DT5M4765 Datasheet

24 位高精度 ADC 和 LCD 驱动的低功耗 32 位 MCU

特性

一CPU 特性

- 32位高性能的RISC CPU
- 工作电压2.2v~5.4V
- 单周期指令
- CPU最高工作频率80MHZ，工作频率可调在100KHZ到80MHZ
- CPU 在ROM运行能做到真正无等待工作频率16Mhz
- CPU 在内部RAM运行能到真正无等待工作频率80Mhz
- 每个中断有中断入口，可减少中断源查询
- 待机功耗<8uA
- 工作电流（20MHZ）<5mA
- 支持深度睡眠（时钟完全静止）和普通睡眠（大部分时钟保持运行）
- 在深度睡眠可以保持PLL运行，这种情况下保证2us内完成唤醒

一存储空间

- 64KByte MTP
- 256byte EEPROM
- 16KByte SRAM 可用作数据或程序快速运行

一非常强的安全性

- 每个客户可以有唯一ID，ID不能擦除或改写
- fuse烧断后，程序空间和RAM的空间都不能访问
- fuse烧断后，SWD通信中断，但按特定模式可把整个程序空间擦除，以便多次使用程序空间

一开发工具

- 支持SWD标准通信
- Keil等标准开发工具可使用

外设特性

-ADC

- 8路ADC
- 分辨率24BIT，ENOB最高19BIT
- 采样频率最快1KHZ
- ADC 内部 PGA用于小信号放大，支持倍数1.1倍~72倍

一独立OP（rail2rail） X2

- 信号输入和输出都有管脚，可用于小信号放大，恒流源，比较器等使用

-无晶振USB

- USB 2.0 device，Full Speed
- 支持4个end points，EP0支持IN/OUT包， EP1-3支持IN包

-I2C

- I2C master & slave， 可到1Mhz通信速度

-UART

- 最高速度115200bps

-SPI master

- 最高通信速度10Mhz

-PWM （共6 路）

- 6路独立PWM，10bit 最大频率32MHz
- 可以配置两路非独立PWM（来至同已PWM），相位互反或一致，并带死区

-数字Sigma Delta 声音（Audio+，Audio-）

- 声音格式PCM，ADPCM
- 最高位数16位
- 声音工作频率可到64MHz

-Timer x2

- 2路32位独立timer，最高工作频率32MHz

-RTC

- 32位RTC
- 工作频率来源于晶振频率32.768KHz
- 深度睡眠保持工作

-WDT

- 32位，工作频率来源于晶振频率32.768KHz
- 深度睡眠保持工作，可睡眠唤醒
- 可产生中断或复位芯片

-SDIO2.0
- 最高工作频率32MHZ
- 可单线或4线通信模式
- 提供最快的通信速度（4线模式）

-低电压检测
- 可配置多档低电压检测阀值
- 可产生中断或复位芯片，由配置位决定

-段码LCD支持
- 8COMx32SEG，6COMx34SEG，4COMx36SEG，可灵活配置LCD矩阵数（COM， SEG）
- 内置charge pump
- 最大输出电压3.0~4.4V 可配置
- LCD管脚可用作普通GPIO

—GPIO（共56 GPIO）
- 所有数字功能脚都可以配置为GPIO

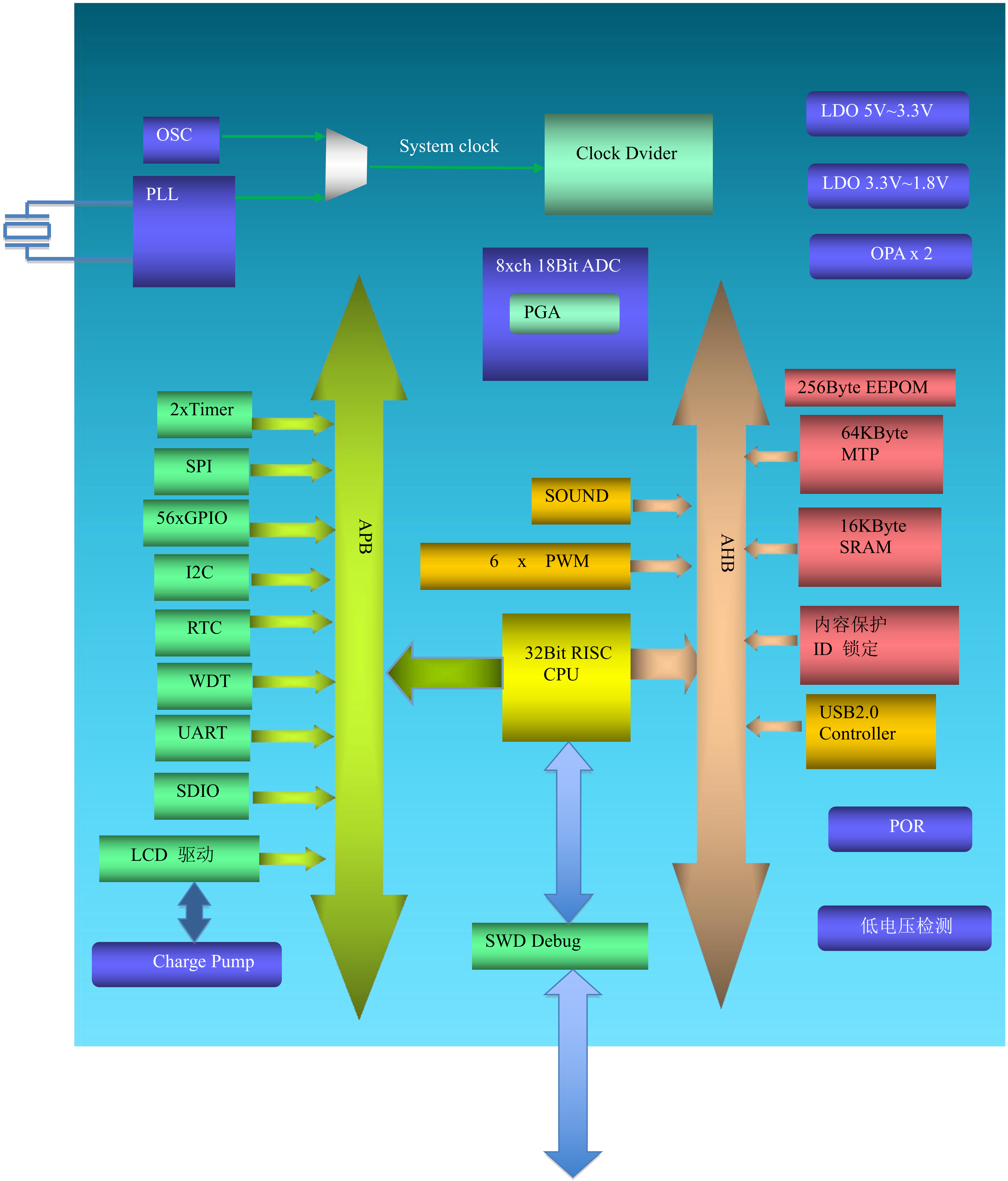
—内置PLL

—两个内置LDO
- 5V到3.3V LDO：输入范围3.6V—5.5V，驱动能力位100ma，可为系统其他芯片提供3.3V电压
- 3.3V 到1.8V LDO：输入电压范围2.0-3.4V，驱动能力30ma

—内部POR
- 可减少外部复位电路

DT5M4765 型号选择

功能 型号	MTP（byte）	EEPROM(byte)	24位ADC (通道数)	24 位单端 ADC (通道数)	Timer (通道数)	10bit 独立 PWM (通道数)	LCD	OP	USB
DT5M4765V8L6	64K	256	8 对	16 个	2	6	8x32	2	1
DT5M4765R8L6	64K	256	3 对	7 个	2	6	8x25	1	N/A
DT5M4765C8Q6	64K	256	2 对	6 个	2	4	4x20	1	N/A
DT5M4765K8Q6	64K	256	5 对	10 个	2	6	N / A	0	1


OSC
PLL
System clock
Clock Dvider
LDO 5V~3.3V
LDO 3.3V~1.8V
OPA x 2
8xch 18Bit ADC
PGA
256Byte EEPOM
64KByte MTP
16KByte SRAM
内容保护 ID 锁定
USB2.0 Controller
POR
低电压检测
APB
AHB
2xTimer
SPI
56xGPIO
I2C
RTC
WDT
UART
SDIO
LCD 驱动
Charge Pump
SOUND
6 x PWM
32Bit RISC CPU
SWD Debug

Pin List （管脚描述）

Number	Name	Bonding Number	Description
1	VSS:	1	外接 GND
2	pad_cp_out	2	Charge pump 输出脚（对地 4.7uf）
3	pad_cp_capn	3	两个 pin 之间接 1uf 电容
4	pad_cp_capp	4	
5	VSS_CHGPUMP	5	Charge pump VSS 外部接 GND
6	VDD:	6	USB PHY 1.8V 内部已经跟 1.8V 电源相连，可用于测量内部 1.8V 是否准确，可外接电容
7	pad_dp	7	USB D+
8	pad_dm	8	USB D-
9	VSS:	9	USB PHY 内部模拟地 外接 GND
10	VSS_USB:	10	USB PHY 内部模拟地（用于 Bandgap）外接 GND
11	VDD33_USB:	11	外接 3.3V
12	VDD33_OP2	12	OP2 供电电压，外接 3.3V
13	pad_op2_vin	13	OP2 负向输入端
14	pad_op2_vip	14	OP2 正向输入端
15	pad_op2_vo	15	OP2 输出端
16	VSS_OP2	16	OP2 GND，外接 GND
17	VDD33_ADC	17	ADC 供电电压，外接 3.3V
18	VSS_ADC	18	ADC GND
19	pad_adc_inp0	19	ADC channel0 独立差分正向输入端，测量效果比跟 gpio 复用的效果更好
20	VDD_ADC	20	ADC 模拟电压输出 约等于 1.7V，可提供 2mA 驱动能力
21	pad_adc_inn0	21	ADC channel0 独立差分负向输入端，测量效果比跟 gpio 复用的效果更好
22	pad_asyn_rstn	22	芯片外部复位脚
23	pad_gpio[3]	23	复用功能 SWD CLK
24	pad_gpio[4]	31	复用功能 SWD DATA
25	pad_gpio[0]	32	ADC_AIN11/ADC_AIP4
26	pad_gpio[1]	33	ADC_AIP11/ADC_AIN4
27	pad_gpio[2]	34	ADC_AIN12/ADC_AIP5
28	pad_gpio[5]	35	复用功能 SPI_DATA_OUT/ADC_AIP12/ADC_AIN5
29	pad_gpio[6]	36	复用功能 SPI_CLK/ADC_AIN13/ADC_AIP6
30	pad_gpio[7]	37	复用功能 SPI_DATA_IN/ADC_AIP13/ADC_AIN6
31	pad_gpio[8]	38	复用功能 SPI_CS/ADC_AIN14/ADC_AIP7
32	pad_gpio[9]	39	复用功能 UART_TXD/ADC_AIP14/ADC_AIN7
33	pad_gpio[10]	40	复用功能 UART_RXD/ADC_AIN15/ADC_AIP8
34	pad_gpio[11]	41	复用功能 I2C_SCL/ADC_AIP15/ADC_AIN8
35	pad_gpio[12]	42	复用功能 I2C_SDA/ADC_AIP9
36	pad_gpio[13]	43	复用功能 PWM1_OUT/ADC_AIN9（默认输出低电平）
37	pad_gpio[14]	44	复用功能 PWM2_OUT/ADC_AIP10（默认输出低电平）
Number	Name	Bonding	Description

		Number	
38	pad_gpio[15]	45	复用功能 AUDIO_PWM+/ADC_AIN10
39	pad_gpio[16]	46	复用功能 AUDIO_PWM-/Timer_clk_in
40	pad_gpio[17]	47	复用功能 LCD COM0
41	pad_gpio[18]	48	复用功能 LCD COM1
42	pad_gpio[19]	49	复用功能 LCD COM2
43	pad_gpio[20]	50	复用功能 LCD COM3
44	VDD33:	51	外接 3.3V
45	VSS:	52	外接 GND
46	pad_gpio[21]	53	复用功能 LCD COM4
47	pad_gpio[22]	54	复用功能 LCD COM5
48	pad_gpio[23]	55	复用功能 LCD COM6/SPI_DATA_OUT
49	pad_gpio[56]	61	复用功能 LCD COM7/SPI_CLK
50	pad_gpio[24]	62	复用功能 LCD SEG0
51	pad_gpio[25]	63	复用功能 LCD SEG1
52	pad_gpio[26]	64	复用功能 LCD SEG2
53	pad_gpio[27]	65	复用功能 LCD SEG3
54	pad_gpio[28]	66	复用功能 LCD SEG4
55	pad_gpio[29]	67	复用功能 LCD SEG5
56	pad_gpio[30]	68	复用功能 LCD SEG6
57	pad_gpio[31]	69	复用功能 LCD SEG7
58	pad_gpio[32]	70	复用功能 LCD SEG8
59	pad_gpio[33]	71	复用功能 LCD SEG9
60	pad_gpio[34]	72	复用功能 LCD SEG10
61	pad_gpio[35]	73	复用功能 LCD SEG11
62	pad_gpio[36]	74	复用功能 LCD SEG12
63	pad_gpio[37]	75	复用功能 LCD SEG13
64	pad_gpio[38]	76	复用功能 LCD SEG14
65	pad_gpio[39]	77	复用功能 LCD SEG15
66	pad_gpio[45]	78	复用功能 LCD SEG21
67	pad_gpio[46]	79	复用功能 LCD SEG22/SPI_DATA_IN
68	pad_gpio[48]	80	复用功能 LCD SEG24/UART_RXD
69	pad_gpio[49]	81	复用功能 LCD SEG25/UART_TXD
70	pad_gpio[50]	82	复用功能 LCD SEG26/I2C_SCL
71	pad_gpio[51]	83	复用功能 LCD SEG27/I2C_SDA
72	VSS:	91	外接 GND
73	VDD33:	92	外接 3.3V
74	pad_gpio[40]	93	复用功能 LCD SEG16
75	pad_gpio[41]	94	复用功能 LCD SEG17
76	pad_gpio[42]	95	复用功能 LCD SEG18
Number	Name	Bonding Number	Description
77	pad_gpio[43]	96	复用功能 LCD SEG19

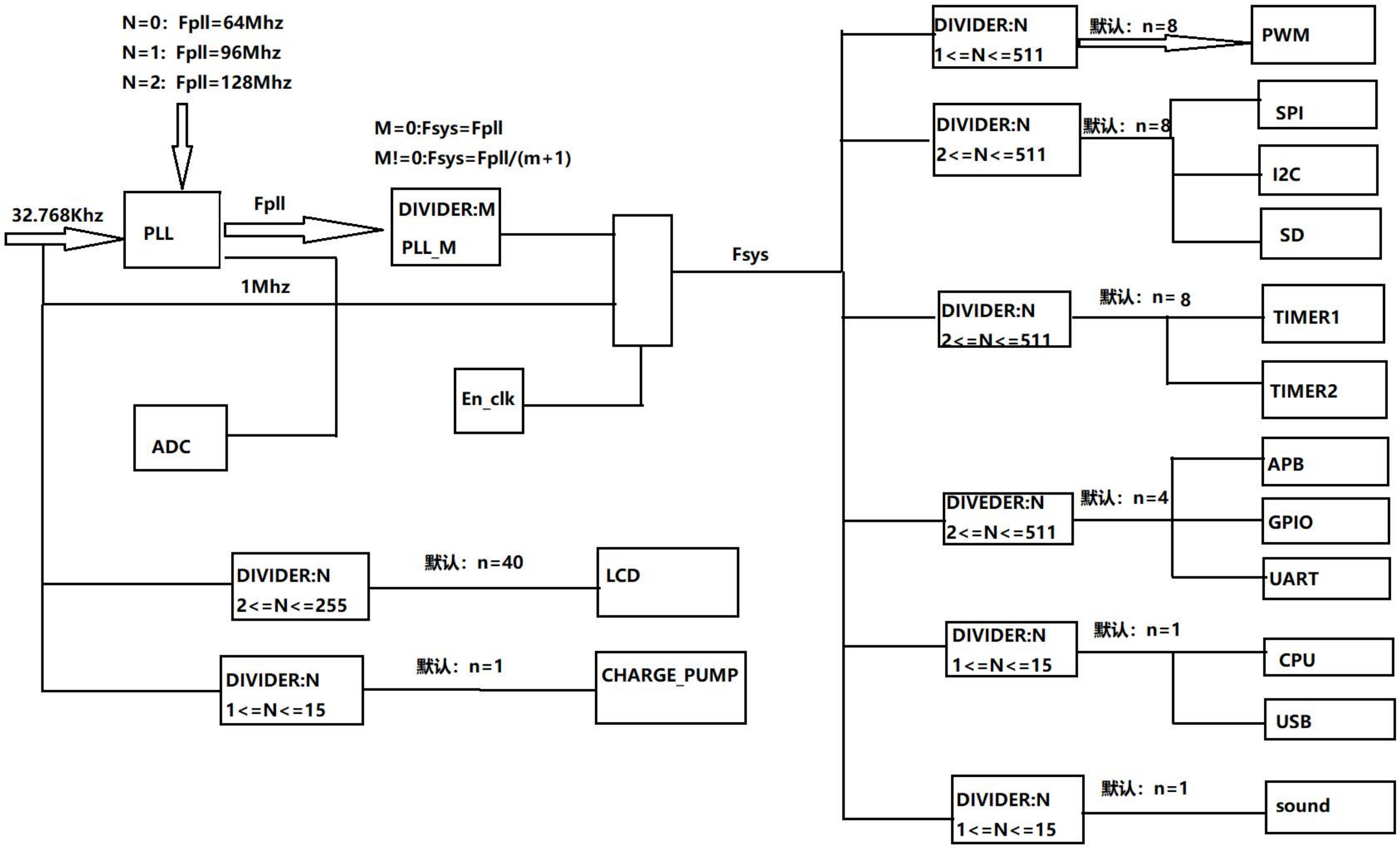
78	pad_gpio[44]	97	复用功能 LCD SEG20
79	pad_gpio[47]	98	复用功能 LCD SEG23/SPI_CS
80	pad_gpio[52]	99	复用功能 LCD SEG28/PWM5（默认输出低电平）
81	pad_gpio[53]	100	复用功能 LCD SEG29/PWM6（默认输出低电平）
82	pad_gpio[54]	101	复用功能 LCD SEG30/PWM3（默认输出低电平）
83	pad_gpio[55]	102	复用功能 LCD SEG31/PWM4（默认输出低电平）
84	pad_XIN	103	外部晶振输入脚 频率 32KHz--对地接 20pf（请根据参考原理图）
85	pad_XOUT	104	外部晶振输出脚 频率 32KHz--对地接 20pf（请根据参考原理图）
86	VDD33_CRYSTAL	105	晶振电源--外接 3.3V（请根据参考原理图）
87	VSS_CRYSTAL	106	外接 GND
88	VDD33_OP1	107	OP1 供电-外接 3.3V
89	pad_op1_vo	108	OP1 输出端
90	pad_op1_vin	109	OP1 反向输入端
91	pad_op1_vip	110	OP1 正向输入端
92	VSS_OP1	111	OP1 GND -外接 GND
93	pad_vdd5:	112	5V~3.3V LDO，5V 电压输入端
94	pad_vdd5:	113	
95	pad_vdd5:	114	
96	pad_vbg	115	LDO 电源基准电压输入脚，如果是使用 5V~3.3V LDO，这个脚接 5V，如果采用外接 3.3V 方案，这个脚接 3.3V
97	VDD33:	116	LDO 3.3V~1.8V 3.3V 输入脚，外接 3.3V
98	pad_vdd33_out:	117	5V~3.3V LDO OUT ，3.3V 输出 ，建议外接 1uF~10uF 电容
99	pad_vdd33_out:	118	
100	pad_vdd33_out:	119	

Memory Map，地址空间分配

地址段	描述
0x0000,0000-0x0000,FFFF	内部程序空间
0x0001,0000-0x0001,00FF	内部 EEPROM 空间
0x0100,0000-0x0100,0FFF	USB
0x0200,0000-0x02FF,FFFF	LCD
0x0300,0000-0x0300,3FFF	内部 RAM
0x0400,0000-0x0400,0FFF	系统配置寄存器及 ADC 配置寄存器
0x0500,0000-0x0500,0FFF	PWM 寄存器
0x0600,0000-0x0600,0FFF	Sound 寄存器
0x0700,0000-0x0700,0FFF	SDIO
0x1000,0000-0x1000,00FF	时钟控制
0x1000,0100-0x1000,01FF	GPIO 控制
0x1000,0300-0x1000,03FF	SPI
0x1000,0400-0x1000,04FF	I2C

0x1000,0500-0x1000,05FF	RTC
0x1000,0700-0x1000,07FF	WDT
0x1000,0800-0x1000,08FF	UART
0x1000,0900-0x1000,09FF	Timer

全局时钟结构图



Note：Fpll 的时钟（PLL 频率）现在是有三档 64MHz 96MHz 128MHz
实际测量跟标称值有一定偏差，实际频率为标称值的负向偏差 4.4%，既为 64MHz – 64MHz *4.4% = 61.184MHz。芯片一致性比较好，如果需要准确定时计算请参照此真实值计算。

2. Memory Map & Register Discrption

2.1 0x02000000~0x02ffffff LCD controller

1) 0x02000000 lcd_ctl default:0x00000000 R/W
Bit 31 : LCD_EN
Bit 30~13: Reserved
Bit 12: LCD Wake up request
Bit 11: LCD Wake up interrupt enable
Bit 10: LCD Power down request

Bit 9: LCD Power down interrupt enable
Bit8: LCD Power down display enable
Bit7: LCD Blink on enable
Bit6~3: Reserved
Bit2~0: Duty select
"000": static, "001": 1/2 duty, "010": 1/3 duty,
"011": 1/4 duty, "100": 1/5 duty, "101": 1/6 duty
"110": 1/7 duty, "111": 1/8 duty

2) 0x02000004 lcd_dispctl default:0x000000 R/W
Bit 31~19: Reserved
Bit18~17: LCD MODE
"00" 8x32 ,"01" 6x34, "10" 4x36 "11" reserved
Bit16: Double drive
Bit15 : Display all on
Bit14 : Display all off
Bit13~11: Reserved
Bit10~8: Reserved
Bit 7~3: Reserved
Bit 2~1: Bias select
“00”:static “01”:1/2 bias “10”:1/3 bias “11”:1/4 bias
Bit0 : Reserved

3) 0x02000008~0x02000024 lcd_mem0~lcd_mem7
Default : 0x00000000 R/W

LCD display memory map

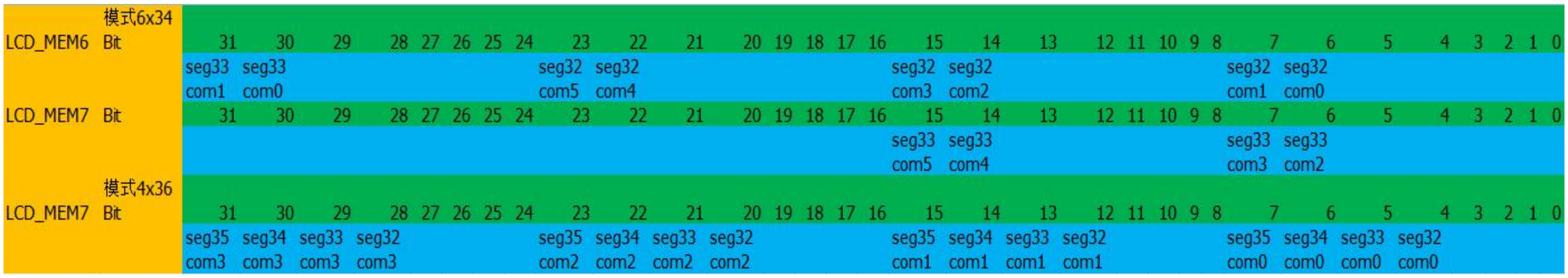
DISPLAY Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COM	COM7	COM6	COM5	COM4	COM3	COM2	OCM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	OCM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	OCM1	COM0	COM7	COM6	COM5	COM4	COM3	COM2	OCM1	COM0
LCD_MEM_7	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
LCD_MEM_6	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
LCD_MEM_5	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
LCD_MEM_4	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
LCD_MEM_3	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
LCD_MEM_2	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG09	SEG09	SEG09	SEG09	SEG09	SEG09	SEG09	SEG09	SEG08	SEG08	SEG08	SEG08	SEG08	SEG08	SEG08	SEG08
LCD_MEM_1	SEG07	SEG07	SEG07	SEG07	SEG07	SEG07	SEG07	SEG07	SEG06	SEG06	SEG06	SEG06	SEG06	SEG06	SEG06	SEG06	SEG05	SEG05	SEG05	SEG05	SEG05	SEG05	SEG05	SEG05	SEG04	SEG04	SEG04	SEG04	SEG04	SEG04	SEG04	SEG04
LCD_MEM_0	SEG03	SEG03	SEG03	SEG03	SEG03	SEG03	SEG03	SEG03	SEG02	SEG02	SEG02	SEG02	SEG02	SEG02	SEG02	SEG02	SEG01	SEG01	SEG01	SEG01	SEG01	SEG01	SEG01	SEG01	SEG00	SEG00	SEG00	SEG00	SEG00	SEG00	SEG00	SEG00

扩展模式：

1）6COM x 34SEG（COM6=SEG32，COM7=SEG33）

2）4COM x 36SEG（COM4=SEG32，COM5=SEG33，COM6=SEG34，COM7=SEG35）

扩展 MAP 图



	模式6x34																																
LCD_MEM6	Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		seg33 com1	seg33 com0							seg32 com5	seg32 com4							seg32 com3	seg32 com2							seg32 com1	seg32 com0						
LCD_MEM7	Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																		seg33 com5	seg33 com4							seg33 com3	seg33 com2						
	模式4x36																																
LCD_MEM7	Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		seg35 com3	seg34 com3	seg33 com3	seg32 com3					seg35 com2	seg34 com2	seg33 com2	seg32 com2					seg35 com1	seg34 com1	seg33 com1	seg32 com1					seg35 com0	seg34 com0	seg33 com0	seg32 com0				

4) 0x02000028~0x0200002c   Reserved

5) 0x02000030   lcd_fcr   default:0x00000000 R/W

Bit 31~10:Reserved

Bit 9~4: LCD frame max count

Bit3~2: LCD frame frequnce pre-scale

“00” : 1 “01” :1/2 “10” :1/4 “11”:1/8

Bit1:Frame counter interrupt enable

Bit0:Frame counter enable

6) 0x02000034 lcd_fcst   default:0x00000000   R

Bit31~3: Reserved

Bit2:   Wake up state flag

Bit1:   Power down state flag

Bit0:   Frame counter flag

2.2 0x03000000~0x03ffffff   SRAM 16K

0x03000000~0x03000fff   sram depth: 4K width:32-bit 总共 16K bytes

0x03001000~0x03ffffff   Reserved

2.3 x01000000~0x01ffffff   usb controller

USB 有 4 个 endpoint, 是固定设置， 不能改变。设置如下：

- 第一个 是 control， 其它 3 个是 IN
- 最大的 Packet 是 8 个 byte
- 第一个 endpoint 有一个输出 FIFO，一个输入 FIFO，各 8 bytes
- 其他的 endpoint 有一个 IN FIFO，各 8 bytes

- 软件把 IN FIFO 写满或写一部分， 等空了（中断或标志查询）再写
- 如果发小于 8 个数据，把数据写到 FIFO 的前半部分，最后把最末尾的指针地址锁到配置为位里，当 IN token 一来，就从第一数据一直发送到这个指针

  这是写 endpoint 0 的例子，只发 4 个数据。

```
WR1:        @lock fifo
        ldr r0,=0x01000008
        ldr r1,=1
        str r1,[r0]
          @write 4 bytes data
        ldr r0,=0x010000C0
        ldr r1,=0x03020100
        str r1,[r0]
        @set last byte position
        ldr r1,=0x3
        str r1,[r0,#8]
        @unlock
        ldr r0,=0x01000008
        ldr r1,=0
         str r1,[r0]
```

- 软件等 OUT FIFO 满了（中断或标志查询），可以把数据读出， 做判断，等的时候可以做别的事。OUT 还有一个非空标志，软件可以判断有数据在 FIFO 里，就读，这主要是为小于 8 个数据的包做处理。

软件只需要等到收到 Package， 读出 package 和一些状态位， 对 package 进行判断， 然后把响应的 package 写到对应的 FIFO 里。

下面 是第一个 USB configure register, 读的时候可以读出 USB 状态， 定义如下：

USB Configure Register 1（0x01000000）可能要多写几次			
Bit number	name	Defaults	描述
0	Enable PHY/FIFO0 IN empty	0	写 1：使能 USB PHY，写 0：power down PHY; 读： 1 是 endpoint 0 IN FIFO empty
1	USB speed/ FIFO0 OUT FULL	1	写 1: full speed, 写 0: low speed; 读： 1 是 endpoint 0 OUT FIFO full
2	PHY suspend/ FIFO1 IN empty	0	写 1: suspend PHY, 读： 1 是 endpoint 1 IN FIFO empty
3	DM pull up/ FIFO0 OUT not empty	0	写 1: enable DM 内部 pull up 读： 1 是 endpoint 0 OUT FIFO has data
4	Resume/ FIFO2 IN empty	0	写 1：在 suspend 后，resume USB,需要手动清零； 读： 1 是 endpoint 2 IN FIFO empty
5	Soft reset	0	写 1：复位整个 USB，一定要写 0 才能放开；
6	Pull Down/ FIFO3 IN empty	1	写 1： pull down DP/DM; 读： 1 是 endpoint 3 IN FIFO empty

7	Reserved	0	
8	Stall EN0	0	写 1: Stall endpoint 0
9	Stall EN1	0	写 1: Stall endpoint 1
10	Stall EN2	0	写 1: Stall endpoint 2
11	Stall EN3	0	写 1: Stall endpoint 3
12	DM Pull up	0	1: 使能 DM 上拉
13	DP Pull up	0	1: 使能 DP 上拉
14	Toggle0	0	写 1: 翻转 endpoint 0 data0/data1
15	Toggle1	0	写 1: 翻转 endpoint 1 data0/data1
16	Toggle2	0	写 1: 翻转 endpoint 2 data0/data1
17	Toggle3	0	写 1: 翻转 endpoint 3 data0/data1

USB Configure Register 2（0x01000008）			
Bit number	name	Defaults	描述
0	LOCK FIFO	0	写 1：要是 USB 要求 FIFO，并且本 register[5：4] 也指向这个 endpoint，就回 NAK，USB 不会对 FIFO 读/ 写，这样防止 FIFO 同时读/写产生错误。
1	Empty pkg	0	写 1：并本 register[0] 为 1，要是 USB 要求 FIFO，并且本 register[5：4] 也指向这个 endpoint,就会回一个空包。要是本 register[2]设为 1，自动清 0.
2	Clrar empty pkg	1	写 1， empty pkg 会在回空包后自动清 0.
3	Reserved		
5：4	Lock addr	0	要 lock 或发空包的 enppont，和本 register，bit 0，1 配合使用

以下是状态 Register，软件可以读出 USB 的状态，定义如下：

USB Status Register 1（0x01000080）			
Bit number	name	Defaults	描述
0	Setup	0	1 是指收到的 package 是 Setup
1	Softreset	0	1 是指收到了 softreset 指令
2	USBreset	1	1 是 USB bus 在 reset 状态，读时清 0
3	suspend	0	1 是 USB bus 在 supend 状态
4	ACK	0	是收到了 ACK，读时清 0
5	SOF	0	是收到 SOF （star of frame），读时清 0
7:6	reserved	0	
8	NAK0	0	Endpoint 0 有 NAK（IN 没有 data 在 FIFO 里或 OUT 包 FIFIO 里还有 data）， 读时清 0
9	NAK1	0	Endpoint 1 有 NAK（IN 没有 data 在 FIFO 里），读时清 0
10	NAK2	0	Endpoint 2 有 NAK（IN 没有 data 在 FIFO 里），读时清 0

11	NAK3	0	Endpoint 3 有 NAK（IN 没有 data 在 FIFO 里），读时清 0
12	DM_status		直接读取 DM 口状态
13	DP_status		直接读取 DP 口状态

USB Configure Register 2（0x01000081，一定 byte 读）			
Bit number	name	Defaults	描述
2:0	Alternate Interface	只读	Set_Interface 命令设的 Alternate Interface number
4:3	Interface	只读	Set_Interface 命令 设的 Interface number
6:5	Configure	只读	Set Configure 命令设的 configure value
7	Not Used	0	

USB Configure Register 3（0x01000082，一定 byte 读）			
Bit number	name	Defaults	描述
10：0	Time Stamp	只读	当验测到 SOF 时，Bit 0 to 10 of Time stamp

FIFO data register, 软件通过这些 register 来读写 data fifo

USB Data Register 1（0x010000C0, 读写）			
Bit number	name	Defaults	描述
7：0	Data0	0	Endpoint 0 FIFO, byte 0
15：8	Data1	0	Endpoint 0 FIFO, byte 1
23：16	Data2	0	Endpoint 0 FIFO, byte 2
31：23	Data3	0	Endpoint 0 FIFO, byte 3

USB Data Register 2（0x010000C4，只写）			
Bit number	name	Defaults	描述
7：0	Data4	0	Endpoint 0 FIFO, byte 4
15：8	Data5	0	Endpoint 0 FIFO, byte 5
23：16	Data6	0	Endpoint 0 FIFO, byte 6
31：23	Data7	0	Endpoint 0 FIFO, byte 7

USB Data Register 3（0x010000D0, 只写）			
Bit number	name	Defaults	描述
7：0	Data0	0	Endpoint 1 FIFO, byte 0
15：8	Data1	0	Endpoint 1 FIFO, byte 1
23：16	Data2	0	Endpoint 1 FIFO, byte 2
31：23	Data3	0	Endpoint 1 FIFO, byte 3

USB Data Register 4（0x010000D4，只写）			
Bit number	name	Defaults	描述

7：0	Data4	0	Endpoint 1 FIFO, byte 4
15：8	Data5	0	Endpoint 1 FIFO, byte 5
23：16	Data6	0	Endpoint 1 FIFO, byte 6
31：23	Data7	0	Endpoint 1 FIFO, byte 7

USB Data Register 5（0x010000E0,　只写）			
Bit number	name	Defaults	描述
7：0	Data0	0	Endpoint 2 FIFO, byte 0
15：8	Data1	0	Endpoint 2 FIFO, byte 1
23：16	Data2	0	Endpoint 2 FIFO, byte 2
31：23	Data3	0	Endpoint 2 FIFO, byte 3

USB Data Register 6（0x010000C4，只写）			
Bit number	name	Defaults	描述
7：0	Data4	0	Endpoint 2 FIFO, byte 4
15：8	Data5	0	Endpoint 2 FIFO, byte 5
23：16	Data6	0	Endpoint 2 FIFO, byte 6
31：23	Data7	0	Endpoint 2 FIFO, byte 7

USB Data Register 7（0x010000F0,　只写）			
Bit number	name	Defaults	描述
7：0	Data0	0	Endpoint 3 FIFO, byte 0
15：8	Data1	0	Endpoint 3 FIFO, byte 1
23：16	Data2	0	Endpoint 3 FIFO, byte 2
31：23	Data3	0	Endpoint 3 FIFO, byte 3

USB Data Register 8（0x010000F4，只写）			
Bit number	name	Defaults	描述
7：0	Data4	0	Endpoint 3 FIFO, byte 4
15：8	Data5	0	Endpoint 3 FIFO, byte 5
23：16	Data6	0	Endpoint 3 FIFO, byte 6
31：23	Data7	0	Endpoint 3 FIFO, byte 7

USB Data Register 9（0x010000C8,　只写）			
Bit number	name	Defaults	描述
2：0	Ptr	0	OUT FIFO 指针，可以用来判断有多少有效 data 在 FIFO 里

USB 中断

所有 USB 的 Full 和 Empty 状态产生一个 USB 总中断， 但每个 USB 中断源可以单独使能。

总中断是 IRQ6。要使能 USB 中断，要往 0xE000E100,写 0x40
USB 里的中断源有

- Endpoint 0-3. IN FIFIO 空
- Endpoint 0， OUT FIFO 满。
- Endpoint 0， OUT FIFO 非空。
- Endpoint 0-3 有 NAK
- USB Host reset
- 收到 setup 包
- SOF
- ACK
- Suspend

USB Configure Register 2（0x01000004）			
Bit number	name	Defaults	描述
0	Empty 0 EN	0	1：使能 Endpoint 0 IN FIFO 空中断
1	Empty 1 EN	0	1：使能 Endpoint 1 IN FIFO 空中断
2	Empty 2 EN	0	1：使能 Endpoint 2 IN FIFO 空中断
3	Empty 3 EN	0	1：使能 Endpoint 3 IN FIFO 空中断
4	Full 0 EN	0	1：使能 Endpoint 0 OUT FIFO 满中断
5	Not empty 0	0	1: 使能 Endpoint 0 OUT FIFO 非空中断
7:6	Reserved	0	0
8	NACK EN 0	0	1：使能 endpoint 0 NACK 中断
9	NACK EN 1	0	1：使能 endpoint 1 NACK 中断
10	NACK EN 2	0	1：使能 endpoint 2 NACK 中断
11	NACK EN 3	0	1：使能 endpoint 3 NACK 中断
12	Setup EN	0	1: 使能 setup 中断
13	USB reset	0	1: 使能 USB reset 中断
14	suspend	0	1: 使能 suspend 中断
15	OUT ACK	0	1: 使能 OUT 包 ACK 中断
16	IN ACK	0	1: 使能 IN 包 ACK 中断
17	Sof	0	1: 使能 sof 中断

USB 强迫 Data0/Data1 翻转
如果应用需要强行把某个 endpoint data0/data1 的包换成 data1/data0，把 0x01000000 的对应的 13 到 16 位设为 1.

2.4 0x05000000~0x05ffffff pwm controller

1)0x05000000 width 32bit pwm1 duty cycle RW
Bit31~12: Reserved
Bit11: Reserved
Bit10: Reserved
Bit9~0: duty_cycle_num

2)0x05000004 width 32bit pwm2 dyty cycle RW
Bit31~12: Reserved
Bit11: Reserved
Bit10: Reserved
Bit9~0: duty_cycle_num

3)0x05000008 width 32bit pwm3 & pwm5 dyty cycle RW
Bit31~26: Reserved
Bit25~16: pwm5_duty_cycle_num (only write)
Bit15~10: Reserved
Bit9~0: pwm3_duty_cycle_num

4)0x0500000C width 32bit pwm4 &pwm6 duty cycle RW
Bit31~26: Reserved
Bit25~16: pwm6_duty_cycle_num (only write)
Bit15~10: Reserved
Bit9~0: pwm4_duty_cycle_num

5)0x05000010 width 32bit pwm1 max count RW
Bit31~12: Reserved
Bit11: Reserved
Bit10: Reserved
Bit9~0: max_count_num

6)0x05000014 width 32bit pwm2 max count RW
Bit31~12: Reserved
Bit11: Reserved
Bit10: Reserved
Bit9~0: max_count_num

7)0x05000018 width 32bit pwm3 & pwm5 max count RW
Bit31~26: Reserved
Bit25~16: pwm5_max_count_num (only write)
Bit15~10: Reserved
Bit9~0: pwm3_max_count_num

6)0x0500001c width 32bit pwm4 &pwm6 max count RW
Bit31~26: Reserved
Bit25~16: pwm6_max_count_num (only write)
Bit15~10: Reserved
Bit9~0: max_count_num

7)0x05000020 width 32bit control register RW
Bit31~23: Reserved
Bit22: gpio56 pull up enable
Bit21: pwm6 count enable
Bit20: pwm5 count enable
Bit18~16: dead zone3 count select 设置 5,6 死区长度

Bit15 :Reserved
Bit14~12: dead zone2 count select 设置 3,4 死区长度
Bit11 :Reserved
Bit10~8 : dead zone1 count select 设置 1,2 死区长度
Bit17 : Reserved
Bit6: dead zone enalbe 3 =1 pwm5,pwm6 相同带死区
Bit5: dead zone enalbe 2 =1 pwm3,pwm4 输出相同带死区
Bit4: dead zone enable 1 =1 pwm1,pwm2 输出相同带死区
Bit3 : pwm4 count enable
Bit2: pwm 3 count enable
Bit1: pwm 2 count enable
Bit0: pwm 1 count enable

2.5 0x04000000~0x04ffffff System &PLL&ADC controller

1)0x04000000 width：32bit WR
Bit1~0:在内部 ROM 中运行程序时 CPU 执行指令等待时间（适应 ROM 最大运行速度）

2)0x04000004 width: 32bit WR
Bit0: BOD 输出控制信号 默认为 0。一般使用不用改写
Bit1：BOD 输出选择信号 默认为 0。一般使用不用改写
Bit2~4：BOD 低电压检测电压选择 (检测 3.3V 输入脚电压)
Bit2=0;Bit3=0;Bit4=0 2.0V
Bit2=0;Bit3=0;Bit4=1 2.3V
Bit2=0;Bit3=1;Bit4=0 2.4V
Bit2=0;Bit3=1;Bit4=1 2.7V
Bit2=1;Bit3=0;Bit4=0 3.0V
Bit2=1;Bit3=0;Bit4=1 3.3V
Bit2=1;Bit3=1;Bit4=0 3.6V
Bit2=1;Bit3=1;Bit4=1 4.1V
Bit5: Reserved
Bit6: BOD 功能使能信号 默认为 0 0 使能，1 关闭
Bit7: Reserved
Bit8: 切换 interrupt vector 到 SRAM 地址

3) 0x04000008 width 32bit WR
Bit0~1: RTC 时钟分频（在 32KHz Cystal clock 基础上）
Bit0=0,Bit1=0 ÷8192
Bit0=0,Bit1=1 ÷16384
Bit0=1,Bit1=1 ÷32768
Bit0=1,Bit1=0 ÷65536
Bit2: 切换 32KHz Cystal clock 到 32K 分频器输出
1：切换 0：保持
Bit3~4 调节 32K 分频器输出频率
Bit4=0,Bit3=0 分频器输出 16KHz
Bit4=0,Bit3=1 分频器输出 8KHz

Bit4=1,Bit3=0 分频器输出 4KHz
Bit4=1,Bit3=1 分频器输出 2KHz
Bit5~31: Reserved

4)0x0400000c width:32bit W default:0x2030D3F
Bit31~0:SysTick calibration register value

5)0x04000010 width: 32bit WR
见 ADC 具体操作寄存器

6)0x04000014 width:32bit WR
见 ADC 具体操作寄存器

7)0x04000018 width:32bit WR default 0x0
Bit1~0：PLL_N PLL 倍频控制
Bit1=0,Bit0=0 PLL 频率输出 Fsys=64MHz
Bit1=0,Bit0=1 PLL 频率输出 Fsys=96MHz
Bit1=1,Bit0=0 PLL 频率输出 Fsys=128MHz
Bit1=1,Bit0=1 PLL 频率无输出
Bit2~31: Reserved

8)0x0400001c width:32bit WR default 0x3
Bit5~0: PLL_M 系统时钟分频控制

$$\text{Fsys} = \frac{\text{Fpll}}{\text{PLL M+1}}$$

注：在配置 PLL_M 和 PLL_N 配置系统时钟时，为了保证时钟变化时程序工作的稳定性，建议先把系统时钟切换到 Cystal 32K，然后再改变原由 PLL 产生的主系统时钟，改变完成并且等时钟稳定之后，再切换回原来的主系统时钟。

默认设置频率：Fsys＝64MHz/(M+1)=16MHz

9)0x04000020 width:32bit WR
Bit0: 休眠系统时钟停止使能 默认为 1
1:休眠时系统时钟停止；0:休眠时系统时钟不停止
如果系统时钟不停止休眠功耗会更大，建议休眠时停掉时钟
Bit1~31:Reserved

10)0x04000024 width:32bit WR
见 ADC 详细操作寄存器

11)0x04000028 width:32bit WR
Reserved

12)0x0400002c width:32bit WR

Reserved

13)0x04000030 width:32bit W default：0x20010

Bit31~19:Reserved

Bit18: Reserved

Bit17：保持为 1

Bit16：VLCD 内部驱动电压使能信号，LCD 使用时必须打开

Bit15~11： Reserved

Bit10~8： VLCD 电压选择信号

111: 3.0V

110: 3.2V

101: 3.4V

100: 3.6V

011:3.8V

010:4.0V

001:4.2V

000:4.4V

Bit7~5： Reserved

Bit4： Reserved

Bit3： charge pump 使能信号

Bit2： VLCD 选择 1/4 BIAS 1：使能 0：关闭

Bit1： VLCD 选择 1/3 BIAS 1：使能 0：关闭

Bit0： VLCD 选择 1/2 BIAS 1：使能 0：关闭

在驱动 1/4 BIAS 屏时 Bit0 和 Bit2 要同时开启

14) 0x04000034 width:32bit W

Bit31~2: Reserved

Bit1: OP1 enable

Bit0: OP2 enable

15) 0x04000038 width:32bit WR

Bit0: start ADC convertion W

ADC convertion done R

16)0x0400003C width 32bit WR

Bit0:清除 ADC convertion done 1：清除 0：释放 WRITE

Bit23:0 ADC 转换结果 READ

17)0x04000040 width 32bit W

Bit3~0:写 EEPROM 和写 MTP 切换

在写 MTP 之前需要在 Bit3~0 中写 0x5；

在写 EEPROM 之前需要在 Bit3~0 中写 0xA；

否则任何往 MTP 和 EEPROM 里面任何写动作都无效。

写完之后要恢复成 0

2.6 x06000000~0x06ffffff sound

1) 0x06000000 width:32bit W
Bit31~0 : data_fifo0

2) 0x06000004 width:32bit W
Bit31~0 : data_fifo1

3) 0x06000008 width:32bit W
Bit31~16 : Reserved
Bit15: en_pcm 为 1 使能 ADPCM 默认为 0
Bit14 : en_unsign 为 1 使能硬件转换无符号数，默认为 1
在 ADPCM 时必须为 1。
在 PCM 时根据软件处理方法选择。
Bit13~0 : sample rate cnt

4) 0x0600000c width : 32bit R
Bit31~4 : Reserved
Bit3 : sound interrupt flag (sample rate)
Bit2 : Reserved
Bit1~0 : fifo pointer

5) 0x06000010 width :32bit W
Bit31~1:Reserved
Bit0: clear interrupt 写 1 清中断

6）0x0600014 width : 32bit W
Bit31~21:Reserved
Bit20 : load start value
Bit19~16: first count
Bit15~0: pcm start value 在 ADPCM 模式下使用

使用方法：data_fifo0,data_fifo1 ,sound 模块自动发送 fifo0，fifo1，按顺序播放，fifo_pointer 可以读出正在发送哪个 fifo，为了写的动作的不影响正在发送数据的 fifo，所以设置两个 fifo，设置采样率，sample rate cnt X clock_sound 周期 = 1 一个采样周期。
Sound interrupt flag 显示两个 sample 数据发送完成，可以填写 fifo 了

例如：clk_sys = 64MHZ clk_sound=clk_sys/4 = 16MHZ 声音播放采样率要求 8KHZ

则 sample rate count = clk_sound / sample_rate = 16M /8K = 2000

2.7 0x10000000~0x1000000f clock generator

1) 0x10000000 width: 16bit RW default:0x4010
Bit15: sound clock enable
Bit14:switch pll & Crystal clock 1=pll clock 0=crystal clock
Bit13: WDT clock enable
Bit12: RTC clock enable

Bit11:USB clock enable
Bit10:UART clock enable
Bit9:I2C clock enable
Bit8:SPI clock enable
Bit7:ADC clock enable
Bit6:Reserved
Bit5:charge pump clock enable
Bit4:APB clock enable
Bit3:Timer clock enable
Bit2:LCD clock enable
Bit1:PWM clock enable
Bit0:Switch Crystal clock and PLL clock in sleep state

2)0x10000002 width: 16bit RW default: 0x8
Bit15~0 : pwm clock divider
Note: 如果想得到除 1 的除频效果，则需将除数设置为 0，除 1 无效。

3)0x10000004 width:16bit RW default:0x8
Bit15~0: spi clock ,i2c clock divider

4)0x10000006 width:16bit RW default: 0x8
Bit15~0: timer clock divider

5)0x10000008 width:16bit RW default:0x4
Bit15~0:APB bus clock divider (同时也作为 uart clock）

6)0x1000000a width:16bit RW default:0x28
Bit15~0: lcd clock divider

7)0x1000000c width:16bit RW default:0x1
Bit15~0 AHB bus clock divider

8)0x1000000e width:16bit RW default:0x1
Bit15~0 sound clock divider

9)0x10000010 width: 16bit RW default:0x1
Bit15~0 charge pump clock divider (source clock 32K)

10)0x10000012 width: 16bit RW default:0x103f
Bit15~14:Reserved
Bit13: WDT reset enable
Bit12:LDO full load
Bit11:Reserved
Bit10~8: wakeup time select
“000” 22ms
“001” 11ms
“010” 6ms
“011” 3ms
“100” 1.5ms

“101” 700us
“110” 350us
“111” 65us

Bit6: Brownout Detect reset enable
Bit5: Brownout Detector select
Bit4: PLL sleep enable
Bit3: software reset ADC
Bit2: software reset USB
Bit1: software reset RTC
Bit 0 :保持为 1，不能写 0

2.8 x10000100~0x100001ff GPIO

1)0x10000100 width :32bit RW default:0xffff9fff
Bit31~0 : GPIO31~0 out register

2)0x10000104 width: 32bit RW default:0x10fffff
Bit31~25: Reserved
Bit24~0: GPIO56~32 outregister

3)0x10000110 width :32bit RW default: 32’hffff9fff
Bit31~0: GPIO 31~0 output enable ‘1’ :input ‘0’:output

4)0x10000114 width :32bit RW default: 0x10fffff
Bit31~25: Reserved
Bit24~0: GPIO56~32 output enable ‘1’ : input ‘0’:output

5)0x10000120 width:32 bit RW default:0x00000018
Bit31~0: GPIO31~0 function select ‘1’:special function ‘0’: gpio function

GPIO2~0	gpio
GPIO3	: SWD_CLK
GPIO4	: SWD_DAT
GPIO5	: SPI_DATAO
GPIO6	: SPI_CLK
GPIO7	:SPI_DATAI
GPIO8	:SPI_CS
GPIO9	:UART TXD
GPIO10	:UART RXD
GPIO11	:I2C SCL
GPIO12	:I2C SDA
GPIO13	:PWM1
GPIO14	:PWM2
GPIO15	:AUDIO +
GPIO16	:AUDIO -/ Timer_clk_in
GPIO17~22	： COM0~5
GPIO23	: COM0/ SPI_DATA_OUT
GPIO24~31	: SEG0~SEG7

6)0x10000124 width:32 bit RW default:0x0
Bit31~25: Reserved

Bit24~0: GPIO56~32 function select ‘1’:special function ‘0’: gpio function

GPIO32~45 :	SEG8~SEG21	
GPIO46:	SEG22/SPI_DATA_IN	
GPIO47:	SEG23/ SPI_CS	
GPIO48:	SEG24/ UART_RXD	
GPIO49:	SEG25/ UART_TXD	
GPIO50:	SEG26/ I2C_SCL	
GPIO51:	SEG27/ I2C_SDA	
GPIO52:	SEG28/PWM5	
GPIO53:	SEG29/PWM6	
GPIO54 :	SEG30/PWM3	
	GPIO54----IOMODE=0	misc_reg6=0
	PWM3 ---- IOMODE=0	misc_reg6=1
	SEG30 ---- IOMODE=1	misc_reg6=X
GPIO55 :	SEG31/PWM4	
	GPIO55----IOMODE=0	misc_reg7=0
	PWM4 ---- IOMODE=0	misc_reg7=1
	SEG31 ---- IOMODE=1	misc_reg7=X
GPIO56 ：	COM7/ SPI_CLK	

7)0x10000130 width:32bit

W default:0x0

Bit31~0 : Reserved

R default:0x0

Bit31~16：interrupt status15~0 注意：读 status 就会清中断

Bit15~0 ： Reserved

8)0x1000134 width:32bit RW

Bit31~16 : GPIO15~0 falling interrupt enable

Bit15~0 : GPIO15~0 rising interrupt enable

9)0x10000138 width :32bit RW default:0x0

Bit31~16: Reserved

Bit15~0: gpio interrupt mask 分别使能 15~0 gpio 中断

10)0x10000140 width: 32bit RW default:0x00000004

Bit31~24: gpio7~0 wake up enable

Bit23~8: gpio15~0 debounce enable

Bit7~0: misc_reg7~0

misc_reg7~6 : PWM3，PWM4 mode

misc_reg5~3 : To be define

misc_reg2: Must be 1

misc_reg1~0 :To be define

11)0x10000144 width: 32bit RW

Bit31~8 : Reserved

Bit7~0 : gpio15~8 wake up enable

Note: 能够唤醒的 GPIO 为 gpio0~15，并且分为高电平唤醒和低电平唤醒。
唤醒方式，高电平唤醒---在休眠之前唤醒脚要保持为低电平，在休眠之后在唤醒脚上给出高电平主控会被唤醒
低电平唤醒---在休眠之前唤醒脚要保持为高电平，在休眠之后在唤醒脚上给出低电平主控会被唤醒
高电平唤醒脚：gpio0，gpio1，gpio2
低电平唤醒脚：gpio3~15 （gpio8 没有睡眠唤醒功能，中断功能正常）

12)0x10000150 width: 32bit RW default 0x0

Bit15~0 : gpio15~0 pull up enable
Note：GPIO9 GPIO10 没有上拉电阻,相应 bit 请保持为 0

13)0x10000160 width: 32bit RW default:0x0
bit31~24 Reserved
bit23: 使能 gpio48 复用为 UART_RXD
bit23=1 &gpio_mode23=0
bit22: 使能 gpio49 复用为 UART_TXD
bit22=1 &gpio_mode22=0
bit21: 使能 gpio51 复用为 I2C_SDA
bit21=1&gpio_mode21=0
bit20: 使能 gpio50 复用为 I2C_SCL
bit20=1&gpio_mode20=0
bit19： 使能 gpio47 复用为 SPI_CS
bit19=1& gpio_mode19=0
bit18： 使能 gpio46 复用为 SPI_DATA_IN
bit18=1& gpio_mode18=0
bit17： 使能 gpio56 复用为 SPI_CLK
bit17=1& gpio_mode17=0
bit16： 使能 gpio23 复用为 SPI_DATA_OUT
bit16=1& gpio_mode16=0
bit15~0 : ADC IO enable 1 : ADC mode GPIO 输入模式无效
0 :GPIO 模式时，IO 与内部 ADC 断开
Note: bit15~0 可读写 对应 gpio15~0 配置
bit31~16 只可写不能读回

10.0x10000300~0x100003ff SPI

Table 1: Memory Map of SPI controller

Name	Address Offset	Width	Description

CTRLR0	0x0	16 bits	Control Register 0 **Reset Value:** Configuration Dependent for some bit fields
CTRLR1	0x04	16 bits	Control Register 1 **Reset Value:** 0x0
SSIENR	0x08	1 bit	SSI Enable Register **Reset Value:** 0x0
MWCR	0x0C	3 bits	Microwire Control Register **Reset Value:** 0x0
SER	0x10	See Description	Slave Enable Register **Width:** *SSI_NUM_SLAVES* **Reset Value:** 0x0
BAUDR	0x14	16 bits	Baud Rate Select **Reset Value:** 0x0
TXFTLR	0x18	TX_ABW	Transmit FIFO Threshold Level Reset Value: 0x0
RXFTLR	0x1C	RX_ABW	Receive FIFO Threshold Level Reset Value: 0x0
TXFLR	0x20	See Description	Transmit FIFO Level Register Width: TX_ABW+1
RXFLR	0x24	See Description	Receive FIFO Level Register Width: RX_ABW+1
SR	0x28	7 bits	Status Register Reset Value: 0x6
IMR	0x2C	See Description	Interrupt Mask Register Width: 6 bits: when SSI_IS_MASTER =1)
ISR	0x30	6 bits	Interrupt Status Register Reset Value: 0x0
RISR	0x34	6 bits	Raw Interrupt Status Register Reset Value: 0x0
TXOICR	0x38	1 bit	Transmit FIFO Overflow Interrupt Clear Register Reset Value: 0x0
RXOICR	0x3C	1 bit	Receive FIFO Overflow Interrupt Clear Register Reset Value: 0x0
RXUICR	0x40	1 bit	Receive FIFO Underflow Interrupt Clear Register Reset Value: 0x0
MSTICR	0x44	1 bit	Multi-Master Interrupt Clear Register Reset Value: 0x0
ICR	0x48	1 bit	Interrupt Clear Register Reset Value: 0x0

DR	0x60 - 0x9C	16 bits	Data Register Reset Value: 0x0

CTRLR0

- **Name:** Control Register 0
- **Size:** 16 bits
- **Address Offset:** 0x0
- **Read/write access:** read/write

This register controls the serial data transfer. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.

Bits	Name	R/W	Description
15:12	CFS	R/W	**Control Frame Size**. Selects the length of the control word for the Microwire frame format. For the field decode, refer to Table 9 on page 112. **Reset Value:** 0x0
11	SRL	R/W	**Shift Register Loop**. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. 0 – Normal Mode Operation 1 – Test Mode Operation **Reset Value:** 0x0

10	SLV_OE	R/W	**Slave Output Enable**. Relevant only when the DW_apb_ssi is configured as a serial-slave device. When configured as a serial master, this bit field has no functionality. This bit enables or disables the setting of the ssi_oe_n output from the DW_apb_ssi serial slave. When SLV_OE = 1, the ssi_oe_n output can never be active. When the ssi_oe_n output controls the tri-state buffer on the txd output from the slave, a high impedance state is always present on the slave txd output when SLV_OE = 1. This is useful when the master transmits in broadcast mode (master transmits data to all slave devices). Only one slave may respond with data on the master rxd line. This bit is enabled after reset and must be disabled by software (when broadcast mode is used), if you do not want this device to respond with data. 0 –- Slave txd is enabled 1 –- Slave txd is disabled **Reset Value:** 0x0
9:8	TMOD	R/W	Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All data in the transmit FIFO is considered control data in this mode. This transfer mode is only valid when the DW_apb_ssi is configured as a master device. 00 –- Transmit & Receive 01 –- Transmit Only 10 –- Receive Only 11 –- EEPROM Read Reset Value: 0x0
7	SCPOL	R/W	Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the DW_apb_ssi master is not actively transferring data on the serial bus. 0 – Inactive state of serial clock is low 1 – Inactive state of serial clock is high Reset Value: SSI_DFLT_SCPOL
6	SCPH	R/W	Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. 0: Serial clock toggles in middle of first data bit 1: Serial clock toggles at start of first data bit Reset Value: SSI_DFLT_SCPOL

5:4	FRF	R/W	Frame Format. Selects which serial protocol transfers the data. 00 –- Motorola SPI 01 –- Texas Instruments SSP 10 –- National Semiconductors Microwire 11 –- Reserved Reset Value: SSI_DFLT_SCPOL
3:0	DFS	R/W	Data Frame Size. Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. For the field decode, refer to Table 8 on page 112. Reset Value: 0x7

Table 8: DFS Decode

DFS Value	Description
0000	Reserved – undefined operation
0001	Reserved – undefined operation
0010	Reserved – undefined operation
0011	4-bit serial data transfer
0100	5-bit serial data transfer
0101	6-bit serial data transfer
0110	7-bit serial data transfer
0111	8-bit serial data transfer
1000	9-bit serial data transfer
1001	10-bit serial data transfer
1010	11-bit serial data transfer
1011	12-bit serial data transfer
1100	13-bit serial data transfer
1101	14-bit serial data transfer
1110	15-bit serial data transfer
1111	16-bit serial data transfer

Table 9: CFS Decode

CFS Value	Description
0000	1-bit control word
0001	2-bit control word
0010	3-bit control word
0011	4-bit control word
0100	5-bit control word
0101	6-bit control word
0110	7-bit control word
0111	8-bit control word

1000	9-bit control word
1001	10-bit control word
1010	11-bit control word
1011	12-bit control word
1100	13-bit control word
1101	14-bit control word
1110	15-bit control word
1111	16-bit control word

CTRLR1

- **Name:** Control Register 1
- **Size:** 16 bits
- **Address Offset:** 0x04
- **Read/write access:** read/write

This register exists only when the DW_apb_ssi is configured as a master device. When the DW_apb_ssi is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. Control register 1 controls the end of serial transfers when in receive-only mode. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.

Bits	Name	R/W	Description
15:0	NDF	R/W	**Number of Data Frames**. When TMOD = 10, this register field sets the number of data frames to be continuously received by the DW_apb_ssi. The DW_apb_ssi continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer. When the DW_apb_ssi is configured as a serial slave, the transfer continues for as long as the slave is selected. Therefore, this register serves no purpose and is not present when the DW_apb_ssi is configured as a serial slave. **Reset Value:** 0x0

SSIENR

- **Name:** SSI Enable Register
- **Size:** 1 bit
- **Address Offset:** 0x08
- **Read/write access:** read/write

This register enables and disables the DW_apb_ssi.

Bits	Name	R/W	Description
0	SSI_EN	R/W	**SSI Enable**. Enables and disables all DW_apb_ssi operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the DW_apb_ssi control registers when enabled. When disabled, the ssi_sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system. **Reset Value:** 0x0

MWCR

- **Name:** Microwire Control Register
- **Size:** 3 bits
- **Address Offset:** 0x0C
- **Read/write access:** read/write

This register controls the direction of the data word for the half-duplex Microwire serial protocol. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.

Bits	Name	R/W	Description
2	MHS	R/W	**Microwire Handshaking**. Relevant only when the DW_apb_ssi is configured as a serial-master device. When configured as a serial slave, this bit field has no functionality. Used to enable and disable the "busy/ready" handshaking interface for the Microwire protocol. When enabled, the DW_apb_ssi checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. 0: handshaking interface is disabled 1: handshaking interface is enabled **Reset Value:** 0x0
1	MDD	R/W	**Microwire Control**. Defines the direction of the data word when the Microwire serial protocol is used. When this bit is set to 0, the data word is received by the DW_apb_ssi MacroCell from the external serial device. When this bit is set to 1, the data word is transmitted from the DW_apb_ssi MacroCell to the external serial device. **Reset Value:** 0x0
0	MWMOD	R/W	**Microwire Transfer Mode**. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. 0 – non-sequential transfer 1 – sequential transfer **Reset Value:** 0x0

BAUDR

- **Name:** Baud Rate Select
- **Size:** 16 bits
- **Address Offset:** 0x14
- **Read/write access:** read/write

This register is valid only when the DW_apb_ssi is configured as a master device. When the DW_apb_ssi is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. The register derives the frequency of the serial clock that regulates the data transfer. The 16-bit field in this register defines the ssi_clk divider value. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.

Bits	Name	R/W	Description

15:0	SCKDV	R/W	**SSI Clock Divider**. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation:
			$F_{sclk\_out} = \frac{F_{ssi\_clk}}{SCKDV}$
			where SCKDV is any even value between 2 and 65534. For example:
			`for Fssi_clk = 3.6864MHz and SCKDV =2`
			`Fsclk_out = 3.6864/2 = 1.8432MHz`
			Reset Value: 0x0

TXFTLR

- **Name:** Transmit FIFO Threshold Level
- **Size:** *TX_ABW*
- **Address Offset:** 0x18
- **Read/write access:** read/write

This register controls the threshold value for the transmit FIFO memory. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.

Bits	Name	R/W	Description
31:TX_ABW+1	Reserved	N/A	Reserved
TX_ABW-1:0	TFT	R/W	**Transmit FIFO Threshold**. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered. For field decode, refer to Table 10. **Reset Value:** 0x0

Table 10: TFT Decode

TFT Value	Description
0000_0000	ssi_txe_intr is asserted when 0 data entries are present in transmit FIFO
0000_0001	ssi_txe_intr is asserted when 1 or less data entry is present in transmit FIFO
0000_0010	ssi_txe_intr is asserted when 2 or less data entries are present in transmit FIFO
0000_0011	ssi_txe_intr is asserted when 3 or less data entries are present in transmit FIFO
: :	: :
1111_1100	ssi_txe_intr is asserted when 252 or less data entries are present in transmit FIFO

1111_1101	ssi_txe_intr is asserted when 253 or less data entries are present in transmit FIFO
1111_1110	ssi_txe_intr is asserted when 254 or less data entries are present in transmit FIFO
1111_1111	ssi_txe_intr is asserted when 255 or less data entries are present in transmit FIFO

RXFTLR

- **Name:** Receive FIFO Threshold Level
- **Size:** *RX_ABW*
- **Address Offset:** 0x1C
- **Read/write access:** read/write

This register controls the threshold value for the receive FIFO memory. It is impossible to write to this register when the DW_apb_ssi is enabled. The DW_apb_ssi is enabled and disabled by writing to the SSIENR register.

Bits	Name	R/W	Reset	Description
31:RX_ABW+1	Reserved	N/A	N/A	Reserved
RX_ABW-1:0	RFT	R/W		**Receive FIFO Threshold**. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered. For field decode, refer to Table 11. **Reset Value:** 0x0

Table 11: RFT

RFT Value	Description
0000_0000	ssi_rxf_intr is asserted when 1 or more data entry is present in receive FIFO
0000_0001	ssi_rxf_intr is asserted when 2 or more data entries are present in receive FIFO
0000_0010	ssi_rxf_intr is asserted when 3 or more data entries are present in receive FIFO
0000_0011	ssi_rxf_intr is asserted when 4 or more data entries are present in receive FIFO
:	:
:	:
1111_1100	ssi_rxf_intr is asserted when 253 or more data entries are present in receive FIFO
1111_1101	ssi_rxf_intr is asserted when 254 or more data entries are present in receive FIFO
1111_1110	ssi_rxf_intr is asserted when 255 or more data entries are present in receive FIFO

1111_1111	ssi_rxf_intr is asserted when 256 data entries are present in receive FIFO

TXFLR

- **Name:** Transmit FIFO Level Register
- **Size:** *TX_ABW* + 1
- **Address Offset:** 0x20
- **Read/write access:** read-only

This register contains the number of valid data entries in the transmit FIFO memory.

Bits	Name	R/W	Description
31:TX_ABW+1	Reserved	N/A	Reserved
TX_ABW:0	TXTFL	R	**Transmit FIFO Level**. Contains the number of valid data entries in the transmit FIFO. **Reset Value:** 0x0

RXFLR

- **Name:** Receive FIFO Level Register
- **Size:** *RX_ABW* + 1
- **Address Offset:** 0x24
- **Read/write access:** read-only

This register contains the number of valid data entries in the transmit FIFO memory. This register can be read at any time.

Bits	Name	R/W	Description
31:RX_ABW+1	Reserved	N/A	Reserved
RX_ABW:0	RXTFL	R	**Receive FIFO Level**. Contains the number of valid data entries in the receive FIFO. **Reset Value:** 0x0

SR

- **Name:** Status Register
- **Size:** 7 bits
- **Address Offset:** 0x28
- **Read/write access:** read-only

This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The status register may be read at any time. None of the bits in this register request an interrupt.

Bits	Name	R/W	Description
6	DCOL	R	**Data Collision Error**. Relevant only when the DW_apb_ssi is configured as a master device. This bit is set if the DW_apb_ssi master is actively transmitting when another master selects this device as a slave. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. 0 – No error 1 – Transmit data collision error **Reset Value:** 0x0
5	TXE	R	**Transmission Error**. Set if the transmit FIFO is empty when a transfer is started. This bit can be set only when the DW_apb_ssi is configured as a slave device. Data from the previous transmission is resent on the txd line. This bit is cleared when read. 0 – No error 1 – Transmission error **Reset Value:** 0x0
4	RFF	R	**Receive FIFO Full**. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full **Reset Value:** 0x0
3	RFNE	R	**Receive FIFO Not Empty**. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty **Reset Value:** 0x0
2	TFE	R	**Transmit FIFO Empty.** When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty **Reset Value:** 0x1
1	TFNF	R	**Transmit FIFO Not Full.** Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full **Reset Value:** 0x1
0	BUSY	R	**SSI Busy Flag.** When set, indicates that a serial transfer is in progress; when cleared indicates that the DW_apb_ssi is idle or disabled. 0 – DW_apb_ssi is idle or disabled 1 – DW_apb_ssi is actively transferring data **Reset Value:** 0x0

IMR

- **Name:** Interrupt Mask Register
- **Size:**
  6 bits: when SSI_IS_MASTER =1)

5 bits: when SSI_IS_MASTER = 0)

- **Address Offset:** 0x2C
- **Read/write access:** read/write

This read/write register masks or enables all interrupts generated by the DW_apb_ssi. When the DW_apb_ssi is configured as a slave device, the MSTIM bit field is not present. This changes the reset value from 0x3F for serial-master configurations to 0x1F for serial-slave configurations.

Bits	Name	R/W	Description
31:6	Reserved	N/A	Reserved
5	MSTIM	R/W	**Multi-Master Contention Interrupt Mask.** This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. 0 – ssi_mst_intr interrupt is masked 1 – ssi_mst_intr interrupt is not masked **Reset Value:** 0x1
4	RXFIM	R/W	**Receive FIFO Full Interrupt Mask** 0 – ssi_rxf_intr interrupt is masked 1 – ssi_rxf_intr interrupt is not masked **Reset Value:** 0x1
3	RXOIM	R/W	**Receive FIFO Overflow Interrupt Mask** 0 – ssi_rxo_intr interrupt is masked 1 – ssi_rxo_intr interrupt is not masked **Reset Value:** 0x1
2	RXUIM	R/W	**Receive FIFO Underflow Interrupt Mask** 0 – ssi_rxu_intr interrupt is masked 1 – ssi_rxu_intr interrupt is not masked **Reset Value:** 0x1
1	TXOIM	RW	**Transmit FIFO Overflow Interrupt Mask** 0 – ssi_txo_intr interrupt is masked 1 – ssi_txo_intr interrupt is not masked **Reset Value:** 0x1
0	TXEIM	RW	**Transmit FIFO Empty Interrupt Mask** 0 – ssi_txe_intr interrupt is masked 1 – ssi_txe_intr interrupt is not masked **Reset Value:** 0x1

ISR

- **Name:** Interrupt Status Register
- **Size:** 6 bits
- **Address Offset:** 0x30
- **Read/write access:** read-only

This register reports the status of the DW_apb_ssi interrupts after they have been masked.

Bits	Name	R/W	Description
31:6	Reserved	N/A	Reserved

5	MSTIS	R	**Multi-Master Contention Interrupt Status**. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. 0 = ssi_mst_intr interrupt not active after masking 1 = ssi_mst_intr interrupt is active after masking **Reset Value:** 0x0
4	RXFIS	R	**Receive FIFO Full Interrupt Status** 0 = ssi_rxf_intr interrupt is not active after masking 1 = ssi_rxf_intr interrupt is full after masking **Reset Value:** 0x0
3	RXOIS	R	**Receive FIFO Overflow Interrupt Status** 0 = ssi_rxo_intr interrupt is not active after masking 1 = ssi_rxo_intr interrupt is active after masking **Reset Value:** 0x0
2	RXUIS	R	**Receive FIFO Underflow Interrupt Status** 0 = ssi_rxu_intr interrupt is not active after masking 1 = ssi_rxu_intr interrupt is active after masking **Reset Value:** 0x0
1	TXOIS	R	**Transmit FIFO Overflow Interrupt Status** 0 = ssi_txo_intr interrupt is not active after masking 1 = ssi_txo_intr interrupt is active after masking **Reset Value:** 0x0
0	TXEIS	R	**Transmit FIFO Empty Interrupt Status** 0 = ssi_txe_intr interrupt is not active after masking 1 = ssi_txe_intr interrupt is active after masking **Reset Value:** 0x0

RISR

- **Name:** Raw Interrupt Status Register
- **Size:** 32 bits
- **Address Offset:** 0x34
- **Read/write access:** read-only

This read-only register reports the status of the DW_apb_ssi interrupts prior to masking.

Bits	Name	R/W	Description
31:6	Reserved	N/A	Reserved
5	MSTIR	R	**Multi-Master Contention Raw Interrupt Status.** This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. 0 = ssi_mst_intr interrupt is not active prior to masking 1 = ssi_mst_intr interrupt is active prior masking **Reset Value:** 0x0
4	RXFIR	R	**Receive FIFO Full Raw Interrupt Status** 0 = ssi_rxf_intr interrupt is not active prior to masking 1 = ssi_rxf_intr interrupt is active prior to masking **Reset Value:** 0x0
3	RXOIR	R	**Receive FIFO Overflow Raw Interrupt Status** 0 = ssi_rxo_intr interrupt is not active prior to masking 1 = ssi_rxo_intr interrupt is active prior masking **Reset Value:** 0x0

2	RXUIR	R	**Receive FIFO Underflow Raw Interrupt Status** 0 = ssi_rxu_intr interrupt is not active prior to masking 1 = ssi_rxu_intr interrupt is active prior to masking **Reset Value:** 0x0
1	TXOIR	R	**Transmit FIFO Overflow Raw Interrupt Status** 0 = ssi_txo_intr interrupt is not active prior to masking 1 = ssi_txo_intr interrupt is active prior masking **Reset Value:** 0x0
0	TXEIR	R	**Transmit FIFO Empty Raw Interrupt Status** 0 = ssi_txe_intr interrupt is not active prior to masking 1 = ssi_txe_intr interrupt is active prior masking **Reset Value:** 0x0

TXOICR

- **Name:** Transmit FIFO Overflow Interrupt Clear Register
- **Size:** 1 bit
- **Address Offset:** 0x38
- **Read/write access:** read-only

Bits	Name	R/W	Description
0	TXOICR	R	**Clear Transmit FIFO Overflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect. **Reset Value:** 0x0

RXOICR

- **Name:** Receive FIFO Overflow Interrupt Clear Register
- **Size:** 1 bit
- **Address Offset:** 0x3C
- **Read/write access:** read-only

Bits	Name	R/W	Description
0	RXOICR	R	**Clear Receive FIFO Overflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect. **Reset Value:** 0x0

RXUICR

- **Name:** Receive FIFO Underflow Interrupt Clear Register
- **Size:** 1 bit
- **Address Offset:** 0x40
- **Read/write access:** read-only

Bits	Name	R/W	Description

0	RXUICR	R	**Clear Receive FIFO Underflow Interrupt**. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect. **Reset Value:** 0x0

MSTICR

- **Name:** Multi-Master Interrupt Clear Register
- **Size:** 1 bit
- **Address Offset:** 0x44
- **Read/write access:** read-only

Bits	Name	R/W	Description
0	MSTICR	R	**Clear Multi-Master Contention Interrupt**. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect. **Reset Value:** 0x0

ICR

- **Name:** Interrupt Clear Register
- **Size:** 1 bit
- **Address Offset:** 0x48
- **Read/write access:** read-only

Bits	Name	R/W	Description
0	ICR	R	**Clear Interrupts**. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect. **Reset Value:** 0x0

DR

- **Name:** Data Register
- **Size:** 16 bits
- **Address Offset:** 0x60 to 0x9C
- **Read/write access:** read/write

The DW_apb_ssi data register is a 16-bit read/write buffer for the transmit/receive FIFOs. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0.

Note

The DR register in the DW_apb_ssi occupies sixteen 32-bit locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the DW_apb_ssi are not addressable.

Bits	Name	R/W	Description

15:0	DR	R/W	**Data Register**. When writing to this register, you must right-justify the data. Read data are automatically right-justified.
			Read = Receive FIFO buffer
			Write = Transmit FIFO buffer
			Reset Value: 0x0

11.0x10000400~0x100004ff I2C

Table 2: Memory Map of i2c controller

Name	Address Offset	Width	R/W	Description
IC_CON	0x00	7 bits	R/W or R-only on bit 4	I^{2}C Control **Reset Value for I2C_DYNAMIC_TAR_UPDATE = 0** *IC_SLAVE_DISABLE, IC_RESTART_EN, IC_10BITADDR_MASTER, IC_10BITADDR_SLAVE, IC_MAX_SPEED_MODE, IC_MASTER_MODE* **Reset Value for I2C_DYNAMIC_TAR_UPDATE = 1** *IC_SLAVE_DISABLE, IC_RESTART_EN, IC_10BITADDR_MASTER, IC_10BITADDR_SLAVE* (read only), *IC_MAX_SPEED_MODE, IC_MASTER_MODE*
IC_TAR	0x04	12 or 13 bits	R/W	I^{2}C Target Address Width: 13, if I2C_DYNAMIC_TAR_UPDATE = 1 12, if I2C_DYNAMIC_TAR_UPDATE = 0 **Reset Value for I2C_DYNAMIC_TAR_UPDATE = 0** *IC_DEFAULT_TAR_SLAVE_ADDR* **Reset Value for I2C_DYNAMIC_TAR_UPDATE = 1** *IC_10BITADDR_MASTER, IC_DEFAULT_TAR_SLAVE_ADDR*
IC_SAR	0x08	10 bits	R/W	I^{2}C Slave Address **Reset Value:** *IC_DEFAULT_SLAVE_ADDR*
IC_HS_MADDR	0x0C	3 bits	R/W	I^{2}C HS Master Mode Code Address **Reset Value:** *IC_HS_MASTER_CODE*
IC_DATA_CMD	0x10	9 (writes) 8 (reads)	R/W	I^{2}C RX/TX Data Buffer and Command **Reset Value:** 0x0
IC_SS_SCL_HCNT	0x14	16 bits	R/W	Standard speed I^{2}C Clock SCL High Count **Reset Value:** *IC_SS_SCL_HIGH_COUNT*
IC_SS_SCL_LCNT	0x18	16 bits	R/W	Standard speed I^{2}C Clock SCL Low Count **Reset Value:** *IC_SS_SCL_LOW_COUNT*

IC_FS_SCL_HCNT	0x1C	16 bits	R/W	Fast speed I^{2}C Clock SCL High Count **Reset Value:** *IC_FS_SCL_HIGH_COUNT*
IC_FS_SCL_LCNT	0x20	16 bits	R/W	Fast speed I2C Clock SCL Low Count Reset Value: IC_FS_SCL_LOW_COUNT
IC_HS_SCL_HCNT	0x24	16 bits	R/W	High speed I2C Clock SCL High Count Reset Value: IC_HS_SCL_HIGH_COUNT
IC_HS_SCL_LCNT	0x28	16 bits	R/W	High speed I2C Clock SCL Low Count Reset Value: IC_HS_SCL_LOW_COUNT
IC_INTR_STAT	0x2C	12 bits	R	I2C Interrupt Status Reset Value: 0x0
IC_INTR_MASK	0x30	12 bits	R/W	I2C Interrupt Mask Reset Value: 0x0
IC_RAW_INTR_STAT	0x34	12 bits	R	I2C Raw Interrupt Status Reset Value: 0x0
IC_RX_TL	0x38	8 bits	R/W	I2C Receive FIFO Threshold Reset Value: IC_RX_TL
IC_TX_TL	0x3C	8 bits	R/W	I2C Transmit FIFO Threshold Reset Value: IC_TX_TL
IC_CLR_INTR	0x40	1 bit	R	Clear Combined and Individual Interrupts Reset Value: 0x0
IC_CLR_RX_UNDER	0x44	1 bit	R	Clear RX_UNDER Interrupt Reset Value: 0x0
IC_CLR_RX_OVER	0x48	1 bit	R	Clear RX_OVER Interrupt Reset Value: 0x0
IC_CLR_TX_OVER	0x4C	1 bit	R	Clear TX_OVER Interrupt Reset Value: 0x0
IC_CLR_RD_REQ	0x50	1 bit	R	Clear RD_REQ Interrupt Reset Value: 0x0
IC_CLR_TX_ABRT	0x54	1 bit	R	Clear TX_ABRT Interrupt Reset Value: 0x0
IC_CLR_RX_DONE	0x58	1 bit	R	Clear RX_DONE Interrupt Reset Value: 0x0
IC_CLR_ACTIVITY	0x5c	1 bit	R	Clear ACTIVITY Interrupt Reset Value: 0x0
IC_CLR_STOP_DET	0x60	1 bit	R	Clear STOP_DET Interrupt Reset Value: 0x0
IC_CLR_START_DET	0x64	1 bit	R	Clear START_DET Interrupt Reset Value: 0x0
IC_CLR_GEN_CALL	0x68	1 bit	R	Clear GEN_CALL Interrupt Reset Value: 0x0

IC_ENABLE	0x6C	1 bit	R/W	I2C Enable Reset Value: 0x0
IC_STATUS	0x70	7 bits	R	I2C Status register Reset Value: 0x6
IC_TXFLR	0x74	TX_ABW+1	R	Transmit FIFO Level Register Reset Value: 0x0
IC_RXFLR	0x78	RX_ABW+1	R	Receive FIFO Level Register Reset Value: 0x0
Reserved	0x7C			
IC_TX_ABRT_SOURCE	0x80	16 bits	R/W	I2C Transmit Abort Status Register Reset Value: 0x0

IC_CON

- **Name:** I^{2}C Control Register 0
- **Size:** 7 bits
- **Address Offset:** 0x00
- **Read/Write Access:**
  If I2C_DYNAMIC_TAR_UPDATE = 0, all bits are Read/Write
  If I2C_DYNAMIC_TAR_UPDATE = 1, bit 4 is Read-only

This register can be written only when the DW_apb_i2c is disabled, which corresponds to the IC_ENABLE register being set to 0. Writes at other times have no effect.

Bits	Name	R/W	Description
15:7	Reserved	N/A	Reserved.
6	*IC_SLAVE_DISABLE*	R/W	This bit controls whether I^{2}C has its slave disabled after reset. The slave can be disabled by programming a '1' into IC_CON[6]. B y default the slave is enabled. 0: slave is enabled 1: slave is disabled **Reset:** *IC_SLAVE_DISABLE*
5	*IC_RESTART_EN*	R/W	Determines whether restart conditions may be sent when acting as a master. Some older slaves do not support handling restart conditions. Restart conditions are used in several DW_apb_i2c operations. Disabling a restart does not allow the master to perform the following functions: • send multiple bytes per transfer (split) • change direction within a transfer (split) • send a start byte • perform any high-speed mode operation • perform combined format transfers in 7- or 10-bit addressing modes (split for 7 bit) • perform a read operation with a 10-bit address Split operations are broken down into multiple DW_apb_i2c transfers with a stop and start condition in between. The other operations are not performed at all and result in setting TX_ABRT. Reset: IC_RESTART_EN

4	*IC_10BITADDR_MASTER or IC_10BITADDR_MASTER_ rd_only*	R/W or R	If I2C_DYNAMIC_TAR_UPDATE = 0, this bit is named IC_10BITADDR_MASTER and controls whether the DW_apb_i2c starts its transfers in 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing If I2C_DYNAMIC_TAR_UPDATE = 1, the function of this bit is handled by bit 12 of IC_TAR, and this bit becomes a read-only copy called IC_10BITADDR_MASTER_rd_only. Dependencies: If I2C_DYNAMIC_TAR_UPDATE = 1, then this bit is read-only. If I2C_DYNAMIC_TAR_UPDATE = 0, then this bit can be read or write. Reset: IC_10BITADDR_MASTER
3	*IC_10BITADDR_SLAVE*	R/W	When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses. 0: 7-bit addressing. The DW_apb_i2c ignores transactions which involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. 1: 10-bit addressing. The DW_apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register. Reset: IC_10BITADDR_SLAVE

2:1	*SPEED*	R/W	Controls at which speed the DW_apb_i2c operates: 0: illegal; writing a 0 results in setting SPEED to IC_MAX_SPEED_MODE 1: standard mode (100 kbit/s) 2: fast mode (400 kbit/s) 3: high speed mode (3.4 Mbit/s) If the DW_apb_i2c is configured for fast or standard mode (1 or 2) and a value of 2 or 3 is written, then IC_MAX_SPEED_MODE is stored. These bits correspond to a 2-bit register. If an APB write is performed to these bits such that the data is decimal 2 or 3, then these would change the maximum speed mode. Hardware prevents this and writes in the value of IC_MAX_SPEED_MODE instead. **Reset:** IC_MAX_SPEED_MODE
0	*MASTER_MODE*	R/W	This bit controls whether the DW_apb_i2c master is enabled or not. The slave is always enabled. 0: master disabled 1: master enabled **Reset:** IC_MASTER_MODE

IC_TAR

- **Name:** I^{2}C Target Address Register
- **Size:** 12 bits or 13 bits; 13 bits only when I2C_DYNAMIC_TAR_UPDATE = 1
- **Address Offset:** 0x04
- **Read/Write Access:** Read/Write

If I2C_DYNAMIC_TAR_UPDATE = 0, this register is 12 bits wide, and bits 15:12 are reserved. However, if I2C_DYNAMIC_TAR_UPDATE = 1, then the register becomes 13 bits wide. Under these conditions, bit 12 and bits

9 through 0 can be dynamically updated as long as the following are true:

- MST_ACTIVITY must be IDLE; that is, IC_STATUS[5] = 0
- Transmit FIFO Completely Empty must occur; that is, IC_STATUS[2] = 0

Bits	Name	R/W	Description
15:13	Reserved	N/A	Reserved.
12	IC_10BITADDR_ MASTER	R/W	Exists in this register only if I2C_DYNAMIC_TAR_UPDATE = 1. Under these circumstances, this bit controls whether the DW_apb_i2c starts its transfers in 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing **Dependencies:** This bit exists in this register only if I2C_DYNAMIC_TAR_UPDATE = 1. **Reset:** IC_10BITADDR_MASTER
11	*SPECIAL*	R/W	This bit indicates whether software would like to perform a general call or start byte I^{2}C command. 0: ignore bit 10 *GC_OR_START* and use IC_TAR normally 1: perform special I^{2}C command as specified in *GC_OR_START* bit **Dependencies:** This bit is writable only when bit 0 of IC_ENABLE = 0. **Reset:** 0x0
10	*GC_OR_START*	R/W	If bit 11 SPECIAL is set to 1, then this bit indicates whether a general call or start byte command is to be performed by the DW_apb_i2c. 0: General Call Address – after issuing a general call, only writes may be performed. Attempting to issue a read command results in setting TX_ABRT. The DW_apb_i2c remains in general call mode until the *SPECIAL* bit value is cleared. 1: Start Byte **Dependencies:** This bit is writable only when bit 0 of IC_ENABLE = 0. **Reset:** 0x0
9:0	*IC_TAR*	R/W	This is the target address for any master transactions. **Reset:** *IC_DEFAULT_TAR_SLAVE_ADDR*, which indicates loopback mode

IC_SAR

- **Name:** I^2**C** Slave Address Register
- **Size:** 10 bits
- **Address Offset:** 0x08
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description
15:10	Reserved	N/A	Reserved.
9:0	IC_SAR	R/W	The IC_SAR holds the slave address when the I^{2}C is operating as a slave. IC_SAR holds the slave address to which the DW_apb_i2c responds. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I^{2}C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. **Reset:** *IC_DEFAULT_SLAVE_ADDR*

IC_HS_MADDR

- **Name:** I²C HS Master Mode Code Address Register
- **Size:** 3 bits
- **Address Offset:** 0x0c
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description
15:3	Reserved	N/A	Reserved.
2:0	*IC_HS_MAR*	R/W	IC_HS_MAR holds the value of the I²C HS mode master code. Valid values are from 0-7. This register goes away and becomes read-only returning 0's if IC_MAX_SPEED_MODE != high. This register can be written only when the I²C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. **Reset:** *IC_HS_MASTER_CODE*

IC_DATA_CMD

- **Name:** I²C RX/TX Data Buffer and Command Register
- **Size:** 9 bits (writes)
  8 bits (reads)
- **Address Offset:** 0x10
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description
15:9	Reserved	N/A	Reserved
8	*CMD*	R/W	This bit controls whether a read or a write is performed. 1 = Read. 0 = Write. For reads, the lower 8 (DAT) bits are ignored by the DW_apb_i2c. However, if the APB_DATA_WIDTH is 8, this "dummy" write is still required as there is coherency in this register. Reading this bit returns 0. Attempting to perform a read operation after a general call command has been sent results in TX_ABRT unless the *SPECIAL* bit in the IC_TAR register has been cleared. If this bit is written to a '1' after receiving RD_REQ, then a TX_ABRT occurs. **Reset:** 0x10
7:0	*DAT*	R/W	This register contains the data to be transmitted or received on the I²C bus. Read these bits to read out the data received on the I²C interface. Write these bits to send data out on the I²C interface. **Reset:** 0x0

IC_SS_SCL_HCNT

- **Name:** Standard Speed I²C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x14
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description
15:0	*IC_SS_SCL_ HCNT*	R/W[1]	This register must be set before any I^2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. The table below shows some sample IC_SS_HCNT calculations. This register can be written only when the I^2C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. **Reset**: *IC_SS_SCL_HIGH_COUNT*
[1] Read-only if IC_HC_COUNT_VALUES = 1.			

I^2C Data Rate (kbps)	ic_clk$_{freq}$ (MHz)	SCL High required min (us)	H_CNT (HEX/decimal)	SCL High Time (us)
100	2	4	0008/8	4.00
100	6.6	4	001B/27	4.09
100	10	4	0028/40	4.00
100	75	4	012C/300	4.00
100	100	4	0190/400	4.00
100	125	4	01F4/500	4.00
100	1000	4	0FA0/4000	4.00

IC_SS_SCL_LCNT

- **Name:** Standard Speed I^2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x18
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description

15:0	*IC_SS_SCL_LCNT*	R/W[1]	This register must be set before any I^{2}C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. The table below shows some sample IC_SS_LCNT calculations. This register can be written only when the I^{2}C interface is disabled which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. **Reset**: *IC_SS_SCL_LOW_COUNT*

[1] Read-only if IC_HC_COUNT_VALUES = 1.

I^{2}C Data Rate (kbps)	ic_clk$_{freq}$ (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
100	2	4.7	000A/10	5.00
100	6.6	4.7	0020/32	4.85
100	10	4.7	002F/47	4.70
100	75	4.7	0161/353	4.71
100	100	4.7	01D6/470	4.70
100	125	4.7	024C/588	4.70
100	1000	4	125C/4700	4.70

IC_FS_SCL_HCNT

- **Name:** Fast Speed I^{2}C Clock SCL High Count Register
- **Size:** 16 bits
- **Address Offset:** 0x1c
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description

15:0	*IC_FS_SCL_HCNT*	R/W[1]	This register must be set before any I^2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_HCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I^2C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. **Reset:** *IC_FS_SCL_HIGH_COUNT*

[1] Read-only if IC_HC_COUNT_VALUES = 1.

I^2C Data Rate (kbps)	ic_clk_{freq} (MHz)	SCL High required min (us)	H_CNT (HEX/Decimal)	SCL High Time (us)
400	10	0.6	0006/6	0.60
400	25	0.6	000F/15	0.60
400	50	0.6	001E/30	0.60
400	75	0.6	002D/45	0.60
400	100	0.6	003C/60	0.60
400	125	0.6	004B/75	0.60
400	1000	0.6	0258/600	0.60
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC_FS_SCL_LCNT

- **Name:** Fast Speed I^2C Clock SCL Low Count Register
- **Size:** 16 bits
- **Address Offset:** 0x20
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description

15:0	*IC_FS_SCL_LCNT*	R/W[1]	This register must be set before any I²C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. The table below shows some sample IC_FS_SCL_LCNT calculations. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I²C interface is disabled, which corresponds to the *IC_ENABLE* register being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. **Reset:** *IC_FS_SCL*_LOW_COUNT
[1] Read-only if IC_HC_COUNT_VALUES = 1.			

I²C Data Rate (kbps)	ic_clk$_{freq}$ (MHz)	SCL Low required min (us)	L_CNT (HEX/Decimal)	SCL Low Time (us)
400	10	1.3	000D/13	1.30
400	25	1.3	0021/33	1.32
400	50	1.3	0041/65	1.30
400	75	1.3	0062/98	1.31
400	100	1.3	0082/130	1.30
400	125	1.3	00A3/163	1.30
400	1000	1.3	0514/1300	1.30
100 (through IC_MAX_SPEED_MODE = standard)	N/A	N/A	disabled	N/A

IC_INTR_STAT

- **Name:** I²C Interrupt Status Register
- **Size:** 12 bits
- **Address Offset:** 0x2C
- **Read/Write Access:** Read

Each bit in this register has a corresponding mask bit in the IC_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the IC_RAW_INTR_STAT register.

Bits	Name	R/W	Description

15:12	Reserved	N/A	Reserved.
11	*R_GEN_CALL*	R	See "IC_RAW_INTR_STAT" for a detailed description of these bits. **Reset:** 0x0
10	*R_START_DET*	R	
9	*R_STOP_DET*	R	
8	*R_ACTIVITY*	R	
7	*R_RX_DONE*	R	
6	*R_TX_ABRT*	R	
5	*R_RD_REQ*	R	
4	*R_TX_EMPTY*	R	
3	*R_TX_OVER*	R	
2	*R_RX_FULL*	R	
1	*R_RX_OVER*	R	
0	*R_RX_UNDER*	R	

IC_INTR_MASK

- **Name:** I^{2}C Interrupt Mask Register
- **Size:** 12 bits
- **Address Offset:** 0x30
- **Read/Write Access:** Read/Write

These bits mask their corresponding interrupt status bits. They are active high; a value of 0 prevents a bit from generating an interrupt.

Bits	Name	R/W	Description
15:12	Reserved	N/A	Reserved.
11	*M_GEN_CALL*	R/W	Masks this bit in the IC_INTR_STAT register. **Reset:** 0x0
10	*M_START_DET*	R/W	
9	*M_STOP_DET*	R/W	
8	*M_ACTIVITY*	R/W	
7	*M_RX_DONE*	R/W	
6	*M_TX_ABRT*	R/W	
5	*M_RD_REQ*	R/W	
4	*M_TX_EMPTY*	R/W	
3	*M_TX_OVER*	R/W	
2	*M_RX_FULL*	R/W	
1	*M_RX_OVER*	R/W	
0	*M_RX_UNDER*	R/W	

IC_RAW_INTR_STAT

- **Name:** I2C Raw Interpol Status Register
- **Size:** 12 bits
- **Address Offset:** 0x34
- **Read/Write Access:** Read/Write

Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the DW_apb_i2c.

Bits	Name	R/W	Description
15:12	Reserved	N/A	Reserved.
11	*GEN_CALL*	R	Indicates that a general call request was received. The DW_apb_i2c stores the received data in the RX buffer. **Reset:** 0x0
10	*START_DET*	R	Indicates whether a start condition has occurred on the I2C interface. **Reset:** 0x0
9	*STOP_DET*	R	Indicates whether a stop condition has occurred on the I2C interface **Reset:** 0x0
8	*ACTIVITY*	R	This bit captures DW_apb_i2c activity and stays set until it is cleared, regardless of the DW_apb_i2c going idle. **Reset:** 0x0
7	*RX_DONE*	R	When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done. **Reset:** 0x0

Bits	Name	R/W	Description
6	*TX_ABRT*	R	In general, this bit is set to 1 when the DW_apb_i2c acting as a master is unable to complete a command that the processor has sent. The conditions that set TX_ABRT are: • no slave acknowledges after the address is sent. • the addressed slave does not acknowledge a byte of data. • arbitration is lost. • attempting to send a master command when configured only to be a slave. • IC_RESTART_EN bit in the IC_CON register is set to 0 (restart condition disabled), and the processor attempts to issue an I^{2}C function that is impossible to perform without using restart conditions. • high-speed master code is acknowledge. • start byte is acknowledged. • general call address is not acknowledged. • when a read request interrupt occurs and the processor has previously placed data in the TX buffer that has not been transmitted yet. This data could have been intended to service a multi-byte RD_REQ that ended up having fewer numbers of bytes requested. Or, if IC_RESTART_EN is disabled and the I^{2}C loses control of the bus between transfers and is then accessed as a slave-transmitter. • if a read command is issued after a general call command has been issued. Disabling the I^{2}C reverts it back to normal operation. • if the processor attempts to issue read command before a RD_REQ is serviced. Anytime this bit is set, the contents of the transmit and receive buffers are flushed. **Reset:** 0x0

5	*RD_REQ*	R	This bit is set to 1 when the DW_apb_i2c is acting as slave and another I^{2}C master is attempting to read data from our module. The DW_apb_i2c holds the I^{2}C bus in waiting state (SCL=0) until this interrupt is serviced. The processor must acknowledge this interrupt and then write the requested data to the IC_DATA_CMD register. **Reset:** 0x0
4	*TX_EMPTY*	R	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register. It is automatically cleared by hardware when buffer level goes above the threshold. **Reset:** 0x0
3	*TX_OVER*	R	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I^{2}C command by writing to the IC_DATA_CMD register. **Reset:** 0x0
2	*RX_FULL*	R	Set when the transmit buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. **Reset:** 0x0
1	*RX_OVER*	R	Set if the receive buffer was completely filled to IC_RX_BUFFER_DEPTH and more data arrived. That data is lost. **Reset:** 0x0

Bits	Name	R/W	Description
0	*RX_UNDER*	R	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. **Reset:** 0x0

IC_RX_TL

- **Name:** I^{2}C Receive FIFO Threshold Register
- **Size:** 8bits
- **Address Offset:** 0x38
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description
15:8	Reserved	N/A	Reserved.
7:0	*RX_TL*	R/W	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt. The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries. **Reset:** *IC_RX_TL*

IC_TX_TL

- **Name:** I²C Transmit FIFO Threshold Register
- **Size:** 8 bits
- **Address Offset:** 0x3c
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description
15:8	Reserved	N/A	Reserved.
7:0	*TX_TL*	R/W	Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt. The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries. **Reset**: *IC_TX_TL*

IC_CLR_INTR

- **Name:** Clear Combined and Individual Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x40
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_INTR*	R	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register. **Reset:** 0x0

IC_CLR_RX_UNDER

- **Name:** Clear RX_UNDER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x44
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_RX_UNDER*	R	Read this register to clear the *RX_UNDER* interrupt. **Reset:** 0x0

IC_CLR_RX_OVER

- **Name:** Clear RX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x48
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_RX_OVER*	R	Read this register to clear the *RX_OVER* interrupt. **Reset:** 0x0

IC_CLR_TX_OVER

- **Name:** Clear TX_OVER Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x4c
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_TX_OVER*	R	Read this register to clear the *TX_OVER* interrupt. **Reset:** 0x0

IC_CLR_RD_REQ

- **Name:** Clear RD_REQ Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x50
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_RD_REQ*	R	Read this register to clear the *RD_REQ* interrupt. **Reset:** 0x0

IC_CLR_TX_ABRT

- **Name:** Clear TX_ABRT Interrupt Register

- **Size:** 1 bit
- **Address Offset:** 0x54
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_TX_ABRT*	R	Read this register to clear the *TX_ABRT* interrupt, and the IC_TX_ABRT_SOURCE register. **Reset**: 0x0

IC_CLR_RX_DONE

- **Name:** Clear RX_DONE Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x58
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_RX_DONE*	R	Read this register to clear the *RX_DONE* interrupt. **Reset:** 0x0

IC_CLR_ACTIVITY

- **Name:** ACTIVITY Status Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x5c
- **Read/Write Access:** Read

Bits	Name	R.W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_ACTIVITY*	R	Read this register to get status of the *ACTIVITY* interrupt. It is automatically cleared by hardware. **Reset:** 0x0

IC_CLR_STOP_DET

- **Name:** Clear STOP_DET Interrupt Register

- **Size:** 1 bit
- **Address Offset:** 0x60
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_STOP_DET*	R	Read this register to clear the *STOP_DET* interrupt. **Reset:** 0x0

IC_CLR_START_DET

- **Name:** Clear START_DET Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x64
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_START_DET*	R	Read this register to clear the *START_DET* interrupt. **Reset:** 0x0

IC_CLR_GEN_CALL

- **Name:** Clear GEN_CALL Interrupt Register
- **Size:** 1 bit
- **Address Offset:** 0x68
- **Read/Write Access:** Read

Bits	Name	R/W	Description
15:1	Reserved	N/A	Reserved.
0	*CLR_GEN_CALL*	R	Read this register to clear the *GEN_CALL* interrupt. **Reset:** 0x0

IC_ENABLE

- Name: I^{2}C Enable Register
- **Size:** 1 bit
- **Address Offset:** 0x6c
- **Read/Write Access:** Read/Write

Bits	Name	R/W	Description

15:1	Reserved	N/A	Reserved.
0	*ENABLE*	R/W	Controls whether the DW_apb_i2c is enabled. Writing a 1 enables the DW_apb_i2c, and writing a 0 disables it. Software should not disable the DW_apb_i2c while it is active. The *ACTIVITY* bit can be polled to determine if the DW_apb_i2c is active. If the module was transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module was receiving, the DW_apb_i2c stops the current transfer at the end of the current byte and does not acknowledge the transfer. In systems with asynchronous pclk and ic_clk (IC_CLK_TYPE = 1), there is a two ic_clk delay when enabling or disabling the DW_apb_i2c. **Reset:** 0x0

IC_STATUS

- **Name:** I^{2}C Status Register
- **Size:** 7 bits
- **Address Offset:** 0x70
- **Read/Write Access:** Read

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.

Bits	Name	R/W	Description
6	*SLV_ACTIVITY*	R	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0 – Slave FSM is in IDLE state so the Slave part of DW_apb_i2c is not Active 1 – Slave FSM is not in IDLE state so the Slave part of DW_apb_i2c is Active **Reset:** 0x0
5	*MST_ACTIVITY*	R	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0 – Master FSM is in IDLE state so the Master part of DW_apb_i2c is not Active 1 – Master FSM is not in IDLE state so the Master part of DW_apb_i2c is Active **Reset:** 0x0

4	*RFF*	R	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full **Reset:** 0x0
3	*RFNE*	R	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty **Reset:** 0x0

Bits	Name	R/W	Description
2	*TFE*	R	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty **Reset:** 0x1
1	*TFNF*	R	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full **Reset:** 0x1
0	*ACTIVITY*	R	I2C Activity Status. **Reset:** 0x0

IC_TXFLR

- **Name:** I^{2}C Transmit FIFO Level Register
- **Size:** TX_ABW + 1
- **Address Offset:** 0x74
- **Read/Write Access:** Read

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared when the I^{2}C is disabled, whenever there is a transmit abort, or whenever the Slave Bulk Transfer mode is aborted. It increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

Bits	Name	R/W	Description
31:TX_ABW+1	Reserved	N/A	Reserved
TX_ABW:0	*TXFLR*	R	**Transmit FIFO Level**. Contains the number of valid data entries in the transmit FIFO. **Reset:** 0x0

IC_RXFLR

- **Name:** I^{2}C Receive FIFO Level Register
- **Size:** RX_ABW + 1
- **Address Offset:** 0x78
- **Read/Write Access:** Read

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared when the I^{2}C is disabled or whenever there is a transmit abort. It increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

Bits	Name	R/W	Description
31:RX_ABW+1	Reserved	N/A	Reserved
RX_ABW:0	*RXFLR*	R	**Receive FIFO Level**. Contains the number of valid data entries in the receive FIFO. **Reset:** 0x0

IC_TX_ABRT_SOURCE

- **Name:** I^{2}C Transmit Abort Source Register
- **Size:** 16 bits
- **Address Offset:** 0x80
- **Read/Write Access:** Read/Write

This register has 16 bits that indicate the source of the tx_abrt signal, This register is cleared whenever the processor reads it or when the processor issues a clear signal to all interrupts.

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15	*ABRT_SLVRD_INTX*	R/W	1 = Slave requesting data to TX and the user wrote a read command into the tx_fifo (9th bit is a 1). **Reset:** 0x0
14	*ABRT_SLV_ARBLOST*	R/W	1 = Slave lost the bus while it is transmitting data to a remote master. *IC_TX_ABRT*[12] will be set at the same time. **Reset:** 0x0
13	*ABRT_SLVFLUSH_TXFIFO*	R/W	1 = Slave has received a read command and some data exists in the tx_fifo so the slave issues a *TX_ABRT* to flush old data in tx_fifo. **Reset:** 0x0

Bits	Name	R/W	Description
12	*ARB_LOST*	R/W	1 = Master has lost arbitration, or if *TX_ABRT_SRC*[14] is also set, then the slave transmitter has lost arbitration. **Reset:** 0x0
11	*ARB_MASTER_DIS*	R/W	1 = User attempted to use disabled Master. **Reset:** 0x0
10	*ABRT_10B_RD_NORSTRT*	R/W	1 = The restart is disabled (*IC_RESTART_EN* bit (ic_con[5]) = 0) and the Master sends a read command in 10-bit addressing mode. **Reset:** 0x0
9	*ABRT_SBYTE_NORSTRT*	R/W	1 = The restart is disabled (*IC_RESTART_EN* bit (ic_con[5]) = 0) and the user is trying to send a Start Byte. **Reset:** 0x0
8	*ABRT_HS_NORSTRT*	R/W	1 = The restart is disabled (*IC_RESTART_EN* bit (ic_con[5]) = 0) and the user is trying to use the master to send data in High Speed mode. **Reset:** 0x0
7	*ABRT_SBYTE_ACKDET*	R/W	1 = Master has sent a Start Byte and the Start Byte was acknowledged (wrong behavior). **Reset:** 0x0
6	*ABRT_HS_ACKDET*	R/W	1 = Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). **Reset:** 0x0
5	*ABRT_GCALL_READ*	R/W	1 = Master sent a general call but the user programmed the byte following the G.CALL to be a read from the bus (9th bit is set to 1). **Reset:** 0x0
4	*ABRT_GCALL_NOACK*	R/W	1 = Master sent a general call and no slave on the bus responded with an ack. **Reset:** 0x0

3	*ABRT_TXDATA_NOACK*	R/W	1 = Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). **Reset:** 0x0
2	*ABRT_10ADDR2_NOACK*	R/W	1 = Master is in 10-bit address mode and the 2nd address byte of the 10-bit address was not acknowledged by any slave. **Reset:** 0x0
1	*ABRT_10ADDR1_NOACK*	R/W	1 = Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. **Reset:** 0x0
0	*ABRT_7B_ADDR_NOACK*	R/W	1 = Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. **Reset:** 0x0

12.0x10000500~0x100005ff RTC

Table 3: Memory Map of RTC controller

Name	**Address Offset (Base +)**	**Width**	**R/W**	**Description**
RTC_CCVR	0x00	see Description	R	Current Counter Value Register **Width:** RTC_CNT_WIDTH **Reset Value:** 0x0
RTC_CMR	0x04	see Description	R/W	Counter Match Register **Width:** RTC_CNT_WIDTH **Reset Value: 0x0**
RTC_CLR	0x08	see Description	R/W	Counter Load Register **Width:** RTC_CNT_WIDTH **Reset Value:** 0x0
RTC_CCR	0x0C	see Description	R/W	Counter Control Register **Width:** Maximum of 4 bits; minimum of 2 bits **Reset Value:** 0x0
RTC_STAT	0x10	32 bits	R	Interrupt Status Register **Reset Value:** 0x0
RTC_RSTAT	0x14	32 bits	R	Interrupt Raw Status Register **Reset Value:** 0x0
RTC_EOI	0x18	32 bits	R	End of Interrupt Register **Reset Value:** 0x0

RTC_CCVR

- **Name:** Current Counter Value Register
- **Size:** *RTC_CNT_WIDTH*
- **Address Offset:** 0x00
- **Read/write access:** read-only

Bits	Name	R/W	Description
RTC_CNT_WIDTH–1:0	Current Counter Value	R	When read, this register is the current value of the internal counter. This value always is read coherently. Bits from RTC_CNT_WIDTH to 31 are read as 0 when RTC_CNT_WIDTH is less than 31. **Reset Value:** 0x0

RTC_CMR

- **Name:** Counter Match Register
- **Size:** *RTC_CNT_WIDTH*
- **Address Offset:** 0x04
- **Read/write access:** read/write

Bits	Name	R/W	Description
RTC_CNT_WIDTH–1:0	Counter Match	R/W	Interrupt Match Register. When the internal counter matches this register, an interrupt is generated, provided interrupt generation is enabled. When appropriate, this value is written coherently. Only when all the bytes are written is the register used by the interrupt detection logic. Bits from RTC_CNT_WIDTH and above are read and written as 0 when RTC_CNT_WIDTH is less than 31. **Reset Value:** 0x0

RTC_CCR

- **Name:** Counter Control Register
- **Size:** 32 bits
- **Address Offset:** 0x0C
- **Read/write access:** read/write

Bits	Name	R/W	Description
31:4	N/A	N/A	Reserved and read as 0.
3	rtc_wen	R/W	*Optional*. Allows the user to force the counter to wrap when a match occurs instead of waiting until the maximum count is reached. 0 = Wrap disabled 1 = Wrap enabled This bit is writable only when RTC_WRAP_MODE = 1. **Reset Value:** 0x0

2	rtc_en	R/W	*Optional*. Allows the user to control counting in the counter. 0 = Counter disabled 1 = Counter enabled This bit does not exit if RTC_EN_MODE = 0. Internally, the counter always is enabled. **Reset Value:** 0x0
1	rtc_mask	R/W	*Allows the user to mask interrupt generation.* *0 = Interrupt unmasked* *1 = Interrupt masked* *Reset Value: 0x0*
0	rtc_ien	R/W	*Allows the user to disable interrupt generation.* *0 = Interrupt disabled* *1 = Interrupt enabled* *Reset Value: 0x0*

RTC_STAT

- **Name:** Interrupt Status Register
- **Size:** 32 bits
- **Address Offset:** 0x10
- **Read/write access:** read-only

Bits	Name	R/W	Description
31:1	N/A	N/A	Reserved and read as 0.
0	rtc_stat	R	This register is the masked raw status 0 = Interrupt is inactive 1 = Interrupt is active (regardless of polarity) **Reset Value:** 0x0

RTC_RSTAT

- **Name:** Interrupt Raw Status Register
- **Size:** 32 bits
- **Address Offset:** 0x14
- **Read/write access:** read-only

Bits	Name	R/W	Description
31:1	N/A	N/A	Reserved and read as 0.
0	rtc_rstat	R	0 = Interrupt is inactive 1 = Interrupt is active (regardless of polarity) **Reset Value:** 0x0

RTC_EOI

- **Name:** End of Interrupt Register
- **Size:** 32 bits
- **Address Offset:** 0x18
- **Read/write access:** read-only

Bits	Name	R/W	Description
31:1	N/A	N/A	Reserved and read as 0.
0	rtc_eoi	R	By reading this location, the match interrupt is cleared. Performing read-to-clear on interrupts, the interrupt is cleared at the end of the read. **Reset Value:** 0x0

13.0x10000700~0x100007ff　WDT

WDT 使用 RTC2 作为 WDT 计数文档请参照 RTC1

14.0x10000800~0x100008ff　UART

Name	Address Offset	Width	R/W	Description
RBR	0x00	32 bits	R	Receive Buffer Register **Reset Value:** 0x0 **Dependencies:** LCR[7] bit = 0
THR		32 bits	W	Transmit Holding Register **Reset Value:** 0x0 **Dependencies:** LCR[7] bit = 0
DLL		32 bits	R/W	Divisor Latch (Low) **Reset Value:** 0x0 **Dependencies:** LCR[7] bit = 1
DLH	0x04	32 bits	R/W	Divisor Latch (High) **Reset Value:** 0x0 **Dependencies:** LCR[7] bit = 1
IER		32 bits	R/W	Interrupt Enable Register **Reset Value:** 0x0 **Dependencies:** LCR[7] bit = 0
IIR	0x08	32 bits	R	Interrupt Identification Register **Reset Value:** 0x01
FCR		32 bits	W	FIFO Control Register **Reset Value:** 0x0
LCR	0x0C	32 bits	R/W	Line Control Register **Reset Value:** 0x0
MCR	0x10	32 bits	R/W	Modem Control Register **Reset Value:** 0x0

LSR	0x14	32 bits	R	Line Status Register **Reset Value:** 0x60
MSR	0x18	32 bits	R	Modem Status Register **Reset Value:** 0x0
SCR	0x1C	32 bits	R/W	Scratchpad Register **Reset Value:** 0x0
Reserved	0x20 - 0x2C	–	–	–
SRBR	0x30 - 0x6C	32 bits	R	Shadow Receive Buffer Register Reset Value: 0x0 Dependencies: LCR[7] bit = 0
STHR		32 bits	W	Shadow Transmit Holding Register Reset Value: 0x0
FAR	0x70	32 bit	R/W	FIFO Access Register Reset Value: 0x0
TFR	0x74	32 bits	R	Transmit FIFO Read Reset Value: 0x0
RFW	0x78	32 bits	W	Receive FIFO Write Reset Value: 0x0
USR	0x7C	32 bits	R	UART Status Register Reset Value: 0x6
TFL	0x80	See Description	R	Transmit FIFO Level Width: FIFO_ADDR_WIDTH + 1
RFL	0x84	See Description	R	Receive FIFO Level Width: FIFO_ADDR_WIDTH + 1
SRR	0x88	32 bits	W	Software Reset Register Reset Value: 0x0
SRTS	0x8C	32 bits	R/W	Shadow Request to Send Reset Value: 0x0
SBCR	0x90	32 bits	R/W	Shadow Break Control Register Reset Value: 0x0
SDMAM	0x94	32 bits	R/W	Shadow DMA Mode Reset Value: 0x0
SFE	0x98	32 bits	R/W	Shadow FIFO Enable Reset Value: 0x0
SRT	0x9C	32 bits	R/W	Shadow RCVR Trigger Reset Value: 0x0

STET	0xA0	32 bits	R/W	Shadow TX Empty Trigger Reset Value: 0x0
HTX	0xA4	32 bits	R/W	Halt TX Reset Value: 0x0
DMASA	0xA8	–	W	DMA Software Acknowledge Reset Value: 0x0
–	0xAC - 0xF0	–	–	–
CPR	0xF4	32 bits	R	Component Parameter Register Reset Value: Configuration-dependent
UCV	0xF8	32 bits	R	UART Component Version Reset Value: See the Releases table in the
CTR	0xFC	32 bits	R	Component Type Register Reset Value: 0x44570110

RBR

- **Name:** Receive Buffer Register
- **Size:** 32 bits
- **Address Offset:** 0x00
- **Read/write access:** read-only

Bits	Name	R/W	Description
31:8	Reserved and read as zero		
7:0	Receive Buffer Register	R	Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs. **Reset Value:** 0x0

DLH

- **Name:** Divisor Latch High
- **Size:** 32 bits
- **Address Offset:** 0x04

- **Read/write access:** read/write

Bits	Name	R/W	Description
31:8	Reserved and read as zero		
7:0	Divisor Latch (High)	R/W	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest DW_apb_uart clock should be allowed to pass before transmitting or receiving data. **Reset Value:** 0x0

DLL

- **Name:** Divisor Latch Low
- **Size:** 32 bits
- **Address Offset:** 0x00
- **Read/write access:** read/write

Bits	Name	R/W	Description
31:8	Reserved and read as zero		
7:0	Divisor Latch (Low)	R/W	Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest uart clock should be allowed to pass before transmitting or receiving data. **Reset Value:** 0x0

IER

- **Name:** Interrupt Enable Register
- **Size:** 32 bits
- **Address Offset:** 0x04
- **Read/write access:** read/write

Bits	Name	R/W	Description
31:8	Reserved and read as zero		

7	PTIME	R/W	Programmable THRE Interrupt Mode Enable that can be written to only when THRE_MODE_USER == Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled **Reset Value:** 0x0
6:4	Reserved and read as zero		
3	EDSSI	R/W	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled **Reset Value:** 0x0
2	ELSI	R/W	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled **Reset Value:** 0x0
1	ETBEI	R/W	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled **Reset Value:** 0x0
0	ERBFI	R/W	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled **Reset Value:** 0x0

IIR

- **Name:** Interrupt Identity Register
- **Size:** 32 bits
- **Address Offset:** 0x08
- **Read/write access:** read-only

Bits	Name	R/W	Description
31:8	Reserved and read as zero		

7:6	FIFOs Enabled (or FIFOSE)	R	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled **Reset Value:** 0x01
5:4	Reserved	N/A	Reserved and read as zero
3:0	Interrupt ID (or IID)	R	Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types: 0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout The interrupt priorities are split into four levels that are detailed in Table7. Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt. **Reset Value:** 0x01

Table 7: Interrupt Control Functions

Interrupt ID				Interrupt Set and Reset Functions			
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0	0	0	1	–	None	None	–
0	1	1	0	Highest	Receiver line status	Overrun/parity/ framing errors or break interrupt	Reading the line status register
0	1	0	0	Second	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)

1	1	0	0	Second	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1 character in it during this time	Reading the receiver buffer register
0	0	1	0	Third	Transmit holding register empty	Transmitter holding register empty (Prog. THRE Mode disabled) or XMIT FIFO at or below threshold (Prog. THRE Mode enabled)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled).
0	0	0	0	Fourth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0	1	1	1	Fifth	Busy detect indication	Master has tried to write to the Line Control Register while the uart is busy (USR[0] is set to one).	Reading the UART status register

LCR

- **Name:** Line Control Register
- **Size:** 32 bits
- **Address Offset:** 0x0C
- **Read/write access:** read/write

Bits	Name	R/W	Description
31:8	Reserved and read as zero		

7	DLAB	R/W	Divisor Latch Access Bit.Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers. **Reset Value:** 0x0
6	Break (or BC)	R/W	Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. **Reset Value:** 0x0
5	Stick Parity	Reserved and read as zero	
4	EPS	R/W	Even Parity Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked. **Reset Value:** 0x0
3	PEN	R/W	Parity Enable. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled **Reset Value:** 0x0
2	STOP	R/W	Number of stop bits. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit **Reset Value:** 0x0

1:0	DLS (or CLS, as used in legacy)	R/W	Data Length Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits **Reset Value:** 0x0

MCR

- **Name:** Modem Control Register
- **Size:** 32 bits
- **Address Offset:** 0x10
- **Read/write access:** read/write

Bits	Name	R/W	Description
31:7	Reserved and read as zero		
6	SIRE	R/W	SIR Mode Enable. Writeable only when SIR_MODE == Enabled, always readable. This is used to enable/disable the IrDA SIR Mode features as described in "IrDA 1.0 SIR Protocol" 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled **Reset Value:** 0x0
5	AFCE	R/W	Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in "Auto Flow Control" 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled **Reset Value:** 0x0

4	LoopBack (or LB)	R/W	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line. **Reset Value:** 0x0
3	OUT2	R/W	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input. **Reset Value:** 0x0
2	OUT1	R/W	OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: 0 = out1_n de-asserted (logic 1) 1 = out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input. **Reset Value:** 0x0
1	RTS	R/W	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input. **Reset Value:** 0x0

0	DTR	R/W	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input. **Reset Value:** 0x0

LSR

- **Name:** Line Status Register
- **Size:** 32 bits
- **Address Offset:** 0x14
- **Read/write access:** read-only

Bits	Name	R/W	Description
31:8	Reserved and read as zero		
7	RFE	R	Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO. **Reset Value:** 0x0
6	TEMT	R	Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty. **Reset Value:** 0x1
5	THRE	R	Transmit Holding Register Empty bit. If THRE_MODE_USER == Disabled or THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER == Enabled AND FIFO_MODE != NONE and both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting. For more details, see "Programmable THRE Interrupt" on page 54. **Reset Value:** 0x1

Bits	Name	R/W	Description

4	BI	R	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of *start time* + *data bits* + *parity* + *stop bits*. If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of *start time* + *data bits* + *parity* + *stop bits*. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. **Reset Value:** 0x0
3	FE	R	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit. **Reset Value:** 0x0
2	PE	R	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit. **Reset Value:** 0x0

Bits	Name	R/W	Description

1	OE	R	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit. **Reset Value:** 0x0
0	DR	R	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode. **Reset Value:** 0x0

MSR

- **Name:** Modem Status Register
- **Size:** 32 bits
- **Address Offset:** 0x18
- **Read/write access:** read-only

Whenever bits 0, 1, 2 or 3 are set to logic one, to indicate a change on the modem control inputs, a modem status interrupt is generated if enabled through the IER, regardless of when the change occurred. Since the delta bits (bits 0, 1, 3) can get set after a reset if their respective modem signals are active (see individual bits for details), a read of the MSR after reset can be performed to prevent unwanted interrupts.

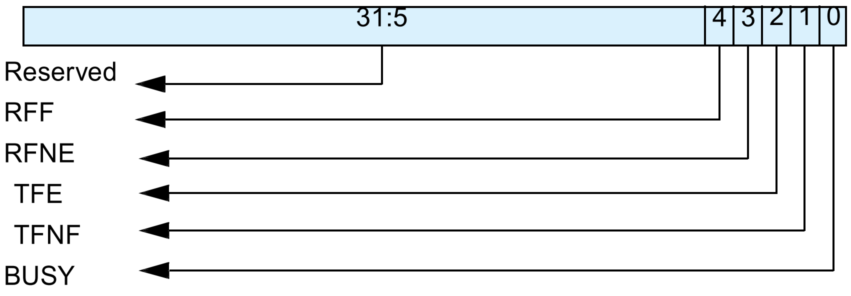
Bits	Name	R/W	Description
31:8	Reserved and read as zero		
7	DCD	R	Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. 0 = dcd_n input is de-asserted (logic 1) 1 = dcd_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), DCD is the same as MCR[3] (Out2). **Reset Value:** 0x0

Bits	Name	R/W	Description
6	RI	R	Ring Indicator. This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. 0 = ri_n input is de-asserted (logic 1) 1 = ri_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), RI is the same as MCR[2] (Out1). **Reset Value:** 0x0
5	DSR	R	Data Set Ready. This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the uart. 0 = dsr_n input is de-asserted (logic 1) 1 = dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), DSR is the same as MCR[0] (DTR). **Reset Value:** 0x0
4	CTS	R	Clear to Send. This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the Uart. 0 = cts_n input is de-asserted (logic 1) 1 = cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS). **Reset Value:** 0x0
3	DDCD	R	Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. 0 = no change on dcd_n since last read of MSR 1 = change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted. **Reset Value:** 0x0
2	TERI	R	Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read. 0 = no change on ri_n since last read of MSR 1 = change on ri_n since last read of MSR Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low. **Reset Value:** 0x0

1	DDSR	R	Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. 0 = no change on dsr_n since last read of MSR 1 = change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). Note, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted. **Reset Value:** 0x0
0	DCTS	R	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. 0 = no change on ctsdsr_n since last read of MSR 1 = change on ctsdsr_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted. **Reset Value:** 0x0

USR

- **Name:** UART Status Register
- **Size:** 32 bits
- **Address Offset:** 0x7C
- **Read/write access:** read-only



Bits	Name	R/W	Description
31:5	Reserved and read as zero		

4	RFF	R	Receive FIFO Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full. **Reset Value:** 0x0
3	RFNE	R	Receive FIFO Not Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty. **Reset Value:** 0x0
2	TFE	R	Transmit FIFO Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty. **Reset Value:** 0x1
1	TFNF	R	Transmit FIFO Not Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full. **Reset Value:** 0x1
0	BUSY	R	UART Busy. This is indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive. 0 = DW_apb_uart is idle or inactive 1 = DW_apb_uart is busy (actively transferring data) Note that it is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled), the assertion of this bit is also delayed by several cycles of the slower clock. **Reset Value:** 0x0

15.0x10000900~0x100009ff TIMER

Table 8: Memory Map of Timer 1 Registers

Name	Address Offset	Width	R/W	Description
Timer1LoadCount	0x00, 0x01, 0x02, 0x03	See Description	R/W	Value to be loaded into Timer1. **Width:** *32* **Range:** 0 to 2^32 **Default value:** *32*'b0
Timer1CurrentValue	0x04, 0x05, 0x06, 0x07	See Description	R	Current Value of Timer1. **Width:** *32* **Range:** 0 to 2^32 **Default value:** *32*'b0
Timer1ControlReg	0x08	3 bits	R/W	Control Register for Timer1. **Default value:** 3'b0
Timer1EOI	0x0c	1 bit	R	Clears the interrupt from Timer1. **Default value:** 1'b0
Timer1IntStatus	0x10	1 bit	R	Contains the interrupt status for Timer1. **Default value:** 1'b0

Table 9: Memory Map of Timer 2 Registers

Name	Address Offset	Width	R/W	Description
Timer1LoadCount	0x14, 0x15, 0x16, 0x17	See Description	R/W	Value to be loaded into Timer2. **Width:** *32* **Range:** 0 to 2^32 **Default value:** *32*'b0
Timer1CurrentValue	0x18, 0x19, 0x1A, 0x1B	See Description	R	Current Value of Timer2. **Width:** *32* **Range:** 0 to 2^32 **Default value:** *32*'b0
Timer1ControlReg	0x1C	3 bits	R/W	Control Register for Timer2. **Default value:** 3'b0
Timer1EOI	0x20	1 bit	R	Clears the interrupt from Timer2. **Default value:** 1'b0
Timer1IntStatus	0x24	1 bit	R	Contains the interrupt status for Timer2. **Default value:** 1'b0

Table 9: Timers System Registers

Name	Address Offset	Width	R/W	Description
TimersIntStatus	0xa0	See Description	R	Contains the interrupt status of all timers in the component. **Width:** *2* **Default value:** *2*'b0
TimersEOI	0xa4	See Description	R	Returns all zeroes (0) and clears all active interrupts. **Width:** *2* **Default value:** *2*'b0
TimersRawIntStatus	0xa8	See Description	R	Contains the unmasked interrupt status of all timers in the component. **Width:** *2* **Default value:** *2*'b0
TIMERS_COMP_ VERSION	0xac	32 bits	R	Current revision number of the timers component.

TimerLoadCount

- **Name:** Timer1/2 Load Count Register
- **Size:** *32*
- **Address Offset:0x00(Timer1) ,0x14(Timer2)**
- **Read/write access:** read/write

Bits	Name	R/W	Description
31:0	Timer1,2 Load Count Register	R/W	Value to be loaded into Timer1/Timer2. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.

TimerCurrentValue

- **Name:** Timer*N* Current Value Register
- **Size:** *32*
- **Address Offset: 0x04(Timer1) 0x18(Timer2)**
- **Read/write access:** read

Bits	Name	R/W	Description
31:0	Timer 1/2 Current Value	R	Current Value of Timer1/Timer2.

TimerControlReg

- **Name:** Timer*N* Control Register

- **Size:** 3 bits
- **Address Offset: 0x8(Timer1) 0x1C(Timer2)**

Bits	Name	R/W	Description
31:3	Reserved, read as zero		
2	Timer Interrupt Mask	R/W	Timer interrupt mask for Timer1/2. 0: not masked 1: masked
1	Timer Mode	R/W	Timer mode for Timer1/2. 0: free-running mode 1: user-defined count mode For more information about these modes, see "Timer Operation" .
0	Timer Enable	R/W	Timer enable bit for Timer1/2. 0: disable 1: enable

Timer*N*EOI

- **Name:** Timer1/2 End-of-Interrupt Register
- **Size:** 1 bit
- **Address Offset: 0x0c(Timer1) 0x20(Timer2)**

Bits	Name	R/W	Description
31:1	Reserved, read as zero		
0	Timer1/2 End-of-Interrupt Register	R	Reading from this register returns all zeroes (0) and clears the interrupt from Timer1/2.

TimerIntStatus

- **Name:** Timer1/2 Interrupt Status Register
- **Size:** 1 bit
- **Address Offset:0x10(Timer1) 0x24(Timer2)**

Bits	Name	R/W	Description
31:1	Reserved, read as zero		
0	Timer1/2 Interrupt Status Register	R	Contains the interrupt status for Timer1/2.

TimersIntStatus

- **Name:** Timers Interrupt Status Register
- **Size:** *2*
- **Address Offset:** 0xa0
- **Read/write access:** read

Bits	Name	R/W	Description
1:0	Timers Interrupt Status Register	R	Contains the interrupt status of all timers in the component. If a bit of this register is 0, then the corresponding timer interrupt is not active – and the corresponding interrupt could be on either the timer_intr bus or the timer_intr_n bus, depending on the interrupt polarity you have chosen. Similarly, if a bit of this register is 1, then the corresponding interrupt bit has been set in the relevant interrupt bus. In both cases, the status reported is the status after the interrupt mask has been applied. Reading from this register does not clear any active interrupts: 0 = either timer_intr or timer_intr_n is not active after masking 1 = either timer_intr or timer_intr_n is active after masking

TimersEOI

- **Name:** Timers End-of-Interrupt Register
- **Size:** *2*
- **Address Offset:** 0xa4
- **Read/write access:** read

Bits	Name	R/W	Description
1:0	Timers End-of-Interrupt Register	R	Reading this register returns all zeroes (0) and clears all active interrupts.

TimersRawIntStatus

- **Name:** Timers Raw Interrupt Status Register
- **Size:** *2*
- **Address Offset:** 0xa8
- **Read/write access:** read

Bits	Name	R/W	Description
1:0	Timers Raw Interrupt Status Register	R	The register contains the unmasked interrupt status of all timers in the component. 0 = either timer_intr or timer_intr_n is not active prior to masking 1 = either timer_intr or timer_intr_n is active prior to masking

Timer Operation

Timers count down from a programmed value and generate an interrupt when the count reaches zero. You can use the TIM_INTR_IO parameter (Single Combined Interrupt) to create a single combined interrupt, which is active whenever any of the individual timer interrupts is active.

The initial value for each timer – that is, the value from which it counts down – is loaded into the timer using the appropriate load count register (TimerNLoadCount). Two events can cause a timer to load the initial count from its Timer*N*LoadCount register:

- Timer is enabled after being reset or disabled
- Timer counts down to 0

All interrupt status registers and end-of-interrupt registers can be accessed at any time.

2.9 Interrupt vector table

0x00000000	Top of Stack
0x00000004	Reset Handler
0x00000008	NMI Handler
0x0000000c	HardFaultHandler
0x00000010	Reserved
0x00000014	Reserved
0x00000018	Reserved
0x0000001c	Reserved
0x00000020	Reserved
0x00000024	Reserved
0x00000028	Reserved
0x0000002c	SVCall
0x00000030	Reserved
0x00000034	Reserved
0x00000038	PendSV
0x0000003c	SysTick
0x00000040	WDT
0x00000044	Timer0
0x00000048	Timer1
0x0000004c	GPIO
0x00000050	GPIO WAKE UP
0x00000054	SOUND
0x00000058	USB
0x0000005c	UART
0x00000060	LCD POWER DOWN REDDY
0x00000064	LCD WAKE UP DONE
0x00000068	LCD FRAME COUNT
0x0000006c	LOW VOLTAGE
0x00000070	SPI
0x00000074	I2C
0x00000078	Reserved
0x0000007c	RTC
0x00000080	ADC
0x00000084	Reserved
0x00000088	Reserved

2.10 Flash ROM 配置位 （Flash Rom 地址 0x0~0xFFFF ----64K Byte，EEPROM 地址 0x10000~0x100FF---256Byte）

Address ： 0xFFFC

Data ： 32bit

OPTION 31~0 如果是空片（出厂），数据都是 1

Description：

Bit 31： LOCK ID “1”锁定 ID（同时需要 ID[1:0]=01 或者 10），不能再改 ID

Bit 30~28: LOCK SWD READ

当 Bit30=0,Bit29=1,Bit28=0 时，锁定读取使能，只能全部擦除 Flash rom(擦除是全部擦成 0)，不能再读取 Flash 内容

或者改写部分 Flash 内容。
Bit27~26：Reserved
Bit25~16: Reserved
Bit15~0: ID

2.11 ADC 具体操作寄存器

1) 0x04000024 bit31~bit0 W
bit31: Reserved
bit12: sel_pll_1MHz
bit11~8: divider N（默认 7）Fclk_adc=Fpll / ((N+1)*2) 基本不用修改
bit7~4 : 0000
bit3~0: Reserved

2) 0x04000010 bit31~bit0 W
bit31: Reserved
bit30: Reserved
bit29: ADC enable 1： enable 0： disable
bit28: ADC reset IP 复位 0:复位 1:工作
bit27: Reserved
bit26: Reserved
bit25: gain1x 1: gain=1x 第 1 级前置放大器跟随器
bit24~22: adc_chop_sel ADC 斩波模式
bit21~20: adc_chop_freq ADC 斩波频率
bit19: Reserved
bit18~16: amp_gain2 前置放大器第 2 级增益选择
bit15~14: amp_gain1 前置放大器第 1 级增益选择
bit13~11: amp_chop_sel 前置放大器斩波波形选择
bit10~9: amp_chop_freq 前置放大器斩波频率选择
bit8: amp_chop_en 前置放大器斩波使能
bit7~0: MUX_select 输入通道选择

mux_sel<3:0>	正端输入		mux_sel<7:4>	负端输入
0000	INP0		0000	INN0
0001	INN0		0001	INP0
0010	内部 VDD/2		0010	内部 VDD/2
0011	内部温度+		0011	内部温度-
0100	INP4		0100	INN4
0101	INP5		0101	INN5
0110	INP6		0110	INN6
0111	INP7		0111	INN7
1000	INP8		1000	INN8
1001	INP9		1001	INN9
1010	INP10		1010	INN10
1011	INP11		1011	INN11
1100	INP12		1100	INN12
1101	INP13		1101	INN13
1110	INP14		1110	INN14
1111	INP15		1111	INN15
Summy			0	

表 1 前置放大器前置放大器斩波频率选择

AMP_CHOP_FREQ [1:0]（二进制）	分频比
00	2
01	4
10	**8**
11	16

AMP_CHOP_SEL 前置放大器斩波波形选择　测试中发现这个设置作用不大，取 100 即可

GAIN_1X　Output　第 1 级前置放大器跟随器

AMP_GAIN1　　前置放大器第 1 级增益选择

表 2 第 1 级前置放大器增益选择

GAIN_1X+ AMP_GAIN1[1:0]（二进制）	放大倍数
1XX	1
000	12
001	20
010	30
011	40

AMP_GAIN2　　前置放大器第 2 级增益选择

表 1 前置放大器第 2 级的增益

AMP_GAIN2 [2:0](二进制)	第 **2** 级增益
000	1.1
001	1.2
010	1.3
011	1.4
100	1.5
101	1.6
110	1.7
111	1.8

AMPAZ_START　　前置放大器零点校准使能

ADC_DONE　　完成指示输出

ADC_START　　ADC 启动信号

ADC_OSR　　ADC 过采样率选择

表 4 ADC 过采样率选择

ADC_OSR [2:0] (二进制)	**OSR for Pressure/Temperature Measurement**
000	oversampling×128
001	oversampling × 64
010	oversampling× 32
011	oversampling× 16
100	oversampling× 8
101	oversampling× 4
110	oversampling × 2
111	oversampling × 1

过采样率选择	ADC_OSR	Conversion Time (ms)
过采样率 128x	000	100
过采样率 64x	001	50

过采样率 32x	010	25
过采样率 16x	011	12.5
过采样率 8x	100	6.25
过采样率 4x	101	3.125
过采样率 2x	110	1.512
过采样率 1x	111	0.8

ADC_OFFSET　　　ADC 偏置选择

表 2ADC offset 信号选择

ADC_OFFSET [2:0]（二进制）	Vin 输入范围（*VDDB）
000	-1/16~15/16
001	-2/16 ~ 14/16
010	-3/16 ~ 13/16
011	-4/16 ~ 12/16
100	-5/16 ~ 11/16
101	-6/16 ~ 10/16
110	-7/16 ~ 9/16
111	-8/16 ~ 8/16

ADC_CHOP_FREQ　　　ADC 斩波频率　固定为“10”

ADC_CHOP_SEL　　　ADC 斩波模式　固定为“11”

ADC_CLK_DIV　　　ADC 时钟分频比

表 3 ADC 采样时钟频率选择

Clk_Divider[7:6](二进制)	Sample Clock Divider	ADC Sample Frequency
00	OSC_freq / 12	~ 333kHz
01	OSC_freq / 10	~ 400kHz
10	OSC_freq / 8	~ 500kHz
11	OSC_freq / 6	~ 666kHz

ADC_DOUT　　　ADC 数据输出

3) 0x04000014　　bit31~bit0　　W

bit31:	Reserved	
bit30：	Reserved	
bit29：	ampaz_start	前置放大器零点校准使能
bit28：	clk_select	IP 时钟选择信号 0:外部输入/1: IP 内部振荡器
bit27~25:	osc_trim	内部振荡器微调
bit24~23:	bias_adc2	第 3 级 ADC 偏置电路控制
bit22~21:	bias_adc1	第 2 级 ADC 偏置电路控制
bit20~19:	bias_adc0	第 1 级 ADC 偏置电路控制
bit18~17：	bias_amp1	第 2 级前置放大器偏置电路控制
bit16~15 :	bias_amp0	第 1 级前置放大器偏置电路控制
bit14~12 :	vref_trim	内部基准电压微调
bit11~10:	adc_clk_div	ADC 时钟分频比

bit9~7: adc_offset ADC 偏置选择
bit6~4 : adc_osr ADC 过采样率选择
bit3:0 : Reserved

OSC_TRIM 内部振荡器微调

表- 3 时钟振荡器的校准

OSC_TRIM [2:0]	**OSC 频率**
111	110% Default Frequency
110	107% Default Frequency
101	103% Default Frequency
100	**4 MHz (缺省值，仿真值)**
011	96% Default Frequency
010	93% Default Frequency
001	87% Default Frequency
000	84% Default Frequency

BIAS_AMP0 第 1 级前置放大器偏置电流控制

BIAS_AMP0 [1:0]	00	01	10	11
偏置电流	0.6uA	0.9uA	1.2uA	1.5uA

BIAS_AMP1 第 2 级前置放大器偏置电流控制

BIAS_AMP1 [1:0]	00	01	10	11
偏置电流	0.6uA	0.9uA	1.2uA	1.5uA

BIAS_ADC0 第 1 级 ADC 偏置电流控制

BIAS_ADC0 [1:0]	00	01	10	11
偏置电流	1.2uA	1.8uA	2.4uA	3.0uA

BIAS_ADC1 第 2 级 ADC 偏置电流控制

BIAS_ADC1 [1:0]	00	01	10	11
偏置电流	0.6uA	0.9uA	1.2uA	1.5uA

BIAS_ADC2 第 3 级 ADC 偏置电流控制

BIAS_ADC2 [1:0]	00	01	10	11
偏置电流	0.6uA	0.9uA	1.2uA	1.5uA

VREF_TRIM 内部基准电压微调

表- 4 基准电压的校准

VREF_TRIM [2:0]	1V Reference Voltage	**Vref_trim[2:0]**	1V Reference Voltage

000	-28mV	**100**	**1.200V (缺省值，仿真值)**
001	-21mV	101	+7mV
010	-14mV	110	+14mV
011	-7mV	111	+21mV

4)　0x04000038　　bit31~bit0　　W/R

bit31~1: Reserved
bit0:　adc_start　　W　　ADC 开始信号
bit0:　adc_done　　R　　ADC 完成信号

5)　0x0400003C　　bit31~bit0　　W/R

bit31~1: Reserved　　W
bit0 :　adc_done clear　　W　　1:clear　0: release
bit23~0 : adc_value　　R　　ADC 数据输出

ADC 具体操作

1. ADC 启动顺序
①首先设置 clk_select
②启动 clk_adc
③10ms 后使能 ADC
④20ms 后放开 ADC reset，并设好配置位
⑤10ms 后开始测试

2. ADC 测量操作
①ADC 启动顺序完成后，首先设置清除 adc_done，然后判断 adc_done=0 证明 adc 处在初始状态。
②释放清除信号。
③设置 adc_start=1 开始测试。
④等待 adc_done=1,等于 1 表示 ADC 转换完成，ADC 转换值为有效值。
⑤读取 ADC 转换值
⑥设置 adc_start=0
⑦清除 adc_done（adc_done_clear=1）
⑧释放清除信号（adc_done_clear=0）
ADC 一个测试循环结束。继续测试按次动作操作。

注：ADC 测量范围以及注意事项
1.PIN20 脚 VDD_ADC 是 ADC 内部输出的电压，具有 2mA 驱动能力，能够为分压电阻，以及其他传感器提供电源。建议优先选择此脚作为传感器或者热敏电阻等的电源。
2.ADC 测量范围，ADC 满量程为 Vf=VDD_ADC 约等 1.65V。
3.ADC 最佳测试范围为 0~0.65 倍 VDD_ADC。建议将所需测量电压转换到此电压范围内再进行测量。
4.在使用 ADC 之前 ADC 的 IO 功能也需要使能为 ADC 功能。具体配置参考 0x10000160 地址配置。

UART 波特率调整公式：

baud rate = (serial clock freq) / (16 * divisor)

serial clock freq = APB clock = fsys/(apb_clk_div)

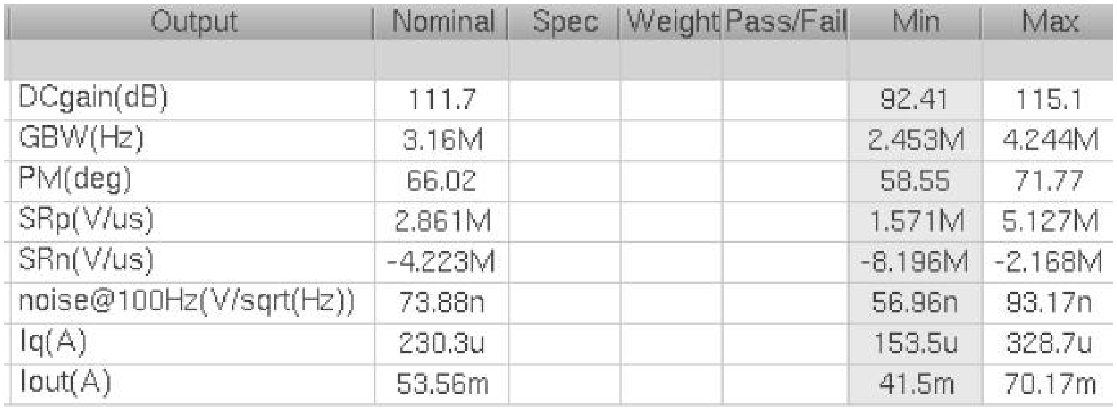
divisor = （DLH<<8+DLL）(DLH high 8bit ,DLL low 8bit)

DLH ,DLL 参见 UART 寄存器列表

apb_clk_div 最快可以除 2

独立 OP 特性：

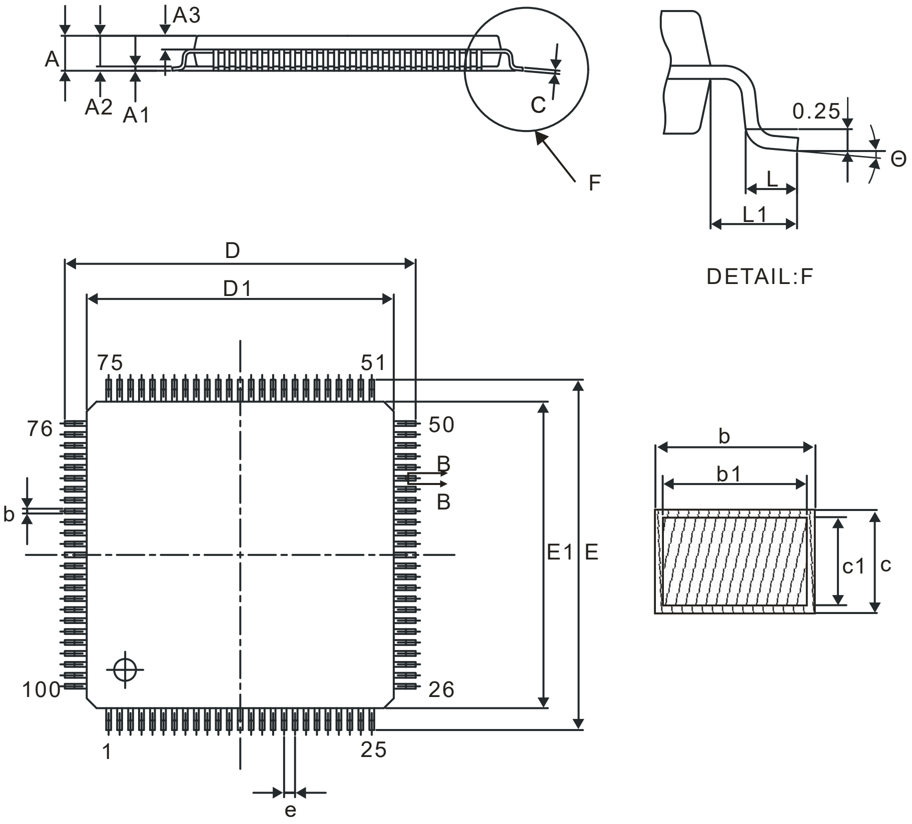
1. Rail2Rail 结构，输入输出范围 0~3.3V
2. 对低频小信号放大时具有低噪声特性
3. 静态功耗 220uA
4. 关断电流小于 1uA
5. 失调电压 200uV
6. 开环增益 110dB
7. 用作比较器时，可以驱动 100pF 的电容，直流增益 80dB 以上



Output	Nominal	Spec	Weight	Pass/Fail	Min	Max
DCgain(dB)	111.7				92.41	115.1
GBW(Hz)	3.16M				2.453M	4.244M
PM(deg)	66.02				58.55	71.77
SRp(V/us)	2.861M				1.571M	5.127M
SRn(V/us)	-4.223M				-8.196M	-2.168M
noise@100Hz(V/sqrt(Hz))	73.88n				56.96n	93.17n
Iq(A)	230.3u				153.5u	328.7u
Iout(A)	53.56m				41.5m	70.17m

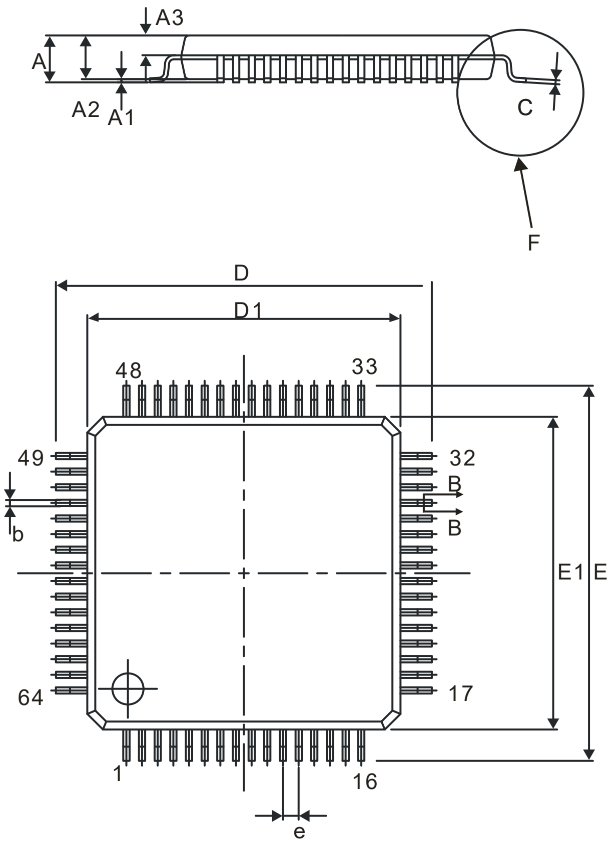
封装规格：

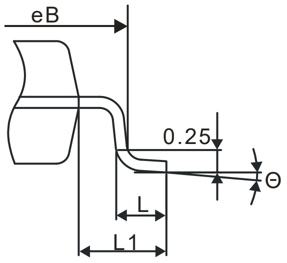
LQFP100 14 x 14 mm, 100-pin 封装尺寸图



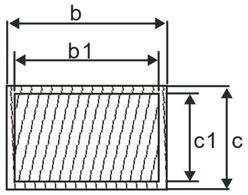
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	——	——	1.60
A1	0.05	——	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	——	0.26
b1	0.17	0.20	0.23
c	0.13	——	0.17
c1	0.12	0.13	0.14
D	15.80	16.00	16.20
D1	13.90	14.00	14.10
E	15.80	16.00	16.20
E1	13.90	14.00	14.10
Eb	15.05	——	15.35
e	0.50BSC		
L	0.45	——	0.75
L1	1.00REF		
Θ	0	——	7

LQFP64 7 x 7 mm, 64-pin 封装尺寸图





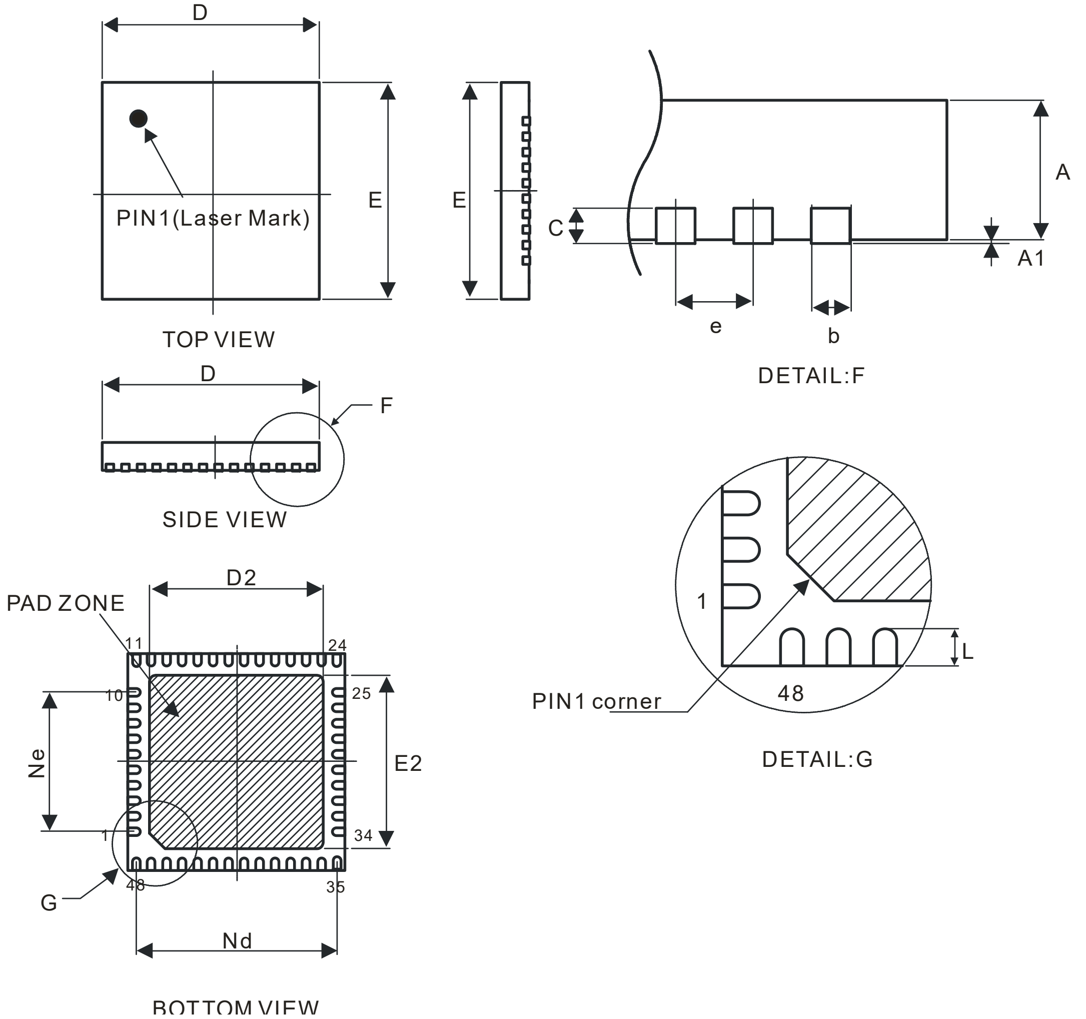
DETAIL:F



SECTION B-B

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	———	———	1.60
A1	0.05	———	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.16	———	0.24
b1	0.15	0.18	0.21
c	0.13	———	0.17
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
Eb	8.10	———	8.25
e	0.40BSC		
L	0.45	———	0.75
L1	1.00REF		
Θ	0	———	7

QFN48 5 x 5 mm, 0.35mm pitch, 48-pin 封装尺寸图



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.80	0.85	0.90
A1	0	0.02	0.05
b	0.13	0.18	0.23
b1	0.12REF		
c	0.10	0.15	0.20
D	4.90	5.00	5.10
D2	3.60	3.70	3.80
e	0.35BSC		
Ne	3.15BSC		
Nd	4.55BSC		
E	4.90	5.00	5.10
E2	3.60	3.70	3.80
L	0.35	0.40	0.45