

Neste "processador" simplificado os endereços de memória são compostos por 4 bits, enquanto os valores são compostos de 16 bits. O processador é composto da ULA e dois registradores A e B.

Lista de instruções:

0000	LOAD A
0001	LOAD B
0010	SUM
0011	WRITE
0100	JUMP IF LESS
0101	OUT
0110	HALT

LOAD A: Carregar o valor do endereço para o registrador A. ex: 0000 0000 ****
**** copia o valor do endereço 0000 para o registrador A.

LOAD B: Carregar o valor do endereço para o registrador A. ex: 0001 0001 ****
**** copia o valor do endereço 0001 para o registrador B.

SUM: Soma os dois valores dentro dos registradores A e B e guarda o resultado no B, sobrescrevendo o valor anterior. ex: A=0001 e B=0010. A instrução 0010 ****
**** **** irá somar A e B. No final A=0001 e B=0011.

WRITE: Salva na memória o valor guardado em um registrador. Ex: 0011 0000 1111 ****, irá copiar o valor do registrador A no endereço de memória 1111. Use 0000 para o registrador A e 0001 para o registrador B.

JUMP IF LESS: Caso o valor encontrado no endereço passado seja menor que o valor no segundo endereço, pula a execução do programa para o terceiro endereço. Ex: 0100 0000 0001 0011, se o valor em 0000 for menor que o valor 0001, o programa irá executar o comando presente no endereço 0011, ao invés do próximo.

OUT: Envia o valor contido em um registrador para um dispositivo de saída. Ex: 0101 0000 **** **** envia o valor que está no registrador A para o dispositivo de saída.

HALT: encerra o programa.

As tabelas abaixo representam o conteúdo da memória principal carregada no computador.

Exemplo 01:

0	0000	0000 1101 0000 0000
1	0001	0001 1111 0000 0000
2	0010	0101 0001 0000 0000
3	0011	0010 0000 0000 0000
4	0100	0011 0001 1111 0000
5	0101	0100 1111 1110 0000
6	0110	0110 0000 0000 0000
7	0111	0000 0000 0000 0000
8	1000	0000 0000 0000 0000
9	1001	0000 0000 0000 0000
10	1010	0000 0000 0000 0000
11	1011	0000 0000 0000 0000
12	1100	0000 0000 0000 0000
13	1101	0000 0000 0000 0001
14	1110	0000 0000 0000 1010
15	1111	0000 0000 0000 0000

Exercício 01:

0	0000	0000 1001 0000 0000
1	0001	0001 1001 0000 0000
2	0010	0010 0000 0000 0000
3	0011	0011 0001 1010 0000
4	0100	0100 1011 1001 0000
5	0101	0001 1010 0000 0000
6	0110	0101 0001 0000 0000
7	0111	0110 0000 0000 0000
8	1000	0000 0000 0000 0000
9	1001	0000 0000 0000 0001
10	1010	0000 0000 0000 0000
11	1011	0000 0000 0000 0000
12	1100	0000 0000 0000 0000

13	1101	0000 0000 0000 0000
14	1110	0000 0000 0000 0000
15	1111	0000 0000 0000 0000

Exercício 02:

0	0000	0000 1101 0000 0000
1	0001	0001 1100 0000 0000
2	0010	0010 0000 0000 0000
3	0011	0011 0001 1100 0000
4	0100	0000 1011 0000 0000
5	0101	0001 1111 0000 0000
6	0110	0010 0000 0000 0000
7	0111	0011 0001 1111 0000
8	1000	0100 1111 1110 0000
9	1001	0001 1100 0000 0000
10	1010	0101 0001 0000 0000
11	1011	0010 0000 0000 0001
12	1100	0000 0000 0000 0000
13	1101	0000 0000 0000 0010
14	1110	0000 0000 0000 0011
15	1111	0000 0000 0000 0000

Perguntas:

- 1) Quais as saídas dos programas?
- 2) O que faz cada programa?
- 3) Considere que as operações SUM, JUMP IF LESS e HALT duram um valor x qualquer. Já as operações que envolvem acessar valores fora da CPU: LOAD A, LOAD B, WRITE e OUT duram 1000x, ou seja mil vezes mais. Quanto tempo, em x, demoram para rodar os programas do Exemplo 1 e Exercício 2?
- 4) Caso a memória principal ficasse dentro do processador, quais seriam os tempos, em x, que os mesmos dois programas levariam para executar?

Neste "processador" simplificado os endereços de memória são compostos por 4 bits, enquanto os valores são compostos de 16 bits. O processador é composto da ULA e dois registradores A e B.

Lista de instruções:

0000	LOAD A
0001	LOAD B
0010	SUM
0011	WRITE
0100	JUMP IF LESS
0101	OUT
0110	HALT

LOAD A: Carregar o valor do endereço para o registrador A. ex: 0000 0000 **** copia o valor do endereço 0000 para o registrador A.

LOAD B: Carregar o valor do endereço para o registrador A. ex: 0001 0001 **** copia o valor do endereço 0001 para o registrador B.

SUM: Soma os dois valores dentro dos registradores A e B e guarda o resultado no B, sobrescrevendo o valor anterior. ex: A=0001 e B=0010. A instrução 0010 **** irá somar A e B. No final A=0001 e B=0011.

WRITE: Salva na memória o valor guardado em um registrador. Ex: 0011 0000 1111 ****, irá copiar o valor do registrador A no endereço de memória 1111. Use 0000 para o registrador A e 0001 para o registrador B.

JUMP IF LESS: Caso o valor encontrado no endereço passado seja menor que o valor no segundo endereço, pula a execução do programa para o terceiro endereço. Ex: 0100 0000 0001 0011, se o valor em 0000 for menor que o valor 0001, o programa irá executar o comando presente no endereço 0011, ao invés do próximo.

OUT: Envia o valor contido em um registrador para um dispositivo de saída. Ex: 0101 0000 **** envia o valor que está no registrador A para o dispositivo de saída.

HALT: encerra o programa.

As tabelas abaixo representam o conteúdo da memória principal carregada no computador.

Exemplo 01:

0	0000	0000 1101 0000 0000 LOAD A I
1	0001	0001 1111 0000 0000 LOAD B C
2	0010	0101 0001 0000 0000 OUT B

3	0011	0010 0000 0000 0000 SUM
4	0100	0011 0001 1111 0000 WRITE B C
5	0101	JUMP IF LESS C X 0000
6	0110	HALT 0110 0000 0000 0000
7	0111	0000 0000 0000 0000
8	1000	0000 0000 0000 0000
9	1001	0000 0000 0000 0000
10	1010	0000 0000 0000 0000
11	1011	0000 0000 0000 0000
12	1100	0000 0000 0000 0000
13	1101	0000 0000 0000 0001 I = 1
14	1110	0000 0000 0000 1010 X = 10
15	1111	0000 0000 0000 0000 C = 0

Exercício 01:

0	0000	0000 1001 0000 0000 LOAD A 9
1	0001	0001 1001 0000 0000 LOAD B 9
2	0010	0010 0000 0000 0000 SUM
3	0011	WRITE B 10 0011 0001 1010 0000
4	0100	0100 1011 1001 0000 JUMP IF LESS 11 9 0000
5	0101	LOAD B 10 0001 1010 0000 0000
6	0110	OUT B 0101 0001 0000 0000
7	0111	HALT
8	1000	0000 0000 0000 0000
9	1001	0000 0000 0000 0001
10	1010	0000 0000 0000 0000
11	1011	0000 0000 0000 0000
12	1100	0000 0000 0000 0000
13	1101	0000 0000 0000 0000

14	1110	0000 0000 0000 0000
15	1111	0000 0000 0000 0000

Exercício 02:

0	0000	0000 1101 0000 0000 LOAD A X
1	0001	0001 1100 0000 0000 LOAD B R
2	0010	010 0000 0000 0000 SUM
3	0011	0011 0001 1100 WRITE B R
4	0100	0000 1011 0000 0000 LOAD A I
5	0101	0001 1111 0000 0000 LOAD B C
6	0110	010 0000 0000 0000 SUM
7	0111	0011 0001 1111 0000 WRITE B C
8	1000	0100 1111 1110 0000 JUMP IF LESS C Y 0000
9	1001	0001 1100 0000 0000 LOAD B R
10	1010	0101 0001 0000 0000 OUT B
11	1011	I = 1 0010 0000 0000 0001
12	1100	R = 0 0000 0000 0000 0000
13	1101	X = 2 0000 0000 0000 0010
14	1110	Y = 3 0000 0000 0000 0011
15	1111	C = 0 0000 0000 0000 0000