

17. Espaço de I/O (entrada/saída).....	2
17.1 Portos paralelos de entrada e saída.....	2
17.1.1 Porto paralelo de saída.....	2
17.1.2 Porto paralelo de entrada.....	3
17.1.1 Exercício.....	3

17. ESPAÇO DE I/O (ENTRADA/SAÍDA)

O objectivo principal de um CPU é providenciar interlocução com o exterior, no sentido de receber, processar e expedir informação de acordo com algoritmos que lhe são determinados por programas carregados em memória. Esta interlocução é estabelecida através de dispositivos periféricos tais como teclados, monitor, discos, etc., onde o CPU escreve e lê informação. A escrita e a leitura nos dispositivos periféricos é realizada através de portos de entrada/saída (Input/Output), e que são acedidos pelo CPU de uma forma semelhante à que acede aos dispositivos de memória. Na verdade o P16_V1 não distingue o espaço de memória do espaço de I/O, ou seja, o I/O é tratado como um dispositivo de memória. Esta funcionalidade tem como vantagem o facto de não ser necessário disponibilizar instruções específicas para a entrada e saída de dados. A desvantagem é tornar a descodificação de endereços mais complexa devido à heterogeneidade existente entre as dimensões dos dispositivos de memória e os de I/O.

17.1 Portos paralelos de entrada e saída

Denomina-se porto paralelo a um porto de entrada ou de saída, que disponibiliza vários bits em simultâneo. Estes portos poderão ser utilizados para controlo individual de actuadores, observação instantânea de sensores ou troca de informação entre sistemas. Na troca de informação entre sistemas é necessário recorrer a um protocolo de validação e controlo de fluxo, ou seja, um conjunto de regras estabelecidas entre os interlocutores que assegure a correcta transferência de informação entre ambos.

17.1.1 Porto paralelo de saída

Um porto paralelo de saída é constituído por um registo inserido no espaço de endereços do CPU no qual é possível escrever uma palavra. A palavra escrita no registo fica disponível em pinos através de um andar de saída, que lhe confere uma impedância capaz de interligar o CPU com os circuitos lógicos digitais mais comuns (Ex. TTL) bem como outro tipo de electrónica. A Figura 17-1 apresenta a estrutura característica de um porto de saída de 8 bits. O registo poderá ter característica *latch* ou *edge-triggered*, desde que a memorização se dê num momento em que os dados no bus se encontrem estáveis.

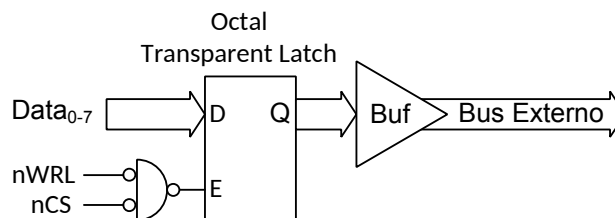


Figura 17-1 - Porto de Saída de 8 bits

17.1.2 Porto paralelo de entrada

Um porto paralelo de entrada, é constituído por um buffer TRISTAT que, quando endereçado pelo CPU, coloca em baixa impedância no bus de dados do CPU a informação presente nos pinos de entrada do porto. A Figura 17-2 apresenta a estrutura base de um porto de entrada paralelo de 16 bits.

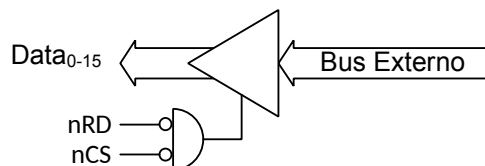


Figura 17-2 – Porto de Entrada de 16 bits

17.1.1 Exercício

Utilizando o processador P16, realizar um sistema para controlo de uma lâmpada temporizada L, accionada por um botão B. Ao premirmos o botão B o sistema tem o seguinte comportamento: se a lâmpada L se encontrar apagada, acende-se de imediato e permanece acesa durante um período de tempo, se já se encontrar acesa apaga-se de imediato.

Admita que o botão está ligado ao bit de peso 2 de um porto de entrada de 8 bits, localizado no endereço 0x8000, a lâmpada está ligado ao bit de menor peso de um porto de saída de 8 bits, localizado no mesmo endereço e o CPU tem uma frequência de trabalho de 100KHz (T=100us).

```
.section .startup
    b      _start
    b      .
_start:
    ldr    sp, addr_stack_top
    bl     main
    b      .

addr_stack_top:
    .word  stack_top

.section .stack
.equ     STACK_SIZE, 64
.space  STACK_SIZE
stack_top:

/*
int main() {
    while (true) {
        port_output(0)      apaga a lâmpada
        while ((port_input() & BUTTON_MASK) != 0)
            ;
        while ((port_input() & BUTTON_MASK) == 0)
            ;
        port_output(LAMP_MASK); acende a lâmpada
        delay_counter = TIME_DELAY;
        while ((port_input() & BUTTON_MASK) != 0 && delay_counter > 0)
            delay_counter--;
    }
}
```

```

        while ((port_input() & BUTTON_MASK) == 0 && delay_counter > 0)
            delay_counter--;
    }
*/

.equ    BUTTON_MASK, 1
.equ    LAMP_MASK, 1
.equ    TIME_DELAY, 10000

.text
main:
while:
    ldr    r1, addr_port
    mov    r0, 0
    strb   r0, [r1]          ; port_output(0)  apaga a lâmpada
    mov    r2, BUTTON_MASK
while1:
    ldrb   r0, [r1]          ; while ((port_input()
                            ;      & BUTTON_MASK) != 0)
    and    r0, r0, r2
    bzc    while1
while2:
    ldrb   r0, [r1]          ; while ((port_input()
                            ;      & BUTTON_MASK) == 0)
    and    r0, r0, r2
    bzs    while2
    mov    r0, LAMP_MASK
    strb   r0, [r1]          ; port_output(LAMP_MASK);
    mov    r3, TIME_DELAY & 0xff    ; delay_counter = TIME_DELAY;
    movt   r3, TIME_DELAY >> 8
while3:
    ldrb   r0, [r1]          ; while ((port_input()
                            ;      & BUTTON_MASK) != 0)
    and    r0, r0, r2
    bzs    while3_end
    sub    r3, r3, 0          ;      && delay_counter > 0
    bzs    while3_end
    sub    r3, r3, 1
    b      while3
while3_end:
while4:
    ldrb   r0, [r1]          ; while ((port_input()
                            ;      & BUTTON_MASK) == 0)
    and    r0, r0, r2
    bzc    while4_end
    sub    r3, r3, 0          ;      && delay_counter > 0
    bzs    while4_end
    sub    r3, r3, 1
    b      while4
while4_end:
    b      while

addr_port:
    .word  0xff00

```