General Purpose IOs

通用并行I/O接口



GPIO main features

- Bit set and reset register (GPIOx_BSRR) for bitwise write access to GPIOx_ODR
- Locking mechanism (GPIOx_LCKR) provided to freeze the I/O configuration
- > Speed selection for each I/O
- Fast toggle capable of changing every two clock cycles





- ① 输出驱动打开,推挽/开 漏取决于GPIOx_OTYPER ② Schmit触发输入激活 ③上/下拉电阻是否激活 取决于GPIOx_PUPDR
- ▶ 根据P-MOS是否激活,可分为 □ 推挽输出、开漏输出
- ➢ 每个AHB1时钟周期,采样一次I/O引脚上的电平,送入输入数据寄存器

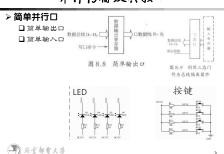
GPIO输出模式 (MODER=01)

- 口读取GPIO_IDR可得到引脚的当前电平 ·读取输出数据寄存器GPIO_ODR只能得到上次在推挽模式下
- > 读取输出数据奇存器GPIO_ODR只能得到上次在推挽模式下输出的值
- 口 开漏模式下软件写1, I/O引脚电平取决于外部电路

GPIO Configuration Modes

MODER(i) [1:0]	OTYPER(i) [1:0]		DR(i) :0]	I/O configuration
		0	0	Output Push Pull
	0	0	1	Output Push Pull with Pull-up Output Push Pull with Pull-down
01		0	0	Output Open Drain
	1	0	1	Output Open Drain with Pull-up
		1	0	Output Open Drain with Pull-down
		0	0	Alternate Function Push Pull
	0	0	1	Alternate Function PP Pull-up
10		1	0	Alternate Function PP Pull-down
10		0	0	Alternate Function Open Drain
	1	0	1	Alternate Function OD Pull-up
		1	0	Alternate Function OD Pull-down
		0	0	Input floating
00	x	0	1	Input with Pull-up
		1	0	Input with Pull-down
11	x	3	x	Analog mode
M 11 2 to				/O speed is configurable through OSPEEDR Iz or 50MHz

并行传输及其接口

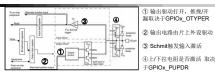


GPIO functional description

- Each port bit of the general-purpose I/O (GPIO) ports can be individually configured by software in several modes:
- Input floating
- ☐ Input pull-up ☐ Input-pull-down
- □ Analog
- June or pull-down carability
- ☐ Output open-drain with pull-up or pull-down capability☐ Output push-pull with pull-up or pull-down capability
- □ Output push-pull with pull-up or pull-down capability
 □ Alternate function push-pull with pull-up or pull-down
- Alternate function push-pull with pull-up or pull-down capability
- Alternate function open-drain with pull-up or pull-down capability



可选功能模式(MODER=10)



- ▶ 輸出信号仅来自片上数字外设,根据P-MOS是否激活可分为 □ 推挽輸出、开漏輸出
- ➢ I/O引脚上的信号可以输入到片上数字外设和GPIO_IDR
 □ 每个AHB1时钟周期采样一次I/O引脚上的电平,送入输入数据寄存
 - □ 读取GPIO_IDR可得到引脚上的电平状态



可选功能多路选择器

- 每个引脚都有一个多路复用选择器,决定哪一个外设功能连到该引脚
- ▶每个多路选择器有16路输入可供选择
- □ AFO: 系统功能(复位后复用选择器的默认连接)
- □ JTAG/SWD, MCO1/2, RTC_50Hz □ AF1~13: 各种数字外设功能
- □ AF14: 保留

· 好京都電大學

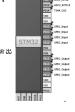
- □ AF15: Cortex-M4 EVENTOUT
- ▶ 每个数字外设功能可以映射到多个I/O引脚□ 具体映射关系参考数据手册(可编程功能映射表)
 - □ 映射以单个引脚为单位

GPIO接口

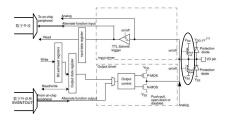
- ▶ 通用輸入輸出接口GPIO
 - □ 用户可以通过编程灵活的对接□进行控制
- ➤ GPIO被分成多组,每组最多有16个 引脚
 - ☐ STM32F407IGT6:

图 北京都電大學

- 図GPIOA、GPIOB、GPIOC 至 GPIOI共9个GPIO端□(port)
- ☑ 所有GPIO引脚都有基本的输入输出 功能和可选功能(alternate function)



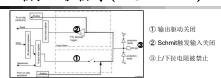
GPIO结构框图



复位之后,JTAG引胸是"输入上/下拉"模式,其他I/O引脚默认是"浮空输入"模式

對注章都定大學

模拟功能模式(MODER=11)



- ➢ Schmitt触发器关闭
 □ 该引脚功耗为0
 - □以与阿以托内∪
- 触发器輸出强制为常数0
- ▶ 读取输入数据寄存器始终得到"0"
- ▶ 模拟功能配置下, I/O引脚不是5V友好的

图 北京都會大學

Alternate function mapping



3 北京都電大學

GPIO main features

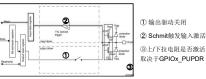
➤ Up to 16 I/Os under control

(alternate function input)

- Highly flexible pin multiplexing allows the use of I/O pins as GPIOs or as one of several peripheral functions
- □ Alternate function input/output selection registers (at most 16 AFs per I/O)
- > Analog function
- Output states: push-pull or open drain + pullup/down
- Output data from output data register (GPIOx_ODR) or peripheral (alternate function output)
- Input states: floating, pull-up/down, analog
 Input data to input data register (GPIOx_IDR) or peripheral



输入模式 (MODER=00)



- ▶ 根据上下拉电阻是否激活,可以分为□ 浮空输入、上拉输入、下拉输入
- ➢ GPIO模块在AHB1总线上,每个AHB1时钟周期,采样一次I/O引脚上的电平,送入输入数据寄存器
- □ 读取GPIO IDR可得到引脚上的电平状态



GPIO引脚模式小结

		输出信号	子来自				
	MODER	OTYPER: 配置推挽输出 或开漏输出	OSPEEDR: 配置GPIO输 出带宽	引脚电平信 息输入到	施密特触发器	PUPDR	
输入模式	00			GPIO_IDR			
输出模式	01	GPIO.	_ODR	GPIO_IDR	激活	配置内部弱 上拉、弱下	
可选功能 模式	10	片上数字外设		外设和 GPIO_IDR	在形态	拉或浮空	
模拟功能	11	片上模拟外设:	ADC/DAC	ADC/DAC	关闭	关闭	

- > 橘黄色框:对应寄存器无效
- > 红色框: 芯片以AHB1时钟频率采样引脚电平状态,送入输入数据寄存
- 蓝色框: 只有这两个模式输出驱动被打开,输出信号分别来自输出数据 寄存器和片上数字外设,由此才需要配置:
- 推挽或开漏输出

23

输出带宽(2MHz、25MHz、50MHz和100MHz)

| The color of the

Alternate function mapping

對京都電大學

30

Alternate function mapping

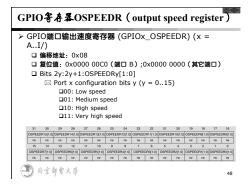
GPIO引脚分布 PortA(16) PortB (16) PortC (16) PortD (16) PD.2 PortE (16) PortF (16) PortG (16) PH0/1(OSC) PH0/1(OSC) PH0/1(OSC) PortH (16) Portl (12) 总共引脚数目 48+1+2=51 80+2=82 112+2=114 128+12=140 GPIO占芯片所有引脚的80% 左右 對京都電大學



GPIO 客 存 器-ODR (output data register) > GPIO端口输出数据寄存器 (GPIOx_ODR) (x = A..I) □偏移地址: 0x14 □复位值: 0x0000 0000 □ Bits 31:16 Reserved, must be kept at reset value. □ Bits 15:0 ODRy □ Port output data (y = 0..15) □ can be read and written by software

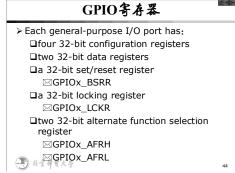
I/O引脚的使用: GPIO - 能置方向 - 输入方向 - 输入方向 - 输入方向 - 输入方向 - 输入管脚 - 配置上/下拉电阻 - 区配上/下拉电阻 - 区配上/下拉电阻 - 区配上/下拉电阻 - 区面上/下拉电阻 - 区面上/下近和输出速度 - GPIOx_OTYPER、GPIOx_PUPDR、GPIOx_OSPEEDER

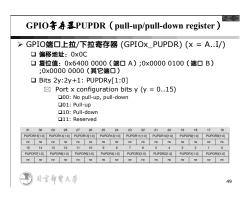
GPIO寄存器	
➤ Each general-purpose I/O port has: □four 32-bit configuration registers □GPIOx_MODER □GPIOx_OTYPER □GPIOx_OSPEEDR □GPIOx_PUPDR □two 32-bit data registers □GPIOx_IDR □GPIOx_ODR	
③ 北京都電大學	43



G	GPIO舎存器BSRR (bit set/reset register)																
> GPIO端口置位/复位寄存器 (GPIOx_BSRR) (x =																	
AI)																	
→ (1.17) □ 偏移地址: 0×18: 复位值: 0×0000 0000																	
, , , , , , , , , , , , , , , , , , , ,																	
\square Bits 31:16 BRy: Port x reset bit y (y = 015)																	
, , , , ,																	
□ 0: No action on the corresponding ODRx bit																	
□ 1: Resets the corresponding ODRx bit																	
									•		_						
		\simeq	wri	te-	only	/ ar	d c	an	be	acc	ess	ed	in۱	vor	d, r	ıalt	-
			wo	rd d	or b	yte	mo	ode									
						to			h:40				h a				
							une	se	DILS	re	turi	is t	ne	VdI	ue		
				0x0	000	0											
	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	890	
	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	Į.
- 4	BS15	BS14	BS13 W	BS12	BS11 W	BS10	BS9 W	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	B90	
7.11																	1

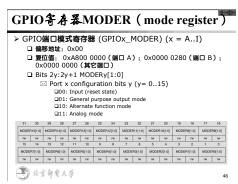
I/O引脚的使用:外设不为能 → 模拟外设ADC/DAC □配置成模拟模式 □关闭上/下拉电阻 □ GPIOX_PUPDR = 00 → 数字外设 □配置成可选功能模式 □ 经报告外决漏、上/下拉和输出速度 □ GPIOX_OTYPER、GPIOX_PUPDR、GPIOX_OSPEEDER □配置多路选择器,把使用到的外设连到对应引脚 □ GPIOX_AFRL或令GPIOX_AFRH

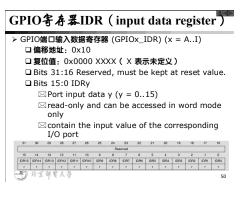


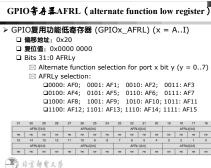


GP	Ю	舎.	存	赛.F	SF	RR	(1	oit	set	/re	set	re	gis	tei	•)
➤ GPIO端口置位/复位寄存器 (GPIOx_BSRR) (x = AI)															
□ 偏移地址: 0×18; 复位值: 0×0000 0000															
□ Bits 31:16 BRy: Port x reset bit y (y = 015)															
☐ Bits 15:0 BSy: Port x set bit y (y= 015)															
, , , , ,															
□ 0: No action on the corresponding ODRx bit															
	D	⊲1:	Se	ts t	he (corr	esp	one	lina	OD	Rx	bit			
		⊴w							_				ord	ha	ılf_
	2				byte			DC	acc	C33	cu i	vv	oru	, 110	
					•							_			
		L	JA r	ead	to th	iese	bits	reti	ırns	the	valu	ie 0x	KUUU	10	
	I No	ote:	If I	ooth	n BS	Sx a	nd	BR	an	e se	et. E	3Sx	has	S	
	nr	iorit	tv								,				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BA7	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	B96	BS5	BS4	BS3	BS2	BS1	BS0
w	w	W	w	w	w	w	w	W	w	w	w	w	w	w	w

I/Os special considerations ➤ During and just after reset: □ the alternate functions are not active and the I/O ports are configured in input floating mode □ the debug pins (JTAG/SWD) are in AF pull-up/pull-down after reset: ☑ PA13: JTMS/SWDIO ☑ PA14: JTCK/SWCLK ☑ PA15: JTDI ☑ PB3: JTDO ☑ PB4: NJTRST







59



□ 偏移地址: 0x24

□ 复位值: 0x0000 0000 ☐ Bits 31:0 AFRHy

 \bowtie Alternate function selection for port x bit y (y = 8..15) ☑ AFRHy selection:

□0000: AF0; 0001: AF1; 0010: AF2; 0011: AF3 □0100: AF4; 0101: AF5; 0110: AF6; 0111: AF7 □1000: AF8; 1001: AF9; 1010: AF10; 1011: AF11 □1100: AF12; 1101: AF13; 1110: AF14; 1111: AF15

→ 北京都電大學

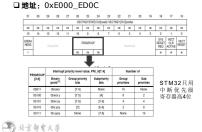
中断和异常向量表

- > STM32F407中断和异常向量
 - □ CM4内核支持256个中断,含16个内核中断和240个外部 中断, STM32F4只使用了10个内部异常和82个外部中断 (STM32F4只实现了Cortex-M4内核中断的部分功能)
- ▶ 每一类异常源对应一个特定的入口地址, 异常或中断 发生时,处理器会把PC设置为该特定地址,也即异 常向量
- > 异常向量按照优先级排列后组成异常向量表



STM32实现的中断优先级寄存器

▶ 应用程序中断及复位控制器SCB_AIRCR



系统配置控制器 (SYSCFG)

- >一组系统配置控制器,主要用途:
 - □ 重映射存储器到代码起始区域 □管理连接到GPIO□的外部中断
 - □ 管理系统的可靠性特性
- > SYSCFG 外部中断配置寄存器 1-4



STM32F4的中断体系结构

- 中断:微处理器外部产生,通过中断通道送入处理器内部,一般由硬件
- 异常:通常在微处理器内部发生,大多由软件引起,如:
- □ 除法出错异常 □ 特权调用显觉 片内外设 CORTE X-M4 定时器 嵌套向 内核指 令执行 单元 量中断控制器 I2C 内核 EXTI 系统定时器 A 北京都會 73

中断和异常向量表

号	优先级	优先级 类型	名称	说明	地址
				保留(实际存的是 MSP 地址)	0X0000 0000
	-3	固定	Reset	复位	0X0000 0004
	-2	固定	NMI	不可屏蔽中断。RCC 时钟安全系统 (CSS) 连接到 NMI 向量	0X0000 0008
	-1	固定	HardFault	所有类型的错误	0X0000 000C
-	0	可编程	MemManage	存储器管理	0X0000 0010
	1	可编程	BusFault	预取指失败,存储器访问失败	0X0000 0014
	2	可编程	UsageFault	未定义的指令或非法状态	0X0000 0018
	-	-	-	保留	0X0000 001C- 0X0000 002B
	3	可编程	SVCall	通过 SWI 指令调用的系统服务	0X0000 002C
	4	可编程	Debug Monitor	调试监控器	0X0000 0030
	-	-	-	保留	0X0000 0034
	5	可编程	PendSV	可挂起的系统服务	0X0000 0038
	6	可编程	SysTick	系统嘀嗒定时器	0X0000 003C
0	7	可编程	-	窗口看门狗中断	0X0000 0040
1	8	可编程	PVD	连接 EXTI 线的可编程电压检测中断	0X0000 0044
2	9	可编程	TAMP STAMP	连接 EXTI 线的入侵和时间戳中断	0X0000 0048
中间 79	部分4	全略,详情 可编程	请参考 STM32F4xx CRYP	(中文参考手册)第十章-中断和事件-向 CRYP加密全局中断	量表部分 0X 0000 017C
80	87	可编程	HASH RNG	哈希和随机数发生器全局中断	0X 0000 0180
81	88	可编程	FPU	FPU 全局中断	0X 0000 0184

中断优先级与嵌套中断

- ▶ 抢占优先级的级别高于响应优先级 □ 数值越小所代表的优先级就越高
- > 同一时刻发生的中断,优先处理优先级较高的中断
- > 高抢占优先级中断可以打断正在进行的低抢占优先级 中断
- □ 抢占优先级相同的中断,高响应优先级中断不可以打断低
- 响应优先级的中断 □ 抢占优先级相同就看响应优先级,同样数值越小优先级越
- □ 如果两个中断的抢占优先级和响应优先级都相同,则看哪 个中断先发生就先执行
 - ☑ 如果同时发生,则优先处理编号较小的中断



STM32 中断系统

- ▶ EXIT支持多达23个事件/中断请求,除16跟GPIO中 断外,另外七根EXTI线连接方式如下:
 - □ EXTI线16连接到PVD输出
 - □ EXTI线17连接到RTC闹钟事件
 - □ EXTI线18连接到USB OTG FS唤醒事件
 - □ EXTI线19连接到以太网唤醒事件
 - □ EXTI线20连接到USB OTG HS唤醒事件
 - □ EXTI线21连接到RTC入侵和时间戳事件
 - □ EXTI线22连接到RTC唤醒事件

· 北京都電大學

中断管理寄存器

- ▶ ICER: 中断禁止(清除使能)寄存器
- > ISPR: 中断挂起寄存器
- □ 中断处理完成后应该清除挂起,表示已处理

70.30	寄存器	CMSIS-Core 符号	动能
OMEDODETIC	中斯設置使維寄存器	NVIC>BER [0] ~ NVIC>BER [7]	可 1 设置使能
0xE000E180-	中斯清除使能寄存器	NVIC-> KER [0] ~ NVIC-> KER [7]	写工資路使能
0xE000E200~ 0xE000E21C	中新设置挂起寄存器	NVIC-> ISPR [0] ~ NVIC-> ISPR [7]	写:设置挂起状态
0xE000E280~ 0xE000E29C	中斯清除社起寄存器	NVIC>ICPR [0] ~ NVIC>ICPR [7]	写1清除性起状态
0xE000E300~ 0xE000E31C	中新活跃位寄存器	NVIC->IABR [0]~ NVIC->IABR [7]	折断状态位。只读
0xE000E4EF	中新优先级寄存器	NVIC>IP[0]~ NVIC>IR[219]	每个中期的中期优先级(8 位宽)
0xE000EF00	软件触发中断寄存器	NVIC->STIR	写中斯偏号设置相应中断的技志 状态

- ➤ ISER: 中断使能寄存器

> STM32F4中断的优先级:

□ 优先级越小优先级越高

□ F407: x=0...81

A 北京都電大學

□用于配置外部中断的优先级

□3个固定的优先级,都是负值,不能改变

□ 16个可编程优先级, 用4个bit位表示

▶ NVIC中断优先级寄存器NVIC_IPRx

0~255,数值越小,优先级越高 ⋈ F407: 只使用高4bit

- ➤ ICPR: 清除中断挂起寄存器

				□ 若中断发生但无法
	表7.9 美	子中數控制的 NVIC 寄存	B 列章	被立即处理(如处
70.1E	寄存器	CMSIS-Core 符号	70 MB	理器已经在处理另
EDGOETIC	中斯設置使維寄存器	NVIC > ISER [0] ~ NVIC > ISER [7]	可1设置使能	一个更高或同等优
E000E180~ E000E19C	中斯清除使能寄存器	NVIC-> KER [0] ~ NVIC-> KER [7]	写:資務快能	先级的中断,或者
E000E200~ E000E21C	中新设置技起寄存器	NVIC>ISPR [0] ~ NVIC>ISPR [7]	写:设置挂起状态	中断被某个中断屏
EDGOEDSO~ EDGOEDSC	中斯雷染丝起寄存器	NVIC>ICPR [0] ~ NVIC>ICPR [7]	写工清除性起状态	蔽寄存器给屏蔽掉).则在其他中断
E000E300~ E000E31C	中新活跃在寄存器	NVIC->IABR [0]~ NVIC->IABR [7]	折断状态位。只读	外理结束前或中断
E000E4E0~ E000E4EF	中新优先级寄存器	NVIC>IP[0]~ NVIC>IR[219]	每个中联的中新优先级(8 位宽)	屏蔽被清除前,挂
E001EF00	软件触发中断寄存器	NVIC>STIR	写中新编号设置相应中新的技起 状态	起请求会一直保待
- W	22457	4		7-

中断的优先级

□ 宽度为 8bit, 原则上每个外部中断可配置的优先级为

> 挂起状态: 中断被置于

会进行处理

一种等待处理器处理的

□ 有些情况下 , 处理

器在中断挂起时就

- ▶ IABR: 中断活跃位寄存器
 - □ 每个外部中断都有一个活跃状态位,当处理器正在处理该中断时,该 位会被置1

中断管理寄存器

- □ 只读
- ▶ IPR: 中断优先级寄存器
- > STIR: 软件触发中断寄存器

地址	寄存器	CMSIS-Core 符号	功能
0xE000E280~ 0xE000E29C	中断清除挂起寄存器	NVIC->ICPR [0] ~ NVIC->ICPR [7]	写1清除挂起状态
0xE000E300~ 0xE000E31C	中断活跃位寄存器	NVIC->IABR [0]~ NVIC->IABR [7]	活跃状态位,只读
0xE000E400~ 0xE000E4EF	中断优先级寄存器	NVIC->IP [0]~ NVIC->IR [239]	每个中断的中断优先级(8位宽)
0xE000EF00	软件触发中断寄存器	NVIC->STIR	写中斯编号设置相应中断的挂起 状态

對京都官大學

75

中断的优先级

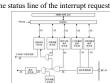
- ➤ STM32F4中断的优先级分组:
 - □ STM32F4将中断分为5个组: 组0~4
 - □ 分组的设置由SCB->AIRCR寄存器的 bit10~8定义
 - □ 中断优先级寄存器NVIC IPRx中的4位,分为抢占优先级 和响应优先级
 - 应 抢占优先级在前,响应伏先级在后
 - 应两个优先级各占几位根据SCB->AIRCR中的中断分 组设置决定

	组	AIRCR[10: 8]	bit[7: 4]分配情况	分配结果
	0	111	0: 4	0 位抢占优先级,4 位响应优先级
	1	110	1: 3	1 位抢占优先级,3 位响应优先级
	2	101	2: 2	2 位抢占优先级, 2 位响应优先级
	3	100	3: 1	3 位抢占优先级,1 位响应优先级
П	4	011	4: 0	4 位抢占优先级,0 位响应优先级

A 北京都電大學

External interrupt/event controller (EXTI)

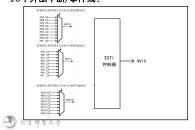
- > consists of up to 23 edge detectors for generating event/interrupt requests
- > Each input line can be independently configured to select the type (interrupt or event) and the corresponding trigger event (rising or falling or both)
- Each line can also masked independently
- > A pending register maintains the status line of the interrupt requests



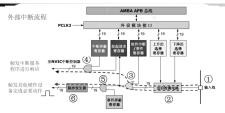
A 北京都會大學

STM32 中断系统

▶ Stm32F4xx将多达140个GPIO通过以下方式连接到 16个外部中断/事件线:



外部中断/事件框图



> An interrupt/event request can also be generated by interrupt/event register software by writing a '1' in the software

到 北京都電大學

本章结束

102