

85-430
FL-933

CORPO EDITORIAL: Antônio Carlos da Rocha Costa
Carla Maria Dal Sasso Freitas

SIMULAÇÃO DE SISTEMAS MODELADOS
COMO REDES DE AGÊNCIAS

por

Flávio Rech Wagner

RT nº 027

CPGCC-UFRGS

NOV/85

Nota Técnica do projeto "BANCO DE DADOS E
FERRAMENTAS PARA CAD DE SISTEMAS DIGITAIS"

Trabalho realizado com o apoio do CNPq

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Av. Osvaldo Aranha, 99
90.210-Porto Alegre-RS-Brasil
Telex (051)2680 Fone: 21.8499

Endereço para Correspondência:

UFRGS/CPGCC/Biblioteca
Cx. Postal 1501
90.001-Porto Alegre-RS-Brasil



UFRGS

SABi



05233482

UFRGS
BIBLIOTECA
CPD/CPGCC

Simulação

Redes Agências

UFRGS
CPD - PGCC
BIBLIOTECA

Lo CHAMADA: FL 0933		No REG: 31856
		DATA: / /
ORIGEM: D	DATA: 26/11/85	PREÇO: CPA 30.000
UNDO: CPD/PGCC	FORM.: FLAVIO WAGNER	

RESUMO

Este trabalho apresenta um modelo de simulação, em um monoprocessador, de sistemas representados como redes de agências. A estrutura básica de um simulador adequado a este modelo é mostrada. Alguns aspectos importantes da implementação de um simulador de redes de agências de propósitos gerais são analisados em mais detalhe.

PALAVRAS-CHAVE: Simulação, sistemas discretos, rede de agências.

ABSTRACT

This work presents a model for simulation in monoprocessors of systems represented as nets of agencies. The basic structure of a simulator, which is suitable for this model, is shown. Some relevant aspects of the implementation of a general purpose simulator for nets of agencies are considered in more detail.

KEYWORDS: Simulation, discrete systems, nets of agencies.

SECRET

CONFIDENTIAL

SECRET

1. The information contained in this document is classified as SECRET because it is information the disclosure of which would be injurious to the national defense.

2. This information is being disseminated to you for your information and use in the performance of your duties.

3. It is the policy of the Department of Defense to limit the dissemination of this information to those personnel who have a valid "need-to-know" and are authorized to receive it.

4. You are authorized to disseminate this information to those personnel who have a valid "need-to-know" and are authorized to receive it.

5. You are prohibited from disseminating this information to any other personnel who do not have a valid "need-to-know" and are not authorized to receive it.

6. If you are a contractor, you are prohibited from disclosing this information to any other personnel who do not have a valid "need-to-know" and are not authorized to receive it.

7. If you are a contractor, you are prohibited from disclosing this information to any other personnel who do not have a valid "need-to-know" and are not authorized to receive it.

8. If you are a contractor, you are prohibited from disclosing this information to any other personnel who do not have a valid "need-to-know" and are not authorized to receive it.

9. If you are a contractor, you are prohibited from disclosing this information to any other personnel who do not have a valid "need-to-know" and are not authorized to receive it.

10. If you are a contractor, you are prohibited from disclosing this information to any other personnel who do not have a valid "need-to-know" and are not authorized to receive it.

SECRET

CONFIDENTIAL

SECRET

SUMÁRIO

1. INTRODUÇÃO	01
2. CONCEITOS BÁSICOS	02
2.1 Simulação de Sistemas Modelados como Redes de Agências Programadas	02
3. SIMULAÇÃO DE REDES DE AGÊNCIAS	06
3.1 Modelo de Simulação	06
3.2 Exemplo de modelamento de um sistema	09
3.2.1 Descrição do Sistema	09
3.2.2 Descrição do Modelo	10
3.2.3 Avaliação da Eficiência do Modelo	13
4. IMPLEMENTAÇÃO DE UM SIMULADOR DE REDES DE AGÊNCIAS DE PROPÓSITOS GERAIS	15
4.1 Modelamento Indireto de Sistemas	15
4.2 Estrutura de Dados para Representação das Interconexões da Rede	16
4.3 Lista de Eventos	18
4.4 Lista de Chamada e Sequência de Chamada	19
BIBLIOGRAFIA	21

1875-1876

The first of these is the fact that the
 population of the district has increased
 considerably since the year 1860. This
 increase is due to the fact that the
 district has become more fertile and
 more productive since that time. The
 increase in the population is also due
 to the fact that the district has
 become more attractive to immigrants
 from other parts of the country. The
 increase in the population is also due
 to the fact that the district has
 become more attractive to immigrants
 from other parts of the country.

1876-1877

The second of these is the fact that
 the population of the district has
 increased considerably since the year
 1860. This increase is due to the
 fact that the district has become
 more fertile and more productive
 since that time. The increase in the
 population is also due to the fact
 that the district has become more
 attractive to immigrants from other
 parts of the country. The increase
 in the population is also due to the
 fact that the district has become
 more attractive to immigrants from
 other parts of the country.

1877-1878

The third of these is the fact that
 the population of the district has
 increased considerably since the year
 1860. This increase is due to the
 fact that the district has become
 more fertile and more productive
 since that time. The increase in the
 population is also due to the fact
 that the district has become more
 attractive to immigrants from other
 parts of the country.

1878-1879

The fourth of these is the fact that
 the population of the district has
 increased considerably since the year
 1860. This increase is due to the
 fact that the district has become
 more fertile and more productive
 since that time. The increase in the
 population is also due to the fact
 that the district has become more
 attractive to immigrants from other
 parts of the country.

1879-1880

1. INTRODUÇÃO

O modelo de redes de agências, introduzido em /WEN 83/ e /WEN 79a/, foi apresentado em trabalhos anteriores /WAG 84a/ /WAG 84b/. Este modelo permite fundamentalmente a representação de sistemas onde se desenrolam processos digitais, i.e., cujas variáveis assumem apenas valores discretos e têm transições de estado apenas em tempos discretos. Este modelo tem sido utilizado com sucesso na representação de diferentes aspectos de sistemas digitais, programáveis ou não. O modelo permite que um sistema computacional seja representado tanto nos seus aspectos de software como de hardware.

Devido à sua abrangência na representação de sistemas digitais, o modelo é potencialmente indicado para servir de base a um ambiente integrado de projeto auxiliado por computador. Num tal ambiente, um sistema pode ser inicialmente especificado em seus aspectos funcionais mais abstratos, sem que se decida sobre uma implementação de software ou de hardware para seus componentes. Num processo de refinamentos sucessivos, mas mantendo sempre a representação do sistema como uma rede de agências, aos poucos são tomadas decisões de implementação que dão à representação uma natureza cada vez mais estrutural.

Este modelo, em função desta sua potencialidade, será usado como alternativa de especificação de sistemas digitais num ambiente de CAD em desenvolvimento no CPGCC da UFRGS /WAG 85/. O ambiente deverá incluir um editor gráfico, cuja especificação se encontra em /DAL 85/, e um simulador.

O objetivo deste trabalho é justamente apresentar um modelo de simulação adequado para redes de agências, e entrar em alguns detalhes de implementação relevantes.

2. CONCEITOS BÁSICOS

2.1. Simulação de sistemas modelados como redes de agências programadas

Em um trabalho anterior /WAG 84a/ foi apresentado o conceito de redes de agências programadas. Foi mostrado que, como resultado da partição de um sistema programado em componentes, resultam agências programadas que se comunicam entre si através de marcas e/ou de operandos, sendo estas formas de comunicação derivadas da representação da estrutura de causalidade do sistema através de uma rede de Petri. A Fig. 1 mostra um modelo interno adequado para explicar o comportamento de uma agência programada, contendo uma unidade de execução, uma memória P para o programa que a agência executa, e uma memória M para a marcação da rede de Petri, que descreve o comportamento da agência.

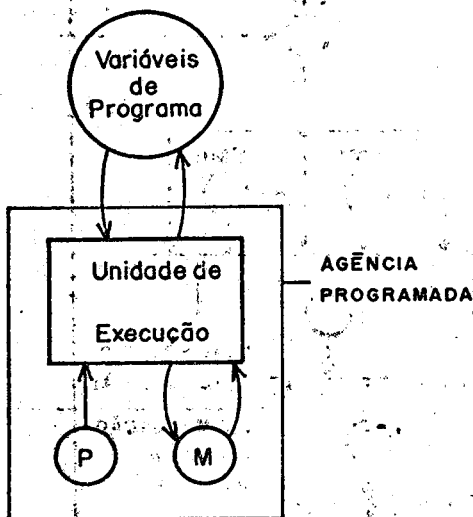


Figura 1. - Agências programadas.

Uma rede de agências programadas implica na existência de várias unidades de execução agindo em paralelo no sistema, uma para cada agência. Se desejamos modelar este sis

tema para fins de simulação em um monoprocessor, precisamos acrescentar à rede de Petri que explica o comportamento do sistema um novo fluxo de marcas, que corresponde à seqüencialização das atividades das agências. Uma diferenciação das marcas em tipos diversos é útil à compreensão do modelo /WEN 83/: a marca de processador indica com quem está o controle do processador hospedeiro, onde a simulação se desenrola, enquanto uma marca de seqüência existe para cada agência, indicando em que ponto SM_i do seu programa a agência se encontra. Este é o ponto a partir do qual a agência-modelo deve executar seu programa, quando ela recebe a marca de processador. Neste momento ambas as marcas se confundem numa só, até que a agência entregue o controle do processador hospedeiro a outra agência do sistema. A Fig. 2 mostra a estrutura de causalidade de uma agência programada genérica num modelo de simulação apropriado para um monoprocessor.

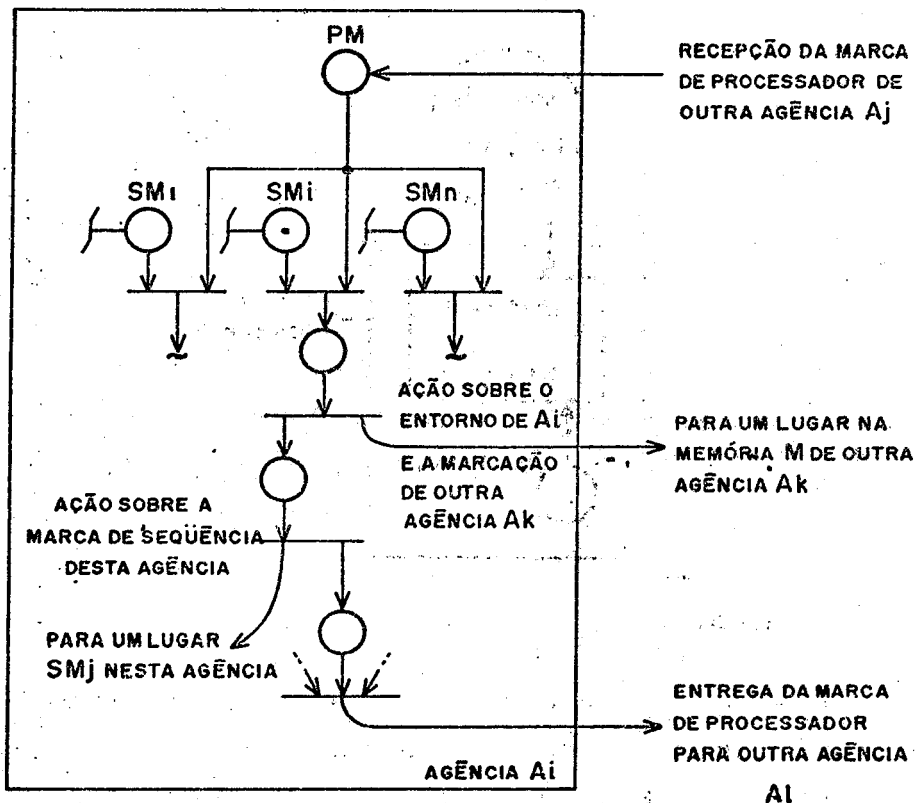


Figura 2. - Estrutura de causalidade de agências programadas.

A sequencialização das atividades das agências é obtida pela introdução de uma agência de coordenação, o coordenador, e pelo modelamento da atividade contínua das agências do sistema real através de uma seqüência de ações instantâneas, que não consomem tempo de modelo /WAG 84b/, obtendo-se para cada agência uma operação "start-stop" /WEN 79b/.

Através da Fig. 2 podemos ver como se desenrola uma ação de uma agência Ai e como ela se comunica com a agência de coordenação: a agência Ai recebe a marca de processador de coordenador, executa uma ação de acordo com sua função e com o seu estado atual (i.e., de acordo com a posição SMi da marca de seqüência), e devolve a marca de processador incondicionalmente ao coordenador. A ação de Ai afeta outras agências de duas formas possíveis: alterando uma variável de saída de Ai que seja entrada para outra agência Aj (comunicação por operandos), ou alterando diretamente a marcação de uma agência Ak (comunicação por marcas). Após a devolução da marca do processador ao coordenador, a agência pode permanecer ativa ou passiva (no sentido de SIMULA /SIM 73/). A agência está ativa quando deve ser novamente chamada pelo coordenador após um intervalo de tempo conhecido, que corresponde no sistema real à execução de alguma atividade que consome tempo (por exemplo comando HOLD em SIMULA). A agência fica passiva se não tem nenhuma atividade futura planejada, sendo ativada apenas por ação de outra agência (p.ex. Comando ACTIVATE em SIMULA).

O coordenador tem então as seguintes funções principais:

a) função de "dispatch", i.e., determinar uma seqüência de chamada para as agências que têm ações a executar num mesmo tempo de modelo;

b) sincronização entre as agências, i.e., imple-

mentação da comunicação por marcas através de algum mecanismo de sincronização;

c) implementação da comunicação por operandos, através da ativação das agências cujas entradas sofreram alteração por ação de outras agências; e

d) incremento do tempo de modelo, quando no tempo atual não há mais nenhuma agência em condições de executar uma ação.

Quando é feita a decomposição de um sistema em agências, para fins de simulação, a função de cada agência e as relações entre as diversas agências devem ser definidas de acordo com a existência no sistema de simulação de um dado coordenador, que implementa de alguma forma as 4 funções citadas acima /POT 82/.

3. SIMULAÇÃO DE REDES DE AGÊNCIAS

3.1. Modelo de simulação

O modelo de simulação de redes de agências a ser aqui adotado /WAG 83/ /WAG 84b/ pode ser explicado através da divisão de tarefas entre as agências e o coordenador. De acordo com a definição de redes de agências, existe apenas comunicação por operandos entre as agências (comunicação por marcas existe apenas entre agências programadas, que são uma generalização do conceito). O Coordenador apoia esta comunicação por operandos, por conhecer as interconexões de rede e poder identificar as agências que são afetadas pela alteração de uma variável do sistema causada pela ação de outra agência.

Uma agência é chamada pelo coordenador se uma de duas condições é preenchida: a) o tempo de modelo chegou, no qual a agência precisa executar uma ação de acordo com sua função (condição T); e b) uma variável de entrada da agência foi alterada pela ação de outra agência (condição E).

Uma agência deve então ao final de uma ação informar ao coordenador a) se ela alterou alguma variável de saída (mensagem de alteração de entorno) e b) em que tempo de modelo futuro ela deve ser novamente chamada (mensagem de tempo), caso permaneça ativa e não passiva.

O coordenador tem três tarefas a executar em cada tempo de modelo: a) ativar as agências para as quais uma condição E foi preenchida como consequência de uma alteração numa variável de entrada; b) chamar as agências, para as quais uma condição T é preenchida no tempo de modelo atual; e c) se nenhuma outra agência tem uma ação a executar no tempo atual, incrementar o tempo até que uma condição T

seja preenchida.

A agência deve devolver a marca de processador ao coordenador apenas quando não tiver mais nenhuma atividade a executar no tempo atual. O coordenador pode chamar uma agência por um de dois motivos, ou por dois motivos simultaneamente (condições E e T). As agências não recebem nenhuma indicação do coordenador a respeito da condição que causou a chamada. Elas próprias devem armazenar seu estado atual em variáveis de estado, e decidir, em função deste estado, do conteúdo das variáveis de entrada e do valor do tempo de modelo atual, qual ação deve ser executada. Isto significa que as próprias agências administram a marca de seqüência, decidindo a ação a ser executada em função da marcação da rede de Petri que especifica a estrutura de causalidade da agência. A fig. 3 mostra a estrutura de causalidade de uma agência genérica, de acordo com este modelo, e a comunicação entre a agência e o coordenador.

Este modelo é fundamentalmente diferente do modelo adotado em SIMULA, por exemplo, onde cabe ao coordenador administrar a marca de seqüência das agências. A fig. 4 mostra a estrutura de causalidade de uma agência genérica de acordo com SIMULA. A descrição da função da agência corresponde a uma série de opções. Os lugares SM_i , que definem a posição atual da marca de seqüência, ficam sob administração do coordenador, que ao chamar uma agência entrega não só a marca de processador como também a marca de seqüência. No modelo de simulação de redes de agências, a agência recebe a marca de processador sempre no mesmo lugar de entrada.

Há uma outra diferença fundamental entre o modelo de simulação aqui adotado e a visão de SIMULA, que não é aparente nas figuras 3 e 4. Em SIMULA é possível a comunicação por marcas entre as agências, caracterizando-se assim agên

cias programadas.

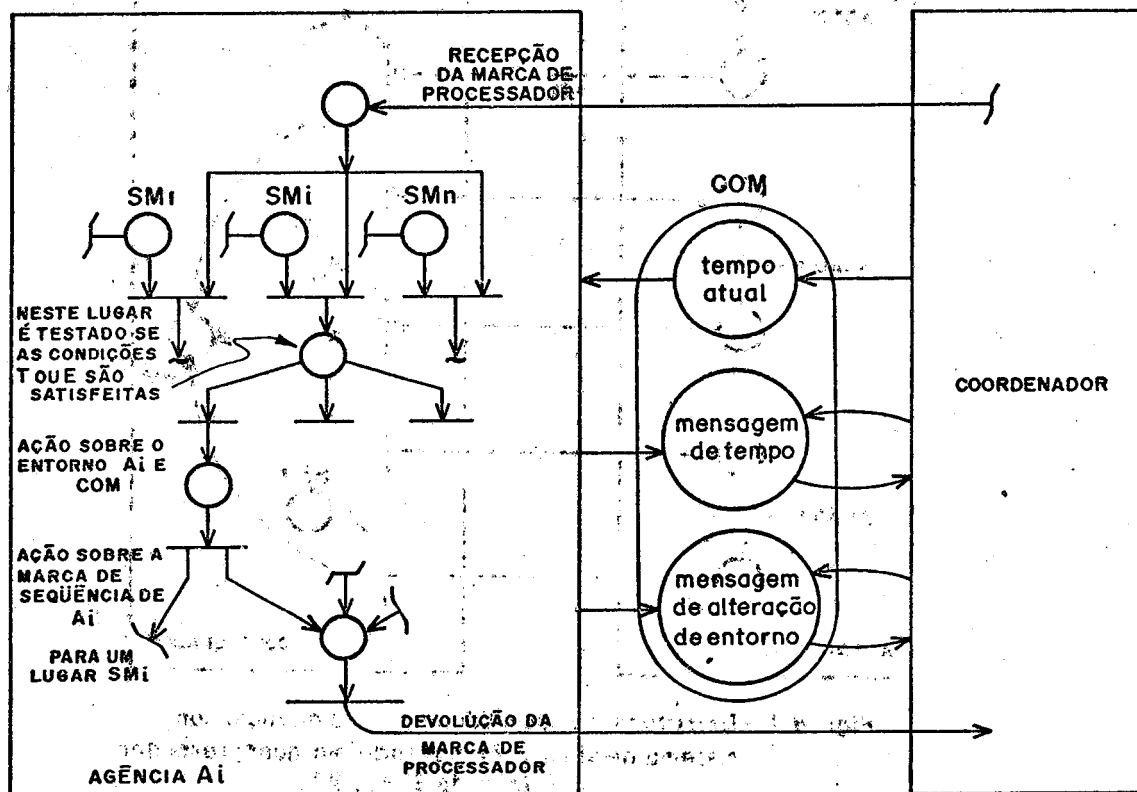


Figura 3. - Estrutura de causalidade de uma agência num sistema de simulação baseado em monoprocessador e comunicação com o coordenador.

Deve ficar claro ser este um dos modelos de simulação possíveis para redes de agências, que supõe uma dada divisão de tarefas entre o coordenador e as agências. Esta divisão de tarefas tem implicações decisivas no modelo a ser feito de um sistema real para fins de utilização do simulador. Este modelamento deve sempre seguir a estrutura de causalidade genérica da fig. 3. Na próxima seção veremos um exemplo concreto de modelamento. Outros modelos de simulação, supondo diferentes divisões de tarefas entre coordenador e agências, são também possíveis.

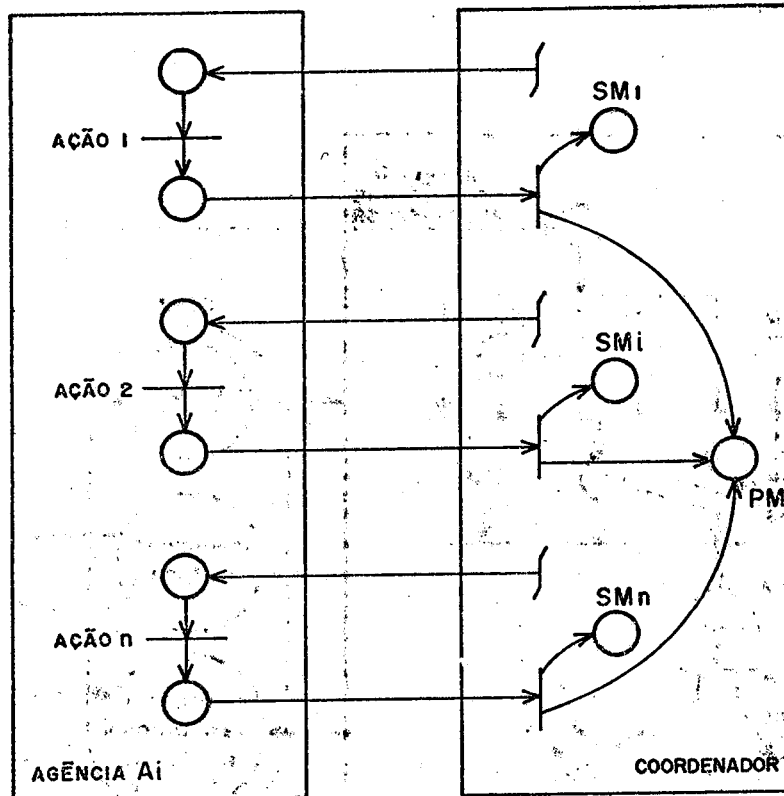


Figura 4. - Estrutura de causalidade de uma agência num sistema de simulação baseado em monoprocessador (visão de SIMULA).

3.2 Exemplo de modelamento de um sistema

3.2.1 Descrição do sistema

Todo sistema que é modelado para fins de compreensão (e análise, p.ex. através de simulação) é na realidade uma abstração de um sistema real, onde só são consideradas características relevantes à análise pretendida. O sistema-objeto é então uma visão parcial de um sistema real. Como exemplo neste trabalho usaremos os sistemas digitais como sistemas reais. O sistema-objeto será um sistema digital representado no nível de portas lógicas. Para que facilitemos o modelo a ser desenvolvido, iremos limitar ainda mais este sistema-objeto, supondo que as portas lógicas têm o seguinte comportamento sim

plificado: avaliar as condições de funcionamento de uma porta lógica. A primeira hipótese é a seguinte: - uma porta lógica altera sua saída exatamente n unidades de tempo após a transição de entrada que provocou a alteração, sendo n constante para cada porta, e independente do sentido da transição do sinal de saída;

- uma porta lógica comporta-se com um elemento ideal com respeito ao "atraso" de propagação, i.e., transições consecutivas na entrada causam transições correspondentes na saída, não havendo interferência entre transições consecutivas (a porta não tem comportamento inercial), e com isto podemos ter num dado instante várias transições futuras do sinal de saída já programadas para a porta lógica;

- os sinais são exclusivamente binários, sendo a transição entre um estado e outro instantânea.

Tais suposições certamente não correspondem a um sistema digital real, porém são consideradas válidas no nível de abstração desejado para o modelamento, em função dos propósitos da análise pretendida.

Serão descritos a seguir dois modelos diferentes para estes sistemas, através dos quais se ganhará uma idéia concreta sobre as implicações e restrições do esquema de simulação adotado.

3.2.2 Descrição do modelo

Cada porta lógica é representada por uma agência, como na Fig. 5, possuindo n sinais de entrada X_1, \dots, X_n e 1 sinal de saída Y , todos eles modelados como variáveis ("memórias") da rede. Note-se que no sistema real um sinal lógico não tem a característica de uma "memória", po-

rém pode ser observado de forma permanente. "Memórias" em uma rede de agências são justamente as variáveis do problema a serem observadas. A agência tem ainda acesso a variáveis de estado, cujas funções serão explicadas adiante.

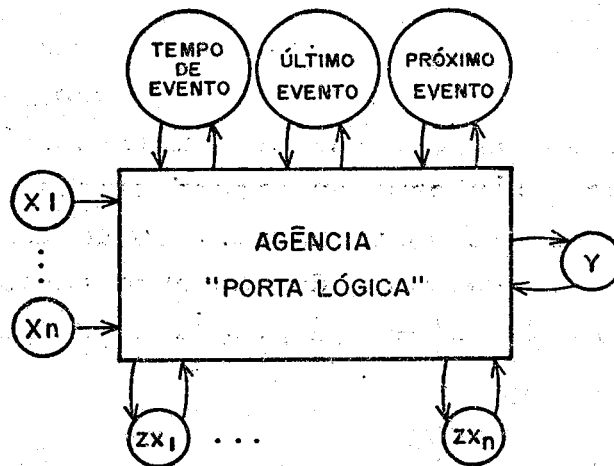


Figura 5. - A agência "Porta Lógica" e seus acessos a variáveis de E/S e de estado (modelo I).

Quando uma agência "porta lógica" é chamada pelo coordenador devido a uma alteração de entorno, e verifica que daí decorre uma transição no sinal de saída, comunica esta transição futura ao coordenador através de uma "mensagem de tempo". Transcorrido o tempo comunicado, e que corresponde ao atraso de propagação da porta, ela voltará a ser chamada pelo coordenador. Nota-se então a existência de 2 motivos diferentes para a chamada da agência:

a) chegou o tempo de efetivar uma alteração no sinal de saída, programada anteriormente, e b) houve uma alteração na entrada e a função lógica deve ser avaliada para se decidir se daí decorre uma transição futura no sinal de saída.

No modelo de simulação adotado, cabe à própria agência decidir, em função do seu estado atual e do valor das variáveis às quais ela tem acesso, qual a ação a ser tomada. O motivo da chamada não é comunicado à agência pelo coordenador. Os dois motivos podem inclusive se sobrepor num dado tempo t_i , se uma alteração na saída coincidir com uma nova transição numa entrada da porta lógica.

Para determinar a ação a ser tomada (i.e., administrar a marca de seqüência, ver seção 3.1), a agência possui as variáveis de estado ZX_1 a ZX_n , TEMPO-DE-EVENTO, ÚLTIMO-EVENTO e PRÓXIMO-EVENTO. ZX_1 a ZX_n armazenam o valor anterior das variáveis correspondentes dos sinais de entrada X_1 e X_n , sendo atualizadas sempre que ocorre uma nova transição. TEMPO-DE-EVENTO é um array, implementado como uma fila circular, que contém em cada nodo o tempo de uma transição futura no sinal de saída. ÚLTIMO-EVENTO é o índice do nodo desta lista onde deve ser inserido um novo tempo quando uma nova transição de saída é programada, e PRÓXIMO-EVENTO é o índice do nodo que tem o valor do tempo para a próxima transição de saída.

Quando a agência é chamada, ela determina se chegou o tempo de alguma transição futura da variável de saída, testando se ÚLTIMO-EVENTO \neq PRÓXIMO-EVENTO, e, em caso afirmativo, comparando TEMPO-DE-EVENTO [PRÓXIMO-EVENTO] com TEMPO-CORRENTE, que é uma variável global do simulador. Independentemente de ter chegado o tempo de uma transição na saída, a agência deve testar se houve alguma transição numa das variáveis de entrada, o que é feito comparando-se X_1 com ZX_1 , X_2 com ZX_2 , etc.

A fig. 6 mostra o algoritmo que descreve o comportamento de uma agência "porta lógica". As ações a serem tomadas em cada caso (chamada por mensagens de tempo ou por mensagem de alteração de entorno) são simples e

podem ser compreendidas facilmente a partir da função das variáveis envolvidas. A figura não mostra as ações a serem tomadas quando da inicialização da simulação.

3.2.3 Avaliação da eficiência do modelo

Certamente um simulador construído especificamente para sistemas digitais representados no nível de portas lógicas será mais eficiente, em termos de gasto de memória e de tempo de simulação, do que um simulador genérico de redes de agências utilizado para o mesmo fim.

Um trabalho bastante exaustivo foi realizado /WAG 83/ para examinar a relação entre o tempos consumidos por um simulador genérico de redes de agências e simuladores específicos de hardware, quando simulado um mesmo sistema. Este estudo considerou a simulação de hardware nos níveis de transferência entre registradores e de portas lógicas. Este estudo provou que, para qualquer sistema digital que se tome, um simulador de redes de agências consumirá entre 2,1 a 2,55 vezes mais tempo de simulação do que um simulador de portas lógicas, e de 1,2 a 3,4 vezes mais tempo de simulação do que um simulador para o nível de transferência entre registradores.

Esta relação de tempos diz respeito unicamente ao algoritmo principal de simulação. Considerando-se que um simulador real consome a maior parte de seu tempo em atividades de entrada e saída e de controle e monitoração, que são idênticas para simuladores de hardware e de redes de agências, chega-se à conclusão de que a perda no tempo durante a execução do algoritmo principal de simulação não é significativa, especialmente num ambiente interativo de projeto.

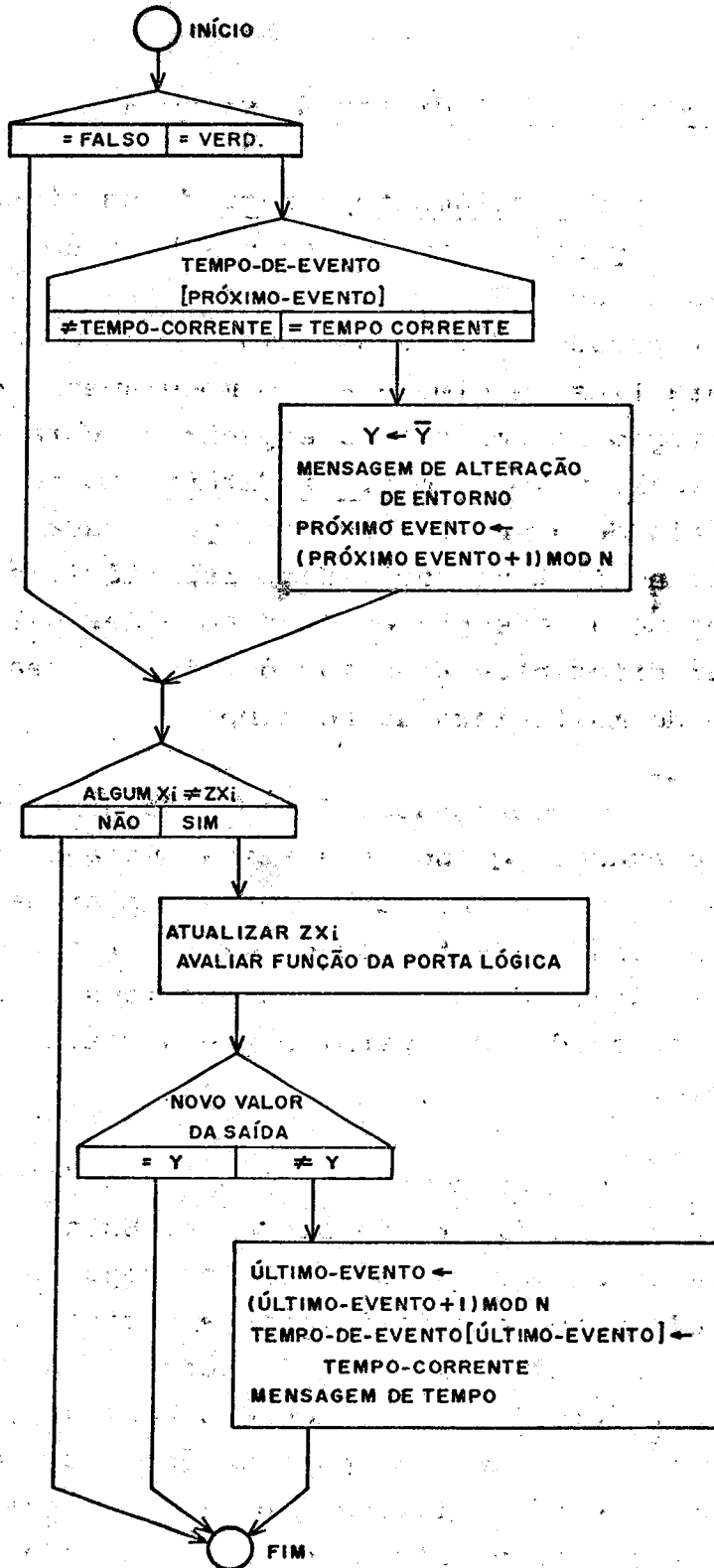


Figura 6. - Algoritmo de descrição do comportamento da agência "Porta Lógica, segundo o modelo I.

4. IMPLEMENTAÇÃO DE UM SIMULADOR DE REDES DE AGÊNCIAS DE PROPÓSITOS GERAIS

4.1 Modelamento indireto de sistemas

O modelamento direto de um sistema através de uma rede de agências implica em que haja um mapeamento 1:1 entre as agências do sistema real e as do modelo desenvolvido para representar este sistema. Se este modelo p.ex. é usado para fins de simulação, o modelamento direto implica na construção de uma rotina específica para cada agência do sistema real. O modelamento indireto, ao contrário, supõe um repertório de agências básicas, para cada tipo que venha a ocorrer no sistema real. Este repertório é então constituído por agências exemplares, que em diferentes momentos irão encarnar diferentes agências do sistema real, segundo um princípio de multiplexação no tempo.

O coordenador deve então gerenciar esta função de multiplexação. Quando ele ativa uma agência, ele na realidade chama uma rotina genérica, exemplar para o tipo em questão. Passando a esta rotina exemplar os nomes das variáveis de entrada/saída e de estado da agência específica a ser ativada, o coordenador permite que a rotina assuma o papel da agência.

O simulador descrito nas próximas seções supõe o modelamento indireto de sistemas. Esta decisão é indispensável se desejamos que o simulador seja de propósitos gerais, pois ela permite a definição de rotinas genéricas para os tipos de agências possíveis de ocorrer (no exemplo da seção 3.2, um repertório de portas lógicas básicas), sem o que um simulador seria apropriado para um único sistema e não para toda uma classe de sistemas (p.ex. a classe dos sistemas digitais descritos no nível lógico de acordo com o modelo apresentado na seção 3.2.1).

4.2 Estrutura de dados para representação das interconexões da rede

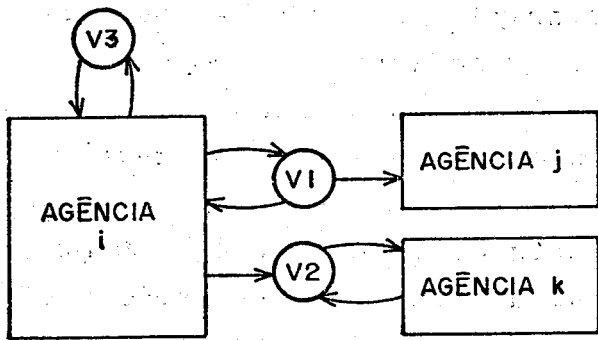
Se tivéssemos optado pelo modelamento direto de sistemas como redes de agências cada rotina corresponderia a uma agência específica do sistema real e poderia ser construída com o conhecimento prévio das interconexões desta agência, i.e., dos nomes reais das suas variáveis de entrada/saída. Como optamos pela construção de rotinas que correspondam a um repertório básico de agências genéricas, estas rotinas não conhecem em princípio os diversos entornos nos quais elas serão utilizadas. Cabe então ao coordenador ter acesso a uma estrutura de dados, onde estão armazenadas informações a respeito das interconexões do sistema.

Para a definição desta estrutura, três propriedades das redes de agências são decisivas: 1) a rede não é dirigida, i.e., variáveis de ligação podem ser bidirecionais, tendo uma agência direitos de acesso a estas variáveis que variam no tempo (direito de leitura num dado estado do sistema, direito de escrita em outro); 2) cada agência pode ter um número qualquer de variáveis de ligação e de estado; 3) as variáveis de ligação e de estado podem ser de qualquer tipo.

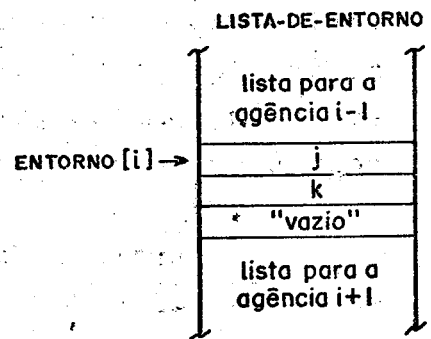
Note-se que estas são propriedades de redes de agências em geral, que devem ser respeitadas se queremos construir um simulador de propósitos gerais. Num simulador orientado p.ex. exclusivamente para sistemas digitais poderíamos fazer suposições bem mais restritivas a respeito das interconexões na rede, otimizando assim a estrutura de dados e as rotinas que a manipulam.

A figura 7 mostra uma estrutura de dados que atende aos requisitos aqui colocados. A justificativa detalhada para esta estrutura é dada em /WAG 83/. Para cada tipo de

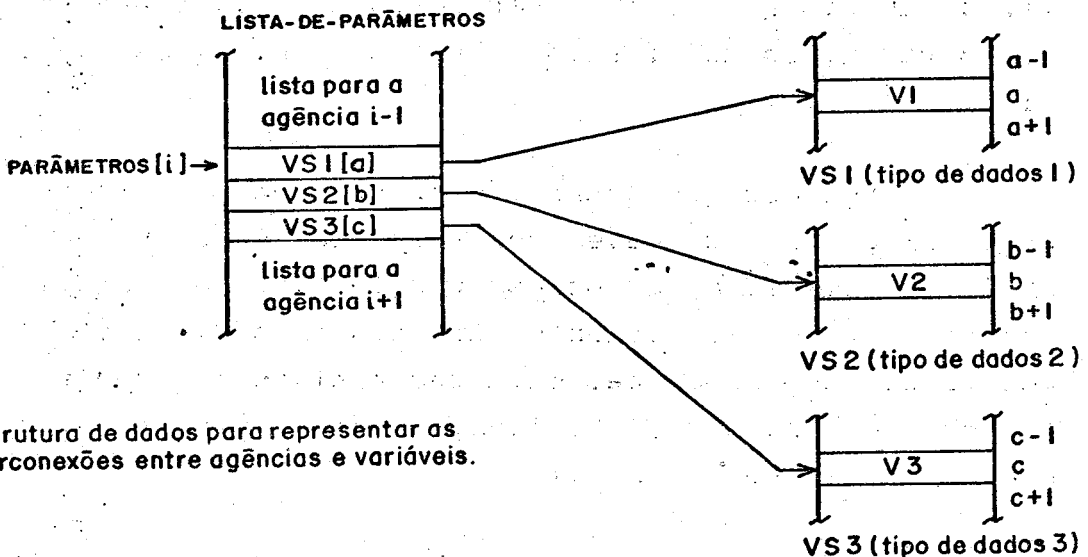
dados encontrável dentre as variáveis de ligação e de estado existe um array VS (Valor dos Sinais). A cada agência i está associada parte de uma LISTA-DE-PARÂMETROS, cujo início é apontado por PARÂMETROS $[i]$. Cada posição desta lista é um apontador para uma posição de um array VS, correspondendo então a uma das variáveis reais (de ligação ou de estado) conectadas à agência. Esta parte da estrutura de dados permite ao simulador instanciar uma agência genérica, que passa a fazer o papel de uma agência real do sistema.



a) exemplo de parte de uma rede de agências.



b) estrutura de dados para as informações de interconexão entre agências.



c) estrutura de dados para representar as interconexões entre agências e variáveis.

Figura 7. - Estrutura de dados para representar uma rede de agências.

Para a propagação das alterações na variáveis, ou seja, das mensagens de alteração de entorno, o simulador tem acesso a uma LISTA-DE-ENTORNO, onde para cada agência i estão relacionadas, a partir da posição apontada por ENTORNO [i], as agências que têm acesso de leitura a variáveis às quais a agência i tem acesso de escrita.

4.3 Lista de eventos

O coordenador realiza a função de avanço do tempo de simulação administrando uma lista de eventos, na qual ele efetua a inserção de uma notícia de evento para cada mensagem de tempo fornecida pelas agências. Cada notícia é um registro com 2 campos (nome da agência e tempo do evento), além de ponteiros para encadeamento da lista.

Três operações básicas são efetuadas sobre esta lista: a) avançar o tempo de simulação até o menor tempo de evento anotado dentre todas as notícias; b) identificar todas as notícias cujo tempo de evento é idêntico ao tempo atual de simulação; c) inserir novas notícias na lista.

A literatura apresenta uma série de organizações para a lista de eventos, cujo objetivo é uma otimização das 3 operações acima mencionadas. Cada organização corresponde a uma estrutura de dados para armazenamento das notícias de evento mais algoritmos para execução das operações de acesso. A quase totalidade das organizações sugeridas inserem notícias segundo a ordem crescente de seus tempos de evento, o que torna as operações a e b de complexidade $O(1)$. As organizações diferem então basicamente no tratamento dado à operação de inserção de novas notícias. Um resumo das principais organizações e uma avaliação crítica de suas eficiências pode ser encontrado em /COM 79/.

4.4 Lista de chamada e seqüência de chamada

A lista de chamada contém os nomes de todas as agências que devem ser ativadas pelo coordenador no tempo atual de simulação. Em princípio a lista deve conter os nomes das agências para as quais existe uma mensagem de tempo para o tempo atual. Além destas serão posteriormente incluídas na lista as agências atingidas por mensagens de alteração de entorno. Apenas duas operações são então efetivadas sobre esta lista: inserção de um nome de agência e busca do nome da próxima agência a ser ativada. Desta forma esta lista pode ter uma organização bastante simples.

A lista de chamada está relacionada com os conflitos de reconhecimento, introduzidos em /WAG 84a/. A figura 8 mostra o exemplo de parte de um sistema digital descrito no nível lógico como uma rede de agências. Num tempo t_i dois eventos estão programados para as agências A e B, que irão respectivamente efetivar as transições de valores VA: 1→0 e VB: 0→1. A lista de chamada conterà inicialmente então os nomes A e B. Existem duas seqüências possíveis de chamada, que fornecem diferentes saídas na agência C. Se C for ativada imediatamente após a alteração em VA, e depois novamente após a alteração em VB, a variável VC passará pela seqüência de valores 1→0→1. Se C for ativada apenas após ambas alterações em VA e VB, o valor de VC não será alterado, permanecendo em 1. Neste caso de sistemas lógicos, o conflito de reconhecimento se manifesta na forma de um pulso de hazard. A ocorrência ou não deste pulso depende da seqüência de chamada adotada pelo coordenador. Esta decisão, do ponto de vista do usuário do simulador, deve ser considerada de natureza aleatória. O usuário não pode modelar um sistema a partir de uma expectativa sobre um detalhe de implementação do simulador, tal como a estratégia de "dispatch" adotada na ativação de agências. A possibilidade de um modelo fornecer aleatoriamente resultados diferentes na simulação está ob-

viamente relacionada a uma imprecisão do modelo, o que é exatamente o caso dos conflitos de reconhecimento e dos pulsos de hazard.

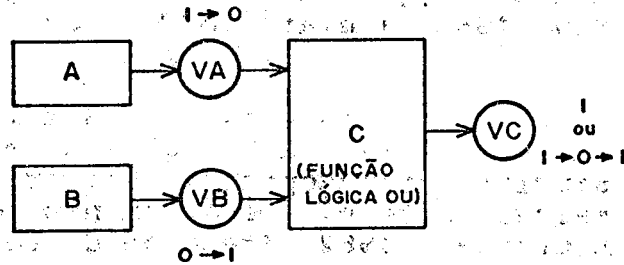


Figura 8.- Conflitos de reconhecimento

BIBLIOGRAFIA

- /COM 79/ COMFORT, J.C. "A Taxonomy and Analysis of Event Set Management Algorithms for Discrete Event Simulation", 12th Annual Simulation Symposium, Tampa, Mar. 1979, pp. 115-146
- /DAL 85/ DAL SASSO-FREITAS, C.M. e F.M. de OLIVEIRA "Editor Gráfico para Sistemas Modelados como Redes de Agências", Porto Alegre, CPGCC-UFRGS, Nov. 1985 (Relatório Técnico, nº 026).
- /POT 82/ POTT, E. "Ein Interaktiver Simulator für Instanzennetze im Vergleich mit anderen Simulationskonzepten", Universidade de Kaiserslautern, Departamento de Engenharia Elétrica, 1982 (tese de doutorado)
- /SIM 73/ BIRTWISTLE, G.M. et al. SIMULA Begin, Auerbach Publ. Inc., Philadelphia, 1973
- /WAG 83/ WAGNER, F.R. "Über die Austauschbarkeit von Universalität und Effizienz bei Instanzennetzsimulatoren, insbesondere für digitale Hardware", Universidade de Kaiserslautern, Departamento de Informática, 1983 (tese de doutorado)
- /WAG 84a/ WAGNER, F.R. "Modelamento de Processos Digitais com Redes de Instâncias". Porto Alegre, CPGCC-UFRGS, Mar. 1984 (Relatório Técnico nº 006)
- /WAG 84b/ WAGNER, F.R. "Modelamento e Simulação de Processos Digitais Usando Redes de Instâncias". In: SEMINÁRIO INTEGRADO DE SOFTWARE E HARDWARE, XI. Viçosa, MG, 21-28 julho 1984. Anais. pp 309-318
- /WAG 85/ WAGNER, F.R., C.M. DAL SASSO FREITAS e L.G. GOLEND ZINER. "O Processo de Projeto de Sistemas Digitais num Ambiente Integrado de CAD". Porto Alegre, CPGCC-UFRGS, Out. 1985 (Relatório Técnico nº 024)
- /WEN 79a/ WENDT, S. "The Programmed Action Module: An Element for System Modelling", Digital Processes, Vol. 5, 1979
- /WEN 79b/ WENDT, S. "Blockorientiertes interaktives Simulationssystem BORIS", Universität Kaiserslautern, Fachbereich Elektrotechnik, Dez. 1979 (Informationsschrift)

/WEN 83/ WENDT, S. "Grundlegende systemtechnische
Begriffe und Modelle der Informationstechnik",
Universität Kaiserslautern, Fachbereich
Elektrotechnik, Jan. 1983 (Bericht 83 B-14)

RELATORIOS TECNICOS MAIS RECENTES

- RT-007: COSTA, A. C. R.
Caracterizacao dos conhecimentos e da arquitetura de um sistema especialista em projeto logico de circuitos digitais.
Porto Alegre, CPGCC/UFRGS, jun/84.
- RT-008: FREITAS, C. M. D. S.
Programacao grafica interativa com o PGE/UFRGS.
Porto Alegre, CPGCC/UFRGS, jun/84.
- RT-009: FREITAS, C. M. D. S.
Descricao do pacote grafico PGE/UFRGS.
Porto Alegre, CPGCC/UFRGS, jul/84.
- RT-010: SAYAO, M. & TOSCANI, S. S.
Sistema multiprogramavel HP2100S - manual de referencia.
Porto Alegre, CPGCC/UFRGS, out/84.
- RT-011: SAYAO, M. & TOSCANI, S. S.
Sistema multiprogramavel HP2100S - manual de usuario.
Porto Alegre, CPGCC/UFRGS, out/84.
- RT-012: WAGNER, F. R.
Basic techniques for gate level simulation - a tutorial.
Porto Alegre, CPGCC/UFRGS, out/84.
- RT-013: WAGNER, F. R.
Hazard detection in logic simulation.
Porto Alegre, CPGCC/UFRGS, nov/84.
- RT-014: COSTA, A. C. R.
Clause machines.
Porto Alegre, CPGCC/UFRGS, jan/85.
- RT-015: COSTA, A. C. R.
Especificacao das tarefas do sistema especialista em projeto logico de circuitos digitais.
Porto Alegre, CPGCC/UFRGS, jan/85.
- RT-016: TOSCANI, L. V. & outros.
Laboratorio de Matematica Computacional - manual do usuario.
Porto Alegre, CPGCC/UFRGS, dez/84.

- RT-017: TOSCANI, L. V. & outros.
Laboratorio de Matematica Computacional
- manual de programas.
Porto Alegre, CPGCC/UFRGS, dez/84.
- RT-018: COSTA, A. C. R.
Introducao aos sistemas especialistas e
descricao informal do sistema muPROSPECTOR.
Porto Alegre, CPGCC/UFRGS, mai/85.
- RT-019: FRIEDRICH, L. F. & COSTA, A. C. R.
Descricao da implementacao do montador MC68000
escrito em Pascal Sequencial.
Porto Alegre, CPGCC/UFRGS, jun/85.
- RT-020: COSTA, A. C. R.
Processando linguagens naturais em PROLOG
- Parte 1: Formalismo gramatical basico.
Porto Alegre, CPGCC/UFRGS, jul/85.
- RT-021: WAGNER, F. R.
Algoritmos de simulacao de hardware
no nivel RT.
Porto Alegre, CPGCC/UFRGS, set/85.
- RT-022: WAGNER, F. R.
On the properties of event oriented logic
simulation according to significant
timing models.
Porto Alegre, CPGCC/UFRGS, set/85.
- RT-023: TAZZA, M.
Performance evaluation using a net theory
based model.
Porto Alegre, CPGCC/UFRGS, set/85.
- RT-024: WAGNER, F. R., FREITAS, C. M. D. S. & GOLENDZINER, L. G.
O processo de projeto de sistemas digitais num ambiente
integrado de CAD.
Porto Alegre, CPGCC/UFRGS, out/85.
- RT-025: TAZZA, M.
Sistemas-C/E como ferramenta de modelagem.
Porto Alegre, CPGCC/UFRGS, out/85.
- RT-026: FREITAS, C.M.D.S. & OLIVEIRA, F.M.
Editor grafico para sistemas modelados como redes de
agencias.
Porto Alegre, CPGCC/UFRGS, nov/85.