



DESENVOLVIMENTO DE FERRAMENTA DE AUXÍLIO DIDÁTICO: IMPLEMENTAÇÃO DE INTERFACES COM O USUÁRIO VIA INTERNET USANDO CGILUA

Bruno de S. Santos – brunodess@telecom.uff.br
Alexandre S. de la Vega – delavega@telecom.uff.br
Grupo PET-Tele – <http://www.telecom.uff.br/pet>
Universidade Federal Fluminense – UFF
Escola de Engenharia – TCE
Departamento de Engenharia de Telecomunicações – TET
Rua Passo da Pátria, 156 / Bloco D / Sala 504
24.210-240 – Niterói – RJ

Resumo: O objetivo principal do presente trabalho foi implementar interfaces com o usuário via Internet para os aplicativos de um conjunto de ferramentas de auxílio didático que o grupo PET-Tele está desenvolvendo, para o ensino de Circuitos Digitais: conversão de expressões lógicas, minimização de expressões lógicas e minimização de máquinas de estados. Uma vez que os aplicativos foram codificados com a linguagem de programação Lua, as interfaces foram naturalmente codificadas através da linguagem de ligação CGILua.

Palavras-chave: Ferramenta didática, Ensino à distância, Circuitos Digitais, Interface com o Usuário, Linguagem de Programação CGILua.

1. INTRODUÇÃO

O Programa de Educação Tutorial (PET) exige que os bolsistas dos grupos PET, ao serem submetidos a uma formação complementar, desenvolvam atividades que possuam, cada uma delas, itens relativos às áreas de Pesquisa, Ensino e Extensão, bem como consigam algum tipo de penetração no curso ao qual o seu grupo pertence.

Trabalhando na linha de ferramentas de auxílio didático, o grupo PET do Curso de Engenharia de Telecomunicações da Universidade Federal Fluminense (PET-Tele) aproveitou o surgimento da tecnologia de Televisão Digital (TV Digital) e desenvolveu um aplicativo interativo multimídia (MATTOS et al., 2009), codificado na linguagem NCL (SOARES & BARBOSA, 2009) (COMUNIDADE GINGA, 2012).

O aplicativo é um manual que, utilizando recursos de áudio, texto e imagem, descreve o algoritmo de minimização de funções booleanas conhecido por Algoritmo de Quine-McCluskey (HILL & PETERSON, 1981) (RHYNE, 1973) (McCLUSKEY, 1956).

A partir do manual desenvolvido, o grupo visualizou a possibilidade de implementar um pacote de aplicativos didáticos para o ensino de Circuitos Digitais, com as opções de interação com o usuário nas formas local, via TV Digital e via rede de computadores.



Esse trabalho descreve o desenvolvimento das interfaces com o usuário, via rede de computadores, para os aplicativos implementados.

Objetivando-se uma ampla e gratuita divulgação, todo o material produzido pelo grupo é colocado à disposição para *download* no *website* do grupo, após devida publicação.

A implementação das interfaces é apresentada a seguir. A Seção 2 comenta o pacote de aplicativos didáticos que o grupo está desenvolvendo. A escolha das linguagens de programação utilizadas é discutida na Seção 3. A Seção 4 apresenta as interfaces desenvolvidas. Exemplos de operação das interfaces são ilustrados na Seção 5. Finalmente, a Seção 6 apresenta as conclusões e os trabalhos futuros.

2. PACOTE DE APLICATIVOS DIDÁTICOS

Um pacote de aplicativos didáticos dedicado ao ensino de Circuitos Digitais, incluindo as interfaces com o usuário para tais aplicativos, encontra-se em desenvolvimento pelo grupo PET-Tele. Em relação aos integrantes do grupo, o objetivo é fazer com que eles colaborem com a melhoria das práticas de ensino. Para os alunos do curso, o objetivo é fornecer a eles ferramentas auxiliares para estudo e para a verificação de resultados.

O passo inicial (MATTOS et al., 2009) foi um manual interativo multimídia que, utilizando recursos de áudio, texto e imagem, descreve o algoritmo de minimização de funções booleanas conhecido por Algoritmo de Quine-McCluskey (HILL & PETERSON, 1981) (RHYNE, 1973) (McCLUSKEY, 1956). O manual foi implementado para TV Digital, usando a linguagem NCL (SOARES & BARBOSA, 2009) (COMUNIDADE GINGA, 2012).

Em seguida, a fim de possibilitar uma maior interação com o usuário, o algoritmo de minimização foi implementado (BEPPU et al., 2010). Dado que a linguagem NCL não possui mecanismos procedurais de processamento, foi utilizada a linguagem de programação Lua (IERUSALIMSKY, 2006) (IERUSALIMSKY, 2012).

A linguagem Lua foi escolhida devido a sua característica de fácil interação com a linguagem NCL e pela decisão, tomada na época, em adotá-la para os dispositivos móveis no Sistema Brasileiro de Televisão Digital (SBTVD) (COMUNIDADE GINGA, 2012).

Uma vez que a linguagem Lua também possibilita uma fácil integração com outras formas de interface com o usuário (local ou via rede de computadores) além de NCL para TV Digital, o grupo decidiu desenvolver outros aplicativos codificados em Lua, cada um com três possibilidades de distribuição e interfaces com o usuário: local, rede via TV Digital e rede via computadores.

Nessa linha, foi desenvolvido um algoritmo que realiza conversões entre os formatos mais comumente utilizados para expressar funções lógicas, seguido da sua codificação em Lua (SANTOS & DE LA VEGA, 2011).

Em complementação, foi implementado também um algoritmo para minimização de circuitos digitais sequenciais (COSTA & DE LA VEGA, 2012 – submetido).

Dando continuidade à construção do pacote de ferramentas didáticas, esse trabalho descreve o desenvolvimento das interfaces com o usuário, via rede de computadores, para os aplicativos implementados.



3. ESCOLHA DAS LINGUAGENS

A linguagem de programação Lua (IERUSALIMSKY, 2006) (IERUSALIMSKY, 2012) foi naturalmente escolhida por uma série de motivos.

Inicialmente, ela já havia sido utilizada na implementação do algoritmo de minimização realizada pelo grupo (BEPPU et al., 2010), devido à sua compatibilidade com a linguagem NCL e à sua adoção no SBTVD (Ginga - NCL). O objetivo foi incorporar tal implementação no manual sobre o algoritmo, também implementado pelo grupo (MATTOS et al., 2009), a fim de aumentar o grau de interatividade do mesmo.

Em seguida, foi percebido que podem ser implementados diferentes tipos de interfaces com o usuário, os quais são facilmente compatíveis com Lua. O primeiro deles é a própria linguagem NCL (SOARES & BARBOSA, 2009) (COMUNIDADE GINGA, 2012), para aplicações em TV Digital. Para uso local (*stand alone*), pode-se associar o código Lua a uma interface gráfica local (PUC-RIO, 2012). O terceiro tipo, para uso via rede de computadores, pode ser obtido ao se utilizar a linguagem de ligação CGI Lua (KEPLER PROJECT, 2012), que permite associar código Lua a formulários e a páginas de Internet (*webpages*).

Utilizando-se a linguagem Lua, podem ser desenvolvidas interfaces para Internet através do Projeto Kepler, *software* aberto, escrito completamente em Lua, que fornece uma plataforma de desenvolvimento portátil e extensível, criada para ser utilizada com a maioria dos ambientes de servidores utilizados atualmente.

O CGI Lua é uma ferramenta que integra o Projeto Kepler e permite a criação de páginas de Internet dinâmicas, utilizando uma mistura de código HTML (*HyperText Markup Language*) e código Lua. Ele é capaz de receber arquivos ou dados inseridos em formulários HTML no navegador do usuário e processá-los em um interpretador Lua localizado no servidor, oferecendo acesso a bancos de dados, criptografia, entrada e saída de dados, bem como é capaz de gerar uma página HTML que será exibida no navegador do usuário.

4. INTERFACES DESENVOLVIDAS

Até o momento, compondo o pacote de aplicativos, foram implementados dois algoritmos (BEPPU et al., 2010) (COSTA & DE LA VEGA, 2012 – submetido) e foi desenvolvido um terceiro (SANTOS & DE LA VEGA, 2011).

Uma vez que o resultado do Algoritmo de Quine-McCluskey é uma expressão mínima, a sua implementação, denominada de QM, foi anexada à implementação do conversor de expressões desenvolvido, que foi chamado de LFE. Assim, a denominação LFE passou a significar a união dos aplicativos “LFE + QM”.

Por outro lado, a implementação do Algoritmo de Paul-Unger, denominada de PU, foi mantida separada das demais.

Por essa razão, foram desenvolvidos dois conjuntos de interfaces. Um para o conjunto de aplicativos LFE e outro para o aplicativo PU.

Procurou-se elaborar interfaces de uso simples e amigável, com dois únicos objetivos: i) obter do usuário os dados necessários para a execução das ferramentas e ii) apresentar o resultado da sua execução em um formato adequado.



As interfaces desenvolvidas em CGI Lua são baseadas principalmente em formulários e tabelas HTML, juntamente com código Lua. Ao acessar as interfaces (*webpages*), o usuário é apresentado a um ou mais formulários, onde pode selecionar opções e inserir os dados necessários à execução de cada aplicativo. A informação fornecida pelo usuário é enviada ao servidor, onde são processados tanto o conteúdo HTML quanto o conteúdo Lua. O conteúdo Lua é responsável por ativar a execução do aplicativo em questão. Os dados que resultam da execução são retornados para o usuário na forma de páginas HTML simples.

A tela inicial da interface para o LFE fornece opções de escolha para o número de variáveis da função lógica (2 até 6) e para o tipo de expressão de entrada (tabela verdade, mapa de Karnaugh, lista de mintermos + lista de *don't cares*, lista de maxtermos + lista de *don't cares*, soma de produtos e produto de somas). Em seguida, é apresentada uma nova tela onde podem ser inseridos os dados relevantes para as escolhas efetuadas. Finalmente, os dados são submetidos ao servidor, processados pelo aplicativo e, como resultado, é retornada uma tela com todos os tipos de expressão para funções lógicas, relacionados com a função lógica fornecida pelo usuário.

A tela inicial da interface para o PU fornece opções de escolha para o tipo de máquina de estados (Moore ou Mealy), para o número de variáveis de entrada aplicadas à máquina e para o número de estados que máquina apresenta. Em seguida, é apresentada uma nova tela onde podem ser inseridos os dados relativos à tabela de estados da máquina especificada. Finalmente, os dados são submetidos ao servidor, processados pelo aplicativo e, como resultado, é retornada uma tela contendo a tabela de estados mínima relativa à máquina de estados definida pelo usuário.

5. EXEMPLOS DE OPERAÇÃO DAS INTERFACES

O número de variações de telas que podem ser geradas pelas interfaces inviabilizam sua exposição completa no corrente texto. Assim, é apresentado aqui apenas um resumo de alguns exemplos de operação das interfaces, os quais ilustram algumas das possibilidades básicas.

5.1. Exemplos para o aplicativo LFE

A seguir, são apresentadas algumas telas referentes à interface com o usuário, via rede de computadores, para o aplicativo de conversão de expressões de funções lógicas (LFE);

A Figura 1 apresenta a tela inicial para o LFE, onde foram selecionadas as opções de 3 variáveis e de entrada do tipo Mapa de Karnaugh.

A Figura 2 mostra a tela seguinte, com o mapa de Karnaugh preenchido para a função lógica dada por $F(A,B,C) = (A \text{ xor } B \text{ xor } C)$, onde “xor” é operador “OU-EXCLUSIVO”.

A Figura 3 ilustra a tela contendo o resultado da execução do aplicativo.

As três figuras exemplificam um fluxo de operação para o conjunto Interface-Applicativo.

As figuras subsequentes usarão, como exemplo, a mesma função lógica.



Figura 1 – Exemplo de tela inicial da interface para o aplicativo LFE, com número de variáveis igual a 3 e entrada do tipo Mapa de Karnaugh.

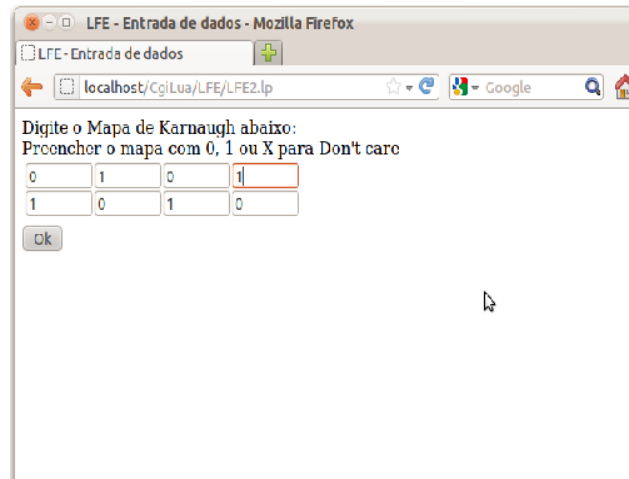


Figura 2 – Exemplo de tela para entrada de dados via Mapa de Karnaugh.

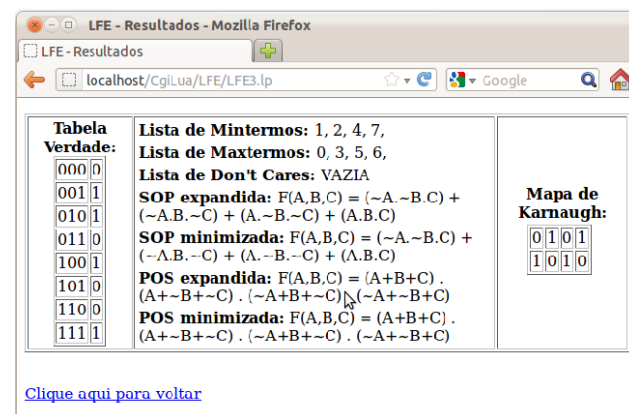


Figura 3 – Exemplo de tela de resultado para o LFE.



As Figuras 4 e 5 apresentam alternativas para a tela de entrada de dados, tendo sido selecionadas as opções de soma de produtos (SOP) e de produto de somas (POS), respectivamente.

LFE - Entrada de dados - Mozilla Firefox

localhost/CgiLua/LFE/LFE2.lp

Digite a SOP abaixo:

(~A)	*	(~B)	*	C	+
(~A)	*	B	*	(~C)	+
A	*	(~B)	*	(~C)	+
A	*	B	*	C	+
()	*	()	*	()	+
()	*	()	*	()	+
()	*	()	*	()	+
()	*	()	*	()	+

Ok

Figura 4 – Exemplo de tela alternativa para entrada de dados: soma de produtos (SOP).

LFE - Entrada de dados - Mozilla Firefox

localhost/CgiLua/LFE/LFE2.lp

Digite a POS abaixo:

A	+	B	+	C	*
A	+	(~B)	+	(~C)	*
(~A)	+	B	+	(~C)	*
(~A)	+	(~B)	+	C	*
()	+	()	+	()	*
()	+	()	+	()	*
()	+	()	+	()	*
()	+	()	+	()	*

Ok

Figura 5 – Exemplo de tela alternativa para entrada de dados: produto de somas (POS).



5.2. Exemplos para o aplicativo PU

A seguir, são apresentadas algumas telas referentes à interface com o usuário, via rede de computadores, para o aplicativo de minimização de máquinas de estado (PU).

A Figura 6 apresenta a tela inicial para o PU, onde foram selecionadas as opções de Máquina de Moore, de 2 variáveis de entrada e de 3 estados.

A Figura 7 mostra a tela seguinte, com a tabela de estados preenchida de tal forma que não haverá simplificações.

Figura 6 – Exemplo de tela inicial da interface para o aplicativo PU, com máquina do tipo Moore, número de variáveis de entrada igual a 2 e número de estados igual a 3.

Vetor de estados Qn	Matriz de próximo estado				Saída atual
	00	01	10	11	
a	a	a	b	b	1
b	b	b	c	c	1
c	c	c	a	b	0

Figura 7 – Exemplo de tela para entrada da tabela de estados.



6. CONCLUSÃO E TRABALHOS FUTUROS

Um pacote de aplicativos didáticos dedicado ao ensino de Circuitos Digitais, incluindo as interfaces com o usuário para tais aplicativos, encontra-se em desenvolvimento pelo grupo PET-Tele. Em relação aos integrantes do grupo, o objetivo é fazer com que eles colaborem com a melhoria das práticas de ensino. Para os alunos do curso, o objetivo é fornecer a eles ferramentas auxiliares para estudo e para a verificação de resultados.

Inicialmente, foi elaborado um manual interativo multimídia para o algoritmo básico de minimização de funções booleanas conhecido por Algoritmo de Quine-McCluskey, codificado em NCL e empregando diferentes mídias (áudio, texto e imagem).

A partir do manual desenvolvido, e visualizando a possibilidade de implementar um pacote de aplicativos, com diversas opções de interação (TV Digital, local ou via rede de computadores), o grupo implementou o Algoritmo de Quine-McCluskey, utilizando a linguagem de programação Lua.

Dando prosseguimento, foi desenvolvido um aplicativo que realiza conversões entre as expressões de funções lógicas mais comumente utilizadas, também utilizando a linguagem de programação Lua. Uma vez que a expressão mínima também é uma forma de expressão para uma função lógica, o aplicativo de minimização anteriormente desenvolvido foi incorporado no atual aplicativo de conversão.

Foi realizada ainda a implementação do algoritmo de minimização de estados conhecido como Algoritmo de Paul-Unger, empregado no ensino de Circuitos Digitais, agora incorporado ao pacote de aplicativos didáticos, e aguardando publicação.

Além disso, o grupo PET-Tele já iniciou o desenvolvimento de interfaces com o usuário, a fim de oferecer uma versão NCL (para TV Digital), uma versão para uso local e uma versão para uso via rede de computadores para o pacote de aplicativos didáticos.

Esse trabalho apresentou o desenvolvimento das interfaces para uso via rede de computadores, utilizando a linguagem de programação CGI Lua.

AGRADECIMENTOS

O grupo PET-Tele faz parte do Programa de Educação Tutorial (PET), financiado pelo Ministério da Educação (MEC).

O grupo agradece aos alunos da disciplina “Circuitos Digitais” que ajudaram a testar o aplicativo desenvolvido.

REFERÊNCIAS BIBLIOGRÁFICAS

BEPPU, M.M.; AMARAL, V.R.L.; DE LA VEGA, A.S. Ferramenta de Auxílio Didático: Algoritmo de Quine-McCluskey em Lua. **Anais: XXXVIII - Congresso Brasileiro de Educação em Engenharia – COBENGE**. Fortaleza, v.1, n.1, p. 1-7, 2010.

COMUNIDADE GINGA. **Ginga-NCL**. Disponível em: <<http://www.gingancl.org.br>>
Acesso em: 01 mai. 2012.



COSTA, B.M.; DE LA VEGA, A.S. Desenvolvimento de Ferramenta de Auxílio Didático: Implementação de Algoritmo para Minimização de Máquinas de Estados em Lua. **Anais: XL - Congresso Brasileiro de Educação em Engenharia – COBENGE**. Belém, 2012 - submetido.

HILL, F. J.; PETERSON, G. R. **Introduction to Switching Theory and Logical Design**. 3.ed. New York: John Wiley, 1981. 636 p, il.

IERUSALIMSKY, R. **Programming in Lua**. 2. ed. Rio de Janeiro: Lightning Source Inc, 2006. 328 p, il.

IERUSALIMSKY, R. **The Programming Language Lua**.
Disponível em: <<http://www.lua.org>> Acesso em: 01 mai. 2012.

KEPLER PROJECT. **CGILua: Building Web Scripts with Lua**.
Disponível em: <<http://keplerproject.github.com/cgilua>> Acesso em: 01 mai. 2012.

MATTOS, H.; SOUZA, T.; DE LA VEGA, A.S.; SAADE, D.M. Ferramenta Didática Interativa Utilizando a Linguagem NCL: Algoritmo de Quine-McCluskey. **8th International Information and Telecommunication Technologies Symposium**, Florianópolis, v.1, n.1, p. 1-4, 2009.

MCCLUSKEY, E. J. Minimization of Boolean Functions. **Bell System Tech. J.**, New Jersey, v.35, n.5., p. 1417-1444, 1956.

Paul, M.C.; Unger, S.H. Minimizing de Number of States in Incompletely Specified Sequential Switching Functions. **IRE Transactions on Electronic Circuits**, v.EC-8, n.3, p. 356-357, 1959.

MINISTÉRIO DA EDUCAÇÃO. **Apresentação – PET**. Disponível em:
<http://portal.mec.gov.br/index.php?option=com_content&view=article&id=12223&ativo=481&Itemid=480> Acesso em: 01 mai. 2012.

PAUL, M.C.; UNGER, S.H. Minimizing de Number of States in Incompletely Specified Sequential Switching Functions. **IRE Transactions on Electronic Computers**, v.EC-8, n.3, p. 356-357, 1959.

PET-TELE. **PET – Engenharia de Telecomunicações da UFF**.
Disponível em: <<http://www.telecom.uff.br/pet>> Acesso em: 01 mai. 2012.

PUC-RIO. **Tecgraf – Tecnologia em Computação Gráfica**.
Disponível em: <<http://www.tecgraf.puc-rio.br>> Acesso em: 01 mai. 2012.

RHYNE, V. T. **Fundamentals of Digital Systems Design**. New York: Prentice-Hall, 1973. 560 p, il.



SANTOS, B.S.; DE LA VEGA, A.S. Ferramenta de Auxílio Didático: Conversão de Expressões de Funções Lógicas em Lua. **Anais: XXXIX - Congresso Brasileiro de Educação em Engenharia – COBENGE**. Blumenau, 2011.

SOARES, L. F. G.; BARBOSA S. D. J. **Programando em NCL 3.0: Desenvolvimento de aplicações para o middleware Ginga**. 1. ed. Rio de Janeiro: Campus, 2009. 341 p, il.

WIKIPEDIA. **ISDB-T International – Wikipedia, the free encyclopedia**.

Disponível em: <http://en.wikipedia.org/wiki/ISDB-T_International> Acesso em: 01 mai. 2012.

DEVELOPMENT OF A LEARNING AID TOOL: USER INTERFACE IMPLEMENTATION USING CGILUA

Abstract: *The aim of this job was to develop Internet User Interfaces by using CGI Lua programming language. The initial motivation to develop such an interfaces was to include them into a learning aid tools package that is under construction by the PET-Tele group: logic function expressions, logic function minimization and state machine minimization. The package is intended to be used on Digital Circuits learning.*

Key-words: *Learning aid tool, Distance education, Digital Circuits, User Interface, CGI Lua Programming Language.*