

# Mobilidade Urbana

Angelo Marcelino Cordeiro<sup>1</sup> & Gildson Bezerra dos Santos<sup>1</sup>

<sup>1</sup>Universidade Federal do Rio Grande do Norte

## Introdução

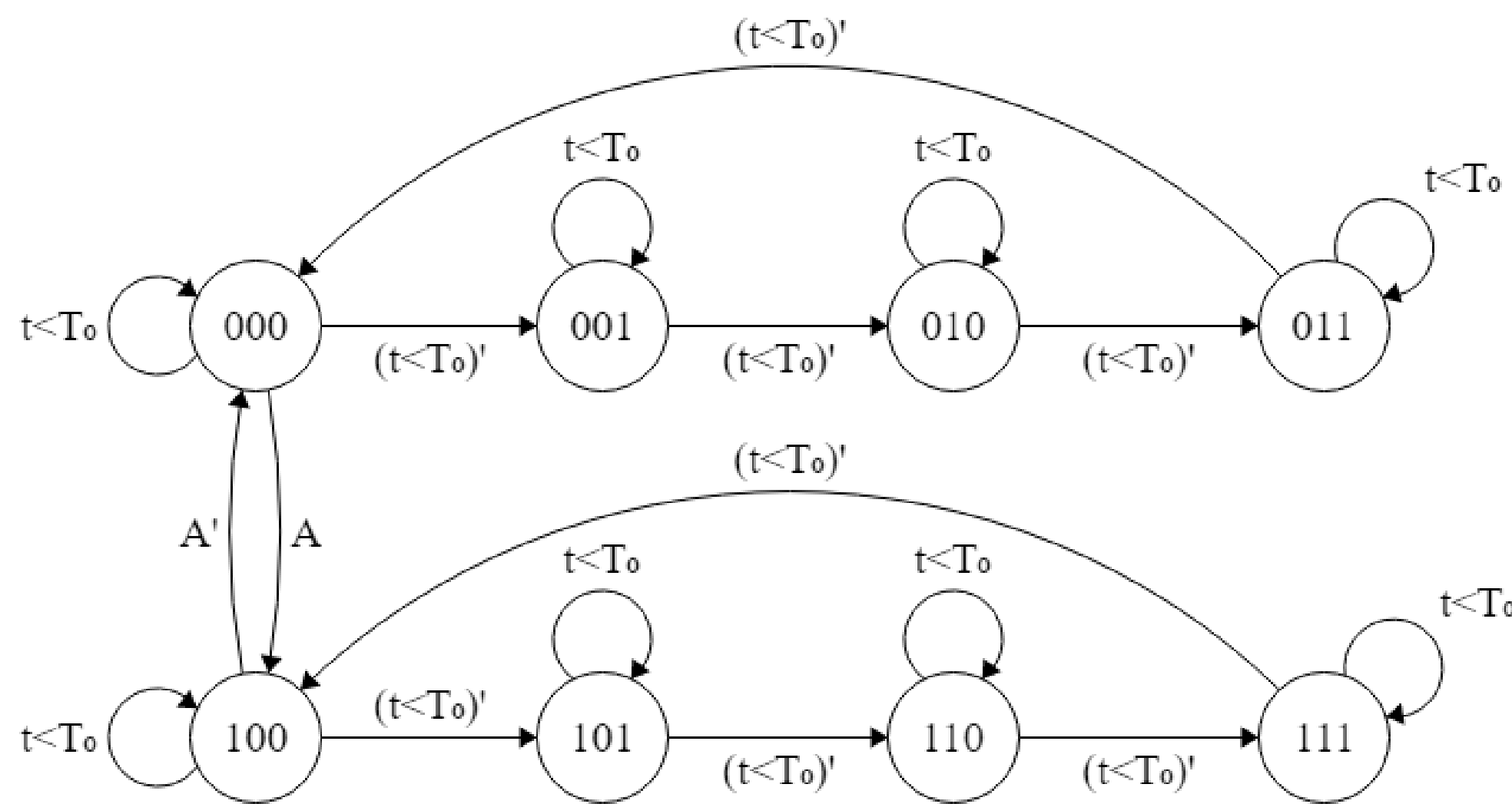
Esse trabalho consiste na implementação de um temporizador de semáforos em um cruzamento juntamente com um radar de velocidade em uma das avenidas. A descrição da atividade está disponível [aqui](#). Para a realização desse trabalho foram utilizados os conceitos de RTL, Controlador, Datapath e Máquina de Estados. Aqui apresentaremos como utilizamos esses conceitos bem como os resultados alcançados em simulações através do software Quartus II.

## Descrição da metodologia

### O Semáforo

O semáforo teve como partida a idealização da máquina de estados de alto nível representada na figura abaixo:

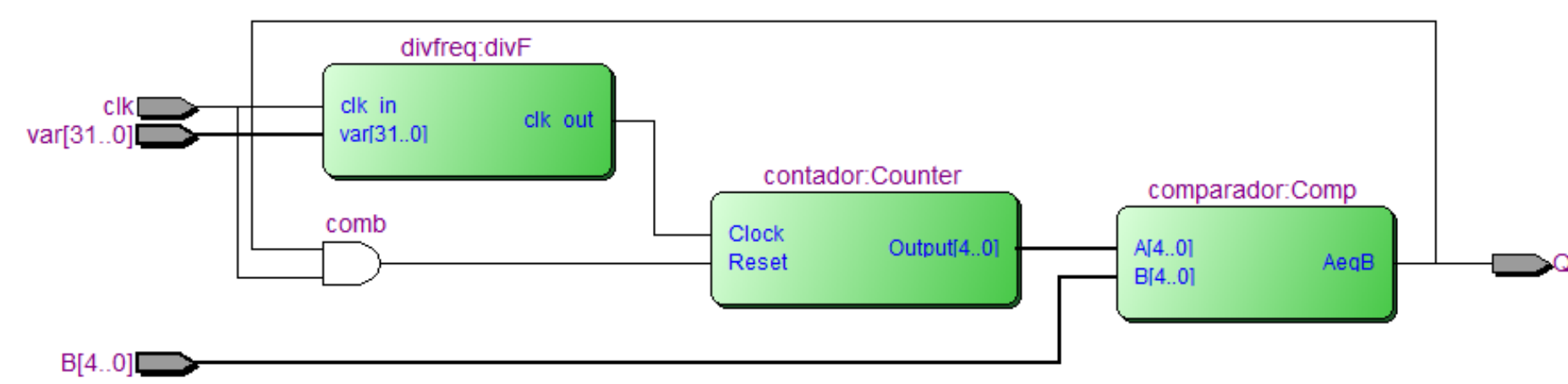
Figure 1: FSM do semáforo



Onde as saídas dos estados dependem de um comparador  $t < T_0$  que, por sua vez, depende de uma "memória" que armazena diferentes  $T_0$  para atender aos diferentes tipos de temporização dos semáforos. A entrada "A" presente na RTL, é necessária devido à mudança para a temporização que faz os semáforos só piscarem amarelo. A implementação dela também é necessária já que as saídas dos estados "100" ao "111" não são análogas às saídas do resto dos estados.

A partir disso, idealizou-se um componente Datapath que é composto de um divisor de frequência, um contador (para contar os segundos), e um comparador (que irá informar ao controlador quando  $t \geq T_0$ , fazendo os estados avançarem).

Figure 2: Datapath do semáforo



Com isso foi feito a Tabela verdade contendo os estados e saídas correspondentes:

Figure 3: Tabela Verdade do semáforo

S2	S1	S0	A	n2	n1	n0	R1	A1	G1	R2	A2	G2
0	0	0	0	0	0	1	1	0	0	0	0	1
0	0	0	1	1	0	0	1	0	0	0	0	1
0	0	1	x	0	1	0	1	0	0	0	1	0
0	1	0	x	0	1	1	0	0	1	1	0	0
0	1	1	x	0	0	0	0	1	0	1	0	0
1	0	0	0	0	0	0	0	1	0	0	1	0
1	0	0	1	1	0	1	0	1	0	0	1	0
1	0	1	x	1	1	0	0	0	0	0	1	0
1	1	0	x	1	1	1	0	1	0	0	1	0
1	1	1	x	1	0	0	0	0	0	0	0	0

Outro aspecto dessa máquina de estados foi a implementação de um MUX que seleciona dentro de si a temporização de cada estado. O funcionamento dele está descrito na Table 1.

Table 1: Mux de seleção de tempo

Switches/Estados	00	01	10	11
00	30	3	10	3
01	20	3	20	3
10	1	1	1	1
11	-	-	-	-

Assim, esse MUX precisa de 4 entradas, duas dos Switches e duas de estado. Ou seja os bits mais significantes serão as entradas  $sw_1$  e  $sw_0$  e os menos significantes serão os bits menos significantes das entradas do estado atual da FSM, que chamaremos de  $s_1$  e  $s_0$ , que também serão saídas do controlador.

Juntando todas os componentes, finalizamos a entidade semáforo que ficou composta dessa maneira e obteve os seguintes resultados:

Figure 4: Circuito do Semáforo

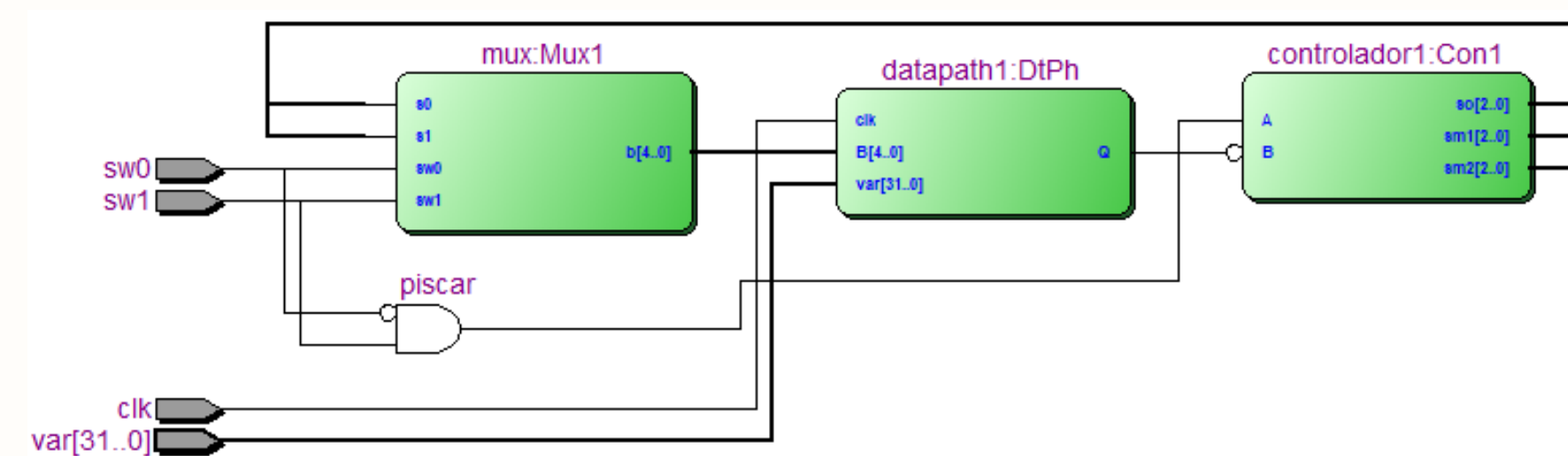


Figure 5: Waveform do Semáforo para os Modos 00 e 01

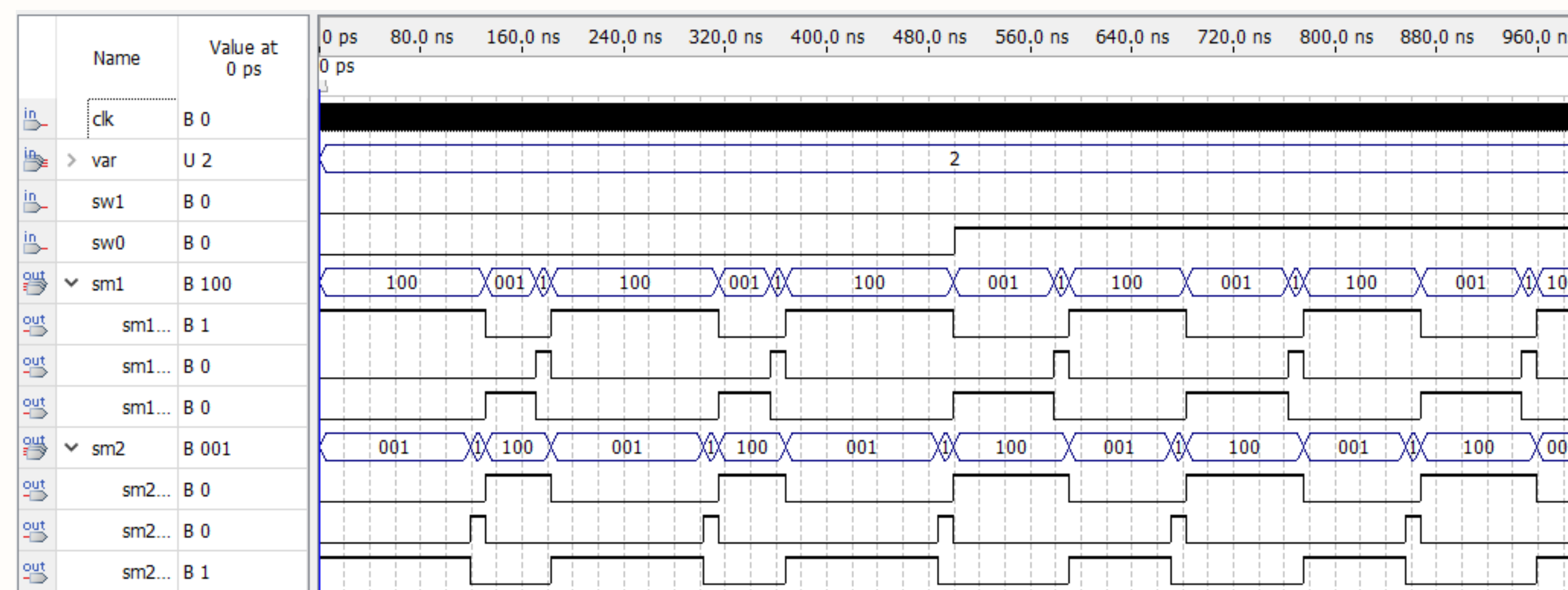
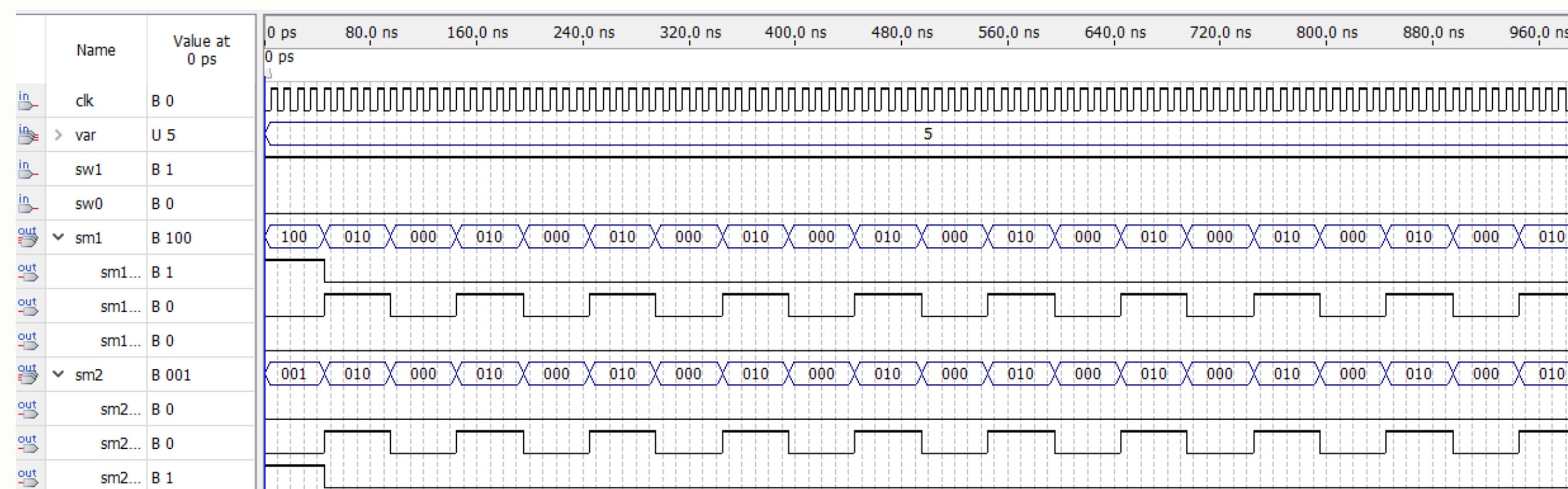


Figure 6: Waveform do Semáforo para o Modo 10



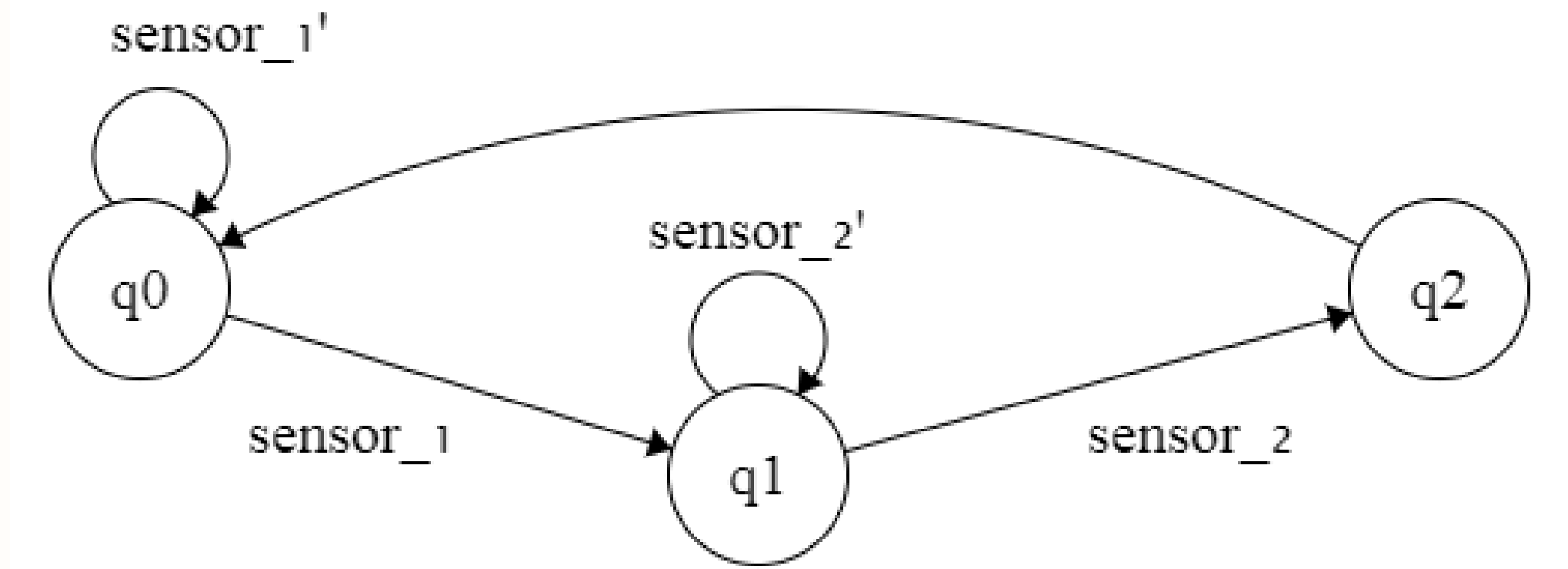
Obs.: A entrada "var" é um inteiro que serve para o divisor de frequência, ela é definida quando se escolher

o pino da FPGA que irá servir de clock. Para fins demonstrativos, dei valores baixos a essa variável, visto que a simulação de clock também é baixa.

### Radar de velocidade

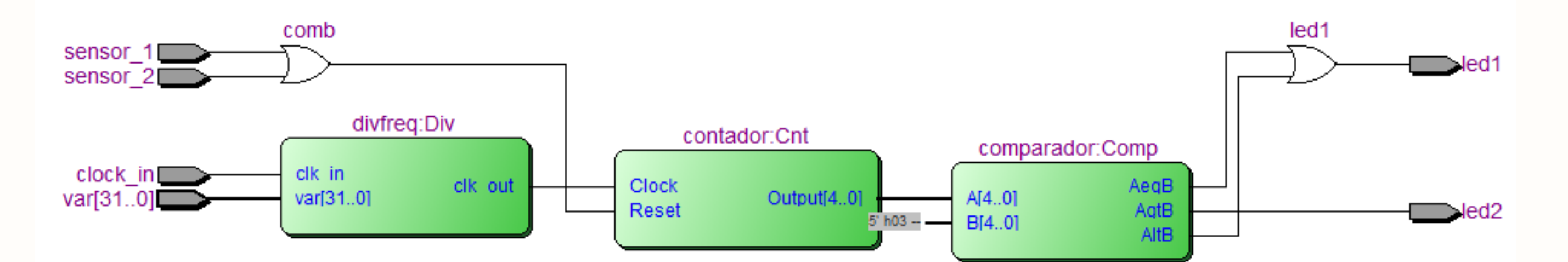
O radar foi idealizado, primeiramente com o seguinte diagrama RTL:

Figure 7: Diagrama RTL do Radar



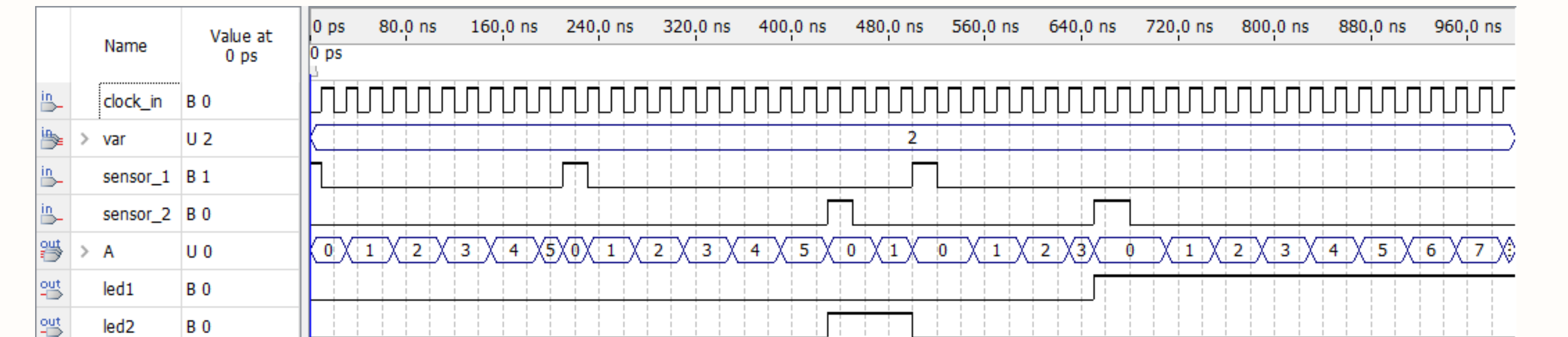
Porém, o projeto do radar foi mais simplificado, como os sensores estão separados em 30m e a velocidade máxima permitida é de 36km/h ou 10m/s, o carro precisa passar pelos dois sensores num intervalo de no mínimo 3 segundos. Precisavamos, então, implementar um contador juntamente de um comparador aliados à um divisor de frequência. Assim, o nosso radar segue o seguinte modelo:

Figure 8: Modelo dos componentes do Radar



Obs.: Esse modelo mudou com a adição da lógica que determina quando as luzes acendem ou apagam. Com isso, percebemos que o sinal mandado dependeria também de qual botão foi apertado. Para tanto, foram criados sinais auxiliares que permitiram que as luzes de "acima da velocidade" ou "na velocidade permitida" fossem acesas somente após o carro passar pelo segundo sensor e desligadas assim que o próximo carro passar pelo primeiro. Por fim foi testado o radar e obtivemos o seguinte Waveform:

Figure 9: Waveform do Radar



Obs.: O  $led_1$  representa o sinal de velocidade permitida, e o  $led_2$  representa o sinal de acima da velocidade permitida. Além disso a variável "A" foi adicionada para melhor visualização da contagem, porém não está presente no arquivo final.

## Conclusão

Por fim, juntamos ambos as entidades "semaforo" e "radar" na implementação de um novo arquivo que conseguisse ser usado no FPGA para ambas funcionalidades. Com esse trabalho conseguimos aprofundar nossos conhecimentos nos assuntos abordados além de ampliar nossa base em VHDL, linguagem tão importante para nosso período acadêmico e vidas profissionais. Os códigos fontes utilizados nesse trabalho estão disponíveis neste [link](#).