

A. A. SUZIM

SUMÁRIO

Este artigo apresenta o trabalho que vem sendo feito para construir um montador automático de blocos funcionais para concepção de circuitos integrados. Primeiramente um modelo geral do circuito é apresentado e, em seguida são definidos os blocos funcionais para os quais são programados os procedimentos geradores. A nova estratégia consiste em encapsular a descrição (gráfica e elétrica) das células numa subrotina geratriz que é chamada pelo usuário. O problema da parametrização das células, difícil de resolver no sistema gráfico pode ser facilmente resolvido pela passagem de opções do usuário através dos parâmetros da subrotina. Isto é comparado com um trabalho anterior que usava o conceito de biblioteca de células.

ABSTRACT

This paper presents the work been done to construct an automatic block assembler for integrated circuits design. Firstly a general model of the circuit is presented then some functional blocks are defined to which building procedures are programmed. The new strategy consists of encapsulating the cells' description (graphical and electrical) in generating procedure called by the user. The problem of parameterisation of cells, difficult to solve in a graphical environment can easily be solved by user options passed as procedure parameters. This is compared with a former work that used the cell library concept.

Curso de Pós-Graduação em Ciência da Computação

Universidade Federal do Rio Grande do Sul

90.000 - PORTO ALEGRE - RS

I - INTRODUÇÃO

A necessidade de ferramentas adequadas para a realização de circuitos integrados complexos tem sido objeto de uma grande quantidade de estudos, trabalhos e publicações nos últimos tempos. Os investimentos que os governos e as indústrias fazem atualmente na área de microeletrônica mostra que a sociedade está apostando nesta nova tecnologia. De fato, as implicações sociais da informatização (sobre as condições de trabalho, modo de vida, nível de emprego, etc.) não podem deixar ninguém indiferente. E a informatização está aliçada sobre a microeletrônica.

As ferramentas adequadas à realização de circuitos integrados complexos estão baseadas em metodologias que tentam automatizar o mais possível a concepção dos circuitos: partindo de uma descrição do problema utilizando conceitos de uma linguagem de alto nível, fases sucessivas de refinamento são aplicadas, decompondo-se os conceitos complexos em elementos cada vez mais simples até chegar ao nível dos elementos materiais. Quando do desenho dos primeiros circuitos integrados, um trabalho artesanal era satisfatório; o degrau seguinte da complexidade pode ser superado apenas com recursos computacionais gráficos. Hoje o trabalho se concentra na pesquisa de novas metodologias.

A Microeletrônica é uma atividade multidisciplinar. Hoje ela incorpora novas disciplinas desde a informática e sistemas até a área de equipamentos como laser, plasma, raios X, etc. Em consequência investimentos cada vez maiores são necessários (sempre contando com um retorno compensador).

Em um trabalho anterior [SUZ 81a, SUZ81b] uma metodologia de geração de partes operativas de circuitos integrados NMOS, assim como um conjunto inicial de células e o protótipo do programa de montagem haviam sido apresentados. A experiência com aquele sistema permitiu validar a metodologia e detectar alguns problemas práticos, como a proliferação de pequenas células. Um conjunto muito grande de células é difícil de manter, de caracterizar e, certamente, de usar. As células armazenadas em uma biblioteca são rígidas, pois o nosso sistema gráfico não permite parametrização.

Este sistema utiliza uma linguagem de programação de alto nível (le-lisp, no caso). As descrições das células são sequenciais de retângulos codificados dentro do próprio programa. São em geral constantes, exceto os pontos que o projetista da célula pretende deixar como parametrizáveis. Estes serão calculados em tempo de execução (variáveis ou expressões do programa).

Um fato novo é que o projetista da célula deverá desenhá-la, descrevê-la e programar o algoritmo de montagem, de tal forma que o usuário veja apenas a função e não se preocupe com a montagem mas apenas com a especificação.

Uma característica de Lisp que facilita este tipo de trabalho é a possibilidade para tra

tar os parâmetros muitas vezes em quantidade variável, transmitidos dentro de uma lista.

Ao mesmo tempo em que esta estratégia é desenvolvida, o trabalho é utilizado para a definição de um conjunto de blocos funcionais na tecnologia CMOS porta de silício e dois níveis de metal. Além dos aspectos topológicos, com características bem diferentes do primeiro trabalho (NMOS), aspectos do comportamento elétrico são tratados.

Finalmente, a essência do problema de concepção de um circuito é a passagem da descrição comportamental de um sistema para o domínio físico geométrico [CON84].

II - MODELO DO CIRCUITO

Ao se propor uma construção automática dos circuitos tem-se em mente um modelo de máquina que supostamente satisfaz a uma classe de problemas. Com esse modelo em vista são propostas primitivas que permitam descrever uma solução para o problema. A descrição do problema continuará sendo uma operação delicada, uma vez que exige um perfeito conhecimento do problema e da semântica das primitivas.

O modelo proposto nas referências citadas apresenta o circuito como composto de dois grandes blocos: bloco de controle e bloco operacional. O primeiro é responsável pelo sequenciamento das operações e o segundo, pela execução. A comunicação entre os blocos se faz por meio de um interface responsável pela adaptação de níveis elétricos, pela conexão e, algumas vezes, pela introdução de tempos (fases). Os sinais que fluem da parte (ou bloco) de controle (PC) para a parte operativa (PO) são chamados em geral de comandos e os que fluem no sentido inverso, de predicados. Existem situações particulares em que essa denominação pode parecer não muito adequada. Microprocessadores do tipo bit-slice (como a série 2900) são exemplos sugestivos da divisão entre PO e PC. Não se deve por outro lado associar o conceito de microprogramação com o modelo proposto mesmo que não raro os dois conceitos casem muito bem. (Fig.1)

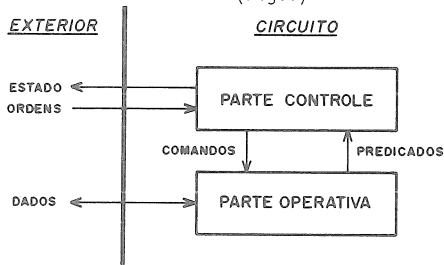


Figura 1. Modelo PC-PO de um circuito ou sistema

Vale ainda mencionar que o modelo se adapta ao conceito de hierarquização da descrição. Em um nível qualquer de trabalho, podem ser usadas primitivas cuja implementação direta seja inviável, sabendo-se que numa fase posterior essa mesma primitiva será detalhada (interpretada). Além disso é possível que uma parte de controle controle mais do que uma parte de operativa e, ainda, que uma parte operativa seja ela mesma uma máquina e, conse-

quentemente, decomponível (por resultado de interpretação) em uma parte controle e uma parte operativa, etc.

III - PARTE OPERATIVA

A parte operativa é caracterizada por conter os elementos de memorização, transformação e comunicação de informações, respectivamente memórias, operadores e barramentos. A necessidade das memórias e dos operadores é evidente. Os barramentos são em geral empregados nos circuitos mais complexos mas não são essenciais à P0. A faculdade de comunicar ou transportar os dados é importante mas pode ser feita sem barramentos.

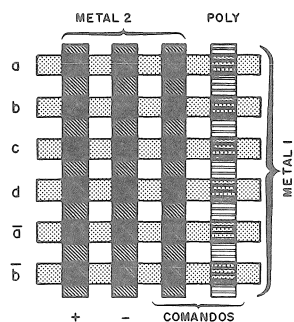
3.1 Os Barramentos

Utiliza-se barramentos quando é proibitivo ou inútil conectar diretamente todos os elementos que devem comunicar entre si. Vendo de outro lado, os barramentos são canais de comunicação partilhados no tempo e sugerem Partes Operativas bem estruturadas construídas ao redor de um sistema de protocolo bem definido. No caso de circuitos integrados, mesmos as características geo-métricas são fixadas (normalizadas). O conjunto de características do barramento define o sistema de barramento (junto a uma família de células). A escolha de um sistema de barramento é, pois, algo bastante delicado. Obviamente existem células que são mais ou menos comprometidas com o sistema de barramento, ou seja, algumas células só operam com um determinado sistema de barramento (sequência de tempos ou níveis lógicos) enquanto outras podem admitir mais de uma forma de operação.

Entre as células mais sensíveis ao tipo de barramento estão as memórias e, muitas vezes, são exatamente essas as primeiras células projetadas dando origem ao sistema de barramento. A primeira proposição, tendo em vista a existência de dois níveis de metal era de reduzir ao máximo a altura do barramento, por exemplo, levando a alimentação em metal dois e deixando o metal 1 para os dados (o interesse de se trabalhar com dois barramentos bifilares foi apresentado nas referências). O fato de ser CMOS sugeriu um re-estudo inclusive da utilidade de haver pré-carga e barramento complementar. Por tentativa se chegou a conclusão que a situação ótima para uma parte operativa que tenha um percentual significativo de sua área ocupado por circuito de memória é a mostrada na figura 2.

É praticamente impossível desenhar uma célula de memória CMOS com as regras de projeto disponíveis dentro da altura de seis passos de metal, mantendo uma forma interessante para a montagem do bloco de memória. Além disso a grande quantidade de interconexões necessárias em CMOS sugeriu a reserva de um dos níveis de metal para conexões locais e, se necessário, para controles, e o outro nível para os dados, alimentação e barramentos reserva. O sistema de barramento resultante é apresentado na figura 3.

O comportamento elétrico do barramento e as características elétricas da memória são mostrados na figura 4.



VERSÃO INICIAL

Figura 2. Barramento da P.O. CMOS

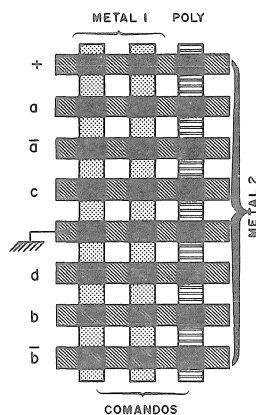


Figura 3. Forma Final do Barramento

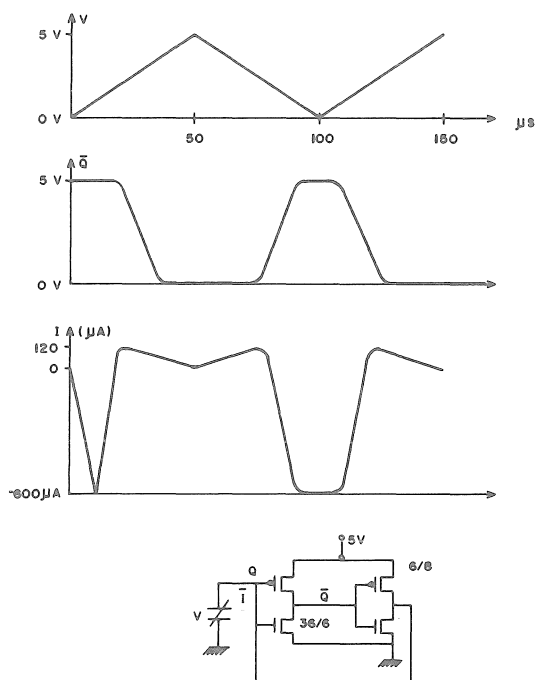


Figura 4. Características da Memória

Dos dois tipos de memória (ROM e RAM) o primeiro em geral é fácil de implantar. Quanto às memórias de leitura e escrita podemos ainda distinguir entre as memórias propriamente ditas que são pensadas para constituírem blocos de memória - caso em que conta muito a otimização de superfície e consumo elétrico - e as memórias isoladas constituindo em geral um registrador especializado onde conta a maleabilidade para adaptar-se a diferentes situações topológicas ou dos sinais de controle. Esses comentários valem obviamente para o contexto deste trabalho ou seja para o desenvolvimento da parte operativa do circuito.

As memórias de leitura/escrita destinadas a "uso geral" dentro dos circuitos podem ser colocadas em bloco se não houver conflito de acesso. Como mencionado acima buscar-se-á otimização em área e consumo. Para NMOS, o esquema da memória com dois inversores realimentados é uma solução perfeitamente adaptada. Tem-se um ponto a seis transistores (ou quatro transistores e dois resistores) e o elemento de carga do inversor tem função apenas de compensar a corrente de fuga. No momento da seleção em escrita no ponto de memória podemos imaginar um circuito equivalente a um NÁOU. Em CMOS o comportamento é diferente já que não existe o elemento de carga: tudo se passa como se as funções fossem calculadas duas vezes e para cada porta lógica de n variáveis, $2 \times n$ transistores são utilizados. Jogando com as características geométricas dos transistores é possível trabalhar com a mesma célula como em NMOS (6 transistores). Naturalmente uma das grandes características do CMOS fica comprometida: seu baixo consumo. Técnicas como chaveamento da fonte de alimentação podem ser utilizados mas não serão tratadas aqui (figura 4).

Outros pontos de memória necessários na parte operativa do circuito muitas vezes exigem que se guarde uma informação gerada por uma função (incrementador, por exemplo) cuja saída, se se desejasse complementar, seria necessário acrescentar mais lógica.

Um circuito interessante, apesar de exigir o comando de escrita e seu complemento, é mostrado na figura 5.

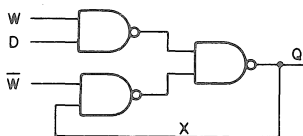


Figura 5. FF D Usado nos Registradores

O circuito é na realidade um multiplexador de duas entradas e os estudantes que desenharam uma PO CMOS utilizaram esta mesma célula para outros elementos como, por exemplo, para realizar lógica; pelo fato de ser utilizada para finalidades diversas a célula foi também chamada de célula mãe.

Essa mesma célula é utilizada como multiplexador simplesmente eliminando o laço de realimentação.

3.3 Os Operadores

Os operadores são circuitos combinacionais que realizam uma função de transformação sobre um ou mais operadores. São funções típicas: incremento, soma, deslocamento, ou, etc. Alguns operadores podem realizar várias funções e um parâmetro escolhe qual delas deve ser realizada num dado momento (caso típico das Unidades Aritméticas e Lógicas). Existem também operadores que podem realizar mais do que uma função ao mesmo tempo: são os circuitos de múltiplas saídas.

Construídos sobre um sistema de barramento, os operadores de uma Parte Operativa tem acesso fácil aos operandos; não há necessidade de "routing". As questões que se colocam são quanto ao tempo necessário para as operações, quanto ao número de registradores e se estes estão na entrada ou na saída do operador e principalmente, quais as operações que determinado operador deve realizar.

Sobre este último item, é conhecido que existe uma relação entre complexidade e área. Portanto, não há interesse em se dispor do mais complexo operador capaz de realizar um conjunto completo de operações. É necessário que se possa construir o menor operador possível capaz de realizar as operações necessárias para uma determinada aplicação. A dificuldade maior está em construir automaticamente este operador sabendo que cada aplicação tem seu conjunto de operações. Ocasionalmente há em que se pode implantar uma Unidade Aritmética e Lógica (UAL) Completa, outras há em que se escolhe um operador simples e acrescenta-se outro, feito especialmente para o circuito uma vez que a função não é frequentemente utilizada. Os operadores que estão ligados ao mesmo barramento podem trabalhar simultaneamente.

4. Montagem

No "lay-out" da memória apresentado na figura 4, aparecem dois bits de duas palavras contíguas e também as conexões com os barramentos. Esses desenhos foram feitos para uma minimização da utilização da área. Entre o desenho da célula básica e a montagem de um bloco de memória existe bastante trabalho. Este trabalho pode ser feito automaticamente desde que se programe um procedimento que permita qualquer combinação de elementos de memória. Por exemplo montar um número ímpar de palavras ou iniciar com um padrão especificado como: a primeira célula não deve ter contatos com o barramento, ou o contrário, deve

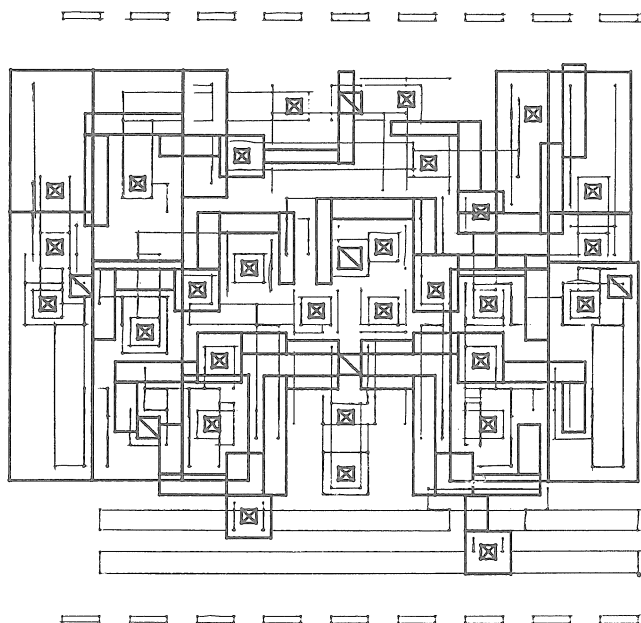


Figura 6. Multiplexador

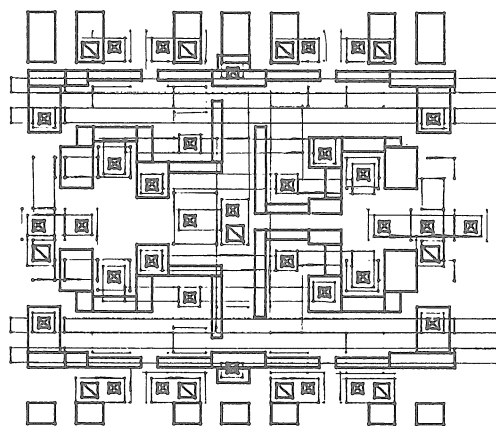


Figura 7. Memória

A descrição de um bloco pode ser extensa e existem algoritmos cuja complexidade exige um trabalho grande. Como exemplo do primeiro caso podemos citar a unidade aritmética e lógica (incluindo o circuito de propagação de transbordo) e como exemplo da segunda classe de problemas pode-se mencionar os programas geradores de PLA, (incluindo-se os algoritmos de otimização lógica e topológica).

A descrição pode ter uma forma como o que segue (em Lisp):

```
(Setq l1 '(
  (5 12 4 7 d)
  (5 27 4 7 d)
```

...

onde os números entre parêntesis definem um retângulo e d é uma variável - o nível no caso. As variáveis podem corresponder a dimensões dos retângulos ou a seu posicionamento. É com esse sentido limitado que se entende a parametrização das células. Conclui-se que:

- Todas as formas possíveis de uma célula devem ser previstas por quem a projeta.
- A parametrização deve ser minimizada - só será possível alterar dentro de um campo limitado de valores e
- O programa de montagem deve fazer a consistência desses valores.

Existem casos em que é preferível codificar não numa lista de retângulos, mas em listas de cada uma das coordenadas - x ou y - e das dimensões dos mesmos. No caso de elementos (retângulos), de mesmo tamanho, a lista pode conter apenas um elemento. O próximo elemento de uma lista é dado por:

```
(de prox (l1)
  (cond ((cdr l1)) (l1))) .
```

Embora pareça evidente que este método de trabalho é interessante, só a experiência poderá dizer quais são as limitações, os casos típicos e as excessões. É até possível que o modelo deva ser refinado, quiçá expandido.

Finalmente, é importante salientar que, uma vez caracterizado um bloco, todas as informações pertinentes podem ser integradas na própria descrição. É assim que os níveis superiores em abrangência poderão inquirir sobre propriedades dos blocos funcionais. Um sistema de avaliação de desempenho, por exemplos, poderá obter de cada bloco o tempo de atraso desse bloco, fornecidos os parâmetros necessários. A modelização é, nesse sentido, um trabalho iterativo, pelo menos na fase inicial. A hierarquia criada admite ramificações e agregações. Este trabalho pretende caracterizar o nível mais baixo da hierarquia: a descrição dos objetos mais elementares sob o ponto de vista funcional.

CONCLUSÃO

Na tentativa de harmonizar as descrições das células dos circuitos integrados com os níveis de descrição mais altos e de disciplinar seu armazenamento e montagem, sugere-se que essa descrição contenha não só a listagem dos elementos geométricos mas também o algoritmo de montagem. Algo como um código genético que permite montar a estrutura. Não só as informações geométricas são gerenciadas, mas também as informações elétricas como consumo, retardo, impedância, etc. O sistema deve ter capacidade de extrapolar para o bloco funcional as características das células. Isso implica inclusive em criar modelos para um conjunto de células (bloco). Todo esse conjunto de dados e algoritmos está dentro de um programa especializado em gerar um determinado bloco funcional.

A etapa seguinte será dedicada à elaboração das operações entre esses objetos. Uma linguagem de montagem de células já foi definida na referência [SUZ 84b]. Uma evolução desta linguagem que permita a utilização de parâmetros para a definição de uma célula e, por outro lado, recupere informações desta mesma célula é, certamente muito interessante e difícil.

AGRADECIMENTOS

Gostaríamos de agradecer ao Prof. Bernard Courtois e à sua equipe, em particular a Mohamed Jerraya. Num curto espaço de tempo foi possível iniciar a organização destas ideias que nos perseguiam há tempo. O intercâmbio faz parte do convênio CAPES-COFECUB. Esse trabalho se situa no contexto do laboratório de microeletrônica do PGCC/UFRGS apoiado por Convênio FINEP e por bolsas e auxílios individuais de pesquisa do CNPq.

BIBLIOGRAFIA

- [SUZ 81a] SUZIM Altamiro Amadeu "Data Processing Section for Microprocessor - "Like Integrated Circuits" IEEE Journal of Solid-State Circuits Vol. SC-16 n° 3 JUN/81 pp. 233-235
- [SUZ 81b] SUZIM, Altamiro Amadeu "Etude des Paramètres Operatives a Elements Modulaires para Processeurs Monolithiques. Thèse Docteur-Ingénieur INP - Grenoble - NOV 1981.
- [CON 84] CONNIE U. Smith and DALLAN John A. Future Directions for VLSI and software Engineering. in VLSI ENGINEERING Edited by G. GOOS and J. HARTMANIS. Springer-Verlag 1984.
- BRUNE, Osmar; CALAZANS, Ney; DILENBURG, Renato & PINHEIRO, Dêrcio. Parte Operativa CMOS. Relatório em preparação.