

## Arquitetura de Computadores II

### Exercícios adicionais de preparação para o exame prático<sup>1</sup>

#### Exercício 1:

- a) Escreva um programa em linguagem *assembly* que, em ciclo infinito, faça a leitura dos 4 bits do porto de entrada (RB3 a RB0) e escreva essa combinação no porto de saída (RE3 a RE0).
- b) Altere o programa que escreveu na alínea anterior de modo a que o valor do porto de entrada RB3-RB0 seja escrito por ordem inversa no porto de saída, i.e., RB3 -> RE0, ..., RB0 -> RE3.

#### Exercício 2:

Escreva um programa em *assembly* que execute, em ciclo infinito, as seguintes tarefas:

Quando o utilizador carrega numa tecla do PC:

- na tecla 0, acenda apenas o LED ligado ao porto RE0;
- na tecla 1, acenda apenas o LED ligado ao porto RE1;
- na tecla 2, acenda apenas o LED ligado ao porto RE2;
- na tecla 3, acenda apenas o LED ligado ao porto RE3;
- noutra tecla qualquer, os 4 leds devem acender e permanecer ligados durante 1 segundo.

#### Exercício 3:

Repita os exercícios anteriores utilizando agora linguagem C.

#### 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 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 circular, módulo 100, com uma frequência de atualização de contagem de 10 Hz. O valor do contador deve ser impresso sempre na mesma linha do ecrã, formatado com 2 dígitos (nota: para começar a impressão sempre na mesma linha use o carater '\r').

- b) Faça as alterações que achar necessárias de modo a que, para além da especificação anterior, quando o utilizador carrega nas teclas 0 a 4 a frequência de atualização da contagem seja modificada de acordo com a seguinte expressão:  $\text{freq} = 10 * (1 + \text{tecla\_carregada})$ .

Sempre que o utilizador carregar na tecla ENTER deverá ser impresso o valor atual da contagem, a seguir ao valor do contador (por exemplo: 76 >> 76).

Utilize o core timer do PIC32 para controlar a frequência de atualização do contador.

#### Exercício 6:

- a) Altere o programa que escreveu em 5b), e utilize o timer 3 para controlar a frequência de atualização do contador. A atualização do contador deve ser feita por interrupção.

- b) Acrescente o módulo de visualização de modo a mostrar nos dois dígitos o valor atual da contagem, quando é premida a tecla ENTER. A frequência de refrescamento dos displays deve ser 50 Hz.

---

<sup>1</sup> Os exercícios aqui apresentados não podem ser encarados como "exercícios tipo" para o exame. São apenas uma coleção de problemas que cobrem toda a matéria relevante, que visam, através da sua resolução atempada e bem suportada nos conhecimentos adquiridos, maximizar a possibilidade de o aluno conseguir realizar o exame prático sem grandes dificuldades.

### Exercício 7:

a) Escreva um programa em linguagem C para escrever no ecrã (1 vez/segundo) o valor de cada um dos bits lidos do porto de entrada. Por exemplo, se o porto de entrada tiver a combinação binária "1100", no ecrã deverá aparecer: RB3: 1, RB2: 1, RB1: 0, RB0: 0

b) Faça as alterações que achar convenientes de modo a que a taxa de atualização da informação escrita no ecrã seja dada pela seguinte expressão:

$$freq = 1 + \frac{VAL_{ADC}}{255} [Hz]$$

Tenha em consideração que a ADC que utiliza é uma ADC de 10 bits e, por esse motivo, o valor da conversão poderá variar entre 0 e 1023.

### Exercício 8:

Retome o exercício anterior. Faça as alterações que permitam processar o fim de conversão da ADC por interrupção e acrescente o módulo de visualização de modo a imprimir nos dois displays de 7 segmentos o valor da frequência (com uma casa decimal). A frequência de atualização dos displays deve ser 100 Hz, obtida por interrupção do timer 2.

### Exercício 9:

Retome o exercício anterior. Configure a UART1 com os seguintes parâmetros de comunicação: 1200 bps, 8 data bits, sem paridade, 1 stop bit. Escreva as funções de transmissão de um carater e de transmissão de uma *string* que lhe permitam substituir a *system call* que usou para imprimir o valor de cada um dos bits do porto de entrada.

Para o teste deste exercício não se esqueça de executar o "pterm" com os mesmos parâmetros de comunicação com que configurou a UART.

### Exercício 10:

Retome o exercício anterior e altere-o de modo a guardar num buffer circular os últimos 16 valores dos 4 bits do porto de entrada (RB3 a RB0). O buffer circular deve ser atualizado aquando do envio do valor do porto para o PC.

Altere a configuração da UART1, de modo a processar a chegada de um novo carater por interrupção. Quando for recebido o carater 'E' o valor das 16 posições do buffer circular deve ser enviado, por ordem cronológica de ocorrência, para o PC. Sugestão: transforme o valor armazenado em cada posição do buffer circular numa *string* binária e use a função de envio de uma *string* para fazer a transmissão do valor para o PC.

## Problemas com grau de dificuldade elevado

### Problema 1:

Num jogo de xadrez de rápidas os dois adversários dispõem de um tempo inicial em minutos e segundos que vão ter de gerir quando estão a pensar na jogada que devem fazer. Sempre que um dos jogadores conclui uma jogada carrega no botão do relógio e o tempo do adversário começa a contar (a ser gasto).

Neste problema vamos implementar um relógio de rápidas que conta em décimos de segundo. O tempo inicial é de 80 segundos. É representado por "00" nos *displays* de 7 segmentos e os 8 leds LED7 a LED0 a "1". Na contagem decrescente os *displays* contam em décimas de segundo e de cada vez que mudam de 00 para 99 apagam o LED mais à esquerda.

Existem 2 relógios, um para as Brancas e outro para as Pretas. O programa arranca com o relógio das Brancas a ser mostrado no conjunto LEDs+Display e a ser decrementado. A tecla B no teclado do PC indica que o jogador das Brancas fez a sua jogada. Uma vez pressionada, o relógio das Brancas para, o relógio das Pretas passa a ser mostrado e decrementado e no PC deve ser mostrada a mensagem "Jogador B a jogar".

Se algum dos relógios chegar a zero esse jogador perde. Se um jogador fizer xeque mate deve carregar no X e o jogo termina com a sua vitória. Não há empates.

Terminado o jogo deve ser enviada para o PC a mensagem "Vitória das Pretas" ou "Vitória das Brancas".

### Problema 2:

Desenvolva um programa na placa de testes que implemente um relógio, enviado para a porta série", com o formato "MM:SS:DD", em que MM são minutos, SS segundos e DD décimas de segundo.

a) Desenvolva o programa de envio, com o relógio iniciado a 00:00:00 após *reset*. O relógio deve contar a partir de uma interrupção gerada por um dos timers do microcontrolador.

b) O valor do relógio pode ser ajustado através do potenciômetro que está ligado à ADC. Se o valor lido pela ADC não estiver a ser alterado dentro de um intervalo de 10% do seu valor máximo (a ADC é de 10 bits, logo o valor máximo é 1023) o ajuste não é feito. Uma alteração de mais de 10% leva a que, durante 10 segundos, o valor do relógio possa ser inicializado nos minutos entre 00 e 99, de forma proporcional ao valor lido na ADC (0 – 00, 512 – 50, 1023 – 99 e restantes em proporção).

c) Passe a permitir que, na placa de teste, os parâmetros da porta série sejam configuráveis através dos *dip switches* da forma indicada abaixo. O número de bits não é programável devendo estar fixo a 8 bits. O relógio deve arrancar quando o DP1 é posto a "1"

- DP1: "0" em programação; "1" arranca o relógio
- DP2: "0" sem paridade; "1" com paridade
- DP3: "0" paridade par; "1" paridade ímpar
- DP4: cada transição de "0" para "1" faz alterar o baud rate o qual deve variar entre os seguintes valores, em sequência: 600, 1200, 2400, 4800, 9600, 19200, 57600 e 115200. A sinalização do baud rate é feita pela ativação de um dos 8 leds da placa, respetivamente LED7, ... LED0.