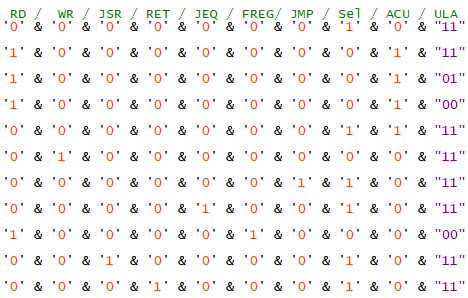
**0. Introdução**

1. Este é o relatório da entrega intermediária do projeto 1 da disciplina de Design de Computadores. Mais especificamente, é o projeto do relógio, mas esta entrega consiste apenas do contador realizado durante o estudo guiado das aulas seis a dez.
2. O presente documento está dividido nos seguintes tópicos, nesta ordem, conforme requerido pelo enunciado:
   * + 1. (04-10) arquitetura do processador,
       2. (11-16) total de instruções e sua sintaxe,
       3. (17-25) formato das instruções,
       4. (26-28) fluxo de dados para o processador,
       5. (29-29) listagem dos pontos de controle e sua utilização,
       6. (30-33) rascunho do diagrama de conexão do processador com os periféricos,
       7. (34-35) rascunho do mapa de memória, e
       8. (36-36) extras no projeto.
3. 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 do processador**

1. Inicialmente, fizemos a implementação do contador, que é a base para o projeto do relógio, usando a arquitetura baseada em acumulador.
2. 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.
3. Após ter o contador funcionando na referida arquitetura, decidimos alterá-la para **a arquitetura registrador-memória**.
4. Na arquitetura acumulador, as operações aritméticas sempre são salvas em um único registrador (no acumulador).
5. Na arquitetura registrador-memória, podemos escolher entre vários registradores o qual que desejamos que o resultado da operação seja salvo.
6. 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.
7. 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 sua sintaxe**

1. O total de instruções existentes é de onze. A sintaxe é definida na tabela a seguir.
2. Tabela de instruções. Autor: Paulo Carlos Santos. Extraída das páginas internas da disciplina e modificada por nós.
3. 
4. 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 abaixo com a mesma ordem das instruções acima.
5. 
6. Além das onze instruções definidas na tabela acima (parágrafo 12), 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 da rubrica C para a rubrica B.

**3. Formato das instruções**

1. As instruções têm o formato definido a seguir.
2. tmp(0) := INS & RX & '0' & x"00";
3. 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.
4. Por exemplo, a instrução:
5. tmp(1) := STA & R2 & '1' & x"01";
6. Ela salva na memória 256 o valor contido no registrador 2.
7. Outro exemplo:
8. tmp(0) := LDI & R0 & '0' & x"05";
9. Carrega o número x05 no registrador R0.

**4. Fluxo de dados para o processador**

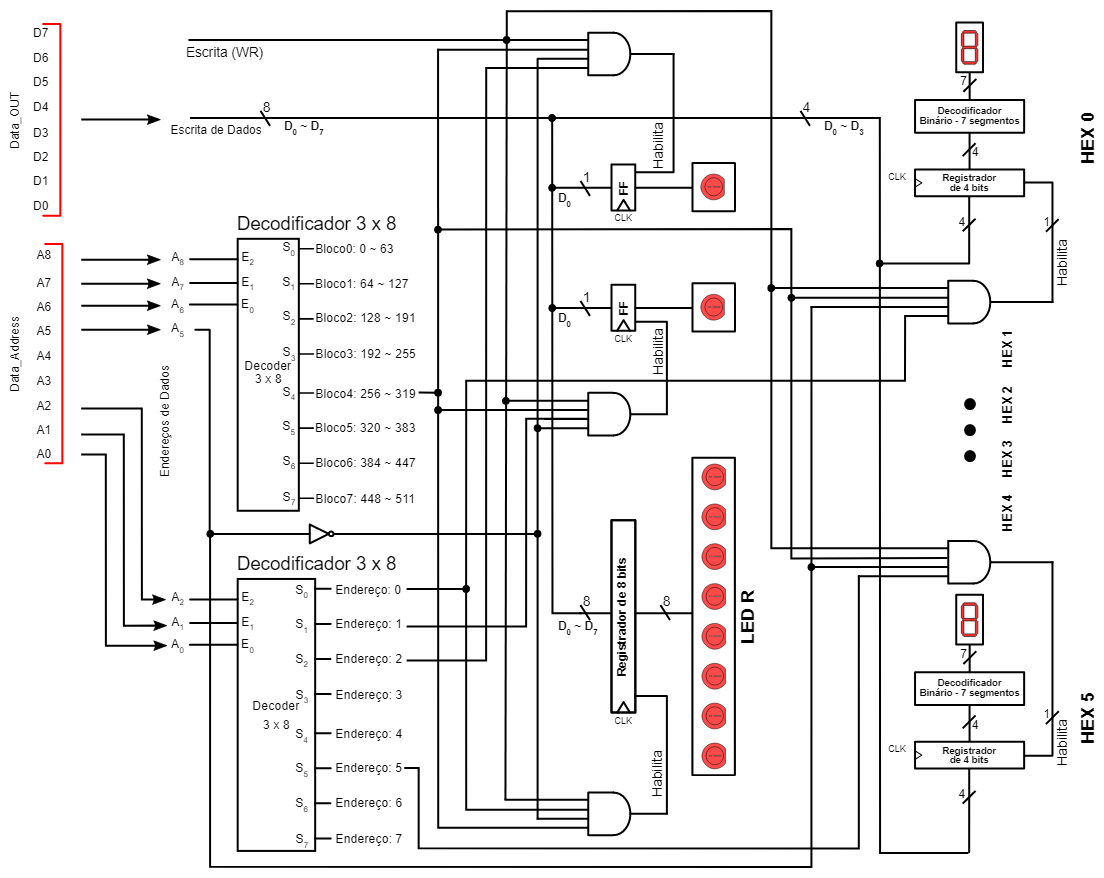
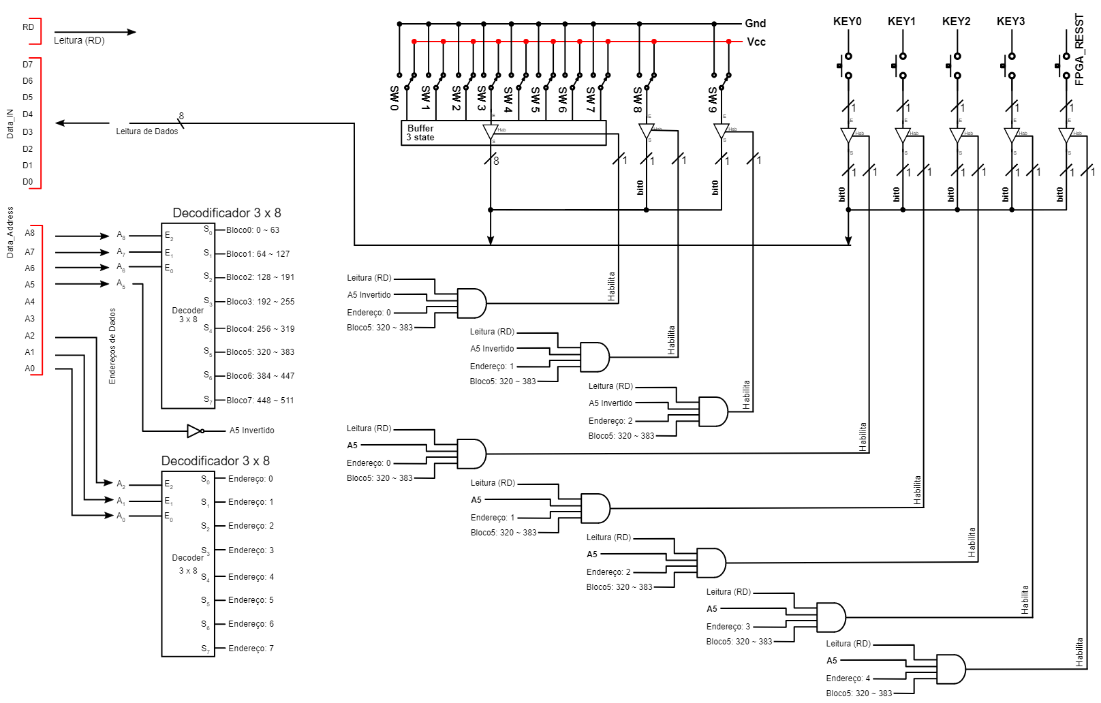
1. O fluxo de dados para o processador é como apresentado a seguir, usando por base o diagrama de autoria de Paulo Carlos Santos, que usamos para o estudo guiado para a implementação deste contador.
2. Diagrama, Esquemático

   Descrição gerada automaticamente
3. Basicamente 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 switchs se o endereco destes estiver na saida Data\_Address. Da mesma maneira o wr permite a escrita nos LEDs e nos displays caso o endereco destes estiver na saida Data\_Address. Ao ler qualquer periferico todos os dados recebidos entram pelo Data\_IN e ao escrever data nos periféricos a escrita é feita pelo Data\_OUT.

**5. Listagem dos pontos de controle e sua utilização**

1. Os pontos de controle não serão descritos nesta seção, pois já foram apresentados e descritos na tabela que consta no parágrafo 15 da seção 2. Lá são descritos os pontos de controle e sua utilização.

**6. Rascunho do diagrama de conexão do processador com os periféricos**

1. 
2. No diagrama acima, elaborado pelo Paulo Carlos Santos estão representadas as conexões com os LEDs e os Displays.
3. 
4. No diagrama acima, elaborado pelo Paulo Carlos Santos estão representadas as conexões com os Botões e Switches.

**7. Rascunho do mapa de memória**

1. 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 |

1. 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.

**8.**  **Extras**

1. Além do básico pedido adicionamos dois itens em busca do A, o primeiro é simples, mas faz toda a diferença que é ter LEDs indicadores quando se está configurando um limite no contador, com isso é possível entender o que está ocorrendo no contador. O segundo é a possibilidade de usar o botão 3 para decrementar a contagem do contador e por último implementamos a possibilidade de escolher um número pré-determinado no contador usando o botão 2 com os Switches.