Problema 3: ”Memorizando!”

Turma P04

[1]MI - Projetos de Circuitos Digitais, Período 2015.1
Tutor: Marcos Paz
Curso de Engenharia de Computação
Universidade Estadual de Feira de Santana (UEFS)

Resumo. *O presente relatório descreve o processo de resolução do terceiro problema do MI - Circuitos Digitais, da supracitada turma,no período de 2015.1, na Universidade Estadual de Feira de Santana. O problema propôs, dado o sucesso dos projetos pretéritos, que os estudantes desenvolvessem um protótipo do jogo Batalha Naval em FPGA, tomando como base o protótipo do controlador para matriz de LED, sendo que a comunicação com este novo circuito deveria acontecer através de um computador pessoal utilizando o conceito de comunicação serial, tornando o projeto a ser desenvolvido, mais confiável em relação a transmissão de dados.*

1. Introdução

Hoje em dia cresce, substancialmente, a ampla utilização de jogos no segmento lúdico de simulação 2D. Estes são utilizados em diversas áreas conhecidas, tendo uma mais elevada utilização na educação. Além do papel que os mesmos podem fornecer à educação, podem servir , também, única e exclusivamente como forma de entretenimento entre os mais jovens, bem como para o público mais adulto. Um exemplo destes jogos, é famigerado Batalha Naval. Sendo no Brasil no ano de 1988, o jogo batalha naval é um dos jogos de maiores sucessos entre o público mundial. Em sua forma mais rústica, dois adversários desenhavam em folhas de papel, navios posicionados em um mar imaginário quadriculado, formando uma grade. Todavia, no contexto atual, além de encontrar versões deste jogo de maneira similar aos seus primórdios, encontra-se, também, aplicativos para diversas arquiteturas que executam o mesmo processo, facilitando assim a sua disseminação entre as pessoas.

Além da existência, como visto, deste tipo de jogo na forma de aplicativos móveis, existe a possibilidade, também, de desenvolvê-lo utilizando circuitos digitais, visando a criação de um jogo lúdico de simulação 2D. É de se notar, que para desenvolvê-lo, faz-se necessário utilizar um conceito muito importante na eletrônica digital, a saber, memória RAM. A criação deste tipo de jogo, traz diversas vantagens para o empreendedor que o desenvolve, justamente pelo fato do mercado de jogos está sempre em constante expansão. lembrando que hoje em dia, aplicações envolvendo a tecnologia digital tem ampla aceitação no mercado, o ser humano necessita de recursos tecnológicos, com forma também de pertença, e inclusão no mundo global tecnológico.

Desta forma, observando a mudança atual do mercado, principalmente no que diz respeito a alta receptividade em relação a jogos eletrônicos, o grupo Inova Digital Bahia S.A resolveu adaptar o projeto desenvolvido anteriormente, para que o mesmo funcionasse como uma espécie de jogo Batalha Naval, seguindo um modelo de simulação $2D$. Sendo que este novo protótipo deveria ser controlado remotamente, a partir de um

computador pessoal, enviando os dados por meio de canal de comunicação serial. Neste jogo, dois modos estarão disponíveis, o modo de gravação e o modo de jogo propriamente dito, cada modo contendo suas especificações e características próprias, as quais serão descritas nas seções posteriores.

É importante lembrar, que para desenvolver este projeto, foram utilizados além dos recursos já conhecidos, fez-se uso também de Máquinas de estados, registrador de descolocamento, porta serial, protocolo de comunicação RS-232, conversores de níveis elétricos e, por último, mas não menos importante, Software para comunicação com Hardware.

2. Fundamentação Teórica

Na seção que segue, serão descritos os conceitos utilizados para resolução do problema proposto, frisando os conceitos que foram de fundamental importância para solucionar o mesmo, tendo contribuição de maneira direta ou indireta.

2.1. Máquinas de Estados

Em se tratando de Circuitos Digitais, diversos dispositivos têm uma ampla importância, dentre estes dispositivos temos os conhecidos contadores, os quais segundo[Tocci Ronald 1997] são Máquinas de estados específicas. Floyd, por sua vez, vai mais adiante, e fala que Máquinas de estados ou circuito sequencial é um circuito que consiste de uma seção de lógica combinacional e uma seção de memória (flip-flops)[Floyd 2011]. Na figura 12podemos observar o circuito típico para uma máquina de estados. Vale acrescentar que a quantidade de estados que uma máquina pode interpretar é diretamente ligada a quantidade de Flip-Flop, a quantidade de estados pode ser obtido fazendo 2^n, sendo n o número de Flip-flops.

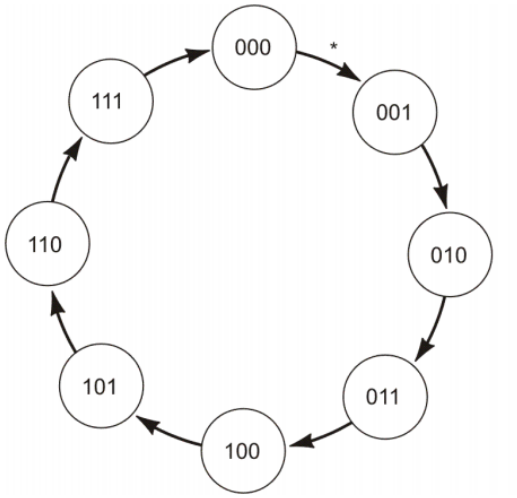


Figura 1. Diagrama de estados de um contador de 3 bits[Floyd 2011]

Existem basicamente, dois modelos de máquinas de estados, a saber, máquina Moore e máquina Mealy. Na máquina Moore, a saída depende única e exclusivamente dos seus estados atuais, enquanto na máquina Mealy além de depender dos seus estados atuais, a saída depende, também, das variáveis de entrada naquele instante. Na figura 2

podemos observar o digrama de transição de estados para uma máquina Moore, já na figura 3 temos o diagrama para uma máquina Mealy. É importante notarmos, através dos diagramas, a diferença referente a saída de cada máquina.

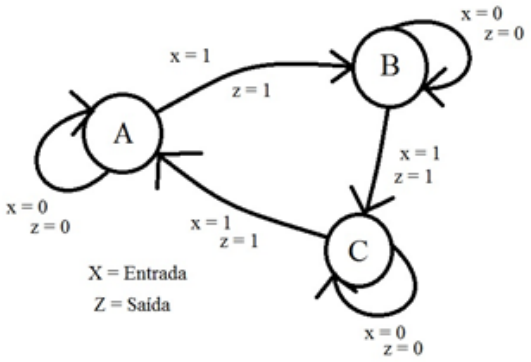


Figura 2. Exemplo de diagrama de estados usando uma máquina Mealy

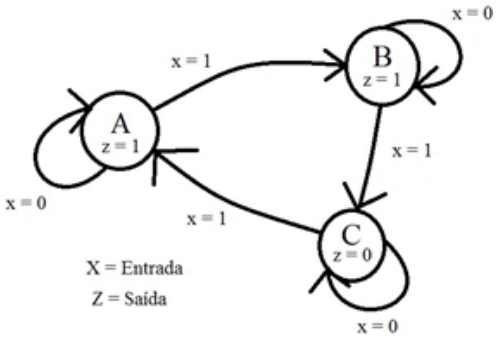


Figura 3. Exemplo de diagrama de estados usando uma máquina Moore

2.2. Flip-Flop T

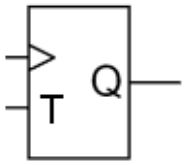
Além dos Flip-Flops até o momento estudados, temos na literatura, outro Flip-flop que é amplamente utilizado, qual seja, o Flip-flop T. Na figura 4 podemos analisar o bloco lógico e a tabela verdade para este Flip-Flop. Quando a entrada T estiver em estado alto, o flip-flop T, onde T significa *toggle*, comutará quando o *clock* for aplicado. Se a entrada T for baixa, o flip-flop mantém o valor do seu estado. Vale lembrar, que o Flip-flop T pode ser obtido utilizando um Flip-flop J-K, basta conectar as entradas J e K em nível lógico alto de maneira intermitente.

2.3. Flip-Flop T

2.4. Registrador de Deslocamento

No universo dos circuitos digitais, principalmente no que diz respeito aos Flip-Flops, encontramos diversas aplicações. Uma, das várias aplicações é na forma de registrador de deslocamento, os quais são amplamente utilizados hoje em dia no desenvolvimento de circuitos robustos. Segundo [Floyd 2011] os registradores de deslocamento consistem de arranjos de Flip-Flops e são importantes em aplicações que envolvem o armazenamento

T	Q	Q*
0	0	0
1	1	0
0	1	1
1	0	1



Q* → Estado seguinte do Q

Figura 4. Tabela Verdade do Flip Flop T

e a transferência de dados em sistemas digitais. É importante esclarecer, que o funcionamente do registrador de deslocamento é diferente de um contador comum, ou seja, ele não tem uma sequência específica de estados, apesar de em alguns casos poder ser utilizado como tal. Na figura 5 podemos ver um circuito típico para um registrador de deslocamento de quatro bits com entrada e saída serial de dados.

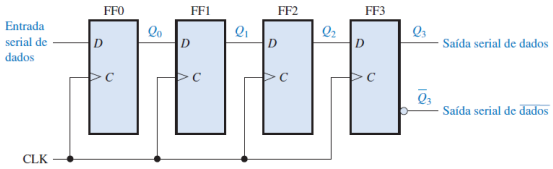


Figura 5. Representação de um registrador de deslocamento de 4 bits [Floyd 2011]

2.5. Flip-Flop T

São diversos, os tipos de registradores de deslocamento, na figura 6 pode-se observar os diversos tipos de registradores de deslocamento existentes, sendo ilustrados de forma abstrata. Dentre eles, um dos mais importantes está o registrador com entrada serial e saída paralela, o qual será abordado com mais detalhe na próxima sub-subseção.

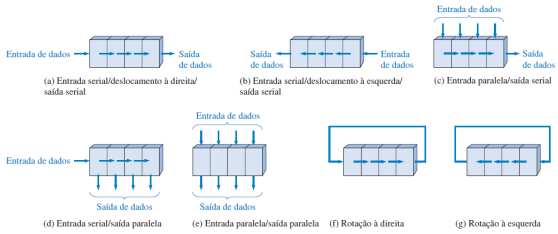


Figura 6. Diferentes tipos de registradores, segundo [Floyd 2011]

2.5.1. Registrador de deslocamento com entrada serial e saída paralela

Como o próprio nome já designa, neste tipo de registrador a entrada de dados se dá de forma serial, contudo, a saída não será serial e sim paralela. Desta forma, segundo

[Floyd 2011] uma vez armazenados os dados, cada bit aparece em sua linha de saída respectiva e todos os bits são disponibilizados simultaneamente, em vez de um bit de cada vez como no registrador com saída serial. Na figura 7 podemos ver uma exemplificação para este circuito. Notem que neste tipo de registrador a saída de cada estágio está disponível.

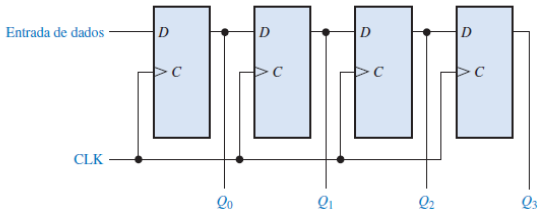


Figura 7. Registrador Serial-Paralelo

2.6. Comunicação Serial

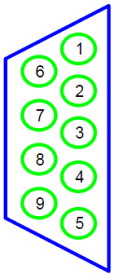
No contexto computacional, temos duas formas de comunicação principais, são elas: comunicação serial e comunicação paralela. cada uma com suas vantagens e desvantagens. Na comunicação paralela os bits são transferidos de maneira simultânea, conferindo para este tipo de comunicação a vantagem de rapidez e simplicidade de interface, contudo essa mesma vantagem acaba gerando algumas desvantagens, pois há um maior número de conexões, o que pode gerar ruído, perda de sincronismo além de um custo maior. Já na comunicação serial os dados são transferidos bit a bit, conferindo a vantagem de se ter menos fios, ou seja, baixo custo, além de ser capaz de transmitir dados em distâncias maiores. Contudo, nesta forma de comunicação, justamente por haver apenas um fio que transmite os dados, ela acaba sendo mais lenta, e com um grau de complexidade maior. Todavia, apesar das desvantagens da comunicação serial, a mesma se mostra mais robusta e mais viável, a ser aplicada em sistemas onde a confiabilidade dos dados são importantes. Assim, esta subseção tratará especificamente deste tipo de comunicação.

2.6.1. Transmissão de Dados

Antes de falarmos acerca de como ocorrer a transmissão de dados de forma serial, é importante lembrar que existe diversas interfaces que empregam este tipo de comunicação, entre as interfaces mais antigas estão as portas $DB9$ e $DB25$, as quais podem ser encontradas nos computadores mais antigos. Na figura 8 temos um exemplo para estas duas portas. Cada pino nestas portas tem uma determinada função, podendo variar de acordo com o seu tipo. Na figura 9 podemos ver a função de cada pino na porta $DB9$.

Na maioria das aplicações, a comunicação serial acontece de forma assíncrona, onde geralmente se tem 11 bits sendo transferidos. Um desses bits é o *start bit*, o qual notifica um receptor que um novo dado serial esta disponível, seguido pelos bits de dados, um bit opcional de paridade *parity* e um ou mais bits de parada *stop bits*. percebam que o número de bits de paridade, de parada podem variar, dependendo apenas do padrão de transmissão escolhido pelo projetista. Na figura 10 pode-se analisar esta forma de comunicação de maneira mais concreta. Vale acrescentar, que enquanto o bit de parada

Figura 8. Portas DB9 e DB25, interfaces seriais encontradas em computadores antigos.



Sigla	Descrição	Pino
DCD	Carrier Detect	1
RX	Receive Data	2
TX	Transmit Data	3
DTR	Data Terminal Ready	4
GND	Ground	5
DSR	Data Set Ready	6
RTS	Request to Send	7
CTS	Clear to Send	8
RI	Ring Indicator	9

Figura 9. Funcionalidades dos pinos presentes na porta DB9

se manter em nível alto, os dados não estarão sendo transmitidos, e sim, apenas, quando ocorrer a mudança de estado do nível lógico alto para baixo, caracterizando assim, o bit de início, ou *start bit*.

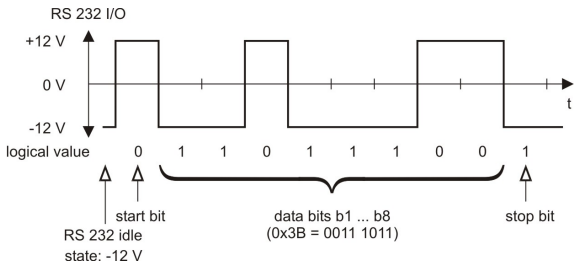


Figura 10. Características de uma transferência de dados serial.

Cada dispositivo que emprega a comunicação serial, tem uma determinada taxa de velocidade, sendo que para se referir a velocidade de transferência devemos usar o termo bps *bits-per-second*, ou seja, bits por segundo. Em alguns equipamentos, costuma-se chamar esta taxa como *baud*. Desta forma, a expressão 9600 bps é equivalente a 9600 bauds.

2.6.2. Padrão RS-232

Segundo (http://www.c2o.pro.br/automacao/x834.html) RS-232 é um padrão definido pela *EIA-Eletronic Industries Association* para os dispositivos usados para comunicação serial. Este padrão está disponível em três versões, quais sejam, (A, B e C). Cada um especifica uma faixa de voltagens para os níveis ligado e desligado. Por exemplo, na

versão RS-232 para representar nível lógico alto utiliza-se as tensões de $-3V$ a $-12V$, enquanto para representar nível lógico baixo é utilizado tensão na faixa de $+3V$ e $+12V$. É importante notar que a representação de nível lógico alto e nível baixo nesse padrão não utiliza o modelo convencional de tensão. Vale lembrar que este padrão é utilizado marjoritariamente nas portas $DB9$ e $DB25$.

2.6.3. Conversor de nível MAX-232

Os dispositivos eletrônicos utilizam níveis de tensão diferentes para identificar nível lógico baixo e nível lógico alto, dependendo apenas da tecnologia que está sendo empregada. Desta forma, quando se vai utilizar a porta serial do computador pessoal para se comunicar com algum dispositivo controlador, tem-se que analisar o padrão que o mesmo utiliza,pois a depender das especificações, será necessário fazer a conversão do nível. Por exemplo, vamos supor que queremos fazer uma comunicação entre o computador pessoal através da porta serial, com uma *FPGA*. No entanto, a *FPGA* utiliza a tecnologia *TTL*, ou seja, 0V a 0,8V representa nível lógico baixo e $2V$ a $5V$ representa nível lógico alto. Percebe-se, assim, uma incompatibilidade entre o padrão RS-232 e o *TTL*, desta forma é necessário fazer a conversão. Todavia, existe alguns dispositivos que tem a capacidade de fazer essa conversão, um deles é o MAX-232, o qual faz a conversão entre os níveis utilizados pelo padrão RS-232 para TTL, e vice-versa. Na figura 11 podemos ver uma configuração comum utilizando um conversor MAX-232.

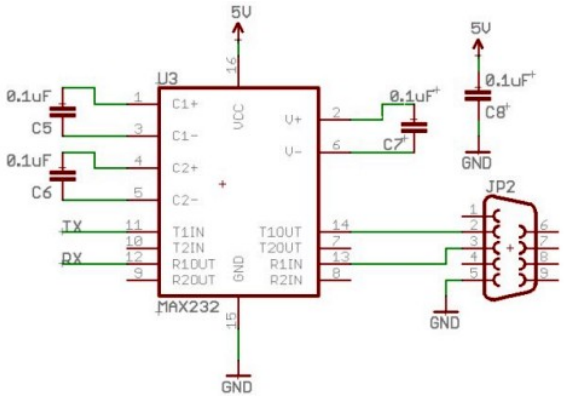


Figura 11. MAX-232: Conversor de RS-232 para TTL.

3. Metodologia

Como definido pela metodologia do PBL, a resolução do problema 3 do MI de Circuitos Digitais, iniciou-se com o levantamento de ideias pelos membros do grupo de quais seriam as possíveis resoluções para o problema. Como de praxe, surgiram diversas ideias sobre como realizar o mesmo procedimento e dúvidas relacionadas ao texto do problema.

Os membros da sessão, solucionando as questões um dos outros quando possível durante o processo, buscaram tomar decisões de projeto para formalização das obrigações, analisando fatores positivos e negativos das soluções propostas, optando pela solução que fosse mais simples e que estivesse de acordo com o problema, de acordo com a visão dos alunos. Com esse método deu-se início a construção da solução do problema 3 do MI -

Circuitos Digitais, começando pela formalização e definição de padrões de projeto e de um diagrama que as mostrasse aplicadas ao circuito.

3.1. Decisões de Projeto

Antes de iniciar a implementação do projeto em si, é necessário traçar o caminho a percorrer durante todo o processo de solução do problema. Essa metodologia foi adotada pelos membros da sessão após algumas decisões de projeto que não deram o resultado esperado nos problemas anteriores, causando incongruências entre os módulos projetados entre diferentes membros da turma.

De início, foi de comum acordo que, mais uma vez, modularizar o problema e dividir tais módulos entre pequenos subgrupos, sendo que cada subgrupo deveria enviar um pequeno relatório contendo todas as informações necessárias para a elaboração do mesmo, para conscientizar os outros alunos do processo de construção, além da descrição do bloco na ferramenta EDA *Quartus II 9.0*. A partir desse ponto, a abordagem mudou em relação aos problemas anteriores, uma vez que houveram problemas com essa segmentação. Para resolver esses problemas de incompatibilidade, a turma resolvou elaborar um diagrama completo do problema, estabelecendo padrões para cada segmento.

3.2. Elaboração do Diagrama

Iniciado na primeira sessão do problema 3 e finalizado na segunda sessão do mesmo problema, o diagrama teve como objetivo fazer com que todos os alunos tivessem consciência plena do funcionamento do circuito, da entrada a saída. Descrevendo as relações entre os blocos e a função, a quantidade de entrada e a quantidade de saídas de cada um deles e, também, padrões de nomenclatura, o diagrama descreve o que cada subgrupo deve fazer para que, quando combinadas, as frações do problema resultem em um circuito funcional e dentro da perspectiva do problema.

O resultado desse processo de descrição am alto nível do projeto pode ser conferido na figura 12, que destaca também as tarefas de cada subgrupo. Por exemplo, o subgrupo amarelo é responsável pela elaboração dos módulos correspondentes ao *decodificador da matriz* e *Bloco do Mux-Coordenada*.

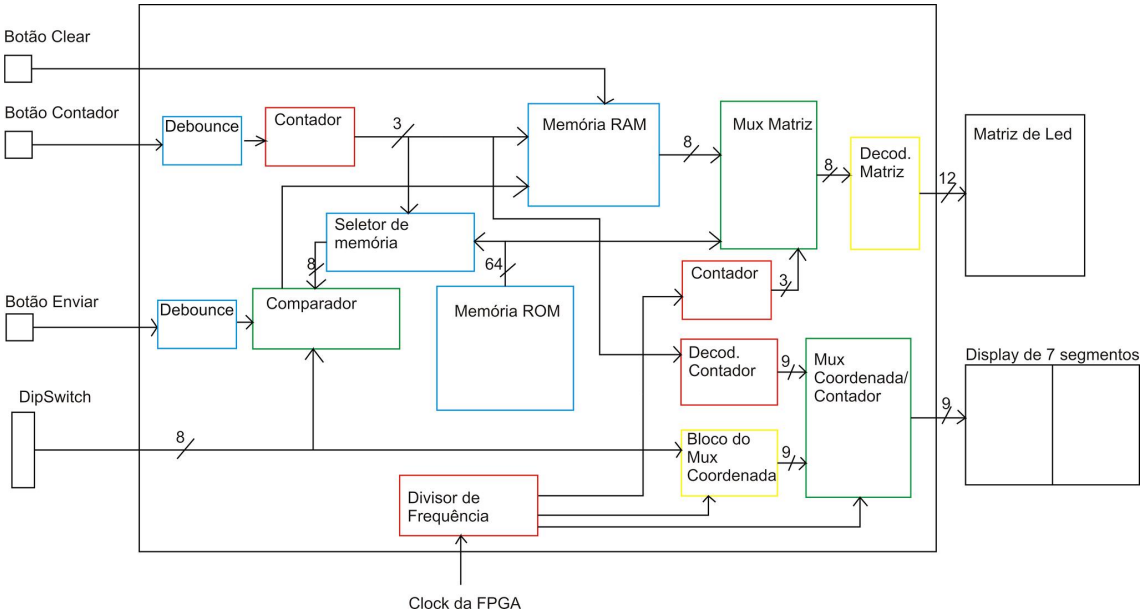


Figura 12. Diagrama de blocos do Problema 3: Memorizando

3.3. Circuito Debounce

Circuito utilizado para amenizar a trepidação da onda do botão, recebe como entrada uma frequência de clock dividida resultando em $33Hz$ e o pulso do botão, e tem como saída um pulso com baixa oscilação caso o botão seja pressionado. A implementação desse circuito pode ser vista na figura 13.

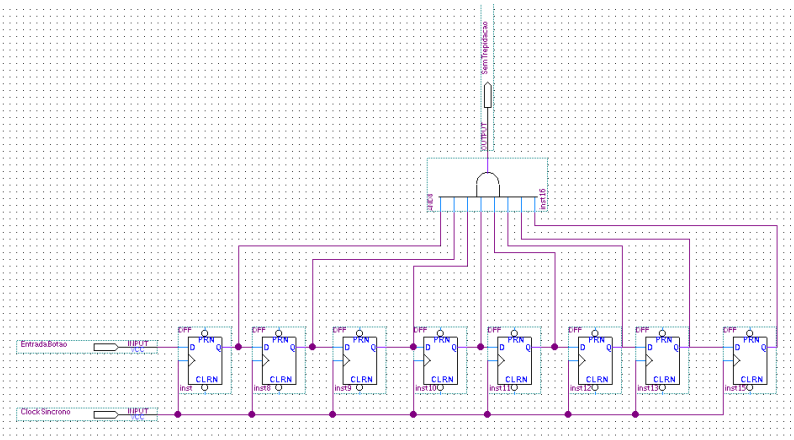


Figura 13. Implementação do circuito Debounce na EDA

3.4. Memória ROM

Necessária para armazenar os dados das coordenadas pré definidas no problema, guarda os dados das coordenadas(linha e coluna) em forma de um arranjo de números binários. Respeitando os endereços fornecidos junto aos pares de coordenadas e convertendo os mesmos para valores binários(demonstrados na tabela 1), foram elaborados os conjuntos de dados a serem programados na memória ROM.

Endereço	Coordenada(CxL)	Valor Binário(C)	Valor Binário(L)
0	B,5	1011	0101
1	B,1	1011	0001
2	C,3	1100	0011
3	D,5	1101	0101
4	D,1	1101	0011
5	E,4	1110	0100
6	E,3	1110	0011
7	E,2	1110	0010

Tabela 1. Informações que devem ser armazenadas e suas versões e binário

Para obter os conjuntos de dados desejados, foram dispostos 8 conjuntos de 8 saídas cada, de modo que as saídas de cada conjunto correspondessem aos dados de cada um dos pares de coordenadas. Como pode ser visto na figura 14, as saídas foram nomeadas no padrão $m\alpha b\beta$, onde α corresponde ao endereço de memória e β o bit de saída(para $\beta > 4$, os bits representam a saída da linha e para $3 < \beta < 8$ a saída da coluna).

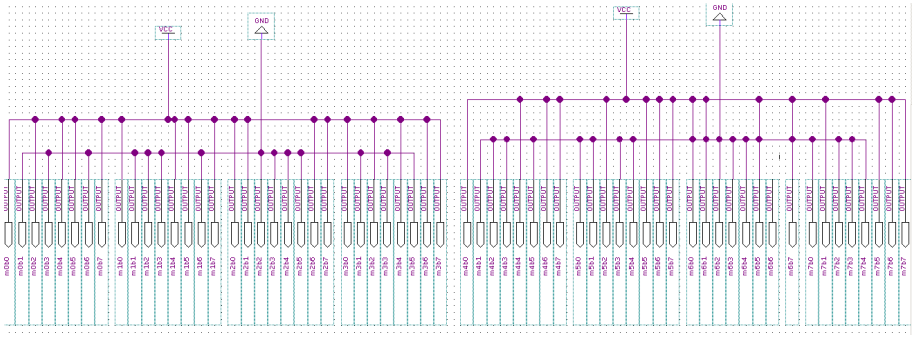


Figura 14. Descrição da memória ROM na EDA

3.4.1. Seletor de Memória

Para efetuar o acesso a memória ROM, é necessário endereçá-la. Buscando isolar os componentes, a turma optou por elaborar um circuito multiplexador que recebe como entrada padrão todas as saídas da memória ROM e, através de 3 seletores b_0, b_1, b_2 que devem ser um contador binário de 3 bits, sendo b_0 o bit menos significativo, põe em uma saída S de 8 bits somente os bits da coordenada de endereço apontado no contador. Para ajudar na tarefa da seleção, sabendo que o contador é um contador binário de 3 bits, foi implementado um decodificador do contador para 8 saídas, cada um representando um endereço e sendo apenas uma ativa por vez. O bloco do multiplexador implementado pode ser visto na 15.

Figura 15. Descrição do seletor da memória ROM na EDA

3.5. Contadores

Usados para controlar o endereçamento da memória e para oscilar as saídas na Matriz de Leds, os contadores c_1 e c_1, respectivamente, são contadores binários síncronos de 3 bits. Por ser responsável pelo endereçamento da memória c_1 tem como entrada o pulso do botão normalizado pelo Debounce, já c_2 um pulso de clock dividido de, aproximadamente, $132.8Hz$. A razão para esse número será explicada em 3.9.

3.6. Comparador

Sendo o tipo de comparador mais simples possível, o comparador utilizado no problema é capaz de comparar dois conjuntos de 8 bits e determinar se eles são iguais ao comparar os bits de mesma magnitude. O comparador presente no problema recebe como entrada o conjunto de dados bruto, oriundo do barramento paralelo de oito bits, e o conjunto de dados armazenados na unidade da memória ROM correspondente ao endereçamento atual determinado pelo contador c_1. Ou seja, caso o contador c_1 esteja em 1, as dados presentes no bloco 1 da memória ROM serão enviados pelo seletor de memória para o comparador.

Como uma característica extra, foi implementado no comparador usado um bit a mais. Tal bit funciona como um $enable$ e é alterado por um botão ligado a um circuito de Debounce.

Sendo $D\_L\alpha$, $D\_L\beta$ e $M\_L\alpha$, $M\_C\beta$ as entradas correspondentes para a entrada do valor inerente linha a coluna disposta no barramento e linha e coluna de referência, respectivamente, o comparados descrito na figura 16 mostra o circuito responsável por comparar coordenadas feito e utilizado pela equipe.

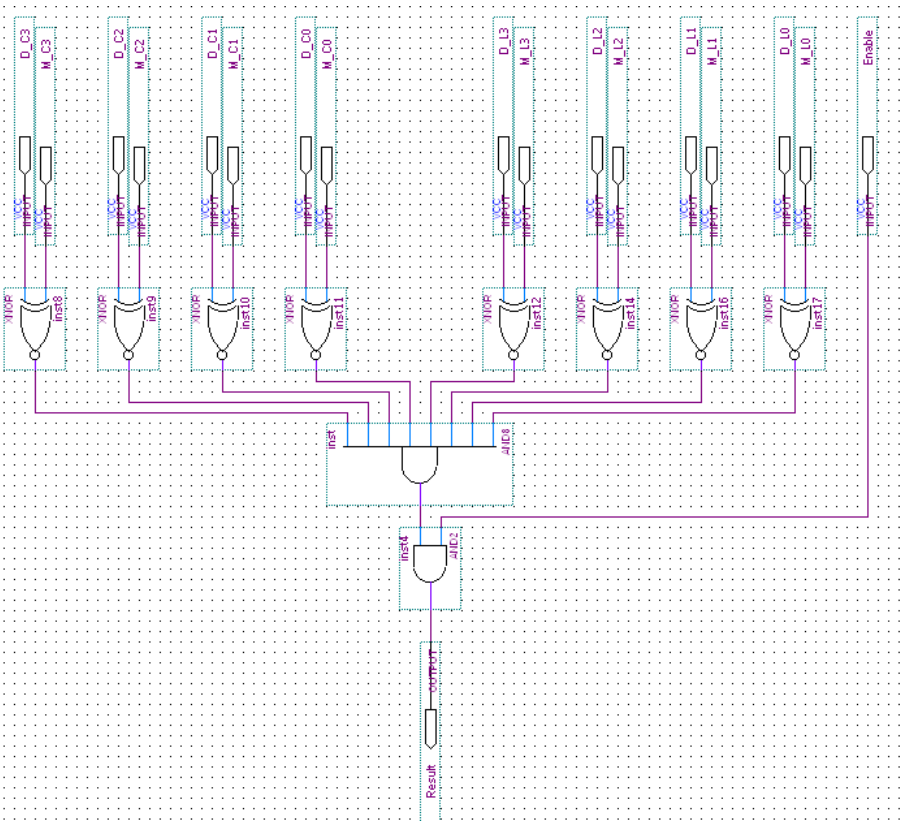


Figura 16. Descrição do comparador de coordenadas na EDA

3.7. Memória RAM

A memória RAM, de acordo com o diagrama elaborado, deve receber como bit de ativação o bit de saída do comparador, chamado de $result$ e a saída de c_1. Caso $result$ esteja em nível alto, esse nível é armazenado no endereço de memória descrito pelo contador c_1. Caso contrário, nada é feito.

Como uma entrada extra, foi colocada um bit de $clear$ para a memória, o mesmo tem a função de limpar completamente a memória em casos que o $reset$ da mesma seja desejado. A figura 17 demonstra o resultado final obtido na ferrmenta EDA, sendo possível perceber o funcionamento, entradas e saídas do respectivo módulo.

3.8. Multiplexador da Matriz de LEDs

Responsável por acender todos os leds, esse item recebe como entrada todos os 8 bits de saída da memória RAM, todas as saídas da memória ROM e as saídas do contador c_2. A função desse bloco é verificar se, segundo a memória RAM, o endereço $-1 < x < 8$ foi dado como válido, e caso positivo, acendê-lo, dado que 0 endereço x da memória RAM, informa se o endereço x da ROM foi ativo.

Dado a incapacidade da matriz de LED's de acender 8 LED's simultaneamente, o contador c_2 alterna entre as 8 coordenadas, liberando o acendimento ou não, de um LED por vez, numa frequência de aproximadamente $132.8Hz$. Essa frequência foi calculada com base na frequência máxima percebida pelo o olho humano e multiplicando-a por 8, fazendo com que a alternância entre os acendimentos não seja notada a olho-nu.

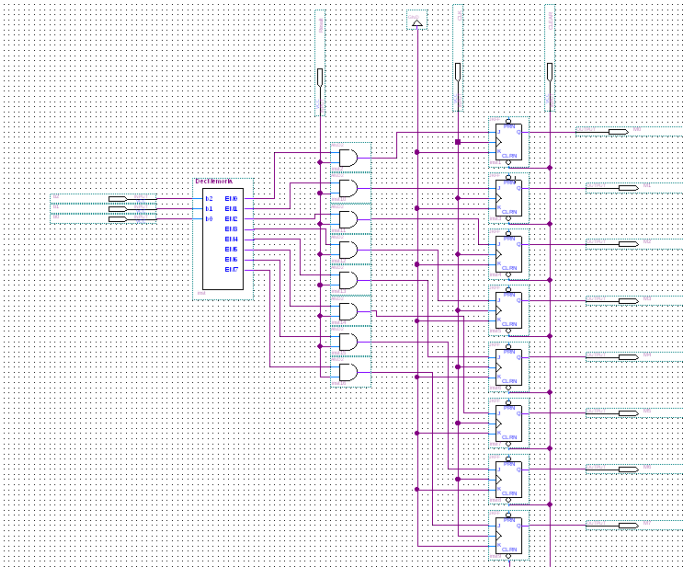


Figura 17. Descrição da memória RAM na EDA

3.9. Multiplexador do Display Dual de Sete Segmentos

No problema anterior, o DSS era responsável apenas por exibir a coordenada na forma de (letra, número) para o usuário. Em discussão, a turma decidiu por implementar um requisito do problema 3 na mesma matriz, sem ferramentas físicas adicionais. As razões para tal escolha foram a economia de material e de tempo refatorando o circuito físico e a aparente facilidade de implementação da solução proposta.

Para exibir o número correspondente ao endereço de memória atual e também as coordenadas, o multiplexador do DSS recebe como entrada todas as saídas do multiplexador do problema 2 e também as saídas do decodificador do contador de c_1 para decimal. Para realizar a com exatidão, o multiplexador também recebe uma frequência de clock de um período $t = 2s$, para tornar a exibição de dois dados em um mesmo display a mais amigável possível.

3.10. Unificação dos blocos em um projeto

Com todos os blocos prontos e devidamente testados em ambiente lógico, foram todos unidos em um único projeto e ligados como descrito no diagrama elaborado previamente. Tal tarefa foi destinada a todos os membros da equipe, como a justificativa de fortalecer o conhecimento sobre o problema e comparar e discutir os projetos em caso de eventuais discrepâncias.

3.11. Refatoração do circuito na protoboard

Nessa seção é mostrado o resultado da dedicação da equipe aproveitar ao máximo a plataforma física já existente, atribuindo as funções que puder ao circuito lógico. Para efeitos de comparação, constam abaixo a figura 18 e a figura 19, representando, respectivamente, os circuitos físicos utilizados no problema 2 e no problema 3, respectivamente.

Como é possível ver, a única mudança foi o acréscimo de 3 botões ao sistema. As funções desses botões são, respectivamente, limpar a memória, alterar o endereço de

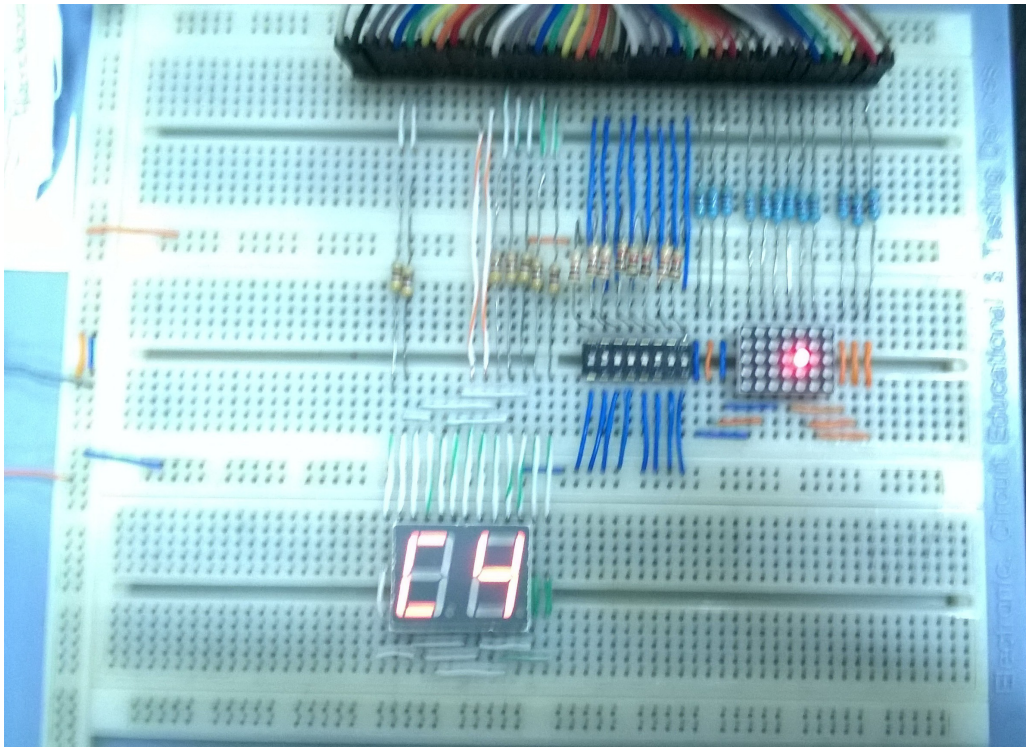


Figura 18. Circuito físico do problema 2

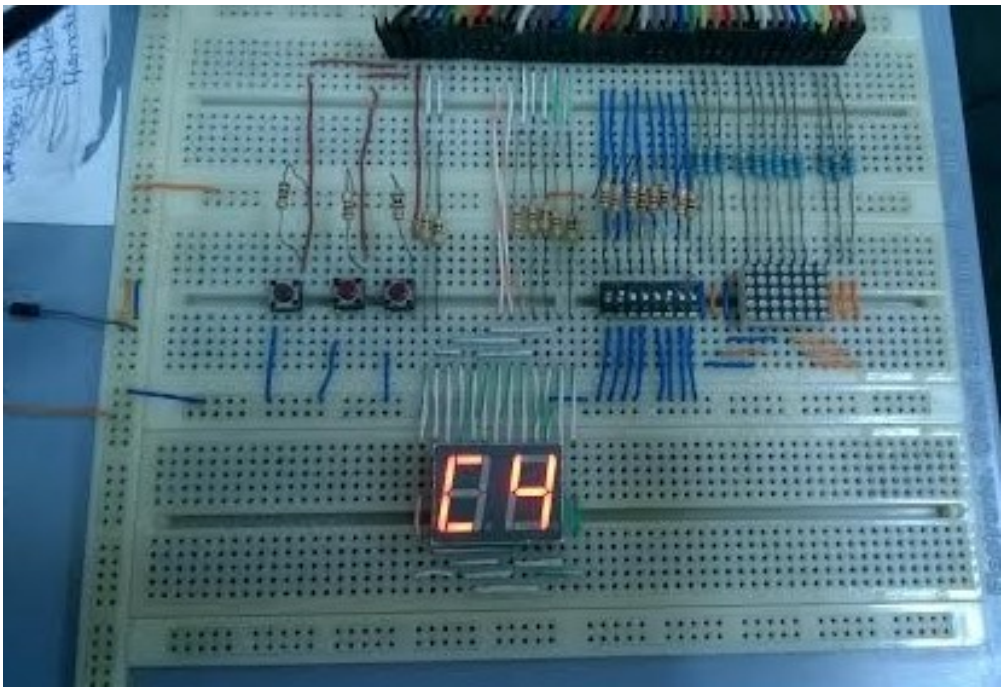


Figura 19. Circuito físico do problema 3

memória e ativar a comparação. Tais alterações foram feitas no Laboratório de Hardware um dia antes do dia marcado para a realização dos teste práticos.

4. Discussões e resultados

Nesta sessão serão abordados os principais resultados obtidos, os testes realizados e, além disso, serão abordados alguns detalhes que dizem respeito ao circuito lógico.

4.1. Testes lógicos no ambiente Quartus II

Como de praxe, antes realizar os testes físicos, foram realizados testes virtuais com o intuito de avaliar o funcionamento do circuito lógico. Para realizar esta análise, foram utilizadas ferramentas fornecidas pelo software da Altera, sendo estes os arquivos de WaveForm e o Simulator Tool.

Foram realizados uma série de testes em cada bloco componente do produto final. Todos apresentavam a eficácia pretendida, com a exceção da memória RAM. De início, tal componente não apresentou resultados compatíveis com os desejados e, ao analisar cuidadosamente, foi percebida que a abordagem inicial de usar FF's do tipo D com o D ligado em VCC e utilizar o $result$ do comparador multiplicado com o pulso do botão como clock não daria certo, uma vez que o pulso do botão, mesmo com o Debounce, se mostra irregular. Para corrigir esse erro, o FF foi substituído por um FF do tipo JK com o K aterrado, o J contendo a multiplicação do $result$ com a entrada do botão e a entrada de clock com o clock do sistema.

Como demonstrado na figura 20, ao receber $result$ em nível alto, estando o $clear$ em estado também alto, a célula de memória passa a armazenar 1 na coordenada apontada pelo contador(o valor em decimal apontado pelo contador corresponde ao endereço da célula de memória). Caso $result$ seja 0, o valor armazenado na célula atual é mantido. Se utilizando da operação de $clear$, toda a memória é limpa e volta ao seu estado primitivo.

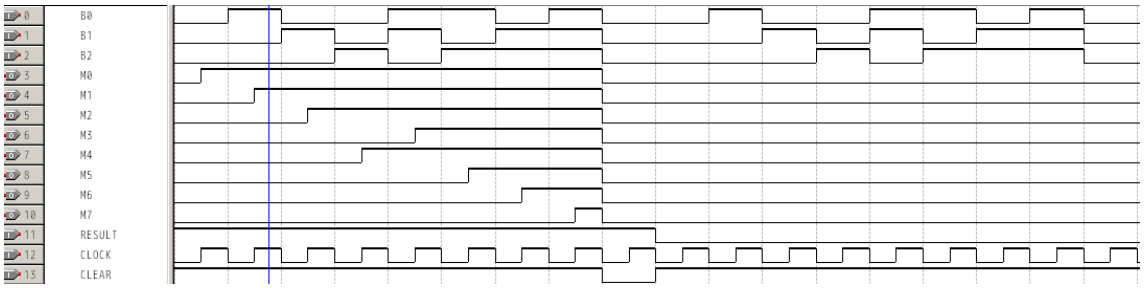


Figura 20. Teste da Memória RAM na EDA

Analisando os requisitos necessários e o comportamento demonstrado pelo dispositivo nos testes, concluiu-se que o mesmo corresponde ao desejado.

4.2. Funcionamento do Circuito

O circuito deve receber uma entrada oriunda de um barramento paralelo de oito bits, que deve ser diretamente impressa em um DSS, e compará-la com a memória presente no endereço atual, endereço esse que deve também estar sendo impresso em um DSS que alterna as duas informações (coordenadas e endereçamento) para exibição. Caso a entrada

e os dados da memória sejam iguais, o LED correspondente a coordenada inserida deverá ser aceso. O controle de endereçamento e de comparação é dado por meio de dois botões distintos. Além disso, a matriz deve voltar ao estado original quando o botão de $clear$ for selecionado.

4.3. Testes práticos

Antes de realizar os testes do circuito na Protoboard, fazia-se necessário fazer a demarcação dos pinos adicionais na ferramenta Quartus II e gravar o circuito lógico do mesmo na FPGA disponível no laboratório. Após a demarcação dos pinos, realizada com experiência adquirida no problema 1, ligou-se o circuito e a gravação foi feita.

Dispondo-se de todas as ferramentas necessárias, o teste foi realizado introduzindo-se todas as coordenadas possíveis e tentando gravá-las na memória. Inicialmente, ao por em prática todo o planejado, foram grandes as falhas. Exibição da coordenada errada, o DSS não alternava entre seus dois estados e a matriz de LED's não acendia mesmo enviando comparações válidas. Como um resultado das práticas de projeto usadas, boa parte da equipe tinha um alto conhecimento em relação a estrutura do circuito, então, em uma revisão de todo o projeto pelo grupo, percebeu-se que tudo não passou de um descuido na hora de interligar os blocos, pois haviam diversos fios desencaixados ou encaixados no lugar errado.

Após a detecção e correção dos erros citados, o circuito funcionou como o imaginado pelo grupo, exibindo, comparando e alternando tudo que deveria na forma que deveria.

5. Conclusão

Observando os requisitos que foram solicitados para o desenvolvimento deste sistema digital, é notório que todas as funcionalidades foram cumpridas. Isto se torna mais concreto devido ao sucesso alcançado através dos testes práticos realizados. Em suma, todos os pré-requisitos para a elaboração do projeto das coordenadas foram obtidos com totalidade.

Tendo sido revisado várias vezes e tentando melhorar quando possível, o grupo chegou num resultado simples e funcional, e, além de tudo, totalmente compatível com o problema anterior.

Além do aspecto técnico, houve uma melhora singular no trabalho em equipe a partir do uso das práticas de projeto citadas. Problemas organizacionais, de incompatibilidade e de centralização de conhecimento, outrora frequêntes, foram eliminados ou amenizados.

Apesar do circuito como um todo estar sendo melhorado, outras possíveis melhorias apontadas no problema anterior continuam não realizadas. Tal fato esclarece que tanto com iniciativas do grupo tanto como exigências do próprio problema, atualizações ainda são possíveis, como o uso de um teclado para a inserção das coordenadas e um aviso de erro mais chamativo, como um aviso sonoro(sugestão que persiste do problema anterior) e a automatização da função $comparar$ e da função $clear$, eliminando botões do sistema e o deixando sua interface com o usuário mais simples.

6. Referências

Nessa seção constam os trabalhos utilizados como base teórica que foram necessários para chegar a solução descrita nesse relat

Referências

Floyd, T. L. (2011). *Digital Fundamentals, 10/e*. Pearson Education India.

Tocci Ronald, J. (1997). Digital systems principles and applications.