

# Arquitetura de Computadores

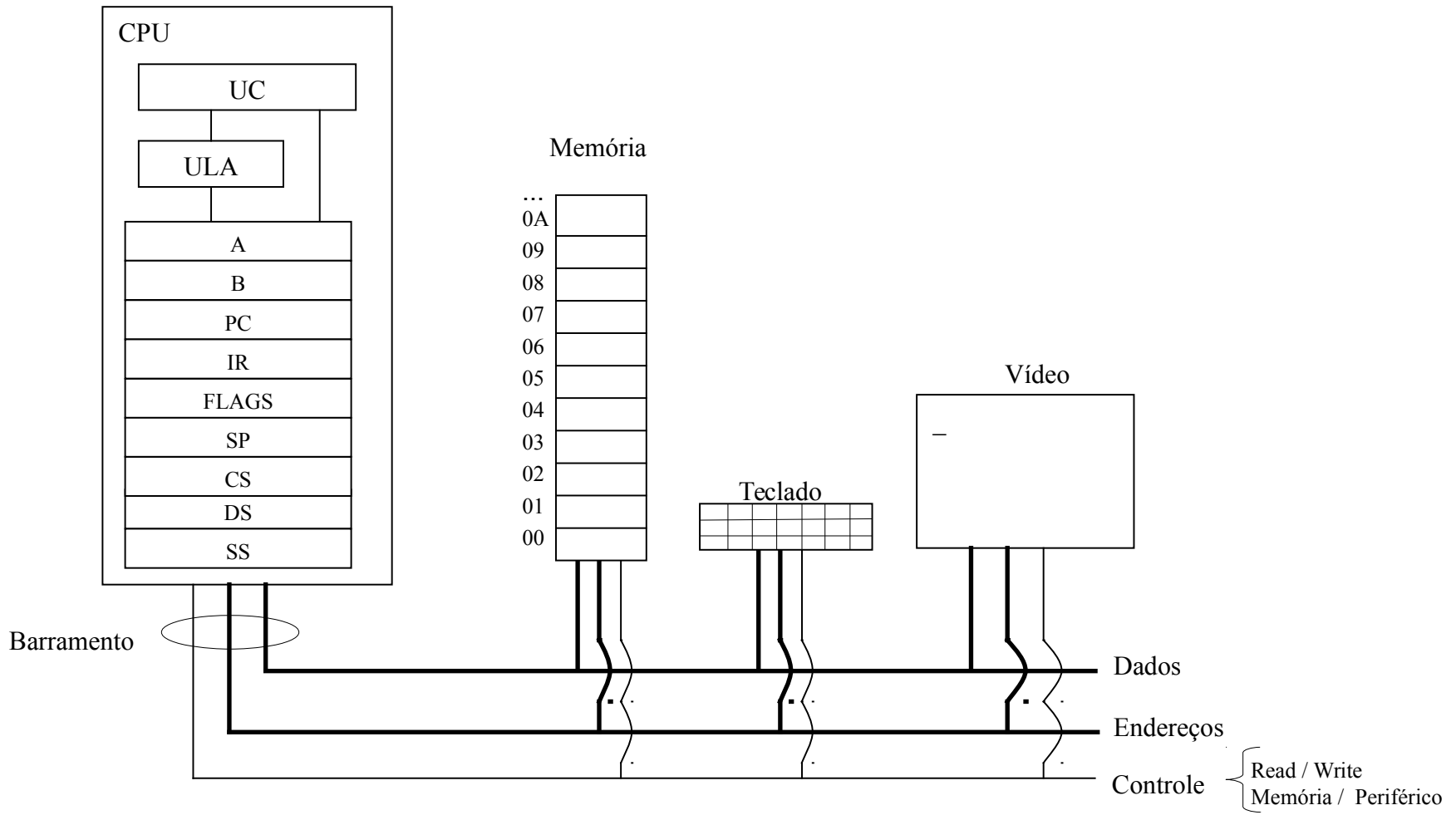
## *Conceitos Fundamentais*

Graduação em Engenharia Elétrica - UFPR

Prof. Carlos Marcelo Pedroso

2016

# Um Computador Muito Simples



# Configuração

- Cada posição de memória possui 8 bits, com memória total de 256 bytes (endereços de 0 a FF em hexadecimal).
- Todos os registradores da UCP (ou CPU) possuem 8 bits.
- O barramento de dados possui 8 vias e o barramento de endereços possui 8 vias.

# Configuração

- O computador possui dois periféricos:
  - um controlador de vídeo, localizado no endereço 01. Todas as informações escritas neste periférico terão o seu correspondente em ASCII impressos no vídeo.
    - Exemplo: caso seja escrito 01 hexa (binário 00000001) no barramento de endereços e 41 hexadecimal (binário 01000001, decimal 65) no barramento de dados , será impressa a letra 'A' no vídeo (neste caso a linha seletora de memória/periférico deverá estar setada com 1 e a linha seletora de leitura/escrita deverá estar com 1).
- um controlador de teclado, localizado no endereço 02.



# Configuração

- Segmentação de memória:
  - Registrador CS- Ponteiro para o segmento de código. Valor inicial: 00
  - Registrador DS- Ponteiro para o segmento de dados. Valor inicial: 7F
- Cada transferência entre a memória ou periférico e a CPU se completa em 1 ciclo de clock.

# Ciclo de busca e execução de instruções

- O valor inicial do PC deve ser 0.
- Sequência de busca de Instruções:
  1. A UC busca a próxima instrução da memória (valor apontado por PC) e transfere para o IR;
  2. A UC atualiza o contador de programa para que ele aponte para instrução seguinte ( $PC = PC + 1$ )
  3. A UC decodifica a instrução;
  4. Se a instrução utiliza dados da memória, determina onde eles estão (ex. instrução 005 faz referência a um endereço de memória). Neste caso, incrementa o contador de programa ( $PC = PC + 1$ )
  5. Executa a instrução

# Instruções Suportadas

CÓDIGO	MNEMÔNICO	DESCRIÇÃO
00	MOV A, VALOR	Transfere o conteúdo da posição de memória seguinte (endereço imediato) para o registrador A.
01	MOV B, VALOR	Transfere o conteúdo da posição de memória seguinte (endereço imediato) para o registrador B.
02	MOV A, B	Transfere o conteúdo de B para A.
03	MOV B, A	Transfere o conteúdo de A para B.
04	IN ENDEREÇO	Lê do endereço de periférico indicado e coloca o resultado em A (deve escrever 1 no seletor MEMÓRIA/PERIFÉRICO, o ENDEREÇO no barramento de endereços, e escrever 0 no seletor READ/WRITE).
05	OUT ENDEREÇO	Escreve o conteúdo do registrador A no endereço indicado (deve escrever 1 no seletor MEMÓRIA/PERIFÉRICO, o ENDEREÇO no barramento de endereços, e escrever 1 no seletor READ/WRITE)
06	MOV A, [ENDEREÇO]	Transfere o conteúdo da posição de memória apontada por DS+ENDEREÇO (endereço direto) para o registrador A.
07	MOV [ENDEREÇO], A	Transfere o conteúdo de A para a posição de memória apontada por DS+ENDEREÇO (endereço direto).
08	MOV B, [ENDEREÇO]	Transfere o conteúdo da posição de memória apontada por DS+ENDEREÇO (endereço direto) para o registrador B
09	MOV [ENDEREÇO], B	Transfere o conteúdo de B para a posição de memória apontada por DS+ENDEREÇO (endereço direto).



# Instruções Suportadas

CÓDIGO	MNEMÔNICO	DESCRIÇÃO
0A	JZ ENDEREÇO	Faz PC=ENDEREÇO caso o registrador FLAGS indique ZERO.
0B	JNZ ENDEREÇO	Faz PC=ENDEREÇO caso o registrador FLAGS NÃO indique ZERO.
0C	JP ENDEREÇO	Faz PC=ENDEREÇO caso o registrador FLAGS indique POSITIVO.
0D	JN ENDEREÇO	Faz PC=ENDEREÇO caso o registrador FLAGS indique NEGATIVO.
0E	JMP ENDEREÇO	Faz PC=ENDEREÇO
20	SUB A, VALOR	Realiza a operação VALOR-A, alterando o valor do registrador FLAGS.
21	ADD A, VALOR	Realiza a operação A+VALOR, alterando o valor do registrador FLAGS.
22	MUL A, VALOR	Realiza a operação A*VALOR, alterando o valor do registrador FLAGS.
23	DIV A, VALOR	Realiza a operação VALOR/A, alterando o valor do registrador FLAGS.
24	SUB A, B	Solicita à UCP para realizar a operação de subtração entre B e A (B-A), com resultado armazenado no registrador A. Esta operação também modifica o conteúdo registrador FLAGS.

# Instruções Suportadas

25	ADD A, B	Solicita à UCP para realizar a operação de adição entre B e A ( $A+B$ ), com resultado armazenado no registrador A.
26	MUL A, B	Solicita à UCP para realizar a operação de multiplicação entre B e A ( $B*A$ ), com resultado armazenado em A
27	DIV A, B	Solicita à UCP para realizar a operação de divisão entre B e A ( $B/A$ ), com resultado armazenado em A.
28	SUB A, [ENDEREÇO]	Realiza a operação $[ENDEREÇO]-A$ , alterando o valor do registrador FLAGS.
29	ADD A, [ENDEREÇO]	Realiza a operação $[DS+ENDEREÇO]+A$ , alterando o valor do registrador FLAGS.
2A	MUL A, [ENDEREÇO]	Realiza a operação $[DS+ENDEREÇO]*A$ , alterando o valor do registrador FLAGS.
2B	DIV A, [ENDEREÇO]	Realiza a operação $[DS+ENDEREÇO]/A$ , alterando o valor do registrador FLAGS.
2C	SUB [ENDEREÇO], A	Realiza a operação $A-[DS+ENDEREÇO]$ , alterando o valor do registrador FLAGS. O resultado será armazenado em $[DS+ENDEREÇO]$ .
2D	ADD [ENDEREÇO], A	Realiza a operação $A+[DS+ENDEREÇO]$ , alterando o valor do registrador FLAGS. O resultado será armazenado em $[DS+ENDEREÇO]$ .
2E	MUL [ENDEREÇO], A	Realiza a operação $A*[DS+ENDEREÇO]$ , alterando o valor do registrador FLAGS. O resultado será armazenado em $[DS+ENDEREÇO]$ .

# Instruções Suportadas

2F	MOV CS, A	Copia o valor do registrador A para CS
30	MOV DS, A	Copia o valor do registrador A para DS
31	MOV SS, A	Copia o valor do registrador A para SS
32	PUSH	Copia o valor de A para [SS+SP] e faz SP=SP+1
33	POP	Copia o valor de [SS+SP] para A e faz SP=SP-1
34	CALL END	Empilha o endereço da próxima instrução, na posição [SS+SP] e faz PC=END e SP=SP+1
35	RET	Faz PC=[SS+SP] e SP=SP-1
36	MOV A, SS:[SP-VALOR]	Copia para A o conteúdo da posição de memória de SS+SP-VALOR

# Operações Aritméticas

- As operações realizadas pela ULA afetam o registrador FLAGS. A interpretação da informação contida neste registrador é binária, com o seguinte formato:

