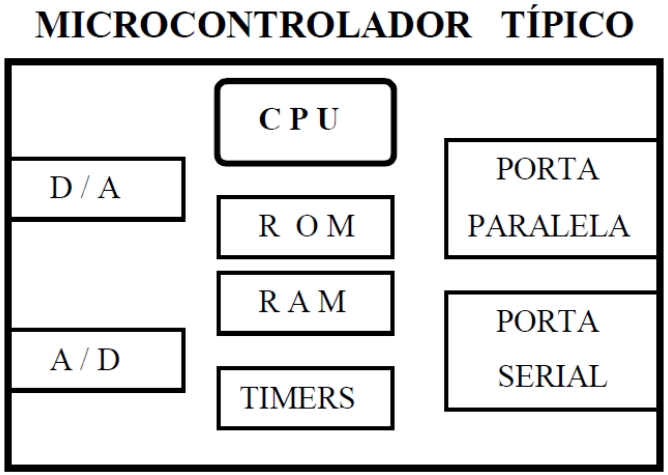
Interface entre Processador e Periféricos

Sub-sistema de entrada e saída – E/S (I/O)

Interface entre Processador e Periféricos

- Periféricos do processador
  - Circuitos na periferia do processador (CPU)



- A comunicação entre o processador e os periféricos é feita através de registradores
  - Os registradores podem estar dentro dos periféricos ou entre o processador e os periféricos
  - Podem ser de leitura, de escrita ou ambos
  - Esses registradores também são chamados de portas de E/S

Interface entre Processador e Periféricos

- Da mesma maneira que o processador acessa a memória fornecendo endereços, outros periféricos também tem endereços de acesso, os quais correspodendem aos registradores
- O endereçamento dos registradores pode seguir duas abordagens
  - Mapeamento em memória
  - Mapeamento em portas de E/S

Interface entre Processador e Periféricos

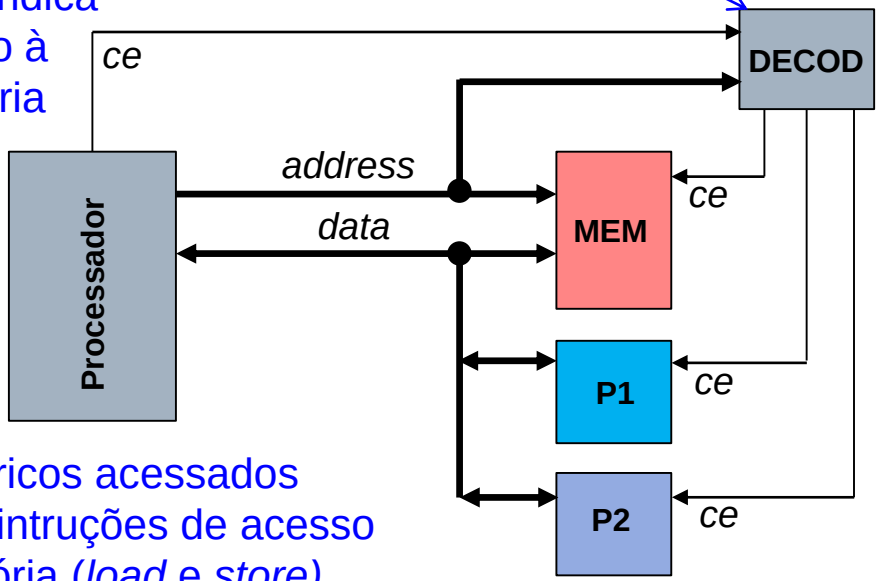
- Mapeamento em memória
  - Espaço de endereçamento único
    - Parte usado para acessar a memória
    - Parte usado para acessar periféricos
    - Divisão do espaço definida pelo projetista
    - Exemplo: espaço de endeçamento 0 – 255 (8 bits)
      - 0 – 249: memória (250 endereços de memória)
      - 250 – 255: periféricos (6 endereços de periféricos)
  - Periféricos acessados utilizando instruções de acesso à memória
    - Load/Store
  - Exemplos
    - MIPS
    - Processadores da Motorola

Interface entre Processador e Periféricos

- Mapeamento em memória
  - Exemplo: barramento de endereços de 8 bits
    - Espaço de endereçamento: 0 – 255
      - 0 – 253: memória
      - 254: P1
      - 255: P2

Habilita periféricos de acordo com o endereço gerado pelo processador

ce ativo indica acesso à memória



Periféricos acessados utilizando intruções de acesso à memória (*load* e *store)*

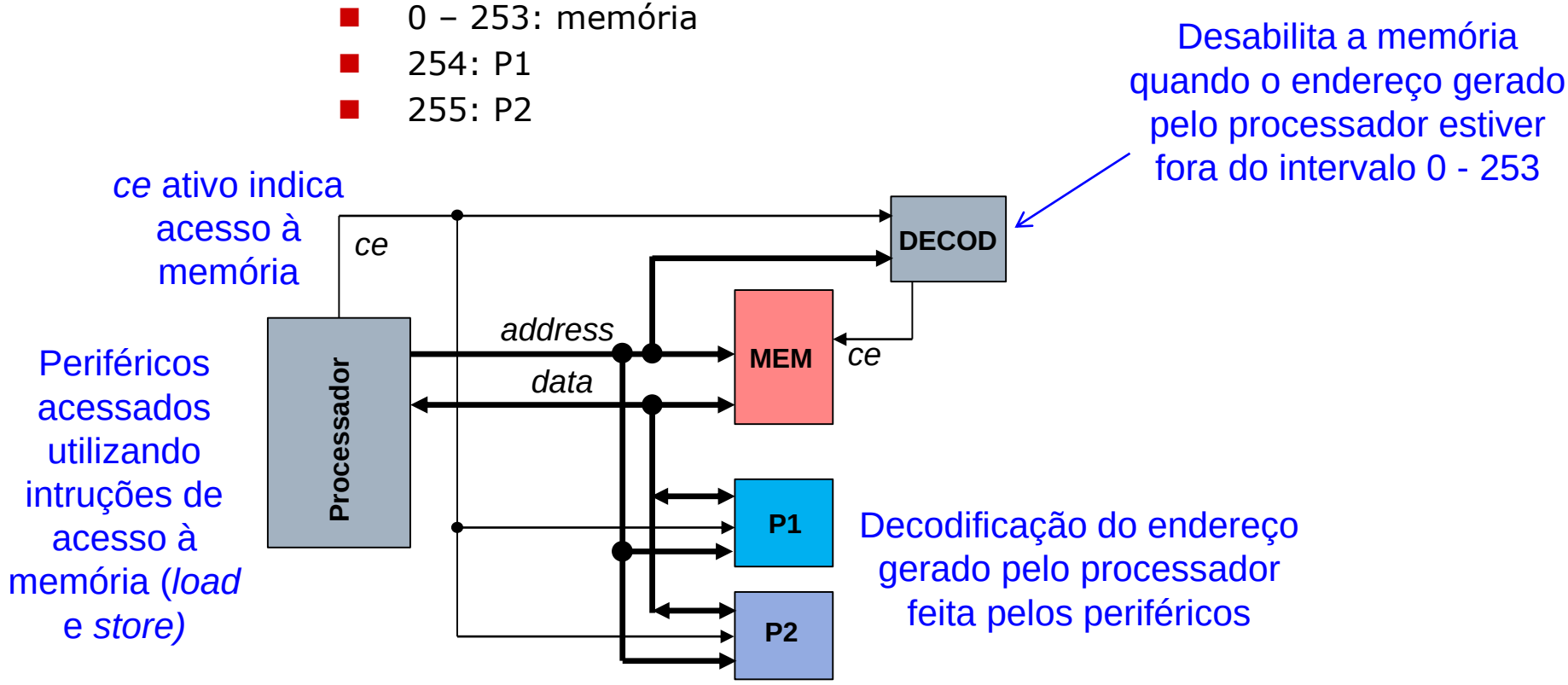
Exemplo:

```
lw $8, 16($0)    ; Lê da memória
lw $9, 254($0)   ; Lê de P1

sw $9, 128($0) ; Escreve na mem
sw $8, 255($0) ; Escreve em P2
```

Interface entre Processador e Periféricos

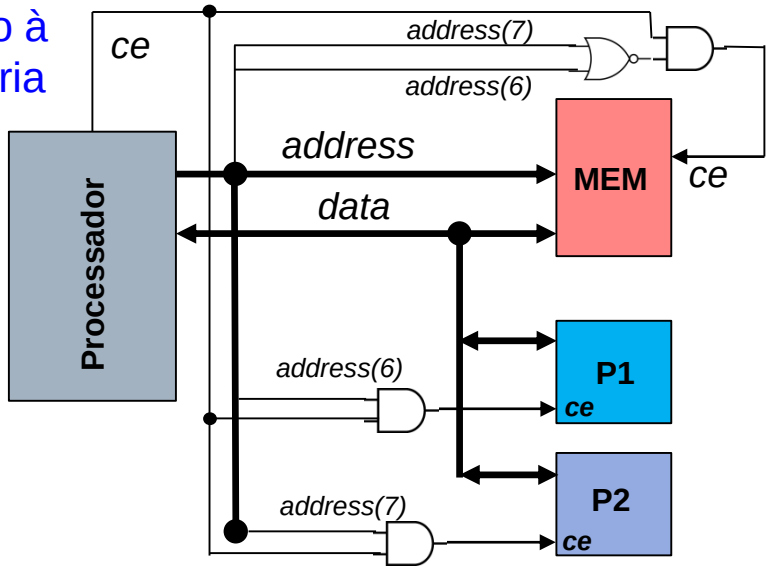
- Mapeamento em memória
  - Exemplo: barramento de endereços de 8 bits
    - Espaço de endereçamento: 0 – 255
      - 0 – 253: memória
      - 254: P1
      - 255: P2



Interface entre Processador e Periféricos

- Mapeamento em memória
  - Exemplo: barramento de endereços de 8 bits
    - Espaço de endereçamento: 0 – 255
      - 0 – 63 (**00**000000-**00**111111): memória
      - 64 – 127 (**01**000000-**01**111111): P1
      - 128-191 (**10**000000-**10**111111): P2
      - 192-255 (**11**000000-**11**111111): P1 e P2 (simultâneos)

ce ativo indica acesso à memória



Periféricos acessados utilizando intruções de acesso à memória (*load* e *store)*

Elimina decodificadores, os quais podem ser bem grandes em termos de portas lógicas

Interface entre Processador e Periféricos

- Mapeamento em portas de E/S
  - Dois espaços de endereçamento isolados
    - Um espaço é usado para acessar a memória
    - Outro é usado para acessar as interfaces de E/S
    - Exemplo considerando 8 bits no barramento de endereços
      - 0 – 255: memória (256 endereços de memória)
      - 0 – 255: periféricos (256 endereços de periféricos)
    - O processador tem uma saída (e.g. io) que indica o espaço de endereçamento em uso (e.g. io = 1: periférico)
  - Periféricos são acessados utilizando instruções específicas
    - In/Out
  - Exemplos
    - Processadores da Intel

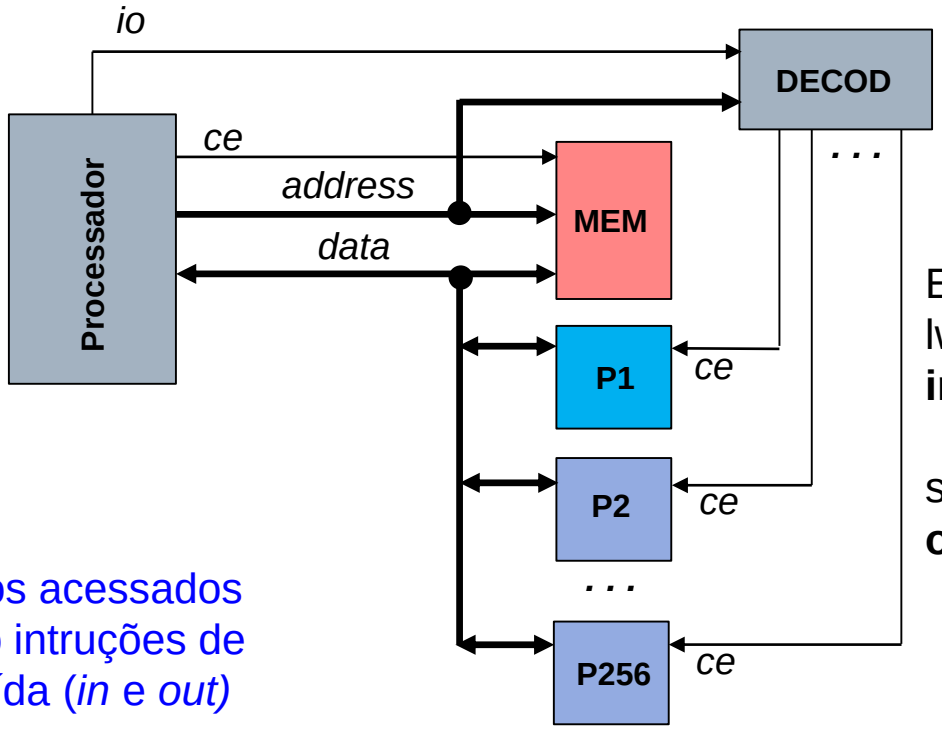
Interface entre Processador e Periféricos

- Mapeamento em portas de E/S
  - Exemplo: barramento de endereços de 8 bits
    - Espaço de endereçamento: 0 – 255
      - 0 – 255: memória
      - 0 – 255: periféricos (0: P1; 1:P2; ... ;255: P256)

io ativo indica acesso a periféricos

ce ativo indica acesso à memória

Habilita periféricos de acordo com o endereço gerado pelo processador



Exemplo:

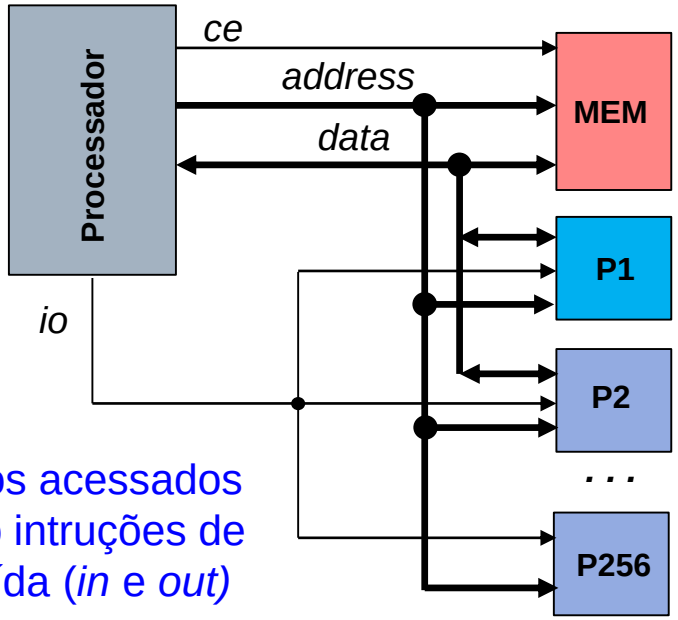
```
lw $8, 0($0)    ; Lê da memória
in $9, 0($0)    ; Lê de P1 (io=1)

sw $9, 1($0)    ; Escreve na mem
out $8, 1($0)   ; Escr. em P2 (io=1)
```

Periféricos acessados utilizando intruções de entra/saída (*in* e *out)*

Interface entre Processador e Periféricos

- Mapeamento em portas de E/S
  - Exemplo: barramento de endereços de 8 bits
    - Espaço de endereçamento: 0 – 255
      - 0 – 255: memória
      - 0 – 255: periféricos (0: P1; 1:P2; ... ;255: P256)



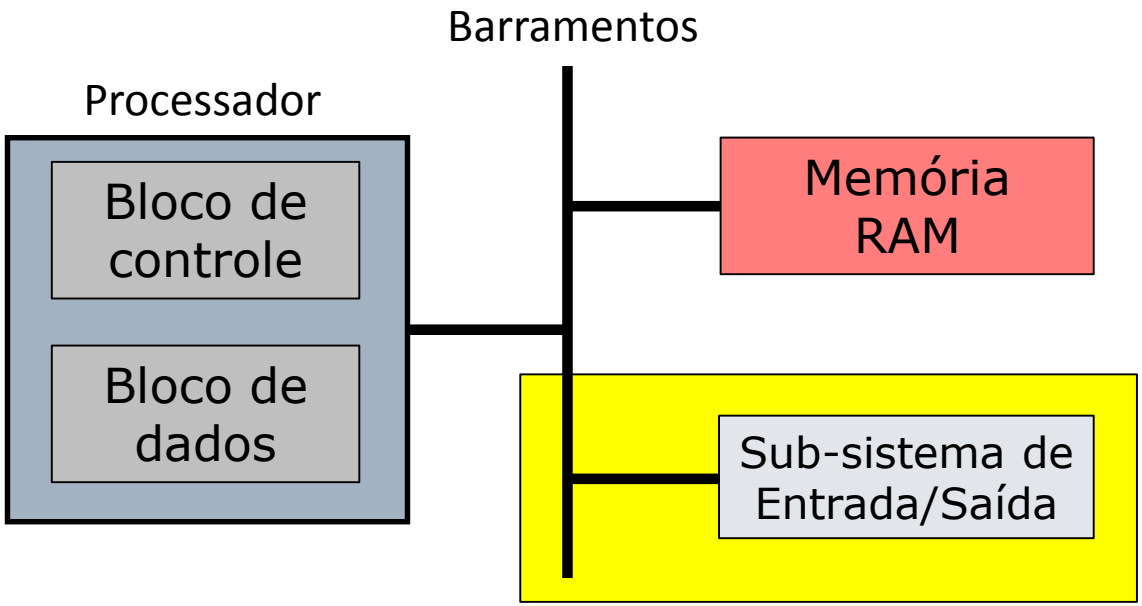
io ativo indica acesso a periféricos

Periféricos acessados utilizando intruções de entra/saída (*in* e *out)*

Decodificação do endereço gerado pelo processador feita pelos periféricos

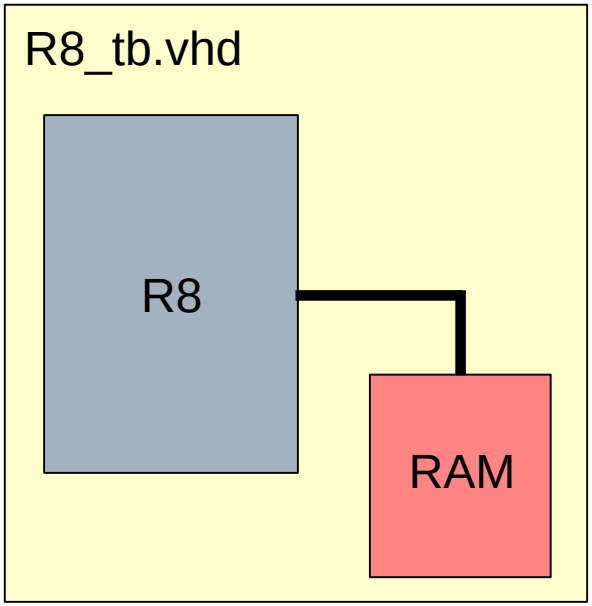
Interface entre Processador e Periféricos

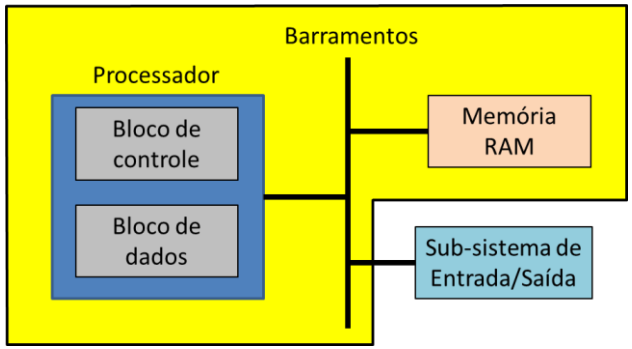
- Estrutura básica de um sistema computacional
  - Principais componentes



Interface entre Processador e Periféricos

- Estrutura básica de um sistema computacional
  - Sistema atual (R8_tb.vhd)

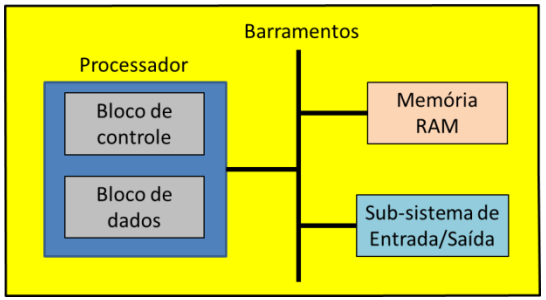




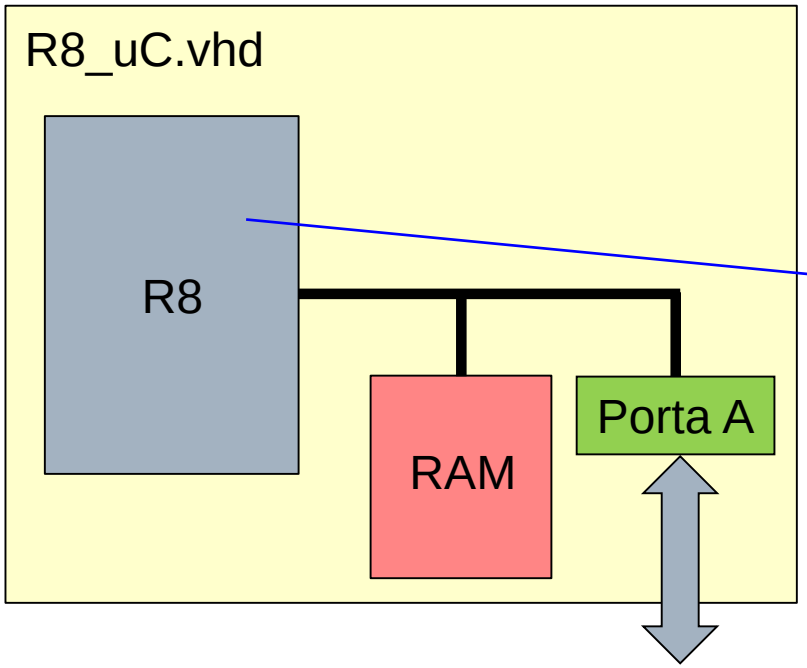
O objetivo do trabalho é adicionar ao sistema duas portas de E/S

Interface entre Processador e Periféricos

- Estrutura básica de um sistema computacional
  - Porta de E/S adicionada



Nova entidade: R8_uC.vhd

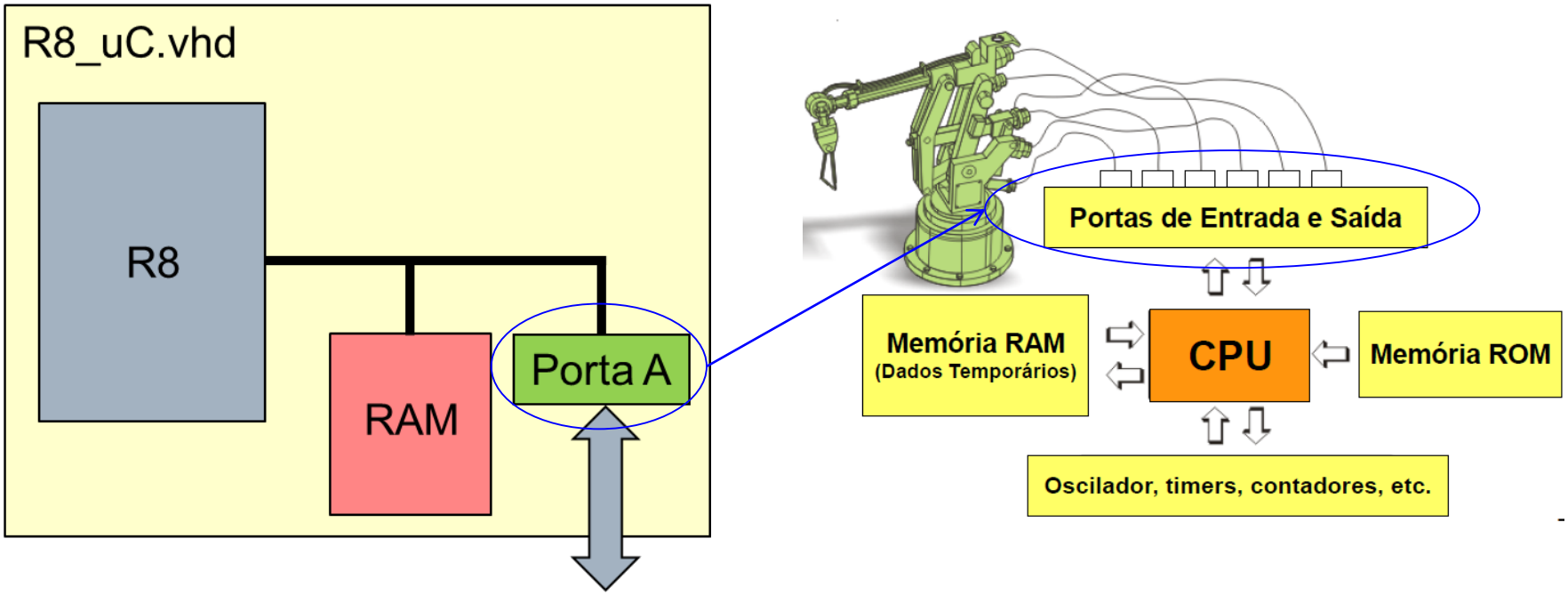


Utilizar a descrição comportamental do trabalho 1

O objetivo do trabalho é adicionar ao sistema uma porta de E/S

Interface entre Processador e Periféricos

- Estrutura básica de um sistema computacional
  - Porta de E/S adicionada

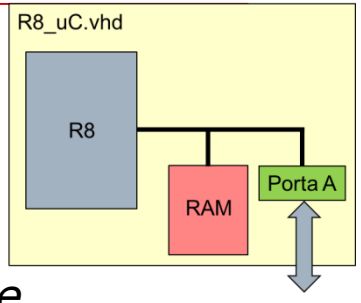


O objetivo do trabalho é adicionar ao sistema uma porta de E/S

Interface entre Processador e Periféricos

- Trabalho 2 – parte 1
  - A porta de E/S deve ter 16 bits para interface com o mundo externo
  - Os bits da porta devem ser individualmente configuráveis como entrada ou saída
    - Exemplo
      - PortaA(15:10) e PortaA(1:0) - entrada
      - PortaA(9:2) - saída
  - A configuração dos bits é controlada por um registrador (16 bits) onde cada bit corresponde à configuração de um bit da porta
    - 0: saída (*output*)
    - 1: entrada (*input*)
  - Deve haver também um registrador que habilita individualmente a utilização dos bits da porta de E/S e um registrador de dados (ambos 16 bits)

Interface entre Processador e Periféricos

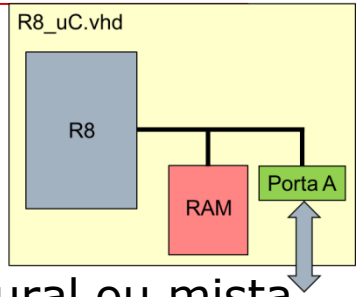


- Trabalho 2 – parte 1
  - Interface VHDL da porta de E/S
    - Arquivo BidirectionalPort.vhd disponível no *moodle*
    - Descrição VHDL pode ser comportamental, estrutural ou mista

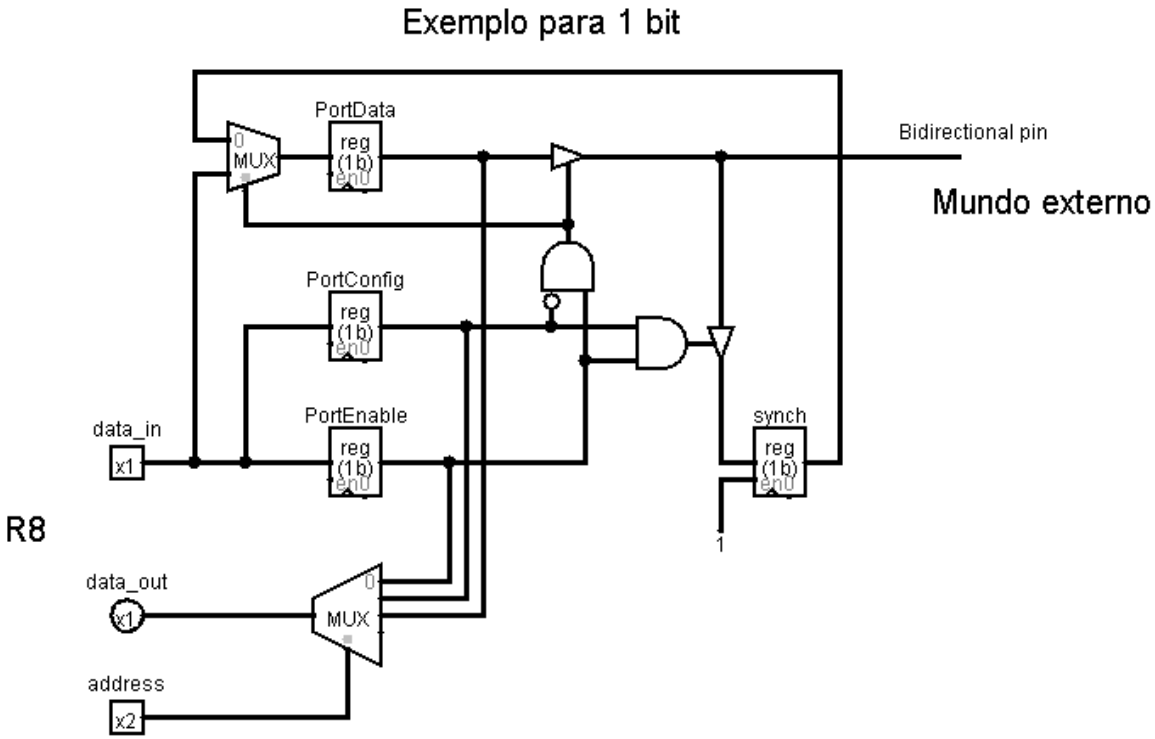
```
entity BidirectionalPort  is
    generic (...);
    port (
      clk      : in std_logic;
      rst      : in std_logic;
       -- Interface com o processador
      data_i   : in std_logic_vector (DATA_WIDTH-1 downto 0);
      data_o   : out std_logic_vector (DATA_WIDTH-1 downto 0);
      address : in std_logic_vector (1 downto 0); -- NÃO ALTERAR!
      rw       : in std_logic; -- 0: read; 1: write
      ce       : in std_logic;
      -- Interface com o mundo externo
      port_io : inout std_logic_vector (DATA_WIDTH-1 downto 0)
);
end BidirectionalPort;
```

Esta interface não deve ser alterada!

Interface entre Processador e Periféricos

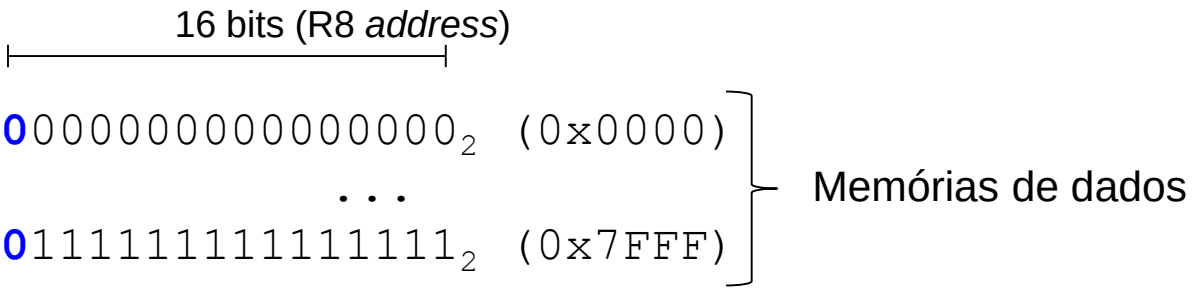


- Trabalho 2 – parte 1
  - Circuito resumido da porta de E/S
    - Descrição VHDL pode ser comportamental, estrutural ou mista

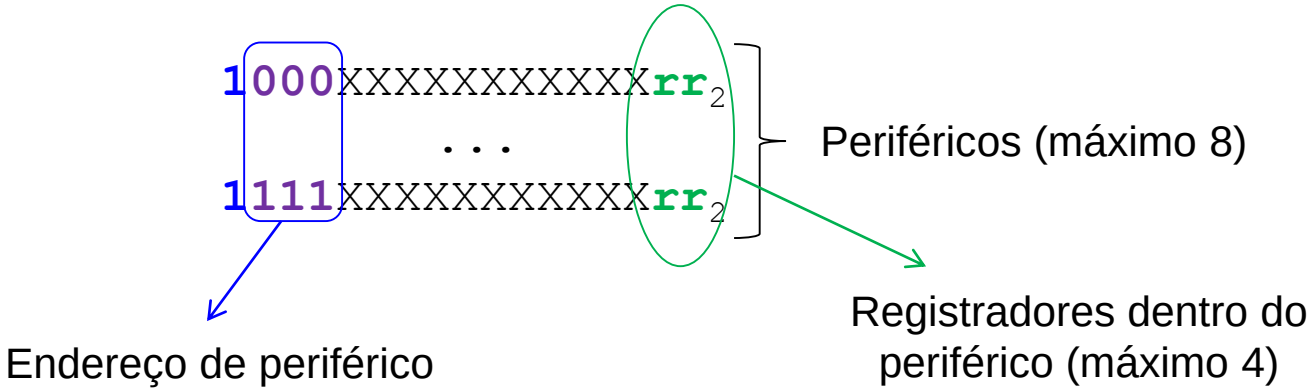


Interface entre Processador e Periféricos

- Trabalho 2 – parte 1
  - Mapear os registradores da porta em memória
  - Os endereços devem seguir o seguinte formato

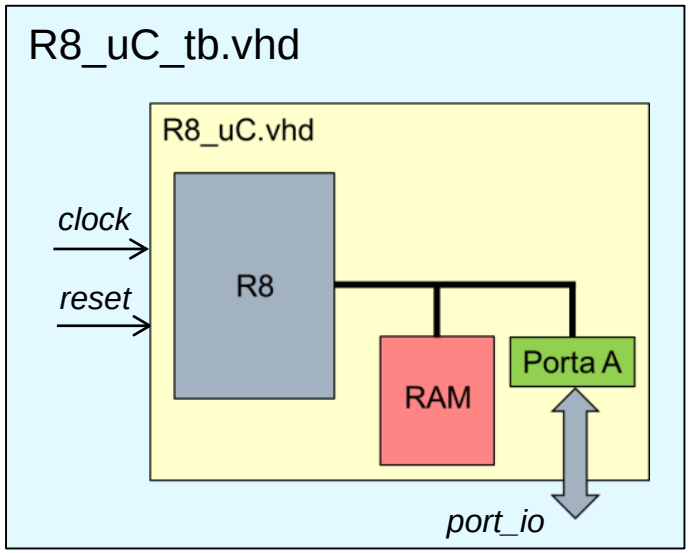


MSb do endereço indica
0: memória
1: Entrada/Saída



Interface entre Processador e Periféricos

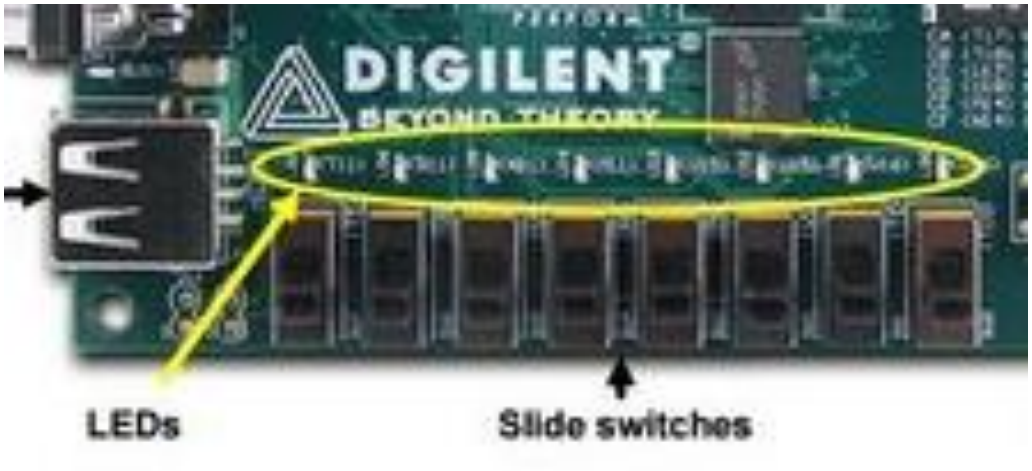
- Trabalho 2 – parte 1
  - A memória de dados deve ser desabilitada durante o acesso aos registradores da porta de E/S
  - Estrutura dos arquivos VHDL
    - A interface do R8_uC contém apenas *clock*, *reset* e *port_io*
    - *O test bench* deve gerar o *clock*, *reset* e estímulos para os bits de *port_io* que forem configurados como entrada



Apresentar diagrama correspondente ao R8_uC.vhd
Devem aparecer as ligações de todas interfaces do processador, memória e porta de E/S (address, data_i, data_o, ce, rw, ...)
Sugestão de editor : Dia Diagram Editor

Interface entre Processador e Periféricos

- Trabalho 2 – parte 1
  - Aplicação
    - Ler o estado das *slide switches* da placa Nexys 3 e mostar nos LEDs utilizando a porta de E/S



Interface entre Processador e Periféricos

- Trabalho 2 – parte 1
  - Manter mesmos grupos do trabalho 1
  - Apresentação dia 28/4
  - A nota do trabalho dará ENORME ÊNFASE à execução correta da simulação e prototipação
  - A apresentação será oral, teórico-prática, frente ao computador, onde o grupo deverá explicar ao professor o projeto, a simulação e a implementação