

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FEE - Departamento de Semicondutores,  
Instrumentos e Fotônica

Contribuição para Especificação de um  
Sistema de Geração de Células Programáveis  
para Utilização em Projeto de CI's<sup>1</sup>

Autor: Rosana C. Di Giorgio  
Orientador: Prof. Carlos I.Z. Mammana



<sup>1</sup>Tese apresentada à Faculdade de Engenharia Elétrica da UNICAMP, como parte dos requisitos exigidos para a obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA. Campinas-SP, 06 de agosto de 1990.

Este exemplar corresponde à redação final da tese  
defendida por Rosana Ceron Di Giorgio  
e aprovada pela Comissão  
Julgadora em 6 / 8 / 90.  
C. I. Z. Mammana  
Orientador

## RESUMO

Este trabalho dá a contribuição do projetista de circuitos integrados para o conjunto de especificações de um sistema de geração de células parametrizadas para uso em projeto de CI's.

Células geradas pelo sistema possuem seus layouts codificados em função de parâmetros geométricos e/ou elétricos. Estes parâmetros, bem como seus valores numéricos, devem estar contidos em um arquivo de dados, de modo que possam ser alterados pelo projetista sempre que este deseje otimizar o layout de uma dada célula. Futuras alterações no layout são obtidas sem que o código do mesmo seja editado. O redimensionamento do layout é feito pelo sistema com um mínimo de intervenção do projetista.

O sistema de parametrização é dividido em duas partes: uma, responsável pela parametrização a nível elétrico e, outra, pela parametrização a nível de geometria.

A especificação dos requisitos de ambas partes é feita neste trabalho, bem como dos dados necessários ao sistema de parametrização.

CONTRIBUIÇÃO PARA A ESPECIFICAÇÃO DE UM  
SISTEMA DE GERAÇÃO DE CÉLULAS PROGRAMÁVEIS  
PARA UTILIZAÇÃO EM PROJETO DE C/P'S

Rosana Ceron Di Giorgio

## Conteúdo

1	Introdução	3
2	Resenha Bibliográfica	7
3	Apresentação da Parametrização de células em diferentes níveis de profundidade	24
3.1	Parametrização visando a otimização do desempenho das células para um dado processo	26
3.2	Parametrização para mudança de processo tecnológico	27
4	Descrição do Sistema de Geração de Células Parametrizadas	30
4.1	Perspectivas do sistema	30
4.2	Funções do sistema	32
4.3	Características do usuário do sistema	35
4.4	Requisitos do sistema	36
4.4.1.	O parametrizador geométrico (PG)	36
4.4.1.1.	Requisitos do parametrizador geométrico (PG)	36
4.4.2.	O parametrizador elétrico (PE)	42
4.4.2.1.	Sistema de geração de modelos representados por tabelas	44
4.4.2.1.1	Especificação de comportamento e operação (B11)	52
4.4.2.1.2	Critérios de escolha de novos valores para os Parâmetros DIM e OPE (B12)	54
4.4.2.2.	Módulo de Mapeamento Vetorado	57
4.4.2.3.	Conclusões	61
5	Bibliografia	71
6	Apêndice	75

# 1 INTRODUÇÃO

A fim de abreviar o ciclo de projeto de CI's dedicados e semidedicados, são utilizadas metodologias de projeto que aceleram a produção do layout final e diminuem o custo, tais como as denominadas "standard-cells" e "gate-arrays". As células padrão utilizadas em ambos os casos possuem dimensões fixas, ou seja, seu layout é fixo para o processo em questão. Assim, com a evolução da tecnologia e a conseqüente mudança nas regras de projeto, regras elétricas e nos parâmetros de processo do fabricante, torna-se necessário frequentemente o reprojetado do circuito elétrico e do layout das células. Neste trabalho será estabelecido um conjunto de técnicas que visam a automatização do redimensionamento do layout e o reprojetado elétrico de células de CI's. Uma ferramenta importante para atingir essa meta é a parametrização do "lay-out", que consiste na programação do mesmo de modo a torná-lo dimensionalmente dependente de determinados parâmetros. A programação do layout pode ser obtida, por exemplo, através de sua codificação na forma de expressões, onde os parâmetros de interesse são variáveis independentes, cujos valores são alterados a critério do projetista. Desta forma, a geometria do layout passa a ser representada por uma codificação fixa de suas relações topológicas, mas com a métrica flexível, determinada por alterações nos valores numéricos dos parâmetros das expressões que compoem seu código. Estes parâmetros podem permanecer contidos em uma base de dados, a qual, uma vez alterada, implicará na alteração das dimensões do lay-out.

A descrição da topologia do lay-out por meio de uma linguagem conveniente, objetiva, também, que o mesmo seja correto por construção, traduzindo exatamente como cada nível de máscara se relaciona com os demais, obedecendo a todas as regras impostas pela tecnologia e pelo próprio projetista.

A parametrização permite, portanto, tornar as máscaras de CI's fácil e rapidamente ajustáveis em função de parâmetros geométricos e elétricos. Desta forma, por exemplo, uma dada célula de biblioteca pode, conservando a mesma topologia, ter sua métrica automaticamente modificada para satisfazer as regras de projeto de outro fabricante ou mesmo restrições elétricas,

desde que parâmetros geométricos e elétricos sejam definidos como variáveis das quais o layout é função. Do mesmo modo, uma dada célula pode, conservando as mesmas características elétricas, ter sua topologia modificada para adequar-se a uma dada área em silício.

No primeiro caso, é necessário que o projetista determine, durante a codificação do layout, quais segmentos do mesmo devem ser tomados como variáveis, isto é, quais influem nos parâmetros elétricos que se deseja otimizar numa dada célula e quais são aqueles que representam as restrições do processo de fabricação, isto é, as regras de projeto. Por exemplo, a largura da máscara de poly que se sobrepõe a uma difusão, constituindo o comprimento de canal do transistor, bem como a largura da mesma difusão, que representa a largura de canal do transistor, são variáveis que influem nas características elétricas dos dispositivos. Já a largura mínima de contato constitui uma regra de projeto do fabricante.

Se todas as coordenadas e segmentos de uma dada célula forem tomados como parâmetros, dependendo da complexidade da célula, o conjunto de expressões e o custo para estabelecê-lo poderão ser altos. Portanto, o projetista deverá definir, com base em critérios derivados do comportamento da célula e da flexibilidade desejada, qual o nível a ser atingido pela parametrização, isto é, deve definir um conjunto mínimo de dimensões a serem tomadas como parâmetros no equacionamento das células, sendo as demais mantidas fixas.

No segundo caso, como o desempenho da célula não deve ser alterado, as medidas dos segmentos do layout que o afetam devem ser mantidas, porém, a topologia do layout poderá ser modificada. Isto pode ser útil, por exemplo, quando a alteração da métrica para uma mesma topologia (caso 1) conduz a layouts que não são adequados para uma dada área em silício. Como exemplo, são mostradas na figura 1 duas alternativas de compactação para o layout de um transistor cuja largura foi incrementada de 3 vezes. Na alternativa 1 a topologia se mantém e, na segunda, é modificada.

A parametrização que visa o reprojetado rápido da geometria (topologia + métrica = layout) e o reprojetado elétrico (desempenho) de forma automatizada, apresenta as seguintes vantagens:

- permite otimizar a confiabilidade do projeto, pois elimina a intervenção humana e portanto a susceptibilidade a erros;

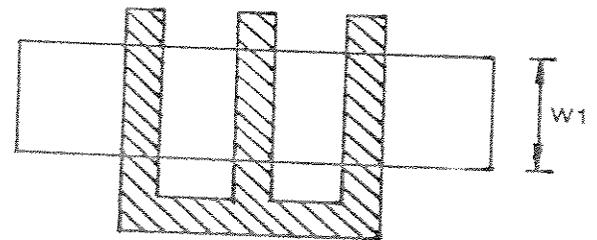
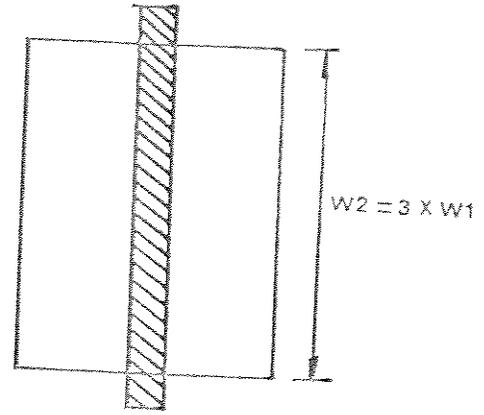
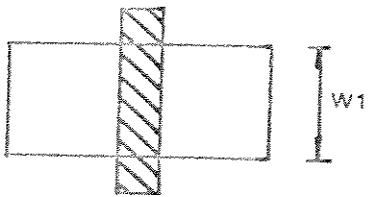


Figura 1: Duas Alternativas de Compactação Para Um Transistor

- dispensa o passo de verificação de regras de projeto (DRC), pois os layouts são corretos por construção. Além disso, os textos das células não são mais modificados após a validação, não havendo necessidade de um *check* a cada alteração do layout, que é feita sem intervenção humana;
- diminui o custo de reprojeto, visto que a tarefa de geração do novo layout é automática;
- permite o acompanhamento da evolução dos processos de fabricação, já que as regras de projeto do fabricante podem ser tomadas como parâmetros, tornando fácil o redimensionamento do layout à medida que as dimensões mínimas são reduzidas, ou quando é trocado o conjunto de regras de projeto de um fabricante por outro;
- permite alteração recursiva das características elétricas da célula, atendendo às especificações de cada projeto em particular, visando a otimização do desempenho.

A viabilidade da parametrização de uma célula, entretanto, depende da profundidade com que é feita, ou seja, do número máximo de variáveis ou parâmetros utilizados. Isto visa reduzir o número e a complexidade das expressões paramétricas envolvidas, além do esforço computacional e memória alocada para resolvê-las. Este ponto, que tem sido o principal gargalo dos programas propostos para este fim, deve ser analisado para a posterior avaliação do potencial requerido para o software cuja especificação é proposta neste trabalho e a consequente viabilidade de sua confecção.

O objetivo do trabalho que será apresentado a seguir, é fornecer a contribuição do projetista de circuitos integrados para o conjunto de requisitos de um sistema de geração de células parametrizadas, para uso em projeto de CI's.

Estas atividades tiveram início em 1985, no CTI (Centro Tecnológico Para Informática, Campinas - SP) onde, utilizando-se a linguagem LPG (20), foram feitos experimentos no sentido de parametrizar células de CI's em função das regras de projeto do fabricante e de dimensões dos dispositivos que exerciam influência no desempenho destas células.

A LPG foi criada para ser utilizada dentro de um sistema denominado SPA-D (38), que trata-se de um sistema de projeto hierarquizado que utilizava a parametrização de células.

## 2 RESENHA BIBLIOGRÁFICA

A parametrização não era quase abordada até 1985. Porém, a partir de então, a atenção sobre o assunto vem aumentando significativamente. Células parametrizadas, apresentadas na forma de rotinas, vêm sendo investigadas e linguagens adequadas à descrição de geometrias vêm sendo propostas.

As referências (1) a (7), (13) a (22), (26) e (31) a (34) apresentam inúmeras propostas, abordando a parametrização em vários níveis. São encontradas variações quanto às linguagens em que estes programas foram escritos, mas as mais comumente utilizadas têm sido PASCAL ou C. Alguns apresentam linguagem de entrada de alto nível, outros não, outros utilizam layout simbólico e há quem chegue a confeccionar um editor gráfico específico para este fim (1,26), uma vez que os já existentes são pacotes fechados, não sendo possível seu uso.

Alguns programas ainda utilizam regras  $\lambda$ , outros já trabalham diretamente com as regras de projeto da tecnologia.

Os trabalhos mais recentes incorporam o uso de primitivas, isto é, micro-blocos ou elementos de circuito, para agilizar o processo de parametrização e reduzir o esforço computacional.

Normalmente a interface com um simulador elétrico é encontrada, de modo a agilizar o reprojeto elétrico de células que são parametrizadas apenas quanto à geometria.

A Digital Equipment Corporation apresentou em em 1985 (6) células programáveis quanto à mudança de tecnologia, por meio de macros pré-definidas, ou seja, rotinas já prontas contendo o equacionamento das células em função das regras de projeto do fabricante. O programa utilizado (PFG - Parameterizable Function Generator) permite ao projetista fornecer os valores numéricos das variáveis das expressões e ainda sugere ao mesmo layouts alternativos, que conduzem a diferentes topologias.

Esta ferramenta entretanto, não permite ao projetista sintetizar novas macros. As rotinas correspondentes às células são elaboradas por especialistas (tendo em vista a complexidade do código requerido) e catalogadas na

máquina, onde permanecem à disposição do usuário. Na época, estavam disponíveis 40 macros, sendo as mais complexas super-buffers e drivers de E/S.

A mesma filosofia é seguida por outros com algumas variações. A LPG (20) por exemplo, permite ao usuário sintetizar suas próprias macros. Entretanto, apresenta as seguintes dificuldades:

- o usuário deve elaborar expressões algébricas muitas vezes de alta complexidade para descrever o layout, tornando o processo lento e difícil
- a depuração exaustiva do texto correspondente a um layout parametrizado segundo a LPG é difícil de ser realizada
- uma vez pronto o texto da figura, sua parametrização só é válida para o conjunto de parâmetros utilizado nas expressões, isto é, se o conjunto de restrições (parâmetros) aumentar, por exemplo devido à evoluções numa dada tecnologia, o texto deve ser refeito a fim de incorporá-las.

Outras linguagens poderosas(1,5,4,31 e 33) facilitam a criação de macros pelo usuário. Diferentemente das anteriores, não necessitam que o projetista crie expressões para descrever o layout, as quais são muitas vezes complexas, acarretando longo tempo de desenvolvimento e elevada incidência de erros. Em (4), o projetista edita seu layout por uma linguagem do tipo PASCAL, onde fornece apenas as posições relativas entre elementos já conhecidos pelo programa, ou seja, microblocos. As posições absolutas são calculadas pelo programa, levando em conta as restrições fornecidas pelo usuário. Os microblocos podem ser quaisquer células pré-definidas para o programa, tais como transistores, fitas de interconexão, contatos, etc, todos com dimensões variáveis, isto é, parametrizados.

É interessante observar que os programas que vêm sendo propostos ao longo dos anos vêm apresentando uma evolução significativa, visto que tornam-se cada vez mais eficientes e simples sob o ponto de vista do usuário. O usuário já não escreve expressões nem efetua cálculos, entrando apenas com uma descrição em alto nível, contendo os elementos que irão compor o layout e suas posições relativas.

A seguir será feita uma melhor análise, sob o ponto de vista de um projetista de circuitos integrados, das características das principais linguagens abordadas anteriormente.

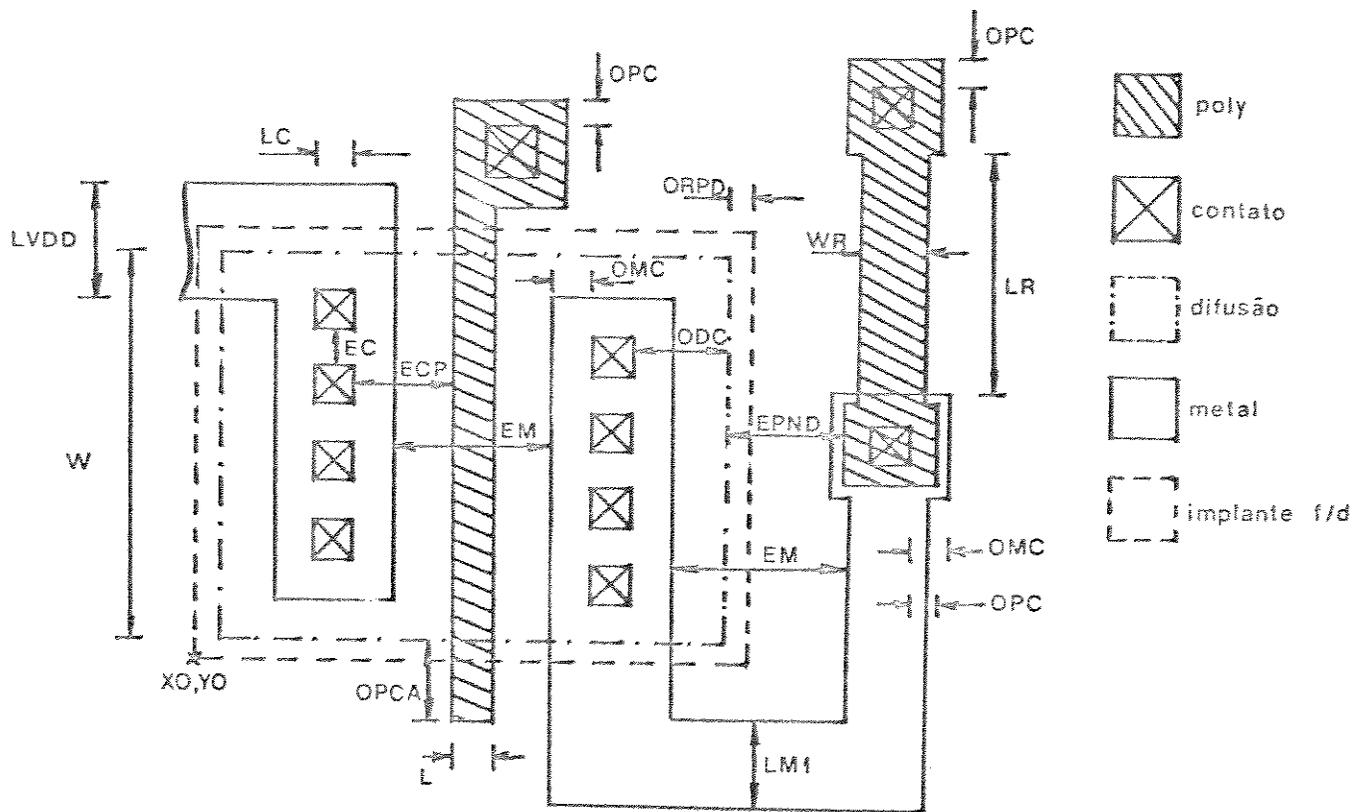


Figura 2: Apresentação de um *Layout* com Indicação das Dimensões e Espaçamentos Tomados como Parâmetros

Para isso, inicia nos com o exemplo de uma descrição manual, do texto de um *layout* parametrizado, para que ele se torne independente da tecnologia e possível de ser otimizado quanto a seu desempenho. Tomemos como exemplo o *layout* da figura 2.

Os espaçamentos indicados, com exceção de W, LVDD, LM1, WR, e LR, constituem o conjunto mínimo de parâmetros necessário para tornar o lay-out independente da tecnologia. Os parâmetros restantes são tomados como tal a critério do projetista, a fim de permitir a otimização da performance da célula. Todos estes parâmetros deverão estar contidos na BDG, onde o projetista terá acesso para atualização dos mesmos com a evolução dos processos de fabricação ou para otimização do desempenho da célula.

Os parâmetros indicados no exemplo da figura 2 significam:

1- LC: largura mínima de contato metal/difusão

- 2- EC: espaçamento mínimo entre contatos metal1/difusão
- 3- ECP: espaçamento mínimo entre contato e poly-gate
- 4- OMC: sobreposição mínima entre metal e contato
- 5- OPC: sobreposição mínima entre poly e contato
- 6- ODC: sobreposição mínima entre difusão e contato
- 7- OPCA: sobreposição mínima do poly-gate sobre o campo
- 8- ORPD: sobreposição mínima entre região P+ e difusão
- 9- EM: espaçamento mínimo entre metal1/metal1
- 10- EPND: espaçamento mínimo entre poly sobre o campo e difusão
- 11- W: largura de canal do transistor
- 12- L: comprimento de canal de transistor
- 13- WR: largura do resistor de poly-Si
- 14- LR: comprimento do resistor de poly-Si
- 15- LVDD: largura da linha de alimentação (Vdd)
- 16- LM1: largura da linha de dreno em metal 1

Os parâmetros 1 a 10 são dimensões mínimas especificadas pelo fabricante, ou seja, as regras de projeto de uma dada tecnologia. As variáveis 11 a 16 são utilizadas na otimização da performance desta célula, da seguinte forma:

- W e L são usados para permitir aumento da capacidade de corrente e portanto da velocidade de chaveamento do transistor.
- WR e LR são usados para permitir a variação da resistência do resistor de poly-Si.
- LM1 e LVDD são tomados como variáveis para que possam ser redimensionados de acordo com as correntes através do resistor e transistor, respectivamente.

Todos estes parâmetros devem estar contidos em uma base de dados, podendo ser atualizados diretamente pelo projetista. Os últimos (12 a 16), por estarem relacionados com o desempenho elétrico do circuito, podem ser calculados pelo P.E., em função dos parâmetros elétricos de que dependem.

A seguir, será mostrado um exemplo de como podem ser feitas as expressões que irão descrever os níveis de máscara de uma figura, de modo que todas as

restrições que regem as interações entre eles sejam respeitadas e também as regras (dimensões) impostas pelo projetista.

Alguns experimentos foram feitos com a linguagem denominada LPG (ref. 20), que é uma linguagem de procedimentos. Segundo a LPG, as interações entre os diferentes níveis são indicadas por meio de expressões algébricas, da forma apresentada abaixo. Estas expressões são elaboradas pelo projetista, tendo em mente as restrições da tecnologia e a dependência entre as dimensões do lay-out e o desempenho da célula.

Para o lay-out da figura 2, o equacionamento segundo a LPG deve ser efetuado da seguinte forma para as coordenadas  $x_1$ ,  $x_2$ ,  $x_3$  e  $y_1$ :

$$\begin{aligned}x_1 &= X_0 + ORPD + ODC + LC + 2 * ECP + L - OMC \\y_1 &= Y_0 + OPCA + W - ODC + OMC \\x_2 &= X_1 + LC + ODC + EPND + OPC \\x_3 &= X_1 + 2 * (OMC - ECP) - L\end{aligned}$$

A figura 3 indica as coordenadas  $x_1$ ,  $x_2$ ,  $x_3$  e  $y_1$  no *layout* da figura 2.

Observa-se que  $x_1, x_2$  e  $x_3$  são somatórias em  $x$  e  $y_1$ , somatória em  $y$ , dos deslocamentos indicados na figura 2. Os segmentos utilizados não são escolhidos aleatoriamente. Óbvio, o projetista deve fazer manipulações algébricas de modo a utilizar nas expressões apenas aqueles espaçamentos que representam parâmetros da célula, isto é, que fazem parte do conjunto mínimo. Por exemplo, a distância entre metal e poly-gate não depende do processo e, portanto, não é incluída na equação para  $x_1$ . Ao contrário, utiliza-se o espaçamento entre contato e poly-gate, que é o parâmetro que sofre alteração com a evolução da tecnologia, sendo por isso uma regra de projeto do fabricante e uma variável que pertence à BDG.

Pode-se observar que nas expressões para  $x_1$ ,  $x_2$  e  $x_3$ , o parâmetro  $EM$  não aparece e, portanto, não existe nada que garanta que as seguintes restrições :

$$x_1 - x_3 \geq EM \text{ e } x_2 - x_4 \geq EM,$$

indicadas na figura, sejam respeitadas.

Conclui-se que as expressões anteriores não são suficientes para garantir as várias condições simultâneas, necessárias para a determinação das coordenadas  $x_1$  a  $x_4$  sem a ocorrência de violação das restrições estabelecidas.

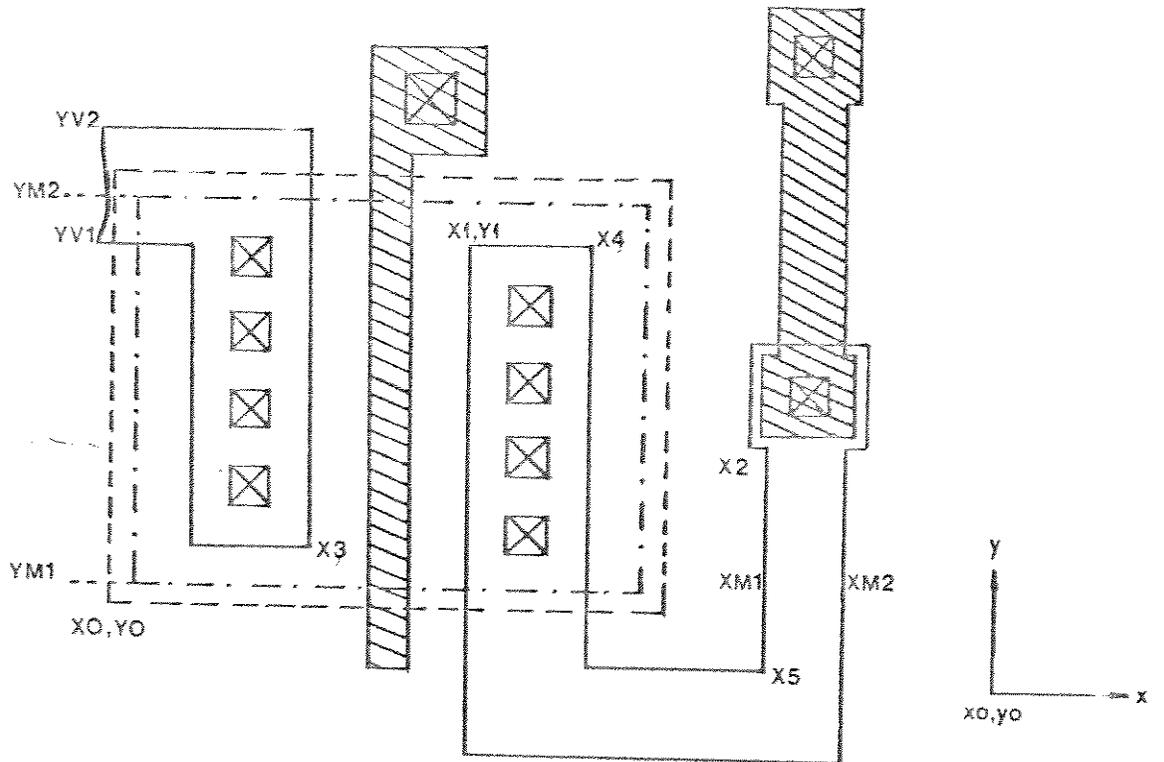


Figura 3: Indicação das Coordenadas  $x_1$ ,  $x_2$ ,  $x_3$  e  $y_1$  no *Layout* Apresentado

Portanto, uma forma de possibilitar o uso da LPG para a parametrização geométrica, seria acrescentar a facilidade de composição e resolução de inequações que *amarrem* todas as restrições, isto é, que incluam todas as relações, todas as dependências entre os parâmetros. Estes sistemas de inequações iriam compor o texto da célula, isto é, o novo código LPG.

Assim, para a determinação das coordenadas  $x_1$  a  $x_5$ , teríamos as seguintes equações e inequações:

$$\begin{aligned} x_1 &= x_0 + ORPD + ODC + LC + 2 * ECP + L - OMC \\ x_3 &= x_1 + 2 * (OMC - ECP) - L \\ x_1 - x_3 &\geq EM \quad (1) \end{aligned}$$

$$\begin{aligned} x_2 &= x_1 + LC + ODC + EPND + OPC \\ x_4 &= x_1 + 2 * OMC + LC \\ x_2 - x_4 &\geq EM \quad (2) \\ x_5 - x_4 &\geq EM \quad (3) \end{aligned}$$

$$x_5 = (x_2 + 2 * OMC + LC - x_2) / 2 - (LM1 / 2) + x_2 = [(2 * OMC + LC - x_2) - LM1] / 2 + x_2$$

As desigualdades (1), (2) e (3) devem ser satisfeitas de modo a incrementar sempre a maior das duas coordenadas e não decrescer a menor. Por exemplo, se  $x_1 - x_3 < EM$ , então  $x_1$  deve ser incrementado e não  $x_3$  decrescido. Desta forma, nunca existirão coordenadas negativas e cada célula poderá estar sempre delimitada por uma envoltória com origem em  $(x_0, y_0)$ , facilitando sua utilização por programas de alocação/roteamento.

A descrição dos retângulos de um *layout* é feita na LPG em um corpo do programa denominado PROCEDIMENTO. Por exemplo, segundo a LPG a descrição do poly-gate do lay-out anterior fica:

FIGURA POLYGATE:= (XP,Y0; 0, CP, WP, -WP, -WP1, -CP2), onde  
 POLYGATE = nome da figura  
 XP,Y0 = origem da figura  
 0, WP, -WP1 = deslocamentos em x  
 CP, -WP, -CP2 = deslocamentos em y

Tanto os deslocamentos como a origem são determinados por expressões paramétricas do tipo das mostradas anteriormente.

Este corpo do programa permite ao usuário definir a topologia do polygate, cuja métrica torna-se função dos parâmetros CP, WP, WP1 e CP2, e a posição, função de XP e Y0.

Nos experimentos realizados com a LPG, o grid foi tomado como parâmetro e nos lay-outs foi imposto, através do equacionamento, a localização de todos os pontos de contato com o meio externo em múltiplos de unidade de grid. As células foram codificadas de modo a possuir também suas dimensões externas em x e y em múltiplos de unidade de reticulado, para qualquer valor do mesmo. Assim, tanto as posições dos terminais, como a altura e largura das células, variavam segundo alterações no valor do grid, no arquivo de dados.

O problema das linguagens de procedimentos, como esta, é que a codificação do lay-out torna-se uma tarefa lenta e trabalhosa, além de ser difícil como provar que o *lay-out* resultante é livre de violações de regras de projeto para qualquer conjunto de parâmetros.

Os experimentos feitos com a LPG auxiliaram a concluir sobre a viabilidade da geração manual de expressões paramétricas, que apresenta os seguintes problemas:

- é um trabalho complexo tendo em vista o elevado número de expressões e variáveis requeridas na descrição de uma figura parametrizada
- torna o processo altamente susceptível a erros e portanto de baixa confiabilidade
- o tempo gasto na descrição do lay-out é excessivo
- a depuração do código correspondente à cada célula é extremamente lenta

ALI (20) e ALI2 (31,32) são linguagens de procedimentos dedicadas à descrição de lay-out com restrições. Para estes programas, o lay-out também é visto como um conjunto de retângulos com relações entre eles. Cada retângulo

deve possuir seus lados orientados na direção dos eixos do sistema de coordenadas cartesianas.

O usuário de ALI é quem declara os retângulos e suas interrelações. O programa então gera o lay-out com a área mínima que satisfaça as restrições especificadas pelo usuário.

Diferentemente da LPG, as restrições em ALI são representadas somente por inequações lineares, da forma:

$$x_i - x_j \geq d, (d > 0) \text{ ou}$$

$$x_i = x_j$$

onde,

$x_i, x_j$  são coordenadas do lay-out e

$d$  é um parâmetro do lay-out.

Estas desigualdades não podem envolver coordenadas  $x$  e  $y$  simultaneamente (da mesma forma que na LPG) e devem ser resolvidas para determinar as dimensões e posições dos elementos de lay-out.

O fato das restrições serem representadas da forma mostrada acima, gera problemas quando o usuário requer dimensões fixas, ou dimensões que não sejam as mínimas. Por exemplo, no lay-out da figura 3 temos as seguintes igualdades:

$$YM2 - YM1 = W$$

$$XM2 - XM1 = LM1$$

$$YV2 - YV1 = LVDD$$

Estas expressões não são resolvidas em ALI e ALI2. Portanto, estas linguagens se prestam somente aos casos em que são requeridas áreas mínimas, pois apenas dão soluções para mínimas dimensões e mínimos espaçamentos.

A diferença básica entre ALI e ALI2, é que a segunda implantou o uso de *system cells*, ou seja, microblocos, na tentativa de reduzir a descrição das

células, que eram muito longas quando efetuadas em termos de retângulos. Estes microblocos consistem em elementos de células tais como contatos, transistores , etc, com dimensões programáveis.

Assim, o lay-out passava a ser descrito hierárquicamente, porém, ao nível de restrições não havia hierarquia e para lay-outs grandes o número de restrições era elevado a ponto de ocasionar *thrashing* de memória.

Um ponto positivo comum a ambas é a linguagem de entrada do usuário. Esta é uma linguagem de alto nível, apresentando certa familiaridade com o PASCAL e os recursos do mesmo. Isto torna muito mais fácil para o usuário descrever o lay-out do que anteriormente, gerando as restrições manualmente, ou da forma mostrada acima, ou por meio de expressões do tipo da LPG. Estas restrições são agora geradas automaticamente.

A figura 4 mostra uma descrição em ALI e o lay-out correspondente.

Conforme mostrado no exemplo da figura anterior, os objetos manipulados em ALI são declarados pelos *nomes e tipos*. Os tipos apresentam a mesma estrutura dos tipos em PASCAL e os objetos podem ser de *tipo simples* ( boxes ) ou de *tipo estruturado*.

Existem tipos denominados *standard*, todos simples, que correspondem às camadas do processo ( ALI foi implementada para um processo nMOS de 6 níveis: metal, poly, difusão, implante, contato e passivação), além do tipo virtual, usado para dar nome a *bounding boxes* e não tendo significado físico. Por exemplo, no programa da fig. anterior, a declaração

vertical : metal

especifica que o retângulo denominado vertical deve fazer parte da camada de metal no lay-out final. ALI usará esta informação para gerar restrições correspondentes a seu tamanho mínimo e seu espaçamento em relação aos elementos de lay-out.

Os tipos estruturados são:

- array (uma coleção de objetos de mesmo tipo) e
- bus (uma coleção de objetos de tipos heterogêneos), que correspondem di-

```

chip simple,
  const
    hnumber = 10;
    length  = 20;
    width   = 6;
  foctype
    htype : array [ 1.. hnumber] of metal;
  var
    i : integer
  box
    horizontal : htype
    vertical   : metal
  begin
    for i= 1 to hnumber - 1 do begin
      above (horizontal [i], horizontal [i+1];
      glueright (horizontal [i], vertical);
      xmore (horizontal [i], length)

    end;
    glueright (horizontal [hnumber], vertical);
    xmore (horizontal [hnumber], length);
    xmore (vertical, width)
  end.

```

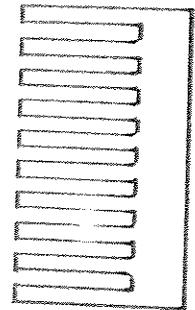


Figura 4: Uma Descrição ALI e o *Layout* Correspondente

retamente aos tipos estruturados *array* e *record* do PASCAL.

Como no PASCAL, ALI permite a criação de tipos definidos pelo usuário, que podem ser simples ou estruturados. Por exemplo, na figura 4, o fragmento

```
htype : array [1..hnumber] of metal,
```

dentro da seção *boxtype* do programa, cria um novo tipo, *htype*, sendo cada objeto deste tipo feito de um número de retângulos de metal, e o fragmento

```
horizontal : htype
```

dentro da seção *box*, cria um objeto deste tipo chamado *horizontal*.

Outra linguagem voltada para o mesmo fim, ALLENDE (4), surgiu da evolução de ALI, ALI2 e CLAY (33).

Com a introdução de restrições do tipo

$$x_i - x_j = e,$$

ALLENDE permite o uso de dimensões fixas, ou seja, não mínimas.

Além disso, ALLENDE possibilita hierarquia ao nível de restrições, resolvendo o problema do elevado número de restrições verificado em lay-outs grandes.

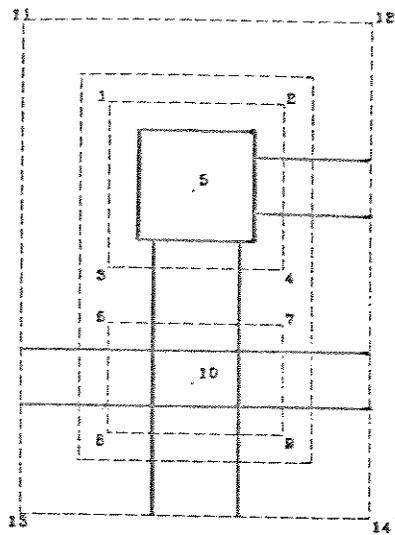
A figura 5 apresenta um lay-out e sua representação a nível de restrições em ALLENDE.

A forma de descrever os retângulos assemelha-se a utilizada na LPG. O contato, por exemplo, é descrito na figura acima como:

```
box cut x5 y5  $\lambda$   $\lambda$   $\lambda$   $\lambda$ ,
```

que representa um retângulo pertencente à camada de contato, com coordenadas  $x_5 - \lambda$ ,  $x_5 + \lambda$ ,  $y_5 - \lambda$ ,  $y_5 + \lambda$ , sendo  $x_5$ ,  $y_5$  sua coordenada central.

É importante observar que ALLENDE foi implementada com regras  $\lambda$ , e que tudo no lay-out varia em função de  $\lambda$ . No nosso caso, queremos que as



$$x_5 = x_{10}$$

$$x_3 = x_6$$

$$x_4 = x_7$$

$$x_5 - x_1 \geq 4$$

$$y_1 - y_5 \geq 4$$

$$x_2 - x_5 \geq 4$$

$$y_5 - y_3 \geq 4$$

$$x_{10} - x_6 \geq 3$$

$$y_6 - y_{10} \geq 2$$

$$x_7 - x_{10} \geq 3$$

$$y_{10} - y_5 \geq 2$$

$$x_1 - x_{11} \geq 2$$

$$y_{11} - y_1 \geq 2$$

$$x_{12} - x_2 \geq 2$$

$$y_6 - y_{13} \geq 2$$

$$y_5 - y_8 \geq 2$$

box metal  $x_5$   $y_5$   $2\lambda$   $2\lambda$   $2\lambda$   $2\lambda$

box diff  $x_5$   $y_5$   $2\lambda$   $2\lambda$   $2\lambda$   $2\lambda$

box cut  $x_5$   $y_5$   $\lambda$   $\lambda$   $\lambda$   $\lambda$

wire metal  $3\lambda$   $y_{13}$   $y_5$   $x_{10}$   $0$   $-2\lambda$

wire poly  $2\lambda$   $x_{11}$   $x_{12}$   $y_{10}$   $0$   $0$

wire diff  $2\lambda$   $x_5$   $x_{12}$   $y_5$   $-2\lambda$   $0$

Figura 5: Um Layout a Nível de Restrições em ALLENDE

dimensões variem em função das regras de projeto da tecnologia para cada nível, e demais parâmetros definidos pelo projetista. Por exemplo, queremos que um contato com dimensões mínimas tenha seus lados expressos em função de LC ( mínima dimensão de contatos metal1/difusão) e não  $\lambda$ .

Em ALLENDE, o usuário descreve o lay-out hierárquicamente por intermédio de uma linguagem de alto nível, do tipo PASCAL A ou C, como uma composição de células. O layout não é mais um conjunto de retângulos como na LPG e ALI e sim, uma composição de microblocos de células.

O usuário fornece as posições relativas entre estes microblocos e o programa realiza a compactação máxima do lay-out em todas as direções, sem violar as regras de projeto e demais restrições, conhecidas pelo mesmo, fornecidas externamente pelo projetista. Estes parâmetros podem ser alterados pelo usuário, cabendo ao programa atualizar os arquivos finais correspondentes às células.

O usuário diz, por exemplo, na concepção de uma porta NAND de 2 entradas, que deseja compor uma célula constituída por 4 transistores, 2 do tipo *pull-up* e 2 do tipo *pull-down*, catalogados como microblocos para o programa.

Os *pull-up's* devem ficar acima dos *pull-down's*, a linha de Vdd deve localizar-se no topo da célula e ligar-se às fontes dos *pull-up's*, a linha de Vss deve localizar-se no extremo inferior da célula e ligar-se ao dreno do *pull-down* inferior. A figura 6 mostra o arquivo ALLENDE de entrada para a porta NAND de 2 entradas em tecnologia nMOS, parametrizada.

A figura 7 mostra outro programa de entrada, do usuário, que após ser transformado para uma linguagem intermediária, gera as restrições da figura 5.

Pontos importantes verificados em ALLENDE:

1. Um programa ALLENDE é descrito em termos de variáveis que recebem valores de um arquivo de externo (como a LPG), de modo que, no texto correspondente ao lay-out, somente os valores variem (à medida que a tecnologia evolui ou se requer alterações de desempenho), enquanto que o texto se mantém o mesmo.
2. O programa muitas vezes conduz a lay-outs que não satisfazem o projetista, e este não tem como interagir. É muito importante que o projetista possa atuar interativamente. No exemplo da porta NAND, o programa

```

program nand(output);
const
#include "/va/allende/usr/const.h"
type
#include "/va/allende/usr/type.h"
var power,ground,p2,d2: wiretype;
#include "/va/allende/usr/proc.h"

procedure contact(w1,w2,w3,w4: wiretype);
begin syscell(CONTACT,w1,w2,w3,w4,0); end;

procedure crossing(w1,w2: wiretype);
begin syscell(CROSSING,w1,w2,w1,w2,0); end;

procedure above;
begin place(ABOVE); end;

procedure nand;
begin
  begincell('nand');
  begincell('column1');
  syscell(LINE,power,nowire,power,nowire,0); above;
  contact(nowire,nowire,nowire,p2,p2); above;
  crossing(ground,p2);
  endcell(' ');
  place(LEFT);
  begincell('column2');
  contact(power,nowire,power,d2); above;
  syscell(PULLUP,nowire,d2,p2,d2,4); above;
  syscell(TRANSISTOR,nowire,d2,p2,d2,0); above;
  syscell(TRANSISTOR,p2,d2,nowire,d2,0); above;
  contact(ground,d2,ground,nowire);
  endcell(' ');
  place(LEFT);
  begincell('column3');
  crossing(power,p2); above;
  contact(p2,p2,nowire,nowire); above;
  contact(p2,nowire,nowire,p2); above;
  crossing(ground,p2);
  endcell(' ');
  endcell(' ');
end;

begin
power:= wire(METAL,5);
ground:= wire(METAL,5);
p2:= wire(POLY,2);
d2:= wire(DIFF,2);
nand;
end.

```

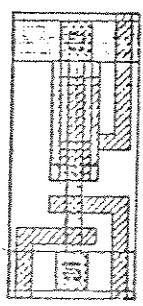
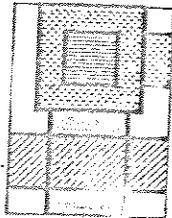


Figura 6: Arquivo ALLENDE de Entrada para a Porta NAND  
22



```
#include "/va/allende/usr/def.h"
main()
{
  begincell ( "example" );
  syscell ( CONTACT, nowire, nowire, diff(2), metal(3), # );
  place ( ABOVE );
  syscell ( CROSSING, poly(2), metal(3), poly(2), metal(3), # );
  endcell ( " " );
}
```

Figura 7: Programa ALLENDE de Entrada para o *Layout* da figura 5

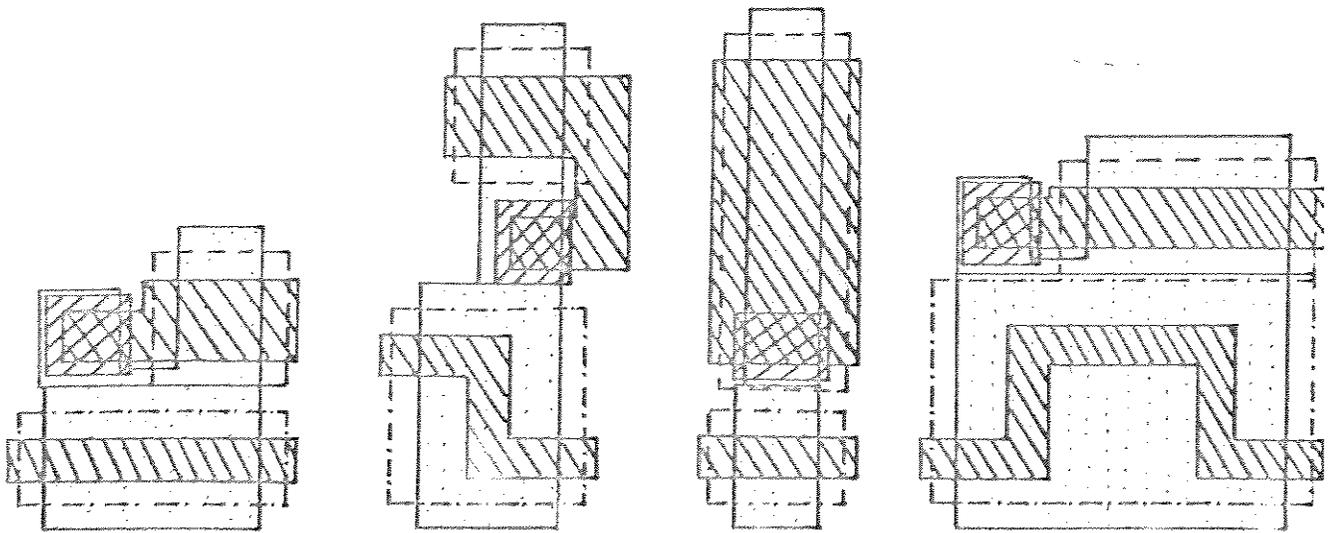


Figura 8: Exemplo de 4 Compactações Diferentes para um Inversor nMOS

irá conduzir as linhas de entrada e saída para qualquer um dos extremos da célula, independente da vontade do projetista.

3. Este programa não tem a flexibilidade de dar a um lay-out compactações diferentes, característica que pode ser implementada com o uso de micro-blocos, recurso que o programa já tem, ou inteligência artificial.

(6) apresenta um gerador de funções parametrizáveis, PFG, que incorpora esta capacidade. Na figura 8, temos o exemplo de 4 lay-outs alternativos obtidos do PFG, para um inversor nMOS.

As células utilizadas num programa ALLENDE são as *system cells* ou microblocos já mencionados anteriormente, quando falamos de ALI2.

ALLENDE utiliza dez tipos de microblocos: CONTACT, ICONTACT, TRANSISTOR, ITRANSISTOR, PULLUP, CROSSING, LINE, JOG, EMF e NULLCELL.

A parametrização, conforme vem sendo abordada, não será o principal alvo deste trabalho.

Alguns dos trabalhos existentes na área, prevêm no futuro um enfoque diferente para a parametrização. Trata-se do que será chamado neste trabalho

de parametrização elétrica, a qual visa a automatização do reprojeto elétrico das células. Este ponto, pouco abordado até então, será visto com ênfase na seção 4.4.2.

### 3 APRESENTAÇÃO DA PARAMETRIZAÇÃO DE CÉLULAS EM DIFERENTES NÍVEIS DE PROFUNDIDADE

Uma célula é função de 6 conjuntos de parâmetros:

- OPE - é o conjunto de parâmetros operacionais, ou seja, tensões de alimentação, temperatura de operação, cargas, etc.
- CMP - é o conjunto de parâmetros comportamentais, resultantes das especificações de uso da célula, como tempos de subida e descida, tempos de propagação, máxima frequência de operação, potência dissipada, margens de ruído, ganho, impedância de E/S, etc.
- DIM - é o conjunto de parâmetros dimensionais, ou seja, dimensões do *layout*, tais como largura e comprimento de canal de um transistor MOS, larguras de linhas de alimentação e terra, etc.
- ELE - é o conjunto de parâmetros elétricos dependentes do processo tecnológico, tais como resistividades de camadas do processo, máxima densidade de corrente permitida nas linhas, etc.
- RPS - é o conjunto de regras de projeto do fabricante, ou seja, regras de desenho, que devem ser seguidas na elaboração do *layout* da célula. São, por exemplo, mínimas dimensões e espaçamentos permitidos para as camadas do processo.
- PPR - é o conjunto de parâmetros do processo do fabricante, como por exemplo a espessura do óxido de porta, dopagem do substrato, difusão lateral sob o gate, etc.

Dependendo do nível de profundidade a ser atingido pela parametrização, o projetista irá determinar o conjunto de parâmetros dos quais o *layout* será função. Este conjunto deverá ser sempre o menor possível, de modo a:

- reduzir a complexidade das expressões paramétricas e, portanto, o esforço computacional necessário à geração e solução destas expressões;

- reduzir o tamanho dos arquivos contendo as descrições paramétricas e, portanto, a memória alocada ao seu armazenamento;
- tornar mais simples e mais rápida a elaboração da descrição inicial da célula; tanto quando textual como quando na forma gráfica, visto que, em ambos os casos, o projetista deve indicar todas as dimensões da célula, uma a uma, que serão tomadas como parâmetros DIM.

Os itens acima acarretam a redução no custo de projeto de uma célula parametrizada.

A definição do conjunto mínimo pode ser feita levando-se em conta a experiência anterior do projetista, ou algébricamente, ou ainda, por meio de um processo de simulações interativas que permitam estabelecer uma descrição funcional da dependência de um dado parâmetro com relação aos demais.

Quanto maior for o número de dimensões mantidas variáveis no *layout*, maior a profundidade da parametrização e maior a flexibilidade em se alterar o *layout* da célula. Maior, também, será o custo inicial de engenharia da célula.

Para examinar a importância da definição da profundidade da parametrização no custo da engenharia envolvida, podemos examinar dois casos de interesse na engenharia de células:

- parametrização visando a otimização de desempenho da célula, para um processo supostamente estável ou muito pouco variável;
- parametrização para acompanhamento das evoluções do processo tecnológico.

Para cada caso temos um conjunto mínimo diferente de parâmetros.

### 3.1 PARAMETRIZAÇÃO VISANDO A OTIMIZAÇÃO DO DESEMPENHO DAS CÉLULAS PARA UM DADO PROCESSO

Os parâmetros de desempenho de uma célula são definidos em sua especificação de uso, e constituem o conjunto de parâmetros CMP mencionados anteriormente.

Na parametrização para otimização do desempenho, o projetista deverá estudar a sensibilidade dos parâmetros CMP, para os quais deseja otimizar a célula, em função dos demais.

Esses parâmetros CMP são dependentes das características do processo de fabricação (parâmetros PPR), das dimensões dos dispositivos (parâmetros DIM) e das especificações de operação (parâmetros OPE). Como os parâmetros do processo de fabricação são constantes, as dimensões dos dispositivos e as especificações de operação tornam-se importantes para a otimização de desempenho.

Assim, neste nível de profundidade da parametrização, as variáveis das expressões que descrevem o *layout* deverão ser o conjunto mínimo de parâmetros DIM que influem no parâmetro de desempenho (CMP), para o qual a célula está sendo otimizada. A título de exemplo, no apêndice é feito um levantamento dos parâmetros elétricos (CMP) dependentes da geometria no transistor MOS (8,9,10).

Os parâmetros OPE, por não serem parâmetros de geometria, não podem ser incluídos no texto descritivo do *layout*. Entretanto, será visto adiante neste trabalho, que estes parâmetros são utilizados na otimização do desempenho da célula durante o processo denominado parametrização elétrica. Através da parametrização elétrica, os parâmetros OPE e CMP são levados em conta no cálculo dos parâmetros DIM utilizados nas expressões do *layout*.

### 3.2 PARAMETRIZAÇÃO PARA MUDANÇA DE PROCESSO TECNOLÓGICO

Na parametrização para mudança de tecnologia a profundidade a ser atingida é muito maior, pois deverá levar em conta todas as restrições dependentes da tecnologia. Uma alteração no layout neste nível, pode tornar necessário revalidar o desempenho das células, implicando em novo ciclo de otimização, conforme descrito no item 3.1. Portanto, a parametrização tendo em vista a mudança de tecnologia exige a definição de um conjunto de parâmetros envolvendo:

- todas as regras de projeto e regras elétricas do fabricante
- todos os parâmetros do processo
- todos os parâmetros empregados na etapa anterior (item 3.1).

Sómente os parâmetros RPS e DIM deverão fazer parte do texto descritivo do *layout*. Os demais são levados em conta durante a parametrização elétrica, a qual será vista posteriormente.

A parametrização neste nível sugere a organização apresentada na figura 9 para as variáveis mencionadas acima e as ferramentas de *software*, integradas a um sistema de CAD:

Com poucas alterações, esta foi a estrutura apresentada na referência (7).

Um simulador estatístico pode ser usado para se prever o desempenho do circuito tendo em vista a dispersão dos parâmetros do processo. Isto também pode ser feito de forma mais simplificada, utilizando-se nas simulações SPICE os parâmetros *slow* (pior caso de temperatura, tensão de alimentação e parâmetros de processo) e *fast* (melhor caso de temperatura, tensão de alimentação e parâmetros de processo), que são os casos *pior* e *melhor*, respectivamente, no que diz respeito à velocidade dos dispositivos.

O arquivo A contém a representação em coordenadas absolutas das células dimensionadas para uma dada tecnologia, isto é, corresponde aos arquivos finais, na forma fixa, utilizados pelo montador do layout para gerar a máscara final e pelo extrator.

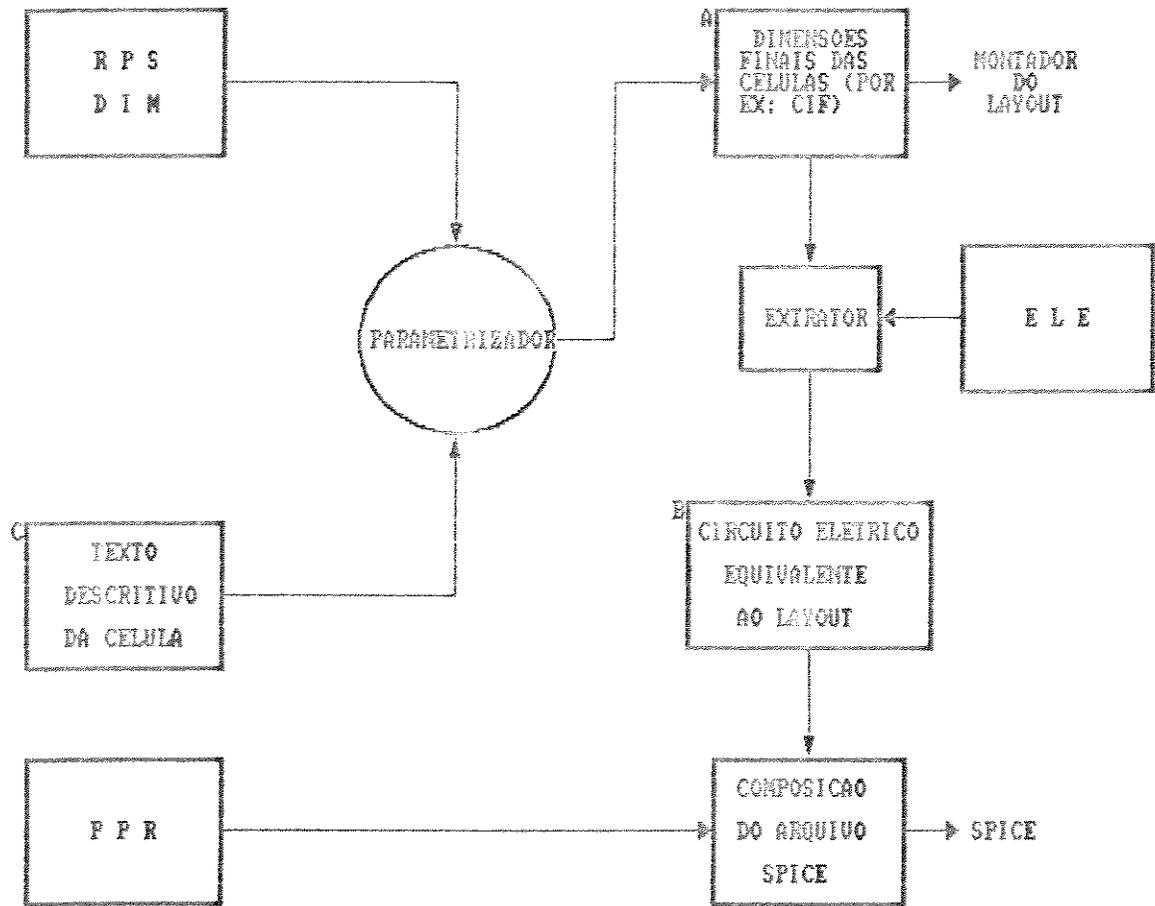


Figura 9: Organização Das Variáveis E Das Ferramentas De *Software* Em Um Sistema De CAD Na Parametrização para Mudança de Processo Tecnológico

O arquivo B contém o circuito elétrico equivalente ao *layout*, gerado pelo extrator.

O arquivo C contém os códigos correspondentes às células parametrizadas.

Os dados do arquivo B são, juntamente com os parâmetros de processo, utilizados na composição do arquivo de entrada do simulador elétrico, por exemplo o SPICE.

Quando ocorrem alterações no processo da tecnologia escolhida, o projetista, utilizando o sistema da figura anterior, pode encontrar um novo conjunto de dimensões para cada célula, de modo a otimizá-las para o novo processo.

A otimização da célula, conforme mostrada na figura 9, é feita interativamente, através da repetição de vários ciclos de simulações elétricas e de redimensionamento do *layout*. No item 4.4.2 deste trabalho, será apresentado um procedimento para automatizar a otimização do desempenho da célula.

A parametrização, na profundidade apresentada acima, presta-se também como ferramenta para apoiar a otimização do próprio processo visando obter melhor desempenho.

## 4 DESCRIÇÃO DO SISTEMA DE GERAÇÃO DE CÉLULAS PARAMETRIZADAS

### 4.1 PERSPECTIVAS DO SISTEMA

O sistema aqui proposto destina-se a ser integrado a um sistema de CAD para circuitos integrados, com a finalidade de auxiliar o projetista nas seguintes tarefas:

1. Gerar a descrição paramétrica de células de circuitos integrados em função dos parâmetros dimensionais escolhidos pelo usuário;
2. Estabelecer modelos comportamentais para as células, que descrevam relações entre os parâmetros comportamentais, dimensionais e operacionais;
3. Alterar de modo automático o *layout* de células já existentes, possibilitando assim:
  - a otimização de células quanto ao desempenho
  - o acompanhamento da evolução dos processos de fabricação
  - a adaptação do *layout* a uma dada área em silício

com base na descrição paramétrica e nos modelos comportamentais definidos para as mesmas;

4. Diminuir o custo do reprojeto de células, através da:
  - eliminação do ciclo de validação (DRC), visto que os *layouts* gerados são corretos por construção
  - automatização tanto da geração do *layout* como do reprojeto elétrico das células, tornando-os mais rápidos

O diagrama em blocos da figura 10 apresenta o sistema e as ferramentas de *software* e *hardware* com as quais se relaciona, dentro de um sistema de CAD para CI's:

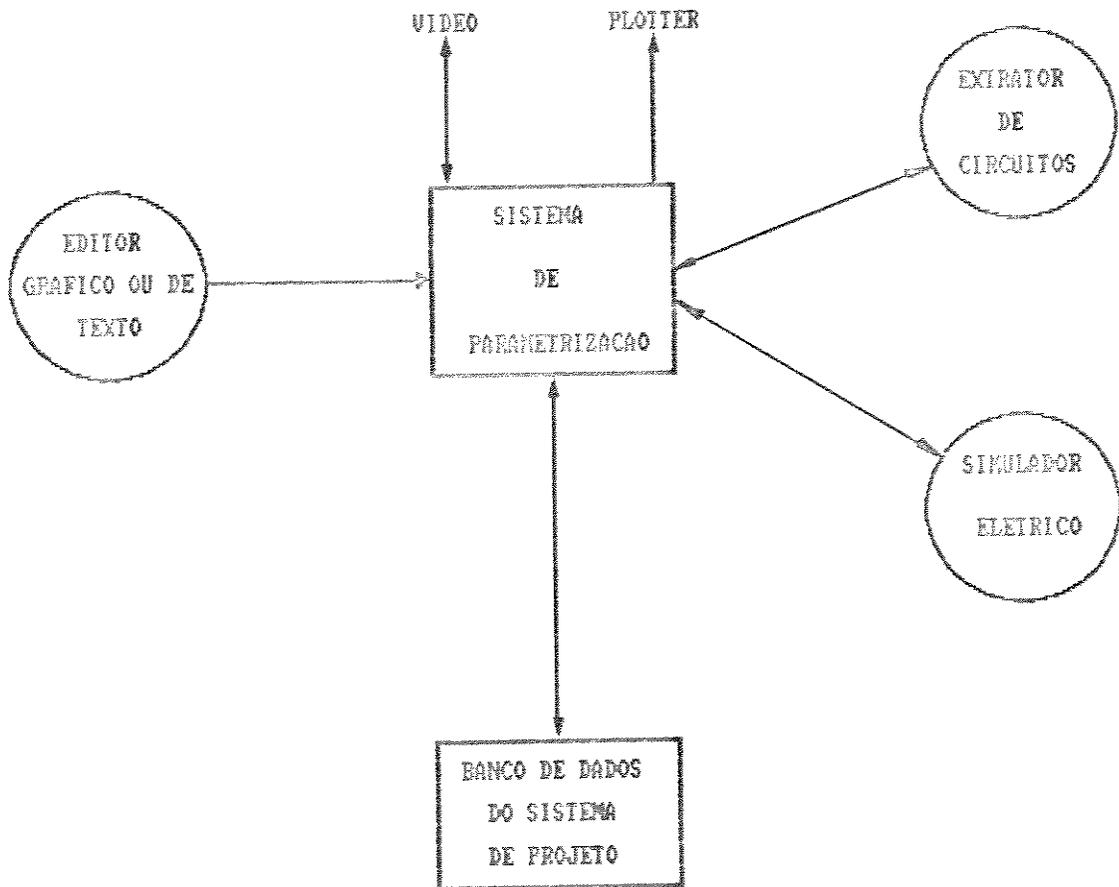


Figura 10: Apresentação da Relação do Sistema com as Ferramentas de *Software* e *Hardware*

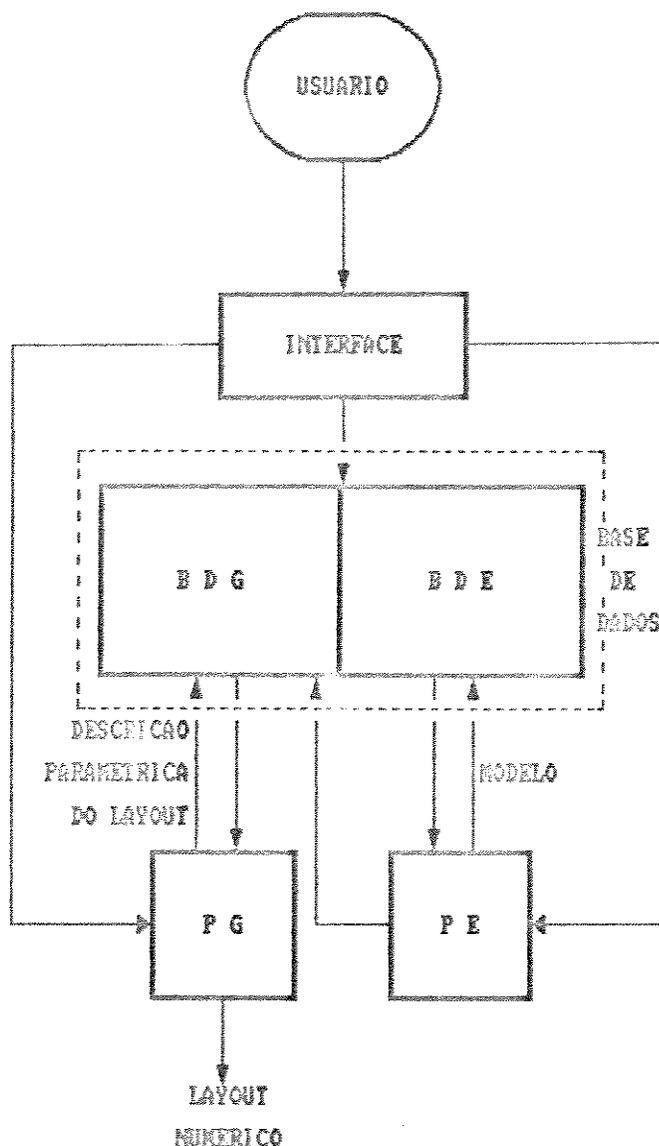


Figura 11: Diagrama em Blocos do Sistema

## 4.2 FUNÇÕES DO SISTEMA

O sistema é composto pelos módulos funcionais descritos a seguir e representado pelo diagrama em blocos da figura 11.

Como módulo entende-se uma unidade do sistema capaz de executar uma função bem definida.

- MÓDULO PG (Parametrizador Geométrico)

O módulo PG é um compilador da geometria da célula e tem as seguintes funções:

- Dado um *layout* (gerado via entrada gráfica ou textual), o PG deve gerar a descrição paramétrica correspondente. O arquivo resultante, contendo o *layout* na forma algébrica, é armazenado na base de dados associada ao PG (BDG).
- Dada a descrição do *layout* de uma célula por meio de um conjunto de expressões paramétricas (restrições), onde as variáveis DIM e RPS aparecem como parâmetros, o PG tem a função de criar o arquivo numérico correspondente ao layout. Estes parâmetros estão contidos na base de dados associada ao PG (BDG) e são atualizados pelo usuário ou pelo parametrizador elétrico.

O PG permite tornar as células elásticas (expandíveis ou condensáveis) em função de parâmetros DIM e RPS.

- MÓDULO PE (Parametrizador Elétrico)

O módulo PE é um compilador do desempenho da célula. Para um dado processo e conservando a topologia da célula, o PE calcula as dimensões dos dispositivos internos à célula tal que a mesma apresente o desempenho requerido pelo projetista. Para que isto seja possível, o PE cria modelos para cada célula, que relacionam parâmetros dimensionais com parâmetros comportamentais (tempos, correntes, etc) e operacionais (tensões de alimentação, cargas, etc). Quando o usuário entra com uma nova especificação comportamental para uma dada célula, o PE determina novas dimensões para os dispositivos internos à esta célula, de modo a satisfazer os parâmetros comportamentais e operacionais de entrada. Os novos parâmetros dimensionais, obtidos como resultado, são repassados ao PG.

- MÓDULO BASE DE DADOS

A base de dados do sistema compreende dois módulos:

- MÓDULO BDG (base de dados associada ao PG)

O módulo BDG tem a função de armazenar dados para uso do PG. Estes dados são os parâmetros dimensionais (DIM), as regras de projeto do fabricante e os códigos contendo as expressões descritivas do layout das células.

– MÓDULO BDE (base de dados associada ao PE)

O módulo BDE tem a função de armazenar dados para uso do PE. Estes dados são parâmetros comportamentais e operacionais, além das expressões de modelo relacionando estes parâmetros com parâmetros dimensionais.

#### • MÓDULO DE INTERFACE

Tem a função de tornar fácil a interação do usuário com os demais módulos do sistema (programa *friendly*).

O sistema opera da seguinte forma:

O usuário, apoiado pelo módulo de interface e pelo PG, cria o layout correspondente a uma dada célula, realizando o equacionamento e a compactação do *layout* em todas as direções, tendo em vista as regras de projeto e demais dimensões catalogadas na BDG. Após, o código contendo as expressões que descrevem o layout, é catalogado na BDG, e o PG cria o arquivo numérico correspondente a este *layout*, para os valores correntes dos parâmetros contidos na BDG. O *layout* na forma numérica deve ser catalogado na biblioteca de células do banco de dados do sistema de CAD, do qual o sistema de parametrização faz parte.

Uma vez criado o código paramétrico da célula, o usuário pode produzir novas versões para esta célula, especificando outros valores para os parâmetros dimensionais, alterando-os diretamente na BDG.

A otimização das dimensões da célula, de modo a atender às especificações de comportamento e operação, é realizada com o apoio do parametrizador elétrico (PE), gerando um modelo que relaciona o parâmetro elétrico a ser otimizado com parâmetros operacionais (OPE) e dimensionais (DIM).

Com base no modelo criado, o usuário pode alterar as características elétricas, ou seja, comportamentais e operacionais, da célula, especificando novos valores para esses parâmetros (CMP e OPE) na BDE. Com base nestes dados de entrada do usuário, o PE irá calcular as novas dimensões para os dispositivos internos à célula, de modo a satisfazer às especificações comportamentais e operacionais requeridas pelo usuário.

Por exemplo, o usuário pode especificar uma determinada corrente de saída, a uma dada carga, temperatura e tensão de alimentação, para uma célula da biblioteca. O PE, então, irá calcular as dimensões dos dispositivos internos à esta célula para que a mesma forneça a corrente de saída especificada para a temperatura, tensão de alimentação e carga estabelecidas na BDE. O modelo gerado no PE deverá conter a relação entre a corrente de saída, os parâmetros dimensionais que nela influem, tais como largura e comprimento de canal dos transistores, e os parâmetros operacionais, como temperatura e tensão de alimentação. As dimensões calculadas pelo PE são armazenadas na BDG, para que o PG possa compilar o layout para estas novas dimensões.

### 4.3 CARACTERÍSTICAS DO USUÁRIO DO SISTEMA

Os usuários do sistema devem ser projetistas de CI's e não devem ser usuários ocasionais.

A necessidade de serem projetistas deve-se ao fato da operação do sistema requerer, em várias etapas do processo, informações referentes à engenharia das células, tais como, determinação das dimensões da célula que devem ser catalogadas como parâmetros DIM na BDG (que influem no desempenho), para serem usadas numa posterior otimização da célula; determinação de um intervalo de variação para as dimensões da célula, no início da elaboração de um modelo, etc.

Não podem ser usuários ocasionais visto que, para tornar possível a execução de algumas etapas do processo, o usuário deve tornar-se um especialista. Por exemplo, se for utilizada uma forma textual para introduzir o layout ao sistema, a linguagem utilizada para descrevê-lo deverá ser conhecida pelo usuário, e somente assim ele será capaz de elaborar estes textos de forma eficiente. A interface com o usuário poderá ter a função de auxiliar na elaboração deste texto.

As exigências curriculares de tais usuários são:

- Formação em engenharia elétrica/eletrônica
- Experiência em projetos de sistemas, circuitos e dispositivos

Estas justificativas serão melhor entendidas quando forem abordados maiores detalhes sobre o sistema, posteriormente neste capítulo.

#### 4.4 REQUISITOS DO SISTEMA

Nas próximas seções serão especificados os requisitos do sistema, sob o ponto de vista de um projetista de circuitos integrados.

O PG não será detalhadamente descrito por já ter sido objeto de trabalhos anteriores (1, 4, 7 e 31 são os principais). Com base na experiência adquirida nestes trabalhos, serão levantados os requisitos do parametrizador geométrico (PG).

Será dada ênfase à descrição do PE, especialmente no tocante ao processo de geração de modelos, o qual possibilita a realização da parametrização elétrica.

A estrutura de arquivos na base de dados do sistema será apresentada.

##### 4.4.1 O PARAMETRIZADOR GEOMÉTRICO (PG)

###### 4.4.1.1 REQUISITOS DO PARAMETRIZADOR GEOMÉTRICO (PG)

Levando em consideração as limitações apresentadas pelas linguagens de descrição de *layouts* de células, será proposto o seguinte conjunto de requisitos, necessários a um programa que permita a confecção de lay-outs parametrizados, de modo a acompanhar a evolução da tecnologia e a permitir a otimização automática de células de CI's:

1. Permitir descrever o *layout*, numa forma intermediária, com restrições do tipo:

$$x_i = x_j$$

$$x_i - x_j \geq d(d > 0, \text{inteiro})$$

$$x_i - x_j = e(e > 0, \text{inteiro}),$$

tornando possível, desta forma, obter sempre a compactação máxima em todas as direções e permitindo não apenas dimensões e espaçamentos mínimos, como também dimensões ou espaçamentos de tamanho fixo, a critério do projetista.

2. Utilizar como entrada, através da interface com o usuário, uma linguagem de alto nível, se a célula for editada de forma textual, ou um editor gráfico, de modo a simplificar ao máximo a tarefa do projetista. A fim de aumentar a eficiência do usuário, este não deve lidar diretamente com as restrições nem efetuar cálculos, que devem ficar a cargo do programa. Isto reduz a incidência de erros durante a concepção e acelera o processo de geração do lay-out. O uso de entrada gráfica, exigirá um editor gráfico especialista, dedicado para este fim, já que os existentes são pacotes fechados, não sendo possível sua adaptação e utilização.

Este editor gráfico especialista deve possuir duas características adicionais que o difere dos editores gráficos comuns:

- permitir que a composição do *layout* seja feita através do posicionamento relativo entre elementos de circuito (microblocos), isto é, o usuário não necessita trabalhar sobre um reticulado (grid) e nem com coordenadas absolutas. Uma vez completada a edição, o PG poderá efetuar a compactação máxima do *layout* e, então, determinar suas coordenadas absolutas.
- permitir a indicação no *layout*, durante a edição, dos nomes das variáveis correspondentes a determinados segmentos do mesmo. Ao término da edição, estes nomes serão catalogados como parâmetros na BDG, com valores definidos pelo usuário.

O uso de um editor especialista é o meio mais indicado, já que é muito mais simples sob o ponto de vista do usuário, permitindo ao mesmo uma melhor visualização do layout, à medida que este está sendo concebido. A cada chamada de um microbloco catalogado na biblioteca, por exemplo,

o mesmo deve ser visualizado na tela, bem como sua adequação ao layout restante.

3. Permitir que, como parâmetros de *lay-out*, sejam utilizadas todas as regras de projeto da tecnologia, além de parâmetros especificados pelo usuário tendo em vista a otimização da célula.
4. Deve permitir que todos os parâmetros que compõem o texto da célula recebam valores de uma base de dados, de modo que, somente estes valores variem enquanto que o texto se mantém o mesmo. Assim, uma vez que o texto descritivo de uma célula é aceito pelo usuário, este *nunca mais* é editado e, a partir daí, o projetista só atua nos valores das variáveis, na base de dados.
5. Deve permitir a utilização de tipos, como os definidos em ALI (*standard types*), a serem escolhidos pelo usuário, tanto em número quanto em significado e suas relações com os demais. Por exemplo, estes tipos podem compreender todas as camadas de um processo CMOS, que são geralmente em torno de 12: poço, implante, difusão, poly, regiões P+, regiões N+, contato, metal1, via, metal2, passivação 1 e passivação 2.
6. Permitir a utilização de microblocos (macros) de modo a:
  - haver hierarquia na descrição do lay-out criado pelo usuário, tornando mais simples tanto a geração de uma entrada textual, como ainda a utilização do editor gráfico especialista.
  - reduzir a descrição do lay-out feito pelo usuário, a qual torna-se longa quando são usados retângulos ou polígonos como elementos primitivos de lay-out.
  - simplificar a implementação de diferentes alternativas de lay-out, de acordo com a área disponível em silício. Isto permite a criação de células adequadas para um dado espaço em silício, isto é, células com diferentes topologias. Suponha que, por exemplo, o usuário quer aumentar a capacidade de corrente de um *buffer*. De acordo com o requerido pelo sistema especificado neste trabalho, o usuário deve entrar com o valor especificado de corrente no PE, o qual calcula o W correspondente dos transistores e as novas larguras das linhas de saída.

Estes valores são repassados ao PG, que deverá, agora sem interação com o projetista, gerar o lay-out correspondente. É claro que, se a topologia se mantém constante, o buffer em questão ficará cada vez mais largo ( $W$  aumenta), podendo ocorrer deste lay-out não mais satisfazer ao projetista. Assim, havendo uma alternativa para o *layout* desse *buffer*, o programa pode substituir o microbloco do tipo transistor, por outro de igual largura e diferente topologia, adequado, por exemplo, a área disponível em silício. Uma interação com o projetista pode ser feita em seguida, para validação da solução.

Todos estes microblocos devem estar catalogados numa biblioteca, podendo ser criados pelo próprio programa, se o mesmo incorporar em suas bases de conhecimento definições estabelecidas pelo usuário. Com o uso de técnicas de inteligência artificial e sistemas especialistas, o programa pode *aprender* o que significa um transistor, isto é, pode incorporar ao seu conhecimento que um transistor é constituído por um cruzamento poly/difusão P+ ou N+ e, assim, ter condições de sugerir compactações diferentes para esta estrutura por ele conhecida, gerando microblocos adequados para determinadas áreas em silício.

A figura 12 apresenta um exemplo de duas compactações alternativas para um dado transistor.

Todo e qualquer microbloco utilizado pelo programa deve ser parametrizado, para que possa ser usado na concepção de células programáveis.

7. Utilizar hierarquia a nível de restrições, de modo a reduzir o número das mesmas, otimizando a memória utilizada e também reduzindo o tempo de processamento.
8. Permitir que o usuário introduza na BDG restrições, por exemplo do tipo:
  - que contatos com o meio externo permaneçam sobre um reticulado cuja dimensão seja um parâmetro;
  - que a altura e largura das células sejam dadas em múltiplos de unidade de reticulado, ainda que isto represente um gasto extra de área em silício;

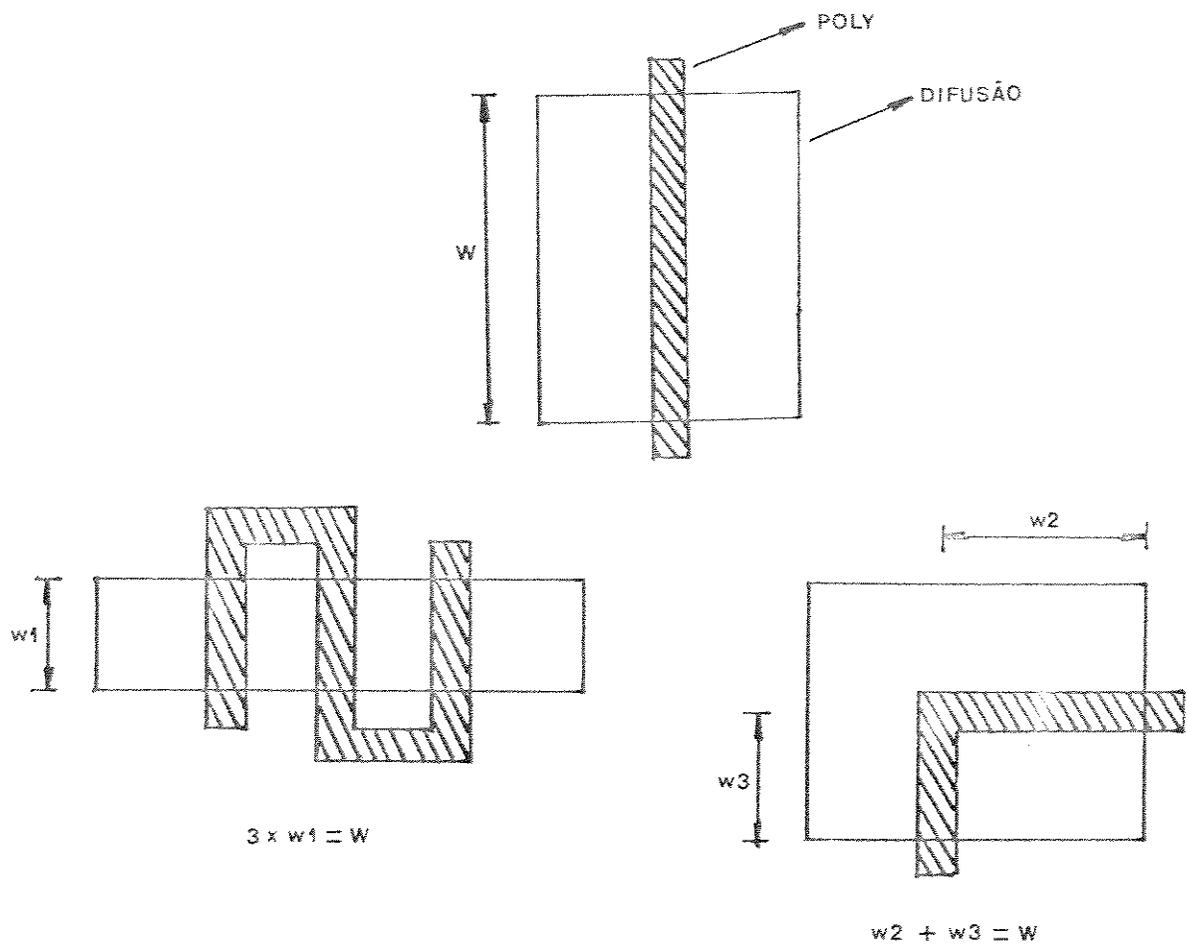


Figura 12: Exemplo de 2 Compactações Alternativas para um Transistor MOS

- que a máscara de contato, sempre que apresentar dimensão maior que a mínima, seja particionada no número máximo de contatos de dimensão mínima permitida pela tecnologia, que couberem na dada área. Isto é fundamental para garantir uma boa distribuição de corrente nas linhas.
9. Permitir que o usuário interaja na compactação obtida pelo programa de modo a alterar, por exemplo, as posições de terminais ou de elementos internos à célula.
  10. Como o usuário manipula livremente os valores das variáveis do banco de dados, podem ocorrer situações impossíveis de serem resolvidas pelo programa, para uma dada célula e um dado conjunto de restrições. Por exemplo, suponha que as células que estejam sendo concebidas não devam ultrapassar uma dada altura, catalogada na BDG como parâmetro H. Se o projetista especificar um W (largura de canal do transistor) ou GR (grid), tais que a altura máxima ultrapasse o limite estabelecido, o programa deve dar um aviso, ou seja, uma mensagem de erro, sobre a impossibilidade de solução para estes valores de entrada. O usuário, então, deverá decidir sobre alterar W, ou H, ou mudar a topologia da célula, etc.
  11. Deve admitir a incorporação de novas restrições com o tempo, atualizando os textos das células já existentes. Por exemplo, com a evolução dos processos de fabricação, ou troca de um fabricante por outro, existe a possibilidade de surgirem novas restrições no processo, isto é, novas R.P.'s e regras elétricas. Estas devem ser facilmente incorporadas ao programa, o qual deve ter uma estrutura modular, de modo a prever o acréscimo de algumas restrições e retirada de outras.
  12. Deve apresentar como saída um layout livre de violações às regras de projeto do fabricante, em formato compatível com as demais ferramentas que compartilham o mesmo ambiente de projeto.
  13. Deve operar em estações de trabalho, pois:
    - estas constituem a tendência mundial no projeto de CI's;
    - são de menor custo que os *main-frames*;

- suportam todas as ferramentas de suporte a projeto de CI's, tais como simuladores, programas de *placement/routing*, etc, algumas das quais irão utilizar as células geradas pelo parametrizador.
14. A fim de reduzir o número de variáveis utilizadas na descrição paramétrica do layout, o esforço computacional requerido na manipulação das mesmas e, conseqüentemente, aumentar a eficiência do programa, torna-se necessário restringir o uso do PG a *layouts* pequenos, ou seja, a células de CI's.

#### 4.4.2 O PARAMETRIZADOR ELÉTRICO (PE)

O parametrizador elétrico visa possibilitar a otimização semi-automática do desempenho da célula. Células parametrizadas eletricamente podem ter seus parâmetros elétricos alterados sem que ocorra um reprojeto das mesmas, ou seja, o projetista não necessita percorrer o ciclo tão conhecido de simulações elétricas e de re-edições do *layout*, apresentado na figura 9.

O PE, conforme proposto neste trabalho, tem a estrutura apresentada no diagrama em blocos da figura 13.

Os bancos de modelos e de parâmetros comportamentais (CMP) e operacionais (OPE), compõem a base de dados associada ao PE (BDE).

O PE é constituído basicamente de 2 módulos funcionais:

- um responsável pela geração de tabelas que descrevem o comportamento de cada célula (TRC - Tabelas Representativas de Comportamento). Estas tabelas contêm um conjunto de pontos que representam curvas relacionando parâmetros comportamentais, operacionais e dimensionais (os parâmetros de tecnologia são considerados invariantes).
- outro responsável pela execução de um procedimento de mapeamento vetorado, ou seja, um procedimento de busca nas tabelas TRC, das dimensões dos dispositivos internos a uma dada célula, de modo que a mesma satisfaça o desempenho requerido pelo usuário.

Nas seções seguintes, será feita a descrição de cada um dos módulos.

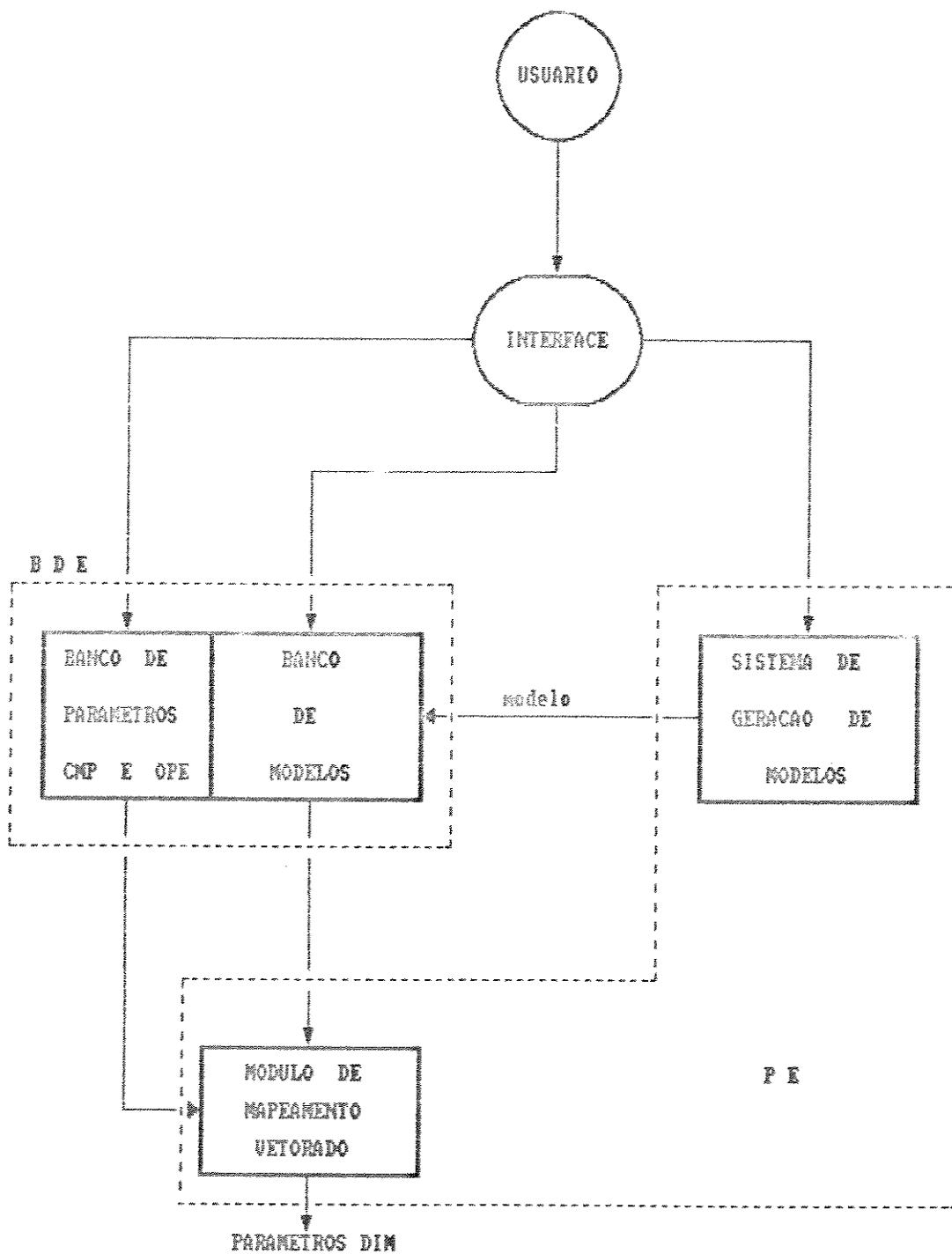


Figura 13: Estrutura do Parametrizador Elétrico (PE) e Sua Interação com a base de dados

#### 4.4.2.1 SISTEMA DE GERAÇÃO DE MODELOS REPRESENTADOS POR TABELAS

Esta seção se refere ao procedimento utilizado no módulo de apoio à geração de modelos, que compõe o PE. O procedimento descrito a seguir é semi-automático e os modelos gerados relacionam parâmetros elétricos com as dimensões dos dispositivos e parâmetros operacionais, de modo que o projetista possa especificar o valor do parâmetro elétrico que deseja otimizar e obter, através do PE, as dimensões dos dispositivos internos à célula para satisfazer este requisito.

Os modelos gerados são representados por tabelas denominadas TRC (Tabelas Representativas do Comportamento), as quais contêm as coordenadas dos pontos das curvas que relacionam os parâmetros mencionados acima.

Como exemplo de parâmetros elétricos comportamentais de interesse no caso do projeto de circuitos digitais podemos citar:

- tempos de propagação
- capacidade de corrente de saída
- potência dissipada em função da frequência

Usando um modelo derivado da descrição física dos transistores, pode-se determinar expressões algébricas que relacionam estes parâmetros comportamentais com os operacionais e as dimensões dos dispositivos:

$$T_p = f(W_{n1}, W_{p1}, L_{n1}, L_{p1}, W_{n2}, \dots, V_{dd}, \text{temperatura, parâmetros de processo})$$

$$I_{smáx} = g(W_{n1}, W_{p1}, L_{n1}, L_{p1}, W_{n2}, \dots, V_{dd}, \text{temperatura, parâmetros de processo})$$

$$P_d = h(W_{n1}, W_{p1}, L_{n1}, L_{p1}, W_{n2}, \dots, V_{dd}, \text{temperatura, parâmetros de processo})$$

As soluções destas expressões, em que o número de incógnitas é normalmente elevado, devem ser obtidas numericamente com base em critérios para escolha de valores para algumas das variáveis.

Essa descrição comportamental algébrica da célula, é limitada pelas seguintes razões:

- embora se possa incluir os efeitos parasitários mais importantes, devido à complexidade dos modelos resultantes estes estarão normalmente reduzidos a expressões simplificadas, com precisão limitada.
- o custo de engenharia necessário à elaboração manual destes modelos poderá ser proibitivo.
- a validação destes modelos algébricos exigirá alto investimento.

Assim, para tornar a geração de modelos:

- mais rápida e eficiente,
- mais precisa, permitindo a inclusão no modelo de mais informações do que é possível na geração manual (por ex. todos os parasitários internos à célula),

será proposta a alternativa descrita a seguir, ilustrada no fluxograma das figuras 14 e 15.

Primeiramente o projetista deve escolher o conjunto mínimo de parâmetros que viabilizem a flexibilidade desejada e obter do PG a descrição paramétrica da célula (ou utilizar a descrição de uma célula já catalogada na biblioteca), etapa que tem lugar no bloco B1.

A seguir a célula deverá ser compilada, ou seja, sua descrição será interpretada no PG para as regras de projeto (R.P.'s) da *foundry*, e para valores iniciais dos parâmetros dimensionais (no exemplo  $W_i$ ,  $L_i$ , etc) escolhidos pelo projetista. Todos estes parâmetros estão contidos na base de dados. Os intervalos de variação dos parâmetros DIM são obtidos ou pela experiência anterior do projetista, a qual poderá estar catalogada na base de dados, ou pela elaboração, também pelo projetista, de um modelo algébrico simplificado, que relacione os parâmetros dimensionais (DIM) e operacionais (OPE) com os comportamentais (CMP), que estão armazenados em B11.

Como valores iniciais dos parâmetros OPE e DIM, devem ser tomados os valores do início do intervalo especificado em B11, para que, posteriormente, para ser varrida toda a faixa de cada parâmetro, baste adicionar ao anterior o incremento determinado também em B11. O B11 representa a porção da base de dados que contém as especificações de entrada, do usuário, para os intervalos de variação de cada um dos parâmetros que irão compor o modelo, ou seja, o intervalo de validade do modelo.

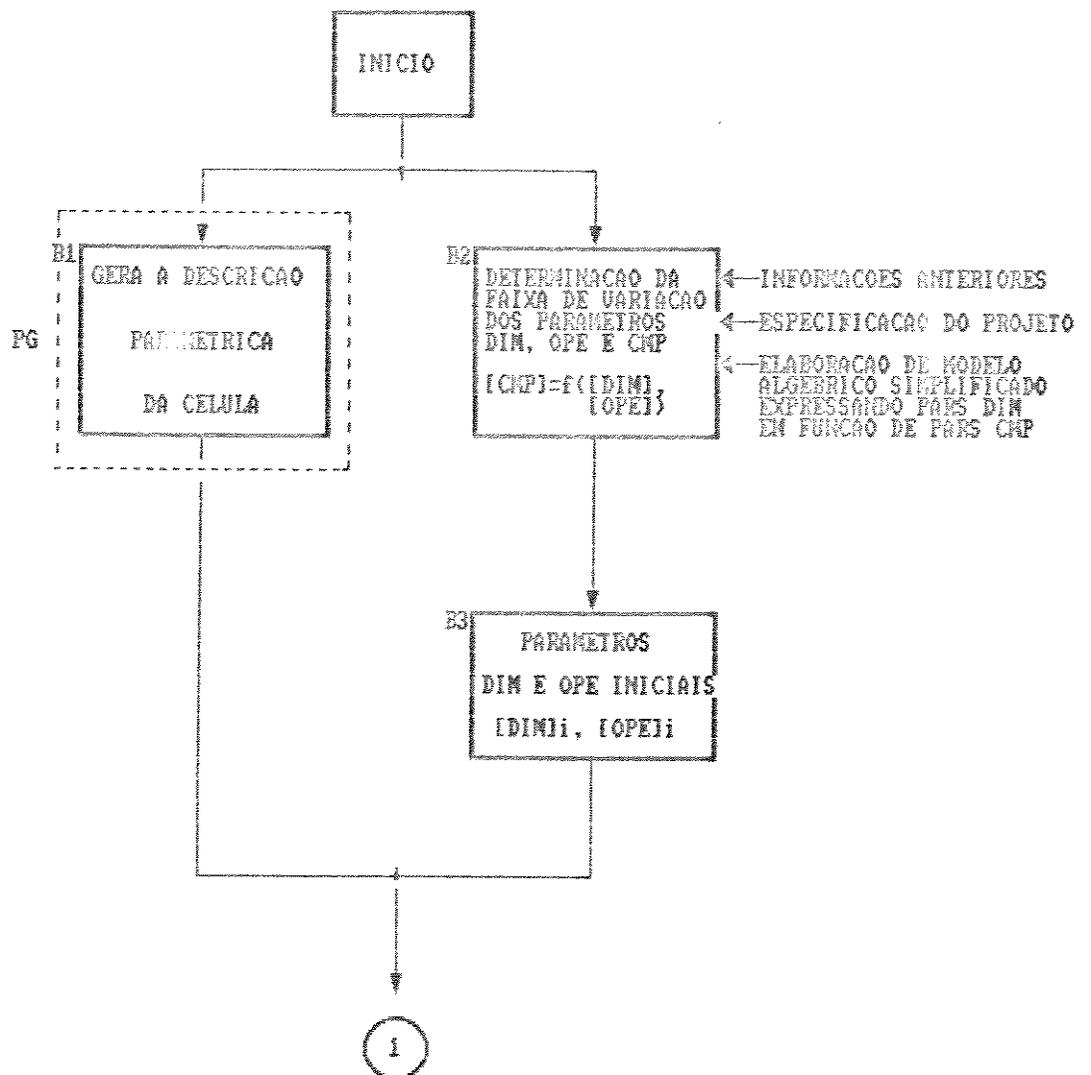


Figura 14: Fluxograma1 - Procedimento para Geraçao de Modelos

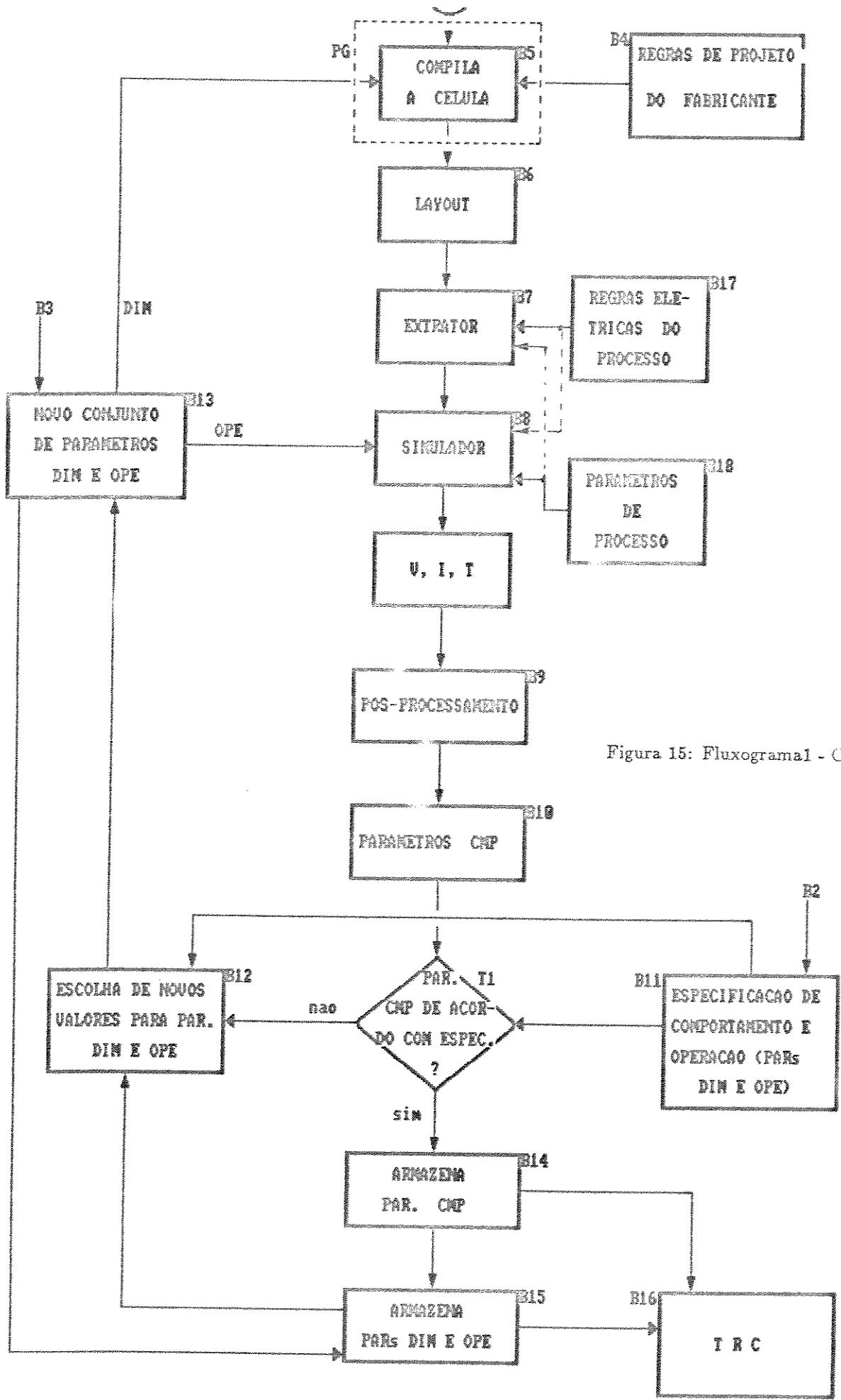


Figura 15: Fluxograma1 - Continuação

De qualquer forma, o projetista precisa estabelecer a faixa de variação do particular parâmetro elétrico para o qual deseja que a célula seja dimensionada, bem como a dos parâmetros operacionais, para poder estimar a faixa de variação dos parâmetros dimensionais que levem o parâmetro elétrico em questão a cair dentro deste intervalo. Este intervalo faz parte das especificações de comportamento contidas em B11. Os dados relativos às regras de projeto da *foundry* estão contidos no B4. O B5 indica a ação de compilação da célula, a qual é efetuada no PG.

O bloco denominado *layout da célula* (B6), é um arquivo que contém o *layout* com suas dimensões finais, gerado pelo PG, em formato compatível com a entrada de um programa extrator (B7), o qual permite obter o circuito elétrico equivalente com todos os parasitários correspondentes. O extrator não será discutido neste texto pelo fato de ser um software já bastante conhecido. A referência (14) aborda aspectos algorítmicos e de integração de tais programas dentro de um sistema de CAD.

A seguir, são realizadas várias simulações elétricas, com o objetivo de se montar a tabela, cujo conteúdo serão as coordenadas dos pontos que compõem as curvas mostradas na figura 16, para o exemplo do parâmetro elétrico tempo de propagação. Nas simulações são utilizados os parâmetros de processo (B18) e as regras elétricas dependentes do processo (B17). Os blocos B17 e B18 representam a parte do base de dados onde estão armazenados estes parâmetros.

As curvas esboçadas no exemplo da figura 16 dão  $t_{plh}$  (tempo de propagação para a saída variando de nível *low* para nível *high*) em função da carga ( $C_l$ ) e da largura de canal do transistor p ( $W_p$ ), sendo  $W_n$  mantido constante. Outros parâmetros poderiam ser plotados, como por exemplo, a tensão de alimentação e a temperatura.

A cada simulação os parâmetros a serem variados são atualizados um a um. Por exemplo, na obtenção do  $t_p \times C_l$ , o valor de  $C_l$  é incrementado a cada simulação, dentro de uma faixa de variação estabelecida em B11, enquanto os demais parâmetros são mantidos constantes. A cada incremento de  $W$ , as capacitâncias intrínsecas do transistor também variam, e são computadas pelo extrator de circuitos.

O projetista deve intervir não apenas para determinar quais parâmetros dimensionais e operacionais deverão variar durante as simulações, como foi

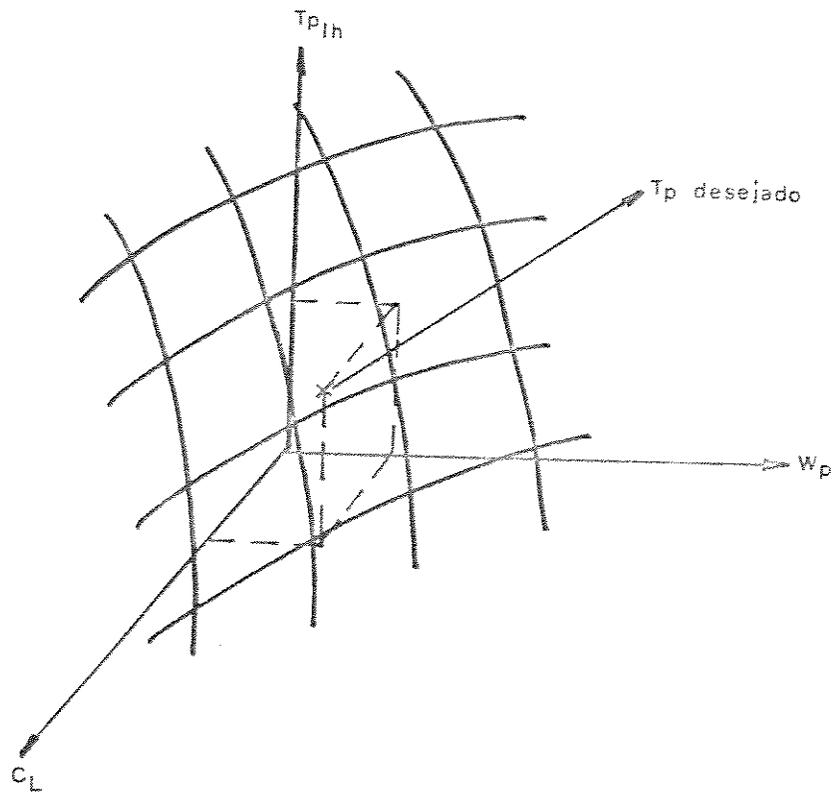


Figura 16: Exemplo de Uma Distribuição de Pontos Obtidos das Simulações Elétricas Durante a Geração de Modelos

mencionado acima, como também para dizer quais saídas o simulador elétrico deverá fornecer, isto é, se são correntes, tensões, frequências, etc, e de quais nós do circuito.

Ao final das simulações teremos como resultado um conjunto de pontos, ou seja, uma tabela gerada pelo simulador, contendo os parâmetros requeridos pelo projetista.

Estes resultados devem ser pós-processados para obtenção dos parâmetros comportamentais. Por exemplo, no caso em questão, tabelas de tensão de saída e entrada em função do tempo, emitidas pelo simulador para determinados parâmetros DIM e OPE, deverão ser pós-processadas para, a partir das mesmas, ser obtido o parâmetro CMP tempo de propagação. Os parâmetros CMP assim obtidos são armazenados no arquivo B10.

Cada parâmetro comportamental obtido é testado em T1, a fim de se verificar se o mesmo se encontra ou não dentro do intervalo especificado em B11. Se o resultado for positivo, o parâmetro CMP testado é armazenado na tabela TRC, procedimento indicado em B14. Os valores dos parâmetros DIM e OPE utilizados na simulação que deu origem ao último parâmetro CMP, são transferidos do B13 para a tabela TRC, procedimento indicado em B15.

A seguir, o B13 será atualizado com novos valores para os parâmetros DIM e OPE calculados por B12, dentro das faixas e incrementos especificados em B11, desde que o intervalo escolhido não tenha sido coberto. O bloco B13 é um arquivo utilizado para armazenar os parâmetros DIM e OPE que serão utilizados na simulação seguinte, os quais são atualizados pelo bloco B12.

Se o resultado do teste T1 for negativo, o valor obtido para o parâmetro CMP em questão, na última simulação, está fora da faixa estabelecida em B11, o que significa que este parâmetro CMP, bem como os parâmetros DIM e OPE utilizados na última simulação, não irão compor a tabela TRC, isto é, serão desprezados. Neste caso, é dado prosseguimento ao processo de varredura dos intervalos de variação dos parâmetros em B11. Assim, em B12 é feita uma nova escolha de valores para os parâmetros DIM. O critério de escolha utilizado em B12 será discutido posteriormente.

Uma vez completado todo ciclo de simulações para todos os valores e incrementos dos parâmetros DIM, CMP e OPE, especificados em B11, temos completada a elaboração da tabela TRC, que corresponde, no caso em questão, a uma tabela contendo  $tp \times Cl \times Vdd \times T$ .

Portanto, a tabela TRC representa um modelo, ou seja, o comportamento de uma dada célula, relacionando parâmetros CMP, DIM e OPE. Assim, esta tabela contém uma relação do tipo:

$$\text{par CMP} = f(\text{par DIM1}, \dots, \text{par DIMn}, \text{par OPE1}, \dots, \text{par OPEn})$$

No nosso exemplo,

$$t_p = f(W_1, \dots, W_n, C_l, V_{dd}, T),$$

sendo que os parâmetros de processo foram considerados constantes durante a elaboração do modelo. Este modelo será catalogado na base de dados do PE (BDE) para a célula específica.

A seguir serão feitas sugestões de implementação para os blocos que compoem o fluxograma das figuras 14 e 15.

#### 4.4.2.1.1 ESPECIFICAÇÃO DE COMPORTAMENTO E OPERAÇÃO (B11)

O arquivo B11 representa a porção da base de dados que contém a especificação, fornecida pelo usuário, da faixa de variação de cada parâmetro comportamental, dimensional e operacional, bem como seu incremento, para os quais cada célula deverá ser modelada. No caso do tempo de propagação, que está sendo usado como exemplo, o conteúdo de B11 para uma célula interna de um CI, seria do tipo:

faixa de  $\Delta$  de  $t_p$ :  $0.1ns \leq t_p \leq 0.7ns$ ; incremento de  $0.05ns$   
faixa de  $\Delta$  de  $Cl$ :  $0.15fF \leq Cl \leq 0.6F$ ; incremento de  $0.1fF$   
faixa de  $\Delta$  de  $V_{dd}$ :  $4.5V \leq V_{dd} \leq 5.25V$ ; incremento de  $0.5V$   
faixa de  $\Delta$  de  $T$ :  $0^{\circ}C \leq T \leq 70^{\circ}C$ ; incremento de  $5^{\circ}C$   
faixa de  $\Delta$  de  $W1$ :  $4\mu m \leq W1 \leq 20\mu m$ ; incremento de  $2\mu m$   
. . .  
faixa de  $\delta$  de  $Wn$ :  $4\mu m \leq Wn \leq 20\mu m$ ; incremento de  $2\mu m$

A experiência do projetista foi utilizada na determinação dos intervalos do exemplo acima, visto que:

- $V_{dd}$ ,  $T$  - dependem da especificação do equipamento para o qual os CI's serão destinados. Ex: telecomunicações, uso militar, etc.
- $t_p$  - depende das velocidades dos circuitos onde a célula será utilizada, o que também é função da aplicação, ou seja, dos equipamentos para os quais os CI's serão destinados.
- $Cl$  - depende da célula ser de I/O ou interna, e da tecnologia utilizada. No exemplo acima, como foi suposto que esta célula seria interna, estimou-se que esta enxergaria no máximo 4 iguais a ela e no mínimo uma, sendo o valor unitário de  $Cl$  estimado em função da tecnologia e dos tamanhos das demais células básicas.

- $W_1$  a  $W_n$  (largura de canal de cada um dos  $n$  transistores que constituem a célula) - depende da familiaridade do projetista com o desempenho da tecnologia. Diferentemente dos demais, os intervalos fornecidos para estes parâmetros constituem uma estimativa, por parte do usuário, de modo a estabelecer para o programa dentro de qual intervalo os parâmetros  $W_n$  poderão ser variados. O valor mínimo dos parâmetros  $W$ , poderá ser o mínimo permitido pela tecnologia, e o máximo, um valor que, com certeza, incorpore o intervalo estimado pelo projetista, de modo a respeitar a faixa especificada em B11, figura 14, para os parâmetros CMP.

Os incrementos devem ser determinados pelo usuário levando-se em conta o compromisso precisão x eficiência durante a geração do modelo. Óbvio, incrementos menores implicam em um maior número de pontos e, conseqüentemente, de simulações elétricas, o que melhora a precisão das curvas (tabelas) geradas. Entretanto, o consumo de tempo também se eleva consideravelmente, pois são as simulações que demandam a maior parte do tempo gasto na geração do modelo. Além disso, as tabelas tornam-se maiores, aumentando a memória necessária ao seu armazenamento.

Como a precisão e a velocidade com que o processo de geração do modelo se efetua dependem do valor escolhido para o incremento, pode-se utilizar um algoritmo que o otimize, a partir de critérios estabelecidos pelo projetista.

#### 4.4.2.1.2 CRITÉRIOS DE ESCOLHA DE NOVOS VALORES PARA OS PARÂMETROS DIM E OPE (B12)

A função de B12 é calcular os valores dos parâmetros DIM e OPE a cada simulação. Os parâmetros OPE e DIM são variados dentro das faixas especificadas em B11, com as taxas de variação especificadas no mesmo bloco. O procedimento de varredura pode ser representado pelo esquema da figura 17.

De acordo com a figura 17, cada parâmetro deve ser variado em toda a extensão de sua faixa, para cada valor dos demais parâmetros, cobrindo todas as combinações possíveis entre os parâmetros, sendo que, a cada combinação, corresponde uma atualização do bloco B13 e uma simulação elétrica.

Assim sendo, após a execução do teste T1, quer o resultado do mesmo seja positivo ou negativo, é acionado o processo de escolha de novos valores para os parâmetros DIM e OPE. Uma vez acionado o processo, o programa irá retomar a varredura esquematizada na figura 17, continuando a partir do ponto em que se encontrava antes da última simulação. Quando todos os parâmetros já tiverem sido variados em função dos demais, é dado por completo o ciclo de simulações.

Para melhorar a eficiência do programa, podem ser feitas as seguintes considerações:

1. Como a faixa de variação dos parâmetros DIM é estimada pelo usuário, este provavelmente irá ampliá-la por um fator de segurança. Isto pode acarretar muitas combinações destes parâmetros DIM com os parâmetros OPE, que, nas simulações, resultem em pontos não aproveitados, isto é, que caiam fora do intervalo especificado e sejam rejeitados no teste T1, portanto, não sendo utilizados na montagem da tabela. Assim, para evitar que a varredura prossiga através destes parâmetros, mal estimados pelo usuário, o programa pode interromper o ciclo de simulações antes que todos os intervalos correspondentes aos parâmetros DIM tenham sido completamente varridos. Por exemplo, se para o último incremento do parâmetro que varia mais lentamente (DIM2 no esquema da figura 17) e sua combinação com todos os demais só resultaram parâmetros CMP, obtidos das simulações, fora da faixa especificada em B11, e, portanto, rejeitada em T1, o programa pode encerrar o ciclo de simulações, visto

Varição Parâmetro 1	1 → n1	...	1 → n1	1 → n1	1 → n1	...	1 → n1	1 → n1	1 → n1	...	1 → n1	1 → n1	...	1 → n1
Varição Parâmetro 2	1	2	...	n2	1	2	...	n2	1	2	...	n2	1	2
Varição Parâmetro 3	1	2											n3	

n1 = enésimo parâmetro dentro do intervalo de variação do parâmetro 1  
 n2 = enésimo parâmetro dentro do intervalo de variação do parâmetro 2  
 n3 = enésimo parâmetro dentro do intervalo de variação do parâmetro 3

Figura 17: Exemplo Do Procedimento de Varredura Efetuado No Bloco B12 Para Três Parâmetros

que, é de se supor que futuros incrementos também acarretarão pontos fora da faixa.

2. Para reduzir o número de combinações possíveis entre todos os parâmetros que irão compor o modelo, o usuário pode fornecer, quando for o caso, informações do tipo:

- todos os transistores *pMOS* terão mesma largura
- todos os transistores *nMOS* terão mesma largura
- todos os transistores *nMOS* terão sempre largura menor ou igual aos transistores *pMOS*

Estas informações reduzem o número de combinações utilizadas para os parâmetros DIM, dentro dos intervalos fornecidos pelo usuário. Como a cada combinação corresponde uma simulação, o resultado final seria bastante beneficiado.

#### 4.4.2.2 MÓDULO DE MAPEAMENTO VETORADO

O módulo de mapeamento vetorado efetua um procedimento através do qual o programa irá buscar, nas tabelas TRC, as dimensões dos dispositivos internos a uma dada célula, de modo que a mesma satisfaça os parâmetros CMP e OPE de entrada, fornecidos pelo usuário. O usuário determina um caminho (vetor) e o programa efetua a busca (mapeamento).

O caminho é indicado pelo usuário, através do fornecimento dos parâmetros por ele conhecidos (parâmetros de entrada), isto é, os parâmetros CMP e OPE, os quais fazem parte da especificação do circuito. Conforme explicado anteriormente, o usuário atribui novos valores a tais parâmetros na base de dados associada ao PE (BDE).

A busca, consiste na consulta à tabela TRC, para encontrar valores para os parâmetros não conhecidos pelo usuário, no caso, as dimensões dos dispositivos.

Como normalmente uma célula é constituída por um conjunto de dispositivos, espera-se que tenhamos com frequência mais de um único dispositivo influenciando no desempenho da célula, e, portanto, um número de parâmetros de saída (dimensões dos dispositivos) maior ou igual ao número de parâmetros de entrada (parâmetros comportamentais e operacionais). Isto significa que poderão existir várias soluções possíveis para as dimensões dos dispositivos, para determinados parâmetros de entrada. Neste caso, existirão várias posições da tabela TRC indicadas pelos parâmetros de entrada, cada uma conduzindo a uma dada combinação de valores para os parâmetros DIM. Todas estas soluções devem ser retornadas ao usuário, para que ele efetue sua escolha.

Caso não existam pontos da tabela coincidentes com os valores dos parâmetros de entrada do usuário, serão tomados os pontos mais próximos da tabela, ocorrendo, no pior caso, para cada parâmetro, um erro igual à metade da taxa de variação deste parâmetro.

Assim, a taxa de variação de cada parâmetro deve ser feita o menor possível, de modo a minimizar este erro.

Óbviamente, uma aproximação para mais ou para menos, de cada um dos parâmetros de entrada, conduz a posições diferentes da tabela. Entretanto, se as taxas de variação destes parâmetros são mínimas, os conjuntos de dimensões correspondentes a cada uma destas não podem ter diferenças signifi-

cativas. Por esta razão, no fluxograma da figura 14, é requerida a intervenção do projetista no fornecimento do intervalo e da taxa de variação de cada um dos parâmetros que irão compor a tabela.

Um ponto importante a ser observado é que, embora a redução da taxa de variação dos parâmetros conduza a uma melhor precisão dos resultados, também aumenta o número de pontos da tabela, e portanto, a memória alocada ao armazenamento das mesmas.

#### 4.4.2.3 CONCLUSÕES

Esta seção apresentou um método semi-automático de geração de modelos para células de CI's, que incorporam a relação entre parâmetros comportamentais (CMP), operacionais (OPE) e dimensionais (DIM). Os modelos gerados segundo este método apresentam as seguintes características:

1. são representados por tabelas cujo conteúdo são pontos obtidos de simulações elétricas, que traduzem as curvas de comportamento das células
2. são válidos dentro do intervalo especificado para os parâmetros CMP e OPE no B11, figura 15
3. são válidos para o processo para o qual foram gerados
4. possuem tempo de geração crescente com:
  - o número de parâmetros OPE e DIM
  - a diminuição da taxa de variação especificada para cada parâmetro em B11, figura 15
  - o aumento do intervalo de variação especificado para cada parâmetro em B11, figura 15
5. a precisão das tabelas geradas é função dos erros obtidos nas simulações elétricas e no módulo de mapeamento vetorado

As vantagens do método anterior são:

1. rapidez e confiabilidade relativamente à geração manual
2. inclui com precisão todos os parasitários internos à célula, visto que o circuito é extraído antes de cada simulação. Portanto, inclui mais informações sobre o circuito que um modelo gerado manualmente
3. dispensa a homologação necessária à geração manual

Para que seja possível reduzir o esforço computacional requerido na elaboração das tabelas TRC, bem como no mapeamento para os parâmetros elétricos (CMP e OPE) de entrada, atualizados pelo projetista na BDE e, portanto, aumentar a eficiência do programa, é necessário assumir as seguintes restrições:

- Retringir o uso do PE a células

Isto reduz o número de parâmetros DIM. Conseqüentemente, reduz o número de simulações elétricas (pois diminui o número de parâmetros a serem variados) e, portanto, o tempo de geração da tabela, além de acarretar tabelas menores, de busca(mapeamento) mais rápido.

- O domínio de aplicação do PE deve ser restrito a um dado grupo de circuitos (digitais, analógicos, etc)

Isto define qual será o simulador elétrico utilizado durante a geração dos modelos. Algumas versões do simulador SPICE podem ser usadas tanto para CI's digitais como para analógicos. Entretanto, alguns blocos analógicos podem consumir tempo excessivo de simulação no SPICE, como por exemplo é o caso dos circuitos a capacitores chaveados. Neste caso, um simulador como o SWITCAP pode ser mais indicado, bem como outros disponíveis no mercado, por exemplo o simulador SWAP. Também, o projetista de analógico pode precisar de um modelamento preciso na região de sublimiar, disponível por exemplo nas versões do SPICE que contém o modelo BSIM.

Assim, torna-se necessário definir sempre o domínio de aplicação para o PE, já que isto irá também definir outros fatores, como por exemplo os formatos dos arquivos intermediários decorrentes do processo de geração das tabelas, que devem ser compatíveis com as saídas do simulador, o erro a ser admitido no módulo de mapeamento vetorado e, portanto, a taxa de variação de cada parâmetro durante a geração dos modelos, etc.

#### 4.4.3 ESTRUTURA DOS ARQUIVOS NA BASE DE DADOS

Os arquivos da base de dados estão estruturados da forma apresentada abaixo:

1	célula1	célula2	. . . . .	célulan
2	esq1	esq2	. . . . .	esqn
3	cód par1	cód par2	. . . . .	cód parn
4	layout1	layout2	. . . . .	layoutn
5	pars DIM1	pars DIM2	. . . . .	pars DIMn
6	pars OPE1	pars OPE2	. . . . .	pars OPEn
7	pars CMP1	pars CMP2	. . . . .	pars CMPn
8	tabs1/val	tabs2/val	. . . . .	tabsn/val
9	pars RPS dos fabricantes 1 a n			
10	pars ELE dos fabricantes 1 a n			
11	pars PPR dos fabricantes 1 a n			
12	informações sobre outros projetos			

Os conteúdos dos arquivos que compoem a base de dados são descritos abaixo:

linha1: nome das células 1 a n, muitas das quais são também microblocos do sistema. As células podem ser quaisquer, desde que pertençam a uma mesma tecnologia. Os microblocos podem ser: transistores de diferentes topologias, contatos, elementos de interconecções, etc.

linha2: esquema lógico e/ou elétrico correspondente às células 1 a n.

linha3: código paramétrico correspondente às células 1 a n.

linha4: layout numérico das células 1 a n, no formato de saída do PG.

linha5: parâmetros dimensionais utilizados no código de cada uma das células, bem como seus valores. Podem ser, por exemplo, larguras e comprimentos de canal, larguras de trilhas de alimentação e terra, altura padrão das células, etc.

linha6: parâmetros operacionais utilizados no modelamento de cada uma das células, bem como seus valores. Podem ser, por exemplo, tensões de alimentação, cargas acopladas ao nó de saída, temperatura, etc.

linha7: parâmetros comportamentais utilizados no modelo de cada célula, bem como seus valores. Podem ser, por exemplo, tempos de propagação, correntes de saída, frequência de operação, margens de ruído, ganho, etc.

linha8: tabelas representando o comportamento de cada célula, geradas no parametrizador elétrico (PE), bem como a faixa de validade das mesmas, para cada parâmetro OPE e CMP.

linha9: conjuntos de regras de projeto dos fabricantes 1 a n. Essas regras dão larguras e espaçamentos mínimos para as camadas que compoem o processo.

linha10: conjuntos de regras elétricas dos fabricantes 1 a n. São elas: densidade de corrente nas linhas de diversas camadas, capacitâncias, resistividades, etc.

linha11: conjuntos de parâmetros de processo dos fabricantes 1 a n. Ex: espessura do óxido de porta, difusão lateral sob o gate, tensão de limiar, etc.

linha12: informações sobre projetos anteriores, que dêem suporte ao usuário em atividades futuras. Por exemplo, a estimativa dos intervalos de variação dos parâmetros DIM, no início do processo de geração dos modelos.

### OBSERVAÇÕES:

1. os layouts da linha 4 são fixos para os valores dos parâmetros DIM da linha 5.
2. os layouts correntes catalogados nos arquivos da linha 4, possuem as características elétricas registradas nas linhas 6 e 7.
3. os microblocos são catalogados como células, e qualquer das células podem ser acessadas pelo usuário para a composição de novos layouts.

#### 4.4.4 INTERFACE COM O USUÁRIO

As opções sugeridas pelo programa, quando o usuário entra no sistema de parametrização, são ilustradas na árvore da figura 18:

##### *OPÇÃO 1: Geração de Novo Layout*

Escolhida a opção 1, o sistema deverá possibilitar ao usuário a entrada num modo de edição, gráfica ou textual, para que seja editado o novo *layout*.

O usuário deverá editar a célula, chamando os microblocos que irão compor esta célula; deverá indicar as dimensões que serão tomadas como parâmetros e dar nomes às mesmas. Estas dimensões serão entendidas pelo sistema como parâmetros DIM, e serão catalogadas na base de dados com os valores utilizados no *layout*.

Se a entrada for textual, uma vez pronta a descrição, o usuário deverá submetê-la a um analisador léxico para depurá-la, até que a mesma esteja sem erros de sintaxe.

Uma vez pronta a edição do *layout*, o usuário avisa ao sistema para que este mostre o *layout* no vídeo, já com a compactação obtida pelo programa, de acordo com as regras de projeto do fabricante e parâmetros DIM da base de dados. Se o usuário não estiver satisfeito poderá interagir, substituir microblocos por outros de diferentes topologias, etc.

Quando o usuário estiver satisfeito, determina que o sistema armazene no base de dados, a descrição paramétrica do *layout* resultante, criada pelo sistema, sob um dado nome.

Esta opção deverá fornecer como saída o *layout* para o vídeo e plotter.

##### *OPÇÃO 2: Otimização do Layout de uma Célula*

Se o usuário deseja alterar os valores das dimensões (parâmetros DIM), ou a topologia dos dispositivos internos a uma célula já existente, o programa

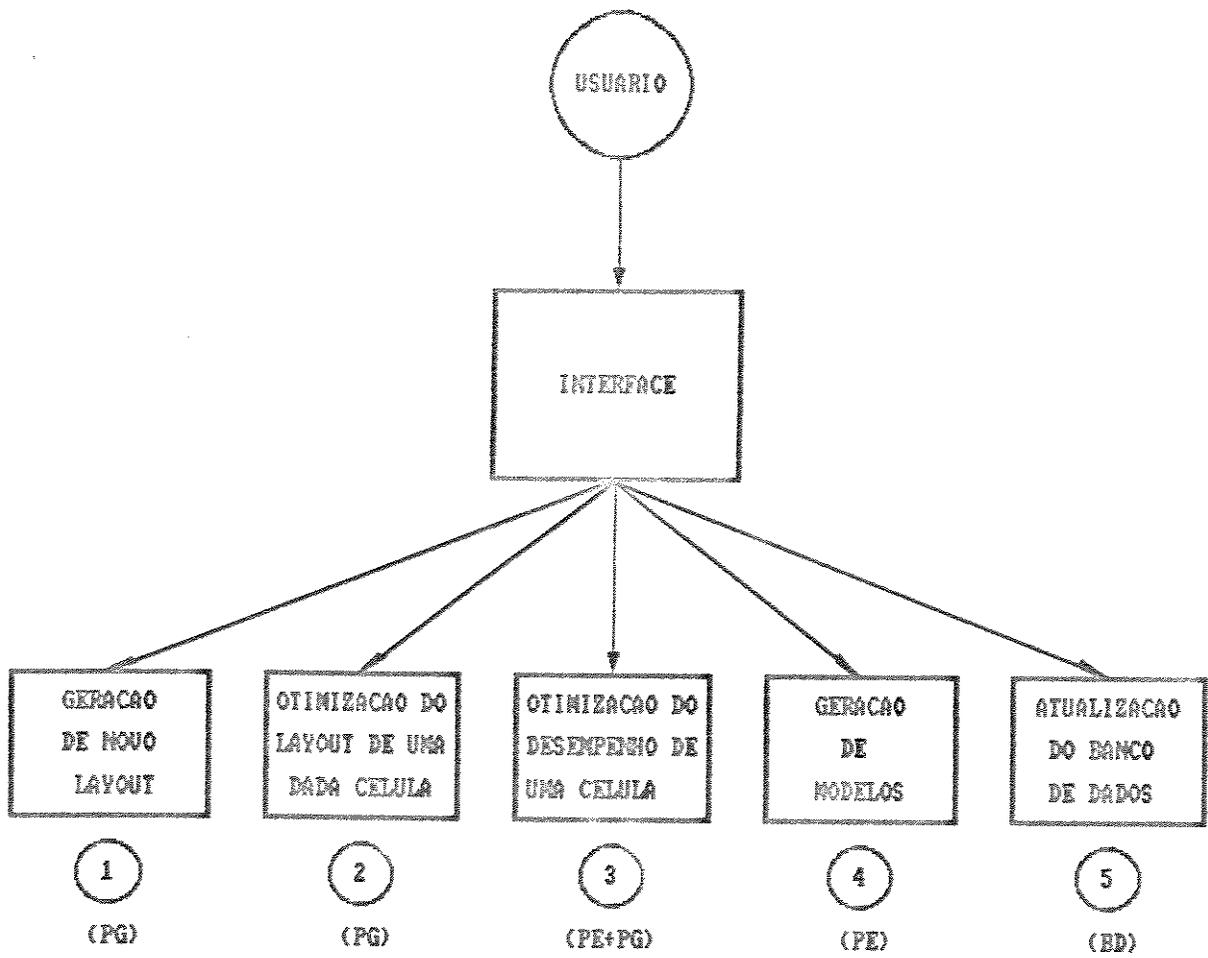


Figura 18: Ilustração das Opções do Usuário no Sistema

deverá primeiramente, pedir o nome da célula a ser alterada. O usuário deverá entrar com o nome da célula, e o sistema deverá retornar ao usuário o layout no vídeo, no qual estão indicadas as dimensões utilizadas como parâmetros DIM, bem como a lista desses parâmetros, catalogadas para esta célula na base de dados, com seus respectivos valores. O usuário poderá, então, editar livremente o *layout* exibido no vídeo.

A substituição de microblocos por outros de diferentes topologias pode ser feita normalmente, da mesma forma que na opção 1.

Se o usuário desejar alterar os valores dos parâmetros DIM, deverá requisitar ao programa, o qual, por sua vez, deverá pedir ao usuário os nomes dos parâmetros DIM a serem alterados, bem como seus novos valores.

Após efetuar as modificações, pedir ao programa uma recompilação da descrição paramétrica desta célula, com os parâmetros DIM atualizados.

O layout resultante deverá ser apresentado ao usuário no vídeo e, se for requerido, também no plotter.

Se o usuário estiver satisfeito com o layout final, pede ao programa para catalogá-lo na base de dados e entra com um nome para a nova célula, senão, pede para fazer novas alterações no *layout* e nos parâmetros DIM.

### *OPÇÃO 3: Otimização do Desempenho de uma Célula*

Para que o usuário utilize esta opção, os modelos para a célula a ser otimizada quanto ao seu desempenho já devem ter sido gerados e catalogados na base de dados, por meio da opção 4.

Quando o usuário selecionar a opção 3, o programa deve pedir ao mesmo o nome da célula a ser otimizada.

O usuário deve entrar com o nome da célula e o programa, recebendo esta informação, deve acessar a base de dados, fornecendo para o usuário, pelo vídeo, os parâmetros CMP e OPE catalogados para esta dada célula, bem como seus respectivos valores.

A seguir, o programa deverá pedir ao usuário o nome dos parâmetros que deseja atualizar e seus novos valores.

O usuário, então, entra com estes dados e aguarda a execução do programa de mapeamento vetorado. Ao final da execução, o programa retorna ao usuário todas as possíveis soluções para as dimensões dos dispositivos, caso exista mais de uma. O usuário, por sua vez, escolhe uma das soluções e pode requerer ao programa a exibição do *layout* redimensionado para o desempenho requerido, no vídeo. Também será gerada saída para *plotter*, a pedido do usuário.

Se os dados de entrada do usuário (parâmetros CMP e OPE) estiverem fora do intervalo especificado para a tabela TRC, o programa deverá comunicar o usuário e aguardar a entrada de novos dados.

#### *OPÇÃO 4: Geração de Modelos*

Para dar início ao processo de geração de modelos (tabelas TRC), o programa deverá pedir ao usuário para entrar com os seguintes dados:

- nome e faixa de variação dos parâmetros CMP, OPE e DIM que irão compor o modelo,
- taxa de variação de cada um dos parâmetros dentro de seus respectivos intervalos,
- nós das células correspondentes a saídas das simulações elétricas,
- estímulos de entrada (os quais irão depender do parâmetro CMP que estiver sendo modelado), bem como o nó da célula onde os mesmos deverão ser injetados durante as simulações elétricas,
- nome da célula para a qual o modelo se destina,

Após receber estes dados, o programa realizará o processo de geração da tabela TRC. Ao término, o programa irá catalogar a tabela na BDE e deverá comunicar o usuário de que o processo foi completado.

## *OPÇÃO 5: Atualização da base de dados*

Quando o usuário utiliza esta opção, o programa deve perguntar que categoria de parâmetros o usuário deseja atualizar. As categorias são as seguintes:

- parâmetros de processo
- regras de projeto
- regras elétricas
- informações sobre outros projetos

O usuário poderá alterar dados existentes ou acrescentar novos.

Uma vez escolhida a categoria, o programa deverá apresentar uma lista destes parâmetros e seus valores correntes.

O usuário fará as alterações no vídeo e o programa poderá armazená-las sobrepondo os valores antigos, ou criando outro subconjunto de parâmetros de nome diferente, a critério do usuário. Por exemplo, se forem alteradas as regras de projeto do fabricante, os parâmetros correspondentes poderão desaparecer e os novos, serem catalogados na base de dados, no lugar dos antigos. Ainda, o usuário poderá requerer que o programa crie um outro conjunto de regras de projeto, mantendo a versão antiga. Assim, no sistema estaria disponível mais um conjunto de regras de projeto, no mesmo banco de dados.

### **OBSERVAÇÕES:**

1. A qualquer instante, em qualquer das opções, o usuário deverá ter condições de consultar os parâmetros que estão na base de dados
2. Mensagens de erro deverão ser enviadas para o usuário através do vídeo, sempre que ocorrerem falhas do tipo:
  - o usuário pedir alterações de células cujos nomes não existam no base de dados

- o usuário pedir mudança de valores em parâmetros inexistentes
- o usuário entrar com parâmetros DIM tal que a confecção do layout, para estes valores, seja impossível

## 5 BIBLIOGRAFIA

1. R. Manione; G. P. Constantino - "Grafic Editor For Parametric Cells Design" - IEEE 1986 Custom Integrated Circuits Conference - Rochester, New York.
2. H. C. Benz - "Parameterized Modules in Standard-Cell Workstation Libraries" - IEEE 1986 Custom Integrated Circuits Conference - Rochester, New York.
3. R. C. Newton; M. D. Lantz - "A High Performance, Multisourceable, Scalable CMOS Cell Family" - IEEE 1986 Custom Integrated Circuits Conference - Rochester , New York.
4. José Monteiro da Mata - "ALLENDE - A Procedural Language for the Hierarchical Specification of VLSI Layouts" - Princeton University, 1984.
5. A. J. Kuchinsky; W. A. Barrett; R. G. Rogers; M. J. Sorens; J. S. Gibson - "ICPL - Integrated Circuit Procedural Language" - IEEE Proceedings of the International Conference on Computer Design.
6. M. C. Bissel, Jr - "Parameterized Layout Synthesis of VLSI Devices" - VLSI Design, february, 1985.
7. R. C. Di Giorgio; C. I. Z. Mammana; A. M. P. Kuniyoshi - "parametrização de Células de CI's" - 5 Simpósio Brasileiro de Microeletrônica, 1985.
8. A. G. da Silva Jr - "Um Sistema de Modelagem Automática de CI's Digitais MOS" - Tese de Mestrado - Faculdade de Engenharia de Campinas - UNICAMP.
9. A. Vladimirescu; S. Liu - "The Simulation of MOS Integrated Circuits Using SPICE2" - Electronics Research Laboratory, College of Engineering - University of California - Berkeley.
10. S. M. Sze - "Physics of Semiconductor Devices" - John Wiley & Sons.
11. Kim, J.; J. McDermott - "TALIB: An IC Layout Design Assistant" - The National Conference on Artificial Intelligence, 1983.

12. Kim, J.; J. McDermott; D. P. Siewiorek - "Exploiting Domain Knowledge in IC Cell Layout" - IEEE Design and Test, 1984.
13. R. C. Di Giorgio; M de Moraes - "LPF : Linguagem de Parametrização de Figuras" - CTI, documento interno, 1986.
14. O. L. Aoki - "Extratores e Extração de Circuitos" - 2 Congresso da Sociedade Brasileira de Microeletrônica, Centro de Convenções Rebouças, julho, 1987, São Paulo, S.P.
15. E. Friedman; W. Marking; E. Iodice; S. Powell - "Parameterized Buffer Cells Integrated Into An Automatic Layout System - "IEEE Custom Integrated Circuits Conference, 1985.
16. Lursinsap, C., et al - "Cell Compilation with Constraints" - ACM IEEE 21st Design Automation Conference Proceedings, june, 1984, pp. 103-108.
17. Law, H. S., et al - "An Intelligent Composition Tool for Regular and Semi-Regular VLSI Structures" - IEEE International Conference on Computer Aided Design Digest of Technical Papers, pp. 169-171, november, 1985.
18. Fillmore, R. - "Standard Cell Expandable Static CMOS RAM" - Proceedings of the IEEE 1983 Custom Integrated Circuits Conference, pp. 115-118, may, 1983.
19. J. M. Mata - "A Methodology for VLSI Design and a Constraint-Based Layout Language" - Ph. D. Thesis, Princeton University, oct., 1984.
20. J. Dimov, C. Z. Mammana - "LPG - Linguagem de Procedimentos Gráficos" - Laboratório de Eletrônica e Dispositivos (LED) - UNICAMP.
21. R. J. Lipton; S. C. North; R. Sedgewick; J. Valdes; G. Vijayan - "ALI - A Procedural Language to Describe VLSI Layouts" - 19th Design Automation Conference, 1982, p. 467.
22. G. P. Constantino, R. Manione - "On Semantic Aspects of Interactive Parametric Cells Design Languages" - ICCAD87.
23. S. Wimer; R. Y. Pinter - "Optimal Chaining of CMOS Transistors in a Functional Cell" - IEEE Transaction on CAD, vol. 6, n. 5, set 87, pp. 795-801.

24. N. Hedenstierna; K. O. Jeppson - "CMOS Circuit Speed and Buffer Optimization" - IEEE Transaction on CAD, vol. 6, n. 2, mar 87, pp. 270-281.
25. H. Mathony; U. G. Baitinger - "CARLOS: An Automated Multilevel Logic Design System for CMOS Semi-Custom Integrated Circuits" - IEEE Transaction on CAD, vol. 7, n. 3, mar 88, pp. 346-355.
26. K. Croes; H. J. De Man - "CAMELEON: A Process-Tolerant Symbolic Layout System" - IEEE Journal on Solid State Circuits, vol. 23 n. 3, jun 1988, pp. 705-713.
27. M. Baffleur - "Contribution à L'étude des Performance et la Conception Sure de Fonctionnement des Circuits Intégrés CMOS Modernes" - Thèse de Doctorat d'État Université Paul Sabatier de Toulouse, capitre 2 , p. 93, mai 1987.
28. M. H. White - "Characterization of CMOS Devices for VLSI" - IEEE Transaction on Electron Devices, vol. ED-29. n. 4, april 1982, p. 578.
29. M. Baffleur, J. Buxo; P. Dall'Ágnese - "Optimization du Design en CMOS - Aplication aux Réseaux Prédifusés"- Journées GCSIS, Toulouse (France), p. 16-17, avril 1986.
30. E. Sicard - "Contribution au Development D'idees de Conception Intelligentede Circuits Intégrés VLSI" - Thèse du Doctorat de L'Université Paul Sebatier de Toulouse, cap. 2, pp. 66-70, juillet 1987.
31. J. Valdes; R. L. Kalin - "ALI2 Documentation and Implementation Guide: Language Overview" - VLSI Memo #8, Princeton University, feb. 1983.
32. G. Vijayan - "Design, Implementation and Theory of a VLSI Layout Language" - Ph. D. Thesis, Princeton University, aug. 1983.
33. S. North - "Molding Clay: A Manual for the CLAY Layout Language" - VLSI Memo #3, Princeton University, july 1983.
34. L. Rijnders; P. Six; H. J. De Man - "Design of a Process-Tolerant Cell Library for Regular Structures Using Symbolic Layout and Hierarchical Compaction - IEEE Journal of Solid-State Circuits, vol. 23, n. 3, june 1988.

35. K. C. Chu; R. Sharma - "A Technology Independent MOS Multiplier Generator" - ACM IEE 21st Design Automation Conference, Proceedings, 1984 - june 25-27, pp. 90-97.
36. L. A. Glasse; L. P. J. Hoyte - "Delay and Power Optimization in VLSI Circuits" - ACM IEE 21st Design Automation Conference, Proceedings, 1984 - june 25-27, pp. 529-535.
37. H. H. Berger - "Contact Resistance and Contact Resistivity" - J. of the Electrochemical Society, vol. 119, n. 4, p. 507-514, april, 1972.
38. P. C. Bezerra - "SPA-D: Um Sistema de Projetos Automatizados de Circuitos Integrados Digitais" - Tese de Doutorado, FEC/UNICAMP, nov, 1980.

## 6 APÊNDICE

A seguir serão abordados os parâmetros elétricos dependentes da geometria no transistor MOS. As expressões apresentadas são as utilizadas no simulador elétrico SPICE2G, no modelo 3 para dispositivos MOS, o qual incorpora efeitos de segunda ordem (canal curto) e é bastante utilizado mundialmente.

Primeiramente será fornecida uma descrição dos símbolos frequentemente utilizados nas expressões de modelamento do transistor MOS.

- $\epsilon_0$  = permissividade absoluta do espaço livre (F/m)
- $\epsilon_{ox}$  = permissividade relativa do óxido (F/m)
- $\epsilon_{si}$  = permissividade relativa do silício (F/m)
- $T_{ox}$  = espessura do óxido de porta ( $\mu\text{m}$ )
- $T_{oxc}$  = espessura do óxido de campo ( $\mu\text{m}$ )
- $W$  = largura de canal do transistor ( $\mu\text{m}$ )
- $L$  = comprimento de canal do transistor ( $\mu\text{m}$ )
- $K$  = constante de Boltzmann ( $1.38 * 10^{-23} J.K^{-1}$ )
- $T$  = temperatura (K)
- $q$  = carga do elétron ( $1.6 * 10^{-19} C$ )
- $N_{SUB}$  = dopagem do substrato (átomos/  $cm^3$ )
- $V_{ds}$  = tensão dreno-fonte (V)
- $V_{gs}$  = tensão porta-fonte (V)
- $V_{bs}$  = tensão substrato-fonte (V)
- $\phi_f$  = potencial do nível de Fermi (V)
- $Q_{ss}$  = densidade de carga de superfície no óxido ( $C/m^2$ )
- $\phi_{ms}$  = função de trabalho metal/semicondutor (V)

As expressões de modelo são as seguintes:

- tensão de limiar

No SPICE2G, modelo 3, a tensão de limiar é dada por:

$$V_{th} = V_{fb} + 2\varphi_f - \sigma V_{ds} + \gamma F_s \sqrt{2\varphi_f - V_{bs}} + F_n(2\varphi_f - V_{bs}),$$

onde a tensão de banda plana,

$$V_{fb} = \Phi_{ms} - Q_{ss}/C_{ox}; C_{ox} = \epsilon_{ox}/T_{ox}$$

$\sigma = ETA * A / (C_{ox} * L^3)$ ;  $\sigma$  = coeficiente de "feedback" eletrostático

A = constante empírica =  $8.15 * 10^{-22} (m^3/F)$

$$F_s = 1 - X_j/L [(LD+W_c)/X_j (\sqrt{1 - [(W_p/X_j)/(1 + W_p/X_j)]^2} - LD/X_j)];$$

sendo  $F_s$  o fator de correção de efeito de canal curto

$$W_c/X_j = d_0 + d_1(W_p/X_j) + d_2(W_p/X_j)^2;$$

$d_0$ ,  $d_1$  e  $d_2$  são constantes empíricas adimensionais de valores:

$$d_0 = 0,0631353$$

$$d_1 = 0,8013292$$

$$d_2 = -0,01110777;$$

$W_c$  = largura da camada de depleção da junção cilíndrica ( $\mu m$ )

$W_p$  = largura da camada de depleção da junção plana ( $\mu m$ )

$X_j$  = profundidade de junção ( $\mu m$ )

$$F_n = \Delta W V_{th} = DELTA * [(\pi * X_j^2)/4C_{ox}W] * (2\varphi_f - V_{bs});$$

sendo  $F_n$  o fator de correção de efeito de canal estreito

$X_d$  = coeficiente de largura de camada de depleção =  $\sqrt{2\epsilon_{si}/qN_{sub}}$

$$\gamma = \sqrt{(2q * \epsilon_{si} * N_{sub})/C_{ox}}$$

$$C_{ox} = \epsilon_{ox}/T_{ox}$$

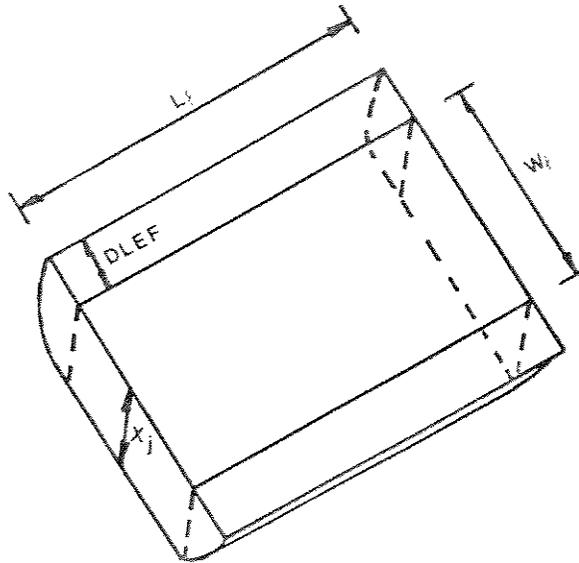


Figura 19: Esboço de Uma Fita de Difusão de Comprimento  $L_f$  e Largura  $W_f$

DELTA e ETA são parâmetros SPICE empíricos, cujos valores são obtidos por ajuste de curvas. Representam a influência em  $V_{th}$  devido ao estreitamento e encurtamento de canal, respectivamente.

Portanto, em transistores MOS de canal curto, os parâmetros de geometria que podem ser controlados pelo projetista e influenciam  $V_{th}$  são  $W$  e  $L$ . Os demais parâmetros dependem exclusivamente do processo, sendo portanto constantes para um mesmo processo.

- Resistências das Difusões de Fonte/Dreno

A resistência de uma fita de difusão equivalente a da figura 19 é dada por:

$$R = R_{\square} * L_f(W_f + 2 * DLEF) , \text{ (ref. 8) , onde}$$

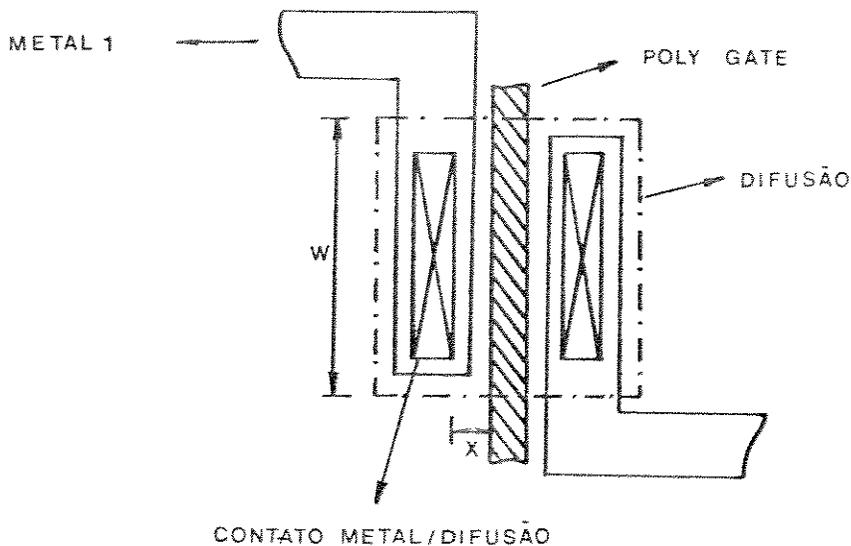


Figura 20: *Layout* de um Transistor MOS

$DLEF$  = difusão lateral efetiva ( $\mu\text{m}$ )

$L_f$  = comprimento da fita de difusão ( $\mu\text{m}$ )

$W_f$  = largura da fita de difusão ( $\mu\text{m}$ )

$R_{\square}$  = resistência de folha por  $\square$  de difusão ( $\Omega/\square$ )

onde  $L_f$  e  $W_f$  são dependentes do layout. Os demais parâmetros dependem exclusivamente do processo.

Calcularemos em seguida, a resistência equivalente das difusões de dreno e fonte de um transistor. Assim, para um transistor como o mostrado na figura 20.

podemos dizer que, aproximadamente:

$$R_{\text{fonte}} = R_c + R_{\square} (X + LD)/(W + 2 * DLEF); \text{ onde}$$

$LD$  = difusão lateral sob o gate

$R_c$  = resistência do contato

$W$  = largura de canal do transistor (máscara de difusão)

$X$  = distância contato/poly-gate, que constitui uma aproximação do caminho percorrido pela corrente da difusão de fonte até o canal.

Na equação anterior para  $R_{fonte}$  foi incluído o aumento na resistência da camada difundida, devido à difusão lateral sob o gate, através do uso do parâmetro  $LD$ . A equação para  $R_{dreno}$  é análoga à de  $R_{fonte}$ .

$R_c$  é dado por (ref.8, 38):

$$R_c = (\rho_c / L_c * d) * \alpha * d * \cotgh(\alpha d),$$

para contato do tipo horizontal, que é o caso dos contatos de F/D, onde:

$$\alpha = \sqrt{R_s / \rho_c};$$

$R_s$  = resistência de folha do semiconductor ( $\Omega$ )

$\rho_c$  = resistividade do contato ( $\Omega.cm^2$ )

$L_c$  = largura do contato ( $\mu m$ )

$d$  = comprimento do contato ( $\mu m$ )

$\rho_c$  pode ser determinado em função da resistividade média da camada difundida. Os parâmetros  $X$ ,  $W$ ,  $L_c$  e  $d$  dependem do lay-out. Os demais são dependentes exclusivamente do processo.

- Capacitâncias de overlap

As capacitâncias de *overlap* no SPICE2G são modeladas por:

– porta/dreno (CGDO) e porta/fonte (CGSO)

$$CGSO = CGDO = (\epsilon * \epsilon_0 * LD * W) / T_{ox}$$

onde:

$LD$  = difusão lateral sob o gate ( $\mu m$ )

– porta/bulk (CGBO)

$$CGBO = (\epsilon_r * \epsilon_0 * L * OPGC) / T_{oxc};$$

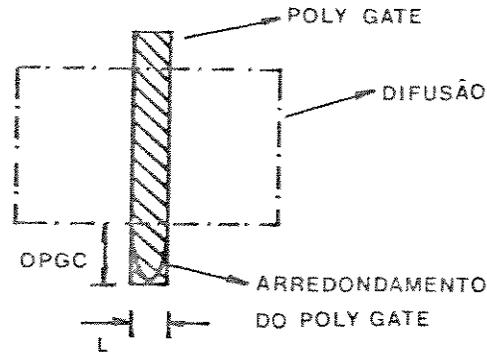


Figura 21: Arredondamento nas Bordas do Poly-gate

onde:

OPGC = sobreposição do poly-gate sobre o canal ( $\mu\text{m}$ )

Os parâmetros do layout que exercem influência nas capacitâncias de overlap no transistor MOS são:  $W$ ,  $L$  e OPGC. O parâmetro OPGC, entretanto, não pode ser alterado a critério do projetista, visto que seu valor mínimo constitui uma regra de projeto estabelecida pelo fabricante. Esta é dependente da resolução do equipamento utilizado pelo mesmo e do erro de alinhamento, e seu objetivo é garantir uma completa cobertura do canal pelo poly-gate, já que este sofre um arredondamento nas bordas, conforme mostrado na figura 21.

Portanto, a capacitância de overlap porta/substrato é algo que o projetista deseja minimizar, mas que normalmente já está reduzida a seu valor mínimo, visto que OPGC mínimo é invariante (definido pelas r.p.'s do fabricante) e  $L$ , embora normalmente utilizado também é o mínimo permitido pela tecnologia, é um parâmetro que pode ser ajustado pelo projetista.

Os demais parâmetros são dependentes exclusivamente do processo.

- Capacitância do Óxido de Porta

$$C_{ox} = (\epsilon_0 * \epsilon_r * A) / T_{ox} = (\epsilon_0 * \epsilon_r * W * L) / T_{ox}$$

As variáveis de lay-out, sobre as quais o projetista tem controle são W e L. As demais dependem apenas do processo.

- Capacitâncias de *Bottom* das Junções Dreno/Substrato (CBD) e Fonte/Sub (CBS)

A capacitância de fundo das junções de fonte/substrato (CBS) e dreno/subs (CBD), podem ser escritas como:

– polarização reversa ( $V_{bs} \leq FC * PB$ )

$$CBS_b = C_{Js} * AS / [1 - (V_{bs}/PB)]^{MJ}$$

– polarização direta ( $V_{bs} > FC * PB$ )

$$CBS_b = C_{Js} * [AS / (1 - FC)^{1+MJ}] * [1 - FC * (1 + MJ) + (V_{bs}/PB) * MJ]$$

As expressões para  $CBD_b$  são análogas às anteriores, substituindo-se AS por AD.

Nas expressões acima, temos:

$C_{Js}$  = capacitância de fundo de junção / unidade de área de junção ( $F/m^2$ )

AS = área de fundo da junção de fonte ( $m^2$ )

AD = área de fundo da junção de dreno ( $m^2$ )

$V_{bs}$  = tensão de polarização *bulk-source* (V)

$PB$  = potencial de junção (V)

$MJ$  = coeficiente de graduação da junção fonte/substrato de fundo

$FC$  = coeficiente de polarização direta da capacitância de junção não ideal

As expressões são análogas para CBD<sub>b</sub>.

Os únicos parâmetros dependentes do lay-out são AS e AD. Os demais só dependem do processo.

- Capacitâncias das paredes laterais("Sidewall") das Junções Dreno/ Substrato e Fonte/Substrato

Essas capacitâncias podem ser modeladas pelas expressões:

- polarização reversa ( $V_{bs} \leq FC * PB$ )

$$CBS_{sw} = CJSW_s * PS / [1 - V_{bs}/PB]^{MJ_{SW}}$$

- polarização direta ( $V_{bs} \geq FC * PB$ )

$$CBS_{sw} = CJSW_s * PS / [1 - FC]^{1+MJ_{SW}} * [1 - FC(1 + MJ_{SW}) + (V_{bs}/PB) * MJ_{SW}]$$

onde:

$CJSW_s$  = capacitância de borda da junção de fonte / unidade de perímetro de junção ( $F/m$ )

$PS$  = perímetro da junção de fonte ( $m$ )

$MJ_{SW}$  = coeficiente de graduação da junção fonte / substrato de borda

$FC, PB, V_{bs}$  definidos no item anterior

As expressões são análogas para CBD<sub>sw</sub>. Os parâmetros dependentes do lay-out, que podem ser controlados pelo projetista são PS e PD. Os demais dependem do processo.

- Corrente Reversa das Junções de Fonte (IS) e Dreno (ID)

A corrente reversa das junções de fonte e dreno podem ser expressas por:

$$IS = JS \exp[V_{bs}/(kt/q)] - 1 * AS$$

onde:

JS = densidade de corrente reversa de junção

AS = área de junção de fonte ( $m^2$ )

Vbs = tensão de polarização fonte/substrato (V)

A equação para ID é análoga à apresentada acima, substituindo-se JS por JD, AS por AD e Vbs por Vbd.

Os parâmetros de lay-out que influenciam IS e ID são AS e AD, respectivamente. Os demais dependem do processo, exclusivamente.

- Parâmetro Transcondutância Intrínseca

O parâmetro transcondutância intrínseca é dado por:

$$Kp = (W/L) * U0 * Cox = (W/L) * U0 * [\epsilon0 * \epsilonr * W * L] / Tox = [W^2 * U0 * \epsilon0 * \epsilonr] / Tox$$

Portanto, o parâmetro de geometria que controla Kp é W.

Como conclusão deste apêndice, pode-se dizer que praticamente todos os parâmetros elétricos do transistor MOS dependem de W e L, já que as demais dimensões do transistor são normalmente reduzidas ao mínimo permitido pelas regras de projeto, isto é, são mantidas fixas.

O mesmo vale para os demais elementos de circuito (resistores, capacitores) que, por serem mais simples, têm como únicos parâmetros de geometria que exercem influência nas características elétricas, suas larguras e comprimentos.

Como exemplo de aplicação da parametrização para uma célula eletricamente flexível, podemos citar um buffer de saída parametrizado em função das larguras de canal (W) dos transistores que o compoem, com o objetivo de fornecer

correntes compatíveis com o fan-out requerido em cada projeto específico. Isto significa que o mesmo buffer (isto é, o código correspondente ao mesmo layout) pode ser redimensionado para tornar-se compatível com cargas MOS ou TTL. Assim, a corrente de saída torna-se um parâmetro elétrico que pode ser controlado por intermédio do parâmetro geométrico  $W$ . Uma vez que a corrente pode ser alterável, é importante que as larguras das linhas de metal que fazem as ligações com VDD, GND e saída do buffer também o sejam, de modo a satisfazerem sempre a densidade de corrente especificada pelo fabricante. Assim, as larguras destas linhas de metal também devem ser tomadas como parâmetros na descrição do layout.