



RELATÓRIO PROJETO

“Verificação integridade com código de Hamming”

Grupo EMEKADE

Matheus Percário Bruder RA: 222327

Daniel Mitio Mori RA: 214755

Kevin Barrios RA: 219643

O projeto apresentado na Faculdade de Tecnologia de Limeira, da disciplina de organização e arquitetura de computadores, fará parte do processo de avaliação.



INTRODUÇÃO

Nosso projeto tem como objetivo verificar se uma sequência de 15 bits está corrompida ou não, através do código de Hamming. Para isso, será utilizado o *software Logisim*, o qual permite a simulação de circuitos combinacionais.

Se a sequência estiver corrompida um LED será aceso, caso contrário continuará desligado.

CIRCUITO COMBINACIONAL COMPLETO

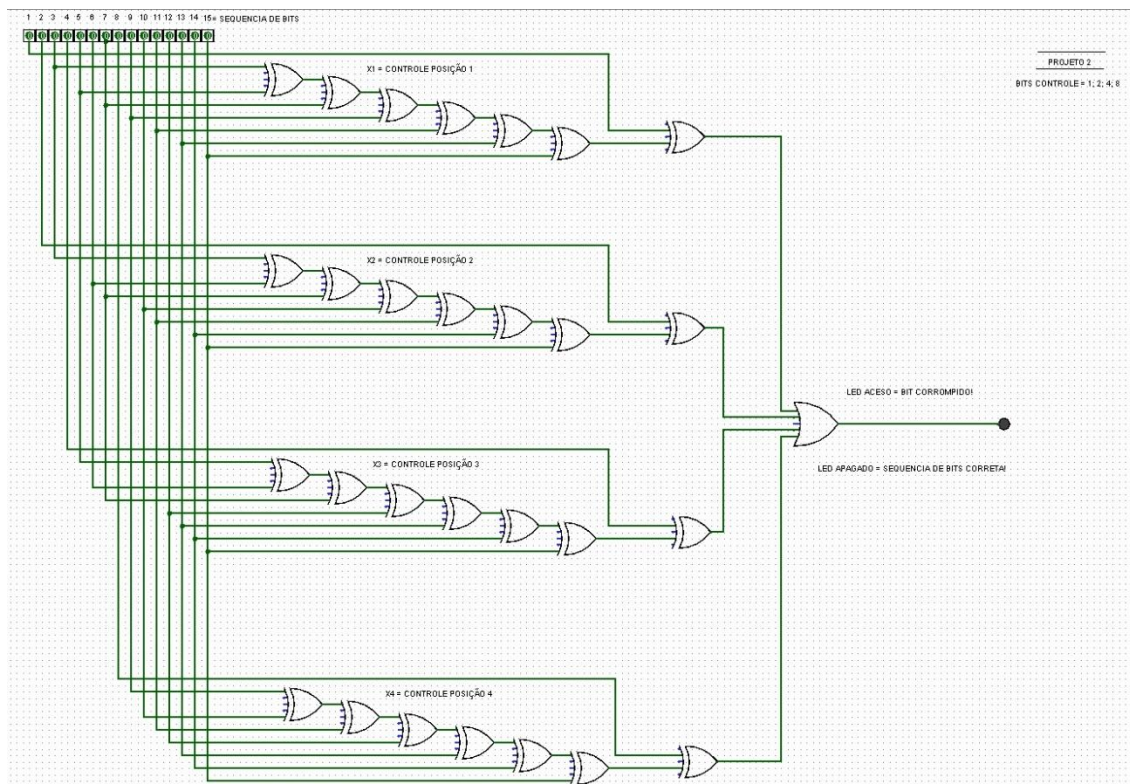


Foto do circuito combinacional implementado pelo software Logisim.



FUNÇÕES LÓGICAS

Para realizar a verificação dos bits, criamos 4 cadeias de XOR, em que cada uma representa um bit de controle. Então, o circuito funcionará da seguinte forma:

1. Usuário digita sequência de 15 bits (incluindo os bits de controle)
2. O circuito combinacional calcula os bits de controle
3. Há comparação entre os bits de controle fornecidos pelo usuário com os bits que o software calculou.

➤ **Legenda:**

- a. Bit de controle: X1, X2, X3, X4
- b. Outros Bits: M1, M2, M3, M4, M5, M6, M7, M8, M9, M10, M11

FUNÇÕES LÓGICAS – Cálculo dos bits de controle realizado pelo software:

- 1º BIT CONTROLE: $X1 = M1 \oplus M2 \oplus M4 \oplus M5 \oplus M7 \oplus M9 \oplus M11$
- 2º BIT CONTROLE: $X2 = M1 \oplus M3 \oplus M4 \oplus M6 \oplus M7 \oplus M10 \oplus M11$
- 3º BIT CONTROLE: $X3 = M2 \oplus M3 \oplus M4 \oplus M6 \oplus M9 \oplus M10 \oplus M11$
- 4º BIT CONTROLE: $X4 = M5 \oplus M6 \oplus M7 \oplus M8 \oplus M9 \oplus M10 \oplus M11$

Após calcular os bits de controle é preciso compara-los com os bits fornecidos pelo usuário e, para isso, foram utilizadas outras portas XOR. Pois, elas possibilitam que o resultado seja zero para entradas iguais. Assim, como na tabela, quando a entrada A e B forem iguais, o resultado da porta XOR será zero e quando as entradas forem diferentes, o resultado será 1.



Tabela OU Exclusivo (\oplus)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

FUNÇÕES LÓGICAS – Comparação entre bits de controle fornecidos pelo usuário

- 1ª COMPARAÇÃO: $X1 \oplus \text{BIT } 1$
- 2ª COMPARAÇÃO: $X2 \oplus \text{BIT } 2$
- 3ª COMPARAÇÃO: $X3 \oplus \text{BIT } 4$
- 4ª COMPARAÇÃO: $X4 \oplus \text{BIT } 8$

Finalmente, após realizar as quatro comparações, é preciso junta-las com a porta logica OR. Porque nela independente da entrada o 1 é dominante, logo, se algum bit estiver corrompido, a entrada será 1 e consequentemente a porta OR também terá saída 1. Dessa maneira, acenderá o LED, possibilitando saber se a sequência está corrompida ou não.

Tabela OU (V)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1



TESTES

Exemplo 1 – Sequencia: 111110111101011

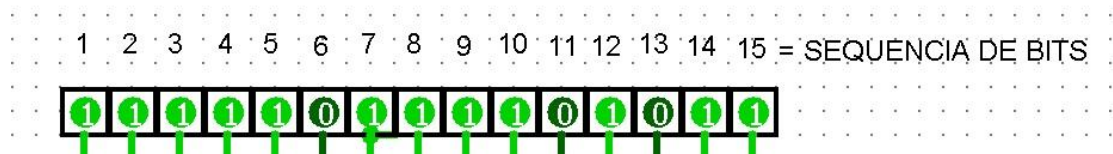


Figura 1

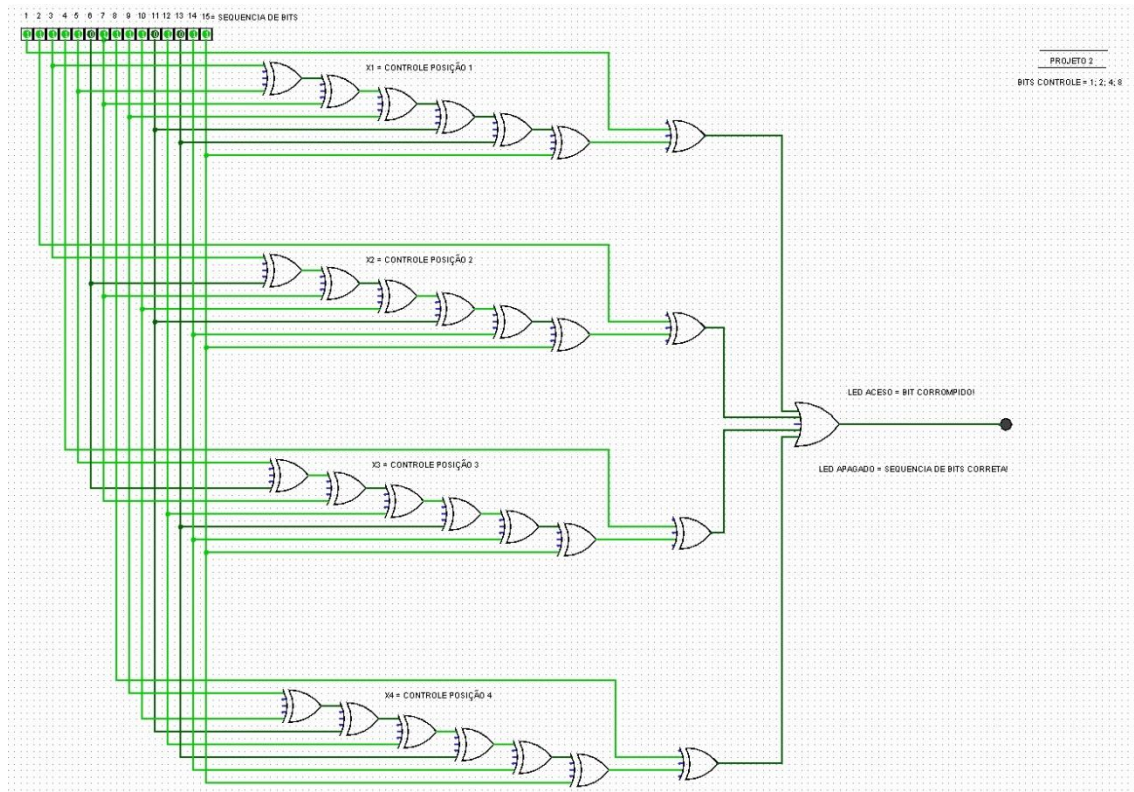


Figura 2

Figura 1 – Sequência de bits a ser analisada

Figura 2 – Resultado: Nenhum bit corrompido, LED apagado



Exemplo 2 – Sequencia: 001110111001010

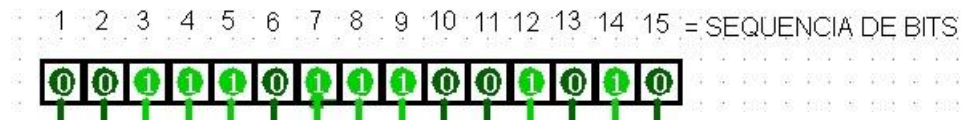


Figura 3

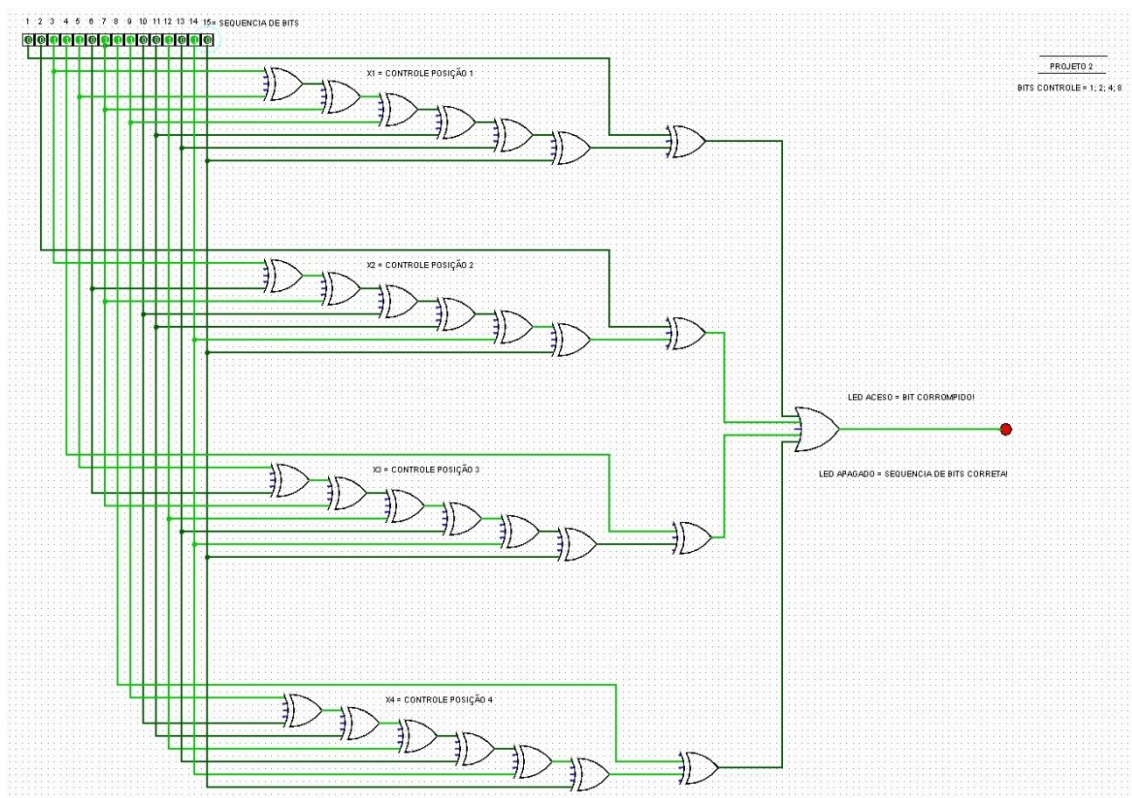


Figura 4

Figura 1 – Sequência de bits a ser analisada

Figura 2 – Resultado: Algum bit corrompido, LED aceso



Exemplo 3 – Sequencia: 110100010111101

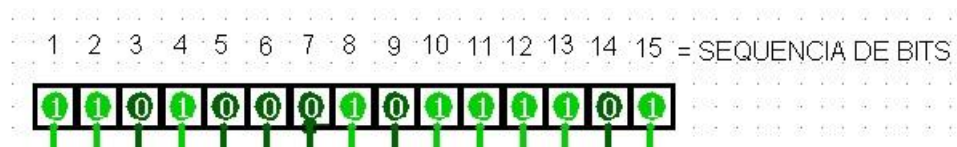


Figura 5

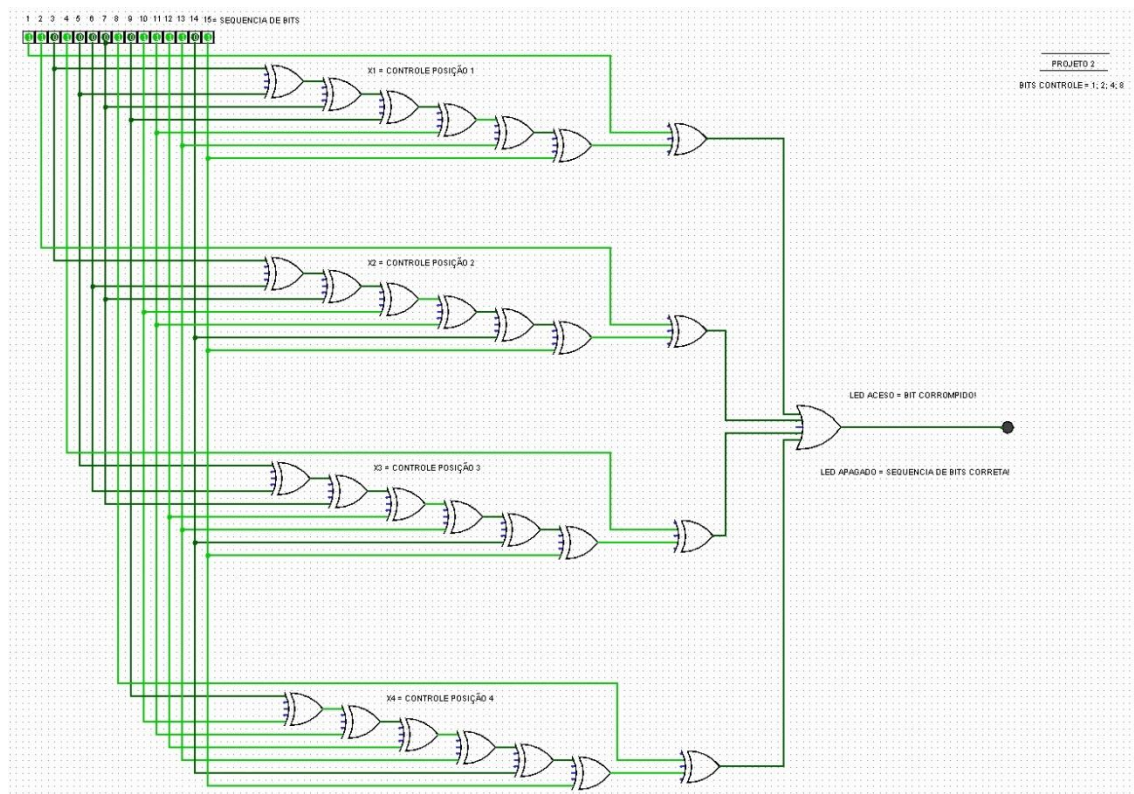


Figura 6

Figura 5 – Sequência de bits a ser analisada

Figura 6 - Resultado: Nenhum bit corrompido, LED apagado



Exemplo 4 – Sequencia: 110001101011000

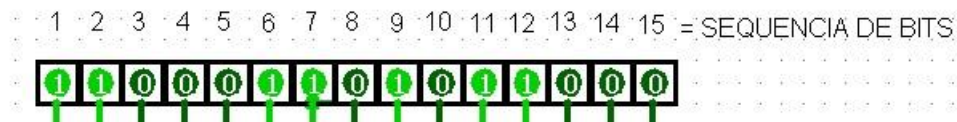


Figura 7

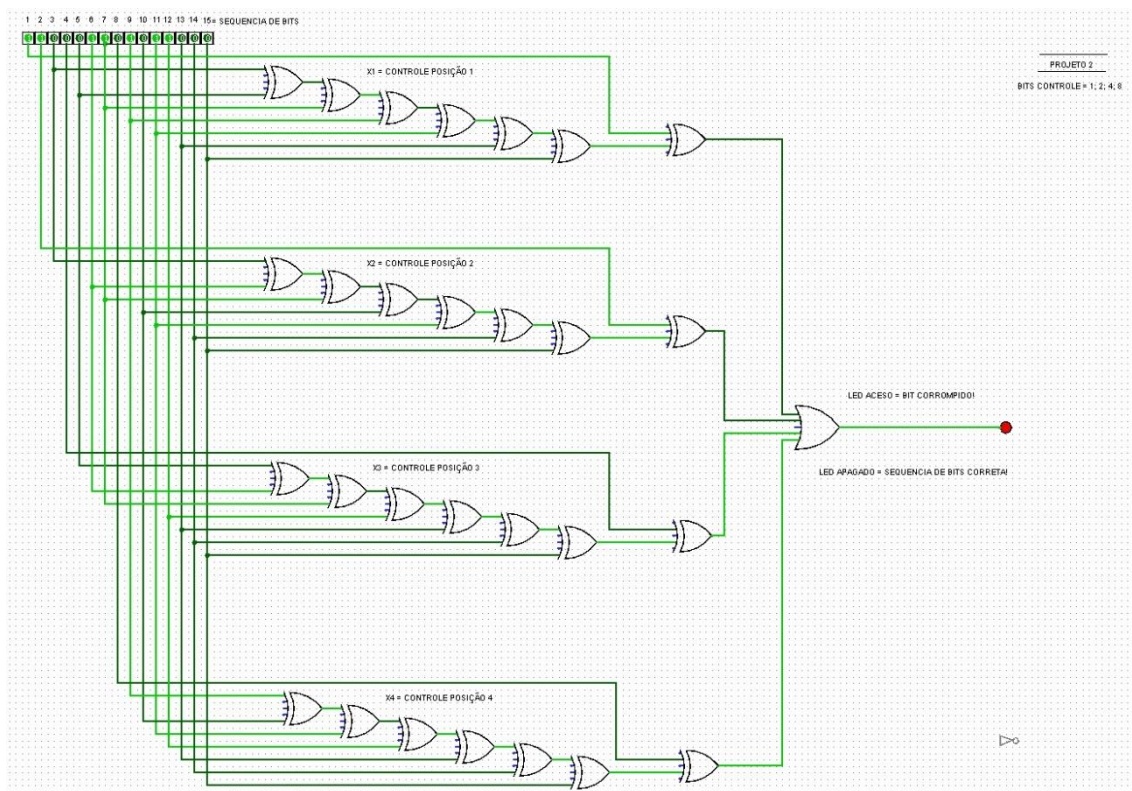


Figura 8

Figura 7 – Sequência de bits a ser analisada

Figura 8 – Resultado: Algum bit corrompido, LED aceso