

Boletim Técnico da Escola Politécnica da USP

Departamento de Engenharia Eletrônica

ISSN 1413-2206

BT/PEE/9825

**DECMEF: Um Sistema de
Decomposição Aplicado à Síntese de
Máquinas de Estados Finitos**

**Carlos Humberto Llanos Quintero
Marius Strum**

São Paulo - 1998

O presente trabalho é um resumo da tese de doutorado apresentada por Carlos Humberto Llanos Quintero, sob orientação do Prof. Dr. Marius Strum.: "DECMEF: Um Sistema Aplicado à Síntese de Máquinas de Estados Finitos Complexas", defendida em maio de 1998, na Escola Politécnica.

A íntegra da tese encontra-se à disposição com o autor e na Biblioteca de Engenharia de Eletricidade da Escola Politécnica/USP.

FICHA CATALOGRÁFICA

Llanos Quintero, Carlos Humberto

DECMEF : um sistema de decomposição aplicado à síntese de máquinas de estados finitos / C.H. Llanos Quintero, M. Strum. -- São Paulo : EPUSP, 1998.

p. -- (Boletim Técnico da Escola Politécnica da USP, Departamento de Engenharia Eletrônica, BT/PEE/9825)

1. Máquinas de estados finitos 2. Síntese lógica 3. Decomposição lógica I. Strum, Marius II. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Eletrônica III. Título IV. Série
ISSN 1413-2206

CDD 004.16

004.22

004.16

DECMEF: Um Sistema de Decomposição Aplicado à Síntese de Máquinas de Estados Finitos

Carlos Humberto Llanos Quintero (*)
Marius Strum ()**

ABSTRACT

This work describes the DECMEF system used to decompose a complex finite state machines (FSMs) (between 100 to 1000 states). FSM decomposition is concerned with the implementation of a FSM as a set of smaller interacting submachines (sub-FSMs). In order to do this it is necessary to obtain the state partitions of the original FSM and the product of the partitions must be the zero-partition.

The decomposition technique is desirable for a number of reasons. The set of smaller machines leads to improved performance as a reduction in the critical path of the circuit, so that the decomposed machine can be clocked faster than the prototype machine. Otherwise the decomposition technique can lead to reduce the final area of the circuit. Partitioning of the logic implementation of the FSM could help the floorplanning task by a simplification of the layout constraint of the circuit resulting in smaller circuit area. FSM decomposition can be applied directly when FPGAs or PLDs are the target technology. A very important point is that the decomposition technique allows logic synthesis systems to find solutions for problems otherwise unsolved (large FSMs).

Several FSM decompositions methods have been proposed but no important results have been achieved for large (complex) FSMs. For instance, there are results for a benchmark with 121 states, 27 inputs and 54 outputs.

In this paper we present the DECMEF software system, a set of tools to interactively decompose large FSMs. The decomposed FSMs are further synthesized with the SIS system. Results for FSMs with up to 1000 states are presented.

The DECMEF system has two main tools: the GERPAR and the GERTAB. The first obtain two partitions so that its product is the zero-partitions. Two state partitioning techniques were implemented in the GERPAR: state factorization and state clustering. These techniques can be used stand alone or in sequence in order to obtain the partitions.

The GERTAB tool determines the state transitions tables for each sub-MSF. This tool is composed by GERTAB1 and GERTAB2. The GERTAB1 tries to improve the implementation by a reduction on the number of interconnecting wires which must exist when the FSM is decomposed. The GERTAB2 tries to improve the circuit by a reduction of the output function in each sub-MEF.

(*) estudante de doutorado, LME (EPUSP)

() professor associado, LME (EPUSP).**

1. INTRODUÇÃO

Máquinas de estados finitos (MEFs) é um modelo que pode ser usado no projeto de unidades de controle para processadores ou microprocessadores. Dentro do ambiente de compilação de Silício duas soluções são dadas para a geração da unidade de controle: geração por sistemas de síntese lógica e geração usando arquiteturas pré-definidas (geralmente baseadas em ROM). As unidades do primeiro tipo são denominadas "de arquitetura fixa" (*hardwired controllers*), enquanto as do segundo tipo são denominadas "de arquitetura micro-programadas" (*micro-programmed controllers*).

O projeto de MEFs fixas é realizado através de ferramentas de síntese lógica comerciais como por exemplo as da Synopsys [Syn95]), ou acadêmicas tais como o sistema SIS de Berkeley [Sis92], ou ainda o sistema ASYL do CSI de Grenoble na França [Sau87]). Nestes sistemas o comportamento da MEF é descrito num formato apropriado, por exemplo KIS/KISS [Sis92], e, em seguida, são usadas diferentes ferramentas de CAD que resolvem as tarefas de síntese (codificação de estados), otimização lógica (minimização de estados, minimização lógica etc.) e mapeamento tecnológico. Ferramentas de síntese lógica são, entretanto, ineficientes para gerar MEFs de grande complexidade. A complexidade de uma MEF depende do número de estados, de entradas, de saídas e de transições. Exercícios realizados com o sistema SIS rodando em Spare20 com 64Mb de memória, mostraram que a geração de MEFs com algumas centenas de estados torna-se muito lenta. MEFs com mais do que aproximadamente 300 estados não conseguiram ser geradas pelo sistema SIS.

O projeto de MEFs micro-programadas faz uso de ferramentas de CAD específicas que buscam otimizar diferentes aspectos das arquiteturas pré-definidas da unidade de controle [Sau87].

O objetivo deste trabalho é encontrar soluções para obter unidades de controle do tipo fixas que possam ser usadas para implementar controladores complexos de alta eficiência [Ver89].

Isto é alcançado através do uso da técnica de **decomposição de uma máquina de estados finitos**. Decompor uma MEF consiste em encontrar um conjunto de duas ou mais sub-máquinas (sub-MEFs) que executam a mesma função da MEF original.

Os benefícios desta técnica são:

1. estender o uso de sistemas de síntese lógica para MEFs com muitos estados;
2. gerar MEFs complexas com bom desempenho elétrico (velocidade);
3. gerar MEFs decompostas cuja área pode ser menor do que a área da MEF original, competindo portanto com soluções micro-programadas;
4. permitir colocar partes da unidade de controle e da via de dados, que interagem entre si de maneira forte, o mais perto possível, o que pode trazer benefícios para o desempenho final do circuito [Dev89];
5. pode ser aplicada para o caso de implementações da MEF em FPGAs (*Field Programmable Gate Arrays*) ou PLDs (*Programmable Logic Devices*) dado que estas tecnologias caracterizam-se por um número limitado de portas lógicas e de elementos de memória [Ash92].

Muitos autores desenvolveram técnicas para decompor MEFs complexas. As técnicas propostas e os resultados publicados, que serão detalhados no capítulo 2, indicaram:

1. resultados experimentais para MEFs de no máximo 121 estados;
2. dificuldade em obter bons resultados para uma grande variedade de MEFs;
3. diferentes funções objetivo, tais como área, velocidade, número de interconexões entre as sub-MEFs, etc.

Dada a dificuldade de se encontrar uma solução geral para o problema da decomposição de MEFs de alta complexidade (mais de 300 estados), isto é, que seja eficiente para um grande número de MEFs, nós desenvolvemos uma ferramenta interativa de decomposição de MEFs complexas (DECMEF) usando diferentes técnicas de decomposição. O sistema parte da descrição KISS da MEF original e gera as descrições KISS das duas sub-MEFs obtidas. Desta maneira, o projetista pode testar várias soluções e escolher aquela que melhor preencha suas especificações. O resultado da decomposição é em seguida fornecido ao sistema SIS [Sis92] que produz o esquema lógico de cada sub-MEF. O sistema de síntese de leiaute LAGER gera o leiaute final. Estas três ferramentas integram o compilador (SINMEF) desenvolvido no Laboratório de Microeletrônica (LME-EPUSP).

2. TRABALHOS ANTERIORES EM DECOMPOSIÇÃO

Devadas e colaboradores [Dev89], tratam o problema de decomposição de MEFs encontrando padrões topológicos no grafo de transição de estados (GTE) da MEF (denominados de fatores) o qual é semelhante ao problema de se obter sub-grafos isomorfos num grafo. Isto apresenta uma limitação desta técnica dado que este último problema é referido como NP-completo [Dev89]. Além do anterior, existem *benchmarks* nos quais é impossível se obter fatores (pelas próprias características do grafo), trazendo, como consequência, a impossibilidade de se obter uma decomposição.

Ashar e colaboradores [Ash90], e [Ash92] tratam o problema de decomposição de MEFs resolvendo um problema de cobertura de implicantes primos generalizados (IPGs). O conjunto de IPGs deve ser minimizado com uma técnica semelhante à proposta por Quine-McCluskey [McK56] para o problema de minimizar uma cobertura de cubos. O problema de cobertura é NP-completo [Gar79].

Kukula e colaboradores [Kuk91] usam um minimizador lógico de dois níveis para validar constantemente a qualidade da decomposição e para determinar o conjunto de pares de transições que levam a uma decomposição melhor. Isto, necessariamente, cria problemas computacionais adicionais para esta técnica.

Yang e colaboradores tentam obter uma decomposição onde seja minimizada a interconexão entre sub-MEFs. A área e desempenho da MEF decomposta não é considerada e, portanto, a qualidade da decomposição com respeito a estes itens não pode ser garantida.

Hasan e colaboradores propõem uma técnica que usa um minimizador lógico de dois níveis para avaliar a qualidade da primeira fase do processo, e posteriormente, na segunda fase, cai num problema de particionamento de um grafo (problema NP-completo).

As diferentes técnicas estudadas trabalham com problemas considerados computacionalmente complexos. Isto pode explicar porque o maior exemplo tratado nos cinco trabalhos é uma MEF de 121 estados, 27 entradas e 54 saídas (dado em Ashar e colaboradores [Ash92]).

3. DEFINIÇÕES GERAIS

Nesta seção serão apresentadas as definições referentes a MEFs e à teoria das partições aplicada ao problema de decomposição. Circuitos lógicos sequenciais são frequentemente modelados usando máquinas de estados finitos (MEFs), a seguir será dada a definição formal de MEF (dada na definição 1).

3.1 Conceito de Máquina de estados finitos

A definição 1 fornece o modelo de máquina de estados finitos.

Definição 1 (Máquina de estados finitos- MEF): Uma *máquina de estados finitos* é definida como uma 6-upla $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$, onde Q é um conjunto finito de estados, Σ é um conjunto finito de entradas, Δ é um conjunto finito de saídas, δ é uma função transitiva $Q \times \Sigma \rightarrow Q$, λ é uma função transitiva $Q \times \Sigma \rightarrow \Delta$ (máquina de Mealy) ou $Q \rightarrow \Delta$ (máquina de Moore), e q_0 é um estado inicial [Boo67].

Uma MEF pode ser representada por um grafo direcionado denominado de *grafo de transição de estados* (GTE). Um GTE é definido como um par $G(V, E)$ onde os vértices (V) do grafo correspondem aos estados da MEF.

3.2 Definições básicas da teoria das partições aplicada a MEF

Hartmanis e colaboradores [Har66] apresentam um estudo teórico profundo sobre o conceito de partição aplicado a problemas de MEFs. As definições que apresentamos nesta seção são referidas em Hartmanis e colaboradores [Har66], Abe e colaboradores [Abe92] Kohavi [Koh78] e Booth [Boo67] e serão usadas no problema de decomposição de MEFs.

Definição 2 (cobertura): seja S um conjunto. Uma coleção $A_1, A_2, A_3, \dots, A_n$ ($n \geq 1$) de subconjuntos de S denomina-se *cobertura* de S se $\cup A_{(i)} = S$ para todo $1 \leq i \leq n$.

No caso do conjunto de estados de uma MEF: $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ uma cobertura pode ser dada como: $A_1 = \{s_1, s_2\}$, $A_2 = \{s_1, s_3\}$, $A_3 = \{s_2, s_3, s_4\}$, $A_4 = \{s_5, s_6\}$

Pode se notar que os conjuntos que formam uma cobertura não precisam ser disjuntos entre si.

Definição 3 (partição): uma *partição* π num conjunto S é uma cobertura de $S = \{b_1, b_2, b_3, \dots, b_n\}$ ($n \geq 1$) onde $b_i \cap b_j = \emptyset$ para todo $i \neq j$. Cada elemento b_i de π é denominado de *bloco* da partição.

O conceito de partição pode ser aplicado ao conjunto de estados de uma MEF. No exemplo, dado o conjunto $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ duas partições do conjunto de estados da MEF são: $\pi_1 = \{(s_1), (s_2, s_3), (s_4), (s_5, s_6)\}$ e $\pi_2 = \{(s_1, s_2), (s_3, s_6), (s_4, s_5)\}$.

Definição 4 (partição nula): a *partição nula* é definida como uma cobertura de um conjunto $S: \{b_1, b_2, b_3, \dots, b_n\}$ onde $b_i \cap b_j = \emptyset$ para todo $i \neq j$ e cada bloco da partição contém só um elemento de S . A partição *nula* denota-se por $\pi(0)$.

No exemplo a partição nula correspondente ao conjunto de estados da MEF $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ é $\pi(0) = \{(s_1), (s_2), (s_3), (s_4), (s_5), (s_6)\}$.

Definição 5 (produto de duas partições): O *produto das partições* π_1 e π_2 (denotado como $\pi_1 \bullet \pi_2$) é definido como a partição π_p , onde cada bloco de π_p corresponde à intersecção de um bloco de π_1 e um bloco de π_2 . Se a intersecção de um par de blocos é vazia ($\emptyset$) o bloco intersecção não faz parte de π_p .

Dadas as partições $\pi_1 \{(s_1), (s_2, s_3), (s_4), (s_5, s_6)\}$ e $\pi_2 = \{(s_1, s_2), (s_3, s_6), (s_4, s_5)\}$ o produto das duas partições $(\pi_1 \bullet \pi_2)$ é $\pi_p = \{(s_1), (s_2), (s_3), (s_4), (s_5), (s_6)\}$.

4. O CONCEITO DE DECOMPOSIÇÃO APLICADO A MEFS

A condição básica para se obter uma decomposição válida de uma MEF em n sub-MEFs é que o conjunto de partições da MEF $S = \{\pi_1, \pi_2, \pi_3, \dots, \pi_n\}$ cumpram a condição de que $\pi_1 \bullet \pi_2 \bullet \pi_3 \bullet \dots \bullet \pi_n = \pi(0)$ ($n \geq 1$) [Har66], [Koh78][Boo67]. Neste caso as n sub-MEFs que formam a MEF decomposta serão capazes (em conjunto) de reproduzir o funcionamento da MEF original [Har66]. Desta maneira um conjunto de partições de uma MEF cujo produto seja a partição nula gerarão uma decomposição válida.

As partições $\pi_1 \{(s_1), (s_2, s_3), (s_4), (s_5, s_6)\}$ e $\pi_2 = \{(s_1, s_2), (s_3, s_6), (s_4, s_5)\}$ formam uma decomposição válida dado que $\pi_1 \bullet \pi_2 = \pi(0)$.

5. O SISTEMA SINMEF

SINMEF é um sistema de síntese de MEFs que baseia-se na técnica de decomposição. Ele é composto por três sistemas: DECMEF, SIS [Sis92] e LAGER [LAG90]. No sistema DECMEF estão contidas as ferramentas desenvolvidas neste trabalho que visam gerar uma descrição funcional da MEF decomposta. O sistema SIS [Sis92] é usado para a síntese e otimização lógica da MEF decomposta. Finalmente, o sistema LAGER [LAG90] é usado para a geração do leiaute do circuito.

6. O SISTEMA DECMEF

Nesta seção será apresentado o sistema DECMEF implementado para a decomposição MEFs complexas.

6.1 A metodologia do sistema DECMEF

A figura 1 mostra o diagrama de blocos do sistema DECMEF. O sistema é composto de um conjunto de ferramentas que permitem ao projetista gerar (rapidamente) diferentes MEFs decompostas. O sistema é interativo e o usuário pode testar várias opções para obter uma implementação ótima (dentro de especificações) do circuito.

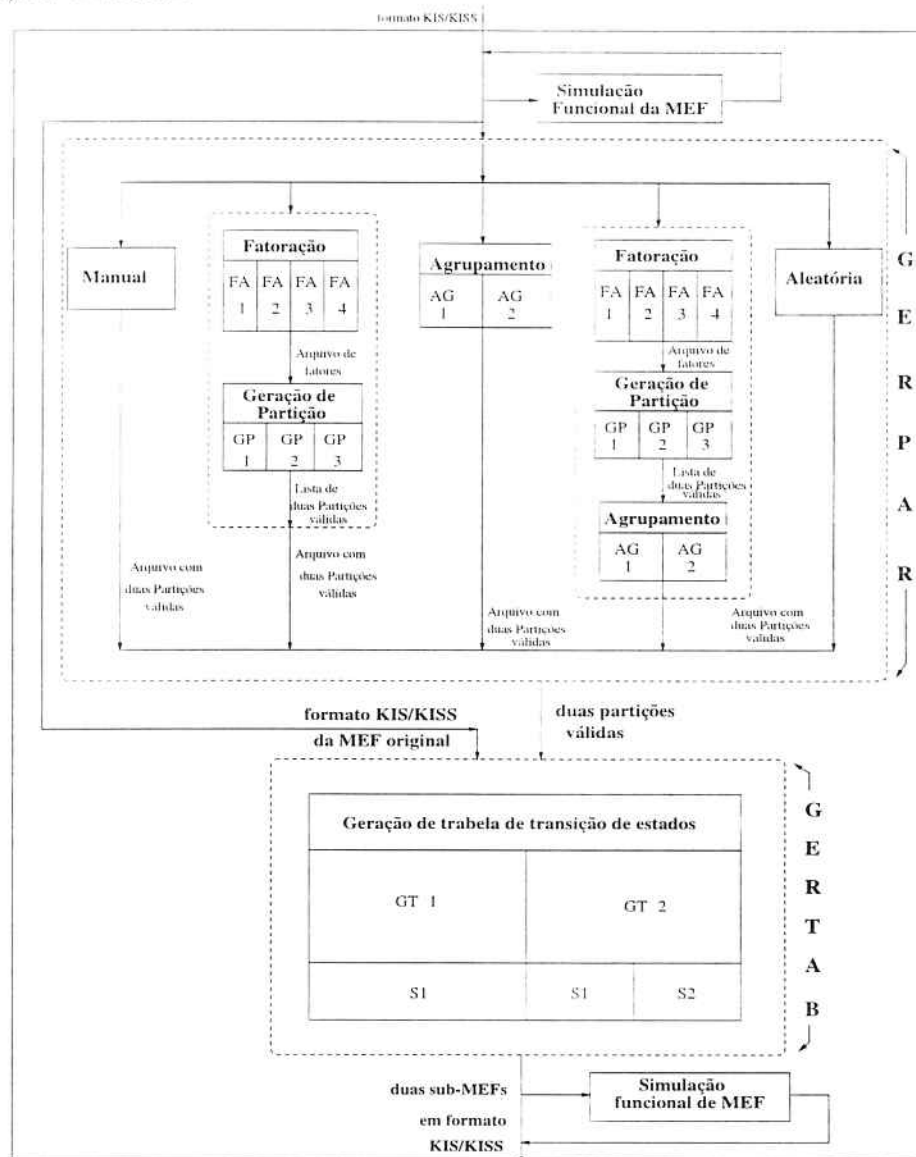


Fig. 1: o sistema DECMEF

O sistema DECMEF é formado, basicamente, por dois conjuntos de ferramentas. O primeiro, denominado de GERPAR, gera duas partições de estados válidas da MEF a ser decomposta. O segundo, denominado GERTAB, gera as tabelas de transição de estados das sub-MEFs das sub-MEFs correspondentes às duas partições, bem como suas descrições KIS/KISS. Um simulador funcional de MEFs decompostas ou não decompostas, denominado SSME [Rus94], permite validar as descrições inicial e final do sistema DECMEF.

O conjunto GERPAR contém cinco opções para a geração de partições:

1. Decomposição manual. Permite ao usuário gerar manualmente duas partições. O DECMEF faz a leitura através de arquivo das partições.

2. Decomposição por fatoração. Permite gerar duas partições usando a técnica de fatoração. Esta alternativa é formada pelas ferramentas FACT e PART. O programa FACT permite obter uma lista de fatores obtidos a partir do GTE da MEF. Este programa oferece quatro opções para o usuário (FA_1 , FA_2 , FA_3 e FA_4 na figura 1). O programa PART permite obter uma lista de fatores compatíveis e gera, a partir desta, duas partições válidas da MEF. O programa PART oferece três opções (GP_1 , GP_2 , GP_3 na figura 1).

3. Decomposição por agrupamento de estados (*clustering*). Permite a geração das partições por uma técnica denominada de *agrupamento de estados* [Lla95], implementada numa ferramenta denominada de AGRUP. O programa AGRUP oferece duas opções para o usuário (AG_1 , AG_2 na figura 1).

4. Decomposição por fatoração seguida de agrupamento de estados. Permite gerar duas partições concatenando as duas técnicas acima listadas.

5. Decomposição aleatória. Permite a geração de duas partições válidas de maneira aleatória.

Em todos os casos o sistema valida as duas partições determinando se o produto entre elas é $\pi(0)$.

O conjunto GERTAB contém duas opções para a geração da tabela de transições de cada sub-MEF ($GT1$ e $GT2$ na figura 1) uma para comunicação entre sub-MEFs através de estados auxiliares de entrada e saída (denominados de EAE e EAS) e a outra para comunicação entre sub-MEFs através de todos seus estados presentes. Na opção $GT2$ existem mais duas opções ($S1$ e $S2$ na figura 1) correspondendo a saídas particionadas ou não.

6.2 As ferramentas do sistema DECMEF

A tabela 1 mostra a lista de tarefas realizadas pelo conjunto GERPAR (vide figura 1) e o nome das ferramentas implementadas para a execução das mesmas.

Tarefa	nome da ferramenta
Fatoração	FACT
Geração de partição de estados	PART
Agrupamento de estados	AGRUP
Decomposição aleatória	AGRUP (opção de agrupamento aleatório)

Tabela 1: tarefas do GERPAR e as ferramentas implementadas

A tabela 2 mostra as tarefas realizadas pelo conjunto GERTAB com as ferramentas implementadas para cada uma delas.

Tarefa	ferramenta
Geração das sub-MEFs com EAS	GERTAB1
Geração das sub-MEFs sem EAS	GERTAB2
Geração das sub-MEFs particionando o espaço de saídas	GERTAB2

Tabela 2: tarefas do GERTAB e as ferramentas implementadas

O usuário do DECMEF escolhe usar uma ferramentas a través de um menu de opções. Cada menu abre sub-menus onde o usuário escolhe uma opção para cada ferramenta.

6.3 Tratamento do problema de transições não-determinísticas no DECMEF

Numa decomposição obtida através de um conjunto de partições $S = \{ \pi_1, \pi_2, \dots, \pi_n \}$ podem surgir transições não-determinísticas nos diferentes blocos o super-estados das partições. A Nesta seção serão mostradas as estratégias para o tratamento de transições não-determinísticas usadas no sistema DECMEF.

6.4 Relação entre as variáveis de saída e as transições não-determinísticas

Numa decomposição ambas as sub-MEFs geram todas as saídas. Entretanto, não é necessário que ambas gerem saídas simultaneamente. Assim, cada sub-MEF tem suas saídas ativadas ou não em cada transição. A implementação desta técnica requer o uso de um multiplexador. A definição de qual sub-MEF está ativada em cada transição pode ser feita pelo projetista ou ser determinada a partir de um critério de otimização, que será explicado a seguir.

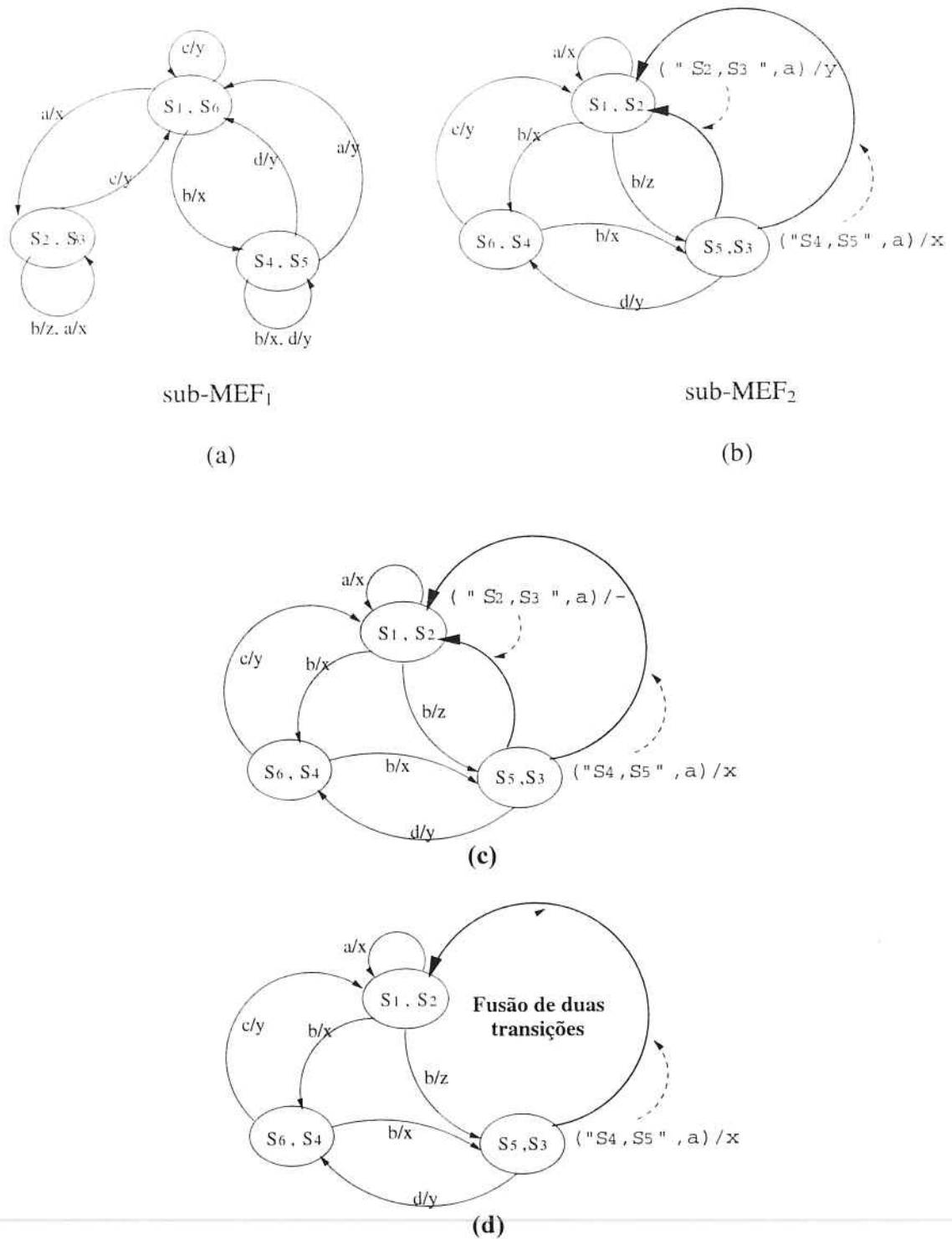


Fig. 2: o problema das transições não-determinísticas

No exemplo da figura 2 as transições (dadas na forma de *entradas, estado presente, estado futuro e saída*) $t_1 = a(s_5, s_3)(s_1, s_2)x$ e $t_2 = a(s_5, s_3)(s_1, s_2)y$, na sub-MEF₂, são não-determinísticas. Entretanto, se a saída gerada por t_2 (y) na sub-MEF₂ for *desativada* e a transição correlacionada da sub-MEF₁, $t_3 = a(s_4, s_5)(s_1, s_6)y$ estiver ativada, então a transição da sub-MEF₂ transforma-se em $t_2 = a(s_5, s_3)(s_1, s_2)-$, onde o símbolo “-” representa *don't care* (vide figura 2.c). Assim o par de transições não-determinísticas deixa de existir (vide figura 2.d). Mais ainda, o novo par de transições pode ser **fundido** numa única transição. A eliminação de transições não-determinísticas pela desativação de saídas, e posteriormente pela fusão delas, pode ser usada como um critério de otimização da MEF decomposta.

Transições que podem ser eliminadas por *fusão* são denominadas, neste trabalho, de *transições redundantes*. No caso de transições não-determinísticas que não possam serem eliminadas por fusão a sub-MEF (onde existe a indeterminação) deve conhecer o espaço de estados da outra sub-MEF. Neste caso as sub-MEFs comunicam-se através dos seus estados presentes. Estes estados presentes correspondem a entradas adicionais para as sub-MEFs com transições não-determinísticas e são denominados de *estados de dependência*, e quando são codificados são denominados de *estados auxiliares de entrada* (EAE).

7. A GERAÇÃO DE PARTIÇÕES NO DECMEF (O GERPAR)

Nesta seção será apresentado o programa GERPAR para a geração de partições que serão usadas no problema de decomposição.

7.1 Geração de partições por fatoração

A técnica de decomposição por fatores foi introduzida por Devadas e colaboradores [Dev89] como uma heurística para se obter partições numa MEF. A técnica de fatoração é uma aplicação direta da teoria das partições (dada por Hartmanis e colaboradores [Har66]) e do princípio básico de decomposição.

7.1.1 Descrição do problema de fatoração

A figura 3 mostra uma MEF original onde os estados s_2, s_3, s_4 e os estados s_5, s_6 , e s_7 têm transições com iguais condições de entrada e de saída. Estes dois conjuntos de estados podem ser extraídos da MEF original como um fator. Neste caso, o fator tem duas ocorrências: s_2, s_3, s_4 e s_5, s_6 , e s_7 . As ocorrências do fator são denominadas de *instâncias do fator*. Na figura 3 os estados s_2 e s_5, s_3 e s_6 e s_4 e s_7 são denominados de *estados correspondentes*. As transições cujas condições de entrada/saída são $a/z, c/x$ e b/x (para cada instância da MEF original) são denominadas de *transições internas* do fator e os pares de transições com condições entrada/saída $(a/z, a/z), (c/x, c/y)$ e $(b/x, b/x)$, para cada instância do fator, são *transições correspondentes*. Os estados s_2, s_3, s_5 e s_6 são denominados de estados internos do fator pois suas transições-saída são transições internas no fator. Os estados s_4, s_7 são denominados de *estados saída* do fator. Na sub-MEF₁, indicada na figura 4.a, as instâncias do fator foram substituídas pelos estados d_1 e d_2 respectivamente. Na mesma figura, os estados s_1 e s_8 são estados da MEF original que não formam parte do fator. O fator é implementado como uma sub-MEF (sub-MEF₂ da figura 4.b) de maneira análoga a uma sub-rotina. Neste caso, os estados s_a, s_b e s_c representam pares de estados correspondentes do fator, isto é, $s_a = \{s_2, s_3\}$, $s_b = \{s_3, s_6\}$, $s_c = \{s_4, s_7\}$. O estado p (da mesma figura) representa o conjunto de estados não fatorados da MEF original (neste caso s_1 e s_8).

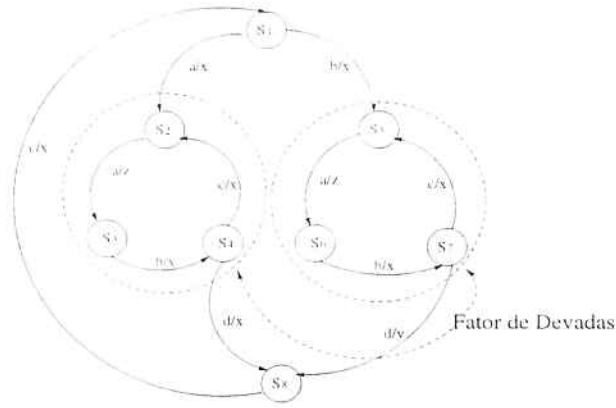


Fig. 3: MEF a ser decomposta usando fatoração

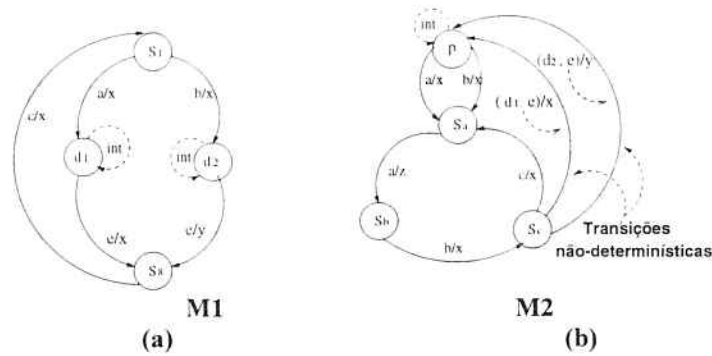


Fig. 4: MEF decomposta usando fatoração

7.1.2 Teoria modificada para a extração de fatores

O programa FACT implementa uma teoria modificada de fator. O conceito de fator foi relaxado, diminuindo as restrições dadas na definição inicial. Isto objetiva aumentar a possibilidade de extrair fatores do GTE (grafo de transição de estados) da MEF original. Os fatores cujas definições serão dadas nesta seção podem ser considerados como fatores inexatos segundo a terminologia usada por Devadas e colaboradores [Dev89] .

Definição 6 (transições tipo alfa): transições do *tipo alfa* são transições com iguais condições de entrada e gerando saídas que podem ser iguais ou diferentes.

Definição 7 (transições tipo beta): transições do *tipo beta* são transições cujas condições de entrada têm interseção cúbica diferente de vazio ($\phi \neq \emptyset$) e suas saídas são iguais.

Definição 8 (transições tipo gama): transições do *tipo gama* são transições cujas condições de entrada têm interseção cúbica diferente de vazio ($\phi \neq \emptyset$) e as suas saídas geradas são iguais ou diferentes.

Definição 9 (fator tipo alfa): um fator do *tipo alfa* é composto de dois ou mais conjuntos de estados tais que os estados correspondentes apresentam:

- transições internas (transições correspondentes) são *tipo alfa*;
- os estados internos e estados-saída são estados correspondentes;

Definição 10 (fator tipo beta): um fator do *tipo beta* é definido de maneira similar ao *tipo alfa* mas com transições internas *tipo beta*.

Definição 11 (fator tipo gama): um fator do *tipo gama* é definido de maneira similar ao *tipo alfa* mas com transições internas do *tipo gama*.

Os fatores do *tipo gama* são uma combinação dos fatores dos *tipos alfa e beta*.

A obtenção de fatores tipo *Devadas*, *alfa*, *beta* e *gama* (de acordo com as definições 9, 10 e 11) é implementada no programa FACT nas opções *FA1*, *FA2*, *FA3* e *FA4* (vide figura 1).

7.1.3 A geração de partições a partir de fatores (o programa PART)

O programa PART (vide figura 1) gera duas partições válidas a partir de um conjunto de fatores obtidos através do programa FACT. O programa PART obtém uma cobertura máxima de estados a partir de fatores que são compatíveis entre si. Dois fatores são compatíveis quando a intersecção entre seus estados é nula. Desta maneira os fatores que geram a partições são disjuntos entre si.

O programa PART tem três opções para a geração das partições: GP1, GP2 e GP3 (vide figura 1). A opção *GP1* implementa um algoritmo de compatibilidade de cliques baseado no algoritmo para obtenção de todos os cliques de um grafo [Bro73], o qual é um problema NP-Completo. As duas opções (*GP2* e *GP3*) implementam duas heurísticas para obter um conjunto de fatores compatíveis.

7.2 Geração de partições por agrupamento de estados (programa AGRUP)

Nesta seção será descrito o programa AGRUP (vide figura 1) para obtenção de partições tendo como ponto de partida a partição nula ($\pi(0)$).

Dada uma MEF de por exemplo 8 estados, a partição inicial $\pi = \pi(0)$ é:

$$\pi = \{(s_1), (s_2), (s_3), (s_4), (s_5), (s_6), (s_7), (s_8)\}$$

Movimentando-se o estado s_8 para o bloco do estado s_2 a partição fica:

$$\pi = \{(s_1), (s_2, s_8), (s_3), (s_4), (s_5), (s_6), (s_7)\}$$

Neste caso o estado s_8 foi *agrupado* com o estado s_2 .

A estratégia consiste em definir duas listas de estados. A primeira lista contém estados que ficarão “fixos” e, neste caso farão um papel de *núcleos* do super-estado (no exemplo anterior é s_2). A Segunda lista contém os estados individuais que podem ser movimentados para formar parte de super-estados.

A criação da lista de estados *núcleos* é feita em duas fases. Na primeira, a lista de estados ($\pi(0)$) é ordenada de forma decrescente pelo número de transições-saída de cada estado. Na segunda fase o usuário especifica o número n de super-estados desejados. Neste caso, os n primeiros estados da lista serão escolhidos como *núcleos* de super-estados. Pode ser observado que o valor de n deve cumprir que $1 < n < N_s$ onde N_s é o número de estados da MEF.

No caso da partição do exemplo anterior se a lista de estados l_s , ordenada em ordem decrescente pelo número de transições-saída de cada estado, for: $l_s = s_7, s_4, s_6, s_1, s_5, s_2, s_3, s_8$ e o usuário escolher que as partições terão 4 super-estados, os quatro primeiros estados da lista serão escolhidos como *núcleos* (s_7, s_4, s_6, s_1) e os outros estados (s_5, s_2, s_3, s_8) serão candidatos para ser movimentados.

A escolha dos n primeiros estados para serem *núcleos* é explicada pelo fato de que estes estados são os que têm mais probabilidades de dar transições não-determinísticas no agrupamento produto da movimentação. Portanto, o critério usado neste trabalho é de deixá-los *fixos* inicialmente, e trabalhar a movimentação de *estados individuais* sobre eles.

Duas condições devem ser levadas em conta na movimentação de um *estado individual* para um super-estado (representado inicialmente pelo *núcleo*). A primeira é de que o produto entre as partições, após a movimentação deve ser a partição nula. A segunda é a boa qualidade da decomposição após terminado o processo de movimentações sucessivas dos *estados individuais*. Para garantir esta qualidade são usadas duas funções custo dadas a seguir:

- 1) minimizar o número de transições não-determinísticas;

2) maximizar o número de transições redundantes [Lla95], [Lla96].

A primeira função custo procura evitar a necessidade de eliminação de transições não-determinísticas usando as técnicas de estados de dependência.

A segunda função custo visa diminuir o número final de transições em cada sub-MEF aumentando o número de transições que podem ser fundidas.

O programa AGRUP tem duas opções de seqüenciamento das funções custo de definidas. Na primeira opção (*AG1*) o programa AGRUP prioriza minimizar o número de transições não-determinísticas numa movimentação de estados. Se existirem dois estados que minimizem no mesmo grau as transições não-determinísticas é escolhido aquele que maximize o número de transições redundantes.

Na segunda opção (*AG2*) o programa AGRUP prioriza maximizar o número de transições redundantes. No caso de que dois ou mais estados maximizem com o mesmo grau as transições redundantes é escolhido aquele que minimize o número de transições não-determinísticas.

A técnica de agrupamento pode ser usada tanto para a geração de partições a partir da partição nula como para a otimização de partições obtidas por fatoração (obtidas usando os programas FACT e PART em seqüência).

8.0 GERAÇÃO DE TABELAS DE TRANSIÇÃO DE ESTADOS (O PROGRAMA GERTAB)

Uma tarefa importante na decomposição é a geração das tabelas de transição de estados para cada sub-MEF. Desta maneira as indeterminações produto da geração de partições devem ser eliminadas. O GERTAB implementa duas estratégias para a eliminação de transições não-determinísticas. A primeira foi implementada num programa denominado de GERTAB1 e a segunda no GERTAB2.

8.1 O programa GERTAB1

O GERTAB1 tenta minimizar o grau de interconexão entre as sub-MEF. Isto é possível dado que para a eliminação de indeterminação muitas vezes é necessário conhecer só um sub-conjunto de estados da outra sub-MEF. O GERTAB codifica este sub-conjunto de estados e estes códigos serão gerados como saídas auxiliares por cada sub-MEF (denominados de estados auxiliares de saída, EAS) e serão recebidos pela outra sub-MEF como estados auxiliares de entrada (EAE) (vide seção 6.4).

O programa GERTAB1 realiza o seguinte conjunto de tarefas:

- 1) leitura da descrição da MEF original;
- 2) leitura da partição de estados;
- 3) validação das partições (determina se as partições são válidas ou não);
- 4) leitura da atribuição *ATIVO/INATIVO* das saídas de cada sub-MEF;
- 5) determinação das transições não-determinísticas;
- 6) determinação do número mínimo de EAE e de EAS;
- 7) codificação dos *estados auxiliares de entrada (EAE)*;
- 8) codificação dos *estados auxiliares de saída (EAS)*;
- 9) fusão de transições redundantes em cada sub-MEF;
- 10) geração das tabelas de transição de estados de cada sub-MEF (formato KIS/KISS).

A figura 5 mostra a arquitetura alvo da MEF decomposta usando o programa GERTAB1. Neste caso, cada sub-MEF gera os EAS para a outra e recebe desta estes estados na forma de EAE.

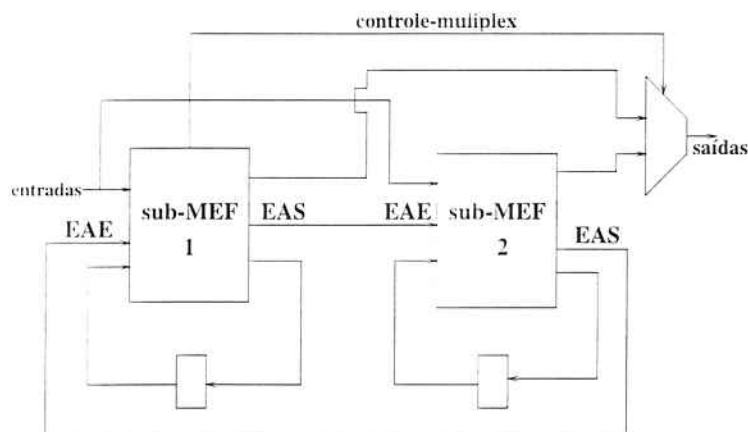


Fig. 5: arquitetura alvo para a MEF decomposta usando o GERTAB1

A sub-MEF₁ gera uma saída de controle para o multiplexador selecionar a sub-MEF cuja saída deve estar ativa/inativa em cada ciclo. Isto é usado para a minimização de transições não-determinísticas (vide seção 6.4). As transições de cada sub-MEF são especificadas pelo seguinte conjunto de 6 valores (em contraste com MEFs convencionais que só requerem 4 valores): *EAE*, *entradas*, *estados presentes*, *estados futuros*, *saídas* e *EAS*.

O GERTAB1 corresponde à opção *GT1* da figura 1.

8.2 O programa GERTAB2

O programa GERTAB2 elimina a necessidade de se usar EAS para eliminar transições não-determinísticas, pela comunicação através de todo os estados presentes das sub-MEFs. Na primeira parte o GERTAB2 realiza, basicamente, os 7 primeiros passos do GERTAB1:

- 1) leitura da descrição da MEF original;
- 2) leitura da partição de estados;
- 3) validação das partições (determina se as partições são válidas ou não);
- 4) leitura da atribuição *ATIVO/INATIVO* das saídas de cada sub-MEF;
- 5) determinação das transições não-determinísticas;
- 6) determinação do número mínimo de EAE;
- 7) codificação dos *estados auxiliares de entrada (EAE)*;

Neste ponto as duas sub-MEF não terão transições não-determinísticas dado que os EAE eliminaram qualquer indeterminação. Depois disto o GERTAB2 usa a ferramenta de codificação de estados NOVA[Vil90] para a codificação dos estados de cada sub-MEF. Os códigos destes estados (gerados pelo NOVA) substituirão os EAE gerados no passo 7. Estes códigos serão armazenados num arquivo e serão os códigos definitivos para os estados no processo de síntese lógica.

As transições de cada sub-MEF são especificadas por: *EAE*, *entradas*, *estados presentes*, *estados futuros* e *saídas*.

O GERTAB2 corresponde à opção *GT2* da figura 1 e tem duas opções: *S1* e *S2*. A primeira usa a estratégia de multiplexação para minimizar transições não-determinísticas. A segunda não usa a opção de multiplexação e particiona o espaço das saídas entre as duas sub-MEFs.

9.0 DECOMPOSIÇÃO RECURSIVA NO DECMEF

A decomposição recursiva pode ser aplicada facilmente para uma MEF com uma ou duas sub-MEFs cujas partições sejam fechadas. A decomposição paralela pode ser obtida usando a estratégia de geração de partições fechadas ou pode ser obtida naturalmente, isto é sem nenhum artifício adicional, em casos particulares.

Neste caso as sub-MEF(s) são independentes. A figura 6 mostra uma decomposição do tipo paralelo onde é possível se fazer uma decomposição recursiva em ambas sub-MEFs (as duas partições π_1 e π_2 são fechadas). A sub-MEF A pode ser novamente decomposta usando as ferramentas GERPAR e GERTAB, gerando as sub-MEFs A_1 e A_2 . O mesmo pode ser feito com a sub-MEF B .

Dada uma MEF decomposta com pelo menos uma sub-MEF independente, as ferramentas de decomposição do DECMEF podem ser aplicadas novamente sobre esta.

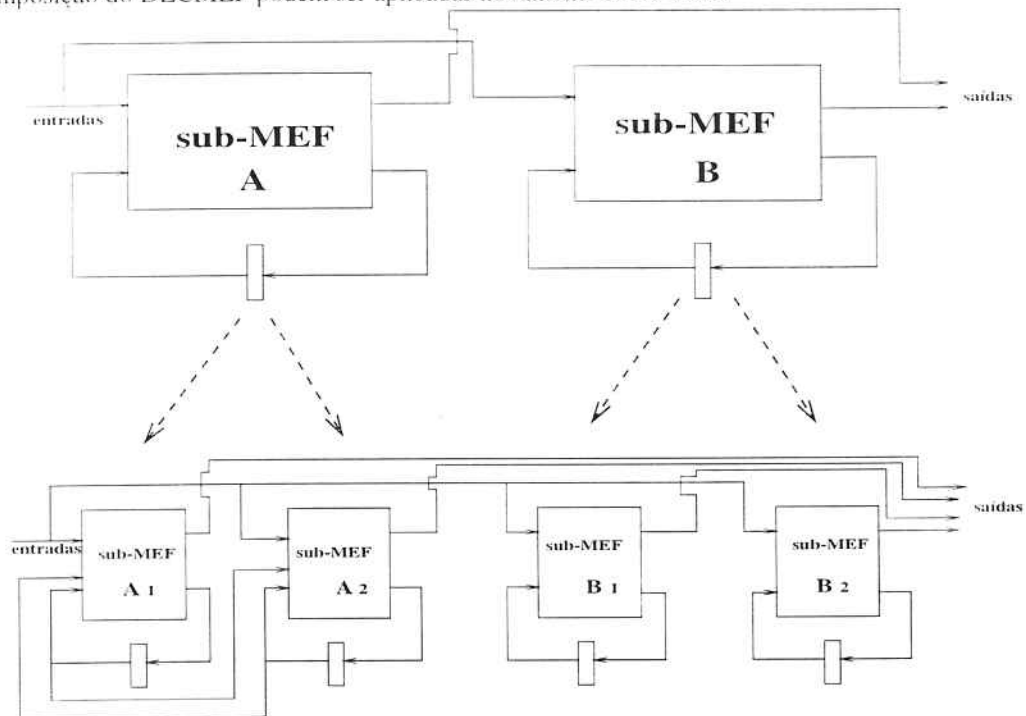


Fig. 6: decomposição recursiva de uma MEF

10. RESULTADOS

A tabela 3 mostra as características de MEFs de diferentes complexidade. Os exemplos *tbk*, *sand*, *planet*, *scf*, *seqbis1*, *s298* e *controller* e *master* são benchmarks de MEFs. Os exemplos *blanca*, *mónica* e *ainda* são MEFs geradas de maneira aleatória usando o programa GERMEF (para a geração aleatória de MEFs). Nas colunas 6 e 7 são mostrados os resultados da síntese usando o sistemas SIS. A biblioteca usada foi a *lib2.genlib*.

nome	Características das MEFs				Resultados da Síntese	
	estados	entradas	saídas	transições	número de portas	caminho crítico (**)
tbk	32	6	3	1569	452	39.86
sand	32	11	9	184	473	38.81
planet	48	7	19	115	540	50.68
scf	121	27	56	166	797	80.58
seqbis1	141	21	33	1096	9865	347.24
s298	218	3	6	1096	1853	120.43
controller	353	2	4	1400	1547	128.35
blanca++	600	2	3	1203	*	*
mónica++	700	5	4	1050	*	*
ainda++	1000	4	2	1500	*	*
master-read	1098	13	7	4989	*	*

(*) resultados não obtidos com o SIS. (**) unidades de tempo relativas.

(++) exemplos gerados aleatoriamente com o programa GERMEF.

Tabela 3 características dos exemplos tratados e resultados de área e caminho crítico

A tabela 4 mostra o tipo de fatores obtidos para os exemplos mostrados (primeira coluna). Pode ser observado que para certos exemplos não foi possível obter fatores. Isto é devido a que os fatores *alfa* e *gama* exigem maiores recursos computacionais. Neste caso o problema apresentou explosão combinatória.

nome	<i>Devadas</i>	<i>alfa</i>	<i>beta</i>	<i>gama</i>
tbk	X	X	X	X
sand	X	X	X	X
planet	X	X	X	X
scf	X	X	X	X
seqbis1	X	X	X	
s298	X		X	
controller	X		X	

Tabela 4: tipos de fatores obtidos em exemplos

A tabela 5 mostra os melhores resultados obtidos usando fatoração e agrupamento de estados. Esta tabela mostra que os melhores resultados foram obtidos usando fatoração (programa FACT e PART) seguido de agrupamento (programa AGRUP). A mesma tabela mostra que para uma partição obtida por fatoração (por exemplo usando fatores do tipo *Devadas*) foram usadas as duas opções de seqüenciamento do programa AGRUP (AG1 e AG2). Neste caso a partição é especificada como *dev_1* para o caso de fatoração tipo *Devadas* seguida de agrupamento tipo *AG1* e *dev_2* para o caso do mesmo tipo de fatoração seguido de agrupamento do tipo *AG2* (vide seção 7.2).

Nome	ganhos em área			ganhos em caminho crítico		
	ganho (%)	tipo	opção	ganho (%)	tipo	opção
tbk	60,62	beta_1	3	50.55	beta_1	3
sand	23.68	gama_1	3	54.83	dev_1	3
planet	18.15	alfa_2	3	52.25	alfa_1	3
scf	13.80	alfa_2, beta_2	2	62.39	beta_1	2
seqbis1	16.60	dev_1	3	54.36	alfa_2	3
s298	35.45	dev_1	3	57.87	dev_1	3
controller	42.21	beta_2	3	67.83	dev_1	3

Tabela 5: melhores resultados obtidos com as decomposições discutidas

A tabela 6 mostra os resultados de partições obtidas por agrupamento de estados partindo da partição nula. A opção 1 indica que foi usado o programa GERTAB1 a opção 2 indica que foi usado o programa GERTAB2 sem particionamento do espaço das saídas (opção GT2/S1) e a opção 3 indica que foi usado o programa GERTAB2 particionando o espaço das saídas entre as duas sub-MEFs (opção GT2/S2) (vide figura 1).

nome	sub-MEF	estados	número de portas			caminho crítico		
			opção 1	opção 2	opção 3	opção 1	opção 2	opção 3
planet	1	19	485	379	367	43.11	32.75	31.00
	2	13	454	314	156	39.73	27.66	17.07
scf	1	47	624	411	450	41.62	31.03	38.28
	2	42	796	475	437	54.71	39.60	37.09
controller	1	123	1888	809	747	106.15	53.17	50.79
	2	124	1938	836	747	104.51	52.16	48.42
blanca	1	250	5865	3296	3564	308.35	192.05	214.80
	2	250	5330	2826	3360	301.95	178.79	195.52
mónica	1	300	6371	3887	3841	469.20	272.50	254.12
	2	300	7079	3966	3898	476.06	263.66	290.85
aida	1	200	5072	5072	4936	302.92	302.92	292.72
	2	200	4861	4861	5221	286.79	286.79	301.97

Tabela 6: resultados para partições obtidos sem fatoração

A tabela 7 mostra os resultados de ganhos em área e caminho crítico para os exemplos *sand*, *planet*, *scf* e *controller*. Pode ser observado um ganho máximo em área de 14.79% para o exemplo *sand* com a opção 3. O ganho máximo em caminho crítico foi de 60.42% para o exemplo *controller* usando a opção 3.

nome	ganhos em área (%)			ganhos em caminho crítico (%)		
	opção 1	opção 2	opção 3	opção 1	opção 2	opção 3
sand	-24.73	10.14	14.79	24.45	46.30	35.42
planet	-73.88	-28.33	3.14	14.93	35.45	38.83
scf	-86.95	-11.16	-11.29	32.10	50.85	52.49
controller	-147.31	-6.33	3.42	17.29	58.57	60.42

Tabela 7: resultados de ganhos em área e caminho crítico para os resultados da tabela 6

As tabelas 8 e 9 mostram um resumo dos melhores resultados em área e caminho crítico obtidos usando fatoração (com otimização) e os obtidos sem fatoração (usando agrupamento a partir de $\pi(0)$). Os exemplos tratados são: *sand*, *planet*, *scf* e *controller*. A coluna 1 (das tabelas 8 e 9) mostra o nome do exemplo, a coluna 2 mostra os melhores resultados obtidos usando fatoração otimizada, a coluna 3 mostra os resultados dos ganhos para as decomposições obtidas por agrupamento a partir de $\pi(0)$ e a coluna 4 mostra a diferença entre os dois resultados (diferença dos ganhos). Pode ser observado que os resultados em área e caminho crítico, para estes exemplos, sempre foram melhores nas partições obtidas por fatoração e otimizadas usando o programa AGRUP.

Nome	ganhos em área (%)		diferença dos ganhos
	fatoração otimizada	agrupamento partindo de $\pi(0)$	
sand	23.68	14.79	8.89
planet	18.15	3.14	15.01
scf	3.80	-11.16	14.06
controller	42.21	3.42	38.19

Tabela 8: comparação dos melhores resultados em área obtidos com fatoração e os obtidos sem fatoração

nome	ganhos em c. crítico (%)		diferencia dos ganhos
	fatoração otimizada	agrupamento partindo de $\pi(0)$	
sand	54.83	46.30	8.53
planet	52.25	38.83	13.42
scf	62.39	52.49	9.90
controller	67.83	60.42	7.41

Tabela 9: comparação dos melhores resultados em caminho crítico obtidos com fatoração e os obtidos sem fatoração

A tabela 10 mostra os resultados da síntese do exemplo *master*. Os resultados foram obtidos usando o GERTAB2 com particionamento do espaço das saídas (opção 3 dos exemplos anteriores). Neste caso, foi detectada uma decomposição paralela. A figura mostra os resultados de área e de caminho crítico para a sub-MEF₁ e a sub-MEF₂. Pode ser observado (comparando com os dados da tabela 5.4) que não foram eliminadas transições por fusão. Este exemplo não apresenta *don't cares* nas entradas e esta condição levou a inexistência de transições não-determinísticas após a decomposição e, por outro lado, dificultou fusão de transições.

sub-mef	estados	entradas	saidas	produtos	área	ca. crítico
1	350	13	4	4989	12602	446.02
2	350	13	3	4989	6954	279.55

Tabela 10: resultados para o exemplo *master* para uma decomposição em duas sub-MEFs

As tabelas 11 e 12 mostram os resultados da decomposição recursiva da sub-MEF₁ e da sub-MEF₂ obtidas na decomposição da tabela 10. Pode ser observado que não foram eliminadas transições por fusão em nenhum dos casos.

sub-mef	estados	entradas	saidas	produtos	área	ca. crítico
1	60	13	3	4989	9848	402.88
2	60	13	2	4989	8870	363.39

Tabela 11: decomposição recursiva da sub-MEF₁ da tabela 10

No caso da sub-MEF₁ (tabela 10), somando as áreas das duas sub-MEFs da tabela 11, obteve-se um aumento da área do circuito. Entretanto, cada sub-MEF (da tabela 11) tem uma área menor que a sub-MEF₁ da tabela 10. O ganho em caminho crítico da decomposição recursiva da sub-MEF₁ foi de 9.67%.

No caso da sub-MEF₂ a decomposição recursiva (tabela 12) não apresentou ganhos nem área nem em caminho crítico.

sub-mef	estados	entradas	saidas	produtos	área	ca. crítico
1	60	60	3	4989	7748	325.01
2	60	60	2	4989	6673	277.76

Tabela 12: decomposição recursiva da sub-MEF₂ da tabela 10

11. CONCLUSÕES

Neste trabalho foram obtidos resultados de decomposição usando fatoração para os 7 *benchmarks* (*tbk*, *sand*, *planet*, *scf*, *seqbis1*, *s298* e *controller*), sendo que este último *benchmark* tem 353 estados. Os resultados mostraram que para os exemplos de menor complexidade (*tbk*, *sand*, *planet* e *scf*) foi possível obter todos os tipos de fatores. Foi analisado o problema de se obter decomposições usando os fatores dos tipos *alfa* e *gama*. Estes fatores tendem a agrupar um maior número de estados em super-estados mas são os que oferecem maiores problemas do ponto de vista computacional para serem obtidos. O problema é definido principalmente na alocação dinâmica de memória. Em parte os problemas foram detectados no próprio compilador de C++ da GNU. Neste caso a liberação de memória não era feita de maneira correta. Fica claro que este tipo de problema poder ser resolvido em parte usando um outro compilador. Entretanto, a própria natureza do problema e do algoritmo de extração de fatores indicam que a geração de fatores *alfa* e *gama* sempre precisará de maiores recursos computacionais que a geração de fatores dos tipos *Devadas* e *beta*.

Os fatores do tipo *Devadas* permitiram fundir um maior número de transições e apresentaram um menor número de EAE e de EAS indicando, portanto, possuir um menor número de transições não-determinísticas. Por outro lado, os resultados da síntese mostram que as novas definições de fatores (*alfa*, *beta* e *gama*) foram responsáveis por 71.42% dos melhores resultados obtidos usando fatoração sem otimização. Isto mostra claramente a importância da nova técnica de fatoração desenvolvida neste trabalho.

A técnica de agrupamento de estados (programa AGRUP) foi desenvolvida neste trabalho com dois objetivos. O primeiro foi o de otimizar as partições obtidas por fatoração e o segundo o de permitir gerar partições válidas por agrupamento iniciando com a partição nula. Os casos de maior movimentação de estados foram obtidos usando a opção *AG2* do AGRUP. Entretanto, a diferença entre as movimentações do *AG1* e *AG2* para os diferentes exemplos não foram muito grandes. Os resultados mostraram movimentações de estados para super-estados de até 89% (partição π_2 do exemplo *s298*) comparando com o número de estados iniciais da partição.

A técnica de agrupamento aplicada a partições geradas por fatoração provou ser muito útil pois todos os resultados da síntese (área e caminho crítico) de decomposições geradas por fatoração e otimizadas foram sempre melhores que os resultados de decomposições usando só fatoração. A opção *AG1* (do AGRUP) foi responsável por 57.14% dos melhores resultados em área e por 85.71% dos melhores resultados em caminho crítico indicando que minimizar os número de EAE e de EAS tem mais impacto que maximizar o número de transições redundantes. Isto pode ser visto do ponto de vista da diferença entre uma redução vertical da MEF (eliminando transições por fusão) e uma redução horizontal (diminuindo a geração de EAE e de EAS). Os resultados provam que priorizar a redução horizontal tende a dar melhores resultados na MEF decomposta.

A técnica de agrupamento foi usada para gerar partições por agrupamento a partir da partição nula. Foram testados 4 *benchmarks* (*sand*, *planet*, *scfe* e *controller*) e 3 exemplos gerados aleatoriamente. Para os *benchmarks* observou-se sempre melhores resultados (em área e caminho crítico) para as partições geradas por fatoração e otimizadas por agrupamento em relação às partições geradas por agrupamento a partir da partição nula. Entretanto, observou-se uma menor diferença nos resultados de caminho crítico sendo que, no caso do *controller*, a diferença foi apenas de 7.41%.

Cada um dos exemplos gerados aleatoriamente (*blanca*, *mónica* e *aida*) apresentam diferentes complexidades. Por exemplo, o número de estados é de 600, 700 e 1000 respectivamente. Dada a complexidade destes exemplos, não foi possível obter resultados de síntese usando o sistema SIS. O problema aconteceu na codificação de estados. Foi usada uma estação de trabalho SUN-SPARC 20 com 64 Mbytes de memória. O programa AGRUP permitiu gerar facilmente um conjunto de partições para estes exemplos. Posteriormente, foram geradas as tabelas de transição de estados usando o programa GERTAB, e as MEFs decompostas foram sintetizadas usando o sistema SIS. Estes resultados mostram claramente a importância da técnica de decomposição que permite resolver o problema de síntese de MEFs complexas em casos em que não é possível resolvê-los pelos métodos usuais.

A decomposição recursiva foi aplicada ao exemplo *master*. Dado que o GERTAB detectou uma decomposição paralela cada sub-MEF foi decomposta novamente. É importante ver o fato de que este tipo de decomposição (paralela) aconteceu para um exemplo (o *master*) cujas entradas não tinham *don't cares*. Neste caso a possibilidade de gerar transições não-determinísticas fica menor devido ao alto número de combinações de entradas possíveis (2^{13} , para 13 entradas do *master*). A decomposição recursiva permitiu melhorar o caminho crítico em 9.67% para a sub-MEF₁. Entretanto, os resultados de área para as sub-MEF₁ e sub-MEF₂ e de caminho crítico para a sub-MEF₂ foram piores se comparados com a primeira decomposição. Um fato importante que pode ter impedido a melhora dos resultados é que o número de transições eliminadas por fusão foi zero na recursão e, portanto, a MEF decomposta não teve redução vertical.

Dado que os melhores resultados foram obtidos para decomposições usando fatoração seguida de agrupamento (para a geração de partições) e o GERTAB2 (para a geração de tabelas de transições de estados) percebe-se um claro compromisso entre qualidade dos resultados e o tempo gasto para obtê-los. Deve ficar claro que para este exemplo (o *controller*, com 353 estados) os tempos máximos de execução das ferramentas do DECMEF foram relativamente pequenos (menos de 15 minutos). Em todos os exemplos percebeu-se que o maior gargalo no processo da síntese foi a tarefa de mapeamento tecnológico. No caso do exemplo *seqbis1* este processo chegou a durar mais de 6 horas para as decomposições obtidas por fatoração não otimizadas.

As análises feitas anteriormente indicam que para exemplos de pequena complexidade o usuário pode tentar diferentes opções do DECMEF sem detrimento do tempo para obter soluções. Para os casos de média complexidade o usuário de DECMEF pode optar, em primeiro lugar, por tentar obter partições de tipo *Devadas* otimizando-as com o AGRUP (AGRUP é executado rapidamente) e depois tentar obter

partições do tipo *beta* otimizadas. Isto é justificado pelo fato de que fatores tipos *Devadas* e *beta* precisam de menores recursos computacionais e são obtidas em tempo menor.

Para exemplos de alta complexidade é recomendável para o usuário gerar as partições usando o programa AGRUP a partir da partição nula, já que a fatoração demonstrou ser difícil para estes casos e quando foi obtida para os exemplos de média complexidade foi um pouco lenta. Os diferentes experimentos indicaram que os tempos de execução do GERTAB1 são pequenos para exemplos complexos (tipicamente gastaram entre 10 e 20 minutos). O tempo de execução do GERTAB2 depende do tempo gasto na codificação de estados, sendo que a codificação de estados para os piores casos chegou a demorar até duas horas e meia aproximadamente (para o exemplo *master*).

12. REFERÊNCIAS BIBLIOGRÁFICAS

- [Abe92] J. Abe, N. Papavento. "Teoria Intuitiva dos Conjuntos". Makron Books, 1992, São Paulo, Brasil.
- [Bro73] C. Bron, J. Kerbosch. "Finding all Cliques of an undirected graph". Communication of ACM, September 1973.
- [Boo67] T. L. Booth. "Sequential Machines and Automata Theory". John Wiley and Sons, Inc. New York, London, Sydney. Capítulos 1, 2 e 3.
- [Bra84] R. K. Brayton, G. D. Hatchel, C. MacMullen, Sangiovanni-Vincentelli. "Logic Minimization Algorithms for VLSI Synthesis" Kluwer Academic Publishers, Boston, Massachusetts, 1984.
- [Bra87] R. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, A. Wang. "MIS: A Multi-level Logic Optimization System". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume CAD-6, pp. 106201081, November 1987.
- [Ash90] P. Ashar, S. Devadas, A. R. Newton. "A Unified Approach to the Decomposition and Redecomposition of Sequential Machines". 27th ACM/IEEE Design Automation Conference, pp. 601-606, June 1990.
- [Ash92] P. Ashar, S. Devadas, A. R. Newton. "Sequential logic Synthesis". Kluwer Academic Publishers, 1992.
- [Dev89] S. Devadas, A. R. Newton. "Decomposition and factorization of Sequential Finite State Machines". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 8, pp. 1206-1217, November 1989.
- [Gar79] M. R. Garey and D.S. Johnson. "Computer and Intratability, A Guide to the Theory of NP Completeness". W.H. Freeman, New York, second edition, 1979.
- [Har66] J. Hartmanins, R. E Stearn. "Algebraic Structure Theory of Sequential Machines". Prentice-Hall, Englewood Cliffs, Nj; Prentice Hall, 1966.
- [Has95] Z. Hasan, M. J. Ciesielski. "FSM Decomposition and Functional Verification of FSMs". VLSI Design: An International Journal of Custom-Chip Design, Simulation and Testing, Vol. 3, No 3-4, pp. 249-265, 1995.
- [Koh78] Z. Kohavi. "Switching and Finite Automata Theory". Computer Science Series, second edition 1978.
- [Kuk91] J. Kukula, S. Devadas. "Finite State Machine Decomposition By Transition Pairing". Procedures of ICCAD, pp. 414-417, November 1991.
- [LAG90] "Lager Tool Set". Manuais do sistema LAGER.

- [Lin89] B. Lin, R. Newton "Synthesis of Multiple Level logic from Symbolic High-Level Description Languages". IFIP International Conference on Very Large Scale Integration, pp 187-196, Munich, Germany, August, 1989.
- [LLa94] C. H. Llanos Quintero, M. Strum. "Uma Nova Técnica de Decomposição de Máquinas de Estados Finitos". Procedures of IX SBMICRO, Rio de Janeiro, pp 168-177, August 1994.
- [LLa95] C. H. Llanos Quintero, M. Strum. "DECMEF - An Interactive FSM Synthesis System". Proc. of X SBMicro, Canela, Brasil, pp. 59-68, August 1995.
- [LLa96] C. H. Llanos Quintero, M. Strum. "GERPAR and GERTAB2, Two FSM Decomposition Optimization Tools". Procedures of XI SBMICRO, Aguas de Lindoia, pp. 28-33, August, 1996.
- [LLa96b] C. H. Llanos Quintero e M. Strum. "GERTAB: uma ferramenta para Otimização de Máquinas de Estado Decompostas". IX SBCCI, pp. 59-68, Recife, março de 1996.
- [LLa96c] C. H. Llanos Quintero e M. Strum. "SINMEF: um Sistema para Decomposição de Máquinas de Estados Finitos". II IBEROCHIP, São Paulo, janeiro de 1996.
- [LLa97] C. H. Llanos Quintero e M. Strum. "Projeto de Máquinas de Estados Finitos Complexas". Procedures of III IBEROCHIP, Cidade do México, fevereiro de 1997.
- [LLa98] C. H. Llanos Quintero e M. Strum. "FSM decomposition using a extended factoring technique". ICIE98, Buenos Aires, abril de 1998.
- [McK56] E. J. McCluskey, "Minimization of Boolean functions". Bell Laboratories Technical Journal, volume 35, pp. 1417-1444, Bell Laboratories, November 1956.
- [Rus94] A. Russo, C. Llanos, M. Strum. "SSME, Um Sistema Simulador de Máquinas de Estados Finitos". IX SBMICRO, Rio de Janeiro, agosto de 1994.
- [Sau87] G. Saucier, M. Crastes de Paulet, P. Sicard. "ASYL: A Rule-Based System for Controllers Synthesis". IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, November 1987.
- [SIS92] A. Sangiovanni-Vicentelli et al. "SIS: A System for Sequential Circuits Synthesis". Electronics Research Laboratory, memorandum No. UCB/ERL M92/41. University of California, Berkeley, may 1992.
- [Syn95] Manuais do sistema Synopsys, 1995.
- [Vil90] T. Villa, A. Sangiovanni-Vicentelli. "NOVA: State Assignment of Finite States Machines for Optimal Two-Level Logic Implementations". IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, volume 9, pp. 905-924, Setemember 1990.
- [Yan93] W. L. Yang, R. Owens, M. Irwin. "Multi-way FSM Decomposition Based on Interconnect Complexity". Proc. of EuroDac, Hamburg, Germany, pp. 390-395, September, 1993.

BOLETINS TÉCNICOS - TEXTOS PUBLICADOS

- BT/PEE/9301 - Oscilador a HEMT - 10 GHz - FÁTIMA S. CORRERA, EDMAR CAMARGO
- BT/PEE/9302 - Representação Senoidal da Voz através dos Polos do Filtro Predictor - MARCELO B. JOAQUIM, NORMONDS ALENS
- BT/PEE/9303 - Blindagens por Grades Conductoras: Cálculo do Campo Próximo - LUIZ CEZAR TRINTINIALIA, ANTONIO ROBERTO PANICALI
- BT/PEE/9304 - Sistema de Otimização e Controle de Produção em Minas de Pequeno e Médio Porte - TSEN CHUNG KANG, VITOR MARQUES PINTO LEITE
- BT/PEE/9401 - Determinação das Frases de Aplicação Forense para o projeto NESPER e Tese de Mestrado IME/94, com Base em Estudos Fonéticos - MARCONI DOS REIS BEZERRA, EUVALDO F. CABRAL JUNIOR
- BT/PEE/9402 - Implementação e Teste de uma Rede Neural Artificial do Tipo KSON (Kohonen Self-Organizing Network) com Entradas Bidimensionais - MARCELO YASSUNORI MATUDA, EUVALDO F. CABRAL JR.
- BT/PEE/9403 - Transformada de Walsh e Haar Aplicadas no Processamento de Voz - ALEXANDRE AUGUSTO OTTATI NOGUEIRA, THIAGO ANTONIO GRANDI DE TOLOSA, EUVALDO F. CABRAL JÚNIOR
- BT/PEE/9404 - Aplicação de Redes Neurais ao Problema de Reconhecimento de Padrões por um Sonar Ativo - ALEXANDRE RIBEIRO MORRONE, CRISTINA COELHO DE ABREU, EDUARDO KOITI KIUKAWA, EUVALDO F. CABRAL JR.
- BT/PEE/9405 - Tudo que se Precisa Saber sobre a Prática da FFT - Transformada Rápida de Fourier (Inclui Software) - ROGÉRIO CASAGRANDE, EUVALDO F. CABRAL JR.
- BT/PEE/9406 - A Survey on Speech Enhancement Techniques of Interest to Speaker Recognition - CELSO S. KURASHIMA, EUVALDO F. CABRAL JR.
- BT/PEE/9407 - Identificação de Pulsos Decádicos em Linhas Telefônicas - ANTONIO P. TIMOSZCZUK, MÁRCIO A. MATHIAS, EUVALDO F. CABRAL JR.
- BT/PEE/9408 - Implementação e Teste de Filtros do Tipo Adaptativo e "Notch" para a Remoção de Interferência de 60 Hz em Sinais de Eletrocardiograma - FLÁVIO ANTÔNIO MENEGOLA, JOSÉ AUGUSTO DE MATTOS, JOSÉ GOMES G. FILHO, SIDNEY SILVA VIANA, EUVALDO F. CABRAL JR.
- BT/PEE/9409 - Compressão de Sinais de Voz utilizando Transformadas de Karhunen-Loève, Fourier e Hadamard - IVAN LUIS VIEIRA, LUIZ FERNANDO STEIN WETZEL, EUVALDO F. CABRAL JR.
- BT/PEE/9410 - "Ray Tracing" Paralelo - EDUARDO TOLEDO SANTOS, JOÃO ANTONIO ZUFFO
- BT/PEE/9411 - Implementação de uma Ferramenta Posicionador para "Gate-Arrays" Tipo Mar de Portas - JORGE W. PERLAZA PRADO, WILHELMUS A. M. VAN NOIJE
- BT/PEE/9412 - Tudo que se Precisa Saber Sobre a Teoria da FFT - Transformada Rápida de Fourier - FÁBIO LUÍS ROMÃO, REINALDO SILVEIRA, ROGÉRIO CASAGRANDE, EUVALDO CABRAL JR.
- BT/PEE/9413 - Análise do Ruído Sonoro em uma Sala de Aquisição de Amostras de Som com Microcomputador - FÁBIO LUÍS ROMÃO, REINALDO SILVEIRA, EUVALDO CABRAL JR.
- BT/PEE/9414 - Cor: Aspectos Relevantes para Visualização de Dados - SÍLVIA DELGADO OLABARRIAGA
- BT/PEE/9415 - Projeto de Filtros Digitais IIR com Fase Aproximadamente Linear Utilizando Redução de Ordem - IVAN F. J. RODRIGUES, MAX GERKEN
- BT/PEE/9416 - GERA-FILTRO: Sistema para Projeto Automático de Filtros Digitais "IIR" (da especificação em alto nível ao leiaute do "ASIC") - RICARDO PIRES, JOSÉ VIEIRA DO VALE NETO
- BT/PEE/9417 - Redes Neurais Artificiais Aplicadas à Identificação de Pulsos Decádicos em Linhas Telefônicas - ANTONIO P. TIMOSZCZUK, EUVALDO F. CABRAL JR.
- BT/PEE/9501 - Estudo Comparativo de Métodos de Cálculo da Frequência Fundamental - MARCOS COSTA HUNOLD, EUVALDO F. CABRAL JR.
- BT/PEE/9502 - Combinando Técnicas de Redes Neurais Artificiais e Informações de Excitação no Reconhecimento Automático do Locutor - ANDRÉ BORDIN MAGNI, EUVALDO F. CABRAL JR.
- BT/PEE/9503 - Utilização de Redes Neurais Artificiais para Detecção e Identificação de Falhas em Circuitos - MÁRCIO YUKIO TERUYA, ROBERTO AMILTON BERNARDES SÓRIA, EUVALDO CABRAL JR.
- BT/PEE/9504 - Uso de Redes Neurais Artificiais no Reconhecimento de Locutores no Domínio Temporal - BENEDITO JOSÉ BARRETO FONSECA JÚNIOR, EUVALDO CABRAL JÚNIOR
- BT/PEE/9505 - Projeto de Filtros Passivos e Ativos em Técnicas de Circuitos Integrados de Microondas - DAVID VIVEIROS JÚNIOR, DENISE CONSONNI
- BT/PEE/9506 - Uma Análise de *Clustering* para as Frases de Projeto NESPER - RONALDO OLIVEIRA MESSINA, EUVALDO F. CABRAL JR.
- BT/PEE/9507 - Controle com Estrutura Variável e Modos Deslizantes - Um Estudo para Aplicação em Controle Carga-frequência da Geração - JOSE PAULO F. GARCIA, JOCELYN FREITAS BENNATON
- BT/PEE/9508 - Recuperação das Margens de Ganho e de Fase para Sistemas de Fase Não Mínima por Realimentação da Saída - MARCO H. TERRA, VITOR M. P. LEITE
- BT/PEE/9509 - Sistema de Inspeção Óptica de Dispositivos Bi-Dimensionais - CASIMIRO DE ALMEIDA BARRETO, PEDRO LUÍS PRÓSPERO SANCHEZ
- T/PEE/9510 - Sistema de Partículas Uma Poderosa Técnica de Animação em Computação Gráfica - RENATO CURTO RODRIGUES, JOÃO ANTÔNIO ZUFFO
- BT/PEE/9511 - Efeito de Ruídos em Sinais de Voz Visualizados em Trajetórias Neurais de Kohonen - CELSO S. KURASHIMA, EUVALDO F. CABRAL JR.
- BT/PEE/9601 - "Um Reconhecedor de Sinais Sonoros Utilizando LVQ" - ALEXANDRE TORNICE, EUVALDO CABRAL JR.
- BT/PEE/9602 - "Coleção Artificial Neural Networks: Uma Visão Geral dos Sistemas Neurais Artificiais de Stephen Grossberg" - CHIU HSIUNG HUANG
- BT/PEE/9603 - "Reactively-Sputtered TiN Formation Using a RF Magnetron System" - SÉRGIO PAULO AMARAL OSÓRIO, LUIZ SÉRGIO ZASNICOFF
- BT/PEE/9604 - Aspectos em Tradução de Linguagens Naturais Através de Redes Neurais Artificiais - CARLOS EDUARDO DANTAS DE MENEZES, EUVALDO F. CABRAL JR.

- BT/PEE/9605 - Implementação de Blocos Passa-Tudo Utilizando Realimentação de Erro - SÉRGIO JOSÉ CARNEIRO LEÃO, MAX GERKEN
- BT/PEE/9606 - Coleção *SANN group* Redes Neurais Artificiais: A Rede Neural de Sakoe - ANDRÉ BORDIN MAGNI, EUVALDO F. CABRAL JR.
- BT/PEE/9607 - Coleção *SANN group* Redes Neurais Artificiais: A Rede Neural de Steinbuch - ROBERTO AMILTON BERNARDES SÓRIA, EUVALDO F. CABRAL JR.
- BT/PEE/9608 - Desenvolvimento de uma Estrutura de Duplo Nível de Metal para a Confeção de Interconexões em Circuitos Integrados - JOSE AUGUSTO DE ALENCAR PEREIRA, LUIZ CARLOS MOLINA TORRES
- BT/PEE/9609 - Determinação de Parâmetros de Processo para Fotomáscara "Balzers" Utilizando Gerador de Padrões - JORGE SEKI, MEGUMI SAITO
- BT/PEE/9610 - Um Ambiente para Desenvolvimento de Sistemas Distribuídos - PEDRO F. ROSA, JOÃO A. ZUFFO
- BT/PEE/9611 - Interpretações Teóricas do Funcionamento Cerebelar: Uma Revisão - MARCUS FRAGA VIEIRA, ANDRÉ FÁBIO KOHN
- BT/PEE/9612 - Marcapasso Cardíaco Temporário Microcontrolado de Demanda e Baixo Consumo - FLAVIO ANTONIO MENEGOLA, JOSÉ CARLOS TEIXEIRA DE BARROS MORAES
- BT/PEE/9613 - Um Sistema de Planejamento de Ação Baseado em Casos para uma Célula Flexível de Manufatura - RICARDO LUÍS DE FREITAS, MÁRCIO RILLO
- BT/PEE/9614 - Aplicações do Boundary-Scan para o Teste de Módulos Multichip - ROBERTO C. COSSI JR., JOSÉ ROBERTO DE A. AMAZONAS
- BT/PEE/9615 - A 2.488 Gb/s GaAs 1:4/1:16 Demultiplexer IC with Skip Circuit for Sonet STS-12/48 Systems - TAUFIK ABRÃO, FATIMA S. CORRERA
- BT/PEE/9616 - Uma Contribuição para a Construção de Algoritmos em Projetos de Redes - ALLAN DE SOUZA, JOSÉ ROBERTO CASTILHO PIQUEIRA
- BT/PEE/9617 - Análise Crítica dos Métodos de Medição do Intervalo QT do Eletrocardiograma - SÍDNEY DA SILVA VIANA, JOSÉ CARLOS TEIXEIRA DE BARROS MORAES
- BT/PEE/9618 - Deposição e Caracterização de Filmes de SiO₂ Crescidos pela Técnica de PECVD a Baixa Temperatura - MARCO ALAYO CHÁVEZ, INÉS PEREYRA
- BT/PEE/9619 - PARSTOOL: Uma Ferramenta de Auxílio à Simulação de Sistemas Paralelos - LI KUAN CHING, LIRIA MATSUMOTO SATO
- BT/PEE/9620 - Análise de um Método de Otimização por Malha no Treinamento de Robôs - OLÍMPIO MURILO CAPELI, JOSÉ CARLOS T. B. MORAES, SADA O ISOTANI
- BT/PEE/9701 - Identification of Unstable Mechanical Systems - ROBERTO MOURA SALES, ANSELMO BITTAR, MICHAEL PORSCH, LAÉRCIO LUCCHESI
- BT/PEE/9702 - Analysis of the Subthreshold Slope Transition Region in SOI nMOSFET - VICTOR SONNENBERG, JOÃO ANTONIO MARTINO
- BT/PEE/9703 - Introduction of the SOI MOSFET Dimensions in the High-Temperature Leakage Drain Current Model - MARCELO BELLODI, JOÃO ANTONIO MARTINO, DENIS FLANDRE
- BT/PEE/9704 - Controle de Largura de Banda Dinâmica para Transmissões *Multicast* para Redes de Alta Velocidade - SANG SOON LEE, SERGIO TAKEO KOFUJI
- BT/PEE/9705 - Uma Modificação Proposta para o Controle Preditivo Generalizado com Filtro de Kalman - JAIME QUINTERO R., OSWALDO L. V. COSTA
- BT/PEE/9706 - Aplicações de Redes Neurais em Previsões Financeiras - OLÍMPIO MURILO CAPELI, EUVALDO F. CABRAL JR.
- BT/PEE/9707 - Sistema Microcontrolado, Multicanal e Portátil para Estimulação Neuromuscular Funcional - ROGÉRIO QUIARIM ZARZA, JOSÉ CARLOS TEIXEIRA DE BARROS MORAES
- BT/PEE/9708 - Requisitos para o Mapeamento Tecnológico em Projetos de Microeletrônica - LUCIANO DE OLIVEIRA CORRÊA DE BRITO, JOSÉ ROBERTO DE ALMEIDA AMAZONAS
- BT/PEE/9709 - Sistemas PRMA com Dados Acoplados - JOSÉ AUGUSTO DE LIMA, PAUL JEAN ETIENNE JESZENSKY
- BT/PEE/9710 - Algoritmos Genéticos (AG's) para a Otimização de Controladores Nebulosos - JULIO CESAR CEBALLOS AYA, OSWALDO L. V. COSTA
- BT/PEE/9711 - Um Estudo Sobre a Redução de Ruídos em Sinais Caóticos - ERNANE JOSÉ XAVIER COSTA, EUVALDO F. CABRAL JR.
- BT/PEE/9712 - Geradores não Lineares de Sequência para uso em Sistemas Spread Spectrum - ANGEL ANTONIO GONZALEZ MARTINEZ, PAUL JEAN ETIENNE JESZENSZY
- BT/PEE/9713 - Modelamento Físico do Sistema Heteroestrutura - Metal - CECÍLIA WETTERLE RODRIGUES, MEGUMI SAITO
- BT/PEE/9714 - Tensões Induzidas em Linhas Aéreas por Descargas Atmosféricas Indiretas - Modelagem e Aplicação ao Cálculo de Interrupções - ALEXANDRE PIANTINI, JORGE M. JANISZEWSKI
- BT/PEE/9715 - RECMAP - Uma Ferramenta para Otimização em Síntese de Alto Nível Baseada em Reconhecimento Funcional e Mapeamento de Componentes - ANDRÉ GERHARD, JOSÉ VIEIRA DO VALE NETO
- BT/PEE/9716 - Estudo da Sinterização de Contatos Al/Ti por Recozimento Térmico Rápido Visando a Aplicação em Circuitos Integrados - ANGELO EDUARDO BATTISTINI MARQUES, ROGÉRIO FURLAN
- BT/PEE/9717 - Mixed H₂/H- Control of Discrete-Time Markovian Jump Linear Systems - OSWALDO L. V. COSTA, RICARDO P. MARQUES
- BT/PEE/9718 - Aluminium Etching with CCl₄-N₂ Plasmas - ANGELA MAKIE MAKAZAWA, PATRICK VERDONCK
- BT/PEE/9719 - O Uso de Resistes Amplificados Quimicamente e de Sililação em Litografia por Feixe de Elétrons - ANTONIO C. SEABRA, PATRICK B. VERDONCK
- BT/PEE/9720 - Implementação de um Simulador de um Circuito Neuro-Medular que Atua no Controle da Força Motora - LUIZ JURANDIR SIMÕES DE ARAÚJO, ANDRÉ FÁBIO KOHN
- BT/PEE/9721 - Avaliação das Características Físico-Químicas e Elétricas de Filmes de SiO₂ Depositados por PECVD a Partir da Reação entre O₂ e TEOS - ALVARO ROMANELLI CARDOSO E CLAUS MARTIN HASENACK
- BT/PEE/9722 - Controle e Simulação Dinâmica de Colunas de Destilação: Aplicação Prática em uma Coluna com Refluxo por Gravidade
- BT/PEE/9723 - Circuitos de Portas Lógicas Primitivas Implementados a Partir de uma Classe de Lógicas Paraconsistentes Anotadas - JOÃO INÁCIO DA SILVA FILHO, JAIR MINORO ABE, PEDRO LUÍS PRÓSPERO SANCHEZ

- BT/PEE/9724 - Lattice Heating and Energy Balance Consideration on the I-V Characteristics of Submicrometer Thin-Film Fully Depleted SOI NMOS Devices - CLAUDIA BRUNETTI, NELSON L. A. BRAGA, LUIZ S. ZASNICOFF
- BT/PEE/9725 - Identificação de um Processo de Neutralização de pH via Redes Neurais - SILVIO FLABOREA, CLAUDIO GARCIA
- BT/PEE/9726 - Uma Estratégia de Migração de Sistemas de Telefonia Móvel com Tecnologia AMPS para a Tecnologia CDMA - RONALD LUIS CLARKSON EISNER, PAUL JEAN ETIENNE JESZENSKY
- BT/PEE/9727 - Controle de pH Usando Conceitos de Invariantes de Reações e Geometria Diferencial - OSCAR A. ZANABARIA S., CLAUDIO GARCIA
- BT/PEE/9728 - Estudo da Influência dos Parâmetros de Recozimento Térmico Rápido na Morfologia dos Filmes de TiSi_2 Formados e sua Correlação com a Tensão Mecânica - SILVANA GASPAROTTO DE SOUZA, ARMANDO ANTONIO MARIA LAGANÁ, SEBASTIÃO GOMES DOS SANTOS FILHO
- BT/PEE/9729 - Analysis of Silicon Surface Microirregularities by LASER Light Scattering - JOSÉ CÂNDIDO DE SOUSA FILHO, SEBASTIÃO GOMES DOS SANTOS FILHO
- BT/PEE/9730 - Wavelets in Music Analysis and Synthesis: Timbres Analysis and Perspectives - REGIS ROSSI ALVES FARIA, RUGGERO ANDREA RUSCHIONI, JOÃO ANTONIO ZUFFO
- BT/PEE/9731 - Estudo de Efeitos Mútuos da Distribuição de Corrente em Condutores - AUGUSTO CARLOS PAVÃO, JORGE MIECZYSLAW JANISZEWSKI
- BT/PEE/9801 - Equivalência entre a Semântica da Lógica de Transações e a Semântica de sua Implementação Prolog - PAULO E. SANTOS, FLÁVIO S. C. DA SILVA
- BT/PEE/9802 - Nash Game in Mixed H_2/H_∞ Control Theory a Convex Optimization Approach - HELENICE OLIVEIRA FLORENTINO, ROBERTO MOURA SALES
- BT/PEE/9803 - Text-Independent Speaker Recognition Using Vector Quantization and Gaussian Mixture Models - THOMAS E. FILGUEIRAS F., RONALDO O. MESSINA E EUVALDO F. CABRAL JR.
- BT/PEE/9804 - Elementos Piezoresistivos para Sensores de Pressão com Tecnologia CMOS - LUIZ ANTONIO RASIA, E. C. RODRIGUEZ
- BT/PEE/9805 - Automação do Processo de Casamento de Impedância em Sistemas de Aquecimento por Microondas - J. C. DE SOUZA, J. T. SENISE, V. C. PARRO, F. M. PAIT
- BT/PEE/9806 - Considerações para o Projeto de Células de Memória SI com Transistores HEMT - JAIME H. LASSO, EDGAR CHARRY R.
- BT/PEE/9807 - Covariance Controller with Structure Constraint and Closed Loop H_∞ Bound - ANTÔNIO CARLOS DE LIMA, ROBERTO MOURA SALES
- BT/PEE/9808 - Controle Unidimensional de Objetos: Uma Aplicação Prática em Fresadora Automática - MÁRCIO A. F. MURATORE, OSWALDO L. V. COSTA
- BT/PEE/9809 - Redes Neurais com Retardos Temporais Aplicadas ao Reconhecimento Automático do Locutor - ROGÉRIO CASAGRANDE, EUVALDO F. CABRAL JR.
- BT/PEE/9810 - Topological Computation and Voluntary Control - HENRIQUE SCHÜTZER DEL NERO, JOSÉ ROBERTO CASTILHO PIQUEIRA, ALFREDO PORTINARI MARANCA
- BT/PEE/9811 - Casamento de Impedância em Guia de Onda Retangular Utilizando como Variável Medida a Potência Refletida - J. C. DE SOUZA JR., J. T. SENISE
- BT/PEE/9812 - Applied Surface Science - RONALDO D. MANSANO, PATRICK VERDONCK, HOMERO S. MACIEL
- BT/PEE/9813 - Contribuição ao Estudo da Morfologia da Superfície e da Interface do Siliceto de Titânio Formado sobre Si (100) Empregando a Técnica de Microscopia de Força Atômica (AFM) - N. M. HASAN, A. A. M. LAGANÁ, S. G. SANTOS FILHO
- BT/PEE/9814 - Estudo Experimental da Tensão Mecânica em Filmes Finos de Cobre Obtidos por Evaporação ou Deposição Eletroquímica Espontânea - A. I. HASHIMOTO, S. G. FILHO
- BT/PEE/9815 - Controle Híbrido de Manipuladores Robóticos - WEBER ALLEGRI, JOSÉ JAIME DA CRUZ
- BT/PEE/9816 - Entropia Informacional e Cronobiologia - ANA AMÉLIA BENEDITO SILVA, JOSÉ ROBERTO CASTILHO PIQUEIRA
- BT/PEE/9817 - Estabilidade de Lyapunov e Controle de Atitude - FERNANDO SOUSA, FREITAS JÚNIOR, PAULO SÉRGIO PEREIRA DA SILVA
- BT/PEE/9818 - Projeto de um Conversor de Frequências Resistivo em Tecnologia MMIC - CLÁUDIA C. A. APARÍCIO, DENISE CONSONNI
- BT/PEE/9819 - Estudo de Sensibilidade de um Sistema de Modulação Digital via Simulação - RONALDO DI MAURO, LUIZ ANTONIO BACCALÁ
- BT/PEE/9820 - Engenharia do Conhecimento Aplicada ao Conhecimento Aplicada ao Domínio de Gerenciamento de Falhas em Redes de Comunicação: Uma Abordagem Baseada em Modelo - MARILZA ANTUNES DE LEMOS, MARCIO RILLO
- BT/PEE/9821 - Uma Biblioteca de Métodos de Resolução de Problemas de Planejamento para Fornecer Apoio à Aquisição de Conhecimento - LELIANE NUNES DE BARROS, MARCIO RILLO
- BT/PEE/9822 - Um Estudo de Técnicas de Aprendizado por Reforço Livre de Modelo - Aplicação ao Pêndulo Invertido - SÉRGIO RIBEIRO AUGUSTO, ADEMAR FERREIRA
- BT/PEE/9823 - Identificação de Sistemas Dinâmicos com Redes Neurais - FRANCISCO CARLOS PONS, CLÁUDIO GARCIA
- BT/PEE/9824 - Comparison between Single and Double Langmuir Probe Techniques for Analysis of Inductively Coupled Plasmas - RAUL M. DE CASTRO, GIUSEPPE A. CIRINO, PATRICK VERDONCK, HOMERO S. MACIEL, MARCOS MASSI, MARCELO B. PISANI, RONALDO D. MANSANO

