

Sistema de inspeção automatizada

Cícero Mercês da Silva¹, Igor Figueredo Soares¹, Joanderson Figueredo dos Santos¹

¹UEFS – Universidade Estadual de Feira de Santana
Av. Transnordestina, s/n, Novo Horizonte
Feira de Santana – BA, Brasil – 44036-900

cicerobrciro@gmail.com, ifs54@hotmail.com

Resumo: *Este relatório tem como objetivo descrever os passos da elaboração de um circuito combinacional, proposto pelo Módulo Integrado Projeto de Circuitos Digitais, por meio do ambiente de desenvolvimento Quartus II. Foi postulado que deveria ser implementado um sistema automatizado de inspeção de caixas plásticas em uma linha de produção da empresa Baemba. Para isso, deveria ser utilizado sensores infravermelhos na análise das condições das caixas de cada lote. Por conseguinte, cada lote deveria ser classificado, com indicações visuais em uma interface, como aceito, comprometido ou rejeitado, de acordo com os critérios do padrão estabelecido.*

1. Introdução

Ao longo dos anos, o mercado vem evoluindo de modo acelerado, fazendo com que empresas adotem ferramentas e disponham de tecnologias com o intuito de manter-se competitiva em seu meio de atuação. Um dos maiores fatores de competitividade é a excelência no que diz respeito ao padrão de qualidade na confecção de um produto.

Desta forma, para garantir um maior nível de qualidade, a padronização de processos é um fator extremamente importante e amplamente utilizado na indústria de modo geral. Com o avanço tecnológico, o processo de validação de padronização vem se tornando cada vez mais automatizado, com o auxílio de sistemas de inspeção extremamente precisos, a fim de garantir um processo de produção homogêneo quanto ao resultado final do produto.

Visando obter uma linha de produção padronizada na manufatura de caixas plásticas para um novo cliente, a empresa feirense Baemba, optou por implementar um sistema para inspecionar e garantir que toda a linha de produção destes produtos apresente conformidade com o padrão requisitado. Desta forma, são verificadas se as caixas de cada lote estão dentro do tamanho solicitado pelo cliente. Em vista disso, a Empresa Júnior de Engenharia de Computação da UEFS (ECompJR) foi contratada para o desenvolvimento deste sistema.

O sistema de inspeção automatizada deve seguir alguns requisitos e especificações, tais como, a utilização de sensores infravermelhos para a medição das caixas, uma interface intuitiva para a sinalização das condições e direcionamento dos lotes (5 caixas), alarmes indicativos e um limite máximo para um lote de até 20% de caixas com tamanho inferior ao padrão estabelecido pelo cliente.

Deste modo, exposta a sintetização supramencionada, o presente relatório tem como objetivo discorrer sobre o processo de desenvolvimento deste sistema de inspeção

automatizada. Abordando, desta forma, aspectos, como as etapas para o desenvolvimento de um projeto de circuitos digitais, de que maneira o sistema foi gerado, suas funcionalidades e os testes envolvidos para a sua validação.

2. Metodologia

O desenvolvimento do sistema de foi baseado na teoria dos circuitos digitais, utilizando como recursos o ambiente de desenvolvimento QUARTUS II em sua versão 20.1, por meio da linguagem de descrição de hardware Verilog. Além disso, foi utilizada a placa CPLD *Altera Max II-EPM240T100C5* como referência para a implementação das funções lógicas.

O sistema é dotado de um fluxo de dados, onde apresenta entradas e saídas. Dessa forma, o sistema dispõe das seguintes entradas: cinco sensores infravermelhos capazes de efetuar a medida das caixas e um botão para a ativação do sistema. Ou seja, são 6 entradas ao todo no sistema. Já as saídas são: um display de LED de 7 segmentos responsável por indicar a situação em que o lote analisado se encontra, e um alarme. O qual é acionado quando um lote é identificado como comprometido ou rejeitado. Com isso, são contabilizadas um total de 8 saídas do sistema. Além disso, é importante salientar que o sistema trabalha com níveis lógicos altos (1) e níveis lógicos baixos (0), deste modo, quando, nesse sistema, alguma saída ou entrada apresentar nível lógico alto significa que está ativada, já caso apresente nível lógico baixo significa que está desativada. Com isso, pode ser definido que, para o sistema em questão, foi adotada a convenção conhecida como lógica positiva.

3. Fundamentação Teórica

Para resolução desse problema foi necessário conhecimento sobre circuitos combinacionais, isto é, circuitos que necessitam apenas de entradas atuais para gerar uma determinada saída. Com isso, a partir da interpretação do que foi requisitado, foi posto em prática os métodos para síntese do circuito. Tais métodos, são divididos em 3 passos principais citados logo abaixo:

1. O primeiro passo é a representação do problema abordado através da tabela-verdade. A qual vem a ser definida a seguir.

A Tabela-verdade pode ser definida como um dispositivo utilizado no estudo da lógica matemática. Com o uso desta tabela é possível definir o valor lógico de uma proposição, isto é, saber quando uma sentença é verdadeira ou falsa.

Nesse sentido, a tabela-verdade foi utilizada para representar em quais condições de entradas, seria ativa (verdadeira) ou não (falsa), uma determinada saída. No problema abordado, foram utilizadas duas tabelas-verdade. Uma para mapear em quais condições de entradas seria gerada uma determinada saída, indicando a condição do lote, e outra para determinar em quais condições de saídas seria aceso ou não um determinado segmento do display, ilustrando por um caractere a condição do lote.

2. O segundo passo é a obtenção de uma função lógica a partir da tabela-verdade.

Uma função lógica é uma função similar à que utilizamos na matemática, a qual a partir de uma determinada entrada, obtemos uma saída. Entretanto, na função lógica, uma variável assume apenas dois estados (verdadeiro) ou (falso).

Dentre as principais funções lógicas temos: NÃO (NOT), E (AND), OU (OR), NÃO-E (NAND), NÃO-OU (NOR), OU EXCLUSIVO (XOR).

Para implementação das funções lógicas citadas acima, utilizamos um conjunto de circuitos básicos denominados de portas lógicas.

Em posse da tabela-verdade podemos encontrar a função para cada saída do circuito. Para isso, foi utilizado o mapa de Karnaugh. Isso porque, com a utilização desse método conseguimos obter a expressão lógica da forma mais simplificada possível. Consequentemente, reduzindo a complexidade do circuito.

3. Por fim, Obtenção do circuito lógico a partir da função lógica.

Tendo a função lógica de cada saída, podemos fazer a representação (construção) do circuito. Para isso, foi utilizado a linguagem de descrição de hardware (HDL - Hardware Description Language). Que é a linguagem textual usada para descrever o circuito a ser sintetizado em uma FPGA(Field Programmable Gate Array, em português, "Arranjo de portas Programáveis em Campo"). Dentre as linguagens mais comuns, a que foi utilizada para resolução do problema foi Verilog.

Além disso, como citado anteriormente, os dados são obtidos por meio de sensores infravermelhos ativos, posteriormente o sistema valida se o lote está dentro dos padrões estabelecidos ou não. Por fim, uma interface composta por um display de LED de 7 segmentos exibe a informação referente ao lote analisado. Caso o lote não apresente conformidade com o padrão estabelecido um alarme é acionado.

3.1 Sensores Infravermelhos ativos

Os sensores infravermelhos ativos (IVA) são constituídos de um transmissor e um receptor de radiação infravermelha. Onde o transmissor emite um sinal de luz no espectro infravermelho e o receptor capta essa radiação após o feixe ser refletido em uma superfície. De modo geral, este tipo de sensor apresenta um potenciômetro ajustável para controle da sensibilidade da distância de detecção.

Desse modo, caso não haja a reflexão da luz infravermelha em alguma superfície, impossibilitando o receptor infravermelho de detectar o feixe emitido, esse acontecimento é reportado por meio de um sinal pelo próprio sensor. Visto

isso, caso o receptor detecte o feixe emitido, um outro sinal é fornecido pelo sensor infravermelho.

3.2 Display de 7 segmentos

Um display de 7 segmentos é um componente eletrônico que permite exibir números, letras e símbolos apagando-se ou acendendo-se segmentos de LEDs.

3.3 Decodificador

O decodificador tem a função de converter uma informação codificada, como um número binário, numa forma não codificada, como a forma de um número decimal ou um caractere do alfabeto. Sendo assim, um decodificador é um circuito combinacional que converte uma entradas binárias de n bits, em no máximo 2^n saídas. Onde cada saída corresponde a uma informação representada pelo número binário de entrada. Por exemplo, se tivermos um número binário de 2 bits como entrada em um decodificador, podemos ter no máximo 4 saídas. Onde apenas uma saída será ativa por vez.

4. Resultados e Discussões

Como citado anteriormente o sistema de inspeção, consiste basicamente em um circuito combinacional, o qual com base em determinadas entradas representadas pelos sensores, vai gerar uma determinada condição de saída. Essa, que por sua vez, vai indicar a situação do lote, podendo ser rejeitado, comprometido ou aceito. Dessarte, foi posto em prática a definição dos tipos de lotes.

4.1 Definição das condições dos lotes

Inicialmente, para projetar o sistema, foi necessário definir as situações previstas dos lotes, tais como, lote aceito, comprometido e rejeitado, levando em consideração a padronização estabelecida. Como definido nas discussões, um lote é composto de 5 caixas plásticas, onde cada sensor analisa uma caixa diferente. Além disso, foi determinado que o limite máximo permitido de caixas fora do padrão em um dado lote seria de 20%, ou seja, 1 caixa fora do padrão.

Em vista disso, foi definido que um lote aceito, é aquele que dispõe de todas as 5 caixas dentro do padrão estabelecido. Enquanto que, se 20% do lote apresentar dimensão fora do padrão, o lote é classificado como comprometido. Assim sendo, um lote comprometido é aquele que está dentro da margem de erro estipulada, porém, ainda assim, apresenta uma caixa fora do padrão. Por outro lado, um lote rejeitado é aquele que extrapola o limite estabelecido de caixas fora do padrão, ou seja, caso um lote apresente duas ou mais caixas com dimensões incompatíveis com o padrão solicitado, o lote em questão é classificado como rejeitado.

Determinado como cada lote seria etiquetado, com base no número de caixas que estão de acordo, ou não, com tamanho requisitado pelo cliente. Foi posto em prática o primeiro passo para a síntese de um circuito combinacional. Sendo assim, foi feita a primeira tabela-verdade do circuito, representada na Tabela 1.

Tabela 1: Tabela-verdade para cada definição do tipo de lote.

| Entradas | | | | | Saídas | | |
|----------|---------|---------|---------|---------|--------|--------------|-----------|
| SENSOR1 | SENSOR2 | SENSOR3 | SENSOR4 | SENSOR5 | ACEITO | COMPROMETIDO | REJEITADO |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Fonte: Próprio Autor.

Existem 5 entradas provenientes dos sensores, portanto tem-se 2^5 possibilidades de entrada de dados, ou seja, 32 possibilidades. Além disso, é evidenciado que a partir dessas 32 combinações são geradas 3 possíveis saídas, sendo elas lote aceito, lote comprometido e lote rejeitado.

Por conseguinte, foi posto em prática o segundo passo para a síntese de um circuito combinacional, que como citado anteriormente, é a obtenção da função lógica de cada saída a partir da tabela-verdade. Segue abaixo a expressão lógica de cada saída, onde S1, S2, S3, S4 e S5 são os sensores:

Expressão do lote Aceito: $A = F(S1, S2, S3, S4, S5) = S1.S2.S3.S4.S5$

Expressão do lote Comprometido : $C = F(S1, S2, S3, S4, S5) = (S1'.S2.S3.S4.S5) + (S1.S2'.S3.S4.S5) + (S1.S2.S3'.S4.S5) + (S1.S2.S3.S4'.S5) + (S1.S2.S3.S4.S5')$

Expressão do Rejeitado: $R = F(A, C) = (A'.C')$

Em seguida, foi posto em prática o terceiro passo para a síntese de um circuito combinacional, que é a sua elaboração. Elaboração essa que vai ser detalhada nas seções abaixo.

4.2 Módulos do circuito

O sistema foi implementado utilizando diversos módulos com a finalidade de deixar o circuito mais modular possível. Desta maneira, temos uma maior organização da descrição do hardware. Além de permitir testes mais precisos, já que os módulos podem ser testados individualmente e em conjunto. Abaixo estão descritos os módulos implementados:

1. Módulo *controlOutputAC*: O módulo em questão, é responsável por analisar se um determinado lote está aceito, ou seja, verificar se as 5 caixas examinadas estão dentro do padrão estabelecido pelo cliente. Este módulo possui 5 entradas provenientes dos sensores e pode retornar em sua saída 1, caso o lote esteja dentro do padrão, ou 0, caso contrário.
2. Módulo *controlOutputCO*: Este módulo é encarregado de analisar se o lote sondado está comprometido, ou seja, o módulo verifica as cinco peças e caso somente uma delas apresente tamanho fora do padrão, sua saída retorna 1, porém caso nenhuma caixa ou mais de uma caixa apresente medidas fora do estipulado, a sua saída retorna 0.
3. Módulo *controlOutputRE*: Avalia se o lote apresenta as características de rejeitado, isto é, caso mais de uma caixa do lote apresente inconformidade quanto ao seu tamanho em relação ao estabelecido, o módulo retorna em sua saída 1, porém caso contrário o módulo retorna 0 em sua saída. Para

essa análise, o módulo recebe em suas duas entradas, as saídas dos módulos *controlOutputAC* e *controlOutputCO*. Posteriormente, por meio de seus componentes lógicos internos, avalia o lote. Se suas duas entradas forem 0, a sua saída será 1, porém caso uma das duas entradas seja 1, sua saída será 0.

4. Módulo *controlOutputDis*: O módulo é um decodificador, responsável pelas entradas que vão ser conectadas no display de 7 segmentos. Este módulo recebe como entrada, as saídas dos módulos *controlOutputAC*, *controlOutputCO*, *controlOutputRE* e o botão de habilitação do sistema. O referido módulo apresenta 7 saídas, as quais apresentam nível lógico alto de acordo com o estado do lote analisado, e do pressionamento do botão. Nesse sentido, caso o lote avaliado seja indicado pelos módulos de análise como aceito, as saídas referentes aos segmentos que formam a letra “A” no display ganham *status* de nível lógico alto (1) assim que o botão de habilitação é acionado. Já caso a avaliação dos módulos indique que o lote apresenta-se comprometido, as saídas que referem-se aos segmentos que formam a letra “C” no display recebem como valor o nível lógico alto após a ativação do botão de habilitação. Por fim, caso o lote seja indicado como rejeitado, as saídas atribuídas aos segmentos que compõem a letra “r” no display conquistam o sinal de nível lógico alto, posteriormente ao acionamento do botão habilitador, enquanto as outras saídas permanecem em nível lógico baixo.
5. Módulo *controlOutputAL*: Este módulo é responsável por acionar o alarme caso o lote seja definido como comprometido ou rejeitado pelos módulos verificadores. O *controlOutputAL*, recebe como entradas as saídas do módulo *controlOutputAC* e o sinal do botão de habilitação. Caso a saída do módulo *controlOutputAC* apresente nível lógico baixo, ou seja, igual a 0, o alarme é acionado assim que o botão de habilitação for ativado. Dessa forma, caso o lote verificado apresente o *status* de aceito, a saída deste módulo apresentará nível lógico baixo, caso contrário, apresentará nível lógico alto assim que o botão for ligado.
6. Módulo *mainCircuit*: O módulo em questão é responsável por instanciar todos os outros módulos do sistema. Este é o nosso módulo principal, o qual está encarregado de unir todos os outros módulos descritos posteriormente.

4.3 Funcionamento do display

Como citado anteriormente, o display foi utilizado para apresentar a condição de estado referente a um lote. Dessarte, como foi utilizado apenas um display, só foi possível representar 1 caractere por vez. Portanto, se quiséssemos exibir a condição do lote aceito, rejeitado ou comprometido, com todos os caracteres da

palavra não seria possível. Assim sendo, foi mapeado quais segmentos deveriam estar em nível lógico ALTO ou não, com a finalidade de exibir cada caractere desejado. O mapeamento é exibido abaixo.

Caractere A: segmentos em nível lógico alto (a, b, c, e, f, g);

Caractere C: segmentos em nível lógico alto (a, d, e, f);

Caractere r: segmentos em nível lógico alto (a, b, e, f);

Além disso, foi necessário a utilização de uma tabela-verdade para definir a função lógica de cada segmento como ilustrado na tabela 2.

Tabela 2: Tabela-verdade para a definição dos segmentos a serem utilizados.

| Entradas | | | | Saídas | | | | | | |
|----------|----|----|----|--------|---|---|---|---|---|---|
| EN | AC | CO | RE | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Fonte: Próprio Autor.

Depois da elaboração da tabela-verdade foi feita a expressão lógica para cada segmento do display. Ilustradas logo abaixo:

Segmento a: $a = F(AC, CO, RE, EN) = (AC'.CO'.RE + AC'.CO.RE' + AC.CO'.RE').EN$

Segmento b: $b = F(AC, CO, RE, EN) = (AC'.CO'.RE + AC.CO'.RE').EN$

Segmento c: $c = F(AC, CO, RE, EN) = (AC.CO'.RE').EN$

Segmento d: $d = F(AC, CO, RE, EN) = (AC'.CO.RE').EN$

Segmento e: $e = F(a) = (a).EN$

Segmento f: $f = F(a) = (a).EN$

Segmento g: $g = F(c) = (c).EN$

Por fim, seria ideal utilizar mais display de 7 segmentos, ou uma matriz de LED, para apresentar de forma mais lúcida a condição do lote ao usuário.

4.4 Recursos utilizados da CPLD

O circuito no total contém 14 pinos. 6 de entradas e 8 de saídas. Os pinos de entradas foram chaves e os de saídas foram os segmentos do display mais o buzzer.

A quantidade total de LEs utilizados foi 8. Onde 4 LEs, utilizaram as 4 entradas de uma LUT, 2 LEs utilizaram 3 entradas e os outros 2 utilizaram apenas duas entradas. Além disso, todos os LEs funcionaram no modo normal.

Os LEs são listados:

1. Full Name: lmainCircuitlcontrolOutputCO:cptd_instlOr0~2
 - 1.1. Location: LC_X4_Y2_N2
 - 1.2. Sum LUT Mask: 2800
 - 1.3. Sum Equation: $A \& D \& (B \$ C)$
2. Full Name: lmainCircuitlcontrolOutputDis:dcfd_instlAnd13
 - 2.1. Location: LC_X4_Y2_N3
 - 2.2. Sum LUT Mask: AAA0
 - 2.3. Sum Equation: $A \& (C \# D)$
3. Full Name: lmainCircuitlcontrolOutputDis:dcfd_instlAnd11~0
 - 3.1. Location: LC_X4_Y2_N4

- 3.2. Sum LUT Mask: 0300
- 3.3. Sum Equation: $!B \& !C \& D$
- 4. Full Name: lmainCircuitlcontrolOutputCO:cptd_instlOr0~3
 - 4.1. Location: LC_X4_Y2_N5
 - 4.2. Sum LUT Mask: 6080
 - 4.3. Sum Equation: $C \& (A \& (B \$ D) \# !A \& B \& D)$
- 5. Full Name: lmainCircuitlcontrolOutputAL:comb_3lAL~0
 - 5.1. Location: LC_X4_Y2_N6
 - 5.2. Sum LUT Mask: 7F00
 - 5.3. Sum Equation: $D \& (!C \# !B \# !A)$
- 6. Full Name: lmainCircuitlcontrolOutputCO:cptd_instlOr0~1
 - 6.1. Location: LC_X4_Y2_N7
 - 6.2. Sum LUT Mask: C0C0
 - 6.3. Sum Equation: $B \& C$
- 7. Full Name: lmainCircuitlcontrolOutputDis:dcfd_instlAnd12
 - 7.1. Location: LC_X4_Y2_N8
 - 7.2. Sum LUT Mask: 8000
 - 7.3. Sum Equation: $A \& B \& C \& D$
- 8. Full Name: lmainCircuitlcontrolOutputCO:cptd_instlOr0~0
 - 8.1. Location: LC_X4_Y2_N9
 - 8.2. Sum LUT Mask: F000
 - 8.3. Sum Equation: $C \& D$

Os Logic Array Block (LAB), as distribuição dos LEs e os pinos representados no *Chip Planner* podem ser visualizados na figura 1.

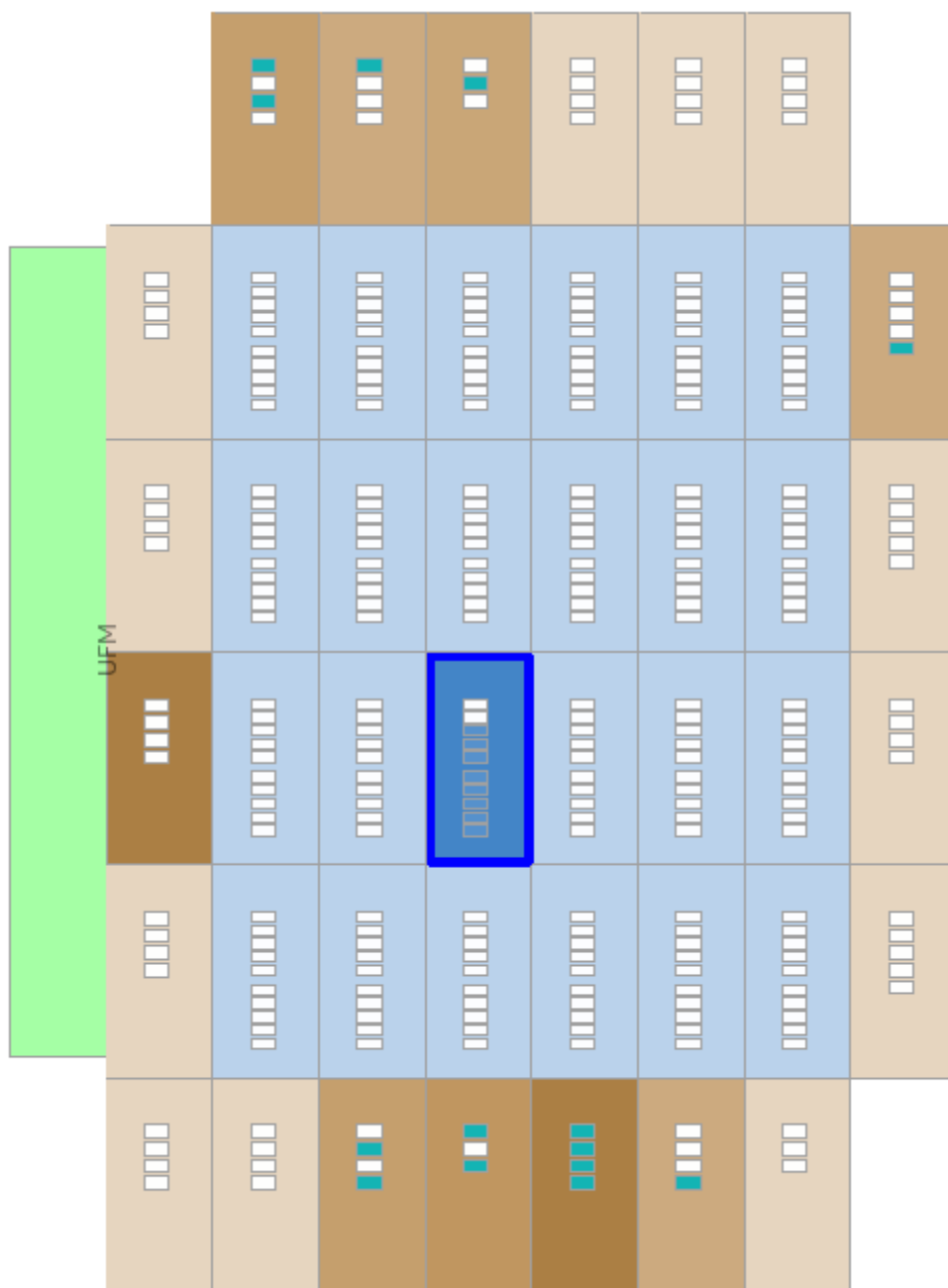


Figura 1. Representação dos pinos, LABs e LEs por meio do *Chip Planner*.

4.5 Representação do sistema em alto nível

O sistema apesar de simples, carece de ser representado de um modo generalista, onde é possível visualizar seus periféricos e módulos, sem muitos detalhes técnicos. Isso fornece uma visão geral do que se trata o sistema, permitindo uma compreensão melhor de todo o conjunto que o compõe. A figura 1 ilustra o sistema nessas condições, em alto nível, possibilitando esta visão mais ampla.

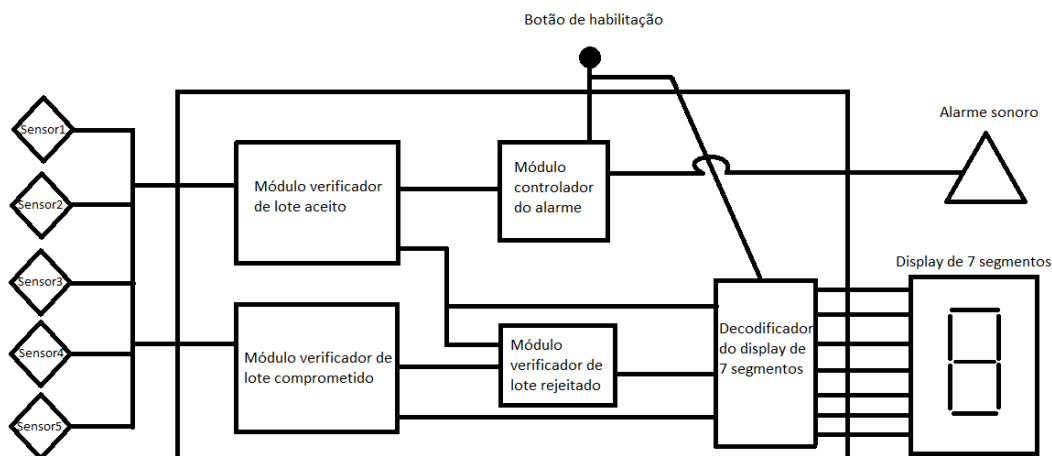


Figura 2. Ilustração do sistema em alto nível.

4.6 Testes

O ambiente de desenvolvimento Quartus II na versão 20.1, dispõe de aplicações integradas que permitem o desenvolvimento e execução de rotinas de testes, possibilitando, dessa forma, efetuar os testes para as verificações de funcionamento do sistema desenvolvido.

Os testes foram efetuados por meio da ferramenta Simulation Waveform Editor. Os testes foram feitos módulo a módulo, além do teste do sistema como um todo, constatando dessa forma, as suas funcionalidades.

Foi testado o módulo *controlOutputAC*, responsável por verificar se o lote está aceito, ou seja, os resultados esperados para o teste desse módulo são: Caso todas as entradas do módulo apresente nível lógico alto, a sua saída deve apresentar nível lógico alto. Porém, caso alguma de suas entradas disponha de um nível lógico baixo, a saída do módulo deve dispor de um nível lógico baixo.

Além disso, o módulo *controlOutputCO* também foi testado. Este módulo tem a função de verificar se o lote analisado está comprometido. Desse modo, os resultados esperados dos testes deste módulo são: caso o quatro de suas cinco entradas apresentem nível lógico alto, a sua saída deve dispor de um nível lógico alto. Todavia, se todas as cinco entradas ou menos de quatro entradas apresentarem nível lógico alto, a saída deste módulo deve emitir sinal de nível lógico baixo.

Também foi testado o módulo *controlOutputRE*, responsável por diagnosticar se um determinado lote está rejeitado. Este módulo recebe duas entradas, uma sendo a saída do módulo *controlOutputCO* e a outra a saída do módulo *controlOutputAC*. Neste caso, o teste para o módulo em questão deve apresentar os seguintes resultados: se as entradas do módulo apresentarem nível lógico baixo, a sua saída deve ter nível lógico alto. Contudo, se o módulo receber nível lógico alto em uma das duas entradas, ou ainda, nas duas entradas, a sua saída terá nível lógico baixo.

Ademais, o módulo *controlOutputDis*, responsável por ser o decodificador e controlar, dessa forma, o display de sete segmentos, também foi testado. Este módulo dispõe de quatro entradas sendo elas: entrada proveniente da saída do módulo *controlOutputAC*, a entrada que recebe a saída do módulo *controlOutputCO*, entrada que tem origem da saída do módulo *controlOutputRE* e a entrada *button* que representa o botão de habilitação do sistema. Por outro lado, o módulo apresenta sete saídas, cada uma responsável por um dos segmentos do display. Visto isso, os resultados provenientes do teste deste módulo devem ser: caso a entrada vinda da saída do módulo *controlOutputAC* tenha nível lógico alto, o botão de habilitação também apresentar nível lógico alto e as outras entradas apresentarem nível lógico baixo, as saídas referentes aos segmentos que compõem a formação da letra “A” no display devem apresentar nível lógico alto. Já caso a entrada proveniente da saída do módulo *controlOutputCO* tenha nível lógico alto, a entrada *button* também e as outras saídas tenham nível lógico baixo, as saídas do módulo responsáveis pelos segmentos que formam a letra “C” no display devem apresentar nível lógico alto, enquanto as outras devem apresentar nível lógico baixo. Porém, caso a entrada que tem origem na saída do módulo *controlOutputRE* disponha de um nível lógico alto, o botão de habilitação também e as demais entradas apresentem nível lógico baixo, as saídas do módulo encarregadas pelos segmentos que forma a letra “r” no display devem ter nível lógico alto, enquanto as outras saídas devem ter nível lógico baixo.

Além do mais, o módulo *controlOutputAL* também foi testado. Este módulo é responsável por acionar o alarme caso o lote seja avaliado como comprometido ou rejeitado. o *controlOutputAL* é dotado de duas entradas: uma provinda do botão de habilitação (*button*) e outra vinda da saída do módulo *controlOutputAC*. Além disso, o módulo apresenta uma saída responsável por ativar o alarme do sistema. Desta maneira, os resultados esperados para o teste deste módulo são: caso a entrada originada da saída do módulo *controlOutputAC*

apresente nível lógico baixo e o botão apresente nível lógico alto, a sua saída deve mandar um sinal de nível lógico alto. Já nos demais casos para o módulo, a sua saída deve ter um nível lógico baixo.

Por fim, o sistema como um todo foi testado. Neste caso, tem-se seis entradas e oito saídas. Desta forma, os resultados esperados do teste do sistema como um todo são:

1. Se todas as entradas referentes aos sensores e o botão apresentarem nível lógico alto, as saídas responsáveis por acender os segmentos do display para a formação da letra “A”, devem apresentar nível lógico alto e o restante nível lógico baixo.
2. Já se apenas uma das entradas relativas aos sensores apresentar nível lógico baixo e as demais, junto ao botão, apresentarem nível lógico alto, as saídas atribuídas para a ativação dos segmentos do display para a formação da letra “C” e a saída de ativação do alarme, deverão apresentar nível lógico alto, enquanto as demais nível lógico baixo.
3. Além disso, se mais de uma entrada dos sensores apresentar nível lógico baixo e o botão for acionado (nível lógico alto), as saídas para a formação da letra “r”, por meio dos segmentos do display, e a saída do alarme, recebem nível lógico alto, já o restante das saídas devem dispor de nível lógico baixo.
4. Por fim, caso ocorra outras combinações de entradas que não as mencionadas acima, as saídas do sistema devem apresentar nível lógico alto.

4.6.1 Teste do módulo de validação de lote aceito (*controlOutputAC*)

Inicialmente foi testado o módulo responsável em verificar se o lote encontra-se aceito. Primeiramente, foi executado o teste funcional do sistema e, posteriormente, o teste temporal, como ilustram as figuras 3 e 4.

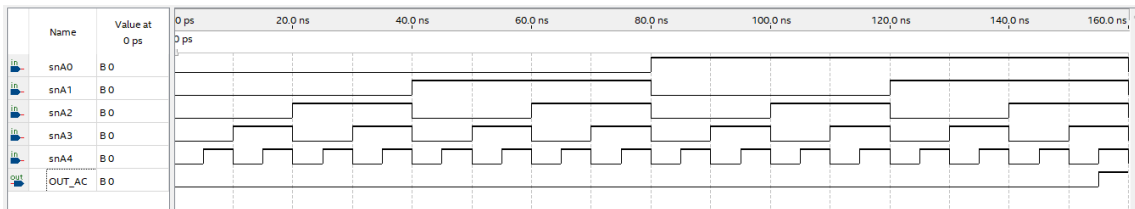


Figura 3. Teste funcional do módulo *controlOutputAC*.

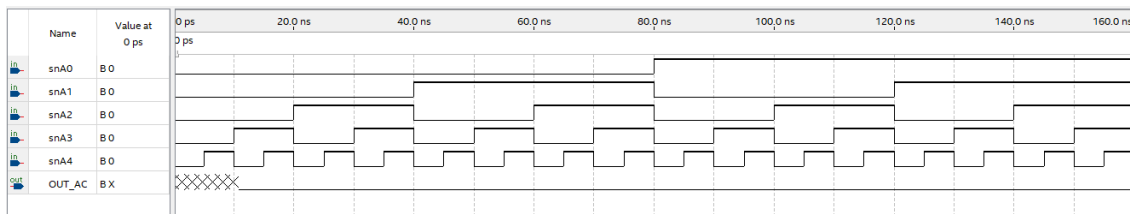


Figura 4. Teste temporal do módulo *controlOutputAC*.

Com isso, pode ser observado na Figura 3 que o módulo testado só apresenta em sua saída valor lógico alto quando todas as suas entradas apresentarem níveis lógicos altos, ou seja, o módulo desempenha sua função da forma esperada. Visto que, quando todas as caixas analisadas estão dentro do padrão os sensores apresentam nível lógico alto, portanto, o módulo só deve ter em sua saída nível lógico alto quando todas as suas entradas forem 1 (nível lógico alto).

Já na simulação temporal, representada na imagem 4, pode ser percebido um atraso de propagação de 10,851 ns.

4.6.2 Teste do módulo de validação de lote comprometido (*controlOutputCO*)

Posteriormente, foi testado o módulo responsável pela verificação se um determinado lote está comprometido. Inicialmente foi feito o teste funcional do módulo como ilustra a figura 5 e, posteriormente, foi executado o teste temporal como é apresentado na figura 6.

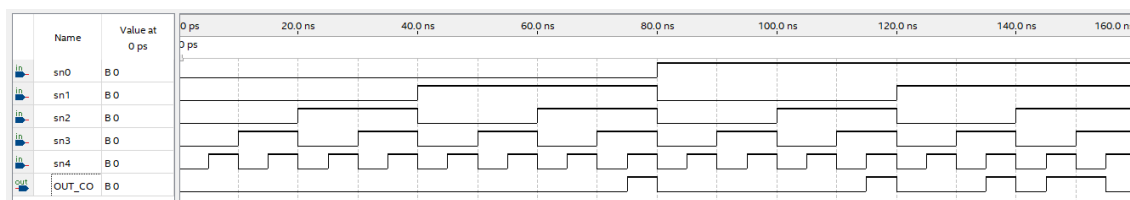


Figura 5: Teste funcional do módulo *controlOutputCO*.

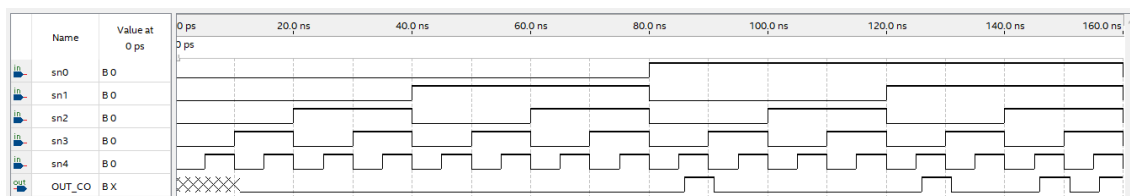


Figura 6: Teste temporal do módulo *controlOutputCO*.

Pode ser observado na figura 5 que o módulo apresenta saída com nível lógico alto quando 4 de suas entradas dispõem de nível lógico alto. Desse modo, fica evidente que o módulo funciona da forma desejada, visto que sua atribuição

é justamente expressar em sua saída um nível lógico alto somente quando 4 das 5 caixas do lote apresentarem conformidade com o padrão de tamanho adotado, ou seja, só deverá reportar 1 em sua saída quando 4 dos 5 sensores apresentarem nível lógico alto.

Já na figura 6, podemos visualizar na simulação temporal do módulo um atraso de propagação na sua saída de 10,966 ns.

4.6.3 Teste do módulo de validação de lote rejeitado (*controlOutputRE*)

Em seguida, foi testado o módulo responsável por avaliar se um lote deve ser classificado como rejeitado. Primeiramente foi efetuado o teste funcional como ilustra a figura 7 e na posteridade, foi executado o teste temporal como apresentado na figura 8.

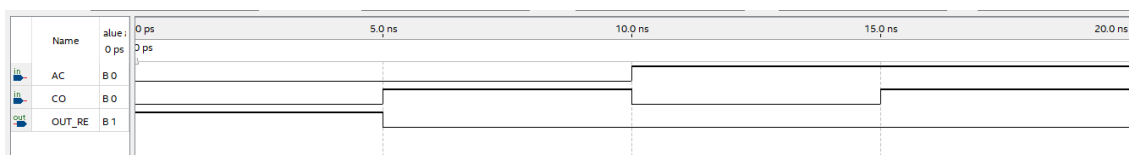


Figura 7: Teste funcional do módulo *controlOutputRE*.

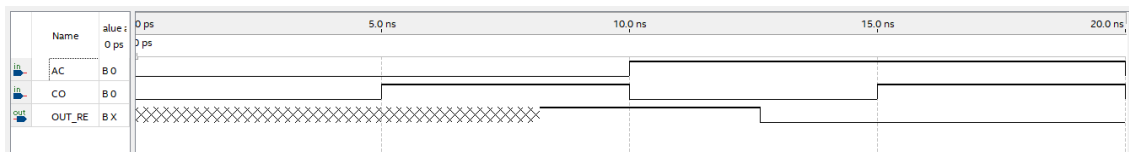


Figura 8: Teste temporal do módulo *controlOutputRE*.

É evidenciado na figura 7 que o módulo só apresenta nível lógico alto quando as suas entradas estão em nível lógico baixo. Isso ocorre porque esse módulo é responsável por verificar as saídas dos módulos *controlOutputCO* e *controlOutputAC*, dessa forma, quando os dois módulos apresentam como saída nível lógico baixo, significa que o lote analisado não foi classificado como aceito, nem comprometido, portanto deve ser classificado como rejeitado. Esse é o funcionamento esperado para o módulo em questão.

Já no teste temporal do módulo, representado na figura 8, pode ser verificado um atraso de propagação de 8,206 ns.

4.6.4 Teste do módulo decodificador do display (*controlOutputDis*)

Na posterioridade, foi efetuado o teste do módulo responsável por controlar o display de 7 segmentos. De forma inicial, foi feito o teste funcional do módulo e

na sequência, foi executado o teste temporal, como apresentado nas figuras 9 e 10, respectivamente.

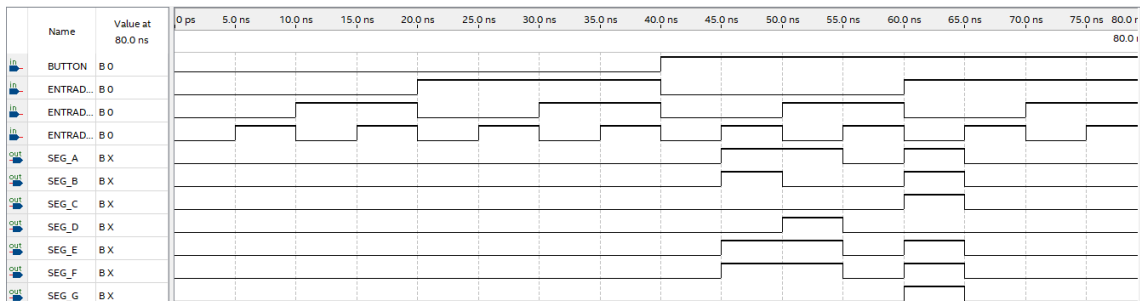


Figura 9: Teste funcional do módulo *controlOutputDis*.

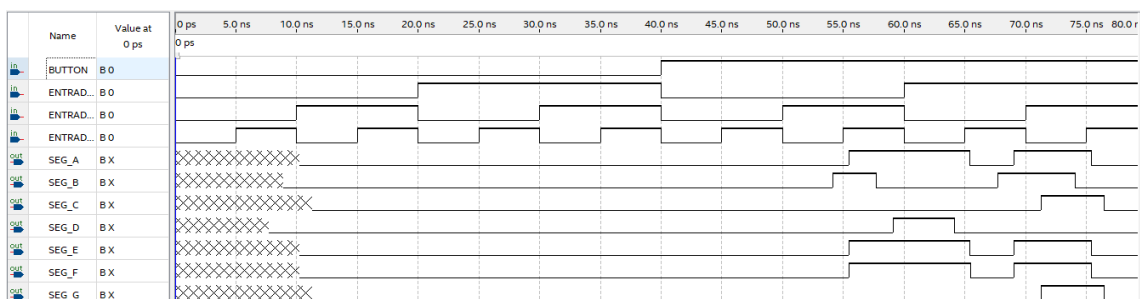


Figura 10: Teste temporal do módulo *controlOutputDis*.

Com base no teste funcional ilustrado na figura 9, pode ser percebido que o módulo apresenta como saídas os segmentos do display. Dessa forma, como é possível visualizar no teste, caso o botão seja ativado (nível lógico alto) e, além disso, apresenta nível lógico alto na entrada no decodificador referente ao lote aceito e as outras entradas apresentem nível lógico baixo, os segmentos do display relativos a letra que indica o lote aceito são ligados. O mesmo ocorre quando cada uma das outras entradas, referentes aos lotes comprometidos e rejeitados, estão em nível lógico alto, junto ao botão e com as demais entradas em nível lógico baixo, porém os segmentos ligados do display são os referentes às respectivas situações do lote analisado.

Já no teste temporal do módulo, o qual é representado na figura 10, pode ser verificado um atraso de propagação de: 10,731 ns na saída *SEG_A*, 9 ns na saída *SEG_B*, 11,2 ns na saída *SEG_C*, 7,77 ns na saída *SEG_D*, 7,77 ns na saída *SEG_E*, 10,731 ns na saída *SEG_F* e 11,12 ns na saída *SEG_G*

4.6.5 Teste do módulo do alarme (*controlOutputAL*)

Sucessivamente, foi efetuado o teste do módulo responsável por controlar o acionamento do alarme. De início o teste funcional foi executado, o qual está

representado na figura 11. Posteriormente, o teste temporal, representado na figura 12, foi efetuado.

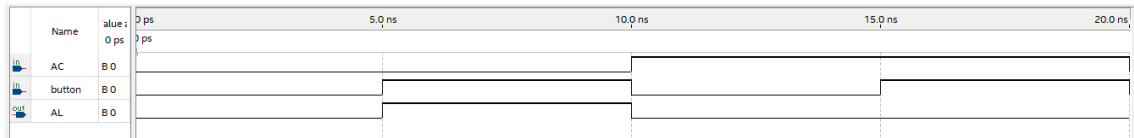


Figura 11: Teste funcional do módulo *controlOutputAL*.

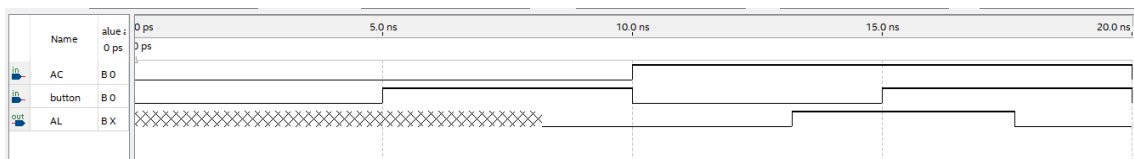


Figura 12: Teste temporal do módulo *controlOutputAL*.

Pode ser observado no teste funcional, ilustrado na figura 11, que o módulo apresenta nível lógico alto em sua saída somente quando a entrada *button* (entrada habilitadora) estiver nível lógico alto e a entrada *AC* estiver em nível lógico baixo. Isso ocorre porque a entrada *button* é a responsável pela habilitação do módulo e a entrada *AC* provém do módulo verificador de lote aceito. Desse modo, o alarme deve ser ativado - saída em nível lógico alto - somente quando o botão habilitador for ativado e um lote for avaliado como comprometido ou rejeitado, portanto, quando *AC* apresenta nível lógico alto as condições, lote rejeitado ou comprometido, não são verdadeiras.

Já no teste temporal, exibido na figura 12, pode ser constatado um atraso de propagação na saída *AL* de 8,206 ns.

4.6.6 Teste de todo o sistema

Por fim, foi efetuado o teste funcional e o teste temporal de todo o sistema, como todos os módulos integrados, como as figuras 13 e 14 ilustram.

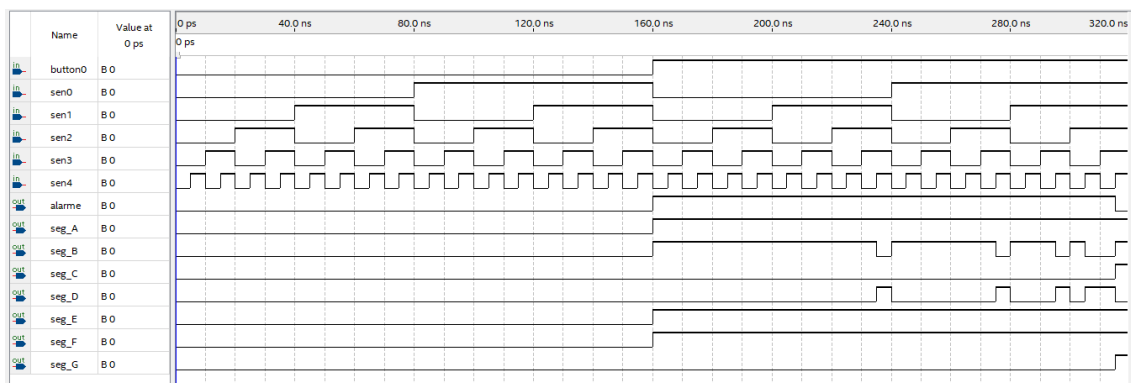


Figura 13: Teste funcional do sistema completo.

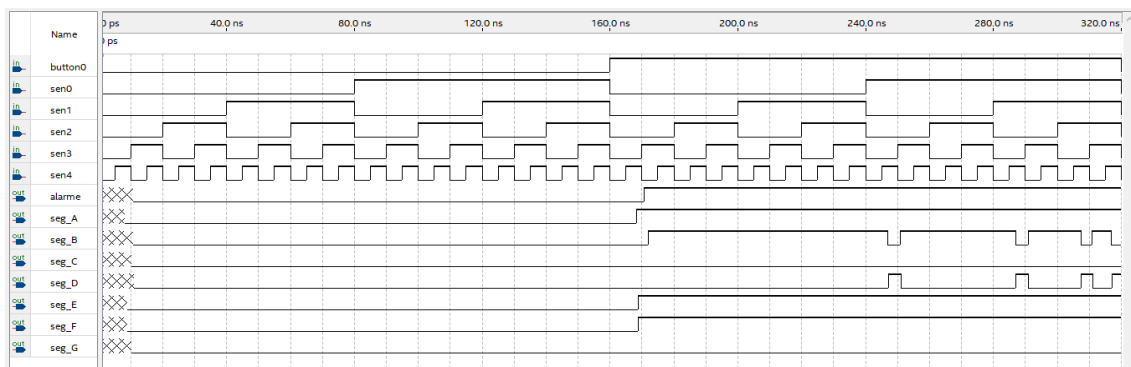


Figura 14: Teste temporal do sistema completo.

De maneira análoga aos testes de cada módulo, o teste do sistema como um todo foi bem sucedido. Pode ser percebido por meio da figura 13, que os segmentos que recebem nível lógico alto como saída estão de acordo com as suas respectivas entradas. Sendo assim, quando 5 sensores emitem nível lógico alto as saídas para a formação da letra correspondente a representação do lote aceito também recebem nível lógico alto. Já quando somente 4 sensores apresentam nível lógico alto, as saídas referentes a ativação dos segmentos do display para a formação da letra correspondente a lote comprometido, ostentam nível lógico alto. Porém, caso somente 3 ou menos sensores disporem de nível lógico alto, as saídas para a ativação dos segmentos que ilustram o símbolo representante de lote rejeitado, ganham o *status* de nível lógico alto. Além disso, pode ser notado que a saída do alarme apresenta nível lógico alto quando o lote é avaliado como comprometido ou rejeitado.

Enquanto que no teste temporal, exposto na figura 14, pode ser verificado atrasos de propagação em todas de saída: *SEG_A* atraso de 8,74 ns, *SEG_B* atraso de 10 ns, *SEG_C* atraso 10 de ns, *SEG_D* atraso de 12,4 ns, *SEG_E* atraso de 8,74 ns, *SEG_F* atraso de 8,74 ns, *SEG_G* atraso de 10 ns e o *alarme* atraso de 10,4 ns.

5. Conclusão

O circuito digital foi elaborado completamente, de forma que atende todos os requisitos solicitados pelo cliente. Entretanto, poderia ser mais simplificado em relação às saídas responsáveis pelas condições do lote. Por exemplo, poderíamos ter utilizado apenas duas saídas e a outra seria representada pela negação das duas. Isso porque, apenas uma saída seria ativa por vez, sendo assim, caso duas saídas estivessem com nível lógico BAIXO, a outra automaticamente teria que estar com nível lógico ALTO. Com isso, o módulo do display conteria apenas duas entradas, e não 3. Além da entrada de habilitação. Em suma, o circuito todo necessita apenas de 2 saídas no que diz respeito a sua elaboração no verilog.

A execução dos testes ratifica o funcionamento dos módulos e do sistema como um todo. De modo geral, seria recomendado a execução dos testes também com os dispositivos físicos referenciados no projeto, a fim de garantir uma melhor confiabilidade dos resultados obtidos. Contudo, com a presente situação pandêmica, não foi possível ter acesso aos equipamentos necessários de forma física para efetuar este tipo de teste.

Dessa maneira, podemos concluir que a elaboração e desenvolvimento do projeto, o qual fez-se levantar questionamentos relevantes quanto a elaboração de um projeto de circuito digital, foi responsável por construir um conhecimento sólido referente às metodologias e os processos de elaboração de circuitos digitais combinacionais.

6. Referências

FLOYD, Thomas L. **Sistemas Digitais: Fundamentos e aplicações**. 9. ed. Porto Alegre: Bookman, 2007.

GOUVEIA, R. **Tabela verdade**. Toda Matéria, Disponível em: <<https://www.todamateria.com.br/tabela-verdade/>>. Acesso em 15 de Mar. de 2020.