

REDES DE PETRI E VHDL NA PROTOTIPAGEM RÁPIDA DE SISTEMAS DIGITAIS

Ricardo J. Machado, João M. Fernandes e Alberto J. Proença

Sumário

O objectivo principal deste artigo é exemplificar a utilização de uma metodologia de especificação de sistemas digitais, baseada em Redes de Petri orientadas por objectos, para obter de uma forma rápida e simplificada um protótipo em VHDL do sistema pretendido. É considerado como exemplo um sistema digital, para o qual se efectua a especificação no modelo RdP-shobi e a partir da qual se pode gerar automaticamente código VHDL. Este exemplo permite demonstrar acerca da capacidade desta metodologia no projecto de sistemas digitais, suportado por princípios de orientação por objectos e por uma ferramenta de EDA concebida para o efeito.

1 Introdução

Nos sistemas digitais de maior complexidade e dimensão podem identificar-se dois componentes distintos do sistema global: o controlador e o sistema controlado. No que respeita à modelação do sistema global, as poucas soluções existentes actualmente recorrem a técnicas em que ambas as partes são especificadas quase independentemente, não conseguindo integrar num único formalismo a representação do sistema global.

Foi desenvolvido o modelo RdP-shobi, baseado em Redes de Petri (RdP), que possibilita a utilização de hierarquia nos sistemas globais e o recurso a objectos para modelar o sistema controlado. Desta maneira, torna-se possível especificar o sistema digital global de uma forma estruturada e incremental. Basicamente, o modelo RdP-shobi substitui as tradicionais marcas por objectos. A invocação de métodos corresponde à leitura de sinais de entrada e à geração de sinais de controlo, que permite, simultaneamente, a modelação do comportamento do sistema controlado. A modelação de alguns controladores paralelos, bem como dos respectivos sistemas controlados, permitiu verificar que o modelo desenvolvido adequa-se à especificação de sistemas digitais de uma forma hierárquica, modular e incremental.

Foi definida a arquitectura de uma aplicação computacional, ferramenta SOFHIA, que suporta directamente o modelo proposto para as RdP e que possibilita uma especificação gráfica do sistema. Como solução para implementação da arquitectura definida, utiliza-se o ambiente SCBA [1], o que exigiu o mapeamento do RdP-shobi num modelo de computação baseado em agentes, o MCBA [2]. Esta aplicação usa o ambiente CONPAR [3], sendo possível obter uma descrição em VHDL da unidade de controlo do sistema especificado.

Abstract

This article aims to show the use of an object-oriented Petri-Net-based methodology to specify digital systems, to easily and rapidly obtain a VHDL prototype of the system. As a case study, an example is considered and its specification with the shobi-PN model is presented, from which VHDL code can be automatically generated. With this example it is possible to demonstrate the capability of this methodology in the project of digital systems, supported by object-oriented programming principles and by an EDA tool conceived for these purposes.

2 O Modelo RdP-shobi

VHDL é uma linguagem standard IEEE para descrição de hardware que explora a concorrência na especificação de sistemas digitais, permitindo a sua simulação e síntese. Por outro lado, as RdP são apropriadas para modelar e analisar formalmente sistemas discretos de grande complexidade [4]. A abordagem, seguida neste trabalho, para implementar, em VHDL, controladores paralelos e síncronos, a partir das suas especificações baseadas em RdP de alto-nível, recorre à única metodologia que modela eficientemente RdP em VHDL [5].

O modelo de especificação foi desenvolvido para suportar a utilização de hierarquia generalizada, bem como para possibilitar a modelação conjunta, quer da parte de controlo quer do sistema controlado. Este modelo, designado de RdP-shobi, incorpora todos os conceitos de sincronismo e interpretação básica do modelo RdP-SI (RdP Síncronas e Interpretadas) [3], juntamente com novos conceitos de hierarquia no sistema global e com conceitos da modelação orientada por objectos. Uma das consequências deste modelo reside no suporte directo a estruturas hierárquicas na unidade de controlo e no sistema controlado.

Algumas das características básicas deste novo modelo revelam a fusão numa única extensão de RdP de conceitos primeiramente definidos no âmbito das RdP Síncronas, RdP Hierárquicas, RdP Coloridas e RdP Orientadas por Objectos [6, 7]. Em RdP-shobi, as marcas, consideradas coloridas, representam objectos que modelam estruturas do sistema controlado. As variáveis de instância representam a informação que flui e é transformada no sistema controlado e os métodos dos objectos são a interface entre a unidade de controlo e o sistema controlado.

Os nodos (transições e lugares) invocam métodos às marcas (objectos), quando estas lá chegam. Contudo, só são invocados aqueles métodos que tem uma relação directa com os sinais de controlo de *hardware*, uma vez que existem métodos adicionais na interface dos objectos que não são utilizados pela RdP. Estes métodos são invocados pelo *software* de simulação para ler e alterar o conteúdo da cada estrutura do sistema controlado.

Cada arco possui uma ou mais cores que o associa aos tipos de objectos que por lá podem passar. Isto implica que passem a existir caminhos bem definidos na RdP-shobi, associados aos percursos que cada estrutura do sistema controlado (marca/objecto) efectua ao longo da mesma. Esta imposição simplifica a leitura estrutural da RdP, além de limitar a capacidade de alguns lugares, pois os objectos que não são invocados num dado caminho da rede não o necessitam de percorrer.

Podem utilizar-se macronodos e macromarcas para implementar a hierarquia no modelo. Os macronodos representam sub-redes e as macromarcas representam hierarquias de submarcas, podendo-se encapsular macro-actividades num macronodo, assim como macro-estruturas do sistema controlado em macromarcas.

Seguem-se alguns conceitos para a compreensão do exemplo apresentado na secção 4: (1) rede de controlo: conjunto de nodos e arcos contíguos (esqueleto) da RdP-shobi que corresponde estruturalmente à RdP-SI, sem reiniciações, para especificar a unidade de controlo; (2) caminho de controlo: percurso definido por uma marca na rede de controlo; (3) nodos de controlo: nodos da rede de controlo; (4) arcos de controlo: arcos da rede de controlo; (5) caminho de fecho: percurso definido por uma marca fora da rede de controlo (podem surgir situações em que este caminho não necessite de existir); (6) nodos de fecho: nodos pertencentes a um caminho de fecho; (7) arcos de fecho: arcos pertencentes a um caminho de fecho; (8) ciclo de fecho: percurso definido por uma marca ao longo da RdP-shobi, que é composto pelo caminho de controlo e pelo caminho de fecho (caso este exista) e pode ser identificado através do rasto da cor da marca associada a todos os arcos do ciclo; (9) rede global: RdP-shobi que especifica o sistema digital global, é composta pela rede de controlo, pelos caminhos de fecho e pelas marcas; (10) rede associada: RdP-SI estruturalmente correspondente à rede de controlo da rede global após as reiniciações para marcas incolores.

A consulta de [8] poderá complementar algumas questões não abordadas, neste artigo sobre o modelo RdP-shobi.

3 A Ferramenta SOFHIA

A utilização do modelo teórico RdP-shobi para especificar sistemas digitais, depende fortemente da existência de uma ferramenta de CAD que

auxilie em muitos dos passos necessários para obter o código final para a sua síntese.

A ferramenta SOFHIA (*Software for Hierarchical Architectures*) (fig. 1) é adequada para especificar sistemas digitais usando o modelo RdP-shobi [9]. A partir de especificações hierárquicas em RdP-shobi, é gerado automaticamente código VHDL, permitindo usar, em fases posteriores do projecto, ferramentas de CAD que aceitem VHDL como entrada.

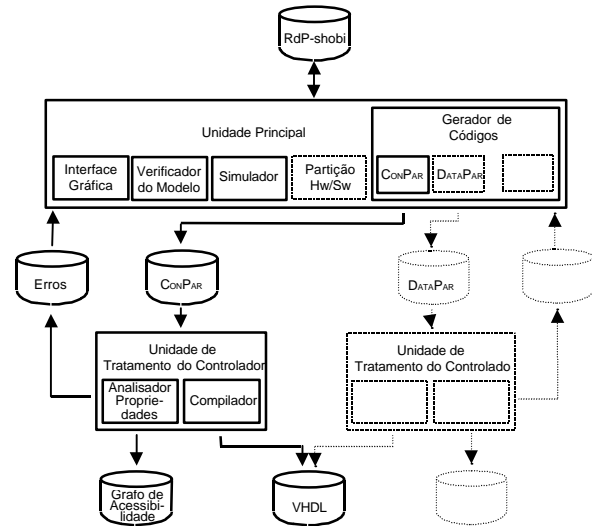


Fig. 1: O Fluxo de Projecto SOFHIA.

Quando um sistema é constituído principalmente por componentes já modelados e reutilizáveis, a atenção do engenheiro de projecto pode centrar-se essencialmente no desenvolvimento das partes novas do projecto. Desta forma, os sistemas podem ser desenvolvidos mais rápida e facilmente. Esta abordagem conduz a uma nova disciplina conhecida como Prototipagem Rápida, muito vantajosa em situações em que os objectivos e requisitos do sistema não são completamente conhecidos na fase inicial do projecto. A ferramenta proposta suporta todas as tarefas para a prototipagem rápida de sistemas digitais, a partir de especificações em RdP, incluindo: (1) as verificações formais de propriedades e consistência do modelo; (2) a possibilidade de simulação do funcionamento do sistema global; (3) a obtenção de código VHDL para síntese final do controlador.

4 Exemplo

Para verificar a adequabilidade do modelo proposto escolheram-se alguns casos práticos de áreas suficientemente diversificadas (interfaces de comunicações, microarquitECTURA de computadores e controlo industrial [10]). Seleccionaram-se exemplos perfeitamente caracterizados e já analisados por outros autores, para validar inquestionavelmente o modelo RdP-shobi. Neste artigo é apresentado o controlador do *Transputer*

Link Adaptor (TLA), circuito integrado IMS-C011 da INMOS [11], que efectua a conversão bidireccional série-paralelo. Este circuito pode ser utilizado para interligar Transputers, controladores de periféricos INMOS, subsistemas de I/O e microprocessadores de diferentes famílias.

Para construir a RdP-shobi, deve modelar-se o sistema controlado associado ao TLA, através da identificação dos objectos existentes e definindo as suas variáveis e métodos de instância.

Analisando pormenorizadamente o sistema é possível identificar três grupos distintos de objectos no sistema controlado: as ligações série (*links*), os barramentos de dados (*buses*) e o registo de deslocamento (*shift register*). De acordo com esta selecção, declararam-se e codificam-se as classes correspondentes aos objectos identificados.

Depois da modelação do sistema controlado, deve especificar-se o controlador, através do desenho da RdP-shobi, utilizando instâncias das classes descritas anteriormente, que definem ciclos de fecho ao longo da rede e onde cada instância é invocada com métodos existentes na sua interface.

É necessário dispor de: (1) duas instâncias da classe *link* (Lin e Lout) para representar os objectos ligação série de entrada e ligação série de saída; (2) duas instâncias da classe *bus* (Bin e Bout) para representar os objectos barramentos de entrada e de saída; (3) uma instância da classe *shift_register* (Reg) para representar o objecto registo de deslocamento. As classes são escritas numa linguagem orientada por objectos.

A RdP-shobi (fig. 2) para especificar o TLA consiste numa máquina de *Mealy* com 30 lugares, 35 transições e 2 macrolugares, um para a conversão paralelo-série (*ParSer*) e outro para a conversão série-paralelo (*SerPar*).

Esta RdP pode ser dividida em quatro partes funcionalmente distintas, correspondendo três delas a ciclos de fecho completos: (1) o troço de rede ($p1, t1, p2, t2, SerPar, t3$) que executa a detecção do pacote de dados na porta série de entrada e efectua a conversão série-paralelo, (2) o ciclo de fecho de Bout ($p29, t3, p13, t4, p14, t5, p15, t6, p16, t7$) que gera o sinal de saída QValid e espera pelo sinal de *acknowledgement* QAck, de modo a completar a conversão série-paralelo; (3) o ciclo de fecho de Bin ($p12, t8, ParSer, t9, p28, t10, p11, t11$) que executa a conversão paralelo-série, bem como a sinalização para início de transmissão onde estão envolvidos os sinais *!Ack* e *!valid*; (4) o ciclo de fecho de Lout ($p17, t8, ParSer, t9, t5, p15, t6$) que gere o acesso à porta série de saída. O único troço de rede que, dos quatro descritos, não define um ciclo de fecho completo, constitui parte do ciclo de fecho de Lin, uma vez que a outra parte do ciclo ($p1, t1, p2, t10$) possibilita a comutação para o modo de conversão paralelo-série. O objecto Reg é o único que possui um ciclo de fecho ($pf1, t2, SerPar, t3$) que é composto pelo caminho de controlo e pelo caminho de fecho, uma vez que, para todos os outros

objectos presentes na RdP-shobi, os ciclo de fecho são somente constituídos pelos respectivos caminhos de controlo.

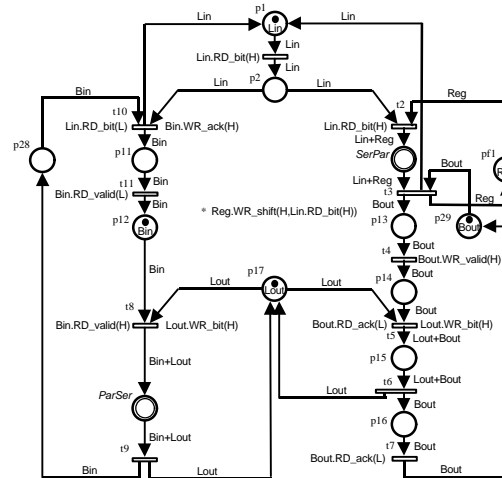


Fig. 2: RdP-shobi do TLA.

A RdP-shobi não possui declarações de nodos de reiniciação, pelo que a rede de controlo é reiniciada com a marcação inicial de alguns dos lugares de controlo ($p1, p12, p17, p29$). Para obter a RdP-SI (estrutura da rede associada) do TLA, a partir da respectiva RdP-shobi, remove-se o único lugar de fecho existente ($pf1$) e os arcos que o ligam às transições de controlo $t2$ e $t3$. O lugar $pf1$ é utilizado para simulação do sistema controlado, não possuindo informações relativas à unidade de controlo. Para interpretar a rede associada, devem retirar-se as referências a cores e transformar as invocações dos métodos em sinais de *hardware*.

Um ficheiro, com código VHDL ao nível fluxo de dados, contendo, alternativamente, um comando Block ou um comando Process, é gerado automaticamente pelo módulo Compilador.

5 Conclusões

Este artigo mostra que o modelo RdP-shobi é útil e eficiente na especificação de sistemas digitais. Este modelo é o único conhecido que utiliza RdP orientadas por objectos para especificar a unidade de controlo paralela e o sistema controlado de uma forma integrada e modular. O modelo RdP-shobi apresenta um comportamento síncrono, técnicas de modelação orientadas por objectos e mecanismos hierárquicos. Esta nova metodologia suporta directamente estruturas hierárquicas na unidade de controlo e no sistema controlado.

A análise de alguns exemplos de aplicação, demonstra que existe uma relação entre a estrutura das RdP e o tipo de abordagem seguida na especificação do sistema considerado. No caso do TLA, a RdP obtida reflecte uma abordagem *data-driven*, porque só existe um caminho de fecho. Este tipo de RdP incorpora na rede de controlo as reiniciações dos objectos, sem a

necessidade de incluir caminhos de fecho. As redes de controlo *data-driven* podem ser totalmente fechadas, ou seja, não possuem nodos de reiniciação; as redes de controlo *control-driven* têm de ser abertas. Esta análise mostra que as RdP-shobi seguem directamente uma abordagem do tipo *data-driven* na especificação de sistemas digitais paralelos e suporta adequadamente a especificação, da unidade de controlo e do sistema controlado, para a prototipagem rápida de sistemas.

6 Referências

- [1] Pina A. *SCBA - Simulação Concorrente Baseada em Agentes*. XX SEMISH, Florianópolis, Brasil, 1993.
- [2] Pina A. *MCBA - Modelo de Computação Baseado em Agentes*. OOP'94, Lisboa, 1994.
- [3] Fernandes J.M., Proença A.J. *Redes de Petri na Especificação e Validação de Controladores Paralelos*. 1º Enc. Nac. do Colégio de Eng^a Electrotécnica, pp. 113-8, OE, Lisboa, Maio 1994.
- [4] Murata T. *Petri Nets: Properties, Analysis and Applications*. Proc. of the IEEE, 77(4):541-80, Abril 1989.
- [5] Pardey J., Bolton M. *Logic Synthesis of Synchronous Parallel Controllers*. IEEE Int. Conf. on Computer Design, pp. 454-7, 1991.
- [6] Lakos C. *The Object Orientation in Object Petri Nets*. 1st Workshop on Object-Oriented Programming and Models of Concurrency, Turim, Itália, Junho 1995.
- [7] Jensen K. *An Introduction to the Theoretical Aspects of Coloured Petri Nets*. Rel. Técnico, Comp. Sc. Dept, Aarhus Univ., Agosto 1994.
- [8] Machado R.J. *Hierarquia em Redes de Petri Orientadas por Objectos na Especificação de Sistemas Digitais*. Tese de Mestrado, Dep. Informática, U. Minho, Novembro 1996.
- [9] Machado R.J., Fernandes J.M., Proença A.J. *SOFHIA: A CAD Environment to Design Digital Control Systems*. Hardware Description Languages and their Applications: Specification, modelling, verification, and synthesis of microelectronics systems. Ed. C.D. Kloos e C. Cerny, Chapman & Hall, Abril 1997.
- [10] Machado R.J., Fernandes J.M., Proença A.J. *Specification of Industrial Digital Controllers with Object-Oriented Petri Nets*. IEEE Int. Symp. on Industrial Electronics - ISIE'97, Guimarães, Portugal, Julho 1997.
- [11] Taylor R. *Transputer Communication Link*. Microprocessors and Microsystems, 10(4):211-5, 1986.
- [12] Fernandes J.M., Adamski M., Proença A.J. *VHDL Generation from Hierarchical Petri Net Specifications of Parallel Controller*. IEE Proc.: Computers and Digital Techniques, 144(2):127-37, Março 1997.

Informação sobre os autores



Ricardo Jorge Silvério de Magalhães Machado é Licenciado em Engenharia Electrotécnica e de Computadores (1994) pela FEUP e Mestre em Informática (1996) pela U.Minho. Foi Investigador do INESC-Porto e da ENSEA-Paris. Como Engenheiro da Texas Instruments, registou, nos EUA, a patente dum sistema electrónico. Investiga na área do co-projecto hw/sw e está a preparar o Doutoramento em Engenharia de Computadores no Dep. de Informática da U.Minho, onde é Assistente.



João Miguel Lobo Fernandes é Licenciado em Engenharia de Sistemas e Informática (1991) e Mestre em Informática (1994) pela U.Minho. Foi Investigador da U.Bristol-UK. Está a preparar o Doutoramento em Engenharia de Computadores no Dep. de Informática da U.Minho, onde é Assistente. Desenvolve investigação na área dos Sistemas Digitais, onde tem várias publicações em congressos e revistas internacionais.



Alberto José Proença é Licenciado em Engenharia Electrotécnica (1976) pela FCTUC e obteve os graus de Mestre e Doutor (1982) em Sistemas Digitais pela UMIST-UK. É Professor Associado no Dep. Informática e Presidente do Centro de Informática na U.Minho. Interesses em I&D: sistemas concorrentes e digitais, na integração da concorrência ao nível do software (computação paralela) com a concorrência híbrida (técnicas de co-projecto em hardware/software e de reconfiguração dinâmica).