

UMA NOVA FRAMEWORK PARA MAPEAMENTO TECNOLÓGICO

JULIO S. DOMINGUES JÚNIOR;
LEOMAR S. DA ROSA JR; FELIPE DE SOUZA MARQUES

Universidade Federal de Pelotas
{jsdomingues,leomarjr,felipem}@inf.ufpel.edu.br

1. INTRODUÇÃO

A área de microeletrônica obteve grande evolução nas últimas décadas, viabilizando o acesso e a popularização da tecnologia a uma fatia maior da sociedade. Essa popularização se deve ao avanço nos processos de fabricação desse tipo de circuito. Neste sentido, a demanda por evolução da tecnologia produzida se torna cada vez maior, por outro lado, o tempo de projeto é cada vez menor. Em outras palavras, o processo de síntese se torna cada vez mais complexo porque novos limites físicos e lógicos são impostos. Dessa forma, é necessário que existam ferramentas de apoio aos projetistas, para que algumas etapas sejam executadas de forma automática (menor tempo) e livre de erros humanos.

Em geral, o projeto de um circuito pode ser dividido em três grandes etapas, são elas: descrição de alto nível, síntese lógica e síntese física. A primeira etapa diz respeito à modelagem do circuito de uma forma de mais alto nível, sem a preocupação com algumas características de implementação. A segunda etapa, chamada de síntese lógica, é focada em tratar a descrição de mais alto nível aplicando algumas otimizações e agregando informações, com a finalidade de gerar uma descrição em nível de portas lógicas. A etapa de síntese lógica é uma das mais importantes, pois define as principais características estruturais e os elementos que serão utilizados para a implementação do circuito em questão. A última etapa, chamada de síntese física, é responsável pelos aspectos geométricos e físicos do circuito como: dimensionamento de transistores, roteamento de células e etc.

Este trabalho aborda a etapa de síntese lógica, mais precisamente uma de suas fases, chamada de mapeamento tecnológico. O mapeamento é considerado uma das fases mais importante, pois ele define quais elementos de uma dada tecnologia serão utilizados para implementar o circuito (Correia,2004). O mapeamento começa com a descrição de alto nível e elementos de uma tecnologia (geralmente uma biblioteca de células). Uma biblioteca de células é um conjunto finito de elementos (portas-lógicas) de uma dada tecnologia. Uma biblioteca de células pode ser estática, onde os elementos são pré-caracterizados eletricamente permitindo grande conhecimento do comportamento das células, ou dinâmica, onde os elementos são projetados sob demanda.

O mapeamento pode ser dividido em três etapas: decomposição, casamento e cobertura. A etapa de decomposição prepara a estrutura de dados para a representação do circuito. A segunda etapa, tenta identificar todos os padrões do circuito (partes do circuito) que são equivalentes aos elementos da tecnologia alvo (portas lógicas ou células). A última etapa elege o melhor conjunto de casamentos (células) para fabricar o circuito. O objetivo do mapeamento, é encontrar a melhor configuração dos elementos da tecnologia alvo para implementar o circuito, minimizando uma função objetivo. Essa função é a minimização de critérios como: atraso, consumo de potência, área.

A etapa de cobertura pode ser efetuada de diversas formas e estratégias, que combinadas com a estrutura de dados, podem facilitar ou dificultar a busca por uma solução ótima. Dentre as estratégias pode-se citar os algoritmos gulosos, programação dinâmica e etc. Essa metodologia apresentada, baseada em biblioteca de células, é chamada de *Standard Cell*. Existem diversas formas assim como ferramentas para executar a síntese de um circuito digital. Cada abordagem, utiliza características diferentes como estrutura de dados, função objetivo, biblioteca de células ou estratégia de cobertura.

Dentre as ferramentas que merecem destaque, existe uma desenvolvida pela Universidade de Berkeley, chamada ABC (Abc,2013). O ABC possui os algoritmos mais atuais no que diz respeito a mapeamento tecnológico. A ferramenta está disponível para *download*, assim como seu código fonte para alterações e acréscimo de novas funcionalidades. Entretanto, para adicionar uma simples funcionalidade ou novo fluxo de mapeamento, é necessário entender e alterar o código fonte da ferramenta. Essa tarefa se torna difícil, pois a ferramenta não teve uma etapa de engenharia de software que permitisse uma organização de código fonte. Além disso, não disponibiliza uma boa documentação do software. Dessa forma, se torna impraticável a rápida prototipação de novos fluxos para mapeamento, ou até mesmo, pequenas alterações.

Neste sentido, este trabalho propõe uma nova ferramenta para mapeamento tecnológico chamada *FlexMap*. O objetivo da ferramenta é proporcionar um *framework* para mapeamento tecnológico, onde seja possível desenvolver novos fluxos ou misturar técnicas existentes sem a necessidade de programação extra, permitindo rápida prototipação de novos fluxos, com a capacidade de configurar cada etapa do mapeamento através de um alto nível de abstração. Assim, espera-se, além de permitir rápidas prototipações, propor futuramente novas abordagens para mapeamento utilizando uma mistura de técnicas permitidas a partir do *framework* desenvolvido. Além disso, espera-se que o *framework* seja capaz de lidar com novas tecnologias emergentes, por exemplo, Nanotubos de carbono e QCA. Essas novas tecnologias podem vir a substituir a atual chamada de *Complementary Metal-Oxide-Semiconductor (CMOS)*, o que acarreta na necessidade de novas metodologias para síntese dos circuitos e ferramentas para auxiliar os projetistas.

2. METODOLOGIA

Esta seção apresenta a modelagem e a engenharia de software feita para o desenvolvimento da ferramenta. Inicialmente, para representação das descrições dos circuitos foram estudados diversos tipos de descrição. A descrição selecionada para ser utilizada como base para o *framework* foi a que representa o circuito através de um grafo com apenas portas primitivas *and* e *inversor*, e por isso chamado de AIG (*And Inverter Graph*). Este tipo de descrição é bem difundida na comunidade de síntese lógica, o que permite um padrão bem especificado e documentado. Um desses padrões adotado é o AIGER, o qual é utilizado em outras ferramentas da área, como o ABC.

O AIG é a descrição sujeito (estrutura de dados) base do *framework*. Embora seja possível carregar descrições diferentes, como o *Equation Format-Eqn*, todas as descrições são convertidas para a descrição sujeito AIG. A próxima etapa foi desenvolver a modelagem da ferramenta utilizando técnicas de engenharia de software. Assim, foi possível desenvolver módulos bem estruturados e de fácil adaptação as necessidades. Um exemplo dessa fácil adaptação foi o uso de padrões de projeto. Padrões de projeto oferecem soluções genéricas e validadas para problemas conhecidos de computação de forma eficiente.

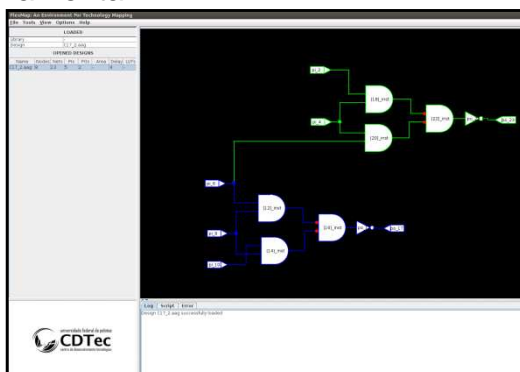
Além da modelagem, foram implementadas as descrições sujeito (estrutura de dados) que são utilizadas nos métodos de mapeamento, como por exemplo: os grafos acíclicos direcionados-DAG e uma especialização chamada de árvores. Outra funcionalidade desenvolvida foi um módulo que permita a definição da função custo no momento do mapeamento. Em outras palavras, a função custo pode ser definida através de identificadores pré-definidos. Com isso, a ferramenta permite que a mesma abordagem para mapeamento considere diferentes características do circuito durante a síntese. A fim de obter uma linha inicial para mapeamento, assim como também validar os módulos desenvolvidos, foi preciso implementar algumas estratégias de cobertura. Neste sentido, três abordagens foram desenvolvidas considerando o foco em tecnologias distintas.

A primeira abordagem é baseada em uma estratégia gulosa, utilizando o algoritmo de *K-cuts* para a etapa de casamentos, e o conceito de *fluxo de área* de Valavan (Manohararajah,2006). Essa abordagem permite a flexibilidade de definir a função custo e também a tecnologia alvo. Uma das alternativas é utilizar o mapeamento voltado para biblioteca de células estáticas, por outro lado é possível direcionar também para a abordagem de *LookUpTables-LUTs*. A LUT é um elemento básico da tecnologia de matrizes reconfiguráveis, presente nos dispositivos chamados *Field Programmable Gate Array-FPGA*.

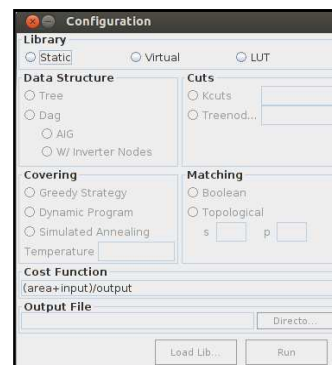
Tendo a primeira abordagem contemplando dois focos distintos, como segunda abordagem o objetivo foi aplicar o mapeamento para outra possibilidade, utilizando bibliotecas virtuais. Neste tipo de abordagem, as células não estão pré-definidas. Então, o mapeamento é guiado através de alguma restrição imposta. Neste sentido, elaborou-se a uma implementação de cobertura utilizando programação dinâmica. Esta abordagem é baseada no algoritmo de proposto por Correia (Correia,2004) e utiliza a topologia das portas-lógicas da tecnologia CMOS como restrição alvo.

Além de permitir o mapeamento com foco em três tecnologias distintas, também é possível reconfigurar para que os mesmos considerem características diferentes. Essa liberdade é proporcionada através da função custo, onde é possível definir com identificadores conhecidos e pré-definidos no *framework*. Essa funcionalidade permite que novos fluxos alternativos sejam testados rapidamente, apenas definindo uma nova função objetivo.

Para permitir uma interface de fácil acesso aos usuários, foi desenvolvido uma interface gráfica inicial da ferramenta. Desse modo a ferramenta apresenta todas as funcionalidades de configuração no modo gráfico. A Figura 1 (a,b) apresenta a interface gráfica da *framework*, onde na Figura 1.a é apresentada uma descrição de circuito carregada. A Figura1.b apresenta a tela de mapeamento, onde é possível verificar todas as possibilidades de configuração da ferramenta.



a) Tela de Apresentação dos Circuitos



b) Configurando Mapeamento

Figura 1. FlexMap Rodando Com Interface Gráfica.

3. RESULTADOS E DISCUSSÃO

Com o objetivo de validar os algoritmos desenvolvidos no *framework*, diversas configurações foram definidas. Um conjunto de 99 circuitos do benchmark ISCAS'85 foram mapeados com o *FlexMap*. Todos os circuitos foram validados através da funcionalidade de equivalência lógica da ferramenta ABC (comando *CEC*). Com isso, garante-se que os métodos implementados estão corretos. Os circuitos foram mapeados tanto no ABC (comando *IF -K 4*), quanto no *FlexMap*. A tecnologia alvo foi FPGAs. O objetivo foi comparar o método implementado em relação a ferramenta ABC. A tabela 1 apresenta os resultados de comparação de alguns circuitos entre os métodos, em relação ao número de LUTs.

Tabela 1. Comparação *FlexMap* x *Abc*

Circuito	Abc	FlexMap	Circuito	Abc	FlexMap
apex5	1266	1222	C3540	379	388
apex6	256	252	C432	85	71
apex7	109	105	C499	74	90
b12	87	79	C5315	502	520
C1355	74	90	i9	513	369
C17	2	2	C7552	577	630
C1908	124	129	C880	122	125
C2670	181	187	Alu2	2039	1916

Embora os resultados da Tabela 1 sejam iniciais, mostram que o *FlexMap* possui métodos com bom potencial. Por outro lado, o método utilizado no ABC foi o método simples e não o iterativo, o qual pode apresentar resultados melhores.

Tabela 2. Estatísticas dos Métodos

	Total Luts	Desvio Padrão	Variância
<i>FlexMap</i>	37.141	671,80	0,70
Abc	37.830	687,69	0,70

A Tabela 2 apresenta o resultado do número total de LUTs utilizados para mapear todos os 99 circuitos, assim como o desvio padrão e a variância das avaliações. Outra avaliação efetuada entre as ferramentas, foi a análise em quantos circuitos o mapeamento pelo *FlexMap* resultou e um número menor de LUTs. Assim, o *FlexMap* apresentou melhores resultados em 50 circuitos dos 99 mapeados. Por outro lado, em 25 circuitos o número foi igual para as duas ferramentas. No restante dos circuitos, em 24, a ferramenta ABC obteve melhor desempenho.

4. CONCLUSÕES

Este trabalho apresentou o desenvolvimento de uma ferramenta para mapeamento tecnológico. A abordagem se difere das demais pois permite flexibilidade nas diversas etapas do mapeamento tecnológico. As avaliações mostraram que os métodos implementados estão corretos. Embora a ferramenta ainda esteja em desenvolvimento, os resultados de comparação com a ferramenta de referência mostraram-se competitivos. Neste sentido, como trabalhos futuros pretende-se aperfeiçoar os métodos, assim como aumentar as funcionalidades do *framework*. O *framework* proposto inova em sua abordagem flexível. Desconhece-se outras metodologias de mapeamento tecnológico que permitam a flexibilização do seu fluxo.

5. REFERÊNCIAS BIBLIOGRÁFICAS

CORREIA, V.; REIS, A. (2004). Advanced technology mapping for standard-cell generators. **SYMPOSIUM ON INTEGRATED CIRCUITS AND SYSTEM DESIGN, SBCCI**.

MANOHARARAJAH, V.; "Heuristics for area minimization in LUT-based FPGA technology map-ping," **IEEE: Transactions On Computer-Aided Design Of Integrated Circuits And Systems**.

ABC, BERKELEY LOGIC SYNTHESIS AND VERIFICATION GROUP; 2013. **Abc: A system for sequential synthesis**. Disponível em <http://www.eecs.berkeley.edu/~alanmi/abc/>.