

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

ESTEVAN VEDOVELLI

**Otimização de Interconexões Através de
Posicionamento e Síntese Lógica**

Trabalho de Diplomação.

Prof. Dr. André Inácio Reis
Orientador

Porto Alegre, dezembro de 2009.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquiria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do ECP: Prof. Gilson Inácio Wirth

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	5
LISTA DE FIGURAS.....	6
LISTA DE TABELAS	7
RESUMO.....	8
ABSTRACT	9
1 INTRODUÇÃO	10
1.1 CONTEXTO	10
1.2 MOTIVAÇÃO.....	13
1.3 OBJETIVOS	13
1.4 ORGANIZAÇÃO DO TRABALHO	15
2 EMBASAMENTO E TRABALHOS ANTERIORES.....	16
2.1 POSICIONADORES EXISTENTES	16
2.1.1 Dragon	16
2.1.2 Capo.....	17
2.1.3 Kraftwerk.....	18
2.2 CAMINHOS NÃO-MONÓTONOS.....	18
2.2.1 Identificação de Caminhos Não-monótonos.....	18
2.2.1.1 Desvio.....	18
2.2.1.2 Direção do Sinal	19
2.3 AIG	20
2.4 TÉCNICAS DE OTIMIZAÇÃO DE ATRASOS	21
2.4.1 Inserção de Buffers	21
2.4.2 Dimensionamento das células.....	21
2.4.3 Realocação	22
2.4.4 Transformações de Sinais	22
2.4.5 Otimização de Caminhos Não monótonos	23
2.5 REPRESENTAÇÕES DE LIMITES E OBJETIVOS DE PROJETO	23
2.5.1 Synopsis® Design Constraints Format.....	23
3 DESCRIÇÃO DO PROJETO	24
3.1 ENTRADAS	25
3.1.1 Descrição Lógica do Circuito	25
3.1.2 Descrição dos Limites e Objetivos de Projeto.....	25
3.1.2.1 Limites Físicos do Circuito	25
3.1.3 Biblioteca de Células	26
3.2 SAÍDAS.....	26
4 POSICIONAMENTO DOS NODOS DO AIG	27
4.1 POSICIONAMENTO INICIAL.....	28
4.2 RECOZIMENTO SIMULADO	28
4.3 DETERMINAÇÃO DA MONOTONICIDADE.....	30

4.4 ANÁLISE ESTÁTICA DE ATRASOS	31
4.5 ANÁLISE DE ÁREA	31
5 EXPERIMENTOS.....	32
5.1 VALIDAÇÃO DOS RESULTADOS.....	32
5.2 INFLUÊNCIA DOS PARÂMETROS DO RECOZIMENTO SIMULADO	34
5.2.1 Influência da Temperatura Inicial de Recozimento	34
5.2.2 Influência do Fator de Resfriamento do Recozimento	35
6 TRABALHOS FUTUROS.....	37
6.1 MAPEAMENTO TECNOLÓGICO	37
6.2 PÓS-POSICIONAMENTO	37
6.2.1 Alocação de Espaços em Branco	38
6.2.2 Redução de Atrasos	38
7 CONCLUSÕES	39
REFERÊNCIAS	41
ANEXO <ESPECIFICAÇÕES DO FORMATO SDC>	43

LISTA DE ABREVIATURAS E SIGLAS

AIG	AND-INV Graph
ALU	Arithmetic Logic Unit
ASIC	Application-specific Integrated Circuit
BAF	Binary AIG Format
BDD	Binary-decision Diagram
CAD	Computer-Aided Design
EDA	Electronic Design Automation
FPGA	Field-Programmable Gate Array
FRAIG	Functionally Reduced AND-INV Graph
ISPD	International Symposium on Physical Design
NMF	Nom-monotone Factor
RAM	Random-Access Memory
STA	Statistic Timing Analysis
SDC	Synopsis® Design Constraints
VLSI	Very-Large-Scale Integration

LISTA DE FIGURAS

Figura 1.1: Os três domínios de projeto na representação em Y de Gajski (GAJSKI, et al., 1983).....	11
Figura 1.2: Fluxo de projeto <i>standard cells</i>	12
Figura 1.3: Novo fluxo de projeto proposto	14
Figura 1.4: Exemplos de caminhos com baixa (a) e alta (b) monotonicidades	15
Figura 2.1: Visão geral do fluxo de posicionamento do Dragon2005.....	17
Figura 2.2: Exemplo de caminho crítico não-monótono (BERAUDO, et al., 2003)	18
Figura 2.3: Detecção de caminhos não-monótonos através da noção de direção do sinal	19
Figura 2.4: Duas AIGs representando a mesma função lógica (MISHCHENKO, et al., 2004).....	20
Figura 2.5: Exemplo de inserção de <i>buffers</i>	21
Figura 2.6: Exemplos de reduções de atraso através de replicação de sinais (CHANG, et al., 2007).....	22
Figura 3.1: Esquema representando as entradas e saídas da ferramenta de posicionamento proposta pelo trabalho	24
Figura 4.1: Etapas globais da ferramenta de posicionamento	27
Figura 4.2: Fluxo de posicionamento do AIG	28
Figura 4.3: Fluxo do recozimento simulado.....	29
Figura 4.4: Exemplo de percurso em AIG para a determinação do fator de não-monotonicidade	30
Figura 5.1: AIG do circuito NAND8 posicionada idealmente	32
Figura 5.2: Resultado do posicionamento da NAND8 em diversas áreas.....	33
Figura 5.3: Influência da área disponível no NMF e no tempo de execução do posicionamento da NAND8.....	33
Figura 5.4: Resultados do posicionamento do circuito <i>cm85a</i> em limites físicos distintos	33
Figura 5.5: Evolução do custo das soluções durante o Recozimento Simulado.....	34

LISTA DE TABELAS

Tabela 4.1: Matriz de fatores de não-monotonicidade	31
Tabela 5.1: Resultados do posicionamento com temperatura $T=0,01$	34
Tabela 5.2: Impacto da variação da temperatura no posicionamento.....	35
Tabela 5.3: Resultados do posicionamento com fator de resfriamento $CF=0,1$	35
Tabela 5.4: Impacto da variação do fator de resfriamento no posicionamento.....	36

RESUMO

No fluxo atual de projeto de circuitos digitais modernos é difícil estimarmos os atrasos que ocorrem nas interconexões, especialmente antes do posicionamento das células. E quando os atrasos são corretamente avaliados, após o posicionamento, devido às diferentes estruturas de dados utilizadas para as diferentes etapas do projeto, eles não podem ser retornados à etapa de síntese lógica facilmente.

Neste trabalho, propomos modificações no fluxo de projeto de circuitos integrados que permitem estimar antecipadamente os atrasos de interconexões e que, com a utilização de uma estrutura de dados única para a otimização lógica, posicionamento e mapeamento, possibilitam o retorno de informações à etapa inicial de otimização lógica. Nosso fluxo mistura a síntese lógica com a síntese física. Após realizarmos a otimização lógica convencional usando AIGs para representarmos cada módulo do circuito, posicionamos cada AIG na área reservada pelo *floorplanning*. Com o posicionamento realizado, é possível estimarmos os atrasos das interconexões, e essas informações podem ser repassadas à otimização lógica que as considerará em seus algoritmos para melhorar a qualidade de seus resultados. Uma vez que o AIG otimizada estiver corretamente posicionada, a etapa de mapeamento tecnológico deve utilizar um algoritmo que priorize a proximidade geométrica na escolha dos nodos que serão agrupados e substituídos pelas células da biblioteca. Por fim, a etapa de pós-posicionamento regulariza a solução e a otimiza para que o roteamento seja realizado.

Implementamos uma ferramenta capaz de posicionar um AIG e analisar os atrasos estáticos das interconexão e células da solução. O posicionador identifica a melhor solução analisando a monotonicidade do circuito. Quanto mais monótono o circuito, menor é o atraso das interconexões. O AIG posicionada com as informações de atrasos poderá ser redirecionada para a síntese lógica ou então mapeada às células da biblioteca.

Palavras-Chave: projeto de circuitos integrados, CAD, posicionamento, síntese lógica

Interconnects Optimization through Placement and Logic Synthesis

ABSTRACT

In current modern digital circuits' design flow, it is hard to estimate interconnection delays, especially before cell's placement. And once placement and STA are done, timing information cannot be easily feedbacked to logic optimization tools due the differences between design processes' structures.

In this work, we propose modifications in the current integrated circuits' design flow which allows the early delay estimation. With the utilization of a common structure from logical optimization to placement and technological mapping, we make it possible to feedback information for logical optimization module. Our flow mixes both logical and physical synthesis. Once conventional logical optimization is done using AIGs as circuit's representation structure, we place the AIGs in the floorplanning reserved area. After placement it is possible to estimate interconnection delays and feedback it to the logical optimization tool. Based on the new information, logical optimization results could be optimized for interconnection delays. Technological mapping process will start when STA and AIG placement steps are validated. The algorithm used by the mapping tool should consider geometric proximity as function cost when choosing AIG nodes to be replaced by library's cells. Finally, a post-placement step validates the solution and optimizes it for routing.

We implemented a tool for placing an AIG and performing STA due to its interconnections and cells. The placement tool identifies the best solutions through circuit's monotonicity analysis. As the circuit's monotonicity increases, the interconnection's delay decreases. The final AIG placement's solution keeps timing and position information and can be feedbacked to logical optimization step or redirected for being mapped in the library's cells.

Keywords: integrated circuits' design, CAD, placement, logical synthesis

1 INTRODUÇÃO

1.1 Contexto

Com a popularização dos computadores após a sua miniaturização, possível graças à invenção dos circuitos integrados por Jack Kilby e Robert Noyce em 1958, novos campos da ciência e engenharia se desenvolveram fortemente, como a criação de algoritmos e aplicações, os processos de fabricação e criação de computadores, e de circuitos digitais.

Uma das áreas de atuação da engenharia é a criação de softwares voltados para o projeto de produtos que são conhecidos como programas de CAD. Na área da microeletrônica, os programas de CAD são também conhecidos como EDA, e são usados durante todas as etapas da concepção e fabricação de circuitos integrados.

Projetos de concepção de circuitos integrados VLSI *full custom* podem ser para circuitos de propósito geral (microprocessadores, memórias, etc.) que são compartilhados por várias aplicações diferentes, ou também para circuitos ASICs (Circuitos Integrados de Aplicação Específica) que são destinados a uma pequena faixa de aplicações. Ambos são produzidos buscando-se a otimização de parâmetros que nem sempre podem ser otimizados simultaneamente. Entre esses parâmetros temos a área ocupada pelo circuito, o tempo necessário para o circuito fornecer as respostas e o consumo de energia do circuito (daqui para frente referidos respectivamente como área, atraso e potência), e também parâmetros indiretos como o tempo de concepção e a testabilidade (GEREZ, 1998).

Outros circuitos integrados muito utilizados são os FPGAs. Eles podem ser usados para implementar qualquer função que um circuito ASIC possa implementar. Além disso, eles podem ser programados para empenhar uma funcionalidade específica e reprogramados mesmo depois de embarcados ao produto, o que possibilita que o mesmo circuito seja amplamente utilizado, reduzindo os custos de projeto. No entanto, por não serem projetados para aplicações específicas, os FPGA não oferecem o mesmo desempenho comparado aos circuitos dedicados.

Segundo (GAJSKI, et al., 1983), o processo de criação de circuitos VLSI *full custom* pode ser dividido em três domínios: domínio comportamental; domínio temporal; e domínio físico. O domínio comportamental descreve a funcionalidade e o comportamento temporal a qual servirá o circuito. Um sistema completo é descrito a partir de uma série de algoritmos, estes podem ser vistos como um conjunto de elementos de memória e funções que computarão o seu próximo estado. No domínio estrutural descrevemos os elementos do circuito a partir de sub-circuitos. Um processador é um conjunto de unidades funcionais (ALUs, RAMs, etc.). Estes são formados a partir de unidades lógicas (*gates*, *flip-flops*, etc.) que são compostos de

transistores. O domínio físico descreve como as sub-partes do domínio estrutural são dispostas fisicamente no chip que é organizado essencialmente em duas dimensões. Os níveis de abstração da parte física vão desde o leiaute de transistores até as partições físicas que representam as unidades funcionais do domínio estrutural. A Figura 1.1 mostra os três domínios e suas hierarquias representados em cada eixo da representação em Y.

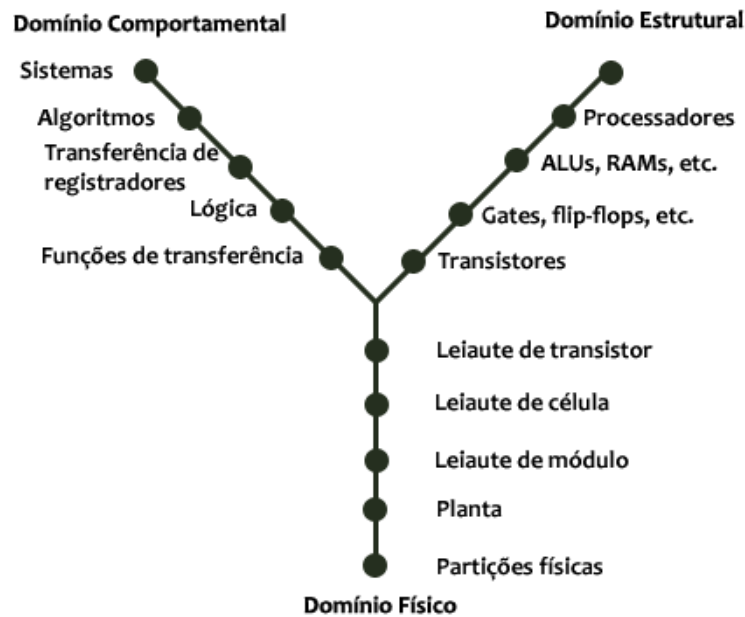


Figura 1.1: Os três domínios de projeto na representação em Y de Gajski (GAJSKI, et al., 1983)

Diversas metodologias de projeto de circuitos digitais podem ser herdadas a partir do diagrama em Y de Gajski, abstraindo ou não algumas etapas. Entre elas, uma das mais importantes e utilizadas atualmente é a metodologia de *standard cells*.

O fluxo de *standard cells* parte da descrição dos objetivos de projeto, que incluem os atrasos esperados entre módulos do circuito e a descrição física da área, da biblioteca de células com os leiautes das células a serem utilizadas, e do respectivo conjunto de regras da tecnologia utilizada na fabricação do circuito. A partir das restrições de atraso e da biblioteca, a etapa de síntese lógica é responsável por traduzir a descrição lógica do circuito na descrição do circuito a ser posicionado, através de otimizações e mapeamento tecnológico. O passo seguinte é a organização dos diversos módulos do circuito na área (*floor planning*) e o planejamento das linhas de alimentação do circuito. Em seguida, com as áreas definidas para cada módulo, as células são posicionadas em cada bloco (*placement*). O planejamento dos *clocks* que controlam os registradores é feito, e finalmente o leiaute das interconexões que ligam as células em cada módulo são definidos pelo roteador (*routing*). O circuito projetado passa por uma série de ferramentas que verificam se ele é elegível para a fabricação. Caso haja algum problema, o projeto deve ser retomado a partir de um dos passos anteriores (RABAEY,

et al., 2003). A Figura 1.2 ilustra o fluxo *standard cell* de projeto de circuitos integrados.

Na etapa de síntese lógica, estruturas de dados diferentes podem ser usadas para a otimização do circuito. O AIG é uma estrutura de fácil manipulação e que, apesar de não ser uma representação canônica do circuito, pode ser reduzida funcionalmente e então utilizada para manipulações lógicas. Ela é um grafo direcionado composto por três tipos de nodos que representam as entradas, as saídas e a função “E de duas entradas” (AND2). Além disso, seus arcos possuem dois valores: invertidos ou não-invertidos, e representam respectivamente uma interconexão com e sem inversor (NOT).

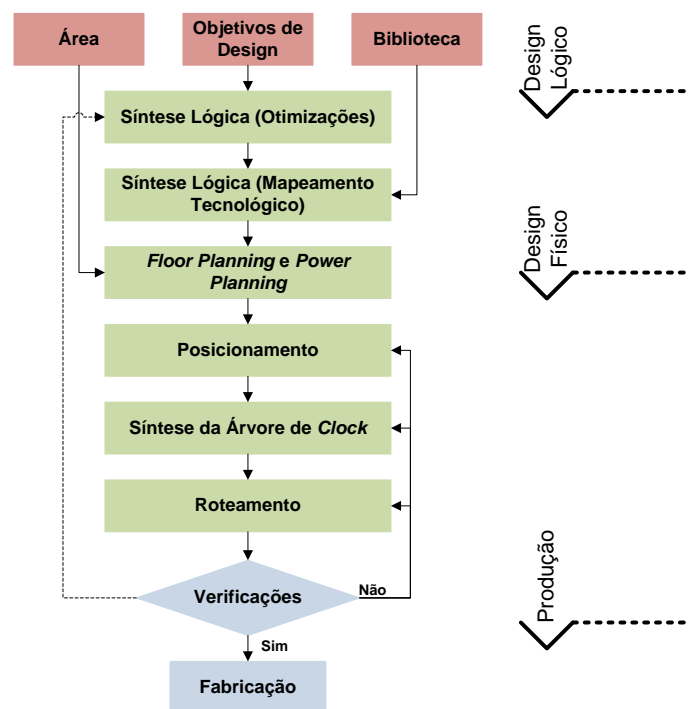


Figura 1.2: Fluxo de projeto *standard cells*

Ainda durante a síntese lógica, temos a fase de mapeamento tecnológico. Um bloco do circuito, que realiza certa funcionalidade, é mapeado para uma célula da biblioteca com a mesma funcionalidade, e assim segue até todo circuito ser representado por células. O mapeamento pode ser feito, por exemplo, de forma a se reduzir a área ocupada pelo circuito.

Ao posicionamento das células no plano damos o nome de *placement*. Nessa etapa o leiaute das células é conhecido, e também é conhecida a descrição de como essas células devem ser interconectadas. O objetivo do posicionamento é atribuir uma localização no chip a cada célula de forma que as interconexões possam ser realizadas. Um bom algoritmo irá minimizar a área e o comprimento das interconexões, reduzindo o atraso devido às mesmas.

Na etapa de roteamento, é criado o leiaute final das interconexões entre as células. O objetivo é definir o local dos *steiner points* e os caminhos pelos quais as interconexões

devem seguir para que sejam obtidos os menores atrasos, considerando-se o compromisso entre o comprimento e os parasitas das interconexões.

1.2 Motivação

No circuito combinacional, otimização lógica é uma ferramenta muito usada na redução do atraso para atingir as metas estabelecidas na especificação dos circuitos integrados. No entanto, com a redução das dimensões dos elementos dos circuitos integrados, as interconexões entre células mostram-se responsáveis por grande fração do atraso em circuitos (THEIS, 2000). O atraso das interconexões não é reduzido junto com suas dimensões, pois o efeito de sua capacitância permanece o mesmo com as reduções. Assim os atrasos nos *gates* perdem importância comparados as interconexões para tecnologias atuais (FLYNN, et al., 2009). Estudos mostram que as interconexões podem contribuir com 50% a 80% do atraso total do circuito para tecnologias de 0.25 μ m ou menores (KEUTZER, et al., 1997). A otimização lógica, que considera apenas os atrasos dos *gates*, está sendo substituída pela otimização de interconexões.

Etapas de pós-posicionamento (*post-placement*) vêm sendo empregadas com sucesso ao desenvolvimento de circuitos integrados combinadas à síntese lógica, reduzindo o atraso oriundo das interconexões. No entanto, mesmo que os algoritmos existentes de pós-posicionamento tenham permitido reduções no atraso de circuitos, a maioria deles está limitada a pequenas transformações de sinais e não é capaz de otimizar más decisões tomadas pelos algoritmos da etapa de síntese lógica (PLAZA, et al., 2008).

Algumas pesquisas sugerem a estimativa da etapa de posicionamento das células durante o processo de síntese lógica. Porém transformações lógicas nesse sentido estão limitadas pela baixa precisão das técnicas de estimativa do posicionamento existentes atualmente. Em alguns casos, técnicas desse gênero, além de não reduzirem o atraso, aumentam a potência e a área do circuito (GOSTI, et al., 1998).

A crescente influência dos atrasos das interconexões nos atrasos totais dos circuitos e escassez de métodos eficientes para o posicionamento de células faz com que novos algoritmos de projeto de circuitos integrados sejam estudados e desenvolvidos (FLYNN, et al., 2009).

1.3 Objetivos

Este trabalho propõe uma técnica diferente para o posicionamento das células de um circuito que une as etapas de síntese lógica e posicionamento. Os objetivos consistem em definir um fluxo de projeto que permita otimizar as interconexões através de síntese lógica. Para isso é necessário se ter informações precisas sobre o posicionamento, para que o efeito das interconexões possa ser corretamente avaliado, e estabelecer uma ligação entre estas informações e a representação lógica do circuito. Também buscamos realizar o posicionamento utilizando uma função custo que, quando minimizada, reduza ao máximo o atraso das interconexões.

A partir do circuito logicamente otimizado e o *floorplan* definido, o AIG que representa o circuito é posicionado na área disponível antes de ser realizado o mapeamento tecnológico. Dessa forma é possível estimarmos quais são os caminhos

críticos do circuito e retomarmos a etapa de síntese lógica com o intuito de reduzirmos os atrasos desses caminhos, ou então a área ocupada pelo mesmo, se necessário.

Após o posicionamento, os nodos do AIG são mapeados às células da biblioteca. Os cortes no AIG para o mapeamento tecnológico devem ser feitos baseados na proximidade geométrica dos nodos preservando-se assim o posicionamento do AIG.

Finalmente a etapa de pós-posicionamento verifica, legaliza de acordo com as regras da tecnologia e aperfeiçoa as posições das células obtidas ao fim do mapeamento.

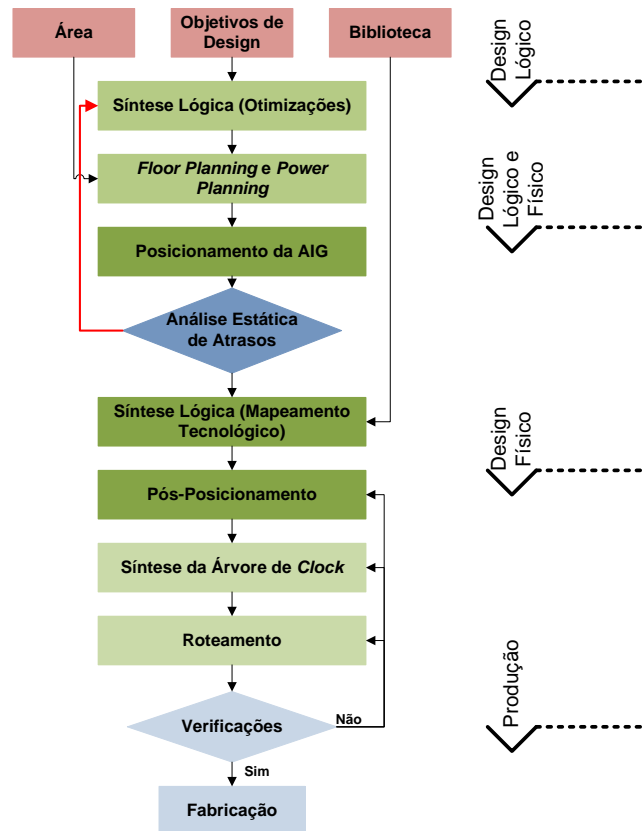


Figura 1.3: Novo fluxo de projeto proposto

O novo fluxo de projeto, modificado a partir do fluxo *standard cell* (Figura 1.2), é apresentado na Figura 1.3. O posicionamento do AIG permite atingir uma boa estimativa da solução final e da influência das interconexões na mesma, mantendo uma ligação direta com a representação lógica do circuito. Com essas informações é possível retomar a síntese lógica e propor melhorias que serão refletidas na solução final.

Durante a etapa de posicionamento do AIG, o posicionador escolhe a posição de cada célula do circuito de forma a obter interconexões monótonas. Para isso foi estabelecida uma função que calcula a medida da monotonicidade do circuito. De acordo com a Figura 1.4, o caminho (a) possui o fator de monotonicidade menor que o caminho (b), pois o caminho (b) segue aproximadamente sempre na mesma direção, enquanto o caminho (a) altera sua direção diversas vezes até chegar ao destino. Os caminhos considerados não-monótonos são os caminhos cujo atraso pode ser reduzido através do reposicionamento das células (BERAUDO, et al., 2003).

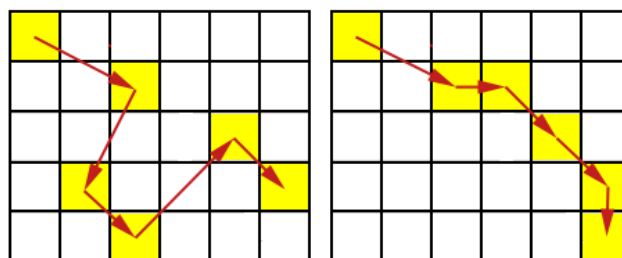


Figura 1.4: Exemplos de caminhos com baixa (a) e alta (b) monotonicidades

Para validarmos a viabilidade do fluxo de projeto proposto, implementamos uma ferramenta automatizada que é capaz de posicionar um dado AIG na área indicada pelo usuário. O posicionamento é feito minimizando-se os caminhos não-monótonos. Também é possível realizar a análise estática de atrasos da solução e a compará-la com os objetivos do projeto. Em caso de não validação do resultado, o AIG é reenviado à etapa de otimização lógica com as informações relativas às modificações necessárias.

1.4 Organização do Trabalho

Na sessão 2, apresentaremos estudos e trabalhos realizados previamente que são fundamentais ou relevantes para o entendimento e embasamento deste trabalho. A sessão 3 descreve o projeto realizado e o fluxo de projeto proposto. Na sessão 4, cobrimos a implementação da ferramenta posicionadora de AIG. Na sessão 5, são apresentados as validações e os resultados dos testes da ferramenta implementada. A sessão 6 explora as próximas etapas ainda não implementadas do fluxo de projeto. Os pontos importantes são resumidos e lembrados na sessão 7.

2 EMBASAMENTO E TRABALHOS ANTERIORES

Nas sessões seguintes apresentaremos trabalhos relevantes para a compreensão do trabalho desenvolvido.

2.1 Posicionadores Existentes

Os algoritmos de posicionamento são geralmente classificados de acordo o processo utilizado para determinar a solução final. Existem posicionadores que utilizam processos estocásticos, baseados em particionamento, analíticos não-lineares, ou analíticos quadráticos.

Métodos estocásticos resolvem o problema de posicionamento procurando pela melhor solução a partir de movimentos aleatórios. Normalmente empregam técnicas de recozimento simulado (*simulated annealing*) que, em analogia com a termodinâmica, reduzem a *temperatura* gradualmente tornando-se cada vez mais restritivos para aceitarem os movimentos das células. Os posicionadores baseados em particionamento cortam o circuito recursivamente, buscando minimizar uma função de custo, que normalmente é a quantidade de conexões que cruzam os cortes no grafo formado pelas células (*min-cut*). Os métodos analíticos determinam uma função de custo e a minimizam através de análise matemática. Nos métodos analíticos não-lineares a função é não-linear e técnicas de otimização matemáticas de funções não lineares são usadas. Nos métodos analíticos quadráticos a função é quadrática e pode ser resolvida por um sistema de equações lineares (NAM, et al., 2007).

Três ferramentas de posicionamento acadêmicas que utilizam um ou mais desses processos serão apresentadas: Dragon, Capo e Kraftwerk. Todas elas realizam posicionamentos de alto desempenho comparáveis com as melhores ferramentas comerciais disponíveis (i.e. ITools®, Cadence QPlace®, etc.)

2.1.1 Dragon

A ferramenta de posicionamento Dragon2000 realiza o posicionamento de células padrão (*standard-cells*) em duas etapas: posicionamento global e posicionamento detalhado. Durante o posicionamento global, o posicionamento das células é realizado através de particionamentos recursivos. Cada área quadrilátera é dividida em quatro novas áreas e as células são redistribuídas igualmente entre elas minimizando-se a quantidade de fios que interligam as diferentes áreas e o comprimento estimado de fio total. O posicionamento global termina assim que restarem apenas algumas células por área. A partir do resultado do posicionamento global, o posicionamento detalhado separa células que se sobrepõem e realiza melhorias locais (WANG, et al., 2000).

Um ponto fraco do algoritmo de *min-cut* é a irreversibilidade, que impede a troca de área de uma célula após ela ter sido atribuída. Para evitar que o algoritmo caia em mínimos locais afastados do mínimo global, técnicas de *Simulated Annealing* são aplicadas às áreas resultantes de cada etapa do *min-cut* (SARRAFZADEH, et al., 2002).

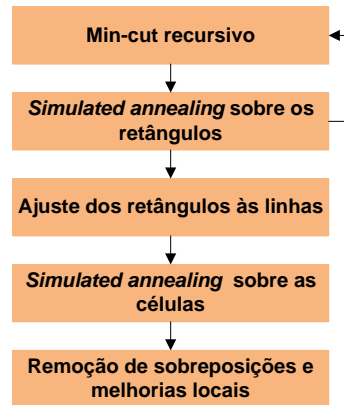


Figura 2.1: Visão geral do fluxo de posicionamento do Dragon2005

O Dragon2005 foi desenvolvido com base no Dragon2000 introduzindo a possibilidade de posicionar células macro. Como as células macros podem ser maiores que as áreas durante a fase de *min-cut*, o Dragon2005 as redimensiona de forma que as células macro possam ser posicionadas sem sobreposição (TAGHAVI, et al., 2005). Uma visão geral do fluxo de posicionamento do Dragon2005 é apresentada na Figura 2.1.

2.1.2 Capô

Como no Dragon, o posicionador Capô realiza o posicionamento das células hierarquicamente, dividindo a área do circuito em duas novas áreas recursivamente até que poucas células restem em cada área. O particionamento é realizado aplicando-se o algoritmo de *min-cut* no hipergrafo formado pelas interconexões do circuito. Conforme é reduzida a quantidade de células em cada área, o posicionamento das células se aproxima de uma solução otimizada (ROY, et al., 2005).

Espaços em branco são espaços localizados na área destinada ao posicionamento que não contém células. Espaços em branco bem distribuídos pela área facilitam o roteamento do circuito pois reduzem congestionamentos. No entanto, devido à tendência das células a se afastarem nesse caso, o comprimento das interconexões aumenta.

Durante o algoritmo de *min-cut*, o Capô pode distribuir os espaços em branco de três maneiras diferentes: uniformemente entre as duas novas partições; minimizando localmente os espaços em branco disponíveis e; distribuindo desigualmente os espaços em branco em retângulos com grande quantidade de espaços em branco. A distribuição uniforme possibilita a redução do congestionamento, porém o aumento do comprimento total das interconexões. Ao contrário, a distribuição desigual diminui o comprimento, mas pode acarretar no congestionamento das interconexões. A minimização local dos espaços em branco possibilita certa redução dos comprimentos de interconexão sem congestionamentos globais (NAM, et al., 2007).

2.1.3 Kraftwerk

A ferramenta Kraftwerk utiliza um método analítico quadrático baseado em forças-direcionadas para posicionar as células. O algoritmo minimiza uma função custo através da resolução de um sistema de equações. A função custo utilizada é dada pela soma do quadrado das distâncias Euclidianas de todas as interconexões (NAM, et al., 2007).

Para se determinar o sistema de equações, modela-se o circuito como um sistema mecânico. Em analogia com a mecânica, os fios que interligam as células agem como molas interconectadas. O posicionamento com o menor custo corresponde ao equilíbrio mecânico do sistema. Campos de força de repulsão e atração são então adicionados respectivamente às áreas de alta e baixa densidade de células para melhor distribuí-las na área disponível (OBERMEIER, et al., 2005).

No resultado gerado pelo Kraftwerk, as células podem estar sobrepostas umas as outras. A ferramenta Domino (DOLL, et al., 1994) é utilizada para remover as sobreposições e otimizar o posicionamento.

2.2 Caminhos Não-monótonos

Caminhos não-monótonos são caminhos percorridos pelos sinais, entre as células iniciais e finais, que contenham ziguezagues (Figura 2.2). Diversas publicações mostram que a eliminação da não-monotonicidade reduz consideravelmente o atraso no circuito, principalmente por reduzir os comprimentos de interconexões (PLAZA, et al., 2008), (GOSTI, et al., 1998), (HWANG, et al., 2006) e (BERAUDO, et al., 2003).

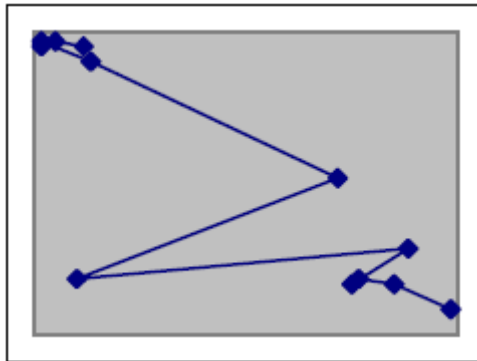


Figura 2.2: Exemplo de caminho crítico não-monótono (BERAUDO, et al., 2003)

2.2.1 Identificação de Caminhos Não-monótonos

2.2.1.1 Desvio

A noção de desvio é introduzida e indica o quanto certo nodo está deslocado da região monótona. Definimos o desvio como:

$$desvio(i) = dist(prev(i), i) + dist(i, next(i)) - dist(prev(i), next(i))$$

onde $dist$ é a distância retilínea entre seus argumentos e $prev(i)$ e $next(i)$ são respectivamente a célula que antecede e a célula que sucede a célula i . O desvio é calculado para cada célula dentro do caminho crítico (BERAUDO, et al., 2003). O caminho é considerado não-monótono caso, para uma célula ou um grupo de células pertencentes a este, o desvio supere um limite estabelecido.

Uma generalização da noção de desvio é alcançada quando definimos o fator de não-monotonicidade (NMF):

$$NMF = \frac{1}{c_{ideal}(x_1, x_k)} \sum_{n=1}^{k-1} c(x_n, x_{n+1})$$

onde temos a função c que define o custo entre as células x_n e x_{n+1} pertencentes ao caminho $\{x_1, \dots, x_k\}$ e pode representar, por exemplo, a distância retilínea entre duas células, como no caso citado acima, ou ainda, o atraso do sinal entre as células. c_{ideal} é o resultado ideal da função c para um caminho não-monótono. Caso trabalharmos com distâncias, c_{ideal} é a distância de Manhattan. Na solução baseada em atrasos temos $c(a, b)$ igual ao tempo de chegada do sinal em b subtraído do tempo de chegada do sinal em a , e definimos $c_{ideal}(a, b)$ como o atraso ideal do caminho entre a e b corretamente *bufferizado*. Uma das vantagens da utilização do NMF é a possibilidade de quantificar a monotonicidade. Para um caminho, quanto mais o seu NMF estiver próximo de um, mais ele será monótono. (PLAZA, et al., 2008).

2.2.1.2 Direção do Sinal

Outra forma de identificar caminhos não-monótonos utiliza a noção de direção do sinal. Para percorrer um caminho a partir de uma entrada I até uma saída O , o sinal deve se propagar na direção \overrightarrow{IO} . Cada célula por onde o sinal passa entre I e O deve possuir coordenadas que sigam a ordem monótona ditada por I e O (HWANG, et al., 2006). Um problema desse método é a dificuldade de quantificar a monotonicidade.

A Figura 2.3 apresenta um exemplo de caminho não-monótono entre as células PI e PO . Podemos verificar a não-monotonicidade das coordenadas de certas células do caminho trançando retas verticais e horizontais e deslocando-as pelo circuito. Sempre que pelo menos uma das retas for cortada por dois ou mais caminhos entre dois nodos do caminho simultaneamente, este será considerado não-monótono.

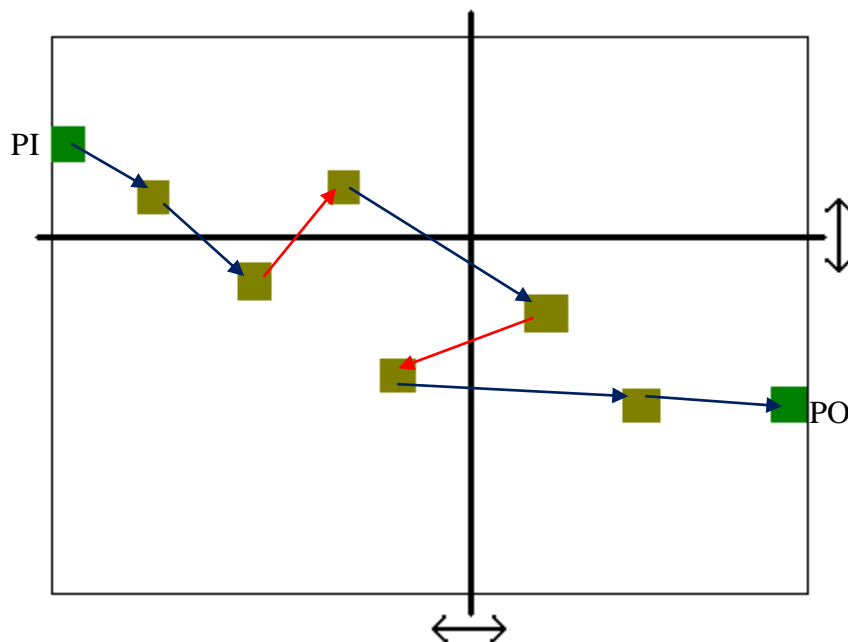


Figura 2.3: Detecção de caminhos não-monótonos através da noção de direção do sinal

2.3 AIG

O AIG é uma forma de representação estrutural da lógica funcional de um circuito. O interesse por essa estrutura surgiu na década de 50 (HELLERMAN, 1963). Nas décadas de 70 e 80 vários algoritmos de transformações de AIGs foram desenvolvidos e sua utilização foi impulsionada em sistemas de otimização e verificação lógica. Recentemente, o AIG tem sido alvo de muitas pesquisas por ser uma estrutura mais eficiente em termos de escalabilidade comparada a estruturas comumente utilizadas, como os BDDs. Neste projeto, os AIGs serão utilizadas como estruturas de representação do circuito.

Um AIG é um grafo direcionado e acíclico, onde existem três tipos de vértices: vértices com apenas um terminal que podem representar o valor “0” ou o valor “1”; vértices sem entrada de arcos representando as entradas primárias do circuito e; vértices com duas entradas representando a função AND (“E” lógico). Os arcos são direcionados de acordo com o fluxo dos sinais, e quando possuem atributos inversores (representaremos um arco inversor com um círculo preenchido sobre ele) implementam o complemento Booleano do sinal (KUEHLMANN, et al., 2002). Um AIG pode ser criada por fatorações de soma de produtos e conversões das funções AND e OR (respectivamente, “E” e “OU” lógicos) em AND2s e inversores usando-se as regras de DeMorgan (MISHCHENKO, et al., 2006).

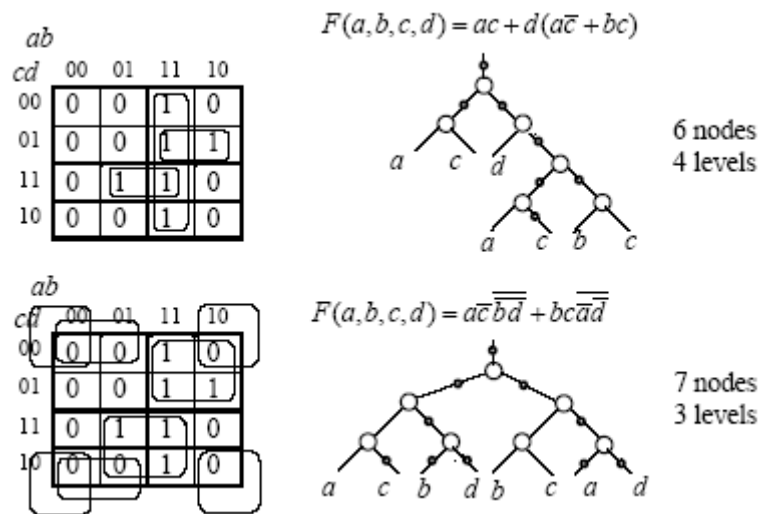


Figura 2.4: Duas AIGs representando a mesma função lógica (MISHCHENKO, et al., 2004)

O tamanho e o tempo de construção de AIGs são proporcionais ao tamanho do circuito representado, o que torna sua criação rápida e escalável e permite prever a quantidade de memória necessária para seu armazenamento (KUEHLMANN, et al., 2002). Um AIG sempre pode ser representado sem exceder três vezes o tamanho de um BDD (MISHCHENKO, et al., 2004).

O AIG não é uma representação canônica de um circuito. Na Figura 2.4, vemos dois AIGs diferentes representando a mesma função lógica. Porém o AIG pode ser reduzido funcionalmente (FRAIG) respeitando as condições $f_{n_1}(x) \neq f_{n_2}(x)$ e $f_{n_1}(x) \neq \overline{f_{n_2}(x)}$

para dois nodos quaisquer n_1 e n_2 , como especificado por (MISHCHENKO, et al., 2004), e assim pode ser usada em aplicações de verificação formal.

O AIG aparece como uma estrutura promissora capaz de unificar a representação do circuito através do fluxo *standard cell*. Além de ser utilizada para otimização lógica e verificação formal, a estrutura também pode ser usada para mapeamento, simulação (MISHCHENKO, et al., 2004) e, como desenvolvido ao longo deste projeto, para posicionamento.

2.4 Técnicas de Otimização de Atrasos

Vários trabalhos propõem diferentes técnicas para a otimização dos atrasos dos circuitos em etapas de pós-posicionamento. Entre as diferentes técnicas destacam-se dimensionamento de *gate*, realocação, otimização de caminhos não-monótonos, inserção de *buffers*, transformações de sinais, entre outros. No geral as técnicas citadas têm como objetivo a redução do atraso de caminhos críticos e do congestionamento de interconexões.

2.4.1 Inserção de *Buffers*

Nesta técnica, células com um grande *fanout*, devido tanto às células conectadas a sua saída quanto às interconexões, serão selecionadas e *buffers* serão inseridos entre essas células e parte das células conectadas a ela. Desse modo reduz-se a capacitância de saída da célula sobrecarregada e, portanto, o atraso desta célula é também reduzido.

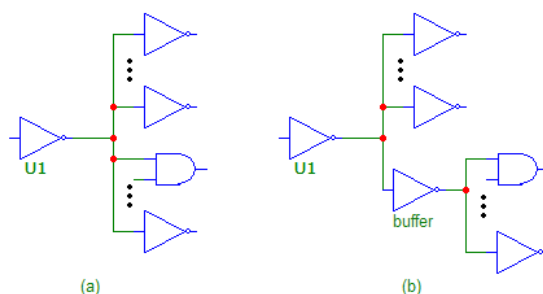


Figura 2.5: Exemplo de inserção de *buffers*

A Figura 2.5 mostra a introdução de um *buffer* na parte inferior do circuito da Figura 2.5(a) resultando no circuito da Figura 2.5(b). Com a redução do *fanout* da célula *U1*, o seu atraso de *gate* é reduzido. Diversos trabalhos desenvolvem métodos diferentes para encontrar-se a quantidade de *buffers* ideal e seus dimensionamentos. Entre eles podemos citar (KANNAN, et al., 1994) e (LI, et al., 2005). Porém a aplicação deste método resulta no acréscimo da área total do circuito.

2.4.2 Dimensionamento das células

O dimensionamento das células consiste na troca de células que se encontram no caminho crítico a ser otimizado por células com a mesma funcionalidade lógica, porém com diferentes características elétricas (ex: maior capacidade de corrente). A célula escolhida para a substituição, entre as células disponíveis na biblioteca, é aquela que

minimiza um determinado custo. A utilização dessa técnica foi empregada com sucesso por (KANNAN, et al., 1994) adicionada à técnica de inserção de *buffers* de interconexões (ver seção 2.4.1). Nos testes realizados, o atraso sofreu reduções de 13% a 22% para uma série de circuitos comerciais. Porém modificações desse gênero geram grandes perturbações no posicionamento das células. Outro trabalho que utiliza essas técnicas (LI, et al., 2005) propõe uma solução que impede grandes perturbações no posicionamento inicial, no entanto o redimensionamento de *gates* ainda resulta no aumento da área final do circuito, podendo gerar resultados que não respeitam as restrições de área.

2.4.3 Realocação

A partir de um circuito com as células já posicionadas e roteadas, o atraso das interconexões é estimado considerando-se as topologias exatas. Em seguida, as células e os *steiner points* pertencentes aos caminhos críticos são movidos de acordo com um algoritmo que encontra o posicionamento que minimiza o atraso dos caminhos críticos. A movimentação das células e *Steiner points* pode resultar na criação de novos caminhos críticos. Para evitar este problema, o aumento do custo total das interconexões não pode superar certo limite. Os resultados experimentais mostram melhorias de 9% a 14% e um pequeno aumento de 1% a 5% da área total ocupada pelo circuito (AJAMI, et al., 2001).

2.4.4 Transformações de Sinais

Diversos métodos de pós-posicionamento exploram transformações e replicações de sinal para reduzir o atraso nos circuitos.

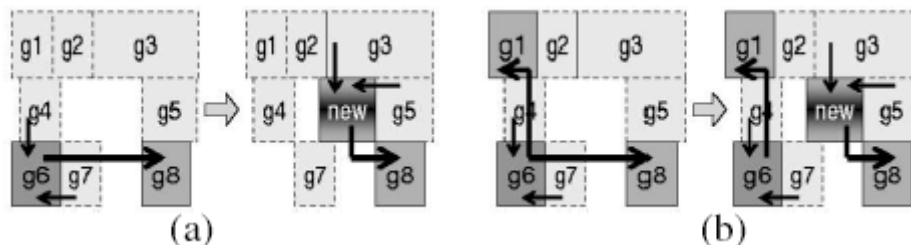


Figura 2.6: Exemplos de reduções de atraso através de replicação de sinais (CHANG, et al., 2007)

Em (CHANG, et al., 2007), os espaços não utilizados são aproveitados para a duplicação local de sinais, visando a redução de longas interconexões, e por consequência, o atraso e o congestionamento causado por estas. A Figura 2.6(a) mostra a inclusão de uma nova célula (*new*) que reproduz o sinal produzido por *g6* reduzindo os custos com interconexões na ligação com *g8*. Na Figura 2.6(b) a nova célula *new* gera o sinal para a célula *g8*, e a célula *g6* é mantida para fornecer o sinal a *g1* e reduzir os atrasos em ambos os caminhos. Neste trabalho, os resultados obtidos reduziram em média 11% os atrasos, com o acréscimo, em média, de 0,2% nos comprimentos dos fios.

Outro método proposto por (CHANG, et al., 2000) apresenta um algoritmo que detecta simetrias funcionais Booleanas em tempo linear e, caso encontre uma solução menos custosa, reconecta as estruturas sem alterações no posicionamento das células. A

verificação experimental realizada com esse método obteve reduções de atraso em média de 5,4% e reduções de área em média de 2,2%.

2.4.5 Otimização de Caminhos Não monótonos

Uma forma de identificar caminhos sujeitos a grandes melhorias é a partir da identificação de caminhos não-monótonos. O fluxo de pós-posicionamento elaborado por (PLAZA, et al., 2008) primeiro encontra os caminhos não-monótonos críticos a serem modificados a partir de uma medida da monotonicidade do caminho. Utilizando-se simulação funcional, transformações lógicas são efetuadas e os novos parâmetros físicos são reavaliados. A funcionalidade do circuito pode ser garantida por *simulation and satisfiability*. Os experimentos realizados resultaram na redução média do atraso em 11,7% com aumento médio da área do circuito menor de 2%.

2.5 Representações de Limites e Objetivos de Projeto

2.5.1 Synopsis® Design Constraints Format

O formato SDC é utilizado para especificar os objetivos de projeto referentes a atrasos, consumo, e área. É um formato aberto utilizado como padrão para a especificação de objetivos em projetos de circuitos integrados (SYNOPSYS®, 2007).

Ele define sintaxes para a especificação da versão do SDC utilizada, das unidades utilizadas no arquivo, dos objetivos de projeto, e dos objetos designados. A lista completa das possíveis descrições encontra-se em Anexo.

3 DESCRIÇÃO DO PROJETO

Durante a etapa de otimização lógica no fluxo de projeto *standard-cell*, com o fluxo atual, não é possível estimar precisamente os atrasos devido às interconexões, pois o posicionamento das células ainda não é conhecido. Após a etapa de posicionamento, os atrasos das interconexões podem ser estimados, porém devido a incompatibilidades entre as estruturas de dados, é difícil enviarmos informações de posicionamento de volta à etapa de otimização lógica. Sendo assim, é evidente a necessidade da utilização de estruturas de dados que permitam a comunicação entre as etapas do projeto e de novos passos que possibilitem a estimativa correta da influência dos atrasos das interconexões no resultado. A estrutura que utilizaremos será o AIG. Com essa única estrutura é possível realizarmos otimizações e manipulações lógicas, mapeamento, posicionamento e verificações funcionais.

Neste projeto descrevemos a criação de uma nova ferramenta de posicionamento de células de circuitos integrados. A partir do conjunto de entradas, os objetivos da ferramenta são estabelecer a posição de cada célula de forma a respeitar limites de localização espacial, minimizar atrasos de propagação dos sinais (respeitando as frequências alvas desejadas) e reduzir a potência do circuito.

O esquema da Figura 3.1 mostra as entradas e saídas da ferramenta de posicionamento.

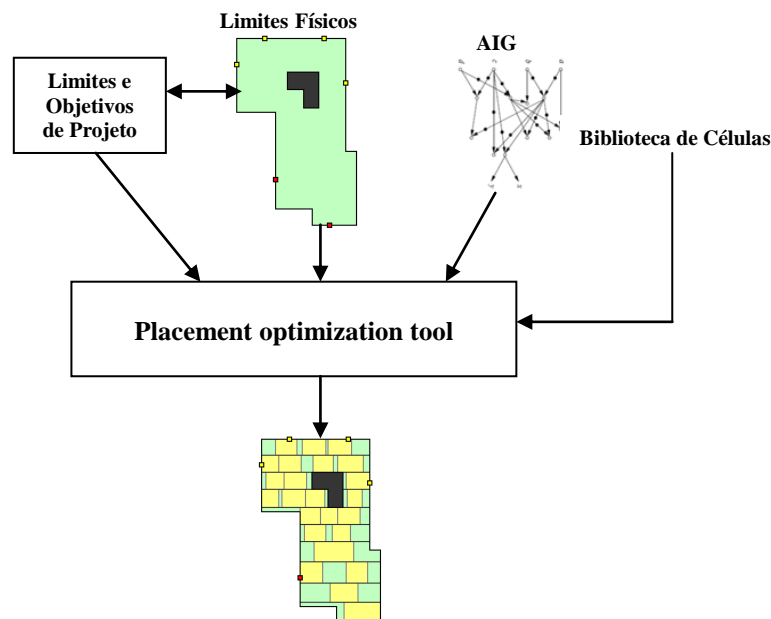


Figura 3.1: Esquema representando as entradas e saídas da ferramenta de posicionamento proposta pelo trabalho

3.1 Entradas

As entradas da ferramenta consistem nas informações necessárias para descrever o funcionamento lógico do circuito, assim como suas características físicas e temporais desejadas.

3.1.1 Descrição Lógica do Circuito

O posicionador necessita da descrição lógica do circuito. Ela deve ser indicada através de um AIG correspondente ao circuito e deve ser fornecida no formato BAF.

O AIG fornecida a partir da etapa de otimização lógica deve corresponder a um bloco de lógica combinacional situado entre blocos registradores. Cada bloco de lógica combinacional é enviado ao posicionador individualmente. A parte sequencial do circuito é posicionada separadamente.

3.1.2 Descrição dos Limites e Objetivos de Projeto

Os limites e objetivos do projeto são as informações que descrevem as condições necessárias para o funcionamento correto do circuito em seu ambiente. Entre as informações, podemos encontrar os atrasos máximos de sinais através de caminhos determinados, o consumo máximo do circuito, etc..

A ferramenta utiliza arquivos apenas com as indicações de atrasos máximos permitidos entre entradas e saídas, o que é suficiente para a realização dos testes de posicionamento de AIGs. Com a futura adição dos módulos de mapeamento e pós-posicionamento, o posicionador de AIGs pode passar a utilizar a descrição de limites e objetivos de projeto a partir de um arquivo SDC (Sessão 2.5.1).

3.1.2.1 Limites Físicos do Circuito

Entre os limites do projeto descritos pelo usuário, devem ser fornecidos os limites físicos que descrevem a área onde o posicionamento é permitido.

A descrição da área pode ser obtida após a etapa de *floorplanning*, que organiza e aloca áreas aos diferentes módulos do circuito total. Cada módulo, com sua área, deve ser enviado ao posicionador juntamente com o bloco correspondente de lógica combinacional.

A descrição é composta pelo conjunto de polígonos retos que, unidos, formam a área onde o posicionamento das células é permitido. Cada polígono é descrito através de um conjunto de pontos ordenados no sentido anti-horário. Os pontos são descritos por suas coordenadas x e y únicas e situadas no primeiro quadrante do plano cartesiano.

Todos os ângulos dos polígonos devem ser ângulos retos. O posicionador verifica se os limites físicos informados são válidos e, caso contrário, informa ao usuário a origem do erro.

3.1.3 Biblioteca de Células

Para o posicionamento completo, a ferramenta deverá receber a biblioteca que contém as células correspondentes a tecnologia do circuito a ser posicionado. Após o posicionamento, o AIG será mapeado para as células da biblioteca que serão posicionadas detalhadamente de acordo com a descrição dos limites e objetivos de projeto.

Na implementação atual da etapa do posicionamento do AIG, apenas as informações da biblioteca relevantes a essa etapa devem ser informadas: a altura e largura da célula AND2, e o atraso desta célula.

3.2 Saídas

Após a conclusão do posicionamento do AIG, caso o resultado seja validado, a ferramenta implementada proporciona à etapa de mapeamento o circuito representado em um AIG com a adição das informações de posicionamento na área. Caso o resultado não seja validado, o AIG, que além de conter as informações do posicionamento contém os atrasos estimados de cada interconexão, é reenviada a etapa de otimização lógica. Com as novas informações presentes, o AIG pode ser otimizada de forma a corrigir as causas do resultado não-otimizado. Para a validação do processo de iteração é necessário um estudo completo sobre a estabilidade e convergência da mesma.

Após as etapas de mapeamento e pós-posicionamento, a serem implementadas no futuro, as saídas da ferramenta posicionadora (Figura 3.1) consistem na lista de células utilizadas pelo posicionador e suas respectivas posições no plano assim como a descrição da conectividade das células umas com as outras. Essas informações serão usadas para a etapa do roteamento concluir o leiaute do circuito.

4 POSICIONAMENTO DOS NODOS DO AIG

Como representado na Figura 4.1, o posicionador determina as posições dos nodos do AIG na área disponível. Os nodos do AIG são então mapeados baseados em sua proximidade geográfica para as células da biblioteca, sem comprometer assim o posicionamento do AIG. Finalmente o posicionamento é otimizado e as células adquirem suas posições finais respeitando os limites impostos pelas regras de projeto.

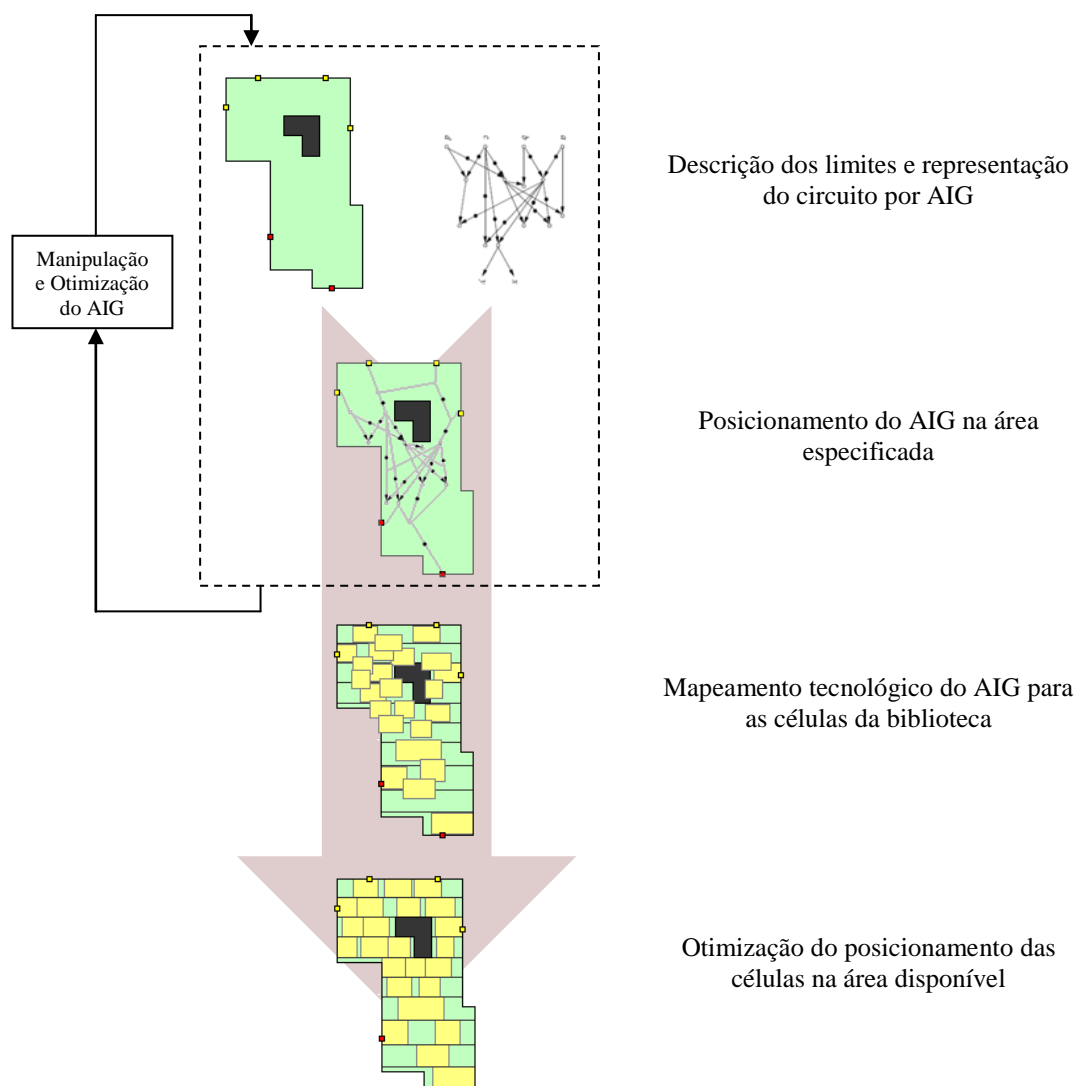


Figura 4.1: Etapas globais da ferramenta de posicionamento

A primeira etapa do posicionamento, localizada pelo retângulo tracejado na Figura 4.1, foi implementada como parte deste trabalho, e consiste em dispor cada nodo do AIG, que representa uma célula AND2, na área disponível descrita como parâmetro de entrada para a ferramenta. O fluxo da etapa de posicionamento do AIG, como implementado, é apresentado na Figura 4.2.

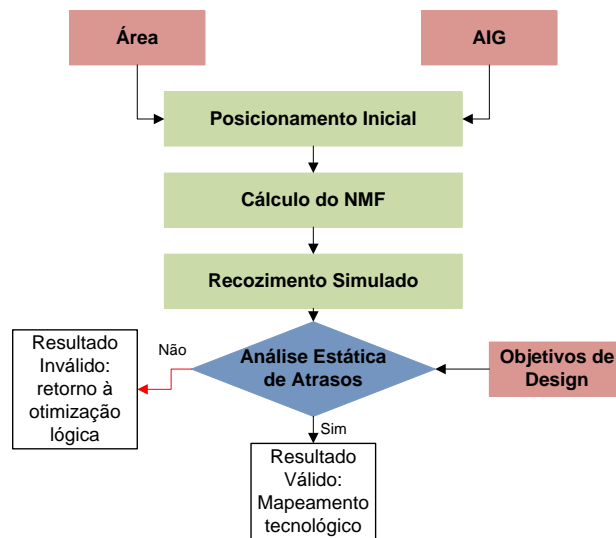


Figura 4.2: Fluxo de posicionamento do AIG

4.1 Posicionamento Inicial

Inicialmente a área é particionada em retângulos do mesmo tamanho e igual ao da célula AND2 da biblioteca. A quantidade de retângulos deve ser maior que a quantidade de nodos do grafo. Caso contrário a ferramenta indica a falta de espaço para o posicionamento e duas decisões podem ser tomadas: reduzir a quantidade de nodos no AIG através de manipulações lógicas; ou aumentar a área disponível para posicionamento.

Caso seja possível o posicionamento do AIG, cada nodo é então posicionado arbitrariamente em um dos retângulos disponíveis. A partir do posicionamento inicial o algoritmo de recozimento simulado é aplicado utilizando como função de custo a soma dos fatores de não-monotonicidade do circuito.

4.2 Recozimento Simulado

O algoritmo de posicionamento utilizado é o algoritmo de recozimento simulado (*simulated annealing*). Ele é um algoritmo lento comparado a outros, como o *Min-Cut*. Porém foi escolhido devido a sua simplicidade à implementação, possibilitando a rápida criação de um protótipo da ferramenta, e a qualidade do resultado obtido, que apesar do longo tempo de processamento, geralmente se aproxima mais do mínimo global comparado ao algoritmo de *Min-Cut*.

A cada passo do algoritmo, dois tipos de movimentos são permitidos entre as células: uma célula se move para um espaço desocupado, ou uma célula troca de lugar com outra célula. Os movimentos ocorrem aleatoriamente.

Os parâmetros que caracterizam o algoritmo de recozimento são a temperatura, o fator de resfriamento e a condição de parada. Após cada movimento a diferença Δc entre o novo custo e o custo anterior é calculada, caso Δc seja positivo o movimento é sempre aceito. Se o novo custo for mais alto que o antigo, a solução será aceita com uma probabilidade dada por:

$$P = e^{\frac{-\Delta c}{T}}$$

onde e é a constante de Euler e T é a temperatura do algoritmo. A temperatura influencia diretamente na probabilidade de aceitação do movimento. Quanto mais alta, maiores as chances de aceitação. Em cada etapa do algoritmo, n movimentos são realizados, onde n corresponde à quantidade de espaços disponíveis para o posicionamento das células. Após n movimentos, o algoritmo atinge o equilíbrio térmico para a temperatura atual. Ao início de uma nova etapa, a temperatura T é multiplicada pelo fator de resfriamento CF , que deve ser maior que 0 e menor que 1. Conforme T se aproxima de 0, apenas os movimentos que reduzem o custo são aceitos e a solução converge para uma solução de custo mínimo local. O algoritmo pára quando a condição de parada é atingida. O valor da condição de parada corresponde à porcentagem máxima de movimentos que devem ser aceitos durante uma etapa do recozimento. Com uma temperatura inicial elevada e $CF \rightarrow 1$, o tempo de execução se aproxima do infinito e a solução obtida é a solução ótima.

Em nossa ferramenta, os parâmetros do recozimento simulado são todos configuráveis. A função de custo utilizada mede a não-monotonicidade do circuito. Se uma modificação torna o circuito mais monótono o Fator de Não-Monotonicidade reduz e a solução é aceita.

É também possível executar o algoritmo com a geração do traço de custos habilitada. Dessa forma a cada etapa o custo será amostrado e adicionado a um arquivo que pode ser consultado posteriormente com intuito de avaliar os parâmetros escolhidos para o algoritmo de recozimento.

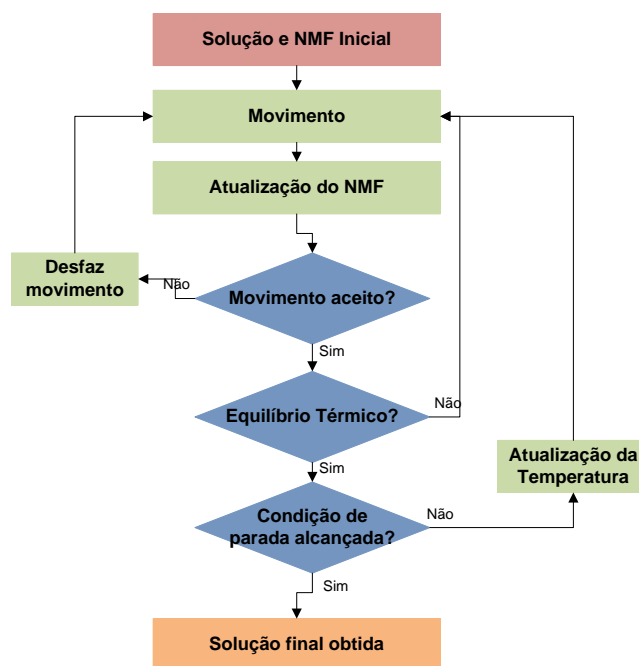


Figura 4.3: Fluxo do recozimento simulado

O fluxo do algoritmo recozimento simulado implementado pode ser visto na Figura 4.3.

4.3 Determinação da Monotonicidade

O fator de não-monotonicidade (apresentado na seção 2.2.1.1) entre dois nodos é calculado através da seguinte fórmula:

$$NMF = \frac{1}{c_{ideal}(x_1, x_k)} \sum_{n=1}^{k-1} c(x_n, x_{n+1})$$

onde a função c representa a distância euclidiana entre as células x_n e x_{n+1} e c_{ideal} é a distância euclidiana entre x_1 e x_k . Quando houver mais de um caminho para o cálculo da não-monotonicidade entre duas células, o NMF será o maior dos valores obtidos.

Para determinar-se o custo total da solução somam-se os fatores de não-monotonicidade entre cada dois nodos do AIG que estejam separados por no máximo uma quantidade de nodos (profundidade K) especificada pelo usuário. O custo é a média dos fatores calculados. Ele é um valor maior ou igual a um e quanto mais próximo de um, melhor será a qualidade do posicionamento obtido.

Na Figura 4.4, todos os caminhos em que o fator de não-monotonicidade será calculado a partir do nodo “a” para a profundidade “K=3” estão representados em tracejado. Entre os nodos “a” e “g”, dois caminhos são possíveis, o NMF é o maior valor calculado. Na Tabela 4.1, os NMF a serem calculados para o grafo da Figura 4.4 estão marcados com um “#”. O “K” indica que o fator não deve ser calculado, pois ultrapassa o limite de profundidade escolhido, e o “X” indica que não há um caminho de um nodo ao outro.

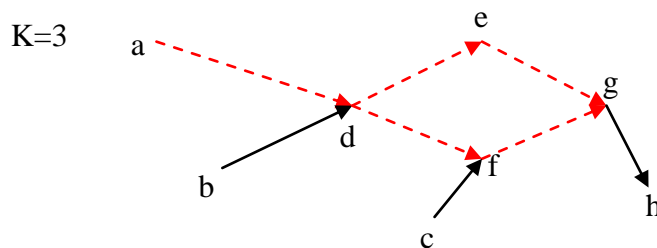


Figura 4.4: Exemplo de percurso em AIG para a determinação do fator de não-monotonicidade

O custo é calculado inicialmente para todos os caminhos do grafo e atualizado dinamicamente de acordo com os deslocamentos das células. Com a necessidade de recalculá-los apenas os NMFs dos caminhos que englobam as células deslocadas, o tempo gasto em computar a função custo cai drasticamente.

Tabela 4.1: Matriz de fatores de não-monotonicidade

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
a	0	X	X	#	#	#	#	K
b	-	0	X	#	#	#	#	K
c	-	-	0	X	X	#	#	#
d	-	-	-	0	#	#	#	#
e	-	-	-	-	0	X	#	#
f	-	-	-	-	-	0	#	#
g	-	-	-	-	-	-	0	#
h	-	-	-	-	-	-	-	0

Fonte: PLAZA, et al., 2008

4.4 Análise Estática de Atrasos

Com o intuito de possibilitar que os atrasos nos caminhos críticos sejam reduzidos através de transformações lógicas, o posicionador gera tabelas com os atrasos máximos e mínimos estimados para os caminhos entre cada uma das entradas e saídas do circuito. Os atrasos são calculados pela soma dos atrasos em cada nodo, obtidos através das informações de atraso da célula AND2 da biblioteca, com os atrasos devido às interconexões desses nodos. Os valores das tabelas são então comparados com os valores determinados no arquivo de limites do projeto. Caso algum valor desrespeite os limites do projeto, a ferramenta de síntese lógica pode manipular o AIG reduzindo a quantidade de nodos no caminho crítico.

4.5 Análise de Área

Sendo n o número de células, l e h respectivamente a largura e a altura da célula AND2 e A a área disponível para o posicionamento, no caso de termos

$$A < n \cdot l \cdot h ,$$

a área disponível é insuficiente para o posicionamento das células e, portanto, o AIG que representa o circuito deve ser retornada a etapa de síntese lógica, com a informação extra do número máximo de nodos que ela poderá ter, para ser manipulada reduzindo-se o número nodos.

5 EXPERIMENTOS

Com o intuito de fornecer às próximas etapas do posicionamento (mapeamento tecnológico e pós-posicionamento) soluções otimizadas, testamos a ferramenta quanto às soluções obtidas. Nesse capítulo iremos apresentar os testes de validação do algoritmo de posicionamento do AIG, assim como a influência de suas diversas variáveis na solução obtida.

Todos os circuitos utilizados para os experimentos foram retirados do conjunto de *benchmarks* MCNC (BLAC, 2001), exceto onde explicito. As dimensões da célula AND2 utilizadas nos testes são baseadas nas dimensões de bibliotecas com as tecnologias de 45nm a 60nm.

Os valores obtidos para o NMF são calculados pelo próprio programa durante o posicionamento. A análise estática de atrasos é feita após a conclusão do recozimento simulado considerando o atraso de cada nodo AND2 como 50ns e o atraso devido às interconexões como 0.05ns/nm. O comprimento da interconexão entre dois nodos é aproximado pela distância euclidiana entre os mesmos.

5.1 Validação dos Resultados

Para validarmos o algoritmo, posicionamos um AIG correspondente ao circuito de uma porta NAND de 8 entradas (NAND8) que não pertence ao conjunto de *benchmarks*, porém que foi selecionada por facilitar o entendimento. O posicionamento ideal do AIG é conhecido e é apresentado na Figura 5.1.

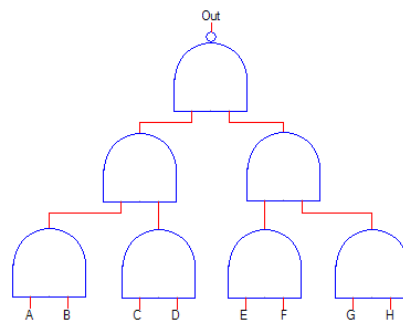


Figura 5.1: AIG do circuito NAND8 posicionada idealmente

Posicionamos o AIG em áreas de tamanhos distintos para compreendermos a influência destas nos resultados e verificarmos que, de acordo com os limites físicos, o algoritmo sempre busca o resultado ótimo. Na Figura 5.2, podemos ver os resultados obtidos para o posicionamento da NAND8 em quatro áreas diferentes. Notamos que para todos os casos os resultados assemelham-se com o caso ideal apresentado na

Figura 5.1. De acordo com a função de custo aplicada, o algoritmo sempre converge para um resultado que prioriza os caminhos monótonos, aproximando-se ou não do melhor resultado dependendo dos parâmetros utilizados para o recozimento simulado.

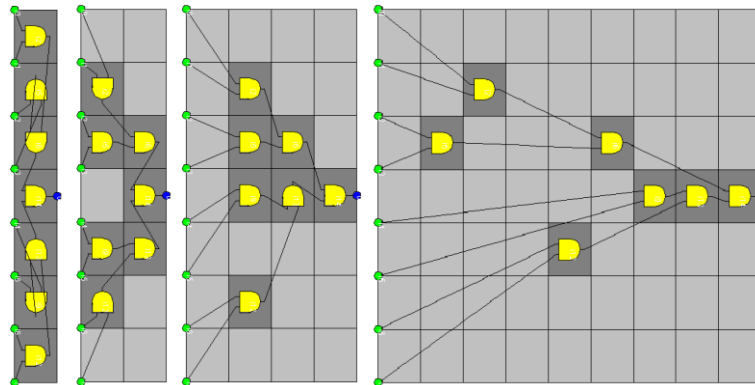


Figura 5.2: Resultado do posicionamento da NAND8 em diversas áreas

A área disponível tem grande influência no resultado. No gráfico da Figura 5.3 podemos ver que, quanto maior o espaço para o posicionamento, a solução converge para o custo mínimo $NMF=1$. No entanto, há um acréscimo nas possibilidades de soluções, e o tempo de convergência do algoritmo aumenta proporcional ao quadrado área.

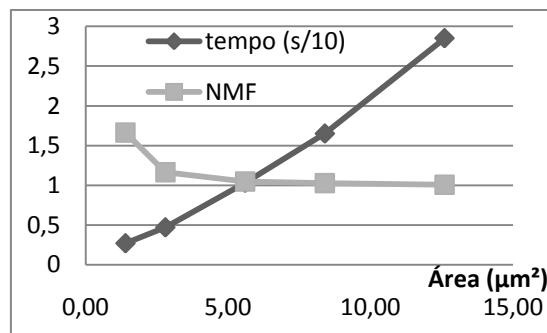


Figura 5.3: Influência da área disponível no NMF e no tempo de execução do posicionamento da NAND8

Na Figura 5.4, apresentamos o resultado do posicionamento da ferramenta em um circuito maior em duas áreas distintas. Quanto maior a área disponível torna-se notável o esforço da ferramenta para a obtenção de um resultado monótono.

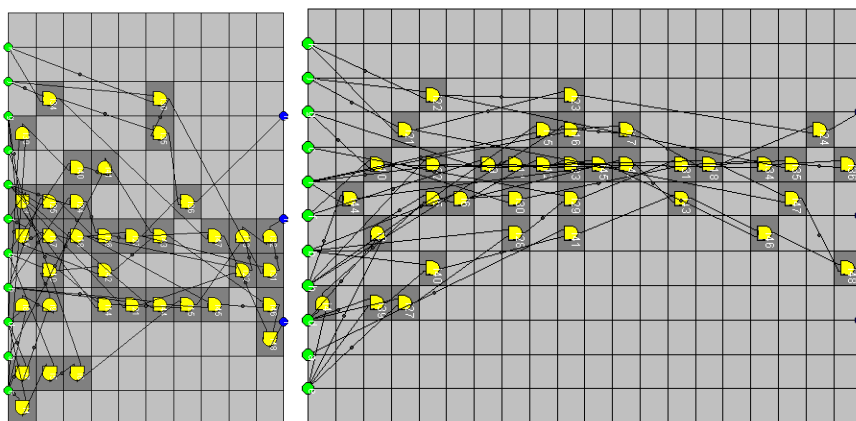


Figura 5.4: Resultados do posicionamento do circuito *cm85a* em limites físicos distintos

A Figura 5.5 mostra a evolução do custo (NMF) das soluções obtidas durante o posicionamento do circuito cm85a na área maior (Figura 5.4). Inicialmente qualquer solução é aceita, mesmo que o custo aumente. Conforme a temperatura do algoritmo diminui, a aceitação das soluções se torna mais restritiva e apenas as soluções que reduzem o custo são aceitas. No final o algoritmo converge a uma solução mínima local, cuja qualidade depende do fator de resfriamento. Quanto mais lento for o resfriamento, maior será a probabilidade de o resultado atingir o mínimo absoluto (GEREZ, 1998).

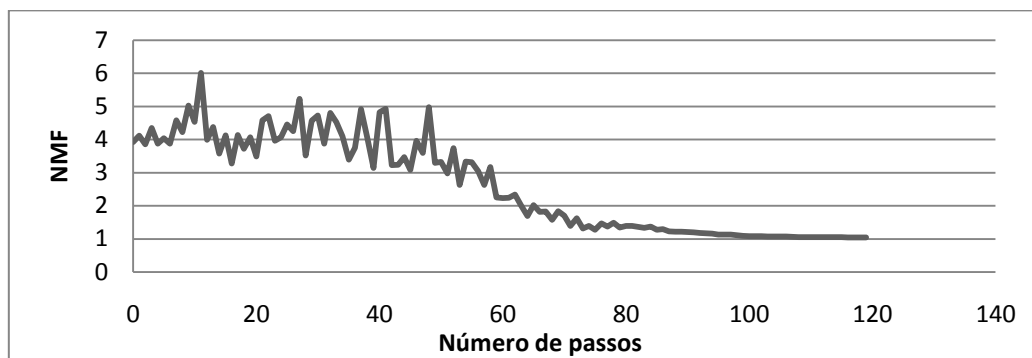


Figura 5.5: Evolução do custo das soluções durante o Recozimento Simulado

5.2 Influência dos Parâmetros do Recozimento Simulado

Analizamos o desempenho da ferramenta de posicionamento variando os dois parâmetros básicos do algoritmo de recozimento simulado, a temperatura e o fator de resfriamento, para verificarmos seus respectivos impactos nas soluções quanto ao atraso médio, ao NMF, e ao tempo à obtenção do resultado.

5.2.1 Influência da Temperatura Inicial de Recozimento

Posicionamos um conjunto de AIGs usando nossa ferramenta posicionadora e obtemos, com a temperatura inicial igual a 0,01 e o fator de resfriamento igual a 0,75, os resultados apresentados na Tabela 5.1. O atraso médio corresponde à média dos atrasos entre as entradas e saídas do AIG. O Tempo indica o tempo total de execução do algoritmo de recozimento. Esses valores servirão como referência para os testes realizados com a variação da temperatura.

Tabela 5.1: Resultados do posicionamento com temperatura $T=0,01$

Circuito	Nodos	Atraso Médio (ns)	NMF	Tempo (s)
b1	19	231,11	1,16	0,14
cm42a	32	272,92	1,47	1,77
cm82a	28	404,39	1,82	1,28
cm85a	50	569,07	1,87	4,56
majority	23	494,60	1,59	0,29
teste	14	355,50	1,91	0,12

x2	71	662,57	1,18	9,10
z4ml	58	549,11	1,83	6,11
Média	36,88	442,41	1,61	2,92

Variando a temperatura de recozimento de 0,1 a 100, obtemos os resultados médios apresentados na Tabela 5.2. Os valores apresentados são com base nos valores iniciais da Tabela 5.1. Notamos que à medida que o NMF diminui, o atraso médio também diminui como proposto em (PLAZA, et al., 2008) se aproximando cada vez mais do limite, porém o tempo de execução cresce quadraticamente. Portanto o tempo necessário para o posicionamento deve ser levado em consideração para a escolha da temperatura de recozimento.

Tabela 5.2: Impacto da variação da temperatura no posicionamento

Temperatura	Atraso Médio	NMF	Tempo
0,1	-3,62%	-7,67%	+26%
1	-4,73%	-8,48%	+129%
10	-5,94%	-9,07%	+322%
100	-6,49%	-9,42%	+707%

5.2.2 Influência do Fator de Resfriamento do Recozimento

Utilizando como parâmetros iniciais a temperatura de recozimento igual a 10 e o fator de resfriamento igual a 0,1, obtemos os atrasos médios entre entradas e saídas, os NMFs e os tempos de execução apresentados na Tabela 5.3.

Tabela 5.3: Resultados do posicionamento com fator de resfriamento CF=0,1

Circuito	Nodos	Atraso Médio (ns)	NMF	Tempo (s)
b1	19	235,50	1,20	0,08
cm42a	32	276,00	1,14	1,01
cm82a	28	456,844	1,98	1,07
cm85a	50	535,99	1,74	3,27
majority	23	467,62	1,51	0,11
teste	14	367,58	1,98	0,05
x2	71	637,00	1,33	5,32
z4ml	58	652,35	1,98	3,78
Média	36,88	453,62	1,61	1,84

Mantendo a temperatura inicial constante, realizamos uma série de experimentos medindo os resultados obtidos para uma variação do fator de resfriamento de 0,1 a 0,95. Os resultados relativos aos valores da Tabela 5.3 são mostrados na Tabela 5.4. Notamos

que ao diminuir a velocidade de resfriamento (aumentar o CF) a ferramenta melhora a qualidade da solução quanto ao atraso médio e ao NMF. No entanto o tempo de execução cresce exponencialmente. Para reduzir o atraso em média 11%, o tempo de execução aumenta em 1789%.

Tabela 5.4: Impacto da variação do fator de resfriamento no posicionamento

Fator de Resfriamento	Atraso Médio	NMF	Tempo
0,4	-1,69%	-5,44%	+61,41%
0,6	-2,43%	-7,73%	+319,57%
0,8	-6,31%	-8,73%	+705,43%
0,95	-10,75%	-14,13%	+1789,46%

Assim como na escolha da temperatura inicial, para se escolher o fator de resfriamento se deve avaliar o compromisso entre o atraso médio alvo e o tempo de execução do algoritmo. Constatamos que conforme o NMF é reduzido, o atraso médio também decresce.

6 TRABALHOS FUTUROS

Para completar o fluxo proposto de posicionamento das células de um circuito, devem ser implementadas ainda as etapas de mapeamento tecnológico e de validação e otimização do posicionamento.

6.1 Mapeamento Tecnológico

O mapeamento tecnológico consiste em substituir um ou mais nodos do AIG por células padrão da biblioteca disponível.

O mapeamento é feito com a varredura do grafo em busca de padrões cujas funções lógicas são conhecidas e presentes na biblioteca. O objetivo é cobrir todo o grafo usando os padrões conhecidos com uma solução ótima segundo uma função de custo (MISHCHENKO, et al., 2004).

Na ferramenta de posicionamento proposta, a função custo deve garantir a prioridade de mapeamento aos padrões que englobam nodos posicionados próximos um ao outro, mantendo-se assim a solução obtida após o posicionamento do AIG.

As células mapeadas são posicionadas com a posição média dos nodos das quais elas derivam. Nesse ponto do posicionamento algumas das regras do projeto podem ser infringidas. Cabe a etapa de pós-posicionamento corrigir as infrações.

6.2 Pós-posicionamento

A última etapa da ferramenta de posicionamento consiste na avaliação e na melhoria da solução.

A solução obtida após o mapeamento deve ser avaliada quanto aos atrasos entre as entradas e saídas, a área total ocupada, a densidade de células, e o consumo do circuito. Caso as especificações do projeto tenham sido atendidas, o posicionador desloca minimamente as células, impedindo que elas se sobreponham, de forma a respeitar as regras para o posicionamento de acordo com a tecnologia empregada e reavalia a solução que, caso aprovada, é final.

Se alguma das especificações não for atendida, o posicionador busca uma nova solução que atenda as especificações. Para isso o algoritmo de recozimento simulado (*simulated annealing*) pode ser aplicado às células em busca da nova solução. A temperatura de recozimento inicial utilizada pode ser maior conforme a diferença entre a solução encontrada e a solução esperada. Conforme o estudo realizado na sessão 2.4, outras técnicas para redução de atrasos também podem ser utilizadas. Diferentes funções de custo são criadas para a redução da densidade local, e do atraso. Elas podem

ser aplicadas simultaneamente. Para a redução da área e do consumo é necessário voltar aos passos anteriores e alterar o mapeamento tecnológico ou o circuito resultante da etapa de síntese lógica para se obter um resultado com menos células e/ou com células com menor consumo.

A ferramenta Domino (DOLL, et al., 1994), também utilizada pelo posicionador Kraftwerk (OBERMEIER, et al., 2005), remove sobreposições e otimiza o posicionamento. Ela pode ser utilizada para essa etapa.

6.2.1 Alocação de Espaços em Branco

A alocação de Espaços em Branco possibilita a redução de congestionamentos de interconexões, facilitando o roteamento das células. No entanto, a alocação de espaços em branco move as células de suas posições ideais resultando em um acréscimo do comprimento total de interconexões (NAM, et al., 2007).

A densidade máxima $d_{máx}$ de células em qualquer região da área determinada para o posicionamento é especificada em termos de porcentagem de ocupação. A densidade de espaços em branco é $1 - d_{máx}$.

O cálculo da densidade local pode ser feito sobre qualquer área onde é possível o posicionamento das células do circuito. Cabe ao usuário determinar o tamanho dessa área de acordo com as dimensões das células de sua biblioteca.

A densidade local é a porcentagem da área ocupada por células. Caso a área do considerada seja muito pequena, menor do que as células, a densidade será, para quase todos os locais, ou zero ou um. Se a área for muito grande, comparável a área total de posicionamento, a densidade obtida omitirá as áreas de alta densidade. Devemos adotar um valor ao meio-termo para produzirmos os resultados práticos esperados.

Também é possível verificar-se a densidade total do circuito, e caso essa supere a densidade estipulada para o projeto, não poderemos obter o resultado esperado apenas através do reposicionamento das células. Será necessário recomeçar o mapeamento tecnológico reduzindo-se a área ocupada pelas células escolhidas.

6.2.2 Redução de Atrasos

Pesquisas mostram que, se eliminarmos os caminhos não-monótonos do circuito, reduziremos os atrasos deste. Portanto, (PLAZA, et al., 2008) propõe um método para redução de atrasos na etapa de pós-posicionamento, utilizando-se simulação funcional, e transformações lógicas sobre os caminhos mais susceptíveis a melhorias detectados através do cálculo de seus fatores de monotonicidade (ver sessão 2.4.5).

Diversos trabalhos propõem técnicas diferenciadas para a otimização de atrasos, entre elas podemos citar a inserção de buffers com o redimensionamento de células (KANNAN, et al., 1994) (LI, et al., 2005), a realocação de células (AJAMI, et al., 2001), e a transformação de sinais (CHANG, et al., 2007), todas explicadas em maiores detalhes na sessão 2.4.

7 CONCLUSÕES

Nas tecnologias atuais, abaixo de 0,1 μm , as interconexões são responsáveis pela maior parte dos atrasos nos circuitos. O fluxo de projeto de circuitos integrados, como ele é, não possibilita de forma otimizada a otimização lógica do circuito visando a redução de atrasos de interconexões. A etapa de síntese lógica normalmente ignora os efeitos das interconexões e procura otimizar ao máximo o atraso nos *gates*. Também percebemos que, por causa das diferentes estruturas de dados utilizadas durante as fases do projeto, se torna difícil retornar dados referentes ao resultado do posicionamento e da influência das interconexões à etapa de síntese lógica. Para considerarmos os efeitos das interconexões e reduzi-los com a utilização das ferramentas de síntese lógica, é necessária a inclusão de novas etapas intermediárias ao fluxo *standard cell* atual. Essas etapas devem ser capazes de estimar os efeitos das interconexões e retornar as informações necessárias para que a ferramenta de otimização lógica seja capaz de aperfeiçoar o circuito as considerando.

Nós propomos algumas alterações no fluxo *standard cell* atual para que os atrasos das interconexões possam ser identificados e otimizados durante a síntese lógica. Em nosso fluxo de projeto, utilizamos AIGs como estruturas de representação do circuito. Além de ser uma estrutura de fácil manipulação lógica, o AIG se assemelha essencialmente com o circuito final após mapeamento tecnológico. Quando a otimização lógica convencional do circuito é concluída, determinamos a área cujos diferentes módulos devem ocupar baseados nos AIGs de cada módulo e, antes do mapeamento tecnológico, posicionamos cada AIG em sua área e avaliamos a qualidade do resultado quanto aos atrasos de *gate* e de interconexões. Caso os objetivos de projeto não sejam obtidos, o AIG, com a adição das informações de atrasos de interconexões, é repassado à etapa de otimização lógica, que considerará as novas informações para a re-otimização do circuito. Depois de validado o posicionamento do AIG, o mapeamento tecnológico agrupa os nodos do AIG de acordo com suas proximidades geométricas, mantendo as posições médias estabelecidas pelo posicionador do AIG. Por fim, a etapa de pós-posicionamento irá validar o posicionamento resultante do mapeamento e o aperfeiçoar de acordo com os objetivos de projeto (área, atraso ou potência). O circuito posicionado é roteado com os algoritmos existentes de roteamento.

O fluxo proposto aumenta a interação entre as etapas de projeto do circuito devido à utilização de uma estrutura padrão para otimização lógica, posicionamento e mapeamento. Ainda com a possibilidade de iteração, a otimização lógica pode ser adaptada para otimizar também os atrasos de interconexões.

Implementamos uma ferramenta que realiza o posicionamento dos nodos de um AIG. O posicionamento é feito de acordo com uma função que quantifica a monotonicidade dos caminhos do circuito. Posicionando o circuito de forma a deixar os caminhos monótonos implica diretamente na redução dos atrasos de interconexão. Os

testes feitos com nossa ferramenta comprovam essa relação. Conforme modificamos os parâmetros de posicionamento para obter um resultado com caminhos mais monótonos, observamos a redução do atraso médio entre as entradas e saídas do circuito, em detrimento do tempo de execução do algoritmo. Vemos também que a área disponível para o posicionamento influi na monotonicidade do resultado. Em grandes áreas, onde a densidade de nodos é baixa, o posicionador tem maior liberdade para obter resultados mais monótonos, comparado aos resultados obtidos em pequenas áreas, onde a densidade é alta.

Nosso posicionador analisa os atrasos estáticos e os compara com os objetivos de projeto. As informações de atrasos de interconexões são adicionadas a estrutura do AIG, e, caso os resultados obtidos não sejam validados, elas são repassadas à ferramenta de otimização lógica. A otimização lógica pode usar as informações de atraso na síntese de um novo AIG otimizada para os atrasos de *gate* e também de interconexões.

Trabalhos futuros devem desenvolver técnicas para considerar as informações dos atrasos das interconexões na etapa de otimização lógica. Também devem adaptar uma função de custo que possibilite o mapeamento dos nodos do AIG para as células da biblioteca de acordo com a proximidade geométrica. Uma vez concluídas as ferramentas que implementam as alterações no fluxo de projeto, deve-se estudar como realizar a iteração, que inclui a síntese lógica e o posicionamento do AIG, garantindo a convergência desta a um resultado ótimo e a automatização dos parâmetros envolvidos.

REFERÊNCIAS

- AJAMI, A. H. e PEDRAM, M. 2001.** Post-Layout Timing-Driven Cell Placement Using an Accurate Net Length Model with Movable Steiner Points. *Proceedings of the 2001 Conference on Asia South Pacific Design Automation*. 2001, 595-600.
- AKERS, S. B. 1978.** Binary Decision Diagrams. *IEEE Transactions on Computers*. 1978, Vols. c-27, 6, pp. 509-516.
- BERAUDO, G. e Lillis, J. 2003.** Timing Optimization of FPGA Placements by Logic Replication. *Proceedings of the 40th Conference on Design Automation*. 2003, pp. 196-201.
- BLAC. 2001.** BLAC CAD Group Benchmarks. *Binghamton Laboratory for Algorithms, Circuits, and Computer Aided Design*. [Online] 2001. <http://vlsicad.cs.binghamton.edu/benchmarks.html>.
- CHANG, C-W., et al. 2000.** Fast Post-placement Rewiring Using Easily Detectable Functional Symmetries. *Annual ACM IEEE Design Automation Conference*. 2000, pp. 286-289.
- CHANG, K-H., MARKOV, I. L. e BERTACCO, V. 2007.** Safe Delay Optimization for Physical Synthesis. *ASP-DAC '07*. 2007, pp. 628-633.
- DOLL, K., JOHANNES, F. M. e ANTREICH, K. J. 1994.** Iterative Placement Improvement by Network Flow Methods. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1994, Vol. vol. 13, no. 10.
- FLYNN, Michael J., HUNG, Patrick e RUDD, Kevin W. 2009.** Deep-Submicron Microprocessor Design Issues. *IEEE Micro*. 1, 2009.
- GAJSKI, D. D. e KUHN, R. H. 1983.** New VLSI Tools. *IEEE Computer*. 1983, Vol. 16, 12, pp. 11-14.
- GEREZ, Sabih H. 1998.** *Algorithms for VLSI Design Automation*. Baffins Lane : John Wiley & Sons Ltd., 1998. ISBN 0-471-98489-2.
- GOSTI, W., et al. 1998.** Wireplanning in Logic Synthesis. *ICCAD*. 1998, pp. 26-33.
- HELLERMAN, Leo. 1963.** A Catalog of Three-Variable Or-Invert and And-Invert Logical Circuits. *IEEE Transactions on Electronic Computers*. 3, 1963, Vols. EC-12, pp. 198-223.
- HWANG, C. e PEDRAM, M. 2006.** Timing-Driven Placement Based on Monotone Cell Ordering Constraints. *Proceedings of the 2006 conference on Asia South Pacific design automation*. 2006, pp. 201-206.
- KANNAN, L. N., SUARIS, P. R. e FANG, H. 1994.** A Methodology and Algorithms for Post-placement Delay Optimization. *Annual ACM IEEE Design Automation Conference*. 1994, pp. 327-332.
- KEUTZER, K., NEWTON, A. R. e SHENOY, N. 1997.** The Future of Logic Synthesis and Physical Design in Deep-submicron Process Geometries. *International Symposium on Physical Design*. 1997, pp. 218-224.

- KUEHLMANN, A., et al. 2002.** Robust Boolean Reasoning for Equivalence Checking and Functional Property Verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2002, Vol. 21, 12, pp. 1377-1394.
- LI, C., KOH, C-K. e MADDEN, P. 2005.** Floorplan Management: Incremental Placement for Gate Sizing and Buffer Insertion. *Proceedings of the 2005 conference on Asia South Pacific design automation*. 2005, pp. 349 - 354.
- MISHCHENKO, A., et al. 2004.** FRAIGs: A Unifying Representation for Logic Synthesis and Verification. *International Conference on Computer Aided Design*. 2004.
- MISHCHENKO, A., et al. 2006.** Improvements to Combinational Equivalence Checking. *International Conference on Computer Aided Design*. 2006, pp. 836-843.
- NAM, G-J. e CONG, J. 2007.** *Modern Circuit Placement*. New York : Springer Science, 2007. ISBN 978-0-387-36837-5.
- NAM, G-J. 2007.** ISPD 2005/2006 Placement Contest Updates. *ISPD*. [Online] 2007. <http://www.ispd.cc/slides/archives/ispd2007/slides07/placement-updates.pdf>.
- OBERMEIER, B., RANKE, H. e JOHANNES, F. M. 2005.** Kraftwerk — A Versatile Placement Approach. *ISPD'05*. 2005, pp. 242-244.
- PLAZA, S. M., L., MARKOV, I. e BERTACCO, V. M. 2008.** Optimizing Non-Monotonic Interconnect using Functional Simulation and Logic Restructuring. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 2008, Vol. 27, No. 12.
- RABAEY, Jam M, CHANDRAKASAN, Anantha P. e NIKOLIC, Borivoje. 2003.** *Digital Integrated Circuits*. 2. s.l. : Paperback, 2003.
- ROY, J. A., et al. 2005.** Capo: Robust and Scalable Open-Source Min-Cut Floorplacer. *Proceedings of the 2005 international symposium on Physical design*. 2005, pp. 224-226.
- SARRAFZADEH, M., WANG, M. e YANG, X. 2002.** *Modern Placement Techniques*. Norwell : Kluwer Academic Publishers, 2002. ISBN 1-4020-7221-X.
- SYNOPSYS®. 2007.** *Using the Synopsys® Design Constraints Format - Version 1.7*. 2007.
- TAGHAVI, T., YANG, X. e CHOI, B-K. 2005.** Dragon2005: Large-scale Mixed-size Placement Tool. *Proceedings of the 2005 international symposium on Physical design*. 2005, pp. 245-247.
- THEIS, T. N. 2000.** The Future of Interconnection Technology. *IBM Journal of Research and Development*. 2000, Vol. 44, No. 3.
- WANG, M., YANG, X. e SARRAFZADEH, M. 2000.** Dragon2000: Standard-cell Placement Tool for Large Industry Circuits. *ICCAD-2000*. 2000, pp. 260-263.

ANEXO <ESPECIFICAÇÕES DO FORMATO SDC>

Especificação da versão de SDC:

Commands	Arguments
set_sdc_version_value	value

Especificação das unidades:

Commands	Arguments
set_units	-capacitance <i>cap_unit</i> -resistance <i>res_unit</i> -time <i>time_unit</i> -voltage <i>voltage_unit</i> -current <i>current_unit</i> -power <i>power_unit</i>

Especificação dos objetivos de projeto:

Type of information	Commands
Operating conditions	set_operating_conditions
Wire load models	set_wire_load_min_block_size set_wire_load_mode set_wire_load_model set_wire_load_selection_group
System interface	set_drive set_driving_cell set_fanout_load set_input_transition set_load set_port_fanout_number
Design rule constraints	set_max_capacitance set_max_fanout set_max_transition set_min_capacitance
Timing constraints	create_clock create_generated_clock group_path set_clock_gating_check set_clock_groups set_clock_latency set_clock_sense

	set_clock_transition set_clock_uncertainty set_data_check set_disable_timing set_ideal_latency set_ideal_network set_ideal_transition set_input_delay set_max_time_borrow set_output_delay set_propagated_clock set_resistance set_timing_derate
Timing exceptions	set_false_path set_max_delay set_min_delay set_multicycle_path
Area constraints	set_max_area
Power constraints	set_max_dynamic_power set_max_leakage_power
Logic assignments	set_case_analysis set_logic_dc set_logic_one set_logic_zero

Especificação dos objetos de projeto:

Design object	Access command	Description
Design	current_design	A container for cells. A block.
Clock	get_clocks all_clocks	A clock in a design. All clocks in a design.
Port	get_ports all_inputs all_outputs	An entry point to or exit point from a design. All entry points to a design. All exit points from a design.
Cell	get_cells	An instance of a design or library cell.
Pin	get_pins	An instance of a design port or library cell pin.
Net	get_nets	A connection between cell pins and design ports.
Library	get_libs	A container for library cells.
Lib_cell	get_lib_cells	A primitive logic element.
Lib_pin	get_lib_pins	An entry point to or exit point from a lib_cell.
Register	all_registers	A sequential logic cell.