

Elementos de Sistemas

Conteúdo 13 – Montagem de Computador

"Às vezes, são as pessoas de quem esperamos nada que fazem coisas que ninguém esperaria."

"Sometimes it is the people no one can imagine anything of who do the things no one can imagine."

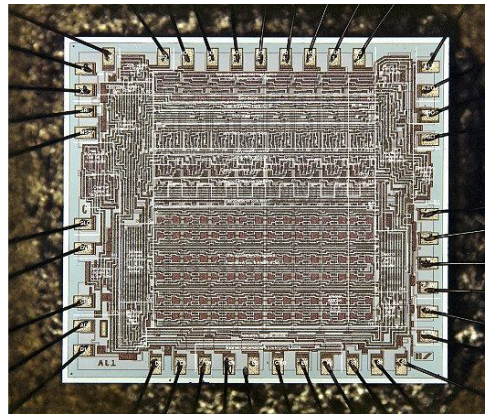
Alan Turing (1912–1954), Cientista da Computação Britânico

Primeiros Microprocessadores

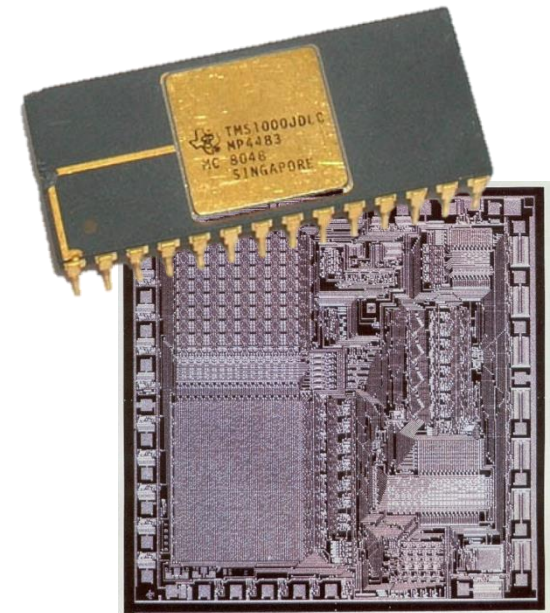
- Garrett AiResearch's Central Air Data Computer (CADC)
- Texas Instruments TMS 1000
- Four-Phase Systems AL1



CADC



AL1



TMS1000

Intel 4004 e MOS 6502

Intel 4004

Chips (MCS-4):

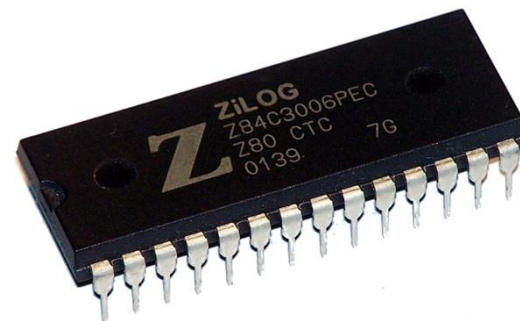
- ROM(4001)
- RAM(4002)
- Shift Register(4003)
- CPU(4004)



MOS 6502



Zilog 80

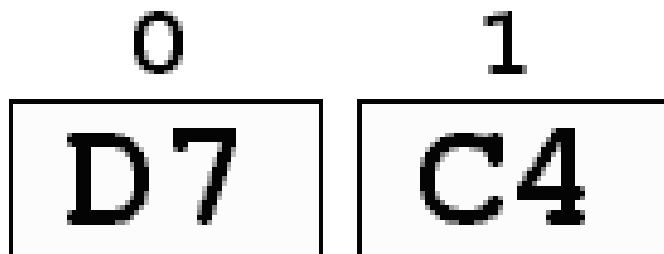


Little Endian vs BigEndian

Armazenando o valor $D7C4_{16}$

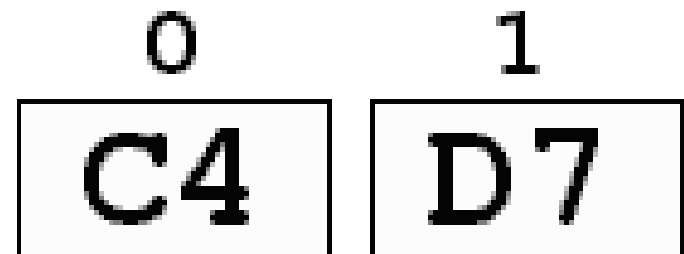
Big Endian

Motorola Processors:
68000, 68030, etc...



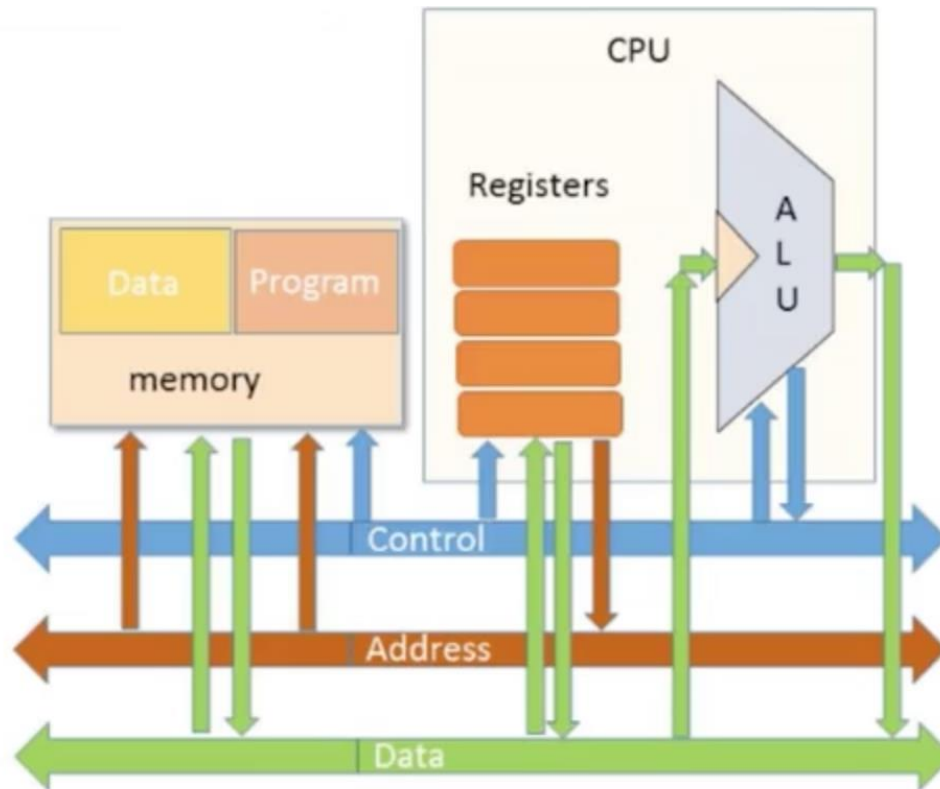
Little Endian

Intel Processors: 80386,
Pentium, etc...



Caminho de Dados (Datapath)

Elementos de Caminho de Dados



Tipos de registradores

Registradores de Dados:

Registradores usados para armazenarem dados temporários.

Registradores de Endereçamento:

Registradores usados para armazenar endereços de memórias a serem usados por instruções futuras.

Ciclo de clock

O que acontece:

FETCH: Le uma instrução da memória

DECODE: decodifica a instrução

READ: Le os operandos (registradores)

EXECUTE: realiza a operação

WRITE-BACK: grava o resultado da operação

NOP

No Operation

instrução que não faz nada na CPU

Como realizar tais operações na CPU

```
if a > 0:  
    b = 2  
else:  
    b = 3
```

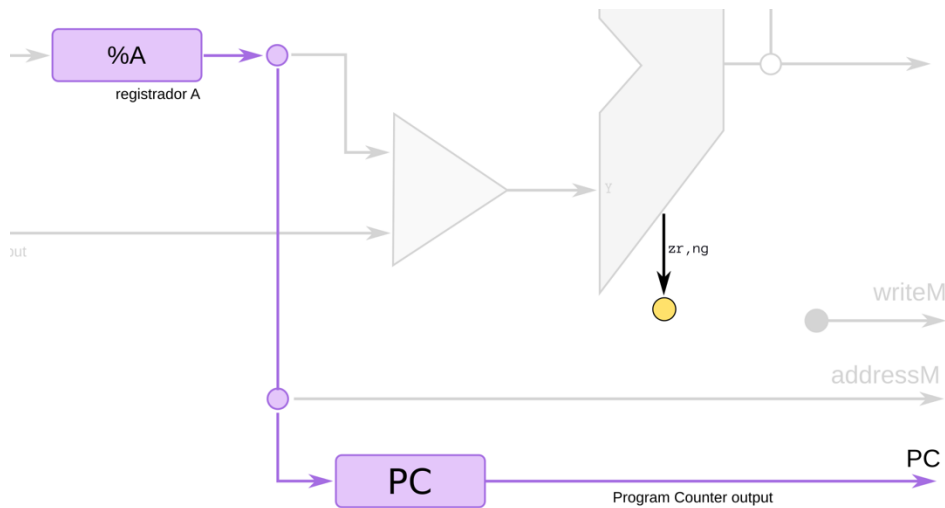
```
while True:  
    b = b+1
```

JUMP

Devido a instrução alterar o contador do Program Counter, maiores cuidados devem ser tomados.



Program Counter



0	1110111010010000
1	0100000000000000
2	1110001100001000
3	01000000000000010
4	1110001100001000
5	010000000000000100
6	1110001100001000
7	0000000000000000
8	111000000000000111
9	1110101010000000
10	1110111010010000
11	0100000000000000
12	1110001100001000
13	01000000000000010
...	...

Operações de Salto

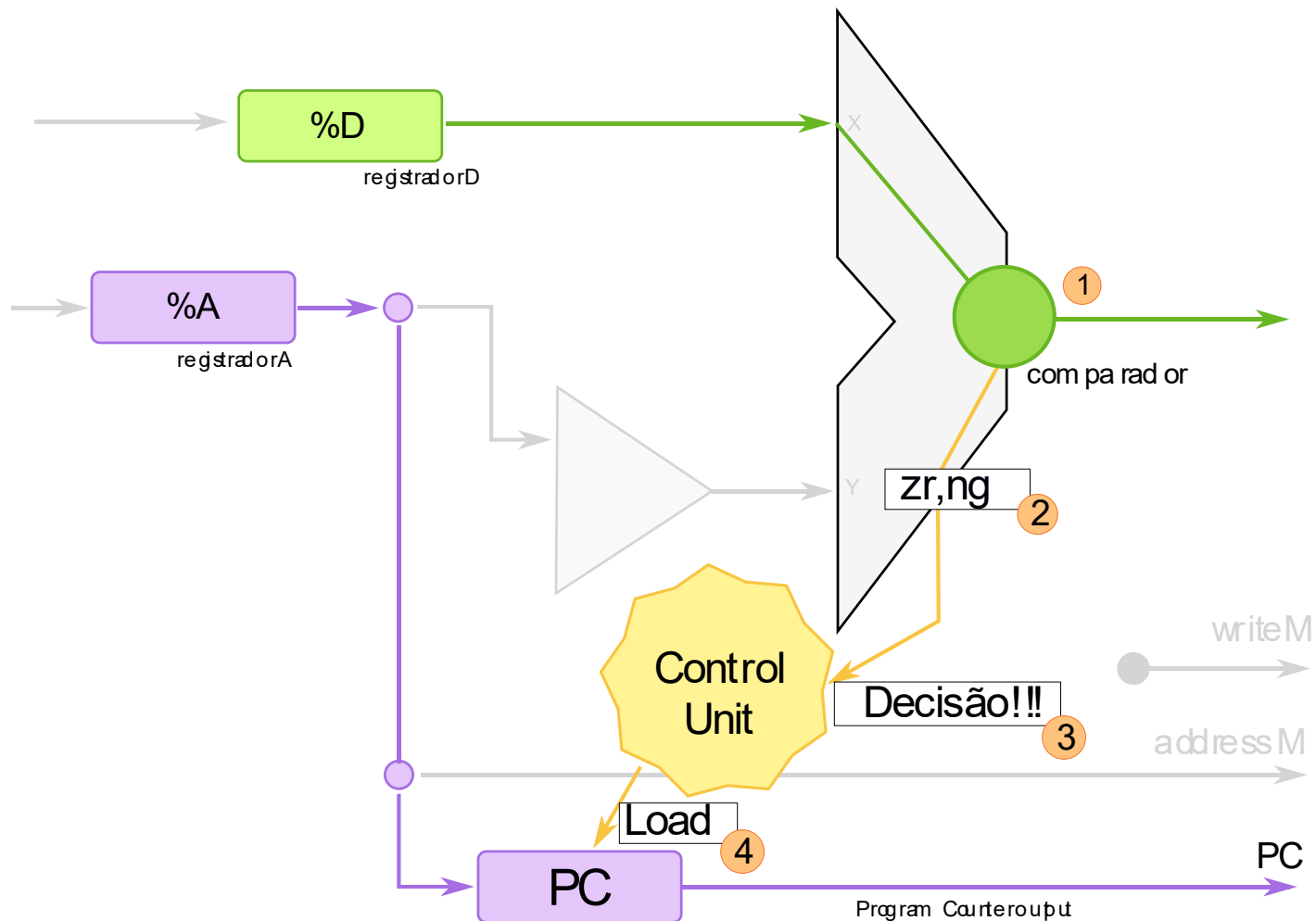
Salto incondicional

- ***jmp***

Salto condicional

- ***je %D***: Salta se valor de ***%D*** é igual a **0**
- ***jne %D***: Salta se valor de ***%D*** for diferente de **0**
- ***jg %D***: Salta se valor de ***%D*** for maior que **0**
- ***jl %D***: Salta se valor de ***%D*** for menor que **0**
- ***jge %D***: Salta se valor de ***%D*** for maior ou igual a **0**
- ***jle %D***: Salta se valor de ***%D*** for menor ou igual a **0**

Como funciona o Salto?



Exemplo

```
if RAM[1] > 0:
    RAM[2] = 1
else:
    RAM[2] = 2
```

```
leaw $1, %A ; faz %A apontar para RAM[1]
movw (%A), %D ; carrega o valor de RAM[1] em %D
leaw $ELSE, %A ; precisamos carregar em %A o valor do salto
jle %D ; salta se valor em %D for menor ou igual a zero
nop ;
; if
leaw $2, %A ;
movw $1, (%A) ;
;
leaw $END, %A ; agora não podemos executar o trecho
jmp ; do else, vamos pular para o fim
nop ; do código
;
ELSE: ; else
;
leaw $2, %A ;
movw %A, (%A) ;
;
END: ;
```

Insper

www.insper.edu.br