

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

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

**AULA 8: definição dos sinais de controle.**

**Laboratório de Arquitetura e Organização de Computadores**

**BELO HORIZONTE  
2022**

**1) Liste os sinais de controle existentes no seu processador e explique o funcionamento de cada um deles. Você deve informar a utilidade do sinal e o impacto no caminho de dados para cada possível valor que o sinal possa assumir.**

**EscreveReg** — É um sinal de 1 bit, quando habilitado (valor = 1), serve para permitir que o resultado da operação da ULA, o valor imediato ou dado do banco de memória seja escrito no registrador de destino (Escreve dado Registradores). É utilizado para impedir que o espaço seja utilizado em operações como *j*, *beq*, *sw* ou *encerra*. Quando desabilitado (valor = 0), serve para desabilitar a escrita no "registrador escrito" do banco de registradores.

**ULASrc** — É um sinal de 1 bit, quando habilitado (valor = 1), serve para passar o valor lido do registrador *Reg 2* (bit 3 da instrução) para a ULA, é utilizado na instrução *mul*, que faz operação entre dois registradores. Quando desabilitado (valor = 0), serve para passar o valor vindo do imediato da instrução (3 - 0 da instrução) para a ULA, é utilizado na instrução *subi*, que é realizado entre o registrador e um imediato.

**Defi** — É um sinal de 1 bit, quando habilitado (valor = 1), serve para permitir que durante a instrução *defi* o valor enviado para ser escrito no registrador no banco de registradores seja o imediato. É utilizada para o adequado funcionamento da instrução *defi*. Quando desabilitado (bit = 0), serve para passar ao registrador a ser escrito o valor do banco de memória ou o resultado da ULA, invés do imediato da função Defi, ou seja, indica que a função em andamento não é a Defi.

**MemtoReg** — É um sinal de 1 bit, quando habilitado (valor = 1), serve para permitir que um dado da memória seja enviado para ser escrito no registrador, decidindo, num MUX, entre a operação na ULA e o dado da memória, pelo dado da memória. É utilizado para o adequado funcionamento da instrução *lw*. Quando desabilitado (valor = 0), indica que o valor a ser enviado para ser escrito no registrador é o resultado da operação feita na ULA, útil para a instrução *subi* e *mul*.

**OpULA** — É utilizado quando é necessário realizar operações ULA, como no MIPS. É um sinal de controle de 1 bit, pois indica dentre as operações de subtração ou multiplicação, qual deverá ser realizada. Não há impacto na estrutura do *data path*.

**Branch** — É um sinal de 1 bit, quando habilitado (valor = 1), indica a instrução de desvio *beq*. Serve como uma das duas entradas de uma porta AND, mostrando que a instrução *beq* está em andamento, quando a outra entrada da AND também é 1, ou seja, quando o *Zero* da ULA for verdadeiro, ele passa o sinal que indica que o salto condicional deve ser realizado. Quando desabilitado (valor = 0), mostra que a instrução *beq* não está em andamento, logo, não deve realizar o salto condicional.

**EscreveMem** — É um sinal de 1 bit, quando habilitado (valor = 1), serve para permitir que durante a operação *sw* a cópia do valor do registrador seja salva na posição indicada na memória. Quando desabilitado (valor = 0), serve para proibir a escrita na memória, que é utilizado quando a operação em andamento não é *sw*.

**LerMem** — É um sinal de 1 bit, quando habilitado (valor = 1), serve para permitir a leitura da posição indicada na memória. É utilizado na instrução *lw*, para que seja possível escrever o valor da memória no registrador. Quando desabilitado (valor = 0), serve para proibir a leitura da memória e é utilizado para impedir a leitura de dados aleatórios da memória durante a execução de outra instrução que não *lw*.

**Jump** — É um sinal de 1 bit, quando habilitado (valor = 1), serve para indicar para um MUX escolher o imediato da instrução para escrever em PC, e não PC + 1 ou valor de salto de *beq*. É utilizado para instrução de salto incondicional *jump*. Quando desabilitado (valor = 0), indica que a instrução *jump* não está em andamento, logo, não irá selecionar o valor imediato da instrução para o salto, e sim o valor PC + 1 ou valor de salto do *beq*.

**Encerra** — Quando habilitado (valor = 1), serve para pausar a escrita em PC. É utilizado junto a outros sinais de controle para parar a escrita em todos os componentes e encerrar a execução do programa. Quando desabilitado (valor = 0), permite a escrita em PC.

2) Se você estiver implementando uma versão de processador uniciclo, para cada instrução suportada pelo seu nRisc, descreva os valores a serem assumidos por todos os sinais de controle (sugestão: use uma tabela).

A definição do sinal de controle será dada por:

*Sub*: 0

*Mul*: 1

Tabela da relação de *instruções* por *sinais de controle*:

	EscreveReg	Defi	EscreveMem	ULASrc	MemtoReg	Jump	LerMem	Branch	OpULA	Encerra
<i>def</i>	1	1	0	X	X	0	0	0	X	0
<i>beq</i>	0	X	0	1	X	0	0	1	0	0
<i>sw</i>	0	X	1	X	X	0	0	0	X	0
<i>lw</i>	1	0	0	X	1	0	1	0	X	0
<i>mul</i>	1	0	0	1	0	0	0	0	1	0
<i>subi</i>	1	0	0	0	0	0	0	0	0	0
<i>j</i>	0	X	0	X	X	1	0	0	X	0
<i>encerra</i>	0	X	0	X	X	X	0	X	X	1

