



第三章 GPIO接口 (General Purpose Input and Output, GPIO)



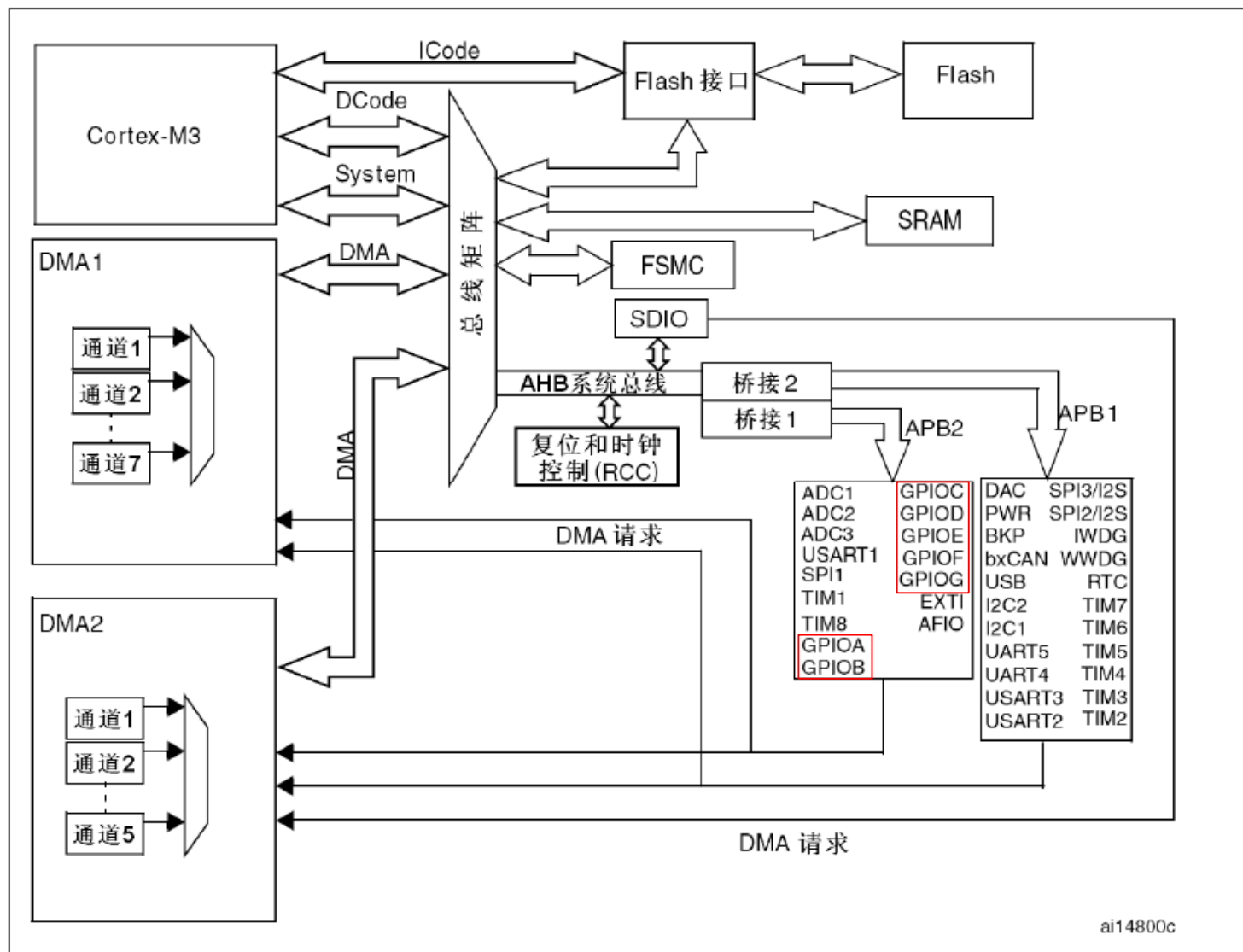
isszym 2020.11.12

概述

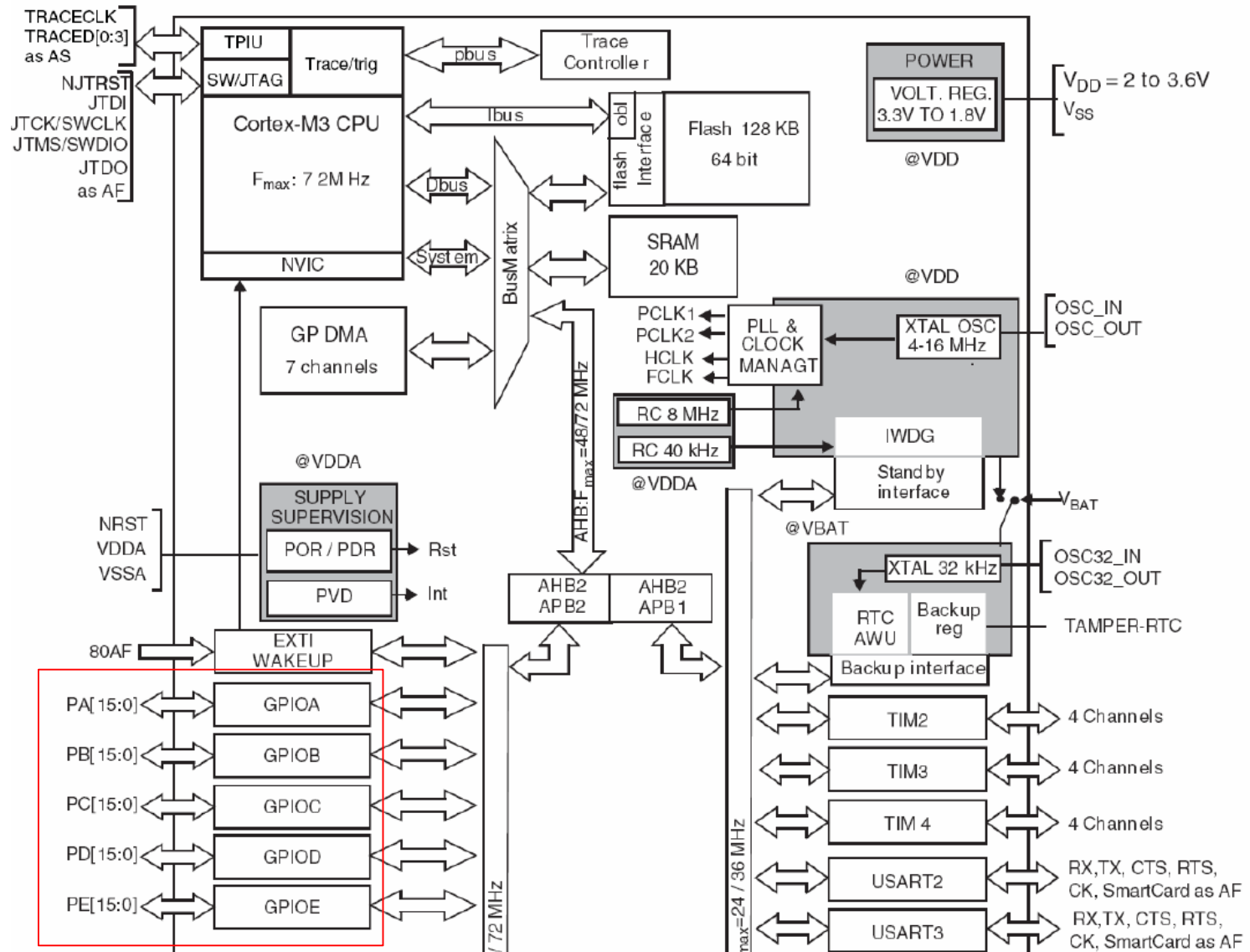
- STM32F103C8T6有五个16位的GPIO接口：GPIOA~GPIOE，后面都用GPIOx表示。
- 每个GPIO端口有两个32位配置寄存器(GPIOx_CRL, GPIOx_CRH)，两个32位数据寄存器(GPIOx_IDR和GPIOx_ODR)，一个32位置位/复位寄存器(GPIOx_BSRR)，一个16位复位寄存器(GPIOx_BRR)和一个32位锁定寄存器(GPIOx_LCKR)。
- GPIO端口的每个位可以由软件分别配置成多种模式：

输入浮空	开漏输出
输入上拉	推挽式输出
输入下拉	推挽式复用功能
模拟输入	开漏复用功能
- 每个I/O端口位可以自由编程，然而必须按照32位字访问I/O端口寄存器(不允许半字或字节访问)。GPIOx_BSRR和GPIOx_BRR寄存器允许对任何GPIO寄存器进行读/更改的独立访问；这样，在读和更改访问之间产生IRQ时不会发生危险。

STM32系统构架



STM32F103C8T6



产品功能和外设配置 中等容量

STM32F103C8T6

外设		STM32F103Tx	STM32F103Cx	STM32F103Rx		STM32F103Vx		
闪存(K字节)		64	64	128	64	128	64	128
SRAM(K字节)		20	20	20	20		20	
定时器	通用	3个(TIM2、TIM3、TIM4)						
	高级控制	1个(TIM1)						
通信接口	SPI	1个(SPI1)	2个(SPI1、SPI2)					
	I ² C	1个(I ² C1)	2个(I ² C1、I ² C2)					
	USART	2个(USART1、USART2)	3个(USART1、USART2、USART3)					
	USB	1个(USB 2.0全速)						
	CAN	1个(2.0B 主动)						
GPIO端口		26	37	51		80		
12位ADC模块(通道数)		2(10)	2(10)		2(16)		2(16)	
CPU频率		72MHz						
工作电压		2.0~3.6V						
工作温度		环境温度：-40℃~+85℃/-40℃~+105℃(见表8) 结温度：-40℃~+125℃(见表8)						
封装形式		VFQFPN36	LQFP48		LQFP64 TFBGA64		LQFP100 LFBGA100	

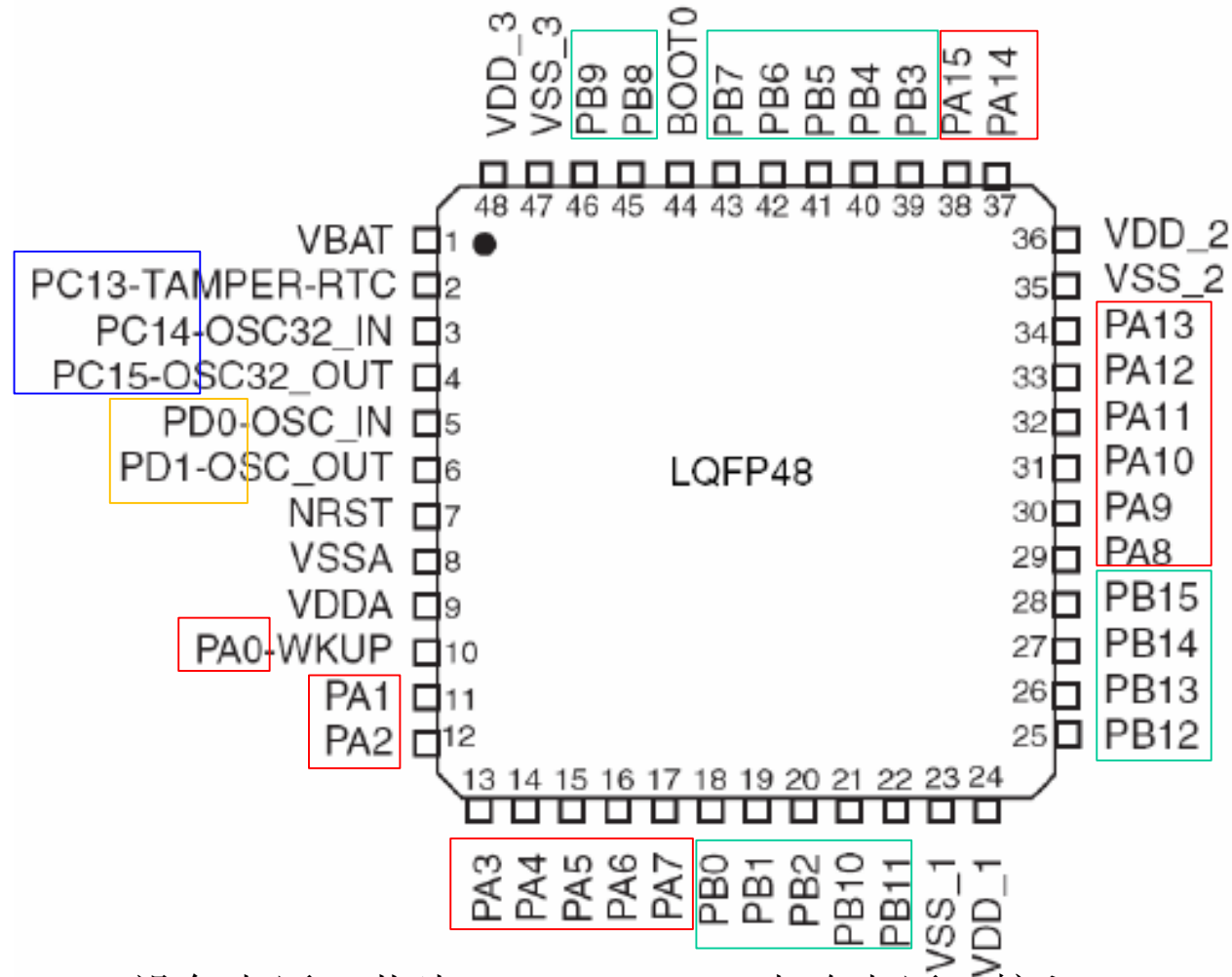
STM32 & STM8 产品型号

STM32F103C8T6



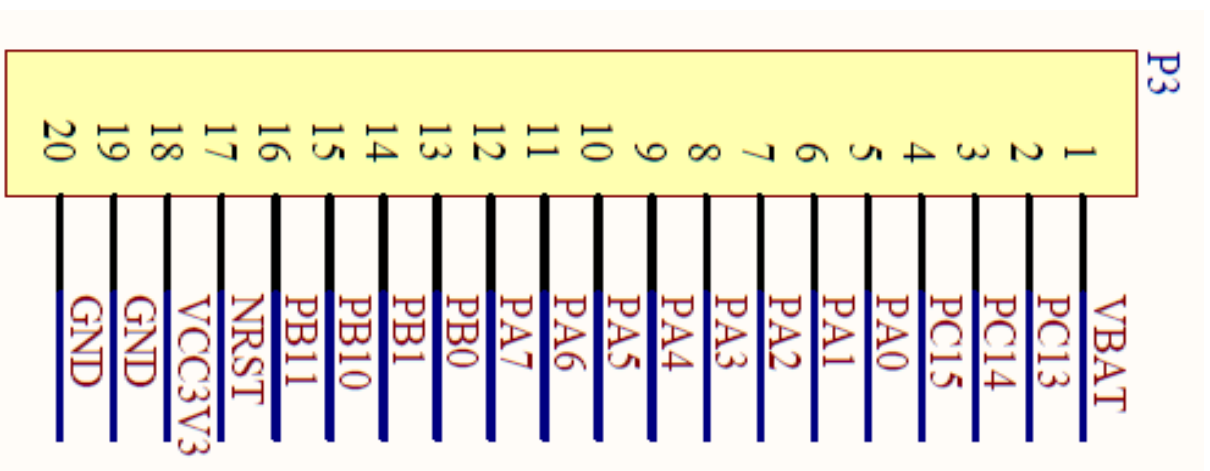
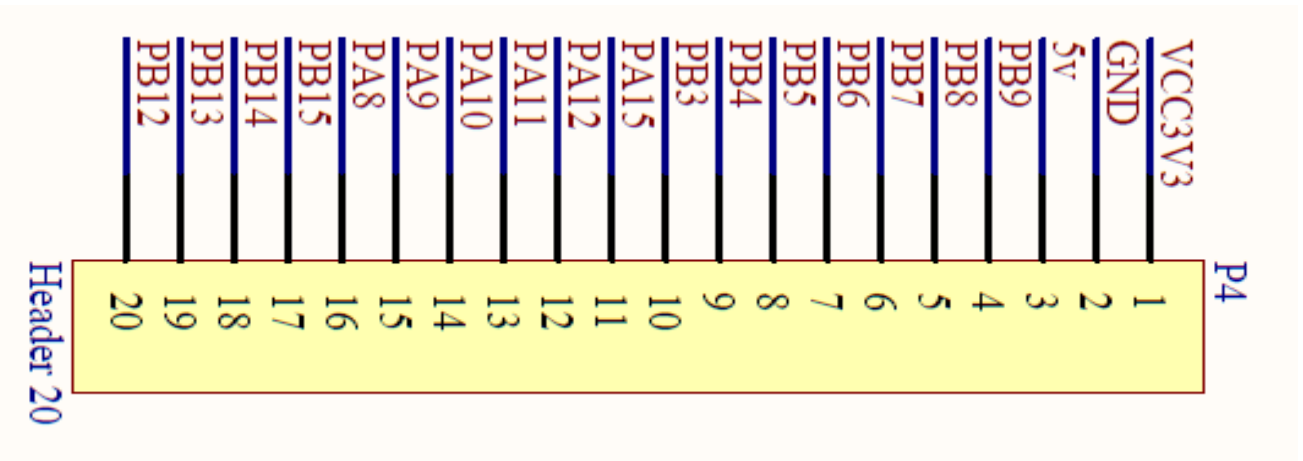
QFP: 方型扁平式封装技术(Quad Flat Package)

37个GPIO引脚： PA0-PA15： 共16个
PB0-PB15： 共16个
PC13-PC15： 共3个
PD0-PD1： 共2个；



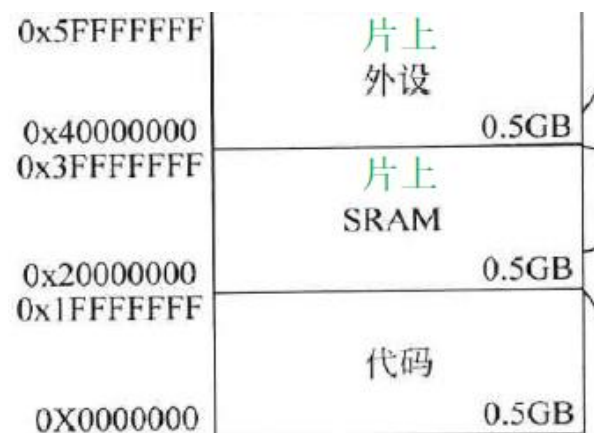
V_{DD} -Device(设备电压，芯片) V_{CC} -Circuit(电路电压，接入) V_{SS} -Serial(公共地)

$$V_{CC} > V_{DD}$$



寄存器组起始地址

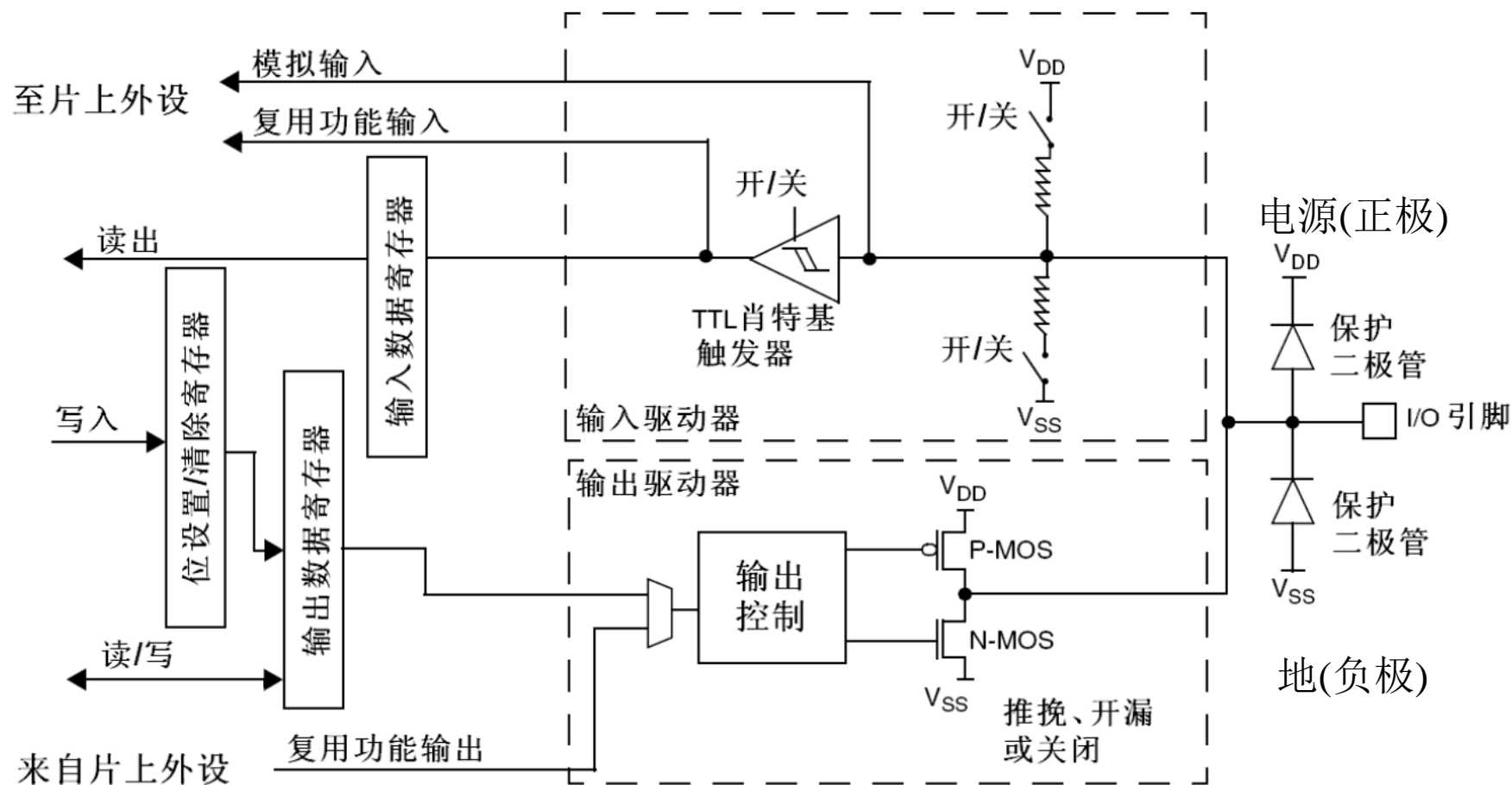
起始地址	外设	总线
0x5000 0000 – 0x5003 FFFF	USB OTG 全速	AHB
0x4003 0000 – 0x4FFF FFFF	保留	
0x4002 8000 – 0x4002 9FFF	以太网	
0x4002 3400 - 0x4002 3FFF	保留	
0x4002 3000 - 0x4002 33FF	CRC	AHB
0x4002 2000 - 0x4002 23FF	闪存存储器接口	
0x4002 1400 - 0x4002 1FFF	保留	
0x4002 1000 - 0x4002 13FF	复位和时钟控制(RCC)	
0x4002 0800 - 0x4002 0FFF	保留	
0x4002 0400 - 0x4002 07FF	DMA2	
0x4002 0000 - 0x4002 03FF	DMA1	
0x4001 8400 - 0x4001 7FFF	保留	
0x4001 8000 - 0x4001 83FF	SDIO	
0x4001 4000 - 0x4001 7FFF	保留	
0x4001 3C00 - 0x4001 3FFF	ADC3	
0x4001 3800 - 0x4001 3BFF	USART1	APB2
0x4001 3400 - 0x4001 37FF	TIM8定时器	
0x4001 3000 - 0x4001 33FF	SPI1	
0x4001 2C00 - 0x4001 2FFF	TIM1定时器	
0x4001 2800 - 0x4001 2BFF	ADC2	
0x4001 2400 - 0x4001 27FF	ADC1	
0x4001 2000 - 0x4001 23FF	GPIO端口G	
0x4001 2000 - 0x4001 23FF	GPIO端口F	
0x4001 1800 - 0x4001 1BFF	GPIO端口E	
0x4001 1400 - 0x4001 17FF	GPIO端口D	
0x4001 1000 - 0x4001 13FF	GPIO端口C	
0x4001 0C00 - 0x4001 0FFF	GPIO端口B	APB1
0x4001 0800 - 0x4001 0BFF	GPIO端口A	
0x4001 0400 - 0x4001 07FF	EXTI	
0x4001 0000 - 0x4001 03FF	AFIO	APB1
0x4000 7800 - 0x4000FFFF	保留	



STM32F103C8T6只有五个GPIO端口: GPIOA~E

0x4001 2000 - 0x4001 23FF	GPIO端口G
0x4001 2000 - 0x4001 23FF	GPIO端口F
0x4001 1800 - 0x4001 1BFF	GPIO端口E
0x4001 1400 - 0x4001 17FF	GPIO端口D
0x4001 1000 - 0x4001 13FF	GPIO端口C
0x4001 0C00 - 0x4001 0FFF	GPIO端口B
0x4001 0800 - 0x4001 0BFF	GPIO端口A

I/O端口位的基本结构

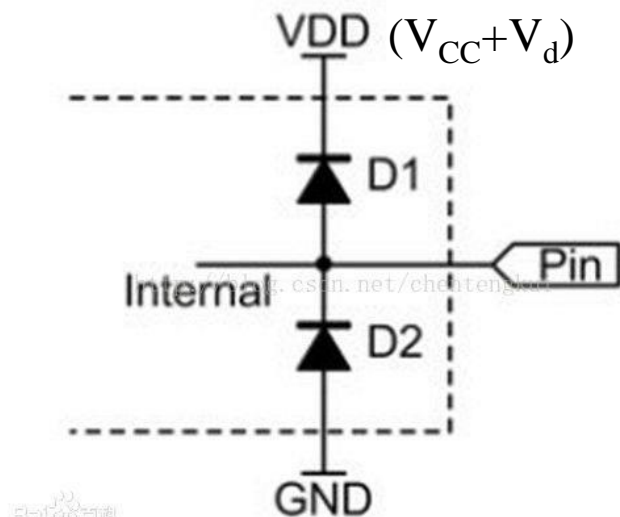


* GND: 公共地 V_{SS} 负极

* 5伏兼容I/O端口位采用 V_{DD_FT} 代替 V_{DD} ，以容忍5伏电压。

□ 双二极管钳位电路的原理

对于正常的二极管，其正向电阻约为几千欧，反向电阻为几百千欧（一般应大于 200 千欧），而MOS管一般内阻都在10M-1000G欧，所以二极管的内阻远小于场效应管的电阻。如果 V_{CC} 为3.3V， V_{DD} 为+5V，电压被钳制在+5V~-1.5V。



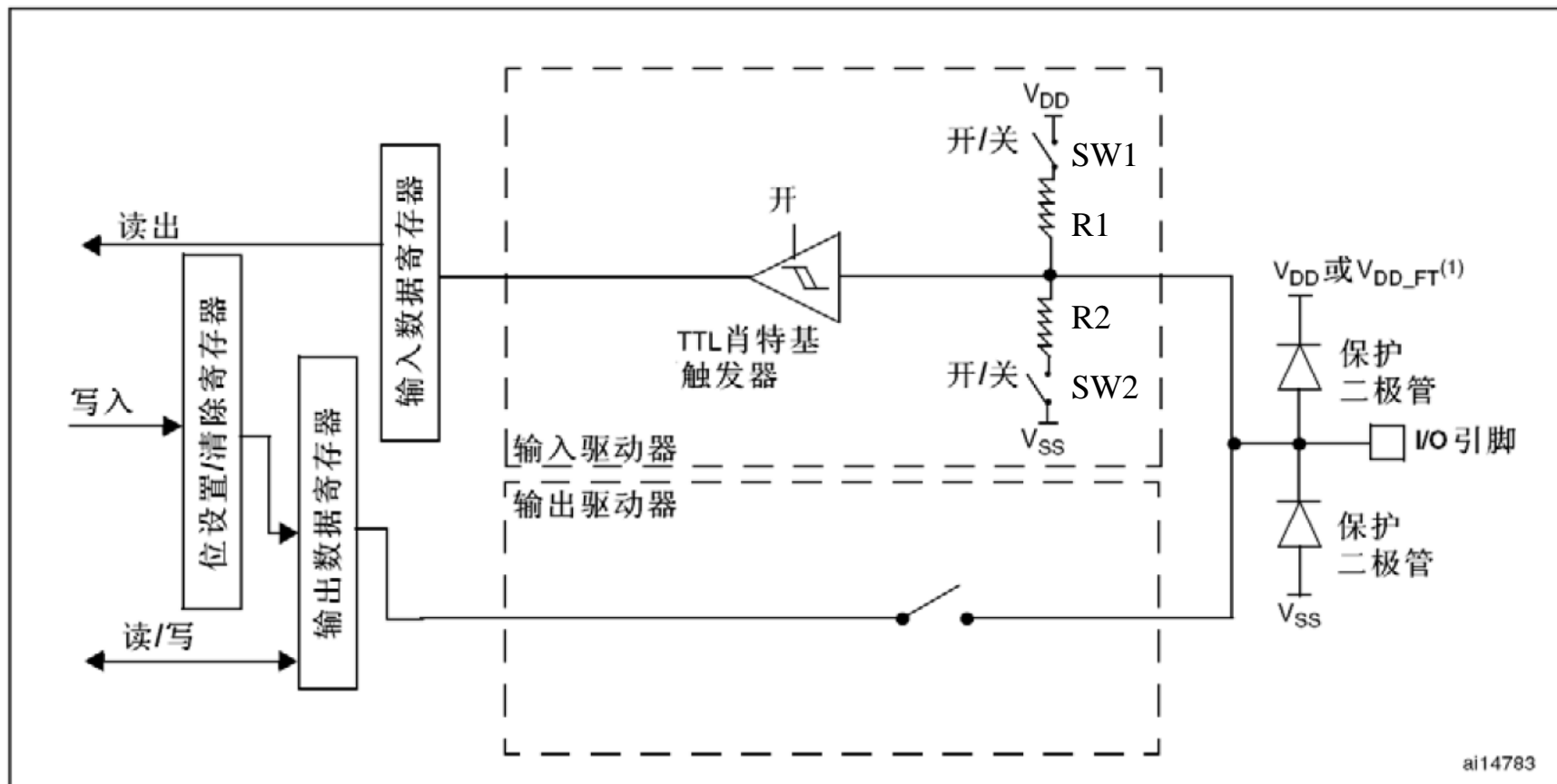
如果输入电压过高，高于 $V_{CC} + V_d$ （二极管导通压降），上面的二极管导通，输出电压钳位于 $V_{CC} + V_d$

如果输入电压过低，低于 $0 - V_d$ （二极管导通压降），下面的二极管导通，输出电压钳位于 $-V_d$ 。

在正常工作下，两个二极管都不导通。

□ 输入配置

施密特触发器（TTL肖特基触发器）的作用是从低到高达多少阈值才会导通，从高到低低到多少才会关闭，中间不变。



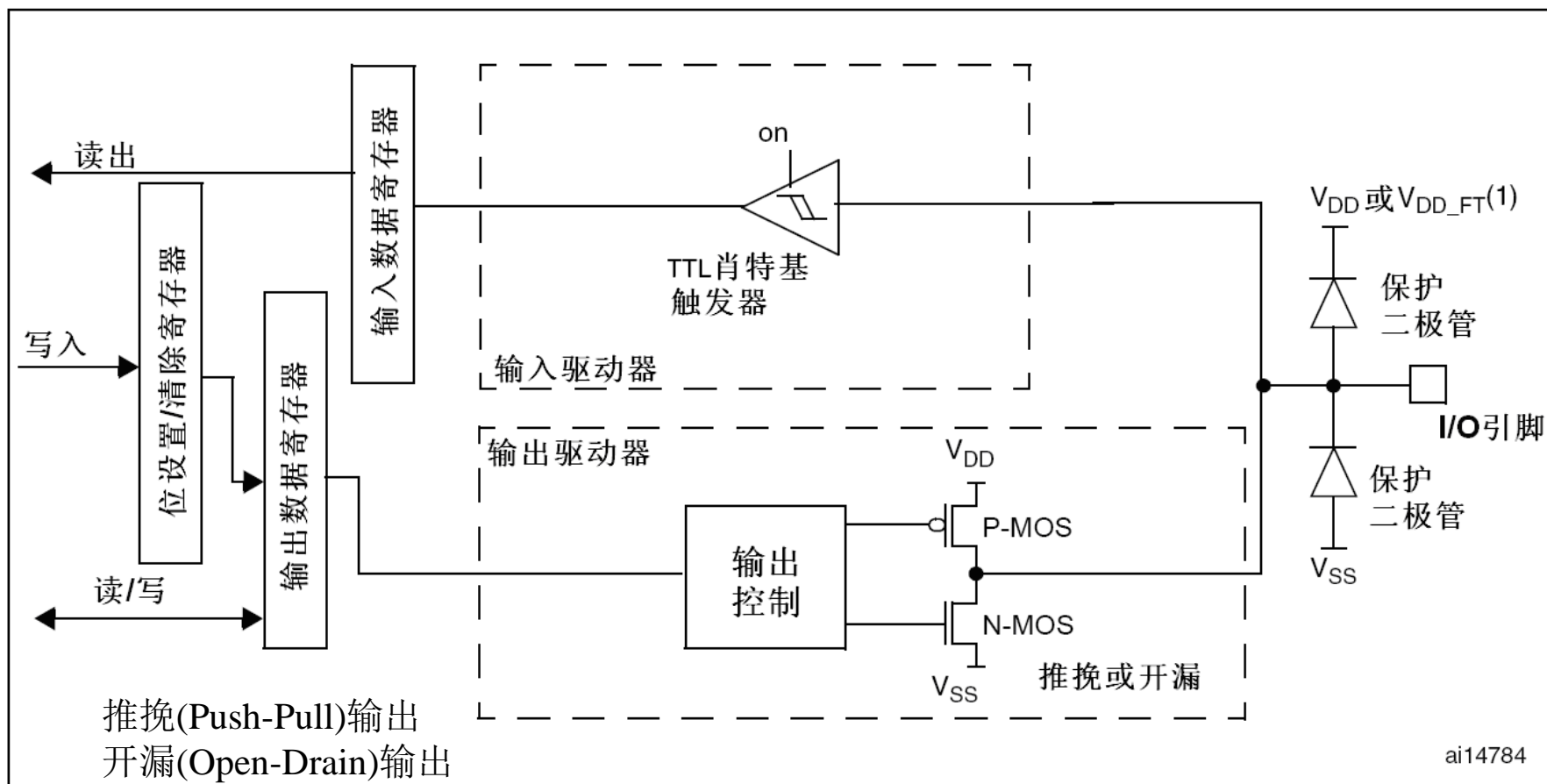
上拉(Pull-Up)输入: 接通了 $SW1$, I/O引脚悬空时经 $R1$ 接电源, 输入为1

下拉(Pull-Down)输入: 接通了 $SW2$, I/O引脚悬空时经 $R2$ 接地, 输入为0

浮空(Floating)输入: 需要外接上拉电阻或下拉电阻。

模拟输入: 用于A/D转换

□ 输出配置



推挽输出使用两个参数相同的三极管或MOSFET，电路工作时，两只对称的开关管每次只有一个导通，所以导通损耗小、效率高。上图，P-MOS和N-MOS只有一个连通，另一个断开。

开漏输出的输出端相当于三极管的集电极。要得到高电平状态需要上拉电阻才行。适合于做电流型的驱动,其吸收电流的能力相对强(一般20ma以内)。 上图，P-MOS一直断开。

复用推挽输出和复用开漏输出

端口配置低寄存器

(GPIOx_CRL) (x=A..E)

configuration register low

GPIOA EQU 0X40010800 ; GPIOA 地址
GPIOA_CRL EQU 0X40010800 ; 低配置寄存器

偏移地址： 0x00 复位值： 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

CNF	意义	CNF	意义	MODE[1:0]	意义
00	模拟输入	00	推挽式输出	00	保留(输入)
01	输入浮空	01	开漏输出	01	最大输出速度为10MHz
10	上拉/下拉输入	10	推挽式复用功能	10	最大输出速度为2MHz
		11	开漏复用功能	11	最大输出速度为50MHz

- GPIOA~GPIOE有16位
- 必须以字(32位)的方式操作这些外设寄存器。

端口配置高寄存器

(GPIOx_CRH) (x=A..E)

configuration register high

偏移地址: 0x04 复位值: 0x4444 4444 GPIOA_CRH EQU 0X40010804 ; 高配置寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

GPIOA的基地址: 0X40010800
GPIOB的基地址: 0X40010C00

位31:30 27:26 23:22 19:18 15:14 11:10 7:6 3:2	CNFy[1:0]: 端口x配置位(y = 0...7) (Port x configuration bits) 软件通过这些位配置相应的I/O端口, 请参考表17端口位配置表。 在输入模式(MODE[1:0]=00): 00: 模拟输入模式 01: <u>浮空输入模式(复位后的状态)</u> 10: 上拉/下拉输入模式 11: 保留 在输出模式(MODE[1:0]>00): 00: 通用推挽输出模式 01: 通用开漏输出模式 10: 复用功能推挽输出模式 11: 复用功能开漏输出模式
位29:28 25:24 21:20 17:16 13:12 9:8, 5:4 1:0	MODEy[1:0]: 端口x的模式位(y = 0...7) (Port x mode bits) 软件通过这些位配置相应的I/O端口, 请参考表17端口位配置表。 00: <u>输入模式(复位后的状态)</u> 01: 输出模式, 最大速度10MHz 10: 输出模式, 最大速度2MHz 11: 输出模式, 最大速度50MHz

复位期间和刚复位后, 复用功能未开启, I/O端口被配置成浮空输入模式:
CNFx[1:0]=01b, MODEx[1:0]=00b

□ 端口位配置表

[参考](#)

配置模式		CNF1	CNF0	MODE1	MODE0	PxODR寄存器		
通用输出	推挽(Push-Pull)	0	0	01 10 11		0 或 1		
	开漏(Open-Drain)		1			0 或 1		
复用功能输出	推挽(Push-Pull)	1	0			不使用		
	开漏(Open-Drain)		1			不使用		
输入	模拟输入	0	0	00		不使用		
	浮空输入		1			不使用		
	下拉输入	1	0			0		
	上拉输入					1		

需要注意的是，下拉输入和上拉输入是通过端口输出寄存器GPIOx_ODR来区分的。

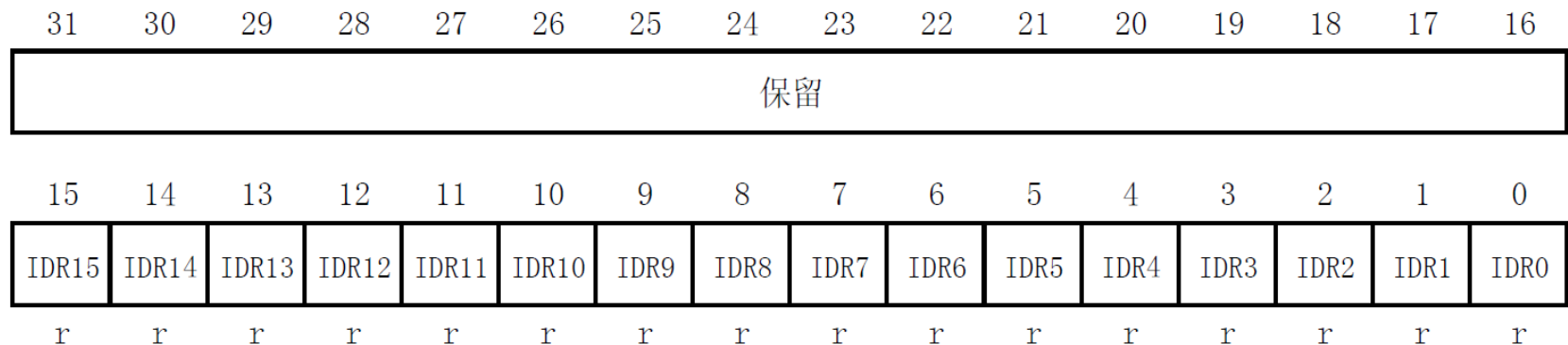
□ 输出模式位

MODE[1:0]	意义
00	保留
01	最大输出速度为10MHz
10	最大输出速度为2MHz
11	最大输出速度为50MHz

端口输入数据寄存器 (GPIOx_IDR) (x=A..E)

input data register

地址偏移: 0x08 复位值: 0x0000 XXXX



位31:16	保留，始终读为0。
位15:0	IDRy[15:0]: 端口输入数据(y = 0...15) (Port input data) 这些位为只读并只能以字(16位)的形式读出。读出的值为对应I/O口的状态。

GPIOA的基地址: 0X40010800

端口输出数据寄存器

(GPIOx_ODR) (x=A..E)

output data register

地址偏移：0Ch 复位值：0x0000 0000



位31:16	保留，始终读为0。
位15:0	ODRy[15:0]: 端口输出数据(y = 0...15) (Port output data) 这些位可读可写并只能以字(16位)的形式操作。 注：对GPIOx_BSRR(x = A...E)，可以分别地对各个ODR位进行独立的设置/清除。

GPIOA的基地址：0X40010800

GPIOA_ODR EQU 0X4001080C

端口位设置/清除寄存器

(GPIOx_BSRR) (x=A..E)

bit set/reset register

地址偏移: 0x10 复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

位31:16	BRy : 清除端口x的位y (y = 0...15) (Port x Reset bit y) 这些位只能写入并只能以字(16位)的形式操作。 0 : 对对应的ODRy位不产生影响 1 : 清除对应的ODRy位为0 注: 如果同时设置了BSy和BRy的对应位, BSy位起作用。
位15:0	BSy : 设置端口x的位y (y = 0...15) (Port x Set bit y) 这些位只能写入并只能以字(16位)的形式操作。 0 : 对对应的ODRy位不产生影响 1 : 设置对应的ODRy位为1

GPIOA的基地址: 0X40010800

GPIOA_BSRR EQU 0X40010810

端口位清除寄存器

(GPIOx_BRR) (x=A..E)

bit reset register

地址偏移：0x14 复位值：0x0000 0000



位31:16	保留。
位15:0	BRy : 清除端口x的位y (y = 0...15) (Port x Reset bit y) 这些位只能写入并只能以字(16位)的形式操作。 0: 对对应的ODRy位不产生影响 1: 清除对应的ODRy位为0

GPIOA的基地址：0X40010800

GPIOA_BRR EQU 0X40010814

端口配置锁定寄存器 (GPIOx_LCKR) (x=A..E)

地址偏移: 0x18 复位值: 0x0000 0000



GPIOA的基地址: 0X40010800

当执行正确的写序列设置了位16(LCKK)时，该寄存器用来锁定端口位的配置。位[15:0]用于锁定GPIO端口的配置。在规定的写入操作期间，不能改变LCKP[15:0]。当对相应的端口位执行了LOCK序列后，在下次系统复位之前将不能再更改端口位的配置。每个锁定位锁定控制寄存器(CRL, CRH)中相应的4个位。

位31:17	保留。
位16	<p>LCKK: 锁键 (Lock key) 该位可随时读出，它只可通过锁键写入序列修改。 0: 端口配置锁键位激活 1: 端口配置锁键位被激活，下次系统复位前GPIOx_LCKR寄存器被锁住。 锁键的写入序列： 写1 -> 写0 -> 写1 -> 读0 -> 读1 最后一个读可省略，但可以用来确认锁键已被激活。 注：在操作锁键的写入序列时，不能改变LCK[15:0]的值。 操作锁键写入序列中的任何错误将不能激活锁键。</p>
位15:0	<p>LCKy: 端口x的锁位y (y = 0...15) (Port x Lock bit y) 这些位可读可写但只能在LCKK位为0时写入。 0: 不锁定端口的配置 1: 锁定端口的配置</p>

APB2外设时钟使能寄存器

(RCC_APB2ENR)

偏移地址：0x18 复位值：0x0000 0000 访问：字，半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3 EN	USART1 EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	IOPG EN	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	保留	AFIO EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

RCC基地址：0X40021000

STM32F103C8T6

GPIO复用功能与设置

PA0	WKUP/USART2_CTS ⁽⁷⁾ ADC12_IN0/ TIM2_CH1_ETR ⁽⁷⁾
PA1	USART2_RTS ⁽⁷⁾ / ADC12_IN1/TIM2_CH2 ⁽⁷⁾
PA2	USART2_TX ⁽⁷⁾ / ADC12_IN2/TIM2_CH3 ⁽⁷⁾
PA3	USART2_RX ⁽⁷⁾ / ADC12_IN3/TIM2_CH4 ⁽⁷⁾
PA4	SPI1_NSS ⁽⁷⁾ /USART2_CK ⁽⁷⁾ /ADC12_IN4
PA5	SPI1_SCK ⁽⁷⁾ /ADC12_IN5
PA6	SPI1_MISO ⁽⁷⁾ / ADC12_IN6/TIM3_CH1 ⁽⁷⁾
PA7	SPI1_MOSI ⁽⁷⁾ / ADC12_IN7/TIM3_CH2 ⁽⁷⁾
PA8	USART1_CK TIM1_CH1 ⁽⁷⁾ /MCO
PA9	USART1_TX ⁽⁷⁾ TIM1_CH2 ⁽⁷⁾
PA10	USART1_RX ⁽⁷⁾ / TIM1_CH3 ⁽⁷⁾

PA11	USART1_CTS/USBDM CAN_RX ⁽⁷⁾ /TIM1_CH4 ⁽⁷⁾
PA12	USART1_RTS/USBDP/ CAN_TX ⁽⁷⁾ /TIM1_ETR ⁽⁷⁾
PB0	ADC12_IN8/TIM3_CH3 ⁽⁷⁾
PB1	ADC12_IN9/TIM3_CH4 ⁽⁷⁾
PB5	I2C1_SMBAL
PB6	I2C1_SCL ⁽⁷⁾ /TIM4_CH1 ⁽⁷⁾
PB7	I2C1_SDA ⁽⁷⁾ /TIM4_CH2 ⁽⁷⁾
PB8	TIM4_CH3 ⁽⁷⁾
PB9	TIM4_CH4 ⁽⁷⁾
PB10	I2C2_SCL/USART3_TX ⁽⁷⁾
PB11	I2C2_SDA/USART3_RX ⁽⁷⁾
PB12	SPI2_NSS/I2C2_SMBAL/ USART3_CK ⁽⁷⁾ /TIM1_BKIN ⁽⁷⁾
PB12	SPI2_NSS/I2C2_SMBAL/ USART3_CK ⁽⁷⁾ /TIM1_BKIN ⁽⁷⁾
PB13	SPI2_SCK/USART3_CTS ⁽⁷⁾ / TIM1_CH1N ⁽⁷⁾
PB14	SPI2_MISO/USART3_RTS ⁽⁷⁾ TIM1_CH2N ⁽⁷⁾
PB15	SPI2_MOSI/TIM1_CH3N ⁽⁷⁾

PC13 ⁽⁵⁾	TAMPER-RTC
PC14 ⁽⁵⁾	OSC32_IN
PC15 ⁽⁵⁾	OSC32_OUT
PD0	OSC_IN ⁽⁸⁾
PD1	OSC_OUT ⁽⁸⁾

所有功能都可以映射到指定的GPIO引脚

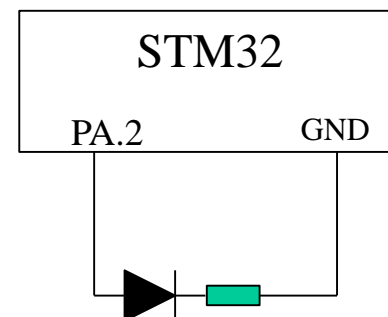
GPIO编程

□ GPIOA.2接LED灯闪烁

```
BIT2      EQU 0X00000004
LED2      EQU BIT2                ; LED2--PA.2
CFGAR     EQU 0x0300              ; PA.2: 推挽输出, 50MHz
```

```
GPIOA      EQU 0X40010800          ; GPIOA 地址
GPIOA_CRL  EQU 0X40010800          ; 低配置寄存器
GPIOA_CRH  EQU 0X40010804          ; 高配置寄存器
GPIOA_ODR  EQU 0X4001080C          ; 输出, 偏移地址0Ch
GPIOA_BSRR EQU 0X40010810          ; 低置位, 高清除偏移地址10h
GPIOA_BRR  EQU 0X40010814          ; 清除, 偏移地址14h
GPIOAEN     EQU 0X00000004          ; GPIOA时钟使能位
RCC_APB2ENR EQU 0X40021018          ; 时钟地址
```

```
STACK_TOP EQU 0X20002000
AREA RESET, CODE, READONLY          ; AREA不能顶格写
DCD STACK_TOP                        ; MSP主堆栈指针
DCD START                             ; 复位, PC初始值
ENTRY                               ; 指示开始执行
```



```

START                                     ; 标号必须顶格写，且无冒号
    LDR    R1, =RCC_APB2ENR    ; 0X40021018
    LDR    R0, [R1]
    LDR    R2, =GPIOAEN
    ORR    R0, R2
    STR    R0, [R1]                ; 使能GPIOA时钟

    MOV    R0, #CFGA            ; 0X0300 PA.2: 推挽输出, 50MHz
    LDR    R1, =GPIOA_CRL        ; 0X40010800
    STR    R0, [R1]
    NOP
    NOP
LOOP
    LDR    R1, =GPIOA_ODR
    LDR    R2, =LED2            ; 将PA.2输出高电平(亮灯)
    STR    R2, [R1]
    BL Delay
    LDR    R2, =0x0              ; 将PA.2输出低电平(灭灯)
    STR    R2, [R1]
    BL Delay
    B LOOP

```

Delay

PUSH {R0,R1,R2,LR}

MOVS R0,#0

MOVS R1,#0

MOVS R2,#0

DelayLoop0

ADDS R0,R0,#1

CMP R0,#330

BCC DelayLoop0

MOVS R0,#0

ADDS R1,R1,#1

CMP R1,#330

BCC DelayLoop0

MOVS R0,#0

MOVS R1,#0

ADDS R2,R2,#1

CMP R2,#15

BCC DelayLoop0

POP {R0,R1,R2,PC}

END

□ 按钮让LED灯亮灭

```

PULL0DOWN EQU 0x00010000 ; GPIOA_BSRR: bit 16:1- PA0OUT=0 pull down (默认)
PULL0UP   EQU 0x00000001 ; GPIOA_BSRR: bit 0: 1- PA0OUT=1 pull up

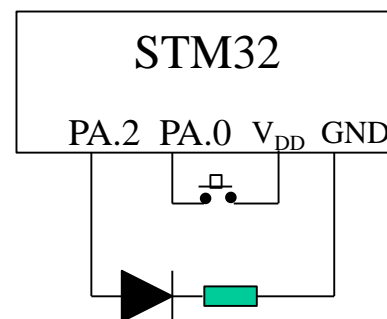
LED2ON     EQU 0x00000004 ; GPIOA_BSRR: bit2:1-PA2 on   bit 16:1- PA0 pull down
LED2OFF    EQU 0x00040000 ; GPIOA_BSRR: bit18:1-PA2 off bit 16:1- PA0 pull down
CFG_A      EQU 0x0308      ; PA.2: 推挽输出, 50MHz; PA.0 下拉输入
    
```

```

GPIOA      EQU 0X40010800 ; GPIOA 地址
GPIOA_CRL  EQU 0X40010800 ; 低配置寄存器
GPIOA_CRH  EQU 0X40010804 ; 高配置寄存器
GPIOA_IDR  EQU 0X40010808 ; 输入
GPIOA_ODR  EQU 0X4001080C ; 输出
GPIOA_BSRR EQU 0X40010810 ; 低置位, 高清除
GPIOA_BRR  EQU 0X40010814 ; 清除, 偏移地址14h
GIOPAEN    EQU 0X00000004 ; GPIOA时钟使能位
RCC_APB2ENR EQU 0X40021018 ; 时钟接口
    
```

```

STACK_TOP EQU 0X20002000
    AREA RESET, CODE, READONLY ; AREA不能顶格写
    DCD STACK_TOP               ; MSP主堆栈指针
    DCD START                   ; 复位, PC初始值
    ENTRY
START                                ; 所有的标号必须顶格写, 且无冒号
    
```



按下亮，断开灭


```
LDR R1, =RCC_APB2ENR
LDR R0, [R1]
LDR R2, =GPIOAEN
ORR R0, R2
STR R0, [R1] ; 使能GPIOA时钟
```

```
MOV R0, #CFGA
LDR R1, =GPIOA_CRL
STR R0, [R1]
```

```
LDR R1, =GPIOA_BSRR
LDR R2, =PULL0DOWN
STR R2, [R1]
```

LOOP

```
LDR R1, =GPIOA_IDR
LDR R2, [R1]
TST R2, #1
BEQ OFF
LDR R1, =GPIOA_BSRR
LDR R2, =LED2ON
STR R2, [R1]
B LOOP
```

OFF

```
LDR R1, =GPIOA_BSRR
LDR R2, =LED2OFF
STR R2, [R1]
B LOOP
```

END