



HY17S58

User's Guide

8-Bit RISC-like Mixed Signal Microcontroller

Table of Contents

1. 阅读导览	8
1.1. 关于这份操作手册	8
1.2. 名词定义, Terms and Definition	9
1.2.1. 常用词汇索引	9
1.2.2. 寄存器相关字汇	10
2. 中央处理器, CPU	11
2.1. 处理器核心, CPU Core	11
2.2. 内存, Memory	12
2.2.1. 程序存储器, Program Memory	13
2.2.2. 数据存储器, DM	16
2.2.3. 寄存器列表-数据存储器	21
3. 振荡器、频率源与功耗管理	35
3.1. 振荡器	35
3.1.1. HAO 振荡器	35
3.1.2. LPO 振荡器	35
3.2. CPU 及外围电路频率源	36
3.2.1. 频率源分配	36
3.2.2. CPU 频率源	36
3.2.3. CPU 周边电路频率源	37
3.3. 寄存器说明-工作频率源控制器	39
3.4. 功率消耗管理与操作状态	42
3.4.1. 运行模式	42
3.4.2. 待机模式	42
3.4.3. 休眠模式	42
4. 复位, RESET	43
4.1. 复位事件说明	43
4.1.1. BOR 电源干扰复位	43
4.1.2. BOR1 为省电 BOR	44
4.1.3. BOR2 为精准 BOR	44

4.1.4.	SKERR 堆栈错误复位	44
4.2.	状态寄存器	44
4.2.1.	复位状态的时序图.....	45
4.3.	寄存器说明-复位状态.....	46
5.	中断,INTERRUPT.....	48
5.1.	寄存器说明-中断.....	49
6.	硬件乘法器.....	55
7.	输入/输出埠,I/O PORT.....	56
7.1.	PORT 相关寄存器介绍	60
7.1.1.	PTEG 中断信号产生条件	60
7.1.2.	PTPU 上拉电阻控制寄存器	60
7.1.3.	TC 输入/输出控制寄存器	60
7.1.4.	PTIO 状态控制寄存器.....	60
7.2.	蜂鸣器,Buzzer	60
7.3.	输入/输出埠 1 ,I/O Port1(以 HY17P58 为例)	61
7.4.	输入/输出埠 2 ,I/O Port2(以 HY17P58 为例)	62
7.1.	输入/输出埠 4 ,I/O Port4(以 HY17P58 为例)	63
7.2.	输入/输出埠 5 ,I/O Port5(以 HY17P58 为例)	64
7.3.	输入/输出埠 6 ,I/O Port6.....	65
7.4.	输入/输出埠 7 ,I/O Port7.....	66
7.5.	输入/输出埠 8 ,I/O Port8.....	67
7.6.	输入/输出埠 9 ,I/O Port9.....	68
7.7.	输入/输出埠 10 ,I/O Port10.....	69
7.8.	输入/输出埠 11 ,I/O Port11	70
7.9.	寄存器说明-PORT	71

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



8. 定电流输入/输出埠,I/O(HY17P48 ONLY)	80
8.1. 定电流模使用说明	80
8.1.1. 定电流模式初始设定	80
8.2. 寄存器说明-CC.....	81
9. 低电压检测, LOW VOLTAGE DETECT	82
9.1. 低电压侦测使用说明.....	82
9.1.1. LVD 初始设置	82
9.1.2. LVD 低电压发生条件及事件记录	83
9.1.3. LVD 启用方式	83
9.2. 寄存器说明-LVD	84
10. 看门狗,WATCH DOG	85
10.1. WDT 使用说明	85
10.1.1. WDT 初始化设置	85
10.1.2. WDT 中断事件服务.....	86
10.1.3. WDT 启用	86
10.2. WDT 控制寄存器列表与说明:	86
11. 定时计数器 TIMER-A1/2	88
11.1. 寄存器说明-TMA1/2	90
12. 16-BIT 定时计数器 B,TMB (16-BIT TIMERB)	92
12.1. TMB 四种计数模式.....	94
12.1.1. 16-bit 计数器.....	94
12.1.2. 17-bit 计数器.....	96
12.1.3. 两组 8-bit 计数器.....	98
12.1.4. 8+8-bit 计数器.....	100
12.2. PWM 脉冲宽度调变.....	102
12.2.1. PWM1O 波形 (16-bit PWM)	103
12.2.2. PWM2O 波形 (16-bit PWM)	105
12.2.3. PWM3O 波形 (8-bit PWM)	107
12.2.4. PWM4O 波形 (8-bit PWM)	109
12.2.5. PWM5O 波形 (8+8-bit PWM)	111

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



12.2.6.	PWM6O 形 (两个 16-bit PWM 波形)	115
12.2.7.	PWM7O 波形 (16-bit PWM 波形)	117
12.3.	TMB1 控制寄存器列表与说明:	119
13.	16-BIT 定时计数器 C,TMC (16-BIT TIMERC)	122
13.1.	寄存器说明-TMC	124
14.	电源系统,POWER SYSTEM.....	126
14.1.	VDDA 使用说明	126
14.1.1.	VDDA 初始化设置:	126
14.1.2.	VDDA 使用外部偏压:	126
14.1.3.	VDDA 启用	126
14.2.	寄存器说明-PWR.....	127
15.	模拟数字转换器,ΣADC.....	129
15.1.	Σ ADC 使用说明.....	135
15.1.1.	Σ ADC 初始化设置	135
15.2.	模拟通道输入特性	138
15.2.1.	TPS 初始化设置与计算方式	139
15.2.2.	TPS 范例说明	139
15.3.	寄存器说明- Σ ADC.....	141
16.	8-BIT RESISTANCE LADDER 网络	150
16.1.	寄存器说明-8-bit resistance ladder	153
16.2.	寄存器说明-Waveform Generator	154
17.	运算放大器 OPAMP.....	156
17.1.	寄存器说明- OPAMP	158
17.2.	寄存器说明-低通滤波器	160
18.	可程序放大器, R-TYPE PGA.....	161
18.1.	寄存器说明-可程序放大器	163

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



19. LCD 驱动器	166
19.1. LCD 使用说明	166
19.1.1. 工作频率与输出帧幅频率设置	166
19.1.2. 倍压电路与 LCD 工作电压设置	167
19.1.3. 闪烁设置	167
19.1.4. LCD 字节寄存器	167
19.2. LCD Type A 输出波形	168
19.3. LCD Type B 输出波形	169
19.4. 寄存器说明-LCD	170
20. LED 背光升压输出	175
20.1. 寄存器说明-LED 背光升压输出	176
21. 同步串行通讯接口, SERIAL PERIPHERAL INTERFACE	177
21.1. SPI 使用说明	178
21.2. SPI 主机模式	179
21.3. SPI 从机模式	181
21.4. SPI 主从机模式传输方式	184
21.5. 寄存器说明- SPI	187
22. 同步串行通讯接口, INTER-INTEGRATED CIRCUIT SERIAL INTERFACE	189
22.1. 数据传输速率计算	191
22.2. 超时计时功能(Time-Out)	191
22.3. I ² C 串行接口通讯流程图	192
22.4. I ² C 寄存器说明	198
23. 异步串行通讯接口, ENHANCED UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER	202
23.1. EUART 使用说明	203
23.1.1. 异步数据发送设置步骤(x=0 or 2)	203
23.1.2. 异步数据接收设置步骤(x=0 or 2)	203

23.1.3.	异步数据接收(9 位, RS-485 模式) 设置步骤(x=0 or 2).....	203
23.2.	串行传输波特率发生器 (BRG)	204
23.2.1.	功耗管理模式下的操作	204
23.2.2.	RC 取样方式	204
23.2.3.	自动波特率.....	204
23.3.	硬件同位检查.....	206
23.4.	EUART 异步模式.....	206
23.4.1.	EUART 异步发送器 (UART TXIF/RCIF flag 由 0->1 发生中断)	207
23.5.	寄存器说明- UART1/2	212
24.	内建 EPROM, BUILD-IN EPROM	215
24.1.	寄存器说明- BIE	217
25.	修订记录	218

1. 阅读导览

1.1. 关于这份操作手册

本文件所述的应用讯息及其他类似内容叙述仅为提供使用者便利，纭康对于内容的使用与因而引起的后果并不负担相关责任。规格内容随时可被更新讯息所替代，用户有责任必需承担并确保应用符合规范。

未经纭康授权，不得将纭康产品使用于生命维持系统中作为关键器件。纭康有不需事先通知即可修改产品的权力，产品最新讯息，请参考我们的网站：

<http://www.hycontek.com>

注意:

- 本说明书中的内容，随着产品的改进，有可能不经过预告而更改。请客户及时到本公司网站下载更新。
- 本规格书中的图形、应用电路等，因第三方工业所有权引发的问题，本公司不承担其责任。
- 本产品单独应用的情况下，本公司保证它的性能、典型应用和功能符合说明书中的条件。当使用在客户的产品或设备中，以上条件我们不作保证，建议客户做充分的评估和测试。
- 请注意输入电压、输出电压、负载电流的使用条件，使 IC 内的功耗不超过封装的容许功耗。对于客户在超出说明书中规定额定值使用产品，即使是瞬间的使用，由此所造成的损失，本公司不承担任何责任。
- 本产品虽内置防静电保护电路，但请不要施加超过保护电路性能的过大静电。
- 本规格书中的产品，未经书面许可，不可使用在要求高可靠性的电路中。例如健康医疗器械、防灾器械、车辆器械、车载器械及航空器械等对人体产生影响的器械或装置，不得作为其部件使用。
- 本公司一直致力于提高产品的质量和可靠度，但所有的半导体产品都有一定的失效概率，这些失效概率可能会导致一些人身事故、火灾事故等。当设计产品时，请充分留意冗余设计，采用安全指标，这样可以避免事故的发生。
- 本规格书中内容，未经本公司许可，严禁用于其他目的之转载或复制。

1.2. 名词定义, Terms and Definition

1.2.1. 常用词汇索引

1MW	1MegaWord	
1KB	1KiloByte	
ADC	Analog to Digital Converter	模拟数字转换器
Bit	bit	位
BOR	Brown-Out Reset	
BSR	Bank Select Register	
Byte	Byte	字节
CCP	Capture and Compare	捕捉器与比较器
CPU	Central Processing Unit	中央处理器
DAC	Digital-to-Analog Converter	数字模拟转换器
DM	Data Memory	数据存储器
ECAP	Enhance Comparator	增强型比较器
FSR	File Select Register	间接寻址指针寄存器
GPR	General Purpose Register	一般用途寄存器
HAO	High Accuracy Oscillator	高精度振荡器
LNOP	Low Noise OP AMP	低噪声放大器
LPO	Low Power Oscillator	低功耗消耗振荡器
LSB	Least Significant Bit	最低有效位
MEM	Memory	内存
MPM	Main Program Memory	
MSB	Most Significant Bit	最高有效位
OTP	One Time Program-EEPROM	一次性写入内存
PC	Program Counter	程序计数器
PPF	PWM and PFD	脉波宽度调整器与频率输出调整器
Σ ADC	Sigma-Delta ADC	模拟数字转换器
SR	Special Register	
SRAM	Static Random Access Memory	静态随机存取内存
STK	Stack	堆栈
WDT	Watch Dog Timer	看门狗定时器
WREG	Work Register	工作寄存器

1.2.2. 寄存器相关字汇

[]	Register length	寄存器长度
< >	Register value	寄存器内容
ABC[7:0]	ABC register had 0 to 7bit	ABC 寄存器总共有 8 位
ABC<111>	ABC register had 3bit and value had 111 of binary	ABC 寄存器总共有 3 位, 内容为二进制 111
ABC<11x>	x: can be neglected, it can be set as 1 or 0	ABC 寄存器总共有 3 位, 内容为二进制。可为 110 或 111
rw	Read/Write	可读可写
r	Read only	只读
r0	Read as 0	读值只有 0
r1	Read as 1	读值只有 1
w	Write only	唯写
w0	Write as 0	写入值只有 0
w1	Write as 1	写入值只有 1
h0	cleared by Hardware	硬件 置<0>
h1	set by Hardware	硬件 置<1>
u0	cleared by User	使用者 设置<0>
u1	set by User	使用者 设置<1>
-	Not use	未使用
!	users are forbidden to change	使用者禁止变更
u	unchanged	无法改变
x	unknown	未知
d	depends on condition	依照设定条件

2. 中央处理器, CPU

2.1. 处理器核心, CPU Core

中央处理器的核心 CPU Core(H08)为了使其有较高的执行效率，采用了 Harvard architecture 理念，将程序内存与数据存储器分别独立且程序内存的地址，增加了用户撰写程序的便利性。

CPU 特色包含：

- 程序内存与数据存储器各自独立设计架构，使得指令执行速度提升且提高 CPU 效率。
- 最多 71 个操作指令包含 16-bit 查表、8x8 硬件乘法器、数据存储器区块切换与堆栈控制
- 一个指令完成寄存器 A 至寄存器 B 的数据搬移且不改变工作寄存器(Work register)的数据
- 一个指令完成最长 16-bit 的 FSR 寄存器数据搬移与寻址 1MW 程序内存的查表指令。
- 数据存储器的操作包含程序计数器(PC)、状态寄存器(Status)与堆栈寄存器(Stack)的数据搬移。
- 处理器核心为精简版 H08D 核心。

2.2. 内存,Memory

内存的构成分为两种，一为程序存储器由 OTP 构成另一为数据存储器由 SRAM 构成。在不同型号的产品上，所规划的内存大小会不一样，故阅读各产品的说明书时必须特别留意该产品的规格说明。

程序存储器：

主存储器区(Main Program Memory,MPM)

程序计数器(Program Counter,PC)

堆栈寄存器(Stack,STK)

数据存储器：

特殊寄存器(Special Register,SR)

一般寄存器(General Purpose Register,GPR)

内存相关寄存器摘要： (x：表示由多个寄存器组成)

PC[12:0]	PCHSR[4:0],PCLATH[4:0],PCLATL[7:0]
TOS[12:0]	TOSH[4:0],TOSL[7:0]
FSRx[9:0]	FSRxH[9:8],FSRxL[7:0]
INDFx	INDF0[7:0],INDF1[7:0],INDF2[7:0]
POINCx	POINC0[7:0], POINC1[7:0], POINC2[7:0]
PODECx	PODEC0[7:0], PODEC1[7:0], PODEC1[7:0]
PRINCx	PRINC0[7:0], PRINC1[7:0], PRINC1[7:0]
PLUSWx	PLUSW0[7:0], PLUSW1[7:0], PLUSW1[7:0]
STKCN	STKFL[0],STKOV[0],STKUN[0],SKPRT[3:0]
PSTAT	SKERR[0]
BSRCN	BSR[1:0]

2.2.1. 程序存储器, Program Memory

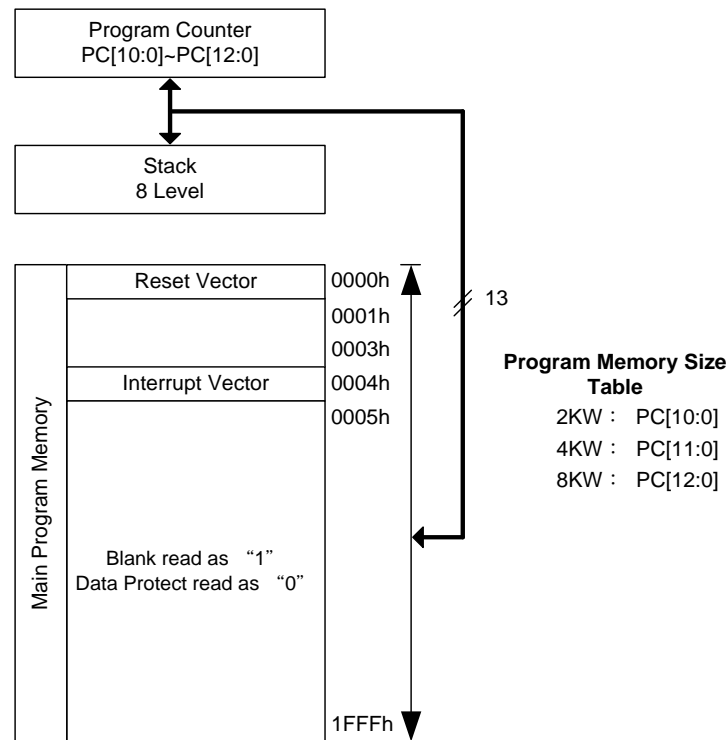


图 2-1 程序存储器架构图

2.2.1.1. 主存储器, MPM

主存储器架构如下：

- 中断服务向量位置(Interrupt Vector)
- 复位向量位置(Reset Vector)

寻址能力由 0x0000h 至 0x1FFFh，总计容量为 8192 位，依不同型号产品大小会有所不同。

芯片在未进行程序写入时，所有地址的数据状态均为 1；写入后，地址将依写入的数据状态呈现 1 或 0。必需注意，程序开发时若仿真软件(HYIDE)的组译选项有设置烧录保护功能，则芯片在烧录所能读出的地址数据状态皆为 0。

2.2.1.2. 程序计数器,PC

程序计数器 PC 由位移寄存器 PCSRH、缓冲寄存器 PCLAT 组成，如图 2-2。

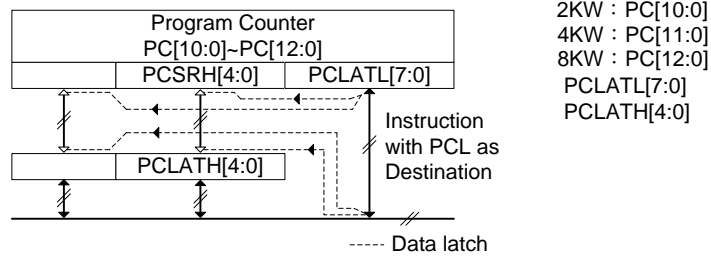


图 2-2 程序计数器架构图

程序计数器 PC[12:0]在开发工具所使用的芯片具有 13 位的数据长度，由两个特殊寄存器 PCSRH [4:0]与 PCLATL [7:0]组成。其中 PCLATL[7:0]与 PCLATH[4:0]可直接读/写，而 PCSRH [4:0]无法直接读/写，必须透过缓冲寄存器 PCLATH[4:0]做间接读/写。

- 读取 PC[12:0]，必须先读取 PCLATL[7:0]接着读取 PCLATH[4:0]才能取得正确数据，顺序反之则将读取到不正确的数据。
- 写入 PC[12:0]，必须先写入 PCLATH[4:0]最后再写 PCLATL[7:0]，顺序反之会写入不正确的数据。

2.2.1.3. 堆栈,STK

堆栈 STK 主要由堆栈指针控制寄存器 STKCN、堆栈错误标志位 SKERR(Stack Error)与堆栈错误复位控制器 SKRST 组成。

当堆栈发生溢位与欠位时可能导致程序有不预期的执行结果，必要时可透过设置重新启动芯片。在程序开发过程中，透过软件设置可将堆栈复位控制位 SKRST¹ 设置<1>，当堆栈发生欠位或溢位时会产生复位信号并将 SKERR 置<1>后重新启动芯片。

- 满位：STKFL 置<1>，PC[12:0]不受影响。
- 欠位：STKUN 置<1>，PC[12:0]移至 0x00000h 位置堆栈指针 SKPRT 指向 0 Level。若 SKRST 设置<1>，则欠位后会产生复位信号且 SKERR 置<1>，复位后 STKUN 置<0>。
- 溢位：STKOV 置<1>，PC[12:0]不受影响但 SKPRT 仍停滞于最后一层且会压入新的数值，即满位后会保存最近一次压入的资料。若 SKRST 设置<1>，则溢位后产生复位信号且 SKERR 置<1>，复位后 STKOV 置<0>。
- 错误：SKERR 置<1>，芯片已发生堆栈错误。若 SKRST 设置<1>，则溢位后产生复位信号且 SKERR 置<1>，复位后 STKUN、STKOV 置<0>。

¹ SKRST 为堆栈错误产生复位信号控制位，无法直接读/写只能在程序开发阶段透过开发软件的设置。即程序开发阶段必须选定是否在堆栈错误时产生复位信号，若选定复位则芯片供电后该位即被设置 1，反之设置<0>。

2.2.1.4. 寄存器说明-程序忆体控制器

“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
Address	File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	R/W
018h	SKCN	SKFL	SKUN	SKOV	-	-	SKPRT[2:0]			000. .000	u\$. .\$\$\$
01Ah	PCLATH	-	-	-	PC[12]	PC[11]	PC[10]	PC[9]	PC[8]0000 0000
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000
02Ch	PSTATUS	POR	PD	TO	IDL	RST	SKERR	MCO	-	\$000 \$00.	uu\$u u\$u.

表 2-1 程序忆体控制寄存器

STKCN: 堆栈控制器

位	名称	描述
Bit7	STKFL	堆栈满位标志 <0> 未发生。 <1> 已发生。
Bit6	STKUN	堆栈欠位标志 <0> 未发生。 <1> 已发生。
Bit5	STKOV	堆栈溢出标志 <0> 未发生。 <1> 已发生。
Bit3~0	SKPRT[2:0]	堆栈指针寄存器 <000> 第 0 层, TOS[12:0]=0x0000h <110> 第 6 层 <111> 第 7 层

PCLATH : 程序计数器高字节, PC[12:8]

PCLATL: 程序计数器低字节, PC[7:0]

PSTAT: 状态寄存器

位	名称	描述
Bit2	SKERR	堆栈错误产生复位标志 <0> 未发生。 <1> 已发生。

2.2.2. 数据存储器,DM

数据存储器DM由特殊寄存器 Specially Register,SR 与一般寄存器 General Purpose Register,GPR 组成,且以每 256byte 为一个区块。区块 0 与区块 1 较为特殊,这两个区块各自包含了 128byte 的特殊寄存器与 128byte 一般寄存器,而其他的区块则 256byte 全为一般寄存器如 图 2-3。

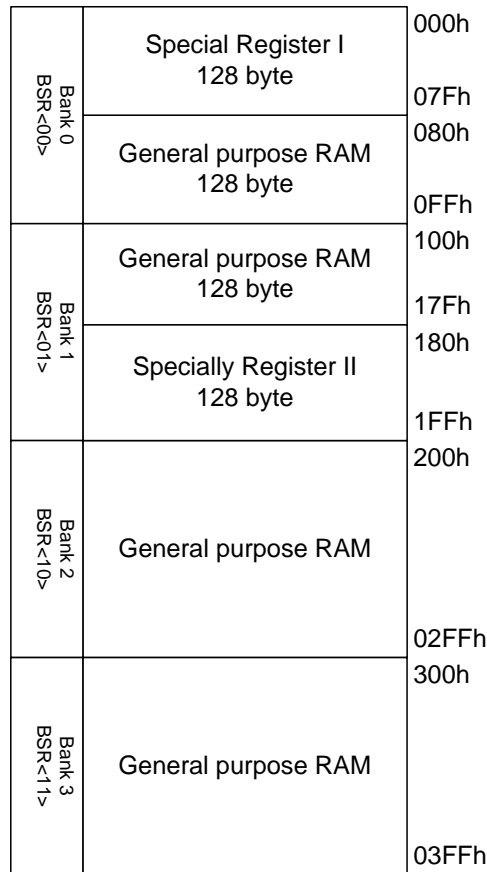


图 2-3 数据存储器架构图

2.2.2.1. 内存与指令

H08 指令集可分为 A、B、D 三版本其在内存运用有很大的差异性,例如寻址能力、硬件乘法器、查表指令、支持功能与参数的定义,在此仅说明指令内存参数的定义。详细的指令参数说明请参见指令集,Instruction 章节。

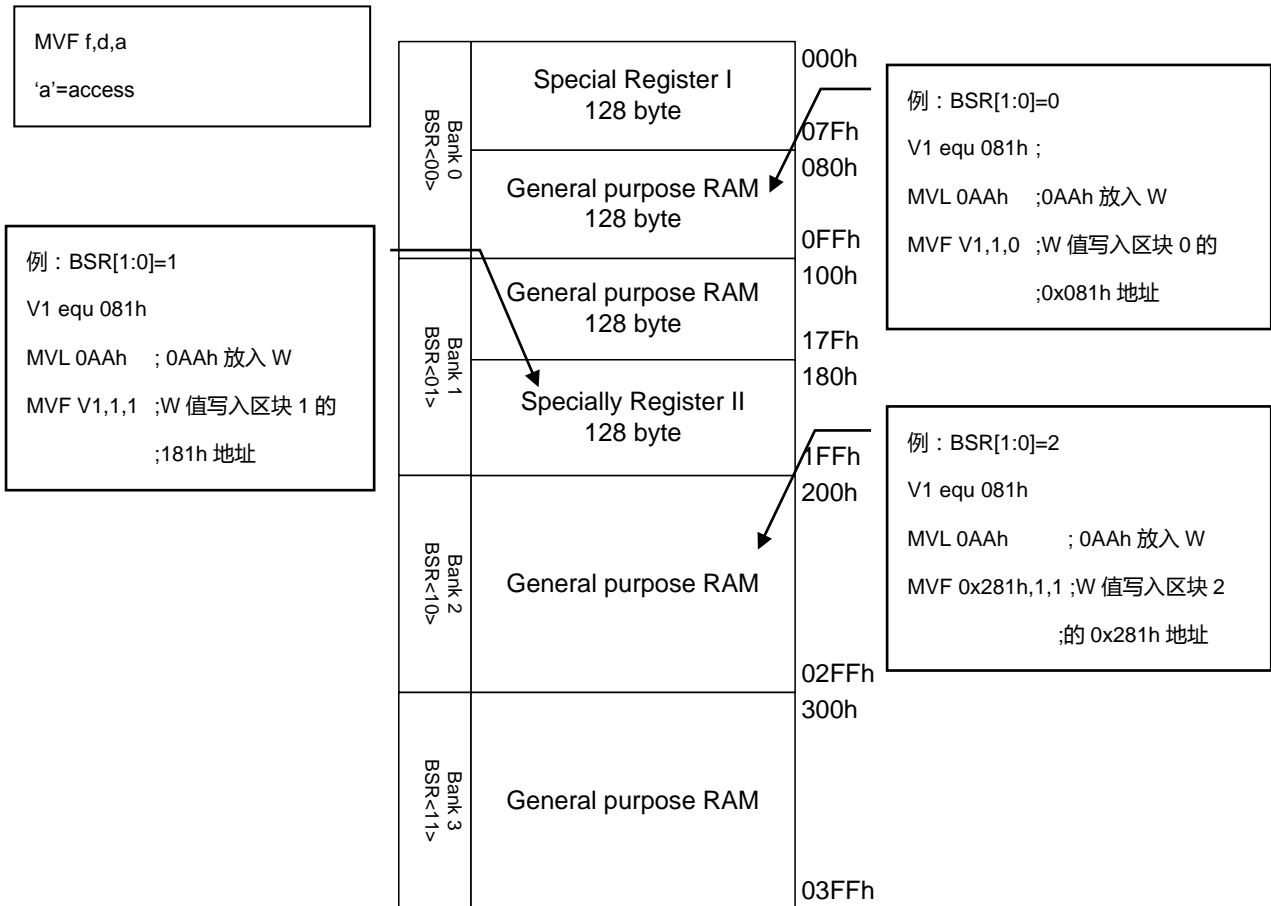
指令集中带有地址运算功能的指令至多会有“f”、“d”、“a”等三个参数。

- “f” 是指数据(Data)或数据寄存器地址(Data Memory Address)。
- “d” 是指运算后的数据要存放地方。d=0 存于 WREG register、d=1 存于 Data Memory Register。
- “a” 是指定数据存储器操作的区块; a=0 操作于区块 0、a=1 操作于 BSR[1:0]指定区块。

2.2.2.2. 区块选择控制寄存器

数据存储器规划为每 256byte 为一个区块即 000h ~ 0FFh 为一区块,若欲读/写地址 0FFh 以后的数据寄存器,则需正确的设置区块控制寄存器 BSR[1:0]与指令的参数“a”,说明如下:

- 当 a = 0 时,无论 BSR[1:0]的指定何区块,指令对数据存储器的读/写只会在区块 0。
- 当 a = 1 时,H08D CPU Core 的指令,对数据存储器的读/写会依照 BSR[1:0]所指定的区块。



范例 2-4 区块选择寄存器范例程序与数据存储器关系

2.2.2.3. 特殊寄存器

特殊寄存器包含 CPU Core 与外围功能的相关寄存器,主要有控制功能寄存器与数据传回寄存器。若对数据寄存器内未定义的地址或地址用的位进行读取,所读取到的数据为 0。

在特殊寄存器中专用于搭配指令的寄存器亦有数个,但在此只介绍两种常用的寄存器一为工作寄存器 WREG,另一为间接寻址寄存器 FSR。其余在此未介绍的特殊寄存器将分散于各章节做详尽的说明。

2.2.2.3.1. 工作寄存器,WREG

工作寄存器简称 W 为搭配指令使用最为频繁的寄存器,举凡数据搬移、运算与判断等等。

2.2.2.3.2. 间接寻址寄存器,FSR 与 INDF

FSR 间接寻址寄存器由指针寄存器 FSR0[9:0]、FSR1[9:0]、FSR2[9:0]与索引寄存器 INDF0[7:0]、INDF1[7:0]、INDF2[7:0]组成，由于功能相当故只针对 FSR0 做说明。

FSR0[9:0]可分为 FSR0H[1:0]与 FSR0L[7:0]两个寄存器，不需设置 BSR[1:0]即能寻址不同区块的地址；且透过特殊指令，可达到使用一个指令即可写入 16-bit 数据。

INDF0[7:0]为索引寄存器，即是可读取 FSR0[9:0]所指向数据存储器地址的数据。

值得注意的是 H08D 指令集支持加强型索引寄存器，功能描述如下：

- POINC0[7:0]: 当透过指令读/写 POINC0[7:0]寄存器时会发生以下事件
 - 先传回目前 FSR0[9:0]所指向地址的内容。
 - 然后指针寄存器 FSR0[9:0]的数值加一指向下一个地址。
- PODEC0[7:0]: 当透过指令读/写 PODEC0[7:0]寄存器时会发生以下事件
 - 先传回目前 FSR0[9:0]所指向地址的内容。
 - 然后指针寄存器 FSR0[9:0]的数值减一指向上一个地址。
- PRINC0[7:0]: 当透过指令读/写 PRINC0[7:0]寄存器时会发生以下事件
 - 先将指针寄存器 FSR0[9:0]的数值加一指向下一个地址。
 - 再传回目前 FSR0[9:0]所指向地址的内容。
- PLUSW0 [7:0]: 当透过指令读/写 PLUSW0 [7:0]寄存器时会发生以下事件
 - 先将指针寄存器 FSR0[9:0]的数值加上工作寄存器 W 的内容。
 - 再传回目前 FSR0[9:0]所指向地址的内容。其中 W 的内容为带有符号位的数值即 $\pm 128d$ 。

2.2.2.3.3. 一般寄存器, General Purpose Register

一般寄存器 GPR 为用户进行数据储存、运算、旗标设置等等自由规划区域。

2.2.2.4. 寄存器说明-数据存储器控制器

“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
Address	File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	R/W
000h	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								XXXX XXXX	***** 1 1 1 1 1 1 1
001h	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-incremented								XXXX XXXX	***** 1 1 1 1 1 1 1
002h	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decremented								XXXX XXXX	***** 1 1 1 1 1 1 1
003h	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-incremented								XXXX XXXX	***** 1 1 1 1 1 1 1
004h	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								XXXX XXXX	***** 1 1 1 1 1 1 1
005h	INDF1	Contents of FSR1 to address data memoryvalue of FSR0 not changed								XXXX XXXX	***** 1 1 1 1 1 1 1
006h	POINC1	Contents of FSR1 to address data memoryvalue of FSR0 post-incremented								XXXX XXXX	***** 1 1 1 1 1 1 1
007h	PODEC1	Contents of FSR1 to address data memoryvalue of FSR0 post-decremented								XXXX XXXX	***** 1 1 1 1 1 1 1
008h	PRINC1	Contents of FSR0 to address data memoryvalue of FSR1 pre-incremented								XXXX XXXX	***** 1 1 1 1 1 1 1
009h	PLUSW1	Contents of FSR1 to address data memoryvalue of FSR0 offset by W								XXXX XXXX	***** 1 1 1 1 1 1 1
00Ah	INDF2	Contents of FSR2 to address data memoryvalue of FSR2 not changed								XXXX XXXX	***** 1 1 1 1 1 1 1
00Bh	POINC2	Contents of FSR2 to address data memoryvalue of FSR2 post-incremented								XXXX XXXX	***** 1 1 1 1 1 1 1
00Ch	PODEC2	Contents of FSR2 to address data memoryvalue of FSR2 post-decremented								XXXX XXXX	***** 1 1 1 1 1 1 1
00Dh	PRINC2	Contents of FSR2 to address data memoryvalue of FSR2 pre-incremented								XXXX XXXX	***** 1 1 1 1 1 1 1
00Eh	PLUSW2	Contents of FSR2 to address data memoryvalue of FSR2 offset by W								XXXX XXXX	***** 1 1 1 1 1 1 1
00Fh	FSR0H	-	-	-	-	-	-	FSR0[9]	FSR0[8] XXXX	~..~.. ***** 1 1 1 1 1 1 1
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								XXXX XXXX	***** 1 1 1 1 1 1 1
011h	FSR1H	-	-	-	-	-	-	FSR1[9]	FSR1[8]XXX	~..~.. ***** 1 1 1 1 1 1 1
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								XXXX XXXX	***** 1 1 1 1 1 1 1
013h	FSR2H	-	-	-	-	-	-	FSR2[9]	FSR2[8]XXX	~..~.. ***** 1 1 1 1 1 1 1
014h	FSR2L	Indirect Data Memory Address Pointer 0 Low Byte,FSR2[7:0]								XXXX XXXX	***** 1 1 1 1 1 1 1
016h	TOSH	-	-	-	TOS[12]	TOS[11]	TOS[10]	TOS[9]	TOS[8]	..XX XXXX	~.~ ***** 1 1 1 1 1 1 1
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								XXXX XXXX	***** 1 1 1 1 1 1 1
029h	WREG	Working Register								XXXX XXXX	***** 1 1 1 1 1 1 1
02Ah	BSRCN	-	-	-	-	-	-	BSR[1]	BSR[0] XXXX	~..~.. ***** 1 1 1 1 1 1 1

表 2-2 数据存储器控制寄存器

INDFx/POINCx/PODECx/PRINCx/PLUSWx: 不同功能性的索引寄存器(x=0、1、2)

INDFx[7:0]: 详见 2.2.2.3.2 间接寻址寄存器,FSR 与说明

POINCx[7:0]: 详见 2.2.2.3.2 间接寻址寄存器,FSR 与说明

PODECx[7:0]: 详见 2.2.2.3.2 间接寻址寄存器,FSR 与说明

PRINCx[7:0]: 详见 2.2.2.3.2 间接寻址寄存器,FSR 与说明

PLUSWx[7:0]: 详见 2.2.2.3.2 间接寻址寄存器,FSR 与说明

FSRx: 间接寻址的指针寄存器(x=0、1、2)

FSRxH[1:0]: 详见 2.2.2.3.2 间接寻址寄存器,FSR 与说明

FSRxL[7:0]: 详见 2.2.2.3.2 间接寻址寄存器,FSR 与说明

WREG: 工作寄存器

WREG[7:0]: 详见 2.2.2.3.1 工作寄存器说明

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

BSRCN: 数据存储器区块读/写控制寄存器

位	名称	描述
Bit1~0	BSR[1:0]	数据存储器读/写区块指针寄存器 <00> 区块 0, 地址 0x000h~0x0FFh <01> 区块 1, 地址 0x100h~0x1FFh <10> 区块 0, 地址 0x200h~0x2FFh。 <11> 区块 0, 地址 0x300h~0x3FFh。

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



2.2.3. 寄存器列表-数据存储器

2.2.3.1. HY17P52 数据存储器

“-”no use,“r”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1
“\$”for event status,“u”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
000h	INDF0	Contents of FSR0 to address data memory value of FSR0 not changed								xxxx xxxx	uuuu uuuu	***** r,r
001h	POINC0	Contents of FSR0 to address data memory value of FSR0 post-incremented								xxxx xxxx	uuuu uuuu	***** r,r
002h	PODEC0	Contents of FSR0 to address data memory value of FSR0 post-decremented								xxxx xxxx	uuuu uuuu	***** r,r
003h	PRINC0	Contents of FSR0 to address data memory value of FSR0 pre-incremented								xxxx xxxx	uuuu uuuu	***** r,r
004h	PLUSW0	Contents of FSR0 to address data memory value of FSR0 offset by W								xxxx xxxx	uuuu uuuu	***** r,r
005h	INDF1	Contents of FSR1 to address data memory value of FSR1 not changed								xxxx xxxx	uuuu uuuu	***** r,r
006h	POINC1	Contents of FSR1 to address data memory value of FSR1 post-incremented								xxxx xxxx	uuuu uuuu	***** r,r
007h	PODEC1	Contents of FSR1 to address data memory value of FSR1 post-decremented								xxxx xxxx	uuuu uuuu	***** r,r
008h	PRINC1	Contents of FSR1 to address data memory value of FSR1 pre-incremented								xxxx xxxx	uuuu uuuu	***** r,r
009h	PLUSW1	Contents of FSR1 to address data memory value of FSR1 offset by W								xxxx xxxx	uuuu uuuu	***** r,r
00Fh	FSR0H	-	-	-	-	-	-	-	FSR0[8] x u	-,-,-,-,-,-,-,-
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu	***** r,r
011h	FSR1H	-	-	-	-	-	-	-	FSR1[8] x u	-,-,-,-,-,-,-,-
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu	***** r,r
016h	TOSH	-	-	-	-	TOS[11:8]		 xxxx uuuu	-,-,-,-,-,-,-,-	
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu	***** r,r
018h	SKCN	SKFL	SKUN	SKOV	-	SKPRT[3:0]			000. 0000	u\$. \$\$\$	rw 0,rw 0,rw 0,-	
01Ah	PCLATH	-	-	-	-	PC[11:8]		 0000 0000	-,-,-,-,-,-,-,-	
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000	***** r,r
01Dh	TBLPTRH	-	-	-	-	TBLPTR[11:8]		 xxxx uuuu	-,-,-,-,-,-,-,-	
01Eh	TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								xxxx xxxx	uuuu uuuu	***** r,r
01Fh	TBLDH	Program Memory Table Latch High Byte								xxxx xxxx	uuuu uuuu	***** r,r
020h	TBLDL	Program Memory Table Latch Low Byte								xxxx xxxx	uuuu uuuu	***** r,r
021h	PRODH	Product Register of Multiply High Byte								xxxx xxxx	uuuu uuuu	***** r,r
022h	PRODL	Product Register of Multiply Low Byte								xxxx xxxx	uuuu uuuu	***** r,r
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	-	-	E1IE	E0IE	0000 0000	0uuu uuuu	***** r,r
024h	INTE1	TA1IE	-	TXIE	RCIE	-	-	-	-	0000 0000	uuuu uuuu	***** r,r
026h	INTF0	-	TA1CIF	ADIF	WDTIF	-	-	E1IF	E0IF	.000 0000	.uuu uuuu	***** r,r
027h	INTF1	TA1IF	-	TXIF	RCIF	-	-	-	-	0000 0000	uuuu uuuu	**** r,r
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu	***** r,r
02Ah	BSRCN	-	-	-	-	-	-	-	BSR[0] x u	-,-,-,-,-,-,-,-
02Bh	MSTAT	-	-	-	C	DC	N	OV	Z	..x xxxx	..u uuuu	-,-,-,-,-,-,-,-
02Ch	PSTAT	BOR	PD	TO	IDL	RST	SKERR	-	-	\$000 \$0..	uu\$u u\$.	rw 0,rw 0,rw 0,rw 0,rw 0,-,-
02Eh	BIECN	1	-	-	ENBVD	VPPHV	ENBCP	BIEWR	BIERD	1.00 \$000	uu\$u u\$.	r1,-,-,-,-,-,-,-
02Fh	BIEARH	-	-	1	1	1	1	1	1	0.xx xxxx	u.uu uuuu	***** r,r
030h	BIEARL	BIE Address Register as BIEAL[5:0]								xxxx xxxx	uuuu uuuu	***** r,r
031h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	***** r,r
032h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	***** r,r
033h	PWRCON	ENBGR	LDOC[2:0]		LDM[1:0]		ENLDO	CSFON		0000 0000	uuuu u00u	*****,w r0,w r0,*
034h	OSCCN0	-	OCS[0]	DHS[1:0]		DMS[2:0]		CUPS		0000 0000	uuuu uuuu	***** r,r
035h	OSCCN1	-	-	DADC[1:0]		-	-	-	LCDS	0000 0000	uuuu uu.	***** r,r
036h	OSCCN2	DLCD[1:0]		-	-	-	HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	***** r,r
037h	CSFCN0	SKRST	HAOTR[6:0]						.1..	-,-,-,-,-,-,-,-
038h	CSFCN1	ENSDRV	-	-	-	-	-	-	-	uuuu uuuu	***** r,r
039h	WDTCN	ENBZ	BZS	DBZ[1:0]		ENWDT	DWDT[2:0]		0000 0000	uuuu \$000	***** rw 1,*,*	
03Ah	AD1H	ADC1 conversion high byte data register								..00 0000	..uu uuuu	***** r,r
03Bh	AD1M	ADC1 conversion middle byte data register								0000 0000	uuuu uuuu	***** r,r
03Ch	AD1L	ADC1 conversion low byte data register								0000 0000	uuuu uuuu	***** r,r
03Dh	AD1CN0	ENAD1	-	OSR[3:0]				CMFR		0... 0000	uuu. uuuu	***** r,r
03Eh	AD1CN1	-	VREGN	PGAGN[1:0]		ADGN[2:0]		x.xx xxxx		u.uu uuuu	***** r,r	
03Fh	AD1CN2	INIS1	-	-	-	DCSET[3:0]	 xxxx	 uuuu	***** r,r	

表 2-3 数据存储器列表

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“.”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
040h	AD1CN3	INP[3:0]			INN[3:0]					xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
041h	AD1CN4	VRH[1:0]		VRL[1:0]		INX[1:0]		VRIS	INIS	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
042h	AD1CN5	ENACM	-	VCMS	LDOPL	-	-	ENTPS	TPSCH	0000 ..0.	uuuu ..u.	***** 1 1 1 1 1 1 1
043h	LVDCN			PWRS			LVDS[3:0]		LVDO	..00 0000	..uu uuuu	***** 1 1 1 1 1 1 1
044h	TMA1CN	ENTMA1	TMACL1	TMAS1	DTMA1[2:0]			-	-	0000 000.	u0uu uuu.	*rw 1 ***** 1 1 1 1 1 1 1
045h	TMA1R	TMA1 counter Register								0000 0000	uuuu uuuu	rw 0,rw 0,rw 0,rw 0,rw 0,rw 0,rw 0,rw 0
046h	TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0,rw0,rw0,rw0,rw0
047h	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	xxxx xxxx	***** 1 1 1 1 1 1 1
048h	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4	TC1.3	TC1.2	TC1.1	TC1.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
049h	PT1DA	-	-	-	-	-	DA1.2	-	-	xxxx x1xx	uuuu uuuu	***** 1 1 1 1 1 1 1
04Ah	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
04Bh	PT1M1	-	-	-	-	INTEG1[1:0]		INTEG0[1:0]		0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
04Ch	PT2	-	-	-	-	-	-	PT2.1	PT2.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
04Dh	TRISC2	-	-	-	-	-	-	TC2.1	TC2.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
04Fh	PT2PU	-	-	-	-	-	-	PU2.1	PU2.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
050h	PT8	-	-	-	-	-	-	PT8.1	PT8.0	xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
051h	TRISC8	-	-	-	-	-	-	TC8.1	TC8.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
053h	PT8PU	-	-	-	-	-	-	PUB.1	PUB.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
054h	UROCN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	***** 1 1 1 1 1 1 1
055h	UROSTA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	- ,r,r,r r,r,r,rw 0
056h	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD 0000 uuuu	- ,r,r,r,r ***** 1 1 1 1 1 1 1
057h	BG0RH	-	-	-	Baud Rate Generator Register High Byte				...x xxxx	...u uuuu	- ,r,r,r,r ***** 1 1 1 1 1 1 1	
058h	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
059h	TX0R	UART Transmit Register								xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
05Ah	RC0REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r,r,r
05Bh	LDCN1	ENLCP	LCDV[2:0]			ENLB	SEL_PCLK	-	LCDPU	0000 00.0	uuuu uu.u	***** 1 1 1 1 1 1 1
05Ch	LDCN2	-	-	-	-	-	-	LcdbL	LCI00uu	***** 1 1 1 1 1 1 1
05Dh	LDCN3	SCM3[1:0]		SCM2[1:0]		SCM1[1:0]		SCM0[1:0]		1111 1111	uuuu uuuu	***** 1 1 1 1 1 1 1
05Eh	LDCN4	-	-	-	-	-	-	SSG15	SSG14	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
05Fh	LCD0	LCD SEG3[4:7] data				LCD SEG2[3:0] data				xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
060h	LCD1	LCD SEG5[4:7] data				LCD SEG4[3:0] data				xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
061h	LCD2	LCD SEG7[4:7] data				LCD SEG6[3:0] data				xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
062h	LCD3	LCD SEG9[4:7] data				LCD SEG8[3:0] data				xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
063h	LCD4	LCD SEG11[4:7] data				LCD SEG10[3:0] data				xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
064h	LCD5	LCD SEG13[4:7] data				LCD SEG12[3:0] data				xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
065h	LCD6	LCD SEG15[4:7] data				LCD SEG14[3:0] data				xxxx xxxx	uuuu uuuu	***** 1 1 1 1 1 1 1
180h ~ 17Fh	SRAM as 256Byte									uuuu uuuu	uuuu uuuu	***** 1 1 1 1 1 1 1

表 2-4 数据存储寄存器列表(续)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



2.2.3.2. HY17P55/56 数据存储器

“-”no use,“r”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1

“\$”for event status,“-”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W	
000h	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								xxxx xxxx	uuuu uuuu	*****	
001h	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-incremented								xxxx xxxx	uuuu uuuu	*****	
002h	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decremented								xxxx xxxx	uuuu uuuu	*****	
003h	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-incremented								xxxx xxxx	uuuu uuuu	*****	
004h	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								xxxx xxxx	uuuu uuuu	*****	
005h	INDF1	Contents of FSR1 to address data memoryvalue of FSR1 not changed								xxxx xxxx	uuuu uuuu	*****	
006h	POINC1	Contents of FSR1 to address data memoryvalue of FSR1 post-incremented								xxxx xxxx	uuuu uuuu	*****	
007h	PODEC1	Contents of FSR1 to address data memoryvalue of FSR1 post-decremented								xxxx xxxx	uuuu uuuu	*****	
008h	PRINC1	Contents of FSR1 to address data memoryvalue of FSR1 pre-incremented								xxxx xxxx	uuuu uuuu	*****	
009h	PLUSW1	Contents of FSR1 to address data memoryvalue of FSR1 offset by W								xxxx xxxx	uuuu uuuu	*****	
00Ah	INDF2	Contents of FSR2 to address data memoryvalue of FSR2 not changed								xxxx xxxx	uuuu uuuu	*****	
00Bh	POINC2	Contents of FSR2 to address data memoryvalue of FSR2 post-incremented								xxxx xxxx	uuuu uuuu	*****	
00Ch	PODEC2	Contents of FSR2 to address data memoryvalue of FSR2 post-decremented								xxxx xxxx	uuuu uuuu	*****	
00Dh	PRINC2	Contents of FSR2 to address data memoryvalue of FSR2 pre-incremented								xxxx xxxx	uuuu uuuu	*****	
00Eh	PLUSW2	Contents of FSR2 to address data memoryvalue of FSR2 offset by W								xxxx xxxx	uuuu uuuu	*****	
00Fh	FSR0H	-	-	-	-	-	-	-	FSR0[9:8]	...xx	...uu	*****	
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu	*****	
011h	FSR1H	-	-	-	-	-	-	-	FSR1[9:8]	...xx	...uu	*****	
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu	*****	
013h	FSR2H	-	-	-	-	-	-	-	FSR2[9:8]	...xx	...uu	*****	
014h	FSR2L	Indirect Data Memory Address Pointer 0 Low Byte,FSR2[7:0]								xxxx xxxx	uuuu uuuu	*****	
016h	TOSH	-	-	-	-	-	-	-	TOS[12:8]	...x xxxx	...u uuuu	*****	
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu	*****	
018h	SKCN	SKFL	SKUN	SKOV						000.0000	u\$. \$.\$\$\$	rw0,rw0,rw0,-	*****
01Ah	PCLATH	-	-	-						PC[12:8]	...0 0000	...0 0000	*****
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000	*****	
01Dh	TBLPTRH	-	-	-						TBLPTR[12:8]	...x xxxx	...u uuuu	*****
01Eh	TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								xxxx xxxx	uuuu uuuu	*****	
01Fh	TBLDH	Program Memory Table Latch High Byte								xxxx xxxx	uuuu uuuu	*****	
020h	TBLDL	Program Memory Table Latch Low Byte								xxxx xxxx	uuuu uuuu	*****	
021h	PRODH	Product Register of Multiply High Byte								xxxx xxxx	uuuu uuuu	*****	
022h	PRODL	Product Register of Multiply Low Byte								xxxx xxxx	uuuu uuuu	*****	
023h	INTE0	GIE	TA1CIE	ADIE	WDIE	TB1IE			E1IE	E0IE	0000 0000	0uuu uuuu	*****
024h	INTE1	TA1IE	SPIIE	TXIE	RCIE	I2CERIE	I2CIE		E3IE	E2IE	0000 0000	uuuu uuuu	*****
025h	INTE2	-	-	-	-	TX2IE	RC2IE		-	BOR2IE	0000 0000	uuuu uuuu	*****
026h	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF			E1IF	E0IF	.000 0000	.uuu uuuu	*****
027h	INTF1	TA1IF	SPIIF	TXIF	RCIF	I2CERIF	I2CIF		E3IF	E2IF	0000 0000	uuuu uuuu	*****
028h	INTF2	-	-	-	-	TX2IF	RC2IF		-	BOR2IF	0000 0000	uuuu uuuu	*****
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu	*****	
02Ah	BSRCN	-	-	-	-	-	-	-	BSR[1:0]	...xx	...uu	*****	
02Bh	MSTAT	-	-	-	C	DC	N		OV	Z	...x xxxx	...u uuuu	*****
02Ch	PSTAT	BOR	PD	TO	IDL	RST	SKERR		BOR2LV	GCRstIF	\$000 \$000	uu\$u u\$uu	rw0,rw0,rw0,rw0,rw0,rw0
02Eh	BIECN	1	-	-	ENBVD	VPPHV	ENBCP		BIEWR	BIERD	1.00 \$000	1.00 \$uuu	r1,*,*,*,*,*
02Fh	BIEARH	-	-	1	1	1	1		1	1	0.xx xxxx	u.uu uuuu	*****
030h	BIEARL	-	-	BIE Address Register as BIEAL[5:0]						xxxx xxxx	uuuu uuuu	*****	
031h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	*****	
032h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	*****	
033h	PWRCN	ENBGR	LDOC[2:0]			LDOM[1:0]		ENLDO	CSFON		0000 0000	uuuu u00u	***** ,wr0,wr0,*
034h	OSCCN0	OSCS[1:0]			DHS[1:0]		DMS[2:0]		CUPS		0000 0000	uuuu uuuu	*****
035h	OSCCN1	CCOPT	LCPS	DADC[1:0]		DTMB[1:0]		TMBS	LCDS		0000 0000	uuuu uu.	*****
036h	OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO		0000 0001	uuuu uu01	*****
037h	CSFCN0	SKRST	HAOTR[6:0]								.1..	*****
038h	CSFCN1	ENSDRV	-	-	BOR_TH[2:0]			BORS	ENBOR2		...0 0011	uuuu uuuu	*****
039h	WDTCN	ENBZ	BZS	DBZ[1:0]		ENWDT	DWDWT[2:0]				0000 0000	uuuu \$000	***** rw1,*,*

表 2-5 数据存储器列表

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“.”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
03Ah	AD1H	ADC1 conversion high byte data register								..00 0000	..uu uuuu
03Bh	AD1M	ADC1 conversion middle byte data register								0000 0000	uuuu uuuu
03Ch	AD1L	ADC1 conversion low byte data register								0000 0000	uuuu uuuu
03Dh	AD1CN0	ENAD1	-	OSR[3:0]			CMFR		000. 0000	uuu. uuuu	
03Eh	AD1CN1	-	-	VREGN	PGAGN[1:0]		ADGN[2:0]		xxxx xxxx	uuuu uuuu	
03Fh	AD1CN2	INS1	-	-	-	DCSET[3:0]			xxxx xxxx	uuuu uuuu	
040h	AD1CN3	INP[3:0]			INN[3:0]			xxxx xxxx	uuuu uuuu		
041h	AD1CN4	VRH[1:0]		VRL[1:0]		INX[1:0]		VRIS	INIS	0000 0000	uuuu uuuu
042h	AD1CN5	ENACM	-	VCMS	LDOPL	-	-	ENTPS	TPSCH	0000 0000	uuuu uuuu
043h	LVDCN	-	-	PWRS	LVDS[3:0]			LVDO		0000 0000	uuuu uuuu
044h	TMA1CN	ENTMA1	TMA1CL	TMA1S	DTMA1[2:0]		-	-	0000 00.0	u0uu uu.u	*,rw 1,*,*,*,*	
045h	TMA1R	TMA1 counter Register								0000 0000	uuuu uuuu	rw0,rv0,rv0,rv0,rv0,rv0,rv0,rv0
046h	TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	rw0,rv0,rv0,rv0,rv0,rv0,rv0,rv0
047h	TB1Flag	-	PWM7A	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	..00 0000	..uu uuuu	.,r,f,r,f,r,f,r,f
048h	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	PWMO1	PWMO0	0000 0000	uuuu u0uu	*,*,*,*,*,*,*,*,*,*
049h	TB1CN1	PA1IV	PWMA1[2:0]			PA0IV	PWMA0[2:0]			0000 0000	uuuu uuuu
04Ah	TB1RH	TimerB1 counter Register [15:8]								xxxx xxxx	uuuu uuuu	r,r,r,f,r,f,r,f,r,f
04Bh	TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu	r,r,r,f,r,f,r,f,r,f
04Ch	TB1COH	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
04Dh	TB1COL	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
04Eh	TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
04Fh	TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
050h	TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu
051h	TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu
052h	TC1CN0	-	TC1S[1:0]		-	-	-	-	-	0000 0000	uuuu uuuu	uuuu uuuu
053h	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	xxxx xxxx
054h	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4	TC1.3	TC1.2	TC1.1	TC1.0	0000 0000	uuuu uuuu
055h	PT1DA	DA1.7	DA1.6	DA1.5	DA1.4	DA1.3	DA1.2	DA1.1	DA1.0	1111 1111	uuuu uuuu
056h	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	uuuu uuuu
057h	PT1M1	-	-	-	-	INTEG1[1:0]		INTEG0[1:0]		0000 0000	uuuu uuuu
058h	PT1INT	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	-	-	0000 0000	uuuu uuuu
059h	PT1INTE	INTE1.7	INTE1.6	INTE1.5	INTE1.4	-	-	-	-	0000 0000	uuuu uuuu
05Ah	PT1INTF	INTF1.7	INTF1.6	INTF1.5	INTF1.4	-	-	-	-	0000 0000	uuuu uuuu
05Bh	PT2	-	-	-	-	PT2.3	PT2.2	PT2.1	PT2.0	0000 0000	uuuu uuuu
05Ch	TRISC2	-	-	-	-	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	uuuu uuuu
05Dh	PT2DA	-	-	-	-	DA2.3	DA2.2	DA2.1	DA2.0	0000 1111	uuuu uuuu
05Eh	PT2PU	-	-	-	-	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	uuuu uuuu
05Fh	PT2INT	-	-	-	-	INTG2.3	INTG2.2	INTG2.1	INTG2.0	0000 0000	uuuu uuuu
060h	PT2INTE	-	-	-	-	INTE2.3	INTE2.2	INTE2.1	INTE2.0	0000 0000	uuuu uuuu
061h	PT2INTF	-	-	-	-	INTF2.3	INTF2.2	INTF2.1	INTF2.0	0000 0000	uuuu uuuu
062h	PT6	PT6.7	PT6.6	PT6.5	PT6.4	PT6.3	PT6.2	PT6.1	PT6.0	xxxx xxxx	uuuu uuuu
063h	TRISC6	TC6.7	TC6.6	TC6.5	TC6.4	TC6.3	TC6.2	TC6.1	TC6.0	0000 0000	uuuu uuuu
064h	PT6DA	DA6.7	DA6.6	DA6.5	DA6.4	DA6.3	DA6.2	DA6.1	DA6.0	1111 1111	uuuu uuuu
065h	PT6PU	PU6.7	PU6.6	PU6.5	PU6.4	PU6.3	PU6.2	PU6.1	PU6.0	0000 0000	uuuu uuuu
066h	PT7	PT7.7	PT7.6	PT7.5	PT7.4	PT7.3	PT7.2	PT7.1	PT7.0	xxxx xxxx	uuuu uuuu
067h	TRISC7	TC7.7	TC7.6	TC7.5	TC7.4	TC7.3	TC7.2	TC7.1	TC7.0	0000 0000	uuuu uuuu
068h	PT7DA	DA7.7	DA7.6	DA7.5	DA7.4	DA7.3	DA7.2	DA7.1	DA7.0	1111 1111	uuuu uuuu
069h	PT7PU	PU7.7	PU7.6	PU7.5	PU7.4	PU7.3	PU7.2	PU7.1	PU7.0	0000 0000	uuuu uuuu
06Ah	PT8	PT8.7	PT8.6	PT8.5	PT8.4	PT8.3	PT8.2	PT8.1	PT8.0	xxxx xxxx	uuuu uuuu
06Bh	TRISC8	TC8.7	TC8.6	TC8.5	TC8.4	TC8.3	TC8.2	TC8.1	TC8.0	0000 0000	uuuu uuuu
06Ch	PT8DA	DA8.7	DA8.6	DA8.5	DA8.4	DA8.3	DA8.2	DA8.1	DA8.0	1111 1111	uuuu uuuu
06Dh	PT8PU	PU8.7	PU8.6	PU8.5	PU8.4	PU8.3	PU8.2	PU8.1	PU8.0	0000 0000	uuuu uuuu

表 2-6 数据存储寄存器列表(续)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W	
06Eh	SSPCN0	ENSSP	CKP	CKE	SMP	-	-	SSPM[1:0]		0000 ..00	uuuu ..uu	***r,*,w*	
06Fh	SSPSTA0	SSPBY	SSPOV	-	-	-	-	-	BF	00.. ...0	uu.. ...u	*r,*,w*	
070h	SSPBUF0	SSP Receive/Transmit Buffer Register								xxxx xxxx	uuuu uuuu	*****	
071h	CFG0	-	-	-	-	-	GCRst	ENI2CT	ENI2C	0000 0000uuu	*,w*,r*	
072h	ACT0	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu	*****	
073h	STA0	MACTF	SACTF	RDBF	RWF	DF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*****	
074h	CRG0	CRG[7:0]								0000 0000	uuuu uuuu	*****	
075h	TOC0	I2CTF	I2C[2:0]			I2CTL[3:0]				0000 0000	uuuu uuuu	*****	
076h	RDB0	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*****	
077h	TDB0	TDB[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu	*****	
078h	SID0	SID0[7:1],The corresponding address of the 7-bit mode								SID0V[0]	0000 0000	uuuu uuuu	*****
079h	UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	***r,*,w*	
07Ah	UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	*,r,r,r,r,r,r,rw 0	
07Bh	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD 0000 uuuu	*,w*,r*	
07Ch	BG0RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	*,w*,r*	
07Dh	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*****	
07Eh	TX0R	UART Transmit Register								xxxx xxxx	uuuu uuuu	*****	
07Fh	RC0REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r	
180h	LCDCN1	ENLCP	LCDV[2:0]			ENLB	SELPCLK	-	LDCPU	0000 00.0	uuuu uu.u	*****	
181h	LCDCN2	-	-	-	-	-	-	LDCBL	LCI	0000 00..	uuuu uu..	*****,-	
182h	LCDCN3	SCM3[1:0]		SCM2[1:0]		SCM1[1:0]		SCM0[1:0]		0000 0000	uuuu uuuu	*****	
183h	LCDCN4	SSG21	SSG20	SSG19	SSG18	SSG17	SSG16	SSG15	SSG14	0000 0000	uuuu uuuu	*****	
185h	LCDCN6	SSG5[1:0]		SSG4[1:0]		SSG3[1:0]		SSG2[1:0]		0000 0000	uuuu uuuu	*****	
186h	LCDCN7	SSG9[1:0]		SSG8[1:0]		SSG7[1:0]		SSG6[1:0]		0000 0000	uuuu uuuu	*****	
187h	LCDCN8	SSG13[1:0]		SSG12[1:0]		SSG11[1:0]		SSG10[1:0]		0000 0000	uuuu uuuu	*****	
188h	LCD0	LCD SEG3[4:7] data				LCD SEG2[3:0] data				xxxx xxxx	uuuu uuuu	*****	
189h	LCD1	LCD SEG5[4:7] data				LCD SEG4[3:0] data				xxxx xxxx	uuuu uuuu	*****	
18Ah	LCD2	LCD SEG7[4:7] data				LCD SEG6[3:0] data				xxxx xxxx	uuuu uuuu	*****	
18Bh	LCD3	LCD SEG9[4:7] data				LCD SEG8[3:0] data				xxxx xxxx	uuuu uuuu	*****	
18Ch	LCD4	LCD SEG11[4:7] data				LCD SEG10[3:0] data				xxxx xxxx	uuuu uuuu	*****	
18Dh	LCD5	LCD SEG13[4:7] data				LCD SEG12[3:0] data				xxxx xxxx	uuuu uuuu	*****	
18Eh	LCD6	LCD SEG15[4:7] data				LCD SEG14[3:0] data				xxxx xxxx	uuuu uuuu	*****	
18Fh	LCD7	LCD SEG17[4:7] data				LCD SEG16[3:0] data				xxxx xxxx	uuuu uuuu	*****	
190h	LCD8	LCD SEG19[4:7] data				LCD SEG18[3:0] data				xxxx xxxx	uuuu uuuu	*****	
191h	LCD9	LCD SEG21[4:7] data				LCD SEG20[3:0] data				xxxx xxxx	uuuu uuuu	*****	
192h	UR2CN	ENSP2	ENTX2	TX92	TX9D2	PARITY2	-	-	WUE2	0000 0..0	uuuu u..u	***r,*,w*	
193h	UR2STA	-	RC9D2	PERR2	FERR2	OERR2	RCIDL2	TRMT2	ABDOVF2	.000 0010	.uuu uuuu	*,r,r,r,r,r,r,rw 0	
194h	BA2CN	-	-	-	-	ENCR2	RC92	ENADD2	ENABD2 0000 uuuu	*,w*,r*	
195h	BG2RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	*,w*,r*	
196h	BG2RL	Baud Rate2 Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*****	
197h	TX2R	UART2 Transmit Register								xxxx xxxx	uuuu uuuu	*****	
198h	RC2REG	UART2 Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r	
080h ~ 0FFh	SRAM as 128Byte									uuuu uuuu	uuuu uuuu	*****	
100h ~ 17Fh	SRAM as 128Byte									uuuu uuuu	uuuu uuuu	*****	
200h ~ 2FFh	SRAM as 256Byte									uuuu uuuu	uuuu uuuu	*****	

表 2-7 数据存储寄存器列表(续)

HY17S58 Emulate Chip User' Guide

Embedded ΣADC
8-Bit RISC-like Mixed Signal Microcontroller



2.2.3.3. HY17P58 数据存储寄存器

00Ch	PODEC2	Contents of FSR2 to address data memoryvalue of FSR2 post-decremented								xxxx xxxx	uuuu uuuu	*****
00Dh	PRINC2	Contents of FSR2 to address data memoryvalue of FSR2 pre-incremented								xxxx xxxx	uuuu uuuu	*****
00Eh	PLUSW2	Contents of FSR2 to address data memoryvalue of FSR2 offset by W								xxxx xxxx	uuuu uuuu	*****
00Fh	FSR0H	-	-	-	-	-	-	-	FSR0[9:8] xxxx uuuu	~*~*~*~*~*
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu	*****
011h	FSR1H	-	-	-	-	-	-	-	FSR1[9:8]xxxuuu	~*~*~*~*~*
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu	*****
013h	FSR2H	-	-	-	-	-	-	-	FSR2[9:8]xxxuuu	~*~*~*~*~*
014h	FSR2L	Indirect Data Memory Address Pointer 0 Low Byte,FSR2[7:0]								xxxx xxxx	uuuu uuuu	*****
016h	TOSH	-	-	-	TOS[12:8]				.xxx xxxx	.uuu uuuu	~*~*~*~*~*	
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu	*****
018h	SKCN	SKFL	SKUN	SKOV	-	SKPRT[3:0]			000. 0000	u\$. \$.\$\$\$	rw 0,rw 0,rw 0,- *~*~*~*	
01Ah	PCLATH	-	-	-	PC[12:8]				.00 0000	.00 0000	*****	
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000	*****
01Dh	TBLPTRH	-	-	-	TBLPTR[12:8]				.xx xxxx	.uu uuuu	~*~*~*~*~*	
01Eh	TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								xxxx xxxx	uuuu uuuu	*****
01Fh	TBLDH	Program Memory Table Latch High Byte								xxxx xxxx	uuuu uuuu	*****
020h	TBLDL	Program Memory Table Latch Low Byte								xxxx xxxx	uuuu uuuu	*****
021h	PRODH	Product Register of Multiply High Byte								xxxx xxxx	uuuu uuuu	*****
022h	PRODL	Product Register of Multiply Low Byte								xxxx xxxx	uuuu uuuu	*****
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE	-	E1IE	E0IE	0000 0000	0uuu uuuu	*****
024h	INTE1	TA1IE	SPIE	TXIE	RCIE	I2CERIE	I2CIE	E3IE	E2IE	0000 0000	uuuu uuuu	*****
025h	INTE2	TA2IE	TA2CIE	TC11IE	TC10IE	TX2IE	RC2IE	-	BOR2IE	0000 0000	uuuu uuuu	*****
026h	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	-	E1IF	E0IF	.000 0000	.uuu uuuu	*****
027h	INTF1	TA1IF	SPIF	TXIF	RCIF	I2CERIF	I2CIF	E3IF	E2IF	0000 0000	uuuu uuuu	*~*~*~*~*~*
028h	INTF2	TA2IF	TA2CIF	TC1IF	TC0IF	TX2IF	RC2IF	-	BOR2IF	0000 0000	uuuu uuuu	*****
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu	*****
02Ah	BSRCN	-	-	-	-	-	-	BSR[1:0]	 xxxx uuuu	~*~*~*~*~*
02Bh	MSTAT	-	-	-	C	DC	N	OV	Z	...x xxxx	...u uuuu	~*~*~*~*~*
02Ch	PSTAT	BOR	PD	TO	IDL	RST	SKERR	BOR2LV	GCRstIF	\$000 \$000	uu\$u u\$uu	rw0.rw0,rw0.rw0,rw0,rw0
031h	BIECN	1			ENBVD	VPPHV	ENBCP	BIEWR	BIERD	1.00 \$000	1.00 \$uuu	r1,-,*~*~*~*
032h	BIEARH	-		1	1	1	1	1	1	0.xx xxxx	u.uu uuuu	*~*~*~*~*~*
033h	BIEARL	BIE Address Register as BIEAL[5:0]								xxxx xxxx	uuuu uuuu	*****
034h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	*****
035h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	*****
036h	PWRCN	ENBGR	LDOC[2:0]			LDOM[1:0]		ENLDO	CSFON	0000 0000	uuuu u00u	*~*~*~*~*~*,w r0,w r0,*
037h	OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]		CUPS	0000 0000	uuuu uuuu	*****	
038h	OSCCN1	CCOPT	LCPS	DADC[1:0]		DTMB[1:0]		TMBS	LCDS	0000 0000	uuuu .uuu.	~*~*~*~*~*~*
039h	OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO	0000 0001	uuuu uu01	*****
03Ah	CSFCN0	SKRST	HAOTR[6:0]						.1.	~*~*~*~*~*~*	
03Bh	CSFCN1	ENSDRV	-	-	BOR_TH[2:0]		BORS	ENBOR2	.0 0011	uuuu uuuu	~*~*~*~*~*~*	
03Eh	WDTCN	ENBZ	BZS	DBZ[1:0]		ENWDT	DWDJT[2:0]		0000 0000	uuuu \$000	-*~*~*~*~*~*~*~*	

表 2-8 数据存储寄存器列表

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1
“\$”for event status,“-”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
03Fh	AD1H	ADC1 conversion high byte data register								..00 0000	..uu uuuu	..*.*.*.*.*.*.*.*
040h	AD1M	ADC1 conversion middle byte data register								0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
041h	AD1L	ADC1 conversion low byte data register								0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
042h	AD1CN0	ENAD1	-	OSR[3:0]			CMFR			000. 0000	uuu. uuuu	*.*.*.*.*.*.*.*
043h	AD1CN1	-	-	VREGN	-	ADGN[2:0]			xxxx xxxx	uuuu uuuu	*.*.*.*.*.*.*.*	
044h	AD1CN2	INIS1	-	-	-	DCSET[3:0]			xxxx xxxx	uuuu uuuu	*.*.*.*.*.*.*.*	
045h	AD1CN3	INP[3:0]			INN[3:0]			xxxx xxxx			uuuu uuuu	*.*.*.*.*.*.*.*
046h	AD1CN4	VRH[1:0]	VRL[1:0]		INX[1:0]		VRIS	INIS	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*	
047h	AD1CN5	ENACM	ENV12	VCMS	LDOPL	-	-	ENTPS	TPSCH	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
048h	LVDCN	-	-	PWRS	LVDS[3:0]			LVDO	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*	
049h	BIACN0	ENPK	ENPKS	ENPKD	PKRST	ENLPF	LPFS[1:0]	ENREFO	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*	
04Ah	DACCN0	-	DANS[2:0]			DAPS[3:0]			0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*	
04Bh	DACCN1	-	-	-	DADCS	DALH	DAOE[1:0]	ENDA	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*	
04Ch	DACCN2	DABIT[7:0]								0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
04Dh	OP1CN0							OP1OS[1:0]	ENOP1	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
04Eh	OP1NET	-	OP1PS[2:0]			-	OP1NS[2:0]			0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
04Fh	OP2CN0							OP2OS[1:0]	ENOP2	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
050h	OP2NET	OP2PS[3:0]			-	OP2NS[2:0]			0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*	
051h	PGACN0	ENCHP	CHM[1:0]		ENHS	-	ENPGA[1:0]			0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
052h	PGACN1	-	-	-	GAINS[1:0]		OPDIEN	ENOPPGA[1:0]		0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
053h	PGANET1	LNOP1NS[7:0]								0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
054h	PGANET2	LNOP1PS[7:0]								0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
055h	PGANET3	LNOP2NS[7:0]								0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
056h	PGANET4	LNOP2PS[7:0]								0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
057h	PGANET5	LNOP2OS[2:0]			OPCS		LNOP1OS[2:0]			0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
058h	TMA1CN	ENTMA1	TMACL1	TMAS1	DTMA1[2:0]			-	-	0000 00.0	u0uu uu.u	*.*.*.*.*.*.*.*
059h	TMA1R	TMA1 counter Register								0000 0000	uuuu uuuu	r0,r0,r0,r0,r0,r0,r0,r0
05Ah	TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	r0,r0,r0,r0,r0,r0,r0,r0
05Bh	TB1Flag	-	PWM7A	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	..00 0000	..uu uuuu	..*.*.*.*.*.*.*.*
05Ch	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	PWMO1	PWMO0	0000 0000	uuuu u0uu	*.*.*.*.*.*.*.*
05Dh	TB1CN1	PA1IV	PWMA1[2:0]			PA0IV	PWMA0[2:0]			0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
05Eh	TB1RH	TimerB1 counter Register [15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r
05Fh	TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r
060h	TB1C0H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*.*.*.*.*.*.*.*
061h	TB1C0L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*.*.*.*.*.*.*.*
062h	TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*.*.*.*.*.*.*.*
063h	TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*.*.*.*.*.*.*.*
064h	TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*.*.*.*.*.*.*.*
065h	TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*.*.*.*.*.*.*.*
066h	TC1CN0	-	TC1S[1:0]		-	-	CPI1P	CPI0P	TCEN	0000 0000	uuuu uuuu	uuuu uuuu
067h	TC1CN1	CPI1R	CPSS	CPI1S[1:0]		CPI1PS[3:0]			0000 0000	uuuu uuuu	uuuu uuuu	
068h	TC1R0H	Capture 0 High Byte Data Register								xxxx xxxx	uuuu uuuu	uuuu uuuu
069h	TC1R0L	Capture 0 Low Byte Data Register								xxxx xxxx	uuuu uuuu	uuuu uuuu
06Ah	TC1R1H	Capture 1 High Byte Data Register								xxxx xxxx	uuuu uuuu	uuuu uuuu
06Bh	TC1R1L	Capture 1 Low Byte Data Register								xxxx xxxx	uuuu uuuu	uuuu uuuu
06Ch	PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	xxxx xxxx	*.*.*.*.*.*.*.*
06Dh	TRISC1	TC1.7	TC1.6	TC1.5	TC1.4	TC1.3	TC1.2	TC1.1	TC1.0	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
06Eh	PT1DA	DA1.7	DA1.6	DA1.5	DA1.4	DA1.3	DA1.2	DA1.1	DA1.0	1111 1111	uuuu uuuu	*.*.*.*.*.*.*.*
06Fh	PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
070h	PT1M1	-	-	-	-	INTEG1[1:0]		INTEG0[1:0]		0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
071h	PT1M2	-	PM1.7[0]	-	PM1.6[0]	-	PM1.5[0]	-	PM1.4[0]	xxxx xxxx	xxxx xxxx	*.*.*.*.*.*.*.*
072h	PT1INT	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	-	-	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
073h	PT1INTE	INTE1.7	INTE1.6	INTE1.5	INTE1.4	-	-	-	-	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*
074h	PT1INTF	INTF1.7	INTF1.6	INTF1.5	INTF1.4	-	-	-	-	0000 0000	uuuu uuuu	*.*.*.*.*.*.*.*

表 2-9 数据存储寄存器列表(续)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1
“\$”for event status, “-”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
075h	PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	0000 0000	uuuu uuuu	*****
076h	TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	uuuu uuuu	*****
077h	PT2DA	DA2.7	DA2.6	DA2.5	DA2.4	DA2.3	DA2.2	DA2.1	DA2.0	1111 1111	uuuu uuuu	*****
078h	PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	uuuu uuuu	*****
07Bh	PT2INT	INTG2.7	INTG2.6	INTG2.5	INTG2.4	INTG2.3	INTG2.2	INTG2.1	INTG2.0	0000 0000	uuuu uuuu	*****
07Ch	PT2INTE	INTE2.7	INTE2.6	INTE2.5	INTE2.4	INTE2.3	INTE2.2	INTE2.1	INTE2.0	0000 0000	uuuu uuuu	*****
07Dh	PT2INTF	INTF2.7	INTF2.6	INTF2.5	INTF2.4	INTF2.3	INTF2.2	INTF2.1	INTF2.0	0000 0000	uuuu uuuu	*****
180h	LCDCN1	ENLCP	LCDV[2:0]			ENLB	SELPClk	-	LCDPU	0000 00.0	uuuu uu.u	*****
181h	LCDCN2						LCDTYPE	LCDBL		0000 0000	uuuu uuuu	*****
182h	LCDCN3	SCM3[1:0]		SCM2[1:0]		SCM1[1:0]		SCM0[1:0]		0000 0000	uuuu uuuu	*****
183h	LCDCN4	SSG21	SSG20	SSG19	SSG18	SSG17	SSG16	SSG15	SSG14	0000 0000	uuuu uuuu	*****
184h	LCDCN5	SSG41	SSG40	SSG39	SSG38	SSG37	SSG36	SSG35	SSG34	0000 0000	uuuu uuuu	*****
185h	LCDCN6	SSG5[1:0]		SSG4[1:0]		SSG3[1:0]		SSG2[1:0]		0000 0000	uuuu uuuu	*****
186h	LCDCN7	SSG9[1:0]		SSG8[1:0]		SSG7[1:0]		SSG6[1:0]		0000 0000	uuuu uuuu	*****
187h	LCDCN8	SSG13[1:0]		SSG12[1:0]		SSG11[1:0]		SSG10[1:0]		0000 0000	uuuu uuuu	*****
188h	LCDCN9	SSG25[1:0]		SSG24[1:0]		SSG23[1:0]		SSG22[1:0]		0000 0000	uuuu uuuu	*****
189h	LCDCN10	SSG29[1:1]		SSG28[1:1]		SSG27[1:1]		SSG26[1:1]		0000 0000	uuuu uuuu	*****
18Ah	LCDCN11	SSG33[1:1]		SSG32[1:1]		SSG31[1:1]		SSG30[1:1]		0000 0000	uuuu uuuu	*****
18Bh	LCD0	LCD SEG3[4:7] data				LCD SEG2[3:0] data				xxxx xxxx	uuuu uuuu	*****
18Ch	LCD1	LCD SEG5[4:7] data				LCD SEG4[3:0] data				xxxx xxxx	uuuu uuuu	*****
18Dh	LCD2	LCD SEG7[7:4] data				LCD SEG6[3:0] data				xxxx xxxx	uuuu uuuu	*****
18Eh	LCD3	LCD SEG9[7:4] data				LCD SEG8[3:0] data				xxxx xxxx	uuuu uuuu	*****
18Fh	LCD4	LCD SEG11[7:4] data				LCD SEG10[3:0] data				xxxx xxxx	uuuu uuuu	*****
190h	LCD5	LCD SEG13[7:4] data				LCD SEG12[3:0] data				xxxx xxxx	uuuu uuuu	*****
191h	LCD6	LCD SEG15[4:7] data				LCD SEG14[3:0] data				xxxx xxxx	uuuu uuuu	*****
192h	LCD7	LCD SEG17[4:7] data				LCD SEG16[3:0] data				xxxx xxxx	uuuu uuuu	*****
193h	LCD8	LCD SEG19[4:7] data				LCD SEG18[3:0] data				xxxx xxxx	uuuu uuuu	*****
194h	LCD9	LCD SEG21[4:7] data				LCD SEG20[3:0] data				xxxx xxxx	uuuu uuuu	*****
195h	LCD10	LCD SEG23[4:7] data				LCD SEG22[3:0] data				xxxx xxxx	uuuu uuuu	*****
196h	LCD11	LCD SEG25[4:7] data				LCD SEG24[3:0] data				xxxx xxxx	uuuu uuuu	*****
197h	LCD12	LCD SEG27[4:7] data				LCD SEG26[3:0] data				xxxx xxxx	uuuu uuuu	*****
198h	LCD13	LCD SEG29[4:7] data				LCD SEG28[3:0] data				xxxx xxxx	uuuu uuuu	*****
199h	LCD14	LCD SEG31[4:7] data				LCD SEG30[3:0] data				xxxx xxxx	uuuu uuuu	*****
19Ah	LCD15	LCD SEG33[4:7] data				LCD SEG32[3:0] data				xxxx xxxx	uuuu uuuu	*****
19Bh	LCD16	LCD SEG35[4:7] data				LCD SEG34[3:0] data				xxxx xxxx	uuuu uuuu	*****
19Ch	LCD17	LCD SEG37[4:7] data				LCD SEG36[3:0] data				xxxx xxxx	uuuu uuuu	*****
19Dh	LCD18	LCD SEG39[4:7] data				LCD SEG38[3:0] data				xxxx xxxx	uuuu uuuu	*****
19Eh	LCD19	LCD SEG41[4:7] data				LCD SEG40[3:0] data				xxxx xxxx	uuuu uuuu	*****
19Fh	PT4	PT4.7	PT4.6	PT4.5	PT4.4	-	-	-	-	xxxx xxxx	xxxx xxxx	*****
1A1h	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	-	-	-	-	1111 0000	uuuu uuuu	*****
1A2h	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
1A5h	PT4INT	INTG4.7	INTG4.6	INTG4.5	INTG4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
1A6h	PT4INTE	INTE4.7	INTE4.6	INTE4.5	INTE4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
1A7h	PT4INTF	INTF4.7	INTF4.6	INTF4.5	INTF4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
1AAh	PT5	-	-	-	-	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	xxxx xxxx	*****
1ACh	PT5DA	-	-	-	-	DA5.3	DA5.2	DA5.1	DA5.0	0000 1111	uuuu uuuu	*****
1ADh	PT5PU	-	-	-	-	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	uuuu uuuu	*****
1B0h	PT5INT	-	-	-	-	INTG5.3	INTG5.2	INTG5.1	INTG5.0	0000 0000	uuuu uuuu	*****
1B1h	PT5INTE	-	-	-	-	INTE5.3	INTE5.2	INTE5.1	INTE5.0	0000 0000	uuuu uuuu	*****
1B2h	PT5INTF	-	-	-	-	INTF5.3	INTF5.2	INTF5.1	INTF5.0	0000 0000	uuuu uuuu	*****
1B5h	PT6	PT6.7	PT6.6	PT6.5	PT6.4	PT6.3	PT6.2	PT6.1	PT6.0	xxxx xxxx	uuuu uuuu	*****
1B6h	TRISC6	TC6.7	TC6.6	TC6.5	TC6.4	TC6.3	TC6.2	TC6.1	TC6.0	0000 0000	uuuu uuuu	*****
1B7h	PT6DA	DA6.7	DA6.6	DA6.5	DA6.4	DA6.3	DA6.2	DA6.1	DA6.0	1111 1111	uuuu uuuu	*****
1B8h	PT6PU	PU6.7	PU6.6	PU6.5	PU6.4	PU6.3	PU6.2	PU6.1	PU6.0	0000 0000	uuuu uuuu	*****

表 2-10 数据存储寄存器列表(续)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1

“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W		
1B9h	PT7	PT7.7	PT7.6	PT7.5	PT7.4	PT7.3	PT7.2	PT7.1	PT7.0	xxxx xxxx	uuuu uuuu	*****		
1BAh	TRISC7	TC7.7	TC7.6	TC7.5	TC7.4	TC7.3	TC7.2	TC7.1	TC7.0	0000 0000	uuuu uuuu	*****		
1BBh	PT7DA	DA7.7	DA7.6	DA7.5	DA7.4	DA7.3	DA7.2	DA7.1	DA7.0	1111 1111	uuuu uuuu	*****		
1BCh	PT7PU	PU7.7	PU7.6	PU7.5	PU7.4	PU7.3	PU7.2	PU7.1	PU7.0	0000 0000	uuuu uuuu	*****		
1BDh	PT8	PT8.7	PT8.6	PT8.5	PT8.4	PT8.3	PT8.2	PT8.1	PT8.0	xxxx xxxx	uuuu uuuu	*****		
1BEh	TRISC8	TC8.7	TC8.6	TC8.5	TC8.4	TC8.3	TC8.2	TC8.1	TC8.0	0000 0000	uuuu uuuu	*****		
1BFh	PT8DA	DA8.7	DA8.6	DA8.5	DA8.4	DA8.3	DA8.2	DA8.1	DA8.0	1111 1111	uuuu uuuu	*****		
1C0h	PT8PU	PU8.7	PU8.6	PU8.5	PU8.4	PU8.3	PU8.2	PU8.1	PU8.0	0000 0000	uuuu uuuu	*****		
1C1h	PT9	PT9.7	PT9.6	PT9.5	PT9.4	PT9.3	PT9.2	PT9.1	PT9.0	xxxx xxxx	uuuu uuuu	*****		
1C2h	TRISC9	TC9.7	TC9.6	TC9.5	TC9.4	TC9.3	TC9.2	TC9.1	TC9.0	0000 0000	uuuu uuuu	*****		
1C3h	PT9DA	DA9.7	DA9.6	DA9.5	DA9.4	DA9.3	DA9.2	DA9.1	DA9.0	1111 1111	uuuu uuuu	*****		
1C4h	PT9PU	PU9.7	PU9.6	PU9.5	PU9.4	PU9.3	PU9.2	PU9.1	PU9.0	0000 0000	uuuu uuuu	*****		
1C5h	PT10	PT10.7	PT10.6	PT10.5	PT10.4	PT10.3	PT10.2	PT10.1	PT10.0	xxxx xxxx	uuuu uuuu	*****		
1C6h	TRISC10	TC10.7	TC10.6	TC10.5	TC10.4	TC10.3	TC10.2	TC10.1	TC10.0	0000 0000	uuuu uuuu	*****		
1C7h	PT10DA	DA10.7	DA10.6	DA10.5	DA10.4	DA10.3	DA10.2	DA10.1	DA10.0	1111 1111	uuuu uuuu	*****		
1C8h	PT10PU	PU10.7	PU10.6	PU10.5	PU10.4	PU10.3	PU10.2	PU10.1	PU10.0	0000 0000	uuuu uuuu	*****		
1C9h	PT11	-	-	-	-	PT11.3	PT11.2	PT11.1	PT11.0	xxxx xxxx	uuuu uuuu	*****		
1CAh	TRISC11	-	-	-	-	TC11.3	TC11.2	TC11.1	TC11.0	0000 0000	uuuu uuuu	*****		
1CBh	PT11DA	-	-	-	-	DA11.3	DA11.2	DA11.1	DA11.0	0000 1111	uuuu uuuu	*****		
1CCh	PT11PU	-	-	-	-	PU11.3	PU11.2	PU11.1	PU11.0	0000 0000	uuuu uuuu	*****		
1CEh	SSPCN0	ENSSP	CKP	CKE	SMP	-	-	SSPM[1:0]		0000 ..00	uuuu ..uu	*****		
1CFh	SSPSTA0	SSPBY	SSPOV	-	-	-	-	-	BF	00.. ...0	uu.. ...u	*****		
1D0h	SSPBUF0	SSP Receive/Transmit Buffer Register								xxxx xxxx	uuuu uuuu	*****		
1D1h	CFG0	-	-	-	-	-	GCRst	ENI2CT	ENI2C	0000 0000 uuuu	*****		
1D2h	ACT0	SLAVE	-	-	I2CER	START	STOP	I2CINT		0000 0000	uuuu uuuu	*****		
1D3h	STA0	MACTF	SACTF	RDBF	RWF	DF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*****		
1D4h	CRG0	CRG[7:0]								0000 0000	uuuu uuuu	*****		
1D5h	TOC0	I2CTF	DI2C[2:0]			I2CTL[3:0]				0000 0000	uuuu uuuu	*****		
1D6h	RDB0	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*****		
1D7h	TDB0	TDB[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu	*****		
1D8h	SID0	SID0[7:1].The corresponding address of the 7-bit mode								SID0V[0]	0000 0000	uuuu uuuu	*****	
1D9h	UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	*****		
1DAh	UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	..00 0010	..uu uuuu	~.r.r.f.r.r.f.r.w0		
1DBh	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD	... 0000	... uuuu	~.r.r.f.r.r.f.r.w0		
1DCh	BG0RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	~.r.r.f.r.r.f.r.w0		
1DDh	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*****		
1DEh	TXOR	UART Transmit Register								xxxx xxxx	uuuu uuuu	*****		
1DFh	RC0REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r.r.f.r.r.f.r.f		
1E0h	UR2CN	ENSP2	ENTX2	TX92	TX9D2	PARITY2	-	-	WUE2	0000 0..0	uuuu u..u	*****		
1E1h	UR2STA	-	RC9D2	PERR2	FERR2	OERR2	RCIDL2	TRMT2	ABDOVF2	..00 0010	..uu uuuu	~.r.r.f.r.r.f.r.w0		
1E2h	BA2CN	-	-	-	-	ENCR2	RC92	ENADD2	ENABD2	... 0000	... uuuu	~.r.r.f.r.r.f.r.w0		
1E3h	BG2RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	~.r.r.f.r.r.f.r.w0		
1E4h	BG2RL	Baud Rate2 Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*****		
1E5h	TX2R	UART2 Transmit Register								xxxx xxxx	uuuu uuuu	*****		
1E6h	RC2REG	UART2 Receive Register								xxxx xxxx	uuuu uuuu	r.r.f.r.r.f.r.f		
1E7h	TMA2CN	ENTMA2	TMA2L2	TMA2S	DTMA2[2:0]			-	-	0000 00..	u0uu uu..	*.rw1.*.*.*.*		
1E8h	TMA2R	TMA2 counter Register								0000 0000	uuuu uuuu	rw0.rw0.rw0.rw0.rw0.rw0.rw0.rw0		
1E9h	TMA2C	TMA2C counter Register								0000 0000	uuuu uuuu	rw0.rw0.rw0.rw0.rw0.rw0.rw0.rw0		
1EAh	LEDCN			ENLEDP	VGGS	ENBLOUT	LEDS[2:0]			0000 0000	uuuu uuuu	*****		
1EBh	IQ0								ENIQ	0000 0000	uuuu uuuu	*****		
1ECh	IQ1	IQOffset[5:0]						IQMODE	IQINV	0000 0000	uuuu uuuu	*****		
1EDh	DGCON1				DGRST	DGDIV[2:0]			DGEN	0000 0000	uuuu uuuu	*****		
1EEh	DGCON2				DGRP[5:0]					0000 0000	uuuu uuuu	*****		
080h - 0FFh	SRAM as 128Byte											uuuu uuuu	uuuu uuuu	*****
100h - 17Fh	SRAM as 128Byte											uuuu uuuu	uuuu uuuu	*****
200h - 2FFh	SRAM as 256Byte											uuuu uuuu	uuuu uuuu	*****
300h - 33Fh	SRAM as 64Byte											uuuu uuuu	uuuu uuuu	*****

表 2-11 数据存储寄存器列表(续)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



2.2.3.4. HY17P48 数据存储器

“-”no use,“r”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1 “\$”for event status,“-”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition												
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
000h	INDF0	Contents of FSR0 to address data memoryvalue of FSR0 not changed								xxxx xxxx	uuuu uuuu	*****
001h	POINC0	Contents of FSR0 to address data memoryvalue of FSR0 post-increment								xxxx xxxx	uuuu uuuu	*****
002h	PODEC0	Contents of FSR0 to address data memoryvalue of FSR0 post-decrement								xxxx xxxx	uuuu uuuu	*****
003h	PRINC0	Contents of FSR0 to address data memoryvalue of FSR0 pre-increment								xxxx xxxx	uuuu uuuu	*****
004h	PLUSW0	Contents of FSR0 to address data memoryvalue of FSR0 offset by W								xxxx xxxx	uuuu uuuu	*****
005h	INDF1	Contents of FSR1 to address data memoryvalue of FSR1 not changed								xxxx xxxx	uuuu uuuu	*****
006h	POINC1	Contents of FSR1 to address data memoryvalue of FSR1 post-increment								xxxx xxxx	uuuu uuuu	*****
007h	PODEC1	Contents of FSR1 to address data memoryvalue of FSR1 post-decrement								xxxx xxxx	uuuu uuuu	*****
008h	PRINC1	Contents of FSR1 to address data memoryvalue of FSR1 pre-increment								xxxx xxxx	uuuu uuuu	*****
009h	PLUSW1	Contents of FSR1 to address data memoryvalue of FSR1 offset by W								xxxx xxxx	uuuu uuuu	*****
00Ah	INDF2	Contents of FSR2 to address data memoryvalue of FSR2 not changed								xxxx xxxx	uuuu uuuu	*****
00Bh	POINC2	Contents of FSR2 to address data memoryvalue of FSR2 post-increment								xxxx xxxx	uuuu uuuu	*****
00Ch	PODEC2	Contents of FSR2 to address data memoryvalue of FSR2 post-decrement								xxxx xxxx	uuuu uuuu	*****
00Dh	PRINC2	Contents of FSR2 to address data memoryvalue of FSR2 pre-increment								xxxx xxxx	uuuu uuuu	*****
00Eh	PLUSW2	Contents of FSR2 to address data memoryvalue of FSR2 offset by W								xxxx xxxx	uuuu uuuu	*****
00Fh	FSR0H	-	-	-	-	-	-	-	FSR0[9:8]	... xxxx	... uuuu	-,-,-,*****
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte,FSR0[7:0]								xxxx xxxx	uuuu uuuu	*****
011h	FSR1H	-	-	-	-	-	-	-	FSR1[9:8]xxxuuu	-,-,-,*****
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte,FSR1[7:0]								xxxx xxxx	uuuu uuuu	*****
013h	FSR2H	-	-	-	-	-	-	-	FSR2[9:8]xxxuuu	-,-,-,*****
014h	FSR2L	Indirect Data Memory Address Pointer 0 Low Byte,FSR2[7:0]								xxxx xxxx	uuuu uuuu	*****
016h	TOSH	-	-	-	TOS[12:8]				.xxx xxxx	.uuu uuuu	-,-,-,*****	
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu	*****
018h	SKCN	SKFL	SKUN	SKOV	-	SKPRT[3:0]			000. 0000	u\$\$ \$\$\$\$	rw0,rw0,rw0,-	*****
01Ah	PCLATH	-	-	-	PC[12:8]				.00 0000	.00 0000	*****	
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000	*****
01Dh	TBLPTRH	-	-	-	TBLPTR[12:8]				.xx xxxx	.uu uuuu	-,-,*****	
01Eh	TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								xxxx xxxx	uuuu uuuu	*****
01Fh	TBLDH	Program Memory Table Latch High Byte								xxxx xxxx	uuuu uuuu	*****
020h	TBLDL	Program Memory Table Latch Low Byte								xxxx xxxx	uuuu uuuu	*****
021h	PRODH	Product Register of Multiply High Byte								xxxx xxxx	uuuu uuuu	*****
022h	PRODL	Product Register of Multiply Low Byte								xxxx xxxx	uuuu uuuu	*****
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE	-	-	E0IE	0000 0000	0uuu uuuu	*****
024h	INTE1	TA1IE	SPIIE	TXIE	RCIE	I2CERIE	I2CIE	-	E2IE	0000 0000	uuuu uuuu	*****
025h	INTE2	TA2IE	TA2CIE	TC12IE	TC11IE	TX2IE	RC2IE	-	BOR2IE	0000 0000	uuuu uuuu	*****
026h	INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	-	-	E0IF	.000 0000	.uuu uuuu	*****
027h	INTF1	TA1IF	SPIIF	TXIF	RCIF	I2CERIF	I2CIF	-	E2IF	0000 0000	uuuu uuuu	*****
028h	INTF2	TA2IF	TA2CIF	TC2IF	TC1IF	TX2IF	RC2IF	-	BOR2IF	0000 0000	uuuu uuuu	*****
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu	*****
02Ah	BSRCN	-	-	-	-	-	-	-	BSR[1:0]xxuu	-,-,-,*****
02Bh	MSTAT	-	-	-	C	DC	N	OV	Z	...x xxxx	...u uuuu	-,-,-,*****
02Ch	PSTAT	BOR	PD	TO	IDL	RST	SKERR	BOR2LV	GCRstIF	\$000 \$000	uu\$u u\$uu	rw0,rw0,rw0,rw0 rw0,rw0,r,rw0
02Eh	INTE3	-	-	-	-	-	-	I2CER2IE	I2C2IE	0000 0000	uuuu uuuu	*****
030h	INTF3	-	-	-	-	-	-	I2CER2IF	I2C2IF	0000 0000	uuuu uuuu	*****
031h	BIECN	1	ENBVD			VPPHV	ENBCP	BIEWR	BIERD	1.00 \$000	1.00 \$uuu	r1,-,-,*,*,*,*,*
033h	BIEARL	BIE Address Register as BIEAL[5:0]								xxxx xxxx	uuuu uuuu	*****
034h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	*****
035h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	*****
036h	PWRCN	ENBGR	LDOC[2:0]			LDOM[1:0]	ENLDO	CSFON		1000 0000	uuuu uu0u	*****
037h	OSCCN0	OSCS[1:0]		DHS[1:0]	DMS[2:0]		CUPS		0000 0000	uuuu uuuu	*****	
038h	OSCCN1	CCOPT	LCPS	DADC[1:0]	DTMB[1:0]	TMBS	-		0000 0000	uuuu uu.	*****	
039h	OSCCN2	-	-	ENXT	XTS[1:0]	HAOM[1:0]	ENHAO		0000 0001	uuuu uu01	*****	

表 2-12 数据存储器列表

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



"r"no use,""read/write,"w"write,"r"read,"r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1
"\$"for event status,""unimplemented bit,""x"unknown,"u"unchanged,"d"depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W		
03Ah	CSFCN0	SKRST	HAOTR[6:0]							..1..	
03Bh	CSFCN1	ENSDRV	-	-	BOR_TH[2:0]		BORS	ENBOR2		...0 0011	uuuu uuuu		
03Eh	WDTCN	ENBZ	BZS	BZ[1:0]		ENWDT	DWDWT[2:0]			0000 0000	uuuu \$000		
03Fh	AD1H	ADC1 conversion high byte data register										..00 0000	..uu uuuu
040h	AD1M	ADC1 conversion middle byte data register										0000 0000	uuuu uuuu
041h	AD1L	ADC1 conversion low byte data register										0000 0000	uuuu uuuu
042h	AD1CN0	ENAD1	-	OSR[3:0]			CMFR			000. 0000	uuu. uuuu		
043h	AD1CN1	-	-	VREGN	PGAGN[1:0]		ADGN[2:0]			xxxx xxxx	uuuu uuuu		
044h	AD1CN2	INIS	-	-	-	DCSET[3:0]				xxxx xxxx	uuuu uuuu		
045h	AD1CN3	INP[3:0]			INN[3:0]					xxxx xxxx	uuuu uuuu		
046h	AD1CN4	VRH[1:0]		VRL[1:0]		INX[1:0]	VRIS	INIS		0000 0000	uuuu uuuu		
047h	AD1CN5	ENACM	-	VCMS	LDOPL	-	TPSCP	ENTPS	TPSCH	0000 0000	uuuu uuuu		
048h	LVDCN	-	-	PWRS	LVDS[3:0]			LVDO		0000 0000	uuuu uuuu		
049h	TMA1CN	ENTMA1	TMAA1	TMAA1	DTMA1[2:0]		-			0000 00.0	u0uu uu.u	*rw 1,*,*,*,*		
04Ah	TMA1R	TMA1 counter Register										0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0,rw0,rw0,rw0,rw0
04Bh	TMA1C	TMA1C counter Register										0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0,rw0,rw0,rw0,rw0
04Ch	TB1Flag	-	PWM7A	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	.000 0000	.uuu uuuu	-,r,r,r,r,r,r,r		
04Dh	TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	PWMO1	PWMO0	0000 0000	uuuu u0uu		
04Eh	TB1CN1	PA1IV	PWMA1[2:0]		PA0IV	PWMA0[2:0]				0000 0000	uuuu uuuu		
04Fh	TB1RH	TimerB1 counter Register [15:8]										xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r
050h	TB1RL	TimerB1 counter Register [7:0]										xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r
051h	TB1COH	TimerB1 counter Condition Register [15:8]										xxxx xxxx	uuuu uuuu
052h	TB1COL	TimerB1 counter Condition Register [7:0]										xxxx xxxx	uuuu uuuu
053h	TB1C1H	TimerB1 counter Condition Register [15:8]										xxxx xxxx	uuuu uuuu
054h	TB1C1L	TimerB1 counter Condition Register [7:0]										xxxx xxxx	uuuu uuuu
055h	TB1C2H	TimerB1 counter Condition Register [15:8]										xxxx xxxx	uuuu uuuu
056h	TB1C2L	TimerB1 counter Condition Register [7:0]										xxxx xxxx	uuuu uuuu
057h	TC1CN0	TC1S[1:0]			CP1P		CP1P	TCEN		0000 0000	uuuu uuuu	uuuu uuuu		
058h	TC1CN1	CP1R	CPSS	CP1S[1:0]		CP1PS[3:0]				0000 0000	uuuu uuuu	uuuu uuuu		
059h	TC1R0H	Capture 0 High Byte Data Register										xxxx xxxx	uuuu uuuu	uuuu uuuu
05Ah	TC1R0L	Capture 0 Low Byte Data Register										xxxx xxxx	uuuu uuuu	uuuu uuuu
05Bh	TC1R1H	Capture 1 High Byte Data Register										xxxx xxxx	uuuu uuuu	uuuu uuuu
05Ch	TC1R1L	Capture 1 Low Byte Data Register										xxxx xxxx	uuuu uuuu	uuuu uuuu
05Dh	PT1	-	-	-	-	-	PT1.2	-	PT1.0	xxxx xxxx	xxxx xxxx		
05Eh	TRISC1	-	-	-	-	-	TC1.2	-	TC1.0	0000 0000	uuuu uuuu		
05Fh	PT1DA	-	-	-	-	-	DA1.2	-	DA1.0	1111 1111	uuuu uuuu		
060h	PT1PU	-	-	-	-	-	PU1.2	-	PU1.0	0000 0000	uuuu uuuu		
061h	PT1M1	-	-	-	-	-	INTEG0[1:0]			0000 0000	uuuu uuuu		
063h	PT1INT	-	-	-	-	-	INTEG2	-	-	0000 0000	uuuu uuuu		
064h	PT2	-	-	-	-	PT2.3	PT2.2	PT2.1	PT2.0	0000 0000	uuuu uuuu		
065h	TRISC2	-	-	-	-	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	uuuu uuuu		
066h	PT2DA	-	-	-	-	DA2.3	DA2.2	DA2.1	DA2.0	1111 1111	uuuu uuuu		
067h	PT2PU	-	-	-	-	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	uuuu uuuu		
06Ah	PT2INT	-	-	-	-	INTG2.3	INTG2.2	INTG2.1	INTG2.0	0000 0000	uuuu uuuu		
06Bh	PT2INTE	-	-	-	-	INTE2.3	INTE2.2	INTE2.1	INTE2.0	0000 0000	uuuu uuuu		
06Ch	PT2INTF	-	-	-	-	INTF2.3	INTF2.2	INTF2.1	INTF2.0	0000 0000	uuuu uuuu		
06Dh	PT4	PT4.7	PT4.6	PT4.5	PT4.4	-	-	-	-	xxxx xxxx	xxxx xxxx		
06Fh	PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	-	-	-	-	1111 0000	uuuu uuuu		
070h	PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	-	-	-	-	0000 0000	uuuu uuuu		
073h	PT4INT	INTG4.7	INTG4.6	INTG4.5	INTG4.4	-	-	-	-	0000 0000	uuuu uuuu		
074h	PT4INTE	INTE4.7	INTE4.6	INTE4.5	INTE4.4	-	-	-	-	0000 0000	uuuu uuuu		
075h	PT4INTF	INTF4.7	INTF4.6	INTF4.5	INTF4.4	-	-	-	-	0000 0000	uuuu uuuu		
076h	PT12	PT12.7	PT12.6	PT12.5	PT12.4	PT12.3	PT12.2	PT12.1	PT12.0	xxxx xxxx	uuuu uuuu		
077h	TRISC12	TC12.7	TC12.6	TC12.5	TC12.4	TC12.3	TC12.2	TC12.1	TC12.0	0000 0000	uuuu uuuu		
078h	PT12DA	DA12.7	DA12.6	DA12.5	DA12.4	DA12.3	DA12.2	DA12.1	DA12.0	1111 1111	uuuu uuuu		
079h	PT12PU	PU12.7	PU12.6	PU12.5	PU12.4	PU12.3	PU12.2	PU12.1	PU12.0	0000 0000	uuuu uuuu		

表 2-13 数据存储寄存器列表

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1
“\$”for event status,“-”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W	
07Ah	TMA2CN	ENTMA2	TMA2	TMAS2	DTMA2[2:0]				-	-	0000 00..	u0uu uu..	*,rw1,*,*,*,*,-
07Bh	TMA2R	TMA2 counter Register								0000 0000	uuuu uuuu	w0,rw0,rw0,rw0,rw0,rw0,rw0,rw0	
07Ch	TMA2C	TMA2C counter Register								0000 0000	uuuu uuuu	w0,rw0,rw0,rw0,rw0,rw0,rw0,rw0	
07Dh	SSPCN0	ENSSP	CKP	CKE	SMP	-	-	SSPM[1:0]		0000 ..00	uuuu ..uu	****,*,*,*	
07Eh	SSPSTA0	SSPBY	SSPOV	-	-	-	-	-	BF	00.. ..0	uu.. ..u	*,*,*,*,*,*	
07Fh	SSPBUF0	SSP Receive/Transmit Buffer Register								xxxx xxxx	uuuu uuuu	*****	
180h	CFG0	-	-	-	-	-	GCRst	ENI2CT	ENI2C	0000 0000uuu	*,*,*,*,*,*	
181h	ACT0	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu	*****	
182h	STA0	MACTF	SACTF	RDBF	RWF	DF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*****	
183h	CRG0	CRG[7:0]								0000 0000	uuuu uuuu	*****	
184h	TOC0	I2CTF	DI2C[2:0]			I2CTL[3:0]				0000 0000	uuuu uuuu	*****	
185h	RDB0	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*****	
186h	TDB0	TDB[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu	*****	
187h	SID0	SID0[7:1],The corresponding address of the 7-bit mode							SID0V[0]	0000 0000	uuuu uuuu	*****	
188h	CFG2	-	-	-	-	-	GCRst	ENI2CT	ENI2C	0000 0000uuu	*,*,*,*,*,*	
189h	ACT2	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu	*****	
18Ah	STA2	MACTF	SACTF	RDBF	RWF	DF	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*****	
18Bh	CRG2	CRG[7:0]								0000 0000	uuuu uuuu	*****	
18Ch	TOC2	I2CTF	DI2C[2:0]			I2CTL[3:0]				0000 0000	uuuu uuuu	*****	
18Dh	RDB2	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*****	
18Eh	TDB2	TDB[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu	*****	
18Fh	SID2	SID0[7:1],The corresponding address of the 7-bit mode							SID0V[0]	0000 0000	uuuu uuuu	*****	
190h	UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	****,*,*,*	
191h	UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	-,r,r,r,r,r,r,w0	
192h	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD 0000 uuuu	*,*,*,*,*,*	
193h	BG0RH	-	-	-	Baud Rate Generator Register High Byte				...x xxxx	...u uuuu	*,*,*,*,*,*		
194h	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*****	
195h	TX0R	UART Transmit Register								xxxx xxxx	uuuu uuuu	*****	
196h	RC0REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r	
197h	UR2CN	ENSP2	ENTX2	TX92	TX9D2	PARITY2	-	-	WUE2	0000 0..0	uuuu u..u	****,*,*,*	
198h	UR2STA	-	RC9D2	PERR2	FERR2	OERR2	RCIDL2	TRMT2	ABDOVF2	.000 0010	.uuu uuuu	-,r,r,r,r,r,r,w0	
199h	BA2CN	-	-	-	-	ENCR2	RC92	ENADD2	ENABD2 0000 uuuu	*,*,*,*,*,*	
19Ah	BG2RH	-	-	-	Baud Rate Generator Register High Byte				...x xxxx	...u uuuu	*,*,*,*,*,*		
19Bh	BG2RL	Baud Rate2 Generator Register Low Byte								xxxx xxxx	uuuu uuuu	*****	
19Ch	TX2R	UART2 Transmit Register								xxxx xxxx	uuuu uuuu	*****	
19Dh	RC2REG	UART2 Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r	
19Eh	CCNT	-	-	-	-	CCLevel[2:0]				xxxx x000	uuuu uuuu	*****	
19Fh	ENCCMode	ENCC7	ENCC6	ENCC5	ENCC4	ENCC3	ENCC2	ENCC1	ENCC0	0000 0000	uuuu uuuu	*****	
1B0h	PT1AIE	-	-	-	-	-	AIE12	-	-	uuuu u0u0	uuuu u0u0	*****	
1B3h	PT4AIE	AIE47	AIE46	AIE45	AIE44	-	-	-	-	0000 uuuu	0000 uuuu	*****	
080h ~ 0FFh	SRAM as 128Byte								uuuu uuuu	uuuu uuuu	*****		
100h ~ 17Fh	SRAM as 128Byte								uuuu uuuu	uuuu uuuu	*****		
200h ~ 2FFh	SRAM as 256Byte								uuuu uuuu	uuuu uuuu	*****		

表 2-14 数据存储寄存器列表

HY17S58 Emulate Chip User' Guide

Embedded ΣADC
8-Bit RISC-like Mixed Signal Microcontroller



2.2.3.5. HY17P51 数据存储器

"no use","read/write","write","read","r0"only read 0,"r1"only read 1,"w0"only write 0,"w1"only write 1 "\$"for event status, "."unimplemented bit, "x"unknown, "u"unchanged, "d"depends on condition												
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
000h	INDF0	Contents of FSR0 to address data memory value of FSR0 not changed								xxxx xxxx	uuuu uuuu	***** r1, r1
001h	POINC0	Contents of FSR0 to address data memory value of FSR0 post-incremented								xxxx xxxx	uuuu uuuu	***** r1, r1
002h	PODEC0	Contents of FSR0 to address data memory value of FSR0 post-decremented								xxxx xxxx	uuuu uuuu	***** r1, r1
003h	PRINC0	Contents of FSR0 to address data memory value of FSR0 pre-incremented								xxxx xxxx	uuuu uuuu	***** r1, r1
004h	PLUSW0	Contents of FSR0 to address data memory value of FSR0 offset by W								xxxx xxxx	uuuu uuuu	***** r1, r1
005h	INDF1	Contents of FSR1 to address data memory value of FSR1 not changed								xxxx xxxx	uuuu uuuu	***** r1, r1
006h	POINC1	Contents of FSR1 to address data memory value of FSR1 post-incremented								xxxx xxxx	uuuu uuuu	***** r1, r1
007h	PODEC1	Contents of FSR1 to address data memory value of FSR1 post-decremented								xxxx xxxx	uuuu uuuu	***** r1, r1
008h	PRINC1	Contents of FSR1 to address data memory value of FSR1 pre-incremented								xxxx xxxx	uuuu uuuu	***** r1, r1
009h	PLUSW1	Contents of FSR1 to address data memory value of FSR1 offset by W								xxxx xxxx	uuuu uuuu	***** r1, r1
00Ah	INDF2	Contents of FSR2 to address data memory value of FSR2 not changed								xxxx xxxx	uuuu uuuu	***** r1, r1
00Bh	POINC2	Contents of FSR2 to address data memory value of FSR2 post-incremented								xxxx xxxx	uuuu uuuu	***** r1, r1
00Ch	PODEC2	Contents of FSR2 to address data memory value of FSR2 post-decremented								xxxx xxxx	uuuu uuuu	***** r1, r1
00Dh	PRINC2	Contents of FSR2 to address data memory value of FSR2 pre-incremented								xxxx xxxx	uuuu uuuu	***** r1, r1
00Eh	PLUSW2	Contents of FSR2 to address data memory value of FSR2 offset by W								xxxx xxxx	uuuu uuuu	***** r1, r1
00Fh	FSR0H	-	-	-	-	-	-	-	FSR0[8]xu	~r1,~r1,~r1,~r1*
010h	FSR0L	Indirect Data Memory Address Pointer 0 Low Byte, FSR0[7:0]								xxxx xxxx	uuuu uuuu	***** r1, r1
011h	FSR1H	-	-	-	-	-	-	-	FSR1[8]xu	~r1,~r1,~r1,~r1*
012h	FSR1L	Indirect Data Memory Address Pointer 0 Low Byte, FSR1[7:0]								xxxx xxxx	uuuu uuuu	***** r1, r1
016h	TOSH	-	-	-	-	TOS[11:8]		 xxxx uuuu	~r1,~r1,~r1,~r1*	
017h	TOSL	Top-of-Stack Low Byte (TOS<7:0>)								xxxx xxxx	uuuu uuuu	***** r1, r1
018h	SKCN	SKFL	SKUN	SKOV	-	SKPRT[3:0]			000. 0000	u\$\$ \$\$\$\$	rw 0, rw 0, rw 0, ~r1,~r1*	
01Ah	PCLATH	-	-	-	-	PC[11:8]		 0000 0000	~r1,~r1,~r1,~r1*	
01Bh	PCLATL	PC Low Byte for PC<7:0>								0000 0000	0000 0000	***** r1, r1
01Dh	TBLPTRH	-	-	-	-	TBLPTR[11:8]		 xxxx uuuu	~r1,~r1,~r1,~r1*	
01Eh	TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								xxxx xxxx	uuuu uuuu	***** r1, r1
01Fh	TBLDH	Program Memory Table Latch High Byte								xxxx xxxx	uuuu uuuu	***** r1, r1
020h	TBLDL	Program Memory Table Latch Low Byte								xxxx xxxx	uuuu uuuu	***** r1, r1
021h	PRODH	Product Register of Multiply High Byte								xxxx xxxx	uuuu uuuu	***** r1, r1
022h	PRODL	Product Register of Multiply Low Byte								xxxx xxxx	uuuu uuuu	***** r1, r1
023h	INTE0	GIE	TA1CIE	ADIE	WDTIE	-	-	E1IE	E0IE	0000 0000	0uuu uuuu	***** r1, r1
024h	INTE1	TA1IE	-	TXIE	RCIE	-	-	-	-	0000 0000	uuuu uuuu	***** r1, r1
026h	INTF0	-	TA1CIF	ADIF	WDTIF	-	-	E1IF	E0IF	.000 0000	.uuu uuuu	***** r1, r1
027h	INTF1	TA1IF	-	TXIF	RCIF	-	-	-	-	0000 0000	uuuu uuuu	***** r1, r1
029h	WREG	Working Register								xxxx xxxx	uuuu uuuu	***** r1, r1
02Ah	BSRCN	-	-	-	-	-	-	-	BSR[0]xu	~r1,~r1,~r1,~r1*
02Bh	MSTAT	-	-	-	C	DC	N	OV	Z	...x xxxx	...u uuuu	~r1,~r1,~r1,~r1*
02Ch	PSTAT	BOR	PD	TO	IDL	RST	SKERR	-	-	\$000 \$0.	uu\$u u\$.	rw0, rw0, rw0, rw0, rw0, ~r1,~r1*
02Eh	BIECN	1	-	-	ENBVD	VPPHV	ENBCP	BIEWR	BIERD	1.00 \$000	1.00 \$uuu	r1,~r1,~r1,~r1*
02Fh	BIEARH	-	-	1	1	1	1	1	1	0.xx xxxx	u.uu uuuu	~r1,~r1,~r1,~r1*
030h	BIEARL	BIE Address Register as BIEAL[5:0]								xxxx xxxx	uuuu uuuu	***** r1, r1
031h	BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	***** r1, r1
032h	BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	***** r1, r1
033h	PWRCN	ENBGR	LDOC[2:0]		LDOM[1:0]		ENLDO	CSFON		1000 0000	uuuu uu0u	***** r1, r1
034h	OSCCN0	-	OSCS[0]	DHS[1:0]		DMS[2:0]		CUPS		0000 0000	uuuu uuuu	***** r1, r1
035h	OSCCN1	CCOPT	-	DADC[1:0]		-	-	-	LCDS	0000 0000	uuuu uuuu	***** r1, r1
036h	OSCCN2	DLCD[1:0]		-	-	HAOM[1:0]		ENHAO		0000 0011	uuuu uu11	***** r1, r1
037h	CSFCN0	SKRST	HAOTR[6:0]						.1.	~r1,~r1,~r1,~r1*
038h	CSFCN1	ENSDRV	-	-	-	-	-	-	-	uuuu uuuu	***** r1, r1
039h	WDTCN	ENBZ	BZS	BZ[1:0]		ENWDT	DWDWT[2:0]		0000 0000	uuuu \$000	~r1,~r1,~r1,~r1*	
03Ah	AD1H	ADC1 conversion high byte data register								.00 0000	.uu uuuu	~r1,~r1,~r1,~r1*
03Bh	AD1M	ADC1 conversion middle byte data register								0000 0000	uuuu uuuu	***** r1, r1
03Ch	AD1L	ADC1 conversion low byte data register								0000 0000	uuuu uuuu	***** r1, r1

表 2-15 数据存储器列表

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ARST	IRST	R/W
03Dh	AD1CN0	ENAD1	OSR[3:0]			CMFR				0000 0000	uuuu uuuu	***** 3 3 3 3 3 3
03Eh	AD1CN1	-	VREGN	PGAGN[1:0]		ADGN[2:0]			xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
03Fh	AD1CN2	INIS1	-	-	CHFlag	DCSET[3:0]			xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
040h	AD1CN3	INP[3:0]			INN[3:0]				xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
041h	AD1CN4	VRH[1:0]	VRL[1:0]		INX[1:0]	VRIS	INIS		0000 0000	uuuu uuuu	***** 3 3 3 3 3 3	
042h	AD1CN5	ENACM	-	VCMS	LDOPL	-	ENTPS	TPSCH		0000 0000	uuuu uuuu	***** 3 3 3 3 3 3
043h	LVDCN	-	-	PWRS	LVDS[3:0]			LVDO		0000 0000	uuuu uuuu	***** 3 3 3 3 3 3
044h	TMA1CN	ENTMA1	TMACL1	TMAS1	DTMA1[2:0]		-			0000 0000	u0uu uuuu	*,rw 1,*****
045h	TMA1R	TMA 1 counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0,rw0,rw0,rw0
046h	TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0,rw0,rw0,rw0
047h	PT1	-	-	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	xxxx xxxx	***** 3 3 3 3 3 3
048h	TRISC1	-	-	TC1.5	TC1.4	TC1.3	TC1.2	TC1.1	TC1.0	xx00 0000	uuuu uuuu	***** 3 3 3 3 3 3
049h	PT1DA	-	-	-	-	-	DA1.2	-	-	xxxx x0xx	uuuu uuuu	***** 3 3 3 3 3 3
04Ah	PT1PU	-	-	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	xx11 1111	uuuu uuuu	***** 3 3 3 3 3 3
04Bh	PT1M1	-	-	-	-	INTEG1[1:0]		INTEG0[1:0]		0000 0000	uuuu uuuu	***** 3 3 3 3 3 3
04Ch	PT8	-	-	-	-	-	-	PT8.1	PT8.0	xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3
04Dh	TRISC8	-	-	-	-	-	-	TC8.1	TC8.0	xxxx xx00	uuuu uuuu	***** 3 3 3 3 3 3
04Eh	PT8DA	-	-	-	-	-	-	DA8.1	DA8.0	xxxx xx00	uuuu uuuu	***** 3 3 3 3 3 3
04Fh	PT8PU	-	-	-	-	-	-	Pu8.1	Pu8.0	xxxx xx11	uuuu uuuu	***** 3 3 3 3 3 3
050h	UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	***** 3 3 3 3 3 3
051h	UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	-,r,r,r,r,r,r,rw 0
052h	BA0CN	-	-	-	-	ENCR	RC9	ENADD	ENABD 0000 uuuu	-,r,r,r,r,r,r,rw 0
053h	BG0RH	-	-	-	Baud Rate Generator Register High Byte				...x xxxx	...u uuuu	-,r,r,r,r,r,r,rw 0	
054h	BG0RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3
055h	TX0R	UART Transmit Register								xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3
056h	RC0REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r
057h	LCDCN1	ENLCP	LCDV[2:0]			ENLB	SELPCLK	-	-	0000 0000	uuuu uuuu	***** 3 3 3 3 3 3
058h	LCDCN2	-	-	-	-	-	-	LCDBL	LCI00uu	***** 3 3 3 3 3 3
059h	LCDCN3	SCM3[1:0]		SCM2[1:0]		SCM1[1:0]		SCM0[1:0]		1111 1111	uuuu uuuu	***** 3 3 3 3 3 3
05Ah	LCDCN4	-	-	-	-	-	-	SSG15	SSG14	0000 0000	uuuu uuuu	***** 3 3 3 3 3 3
05Bh	LCD0	LCD SEG3[4:7] data			LCD SEG2[3:0] data				xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
05Ch	LCD1	LCD SEG5[4:7] data			LCD SEG4[3:0] data				xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
05Dh	LCD2	LCD SEG7[4:7] data			LCD SEG6[3:0] data				xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
05Eh	LCD3	LCD SEG9[4:7] data			LCD SEG8[3:0] data				xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
05Fh	LCD4	LCD SEG11[4:7] data			LCD SEG10[3:0] data				xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
060h	LCD5	LCD SEG13[4:7] data			LCD SEG12[3:0] data				xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
061h	LCD6	LCD SEG15[4:7] data			LCD SEG14[3:0] data				xxxx xxxx	uuuu uuuu	***** 3 3 3 3 3 3	
080h ~ 17Fh	SRAM as 256Byte									uuuu uuuu	uuuu uuuu	***** 3 3 3 3 3 3

表 2-5 数据存储寄存器列表(续)

3. 振荡器、频率源与功耗管理

HY17P 系列具有 HAO、LPO 两个频率源，如 表 3-1。透过频率控制器寄存器的设置可弹性的分配与管理 CPU 与外围工作频率，更能适当调整芯片消耗功率达到节约能源的目的。

频率控制寄存器摘要：

OSCCN0 OSCS[1:0], DHS[1:0], DMS[2:0], CUPS
OSCCN1 LCPS, DADC[1:0], DTMB[1:0], TMBS, LCDS
OSCCN2 DLCD[1:0], ENXT, XTS[1:0], HAOM[1:0], ENHAO

符号	频率	频率源控制器 CLKCN[7:0]配置		指令执行状态	
		ENHAO	HAOM[1:0]	SLP	IDLE
HAO	1.843MHz	1	00	停止	振荡
	3.686MHz	1	01	停止	振荡
	-	1	10	停止	振荡
	7.834MHz	1	11	停止	振荡
LPO	14.5KHz	芯片上电后即起振		停止	振荡

表 3-1 内部 RC 振荡器参数、频率控制器配置与指令状态

3.1. 振荡器

3.1.1. HAO 振荡器

HAO 为内部高速 RC 振荡器，典型输出频率为 1.843~7.834MHz。

HY17P 系列产品在 CPU 使用其他的振荡器作为工作频率源时，可透过 ENHAO 设置<0>将 HAO 振荡器关闭。

3.1.2. LPO 振荡器

LPO 为内部低速 RC 振荡器，典型的输出频率为 14.5KHz。主要应用于低速省电的 CPU 工作模式频率源。

HY17P 系列产品在执行 Sleep 指令后 LPO 振荡器会被关闭，而当芯片被唤醒时 LPO 将自动开启振荡器。

3.2. CPU 及外围电路频率源

3.2.1. 频率源分配

两组振荡器输出(HS_CK、LS_CK)会先经过前置工作频率分配器进行启用/停止、切换与预先除频后再进入芯片的 CPU 与各外围电路。如图 3-1 所示。

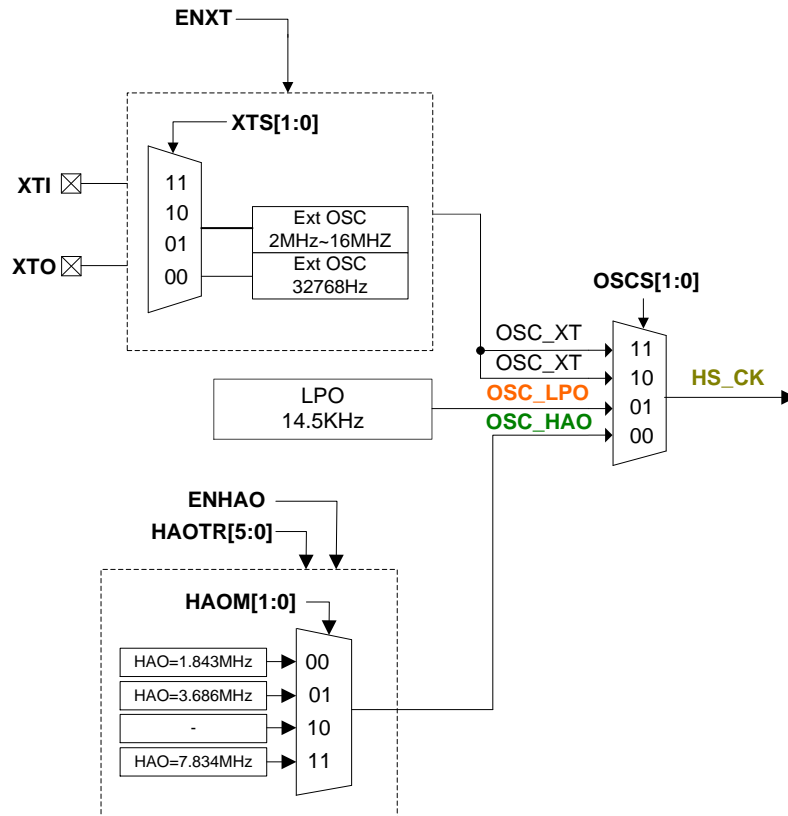


图 3-1 前置工作频率分配器

3.2.2. CPU 频率源

CPU 有多种工作频率可以选择，透过 CPUS 可选择工作频率来自 HS_CK 或 DHS_CK。

指令工作频率则采 1/4 的 CPU_CK 设计且分频出 INTR_CK 的频率源。

- 操作 Σ ADC 时，建议 CPU 使用 HS_CK 后分频当工作频率，以得到较佳的性能。
- 当 CPU_CK 的频率与指令执行周期，如表如图 3-2。表 3-2 简略列出 CPU 工作频率与指令周期的关系。

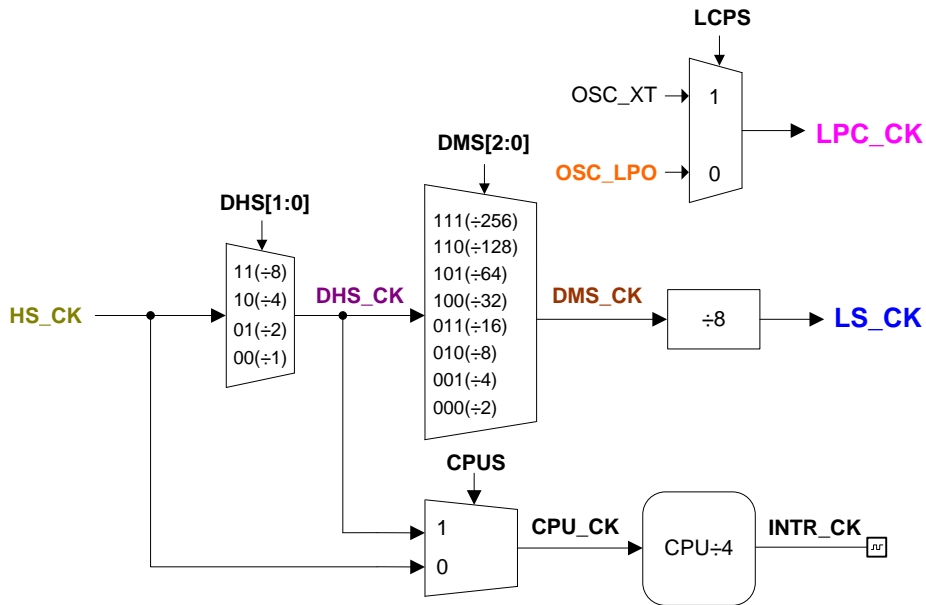


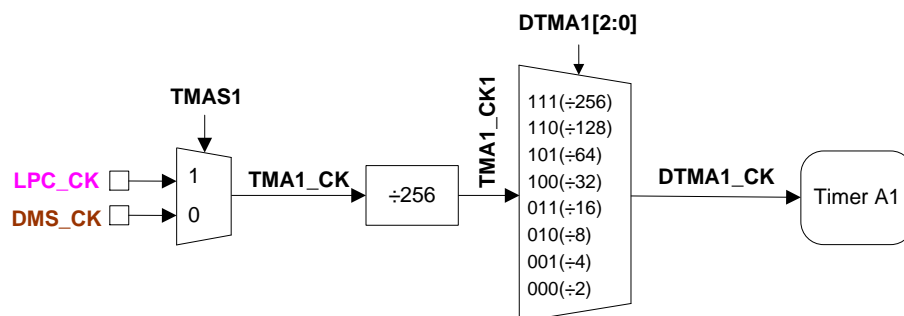
图 3-2 CPU 与外围工作频率

工作频率 CPU_CK	指令		
	频率	频率	周期
7.834MHZ	7.834MHZ	1.96MHZ	0.51us
3.686MHZ	3.686MHZ	0.92MHZ	1.08us
1.843MHZ	1.843MHZ	0.46MHZ	2.17us
32.768KHZ	32.768KHZ	16.384KHZ	122.07us
14.5KHZ	14.5KHZ	3.625KHZ	275.86us

表 3-2 CPU 工作频率与指令执行周期

3.2.3. CPU 周边电路频率源

HY17P 系列外围电路的工作频率系由不同的分配控制器与分频器进行配置，该配置将于各外围单元作详细说明故于此只附上外围工作频率配置图，如图 3-3。



HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

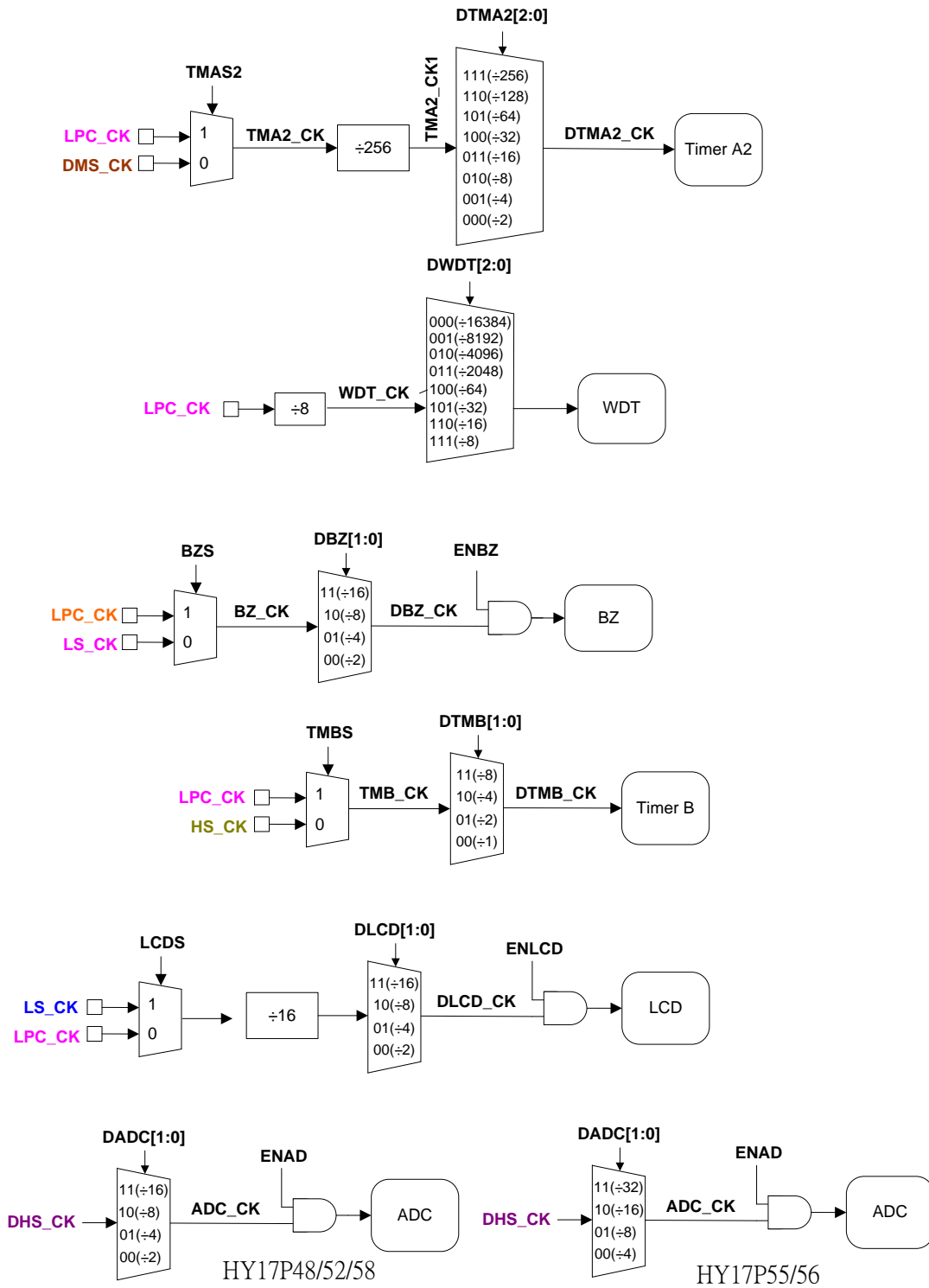


图 3-3 外围工作频率配置图

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



3.3. 寄存器说明-工作频率源控制器

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
PWRCN								CSFON	0000 0000	uuuu u00u	***,*,wr0,wr0,*
OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	***,*,*,*,*
OSCCN1		LCPS	DADC[1:0]		DTMB[1:0]		TMBS	LCDS	0000 0000	uuuu uu.	***,*,*,*,-
OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	***,*,*,*,f
CSFCN0			HAOTR[5:0]					0u	..,.,.,.,.,*

表 3-3 工作频率源控制寄存器

OSCCN0[7:0] : 晶片工作频率控制暂存器

位	名称	描述
Bit7~6	OSCS[1:0]	HS_CK 的频率选择器 <00>OSC_HAO <01>OSC_LPO <10> OSC_XT <11>OSC_XT
Bit5~4	DHS[1:0]	DHS_CK 分频器的分频选择器 <00>HS_CK ÷ 1 <01>HS_CK ÷ 2 <10>HS_CK ÷ 4 <11>HS_CK ÷ 8
Bit3~1	DMS[2:0]	DMS_CK 分频器的分频选择器 <000>DHS_CK ÷ 2 <001>DHS_CK ÷ 4 <010>DHS_CK ÷ 8 <011>DHS_CK ÷ 16 <100>DHS_CK ÷ 32 <101>DHS_CK ÷ 64 <110>DHS_CK ÷ 128 <111>DHS_CK ÷ 256
Bit0	CUPS	CPU_CK 的频率选择器 <0>HS_CK <1>DHS_CK

OSCCN1[7:0] : 晶片工作频率控制暂存器

位	名称	描述
Bit6	LCPS	LPC_CK 的频率选择器 <0>OSC_LPO <1>OSC_XT

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述																								
Bit5~4	DADC[1:0]	ADC_CK 频率分频选择器 <table border="1"> <thead> <tr> <th colspan="2">HY17P48/52/58</th> <th colspan="2">HY17P55/56</th> </tr> <tr> <th>DADC[1:0]</th> <th>Pre-scale</th> <th>DADC[1:0]</th> <th>Pre-scale</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>DHS_CK le2</td> <td>00</td> <td>DHS_CK le4</td> </tr> <tr> <td>01</td> <td>DHS_CK le4</td> <td>01</td> <td>DHS_CK le8</td> </tr> <tr> <td>10</td> <td>DHS_CK le8</td> <td>10</td> <td>DHS_CK le16</td> </tr> <tr> <td>11</td> <td>DHS_CK le16</td> <td>11</td> <td>DHS_CK le32</td> </tr> </tbody> </table>	HY17P48/52/58		HY17P55/56		DADC[1:0]	Pre-scale	DADC[1:0]	Pre-scale	00	DHS_CK le2	00	DHS_CK le4	01	DHS_CK le4	01	DHS_CK le8	10	DHS_CK le8	10	DHS_CK le16	11	DHS_CK le16	11	DHS_CK le32
HY17P48/52/58		HY17P55/56																								
DADC[1:0]	Pre-scale	DADC[1:0]	Pre-scale																							
00	DHS_CK le2	00	DHS_CK le4																							
01	DHS_CK le4	01	DHS_CK le8																							
10	DHS_CK le8	10	DHS_CK le16																							
11	DHS_CK le16	11	DHS_CK le32																							
Bit3~2	DTMB[1:0]	DTMB_CK 频率分频选择器 <table border="1"> <thead> <tr> <th>DTMB[1:0]</th> <th>Pre-scale</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>TMB_CK ÷ 1</td> </tr> <tr> <td>01</td> <td>TMB_CK ÷ 2</td> </tr> <tr> <td>10</td> <td>TMB_CK ÷ 4</td> </tr> <tr> <td>11</td> <td>TMB_CK ÷ 8</td> </tr> </tbody> </table>	DTMB[1:0]	Pre-scale	00	TMB_CK ÷ 1	01	TMB_CK ÷ 2	10	TMB_CK ÷ 4	11	TMB_CK ÷ 8														
DTMB[1:0]	Pre-scale																									
00	TMB_CK ÷ 1																									
01	TMB_CK ÷ 2																									
10	TMB_CK ÷ 4																									
11	TMB_CK ÷ 8																									
Bit1	TMBS	TMB_CK 的频率选择器 <0>HS_CK <1>LPC_CK																								
Bit0	LCDS	LCD_CK 的频率选择控制制器。 <0> LPC_CK <1> LS_CK																								

OSCCN2[7:0] : 芯片工作频率控制寄存器

位	名称	描述										
Bit7~6	DLCD[1:0]	DLCD_CK 频率分频控制器 注意：在 LCD Mode 时候 DLCD 为分频控制器 <table border="1"> <thead> <tr> <th>DLCD[1:0]</th> <th>Pre-scale</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>LCD_CK le2</td> </tr> <tr> <td>01</td> <td>LCD_CK le4</td> </tr> <tr> <td>10</td> <td>LCD_CK le8</td> </tr> <tr> <td>11</td> <td>LCD_CK le16</td> </tr> </tbody> </table>	DLCD[1:0]	Pre-scale	00	LCD_CK le2	01	LCD_CK le4	10	LCD_CK le8	11	LCD_CK le16
DLCD[1:0]	Pre-scale											
00	LCD_CK le2											
01	LCD_CK le4											
10	LCD_CK le8											
11	LCD_CK le16											
Bit5	ENXT	外部振荡器 XT 启用控制器 <0>关闭外部 XT 振荡器 <1>启用外部 XT 振荡器										

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



位	名称	描述										
Bit4~3	XTS[1:0]	外部振荡器振荡频率选择器 <table border="1"> <thead> <tr> <th>XTS[1:0]</th> <th>外部振荡器频率范围</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>~ 32768Hz(正常驱动)</td> </tr> <tr> <td>01</td> <td>~ 32768Hz(加强驱动电流)</td> </tr> <tr> <td>10</td> <td>2M (low power)</td> </tr> <tr> <td>11</td> <td>2~ 16MHz</td> </tr> </tbody> </table>	XTS[1:0]	外部振荡器频率范围	00	~ 32768Hz(正常驱动)	01	~ 32768Hz(加强驱动电流)	10	2M (low power)	11	2~ 16MHz
XTS[1:0]	外部振荡器频率范围											
00	~ 32768Hz(正常驱动)											
01	~ 32768Hz(加强驱动电流)											
10	2M (low power)											
11	2~ 16MHz											
Bit2~1	HAOM[1:0]	内部振荡器 HAO 振荡频率选择器 <table border="1"> <thead> <tr> <th>HAOM[1:0]</th> <th>HAO 振荡频率</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>1.843 MHz</td> </tr> <tr> <td>01</td> <td>3.686 MHz</td> </tr> <tr> <td>10</td> <td>不可设置</td> </tr> <tr> <td>11</td> <td>7.834 MHz</td> </tr> </tbody> </table>	HAOM[1:0]	HAO 振荡频率	00	1.843 MHz	01	3.686 MHz	10	不可设置	11	7.834 MHz
HAOM[1:0]	HAO 振荡频率											
00	1.843 MHz											
01	3.686 MHz											
10	不可设置											
11	7.834 MHz											
Bit0	ENHAO	内部 HAO 启用控制位 <0> 停止 <1> 启用										

PWRCN[7:0]线性稳压器与模拟共地控制寄存器

位	名称	描述
Bit0	CSFON	CSF(Chip Special Function)启用写入控制器 <0> 不启用 CSF 功能 <1> 启用CSF 写入功能,当用户需要进行此区控制寄存器设置时,必须将CFSON[0]设置<1>才能写入 CSFCN0[7:0]及 CSFCN1[7:0]。

CSFCN0[7:0]特殊控制位寄存器

位	名称	描述
Bit6~0	HAOTR	HAO 频率中心调整控制器 <0000000>调整 34.00%(最大) . <1000000>中心点 0.00% . <1111111>调整 -34%(最小)

3.4. 功率消耗管理与操作状态

HY17P 系列 CPU 提供三种工作模式让用户可以再执行效能与省电得到最佳管理,三种模式分别为运行模式、待机模式(IDLE)、休眠模式(SLEEP)。

3.4.1. 运行模式

运行模式主要 CPU 依循频率源处理所有发生的事件,此时芯片所有周边外设皆可以正常运作,且功率消耗在同频率下时处于最大的状态。

3.4.2. 待机模式

待机模式是透过 IDLE 指令进入待机模式中,主要指 CPU 进入节能的状态即停止运作等待唤醒,并将 PSTAT 寄存器的 IDLEB 标志位置<1>。在此模式下芯片周边外设仍正常运作,当周边外设产生中断事件时将会唤醒 CPU。另外,看门狗计数溢出产生的信号在此模式下属于中断事件而非复位信号。

在待机模式下,CPU 为暂停模式,停在 IDLE 指令下,其内部振荡器均不受影响也未关闭,使用者若为达更节能状态,需视应用情形,适时关闭周边外设或者振荡器等资源。而芯片则必须透过外部中断源,或者其他周边外设资源中断信号达到离开待机模式状态。

在待机模式下,若遇到中断而离开待机模式时,则需要 2 个指令周期时间才能回到中断向量 04H 位置。若在待机模式下,CPU 频率源为内部 LPO,且内部 HAO 被关闭的情况下。而在唤醒之后启动 HAO 振荡器,则需要 2 个 LPO 指令周期,加上 128 个 HAO 指令周期的起振时间

3.4.3. 休眠模式

休眠模式是透过 SLP 指令进入休眠模式中,主要指芯片处于停止运作状态 CPU、内部振荡器(HAO 与 LPO)与外部振荡器(XT)将停止工作,外围使用 HAO、LPO、XT 等相关频率源亦停止运作并将 PTATUS 复位寄存器的 PD 标志位置<1>在此模式下,芯片将处于等待中断事件进行唤醒,因为部分外围虽停止运作但其功能仍处于开启状态,故为了达到休眠模式芯片功耗处于最低的情况下非用于唤醒芯片的外围必需做进一步的功能关闭。

在休眠模式前,若 CPU 频率源为 HAO 频率源,则在休眠后遇到中断源而离开休眠模式时,重新启动 HAO 频率源则需要 1024 个 HAO 个指令时间。若 CPU 频率源为 LPO 频率源,则在休眠后遇到中断源而离开休眠模式时,重新启动 LPO 频率源则需要 1024 个 LPO 指令时间。

4. 复位,RESET

HY17P 系列的复位线路包含以下几种事件来触发复位讯号，复位方块图如 图 4-1。

- **BOR** 电源干扰复位。
- **RST** 外部复位输入引脚。
- **WDT** 看门狗复位。
- **SKERR** 堆栈错误复位。(使用者决定)

操作状态寄存器摘要：

PSTAT POR[0],PD[0],TO[0],IDL[0],SKERR[0],BOR2LV[0],GCRstIF[0]

PWRCN CSFON[0]

CSFCN0 SKRST[0]

CSFCN1 BOR_TH[2:0],BORS[0],ENBOR2[0]

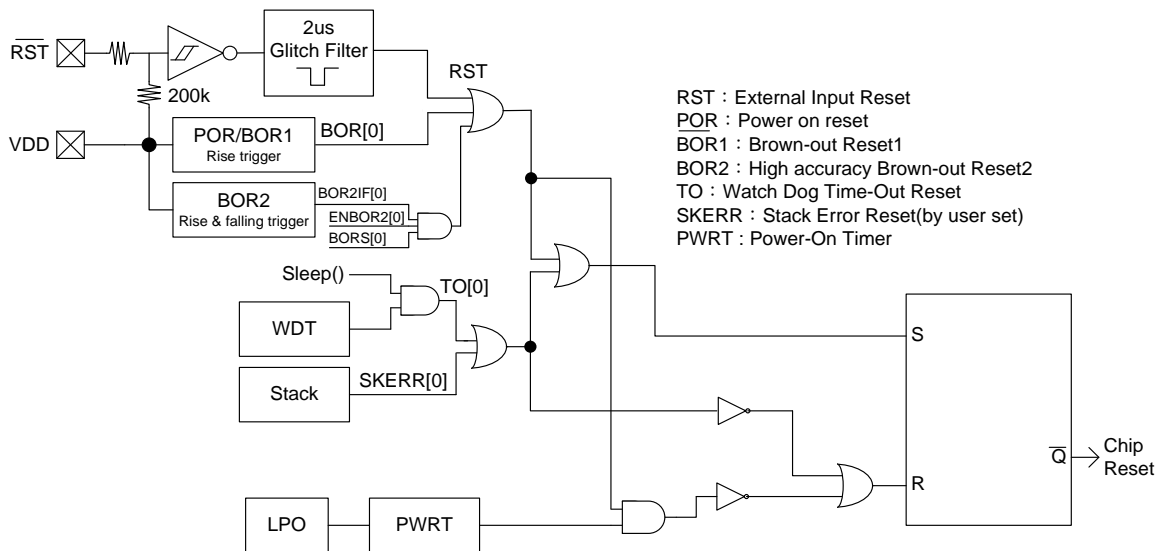


图 4-1 复位方块图

这些复位事件可区分为硬件复位及软件复位，说明如表 4-1。CPU 经复位后程序由 0x0000h 启动。

复位种类	事件	符号	说明
硬件复位	BOR	A-RESET	CPU 重新启动，须等待内部振荡器启动计数完成后方能进入正常工作状态。
	RST		
低阶复位	WDT	I-RESET	仅清除部分寄存器，CPU 快速回到正常工作状态。
	SKERR		

表 4-1 复位等级表

4.1. 复位事件说明

4.1.1. BOR 电源干扰复位

当 CPU 在上电过程或电源受外界干扰时, CPU 会由不正常工作的过低工作电压进入正常工作电压。因此, 如 CPU 在过低工作电压时无法处于复位状态, 将会造成 CPU 当机使外围电路工作异常。所以必须靠着 BOR 线路功能, 在侦测到工作电压受到干扰且电压准位低于设计值, 会产生复位信号使芯片进入重新启动状态, 直至回复工作电压才会解除复位信号使芯片进入正常工作模式。

4.1.2. BOR1 为省电 BOR

当发生 BOR 复位时, PSTAT[7:0]寄存器中的 BOR 标志会被置<1>以记录发生的事件。

BOR1 线路会产生约 0.2uA 的电流消耗, 无法透过程序或其他设置方式使其关闭。

4.1.3. BOR2 为精准 BOR

BOR2 预设会透过 POR/BOR1 触发启动, BOR2 功能可以透过软件关闭。关闭位 ENBOR2[0]在 normal mode 使用者可以操作。该位具有保护, 需要连动开关启动才可以有效设置。

触发 BOR2 行为可以设定为芯片 Reset 或是中断 BOR2IF 标志。

控制位 BORS[0]在正常模式用户可以操作。该位具有保护, 需要连动开关 CSFON[0]启动才可以有效设置。

4.1.4. SKERR 堆栈错误复位

程序发生堆栈溢出或欠位时会产生复位信号使芯片进入快速启动状态。当发生 SKERR 堆栈错误复位时 PSTAT[7:0]寄存器中的 SKERR 标志会被置<1>以记录发生的事件。详细的操作说明请参见[内存](#)章节。

4.2. 状态寄存器

芯片的操作状态显示于 PSTAT[7:0]复位寄存器, 相互间关系如表 4-2。

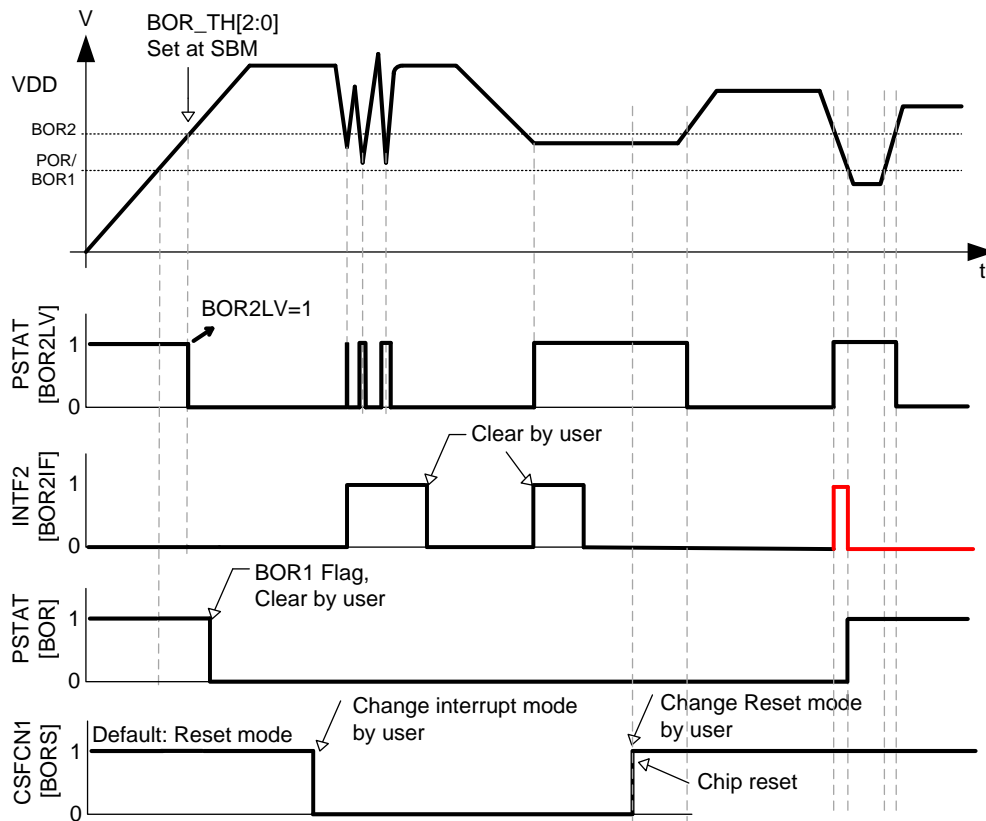
“0”：未发生,“1”：已发生,“u”：不改变,“-”：未使用

名称/状态	地址	7	6	5	4	3	2	1	0
PSTAT	02CH	BOR	PD	TO	IDL	RST	SKERR	-	-
硬件复位 (A-RESET)	BOR	1	0	0	0	0	0	-	-
	RST	0	0	0	0	1	0	-	-
软件复位 (I-RESET)	WDT	u	u	1	u	u	u	-	-
	SKERR	u	u	u	u	u	1	-	-

表 4-2 复位状态标志关系表

4.2.1. 复位状态的时序图

硬件复位信号发生后至芯片进入操作状态的时序图，如图 4-2。不同复位信号信号发生后至芯片进入操作状态的时间。



*. BOR2 触发后, Power up count: $1024 \text{ HAO} + 1024 \text{ LPO} = 1024 * (1/1.843\text{Mhz}) + 1024 * (1/14\text{Khz}) = 73.7\text{msec}$.

图 4-2 复位及操作模式与状态标志时序图

“-”：无定义

复位信号	延迟时间			操作状态		
	符号	T1	T2	运行	待机	休眠
BOR	t_{RST}	T1 + T2		有效	有效	有效
SKERR	-	-		有效	无效	无效

表 4-3 复位状态的延迟时间与操作状态关系表

4.3. 寄存器说明-复位状态

“.”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1 “\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE2								BOR2IE	0000 0000	uuuu uuuu	*** **
INTF2								BOR2IF	0000 0000	uuuu uuuu	*** **
PSTAT	BOR	PD	TO	IDL	RST	SKERR	BOR2LV	GCRstIF	\$000 \$000	uu\$u u\$uu	w0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
PWRCN								CSFON	0000 0000	uuuu u00u	*** ** ,wr0,wr0,*
CSFCN0	SKRST								.1.....	*** ** **
CSFCN1				BOR_TH[2:0]		BORS	ENBOR2		...0 0011	uuuu uuuu	*** **

表 4-4 复位寄存器

INTE2/INTF2: 详见中断 Interrupt 章节

PSTAT: 状态寄存器

位	名称	描述
Bit7	BOR	电源干扰复位标志 <0> 清除需透过指令。 <1> BOR 发生作用时置<1>。
Bit6	PD	休眠状态标志 <0> 清除需透过 BOR、RST 或指令。 <1> 执行 SLEEP 指令时置<1>。
Bit5	TO	看门狗定时器标志 <0> 清除需透过 BOR、RST 或指令。 <1> 看门狗计数终止时置<1>。
Bit4	IDL	待机状态标志 <0> 清除需透过 BOR、RST 或指令。 <1> 执行 IDLE 指令时置<1>。
Bit3	RST	外部 RST 引脚低电位复位事件标志 <0> 未发生 RST 引脚复位事件 <1> 已发生 RST 引脚复位事件；清除需透过 BOR、或指令
Bit2	SKERR	堆栈错误复位标志 <0> 清除需透过 BOR、RST 或指令。 <1> 堆栈错误时置<1>。
Bit1	BOR2LV	BOR2 状态标志 <0> 表示 VDD 电压>BOR_TH[2:0] <1> 表示 VDD 电压<=BOR_TH[2:0]
Bit0	GCRstIF	I ² C Reset 命令标志 <0> 未发生 <1> 已发生

PWRCN[7:0]:线性稳压器与模拟共地控制寄存器

位	名称	描述
---	----	----

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



位	名称	描述
Bit0	CSFON	CSF(Chip Special Function)启用写入控制器 <0> 不启用 CSF 功能 <1> 启用 CSF 写入功能,当用户需要进行此区控制寄存器设置时,必须将 CSFON[0] 设置<1>才能写入 CSFCN0[7:0]及 CSFCN1[7:0]。

CSFCN0: 特殊控制位寄存器 0

位	名称	描述
Bit7	SKRST	堆栈错误复位控制器 <0> 不启用错误复位芯片 <1> 启用错误复位芯片

※ CSFCN0 在正常模式用户可以操作,该位具有保护,需要须将 CSFON[0]设为 1,才能修改此位设定。

CSFCN1: 特殊控制位寄存器 1

位	名称	描述																											
Bit4~2	BOR_TH[2:0]	BOR2 侦测电压设定 <table border="1"> <thead> <tr> <th>BOR_TH[2:0]</th> <th>BOR2 电压</th> <th>说明</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>1.7V</td> <td>为芯片上电默认值。</td> </tr> <tr> <td>001</td> <td>2.0V</td> <td>当电池使用 1.5V*2,则电池为 2V(=1V*2)表示低电压</td> </tr> <tr> <td>010</td> <td>2.2V</td> <td></td> </tr> <tr> <td>011</td> <td>2.5V</td> <td>for VDDA=2.4V mode, VDD>=2.45V</td> </tr> <tr> <td>100</td> <td>2.75V</td> <td>当电池使用 1.5V*3,则电池为 2.7V(=0.9V*3)表示低电压</td> </tr> <tr> <td>101</td> <td>3.0V</td> <td>当电池使用 1.5V*3,则电池为 3V(=1V*3)表示低电压</td> </tr> <tr> <td>110</td> <td>3.65V</td> <td>当电池使用 1.5V*4,则电池为 3.6V(=0.9V*4)表示低电压</td> </tr> <tr> <td>111</td> <td>4.0V</td> <td>当电池使用 1.5V*4,则电池为 3.6V(=1.0V*4)表示低电压</td> </tr> </tbody> </table>	BOR_TH[2:0]	BOR2 电压	说明	000	1.7V	为芯片上电默认值。	001	2.0V	当电池使用 1.5V*2,则电池为 2V(=1V*2)表示低电压	010	2.2V		011	2.5V	for VDDA=2.4V mode, VDD>=2.45V	100	2.75V	当电池使用 1.5V*3,则电池为 2.7V(=0.9V*3)表示低电压	101	3.0V	当电池使用 1.5V*3,则电池为 3V(=1V*3)表示低电压	110	3.65V	当电池使用 1.5V*4,则电池为 3.6V(=0.9V*4)表示低电压	111	4.0V	当电池使用 1.5V*4,则电池为 3.6V(=1.0V*4)表示低电压
BOR_TH[2:0]	BOR2 电压	说明																											
000	1.7V	为芯片上电默认值。																											
001	2.0V	当电池使用 1.5V*2,则电池为 2V(=1V*2)表示低电压																											
010	2.2V																												
011	2.5V	for VDDA=2.4V mode, VDD>=2.45V																											
100	2.75V	当电池使用 1.5V*3,则电池为 2.7V(=0.9V*3)表示低电压																											
101	3.0V	当电池使用 1.5V*3,则电池为 3V(=1V*3)表示低电压																											
110	3.65V	当电池使用 1.5V*4,则电池为 3.6V(=0.9V*4)表示低电压																											
111	4.0V	当电池使用 1.5V*4,则电池为 3.6V(=1.0V*4)表示低电压																											
Bit1	BORS	BOR2 行为设定 <0> BOR2 为中断唤醒功能, BOR2IE=1 且 BOR2IF=1 时,产生中断事件。 <1> BOR2 为芯片重置功能, BOR2IF=0 时重置芯片。为芯片上电默认值。																											
Bit0	ENBOR2	BOR2 启用与关闭控制器 <0> 关闭 BOR2 <1> 启用 BOR2																											

※ CSFCN1 在正常模式用户可以操作,该位具有保护,需要须将 CSFON[0]设为 1,才能修改此位设定。

5. 中断, Interrupt

中断 Interrupt 由中断使能控制器 INTE 与中断标志位寄存器 INTF 组成。中断服务 Interrupt service 成立时若产生中断事件，将使得程序计数器 PC 跳至程序内存的中断向量地址 0x0004h 执行中断服务程序。

中断控制寄存器寄存器摘要：

INTE0	GIE, TA1CIE, ADIE, WDTIE, TB1IE, E1IE, E0IE
INTE1	TA1IE, SPIIE, TXIE, RCIE, I2CERIE, I2CIE, E3IE, E2IE
INTE2	TA2IE, TA2CIE, TC11IE, TC10IE, TX2IE, RC2IE, BOR2IE
INTF0	TA1CIF, ADIF, WDTIF, TB1IF, E1IF, E0IF
INTF1	TA1IF, SPIIF, TXIF, RCIF, I2CERIF, I2CIF, E3IF, E2IF
INTF2	TA2IF, TA2CIF, TC11IF, TC10IF, TX2IF, RC2IF, BOR2IF
PT1INTE	INTE1.7, INTE1.6, INTE1.5, INTE1.4
PT1INTF	INTF1.7, INTF1.6, INTF1.5, INTF1.4
PT2INTE	INTE2.7, INTE2.6, INTE2.5, INTE2.4, INTE2.3, INTE2.2, INTE2.1, INTE2.0
PT2INTF	INTF2.7, INTF2.6, INTF2.5, INTF2.4, INTF2.3, INTF2.2, INTF2.1, INTF2.0

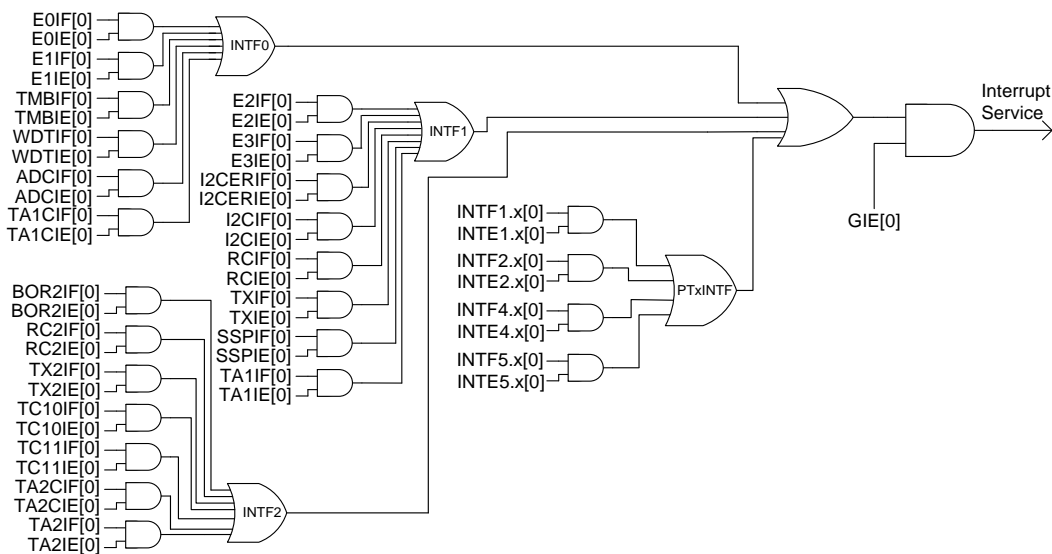


图 5-1 中断向量方块图

中断服务事件的控制器共有两层，最高层为中断服务控制器 GIE、次一层为中断事件的使能控制位。

- 使能中断事件只需将中断事件使能控制器 INTE_x[7:0]的相对应控制位设置<1>即可；反之，设置<0>则为关闭中断事件。
- 使能中断服务只需将中断控制寄存器 INTE0[7:0]的中断服务控制位 GIE 设置<1>即可；反之，设置<0>则为关闭中断服务。

当进入中断服务向量时 GIE 会自动被置<0>，在中断服务程序执行完毕后欲返回中断发生地址时可直接执行中断返回指令 RETI，此时 GIE 将自动被置<1>；或执行返回指令 RET，此时 GIE 状态维持 0。

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



5.1. 寄存器说明-中断

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	TA1CIE	ADIE	WDTIE	TB1IE	-	E1IE	E0IE	0000 0000	0uuu uuuu	***** 1 1 1 1 1 1 1
INTE1	TA1IE	SPIIE	TXIE	RCIE	I2CERIE	I2CIE	E3IE	E2IE	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
INTE2	TA2IE	TA2CIE	TC11IE	TC10IE	TX2IE	RC2IE	-	BOR2IE	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	-	E1IF	E0IF	.000 0000	.uuu uuuu	***** 1 1 1 1 1 1 1
INTF1	TA1IF	SPIIF	TXIF	RCIF	I2CERIF	I2CIF	E3IF	E2IF	0000 0000	uuuu uuuu	*** r, r, ** 1 1 1 1 1 1 1
INTF2	TA2IF	TA2CIF	TC1IF	TC0IF	TX2IF	RC2IF	-	BOR2IF	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
PT1INTE	INTE1.7	INTE1.6	INTE1.5	INTE1.4	-	-	-	-	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
PT1INTF	INTF1.7	INTF1.6	INTF1.5	INTF1.4	-	-	-	-	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
PT2INTE	INTE2.7	INTE2.6	INTE2.5	INTE2.4	INTE2.3	INTE2.2	INTE2.1	INTE2.0	0000 0000	uuuu uuuu	
PT2INTF	INTF2.7	INTF2.6	INTF2.5	INTF2.4	INTF2.3	INTF2.2	INTF2.1	INTF2.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
PT4INTE	INTE4.7	INTE4.6	INTE4.5	INTE4.4	-	-	-	-	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
PT4INTF	INTF4.7	INTF4.6	INTF4.5	INTF4.4	-	-	-	-	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
PT5INTE	-	-	-	-	INTE5.3	INTE5.2	INTE5.1	INTE5.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1
PT5INTF	-	-	-	-	INTF5.3	INTF5.2	INTF5.1	INTF5.0	0000 0000	uuuu uuuu	***** 1 1 1 1 1 1 1

表 5-1 中断寄存器

INTE0: 中断使能控制寄存器 0

位	名称	描述
Bit7	GIE	中断服务控制器 <0> 关闭。 <1> 启用。
Bit6	TA1CIE	Timer-A1 比较事件使能控制器 <0> 关闭。 <1> 启用。(比较事件/计数器 A1)
Bit5	ADCIE	ADC 中断事件使能控制器 <0> 关闭。 <1> 启用。(模拟数字转换器, Σ ADC)
Bit4	WDTIE	Watch Dog 中断事件使能控制器 <0> 关闭。 <1> 启用。(看门狗,WDT)
Bit3	TMBIE	Timer-B 中断事件使能控制器 <0> 关闭。 <1> 启用。(计时/计数器 B, TMB)
Bit1	E1IE	输入引脚 1 中断事件使能控制器 <0> 关闭。 <1> 启用。(外部输入引脚, PT1.1)
Bit0	E0IE	输入引脚 0 中断事件使能控制器 <0> 关闭。 <1> 启用。(外部输入引脚, PT1.0)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

INTE1: 中断使能控制寄存器 1

位	名称	描述
Bit7	TA1IE	Timer-A1 中断事件使能控制器 <0> 关闭。 <1> 启用。(计时/计数器 A1,TMA1)
Bit6	SPIIE	SPI 中断事件使能控制器 <0> 关闭。 <1> 启用。(通讯接口,SPI)
Bit5	TXIE	TX 中断事件使能控制器 <0> 关闭。 <1> 启用。(通讯接口,EUART)
Bit4	RCIE	RC 中断事件使能控制器 <0> 关闭。 <1> 启用。(通讯接口,EUART)
Bit3	I2CERIE	外围 I2C 错误中断向量服务控制器 <0>关闭 I2C 中断向量服务 <1>启用 I2C 中断向量服务
Bit2	I2CIE	外围 I2C 中断向量服务控制器 <1>启用 I2C 中断向量服务 <0>关闭 I2C 中断向量服务
Bit1	E3IE	输入引脚 3 中断事件使能控制器 <0> 关闭。 <1> 启用。(外部输入引脚,PT1.3)
Bit0	E2IE	输入引脚 2 中断事件启用控制器 <0> 关闭。 <1> 启用。(外部输入引脚,PT1.2)

INTE2: 中断使能控制寄存器 2

位	名称	描述
Bit7	TA2IE	Timer-A2 中断事件使能控制器 <0> 关闭。 <1> 启用。(计时/计数器 A2,TMA2)
Bit6	TA2CIE	Timer-A2 比较事件使能控制器 <0> 关闭。 <1> 启用。(比较事件/计数器 A2)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述
Bit5	TC11IE	外围 TMC11 中断向量服务控制器 <1> 启用 TMC11 中断向量服务 <0> 关闭 TMC11 中断向量服务
Bit4	TC10IE	外围 TMC10 中断向量服务控制器 <1> 启用 TMC10 中断向量服务 <0> 关闭 TMC10 中断向量服务
Bit3	TX2IE	TX2 中断事件使能控制器 <0> 关闭。 <1> 启用。(通讯接口,EUART2)
Bit2	RC2IE	RC2 中断事件使能控制器 <0> 关闭。 <1> 启用。(通讯接口,EUART2)
Bit0	BOR2IE	BOR2 中断事件使能控制器 <0> 关闭。 <1> 启用。

INTF0: 中断标志位寄存器 0

位	名称	描述
Bit6	TA1CIF	Timer-A1 比较中断标志 <0> 关闭。 <1> 启用。(比较事件/计数器 A1)
Bit5	ADCIF	ADC 中断标志 <0> 未发生。 <1> 已发生。(模拟数字转换器, Σ ADC)
Bit4	WDTIF	Watch Dog 中断标志 <0> 未发生。 <1> 已发生。(看门狗,WDT)
Bit3	TMBIF	Timer-B 中断标志 <0> 未发生。 <1> 已发生。(计时/计数器 B,TMB)
Bit1	E1IF	输入引脚 1 中断标志 <0> 未发生。 <1> 已发生。(外部输入引脚,PT1.1)
Bit0	E0IF	输入引脚 0 中断标志 <0> 未发生。 <1> 已发生。(外部输入引脚,PT1.0)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

INTF1: 中断标志位寄存器 1

位	名称	描述
Bit7	TA1IF	Timer-A1 中断标志 <0> 未发生。 <1> 已发生。(计时/计数器 A1,TMA1)
Bit6	SPIIF	SPI 中断标志 <0> 未发生。 <1> 已发生。(通讯接口,SPI)
Bit5	TXIF	TX 中断标志 <0> 未发生。 <1> 已发生。(通讯接口,EUART1)
Bit4	RCIF	RC 中断标志 <0> 未发生。 <1> 已发生。(通讯接口,EUART1)
Bit3	I2CERIF	外围 I ² C 错误中断标志 <0>未发生 I ² C 中断事件 <1>已发生 I ² C 中断事件
Bit2	I2CIF	外围 I ² C 中断事件标志 <0>未发生 I ² C 中断事件 <1>已发生 I ² C 中断事件
Bit1	E3IF	输入引脚 3 中断标志 <0> 未发生。 <1> 已发生。(外部输入引脚,PT1.3)
Bit0	E2IF	输入引脚 2 中断标志 <0> 未发生。 <1> 已发生。(外部输入引脚,PT1.2)

INTF2: 中断标志位寄存器 2

位	名称	描述
Bit7	TA2IF	Timer-A2 中断标志 <0> 未发生。 <1> 已发生。(计时/计数器 A2,TMA2)
Bit6	TA2CIF	Timer-A2 比较标志 <0> 关闭。 <1> 启用。(比较事件/计数器 A2)
Bit5	TC1IF	外围 TMCI1 中断标志 <1>已发生 TMCI1 中断事件 <0>未发生 TMCI1 中断事件

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述
Bit4	TC0IF	外围 TMCIO 中断标志 <1>已发生 TMCIO 中断事件 <0>未发生 TMCIO 中断事件
Bit3	TX2IF	TX 中断标志 <0> 未发生。 <1> 已发生。(通讯接口,EUART2)
Bit2	RC2IF	RC 中断标志 <0> 未发生。 <1> 已发生。(通讯接口,EUART2)
Bit0	BOR2IF	BOR2 中断标志 <0> 未发生。 <1> 已发生。

PT1INTE: PT1 I/O 中断使能控制寄存器

位	名称	描述
Bit7~4	INTE1.x	中断使能控制器 ($4 \leq x \leq 7$) <0>关闭。(预设) <1>启用。

PT1INTF: PT1 I/O 中断标志位

位	名称	描述
Bit7~4	INTF1.x	中断标志 ($4 \leq x \leq 7$) <0>关闭。(预设) <1>启用。

PT2INT: PT2 I/O 中断产生触发条件

位	名称	描述
Bit7~0	INTG2.x	中断产生触发条件 ($0 \leq x \leq 7$) <0>下降缘 ($1 \rightarrow 0$) (预设) <1>上升缘 ($0 \rightarrow 1$)

PT2INTE: PT2 I/O 中断使能控制寄存器

位	名称	描述
Bit7~0	INTE2.x	中断使能控制器 ($0 \leq x \leq 7$) <0> 关闭。(预设) <1> 启用。

PT2INTF: PT2 I/O 中断标志位

位	名称	描述
Bit7~0	INTF2.x	中断标志 ($0 \leq x \leq 7$) <0> 关闭。(预设) <1> 启用。

PT4INTE: PT4 I/O 中断使能控制寄存器

位	名称	描述
Bit7~4	INTE4.x	中断使能控制器 ($4 \leq x \leq 7$) <0> 关闭。(预设) <1> 启用。

PT4INTF: PT4 I/O 中断标志位

位	名称	描述
Bit7~4	INTF4.x	中断标志 ($4 \leq x \leq 7$) <0> 关闭。(预设) <1> 启用。

PT5INTE: PT5 I/O 中断使能控制寄存器

位	名称	描述
Bit3~0	INTE5.x	中断使能控制器 ($0 \leq x \leq 3$) <0> 关闭。(预设) <1> 启用。

PT5INTF: PT5 I/O 中断标志位

位	名称	描述
Bit3~0	INTF5.x	中断标志 ($0 \leq x \leq 3$) <0> 关闭。(预设) <1> 启用。

6. 硬件乘法器

H08D 指令集具有 8x8 硬件乘法器的处理指令“MULF 和 MULL”。8x8 硬件乘法器的运算结果会放至乘法器寄存器 PRODH[7:0]与 PRODL[7:0]且不会改变 PSTAT[7:0]状态寄存器中的任何标志。而 PRODH[7:0]与 PRODL[7:0]为只读寄存器，使用时必须注意。

硬件乘法器可进行有号数与无号数运算，如范例 6-1 与范例 6-2

```

例 1 : V1 x V2 = V
MVL    V1
MVF    BUF0,1,0    ; V1 值放入内存区块 0 的 BUF0 寄存器
MVL    V2           ; V2 值放入 W 寄存器。
MULF   BUF0,0      ; 执行 V1 x V2 并将运算结果放入 PRODH/L
    
```

范例 6-1 无号数运算

```

例 2 : N1 x N2 = N ,s=7,B
MVL    N1           ; N1 值放入 W 寄存器
MVF    BUF0,1,0    ; N1 值放入内存区块 0 的 BUF0 寄存器
MVL    N2           ; N2 值放入 W 寄存器。
MVF    BUF1,1,0    ; N2 值放入 BUF1 寄存器。
MULF   BUF0,0      ; 执行 V1 x V2 并将运算结果放入 PRODH/L
MVFF   PRODH,SWP   ; 将 PRODH 寄存器内的值放入 SWP 寄存器
BTSZ   BUF0,s      ; 判断 N1, 若为负数则
SUBF   SWP,1,0     ; 将 SWP - N2 放入 SWP 寄存器
MVF    BUF0,0,0    ; 将 N1 值放入 W 寄存器
BTSZ   BUF1,s      ; 判断 N2, 若为负数则
SUBF   SWP,1,0     ; 将 SWP - N1 放入 SWP 运算处理后, N = SWP/PRODL
; -----
; N1=07Fh,N2=0FFh 乘法器运算后得到 PRODH/L = 7E81h
; 判断 N1 是否为负数,若是负数则将 PRODH - N2
; 判断 N2 是否为负数,若是负数则将 PRODH - N1
; 运算处理后即可得到有号数 N 的值
; 7Fh x FFh = 7Fh x ( 0FFh - 100h )
;           = 7Fh x 0FFh - 7Fh x 100h
;           = 7E81h - 7F00h
;           = FF81h
    
```

范例 6-2 有号数运算

7. 输入/输出埠, I/O PORT

每个 I/O PORT 包含一个或多个输入/输出引脚, 输入/输出 I/O 每个引脚为一个端口, 可作为数字的输入与输出通道。每个 I/O PORT 由一组寄存器做控制。

I/O 相关寄存器摘要 :

PT	PT1[7:0], PT2[7:0], PT4[7:4], PT5[5:0], PT6[7:0], PT7[7:0], PT8[7:0], PT9[7:0], PT10[7:0], PT11[3:0]
TRISC	TC1[7:0], TC2[7:0], TC6[7:0], TC7[7:0], TC8[7:0], TC9[7:0], TC10[7:0], TC11[3:0],
PTDA	DA1[2], DA4[7:4], DA5[5:0],
PTPU	PU1[7:0], PU2[7:0], PU4[7:4], PU5[5:0],
PT1M1	INTEG1[1:0], INTEG0[1:0]
PT1INT	INTEG7[0], INTEG6[0], INTEG5[0], INTEG4[0], INTEG3[0], INTEG2[0],
PT2INT	INTG2.7, INTG2.6, INTG2.5, INTG2.4, INTG2.3, INTG2.2, INTG2.1, INTG2.0

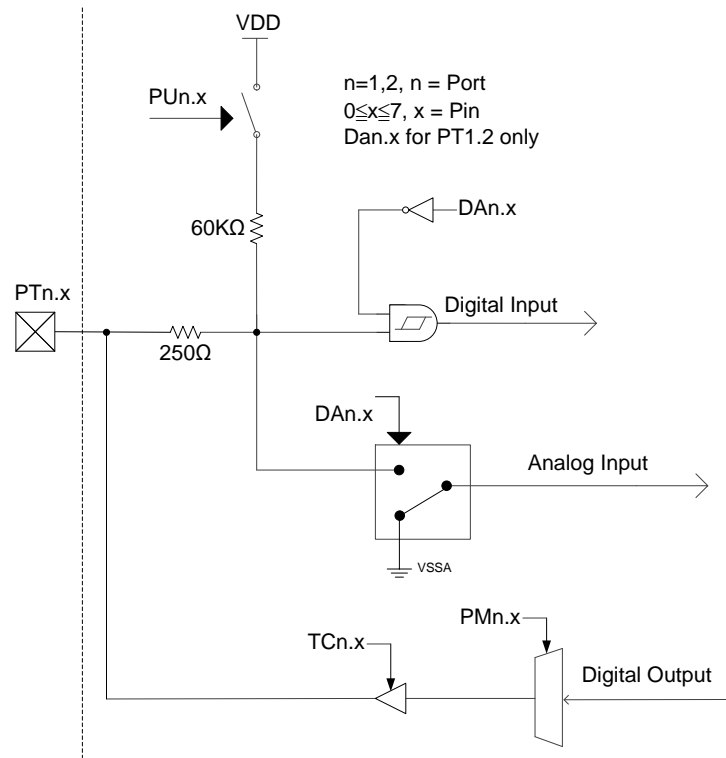


图 7-1 I/O PORT1~2 架构方块图

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

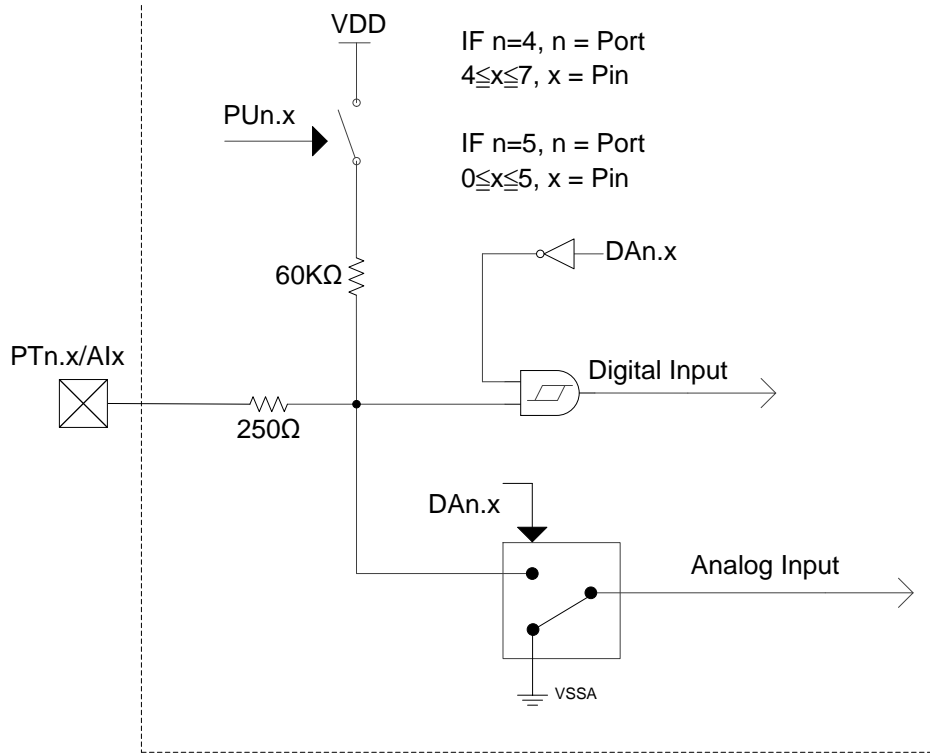
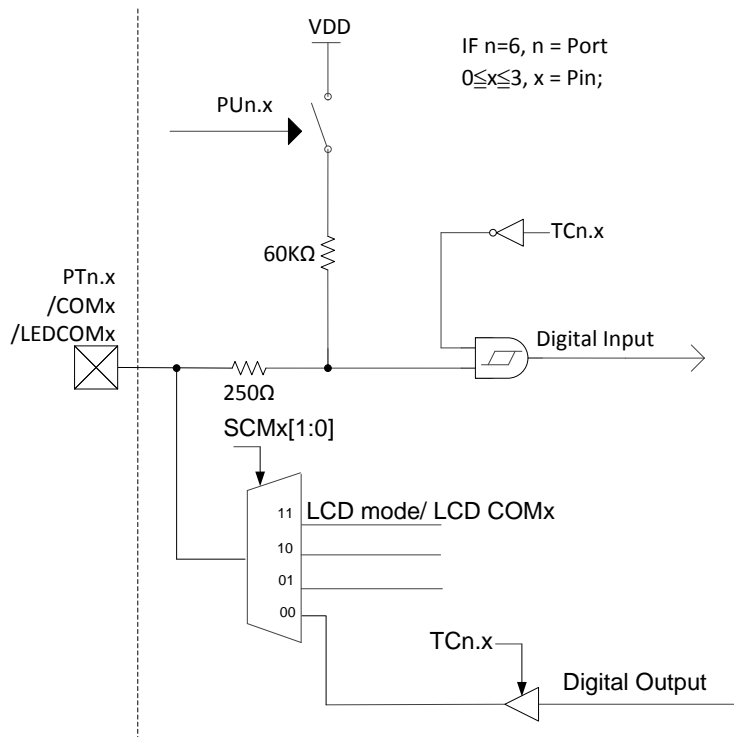


图 7-2 I/O PORT4/5 架构方块图



HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

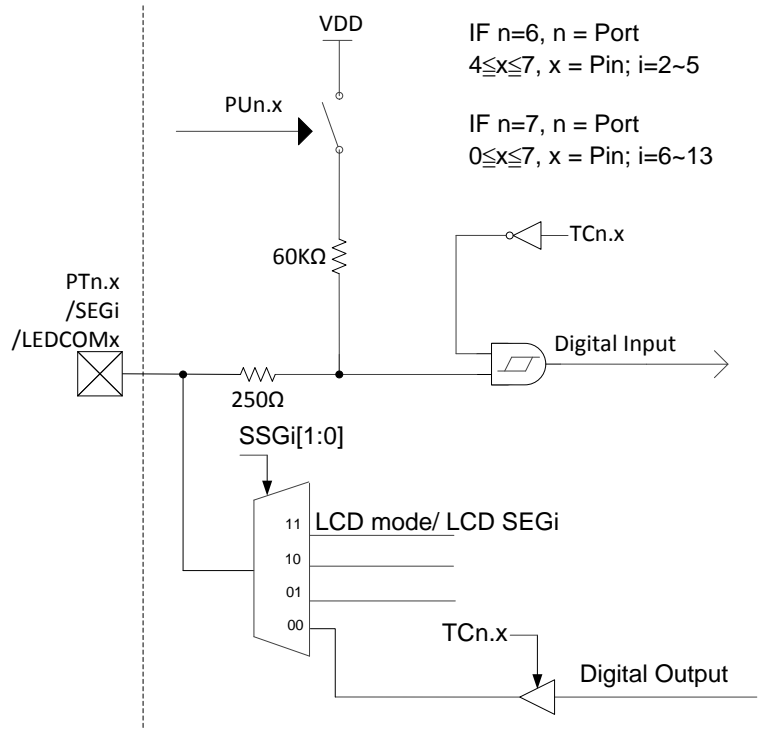


图 7-3 I/O PORT6~7 架构方块图

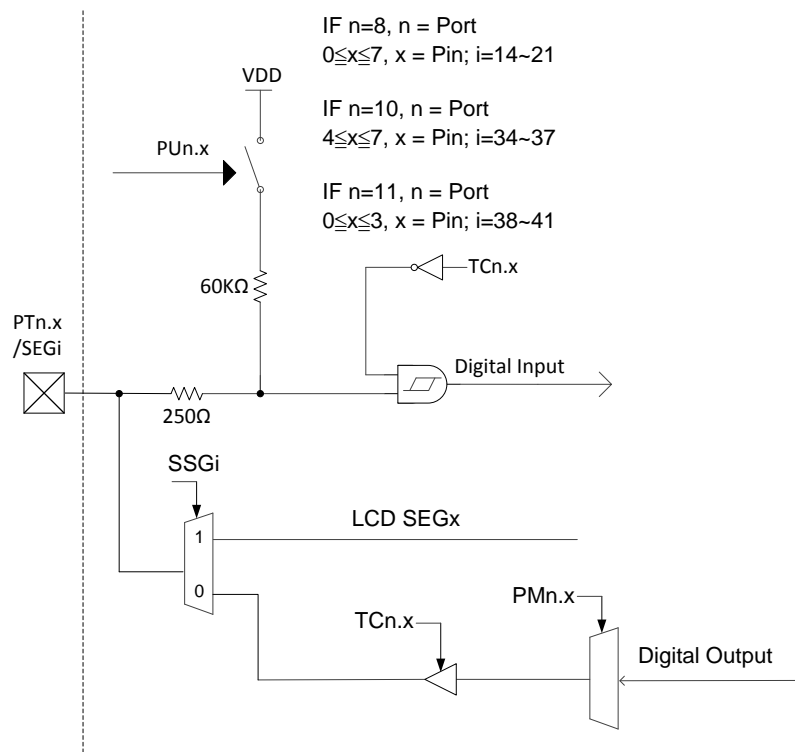


图 7-4 I/O PORT8,10,11 架构方块图

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

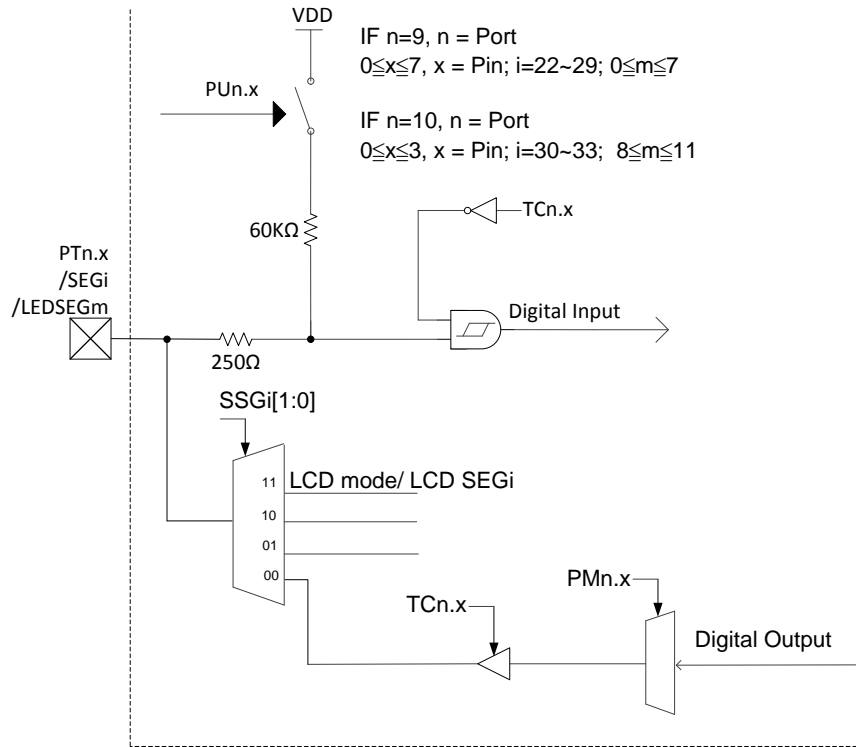


图 7-5 I/O PORT9~10 架构方块图

7.1. PORT 相关寄存器介绍

PORT 主要提供数字的信号输入与输出引脚。

7.1.1. PTEG 中断信号产生条件

I/O 外部输入电位属于何种变化时产生中断信号，电位变化条件可分上升缘（0→1）变化、下降缘（1→0）变化与电位转态（0→1 或 1→0）变化。

7.1.2. PTPU 上拉电阻控制寄存器

设定 I/O 上拉电阻功能是否启用，设置<1>则启用上拉电阻、设置<0>断开上拉电阻。在芯片进入休眠模式前，若 I/O 设置为数字输入状态且外部电路连接方式会造成 I/O 有浮接现象时即可启用上拉电阻，以避免 I/O 浮接而导致芯片进入休眠模式后产生漏电流。

7.1.3. TC 输入/输出控制寄存器

选择 I/O 作为输入或输出模式，设置<1>为输出状态、设置<0>为输入状态。当 I/O 设定为输入状态，则在芯片进入休眠模式时必须给定一明确的输入电位，不可让 I/O 呈现浮接状态，以避免造成芯片产生漏电现象。

7.1.4. PTIO 状态控制寄存器

当 I/O 被设置为输入模式时，则在对应的寄存器位置可以读得目前 I/O 的输入状态；读值 1 则此时的 I/O 输入高电平、读值 0 则此时的 I/O 输入低电平。

当 I/O 被设置为输出模式时，则在对应的寄存器位置可以控制输出状态，设置<1>则 I/O 输出为高电平、设置<0>则 I/O 输出为低电平。

7.2. 蜂鸣器,Buzzer

BZ 可产生多种不同的频率以推动外部蜂鸣器，由 BZS 选择工作频率源，BZ 工作频率分频器 BBZ[1:0] 可设置多种输出频率。

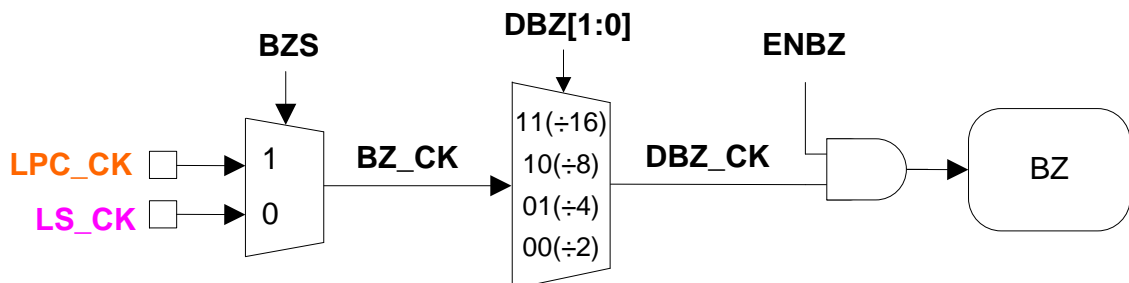


图 7-6 BZ 方块图

7.3. 输入/输出埠 1 ,I/O Port1(以 HY17P58 为例)

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PM[0]	
PT1.0	i/o	s/c	x	-	-	数字输入/输出引脚
INT0	i	s	0	-	-	外部中断源
SDI	i	s	0	-	-	SPI 通讯接口 SDI 引脚
TCI2	i	s	0	-	-	TimerC 频率输入接口
PT1.1	i/o	s/c	x	-	-	数字输入/输出引脚
INT1	i	s	0	-	-	外部中断源
SCE	i	s	0	-	-	SPI 通讯接口 SCE 引脚
TBI0	i	s	0	-	-	TimerB 启动输入接口
PT1.2	i/o	s/c	x	0	-	数字输入/输出引脚
INT2	i	s	0	0	-	外部中断源
LVDIN	a	a	0	1	-	LVD 外部信号输入引脚
PT1.3	i/o	s/c	x	-	-	数字输入/输出引脚
INT3	i	s	0	-	-	外部中断源
RC	i	s	0	-	-	EUART 通讯接口 RC 引脚
PT1.4	i/o	s/c	x	-	0	数字输入/输出引脚
INT4	i	s	0	-	0	外部中断源
TX	o	c	1	-	1	EUART 通讯接口 TX 引脚
PT1.5	i/o	s/c	x	-	0	数字输入/输出引脚
INT5	i	s	0	-	0	外部中断源
SDO	o	c	1	-	1	SPI 通讯接口 SDO 引脚
TBI1	i	s	0	-	0	TimerB 启动输入接口
PT1.6	i/o	s/c	x	-	0	数字输入/输出引脚
INT6	i	s	0	-	0	外部中断源
SCK	i/o	s	1/0	-	1	SPI 通讯接口 SCK 引脚, Master Mode 时 TC 为"1", Slave Mode 时 TC 为"0"
PT1.7	i/o	s/c	x	-	0	数字输入/输出引脚
INT7	i	s	0	-	0	外部中断源
BZ	o	c	1	--	1	Buzzer 输出引脚

表 7-1 PORT1 功能

7.4. 输入/输出埠 2 ,I/O Port2(以 HY17P58 为例)

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PM[0]	
PT2.0	i/o	s/c	x	-	-	数字输入/输出引脚
INT2.0	i	s	0	-	-	外部中断源
XTO	a	a	0	-	-	外部震荡器引脚
PT2.1	i/o	s/c	x	-	-	数字输入/输出引脚
INT2.1	i	S	0	-	-	外部中断源
XTI	a	A	0	-	-	外部震荡器引脚
PT2.2	i/o	s/c	x	-	0	数字输入/输出引脚
INT2.2	i	s	0	-	0	外部中断源
SCL	i/o	s	1	-	1	I ² C 通讯接口引脚
PT2.3	i/o	s/c	x	-	-	数字输入/输出引脚
INT2.3	i	s	0	-	-	外部中断源
SDA	i/o	s	1	-	-	I ² C 通讯接口引脚
PT2.4	i/o	s/c	x	-	-	数字输入/输出引脚
INT2.4	i	s	0	-	-	外部中断源
RC2	i/o	s	0	-	-	EUART2 通讯接口接口
PT2.5	i/o	s/c	x	-	0	数字输入/输出引脚
INT2.5	i	s	0	-	0	外部中断源
TX2	o	s	1	-	1	EUART2 通讯接口接口
PT2.6	i/o	s/c	x	-	-	数字输入/输出引脚
INT2.6	i	s	0	-	-	外部中断源
TCI1	i	s	0	-	-	TimerC 频率输入接口
PT2.7	i/o	s/c	x	-	-	数字输入/输出引脚
INT2.7	i	s	0	-	-	外部中断源
TBI2	i	s	0	-	-	TimerB 启动输入接口

表 7-2 PORT2 功能

7.1. 输入/输出埠 4 ,I/O Port4(以 HY17P58 为例)

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PM[0]	
PT4.4	i	s	x	0	-	数字输入引脚
AI4	a	a	0	1	-	模拟输入通道
RFIN	a	a	0	1	-	RF1/RF2 参考电阻输入端
OPO2	a	a	0	1	-	R2ROP2 输出
PT4.5	i	s	x	0	-	数字输入引脚
AI5	a	a	0	1	-	模拟输入通道
RF1OUT	a	a	0	1	-	RF1 参考电阻输出端
PT4.6	i	s	x	0	-	数字输入引脚
AI6	a	a	0	1	-	模拟输入通道
RF2OUT	a	a	0	1	-	RF2 参考电阻输出端
PT4.7	i	s	x	0	-	数字输入引脚
AI7	a	a	0	1	-	模拟输入通道
FIR	a	a	0	1	-	体脂测量输入接口(右脚掌)

表 7-3 PORT4 功能

7.2. 输入/输出埠 5 ,I/O Port5(以 HY17P58 为例)

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PM[0]	
PT5.0	i	s	-	0	-	数字输入引脚
AI8	a	a	-	1	-	模拟输入通道
FVL	a	a	-	1	-	体脂测量输入接口(左脚跟)
LNOP2O	a	a	-	1	-	LNOP2 输出
PT5.1	i	s	-	0	-	数字输入引脚
AI9	a	a	-	1	-	模拟输入通道
FIL	a	a	-	1	-	体脂测量输入接口(左脚掌)
PT5.2	i	s	-	0	-	数字输入引脚
AI10	a	a	-	1	-	模拟输入通道
FVR	a	a	-	1	-	体脂测量输入接口(右脚跟)
PT5.3	i	s	-	0	-	数字输入引脚
AI11	a	a	-	1	-	模拟输入通道
PEAKO	a	a	-	1	-	R2ROP2 输出(Peak Hold 接口)

表 7-4 PORT5 功能

7.3. 输入/输出埠 6 ,I/O Port6

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PU[0]	
PT6.0	i/o	s/c	x	0	x	数字输入/输出引脚
COM0	o	a	0	-	0	LCD Common 输出
PT6.1	i/o	s/c	x	0	x	数字输入/输出引脚
COM1	o	a	0	-	0	LCD Common 输出
PT6.2	i/o	s/c	x	0	x	数字输入/输出引脚
COM2	o	a	0	-	0	LCD Common 输出
PT6.3	i/o	s/c	x	0	x	数字输入/输出引脚
COM3	o	a	0	-	0	LCD Common 输出
PT6.4	i/o	s/c	x	0	x	数字输入/输出引脚
SEG2	o	a	0	-	0	LCD Segment 输出
PT6.5	i/o	s/c	x	0	x	数字输入/输出引脚
SEG3	o	a	0	-	0	LCD Segment 输出
PT6.6	i/o	s/c	x	0	x	数字输入/输出引脚
SEG4	o	a	0	-	0	LCD Segment 输出
PT6.7	i/o	s/c	x	0	x	数字输入/输出引脚
SEG5	o	a	0	-	0	LCD Segment 输出

表 7-5 PORT6 功能

7.4. 输入/输出埠 7 ,I/O Port7

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PU[0]	
PT7.0	i/o	s/c	x	0	x-	数字输入/输出引脚
SEG6	o	a	0	-	0	LCD Segment 输出
PT7.1	i/o	s/c	x	0	x	数字输入/输出引脚
SEG7	o	a	0	-	0	LCD Segment 输出
PT7.2	i/o	s/c	x	0	x	数字输入/输出引脚
SEG8	o	a	0	-	0	LCD Segment 输出
PT7.3	i/o	s/c	x	0	x	数字输入/输出引脚
SEG9	o	a	0	-	0	LCD Segment 输出
PT7.4	i/o	s/c	x	0	x	数字输入/输出引脚
SEG10	o	a	0	-	0	LCD Segment 输出
PT7.5	i/o	s/c	x	0	x	数字输入/输出引脚
SEG11	o	a	0	-	0	LCD Segment 输出
PT7.6	i/o	s/c	x	0	x	数字输入/输出引脚
SEG12	o	a	0	-	0	LCD Segment 输出
PT7.7	i/o	s/c	x	0	x	数字输入/输出引脚
SEG13	o	a	0	-	0	LCD Segment 输出

表 7-6 PORT7 功能

7.5. 输入/输出埠 8 ,I/O Port8

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PI[0]	
PT8.0	i/o	s/c	x	0	x	数字输入/输出引脚
SEG14	o	a	0	-	0	LCD Segment 输出
PT8.1	i/o	s/c	x	0	x	数字输入/输出引脚
SEG15	o	a	0	-	0	LCD Segment 输出
PT8.2	i/o	s/c	x	0	x	数字输入/输出引脚
SEG16	o	a	0	-	0	LCD Segment 输出
PT8.3	i/o	s/c	x	0	x	数字输入/输出引脚
SEG17	o	a	0	-	0	LCD Segment 输出
PT8.4	i/o	s/c	x	0	x	数字输入/输出引脚
SEG18	o	a	0	-	0	LCD Segment 输出
PT8.5	i/o	s/c	x	0	x	数字输入/输出引脚
SEG19	o	a	0	-	0	LCD Segment 输出
PT8.6	i/o	s/c	x	0	x	数字输入/输出引脚
SEG20	o	a	0	-	0	LCD Segment 输出
PT8.7	i/o	s/c	x	0	x	数字输入/输出引脚
SEG21	o	a	0	-	0	LCD Segment 输出

表 7-7 PORT8 功能

7.6. 输入/输出埠 9 ,I/O Port9

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PU[1:0]	
PT9.0	i/o	s/c	x	0	x	数字输入/输出引脚
SEG22	o	a	0	-	0	LCD Segment 输出
PT9.1	i/o	s/c	x	0	x	数字输入/输出引脚
SEG23	o	a	0	-	0	LCD Segment 输出
PT9.2	i/o	s/c	x	0	x	数字输入/输出引脚
SEG24	o	a	0	-	0	LCD Segment 输出
PT9.3	i/o	s/c	x	0	x	数字输入/输出引脚
SEG25	o	a	0	-	0	LCD Segment 输出
PT9.4	i/o	s/c	x	0	x	数字输入/输出引脚
SEG26	o	a	0	-	0	LCD Segment 输出
PT9.5	i/o	s/c	x	0	x	数字输入/输出引脚
SEG27	o	a	0	-	0	LCD Segment 输出
PT9.6	i/o	s/c	x	0	x	数字输入/输出引脚
SEG28	o	a	0	-	0	LCD Segment 输出
PT9.7	i/o	s/c	x	0	x	数字输入/输出引脚
SEG29	o	a	0	-	0	LCD Segment 输出

表 7-8 PORT9 功能

7.7. 输入/输出埠 10 ,I/O Port10

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置			描述
	型式	缓冲	TC[0]	DA[0]	PU[1:0]	
PT10.0	i/o	s/c	x	0	x	数字输入/输出引脚
SEG30	o	a	0	-	0	LCD Segment 输出
PT10.1	i/o	s/c	x	0	x	数字输入/输出引脚
SEG31	o	a	0	-	0	LCD Segment 输出
PT10.2	i/o	s/c	x	0	x	数字输入/输出引脚
SEG32	o	a	0	-	0	LCD Segment 输出
PT10.3	i/o	s/c	x	0	x	数字输入/输出引脚
SEG33	o	a	0	-	0	LCD Segment 输出
PT10.4	i/o	s/c	x	0	x	数字输入/输出引脚
SEG34	o	a	0	-	0	LCD Segment 输出
PT10.5	i/o	s/c	x	0	x	数字输入/输出引脚
SEG35	o	a	0	-	0	LCD Segment 输出
PT10.6	i/o	s/c	x	0	x	数字输入/输出引脚
SEG36	o	a	0	-	0	LCD Segment 输出
PT10.7	i/o	s/c	x	0	x	数字输入/输出引脚
SEG37	o	a	0	-	0	LCD Segment 输出

表 7-9 PORT10 功能

7.8. 输入/输出埠 11 ,I/O Port11

“i”：输入,“o”：输出,“a”：模拟,“s”：史密斯触发,“c”：cmos i/o,“x”：未定义,“p”：电源,

引脚名称	设计		寄存器设置				描述
	型式	缓冲	TC[0]	DA[0]	PU[1:0]	PM[0]	
PT11.0	i/o	s/c	x	0	x	-	数字输入/输出引脚
SEG38	o	a	0	-	0	-	LCD Segment 输出
PT11.1	i/o	s/c	x	0	x	-	数字输入/输出引脚
SEG39	o	a	0	-	0	-	LCD Segment 输出
PT11.2	i/o	s/c	x	0	x	-	数字输入/输出引脚
SEG40	o	a	0	-	0	0	LCD Segment 输出
PWM1	o	c	1	-		1	PWM1 输出接口
PT11.3	i/o	s/c	x	0	x	-	数字输入/输出引脚
SEG41	o	a	0	-	0	0	LCD Segment 输出
PWM0	o	c	1	-		1	PWM0 输出接口

表 7-10 PORT11 功能

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



7.9. 寄存器说明-PORT

“.”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1

“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition

File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	-	ADIE	WDTIE	TB1IE	TMAIE	E1IE	E0IE	0000 0000	0uuu uuuu	*****
INTE1	-						E3IE	E2IE	0000 0000	uuuu uuuu	*****
INTF0	-	-	ADIF	WDTIF	TB1IF	TMAIF	E1IF	E0IF	.000 0000	.uuu uuuu	*****
INTF1	-						E3IF	E2IF	0000 0000	uuuu uuuu	***r,r**
WDTCN	ENBZ	BZS	BZ[1:0]		ENWDT	DWD[2:0]			0000 0000	uuuu \$000	-.***rw1,***
PT1	PT1.7	PT1.6	PT1.5	PT1.4	PT1.3	PT1.2	PT1.1	PT1.0	xxxx xxxx	xxxx xxxx	*****
TRISC1	TC1.7	TC1.6	TC1.5	TC1.4	TC1.3	TC1.2	TC1.1	TC1.0	0000 0000	uuuu uuuu	*****
PT1DA	-	-	-	-	-	DA1.2	-	-	0000 0000	uuuu uuuu	*****
PT1PU	PU1.7	PU1.6	PU1.5	PU1.4	PU1.3	PU1.2	PU1.1	PU1.0	0000 0000	uuuu uuuu	*****
PT1M1	-	-	-	-	INTEG1[1:0]		INTEG0[1:0]		0000 0000	uuuu uuuu	*****
PT1INT	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3		-		0000 0000	uuuu uuuu	*****
PT1INTE	INTE1.7	INTE1.6	INTE1.5	INTE1.4	-	-	-	-	0000 0000	uuuu uuuu	*****
PT1INTF	INTF1.7	INTF1.6	INTF1.5	INTF1.4	-		-		0000 0000	uuuu uuuu	*****
PT2	PT2.7	PT2.6	PT2.5	PT2.4	PT2.3	PT2.2	PT2.1	PT2.0	0000 0000	uuuu uuuu	*****
TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	uuuu uuuu	*****
PT2DA	DA2.7	DA2.6	DA2.5	DA2.4	DA2.3	DA2.2	DA2.1	DA2.0	1111 1111	uuuu uuuu	*****
PT2PU	PU2.7	PU2.6	PU2.5	PU2.4	PU2.3	PU2.2	PU2.1	PU2.0	0000 0000	uuuu uuuu	*****
PT2INT	INTG2.7	INTG2.6	INTG2.5	INTG2.4	INTG2.3	INTG2.2	INTG2.1	INTG2.0	0000 0000	uuuu uuuu	*****
PT2INTE	INTE2.7	INTE2.6	INTE2.5	INTE2.4	INTE2.3	INTE2.2	INTE2.1	INTE2.0	0000 0000	uuuu uuuu	
PT2INTF	INTF2.7	INTF2.6	INTF2.5	INTF2.4	INTF2.3	INTF2.2	INTF2.1	INTF2.0	0000 0000	uuuu uuuu	*****
PT4	PT4.7	PT4.6	PT4.5	PT4.4	-	-	-	-	xxxx xxxx	xxxx xxxx	*****
TRISC4	TC4.7	TC4.6	TC4.5	TC4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
PT4DA	DA4.7	DA4.6	DA4.5	DA4.4	-	-	-	-	1111 1111	uuuu uuuu	*****
PT4PU	PU4.7	PU4.6	PU4.5	PU4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
PT4INT	INTG4.7	INTG4.6	INTG4.5	INTG4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
PT4INTE	INTE4.7	INTE4.6	INTE4.5	INTE4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
PT4INTF	INTF4.7	INTF4.6	INTF4.5	INTF4.4	-	-	-	-	0000 0000	uuuu uuuu	*****
PT5	-	-	PT5.5	PT5.4	PT5.3	PT5.2	PT5.1	PT5.0	xxxx xxxx	xxxx xxxx	*****
TRISC5	TC5.7	TC5.6	TC5.5	TC5.4	TC5.3	TC5.2	TC5.1	TC5.0	0000 0000	uuuu uuuu	*****
PT5DA	-	-	DA5.5	DA5.4	DA5.3	DA5.2	DA5.1	DA5.0	1111 1111	uuuu uuuu	*****
PT5PU	-	-	PU5.5	PU5.4	PU5.3	PU5.2	PU5.1	PU5.0	0000 0000	uuuu uuuu	*****
PT5INT	-	-	-	-	INTG5.3	INTG5.2	INTG5.1	INTG5.0	0000 0000	uuuu uuuu	*****
PT5INTE	-	-	-	-	INTE5.3	INTE5.2	INTE5.1	INTE5.0	0000 0000	uuuu uuuu	*****
PT5INTF	-	-	-	-	INTF5.3	INTF5.2	INTF5.1	INTF5.0	0000 0000	uuuu uuuu	*****

表 7-11 PORT 控制寄存器

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition

File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
PT6	PT6.7	PT6.6	PT6.5	PT6.4	PT6.3	PT6.2	PT6.1	PT6.0	xxxx xxxx	uuuu uuuu	*****
TRISC6	TC6.7	TC6.6	TC6.5	TC6.4	TC6.3	TC6.2	TC6.1	TC6.0	0000 0000	uuuu uuuu	*****
PT6DA	DA6.7	DA6.6	DA6.5	DA6.4	DA6.3	DA6.2	DA6.1	DA6.0	1111 1111	uuuu uuuu	*****
PT6PU	PU6.7	PU6.6	PU6.5	PU6.4	PU6.3	PU6.2	PU6.1	PU6.0	0000 0000	uuuu uuuu	*****
PT7	PT7.7	PT7.6	PT7.5	PT7.4	PT7.3	PT7.2	PT7.1	PT7.0	xxxx xxxx	uuuu uuuu	*****
TRISC7	TC7.7	TC7.6	TC7.5	TC7.4	TC7.3	TC7.2	TC7.1	TC7.0	0000 0000	uuuu uuuu	*****
PT7DA	DA7.7	DA7.6	DA7.5	DA7.4	DA7.3	DA7.2	DA7.1	DA7.0	1111 1111	uuuu uuuu	*****
PT7PU	PU7.7	PU7.6	PU7.5	PU7.4	PU7.3	PU7.2	PU7.1	PU7.0	0000 0000	uuuu uuuu	*****
PT8	PT8.7	PT8.6	PT8.5	PT8.4	PT8.3	PT8.2	PT8.1	PT8.0	xxxx xxxx	uuuu uuuu	*****
TRISC8	TC8.7	TC8.6	TC8.5	TC8.4	TC8.3	TC8.2	TC8.1	TC8.0	0000 0000	uuuu uuuu	*****
PT8DA	DA8.7	DA8.6	DA8.5	DA8.4	DA8.3	DA8.2	DA8.1	DA8.0	1111 1111	uuuu uuuu	*****
PT8PU	PU8.7	PU8.6	PU8.5	PU8.4	PU8.3	PU8.2	PU8.1	PU8.0	0000 0000	uuuu uuuu	*****
PT9	PT9.7	PT9.6	PT9.5	PT9.4	PT9.3	PT9.2	PT9.1	PT9.0	xxxx xxxx	uuuu uuuu	*****
TRISC9	TC9.7	TC9.6	TC9.5	TC9.4	TC9.3	TC9.2	TC9.1	TC9.0	0000 0000	uuuu uuuu	*****
PT9DA	DA9.7	DA9.6	DA9.5	DA9.4	DA9.3	DA9.2	DA9.1	DA9.0	1111 1111	uuuu uuuu	*****
PT9PU	PU9.7	PU9.6	PU9.5	PU9.4	PU9.3	PU9.2	PU9.1	PU9.0	0000 0000	uuuu uuuu	*****
PT10	PT10.7	PT10.6	PT10.5	PT10.4	PT10.3	PT10.2	PT10.1	PT10.0	xxxx xxxx	uuuu uuuu	*****
TRISC10	TC10.7	TC10.6	TC10.5	TC10.4	TC10.3	TC10.2	TC10.1	TC10.0	0000 0000	uuuu uuuu	*****
PT10DA	DA10.7	DA10.6	DA10.5	DA10.4	DA10.3	DA10.2	DA10.1	DA10.0	1111 1111	uuuu uuuu	*****
PT10PU	PU10.7	PU10.6	PU10.5	PU10.4	PU10.3	PU10.2	PU10.1	PU10.0	0000 0000	uuuu uuuu	*****
PT11	-	-	-	-	PT11.3	PT11.2	PT11.1	PT11.0	xxxx xxxx	uuuu uuuu	*****
TRISC11	-	-	-	-	TC11.3	TC11.2	TC11.1	TC11.0	0000 0000	uuuu uuuu	*****
PT11DA	-	-	-	-	DA11.3	DA11.2	DA11.1	DA11.0	0000 1111	uuuu uuuu	*****
PT11PU	-	-	-	-	PU11.3	PU11.2	PU11.1	PU11.0	0000 0000	uuuu uuuu	*****

表 7-12 PORT 控制寄存器(续)

INTE0/INTE1/PT1INTE/PT2INTE/PT4INTE/PT5INTE: 详见中断 Interrupt 章节

/INTF0/INTF1/PT1INTF/PT2INTF/PT4INTF/PT5INTF: 详见中断 Interrupt 章节

WDTCN[7:0]看门狗控制寄存器

位	名称	描述										
Bit7	ENBZ	蜂鸣器 BZ 启用与关闭控制器 <0>关闭 <1>启用										
Bit6	BZS	蜂鸣器 BZ 工作频率选择器 <0>LS_CK <1>LPC_CK										
Bit5~4	DBZ[1:0]	蜂鸣器输出频率控制器 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>DBZ[1:0]</th> <th>Pre-scale</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>BZ_CK ÷ 2</td> </tr> <tr> <td>01</td> <td>BZ_CK ÷ 4</td> </tr> <tr> <td>10</td> <td>BZ_CK ÷ 8</td> </tr> <tr> <td>11</td> <td>BZ_CK ÷ 16</td> </tr> </tbody> </table>	DBZ[1:0]	Pre-scale	00	BZ_CK ÷ 2	01	BZ_CK ÷ 4	10	BZ_CK ÷ 8	11	BZ_CK ÷ 16
DBZ[1:0]	Pre-scale											
00	BZ_CK ÷ 2											
01	BZ_CK ÷ 4											
10	BZ_CK ÷ 8											
11	BZ_CK ÷ 16											

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

PT1: PT1 引脚输入状态与输出控制寄存器

位	名称	描述		
Bit7~0	PT1.x	PT1.x 引脚的输入状态与输出控制器, $0 \leq x \leq 7$		
		PT1.x	当 TC1x 设置<0>	当 TC1x 设置<1>
		0	PT1.x 输入为低电位(L)	PT1.x 输出低电位(L)
		1	PT1.x 输入为高电位(H)	PT1.x 输出高电位(H)

TRISC1: PT1 引脚特性控制寄存器

位	名称	描述
Bit7~0	TC1.x	PT1.x 引脚输出特性控制器, $0 \leq x \leq 7$ <0> 关闭输出功能, 引脚只为输入特性 <1> 启用输出功能, 引脚为输出/输入特性

PT1DA: PT1 模拟输入控制寄存器

位	名称	描述
Bit7~0	DA1.2	PT1.2 模拟输入控制器 <0> 关闭 <1> 启用

PT1PU: PT1 引脚上拉电阻控制寄存器

位	名称	描述
Bit7~0	PU1.x	PT1.x 引脚上拉电阻控制器, $0 \leq x \leq 7$ <0>关闭 <1>启用

PT1M1: 数字输出模式选择寄存器 1

位	名称	描述	
Bit3~2	INTEG1[1:0]	PT1.x 中断信号产生条件 ($0 \leq x \leq 1$)	
Bit1~0	INTEG0[1:0]	INTEGx[1:0]	中断信号产生条件
		00	下降缘 (1→0)
		01	上升缘 (0→1)
		10	电位转态 (0→1 或 1→0)
		11	电位转态 (0→1 或 1→0)

PT1INT: I/O 中断触发产生条件

位	名称	描述
Bit7~2	INTEGx	中断触发产生条件 ($2 \leq x \leq 7$) <0> 下降缘 (1→0)

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

		<1> 上升缘 (0→1)
--	--	---------------

PT2: PT2 引脚输入状态与输出控制寄存器

位	名称	描述									
Bit7~0	PT2.x	PT2.x 引脚的输入状态与输出控制器, $0 \leq x \leq 7$									
		<table border="1"><thead><tr><th>PT2.x</th><th>当 TC2x 设置<0></th><th>当 TC2x 设置<1></th></tr></thead><tbody><tr><td>0</td><td>PT2.x 输入为低电位(L)</td><td>PT2.x 输出低电位(L)</td></tr><tr><td>1</td><td>PT2.x 输入为高电位(H)</td><td>PT2.x 输出高电位(H)</td></tr></tbody></table>	PT2.x	当 TC2x 设置<0>	当 TC2x 设置<1>	0	PT2.x 输入为低电位(L)	PT2.x 输出低电位(L)	1	PT2.x 输入为高电位(H)	PT2.x 输出高电位(H)
		PT2.x	当 TC2x 设置<0>	当 TC2x 设置<1>							
		0	PT2.x 输入为低电位(L)	PT2.x 输出低电位(L)							
1	PT2.x 输入为高电位(H)	PT2.x 输出高电位(H)									
0	PT2.x 输入为低电位(L)	PT2.x 输出低电位(L)									
1	PT2.x 输入为高电位(H)	PT2.x 输出高电位(H)									

TRISC2: PT2 引脚特性控制寄存器

位	名称	描述
Bit7~0	TC2.x	PT2.x 引脚输出特性控制器, $0 \leq x \leq 7$ <0> 关闭输出功能, 引脚只为输入特性 <1> 启用输出功能, 引脚为输出/输入特性

PT2PU: PT2 引脚上拉电阻控制寄存器

位	名称	描述
Bit7~0	PU2.x	PT2.x 引脚上拉电阻控制器, $0 \leq x \leq 7$ <0>关闭 <1>启用

PT2INT: I/O 中断触发产生条件

位	名称	描述
Bit7~0	INTEGx	中断触发产生条件 ($0 \leq x \leq 7$) <0> 下降缘 (1→0) <1> 上升缘 (0→1)

PT4: PT4 引脚输入状态与输出控制寄存器

位	名称	描述									
Bit7~4	PT4.x	PT4.x 引脚的输入状态与输出控制器, $4 \leq x \leq 7$									
		<table border="1"><thead><tr><th>PT4.x</th><th>当 TC4.x 设置<0></th><th>当 TC4.x 设置<1></th></tr></thead><tbody><tr><td>0 (预设)</td><td>PT4.x 输入为低电位(L)</td><td>PT4.x 输出低电位(L)</td></tr><tr><td>1</td><td>PT4.x 输入为高电位(H)</td><td>PT4.x 输出高电位(H)</td></tr></tbody></table>	PT4.x	当 TC4.x 设置<0>	当 TC4.x 设置<1>	0 (预设)	PT4.x 输入为低电位(L)	PT4.x 输出低电位(L)	1	PT4.x 输入为高电位(H)	PT4.x 输出高电位(H)
		PT4.x	当 TC4.x 设置<0>	当 TC4.x 设置<1>							
		0 (预设)	PT4.x 输入为低电位(L)	PT4.x 输出低电位(L)							
1	PT4.x 输入为高电位(H)	PT4.x 输出高电位(H)									
0 (预设)	PT4.x 输入为低电位(L)	PT4.x 输出低电位(L)									
1	PT4.x 输入为高电位(H)	PT4.x 输出高电位(H)									

TRISC4: PT4 引脚特性控制寄存器

位	名称	描述
Bit7~4	TC4.x	PT4.x 引脚输出特性控制器, $4 \leq x \leq 7$ <0>关闭输出功能, 引脚只为输入特性(预设) <1>启用输出功能, 引脚为输出/输入特性

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

PT4DA: PT4 模拟输入控制寄存器

位	名称	描述
Bit7~4	DA4.x	PT4.x 模拟输入控制器, $4 \leq x \leq 7$ <0>关闭 <1>启用(预设)

PT4PU: PT4 引脚上拉电阻控制寄存器

位	名称	描述
Bit7~4	PU4.x	PT4.x 引脚上拉电阻控制器, $4 \leq x \leq 7$ <0>关闭(预设) <1>启用

PT6: PT6 引脚输入状态与输出控制寄存器

位	名称	描述									
Bit7~0	PT6.x	PT6.x 引脚的输入状态与输出控制器, $0 \leq x \leq 7$									
		<table border="1"><thead><tr><th>PT6.x</th><th>当 TC6.x 设置<0></th><th>当 TC6.x 设置<1></th></tr></thead><tbody><tr><td>0 (预设)</td><td>-</td><td>PT6.x 输出低电位(L)</td></tr><tr><td>1</td><td>-</td><td>PT6.x 输出高电位(H)</td></tr></tbody></table>	PT6.x	当 TC6.x 设置<0>	当 TC6.x 设置<1>	0 (预设)	-	PT6.x 输出低电位(L)	1	-	PT6.x 输出高电位(H)
		PT6.x	当 TC6.x 设置<0>	当 TC6.x 设置<1>							
0 (预设)	-	PT6.x 输出低电位(L)									
1	-	PT6.x 输出高电位(H)									

TRISC6: PT6 引脚特性控制寄存器

位	名称	描述
Bit7~0	TC6.x	PT6.x 引脚输出特性控制器, $0 \leq x \leq 7$ <0>关闭输出功能, 引脚只为输入特性(预设) <1>启用输出功能, 引脚为输出/输入特性

PT6DA: PT6 模拟输入控制寄存器

位	名称	描述
Bit7~0	DA6.x	PT6.x 模拟输入控制器, $0 \leq x \leq 7$ <0> 关闭, 为数字输出/输入模式 <1> 启用(预设)

PT6PU: PT6 引脚上拉电阻控制寄存器

位	名称	描述
Bit7~0	PU6.x	PT6.x 引脚上拉电阻控制器, $0 \leq x \leq 7$ <0>关闭 <1>启用

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

PT7: PT7 引脚输入状态与输出控制寄存器

位	名称	描述									
Bit7~0	PT7.x	PT7.x 引脚的输入状态与输出控制器, $0 \leq x \leq 7$									
		<table border="1"><thead><tr><th>PT7.x</th><th>当 TC7.x 设置<0></th><th>当 TC7.x 设置<1></th></tr></thead><tbody><tr><td>0 (预设)</td><td>-</td><td>PT7.x 输出低电位(L)</td></tr><tr><td>1</td><td>-</td><td>PT7.x 输出高电位(H)</td></tr></tbody></table>	PT7.x	当 TC7.x 设置<0>	当 TC7.x 设置<1>	0 (预设)	-	PT7.x 输出低电位(L)	1	-	PT7.x 输出高电位(H)
		PT7.x	当 TC7.x 设置<0>	当 TC7.x 设置<1>							
		0 (预设)	-	PT7.x 输出低电位(L)							
1	-	PT7.x 输出高电位(H)									
0 (预设)	-	PT7.x 输出低电位(L)									
1	-	PT7.x 输出高电位(H)									

TRISC7: PT7 引脚特性控制寄存器

位	名称	描述
Bit7~0	TC7.x	PT7.x 引脚输出特性控制器, $0 \leq x \leq 7$ <0>关闭输出功能, 引脚只为输入特性(预设) <1>启用输出功能, 引脚为输出/输入特性

PT7DA: PT7 模拟输入控制寄存器

位	名称	描述
Bit7~0	DA7.x	PT7.x 模拟输入控制器, $0 \leq x \leq 7$ <0> 关闭, 为数字输出/输入模式 <1> 启用(预设)

PT7PU: PT7 引脚上拉电阻控制寄存器

位	名称	描述
Bit7~0	PU7.x	PT7.x 引脚上拉电阻控制器, $0 \leq x \leq 7$ <0>关闭 <1>启用

PT8: PT8 引脚输入状态与输出控制寄存器

位	名称	描述									
Bit7~0	PT8.x	PT8.x 引脚的输入状态与输出控制器, $0 \leq x \leq 7$									
		<table border="1"><thead><tr><th>PT8.x</th><th>当 TC8.x 设置<0></th><th>当 TC8.x 设置<1></th></tr></thead><tbody><tr><td>0 (预设)</td><td>-</td><td>PT8.x 输出低电位(L)</td></tr><tr><td>1</td><td>-</td><td>PT8.x 输出高电位(H)</td></tr></tbody></table>	PT8.x	当 TC8.x 设置<0>	当 TC8.x 设置<1>	0 (预设)	-	PT8.x 输出低电位(L)	1	-	PT8.x 输出高电位(H)
		PT8.x	当 TC8.x 设置<0>	当 TC8.x 设置<1>							
		0 (预设)	-	PT8.x 输出低电位(L)							
1	-	PT8.x 输出高电位(H)									
0 (预设)	-	PT8.x 输出低电位(L)									
1	-	PT8.x 输出高电位(H)									

TRISC8: PT8 引脚特性控制寄存器

位	名称	描述
Bit7~0	TC8.x	PT8.x 引脚输出特性控制器, $0 \leq x \leq 7$ <0>关闭输出功能, 引脚只为输入特性(预设) <1>启用输出功能, 引脚为输出/输入特性

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

PT8DA: PT8 模拟输入控制寄存器

位	名称	描述
Bit7~0	DA8.x	PT8.x 模拟输入控制器, $0 \leq x \leq 7$ <0> 关闭, 为数字输出/输入模式 <1> 启用(预设)

PT8PU: PT8 引脚上拉电阻控制寄存器

位	名称	描述
Bit7~0	PU8.x	PT8.x 引脚上拉电阻控制器, $0 \leq x \leq 7$ <0>关闭 <1>启用

PT9: PT9 引脚输入状态与输出控制寄存器

位	名称	描述									
Bit7~0	PT9.x	PT9.x 引脚的输入状态与输出控制器, $0 \leq x \leq 7$ <table border="1"><thead><tr><th>PT7.x</th><th>当 TC9.x 设置<0></th><th>当 TC9.x 设置<1></th></tr></thead><tbody><tr><td>0 (预设)</td><td>-</td><td>PT7.x 输出低电位(L)</td></tr><tr><td>1</td><td>-</td><td>PT7.x 输出高电位(H)</td></tr></tbody></table>	PT7.x	当 TC9.x 设置<0>	当 TC9.x 设置<1>	0 (预设)	-	PT7.x 输出低电位(L)	1	-	PT7.x 输出高电位(H)
PT7.x	当 TC9.x 设置<0>	当 TC9.x 设置<1>									
0 (预设)	-	PT7.x 输出低电位(L)									
1	-	PT7.x 输出高电位(H)									

TRISC9: PT9 引脚特性控制寄存器

位	名称	描述
Bit7~0	TC9.x	PT9.x 引脚输出特性控制器, $0 \leq x \leq 7$ <0>关闭输出功能, 引脚只为输入特性(预设) <1>启用输出功能, 引脚为输出/输入特性

PT9DA: PT9 模拟输入控制寄存器

位	名称	描述
Bit7~0	DA9.x	PT9.x 模拟输入控制器, $0 \leq x \leq 7$ <0> 关闭, 为数字输出/输入模式 <1> 启用(预设)

PT9PU: PT9 引脚上拉电阻控制寄存器

位	名称	描述
Bit7~0	PU9.x	PT9.x 引脚上拉电阻控制器, $0 \leq x \leq 7$ <0>关闭 <1>启用

PT10: PT10 引脚输入状态与输出控制寄存器

位	名称	描述
---	----	----

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

Bit7~0	PT10.x	PT10.x 引脚的输入状态与输出控制器, $0 \leq x \leq 7$		
		PT10.x	当 TC10.x 设置<0>	当 TC10.x 设置<1>
		0 (预设)	-	PT10.x 输出低电位(L)
		1	-	PT10.x 输出高电位(H)

TRISC10: PT10 引脚特性控制寄存器

位	名称	描述
Bit7~0	TC10.x	PT10.x 引脚输出特性控制器, $0 \leq x \leq 7$ <0>关闭输出功能, 引脚只为输入特性(预设) <1>启用输出功能, 引脚为输出/输入特性

PT10DA: PT10 模拟输入控制寄存器

位	名称	描述
Bit7~0	DA10.x	PT10.x 模拟输入控制器, $0 \leq x \leq 7$ <0> 关闭, 为数字输出/输入模式 <1> 启用(预设)

PT10PU: PT10 引脚上拉电阻控制寄存器

位	名称	描述
Bit7~0	PU10.x	PT10.x 引脚上拉电阻控制器, $0 \leq x \leq 7$ <0>关闭 <1>启用

PT11: PT11 引脚输入状态与输出控制寄存器

Bit3~0	PT11.x	PT11.x 引脚的输入状态与输出控制器, $0 \leq x \leq 3$		
		PT11.x	当 TC11.x 设置<0>	当 TC11.x 设置<1>
		0 (预设)	-	PT11.x 输出低电位(L)
		1	-	PT11.x 输出高电位(H)

TRISC11: PT11 引脚特性控制寄存器

位	名称	描述
Bit3~0	TC11.x	PT11.x 引脚输出特性控制器, $0 \leq x \leq 3$ <0>关闭输出功能, 引脚只为输入特性(预设) <1>启用输出功能, 引脚为输出/输入特性

PT11DA: PT11 模拟输入控制寄存器

位	名称	描述
---	----	----

HY17S58 Emulate Chip User' Guide

Embedded $\Sigma\Delta$ ADC
8-Bit RISC-like Mixed Signal Microcontroller

Bit3~0	DA11.x	PT11.x 模拟输入控制器, $0 \leq x \leq 3$ <0> 关闭, 为数字输出/输入模式 <1> 启用(预设)
--------	--------	---

PT11PU: PT11 引脚上拉电阻控制寄存器

位	名称	描述
Bit3~0	PU11.x	PT11.x 引脚上拉电阻控制器, $0 \leq x \leq 3$ <0>关闭 <1>启用

8. 定电流输入/输出埠, I/O (HY17P48 only)

定电流的 I/O 仅有 HY17P48 支持，其余 IC 型号没有该功能。

定电流的 I/O，主要功能为驱动 LED，透过软件方式扫描完成 LED 显示功能。

定电流 I/O 有以下特色。

- 一共支持 8 I/O 最大可驱动 LED 点数为 56 点。
- 有 3-bit 电流控制器，可调整电流支持 2mA~15mA
- 不需要限流电阻
- LED 亮度不随 VDD 电压、温度、点亮点数多寡而改变

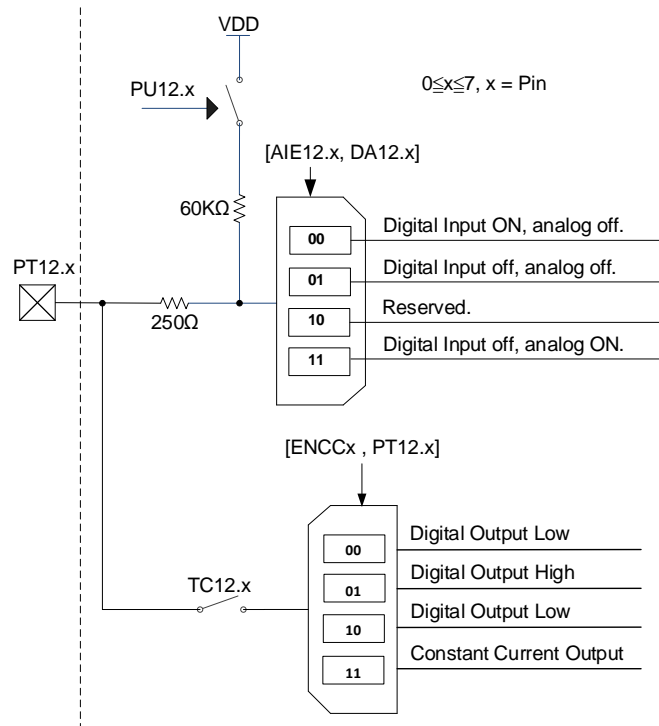


图 8-1 I/O PORT12 架构方块图

8.1. 定电流模使用说明

8.1.1. 定电流模式初始设定

设定 ENCCMode 寄存器，ENCC7~ENCC0 分别对应 PT12.7~PT12.0 引脚，置 1 为开启定电流模式。
透过 CCLevel[2:0] 设定开启定电流脚位的电流大小。

在 LED 的点亮，还需要将 PT12 设定成 IO 的输出模式，并设置成输出高电平。但 IO 模式上的切换与 IO 的输出状况，还需视 LED 点亮的需求做调整。

8.2. 寄存器说明-CC

“.”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
CCNT	-	-	-	-	-	CCLevel[2:0]			xxxx x000	uuuu uuuu	*,*,*,*,*,*
ENCCMode	ENCC7	ENCC6	ENCC5	ENCC4	ENCC3	ENCC2	ENCC1	ENCC0	0000 0000	uuuu uuuu	*,*,*,*,*,*

表 8-1 定电流控制寄存器

CCNT: CCNT 控制寄存器

位	名称	描述																		
Bit2~0	CCLevel	定电流大小控制																		
		<table border="1"> <thead> <tr> <th>CCNT[2:0]</th> <th>电流大小</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>2mA</td> </tr> <tr> <td>001</td> <td>3mA</td> </tr> <tr> <td>010</td> <td>5mA</td> </tr> <tr> <td>011</td> <td>7mA</td> </tr> <tr> <td>100</td> <td>9mA</td> </tr> <tr> <td>101</td> <td>11mA</td> </tr> <tr> <td>110</td> <td>13mA</td> </tr> <tr> <td>111</td> <td>15mA</td> </tr> </tbody> </table>	CCNT[2:0]	电流大小	000	2mA	001	3mA	010	5mA	011	7mA	100	9mA	101	11mA	110	13mA	111	15mA
		CCNT[2:0]	电流大小																	
		000	2mA																	
		001	3mA																	
		010	5mA																	
		011	7mA																	
		100	9mA																	
		101	11mA																	
110	13mA																			
111	15mA																			

ENCCMode: ENCCMode 控制寄存器

位	名称	描述
Bit7~0	ENCCx	定电流控制开关 <0> 关闭 <1> 开启

9. 低电压检测, Low Voltage Detect

低电压检测 LVD 具有侦测工作电压或是外部输入电压的功能，其可以让用户准确的决定所监测的电压在多低的电压准位产生低电压事件。

LVD 具有以下功能。

- VDD 工作电压与外部输入电压侦测功能。
- 可设置十四段工作电压的低电压检测点。
- 简易外部输入电压比较功能。

LVD 相关寄存器摘要：

LVDCN PWRSP[0], VLDX[3:0], LVDO[0]

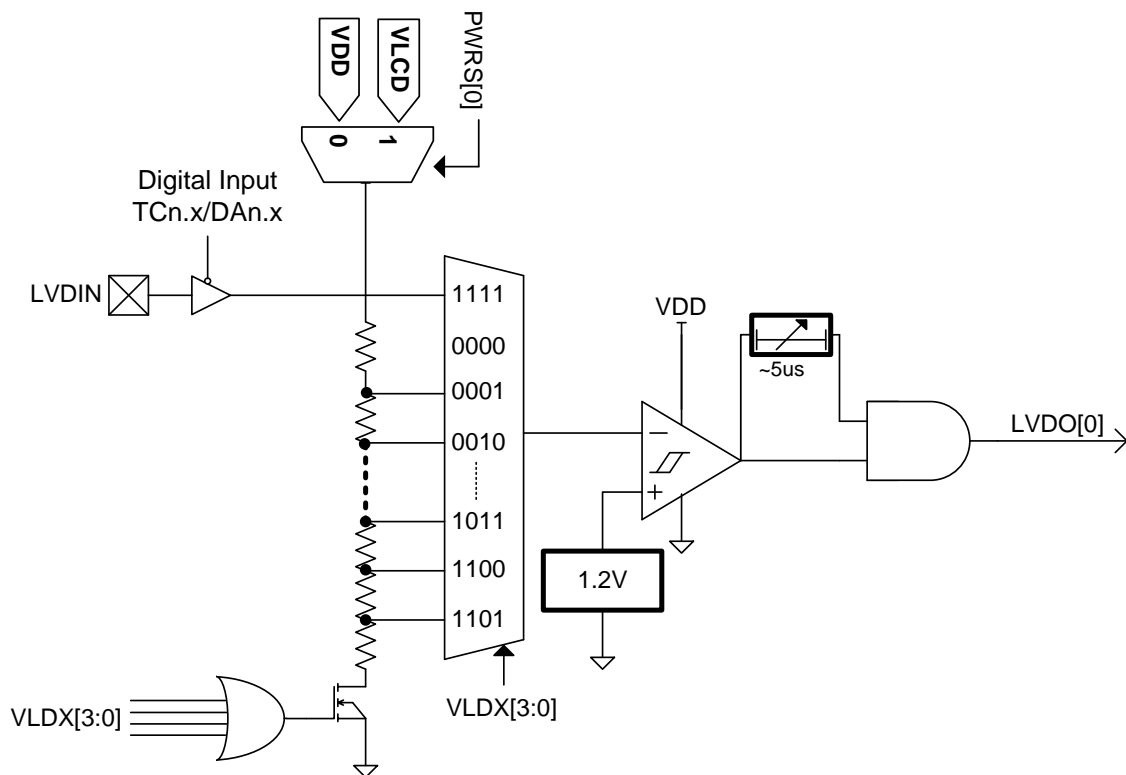


图 9-1 低电压检测方块图

9.1. 低电压侦测使用说明

9.1.1. LVD 初始设置

设置监测电压选择器 VLDX[3:0]可以决定 LVD 侦测的电压是由工作电压分压产生或是外部输入引脚 LVDIN。

当 VLDX[3:0]设置为侦测工作电压时，即可透过分压电阻的匹配达到 14 段的监测的电压点；若设置为外部输入时，用户可能需要设计电压分压电路产生适当的电压信号后再由 LVDIN 引脚输入至 LVD 比较器。

9.1.2. LVD 低电压发生条件及事件记录

LVD 侦测电压的方式是透过内部比较器将监测的电压信号与内部参考电压进行比较,若侦测电压比参考电压高,则比较器输出 1 反之输出 0。运作期间为了确保低电压侦测动作正常必须参考稳定标志 LVDO[0]的状态,在每次变更 VLDX[3:0]的设置值后必须等待延迟时间 5 μ s,使线路达到稳定 LVDO[0]置<1>才可判断低电压反应旗标 LVD[0]。反之,若 LVDO[0]置<0>则所判断的 LVD[0]可能是错误信息。

9.1.3. LVD 启用方式

当 VLDX[3:0]的设置不等于<0000>, LVD 即自动启用。反之, VLDX[3:0]=<0000>则 LVD 自动关闭。

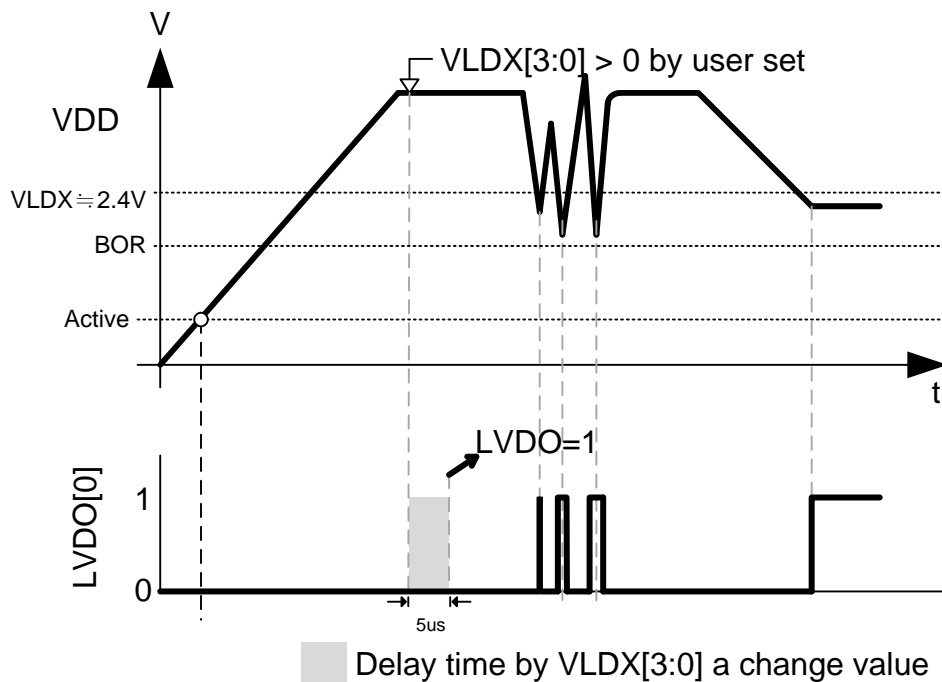


图 9-2 LVD 时序图

9.2. 寄存器说明-LVD

“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
LVDCN	-	-	PWRS	LVDX[3:0]				LVDO	0000 0000	uuuu uuuu	* * * * *

表 9-1 LVD 控制寄存器

PSTAT: 详见 复位 RESET 章节

PT1/PT1DA/PT1PU: 详见 输入/输出埠 I/O 章节

LVDCN: LVD 控制寄存器

位	名称	描述																																				
Bit5	PWRS	used to select which power source is monitored <0> VDD <1> VLCD(HY17P48 没有 VLCD。故不能选择为 1)																																				
Bit4~1	VLDX[3:0]	监测电压选择器 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>VLDX[3:0]</th> <th>监测电压</th> <th>VLDX[3:0]</th> <th>监测电压</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>LVDOFF</td> <td>1000</td> <td>2.7V</td> </tr> <tr> <td>0001</td> <td>2.0V</td> <td>1001</td> <td>2.8V</td> </tr> <tr> <td>0010</td> <td>2.1V</td> <td>1010</td> <td>2.9V</td> </tr> <tr> <td>0011</td> <td>2.2V</td> <td>1011</td> <td>3.0V</td> </tr> <tr> <td>0100</td> <td>2.3V</td> <td>1100</td> <td>3.3V</td> </tr> <tr> <td>0101</td> <td>2.4V</td> <td>1101</td> <td>3.6V</td> </tr> <tr> <td>0110</td> <td>2.5V</td> <td>1110</td> <td>4.0V</td> </tr> <tr> <td>0111</td> <td>2.6V</td> <td>1111</td> <td>PT1.2(LVDIN)</td> </tr> </tbody> </table>	VLDX[3:0]	监测电压	VLDX[3:0]	监测电压	0000	LVDOFF	1000	2.7V	0001	2.0V	1001	2.8V	0010	2.1V	1010	2.9V	0011	2.2V	1011	3.0V	0100	2.3V	1100	3.3V	0101	2.4V	1101	3.6V	0110	2.5V	1110	4.0V	0111	2.6V	1111	PT1.2(LVDIN)
VLDX[3:0]	监测电压	VLDX[3:0]	监测电压																																			
0000	LVDOFF	1000	2.7V																																			
0001	2.0V	1001	2.8V																																			
0010	2.1V	1010	2.9V																																			
0011	2.2V	1011	3.0V																																			
0100	2.3V	1100	3.3V																																			
0101	2.4V	1101	3.6V																																			
0110	2.5V	1110	4.0V																																			
0111	2.6V	1111	PT1.2(LVDIN)																																			
Bit0	LVDO	LVD output 当 VLDX[3:0]=0001~1110 时候的状态 <0> higher than detection voltage <1> lower than detection voltage 当 VLDX[3:0]=1111=PT1.2(LVDIN)时候的状态 <0> higher than detection voltage <1> lower than detection voltage ※HY17P48 需先将 LVDIN 脚位，切换至模拟功能使用 [AIE1.2, DA1.2] = 11b																																				

10. 看门狗, Watch Dog

看门狗 WDT 为顾名思义的为芯片的看守者，主要用于复位或唤醒芯片。

- 运行模式
 - 看门狗计数器溢出产生复位信号，重新启动芯片
 - 可用软件清零定时计数器
- 休眠模式
 - 看门狗 WDT 关闭，无法使用
- 待机模式
 - 看门狗计数器溢出产生中断事件，唤醒芯片

WDT 相关寄存器摘要：

INTE0 GIE[0], WDTIE[0]
INTF0 WDTIF[0]
PSTAT TO[0]
WDTCN ENWDT[0], DWDT[2:0]

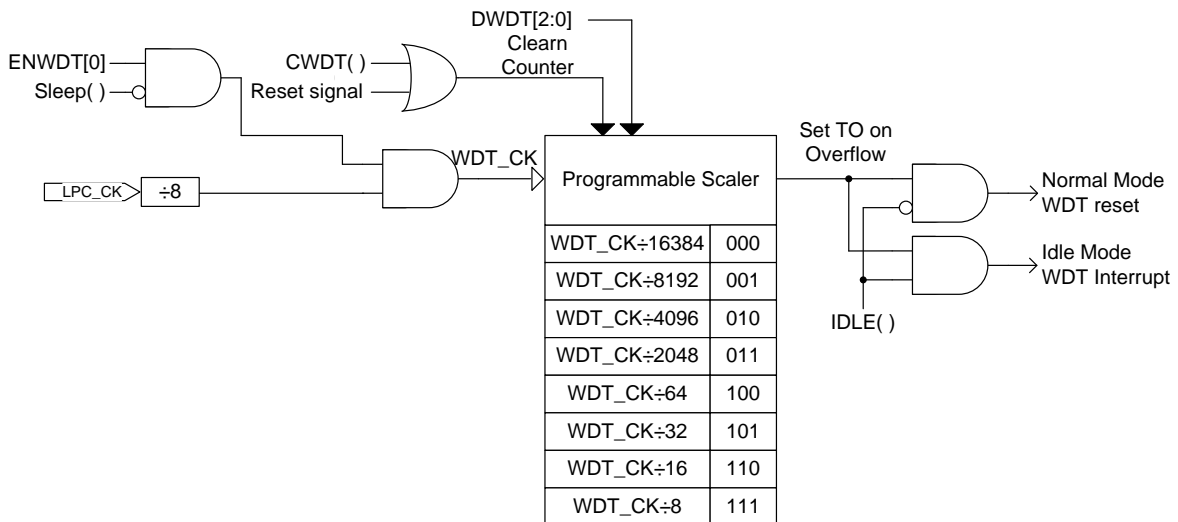


图 10-1 看门狗方块图

10.1. WDT 使用说明

10.1.1. WDT 初始化设置

WDT 计时控制器 `DWDT[2:0]` 可决定 WDT 计数器的工作频率 `WDT_CK` 与计数溢出值，计数器溢出后可产生 WDT 复位信号 `TO` 或中断事件 `WDTIF`²。

² WDT 使用内部时钟源 LPO，故可操作在晶片处于运行模式(Normal Mode)与待机模式(Idle Mode)；运行模式下可用软件方式清零计数器使它不会因计数溢出而复位，但在待机模式下无法透过任何方式使 WDT 计数器清零。

10.1.2. WDT 中断事件服务

WDT 中断事件只能操作在芯片处于待机模式，当 WDTIE[0]与 GIE[0]设置<1>时 WDT 计数器溢出后会产生中断事件将 WDTIF[0]置<1>且程序计数器 PC 跳至中断向量位置<0>x0004h。反之，WDTIE[0]与 GIE[0]设置<0>时不会产生任何中断。

10.1.3. WDT 启用

WDT 必须在芯片处于运行模式下启用，即是将 WDT 启用控制器 ENWDT[0]设置<1>以启用 WDT。启用后即无法用软件再将 ENWDT[0]设置<0>，不论在**运行模式**或者**待机模式**都无法用软件再将 ENWDT[0]设置<0>。DWDT[2:0]设定后，当发生 WDT 重置或中断，DWDT 将会被清除为 000b，需要软件重新再设定。

10.2. WDT 控制寄存器列表与说明：

“.”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	-	ADIE	WDTIE	TB1IE	TMAIE	E1IE	E0IE	0000 0000	0uuu uuuu	*,*,*,*,*,*
INTF0	-	-	ADIF	WDTIF	TB1IF	TMAIF	E1IF	E0IF	.000 0000	.uuu uuuu	*,*,*,*,*,*
PSTAT			TO						\$000 \$00.	uu\$u u\$u.	rw0,rw0,rw0,rw0 rw0,rw0,-
WDTCN					ENWDT			DWDT[2:0]	0000 0000	uuuu \$000	-,*,*,* rw1,*,*,*

表 10-1 WDT 相关寄存器

INTE0/INTF0: 详见 中断 章节

PSTAT[7:0]外设状态标志寄存器

位	名称	描述
Bit5	TO	看门狗 WDT 运行模式计数溢出标志 <0>未发生 WDT 计数溢出事件 <1>已发生 WDT 复位事件；清除需透过 BOR、RST 或指令

WDTCN[7:0]看门狗控制寄存器

位	名称	描述
Bit3	ENWDT[0]	WDT 计数器启用与关闭控制器 <0> 关闭 <1> 启用

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述
Bit2~0	DWDT[2:0]	看门狗 WDT_CK 工作频率选择器
		DWDT[2:0] Pre-scale
		000 WDT_CK 除 16384
		001 WDT_CK 除 8192
		010 WDT_CK 除 4096
		011 WDT_CK 除 2048
		100 WDT_CK 除 64
		101 WDT_CK 除 32
		110 WDT_CK 除 16
111 WDT_CK 除 8		

11. 定时计数器 Timer-A1/2

定时计数器 Timer-A1/2 为 8-bit 的设计架构，TMA1/2 可工作于运行模式与待机模式。

- 递增型计数器
- 八段溢出数值选择
- 8-Bit 溢出次数比较器
- 溢出产生中断事件
- 可读取计数器的值

TMA 寄存器摘要：

TMA1CN	ENTMA1[0], TMA1CK[0], TMA1R[7:0], DTMA1[2:0]
TMA1R	TMA1R[7:0]
TMA1C	TMA1C[7:0]
TMA2CN	ENTMA2[0], TMA2CK[0], TMA2R[7:0], DTMA2[2:0]
TMA2R	TMA2R[7:0]
TMA2C	TMA2C[7:0]
INTE0	GIE, TA1CIE
INTE1	TA1IE
INTE2	TA2IE, TA2CIE
INTF0	TA1CIF
INTF1	TA1IF
INTF2	TA2IF, TA2CIF

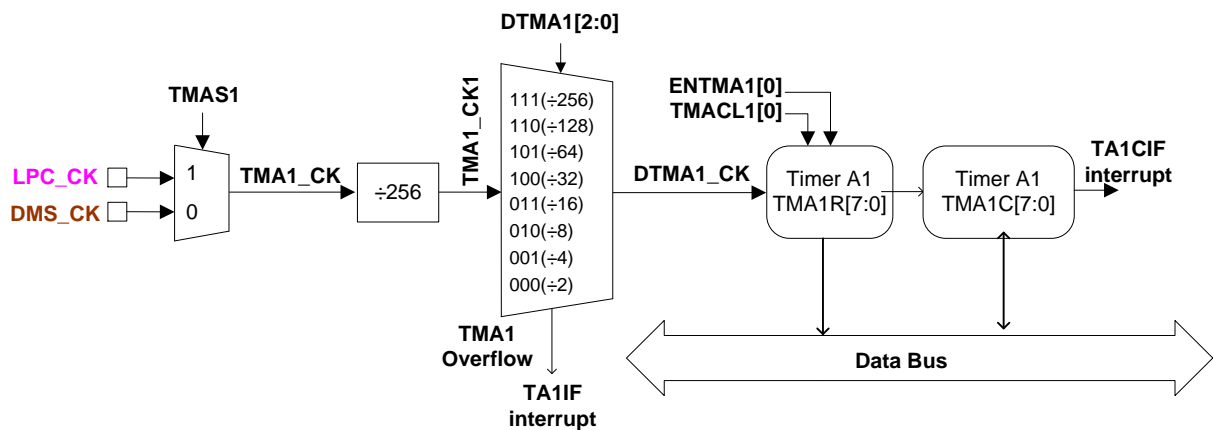


图 11-1 8-Bit 定时计数器 Timer A1 方块图

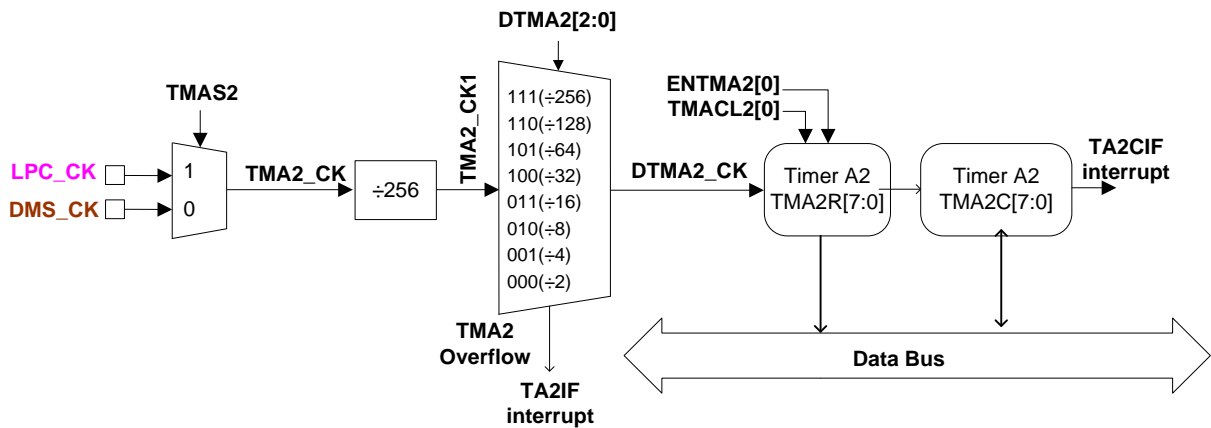


图 11-2 8-bit 定时计数器 Timer A2 方块图

■ 定时计数器操作说明：(x=1 or 2)

设置 $TMAS_x[0]$ 选择 TMA_x_CK 的频率，并经过预分频器 256 降低频率再输入 $DTMA_x$ 分频器。

将 $ENTMA_x[0]$ 设置 $\langle 1 \rangle$ 启用 TMA_x ；反之，设置 $\langle 0 \rangle$ 则关闭并清除 $TMA_xR[7:0]$ 。

$DTMA_x[2:0]$ 计数条件成立产生中断事件，并使得 $TMA_xR[7:0]$ 累进加 1。

TMA_x 中断事件 $TAxIF[0]$ 必须在 $TAxIE[0]$ 设置 $\langle 1 \rangle$ 且 $GIE[0]$ 设置 $\langle 1 \rangle$ 才有中断服务。

读取 $TMA_xR[7:0]$ 不会使得 TMA_x 计数器归零。

使用者将 $TMA_xCL[0]$ 设置 $\langle 1 \rangle$ 清除 TMA_x 所有计数器后， $TMA_xCL[0]$ 自动置 $\langle 0 \rangle$ 。

$TMA_xR[7:0]$ 可读取 TMA 累进计数器的数值，并可以以写入任意值动作清除 $TMA_xR[7:0]$ 计数数值。

■ 定时计数器的比较功能操作说明: (x=1 or 2)

设置 $TMAS_x$ 选择 TMA_x_CK 的频率 先经过预除频 256 之后 为 TMA_x_CK 频率源后再输入 $DTMA_x$ 除频器。

将 $ENTMA_x$ 设置 $\langle 1 \rangle$ 启用 TMA_x ，并清除 TMA_x_CK 、 $DTMA_x_CK$ 、 TMA_xR 等计数器，从 0 开始计数；反之，设置 $\langle 0 \rangle$ 则关闭 TMA_x 。

$DTMA_x[2:0]$ 计数条件成立产生中断事件($TAxIF$)，并使得 $TMA_xR[7:0]$ 累进加 1。

TMA_x 中断事件 $TAxIF$ 必须在 $TAxIE$ 设置 $\langle 1 \rangle$ 且 GIE 设置 $\langle 1 \rangle$ 才有中断服务。

读取 $TMA_xR[7:0]$ 不会使得 TMA_x 计数器归零。

使用者将 TMA_xCL_x 设置 $\langle 1 \rangle$ 清除 TMA_x_CK 、 $DTMA_x_CK$ 计数器后， $TMA_xCL_x[0]$ 由硬件自动置 $\langle 0 \rangle$ 。

$TMA_xR[7:0]$ 可读取 TMA_x 累进计数器的数值 并可以以写入动作会视为清除 $TMA_xR[7:0]$ 计数数值，从 $TMA_xR[7:0]=0$ 开始重新计算。

$TMA_xC[7:0]$ 是 $Timer Ax$ 的比较点寄存器，可以被读写。当 BOR/POR 发生之后， $TMA_xC[7:0]$ 会被设成 0。当 $TMA_xR[7:0]$ 值累加到和 $TMA_xC[7:0]$ 一样时， $TAxCIF$ 旗标都被置 1。同样 $TAxCIF$ 必须在 $TAxCIE$ 设置 $\langle 1 \rangle$ 且 GIE 设置 $\langle 1 \rangle$ 才有中断服务。

当设计 $TAxIF$ 为 1 秒中断，如果需求为 60 秒中断唤醒，则可以设定 $TAxCIF$ 为 60 秒中断，并开启 $TAxCIE$ 中断要求来唤醒芯片。

11.1. 寄存器说明-TMA1/2

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	TA1CIE							0000 0000	0uuu uuuu	*; ; ; ; ; ; ; ;
INTE1	TA1IE	SPIIE	TXIE	ROIE	I2CERIE	I2CIE	E3IE	E2IE	0000 0000	uuuu uuuu	*; ; ; ; ; ; ; ;
INTE2	TA2IE	TA2CIE							0000 0000	uuuu uuuu	*; ; ; ; ; ; ; ;
INTF0	-	TA1CIF	ADIF	WDTIF	TB1IF	TMAIF	E1IF	E0IF	.000 0000	.uuu uuuu	*; ; ; ; ; ; ; ;
INTF1	TA1IF								0000 0000	uuuu uuuu	*; ; ; ; r; r; ; ;
INTF2	TA2IF	TA2CIF	TC1IF	TC0IF	TX2IF	RC2IF		BOR2IF	0000 0000	uuuu uuuu	*; ; ; ; ; ; ; ;
OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*; ; ; ; ; ; ; ;
OSCCN1		LCPS							0000 0000	uuuu uu.	*; ; ; ; ; ; ; ;-
OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	*; ; ; ; ; ; ; ;r
TMACN	ENTMA	TMACL	TMAS	DTMA[2:0]			-	-	0000 00..	u0uu uu..	*;rw1; ; ; ; ; ; ; ;-
TMAR	TMA counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
TMA1C	TMA1C counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
TMA2CN	ENTMA2	TMACL2	TMAS2	DTMA2[2:0]			-	-	0000 00..	u0uu uu..	*;rw1; ; ; ; ; ; ; ;-
TMA2R	TMA2 counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0
TMA2C	TMA2C counter Register								0000 0000	uuuu uuuu	rw0,rw0,rw0,rw0 rw0,rw0,rw0,rw0

表 11-1 TMA1/2 控制寄存器

INTE0/INTF0/INTE1/INTF1/INTE2/INTF2: 详见 中断 章节

OSCCN0/OSCCN1/OSCCN2: 详见 錯誤! 找不到參照來源。 章节

TMAxCN: 定时计数器 Timer Ax 控制寄存器(x=1 or 2)

位	名称	描述
Bit7	ENTMAx	Timer-Ax 启用控制器 <0> 关闭。 <1> 启用, 并清除 TMAx_CK、DTMAx_CK、TMAxR 等计数器。
Bit6	TMACLx	TMAx 分频器计数归零 <0>TMAx 分频器正常计数。 <1>TMAx 分频器计数归零。TMACLx 设置<1>清除 TMAx_CK、DTMAx_CK 计数器后, TMACLx 由硬件自动置<0>。
Bit5	TMASx	TMAx 工作频率选择器 <0>DMS_CK <1>LPC_CK

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述																																															
Bit4~2	DTMAx[2:0]	<p>启用与关闭控制器</p> <table border="1"> <thead> <tr> <th>DTMA[2:0]</th> <th>频率分配选择器</th> <th>DTMA[2:0]</th> <th>频率分配选择器</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>TMAx_CK ÷ 2</td> <td>100</td> <td>TMAx_CK ÷ 32</td> </tr> <tr> <td>001</td> <td>TMAx_CK ÷ 4</td> <td>101</td> <td>TMAx_CK ÷ 64</td> </tr> <tr> <td>010</td> <td>TMAx_CK ÷ 8</td> <td>110</td> <td>TMAx_CK ÷ 128</td> </tr> <tr> <td>011</td> <td>TMAx_CK ÷ 16</td> <td>111</td> <td>TMAx_CK ÷ 256</td> </tr> </tbody> </table> <p>计算后的中断时间为：</p> <p>0.555msec~ 71.118msec (at clock source=DMS_CK, and HAO=1.843MHz) 35.3msec~ 4519.7msec (at clock source=LPC_CK, and LPO=14.5kHz)</p> <table border="1"> <thead> <tr> <th>TMAx_CK(kHz)</th> <th>TMAx_CK1(kHz)</th> <th>DTMAx[2:0]</th> <th>DTMAx_CK(kHz)</th> <th>TAxIF(msec)</th> </tr> </thead> <tbody> <tr> <td rowspan="3">921.5</td> <td rowspan="3">3.5996093</td> <td>000b</td> <td>1.799804687</td> <td>0.555</td> </tr> <tr> <td>100b</td> <td>0.11248779</td> <td>8.889</td> </tr> <tr> <td>111b</td> <td>0.014060974</td> <td>71.118</td> </tr> <tr> <td rowspan="3">14.5</td> <td rowspan="3">0.05664063</td> <td>000b</td> <td>0.028320315</td> <td>35.3</td> </tr> <tr> <td>100b</td> <td>0.00177002</td> <td>564.96</td> </tr> <tr> <td>111b</td> <td>0.00022125</td> <td>4519.7</td> </tr> </tbody> </table>	DTMA[2:0]	频率分配选择器	DTMA[2:0]	频率分配选择器	000	TMAx_CK ÷ 2	100	TMAx_CK ÷ 32	001	TMAx_CK ÷ 4	101	TMAx_CK ÷ 64	010	TMAx_CK ÷ 8	110	TMAx_CK ÷ 128	011	TMAx_CK ÷ 16	111	TMAx_CK ÷ 256	TMAx_CK(kHz)	TMAx_CK1(kHz)	DTMAx[2:0]	DTMAx_CK(kHz)	TAxIF(msec)	921.5	3.5996093	000b	1.799804687	0.555	100b	0.11248779	8.889	111b	0.014060974	71.118	14.5	0.05664063	000b	0.028320315	35.3	100b	0.00177002	564.96	111b	0.00022125	4519.7
DTMA[2:0]	频率分配选择器	DTMA[2:0]	频率分配选择器																																														
000	TMAx_CK ÷ 2	100	TMAx_CK ÷ 32																																														
001	TMAx_CK ÷ 4	101	TMAx_CK ÷ 64																																														
010	TMAx_CK ÷ 8	110	TMAx_CK ÷ 128																																														
011	TMAx_CK ÷ 16	111	TMAx_CK ÷ 256																																														
TMAx_CK(kHz)	TMAx_CK1(kHz)	DTMAx[2:0]	DTMAx_CK(kHz)	TAxIF(msec)																																													
921.5	3.5996093	000b	1.799804687	0.555																																													
		100b	0.11248779	8.889																																													
		111b	0.014060974	71.118																																													
14.5	0.05664063	000b	0.028320315	35.3																																													
		100b	0.00177002	564.96																																													
		111b	0.00022125	4519.7																																													

TMAxR: TMAx 的递增型计数器，可读取不可写入。

写入动作会视为清除 TMAxR[7:0]计数数值，从 TMAxR[7:0]=0 开始重新计算。

TMAxC: Timer Ax 计数器的比较点，可读取可写入。

12.16-bit 定时计数器 B, TMB (16-bit TimerB)

定时计数器 B (以下简称 TMB), TMB 具有两个 PWM 输出, 分别为 PWMA0/1。而每个 TMB 具有四种操作模式, 每个模式的计数器皆具有特殊功能设计, 以满足不同的应用方式。

TMB 寄存器摘要 :

INTE0	GIE, TMBIE
INTF0	TMBIF
OSCCN1	DTMB[1:0], TMBS
TB1Flag	PWM6A, PWM5A, PWM4A, PWM3A, PWM2A, PWM1A
TB1CN0	ENTB1, TB1M[1:0], TB1RT[1:0], TB1CL
TB1CN1	PA1IV, PWMA1[2:0], PA0IV, PWMA0[2:0]
TB1R	TB1RH[15:8], TB1RL[7:0]
TB1C0	TB1C0[15:8], TB1C0L[7:0]
TB1C1	TB1C1H[15:8], TB1C1L[7:0]
TB1C2	TB1C2H[15:8], TB1C2L[7:0]
TC1CN0	TC1S[1:0]

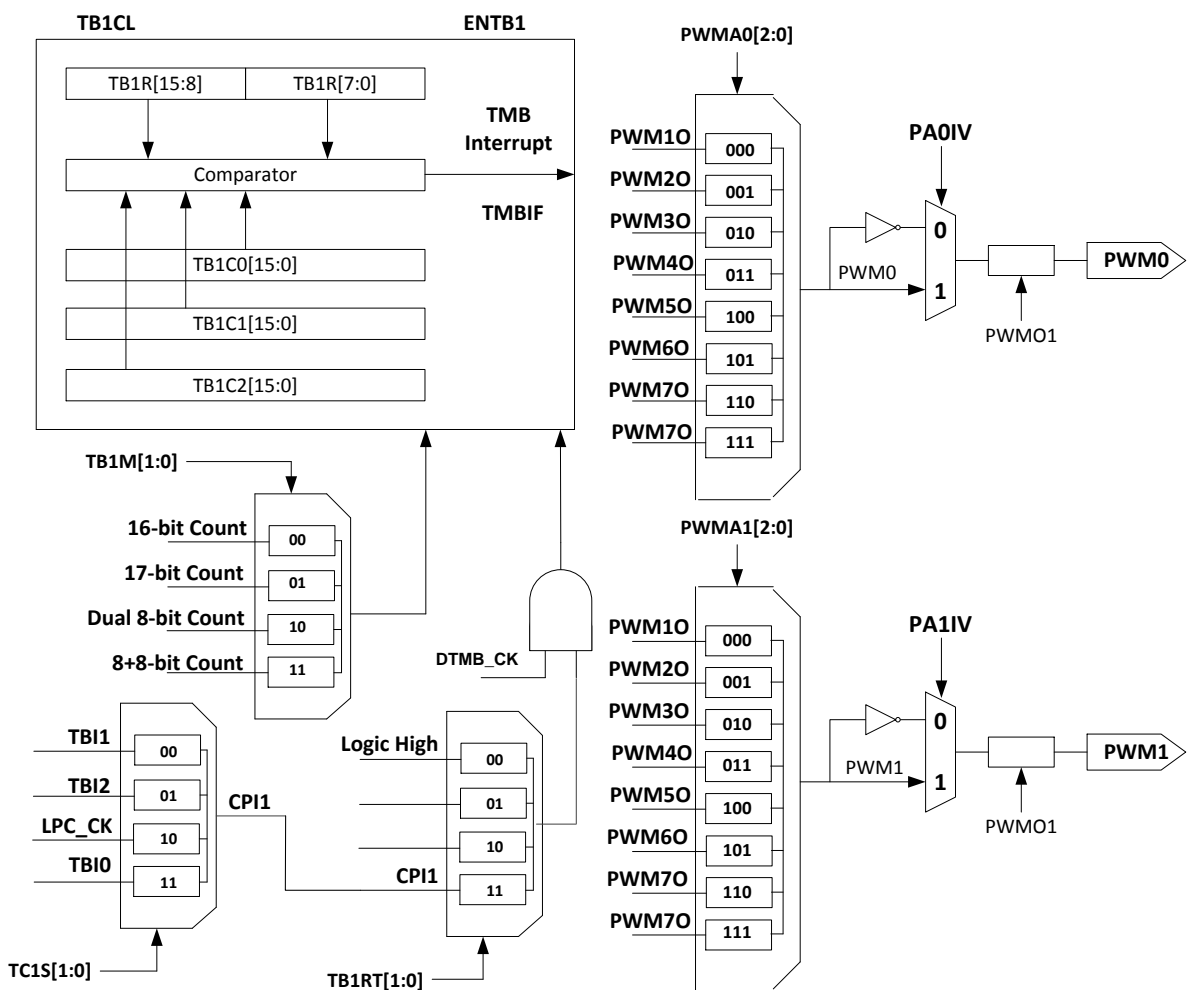


图 12-1 TMB 计数器架构图

- TMB 的计数寄存器分别为

递增/递减式计数器 TB1R[15:0]

计数溢出事件条件控制器 TB1C0[15:0]

PWMA 条件控制器 TB1C1[15:0]

PWMA 条件控制器 TB1C2[15:0]

启用控制器 ENTB1[0]

模式控制器 TB1M[1:0]

触发控制器 TB1RT[1:0]

归零控制器 TB1CL[0]

PWM0 输出波形选择器 PWMA0[2:0]

PWM0 输出反相控制器 PA0IV[0]

PWM1 输出波形选择器 PWMA1[2:0]

PWM1 输出反相控制器 PA1IV[0]

工作频率源选择器 TMBS[0]

工作频率分频器 DTMB[1:0]

- TMB 四种计数模式

- 16-bit 计数

- 17-bit 计数

- 两组 8-bit 计数

- 8+8-bit 计数

- TMB 的系统功耗操作

- 运行模式

- 待机模式

- 休眠模式

- TB1R[15:0]归零重新计数条件

- 读取 TMB 相关寄存器，不会使得 TB1R[15:0]归零重新计数

- 写入 TB1R[15:0](只读)、TB1C0[15:0]、TB1C1[15:0]与 TB1C2 [15:0]不会使得 TB1R[15:0]归零重新计数

- 写入 TB1CN0 与 TB1CN1 控制寄存器不会使得 TB1R[15:0]归零重新计数。

- TB1R[15:0]采累进计数至大于 TB1C0[15:0]将使得 TB1R[15:0]归零重新计数。

- 使用者将 TB1CL[0]设置<1>清除 TB1R[15:0]计数器后，TB1CL[0]自动置<0>。

12.1. TMB 四种计数模式

定时计数器 TMB 有四种计数方式，可透过计数模式选择器 TB1M[1:0]进行选择。而每种计数模式具有不同的溢出与中断产生方式，在此章节说明四种计数模式的操作方法。

另外，不同的计数模式与 PWM 条件选择器搭配后可产生七种不同的 PWM 波形。在后续章节会逐一描述。

12.1.1. 16-bit 计数器

将计数模式选择器 TB1M[1:0]设置<00>使得 TMB 操作在 16-bit 计数模式下，在此模式下具有以下特性：

- TB1R[15:0]计数器的计数开始可由 TB1RT[1:0]设置不同事件触发。
- TB1R[15:0]累加计数至等于 TB1C0[15:0]时，计数溢出硬件置 TB1IF[0]为 1，并将 TB1R[15:0]归零重新计数。

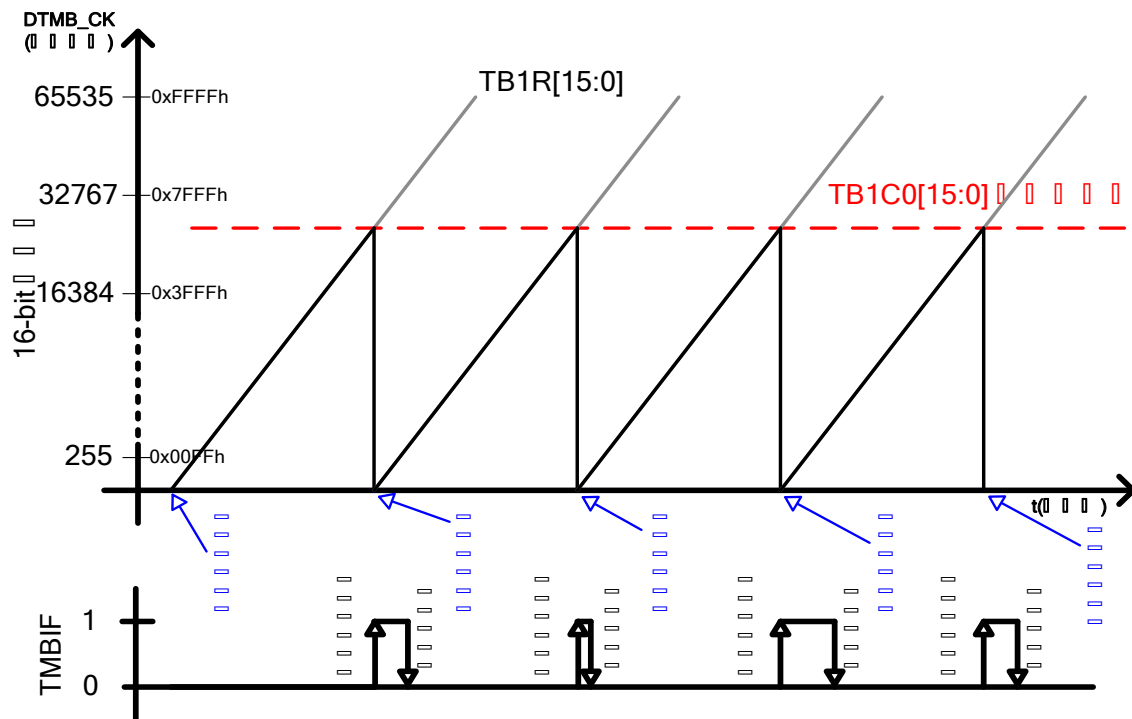


图 12-2 16-bit 计数器波形与使用示意图

- 16-bit 计数模式操作说明
- ◆ 初始化
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源，设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<00>，将 TMB1 规划为 16-bit 计数器。
 - 写入数据至 TB1C0[15:0]。
- ◆ 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用状态 (Always Enable)，即循环计数。
- ◆ 将 ENTB1[0]设置<1>以启用计数器
 - 当 TB1R[15:0]计数数值至等于 TB1C0[15:0]时，产生计数溢出事件使得 TB1IF[0]置<1>并归零重新递增计数，此时 TB1IE[0]设置<1>则会产生中断事件服务。
 - 计数过程，使用者可利用计数归零控制器 TB1CL[0]设置<1>以重新计数，且 TB1CL[0]自动置<0>。

- ◆ 将 ENTB1[0]设置<0>以关闭计数器。
- ◆ 范例 1
TMB 搭配外振 32768Hz 产生 1 秒中断
 - 将 TMB 频率源选到 LPC_CK，并将 DTMB[1:0]设定为 00 除频为 1
 - TB1M[1:0]设置<00>，将 TMB1 规划为 16-bit 计数器。
 - 产生 1 秒的中断需计数 32768 次。但由于 TB1R[15:0]是从 0 开始计数。故是从 0 数到 32767 就完成了 32768 次。
 - 所以 TB1C0[15:0] 因填入 32767 也就是 7FFFH
- ◆ 范例 2
TB1M[1:0]设置<00>，将 TMB1 规划为 16-bit 计数器；TMB 频率源选到 LPC_CK(32768Hz)，并将 DTMB[1:0]设定为 00。
 - 根据范例 2 的设定将 TB1C0[15:0]写入 1。
当 TB1R[15:0]从 0 开始数，数到 1 时产生中断。
故实际上是数了 2 次。
故中断频率为 $32768/(TB1C0+1)=16384\text{Hz}$
 - 根据范例 2 的设定将 TB1C0[15:0]写入 7。
当 TB1R[15:0]从 0 开始数，数到 7 时产生中断。
故实际上是数了 8 次。
故中断频率为 $32768/(TB1C0+1)=4096\text{Hz}$

12.1.2. 17-bit 计数器

将计数模式选择器 TB1M[1:0]设置<01>使得 TMB 操作在 17-bit 计数模式下，在此模式下具有以下特性：

- TB1R[15:0]计数器的计数开始可由 TB1RT[1:0]设置不同事件触发。
- TB1R[15:0]递增累加计数至等于 TB1C0[15:0]时延迟半个指令周期后改成递减计数，且当递减计数至 TB1R[15:0]为 0000h 时则产生计数溢出事件使得 TB1IF[0]置 1，并重新递增计数。

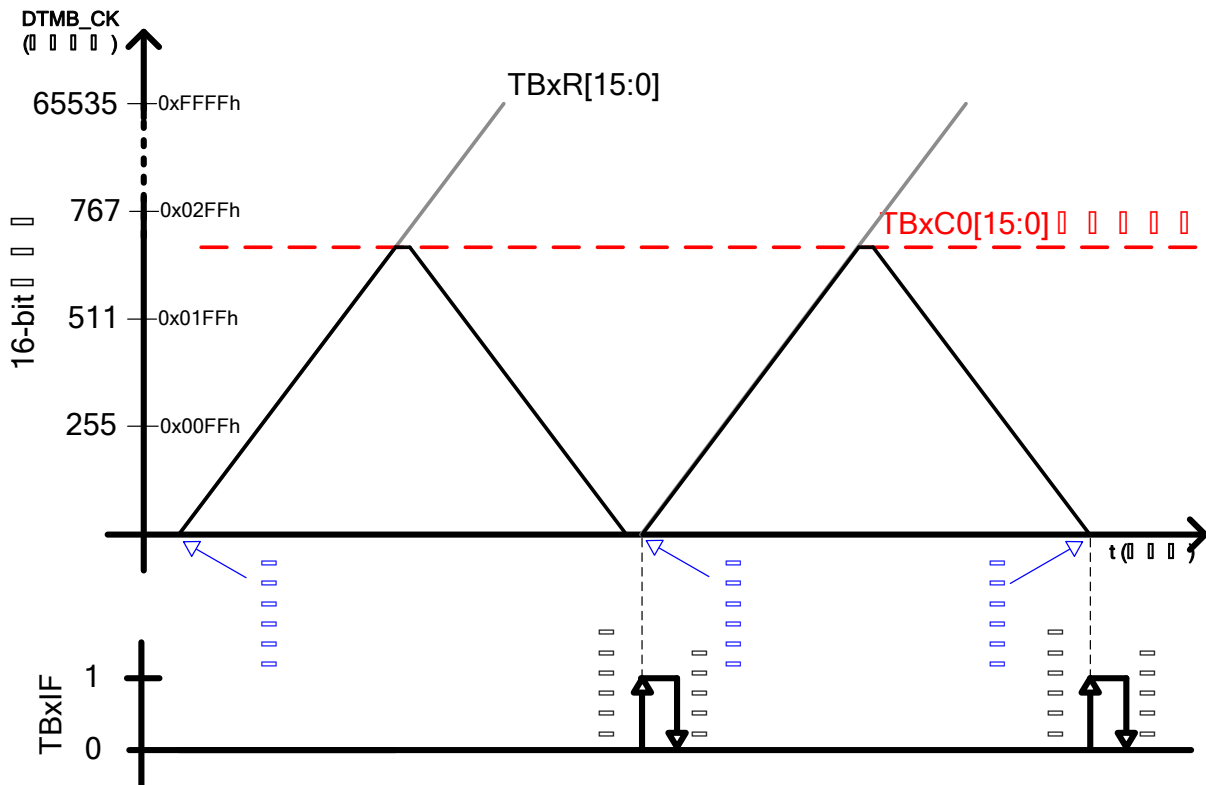


图 12-3 17-bit 计数器波形与使用示意图

- 17-bit 计数模式操作说明
- ◆ 初始化
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源，设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<01>，将 TMB1 规划为 17-bit 计数器。
 - 写入数据至 TB1C0[15:0]。
- ◆ 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用状态 (Always Enable)，即循环计数。
- ◆ 将 ENTB1[0]设置<1>以启用计数器
 - 当 TB1R[15:0]计数数值至等于 TB1C0[15:0]时延迟半个指令周期后改成递减计数，且当递减计数至 TB1R[15:0]为 0000h 则产生溢位事件使得 TB1IF[0]置<1>并归零重新递增计数，此时 TB1IE[0]设置<1>则会产生中断事件服务。
 - 计数过程，使用者可利用计数归零控制器 TB1CL[0]设置<1>以重新计数，且 TB1CL[0]自动置<0>。
- ◆ 将 ENTB1[0]设置<0>以关闭计数器。
- ◆ 范例 1
TMB 搭配外振 32768Hz 产生 1 秒中断

- 将 TMB 频率源选到 LPC_CK，并将 DTMB[1:0]设定为 00 除频为 1
- TB1M[1:0]设置<01>，将 TMB1 规划为 17-bit 计数器。
- 产生 1 秒的中断需计数 32768 次。但由于 TB1R[15:0]是从 0 开始计数。故是从 0 数到 16383 就完成上数的 16384 次，经过 2 个指令周期的延迟开始下数从 16383 下数到 0，一共完成 32768 次。
- 所以 TB1C0[15:0] 因填入 16383 也就是 3FFFH

◆ 范例 2

TB1M[1:0]设置<01>，将 TMB1 规划为 17-bit 计数器；TMB 频率源选到 LPC_CK(32768Hz)，并将 DTMB[1:0]设定为 00。

- 根据范例 2 的设定将 TB1C0[15:0]写入 1。
当 TB1R[15:0]从 0 开始数数到 1，再由 1 数回 0 时产生中断
故实际上是数了 4 次。
故中断频率为 $32768 / ((TB1C0 + 1) * 2) = 8192\text{Hz}$
- 根据范例 2 的设定将 TB1C0[15:0]写入 7。
当 TB1R[15:0]从 0 开始数数到 7，再由 7 数回 0 时产生中断。
故实际上是数了 16 次。
故中断频率为 $32768 / ((TB1C0 + 1) * 2) = 2048\text{Hz}$

12.1.3. 两组 8-bit 计数器

将计数模式选择器 TB1M[1:0]设置<10>使得 TMB 操作在两组 8-bit 计数模式下,在此模式下具有以下特性:

- 两个 8-bit 计数器 TB1R[7:0]与 TB1R[15:8]的计数开始可由 TB1RT[1:0]设置不同事件做同时触发。
- TB1R[7:0]累加计数至等于 TB1C0[7:0]时产生计数溢出事件 TB1IF[0],并将 TB1R[7:0]归零重新计数。
- TB1R[15:8]累加计数至等于 TB1C0[15:8]时产生计数溢出并将 TB1R[15:8]归零重新计数。

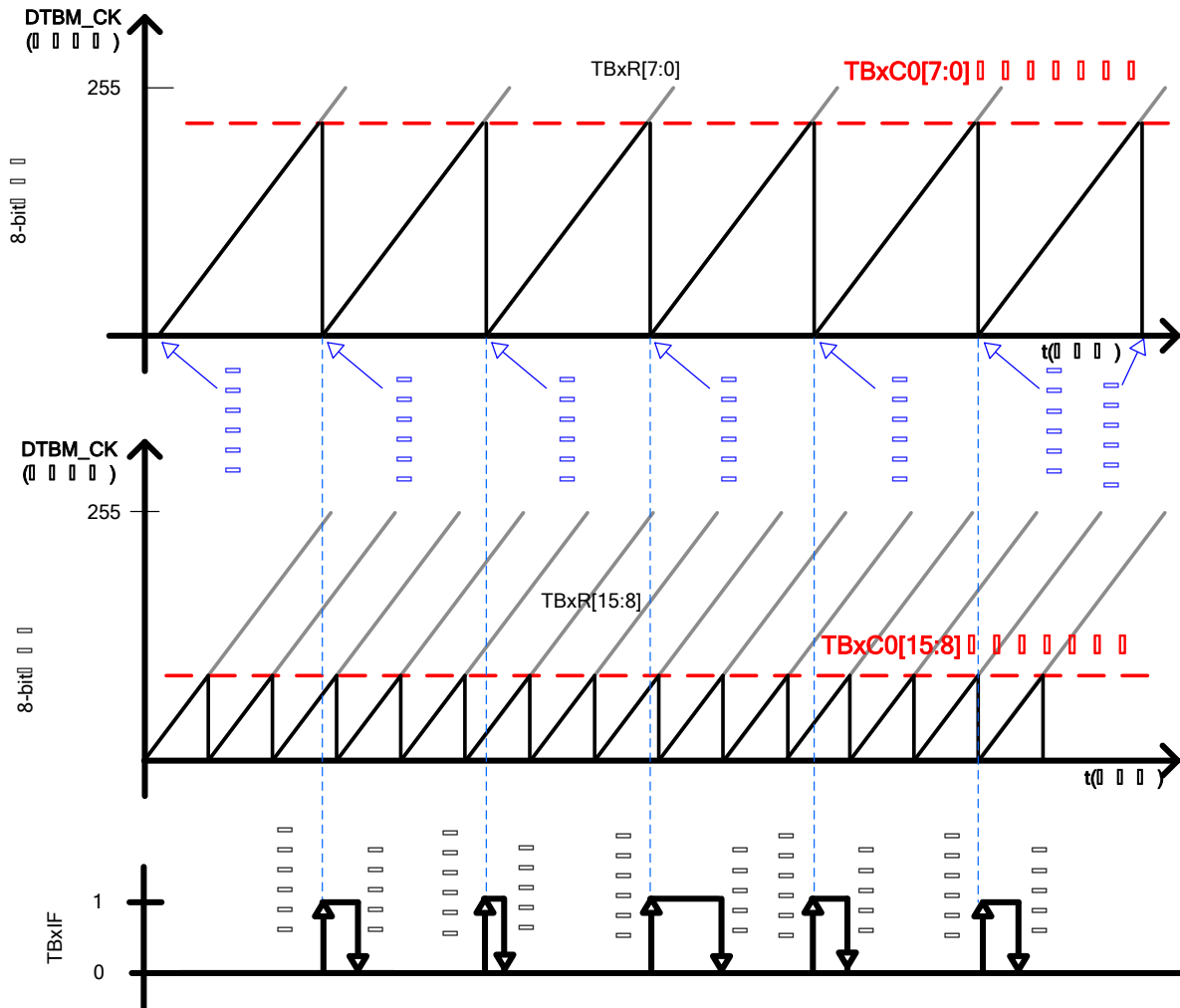


图 12-4 两组 8-bit 计数器波形与使用示意图

- 两组 8-bit 计数模式操作说明
- ◆ 初始化
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源，设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<10>，将 TMB1 规划为两组 8-bit 计数器。
 - 分别写入数据至 TB1C0[7:0]与 TB1C0[15:8]。
- ◆ 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用状态 (Always Enable)，即循环计数。
- ◆ 将 ENTB1[0]设置<1>以启用计数器
 - 当 TB1R[7:0]计数数值至等于 TB1C0[7:0]时，产生计数溢出事件使得 TB1IF[0]置<1>并归零重新递增计数，此时 TB1IE[0]设置<1>则会产生中断事件服务。
 - 当 TB1R[15:8]计数数值至等于 TB1C0[15:8]时，产生计数溢出并且 TB1R[15:8]归零重新递增计数。
 - 计数过程，使用者可利用计数归零控制器 TB1CL[0]设置<1>以使得 TB1R[7:0]与 TB1R[15:8]同时重新计数，且 TB1CL[0]自动置<0>。
- ◆ 将 ENTB1[0]设置<0>以关闭计数器。
- ◆ 范例 1

TB1M[1:0]设置<10>，将 TMB1 规划为两组 8-bit 计数器；TMB 频率源选到 LPC_CK(32768Hz)，并将 DTMB[1:0]设定为 00。

 - 根据范例 1 的设定将 TB1C0[15:0]写入 1。
当 TB1R[15:0]从 0 开始数，数到 1 时产生中断。
故实际上是数了 2 次。
故中断频率为 $32768/(TB1C0+1)=16384\text{Hz}$
 - 根据范例 1 的设定将 TB1C0[15:0]写入 7。
当 TB1R[15:0]从 0 开始数，数到 7 时产生中断。
故实际上是数了 8 次。
故中断频率为 $32768/(TB1C0+1)=4096\text{Hz}$

12.1.4. 8+8-bit 计数器

将计数模式选择器 TB1M[1:0]设置<11>使得 TMB 操作在两组 8+8-bit 计数模式下,在此模式下具有以下特性:

- 8+8-bit 计数器 TB1R[15:8]与 TB1R[7:0]的计数开始可由 TB1RT[1:0]设置不同事件触发。
- TB1R[7:0]累加计数至等于 TB1C0[7:0]时产生计数溢出事件 TB1IF[0], 并使得 TB1R[15:8]计数器累加 1 且 TB1R[7:0]归零重新计数。

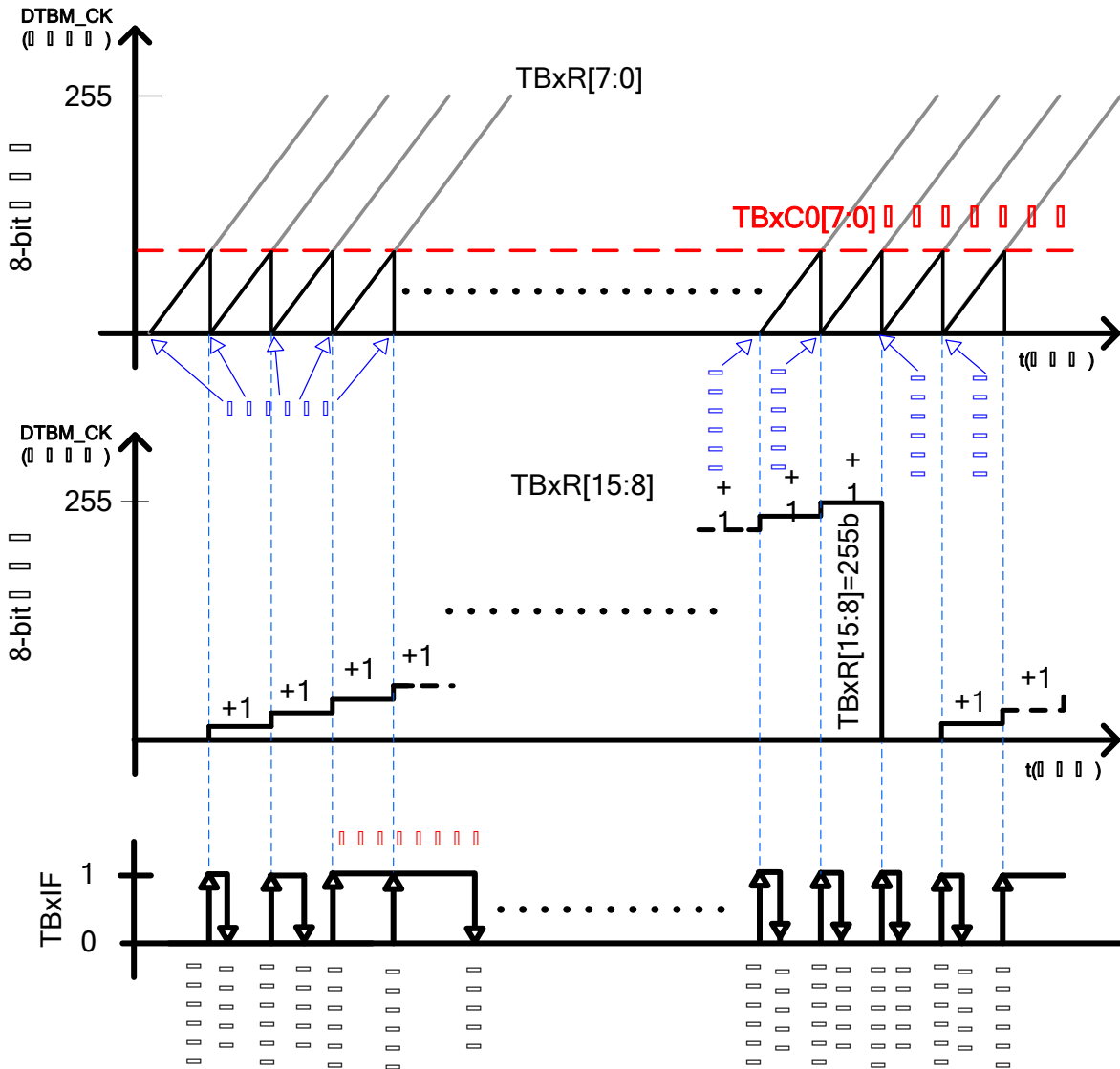


图 12-5 8+8-bit 计数器波形与使用示意图

- 8+8-bit 计数模式操作说明
- ◆ 初始化
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源，设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<11>，将 TMB1 规划为 8+8-bit 计数器。
 - 写入数据至 TB1C0[7:0]。
- ◆ 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用状态 (Always Enable)，即循环计数。
- ◆ 将 ENTB1[0]设置<1>以启用计数器
 - 当 TB1R[7:0]计数数值等于 TB1C0[7:0]时，产生计数溢出事件使得 TB1IF[0]置<1>且 TB1R[15:8]计数器累加 1，此时 TB1IE[0]设置<1>则会产生中断事件并归零重新递增计数。
 - 当 TB1R[15:8]计数数值等于 TB1R[15:8]=255b 时，再加 1 会使得 TB1R[15:8]归零重新递增计数。
 - 计数过程，使用者可利用计数归零控制器 TB1CL[0]设置<1>以使得 TB1R[7:0]与 TB1R[15:8]同时重新计数，且 TB1CL[0]自动置<0>。
- ◆ 将 ENTB1[0]设置<0>以关闭计数器。

12.2. PWM 脉冲宽度调变

当 TMB 不同的计数模式与脉冲宽度调变(简称 PWM)模式选择器进行组合使用,可产生多种形式的 PWM 波形,其中 PWMA0/PWMA1 为实际可输出之引脚。本章节介绍七种不同的使用方式供使用者参考。

- TMB 与 PWM 输出的关系与基本操作说明
- ◆ TMB1 控制 PWMA0 与 PWMA1 输出
 - 由 PWM 模式选择器 PWMA0[2:0]与 PWMA1[2:0], 分别设置 PWMA0 与 PWMA1 输出波形为 PWM1O ~ PWM7O 的其中一种。
 - 由波形状态标志 PWMA1[0] ~ PWMA6[0]可以分别读到 PWM1O ~ PWM6O 为“H”或“L”状态。
 - 透过 PWM 输出反相器 PA0IV[0]与 PA1IV[0], 可分别设置 PWMA0 与 PWMA1 实际输出波形是否反相。
 - PWMA0 与 PWMA1 可分别由引脚 PT2.2 与 PT2.3 输出。
- ◆ PWM 模式选择器 PWMA0/1[2:0], 可输出 PWM1O ~ PWM7O 等波形。必须注意, 当搭配不同 TMB 计数模式时 PWM1O ~ PWM7O 可输出截然不同的波形, 以下章节将以基本型态与常见应用描述。

12.2.1. PWM1O 波形 (16-bit PWM)

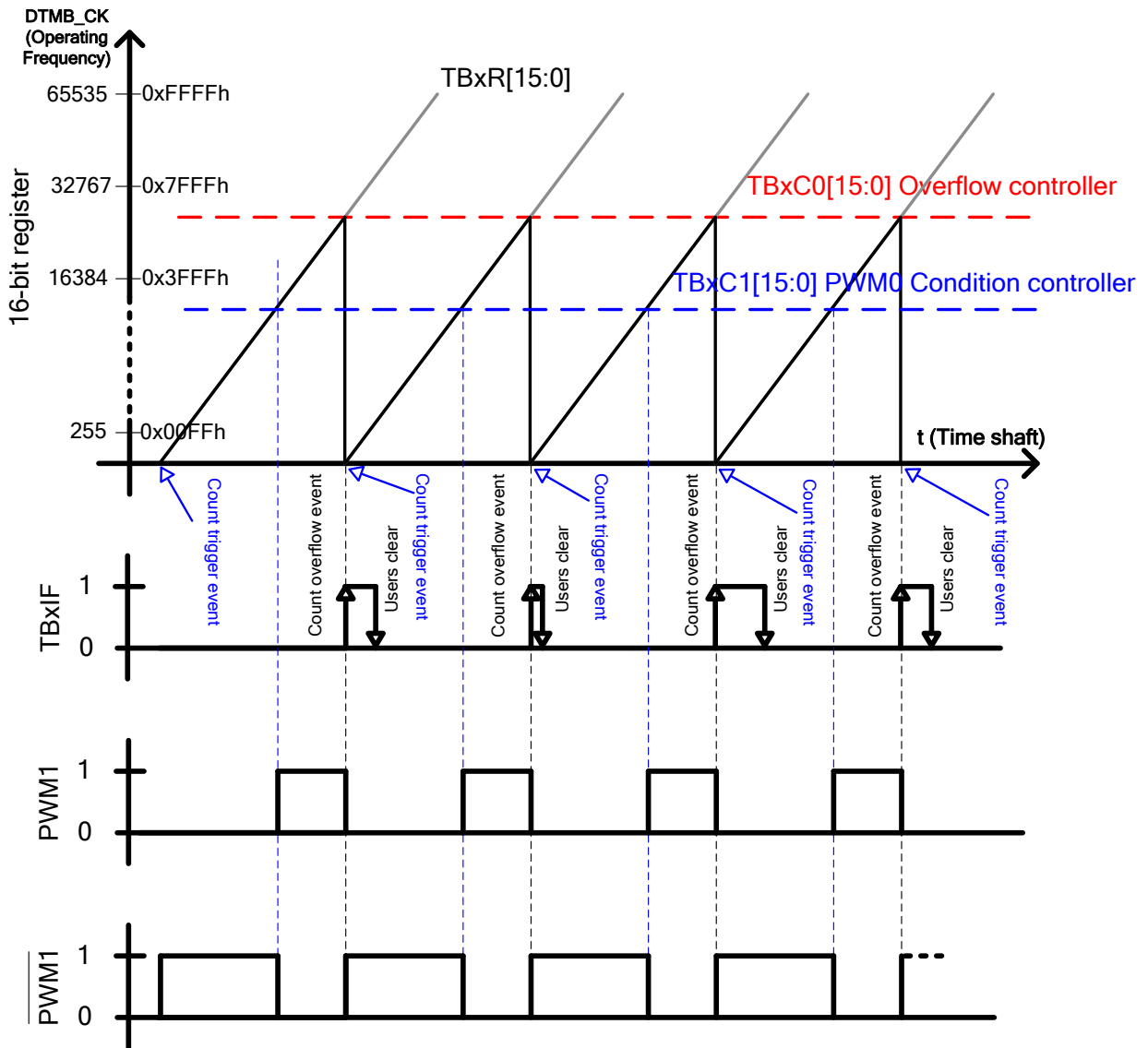


图 12-6 PWM1O 波形与使用示意图

- PWM1O 操作说明
- ◆ 初始化 (PWM 频率与工作周期设置)
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源, 设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<00>, 将 TMB1 规划为 16-bit 计数器。
 - PWMA0/1[2:0]设置<000>以输出 PWM1O 波形
 - 将 TB1RT[1:0]设置<00>以选择触发计数信号为 Logic High。
 - 写入数据至 TB1C0[15:0], 以决定 PWM 之频率。
 - 写入数据至 TB1C1[15:0], 以决定 PWM 之工作周期(Duty Cycle)。
 - 将 ENTB1[0]设置<1>启用计数器。
- ◆ 产生 PWM1O 波形
 - 当 TB1R[15:0]计数数值等于 TB1C1[15:0]时, 使得 PWM1O 状态由 0→1。
 - 当 TB1R[15:0]再计数数值等于 TB1C0[15:0]时, 使得 PWM1O 状态由 1→0; 并产生计数溢出事件使得 TB1IF[0]置<1>并归零重新递增计数, 此时 TB1IE[0]设置<1>则会产生中断事件服务。
- ◆ PWM 输出控制
 - 设置 PWMO0/1[0]设置<1>, 开启 PWM Mode。
 - 设置 PA0/1IV[0]以决定引脚输出波形是否反相。
- ◆ 将 ENTB1[0]设置<0>则关闭计数器与 PWM 输出。
- ◆ PWM1O 频率与工作周期计算公式 :

$$\text{PWM1O Frequency} = \frac{\text{DTMB_CK}}{\text{TB1C0}[15 : 0] + 1}$$

$$\text{PWM1O Duty Cycle} = \frac{(\text{TB1C0}[15 : 0] + 1) - \text{TB1C1}[15 : 0]}{\text{TB1C0}[15 : 0] + 1}$$

12.2.2. PWM2O 波形 (16-bit PWM)

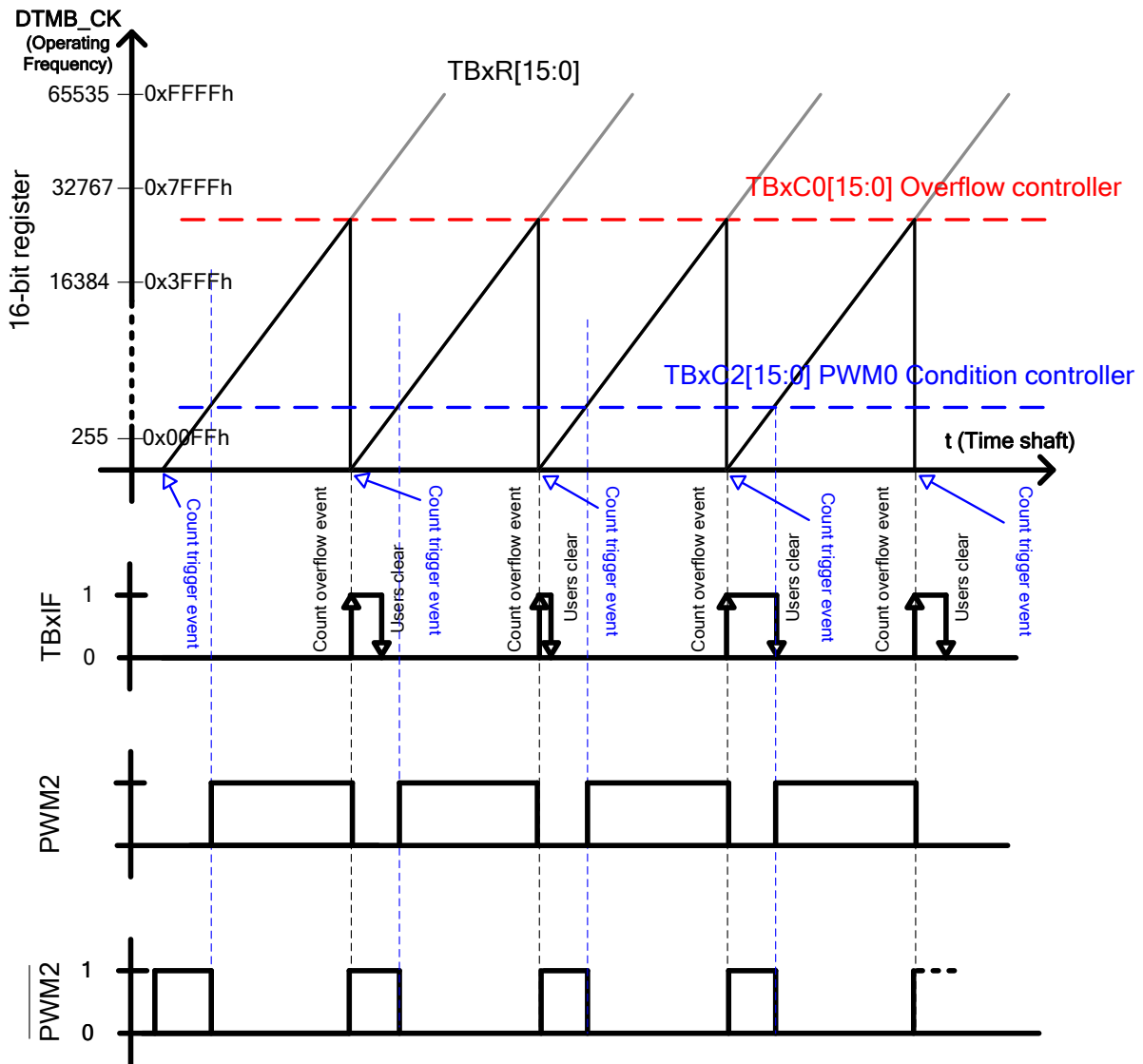


图 12-7 PWM2O 波形与使用示意图

- PWM2O 操作说明
- ◆ 初始化 (PWM 频率与工作周期设置)
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源, 设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<00>, 将 TMB1 规划为 16-bit 计数器。
 - PWMA0/1[2:0]设置<001>以输出 PWM2O 波形
 - 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用 (Always Enable), 即循环计数。
 - 写入数据至 TB1C0[15:0], 以决定 PWM 之频率。
 - 写入数据至 TB1C2[15:0], 以决定 PWM 之工作周期(Duty Cycle)。
 - 将 ENTB1[0]设置<1>启用计数器。
- ◆ 产生 PWM2O 波形
 - 当 TB1R[15:0]计数数值等于 TB1C2[15:0]时, 使得 PWM2O 状态由 0→1。
 - 当 TB1R[15:0]再计数数值等于 TB1C0[15:0]时, 使得 PWM2O 状态由 1→0; 并产生计数溢出事件使得 TB1IF[0]置<1>并归零重新递增计数, 此时 TB1IE[0]设置<1>则会产生中断事件服务。
- ◆ PWM 输出控制
 - 设置 PWMO0/1[0]设置<1>, 开启 PWM Mode。
 - 设置 PA0/1IV[0]以决定引脚输出波形是否反相。
- ◆ 将 ENTB1[0]设置<0>则关闭计数器与 PWM 输出。
- ◆ PWM2O 频率与工作周期计算公式 :

$$\text{PWM2O Frequency} = \frac{\text{DTMB_CK}}{\text{TB1C0}[15 : 0] + 1}$$

$$\text{PWM2O Duty Cycle} = \frac{(\text{TB1C0}[15 : 0] + 1) - \text{TB1C2}[15 : 0]}{\text{TB1C0}[15 : 0] + 1}$$

12.2.3. PWM3O 波形 (8-bit PWM)

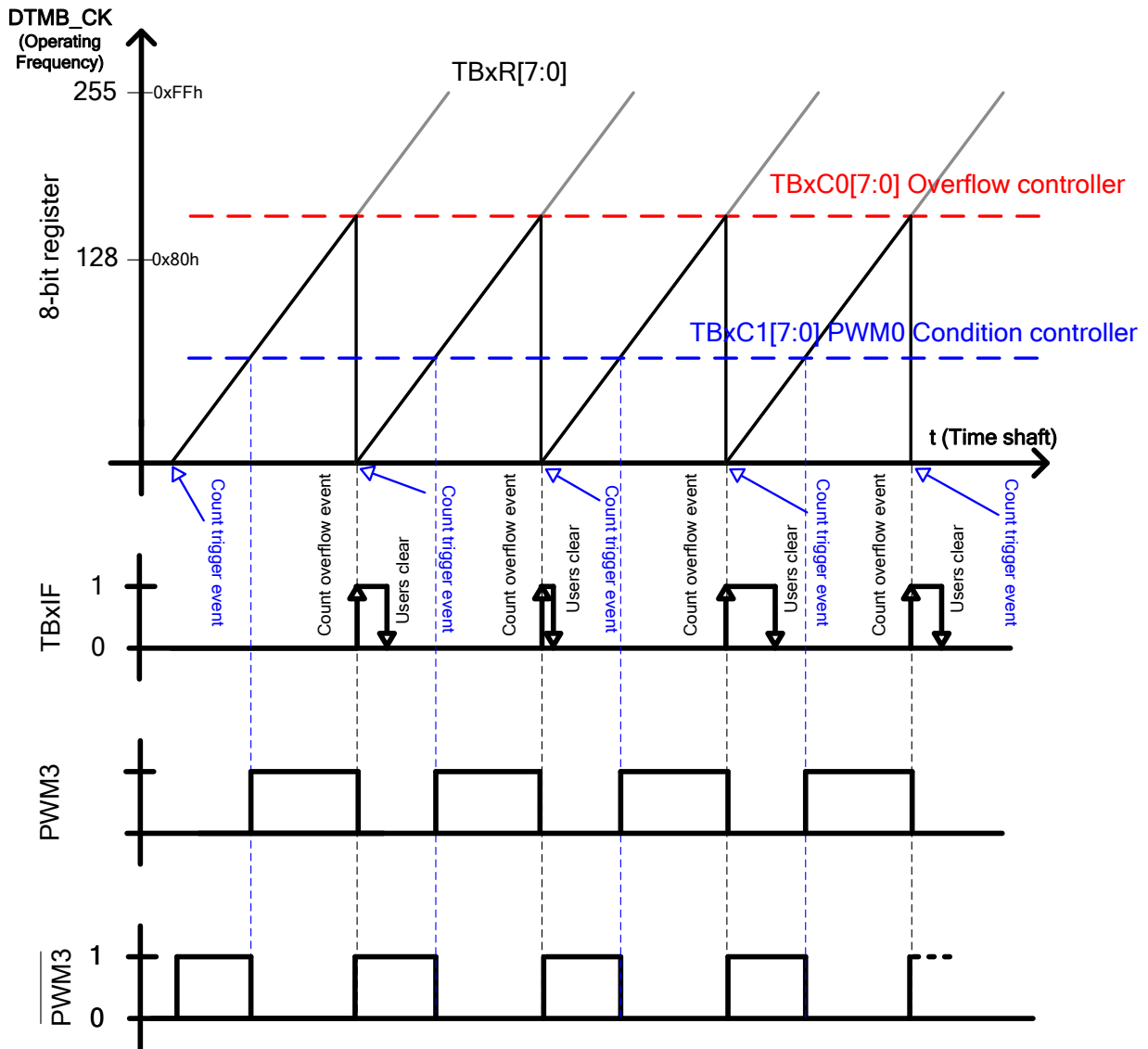


图 12-8 PWM3O 波形与使用示意图

- PWM3O 输出操作说明
- ◆ 初始化 (PWM 频率与工作周期设置)
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源, 设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<10>, 将 TMB1 规划为两组 8-bit 计数器。
 - PWMA0/1[2:0]设置<010>以输出 PWM3O 波形
 - 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用 (Always Enable), 即循环计数。
 - 写入数据至 TB1C0L[7:0], 以决定 PWM 之频率。
 - 写入数据至 TB1C1L[7:0], 以决定 PWM 之工作周期(Duty Cycle)。
 - 将 ENTB1[0]设置<1>启用计数器。
- ◆ 产生 PWM3O 波形
 - 当 TB1RL[7:0]计数数值等于 TB1C1L[7:0]时, 使得 PWM3O 状态由 0→1。
 - 当 TB1RL[7:0]再计数数值等于 TB1C0L[7:0]时, 使得 PWM3O 状态由 1→0; 并产生计数溢出事件使得 TB1IF[0]置<1>并归零重新递增计数, 此时 TB1IE[0]设置<1>则会产生中断事件服务。
- ◆ PWM 输出控制
 - 设置 PWMO0/1[0]设置<1>, 开启 PWM Mode。
 - 设置 PA0/1IV[0]以决定引脚输出波形是否反相。
- ◆ 将 ENTB1[0]设置<0>则关闭计数器与 PWM 输出。
- ◆ PWM3O 频率与工作周期计算公式 :

$$\text{PWM3O Frequency} = \frac{\text{DTMB_CK}}{\text{TB1C0L}[7 : 0] + 1}$$

$$\text{PWM3O Duty Cycle} = \frac{(\text{TB1C0L}[7 : 0] + 1) - \text{TB1C1L}[7 : 0]}{\text{TB1C0L}[7 : 0] + 1}$$

12.2.4. PWM4O 波形 (8-bit PWM)

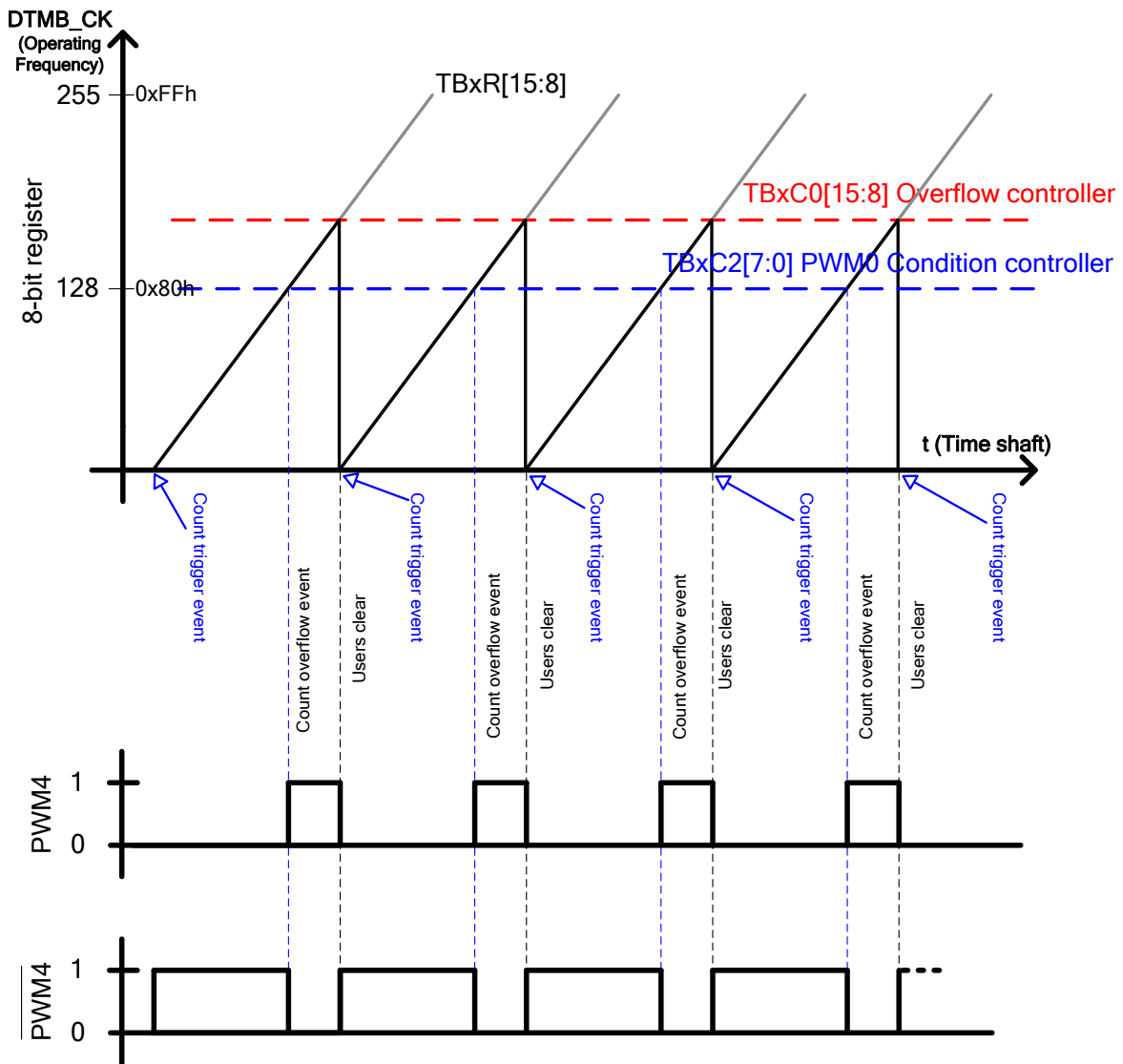


图 12-9 PWM4O 波形与使用示意图

- PWM4O 输出操作说明
- ◆ 初始化 (PWM 频率与工作周期设置)
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源, 设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<10>, 将 TMB1 规划为两组 8-bit 计数器。
 - PWMA0/1[2:0]设置<011>以输出 PWM4O 波形
 - 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用 (Always Enable), 即循环计数。
 - 写入数据至 TB1C0H[15:8], 以决定 PWM 之频率。
 - 写入数据至 TB1C2L[7:0], 以决定 PWM 之工作周期(Duty Cycle)。
 - 将 ENTB1[0]设置<1>启用计数器。
- ◆ 产生 PWM4O 波形
 - 当 TB1R[7:0]计数数值等于 TB1C2L[7:0]时, 使得 PWM4O 状态由 0→1。
 - 当 TB1RL[7:0]再计数数值等于 TB1C0H[15:8]时, 使得 PWM4O 状态由 1→0 并归零重新递增计数。
- ◆ PWM 输出控制
 - 设置 PWMO0/1[0]设置<1>, 开启 PWM Mode。
 - 设置 PA0/1IV[0]以决定引脚输出波形是否反相。
- ◆ 将 ENTB1[0]设置<0>则关闭计数器与 PWM 输出。
- ◆ PWM4O 频率与工作周期计算公式 :

$$\text{PWM4O Frequency} = \frac{\text{DTMB_CK}}{\text{TB1C0H}[15 : 8] + 1}$$

$$\text{PWM4O Duty Cycle} = \frac{(\text{TB1C0H}[15 : 8] + 1) - \text{TB1C2L}[7 : 0]}{\text{TB1C0H}[15 : 8] + 1}$$

12.2.5. PWM5O 波形 (8+8-bit PWM)

将 TMB 计数器设置在 8+8-bit 模式且 PWM 输出波形选择 PWM5O，则可得到 8+8bit PWM 输出。

8+8-bit PWM 由 TB1R[7:0]、TB1C0[7:0]、TB1C1[7:0]与 TB1C2[7:0]等控制寄存器以及内部数字电路组成。其中 TB1R[7:0]为累加计数器、TB1C0[7:0]为 PWM 频率控制器、TB1C1[7:0]为 PWM 工作周期控制器、TB1C2[7:0]为 8+8-bit PWM 工作周期微调器。

8+8-bit PWM 工作周期微调器 TB1C2[7:0]设置与说明，如下表所示。其中 N 为工作周期中的脉波宽度 (N = TB1C0[7:0] – TB1C1[7:0])。

■ 基本型

PWM 工作周期微调		说明
TB1C2[7:0]	加权量	
80h	1/2	21 个波形为一组循环，其中有(21-1)个宽度为 N+1 的波形
40h	1/4	22 个波形为一组循环，其中有(22-1)个宽度为 N+1 的波形
20h	1/8	23 个波形为一组循环，其中有(23-1)个宽度为 N+1 的波形
10h	1/16	24 个波形为一组循环，其中有(24-1)个宽度为 N+1 的波形
08h	1/32	25 个波形为一组循环，其中有(25-1)个宽度为 N+1 的波形
04h	1/64	26 个波形为一组循环，其中有(26-1)个宽度为 N+1 的波形
02h	1/128	27 个波形为一组循环，其中有(27-1)个宽度为 N+1 的波形
01h	1/256	28 个波形为一组循环，其中有(28-1)个宽度为 N+1 的波形

表 12-1 工作周期微调器设置表

■ 逻辑运算 OR 迭合型

- 当 TB1C2[7:0]不只设置 1bit 时，其加权量进行加总后，即为总加权量。亦为一组循环中，各脉波宽度(N, N+1)的数量。

$$\text{加权量} = \frac{\alpha}{\beta} \quad \begin{array}{l} \alpha = \text{一组循环中，脉波宽度为 N 的波形数} \\ \beta = \text{波一组循环中的总波形数} \end{array}$$

以下范例以随机数值进行规律说明。

- TB1C2[7:0]设置为 C0h(80h+40h)时，会使得 PWM 工作周期产生 3/4(1/2+ 1/4)的加权量变化。而波形的变化，则是以 4 个输出周期为一组，其中会有 3 个宽度 N 的波形与 1(4-3)个宽度为(N+1)的波形。
- TB1C2[7:0]设置为 A0h(80h+20h)时，会使得 PWM 工作周期产生 5/8(1/2+ 1/8)的加权量变化。而波形的变化，则是以 8 个输出周期为一组，其中会有 5 个宽度 N 的波形与 3(8-5)个宽度为(N+1)的波形。
- TB1C2[7:0]设置为 57h(40h+10h+04h+02h+01h)时，会使得 PWM 工作周期产生 87/256 (1/4+ 1/16+ 1/64+ 1/128+ 1/256)的加权量变化。而波形的变化，则是以 256 个输出周期为一组，其中会有 87 个宽度 N 的波形与(256-87)个宽度为(N+1)的波形。
- TB1C2[7:0]设置为 86h(80h+04h+02h)时，会使得 PWM 工作周期产生 67/128(1/2+ 1/64+ 1/128)的加权量变化。而波形的变化，则是以 128 个输出周期为一组，其中会有 67 个宽度 N 的波形与(128-67)个宽度为(N+1)的波形。

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

- TB1C2[7:0]设置为 FFh(80h+40h+20h+10h+08h+04h+02h+01h)时, 会使得 PWM 工作周期产生 255/256 的加权重变化。而波形的变化, 则是以 256 个输出周期为一组, 其中会有 255 个宽度 N 的波形与 1 个宽度为(N+1)的波形。

- 下表 12 -2、图 12-10 与图 12-10 部分列出 TB1C2[7:0]在不同设置下, 8+8-bit PWM 波形变化以供使用者参考。

形态	TB1C2 [7:0]	加重量	0	1	2	3	4	5	6	7	8	9	~	127	128	129	~	254	255	
基本波形	00h	-	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	~	N+1	N+1	N+1	~	N+1	N+1	
	80h	1/2	N+1	N	N+1	N	N+1	N	N+1	N	N+1	N	~	N	N+1	N	~	N+1	N	
	40h	1/4	N+1	N+1	N	N+1	N+1	N+1	N	N+1	N+1	N+1	~	N+1	N+1	N+1	~	N	N+1	
	20h	1/8	N+1	N+1	N+1	N+1	N	N+1	N+1	N+1	N+1	N+1	~	N+1	N+1	N+1	~	N+1	N+1	
	10h	1/16	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N	N+1	~	N+1	N+1	N+1	~	N+1	N+1	
	08h	1/32	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	~	N+1	N+1	N+1	~	N+1	N+1
	04h	1/64	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	~	N+1	N+1	N+1	~	N+1	N+1
	02h	1/128	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	~	N+1	N+1	N+1	~	N+1	N+1
	01h	1/256	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	N+1	~	N+1	N	N+1	~	N+1	N+1
逻辑运算迭合型	C0h	3/4	N+1	N	N	N	N+1	N	N	N	N+1	N	~	N	N+1	N	~	N	N	
	A0h	5/8	N+1	N	N+1	N	N	N	N+1	N	N+1	N	~	N	N+1	N	~	N+1	N	
	E0h	7/8	N+1	N	N	N	N	N	N	N	N+1	N	~	N	N+1	N	~	N	N	
	F0h	15/16	N+1	N	N	N	N	N	N	N	N	N	~	N	N+1	N	~	N	N	
	F8h	31/32	N+1	N	N	N	N	N	N	N	N	N	~	N	N+1	N	~	N	N	
	FCh	63/64	N+1	N	N	N	N	N	N	N	N	N	~	N	N+1	N	~	N	N	
	FEh	127/128	N+1	N	N	N	N	N	N	N	N	N	~	N	N+1	N	~	N	N	
	FFh	255/256	N+1	N	N	N	N	N	N	N	N	N	~	N	N	N	~	N	N	
	57h	87/256	N+1	N+1	N	N+1	N+1	N+1	N	N+1	N	N+1	~	N+1	N	N+1	~	N	N+1	
	86h	67/128	N+1	N	N+1	N	N+1	N	N+1	N	N+1	N	~	N	N+1	N	~	N+1	N	
	32h	25/128	N+1	N+1	N+1	N+1	N	N+1	N+1	N+1	N	N+1	~	N+1	N+1	N+1	~	N+1	N+1	

表 12 -2 PWM50 输出波形示意图

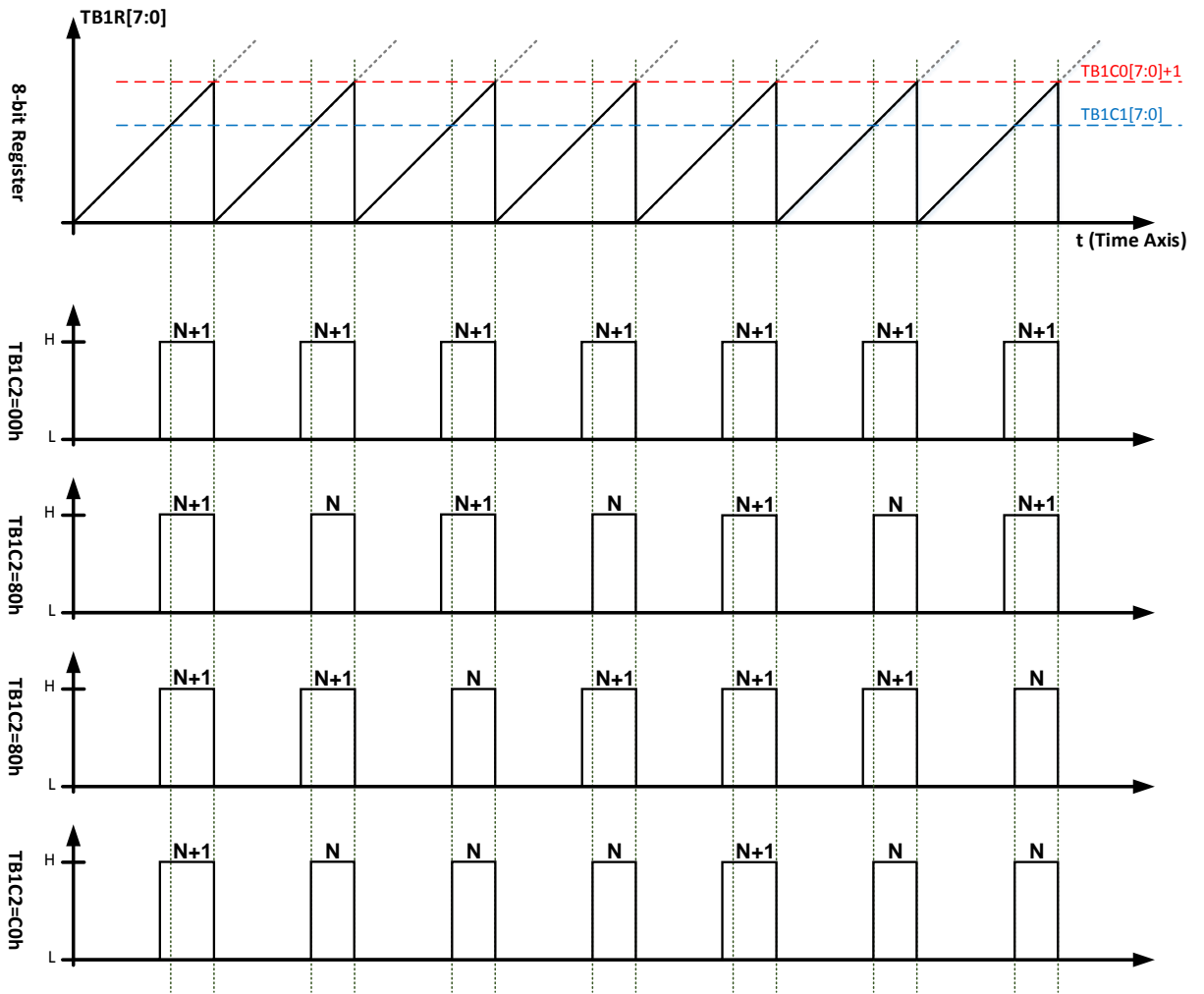


图 12-10 PWM5O 输出波形示意图 1

- PWM5O 输出操作说明
- ◆ 初始化 (PWM 频率与工作周期设置)
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源, 设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<11>, 将 TMB1 规划为 8+8-bit 计数器。
 - PWMA0/1[2:0]设置<100>以输出 PWM5O 波形
 - 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用 (Always Enable), 即循环计数。
 - 写入数据至 TB1C0L[7:0], 以决定 PWM 之频率。
 - 写入数据至 TB1C1L[7:0], 以决定 PWM 之工作周期(Duty Cycle)。
 - 写入数据至 TB1C2L[7:0], 以决定 PWM 之工作周期(Duty Cycle)微调方式。
 - 将 ENTB1[0]设置<1>启用计数器。
- ◆ 产生 PWM5O 波形
 - 当 TB1RL[7:0]计数数值等于 TB1C1L[7:0]时, 使得 PWM5O 状态由 0→1。
 - 当 TB1RL[7:0]再计数数值等于 TB1C0L[7:0]时, 使得 PWM5O 状态由 1→0 ;
 - 并产生计数溢出事件使得 TB1IF[0]置<1>并归零重新递增计数, 此时 TB1IE[0]设置<1>则会产生中断事件服务。
 - 此时, TB1C2L[7:0]所设置的数据, 将调整 PWM5O 输出之脉波宽度为 N+1 或 N。如表 12-1 所描述, 其中 $N = TB1C0[7:0] - TB1C1[7:0]$ 。
- ◆ PWM 输出控制
 - 设置 PWMO0/1[0]设置<1>, 开启 PWM Mode。
 - 设置 PA0/1IV[0]以决定引脚输出波形是否反相。
- ◆ 将 ENTB1[0]设置<0>则关闭计数器与 PWM 输出。
- ◆ PWM5O 频率与工作周期计算公式 :

$$\text{PWM5O Frequency} = \frac{DTMB_CK}{TBxCO[7:0] + 1}$$

$$\text{PWM5O Duty Cycle} = \frac{(TBxCO[7:0] + 1) - TBxC1[7:0] - TBxC2[7:0]/256}{TBxCO[7:0] + 1}$$

12.2.6. PWM6O 形 (两个 16-bit PWM 波形)

将 TMB 计数器设置在 17-bit 模式且 PWM 输出波形选择 PWM6O 则可产生两个 16-bit PWM 波形。

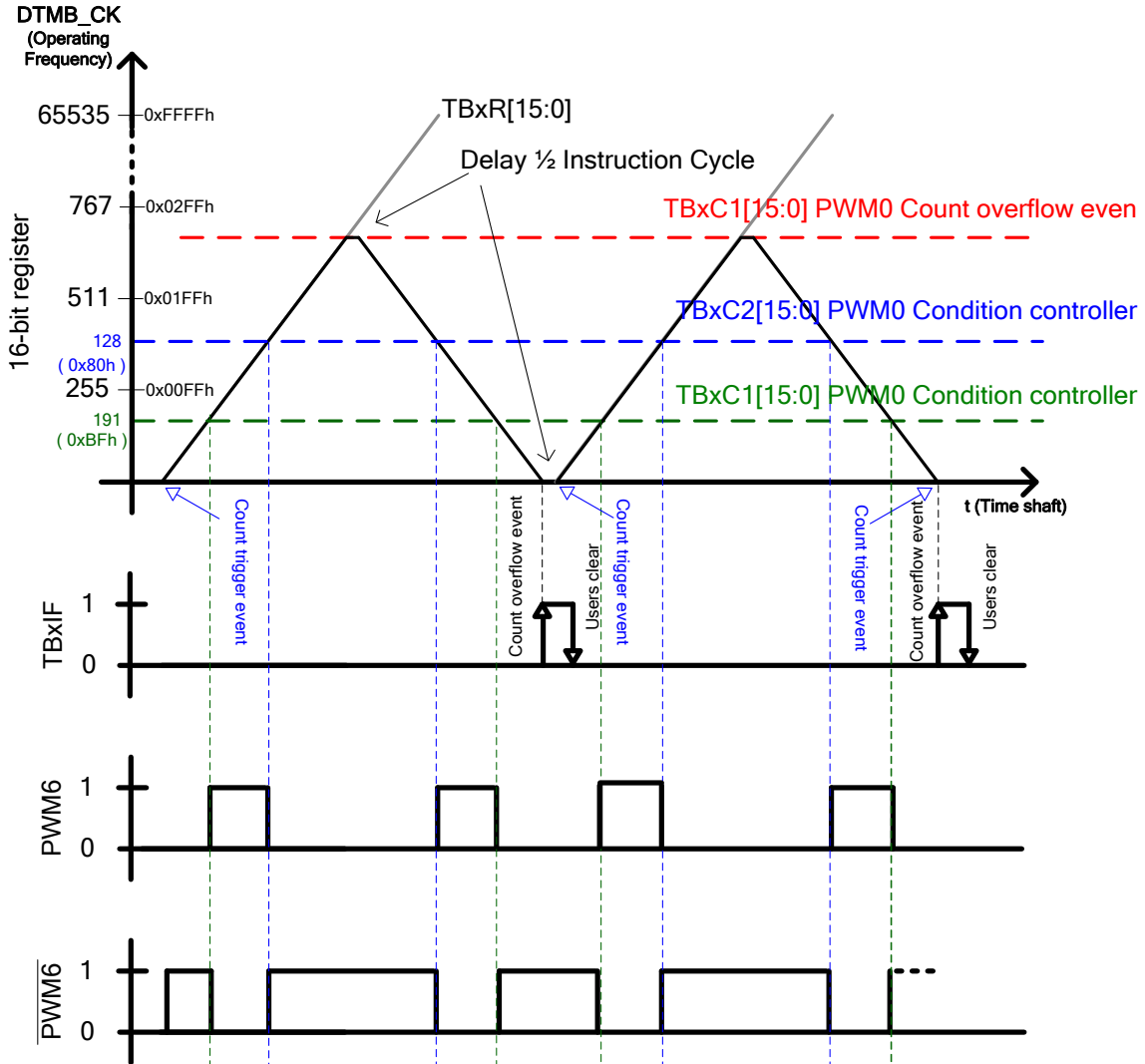


图 12-11 PWM6O 形与使用示意图

- 17-bit PWM 输出操作说明
- ◆ 初始化 (PWM 频率与工作周期设置)
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源, 设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<01>, 将 TMB1 规划为 17-bit 计数器。
 - PWMA0/1[2:0]设置<101>以输出 PWM6O 波形
 - 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用 (Always Enable), 即循环计数。
 - 写入数据至 TB1C0H[15:8], 以决定 PWM 之频率。
 - 写入数据至 TB1C1L[15:0]与 TB1C2[15:0], 以决定 PWM 之工作周期(Duty Cycle)。
 - 将 ENTB1[0]设置<1>启用计数器。
- ◆ 产生双波形 PWM6O 形
 - 第一波形条件
 - ✓ 当 TB1R[15:0]递增计数数值等于 TB1C1[15:0]时, 使得 PWM6O 状态由 0→1。
 - 当 TB1R[15:0]再递增计数数值等于 TB1C2[15:0]时, 使得 PWM6O 状态由 1→0。
 - ✓ 接着, 当 TB1R[15:0]计数数值等于 TB1C0[15:0]时, 使得 TB1R[15:0]转为递减计数。
 - 第二波形条件
 - ✓ 当 TB1R[15:0]递增计数数值等于 TB1C2[15:0]时, 使得 PWM6O 状态由 0→1。
 - ✓ 当 TB1R[15:0]再递增计数数值等于 TB1C1[15:0]时, 使得 PWM6O 状态由 1→0。
 - ✓ 接着, 当 TB1R[15:0]计数数值等于 0x0000h 时产生计数溢出事件使得 TB1IF[0]置<1>归零重新递增计数, 此时 TB1IE[0]设置<1>则会产生中断事件服务。
- ◆ PWM 输出控制
 - 设置 PWMA0/1[0]设置<1>, 开启 PWM Mode。
 - 设置 PA0/1IV[0]以决定引脚输出波形是否反相。
- ◆ 将 ENTB1[0]设置<0>则关闭计数器与 PWM 输出。
- ◆ PWM6O 率与工作周期计算由于产生的波形特殊, 在此不描述。

12.2.7. PWM7O 波形 (16-bit PWM 波形)

将 TMB 计数器设置在 16-bit 模式且 PWM 输出波形选择 PWM7O，则可产生周期性的 PWM 波形。

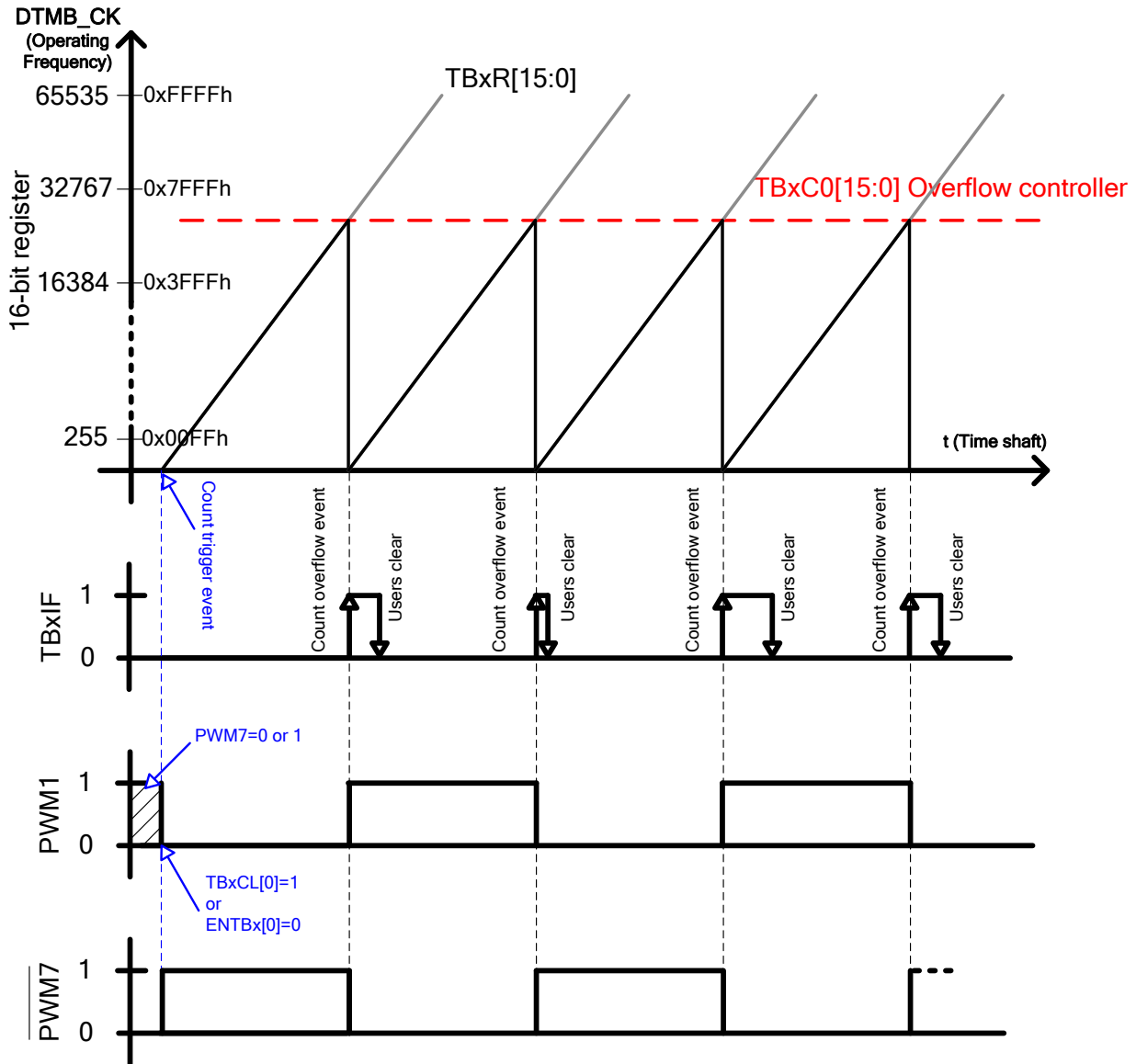


图 12-12 PWM7O 波形与使用示意图

- PWM7O 操作说明
- ◆ 初始化 (PWM 频率与工作周期设置)
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源，设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<00>，将 TMB1 规划为 16-bit 计数器。
 - PWMA0/1[2:0]设置<111>以输出 PWM7O 波形
 - 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用 (Always Enable)，即循环计数。
 - 写入数据至 TB1C0[15:0]，以决定 PWM 之频率。
 - 将 ENTB1[0]设置<1>启用计数器。
- ◆ 产生 PWM7O 波形

- TMB1 未启用时 PWM7O 状态未定 ,但当 ENTB1[0]设置<1>或者 TB1CL[0]设置<1>时 ,PWM7O 输出 0 直至发生计数溢出事件后 PWM7O 转态输出 1 且于下次再发生计数溢出事件时转态为 0 , 产生周期性波形。
 - 当 TB1R[15:0]再计数数值等于 TB1C0[15:0]时 ,使得 PWM7O 转态 ;并产生计数溢出事件使得 TB1IF[0]置<1>并归零重新递增计数 ,此时 TB1IE[0]]设置<1>则会产生中断事件服务。
 - ◆ PWM 输出控制
 - 设置 PWMO0/1[0]设置<1> , 开启 PWM Mode。
 - 设置 PA0/1IV[0]以决定引脚输出波形是否反相。
 - ◆ 将 ENTB1[0]设置<0>则关闭计数器与 PWM 输出。
 - ◆ PWM7O 频率与工作周期计算公式 :
- $$\text{PWM7O Frequency} = \frac{\text{DTMB_CK}}{\text{TBxC0}[15 : 0] + 1} \div 2$$
- $$\text{PWM7O Duty Cycle} = 50\%$$

12.3. TMB1 控制寄存器列表与说明：

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	-	ADIE	WDTIE	TB1IE	TMAIE	E1IE	EOIE	0000 0000	0uuu uuuu	*****
INTF0	-				TB1IF				.000 0000	.uuu uuuu	*****
OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]		CUPS		0000 0000	uuuu uuuu	*****
OSCCN1		LCPS			DTMB[1:0]		TMBS		0000 0000	uuuu uu.	*****
OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	*****r
BSRCN	-	-	-	-	-	-	BSR[1]	BSR[0] xxxx uuuu*
TB1Flag	-	PWM7A	PWM6A	PWM5A	PWM4A	PWM3A	PWM2A	PWM1A	.00 0000	.uu uuuu	.,.,.,.,.,.,.
TB1CN0	ENTB1	TB1M[1:0]		TB1RT[1:0]		TB1CL	PWMO1	PWMO0	0000 0000	uuuu u0uu	*****r,w1,*
TB1CN1	PA1IV	PWMA1[2:0]		PA0IV	PWMA0[2:0]				0000 0000	uuuu uuuu	*****
TB1RH	TimerB1 counter Register [15:8]								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r
TB1RL	TimerB1 counter Register [7:0]								xxxx xxxx	uuuu uuuu	r,r,r,r,r,r,r,r
TB1C0H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*****
TB1C0L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*****
TB1C1H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*****
TB1C1L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*****
TB1C2H	TimerB1 counter Condition Register [15:8]								xxxx xxxx	uuuu uuuu	*****
TB1C2L	TimerB1 counter Condition Register [7:0]								xxxx xxxx	uuuu uuuu	*****
TCCN0	-	TC1S[1:0]		-	-	-	-	-	0000 0000	uuuu uuuu	*****

表 12-3 TMB1/2/3 相关寄存器

BSRCN: 详见 内存章节

INTE0/INTF0: 详见 中断 章节

OSCCN0/OSCCN1/OSCCN2: 详见 錯誤! 找不到參照來源。 章节

TB1Flag: 定时计数器 TMB1 产生 PWM 波形状态标志

位	名称	描述
Bit6~0	PWMxA	PWMx 波形状态, $1 \leq x \leq 7$ <0> 低电位 L <1> 高电位 H

TB1CN0: 定时计数器 TMB1 控制寄存器 0

位	名称	描述
Bit7	ENTB1	启用与关闭 TMB1 <0> 关闭。 <1> 启用
Bit6~5	TB1M[1:0]	计数器 TMB1 操作模式 <00> 16-bit 计数器 <01> 17-bit 计数器 <10> 两组 8-bit 计数器 <11> 8+8-bit 计数器

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述
Bit4~3	TB1RT[1:0]	计数器 TMB1 计数触发选择器 <00> Logic High <11> CPI1
Bit2	TB1CL	TB1R 计数归零控制器 <0> 计数不归零。 <1> 计数归零。(设置<1>有效,计数器归零后自动置<0>)
Bit1	PWMO1	PWM1 引脚输出控制器 <0> 不输出 <1> 输出
Bit0	PWMO0	PWM0 引脚输出控制器 <0> 不输出 <1> 输出

TB1CN1: 计数器 TMB1 控制寄存器 1

位	名称	描述				
Bit7	PA1IV	引脚 PWMAx 波形输出相位 ($0 \leq x \leq 1$) <0> 反相。 <1> 同相。				
Bit3	PA0IV					
Bit6~4	PWMA1[2:0]	引脚 PWMAx 波形输出选择器 ($0 \leq x \leq 1$)				
Bit2~0	PWMA0[2:0]		PWMAx[2:0]	输出选择器	PWMAx[2:0]	输出选择器
			000	PWM1O	100	PWM5O
			001	PWM2O	101	PWM6O
			010	PWM3O	110	PWM7O
011	PWM4O	111	PWM7O			

TB1R: TMB1 计数值寄存器

位	名称	描述
Bit15~8	TB1RH[7:0]	TMB1 计数器
Bit7~0	TB1RL[7:0]	

TB1C0: TMB1 计数溢出控制

位	名称	描述
Bit15~8	TB1C0RH[7:0]	TMB1 计数溢出控制
Bit7~0	TB1C0RL[7:0]	

TB1C1: PWM0 占空比控制寄存器

位	名称	描述

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述
Bit15~8	TB1C1RH[7:0]	PWM0 占空比控制
Bit7~0	TB1C1RL[7:0]	

TB1C2: PWM1 占空比控制寄存器

位	名称	描述
Bit15~8	TB1C2RH[7:0]	PWM1 占空比控制
Bit7~0	TB1C2RL[7:0]	

TC1CN0: 定时计数器 TMC 控制寄存器 0

位	名称	描述
Bit6~5	TC1S[1:0]	计数器 TC1 事件输入选择器 <00> TBI2 来自 GPIO 口的输入。(预设) <01> TBI1 来自 GPIO 口的输入。 <10> 低频频率源 LPC_CK <11> TBI0 来自 GPIO 口的输入。

13.16-bit 定时计数器 C, TMC (16-bit TimerC)

定时计数器 C 应用于信号捕捉与比较, 使用时须配合 TMB1 的 TB1R[15:0]计数器。其透过输入信号选择器 CPI1S[1:0]与 CPSS[0]可组合而成不同的信号捕捉应用方式, 并利用信号分频器 CP1PS[3:0]与触发型态选择器 CPI1P[0]与 CPI2P[0], 在判断条件成立下使 TB1R[15:0]计数器数值分别送至 TMC 条件成立寄存器 TC1R0[15:0]及 TC1R1[15:0], 并分别产生中断事件 TC0IF[0]与 TC1IF[0]。

TMC 寄存器摘要 :

INTE0	GIE
INTE2	TC11IE, TC10IE
INTF2	TC11F, TC01F
TC1CN0	CPI1P, CPI0P, TCEN
TC1CN1	CPSS, CPI1S[1:0], CP1PS[3:0]
TC1R0	TC1R0H[7:0], TC1R0L[7:0]
TC1R1	TC1R1H[7:0], TC1R1L[7:0]

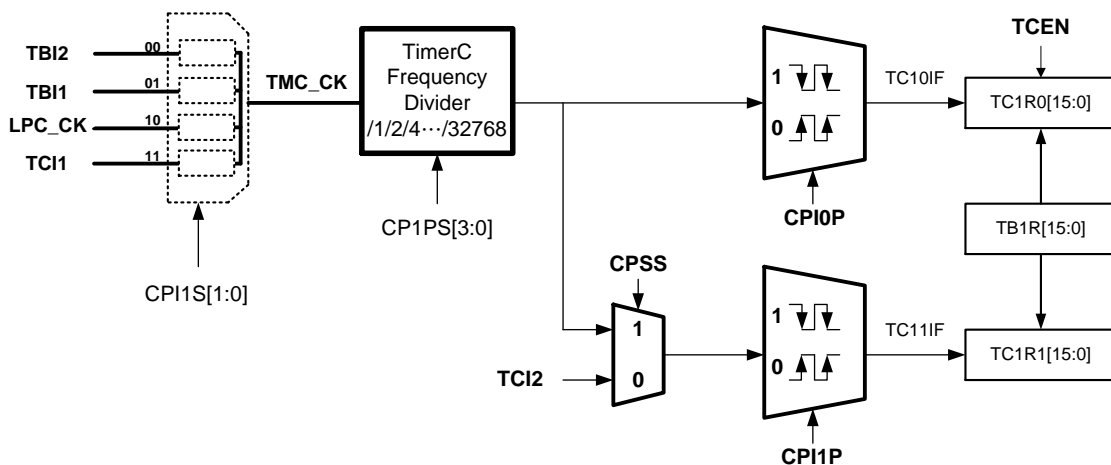


图 13-1 TMC 计数器架构图

- TMC 操作说明
- ◆ 初始化 TMB1
 - 设置 TMBS[1:0]可选择 TMB 的工作频率源, 设置 DTMB[1:0]以决定 TMB 工作频率。
 - TB1M[1:0]设置<00>, 将 TMB1 规划为 16-bit 计数器。
 - 写入数据至 TB1C0[15:0]。
 - 将 TB1RT[1:0]设置<00>以选择触发计数信号为总是启用 (Always Enable), 即循环计数。
 - 将 ENTB1[0]设置<1>以启用计数器
 - ✓ 当 TB1R[15:0]计数数值等于 TB1C0[15:0]时, 产生计数溢出事件使得 TB1IF[0]置<1>并归零重新递增计数, 此时 TB1IE[0]设置<1>则会产生中断事件服务。
 - ✓ 计数过程, 使用者可利用计数归零控制器 TB1CL[0]设置<1>以重新计数, 且 TB1CL[0]自动置<0>。
- ◆ 初始化 TMC
 - 设置 CPI1S[1:0]与 CPSS[0]以决定捕捉的信号源及信道
 - 设置 CP1PS[3:0]以决定 TC0 所需的捕捉信号分频频率。

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

- 设置 CPI0P[0]与 CPI1P[0]以决定捕捉信号为上升缘或下降缘。
- 设置输入 TMCi0 与 TMCi1 之引脚为输入状态，并确认引脚相关设置是否正确。
- 将 TCEN[0]设置<1>以启用 TMC。
- ◆ TMC 捕捉条件成立使得捕捉中断事件发生 TC0IF[0]置<1>或 TC1IF[0]置<1>，此时 TC10IE[0]置<1>或 TC11IE[0]置<1>则会产生中断事件。（比较事件成立至 TC0IF/TC1IF 转态时间约 $4 \cdot DTMB_CK$ ，即是 $DTMB_CK=32768\text{Hz}$ 则转态延迟时间约为 $122\mu\text{s}$ ）
- ◆ 将 ENTB1[0]设置<0>与 TCEN[0] 设置<0>则关闭 TMB1 与 TMC 计数器。
- ◆ 捕捉的信号限制为<TimerB/8

13.1. 寄存器说明-TMC

“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE								0000 0000	0uuu uuuu	*****
INTE2			TC11IE	TC10IE					0000 0000	uuuu uuuu	*****
INTF2			TC11F	TC01F					0000 0000	uuuu uuuu	*****
TC1CN0						CP1P	CP0P	TCEN	0000 0000	uuuu uuuu	uuuu uuuu
TC1CN1		CPSS	CP1S[1:0]		CP1PS[3:0]				0000 0000	uuuu uuuu	uuuu uuuu
TC1R0H	Capture 0 High Byte Data Register								xxxx xxxx	uuuu uuuu	uuuu uuuu
TC1R0L	Capture 0 Low Byte Data Register								xxxx xxxx	uuuu uuuu	uuuu uuuu
TC1R1H	Capture 1 High Byte Data Register								xxxx xxxx	uuuu uuuu	uuuu uuuu
TC1R1L	Capture 1 Low Byte Data Register								xxxx xxxx	uuuu uuuu	uuuu uuuu

表 13-1 TMC 相关寄存器

INTE0/INTE2/INTF2: 详见 中断 章节

TC1CN0:定时计数器 TimerC1 控制寄存器 0

位	名称	描述
Bit2	CP1P	Capture1 触发源设置 <0>上升源触发。(预设) <1>下降源触发。
Bit1	CP10P	Capture0 触发源设置 <0>上升源触发。(预设) <1>下降源触发。
Bit0	TCEN	启用与关闭 TMC1 <0>关闭。(但不清除 TC1R0 及 TC1R1) (预设) <1>启用

TC1CN1:定时计数器 TimerC1 控制寄存器 1

位	名称	描述
Bit6	CPSS	Capture 1 (Timer C Channel 2)捕捉触发源选择 <0>TCI2 来自 GPIO 口的输入。(预设) <1>与 Capture 0 (Timer C Channel 1, CP1)一样的捕捉触发源。
Bit5~4	CP1S[1:0]	Capture 0 (Timer C Channel 1)捕捉触发源选择 <00>TBI2 来自 GPIO 口的输入。(预设) <01>TBI1 来自 GPIO 口的输入。 <10>低频频率源 LPC_CK。 <11> TCI1 来自 GPIO 口的输入。

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述			
Bit3~0	CP1PS[3:0]	Capture1 触发源的分频器设置			
		CP1PS[3:0]	CP1PS[3:0]		
		0000	TMC_CK/1 (预设)	1000	TMC_CK/256
		0001	TMC_CK/2	1001	TMC_CK/512
		0010	TMC_CK/4	1010	TMC_CK/1024
		0011	TMC_CK/8	1011	TMC_CK/2048
		0100	TMC_CK/16	1100	TMC_CK/4096
		0101	TMC_CK/32	1101	TMC_CK/8192
		0110	TMC_CK/64	1110	TMC_CK/16384
		0111	TMC_CK/128	1111	TMC_CK/32768

TC1R0: Capture 0 捕捉计数器

位	名称	描述
Bit15~8	TC1R0H[7:0]	TMC1 Capture 1 捕捉计数器
Bit7~0	TC1R0L[7:0]	

TC1R1: Capture 1 捕捉计数器

位	名称	描述
Bit15~8	TC1R1H[7:0]	TMC1 Capture 2 捕捉计数器
Bit7~0	TC1R1L[7:0]	

14. 电源系统, Power System

电源系统 PWR 具备一个线性稳压电源 VDDA 以及模拟电路共地电源 ACM，其提供芯片模拟外围电路使用并可适当的用来驱动外部电路。

PWR 寄存器摘要：

PWRCN ENBGR[0], LDOC[2:0], LDOM[1:0], ENLDO[0]

AD1CN5 LDOPL[0]

BIACN0 ENREFO[0]

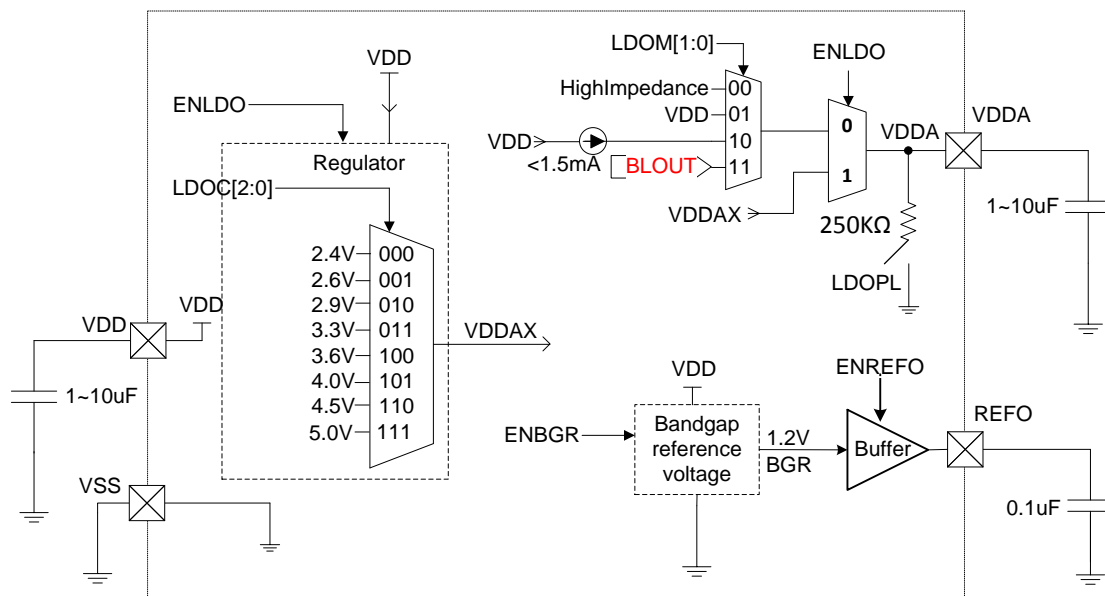


图 14-1 Power System 方块图

14.1. VDDA 使用说明

14.1.1. VDDA 初始化设置：

稳压电压值选择器 LDOC[2:0] 可设置 VDDA 引脚输出的电压计可由 2.4V~4.5V，共有 7 段电压。由于 VDDA 为一线性稳压电源，使用时必须注意 VDD 工作电压的电压值是否低于 VDDA 输出电压的设定值以免造成不可预期的电路误动作。

14.1.2. VDDA 使用外部偏压：

VDDA 可采用外部输入电压设计，当用户欲自行提供电压源则必须由 VDDA 引脚外灌电压方式输入。采用此方式时必须关闭 VDDA，即 LDOM [1:0] 设置 00。必须注意，此使用方式可能会影响模拟电路的效能故需谨慎。

14.1.3. VDDA 启用

ENLDO[0] 设置 <1> 则会启用 VDDA 稳压器。启动 VDDA 稳压器须避免 Σ ADC 处于启用状态，而且需要等到 VDDA 电压稳定后才可以启用 Σ ADC。当外接 1uF(10uF) 稳压电容时约需要 500uS(5mS) 的稳定时间。

14.2. 寄存器说明-PWR

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
PWRCN	ENBGR	LDOC[2:0]			LDO[0]	LDO	ENLDO		0000 0000	uuuu u00u	*,*,*,*,*,wr0,wr0,*
AD1CN5				LDOPL		-			0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*,*
BIACN0								ENREFO	0000 0000	uuuu uuuu	*,*,*,*,*,*,*,*,*

表 14-1 PWR 寄存器

PWRCN: 电源系统控制寄存器

位	名称	描述																				
Bit7	ENBGR	内部参考电压控制器 <0>关闭 <1>启用, 当开启 ADC 及 TPS 时, 必须先设为'1', 后再开启。 此 bit 与 HAO 为连动, 故只要 HAO 为开启的。就算此 bit 写 0, 实际上 BGR 还是开启的																				
Bit6~4	LDOC[2:0]	VDDAX 由 LDO 稳压输出电压选择器 当 ENLDO 为'1'时, 此设定电压才会输出至 VDDA 接脚上。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LDOC[2:0]</th> <th>VDDAX 输出电压</th> <th>LDOC[2:0]</th> <th>VDDAX 输出电压</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>2.4V</td> <td>100</td> <td>3.6V</td> </tr> <tr> <td>001</td> <td>2.6V^{*1}</td> <td>101</td> <td>4.0V^{*1}</td> </tr> <tr> <td>010</td> <td>2.9V^{*1}</td> <td>110</td> <td>4.5V^{*1}</td> </tr> <tr> <td>011</td> <td>3.3V^{*1}</td> <td>111</td> <td>Reserved</td> </tr> </tbody> </table> ^{*1} HY17P52 不具有此选项	LDOC[2:0]	VDDAX 输出电压	LDOC[2:0]	VDDAX 输出电压	000	2.4V	100	3.6V	001	2.6V ^{*1}	101	4.0V ^{*1}	010	2.9V ^{*1}	110	4.5V ^{*1}	011	3.3V ^{*1}	111	Reserved
LDOC[2:0]	VDDAX 输出电压	LDOC[2:0]	VDDAX 输出电压																			
000	2.4V	100	3.6V																			
001	2.6V ^{*1}	101	4.0V ^{*1}																			
010	2.9V ^{*1}	110	4.5V ^{*1}																			
011	3.3V ^{*1}	111	Reserved																			
Bit3~2	LDO[1:0]	VDDA 电压来源选择器 当 ENLDO 为'0'时, 此设定才会输出至 VDDA 接脚上。 <00> 关闭具高输入阻抗模式 (VDDA 由内部 LDO 输出或外部输入选择此模式) <01> 输出 VDD 电压 <10> Pull high to VDD by 1.5mA. (It is use to initial VDDA when a small current) <11> BLOUT(功能仅在 ICE 才可以实现, HY17P58 不带有该功能!)																				
Bit1	ENLDO	内部线性稳压器控制器 <0>关闭 <1>启用																				

AD1CN5: Σ ADC 控制寄存器 5

位	名称	描述
Bit4	LDOPL	VDDA 输出端内部 250k Ω 电阻下拉开关 <0>关闭 (预设) <1>启用。

HY17S58 Emulate Chip User' Guide

Embedded $\Sigma\Delta$ ADC
8-Bit RISC-like Mixed Signal Microcontroller

		以下几点务必使 LDOPL 为 1，否则结果会不如预期 ※ 使用内部 LDO 输出 ※ ADC 参考电压选用 VDDA/2-VSS
--	--	---

BIACN0: Peak Hold 控制寄存器

位	名称	描述
Bit0	ENREFO	REFO 电压源输出控制 <0> 关闭，处于高阻态 (预设) <1> 电压源输出.

15. 模拟数字转换器, $\Sigma\Delta$ ADC

$\Sigma\Delta$ ADC 为高分辨率超采样和差型模拟数字转换器(Over Sampling Sigma Delta Analog-to-Digital Converter), 具有 24 位的输出。其包含多功能的输入多任务器、输入缓冲器(Input Buffer)与前置低噪声放大器(PGA, Programmable Gain Amplifier)、 $\Sigma\Delta$ 调变器 ($\Sigma\Delta$ AD, Sigma Delta Modulator)、梳状滤波器(Comb Filter)等 4 部分。

- 多功能的输入多任务器
 - 可切换选择多组不同的输入信道,单一芯片可做多种量测
 - 输入通道可做短路, 消除 ADC 的零点偏移
 - 内置温度感测电路输出电压
- $\Sigma\Delta$ 调变器
 - 可调整输入电压放大倍率, 倍率为 1/4~16 倍
 - 可选择参考电压的倍率为 1 或 1/2
 - 4 位的直流输入偏压设定
- 梳状滤波器(Comb filter)
 - 可调整 OSR(Over Sampling Ratio)= 64~65536
 - 支持 2nd + 3rd 架构
 - 产生中断事件

$\Sigma\Delta$ ADC 寄存器摘要 :

AD1CN0	ENAD1, OSR[3:0], CMFR
AD1CN1	VREGN, PGAGN[1:0], ADGN[2:0]
AD1CN2	DCSET[3:0]
AD1CN3	INP[3:0], INN[3:0]
AD1CN4	VRH[1:0], VRL[1:0], INX[1:0], VRIS, INIS
AD1CN5	ENACM, ENV12, VCMS, LDOPL, ENTPS, TPSCH

HY17S58 Emulate Chip User' Guide

Embedded $\Sigma\Delta$ ADC
8-Bit RISC-like Mixed Signal Microcontroller

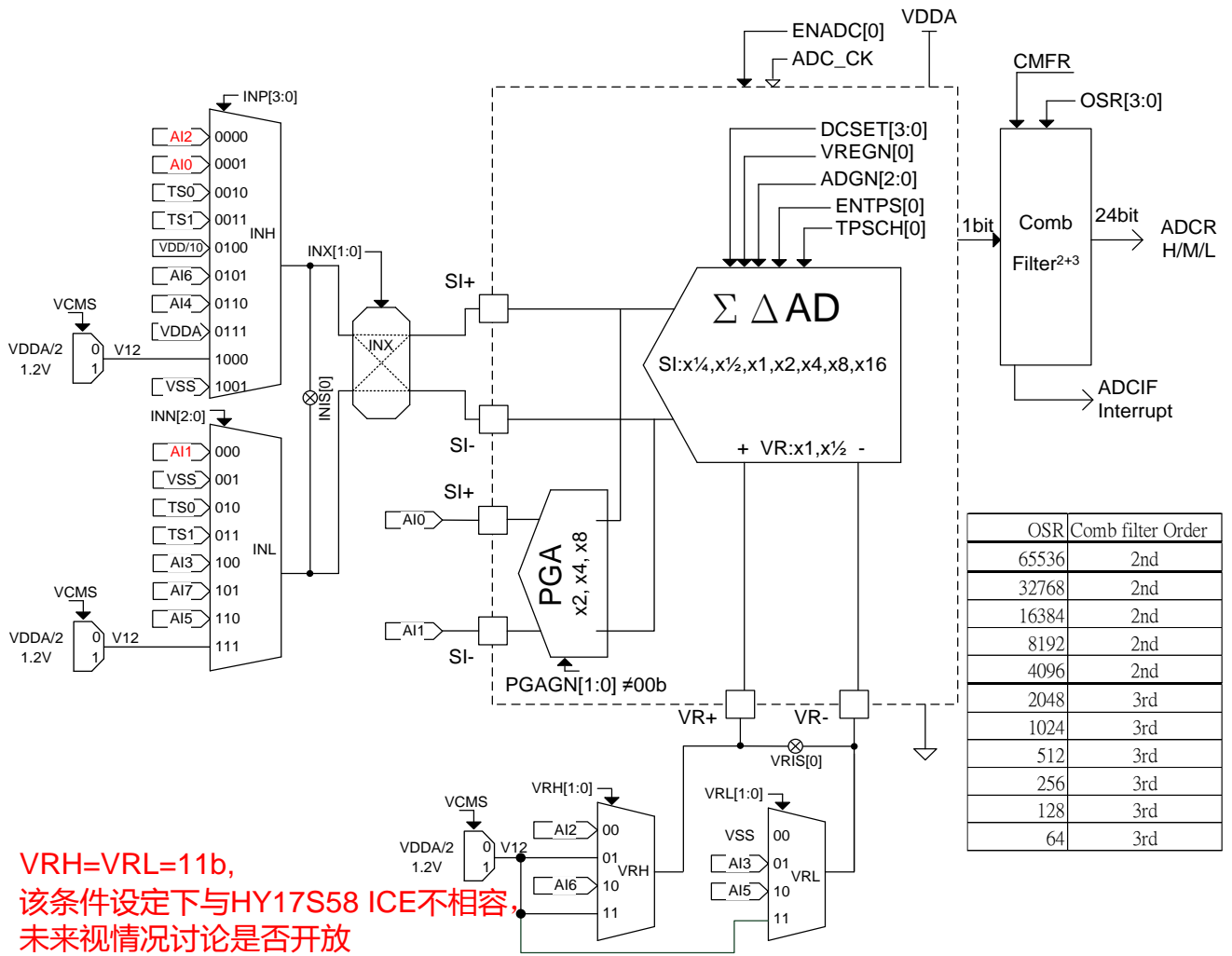


图 15-1 $\Sigma\Delta$ ADC 方块图(For HY17P48)

HY17S58 Emulate Chip User' Guide

Embedded $\Sigma\Delta$ ADC
8-Bit RISC-like Mixed Signal Microcontroller

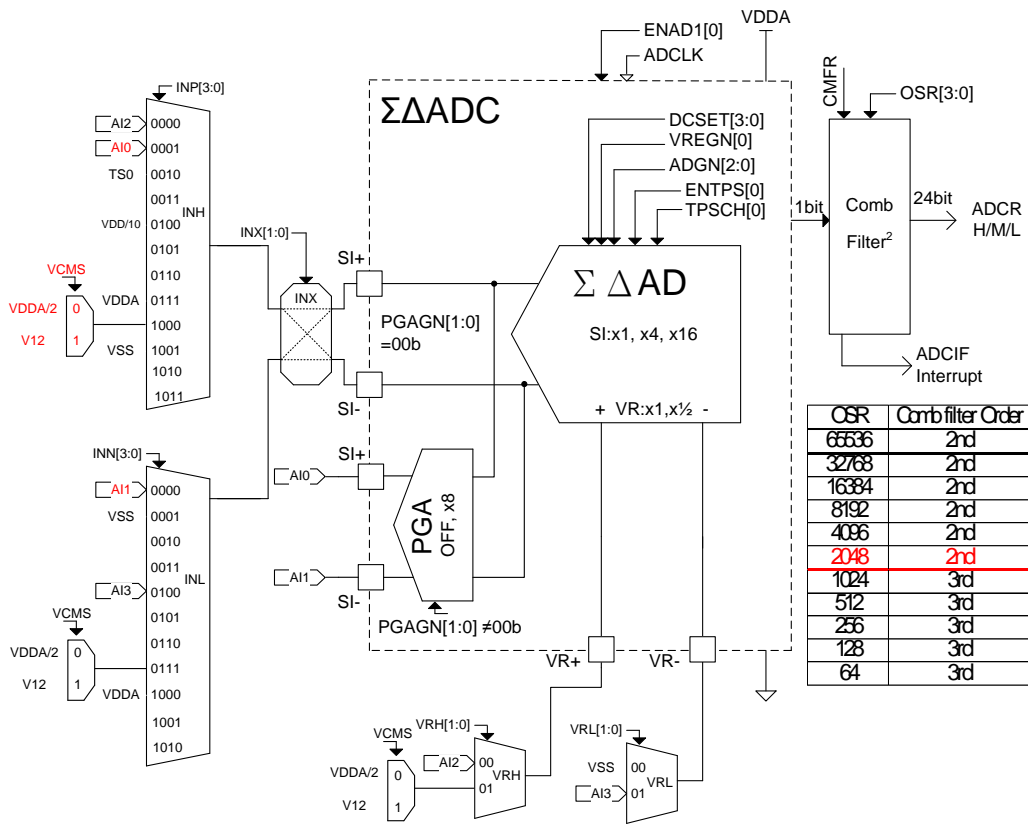


图 15-2 $\Sigma\Delta$ ADC 方块图(For HY17P51)

HY17S58 Emulate Chip User' Guide

Embedded $\Sigma\Delta$ ADC
8-Bit RISC-like Mixed Signal Microcontroller

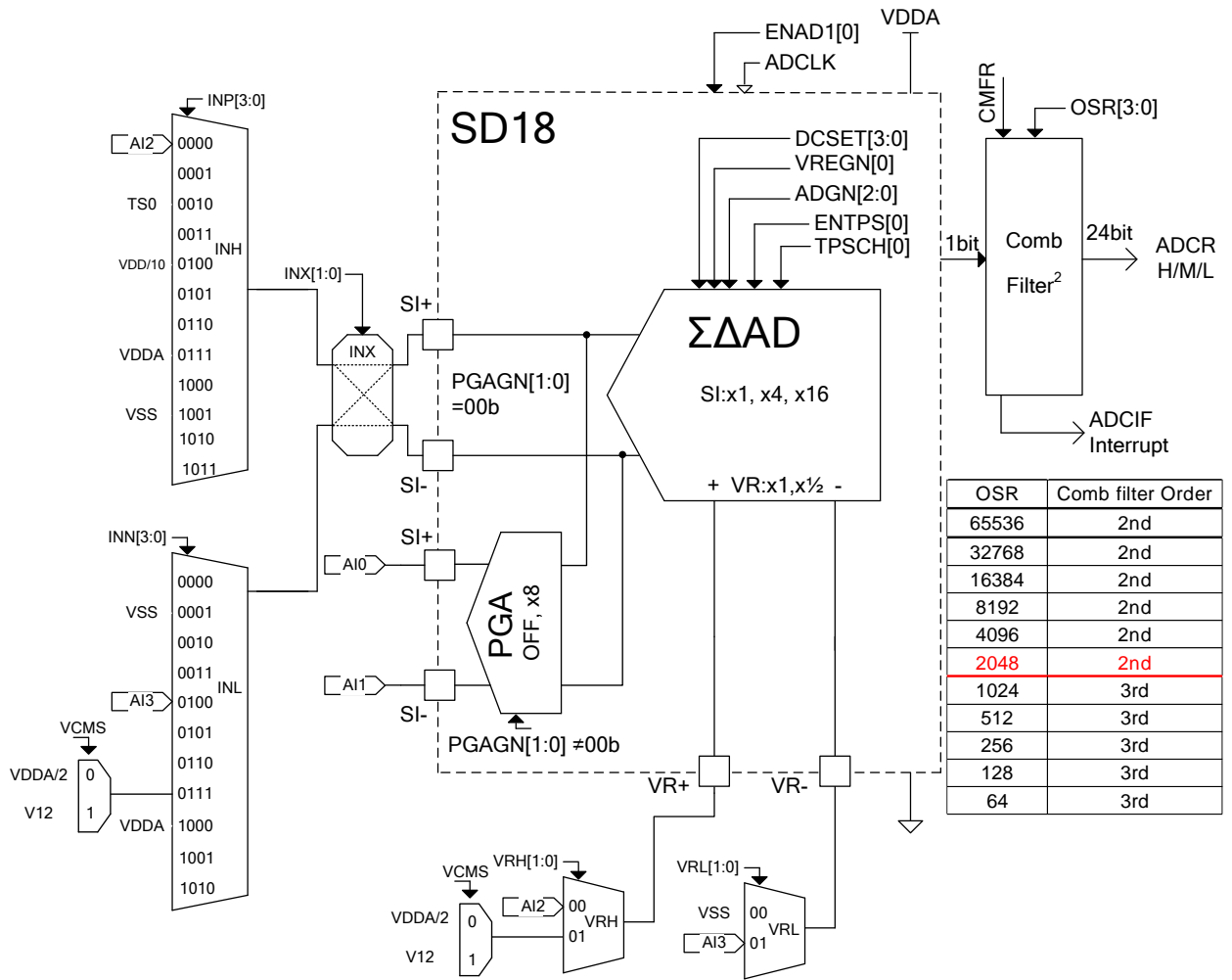


图 15-3 $\Sigma\Delta$ ADC 方块图(For HY17P52)

HY17S58 Emulate Chip User' Guide

Embedded $\Sigma\Delta$ ADC
8-Bit RISC-like Mixed Signal Microcontroller

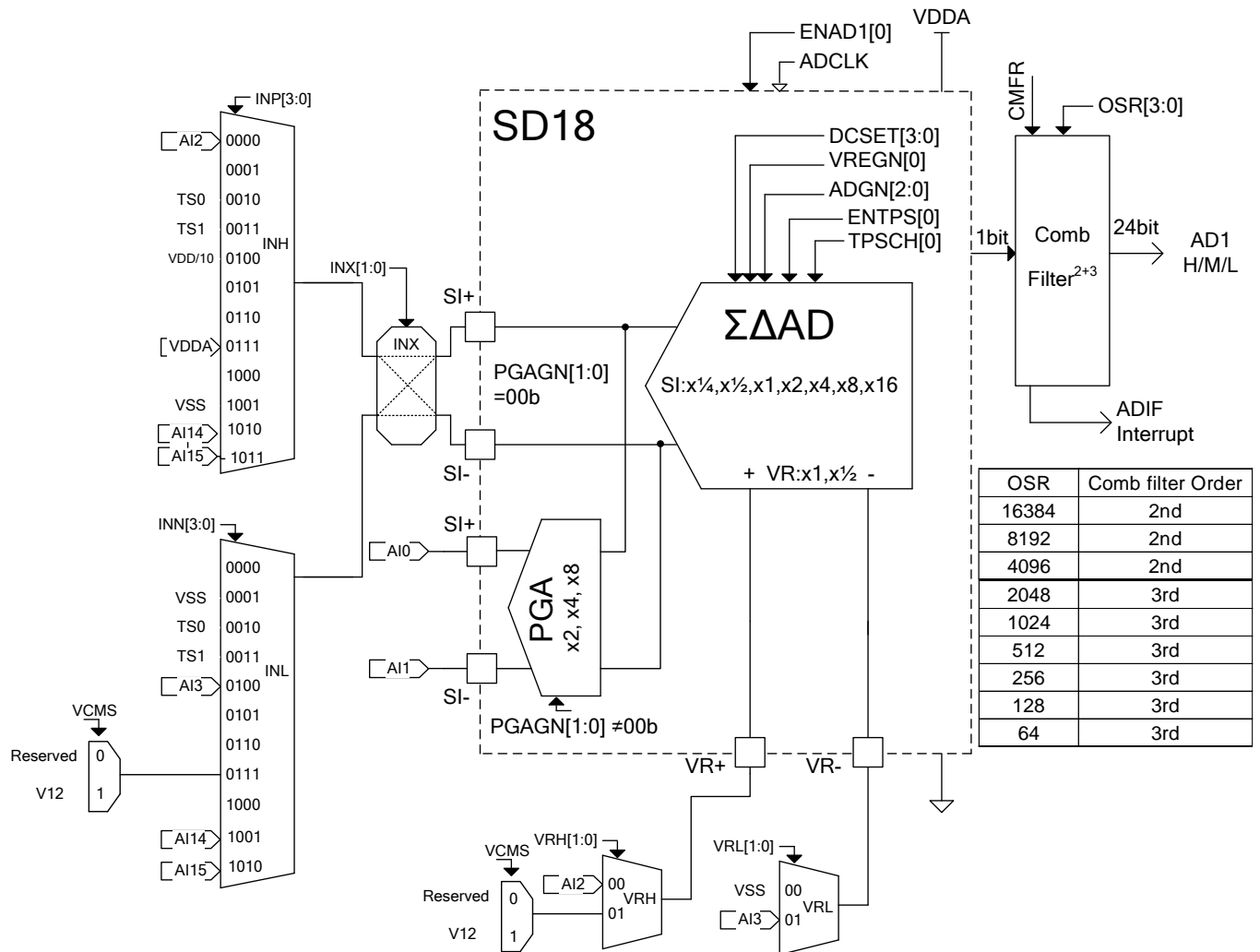


图 15-4 $\Sigma\Delta$ ADC 方块图(For HY17P55/HY17P56)

HY17S58 Emulate Chip User' Guide

Embedded $\Sigma\Delta$ ADC
8-Bit RISC-like Mixed Signal Microcontroller

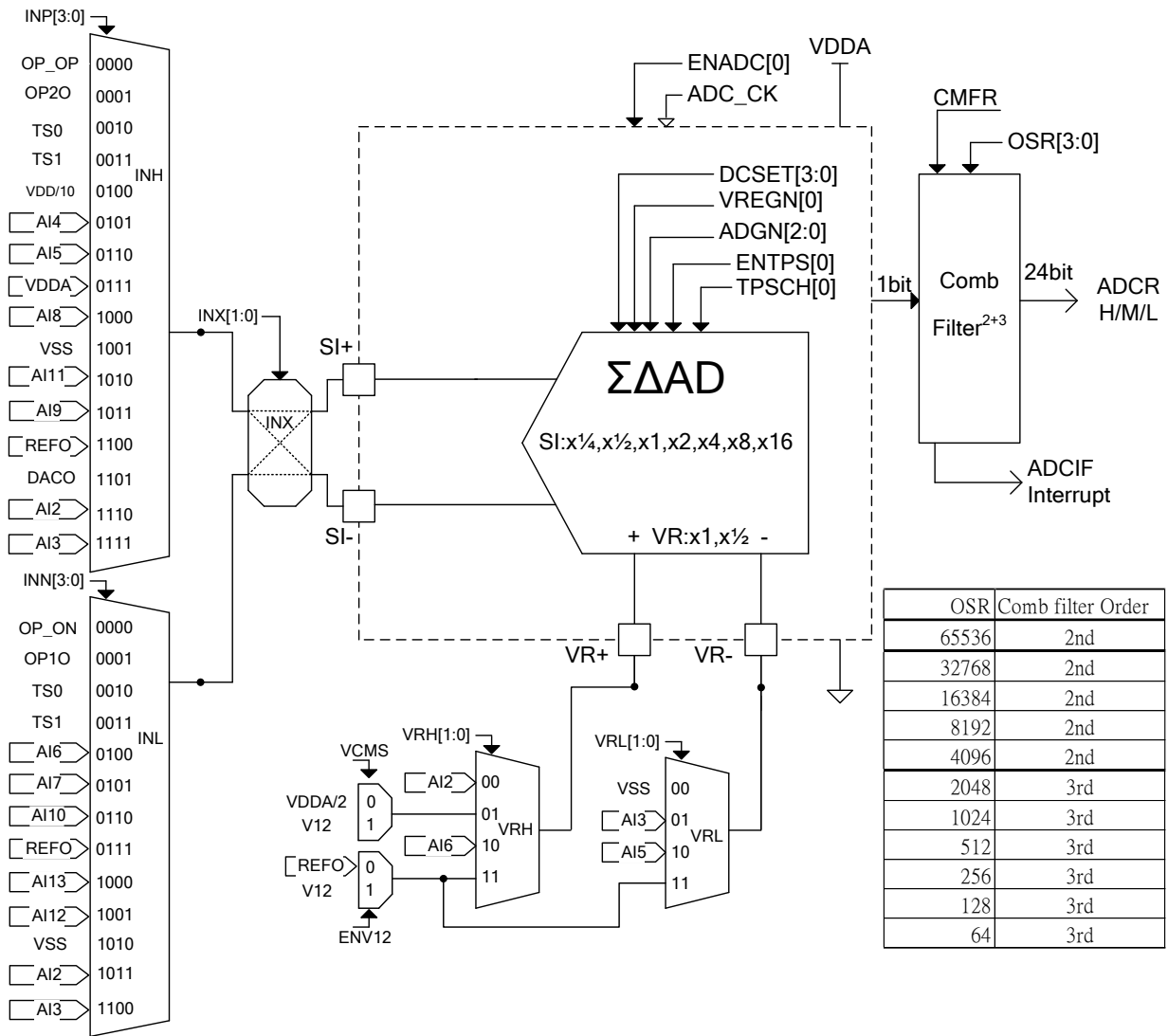


图 15-5 $\Sigma\Delta$ ADC 方块图(For HY17P58)

15.1. $\Sigma\Delta$ ADC 使用说明

15.1.1. $\Sigma\Delta$ ADC 初始化设置

15.1.1.1. 工作频率配置方式

$\Sigma\Delta$ ADC 的采样频率可经由采样频率选择器 ADCCK[0]设置 $\Sigma\Delta$ ADC 的工作频率由 DHS_CK 提供，其最高采样频率不可大于 1MHz(HY17P56/55 不可大于 500KHz)。较快的采样频率可在相同的输出速度下得到较好的分辨率，但其输入阻抗也会降低(参考: 錯誤! 找不到參照來源。錯誤! 找不到參照來源。)。当 DHS_CK 频率超过最大允许值时则必须透过采样频率预分频器 DADC[1:0]进行频率调整。

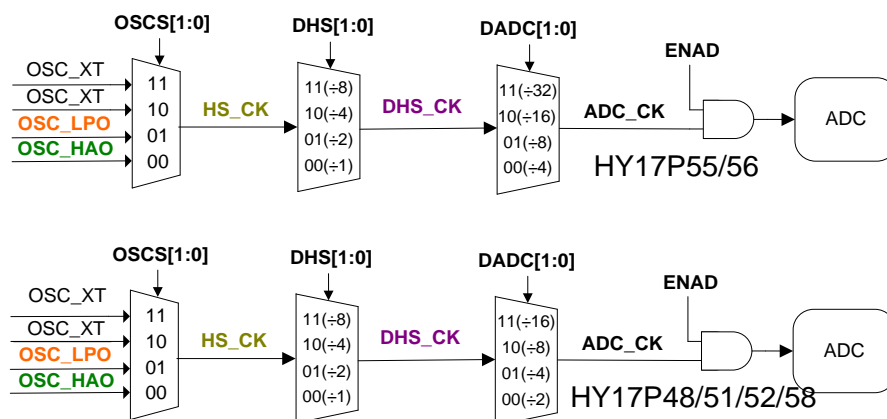


图 15-6 $\Sigma\Delta$ ADC 工作频率方块图

15.1.1.2. 多功能的输入多任务器配置方式

$\Sigma\Delta$ ADC 采用二阶的 $\Sigma\Delta$ 调变器，其待测信号及参考电压都可经由已下设置进行倍率及偏压调整。

- ΔVR_{\pm} 倍率调整器 VREGN[0]设置<1>时，会将参考电压的信号进行 1/2 倍率的调整也会因改变输入信号的 $\Delta SI_{\pm} = (SI+ - SI-)$ 与 $\Delta VR_{\pm} = (VR+ - VR-)$ 的比值；设置<0>则进行 1 倍调整。
- 输入信号经倍率调整器 ADGN[2:0]的设置，最大可达 16 倍的信号放大倍率，如表 15-1(a)。
- 输入信号 SI_{\pm} 透过直流输入偏压调整器 DCSET[3:0]，可调整输入信号零点位置以增加量测范围。偏压方式采加权参考信号 VR_{\pm} 的倍率值，如 表 15-1(b)。
- 信号测量时，需注意外部输入信号阻抗与 ADC 匹配问题。详细说明请参见 錯誤! 找不到參照來源。錯誤! 找不到參照來源。

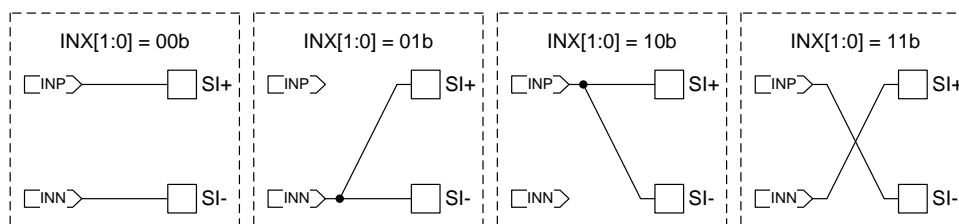


图 15-7 INX 输入信号转置器四种组合方式

设置	ADGN[2:0]							
输入	000	001	010	011	100	101	110	111

AD Gain	x1/4	x1/2	x1	x2	x4	x8	x16	RSVD
---------	------	------	----	----	----	----	-----	------

表 15-1 (a)ADGN[2:0]放大倍率配置表

设置	DCSET[3:0]							
输入	0000	0001	0010	0011	0100	0101	0110	0111
SI±	+0	+1/8 * Vref	+2/8 * Vref	+3/8 * Vref	+4/8 * Vref	+5/8 * Vref	+6/8 * Vref	+7/8 * Vref
设置	DCSET[3:0]							
输入	1000	1001	1010	1011	1100	1101	1110	1111
SI±	-0	-1/8 * Vref	-2/8 * Vref	-3/8 * Vref	-4/8 * Vref	-5/8 * Vref	-6/8 * Vref	-7/8 * Vref

单位：VR±

表 15-1 (b) SI±输入信号加权参考电压倍率一览表

$\Sigma\Delta$ 调变器经前置 PGA 及调变器本身的倍率偏压调整后，其等效的待测信号 ΔSI_I 与等效的参考电压 ΔVR_I 的计算公式分别如下：

式 15 -1

$$\Delta SI_I = PGAGN \times ADGN \times \Delta SI_{\pm} + (DCSET \times \Delta VR_{\pm})$$

式 15 -2

$$\Delta VR_I = VREGN \times VR_{\pm}$$

必须注意，为了使 $\Sigma\Delta$ 调变器输出得到较高的分辨率及线性度，故等效的参考电压 ΔVR_I 建议落在 $\Delta VR_I=0.8V\sim 1.2V$ ，而等效的待测信号 ΔSI_I 则操作在 $\Delta SI_I=\pm 0.9 \times \Delta VR_I$ 之间。

15.1.1.3. 梳状滤波器 Comb Filter 设置方式

$\Sigma\Delta$ 调变器输出 1-bit 数据至二阶梳状滤波器 Comb Filter，再由 Comb Filter 转成 24-bit 的数值存放于 AD1[23:0]寄存器。AD1[23:0]数据的更新速率即为 $\Sigma\Delta$ ADC 的输出速率，计算方式为 $\Sigma\Delta$ ADC 采样频率与 $\Sigma\Delta$ ADC 输出速率频率比值， $\Sigma\Delta$ ADC 输出速率频率又称为 OSR (Over Sampling Ratio)。

所以 $\Sigma\Delta$ ADC 输出速率为 $ADC_CK \div OSR$ ，而 OSR 数值可透过 OSR[3:0]设置以产生不同的 $\Sigma\Delta$ ADC 输出转换频率，如表 15-1(c)。

设置	OSR[3:0]										
ADC_ CK	65536	32768	16384	8192	4096	2048	1024	512	256	128	64
1000k	15	30	61	122	244	488	976	1953	3906	7812	15624
500K	7	15	30	61	122	244	488	976	1953	3906	7812
250K	3	7	15	30	61	122	244	488	976	1953	3906

表 15-1 (c) $\Sigma\Delta$ ADC 超采样频率配置简表

AD1[23:0]分别由 AD1H[7:0]、AD1M[7:0]及 AD1L[7:0]组成，其用于存放 Comb Filter 输出的 24-bit 数据。Comb Filter 的数据格式组成成分如表 15-2 所示。

+FSR/-FSR : 正相与负相最大量测范围

	等效待测信号	AD1[23:0]	
		十六进制	二进制
两极性输出 二补码格式	ΔVR_I	7FFFFFFF	0111-1111 1111-1111 1111-1111
	$\Delta VR_I \times \frac{1}{2^{23}}$	000001	0000-0000 0000-0000 0000-0001
	0	000000	0000-0000 0000-0000 0000-0000
	$-\Delta VR_I \times \frac{1}{2^{23}}$	FFFFFFF	1111-1111 1111-1111 1111-1111
	$-\Delta VR_I$	800000	1000-0000 0000-0000 0000-0000

表 15-2 AD1[23:0]与输入信号关系表

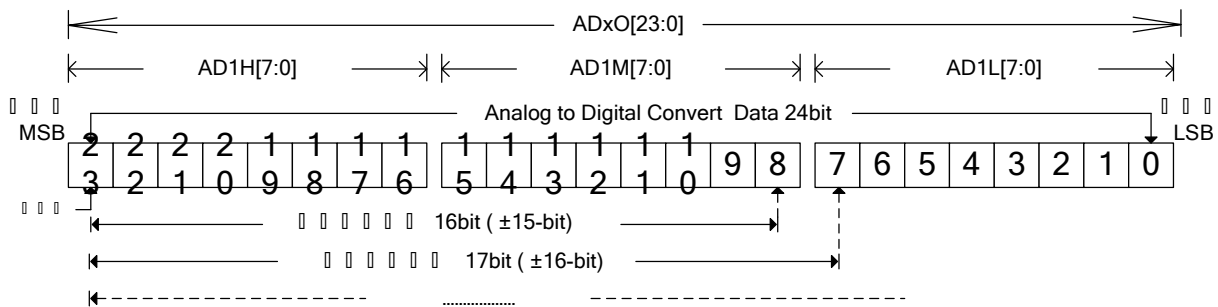


图 15-8 AD1[23:0]分辨率示意图

15.1.1.4. ADC 使用注意说明

- 当启动 ADC 时，必须设定 ADC Common Voltage，可以选择 VCMS=0b(ACM=VDDA/2)或是 VCMS=1b(ACM=1.2V)。
- 如果启动内部 VDDA 稳压时，除 ENLDO=1b，还需要启动 LDOPL=1b，才能正确稳压。
- LDOPL bit 与 VCMS bit 有连动关系。如果 LDOPL=1b，则 VCMS 可以选择 VCMS=0b(ACM=VDDA/2)，或是 VCMS=1b(ACM=1.2V)使用；如果 LDOPL=0b，则 VCMS 只可以选择 VCMS=1b(ACM=1.2V)使用。
- 如果 VDDA 为外灌电压模式时，则须设定 ENLDO=0b 关闭 LDO，切换 LDOM=00b=high impedance，才可由外部输入电压。如果设定了 LDOPL=0b，则可以关闭 pull down 电阻达省电效果。因此在设定 LDOPL=0b 之后，则需要设定 ADC Common Voltage，VCMS=1b(ACM=1.2V)。

15.2. 模拟通道输入特性

$\Sigma\Delta$ ADC 是使用切换式电容线路来进行模拟信号处理，当输入缓冲器不使用时为了保证取样电容的电压可以得到正确的值，输入信号的最大输出阻抗必须受到限制，而且会与 $\Sigma\Delta$ ADC 的采样频率及信号倍率选择有相互牵制的关系。

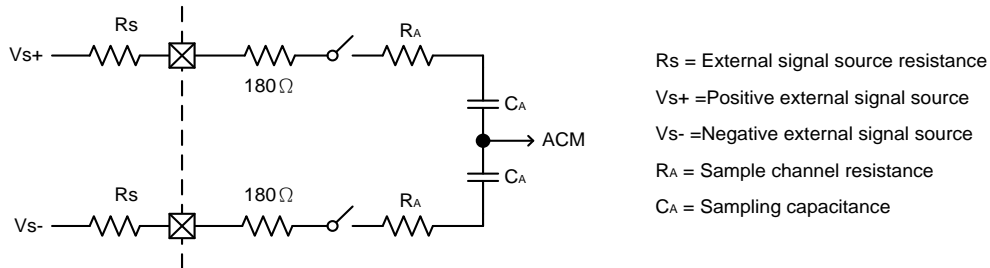


图 15-9 Alx 输入电容与阻抗模块

由图 15 可知，当输入信号不经缓冲器直接输入时必须进一步考虑输入信号内阻 R_s 与 $\Sigma\Delta$ ADC 的采样频率 ADC_CK 及寄生电阻 R_A 、电容 C_A 的效应。相关的计算公式如下：

式 15-3

$$t_s > (R_s + R_A + 180\Omega) \times C_A \times [\ln(2^{\text{ENOB}} \times \text{Gain}) + 2]$$

t_s : $\Sigma\Delta$ ADC 最短采样时间

ENOB : 期望得到 $\Sigma\Delta$ ADC 的有效位数

Gain : ($\Sigma\Delta$ AD Gain)

式 15 -4

$$F_s = \frac{1}{2 \times t_s}$$

F_s : $\Sigma\Delta$ ADC 最短采样频率

由于 $\Sigma\Delta$ ADC 组成包含 PGA 与 $\Sigma\Delta$ AD，此两部分在设计上存在各自的 R_A 与 C_A 值，而最短采样时间 t_s 的计算是依直接与输入信号匹配的部分来考虑。

$\Sigma\Delta$ AD Gain	C_A	R_A
x1/4		
x1/2		
x1	0.5pF	10k Ω
x2	1pF	10k Ω
x4	2pF	10k Ω
x8	4pF	5k Ω
X16		

表 15-3(a) $\Sigma\Delta$ ADC Gain 与 R_A 及 C_A 关系表

VR Gain	C _A	R _A
x1/2	0.25pF	10k Ω
X1	0.5pF	10 kohm

表 15-3(b) VR Gain 与 R_A 及 C_A 关系表

$\Sigma\Delta$ ADC 主要应用是要量测低频的信号，但在真实世界里待测信号会含有许多高频的噪声，根据信号采样原理超过采样频率的高频噪声经过采样后会产生零点飘移及低频噪声，进而造成量测的误差。因此我们建议在芯片差动待测信号及参考电压端加上 10nF~100nF 的滤波电容以加强量测的准确性。

15.2.1. TPS 初始化设置与计算方式

- TPS 的启用，除了 ENAD1 设置为 1b，还须将 ENTSPS 设置为 1b 才能启用。
- Gain=1、PGA=1、VR=VCMS-VSS、VCMS=V12、OSR=32768
- ADC 输入信号须测试两次。分别测得 ADC_{TPS0}、ADC_{TPS1} 两次的信号测是 ADC 信道配置参考下表

	HY17P52	HY17P48/55/56/58
ADC _{TPS0}	INP=0010 INN=0001 TPSCH=0b	INP=0010 INN=0010 TPSCH=0b
ADC _{TPS1}	INP=0010 INN=0001 TPSCH=1b	INP=0011 INN=0011 TPSCH=0b

- 在同一温度 T_A(温度 0 下， $\Sigma\Delta$ 度测量得到 ADC_{TPS0} 与 ADC_{TPS1} 的数值后，将两数相加并取平均值即可求得在温度 T_A 下测得 TPS 相对应的值 ADC_{TPS@T_A}。
- TPS 的输出值 V_{TPS} 对温度变化为一线性曲线，故可推导得出其增益值 G_{TPS}(或称斜率)。

式 15 -5 TPS 增益公式

$$G_{TPS} = \frac{ADC_{TPS@T_A}}{(273.15 + T_{offset} + T_A)K}$$

G_{TPS}:温度计算斜率 $\frac{ADC \text{ count}}{K}$

ADC_{TPS@T_A}:校正温度下所测得的 ADC 值

K=°C+273.15

T_{offset}=由于 TPS 在温度单位转换上的不理想，故会有一偏差值

- TPS 在温度转换上的不理想，因此实际上并非于 °C=K-273.15，而是 °C=K+KT=K+(-273.15-T_{offset}) 其中的 KT 值请参考该颗 IC Data sheet ADC 章节内 TPS 规格。

IC 型号	KT 值
HY17P48	-272
HY17P52	-279
HY17P55/56	-284
HY17P58	-284

15.2.2. TPS 范例说明

假设将于 25°C 进行 TPS 的校正。校正后将 IC 移动置一较高温环境(65°C)，测试该环境下的温度。

- (1) 设定 INP=0010b=TS0、INN=0010b=TS0、AD1CN5[TPSCH]=0b、AD1CN5[ENTPS]=1b ,ADC 量测得到一个数字码 ADCTPS0=5897634。
- (2) 设定 INP=0011b=TS1、INN=0011b=TS1、AD1CN5[TPSCH]=0b、AD1CN5[ENTPS]=1b ,ADC 量测得到一个数字码 ADCTPS1=5827679。
- (3) 计算 $ADCTPS@25=(ADCTPS0 +ADCTPS1)/2=5862656$ 。此动作可消除 Temperature Sensor 的 Offset。
- (4) 计算 GTPS :

$$G_{TPS} = \frac{ADC_{TPS@T_A}}{(273.15 + T_{offset} + T_A)K} = \frac{5862656}{(284 + 25)K} = 18973$$

- (5) 将 IC 移置高温环境(65°C)后一段时间后，参考步骤(1)~(3) 再次测得 $ADCTPS@65 : 6630103$

$$T_x = \frac{ADC_{TPS@65}}{G_{TPS}} - [273.15 + T_{offset}] = \frac{6630103}{18973} - 284 = 65.45^\circ C$$

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



15.3. 寄存器说明- Σ ADC

“.”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	-	ADIE	WDTE	TB1E	TMAE	E1E	E0E	0000 0000	0uuu uuuu	***** r r r r
INTF0	-	-	ADIF	WDTF	TB1F	TMAF	E1F	E0F	.000 0000	.uuu uuuu	***** r r r r
PWRCN	ENBGR	LDOC[2:0]			LDOM[0]	LDOM	ENLDO	CSFON	0000 0000	uuuu u00u	* * * * *,w r0,w r0,*
AD1H	ADC1 conversion high byte data register								..00 0000	..uu uuuu	-,-,***** r r r r
AD1M	ADC1 conversion middle byte data register								0000 0000	uuuu uuuu	***** r r r r
AD1L	ADC1 conversion low byte data register								0000 0000	uuuu uuuu	***** r r r r
AD1CN0	ENAD1	-	-	OSR[3:0]				CMFR	000. 0000	uuu. uuuu	***** r r r r
AD1CN1	-	-	VREGN	PGAGN[1:0]		ADGN[2:0]		xxxx xxxx	uuuu uuuu	***** r r r r	
AD1CN2	INIS1	-	-	-	DCSET[3:0]			xxxx xxxx	uuuu uuuu	***** r r r r	
AD1CN3	INP[1:0]			INN[1:0]				xxxx xxxx	uuuu uuuu	***** r r r r	
AD1CN4	-	VRH[0]	-	-	INX[1:0]		VRIS	INIS	0010 0000	uuuu uuuu	***** r r r r
AD1CN5	-	-	-	-	-	TPSCP	ENTPS	TPSCH	0000 0000	uuuu uuuu	***** r r r r

表 15-4 Σ ADC 寄存器

INTE0/INTF0: 详见 中断 章节

PWRCN: 详见电源系统章节

AD1[23:0]模拟数字转换寄存器

AD1H[7:0] AD1 模拟数字转换高字节数据寄存器

AD1M[7:0] AD1 模拟数字转换次高字节数据寄存器

AD1L[7:0] AD1 模拟数字转换低字节数据寄存器

AD1CN0: Σ ADC 控制寄存器 0

位	名称	描述																																																
Bit7	ENAD1	Σ ADC 启用控制器 <0> 关闭 <1> 启用																																																
Bit4~1	OSR<3:0>	Σ ADC 超采样率分频器(HY17P48/51/52/58)																																																
		<table border="1"> <thead> <tr> <th>OSR<3:0></th> <th>OSR</th> <th>Comb filter Order</th> <th>OSR<3:0></th> <th>OSR</th> <th>Comb filter Order</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>65536</td> <td>2nd</td> <td>1000</td> <td>256</td> <td>3rd</td> </tr> <tr> <td>0001</td> <td>32768</td> <td>2nd</td> <td>1001</td> <td>128</td> <td>3rd</td> </tr> <tr> <td>0010</td> <td>16384</td> <td>2nd</td> <td>1010</td> <td>64</td> <td>3rd</td> </tr> <tr> <td>0011</td> <td>8192</td> <td>2nd</td> <td>1011</td> <td>65536</td> <td>2nd</td> </tr> <tr> <td>0100</td> <td>4096</td> <td>2nd</td> <td>1100</td> <td>65536</td> <td>2nd</td> </tr> <tr> <td>0101</td> <td>2048</td> <td>2rd(HY17P51/52) 3rd(HY17P48/58)</td> <td>1101</td> <td>65536</td> <td>2nd</td> </tr> <tr> <td>0110</td> <td>1024</td> <td>3rd</td> <td>1110</td> <td>65536</td> <td>2nd</td> </tr> </tbody> </table>	OSR<3:0>	OSR	Comb filter Order	OSR<3:0>	OSR	Comb filter Order	0000	65536	2nd	1000	256	3rd	0001	32768	2nd	1001	128	3rd	0010	16384	2nd	1010	64	3rd	0011	8192	2nd	1011	65536	2nd	0100	4096	2nd	1100	65536	2nd	0101	2048	2rd(HY17P51/52) 3rd(HY17P48/58)	1101	65536	2nd	0110	1024	3rd	1110	65536	2nd
OSR<3:0>	OSR	Comb filter Order	OSR<3:0>	OSR	Comb filter Order																																													
0000	65536	2nd	1000	256	3rd																																													
0001	32768	2nd	1001	128	3rd																																													
0010	16384	2nd	1010	64	3rd																																													
0011	8192	2nd	1011	65536	2nd																																													
0100	4096	2nd	1100	65536	2nd																																													
0101	2048	2rd(HY17P51/52) 3rd(HY17P48/58)	1101	65536	2nd																																													
0110	1024	3rd	1110	65536	2nd																																													

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



位	名称	描述																																																												
		<table border="1"> <tr> <td>0111</td> <td>512</td> <td>3rd</td> <td>1111</td> <td>65536</td> <td>2nd</td> </tr> </table> <p>ΣADC 超采样率分频器(HY17P55/56)</p> <table border="1"> <thead> <tr> <th>OSR<3:0></th> <th>OSR</th> <th>Comb filter Order</th> <th>OSR<3:0></th> <th>OSR</th> <th>Comb filter Order</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>RSV</td> <td>RSV</td> <td>1000</td> <td>128</td> <td>3rd</td> </tr> <tr> <td>0001</td> <td>16384</td> <td>2nd</td> <td>1001</td> <td>64</td> <td>3rd</td> </tr> <tr> <td>0010</td> <td>8192</td> <td>2nd</td> <td>1010</td> <td>Rsd</td> <td>3rd</td> </tr> <tr> <td>0011</td> <td>4096</td> <td>2nd</td> <td>1011</td> <td>16384</td> <td>2nd</td> </tr> <tr> <td>0100</td> <td>2048</td> <td>2nd</td> <td>1100</td> <td>16384</td> <td>2nd</td> </tr> <tr> <td>0101</td> <td>1024</td> <td>3rd</td> <td>1101</td> <td>16384</td> <td>2nd</td> </tr> <tr> <td>0110</td> <td>512</td> <td>3rd</td> <td>1110</td> <td>16384</td> <td>2nd</td> </tr> <tr> <td>0111</td> <td>256</td> <td>3rd</td> <td>1111</td> <td>16384</td> <td>2nd</td> </tr> </tbody> </table>	0111	512	3rd	1111	65536	2nd	OSR<3:0>	OSR	Comb filter Order	OSR<3:0>	OSR	Comb filter Order	0000	RSV	RSV	1000	128	3rd	0001	16384	2nd	1001	64	3rd	0010	8192	2nd	1010	Rsd	3rd	0011	4096	2nd	1011	16384	2nd	0100	2048	2nd	1100	16384	2nd	0101	1024	3rd	1101	16384	2nd	0110	512	3rd	1110	16384	2nd	0111	256	3rd	1111	16384	2nd
0111	512	3rd	1111	65536	2nd																																																									
OSR<3:0>	OSR	Comb filter Order	OSR<3:0>	OSR	Comb filter Order																																																									
0000	RSV	RSV	1000	128	3rd																																																									
0001	16384	2nd	1001	64	3rd																																																									
0010	8192	2nd	1010	Rsd	3rd																																																									
0011	4096	2nd	1011	16384	2nd																																																									
0100	2048	2nd	1100	16384	2nd																																																									
0101	1024	3rd	1101	16384	2nd																																																									
0110	512	3rd	1110	16384	2nd																																																									
0111	256	3rd	1111	16384	2nd																																																									
Bit0	CMFR	<p>ΣDFR4 与梳状滤波器复位控制器</p> <p><0> 不复位</p> <p><1> 复位；写入动作即发生复位</p>																																																												

AD1CN1: Σ ADC 控制寄存器 1

位	名称	描述																												
Bit5	VREGN	<p>VR\pm倍率调整器</p> <p><0> x1</p> <p><1> x1/2</p>																												
Bit4~3	PGAGN	<p>PGA 倍率调整器</p> <table border="1"> <thead> <tr> <th colspan="2">HY17P58/56/55/48</th> <th colspan="2">HY17P51/52</th> </tr> <tr> <th>PGAGN</th> <th>Gain</th> <th>PGAGN</th> <th>Gain</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>x1</td> <td>00</td> <td>x1</td> </tr> <tr> <td>01</td> <td>x2</td> <td>01</td> <td>RSVD</td> </tr> <tr> <td>10</td> <td>x4</td> <td>10</td> <td>RSVD</td> </tr> <tr> <td>11</td> <td>x8</td> <td>11</td> <td>x8</td> </tr> </tbody> </table>	HY17P58/56/55/48		HY17P51/52		PGAGN	Gain	PGAGN	Gain	00	x1	00	x1	01	x2	01	RSVD	10	x4	10	RSVD	11	x8	11	x8				
HY17P58/56/55/48		HY17P51/52																												
PGAGN	Gain	PGAGN	Gain																											
00	x1	00	x1																											
01	x2	01	RSVD																											
10	x4	10	RSVD																											
11	x8	11	x8																											
Bit2~0	ADGN[2:0]	<p>AD 倍率调整器</p> <table border="1"> <thead> <tr> <th colspan="2">HY17P58/56/55/48</th> <th colspan="2">HY17P51/52</th> </tr> <tr> <th>ADGN[2:0]</th> <th>Gain</th> <th>ADGN[2:0]</th> <th>Gain</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>x1/4</td> <td>000</td> <td>x1</td> </tr> <tr> <td>001</td> <td>x1/2</td> <td>001</td> <td>x4</td> </tr> <tr> <td>010</td> <td>x1</td> <td>010</td> <td>x16</td> </tr> <tr> <td>011</td> <td>x2</td> <td>011</td> <td>x16</td> </tr> <tr> <td>100</td> <td>x4</td> <td>100</td> <td>x16</td> </tr> </tbody> </table>	HY17P58/56/55/48		HY17P51/52		ADGN[2:0]	Gain	ADGN[2:0]	Gain	000	x1/4	000	x1	001	x1/2	001	x4	010	x1	010	x16	011	x2	011	x16	100	x4	100	x16
HY17P58/56/55/48		HY17P51/52																												
ADGN[2:0]	Gain	ADGN[2:0]	Gain																											
000	x1/4	000	x1																											
001	x1/2	001	x4																											
010	x1	010	x16																											
011	x2	011	x16																											
100	x4	100	x16																											

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



位	名称	描述			
		101	x8	101	x16
		110	x16	110	x16
		111	RSVD	111	RSVD

AD1CN2: Σ ADC 控制寄存器 2

位	名称	描述																																				
Bit7	INIS1	C-Type PGA SI \pm 输入信号短路控制器 <0> 未短路 <1> 短路(测试用, 不建议设定)																																				
Bit3~0	DCSET[3:0]	SI \pm 偏压调整器 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>DCSET<3:0></th> <th>Offset</th> <th>DCSET<3:0></th> <th>Offset</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>+0*(REFP – REFN)</td> <td>1000</td> <td>-0*(REFP – REFN)</td> </tr> <tr> <td>0001</td> <td>+1/8*(REFP – REFN)</td> <td>1001</td> <td>-1/8*(REFP – REFN)</td> </tr> <tr> <td>0010</td> <td>+2/8*(REFP – REFN)</td> <td>1010</td> <td>-2/8*(REFP – REFN)</td> </tr> <tr> <td>0011</td> <td>+3/8*(REFP – REFN)</td> <td>1011</td> <td>-3/8*(REFP – REFN)</td> </tr> <tr> <td>0100</td> <td>+4/8*(REFP – REFN)</td> <td>1100</td> <td>-4/8*(REFP – REFN)</td> </tr> <tr> <td>0101</td> <td>+5/8*(REFP – REFN)</td> <td>1101</td> <td>-5/8*(REFP – REFN)</td> </tr> <tr> <td>0110</td> <td>+6/8*(REFP – REFN)</td> <td>1110</td> <td>-6/8*(REFP – REFN)</td> </tr> <tr> <td>0111</td> <td>+7/8*(REFP – REFN)</td> <td>1111</td> <td>-7/8*(REFP – REFN)</td> </tr> </tbody> </table>	DCSET<3:0>	Offset	DCSET<3:0>	Offset	0000	+0*(REFP – REFN)	1000	-0*(REFP – REFN)	0001	+1/8*(REFP – REFN)	1001	-1/8*(REFP – REFN)	0010	+2/8*(REFP – REFN)	1010	-2/8*(REFP – REFN)	0011	+3/8*(REFP – REFN)	1011	-3/8*(REFP – REFN)	0100	+4/8*(REFP – REFN)	1100	-4/8*(REFP – REFN)	0101	+5/8*(REFP – REFN)	1101	-5/8*(REFP – REFN)	0110	+6/8*(REFP – REFN)	1110	-6/8*(REFP – REFN)	0111	+7/8*(REFP – REFN)	1111	-7/8*(REFP – REFN)
DCSET<3:0>	Offset	DCSET<3:0>	Offset																																			
0000	+0*(REFP – REFN)	1000	-0*(REFP – REFN)																																			
0001	+1/8*(REFP – REFN)	1001	-1/8*(REFP – REFN)																																			
0010	+2/8*(REFP – REFN)	1010	-2/8*(REFP – REFN)																																			
0011	+3/8*(REFP – REFN)	1011	-3/8*(REFP – REFN)																																			
0100	+4/8*(REFP – REFN)	1100	-4/8*(REFP – REFN)																																			
0101	+5/8*(REFP – REFN)	1101	-5/8*(REFP – REFN)																																			
0110	+6/8*(REFP – REFN)	1110	-6/8*(REFP – REFN)																																			
0111	+7/8*(REFP – REFN)	1111	-7/8*(REFP – REFN)																																			

HY17S58 Emulate Chip User' Guide

Embedded ΣADC
8-Bit RISC-like Mixed Signal Microcontroller



AD1CN3: ΣADC 控制寄存器 3

位	名称	描述																																				
Bit7~4	INP[3:0]	SI±“+”输入信号选择器																																				
		HY17P48																																				
		<table border="1"> <thead> <tr> <th>INP<3:0></th> <th>ADC 输入通道</th> <th>INP<3:0></th> <th>ADC 输入通道</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>AI2</td> <td>1000</td> <td>VCMS</td> </tr> <tr> <td>0001</td> <td>AI0</td> <td>1001</td> <td>VSS</td> </tr> <tr> <td>0010</td> <td>TS0</td> <td>1010</td> <td>-</td> </tr> <tr> <td>0011</td> <td>TS1</td> <td>1011</td> <td>-</td> </tr> <tr> <td>0100</td> <td>VDD/10 注 1</td> <td>1100</td> <td>-</td> </tr> <tr> <td>0101</td> <td>AI6</td> <td>1101</td> <td>-</td> </tr> <tr> <td>0110</td> <td>AI4</td> <td>1110</td> <td>-</td> </tr> <tr> <td>0111</td> <td>VDDA</td> <td>1111</td> <td>-</td> </tr> </tbody> </table>	INP<3:0>	ADC 输入通道	INP<3:0>	ADC 输入通道	0000	AI2	1000	VCMS	0001	AI0	1001	VSS	0010	TS0	1010	-	0011	TS1	1011	-	0100	VDD/10 注 1	1100	-	0101	AI6	1101	-	0110	AI4	1110	-	0111	VDDA	1111	-
		INP<3:0>	ADC 输入通道	INP<3:0>	ADC 输入通道																																	
		0000	AI2	1000	VCMS																																	
		0001	AI0	1001	VSS																																	
		0010	TS0	1010	-																																	
		0011	TS1	1011	-																																	
		0100	VDD/10 注 1	1100	-																																	
		0101	AI6	1101	-																																	
		0110	AI4	1110	-																																	
		0111	VDDA	1111	-																																	
		注 1:HY17P48 使用 VDD/10 通道时, INN 固定要选为 AI3。如需要对 VSS 测量。须从 IC 外部把 AI3 短路到 VSS ; 或者开启高精度模式配合, 此用法建议与代理商工程窗口确认																																				
		HY17P51/52																																				
		<table border="1"> <thead> <tr> <th>INP<3:0></th> <th>ADC 输入通道</th> <th>INP<3:0></th> <th>ADC 输入通道</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>AI2</td> <td>1000</td> <td>由 VCMS[0]控制位决定(HY17P51 Only)</td> </tr> <tr> <td>0001</td> <td>AI0(HY17P51 Only)</td> <td>1001</td> <td>VSS</td> </tr> <tr> <td>0010</td> <td>TS0</td> <td>1010</td> <td>-</td> </tr> <tr> <td>0011</td> <td>-</td> <td>1011</td> <td>-</td> </tr> <tr> <td>0100</td> <td>VDD/10</td> <td>1100</td> <td>-</td> </tr> <tr> <td>0101</td> <td>-</td> <td>1101</td> <td>-</td> </tr> <tr> <td>0110</td> <td>-</td> <td>1110</td> <td>-</td> </tr> <tr> <td>0111</td> <td>VDDA</td> <td>1111</td> <td>-</td> </tr> </tbody> </table>	INP<3:0>	ADC 输入通道	INP<3:0>	ADC 输入通道	0000	AI2	1000	由 VCMS[0]控制位决定(HY17P51 Only)	0001	AI0(HY17P51 Only)	1001	VSS	0010	TS0	1010	-	0011	-	1011	-	0100	VDD/10	1100	-	0101	-	1101	-	0110	-	1110	-	0111	VDDA	1111	-
INP<3:0>	ADC 输入通道	INP<3:0>	ADC 输入通道																																			
0000	AI2	1000	由 VCMS[0]控制位决定(HY17P51 Only)																																			
0001	AI0(HY17P51 Only)	1001	VSS																																			
0010	TS0	1010	-																																			
0011	-	1011	-																																			
0100	VDD/10	1100	-																																			
0101	-	1101	-																																			
0110	-	1110	-																																			
0111	VDDA	1111	-																																			
HY17P55/HY17P56																																						
<table border="1"> <thead> <tr> <th>INP<3:0></th> <th>ADC 输入通道</th> <th>INP<3:0></th> <th>ADC 输入通道</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>AI2</td> <td>1000</td> <td>-</td> </tr> <tr> <td>0001</td> <td>-</td> <td>1001</td> <td>VSS</td> </tr> <tr> <td>0010</td> <td>TS0</td> <td>1010</td> <td>AI14</td> </tr> <tr> <td>0011</td> <td>TS1</td> <td>1011</td> <td>AI15</td> </tr> <tr> <td>0100</td> <td>VDD/10</td> <td>1100</td> <td>-</td> </tr> <tr> <td>0101</td> <td>-</td> <td>1101</td> <td>-</td> </tr> <tr> <td>0110</td> <td>-</td> <td>1110</td> <td>-</td> </tr> <tr> <td>0111</td> <td>VDDA</td> <td>1111</td> <td>-</td> </tr> </tbody> </table>	INP<3:0>	ADC 输入通道	INP<3:0>	ADC 输入通道	0000	AI2	1000	-	0001	-	1001	VSS	0010	TS0	1010	AI14	0011	TS1	1011	AI15	0100	VDD/10	1100	-	0101	-	1101	-	0110	-	1110	-	0111	VDDA	1111	-		
INP<3:0>	ADC 输入通道	INP<3:0>	ADC 输入通道																																			
0000	AI2	1000	-																																			
0001	-	1001	VSS																																			
0010	TS0	1010	AI14																																			
0011	TS1	1011	AI15																																			
0100	VDD/10	1100	-																																			
0101	-	1101	-																																			
0110	-	1110	-																																			
0111	VDDA	1111	-																																			
位	名称	描述																																				

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



		<p>HY17P58</p> <table border="1"> <thead> <tr> <th>INP<3:0></th> <th>ADC 输入通道</th> <th>INP<3:0></th> <th>ADC 输入通道</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>OP_OP</td> <td>1000</td> <td>AI8</td> </tr> <tr> <td>0001</td> <td>OP20</td> <td>1001</td> <td>VSS</td> </tr> <tr> <td>0010</td> <td>TS1</td> <td>1010</td> <td>AI11</td> </tr> <tr> <td>0011</td> <td>TS0</td> <td>1011</td> <td>AI9</td> </tr> <tr> <td>0100</td> <td>VDD/10</td> <td>1100</td> <td>REFO</td> </tr> <tr> <td>0101</td> <td>AI4</td> <td>1101</td> <td>DACO</td> </tr> <tr> <td>0110</td> <td>AI5</td> <td>1110</td> <td>AI2</td> </tr> <tr> <td>0111</td> <td>VDDA</td> <td>1111</td> <td>AI3</td> </tr> </tbody> </table>	INP<3:0>	ADC 输入通道	INP<3:0>	ADC 输入通道	0000	OP_OP	1000	AI8	0001	OP20	1001	VSS	0010	TS1	1010	AI11	0011	TS0	1011	AI9	0100	VDD/10	1100	REFO	0101	AI4	1101	DACO	0110	AI5	1110	AI2	0111	VDDA	1111	AI3																																				
INP<3:0>	ADC 输入通道	INP<3:0>	ADC 输入通道																																																																							
0000	OP_OP	1000	AI8																																																																							
0001	OP20	1001	VSS																																																																							
0010	TS1	1010	AI11																																																																							
0011	TS0	1011	AI9																																																																							
0100	VDD/10	1100	REFO																																																																							
0101	AI4	1101	DACO																																																																							
0110	AI5	1110	AI2																																																																							
0111	VDDA	1111	AI3																																																																							
Bit3~0	INN[3:0]	<p>SI±“-”输入信号选择器</p> <p>HY17P48</p> <table border="1"> <thead> <tr> <th>INN<3:0></th> <th>ADC 输入通道</th> <th>INN<3:0></th> <th>ADC 输入通道</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>AI1</td> <td>1000</td> <td>-</td> </tr> <tr> <td>0001</td> <td>VSS</td> <td>1001</td> <td>-</td> </tr> <tr> <td>0010</td> <td>TS0</td> <td>1010</td> <td>-</td> </tr> <tr> <td>0011</td> <td>TS1</td> <td>1011</td> <td>-</td> </tr> <tr> <td>0100</td> <td>AI3 注 2</td> <td>1100</td> <td>-</td> </tr> <tr> <td>0101</td> <td>AI7</td> <td>1101</td> <td>-</td> </tr> <tr> <td>0110</td> <td>AI5</td> <td>1110</td> <td>-</td> </tr> <tr> <td>0111</td> <td>VCMS[0]=0 时, 为 VDDA/2 VCMS[0]=1 时, 为 V12</td> <td>1111</td> <td>-</td> </tr> </tbody> </table> <p>注 2 进低功耗模式前, 须把 INN 选离 AI3 通道</p> <p>HY17P51/52</p> <table border="1"> <thead> <tr> <th>INN<3:0></th> <th>ADC 输入通道</th> <th>INN<3:0></th> <th>ADC 输入通道</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>AI1(HY17P51 Only)</td> <td>1000</td> <td>VDDA</td> </tr> <tr> <td>0001</td> <td>VSS</td> <td>1001</td> <td>-</td> </tr> <tr> <td>0010</td> <td>-</td> <td>1010</td> <td>-</td> </tr> <tr> <td>0011</td> <td>-</td> <td>1011</td> <td>-</td> </tr> <tr> <td>0100</td> <td>AI3</td> <td>1100</td> <td>-</td> </tr> <tr> <td>0101</td> <td>-</td> <td>1101</td> <td>-</td> </tr> <tr> <td>0110</td> <td>-</td> <td>1110</td> <td>-</td> </tr> <tr> <td>0111</td> <td>VCMS[0]=0 时, 为 VDDA/2 VCMS[0]=1 时, 为 V12</td> <td>1111</td> <td>-</td> </tr> </tbody> </table>	INN<3:0>	ADC 输入通道	INN<3:0>	ADC 输入通道	0000	AI1	1000	-	0001	VSS	1001	-	0010	TS0	1010	-	0011	TS1	1011	-	0100	AI3 注 2	1100	-	0101	AI7	1101	-	0110	AI5	1110	-	0111	VCMS[0]=0 时, 为 VDDA/2 VCMS[0]=1 时, 为 V12	1111	-	INN<3:0>	ADC 输入通道	INN<3:0>	ADC 输入通道	0000	AI1(HY17P51 Only)	1000	VDDA	0001	VSS	1001	-	0010	-	1010	-	0011	-	1011	-	0100	AI3	1100	-	0101	-	1101	-	0110	-	1110	-	0111	VCMS[0]=0 时, 为 VDDA/2 VCMS[0]=1 时, 为 V12	1111	-
INN<3:0>	ADC 输入通道	INN<3:0>	ADC 输入通道																																																																							
0000	AI1	1000	-																																																																							
0001	VSS	1001	-																																																																							
0010	TS0	1010	-																																																																							
0011	TS1	1011	-																																																																							
0100	AI3 注 2	1100	-																																																																							
0101	AI7	1101	-																																																																							
0110	AI5	1110	-																																																																							
0111	VCMS[0]=0 时, 为 VDDA/2 VCMS[0]=1 时, 为 V12	1111	-																																																																							
INN<3:0>	ADC 输入通道	INN<3:0>	ADC 输入通道																																																																							
0000	AI1(HY17P51 Only)	1000	VDDA																																																																							
0001	VSS	1001	-																																																																							
0010	-	1010	-																																																																							
0011	-	1011	-																																																																							
0100	AI3	1100	-																																																																							
0101	-	1101	-																																																																							
0110	-	1110	-																																																																							
0111	VCMS[0]=0 时, 为 VDDA/2 VCMS[0]=1 时, 为 V12	1111	-																																																																							

位	名称	描述
---	----	----

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述			
		HY17P55/HY17P56			
		INN<3:0>	ADC 输入通道	INN<3:0>	ADC 输入通道
		0000	-	1000	-
		0001	VSS	1001	AI14
		0010	TS0	1010	AI15
		0011	TS1	1011	-
		0100	AI3	1100	-
		0101	-	1101	-
		0110	-	1110	-
		0111	VCMS[0]=0 时, 为 VDDA/2 VCMS[0]=1 时, 为 V12	1111	-
		HY17P58			
		INN<3:0>	ADC 输入通道	INN<3:0>	ADC 输入通道
		0000	OP_ON	1000	AI13
		0001	OP10	1001	AI12
		0010	TS0	1010	VSS
		0011	TS1	1011	AI2
		0100	AI6	1100	AI3
		0101	AI7	1101	-
		0110	AI10	1110	-
		0111	REFO	1111	-

AD1CN4: Σ ADC 控制寄存器 4

位	名称	描述	
Bit7~6	VRH[1:0]	ADC 参考电压 VR \pm “+”电压信号选择器	
		HY17P58	
		VRH[1:0]	ADC 参考电压+
		00	AI2
		01	VCMS[0]=0 时, 为 VDDA/2(保留, 不开放) VCMS[0]=1 时, 为 V12
		10	AI6
		11	ENV12[0]=0 时, 为 REFO ENV12[0]=1 时, 为 V12
		HY17P51/52/55/56	
		VRH[1:0]	ADC 参考电压+
		00	AI2
		01	VCMS[0]=0 时, 为 VDDA/2(保留, 不开放) VCMS[0]=1 时, 为 V12
		10	-
		11	-
		HY17P48	
		VRH[1:0]	ADC 参考电压+
		00	AI2
		01	VCMS[0]=0 时, 为 VDDA/2(保留, 不开放) VCMS[0]=1 时, 为 V12
10	AI6		
11	ENV12[0]=0 时, 为 VDDA/2 ENV12[0]=1 时, 为 V12		

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



位	名称	描述																														
Bit5~4	VRL[1:0]	<p>ADC 参考电压 VR± “-”电压信号选择器</p> <p>HY17P58</p> <table border="1"> <thead> <tr> <th>VRL[1:0]</th> <th>ADC 参考电压-</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>VSS</td> </tr> <tr> <td>01</td> <td>AI3</td> </tr> <tr> <td>10</td> <td>AI5</td> </tr> <tr> <td>11</td> <td>ENV12[0]=0 时, 为 REFO ENV12[0]=1 时, 为 V12</td> </tr> </tbody> </table> <p>HY17P51/52/55/56</p> <table border="1"> <thead> <tr> <th>VRL[1:0]</th> <th>ADC 参考电压-</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>VSS</td> </tr> <tr> <td>01</td> <td>AI3</td> </tr> <tr> <td>10</td> <td>-</td> </tr> <tr> <td>11</td> <td>-</td> </tr> </tbody> </table> <p>HY17P48</p> <table border="1"> <thead> <tr> <th>VRL[1:0]</th> <th>ADC 参考电压-</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>VSS</td> </tr> <tr> <td>01</td> <td>AI3</td> </tr> <tr> <td>10</td> <td>AI5</td> </tr> <tr> <td>11</td> <td>ENV12[0]=0 时, 为 VDDA/2 ENV12[0]=1 时, 为 V12</td> </tr> </tbody> </table>	VRL[1:0]	ADC 参考电压-	00	VSS	01	AI3	10	AI5	11	ENV12[0]=0 时, 为 REFO ENV12[0]=1 时, 为 V12	VRL[1:0]	ADC 参考电压-	00	VSS	01	AI3	10	-	11	-	VRL[1:0]	ADC 参考电压-	00	VSS	01	AI3	10	AI5	11	ENV12[0]=0 时, 为 VDDA/2 ENV12[0]=1 时, 为 V12
VRL[1:0]	ADC 参考电压-																															
00	VSS																															
01	AI3																															
10	AI5																															
11	ENV12[0]=0 时, 为 REFO ENV12[0]=1 时, 为 V12																															
VRL[1:0]	ADC 参考电压-																															
00	VSS																															
01	AI3																															
10	-																															
11	-																															
VRL[1:0]	ADC 参考电压-																															
00	VSS																															
01	AI3																															
10	AI5																															
11	ENV12[0]=0 时, 为 VDDA/2 ENV12[0]=1 时, 为 V12																															
Bit3~2	INX	<p>SI±输入信号转置器</p> <p><11> INP→ADL, INN→ADH</p> <p><10> INN 浮接, INP→ADH & ADH</p> <p><01> INN→ADH & ADL, INP 浮接</p> <p><00> INP→ADH, INN→ADL</p>																														
Bit1	VRIS	<p>VR±输入信号短路控制器</p> <p><0> 未短路</p> <p><1> 短路(测试用, 不建议设定)</p>																														
Bit0	INIS	<p>SI±输入信号短路控制器</p> <p><0> 未短路</p> <p><1> 短路(测试用, 不建议设定)</p>																														

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

AD1CN5: Σ ADC 控制寄存器 5

位	名称	描述
Bit7	ENACM	ADC Common Mode Voltage <0> 关闭。 <1> 启用。
Bit6	ENV12	Voltage Source 1.2V (Buffer Enable) <0> 选择 REFO。 <1> 选择 V12。
Bit5	VCMS	ADC Common Voltage. <0> VDDA/2。 (HY17P55/56/58 不开放。 HY17P52 开放) <1> 1.2V。
Bit4	LDOPL	内部 250kL 电阻下拉开关 <0>关闭 (预设) <1>启用。 以下几点务必使 LDOPL 为 1，否则结果会不如预期 ※ 使用内部 LDO 输出 ※ ADC 参考电压选用 VDDA/2-VSS
Bit1	ENTPS	内部 TPS 启用控制 <0> 关闭 <1> 启用，需设置相对的 ADC 网络
Bit0	TPSCH	TPS 输出电压反向控制 <0> 正常 <1> 反向

16.8-bit Resistance Ladder 网络

芯片内嵌一个 8-bit resistance ladder 网络，它是由一个保证单调性数字电阻器所构成。搭配 Waveform Generator，可产生波形输出。

- 8-bit resistance ladder 特性包括:
 - 8 位的单调输出
 - 内部或外部基准的可编程选择
 - 可用来当作可编程电阻

8-bit resistance ladder 寄存器摘要：

DACCN0	DANS[2:0], DAPS[3:0]
DACCN1	DALH, DAOE[1:0], ENDA
DACCN2	DABIT[7:0]

Waveform Generator 寄存器摘要：

DGCON1	DGRST, DGDiv[2:0], DGEN
---------------	-------------------------

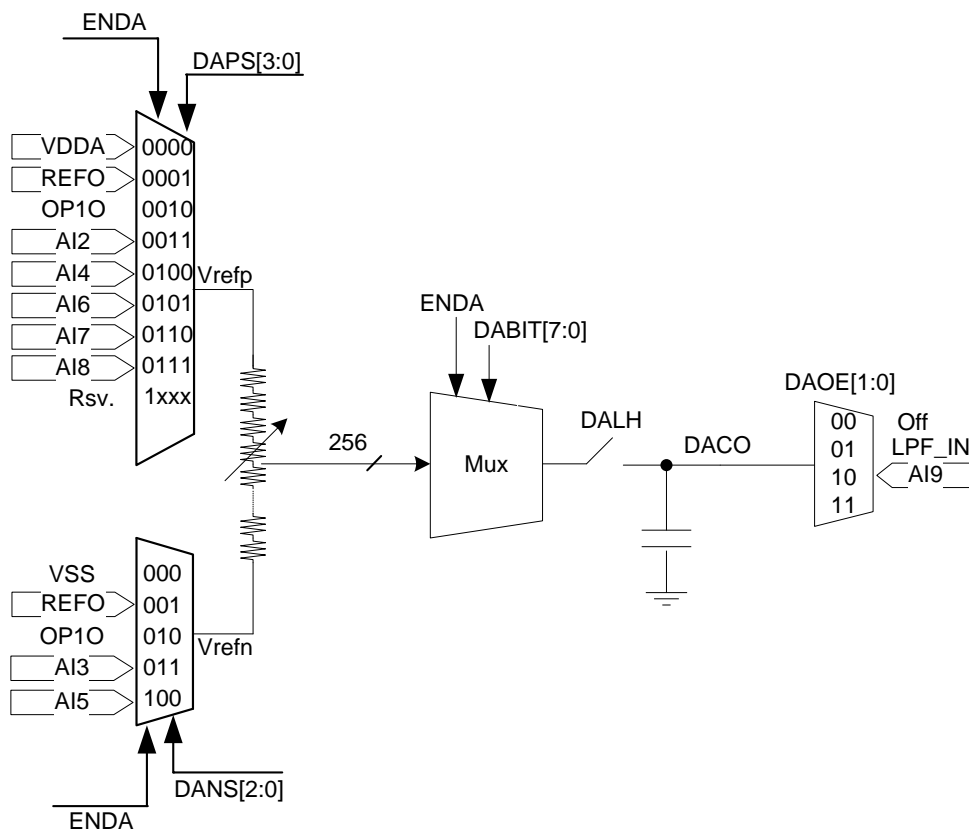


图 16-1 8-bit DAC 方块图

- 8-bit resistance ladder 的运作:

当 ENDA 是 0 ,则 8-bit resistance ladder 会被关闭,就不会消耗电源。DA_Vrefp 多任务器被关闭, 变成一个高阻抗节点。如果 DAOE 被设为 1, 就会变成具有标量欧姆值且每一步骤的可编程电阻。

■ 8-bit resistance ladder 输出:

DAO 依据储存在 DABIT 和 DA_Vrefp – DA_Vrefn 的数据来产生电压输出。

DABIT 是直二进制数据格式。下图显示传输功能图。

$$DAO = (V_{DAC_Vrefp} - V_{DAC_Vrefn}) \times \frac{DAbit_in}{256} + V_{DAC_Vrefn}$$

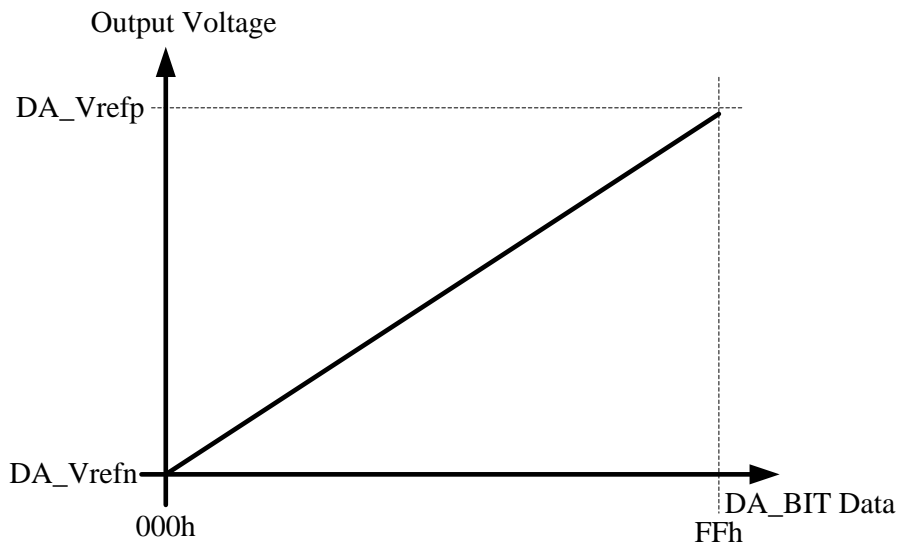


图 16-2 8-bit resistance ladder 转换图

■ 8-bit resistance ladder 初始化配置 :

开启 VDDA 电压 ENLDO 与设置 VDDA 稳压电压输入源 LDOC[2:0] ,开启共模参考电压 ENACM=<1> , VDDA 电压要大于 2.4V , 等待稳定时间。

设置 8-bit resistance ladder 正向与负向的参考电压输入(DANS[2:0]/DAPS[3:0]寄存器) , 并且设置 8-bit resistance ladder 输出电压的初始比例值(DABIT[7:0]寄存器)。

开启 8-bit resistance ladder 输出开关控制, 设置 DAOE[1:0]。

8-bit resistance ladder 功能开启, 设置 ENDA=<1>。

Waveform Generator Function

波形产生器最多有 64 个点(SRAM 地址: 0x300~ 0x33F) ,它可以自己重复。当 DGEN 寄存器为 0 时 ,MCU 可以将数据读入/写入该 SRAM。由 DGRP 设置波形的长度。如果不使用 DAC 波形产生器 , 它可以用作扩展 SRAM。

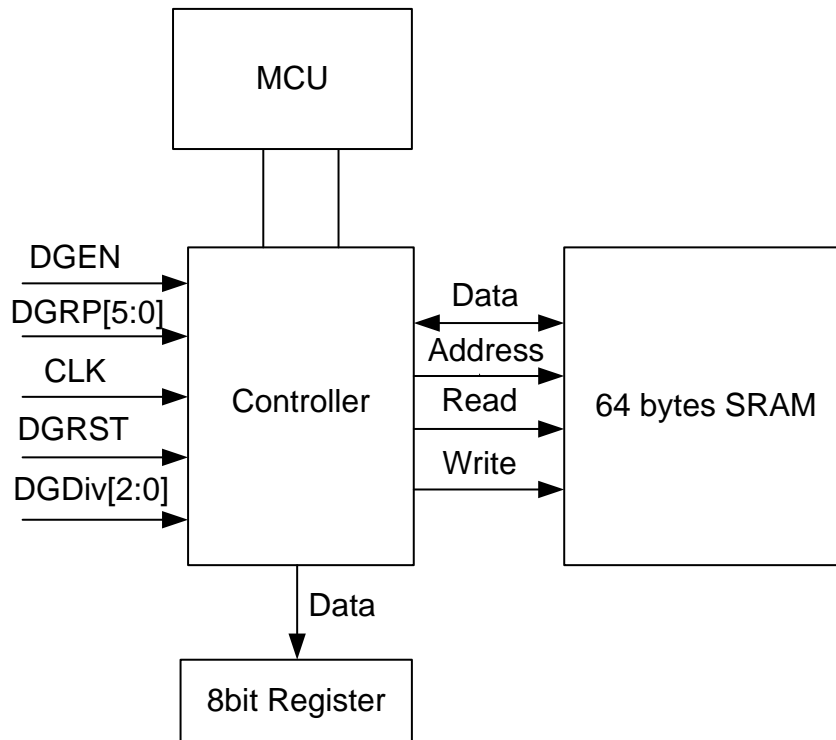


图 16-3 Wave Generator 方块图

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



16.1. 寄存器说明-8-bit resistance ladder

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
DACCN0		DANS[2:0]			DAPS[3:0]				0000 0000	uuuu uuuu	*****
DACCN1				DADCS	DALH	DAOE[1:0]		ENDA	0000 0000	uuuu uuuu	*****
DACCN2	DABIT[7:0]								0000 0000	uuuu uuuu	*****

表 16-1 8-bit resistance ladder 寄存器

DACCN0: DAC 控制寄存器 0

位	名称	描述																																				
Bit6~4	DANS[2:0]	8-bit resistance ladder 负向输入源选择 <table border="1"> <thead> <tr> <th>DANS[2:0]</th> <th>DA_Vrefn</th> <th>DANS[2:0]</th> <th>DA_Vrefn</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>VSS</td> <td>100</td> <td>AI5</td> </tr> <tr> <td>001</td> <td>REFO</td> <td>101</td> <td>-</td> </tr> <tr> <td>010</td> <td>OP1O</td> <td>110</td> <td>-</td> </tr> <tr> <td>011</td> <td>AI3</td> <td>111</td> <td>-</td> </tr> </tbody> </table>	DANS[2:0]	DA_Vrefn	DANS[2:0]	DA_Vrefn	000	VSS	100	AI5	001	REFO	101	-	010	OP1O	110	-	011	AI3	111	-																
DANS[2:0]	DA_Vrefn	DANS[2:0]	DA_Vrefn																																			
000	VSS	100	AI5																																			
001	REFO	101	-																																			
010	OP1O	110	-																																			
011	AI3	111	-																																			
Bit3~0	DAPS[3:0]	8-bit resistance ladder 正向输入源选择 <table border="1"> <thead> <tr> <th>DAPS[3:0]</th> <th>DA_Vrefp</th> <th>DAPS[3:0]</th> <th>DA_Vrefp</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>VDDA</td> <td>1000</td> <td>-</td> </tr> <tr> <td>0001</td> <td>REFO</td> <td>1001</td> <td>-</td> </tr> <tr> <td>0010</td> <td>OP1O</td> <td>1010</td> <td>-</td> </tr> <tr> <td>0011</td> <td>AI2</td> <td>1011</td> <td>-</td> </tr> <tr> <td>0100</td> <td>AI4</td> <td>1100</td> <td>-</td> </tr> <tr> <td>0101</td> <td>AI6</td> <td>1101</td> <td>-</td> </tr> <tr> <td>0110</td> <td>AI7</td> <td>1110</td> <td>-</td> </tr> <tr> <td>0111</td> <td>AI8</td> <td>1111</td> <td>-</td> </tr> </tbody> </table>	DAPS[3:0]	DA_Vrefp	DAPS[3:0]	DA_Vrefp	0000	VDDA	1000	-	0001	REFO	1001	-	0010	OP1O	1010	-	0011	AI2	1011	-	0100	AI4	1100	-	0101	AI6	1101	-	0110	AI7	1110	-	0111	AI8	1111	-
DAPS[3:0]	DA_Vrefp	DAPS[3:0]	DA_Vrefp																																			
0000	VDDA	1000	-																																			
0001	REFO	1001	-																																			
0010	OP1O	1010	-																																			
0011	AI2	1011	-																																			
0100	AI4	1100	-																																			
0101	AI6	1101	-																																			
0110	AI7	1110	-																																			
0111	AI8	1111	-																																			

DACCN1: DAC 控制寄存器 1

位	名称	描述
Bit3	DALH	8-bit resistance ladder 内部输出控制 <0> 关闭 <1> 开启
Bit2~1	DAOE[1:0]	8-bit resistance ladder 输出开启控制. <00> 关闭, 处于高阻态 <01> LPF_IN <10> A9 <11> 保留
Bit0	ENDA	8-bit resistance ladder 功能开启控制 <0> 关闭

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



位	名称	描述
		<1> 开启

DACCN2: DAC 控制寄存器 2

位	名称	描述
Bit7~0	DABIT[7:0]	输出电压的比例值设定，即是 DAO[7:0]/256

16.2. 寄存器说明-Waveform Generator

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
DGCON1				DGRST	DGDIV[2:0]			DGEN	0000 0000	uuuu uuuu	***** r r r r r r r
DGCON2				DGRP[5:0]					0000 0000	uuuu uuuu	***** r r r r r r r

表 16-2 Waveform Generator 寄存器

DGCON1: 波形产生器控制寄存器 1

位	名称	描述																				
Bit4	DGRST	DAC 计数器重置控制器 该位只能设置为“1”，控制器复位后将返回“0”。可以将点数计数器和输出寄存器重置为“0”。 <0> 正常 <1> 重置																				
Bit3~1	DGDIV[2:0]	波形产生器频率分除器 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>DGDIV[2:0]</th> <th>Clock divider</th> <th>DGDIV[2:0]</th> <th>Clock divider</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Disable (divided by 1)</td> <td>100</td> <td>CLK divided by 16</td> </tr> <tr> <td>001</td> <td>CLK divided by 2</td> <td>101</td> <td>CLK divided by 32</td> </tr> <tr> <td>010</td> <td>CLK divided by 4</td> <td>110</td> <td>CLK divided by 64</td> </tr> <tr> <td>011</td> <td>CLK divided by 8</td> <td>111</td> <td>CLK divided by 128</td> </tr> </tbody> </table>	DGDIV[2:0]	Clock divider	DGDIV[2:0]	Clock divider	000	Disable (divided by 1)	100	CLK divided by 16	001	CLK divided by 2	101	CLK divided by 32	010	CLK divided by 4	110	CLK divided by 64	011	CLK divided by 8	111	CLK divided by 128
DGDIV[2:0]	Clock divider	DGDIV[2:0]	Clock divider																			
000	Disable (divided by 1)	100	CLK divided by 16																			
001	CLK divided by 2	101	CLK divided by 32																			
010	CLK divided by 4	110	CLK divided by 64																			
011	CLK divided by 8	111	CLK divided by 128																			
Bit0	DGEN	波形产生器启用控制器 <0> 关闭 <1> 启用																				

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



DGCON2: 波形产生器控制寄存器 2

位	名称	描述																																																																																																														
Bit5~0	DGRP[5:0]	<p>波形产生器重复点数控制器</p> <p><000000> Only DC, no AC</p> <p><000001> 1st → 2nd → 1st → 2nd</p> <p><000010> 1st → 2nd → 3rd → 1st → 2nd → 3rd</p> <p>.....</p> <p><1111111> 1st → 2nd → 3rd → 4th → → 64th → 1st → 2nd</p> <p>Example:</p> <p>DAC 输出所存放在(SRAM 的 0x300~0x33F)数据代表波形振幅, 8 位数据中, MSB 代表正负号, 总共长度为+/-7bit。</p> <p>振幅由 8 位 DAC 的 Vrefp-Vrefn 所决定。</p> <p>如果 SRAM 数据中心值为 0x80, 则最大值则为 0xFF; 最小值则为 0x01。该模式下 SRAM 数值不为 0。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #d9ead3;"> <th>HAO(kHz)</th> <th colspan="3">8000</th> <th colspan="3">4000</th> <th colspan="3">2000</th> </tr> </thead> <tbody> <tr> <td>DGDiv[2:0]</td> <td>5</td><td>0</td><td>0</td> <td>5</td><td>0</td><td>0</td> <td>4</td><td>0</td><td>0</td> </tr> <tr> <td>DG_Divider</td> <td>32</td><td>1</td><td>1</td> <td>32</td><td>1</td><td>1</td> <td>16</td><td>1</td><td>1</td> </tr> <tr> <td>DGRP[5:0]</td> <td>49</td><td>39</td><td>31</td> <td>24</td><td>19</td><td>15</td> <td>24</td><td>9</td><td>7</td> </tr> <tr> <td>Frequency of Waveform Generator (kHz)</td> <td>5</td><td>200</td><td>250</td> <td>5</td><td>200</td><td>250</td> <td>5</td><td>200</td><td>250</td> </tr> <tr style="background-color: #d9ead3;"> <th>HAO(kHz)</th> <th colspan="3">7834</th> <th colspan="3">3686</th> <th colspan="3">1843</th> </tr> <tr> <td>DGDiv[2:0]</td> <td>5</td><td>0</td><td>0</td> <td>4</td><td>0</td><td>0</td> <td>3</td><td>0</td><td>0</td> </tr> <tr> <td>DG_Divider</td> <td>32</td><td>1</td><td>1</td> <td>16</td><td>1</td><td>1</td> <td>8</td><td>1</td><td>1</td> </tr> <tr> <td>DGRP[5:0]</td> <td>48</td><td>38</td><td>30</td> <td>45</td><td>18</td><td>14</td> <td>45</td><td>8</td><td>6</td> </tr> <tr> <td>Frequency of Waveform Generator (kHz)</td> <td>4.996</td><td>200.</td><td>252.</td> <td>5.00</td><td>194</td><td>245.</td> <td>5.008</td><td>204.8</td><td>263.3</td> </tr> <tr> <td></td> <td></td><td>9</td><td>7</td> <td>8</td><td></td><td>7</td> <td></td><td></td><td></td> </tr> </tbody> </table> <p>Frequency of Waveform Generator(Full-Cycle Mode) = SystemFrequency/(DGDiv)/(DGRP+1)</p>	HAO(kHz)	8000			4000			2000			DGDiv[2:0]	5	0	0	5	0	0	4	0	0	DG_Divider	32	1	1	32	1	1	16	1	1	DGRP[5:0]	49	39	31	24	19	15	24	9	7	Frequency of Waveform Generator (kHz)	5	200	250	5	200	250	5	200	250	HAO(kHz)	7834			3686			1843			DGDiv[2:0]	5	0	0	4	0	0	3	0	0	DG_Divider	32	1	1	16	1	1	8	1	1	DGRP[5:0]	48	38	30	45	18	14	45	8	6	Frequency of Waveform Generator (kHz)	4.996	200.	252.	5.00	194	245.	5.008	204.8	263.3			9	7	8		7			
HAO(kHz)	8000			4000			2000																																																																																																									
DGDiv[2:0]	5	0	0	5	0	0	4	0	0																																																																																																							
DG_Divider	32	1	1	32	1	1	16	1	1																																																																																																							
DGRP[5:0]	49	39	31	24	19	15	24	9	7																																																																																																							
Frequency of Waveform Generator (kHz)	5	200	250	5	200	250	5	200	250																																																																																																							
HAO(kHz)	7834			3686			1843																																																																																																									
DGDiv[2:0]	5	0	0	4	0	0	3	0	0																																																																																																							
DG_Divider	32	1	1	16	1	1	8	1	1																																																																																																							
DGRP[5:0]	48	38	30	45	18	14	45	8	6																																																																																																							
Frequency of Waveform Generator (kHz)	4.996	200.	252.	5.00	194	245.	5.008	204.8	263.3																																																																																																							
		9	7	8		7																																																																																																										

17. 运算放大器 OPAMP

OPAMP 是轨对轨运算放大器，可单独使用或搭配 $\Sigma\Delta$ 独使用。

- R2ROP1 特性包括:
可设计为 DAC Output buffer,
可设计为 ADC input buffer,(ADC negative channel)
具单独输入/输出 PAD.
- R2ROP2 特性包括:
输出端可搭配 Peak hold 线路设计,
输入端可以连接身体阻抗测量输入点.
可设计为 ADC input buffer,(ADC positive channel)
单独输入/输出 PAD.

OPAMP 寄存器摘要：

OP1CN0	OP1OS[1:0], ENOP1
OP1NET	OP1PS[2:0], OP1NS[2:0]
OP2CN0	OP2OS[1:0], ENOP2
OP2NET	OP2PS[3:0], OP2NS[2:0]
IQ0	ENIQ[0]
IQ1	QOffset[5:0], IQMODE[0], IQINV[0]

低通滤波器寄存器摘要：

BIACN0	ENLPF, LPFS[1:0]
---------------	------------------

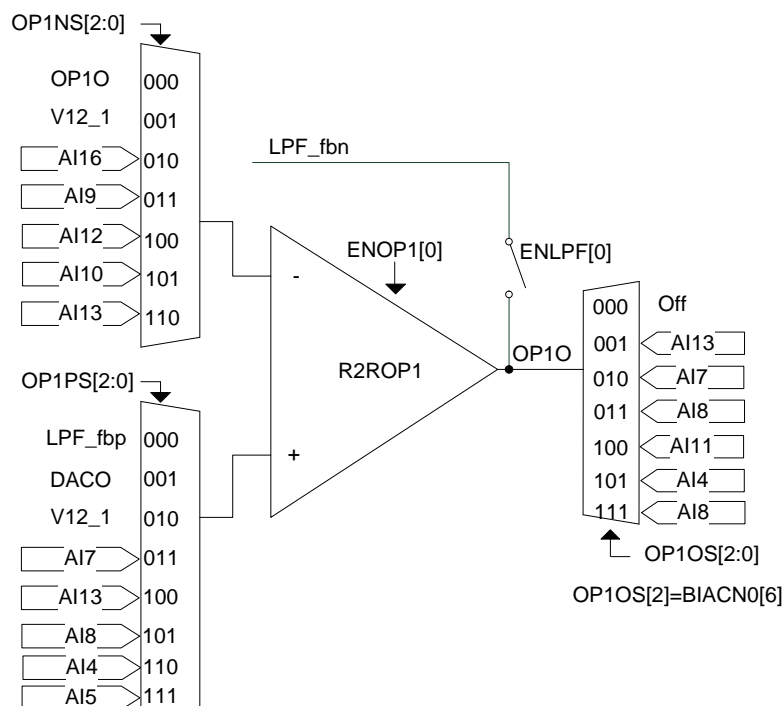


图 17-1 Rail-to-Rail OPAMP1 方块图

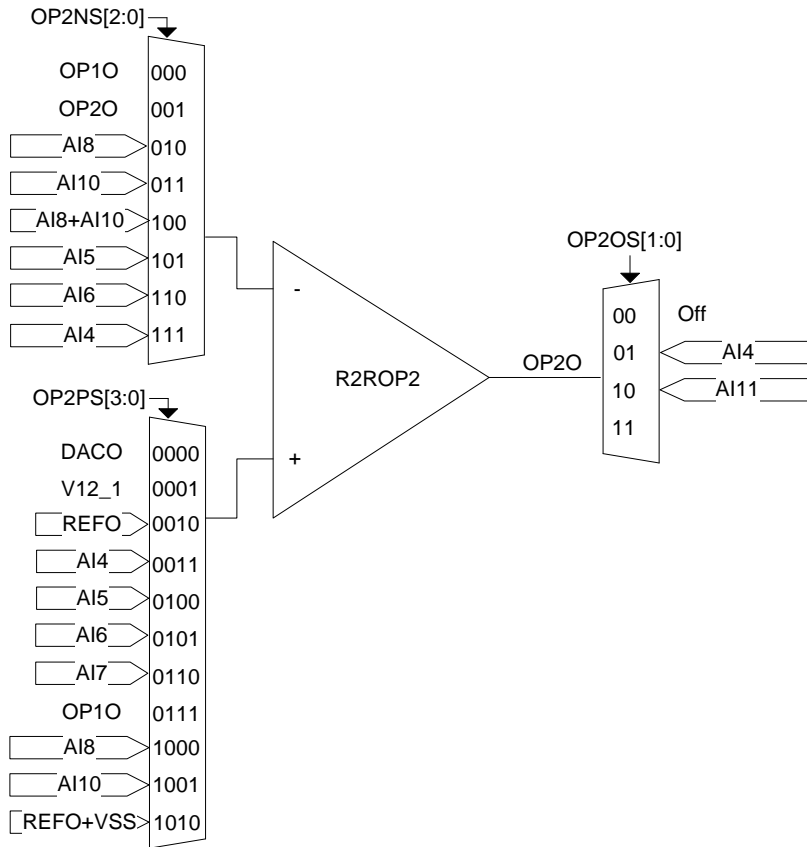


图 17-2 Rail-to-Rail OPAMP2 方块图

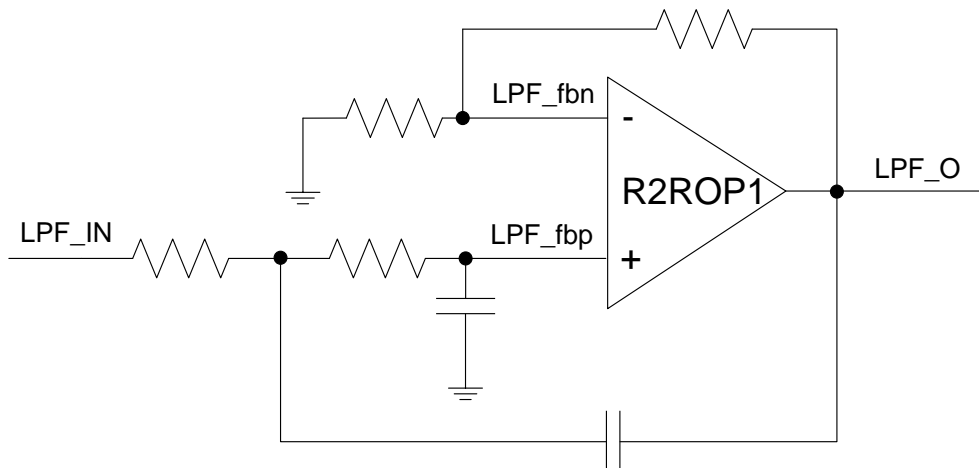


图 17-3 低通滤波器方块图

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



17.1. 寄存器说明- OPAMP

“.”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
OP1CN0						OP1OS[1:0]		ENOP1	0000 0000	uuuu uuuu	*****
OP1NET	OP1PS[2:0]				OP1NS[2:0]				0000 0000	uuuu uuuu	*****
OP2CN0						OP2OS[1:0]		ENOP2	0000 0000	uuuu uuuu	*****
OP2NET	OP2PS[3:0]				OP2NS[2:0]				0000 0000	uuuu uuuu	*****
IQ0								ENIQ	0000 0000	uuuu uuuu	*****
IQ1	IQOffset[5:0]						IQMODE	IQINV	0000 0000	uuuu uuuu	*****

表 17-1 OPAMP 寄存器

OP1CN0: OPAMP1 控制寄存器 0

位	名称	描述																				
Bit2~1	OP1OS[1:0]	OPAMP1 输出信号选择器 <table border="1"> <thead> <tr> <th>OP1OS[2:0]</th> <th>Output</th> <th>OP1OS[2:0]</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Off</td> <td>100</td> <td>AI11</td> </tr> <tr> <td>001</td> <td>AI13</td> <td>101</td> <td>AI4</td> </tr> <tr> <td>010</td> <td>AI7</td> <td>110</td> <td>-</td> </tr> <tr> <td>011</td> <td>AI8</td> <td>111</td> <td>AI8</td> </tr> </tbody> </table>	OP1OS[2:0]	Output	OP1OS[2:0]	Output	000	Off	100	AI11	001	AI13	101	AI4	010	AI7	110	-	011	AI8	111	AI8
OP1OS[2:0]	Output	OP1OS[2:0]	Output																			
000	Off	100	AI11																			
001	AI13	101	AI4																			
010	AI7	110	-																			
011	AI8	111	AI8																			
Bit0	ENOP1	R2ROP1 启用控制器 <0> Disable <1> Enable																				

OP1NET: OPAMP1 输入端选择寄存器

位	名称	描述																				
Bit6~4	OP1PS[2:0]	OPAMP1 正端输入信号选择器 <table border="1"> <thead> <tr> <th>OP1PS[2:0]</th> <th>Input</th> <th>OP1PS[2:0]</th> <th>Input</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>LPF_fbp</td> <td>100</td> <td>AI13</td> </tr> <tr> <td>001</td> <td>DACO</td> <td>101</td> <td>AI8</td> </tr> <tr> <td>010</td> <td>V12_1</td> <td>110</td> <td>AI4</td> </tr> <tr> <td>011</td> <td>AI7</td> <td>111</td> <td>AI5</td> </tr> </tbody> </table>	OP1PS[2:0]	Input	OP1PS[2:0]	Input	000	LPF_fbp	100	AI13	001	DACO	101	AI8	010	V12_1	110	AI4	011	AI7	111	AI5
OP1PS[2:0]	Input	OP1PS[2:0]	Input																			
000	LPF_fbp	100	AI13																			
001	DACO	101	AI8																			
010	V12_1	110	AI4																			
011	AI7	111	AI5																			
Bit2~0	OP1NS[2:0]	OPAMP1 负端输入信号选择器 <table border="1"> <thead> <tr> <th>OP1NS[2:0]</th> <th>Input</th> <th>OP1PS[2:0]</th> <th>Input</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>OP1O</td> <td>100</td> <td>AI12</td> </tr> <tr> <td>001</td> <td>V12_1</td> <td>101</td> <td>AI10</td> </tr> <tr> <td>010</td> <td>AI16</td> <td>110</td> <td>AI13</td> </tr> <tr> <td>011</td> <td>AI9</td> <td>111</td> <td>-</td> </tr> </tbody> </table>	OP1NS[2:0]	Input	OP1PS[2:0]	Input	000	OP1O	100	AI12	001	V12_1	101	AI10	010	AI16	110	AI13	011	AI9	111	-
OP1NS[2:0]	Input	OP1PS[2:0]	Input																			
000	OP1O	100	AI12																			
001	V12_1	101	AI10																			
010	AI16	110	AI13																			
011	AI9	111	-																			

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

OP2CN0: OPAMP2 控制寄存器 0

位	名称	描述										
Bit2~1	OP2OS[1:0]	OPAMP2 输出信号选择器 <table border="1" data-bbox="478 358 766 604"> <thead> <tr> <th>OP2OS[1:0]</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>Off</td> </tr> <tr> <td>01</td> <td>AI4</td> </tr> <tr> <td>10</td> <td>AI11</td> </tr> <tr> <td>11</td> <td>-</td> </tr> </tbody> </table>	OP2OS[1:0]	Output	00	Off	01	AI4	10	AI11	11	-
OP2OS[1:0]	Output											
00	Off											
01	AI4											
10	AI11											
11	-											
Bit0	ENOP2	R2ROP2 启用控制器 <0> Disable <1> Enable										

OP2NET: OPAMP2 输入端选择寄存器

位	名称	描述																																				
Bit7~4	OP2PS[3:0]	OPAMP2 正端输入信号选择器 <table border="1" data-bbox="478 940 1101 1388"> <thead> <tr> <th>OP2PS[3:0]</th> <th>Input</th> <th>OP2PS[3:0]</th> <th>Input</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>DACO</td> <td>1000</td> <td>AI8</td> </tr> <tr> <td>0001</td> <td>V12_1</td> <td>1001</td> <td>AI10</td> </tr> <tr> <td>0010</td> <td>REFO</td> <td>1010</td> <td>REFO+VSS</td> </tr> <tr> <td>0011</td> <td>AI4</td> <td>1011</td> <td>-</td> </tr> <tr> <td>0100</td> <td>AI5</td> <td>1100</td> <td>-</td> </tr> <tr> <td>0101</td> <td>AI6</td> <td>1101</td> <td>-</td> </tr> <tr> <td>0110</td> <td>AI7</td> <td>1110</td> <td>-</td> </tr> <tr> <td>0111</td> <td>OP10</td> <td>1111</td> <td>-</td> </tr> </tbody> </table>	OP2PS[3:0]	Input	OP2PS[3:0]	Input	0000	DACO	1000	AI8	0001	V12_1	1001	AI10	0010	REFO	1010	REFO+VSS	0011	AI4	1011	-	0100	AI5	1100	-	0101	AI6	1101	-	0110	AI7	1110	-	0111	OP10	1111	-
OP2PS[3:0]	Input	OP2PS[3:0]	Input																																			
0000	DACO	1000	AI8																																			
0001	V12_1	1001	AI10																																			
0010	REFO	1010	REFO+VSS																																			
0011	AI4	1011	-																																			
0100	AI5	1100	-																																			
0101	AI6	1101	-																																			
0110	AI7	1110	-																																			
0111	OP10	1111	-																																			
Bit2~0	OP2NS[2:0]	OPAMP2 负端输入信号选择器 <table border="1" data-bbox="478 1433 1069 1680"> <thead> <tr> <th>OP2NS[2:0]</th> <th>Input</th> <th>OP2NS[2:0]</th> <th>Input</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>OP10</td> <td>100</td> <td>AI8+AI10</td> </tr> <tr> <td>001</td> <td>OP20</td> <td>101</td> <td>AI5</td> </tr> <tr> <td>010</td> <td>AI8</td> <td>110</td> <td>AI6</td> </tr> <tr> <td>011</td> <td>AI10</td> <td>111</td> <td>AI4</td> </tr> </tbody> </table>	OP2NS[2:0]	Input	OP2NS[2:0]	Input	000	OP10	100	AI8+AI10	001	OP20	101	AI5	010	AI8	110	AI6	011	AI10	111	AI4																
OP2NS[2:0]	Input	OP2NS[2:0]	Input																																			
000	OP10	100	AI8+AI10																																			
001	OP20	101	AI5																																			
010	AI8	110	AI6																																			
011	AI10	111	AI4																																			

IQ0:

位	名称	描述
Bit0	ENIQ[0]	启动 IQ Clock, 以及启动 Vg_peak 讯号 AND IQ Clock 输出至 AI11 PAD.

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



IQ1:

位	名称	描述
Bit7~3	IQOffset[5:0]	设定 IQ Clock 输出的延迟时间. Max value \leq DNR[5:0].
Bit1	IQMODE[0]	设定 IQ Clock 模式; <0> IQ Clock phase delay= IQOffset[5:0] <1> IQ Clock phase delay= DNR[5:0]/4 + IQOffset[5:0]; (/4 表示 Clock 相位相差 90 相差实际以设计为主;)
Bit0	IQINV[0]	决定 IQ Clock 是否反向; <0> 正向 IQ Clock. <1> 反向 IQ Clock.

17.2. 寄存器说明-低通滤波器

“_”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
BIACN0					ENLPF	LPFS[1:0]			0000 0000	uuuu uuuu	*****

表 17-2 低通滤波器寄存器

BIACN0:寄存器

位	名称	描述
Bit3	ENLPF	Enable the output of low pass filter <0> Disable <1> Enable
Bit2~1	LPFS[1:0]	Select the cutoff frequency of the low pass filter <00> Fc=5kHz <01> Fc=50kHz <1x> Fc=300kHz

18. 可程序放大器, R-Type PGA

R-Type PGA 可单独使用或将信号输出到模拟接脚上

(R-Type PGA 仅 HY17P58 支援, HY17P56/55/52/48 C Type 的操作参考 15 章 模拟数字转换器, $\Sigma\Delta$ ADC)

■ PGA 特性包括:

PGA 包含 LNOP1 及 LNOP2, 支持 x4~x32 倍放大倍率。

LNOP1 可做身体阻抗测量线路使用。

PGA 输入端带有 AI0~AI3 纯模拟输入接脚, 可以支持四脚扫描秤需求。

波形产生器讯号输出后, 可以直接透过内部 2k Ω 电阻接入 LNOP1 输入负端。

LNOP1/LNOP2 输入端网络都是单独控制开关, 可以同时启动多个输入端。

LNOP1 负端回授具有积分电容回路。

LNOP1/LNOP2 输出端可以透过 AIx(x=5~9)输出到芯片接脚。

PGA 寄存器摘要 :

PGACN0 CHM[1:0], ENHS, ENPGAEXT[1:0], ENPGA [1:0]

PGACN1 GAINS[1:0], OPDIEN

PGANET1 LNOP1NS[2:0]

PGANET2 LNOP1PS[2:0]

PGANET3 LNOP2NS[2:0]

PGANET4 LNOP2PS[2:0]

PGANET5 LNOP2OS[2:0], OPCS[0], LNOP1OS[2:0]

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

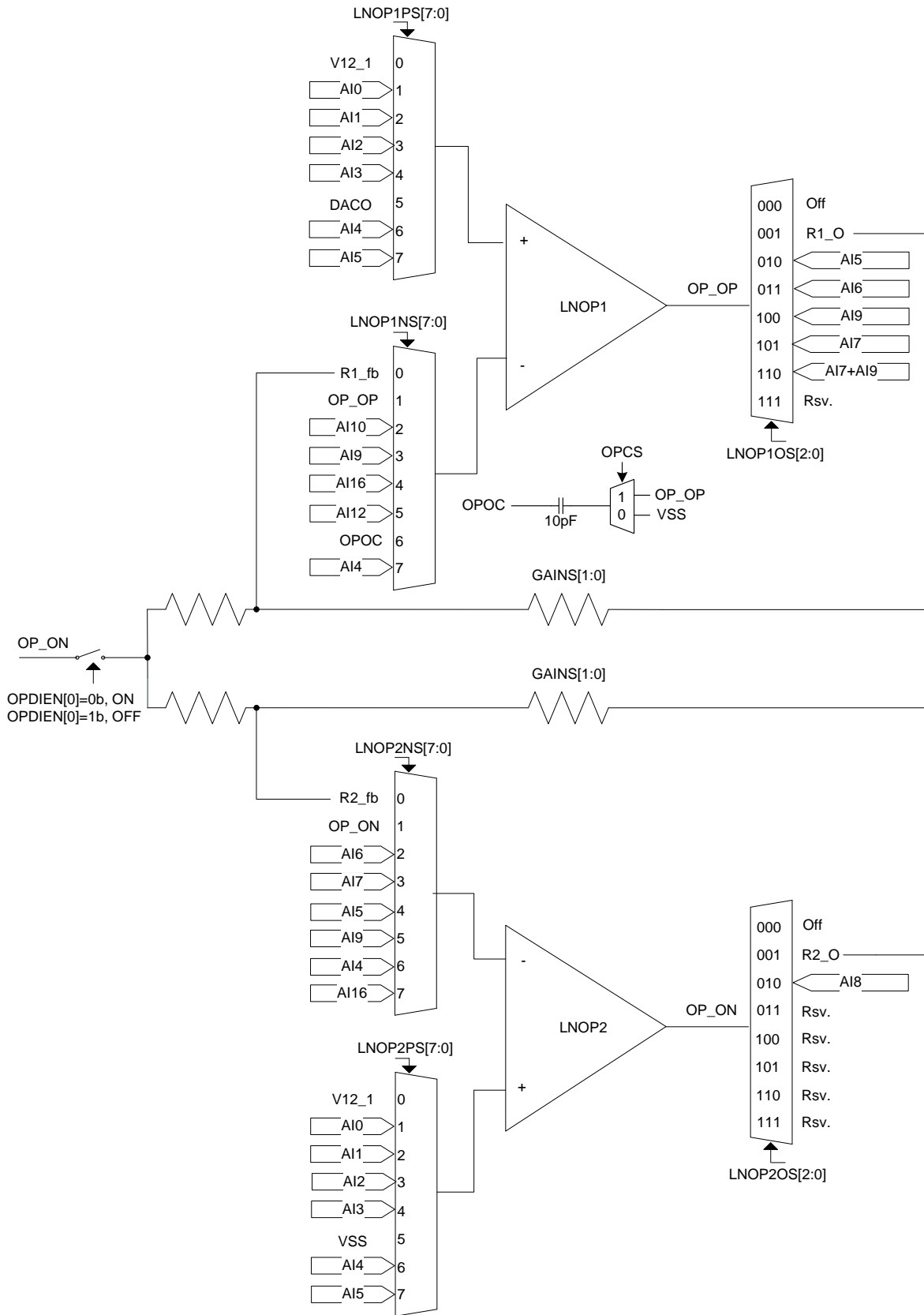


图 18-1 R-type PGA 方块图

18.1. 寄存器说明-可程序放大器

“.”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
PGACN0	-	CHM[1:0]		ENHS	-	-	ENPGA[1:0]		0000 0000	uuuu uuuu	* * * * *
PGACN1	-	-	-	GAINS[1:0]		OPDIEN	ENOPPGA[1:0]		0000 0000	uuuu uuuu	* * * * *
PGANET1	LNOP1NS[7:0]								0000 0000	uuuu uuuu	* * * * *
PGANET2	LNOP1PS[7:0]								0000 0000	uuuu uuuu	* * * * *
PGANET3	LNOP2NS[7:0]								0000 0000	uuuu uuuu	* * * * *
PGANET4	LNOP2PS[7:0]								0000 0000	uuuu uuuu	* * * * *
PGANET5	-	LNOP2OS[2:0]		OPCS	LNOP1OS[2:0]			0000 0000	uuuu uuuu	* * * * *	

表 18-1 可程序放大器寄存器

PGACN0: 可程序放大器控制寄存器 0

位	名称	描述
Bit7	ENCHP	OP Chopper 频率控制器 <0> Chopper frequency= ADC clock/16 <1> Chopper frequency= ADC clock/64
Bit6~5	CHM[1:0]	Chopper 模式控制器 <00> No chopper <01> Enable the loading chopper <10> Enable the input pair chopper <11> Enable both loading and input pair chopper
Bit4	ENHS	Enable high speed OP, but with poor noise performance if no chopper <0> Disable <1> Enable
Bit1	ENPGA[1]	LNOP2 启用控制器 <0> 关闭 LNOP2 <1> 启用 LNOP2。
Bit0	ENPGA[0]	LNOP1 启用控制器 <0> 关闭 LNOP1 <1> 启用 LNOP1。

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

PGACN1: 可程序放大器控制寄存器 1

位	名称	描述
Bit4~3	GAINS[1:0]	Select the gain of OP in PGA mode <00> 4x(不开放) <01> 8x <10> 16x <11> 32x
Bit2	OPDIEN	Enable the fully differential PGA in PGA mode <0> 与 OP_ON 短路 <1> 与 OP_ON 开路

PGANET1: 可程序放大器网络控制寄存器 1

位	名称	描述																				
Bit2~0	LNOP1NS[2:0]	LNOP1 负端输入信号选择器 <table border="1"><thead><tr><th>LNOP1NS</th><th>Input</th><th>LNOP1NS</th><th>Input</th></tr></thead><tbody><tr><td>Bit0</td><td>R1_fb</td><td>Bit4</td><td>AI16</td></tr><tr><td>Bit1</td><td>OP_OP</td><td>Bit5</td><td>AI12</td></tr><tr><td>Bit2</td><td>AI10</td><td>Bit6</td><td>OPOC</td></tr><tr><td>Bit3</td><td>AI9</td><td>Bit7</td><td>AI4</td></tr></tbody></table>	LNOP1NS	Input	LNOP1NS	Input	Bit0	R1_fb	Bit4	AI16	Bit1	OP_OP	Bit5	AI12	Bit2	AI10	Bit6	OPOC	Bit3	AI9	Bit7	AI4
LNOP1NS	Input	LNOP1NS	Input																			
Bit0	R1_fb	Bit4	AI16																			
Bit1	OP_OP	Bit5	AI12																			
Bit2	AI10	Bit6	OPOC																			
Bit3	AI9	Bit7	AI4																			

PGANET2: 可程序放大器网络控制寄存器 2

位	名称	描述																				
Bit2~0	LNOP1PS[2:0]	LNOP1 正端输入信号选择器 <table border="1"><thead><tr><th>LNOP1PS</th><th>Input</th><th>LNOP1PS</th><th>Input</th></tr></thead><tbody><tr><td>Bit0</td><td>V12_1</td><td>Bit4</td><td>AI3</td></tr><tr><td>Bit1</td><td>AI0</td><td>Bit5</td><td>DACO</td></tr><tr><td>Bit2</td><td>AI1</td><td>Bit6</td><td>AI4</td></tr><tr><td>Bit3</td><td>AI2</td><td>Bit7</td><td>AI5</td></tr></tbody></table>	LNOP1PS	Input	LNOP1PS	Input	Bit0	V12_1	Bit4	AI3	Bit1	AI0	Bit5	DACO	Bit2	AI1	Bit6	AI4	Bit3	AI2	Bit7	AI5
LNOP1PS	Input	LNOP1PS	Input																			
Bit0	V12_1	Bit4	AI3																			
Bit1	AI0	Bit5	DACO																			
Bit2	AI1	Bit6	AI4																			
Bit3	AI2	Bit7	AI5																			

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



PGANET3: 可程序放大器控制寄存器 3

位	名称	描述																				
Bit2~0	LNOP2NS[2:0]	LNOP2 负端输入信号选择器																				
		<table border="1"> <thead> <tr> <th>LNOP2NS</th> <th>Input</th> <th>LNOP2NS</th> <th>Input</th> </tr> </thead> <tbody> <tr> <td>Bit0</td> <td>R2_fb</td> <td>Bit4</td> <td>AI5</td> </tr> <tr> <td>Bit1</td> <td>OP_ON</td> <td>Bit5</td> <td>AI9</td> </tr> <tr> <td>Bit2</td> <td>AI6</td> <td>Bit6</td> <td>AI4</td> </tr> <tr> <td>Bit3</td> <td>AI7</td> <td>Bit7</td> <td>AI16</td> </tr> </tbody> </table>	LNOP2NS	Input	LNOP2NS	Input	Bit0	R2_fb	Bit4	AI5	Bit1	OP_ON	Bit5	AI9	Bit2	AI6	Bit6	AI4	Bit3	AI7	Bit7	AI16
LNOP2NS	Input	LNOP2NS	Input																			
Bit0	R2_fb	Bit4	AI5																			
Bit1	OP_ON	Bit5	AI9																			
Bit2	AI6	Bit6	AI4																			
Bit3	AI7	Bit7	AI16																			

PGANET4: 可程序放大器控制寄存器 4

位	名称	描述																				
Bit2~0	LNOP2PS[2:0]	LNOP2 正端输入信号选择器																				
		<table border="1"> <thead> <tr> <th>LNOP2PS</th> <th>Input</th> <th>LNOP2PS</th> <th>Input</th> </tr> </thead> <tbody> <tr> <td>Bit0</td> <td>V12_1</td> <td>Bit4</td> <td>AI3</td> </tr> <tr> <td>Bit1</td> <td>AI0</td> <td>Bit5</td> <td>VSS</td> </tr> <tr> <td>Bit2</td> <td>AI1</td> <td>Bit6</td> <td>AI4</td> </tr> <tr> <td>Bit3</td> <td>AI2</td> <td>Bit7</td> <td>AI5</td> </tr> </tbody> </table>	LNOP2PS	Input	LNOP2PS	Input	Bit0	V12_1	Bit4	AI3	Bit1	AI0	Bit5	VSS	Bit2	AI1	Bit6	AI4	Bit3	AI2	Bit7	AI5
LNOP2PS	Input	LNOP2PS	Input																			
Bit0	V12_1	Bit4	AI3																			
Bit1	AI0	Bit5	VSS																			
Bit2	AI1	Bit6	AI4																			
Bit3	AI2	Bit7	AI5																			

PGANET5: 可程序放大器控制寄存器 5

位	名称	描述																				
Bit6~4	LNOP2OS[2:0]	LNOP2 输出信号选择器																				
		<table border="1"> <thead> <tr> <th>LNOP2OS[2:0]</th> <th>Output</th> <th>LNOP2OS[2:0]</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Off</td> <td>100</td> <td>-</td> </tr> <tr> <td>001</td> <td>R2_O</td> <td>101</td> <td>-</td> </tr> <tr> <td>010</td> <td>AI8</td> <td>110</td> <td>-</td> </tr> <tr> <td>011</td> <td>-</td> <td>111</td> <td>-</td> </tr> </tbody> </table>	LNOP2OS[2:0]	Output	LNOP2OS[2:0]	Output	000	Off	100	-	001	R2_O	101	-	010	AI8	110	-	011	-	111	-
LNOP2OS[2:0]	Output	LNOP2OS[2:0]	Output																			
000	Off	100	-																			
001	R2_O	101	-																			
010	AI8	110	-																			
011	-	111	-																			
Bit3	OPCS[0]	LNOP1 内置电容用途设置 <0> Connect 10pF with OPOC and VSS. <1> Connect 10pF with OPOC and OP_OP.																				
Bit2~0	LNOP1OS[2:0]	LNOP1 输出信号选择器																				
		<table border="1"> <thead> <tr> <th>LNOP1OS[2:0]</th> <th>Output</th> <th>LNOP1OS[2:0]</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Off</td> <td>100</td> <td>AI9</td> </tr> <tr> <td>001</td> <td>R1_O</td> <td>101</td> <td>AI7</td> </tr> <tr> <td>010</td> <td>AI5</td> <td>110</td> <td>AI7+ AI9</td> </tr> <tr> <td>011</td> <td>AI6</td> <td>111</td> <td>-</td> </tr> </tbody> </table>	LNOP1OS[2:0]	Output	LNOP1OS[2:0]	Output	000	Off	100	AI9	001	R1_O	101	AI7	010	AI5	110	AI7+ AI9	011	AI6	111	-
LNOP1OS[2:0]	Output	LNOP1OS[2:0]	Output																			
000	Off	100	AI9																			
001	R1_O	101	AI7																			
010	AI5	110	AI7+ AI9																			
011	AI6	111	-																			

19. LCD 驱动器

液晶驱动电路适用于 TN-LCD 与 STN-LCD 等制程的液晶显示器，其具有以下特点：

- 内置倍压电路(Regulated charge pump)
- 多段可调式驱动电压准位
- 支持 4-duty, 1/3 bias 的 LCD 波形的操作方式
- 可选择输入频率源与可规划输出频率
- 具闪烁控制功能(Blinking capability)

LCD 寄存器摘要：

LCDCN1 ENLCP[0], LCDV[2:0], ENLB[0], SEL_PCLK[0], LCDPU[0]

LCDCN2 LCDTYPE[0], LCDBL[0]

LCD[159:0] LCD0[7:0]~LCD20[7:0]

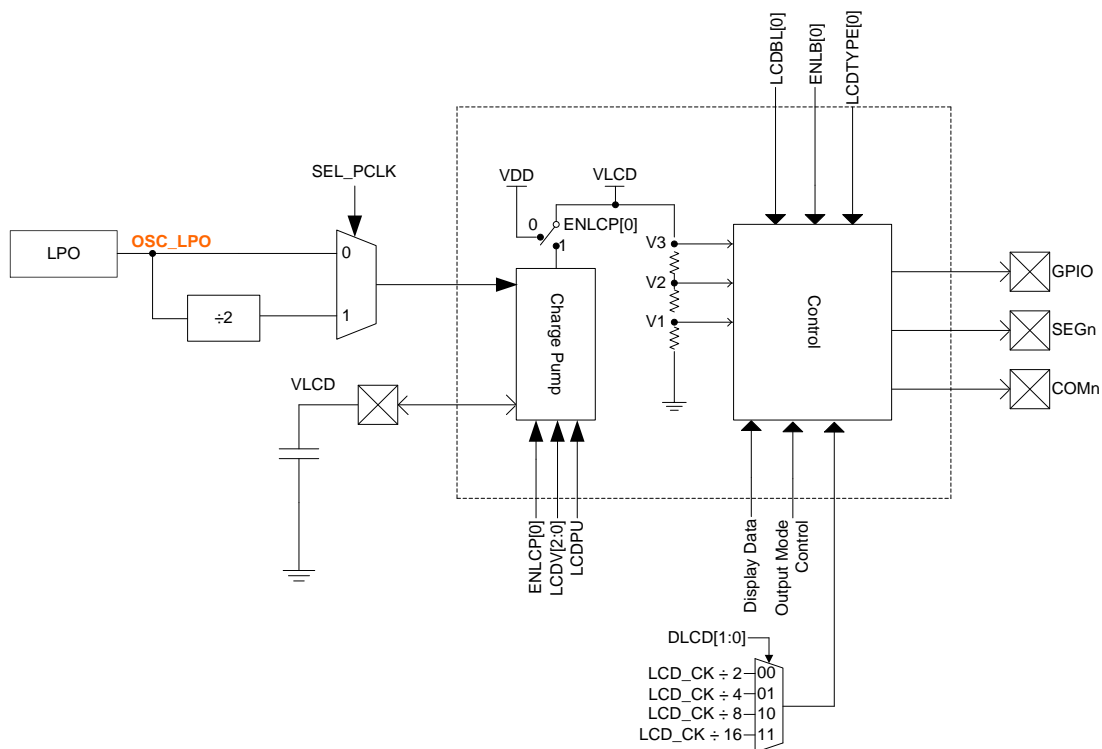


图 19-1 LCD 方块图

19.1. LCD 使用说明

19.1.1. 工作频率与输出帧幅频率设置

工作频率来源由 LCDS 选择，经工作频率预分频器 DLCD[1:0]分频后提供适当的工作频率予 LCD 输出帧幅频率。

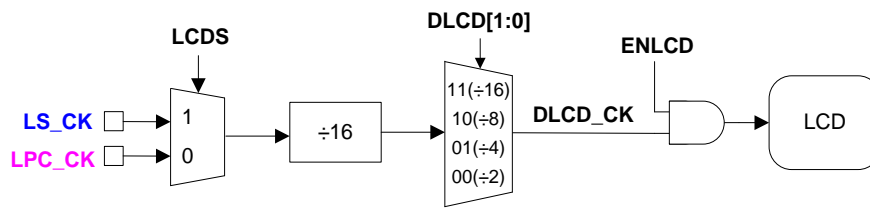


图 19-2 LCD 工作频率与帧频率

19.1.2. 倍压电路与 LCD 工作电压设置

LCD 工作电压源为 VLCD，其产生有两种方式：

- 由外部输入 VLCD 电压源，必须将 ENLCP[0]设置<0> 关闭倍压电路，然后由外部 VLCD 引脚灌入电压以决定 LCD 工作电压。使用外部输入时，LCDV[2:0]的设置对 LCD 工作电压不会产生任何影响。
 - 当推动尺寸或负载较大的 LCD 显示器时，可将 LCD 输出缓冲器 ENLB[0]设置<1>，启用缓冲器以增加 LCD 的驱动能力。反之，ENLB[0]设置<0>则关闭缓冲器亦减少 LCD 的消耗电流。
- 由内部倍压电路产生，将倍压电路控制器 ENLCP[0]设置<1>，并配置倍压电压准位控制器 LCDV[2:0]即可产生 VLCD 电压源供给 LCD，使其不随芯片工作电压的变化而影响显示。
 - LCDV[2:0]可设置不同的工作电压且必须在倍压电路启用时才有效，而倍压电路可能会影响模拟数字转换器 Σ ADC 在高分辨率转换的效能。
 - 使用内部倍压电路产生 VLCD 电压源时，LCD 缓冲器会由内部硬件线路自动启用。
 - 在 LCD 正常显示的时候建议将 LCDPU=1，可以减少 LCD 的耗电。

19.1.3. 闪烁设置

闪烁效果是让 LCD 显示由显示状态快速转为全灭或再转为显示，此等循环过程只需透过字节闪烁控制器 LCDBL[0]设置<1>全灭或设置<0>显示，即可达到快速显示或全灭的状态。故 LCDBL[0]设置<1>时，LCD 显示器不会点亮任何字节。反之，LCDBL[0]设置<0>时，LCD 显示器会依字节寄存器 LCD0[7:0]~ LCD19[7:0]的设置来点亮。

19.1.4. LCD 字节寄存器

每一个字节寄存器 LCDn[7:0]控制两个字节引脚 SEGn，而每个字节引脚具有 4-bit 控制位 SEGn[3:0]。

19.2.LCD Type A 输出波形

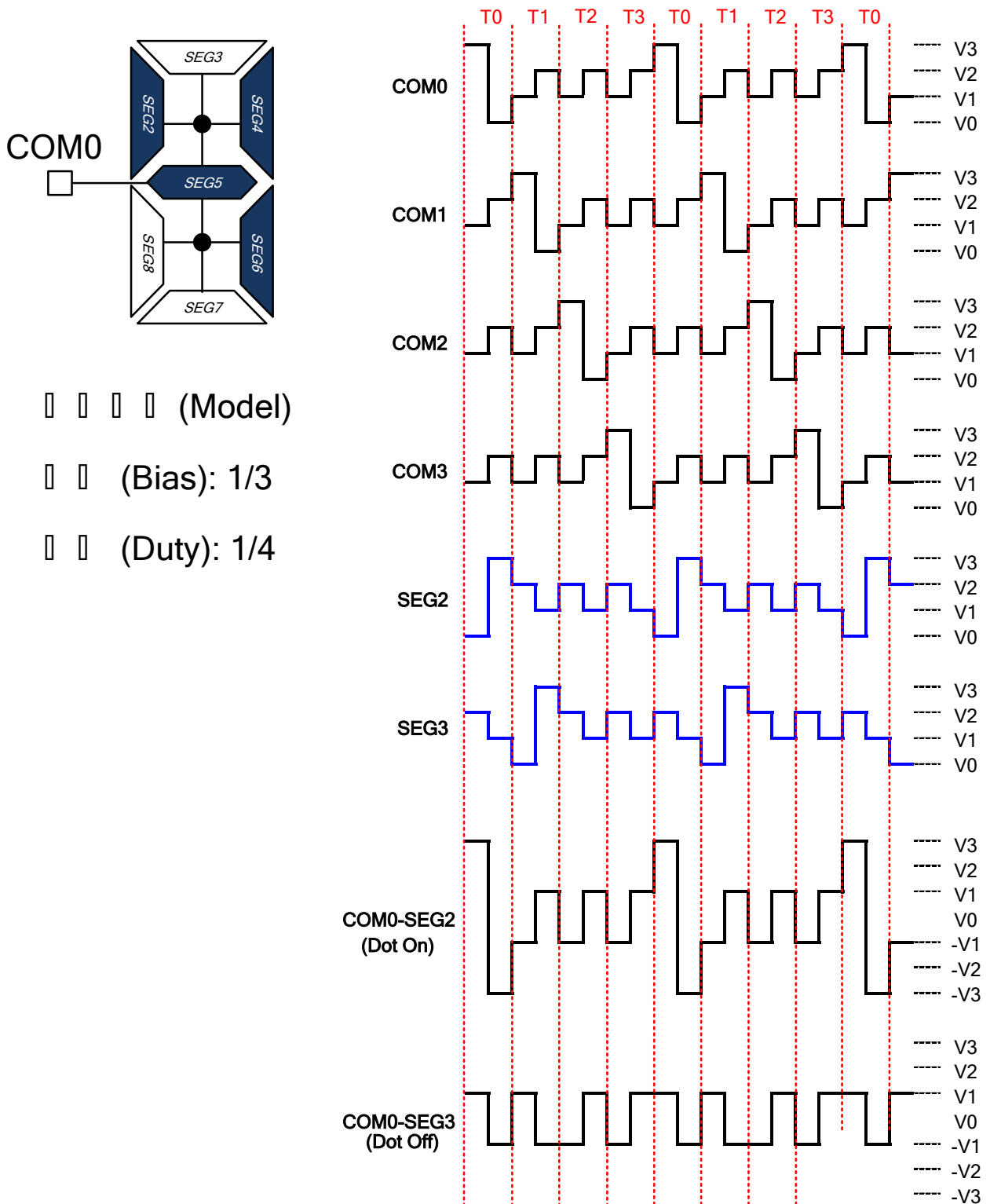
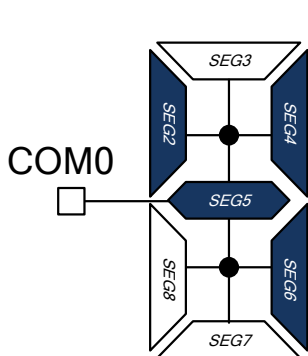


图 19-3 输出波形-4 共线

19.3.LCD Type B 输出波形



□ □ □ □ (Model)

□ □ (Bias): 1/3

□ □ (Duty): 1/4

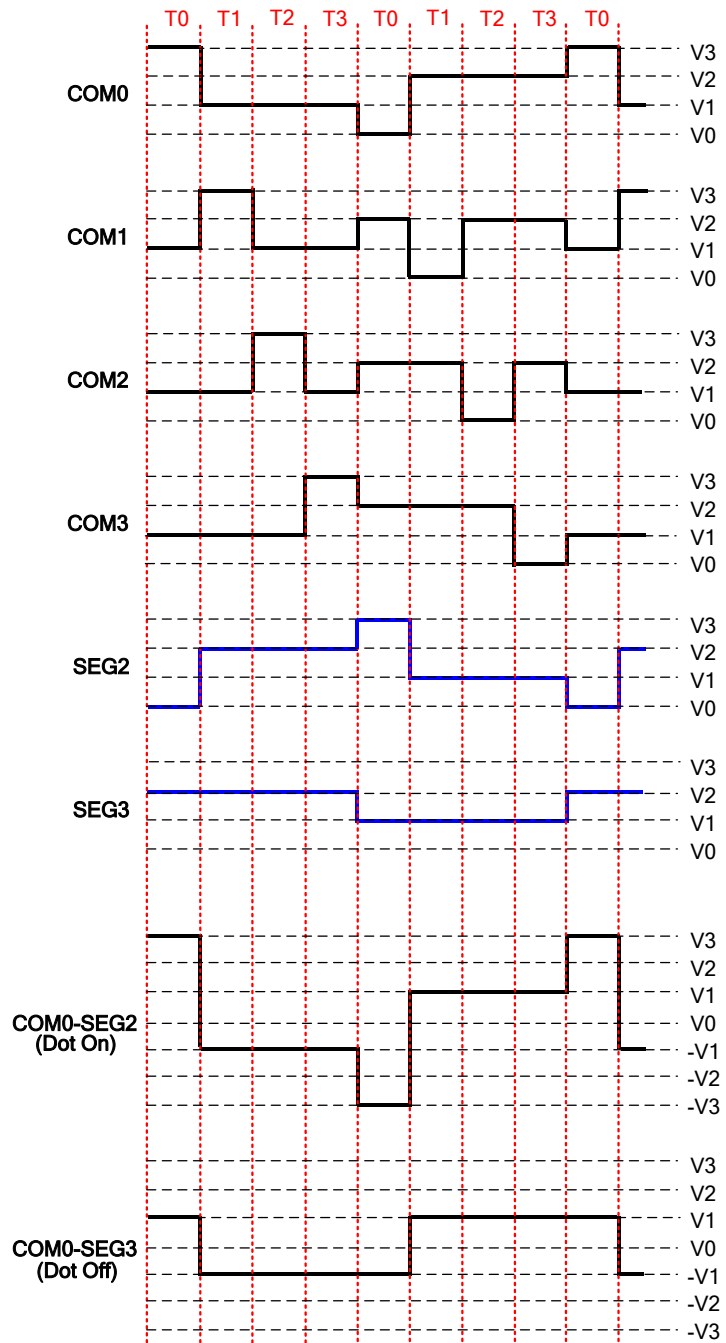


图 19-4 输出波形-4 共线

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



19.4. 寄存器说明-LCD

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
BSRCN	-	-	-	-	-	-	BSR[1]	BSR[0] xxxx uuuu	-, -, -, *, *, *
OSCCN0	OSCS[1:0]		DHS[1:0]		DMS[2:0]			CUPS	0000 0000	uuuu uuuu	*, *, *, *, *, *
OSCCN1								LCDS	0000 0000	uuuu uu.	*, *, *, *, *, *
OSCCN2	DLCD[1:0]		ENXT	XTS[1:0]		HAOM[1:0]		ENHAO	0000 0011	uuuu uu11	*, *, *, *, *, *
LCDCN1	ENLCP	LCDV[2:0]			ENLB	SELCLK	-	LCDFPU	0000 00..	uuuu uu..	*, *, *, *, *, *
LCDCN2	-	-	-	-	-	LCDFTYPE	LCDFBL	LCI	0000 00..	uuuu uu..	*, *, *, *, *, *
LCDCN3	SCM3[1:0]		SCM2[1:0]		SCM1[1:0]		SCM0[1:0]		0000 0000	uuuu uuuu	*, *, *, *, *, *
LCDCN4	SSG21	SSG20	SSG19	SSG18	SSG17	SSG16	SSG15	SSG14	0000 0000	uuuu uuuu	*, *, *, *, *, *
LCDCN5	SSG41	SSG40	SSG39	SSG38	SSG37	SSG36	SSG35	SSG34	0000 0000	uuuu uuuu	*, *, *, *, *, *
LCDCN6	SSG5[1:0]		SSG4[1:0]		SSG3[1:0]		SSG2[1:0]		0000 0000	uuuu uuuu	*, *, *, *, *, *
LCDCN7	SSG9[1:0]		SSG8[1:0]		SSG7[1:0]		SSG6[1:0]		0000 0000	uuuu uuuu	*, *, *, *, *, *
LCDCN8	SSG13[1:0]		SSG12[1:0]		SSG11[1:0]		SSG10[1:0]		0000 0000	uuuu uuuu	*, *, *, *, *, *
LCDCN9	SSG25[1:0]		SSG24[1:0]		SSG23[1:0]		SSG22[1:0]		0000 0000	uuuu uuuu	*, *, *, *, *, *
LCDCN10	SSG29[1:1]		SSG28[1:1]		SSG27[1:1]		SSG26[1:1]		0000 0000	uuuu uuuu	*, *, *, *, *, *
LCDCN11	SSG33[1:1]		SSG32[1:1]		SSG31[1:1]		SSG30[1:1]		0000 0000	uuuu uuuu	*, *, *, *, *, *
LCD0	LCD SEG3[4:7] data				LCD SEG2[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD1	LCD SEG5[4:7] data				LCD SEG4[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD2	LCD SEG7[7:4] data				LCD SEG6[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD3	LCD SEG9[7:4] data				LCD SEG8[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD4	LCD SEG11[7:4] data				LCD SEG10[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD5	LCD SEG13[7:4] data				LCD SEG12[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD6	LCD SEG15[4:7] data				LCD SEG14[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD7	LCD SEG17[4:7] data				LCD SEG16[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD8	LCD SEG19[4:7] data				LCD SEG18[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD9	LCD SEG21[4:7] data				LCD SEG20[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD10	LCD SEG23[4:7] data				LCD SEG22[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD11	LCD SEG25[4:7] data				LCD SEG24[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD12	LCD SEG27[4:7] data				LCD SEG26[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD13	LCD SEG29[4:7] data				LCD SEG28[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD14	LCD SEG31[4:7] data				LCD SEG30[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD15	LCD SEG33[4:7] data				LCD SEG32[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD16	LCD SEG35[4:7] data				LCD SEG34[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD17	LCD SEG37[4:7] data				LCD SEG36[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD18	LCD SEG39[4:7] data				LCD SEG38[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *
LCD19	LCD SEG41[4:7] data				LCD SEG40[3:0] data				xxxx xxxx	uuuu uuuu	*, *, *, *, *, *

表 19-1 LCD 寄存器

BSRCN: 详见 内存章节

OSCCN0/OSCCN1/OSCCN2: 详见 錯誤! 找不到參照來源。章节

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



LCDCN1: LCD 控制寄存器 1

位	名称	描述																				
Bit7	ENLCP	LCD 倍压电路控制器 <0> 关闭；VLCD 电压源由外部引脚灌入。 <1> 启用；VLCD 电压源由芯片内部产生。																				
Bit6~4	LCDV[2:0]	倍压电压准位选择控制器(Test Condition : VDD=3.0V, ENLCP[0]=1, C _{VLCD} =1uF) <table border="1"><thead><tr><th>LCDV [2:0]</th><th>VLCD 输出电压</th><th>LCDV [2:0]</th><th>VLCD 输出电压</th></tr></thead><tbody><tr><td>000</td><td>5.05V</td><td>100</td><td>3.0V</td></tr><tr><td>001</td><td>4.5V</td><td>101</td><td>2.8V</td></tr><tr><td>010</td><td>4.05V</td><td>110</td><td>2.6V</td></tr><tr><td>011</td><td>3.3V</td><td>111</td><td>2.4V</td></tr></tbody></table>	LCDV [2:0]	VLCD 输出电压	LCDV [2:0]	VLCD 输出电压	000	5.05V	100	3.0V	001	4.5V	101	2.8V	010	4.05V	110	2.6V	011	3.3V	111	2.4V
LCDV [2:0]	VLCD 输出电压	LCDV [2:0]	VLCD 输出电压																			
000	5.05V	100	3.0V																			
001	4.5V	101	2.8V																			
010	4.05V	110	2.6V																			
011	3.3V	111	2.4V																			
Bit3	ENLB	LCD 输出缓冲器 <0> 关闭。 <1> 启用，在 LCD 输出模式建议设置为'1'。																				
Bit2	SELPCLK	LCD 倍压电路频率控制 <0> 14.5kHz (预设) <1> 7.25kHz，速度慢可较省电																				
Bit0	LCDPU	LCD 倍压电路能力 <0> 高(启动 BIE 功能时，选择此设定) <1> 低(正常使用 LCD 功能时，选用此设定)																				

LCDCN2: LCD 控制寄存器 2

位	名称	描述
Bit2	LCDTYPE	LCD 驱动模式(HY17P58 only) <0> TYPE A。 <1> TYPE B。
Bit1	LCDBL	LCD 字节闪烁控制器 <0> LCD 字节正常显示。 <1> LCD 字节全灭，只有关闭显示输出。 只需要控制此位，即可做闪烁字节的显示
Bit0	LCI	不开放使用须设定为 0

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

LCDCN3: LCD 控制寄存器 3

位	名称	描述
Bit7~6	SCM3[1:0]	引脚复用功能选择。(x=0~3)
Bit5~4	SCM2[1:0]	<00> PT6.x 数字引脚功能。需自行设定 TRISC6 寄存器。(预设)
Bit3~2	SCM1[1:0]	<11> LCD COMx 引脚设定 ,只要有任一 COM 引脚设为 LCD 模式 ,则 ENLCD = 1。
Bit1~0	SCM0[1:0]	

LCDCN4: LCD 控制寄存器 4

位	名称	描述
Bit7	SSG21	引脚功能选择。(x=14~21)
Bit6	SSG20	<0> PT8 数字引脚功能。需自行设定 TRISC8 寄存器。(预设)
Bit5	SSG19	<1> LCD SEGx 引脚设定。
Bit4	SSG18	
Bit3	SSG17	
Bit2	SSG16	
Bit1	SSG15	
Bit0	SSG14	

LCDCN5: LCD 控制寄存器 5

位	名称	描述
Bit7	SSG41	引脚功能选择。(x=34~41)
Bit6	SSG40	<0> PT11 数字引脚功能。需自行设定 TRISC11 寄存器。(预设)
Bit5	SSG39	<1> LCD SEGx 引脚设定。
Bit4	SSG38	
Bit3	SSG37	
Bit2	SSG36	
Bit1	SSG35	
Bit0	SSG34	

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



LCDCN6: LCD 控制寄存器 6

位	名称	描述
Bit7~6	SSG5[1:0]	引脚复用功能选择。(x=2~5, n=4~7) <00>PT6.n 数字引脚功能。需自行设定 TRISC6 寄存器。(预设) <11> LCD SEGx 引脚设定。
Bit5~4	SSG4[1:0]	
Bit3~2	SSG3[1:0]	
Bit1~0	SSG02[1:0]	

LCDCN7: LCD 控制寄存器 7

位	名称	描述
Bit7~6	SSG9[1:0]	引脚复用功能选择。(x=6~9, n=0~3) <00> PT7 数字引脚功能。需自行设定 TRISC7 寄存器。(预设) <11> LCD SEGx 引脚设定。
Bit5~4	SSG8[1:0]	
Bit3~2	SSG7[1:0]	
Bit1~0	SSG6[1:0]	

LCDCN8: LCD 控制寄存器 8

位	名称	描述
Bit7~6	SSG13[1:0]	引脚复用功能选择。(x=10~13, n=4~7) <00> PT7 数字引脚功能。需自行设定 TRISC8 寄存器。(预设) <11> LCD SEGx 引脚设定。
Bit5~4	SSG12[1:0]	
Bit3~2	SSG11[1:0]	
Bit1~0	SSG10[1:0]	

LCDCN9: LCD 控制寄存器 9

位	名称	描述
Bit7~6	SSG25[1:0]	引脚复用功能选择。(x=22~25, n=0~3) <00>PT9 数字引脚功能。需自行设定 TRISC9 寄存器。(预设) <11> LCD SEGx 引脚设定。
Bit5~4	SSG24[1:0]	
Bit3~2	SSG23[1:0]	
Bit1~0	SSG22[1:0]	

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

LCDCN10: LCD 控制寄存器 10

位	名称	描述
Bit7~6	SSG29[1:0]	引脚复用功能选择。(x=26~29, n=4~7) <00> PT9 数字引脚功能。需自行设定 TRISC9 寄存器。(预设) <11> LCD SEGx 引脚设定。
Bit5~4	SSG28[1:0]	
Bit3~2	SSG27[1:1]	
Bit1~0	SSG26[1:0]	

LCDCN11: LCD 控制寄存器 11

位	名称	描述
Bit7~6	SSG33[1:0]	引脚复用功能选择。(x=30~33, n=8~11) <00> PT10 数字引脚功能。需自行设定 TRISC10 寄存器。(预设) <11> LCD SEGx 引脚设定。
Bit5~4	SSG32[1:0]	
Bit3~2	SSG31[1:0]	
Bit1~0	SSG30[1:0]	

LCD0~LCD19 : LCD 字节数据寄存器

名称	MODE	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCD0	LCD	Segment SEG3				Segment SEG2			
LCD1	LCD	Segment SEG5				Segment SEG4			
LCD2	LCD	Segment SEG7				Segment SEG6			
LCD3	LCD	Segment SEG9				Segment SEG8			
LCD4	LCD	Segment SEG11				Segment SEG10			
LCD5	LCD	Segment SEG13				Segment SEG12			
LCD6	LCD	Segment SEG15				Segment SEG14			
LCD7	LCD	Segment SEG17				Segment SEG16			
LCD8	LCD	Segment SEG19				Segment SEG18			
LCD9	LCD	Segment SEG21				Segment SEG20			
LCD10	LCD	Segment SEG23				Segment SEG22			
LCD11	LCD	Segment SEG25				Segment SEG24			
LCD12	LCD	Segment SEG27				Segment SEG26			
LCD13	LCD	Segment SEG29				Segment SEG28			
LCD14	LCD	Segment SEG31				Segment SEG30			
LCD15	LCD	Segment SEG33				Segment SEG32			
LCD16	LCD	Segment SEG35				Segment SEG34			
LCD17	LCD	Segment SEG37				Segment SEG36			
LCD18	LCD	Segment SEG39				Segment SEG38			
LCD19	LCD	Segment SEG41				Segment SEG40			

20. LED 背光升压输出

LED 背光升压稳压线路，其具有以下特点：

- 具有 15mA 驱动能力
- 八段可调式驱动电压准位

LED 背光升压寄存器摘要：

LEDCN ENLEDP, LEDS[2:0]

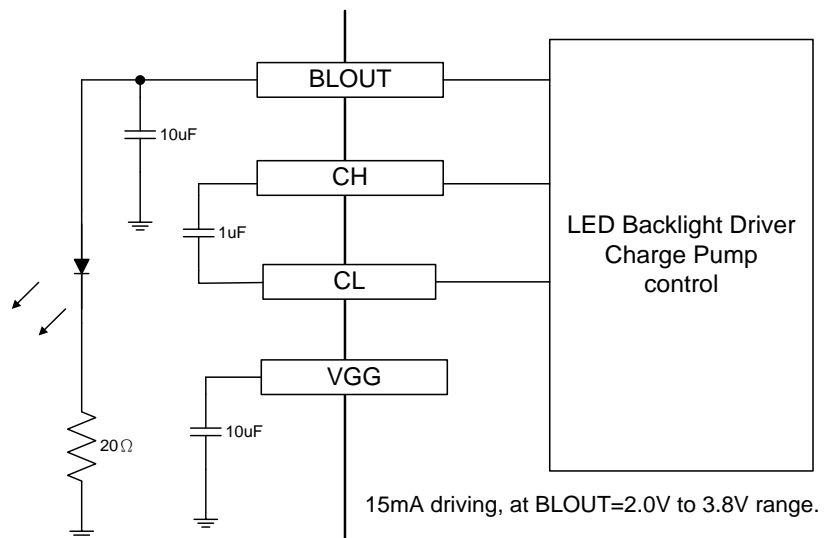


图 20-1 LED 背光方块图

20.1. 寄存器说明-LED 背光升压输出

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
LEDCN			ENLEDP	VGGS	ENBLOUT	LEDS[2:0]			0000 0000	uuuu uuuu	***** r w r w

表 20-1 LED 背光升压输出寄存器

LEDCN: LED 升压控制寄存器

位	名称	描述																				
Bit5	ENLEDP	启动背光升压稳压电源(VGG)控制 <0> 关闭(预设) <1> 启动																				
Bit4	VGGS	VGG 升压启动选择 <0> VGG=VDD(预设) <1> 启动 VGG 升压功能, VGG \approx 2*VDD, ENLEDP 启动后才有效。																				
Bit3	ENBLOUT	BLOUT 稳压启动选择 <0> 关闭, 处于高阻态 (预设) <1> 启动 BLOUT 稳压功能, 稳压输入来自 VGG, ENLEDP 启动后才有效。																				
Bit2~0	LEDS[2:0]	升压稳压电源(BLOUT)输出电压选择弃 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>LEDS[2:0]</th> <th>Output</th> <th>LEDS[2:0]</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>2.0V (预设)</td> <td>100</td> <td>3.2V</td> </tr> <tr> <td>001</td> <td>2.2V</td> <td>101</td> <td>3.4V</td> </tr> <tr> <td>010</td> <td>2.4V</td> <td>110</td> <td>3.6V</td> </tr> <tr> <td>011</td> <td>3.0V</td> <td>111</td> <td>3.8V</td> </tr> </tbody> </table>	LEDS[2:0]	Output	LEDS[2:0]	Output	000	2.0V (预设)	100	3.2V	001	2.2V	101	3.4V	010	2.4V	110	3.6V	011	3.0V	111	3.8V
LEDS[2:0]	Output	LEDS[2:0]	Output																			
000	2.0V (预设)	100	3.2V																			
001	2.2V	101	3.4V																			
010	2.4V	110	3.6V																			
011	3.0V	111	3.8V																			

21. 同步串行通讯接口, Serial Peripheral Interface

串行通讯接口 Serial Peripheral Interface, SPI 外围, SPI 具有以下功能：

- SPI 模块允许同步发送和接收 8 位数据。
- 可用于其他装置进行通信的串行接口, 装置多为 EEPROM、移位寄存器等。
- 供主机模式及从机模式使用。
- 主机模式的引脚配置如下, 使用时须设将 SCK 设定成 Output

	HY17P58	HY17P56	HY17P48
串行数据输出 SDO	PT1.5	PT1.5	PT12.2
串行数据输入 SDI	PT1.0	PT1.2	PT12.1
串行时钟源 SCK	PT1.6	PT1.6	PT12.0

- 从机模式下可以多使用 1 只引脚控制, 被动同步选择引脚 SCE

	HY17P58	HY17P56	HY17P48
被动同步选择引脚 SCE	PT1.1	PT1.1	PT12.3

SPI 寄存器摘要：

SSPCN0 ENSSP[0],CKP[0],CKE[0],SMP[0],SSPM[1:0]

SSPSTA SSPBY[0],SSPOV[0],BF[0]

SSPBUF SSPBUF[7:0]

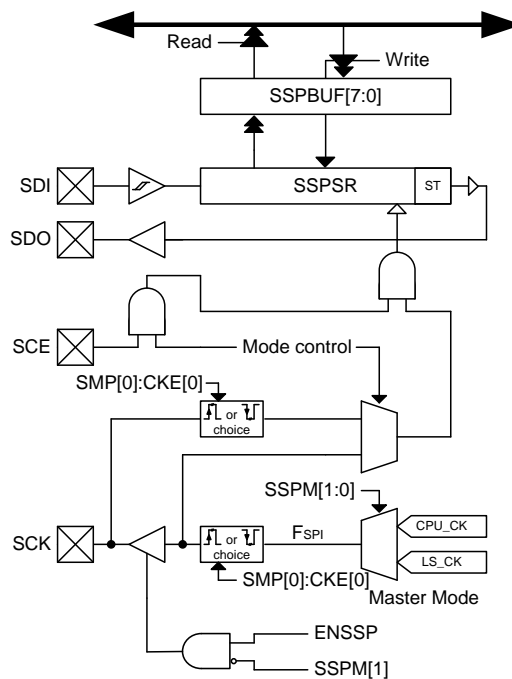


图 21-1 SPI 方块图

21.1. SPI 使用说明

使用时除了相对应引脚须设定为输入输出脚位之外，也因主机从机模式选择，使用脚位不同。

- 主机模式：使用引脚为 SCK(时钟输出), SDI(数据输入), SDO(数据输出)
- 从机模式：使用引脚为 SCK(时钟输入), SDI(数据输入), SDO(数据输出), SCE(同步数据接收致能)

透过配置 SSPCON1 寄存器可设定 SPI 等控制位：

- 设置 ENSSP 位可用于启动 SPI 通讯模块。
- 设置 CKP 位，决定传输完成后 SCK 所处电平状态。
- 设置 CKE 位，决定传输数据于 SCK 上升或下降缘。
- 设置 SMP 位，决定输入数据采样时间点。(频率中间或末尾采样)。
- 设置 SSPM[1:0]字节，决定主机模式 SCK 频率源或从机模式 SCE 引脚是否启用。
- 透过配置 SSPBUF 寄存器决定传输数据或接受数据，SSPSTA 寄存器则反映传送及接收状态：
- SPI 传送及接收寄存器由 SSPBUF 及 SSPSR 寄存器所组成。
- SSPBUF 会保持上次写入 SSPSR 的数据，直到下一次接收到的数据准备好为止。当 8 位数据接收完成后，该数据就会被移入 SSPBUF 寄存器中，而 SSPSTA 寄存器中 BF 位及 INTF2 寄存器中 SSPIF 位将会被设置为 1。
- 双重寄存器将可以允许在读取该笔接收资料(Read SSPBUF)的同时，接收下一笔数据在 SSPSR 寄存器中。当数据接收时，应先判断 SSPSTA 寄存器 BF 位是否被设置为 1，若为 1 则表示 SSPBUF 中有接收完成的数据但却未被用户读取，用户应先读取 SSPBUF 数据，读取后 BF 位将被硬件自动清除为 0。若使用者在 BF 被设置为 1 的情形下，却未读取 SSPBUF 的数据，当再次接收到数据时，SSPSTA 寄存器 SSPOV 位将会被设置为 1，而该笔数据将会遗失不会被写入到 SSPBUF 寄存器中。
- 当数据在传送时 SSPSTA 寄存器 SSPBY 位将被设置为 1，任何写入 SSPBUF 寄存器的动作都会被忽略。传送数据完成后，SSPBY 位将自动被清为 0。
- 当 SPI 为主机模式时，则可以不理睬 SSPBUF 寄存器所接收的数值，只须写入欲传送数据即可。

21.2. SPI 主机模式

SPI 设定为主机模式时，可以随时启动数据传输动作，只要在数据被写入 SSPBUF 寄存器时，数据就会透过 SDO 引脚搭配 SCK 频率源输出。

此时若接收数据模块同为 SPI 模块时，可以将 SPI 从机模块 SDO 引脚配置为输入脚位，则不会传回失效数据给主机模块。若接收模块须同步传回数据给主机模块，则可将从机模块 SDO 引脚配置为输出脚位，则主机模块 SDI 引脚则会连续接收到移入数据，等接收完成后，会将数据写入 SSPBUF 寄存器中，并且相对应的 SSPSTA 寄存器 BF 位及 INTF1 寄存器中 SPIIF 位将会被设置为 1。

同时可以透过 SSPCON1 寄存器 SSPM[1:0]字节来设定主机模块频率源，并透过 CKE, CKP 位设定值来决定频率源的极性。

主机模式配置说明：

- 驱动引脚设定：使用引脚为 SCK(时钟输出), SDI(数据输入), SDO(数据输出)
- 须先正确设定 I/O 引脚输入输出功能，及 SCK, SDO 输出模块功能。
- 配置 SSPCON1 寄存器达 SPI 功能控制位
- 设置 CKP 位，决定传输完成后 SCK 所处电平状态。
- 设置 CKE 位，决定传输数据于 SCK 上升或下降缘。
- 设置 SMP 位，决定输入数据采样时间点。(频率中间或末尾取样)。
- 设置 SSPM[1:0]字节，决定主机模式 SCK 频率源。
- 设置 ENSSP 位可用于启动 SPI 通讯模块。
- 配置 SSPBUF 寄存器决定传输数据，SSPSTA 寄存器则反映传送状态
- SPI 传送寄存器由 SSPBUF 及 SSPSR 寄存器所组成。
- 当数据写入 SSPBUF 后，硬件动作会将 SSPBUF 数据搬移到 SSPSR 寄存器中，并将 SSPSR 数据随 SCK 频率源送出。
- 当数据传送出去之后，SSPSR 会同步接收到 SDI 引脚上的数据输入，并在接收完成后，硬件动作会将 SSPSR 内的数据搬移到 SSPBUF 中。可透过 BF 标志判断是否接收完成。
- SSPBUF 会保持上次从 SSPSR 所接收到的数据，直到下一次接收到的数据准备好为止。
- 当 8 位数据接收完成后，该数据就会被移入 SSPBUF 寄存器中，而 INTF1 寄存器中 SPIIF 位将在 SCK 传送完成 8 个位数据后，会被设置为 1。而 SSPSTA 寄存器中 BF 位将在数据接收完成并由硬件搬移到 SSPBUF 后被设置为 1。
- 当数据在传送时 SSPSTA 寄存器 SSPBY 位将被设置为 1，任何写入 SSPBUF 寄存器的动作都会被忽略。传送数据完成后，SSPBY 位将自动被清为 0。
- 当 SPI 为主机模式时，则可以不理睬 SSPBUF 寄存器所接收的数值，只须写入欲传送数据即可。若 SSPBUF 所接收的数值可使用时，用户应自行将数据搬移后，再写入欲送的数据。

相关配置可见下图说明：

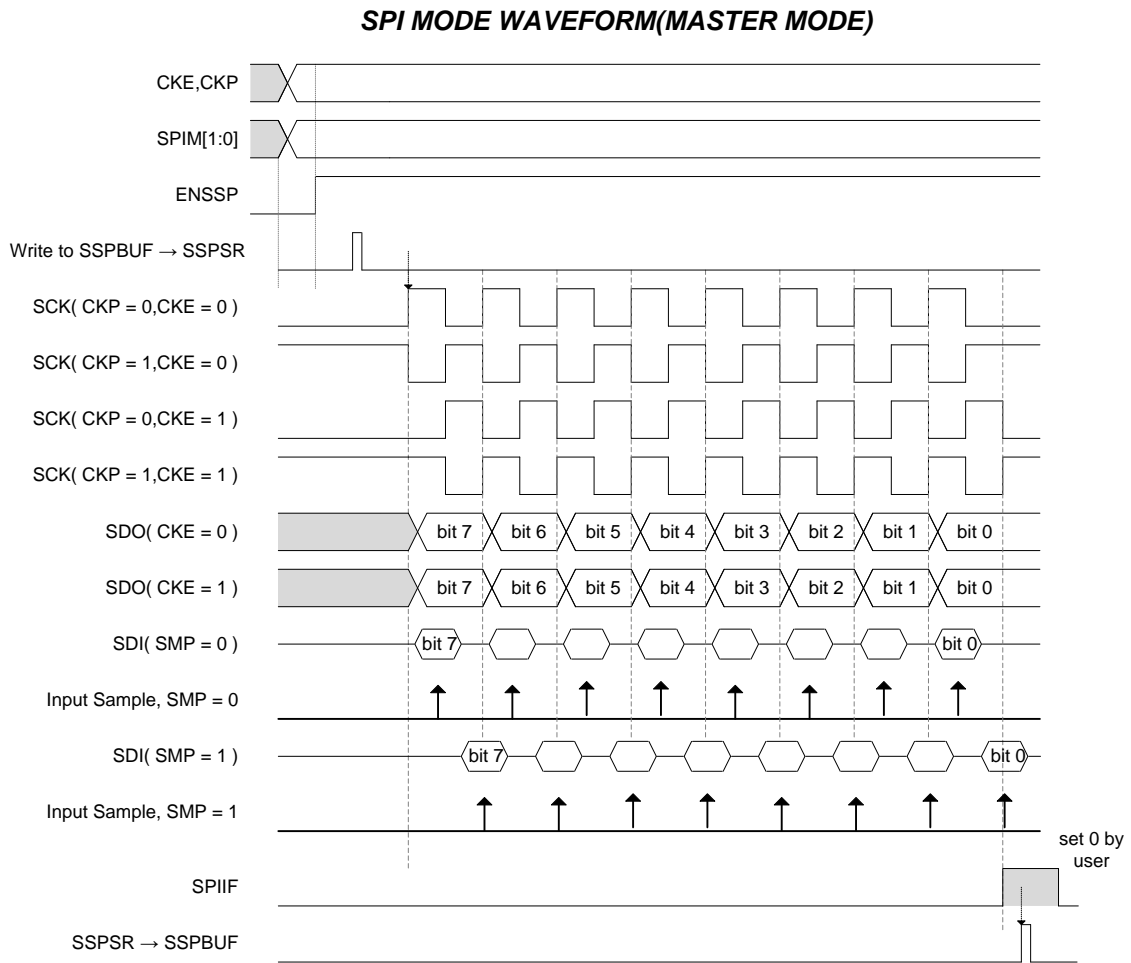


图 21-2 SPI 主机模式时序波形

21.3. SPI 从机模式

从机模式下，SCK 频率源由外部所提供，且 SCK 引脚必须处于空闲状态，并须将 SCK 引脚配置为输入脚位。在搭配主机模块频率源极性设定时，可以透过 CKE, CKP 位设定值来决定从机模块频率源的极性。

从机模式配置说明：

- 驱动引脚设定：使用引脚为 SCK(时钟输入), SDI(数据输入), SDO(数据输出), SCE(同步数据接收致能)
- 须先正确设定 I/O 引脚输出输入功能，及 SDO 输出模块功能。
- 配置 SSPCON1 寄存器达 SPI 功能控制位
- 设置 CKP 位，决定传输完成后 SCK 所处电平状态。
- 设置 CKE 位，决定传输数据于 SCK 上升或下降缘。
- 设置 SMP 位，决定输入数据采样时间点。(频率中间或末尾采样)
- 设置 SSPM[1:0]字节，决定从机模式 SCE 引脚是否启用。
- 设置 ENSSP 位可用于启动 SPI 通讯模块。
- 透过配置 SSPBUF 寄存器决定接受与同步发送数据，SSPSTA 寄存器则反映接收状态
- SPI 接收寄存器由 SSPBUF 及 SSPSR 寄存器所组成。
- 接收数据前，应先将同步发送的数据写入 SSPBUF 中，等待主控端频率源到达。
- 即使不要同步发送数据，仍须完成写入 SSPBUF 的动作，并建议写入 0FFh 的数据。
- 写入同步发送数据之后到主控端频率源到达之前的时间需要延迟 5 个指令周期的时间，以利硬件将 SSPBUF 数据正确搬移到 SSPSR 寄存器中。
- 当主控端 SCK 频率源输入时，从机模块除撷取 SDI 输入引脚数据外，仍会同步将 SSPSR 的数据由 SDO 引脚输出给主控端。
- SSPBUF 会保持上次接收到的 SSPSR 的数据，直到下一次接收到的数据准备好为止。
- 当 8 位数据接收完成后，该数据就会被移入 SSPBUF 寄存器中，而 INTF1 寄存器中 SPIIF 位将在主控 SCK 传送完成 8 个位数据后，会被设置为 1。而 SSPSTA 寄存器中 BF 位将在数据接收完成并由硬件搬移到 SSPBUF 后被设置为 1。
- 双重寄存器将可以允许在读取该笔接收资料(Read SSPBUF)的同时，接收下一笔数据在 SSPSR 寄存器中。当数据接收时，应先判断 SSPSTA 寄存器 BF 位是否被设置为 1，若为 1 则表示 SSPBUF 中有接收完成的数据但却未被用户读取，用户应先读取 SSPBUF 数据，读取后 BF 位将被硬件自动清除为 0。若使用者在 BF 被设置为 1 的情形下，却未读取 SSPBUF 的数据，当再次接收到数据时，SSPSTA 寄存器 SSPOV 位将会被设置为 1，而该笔数据将会遗失不会被写入到 SSPBUF 寄存器中。

相关配置可见下图说明：

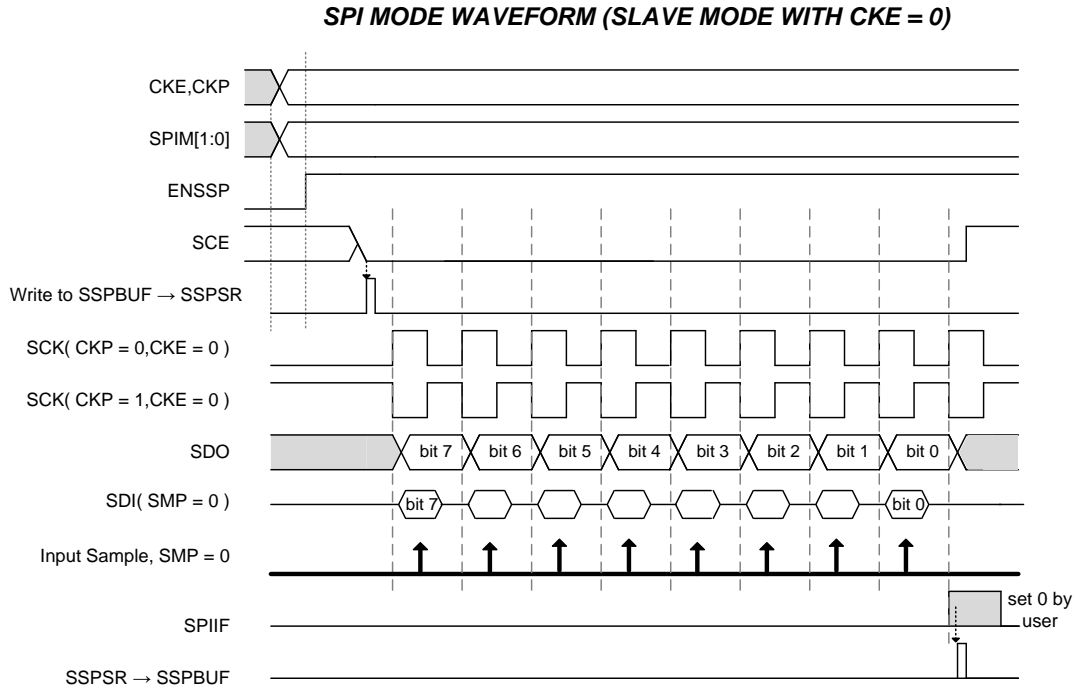


图 21-3 SPI 从机模式时序波形(CKE=0)

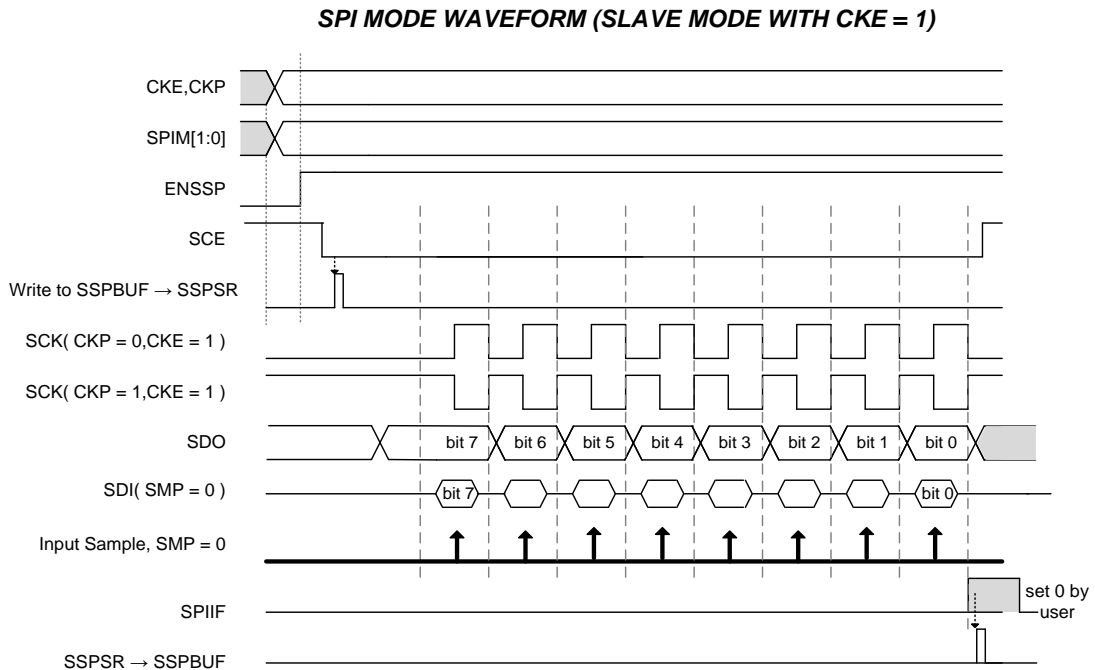


图 21-4 SPI 从机模式时序波形(CKE=1)

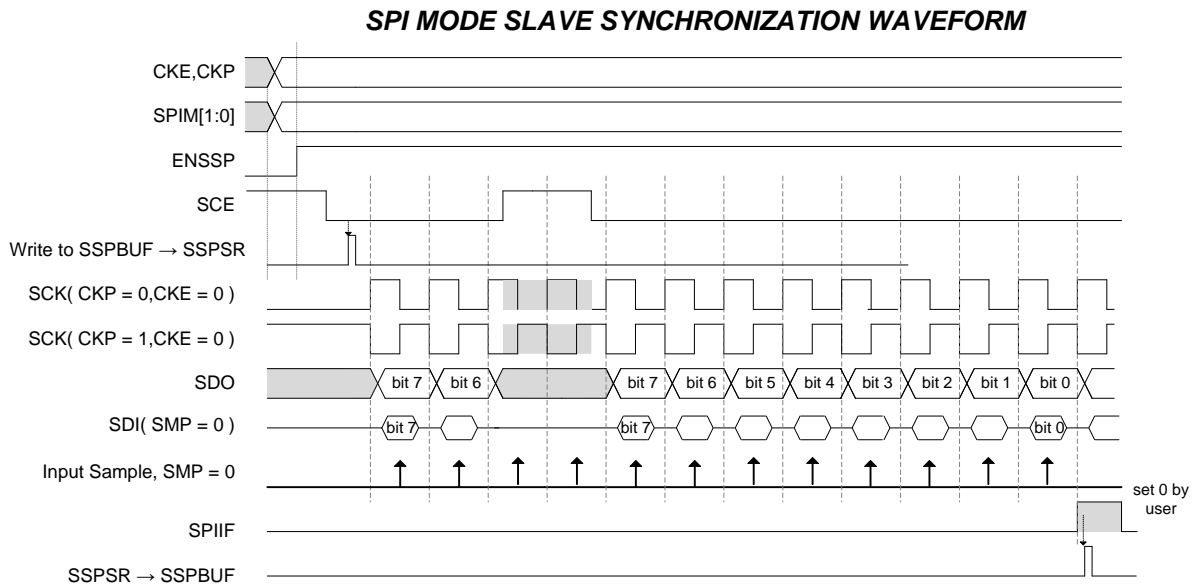


图 21-5 SPI 同步从机模式时序波形

在睡眠模式下(SLP MODE),若已经启动 INTE1 寄存器 SPIIE 位中断致能,则在接收到一笔完整 8 位数据后芯片将被唤醒。

从机模块中可以有控制另一只引脚 SCE, SCE 引脚的配置允许达到被动同步模式,可以透过 SSPCN0 寄存器 SSPM[1:0]字节达到该引脚设定。

当 SCE 引脚处于低电平时,资料可以作正常传送及接收动作,同时 SDO 引脚可以正常驱动。在 SCE 处于高电平时, SDO 输出引脚将被悬空不被驱动。

21.4. SPI 主从机模式传输方式

下图为两组纮康处理器中 SPI 模块主机从机连接方式:

主机模块会将 SSPBUF 寄存器数据透过 SSPSR 移位寄存器做数据传送, 并透过 SDO 数据传输引脚输出。传送数据时, 同步可以接收从机模块所传回数据于 SSPSR 移位寄存器中, 当接收完成后会将数据写入 SSPBUF 寄存器中。

从机模块会将接收到的数据暂存于 SSPSR 移位寄存器中, 待接收完成后才会将数据写入 SSPBUF 寄存器中。

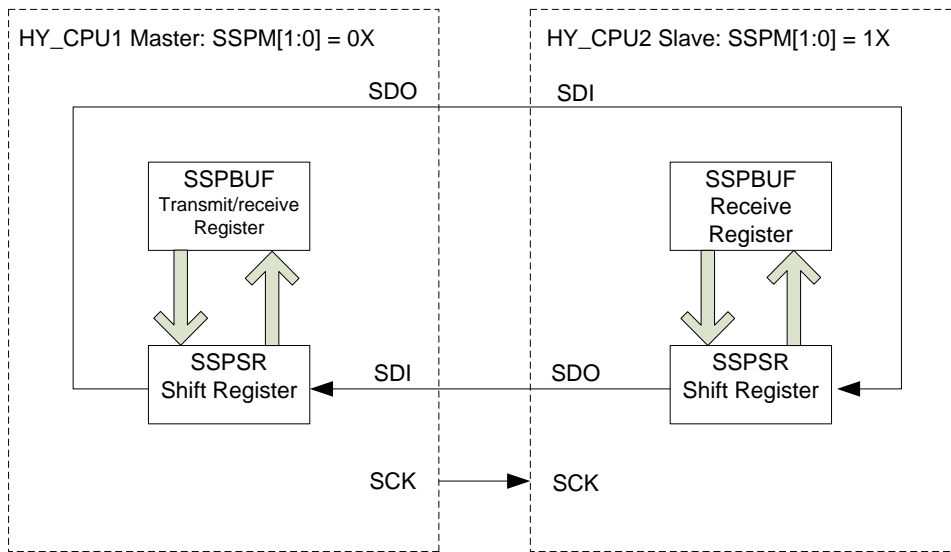


图 21-6 两组处理器 SPI 主机从机连接方式


```
CLRF    PT1DA,0      ; 设定 PT16(SCK), PT15(SDO)为数字输出脚位
        ; 设定 PT12(SDI)为数字输入脚位
CLRF    PT1PU,0
MVL     060H
MVF     TRISC1,1,0
MVL     00010100b
MVF     PT1M2,1,0    ; 设定 SCK, SDO 功能
BSF     INTE1,2,0    ; 设置 SPIIE 中断服务
MVL     080H         ; 启动 SPI 功能, 设定频率源为 CPU_CK
MVF     SSPCON1,1,0 ; 设定为主机模式, 配置 CKP, CKE 设定频率源波形
MVL     055H         ; 写入主机模块欲传送的数据
MVF     SSPBUF,1,0
....

SPI Interrupt :      ; SPI 中断事件服务程序
BCF     INTF1,SPIIF,0 ; 清除 SPI 中断标志
BTSS    SSPSTA,BF,0
RJ      SPI Interrupt ; 判断同步接收从机模块所传送数据是否已接收完成
MVFF    SSPBUF,BUF0 ; 将从从机模块接收到数据搬移到 BUF0 缓存器
...

RETI    ; 中断服务返回
```

范例 21-3 SPI 主机模式范例程序

```
CLRF    PT1DA,0      ; 设定 PT15(SDO)为数字输出脚位
        ; 设定 PT16(SCK), PT12(SDI)为数字输入脚位
CLRF    PT1PU,0
MVL     020H         ; PT1.5 ( SDO),PT.6(SCK),PT1.2(SDI),PT1.1(SCE)
MVF     TRISC1,1,0
MVL     00000100b    ; 设定 SDO 功能
MVF     PT1M2,1,0
BSF     INTE1,SPIIE,0 ; 使能 SPI 中断
MVL     11000011b    ; 启动 SPI,下降沿数据发送
MVF     SSPCN0,1,0   ; 设为从机模式并启动 SCE 引脚控制功能
MVF     SSPBUF,0,0   ; 读出 SPI 缓冲区的值
CLRF    SSPSTA,0
MVL     05Ah         ; 往 SSPBUF 写入数据准备同步发送数据
MVF     SSPBUF,1,0
NOP
        ; 其他程序动作
.....
SPI_interrupt:
BCF     INTF1,SPIIF,0
BTSS    SSPSTA,BF,0
RJ      SPI_Interrupt ; 判断数据是否已接收完成
MVFF    SSPBUF,BUF0 ; 存取主机端发过来的数据
BTSZ    SSPSTA, BF,0
RJ      SPI_InterruptA ; 判断接收同时, 是否有另一笔数据又再输入
BTSZ    SSPSTA, SSPOV,0
RJ      SPI_InterruptB ; 判断资料是否溢出
BCF     SSPSTA, SSPOV,0 ; 若有冲突须清除
MVL     069h         ; 往 SSPBUF 写入一个数据,
        ; 准备接收主机发过来的下笔数据时, 同步发送出去
        ; 若不需同步发送, 请填 0FFh 数据
MVF     SSPBUF,1,0   ; SSPBUF 这个写入的数值为同步发送出去的数值
SPI_InterruptA:
...
SPI_InterruptB:
...
RETI
```

21.5. 寄存器说明- SPI

“.”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“.”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	-	ADIE	WDTIE	TB1IE	TMAIE	E1IE	E0IE	0000 0000	0uuu uuuu	***,***
INTE1	-	SPIIE							0000 0000	uuuu uuuu	***,***
INTF1	-	SPIIF							0000 0000	uuuu uuuu	***,r,r,***
SSPCN0	ENSSP	CKP	CKE	SMP	-	-	SSPM[1:0]		0000 ..00	uuuu ..uu	***,*,***
SSPSTA	SSPBY	SSPOV	-	-	-	-	-	BF	00.. ...0	uu.. ...u	***,*,***
SSPBUF	SSP Receive/Transmit Buffer Register								xxxx xxxx	uuuu uuuu	***,***

表 21-2 SPI 寄存器

INTE0/INTE1/INTF1 : 详见 中断 章节

OSCCN0/OSCCN1/OSCCN2: 详见 錯誤! 找不到参照來源。 章节

SSPCN0: SPI 控制寄存器

位	名称	描述										
Bit7	ENSSP	启用与关闭控制器 <0> 关闭 <1> 启用										
Bit6	CKP	工作频率极性控制器 <0> 低电位为空闲 <1> 高电位为空闲 CKP=1 及 CKE=0 的模式下，必须将 SMP 设为 1 才能收到正确的数据。										
Bit5	CKE	数据发送控制器 <0> 当工作频率从空闲变成有效时发送 <1> 当工作频率从有效变成空闲时发送 CKP=1 及 CKE=0 的模式下，必须将 SMP 设为 1 才能收到正确的数据。										
Bit4	SMP	数据发送控制器 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>设定</th> <th>SPI 主机模式</th> <th>SPI 从机模式</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>在数据输出时间的中间采样输入数据</td> <td>被动模式，用户须将 SMP Bit 设置 <0></td> </tr> <tr> <td>1</td> <td>在数据输出时间的末端采样输入数据</td> <td>不可使用</td> </tr> </tbody> </table>	设定	SPI 主机模式	SPI 从机模式	0	在数据输出时间的中间采样输入数据	被动模式，用户须将 SMP Bit 设置 <0>	1	在数据输出时间的末端采样输入数据	不可使用	
设定	SPI 主机模式	SPI 从机模式										
0	在数据输出时间的中间采样输入数据	被动模式，用户须将 SMP Bit 设置 <0>										
1	在数据输出时间的末端采样输入数据	不可使用										
Bit1~0	SSPM[1:0]	模式选择字节 <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>设定</th> <th>操作模式</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>SPI 主机模式，频率源 = LS_CK</td> </tr> <tr> <td>01</td> <td>SPI 主机模式，频率源 = CPU_CK</td> </tr> <tr> <td>10</td> <td>SPI 从机模式，频率源 = SCK 引脚，SCE 引脚控制功能关闭，SCE 引脚组态为 I/O 使用</td> </tr> <tr> <td>11</td> <td>SPI 从机模式，频率源 = SCK 引脚，SCE 引脚控制功能启动 TAxTS[1:0]触发事件选择器</td> </tr> </tbody> </table>	设定	操作模式	00	SPI 主机模式，频率源 = LS_CK	01	SPI 主机模式，频率源 = CPU_CK	10	SPI 从机模式，频率源 = SCK 引脚，SCE 引脚控制功能关闭，SCE 引脚组态为 I/O 使用	11	SPI 从机模式，频率源 = SCK 引脚，SCE 引脚控制功能启动 TAxTS[1:0]触发事件选择器
设定	操作模式											
00	SPI 主机模式，频率源 = LS_CK											
01	SPI 主机模式，频率源 = CPU_CK											
10	SPI 从机模式，频率源 = SCK 引脚，SCE 引脚控制功能关闭，SCE 引脚组态为 I/O 使用											
11	SPI 从机模式，频率源 = SCK 引脚，SCE 引脚控制功能启动 TAxTS[1:0]触发事件选择器											

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



SSPSTA: SPI 控制暂存器

位	名称	描述									
Bit7	SSPBY	写入冲突位检测(仅供传送数据使用) <0> 未发生冲突 <1> 当数据仍然在传送状态									
Bit6	SSPOV	接收溢出旗标位 <table border="1"><thead><tr><th>设定</th><th>SPI 从机模式</th><th>SPI 主机模式</th></tr></thead><tbody><tr><td>0</td><td>未发生溢位状态</td><td>SSPOV 位将不会被设置为 1, 因每次传送(接收)数据都需要写入 SSPBUF 寄存器</td></tr><tr><td>1</td><td>SSPBUF 已经保持着上一笔数据, 而继续接收到一笔新数据, 一旦 SSPSR 溢出, 其 SSPSR 寄存器中的数据将会遗失。SSPOV 只发生在从机模式下, 即使只是发送数据, 用户也必须读取 SSPBUF 寄存器 以避免 SSPOV 被设置为 1.(须用指令清除)</td><td>不会发生</td></tr></tbody></table>	设定	SPI 从机模式	SPI 主机模式	0	未发生溢位状态	SSPOV 位将不会被设置为 1, 因每次传送(接收)数据都需要写入 SSPBUF 寄存器	1	SSPBUF 已经保持着上一笔数据, 而继续接收到一笔新数据, 一旦 SSPSR 溢出, 其 SSPSR 寄存器中的数据将会遗失。SSPOV 只发生在从机模式下, 即使只是发送数据, 用户也必须读取 SSPBUF 寄存器 以避免 SSPOV 被设置为 1.(须用指令清除)	不会发生
设定	SPI 从机模式	SPI 主机模式									
0	未发生溢位状态	SSPOV 位将不会被设置为 1, 因每次传送(接收)数据都需要写入 SSPBUF 寄存器									
1	SSPBUF 已经保持着上一笔数据, 而继续接收到一笔新数据, 一旦 SSPSR 溢出, 其 SSPSR 寄存器中的数据将会遗失。SSPOV 只发生在从机模式下, 即使只是发送数据, 用户也必须读取 SSPBUF 寄存器 以避免 SSPOV 被设置为 1.(须用指令清除)	不会发生									
Bit0	BF	缓冲器满状态位 (仅供接收数据使用) <0> 接收未完成, 接收缓冲寄存器为空 <1> 接收完成, 接收缓冲寄存器已满									

SSPBUF[7:0] 接收缓冲寄存器或是发送缓冲寄存器

22. 同步串行通讯接口, Inter-Integrated Circuit Serial interface

I²C 通讯接口包含主机(Master)与从机(Slave)两种运作模式, 主机模式可以根据系统的需求, 结合传送控制器(Transmission Controller, Tx Controller)传送 I²C 封包格式的信号至 I²C Bus, 并以 Clock Generator 决定所需的传输速度。而 Slave Controller 可以接收 I²C Bus 上的信号, 以从机模式接受 Bus 上的主机之通讯需求, 并结合传送控制器回传主机所需要的数据, 除此之外, Slave 控制器内含之数据接收电路也是 Master Controller 接收回传数据的信道。

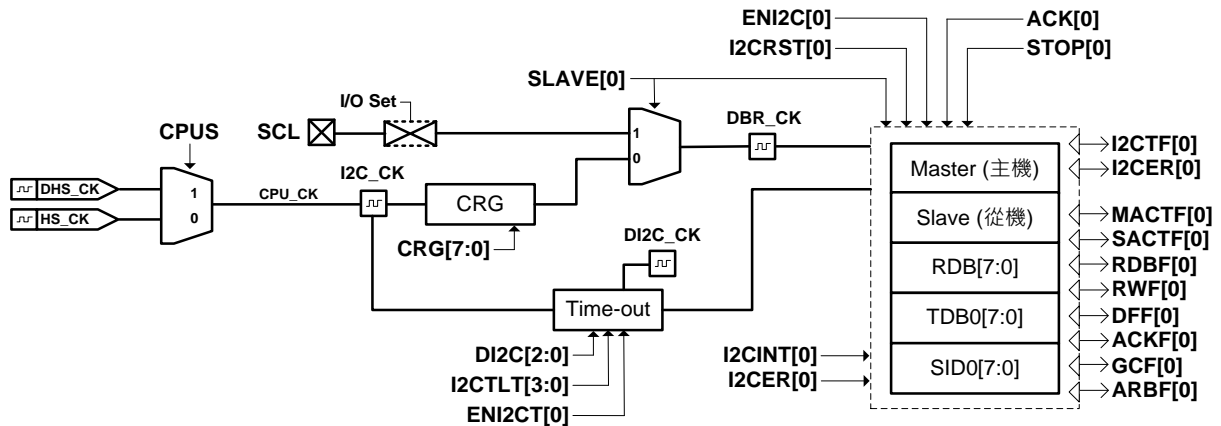


图 22-1 I²C 系统架构图

I²C 串行接口功能特性：

- 标准 I²C 串行接口包含 2 个接脚, 为串行数据(SDA)、串行频率(SCL)。
- 接脚为 Open Drain 输出结构, 需要外部提升电阻, 确保高电位输出。
- 标准 I²C 串行接口可配置为主机(Master)、从机(Slave)或主/从机模式。
- 可程序频率, 允许调整 I²C 传输速率。
- 主机和从机之间的数据传输为双向的。
- I²C 允许相当大的工作电压范围。
- I²C 的参考设计使用一个 7 位长度的地址空间但保留了 16 个地址, 所以在一组总线最多可和 112 个节点通讯。

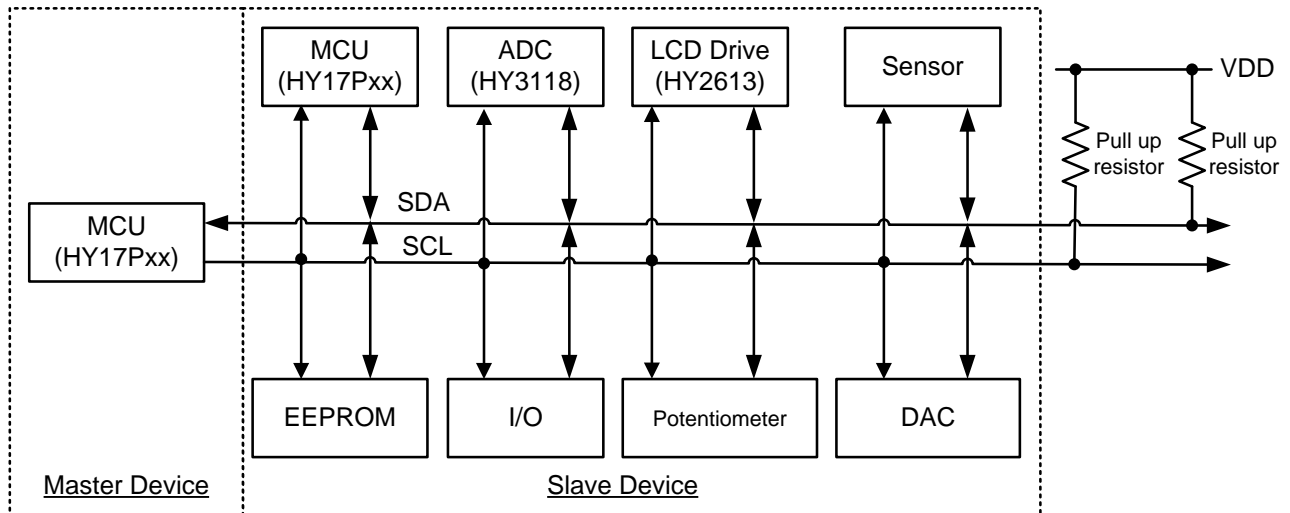


图 22-2 I²C 通讯接线示意图

I²C 串行接口信号：

- 起始信号(START)：主机 SCL 为高电位时，发出 SDA 由高电位转为低电位，开始数据传送。
- 数据(DATA)或地址(ADDRESS)信号：I2C 串行接口协议要求只有在 SCL 为低电位时，SDA 上数据才可以改变。
- 应答信号(Acknowledge)：接收数据的装置(从机)在接收到第 8 位后，向发送数据的装置(主机)发送低电位，表示已收到资料。
- 停止信号(STOP)：主机 SCL 为高电位时，发出 SDA 由低电位转为高电位，结束数据传送。

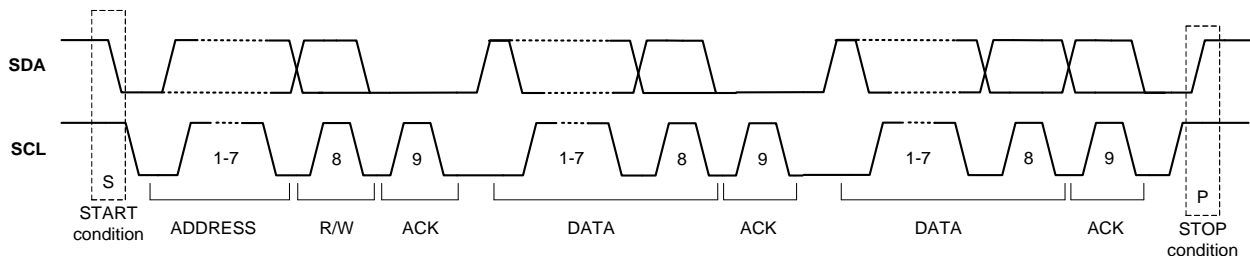


图 22-3 I²C 总线时序图

22.1. 数据传输速率计算

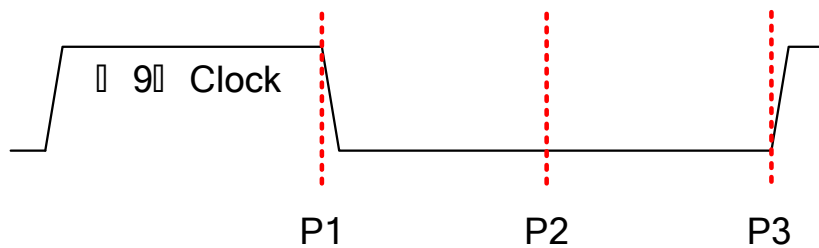
- Master Mode (主机模式)

I²C 内部寄存器 CRG[7:0]可以控制主机模式传送数据的速度，CRG[7:0]的数值经由内部计数器产生主机的 SCL 的引脚信号，所以数据传送率可以根据 I²C 的频率源 I2C_CK 的频率，利用下列公式计算：

$$\text{DataBaudRate(Hz)} = \frac{\text{I2C_CK}}{[4 \times (\text{CRG}[7:0] + 1)]}$$

- Slave Mode (从机模式)

当 Master 端使用的是标准硬件 I²C 或者是会判断 SCL 状态才做动的装置时，CRG[7:0]建议填值为 01H。如果 Master 端是使用 I/O 仿真而成的装置，如需配合第九个 Clock 长度可依实际情况调整。公式计算为下



P1~P2 0 0 0 0 0 ISR 0 0 0 I2CINT 0
P2~P3 0 0 0 CRG[7:0] 0
Min=(CRG+1)*(CPU_CK 0 0)
Max=2*(CRG+1)*(CPU_CK 0 0)

22.2. 超时计时功能(Time-Out)

- Time-out 控制是为了避免 I2C 控制器将 I2C 通讯总线死锁，I2C 在操作的过程中为了提供 MCU 足够的时间处理 I2C 控制器的需求，因此 I2C 控制器在每一个响应位之后都会将 SCL 拉为 Low，使 Master 无法传出下一个频率信号，即是发生通讯延时 (Clock Stretching)。但是当 MCU 过于忙碌或任何原因无法响应 I2C 控制器的需求时，I2C 通讯总线的 SCL 将有可能被死锁于 Low。
- 为了避免上述情况发生，Time-out 控制器可以根据用户透过工作频率分频器 DI2C[2:0]及时间条件控制器 I2CTLT[3:0]，决定 SCL 处于 Low 状态的 Time-out 条件。条件处理有以下状态：
- 当侦测 SCL 被本机拉为 Low 的时间，满足条件后 I2C 控制器会强制将 SCL 放掉并且发出中断事件至 CPU。
- 当 SCL 在未达到 Time-out 时间被释放为 High，则 Time-out 控制器内部的计数器将被重置，并于下次 SCL 再被拉为 Low 时重新计数。

22.3. I²C 串行接口通讯流程图

I²C 串行接口术语

- (SPIA) : 代表对动作寄存器(ACT)所下达指令, S 为 Start 指令, P 为 Stop 指令, I 为中断标志, A 为 Acknowledge 应答指令。
- SPIA : 代表读取动作寄存器(ACT)之值, 可以用于判读中断标志或其它指令是否运作完成。
- STA : 读取 Status 寄存器(STA)之值, 用以表示目前 I²C 电路运作状态。
- 下列流程图会以图 16-4 所示之“灰底圆框”、“白底圆框”、“方框”分别表示 I²C 界面之状态 :
 - 灰底圆框 : 表示中断标志已被设立之 I²C 状态。
 - 白底圆框 : 表示中断标志未被设立, 需由 MCU 主动读取之 I²C 状态。
 - 方框 : 表示需由 MCU 对 I²C 下达指令。

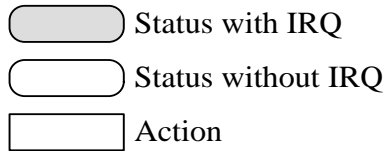


图 22-4 流程图符号

HY17S58 Emulate Chip User' Guide

Embedded ΣADC
8-Bit RISC-like Mixed Signal Microcontroller

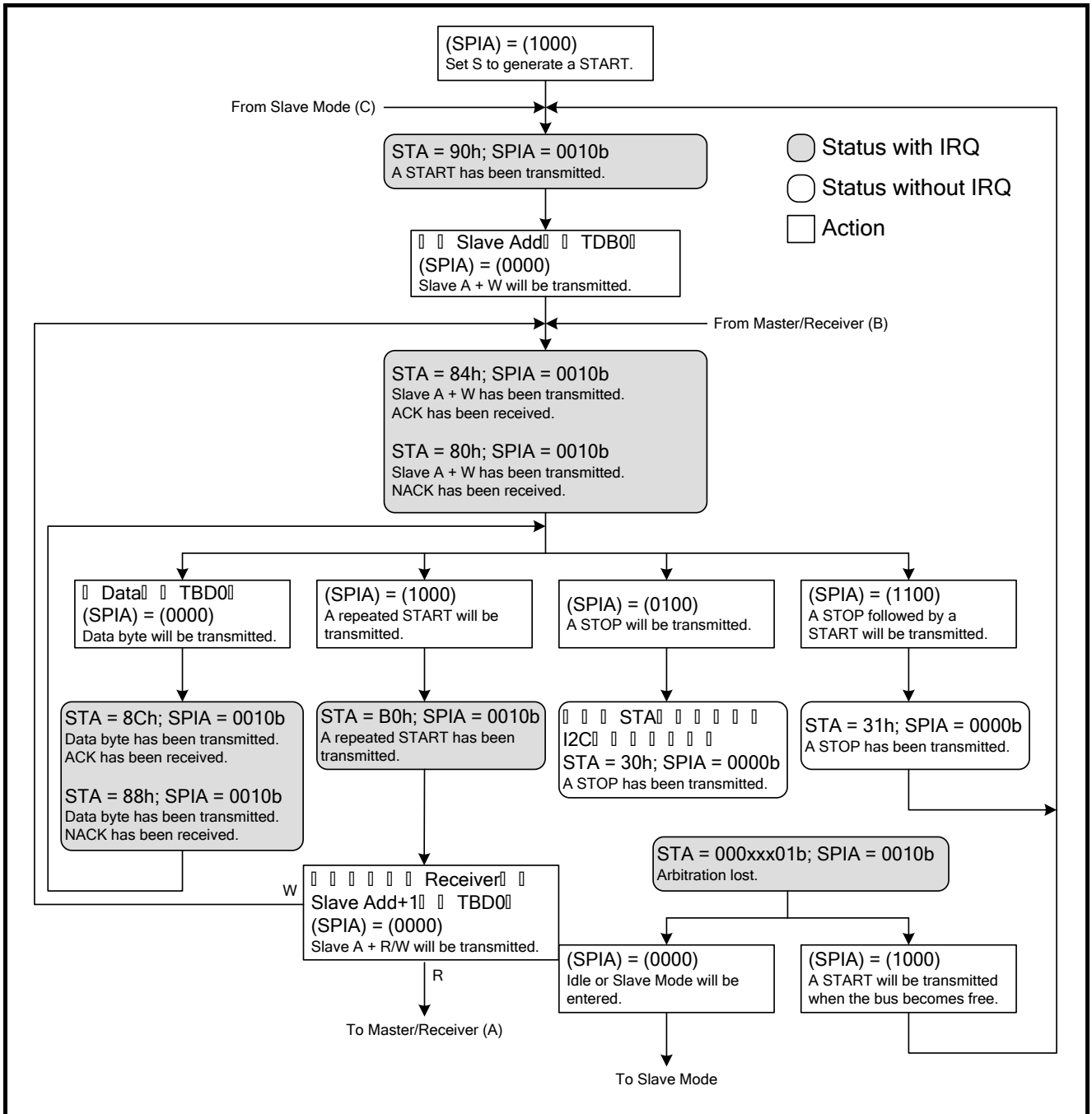
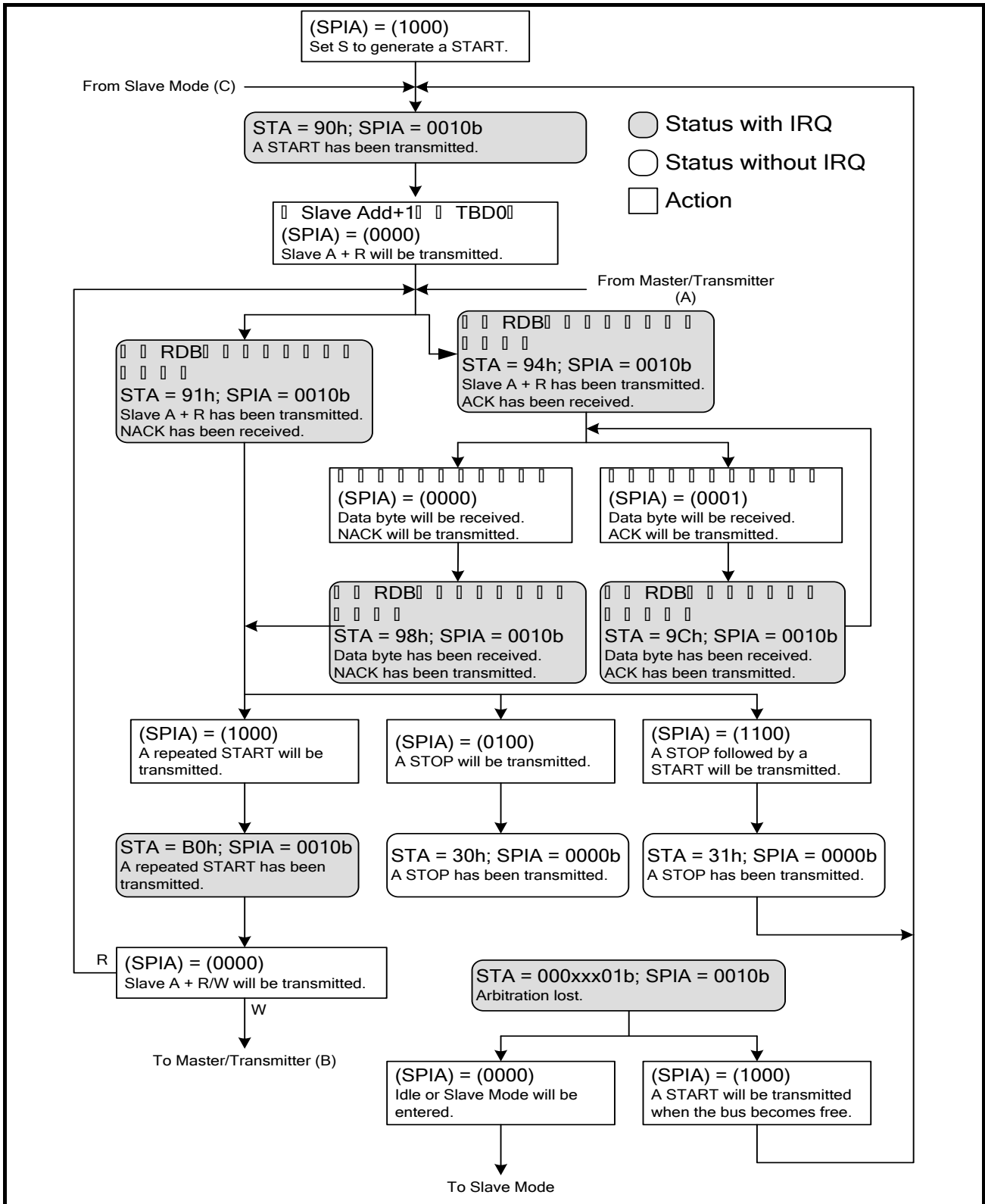


图 22-5 Master Transmitter Mode

HY17S58 Emulate Chip User' Guide

Embedded ΣADC
8-Bit RISC-like Mixed Signal Microcontroller



HY17S58 Emulate Chip User' Guide

Embedded ΣΔADC
8-Bit RISC-like Mixed Signal Microcontroller

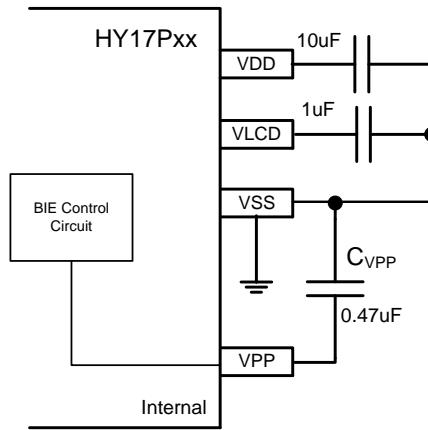


图 22-6 Master Receiver Mode

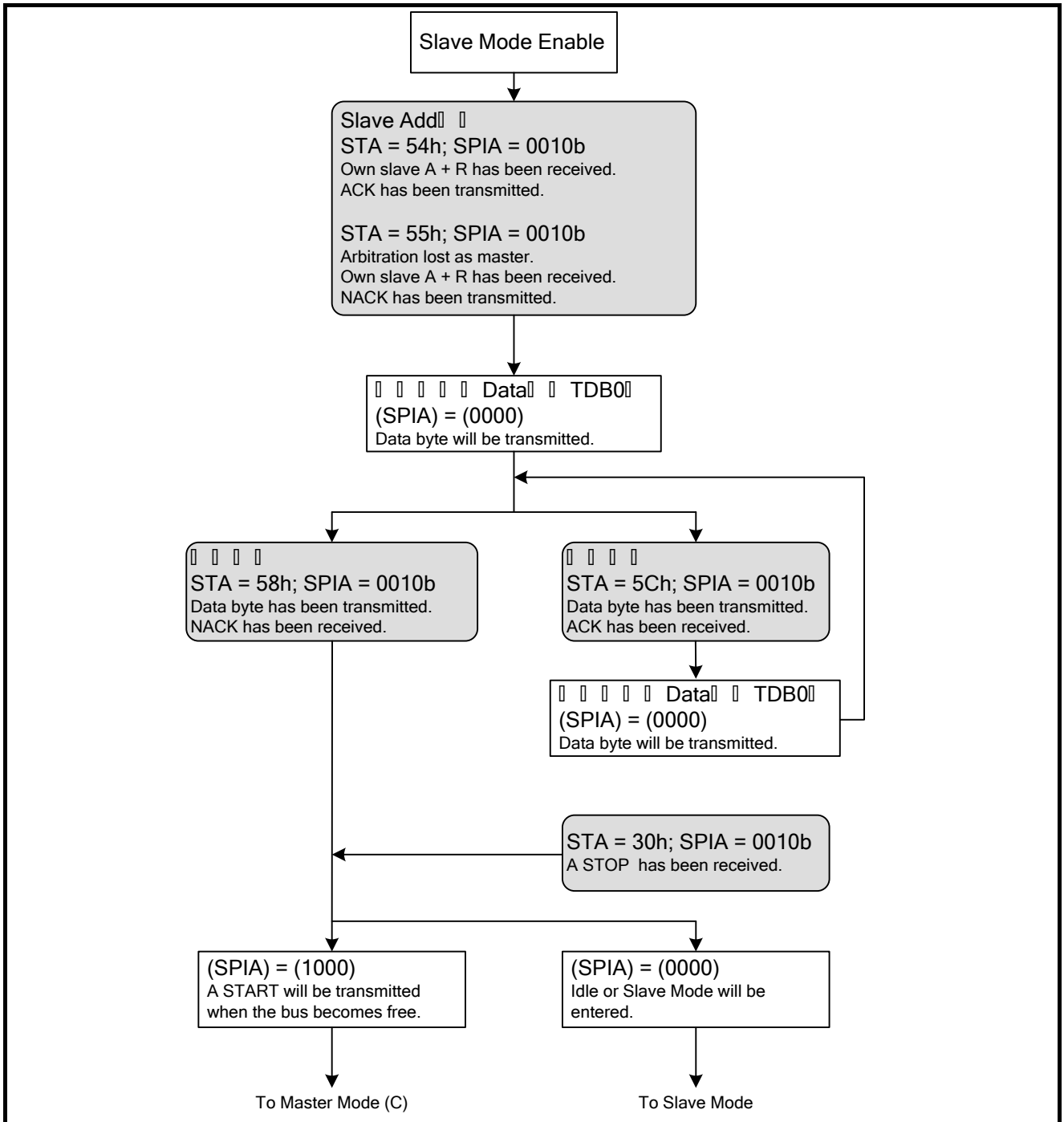


图 22 -7 Slave Transmitter Mode

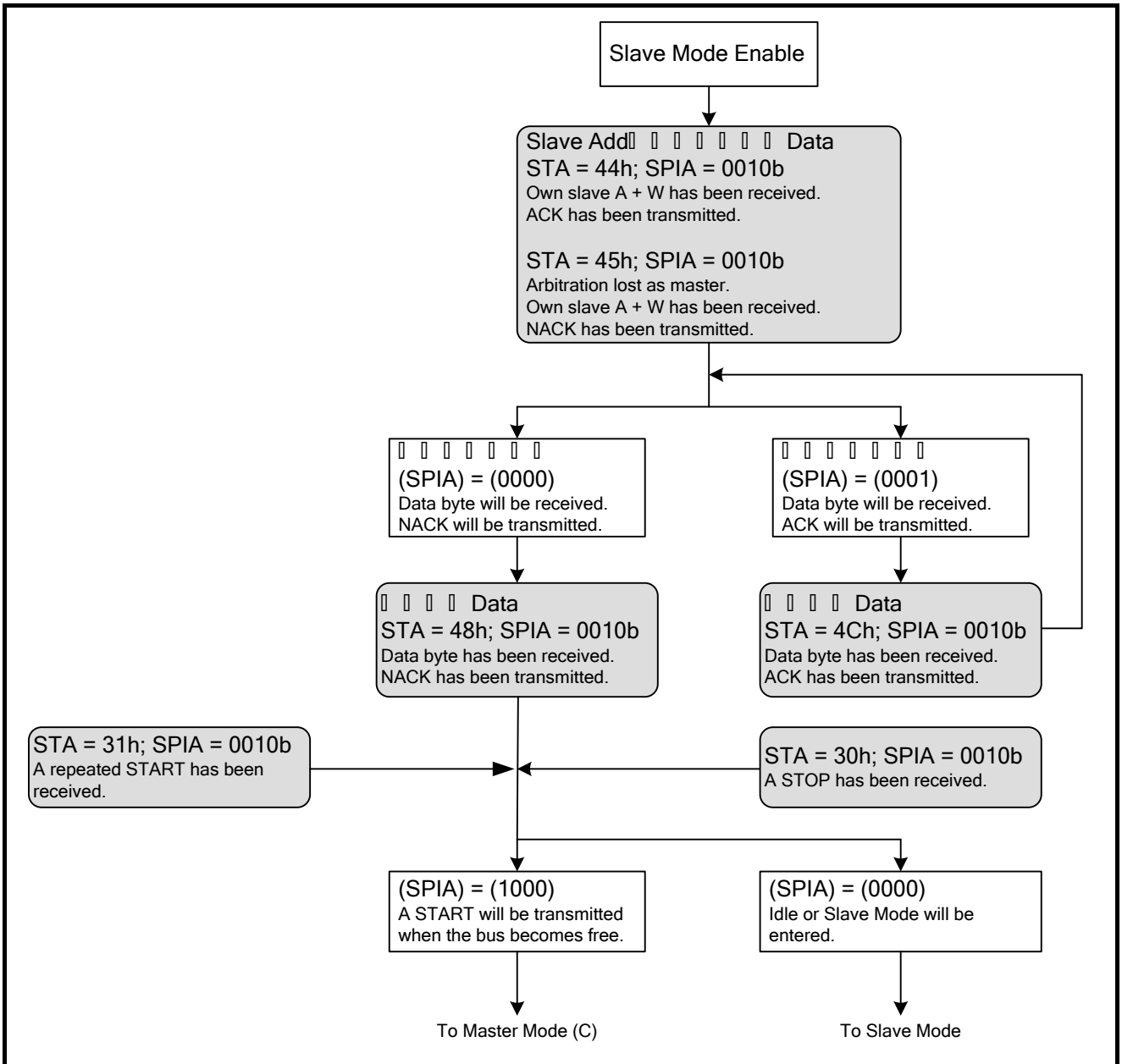


图 22-8 Slave Receiver Mode

HY17S58 Emulate Chip User' Guide

Embedded ΣADC
8-Bit RISC-like Mixed Signal Microcontroller

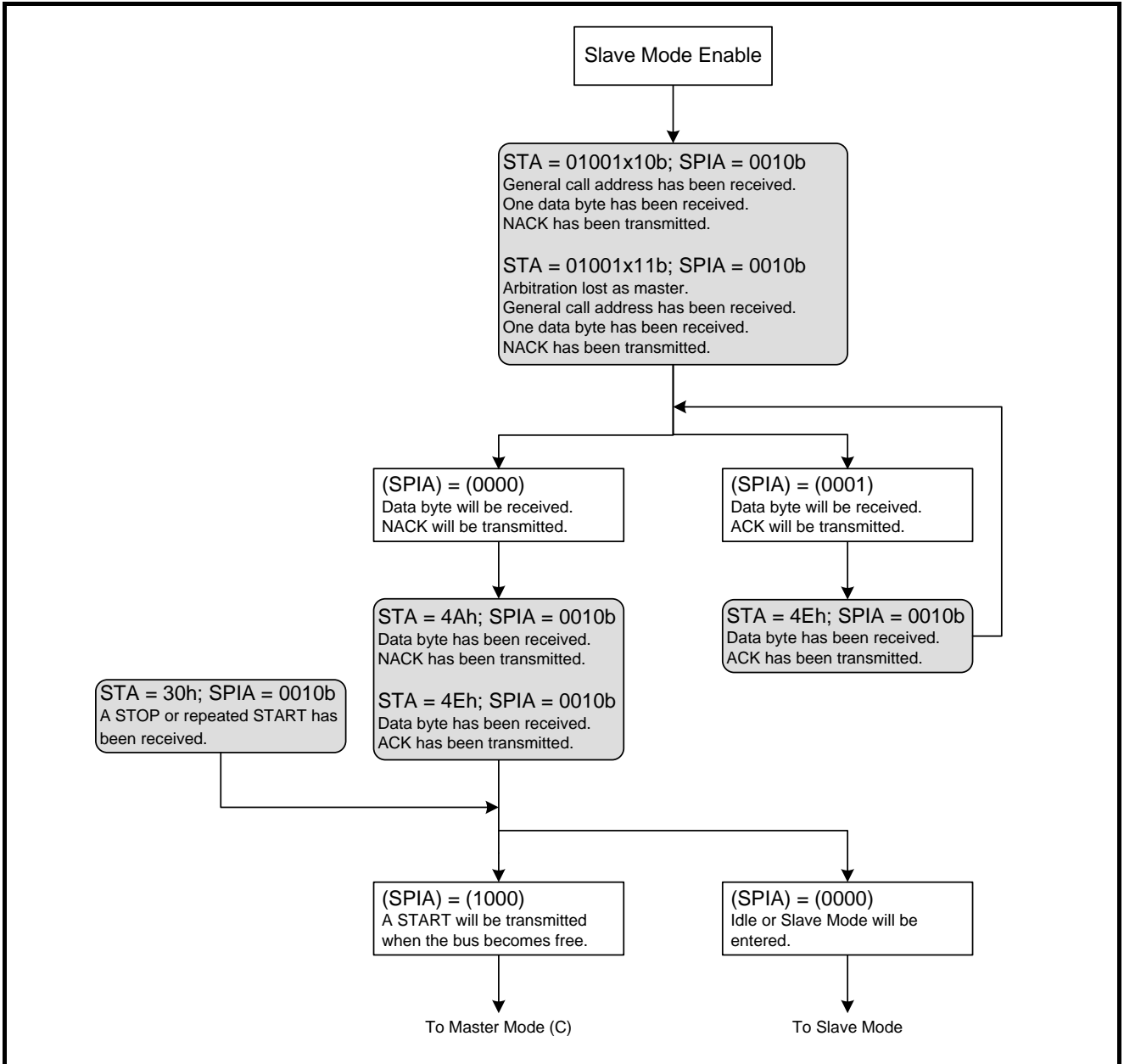


图 22-9 General Call Mode

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



22.4. I²C 寄存器说明

“-”no use,“*”read/write,“w”write,“r”read,“r0”only read 0,“r1”only read 1,“w0”only write 0,“w1”only write 1											
“\$”for event status,“-”unimplemented bit,“x”unknown,“u”unchanged,“d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	-	ADIE	WDTIE	TB1IE	TMAIE	E1IE	EOIE	0000 0000	0uuu uuuu	*****
INTE1	-				I2CERIE	I2CIE			0000 0000	uuuu uuuu	*****
INTF1	-				I2CERIF	I2CIF			0000 0000	uuuu uuuu	*****
CFG	Rsv.					GCRst	ENI2CT	ENI2C000uuu	***r***
ACT	SLAVE	-	-	I2CER	START	STOP	I2CINT	ACK	0000 0000	uuuu uuuu	*****
STA	MACTF	SACTF	RDBF	RWF	DFE	ACKF	GCF	ARBF	0001 0000	uuuu uuuu	*****
CRG	CRG[7:0]								0000 0000	uuuu uuuu	*****
TOC	I2CTF	DI2C[2:0]			I2CTLT[3:0]				0000 0000	uuuu uuuu	*****
RDB	RDB[7:1]							RDB[0]	xxxx xxxx	uuuu uuuu	*****
TDB0	TDB0[7:1]							TDB[0]	xxxx xxxx	uuuu uuuu	*****
SID0	SID[7:1],The corresponding address of the 7-bit mode							SIDV[0]	0000 0000	uuuu uuuu	*****
TRISC2	TC2.7	TC2.6	TC2.5	TC2.4	TC2.3	TC2.2	TC2.1	TC2.0	0000 0000	uuuu uuuu	*****
PT2M1	-	PM2.3[0]	-	-	-	-	-	-	0000 0000	uuuu uuuu	*****

表 22-1 I²C 寄存器

INTE0/INTE1/INTF1 : 详见 中断 章节

CFGx : I²C 设置寄存器(Configuration Register)

位	名称	描述
Bit2	GCRst	I ² C 全呼复位使能控制 <0>关闭 <1>开启 当 I ² C Slave 模式与 GCRst 功能同时被开启时 如果 I ² C Controller 接收 General call ID 00h 并且第一笔资料为"06h"即为 General Call Reset 条件成立,此时原本会发送至本机处理器的中断信号(Interrupt)将被重置信号(Reset)取代,提供外部主机可以经由 I2C Bus 重置本机芯片之功能。
Bit1	ENI2CT	开启 I ² C 超时监控功能位 0 : 关闭 <1>开启 I ² C Time-out 监控功能
Bit0	ENI2C	开启 I ² C 功能控制位 <0>关闭 <1>开启 I ² C 通讯接口 ※ 注意事项: 当 ENI2C 关闭时,将关闭 I ² C 内部的 Clock,除了 Configuration Register 可以进行写入动作,其余寄存器将无法写入数据。

ACTx: 动作寄存器(Action Register)

位	名称	描述
Bit7	SLAVE	从机开启控制 <0> 关闭 <1> 开启

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

位	名称	描述
Bit4	I2CER	错误中断标志 <0> 正常, 写 0 将会清除错误中断标志, 使 I ² C 往下一个状态执行。 <1> 发生错误中断
Bit3	START	开始命令位 <0> 正常 <1> 于 I ² C Bus 产生 Start 信号
Bit2	STOP	停止命令位 <0> 正常 <1> 于 I ² C Bus 产生 Stop 信号
Bit1	I2CINT	中断标志 <0> 正常, 写 0 将会清除中断标志, 使 I ² C 往下一个状态执行。 <1> 发生 I ² C 中断
Bit0	ACK	ACK(Acknowledge)应答控制位 <0> 未回复 ACK 或回复 NACK <1> ACK 已回复

STAx: I²C 状态寄存器

位	名称	描述
Bit7	MACTF	主机模式启用标志(Master Mode Active Flag) <0> 未启用 <1> 启用
Bit6	SACTF	主机模式启用标志(Slave Mode Active Flag) <0> 未启用 <1> 启用
Bit5	RDBF	接收停止或重新开始标志(Received Stop/Repeat-Start Flag) <0> 正常 <1> 接收停止或重新开始标志已被发送或接收。
Bit4	RWF	读写状态标志(Read/Write State Flag) <0> 写命令已被发送或接收。 <1> 读命令已被发送或接收。
Bit3	DFB	资料旗标(Data Field Flag) <0> 正常 <1> I ² C 数据被发送或接收。
Bit2	ACKF	ACK 应答标志(Acknowledge Flag) <0> ACK 未发送或接收。 <1> ACK 已发送或接收。

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



位	名称	描述
Bit1	GCF	General Call Flag <0> 正常 <1> Currently General Call Operation
Bit0	ARBF	仲裁漏失标志标(Arbitration Lost Flag) <0> 正常 <1> 仲裁漏失

CRGx: I²C 频率控制寄存器

位	名称	描述
Bit7~0	CRGx[7:0]	I ² C Bus Data Baud Rate Control I ² C Bus 上的数据传送是以 SCL 引脚上的频率信号所决定，而 SCL 引脚上的频率率可以由 I ² C 电路之频率源的频率 CPU_CK 与 CRG 经由下列公式计算： $\text{Data Baud Rate(Hz)} = \frac{\text{I2C_CK}}{[4 \times (\text{CRG}[7:0] + 1)]}$

TOCx: I²C 超时控制寄存器

位	名称	描述																																				
Bit7	I2CTF	超时标志 <1> I ² C Bus Clock Stretching Time-out <0> Normal																																				
Bit6~4	DI2C[2:0]	Time-out Clock Pre-scale <table border="1"> <thead> <tr> <th>DI2C[2:0]</th> <th>Pre-scale</th> <th>DI2C[2:0]</th> <th>Pre-scale</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>CLKPS = CPU_CK / 1</td> <td>100</td> <td>CLKPS = CPU_CK / 16</td> </tr> <tr> <td>001</td> <td>CLKPS = CPU_CK / 2</td> <td>101</td> <td>CLKPS = CPU_CK / 32</td> </tr> <tr> <td>010</td> <td>CLKPS = CPU_CK / 4</td> <td>110</td> <td>CLKPS = CPU_CK / 64</td> </tr> <tr> <td>011</td> <td>CLKPS = CPU_CK / 8</td> <td>111</td> <td>CLKPS = CPU_CK / 128</td> </tr> </tbody> </table>	DI2C[2:0]	Pre-scale	DI2C[2:0]	Pre-scale	000	CLKPS = CPU_CK / 1	100	CLKPS = CPU_CK / 16	001	CLKPS = CPU_CK / 2	101	CLKPS = CPU_CK / 32	010	CLKPS = CPU_CK / 4	110	CLKPS = CPU_CK / 64	011	CLKPS = CPU_CK / 8	111	CLKPS = CPU_CK / 128																
DI2C[2:0]	Pre-scale	DI2C[2:0]	Pre-scale																																			
000	CLKPS = CPU_CK / 1	100	CLKPS = CPU_CK / 16																																			
001	CLKPS = CPU_CK / 2	101	CLKPS = CPU_CK / 32																																			
010	CLKPS = CPU_CK / 4	110	CLKPS = CPU_CK / 64																																			
011	CLKPS = CPU_CK / 8	111	CLKPS = CPU_CK / 128																																			
Bit3~0	I2CTLTLT[3:0]	Time-out Limit ; Time-out 的发生是以 CLKPS 计数 I2CTLTLT + 1 次后触发 <table border="1"> <thead> <tr> <th>I2CTLTLT[3:0]</th> <th>Limit</th> <th>I2CTLTLT[3:0]</th> <th>Limit</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>1 x CLKPS Cycle</td> <td>1000</td> <td>9 x CLKPS Cycle</td> </tr> <tr> <td>0001</td> <td>2 x CLKPS Cycle</td> <td>1001</td> <td>10 x CLKPS Cycle</td> </tr> <tr> <td>0010</td> <td>3 x CLKPS Cycle</td> <td>1010</td> <td>11 x CLKPS Cycle</td> </tr> <tr> <td>0011</td> <td>4 x CLKPS Cycle</td> <td>1011</td> <td>12 x CLKPS Cycle</td> </tr> <tr> <td>0100</td> <td>5 x CLKPS Cycle</td> <td>1100</td> <td>13 x CLKPS Cycle</td> </tr> <tr> <td>0101</td> <td>6 x CLKPS Cycle</td> <td>1101</td> <td>14 x CLKPS Cycle</td> </tr> <tr> <td>0110</td> <td>7 x CLKPS Cycle</td> <td>1110</td> <td>15 x CLKPS Cycle</td> </tr> <tr> <td>0111</td> <td>8 x CLKPS Cycle</td> <td>1111</td> <td>16 x CLKPS Cycle</td> </tr> </tbody> </table>	I2CTLTLT[3:0]	Limit	I2CTLTLT[3:0]	Limit	0000	1 x CLKPS Cycle	1000	9 x CLKPS Cycle	0001	2 x CLKPS Cycle	1001	10 x CLKPS Cycle	0010	3 x CLKPS Cycle	1010	11 x CLKPS Cycle	0011	4 x CLKPS Cycle	1011	12 x CLKPS Cycle	0100	5 x CLKPS Cycle	1100	13 x CLKPS Cycle	0101	6 x CLKPS Cycle	1101	14 x CLKPS Cycle	0110	7 x CLKPS Cycle	1110	15 x CLKPS Cycle	0111	8 x CLKPS Cycle	1111	16 x CLKPS Cycle
I2CTLTLT[3:0]	Limit	I2CTLTLT[3:0]	Limit																																			
0000	1 x CLKPS Cycle	1000	9 x CLKPS Cycle																																			
0001	2 x CLKPS Cycle	1001	10 x CLKPS Cycle																																			
0010	3 x CLKPS Cycle	1010	11 x CLKPS Cycle																																			
0011	4 x CLKPS Cycle	1011	12 x CLKPS Cycle																																			
0100	5 x CLKPS Cycle	1100	13 x CLKPS Cycle																																			
0101	6 x CLKPS Cycle	1101	14 x CLKPS Cycle																																			
0110	7 x CLKPS Cycle	1110	15 x CLKPS Cycle																																			
0111	8 x CLKPS Cycle	1111	16 x CLKPS Cycle																																			

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

RDBx: 接收数据寄存器

位	名称	描述
Bit7~1	RDBx[7:1]	内容为接收地址(A7~A1)或数据(D7~D1)
Bit0	RDBx[0]	内容为接收读写命令或数据(D0)

TDBx: 传送数据寄存器

位	名称	描述
Bit7~1	TDBx[7:1]	内容为传送地址(A7~A1)或数据(D7~D1)
Bit0	TDBx[0]	内容为传送读写命令或数据(D0)

※ 注意事项: 在通讯过程中当本机属于非传送 Address 或 Data 的状态时必须将此寄存器设为 FFh, 因为 TDB0 的 Bit 7 为 0 有可能将 SDA Bus 死锁于 Low。

SID0: 从机模式 ID (从机地址) 码设置寄存器

位	名称	描述
Bit7~1	SID[7:1]	从机 ID 码(A7~A1)
Bit0	SIDV[0]	从机 ID 码有效控制 <0> 从机 ID 码无效 <1> 从机 ID 码有效

23. 异步串行通讯接口, Enhanced Universal Asynchronous Receiver

Transmitter

增强型异步收发器 Enhanced Universal Asynchronous Receiver Transmitter, EUART 外围通常也称为串行通信接口或 SCI, EUART 可以被配置为能与 CRT 终端和个人计算机等外设通信的全双工异步系统;也可以被配置成能够与 A/D 或 D/A 集成电路、串行 EEPROM 等外设通信的半双工同步系统。

增强型 EUART 是在标准型 UART 的基础上增加了帧(Frame)错误检测和自动地址识别两种功能, 帧错误检测通过检测一帧信息的停止位来判断该帧是否有效。自动地址识别功能自动将收到的地址帧内容与单片机的地址进行比较, 只有匹配时才产生串行中断。此版的这两个功能前者由内置硬件电路完成后者由用户软件达成。

EUART 可以配置为以下几种工作及侦错模式:

- 带有以下功能的全双工异步模式:
 - 传输波特率发生器
- 侦错模式
 - 帧(Frame)错误检测³
 - 溢出 (Overrun) 错误检测⁴
 - 硬件同位检查码
- 数据传输与接收
 - 异步发送(8 位或 9 位)
 - 异步接收(8 位或 9 位)
- 字符接收自动唤醒功能

EUART 寄存器摘要:

URxCN	ENSP[0], ENTX[0], TX9[0], TX9D[0], PARITY[1:0]
URxSTA	RC9D [0], PERR[0], FERR[0], OERR[0], RCIDL[0], TRMT[0], ABDOVF[0]
BAxCN	ENCR[0], RC9[0], ENADD[0], ENABD[0]
BRxGR[15:0]	BGxRH[7:0], BGxRL[7:0]
TXxR	TXxR[7:0]
RCxREG	RCxREG[7:0]

³ 帧错误检测(FERR): UART 没有收到起始位, 亦即收到不知从何时开始到何时结束的信号时称之; 这通常由信号在线的噪声引起, 会使 UART 无法在移位寄存器上得到正确数据。

⁴ 溢出错误检测(OERR): 最近的一笔数据已覆盖先前未取走的数据。

23.1. EUART 使用说明

23.1.1. 异步数据发送设置步骤(x=0 or 2)

- 将 ENSP 置为 1 后, IC 会将相对应的 I/O 口自动切换为 UART 引脚。
- 配置 INTE_x 寄存器 TXIE 位及 INTE0 寄存器 GIE 位决定是否允许传送中断致能。(INTF_x 寄存器 TXIF 位默认为 High, 相关中断使能须确定后再设定)。
- 配置 BG_xRH, BG_xRL 寄存器, 决定合适的波特率值。
- 配置 URXC_N 寄存器 ENSP 位启动 EUART 串行 I/O 模块。
- 配置 URXC_N 寄存器 TX9 位决定是否启动第 9 位数据发送功能。(若启动第 9 位数据发送功能, 则须将该数据填入 TX9D 位中。第 9 位位可以是地址或是数据)。
- 配置 URXC_N 寄存器 ENTX 位启动数据传送功能。
- 写入 TX_xR 寄存器, 决定传送数据。(写入后启动发送)

23.1.2. 异步数据接收设置步骤(x=0 or 2)

- 将 ENSP 置为 1 后, IC 会将相对应的 I/O 口自动切换为 UART 引脚。
- 配置 INTE1, 寄存器 RCIE 位及 INTE0 寄存器 GIE 位决定是否允许接收中断致能。
- 配置 BG_xRH, BG_xRL 寄存器, 决定合适的波特率值。
- 配置 UR_xC_N 寄存器 ENSP 位启动 EUART 串行 I/O 模块。
- 配置 BA_xC_N 寄存器 RC9 位决定是否启动第 9 位数据接收功能。
- 配置 BA_xC_N 寄存器 ENCR 位启动数据接收功能。
- 读取 UR_xSTA 寄存器 RC9D 位用以撷取接收数据第 9 位数据(RC9 设置情形下), 并判断接收过程是否发生错误。
- 读取 RC_xREG 寄存器用以撷取接收数据共 8 位数据。
- 读取 UR_xSTA 寄存器 FERR 位是否被设置, 确定读取数据是否错误, 可透过清除 ENCR 位解除 FERR 位。

23.1.3. 异步数据接收(9 位, RS-485 模式) 设置步骤(x=0 or 2)

- 将 ENSP 置为 1 后, IC 会将相对应的 I/O 口自动切换为 UART 引脚。
- 配置 BG_xRH, BG_xRL 寄存器, 决定合适的波特率值。
- 配置 UR_xC_N 寄存器 ENSP 位启动 EUART 串行 I/O 模块。
- 配置 BA_xC_N 寄存器 RC9 位否启动第 9 位数据接收功能。
- 配置 BA_xC_N 寄存器 ENADD 位使能地址检测功能。
- 配置 BA_xC_N 寄存器 ENCR 位启动数据接收功能。
- 配置 INTE_x, 寄存器 RCIE 位及 INTE0 寄存器 GIE 位决定是否允许接收中断致能。当数据接收完成时 RCIF 位将被设置。
- 读取 UR_xSTA 寄存器 RC9D 位用以撷取接收数据第 9 位数据(RC9 设置情形下), 并判断接收过程是否发生错误。
- 读取 RC_xREG 寄存器用以撷取接收数据共 8 位数据。

- 读取 URxSTA 寄存器 FERR 位是否被设置, 确定读取数据是否错误, 可透过清除 ENCR 位解除 FERR 位。
- 配置 BxCN 寄存器 ENADD 位用以关闭地址检测, 使得下一笔数据接收。

23.2. 串行传输波特率发生器 (BRG)

BRG 是一个专用的 13 位发生器, 支持 EUART 的异步模式。BGxR[15:0]寄存器是一个独立运行定时器的周期控制器。表 23-1 为串行传输波特率计算公式, 但仅适用于主控模式。

在给定目标串行传输波特率且工作频率为 CPU_CK 的情况下, 可以使用表 23-1 中的公式计算 BGxR[15:0]寄存器中的近似整数值, 从而确定串行传输波特率误差。范例 21-1 描述出串行传输波特率与误差率的计算方式。

BRG/EUART MODE	串行传输波特率计算公式
13 位/异步	$CPU_CK \div [4(n + 1)]$
CPU_CK = 工作频率; n = BGxRH:BGxRL 寄存器对的值	

表 23-1 串行传输波特率公式

工作在异步模式下, 其工作频率为 CPU_CK (2MHz), 而目标串行传输波特率为 9600bps。
 求解 $BGxR[15:0] = \langle ? \rangle$ 即 $BGxRH[7:0]:BGxRL[7:0] = \langle ? \rangle$
 已知公式: 目标串行传输波特率 = $CPU_CK \div (4 \times (BRGR[15:0] + 1))$:
 故 $BGxR[15:0] = ((CPU_CK \div \text{目标串行传输波特率}) \div 4) - 1$
 $= ((2000000 \div 9600) \div 4) - 1$
 $= 51.08$
 ≈ 51 即 $BGxRH[7:0] = \langle 00 \rangle$, $BGxRL[7:0] = \langle 33 \rangle$; 注 33 为 16 进制
 而实际上 BRG 计算结果为: 实际串行传输波特率 = $2000000 \div (4 \times (51 + 1)) = 9615.38$
 所以存在一定误差, 其计算方式为:
 误差率 = $(\text{实际串行传输波特率} - \text{目标串行传输波特率}) / \text{目标串行传输波特率}$
 $= (9615 - 9600) / 9600$
 $= 0.16\%$

范例 23-1 计算串行传输波特率误差

23.2.1. 功耗管理模式下的操作

芯片频率用于产生所需的串行传输波特率。当进入一种功耗管理模式时, 新频率源可能会工作在一个不同的频率下。这可能需要调整 BGxR[15:0]寄存器中的值。

23.2.2. RC 取样方式

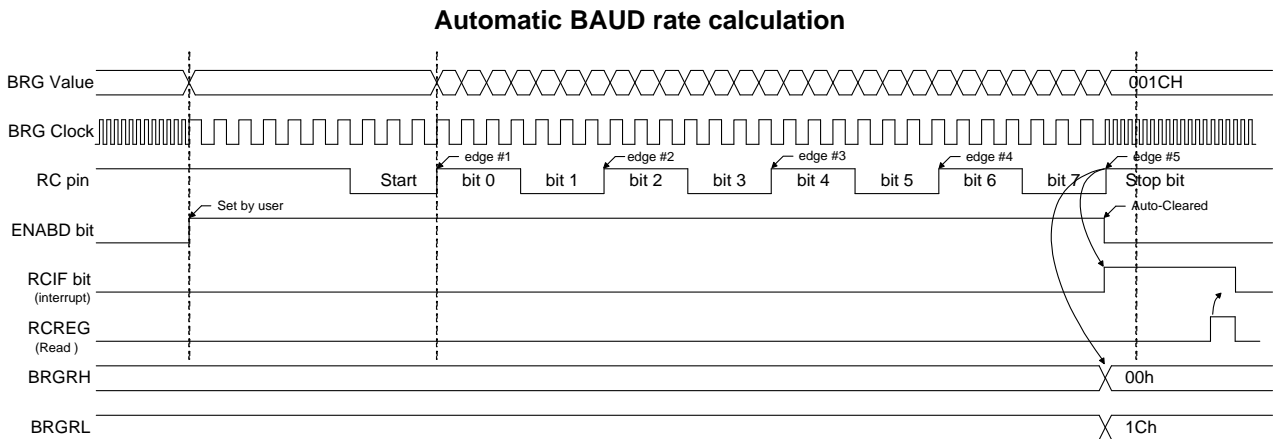
取样电路会在传输波特率周期的中心点进行取样, 以判定 RC 引脚上出现的是高电位还是低电位。

23.2.3. 自动波特率

EUART 模块支持自动检测和校准功能 亦可称之为自动波特率。自动波特率必须在唤醒启用控制器 WUE[0] 设置 0 时才有效，并将自动波特率启用控制器 ENABD[0]设置 1 即可启用。

在接收到开始状态后，即开始进行自动波特率检测功能(接收值必须为 055H，且不能是偶同位检察)。在自动检测和校准完成后会将计算出了结果写入 BGxRH[7:0]与 BGxRL[7:0]，相关时序如图 23-1。

当 BGxR[15:0]计算满出时即其内容由 01FFFH 到 00000H 产生溢位，则自动波特率溢位标志 ABDOVF[0] 会被置 1，用户可以利用指令将 ABDOVF[0]设置 0 或透过 ENABD[0]设置 0 以让 ABDOVF[0]置 0。在 ABDOVF[0]置 1 后，ENABD[0]的状态仍会保持在 1，相关时序如图 23-2。



Note : The ABD sequence requires the EUART module to be configured in WUE = 0

图 23-1 自动波特率计算波形

BRG Overflow Sequence

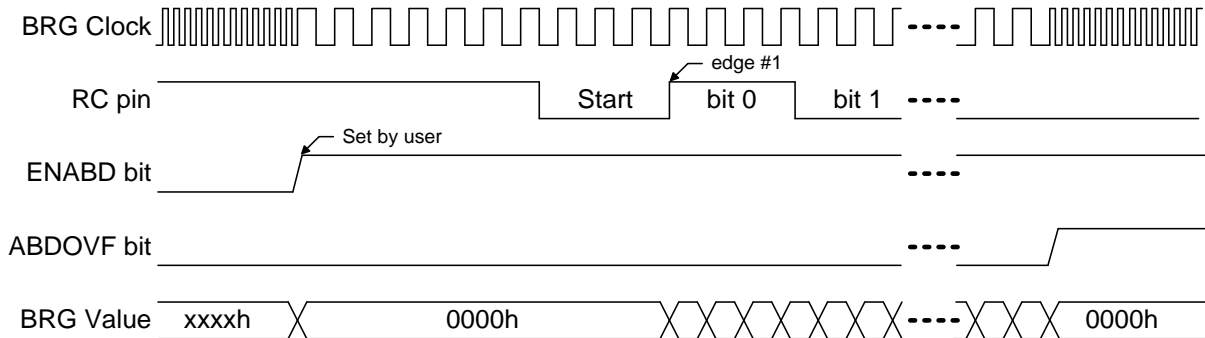


图 23-2 自动波特率溢位(ABDOVF)波形

23.3. 硬件同位检查

EUART 支持硬件奇偶校验功能，校验位保存在第 9 个数据位中。使用时依用户寄存器设定而进行同位检查(ENADD[0]=0)，相关设定如表 23-2

传送/接收 8/9 位数据		PARITY	状态
TX9	RC9		
0	0	0	传送/接收数据无同位检查讯息
0	0	1	传送/接收数据无同位检查讯息
0	1	0	接收数据具同位检查功能，偶同位
0	1	1	接收数据具同位检查功能，奇同位
1	0	0	传送数据具同位检查码，偶同位
1	0	1	传送数据具同位检查码，奇同位
1	1	0	
1	1	1	

注：当 RC9[0]设置 1 时，启用同位检查功能而当奇偶同位错误时 PERR[0]置 1。
假使 RC9[0]与 ENADD[0]同时设置 1，则忽略 PERR[0]错误位的值。

表 23-2 同位检查状态表

23.4. EUART 异步模式

此模式使用标准的“不归零”（Non-Return-to-Zero, NRZ）格式，即是 1 个起始位加上 8 个或 9 个数据位最后为 1 个停止位，最常用的数据格式为 8 位。而芯片上专用 13 位串行传输波特率发生器，可借助于工作频率振荡器产生标准串行传输波特率频率。

再者，EUART 首先发送和接收的数据是最低有效位，发送器和接收器在功能上是各自独立的，但采用相同的数据格式和串行传输波特率。其更支持硬件奇偶校验功能，校验位会保存在第 9 个数据位中。

23.4.1. EUART 异步发送器 (UART TXIF/RCIF flag 由 0->1 发生中断)

图 23-3 为 EUART 发送器的时序图,发送器的核心是以串行方式发送移位寄存器 (Transmit Shift Register, TSR) 内的数据,而用户是无法读/写 TSR。

TSR 从读/写发送缓冲寄存器 TXxR[7:0]中获取资料。TXxR[7:0]寄存器中的数据由软件写入,在前一次加载的停止位发送完成前,不会再向 TSR 寄存器加载数据;一旦停止位发送完毕, TXxR 寄存器中的新数据(如果有的话)就会被加载 TSR。一旦 TXxR 寄存器向 TSR 寄存器传输了数据, TXxR 寄存器就为空(未再有写入动作),同时标志位 TXIF 由 1 置 0(当 URXCN 寄存器 ENTX 位被设置之后, TXIF 位将被设置为 1)。而 TXIF 不会在 TXxR 装入新数据时立即被清零,而是在装入新数据后的第二个指令周期被清零。当 TXIF 被置 0 之后,会在一个指令周期后再被置 1。可以通过将中断允许位 TXIE 置 1 或清零来允许/禁止该中断。不管 TXIE 的状态如何,只要中断发生, TXIF 就会由 1 置 0 并且不能由软件清零,并且会在一个指令周期后再被置 1。倘若此时 TSR 寄存器内的数据尚未发送完毕, TXxR 寄存器又被写入数据,则在装入新数据后的第二个指令周期后 TXIF 会再被清零并维持到 STOP BIT 发生时才会被置 1。

因此在 TXxR 装入新数据后立即查询 TXIF,其回传的值是不可参考。TXIF 表示的是 TXxR 寄存器的状态,而另一个位 TRMT 则表示 TSR 寄存器的状态。TRMT 是只读位,它在 TSR 寄存器为空(未再次有载入动作)时被置 1。TRMT 位与任何中断逻辑均无关联,因此要确定 TSR 寄存器是否为空,用户只能对此位进行查询。而数据异步发送时序图可参考下图 23-4,图 23-5 说明

- UART 动作除了读写以外,与 CPU 指令周期无关
- TXIF,RCIF 表示中断用途,与其他任何事件无关
- 要用 CPU 去检视周边组件时,须先注意相对操作速度

EUART TRANSMIT BLOCK DIAGRAM

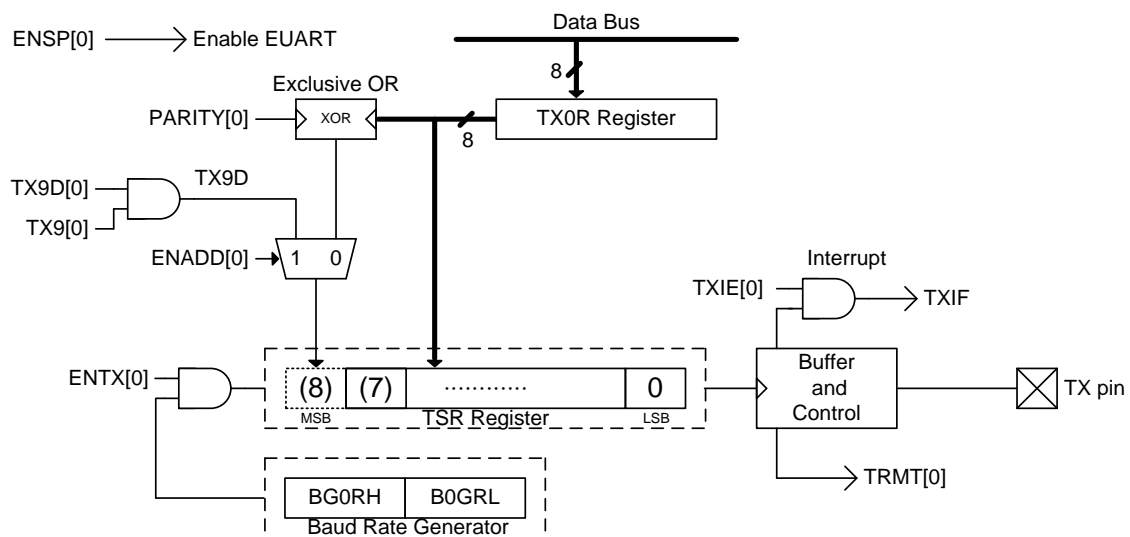


图 23-3 EUART 传送方块图

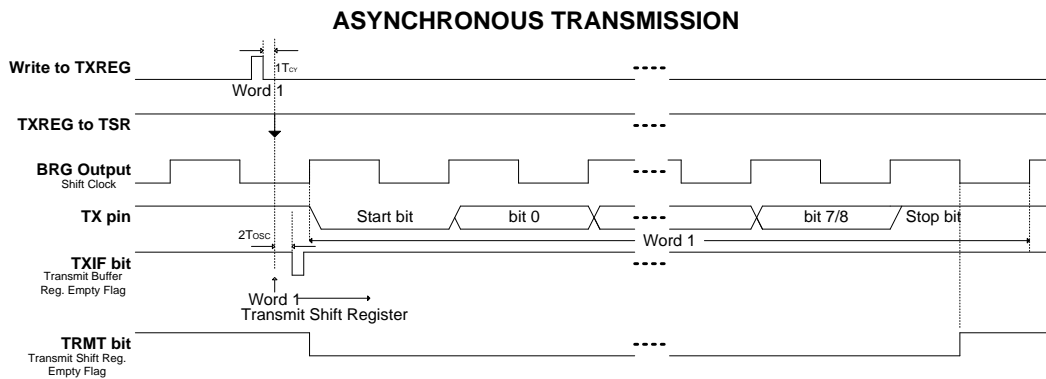


图 23-4 异步发送时序图

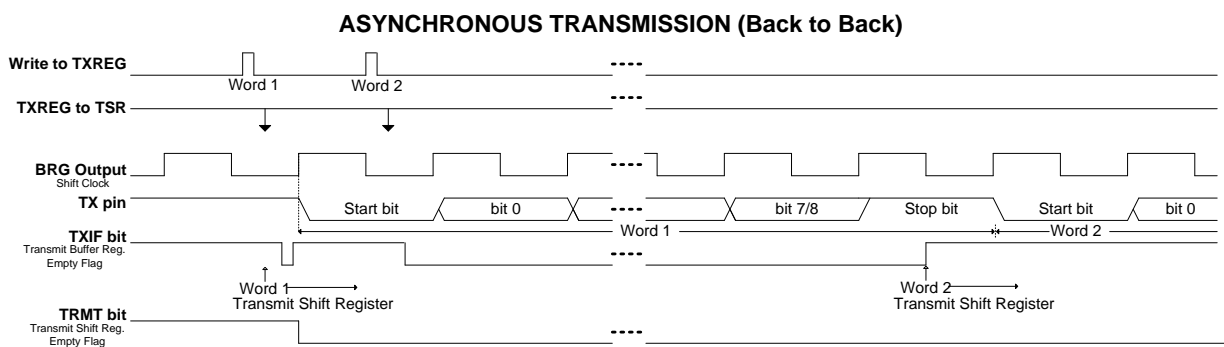


图 23-5 异步发送时序(背对背)

■ EUART 异步接收器

图 23-6、图 23-7 显示了接收器的方块图。图 23-8 显示了异步接收时序。在 RC 引脚上接收数据，并驱动数据恢复电路。数据恢复电路实际上是一个以 13 位串行传输波特率为工作频率的高速移位元器，而主接收串行移位元器的工作频率等于波特率或 OSC_RC2M。此模式通常用于 RS-232 系统。若 RC 引脚接收数据时未接收到完整字节(开始、8(9)位数据、结束)，FERR 位将被设置为 1，可透过清除 ENCR 位清除 FERR 位。

当 RC 引脚已经接收到 2 笔完整字节数据后(皆未从 RCxREG 寄存器中将数据读出)，OERR 位将被设置为 1，可透过清除 ENCR 位清除 OERR 位。

当完整数据接收完成时，INTF2 寄存器 RCIF 位将会被设置，而 RCIF 位被设置时将无法使用指令进行清除，执行读取 RCxREG 寄存器的动作将可以清除 RCIF 状态。

URxSTA 寄存器 RCIDL 位反应是否处于接收状态。用户间接由此判断数据接收是否完成。

数据接收时，硬件将会针对接收的 8 位数据进行互斥或运算(exclusive or)，若 RC9 设置为 1 时，将针对包含接收到的 RC9D 数据(共 9 位)进行互斥或运算(exclusive or)。运算后再与用户设定的 PARITY 位再次进行互斥或运算(exclusive or)，并将运算结果显示在 PERR 位。若接收数据正确，则 PERR 设定为 0。若接收数据错误，则 PERR 设定为 1。PERR 位无法使用指令清除，在下一笔数据接收正确时，PERR 将会被设置为 0。

EUART 8-BITS RECEIVE BLOCK DIAGRAM

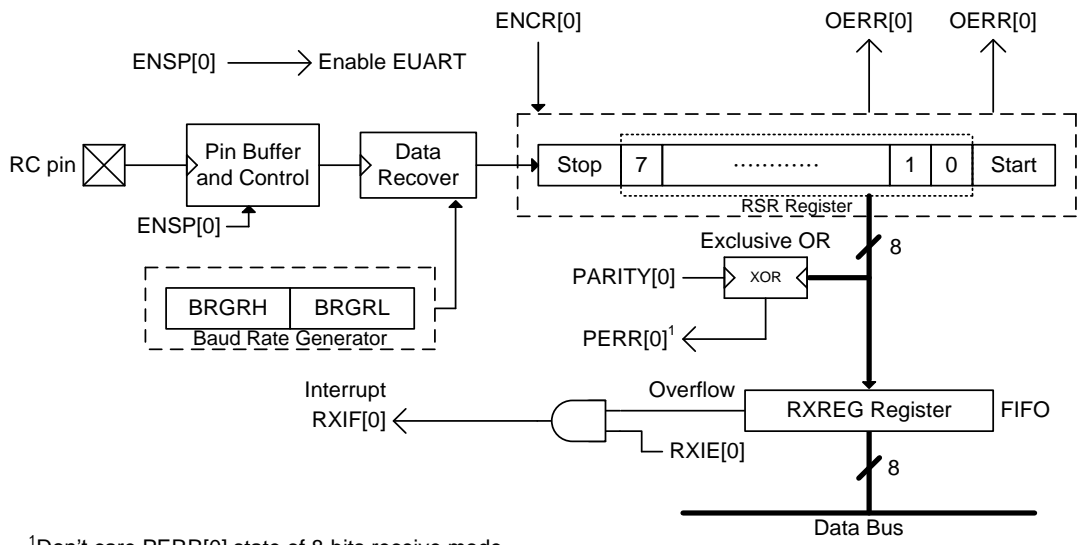


图 23-6 EUART 8-bits 接收方块图

EUART 9-BITS RECEIVE BLOCK DIAGRAM

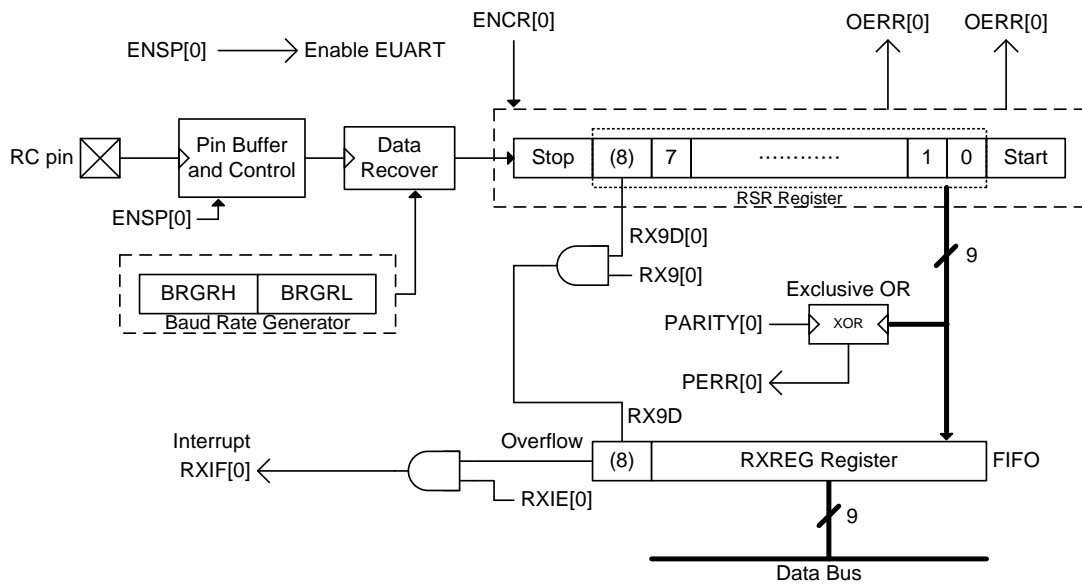


图 23-7 EUART 9-bits 接收方块图

ASYNCHRONOUS RECEPTION

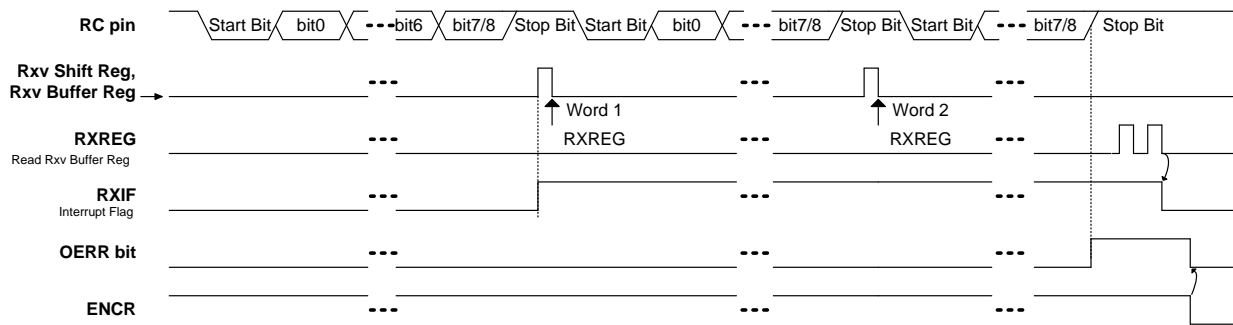


图 23-8 异步接收时序

■ 地址检测功能的 9 位模式

此模式通常用于 RS-485 系统。可依照 EUART 使用说明章节配置带有地址检测功能的异步接收操作。用户可透过 BAxCN 寄存器 ENADD 位设置决定为地址检测或是数据检测。

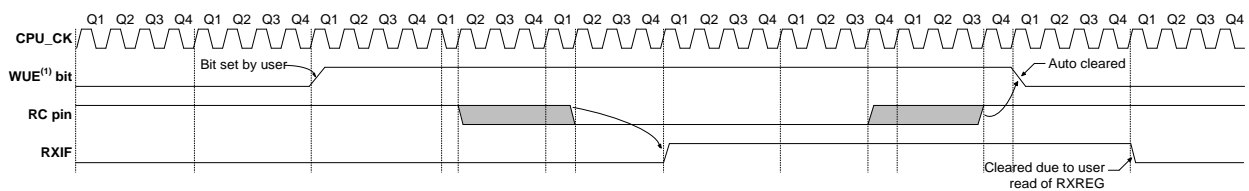
■ 字符接收自动唤醒

在休眠(sleep)或等待 (idle) 模式下，EUART 的所有频率源都会暂停。因此，串行传输波特率发生器处于非启动状态(ILDE UART 会动作)，并且无法进行正确的字节接收。自动唤醒功能允许在 RC 在线有事件发生时唤醒控制器，该功能需要 EUART 工作在异步模式下通过将 URXCN 寄存器 WUE 位设置 1(Sleep 需要置 1, Idle 不需要)，使能自动唤醒功能。该功能启用后，将禁止 RC 上的典型接收操作，且 EUART 保持在空闲状态并监视唤醒事件（与 CPU 运行模式无关）。

唤醒事件是指 RC 在线发生高电位到低电位的转换。在唤醒事件后，模块会产生一个 RCIF 中断，在正常工作模式下中断会与 Q 频率同步产生，可参考下图 23-9；如果芯片处于休眠或等待模式，则两者不同步，可参考下图 23-10 说明。通过读 RCxREG 寄存器可清除中断条件。

唤醒事件后，当 RC 在线出现由低向高的电位转换时，WUE 位自动清零。此时，EUART 模块将从空闲状态返回正常工作模式。由此用户可知事件已经结束。

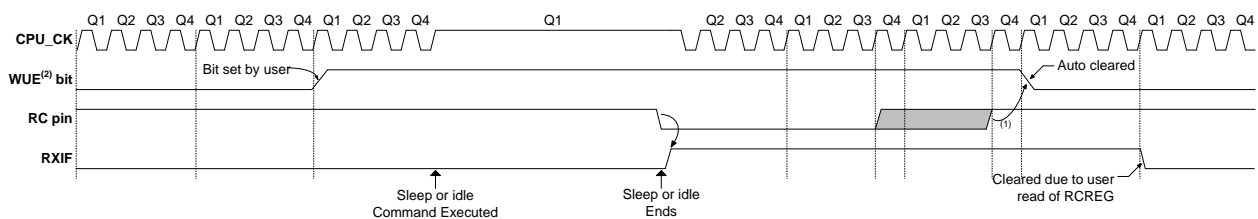
AUTO-WAKE-UP BIT (WUE) TIMINGS DURING NORMAL OPERATION



Note : ⁽¹⁾ The EUART remains in Idle while the WUE bit is set.

图 23-9 正常模式下自动唤醒时序

AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP OR IDLE



Note : ⁽¹⁾ If the wake-up event requires long oscillator warm-up time, the auto-clear of the WUE bit can occur before the oscillator is ready. This sequence should not depend on the presence of Q clocks.
⁽²⁾ The EUART remains in Idle while the WUE bit is set.

图 23-10 休眠或等待模式下自动唤醒时序

- 使用自动唤醒功能的注意事项

因为自动唤醒功能是通过检测 RC 上的上升沿跳变实现的，所以在停止位前该引脚上任何的状态改变都可能会产生错误的结束信号并导致数据或帧错误。因此，为了确保正确的传输，必须首先发送全 00H。(如果是 9bit 必须 9 个 bit 都是 0)

另外还必须考虑振荡器起振时间，尤其在采用起振延迟较长的振荡器应用中更要注意这一点。或唤醒信号字符必须足够长，并且跟有足够长的时间间隔，以便使选定振荡器有充足的时间起振并保证 EUART 正确初始化。

- 使用 WUE 位的注意事项

WUE 和 RCIF 事件的时序来判断接收数据的有效性可能会引起混淆。如前所述，将 WUE 位置 1 会使 EUART 进入空闲模式。唤醒事件会产生一个接收中断，并将 RCIF 位置 1。此后当 RC 出现上升沿时 WUE 位被清零。然后通过读 RCxREG 寄存器清除中断条件。

一般情况下，此时唤醒后 RCxREG 中的数据是无效数据，应该丢弃。WUE 位清零（或仍然置 1）且 RCIF 标志位置 1 并不能表明 RCxREG 中数据接收是完整的。用户应该考虑使用固件验证是否完整地接收了数据。要确保没有丢失有效数据，应检查 RCIDL 位来验证是否还在接收数据。如果不在进行接收，则可将 WUE 位置 1，使芯片立即进入休眠模式。

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



23.5. 寄存器说明- UART1/2

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1											
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition											
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W
INTE0	GIE	-	ADIE	WDTIE	TB1IE	TMAIE	E1IE	E0IE	0000 0000	0uuu uuuu	* * * * *
INTE1	-		TXIE	RCIE					0000 0000	uuuu uuuu	* * * * *
INTF1	-	SPIIF	TXIF	RCIF	I2CERIF	I2CIF	E3IF	E2IF	0000 0000	uuuu uuuu	* * * * r, r, *
UR0CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	* * * * *, -, *
UR0STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	-, r, r, r, r, r, rw0
BA1CN	-	-	-	-	ENCR	RC9	ENADD	ENABD 0000 uuuu	-, -, -, *, *, *
BG1RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	-, -, -, *, *, *
BG1RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	* * * * *
TX1R	UART Transmit Register								xxxx xxxx	uuuu uuuu	* * * * *
RC1REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r, r, r, r, r, r, r
UR2CN	ENSP	ENTX	TX9	TX9D	PARITY	-	-	WUE	0000 0..0	uuuu u..u	* * * * *, -, *
UR2STA	-	RC9D	PERR	FERR	OERR	RCIDL	TRMT	ABDOVF	.000 0010	.uuu uuuu	-, r, r, r, r, r, rw0
BA2CN	-	-	-	-	ENCR	RC9	ENADD	ENABD 0000 uuuu	-, -, -, *, *, *
BG2RH	-	-	-	Baud Rate Generator Register High Byte					...x xxxx	...u uuuu	-, -, -, *, *, *
BG2RL	Baud Rate Generator Register Low Byte								xxxx xxxx	uuuu uuuu	* * * * *
TX2R	UART Transmit Register								xxxx xxxx	uuuu uuuu	* * * * *
RC2REG	UART Receive Register								xxxx xxxx	uuuu uuuu	r, r, r, r, r, r, r

表 23-3 UART1/2 寄存器

INTE0/INTE1/INTF1: 详见 中断 章节

URxCN: UART1/2 控制寄存器(x=0 or 2)

位	名称	描述
Bit7	ENSP	UARTx 端口功能致能位 <0> 关闭 UARTx 端口, 并将 TX, RC 引脚组态为 I/O 使用 <1> 启动 UARTx 端口, 并将 TX, RC 引脚组态为 UARTx 端口使用 注: 当启动 UARTx 串行端口之后, 须适当配置输入或输出脚位使用.
Bit6	ENTX	UARTx 传送功能使能位 <0> 关闭 <1> 启动
Bit5	TX9	传送第 9 位功能使能 <0> 关闭 <1> 启动
Bit4	TX9D	传送第 9 位数据 <0> 资料为“0” <1> 资料为“1”
Bit3	PARITY	奇/偶同位检查设定 <0> 偶同位检查 <1> 奇同位检查

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

Bit0	WUE	字符接收自动唤醒使能位 <0> 关闭 <1> 启动
------	-----	---------------------------------

URxSTA: UART1/2 状态寄存器

位	名称	描述
Bit6	RC9D	接收第 9 位数据 <0> 资料为"0" <1> 资料为"1"
Bit5	PERR	资料同位检查结果标志 <0> 接收同位检查正确 <1> 接收同位检查错误
Bit4	FERR	UART 数据接收不完整(开始、8(9)位数据、结束)标志 <0> 表示数据接收完整 <1> 表示数据接收不完整
Bit3	OERR	已接收到 2 笔数据未处理状态标志 <0> 未发生 <1> 已发生
Bit2	RCIDL	反应是否为接收状态标志 <0> 在接收状态 <1> 不在接收状态
Bit1	TRMT	表示传送移位寄存器(TSR)状态标志 <0> 表示 TSR 寄存器有数据 <1> 表示 TSR 寄存器为空的
Bit0	ABDOVF	自动波特率溢位标志 <0> 未发生 <1> 已发生

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller

BAxCN: UART 接收数据控制寄存器

位	名称	描述
Bit3	ENCR	数据接收功能使能位 <0> 关闭。 <1> 启动。
Bit2	RC9	接收第 9 位功能使能位 <0> 关闭 <1> 启动。
Bit1	ENADD	地址检测位 <0> 关闭。 <1> 启动。
Bit0	ENABD	自动波特率控制器使能位 <0> 关闭。 <1> 启动。

BRxRH/BRxRL: Baudrate1/2 控制寄存器

TXxR: UART1/2 数据传送寄存器

RCxREG: UART1/2 数据接收寄存器

24. 内建 EPROM, Build-In EPROM

Build-In EPROM(BIE)功能可储存产品序号、安全密码、程序运算后产生的数据数据...等，外部硬件仅需外接 VBIE 为 8.5V 或 7.75V(HY17P51)电压于 VPP/RST 引脚，或使用低压烧录控制电路，并搭配烧录指令使用此功能，HY17P 系列储存地址范围 00H~3FH 共 64 words 同等于 128 bytes，视各芯片不同 Build-In EEPROM 大小也会不同。

当使用外部 VBIE 电源(8.5V 或 7.75V)烧录 BIE 区块时，可以透过指令一次烧录一个字组(word)资料于 BIE 区块内；当使用 HY17P 系列芯片启动低压烧录控制电路时，则不需外接 VBIE 电源仍可烧录 BIE 区块，但须呼叫烧录子程序(LV17PWRBIE)才能进行烧录；每次呼叫烧录子程序进行烧录动作，仅能烧录一个字组(word)数据，所花费时间约 500msec。

BIE 寄存器摘要：

BIECN	ENBVD[0], VPPHV[0], ENBCP[0], BIEWR[0], BIERD[0]
BIEARH	ENBIE[0]
BIEARL	BIE_ADDR[5:0]
BIEDH	BIE_DATA[15:8]
BIEDL	BIE_DATA[7:0]

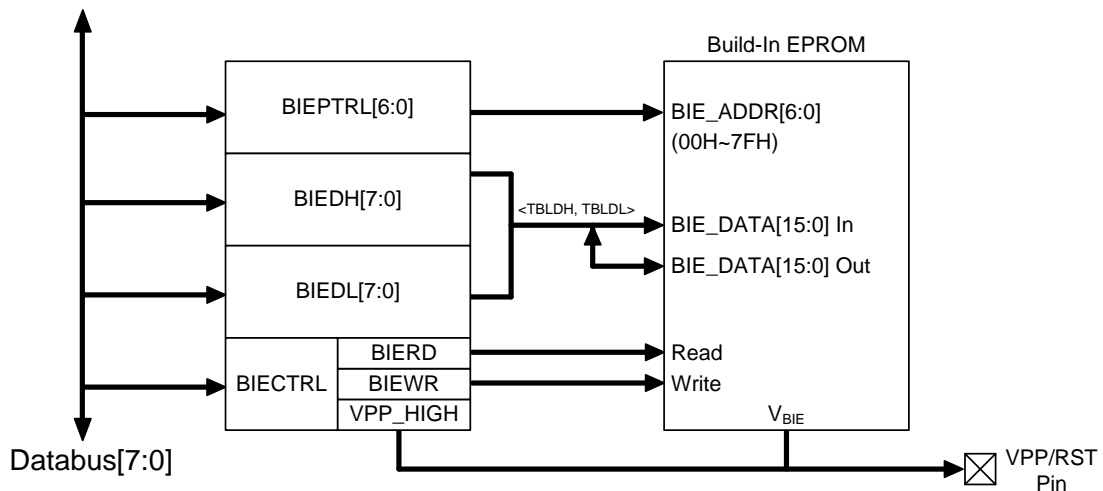


图 24-1 BIE 方块图

HY17S58 Emulate Chip User' Guide

Embedded $\Sigma\Delta$ ADC
8-Bit RISC-like Mixed Signal Microcontroller

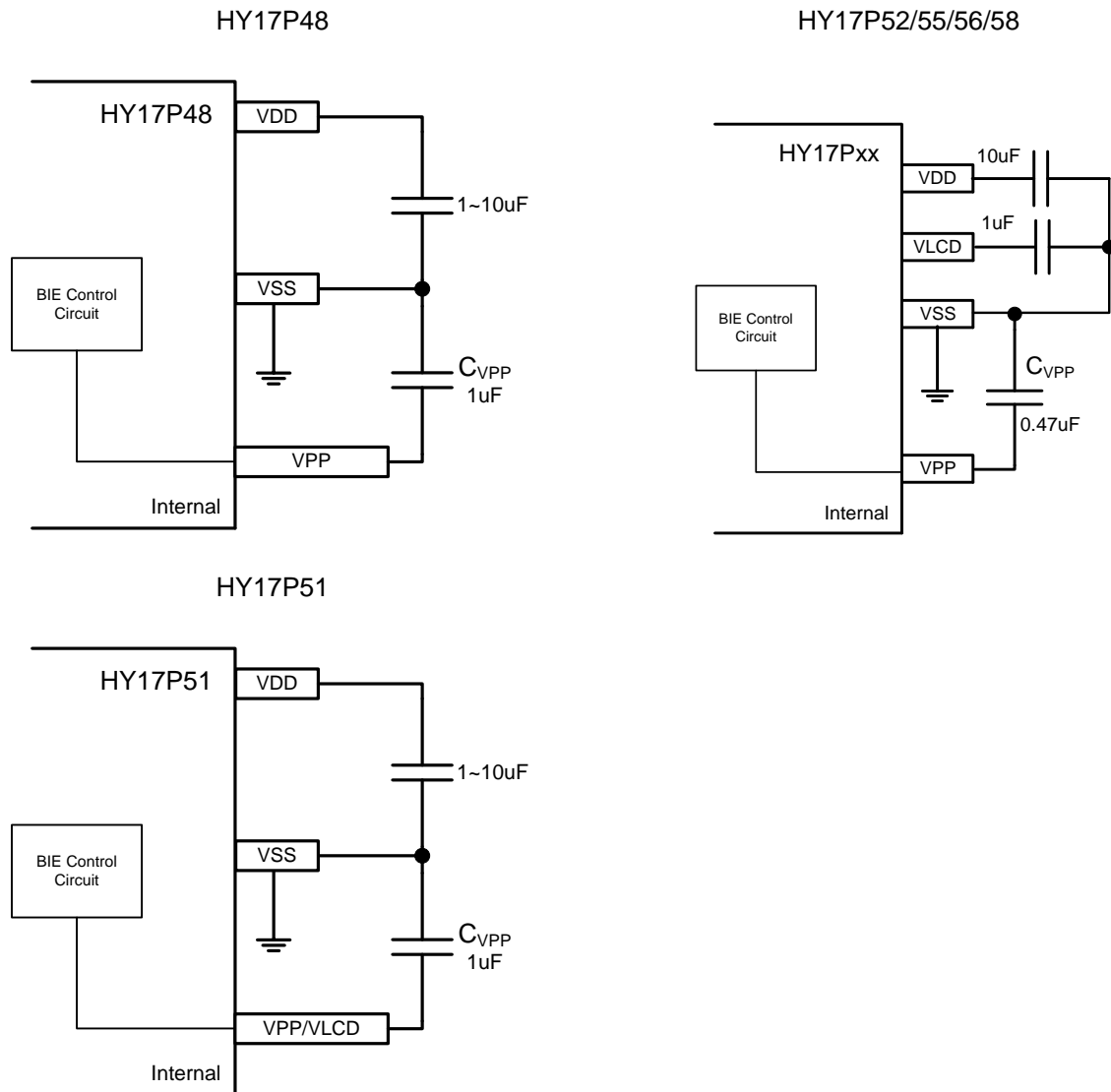


图 24-2 BIE 升压方块图

24.1. 寄存器说明- BIE

“-”no use, “*”read/write, “w”write, “r”read, “r0”only read 0, “r1”only read 1, “w0”only write 0, “w1”only write 1												
“\$”for event status, “.”unimplemented bit, “x”unknown, “u”unchanged, “d”depends on condition												
File Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	A-RESET	I-RESET	R/W	
BIECN	1			ENBVD	VPPHV	ENBCP	BIEWR	BIERD	1... \$000	1... \$uuu	r1,-,-,r,-,*	
BIEARH	-		1	1	1	1	1	1	0... xxxx	u... uuuu	*,-,-,*,-,*	
BIEARL			BIE Address Register as BIEAL[5:0]							xxxx xxxx	uuuu uuuu	*,-,-,*,-,*
BIEDRH	BIE High Byte Data Register								xxxx xxxx	uuuu uuuu	*,-,-,*,-,*	
BIEDRL	BIE Low Byte Data Register								xxxx xxxx	uuuu uuuu	*,-,-,*,-,*	

表 24-1 BIE 寄存器

BIECN: BIE 控制寄存器

位	名称	描述
Bit4	ENBVD	BIE 电压侦测控制寄存器 <0> 关闭 <1> 启用
Bit3	VPPHV	used to indicate if BIE voltage is good for OTP programming. It is a flag for MCU. Notice it needs to set ENBVD=1 first <0> Not ready (lower than 8.5V) <1> Ready (over 8.5V) Check VPP <0> VPP 未外接烧录电源 8.5V <1> VPP 已外接烧录电源 8.5V
Bit2	ENBCP	BIE charge pump 启用控制寄存器 <0> 关闭 <1> 启用
Bit1	BIEWR	写入 EPROM 控制位 <0> 不可写入 <1> 可写入
Bit0	BIERD	读取 EPROM 控制位 <0> 不可读取 <1> 可读取

BIEARL: EPROM Low Byte 地址定义

BIEAL[5:0]: OTP address

BIEDH: EPROM High Byte 数据定义

BIEDL: EPROM Low Byte 数据定义

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



25. 修订记录

以下描述本文件差异较大的地方，而标点符号与字形的改变不在此描述范围。

文件版次	日期	页次	摘要
V01		All	初版发行
V02		All	更新各功能说明
V03		21~29	寄存器列表
		30~36	CPU 及周边电路频率章节
		97~114	修改 PWM 说明
		119~135	模拟数字转换器 Σ ADC 章节
		157	LCD 寄存器描述
V04		132~133	修改 TPS 使用说明
		53~77	修改 LCD IP 相关描述
		158~166	修改 LCD IP 相关描述
		198~202	修正 UART 操作描述
		154~157	修正可程序放大 PGA 说明
		82	修正 HY17P48 LVD 使用方式
		150~154	修改 OPAMP1 网络说明
		156~159	修改 LNOP2 网络说明
V05	2020/4/24	131~136	修改 TPS 操作说明，并增加 HY17P48/58 描述
		161~165	修改 LCD 操作说明，主要增加 LCDPU 功能描述因应 RE 测试需求
			修改 SPI 说明，并移除没有的寄存器
		159	修正 OPDIEN 描述
		151-160	修改 R2ROP、LNOP 通道与说明
		140	补充 HY17P48 INN=A13 应用限制
V06	2020/9/8	80	补充 LED 定电流操作说明
			补充 3.4 章节功率消耗管理与操作状态
		38、87	修正 WDT 频率源说明
	2021/1/5	33、34	新增 HY17P51 寄存器列表
		129~147	新增 HY17P51 ADC 说明
		213	新增 HY17P51 BIE 说明
	2021/2/2	87	修正 WDT 除频说明
	2021/4/20	164~172	新增 LCD Type 描述
		41	修改 XTS 寄存器描述
		122~125	修改 TMC 输入频率错误描述
	2021/5/5	92~99	新增 TMB 中断产生说明
	2021/8/27	140~141	修改 ADC Gain 说明

HY17S58 Emulate Chip User' Guide

Embedded Σ ADC
8-Bit RISC-like Mixed Signal Microcontroller



V07	2022/6/8	131~132	增加 HY17P51/52 ADC 网络图
		164	修改 OPDIEN 寄存器说明