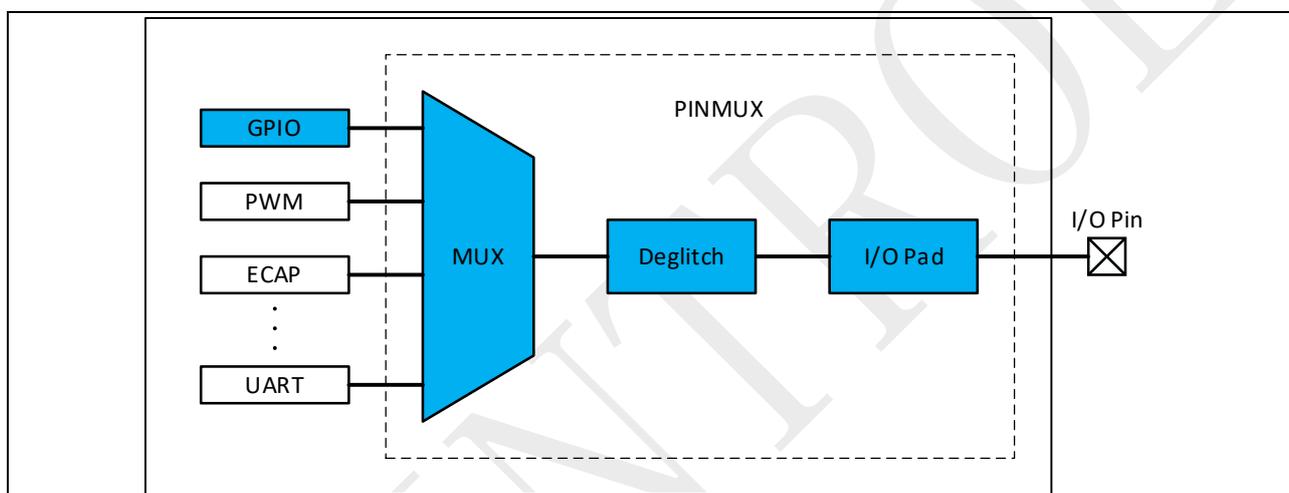


概述

GPIO（通用输入输出）是一种用于与外部设备进行数字通信的接口，它允许我们通过控制电平的高低来读取输入信号或控制输出信号，SPC1168 的 I/O 控制的结构由两部分组成（PINMUX 以及 GPIO），其结构框图如下图所示。



目录

1	PINMUX.....	6
2	GPIO.....	8
3	配置 GPIO 功能.....	9
4	配置 GPIO 输入.....	10

SPIN TROL

图片列表

图 1-1: PINMUX 结构框图.....	6
图 2-1: GPIO 结构框图.....	8
图 4-1: PINMUX 结构框图.....	10
图 4-2: GPIO 结构框图.....	10

SPIN TROL

版本历史

版本	日期	作者	状态	变更
A/0	2023-06-15	CanChai	Outdated	首次发布。
C/0	2024-03-26	Jiali Zhou	Released	修改排版格式。

SPIN TROL

术语或缩写

术语或缩写	描述
/	/

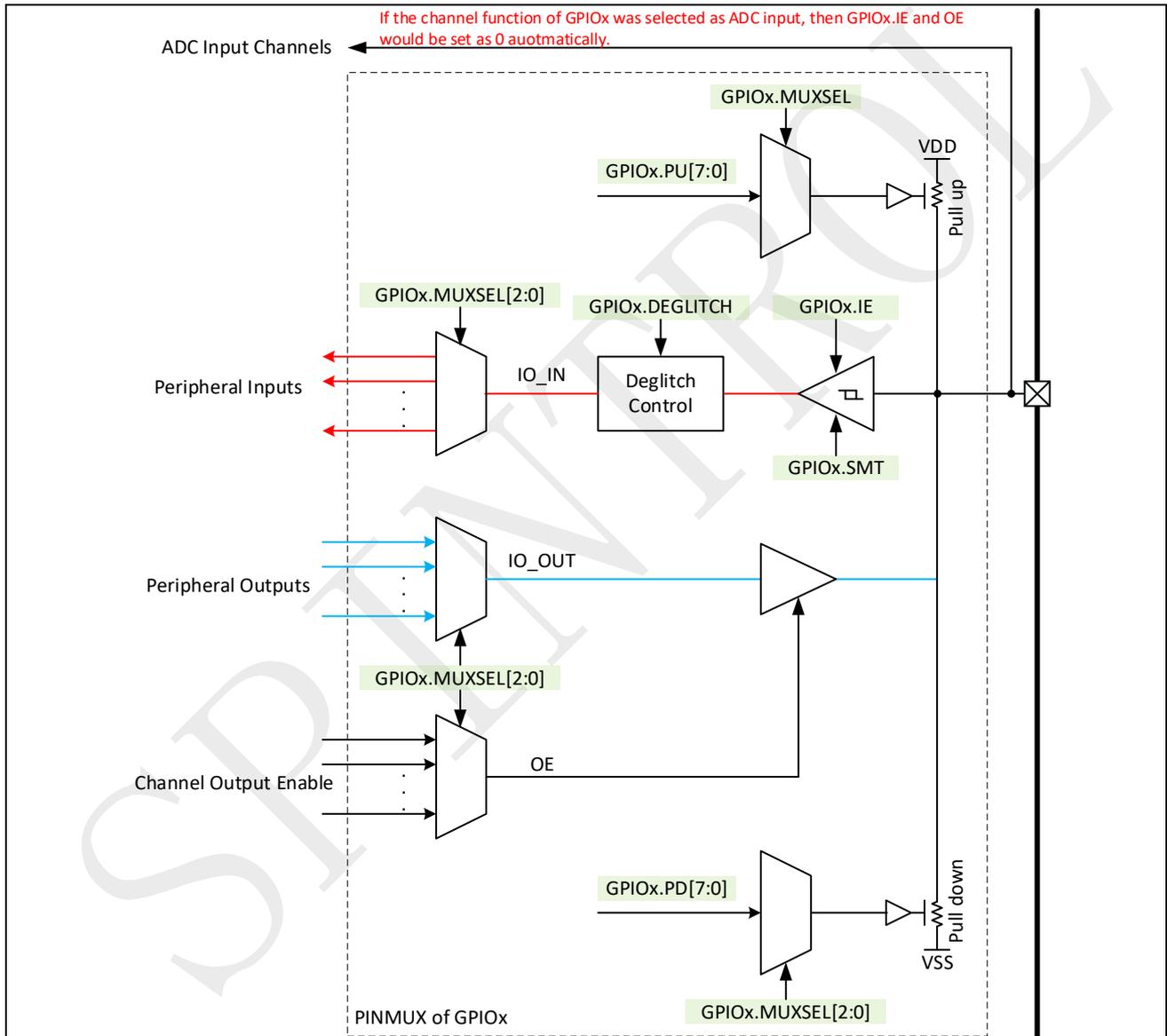
SPIN TROL

1 PINMUX

PINMUX 功能有：

- I/O Pad: 上拉和下拉控制，输入/输出控制，施密特输入控制以及输出驱动强度控制；
- Deglitch: 输入消抖控制；
- MUX: 引脚多路复用；

图 1-1: PINMUX 结构框图



PINMUX 的结构框图如图 1-1 所示，每个 I/O 引脚的多路选择器支持 8 个功能通道，其中红色信号通路代表着 PIN 输入到外设的信号（其中需要注意，PIN 脚输入到 ADC 外设的信号走的是如图中文字标识的信号通路），蓝色信号通路代表着外设输出到 PIN 脚的信号。

在此需要特别提醒的是，GPIO 也属于外设，同样也与 PINMUX 结构链接。所以由此可以看出，实际上 PIN 脚（在这篇文档中称为 I/O）直接与 PINMUX 链接，而 GPIO 与 PINMUX 链接，这个概念需要明确。

从图 1-1 中可以看出，I/O 在 PINMUX 结构中，可以设置上拉及下拉功能，当 I/O 引脚处于输入模式下，输出缓冲器禁用，并处于高阻态；可以通过配置毛刺信号抑制模块或者施密特触发器模块对输入信号进行过滤。

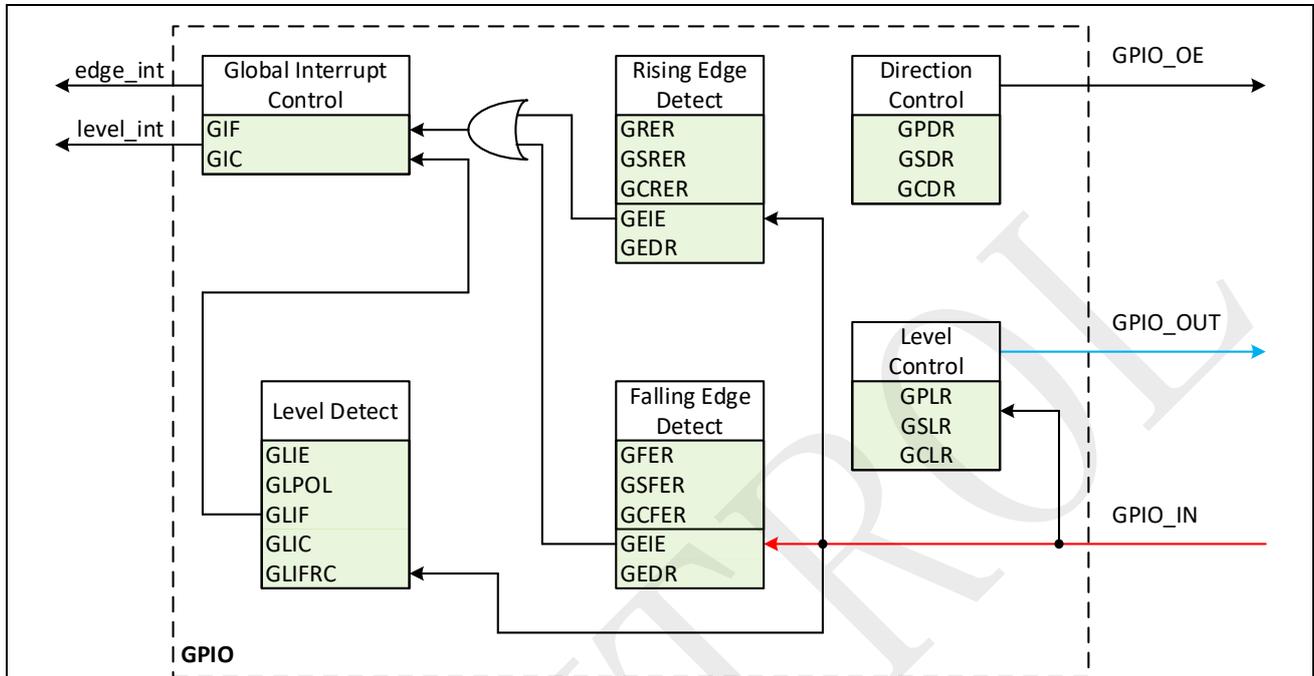
在此需要特别提出的是：如果 I/O 选择为 ADC 输入功能，那么 GPIOx.IE 和 OE 会自动设置为 0。

SPIN
TROL

2 GPIO

GPIO 模块的结构框图，如图 2-1 所示。

图 2-1: GPIO 结构框图



从图 2-1 中可以看出，GPIO 可以设置为输入，也可被设置为输出，图中红色信号为来自 PINMUX 结构的信号，蓝色部分为送往 PINMUX 结构的信号。GPIO 引脚被封为两组：GPIO_PORT0 和 GPIO_PORT1。每组分为 32bits 宽度，支持位控制。GPIO_PORT1 中仅使用了低 9 位。

GPIO 引脚默认为输入模式。可以通过 GPIO 方向寄存器（GPDR0/GPDR1）或者 GPIO 设置输出寄存器（GSDR0/GSDR1）或者 GPIO 清除输出寄存器（GCDR0/GCDR1）配置输入或者输出模式。GSDR 和 GCDR 是位控制寄存器，相应位写 0 无效，写 1 使能。

当 GPIO 被配置为输出，可以通过 GPIO 电平寄存器（GPLR0/GPLR1）或者 GPIO 设置电平寄存器（GSLR0/GSLR1）或者 GPIO 清除电平寄存器（GCLR0/GCLR1）配置输出高电平或者低电平。GSLR 和 GCLR 是位控制寄存器，相应位写 0 无效，写 1 使能。

注意： 无论向 GPLR 寄存器或者 GPDR 寄存器写入任何值，读取的 GPLR 值只反映当前引脚的状态（高电平或者低电平）。

在 SPC1168 中，如果 GPIO31 ~ GPIO33 没有绑定到芯片外部引脚，那么 GPIO31 和 GPIO32 在芯片内部保持浮空状态，而 GPIO33 在芯片内部被连接到 DVDD。因此，GPLR0[31]和 GPLR1[0]的值是不确定的，GPLR1[1]则总是高电平。另外，GPIO33 不能被配置为输出模式。

3 配置 GPIO 功能

假设需要将 GPIO 输出为高电平，需要执行如下步骤：

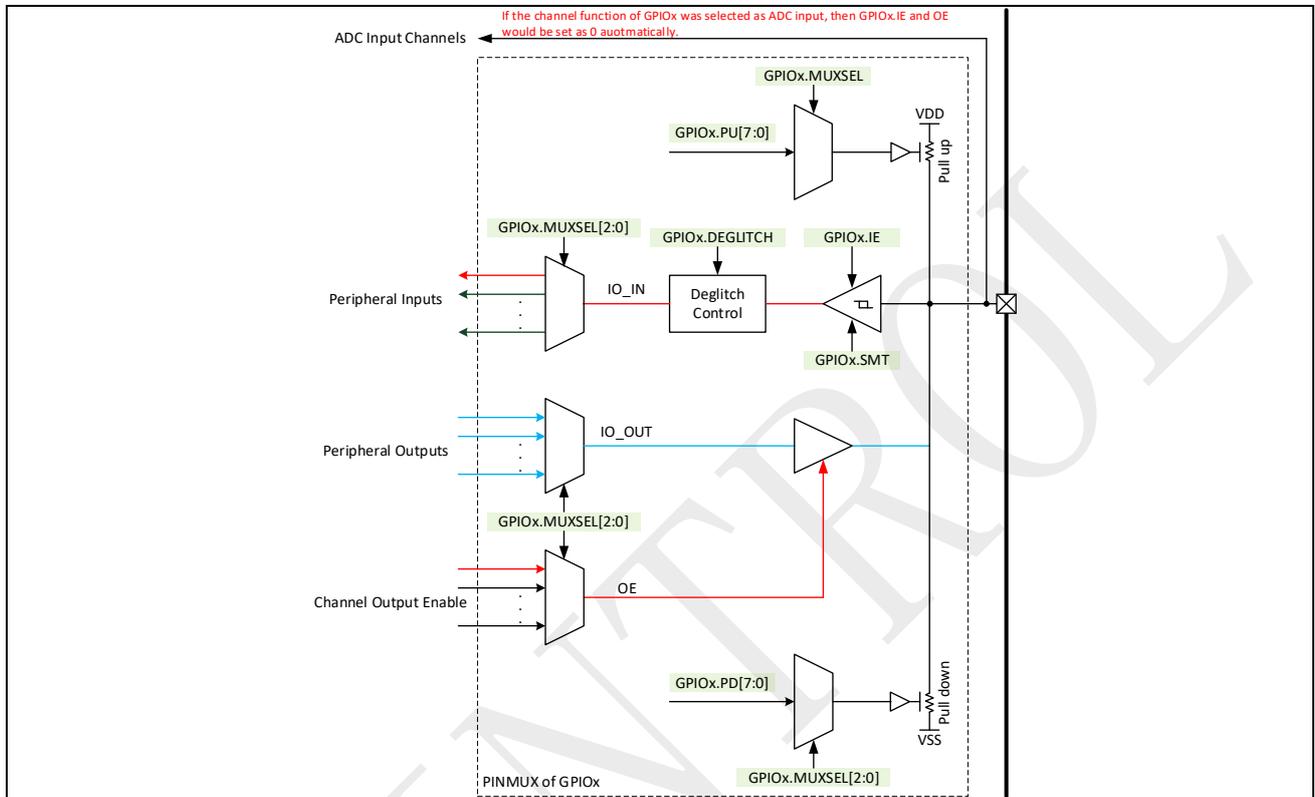
- PINMUX 中将 I/O 通路切到 GPIO 外设，可以通过设置图 1-1 中的 MUXSEL 位域完成，也可直接调用 SDK 中的 GPIO_SetPinChannel()接口完成；
- 在 GPIO 中设置 GPIO 为输出功能，可以通过设置图 2-1 中的方向控制寄存器 GSDR 完成，也可直接调用 SDK 中 GPIO_SetPinDir()接口完成；
- 设置 GPIO 为高电平，可以通过设置图 2-1 中的 GSLR 寄存器完成，也可直接调用 SDK 中的 GPIO_WritePin()接口完成。

SPIN
TROL

4 配置 GPIO 输入

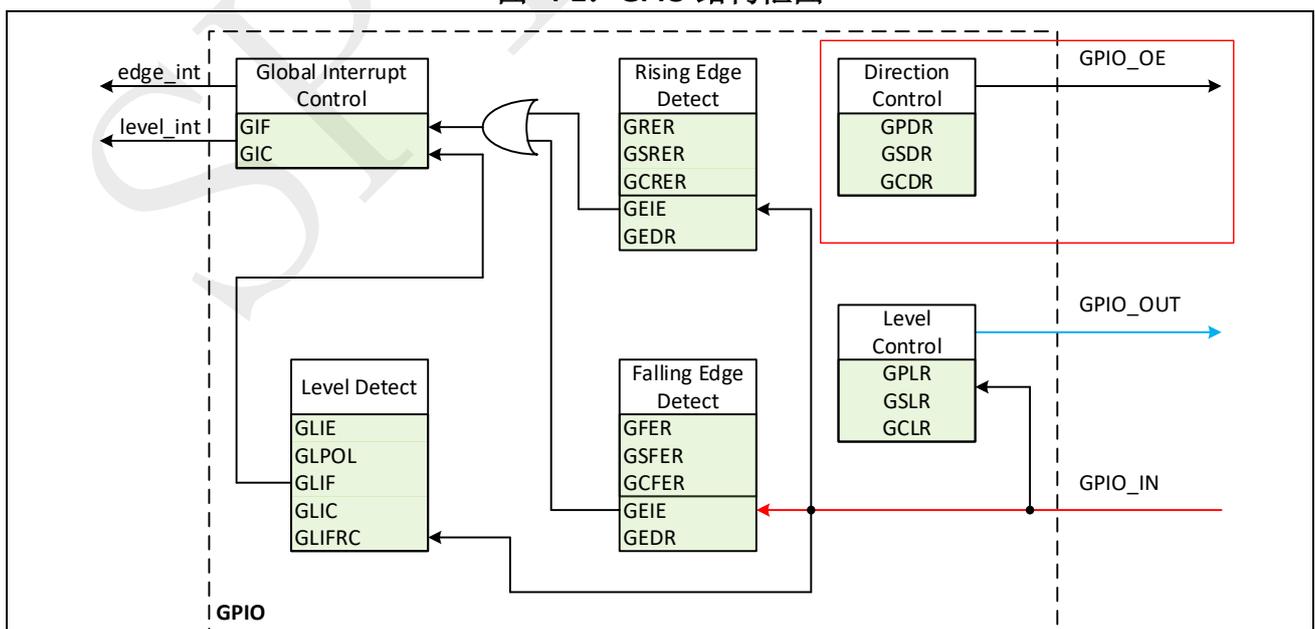
当需要配置 GPIO 作为输入功能的时候，需要对图 4-1 中的上下两路路径进行配置，下路用来关闭三态门，从而让总线上出现高阻态。上路用来将输入信号通过 PINMUX 传入 GPIO。

图 4-1: PINMUX 结构框图



此时 OE 控制信号来源于 GPIO_OE，如图 4-2 所示。

图 4-2: GPIO 结构框图



示例代码演示将 GPIO_37 作为输入，控制 ADC 进行采样。其中 ADC_SetExternalSOC() 用于关闭 PINMUX 中受 OE 控制的三态门，从而让总线上出现高阻态，以及使能输入信号通过 PINMUX 传入 GPIO。

Example Code

```
int main()
{
    FLASH_WALLOW();

    #if defined(SPC1158)
        FLASH_SetTiming(100000000);
        /* Disable flash write access after flash operation had done */
        FLASH_WDIS();

        CLOCK_InitWithRCO(CLOCK_HCLK_100MHZ);
    #else
        FLASH_SetTiming(200000000);
        /* Disable flash write access after flash operation had done */
        FLASH_WDIS();

        CLOCK_InitWithRCO(CLOCK_HCLK_200MHZ);
    #endif

    Delay_Init();

    /*
     * Init the UART
     *
     * 1.Set the GPIO34/35 as UART FUNC
     *
     * 2.Enable the UART CLK
     *
     * 3.Set the rest para
     */
    GPIO_SetPinChannel(GPIO_34, GPIO34_UART_TXD);
    GPIO_SetPinChannel(GPIO_35, GPIO35_UART_RXD);
    CLOCK_EnableModule(UART_MODULE);
    UART_Init(UART, 38400);

    /*ADC Init*/
    GPIO_SetPinChannel(GPIO_8, GPIO8_ADC8);
    ADC_Init(ADC_SOC_0, ADC_SHC_P_ANA_IN8, ADC_SHC_N_GND, SHC, ADCTRIG_EXTSOC);

    ADC_SetExternalSOC(GPIO_37, GPIO_LEVEL_LOW);

    NVIC_EnableIRQ(ADC0_IRQn);

    while (1)
    {

    }
}

void ADC0_IRQHandler(void)
{
    i32VSP = ADC_GetResult(ADC_SOC_0);
    printf("ADC SOC0 Voltage = %2fv(data is %d)\n",
    (double)ValueToVoltage(i32VSP), i32VSP);

    /* Clear SOC0 INT flag */
    ADC_ClearInt(ADC_SOC_0);
}
```