ChipCFlow – Uma Ferramenta para execução de Algoritmos Utilizando o Modelo a Fluxo de Dados Dinâmico em Hardware Reconfígurável – Organização de Memória

Rodrigo P. Zeli[1], Jorge L. Silva[1]

[1]Instituto de Ciências Matemáticas e Computação – Universidade de São Paulo (USP)
São Carlos – SP – Brasil

zeli@grad.icmc.usp.br, jsilva@icmc.usp.br

Abstract. *To generate algorithms using the Dataflow Model, associated to the concept of reconfigurable hardware, to accelerate the execution of application programs written in C, in particular, in the intensive parts of the program, and to execute this algorithms direct into the hardware, is the main characteristic of the ChipCflow project, that means to provide parallelism in a natural form. In this paper, the structure of the Memory Organization to supply data to the dynamic dataflow graphs is presented and some proof-of-concept is discussed to end of the paper. The implementation is based on Xilinx Spartan 3E Starter Platform.*

Resumo. *A Ferramenta ChipCFow vem sendo desenvolvida nos últimos dois anos, e trata-se de uma ferramenta de compilação que executa algoritmos através do Modelo de Arquitetura a Fluxo de Dados associado ao conceito de arquiteturas reconfiguráveis, com o objetivo de acelerar a execução de programas de aplicação escritos em C, aproveitando ao máximo o paralelismo considerado natural nesse modelo e as características do hardware reconfigurável. Uma das bases para isso é possuir uma organização de memória capaz de atender aos requisitos de velocidade e demanda da ferramenta. Este artigo descreve um protótipo da organização de memória que será utilizada na ferramenta ChipCFlow.*

1. Introdução

Durante os anos 70 até meados dos anos 90 já existiam algumas implementações envolvendo arquitetura a fluxo de dados, cuja principal diferença em relação à tradicional arquitetura de Von Neumann é que esta é baseada no controle seqüencial, enquanto que nas máquinas a fluxo de dados o controle é executado pela presença dos dados. A representação dessa arquitetura é feita por meio de um grafo a fluxo de dados, que é um conjunto de operadores interligados por arcos, onde a presença de dados em cada arco de um operador irá disparar sua execução [ARVIND, 2005], [DENNIS, 1975] e [VEEN, 1986].

Existem dois tipos de arquiteturas a fluxo de dados: estática e dinâmica. No modelo estático apenas um dado está presente em um arco a espera de seus parceiros para completar a execução. Um protocolo deve garantir o sincronismo entre os

operadores para garantir a permanência de apenas um dado por arco, limitando o paralelismo ao conjunto de operadores cujos arcos possuem apenas um dado.

No modelo dinâmico, vários dados podem estar presentes em um mesmo arco. Nesse caso o protocolo deve garantir que as operações só sejam disparadas quando todos os dados necessários estejam presentes nos arcos. A metodologia "Tagged token" pode ser utilizada no protocolo para controlar os dados parceiros em cada arco. Neste modelo, o paralelismo acontece quando da execução dos vários operadores cujos dados estejam todos disponíveis [VEEN, 1986] [SWANSON,2006].

A computação reconfigurável, por sua vez, ainda é uma área em desenvolvimento e a validação de novos e diferentes tópicos nessa arquitetura tem sido um desafio dos pesquisadores da área nos últimos 20 anos.

O que tem tornado a computação reconfigurável uma realidade são os FPGAs, já que inúmeros sistemas com aplicações específicas e complexas vêm utilizando essa tecnologia em seus projetos, pois ela reduz os riscos de desenvolvimento e ao mesmo tempo proporciona flexibilidade para correções e atualizações de forma rápida e segura.

A Ferramenta ChipCFlow trata-se de uma ferramenta de compilação que executa algoritmos através do Modelo de Arquitetura a Fluxo de Dados associado ao conceito de arquiteturas reconfiguráveis, com o objetivo de acelerar a execução de programas de aplicação escritos em C, aproveitando ao máximo o paralelismo considerado natural do modelo a Fluxo de Dados e das características do hardware reconfigurável.

Para se obter o máximo da arquitetura a Fluxo de dados dinâmica, é necessário dispor de organização de memória que forneça e receba dados, na velocidade de trabalho da arquitetura a Fluxo de Dados. A seguir é descrita a proposta do modelo de organização de memória sendo desenvolvido para o projeto.

2. Organização de Memória

A arquitetura a fluxo de dados é baseada em operadores que executam paralelamente realizando operações aritméticas, lógicas e de controle. Como o próprio nome já menciona diferentemente das arquiteturas tradicionais, na arquitetura a fluxo a dados são os dados que definem o ritmo e as características de execução que nas arquiteturas tradicionais são tratadas por diretivas de controle. Por isso, uma demora no fornecimento de dados afeta drasticamente toda a execução do programa, acarretando uma gigantesca perda de performance.

Em particular, no projeto ChipCFlow, um operador roda a 6ns [LOPES, 2008], e para que a organização de memória seja transparente para o operador, esta deve fornecer dados em tempos inferiores a 6ns. Além disso, como qualquer aplicação da ferramenta é composta por diversos operadores de entrada, devemos supor diversas entradas em paralelo, nas quais os dados devem ser fornecidos na forma de rajada de dados. Tal situação pode ser visualizada na Figura 1.

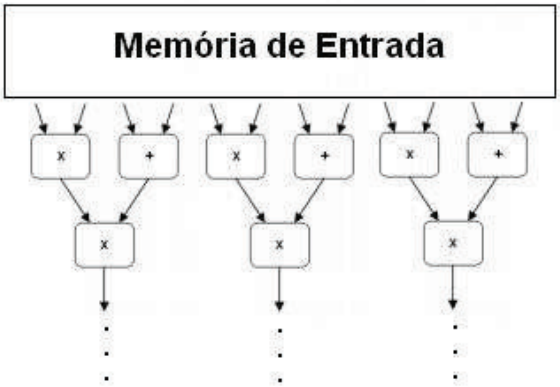

Figure 1. Representação da Memória de Entrada abastecendo múltiplos operadores

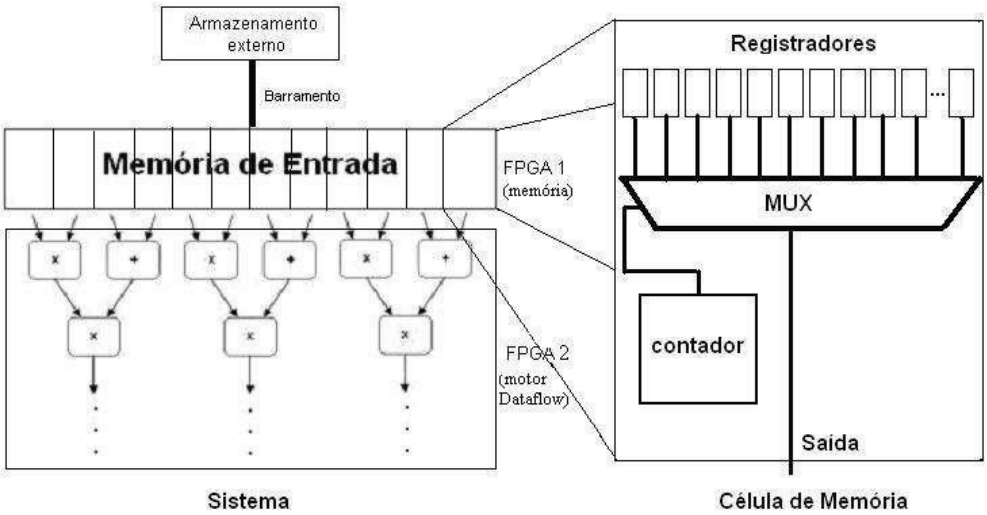

Figure 2. Representação da Estrutura de Memória

Como as entradas são definidas a partir do programa C, a organização de memória também deve ser passível de reestruturação em tempo de execução.

Visando atender primeiramente aos requisitos de tempo e de múltiplas entradas, como proposta inicial optou-se por uma estrutura extremamente simples. Na intenção de diminuir ao mínimo o tempo gasto no acesso a memória, optou-se por não utilizar nenhuma unidade de controle complexa. Decidiu-se então definir a seguinte estrutura: um banco de 2^n registradores de 32 bits selecionados através de um contador de n bits e um multiplexador de 2^n entradas de 32 bits. Dessa forma, o tempo gasto pela memória fica restrito apenas ao tempo de comutação do multiplexador.

Para lidar com múltiplos operadores recria-se a estrutura acima mencionada para cada uma das duas entradas de cada operador. Porém, como a implementação é baseada

em FPGA, dispõem-se apenas da capacidade interna do FPGA para construir os registradores, temos então que ***n*** é inversamente proporcional ao número de entradas, por isso o valor de ***n*** deve ser definido de acordo com o número de operadores de entrada e de acordo com a capacidade do FPGA utilizado.

Dada a importância da memória para a ferramenta, bem como a grande quantidade de recursos do FPGA por ela ocupada, optou-se por estruturar a memória e a lógica da ferramenta (motor Dataflow) em FPGAs diferentes, a fim de permitir uma maior quantidade de memória e total independência de funcionamento das duas partes. Isso pode ser melhor observado na Figura 2 sendo o FPGA 1 o que contém a síntese de memória e o FPGA 2 o que contém o motor Dataflow da aplicação. Dessa maneira, a memória inicialmente é carregada a partir de um dispositivo de armazenamento secundário (Hard Disk, Compact Flash) e somente depois inicia o fornecimento de dados para a ferramenta.

3. Conclusão

Neste artigo foi descrito o projeto ChipCFlow, uma ferramenta para execução de algoritmos escritos originalmente em C e convertido no modelo a Fluxo de Dados a ser executado diretamente em hardware reconfigurável e os requisitos necessários para a implementação de uma memória que seja capaz de responder na velocidade dos operadores em rajadas de dados, à cada 6ns, que é o tempo de processamento de cada operador no projeto ChipCflow. O projeto encontra-se em desenvolvimento e novos resultados deverão ser publicados brevemente.

Referências

ARVIND. (2005) DataFlow: Passing the token ISCA Keynote.

DENNIS, J. B. (1975) "A preliminary architecture for a basic dataflow processor", Proceedings of the 2nd Annual Symposium on Computer Architecture.

LOPES, J. J.,(2008) ChipCFlow - uma ferramenta para execução de algoritmos utilizando o modelo a Fluxo de dados dinâmico em hardware reconfigurável. Monografia apresentada no Programa de Pós-Graduação do ICMC-USP São Carlos, para obtenção do título de Doutor. São Carlos, SP, Brasil.

SWANSON, S. et al (2006) Area-Performance Trade-o® in Tiled DataFlow Architectures In-11ternational Symposium on Computer Architecture (ISCA).

VEEN, A. H. (1986) DataFlow Machine Architecture ACM Computing Surveys number, 365-396.