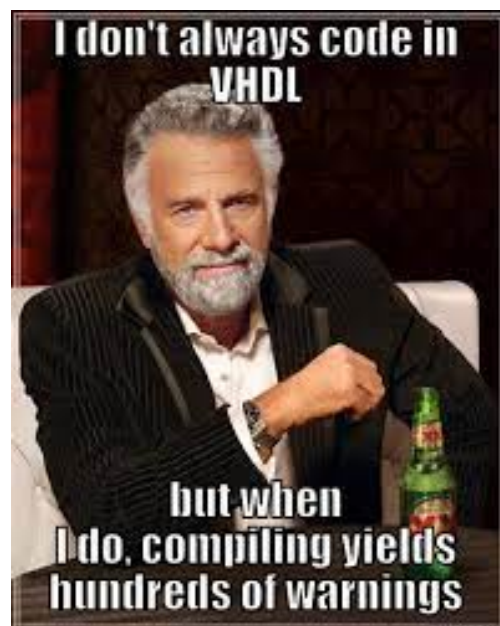


Relatório Design de Computadores

Projeto 1 - Relógio



Paulo Kim

Raphael Lahiry

Rodrigo Guimarães Coelho

Introdução

Neste relatório vamos descrever um pouco o processador que criamos, sua interação com periféricos, suas palavras de comando e outras descrições importantes para o entendimento do projeto como um todo.

Modo de uso

Ao iniciar a placa, o usuário fica no **loop principal** no qual ele pode seguir por dois caminhos:

- **KEY_0**: Acelerar a passagem de tempo
- **KEY_1**: Definir o horário

Caso o usuário escolha definir um horário, o display inteiro do 7SEG irá zerar e o LEDR0 irá acender, indicando que o primeiro valor a ser alterado será o das unidades do segundo. Manipulando os **switches de 0 a 3** utilizando uma lógica binária, podemos escolher um valor de 0 a 9 e então pressionar novamente o **KEY_0** para definir o valor das unidades. Com isso, o LEDR0 irá apagar e o LEDR1 irá acender, indicando que agora o valor a ser alterado é o das dezenas e assim por diante.

Por fim, ao definir o valor da dezena de hora, os LEDs de 7 a 0 irão acender indicando que a configuração do horário está completa.

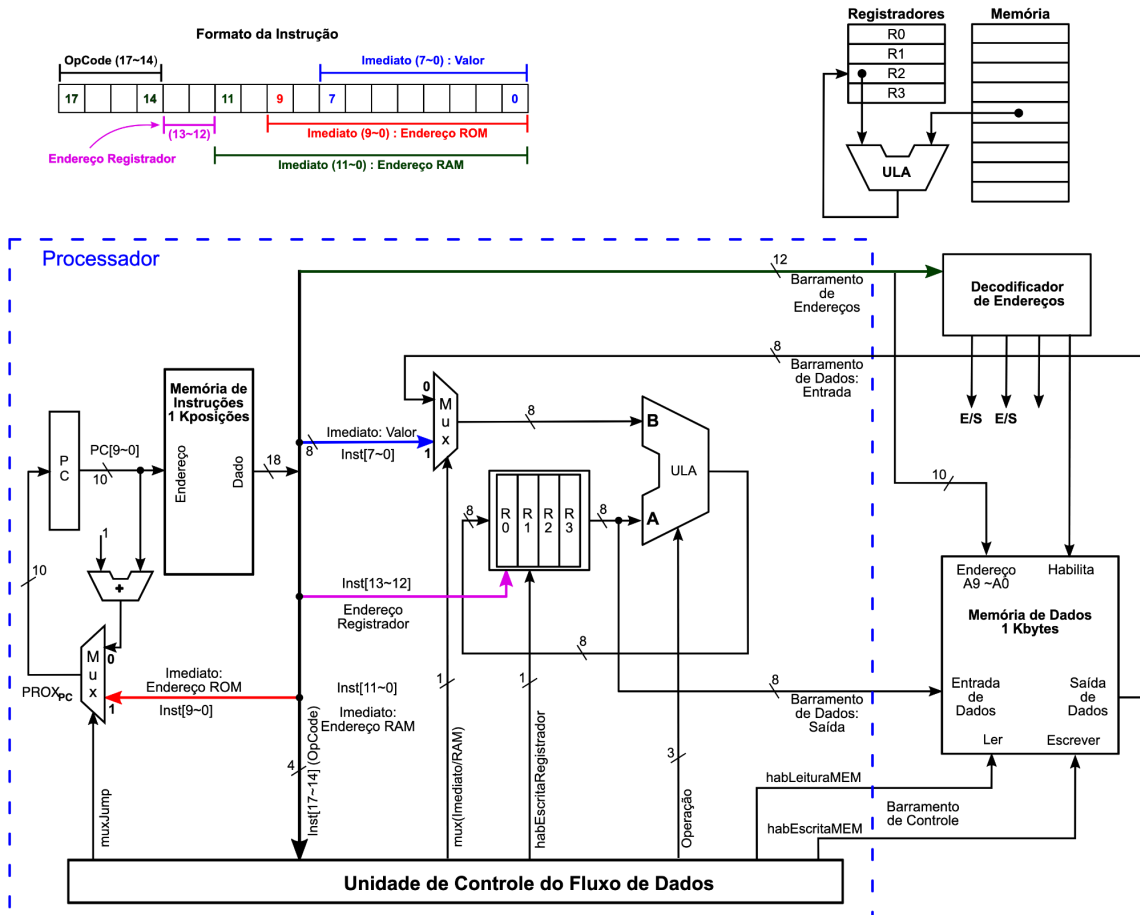
Observações importantes:

- Vale ressaltar que caso o usuário escolha um **valor acima de 9**, ou seja: A, B, C, D, E ou F; o software irá definir automaticamente o **limite dessa casa decimal como 9** nas casas de unidade de segundo e minuto e 5 nas casas de dezena do segundo e minuto (existem mais alguma especificidades por conta dos números na hora)

Arquitetura do processador

A arquitetura do nosso processador é a Registrador-Memória. Com essa arquitetura temos 4 registradores em que podemos armazenar momentaneamente valores para serem usados na ULA.

Fluxo de Dados - Arquitetura Registrador Memória



Instruções e sua sintaxe

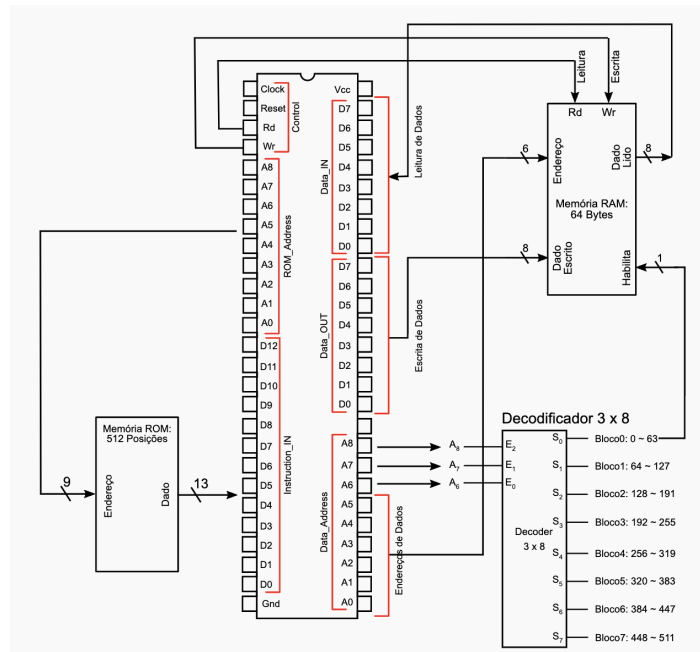
Mnemonico	Código binário	Descrição
NOP	0000	Nada
LDA	0001	Carrega um valor da memória no acumulador
SOMA	0010	Soma A e B e armazena no acumulador
SUB	0011	Subtrai B de A e armazena no acumulador
LDI	0100	Carrega o valor do imediato no acumulador
STA	0101	Salva o conteúdo do acumulador na memória
JMP	0110	Desvio incondicional
JEQ	0111	Desvio condicional
CEQ	1000	Compara se A igual a B
JSR	1001	Desvio para subrotina
RET	1010	Retorno da subrotina
ADDI	1011	Operação de AND com o imediato

Formato das instruções

LDI reg0 \$9 → tmp(0) := LDI & reg0 & '0' & x"09";

- Posição na memória ROM
- Instrução (Opcode)
- Endereço do registrador
- A8
- Imediato

Fluxo de dados para o processador

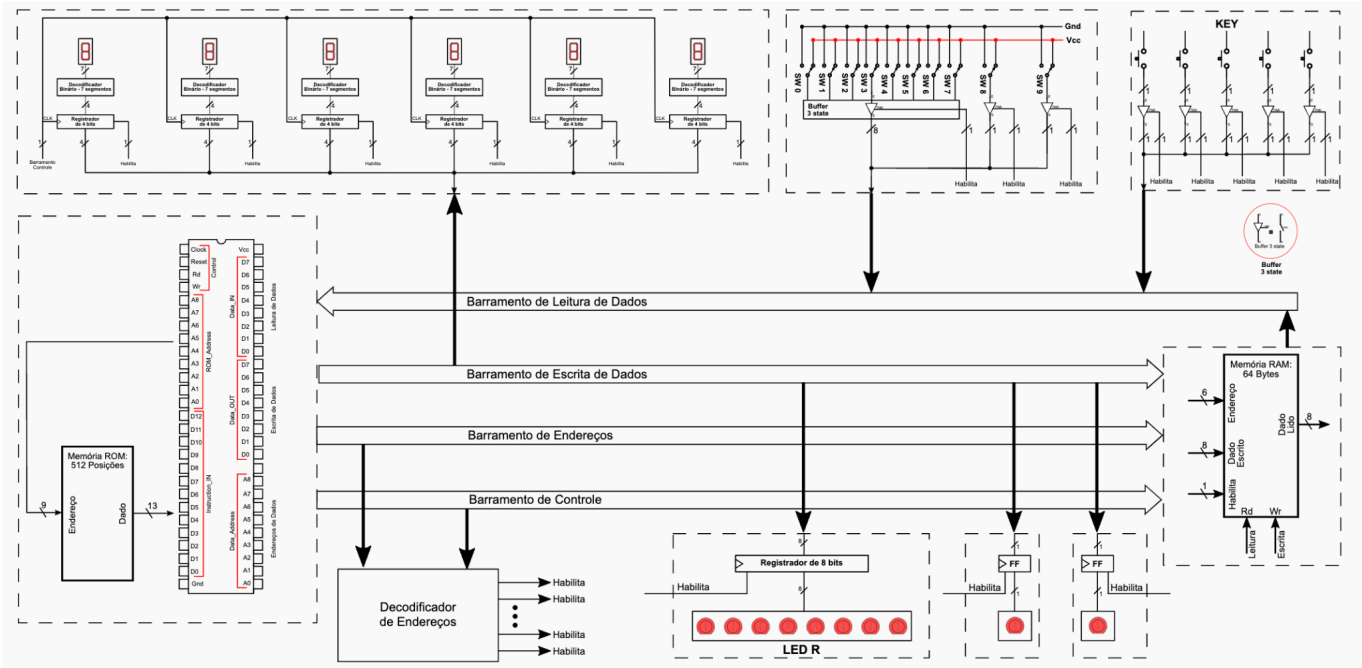


O código a ser executado pela placa está contido na Memória ROM. Assim, sequencialmente, as instruções entram no processador e dentro dele executam diferentes caminhos, ativando e desativando pontos de controles (descritos na próxima sessão). Saindo do processador temos o clock, que é quem sincroniza a execução de todos os periféricos, e temos também caminhos de dados que alimentam a RAM, decodificadores, display 7SEG, switches, buttons e LEDs.

Pontos de controle e utilização

Mnemonico	Código binário	Hab Escrita Retorno	JMP	RET	JSR	JEQ	Sel MUX	Hab_A	Operação	habFlag=	RD	WR
NOP	0000	0	0	0	0	0	X	0	XXX	0	0	0
LDA	0001	0	0	0	0	0	0	1	010	0	1	0
SOMA	0010	0	0	0	0	0	0	1	001	0	1	0
SUB	0011	0	0	0	0	0	0	1	000	0	1	0
LDI	0100	0	0	0	0	0	1	1	010	0	0	0
STA	0101	0	0	0	0	0	0	0	XXX	0	0	1
JMP	0110	0	1	0	0	0	X	0	XXX	0	0	0
JEQ	0111	0	0	0	0	1	X	0	XXX	0	0	0
CEQ	1000	0	0	0	0	0	0	0	000	1	1	0
JSR	1001	1	0	0	1	0	X	0	XXX	0	0	0
RET	1010	0	0	1	0	0	X	0	XXX	0	0	0
ADDI	1011	0	0	0	0	0	1	1	100	0	1	0

Diagrama de conexão do processador com os periféricos



Mapa de memória

Mapa de Memória				
Endereço em Decimal	Periférico	Largura dos Dados	Tipo de Acesso	Bloco (Página) de Memória
0 ~ 63	RAM	8 bits	Leitura/Escrita	0
64 ~ 127	Reservado	—	—	1
128 ~ 191	Reservado	—	—	2
192 ~ 255	Reservado	—	—	3
256	LEDR0 ~ LEDR7	8 bits	Escrita	4
257	LEDR8	1 bit	Escrita	4
258	LEDR9	1 bit	Escrita	4
259 ~ 287	Reservado	—	—	4
288	HEX0	4 bits	Escrita	4
289	HEX1	4 bits	Escrita	4
290	HEX2	4 bits	Escrita	4
291	HEX3	4 bits	Escrita	4
292	HEX4	4 bits	Escrita	4
293	HEX5	4 bits	Escrita	4
294 ~ 319	Reservado	—	—	4
320	SW0 ~ SW7	8 bits	Leitura	5
321	SW8	1 bit	Leitura	5
322	SW9	1 bit	Leitura	5
323 ~ 351	Reservado	—	—	5
352	KEY0	1 bit	Leitura	5
353	KEY1	1 bit	Leitura	5
354	KEY2	1 bit	Leitura	5
355	KEY3	1 bit	Leitura	5
356	FPGA_RESET	1 bit	Leitura	5
357 ~ 383	Reservado	—	—	5
384 ~ 447	Reservado	—	—	6
448 ~ 510	Reservado	—	—	7
510	Limpa Leitura KEY1	—	Escrita	7
511	Limpa Leitura KEY0	—	Escrita	7

MEM[0]	Unidade de segundo
MEM[1]	Dezena de segundo
MEM[2]	Unidade de minuto
MEM[3]	Dezena de minuto
MEM[4]	Unidade de hora
MEM[5]	Dezena de hora
MEM[9]	Valor 128
MEM[10]	Limite unidade de segundo
MEM[11]	Limite dezena de segundo
MEM[12]	Limite unidade de minuto
MEM[13]	Limite dezena de minuto

MEM[14]	Limite unidade de hora
MEM[15]	Limite dezena de hora
MEM[16]	Valor temp de unidade de segundo
MEM[17]	Valor temp de dezena de segundo
MEM[18]	Valor temp de unidade de minuto
MEM[19]	Valor temp de dezena de minuto
MEM[20]	Valor temp de unidade de hora
MEM[21]	Valor temp de dezena de hora
MEM[50]	Flag de barrar contagem

Programa em assembly

O montador assembly pode ser encontrado [aqui](#).

Agradecimentos

“Eu amo codar” - Santos, Paulo (2022)