

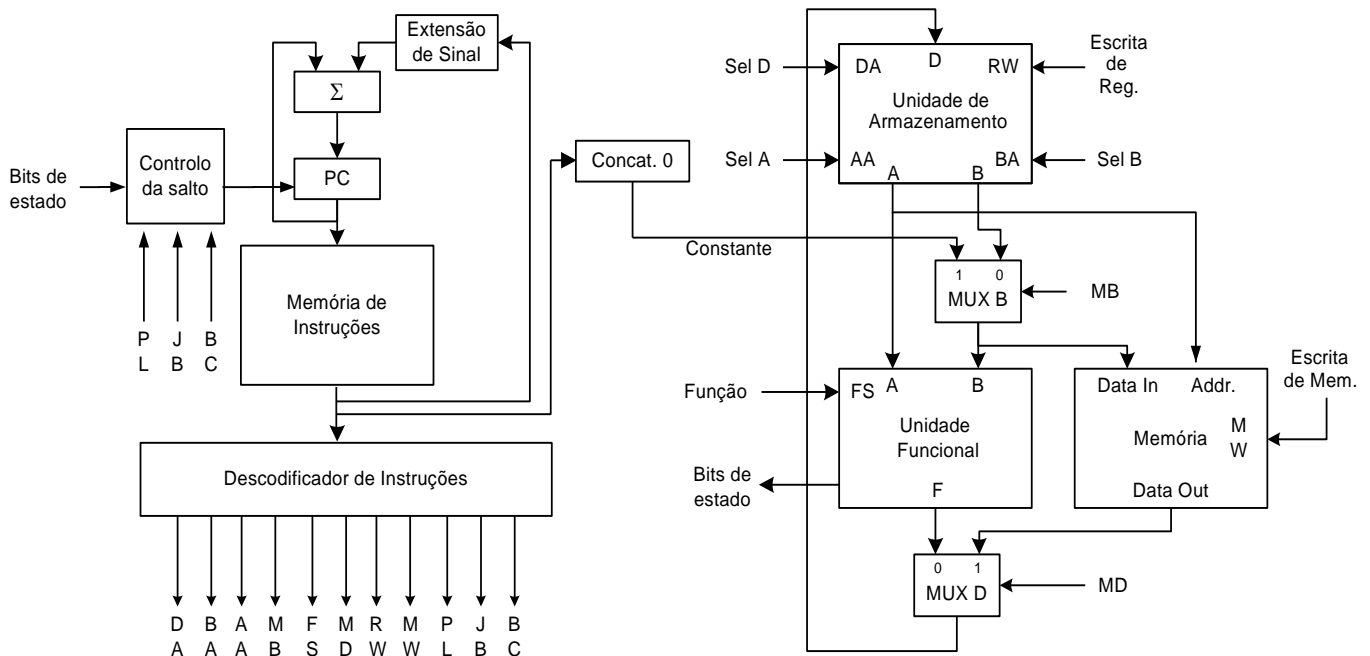
# Arquitectura de Computadores

## 2ª Série de Problemas

### Unidade de Controlo

1. Pretende-se projectar uma **unidade de controlo (UC) de ciclo único**, que actuará sobre o conjunto unidade processamento (UP)/memória ilustrado na figura. O conjunto de instruções será restringido às seguintes categorias:

1. Leitura/escrita entre memória e registos: Duas instruções (LOAD e STORE).
2. Operação de transferência, soma ou subtracção da UF sobre um registo e um operando constante. Três instruções (MOVI, ADDI, SUBI). Nas instruções esse operando constante será representado com 3 bits.
3. Operação da UF sobre registos. Oito instruções uma para cada função da unidade funcional.
4. Salto condicional (considerando os bits de estado N e Z ou o seu complemento) ou incondicional (J, JZ, JN, JNN, JNZ). Nas instruções o offset do salto será representado com 6 bits.



A UP é constituída pelas unidades de armazenamento (UA) e funcional (UF). A UA possui um conjunto de 8 registos de 4 bits, R0 a R7, que podem ser livremente utilizados como operandos para a UF. A UF caracteriza-se pelo conjunto de instruções apresentado na tabela, gerando os bits de estado Z, N (activos quando o resultado da operação é zero ou negativo, respectivamente) e C (activo quando existe *carry*).

FS	Função	Descrição	Mnemónica
000	$F = B$	Transferência	MOVB
001	$F = A + 1$	Incremento	INC
010	$F = A - 1$	Decremento	DEC
011	$F = A + B$	Adição	ADD
100	$F = A - B$	Subtracção	SUB
101	$F = \overline{A}$	Complemento	NOT
110	$F = A \wedge B$	AND	AND
111	$F = A \vee B$	OR	OR

Na unidade de controlo, o bloco de controlo de salto, tem três entradas, PL, JB e BC. PL activo indica que foi descodificada uma instrução de controlo de salto, JB indica que o salto é condicional e BC indica a condição de salto, nomeadamente as seguintes correspondências: 00-JZ, 01-JN, 10-JNN, 11-JNZ.

- Qual a utilidade dos blocos “extensão de sinal” e “concatenação 0”?
- Descreva a estrutura de cada um dos tipos de instrução indicando os sinais que devem ser aplicados directamente (sem lógica) à unidade de processamento ou à unidade de controlo. Escolha a dimensão do Opcode e a largura da instrução.
- Estabeleça o formato detalhado das instruções e determine as expressões booleanas para geração das linhas de controlo para a UP e UC a partir destas (descodificador de instruções). Tenha em atenção que os códigos de operação devem ser atribuídos de modo a simplificar a lógica de descodificação de instruções. Atribua a cada instrução uma mnemónica.
- Considere que pretende implementar o descodificador de instruções com recurso a uma memória. Indique: (1) o conteúdo da memória para uma implementação tendo como linhas de endereçamento o opcode das instruções; (2) a dimensão mínima da memória e o seu conteúdo nesse caso.
- Realize a síntese dos blocos de controlo de salto e de actualização do PC.
- Indique como deveria alterar a arquitectura apresentada e/ou quais os sinais de controlo para permitir a execução das seguintes operações:

$$RD < K1 + K2$$

$$RD < M(RA) + RB$$

$$RD < RA+1, M(RA) < RB \quad (\text{estas em simultâneo})$$

- Utilizando as mnemónicas definidas em c) escreva um programa que determine o maior número de uma tabela com N palavras. O endereço da tabela é passado em R0 e o primeiro elemento da tabela contém o número de elementos que se seguem, ou seja, N. O resultado deve ser colocado em R3.

- Escreva as instruções em binário que permitem executar as seguintes operações:

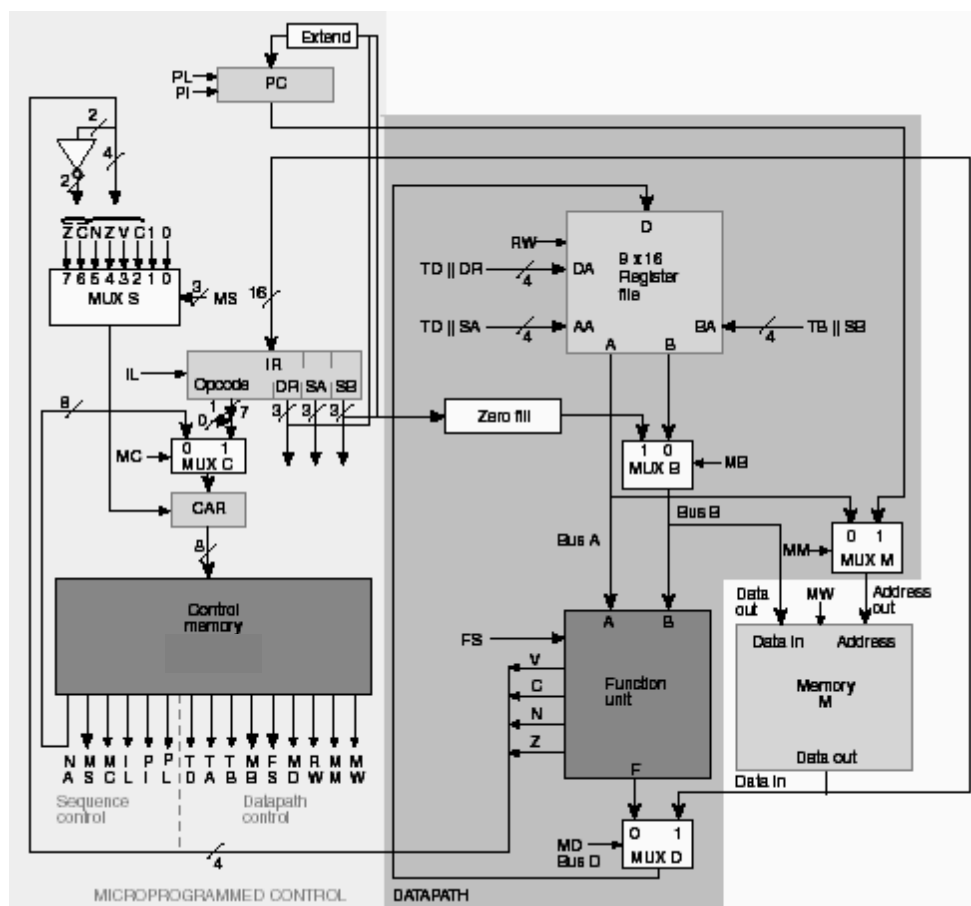
$$R_1 \leftarrow R_1 \vee R_2$$

$$R_1 \leftarrow M[R_4]$$

$$R_3 \leftarrow R_1 + 3$$

$$\text{se } (R1 = 0) : PC \leftarrow PC + 1, \text{ se } (R1 \neq 0) : PC \leftarrow PC + 20$$

2. Considere o computador microprogramado de **ciclo múltiplo** representado na figura.



Código	Função	Descrição	Código	Função	Descrição
000	$F = A$	Transferência	100	$F = \overline{A + B}$	NOR
001	$F = A + 1$	Incremento	101	$F = \overline{AB}$	NAND
010	$F = A + B$	Adição	110	$F = shl\ A$	deslocamento esquerda
011	$F = A - 1$	Decremento	111	$F = shr\ A$	deslocamento direita

A unidade de armazenamento (UA) é constituída por 8 registos de uso geral (R0 a R7), seleccionados por campos de 3 bits (SA, SB e DR) que fazem parte de cada instrução a executar. Existe um nono registo auxiliar, designado por T, que pode ser lido ou escrito durante a execução dos microprogramas. O seu conteúdo pode ser colocado nas saídas A e/ou B da UA activando as linhas TA e TB, respectivamente, independentemente dos valores presentes nas entradas de selecção SA e SB. Da mesma forma, este registo é seleccionado para escrita por activação de TD independentemente do valor de DR.

A unidade funcional (UF) caracteriza-se pelo conjunto de instruções apresentado na tabela, gerando os bits de estado Z e N (quando o resultado da operação é zero ou negativo, respectivamente), C (carry) e V (overflow envolvendo números em complemento para dois). Estes 4 bits são internamente armazenados num registo que é actualizado em todas as operações da UF, excepto transferência, e cuja saída está ligada ao MUX D da unidade de controlo (UC).

Dentro da UC, tanto o *program counter* (PC) como o *control address register* (CAR) são contadores ascendentes com carregamento paralelo. O carregamento do PC é controlado pela linha PL existindo adicionalmente uma entrada PI para *enable* de contagem. O carregamento do CAR é controlado pela saída do MUX S.

- a) Qual a finalidade do registo IR e do multiplexer MUX M?
- b) Pretende desenvolver-se um microprograma que implemente um salto condicional quando  $Z=0$ , com endereço absoluto especificado na instrução (concatenação dos campos DR||SA||SB). Desenhe o fluxograma ASM e escreva o microprograma completo, incluindo as microinstruções relativas à aquisição e à execução da instrução, assumindo que o opcode é 0000101. Tenha em atenção que as flags devem ser preservadas (porquê?). Quantos ciclos de relógio são necessários para executar esta instrução?
- c) Repita a alínea anterior para a instrução  $M[R[SA]] \leftarrow M[R[SA]] + R[SB]$ , assumindo que o opcode é 0001011.
3. Considere uma arquitectura semelhante à anteriormente apresentada, mas onde o PC é agora um registo incluído na UA, contendo esta também um novo registo auxiliar SP (*stack pointer*). O acesso para leitura ou escrita de T, PC e SP especifica-se pelos valores de TD, TA e TB, que passam a ser *buses* de 2 linhas com a seguinte codificação 0:R0...R7, 1:T, 2:PC, 3:SP. Nesta nova arquitectura as linhas PL e PI deixam de existir mas é adicionada uma linha FL que controla o carregamento do registo onde são armazenados os bits de estado. Admita que SP aponta para o último valor guardado na pilha e que esta cresce dos endereços mais baixos para os mais altos.
- a) Escreva um microprograma que permita chamar uma subrotina (CALL), com endereço relativo (deslocamento) especificado no campo SB da instrução. Assuma que o opcode é 0000100
- b) Escreva o microprograma correspondente à instrução RET, assumindo que o opcode é 0001000.
- c) Escreva o microprograma correspondente à instrução POP R[DR], assumindo que o opcode é 0001100.
4. Suponha que, num dado momento, se têm os seguintes valores para alguns dos registos internos e posições de memória de um computador:

Registos				Memória			
Reg	(Reg)	Reg	(Reg)	End	(End)	End	(End)
R0	33	PC	15	20H	21	53H	52
R1	10	SP	35	21H	20	54H	42
R2	84			22H	92	55H	85
R3	51			23H	10	56H	83

Complete a seguinte tabela de acordo com os valores fornecidos, indicando na coluna **Resultado** quais são as alterações nos registos e/ou na memória e deixando em branco os campos não aplicáveis a cada instrução.

Admita que nas instruções de salto o endereço especificado é relativo.

Instrução	Modo endereçamento	Endereço Efectivo	Resultado
MOV R3,10			
XCH R1,(R2)			
JMP 34			
RET			
LD R1,(35)			
INC ((86))			
ADD R0,R1,R2			
JMP R1			
CALL 32			
POP R1			