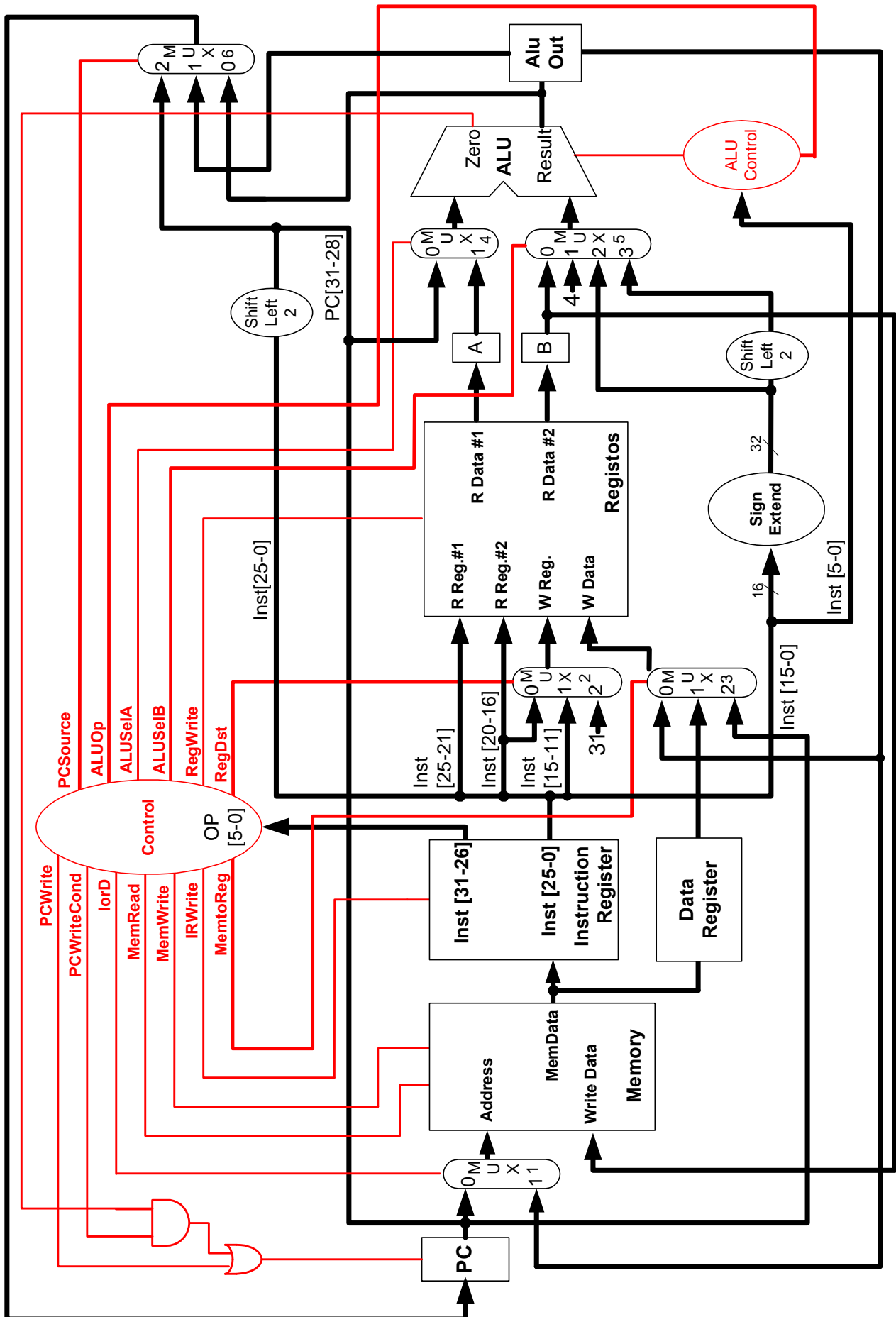


PARTE I (sem consulta)

NOTA: Não é permitida a presença de calculadoras e telemóveis na sala

1. **NOTA:** Use no máximo 40 palavras para responder a cada uma das 5 alíneas seguintes:
 - a) Diga o que distingue uma excepção duma interrupção.
 - b) Numa arquitectura *pipelined*, qual o tipo de *hazards* que se pretendem resolver com técnicas de predição? Justifique minimamente a resposta.
 - c) Indique os aspectos fundamentais que caracterizam o modelo hierárquico dos sistemas de memória dos computadores actuais.
 - d) Descreva de forma sucinta a tarefa executada pela instrução do MIPS “lui \$1, 0x2349”.
 - e) Diga o que entende por estratégia *callee saved*.
2. Desenhe o diagrama de blocos de uma arquitectura de 32 bits para multiplicação de inteiros. Identifique os elementos operativos, de estado e de controlo, e a dimensão dos barramentos.
3. Considere que o conteúdo dos registos \$f2 e \$f4 é respectivamente:
\$f2 = 0 00010010 000100000000000000000000
\$f4 = 1 01101100 001100000000000000000000
 - a) Obtenha a representação das quantidades armazenadas naqueles registos na forma $\pm 0, m * 2^{\text{exp}}$, em que “m” e “exp” estão em decimal, sabendo que estão codificadas segundo a norma IEEE 754.
 - b) Determine o resultado da instrução **mul.s \$f12, \$f2, \$f4**, realizando, em binário, os passos necessários à sua obtenção pela FPU. Explícite, em binário, o conteúdo do registo \$f12 após a execução da instrução.
4. Considere, na **figura 2** fornecida em anexo, o trecho de código *Assembly* ali apresentado. Admita que o valor presente em **\$PC** corresponde ao *label* “**fc1:**”, que nesse instante o conteúdo dos registos é o indicado, e que vai iniciar-se o “*instruction fetch*” da próxima instrução. Considere ainda o *datapath* e a unidade de controlo fornecido na próxima página, no pressuposto de que corresponde a uma implementação simplificada -ciclo sem *pipelining*. Note que este *datapath* é ligeiramente diferente do estudado nas aulas.
 - a) Escreva, nas duas colunas do lado direito da fig.2, em hexadecimal, o endereço em que se encontra armazenada cada uma das instruções do código fornecido, e o respectivo conteúdo, sabendo que a primeira linha corresponde à instrução presente no *label* “**fc1:**”.
 - b) Preencha a tabela fornecida em anexo com o nome de cada uma das fases de execução da **0x0040004C**, e com o valor que tomam, em cada uma delas, os sinais do *datapath* e os sinais de controlo ali indicados. Admita que o valor lógico “1” corresponde ao estado activo. Não se esqueça de preencher o cabeçalho (nome, curso e N.M).
 - c) Sabendo que a frequência do relógio do CPU é de 1 GHz, determine o tempo total que demora a executar o código fornecido, desde o instante inicial do *instruction fetch* da instrução presente em “**fc1:**” até ao momento em que vai iniciar-se o *instruction fetch* da instrução presente em “**next:**”. Justifique adequadamente a sua resposta.
 - d) A versão alterada do *datapath* fornecido permite a execução de uma instrução que não era suportada pelo *datapath* original estudado nas aulas. Determine qual é essa instrução, justificando adequadamente a sua resposta.

Cotações: 1a) a 1e) – 0,6; 2 – 2,0; 3a) – 1; 3b) – 1,5; 4a) – 1,0; 4b) – 1,5; 4c) – 1,0; 4d) – 1,0



Nome: _____

Curso: _____ N° Mecanográfico: _____

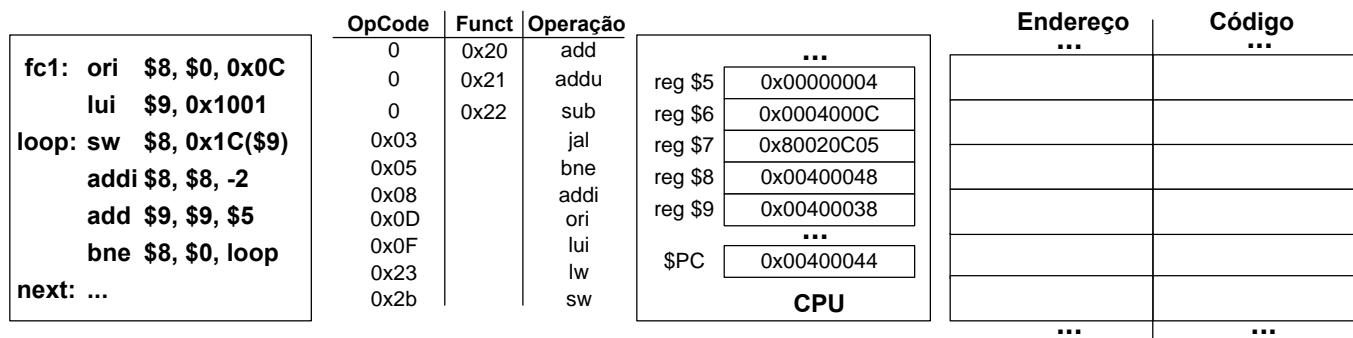


Figura 2 (Problema 4)

	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5
Nome da fase					

Datapath					
A					
B					
Data Register					
ALU Result					
ALU Out					
ALU Zero					

Controlo					
PCWrite					
PCWriteCond					
RegWrite					
MemRead					
MemWrite					
ALUOp					
IRWrite					
PCSource					
ALUSelA					
ALUSelB					
IorD					
MemtoReg					
RegDst					

Arquitectura de Computadores I

Exame

Parte Prática

07/01/2004

NOTE BEM: Leia atentamente todas as questões, comente o código usando, preferencialmente, a linguagem C e respeite a convenção de passagem de parâmetros e salvaguarda de registos. Respeite rigorosamente os aspectos estruturais e a sequência de instruções indicadas no código original fornecido, bem como as indicações sobre quais os registos a usar para cada variável.

O programa que se segue lê um ficheiro que contém as notas do 3º mini-teste de ACI. Este programa calcula, e apresenta na consola, o histograma e a média das notas.

Considere que as funções envolvidas têm os seguintes protótipos:

```
int    file_open(char *filename);
void   file_rewind(int f_id);
void   file_close(int f_id);

void   build_histogram(int f_id, int *histogram);
float  calc_average(int f_id);
void   print_histogram(int *histogram);
```

e que a função main tem a seguinte codificação em C:

```
int main(int argc, char *argv[])
{
    int    f_id;                /* reside em $s0 */
    float  average;             /* reside em $f20 */
    static int histogram[21];

    if(argc != 1) {
        print_string("ERRO: parâmetros inválidos"); /* syscall */
        return 1;
    }

    f_id = file_open(argv[0]);
    if(f_id == -1) {
        print_string("ERRO ao abrir o ficheiro");
        return 2;
    }

    build_histogram(f_id, histogram);

    file_rewind(f_id);          /* leitura do ficheiro volta ao início */
    average = calc_average(f_id);

    print_histogram(histogram);

    print_string("Média = ");
    print_float(average);

    file_close(f_id);
    return 0;
}
```

a) Codifique em *assembly* do MIPS a função main.

b) Considere que a função `file_read` tem o seguinte protótipo:

```
unsigned int file_read(int f_id);
```

Esta função lê um inteiro do ficheiro `f_id`. Retorna o inteiro lido ou o valor `0xFFFFFFFF` no caso de ter sido atingido o fim do ficheiro e avança a leitura para o inteiro seguinte.

Traduza para *assembly* do MIPS a função `build_histogram`.

```
void build_histogram(int f_id, int *histogram)
{
    unsigned int nota;

    for( nota = 0; nota <= 20; nota++ ) {
        histogram[nota]=0;
    }
    nota = file_read(f_id);
    while( nota != 0xFFFFFFFF ) {
        histogram[nota] = histogram[nota] + 1;
        nota = file_read(f_id);
    }
}
```

c) Escreva em C e codifique em *assembly* do MIPS a função `calc_average` que determina, e devolve, a média das notas armazenadas no ficheiro `f_id`. Considere que o protótipo da função é:

```
float calc_average(int f_id);
```

Tenha em atenção que o valor de retorno da função é do tipo `float` e deve ser devolvido através do registo `$f0`. Considere ainda a utilização da função `file_read` descrita em b).