

MICROPROCESSADORES E MICROCONTROLADORES



CONJUNTO DE INSTRUÇÕES

Está provado que só é possível
filosofar em alemão.
Caetano Veloso

CONJUNTO DE INSTRUÇÕES

YAMABUKI YÁ
HA NI HANÁ NI HA NI
HANÁ NI HA NI
Taigi

Tradução livre:

a montanha sopra
folha em flor em folha em
flor em folha em

CONJUNTO DE INSTRUÇÕES

YAMABUKI YÁ

HA NI HANÁ NI HA NI

HANÁ NI HA NI

Taigi

yama-búki (montanha-sopra) é o nome da
rosa amarela

Tradução livre:

a montanha sopra
folha em flor em folha em
flor em folha em

CONJUNTO DE INSTRUÇÕES

YAMARUKI YÁ

HA NI HANÁ NI HA NI

HANÁ NI HA NI

Taigi

ha é folha
haná é flor

Tradução livre:

a montanha sopra
folha em flor em folha em
flor em folha em

CONJUNTO DE INSTRUÇÕES

YAMARUKI YÁ

HA NI HANÁ NI HA NI

HANÁ NI HA NI

Taigi

ni pode ser tanto 'e'
quanto 'em':
'folha e flor',
'folha em flor'

Tradução livre:

a montanha sopra
folha em flor em folha em
flor em folha em

CONJUNTO DE INSTRUÇÕES

YAMARUKI YÁ

HA NI HANÁ NI HA NI

HANÁ NI HA NI

Taigi

haná é flor
hana é nariz

Tradução livre:

a montanha sopra
folha em flor em folha em
flor em folha em

CONJUNTO DE INSTRUÇÕES

YAMABUKI YÁ
HA NI HANÁ NI HA NI
HANÁ NI HA NI

Em um minúsculo haiku, Taigi apresenta diversas imagens, sons e cheiros que se perdem na tradução.

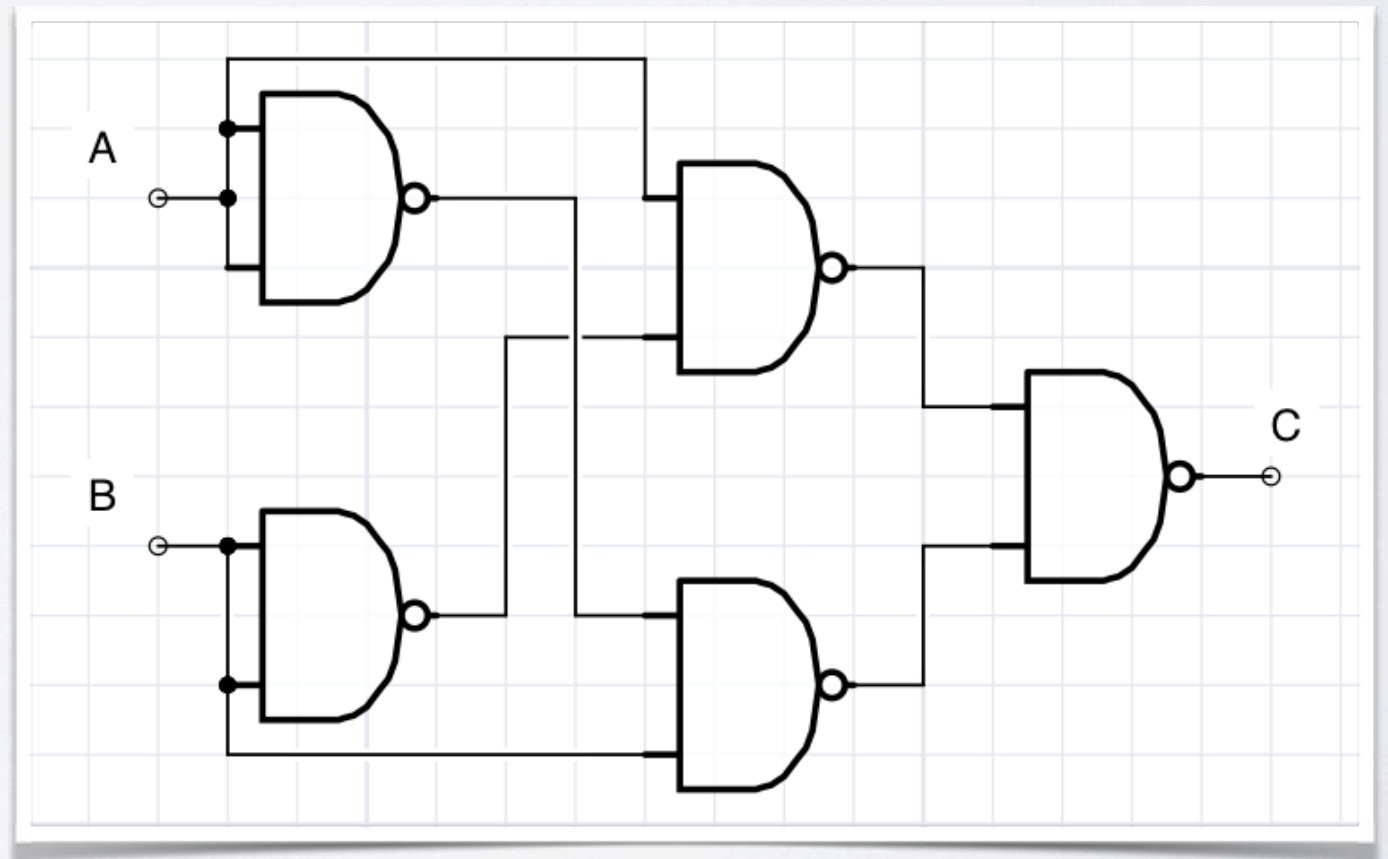
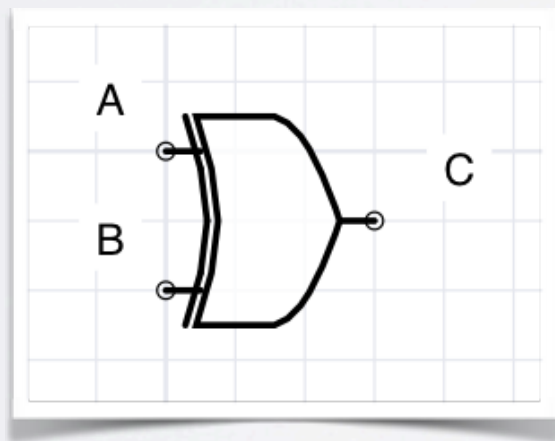
a montanha sopra
folha em flor em folha em
flor em folha em

CONJUNTO DE INSTRUÇÕES

A linguagem disponível afeta drasticamente a descrição e a compreensão do mundo, além da resolução de problemas.

CONJUNTO DE INSTRUÇÕES

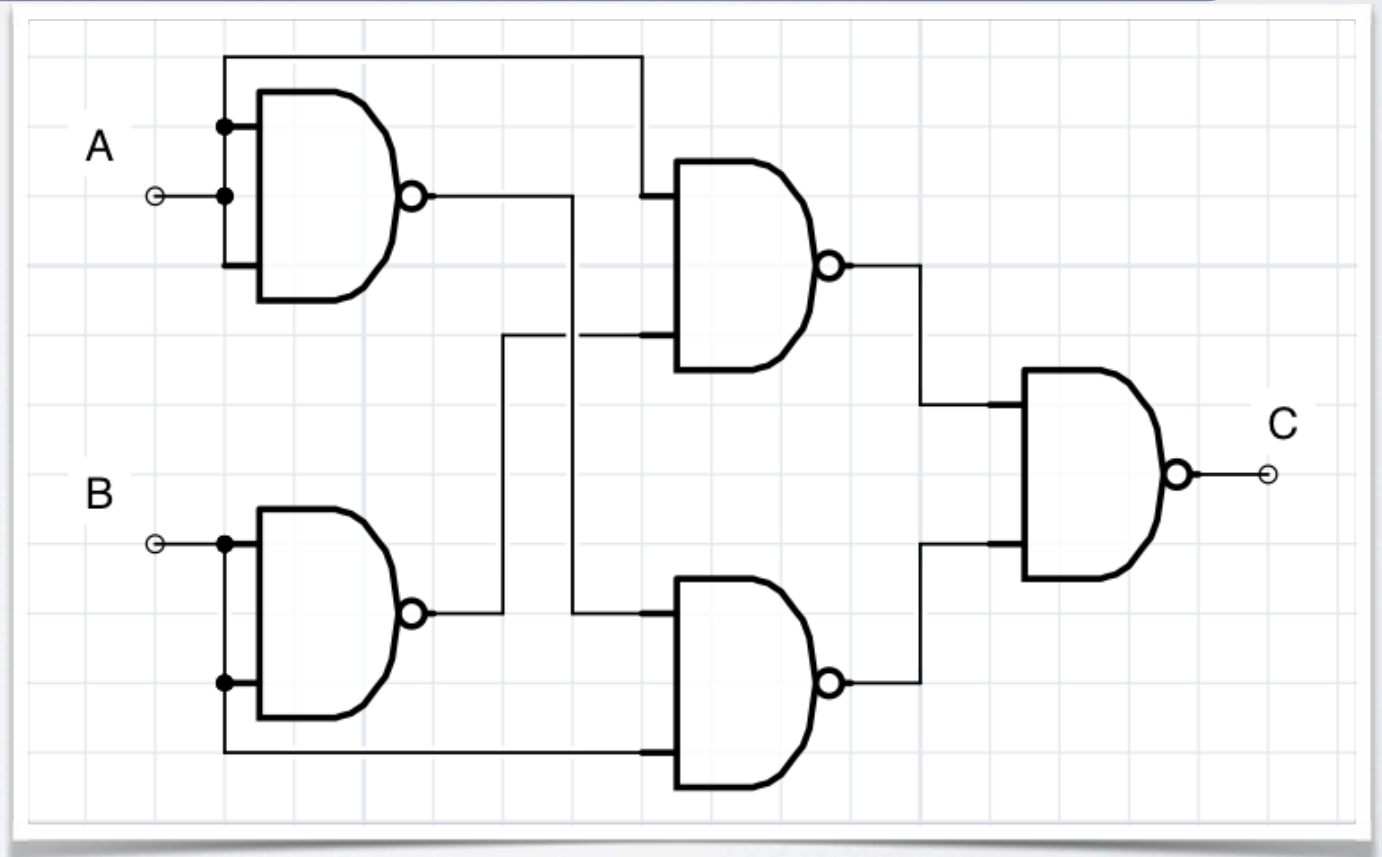
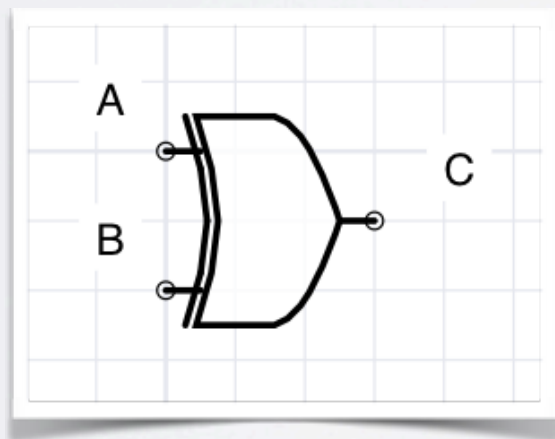
A linguagem disponível afeta drasticamente a descrição e a compreensão do mundo, além da resolução de problemas.



CONJUNTO DE INSTRUÇÕES

A linguagem disponível afeta drasticamente a descrição e a compilação das.

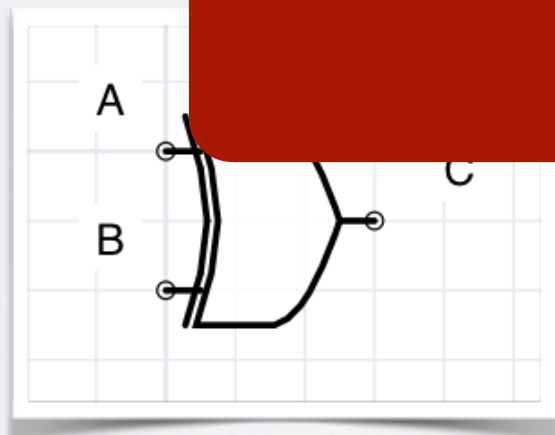
Mesma função lógica, dois circuitos



CONJUNTO DE INSTRUÇÕES

A linguagem disponível afeta drasticamente a descrição e a compreensão do mundo, além da resolução de problemas.

**Tente explicar para um americano
o que é saudade.**



CONJUNTO DE INSTRUÇÕES

Instruções são as palavras do computador, e o conjunto de instruções é seu vocabulário.

CONJUNTO DE INSTRUÇÕES

Instruções são as palavras do computador, e o conjunto de instruções é seu vocabulário.

MOV	#0AA55h,R5	; Load mask into register R5
AND	R5,TOM	; mask word addressed by TOM with R5
JZ	TONI	;

Instruções do MSP430

CONJUNTO DE INSTRUÇÕES

Instruções são as palavras do computador, e o conjunto de instruções é seu vocabulário.

```
MOV      #0AA55h,R5      ; Load mask into register R5
AND      R5,TOM           ; mask word addressed by TOM with R5
JZ       TONI            ;
```

Instruções do MSP430

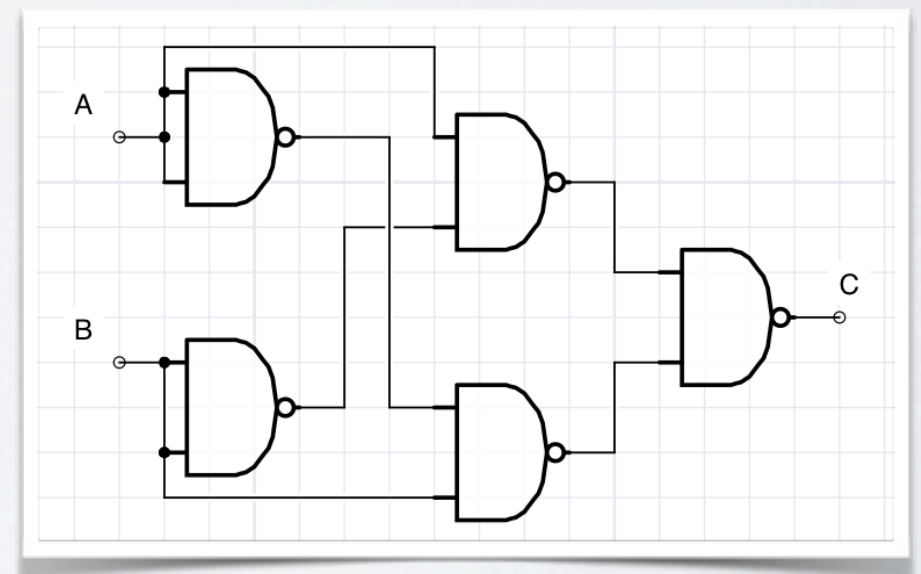
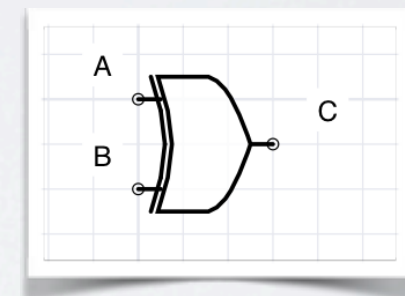
```
add a, b, c      # The sum of b and c is placed in a.
add a, a, d      # The sum of b, c, and d is now in a.
add a, a, e      # The sum of b, c, d, and e is now in a.
```

Instruções MIPS

CONJUNTO DE INSTRUÇÕES

A escolha do conjunto de instruções afeta:

- o hardware;
- os compiladores;
- a performance;
- o custo de um processador.



CONJUNTO DE INSTRUÇÕES

A escolha de um conjunto de instruções depende de muitas considerações, que fazem com que muitas linguagens se pareçam entre si.

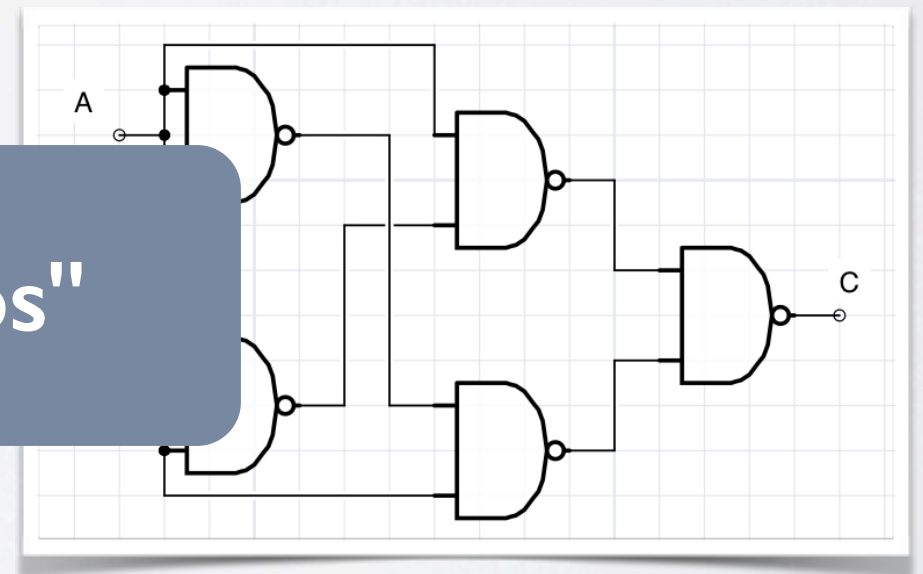
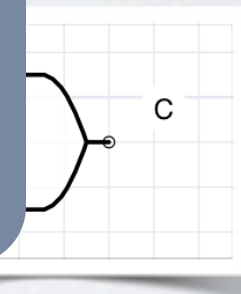
- o hardware;

- os compiladores;

- a performance;

- o custo de um processador.

Diferentes "dialetos"



CONJUNTO DE INSTRUÇÕES

Conceito do programa armazenado

A memória do computador armazena tanto dados de diferentes tipos quanto instruções, o que possibilita o desenvolvimento para computadores.



CONJUNTO DE INSTRUÇÕES

Conjuntos de instruções são construídos de acordo com os seguintes princípios:

1. A simplicidade favorece a regularidade.
2. Menor é mais rápido.
3. O caso comum deve ser mais rápido.
4. Um bom projeto demanda compromisso.



OPERAÇÕES ARITMÉTICAS

Todas possuem dois operandos: uma fonte e um destino.

```
mov.w    R5, R6
```


OPERAÇÕES ARITMÉTICAS

Todas possuem dois operandos: uma fonte e um destino.

A rectangular box with a thin black border and a light gray background. It contains the assembly instruction `mov.w R5, R6` in a monospaced font. The text is centered horizontally and vertically within the box.

```
mov.w R5, R6
```

Se a quantidade de operandos fosse variável, o hardware seria mais complexo.

OPERAÇÕES ARITMÉTICAS

Todas possuem dois operandos: uma fonte e um destino.

```
mov.w    R5, R6
```

Se a quantidade de operandos fosse variável, o hardware seria mais complexo.

I. A simplicidade favorece a regularidade.

OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
f = (g + h) - (i + j);
```


OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
f = (g + h) - (i + j);
```

```
mov.w g, t0    ; A variável intermediária 't0' guarda 'g'
add.w h, t0     ; A variável intermediária 't0' guarda uma soma
mov.w i, t1     ; A variável intermediária 't1' guarda 'i'
add.w j, t1     ; A variável intermediária 't1' guarda outra soma
sub.w t1, t0    ; A variável intermediária 't0' guarda a subtração
mov.w t0, f     ; A variável 'f' recebe o valor final
```

OPERAÇÕES ARITMÉTICAS

Os operandos em Assembly são diferentes de variáveis em C, Java etc.

```
mov.w    R5, R6
```

As operações são todas feitas com registradores, que armazenam dados diretamente no hardware (CPU etc.) ao invés armazená-los na memória.

OPERAÇÕES ARITMÉTICAS

A CPU do MSP430 possui 16 registradores de 16 bits, o que acelera a troca de dados entre CPU e periféricos.

Usaremos alguns desses registradores para guardar variáveis, e outros para guardar valores temporários.

OPERAÇÕES ARITMÉTICAS

A CPU do MSP430 possui 16 registradores de 16 bits, o que acelera a troca de dados entre CPU e periféricos.

2. Menor é mais rápido.

OPERAÇÕES ARITMÉTICAS

A CPU do MSP430 possui 16 registradores de 16 bits, o que acelera a troca de dados entre CPU e periféricos.

2. Menor é mais rápido.

Usaremos alguns desses registradores para guardar variáveis, e outros para guardar valores temporários.

OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
f = (g + h) - (i + j);
```

```
mov.w R5, R10 ; R10 recebe g (R5)
add.w R6, R10 ; R10 g+h (R5+R6)
mov.w R7, R11 ; R11 recebe i (R7)
add.w R8, R11 ; R11 = i+j (R7+R8)
sub.w R11, R10 ; R10 = R10-R11
mov.w R10, f ; f (R4) recebe o valor final
```


OPERANDOS NA MEMÓRIA

Registradores podem armazenar algumas variáveis importantes, mas nem sempre são suficientes.

Vetores, matrizes e estruturas são armazenados na memória.

Já as operações aritméticas são feitas com registradores.

OPERANDOS NA MEMÓRIA

Registradores podem armazenar algumas

São necessárias formas de:

- escrever um valor na memória em um registrador;
- escrever na memória um valor em um registrador.

registradores.

OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
g = h + A[8];
```


OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
g = h + A[8];
```

```
mov.w 16(R9), R5
```

```
add R6, R5
```

OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
g = h + A[8];
```

```
mov.w 16(R9), R5
```

```
add R6, R5
```

Leia o endereço guardado em R9, ande 16 bytes à frente, leia o valor armazenado neste endereço e guarde-o em R5

OPERAÇÕES ARITMÉTICAS

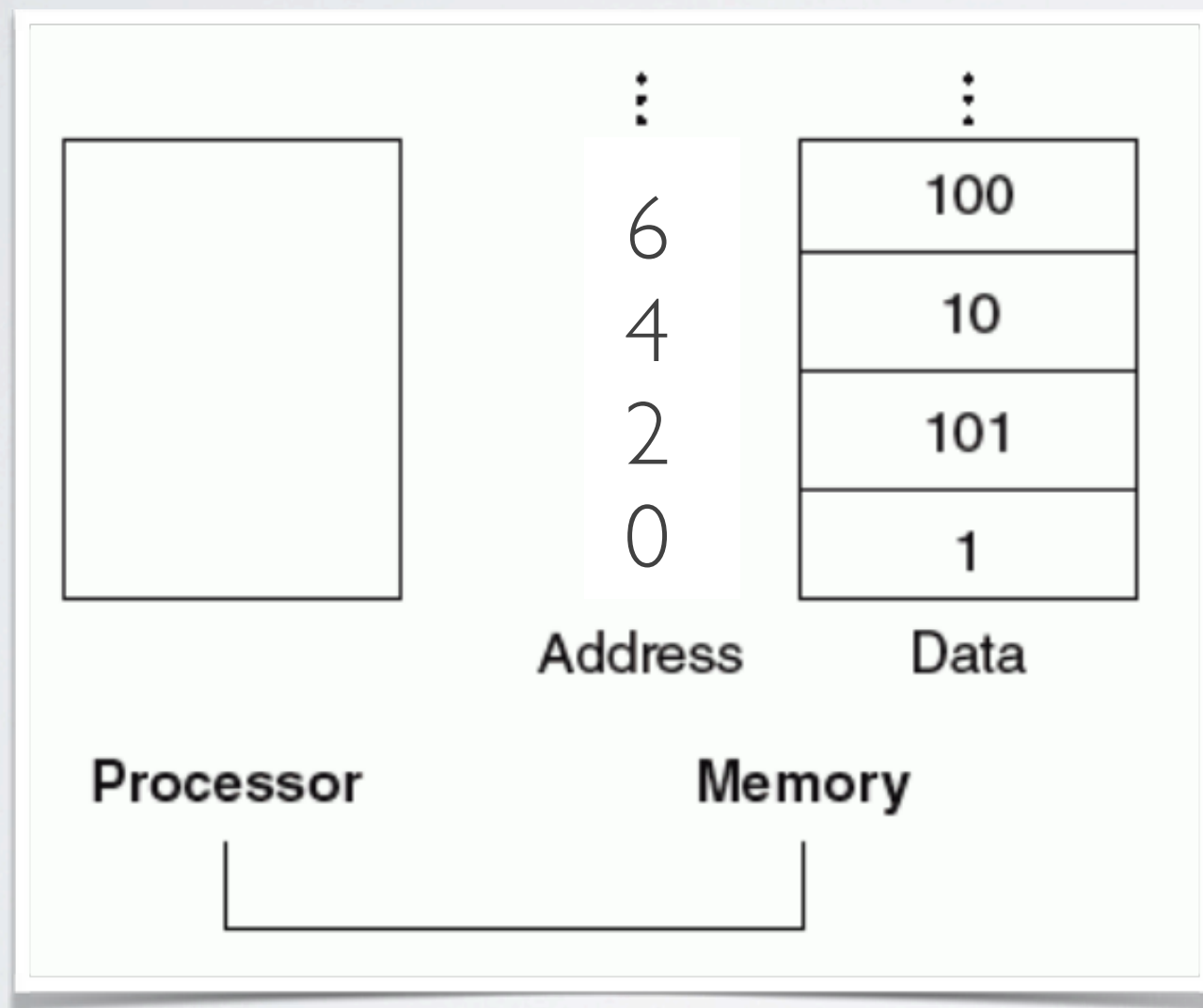
Como a seguinte linha em C é compilada para o MSP430?

```
g = h + A[8];
```

```
mov.w 16(R9), R5 ; R9 = &(A[0]) = A
```

```
add R6, R5 ; R5 = g, R6 = h
```

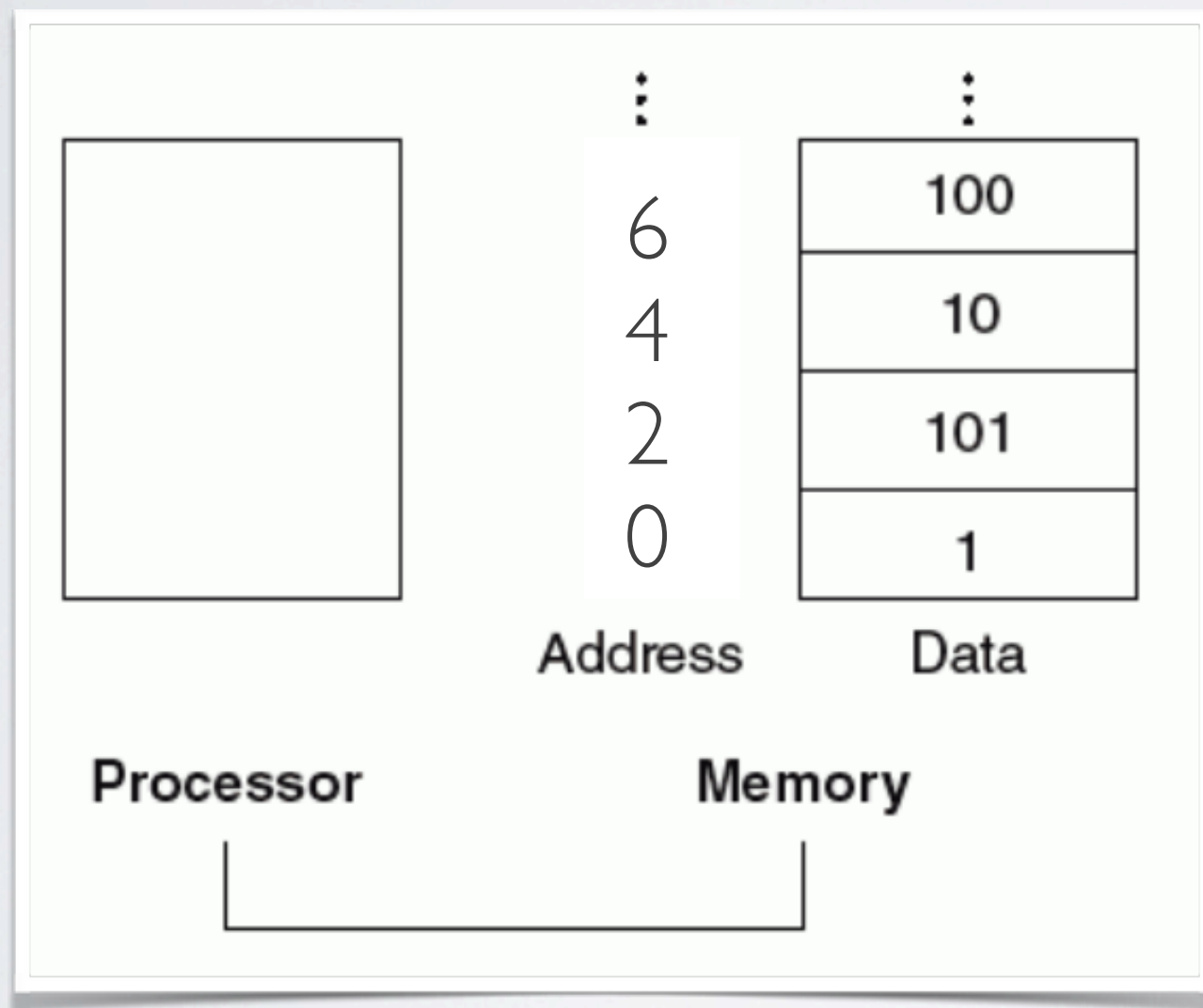

OPERANDOS NA MEMÓRIA



A palavra no MSP430 ocupa 2 bytes (16 bits).

Qual é a ordem destes bytes na memória?

OPERANDOS NA MEMÓRIA

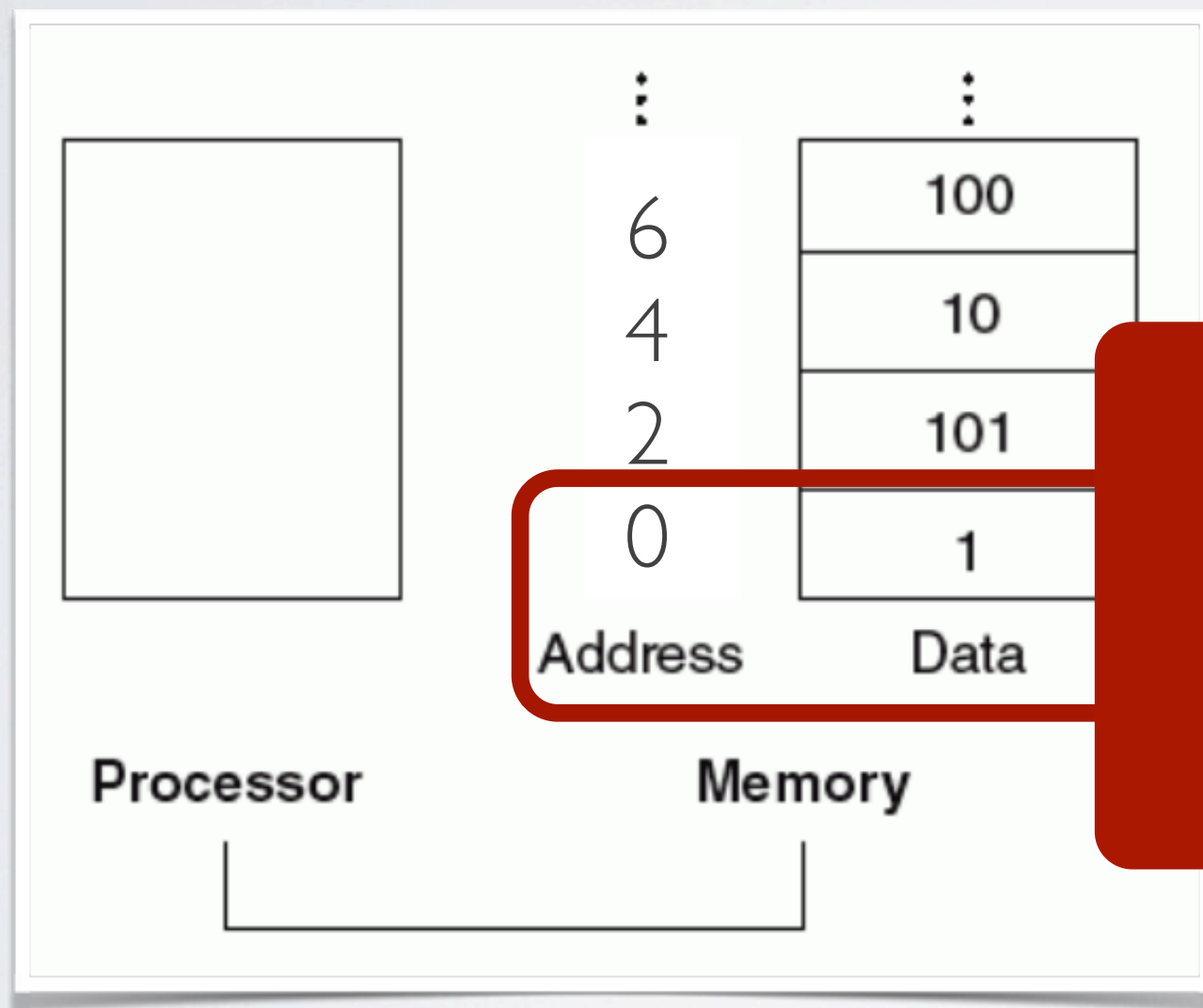


A palavra no MSP430 ocupa 2 bytes (16 bits).

Qual é a ordem destes bytes na memória?

No MSP430, o byte menos significativo fica na memória mais baixa.

OPERANDOS NA MEMÓRIA

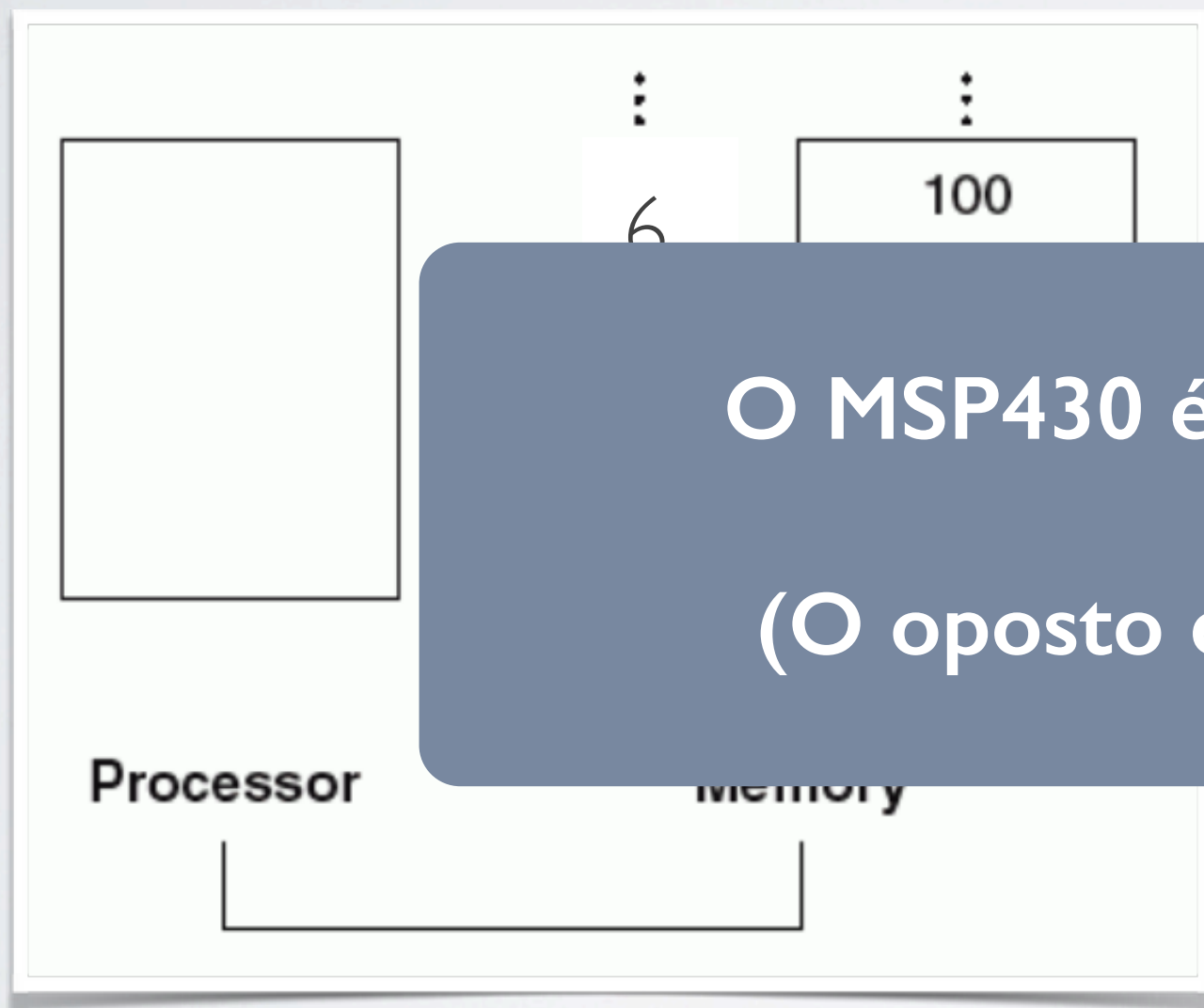


A palavra no MSP430 ocupa 2 bytes (16 bits).

O endereço 3 contém 0x0, o endereço 2 contém 0x0, o endereço 1 contém 0x0 e o endereço 0 contém 0x1.

mais baixa.

OPERANDOS NA MEMÓRIA



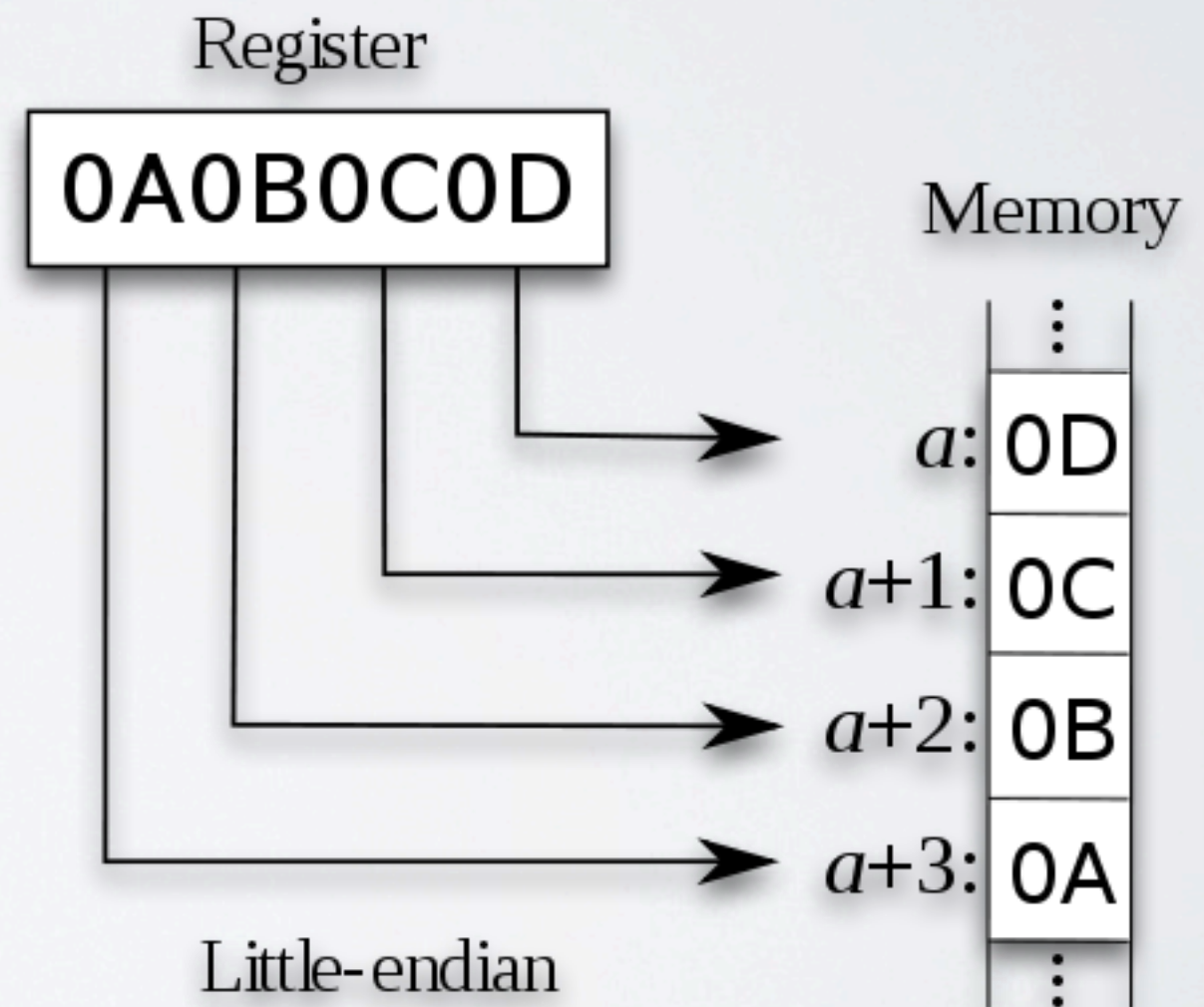
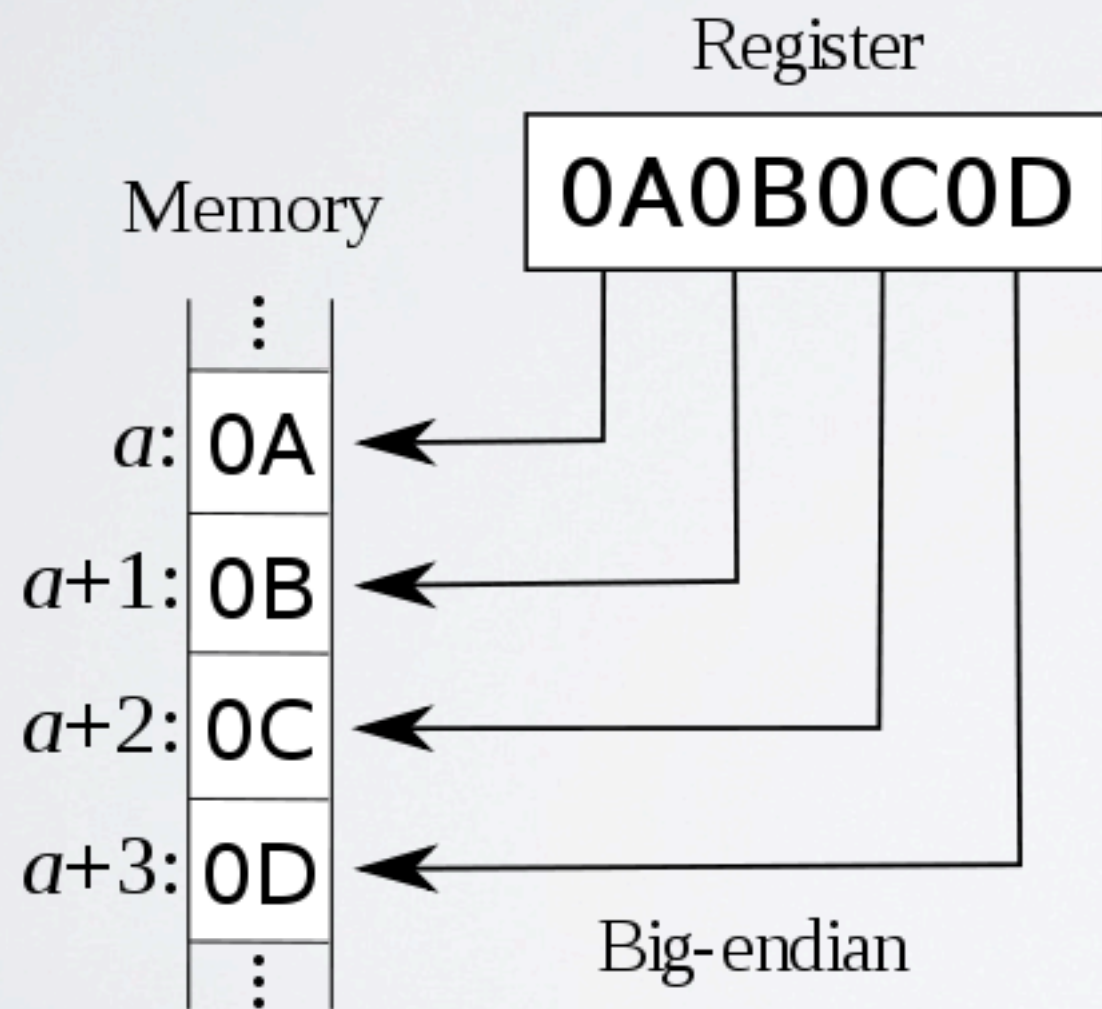
A palavra no MSP430 ocupa 2 bytes (16 bits).

○ MSP430 é Little-Endian.

(O oposto é Big-Endian.)

tes bytes
?
e menos
significativo fica na memória
mais baixa.

OPERANDOS NA MEMÓRIA



OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
A[12] = h + A[8];
```

```
mov.w 16(R9), R10  
add.w R6, R10  
mov.w R10, 24(R9)
```

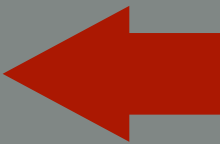

OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
A[12] = h + A[8];
```

```
mov.w 16(R9), R10  
add.w R6, R10  
mov.w R10, 24(R9)
```

**Leia o endereço guardado em R9,
ande 24 bytes à frente e guarde
neste endereço o valor armazenado
em R10**



OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
A[12] = h + A[8];
```

```
mov.w 16(R9), 24(R9)
```

```
add.w R6, 24(R9)
```

OPERAÇÕES ARITMÉTICAS

Como a seguinte linha em C é compilada para o MSP430?

```
A[12] = h + A[8];
```

```
mov.w 16(R9), 24(R9)
```

```
add.w R6, 24(R9)
```

Para economizar uma instrução, joga-se o valor de A[8] diretamente em A[12] antes de somar o valor em h (R6).

OPERANDOS CONSTANTES

Em muitos casos, uma soma é feita com valores constantes.
Por exemplo, as seguintes instruções em C:

```
i++;  
k--;
```

OPERANDOS CONSTANTES

Em muitos casos, uma soma é feita com valores constantes.
Por exemplo, as seguintes instruções em C:

```
i++;  
k--;
```

Operações usando valores constantes recebem instruções específicas:

clr.w	dst	; clear	dst = 0	emulated
dec.w	dst	; decrement	dst--	emulated
dec.d.w	dst	; double decrement	dst -= 2	emulated
inc.w	dst	; increment	dst++	emulated
inc.d.w	dst	; double increment	dst += 2	emulated
tst.w	dst	; test (compare with 0)	(dst - 0)	emulated

OPERANDOS CONSTANTES

Em muitos casos, uma soma é feita com valores constantes.
Por exemplo, as seguintes instruções em C:

`i++`

3. O caso comum deve ser mais rápido.

Operações específicas:

clr.w	dst	; clear	dst = 0	emulated
dec.w	dst	; decrement	dst--	emulated
dec.d.w	dst	; double decrement	dst -= 2	emulated
inc.w	dst	; increment	dst++	emulated
inc.d.w	dst	; double increment	dst += 2	emulated
tst.w	dst	; test (compare with 0)	(dst - 0)	emulated