

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Licenciatura em Engenharia Informática e de Computadores



Relatório do 2.º Laboratório
de
Lógica e Sistemas Digitais

Circuitos Sequenciais

5 de janeiro de 2022

1 Introdução

Este terceiro laboratório para avaliação de Lógica e Sistemas Digitais tem como objetivo o desenvolvimento de um circuito sequencial a partir de uma especificação, simular e implementar o circuito com a placa de desenvolvimento DE10-Lite da Intel.

Este circuito pretende projetar um sistema de controlo dos semáforos de uma passadeira. Junto à passadeira encontram-se dois semáforos, S1 que é para os automóveis contendo três luzes (V – Verde, A – Amarelo, E – Encarnado) e outro semáforo, S2 que é o semáforo dos peões, tendo duas cores, (VP – Verde, EP -Vermelho).

O semáforo S2 tem também um botão de pressão (B) para que os peões possam solicitar passagem.

O funcionamento dos semáforos é o seguinte:

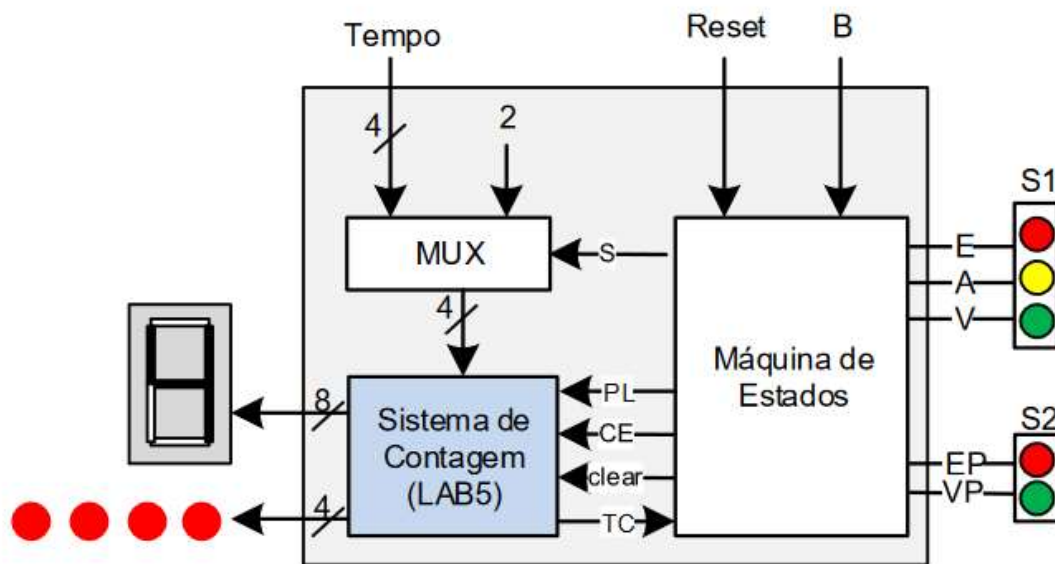


Imagem 1 – Diagrama de blocos do controlo de semáforos

2 Desenvolvimento do Trabalho

2.1 Descrição das funções lógicas e os respectivos diagramas lógicos

Começamos por desenvolver o ASM-chart da máquina de estados:

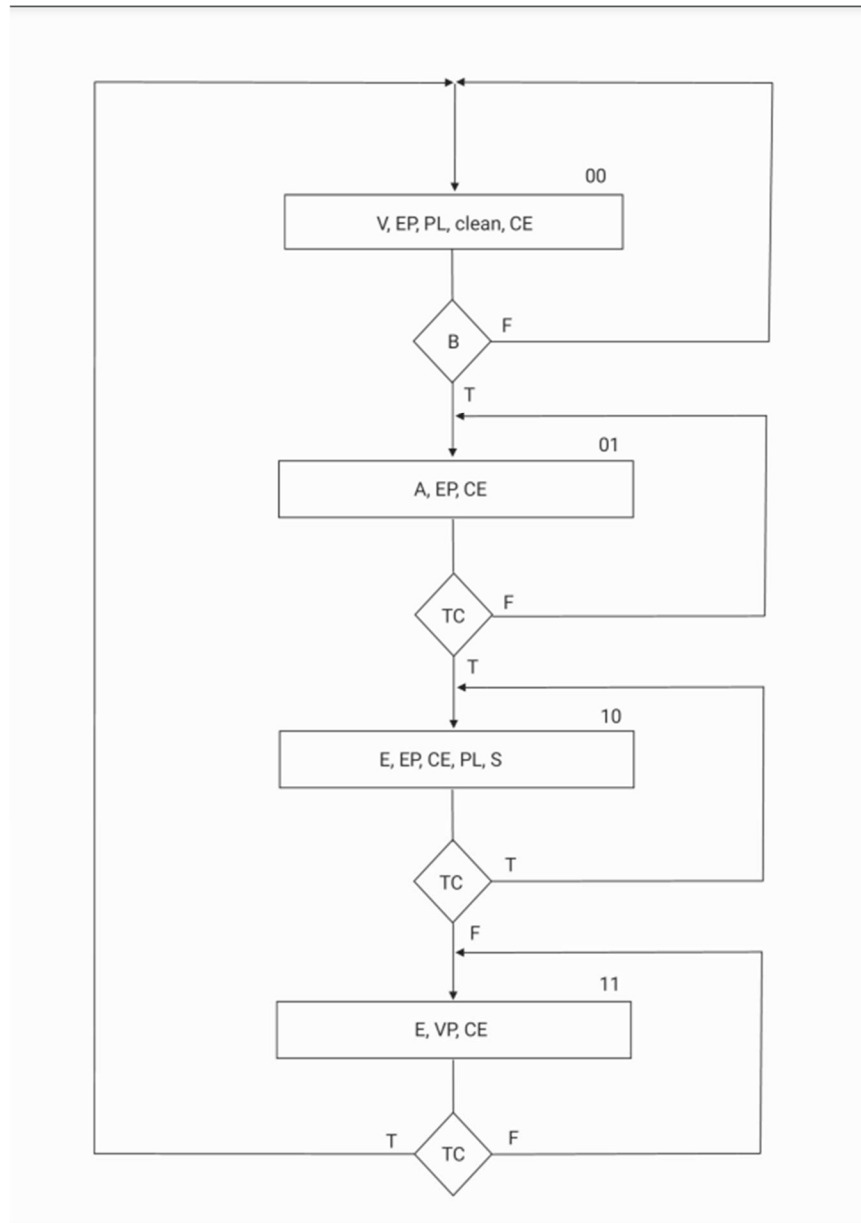


Imagem 2 - ASM-chart da máquina de estados

De seguida determinamos as fórmulas lógicas da máquina de estados que irão ser apresentadas mais à frente neste relatório.

A máquina de estados começa no estado 00, em que o semáforo está verde para os automóveis e está vermelho para os peões.

Para a mudança de estado ocorrer, o botão (B) tem de ser premido. Assim que tal aconteça, passamos para o segundo estado (01). Neste estado o semáforo dos automóveis fica amarelo durante 2 segundos. Quando passam os 2 segundos, o TC (Terminal Count) passa a 1, ou seja, indica o fim da contagem. Quando TC ficar a 1, passamos então ao próximo estado 10.

No terceiro estado 10, o semáforo fica vermelho tanto para os automóveis como para os peões durante 1 segundo.

Com a passagem desse segundo, passamos então ao estado 11 onde o sinal dos automóveis se encontra vermelho e o dos peões se encontra verde. Na passagem do estado 10 para o estado 11 o TC tem de estar a 0, sendo que necessita de carregar um novo valor (1 segundo) para passar ao próximo estado.

Após *Tempo* segundos passados, voltamos ao estado inicial.

2.1.1 Módulo máquina de estados

Este módulo foi desenvolvido a partir do ASM-chart, do qual retiramos as seguintes expressões:

$$V = \overline{Q1} \cdot \overline{Q0}$$

$$EP = (\overline{Q1} \cdot \overline{Q0}) + (\overline{Q1} \cdot Q0) + (Q1 \cdot \overline{Q0})$$

$$PL = (\overline{Q1} \cdot \overline{Q0}) + (Q1 \cdot \overline{Q0})$$

$$clear = \overline{Q1} \cdot \overline{Q0}$$

$$CE = (\overline{Q1} \cdot \overline{Q0}) + (\overline{Q1} \cdot Q0) + (Q1 \cdot \overline{Q0}) + (Q1 \cdot Q0)$$

$$A = \overline{Q1} \cdot Q0$$

$$S = Q1 \cdot \overline{Q0}$$

$$VP = Q1 \cdot Q0$$

$$E = (Q1 \cdot \overline{Q0}) + (Q1 \cdot Q0)$$

$$D0 = (\overline{Q1} \cdot \overline{Q0} \cdot B) + (\overline{Q1} \cdot Q0 \cdot \overline{TC}) + (Q1 \cdot \overline{Q0} \cdot \overline{TC}) + (Q1 \cdot Q0 \cdot \overline{TC})$$

$$D1 = (Q1 \cdot Q0 \cdot TC) + (Q1 \cdot \overline{Q0} \cdot TC) + (Q1 \cdot \overline{Q0} \cdot \overline{TC}) + (Q1 \cdot Q0 \cdot \overline{TC})$$

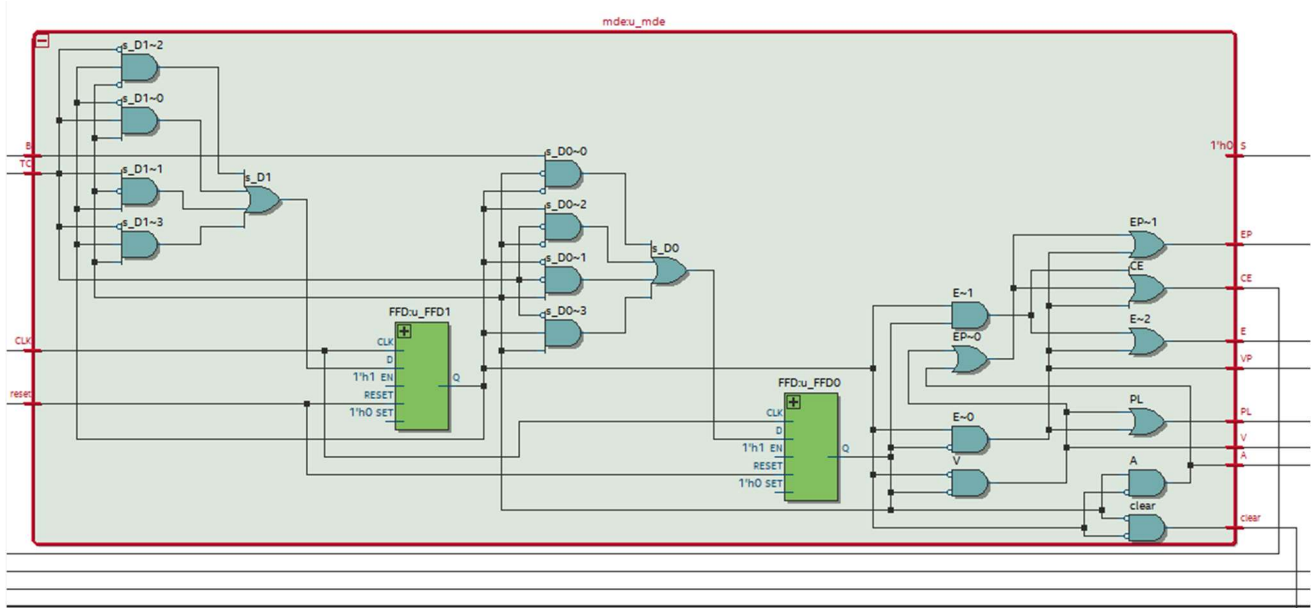


Imagem 3 – máquina de estados

Dentro desta máquina de estados encontram-se 2 flip flops que estão encarregues pela mudança de estados e pelo seu armazenamento também. Usamos 2 flip flops, sendo que há 4 estados possíveis.

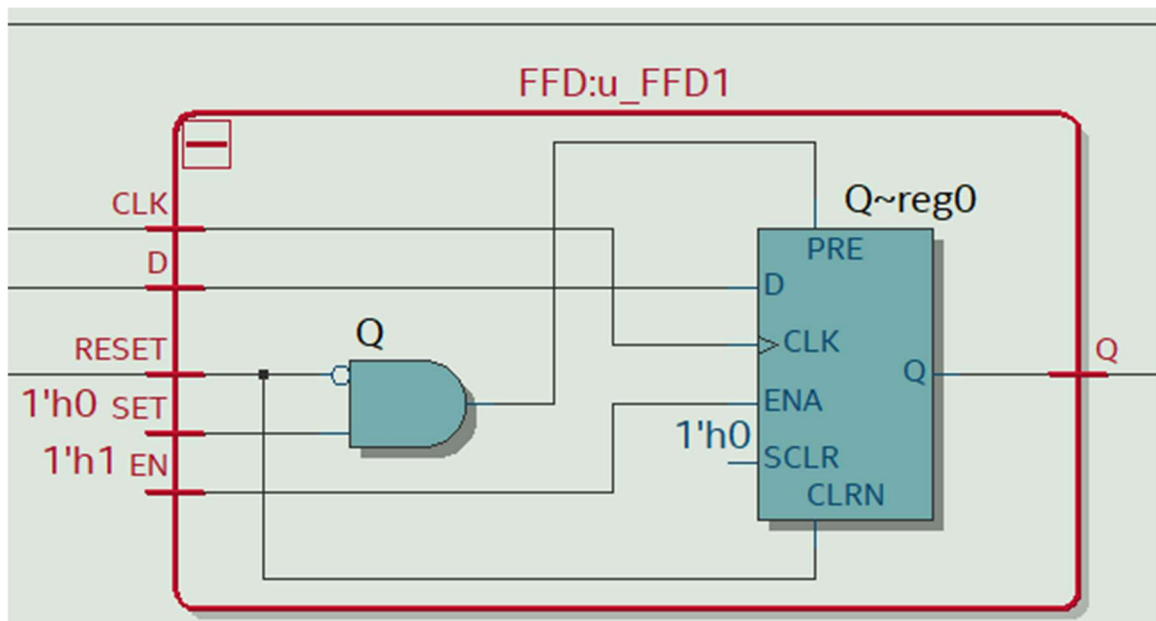


Imagem 4 – um dos flip flops da máquina de estados

2.1.3 Módulo Lab5

O lab5 foi realizado em aula e é constituído por um adder, que é composto por 4 full adders, por um multiplexer e por um registo, todos estes desenvolvidos pelo grupo. O lab5 é também formado por um clkDIV, decoderHex, FFD e int7seg, no entanto, estes módulos foram fornecidos pelo docente.

Este módulo tem como objetivo a contagem decrescente dos segundos com parallel load.

A entidade de topo do lab5 que une todos os seus componentes tem o seguinte aspeto:

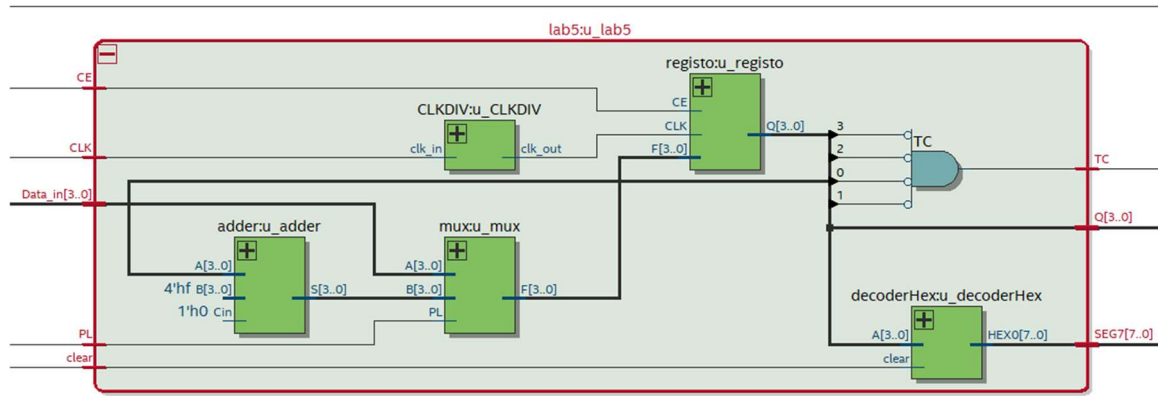


Imagem 4 – entidade de topo lab5

Os restantes componentes do lab 5 têm o seguinte aspeto:

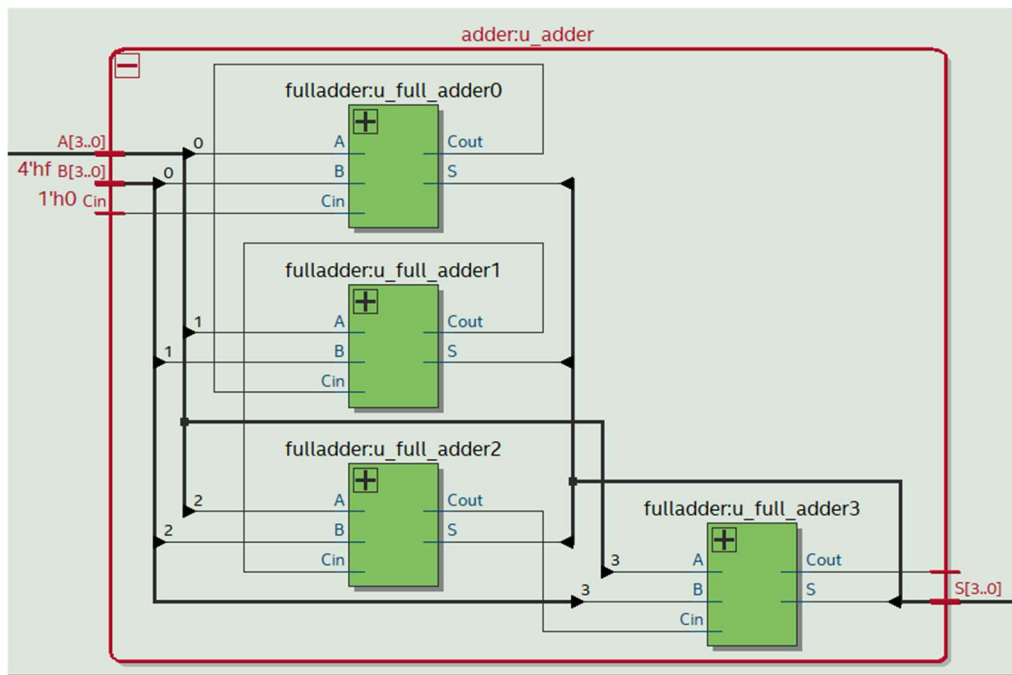


Imagem 5 – Adder

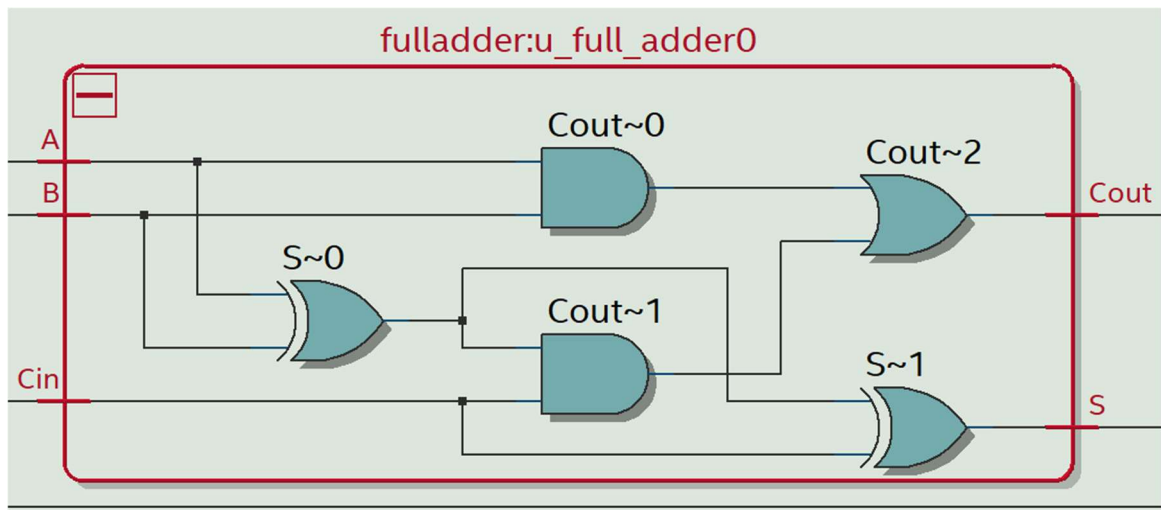


Imagem 6 – exemplo de um dos full adders

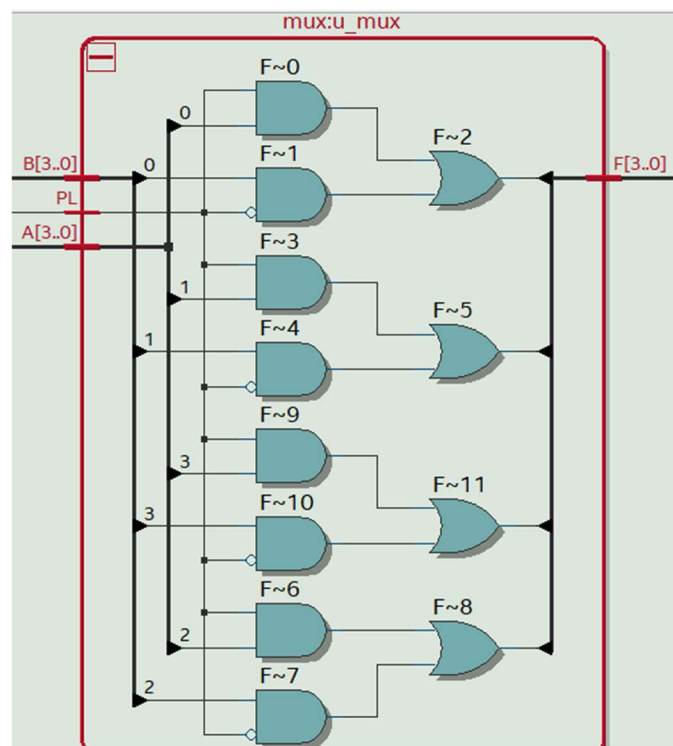


Imagem 7 – Multiplexer do lab 5

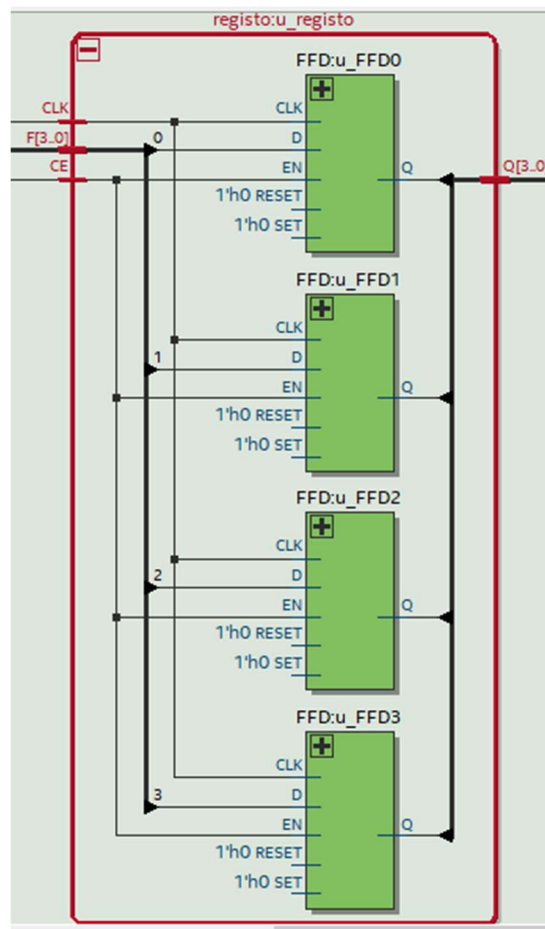


Imagem 8 - Registo

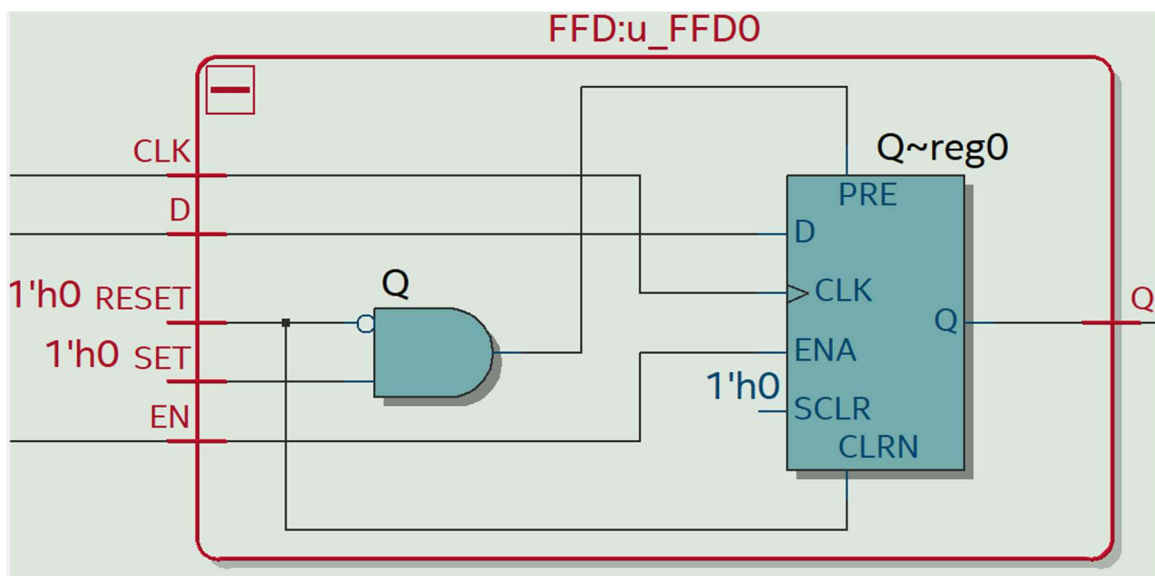


Imagem 9 -Flip flop

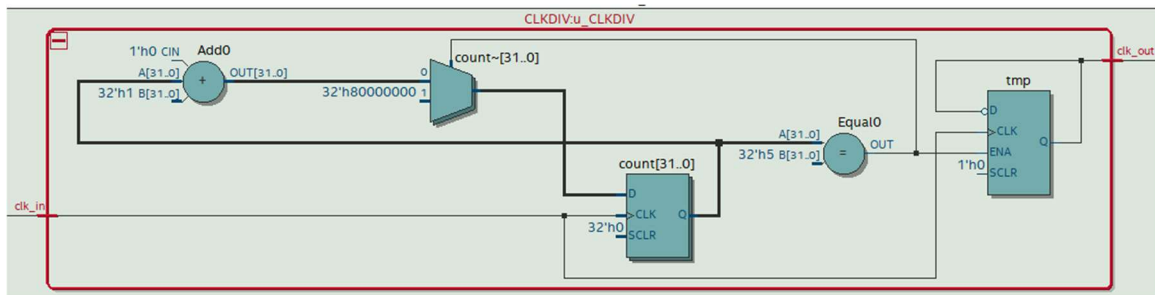


Imagem 10 – Clock div (fornecido pelo docente)

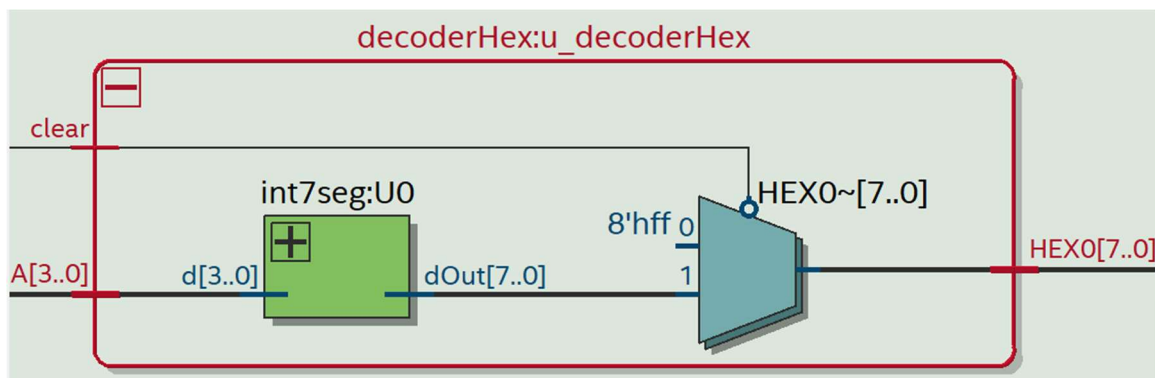


Imagem 11 – Decoder (fornecido pelo docente)

2.1.4 Módulo Multiplexer

Este multiplexer é responsável pela seleção da contagem entre *tempo* segundos ou 2 segundos.

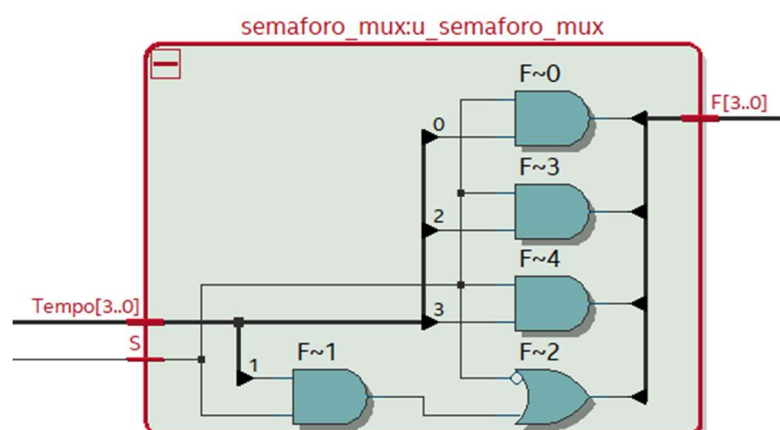


Imagem 12 – Multiplexer da entidade de topo

3 Montagem e teste laboratorial

Após obtermos todos os módulos necessários, passámos à montagem da entidade de topo onde fizemos todas as ligações necessárias entre os diversos módulos de forma a obtermos um circuito funcional e com o objetivo de ser posteriormente testado na placa de desenvolvimento DE-Lite da Intel. A entidade de todo recebe como entrada um B (botão), reset, clock e *tempo* com 4 bits. De saída tem E e EP (encarnado para os automóveis e para os peões, respetivamente), V e VP (verde para os automóveis e para os peões, respetivamente), A (amarelo para os automóveis), Q que apresenta os números em leds e HEXo que apresenta a contagem num ecrã da placa.

A entidade de topo tem o seguinte aspeto:

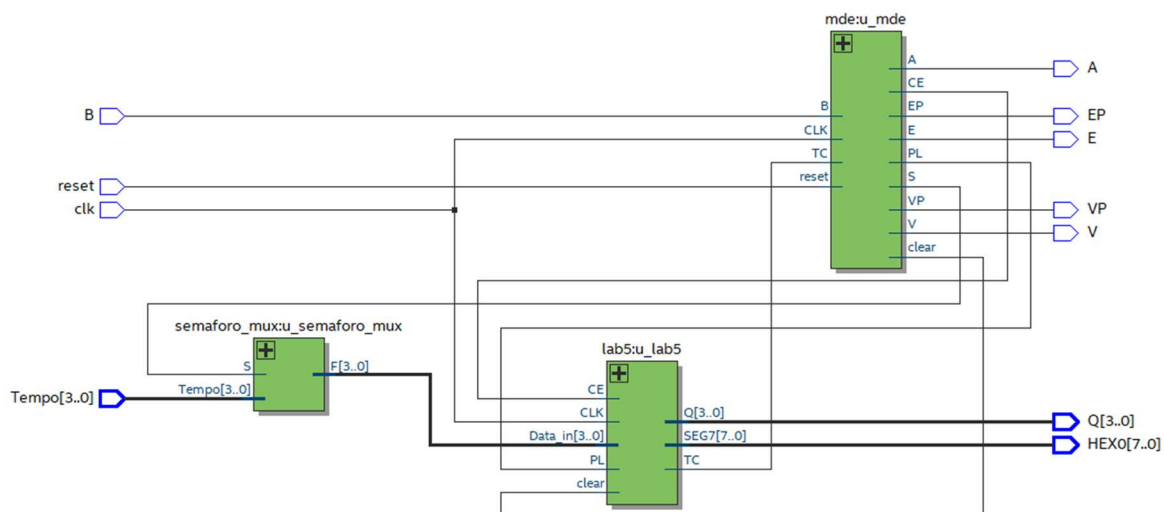


Imagem 14 – Entidade de topo

Por fim, e antes de implementar na placa, utilizamos a ferramenta de simulação do programa Quartus Prime para verificarmos se as alterações estavam a ser feitas de forma correta e nos intervalos de tempo indicados pelo docente.

4 Conclusão

Concluindo o trabalho, conseguimos aprofundar os nossos conhecimentos de circuitos sequenciais utilizando *VHDL*. Através do mesmo conseguiu-se implementar e aplicar diferentes conceitos aprendidos na cadeira como por exemplo a utilização de *flip-flops*, máquina de estados, contadores com *parallel load*, elaboração de um *ASM-chart* e também a simulação do circuito no programa Quartus Prime e verificar os resultados obtidos.