Entrega Relógio Design de Computadores

Integrantes:

Rafael dos Santos
João Pedro Junqueira Meirelles
Thomas de Queiroz Barros Schneider

Manual de utilização do relógio

SW0 muda entre modo normal e modo rapido

SW1 muda entre contagem regressiva e contagem normal KEY3 sai da configuração do tempo

KEY2 incrementa a hora em 1

KEY1 incrementa o minuto em 1

KEY0 incrementa o segundo em 1

LEDs9 e 8 indicam se é AM ou PM

Arquitetura do processador

Registrador registrador - alta eficiência para um projeto pequeno e simples que não vai necessitar uma grande quantidade de informações armazenadas em memória. Para contar o tempo não necessários poucos bytes de informações que cabem nos registradores.

Total de instruções e sua sintaxe

Operações aritméticas:

Operacao	Aritmetica
add RX, RY, RZ	(RX = RY + RZ)
sub RX, RY, RZ	(RX = RY - RZ)
and RX, RY, RZ	(RX = RY & RZ)
or RX, RY, RZ	(RX = RY | RZ)
not RX, RY	(RX = !RY)
inc RX, RY	(RX = RY + 1)
dec RX, RY	(RX = RY - 1)

Cmp:

cmp RY, RZ (F0 = (RY == RZ), FNEG = (RZ > RY))

Nop:

nop

Limpa Base Tempo:

rst

Manipulação de dados:

ld RX, $Numero (RX = mem(Numero))

st $Numero, RX (mem(Numero) = RX)

lea RX, $Numero (RX = Numero)

Jumps: (Endereço absoluto)

jmp label (Pula para a endereço da ROM)

jle/jl/jg/jge/je/jne label (Pula para o endereço da ROM se condição for real)

Formato das instruções (distribuição dos campos na palavra de instrução)

Operações aritméticas entre registradores (3 argumentos)

Opcode					**RX**			**RY**			**RZ**			**Padding**	

Opcode: 5 bits

Endereçamento de até 3 registradores: 9 bits (8 registradores no total, 3 bits para representar 0-7)

Padding: 2 bits para manter o tamanho da instrução fixo

cmp e operações aritméticas entre registradores (2 argumentos)

Opcode					RX			RY			Padding				

Opcode: 5 bits

Endereçamento de 2 registradores: 6 bits (8 registradores no total, 3 bits para representar 0-7)

Padding: 5 bits para manter o tamanho da instrução fixo

nop e rst

Opcode					Padding										

Opcode: 5 bits

Padding: 11 bits para manter o tamanho da instrução fixo

Jumps

Opcode					Endereço										Pad

Opcode: 5 bits

Endereço da rom para o jump: 10 bits (1024 bytes é o tamanho da ROM)

Padding: 1 bit para manter o tamanho da instrução fixo

st

Opcode					RX			Address							

Opcode: 5 bits

Endereçamento do registrador: 3 bits (8 registradores no total, 3 bits para representar 0-7)

Endereço da RAM a ser carregado com os conteúdos registrador 0: 8 bits

ld

Opcode					RX			Address							

Opcode: 5 bits

Endereçamento do registrador: 3 bits (8 registradores no total, 3 bits para representar 0-7)

Endereço da RAM a ser carregado no registrador 0: 8 bits

lea

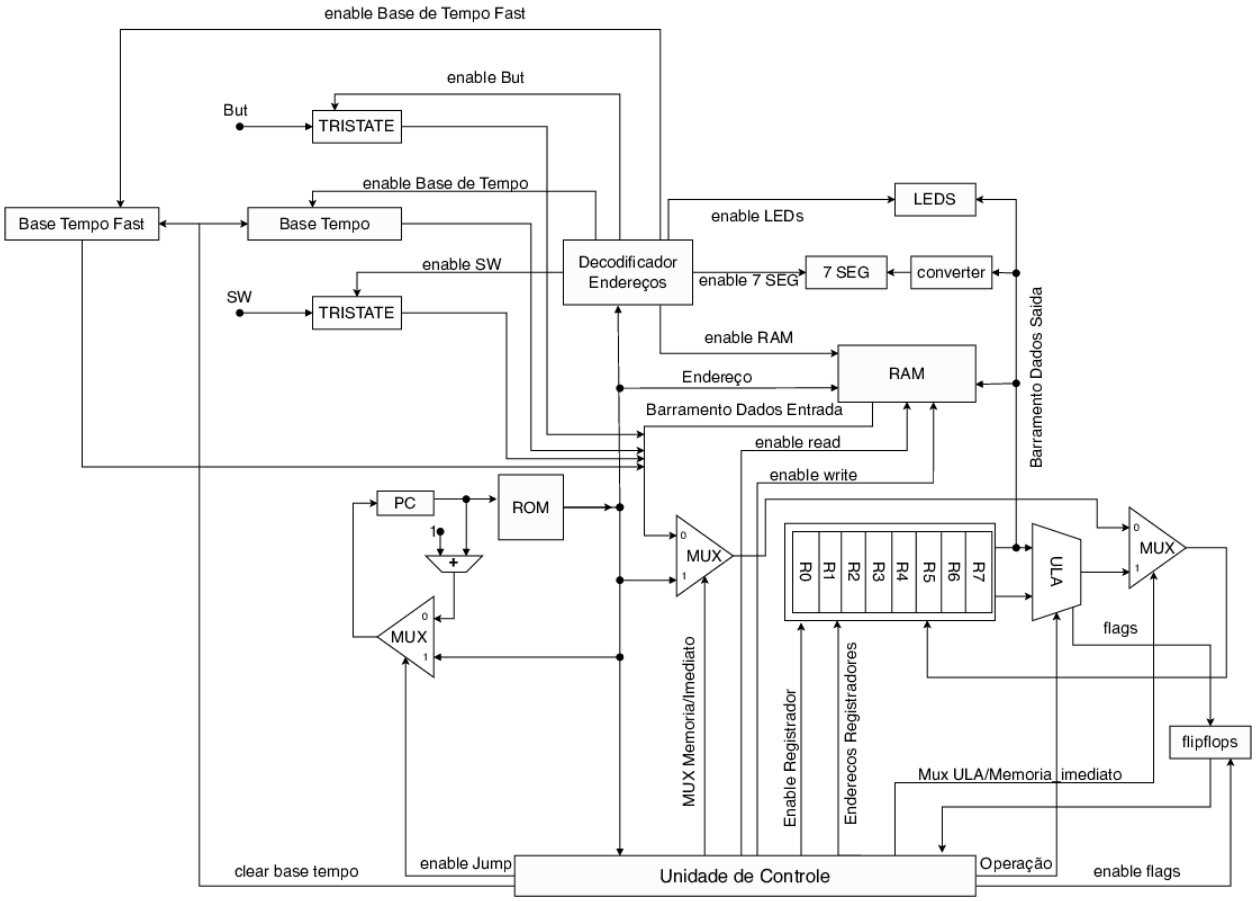
Opcode					RX			Número							

Opcode: 5 bits

Endereçamento do registrador: 3 bits (8 registradores no total, 3 bits para representar 0-7)

Número a ser carregado no registrador 0: 8 bits

Fluxo de dados do processador



Listagem dos pontos de controle e sua utilização

enable Jump	Escolhe se o PC vai avançar em 1 ou se ele vai dar um jump, sendo que para avançar em 1 o seletor fica 0 e para dar um jump o seletor fica em 1
enable Registrador	Escolhe quais registradores devem receber um valor novo
Operacao	Reprograma a ULA para a operação que desejamos realizar
MUX Memoria/Imediato	Responsável por decidir se o valor que vai para o MUX ULA/Memoria_imediato vem do barramento de memória ou da ROM
MUX ULA/Memoria_imediato	Responsável por decidir se o valor que vai para os registradores vem da ULA ou da Memória/Imediato

Mapa de memória

	Inicio	Fim	Tamanho
RAM	0	220	221
BASE DE TEMPO	221	221	1
LED0	222	222	1
LED1	223	223	1
SW0	224	224	1
SW1	225	225	1
DISPLAY 0/1	226	226	1
DISPLAY 2/3	227	227	1
DISPLAY 4/5	228	228	1
BOTÕES	229	229	1

BASE DE TEMPO FAST	230	230	1
RESERVADO	231	255	25

A ROM tem tamanho 1024 mas não se encontra no mapa de memória já que não há a necessidade de acessar seus valores diretamente e carregá-los em algum registrador. Ela também é muito maior que o endereçamento disponível pelos registradores de 8 bits (255). Os jumps sempre são absolutos e tem tamanho de 10 bits, eliminando a necessidade de utilizar registradores para comunicar com o PC/ROM.

Assembler

Como utilizar: ./assembler.py -o output.mif input.asm

Ajuda de utilização: ./assembler.py -h

Sintaxe intel

Instruções add, sub, and, or podem receber 2 ou 3 registradores (“add R0, R1” vai ser automaticamente convertido para “add R0, R1, R1”)

Numeros utilizados nos comandos ld, st e lea tem prefixo $

Labels são definidos por uma linha com uma palavra (o label) seguida imediatamente de “:”