



**INSTITUTO
FEDERAL**

Santa Catarina

Câmpus
São José

Contador BCD

Eletrônica Digital II

Bernardo Souza Muniz

09 de Maio de 2025

Engenharia de Telecomunicações - IFSC-SJ

Sumário

1. Introdução	3
2. Código BDC	3
2.1. Parte sequencial - State Register	3
2.2. Parte combinacional - Next State Logic	3
3. Desempenho do Código	4
3.1. Fmax	5
3.1.1. Fmax com restrição de frequência máxima de Clock	5
3.1.2. Fmax com Seed	6
4. Resultados	7
4.1. Tabela de resultados	8
5. Conclusão	8

1. Introdução

O objetivo deste documento é apresentar um código em **VHDL** (*VHSIC Hardware Description Language*) que realiza a contagem binária de 000 até 999, estruturado em dois segmentos: um de lógica sequencial e outro de lógica combinacional. Além do desenvolvimento do código, são apresentados dados de verificação de desempenho, como a frequência máxima (Fmax), número de pinos utilizados e quantidade de elementos lógicos consumidos. A plataforma de desenvolvimento utilizada foi o Quartus Prime, e a simulação foi realizada com o ModelSim.

2. Código BDC

O código de contagem é dividido em dois blocos principais: um responsável pela lógica sequencial (State Register) e outro pela lógica combinacional (Next State Logic). Ele realiza a contagem binária de 000 até 999, separando os dígitos em unidade, dezena e centena. O objetivo final é que todas as casas numéricas atinjam o valor “1001” ao término da contagem.

2.1. Parte sequencial - State Register

O primeiro segmento do código se trata do State register, que é sensível ao clock e ao reset. O objetivo do State Register é resetar todas as unidades de contagem para “000” quando a variável **enable** for igual a “1”. Caso não estiver ativo, o processo só executa durante uma borda de subida (rising edge) do clock. Durante o clock os valores do próximo estado são armazenados nos registradores.

```
1 process(clock, reset)
2 begin
3     if (reset = '1') then
4         reg_unidade <= (others => '0');
5         reg_dezena   <= (others => '0');
6         reg_centena  <= (others => '0');
7
8     elsif rising_edge(clock) then
9         reg_centena <= centena_next;
10        reg_dezena  <= dezena_next;
11        reg_unidade <= unidade_next;
12    end if;
13 end process;
```

2.2. Parte combinacional - Next State Logic

O segundo segmento se baseia no cálculo do próximo estado. Dentro desse processo, as variáveis de próximo estado são atualizadas cada vez que as variáveis de registro são igual a “1001” — 9 em binário.

```
1 process(reg_unidade, reg_dezena, reg_centena, enable)
2 begin
3     -- para evitar o latch quando enable for diferente de 1,
```

```

4      -- nesse caso os valores permanecem os mesmos e nao mudam
5      unidade_next <= reg_unidade;
6      dezena_next  <= reg_dezena;
7      centena_next <= reg_centena;
8
9      if enable = '1' then
10         if reg_unidade = "1001" then -- 9 em binário
11             unidade_next <= (others => '0');
12
13             if reg_dezena = "1001" then
14                 dezena_next <= (others => '0');
15
16                 if reg_centena = "1001" then
17                     centena_next <= (others => '0');
18                 else
19                     centena_next <= reg_centena + 1;
20                 end if;
21             else
22                 dezena_next <= reg_dezena + 1;
23             end if;
24         else
25             unidade_next <= reg_unidade + 1;
26         end if;
27     end if;
28 end process;

```

3. Desempenho do Código

Uma vez feito a descrição do código em VHDL, foi utilizado a família de FPGA **CYCLONE IV** e o dispositivo **EP4CE6E22A7**. Foram verificados os valores de tensão de alimentação do núcleo e o número de elementos lógicos.

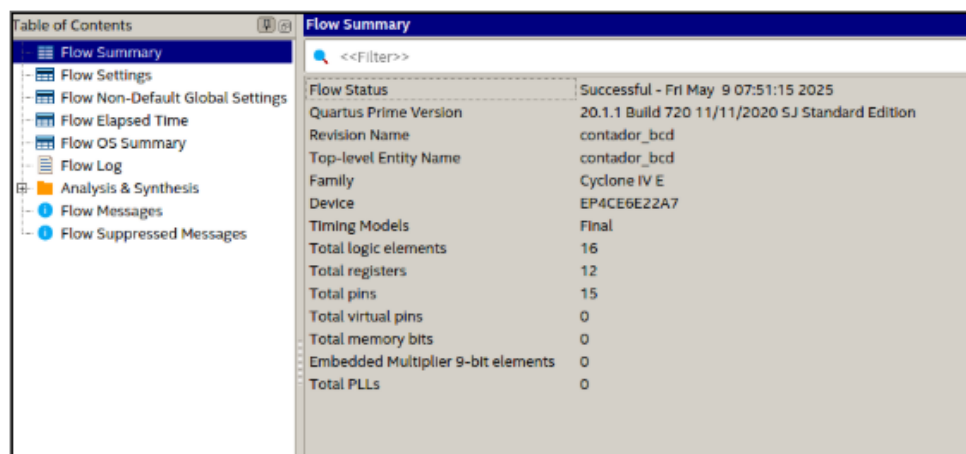
Figura 1: Elaborada pelo Autor



Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elements	PLLs	Global Clocks
EP4CE6E22A7	1.2V	6272	92	92	276480	30	2	10

Core voltage

Figura 2: Elaborada pelo Autor



Flow Summary	
Flow Status	Successful - Fri May 9 07:51:15 2025
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Standard Edition
Revision Name	contador_bcd
Top-level Entity Name	contador_bcd
Family	Cyclone IV E
Device	EP4CE6E22A7
Timing Models	Final
Total logic elements	16
Total registers	12
Total pins	15
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

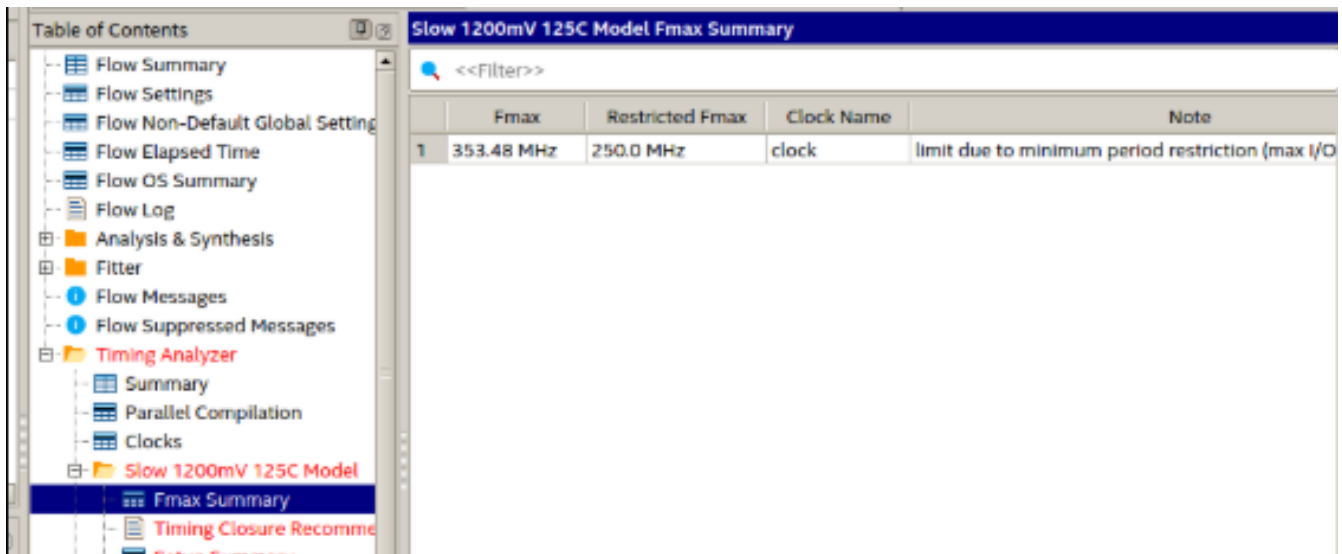
Número de elementos lógicos

Após a compilação do projeto, foi verificado um total de 16 elementos lógicos e uma tensão de 1.2 V.

3.1. Fmax

Para a obtenção dos valores da Fmax do projeto, foram utilizados dados presentes no **Compilation Report** na pasta **Timing Analyzer**. Os resultados obtidos estão presentes na figura 3.

Figura 3: Elaborada pelo Autor



	Fmax	Restricted Fmax	Clock Name	Note
1	353.48 MHz	250.0 MHz	clock	limit due to minimum period restriction (max I/O

Fmax na primeira compilação

Ao fazer a primeira compilação do projeto sem demais alterações para otimização, foi obtido um valor Fmax de 353.4 Mhz.

3.1.1. Fmax com restrição de frequência máxima de Clock

No intuito de diminuir a frequência do Fmax, foi utilizado um arquivo **.sdc** contendo um comando para a configuração de restrição do Clock do projeto.

```
1 create_clock -name CLOCK -period 20 [get_ports -no_case {clock}]
```

Após a execução do projeto utilizando o código com restrições no arquivo **.sdc**, o valor de Fmax foi reduzido para um valor de 259.47 MHz e os mesmos valores de pinos e elementos lógicos.

Figura 4: Elaborada pelo Autor

	Fmax	Restricted Fmax	Clock Name	Note
1	259.47 MHz	250.0 MHz	CLOCK	limit due to minimum period restriction (max I/O)

Valor do Fmax após a utilização da restrição de frequência do Clock

3.1.2. Fmax com Seed

Foi feito também a utilização de uma Seed para a configuração do compilador com o intuito de observar os valores do Fmax após a alteração.

Fonte: Elaborado pelo autor

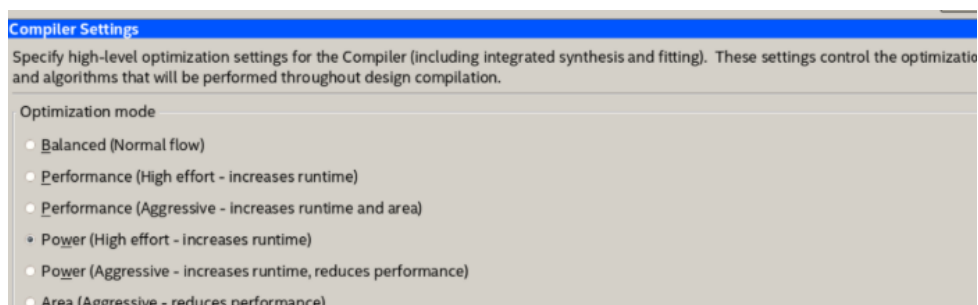


Figura 5: Optimization mode

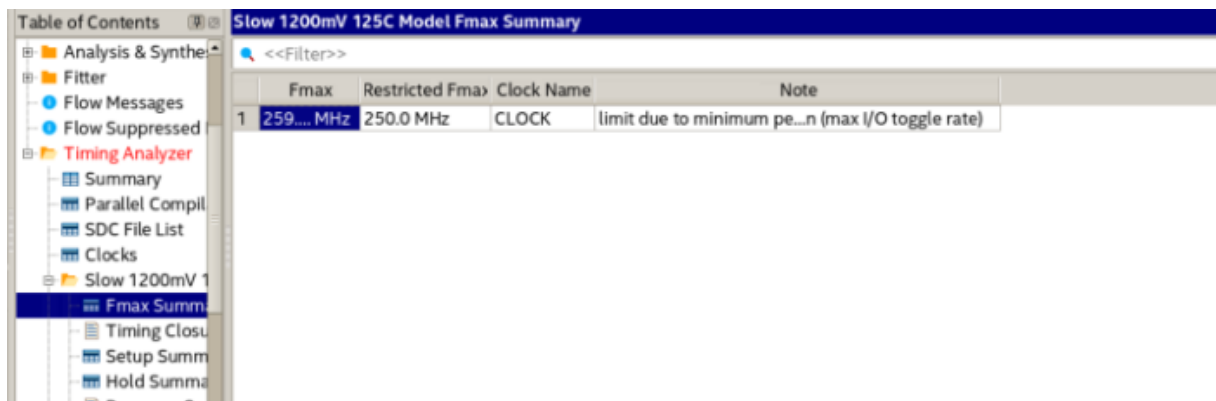
Foi selecionado as opções de “Power (High effort - increases runtime)” e configurado uma Seed com valor 1.

Fonte: Elaborado pelo autor

Name:	Setting:
Enable Beneficial Skew Optimization	On
Enable Bus-Hold Circuitry	Off
Final Placement Optimizations	Automatic...
Fit Attempts to Skip	0
Fitter Aggressive Routability Optimizations	Always
Fitter Effort	Auto Fit
Fitter Initial Placement Seed	1
Force Fitter to Avoid Periphery Placement Warnings	Off
I/O Placement Optimizations	On
Limit to One Fitting Attempt	Off
Logic Cell Insertion - Logic Duplication	Auto
Maximum number of global clocks allowed	-1 (UNLIM...
Optimize Design for Metastability	On

Figura 6: Seed

Fonte: Elaborado pelo autor



	Fmax	Restricted Fmax	Clock Name	Note
1	259... MHz	250.0 MHz	CLOCK	limit due to minimum pe...n (max I/O toggle rate)

Figura 7: Fmax com Seed

Ao compilar novamente o projeto, foi obtido um tempo de Fmax igual a 259.47 MHz e os mesmos valores de pinos e elementos lógicos.

4. Resultados

Após a compilação do projeto, foi efetuado o teste do código no Modelsim para a visualização das unidades do contador.

Fonte: Elaborado pelo autor

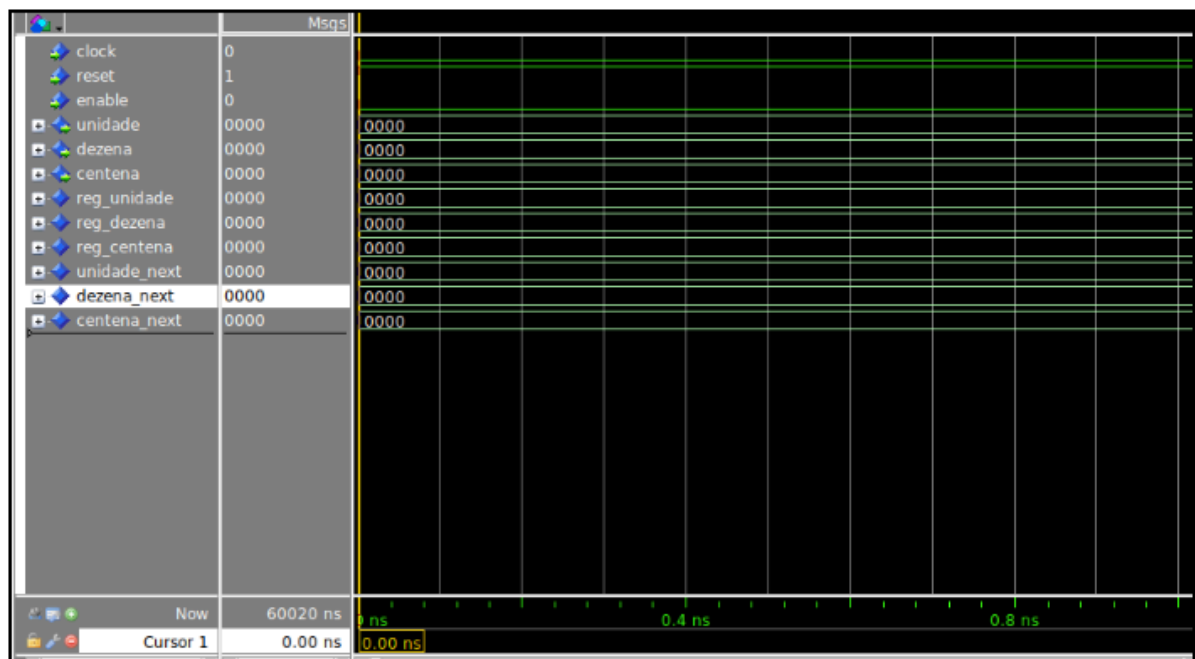


Figura 8: Início do contador

Após definir uma variação de clock e definir o reset com valor 1, todas as variáveis ficaram zeradas.

Fonte: Elaborado pelo autor

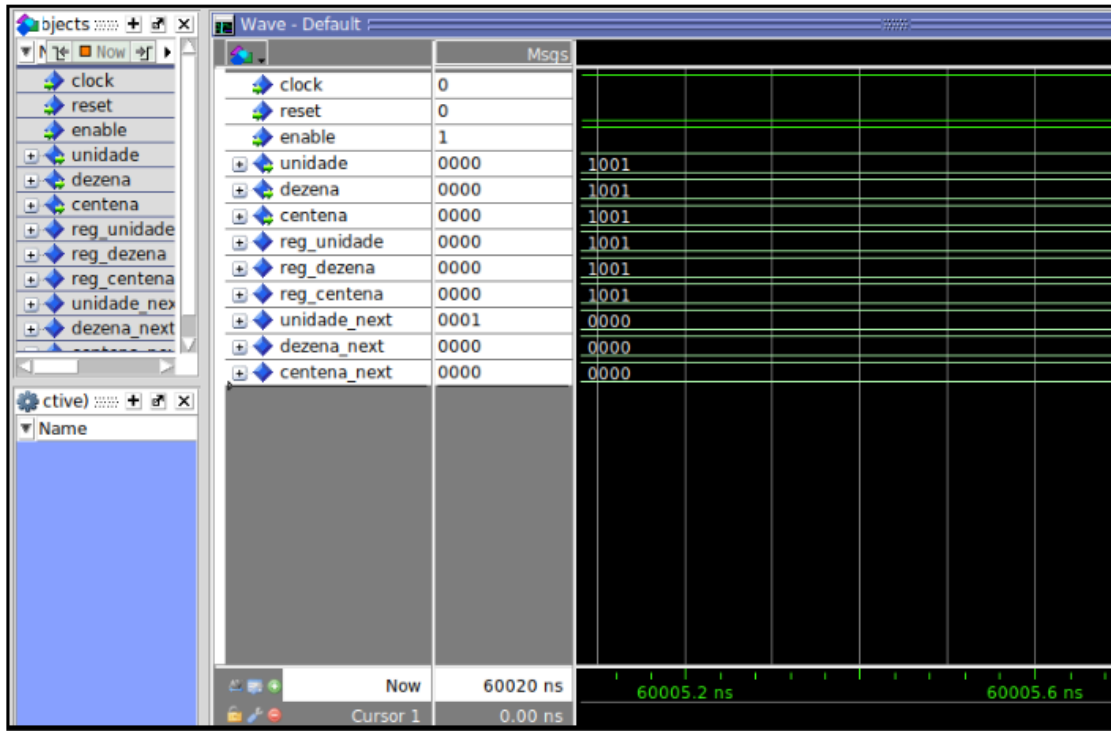


Figura 9: Início do contador

Agora com o enable e a variação de clock ativa, após avançar no tempo verifica-se que todas as variáveis de contagem ficaram com valor “1001”, como era esperado.

4.1. Tabela de resultados

A tabela a seguir mostra todos os resultados obtidos durante as 3 compilações feitas, incluindo o número de elementos lógicos, pinos e Fmax obtida.

Tabela 1: Elaborada pelo Autor

Compilação	Fmax (MHz)	Elementos lógicos	Pinos
Primeira compilação	353,48	16	15
Restrição de Clock	259,47	16	15
Compilação com Seed	259,47	16	15

Tabela de resultados de compilação

5. Conclusão

Conclui-se que a utilização de dois segmentos no desenvolvimento do código proporcionou resultados significativamente mais consistentes em comparação à abordagem com apenas um segmento. Os objetivos definidos no início do projeto foram plenamente alcançados, culminando na implementação bem-sucedida de um contador binário que vai de 000 até 999.