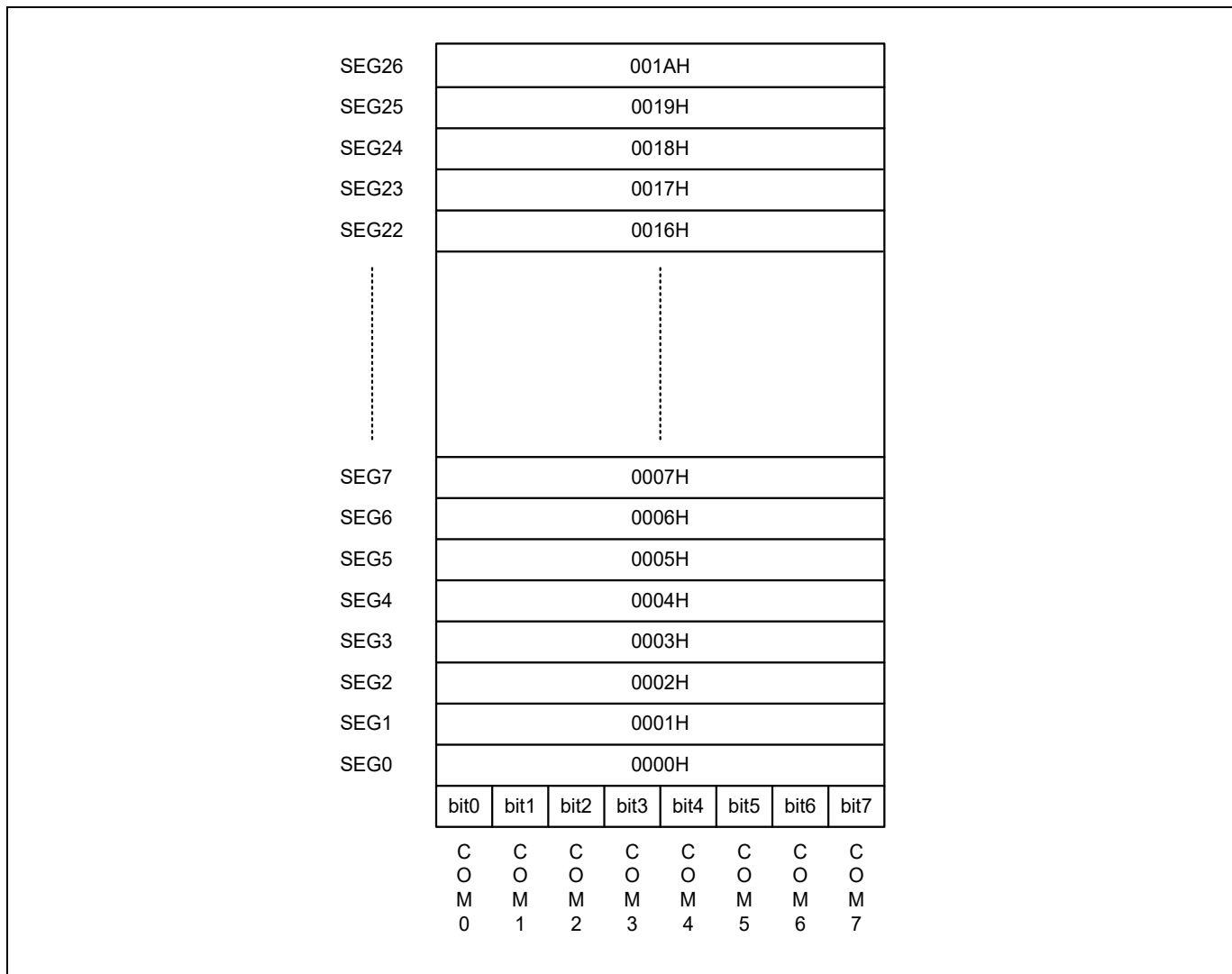


Figure 11.74 LCD Circuit Block Diagram

11.13.3 LCD 信号波形

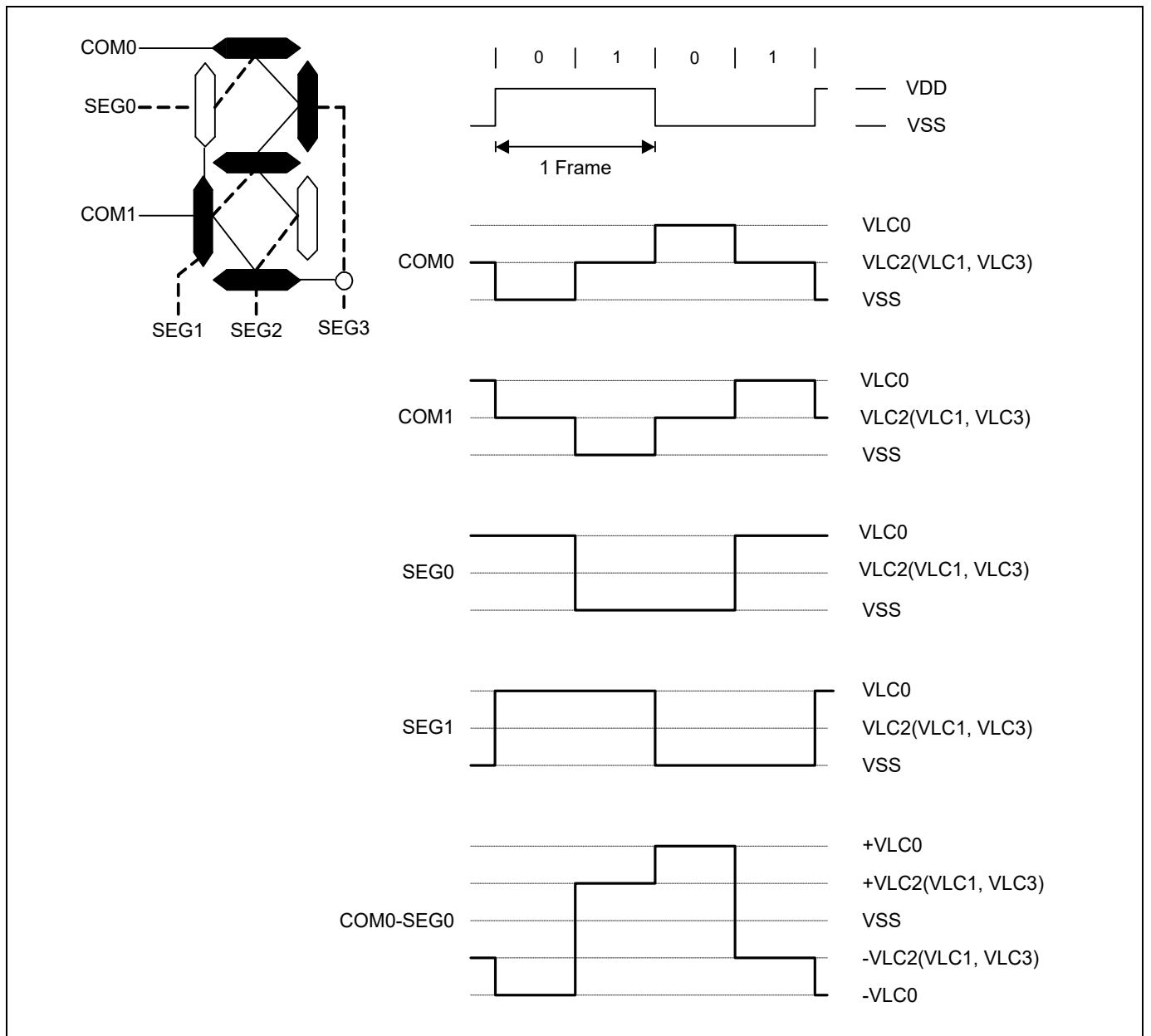


Figure 11.75 LCD 信号波形 (1/2Duty, 1/2Bias)

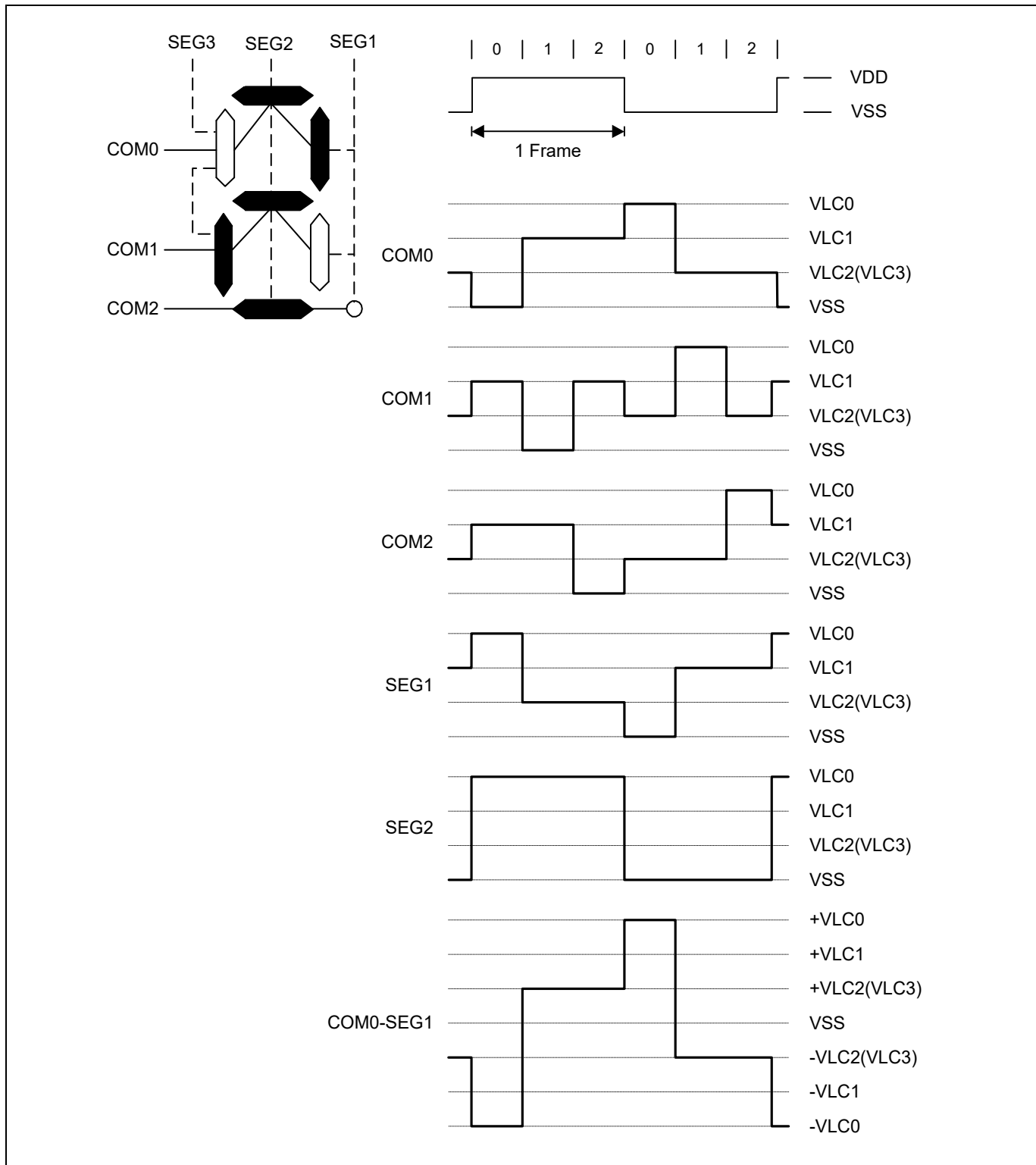


Figure 11.76 LCD 信号波形 (1/3Duty, 1/3Bias)

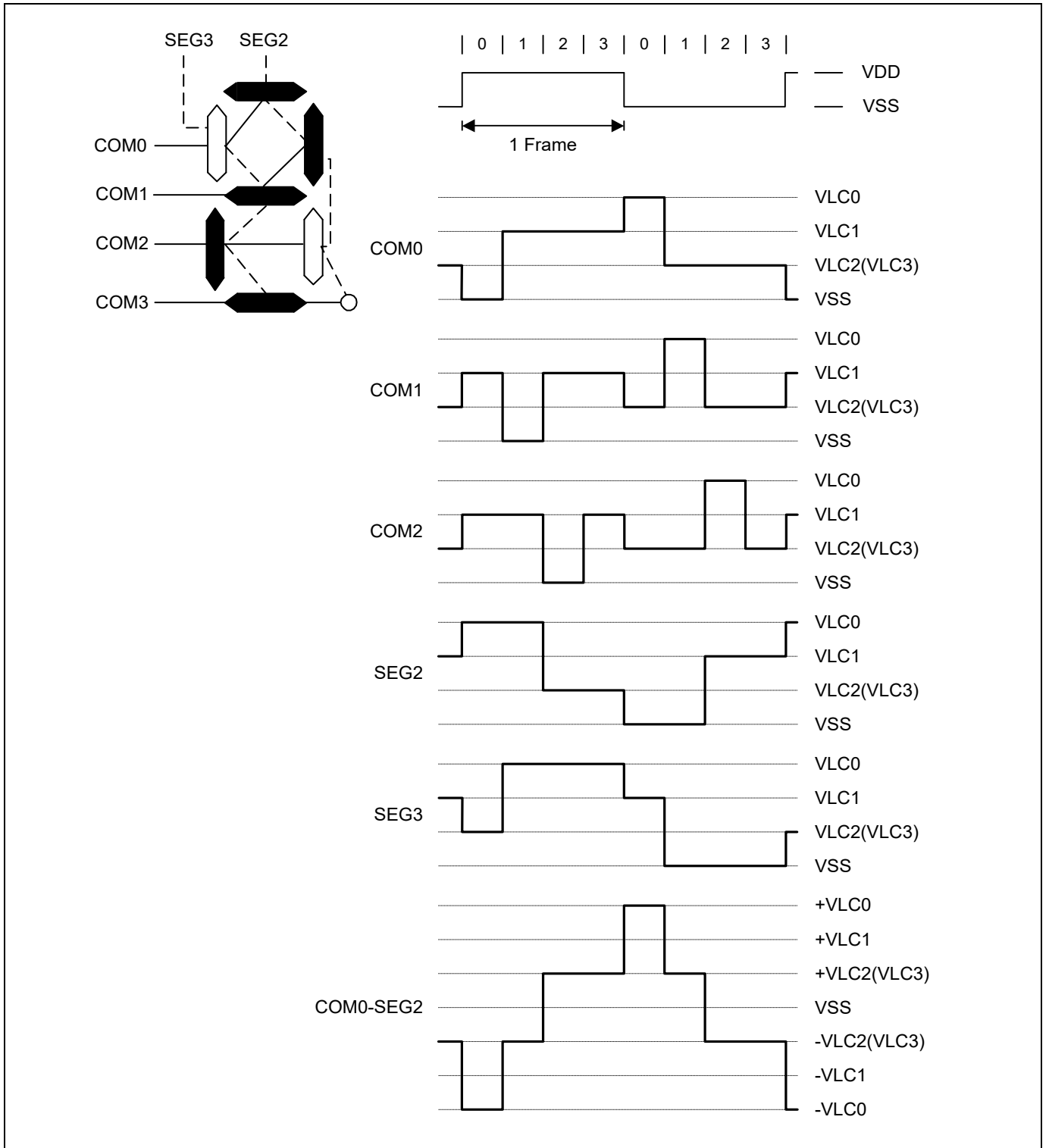


Figure 11.77 LCD 信号波形 (1/4Duty, 1/3Bias)

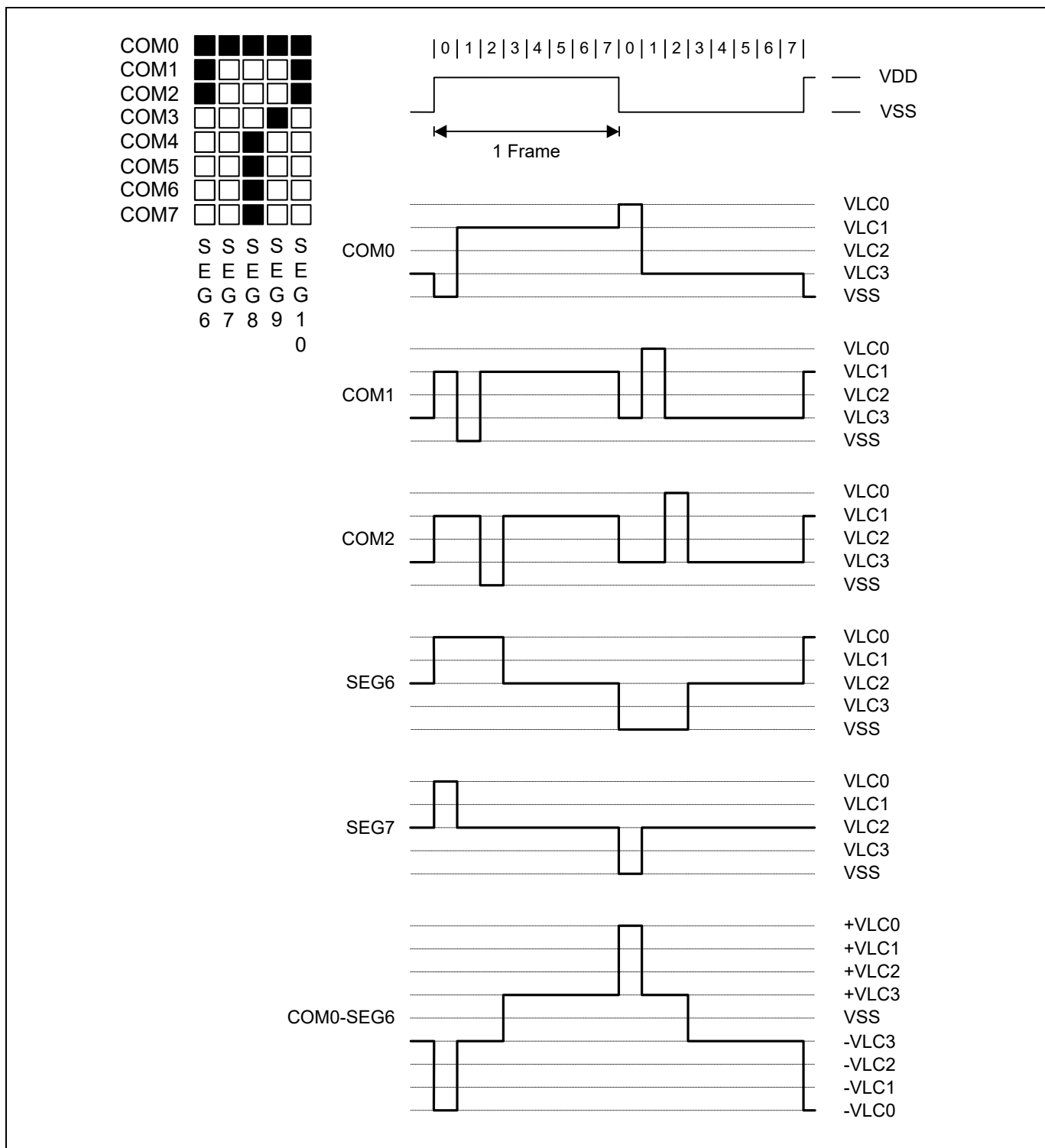


Figure 11.78 LCD 信号波形 (1/8Duty, 1/4Bias)

11.13.4 LCD 分压电阻连接

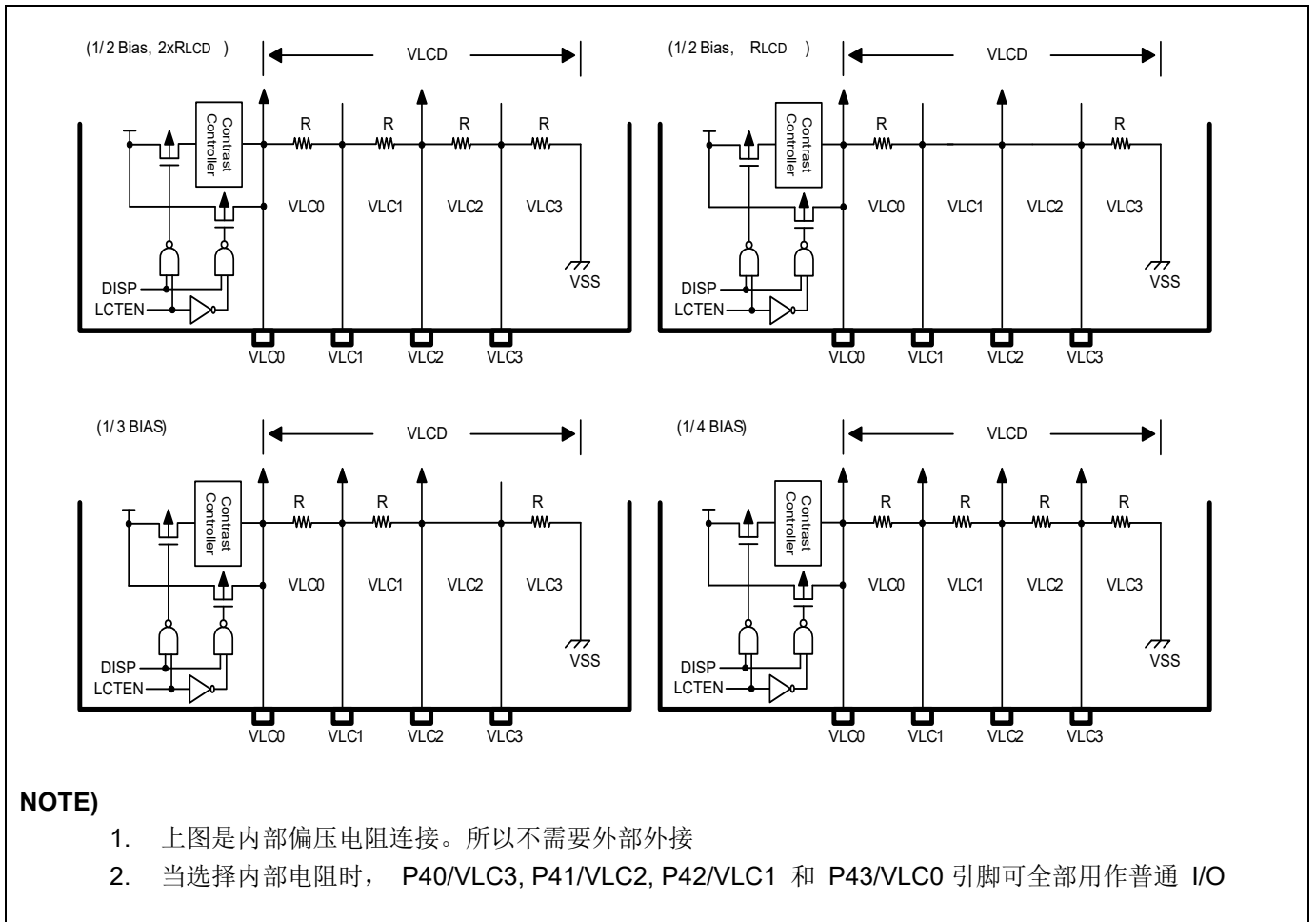
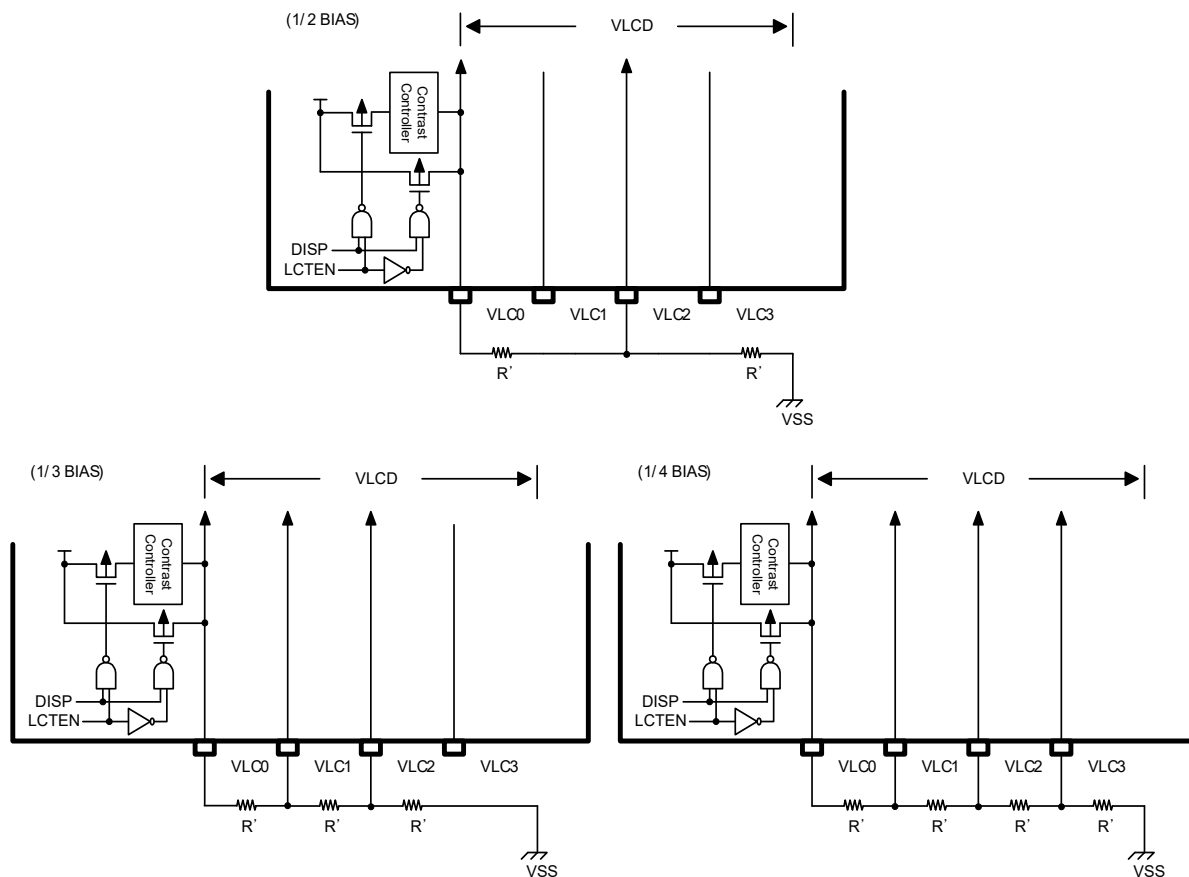


Figure 11.79 内部偏压电阻连接



NOTE)

1. 选择外部偏压电阻时，内部偏压电阻不连接.选择外部偏压电阻时，分压电阻应该依照图上连接
2. 同时所需用到的分压脚通过 P4FSR 选择为 LCD 偏压功能 (VLC0, VLC1, VLC2, and VLC3).
3. 选择 1/2 bias, P43/VLC0 和 P41/VLC2 脚选择为 VLC0 和 VLC2 功能.其他脚可以用作普通 I/O
4. 选择 1/3 bias, P43/VLC0, P42/VLC1, 和 P41/VLC2 选为 VLC0, VLC1, 和 VLC2 功能. 另一个脚可以用作普通 I/O.
5. 选择 1/4 bias, P43/VLC0, P42/VLC1, P41/VLC2, 和 P40/VLC3 选为 VLC0,VLC1, VLC2, 和 VLC3 功能

Figure 11.80 外部偏压电阻连接

11.13.5 方框图

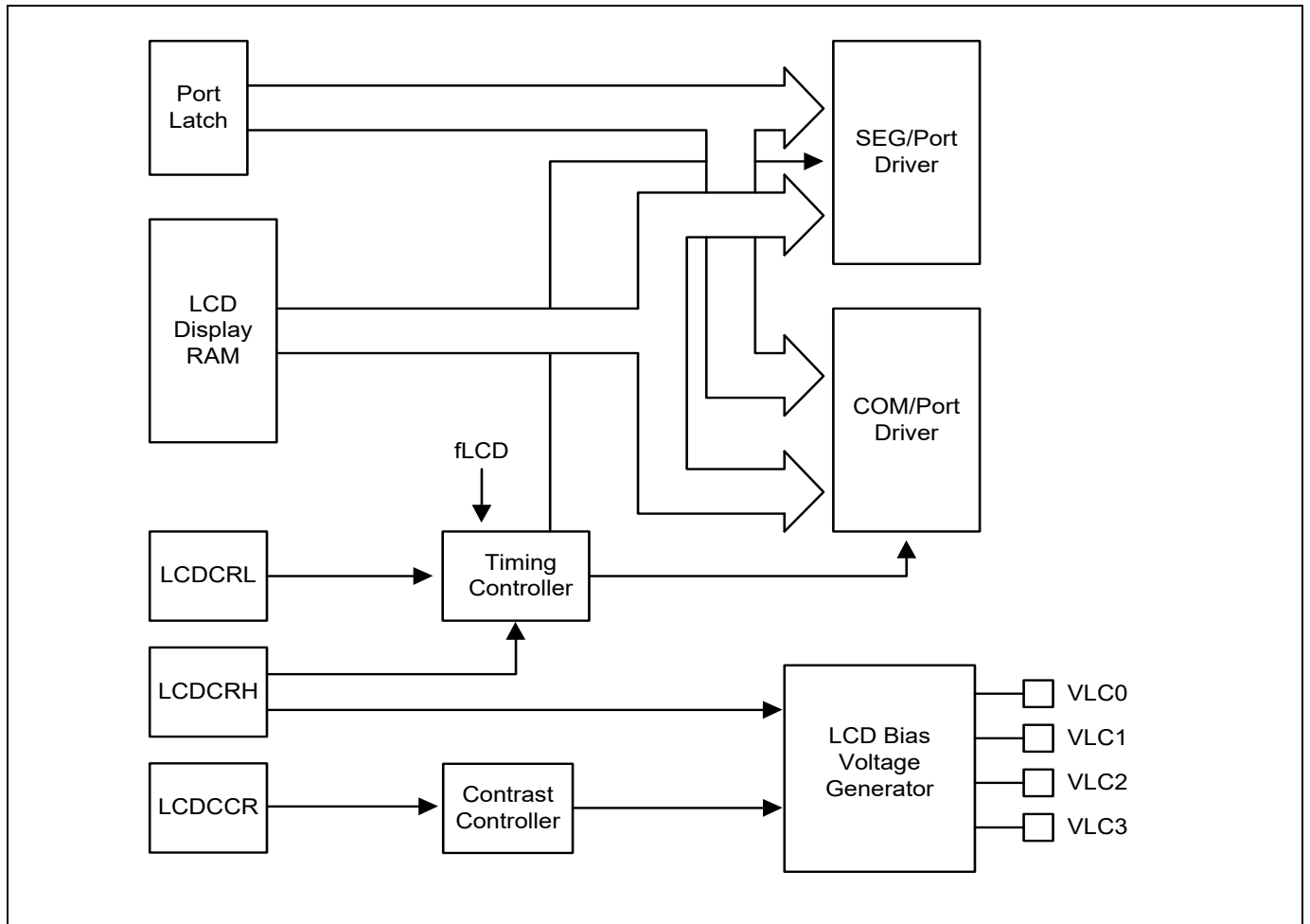


Figure 11.81 LCD 电路方框图

11.13.6 寄存器图

Name	Address	Direction	Default	Description
LCDCRH	9AH	R/W	00H	LCD 驱动控制高位寄存器
LCDCRL	99H	R/W	00H	LCD 驱动控制低位寄存器
LCDCCR	9BH	R/W	00H	LCD 驱动对比度控制寄存器

Table 11.23 LCD 寄存器

11.13.7 LCD 驱动寄存器说明

LCD 驱动有两个控制寄存器，LCDCRH, LCDCRL和 LCDCCR.

11.13.8 LCD 驱动寄存器说明

LCDCRH (LCD 驱动控制高位寄存器): 9AH

7	6	5	4	3	2	1	0
-	-	-	COMCHG	-	-	LCDDR	DISP
-	-	-	RW	-	-	RW	RW

Initial value: 00H

- COMCHG** Common 信号输出口更改控制
- 0 COM0 – COM3 信号通过 P37-P34 输出
 - 1 COM0 – COM3 信号通过 P33-P30 输出

NOTE)

1. COM0/COM1/COM2/COM3 信号可以分别通过 P33/P32/P31/P30 输出.
2. 例如, COM0 信号可通过 P33 输出如果 P3FSR.3 设置为 “1b” COMCHG 位设置为 “1b”.
3. 参考 P3 功能选择寄存器 (P3FSR).
4. 仅适用于 1/4 duty 以下.

- LCDDR** LCD 驱动寄存器Bias选择
- 0 内部 LCD 偏压
 - 1 外部 LCD 偏压

- DISP** LCD 显示控制
- 0 禁止
 - 1 正常显示

LCDCRL (LCD 驱动控制地位寄存器): 99H

7	6	5	4	3	2	1	0
IRSEL	-	DBS3	DBS2	DBS1	DBS0	LCLK1	LCK0
RW	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

IRSEL Internal LCD Bias Dividing Resistor Select
 0 $R_{LCD} = 60k\Omega (R_{LCD1})$
 1 $R_{LCD} = 120k\Omega (R_{LCD2})$

DBS[3:0] LCD Duty and Bias Select (NOTE)

DBS3	DBS2	DBS1	DBS0	Description
0	0	0	0	1/8Duty, 1/4Bias, R_{LCD}
0	0	0	1	1/6Duty, 1/4Bias, R_{LCD}
0	0	1	0	1/5Duty, 1/3Bias, R_{LCD}
0	0	1	1	1/4Duty, 1/3Bias, R_{LCD}
0	1	0	0	1/3Duty, 1/3Bias, R_{LCD}
0	1	0	1	1/3Duty, 1/2Bias, R_{LCD}
0	1	1	0	1/3Duty, 1/2Bias, $2xR_{LCD}$
0	1	1	1	1/2Duty, 1/2Bias, R_{LCD}
1	0	0	0	1/2Duty, 1/2Bias, $2xR_{LCD}$

 Other values Not available

LCLK[1:0] LCD Clock Select (When f_{wck} (Watch timer clock)= 32.768kHz)

LCLK1	LCLK0	Description
0	0	$f_{LCD} = 128Hz$
0	1	$f_{LCD} = 256Hz$
1	0	$f_{LCD} = 512Hz$
1	1	$f_{LCD} = 1024Hz$

NOTE)

1. LCD 设置通过 watch timer (f_{wck})产生. 所以 LCD 打开时 watch timer 应该使能.

LCD Clock Frequency (f_{LCD})	LCD Frame Frequency (f_{FRAME})						Unit
	1/2 Duty	1/3 Duty	1/4 Duty	1/5 Duty	1/6 Duty	1/8 Duty	
128	64	43	32	26	21	16	Hz
256	128	85	64	51	43	32	
512	256	171	128	102	85	64	
1024	512	341	256	205	171	128	

Table 11.24 LCD Frame Frequency

LCD 频率通过以下公式计算:

$$LCD \text{ Frame Frequency } (f_{FRAME}) = f_{LCD} \times \text{Duty Hz}$$

Ex) 如果是 1/4 duty $f_{LCD}=512Hz$, $f_{FRAME} = f_{LCD} \times 1/4 = 512 \times 1/4 = 128Hz$

LCDCCR (LCD 驱动对比度控制低位寄存器): 9BH

7	6	5	4	3	2	1	0
LCTEN	-	-	-	VLCD3	VLCD	VLCD1	VLCD0
RW	-	-	-	RW	RW	RW	RW

Initial value: 00H

LCTEN 控制 LCD 驱动对比度

0 LCD 驱动对比度禁用

1 LCD 驱动对比度启用

VLCD[3:0] VLC0 Voltage Control when the contrast is enabled

VLCD3	VLCD 2	VLCD 1	VLCD 0	Description
0	0	0	0	VLC0 = VDD x 16/31 step
0	0	0	1	VLC0 = VDD x 16/30 step
0	0	1	0	VLC0 = VDD x 16/29 step
0	0	1	1	VLC0 = VDD x 16/28 step
0	1	0	0	VLC0 = VDD x 16/27 step
0	1	0	1	VLC0 = VDD x 16/26 step
0	1	1	0	VLC0 = VDD x 16/25 step
0	1	1	1	VLC0 = VDD x 16/24 step
1	0	0	0	VLC0 = VDD x 16/23 step
1	0	0	1	VLC0 = VDD x 16/22 step
1	0	1	0	VLC0 = VDD x 16/21 step
1	0	1	1	VLC0 = VDD x 16/20 step
1	1	0	0	VLC0 = VDD x 16/19 step
1	1	0	1	VLC0 = VDD x 16/18 step
1	1	1	0	VLC0 = VDD x 16/17 step
1	1	1	1	VLC0 = VDD x 16/16 step

NOTE) The LCD contrast step is based on 1/4 bias and R_{LCD}=60kΩ.

1. 1/4 bias : $VDD \times 16 / (31 - VLC[3:0])$ when $R_{LCD} = 60k\Omega$
 $VDD \times 32 / (47 - VLC[3:0])$ when $R_{LCD} = 120k\Omega$
5. 1/3 bias : $VDD \times 12 / (27 - VLC[3:0])$ when $R_{LCD} = 60k\Omega$
 $VDD \times 24 / (39 - VLC[3:0])$ when $R_{LCD} = 120k\Omega$
6. 1/2 bias : $VDD \times 8 / (23 - VLC[3:0])$ when $R_{LCD} = 60k\Omega$
 $VDD \times 16 / (31 - VLC[3:0])$ when $R_{LCD} = 120k\Omega$

12 省电操作

12.1 概述

MC96F6432A 有两个省电模式减少设备电量消耗. 在省电模式中, 电量消耗很多。设备提供3种节电功能Main-IDLE, Sub-IDLE 和 STOP模式. 在3中模式下, 程序停止.

12.2 IDLE/STOP 模式下外围操作

Peripheral	IDLE Mode	STOP Mode
CPU	ALL CPU Operation are Disable	ALL CPU Operation are Disable
RAM	Retain	Retain
Basic Interval Timer	Operates Continuously	Stop
Watch Dog Timer	Operates Continuously	Stop (Can be operated with WDTRC OSC)
Watch Timer	Operates Continuously	Stop (Can be operated with sub clock)
Timer0~4	Operates Continuously	Halted (Only when the Event Counter Mode is Enabled, Timer operates Normally)
ADC	Operates Continuously	Stop
BUZ	Operates Continuously	Stop
SPI	Operates Continuously	Only operate with external clock
USI0/1	Operates Continuously	Only operate with external clock
LCD Controller	Operates Continuously	Stop (Can be operated with sub clock)
Internal OSC (16MHz)	Oscillation	Stop when the system clock (fx) is fIRC
WDTRC OSC (5kHz)	Can be operated with setting value	Can be operated with setting value
Main OSC (0.4~8.5MHz)	Oscillation	Stop when fx = fXIN
Sub OSC (32.768kHz)	Oscillation	Stop when fx = fSUB
I/O Port	Retain	Retain
Control Register	Retain	Retain
Address Data Bus	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, Timer Interrupt (EC0, EC1, EC3), SPI (External clock), External Interrupt, UART by RX, WT (sub clock), WDT

Table 12.1 省电模式外围操作

12.3 IDLE 模式

电源控制寄存器设置为 '01h' 进入到 IDLE 模式. 该模式下, 内部振荡电路保持活动. 振荡继续同时外围设备也正常操作, 但是CPU 停止. 通过复位或振荡来唤醒. 中断唤醒时, 中断需要在IDLE模式前使能. 如果使用复位, 由于设备变为初始化状态, 寄存器也是复位值.

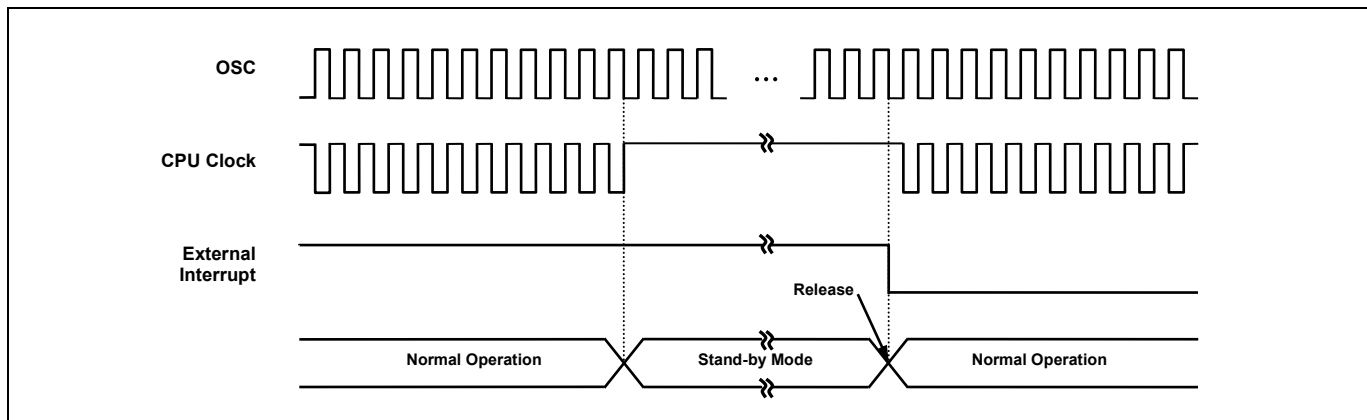


Figure 12.1 外部中断唤醒 IDLE 模式时序图

12.4 STOP Mode

电源控制寄存器设置为 '03H'进入 STOP 模式. 在该模式下, 所选振荡器, 系统时钟或外围时钟停止, 但是watch timer 可以继续通过sub clock继续操作. 随着时钟的冻结, 所有功能都停止, 只有片内RAM 和控制寄存器仍然保持. 例如, 如果内部RC (f_{IRC}) 被选做系统时钟并且sub clock (f_{SUB})是振荡的, 在STOP模式时内部RC停止振荡, sub clock仍然继续振荡. 此时, watch timer可以通过 sub clock继续操作

STOP模式的唤醒源有硬件复位和中断. 复位时重新定义所有控制寄存器. 退出STOP 时, 需要足够的时间使振荡稳定到正常操作. 图 12.2所示为时序图. 当从STOP退出时, 唤醒情况下基本间隔定时器是激活的, 所以, 在STOP指令之前, 用户必须设置它的有关分频使其有足够长的时间, 这是保证振荡器开始时稳定振荡.

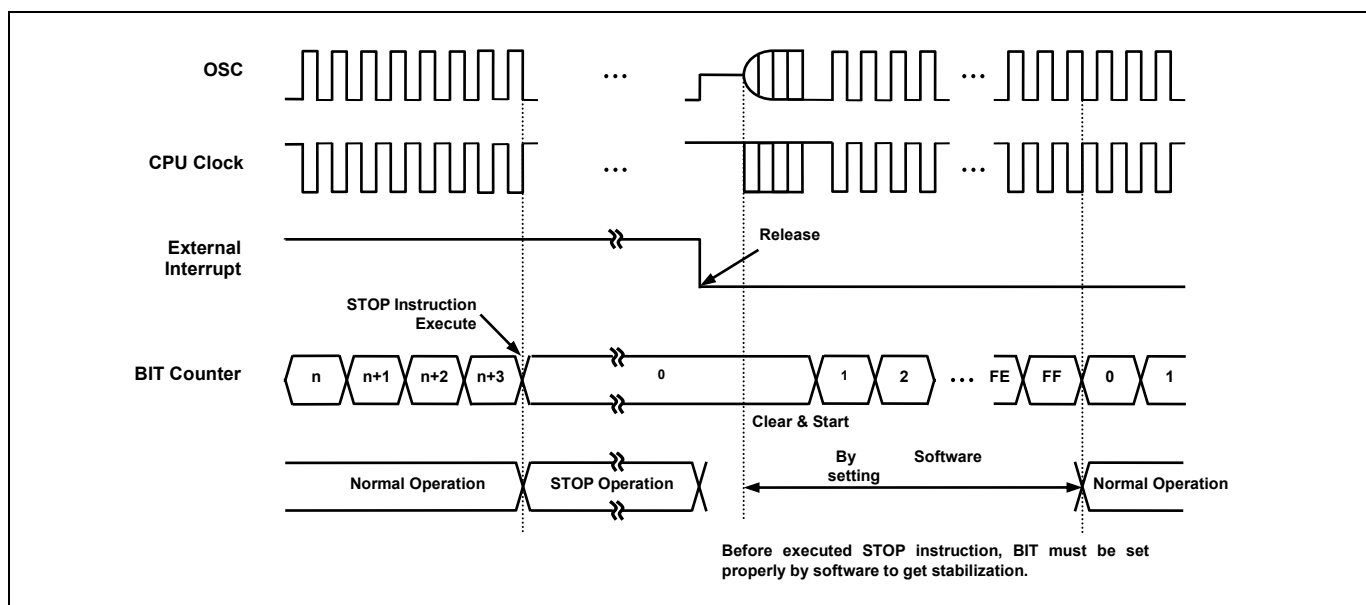


Figure 12.2 外部中断唤醒 STOP 模式

12.5 STOP 模式唤醒

退出STOP 模式之后, 操作根据进入STOP 模式前的中断寄存器开始进行 (图 12.3). 如果总中断使能位(IE.EA)设置为`1`, 唤醒STOP 的中断使能位flag = `1`, 同时 CPU 跳到相应的中断服务程序。即使IE.EA 被清除为`0`, 唤醒 STOP 的中断的使能位设置为`1`.

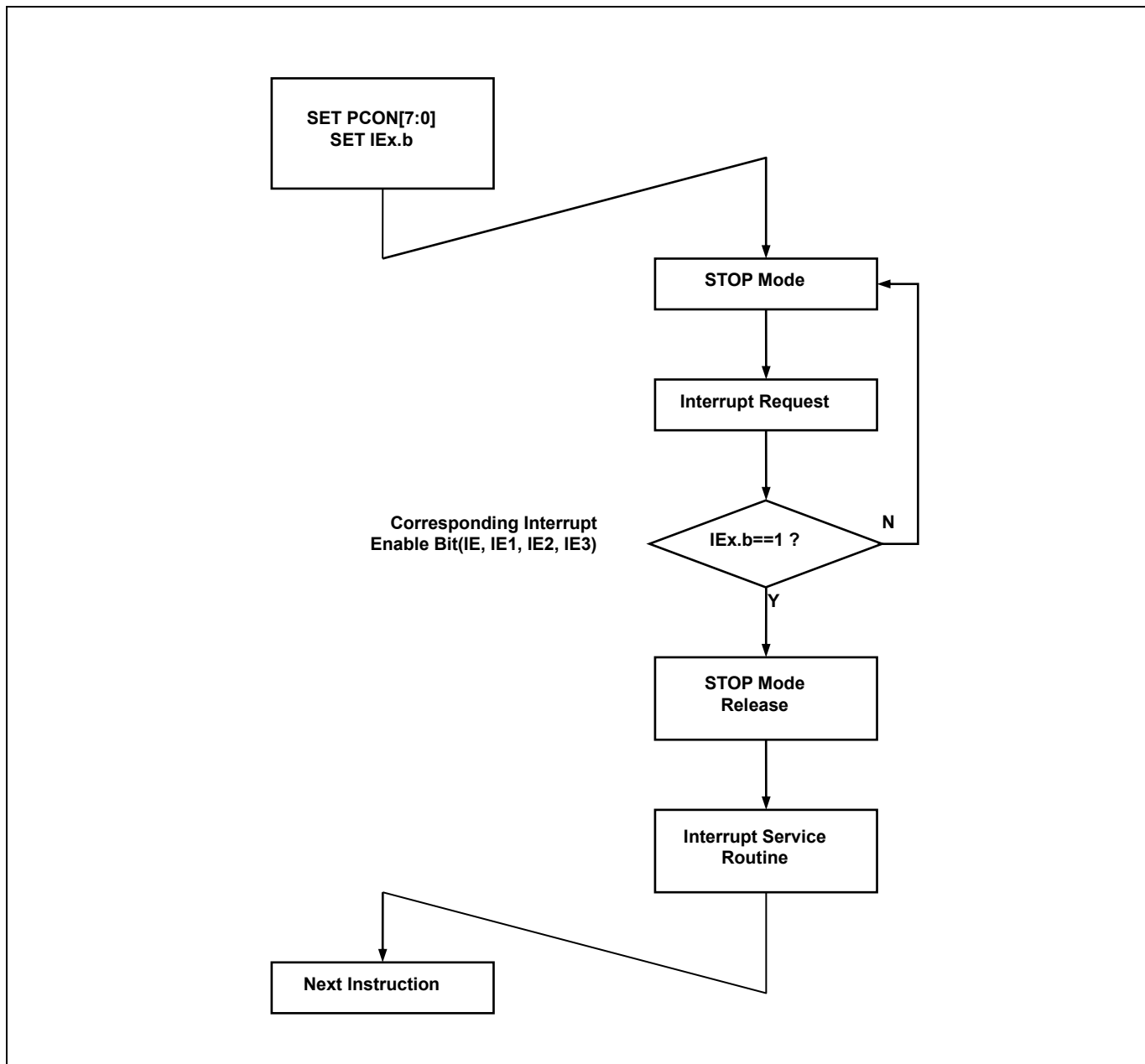


Figure 12.3 STOP 模式唤醒流程

12.6 寄存器图

Name	Address	Direction	Default	Description
PCON	87H	R/W	00H	电源控制寄存器

Table 12.2 省电操作寄存器

12.7 省电操作寄存器说明

省电模式寄存器由电源控制寄存器组成 (PCON).

12.8 省电操作寄存器说明

PCON (电源控制寄存器): 87H

7	6	5	4	3	2	1	0
PCON7	–	–	–	PCON3	PCON2	PCON1	PCON0
RW	–	–	–	RW	RW	RW	RW

Initial value: 00H

PCON[7:0] Power Control
 01H IDLE 模式使能
 03H STOP 模式使能
 Other Value Normal operation
 s

NOTE)

1. 进入 IDLE 模式, PCON 必须设置为 '01H'.
2. 进入 STOP 模式, PCON 必须设置为 '03H'.
3. PCON 通过退出 STOP/IDLE 信号寄存器自动清除..
4. 跟随 STOP/IDLE 模式之后必须立即写入三个以上的 NOP 指令..

```
Ex1) MOV  PCON, #01H    ; IDLE mode
      NOP
      NOP
      NOP
      .
      .
      .
```

```
Ex2) MOV  PCON, #03H    ; STOP mode
      NOP
      NOP
      NOP
      .
      .
      .
```

13 复位

13.1 概述

以下是硬件设置值.

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Refer to the Peripheral Registers

Table 13.1 Reset State

13.2 复位源

MC96F6432A 有5种类型的复位源. 以下是复位源.

- External RESETB
- Power ON RESET (POR)
- WDT 溢出复位(WDTEN = `1`)
- 低电压复位 (LVREN = `0`)

13.3 RESET 方框图

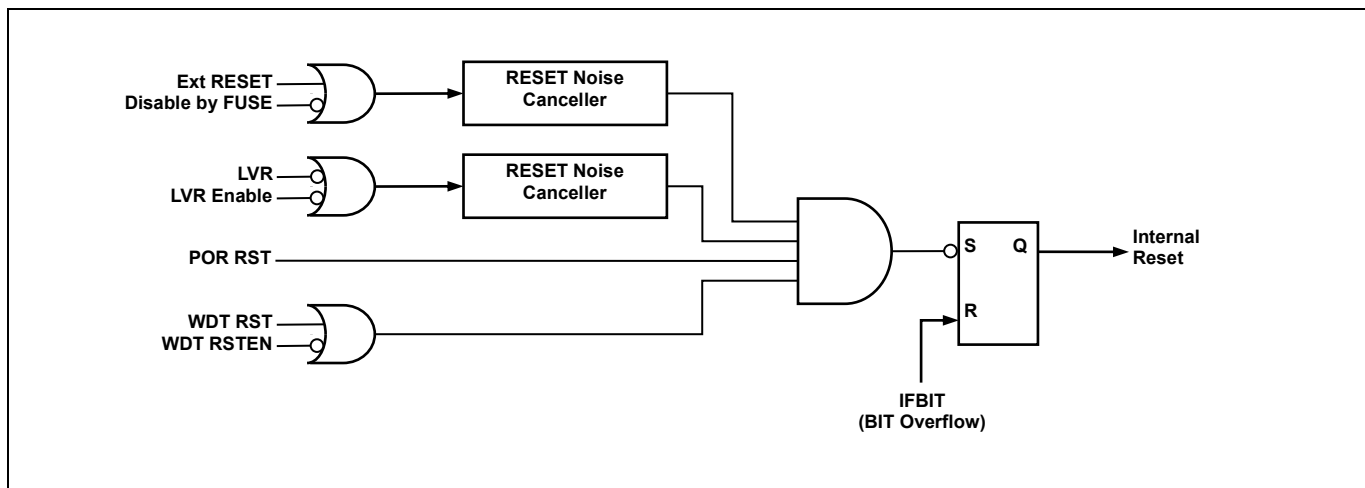


Figure 13.1 RESET 方框图

13.4 RESET 干扰消除

图 13.2是噪音消除器复位噪音消除图解。噪音消除值大概2us (@V_{DD}=5V) 到系统复位输入低时.

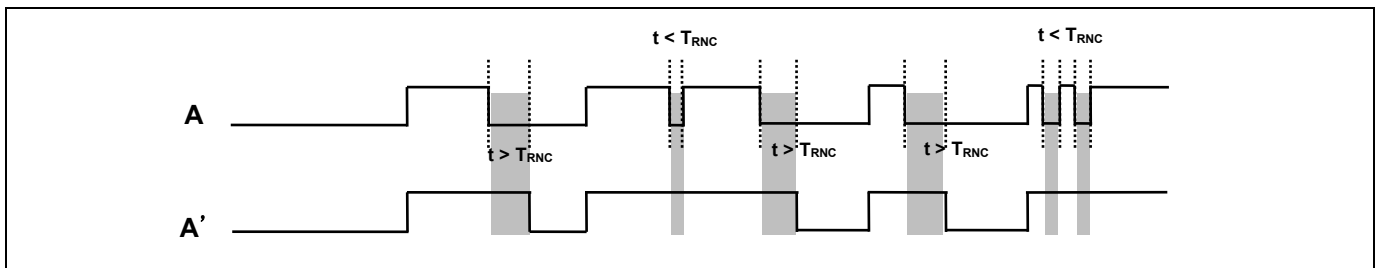


Figure 13.2 Reset noise canceller timer diagram

13.5 上电复位

上电时, POR (Power On Reset) 有复位设备的功能, 如果用到POR, 会重置设备而不是复位芯片或复位电路.

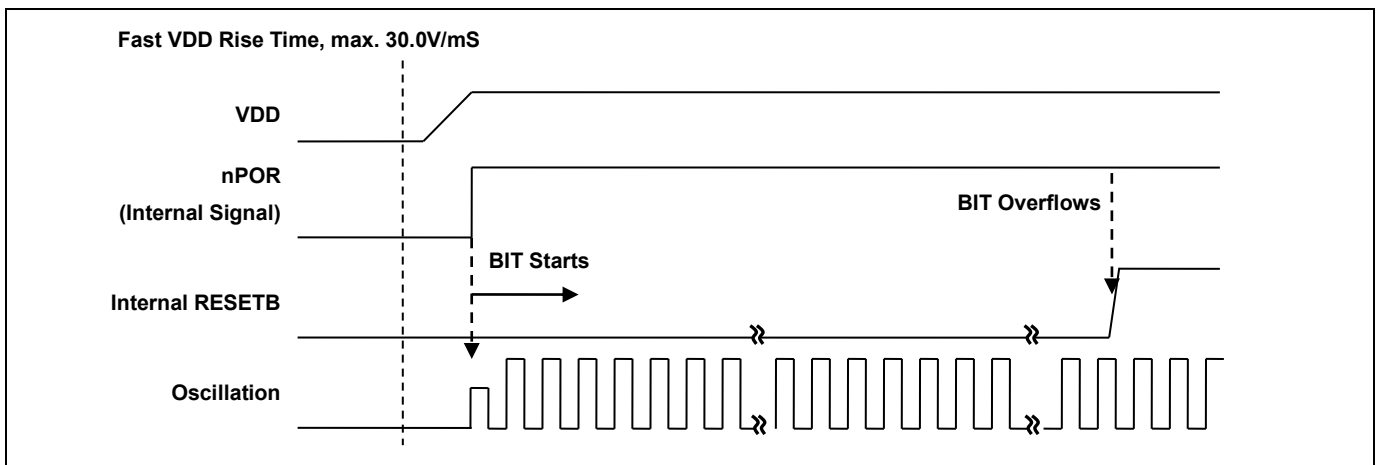


Figure 13.3 VDD 快速上电时序

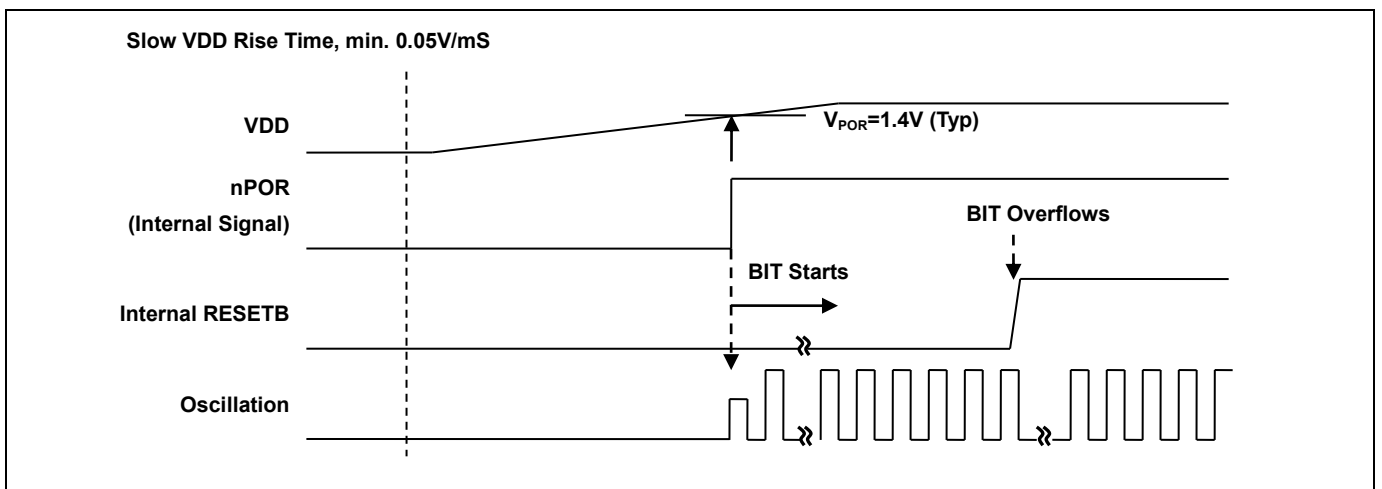


Figure 13.4 电压升高内部复位唤醒时序

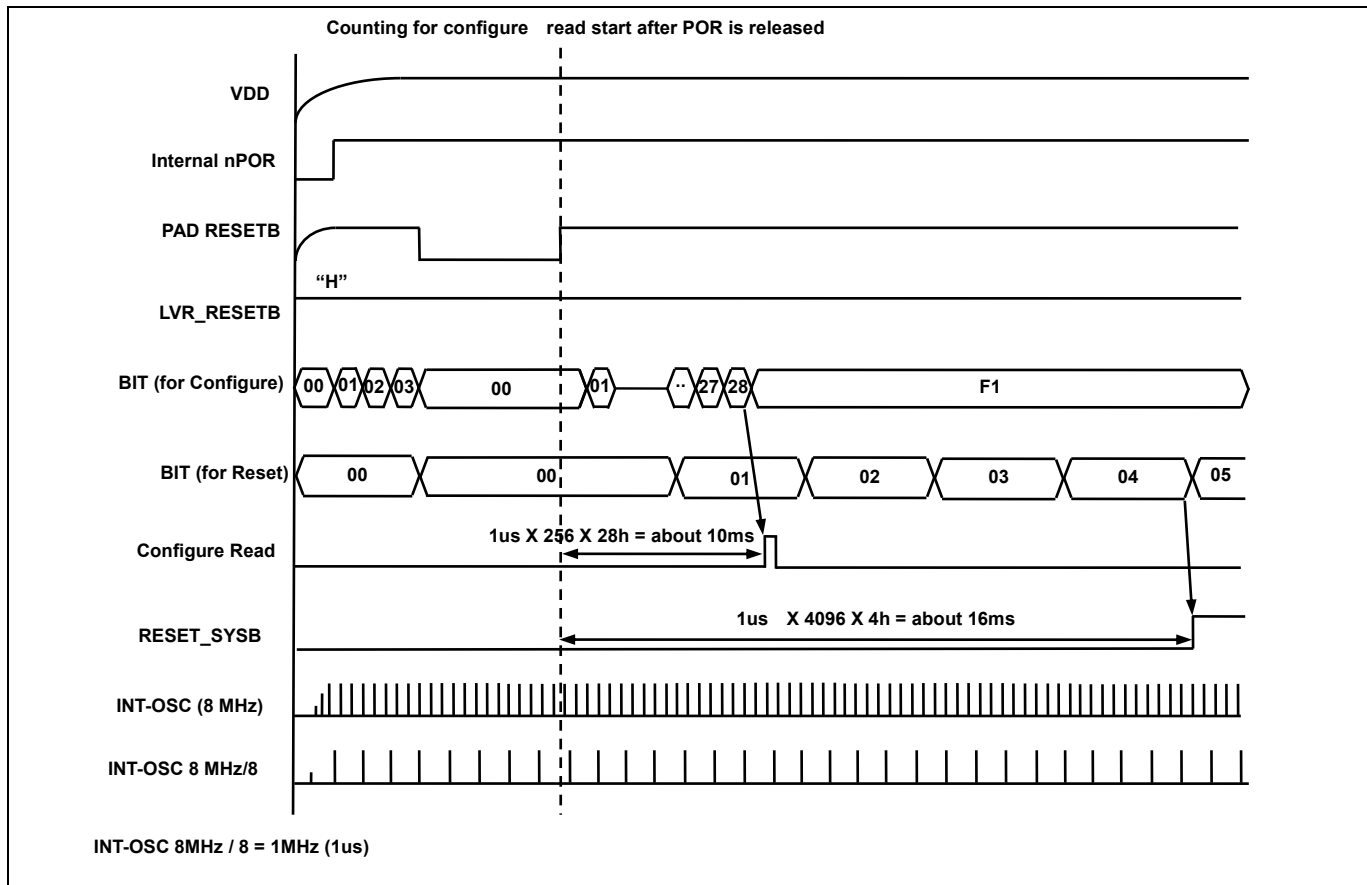


Figure 13.5 上电组态时序

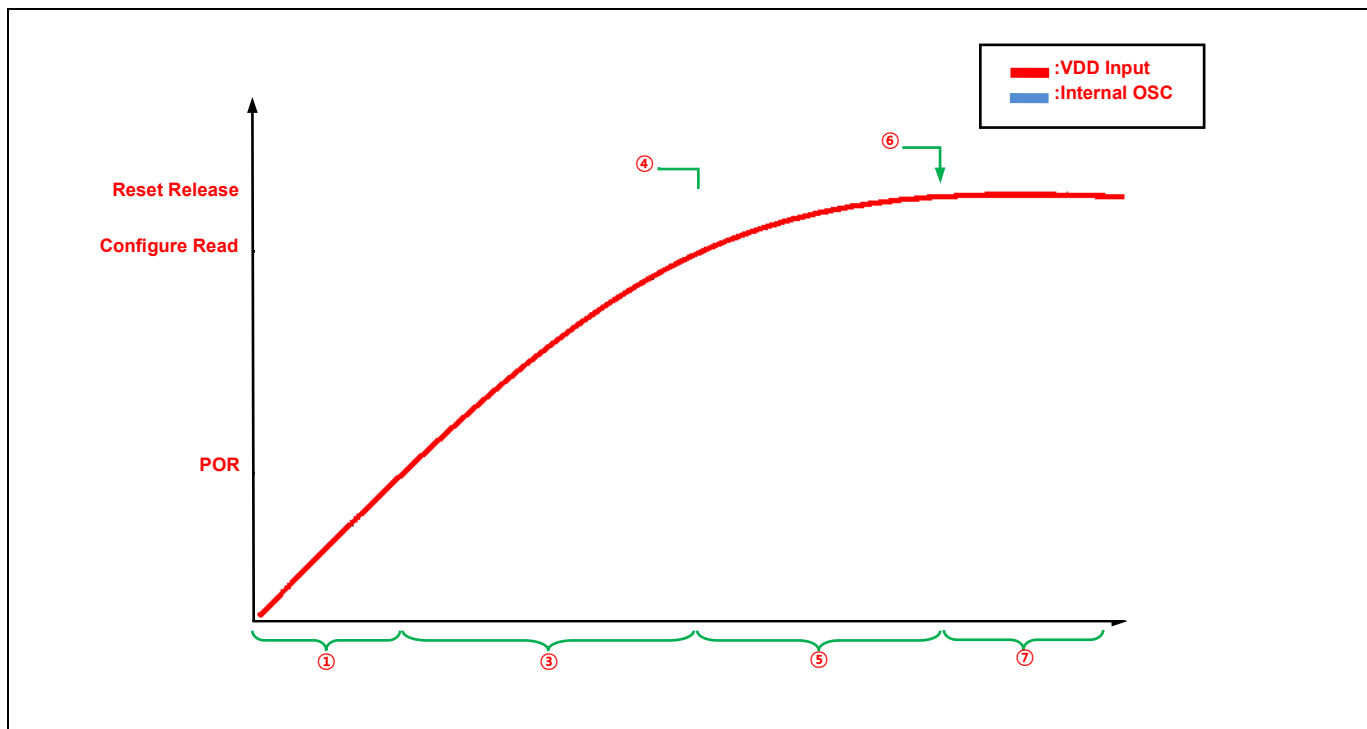


Figure 13.6 启动过程波形图

Process	Description	Remarks
①	- No Operation	
②	-1st POR level Detection	-about 1.4V
③	- (INT-OSC 8MHz/8)x256x28h Delay section (=10ms) -VDD input voltage must rise over than flash operating voltage for Configure option read	-Slew Rate $\geq 0.05V/ms$
④	- Configure option read point	-about 1.5V ~ 1.6V -Configure Value is determined by Writing Option
⑤	- Rising section to Reset Release Level	-16ms point after POR or Ext_reset release
⑥	- Reset Release section (BIT overflow) i) after 16ms, after External Reset Release (External reset) ii) 16ms point after POR (POR only)	- BIT is used for Peripheral stability
⑦	-Normal operation	

Table 13.2 Boot Process Description

13.6 外部复位

外部复位是施密特触发器输入. 如果 在操作电压范围及稳定振荡情况下RESETB 脚保持低电平10us 以上, 就会请求复位, 内部状态被初始化. 复位状态变为‘1’后, 需要16ms 的稳定时间, 内部 RESET 变为 ‘1’. 复位过程需要5个振荡时钟. 程序在向量地址0000H 处开始.

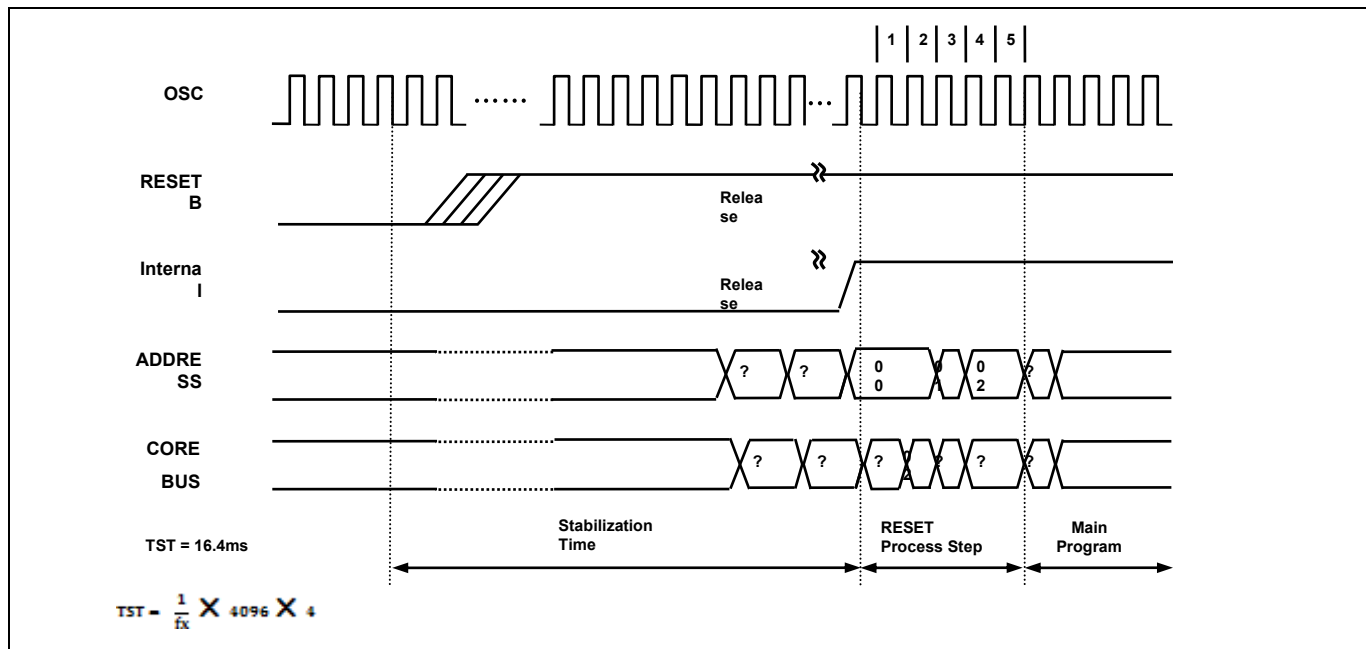


Figure 13.7 复位后时序图

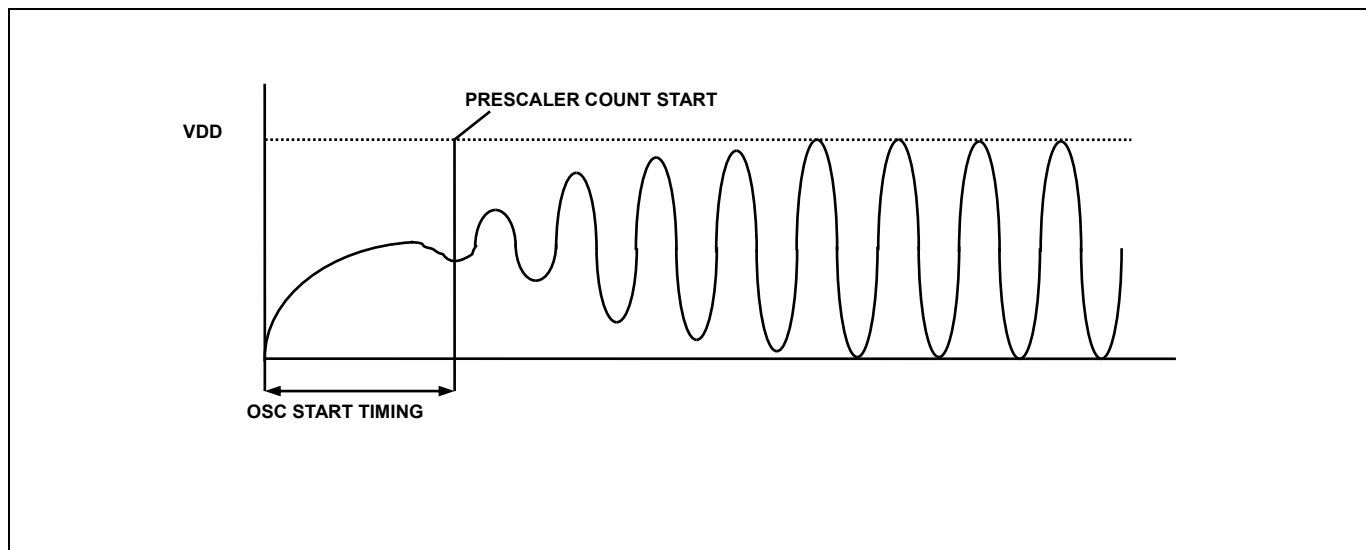


Figure 13.8 振荡器产生波形举例

NOTE)

1. 如图 13.8 所示, 稳定时间不包括启动时间.
2. RESETB 引脚有上拉寄存器

13.7 低电压检测处理

MC96F6432A 有片内brown-out detection circuit (BOD) 通过和固定的触发电平相比较来监测VDD 电平. 触发电平可以通过LVRVS[3:0]设置为 1.85V, 2.20V,2.32V, 2.44V, 2.59V, 2.75V, 2.93V, 3.14V, 3.38V, 3.67V, 4.00V, 4.40V. 在 STOP 模式时, 它将会加大整个电流消耗, 所以为了减小电流消耗, LVREN 位被软件设定在关闭状态.

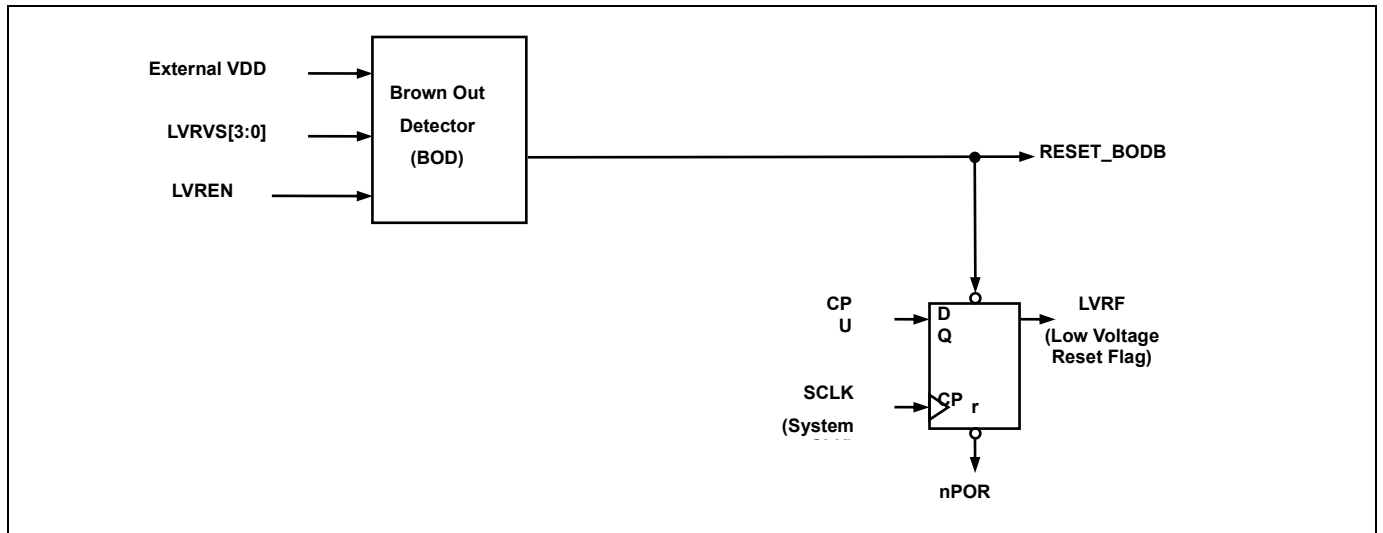


Figure 13.9 BOD 方框图

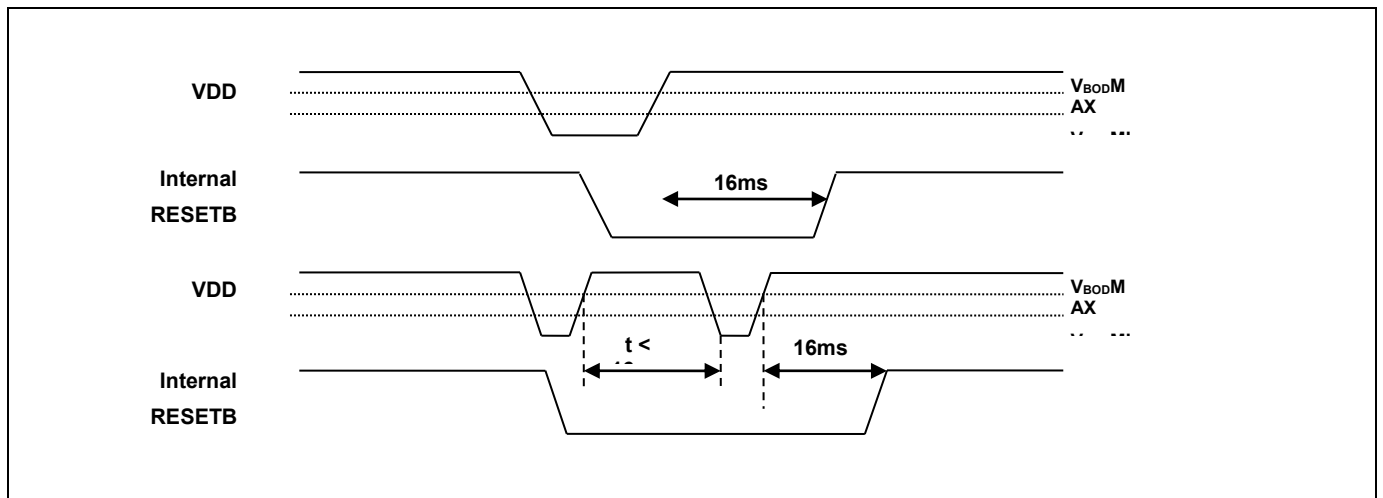


Figure 13.10 电源故障时内部复位

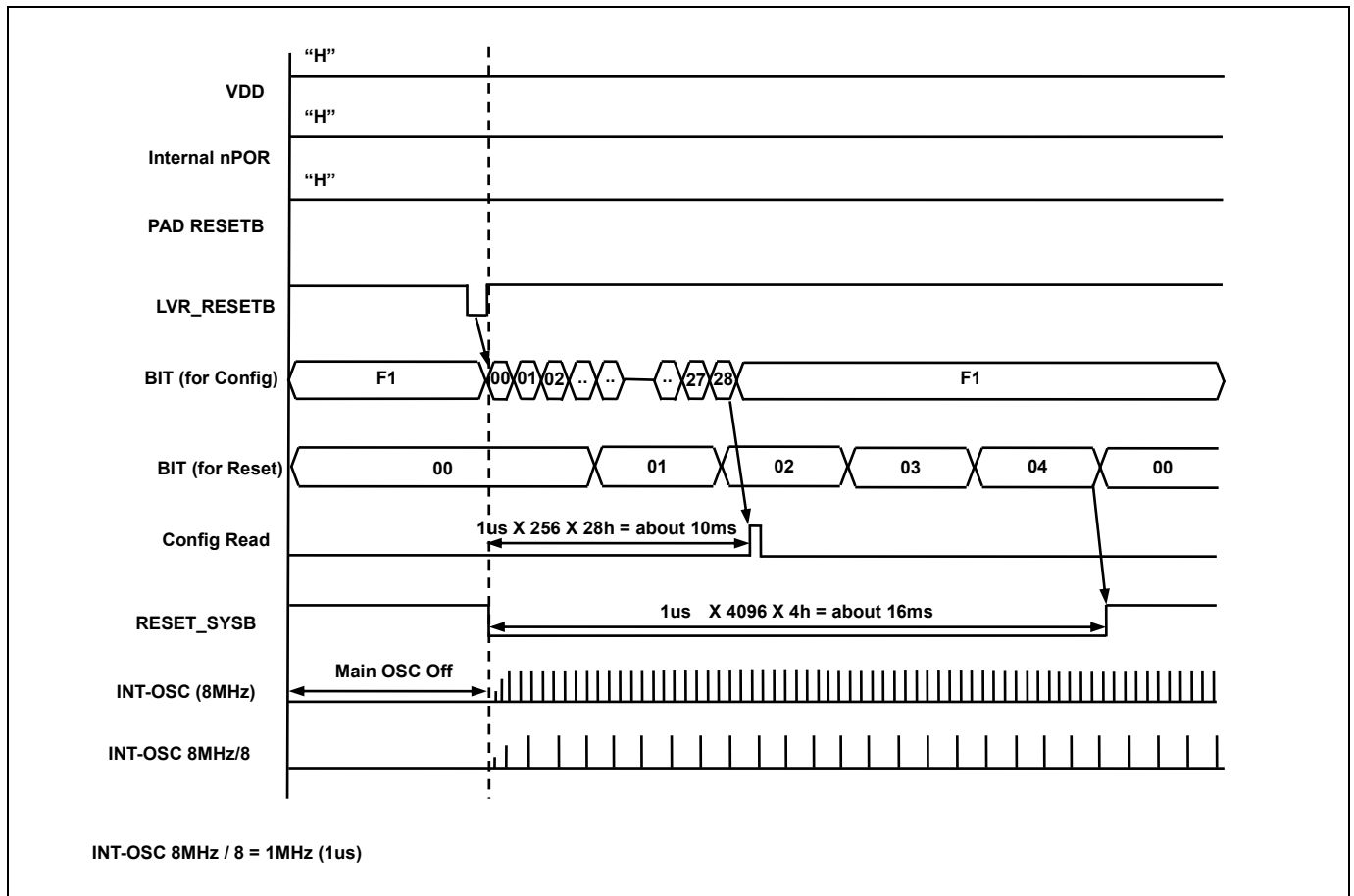


Figure 13.11 BOD RESET时配置时序

13.8 LVI 方框图

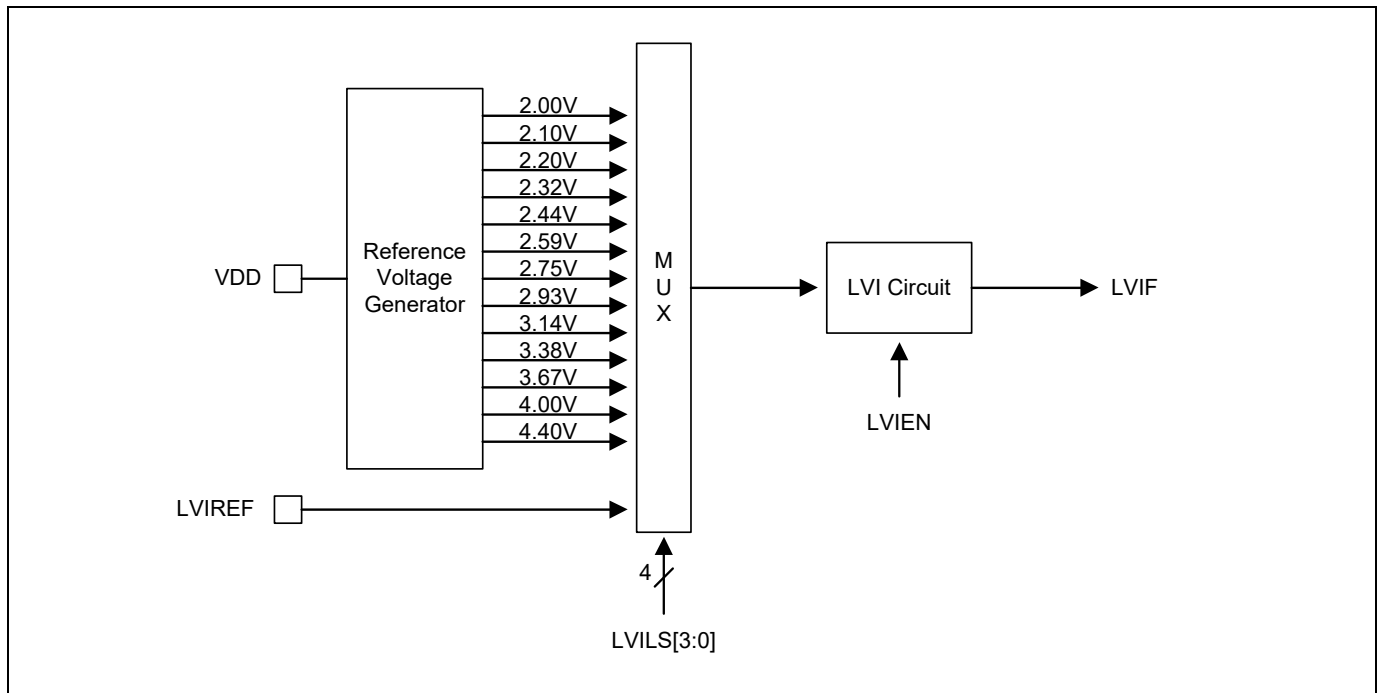


Figure 13.12 LVI 框图

13.9 寄存器图

Name	Address	Direction	Default	Description
RSTFR	E8H	R/W	80H	复位标志位寄存器
LVRCR	D8H	R/W	00H	低电压复位控制寄存器
LVICR	86H	R/W	00H	低电压指示控制寄存器

Table 13.3 复位操作寄存器

13.10 复位操作寄存器说明

复位控制寄存器由(RSTFR), (LVRCR)和 (LVICR)组成.

13.11 复位操作寄存器

RSTFR (复位标志位寄存器): E8H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	–	LVRF	–	–	–
RW	RW	RW	–	RW	–	–	–

Initial value: 80H

- PORF** 上电复位标志位. 写入 '0' 清除该位.
 0 没有检测到
 1 检测到
- EXTRF** 外部复位 (RESETB) 标志位. 写入 '0' 或通过上电复位对该位进行复位.
 0 没有检测到
 1 检测到
- WDTRF** 看门狗复位标志位, 该位通过写入 '0' 或上电复位清除.
 0 没有检测到
 1 检测到
- LVRF** 低电压复位标志位. 该位通过写入 '0' 或上电复位清除.
 0 没有检测到
 1 检测到

NOTE)

1. 上电复位发生时, PORF 只唯一是“1”的, 其他标志位(WDTRF 和 OCDRF) 都清零..
2. 上电复位发生时. EXTRF 是未知的, 此时, EXTRF 可以被设置“1” 当外部复位时(RESETB).
3. 上电复位发生时, LVRF 是未知的, 此时, LVRF 可以被设置“1” 当 LVR 复位发生时.
4. 除了 POR 之外的复位发生时, 相应的标志位为“1”, 其他标志位保持先前值.

LVRCR (低电压复位控制寄存器): D8H

7	6	5	4	3	2	1	0
LVRST	–	–	LVRVS3	LVRVS2	LVRVS1	LVRVS0	LVREN
RW	–	–	RW	RW	RW	RW	RW

Initial value: 00H

LVRST 当退出 Stop 时 LVR 使能
 0 退出睡眠模式时不影响
 1 退出睡眠模式时 LVR 使能

NOTE)

1. 当该位是‘1’, LVREN 位被退出睡眠信号清零 (LVR 使能)
2. 当该位是 ‘0’, LVREN 位对唤醒睡眠信号没反应.

LVRVS[3:0] LVR 电压选择

LVRVS3	LVRVS2	LVRVS1	LVRVS0	Description
0	0	0	0	1.85V
0	0	0	1	2.20V
0	0	1	0	2.20V
0	0	1	1	2.20V
0	1	0	0	2.32V
0	1	0	1	2.44V
0	1	1	0	2.59V
0	1	1	1	2.75V
1	0	0	0	2.93V
1	0	0	1	3.14V
1	0	1	0	3.38V
1	0	1	1	3.67V
1	1	0	0	4.00V
1	1	0	1	4.40V
1	1	1	0	Not available
1	1	1	1	Not available

LVREN LVR 操作
 0 LVR 使能
 1 LVR 禁止

NOTE)

1. LVRST, LVRVS[3:0] 位被上电复位清零但是其他复位时仍保留.
2. LVRVS[3:0] 位应该设置为 ‘0000b’ 当 LVREN =“1”.
3. LVRVS[3:0] 位应该设置为 ‘0x1’-‘0xd’ 当 LVREN =“0b”.

LVICR (低电压指示控制寄存器): 86H

7	6	5	4	3	2	1	0
-	-	LVIF	LVLEN	LVILS3	LVILS2	LVILS1	LVILS0
-	-	RW	RW	RW	RW	RW	RW

Initial value: 00H

LVIF 低电压指示标志位
 0 没有检测到
 1 检测到

LVLEN LVI 使能/禁止
 0 禁止
 1 使能

LVILS[3:0] LVI 电平选择

LVILS3	LVILS2	LVILS1	LVILS0	Description
0	0	0	0	2.20V
0	0	0	1	2.20V
0	0	1	0	2.20V
0	0	1	1	2.32V
0	1	0	0	2.44V
0	1	0	1	2.59V
0	1	1	0	2.75V
0	1	1	1	2.93V
1	0	0	0	3.14V
1	0	0	1	3.38V
1	0	1	0	3.67V
1	0	1	1	4.00V
1	1	0	0	4.40V
Other Values				Not available

14 片内调试系统(MC96F6432 ONLY)

14.1 概述

14.1.1 说明

MC96F6432A 不支持片内调试(OCD). MC96F6432A 没有片内仿真系统. 我们推荐使用 MC96F6432 进行仿真和调试. MC96F6432片内调试系统可作为烧录程序使用和片内调试. 详细内容请参考以下内容.

图 14.1 所示为OCD接口和片内调试系统的方框图.

14.1.2 特性

- 两根外部接口线: 一根是时钟输入, 一根是串行数据总线
- 调试访问:
 - 所有的内部外围模块
 - 内部数据 RAM
 - 程序计数器
 - Flash 和 Data EEPROM 存储器
- 片内调试支持中断,包括
 - 中断指令
 - 单步中断
 - 程序存储器信号地址断点
 - 通过两根总线对Flash, EEPROM, Fuses, 和锁存位进行编程
 - 支持Dr.Choice® 片内调试
- 工作频率
 - 支持目标 MCU 的最大频率

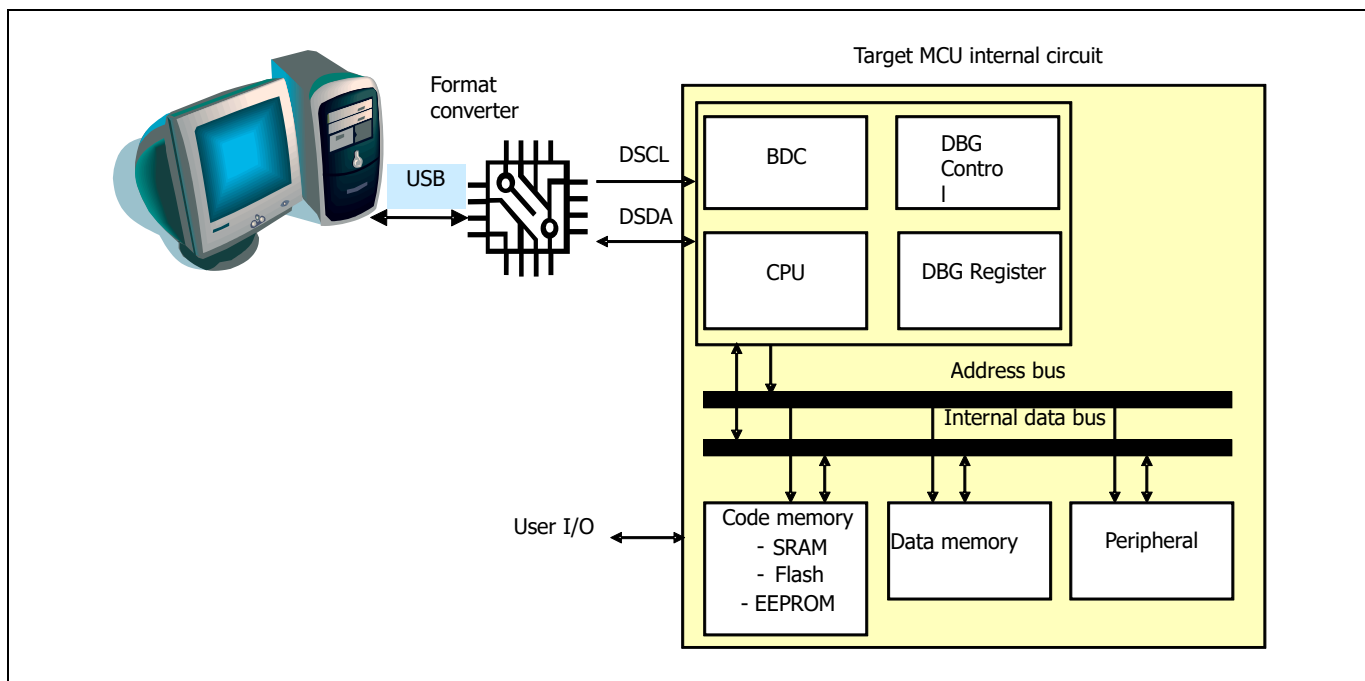


Figure 14.1 OCD框图

14.2 双线外围接口

14.2.1 基本传输

- 10-bit 传输包用于双线接口.
- 1个传输包包括 8位数据, 1位校验和1位应答.
- 8位数据传送时校验位是1
- 8位数据传送和校验位无误时接收器产生应答 '0'.
- 发送器没有确认(应答位是'1'), 错误处理在发生器中执行.
- 产生应答错误时, 主机产生停止命令.
- 后台调试器由一系列数据包组成.
- Start 和stop 命令通知后台调试器开始或停止.

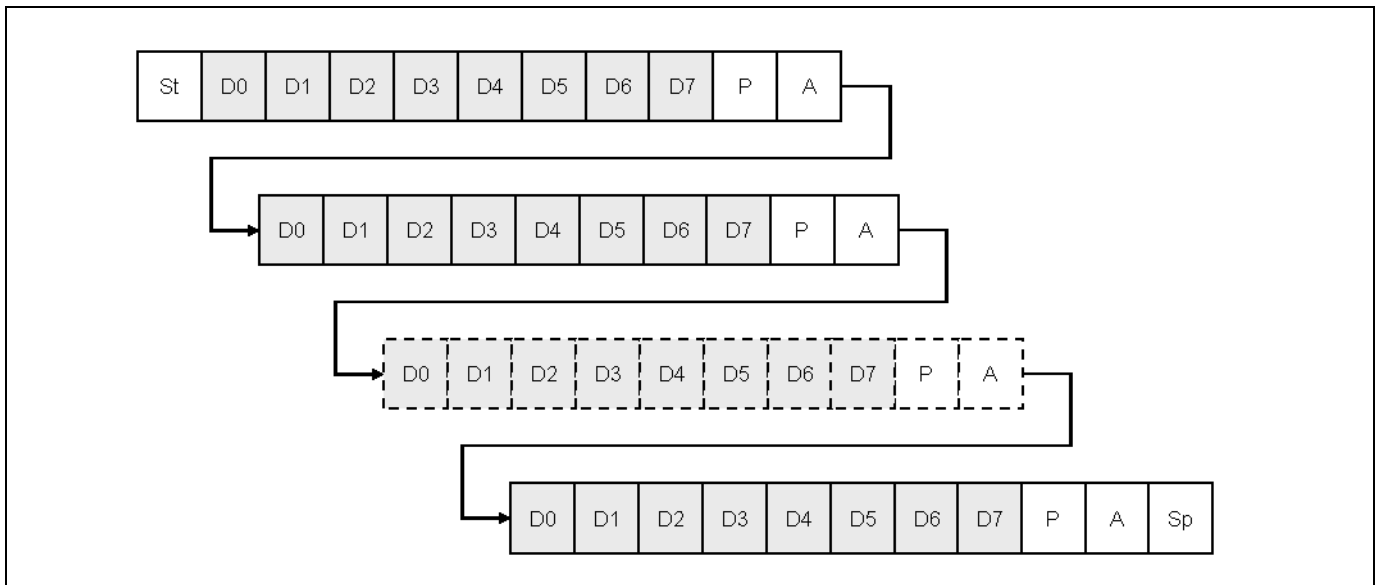


Figure 14.2 10-bit 传输数据包

14.2.2 数据包传输时序

14.2.2.1 数据发送

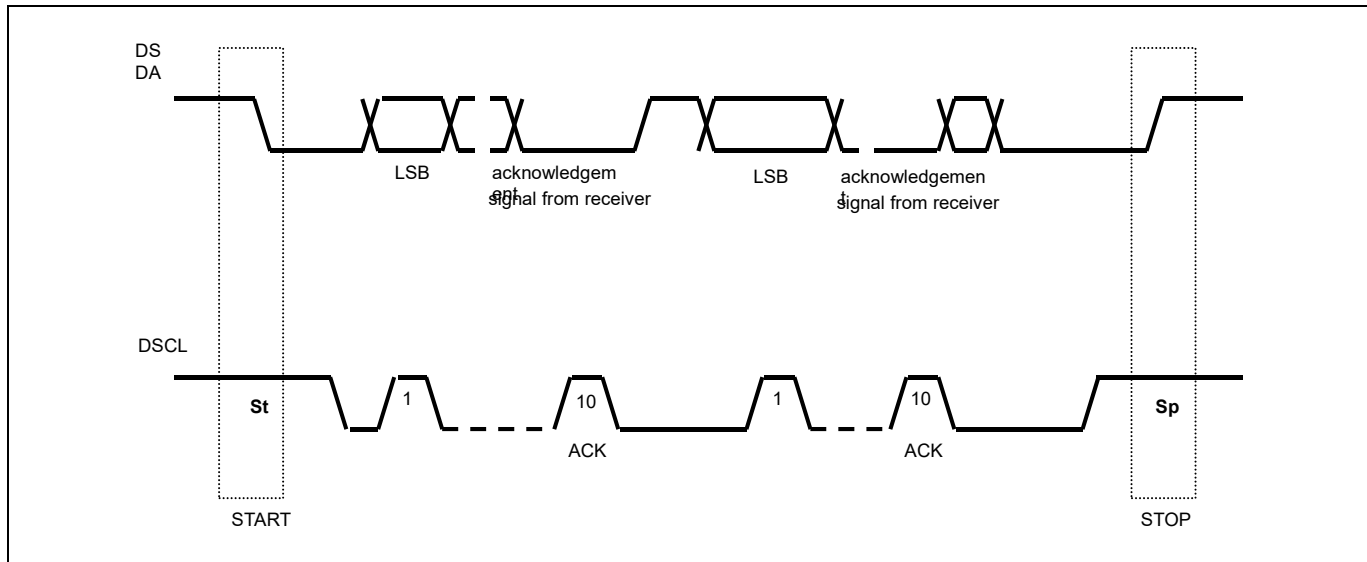


Figure 14.3 两条总线上的数据传输

14.2.2.2 位传输

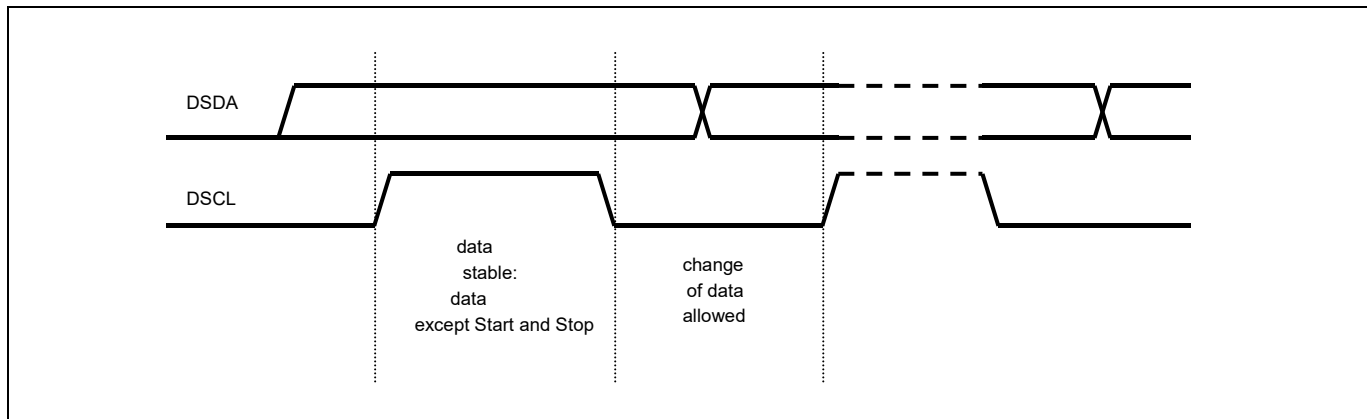


Figure 14.4 串行总线位传输

14.2.2.3 Start 和 Stop 条件

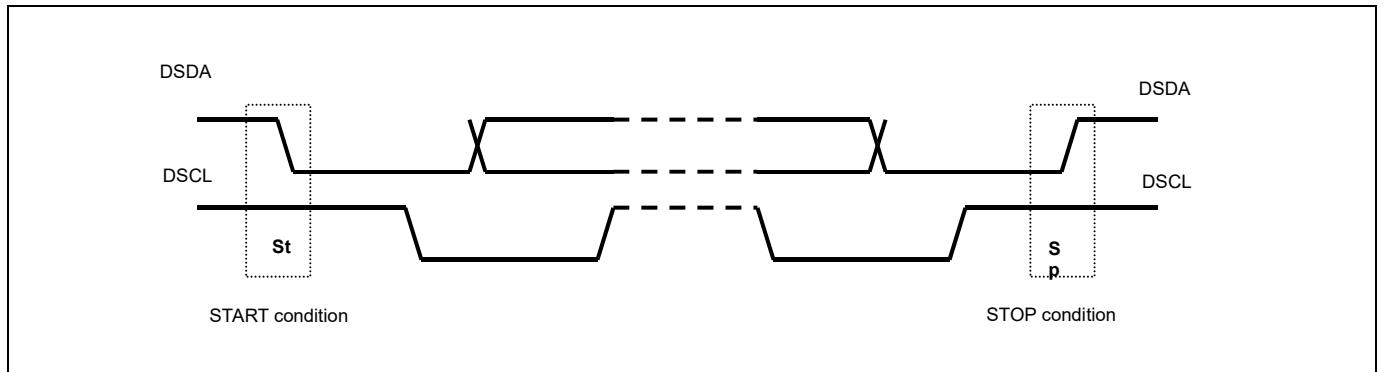


Figure 14.5 Start 和 Stop 条件

14.2.2.4 应答位

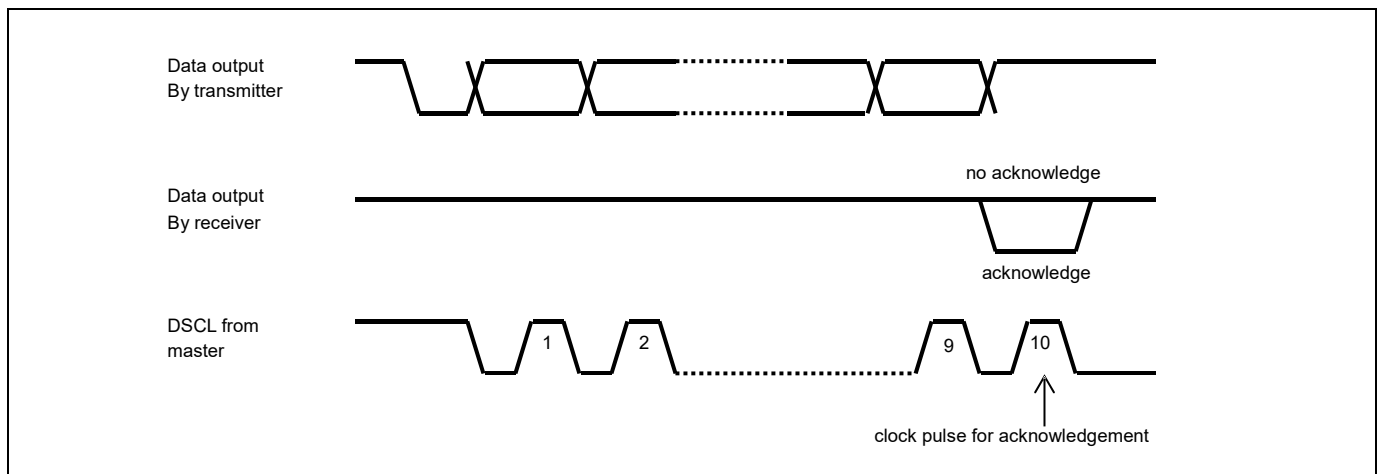


Figure 14.6 串行总线上的应答位

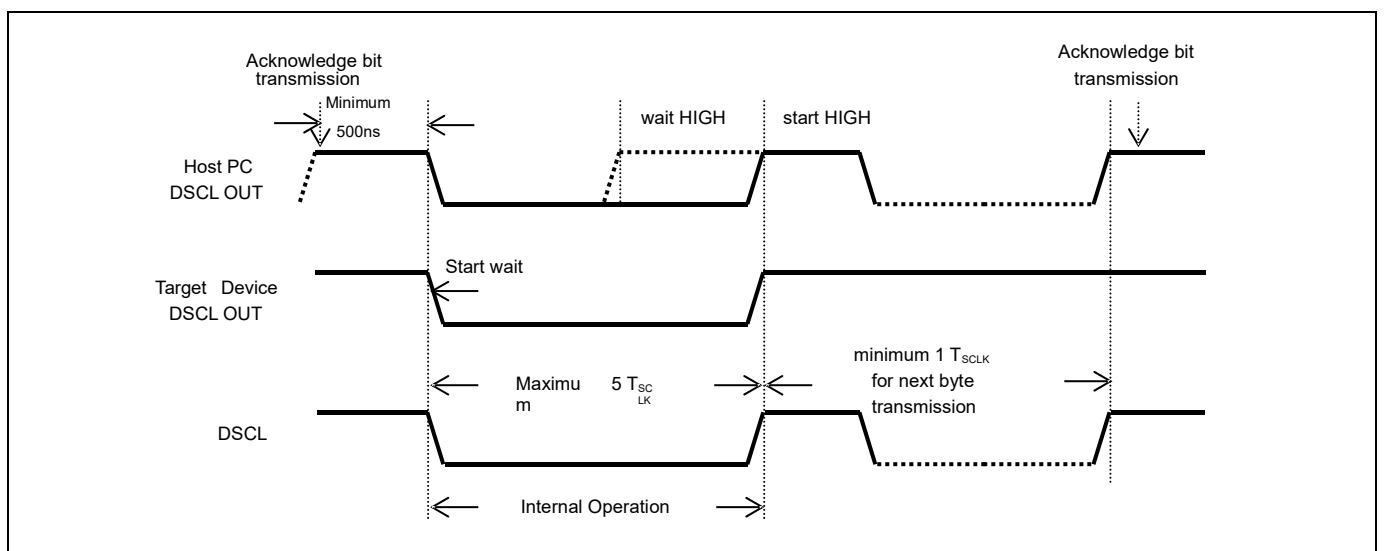


Figure 14.7 等待过程中的时钟同步

14.2.3 传输线路

双脚接口使用漏极开路线路(线与双向 I/O).

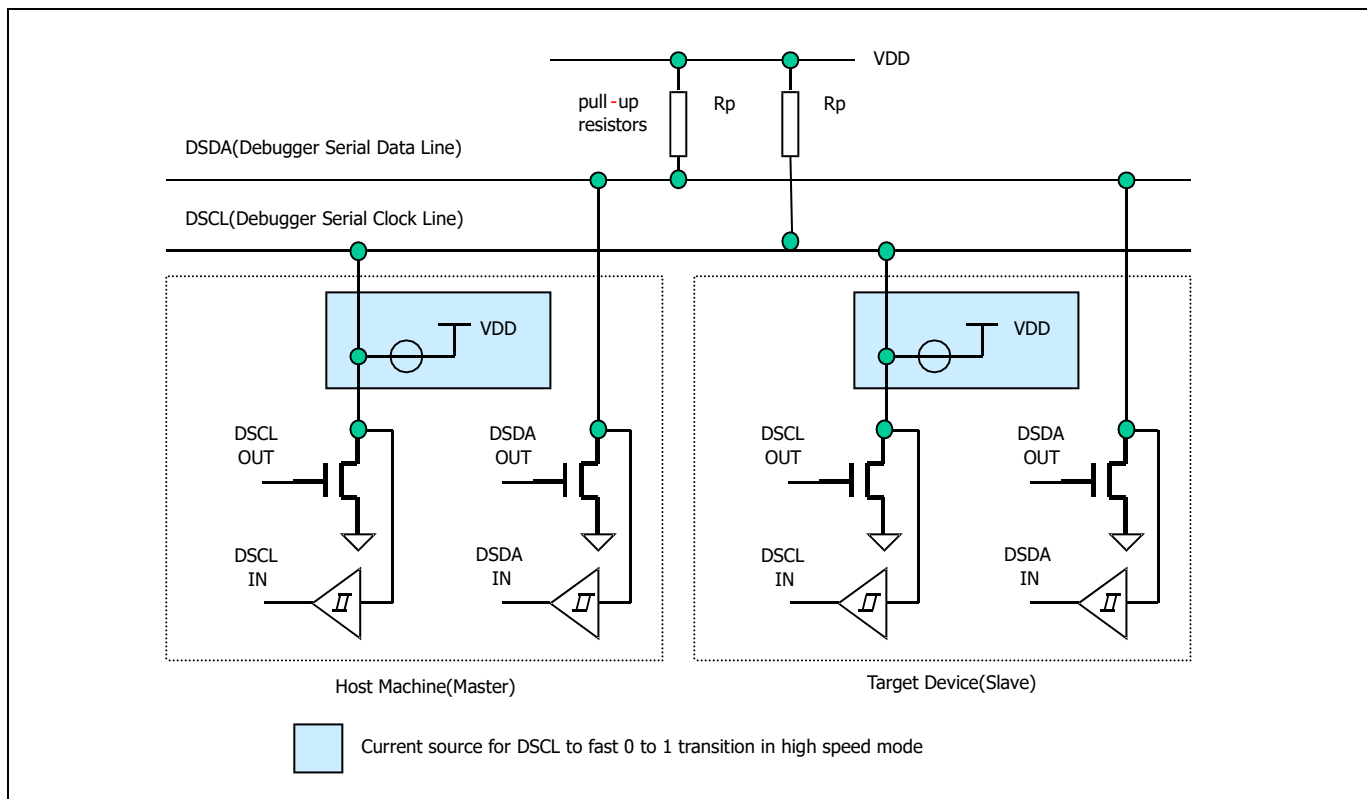


Figure 14.8 传输线路

15 Flash 存储器

15.1 概述

15.1.1 说明

MC96F6432A 集成了flash 存储器，可写入、擦除，和覆盖写入。flash 存储器可以通过‘MOVC’ 指令来读取并且支持串行 ISP 模式或用户编程模式

- Flash Size : 32Kbytes
- 单电源编程和擦写
- 命令接口快速编程和擦写操作
- 标准温度和电压下，高达 100,000(Sector 0~507)/100,000(Sector 508~511) 编程/擦除

NOTE)

1. Flash 在擦写、写入之前，USI0CR2/USI1CR2 寄存器的 RXE0/1 位应该被禁用。

15.1.2 Flash存储器 ROM结构

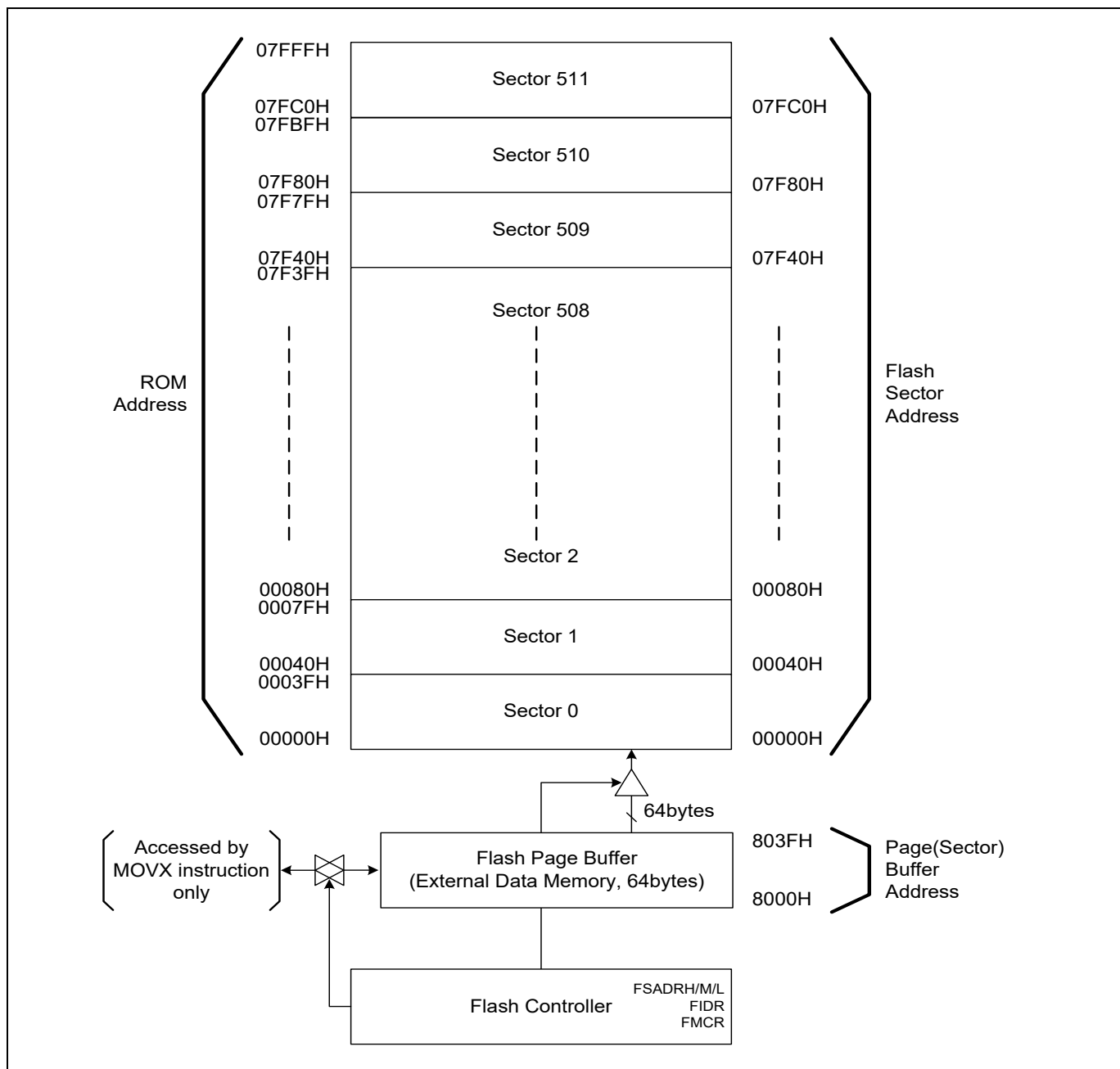


Figure 15.1 Flash存储器 ROM结构

15.1.3 寄存器图

Name	Address	Direction	Default	Description
FSADRH	FAH	R/W	00H	Flash 页地址寄存器 High
FSADRM	FBH	R/W	00H	Flash 页地址寄存器 Middle
FSADRL	FCH	R/W	00H	Flash 页地址寄存器 Low
FIDR	FDH	R/W	00H	Flash 识别寄存器
FMCR	FEH	R/W	00H	Flash 模式控制寄存器

Table 15.1 Flash 存储器寄存器

15.1.4 Flash 存储器控制和状态寄存器说明

Flash 控制寄存器由FSADRH, FSADRM, FSADRL, FIDR, FMCR组成. 映射到SFR 区域, 只有在编程模式时可以进入.

15.1.5 Flash 寄存器说明

FSADRH (Flash 页地址寄存器 High): FAH

7	6	5	4	3	2	1	0
-	-	-	-	FSADRH3	FSADRH2	FSADRH1	FSADRH0
-	-	-	-	RW	RW	RW	RW

Initial value: 00H

FSADRH[3:0] Flash 页地址 High

FSADRM (Flash 页地址寄存器 Middle): FBH

7	6	5	4	3	2	1	0
FSADRM7	FSADRM6	FSADRM5	FSADRM4	FSADRM3	FSADRM2	FSADRM1	FSADRM0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

FSADRM[7:0] Flash 页地址 Middle

FSADRL (Flash 页地址寄存器 Low): FCH

7	6	5	4	3	2	1	0
FSADRL7	FSADRL6	FSADRL5	FSADRL4	FSADRL3	FSADRL2	FSADRL1	FSADRL0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

FSADRL[7:0] Flash 页地址 Low

FIDR (Flash 识别寄存器): FDH

7	6	5	4	3	2	1	0
FIDR7	FIDR6	FIDR5	FIDR4	FIDR3	FIDR2	FIDR1	FIDR0
RW	RW	RW	RW	RW	RW	RW	RW

Initial value: 00H

FIDR[7:0] Flash 识别

Others 其他值不识别
 10100101 flash 模式识别数值
 (一次操作后, 立即自动清零)

FMCR (Flash Mode Control Register): FEH

7	6	5	4	3	2	1	0
FMBUSY	–	–	–	–	FMCR2	FMCR1	FMCR0
R	–	–	–	–	RW	RW	RW

Initial value: 00H

FMBUSY Flash 模式忙碌位. 该位只有在调试时使用.

0 写入“1”没有作用

1 忙碌

FMCR[2:0] Flash 模式控制位. 操作 flash 模式时, CPU保持工作, 无论 IE.7 (EA) 位如何, 全局中断都处于禁止状态.

FMCR2	FMCR1	FMCR0	Description
-------	-------	-------	-------------

0	0	1	选择 flash 页缓存器复位模式 并开始启动而不管FIDR的值 (清除所有64个字节)
---	---	---	--

0	1	0	选择 flash 页擦除模式并开始启动 当 FIDR="10100101b"
---	---	---	---

0	1	1	选择 flash 页写入模式并开始启动 当 FIDR="10100101b"
---	---	---	---

1	0	0	选择 flash 页锁定模式并开始启动 当 FIDR="10100101b"
---	---	---	---

其他值: 无操作

(一次操作后, 立即自动清零)

15.1.6 串行ISP模式

串行ISP使用两根线作为接口. 具体参考14 章

15.1.7 保护区域 (用户编程模式)

MC96F6432A 可对它本身的flash 存储器进行编程(保护区域). 保护区域是不可以被擦除的也不能烧录的. 保护区域只有在PAEN 位清零时可供使用。就是说，如果需要使能第2个配置项的保护区域的话，如果该区域没有使能(PAEN = '1'), 该区域被用于正常的程序存储器.

保护区域的大小是可以通过配置的第二个选项改变.

Protection Area Size Select			Size of Protection Area	Address of Protection Area
PASS2	PASS1	PASS0		
0	0	0	0.7Kbytes	0100H – 0FFFH
0	0	1	1.7Kbytes	0100H – 07FFH
0	1	0	2.7Kbytes	0100H – 03FFH
0	1	1	3.7Kbytes	0100H – 01FFH
1	0	0	29.7Kbytes	0100H – 77FFH
1	0	1	30.7Kbytes	0100H – 7BFFH
1	1	0	31.2Kbytes	0100H – 7DFFH
1	1	1	31.5Kbytes	0100H – 7EFFH

Table 15.2 Protection Area size

NOTE)

1. 参考 16 章配置选项控制.

15.1.8 擦除模式

页擦除模式时编程步骤

1. 清除页缓存器(FMCR=0x01)
2. 页缓存器写入 '0'
3. 设置页擦除地址寄存器(FSADRH/FSADRM/FSADRL).
4. 设置flash 识别寄存器(FIDR).
5. 检查用户ID为了避免无效的擦除、写入(Notes) .
6. 设置flash 模式控制寄存器(FMCR).
7. Erase verify

Notes) 1. 请参考章节“保护无效的擦除、写入”.

2. 对flash擦除和写入时，应该禁用USI0CR2/USI1CR2寄存器的RXE0/1位.

Program Tip – sector erase

```

ANL    EO,#0xF8                ;Set DPTR0
MOV    FMCR,#0x01              ;page buffer clear
NOP    ;Dummy instruction, This instruction must be needed.
NOP    ;Dummy instruction, This instruction must be needed.
NOP    ;Dummy instruction, This instruction must be needed.

MOV    A,#0
MOV    R0,#SectorSize          ;Sector size of Device
MOV    DPH,#0x80                ;Page Buffer Address is 8000H
MOV    DPL,#0

Pgbuf_clr:
MOVX   @DPTR,A
INC    DPTR
DJNZ   R0,Pgbuf_clr            ;Write '0' to all page buffer

MOV    FSADRH,#SAH              ;Sector Address High Byte.
MOV    FSADRM,#SAM              ;Sector Address Middle Byte
MOV    FSADRL,#SAL              ;Sector Address Low Byte
MOV    FIDR,#0xA5               ;Identification value

MOV    A,#ID_DATA_1             ;Check the UserID(written by user)
CJNE   A,UserID1,No_WriteErase;This routine for UserID must be needed.
MOV    A,#ID_DATA_2
CJNE   A,UserID2,No_WriteErase

MOV    FMCR,#0x02              ;Start flash erase mode
NOP    ;Dummy instruction, This instruction must be needed.
NOP    ;Dummy instruction, This instruction must be needed.
NOP    ;Dummy instruction, This instruction must be needed.

LJMP   Erase_verify
---
No_WriteErase:
MOV    FIDR,#00H
MOV    UserID1,#00H
MOV    UserID2,#00H
---
Erase_verify:
---
Verify_error:
---
```

15.1.9 写模式

用户编程模式下的页写模式

1. 清除页缓存器(FMCR=0x01)
2. 写数据到页缓存器
3. 设置地址寄存器(FSADRH/FSADRM/FSADRL).
4. 设置flash 识别寄存器(FIDR).
5. 检查用户ID为了避免无效的擦除、写入(Notes).
6. 设置 flash 模式控制寄存器(FMCR).
7. Erase verify

Notes) 1. 请参考章节“保护无效的擦除、写入”。

2. 对flash擦除和写入时，应该禁用USI0CR2/USI1CR2寄存器的RXE0/1位。

Program Tip – sector write

```

        ANL    EO,#0xF8                ;Set DPTR0
        MOV    FMCR,#0x01            ;page buffer clear
        NOP                                ;Dummy instruction, This instruction must be needed.
        NOP                                ;Dummy instruction, This instruction must be needed.
        NOP                                ;Dummy instruction, This instruction must be needed.

        MOV    A,#0
        MOV    R0,#SectorSize        ;Sector size of Device
        MOV    DPH,#0x80              ;Page Buffer Address is 8000H
        MOV    DPL,#0

Pgbuf_WR:  MOVX   @DPTR,A
          INC    A
          INC    DPTR
          DJNZ   R0,Pgbuf_WR          ;Write data to all page buffer

          MOV    FSADRH,#SAH          ;Sector Address High Byte.
          MOV    FSADRM,#SAM          ;Sector Address Middle Byte
          MOV    FSADRL,#SAL          ;Sector Address Low Byte
          MOV    FIDR,#0xA5           ;Identification value

          MOV    A,#ID_DATA_1         ;Check the UserID(written by user)
          CJNE   A,UserID1,No_WriteErase;This routine for UserID must be needed.
          MOV    A,#ID_DATA_2
          CJNE   A,UserID2,No_WriteErase

          MOV    FMCR,#0x03           ;Start flash write mode
          NOP                                ;Dummy instruction, This instruction must be needed.
          NOP                                ;Dummy instruction, This instruction must be needed.
          NOP                                ;Dummy instruction, This instruction must be needed.

          LJMP   Write_verify
          ---

No_WriteErase:
          MOV    FIDR,#00H
          MOV    UserID1,#00H
          MOV    UserID2,#00H
          ---

Write_verify:
          ---

Verify_error:
          ---
    
```

字节写编程步骤

1. 清除页缓存器(FMCR=0x01)
2. 写数据到页缓存器
3. 设置地址寄存器(FSADRH/FSADRM/FSADRL).
4. 设置 flash 识别寄存器(FIDR).
5. 检查用户ID为了避免无效的擦除、写入^(Note).
6. 设置 flash 模式控制寄存器(FMCR).
7. Erase verify

- Notes)**
1. 请参考章节“保护无效的擦除、写入”。
 2. 写数据之前需要先写"00H"到数据地址。。
 3. 对flash擦除和写入时，应该禁用USI0CR2/USI1CR2寄存器的RXE0/1位。

Program Tip – byte write

```

ANL    EO,#0xF8                ;Set DPTR0
MOV    FMCR,#0x01             ;page buffer clear
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.

MOV    A,#5
MOV    DPH,#0x80
MOV    DPL,#0
MOVX   @DPTR,A                ;Write data to page buffer

MOV    A,#6
MOV    DPH,#0x80
MOV    DPL,#0x05
MOVX   @DPTR,A                ;Write data to page buffer

MOV    FSADRH,#SAH            ;Sector Address High Byte.
MOV    FSADRM,#SAM            ;Sector Address Middle Byte
MOV    FSADRL,#SAL            ;Sector Address Low Byte
MOV    FIDR,#0xA5             ;Identification value

MOV    A,#ID_DATA_1           ;Check the UserID(written by user)
CJNE   A,UserID1,No_WriteErase;This routine for UserID must be needed.
MOV    A,#ID_DATA_2
CJNE   A,UserID2,No_WriteErase

MOV    FMCR,#0x03             ;Start flash write mode
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.
NOP                                ;Dummy instruction, This instruction must be needed.

LJMP   Write_verify
---
No_WriteErase:
MOV    FIDR,#00H
MOV    UserID1,#00H
MOV    UserID2,#00H
---
Write_verify:
---
Verify_error:
---
```

15.1.10 擦除\写入的保护

编写对 flash 进行擦除\写入操作的代码时要注意.

必须有防错操作, 防止程序错误跳转到 flash 擦除\写入的代码, 以及因断电或干扰而造成的错误的擦除\写入操作.

NOTE) 有关详细信息, 请参考附录“Flash 保护无效的擦除、写入”.

1. 在FLASH擦除\写入代码中要检查用户ID.

```
ErWt_rtn:
---
MOV    FIDR,#10100101B      ;ID Code
MOV    A,#ID_DATA_1        ;Ex) ID_DATA_1: 93H, ID_DATA_2: 85H, ID_DATA_3: 5AH
CJNE   A,UserID1,No_WriteErase
MOV    A,#ID_DATA_2
CJNE   A,UserID2,No_WriteErase
MOV    A,#ID_DATA_3
CJNE   A,UserID3,No_WriteErase
MOV    FMCR,#0x??          ;0x03 if write, 0x02 if erase
---
---
RET

No_WriteErase:
MOV    FIDR,#00H
MOV    UserID1,#00H
MOV    UserID2,#00H
MOV    UserID3,#00H
MOV    Flash_flag,#00H
RET
```

如果像上面这样编写代码, 就可以防止错误的FLASH擦除\写入操作.

NOTE) 在闪存擦除和写入时, 应该禁止 USInCR2 寄存器的 RXE 位.

2. 写入 UserID1/2/3 是很重要的. 如果 UserID1/2/3 是在“MOV FIDR,#10100101B”这行指令上面写入的, 将不能防止错误的FLASH擦除\写入操作, 所以最好在返回指令之后写入 UserID1/2/3.

```
Decide_ErWt:
---
MOV    Flash_flag1,#38H    ;Random value for example, in case of erase/write needs
MOV    FSADRL,#20H        ;Here 20H is example,
MOV    Flash_flag2,#75H
RET
```


3. Flash 扇区地址寄存器 (FSADRH/FSADRM/FSADRL) 里的地址值应该始终保持在用户数据区地址范围内, 例如, FSADRH/FSADRM 的值始终为 0x00/0x7f (如果用户数据区地址为 0x7f00 到 0x7fff).

4. 概述 main

```

---
CALL    Work1
CALL    Decide_ErWt
CALL    Work2
CALL    ID_write
CALL    Work3
CALL    Flash_erase
CALL    Flash_write
---
---
---
ID_wire:
MOV     A,#38H
CJNE   A,Flash_flag1,No_write_ID
MOV     A,#75H
CJNE   A,Flash_flag2,No_write_ID
MOV    UserID1,#ID_DATA_1      ;Write User ID1
MOV     A,#38H
CJNE   A,Flash_flag1,No_write_ID
MOV     A,#75H
CJNE   A,Flash_flag2,No_write_ID
MOV    UserID2,#ID_DATA_2      ;Write User ID2
MOV     A,#38H
CJNE   A,Flash_flag1,No_write_ID
MOV     A,#75H
CJNE   A,Flash_flag2,No_write_ID
MOV    UserID3,#ID_DATA_3      ;Write User ID3
RET

No_write_ID:
MOV     UserID1,#00H
MOV     UserID2,#00H
MOV     UserID3,#00H
RET

```

15.1.10.1 无效擦除/写入的保护流程

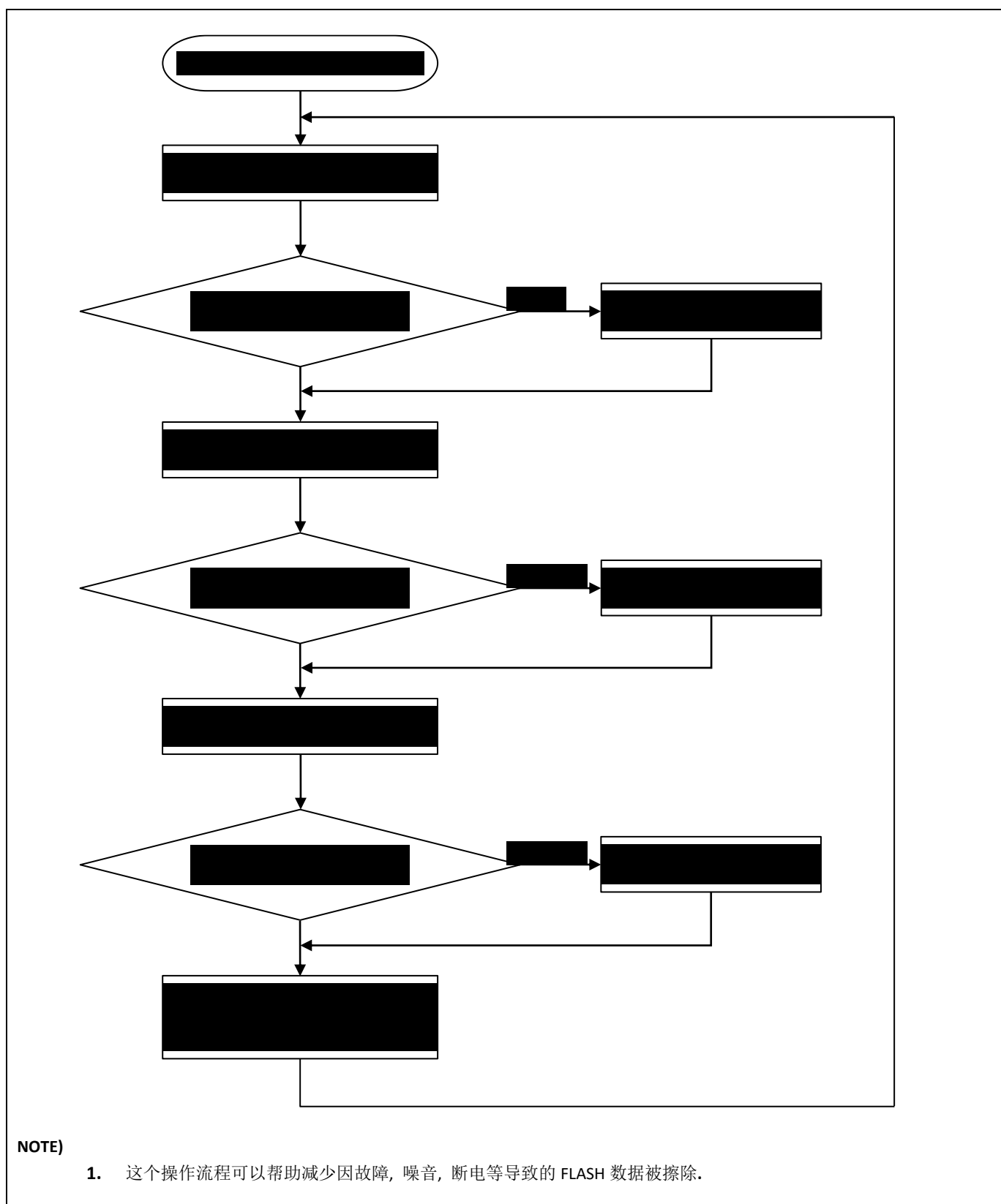


Figure 15.2 无效擦除/写入的保护流程

15.1.11 读模式

用户程序模式下的读模式编程步骤

1. 通过MOVC指令间接寻址从存储器加载接收数据.

Program Tip - reading

```

MOV     A, #0
MOV     DPH, #0x7F
MOV     DPL, #0x40           ;flash memory address

MOVC    A, @A+DPTR          ;read data from flash memory

```

15.1.12 写保护模式

用户程序模式下的写保护模式编程步骤

1. 设置flash 识别寄存器 (FIDR).
2. 检查用户ID为了避免无效的擦除、写入(Note).
3. 设置flash 模式控制寄存器 (FMCR).

NOTE) 有关详细信息，请参考附录“Flash保护无效的擦除、写入”

Program Tip - Code Write Protection

```

MOV     FIDR, #0xA5           ;Identification value

MOV     A, #ID_DATA_1         ;Check the UserID(written by user)
CJNE   A, UserID1, No_WriteErase;This routine for UserID must be needed.
MOV     A, #ID_DATA_2
CJNE   A, UserID2, No_WriteErase

MOV     FMCR, #0x04           ;Start flash Code Write Protection mode
NOP     ;Dummy instruction, This instruction must be
needed.

NOP     ;Dummy instruction, This instruction must be
needed.

NOP     ;Dummy instruction, This instruction must be
needed.

No_WriteErase:
MOV     FIDR, #00H
MOV     UserID1, #00H
MOV     UserID2, #00H
---
```

16 配置选项

16.1 配置选项控制

配置项的数据应该由烧录器写入到配置选项区域 (003EH – 003FH).

CONFIGURE OPTION 1 : ROM Address 003FH

7	6	5	4	3	2	1	0
R_P	HL	–	VAPEN	–	–	–	RSTS

Initial value : 00H

R_P	Code Read Protection
0	禁止
1	使能
HL	Code Write Protection
0	禁止
1	使能
VAPEN	Vector Area (00H – FFH) Protection Enable/Disable
0	Disable Protection (Erasable by instruction)
1	Enable Protection (Not erasable by instruction)
RSTS	Select RESETB pin
0	Disable RESETB pin(P55)
1	Enable RESETB pin

CONFIGURE OPTION 2: ROM Address 003EH

7	6	5	4	3	2	1	0
–	–	–	–	PAEN	PASS2	PASS1	PASS0

Initial value : 00H

PAEN	Enable Specific Area Write Protection		
0	禁止保护 (可通过指令消除)		
1	使能保护 (不可通过指令消除)		
PASS [2:0]	Select Specific Area for Write Protection		
NOTE)			
1. 当 PAEN = '1' 时, 保护被应用.			
PASS2	PASS1	PASS0	Description
0	0	0	0.7Kbytes (Address 0100H – 03FFH)
0	0	1	1.7Kbytes (Address 0100H – 07FFH)
0	1	0	2.7Kbytes (Address 0100H – 0BFFH)
0	1	1	3.7Kbytes (Address 0100H – 0FFFH)
1	0	0	29.7Kbytes (Address 0100H – 77FFH)
1	0	1	30.7Kbytes (Address 0100H – 7BFFH)
1	1	0	31.2Kbytes (Address 0100H – 7DFFH)
1	1	1	31.5Kbytes (Address 0100H – 7EFFH)

17 附录

17.1 指令表

以下列出的指令中‘Bytes’可能是1, 2 或 3 个字节长度.

下表所列的每个指令需要1, 2 或 4 个机器周期. 1 个机器周期包含两个时钟周期.

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

LOGICAL				
Mnemonic	Description	Bytes	Cycles	Hex code
ANL A,Rn	AND register to A	1	1	58-5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56-57
ANL A,#data	AND immediate to A	2	1	54
ANL dir,A	AND A to direct byte	2	1	52
ANL dir,#data	AND immediate to direct byte	3	2	53
ORL A,Rn	OR register to A	1	1	48-4F
ORL A,dir	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect memory to A	1	1	46-47
ORL A,#data	OR immediate to A	2	1	44
ORL dir,A	OR A to direct byte	2	1	42
ORL dir,#data	OR immediate to direct byte	3	2	43
XRL A,Rn	Exclusive-OR register to A	1	1	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	1	65
XRL A,@Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	2	1	64
XRL dir,A	Exclusive-OR A to direct byte	2	1	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03
RRC A	Rotate A right through carry	1	1	13

DATA TRANSFER				
Mnemonic	Description	Bytes	Cycles	Hex code
MOV A,Rn	Move register to A	1	1	E8-EF
MOV A,dir	Move direct byte to A	2	1	E5
MOV A,@Ri	Move indirect memory to A	1	1	E6-E7
MOV A,#data	Move immediate to A	2	1	74
MOV Rn,A	Move A to register	1	1	F8-FF
MOV Rn,dir	Move direct byte to register	2	2	A8-AF
MOV Rn,#data	Move immediate to register	2	1	78-7F
MOV dir,A	Move A to direct byte	2	1	F5
MOV dir,Rn	Move register to direct byte	2	2	88-8F
MOV dir,dir	Move direct byte to direct byte	3	2	85
MOV dir,@Ri	Move indirect memory to direct byte	2	2	86-87
MOV dir,#data	Move immediate to direct byte	3	2	75
MOV @Ri,A	Move A to indirect memory	1	1	F6-F7
MOV @Ri,dir	Move direct byte to indirect memory	2	2	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	2	1	76-77
MOV DPTR,#data	Move immediate to data pointer	3	2	90
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	2	93
MOVC A,@A+PC	Move code byte relative PC to A	1	2	83
MOVX A,@Ri	Move external data(A8) to A	1	2	E2-E3
MOVX A,@DPTR	Move external data(A16) to A	1	2	E0
MOVX @Ri,A	Move A to external data(A8)	1	2	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	1	2	F0
PUSH dir	Push direct byte onto stack	2	2	C0
POP dir	Pop direct byte from stack	2	2	D0
XCH A,Rn	Exchange A and register	1	1	C8-CF
XCH A,dir	Exchange A and direct byte	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	1	1	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	1	D6-D7

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

BRANCHING				
Mnemonic	Description	Bytes	Cycles	Hex code
ACALL addr 11	Absolute jump to subroutine	2	2	11→F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	2	22
RETI	Return from interrupt	1	2	32
AJMP addr 11	Absolute jump unconditional	2	2	01→E1
LJMP addr 16	Long jump unconditional	3	2	02
SJMP rel	Short jump (relative address)	2	2	80
JC rel	Jump on carry = 1	2	2	40
JNC rel	Jump on carry = 0	2	2	50
JB bit,rel	Jump on direct bit = 1	3	2	20
JNB bit,rel	Jump on direct bit = 0	3	2	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	2	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	2	60
JNZ rel	Jump on accumulator ≠0	2	2	70
CJNE A,dir,rel	Compare A,direct jne relative	3	2	B5
CJNE A,#d,rel	Compare A,immediate jne relative	3	2	B4
CJNE Rn,#d,rel	Compare register, immediate jne relative	3	2	B8-BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	2	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	3	2	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5

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

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

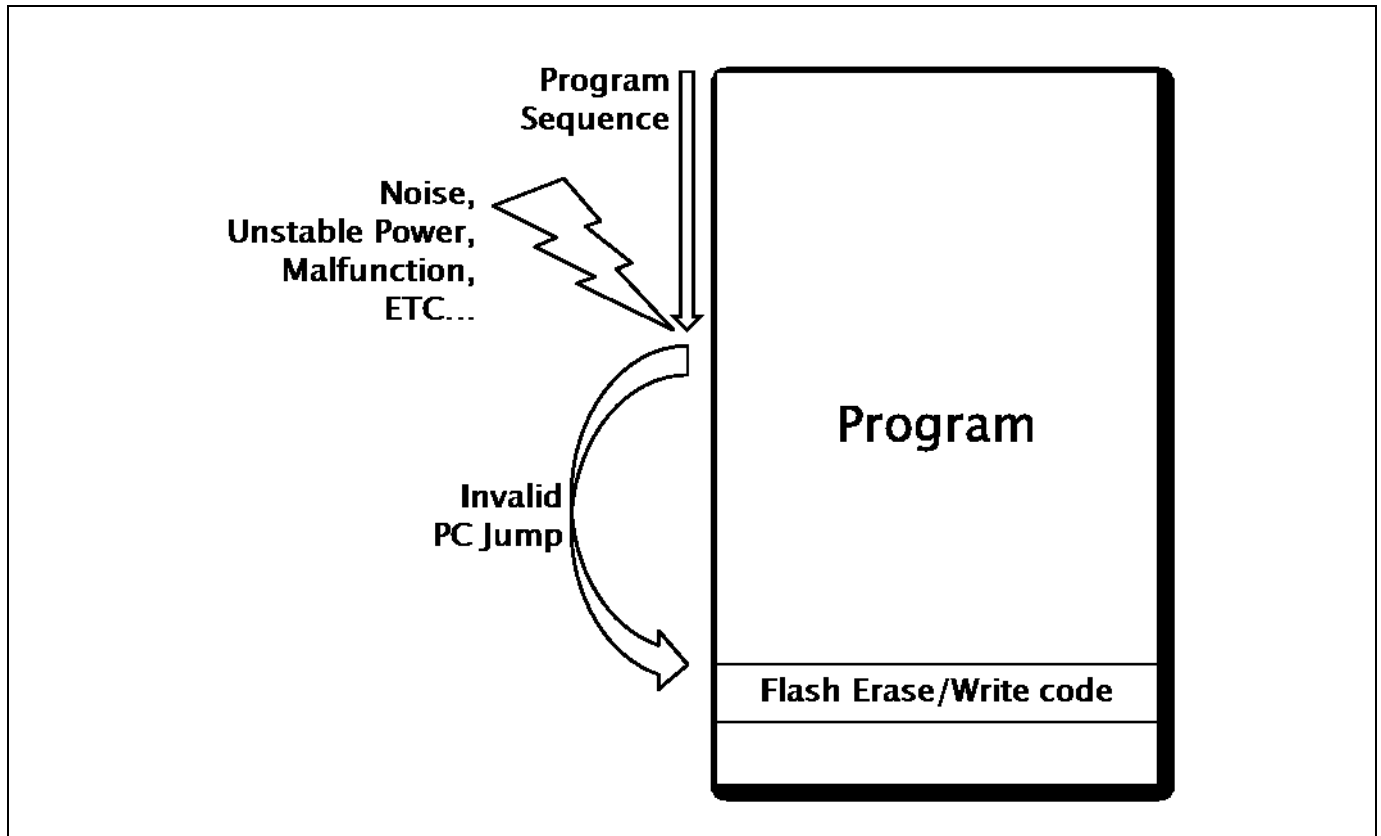
In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps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

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

17.2 Flash Protection for Invalid Erase/Write

➤ Overview

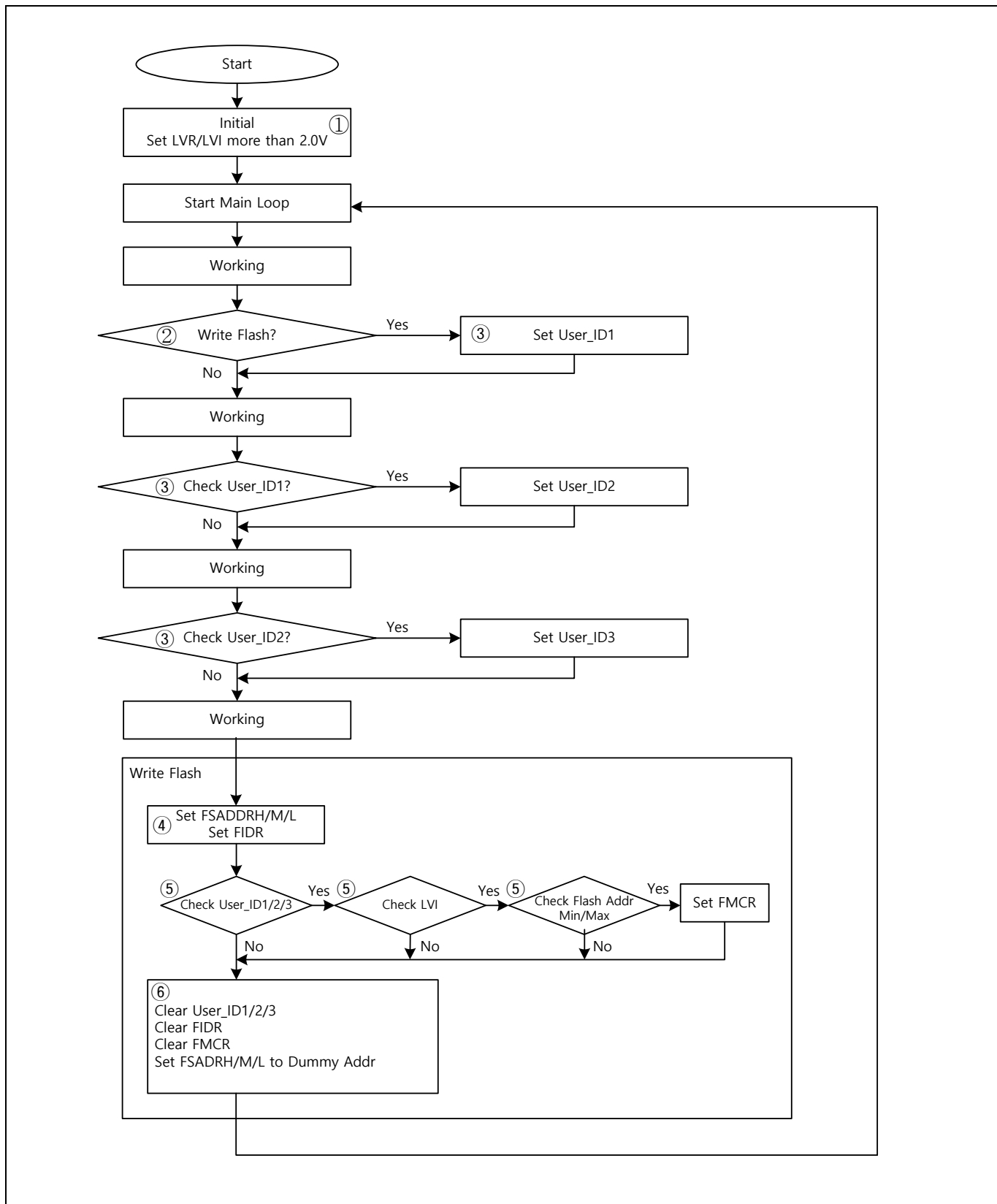
This is example to prevent changing code or data in flash by abnormal operation(noise, unstable power, malfunction, etc...).



➤ How to protect the flash

- Divide into decision and execution to Erase/Write in flash.
 - Check the program sequence from decision to execution in order of precedence about Erase/Write.
 - Setting the flags in program and check the flags in main loop at the end
 - When the Flash Erase/Write is executed, check the flags. If not matched, do not execute.
- Check the range of Flash Sector Address
 - If the flash sector address is outside of specific area, do not execute.
- Use the Dummy Address
 - Set the flash sector address to dummy address in usually run time.
 - Change the flash sector address to real area range shortly before Erase/Write.
 - Even if invalid Erase/Write occurred, it will be Erase/Write in dummy address in flash.
- Use the LVR/LVI
 - Unstable or low powers give an adverse effect on MCU. So use the LVR/LVI

➤ Flowchart



➤ Descript of Flowchart

- ① Initialization
 - Set the LVR/LVI
 - Check the power by LVR/LVI and do not execute under unstable or low power.
 - Initialize User_ID1/2/3
 - Set Flash Sector Address High/Middle/Low to Dummy address
 - Dummy address is set to unused area range in flash.
- ② Decide to Write
 - When the Erase/Write are determined, set flag. Do not directly Erase/Write in flash.
 - Make the user data.
- ③ Check and Set User_ID1/2/3
 - In the middle of source, insert code which can check and set the flags.
 - By setting the User_ID 1/2/3 sequentially and identify the flow of the program.
- ④ Set Flash Sector Address
 - Set address to real area range shortly before Erase/Write in flash.
 - Set to Dummy address after Erase/Write
 - Even if invalid work occurred, it will be Erase/Write in Dummy address in flash.
- ⑤ Check Flags
 - If every flag(User_ID1/2/3, LVI, Flash Address Min/Max) was set, than do Erase/Write.
 - If the Flash Sector Address is outside of Min/Max, do not execute
 - Address Min/Max is set to unused area.
- ⑥ Initialize Flags
 - Initialize User_ID1/2/3
 - Set Flash Sector Address to Dummy Address
- Sample Source
 - Refer to the ABOV homepage.
 - It is created based on the MC97F2664.
 - Each product should be modified according to the Page Buffer Size and Flash Size

➤ Etc

- Protection by Configure option
 - Set flash protection by MCU Write Tool(OCD, PGM+, etc...)
 - Vector Area :
 - 00H~FFH
 - Specific Area :
 - 3.7k Bytes (Address 0100H – 0FFFH)
 - 1.7k Bytes (Address 0100H – 07FFH)
 - 768 Bytes (Address 0100H – 03FFH)
 - 256 Bytes (Address 0100H – 01FFH)
 - The range of protection may be different each product.

目录

修订记录.....	2
1 综述.....	3
1.1 说明.....	3
1.2 性能.....	4
1.3 开发工具.....	5
1.3.1 编译器.....	5
1.3.2 OCD(On-chip debugger) 调试器.....	5
1.3.3 烧写器.....	7
1.4 MTP 烧写.....	9
1.4.1 概述.....	9
1.4.2 在线烧录.....	9
1.4.2.1 电路设计指导.....	9
2 方框图.....	11
3 引脚分配.....	12
4 封装图.....	16
5 管脚说明.....	21
6 端口结构.....	26
6.1 普通 I/O 口.....	26
6.2 外部中断 I/O 口.....	27
7 电气特性.....	28
7.1 额定参数范围.....	28
7.2 推荐操作参数.....	28
7.3 A/D 转换特性.....	29
7.4 上电复位特性.....	29
7.5 低电压复位和低电压检测特性.....	30
7.6 内部高速 RC 振荡器特性.....	30
7.7 内部看门狗定时器 RC 振荡器特性.....	31
7.8 LCD 电压特性.....	31
7.9 DC 特性.....	32
7.10 AC 特性.....	33
7.11 SPI0/1/2 特性.....	34
7.12 UART0/1 特性.....	35
7.13 I2C0/1 Characteristics.....	36
7.14 Stop 模式下保存数据所需电压.....	37
7.15 内部 Flash Rom 特性.....	38
7.16 输入/输出 特性.....	38
7.17 主时钟振荡特性.....	39
7.18 副时钟振荡特性.....	40
7.19 主时钟振荡稳定特性.....	41
7.20 副时钟特性.....	41
7.21 工作电压范围.....	42
7.22 推荐电路和布板.....	43
7.23 推荐开关电源的电路布局.....	44
7.24 典型特性.....	45
8 存储器.....	48
8.1 程序存储器.....	48
8.2 数据存储器.....	50
8.3 外部数据存储器.....	52

8.4	SFR.....	53
8.4.1	SFR 图表摘要	53
8.4.2	SFR	55
8.4.3	SFR Map	60
9	I/O 口.....	62
9.1	I/O 口.....	62
9.2	端口寄存器.....	62
9.2.1	数据寄存器 (Px).....	62
9.2.2	方向寄存器 (PxIO).....	62
9.2.3	上拉电阻选择寄存器 (PxPU).....	62
9.2.4	漏极开路选择寄存器(PxOD).....	62
9.2.5	消抖使能寄存器 (PxDB).....	62
9.2.6	端口功能选择寄存器 (PxFSR).....	63
9.2.7	寄存器图	63
9.3	P0 口	64
9.3.1	P0 端口说明	64
9.3.2	P0寄存器说明	64
9.4	P1 口	68
9.4.1	P1 端口说明	68
9.4.2	Register description for P1	68
9.5	P2 口	72
9.5.1	P2 端口说明	72
9.5.2	P2 寄存器说明	72
9.6	P3 口	75
9.6.1	P3 端口说明	75
9.6.2	P3 寄存器说明	75
9.7	P4 口	77
9.7.1	P4 端口说明	77
9.7.2	P4 寄存器说明	77
9.8	P5 口	79
9.8.1	P5 端口说明	79
9.8.2	P5 寄存器说明	79
10	中断控制.....	81
10.1	概述.....	81
10.2	外部中断.....	83
10.3	方框图.....	84
10.4	中断向量表.....	85
10.5	中断顺序.....	86
10.6	中断控制位之后的作用时序.....	87
10.7	多中断复用.....	88
10.8	中断使能接受时序.....	89
10.9	中断服务程序地址.....	89
10.10	通用寄存器的设置和复位.....	89
10.11	中断时序.....	90
10.12	中断寄存器概述.....	90
10.12.1	中断使能寄存器 (IE, IE1, IE2, IE3).....	90
10.12.2	中断优先级寄存器 (IP, IP1).....	90
10.12.3	外部中断标志位寄存器 (EIFLAG0, EIFLAG1).....	91
10.12.4	外部中断优先级寄存器 (EIPOL0L, EIPOL0H, EIPOL1).....	91
10.12.5	中断寄存器图	91
10.12.6	中断寄存器说明.....	91
10.12.7	中断寄存器说明.....	92
11	外围硬件.....	98

11.1	时钟发生器	98
11.1.1	概述	98
11.1.2	方框图	99
11.1.3	寄存器图	100
11.1.4	时钟发生器寄存器说明	100
11.1.5	时钟发生器寄存器说明	100
11.2	BIT	103
11.2.1	概述	103
11.2.2	方框图	103
11.2.3	寄存器图	103
11.2.4	BIT 寄存器说明	104
11.2.5	BIT寄存器	104
11.3	WDT	105
11.3.1	概述	105
11.3.2	WDT 中断时序波形图	105
11.3.3	方框图	106
11.3.4	寄存器图	106
11.3.5	WDT寄存器说明	106
11.3.6	WDT寄存器	107
11.4	WT	108
11.4.1	概述	108
11.4.2	方框图	108
11.4.3	寄存器图	109
11.4.4	WT说明	109
11.4.5	WT寄存器	109
11.5	定时器 0	111
11.5.1	概述	111
11.5.2	8-bit 定时器/计数器 模式	112
11.5.3	8-bit PWM 模式	113
11.5.4	8-bit 捕捉模式	115
11.5.5	方框图	117
11.5.6	寄存器图	117
11.5.7	定时器/计数器0 寄存器说明	117
11.5.8	定时器/计数器0 寄存器	118
11.6	定时器 1	120
11.6.1	概述	120
11.6.2	16-bit 定时器/计数器 模式	120
11.6.3	16-bit 捕捉模式	122
11.6.4	16-bit PPG 模式	124
11.6.5	方框图	126
11.6.6	寄存器图	126
11.6.7	定时器/计数器 1 寄存器说明	127
11.6.8	定时器/计数器 1 寄存器	127
11.7	定时器 2	130
11.7.1	概述	130
11.7.2	16-bit 定时器/计数器 模式	131
11.7.3	16-bit 捕捉模式	133
11.7.4	16-bit PPG 模式	135
11.7.5	方框图	137
11.7.6	寄存器图	137
11.7.7	定时器/计数器 2 寄存器说明	138
11.7.8	定时器/计数器 2 寄存器	138
11.8	定时器 3, 4	141
11.8.1	概述	141
11.8.2	8-bit 定时器/计数器 3, 4 模式	142

11.8.3	16-bit 定时器/计数器 3 模式	143
11.8.4	8-bit 定时器 3, 4 捕捉模式	144
11.8.5	定时器3 16-bit 捕捉模式	146
11.8.6	定时器4 10-bit PWM 模式	147
11.8.7	方框图	157
11.8.8	寄存器图	159
11.8.9	定时器/计数器 3 寄存器说明	160
11.8.10	定时器/计数器 3 寄存器	160
11.8.11	定时器/计数器 4 计数器说明	162
11.8.12	定时器/计数器 4 寄存器	162
11.9	蜂鸣器驱动	170
11.9.1	概述	170
11.9.2	方框图	170
11.9.3	寄存器图	171
11.9.4	蜂鸣器驱动寄存器说明	171
11.9.5	蜂鸣器驱动寄存器	171
11.10	SPI 2	172
11.10.1	概述	172
11.10.2	方框图	172
11.10.3	数据发送/接收	173
11.10.4	SS2 引脚功能	173
11.10.5	SPI 2 时序图	174
11.10.6	寄存器图	175
11.10.7	SPI 2 寄存器说明	175
11.10.8	SPI 2 寄存器	175
11.11	12-bit A/D 转换器	178
11.11.1	概述	178
11.11.2	转换时序	178
11.11.3	方框图	179
11.11.4	ADC 操作	180
11.11.5	寄存器图	181
11.11.6	ADC 寄存器说明	181
11.11.7	ADC寄存器	182
11.12	USI (USART + SPI + I2C)	184
11.12.1	Overview	184
11.12.2	USIn UART 模式	185
11.12.3	USIn UART 框图	186
11.12.4	USIn 时钟发生器	187
11.12.5	USIn 外部时钟 (SCK0)	188
11.12.6	USIn 同步模式	188
11.12.7	USIn UART 数据格式	189
11.12.8	USIn UART 校验位	190
11.12.9	USIn UART 发送	190
11.12.9.1	USIn UART 发送 TX 数据	190
11.12.9.2	USIn UART 发送标志位和中断	190
11.12.9.3	USIn UART 奇偶校验	191
11.12.9.4	USIn UART 禁止发送	191
11.12.10	USIn UART 接收	191
11.12.10.1	USIn UART 接收 RX 数据	191
11.12.10.2	USIn UART 接收标志位和中断	192
11.12.10.3	USIn UART 奇偶校验	192
11.12.10.4	USIn UART 禁止接收	192
11.12.10.5	USIn 异步数据接收	193
11.12.11	USIn SPI 模式	195
11.12.12	USIn SPI 时钟格式和时序	195

11.12.13	USIn SPI 框图	198
11.12.14	USIn I2C 模式	199
11.12.15	USIn I2C 位传送	199
11.12.16	USIn I2C 开始 / 重复开始/ 停止信号	200
11.12.17	USIn I2C 数据传送	200
11.12.18	USIn I2C 应答	201
11.12.19	USIn I2C 同步 / 仲裁	201
11.12.20	USIn I2C 操作	202
11.12.20.1	USIn I2C 主机发送	203
11.12.20.2	USIn I2C 主机接收	205
11.12.20.3	USIn I2C 从机发送	207
11.12.20.4	USIn I2C 从机接收	208
11.12.21	USIn I2C 方框图	209
11.12.22	寄存器图	210
11.12.23	USIn 寄存器说明	210
11.12.24	USIn 寄存器说明	211
11.12.25	波特率设置(举例)	220
11.13	LCD 驱动	221
11.13.1	概述	221
11.13.2	LCD Display RAM Organization	222
11.13.3	LCD 信号波形	223
11.13.4	LCD 分压电阻连接	227
11.13.5	方框图	229
11.13.6	寄存器图	229
11.13.7	LCD 驱动寄存器说明	229
11.13.8	LCD 驱动寄存器说明	230
12	省电操作	233
12.1	概述	233
12.2	IDLE/STOP 模式下外围操作	233
12.3	IDLE 模式	234
12.4	STOP Mode	235
12.5	STOP 模式唤醒	236
12.6	寄存器图	237
12.7	省电操作寄存器说明	237
12.8	省电操作寄存器说明	237
13	复位	238
13.1	概述	238
13.2	复位源	238
13.3	RESET 方框图	238
13.4	RESET 干扰消除	239
13.5	上电复位	239
13.6	外部复位	242
13.7	低电压检测处理	243
13.8	LVI 方框图	245
13.9	寄存器图	245
13.10	复位操作寄存器说明	245
13.11	复位操作寄存器	246
14	片内调试系统(MC96F6432 ONLY)	249
14.1	概述	249
14.1.1	说明	249
14.1.2	特性	250
14.2	双线外围接口	251
14.2.1	基本传输	251
14.2.2	数据包传输时序	252

14.2.2.1	数据发送	252
14.2.2.2	位传输	252
14.2.2.3	Start 和 Stop 条件	253
14.2.2.4	应答位	253
14.2.3	传输线路	254
15	Flash 存储器	255
15.1	概述	255
15.1.1	说明	255
15.1.2	Flash 存储器 ROM 结构	256
15.1.3	寄存器图	257
15.1.4	Flash 存储器控制和状态寄存器说明	257
15.1.5	Flash 寄存器说明	258
15.1.6	串行ISP模式	260
15.1.7	保护区域 (用户编程模式)	260
15.1.8	擦除模式	261
15.1.9	写模式	262
15.1.10	擦除\写入的保护	264
15.1.10.1	无效擦除/写入的保护流程	266
15.1.11	读模式	267
15.1.12	写保护模式	267
16	配置选项	268
16.1	配置选项控制	268
17	附录	269
17.1	指令表	269
17.2	Flash Protection for Invalid Erase/Write	273
目录	276