MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO DE PESQUISAS ESPACIAIS

AUTORIZAÇÃO PARA PUBLICAÇÃO
AUTHORIZATION FOR PUBLICATION

PALAVRAS CHAVES/KEY WORDS

AUTORES / AUTHORS

DIAGNÓSTICO DE FALHAS - SISTEMAS ESPECIALISTAS
CIRCUITOS DIGITAIS - SISTEMAS DE REGRAS

AUTORIZADA POR/AUTHORIZED BY

Dr. Marco Antonio Raupp
Diretor Geral

AUTOR RESPONSAVEL
RESPONSIBLE AUTHOR

Laércio Massaru Namikawa

DISTRIBUIÇÃO/DISTRIBUTION

- [ ] INTERNA / INTERNAL
- [x] EXTERNA / EXTERNAL
- [ ] RESTRITA / RESTRICTED

REVISADA POR / REVISED BY

CDU/UDC

681.3.019

DATA / DATE

Dezembro 1988

PUBLICAÇÃO Nº
PUBLICATION NO

INPE-4743-PRE/1417

ORIGEM
ORIGIN

DPI

PROJETO
PROJECT

ANIMA

Nº DE PAG. NO OF PAGES

11

ULTIMA PAG. LAST PAGE

10

VERSÃO VERSION

Nº DE MAPAS NO OF MAPS

TÍTULO/TITLE

UM SISTEMA ESPECIALISTA PARA DIAGNÓSTICO DE FALHAS EM CIRCUITOS DIGITAIS

AUTORES/AUTHORSHIP

Laércio Massaru Namikawa
Flávio Roberto Dias Velasco

RESUMO - NOTAS / ABSTRACT - NOTES

Este trabalho apresenta o sistema SEDIG para diagnóstico de falhas em circuitos digitais. SEDIG, escrito na linguagem OPS5, é um sistema baseado em regras que separa o conhecimento acerca do circuito do conhecimento em diagnóstico, tornando-o aplicável a uma grande gama de circuitos digitais. O trabalho apresenta a estrutura do sistema e sua aplicação para uma placa de visualização de imagens.

OBSERVAÇÕES / REMARKS

Trabalho apresentado no V Simpósio Brasileiro de Inteligência Artificial, realizado em Natal, RN, de 07 a 11 de novembro de 1988.

UM SISTEMA ESPECIALISTA PARA DIAGNÓSTICO DE FALHAS EM CIRCUITOS DIGITAIS*

Laércio Massaru Namikawa
Flávio Roberto Dias Velasco

Ministério da Ciência e Tecnologia
Instituto de Pesquisas Espaciais
Departamento de Processamento de Imagens
Caixa Postal 515 - 12201 - São José dos Campos - SP

RESUMO

Este trabalho apresenta o sistema SEDIG para diagnóstico de falhas em circuitos digitais. SEDIG, escrito na linguagem OPS5, é um sistema baseado em regras que separa o conhecimento acerca do circuito do conhecimento em diagnóstico, tornando-o aplicável a uma grande gama de circuitos digitais. O trabalho apresenta a estrutura do sistema e sua aplicação para uma placa de visualização de imagens.

1. Introdução

Embora sejam relativamente recentes, sistemas especialistas vêm sendo cada vez mais empregados em tarefas de engenharia (Hong, 1986). Uma das aplicações possíveis é o do diagnóstico de falhas em sistemas em geral (Talukdar et al., 1986; Larner, 1985) e, em especial, a circuitos digitais (Pipitone, 1986; Prevost e Laffey, 1985; Mitchell et al. 1985).

Circuitos digitais são circuitos eletrônicos que se baseiam em apenas dois níveis de sinal elétrico, os níveis lógicos 0 e 1. A implementação de um circuito digital é feita a partir de circuitos combinacionais elementares e de elementos de memória. Um defeito é um desvio qualquer na especificação de um componente. Em um circuito combinacional elementar, o defeito é caracterizado por níveis lógicos nas saídas diferentes dos preestabelecidos para um determinado conjunto de níveis nas entradas. Em um elemento de memória, os níveis lógicos na saída não obedecerão aos níveis previamente armazenados e ao conjunto de níveis em suas entradas.

Uma falha é um desvio no comportamento esperado de um sistema. Um circuito digital é constituído de circuitos combinacionais e elementos de memória interligados e o defeito em um destes componentes faz com que haja um fluxo de níveis lógicos incorretos no circuito, alterando o comportamento esperado. A extensão da falha dependerá da importância do elemento defeituoso no sistema: desde uma falha que não será

* Trabalho financiado pela SID-Informática, projeto ESTRA

percebida exceto para um único conjunto de dados de entrada até uma parada total do sistema.

A identificação do elemento responsável pela falha não é uma tarefa trivial, pois depende de uma série de fatores. Os principais são:

- uma determinada falha pode ser originária de blocos diferentes ao mesmo tempo;
- o mesmo tipo de falha pode ter origem em blocos diferentes;
- o defeito em um bloco afeta o funcionamento de outros blocos.

Uma lista de relações entre falha e elemento com defeito pode ser empregada para definir um conjunto de possíveis elementos com problemas. Para determinar qual destes elementos é o defeituoso, é necessário verificar o funcionamento de cada um deles. Sendo detetado um elemento com funcionamento duvidoso, não se pode classificá-lo como com defeito. Deve-se suspeitar também de todos os componentes que estão ligados a ele.

Para o diagnóstico, o especialista vale-se de sua experiência anterior e do raciocínio sobre as funções de cada bloco aplicadas à topologia do sistema. Um dado importante proveniente da experiência passada é a definição de qual bloco está mais sujeito a falhas do que outros, permitindo economia de tempo e esforço por possibilitar a sua verificação antes dos outros blocos.

Este trabalho mostra um sistema especialista para diagnóstico de falhas em circuitos digitais. No sistema, as informações específicas sobre o sistema (topologia do circuito, forma esperada dos sinais) são mantidas separadas do conhecimento especializado em diagnóstico. Desta forma, pode-se configurar o sistema para uma grande classe de circuitos. Na aplicação particular desenvolvida, foi usada a placa de visualização de imagens UVI (Engespaço, 1988).

O sistema foi desenvolvido em um microcomputador na linguagem OPS5 (Forgy, 81) usando o interpretador do Departamento de Processamento de Imagens (Velasco, 1987). OPS5 é uma linguagem baseada no modelo de regras que tem sido usada na construção de vários sistemas especialistas. OPS5 dispõe dos mecanismos gerais de controle e representação necessários para a engenharia do conhecimento, mas não é orientado para estratégias e esquemas de representação particulares.

Um programa em OPS5 consiste de duas partes: uma secção de declarações e outra de regras (produções). A secção de declarações define classes de dados e funções externas que podem ser chamadas. Durante a execução, os dados que o programa manipula ficam na "memória de trabalho". A memória de trabalho e a memória de regras

constituem a base de conhecimento utilizada para resolver o problema. Mais informações sobre a linguagem podem ser encontradas em Brownston et al. (1985).

2. O sistema SEDIG

Um sistema especialista no diagnóstico de falhas em circuitos digitais deve englobar os dados sobre a topologia dos blocos, os testes específicos para cada bloco, a lista de relações entre os blocos suspeitos de serem os defeituosos para cada falha e as estratégias utilizadas pelo especialista.

2.1. A memória de trabalho

Na memória de trabalho do sistema estão a maioria dos dados relativos a um determinado circuito digital. Cada um destes dados é um elemento atributo-valor e são divididos nas classes BLOCO, SINAL, LISTA e TESTE.

Os elementos atributo-valor das classes BLOCO e SINAL englobam todos os conhecimentos que podem ser extraídos do diagrama esquemático de um circuito digital, sem preocupações com a funcionalidade dos componentes. A classe BLOCO tem o conjunto de dados de identificação do bloco, a descrição de sua situação e o número de entradas ruins. A estrutura da classe SINAL contém a identificação de um sinal, a descrição de sua situação, o bloco gerador e os receptores deste sinal, um receptor a ser analisado e um índice auxiliar para a extração deste receptor do atributo vetor receptores. As listas de relação entre falha e bloco ou blocos defeituosos estão implementadas através dos elementos da classe LISTA, a qual tem o nome da falha e os nomes dos blocos suspeitos para esta falha. Estes elementos de memória, que definem o circuito são construídos através o comando **make**. A Figura 2.1 mostra exemplos de cada uma destas classes para o caso da placa de visualização de memória.

```
(make SINAL ^nome_sinal A(16-17) ^bloco_gerador X23
                    ^receptores DEC_MEM CONT_MEM fim)
(make BLOCO ^nome_bloco X23 ^estado_bloco bom)
(make LISTA ^lista_falha FB ^blocos_lista GER_VDEO
                                CONT_DAT MUX_WR fim)
```

Figura 2.1. Exemplos de elementos de especificação do circuito.

Os elementos atributo-vetor de ^blocos_lista são implementados de forma que o bloco mais propício a aquela falha seja o primeiro elemento e assim sucessivamente até o último.

Os testes específicos para cada bloco são chamados como funções externas e são referenciados através dos elementos da classe TESTE que armazena o nome do bloco

para o qual o teste é executado e a situação do teste. A chamada a função externa é feita através de **call testa_bloco (bloco)**, onde testa_bloco é o nome da função externa e (bloco) é a variável passada à função externa especificando qual bloco deve ser testado. A função externa testa_bloco lê o nome do bloco a ser testado e dispara o teste específico que executa a seguinte sequência:

- Verifica os sinais de entrada do bloco.
- Se um ou mais deles estiverem ruins, retorna ao sistema os nomes dos sinais ruins.
- Se todas as entradas estiverem boas, então verifica as saídas e devolve os nomes das saídas ruins.
- Caso não hajam nem entradas ruins, nem saídas ruins, então o teste responde que o bloco está bom.

Para a verificação dos sinais do bloco, o teste específico deve fornecer os seguintes dados ao operador do sistema:

- As condições necessárias para obter o conjunto de dados de teste nas entradas do bloco.
- Qual instrumento de medida deve ser utilizado.
- Os procedimentos para o ajuste do instrumento de medida.
- A localização física do ponto onde deve ser colocada a ponta de prova do instrumento.
- O nível ou forma de onda do sinal a ser testado.

O retorno de testa_bloco devolve ao sistema os dados a respeito dos sinais do bloco testado através de três classes: ENTRADAS_RUINS, SAIDAS_RUINS e RESPOSTA, que indicam quais sinais de entrada ou saída são ruins.

Na memória de trabalha do sistema especialista existem outros elementos que auxiliam no processo de diagnóstico. Estes elementos são da classe FALHA e são utilizados para armazenar os dados relativos à falha selecionada pelo usuário.

2.2. A memória de regras

Na memória de regras do SEDIG estão implementadas as funções que compõe a sessão de diagnóstico. O fluxo de controle é ativado pela criação e modificação dos elementos de memória. Assim, ao ser iniciada a sessão de diagnóstico, o primeiro dado que deve ser ativado refere-se à falha apresentada pelo circuito em diagnóstico. Será criado desta maneira um elemento de memória que disparará as regras que casam a este dado. A Figura 2.2 mostra a regra que lê o nome da falha selecionada pelo usuário. Note o uso da função externa "menu" que provê a interface com o usuário.

```
(p le-falha-selecionada
    {(FALHA ^falha_selecionada nil) (falha) }
--)
   (bind (selecao) (menu "falhas.mnu" "TIPO" "OPCOES"))
   (modify (falha) ^falha_selecionada (selecao))
)
```

Figura 2.2. Regra de leitura de falha.

Novos dados serão criados ou modificados até que não haja mais regras a serem disparadas, significando que a sessão de diagnóstico está encerrada, tenha sido com sucesso ou não.

As regras iniciais, a partir da falha selecionada, atribuem ao primeiro bloco dos suspeitos para a falha o estado suspeito e verifica se o estado é bom ou se o bloco é do tipo não testável, sempre considerado bom. Caso seja, remove o bloco de suspeito pela falha e atribui este estado ao próximo bloco dos suspeitos. Uma regra é necessária se nenhum destes elementos for ruim, verificado através do valor **fim** na lista de blocos para a falha escolhida. Esta regra chama outra vez a função externa que apresenta as falhas, requisitando ao usuário o nome de outra falha ou se deseja encerrar o diagnóstico. Para o caso de seleção de fim do processo, é apresentada uma tela indicando o fim do diagnóstico.

A partir da atribuição do estado suspeito a um bloco, serão disparadas as regras que executam testes sobre o funcionamento do bloco. Estes testes criam elementos a partir dos quais é possível determinar o bloco defeituoso.

Os conjuntos de regras que executam as tarefas de determinação do elemento defeituoso baseiam-se nos estados dos blocos de modo que pode-se dividí-los nos seguintes grupos:

- Bloco suspeito.
- Bloco ruim.
- Bloco trocado.
- Bloco saída_ruim.
- Bloco entrada_ruim.
- Bloco entrada_suspeita.

Para um bloco suspeito, inicialmente será ativada a regra que executa o teste do bloco, criando elementos que determinarão a situação do bloco. Estes elementos podem ser das classes ENTRADAS_RUINS, SAIDAS_RUINS E RESPOSTA. O retorno de cada uma destas classes tem um significado diferente e as regras que casam a cada situação modificam o estado do bloco para ruim, bom ou saída_ruim. A Figura 2.3 mostra uma regra típica para o grupo "bloco suspeito". O processamento para os outros grupos é semelhante.

```
(p   marca-bloco-ruim
   { (SAIDAS_RUINS ^saida_bloco (bloco)) (saidas)}
   { (BLOCO ^nome_bloco (bloco)
            ^estado_bloco suspeito) (bloco_ruim) }
--)
   (remove (saidas))
   (modify (bloco_ruim) ^estado_bloco ruim)
)
```

Figura 2.3. Regra para análise de bloco suspeito.

As regras dos diversos grupos mencionados englobam todo o raciocínio para a determinação do bloco defeituoso. Existem outras que retornam as condições iniciais dos elementos da classe SINAL e que modificam os estados do blocos para **nil** caso não sejam bons.

3. Visão externa do sistema especialista

A interface com usuário é muito importante no sistema especialista porque o processo de diagnóstico é guiado pelos dados criados com as entradas do usuário. A entrada destes dados deve ser efetuada de maneira correta e para isto o sistema apresenta opções de modo a não permitir que hajam dúvidas, assim como fornece as informações necessárias para verificar um sinal.

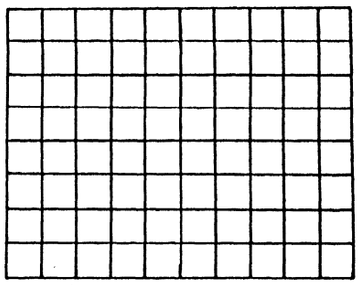
A entrada dos dados é feita através de telas de menu na qual o usuário faz a sua opção. Existem duas telas de menu para entrada de dados, sendo a primeira para a seleção da falha e a outra para a entrada da informação sobre o estado de um sinal.

A tela de menu de falha é dividida em duas, sendo que a primeira faz a seleção do tipo de falha apresentada utilizando apenas as teclas de cursor e de (enter) do teclado. A segunda tela de menu apresenta as opções de falha do tipo selecionado, utilizando também as teclas de cursor e de (enter), e permite ainda que se retorne ao menu anterior.

Para o teste de um bloco a seguinte tela é apresentada:

TESTE DO BLOCO "nome do bloco"

FORMA DE ONDA DO SINAL "nome do sinal"



ESCALA NO EIXO X
"escala x"

COMPARE O SINAL
E ENTRE COM
(B)OM OU (R)UIM
SINAL : _

ESCALA NO EIXO Y
"escala y"

"linhas de texto"

Figura 3.1. Tela de teste de bloco.

No centro da Figura 3.1. está a forma de onda do sinal que serve de referência ao usuário; as informações sobre as escalas nos eixos X e Y estão à direita da forma de onda; abaixo são colocadas até nove linhas de texto com informações sobre os procedimentos de ajuste dos equipamentos, a localização dos pontos onde devem ser colocadas as pontas de prova e outros dados adicionais. O campo que permite a entrada da resposta do usuário está à esquerda, e a seleção é efetuada através da entrada da letra "b" para bom ou "r" para ruim.

Existem telas também para a saída de informações durante o diagnóstico. Estas telas são duas, uma indicando em qual bloco deve ser aplicada a terapia por ter o estado ruim e a outra indicando ao usuário o fim da sessão de diagnóstico.

O diagnóstico de um dos tipos de defeitos mais comuns em circuitos digitais é apresentado sob a ótica do usuário, demonstrando o uso das telas descritas acima. Este tipo é o defeito em uma saída de um bloco suspeito pela falha e para este tipo, a seguinte sequência de telas é apresentada:

- Tela para seleção do tipo de falha.
- Tela para seleção da falha apresentada.
- Tela de teste da primeira entrada do bloco suspeito com a forma de onda deste sinal.
- Tela de teste da última entrada do suspeito com a forma de onda de entrada deste sinal.

Como não há nenhuma entrada ruim para este tipo de defeito,as seguintes telas são apresentadas:

- Tela de teste da primeira saída do bloco suspeito com a forma de onda deste sinal.

- Tela de teste da última saída do bloco suspeito com a forma de onda deste sinal.

Como o defeito está na saída, um destes sinais será ruim e a tela que indica que a terapia no bloco é necessária é apresentada.

O sistema apresentará novamente as telas com os sinais do bloco para que o usuário verifique se o bloco não apresenta mais defeitos. Se não há mais problemas, o sistema apresenta a tela de seleção da falha para o usuário selecionar FINALIZA. O sistema apresenta então a tela que indica o fim da sessão de diagnóstico.

Para os outros casos com tipos de falhas diferentes, as telas a serem apresentadas são as mesmas mudando apenas os nomes dos blocos a serem testados e a serem submetidos à terapia.

4. Conclusão

Embora muitos sistemas de diagnóstico, dos quais o sistema MYCIN (Barr e Feigenbaum, 1982) é o exemplo mais famoso, utilizem encadeamento retroativo de regras, no sistema SEDIG o encadeamento é estritamente progressivo. (É possível simular, em OPS5, o encadeamento regressivo.) Isto é devido a peculiaridades da tarefa de diagnóstico de circuitos digitais, que é substancialmente mais simples que o diagnóstico médico. A partir de uma lista inicial de falhas e blocos suspeitos, o sistema segue pelo circuito testando sinais até a deteção do bloco com defeito. Não há, também, necessidade de tratamento de incerteza.

Uma filosofia seguida no desenvolvimento do sistema SEDIG foi manter as regras o mais próximo possível do raciocínio do especialista humano. Isto auxiliou no desenvolvimento do sistema, quando foi possível criar novas regras de maneira simples.

Os dados relativos à lista de relação entre falha e blocos suspeitos pela falha não estão completos, uma vez que a quantidade de falhas que podem ocorrer é grande e a diferença entre uma e outra pode ser imperceptível. Para o sistema não é necessário que haja uma falha para cada elemento, uma vez que o sistema tem capacidade de testar os blocos geradores e receptores dos sinais de um bloco suspeito. É importante apenas que o sistema tenha definido pelo menos uma falha para cada via de fluxo de sinal.

Os sinais do tipo "tri-state" e "wired-or" são de difícil tratamento por poderem ter vários blocos geradores. Um sinal "tri-state" é tratado pelo sistema dividido em vários sinais, sendo cada um gerado por um dos blocos geradores. Este tratamento é possível uma vez que apenas um dos blocos está gerando o sinal ao

mesmo tempo. Quanto ao sinal do tipo "wired-or" um tratamento possível seria considerar o nó como sendo um elemento do tipo porta lógica OU.

Várias extensões podem melhorar o sistema especialista. Algumas serão discutidas a seguir:

- A terapia em um bloco que é composto de vários componentes elementares pode ser feita através da ativação de elementos de memória relativos a este bloco, que serão submetidos às regras existentes, uma vez que o conjunto das regras é válido para qualquer bloco. Esta tarefa pode ser implementada de modo que apenas os elementos de memória necessários estejam ativos em um instante do diagnóstico, tendo os outros armazenados, por exemplo, em disco e feita a carga, quando ativados, por meio de uma função externa.

- Para o teste de um bloco é necessário comparar um determinado sinal ao dado apresentado pelo sistema. A indicação da localização física do ponto na placa onde deve ser colocada a ponta de prova do instrumento pode ser efetuada por meio de uma figura. Isto permite ao usuário uma rápida visão, tornando desnecessário o uso de qualquer diagrama do circuito.

- A indicação dos blocos considerados suspeitos e dos que realmente estavam ruins, além dos sinais que estavam ruins podem ser apresentados ao final da sessão de diagnóstico, permitindo ao usuário avaliar o estado geral em que se encontrava o circuito, assim como a eficiência do sistema especialista.

REFERÊNCIAS

BARR, A., e FEIGENBAUM, E. A., editores, "The Handbook of Artificial Intelligence", 3 volumes, HeurisTech Press, Stanford, 1982.

BROWNSTON, L.; FARREL, R.; KANT, E.; MARTIN., N., "Programming expert systems in OPS5", Addison- Wesley, Reading, Massachusetts, 1985.

ENGESPAÇO, "Manual do SITIM-150 - Unidade de visualização de imagens", São José dos Campos,SP, 1988. Manual técnico.

HONG, S.J., "Expert systems - Guest editor's introduction", Computer, vol. 19(7):12-15, julho 1986.

LARNER, D.A.; "A recursive expert troubleshooting system utilizing general and specific knowledge", The Second Conference on Artificial Intelligence Applications, pp. 34-41, 1985.

MITCHELL, T.M.; STEINBERG, L.; SMITH, R.G.; SCHOOLEY, P.; JACOBS, H.; KELLY, V.; "Representations for reasoning about digital circuits", Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp. 343-344, 1981.

PIPITONE,F.; "The FIS Electronics Troubleshooting System", Computer, vol. 19(7):68-76, julho 1986.

PREVOST, M.P.; LAFFEY, T.J.; "Knowledge-based diagnosis of electronic instrumentation", The Second Conference on Artificial Intelligence Applications, pp. 42-48, 1985.

TALUKDAR, S.N.; CARDOSO, E.; LEÃO, L.V.; "Toast: The Power System Operator's Assistant", Computer, vol. 19(7):53-60, julho 1986.

VELASCO, F.R.D.; "Uma implementação da linguagem OPS5 para microcomputadores compatíveis com o IBM-PC", Anais do 4o. Simpósio Brasileiro de Inteligência Artificial, Uberlândia, outubro 1987.