Universidade Federal da Fronteira Sul
Curso de Ciência da Computação
Campus Chapecó

Arquitetura do Conjunto de Instruções e Formato de Representação das Instruções

Prof. Luciano L. Caimi
lcaimi@uffs.edu.br

Classes de instruções

- Instruções lógicas e aritméticas
  - ADD, SUB, AND, XOR, etc...
- Instruções de desvio condicional
  - BNE, BEQ, BLT, etc...
- Instruções de desvio incondicional
  - JUMP, CALL, etc...
- Instruções de movimentação de dados
  - MOV, LOAD, STORE, etc...
- Instruções de controle
  - NOP, EN_INT,

Arquitetura do Conjunto de Inst.

Instrução Assembly

As instruções assembly são definidas a partir da sua sintaxe e semântica.

A sintaxe refere-se ao conjunto de regras que regem a formação do "*texto*".

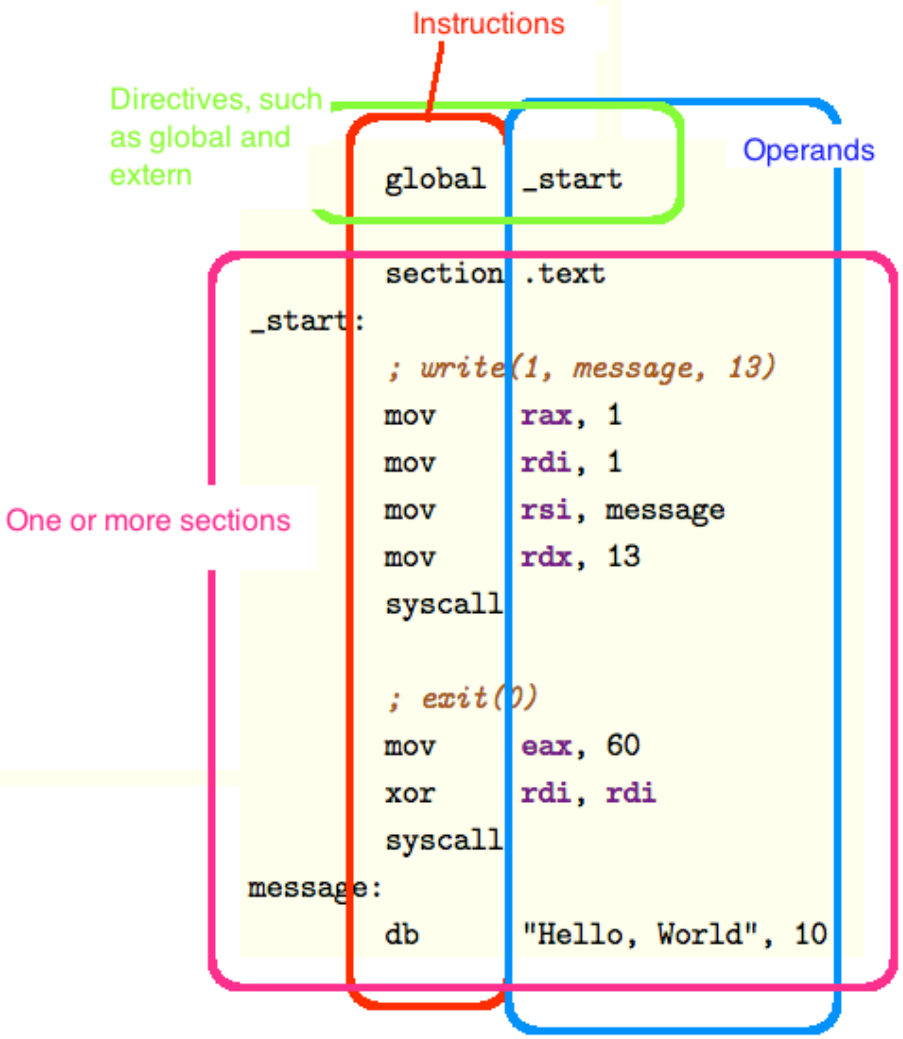
A semântica refere-se ao significado ou a interpretação deste "*texto*".

Sintaxe	Semântica
push op ;	$pilha_{topo} \leftarrow (op)$

onde: ( ) indica acesso a memória

Arquitetura do Conjunto de Inst.

Arquivo ASM



Arquitetura do Conjunto de Inst.

Classificação da Arq. Do Conj. de Inst.

Quanto ao tipo de armazenamento interno:

- Arquitetura de Pilha
- Arquitetura de Acumulador
- Arquitetura de Registradores de Propósito Geral
  - *arquitetura register-memory*
  - *arquitetura load-store* (*register-register*)
- Arquitetura Memória-Memória

Os operandos podem ser chamados explicita-mente ou implicitamente

Arquitetura do Conjunto de Inst.

Classificação da Arq. Do Conj. de Inst.

Exemplo: C = A+B

Pilha	Acumulador	Registrador (register-memory)	Registrador (load-store)
Push A	Load A	Load R1, A	Load R1, A
Push B	Add B	Add R1, B	Load R2, B
Add	Store C	Store C, R1	Add, R3, R1, R2
Pop C			Store C, R3

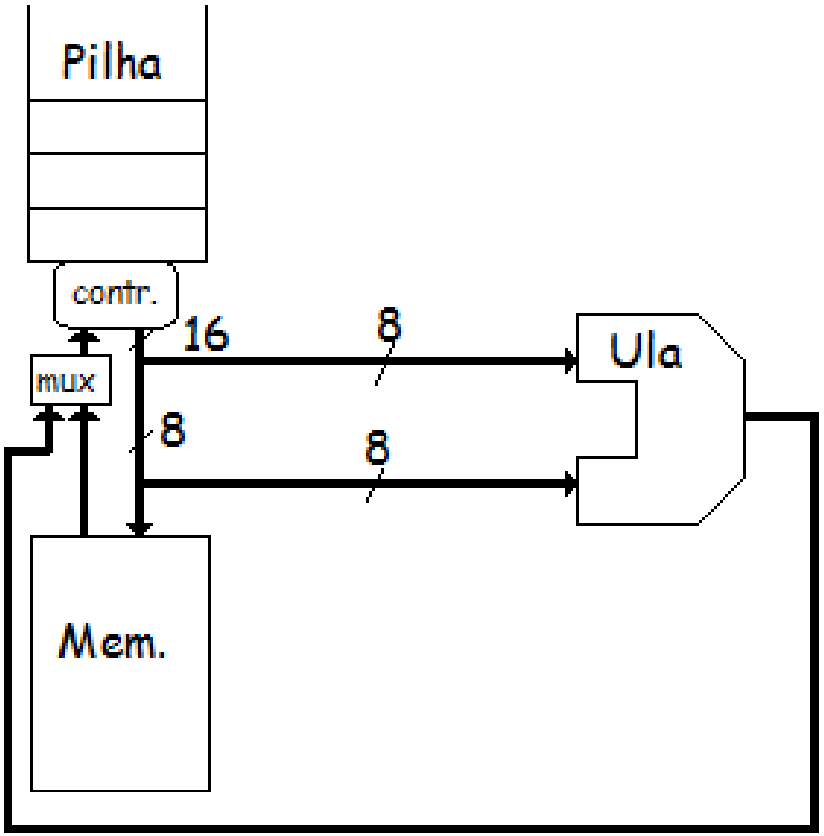
Arquitetura do Conjunto de Inst.

Pilha

```
Pop  op; (op) ← topo
Push op; topo ← (op)
Add    ; topo ← topo + topo-1
Sub    ; topo ← topo - topo-1
Mul    ; topo ← topo * topo-1
Div    ; topo ← topo / topo-1
```

Arquitetura do Conjunto de Inst.

Pilha



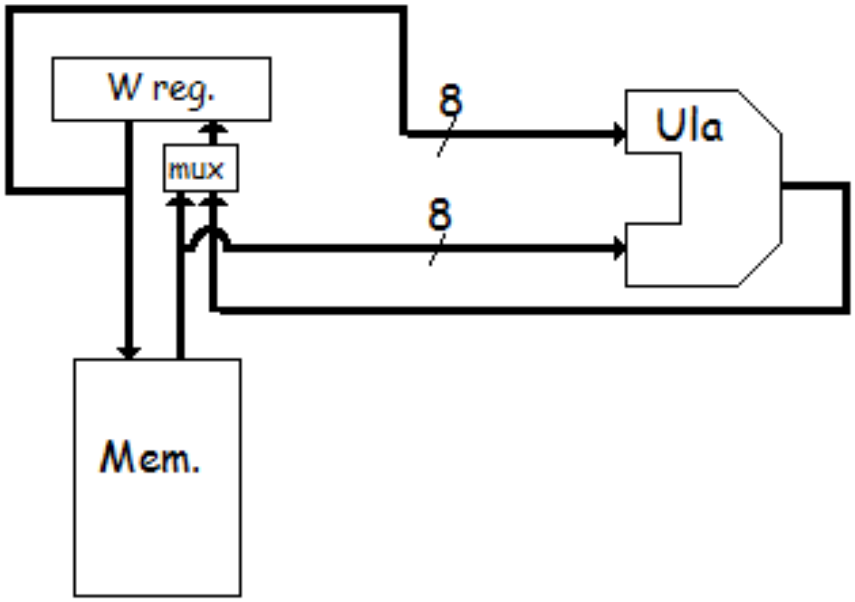
Arquitetura do Conjunto de Inst.

Acumulador (work register)

```
Movmw op; w ← (op)
Movwm op; (op) ← w
Add   op; w ← w + (op)
Sub   op; w ← w - (op)
Mul   op; w ← w * (op)
Div   op; w ← w / (op)
```

Arquitetura do Conjunto de Inst.

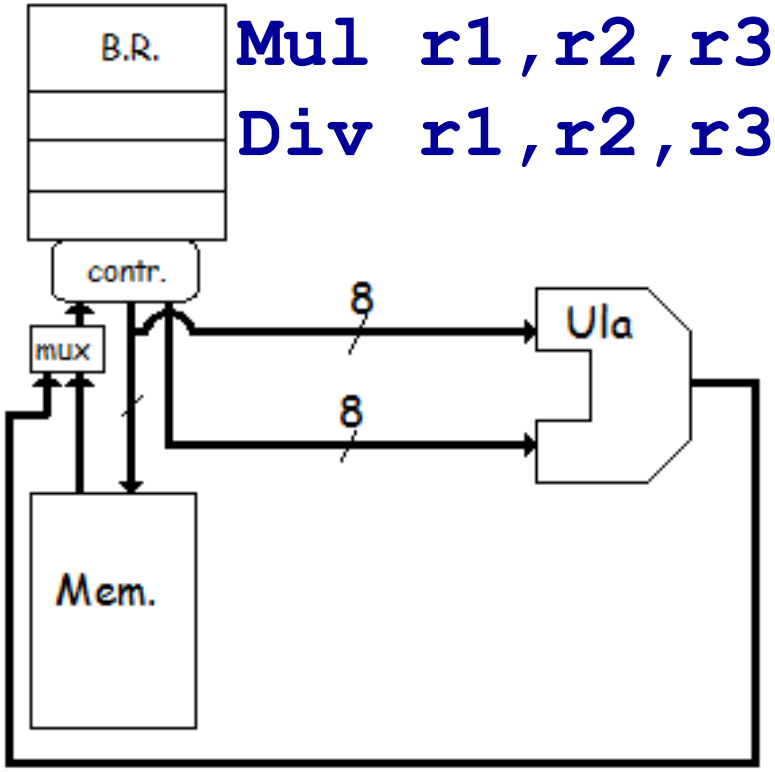
Acumulador (work register)



Arquitetura do Conjunto de Inst.

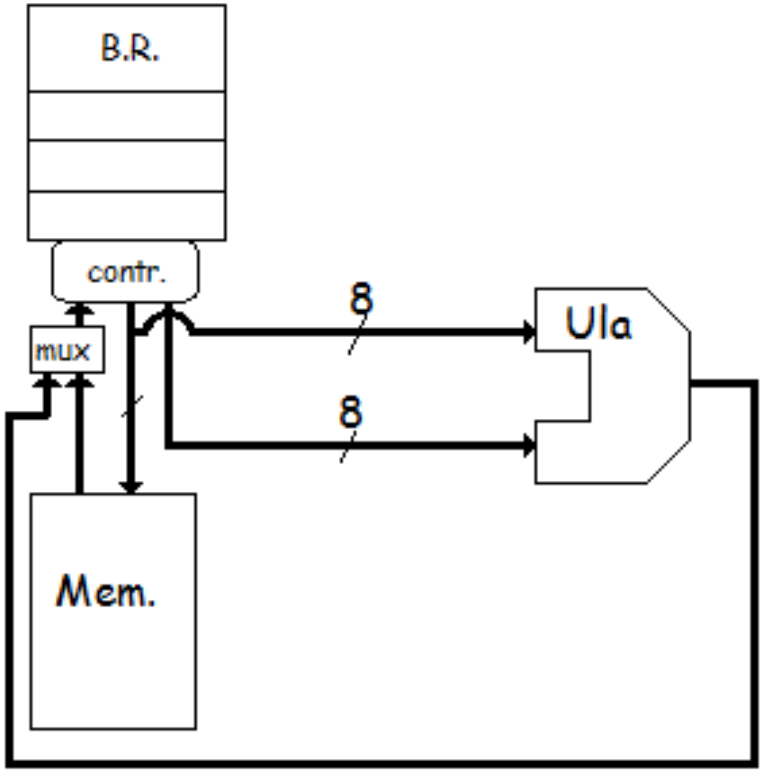
Load-Store

```
Load  op, R   ; R ← (op)
Store op, R   ; (op) ← R
Add r1,r2,r3  ; r1 ← r2 + r3
Sub r1,r2,r3  ; r1 ← r2 - r3
Mul r1,r2,r3  ; r1 ← r2 * r3
Div r1,r2,r3  ; r1 ← r2 / r3
```



Arquitetura do Conjunto de Inst.

Load-Store



Instrução Assembly

Exercício:

Defina um conjunto de instruções assembly com armazenamento interno tipo pilha para implementar a equação:

$S = (A - ((C + B) / A)) + ((D - B) / 2)$

Arquitetura do Conjunto de Inst.

Amazenamento Interno

Load-Store (Register/Register) - (0, 3)

acessos a memória

qtd operandos

VANTAGENS

- simples;
- codificação de comprimento fixo para instruções;
- modelo simples de geração de códigos;
- as instruções gastam um número de *clocks* semelhantes.

DESVANTAGENS

- para uma determinada tarefa, requer-se um número maior de instruções, com relação as arquiteturas cujas instruções possuem referências à memória;
- algumas instruções são pequenas, ocasionando desperdício de *bits* na codificação.

Arquitetura do Conjunto de Inst.

Amazenamento Interno

Register/Memory - (1, 2)

VANTAGENS

- pode-se acessar o dado sem o uso da instrução LOAD;
- o formato das instruções proporciona uma fácil decodificação e produz uma boa densidade.

DESVANTAGENS

- os operandos não são equivalentes, pois um operando fonte é destruído em uma operação binária;
- a codificação de um número de registro, e de um endereço de memória em cada instrução, pode restringir o número de registradores;
- os clocks por instrução variam pela localização do operando.

Amazenamento Interno

Memory/Memory - (3, 3)

VANTAGENS

- Programas mais compactos (menos linhas de código);
- não gasta registradores para armazenamentos temporários.

DESVANTAGENS

- grande variação do comprimento da instrução, especialmente para instruções de três operandos;
- grande variação no trabalho por instrução;
- geração de gargalo nos acessos à memória,

Tipos de Ordenação de dados

Refere-se a ordem que os dados são armazenas dos na memória. Imagine o valor hexa 0x3AF2 a ser armazenado a partir do endereço 0x50

LITTLE ENDIAN

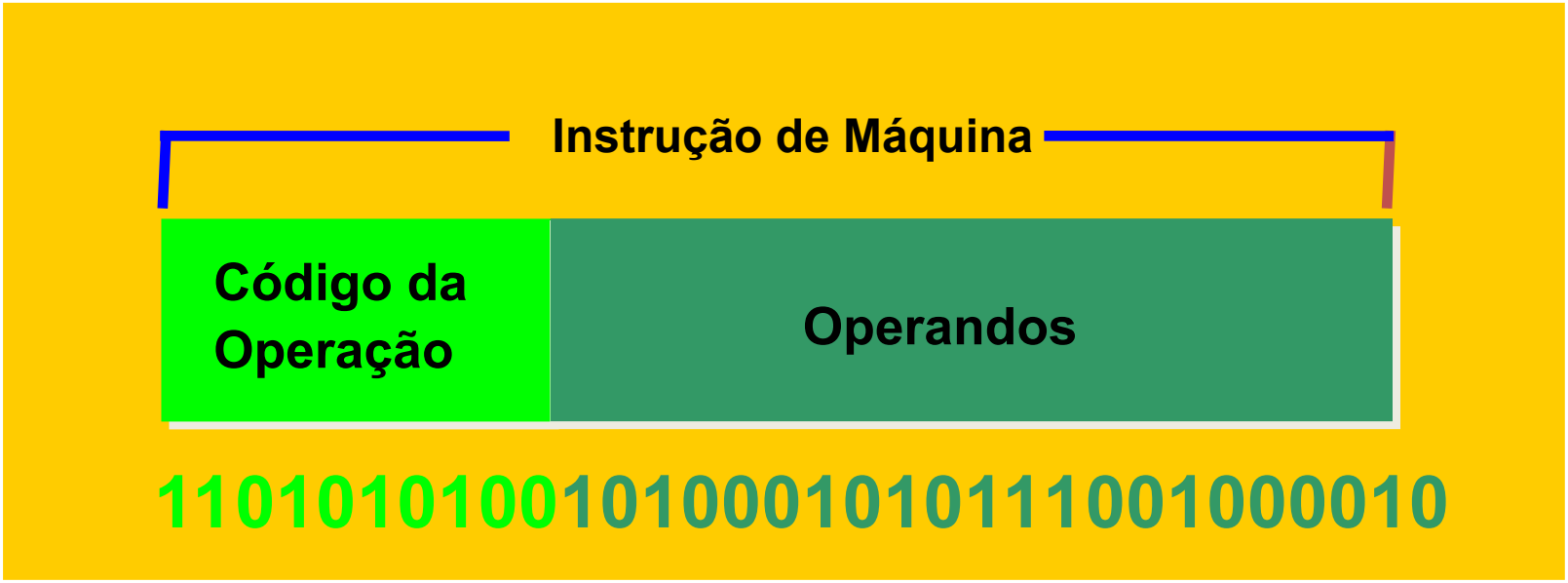
o endereço de um dado (0x50) é o endereço do byte menos significativo (0xF2).

BIG ENDIAN

o endereço de um dado (0x50) é o endereço do byte mais significativo (0x3A).

Arquitetura do Conjunto de Inst.

Formato da Instrução



Código da Operação: o que deve-se fazer

Operando: onde estão os dados a serem manipulados

Arquitetura do Conjunto de Inst.

O campo operando

Quantidade de operandos:

- 3 operandos SUB R1,R2,R3
- 2 operandos MOV R1,R2
- 1 operando ADD R1
- 0 operandos NOP

Modos de Endereçamento: como interpretar o campo operando no que diz respeito a onde se encontra o dado utilizado pela instrução

Modo de endereçamento

Modo Imediato:

- o dado a ser manipulado está indicado no próprio campo operando da instrução;
- Utilizadas na inicialização de variáveis e ponteiros; operações com constantes e desvios;
- Vantagem: poucos acessos a memória;
- Desvantagem: limitação do campo operando restringe o valor máximo manipulado.

Exemplo: JMP Op; CI ← Op; C.Op. = Ah

Instrução: A35h

Modo de endereçamento

Modo Direto

- O valor contido no campo operando indica o endereço de memória onde se localiza o dado a ser manipulado.
- Pode ser o endereço inicial do dado na memória;
- Um dos formatos naturais de implementar as variáveis do programa. Cada variável representa um endereço de memória;

Exemplo: LDA Op; W ← (Op); C.Op. = 7h
Instrução: 735h

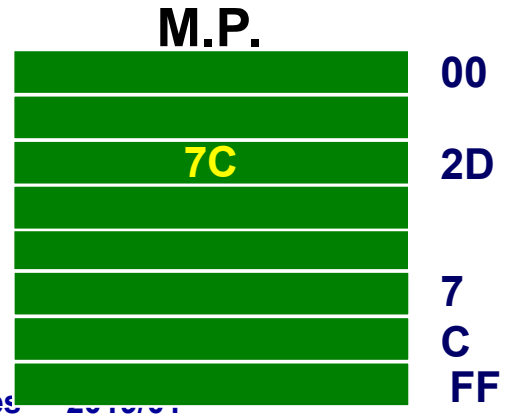
Modo de endereçamento

Modo Indireto

- O campo operando representa uma célula de memória, entretanto o valor contido neste endereço não é o dado e sim o endereço onde o dado se encontra;
- Utilizado na implementação de ponteiros;

Exemplo: LDAI Op; W ← ((Op)) ; C.Op. = 9h

Instrução: 92Dh



Modo de endereçamento

Direto por Registrador

- O campo operando representa o número do registrador onde o dado se encontra;
- Vantagens:
  - acesso aos registradores é mais rápida que o acesso a memória;
  - número menor de bits para endereçar registradores;
- Principal modo de endereçamento nas arquiteturas RISC;

Exemplo: ADD R1 W ← W + R1; C.Op. = 8h
Instrução: 805h

Modo de endereçamento

Indireto por Registrador

- O operando (registrador) aponta para o endereço de memória onde o dado se encontra;
- Pode ser implementado de várias maneiras; os modos a seguir são casos especiais deste modo:
  - Indireto
  - Base+deslocamento

Modo de endereçamento

Modo por Base + Deslocamento

- Neste modo o registrador base aponta para o início de um bloco e o deslocamento informa qual é o deslocamento dentro daquele bloco.
- O endereço onde o dado se encontra é obtido:
  - a partir da soma entre o valor contido no registrador base e no deslocamento
  - a partir da concatenação entre o valor contido no registrador base e no deslocamento.

Arquitetura do Conjunto de Inst.

Quantidade de Operandos

3 Operandos

(Op1) ← (Op2) operação (Op3)

ADD Op1, Op2, Op3; (Op1) ← (Op2) + (Op3)

SUB Op1, Op2, Op3; (Op1) ← (Op2) – (Op3)

MUL Op1, Op2, Op3; (Op1) ← (Op2) * (Op3)

DIV Op1, Op2, Op3; (OP1) ← (Op2) / (Op3)

- Instrução é completa (não sobrescreve valor);
- Quantidade de operandos pode limitar o espaço de endereçamento ou valor representado.

Arquitetura do Conjunto de Inst.

Quantidade de Operandos

2 Operandos

(Op1) ← (Op1) operação (Op2)

ADD Op1, Op2; (Op1) ← (Op1) + (Op2)

SUB Op1, Op2; (Op1) ← (Op1) – (Op2)

MUL Op1, Op2; (Op1) ← (Op1) * (Op2)

DIV Op1, Op2; (Op1) ← (Op1) / (Op2)

MOV Op1, Op2; (Op1) ← (Op2)

- Instrução não é completa;
- Instrução MOV deve ser implementada.

Arquitetura do Conjunto de Inst.

Quantidade de Operandos

1 Operando

W ← W operação (Op)

ADD Op; W ← W + (Op)

SUB Op; W ← W – (Op)

MUL Op; W ← W * (Op)

DIV Op; W ← W / (Op)

LDA Op; W ← (Op)

STR Op (Op) ← W

- Utilização de um registrador especial; As operações ocorrem entre o operando e este registrador;
- Instruções LDA e STR devem ser implementadas;

Arquitetura do Conjunto de Inst.

Quantidade de Operandos

0 Operandos

Utilizado em casos em que a própria instrução indica o dado/operando manipulado, ou quando a instrução não exige dado ou operando;

Exemplos: NOP ; no operation

CLRW ; W ← 0