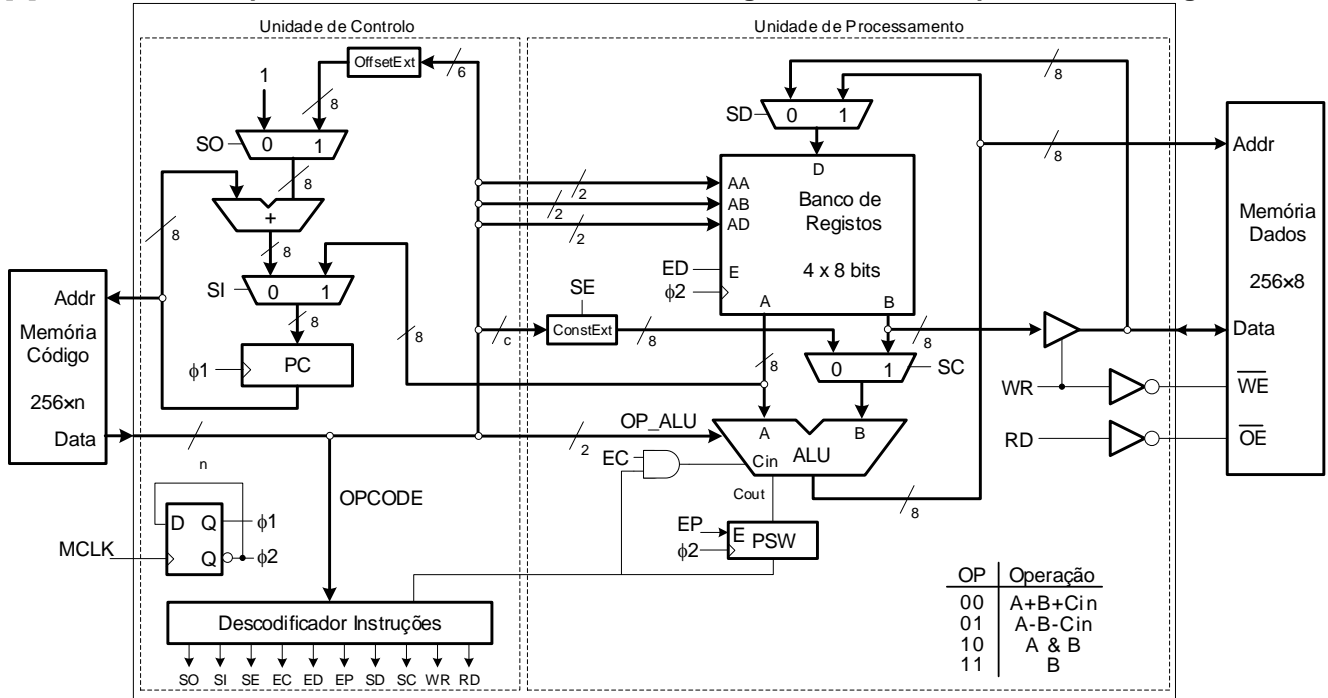


INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
LEIC, LEETC, LEIRT
Arquitetura de Computadores
2.º Teste (03/jul/2018)

Duração do Teste: 2 horas e 30 minutos

[1] Considere um processador, de ciclo único, com o diagrama de blocos apresentado na figura.

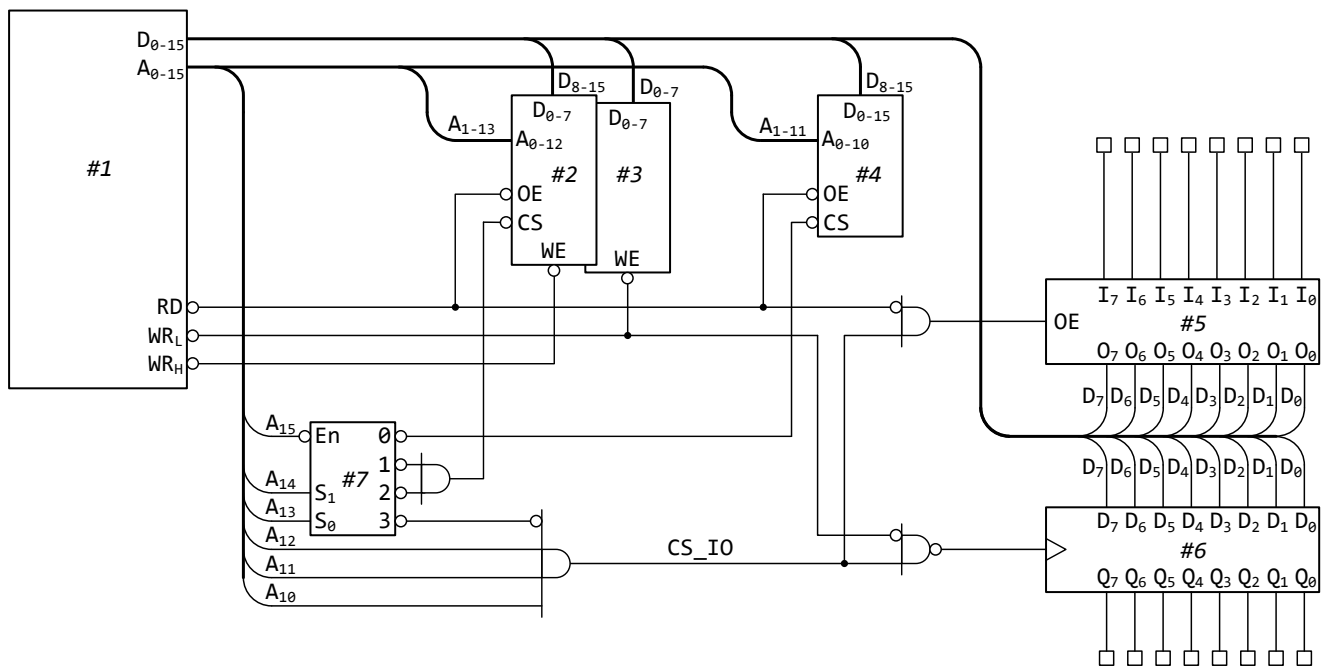


O processador suporta a execução do seguinte conjunto de instruções, em que a constante `const` representa números naturais e a constante `offset` representa números relativos:

N.º	Instrução	Codificação									Descrição
		b ₈	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	
1	ldi rx, #const	0	1	1	C ₃	C ₂	C ₁	C ₀	rx ₁	rx ₀	rx = const
2	ld rx, [ry]	A definir									rx = M[ry]
3	and rx, ry, rz	0	1	0	ry ₁	ry ₀	rz ₁	rz ₀	rx ₁	rx ₀	rx = ry & rz
4	sbb rx, ry, rz	A definir									rx = ry - rz - cin
5	inc rx	A definir									rx = rx + 1
6	jmp rx	A definir									PC = rx
7	jnc offset	1	1	0	O ₅	O ₄	O ₃	O ₂	O ₁	O ₀	(C==0) ? PC = PC + offset : PC = PC + 1
8	decbst rx, [ry]	A definir									--ry; M[ry] = rx

- Codifique as instruções `ld`, `sbb`, `inc`, `jmp` e `decbst` utilizando uma codificação linear a 3 bits. Explícite os bits do código de instrução que correspondem aos sinais `AA`, `AB`, `AD`, `OP_ALU` e `OPCODE`. [2,5 val.]
- Proponha um diagrama lógico para o módulo `ConstExt`. Justifique a sua resposta. [0,5 val.]
- Considerando que o módulo `Descodificador Instruções` é implementado usando exclusivamente uma ROM, apresente o conteúdo desta memória e indique a sua capacidade em bits. [2 val.]

[2] Considere o sistema computacional baseado no PDS16 representado na figura.



Legenda do Diagrama de Blocos

(inclui uma breve descrição dos elementos representados)

- #1 : módulo do CPU – contém o microprocessador **PDS16** e o registo externo de endereços (*Address Latch*)
 #2 : dispositivo de memória do tipo: _____ – dimensão: ____ K * ____
 #3 : dispositivo de memória do tipo: _____ – dimensão: ____ K * ____
 #4 : dispositivo de memória do tipo: _____ – dimensão: ____ K * ____
 #5 : porto de _____ a ____ bits, baseado num *Octal Buffer with Three-state Outputs*
 #6 : porto de _____ a ____ bits, baseado num _____
 #7 : Descodificador, com entrada de controlo (*Enable*) e quatro saídas, todas 'ativas em zero' (*active low*)

- a) Complete a legenda, onde se inscrevem a designação e as características principais dos elementos que compõem o diagrama de blocos. [1 val.]
- b) Desenhe o mapa de endereçamento do sistema (incluindo as modificações que vier a realizar nas alíneas seguintes), indicando a funcionalidade, as dimensões, os endereços de início e de fim do espaço atribuído a cada dispositivo, inscrevendo igualmente, se for o caso, a ocorrência de *fold-back*. [2 val.]
- c) Recorrendo a dispositivos do tipo dos já incluídos no diagrama apresentado (#5 e #6), desenhe o esquema que estende os portos existentes, de modo a que passem ambos a poder ser usados também *word-wise*. [1 val.]
- d) Desenhe o esquema de um módulo de RAM adicional, com apenas dois circuitos RAM de 32K*8, acessível *byte-wise* e *word-wise*, que ocupe todo o espaço de endereçamento que tenha restado livre. [1 val.]

[3] Considere as seguintes funções expressas em linguagem C:

```
uint8 is_parity_odd(uint16 value) {
    uint8 counter = 0;
    while (value > 0) {
        if ((value & 1) == 1) {
            counter++;
        }
        value = value >> 1;
    }
    return (counter & 1);
}

void array_parity_odd(uint16 in[], uint8 out[], uint8 size) {
    for (uint8 i = 0; i < size; ++i)
        out[i] = is_parity_odd(in[i]);
}
```

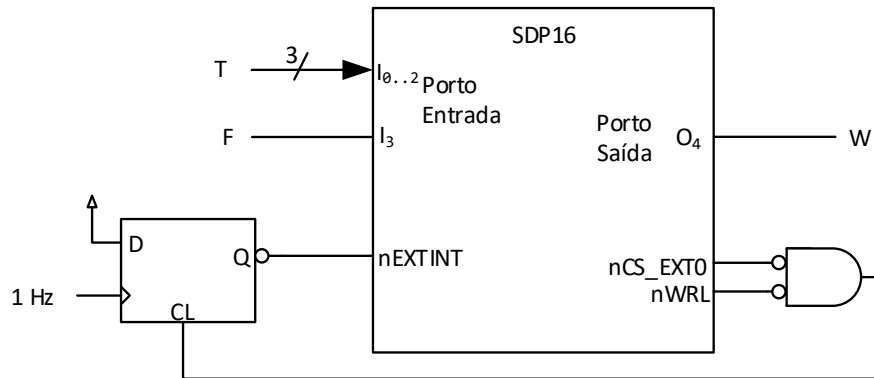
- a) Traduza para linguagem *assembly* do PDS16 a função `is_parity_odd` e, se necessário, defina as respectivas variáveis. [2,5 val.]
- b) Traduza para linguagem *assembly* do PDS16 a função `array_parity_odd` e, se necessário, defina as respectivas variáveis. [2,5 val.]

Notas:

1. Com vista ao alojamento de variáveis, assuma que a secção ".data" está localizada na zona de memória acessível com endereçamento direto.
2. Na programação em *assembly* deve usar as seguintes convenções: os parâmetros das funções são passados em registos, ocupando a quantidade necessária, pela ordem `r0`, `r1`, `r2` e `r3`; o valor de retorno de uma função, caso exista, é devolvido em `r0`; `int8` e `int16` significam valores inteiros com sinal representados a 8 e a 16 bit, respetivamente; `uint8` e `uint16` significam valores inteiros sem sinal representados a 8 e a 16 bit. A função preserva os registos que utiliza para além dos usados para parâmetros.

[4] Tendo como base o sistema SDP16, pretende-se implementar um sistema que gera uma onda quadrada.

- Enquanto o sinal F estiver ativo, o sistema gera onda completa. Caso contrário, mantém a saída a '1';
- O período da onda W, em segundos, é igual ao dobro do valor da entrada T;
- A onda deve ter um *duty cycle* de 50%.



- Escreva em *assembly* do PDS16 a rotina de atendimento da interrupção. [2,5 val]
- Tendo como base a rotina de atendimento da interrupção, escreva em *assembly* do PDS16 o sistema de geração da onda quadrada. [2,5 val]