

Processador

Equipe:

- Guilherme de Menezes Furtado
- Murilo Vinicius Almeida Pinheiro

Circuito multiplicador

- Implementa multiplicação em um passo
- Toma a posição da operação ~b na alu
- Circuito do logisim presente no .zip

Estrutura do processador

Registradores

- MDR
- MAR
- PC
- MBR
- V
- W
- X
- Y
- H

Operações da ALU

- a
- b
- $\neg a$
- $a \times b$
- $a + b$
- $a + b + 1$
- $a + 1$
- $b + 1$
- $b - a$
- $b - 1$
- $-a$
- $a \wedge b$
- $a \vee b$
- 0
- 1
- -1

Palavra de instrução (34 bits)

ADR	J (Salto)	SHIFT	ALU	C (Escrever Reg)	M (Pega da Mem)	B (Ler Reg)
-----	-----------	-------	-----	------------------	-----------------	-------------

ADR	J (Salto)	SHIFT	ALU	C (Escrever Reg)	M (Pega da Mem)	B (Ler Reg)
000000000	000	00	000000	00000000	000	000

ADR

J

- 000 → GO TO MPC
- 001 → GO TO MPC + 256 IF resultado = 0
- 010 → GO TO MPC + 256 IF resultado ≠ 0
- 100 → GO TO MBR

SHIFT

- 01 → resultado << 1
- 10 → resultado >> 1
- 11 → resultado << 8

ALU

- 011000 → a
- 010100 → b
- 011010 → ~a
- 101100 → a * b
- 111100 → a + b
- 111001 → a + b + 1
- 111001 → a + 1
- 110101 → b + 1
- 111111 → b - a
- 110110 → b - 1
- 111011 → -a
- 001100 → a & b
- 011100 → a | b
- 010000 → 0
- 110001 → 1
- 110010 → -1

C

- 10000000 → MAR
- 01000000 → MDR
- 00100000 → PC
- 00010000 → X
- 00001000 → Y
- 00000100 → W
- 00000010 → V
- 00000001 → H

M

- 001 → Lê 1 Byte da Memória, endereço PC e coloca no MBR
- 010 → Lê 1 Word(4 Bytes) da Memória, endereço MAR e coloca no MDR
- 100 → Escreve 1 Word na Memória, mem[MAR] recebe MDR

B

- 000 → MDR
- 001 → PC
- 010 → MBR
- 011 → X
- 100 → Y
- 101 → W
- 110 → V

Funções do microprograma/assembly

Assembly	Posição no firmware	Entrada	Operação
----------	---------------------	---------	----------

Assembly	Posição no firmware	Entrada	Operação
Sem assembly	0	Nada	$PC \leftarrow PC + 1$; fetch; GO TO MBR
add x, y	2	x (sempre esta letra) y = endereço de memória	$X \leftarrow X + \text{mem}[y]$
mov y	6	y = endereço de memória	$\text{mem}[y] \leftarrow X$
goto y	9	y = endereço de memória	$PC \leftarrow \text{mem}[y]$; GO TO mem[y]
jz x, y	11	x (sempre esta letra) y = endereço de memória	Se $X = 0$: GO TO 9 Senão: $PC \leftarrow PC + 1$; GO TO 0
sub x, y	13	x (sempre esta letra) y = endereço de memória	$X \leftarrow X - \text{mem}[y]$
min x, y	25	x (sempre esta letra) y = endereço de memória	$\text{mem}[1] \leftarrow X$, se $X > \text{mem}[y]$ $\text{mem}[1] \leftarrow \text{mem}[y]$, caso contrário
div x, y	35	x (sempre esta letra) y = endereço de memória	$\text{mem}[1] \leftarrow \text{mem}[\text{adress}] \% X$
z	52	Nada	$X \leftarrow 0$
u	53	Nada	$X \leftarrow 1$
sqr y	64	y = endereço de memória	$\text{mem}[1] \leftarrow \lfloor \sqrt{\text{mem}[y]} \rfloor$
and x, y	81	x (sempre esta letra) y = endereço de memória	$X \leftarrow X \wedge \text{mem}[\text{adr}]$
prim y	86	y (sempre esta letra)	$X \leftarrow Y$; se Y é primo $X \leftarrow 0$; caso contrário
rcv a, b	103 106 109 114	a = nome do registrador (x, y, pc ou w) b = endereço de memória	$a \leftarrow \text{mem}[b]$
inc	117	Nada	MDR e $Y \leftarrow Y + 1$
dec	121	Nada	$Y \leftarrow Y - 1$
halt	255	Nada	Para o programa

Assembly

questao1.asm

- Escreve na word 1 o valor 1 se o número na word 1 corresponder a um ano bissexto
- Entrada no assembly:

```
goto inicio
wb 0

e ww 2513 <- escreve o número na word 1
a ww 400
b ww 100
c ww 3
d ww 0

inicio rcv x, e
mov x, d
rcv x, a
div x, d
rcv x, e
jz x, bis
```

```
rcv x, b
div x, d
rcv x, e
jz x, nbis
```

```
rcv x, c
and x, d
jz x, bis
```

```
nbis z
mov x, e
halt
```

```
bis u
mov x, e
halt
```

- Basta substituir 2513 pelo valor que se deseja.

questao2.asm

- Escreve recebe um número n da word 1 e escreve na word 1 o n-ésimo número primo.
- Entrada no assembly:

```
goto main
wb 0
r ww 100      <- escreve o número na word 1
m ww 1
n ww 0
main rcv y, m
    inc
    rcv x, r
    sub x, m
    mov x, n
jz x, fdois
loop jz x, final
    inc
    mov y, r
    sqr r
    rcv w, r
    prim y
    jz x, pulo
    rcv x, n
    sub x, m
    mov x, n
pulo inc
rcv x, n
goto loop
final mov y, r
rcv x, r
sub x, m
mov x, r
    halt
fdois mov y, r
halt
```

- Basta substituir 100 pelo valor que se deseja.

questao3.asm

- Recebe 3 números, um na word 1, outro na word 2 e o último na word 3, e escreve na word 1 o menor entre eles.
- Entradas no assembly:

```

goto main
wb 0

a ww 22    <- primeiro numero
b ww 11    <- segundo numero
c ww 99    <- terceiro numero

main add x, a
      min x, b
      z
      add x, a
      min x, c
      halt

```

- Basta substituir 22, 11 e 99 pelos valores que se deseja.

questao4.asm

- Recebe um número n na word 1 e escreve na word 1 o piso da raiz quadrada de n.
- Entrada no assembly:

```

goto main
wb 0

a ww 5930  <- escreve o número na word 1

main sqr a
      halt

```

- Basta substituir 5930 pelo valor que se deseja