πFlowMR

um protótipo dataflow escalável, implementado em um cluster de FPGAs de baixo custo

Silva Junior, J. T.[1]**; Matias P.**[2]**; Ruggiero, C.** A.[1]

[1]Instituto de Física de São Carlos - Universidade de São Paulo
[2]Departamento de Computação - Universidade Federal de São Carlos

Índice

Introdução

Motivação

- Explorar recursos paralelos na arquitetura de *Von Neumann* exige muito esforço do programador;
- A computação paralela pode remover a necessidade de intervenção explícita no código do programa; [Barahona and Gurd, 1986]
- Computação dirigida por dados:
  - Instruções são executadas a partir de suas disponibilidades; [Arvind and Culler, 1986]
  - Modelo a fluxo de dados; [Sharp, 1992]
  - Máquina Dataflow de Manchester (*MDFM*). [Gurd, 1985]

Objetivos

A granularidade da MDFM garante a exploração de uma quantidade significativa de paralelismo. Mas essa característica exige uma estrutura demasiada robusta. [Sargeant and Ruggiero, 1987]

- Estudo do Multianel, descentralizando a execução dos programas: [Barahona and Gurd, 1986]
  - Aumento sensível na granularidade da máquina;
  - Aumento considerável na quantidade de unidades de processamento, explorando sua capacidade de escalabilidade.

Modelo a fluxo de dados

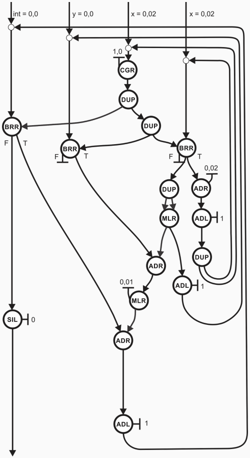

Figura 1: Representação gráfica da implementação de um programa que realiza a integração da função $y = x^2$ no modelo a fluxo de dados. [Gurd, 1985]

Máquina Dataflow de Manchester

Composição de uma Ficha:
<dado (37-bits); rótulo (36-bits); destino (22-bits); marcador (1-bit)>

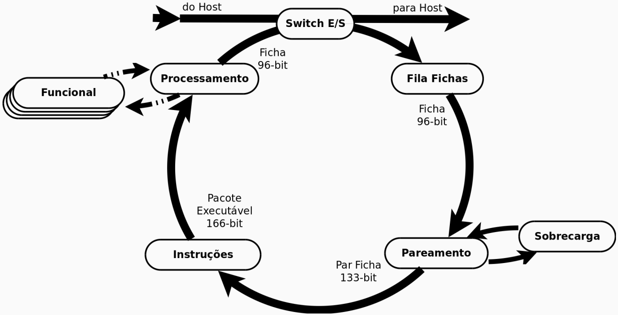

Figura 2: Visão geral da Máquina Dataflow de Manchester. Uma ficha é composta por 4 informações principais que armazenam informações diversas sobre a instrução que ela deve executar. [Gurd, 1985]

πFlowMR

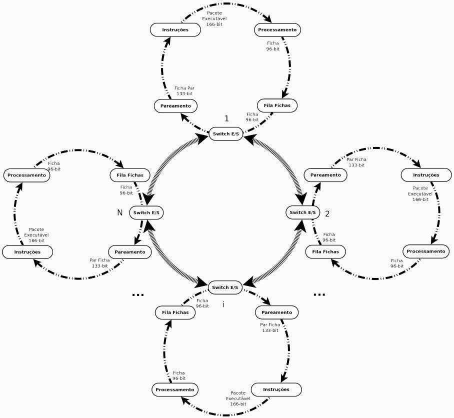

Figura 3: Visão geral da πFlowMR e a topologia proposta para as placas de FPGA utilizadas para sua composição. Note que a comunicação é realizada em um formato cíclico, bem como a estrutura de um único anel.

Distribuição de fichas

Para a descoberta do anel ao qual pertence uma ficha, foi utilizado apenas o rótulo dinâmico da mesma.

- Cada anel, passa a ser equivalente a uma Unidade Funcional da estrutura;
- Essa característica é suficiente para aumentar sensivelmente a granularidade da arquitetura; [Magna, 1997]
- Apesar disso, cada anel mantém a granularidade da MDFM;

Resultados

Últimos resultados

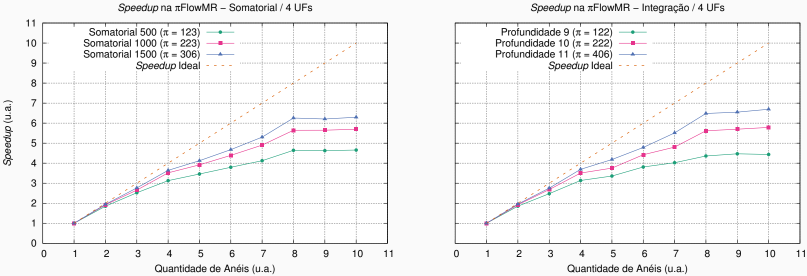

Figura 4: Resultados obtidos nas últimas versões do desenvolvimento, com maior largura de banda nos módulos de comunicação entre os anéis da estrutura da máquina.

Escalabilidade

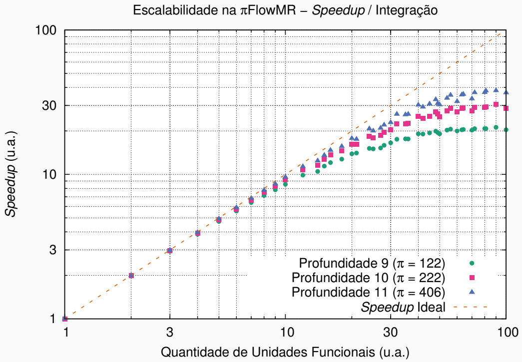

Figura 5: Na implementação mais recente podemos verificar que o desempenho está muito próximo daquele esperado para uma arquitetura paralela, considerando as limitações impostas pelo algoritmo do programa e gargalos da arquitetura.

Obrigado!

Perguntas?

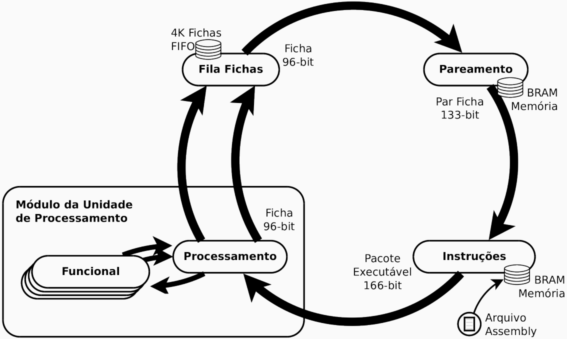

Figura 6: Visão geral da pimeira versão da πFlow. Note a semelhança com a estrutura da MDFM, a menos da dupla ligação entre a PU e a TQ, além de alguns artifícios característicos do Bluespec destacados.

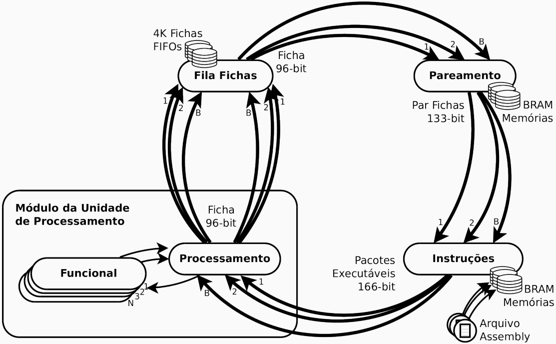

Figura 7: Visão geral da πFlowBW, introduzindo o conceito de dutos de largura de banda ao projeto. Nesta versão, o *throughput* da máquina é proporcional a um valor constante.

Síntese da solução

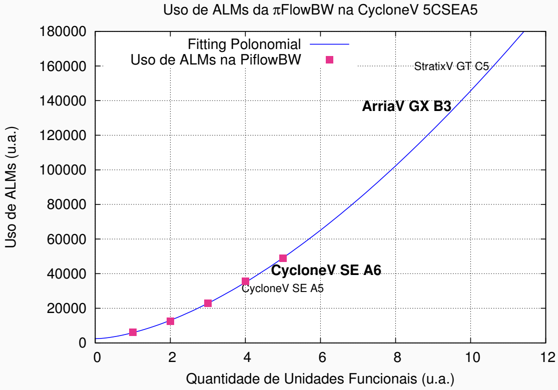

Figura 8: Apesar dos resultados terem sido obtidos em simulação, fizemos a análise da porcentagem de utilização dos recursos das FPGAs como função da quantidade de Unidades Funcionais na estrutura do anel.

Modelo a fluxo de dados

```
export Integrate
function Integrate (returns real)
for initial
    int := 0.0;
    y   := 0.0;
    x   := 0.02;
while
    x < 1.0
repeat
    int := 0.01 * (old y + y);
    y   := old x * old x;
    x   := old x + 0.02
returns
    value of sum int
end for
end function
```

Figura 9: Programa escrito na linguagem SISAL, implementa o cálculo da integração da curva $y = x^2$, nos limites de 0.0 à 1.0. [Gurd, 1985]

Modelo a fluxo de dados

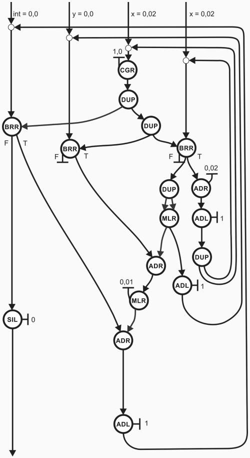

Figura 9: Representação gráfica de um programa implementado para realizar a integração da função $y = x^2$ entre os limites de 0 à 1 no modelo a fluxo de dados. [Gurd, 1985]

Modelo a fluxo de dados

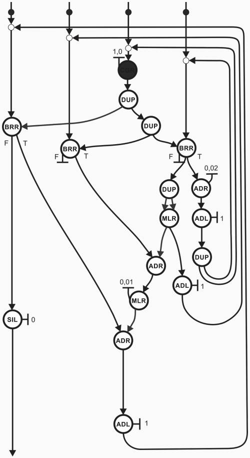

Figura 9: Exemplo de execução na representação gráfica do programa de integração no modelo a fluxo de dados, para o cálculo da integral da função $y = x^2$, entre os limites de 0 à 1. [Gurd, 1985]

Modelo a fluxo de dados

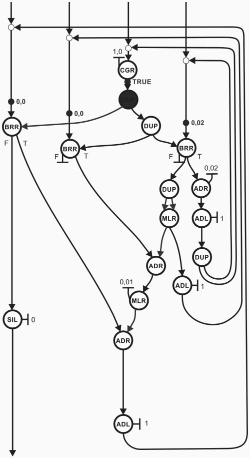

Figura 9: Exemplo de execução na representação gráfica do programa de integração no modelo a fluxo de dados, para o cálculo da integral da função $y = x^2$, entre os limites de 0 à 1. [Gurd, 1985]

Modelo a fluxo de dados

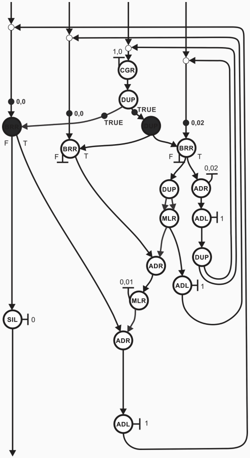

Figura 9: Exemplo de execução na representação gráfica do programa de integração no modelo a fluxo de dados, para o cálculo da integral da função $y = x^2$, entre os limites de 0 à 1. [Gurd, 1985]

Modelo a fluxo de dados

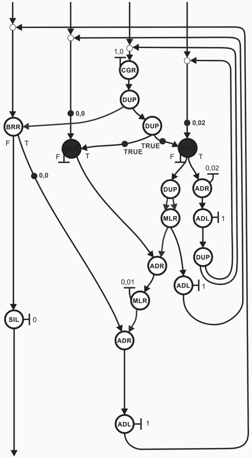

Figura 9: Exemplo de execução na representação gráfica do programa de integração no modelo a fluxo de dados, para o cálculo da integral da função $y = x^2$, entre os limites de 0 à 1. [Gurd, 1985]

Modelo a fluxo de dados

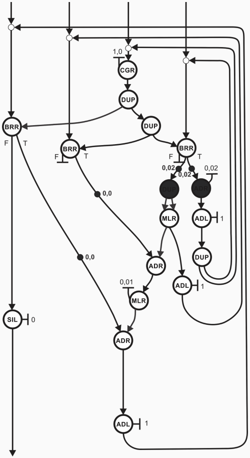

Figura 9: Exemplo de execução na representação gráfica do programa de integração no modelo a fluxo de dados, para o cálculo da integral da função $y = x^2$, entre os limites de 0 à 1. [Gurd, 1985]

Modelo a fluxo de dados

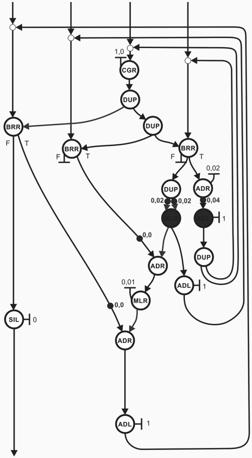

Figura 9: Exemplo de execução na representação gráfica do programa de integração no modelo a fluxo de dados, para o cálculo da integral da função $y = x^2$, entre os limites de 0 à 1. [Gurd, 1985]

Modelo a fluxo de dados

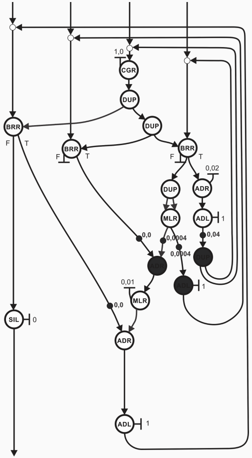

Figura 9: Exemplo de execução na representação gráfica do programa de integração no modelo a fluxo de dados, para o cálculo da integral da função $y = x^2$, entre os limites de 0 à 1. [Gurd, 1985]

Modelo a fluxo de dados

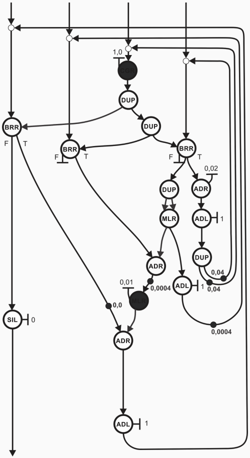

Figura 9: Exemplo de execução na representação gráfica do programa de integração no modelo a fluxo de dados, para o cálculo da integral da função $y = x^2$, entre os limites de 0 à 1. [Gurd, 1985]

Métricas para avaliação de desempenho

- ***Speedup***: Desempenho da máquina utilizando uma única unidade funcional, com relação ao desempenho utilizando N unidades funcionais.

$$S_N = T_1/T_N \tag{1}$$

- **Paralelismo médio**: Quantidade de nós do grafo em fluxo de dados, com relação ao caminho crítico do grafo.

$$\pi_{av} = S_1/S_\infty \tag{2}$$

Programas de validação

Para validação da estrutura foram utilizados dois programas implementados de forma duplamente recursiva, gerados pelo compilador da MDFM.

```
export Main
function Fact (x, y : integer returns integer)
    if x <= y then y
    else Fact(x, (x+y)/2+1) + Fact((x+y)/2, y)
    end if
end function

function Main (returns integer)
    Fact(5, 1)
end function
```

```
export Main
function F (x : integer returns integer)
  x*x-6*x-10
end function

function Area (l, r, depth : integer returns integer)
  let
     mid := (l+r)/2;
  in
     if depth <= 1 then
         (r-l)*F(mid)
     else
          Area(l, mid, depth-1) + Area(mid, r, depth-1)
     end if
  end let
end function
```

Figura 10: À esquerda a implementação do programa para o cálculo do fatorial de um valor definido na função principal. À direita a implementação do programa para o cálculo da integral da função $f(x) = x^2 - 6x - 10$.

Referências

Arvind and David E. Culler. Annual review of computer science vol. 1, 1986. chapter Dataflow Architectures, pages 225–253. Annual Reviews Inc., Palo Alto, CA, USA, 1986. ISBN 0-8243-3201-6. URL `http://dl.acm.org/citation.cfm?id=17814.17824`.

P. M. C. C. Barahona and J. R. Gurd. Processor allocation in a multi-ring dataflow machine. *Journal of Parallel and Distributed Computing*, 3: 305–327, 1986.

J. R. Gurd. The manchester dataflow machine. *Computer Physics Communications*, 37(1):49–62, 1985.

P. Magna. *Proposta e Simulação de uma Arquitetura a Fluxo de Dados de Segunda Geração*. PhD thesis, Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 1997.

J. Sargeant and C. A. Ruggiero. Control of parallelism in the manchester dataflow machine. *Lecture Notes in Computer Science*, 274:1–15, 1987.

J. A. Sharp. *Dataflow computing*. Ablex Publishing Company, New York, 1992.