

MIPS DLX

MANUAL DE USO

Microsoft Office User

**ERIC FERNANDO OTOFUJI ABRANTES
HENRIQUE MUALEM MARTI
MARCO MOLITERNO PENA PIACENTINI**

INTENCIONALMENTE EM BRANCO

SUMÁRIO
CONTEÚDO

SEÇÃO 1 – GERAL

Contextualização.....	1-1
Convenção uso registradores.....	1-2
Endereçamento.....	1-2
Instruções.....	1-3
Limitações.....	1-3

SEÇÃO 2 – INST R

Contextualização.....	2-1
Estrutura da instrução.....	2-1
FD INST R.....	2-2

SEÇÃO 3 – INST I

Contextualização.....	3-1
Estrutura da instrução.....	3-1
FD INST I.....	3-2

SEÇÃO 4 – INST J

Contextualização.....	4-1
Estrutura da instrução.....	4-1
FD INST J.....	4-2

SEÇÃO 5 – ULA

Contextualização.....	5-1
FD ULA.....	5-1
CTR ULA	5-1

SEÇÃO 6 – PIPELINE

Ciclo único.....	6-1
------------------	-----

SEÇÃO 7 – FD

FD ciclo único.....	7-1
CTR FD.....	7-2

SEÇÃO 8 – EXTRA

Informações complementares.....	8-1
---------------------------------	-----

**SEÇÃO 1
GERAL**

CONTEÚDO

Contextualização.....	1-1
Convenção uso registradores.....	1-2
Endereçamento.....	1-3
Instruções.....	1-3
Limitações.....	1-3

CONTEXTUALIZAÇÃO

Este é um projeto acadêmico realizado na disciplina Design de Computadores para o curso de Engenharia de Computação do Insper Instituto de Ensino e Pesquisa.

AVISO

Não é destinado para uso comercial nem possui qualquer garantia de funcionamento fora de ambiente acadêmico.

Trata-se da implementação de um processador RISC de 32 bits compatível com o MIPS DLX (Stanford/Berkeley), ciclo único, baseado no MIPS 1 de domínio público após o prazo de queda de patente.

NOTA

Ciclo único na versão da entrega intermediária de 15 de novembro de 2021, apenas. Pretendemos implementar pipeline para a versão final futuramente.

Com base nas páginas internas da disciplina, elencamos as principais características do MIPS DLX.

1. Processador RISC de 32 bits;
2. Arquitetura do tipo Load/Store;
3. 32 registradores de uso geral;
4. Endereçamento de 2^{30} palavras de memória (2^{32} bytes = 4 GB);

5. O endereço de memória tem granularidade de um byte (endereços de 32 bits);
6. O acesso à memória é feito com endereços que diferem de 4 bytes (uma word – obrigatório);
7. Formato regular para as instruções;
8. Conjunto de instruções dividido em seis classes;
9. Não possui flags de estado;
10. Big Endian;
11. Não possui suporte de hardware para gerenciar pilhas;
12. Pipeline de cinco estágios; e
13. Interrupção.

AVISO

Mesmo quando não mencionado nas próximas seções, muito se foi baseado nas páginas internas da disciplina para a realização deste projeto, inclusive deste manual de uso.

Possui 32 registradores de uso geral. Por convenção, usa-se o seguinte padrão na programação.

CONVENÇÃO USO REGISTRADORES

REG	NOME	DESCRIÇÃO
0	\$zero	Sempre retorna zero
1	\$at	Reservado assembler
2-3	\$v0, \$v1	Retorno de sub-rotina
4-7	\$a0-\$a3	Argumentos
8-15	\$t0-\$t7	Temporários
16-23	\$s0-\$s7	Variáveis de sub-rotina
24-25	\$t8, \$t9	Temporários
26-27	\$k0, \$k1	Reservado trap
28	\$gp	Ponteiro global
29	\$sp	Ponteiro da pilha
30	\$s8 ou \$fp	Ponteiro da estrutura
31	\$ra	Endereço ret sub-rotina

ENDEREÇAMENTO

Restrito para instruções de leitura e escrita na memória e instruções de desvio. Pode ser direto imediato (destino de label definido no programa), indireto (valor de registrador) ou indexado (deslocamento offset em relação a endereço base).

INSTRUÇÕES

Instruções disponíveis no processador.

1. INST R	Soma	ADD
2. INST R	Subtração	SUB
3. INST R	E lógico	AND
4. INST R	OU lógico	OR
5. INST R	Seleção se menor que	SLT
6. INST I	Carrega palavra	LW
7. INST I	Armazena palavra	SW
8. INST I	Desvia se igual	BEQ
9. INST J	Salto incondicional	J

LIMITAÇÕES

Não possui NAND, NOR, SHIFT.

SEÇÃO 2

INST R

CONTEÚDO

Contextualização.....	2-1
Estrutura da instrução.....	2-1
FD INST R.....	2-2

CONTEXTUALIZAÇÃO

INST R é o nome dado às Instruções do Tipo R do MIPS DLX, que é um dos três tipos de instrução. Cada instrução é composta por uma palavra de 32 bits e é organizada de uma forma específica.

Este é o grupo das instruções lógicas e aritméticas que operam somente com os registradores e com o uso da ULA. O uso de registradores permite uma velocidade de execução maior do que se utilizasse a memória externa ao processador.

ESTRUTURA DA INSTRUÇÃO

Opcode	Rs	Rt	Rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

Opcode contém o código da instrução a ser executada. Rs é o endereço do registrador com o primeiro operando da instrução definida em opcode. Rt é o endereço do registrador com o segundo operando da instrução definida em opcode. Rd é o endereço do registrador de destino para o resultado da operação definida em opcode.

Shamt é o total deslocamento que é usado no MIPS original, mas que para este projeto não é utilizado porque não há SHIFT implementado, conforme mencionado na seção de limitações; por conveniência, mantivemos a estrutura original mesmo assim, apenas não usando seus bits.

Funct seleciona a operação que será aplicada aos operandos, pois o opcode só define que a instrução é do tipo R, mas nada definindo a respeito da função.

Nas instruções tipo R, o opcode é sempre o mesmo: zero. Sabendo que se trata deste tipo de instrução, o processador fará o encaminhamento no fluxo de dados para executar esse

tipo de operação, enviando para a ULA a função definida acima.

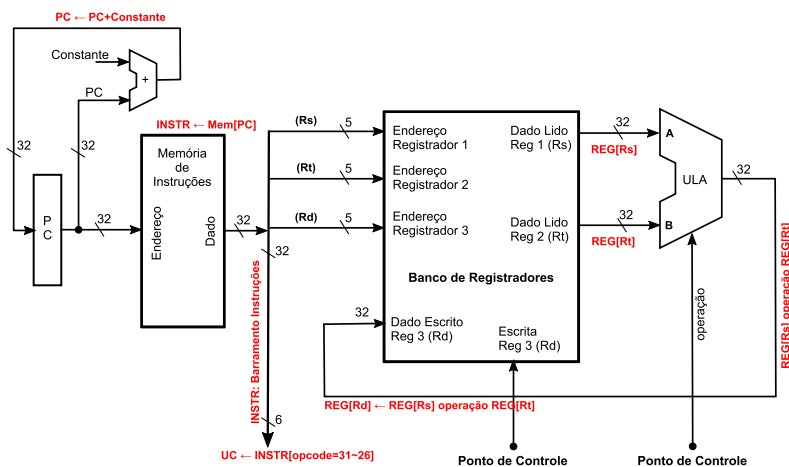
A sintaxe dessa instrução é do tipo indicado a seguir.

$$R[rd] = R[rs] \text{ OPS } R[rt]$$

As instruções implementadas deste grupo são as definidas na seção 1, em instruções, que possuem a marcação INST R.

FD INST R

O FD parcial usado pelas instruções do tipo R é apresentado a seguir. Uma visão completa do fluxo de dados do MIPS DLX pode ser vista na seção 7 – FD.



Fluxo de Dados para Instruções Aritméticas

SEÇÃO 3
INST I

CONTEÚDO

Contextualização.....	3-1
Estrutura da instrução.....	3-1
FD INST I.....	3-2

CONTEXTUALIZAÇÃO

INST I é o nome dado às Instruções do Tipo I do MIPS DLX, que é um dos três tipos de instrução. Cada instrução é composta por uma palavra de 32 bits e é organizada de uma forma específica.

Este é o grupo das instruções lógicas e aritméticas que operam com valores imediatos nos registradores. O uso de registradores permite uma velocidade de execução maior do que se utilizasse a memória externa ao processador.

ESTRUTURA DA INSTRUÇÃO

Abaixo, temos a estrutura geral de instruções do MIPS DLX.

Opcode	Rs	Rt	Rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

INST I agrupa os campos Rd, shamt e funct para formar o campo imediato, necessário ao funcionamento dessa instrução.

Opcode	Rs	Rt	Imediato
6 bits	5 bits	5 bits	16 bits ;

Opcode contém o código da instrução a ser executada. Rs é o endereço do registrador com o primeiro operando da instrução definida em opcode. Rt é o endereço do registrador com o segundo operando da instrução definida em opcode.

Imediato é o valor imediato de 16 bits, usado para INST I. Nas instruções tipo I, o opcode é variável, ao contrário de INST R.

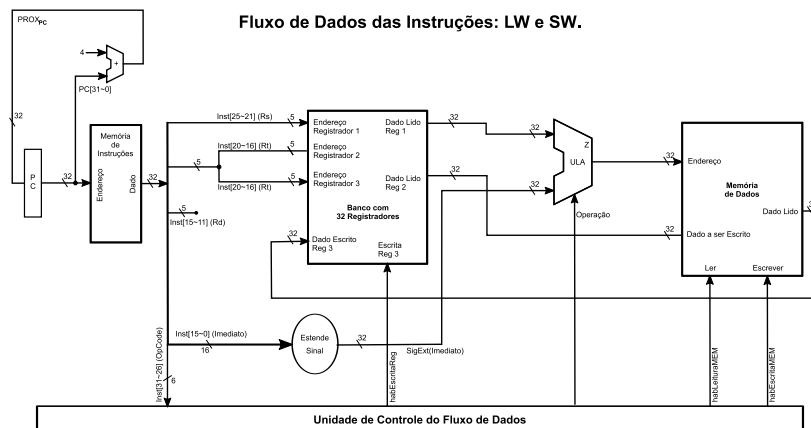
A sintaxe dessa instrução é do tipo indicado a seguir. Para fins de exemplo, estamos executando uma adição com imediato, mas não implementada.

ADDI rt, rs, IMM[16]

As instruções implementadas deste grupo são as definidas na seção 1, em instruções, que possuem a marcação INST I. São usadas para fazer operações lógicas e aritméticas com valores imediatos, escrever na memória, ler da memória e fazer desvios condicionais.

FD INST I

O FD parcial usado pelas instruções do tipo I é apresentado a seguir. Uma visão completa do fluxo de dados do MIPS DLX pode ser vista na seção 7 – FD.



SEÇÃO 4

INST J

CONTEÚDO

Contextualização.....	4-1
Estrutura da instrução.....	4-1
FD INST J.....	4-2

CONTEXTUALIZAÇÃO

INST J é o nome dado às Instruções do Tipo J do MIPS DLX, que é um dos três tipos de instrução. Cada instrução é composta por uma palavra de 32 bits e é organizada de uma forma específica.

Este é o grupo das instruções de jump incondicional.

ESTRUTURA DA INSTRUÇÃO

Abaixo, temos a estrutura geral de instruções do MIPS DLX.

Opcode	Rs	Rt	Rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

INST J agrupa os campos Rs, Rt, Rd, shamt e funct para formar o campo imediato de 26 bits que será utilizado para definir o endereço de destino do salto incondicional.

Opcode	Imediato
6 bits	26 bits

Opcode contém o código da instrução a ser executada. Imediato é o endereço de destino na memória RAM para o salto incondicional (26 bits). Nas instruções tipo J, o opcode é fixo, assim como INST R, mas não nulo.

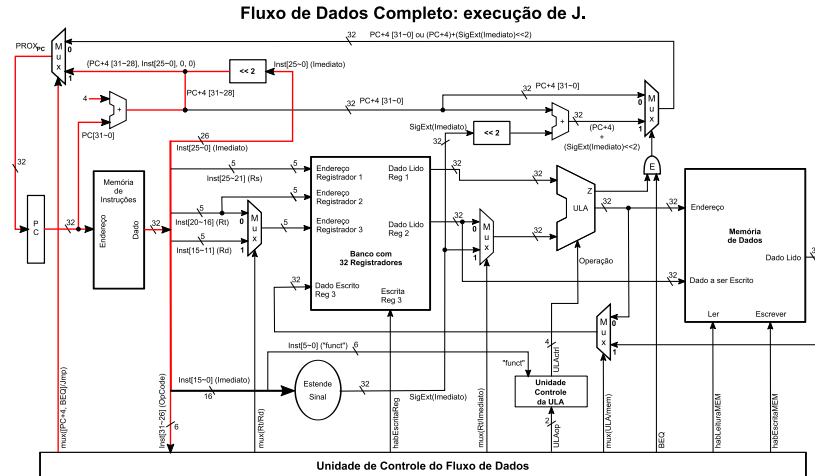
A sintaxe dessa instrução é do tipo indicado a seguir.

J ENDEREÇO

As instruções implementadas deste grupo são as definidas na seção 1, em instruções, que possuem a marcação INST J.

FD INST J

O FD parcial usado pelas instruções do tipo J é apresentado a seguir. Trata-se visão completa do fluxo de dados do MIPS DLX pode ser vista na seção 7 – FD, mas com destaque em vermelho para o caminho do FD INST J, que ocupa apenas a parte inicial do FD. Não chega a acessar o banco de registradores, retornando para o PC já na primeira etapa.



SEÇÃO 5
ULA

CONTEÚDO

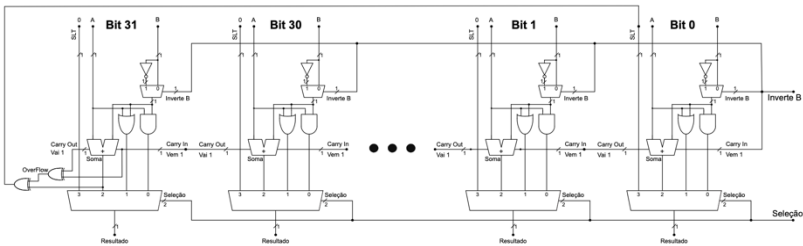
Contextualização.....5-1
FD ULA.....5-1
CTR ULA5-1

CONTEXTUALIZAÇÃO

A ULA é um componente crítico do FD MIPS DLX para INST R, INST I que tem a dedicação exclusiva desta seção do manual. Faz operações SUM, AND, OR, SUB e SLT.

FD ULA

O FD ULA é apresentado no diagrama a seguir, abstraindo os bits 2 a 29.



CTR ULA

CTR ULA é o nome dado à Unidade de Controle da Unidade Lógica Aritmética do MIPS DLX, que decodifica dois conjuntos de sinais de entrada.

O primeiro conjunto é o proveniente do CTR FD, que informa o tipo de operação a ser executada pela ULA.

O segundo conjunto é o campo funct da instrução sendo executada que é utilizado em INST R.

AVISO

O campo funct é usado somente em INST R. O formato de instruções de INST I não contempla a existência de um campo funct. Operações lógicas e aritméticas com imediato requerem modificações substanciais de hardware, que não estão contempladas na entrega intermediária do projeto.

AVISO

INT J não utiliza a ULA e, portanto, a CTR ULA não admite sinais desse tipo.

A saída da CTR ULA é efetivamente o controle de qual operação a ULA irá realizar, fazendo o chaveamento no MUX da ULA, controlando seu FD.

**SEÇÃO 6
PIPELINE**

CONTEÚDO

Ciclo único.....6-1

CICLO ÚNICO

O estado atual do MIPS DLX é ciclo único. Ou seja, não há pipeline implementado. Esta seção deverá ser atualizada na próxima entrega do documento em 15 de dezembro de 2021.

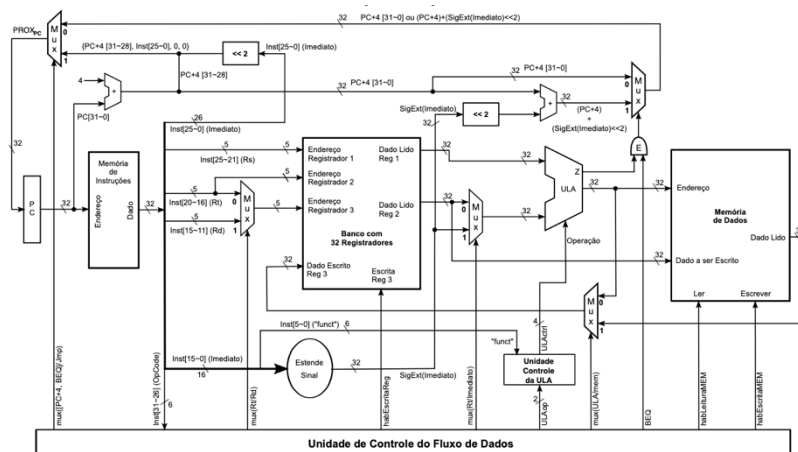
SEÇÃO 7
FD

CONTEÚDO

FD ciclo único.....	7-1
CTR FD.....	7-2

FD CICLO ÚNICO

FD é o nome dado ao Fluxo de Dados. O FD completo do MIPS DLX de ciclo único com apenas as instruções básicas é apresentado abaixo, em diagrama feito por Paulo Carlos Ferreira dos Santos.



NOTA

Ciclo único na versão da entrega intermediária de 15 de novembro de 2021, apenas. Pretendemos implementar pipeline para a versão final futuramente.

O FD parcial pelas instruções de diferentes tipos foi apresentado nas seções correspondentes a cada tipo de instrução (Seção 2 – INST R, Seção 3 – INST I e Seção 4 – INST J) anteriormente.

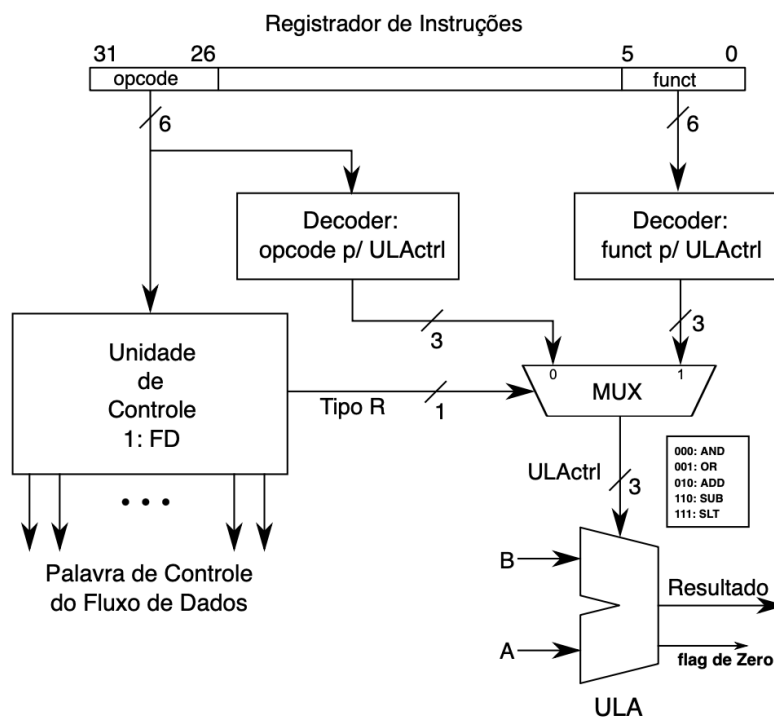
CTR FD

CTR FD é o nome dado à Unidade de Controle do Fluxo de Dados do MIPS DLX, que decodifica o opcode, gerando dois conjuntos de sinais na sua saída.

O primeiro conjunto abrange os pontos de controle do fluxo de dados.

O segundo conjunto é a informação de qual operação a ULA deve executar e é enviado para CTR ULA.

FD CTR FD (Fluxo de Dados da Unidade de Controle do Fluxo de Dados) é apresentado a seguir.



**SEÇÃO 8
EXTRA**

CONTEÚDO

Informações complementares.....7-1

INFORMAÇÕES COMPLEMENTARES

Esta seção é destinada a conter informações complementares a respeito do MIPS DLX que não foram contempladas nas seções interiores. Por ocasião da entrega intermediária, não temos informações complementares a reportar, mas talvez seja interessante voltar a ler esta seção quando for feita a entrega final, prevista para o dia 15 de dezembro de 2021.