

0. INTRODUÇÃO

1. O presente documento está dividido nos seguintes tópicos, nesta ordem, conforme requerido pelo enunciado:
 1. (03-09) arquitetura,
 2. (10-13) total de instruções e sintaxe,
 3. (14-22) formato das instruções,
 4. (23-25) fluxo de dados,
 5. (26-27) pontos de controle,
 6. (28-29) conexões internas do processador,
 7. (30-35) conexão com os periféricos,
 8. (36-37) mapa de memória, e
 9. (38-38) extras.
2. Para fins de organização, não apenas os tópicos estão numerados, como também os parágrafos, que servem para fazer referências a conteúdo já presente no texto e, também, pode ser útil para correção.

1. ARQUITETURA

3. Inicialmente, fizemos a implementação do contador, que é a base para o projeto do relógio, usando a arquitetura baseada em acumulador.
4. Nesse primeiro momento, não houve uma decisão consciente para o uso dessa arquitetura. Ela foi usada pois foi a arquitetura usada em sala de aula ao longo dos estudos guiados para a montagem do contador.
5. Após ter o contador funcionando na referida arquitetura, decidimos alterá-la para ARQUITETURA REGISTRADOR-MEMÓRIA.
6. Na arquitetura acumulador, as operações aritméticas sempre são salvas em um único registrador (no acumulador).
7. Na ARQUITETURA REGISTRADOR-MEMÓRIA, podemos escolher entre vários registradores o qual que desejamos que o resultado da operação seja salvo.
8. Portanto, a ARQUITETURA REGISTRADOR-MEMÓRIA otimiza o circuito, permitindo que se façam operações simultaneamente, salvando seus

resultados em registradores diferentes para uso posterior, deixando-a mais rápida.

9. Além disso, nosso processador recebe 13 bits de instruções, tem 8 bits de entrada e saída de dados, 9 bits de endereçamento e 2 bits para ativar leitura e escrita em periféricos.

2. TOTAL DE INSTRUÇÕES E SINTAXE

10. O total de instruções existentes é de onze. A sintaxe é definida na tabela a seguir.
11. Tabela de instruções. Autor: Paulo Carlos Santos. Extraída das páginas internas da disciplina e modificada por nós.

Instrução	Mnemônico	Código Binário
Sem Operação	NOP	0000
Carrega valor da memória para A	LDA	0001
Soma A e B e armazena em A	SOM	0010
Subtrai B de A e armazena em A	SUB	0011
Carrega valor imediato para A	LDI	0100
Salva valor de A para a memória	STA	0101
Desvio de execução	JMP	0110
Desvio condicional de execução	JEQ	0111
Comparação	CEQ	1000
Chamada de Sub Rotina	JSR	1001
Retorno de Sub Rotina	RET	1010

- 12.
13. A tabela acima representa as possíveis instruções a serem dadas para o processador com uma breve descrição juntamente com seu mnemônico, que é a forma que é usada na implementação em VHDL no projeto para acionamento da instrução. Os pontos de controle e a leitura e escrita estão listados na tabela que consta na seção 5 – PONTOS DE CONTROLE.

3. FORMATO DAS INSTRUÇÕES

14. As instruções têm o formato definido a seguir.

15. tmp(0) := INS & RX & '0' & x"00";

16. Onde INS é uma das instruções definidas na tabela do parágrafo 12 do item 2, RX é o registrador utilizado e 0 são os campos de bits.

17. Por exemplo, a instrução:

18. tmp(1) := STA & R2 & '1' & x"01";

19. Ela salva na memória 256 o valor contido no registrador 2.

20. Outro exemplo:

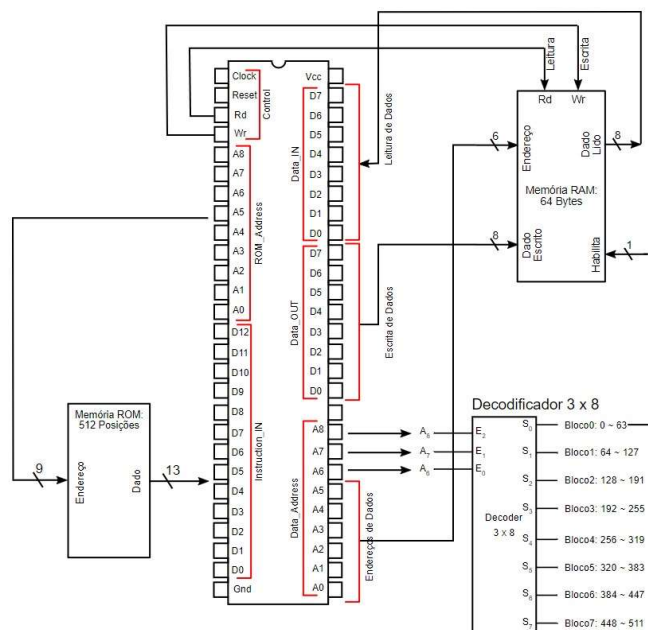
21. tmp(0) := LDI & R0 & '0' & x"05";

22. Carrega o número x05 no registrador R0.

INTENCIONALMENTE EM BRANCO

4. FLUXO DE DADOS

23. O fluxo de dados para o processador é como apresentado a seguir, usando por base o diagrama de autoria de Paulo Carlos Santos, que usamos para o estudo guiado para a implementação deste contador.



24.

25. No diagrama acima, o ROM_ADDRESS diz pra ROM qual a próxima instrução, e a ROM responde pela porta INSTRUCTION_IN do processador. O RD e o WR indicam se a memória RAM deve ser lida ou escrita. Além disso, o RD permite a leitura de botões e switches se o endereço destes estiver na saída DATA_ADDRESS. Da mesma maneira o WR permite a escrita nos LEDs e nos displays caso o endereço destes estiver na saída DATA_ADDRESS. Ao ler qualquer periférico todos os dados recebidos entram pelo DATA_IN e ao escrever data nos periféricos a escrita é feita pelo DATA_OUT.

INTENCIONALMENTE EM BRANCO

5. PONTOS DE CONTROLE

```

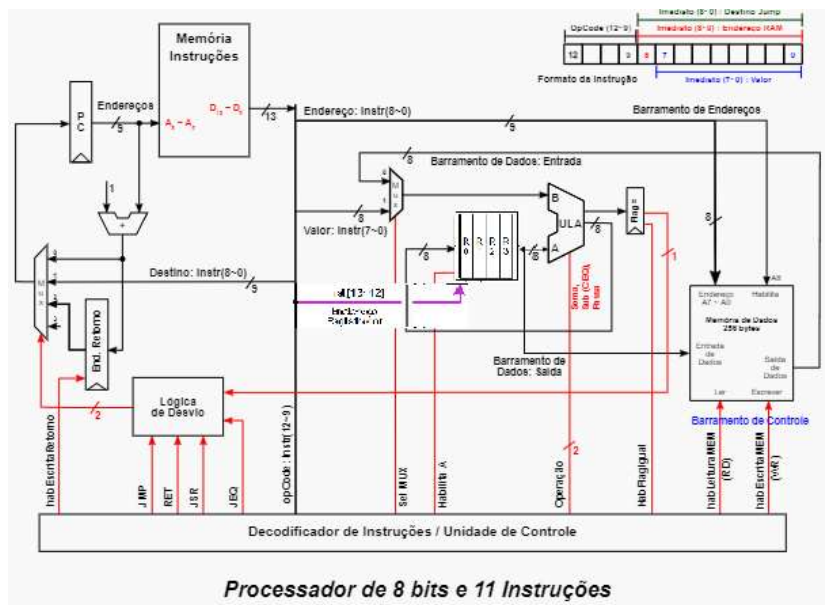
RD / WR / JSR / RET / JEQ / FREG/ JMP / Sel / ACU / ULA
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '1' & '0' & "11" -- NOP
'1' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '1' & "11" -- LDA
'1' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '1' & "01" -- SOM
'1' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & '1' & "00" -- SUB
'0' & '0' & '0' & '0' & '0' & '0' & '0' & '1' & '1' & "11" -- LDI
'0' & '1' & '0' & '0' & '0' & '0' & '0' & '0' & '0' & "11" -- STA
'0' & '0' & '0' & '0' & '0' & '0' & '1' & '1' & '0' & "11" -- JMP
'0' & '0' & '0' & '0' & '1' & '0' & '0' & '1' & '0' & "11" -- JEQ
'1' & '0' & '0' & '0' & '0' & '1' & '0' & '0' & '0' & "00" -- CEQ
'0' & '0' & '1' & '0' & '0' & '0' & '0' & '1' & '0' & "11" -- JSR
26. '0' & '0' & '0' & '1' & '0' & '0' & '0' & '1' & '0' & "11" -- RET

```

27. Além das onze instruções definidas na seção 2 – TOTAL DE INSTRUÇÕES E SINTAXE, temos também que selecionar o registrador que será usado com base na arquitetura registrador-memória. São eles: R0, R1, R2 e R3. Esses quatro registradores foram adicionados posteriormente no caminho para progredir de rubrica.

INTENCIONALMENTE EM BRANCO

6. CONEXÕES INTERNAS DO PROCESSADOR

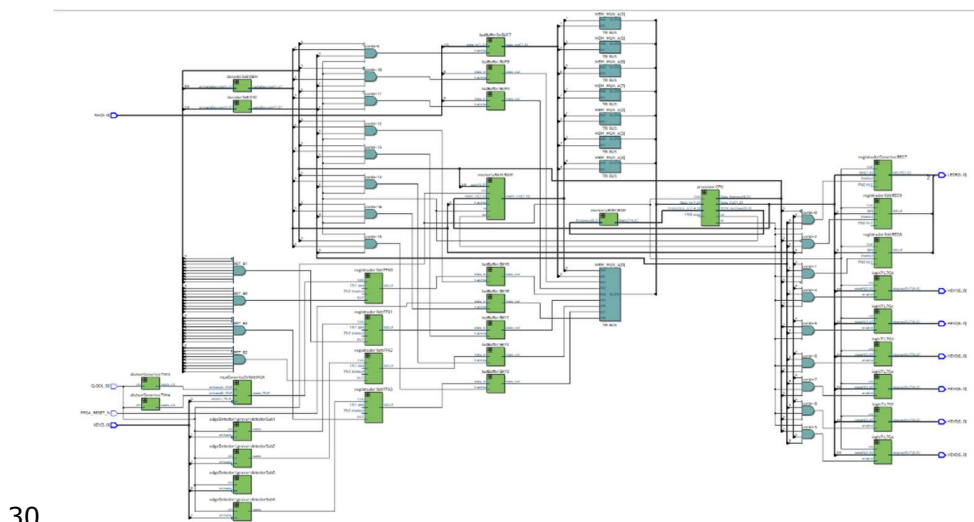


28.

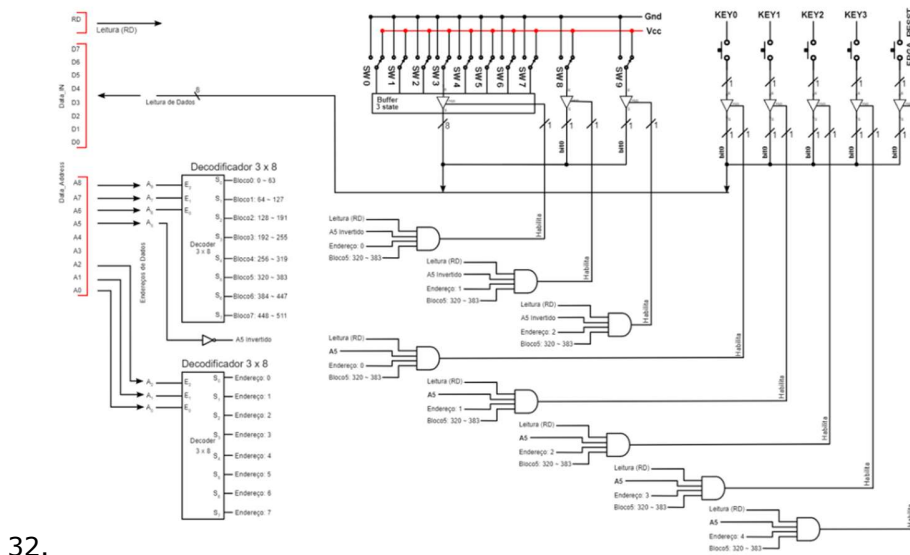
29. No diagrama acima, de autoria de Paulo Carlos Santos, temos a visão geral do nosso processador, que utiliza a arquitetura de registrador-memória, e sua conexão com a ROM e a RAM (note que ainda tem conexões com outros periféricos que não estão listadas neste diagrama).

INTENCIONALMENTE EM BRANCO

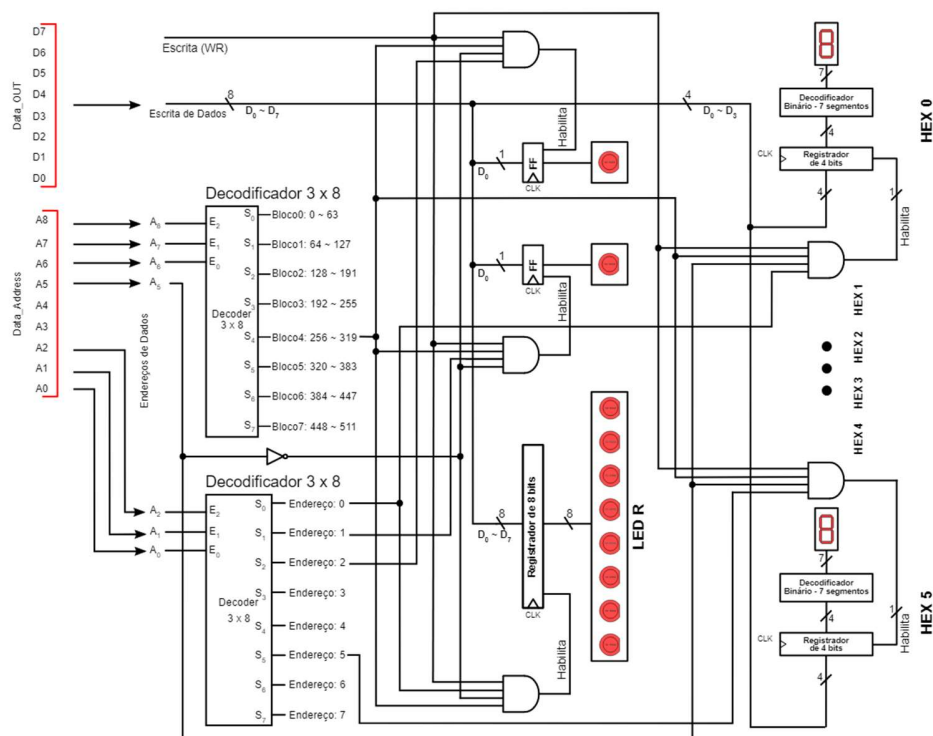
7. CONEXÃO COM PERIFÉRICOS



31. No diagrama acima, temos a visão geral do nosso circuito, que é o esquema de conexão do processador com os periféricos, extraído do RTL Viewer do Quartus Studio.



33. No diagrama acima, elaborado pelo Paulo Carlos Santos, estão representadas as conexões com os botões e switches.



34.

35. No diagrama acima, elaborado pelo Paulo Carlos Santos, estão representadas as conexões com os LEDs e Displays de 7 segmentos.

8. MAPA DE MEMÓRIA

36. O mapa de memória é apresentado a seguir. Usamos o mapa de memória descrito no estudo guiado da aula 9, elaborado por Paulo Carlos Santos.

Endereço	Periférico	Largura	Tipo	Bloco
0-63	RAM	8	RD/WR	0
64-127	Reservado	-	-	1
128-191	Reservado	-	-	2
192-255	Reservado	-	-	3
256	LED0 a 7	8	WR	4

257	LED 8	1	WR	4
258	LED 9	1	WR	4
259-287	Reservado	-	-	4
288	HEX0	4	WR	4
289	HEX1	4	WR	4
290	HEX2	4	WR	4
291	HEX3	4	WR	4
292	HEX4	4	WR	4
293	HEX5	4	WR	4
294-319	Reservado	-	-	4
320	SW0 A 7	8	RD	5
321	SW8	1	RD	5
322	SW9	1	RD	5
323-351	Reservado	-	-	5
352	KEY0	1	RD	5
353	KEY1	1	RD	5
354	KEY2	1	RD	5
355	KEY3	1	RD	5
356	FPGA RST	1	RD	5
357-507	Reservado	-	-	5,6,7
508	Z-KEY3	-	RD/WR	7
509	Z-KEY2	-	RD/WR	7
510	Z-KEY1	-	RD/WR	7
511	ZKEY0	-	RD/WR	7

37. Essa tabela apresenta o endereço em decimal para cada parte de memória e o periférico que é representado por esse endereço de memória. Além disso, também apresenta a largura dos dados para cada periférico e o tipo de acesso. Também apresenta o bloco de memória.

9. EXTRAS

38. Além do básico pedido, adicionamos dois itens em busca do B+: um despertador que pode ser configurado usando o botão B1. Após o despertador tocar, é possível desligá-lo usando o botão B3 e, além disso, usamos a instrução de chamada de sub-rotina com um nível na nossa ROM.