

Arquitetura de Computadores II

Exercícios adicionais da componente prática (guiões 1 a 6)

Exercício 1

- Escreva um programa em linguagem *assembly* que, em ciclo infinito, faça a leitura dos 4 bits do *dip-switch* da placa DETPIC32 (ligado aos portos RB3 a RB0) e mostre o valor de cada um deles nos 4 LED ligados aos portos RE3 a RE0.
- Altere o programa que escreveu na alínea a), de modo a que o valor de cada um dos bits do *dip-switch* seja mostrado por ordem inversa nos 4 LED (i.e. RB3 -> RE0, ..., RB0 -> RE3).
- Faça uma nova alteração ao programa que escreveu em a), de modo a que o valor de cada um dos bits do *dip-switch* seja mostrado por ordem inversa nos 4 LED ligados aos 4 bits mais significativos do porto E (i.e. RB3 -> RE4, ..., RB0 -> RE7); os 4 LED ligados aos 4 bits menos significativos do porto E devem mostrar o complemento para 1 do valor presente nas entradas RB3-RB0.

Exercício 2

Escreva um programa em *assembly* que execute em ciclo infinito, e que, sempre que o utilizador prime uma tecla no PC, faça as seguintes tarefas:

- tecla 0: deve acender apenas o LED ligado ao porto RE0;
- tecla 1: deve acender apenas o LED ligado ao porto RE1;
- tecla 2: deve acender apenas o LED ligado ao porto RE2;
- tecla 3: deve acender apenas o LED ligado ao porto RE3;
- outra tecla qualquer: os 4 LED devem acender e permanecer ligados durante 1 segundo; no final desse tempo os 4 LED devem ser apagados (este deve ser o estado inicial dos 4 LED). Qualquer tecla premida dentro desse intervalo de tempo deve ser ignorada.

Utilize o *core timer* do PIC32 para controlar a temporização.

Exercício 3

- Repita os exercícios anteriores escrevendo os respetivos programas em linguagem C usando, para o acesso aos bits dos registos dos portos, as estruturas de dados definidas pelo fabricante (e.g. `LATEbits.LATE0=PORTBbits.RB0;`).
- Repita a alínea anterior sem usar as estruturas de dados, ou seja, usando exclusivamente o acesso aos registos com sequências *read-modify-write* (e.g. `LATE=(LATE & ...)`).

Exercício 4

Acrescente à resolução que fez, em linguagem C, do exercício 2, a visualização nos dois *displays* de 7 segmentos. Se a tecla premida estiver na gama 0 a 3, o valor da tecla deve aparecer nos *displays* (por exemplo, 02). Se for premida outra qualquer tecla, deve aparecer o valor FF durante 1 segundo e a seguir os *displays* devem apagar-se (i.e., todos os segmentos devem ser desligados, a que corresponde também o estado inicial dos dois *displays*). Qualquer tecla premida dentro desse intervalo de tempo deve ser ignorada.

A frequência de atualização dos *displays* deve ser 100 Hz. Utilize o *core timer* do PIC32 para gerar as temporizações.

Exercício 5

- a) Escreva um programa em linguagem C que implemente, em ciclo infinito, um contador crescente módulo 100, atualizado com uma frequência de 10 Hz. O valor do contador deve ser impresso sempre na mesma linha do ecrã, formatado com 2 dígitos (nota: para fazer a impressão sempre na mesma linha comece por imprimir o caráter '\r').
Utilize o core timer do PIC32 para controlar a frequência de incremento do contador.
- b) Altere o programa que escreveu em a) de modo a que, adicionalmente, quando o utilizador carrega nas teclas 0 a 4 a frequência de incremento do contador seja modificada de acordo com a seguinte expressão: $\text{freq} = 2 * (1 + \text{tecla_carregada})$.
Sempre que o utilizador carregar na tecla ENTER deverá ser impresso o valor atual da frequência de contagem, a seguir ao valor do contador (por exemplo: 83, 20 Hz).
- c) Acrescente o módulo de visualização de modo a que, sempre que seja premida a tecla ENTER, seja mostrado nos dois *displays* o valor atual do contador (o valor mostrado deve permanecer inalterado, até ser novamente premida a tecla ENTER). A frequência de refrescamento dos *displays* deve ser 50Hz.

Exercício 6

Resolva as alíneas a) e b) do exercício anterior, escrevendo os respetivos programas em linguagem *assembly*.

Exercício 7

- a) Escreva um programa em linguagem C para escrever no ecrã, com uma frequência de 1Hz, o valor de cada um dos 4 bits do *dip-switch* ligado aos portos RB3-RB0. Por exemplo, se o porto tiver a combinação binária "1100", no ecrã deverá aparecer: DS4=1, DS3=1, DS2=0, DS1=0.
Nota: apenas pode usar os system calls `printStr()` e `putChar()`. Utilize o *core timer* para controlar a temporização.
- b) Altere o programa que escreveu em a) de modo a que a taxa de atualização da informação escrita no ecrã seja dependente do resultado da conversão da ADC, e esteja na gama [1, 5] Hz. Tenha em consideração que a ADC que utiliza é uma ADC de 10 bits, ou seja, o resultado da conversão está na gama [0, 1023].

Exercício 8

Resolva a alínea a) do exercício anterior escrevendo o respetivo programa em linguagem *assembly*.