



Comportamento Temporal de Circuitos Digitais Utilizando o Logisim

Luiz Carlos Carchedi

JUIZ DE FORA

MARÇO, 2016

Comportamento Temporal de Circuitos Digitais Utilizando o Logisim

LUIZ CARLOS CARCHEDI

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Prof. Eduardo Barrére

JUIZ DE FORA

MARÇO, 2016

COMPORTAMENTO TEMPORAL DE CIRCUITOS DIGITAIS UTILIZANDO O LOGISIM

Luiz Carlos Carchedi

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Prof. Eduardo Barrére
D. Sc. em Engenharia de Sistemas e Computação, COPPE/UFRJ

Profa. Luciana Conceição Dias Campos
D. Sc. em Engenharia Elétrica, PUC - Rio

Prof. Daves Márcio Silva Martins
M. Sc. em Sistemas Computacionais, UFRJ

JUIZ DE FORA
04 DE MARÇO, 2016

*Aos meus avós, por terem construído um mundo
melhor pra mim.*

Aos meus pais, pelo apoio e confiança.

Resumo

A falta da possibilidade de visualização gráfica das formas de onda dos projetos feitos no Logisim dificulta a compreensão do aluno sobre uma parte importante da disciplina e dificulta a explicação da mesma por parte do professor. O objetivo deste trabalho é o desenvolvimento de uma nova versão do Logisim que contenha uma ferramenta que permita a visualização das formas de onda dos projetos desenvolvidos com este software.

Palavras-chave: Circuitos Digitais, Formas de onda, Logisim, Ensino de Informática

Abstract

The lack of possibility of graphical visualization of the waveforms of the projects done with Logisim software makes it difficult task for the teacher to explain the subject and for the for the student to fully understand a important part of such subject. The goal of this work is to develop a new version of the Logisim containing a tool that allows the visualization of the wave forms of the projects done with that software.

Keywords: Digital Circuits, waveforms, Logisim, Teaching Informatics

Agradecimentos

A todos os meus familiares, pelo encorajamento e apoio.

Muito especialmente à minha mãe, que tornou possível a realização da minha graduação, fornecendo sempre tudo o que foi preciso, e muito mais que isso, com carinho e amor.

Aos meus avós, que fizeram do meu mundo um lugar melhor.

Aos meus valiosos amigos, sem os quais, a vida seria triste.

A todos aqueles que, ao meu lado, me fizeram acreditar que era possível realizar esse sonho.

Ao professor Eduardo Barrére pela orientação, amizade e principalmente, pela paciência, sem a qual este trabalho não se realizaria.

“Trabalha, estuda, ganha teu dia com dignidade.”

Autor desconhecido

Sumário

Lista de Figuras	7
Lista de Abreviações	8
1 Introdução	9
2 Revisão Bibliográfica	10
3 O Estudo de Circuitos Digitais	12
3.1 A relação entre a Computação e o estudo da Eletrônica	12
3.2 Os Circuitos Digitais e sua relevância dentro da Computação	12
3.3 A vantagens do uso de ferramentas auxiliares no ensino	13
3.4 O Logisim	14
3.5 Agregar valor ao ensino com o auxílio do Logisim	15
4 A Criação da Ferramenta	17
4.1 Projeto	17
4.2 Desenvolvimento da ferramenta	17
4.2.1 Formas de Onda	18
4.2.2 Organização	19
4.2.3 Funcionamento	20
4.3 Apresentação da ferramenta	25
5 Conclusão	27
Referências Bibliográficas	29

Lista de Figuras

3.1	Janela principal do Logisim	15
4.1	Tabela verdade de um circuito digital	18
4.2	Forma de onda que representa o comportamento temporal de uma entrada de um circuito digital	19
4.3	Janela da Ferramenta	25

Lista de Abreviações

DCC	Departamento de Ciência da Computação
MML	Multimedia Logic
MPDSim	Mips Pipeline Didatic Simulator
UFJF	Universidade Federal de Juiz de Fora

1 Introdução

O principal objetivo de um curso é ensinar ao aluno da forma mais ampla e completa possível, mas ainda hoje existem disciplinas nas quais infelizmente o aluno ainda parece estar distante do conteúdo ministrado, o que trás a quem ministra a disciplina o desafio de conseguir fazer o aluno absorver o conteúdo da forma mais proveitosa possível. Pensando nisso, e levando em conta o uso da informática na educação, desenvolveram-se muitas ferramentas voltadas ao ensino em diversas áreas.

Uma dessas disciplinas que trazem esse desafio ao professor é a disciplina de Circuitos Digitais, na qual, de certa forma, a teoria ainda é distante da prática. Essa distância é devida a uma série de restrições, sendo as principais a dificuldade de disponibilidade de materiais específicos a serem trabalhados por todos os alunos, o risco de queima/estragos nas peças utilizadas, a infraestrutura disponível (laboratórios) e limitações no tempo em que os alunos podem ter acesso a esses laboratórios.

Entre as diversas ferramentas para o ensino de Circuitos Digitais, encontramos o Logisim(1) que é amplamente utilizado por ser uma ferramenta gratuita, de fácil acesso e de fácil aprendizado e será utilizada nesse trabalho por ser uma ferramenta simples e que permite desde o desenvolvimento dos projetos mais básicos até projetos específicos, mais complexos e mais elaborados.

Este trabalho é voltado para o desenvolvimento de uma nova versão do software Logisim que contenha uma ferramenta que possibilite ao aluno ver as formas de onda (comportamento temporal) relacionadas a seus projetos e dessa forma tenha uma compreensão melhor dos mesmos; e que ainda facilite ao professor nas explicações a respeito desses resultados, uma vez que a falta de uma ferramenta gráfica no Logisim dificulta a compreensão do aluno sobre uma parte importante da disciplina e dificulta a explicação da mesma por parte do professor. Para isso, serão utilizados os conceitos teóricos de Circuitos Digitais e o resultado poderá ser aplicado às aulas desta disciplina.

2 Revisão Bibliográfica

Atualmente é comum o uso da informática na educação por meio de ferramentas virtuais para o ensino de uma forma geral, o que nos fornece um grande auxílio e visa proporcionar facilidade no desenvolvimento e no aprendizado por parte dos alunos.

Segundo Damasceno et al (16), existem importâncias e vantagens no uso de ferramentas digitais num ambiente virtual como apoio pedagógico. Outro fato relevante é exposto em Lucena (17) onde o autor comenta que, além dos simuladores trazerem facilidade, permitem que os alunos tenham contato com o ambiente virtual por mais tempo do que apenas em aula e isso trás a possibilidade de explorarem mais e irem além do que é passado pelo professor na sala de aula.

O uso de ambientes virtuais também é amplamente utilizado no ensino da informática. Ao ensinar disciplinas relacionadas à parte de hardware com o auxílio de uma ferramenta virtual, o professor tem muitas vantagens que não seriam possíveis sem as mesmas, como citado ainda em Damasceno et al (16) e em Netto et al (18), o ensino dessa área na prática traria varias conseqüências onerosas.

Outra possibilidade no uso de ambientes virtuais para o ensino de disciplinas relacionadas à área de hardware é abordada em Rotondo et al (20) onde o autor trata do uso das ferramentas *Blogics!* (2) e *Logisim* para o estudo de avaliação e complexidade de circuitos digitais; o uso de ferramentas para simulação destes conteúdos é uma prática pedagógica comum neste tipo de disciplina.

Um estudo apresentado em Rodrigues et al (19), nos mostra como o autor comparou simuladores de memória cache, destacando os pontos positivos e negativos de cada um e suas características desejáveis; como mostrado no texto, esse estudo só é possível graças ao uso de ferramentas virtuais, pois seria inviável sem as mesmas.

Outro estudo apresentado em Avelar et al (14) mostra uma ferramenta desenvolvida para o ensino do *Pipeline*¹ motivada pelo fato de que os alunos apresentam

¹Processo pelo qual uma instrução de processamento é subdividida em etapas, de forma a possibilitar que partes distintas do processador processem etapas distintas, aumentando assim a eficiência do processamento.

difficuldade em compreender o seu funcionamento. Com o propósito de reduzir essa dificuldade e melhorar ainda mais o aprendizado, foi desenvolvida uma nova ferramenta, o MPDSim (MIPS Pipeline Didactic Simulator 32 bits), simulador didático de Pipeline, que possui os adiantamentos padrões e contribui no incentivo ao estudo, a aprendizagem e a pesquisa em Pipeline. Por meio dos resultados obtidos foi possível observar que o MPDSim é uma ferramenta que responde aos objetivos propostos, sendo um simulador de grande ajuda na aprendizagem, o que reforça a idéia fundamental na vantagem do uso de simuladores.

Em Stanley (22) vemos como o Logisim foi adotado nas aulas de arquitetura de computadores que antes usavam o Multimedia Logic(MML). Apesar de ambos os aplicativos serem gratuitos, o numero de atualizações e melhorias no Logisim fez com que optassem pela mudança. Tal artigo discute o valor pedagógico e a facilidade de uso dessa ferramenta.

Em Stanley (22) e Carl Burch (15) temos ainda que uma das vantagens do Logisim, quando utilizado para a construção de projetos por alunos, é os possibilitar uma interação com o projeto que faz com que desenvolvam e simulem circuitos lógicos vistos em sala de aula. O Carl Burch (15) ainda compara a variedade de sistemas similares ao Logisim e explora detalhadamente as ferramentas do Logisim e o efeito de usá-lo no ensino de Circuitos Digitais.

O Logisim é uma ferramenta gratuita que pode ser adquirida na internet por meio do site da ferramenta (1).

Finalmente em Shuurman (21) vemos o passo a passo da construção de uma CPU de 8 bits. Começando pela ALU com 8 operações básicas, depois um caminho de dados de 8 bits. O terceiro passo é o desenvolvimento de uma unidade de controle usando um microcontrolador e o ultimo passo e a adição de memória ao projeto e um meio de entrada e saída. Para tudo isso é utilizado o Logisim como ferramenta educacional, construindo circuitos complexos a partir de sub-circuitos mais simples. Construindo isso dessa maneira, os alunos tem como alterar as configurações dos projetos e ver como é o comportamento do projeto como um todo a partir de cada alteração e isso os permite visualizar na prática como funciona uma CPU.

3 O Estudo de Circuitos Digitais

3.1 A relação entre a Computação e o estudo da Eletrônica

Desde a primeira metade de século XX, os computadores, que até então eram principalmente máquinas mecânicas para a realização de cálculos, passaram a utilizar a eletricidade em seu funcionamento (3). A partir de então percebeu-se que a eletricidade poderia ser usada para representar informações dentro de uma máquina, tais informações poderiam ser tanto os dados a serem utilizados pela máquina, quanto as instruções sobre o que a máquina deveria fazer com esses dados para a realização dos cálculos pretendidos. Desse modo nasce a eletrônica, que nos trouxe a idéia de usar a eletricidade pra representar informações e poder manipulá-las através de circuitos projetados com esse fim.

3.2 Os Circuitos Digitais e sua relevância dentro da Computação

Com o objetivo de representar uma informação por meio da energia elétrica, surgiu a necessidade de criar um padrão para a representar a informação. Desde o início do século XIX já havia a idéia do uso de cartões perfurados, que surgiu com Joseph Marie Jacquard e a construção de um tear inteiramente automatizado (4). O uso desses cartões permitiam representar informações a partir de áreas perfuradas nos mesmos.

Essa forma de representação foi a base para a representação adotada na eletrônica. A representação em cartões perfurados nada mais é que um modo de representar a informação utilizando-se de um padrão pré-estabelecido (por exemplo, o local no cartão onde haveria furos e o que cada furo significava), e de dois estados possíveis pra cada local (furado ou não).

Essa representação foi facilmente passada para o contexto eletrônico, uma vez

que os estados possíveis de cada área do cartão (furado ou não) simplesmente foram representados por estados elétricos diferentes. Pouco depois, percebeu-se que, esse sistema poderia ser representado por dígitos, onde cada dígito representaria um dos estados possíveis passando a ser chamado “Sistema Digital”.

Outra característica do sistema digital é utilizar os dígitos “0” e “1”, para representar os estados possíveis e por isso, o sistema também é chamado “sistema binário”. A partir daí, utilizando-se a algebra de boole (13) e o sistema binário, por meio da eletrônica foi possível manipular informações e submetê-las a um processo pré-estabelecido que gera resultados, a isso é dado o nome de processamento.

Utilizando essas idéias, primeiro houveram computadores com o objetivo de realizar processamentos específicos e mais tarde computadores programáveis, ou seja, computadores que mesmo depois de prontos, podiam ser configurados (programados) pra realizar diferentes cálculos.

O ensino da eletrônica mais básica, que é a origem dos sistemas computais atuais começa com o ensino de circuitos digitais, que é fundamental para o entedimento de todo o restante da eletrônica. No enisno de circuitos digitais, são apresentadas as formas mais simples de processamento e manipulação de entradas através de ciruitos montados com os componentes básicos.

3.3 A vantagens do uso de ferramentas auxiliares no ensino

Um das formas mais eficazes no ensino de uma disciplina é conseguir envolver o aluno com o conteúdo da mesma, e quanto mais teórico o conteúdo, maior a necessidade de um meio de envolvê-lo. Por isso, as disciplinas que se utilizam de ferramentas auxiliares, tende a ter um melhor aproveitamento porque o aluno participa de forma ativa do conteúdo ministrado (5). Levando em conta tais informações, foram construídas ferramentas nas mais diferentes áreas com o objetivo de aproximar o aluno do conteúdo.

Com a popularização do computador no final do século XX (3), tornou-se muito mais fácil a criação dessas ferramentas de modo digital e também a difusão destas.

Uma das disciplinas que trazem vantagens ao aluno e ao professor ao se utilizar de ferramentas auxiliares é a disciplina de Circuitos Digitais, que, de certa forma, a teoria ainda é distante da prática. Essa distância é devida a uma série de restrições, sendo as principais a dificuldade de disponibilidade de materiais específicos a serem trabalhados por todos os alunos, o risco de queima ou estragos nas peças utilizadas, a infraestrutura disponível e limitações no tempo em que os alunos podem ter acesso a esses laboratórios.

Entre as ferramentas para o ensino de Circuitos Digitais, encontramos ferramentas como o Multisim, DigitalWorks, Quartus II e o Logisim. Nesse trabalho, no entanto, foi utilizada a ferramenta Logisim por se tratar de uma ferramenta voltada ao ensino, além de ser uma ferramenta simples, gratuita e de fácil acesso.

Ao utilizar ferramentas de apoio tal qual o Logisim, os alunos conseguem suprir essa necessidade de uma parte mais prática que auxilie o aprendizado do conteúdo ministrado em Circuitos Digitais.

3.4 O Logisim

O Logisim é uma ferramenta gratuita disponibilizada na internet, que permite o desenho, o projeto e a simulação de circuitos digitais através de uma interface gráfica. A ideia inicial do projeto foi do professor Carl C Burch, e posteriormente foi continuada por grupos de estudos.

O programa é escrito na linguagem *Java* (6) e tem o código aberto, o que permite que sejam feitas atualizações e até mesmo novas versões. Por conta disso, surgiram algumas versões do Logisim voltadas a objetivos específicos, como é o caso do “*Logisim Evolution*” (7), “*Logisim Lawrance version*” (8), “*Logisim IITD*” (9) e do “*Logisim CornellCS Version*” (10).

Embora seja possível aos usuários realizarem implementações bastante complexas e com qualidade, o programa se destina em princípio ao uso educacional, com o objetivo que os alunos consigam familiaridade na utilização de circuitos sem a necessidade específica de contato com circuitos reais.

O Logisim possui ferramentas que possibilitam a mudança de estados nas entradas e propaga automaticamente os sinais pelos circuitos, o que permite ao aluno ir visualizando

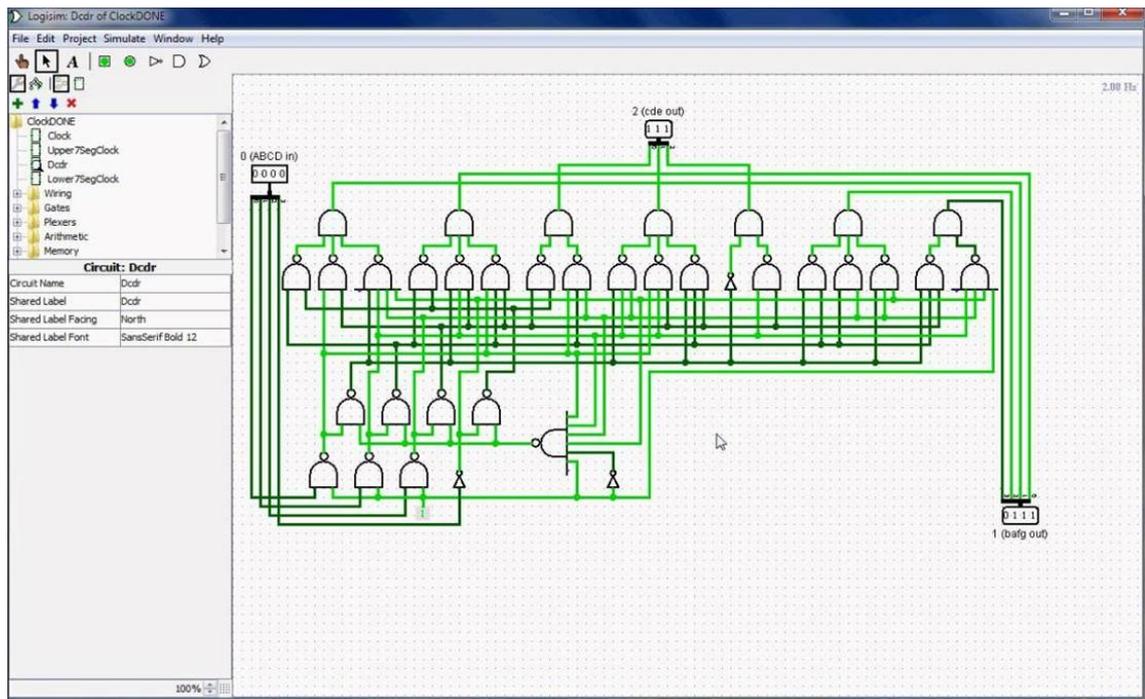


Figura 3.1: Janela principal do Logisim

as respostas do circuito em tempo real; isso dá ao Logisim uma vantagem muito poderosa em relação aos outros meios de estudos de circuitos, como por exemplo, impresso em papel, que não mostra sua evolução. Há ainda a possibilidade de trabalho com hierarquias de circuitos, permitindo trabalhar um projeto por áreas e dessa forma conseguir com muita facilidade desenvolver projetos complexos que, impressos, se tornam inviáveis na realização de testes.

Há ainda o uso profissional, que envolve áreas mais específicas e complexas na utilização do software.

Devido à própria idéia do projeto, o Logisim não apresenta a possibilidade de simulação de circuitos analógicos.

3.5 Agregar valor ao ensino com o auxílio do Logisim

A idéia do projeto é agregar valor ao ensino com o auxílio do Logisim e incrementá-lo com uma ferramenta capaz de gerar os gráficos dos circuitos construídos com ele.

Embora o Logisim seja uma ferramenta amplamente utilizada, na própria ideia de sua concepção já há a previsão de melhorias a serem implementadas ao longo do tempo, o que de fato veio acontecendo com o surgimento de muitas outras versões do Logisim,

que hoje distinguem-se por serem versões voltadas a objetivos específicos.

A idéia fundamental desse trabalho é desenvolver uma nova ferramenta para o Logisim que possibilite ao aluno ver as formas de onda ² (comportamento temporal) de seus projetos e dessa forma tenha uma compreensão melhor dos mesmos; e que ainda facilite ao professor quanto às explicações a respeito de tais resultados, uma vez que a falta de uma ferramenta para a visualização das formas de onda no Logisim dificulta a compreensão do aluno sobre uma parte importante da disciplina e dificulta a explicação da mesma por parte do professor.

A princípio, esse projeto idealizava a construção de um *Plugin* que pudesse ser inserido em qualquer versão já existente do Logisim e trouxesse essa ferramenta, mas, depois de pesquisar e até mesmo entrar em contato com uma equipe que atualmente leva a diante o projeto do Logisim, descobriu-se que o Logisim enquanto ferramenta, não é construído com uma estrutura que permita uma comunicação com ferramentas externas, os *plugins*. A partir de então, o projeto passou a ser a criação de uma nova versão do Logisim. O fato de o Logisim ser uma ferramenta de código aberto faz com que existam algumas versões específicas do software voltadas a objetivos próprios, este trabalho baseou-se na última versão do projeto original do Logisim para criar a nova versão que contenha a ferramenta que possibilite a visualização das formas de onda.

Para isso, foram utilizados os conceitos teóricos de circuitos digitais e os resultados podem ser aplicados nas aulas da disciplina Circuitos Digitais.

²Forma de onda é a representação gráfica da forma com que uma onda evolui ao longo do tempo. Normalmente os fenômenos ondulatórios, tais como o som ou ondas eletromagnéticas obedecem a funções matemáticas periódicas. Para cada função, a evolução da amplitude da onda ao longo do tempo é diferente e define uma forma de onda diferente.

4 A Criação da Ferramenta

4.1 Projeto

Como o Logisim é um software construído com a linguagem *Java*, primeiro foi desenvolvida apenas uma aplicação em *Java* que gerasse gráficos de circuitos digitais de acordo com dados fornecidos pelo próprio usuário.

Dando continuidade ao trabalho, obteve-se o projeto do Logisim como ponto de partida para o desenvolvimento da nova versão.

De posse deste projeto, o próximo passo, foi alterá-lo de modo que, dentro dele, houvesse uma opção no *menu* principal que permitisse a visualização de uma nova janela que apresentasse a aplicação feita anteriormente.

A partir daí, a mudança a ser feita foi fazer com que a ferramenta deixasse de trabalhar apenas com os dados fornecidos pelo usuário e passasse a trabalhar recebendo dados do usuário e do circuito já construído na parte principal do Logisim quando a ferramenta fosse acionada.

4.2 Desenvolvimento da ferramenta

Um circuito digital é uma representação da relação entre entradas e saídas de um circuito onde, o valores obtidos nas saídas, depende dos valores fornecidos nas entradas.

Uma situação onde o valor de cada entrada está definido (como 0 ou 1) será chamada doravante de *estado* do circuito.

A ferramenta desenvolvida, tem por objetivo gerar gráficos onde sejam desenhadas as relações das entradas e saídas do circuito. A área de plotagem do gráfico dentro da janela da ferramenta, é dividida horizontalmente em faixas destinadas a cada entrada/saída do circuito e verticalmente dividida em faixas que representam o estado do circuito. Logo, uma faixa vertical, cobrindo toda a área plotável representa um único estado do circuito e mostra os valores de todas as entradas e saídas durante aquele estado.

a	b	c	d	x	y	z
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	1	0	1
0	0	1	1	1	0	1
0	1	0	0	0	1	0
0	1	0	1	0	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	1
1	0	0	0	0	1	0
1	0	0	1	0	0	1
1	0	1	0	1	0	1
1	0	1	1	1	0	1
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	0	1

Figura 4.1: Tabela verdade de um circuito digital

O Logisim já possui uma ferramenta que gera o que é conhecido como “*Tabela Verdade*”. A *Tabela Verdade* é uma tabela que relaciona cada estado possível do circuito com as saídas obtidas a partir dele.

Uma vez recebidos os valores a serem atribuídos às entradas para gerar um estado possível do circuito, é possível obter via *Tabela Verdade* os valores relativos às saídas e assim torna-se possível desenhar cada um dos gráficos.

4.2.1 Formas de Onda

As formas de onda que representam o comportamento temporal de uma entrada ou saída de um circuito digital tem um formato muito bem definido. São desenhadas a partir de dois eixos ortogonais e ocupam apenas o primeiro quadrante. Uma vez que verticalmente representa-se os valores possíveis da entrada ou saída (0 ou 1), e horizontalmente representa-se o comportamento temporal, não faz sentido representações com valores negativos em nenhuma das suas duas dimensões. Isso significa que o gráfico é di-

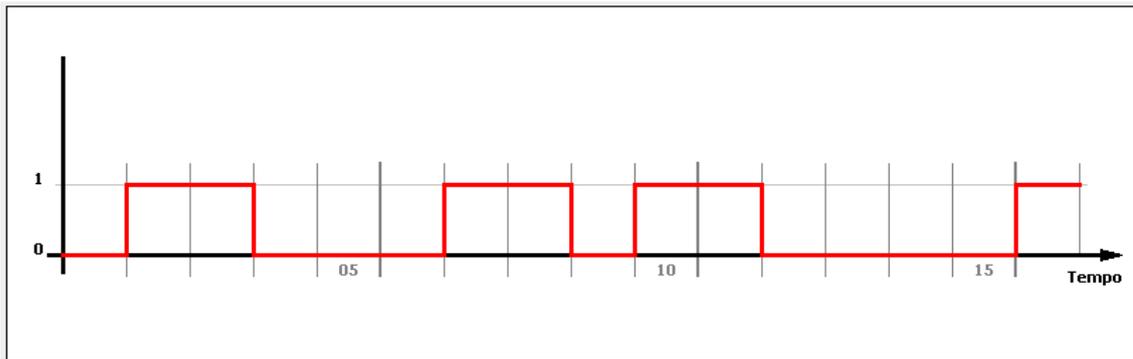


Figura 4.2: Forma de onda que representa o comportamento temporal de uma entrada de um circuito digital

vidido em faixas verticais que representam os valores daquela entrada ou saída de acordo com o estado do circuito, e ainda, da esquerda para a direita, tais faixas representam a ordem que foram inseridas. Visto que cada faixa vertical representa apenas um estado do circuito, e que a entrada ou saída não é alterada durante o tempo em que o circuito permanece nesse estado, é uma constante. O gráfico representa essa entrada ou saída durante esse estado por um segmento de reta horizontal, na altura do valor que a entrada ou saída apresenta. Para dar a ideia de continuidade temporal ao gráfico gerado pela ferramenta, entre os segmentos de reta horizontais consecutivos e em alturas distintas, desenha-se um segmento de reta vertical unindo-os.

4.2.2 Organização

No projeto original do Logisim, utilizado como base para o desenvolvimento desta nova versão, existe a classe *LogisimMenuBar* dentro do pacote principal, que é a classe responsável pela organização do *menu* principal do Logisim. Dentro dessa classe estão definidos todos os *menus* da janela principal do Logisim; a partir do padrão encontrado nessa classe, foi criado mais um objeto do tipo *LogisimMenuItem*, dentro do *menu* “Projeto”, nomeado “GetCircuitGraphs”, que é o modo de acesso à nova funcionalidade inserida no Logisim. Ainda no mesmo pacote, há a classe, *MenuProject* que trata mais especificamente de como será exibido o *menu* “Projeto” na janela principal do Logisim e ainda de qual diretório e arquivo o software deve obter o texto a ser apresentado no *menu*; isso porque o Logisim oferece a opção de ser utilizado em diferentes idiomas onde o texto de cada idioma é escrito em um documento de texto externo que deve ser colocado no diretório

correto durante a instalação do Logisim. Feitas as devidas alterações nessas classes, o projeto passou a apresentar a opção “Obter gráficos do circuito”.

Utilizando-se da estrutura já presente no projeto para a criação de janelas, foi possível adicionar ao Logisim uma nova janela nomeada “Obter gráficos do circuito” e configurá-la de acordo com os parâmetros necessários. Essa nova janela passou a ser exibida no Logisim quando selecionado o *menu* de mesmo nome. As janelas no Logisim são também uma classe presente no mesmo pacote principal do projeto.

A partir de então, por meio da ferramenta “*WindowBuilder*” (11) disponível como um *plugin* para o *Eclipse IDE for Java Developers* (12) foi possível construir a interface desejada contendo todas as funcionalidades necessárias para o funcionamento da ferramenta no Logisim, essa interface foi a mesma utilizada no primeiro aplicativo que já desenhava os gráficos antes de estar ligado ao Logisim.

Quando a janela “*Obter gráficos do circuito*” já estava pronta e com a interface desejada faltava ainda o último passo que era fazer com que os gráficos gerados nessa janela, fossem os gráficos que representassem o circuito desenhado na janela principal do Logisim. Para isso, foi preciso obter os dados do circuito pra que houvesse como gerar gráficos coerentes com o mesmo. Esse dados são obtidos da Tabela Verdade do circuito, já construída pelo Logisim.

No projeto do Logisim, a Tabela Verdade pode ser obtida através do método “*GetTruthTable*” do objeto da classe “*AnalyzerModel*” presente no pacote ***logisim.analyse.model***, a grande vantagem encontrada aqui é que uma vez que há um circuito no Logisim, com este método, já é possível criar uma matriz e preenchê-la tal qual a Tabela Verdade do circuito, relacionando os estados possíveis do circuito com as respectivas saídas que eles geram. De posse dessa relação, a cada vez que o usuário insere um novo estado na ferramenta, é possível descobrir, via Tabela Verdade, o valor de todas as saídas do circuito e então desenhar os gráficos relacionados a cada entrada e cada saída do circuito.

4.2.3 Funcionamento

Para o funcionamento da ferramenta proposta, o usuário deve informar um estado possível do circuito, e a ferramenta se encarrega de calcular quais serão os valores das

saídas do circuito e em seguida desenhar gráficos que representam, de forma individual, a cada entrada e cada saída do circuito.

Cálculo das Saídas a partir das entradas

A tabela verdade de um circuito possui um método correto de ser construída; ela precisa ter uma coluna para cada entrada e uma para cada saída do circuito. Além disso, a tabela verdade precisa ter um número de linhas suficiente para apresentar *todas* as possibilidades de estados do circuito. Cada linha da tabela verdade, quanto às entradas, representa um estado possível do circuito, como cada entrada pode assumir dois valores distintos, cada entrada adicionada ao circuito, dobra o número de estados possíveis do circuito, uma vez que agora, com uma nova entrada, o circuito pode assumir todos os estados que já podia setando a nova entrada com o valor 0, e mais todos os estados que já podia setando a nova entrada em 1. Dessa maneira a tabela verdade possui:

- Colunas: $in + out$
- Linhas: 2^{in}

Onde *in* representa a quantidade de entradas do circuito e *out* representa a quantidade de saídas do circuito.

As linhas da tabela verdade, levando em consideração suas entradas, representam valores, em base 2, que vão do 0 até $(2^{in} - 1)$.

Desse modo, ao receber um estado possível do circuito, a ferramenta constrói uma *string* com os valores das entradas ordenados segundo um padrão previamente estabelecido. A seguir utiliza a função “*GetLine*” abaixo, que recebe essa string formando um valor numérico na base 2, indicando qual linha da tabela verdade apresenta aquele estado, e retorna esse mesmo valor na base 10, desse modo, consegue-se a linha que indica os valores das saídas geradas por aquele estado.

Gerenciando as entradas e saídas

A área de plotagem do gráfico tem um tamanho fixo na janela da ferramenta, tendo em vista uma maior coerência com a utilização, os gráficos desenhados sempre

Algoritmo 1: Funcao GetLine

```

Input: value
Output: A linha da Tabela Verdade referente às saídas desejadas.
 $x \leftarrow 0$ 
for  $i \leftarrow 0$  to value.length do
  | if value.at(value.length - i) = 1 then
  | |  $x \leftarrow x + 2^i$ 
  | end if
end for
return  $x$ 

```

buscam ocupar toda a extensão desta área. Entretanto é fundamental para o correto funcionamento da ferramenta que os gráficos sejam desenhados a medida que o usuário for inserindo os estados desejados do circuito, desse modo, a cada nova inserção, a ferramenta precisa redesenhar todos os gráficos.

Para que seja possível todo esse processo, foi criada na ferramenta uma matriz nomeada como “*MatrizHistorica*” para que quando for inserido um novo estado do circuito, ele se torne uma nova linha no final dessa matriz, e dessa forma, sempre que necessário redesenhar os gráficos, haja um local de onde se possa obter cada um dos estados do circuito inseridos anteriormente, organizados de forma temporal.

Essa *MatrizHistorica* possui duas dimensões, onde cada coluna representa uma entrada/saída do circuito e cada linha um estado do circuito inserido pelo usuário. A diferença fundamental entre a *MatrizHistorica* e a *tabela verdade* é que o número de linhas da tabela verdade se limita à quantidade de estados possíveis do circuito, enquanto que na *MatrizHistorica*, a quantidade de linhas depende da quantidade de estados inseridos pelo usuário, podendo ainda conter estados inseridos mais de uma vez, o que não ocorre na tabela verdade. Trabalhando desse modo, cada coluna da *MatrizHistorica* estará sempre preenchida por 0’s e 1’s modo que, varrida de cima pra baixo, a sequência obtida gera exatamente os valores fornecidos pelo usuário àquela entrada em ordem cronológica.

Desenhando os Gráficos

Uma vez que a *MatrizHistorica* contenha dados, isso indica que o usuário já inseriu um ou mais estados do circuito, e portanto, é possível desenhar os gráficos; o que é gerenciado através da função “*GraphManager*”.

A função “*GraphManager*” foi desenvolvida para organizar a forma que os diversos gráficos (um pra cada entrada e um pra cada saída) dividem o espaço total para plotagem de gráficos na janela.

Seu funcionamento é acionado a cada inserção de um novo estado do circuito. A cada vez ela redesenha todos os gráficos levando em conta todos os estados inseridos pelo usuário. Em seu funcionamento, ela divide a área de plotagem dos gráficos em faixas horizontais de dimensões iguais, uma para cada entrada/saída do circuito. Esse cálculo é feito de forma simples, dividindo a altura da área plotável pela quantidade (entradas+saídas) do circuito. Feito isso, pra cada entrada/saída do circuito é chamada a função “*DrawChart*”.

A função “*DrawChart*” recebe da função “*GraphManager*” a posição e a altura da faixa horizontal na qual ela deve desenhar um gráfico; recebe ainda um vetor que é montado a partir de uma coluna da “*MatrizHistorica*”, o tipo de elemento do circuito (entrada ou saída) e o nome atribuído àquela entrada/saída.

A partir destes dados, a função pode calcular as margens internas à faixa destinada ao gráfico em questão, que precisam ser respeitadas para que o desenho do gráfico fique bem posicionado, centralizado na faixa a ele destinada e dessa forma não sobreponha e nem mesmo fique muito próximo das áreas destinadas aos demais gráficos. Calculadas essas margens, toma-se a parte mais à esquerda e desenha-se na cor preta um eixo vertical com os valores 0 e 1, e um eixo horizontal que se estende por toda a base da área destinada a esse gráfico. Na sequência, a função escreve na parte superior esquerda o nome da entrada/saída e depois calcula a extensão horizontal que cada segmento de reta representando um estado deverá ter para que, tal qual verticalmente, horizontalmente o gráfico possa ocupar toda a área a ele destinada, com segmentos de reta de mesmo comprimento. Feitos todos esses cálculos, a função desenha na cor cinza a marcação da altura vertical do valor 1 da entrada, e faz marcações dividindo o eixo horizontal em segmentos com a largura calculada para cada estado, dando destaque às marcações múltiplas de 5 unidades que também recebem um valor indicando a quantidade de marcações até ali.

A última coisa feita pela função “*DrawChart*” é desenhar a curva do gráfico. Caso seja o gráfico de uma entrada, é desenhado na cor vermelha, caso seja de uma saída,

desenhado na cor azul. Essa distinção visa somente facilitar a identificação por parte do aluno e a explicação por parte do professor.

A medida que vão sendo inseridos estados pelo usuário, se a função mantiver a altura do valor 1 no eixo vertical, o gráfico tende a mudar de forma, visto que o comprimento horizontal de cada segmento diminui a cada vez que novos estados do circuito são inseridos. Para evitar que o gráfico pareça destorcido (embora não esteja errado), a cada inserção de um novo estado, a altura do valor 1 no eixo vertical também é recalculada de modo a manter (dentro do possível) um padrão no formato das formas de onda. Por padrão, a função procura estabelecer que o comprimento no eixo vertical do valor 1 seja o mesmo comprimento calculado para o segmento de reta de cada estado, entretanto, visando uma melhor disposição do gráfico gerado e uma melhor visualização do mesmo, a função também estabelece um valor mínimo e um valor máximo para essa altura, de modo que ela pode variar, mas sempre dentro dos limites estabelecidos.

Caso haja no circuito uma saída que não esteja conectada a qualquer entrada, ou ainda que por qualquer outro motivo, o Logisim ao construir a tabela verdade do circuito, não tenha como atribuir um valor a essa saída para um dado estado, ao ser desenhado o gráfico desse estado específico, a ferramenta desenhará um “X” na faixa vertical destinada a esse estado.

Destaque na Área do Gráfico

A última funcionalidade da ferramenta, permite que, depois que houverem valores inseridos e um gráfico já desenhado, ao selecionar um dos estados inseridos na lista de estados, a faixa vertical do gráfico referente àquele estado fique destacada. Para que isso aconteça, existe a função “*ItemMark*” que é chamada a cada vez que um item na lista de estados é selecionado.

A função “*ItemMark*” funciona desenhando sobre a área do gráfico referente ao estado selecionado um retângulo verde, com certo grau de transparência. O objetivo dessa funcionalidade da ferramenta é destacar no gráfico a área referente ao estado selecionado, para ajudar ao aluno na hora de relacionar um item da lista de estados inseridos e o gráfico apresentado. Por isso, só é permitido ao usuário utilizá-la em apenas um estado

atribuído quando foi inserida no circuito (caso o usuário não atribua um nome, o Logisim se encarrega de fazer isso automaticamente).

Nessa lista, o usuário deve selecionar, para cada entrada do circuito, um dos valores possíveis (0 ou 1), e desse modo, gerar um estado para o circuito. Feito isso, basta inseri-lo na lista de estados presente logo abaixo. Na direita da janela se encontra a área de plotagem dos gráficos, que é atualizada a cada inserção de um novo estado.

Na parte inferior da janela, há a opção de remover um estado específico selecionado na lista de estados e a opção de remover todos os estados da lista. Há também uma linha textual, abaixo da área de plotagem do gráfico, sob o rótulo de “saída”, que mostra quais são as saídas do gráfico, para cada estado do circuito inserido na lista de acordo com a ordem de inserção.

5 Conclusão

O software Logisim, amplamente utilizado no ensino de Circuitos Digitais, apresentava uma característica consideravelmente desfavorável ao aprendizado de parte importante da disciplina. Isto ocorria devido a ausência de uma ferramenta que possibilitasse a visualização das formas de onda (comportamento temporal) dos projetos construídos no Logisim, dificultando não somente a compreensão do aluno sobre o assunto, mas também comprometia a abordagem pedagógica utilizada pelo professor na disciplina de Circuitos Digitais.

Este trabalho direcionou-se para o desenvolvimento de uma nova versão do Logisim que contenha uma ferramenta que possibilite ao usuário ver as formas de onda (comportamento temporal) relacionadas aos circuitos construídos nessa versão, e dessa forma, tenha uma compreensão melhor dos mesmos. Para isso, utilizou-se os conceitos teóricos de Circuitos Digitais.

Além de promover um melhor aproveitamento do software, essa nova versão visa estabelecer ampla conexão entre o ensino e os usuários, assegurando uma melhor performance.

É importante ressaltar que, num circuito digital, há a possibilidade de se colocar temporizadores (*“clock”*) como parte do circuito, entretanto, na ferramenta desenvolvida nesse projeto, a noção de tempo é dada pelo usuário, de forma que não há como usar esta ferramenta em projetos que tenham temporizadores. Um dos motivos que levam a esse resultado é o fato de a ferramenta que desenha os gráficos ser completamente dependente da *“Tabela Verdade”* gerada pelo Logisim, e o Logisim não gera Tabelas Verdade que levam em conta temporizadores.

A criação de uma nova versão do Logisim contendo uma ferramenta voltada especificamente para o ensino, como é o caso deste trabalho, mostra a possibilidade de surgimento de novas versões do Logisim com mais melhorias voltadas a objetivos específicos.

Como sugestão para trabalhos futuros, é proposto para a ferramenta desenvolvida neste trabalho, a adição de uma funcionalidade que permita ao usuário salvar os estados

inseridos para que os gráficos gerados pela ferramenta possam ser recuperados.

Referências Bibliográficas

- [1] Burch, C. C. **Logisim website**. <http://www.cburch.com/logisim>. Accessed May 22, 2015.
- [2] **Blogics website**. <https://code.google.com/archive/p/blogics/>. Accessed March 05, 2016.
- [3] **A evolução dos computadores**. <http://www2.ic.uff.br/~aconci/evolucao.html>. Accessed March 05, 2016.
- [4] **Joseph-marie jacquard**. https://pt.wikipedia.org/wiki/Joseph-Marie_Jacquard. Accessed March 05, 2016.
- [5] **Diferentes usos do computador na educação**. <http://ffalm.br/gied/site/artigos/diferentesusoscomputador.pdf>. Accessed March 05, 2016.
- [6] **Java website**. https://www.java.com/pt_BR/about/whatis_java. Accessed Feb 15, 2016.
- [7] **Logisim evolution website**. <https://github.com/reds-heig/logisim-evolution>. Accessed May 22, 2015.
- [8] Lawrance, J. **Logisim lawrance version**. <https://github.com/lawrancej/logisim>. Accessed May 22, 2015.
- [9] **Logisim iitd**. <https://code.google.com/archive/p/logisim-iitd/>. Accessed May 22, 2015.
- [10] **Logisim cornellcs version**. <http://www.cs.cornell.edu/courses/cs3410/2015sp/>. Accessed May 22, 2015.
- [11] **Window builder website**. <https://eclipse.org/windowbuilder/>. Accessed February 16, 2016.
- [12] **Eclipse foundation website**. <https://eclipse.org/org/>. Accessed February 16, 2016.
- [13] Szejnberg, M. **Eletrônica digital - teoria, componentes e aplicações**. 2013.
- [14] Avelar, C. P.; Filho, J. J. C.; Carvalho, V. ; Martins, C. A. P. S. **Mpdsim: Simulador didático do pipeline do mips de 32 bits**. Campo Grande, Brasil, 2008.
- [15] Burch, C. **Logisim: a graphical system for logic circuit design and simulation**. In: Journal on Education Resources in Computing (JERIC) Volume 2 Issue 1, March 2002 Pages 5–16, 2002.
- [16] Damasceno, E. F.; Oliveira, D. C. **Um ambiente virtual para ensino de instalação e manutenção de microcomputadores**. p. 171–183, Rio Verde, Brasil, 2013.

-
- [17] **Desenvolvimento de uma Ferramenta Computacional para Modelagem e Simulação de Circuitos Eletrônicos Digitais**, 2013.
- [18] **Ensino de Arquitetura de Computadores para Cursos de Tecnologia com Ênfase em Software**, 2007.
- [19] das Chagas Rodrigues, J.; Xavier, M. A. S. ; de Lima Júnior, O. A. **Simuladores de memória cache: Um estudo comparativo e sua aplicabilidade na educação**. In: International Journal of Computer Architecture Education (IJCAE), p. 30–38, 2012.
- [20] Rotondo, G.; Ferreira, L. S.; Machado, D.; Piovesan, S. D.; Camargo, S. S. ; Amaral, E. M. H. **Modelo para mensuração de projetos de circuitos lógicos em atividades acadêmicas**.
- [21] Schuurman, D. **Step-by-step design and simulation of a simple cpu architecture**. In: Proceeding of the 44th ACM technical symposium on Computer science education. Pages 335–340, 2012.
- [22] Stanley, T. **Teaching computer architecture through simulation: (a brief evaluation of cpu simulators)**. In: Journal of Computing Sciences in Colleges, Volume 27 Issue 4, April 2012 Pages 37–44, 2012.