

# Projeto de Circuitos Integrados

## Implementação de um Multiplicador por Somas Sucessivas e um Extrator de Raiz Quadrada

Cristina Kulczynski

Universidade de Santa Cruz do Sul (UNISC)  
Santa Cruz do Sul – RS – Brasil

kulczynski@mx2.unisc

### Sumário

1.	INTRODUÇÃO .....	2
2.	ESTRUTURA GERAL .....	2
2.1.	Máquina de estados MSS e parte operacional .....	2
2.2.	Máquina de estados SQRT .....	4
3.	DETALHAMENTO DOS MÓDULOS E RTL.....	6
3.1.	TOP LEVEL .....	6
3.1.1.	SIMULAÇÃO TEST BENCH .....	8
3.2.	MSS .....	8
3.2.1.	OPERATIVA.....	9
3.2.2.	CONTROLE .....	9
3.2.3.	SIMULAÇÃO TEST BENCH .....	9
3.3.	SQRT .....	10
3.3.1.	OPERATIVA.....	11
3.3.2.	CONTROLE .....	11
3.3.3.	SIMULAÇÃO TEST BENCH .....	11
4.	RESULTADOS DE SÍNTESE .....	12
4.1.	Uso de Células .....	12
4.2.	Relatório de utilização do dispositivo .....	14
4.3.	Relatório de temporização.....	15
4.4.	Post Place and Route .....	16
5.	CONCLUSÃO .....	18

## 1. INTRODUÇÃO

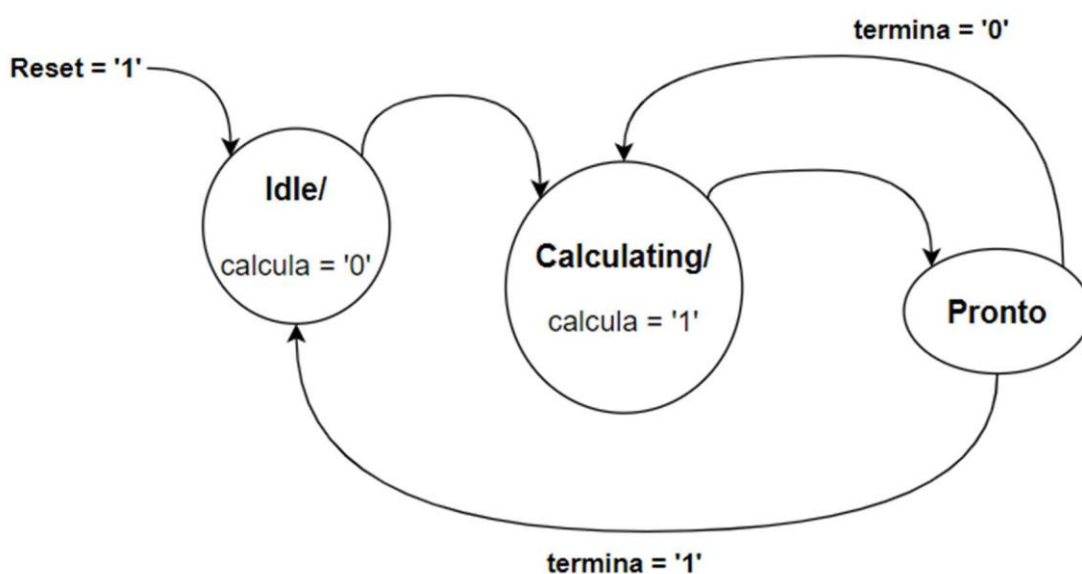
O propósito deste relatório é fornecer as especificações detalhadas e os resultados obtidos no Trabalho 2, que consiste em um projeto prático realizado no âmbito da disciplina de Projeto de Circuitos Integrados. Tendo como objetivo do projeto a implementação de uma calculadora com suporte às operações de MSS (Multiplicação por Somas Sucessivas) e ERQ (Extração de Raiz Quadrada), utilizando a linguagem de descrição de hardware VHDL. Os componentes que compõem esta calculadora incluem as partes de controle e operativa da MSS e ERQ, cada uma com seus respectivos arquivos de encapsulamento, além do módulo que integra os dois encapsulamentos em um único conjunto. Ademais, realizamos testes e simulações complementares de cada módulo, os resultados encontram-se documentados com as suas respectivas imagens ilustrativas e outros elementos complementares. Uma observação importante é que o extrator de raiz quadrada não é capaz de extrair de todos os números, isso o deixa limitado.

## 2. ESTRUTURA GERAL

Neste tópico, exploraremos a lógica, a máquina de estados, buscando uma explicação e compreensão mais eficazes do projeto. Para facilitar essa compreensão, optamos por dividir o projeto em pequenos tópicos e subtópicos que serão abordados ao longo deste relatório, oferecendo uma visão mais detalhada e organizada.

### 2.1. Máquina de estados MSS e parte operacional

Na figura 1 podemos observar o funcionamento da máquina de estados do módulo de multiplicação por somas sucessivas. O processo é acionado pela borda de subida do sinal de clock. FIGURA 1

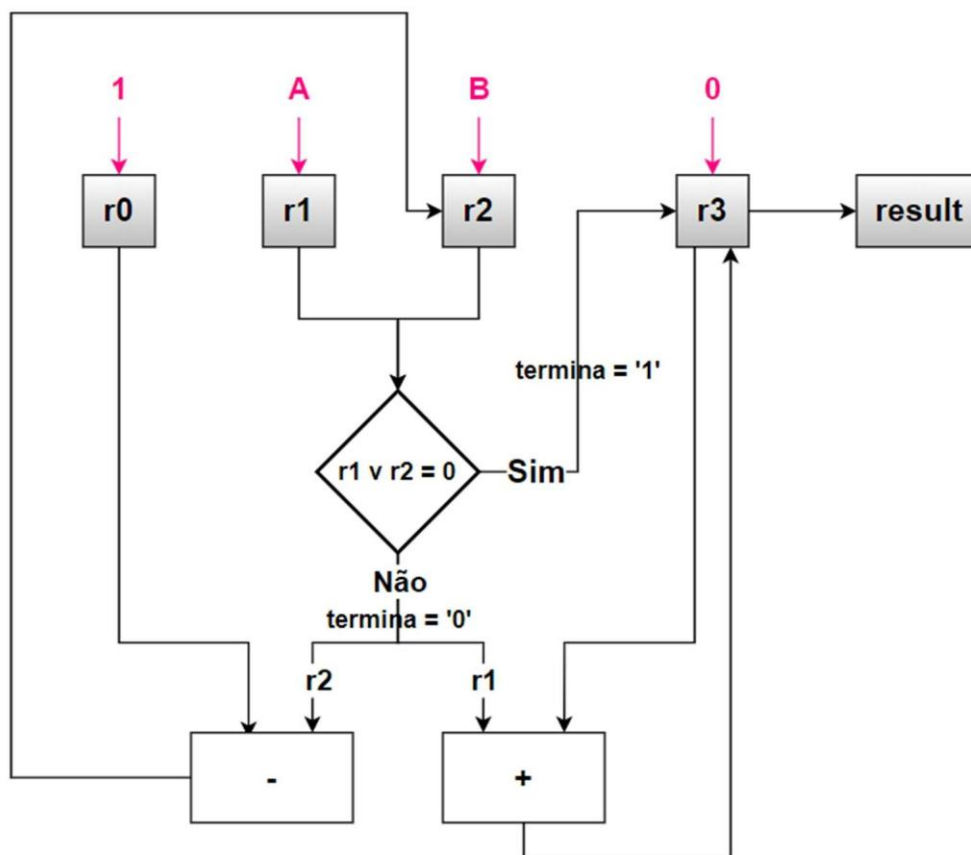


O funcionamento ocorre da seguinte forma:

- Se o sinal de reset for igual a 1, o sistema passa para o próximo estado, que é o Idle. Indicando que está em um estado de repouso ou reinicialização.
- No estado Idle a variável de saída calcula recebe o valor 0 e depois passa para próximo estado Calculating. O sinal de calcula será enviado para a parte operativa e ela irá inicializar todos os registradores e variáveis em zero.
- No estado Calculating a variável calcula recebe o valor 1, indicando para a parte operativa que é o momento de realizar o cálculo, e passa para o próximo estado Pronto.
- No estado Pronto, o sistema verifica constantemente o valor da variável "termina". Essa variável é proveniente da parte operativa do sistema e indica se o cálculo foi finalizado ou não. Se o valor de "termina" for 0, significa que o cálculo ainda está em andamento e o sistema volta para o estado Calculating para continuar o processo de cálculo. Caso contrário, se o valor de "termina" for diferente de 0, o próximo estado será o estado Idle, indicando que o cálculo foi concluído e o sistema está novamente em um estado de repouso.

A figura 2 é uma representação em formato de diagrama da parte operacional do MSS, apesar não ser totalmente fiel ao código, pois não foram mostrados alguns componentes tais como os sinais clk, reset, a entrada calcula e uma variável intermediária.

FIGURA 2



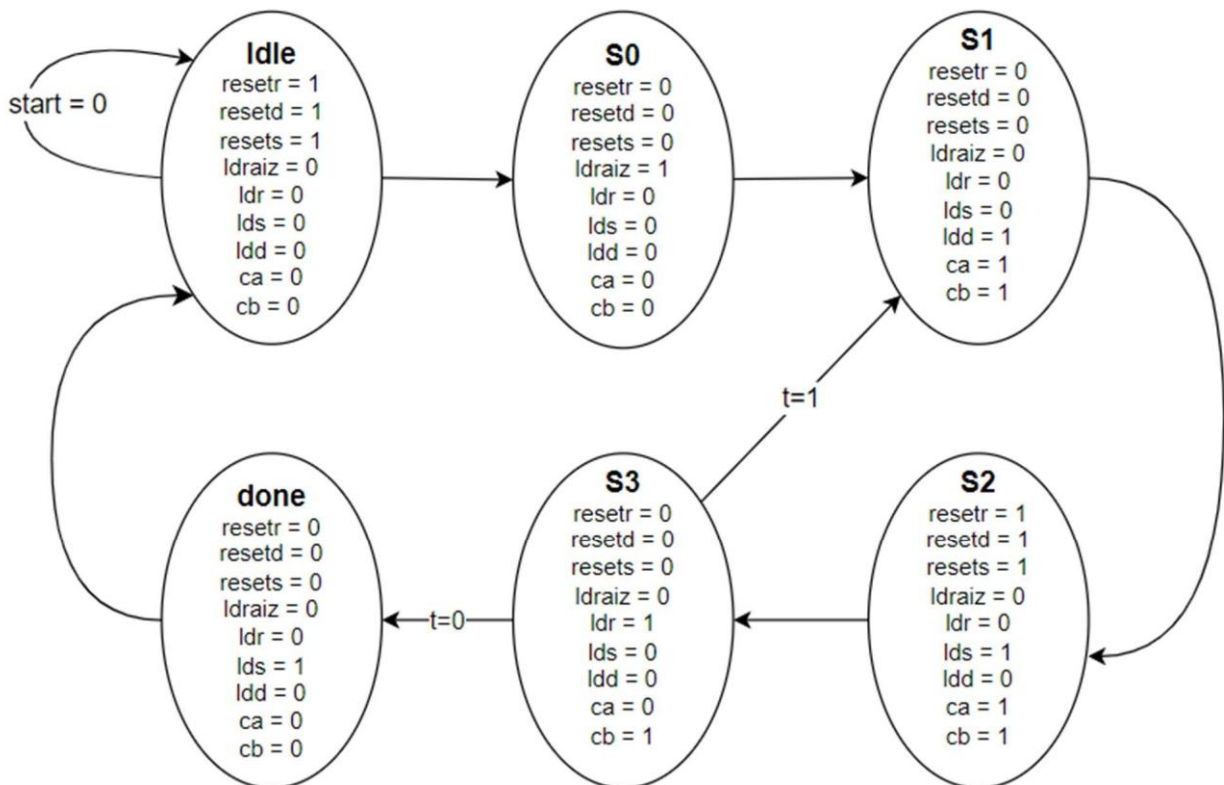
Podemos descrever o funcionamento da seguinte maneira:

- Registrador r0 recebe valor 1.
- Registrador r1 recebe valor da entrada A que é definida no Test Bench.
- Registrador r2 recebe valor da entrada B que também é definida no Test Bench.
- Registrador r3 recebe valor 0.
- Se r1 ou r2 forem iguais a 0, result recebe o valor de r3 a variável termina recebe 1 e acaba o cálculo.
- Se r1 e r2 forem diferentes de 0, a variável termina recebe 0 e continua calculando.
- Soma r1 e r3 e guarda no registrador r3.
- Diminui r0 de r2 e guarda em r2.
- O processo volta ao início e só irá parar quando r2 for igual a 0.

## 2.2. Máquina de estados SQRT

O diagrama de estados apresentado na Figura 3 revela um comportamento significativamente diferente do diagrama anterior. Neste caso, é perceptível que a parte de controle exerce total domínio sobre a parte operativa.

FIGURA 3

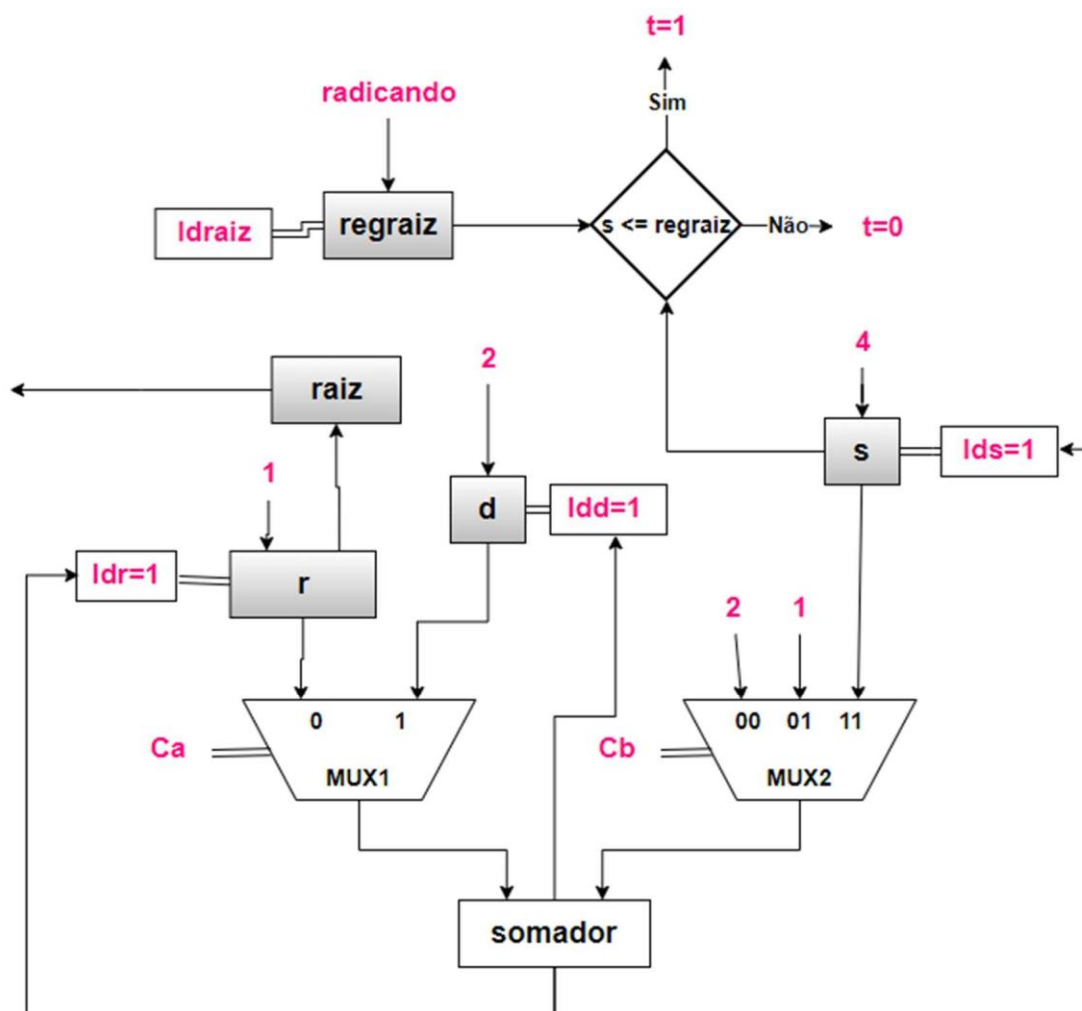


A seguir, apresenta-se de forma sucinta a explicação sobre o funcionamento do diagrama representado na Figura 3.

- Se start for igual a 0 continua no estado Idle, caso contrário passa para S0
- O sistema passará pelos estados S0, S1 e S2, e quando chegar ao estado S3, verificará se t é igual a 1. Se for o caso, retornará ao estado S1 e permanecerá em loop até que t seja igual a 0. Caso contrário, o sistema avançará para o estado "done" e, em seguida, para o estado "Idle".
- Se o valor de Load (ldraiz, ldr, lds ou ldd) for igual a 1, ocorrerá o carregamento de um novo valor, caso contrário ocorrerá o carregamento do valor atual.
- Ca e cb são responsáveis por controlar os multiplexadores.
- Os sinais de reset (resetr, resetd, resets) são utilizados para reiniciar os registradores com valores pré-definidos, conforme o algoritmo de extração de raízes.

A Figura 4 ilustra a parte operativa, que inclui as entradas, cargas, resets, registradores, multiplexadores e um somador. Além disso, é apresentada a lógica responsável por verificar se o cálculo foi concluído.

FIGURA 4



Para descrever o funcionamento do diagrama, serão utilizadas as nomenclaturas presentes no código original. No entanto, é importante ressaltar que as variáveis **resetr**, **resetd** e **resets** não são representadas no diagrama. Essas variáveis determinam o estado dos registradores **r**, **d** e **s**, atribuindo-lhes os valores 1, 2 e 4 quando estão ativadas, respectivamente.

- Os registradores r, d e s são inicializados com os valores 1, 2 e 4, respectivamente. O valor do radicando é fornecido no Test Bench e, neste caso, é 9. Neste ponto, o loop é iniciado.
  - o  $r = 1, d = 2, s = 4, \text{radicando} = 9$
- No início, Ca é definido como 0, fazendo com que o valor de r seja processado.
- Em seguida, Cb é definido como 01, permitindo que o valor 1 seja processado.
- O Somador adiciona 1 a r e armazena o resultado em r se ldr for igual a 1.
  - o  $r = r + 1 \ (r = 2)$
- Depois disso, Ca é definido como 1, permitindo que o valor de d seja processado.
- Cb recebe 00, o que faz com que o valor 2 passe.
- O Somador adiciona 2 a d e armazena o resultado novamente em d. o  $d = d + 2 \ (d = 4)$
- Agora, Cb recebe 1 e, em seguida, recebe 11 e, finalmente, 01.
- Assim o somador soma s, d e 1 e armazena em s. o  $s = s + d + 1 \ (s = 4 + 4 + 1 = 9)$
- É verificado se s é menor ou igual à raiz quadrada, se for, t é definido como 0; caso contrário, t é definido como 1.
- Se t for igual a 0, o loop é interrompido e o resultado de r é obtido.

### 3. DETALHAMENTO DOS MÓDULOS E RTL

Neste tópico e subtópicos será mostrado o esquemático em nível de transferência de registros, podemos ver se a lógica do circuito está correta, se os sinais estão sendo transferidos.

#### 3.1. TOP LEVEL

Na figura 5 podemos observar a estrutura externa da calculadora de MSS e SQRT, sendo essa a interface principal do projeto.

FIGURA 5

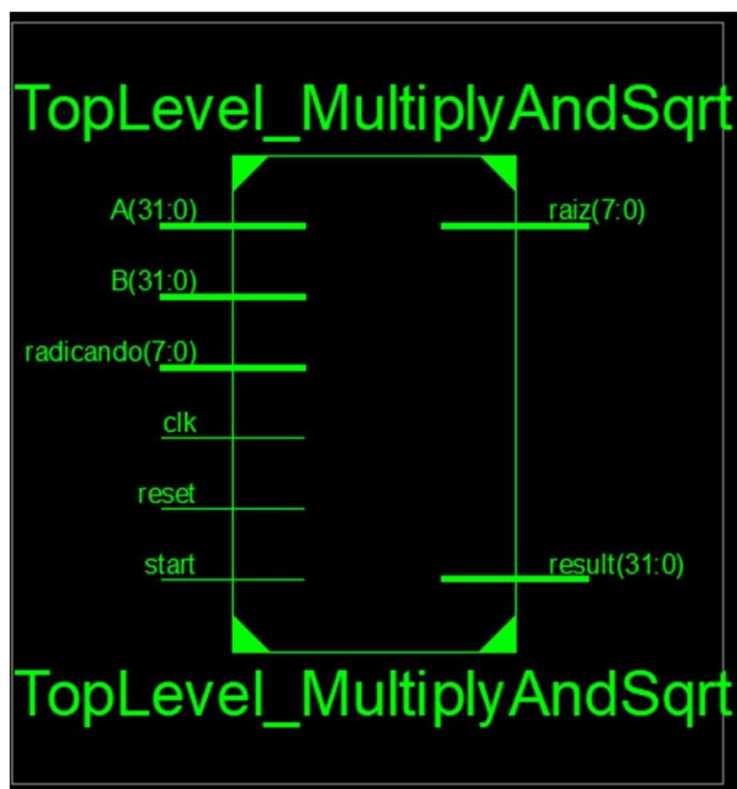
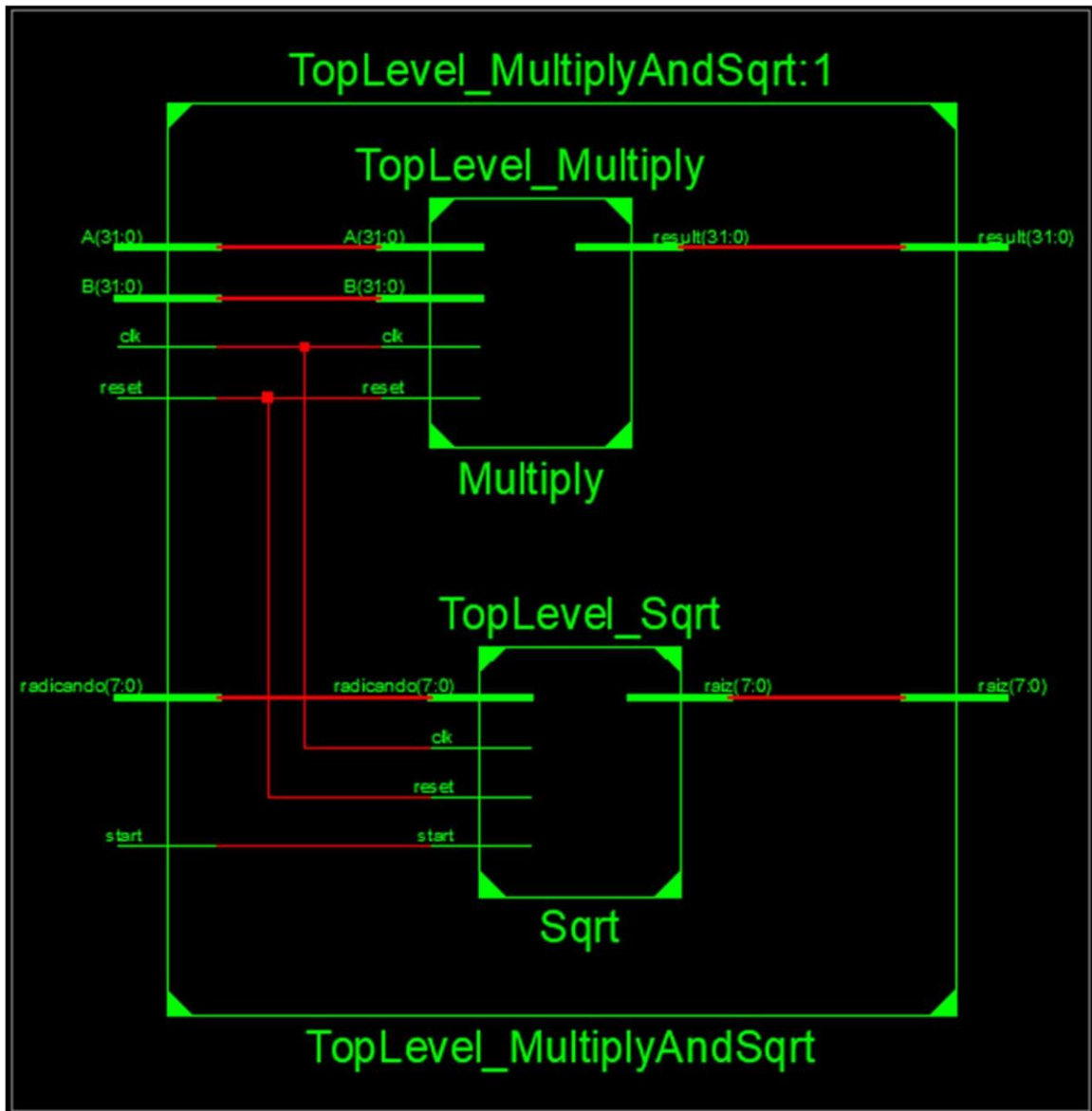


FIGURA 6



A figura 6 apresenta a parte interna do **TopLevel\_MultiplyAndSqrt**, podemos observar seus componentes e a interconexão entre eles.

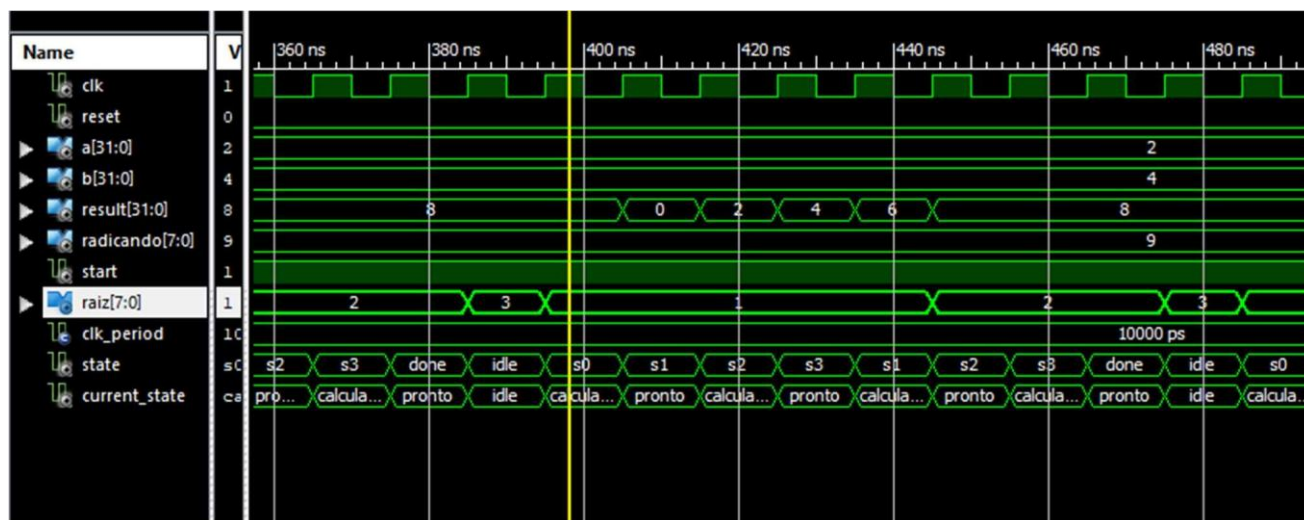
Abaixo temos a pinagem:

Nome do pino	Descrição detalhada	Tipo
reset	Restaurar o circuito.	Entrada
start	Inicia os estados do extrator de raiz quadrada.	Entrada
clk	Controla o tempo para cada ação ocorrer.	Entrada
radicando(7:0)	Valor do radicando para o cálculo da raiz.	Entrada
A(31:0)	Valor de A que irá somar com ele mesmo.	Entrada
B(31:0)	Valor de B que irá diminuir até 0.	Entrada

raiz	Resultado da raiz.	Saída
result	Resultado da multiplicação por somas sucessivas.	Saída

### 3.1.1. SIMULAÇÃO TEST BENCH

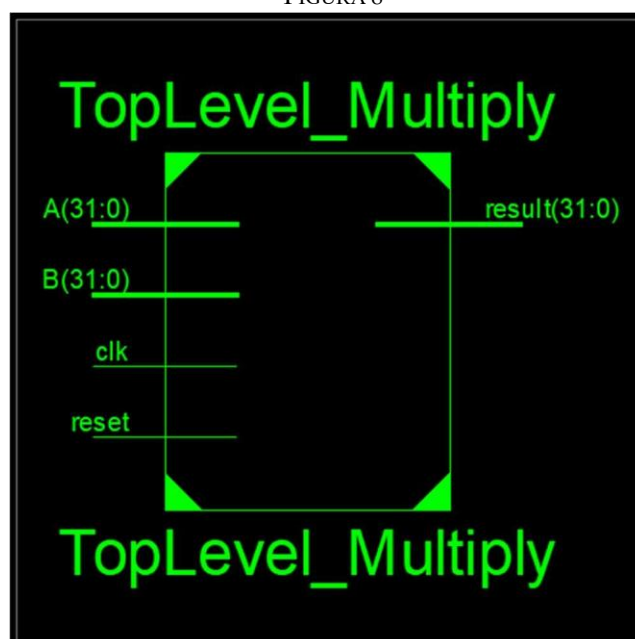
FIGURA 7



### 3.2. MSS

Este módulo TopLevel\_Multiply serve para encapsular a parte operativa da parte controle com o intuito de ligar os sinais

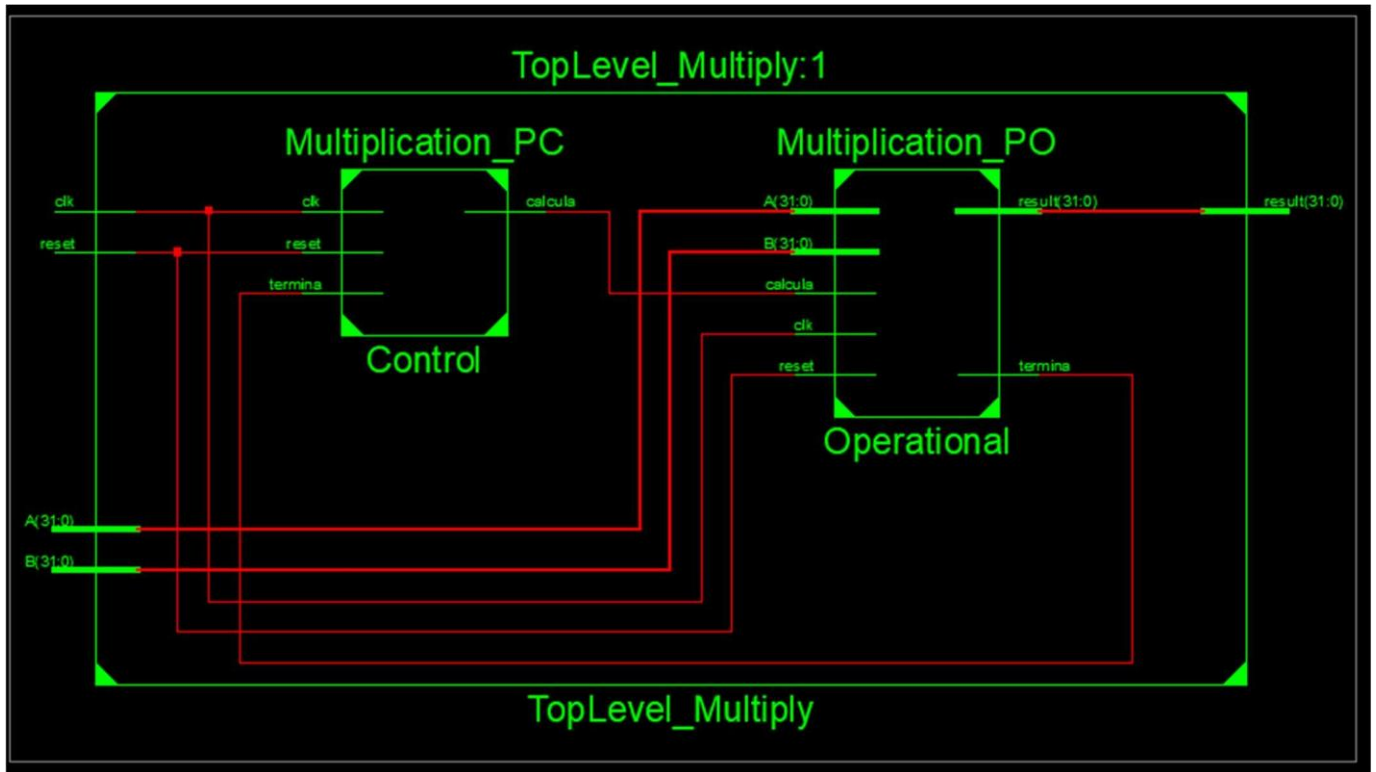
FIGURA 8



Parte interna:



FIGURA 9



O controle e a parte operacional se conectam pelas portas calcula e termina.

### 3.2.1. OPERATIVA

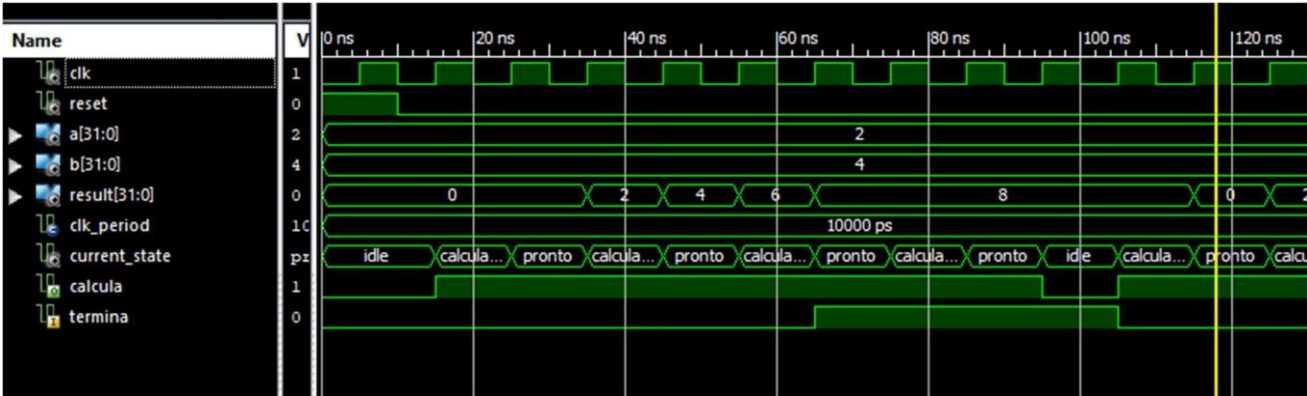
O componente possui várias portas de entrada e saída, incluindo clk (sinal de clock), reset (sinal de reinicialização), termina (sinal de finalização), calcula (sinal de controle de cálculo), A (primeiro número de entrada), B (segundo número de entrada) e result (resultado da multiplicação). Resumindo, o módulo é um componente que realiza a multiplicação de dois números inteiros sem sinal de 32 bits quando um sinal de controle é ativado. Ele acompanha o estado do cálculo e sinaliza quando a multiplicação é concluída.

### 3.2.2. CONTROLE

O funcionamento desse módulo se resume em circuito de multiplicação controlado por estados. O estado Idle indica que a multiplicação não está ocorrendo. O estado Calculating ativa a multiplicação. O estado Pronto permite a transição de volta ao estado Idle se o sinal termina for ativado, caso contrário, volta para Calculating. O sinal calcula controla o cálculo da multiplicação. O sinal de relógio clk atualiza os estados. O sinal de reset redefine o estado para Idle. Clk, reset e termina são sinais de entrada enquanto calcula é um sinal de saída.

### 3.2.3. SIMULAÇÃO TEST BENCH

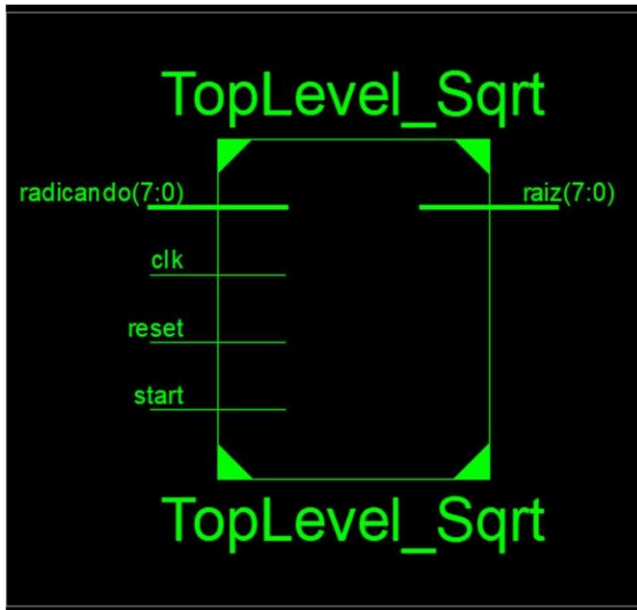
10



### 3.3. SQRT

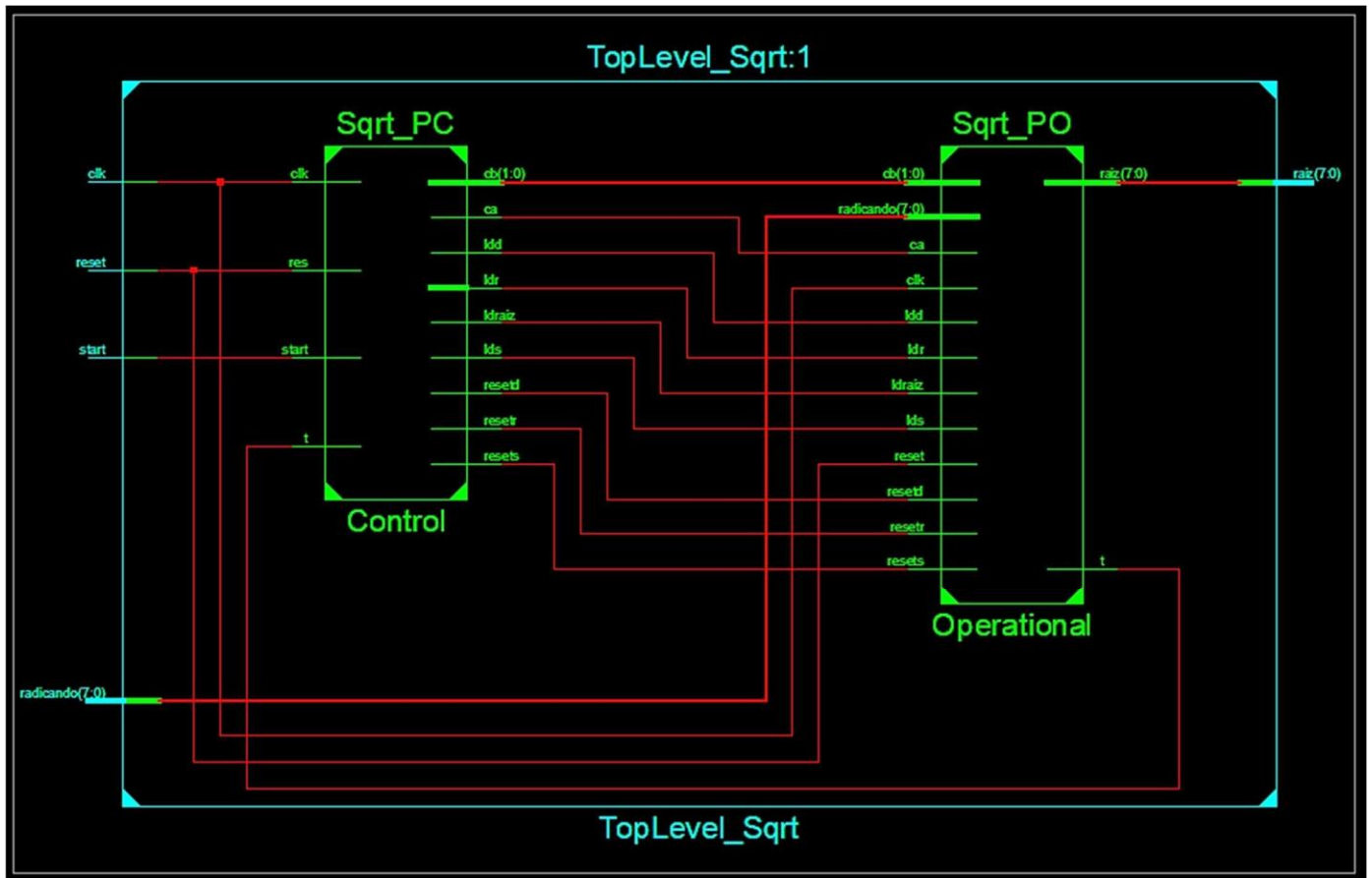
O `TopLevel_Sqrt` tem a função de ser um encapsulamento para a parte operativa e controle do circuito, unindo as duas.

FIGURA 11



Na figura 12 vemos a parte interna.

FIGURA



### 3.3.1. OPERATIVA

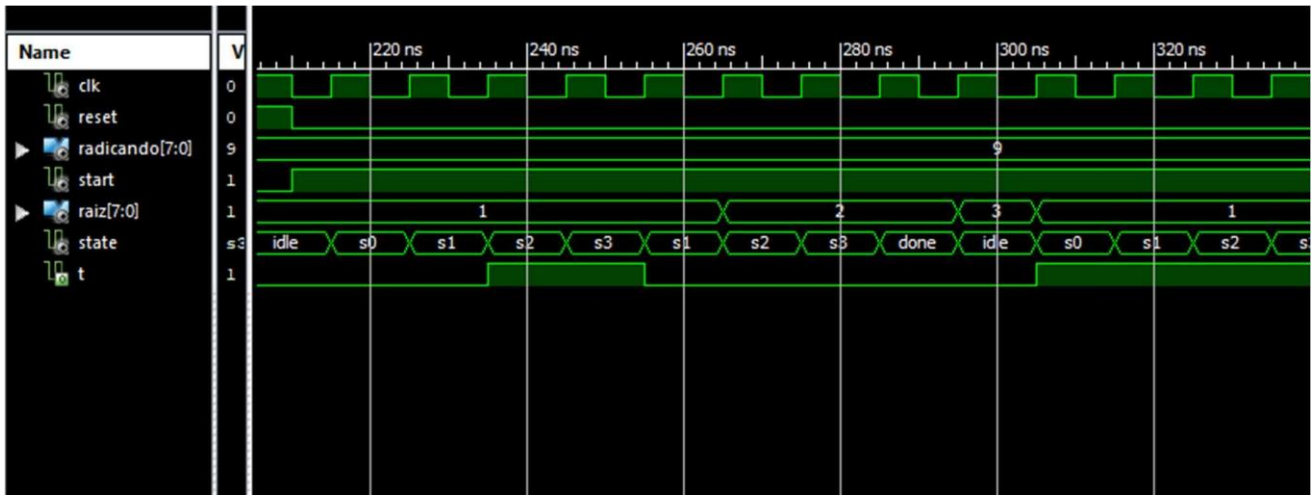
A entidade "Sqrt\_PO" possui vários sinais de entrada e saída, incluindo sinais para controlar o clock, reset, o número cuja raiz quadrada será calculada, sinais de controle para carregar valores nos registradores, sinais de controle para selecionar entradas dos multiplexadores e sinais de saída para o resultado da raiz quadrada.

### 3.3.2. CONTROLE

O módulo "Sqrt\_PC" é um controlador de estado que gera sinais de controle para um circuito de cálculo da raiz quadrada de um número. Ele possui sinais de entrada e saída para controlar o clock, reset e início do cálculo, com o sinal "t" sendo uma entrada adicional. Os sinais de saída incluem vários controles (resetr, resetd, resets, ldraiz, ldr, lds, ldd, ca e cb). Em resumo, o código implementa um controlador que utiliza diferentes estados para controlar os sinais de saída com base nas transições de estado e nas condições de entrada, garantindo o controle adequado do circuito durante o cálculo da raiz quadrada.

### 3.3.3. SIMULAÇÃO TEST BENCH

FIGURA

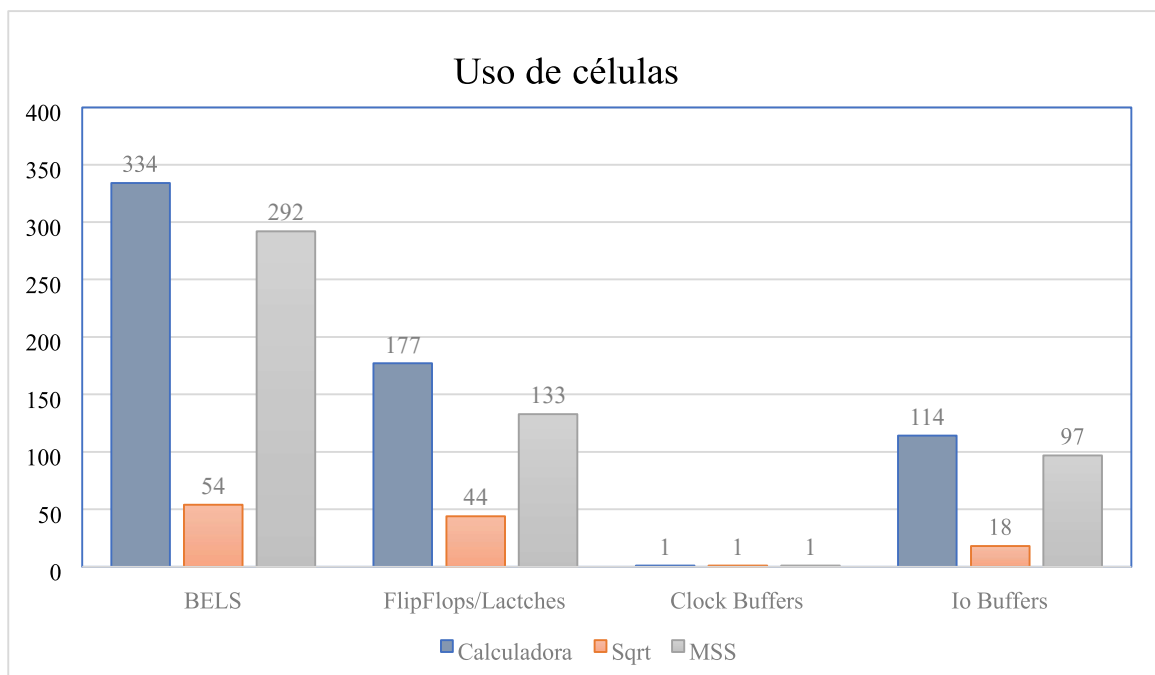


#### 4. RESULTADOS DE SÍNTESE

O objetivo deste tópico é apresentar e comparar os resultados da síntese realizada por meio da ferramenta Synthesize XST e FPGA Editor para a calculadora e seus módulos. Serão analisados os seguintes aspectos: o uso de células, relatório de utilização do dispositivo, relatório de temporização e a visão geral de posicionamento e roteamento.

##### 4.1. Uso de Células

Utilizando a ferramenta Design Summary/Reports (Summary) e acessando o Synthesis Report → Cell Usage, foram obtidas informações sobre a quantidade e tipos de células utilizadas na síntese do projeto da calculadora e seus módulos. Na tabela abaixo podemos observar o uso delas e a diferença que apresentam.



FIGURA

A figura 14 mostra o uso das células da Calculadora, é esperado que ela utilize mais componentes, pois ela integra os dois módulos MSS e SQRT.

FIGURA 14 - CALCULADORA

```
Cell Usage :
# BELS : 344
# GND : 1
# INV : 2
# LUT2 : 106
# LUT3 : 44
# LUT4 : 25
# MUXCY : 93
# VCC : 1
# XORCY : 72
# FlipFlops/Latches : 177
# FDC : 5
# FDCE : 159
# FDE : 6
# FDP : 1
# FDPE : 3
# FDR : 2
# LD_1 : 1
# Clock Buffers : 1
# BUFGP : 1
# IO Buffers : 114
# IBUF : 74
# OBUF : 40
```

Na figura 15. é visível que o circuito utiliza uma grande quantidade de células em relação ao Sqrt

15 - MSS

```
Cell Usage :
# BELS : 292
# GND : 1
# INV : 1
# LUT2 : 95
# LUT3 : 35
# LUT4 : 17
# MUXCY : 78
# VCC : 1
# XORCY : 64
# FlipFlops/Latches : 133
# FDCE : 130
# FDR : 2
# LD_1 : 1
# Clock Buffers : 1
# BUFGP : 1
# IO Buffers : 97
# IBUF : 65
# OBUF : 32
```

Na figura 16 por exemplo podemos observar que o número total de blocos elementares lógicos representa apenas 16 % em relação aos utilizados pela Calculadora

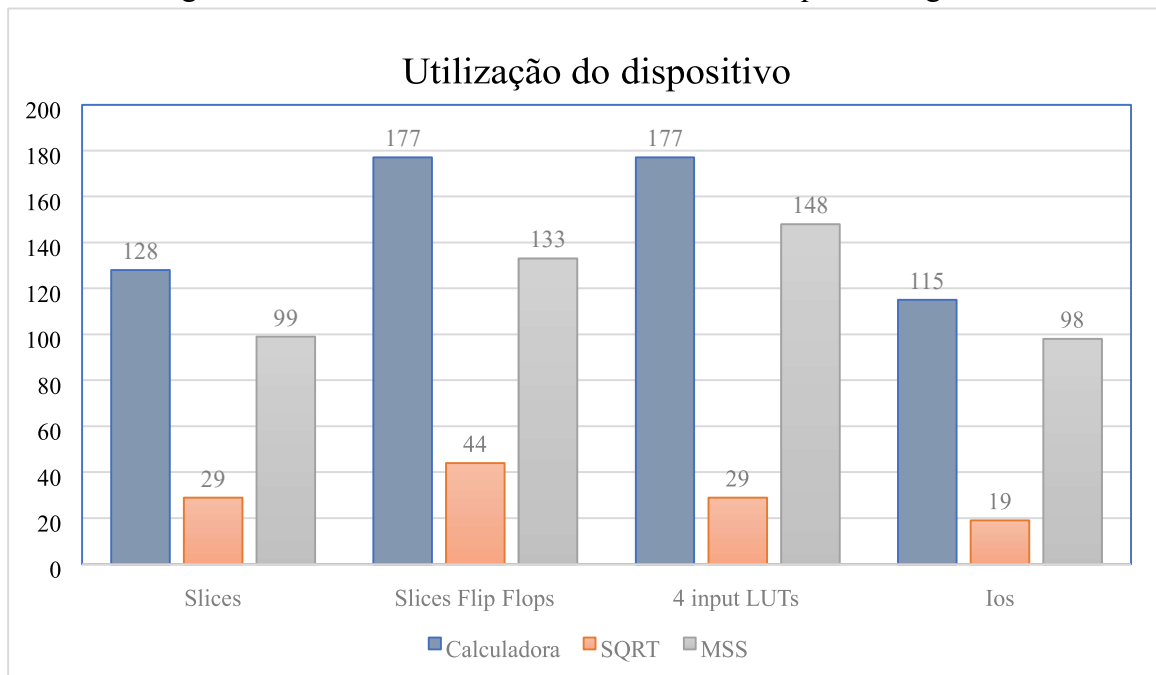
FIGURA 16 - SQRT

```
Cell Usage :
# BELS : 54
# GND : 1
# INV : 1
# LUT2 : 11
# LUT3 : 9
# LUT4 : 8
# MUXCY : 15
# VCC : 1
# XORCY : 8
# FlipFlops/Latches : 44
# FDC : 5
# FDCE : 29
# FDE : 6
# FDP : 1
# FDPE : 3
# Clock Buffers : 1
# BUFGP : 1
# IO Buffers : 18
# IBUF : 10
# OBUF : 8
```

#### 4.2. Relatório de utilização do dispositivo

Este relatório apresenta um resumo da utilização geral do dispositivo de destino. Ele fornece informações sobre a quantidade de recursos (como LUTs, flip-flops, memória etc.) utilizados pelo projeto. Isso permite verificar se o design está dentro dos limites do dispositivo escolhido.

Veremos no gráfico abaixo as nuances entre os circuitos e o que elas significam.



□ Calculadora:

FIGURA 17

```
Device utilization summary:
-----
Selected Device : 3s200ft256-4

Number of Slices:          128 out of 1920    6%
Number of Slice Flip Flops: 177 out of 3840    4%
Number of 4 input LUTs:    177 out of 3840    4%
Number of IOs:             115
Number of bonded IOBs:      115 out of 173    66%
Number of GCLKs:            1 out of 8        12%
```

□ SQRT:

FIGURA 18

```
Device utilization summary:
-----

Selected Device : 3s200ft256-4

Number of Slices:                29 out of 1920    1%
Number of Slice Flip Flops:      44 out of 3840    1%
Number of 4 input LUTs:         29 out of 3840    0%
Number of IOs:                   19
Number of bonded IOBs:          19 out of 173    10%
Number of GCLKs:                 1 out of 8       12%
```

□ MSS:

FIGURA 19

```
Device utilization summary:
-----

Selected Device : 3s200ft256-4

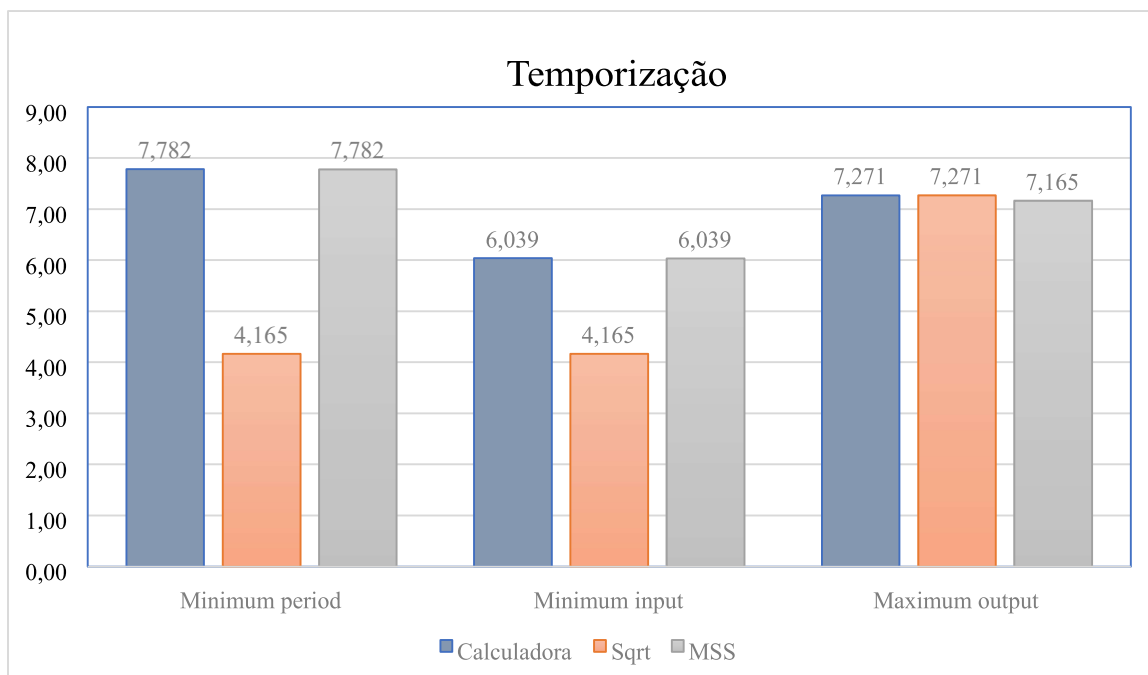
Number of Slices:                99 out of 1920    5%
Number of Slice Flip Flops:      133 out of 3840   3%
Number of 4 input LUTs:         148 out of 3840   3%
Number of IOs:                   98
Number of bonded IOBs:          98 out of 173    56%
Number of GCLKs:                 1 out of 8       12%
```

Após a análise do gráfico e figuras 17, 18 e 19 vimos que os Blocos de Entrada/Saída foram demasiadamente utilizados o que pode comprometer o sistema se não for bem gerenciado.

#### 4.3. Relatório de temporização

Este tópico fornece informações sobre o desempenho do circuito em relação aos requisitos de temporização. Ele inclui detalhes como o período mínimo necessário para a operação correta do circuito e a frequência máxima alcançada. Isso ajuda a garantir que o projeto atenda aos requisitos de velocidade do sistema.





Calculadora:

FIGURA 20

```
Timing Summary:
-----
Speed Grade: -4

Minimum period: 7.782ns (Maximum Frequency: 128.500MHz)
Minimum input arrival time before clock: 6.039ns
Maximum output required time after clock: 7.271ns
Maximum combinational path delay: No path found
```

MSS:

FIGURA 21

```
Timing Summary:
-----
Speed Grade: -4

Minimum period: 7.782ns (Maximum Frequency: 128.500MHz)
Minimum input arrival time before clock: 6.039ns
Maximum output required time after clock: 7.165ns
Maximum combinational path delay: No path found
```

SQRT:

FIGURA 22

```
Timing Summary:
-----
Speed Grade: -4

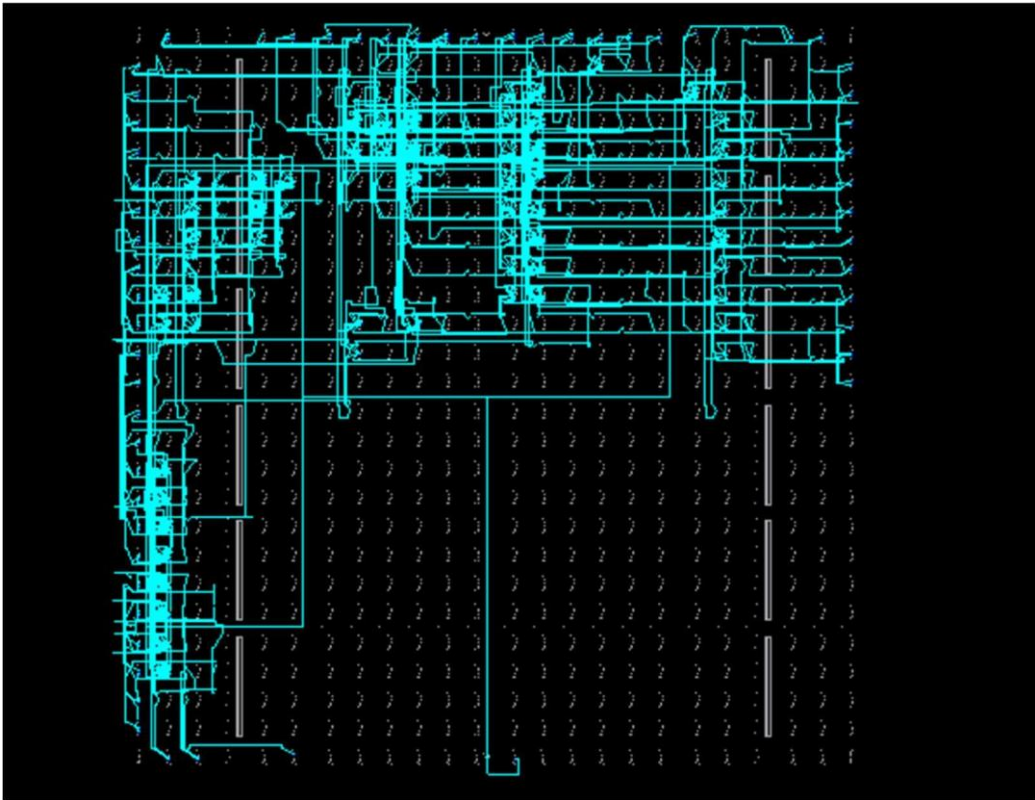
Minimum period: 5.896ns (Maximum Frequency: 169.606MHz)
Minimum input arrival time before clock: 4.165ns
Maximum output required time after clock: 7.271ns
Maximum combinational path delay: No path found
```

#### 4.4. Post Place and Route

Abaixo veremos a representação visual do posicionamento dos componentes do circuito e as conexões entre eles.

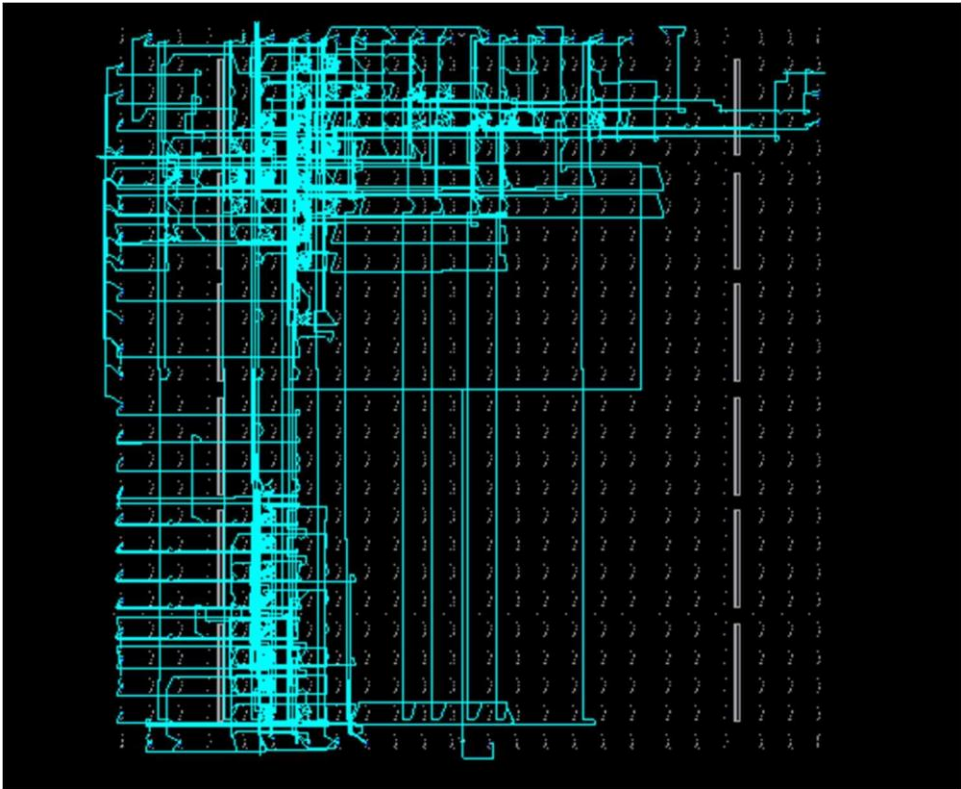
Calculadora:

FIGURA 23



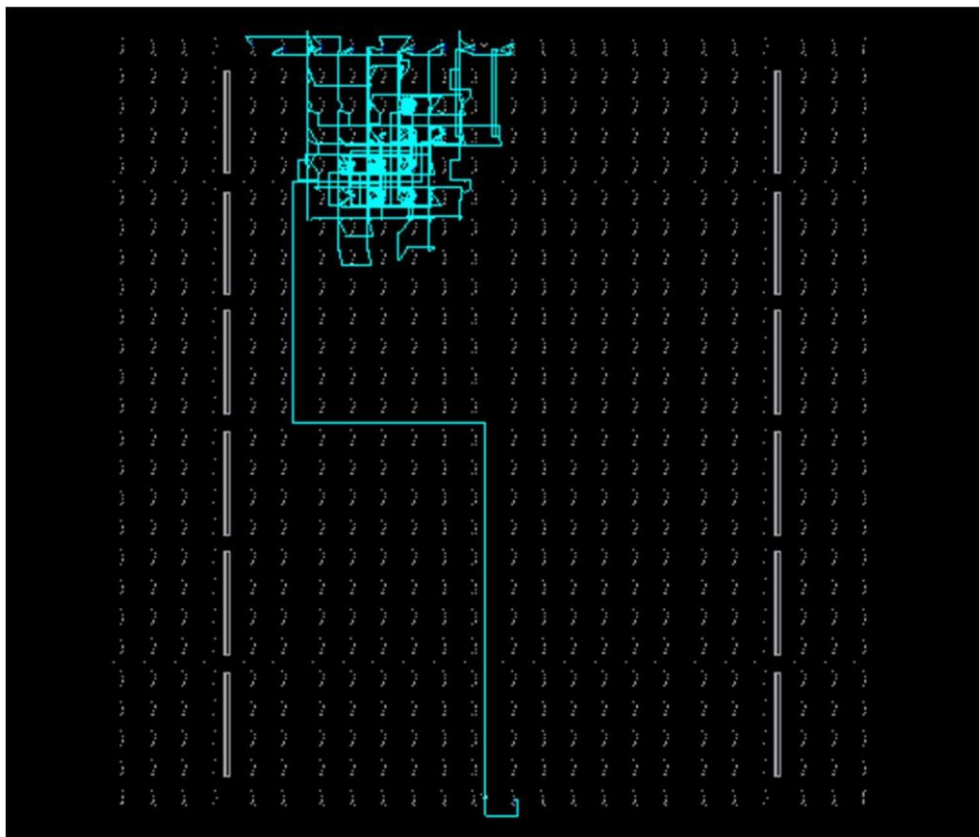
MSS:

FIGURA 24



SQRT:

FIGURA 25



## 5. CONCLUSÃO

A implementação da calculadora com suporte às operações de MSS e ERQ utilizando VHDL foi bem-sucedida de acordo com as simulações. A especificação, projeto, implementação, e avaliação dos circuitos integrados foram realizados com o uso de teorias e materiais disponibilizados em aula, também foram utilizadas práticas e ferramentas apropriadas. Os níveis de abstração e descrição foram considerados para proporcionar uma visão abrangente e detalhada do projeto. A separação de conceitos entre hardware e software, aplicação e arquitetura, e processamento e comunicação também foi abordada. Cabe ainda salientar que, durante o projeto, foram empregadas técnicas de verificação funcional, incluindo a elaboração e geração de testbenches, assim como a análise de diagramas de formas de onda. Essas práticas garantiram o funcionamento correto dos circuitos integrados implementados.