

# Unidade de Controle para instruções I e J

Kaike Carvalho, Eduardo Knabben Tiyo, Felipe Kenzo Suguimoto

*Universidade Tecnológica Federal do Paraná – UTFPR*

*COCIC – Coordenação do Curso de Bacharelado em Ciência da Computação*

*Campo Mourão, Paraná, Brasil*

kaikecarvalho@alunos.utfpr.edu.br

tiyo@alunos.utfpr.edu.br

felipekenzo@alunos.utfpr.edu.br

## Resumo

Para instruções do tipo I e J, a unidade de controle seleciona o caminho de dados apropriado para operações de leitura e escrita em registradores ou memória, além de gerenciar alterações no Program Counter (PC) e controlar a Unidade Lógica e Aritmética (ULA).

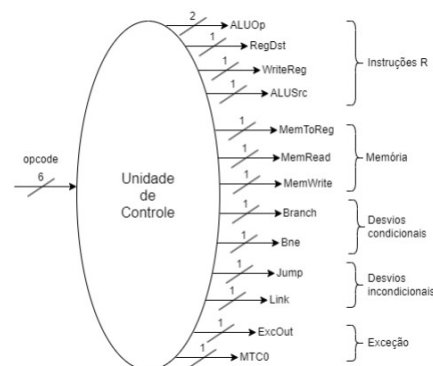
Em resumo, a unidade de controle desempenha um papel central na coordenação e direcionamento das operações do datapath, ajustando dinamicamente o fluxo de dados e as operações de acordo com os requisitos específicos das instruções processadas pelo processador.

## 1. Introdução

Neste relatório será abordado questões sobre a Unidade de Controle para as instruções I e J, mostrando em detalhes seus componentes e seu funcionamento.

## 2. Unidade de Controle Principal

A Figura 1 mostra um esquema de entradas e saídas da unidade de controle principal do Datapath. O campo OPCODE fornecido pelo decodificador de instruções é a única entrada da unidade de controle principal.



**Figura 1 – Entradas e saídas da unidade de controle principal.**

O comportamento das instruções especificadas são da seguinte maneira:

- **R:** Ativa o REGDST e WRITEREG
- **J:** Ativa o JUMP
- **JAL:** Ativa o WriteReg , JUMP e LINK
- **ADDI:** Ativa os sinais WriteReg e ALUSrc. O ALUOP seleciona a operação de soma na ULA.
- **SW:** Ativa os sinais MemWrite , que escreve o valor do registrador na memória e ALUSrc (segunda entrada da ULA é o imediato). O ALUOP seleciona a operação de soma para calcular o endereço da memória pelo imediato.
- **LW:** Seta os sinais MemToReg, que escreve o valor da memória em registrador, MemRead, WriteReg e ALUSrc (segunda entrada da ULA é o imediato). O ALUOP seleciona a operação de soma para calcular o endereço da memória pelo imediato.
- **BEQ:** Coloca como 1 o sinal Branch e informa em ALUOP que se trata de um branch para realizar subtração.
- **BNE:** os sinais Branch e BNE são ativados, além disso, é informado em ALUOP que se trata de um branch para realizar subtração.
- **EXCOUT:** Ativa o EXCOUT, indicando que o valor de PC voltará para a execução do programa principal.
- **MTC0:** Ativa o MTC0, que indica que a operação é de escrita no registrador da unidade de exceção.

RFE: Ativa o WriteReg e o RFE.

Segue abaixo uma tabela abaixo para especificar o OP CODE das respectivas operações:

OPERAÇÃO	OPCODE
R	000000
J	000010
JAL	000011
ADDI	001000
SW	101011
LW	100011
BEQ	000100

BNE	000101
EXCOUT	100000
MTC0	100001
RFE	100010

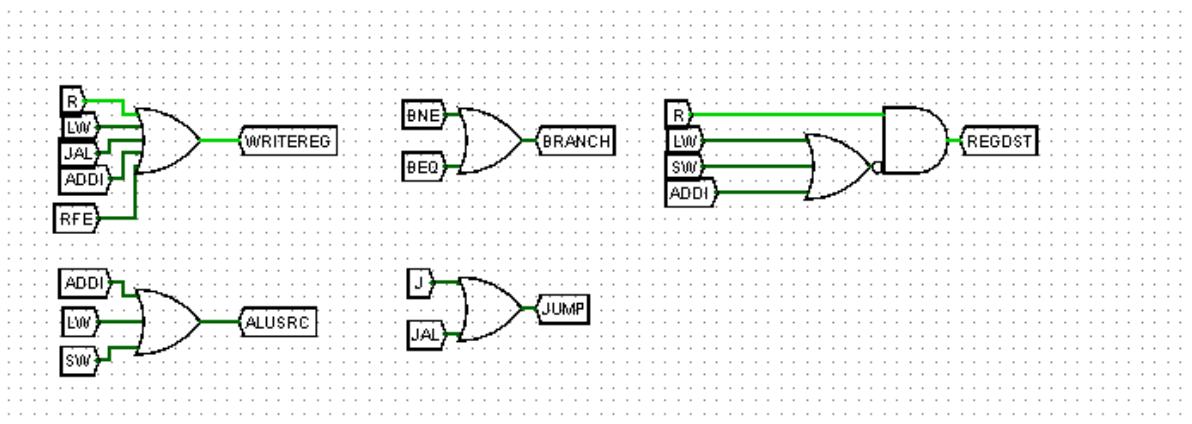


Figura 1.1

De acordo com a Figura 1.1, o WriteReg, ALUSrc, BRANCH, JUMP e REGDST serão ativados quando pelo menos uma operação acima for utilizada, assim, dando continuidade ao circuito.

### 3. Unidade de controle da ULA

A entrada FUNCT na UCA (unidade de controle da ULA) decidirá qual operação será realizada quando o OPCODE for 000000 (instrução R). Serão utilizados apenas os 4 bits mais significativos da funct para determinar a operação, por isso, na imagem 2 há dois distribuidores, os quais seleciona os bits e levarão para um MUX entre a FUNCT da instrução, o OPCODE 0100 (soma para o LW e SW), ou o OPCODE 0101 (subtração para BRANCH). A saída será determinada pelo aluOp, que se for 0 é instrução R, 01 é instrução I sem BRANCH, 10 é I com BRANCH. Caso a instrução seja JR (FUNCT 110000), o OPCODE teoricamente sairia 1100, mas não há essa operação, e para fazer o JR, foi utilizada a operação soma, para somar em PC o endereço guardado mais o valor 0. O circuito vê se FUNCT é 110000 e se aluOp for 00, então a instrução é JR. Portanto, em outro MUX, se a instrução for JR, ele determinará soma (0100) para o op da ULA. Caso contrário, seguirá normalmente com o resultado do primeiro MUX.

FUNCT	OPERAÇÃO
000000	AND

000100	OR
001000	NOR
001100	XOR
010000	ADD
010100	SUB
011000	SLL
011100	SLR
100000	SLT
110000	JR

#### 4. Unidade de controle de exceção

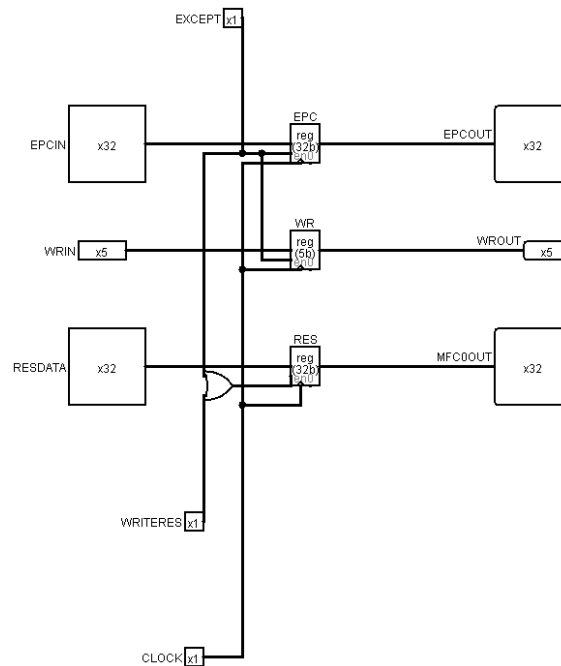
A exceção é uma mudança inesperada no fluxo de controle independentemente da origem, e pode ser tanto síncrona quanto assíncrona. Quando uma exceção ocorre, o controle é enviado para um programa chamado "exception handler", que resolverá o problema e posteriormente retornará no ponto onde o programa parou. O co-processador 0 (Unidade de tratamento de exceções) possui registradores internos que armazenam informações sobre o tratamento de exceções. O co-processador criado para este trabalho trata apenas os overflows aritméticos. Esse circuito possui dois registradores internos de 32 bits o EPC e res e um de 5 bits o WR. A unidade de controle de exceções possui 5 entradas e 3 saídas:

Entradas	Função
Except	Indica se a instrução atual causou uma exceção.
EPC_in	Salva o valor que deve ser gravado no registrador EPC, que contém o endereço da instrução causadora da exceção.

WR_in	Identificador do registrador que receberia o resultado incorreto gerado que será gravado no registrador interno WR.
RES_data	Valor que vai ser gravado no registrador interno “res”, que pode ter origem no resultado incorreto gerado na ULA ou no registrador lido no caso de instrução mtc0.
WriteRes	Habilita a escrita do valor res_data no registrador interno res.

Saídas	Função
EPC_out	Último valor gravado no registrador EPC.
WR_out	Último valor gravado no registrador WR.
MFC0_out	Último valor gravado no registrador RES.

A Figura 1.3 representa o circuito da unidade de controle da exceção, e ela funciona da seguinte maneira: O bit except é uma flag que permite a escrita nos registradores epc, wr e res ela é ativada caso ocorra overflow, o bit writeres é uma flag que permite a escrita do registrador res ela é ativada dependendo do valor de MTC0, o clock é ligado em todos os registradores internos e tem a função de atualizar os valores recebidos, o EPC\_in recebe o valor do registrador PC +4 e envia para o registrador interno EPC que por sua vez envia para a saída EPC\_out, essa função serve para salvar o endereço que contém o problema de exceção, o WR\_in recebe o valor de WR esse valor é enviado para o registrador interno WR que envia para a saída WR\_out e tem a função de identificar o registrador que receberia o resultado incorreto, o RES\_data recebe o valor que vai ser gravado no registrador interno res e será enviado para a saída MFC0\_OUT, caso o registrador res seja ativado pela flag except ocorreu algum erro de exceção e se ela for ativada pela flag writeres não ocorreu erro.

**Figura 1.3 - Circuito da Unidade de controle de Exceção**

## 5. Conclusões

Em resumo, a unidade de controle exerce uma função fundamental no desempenho do processador, assegurando a execução eficiente das instruções. Sua adaptabilidade a diferentes tipos de instruções e a flexibilidade para realizar diversas operações no datapath são características essenciais para o funcionamento do processador.

## 6. Referências

[1] Patterson, David A. Hennessy, John L. Organização e Projeto de Computadores. Disponível em: Minha Biblioteca, (5a. edição). Grupo GEN, 2017.