

SUMÁRIO

1	INTRODUÇÃO.....	3
1.1	Objetivos	3
1.2	Problemas	4
2	METODOLOGIA.....	4
2.1	Máquina de Estado Alto Nível	5
2.2	ASMChart	7
2.3	Datapath/Caminho de Dados.....	8
2.4	Máquina de Estado Baixo Nível	9
3	RESULTADOS E SIMULAÇÕES	10
3.1	Porta OR.....	10
3.2	Contadores	10
3.3	Multiplexadores	11
3.4	Comparadores	11
3.5	Máquina de estado baixo nível.....	12
3.6	Datapath	12
3.7	Simulação do Projeto RTL.....	13
3.8	Simulação do Projeto RTL para via A.....	14
3.9	Simulação do Projeto RTL para via B.....	15
4	CONCLUSÃO.....	16
	REFERÊNCIAS BIBLIOGRÁFICAS.....	17

1 INTRODUÇÃO

O sistema de controle de tráfego é um sistema importante para controlar a segurança da operação dos veículos e pessoas em uma determinada via. Este sistema pode ser implementado utilizando um microcontrolador, FPGA ou um ASIC. A FPGA possui algumas vantagens sobre o microcontrolador como: velocidade, maiores números de portas de entradas e saídas e também possui o melhor desempenho em geral. Em outro lado, o ASIC é mais otimizado, porém o custo de implementação é maior em comparação com a implementação em FPGA. Devido a configurabilidade dos circuitos internos da FPGA, permite a prototipação e a atualização rápida de um hardware para uma aplicação específica.

Através de projeto RTL, é possível capturar o funcionamento de um sistema real através da máquina de estado alto nível e o transformar em blocos operacionais (datapath) e controladores (máquina de estado baixo nível) formando uma inteligência capaz de gerenciar o funcionamento de um determinado sistema digital.

Para verificar o funcionamento de um sistema digital modelado pelo projeto RTL é necessário fazer a simulação antes de embarca-lo em um hardware. Este passo é realizável através das linguagens de descrição de hardware como VHDL.

1.1 Objetivos

Este trabalho tem o objetivo de descrever o funcionamento do projeto da segunda unidade da disciplina de sistemas digitais do período remoto 2021.2 ministrado pelo Professor Antonio Wallace Antunes Soares.

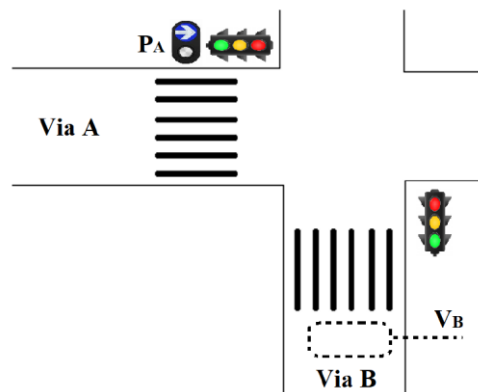
O projeto abrange a implementação de um sistema de controle de sinais de tráfego em um cruzamento utilizando a FPGA, projeto RTL e VHDL. Neste cruzamento existe duas vias (A e B) sendo um é uma via principal e o outro é uma via secundária. Além disso, existem também os sensores para detectar a presença dos veículos e dos pedestres.

1.2 Problemas

O problema deste trabalho consiste em cruzamento entre uma via principal (A) e uma via secundária (B) como mostra a Figura 1. A mudança de sinal na via principal ocorre quando houve um pedestre na via principal (P_A) ou houve um carro passar na via secundária (V_B). Se esta condição for satisfeita, então o sinal verde na via principal tornando-se para amarelo por 2 segundos e depois seguindo o sinal verde na via secundária por 5 segundos, seguido por um amarelo de 2 segundos.

A outra condição é quando não houve a passagem de carro na via secundária e não houve o pedestre na via principal, se isso for acontecido deve ser dar um acréscimo de um segundo para o sinal verde na via principal até no máximo doze segundos. Se esta condição continua mantido, deve haver mais um acréscimo a cada três segundos para verificação da condição de haver carros ou pedestres na via secundária e na via principal.

Figura 1 - Cruzamento de duas vias.



Fonte: Notas da aula.

2 METODOLOGIA

Para capturar o problema definido com o projeto RTL, o primeiro passo é determinar a máquina de estado de alto nível. Feito isso, o segundo passo do projeto é utilizar o ASMChart para determinar os blocos sequenciais e combinacionais. O terceiro passo é determinar o datapath/caminho de dados e por fim determinar a máquina de estado de baixo nível.

2.1 Máquina de Estado Alto Nível

Para capturar o funcionamento do sistema de controle de tráfego definido no problema do projeto, foi utilizado a máquina de estado alto nível com oito estados como mostra a Figura 2. As entradas da máquina de estado são o clock, sensor de presença do pedestre na via principal (P_A) e o sensor de detecção de veículos na via secundária (V_B). Além disso, foi definido duas variáveis de contagem (Cont1 e Cont2), a primeira variável de contagem tem a função de armazenar e atualizar a contagem do tempo a cada três segundos do sinal verde da via principal. A outra variável de contagem tem a função de armazenar e atualizar a contagem do tempo na via secundária quando houve a presença de pedestres ou carros. As saídas são os sinais vermelhos, os sinais amarelos e os sinais verdes na via A e via B, respectivamente. Os sinais de saídas são representados no diagrama da máquina de estado alto nível da seguinte maneira:

- RA – Significa o sinal vermelho na via A.
- YA – Significa o sinal amarelo na via A.
- GA – Significa o sinal verde na via A.
- RB – Significa o sinal vermelho na via B.
- YB – Significa o sinal amarelo na via B.
- GB – Significa o sinal verde na via B.

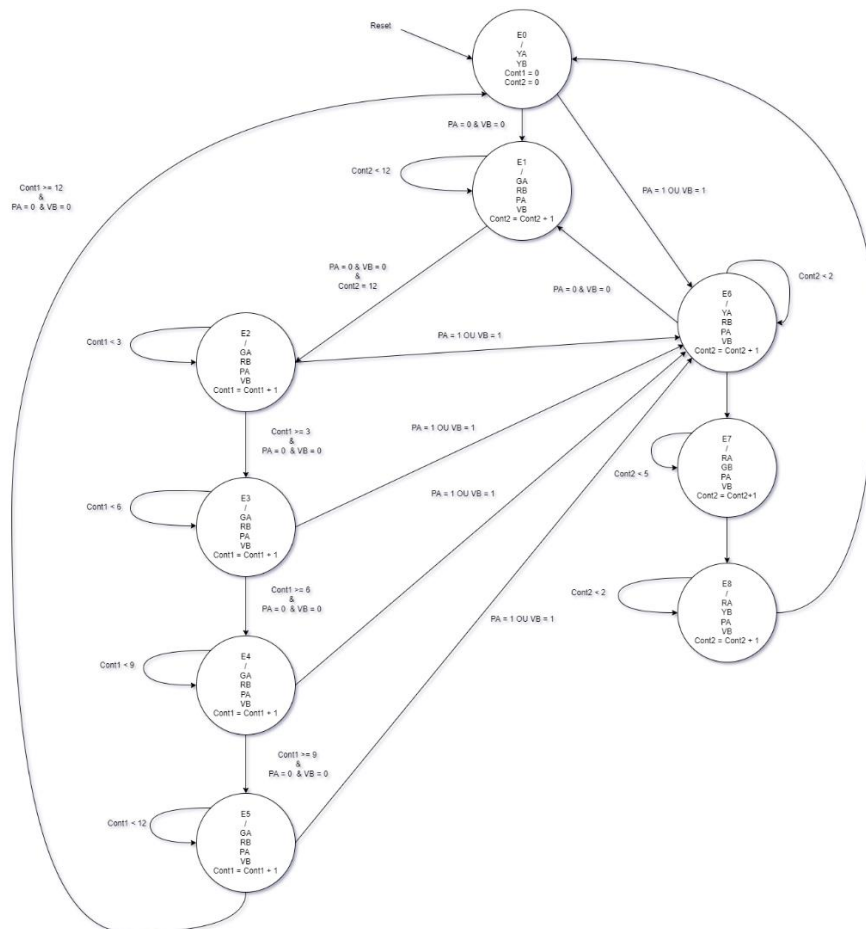
No estado E0 (Estado inicial) quando a máquina de estado é ligada às duas variáveis de contagens são zerados. Por conveniência os sinais de saídas foram definidas como YA e YB.

A transição para o estado E1 (Estado de sinal verde de doze segundos com acréscimo de um segundo) ocorre quando não houve pedestre na via A ($P_A = 0$) e não houve a detecção de carros na via B ($V_B = 0$). Após o tempo de doze segundos estourar ocorre transição para estado E2 (Estado de sinal verde de acréscimo 3 segundos). Esta condição vai permanecer até 3 segundos. A transição para estado E3 (Estado sinal verde acréscimo 6) ocorre quando a verificação de que não houve a presença de carros na via secundária e também a presença de pedestre na via primária é confirmado, caso contrário a transição é para o estado E6 (Estado quando

houve pedestres ou veículos). Esta condição também vai permanecer até 3 segundos. A mesma lógica é aplicada para os estados E4 (Estado sinal verde acréscimo 9) e E5 (Estado sinal verde acréscimo 12) completando a duração máximo de 12 segundos. Após o fim dessas condições ocorre a transição para o estado E0 para zerar as contagens (Cont1 e Cont2) e recomeçar o processo.

A transição para o estado E6 (Estado de sinal YA e RB) ocorre quando houve pedestre na via A ou a presença de carros na via B. Esta condição vai permanecer até 2 segundos. Após este tempo estourar ocorre a transição para o estado E7 (Estado de sinal RA e GB) e permanecer até 5 segundos. Após este tempo de 5 segundos ocorre a transição para o estado E8 (Estado de sinal RA e YB) e permanecer até 2 segundos. Ao completar essas condições a contagem (Cont1) é zerada (transição para estado E0) e recomeçar o processo.

Figura 2 - Máquina de Estado Alto Nível.



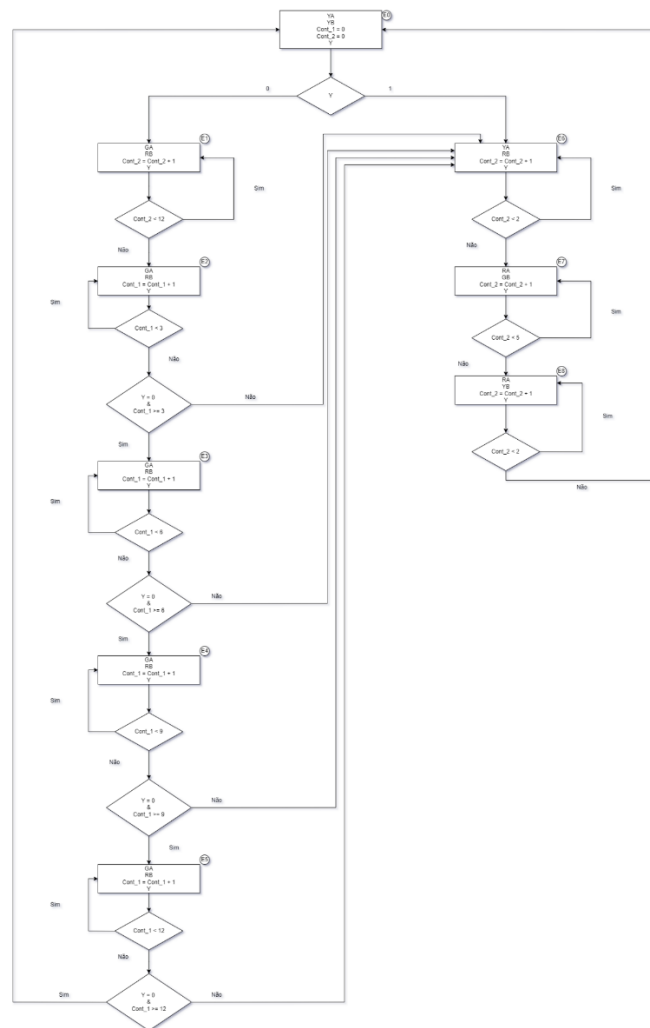
Fonte: Próprio Autor.

2.2 ASMChart

A ASMChart mostrado na Figura 3 foi obtido a partir de máquina de estado alto nível apresentado na Figura 2. A partir deste diagrama foi possível tirar os blocos combinacionais e sequenciais necessários para formar os blocos operacionais (datapath) do projeto. Avaliando a Figura 3, é possível identificar os seguintes blocos combinacionais e sequenciais:

- Dois contadores de 4 bits.
- Duas multiplexadores de 4 bits.
- Duas comparadores de igualdade com duas entradas de 4 bits e a saída de um bit.

Figura 3 - ASMChart.

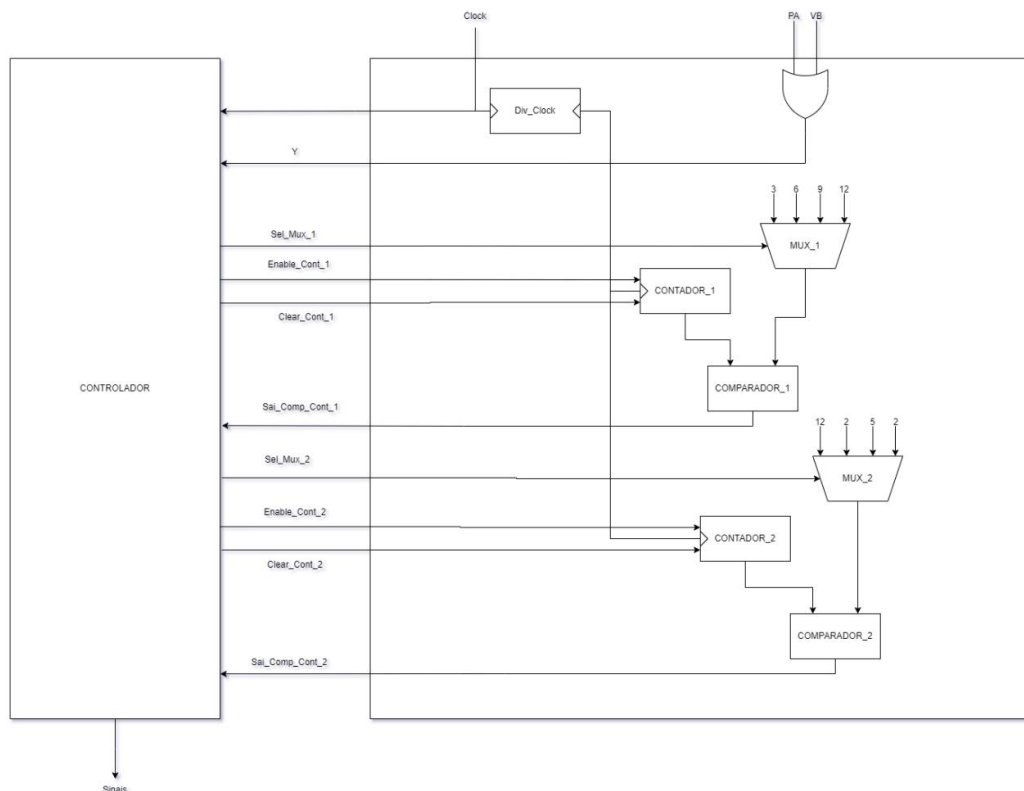


Fonte: Próprio Autor.

2.3 Datapath/Caminho de Dados

A configuração dos caminhos de dados e a comunicação com o controlador é apresentado na Figura 4. O primeiro bloco é a porta lógica OR para capturar a variável P_A de um bit que determina presença de pedestre na via A e a variável V_B de um bit que determina a presença de carros na via B. A saída desta combinação é levada para o controlador. Para realizar a contagem das condições especificadas para via A pela máquina de estado alto nível são utilizados os blocos de multiplexadores, os blocos de contadores e os blocos de comparadores. Onde a saída de comparador é levada para o controlador. Os sinais de enable e clear do contador é decidido pelo controlador, assim como os sinais das seleções dos valores a ser comparado do bloco de multiplexador. A mesma lógica é aplicada para as condições especificadas para via B pela máquina de estado alto nível. O único bloco geral conectado o controlador para o datapath é o divisor de clock. Os sinais que representa semáforos são levados direto de controlador para a placa já que nenhuma operação é realizada sobre esta variável.

Figura 4 - Datapath/Caminho de Dados.

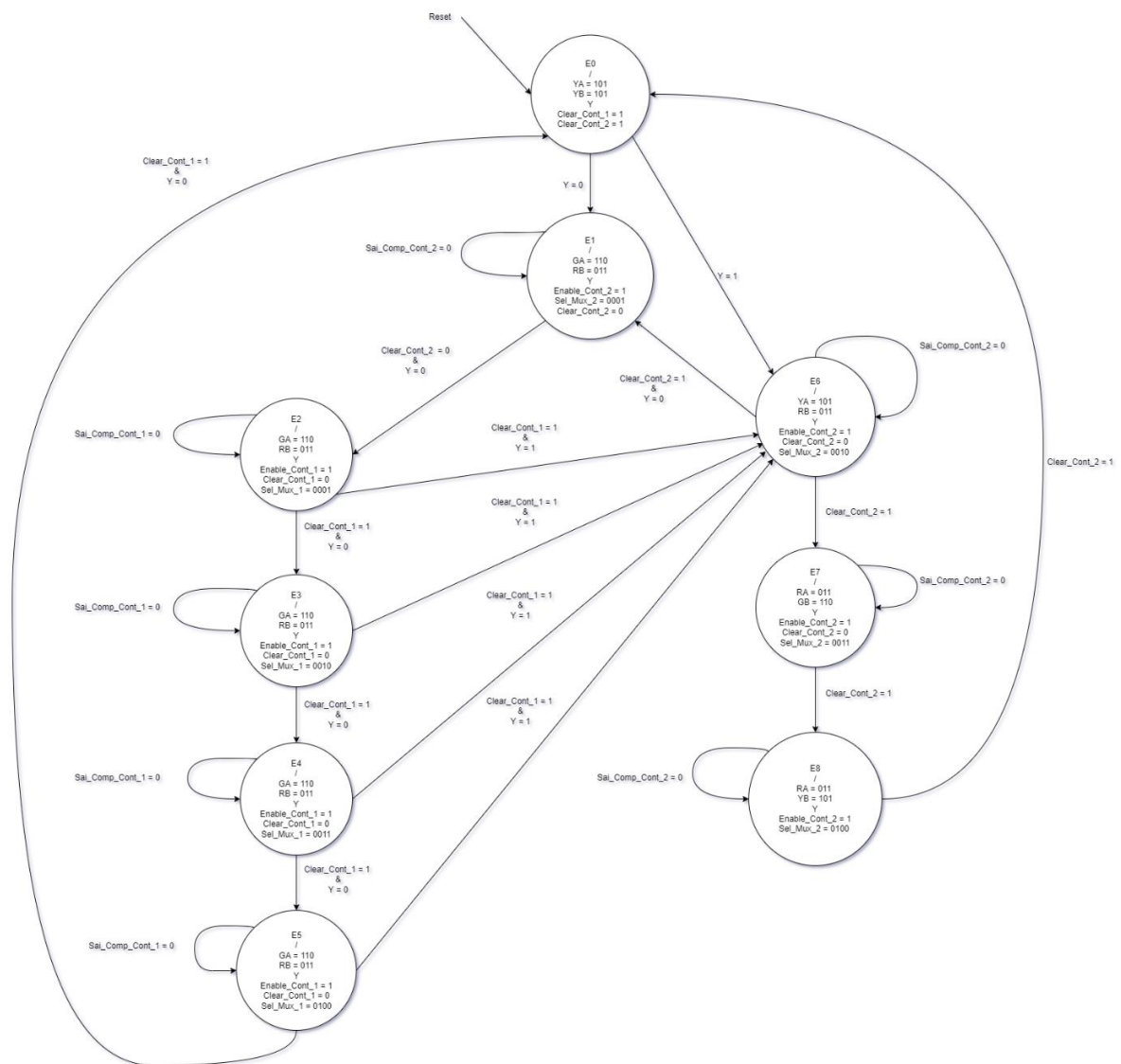


Fonte: Próprio Autor.

2.4 Máquina de Estado Baixo Nível

A Figura 5 apresenta a máquina de estado baixo nível (controlador) para comunicar com o datapath definido na Figura 4. Nesta máquina de estado de baixo nível foi definido os valores de habilitação e desabilitação de contadores como sendo um (habilitar) e zero (desabilitar) e as saídas de comparadores de um sendo igual e não é igual sendo zero. A seleção dos multiplexadores foi definida para a faixa de 0001 até 0100 para selecionar os valores a ser comparados com a saída de contadores conforme o que foi mostrado no datapath.

Figura 5 - Máquina de Estado Baixo Nível.



Fonte: Próprio Autor.

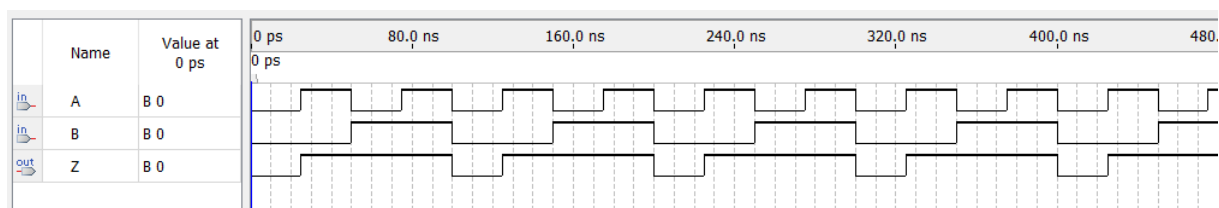
3 RESULTADOS E SIMULAÇÕES

Para verificar o funcionamento do projeto de VHDL o passo importante é fazer as simulações para verificar todos os funcionamentos dos componentes combinacionais e sequenciais que formam o caminho de dados, assim como o controlador que executa a inteligência do sistema.

3.1 Porta OR

A porta OR é um bloco combinacional que recebe como a entrada duas variáveis e retorna a saída em forma de variável booleana. A saída é verdadeira (bit 1) quando uma das entradas é verdadeira. Para este projeto foi utilizado esta porta para capturar os sensores de veículos na via secundária e a presença de pedestre na via principal. A simulação do funcionamento deste bloco é mostrada na Figura 6.

Figura 6 - Simulação da Porta OR.

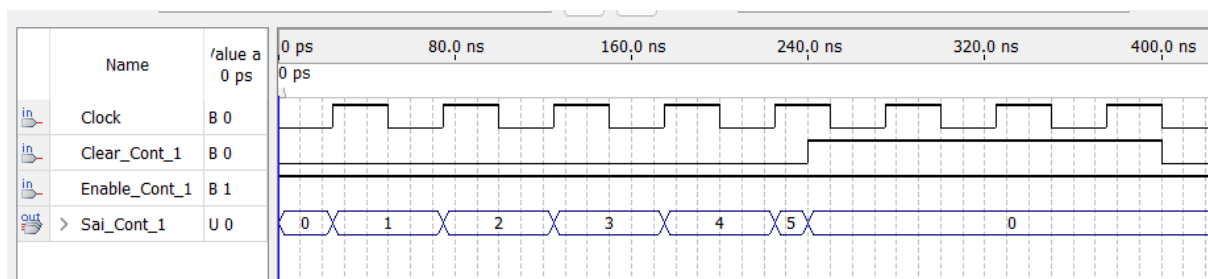


Fonte: Próprio Autor.

3.2 Contadores

Contadores são blocos sequenciais que recebe como a entrada o clock, a clear e o enable. Quando habilitar a entrada enable deste bloco a contagem é feita indefinidamente até que especificar a condição de parada através de habilitação da entrada de clear. Este bloco é utilizado neste projeto para contar a duração do tempo para os sinais definido em cada estado do projeto. A simulação deste bloco é apresentada na Figura 7.

Figura 7 - Simulação do Contador.

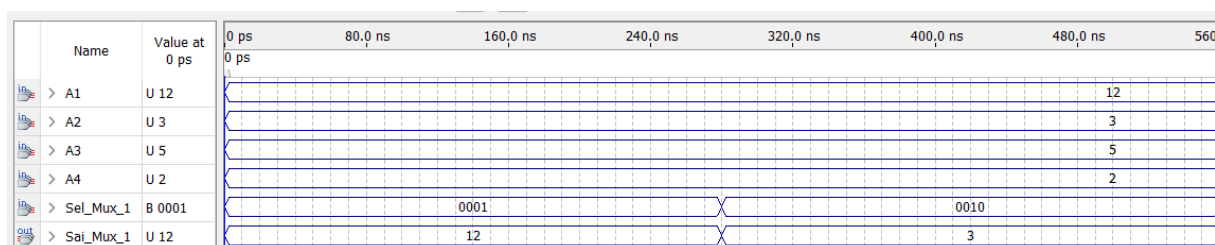


Fonte: Próprio Autor.

3.3 Multiplexadores

Os multiplexadores são blocos combinacionais que recebem múltiplos sinais de entradas e deixar apenas passar um sinal conforme a entrada de seleção. Para este projeto foi utilizado este bloco para selecionar a constante a ser comparado com a contagem de tempo do contador. A simulação deste bloco é apresentada na Figura 8.

Figura 8 - Simulação do Multiplexador.

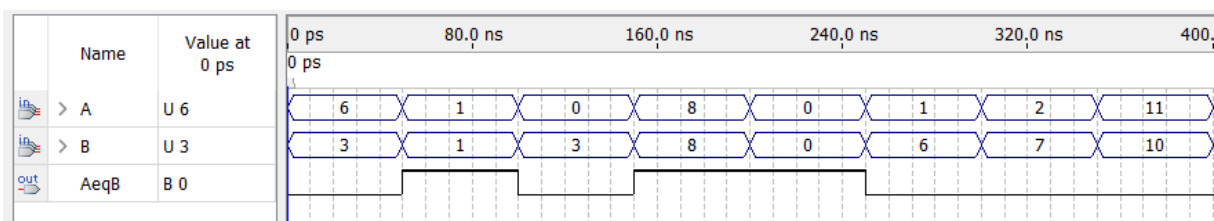


Fonte: Próprio Autor.

3.4 Comparadores

Os comparadores são blocos combinacionais que recebem como duas entradas de qualquer tamanho e realizando as operações de comparações como igual, maior do que e o menor que, retornando como a saída em forma de variável booleana. Para este projeto foi utilizado apenas a operação igual para comparar a saída dos contadores e dos multiplexadores. A simulação deste bloco é apresentada na Figura 9.

Figura 9 - Simulação do Comparador.

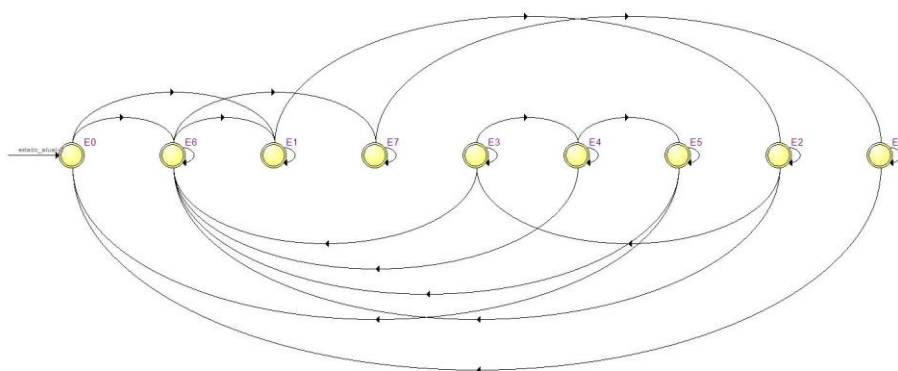


Fonte: Próprio Autor.

3.5 Máquina de estado baixo nível

A Figura 10 mostra o resultado da simulação da máquina de estado baixo nível em VHDL. Pela comparação estrutural com a Figura 5 pode-se observar que, estes diagramas são iguais.

Figura 10 - Máquina de Estado Baixo Nível em VHDL.

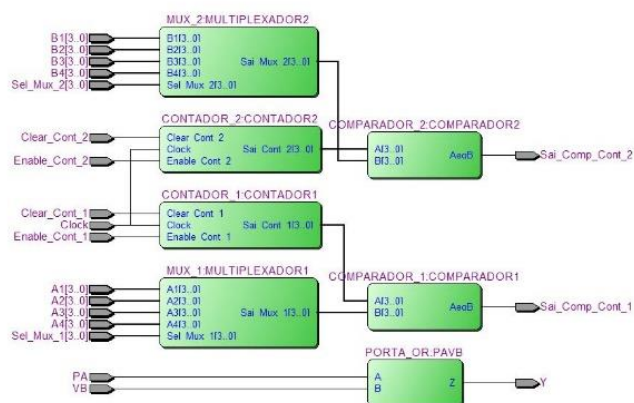


Fonte: Próprio Autor.

3.6 Datapath

A Figura 11 apresenta o resultado da simulação de datapath em VHDL. As variáveis A1, A2, A3 e A4 são os valores de entradas do multiplexador utilizado para comparar o contador da via principal. As variáveis B1, B2, B3, e B4 são os valores de entradas do multiplexador utilizado para comparar o contador da via secundária. Pela comparação estrutural com a Figura 4 pode-se observar que os diagramas são iguais.

Figura 11 - Datapath em VHDL.

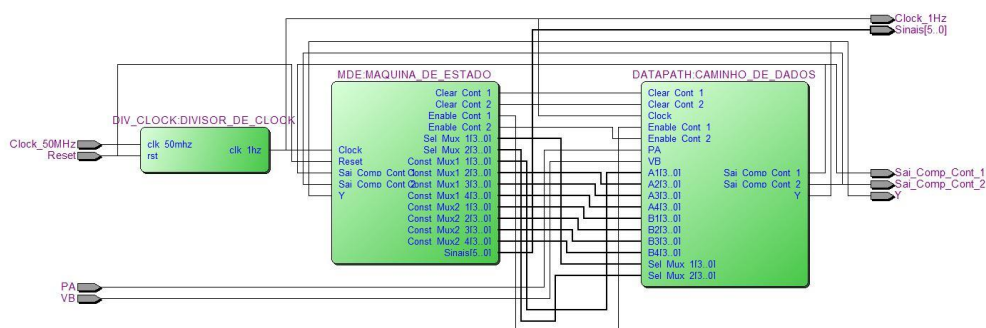


Fonte: Próprio Autor.

3.7 Simulação do Projeto RTL

A Figura 11 apresenta simulação do Projeto RTL, que mostra a conexão em detalhe entre o controlador e os blocos operacionais.

Figura 12 - Projeto RTL em VHDL.



Fonte: Próprio Autor.

3.8 Simulação do Projeto RTL para via A.

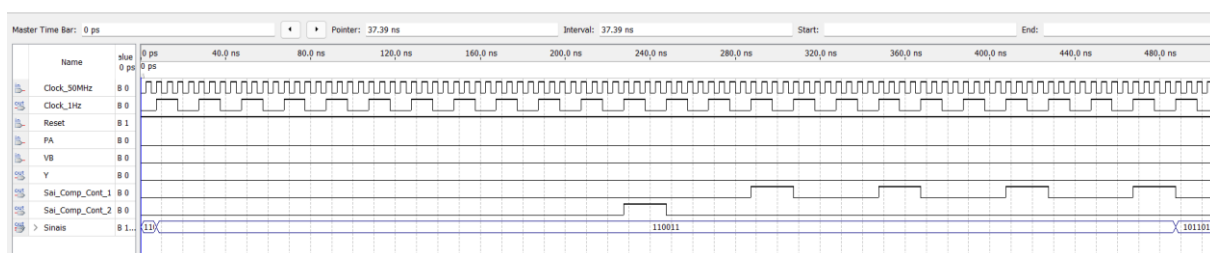
A Figura 13 mostra o resultado da simulação para a situação onde não houve pedestre na via principal e também não houve a presença de carros na via secundária. Para garantir esta situação, foi atribuído o valor das entradas PA e VB para zero, como a consequência a saída Y também igual a zero devido à passagem destas duas variáveis pela porta OR definido no datapath.

A mudança de sinal ocorre pela contagem do clock de 1 Hz que, para o propósito de simulação foi convertido a partir de um clock geral de 200 MHz. Sabendo que o clock da FPGA é 50 MHz, na pinagem deve ser alterado o valor de contagem no bloco divisor de clock se adequar a conversão de 50 MHz para 1 Hz.

Para verificar se o estado E1 tem a duração de 12 segundos como mostrado na máquina de estado alto nível, pode se observar na figura abaixo pela variável Sai_Comp_Cont_2 que vai para um quando o clock de 1 Hz é igual a doze. Isso significa que a comparação entre a saída do contador e a saída do multiplexador é igual e ocorre a transição para o estado E2.

Na verificação do estado E2 (acrécimo um segundo) e dos estados E3, E4 e E5 que são os estados de acréscimos a cada três segundos que tem o objetivo de verificar a condição em que existe pedestre na via principal ou a presença de carro na via secundária, observa-se que, a variável Sai_Comp_Cont_2 vai para zero e a variável Sai_Comp_Cont_1 vai para um cada vez que o clock de 1 Hz igual três.

Figura 13 - Simulação do Projeto RTL para via A.



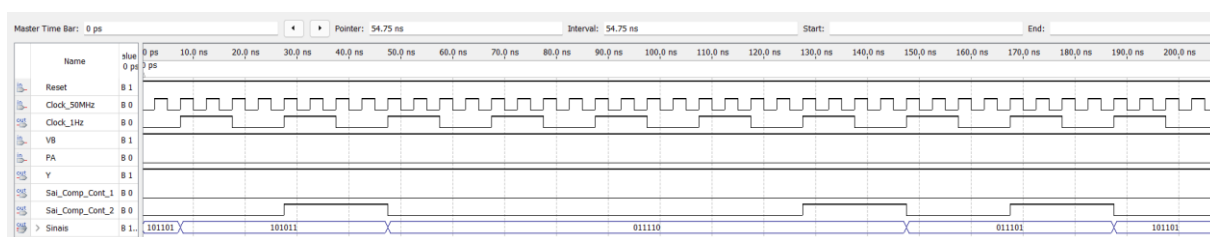
Fonte: Próprio Autor.

3.9 Simulação do Projeto RTL para via B.

A Figura 14 apresenta o resultado da simulação da condição em que houve pedestre na via principal ou a presença de carro na via secundária. Para garantir esta condição foi atribuído o valor de um para a variável VB como mostra a figura abaixo. Como a consequência a saída Y também igual a um devido à passagem destas duas variáveis pela porta OR definida no datapath. A configuração do clock é a mesma com a simulação da Figura 13.

Para verificar se ocorre a transição do estado E0 para os estados E6, E7 e E8 é mais fácil nesta simulação porque ocorre a mudança de sinais para cada estado em comparação com a simulação anterior. Como a duração de estado E6 igual a dois segundos, observa-se que, o sinal amarelo na via principal e o sinal vermelho na via secundária representado por número binário 101011 tem a duração de dois ciclos do clock de 1 Hz. A transição para o estado E7 com a duração de cinco segundos com o sinal vermelho na via principal e o sinal verde na via secundária representado por número binário 011110 tem a duração de cinco ciclos do clock de 1 Hz. Na transição de estado E8 com a duração de dois segundos onde o sinal vermelho na via principal e o sinal amarelo na via secundária representado por número binário 011101 tem a duração de dois ciclos do clock de 1 Hz.

Figura 14 - Simulação do Projeto RTL para via B.



Fonte: Próprio Autor.

4 CONCLUSÃO

Neste trabalho foi feito a modelagem de um sistema de controle de tráfego em FPGA através do projeto RTL e VHDL. Através das conexões entre os blocos operacionais e o controlador é possível formar uma inteligência capaz de gerenciar o funcionamento deste sistema de controle de tráfego. Utilizando a máquina de estado de alto nível, ASMChart, datapath e máquina de estado baixo nível é possível capturar o problema e sintetizar os blocos sequenciais e os blocos combinacionais necessários para ser codificado em VHDL. Os resultados em VHDL foram simulados e verificados, conclui-se que corresponde com o que foi modelado pelo projeto RTL.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Notas de Aula Sistemas Digitais.

[2] VAHID, Frank. Sistemas Digitais. Bookman Editora, 2009.

[3] GADAWA, Nour T.; QADDOORI, Sahar L. Design and implementation of smart traffic light controller using VHDL language. **International Journal of Engineering & Technology**, v. 8, n. 4, p. 596-602, 2019.

[4] [VHDL_5_MC_Contadores_Regs_v1 \(unicamp.br\)](#)

[5] [VHDL_2_MC_Circ_Comb_v2 \(unicamp.br\)](#)