

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS
GERAIS –
CEFET/MG**

Celso França Neto (20203018570)
Diogo Emanuel Antunes Santos (20213002091)

**AULA 5 - Projeto de um Processador: planejamento do conjunto de
instruções.**

Laboratório de Arquitetura e Organização de Computadores

**BELO HORIZONTE
2022**

1) Explique o que o programa embarcado deverá fazer.

O nosso programa realiza o cálculo do fatorial de um número salvo na memória, a lógica é bem simples, o programa pega esse número dado, e calcula dentro de uma estrutura de repetição, o número multiplicado por todos seus antecessores, ou seja, executa a fórmula de fatorial. Quando a multiplicação é pelo valor 1 o programa se encerra.

2) Apresente a lista de instruções suportadas pelo seu processador.

O nosso processador suporta as instruções:

“lw”, “defi”, “sw”, “beq”, “mul”, “subi”, “j” & “encerra”.

3) Explique a operação realizada por cada uma das instruções.

4) Mostre a representação (sintaxe) em assembly de cada instrução.

5) Indique o formato binário de cada uma das instruções, apontando o tamanho (em número de bits) e a função de cada campo das instruções.

- defi, (define um valor inteiro para um registrador).

Opcode	Destino	Valor inteiro
000	1	0001

A instrução em assembly:

defi reg, inteiro

- beq, (compara 2 registradores, se forem iguais salta para determinada operação).

Opcode	Registrador 1	Registrador 2	Endereço do salto
001	0	1	101

A instrução em assembly:

beq reg1, reg2, operação

- sw, (armazena o valor de um registrador em um local na memória principal).

Opcode	Origem	Endereço de memória
011	1	0010

A instrução em assembly:

sw reg, memória

- lw, (acessa um valor armazenado na memória e passa pro registrador).

Opcode	Destino	Endereço de memória
0 1 0	1	0 0 1 0

A instrução em assembly:

lw reg, memória

- mul, (multiplica dois registradores e salva no primeiro da multiplicação o resultado).

Opcode	Registra dor 1 e destino	Registr ador 2	Completando 8 bits
1 0 0	1	0	0 0 0

A instrução em assembly:

mul reg, reg.

- subi, (efetua a subtração de um registrador por um valor inteiro, salvando nesse registrador o resultado).

Opcode	Registra dor e Destino	Valor inteiro
1 0 1	0	0 0 0 1

A instrução em assembly:

subi reg, inteiro

- j, (realiza uma operação jump, leva o retorno operação que recomeça o loop).

Opcode	Endereço de retorno
1 1 0	0 0 0 1 1

A instrução em assembly:

j operação

- encerra (encerra o programa quando o loop finaliza).

Opcode			Completando 8 bits				
1	1	1	0	0	0	0	0

A instrução em assembly:

encerra

6) Justifique todas as suas decisões de projeto.

Os motivos que nos levaram a escolher essa quantidade de bits destinada a diferenciar as equações foi a necessidade de usarmos 8 instruções em nosso programa, com 2 bits não seria possível, essa quantidade de instruções só é possível a partir de 3 bits. Sendo essa adaptação ao programa o motivo da nossa escolha.

Com esse valor de operações definido, escolhemos a quantidade de registradores para 2, pois dessa maneira não nos prejudicamos na elaboração do software e nem do hardware, sendo assim possível salvar os valores dos registradores nos endereços de memória corretos, acessá-los e realizar deslocamentos com os 3 bits que sobraram.

Assim, essas questões aritméticas foram os motivos de escolhermos essa quantidade de bits para as determinadas instruções, registradores, cálculos absolutos e deslocamentos.