

UNIFICAÇÃO DE MÚLTIPLOS OBJETIVOS DIDÁTICOS POR MEIO DE PROJETO DE CIRCUITO DIGITAL COMBINACIONAL SIMPLES

*Primeiro Autor – e-mail**

*Instituição de Ensino, Faculdade ou Departamento**

*Endereço**

*CEP – Cidade – Estado**

*Segundo Autor – e-mail**

*Instituição de Ensino, Faculdade ou Departamento**

*Endereço**

*CEP – Cidade – Estado**

*Terceiro Autor – e-mail**

*Instituição de Ensino, Faculdade ou Departamento**

*Endereço**

*CEP – Cidade – Estado**

Resumo: *Visando a sua aplicação no ensino de Circuitos Digitais, em períodos iniciais de cursos de graduação e/ou em cursos que não possuam conteúdo de eletrônica analógica, o presente trabalho descreve o projeto de um circuito digital combinacional simples. O projeto em questão visa atender, de forma unificada, a múltiplos objetivos didáticos, por meio da abordagem dos seguintes tópicos: projeto hierárquico e projeto modular (diferentes metodologias de projeto), uso de lógica combinacional simples ao invés do emprego de chaves analógicas ou multiplexadores (alternativas de solução e soluções otimizadas), solução não numérica para um problema originalmente numérico (quebra de paradigma e diversidade de soluções), simplificação conjunta de funções lógicas (variações de técnicas básicas), contraexemplo para o uso de don't cares e para a seleção de implicantes primos na simplificação de funções lógicas (identificação de condicionantes fictícios), bem como o uso de diferentes representações de circuitos (esquemático e linguagem de descrição de hardware) em ambientes computacionais de projeto (identificação de prós e contras em decisões de projeto).*

Palavras-chave: *Programa de Educação Tutorial (PET). Unificação de múltiplos objetivos didáticos. Quebra de paradigma. Diversidade de soluções. Ensino de Circuitos Digitais.*

1 INTRODUÇÃO

O Programa de Educação Tutorial (PET) (MINISTÉRIO DA EDUCAÇÃO, 2018) exige que os bolsistas dos seus grupos, ao serem submetidos a uma formação complementar, desenvolvam atividades que possuam, conjuntamente, itens relativos às áreas de Pesquisa, Ensino e Extensão, que consigam algum tipo de penetração no curso ao qual pertencem e que realizem trabalhos de cooperação com outros grupos, ligados ou não ao seu curso de origem. Logo, o PET busca atitudes inovadoras em educação.

Como parte de uma das atividades do Planejamento de 2018, que visa promover uma conexão entre as disciplinas de Circuitos Digitais e de Processamento Digital de Sinais, o grupo PET do Curso de Engenharia de Telecomunicações da Universidade Federal Fluminense (PET-Tele/UFF) (PET-TELE, 2018) realizou um grupo de estudos sobre projetos de circuitos digitais e uma série de objetivos didáticos a serem trabalhados durante a atividade de projeto (HILL & PETERSON, 1981), (IDOETA & CAPUANO, 2008), (PEDRONI, 2010), (RHYNE, 1973), (TAUB, 1982). O presente trabalho é fruto desse grupo de estudos e trata da unificação de múltiplos objetivos didáticos por meio de projeto de circuito digital combinacional simples.

O projeto proposto é apresentado a seguir. A Seção 2 trata da unificação dos múltiplos objetivos didáticos. A apresentação dos problemas e o encaminhamento das soluções pertinentes ao projeto são abordados na Seção 3. A Seção 4 descreve o desenvolvimento do projeto. Finalmente, as considerações finais são realizadas na Seção 5.

2 UNIFICAÇÃO DOS MÚLTIPLOS OBJETIVOS DIDÁTICOS

Ao se desenvolver o exemplo de projeto em questão, levou-se em consideração que o público alvo seria um determinado tipo de aluno e procurou-se atender, de forma unificada, a múltiplos objetivos didáticos, por meio da abordagem dos seguintes tópicos: projeto hierárquico e projeto modular (*diferentes metodologias de projeto*), uso de lógica combinacional simples ao invés do emprego de chaves analógicas ou multiplexadores (*alternativas de solução e soluções otimizadas*), solução não numérica para um problema originalmente numérico (*quebra de paradigma e diversidade de soluções*), simplificação conjunta de funções lógicas (variações de técnicas básicas), contraexemplo para o uso de *don't cares* e para a seleção de implicantes primos na simplificação de funções lógicas (*identificação de condicionantes fictícios*), bem como o uso de diferentes representações de circuitos (esquemático e linguagem de descrição de hardware) em ambientes computacionais de projeto (*identificação de prós e contras em decisões de projeto*).

Utilizando o exemplo dos cursos de graduação em Ciência da Computação e em Engenharia de Telecomunicações, da Universidade Federal Fluminense (UFF), ao se ministrar a disciplina de Circuitos Digitais em períodos iniciais de um curso de graduação e/ou em cursos que não possuam conteúdo de eletrônica analógica, o nível de abstração mais próximo de uma implementação física que se consegue alcançar é o do emprego de chaves mecânicas ou eletromecânicas. Assim, é aparentemente mais confortável para o aluno trabalhar no nível de abstração das portas lógicas.

Dependendo do tipo de circuito digital a ser projetado, diferentes metodologias de projeto podem ser utilizadas. Para circuitos de baixa complexidade, pode-se adotar o processo sistemático da tradução da especificação em uma tabela verdade, dessa em um mapa de Karnaugh e do uso desse na busca de um equação mínima. Para alguns tipos de circuitos, pode-se adotar uma técnica de escalamento da solução, iniciando-se com soluções contendo um número pequeno de variáveis de entrada e, em seguida, inferindo-se uma solução para um número genérico de variáveis de entrada. Em alguns tipos de problemas, torna-se necessário o emprego da técnica de *dividir-para-conquistar*, particionando-se o problema original em problemas menores e mais fáceis de serem gerenciados. Esse tipo de abordagem naturalmente leva a um tipo de projeto hierárquico, onde um nível hierárquico de solução mais complexo utiliza-se de níveis de soluções mais simples. Nesse caso, os circuitos são encapsulados e utilizados como elementos constituintes. Um caso particular desse tipo de abordagem é o projeto modular.

Alguns problemas envolvendo roteamento de sinais podem ser diretamente solucionados com o uso de chaves analógicas, o que pode representar a solução mais simples do ponto de vista físico, ou de multiplexadores, o que representa a solução digital equivalente às chaves analógicas, mas nem sempre otimizada. Porém, do ponto de vista de soluções alternativas e de soluções otimizadas,

pode-se tentar uma solução que empregue portas lógicas genéricas e uma metodologia de projeto segmentado (hierárquico e/ou modular).

Usualmente, ao se apresentar um problema de origem numérica, os alunos naturalmente apresentam uma solução que envolve blocos de circuitos com funções numéricas. A fim de estimular as ideias de *quebra de paradigma* e de *busca por uma diversidade de soluções*, pode-se tentar propor soluções alternativas com características diferentes das originais, simplesmente olhando-se o problema a partir de uma ótica diferente.

Na tarefa de simplificação de várias funções lógicas, naturalmente costuma-se adotar uma simplificação independente para cada função. Uma variação dessa técnica básica é a simplificação conjunta de funções lógicas. Em alguns casos, o processo de simplificação conjunta das funções pode apresentar uma simplificação global maior do que no caso isolado. Quase sempre, na tentativa de se mostrar o ganho do processo de simplificação conjunta, são utilizados exemplos complexos. Por essa razão, usualmente evita-se abordar tal assunto em cursos introdutórios de circuitos digitais. Porém, pode-se questionar que, se ao invés de evitar o assunto, não se poderia fazer uso de exemplos mais simples e estimular a curiosidade dos alunos.

Por vezes, ao se abordar um determinado assunto, são definidas algumas recomendações genéricas e básicas que, erroneamente, passam a ser assumidas como regras imutáveis. Isso costuma acontecer quando se parte de condicionantes que, na realidade, não existem. Tecnicamente, tais elementos são denominados de *condicionantes fictícios*. Esse tipo de bloqueio pode impedir a visualização de soluções simples e, portanto, deve-se estar sempre atento para que eles não ocorram. Para o conteúdo em questão, isso pode ser exemplificado por meio de contraexemplo para o uso de *don't cares* e para a seleção de implicantes primos na simplificação de funções lógicas.

No uso de ambientes integrados de desenvolvimento (*Integrated Development Environment* ou IDE) para a realização de projetos e de testes, diversas formas de representação para um circuito podem ser aceitas. Além de se ter o conhecimento sobre como utilizá-las, é importante que se tenha o discernimento sobre os prós e os contras no uso de cada representação disponível. As representações usualmente utilizadas são o desenho esquemático do circuito e o uso de uma linguagem de descrição de *hardware* (*Hardware Description Language* ou HDL).

Portanto, procurando ser útil à abordagem dos tópicos citados acima, foi desenvolvido o projeto descrito nas próximas seções.

3 APRESENTAÇÃO DOS PROBLEMAS E ENCAMINHAMENTO DAS SOLUÇÕES

Em cursos fundamentais de Circuitos Digitais, dois problemas com baixa complexidade e aparentemente independentes entre si são comumente apresentados como exemplos de projetos: 1) a separação dos *bits* com valores 1 e 0 em uma dada palavra digital e 2) a relação entre as quantidades de *bits* com valores 1 e 0 em uma dada palavra digital.

O primeiro problema consiste em, dada uma palavra digital de N *bits*, gerar uma nova palavra de N *bits*, com os *bits* de valor 1 agrupados de um lado da palavra e os *bits* de valor 0 do outro lado. Por sua vez, dada uma palavra digital com N *bits*, o segundo problema envolve a geração de dois ou três sinais que indiquem a relação entre as quantidades de *bits* com valores 1 e 0 (*menos_que*, *igual_a* e *mais_que*).

Em uma análise preliminar, o primeiro problema pode ser identificado como um problema de classificação (separação), podendo-se sugerir uma solução que empregue circuito combinacional, envolvendo um roteamento de *bits* para as posições desejadas. Dessa forma, as soluções mais comumente sugeridas envolvem o uso de chaves analógicas ou de multiplexadores. Por outro lado, pelo fato do segundo problema envolver quantidades de *bits*, o senso comum conduz a propostas de solução que empregam blocos numéricos, possivelmente com o uso de circuitos sequenciais.

Uma análise mais cuidadosa, que leve em consideração o que foi abordado na Seção 2, pode mostrar que: 1) uma solução usando apenas portas lógicas básicas, e envolvendo um projeto modular, pode ser proposta para o primeiro problema, 2) uma solução não numérica, e empregando apenas circuitos combinacionais, pode ser proposta para o segundo problema e 3) os problemas, embora considerados independentes entre si na sua origem, podem ser empregados de forma conjunta na proposta de uma solução para o segundo problema. Essas ideias são desenvolvidas na próxima seção.

4 DESENVOLVIMENTO DO PROJETO

A seguir, adotando-se uma metodologia de projeto hierárquico, são apresentadas soluções modulares escaláveis, que utilizam apenas circuitos combinacionais, baseados em portas lógicas, para os dois problemas definidos na Seção 3, quais sejam: 1) o bloco funcional Separa *Bits* e 2) o bloco funcional Relaciona Quantidades.

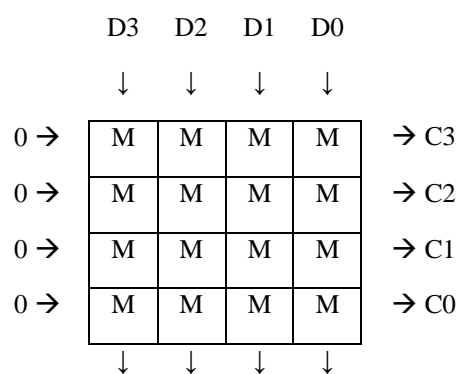
4.1 Bloco funcional Separa *Bits*

A título de exemplo, e sem perda da generalidade, será considerada uma palavra digital de entrada com $N = 4$ bits, identificada por $\mathbf{D} = \{D3 D2 D1 D0\}$, onde os índices 3 e 0 indicam os dígitos mais e menos significativos, respectivamente. Por sua vez, a palavra digital de saída será identificada por $\mathbf{C} = \{C3 C2 C1 C0\}$.

Será assumido, para essa solução, que os bits da palavra \mathbf{D} com valor lógico 1 deverão ser deslocados para os bits mais significativos da palavra \mathbf{C} .

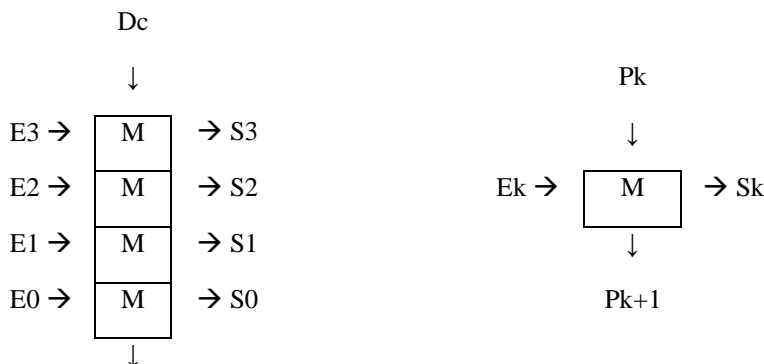
Uma estrutura modular para o bloco funcional é apresentada na Figura 1. Nela, cada célula M representa uma unidade funcional básica. Os valores lógicos 0, injetados à esquerda do bloco, bem como as saídas presentes na parte de baixo do bloco, são necessários para manter a modularidade no projeto das células, como será explicado a seguir.

Figura 1 – Estrutura modular para o bloco funcional Separa *Bits*.



As estruturas de cada coluna e de cada célula M podem ser vistas na Figura 2. A estrutura da coluna justifica a inserção de valores lógicos 0 à esquerda do bloco funcional. Por sua vez, a estrutura da célula justifica as saídas presentes no fundo do bloco funcional.

Figura 2 – Estruturas da coluna e da célula M, do bloco funcional *Separa Bits*.



De forma a cumprir a especificação aqui definida para o bloco funcional, as saídas da célula M deverão atender às especificações ilustradas na Figura 3.

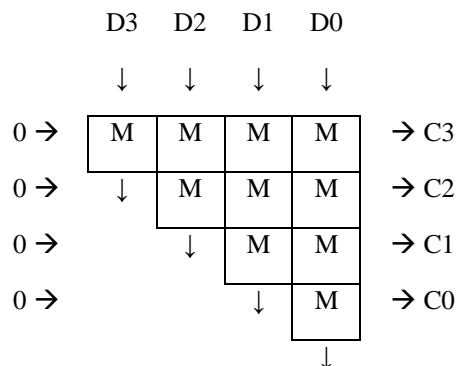
Figura 3 – Especificação da célula M que atende à especificação do bloco funcional.

Ek	Pk	Sk
0	0	0
0	1	1
1	0	1
1	1	1

Ek	Pk	Pk+1
0	0	0
0	1	0
1	0	0
1	1	1

Por fim, observando-se que as saídas das células M da metade inferior e à esquerda do bloco funcional sempre assumirão o valor lógico 0, a estrutura pode ser otimizada, assumindo a forma encontrada na Figura 4. Nota-se uma economia de $[N(N-1) / 2]$ células M.

Figura 4 – Estrutura otimizada para o bloco funcional *Separa Bits*.



4.2 Bloco funcional Relaciona Quantidades

Dada uma palavra digital com N bits, a função do bloco funcional Relaciona Quantidades é gerar dois ou três sinais que indiquem a relação entre as quantidades de bits com valores 1 e 0 (*menos_que*, *igual_a* e *mais_que*).

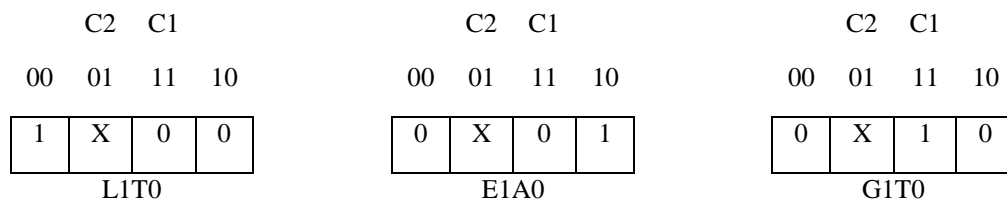
Apesar de ser um bloco cuja função envolve a análise da quantidade de bits de um determinado valor e da comparação das quantidades de bits com valores 1 e 0, a função deste bloco pode ser realizada de uma maneira puramente combinacional.

Dada uma palavra digital D e supondo-se o bloco funcional Separa Bits, com entrada D e saída C , pode-se relacionar as quantidades de bits com os valores 1 e 0 de D simplesmente analisando-se de forma lógica os bits do meio da palavra C .

A título de exemplo, e sem perda da generalidade, serão considerados os casos de uma palavra digital com $N_p = 4$ bits e com $N_i = 5$ bits. Será considerada ainda a geração dos seguintes sinais: *menos_1_que_0* (L1T0), *1_igual_a_0* (E1A0) e *mais_1_que_0* (G1T0).

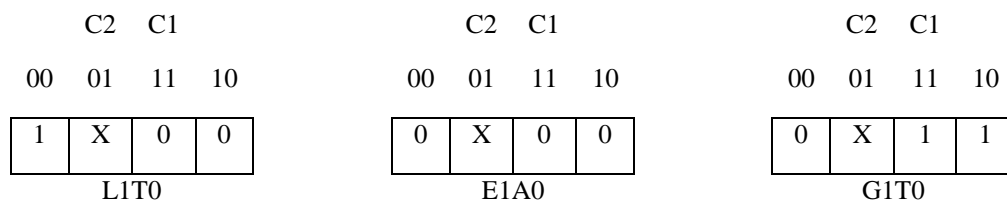
Utilizando-se os bits Cmeio e Cmeio-1, que são os bits C2 e C1, para o caso de $N_p = 4$ bits, podem ser montados os mapas de Karnaugh mostrados na Figura 5. Dos respectivos mapas, podem ser obtidas as seguintes funções: $E1A0 = (\text{NOT } L1T0) \text{ AND } (\text{NOT } G1T0) = C2 \text{ AND } (\text{NOT } C1)$, $L1T0 = (\text{NOT } C2)$ e $G1T0 = C1$.

Figura 5 – Mapas de Karnaugh para o caso de $N_p = 4$ bits.



Utilizando-se os bits Cmeio e Cmeio-1, que são os bits C2 e C1, para o caso de $(N_i - 1) = 4$ bits, podem ser montados os mapas de Karnaugh mostrados na Figura 6. Dos respectivos mapas, podem ser obtidas as seguintes funções: $E1A0 = (\text{NOT } L1T0) \text{ AND } (\text{NOT } G1T0) = 0$, $L1T0 = (\text{NOT } C2)$ e $G1T0 = C2$.

Figura 6 – Mapas de Karnaugh para o caso de $N_i = 5$ bits.



Das equações lógicas obtidas para os casos de N par e de N ímpar, conclui-se que, utilizando-as na sua forma original, não é possível adotar uma solução modular que atenda a ambos os casos, pois as equações de G1T0 são diferentes.

Uma primeira solução para unificar as funções seria usar um multiplexador, controlado por um sinal binário extra, que identificasse os dois casos.

Uma outra solução pode ser proposta, utilizando-se as próprias características do problema. Observando-se o mapa de Karnaugh de $G1T0$ para o caso de N ímpar, pode-se propor a equação $G1T0 = C2 \text{ OR } C1$, embora isso seja contra a simplificação de função. Utilizando-se o *bit* $Cmsb$, que nesse caso é o *bit* $C4=1$, pode-se propor a equação $G1T0 = C2 \text{ OR } C1 = (C4 \text{ AND } C2) \text{ OR } C1$. Para o caso de N par, $C4$ não existe. Porém, pode-se adotar $C4=0$ e utilizar a mesma equação, de tal forma que $G1T0 = C1 = (C4 \text{ AND } C2) \text{ OR } C1$.

Uma forma menos empírica de se obter a segunda solução é assumir a existência do *bit* $Cmsb=0$ para o caso de N par e assumir que os sinais $L1T0$, $E1A0$ e $G1T0$, dependem dos sinais $Cmsb$, $Cmeio$ e $Cmeio-1$. Isso conduz aos mapas de Karnaugh que são apresentados nas Figuras 7 e 8. Realizando-se uma simplificação conjunta dos mapas relativos aos casos par e ímpar, por meio da superposição dos mapas, as funções definidas acima aparecem naturalmente. Deve-se ressaltar que os mesmos mapas, analisados de forma independente, conduziram a simplificações completamente diferentes, as quais resultariam nas funções inicialmente definidas.

Figura 7 – Mapas de Karnaugh para o caso de $N_p = 4$ bits.

		C2	C1		
		00	01	11	10
C4	0	1	X	0	0
	1	X	X	X	X
		L1T0			

		C2	C1		
		00	01	11	10
C4	0	0	X	0	1
	1	X	X	X	X
		E1A0			

		C2	C1		
		00	01	11	10
C4	0	0	X	1	0
	1	X	X	X	X
		G1T0			

Figura 8 – Mapas de Karnaugh para o caso de $N_i = 5$ bits.

		C2	C1		
		00	01	11	10
C4	0	X	X	X	X
	1	1	X	0	0
		L1T0			

		C2	C1		
		00	01	11	10
C4	0	X	X	X	X
	1	0	X	0	0
		E1A0			

		C2	C1		
		00	01	11	10
C4	0	X	X	X	X
	1	0	X	1	1
		G1T0			

4.3 Tipos de descrição dos circuitos

Na utilização de ambientes destinados ao projeto e a testes de circuitos digitais, tais como MaxPlusII / QuartusII (ALTERA, 2018), ModelSim (MODELSIM, 2018) e Qucs (QUCS, 2018), é importante que se trabalhem as diferentes formas de descrição dos circuitos aceitas pela ferramenta em questão.

Deve-se ressaltar que, embora, para alguns alunos, a descrição por meio de desenhos esquemáticos apresente uma forma mais clara de se compreender a estrutura geral do circuito, a mesma apresenta uma dificuldade natural de escalamento do projeto, uma vez que, nesse caso, blocos deverão ser incluídos ou excluídos manualmente. O mesmo pode ser dito para pequenas modificações e/ou manutenções do circuito projetado.

Por sua vez, a utilização de uma HDL pode ser bastante útil para que se obtenha um escalamento automático, por meio de uma simples redefinição de parâmetros. Da mesma forma,

pequenas alterações no circuito podem ser conseguidas por meio de alterações de algumas linhas de código. Porém, o uso de uma HDL exige do aluno uma compreensão mais abstrata do que visual, bem como o domínio de linguagens de programação, além do fato da HDL não ser uma linguagem de programação convencional, mas sim uma linguagem de construção automática de *hardware*.

5 CONSIDERAÇÕES FINAIS

A fim de cumprir uma das atividades do Planejamento de 2018, que visa promover uma conexão entre as disciplinas de Circuitos Digitais e de Processamento Digital de Sinais, o grupo PET-Tele da UFF (PET-TELE, 2018) realizou um grupo de estudos sobre projetos de circuitos digitais e uma série de objetivos didáticos a serem trabalhados durante a atividade de projeto. O presente trabalho é fruto desse grupo de estudos e trata da unificação de múltiplos objetivos didáticos por meio de projeto de circuito digital combinacional simples.

O projeto já foi incorporado em apostila escrita pelo tutor do grupo (DE LA VEGA, 2018), com *download* gratuito, relativa a projetos demonstrativos usando HDL, e está sendo aplicado nas turmas sob a responsabilidade do tutor.

Agradecimentos

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

Os autores agradecem aos demais bolsistas do grupo PET-Tele por ajudarem no desenvolvimento do projeto e na correção do presente artigo.

REFERÊNCIAS

ALTERA. **Altera, Inc.**. Disponível em: <<http://www.altera.com>> Acesso em: 25 abril 2018.

DE LA VEGA, Alexandre S.. **Apostila com Códigos de Programas Demonstrativos para Circuitos Digitais**. Disponível em: <<http://www.telecom.uff.br/~delavega>> Acesso em: 25 abril 2018.

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

IDOETA, I. V.; CAPUANO, F. G.. **Elementos de Eletrônica Digital**. Editora Érica, 2008.

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: 25 abril 2018.

MODELSIM. **ModelSim**. Disponível em: <<http://www.altera.com/products/design-software/model---simulation/modelsim-altera-software.html>> Acesso em: 25 abril 2018.

PEDRONI, V. A.. **Eletrônica Digital Moderna e VHDL**. Rio de Janeiro: Elsevier, 2010.

PET-TELE. **PET – Engenharia de Telecomunicações da UFF**.

Disponível em: <<http://www.telecom.uff.br/pet>> Acesso em: 25 abril 2018.

QUCS. Qucs. Disponível em: <<http://qucs.sourceforge.net/download.html>>
Acesso em: 25 abril 2018.

RHYNE, V. T.. **Fundamentals of Digital Systems Design**. Englewood Cliffs: Prentice-Hall, 1973.

TAUB, H.. **Digital Circuits and Microprocessors**. New York: McGraw-Hill, 1982.

UNIFICATION OF MULTIPLE DIDACTIC AIMS BY MEANS OF A DESIGN OF A SIMPLE COMBINATIONAL DIGITAL CIRCUIT

Abstract: *This work presents the design of a simple combinational digital circuit intended to be used on the beginning periods of undergraduate courses and/or undergraduate courses that do not include analogue electronics as a subject matter. The presented design wishes to embody multiple didactic aims by means of the following topics: hierarchical design and modular design (different design methodologies), usage of simple combinational logic instead of analogue switches or multiplexers (alternative solutions and optimized solutions), non-numerical solution for an originally numerical problem (paradigm rupture and diversity of solutions), jointly minimization of logic functions (changes over fundamental technics), counterexample on the use of don't cares and the prime implicants selection for logic function minimization (non-real constraints identification), as well as the usage of different circuit descriptions (schematic diagrams and hardware description languages) on computational design environments (identification of the pros and cons on hardware description).*

Key-words: *Tutorial Education Program (PET). Unification of multiple didactic aims. Paradigm rupture. Diversity of solutions. Teaching of Digital Circuits.*