

# Sistemas Digitais I

## 1ª Repetição Escrita de 2003/04

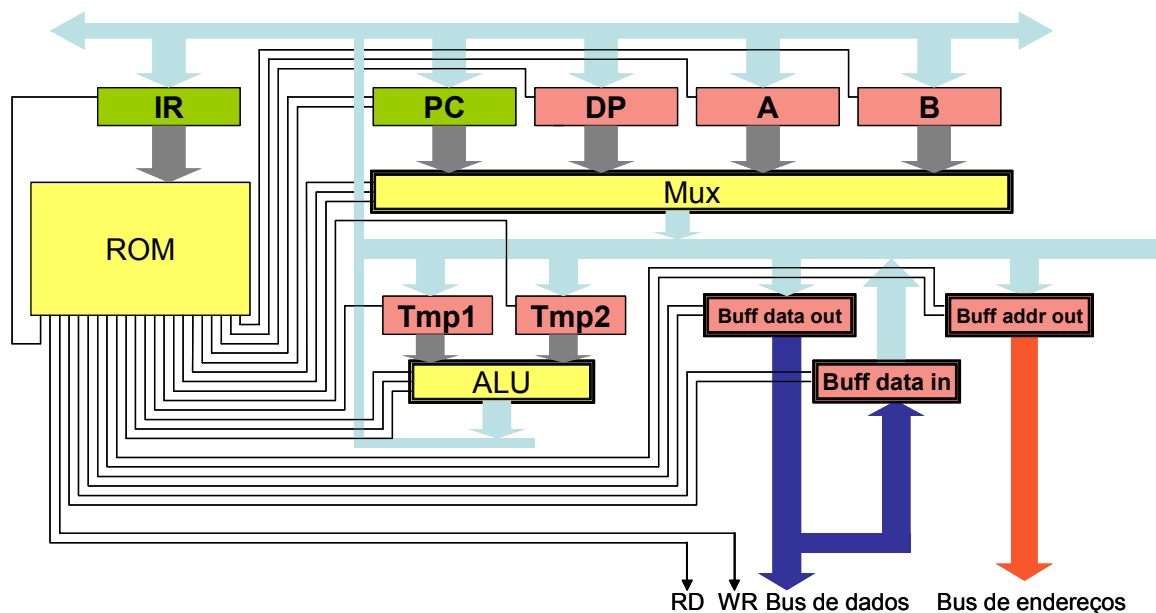
Leia atentamente o enunciado. Seja breve nas respostas, mas justifique-as convenientemente. **Por favor**, use letra legível ! Com os melhores desejos de bom trabalho nesta repetição ...

### I

As “canetas de memória flash” são cada vez mais populares. Explique qual o elemento que serve de base para esta tecnologia e como funciona, e explique as vantagens e desvantagens deste tipo de memória quando comparado com outros tipos de memórias dados nas aulas.

### II

A arquitectura do microprocessador com controlo por ROM dado nas aulas (o 1º) é extraordinariamente simples e limitado. Por isso, vamos desenhar um novo microprocessador, a que chamaremos EN2. Alguém propôs a arquitectura apresentada na figura. Por uma questão de simplicidade os sinais de relógio (CLK) não estão representados, mas são subentendidos nos dispositivos que os necessitam. Todos os buses representados têm 8 bits de largura, pelo que quer os dados, quer os endereços de memória têm 8 bits.



Os sinais de controlo que saem da ROM estão descritos abaixo, numerados da esquerda para a direita, e os dispositivos a que estão ligados são igualmente descritos.

- 1 - IR - Quando é 1, provoca um **PARALLEL LOAD** no **contador IR**, quando é 0, deixa que o contador IR passe ao número seguinte.
- 2 - RD - Sinal de controlo enviado para o BUS. Quando é 1 provoca uma operação de leitura à memória.
- 3 - WR - Sinal de controlo enviado para o BUS: Quando é 1 provoca uma operação de escrita à memória.
- 4 - “Buffer data in” output enable - Quando é 1, o registo buff\_data\_in põe o seu conteúdo à saída. Quando é 0, as saídas ficam em tri-state.

- 5 - “Buffer data in” load - Quando é 1, o registo buff\_data\_in carrega o que tiver à entrada. Quando é 0, as mantém o seu estado.
- 6 - “Buffer data out” output enable - Quando é 1, o registo buff\_data\_out põe o seu conteúdo à saída. Quando é 0, as saídas ficam em tri-state.
- 7 - “Buffer data out” load - Quando é 1, o registo buff\_data\_out carrega o que tiver à entrada. Quando é 0, as mantém o seu estado.
- 8 - “Buffer addr out” output enable - Quando é 1, o registo buff\_addr\_out põe o seu conteúdo à saída. Quando é 0, as saídas ficam em tri-state.
- 9 - “Buffer addr out” load - Quando é 1, o registo buff\_addr\_out carrega o que tiver à entrada. Quando é 0, as mantém o seu estado.
- 10- ALU output enable - Quando é 1, a ALU põe o seu conteúdo à saída. Quando é 0, as saídas ficam em tri-state.
- 11- ALU S<sub>0</sub> - Sinal que em conjunto com ALU S<sub>1</sub> controla a operação efectuada pela ALU. Com S<sub>1</sub>S<sub>0</sub> = “00” a ALU faz uma soma de TMP1 com TMP2, com “01” faz TMP1-TMP2, com “10” faz a operação AND entre os operandos, e com 11 faz OR.
- 12- ALU S<sub>1</sub> - Outro sinal de controlo da ALU-
- 13- TMP1 load - Carrega o registo TMP1.
- 14- TMP2 load - Carrega o registo TMP2.
- 15- MUX output enable - Quando é 1, o MUX põe um dado à saída. Quando é 0, as saídas ficam em tri-state.
- 16- MUX S<sub>0</sub> - Sinal que em conjunto com MUX S<sub>1</sub> faz a selecção do que o MUX põe à saída. S<sub>1</sub>S<sub>0</sub>=”00” selecciona PC, “01” selecciona DP, “10” selecciona A, e “11” selecciona B.
- 17- MUX S<sub>1</sub> - Outro sinal de controlo do MUX.
- 18 – PC load - Faz um PARALLEL LOAD ao contador PC.
- 19- PC increment - Incrementa o contador PC.
- 20- DP load - Carrega o registo DP.
- 21- A load - Carrega o registo A.
- 22- B load - Carrega o registo B.

**II.1 )** Quanta memória pode ser acedida por este microprocessador ?

**II.2)** Escreva o microcódigo para uma instrução máquina chamada “JMP addr” que faz com que o microprocessador salte para o endereço indicado.

**II.3)** Escreva o microcódigo para uma instrução máquina chamada “ADD A,B” que soma o registo A com o registo B, deixando o resultado no registo A.

### III

Escreva uma rotina em assembler de 8085 que faça a operação de divisão entre dois números de 8 bits. A rotina deverá receber no stack os dois números, estando o numerador no byte mais significativo e o denominador no byte menos significativo. O resultado deverá ser devolvido no acumulador. Pode usar o algoritmo que quiser, mas sugere-se que use divisões sucessivas.

Boa sorte !

