

Endereçamento e Formato de Instruções

S. W. Song

MAC 412 - Organização de Computadores

Endereçamento e Formato de Instruções

Veremos:

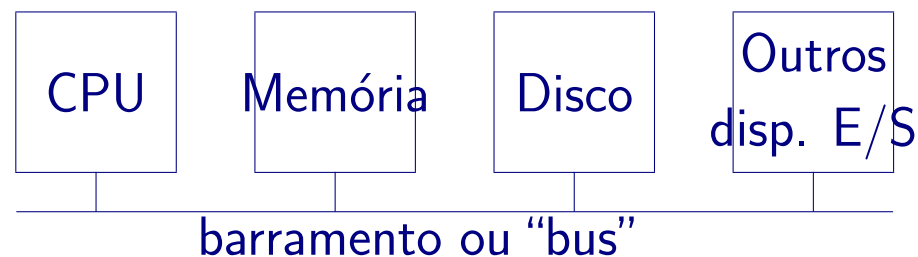
- Espaço de endereçamento.
- Formato de instruções.
- Códigos de operação (opcodes) que expandem.

Projeto do Formato de Instruções

Na arquitetura de von Neumann o programa está na memória e cada instrução é executada no processador.

O processador (datapath) pode operar dados de um certo número de bits, digamos 64 bits. O barramento (bus), pode transportar em paralelo, um certo número de bits, digamos 64 bits. Nesse caso, diremos que o processador é de 64 bits.

O formato de instruções portanto é importante, uma vez que cada instrução deve ser levada da memória para o processador, para ali ser executada.



Espaço de endereçamento

Um endereço absoluto de m bits pode endereçar 2^m posições de memória.

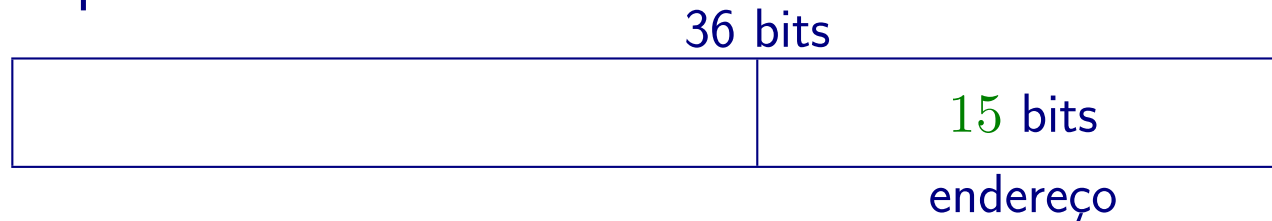
Há várias maneiras de obter tal endereço absoluto.

- Contido na própria instrução
- Uso de registradores base
- Uso de registradores de segmento
- Uso de registradores de banco
- Uso de modos de endereçamento

Contido na própria instrução

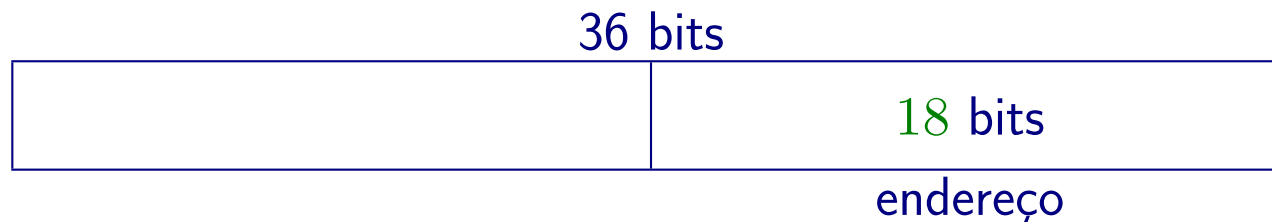
A instrução contém espaço suficiente para incluir todos os m bits do endereço absoluto.

Exemplo - IBM 7094:



$$2^{15} = 32 \text{ K palavras}$$

Exemplo - DEC PDP-11:



$$2^{18} = 256 \text{ K palavras}$$

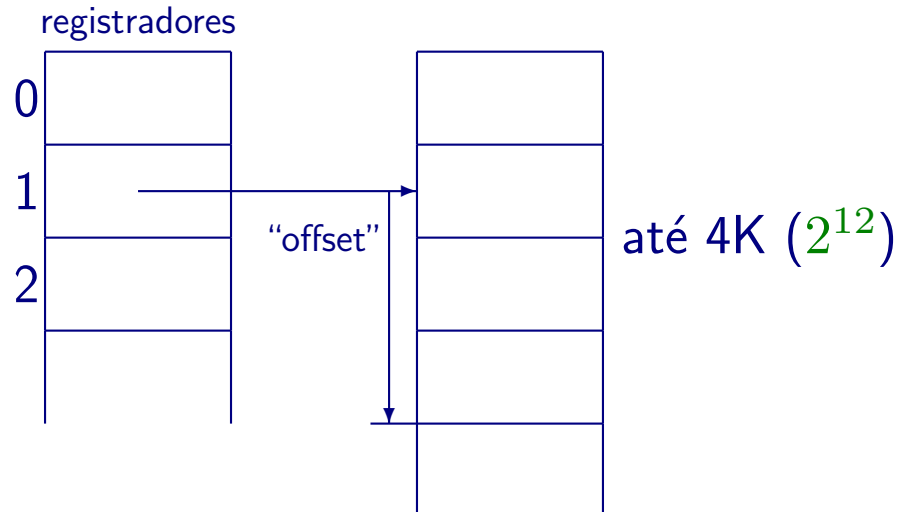
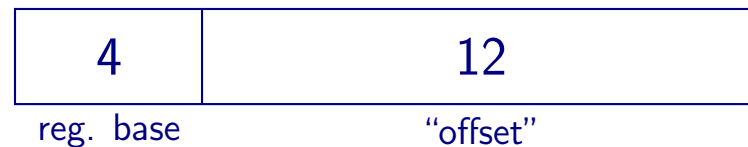
Esse esquema simples apresenta alguns problemas:

- Requer instruções de muitos bits
- Espaço de endereçamento limitado

Uso de registradores base

O endereço é obtido somando-se o conteúdo de um registrador base e um deslocamento (“offset”).

Exemplo: IBM 360 e 370 possuem 16 registradores, com 24 bits de cada registrador dedicados para endereçamento.



Uso de registradores de segmento

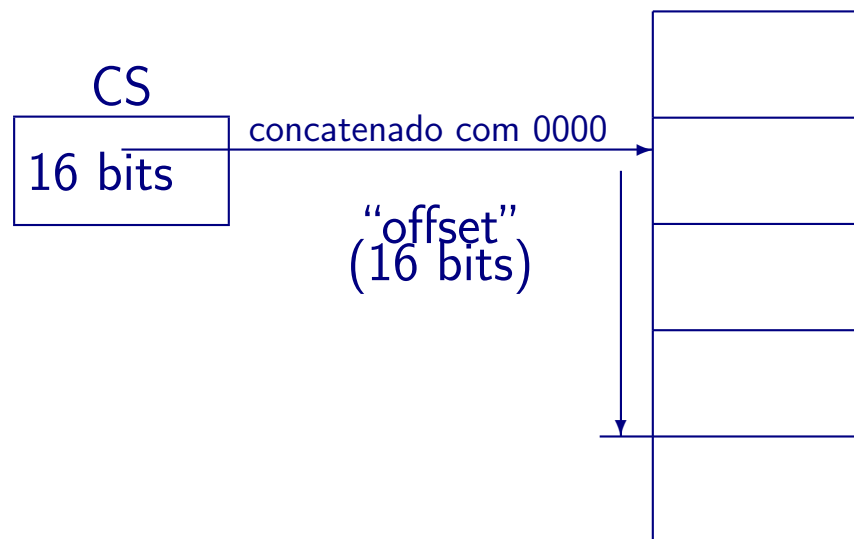
Registradores de segmento apontam para determinados segmentos (programa, dados, pilha, etc.). Um deslocamento (“offset”) determina a posição dentro do segmento.

Exemplo - Intel 8088 e 8086 apresentam 4 registradores de segmento chamados

- CS code segment (programa)
- DS data segment (dados)
- SS stack segment (stack ou pilha)
- ES extra segment (outros dados)

Registradores de Segmento Intel 8088/8086

O Intel 8088 ou 8086 usa endereço absoluto de 20 bits, dando 2^{20} ou 1M posições endereçáveis.

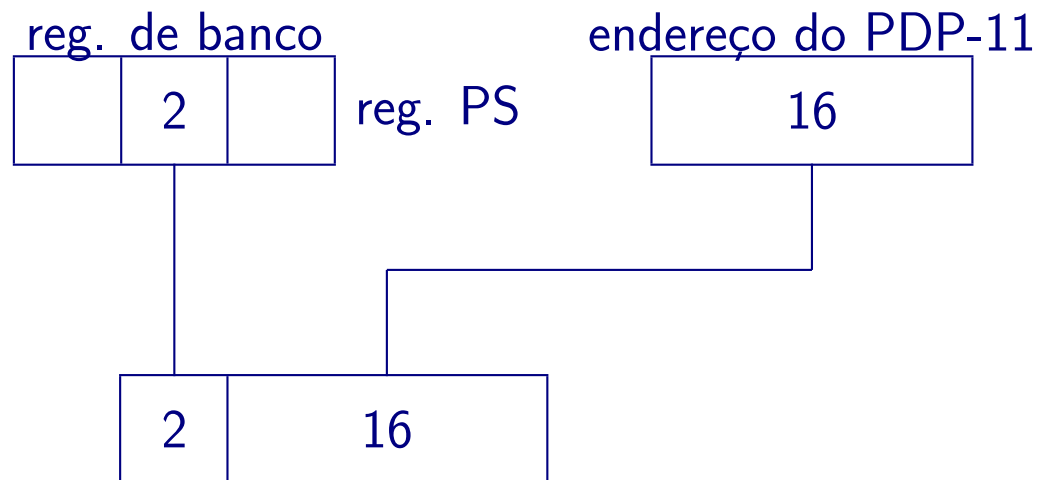


O valor do registrador de segmento é primeiro concatenado com 4 bits iguais a 0000. Depois somado ao valor do offset dá o endereço final de 20 bits.

Uso de registradores de banco

O endereço é o conteúdo do registrador de banco concatenado com o campo de offset.

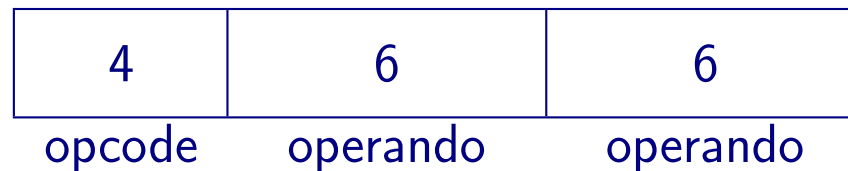
Exemplo - O multiprocessador C.mmp (constituído de 16 processadores PDP-11):



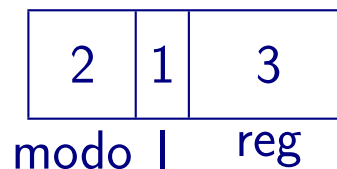
Uso de modos de endereçamento

Introduzido pelo PDP-11, esse esquema permite uma grande economia no tamanho da instrução.

PDP-11: instrução de apenas 16 bits para gerar 2 endereços de operandos.

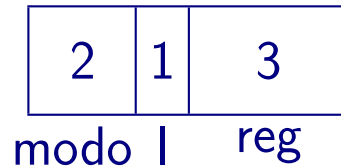


Cada operando é constituído de 3 campos.



Obs: I = endereçamento indireto

Modos de endereçamento



- modo 00 - registrador: reg contém o operando
- modo 01 - auto-incr: reg é incrementado após uso como endereço do operando
- modo 10 - auto-decr: reg é decrementado e usado como endereço do operando
- modo 11 - indexado: conteúdo da palavra seguinte à instrução é somado com conteúdo do registrador reg para dar o endereço do operando

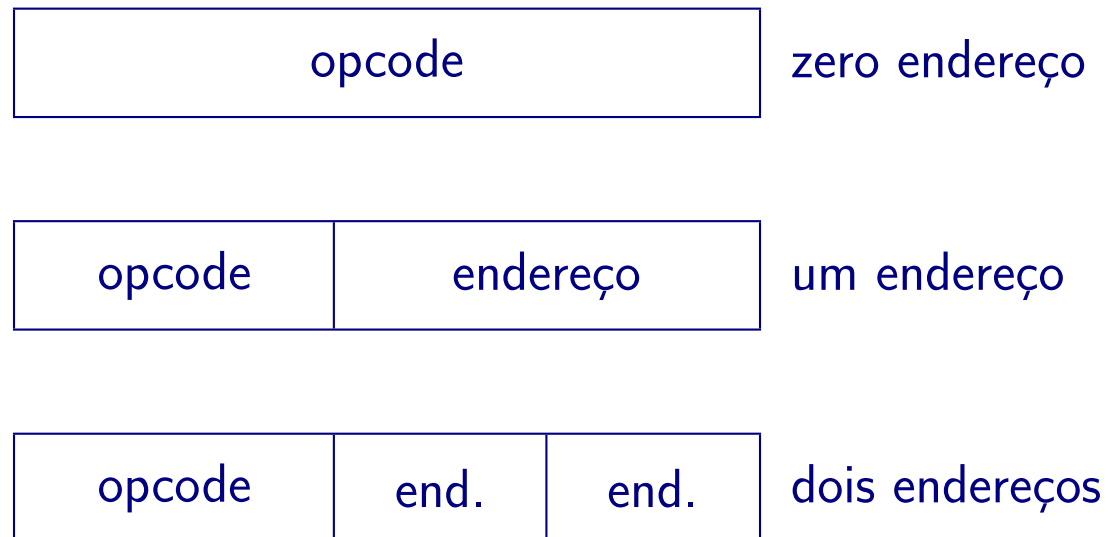
Outros exemplos: Intel e Motorola usam esquemas analógos, porém mais complexos.

Formato de instruções

Uma instrução contém:

- opcode: código de operação
- endereços de operandos

Pode haver vários tipos de instruções.



“Opcodes que expandem”

Consideremos um exemplo de uma máquina cujas instruções são de 16 bits e endereços de 4 bits.



Cada endereço de 4 bits especificam, por exemplo, um dos 16 registradores da máquina. Com o formato acima podemos ter 16 instruções de 3 endereços.

“Opcodes que expandem”

Suponhamos, porém, que precisamos de

- 15 instruções de 3 endereços
- 14 instruções de 2 endereços
- 31 instruções de 1 endereço
- 16 instruções de 0 endereço

“Opcodes que expandem”

- 15 instruções de 3 endereços:

opcode			
0000	xxxx	yyyy	zzzz
0001	xxxx	yyyy	zzzz
	.		
	.		
1110	xxxx	yyyy	zzzz

- 14 instruções de 2 endereços:

opcode			
1111	0000	yyyy	zzzz
1111	0001	yyyy	zzzz
1111	0010	yyyy	zzzz
	.		
	.		
1111	1101	yyyy	zzzz

“Opcodes que expandem”

- 31 instruções de 1 endereço:

opcode			
1111	1110	0000	zzzz
1111	1110	0001	zzzz
	.		
	.		
1111	1110	1111	zzzz
1111	1111	0000	zzzz
1111	1111	0001	zzzz
	.		
	.		
	.		
1111	1111	1110	zzzz

“Opcodes que expandem”

- 16 instruções de 0 endereço:

opcode			
1111	1111	1111	0000
1111	1111	1111	0001
	.		
	.		
	.		
1111	1111	1111	1111

Em computadores reais, os opcode não são expandidos como no exemplo, de maneira tão regular e limpa, como veremos abaixo.

Formato de instrução no Motorola 68030

Teve uma influência grande do formato de instruções do PDP-11, porém muito mais complexo, devido a maior número de instruções. Além disso, o Motorola 68030 deve ainda considerar 3 tipos de operandos:

- byte 8 bits
- word 16 bits
- long 32 bits

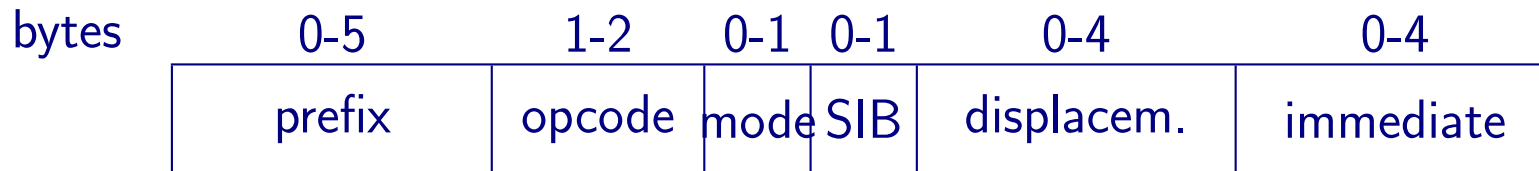
A figura seguinte mostra as primeiras palavras de 18 formatos diferentes.

Formato de instrução no Motorola 68030

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00

op	size	operand			operand			MOVE
opcode		reg	mod		operand			ADD, AND
opcode		reg	op		operand			CHK, DIVS
opcode		reg	mod		op	reg		MOVEP
opcode		reg	op size		op	reg		ASL, ROL
opcode		reg		opcode		reg		ABCD
opcode		reg	op	data				MOVEQ
opcode		count	op size		op	reg		ASL, ROL
opcode		data	op size		operand			ADDQ
opcode		condition	op		operand			Scc
opcode		condition		displacement				Bcc
opcode		condition		opcode		reg		DBcc
opcode		size			operand			ADD1
opcode		sz			operand			MOVEM
opcode					operand			JMP
opcode					vector			TRAP
opcode					reg			EXT, SWAP
opcode								NOP

Formato de instrução no Intel 80386



O formato de instrução no Intel 80386 é também muito complicado, sendo a instrução mais curta de 1 byte e a mais comprida até 17 bytes.

Há pouca estrutura ou regularidade no formato, exceto talvez nos 2 últimos bits do opcode:

- O bit da direita do opcode indica se o endereço de memória, se usado, é fonte ou destino da operação.
- O penúltimo bit da direita do opcode indica se o operando é tipo byte ou word.

Formato de instrução no Intel 80386

A figura mostra os campos opcode e mode.

SIB é presente apenas no Motorola 68030, para dar um byte adicional ao modo.



f/d=fonte/destino

b/w=byte/word