



**INSTITUTO  
FEDERAL**

Santa Catarina

---

Câmpus  
São José

## **Relatório Atividade extra-classe 5**

### **Relógio HH-MM-SS**

Dispositivos lógicos programáveis

**Rhenzo Hideki Silva Kajikawa**

26 de Setembro de 2023

# Sumário

<b>1. Introdução .....</b>	<b>3</b>
1.1. Objetivos .....	3
1.2. Motivação .....	3
1.3. Procedimentos .....	3
<b>2. Descrição do Projeto .....</b>	<b>4</b>
2.1. Componentes utilizados .....	4
2.2. Sistema Completo .....	5
2.2.1. Pinagem .....	5
2.2.2. Número de elementos lógicos .....	5
<b>3. Resultados obtidos .....</b>	<b>8</b>
3.1. RTL viewer .....	8
3.2. Implementação na placa .....	10
<b>4. Conclusão .....</b>	<b>10</b>

# **1. Introdução**

## **1.1. Objetivos**

Este projeto feito em aula do relógio HHMMSS tem como objetivo indicar as horas (HH) , minutos (MM) e segundos (SS) , mostrando as unidades em 3 pares de displays de 7 segmentos, estes que estão são encontrados no kit DE2-115 da TERASIC , encontrado no laboratório de sinais digitais.

## **1.2. Motivação**

Em aula foram ensinadas vários conceitos de VHDL e boas maneiras para um projeto. Porém até o momento não havia sido aplicado o conjunto dessas ideias em um projeto. Dessa forma este projeto vem para amarrar essas ideias em aplica-las em aula.

## **1.3. Procedimentos**

Em aula foram feitas 4 entidades. Essas entidades foram separadas da seguinte forma , divisor de clock que foi chamado `div_clk` , um contador BCD (Binary-coded decimal) que foi chamado de `contador_bcd` , um conversor BCD para SSD (Seven-segment display) que foi chamado de `bcd2ssd` , e por fim a entidade que integra todo o conjunto o relógio , chamado de `relógio_HH-MMSS`. Com essas entidades feitas é possível junta-los em um arquivo para assim obter-se o projeto final como um todo.

## 2. Descrição do Projeto

Como falado anteriormente , O projeto foi separado em seções menores , atacando o projeto em partes menores para uma melhor manutenção do código.

O primeiro levantamento feito em aula é representado pela Figura 1 , que foi feita em aula. Nela é possível observar as pequenas partes que compõem o todo. Se bem observado tem para cada display de sete segmentos um conversor BCD para SSD , estes que recebem de o valores de um dos 3 contadores , e por fim temos um o clock de 50MHz conectados em todos os contadores e um divisor de clock que habilita o contador de segundos.

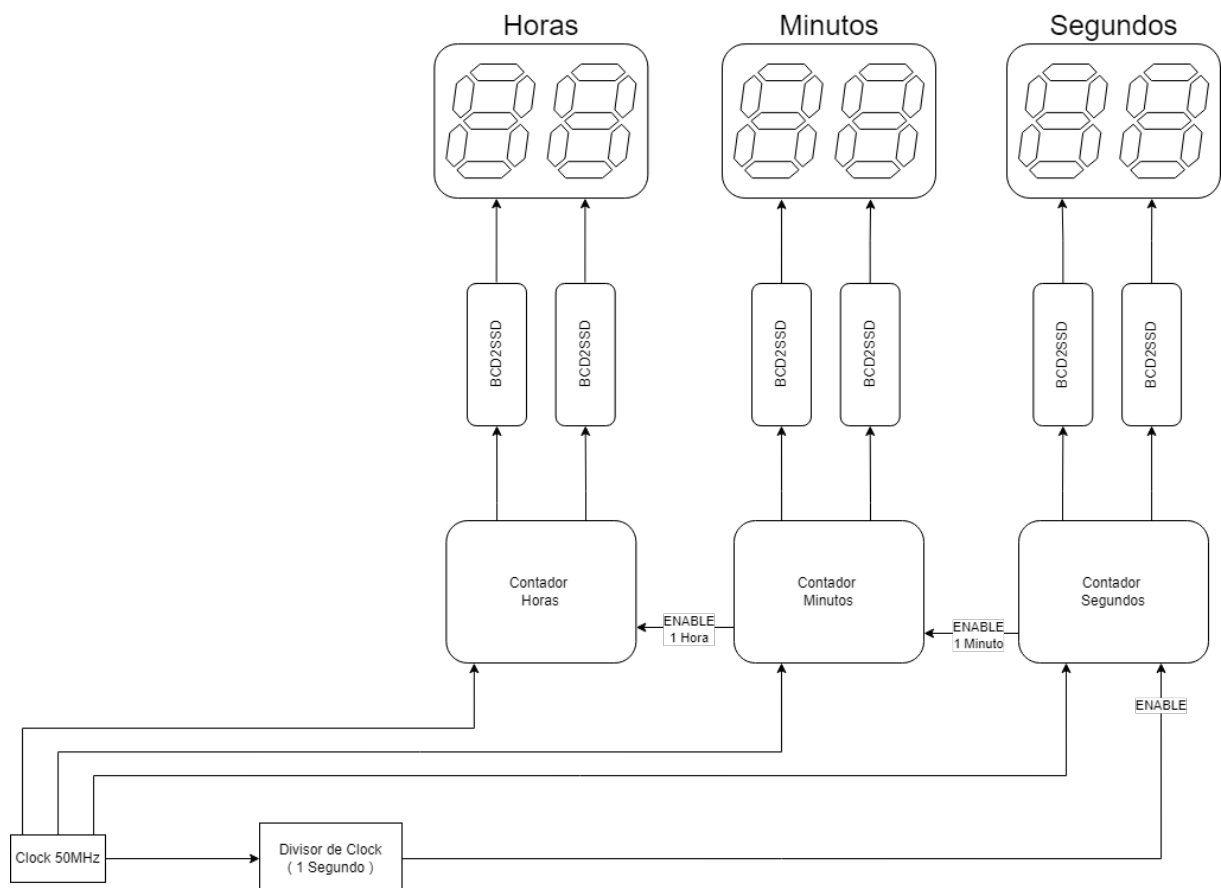


Figura AE5 1: Elaboração do Projeto

Fonte: Elaborada pelo autor

### 2.1. Componentes utilizados

O componentes utilizados para o projeto podem ser visualizados na Figura 1.

No total foram Utilizados 6 displays de sete segmentos , 6 conversores de BCD para SSD , 3 contadores , 1 divisor de clock e 1 clock de 50MHz. Contabilizando o total de 17 componentes para cumprir com os objetivos desse projeto. Como dito anteriormente , nesse projeto foi abordado como criar componentes a partir de uma entidade .vhdl , dessa forma foram criados 3 componentes menores.

Um divisor de clock com 2 entradas , reset e clock in , e uma saída clock out . Esse divisor tem como objetivo diminuir os pulsos de enable dos segundos , podendo ajustar toda

contagem do sistema para diferentes clocks . Apenas trocando o valor genérico do componente “div”.

Para os contadores foram criados com 2 entradas , clock e reset , e três saídas bcd unidade , bcd dezena e um clock out. Nos contadores foi pensado inicialmente em fazer contadores que contavam de 0 a 59 e posteriormente esse dado teria de ser tratado , porém o professor nos sugeriu a construção de 2 contadores , 1 para unidade e o outro para dezena , dessa forma o componente extra foi descartado. Da mesma forma que o divisor , é possível entrar com valores para ajustar os generics dos contadores uma vez que para o projeto é necessário 2 contadores de 59 e um de 24 ou 12.

Para o conversor BCD para SSD , foi a tarefa mais simples dos 3 componentes menores , uma vez que o componente é apenas uma tabela de conversão , foi utilizado apenas um when case ou with select.

Uma vez que o sistema tinha todos seu componentes menores feito o trabalho maior foi de ligar todos os componentes criados dentro do clock.vhd.

## 2.2. Sistema Completo

Após o sistema ter todos seu componentes montas foi feita a pinagem e compilação completa.

### 2.2.1. Pinagem

A pinagem

### 2.2.2. Número de elementos lógicos

O numero de elementos lógicos por componente:

Flow Status	Successful - Fri Nov 24 22:37:36 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Standard Edition
Revision Name	relogio
Top-level Entity Name	div_clk
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	11
Total registers	7
Total pins	3
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Figura AE5 2: Compilação do divisor de clock

Fonte: Elaborada pelo autor

Pode ser observado o uso de 11 elementos lógicos no divisor de clock

Flow Status	Successful - Fri Nov 24 22:30:45 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Standard Edition
Revision Name	relogio
Top-level Entity Name	bcd_counter
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	16 / 114,480 ( < 1 % )
Total registers	9
Total pins	11 / 529 ( 2 % )
Total virtual pins	0
Total memory bits	0 / 3,981,312 ( 0 % )
Embedded Multiplier 9-bit elements	0 / 532 ( 0 % )
Total PLLs	0 / 4 ( 0 % )

Figura AE5 3: Compilação do contador

Fonte: Elaborada pelo autor

Pode ser observado o uso de 16 elementos lógicos para cada par de contadores , tendo como uma projeção de se ter 3 vezes mais , aproximadamente 48 elementos para o conjunto todo. Observação o contador estava configurado para contar 24 horas.

Flow Status	Successful - Fri Nov 24 22:50:28 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Standard Edition
Revision Name	relogio
Top-level Entity Name	bcd2ssd
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	14
Total registers	0
Total pins	12
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Figura AE5 4: Compilação do conversor bcd para ssd

Fonte: Elaborada pelo autor

Para o conversor bcd para ssd foram utilizados 14 elementos , porém cada elemento trabalha de forma separada . Assim a estimativa para o total de elementos lógicos presente no projeto seria de 84 elementos lógicos para todo o grupo de conversores.

Flow Status	Successful - Fri Nov 24 22:55:48 2023
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Standard Edition
Revision Name	relógio
Top-level Entity Name	clock
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	168
Total registers	53
Total pins	46
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Figura AE5 5: Compilação do relógio (clock)

Fonte: Elaborada pelo autor

Apesar de na soma de todos os componentes terem dado 143 elementos lógicos, a compilação resultou em um total 168 de elementos . Isso pode ter ocorrido pois alguns componentes flu-  
tuavam os valores de elementos lógicos dependendo de como eram configurados.

### 3. Resultados obtidos

#### 3.1. RTL viewer

Aqui estão os RTLs viewers para cada componente:

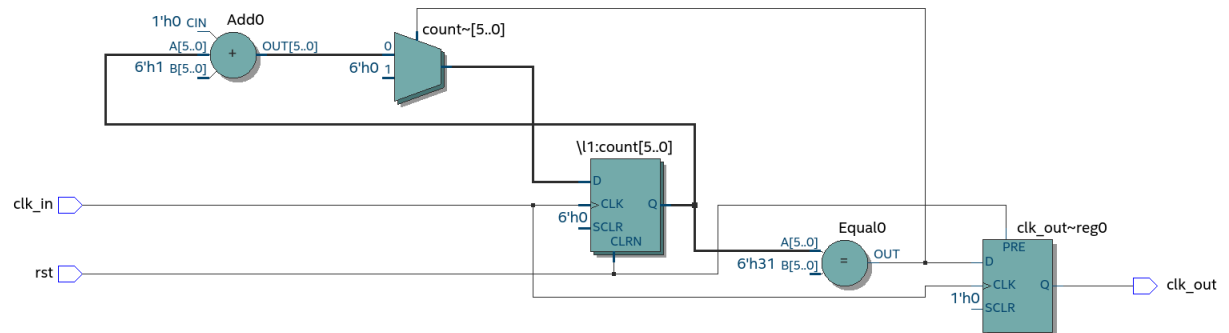


Figura AE5 6: RTL viewer do divisor de clock

Fonte: Elaborada pelo autor

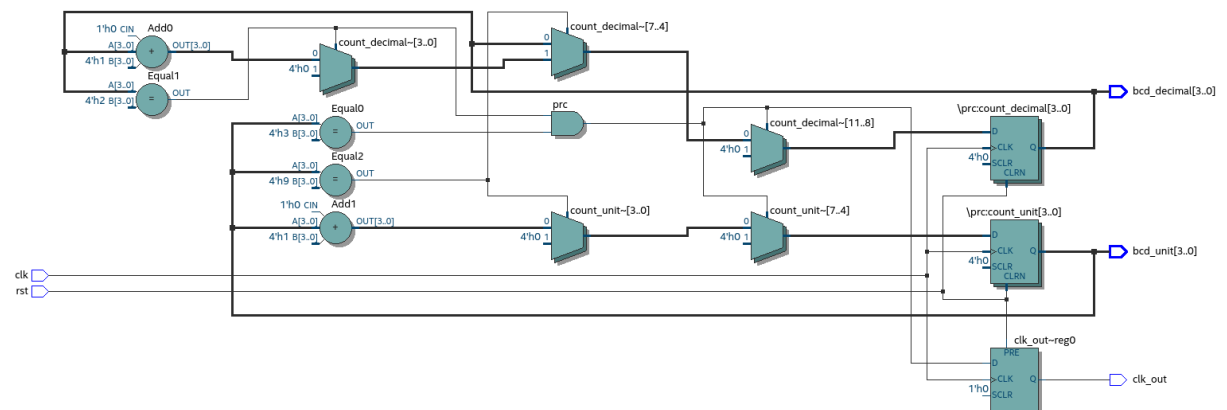


Figura AE5 7: RTL viewer do contador bcd

Fonte: Elaborada pelo autor



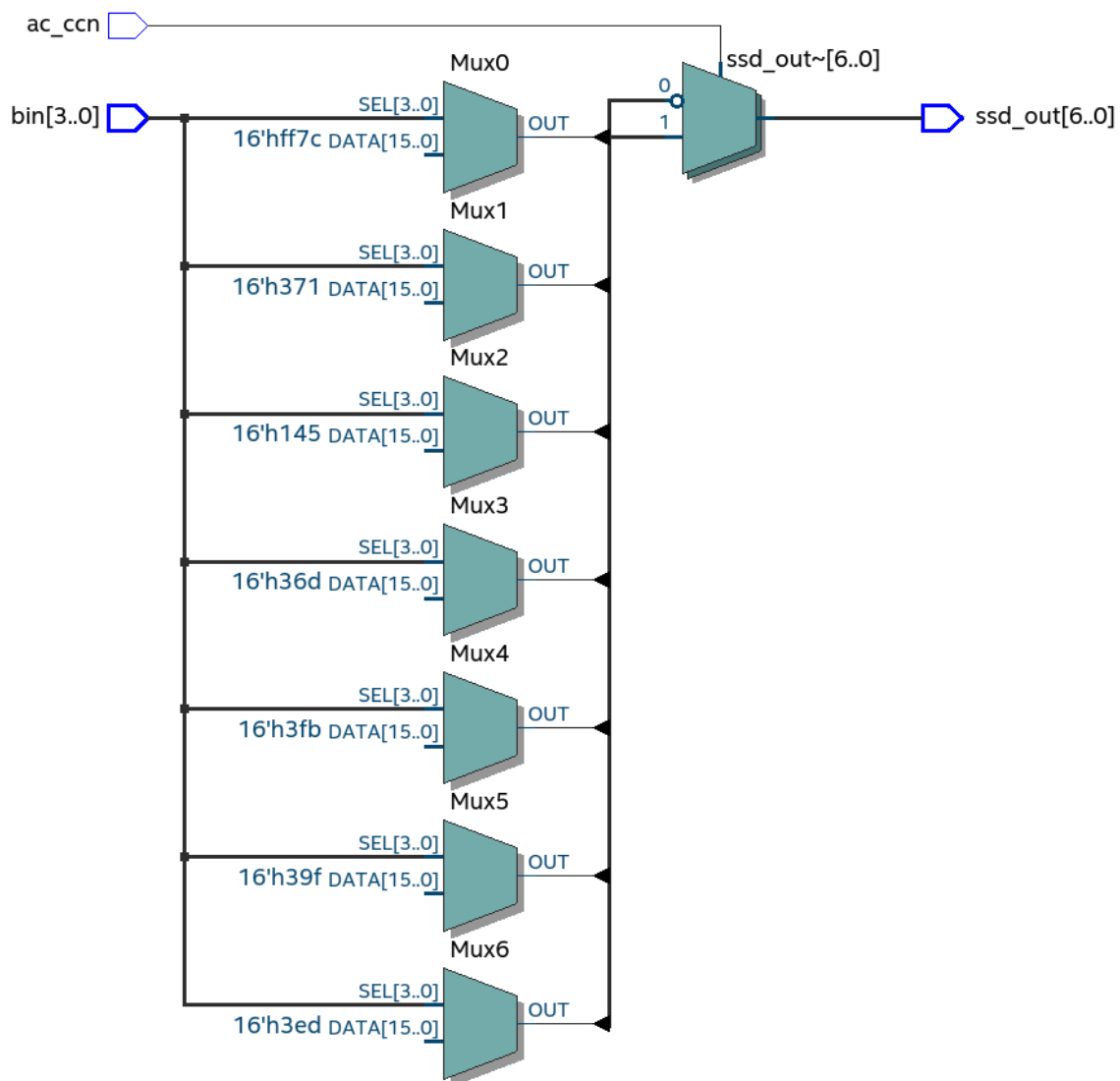


Figura AE5 8: RTL viewer do conversor bcd2ssd

Fonte: Elaborada pelo autor

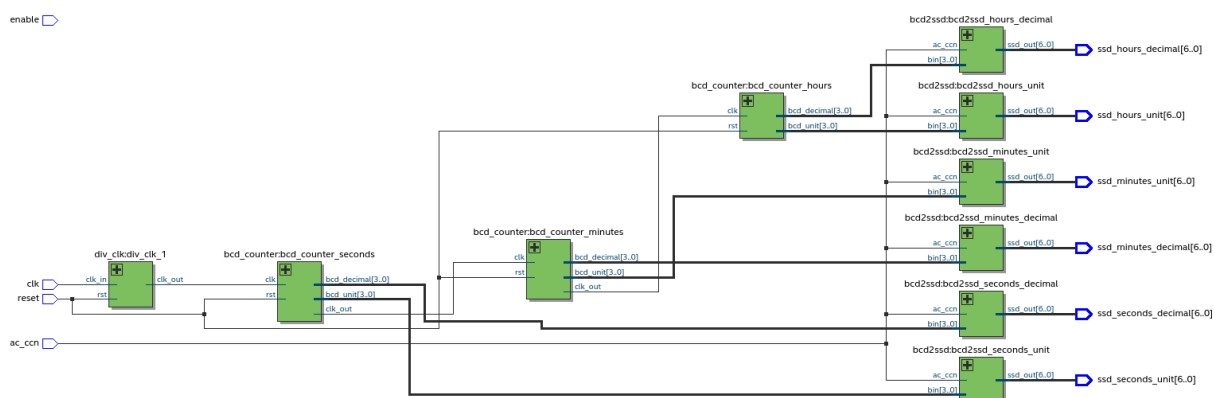


Figura AE5 9: RTL viewer do relógio (clock)

Fonte: Elaborada pelo autor

### **3.2. Implementação na placa**

A implementação na placa foi feita em aula com o kit DE2-115 da TERASIC. Foi possível observar a placa funcionando como esperado.

Na implementação podemos ver a contagem sendo feita de 1 em 1 segundo , porém para fins práticos depois aumentamos a frequência de contagem para observar o funcionamento de todos os contadores e todos displays de sete segmentos. O projeto funcionou com sucesso na placa utilizada que foi a

## **4. Conclusão**

Com esse projeto feito em aula foi possível visualizar soluções mais eficientes para lidar com alguns componentes , além de ver o funcionamento e a criação de componente menores para um projeto um pouco maior. Além de utilizar diferentes partes tanto em paralelo quanto sequencial para este projeto . Dessa maneira conseguimos implementar praticamente todo o conhecimento que foi dado em aula neste projeto