



# Arquitetura de Computadores

LIC. EM ENG.<sup>a</sup> INFORMÁTICA  
FACULDADE DE CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA



## Lab 3 – Acesso a display de 7 segmentos e teclado hexadecimal

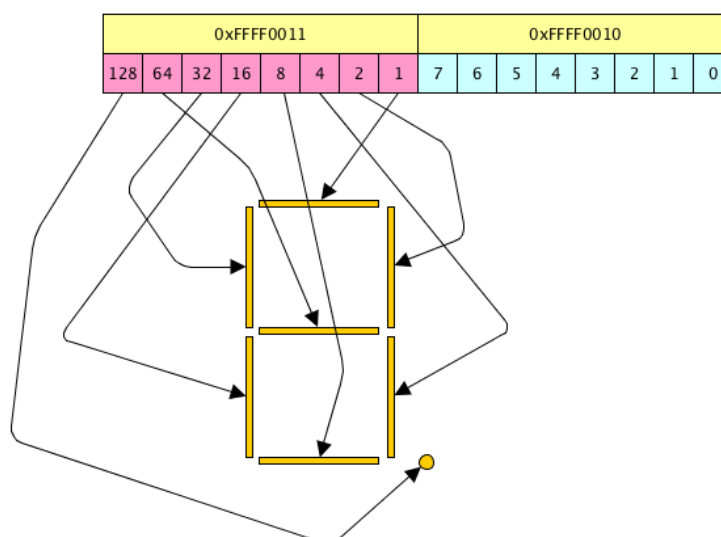
Os exercícios que se seguem devem ser executados através do MARS, um simulador da arquitetura MIPS disponível em <http://courses.missouristate.edu/KenVollmar/MARS/>. Este simulador inclui alguns periféricos simulados acessíveis através do menu Tools.

Na aula deve explicar todos os detalhes dos exercícios realizados.

### 1. Display de 7 segmentos

O processador MIPS não tem instruções específicas para aceder a periféricos (ao contrário de processadores como os da família Intel x86). Assim os periféricos são mapeados no espaço de endereçamento normal. Ou seja, no caso do "Digital Lab Sim", acessível no menu Tools, existem 3 periféricos: 2 displays de 7 segmentos e um teclado hexadecimal.

Para aceder ao display da direita basta escrever um byte no endereço 0xFFFF0010, enquanto o display da esquerda pode ser acedido através do endereço 0xFFFF0011. Note que cada bit do byte escrito corresponde a um dos elementos (barras e ponto) desse display. O bit 0 corresponde ao segmento no topo, o bit 1 corresponde ao segmento de cima à direita e assim sucessivamente. O bit 7 corresponde ao ponto. A figura ajuda a compreender a correspondência entre cada bit e o respetivo segmento.



- Faça um pequeno programa que num ciclo vá enviando bytes com apenas 1 bit a 1 para o display direito. Faça uso da instrução *shift left logical* (sll) para que, começando com 1 no bit menos significativo, vá deslocando o 1 um bit para a esquerda. Execute o programa passo a passo e verifique que os bits correspondentes a cada um dos segmentos estão de acordo com a figura acima.
- Faça uma tabela onde associe a cada número de 0 a F (hexadecimal) o correspondente valor que terá de ser enviado para o display de forma que sejam

apresentados os dígitos pretendidos.

- c) Escreva um programa que conte de 0 até 15 (0xF) e mostre o valor num dos displays disponíveis. *Note que pode carregar inicialmente em memória os códigos do display de 7 segmentos para cada um dos símbolos hexadecimais definido por si na alínea anterior.*
- d) Escreva um programa que vá contando de 0 até 255 (0xFF) e mostre nos dois displays o correspondente número em hexadecimal.
- e) Modifique o programa anterior para mostrar contagens de 0 a 99 em decimal.

## 2. Leitura do teclado por varrimento

O teclado simula um teclado normal em que as teclas são lidas através de um mecanismo de "scan". Isto significa que temos de ir testando linha a linha do teclado para ver se há alguma tecla premida. Assim para ler o teclado teremos de em ciclo ir enviando para o endereço 0xFFFF0012 o número da linha a testar, enviando sucessivamente cada um dos bits entre 0 e 3 activos. De seguida vamos ler no endereço 0xFFFF0014 se há alguma tecla premida nessa linha. O resultado lido se houver tecla premida será o composto por o número da linha (*nibble* menos significativo) e o número da coluna (*nibble* mais significativo), ou seja:

tecla 0 → 0x11, tecla 1 → 0x21, tecla 2 → 0x41, tecla 3 → 0x81, tecla 4 → 0x12..., tecla f → 0x88

Usando as funções desenvolvidas na questão anterior, faça um programa que esteja continuamente a testar o teclado e se houver uma tecla premida mostre no display direito o valor hexadecimal correspondente. Caso não haja tecla premida o display deve ficar apagado.