

Alan Araújo dos Reis - 5096  
Gabriel Rodrigues Marques - 5097

## INTRODUÇÃO

Neste trabalho prático, foi proposto o desenvolvimento, implementação e simulação de um sistema de jogo de loteria por meio do uso de lógica sequencial/combinacional e máquina de estados finita em Verilog, exibindo os resultados obtidos em uma FPGA. O objetivo final deste trabalho, além de simular um jogo de loteria, é determinar, de forma automática, se um jogo foi vencedor e, em caso afirmativo, identificar qual prêmio foi ganho.

O jogo de loteria proposto consiste na escolha sequencial de 5 números, e a loteria também sorteia outros 5 números. O número sorteado pela loteria é fixo para facilitar questões de implementação e é composto pela combinação de duas matrículas (50967). O quinto número funciona como uma segunda chance para ganhar os prêmios. Na loteria existem os seguintes prêmios com as respectivas condições de vitória:

***Prêmio 0 - Não atender a nenhuma das condições de vitória.***

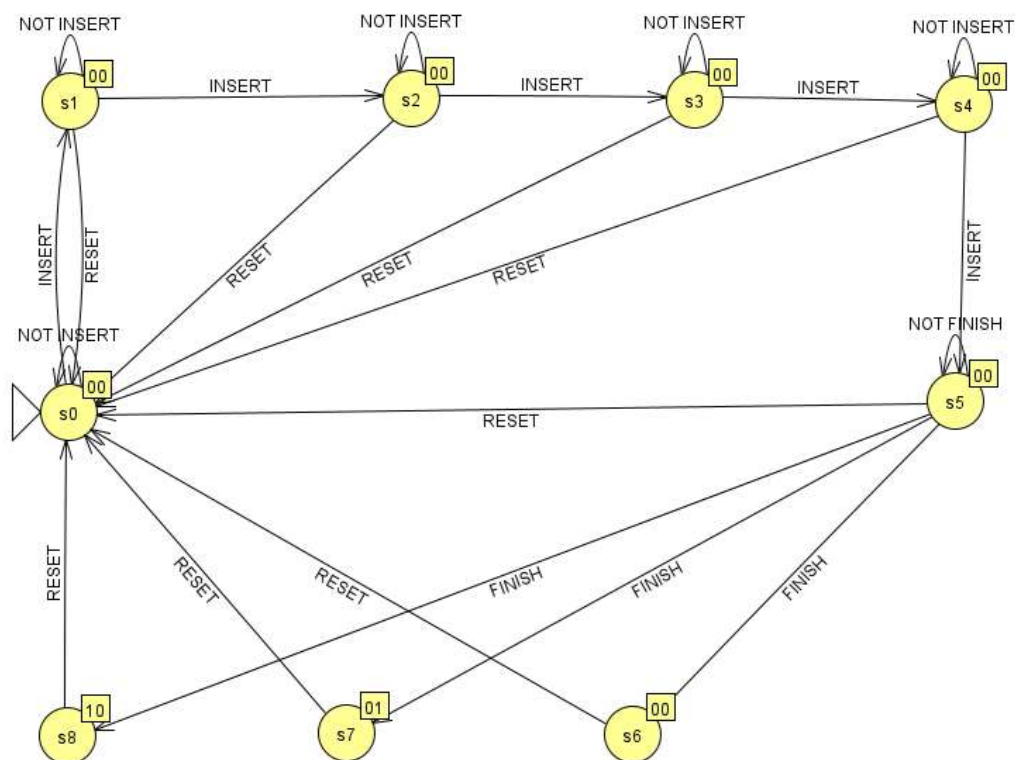
***Prêmio 1 - Acertar os 4 primeiros números sorteados.***

***Prêmio 1 - Acertar 3 números e o quinto número.***

***Prêmio 2 - Acertar 2 números consecutivos e o quinto número.***

## DESENVOLVIMENTO

Inicialmente, foi elaborado um diagrama de transição de estados que apresenta o comportamento do circuito do sistema de jogo de loteria, foi utilizado a ferramenta JFlap para sua elaboração. O diagrama representa um “fluxo” de controle do jogo, transicionando os estados e verificando os números inseridos e qual foi o resultado do jogo realizado, retornando como saída o prêmio ganho.



**Figura 1.** Diagrama de transição de estados da máquina de estados finita.

- s0 - Aguardando inserção do primeiro número.*
- s1 - Aguardando inserção do segundo número.*
- s2 - Aguardando inserção do terceiro número.*
- s3 - Aguardando inserção do quarto número.*
- s4 - Aguardando inserção do quinto número.*
- s5 - Finalizando e verificando o resultado do jogo.*
- s6 - Estado Prêmio 0.*
- s7 - Estado Prêmio 1.*
- s8 - Estado Prêmio 2.*

**INSERT** - O número inserido é válido e o sinal de INSERT está ativo, vai para próximo estado.

**NOT INSERT** - O número inserido é inválido ou o sinal de INSERT não está ativo, permanece no estado.

**FINISH** - Sinal de FINISH está ativo, vai para próximo estado

**NOT FINISH** - Sinal de FINISH não está ativo, permanece no estado.

Com o diagrama de transições de estados devidamente elaborado, o módulo Loteria foi desenvolvido utilizando a linguagem de descrição de hardware Verilog. No módulo está contida toda a lógica necessária para o jogo de loteria. Nas tabelas (Tabela 1 e 2) abaixo, pode ser verificado os sinais de entrada, saída e variáveis auxiliares que foram utilizadas na implementação do módulo.

Nome	Tamanho	E/S	Descrição
clk	1 bit	Entrada	Pulso de clock do sistema.
num	4 bits	Entrada	Número entre 0 e 9 que compõem o número do jogo.
insert	1 bit	Entrada	Sinal de controle insert que sincroniza a entrada de um número na máquina de estados. Deve ser acionado mediante a inserção de cada novo número.
finish	1 bit	Entrada	Sinal de controle que indica o fim da inserção dos números de um jogo, apresentando o seu resultado.
reset	1 bit	Entrada	Sinal reset responsável por colocar a máquina de estados de volta ao seu estado inicial.
prm	2 bit	Saída	Prêmio que foi ganho no jogo.
LEDR	9 bits	Saída	Estado atual da máquina de estados.
HEX0	6 bits	Saída	Quinto número inserido.
HEX1	6 bits	Saída	Quarto número inserido.
HEX2	6 bits	Saída	Terceiro número inserido.
HEX3	6 bits	Saída	Segundo número inserido.
HEX4	6 bits	Saída	Primeiro número inserido.
HEX5	6 bits	Saída	Separador “-”.
HEX6	6 bits	Saída	Prêmio que foi ganho no jogo.
HEX7	6 bits	Saída	Letra “P”.
LEDG	1 bit	Saída	Vitória ou derrota.

**Tabela 1.** Descrição dos sinais do módulo Loteria.

Nome	Tamanho	Descrição
b0, b1, b2, b3, b4	4 bits/número	Números entre 0 e 9 que compõe o número do sorteio
s0, s1, s2, s3, s4, s5, s6, s7, s8	4 bits/estado	Estados da FSM.
num0, num1, num2, num3, num4	4 bits/número	Números entre 0 e 9 inseridos que compõem o número do jogo.
hits	3 bits	Número da sequência de acertos.
auxHits	3 bits	Número da maior sequência de acertos.
lastTrue	1 bit	Acertou ou errou o último número inserido.
win	1 bit	Vitória ou derrota.
p0	2 bits	Prêmio ganho no jogo.
sdm	12 segmentos 7 bits	Mapa de segmentos do display.
lrm	8 estados de LED 8 bits	Mapa de LED's.

**Tabela 2.** Descrição de variáveis auxiliares do módulo Loteria.

O módulo Loteria possui três blocos de código principais, sendo eles: bloco de inicialização, bloco da máquina de estados e bloco de atualização de saídas.

**Bloco de Inicialização:** inicializa todas as variáveis auxiliares do código, definindo estado inicial do módulo e valores padrão.

**Bloco da Máquina de Estados:** realiza o controle de fluxo da máquina de estado e do jogo com base nas entradas e no estado atual a cada borda de subida do clock. Aqui é realizada toda lógica do jogo, inicialmente verificando a validade e igualdade de cada número inserido e ao final as condições de vitória que direcionam ao prêmio ganho.

**Bloco de Atualização de Saídas:** atualiza os valores das saídas sempre que elas são alteradas, definindo prêmios, números inseridos e se o jogo foi vencido.

Os objetivos opcionais não foram implementados, para que outros aspectos fossem melhor lapidados. As entradas inválidas foram tratadas com a permanência no estado atual mesmo que o sinal de inserção fosse acionado, o estado só avança quando um número de 0 a 9 é inserido. Mais detalhes sobre como cada bloco de código funciona pode ser visto no código e no vídeo enviado juntamente com esta documentação.

Finalmente, com o módulo implementado foi preciso configurar todo ambiente para trabalhar com a FPGA. Foi utilizado uma FPGA Cyclone IV da Altera, modelo EP4CE115F29C7, e dois programas: Intel Quartus Prime Design Software e Terasic DE2-115 System Builder. No desenvolvimento, não foram necessárias mudanças no código previamente implementado e as conexões das entradas e saídas do módulo com a FPGA foram devidamente feitas, realizando sua compilação e síntese.

## RESULTADOS

Como resultado tem-se um sistema de jogo de loteria funcional, que atende ao que foi solicitado. Além disso, alguns adicionais foram implementados como é possível ver na figura abaixo (Figura 2). Além do prêmio ganho ser exibido, um LED em verde é aceso em caso de vitória. Nos displays é exibido a sequência de números inseridos, que é atualizado a cada inserção. Os estados também estão sendo exibidos por meio de LED's, acendendo uma quantidade igual ao estado atual, isso foi feito para fins de controle.



Figura 2. FPGA.

**AMARELO** - Prêmio ganho (P0, P1, P2).

**VERDE** - Números que compõem o jogo.

**AZUL** - Vitória?

**ROSA** - Estado atual.

**ROXO** - Sinais de controle INSERT, FINISH, RESET.

**VERMELHO** - CLOCK.

	JOGO					SORTEIO					VENCEU	PRÊMIO
	N0	N1	N2	N3	N4	B0	B1	B2	B3	B4	S/N	[0, 1, 2]
1	5	0	9	6	7	5	0	9	6	7	S	1
2	0	0	0	2	7	5	0	9	6	7	N	0
3	7	0	9	3	7	5	0	9	6	7	S	2
4	1	0	9	6	7	5	0	9	6	7	S	1
5	9	0	1	6	7	5	0	9	6	7	N	0

Tabela 2. Exemplos utilizados no vídeo da FPGA.

**OBS.:** mais detalhes sobre o desenvolvimento e resultados podem ser vistos no vídeo que foi enviado juntamente com esta documentação e no módulo desenvolvido, que contém diversos comentários ao longo do código.

## **CONCLUSÃO**

Por fim, com o término do desenvolvimento, implementação e simulação, utilizando todas as ferramentas disponíveis para sua execução, este trabalho foi bem-sucedido, atendendo as especificações do projeto, em seu objetivo final: simular um jogo de loteria, com a verificação/determinação automática de um jogo e de qual foi o prêmio ganho. O projeto demonstrou a aplicação prática de diversos conceitos trabalhados ao longo da disciplina CCF 251 - Introdução aos Sistemas Lógicos Digitais. Além disso, reforçou e ampliou nosso conhecimento em relação ao desenvolvimento de módulos Verilog e implementação e simulação em FPGA.

## **REFERÊNCIAS**

- [1] R. Katz, G. Borriello, Contemporary Logic Design, 2ª edição, Prentice Hall, 2004;
- [2] TANENBAUM, A.S. Organização Estruturada de Computadores. 5. ed. Editora Pearson Prentice Hall, 2007;
- [3] HDLBits — Verilog Practice. Disponível em: <[https://hdlbits.01xz.net/wiki/Main\\_Page](https://hdlbits.01xz.net/wiki/Main_Page)>
- [4] Github. Disponível em: <<https://github.com/gabridulol/CircuitosSequenciais>>;
- [5] Icarus Verilog;
- [6] JFlap;
- [7] Intel Quartus Prime Lite Edition Design Software;
- [8] Terasic DE2-115 System Builder;