

Duração do Teste: 2 horas e 30 minutos

O diagrama ilustra a arquitetura de um processador de 8 bits, dividido em duas unidades principais: a Unidade de controle e a Unidade de processamento.

Unidade de controle: Contém o PC (Program Counter), o SR (Status Register) e o Descodificador de Instruções. O PC recebe o endereço da memória e envia o endereço da próxima instrução. O SR recebe o status do processador e envia o endereço da próxima instrução. O Descodificador de Instruções recebe o código de instrução (OP_CODE) e envia os sinais de controle (SO, SR, SD, RD, WR, ER, EP) para as outras unidades.

Unidade de processamento: Contém o Sign Ext (Extensão de Sinal), o Zero Fill (Preenchimento Zero), o Banco de Registos (8 x 8 bits), o ALU (Arithmetic Logic Unit) e o PSW (Program Status Word). O Sign Ext recebe o sinal de extensão de sinal (SO) e envia o sinal de extensão de sinal (SE). O Zero Fill recebe o sinal de preenchimento zero (ZF) e envia o sinal de preenchimento zero (ZF). O Banco de Registos recebe os dados de 8 bits e envia o endereço da próxima instrução. O ALU recebe os dados de 8 bits e envia o resultado de 8 bits. O PSW recebe o sinal de status (EP) e envia o sinal de status (EP).

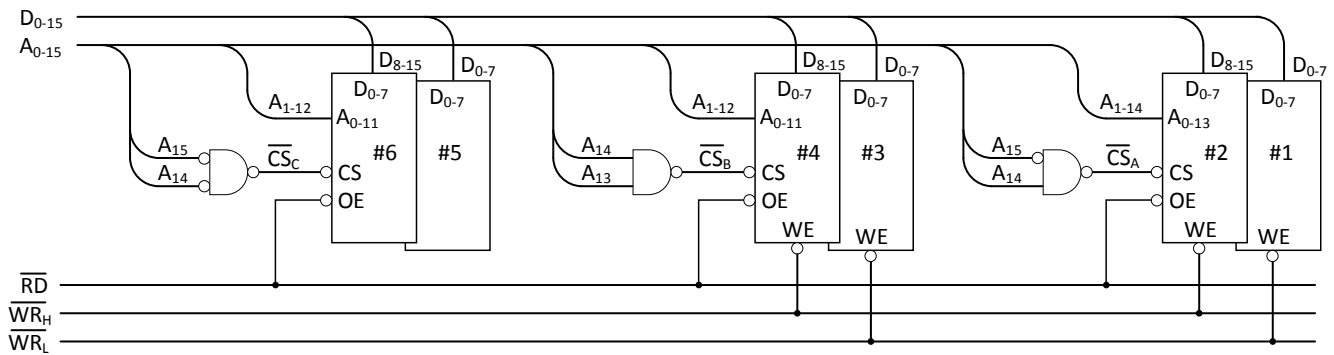
Memória: A Memória Código (256x8) recebe o endereço da próxima instrução e envia o código de instrução (OP_CODE). A Memória Dados (256x8) recebe o endereço da próxima instrução e envia o dado de 8 bits.

Operações do ALU:

OP_ALU	Operação
00	A + B
01	A - B
10	A - 1
11	A

N.º	Instrução	Codificação									Descrição
		b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	
1	str rx, [ry]	A definir									M[ry] = rx
2	mov rx, #imm3	A definir									rx = imm3
3	add rx, ry	1	0	0	rx ₂	rx ₁	rx ₀	ry ₂	ry ₁	ry ₀	rx = rx + ry
4	push rx	A definir									M[--SP] = rx
5	bne offset6	1	0	1	o ₅	o ₄	o ₃	o ₂	o ₁	o ₀	(Z==0) ? PC = PC + offset6 : PC = PC + 1
6	b rx	A definir									PC = rx

[2] Considere o diagrama da figura abaixo que descreve um exemplo de descodificação de endereços, correspondente ao mapeamento destes dispositivos de memória, em torno de um processador P16.



- a) [1,0 val] Indique os tipos e as dimensões (endereços e dados) dos dispositivos #1 a #6, individualmente tomados, e as capacidades (em quilobytes – KB) dos pares que formam.
- b) [1,0 val] Desenhe o mapa de endereçamento do conjunto, indicando os tipos, as dimensões e os endereços de início e de fim do espaço atribuído a cada par de dispositivos, inscrevendo igualmente, se for o caso, a ocorrência de subaproveitamento ou de *fold-back* e a localização de eventuais zonas livres e de zonas interditas (também designadas por “conflito”).
- c) [1,5 val] Continue o exemplo da tabela abaixo, para as instruções nas linhas 24 a 30 do troço de código dado, completando na sua folha de teste o registo da atividade dos barramentos e dos sinais em referência, observados passo-a-passo (*single step*) durante a sua execução, admitindo que o código é executado sobre o sistema apresentado na figura. A listagem foi produzida pelo *Assembler PAS v1.2.2*.

21

22 1000 2330 str r3, [r2, r0]

23 1002 5365 mov r3, 0x55

24 1004 1338 strb r3, [r1, r0]

25 1006 80A0 add r0, r0, 1

26 1008 1438 strb r4, [r1, r0]

27 100A 2300 ldr r3, [r2]

28 100C 1400 ldr r4, [r1]

29 100E 0178 movt r1, 0x80

30 1010 1400 ldr r4, [r1]

31

CTRL			ADDR	DATA	Instruction
nWRH	nWRL	nRD	A15 ... A0	D15 ... D0	
H	H	L	1000	3023	str r3, [r2, r0]
L	L	H	E000	1234	
H	H	L	1002	6553	mov r3, 0x55
			1004		

Exemplo para copiar e completar.

Atividade dos barramentos observados passo-a-passo, com os seguintes valores iniciais:

r0 = 0x0000; r1 = 0x6000; r2 = 0xE000;

r3 = 0x1234; r4 = 0x89AB; pc = 0x1000.

Nota – genericamente, no barramento de dados pode ocorrer: um valor concreto; alta impedância – ZH; ou conflito – conf.

- d) [1,5 val] Pretende-se reformular a seleção de endereços dos módulos RAM, cumprindo os seguintes critérios:

- A dimensão do espaço atribuído a cada memória RAM é coincidente com a sua capacidade;
- Não há zonas interditas (de “conflito”);
- As memórias RAM ficam em endereços contíguos entre si.

Desenhe o novo mapa de endereçamento e apresente as expressões lógicas dos novos sinais de *chip select*.

[3] Atente as seguintes implementações das funções `compare_and_swap` e `bubble_sort`, em que os tipos `int16_t` e `uint16_t` representam valores inteiros e naturais, respetivamente, codificados com 16 bits.

```
int16_t compare_and_swap( int16_t v[], uint16_t idx ) {
    int16_t temp;

    if ( v[idx] > v[idx + 1] ) {
        temp = v[idx];
        v[idx] = v[idx + 1];
        v[idx + 1] = temp;
        return 0;
    }
    return 1;
}

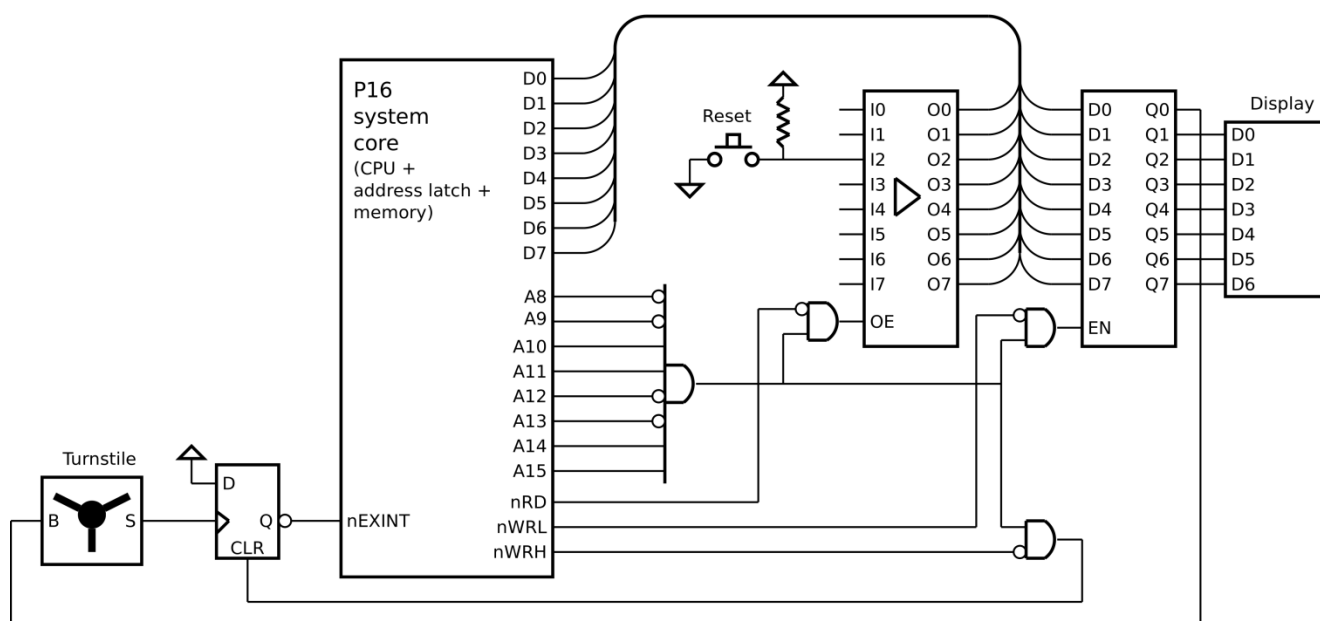
void bubble_sort( int16_t v[], uint16_t n ) {
    uint16_t i, j;

    for ( i = 1; i < n; i++ )
        for ( j = 0; j < n - i; j++ )
            compare_and_swap( v, j );
}
```

Considerando que *i)* os parâmetros das funções são passados nos primeiros quatro registos de uso geral do processador, o primeiro em `r0`, e o segundo em `r1` e assim sucessivamente, *ii)* o valor de retorno das funções é devolvido no registo `r0`, *iii)* as funções devem preservar o conteúdo dos registos que utilizam para além dos convencionados para passagem de parâmetros (`r0` a `r3`) e *iv)* o *stack pointer* foi devidamente inicializado,

- a) [2,5 val] Implemente a função `compare_and_swap` na linguagem *assembly* do P16, definindo, se necessário, as respetivas variáveis.
- b) [2,5 val] Implemente a função `bubble_sort` na linguagem *assembly* do P16, definindo, se necessário, as respetivas variáveis.

[4] Para gerir a ocupação de um recinto de espetáculos com capacidade para 100 pessoas, pretende-se implementar um sistema de controlo de acesso recorrendo ao processador P16 e ao *hardware* indicado na figura.



Neste sistema, o torniquete (Turnstile) é utilizado para controlar a entrada de pessoas no recinto, o mostrador (Display), com três dígitos decimais, informa sobre o número de pessoas presentes nesse espaço e o botão Reset serve para iniciar, a qualquer momento, o contador de pessoas com o valor zero.

Relativamente ao torniquete, a entrada **B** serve para bloquear a entrada de pessoas no recinto. Quando a entrada **B** não está ativa, a passagem de uma pessoa no torniquete faz ativar a sua saída **S**, por um período de tempo variável, igual ao da passagem da pessoa.

Com vista à realização global deste sistema:

- [1,5 val] Escreva a rotina que avalia o estado do botão Reset e retorna um quando ele está pressionado e zero no caso contrário.
- [1,5 val] Escreva a rotina que, sem comprometer a condição de controlo do torniquete, afixa no mostrador o valor que lhe é passado como argumento, manipulando a variável global `outport_image`.
- [1,5 val] Escreva a rotina para o atendimento da interrupção (ISR), responsável por *i*) manter atualizado o valor da variável global `people_counter`, que representa o número de pessoas presentes no recinto; e *ii*) bloquear o funcionamento do torniquete, por manipulação da variável global `outport_image`, quando a quantidade de pessoas presentes no recinto atingir o número 100.
- [0,5 val] Indique a motivação para a utilização do *flip-flop* do tipo D na ligação da saída **S** do torniquete à entrada de interrupções do processador P16 (`nEXINT`).

Nota: Defina todos os símbolos e variáveis que entender necessários para a realização das alíneas a) a c).