

A.A.S. TELES* e L.F.P. SOUZA**

SUMÁRIO

O objetivo deste artigo é descrever o algoritmo e a implementação do extrator de circuitos integrados NMOS desenvolvido no NCE/UFRJ, dando ênfase às modificações feitas para a versão 3.0, que permitiram a extração da capacitância dos fios e resistência dos transistores.

ABSTRACT

The intent of this paper is to describe the algorithm and implementation of a NMOS IC extractor developed at NCE/UFRJ. The emphasis is in changes to version 3.0, which allow the extraction of node capacitance and resistance of transistors.

* Mestre em Sistemas Digitais (UFRJ); Inteligência Artificial, Compiladores, Cad para VLSI; Núcleo de Computação Eletrônica da UFRJ, Cidade Universitária, CCMN, Rio de Janeiro, Caixa Postal 2324, CEP 20001, telefone (021) 290-3212, ramal 293;

** Estudante de Informática (UFRJ); Software Básico, Sistemas Operacionais, Compiladores, Cad para VLSI; Núcleo de Computação Eletrônica da UFRJ, Cidade Universitária, CCMN, Rio de Janeiro, Caixa Postal 2324, CEP 20001, telefone (021) 290-3212, ramal 293.

No Sistema de Apoio a Projetos em Circuitos Integrados NMOS atualmente implementado no NCE/UFRJ, adotou-se a metodologia Mead & Conway em que o circuito é projetado em função de uma unidade padrão chamada Lambda, que varia de fabricante para fabricante. A primeira fase do projeto consiste na definição do funcionamento do chip numa linguagem lógica e a sua simulação funcional (SIMF). A partir do circuito validado, as PLA's são geradas automaticamente (MKPLA), e podem ser aproveitados trechos padrões (biblioteca de células), entretanto grande parte dos blocos funcionais e todas as suas conexões precisam ser feitas através de editores gráficos (EDMOS e EDCI). Terminada a edição o projetista tem um banco de dados que armazena hierarquicamente a geometria do circuito, baseado na linguagem Cif (formato intermediário da Caltech), e poderia gerar uma fita com a descrição em Cif e enviá-lo ao fabricante, entretanto como a edição é um processo manual faz-se necessário algumas ferramentas de validação, como verificador de regras estáticas elétricas e lógicas (ESTÁTICO), simulador lógico (SIML), simulador elétrico (SPICE) e avaliador de desempenho (CRÍTICO).

Como as ferramentas de verificação lógica e elétrica não manipulam eficientemente uma descrição geométrica, torna-se necessário a construção um mecanismo que a partir do lay-out extraia uma descrição a nível de fios e transistores. Tanto o extrator (EXTRA1) como o verificador de regras geométricas (DRC) precisam da mesma imagem não hierárquica do lay-out, então foi criada uma ferramenta intermediária (GMAP), que do banco de dados gera a imagem rastreada do circuito e um arquivo com as coordenadas dos pontos nomeados pelo projetista. Estes dois arquivos constituem a entrada do extrator.

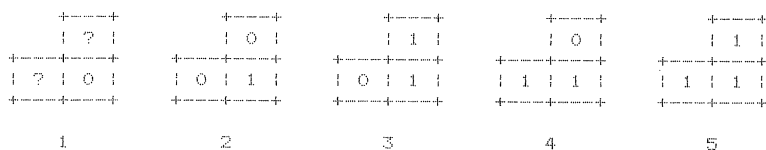
DESCRIÇÃO

Projetar um circuito integrado é basicamente determinar quais as máscaras geométricas a serem usadas na construção de cada camada. As camadas na tecnologia NMOS são, difusão, poli-silício, metal, corte e implante. As camadas difusão, poli-silício e metal são condutoras, a camada de corte faz contato entre uma camada de metal e outra de poli-silício ou difusão. Quando um fio de poli-silício cruza um de difusão, forma-se um transistor onde o poli-silício é o gate e cada pedaço da difusão um dos drenos.

Os editores gráficos criam uma descrição do circuito em CIF a partir dos lay-outs projetados. Esta descrição pode ser representada por um mapa de pixel, onde cada byte é um quadrado de lambda por lambda do lay-out e seus bits indicam a existência ou não de cada camada neste ponto.

O mapa de pixel descrito acima é a principal entrada do extrator, e nele é aplicado o algoritmo de ilhas e lagos descrito em BACKER [1]. O algoritmo procura resolver o seguinte problema, dada uma matriz de zeros e uns, onde os uns indicam terra e os zeros água, atribuir um código a cada região contínua de terra e determinar sua área. A solução é um autômato bidimensional que varre o circuito da esquerda para direita e de cima para bai

xo atribuindo um código a cada posição, este código é função do código da posição imediatamente acima e do código da imediatamente a esquerda. O código zero indica que a região é água. A função é descrita abaixo para cada configuração.



- 1 - Se a posição atual na matriz for zero, o código correspondente é zero por definição;
- 2 - Se a posição atual for um e a superior e esquerda forem zeros, uma nova região foi encontrada, logo é criado um novo código e atribuído a este ponto;
- 3 - Se a posição atual e superior forem um, e a esquerda zero, significa que se está sobre uma aresta vertical de uma região, logo o código da posição atual será o mesmo da posição superior;
- 4 - Se a posição atual e a esquerda forem um, e a superior for zero, significa que se está sobre uma aresta horizontal de uma região, logo o código da posição atual será o mesmo da posição esquerda;
- 5 - Se a posição atual, a superior e a esquerda forem um, temos dois casos:
 - a) Os códigos das regiões superior e esquerda são os mesmos, logo se está dentro de uma região contínua, o código do ponto será o código desta região, ou seja, o código das posições superior e esquerda;
 - b) Os códigos são diferentes, então se está sobre a união de duas partes da mesma região em formato de um L invertido, logo deve ser feita uma equivalência entre estas duas regiões, e o código equivalente será atribuído a posição atual.

Como foi mencionado deve ser montada uma relação de equivalência, que terá suas características discutidas abaixo.

O algoritmo descrito acima extrai apenas fios. Entretanto o mesmo autômato pode ser usado para identificar regiões periféricas aos transistores, desta forma numerando-os e de terminando suas portas e drenos, além de fornecer elementos para determinar suas dimensões. Algumas destas situações são indicadas a seguir, onde D indica difusão, P poli-silício e T difusão + poli-silício.

desta linha. Paralelo a esta linha teremos uma matriz que representa uma função que dado um ponto e uma camada fornece qual o código da camada naquele ponto. O autômato em que se baseia o algoritmo necessita do código da posição superior e da esquerda ao ponto que se está analisando, entretanto a matriz que armazena os códigos não precisa possuir os códigos destas duas linhas, basta que se considere os códigos a direita da posição sendo analisada como referentes a linha superior, e que coloque os novos códigos na matriz a medida que se anda para direita. Desta forma os códigos a esquerda da posição sendo analisada se referirão a linha atual. Na realidade como o mapa é percorrido no mesmo sentido que um texto são necessitados ter a cada momento na memória um pixel, e a matriz que representa os códigos, como tem seu acesso bastante localizado, se presta otimamente para o mapeamento.

Para podermos representar os códigos dos transistores e tratar as configurações das camadas de forma melhor, criamos uma pseudo-camada que chamamos de TRANSISTOR e modificamos os bits de cada ponto do mapa em análise. Esta modificação faz com que o mapa represente mais fielmente as disposições físicas reais das camadas. As modificações são duas: liga-se o bit que indica a presença de transistor nos pontos onde ocorrem difusão e polissilício, e apaga-se o bit de difusão nos pontos onde ocorre polissilício, o que se aproxima mais da construção das portas (gates) como indicado abaixo.



A relação de equivalência necessária para o algoritmo de ilhas e lagos é uma relação reflexiva, simétrica e transitiva. Toda vez que uma nova região de código N é identificada, obrigamos que $N \ R \ N$, ou seja $EQUIV(N) = N$. Para cada conjunto de regiões equivalentes teremos uma lista de equivalência construída de tal modo que $EQUIV(N) = N$. Desta forma, faz-se a equivalência de duas regiões previamente distintas N e M , seguindo as listas de equivalência de N e M até, em cada lista, termos $EQUIV(T) = T$. Supondo que a lista de equivalência de N leve a uma região T_n e a de M , a uma região T_m , faz-se $EQUIV(T_m) = T_n$, se $T_n < T_m$. Caso contrário, faz-se $EQUIV(T_n) = T_m$. Desta forma garantimos a transitividade da relação, ou seja dada tres regiões A , B e C , se obrigarmos $A \ R \ B$, e mais tarde $B \ R \ C$, automaticamente será montada a relação $A \ R \ C$, independente dos códigos numéricos de A , B e C .

Para a extração dos transistores, utiliza-se o automato do algoritmo de ilhas e lagos, identificando os transistores e suas regiões de porta e drenos. Para isto é montada uma lista onde é depositado um registro para cada pixel de perímetro entre difusão e transis

tor. Desta forma depois de todo o mapa percorrido e esta lista ordenada pelo código do transistor pode se percorrer esta lista e identificar quais as regiões de difusão conectadas a cada transistor. Podem ocorrer então tres casos, se sõ houver uma região de di fusão, então na verdade não se tratava de um transistor real, mas provavelmente um contato entre difusão e poli-silício, que ẽ então desprezado. Se houver duas regiões de di fusão então ẽ um transistor normal, se houver mais de duas regiões, então ẽ um transis - tor, que embora teoricamente possível, deve ser indicado como inválido.

Outras informações também são extraídas dos transistores, a existência de implante, a existência de corte junto ao transistor, o perímetro entre poli-silício e transistor, a área do transistor e a direção de cada fronteira entre difusão e transistor. Com esta di reção sabe-se mais tarde se na região de fronteira a difusão estava a esquerda, direita, acima ou abaixo do transistor. A partir destes dados pode se avaliar a capacitância e as dimensões do transistor.

A capacitância de porta do transistor ẽ simplesmente proporcional a área do transistor. Para as dimensões existem dois algoritmos, um para transistores simples, e outro para transistores que apresentem um contato entre porta e dreno, como pull-ups e alguns pull-downs.

Se o transistor não apresenta contato, então considera-se que suas dimensões sejam constantes ao longo de seu comprimento, então a largura ẽ obtida dividindo-se o perímetro em difusão por dois, e o comprimento dividindo-se o perímetro em poli-silício por dois, de pois dos comprimentos calculados ẽ feito uma comparação para verificar de o produto da largura pelo comprimento estã entre 80% e 120% da área real do transistor, caso esteja fora desta faixa o usuário ẽ advertido.

Nos transistores que apresentam contatos, geralmente estes ocorrem entre a porta e um dos drenos, então o corte fica numa região de fronteira entre o transistor e a difusão, logo a difusão neste ponto ẽ mais larga que no restante do transistor para atender às regras geométricas de construção de contato. O extrator utiliza a informação de direção de cada pixel das fronteiras para separar o perímetro de difusão em que ocorre o corte do perímetro normal, feito isto sabe-se que a largura efetiva serã o perímetro da região normal, que ẽ menor que da região em que ocorre o corte, e o comprimento serã a metade da diferença entre o perímetro em poli-silício e a extensão do perímetro em poli-silício devida ao alargamento do transistor; esta extensão ẽ calculada pela diferença entre os perímetros em difusão. Caso esta diferença não possa ser calculada (transistor em forma de U) então ẽ assumido o valor 2.

A capacitância ẽ calculada somando em cada região a capacitância de cada pixel, que depende unicamente da configuração das camadas naquele pixel. Entretanto dois cuidados devem ser tomados. O primeiro ẽ quando analisando a capacitância devida a separação do metal e poli-silício ou entre metal e difusão, se as regiões a princípio são diferentes na da impede que ao longo da extração apareça um contato entre estas regiões. Para contor

nar este problema estas capacitâncias não são somadas diretamente aos nos, ao invés disso são colocadas numa lista, que depois de terminada a varredura do mapa é percorrida somando-se aos nos as capacitâncias convenientes.

Outro cuidado que deve ser tomado é quanto a capacitância devida a área e ao perímetro de difusão. Esta capacitância embora pertencente aos nos, e assim tratada pela maioria das ferramentas, deve ser posta em separado, pois desta forma o simulador eletrônico SPICE pode simular mais precisamente a interação entre estas capacitâncias e os transistores.

Os registros do arquivo de saída do extrator podem ser de quatro tipos: dimensionamento, transistor, nos e comentário. Os registros de dimensionamento informam quantos transistores compõem o circuito, antes que o primeiro registro de transistor apareça, idem para os nós. Isto é muito útil nos programas que alocam memória dinamicamente.

Os registros de transistor fornecem os seguintes dados: para cada transistor, tipo (Depletion ou Enhancement), código do transistor, código da região de gate, códigos dos drenos, comprimento, largura, caracter de consistência das dimensões (advertência), coordenadas horizontal e vertical e capacitância de gate (aF).

Os registros de nos possuem, tipo do no (Input ou Normal), nome do no, código, coordenada horizontal e vertical, capacitância total, inclusive a devida a difusão (aF), área em difusão e perímetro em difusão.

O nosso formato de saída não é compatível com o de entrada do SPICE, por isto foi criada um filtro que além de modificar o formato, distribui a capacitância devida a difusão pelos transistores, ponderadamente de acordo com a largura de cada transistor.

CONCLUSÃO

A principal preocupação na implementação do extrator foi o tempo de processamento. Desta forma obrigamos a todos os arquivos sequenciais a serem bufferizados e todos os de acesso direto a serem mapeados na memória principal, onde também foram colocadas inteiramente as estruturas de tamanho médio frequentemente acessadas. Daí vem as duas limitações do programa, a largura do circuito deve ser menor ou igual a 3000 lambdas e o número de transistores, verdadeiros e falsos como os contados de emenda, não deve ultrapassar a 20000. A redução do tempo de processamento foi de 5 horas e 30 minutos, da versão anterior, para 1 hora e 20 minutos de CPU do PDP 11/70 da DEC, para um circuito de 1900 lambdas por 1900 lambdas.

Outro objetivo que era passar a extrair a capacitância dos nos e transistores, foi também conseguido. Sendo que a diferença entre a capacitância fornecida pelo extrator e a extraída manualmente por um projetista, é igual a zero, isto é, se o projetista não errar nem fizer algumas simplificações.

Os algoritmos de extração das dimensões dos transistores foram revistos, sendo o resultado atual bastante satisfatório, cobrindo todas as construções usuais.

A resistência entre transistores, devidas aos fios, não é fornecida, devido a limitações de memória e processamento.

Uma nova versão do extrator deverá ser feita ainda este ano, visando o Sistema de Apoio a Projetos CMOS, atualmente fase de definição. Embora a filosofia do novo sistema seja a não intervenção do projetista a nível de transistores e fios, acreditamos que o extrator será necessário tanto na fase de teste do sistema, como na confecção da biblioteca de células básicas a partir da qual o novo sistema montará os circuitos automaticamente.

BIBLIOGRAFIA

BACKER, Clark, "Artwork Analysis Tools for VLSI Circuits", MIT/LCS/TR-239, USA, Massachusetts Institute of Technology, 1980;

BRYANT, Randal E., "An Algorithm for MOS Logic Simulation", Lambda, USA, 1(4): 46-53, 1980;

TELES, Antonio Anibal S., "Verificação de Projetos em Circuitos Integrados", Tese MSc. COPPE - UFRJ, 1983.

FERRAMENTAS CITADAS

01. SIMF - simulador funcional, FORTRAN;
02. MKPLA - gerador automático de PLA's, FORTRAN;
03. EDMOS - editor de células, FORTRAN ou C;
04. EDCI - editor hierárquico de circuitos, C;
05. GMAP - gerador de mapa de pixels, C;
06. DRC - verificador de regras geométricas, C;
07. ESTÁTICO - verificador de regras estáticas lógicas e elétricas, C;
08. SIML - simulador lógico NMOS/CMOS, C;
09. SPICE - simulador elétrico (cedido), FORTRAN e ALGOL;
10. CRÍTICO - avaliador de desempenho (em fase de validação), C.