

# Arquitectura de Computadores 2

## Aula 3 Configuração de Portos de I/O

Pedro Miguel Lavrador

## Objectivos

- Conhecer a estrutura básica e o modo de configuração de um porto de I/O no microcontrolador PIC32.
- Configurar em *assembly* os portos de I/O do PIC32 e aceder para enviar / receber informação do exterior.

## Descrição Breve

---

- Vários portos de I/O
  - Cada um dos bits (de cada porto) pode ser configurado como entrada ou saída.
  - Podemos considerar 1 porto de N bits ou N portos de 1 bit

PML - AC2 - 2014

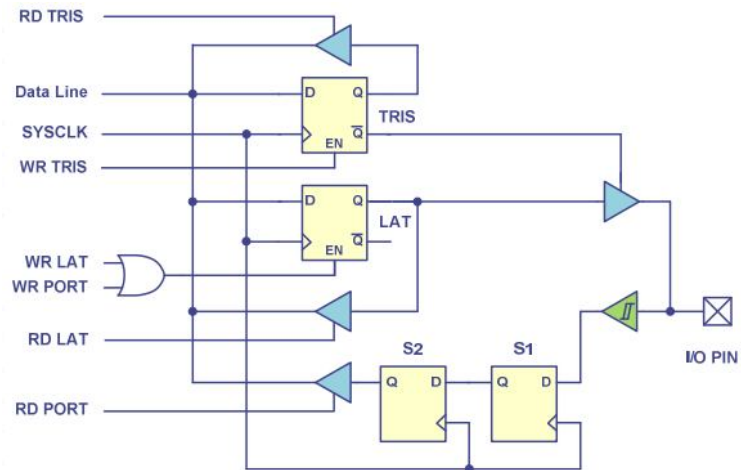
## Modelo de Programação

---

- O Registo TRISx
  - Para cada bit define se é IN (1) ou OUT (0)
- Os Registos PORTx e LATx
  - São acedidos para leitura ou escrita de dados.
  - (já vamos ver porque são dois registos diferentes)

PML - AC2 - 2014

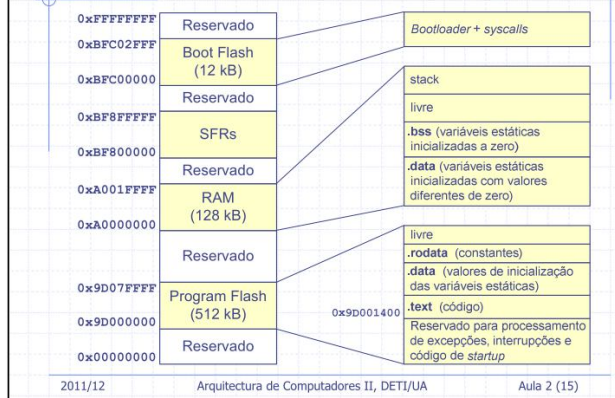
## Esquema simplificado de um porto de I/O



PML - AC2 - 2014

## O espaço de memória: SFR's

Mapa de memória do PIC32 (perspectiva do programador)



2011/12

Arquitectura de Computadores II, DETI/UA

Aula 2 (15)

PML - AC2 - 2014

## O Datasheet

TABLE

Virtual Address (BF88_#)
--------------------------

6100	TR
6110	PI
6120	L
6130	O

Legend:  
Note 1:

Virtual Address (BF88_#)	Register Name
6100	TRISE
6110	PORTE

OR PIC32MX534F064H, PIC32MX564F064H, PIC32MX564F128H, PIC32MX575F256H, MX664F064H, PIC32MX664F128H, PIC32MX675F256H, PIC32MX675F512H, PIC32MX695F512H MX775F512H AND PIC32MX795F512H DEVICES<sup>(1)</sup>

Bits												
11/2	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0
---	---	---	---	---	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0
---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0
---	---	---	---	---	---	---	---	---	---	---	---	---
---	---	---	---	---	ODCE7	ODCE6	ODCE5	ODCE4	ODCE3	ODCE2	ODCE1	ODCE0

Read as '0'. Reset values are shown in hexadecimal.

TRISE and INVD registers at their virtual addresses, plus offsets of 0x4, 0x8 and 0xC, respectively. See Section 12.1.1 "CLR, SET and INV Registers" for more

PML - AC2 - 2014

## Exemplo de Configuração PortE

- Definição de Constantes:

```
.equ SFR_BASE_HI, 0xBF88      # 16 MSbits of SFR area
.equ TRISE, 0x6100           # TRISE address is 0xBF886100
.equ PORTE, 0x6110           # PORTE address is 0xBF886110
.equ LATE, 0x6120            # LATE address is 0xBF886120
```

- Bit0 e bit3 configurados como Saídas:

```
lui    $t1, SFR_BASE_HI      #
lw     $t2, TRISE($t1)        # Mem_addr = 0xBF880000 + 0x6100
andi   $t2, $t2, 0xFFFF6     # bit0 = bit3 = 0 (0 means OUTPUT)
sw     $t2, TRISE($t1)        # Write TRISE register
```

- RE0 e RE3 tomam o valor 1:

```
lui    $t1, SFR_BASE_HI      #
lw     $t2, LATE($t1)         # Read LATE register
ori    $t2, $t2, 9            # Set bit0 and bit3
sw     $t2, LATE($t1)         # Write LATE register
```

PML - AC2 - 2014

## Manipulação bitwise de um porto

```
lw $t0, PORTE($a0)      # RD PORT
ori $t0, 0x0001
sw $t0, PORTE($a0)      # (RE0 = 1)
...
lw $t0, PORTE($a0)      # RD PORT
andi $t0, 0xFFFE
sw $t0, PORTE($a0)      # (RE0 = 0)
lw $t1, PORTE($a0)      # RD PORT
```

**Qual o valor lido no bit 0 de \$t1?**

13-03-2014

PML - AC2 - 2013

9

## Sugestões para a elaboração do Guião

- Parte I
  - Entrada/Saída:
    - Começar por testar isoladamente a saída e a entrada, de modo a detectar e resolver possíveis erros.
- Parte II
  - Algoritmo do contador de Johnson
    - Deslocamento à esquerda 1 casa;
    - Entra (à direita) o bit mais significativo (anterior) negado
    - 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000, 0000
- Parte III
  - Recuperar o conhecimento da aula 2 para criar esperas de 1 segundo.

PML - AC2 - 2014