

INSTITUTO FEDERAL  
SANTA CATARINA  
Campus São José

## MINISTÉRIO DA EDUCAÇÃO

SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES

Aluno: Vinícius da Luz Souza  
Matéria: DLP II - 2019/1

Data: 09/06/2019

# Prática Circuitos Sequenciais

## Relatório A3

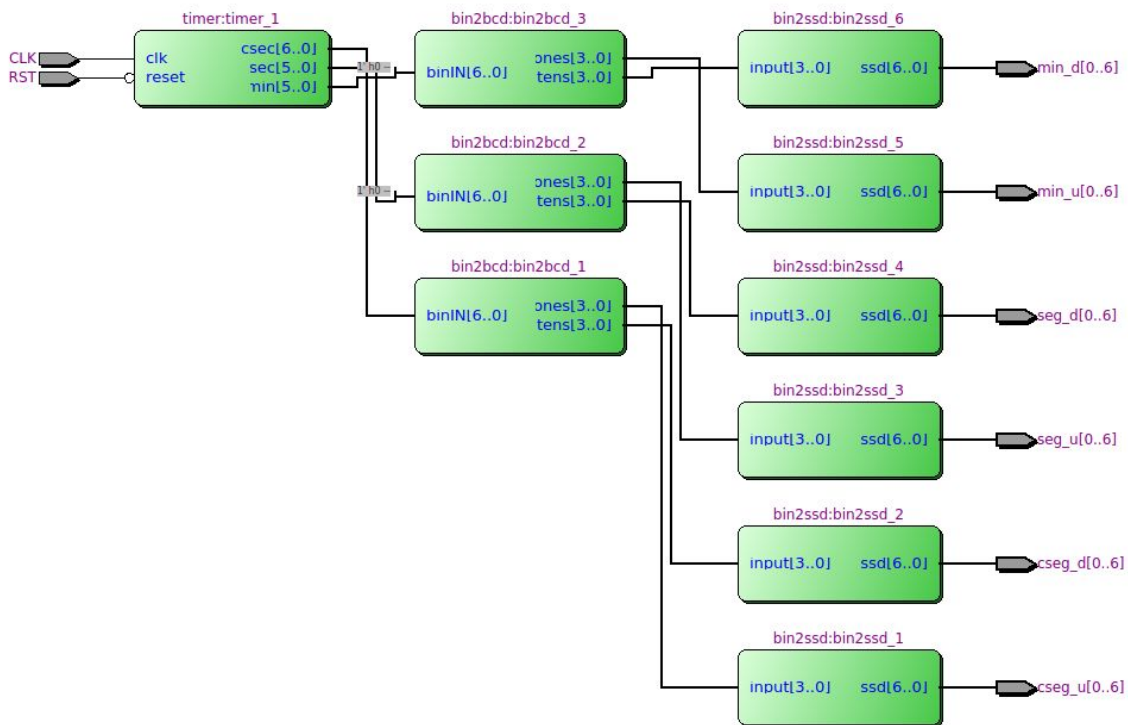
Este relatório tem por objetivo descrever e analisar diferentes implementações para um circuito de *Timer*, implementado em hardware, através da descrição VHDL. Os circuitos foram implementados utilizando lógica combinacional e sequencial, e os parâmetros que utilizaremos para comparar as diferentes implementações serão: área do circuito (número de elementos lógicos) e frequência máxima alcançada. Os testes foram realizados utilizando *software* Quartus II e kit FPGA DE2-115 da família Cyclone IV E (Altera - TERCASIC), *device* EP4CE115F29C7.

O circuito de *Timer* utilizado no experimento possui três contadores: centésimos de segundo (0 a 99), segundos (0 a 59) e minutos (0 a 59), os quais deverão ser exibidos nos displays de 7 segmentos disponíveis no kit FPGA. Ao atingir os limites máximos (59 minutos, 59 segundos e 99 centésimos), o circuito deve zerar a contagem. De outra maneira, a contagem também deverá ser zerada quando o botão de *reset* for pressionado.

Este relatório foi segmentado da seguinte forma: primeiramente apresentaremos cada arquitetura utilizada para implementação do *Timer* e, por fim, faremos um comparativo entre as implementações.

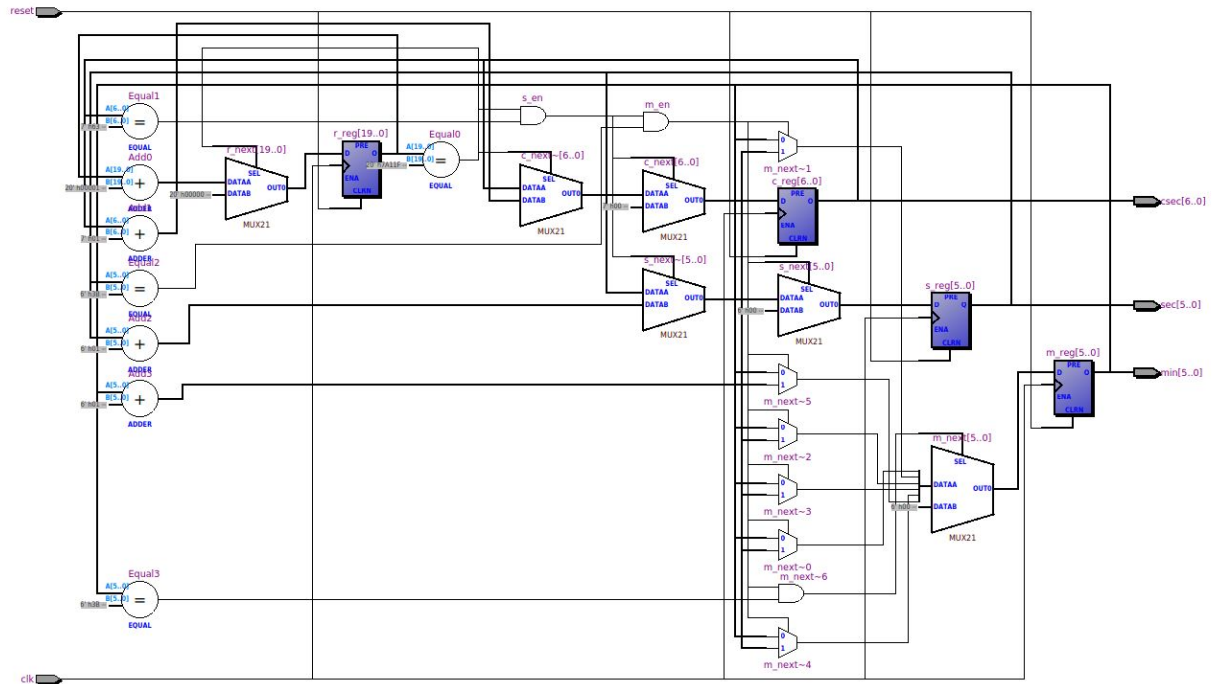
### Arquitetura 1

A primeira arquitetura utilizada no experimento possui três níveis de encadeamento. No primeiro nível, o circuito denominado *timer* é responsável pela contagem dos valores de centésimos de segundos, segundos e minutos. No segundo estágio, denominado *bin2bcd*, o objetivo é separar os valores recebidos pelo primeiro estágio em valores de uma só unidade (exemplo: separar o valor 45 em um 4 e um 5), que serão exibidos nos displays. No terceiro nível, denominado *bin2ssd*, os valores são convertidos para serem exibidos no display de 7 segmentos.



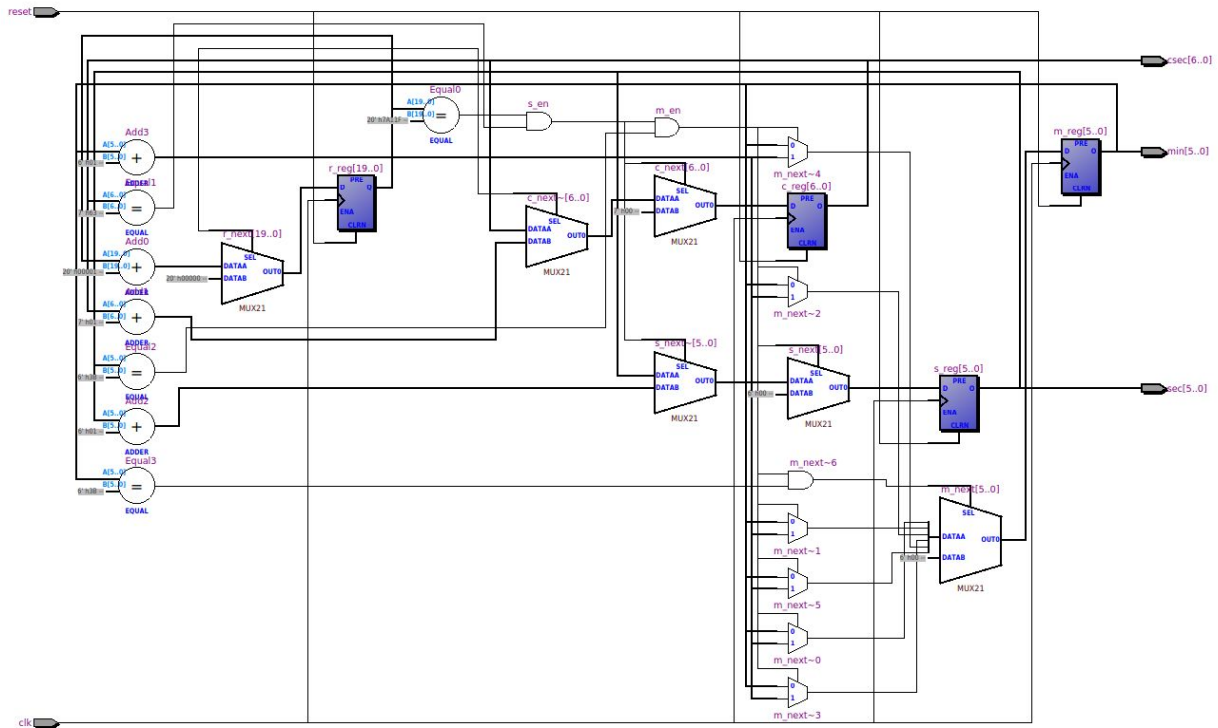
**Figura 1** - Arquitetura 1

Nesta arquitetura há duas implementações para o circuito *Timer* (nível 1). Na primeira, denominada *multi\_clock*, são gerados *clocks* a partir do *clock* de 50 MHz disponível na FPGA. Desta forma, há um *process* para cada *clock* gerado. O primeiro *process* processa o *clock* de 50 MHz e, combinacionalmente, um novo *clock* é gerado para a contagem dos centésimos de segundos. Em seguida, é utilizado um *process* para processar os centésimos de segundos a partir do *clock* gerado anteriormente e, novamente combinacionalmente, um novo *clock* é gerado para a contagem dos segundos. Isto é feito sucessivamente até a contagem dos minutos. O resultado é apresentado no diagrama RTL (*Register Transfer Level*) a seguir.



**Figura 2 - RTL timer *multi\_clock***

A segunda implementação, denominada *single\_clock*, para o mesmo circuito de *Timer* desta mesma arquitetura, utiliza somente um *process* para processar todos os blocos de contagem (centésimos de segundos, segundos e minutos). Isso faz com que todos os *FlipFlops* gerados possuam o mesmo *clock* de entrada e, desta forma, combinacionamente, os *FlipFlops* são habilitados ou não, resultando na contagem de cada um dos valores necessários (centésimos de segundos, segundos e minutos). Neste caso, o próximo estado que será disponibilizado na saída do *FlipFlop* já é pré-calculado, e somente aparece na saída, quando este *FlipFlop* for habilitado. O resultado é apresentado no diagrama RTL a seguir.



**Figura 3** - RTL timer *single\_clock*

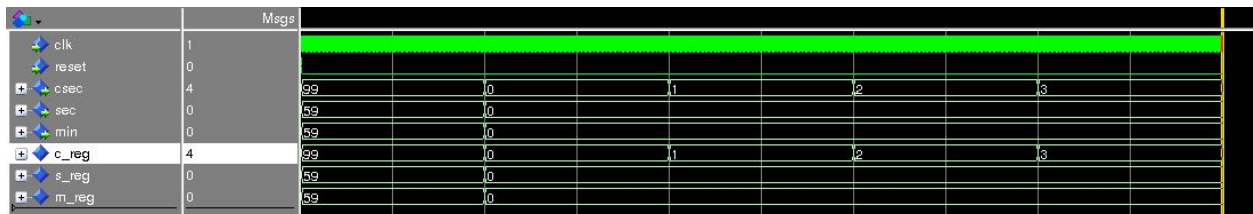
Os valores de área e frequência máxima atingidos pelas diferentes implementações estão descritos na tabela a seguir.

Implementação	Área (LE)	Frequência Máxima
<i>multi_clock</i>	157	296.47 MHz
<i>single_clock</i>	163	252.72 MHz

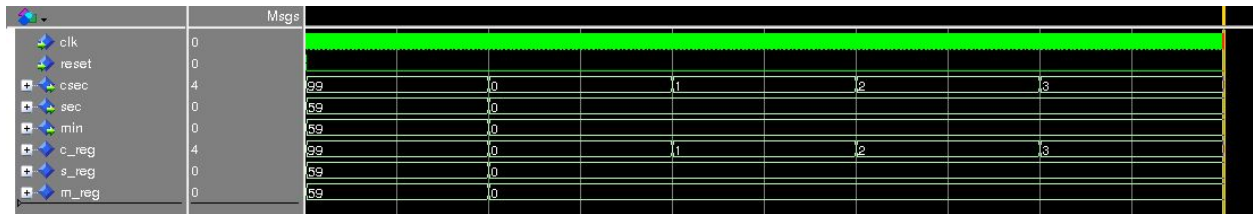
**Tabela 1** - Comparativo implementações arquitetura 1

Em ambas as implementações, a máxima *constraint* de frequência alcançada foi de 250 MHz, definida a partir da seguinte linha SDC: *create\_clock -name CLK -period 250MHz [get\_ports {CLK}]*

O resultado dos testes realizados nas duas implementações (*multi\_clock* e *single\_clock*) podem ser observados nas imagens a seguir.



**Figura 4** - Simulação arquitetura 1 com timer *multi\_clock*



**Figura 5** - Simulação arquitetura 1 com timer *single\_clock*

Em ambos os testes, foi realizado um *reset* inicial e os valores foram setados para 0. Em seguida, forçamos os valores em seus limites (99 centésimos de segundos, 59 segundos e 59 minutos), após executar o teste por 50 ms, observamos a mudança dos três valores para 00 e em seguida, a contagem continuou de forma crescente. A alteração dos centésimos de segundo ocorreu sempre a cada 10 ms, ou seja, 1 centésimo de segundo, de acordo com o especificado.

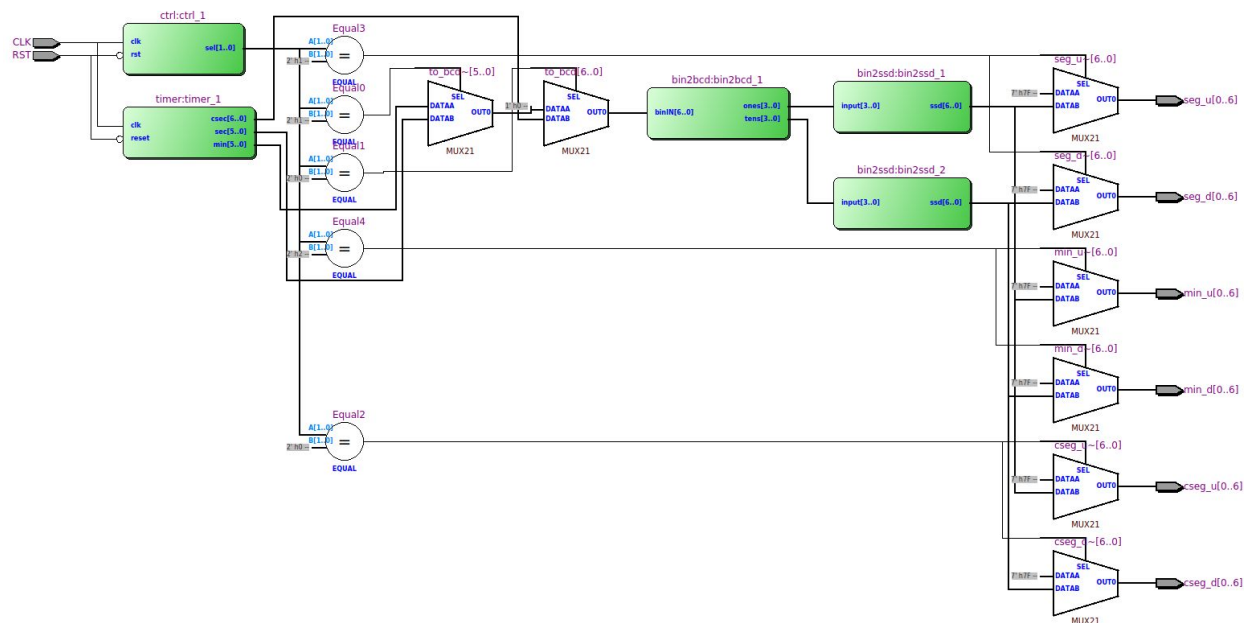
Para execução do teste é necessário abrir a ferramenta ModelSim RTL Simulator e executar o comando “do test\_timer.do”.

Obs.: para funcionamento da simulação, o arquivo *test\_timer.do* deve estar no caminho: <diretorio\_do\_projeto>/simulation/modelsim .

## Arquitetura 2

Para implementação da segunda arquitetura, nos baseamos no fato de que, segundo estudos recentes, a frequência máxima de piscagem observada pelo olho humano é de cerca de 500 Hz, em sua máxima performance [1]. Com base nisso, foi implementado somente um bloco da entidade *bin2bcd* e dois da entidade *bin2ssd* (os mesmos da arquitetura 1), responsáveis por acender os leds dos *displays* de 7 segmentos, fazendo com que somente um valor de dezena e unidade acendam por vez. Desta forma, um bloco de controle foi adicionado, sendo este o responsável por fazer a multiplexação das entradas e saídas dos dois blocos *bin2bcd* e *bin2ssd*, para os *displays* de 7 segmentos. Esta multiplexação é feita a uma taxa de 1 MHz, ou seja, bem longe do que o olho humano é capaz de enxergar, fazendo com que não visualizemos o apagar e acender dos leds e, desta forma, teoricamente, reduzimos a quantidade de elementos lógicos, já que trocamos blocos *bin2bcd* e *bin2ssd* por simples multiplexadores e demultiplexadores.

O diagrama RTL desta arquitetura é apresentado a seguir.



**Figura 6 - Arquitetura 2**

O bloco *timer* utilizado nesta arquitetura é o mesmo utilizado na arquitetura 1, com a implementação *single\_clock*. Também utilizamos *constraint* de 250 MHz, resultando nos valores da tabela a seguir.

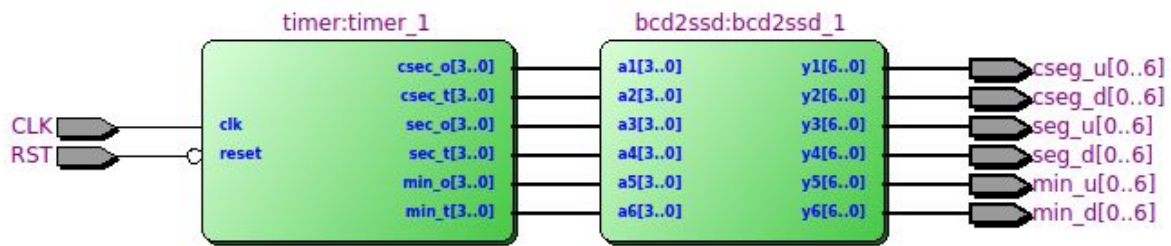
Área (LE)	Frequência Máxima
178	258.53 MHz

**Tabela 2 - Dados comparativos arquitetura 2**

Obs.: como não houve alteração no bloco *timer* (*single\_clock*), não apresentaremos a simulação, já que o resultado obtido é o mesmo da arquitetura 1.

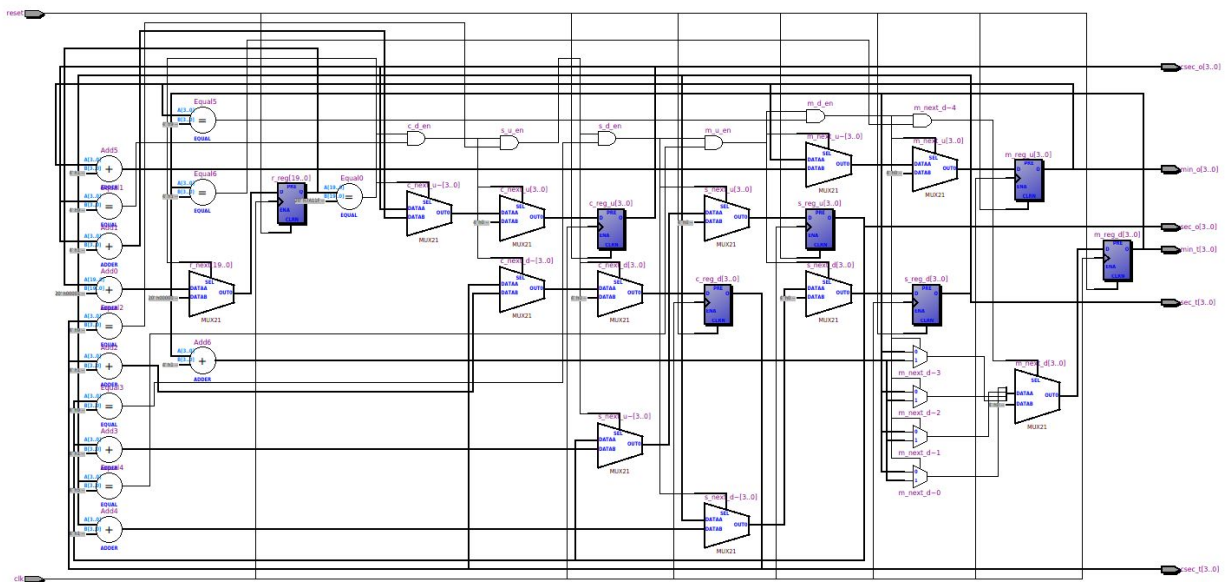
### Arquitetura 3

Nas arquiteturas anteriores, o bloco *Timer* era responsável por fazer a contagem dos valores de 0 a 99 para centésimos de segundos e 0 a 59 para segundos e minutos. Já a separação dos algarismos para representação no *display* de 7 segmentos era realizada nos blocos *bin2bcd*. Nesta terceira arquitetura, o bloco *Timer* é responsável por fazer a contagem já em BCD, ou seja, não há a necessidade do bloco *bin2bcd*. O intuito desta implementação é reduzir ainda mais a quantidade de elementos lógicos e buscar aumentar a frequência máxima do circuito. O diagrama RTL a seguir mostra a visão de topo da arquitetura.



**Figura 7 - Arquitetura 3**

Neste caso, foi necessário alterarmos o bloco *Timer*, resultando no diagrama RTL a seguir.



**Figura 8 - RTL timer arquitetura 3**

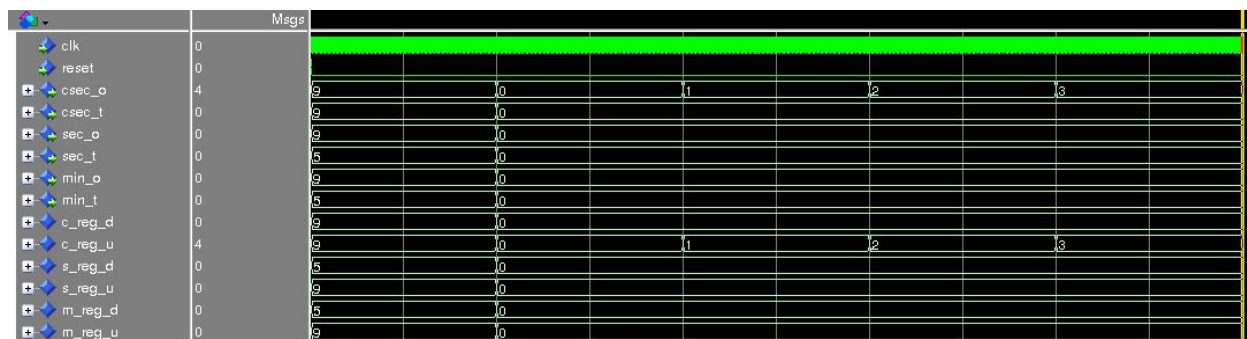
Nesta implementação o valor máximo de constraint de frequência que conseguimos alcançar foi 240 MHz, obtido através da seguinte linha SDC: `create_clock -name CLK -period 240MHz [get_ports {CLK}]`

Os resultados obtidos estão descritos na tabela a seguir.

Área (LE)	Frequência Máxima
125	245.94 MHz

**Tabela 4 - Dados barrel shifter genérico 32 bits**

Para validar o funcionamento do novo *Timer*, foram realizados os mesmos testes feitos na arquitetura 1, alterando apenas as entradas e saídas. Os resultados podem ser vistos na imagem a seguir.



**Figura 9** - Simulação timer arquitetura 3

Para execução do teste é necessário abrir a ferramenta ModelSim RTL Simulator e executar o comando “do test\_timer.do”.

Obs.: para funcionamento da simulação, o arquivo *test\_timer.do* deve estar no caminho: <diretorio\_do\_projeto>/simulation/modelsim .

### Comparativo entre arquiteturas

A tabela a seguir exibe os valores obtidos de área e frequência máxima em cada uma das arquiteturas e implementações.

Arquitetura	Implementação Timer	Área (LE)	Frequência Máxima
Arquitetura 1	<i>multi_clock</i>	157	296.47 MHz
	<i>single_clock</i>	163	252.72 MHz
Arquitetura 2	<i>single_clock</i>	178	258.53 MHz
Arquitetura 3	<i>bcd_counter</i>	125	245.94 MHz

**Tabela 5** - Comparativo entre arquiteturas e implementações

Comparando os resultados obtidos entre as duas implementações da arquitetura 1, notamos que o número de elementos lógicos ficou bem próximo, basicamente pelo fato de que não há mudança explícita de hardware de uma implementação para outra. Todavia, no parâmetro de frequência máxima, era esperado um valor maior para a implementação *single\_clock*, em relação à *multi\_clock*. Esta diferença, apesar de não ter se concretizado, era esperada pelo seguinte motivo: na implementação *multi\_clock*, múltiplos *clocks* são gerados a partir do *clock* de 50 MHz fornecido pela FPGA, e essa criação de novos *clocks* acaba gerando



um *delay* maior na troca de estado dos *FlipFlops*, uma vez que esse atraso é propagado de cima para baixo na topologia. Esse atraso gerado resultaria num *delay* total maior e, teoricamente, faria com que a frequência máxima atingida no circuito *multi\_clock* fosse menor em relação ao *single\_clock*. O motivo pelo qual esta previsão não tenha se concretizado pode ter origem em alguma otimização realizada pela ferramenta de compilação na primeira implementação.

Já comparando a arquitetura 2 com a arquitetura 1, era esperado um número de elementos lógicos menor na segunda topologia, já que nos livramos de alguns blocos *bin2bcd* e *bin2ssd*. Todavia, essa previsão também não se concretizou e, apesar de estarem próximos, o número de elementos da segunda arquitetura foi maior que na primeira. Novamente, é possível que alguma otimização realizada pela ferramenta de compilação tenha reduzido a quantidade de elementos lógicos da primeira arquitetura, fazendo com que esta ficasse mais otimizada, neste quesito, em relação à segunda.

Por fim, a terceira arquitetura comparada com as demais, apresentou um número de elementos lógicos menor, como era o esperado, já que removemos os blocos *bin2bcd* da topologia.

Obs.: todas as três arquiteturas tiveram seu funcionamento validado diretamente na placa FPGA, garantido que a contagem dos centésimos de segundos, segundos e minutos, estavam sendo feitas corretamente.

## Referências

[1] DAVIS, James; HSIEH, Yi-hsuan; LEE, Hung-chi. Humans perceive flicker artifacts at 500 Hz. **Scientific Reports**, [s.l.], v. 5, n. 1, 3 fev. 2015. Disponível em: <<https://www.nature.com/articles/srep07861.pdf>>. Acesso em: 08 jun. 2019.