

Nome: Leonardo Amorim de Araújo

Matrícula: 15/0039921

Questão 1

Esquemático RLT do projeto

Primeiramente, para cada equação foi gerado um componente. Na Figura 1 segue os componentes utilizados:

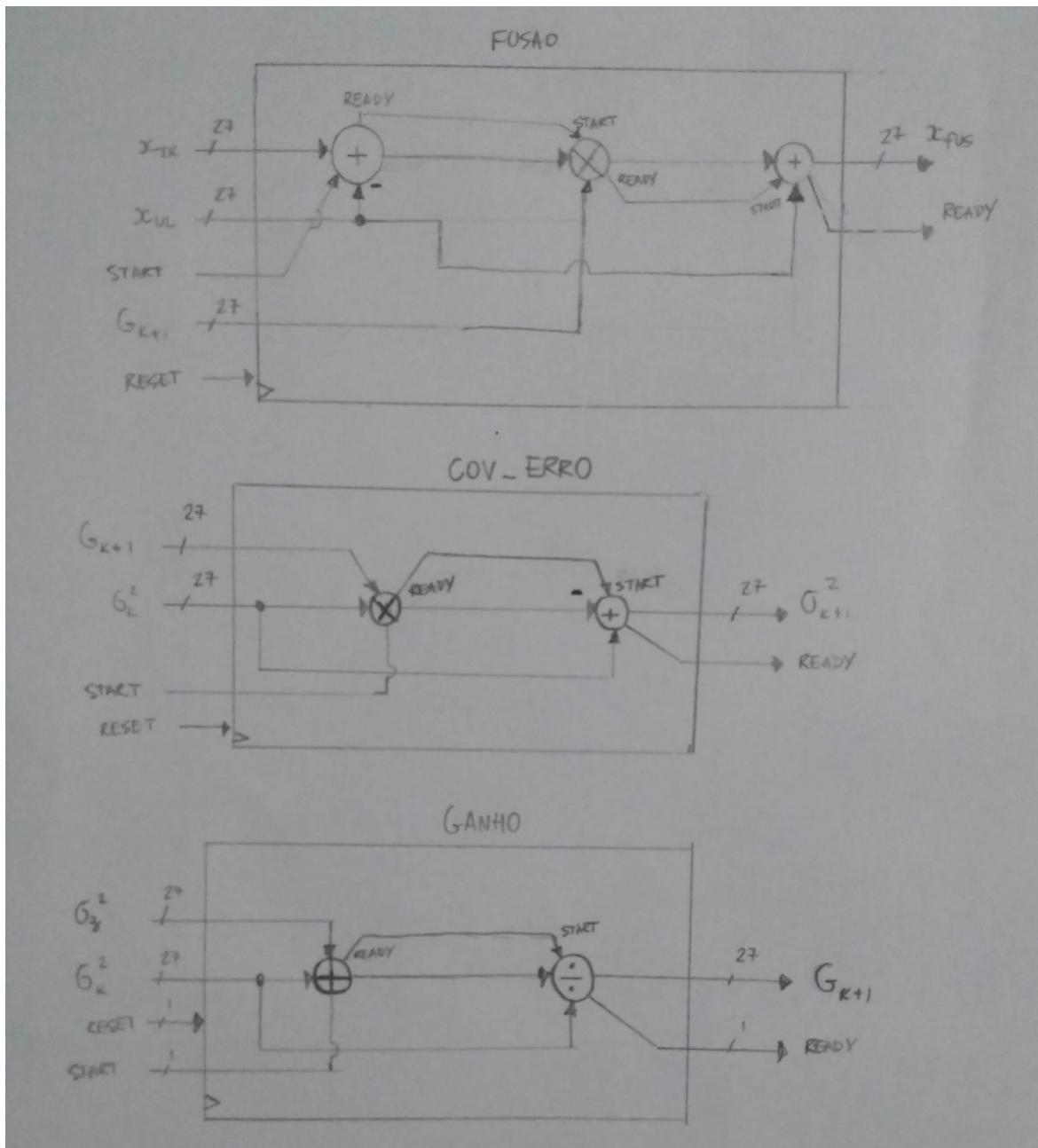


Figura 1: Componentes cov_erro, ganho e fusao que implementam as três equações

O componente **fusao.vhd** implementa a equação final que determina x_{fus} . O componente **cov_erro.vhd** implementa a equação que faz o cálculo da covariância σ_k . Já o componente

ganho.vhd implementa a equação que computa G_{k+1} . O RTL dos três componentes pode ser visualizado na Figura 2 abaixo:

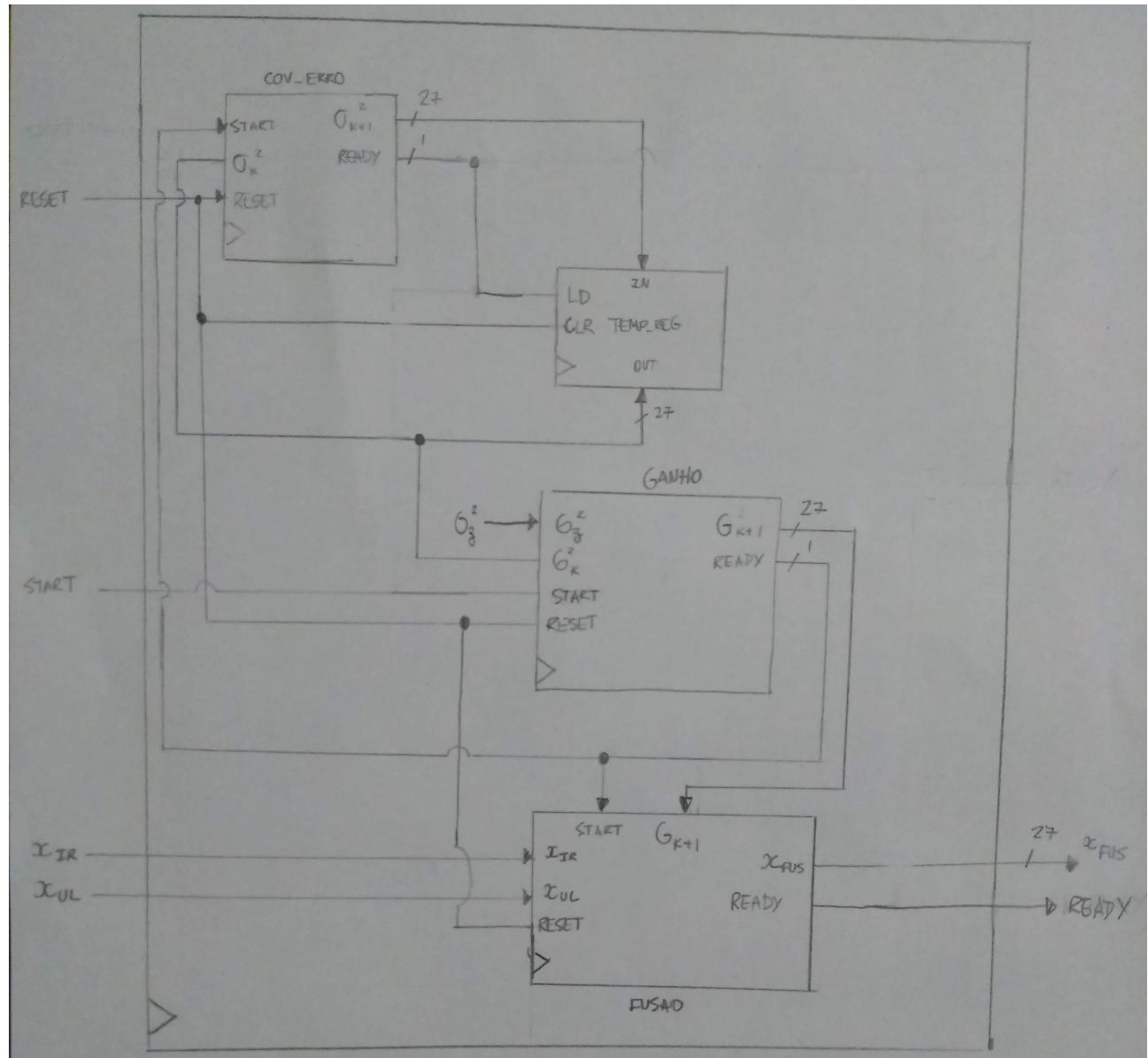


Figura 2: RLT que implementa as três equações e apresenta o resultado final na saída

Questão 2

Na pasta chamada **simauto** encontram-se todos os arquivos **.m** utilizados na simulação. Com o arquivo **simauto.m** foram gerados 100 valores de teste utilizando o gerador aleatório com a semente **rng('twister',150039921)**. Os valores de covariância foram gerados utilizando a função **rand()** dentro do intervalo especificado pelo problema, e os valores encontrados foram $\sigma_k = 0,1912$ e

$\sigma_z = 0.588$. Gerou-se também utilizando a função `randn()` 100 valores com distribuição normal, média em 100 cm e variância σ_z para `x_ul` e σ_k para `x_ir`. Gerou-se também aleatoriamente o valor inicial de $\sigma_k = 0,1912$ e $\sigma_z = 0.588$.

Os valores gerados para estas variáveis foram enviados para os arquivos `.txt` chamados de `x_ir_bin.txt` e `x_ul_bin.txt`, que foram utilizados no testbench, denominado `tb_sensorial.vhd`. O arquivo com os valores de saída obtidos no testbench é chamado `x_fus_bin.txt`.

Utilizando o arquivo `deco_sensorial.m` os valores obtidos pelo testbench foram comparados com os valores calculados no MATLAB. Para o erro quadrático de cada amostra, obteve-se o seguinte gráfico da Figura 3, que mostra que os erros ficaram dentro de uma precisão razoável para cada amostra:

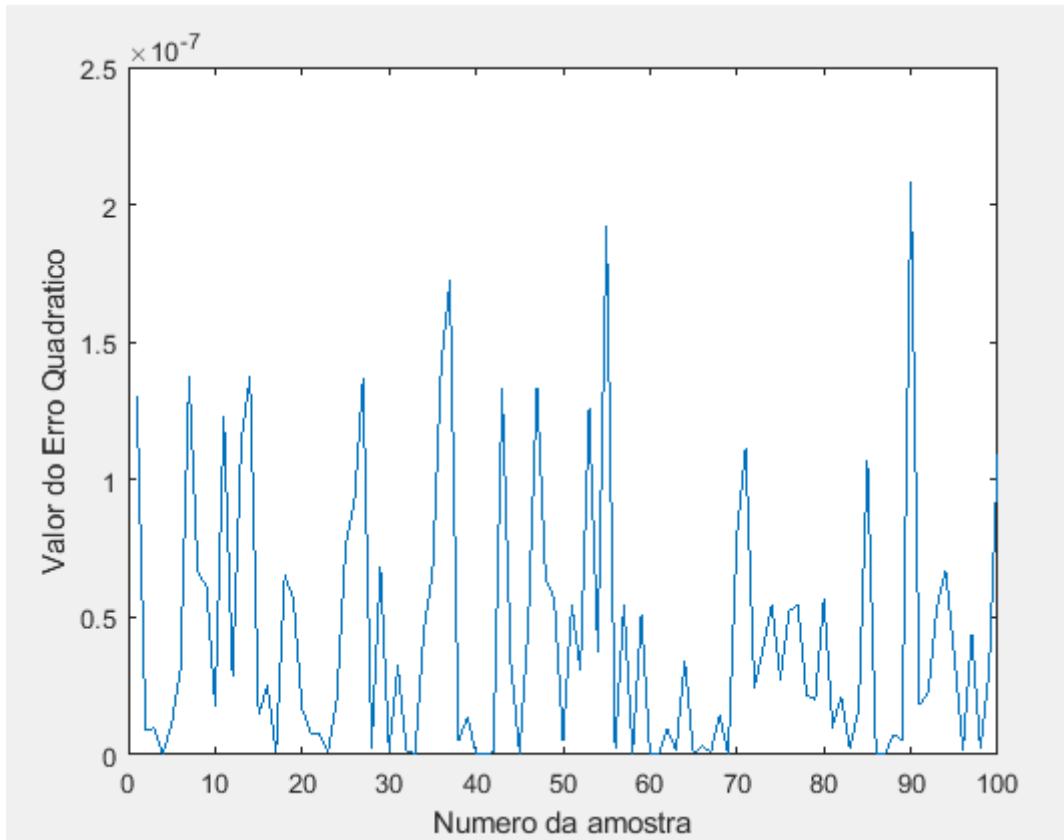


Figura 3: Erro quadrático para cada amostra

O erro quadrático médio calculado foi $4,34 \times 10^{-8}$.

Questão 3

O clock utilizado para esta simulação possui 100 MHz de frequência, ou 10 ns de período.

Desde o primeiro start, ate que a saída de número 100 fosse calculada, verificou-se que o tempo de execução foi de 19,99 us. Ou seja, gastou-se 1999 ciclos de clock até que saída de número 100 fosse calculada, como pode ser visualizado na Figura 4.

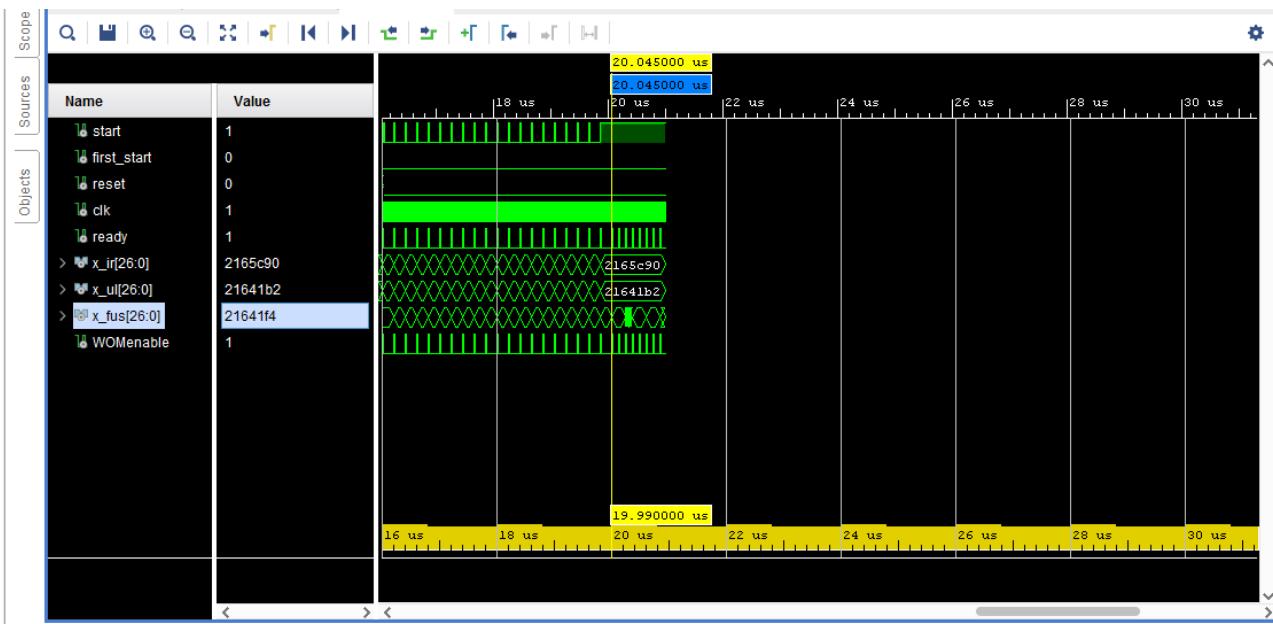


Figura 4: Simulação demonstrando o tempo necessário para executar todos os 100 primeiros valores de teste

Na Figura 5, pode ser visualizado que a latência da solução ficou em 190 ns, ou seja, 19 ciclos de clock.

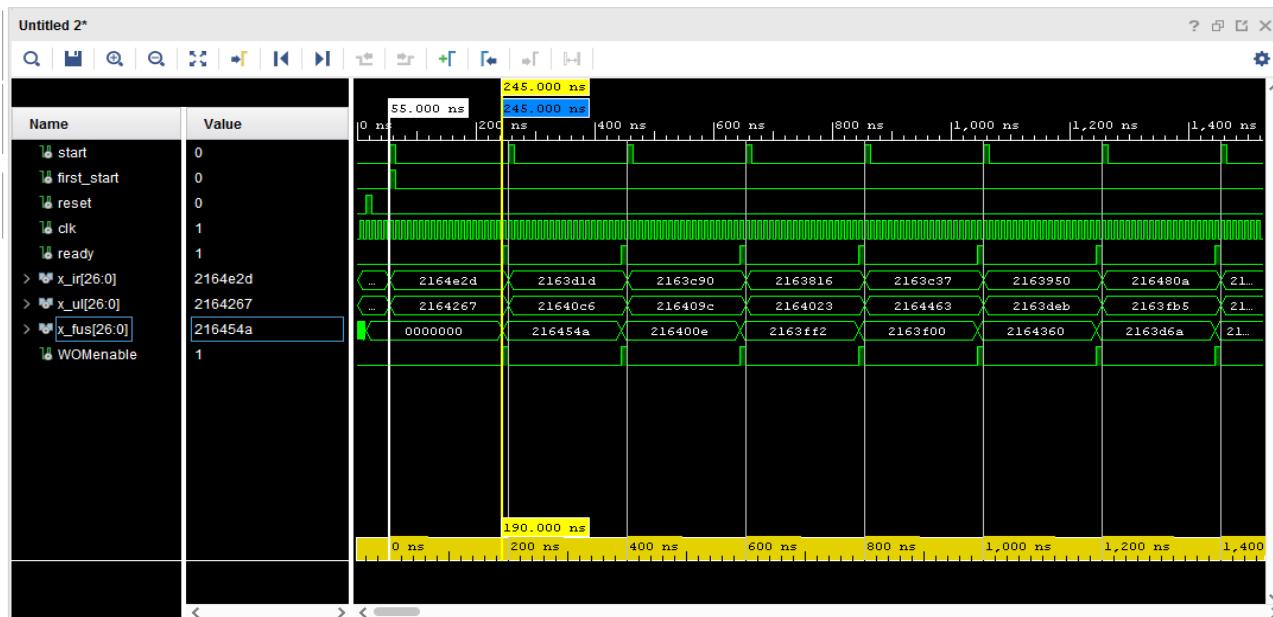


Figura 5: Latência da Solução encontrada

Já o throughput pode ser visualizado na Figura 6. Como pode ser visto, o throughput da solução foi 200 ns, ou seja, 20 ciclos de clock.

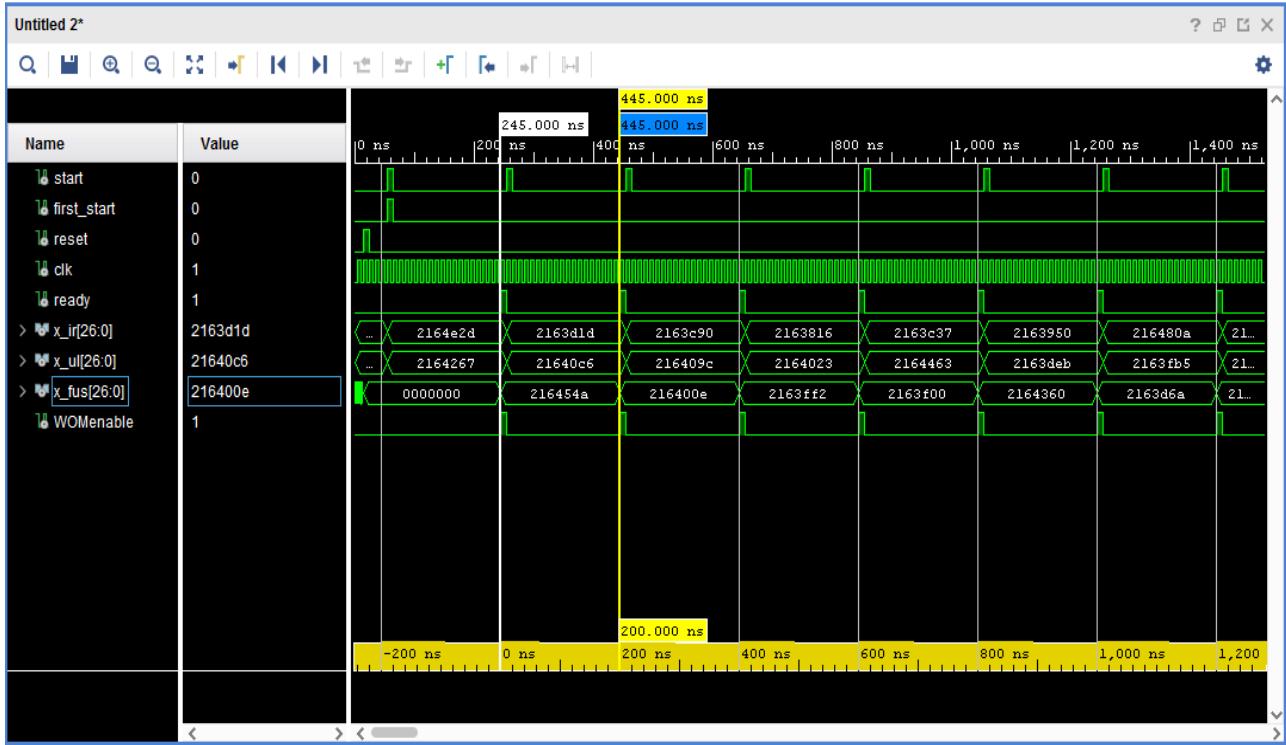


Figura 6: Throughput da solução

Questão 4

Um **bloco operacional** foi criado, com duas memórias ROM geradas pelo IP Generator do Vivado. Cada memória tem 100 posições de endereço e 27 bits de dados. Os dados foram inseridos nas memórias através de dois arquivos .coe gerados pelo arquivo **simaauto.m** no MATLAB. O componente **sensorial**, que é o bloco que realiza as equações, foi instanciado dentro deste bloco operacional, recebendo os valores de x_{IR} e x_{UL} das memórias. Um contador de endereços para as memórias ROM foi implementado para ter seu valor atualizado cada vez que um novo valor de saída x_{FUS} é gerado. O formato final do bloco operacional pode ser visualizado na Figura 7.

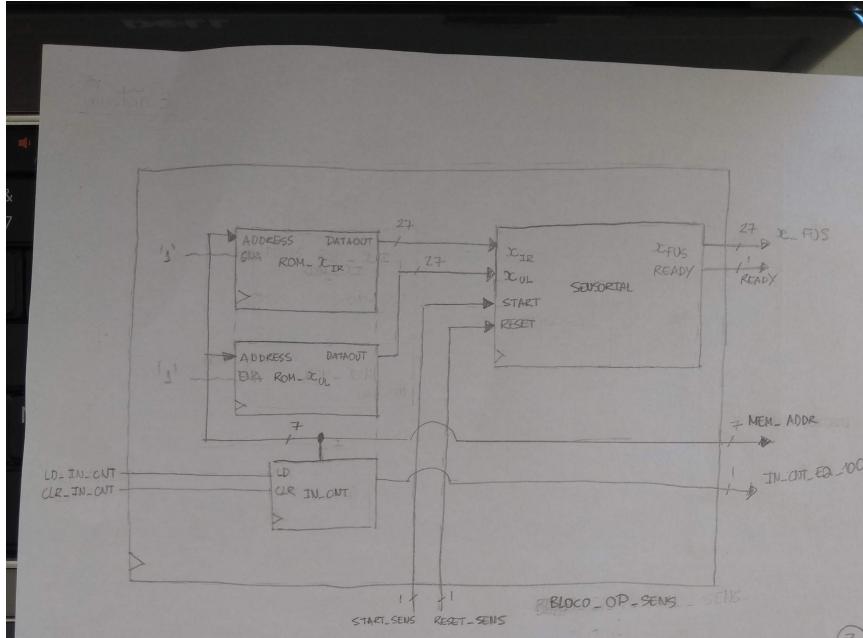


Figura 7: Formato do bloco operacional utilizando duas memórias ROM implementadas pelo IP Generator do Vivado

As entradas do bloco operacional da Figura 7 e suas funções são explicadas abaixo:

- Entradas
 - **ld_in_cnt** : Entrada que habilita o contador para incrementar o valor de sua saída. Quando esta entrada é igual a 1, o valor do contador é incrementado na próxima borda de subida de block.
 - **clr_in_cnt** : Entrada que reinicia a contagem do contador, voltado para 0. Essa entrada é assíncrona.
 - **start_sens** : Entrada que aciona o bloco **sensorial** permitindo com que esta realize os cálculos.
 - **reset_sens** : Entrada que reinicia o bloco **sensorial**, reiniciando seus registradores e levando sua saída para zero.
- Saídas
 - **x_fus** : saída com o valor de distância calculado no instante k.
 - **mem_addr** : valor atual do endereço de memória que está sendo calculado.
 - **ready** : flag que alerta que um novo valor de **x_fus** foi gerado na saída.

Para este bloco operacional, uma máquina de estados de alto nível foi gerada de forma que pudesse controlar a inserção de dados de forma síncrona. Esta pode ser visualizada na Figura 8.

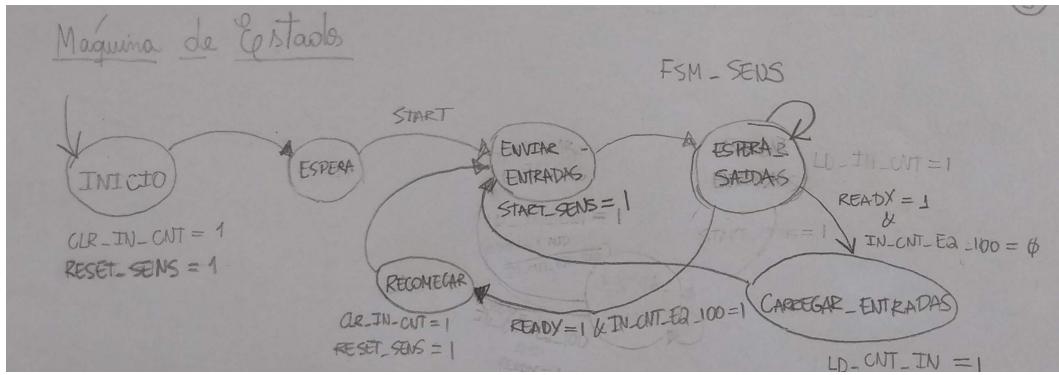


Figura 8: Máquina de Estados de Alto Nível

O top level, chamado de `sens_top_level` pode ser visualizado na Figura 9.

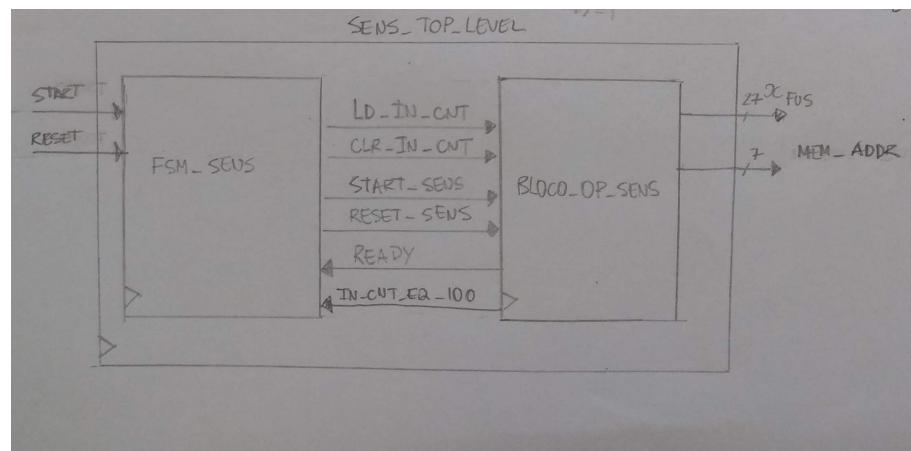


Figura 9: Top-level gerado

Um testbench foi criado para este top-level, de forma a realizar a validação.

A latência foi de 200 ns para o clock de 10 ns, ou 20 ciclos de clock, como pode ser visto na Figura 10.

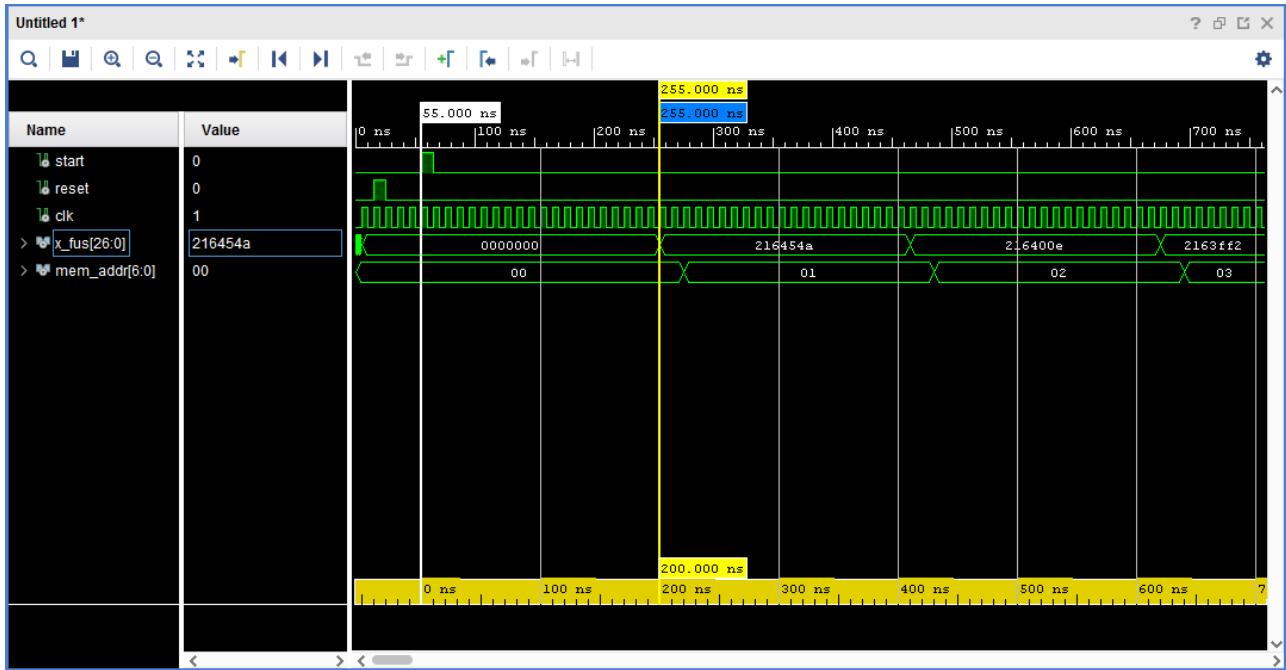


Figura 10: Latência para o top-level *sens_top_level*

Já o throughput foi de 210 ns, ou 21 ciclos de clock, como pode ser visualizado na Figura 11.

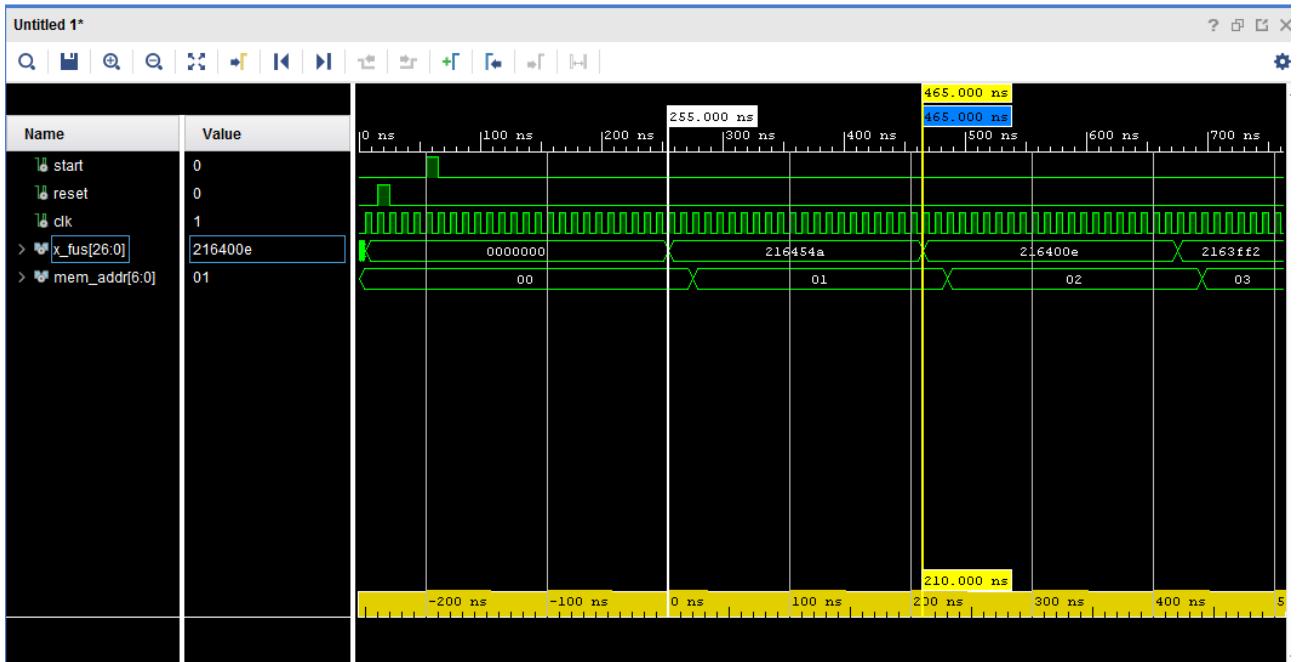


Figura 11: Throughput para o top-level *sens_top_level*

Funcionamento do projeto na FPGA

Para que fosse possível visualizar o projeto na FPGA, foi implementado o bloco da Figura 12.

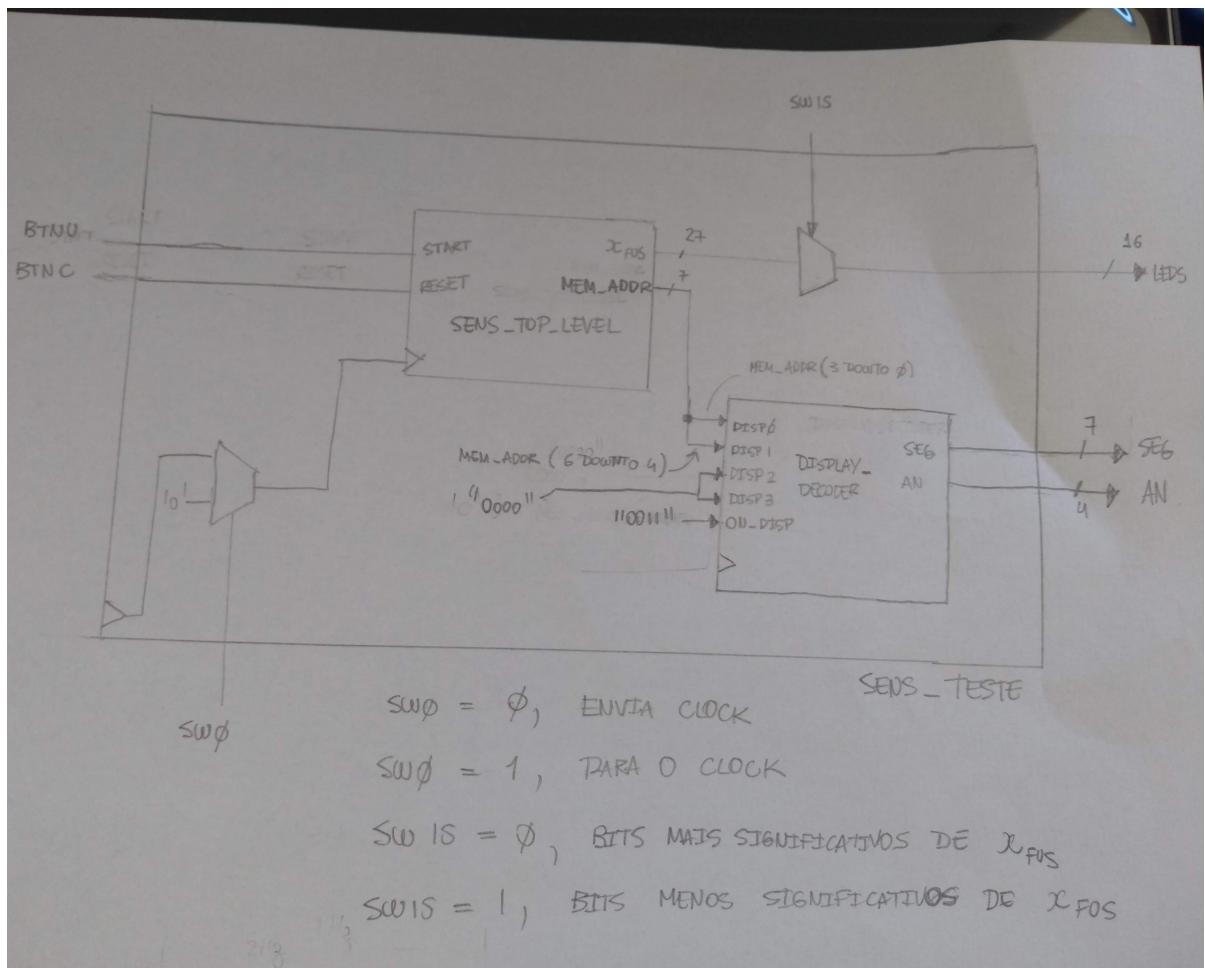


Figura 12: Bloco test_sens para que os dados calculados pudessem ser visualizados na FPGA

O bloco **sens_top_level** foi conectado ao bloco **display_decoder**, que nada mais é que um decodificador e multiplexador para os displays de 7 segmentos da Basys 3. Isto foi realizado para que o valor do endereço de memória atual que está tendo o valor calculado pelo **sens_top_level** fosse mostrado nos dois primeiros displays de 7 segmentos e pudesse se comparar se a saída gerada nos leds condizia com os valores obtidos em simulação.

As entradas têm a seguinte função:

- Botão **btnU** da Basys 3 : Fornece o start ao sistema, onde os cálculos começarão a ser realizados varrendo todas as posições das memórias ROM e apresentando os resultados na saída, reiniciando os cálculos, após as 100 posições terem sido percorridas.
- Botão **btnC** da Basys 3: Fornece o reset ao sistema, de forma que este reiniciará todos os valores e aguardará o botão **btnU** ser pressionado
- Chave **SW0** : Para de fornecer clock ao bloco **sens_top_level** e então o valor da posição de memória e o valor de saída nos leds pode ser verificado. Quando $SW0 = 0$, clock é fornecido, quando $SW0 = 1$, para-se o fornecimento de clock.
- Chave **SW15** : Alterna entre os bits mais e menos significativos da saída x_{FUS} . Quando $SW15 = 0$, são enviados os bits mais significativos para os leds, e quando $SW15 = 1$, os bits menos significativos.

O resultado, após a geração do *bitstream* e o envio para a FPGA, pode ser visualizado nas Figuras 13 e 14. Vê-se que quando o endereço lido é o 0x03, a saída nos leds é 01000010110001111111110010, que é idêntica a linha 3 do arquivo **x_bin_fus.txt**, validando portanto a solução.

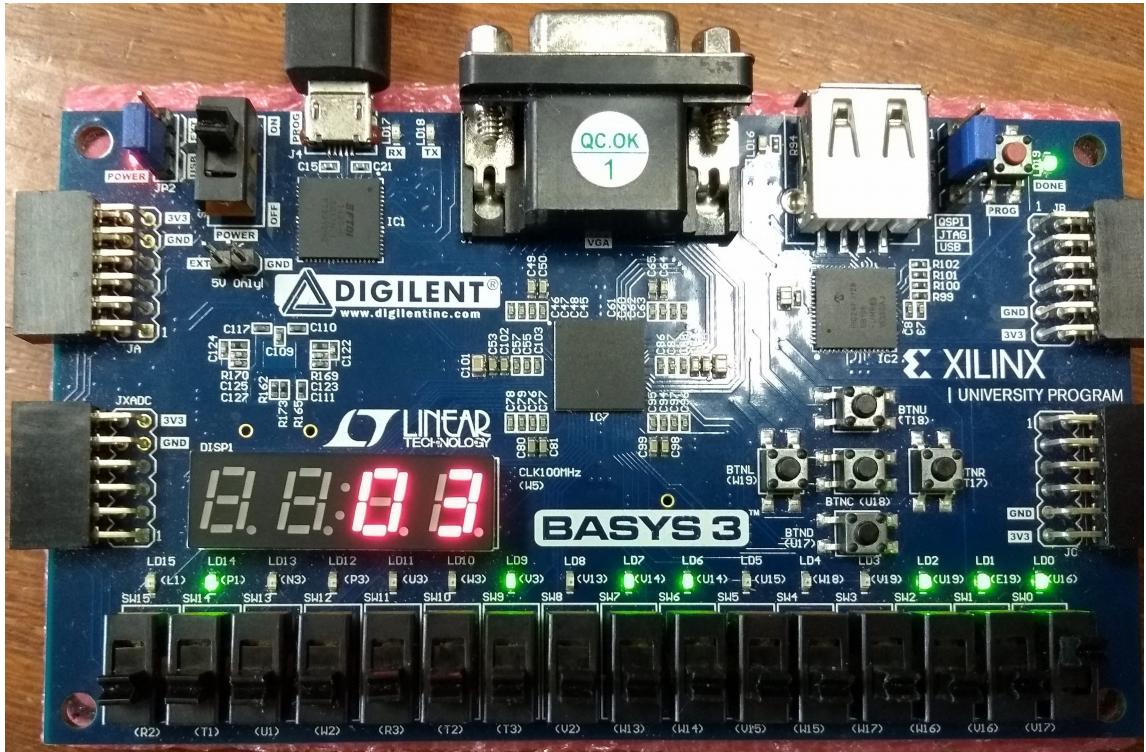


Figura 13: Bits mais significativos da saída gerada pelo cálculos dos valores enviados pelo endereço de número 0x03 das memórias ROM

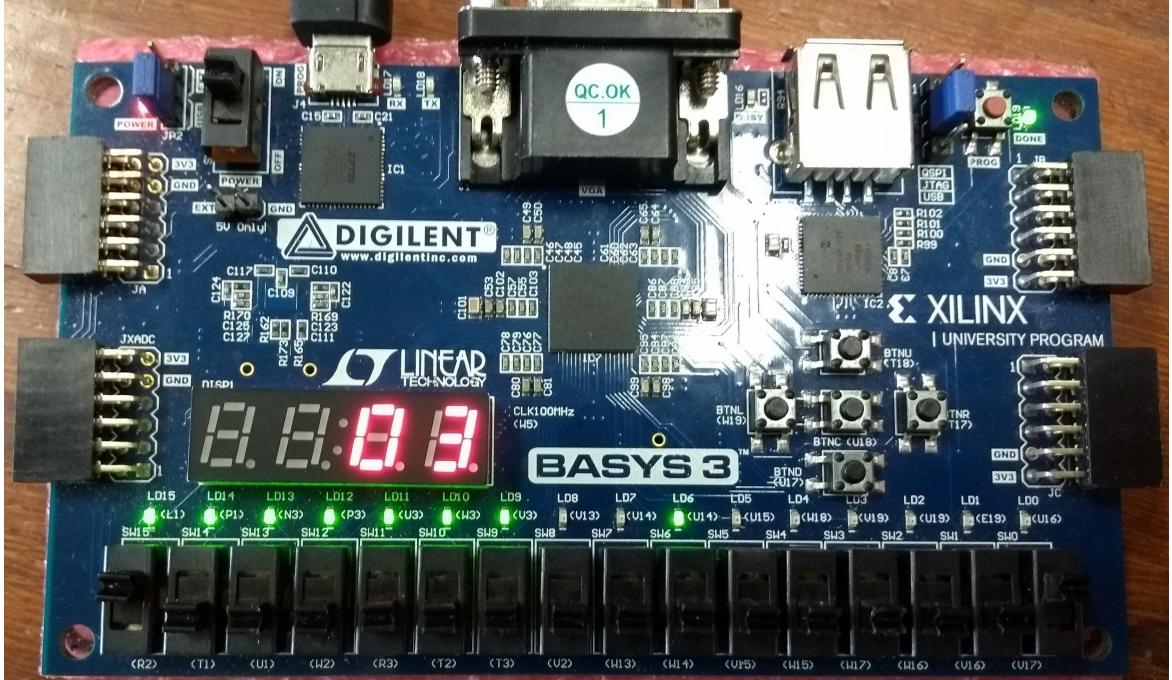


Figura 14: Bits menos significativos da saída gerada pelo cálculos dos valores enviados pelo endereço de número 0x03 das memórias ROM

A Figura 15 apresenta o report de timing, sendo a frequência máxima de operação do circuito 100 MHz.

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5,366 ns	Worst Hold Slack (WHS): 0,117 ns	Worst Pulse Width Slack (WPWS): 4,500 ns
Total Negative Slack (TNS): 0,000 ns	Total Hold Slack (THS): 0,000 ns	Total Pulse Width Negative Slack (TPWNS): 0,000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 129	Total Number of Endpoints: 129	Total Number of Endpoints: 67

All user specified timing constraints are met.

Figura 15: Report de Timing

Questão 5 – Recursos de Harware Utilizados

A Tabela da Figura 16 abaixo mostra o consumo de recursos de hardware pelo projeto após a implementação. Vê-se que apesar da complexidade, uma porcentagem muito pequena dos recursos de hardware foi utilizado.

Resource	Utilization	Available	Utilization %
LUT	1684	20800	8.10
FF	482	41600	1.16
BRAM	1	50	2.00
DSP	3	90	3.33
IO	32	106	30.19

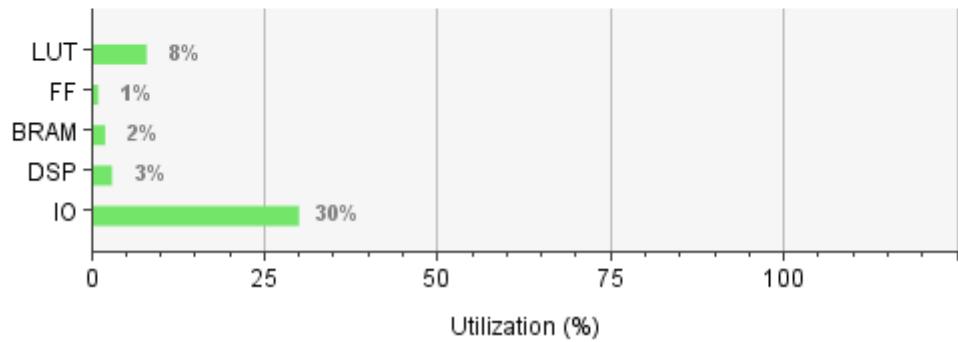


Figura 16: Tabela de utilização dos recursos de hardware após a implementação

Questão 6 – Esquemático Final, Layout do Circuito e Consumo de Potência

O esquemático final pode ser visualizado na Figura 17. Já o layout final gerado pode ser visualizado na Figura 18. O consumo total de potência pode ser visualizado na Figura 19.

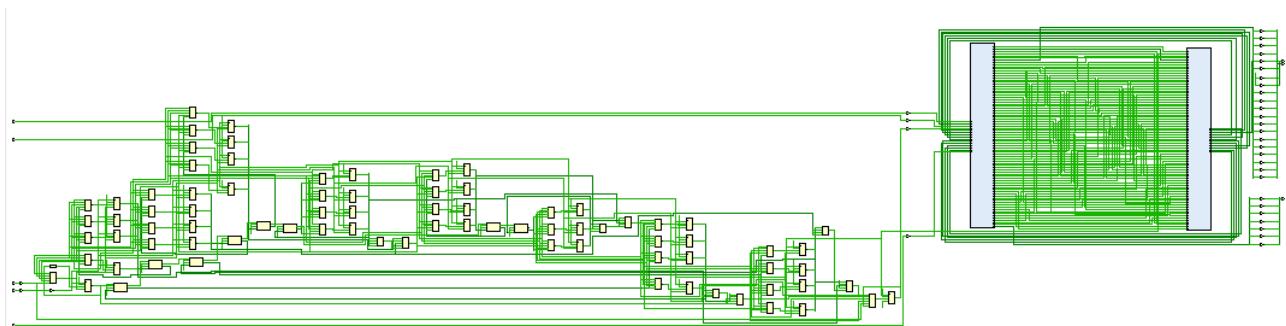


Figura 17: Esquemático gerado após a implementação

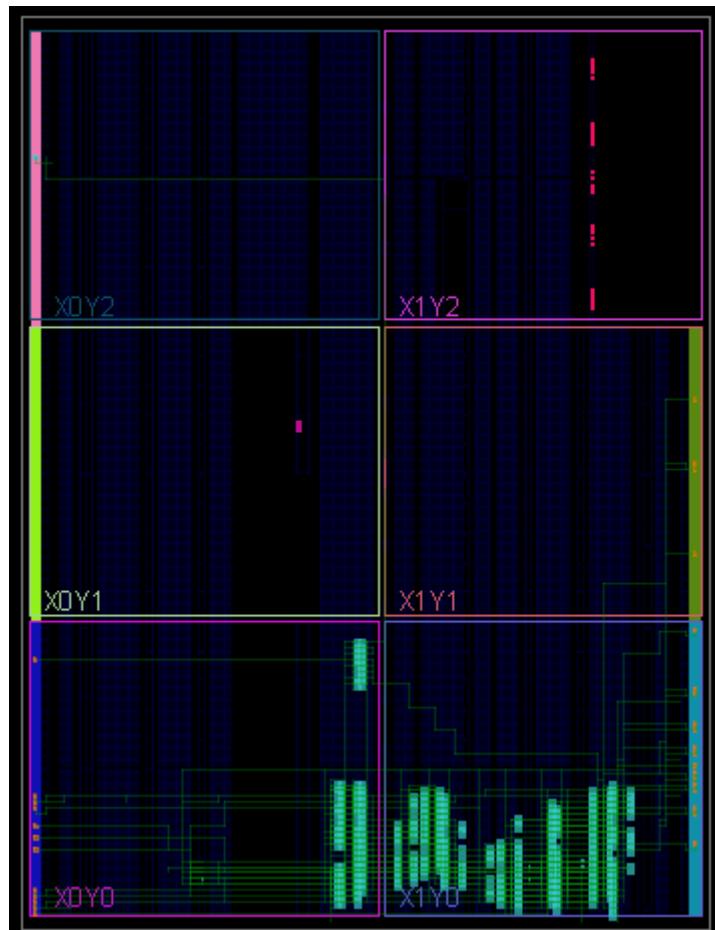


Figura 18: Layout final do circuito

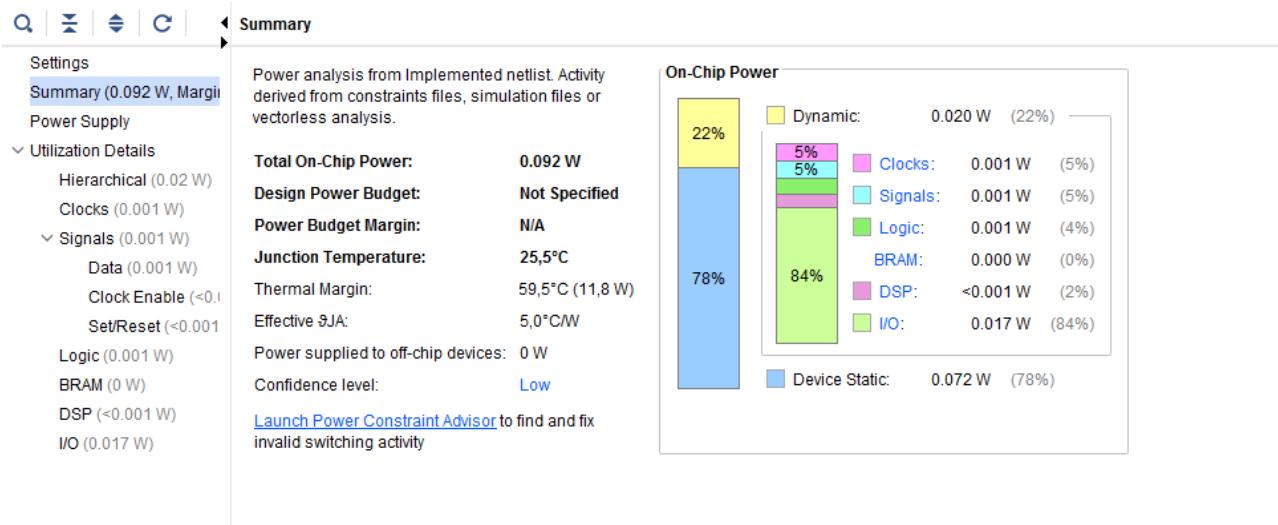


Figura 19: Consumo de potência do circuito