



# ***Licenciatura em Telecomunicações e Informática***

## ***Arquitetura e Tecnologia de Computadores interface ao exterior***

**Paulo Cardoso**

Grupo de Sistemas Embebidos  
Departamento de Eletrónica Industrial  
Escola de Engenharia  
Universidade do Minho





# *Sumário*

- GPIO
- Configuração
- Operação
- Características elétricas



- Microcontroladores
  - Resposta a uma necessidade da indústria
    - Sistemas computacionais para ambientes de controlo
      - Onde para além de um “computador” é necessário
        - I/O digital e analógico
        - Temporizadores
        - PWM (Pulse Width Modulation)
        - Comunicação série (vários protocolos)
    - Por outro lado os ambientes industriais são agressivos
      - Condições ambientais, ruído eletromagnético
        - Fazer um computador com processador, memória e os circuitos integrados (CI) para implementar as funcionalidades acima
          - Solução mais sujeita a falhas e mais cara



- Microcontroladores
  - Resposta a uma necessidade da indústria
    - A solução é um CI único
      - Onde processador e periféricos estão integrados
        - Não apenas empacotados em conjunto no mesmo CI mas arquiteturalmente integrados
          - I.e. o processador “conhece” os periféricos
  - Assim, quando se fala de um microcontrolador
    - Fala-se de um sistema que integra
      - Processador, memória
      - Pinos de I/O digital e analógico (com DACs/ADCs)
      - *Timers*, PWM
      - Comunicação série básica e uma variedade de protocolos tais como I2C, SPI, CAN



# *O microcontrolador 8051*

## *o 8051*

- O 8051
  - Microcontrolador de 8 bits lançado em 1980
    - O QUÊ???????
    - Hoje em dia o mundo é dominado por microcontroladores ARM de 16 e 32 bits
  - Ainda muito usado atualmente
    - A patente expirou
    - Vários fabricantes introduziram novas versões do 8051
      - Mantendo a compatibilidade original
      - Usando as últimas técnicas de desenho de processadores
        - Aumentando significativamente o desempenho
        - E.g.: O Intel 8051 original executava a 12MHz e usava 12 ciclos para executar 1 instrução (1MHz, 1MIP- milhões de instruções por segundo)
        - O nosso micro executa a 100MHz a 100 MIPS (100x mais rápido que o 8051 clássico uma frequência apenas cerca de 8x mais)
      - Introduzindo novas características



# *O microcontrolador 8051*

## *o 8051*

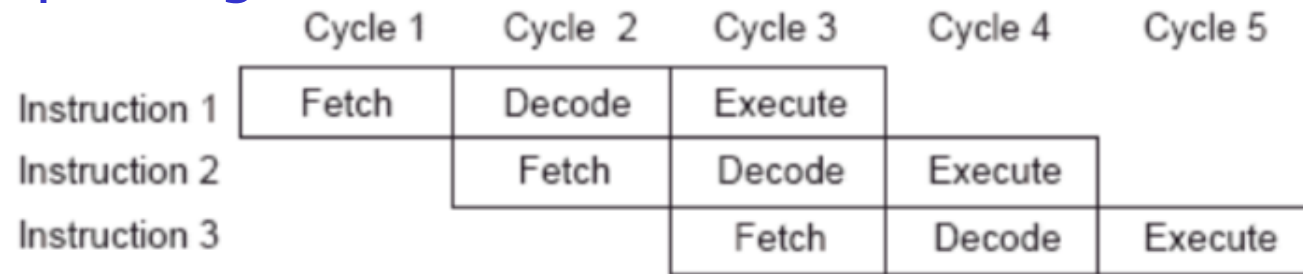
- O 8051
  - Ainda muito usado atualmente
    - Uma base instalada de milhões de linhas de código
    - Para além disso tem algumas vantagens arquiteturais
      - Implementa a capacidade de realizar operações lógicas diretamente em alguns registos e locais de memória
        - Este é um recurso muito útil em quase todas as aplicações de controlo, sejam elas industriais ou outras
          - Essas operações podem ser realizadas num único ciclo, permitindo um melhor tempo de resposta de controlo
    - Os portos de I/O são acedidos à velocidade do processador
      - Noutras arquiteturas estão separados por um barramento com desempenho inferior



# *O microcontrolador 8051 o 8051*

- Os 8051 atuais
  - Algumas novidades de implementação

- Pipelining



- MAC (multiply–accumulate)
    - Operações usadas em processamento digital de sinal
  - Melhor sistema de interrupções
    - Tempo de resposta (latência) melhorado
    - Mecanismos mais sofisticados de prioridades
  - *Debug* por hardware (*in-system debugging*)



# *O microcontrolador 8051 o 8051*

- Os 8051 atuais
  - Vantagens do ponto de vista comercial
    - Consome menos que os micros mais poderosos
    - Custo mais baixo
    - Atrativos para soluções menos exigentes
      - E para soluções *low cost*
    - Apelativo para muitas soluções IoT
      - Onde o *sensing* é mais importante que o *processing*





# *O microcontrolador 8051 o 8051*

- Os 8051 atuais
  - Vantagens de programação
    - Debug por hardware
      - Tal como os os micros mais poderosos
    - Mais fácil de programar
      - Dado ter complexidade mais baixa



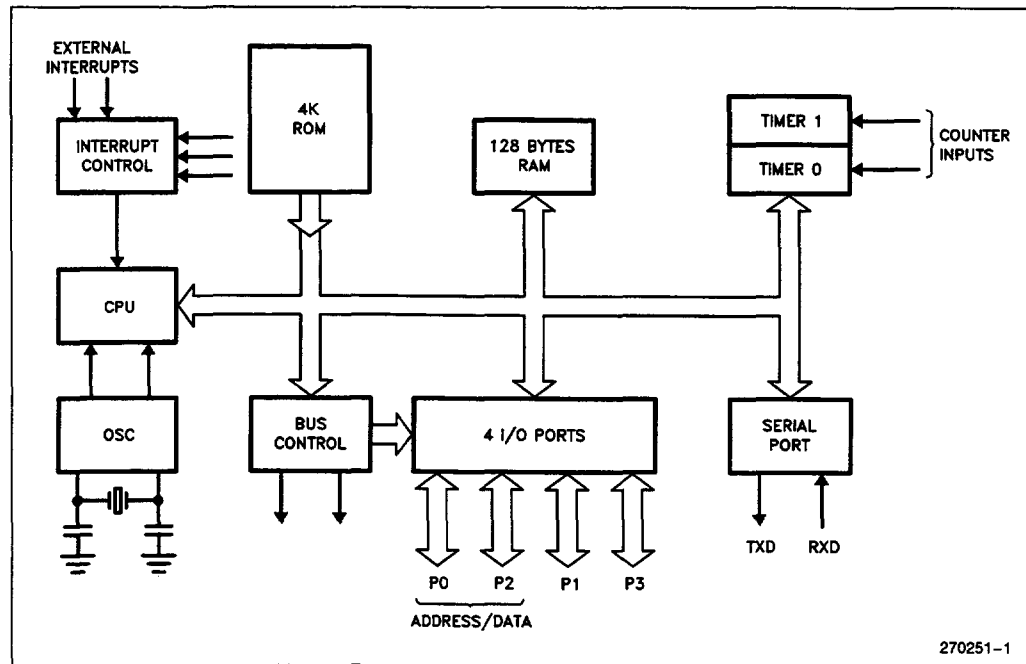
# *O microcontrolador 8051 o 8051*

- Os 8051 atuais
  - Vantagens de ensino
    - Micro de baixa complexidade
      - 44 mnemónicas *assembly*
        - Embora cada uma tenha declinações (diferentes opcodes)
        - 109 instruções
    - Implementa os conceitos relevantes
      - Classes de instruções
      - Modos de endereçamento
    - Periféricos
      - Baixa complexidade



# *O microcontrolador 8051* *arquitetura*

- Arquitetura do MCS 8051
  - O diagrama de blocos original



Fonte: <https://web.mit.edu/6.115/www/document/8051.pdf>

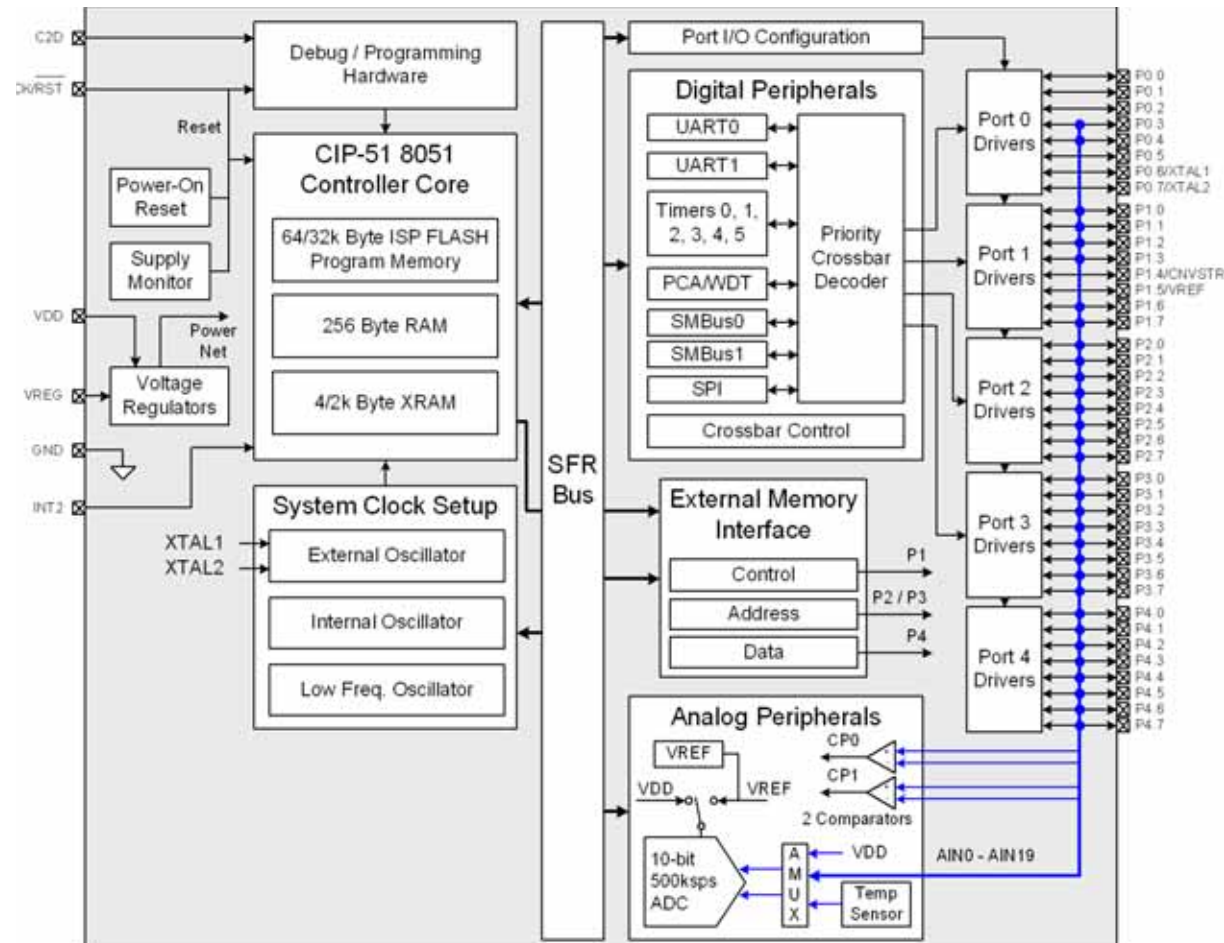


# *O microcontrolador 8051 arquitetura*



Embedded Systems  
Research Group

- Silicon Labs C8051F388
- Usado no nosso kit





# *O microcontrolador 8051*

## *Silicon Labs C8051F388*



## **C8051F388/9/A/B**

### **Flash MCU Family**

#### **Analog Peripherals**

- **10-Bit ADC**
  - Up to 500 kbps
  - Built-in analog multiplexer with single-ended and differential mode
  - VREF from external pin, internal reference, or  $V_{DD}$
  - Built-in temperature sensor
  - External conversion start input option
- **Two comparators**
- **Internal voltage reference**
- **Brown-out detector and POR Circuitry**

#### **On-Chip Debug**

- On-chip debug circuitry facilitates full speed, non-intrusive in-system debug (No emulator required)
- Provides breakpoints, single stepping, inspect/modify memory and registers
- Superior performance to emulation systems using ICE-chips, target pods, and sockets

#### **Voltage Supply Input: 2.7 to 5.25 V**

- Voltages from 2.7 to 5.25 V supported using On-Chip Voltage Regulators

#### **High Speed 8051 $\mu$ C Core**

- Pipelined instruction architecture; executes 70% of instructions in 1 or 2 system clocks
- Up to 48 MIPS operation
- Expanded interrupt handler

#### **Memory**

- 4352 or 2304 Bytes RAM
- 64 or 32 kB Flash; In-system programmable in 512-byte sectors

#### **Digital Peripherals**

- 40/25 Port I/O; All 5 V tolerant with high sink current
- Hardware enhanced SPI™, two I²C/SMBus™, and two enhanced UART serial ports
- Six general purpose 16-bit counter/timers
- 16-bit programmable counter array (PCA) with five capture/compare modules
- External Memory Interface (EMIF)

#### **Clock Sources**

- Internal Oscillator:  $\pm 1.5\%$  accuracy. Supports all UART modes
- External Oscillator: Crystal, RC, C, or clock (1 or 2 Pin modes)
- Low Frequency (80 kHz) Internal Oscillator
- Can switch between clock sources on-the-fly

#### **Packages**

- 48-pin TQFP (C8051F388/A)
- 32-pin LQFP (C8051F389/B)
- 5x5 mm 32-pin QFN (C8051F389/B)

**Temperature Range: -40 to +85 °C**



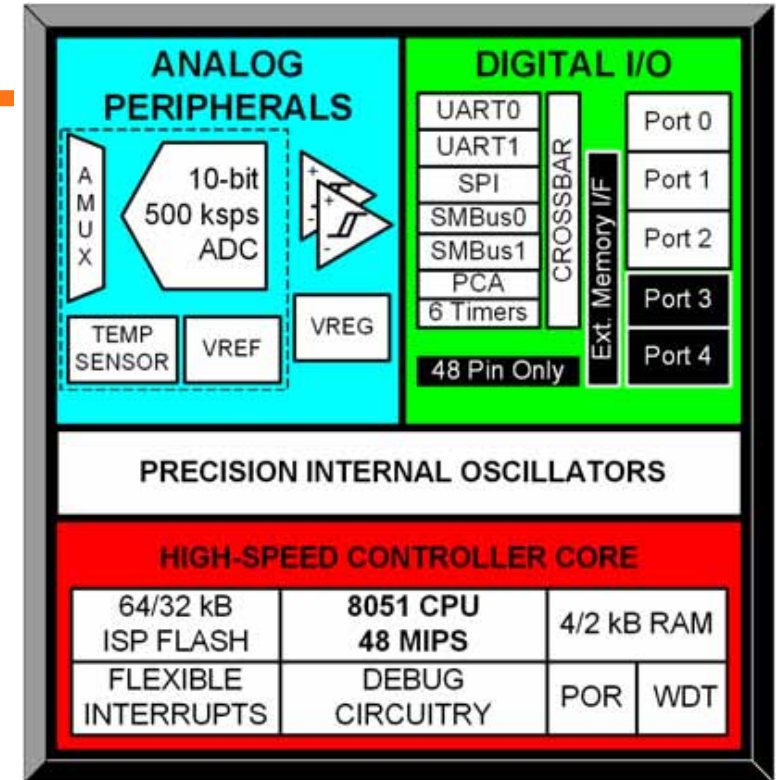
# *O microcontrolador 8051*

## *Silicon Labs C8051F388*

- Silicon Labs C8051F388

- **Blocos**

- Periféricos analógicos
- Periféricos digitais
- Relógio do sistema
- *Core do 8051*



- **Periféricos**

- **Módulos de hardware especializados**

- Exemplo: temporizadores, porta série, ADCs, DACs, ...
  - Estes recursos são disponibilizados através de pinos de I/O
  - Estes pinos de I/O podem ser também usado para I/O genérico (GPIO)
  - Cada pino ser configurado para uma função diferente dependendo da aplicação



# *O interface ao exterior*

## *GPIO*

- Interface ao exterior
  - O GPIO (General purpose I/O)
    - Interface mais simples que o micro suporta
      - Divididos em conjuntos de 8 pinos de I/O (portos)
        - O micro original disponibiliza 4 portos de I/O
          - O nosso micro suporta 5 portos
        - Cada um dos pinos de cada porto pode ser configurado como
          - GPIO digital
          - Entrada analógica;



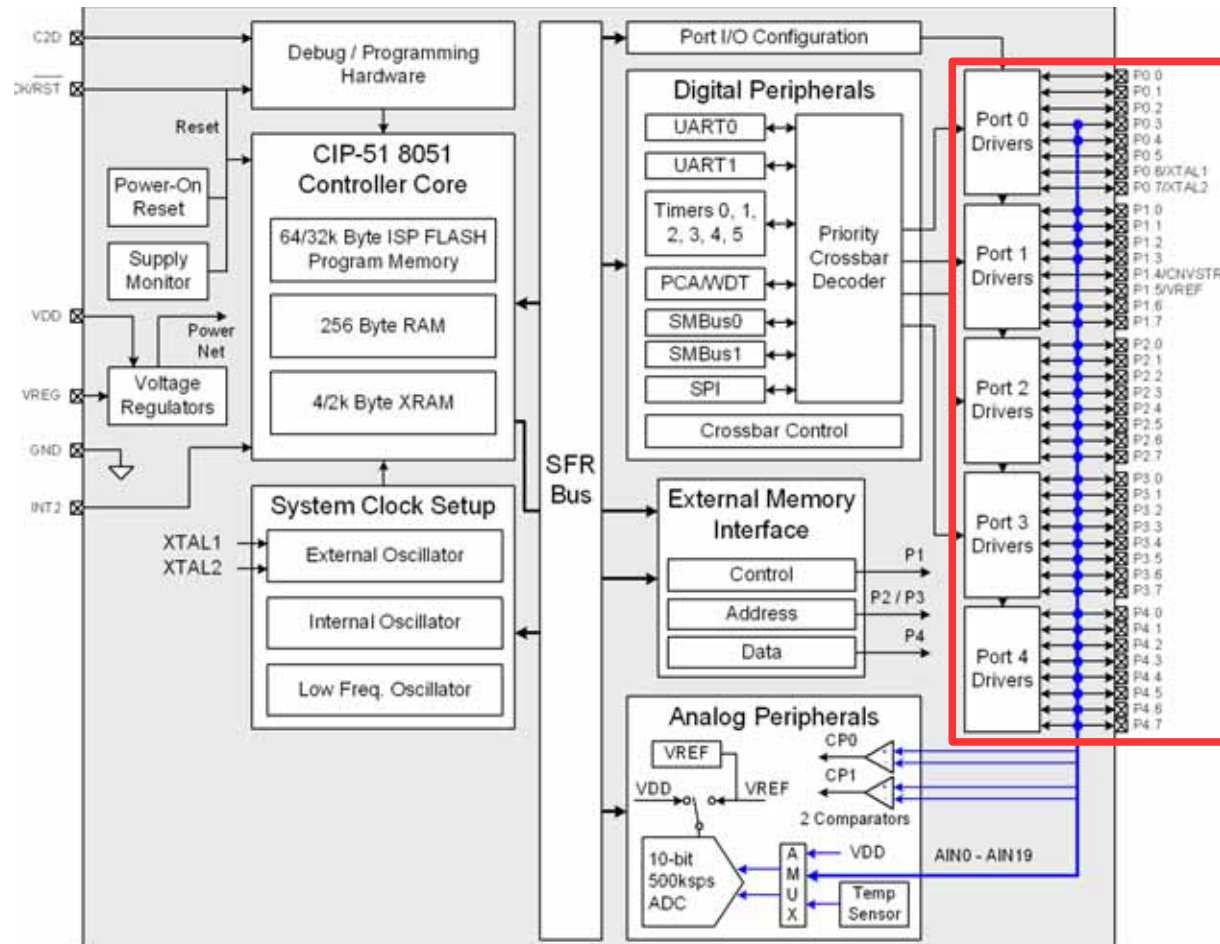
# *O interface ao exterior*

## *GPIO*



Embedded Systems  
Research Group

- O GPIO do Silicon Labs C8051F388







- O GPIO do Silicon Labs C8051F388
  - Permite a associar de forma flexível um pino a uma função
    - Priority Crossbar Decoder
      - Funções
        - UART 0/1
        - Timer 0/1
        - SPI 0/1
        - I2C
        - ...
      - As funções não estão fixas aos pinos
      - Após *reset* a *crossbar* está desativada



# *O interface ao exterior*

## *GPIO*

- Caso se pretenda usar os pinos
  - Numa configuração base apenas como GPIO
    - Basta ativar a crossobar
      - **XBARE** = 1 (registo **XBR1** |= 0x40)
    - Após *reset* todos os pinos ficam como entradas digitais
      - Com weak pull-up



# *O interface ao exterior*

## *GPIO*

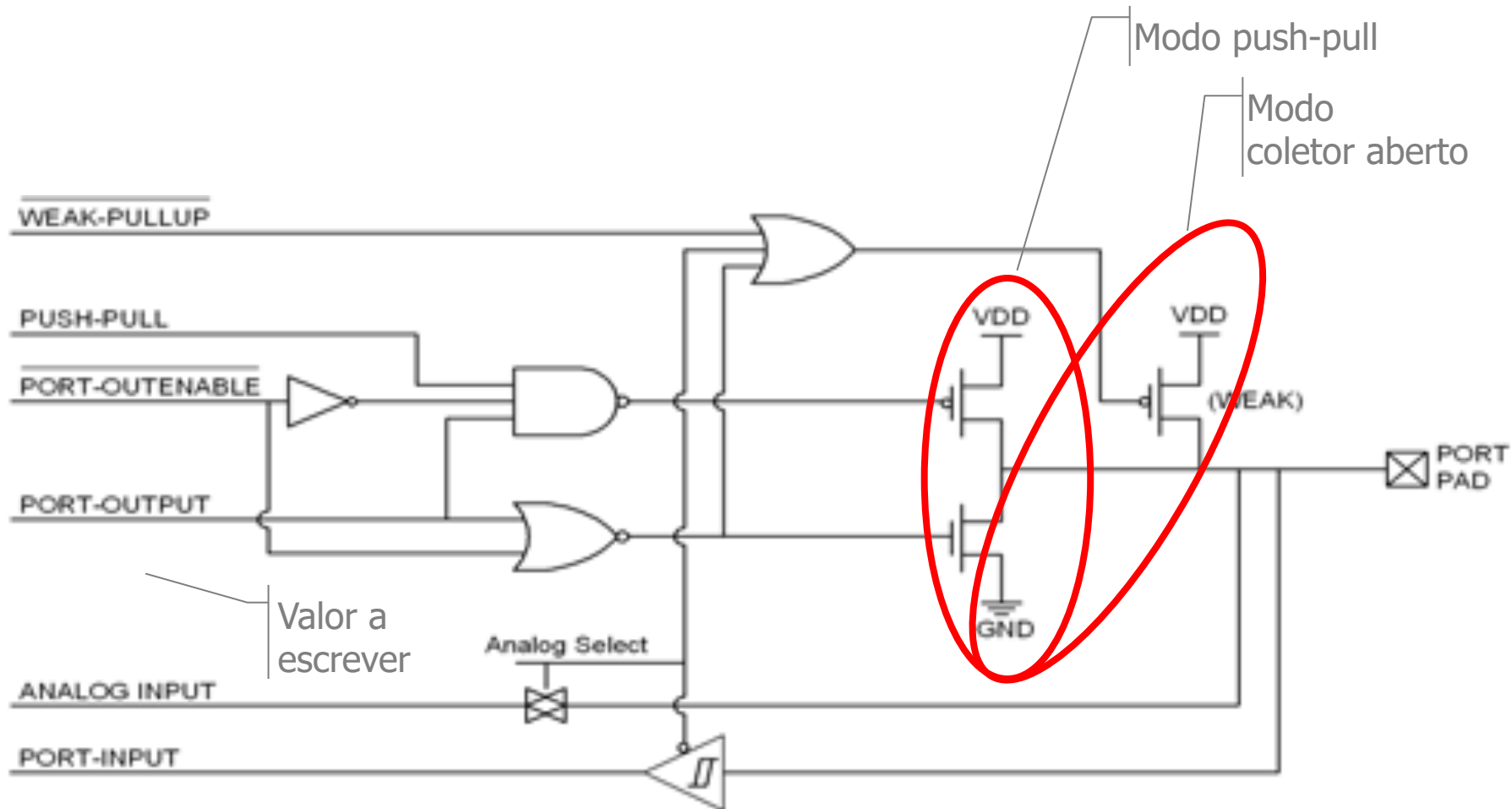
- Funcionamento dos pinos do porto
  - Como entrada
    - Digital ou analógico
      - Modo coletor aberto
        - Com ou sem weak pull-up
  - Como saída
    - Modo coletor aberto ou modo push-pull



# *O interface ao exterior*

## *GPIO*

- Um pino do porto





# *O interface ao exterior*

## *GPIO*

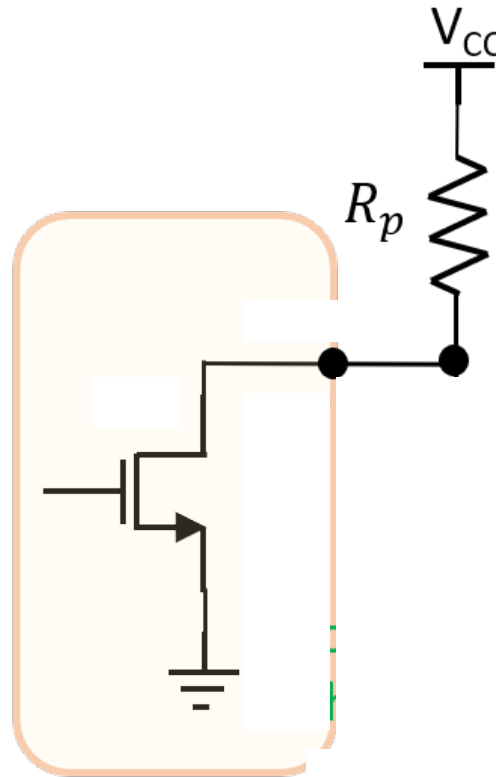
- Modo coletor aberto (open drain)
  - Usado nos pinos de *input* e *output*
  - Neste modo o pino só pode “consumir corrente”
    - O pino pode estar em
      - alta impedância
        - O pino está “desligado”
          - Para evitar que o pino fique a “flutuar” pode ser usado um “weak pull-up”
            - Estes estão ativos quando configurados, o pino está em modo digital, definido como “open-drain” e é escrito 1 no porto
            - São usados tipicamente quando o pino está como entrada
      - Ligado à massa
        - Indo a zero (consumindo corrente)
  - É comum ser usada uma resistência externa de *pull-up*



# *O interface ao exterior*

## *GPIO*

- Modo coletor aberto
  - Resistência de pull-up

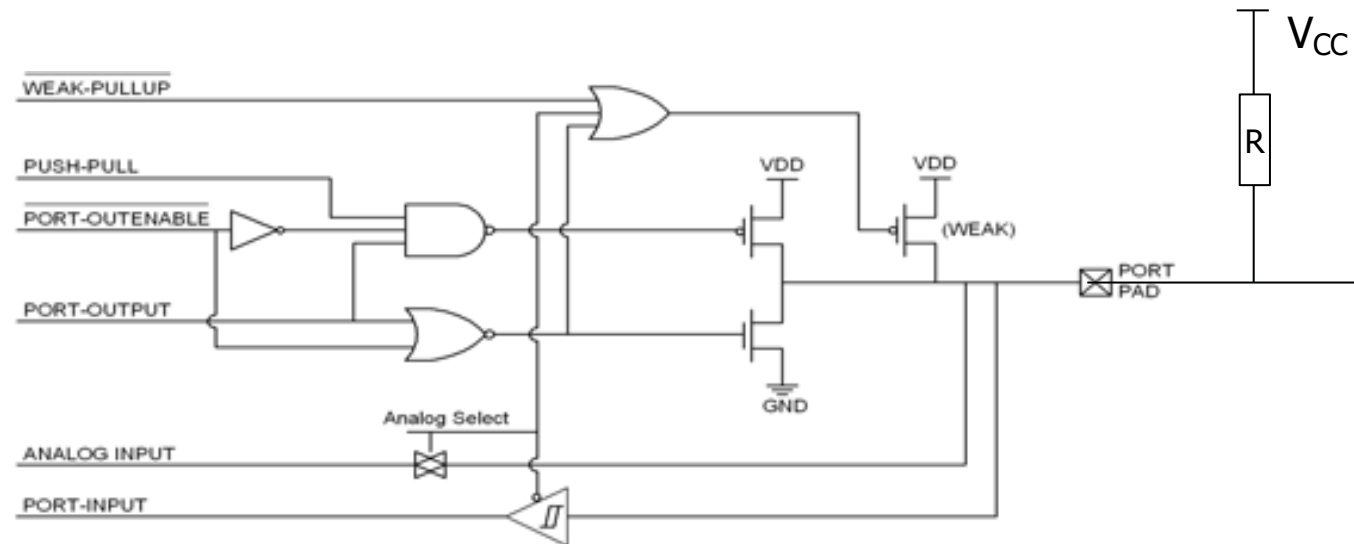




# *O interface ao exterior*

## *GPIO*

- Modo coletor aberto
  - Configuração típica com resistência de "pull-up"

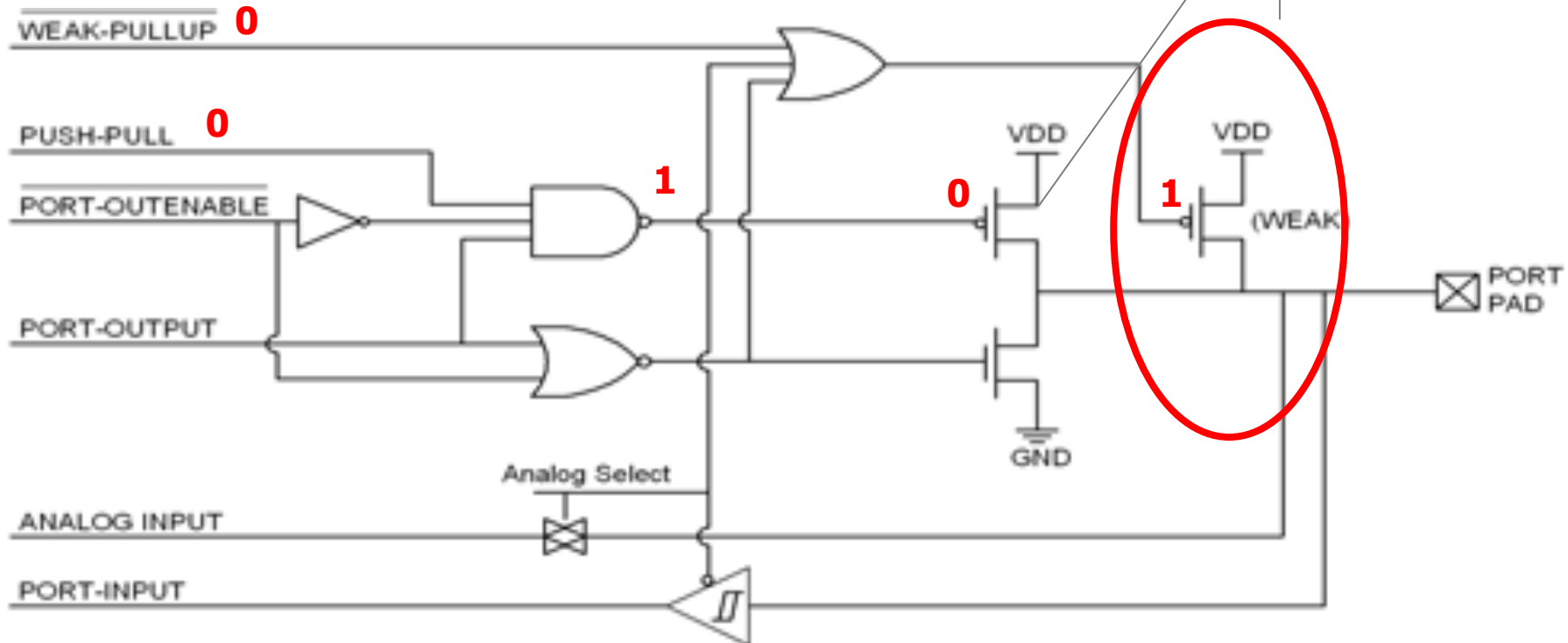




# O interface ao exterior

## GPIO

- Modo coletor aberto
  - Weak pull-up ativo
    - Usado nos pinos de **output** e **input**
      - Mais usado em **input**



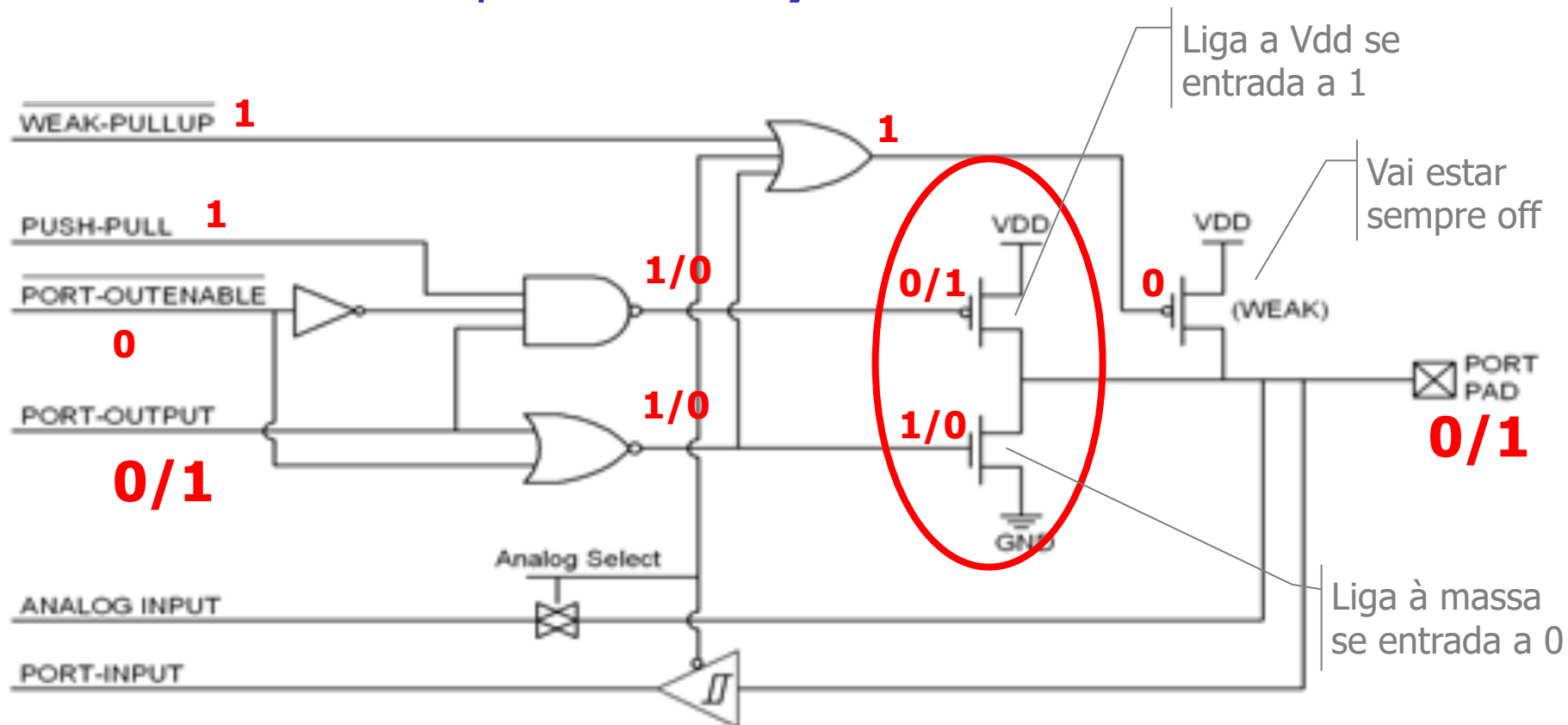




# O interface ao exterior

## GPIO

- Modo push-pull
  - Neste modo o pino fornece o "1" e o "0"
  - Usado nos pinos de **output**





# *O interface ao exterior*

## *GPIO*

- Funcionamento dos pinos do porto
  - Como entrada
    - Digital ou analógico
      - Modo coletor aberto
        - Com ou sem weak pull-up
  - Como saída
    - Modo coletor aberto ou modo push-pull



# *O interface ao exterior*

## *GPIO*

- A utilização dos GPIO  
(ou qualquer outro periférico)
  - Está dividida em duas partes
    - Configuração
      - Geralmente os periféricos têm vários modos de operação
        - Parametrizados através de registos de configuração (SFRs)
    - Operação
      - Depois de configurado o periférico está pronto a ser usado
        - Os dados sobre os quais os periféricos operam estão também em registos (SFRs)
        - Outros registos ou *flags* (bits) poderão ser alterados em função do estado dos dados (SFRs)



# *O interface ao exterior configuração*

- Configuração
  - Como entrada
    - Digital
      - Registo **PxMDIN**
        - Bit respetivo a 1 (valor de defeito)
      - Weak pull-up
        - Registo **XBR1**, bit **WEAKPUD** (valor de defeito)
      - Registo **PxMDOUT**
        - Bit respetivo a 0 (valor de defeito)
          - Um pino de entrada só pode estar em “open drain”
      - Escrita de 1 para o pino do porto
    - Após *reset* todos os pinos ficam como entradas digitais
      - Com weak pull-up; Os pinos do porto estão a 1



# *O interface ao exterior configuração*

- Configuração
  - Como entrada
    - Analógica
      - Registo **PxMDIN**
        - Respetivo bit a 0
      - Registo **PxSKIP**
        - Respetivo bit a 1
      - Weak pull-up
        - Modo analógico: desativado



# *O interface ao exterior configuração*

- Configuração - como entrada
  - Digital ou analógico
    - Registo **P<sub>x</sub>MDIN**

**SFR Definition 20.5. P0MDIN: Port 0 Input Mode**

Bit	7	6	5	4	3	2	1	0
Name	P0MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF1 SFR Page = All Pages

Bit	Name	Function
7:0	P0MDIN[7:0]	<b>Analog Configuration Bits for P0.7–P0.0 (respectively).</b> Port pins configured for analog mode have their weak pullup, digital driver, and digital receiver disabled. 0: Corresponding P0.n pin is configured for analog mode 1: Corresponding P0.n pin is not configured for analog mode



# *O interface ao exterior configuração*

- Configuração - como entrada
  - Digital ou analógico
  - Registo **PxSKIP**

## SFR Definition 20.7. **P0SKIP: Port 0 Skip**

Bit	7	6	5	4	3	2	1	0
Name	P0SKIP[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD4 SFR Page = All Pages

Bit	Name	Function
7:0	P0SKIP[7:0]	<b>Port 0 Crossbar Skip Enable Bits.</b> These bits select Port 0 pins to be skipped by the Crossbar Decoder. Port pins used for analog, special functions or GPIO should be skipped by the Crossbar. 0: Corresponding P0.n pin is not skipped by the Crossbar. 1: Corresponding P0.n pin is skipped by the Crossbar.



# *O interface ao exterior configuração*

- Configuração - como entrada
  - **Weak pull-up**
    - Registo **XBR1**, bit **WEAKPUD**

**SFR Definition 20.2. XBR1: Port I/O Crossbar Register 1**

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	T1E	T0E	ECIE	PCA0ME[2:0]		
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE2 SFR Page = All Pages

Bit	Name	Function
7	WEAKPUD	<b>Port I/O Weak Pullup Disable.</b> 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	<b>Crossbar Enable.</b> 0: Crossbar disabled. 1: Crossbar enabled.

- Afeta todos os pinos de todos os portos!







# *O interface ao exterior configuração*

- Configuração
  - Como saída
    - Registo **P<sub>x</sub>MDOUT**
      - Seleciona modo open drain ou push-pull
  - Weak pull-up
    - Registo **XBR1**, bit **WEAKPUD**
  - Após *reset* todos os pinos ficam como entradas digitais
    - Com weak pull-up



- **Registo P<sub>x</sub>MDOUT**

**SFR Definition 20.6. P0MDOUT: Port 0 Output Mode**

Bit	7	6	5	4	3	2	1	0
Name	P0MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

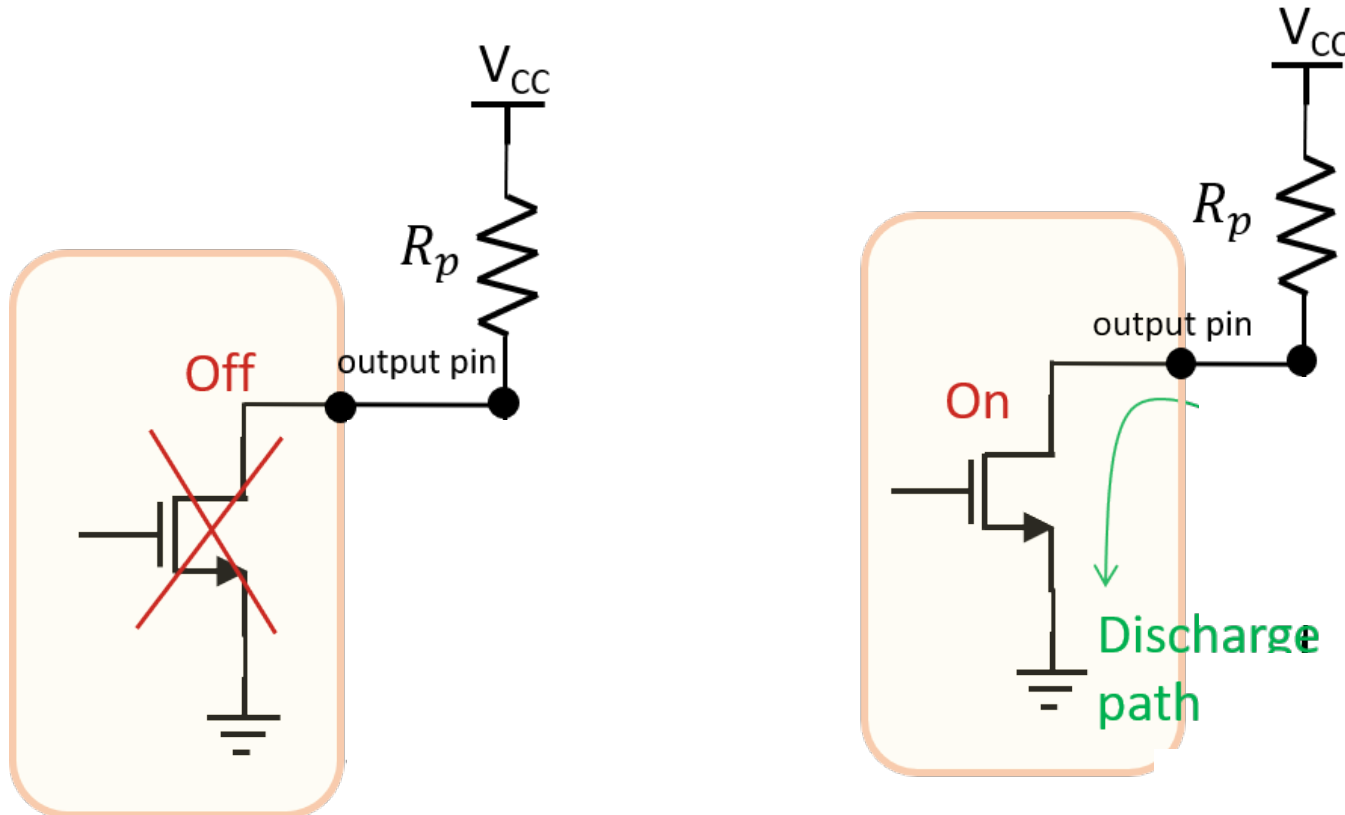
SFR Address = 0xA4 SFR Page = All Pages

Bit	Name	Function
7:0	P0MDOUT[7:0]	<b>Output Configuration Bits for P0.7–P0.0 (respectively).</b> These bits are ignored if the corresponding bit in register P0MDIN is logic 0. <b>0: Corresponding P0.n Output is open-drain</b> 1: Corresponding P0.n Output is push-pull.



# *O interface ao exterior configuração*

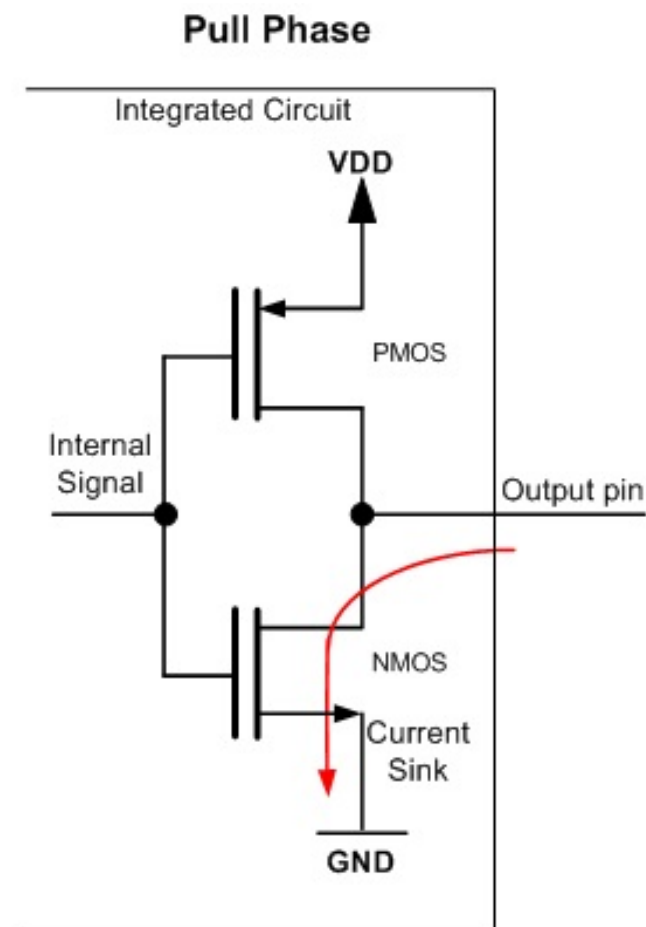
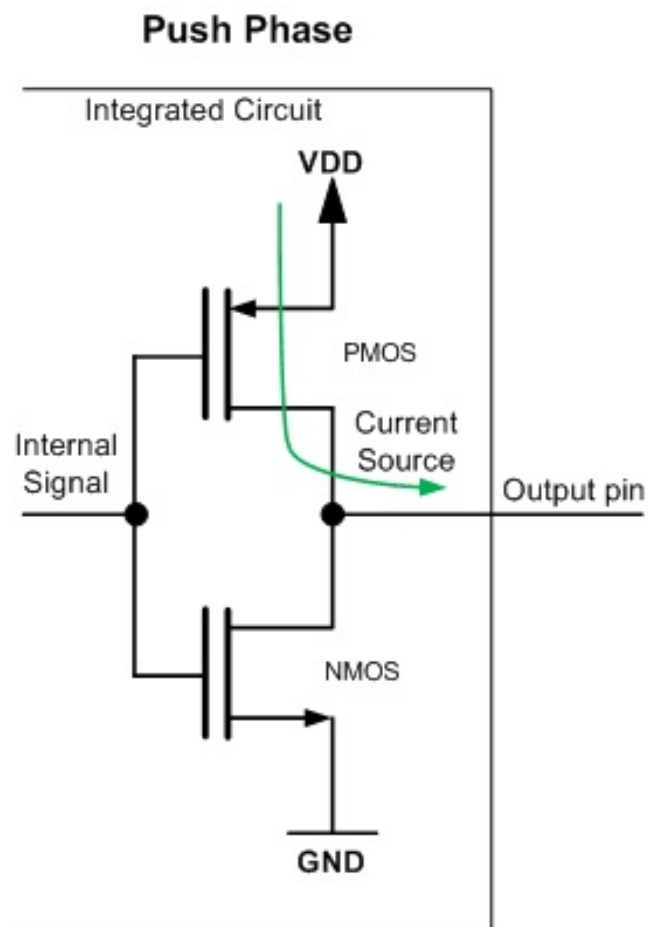
- Modo coletor aberto
  - Resistência de pull-up





# *O interface ao exterior configuração*

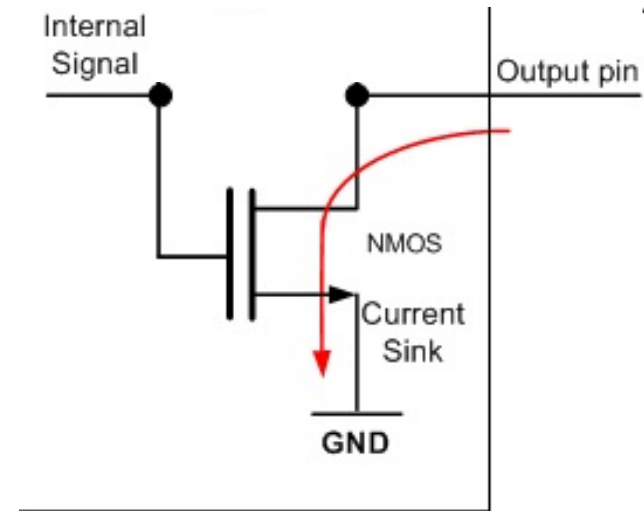
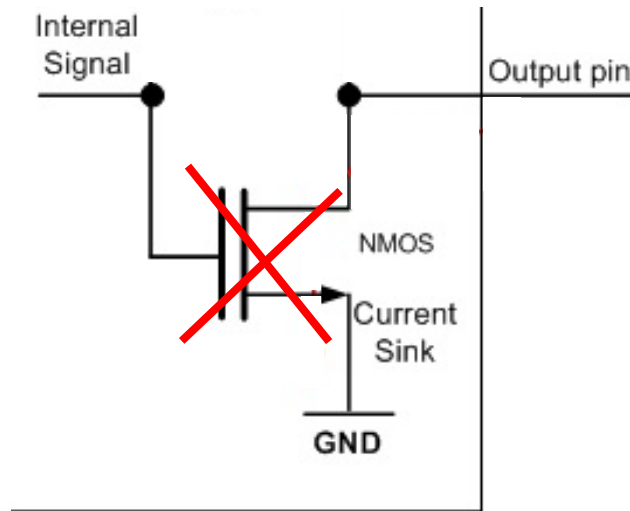
- Modo push-pull





## *O interface ao exterior configuração*

- Modo coletor aberto (open drain)





# *O interface ao exterior operação*

- Operação
  - Depois dos pinos estarem configurados
    - A leitura/escrita é simplesmente aceder ao pino do porto
      - Ao escrever em cada pino
        - O valor é guardado no registo
          - Para manter o valor dos dados de saída em cada pino
      - Ao ler de cada pino
        - Os níveis lógicos dos pinos de entrada da porta são retornados
          - A exceção são as instruções “read-modify-write”
            - ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ e MOV, CLR ou SETB (quando o destino é um bit individual num porto SFR)
            - Para essas instruções, o valor do registo (não o pino) é lido, modificado e escrito de volta no SFR



# *O interface ao exterior operação*

- O registo do porto P0
  - SFR 0x80

SFR Definition 20.4. P0: Port 0

Bit	7	6	5	4	3	2	1	0
Name	P0[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0x80 SFR Page = All Pages; Bit Addressable

Bit	Name	Description	Write	Read
7:0	P0[7:0]	<b>Port 0 Data.</b> Sets the Port latch logic value or reads the Port pin logic state in Port cells configured for digital I/O.	0: Set output latch to logic LOW. 1: Set output latch to logic HIGH.	0: P0.n Port pin is logic LOW. 1: P0.n Port pin is logic HIGH.





## *O interface ao exterior operação*

- Cada porto é suportado por um registo
  - Que armazena o estado de cada pino do porto
    - De onde podemos ler e escrever
  - Mapeado em memória na área do SFR
    - P0:0x80
    - P1:0x90
    - P2:0x0A0
    - P3:0x0B0
    - P4:0x0C7



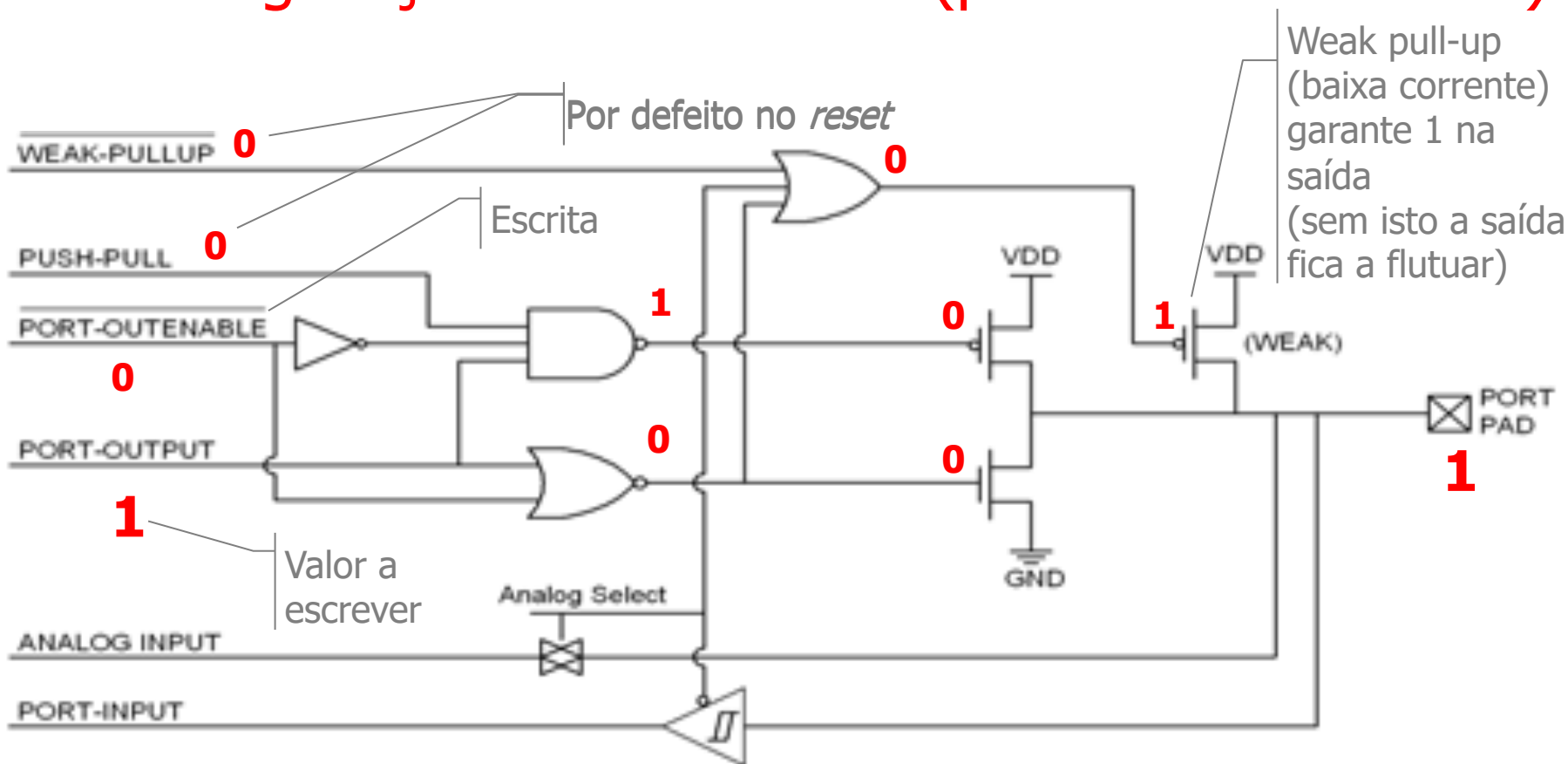
## *O interface ao exterior operação*

- Cada porto é suportado por um registo
  - **Acesso aos pinos via software**
    - Exemplo: escrita num porto (neste caso porto P2)
      - `P2 = 0xC0;`
    - Exemplo: condição com pino do porto (neste caso pino 6 do porto P0)
      - `if (!P0_6)`



# *O interface ao exterior operação*

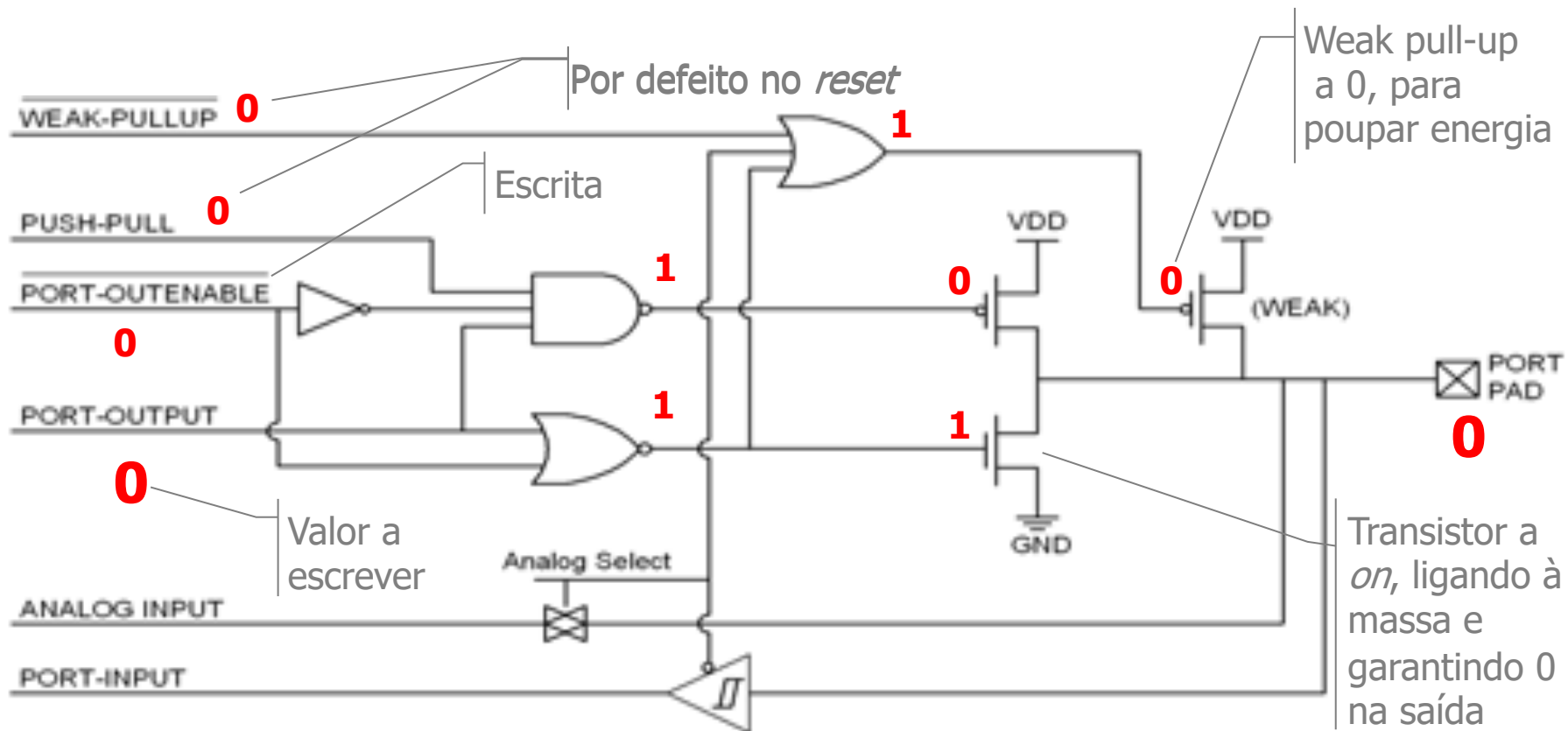
- Exemplo: Escrita de 1 num pino do porto
  - Configuração coletor aberto (por defeito no *reset*)





# *O interface ao exterior operação*

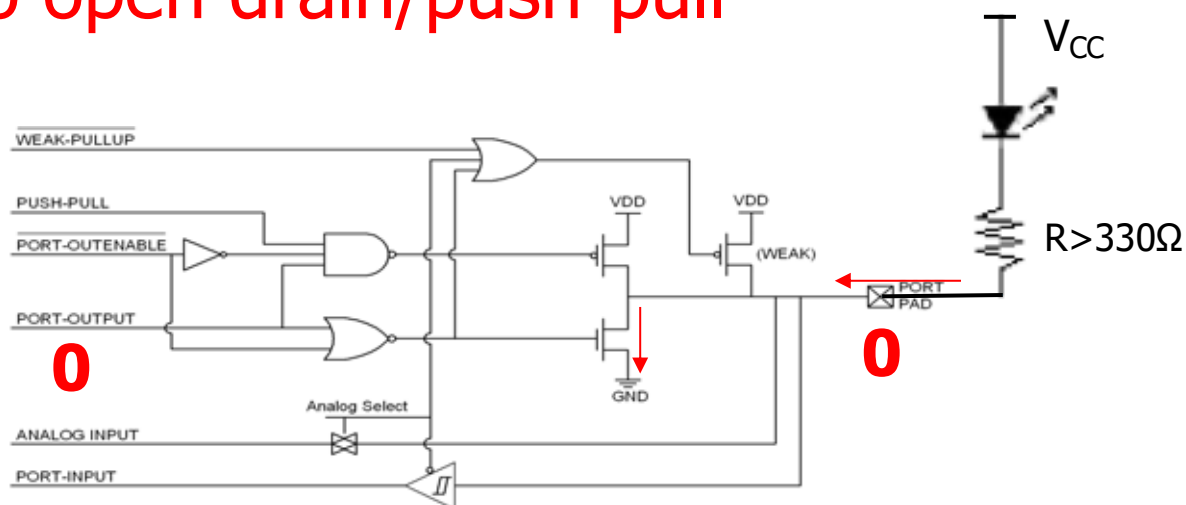
- Exemplo: Escrita de 0 num pino do porto
  - Configuração coletor aberto (por defeito no *reset*)





# *O interface ao exterior operação*

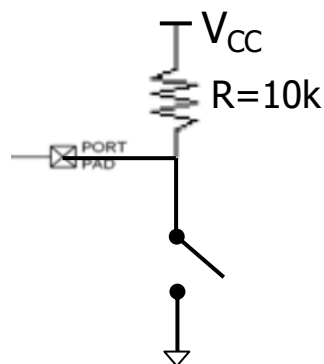
- Ligação de um led (output)
  - Modo open drain/push-pull



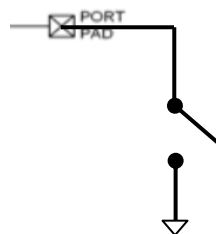


## *O interface ao exterior operação*

- Ligação de um botão de pressão (input)
  - Modo open drain/push-pull



- Modo open drain
  - Com weak pull-up





# *O interface ao exterior*

## *Exemplo*

- Questões
  - Verificar tipo de dados de **bitVar**
  - Pinos **P0\_6**, **P0\_7** (pinos de entrada)
    - Verificar no esquemático como estão ligados
  - Pino **P0\_7** (pino de saída)
    - Verificar no esquemático como estão ligados
  - O que faz o programa?

```
#include <c8051f380.h>

void main (void) {

    __bit bitVar;

    /* Faz Disable do watchdog */
    PCA0MD = 0x00;

    /* Ativa o Crossbar */
    XBR1 = 0x40;

    while (1) {
        bitVar = P0_6;
        bitVar &= P0_7;

        P2_7 = bitVar;
    }
}
```



# *O interface ao exterior*

## *Caraterísticas elétricas*

- Caraterísticas elétricas do portos

Parameter	Test Condition	Min	Typ	Max	Unit
Output High Voltage	$I_{OH} = -3 \text{ mA}$ , Port I/O push-pull	$V_{DD} - 0.7$	—	—	V
	$I_{OH} = -10 \text{ }\mu\text{A}$ , Port I/O push-pull	$V_{DD} - 0.1$	—	—	
	$I_{OH} = -10 \text{ mA}$ , Port I/O push-pull	—	$V_{DD} - 0.8$	—	
Output Low Voltage	$I_{OL} = 8.5 \text{ mA}$	—	—	0.6	V
	$I_{OL} = 10 \text{ }\mu\text{A}$	—	—	0.1	
	$I_{OL} = 25 \text{ mA}$	—	1.0	—	
Input High Voltage		2.0	—	—	V
Input Low Voltage		—	—	0.8	V
Input Leakage Current	Weak Pullup Off	—	—	$\pm 1$	$\mu\text{A}$
	Weak Pullup On, $V_{IN} = 0 \text{ V}$	—	15	50	
INT2 Detection Input Low Voltage		—	—	1.0	V
INT2 Detection Input High Voltage		3.0	—	—	V

$V_{DD} = 2,7 \text{ a } 3,6\text{V}$





# *O interface ao exterior*



Embedded Systems  
Research Group



Obrigado