

Problema 4: "Batalha Naval"

Turma P04

¹MI - 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, das disciplinas da turma, no período de 2015.1, na Universidade Estadual de Feira de Santana.

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 o famoso 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 maneiras similares aos seus primórdios, encontra-se, também, aplicativos para diversas arquiteturas que executam o mesmo processo, facilitando assim

Além da existência, como visto, deste tipo de jogo na forma de aplicativos móveis, existe a possibilidade, também, de desenvolver – utilizando circuitos digitais, visando criar um jogo lúdico de simulação 2D. É de se notar, portanto, a necessidade de utilizar um conceito muito importante na eletrônica digital, a saber, memória.

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 uma conexão de rede, de modo que o jogo fosse propriamente dito, de qualquer modo, contendo suas especificações técnicas 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 deslocamento, 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 sequência 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 contribuído 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 [?] 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 sequência de lógicas combinacionais e uma memória (flip-flops) [?]. Na figura

?? podemos observar o circuito típico para uma máquina de estados. Vale acrescentar que a quantidade de estados que uma máquina pode ter é dada por 2^n , sendo n o número de Flip-flops.

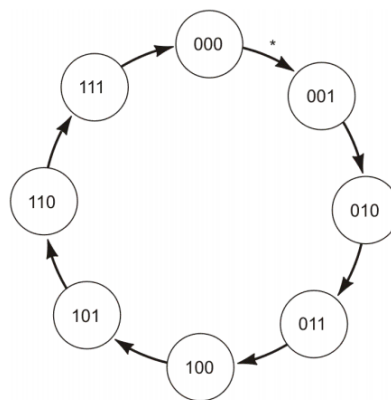


Figura 1. Diagrama de estados de um contador de 3 bits [?]

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

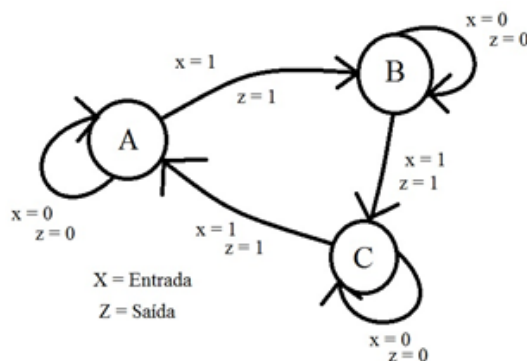


Figura 2. Exemplo de diagrama de estados usando uma máquina Mealy

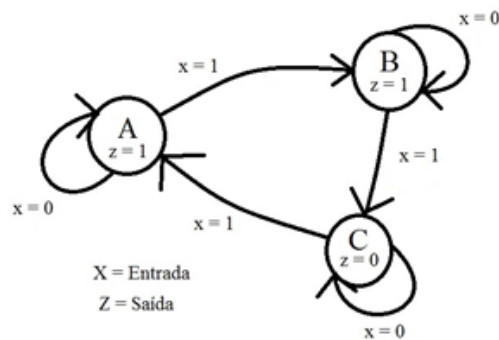


Figura 3. Exemplo de diagrama de estados usando uma máquina Moore

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 ?? 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 alto de maneira intermitente.

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

2.3. Registrador de Deslocamento

No universo dos circuitos digitais, principalmente no que diz respeito aos Flip-Flops, encontramos diversas aplicações. Uma das aplicações é na forma de registrador de deslocamento, os quais são importantes e aplicados em diversas situações, como armazenamento e transferência de dados, apesar de alguns casos poderem ser utilizados como tal. Na figura ?? podemos ver um circuito para um registrador de deslocamento de quatro bits com entrada serial de dados.

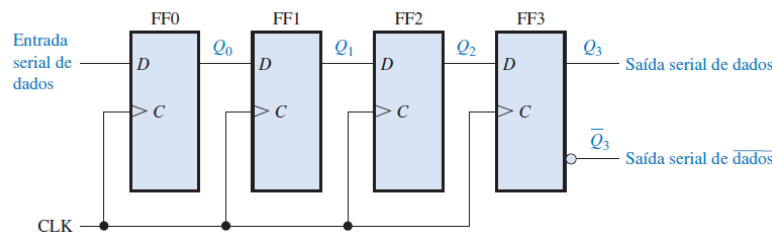


Figura 5. Representação de um registrador de deslocamento de 4 bits [?]

2.4. Flip-Flop T

São diversos, os tipos de registradores de deslocamento, na figura ?? 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 subseção.

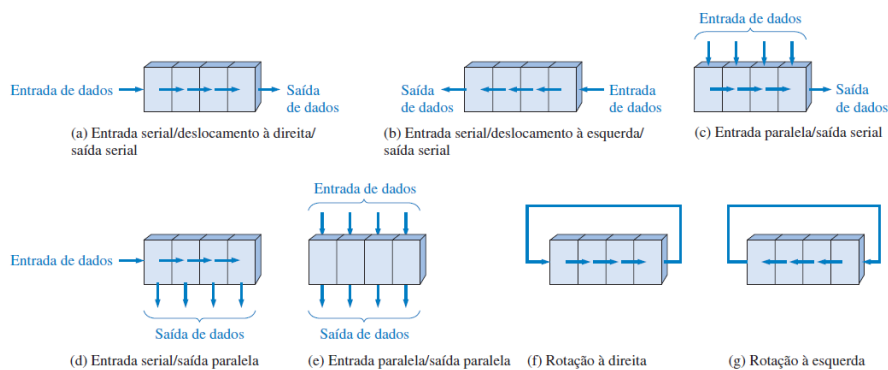


Figura 6. Diferentes tipos de registradores, segundo [?]

2.4.1. Registrador de deslocamento com entrada serial e saída paralela

Como o prefixo designa, neste tipo de registrador a entrada de dados é de forma serial, e a saída é paralela. Desta forma, segundo [?] uma vez armazenados os dados, cada bit pode ser acessado individualmente e todos os bits são disponibilizados simultaneamente, em vez de um bit de cada vez como na entrada serial. Na figura ?? podemos ver uma exemplificação para este circuito. Notem que neste tipo de registrador cada bit está disponível.

2.5. 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 e um custo maior. Já na comunicação serial os dados são transferidos

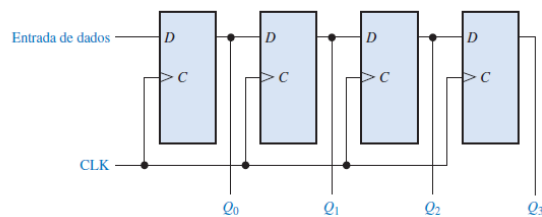


Figura 7. Registrador Serial-Paralelo

2.5.1. Transmissão de Dados

Antes de falarmos acerca de como ocorrer a transmissão de dados de forma serial, é importante lembrar que existem 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 ?? 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 ?? podemos ver a função de cada pino na porta *DB9*.



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

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 a velocidade, seguido pelos bits de dados, um bit opcional de paridade e um ou mais bits de parada *stop bits* [?]. Para analisar esta forma de comunicação de maneira mais concreta, vale acrescentar, que quanto o bit de *start* vai para o alto, os dados vão para o baixo, caracterizando assim, o bit de *start*.

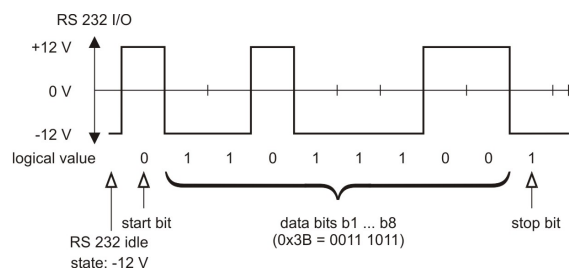


Figura 10. Características de uma transferência de dados serial.

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.

2.5.2. Padrão RS-232

Segundo (<http://www.c2o.pro.br/automacao/x834.html>) RS-232 é um padrão definido pela *EIA-Electronic Industries Association* para os dispositivos usados para comunicação serial. Este padrão está disponível em três versões, quaisquer sejam, (A, B e C). Cada uma especifica uma faixa de voltagens para o sinal ligado ou desligado. Por exemplo, na versão RS-232 para representar o nível lógico alto utiliza-se a tensão de -3V a -12V, enquanto para representar o nível lógico baixo utiliza-se a tensão de +3V a +12V. É importante notar que a representação do nível lógico alto é a tensão negativa e a do nível lógico baixo é a tensão positiva. Vale lembrar que o padrão RS-232 utiliza o modelo convencional de tensão.

2.5.3. Conversor de nível MAX-232

Os dispositivos eletrônicos utilizam níveis de tensão diferentes para identificar o nível lógico baixo e o nível lógico alto, dependendo da tecnologia que está sendo empregada. Desta forma, quando se vai analisar o padrão que o mesmo utiliza, pois depende da especificação, será necessário verificar a necessidade de um conversor. Por exemplo, vamos supor que queremos fazer uma comunicação entre o computador pessoal e o nível lógico baixo de 2V a 5V representam o nível lógico alto. Percebe-se, assim, uma incompatibilidade entre o padrão RS-232 e o TTL, desta forma é necessário fazer um conversor. Todavia, existem alguns dispositivos que fazem o conversor entre os níveis utilizados pelo padrão RS-232 para TTL, e vice-versa. Na figura ?? podemos ver uma configuração comum utilizando um conversor RS-232.

2.5.4. Software de comunicação

Para poder coletar os dados diretamente do usuário e enviá-los para o circuito, os mesmos precisam passar por um tratamento. Entre o caminho percorrido está abrir a

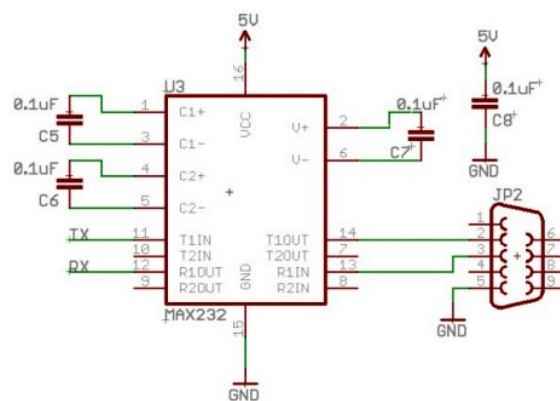


Figura 11. MAX-232: Conversor de RS-232 para TTL.

porta de comunicação, converter os dados coletados por o usuário em binário e, finalmente, enviá-los através do DB – 9.

Para ajudar as realizar as tarefas correspondentes ao manuseio de permissões e envio de dados, foi utilizada a biblioteca RS232, construída em linguagem C e disponível documentada em <http://www.teuniz.net/RS-232/>.

3. Metodologia

Utilizando-se, mais uma vez, da base teórica fornecida pela metodologia PBL, os alunos componentes das etapas do problema, bem assim como ter uma ideia de como iniciar o desenvolvimento do mesmo.

Os membros da sessão, solucionando as questões e outros quando possível durante o processo, buscaram tomar decisões de projeto para a construção das soluções do problema 4 do MI – Circuitos Digitais, começando pela formalização e definição de padrões de projeto e de um di-

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, cau-

Dado o sucesso da abordagem no problema anterior, foi de comum acordo que a turma deveria modularizar o problema e dividir tais módulos entre pequenos subgrupos, sendo que cada subgrupo deveria enviar um pequeno relatório contendo

3.2. Elaboração do Diagrama

Iniciado na primeira sessão do problema 4 e finalizado na terceira sessão do mesmo problema, o diagrama teve como objetivo fazer com que todos os alunos tivessem consciência plena do funcionamento do circuito, da entrada saída. Descrevendo as relações entre os blocos e a função, a quantidade de entrada e a quantidade de saída de cada um deles, também, padrões de nomenclatura, o diagrama descreve o que cada subgrupo do-

O resultado desse processo de descrição em alto nível do projeto pode ser conferido na figura ??, que destaca também as tarefas de cada subgrupo.

Por exemplo, o subgrupo vermelho $\tilde{\text{A}}\text{C}$ responsável pela elaboração dos dados correspondentes ao comparador Bloco do Mux-Coordenada.

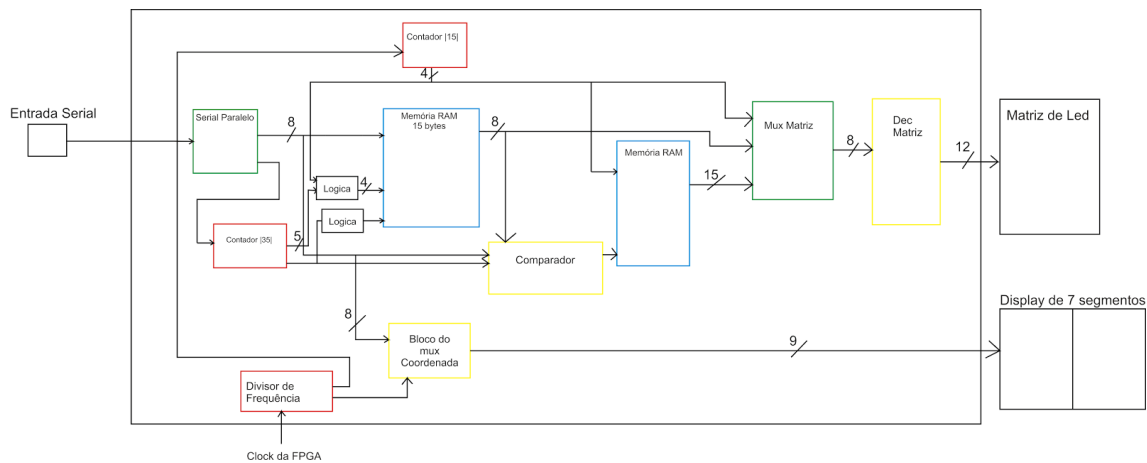


Figura 12. Diagrama de blocos do Problema 4: [COLOCAR NOME DO PROBLEMA]

3.3. Memórias RAM

Como explicitado nas regras determinadas pela turma, o jogo será composto de 15 jogadas de preenchimento do tabuleiro mais 20 jogadas na qual o jogador tentará acertar o barco do oponente. A partir desse dado, $\tilde{\text{A}}\text{C}$ percebeu que precisam ser armazenados 15 posições diferentes, sendo cada uma correspondente a um pixel de um barco que foi inserido no tabuleiro. Se a conclusão é que $\tilde{\text{A}}\text{C}$ precisa de uma memória RAM de 1 byte, e que a memória deve conter 15 dessas posições.

Tendo em vista o mercado e a praticidade para uso em grandes escalas, a equipe resolveu construir uma memória com 16 células de 1 byte. Tal decisão foi tomada a partir dos princípios que o $\tilde{\text{A}}\text{C}$ adotou de endereçamento de memória, onde a 16ª célula ficaria ociosa.

Diferentemente do problema anterior, no Problema 4, o endereço de memória foi integrado na mesma, objetivando uma maior congruência de projeto e uma visibilidade mais estética, como pode ser visto na figura ?? a memória elaborada conta com 8 bits de entrada, um bit de clock, um bit de endereço e 4 bits de endereçamento. Essa última característica segue o padrão crescente de 0 a 15, ou seja, o endereço 0 corresponde a primeira célula de memória, a segunda e assim por diante.

3.3.1. Célula de Memória

Para elaborar o item anterior, fez-se necessária a elaboração de uma unidade menor de memória, sendo esta a unidade básica utilizada no projeto, 1 byte. Em suma, cada célula de memória estará selecionada.

Buscando melhorar o desempenho da memória, em suas células foram implementados diferentes dados duplicados e alterados indevidamente os mesmos. O segundo mecanismo pode ser visto na figura da Q sua entrada I forem diferentes, e, além disso, o mecanismo anterior deve estar em nível lógico 1, ou seja: $T = Z(Q \oplus I)$.

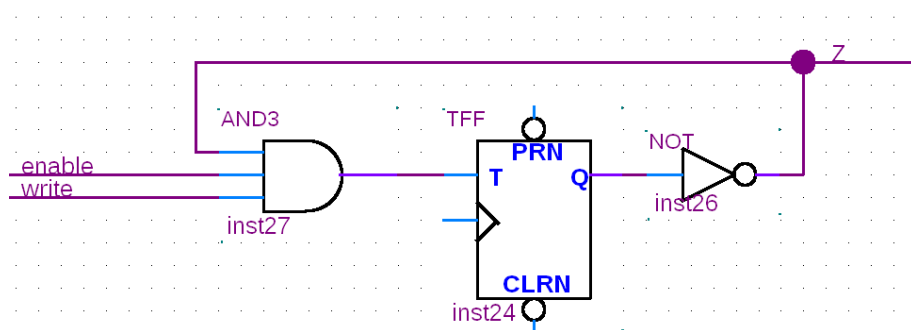
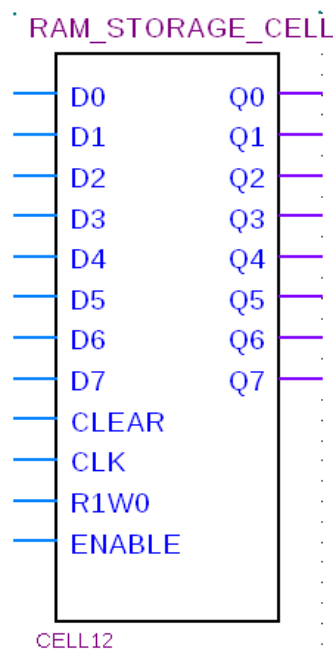
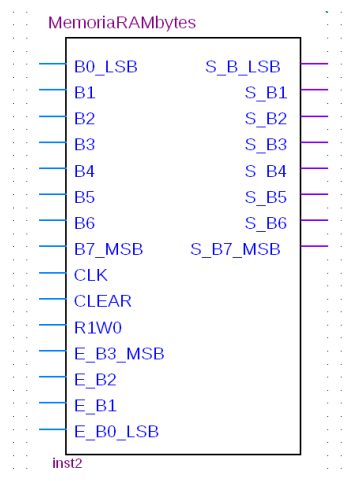


Figura 15. Máquina de estados para máquina de uma moeda a 50¢.

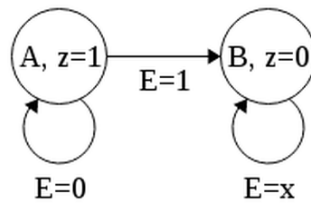


Figura 16. Diagrama de estados para máquina de uma única ação.

3.3.2. Comunicação Serial

Um dos vários requisitos propostos pelo problema quatro, diz respeito à comunicação serial que deveria acontecer entre o computador pessoal e o Circuito Digital desenvolvido, onde esta, haveria de ser configurada no modo *start – stop* assíncrono, sendo que o frame de dados que seria utilizado nesse modo seria composto por 11 *bits*. Assim, pode-se notar que a transferência de dados ocorreria utilizando alguma das portas seriais existentes em um computador pessoal. Desta forma, foi definido entre os membros do grupo que seria utilizada a porta serial *DB9* para fazer essa comunicação no projeto que estava sendo desenvolvido, ou seja, seria utilizado o padrão *RS – 232* de comunicação de dados.

Depois de definir que seria utilizado a porta *DB9*, era necessário criar um circuito que fosse capaz de identificar quando essa transferência estivesse acontecendo, ou seja, um circuito que fosse capaz de reconhecer quando o computador estava enviando um *frame* de dados para *FPGA*. Este circuito deveria ser capaz, ainda, de transformar os dados que estavam sendo transferidos de forma serial, para o formato paralelo, tornando assim possível a leitura correta dos dados, possibilitando o seu uso em outra parte do circuito. A FIGURA AQUI mostra os blocos lógicos para este circuito. Nesta FIGURA podemos observar três blocos de grande importância, a saber: bit detectado, a entrada *Enable* deste bloco mudará para *Ativo*, alto ativando assim o bloco *RegistraDados*, o qual iniciará a leitura do frame de dados enviado pelo computador, tornando possível uma nova leitura. Sendo que a entrada *Stop* correspondente ao *stop-bit*, ou seja, o último bit do frame de dados, indicando que aqueles determinados dados foram lidos com sucesso.



Figura 17. Circuito RegistraDados

Já o bloco Contador, tem uma função por demais simples, ele é ativado quando se inicia a leitura de um *frame* de dados. Ele é um contador de



Figura 18. Circuito AtivaRecepção.

modo 10, ou seja, é capaz de gerar dez estados distintos. Ele é utilizado para ativar a saída dos dados lidos de forma serial, ou seja, depois que os bits vindos do canal de comunicação serial forem dados do registrador paralelo, pois esse valor decodificado gerará um nível lógico alto, onde esse valor alto é a entrada do clock do registrador paralelo. Essa ideia tornará mais claro depois que o circuito interno do bloco RegistraDados for observado. É importante salientar

O bloco RegistraDados, dentre os blocos descritos, é o mais importante, pois ele é responsável, por, de fato, armazenar e transformar os bits do formato serial para paralelo. O circuito interno para este bloco pode ser observado no ANEXO 1. O funcionamento deste bloco é bem simples. Depois de ativado, inicia-se a leitura e conversão dos dados para o formato paralelo a cada pulso do clock, sendo que o registrador de entrada de dados, está conectado, também a outro registrador de saída de dados, onde cada Flip – Flop de acordo com seu valor posicional estão conectados entre si. Observem que os dados que estão no registrador de saída de dados, estarão disponíveis, realmente, quando o valor do contador estiver em 9, evidenciando assim, que todos os bits foram lidos.

Vale asseverar, que os bits advindos da porta serial estão sendo transferidos a uma taxa de 9600 bits por segundo, ou 9,6 kbs. Isto foi definido no software que fora desenvolvido tendo como objetivo justamente fazer esta interface entre o usuário e o circuito projetado.

Quando se fala em comunicação serial, o sincronismo utilizado para recepção dos dados, deve ser observado de forma minuciosa, visando a leitura correta dos dados que estão no frame. Desta forma, tentando resolver o problema do sincronismo no projeto desenvolvido foi utilizado o seguinte pensamento: como a frequência do clock da FPGA é de 32,768 MHz e a taxa de bits por segundo utilizada no programa é de 9600 bps, fazendo $32768000/x = 9600$, teremos X valendo aproximadamente 3413, ou seja, usando um contador de modo 17 para alternar um Flip-Flop Teobter uma frequência 34 vezes menor, sendo que em seguida foi necessário inserir um contador de

3.4. Unificação dos blocos em um projeto

Diferentemente do problema anterior, no problema 4 a tarefa dada como prioridade a um subgrupo, assim como a tarefa de refatoração do circuito e montagem do cabo de transmissão foram dados como prioridades a outros grupos. Apesar de estarem responsáveis, o respectivo

grupo não tirava de ninguém a responsabilidade de montar por si só o circuito novamente, em busca de aprender por inteiro o funcionamento do mesmo.

Ao receber os blocos, não foi difícil montá-los em um único projeto, uma vez que, semelhante ao problema anterior, teve-se ao lado um diagrama com todas as conexões padrão de cada unidade lógica presente, e, além disso, todas já estavam devidamente

3.5. Refatoração do circuito na protoboard

Para poder comportar a transmissão serial, o circuito físico precisava ser modificado, adicionando suporte a um conversor de tensão, alguns capacitores e um cabo de transmissão serial. Além disso, muitos dos componentes do problema anterior puderam ser retirados, muitos deles relacionados a entrada de dados, tais como o DIP Switch e resistores de *pull – down*. Um dos itens adicionados a protoboard foi o MAX-232, um circuito que converte a tensão vinda da porta DB9 do computador para a tensão na faixa TTL. Para funcionar corretamente, o mesmo foi adicionado ao circuito seguindo as instruções da figura ??

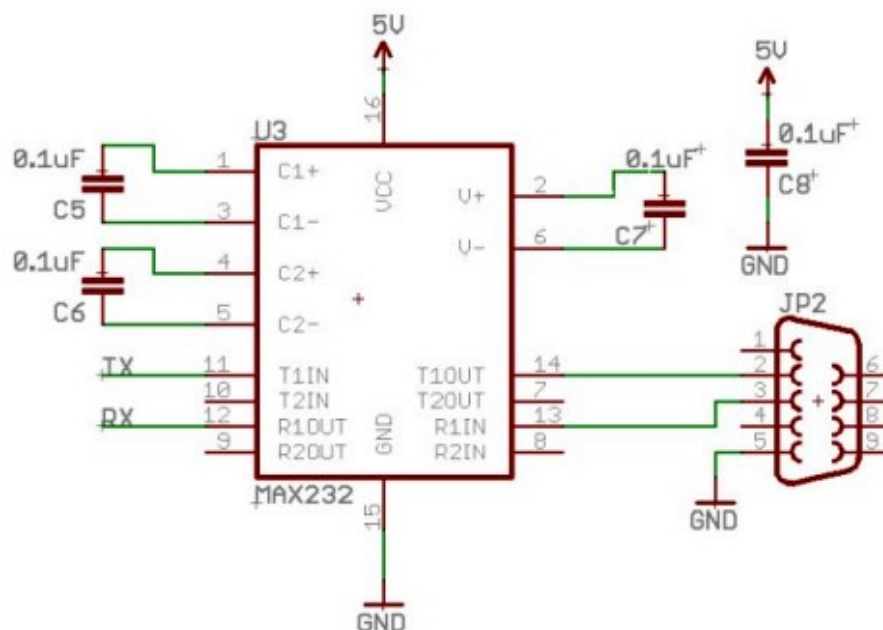


Figura 19. Circuito do MAX-232.

Para o cabo de comunicação, foi utilizado um cabo antigo e uma porta DB9 fêmea. No DB9 foram soldados fios nas entradas TX, RX e GND, mesmo sabendo que apenas a porta TX seria usada, uma vez que os dados seguiriam uma via única do computador para o circuito.

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

Passados como tarefas para os subgrupos, os testes lógicos foram realizados bloco – a – bloco, tentando diminuir a chance de qualquer erro de implementação lógica. Para os testes em paralelo, dado o fato que esses são os primeiros que o grupo não trabalhou ou conheceu a escala exigida.

4.1.1. Testes da Memória RAM

Inicialmente, foi feito um conjunto de testes básicos para verificar se a memória estava sendo capaz de distinguir os momentos de gravação/leitura gravando apenas um dado (EB1=0, EB2=0, EB3=0). Avaliando os resultados, percebe-se uma constante nessas ondas 0,1,2,3,4,5,6,7 que equivale aos primeiros valores enviados no modo de gravação, apontando os

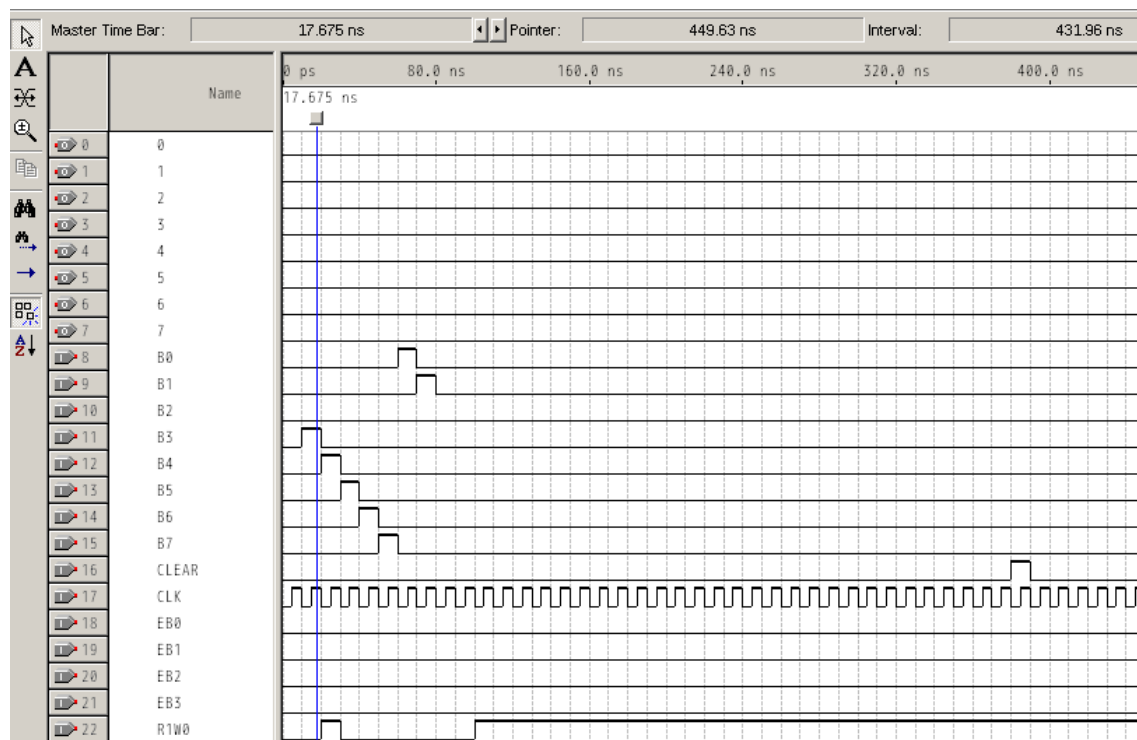


Figura 20. Teste de persistência e modo de operação

Prosseguindo os testes, o teste anterior foi repetido em uma escala de variação maior e utilizando-se todos os endereços disponíveis da memória RAM. O esperado é que a saída equivalesse ao primeiro valor de entrada em cada uma das células, e para cada período de clock entre 1 e 15, as entradas deveriam equivaler à saída quando a memória estiver em modo de leitura. Ou seja, se em loop de entradas constantes, quando for

4.2. Funcionamento do Circuito

Segundo os requisitos do Problema 4, o usuário deve entrar com coordenadas em um software que se encarregará de direcionar os dados para o circuito, no qual o tratamento de dados será feito. A partir desses dados, existem dois momentos:

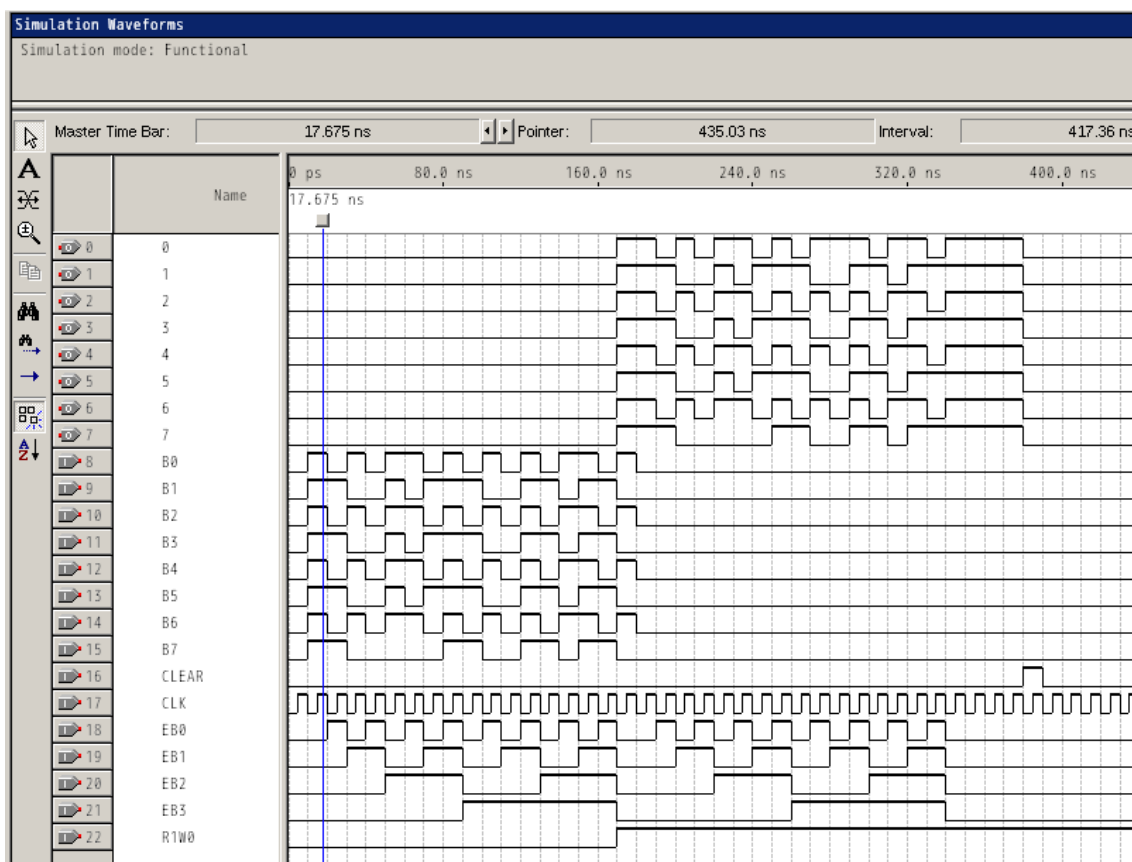


Figura 21. Teste de persistência e modo de operação

- Modo Planejamento: O usuário insere 15 coordenadas nas quais deseja que fiquem seus barcos, inicialmente todos os pixels da matriz estão apagados.
- Modo Jogo: O usuário tenta acertar os pixels que representam barcos, caso acerte, o pixel acende. O número máximo de tentativas é de 20, e caso não consiga encontrar todos os barcos, o jogo é encerrado.

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 no *3gicodomesmonaFPGA* disponível no laboratório. Dada a seguinte configuração de pinos do problema 3 para o problema 4, não foi possível

Dispondo-se de todas as ferramentas necessárias, o teste foi realizado introduzindo-se todas as coordenadas possíveis via software...

5. Conclusão

Observando os requisitos para o desenvolvimento deste circuito digital, não notável que nem todas as funcionalidades foram alcançadas. A configuração *start – stop* assíncrona não foi obtida com sucesso, mesmo depois de realizados diversos testes no circuito projetado para realizar esta função, não se obteve sucesso, tornando, a partir disso, inviável a utilização do modo jogo a partir da comunicação serial. Não se sabe ao certo o que motivou esta falha, uma suposição se refere ao fato de que possivelmente foi utilizada uma frequência errônea para se ler os dados oriundos da porta serial.

Afora este problema relatado acima, todas as outras funcionalidades estão funcionando conforme o solicitado, ou seja, a implementação de resolução do problema está correta. Isto se torna mais concreto devido ao sucesso da implementação através dos testes realizados. Tendo sido revisado e simplificado várias vezes utilizando-se ferramentas mais simples, funcionais e profissionais. De maneira semelhante aos problemas anteriores, o circuito final deste problema manteve uma boa velocidade de inicialização e outra partida de jogo batalha naval.

Vale acrescentar, que os conteúdos estudados para resolução do problema, os quais de uma certa maneira gerou conhecimento, tanto em grupo quanto individualmente, são de fundamental importância na área da eletrônica, principalmente em relação à comunicação serial de dados, a qual é amplamente utilizada hoje em dia. Não se pode esquecer, é claro, das máquinas de estados, as quais são utilizadas em grandes aplicações. Sendo assim, estes conteúdos, e outros mais, irão contribuir de forma significativa para

6. Referências

Nessa seção constam os trabalhos utilizados como base teórica que foram necessários para chegar à solução descrita nesse relatório.