



**UNIVERSIDADE
FEDERAL DO CARIRI**

**UNIVERSIDADE FEDERAL DO CARIRI
CENTRO DE CIÊNCIAS E TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**Projeto 01 - Guess The Number
Professor: Ramon Santos Nepomuceno**

**Wanderson Faustino Patricio - MAT: 2022005052
Francisco Anderson Maciel Cruz - MAT: 2022005876**

1 Apresentação do projeto

Este projeto introduz o conceito de um jogo de adivinhação de 4 bits. Duas pessoas participam da atividade, competindo para acertar um número aleatório de 4 bits. Cada jogador faz palpites alternados e recebe um feedback sobre se o número que eles sugeriram é maior, menor ou igual ao "número mágico" definido pelo programador. O objetivo principal é ser o primeiro a adivinhar corretamente o número em questão e, assim, conquistar a vitória no jogo. Empregamos o software Logism para criar os circuitos lógicos que implementam as funções essenciais de forma que o layout inicial do jogo é apresentado da seguinte forma.

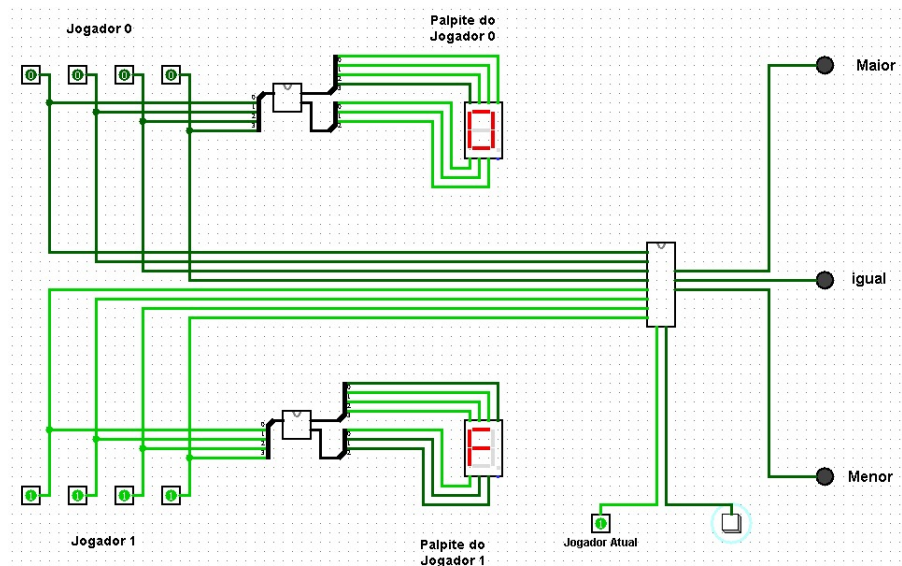


Figura 1: layout do jogo

Para o funcionamento do jogo utilizamos 3 circuitos principais: um circuito decodificador para o display de 7 segmentos, um circuito comparador entre dois números e um circuito multiplexador (ou seletor) para alternar as rodadas entre os jogadores.

2 Circuito Decodificador

2.1 Motivação para o uso do circuito decodificador

A entrada de dados para o circuito principal do game é feita através de 4 pinos de entrada, que podem ter seus valores alternados entre 0 e 1. Cada combinação distinta de valores nos pinos de entrada corresponde a um número binário diferente, sendo ao total 16 combinações (0 ao 15).

Para tornar o game mais acessível e de fácil visualização, torna-se necessária a utilização de um display (no nosso caso um display de 7 segmentos) para informar de maneira mais direta ao usuário qual número está sendo tomado como palpite na rodada. Com o display é possível imprimir qualquer número de 0 a 9, além de algumas letras do alfabeto. Tendo isto em mente, a utilização do sistema de numeração hexadecimal torna-se o mais indicado para a interface com o usuário, pois será necessário apenas um display para cada jogador.

Fazendo a correspondência entre os sistemas de numeração podemos montar a seguinte tabela de conversão.

Binário	Decimal	Hexadecimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Tabela de conversão Bin-Dec-Hex

2.2 Montagem da tabela verdade do decodificador

Para formar algum número no display é necessário permitir a passagem de energia em entradas específicas, enquanto bloqueia a passagem nas outras. Nomeando cada pino do display por uma letra temos a seguinte disposição:

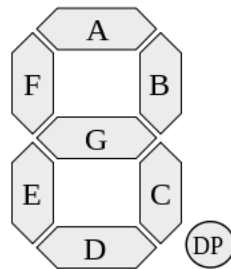


Figura 2: Pinos do display de 7 segmentos

Formando cada número temos:

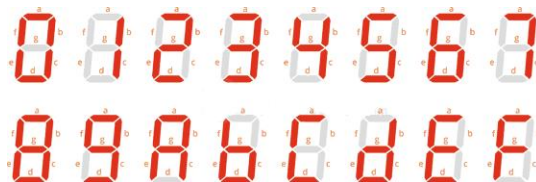


Figura 3: Números no display de 7 segmentos

A partir da formação dos números podemos montar a tabela verdade para todas as combinações de bits.

I_3	I_2	I_1	I_0		a	b	c	d	e	f	g
0	0	0	0		1	1	1	1	1	1	0
0	0	0	1		0	1	1	0	0	0	0
0	0	1	0		1	1	0	1	1	0	1
0	0	1	1		1	1	1	1	0	0	1
0	1	0	0		0	1	1	0	0	1	1
0	1	0	1		1	0	1	1	0	1	1
0	1	1	0		1	0	1	1	1	1	1
0	1	1	1		1	1	1	0	0	0	0
1	0	0	0		1	1	1	1	1	1	1
1	0	0	1		1	1	1	1	0	1	1
1	0	1	0		1	1	1	0	1	1	1
1	0	1	1		0	0	1	1	1	1	1
1	1	0	0		1	0	0	1	1	1	0
1	1	0	1		0	1	1	1	1	0	1
1	1	1	0		1	0	0	1	1	1	1
1	1	1	1		1	0	0	0	1	1	1

Tabela verdade do display

Com a tabela verdade é possível montar o circuito combinacional de cada pino do display. Para exemplificar, mostraremos a expressão lógica do pino a.

		$I_1 I_0$			
		00	01	11	10
$I_3 I_2$	00	1	0	1	1
	01	0	1	0	1
	11	1	1	1	0
	10	1	1	1	1

Figura 4: Mapa de Karnaugh para o pino a

Através do mapa de Karnaugh construímos a expressão

$$a = \bar{I}_3 I_1 + I_2 I_1 + I_3 \bar{I}_0 + \bar{I}_2 \bar{I}_0 + \bar{I}_3 I_2 I_0 + I_3 \bar{I}_2 \bar{I}_1$$

Análogo ao procedimento descrito acima faremos às demais entradas.

$$a = \bar{I}_3 I_1 + I_2 I_1 + I_3 \bar{I}_0 + \bar{I}_2 \bar{I}_0 + \bar{I}_3 I_2 I_0 + I_3 \bar{I}_2 \bar{I}_1$$

$$b = \bar{I}_3 \bar{I}_2 + \bar{I}_3 \bar{I}_1 \bar{I}_0 + \bar{I}_3 I_1 I_0 + I_3 \bar{I}_1 I_0 + I_3 \bar{I}_2 \bar{I}_1 + \bar{I}_2 I_1 \bar{I}_0$$

$$c = \bar{I}_1 I_0 + I_3 \oplus I_2 + \bar{I}_3 (\overline{I_1 \oplus I_0})$$

$$d = I_3 \bar{I}_1 \bar{I}_0 + I_2 (I_1 \oplus I_0) + \bar{I}_3 \bar{I}_2 (I_1 + \bar{I}_0) + I_3 \bar{I}_2 I_0$$

$$e = I_1 \bar{I}_0 + I_3 I_2 + \bar{I}_2 \bar{I}_1 \bar{I}_0 + I_3 I_1 I_0$$

$$f = \bar{I}_1 \bar{I}_0 + I_3 \bar{I}_2 + I_2 (\overline{I_3 \oplus I_1}) + I_2 I_1 \bar{I}_0$$

$$g = I_3 \bar{I}_2 + I_1 \bar{I}_0 + I_3 I_0 + \bar{I}_2 I_1 + \bar{I}_3 I_2 \bar{I}_1$$

Após montar o circuito a partir das expressões teremos o nosso decodificador.

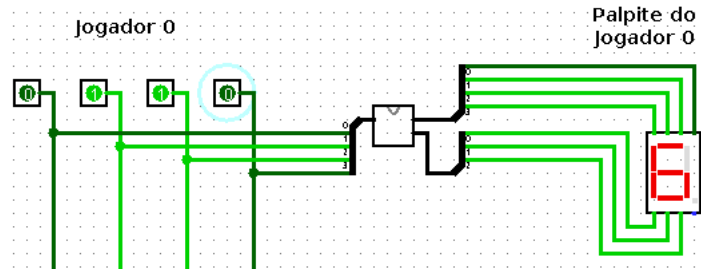


Figura 5: Decodificador em funcionamento

3 Circuito Multiplexador

3.1 Motivação para o uso do circuito MULTIPLEXADOR

O multiplexador desempenha o papel de encaminhar o chute efetuado pelo jogador atual entre Jogador 0 e Jogador 1. Para realizar essa tarefa, empregamos uma estrutura composta por portas lógicas AND e OR. Se o valor do pino de seleção for 0 a saída do multiplexador será igual a saída do pino A, caso contrário, será igual a do pino B.

A sua expressão lógica é: $Saida = \bar{S}A + SB$.

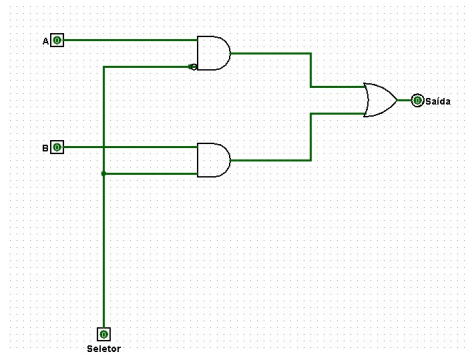


Figura 6: Multiplexador 1 bit

3.2 Agrupamento do MULTIPLEXADOR

A ideia é encapsular o multiplexador para que ele consiga atender a necessidade da entrada do jogo. Para fazer a seleção entre mais bits basta aninhar vários multiplexadores em paralelo.

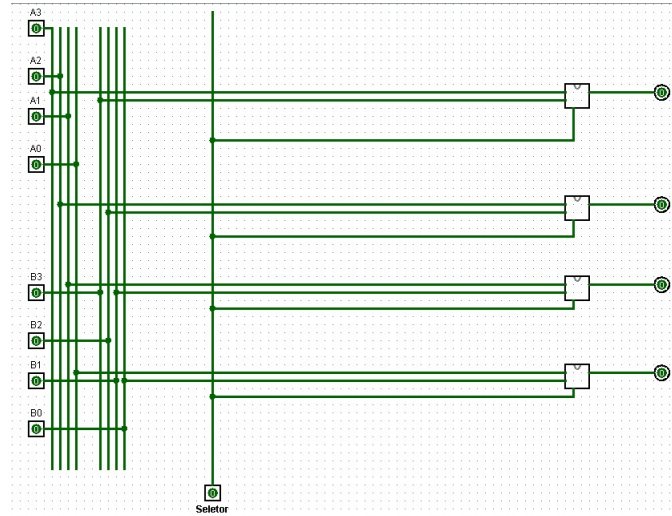


Figura 7: Multiplexador 4 bits

4 Circuito Comparador

O componente principal do jogo é o comparador de magnitude, que compara dois números binários.

O comparador aceita dois números binários de 4 bits como entrada e produz três resultados distintos: um para indicar se o valor é superior, outro para quando é inferior e um terceiro quando coincide com um "número mágico" estabelecido dentro do comparador. Introduzimos uma constante que representa esse valor especial. Nosso dispositivo comparador é desenvolvido empregando uma combinação de portas lógicas XNOR, AND, NOT, OR e NOR.

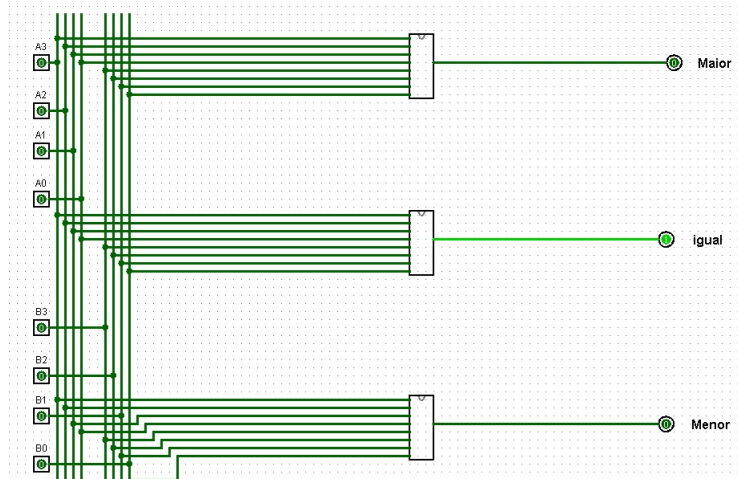


Figura 8: Comparador

4.1 Comparador Igual

Para representar a igualdade usamos XNOR, visto que sua saída é 1 apenas quando as duas entrada possuem o mesmo valor, e comparamos os dois números indo do bit mais significativo para o menos significativo.

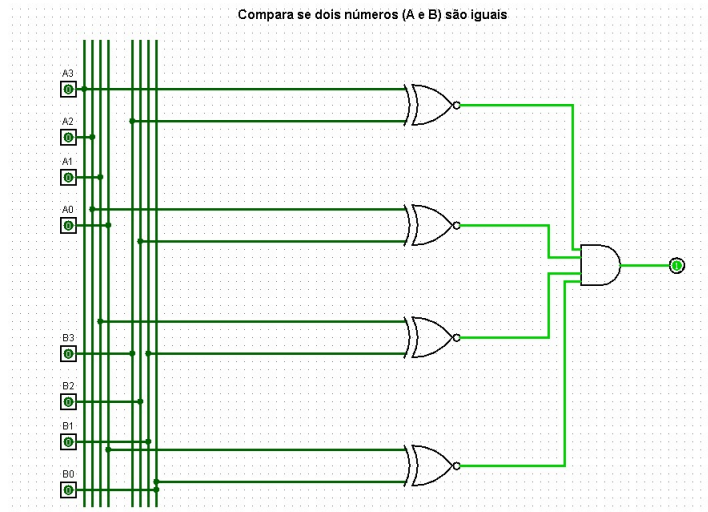


Figura 9: Comparador Igual

4.2 Comparador maior

Para comparar se o chute tem magnitude maior que o valor mágico, comparamos os dois números olhando do bit mais significativo para o menos significativo.

Se o bit atual de A é 1 e o de B é 0, e os bits mais significativos que o atual são iguais, então o número é maior.

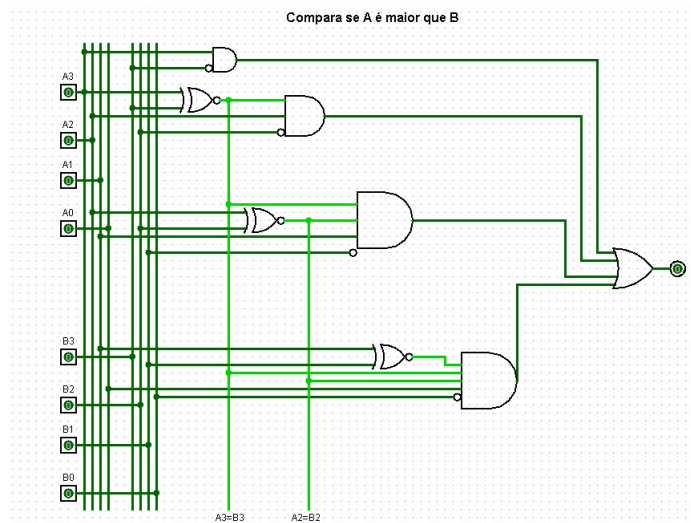


Figura 10: Comparador Maior

4.3 Comparador menor

utilizando das estruturas das seções 4.1 e 4.2 podemos construir o comparador menor, visto que, pela tricotomia dos números reais, um número sempre será maior ou igual, caso contrário ele será menor. Desta forma temos:

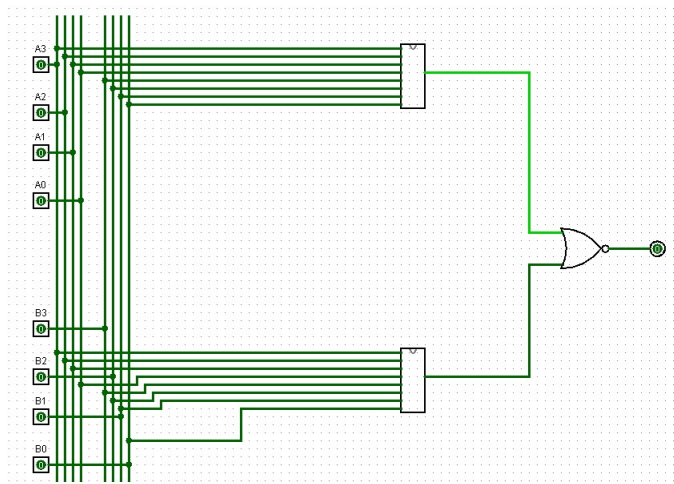


Figura 11: Comparador Menor

5 Game

O circuito final do jogo é formado por um circuito multiplexador de 4 bits e um comparador para informar se o palpite do jogador é maior, menor ou igual ao número mágico definido. Além disso, há um botão power, para informar a comparação do número apenas quando este for apertado.

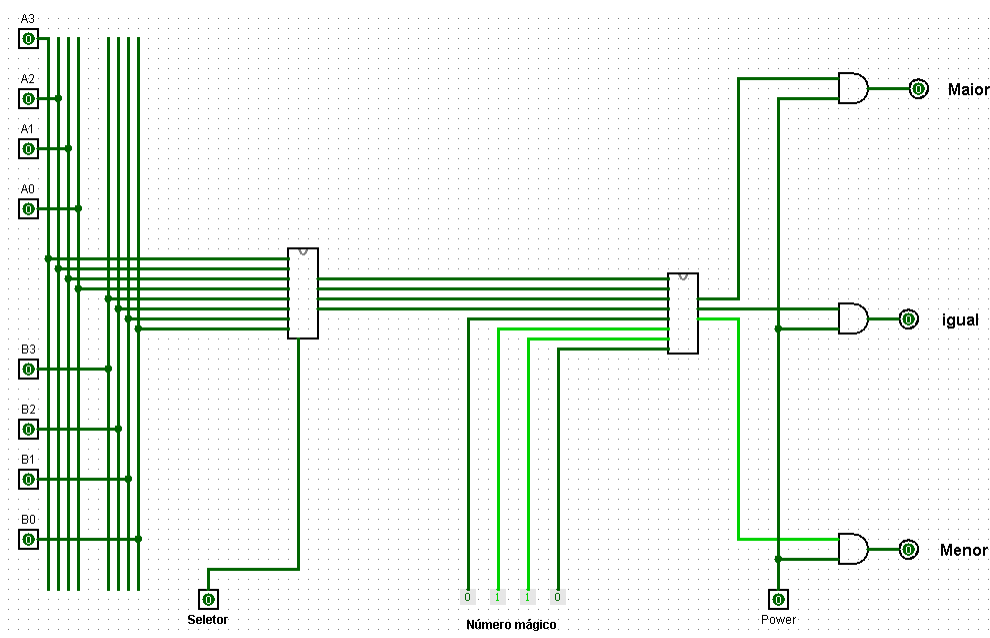


Figura 12: Game