

Conteúdo

1	Revisão Bibliográfica	1
1.1	Classes de Reconfiguração	1
1.1.1	Reconfiguração Total	1
1.1.2	Reconfiguração Parcial	1
1.1.3	Reconfiguração Estática	2
1.1.4	Reconfiguração Dinâmica	2
1.1.5	Autorreconfiguração	3
1.2	Ferramentas	3
1.2.1	Xilinx ISE Design Suite	3

Capítulo 1

Revisão Bibliográfica

Este capítulo visa apresentar uma breve descrição sobre reconfiguração dinâmica, autorreconfiguração e as ferramentas necessárias para suas realizações.

Com a chegada de CPLDs e FPGAs, requisitos cada vez mais complexos foram sendo introduzidos ao projeto de sistemas digitais. Tais requisitos forçaram as ferramentas de síntese a suportar diferentes classes de reconfiguração. Note que reconfiguração diz respeito, como dito anteriormente, a modificação do comportamento, ou configuração, de um dispositivo reconfigurável.

1.1 Classes de Reconfiguração

1.1.1 Reconfiguração Total

A reconfiguração total, herdada da tecnologia tradicional, compreende a mudança do comportamento de todo o dispositivo reconfigurável, sem exceção de blocos lógicos. Tal reconfiguração é bastante dispendiosa, visto que maior parte das alterações realizadas são incrementais e dizem respeito à apenas uma pequena parte do dispositivo. Apesar disto, todos os FPGAs dão suporte a este tipo de reconfiguração.

1.1.2 Reconfiguração Parcial

A reconfiguração parcial, ao contrário da reconfiguração total, diz respeito à programação de apenas parte de um dispositivo reconfigurável (??). Para tal, faz-se necessário a divisão do dispositivo nas chamadas partições, cada uma com sua configuração individual. Desta forma, mudanças feitas em uma partição não afetam as outras, acelerando o processo de síntese e programação. Outro processo que é acelerado é o de roteamento, uma vez que o particionamento introduz limitações no mapeamento das funções. Nem todos os FPGAs dão suporte a este tipo de reconfiguração, que pode ser realizado tanto de forma dinâmica (1.1.4) quando estática (1.1.3).

1.1.3 Reconfiguração Estática

O termo reconfiguração estática se refere a programação de um dispositivo reconfigurável enquanto ele não estiver executando. No caso em que alguma programação já tenha sido transferida para ele e ele a esteja executando, esta é parada para que o dispositivo seja configurado novamente. Por ser mais fácil de ser implementada e não necessita de circuitos adicionais, todos os FPGAs dão suporte a este tipo de reconfiguração.

1.1.4 Reconfiguração Dinâmica

A reconfiguração dinâmica acontece frente à necessidade de reprogramação parcial do dispositivo sem que ele pare de funcionar. As funcionalidades modificadas são interrompidas e substituídas sem afetar o funcionamento do todo. Normalmente este processo, quando não associado a autorreconfiguração, é realizado através de um circuito externo à FPGA, tal como um controlador, um microcontrolador, ou mesmo um computador, como apresentado na figura 1.1. Quase todos os FPGAs modernos dão suporte a esta tecnologia.

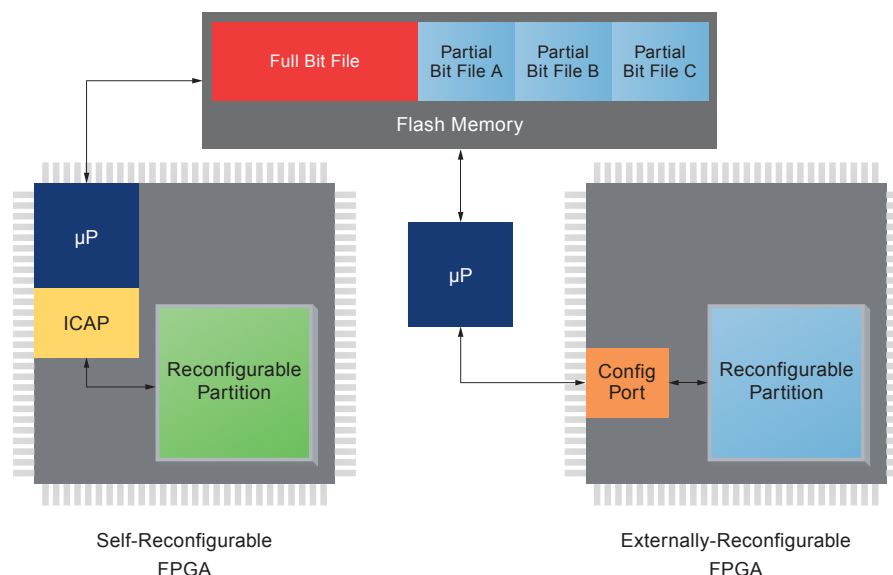


Figura 1.1: Imagem ilustrativa para diferenciação entre autorreconfiguração e reconfiguração externa, extraído de (??).

É implícito o uso da reconfiguração parcial com a reconfiguração dinâmica, uma vez que faz pouco sentido reconfigurar todo o FPGA enquanto ela ainda está em execução. Note que este tipo de reconfiguração em geral também necessita de uma parte permanentemente estática, para interfacear com o circuito controlador. Esta partição estática é responsável pelo menos por controlar a comunicação com o circuito controlador.

Vale lembrar que a este tipo de reconfiguração, apesar de abrir muitas possibilidades, introduz uma necessidade de preocupação com *overheads* de reconfiguração (??). Este *overhead* é proporcional ao tamanho da partição que se deseja modificar e inversamente proporcional às velocidades das interfaces de

reconfiguração. Tal fator pode se tornar crucial na escolha entre o uso desta tecnologia ou alguma outra alternativa, possivelmente multiplexada. Note que existem outros fatores que influenciam na opção por reconfiguração dinâmica, tais como preço, capacidade e potência, dentre outros, que não serão considerados aqui.

1.1.5 Autorreconfiguração

Modalidade de reconfiguração dinâmica parcial onde a reconfiguração do dispositivo é decidida por uma lógica pertencente a ele mesmo. Normalmente usa-se um microcontrolador ou uma máquina de estados finitos para controlar a mudança de configurações. Esta tecnologia é nova e ainda representa uma forte área de pesquisa. Por isso não são todos os FPGAs que dão suporte a este tipo de reconfiguração.

Para que a autorreconfiguração aconteça, os *bitstreams*, resultado da sintetização, devem ser passados para uma memória acessível ao FPGA durante a execução do mesmo. O circuito controlador identifica então um padrão que defina a necessidade de reconfiguração e transfere o *bitstream* correspondente a esta nova necessidade para a partição destino, que assim muda seu comportamento. Note que para tal, as entradas e saídas das partições tem que ser fixas, para que a mudança nas configurações das partições não danifique o FPGA em si.

1.2 Ferramentas

Diversas ferramentas foram utilizadas para a realização deste trabalho. Dentre eles pode-se citar os programas da Xilinx, interpretadores da linguagem Python, Perl e Tcl, compiladores de L^AT_EX e a ferramenta de controle de versão Git. Abaixo segue uma pequena descrição sobre as ferramentas mais críticas destas.

1.2.1 Xilinx ISE Design Suite

O *ISE Design Suite* é um conjunto de ferramentas da Xilinx para o desenvolvimento de projetos de *hardware*. Estes programas estão apresentados a seguir.

1.2.1.1 Xilinx ISE Design Tools

O *ISE Design Tools*, disponível para os sistemas Windows XP (32 e 64 bits), Windows 7 (32 e 64 bits), Windows Server 2008 (64 bits), Red Hat Enterprise 5 e 6 (32 e 64 bits) e SUSE Linux Enterprise 11 (32 e 64 bits) (??), controla todos os aspectos do fluxo de projeto (??). Através do *Project Navigator*, sua principal ferramenta, é possível acessar todas as configurações e ferramentas de implementação de configurações.

Project Navigator O *Project Navigator* é a principal ferramenta do *ISE Design Tools*. Através dela é possível criar projetos, incluir arquivos de descrição de *hardware*, seja em VHDL, Verilog ou esquemáticos,

construir componentes de propriedade intelectual, impor restrições e compilá-los, dentre outras coisas. Em geral, todo o desenvolvimento começa através desta ferramenta para fins de verificação de lógica.

iMPACT O iMPACT é utilizado para se construir arquivos para a inicialização de memórias Flash e para a programação de FPGAs e suas memórias. Ele pode ser acessado por linha de comando, permitindo que outras ferramentas incorporem suas funções.

CORE Generator O *CORE Generator* é uma ferramenta muito útil, utilizada para a construção de blocos lógicos prontos de funções variadas. Em geral, seus blocos instanciam algum componente de propriedade intelectual da Xilinx.

1.2.1.2 Embedded Development Kit

O *Embedded Development Kit* é um conjunto de ferramentas voltados para o desenvolvimento de sistemas microprocessados embarcados em FPGA.

Xilinx Platform Studio A *Xilinx Platform Studio* permite a construção de sistemas microprocessados. Ela possui uma gama muito grande de componentes periféricos que podem acrescentados a este sistema e integrados de forma automática ou manual.

Xilinx Software Development Kit Com as informações do sistema microprocessado, é possível utilizar *Xilinx Software Development Kit* para a criação do programa que será embarcado. Esta ferramenta permite a programação do dispositivo, bem como da memória Flash, com ou sem o programa já adicionado, tornando-se uma ferramenta muito simples e útil.

1.2.1.3 PlanAhead

O PlanAhead é uma das ferramentas mais poderosas de toda a *suite*. Ele permite que projetos sejam integrados e pré-alocados em FPGAs, bem como gerenciar, modificar e verificar restrições de tempo, alocação e mapeamento, dentre outros. Ele permite ainda que módulos reconfiguráveis sejam acrescentados ao projeto, tornando a reconfiguração dinâmica um problema bem mais acessível.

1.2.1.4 Ferramentas de Linha de Comando

Além de todas as ferramentas já apresentadas, a plataforma da Xilinx ainda oferece ferramentas de linha de comando, acessadas pelo *prompt* to Windows. Através destas ferramentas é possível realizar qualquer tarefa realizavel pela ferramentas com interface gráfica e mais algumas outras. Abaixo menciona-se a ferramenta mais importante e mais utilizada deste conjunto.

Xilinx Sinthesis Tool (XST) A XST é o equivalente ao compilador para a programação convencional. Ele realiza verificações léxicas e sintáticas dos arquivos para gerar na saída algum arquivo compilado, tipicamente NGC ou BIT.