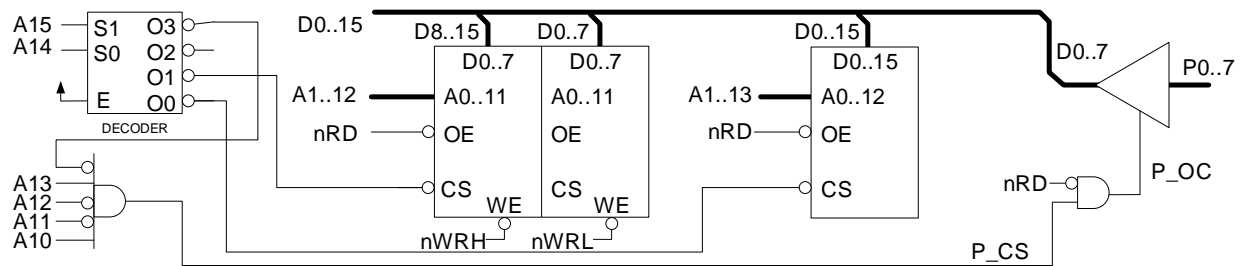


[2] Considere o sistema computacional baseado no PDS16 representado na figura.



- 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.]
- Adicione uma nova RAM de 16 kbytes e uma nova ROM de 8 Kbytes utilizando circuitos integrados de 8K*8 e os circuitos necessários para completar a seleção de endereços. [2 val.]
- Adicione um porto de saída de 16 bits com acesso a 8 e a 16 bits e estenda o porto de entrada para dar suporte a acessos a 16 bits. [1 val.]

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

```

uint8 positiveAcc(int16 array[], uint8 size) {
    uint8 counter = 0;
    for (i = 0; i < size; i++)
        if (array[i] >= 0) {
            counter++;
        }
    return counter;
}

int8 isNegative(int16 array[], uint8 size) {
    int16 acc;

    if (positiveAcc(array, size) = 0)
        return 1;
    else
        return -1;
}

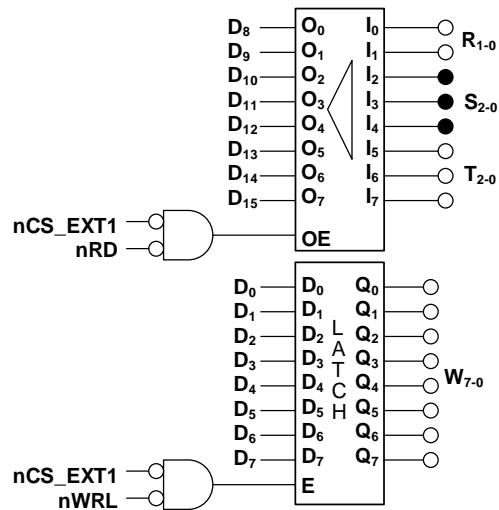
```

- a) Traduza para linguagem *assembly* do PDS16 a função `positiveAcc` e, se necessário, defina as respectivas variáveis. [3,0 val.]
- b) Traduza para linguagem *assembly* do PDS16 a função `isNegative` 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] Considere o circuito representado na figura como parte de um sistema baseado no kit SDP16.



- Programa a função `uint8 input(uint8 sel)` que realiza a leitura do porto de entrada e devolve o sinal S se `sel = 0`, T se `sel = 1`, $S + T$ se `sel = 2` e $S - T$ se `sel = 3`. S e T são números relativos. [3 val]
- Escreva um programa que, continuamente, lê os valores de R, S e T e escreve no porto de saída o resultado das operações de acordo com a alínea anterior. Deve utilizar a função desenvolvida em a). [1,5 val]