UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

BEATRIZ MATIAS SANTANA MAIA
LUANA GABRIELE DE SOUSA COSTA
SOPHIE DILHON GAMA

RELATÓRIO DO SEGUNDO TRABALHO:

Construção de um verificador de CRC de 8 bits

VITÓRIA
2019

BEATRIZ MATIAS SANTANA MAIA
LUANA GABRIELE DE SOUSA COSTA
SOPHIE DILHON GAMA

RELATÓRIO DO SEGUNDO TRABALHO:

Construção de um verificador de CRC de 8 bits

Relatório apresentado ao curso de Ciência da Computação da Universidade Federal do Espírito Santo como parte dos requisitos necessários à aprovação parcial na disciplina Elementos de Lógica Digital.

Orientador: Jadir Eduardo Sousa Lucas M.Sc.

VITÓRIA
2019

SUMÁRIO

1 INTRODUÇÃO

O verificador de CRC consiste em um circuito baseado na divisão de dois polinômios, sendo um a mensagem a ser transmitida e o outro o polinômio gerador de grau N, que obtém como resto o valor de CRC de tamanho N bits, sendo a divisão operações XOR bit por bit, tendo o primeiro bit sempre de valor 1.
O objetivo do circuito é verificar se o valor do CRC é ou não diferente de zero, no caso de ser diferente, é relatado que houve um erro na transmissão da mensagem.

2 DESENVOLVIMENTO

2.1 DIVISÃO DAS ETAPAS

Antes de elaborar o circuito do CRC de 8 bits, precisou-se criar um gerador de mensagem (que já possui o CRC nos seus últimos bits) para poder simular o funcionamento do circuito, além disso, para o compreender melhor, houve a necessidade de analisar o circuito de um CRC mais simples, tendo uma quantidade menor de bits. Projetou-se um CRC de 4 bits, com a mensagem de tamanho 15 bits, para facilitar a visualização da divisão polinomial.
Após a simulação do CRC de 4 bits e da criação do gerador de mensagens e das mensagens, projetou-se o CRC de 8 bits.
Todos os projetos foram desenvolvidos na plataforma *Logisim Evolution*, nas versões 2.14 (versão dos laboratórios da UFES) e 2.15 (versão dos computadores particulares).

2.2 GERADOR DE MENSAGEM

2.2.1 Componentes utilizados

Os componentes utilizados no Logisim para a construção do Gerador de Mensagens foram:

- portas lógicas;

- contador de 3 bits;
- *Random Generator*;
- RAM 512 x 1;
- clock.

A escolha da RAM 512 x 1 foi baseada no tamanho máximo da mensagem de 500 bits mais a quantia de bits do tamanho do CRC, necessitando de uma memória com no mínimo 508 endereços.

Inicialmente, quando o objetivo era criar um CRC de 32 bits em vez de 8 bits, escolheu-se uma RAM 1k x 1 devido ao fato da mensagem possuir 500 bits e o CRC ser de 32 bits. Uma RAM de 512 x 1 não seria suficiente para tal projeto.

2.2.2 Condição de parada

Para gerar uma mensagem randômica de 508 bits, necessitou-se implementar um contador até 508. Para criar a condição de parada do contador, utilizou-se uma porta E de 9 entradas, que recebeu como entrada o número gerado pelo contador, e uma saída negada de forma que ao sinalizar *1*, representasse um número diferente de 508.

Para gerar a condição de parada, precisou-se converter o número de base decimal para base binária.

$$508_{10} = 111111100_{1} \tag{1}$$

Utilizou-se a *Tabela 1* para a descobrir a função lógica para a condição de parada.

TABELA 1 - TRECHO DA TABELA VERDADE DA CONDIÇÃO DE PARADA

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	S
1	1	1	1	1	1	1	0	0	0

A função lógica desenvolvida a partir da soma de produtos da *Tabela 1* se encontra na equação (2) na próxima página:

$$S = b_8 \cdot b_7 \cdot b_6 \cdot b_5 \cdot b_4 \cdot b_3 \cdot b_2 \cdot (\overline{b_1}) \cdot (\overline{b_0}) \quad (2)$$

A *Figura 1* indica o circuito desenvolvido a partir da função lógica representada pela equação (2), e a *Figura 2* demonstra as entradas da porta E, com as entradas 1 e 2, que representam respectivamente b_0 e b_1 da equação (2), negadas. O circuito foi denominado *n508* (não 508).

FIGURA 1 - Circuito identificador de um número não 508

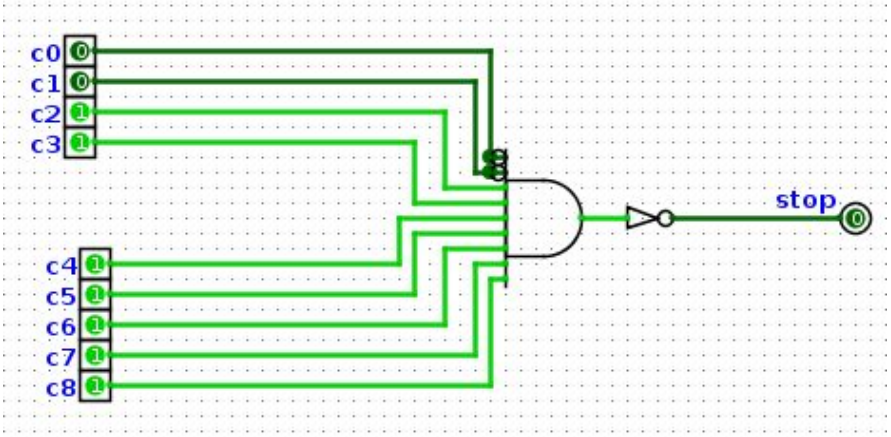


FIGURA 2 - Configuração da porta E utilizada para verificação do número 508

Seleção: Porta AND	
VHDL	Verilog
Posição	Leste
Bits de dados	1
Tamanho da porta	Médio
Quantidade de entradas	9
Valor de saída	0/1
Rótulo	
Fonte do rótulo	SansSerif Negrito 16
Negar 1 (Em cima)	Sim
Negar 2	Sim
Negar 3	Não
Negar 4	Não
Negar 5	Não
Negar 6	Não
Negar 7	Não
Negar 8	Não
Negar 9 (Embaixo)	Não

Fonte: Logisim Evolution 2.14

2.2.3 Criação do Gerador de Mensagens

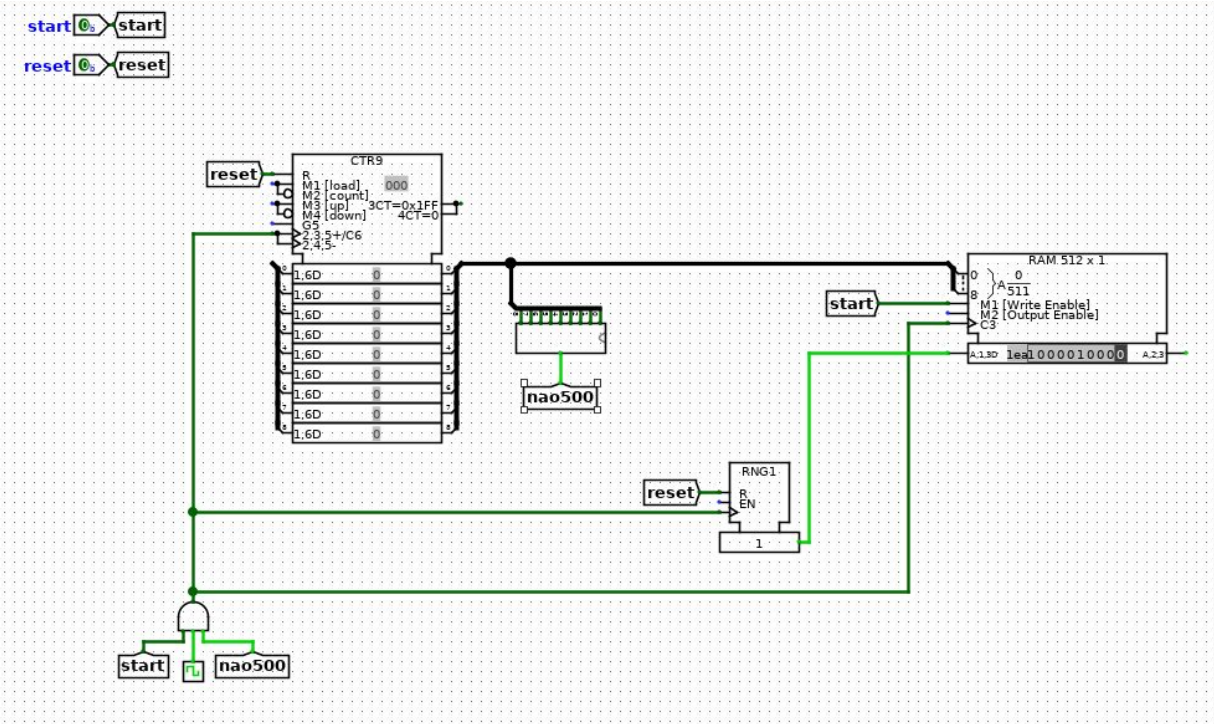
Inicialmente, usou-se um contador de 9 bits com suas saídas ligada ao circuito *n508.* Para que o *clock* parasse de pulsar em 508, terminando a geração da mensagem, posicionou-se uma porta E à entrada do *clock* do contador. A porta E recebeu como entradas a saída do *n508*, um clock do *Logisim Evolution* e uma condição de inicialização do circuito (start).

Para gerar a mensagem, necessitou-se utilizar o *Random Generator* de 1 bit do *Logisim Evolution* para gerar um número aleatório de 1 bit e uma memória RAM 512 x 1 para armazenar o resultado da mensagem. Para acessar cada endereço da memória RAM, foi utilizado o resultado do contador de 9 bits. O dado escrito na memória foi gerado a partir do *Random Generator*.

Para reiniciar o circuito e gerar outra mensagem, é necessário pressionar o botão *reset*.

A *Figura 3* indica o circuito Gerador de Mensagem .

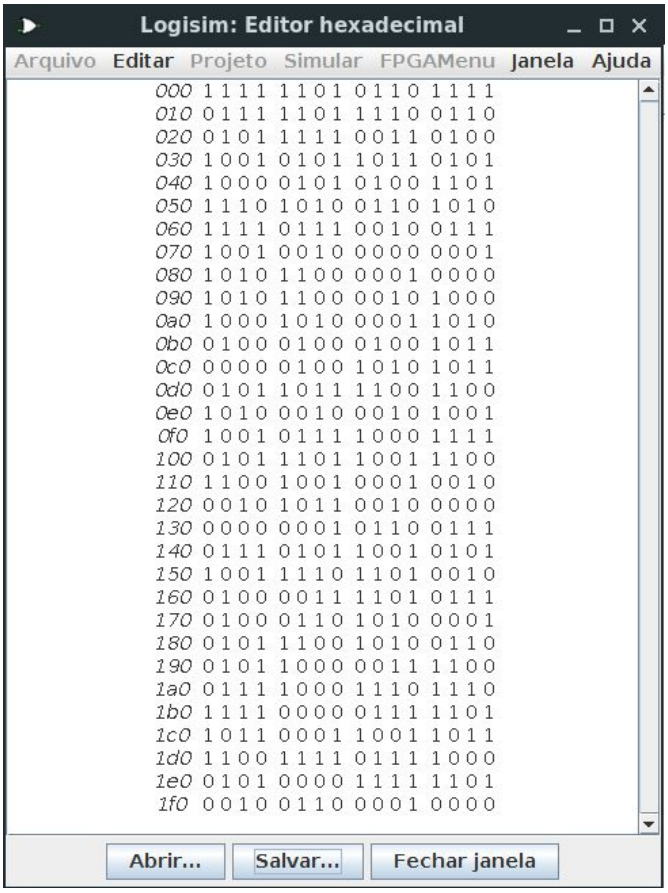
FIGURA 3 - Gerador de Mensagens



2.2.4 Funcionamento

Ao inicializar o clock e pressionar o botão *start*, o clock irá pulsar 508 vezes (indicado pelo contador). A mensagem gerada é armazenada na RAM do *Logisim Evolution*, indicada pela *Figura 4*.

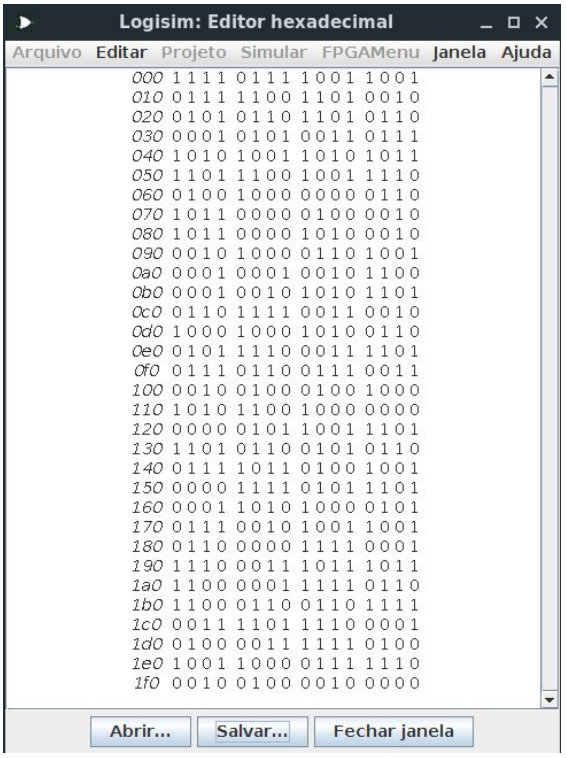
FIGURA 4 - Mensagem *mensagem1* gravada na RAM



Fonte: Logisim Evolution 2.14

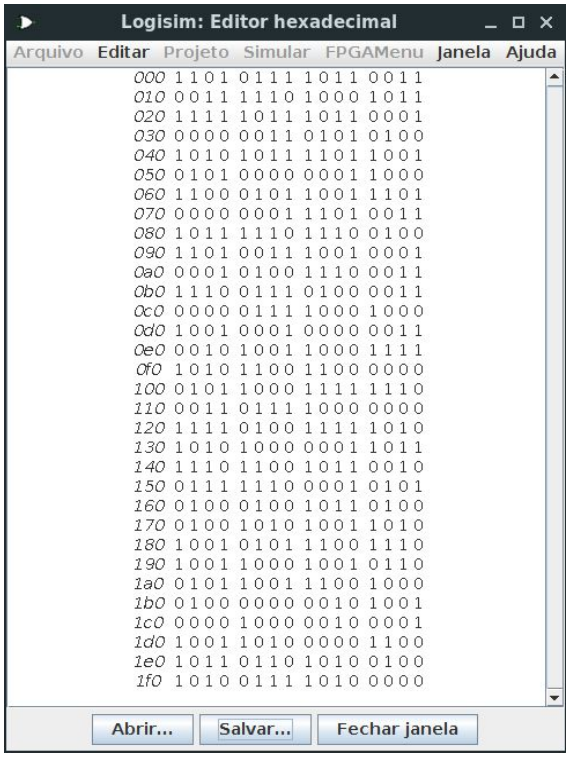
Para utilizar as mensagens geradas posteriormente, necessitou-se armazená-las no computador após o contador chegar em 508. Foram criadas 3 arquivos de mensagens.

FIGURA 5 - Mensagem *mensagem2* gravada na RAM



Fonte: Logisim Evolution 2.14

FIGURA 6 - Mensagem *mensagem3* gravada na RAM



Fonte: Logisim Evolution 2.14

2.3 CRC-4

2.3.1 Componentes utilizados

Os componentes utilizados no Logisim para a construção do CRC-4 foram:

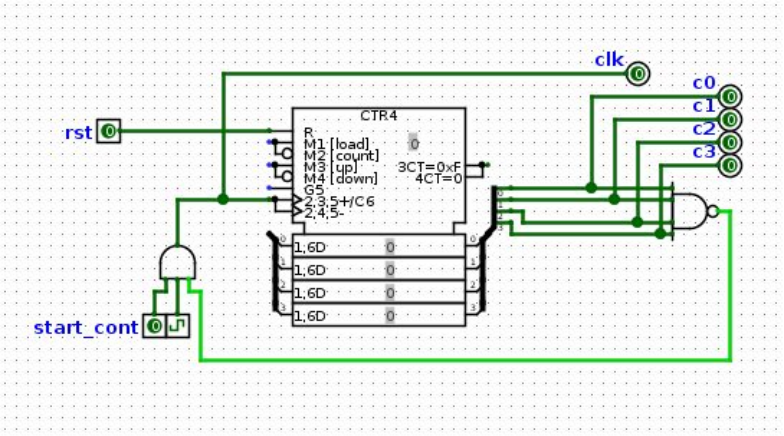
- portas lógicas;
- contador de 4 bits;
- flip-flop tipo D;
- ROM 16 x 1;
- clock.

A memória ROM 16 x 1 foi escolhida com o intuito de representar mensagens de até 15 bits, de forma que facilitasse os cálculos e visualização do funcionamento do CRC de 4 bits.

2.3.2 Condição de parada

A condição de parada do CRC-4 depende do tamanho da mensagem a ser lida (15 bits). Portanto, foi utilizado um contador de 4 bits, que possui como entrada ao seu clock as suas saídas, uma condição inicial (o *start_cont*) e um dispositivo clock. Isso ocorre devido ao uso duma porta E que recebe-se como entrada as saídas do contador e possui como saída o resultado da operação E negada.

Figura 7 - Contador de 3 bits



2.3.3 Criação do CRC-4

2.3.3.1 Componente de operação

A divisão polinomial de grau 4 será representado de forma:

$$b_4 \times x^4 + b_3 \times x^3 + b_2 \times x^2 + b_1 \times x^1 + b_0 \times x^0 \quad (3)$$

No qual o termo a ser utilizado na divisão será representado de forma binária $(b_4 b_3 b_2 b_1 b_0)_2$. Para executar a divisão, precisa-se de um circuito capaz de fazer paralelamente a operação XOR com o número $(b_4 b_3 b_2 b_1 b_0)_2$, que possui 5 bits, portanto necessita-se de 5 componentes que recebem como entrada 1 bit do polinômio gerado. Há uma condição especial na divisão, ela não deverá ser feita se o bit mais significativo da mensagem for igual a 0. Quando isso ocorrer, é necessário que a mensagem seja deslocada até que b_4 seja igual a 1.

Foi criado uma tabela verdade (*Tabela 2*) para analisar como deverá ser construído o circuito. O termo mais significativo foi denominado de c_4, ele representa tanto o resultado da operação XOR entre o quarto bit do polinômio e o quarto bit do início da mensagem ou de seu resto, quanto o valor do bit mais significativo. Caso o último termo seja igual a 0, independente do valor do polinômio, a saída c_{n+1}, onde n < 5, deverá ser igual a c_n, indicando a necessidade de um deslocador. Caso o último termo seja igual a 1, a sáida c_{n+1} deverá ser o resultado da operação XOR entre o bit do polinômio e c_n.

Tabela 2 - Tabela Verdade com a e b possuindo valores binário de 1 bit e $n < 5$.

c_4	c_n	Termo do polinômio (P)	c_{n+1}
0	a	b	a
1	a	b	$a \oplus b$

A equação (4) indica a equação de excitação gerada a partir da tabela acima, e a *Figura 8* demonstra a construção do circuito. A necessidade de utilizar um flip-flop tipo D é devido a necessidade de deslocar o bit para efetuar a leitura de todos os bits da mensagem e manter o circuito síncrono.

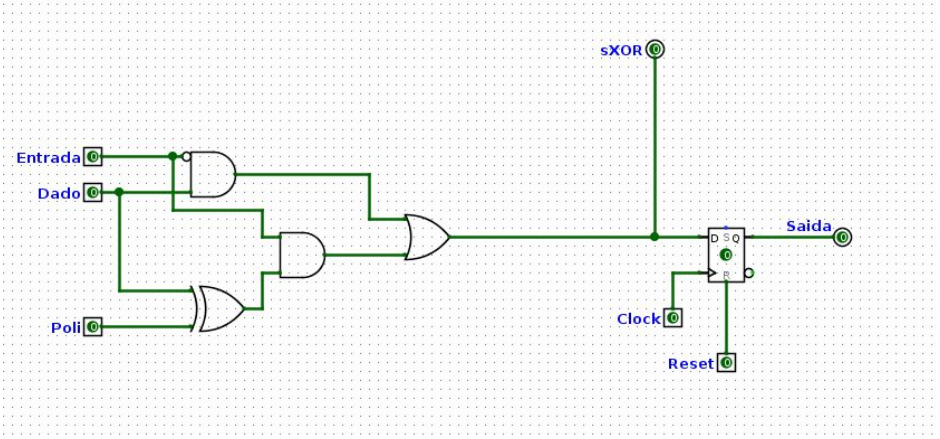
Tabela 3 - Tabela Verdade Estendida. Obs.: Os termos 100 e 101 não existem quando n = 4.

C4	Cn	Pn	Cn1
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Fonte: *Logisim Evolution 2.14*

$$\overline{c_4} \cdot c_n \; + \; c_4 \cdot (P \oplus c_n) \tag{4}$$

Figura 8 - Circuito do componente de operação necessário para a construção do CRC

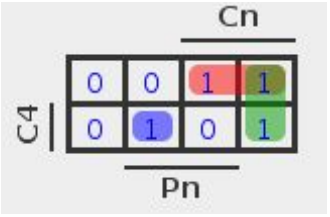


Para visualizar o comportamento do circuito, independente do clock, foi criado uma saída sXOR, como retratada abaixo na *Figura 10*, porém ela não é utilizada no circuito CRC-4.

Figura 9 - Bloco do componente de operação

Na utilização do Mapa de Karnaugh (*Figura 10)*, é possível encontrar uma equação, indicada pela equação 5, simplificada utilizando apenas portas E e OU.

Figura 10 - Mapa de Karnaugh do circuito



Fonte: *Logisim Evolution 2.14*

$$\overline{c_4} \cdot c_n + c_n \cdot \overline{P} + c_4 \cdot \overline{c_n} \cdot P \tag{5}$$

Para a construção do circuito, foi utilizado a equação (4) devido ao uso significativo menor de portas e desconsiderando a possibilidade de atraso da porta XOR (simulação de um circuito com condições ideais).

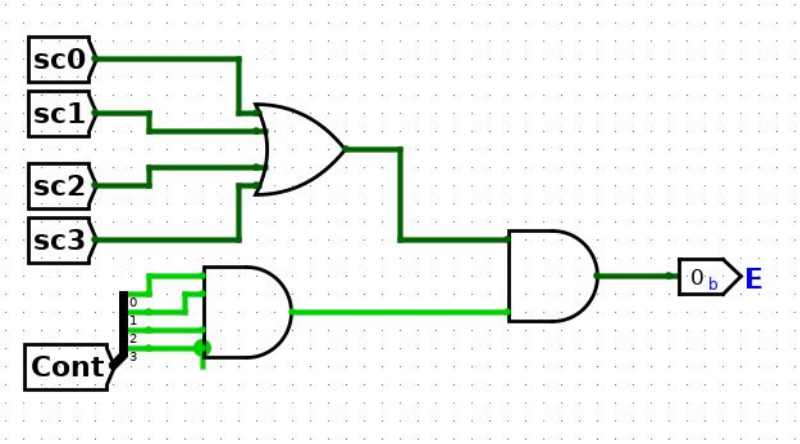
2.3.3.2 Indicador de erro

Ao terminar os processos anteriores, têm-se como resultado a ser analisado os 4 bits menos significativos, o CRC.

O circuito é composto por uma porta OU de n entradas, nesse caso, quatro entradas sendo elas os bits do CRC, e uma porta E com duas entradas, sendo elas a condição de parada, e o resultado do OU acima.

No caso de todos os bits do CRC serem iguais a zero, e a condição de parada for igual a 1, o resultado será zero, portanto não houve erro, porém, caso haja algum bit de valor 1 no CRC, mostra-se que houve erro na transmissão da mensagem, com a saída tendo valor igual a 1. Segue abaixo uma imagem do circuito.

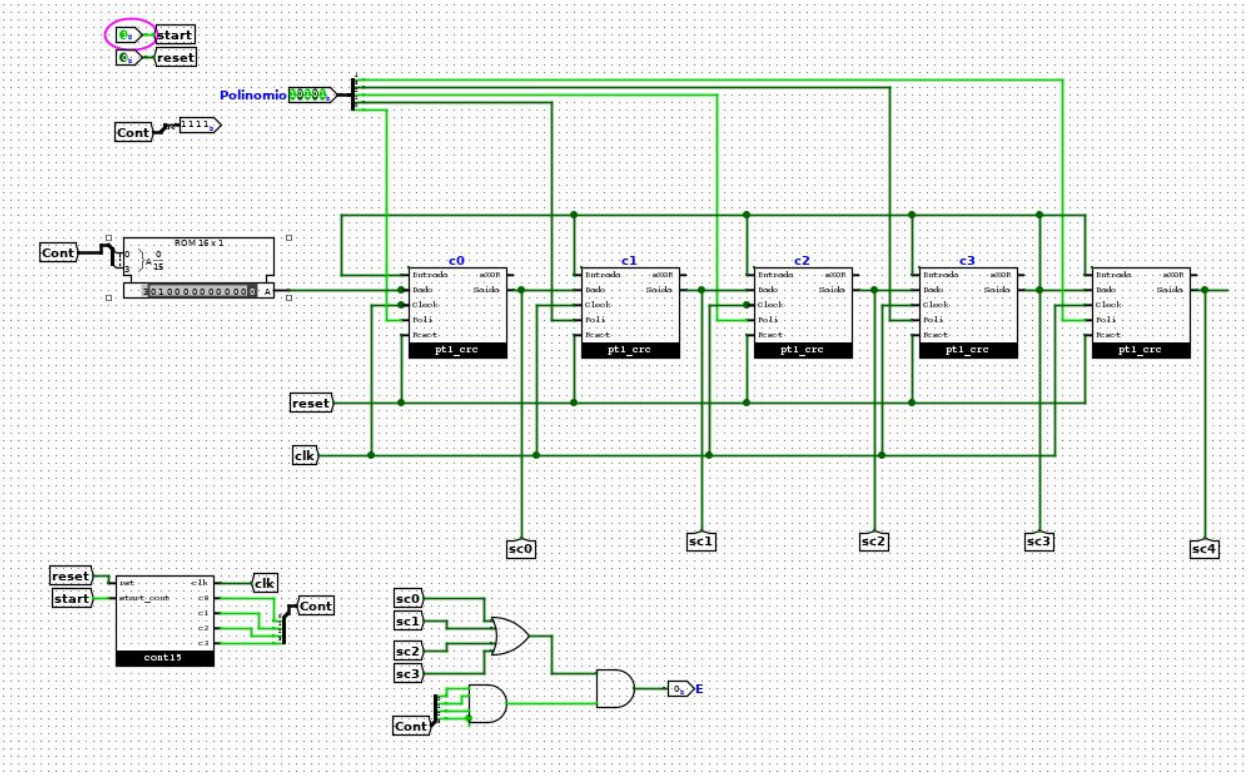
Figura 11 - Indicador de erro



2.3.3.3 Funcionamento do circuito

O polinômio escolhido para o teste foi 10101_2. As mensagens foram escritas manualmente na memória ROM. O circuito foi criado com 5 componentes de operação simbolizados por c_0, c_1, c_2, c_3 e c_4. c_n irá executar um XOR com o bit b_n do polinômio caso c_4 sinalizar 1. A *Figura 12* representa o circuito CRC-4.

Figura 12 - CRC-4



2.4 CRC-8

2.4.1 Componentes utilizados

Os componentes utilizados no Logisim para a construção do CRC-8 foram:

- portas lógicas;
- contador de 9 bits;
- flip-flop tipo D;
- ROM 512 x 1;
- clock.

2.4.2 Criação do CRC-8

2.4.2.1 Condição de parada

A condição de parada do CRC-8 depende do tamanho da mensagem a ser lida (508 bits). Portanto, foi utilizado um contador de 9 bits (*Figura 14)*, que possui como entrada ao seu clock as suas saídas, uma condição inicial (o *start*) e um dispositivo clock. Similar a condição de parada vista no ítem *2.2.2*, o contador para ao chegar em 508_{10}. A *Figura 13* mostra a configuração da porta E que possui como entrada as saídas do contador. Quando o contador indicar um número que não seja 508_{10}, a sua saída será 1. Devido a isso, o circuito se chama *n508* (nao 508).

Figura 13 - Circuito identificador de um número não 508

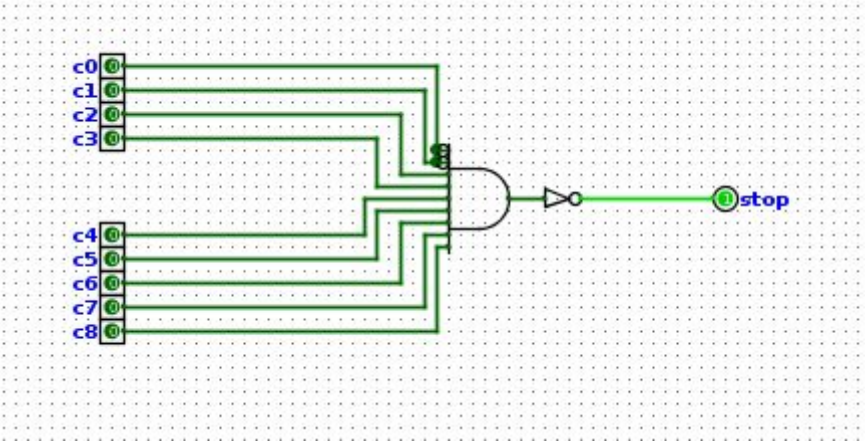
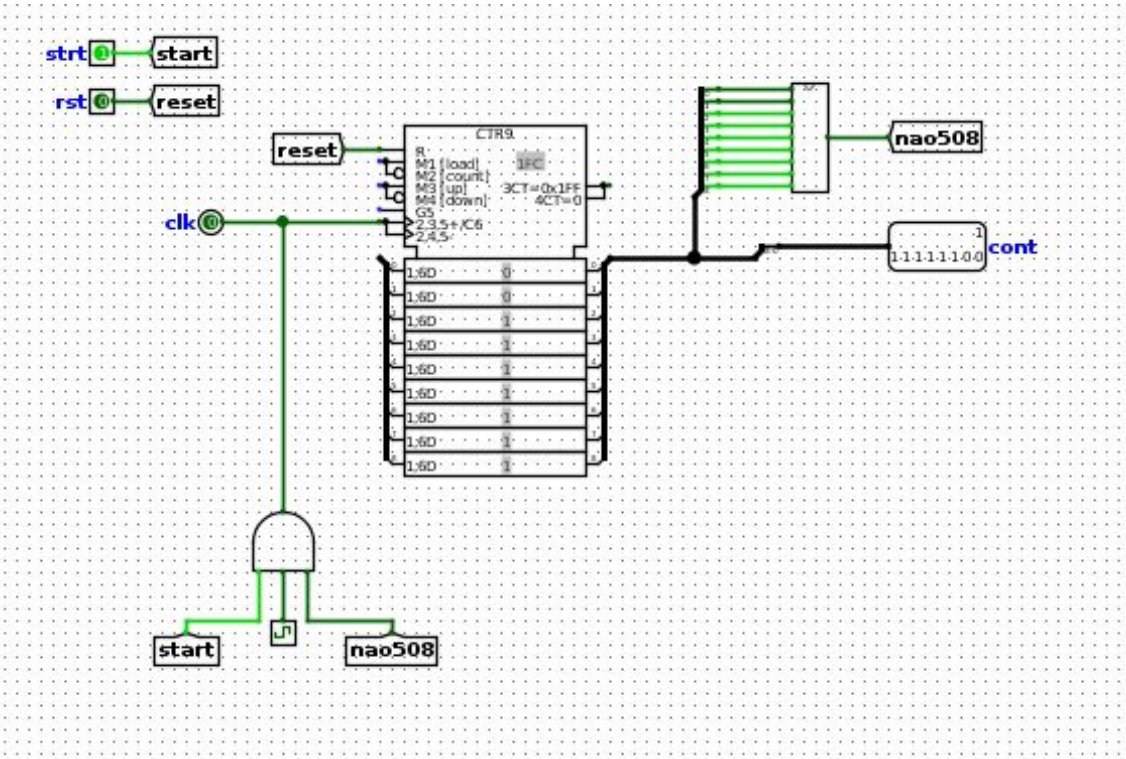


Figura 14 - Contador até 508



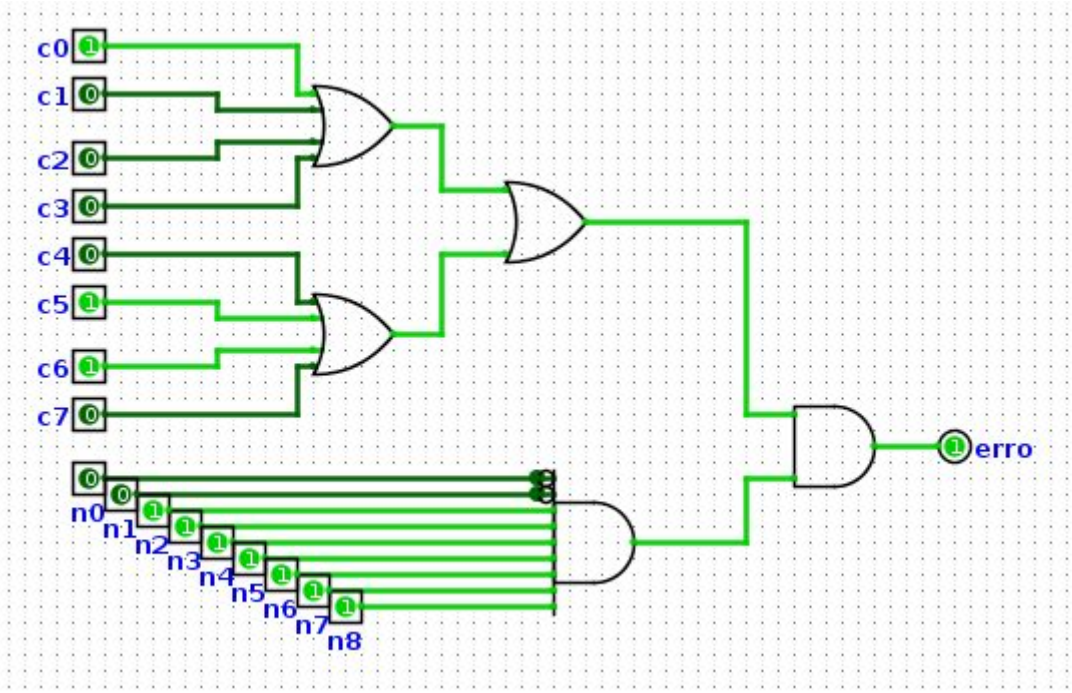
2.4.2.2 Componente de operação

O componente de operação utilizado foi o mesmo criado no ítem *2.3.3.1.*

2.4.2.3 Indicador de erro

Para indicar erro, basta que a divisão apresente resto diferente de 0, ou seja, que as saídas do circuito sejam diferentes de 0. Também deverá apresentar caso houve erro ou não apenas após a verificação (após contar 508). Como o resto basta ter ao menos 1 bit diferente de 0 para indicar erro, as saídas do circuito foram colocadas como entrada a uma porta OU. O erro ocorrerá caso o resultado da porta OU seja 1 e caso o circuito tenha contado 508. A *Figura 15* demonstra o circuito.

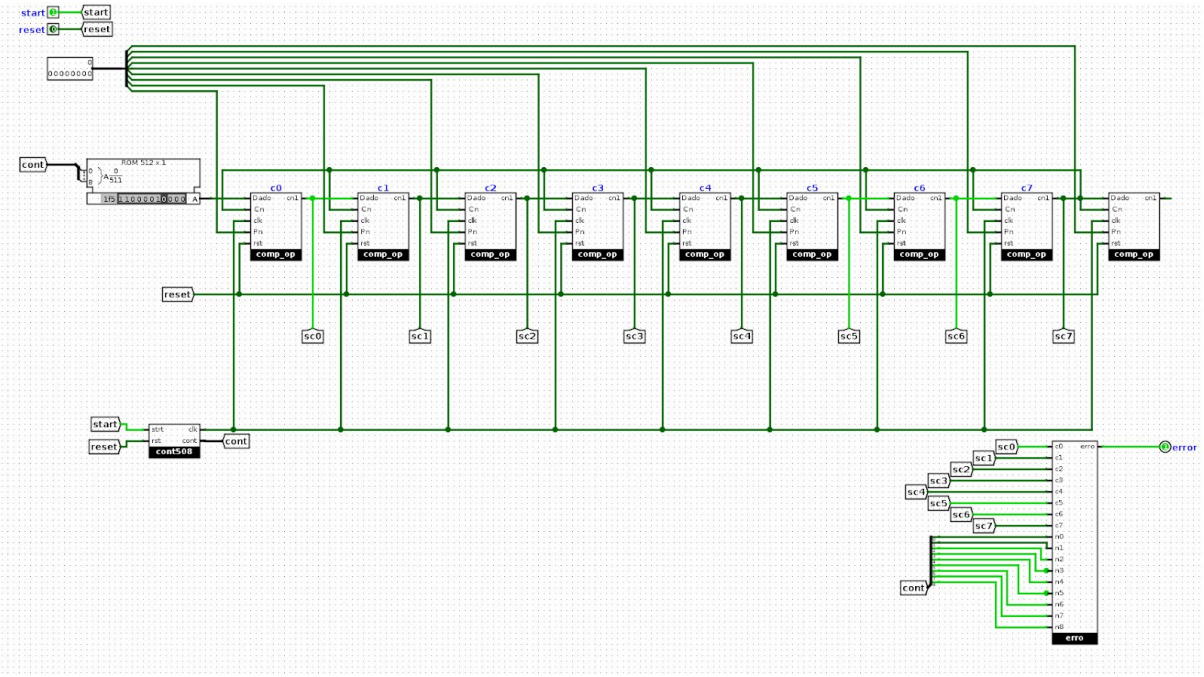
Figura 15 - Circuito Erro



2.4.3 Funcionamento do circuito

Similar ao funcionamento do circuito CRC-4, foi escolhido um polinômio e utilizados 9 componentes de operação. O polinômio escolhido foi 111010101_2 e os componentes de operação são simbolizados por c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7 e c_8. As mensagens geradas no gerador de mensagem foram colocadas na memória RAM 512x1. A *Figura 16* representa o circuito CRC-8. As saídas do circuito são utilizadas como entrada para o circuito indicador de erro.

Figura 16 - CRC-8



3 CONCLUSÃO

3.1 RESULTADOS OBTIDOS

Para as mensagens *mensagem1*, *mensagem2*, *mensagem3* o circuito indicou erro. Isso ocorre devido ao fato das mensagens não terem o CRC apropriado acrescentado ao final, e após as operações de divisão serem realizadas, o resultado obtido não ser de todos bits com valor 0.

REFERÊNCIAS BIBLIOGRÁFICAS

GEEKFORGEEKS. **Cyclic Redundancy Check and Modulo-2 Division**. Disponível em: https://www.geeksforgeeks.org/modulo-2-binary-division/. Acesso em 22 de novembro de 2019.

TECHNOPEDIA. **Cyclic Redundancy Check (CRC)**: What does Cyclic Redundancy Check (CRC) mean? Disponível em: https://www.techopedia.com/definition/1793/cyclic-redundancy-check-crc. Acesso em 22 de novembro de 2019.

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO - BIBLIOTECA CENTRAL, “Normalização e Apresentação de Trabalhos Científicos e Acadêmicos,” *EDUFES*, acesso em 22 de novembro de 2019, http://edufes.ufes.br/items/show/325.