Indentificação de Instruções Complexas e Potenciais Aplicações

Alexandre S. Nery[1]

[1]Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, Brasil

25 de novembro de 2016

PESC Programa de Engenharia de Sistemas e Computação

UERJ

TU/e Technische Universiteit Eindhoven University of Technology

Quem sou eu?

- Bacharel em Ciência da Computação – Universidade Católica de Brasília (2006)
  - Maior índice VIDA (Valor Indicativo de Desempenho Acadêmico)
- Mestre (2010) e Doutor (2014) PESC/COPPE
  - Linha: Arquitetura de Computadores e Sistemas Operacionais (ASO)
  - Dissertação Mestrado: *"GridRT: Uma Arquitetura Paralela para Ray Tracing Utilizando Volumes Uniformes"*
    - Orientadores: Felipe M.G. França (PESC) e Nadia Nedjah (UERJ)
  - Tese de Doutorado: *"Automatic Complex Instruction Identification with Hardware Sharing for Efficient Application Mapping onto ASIPs"* – *Prêmio Gilberto Velho*
    - Sanduíche: Universidade Técnica de Eindhoven (TU/e) – quase 2 anos
    - Orientadores: Felipe M.G. França (PESC), Nadia Nedjah (UERJ), Henk Corporaal (TU/e), Lech Józwiak (TU/e)
- Prof. Instituto de Matemática e Estatística – UERJ/DICC
  - Prof. Permanente Mestrado Ciências Computacionais – UERJ/CCOMP
- Áreas de interesse:
  - Computação de Alto Desempenho (Programação e Arquiteturas Paralelas)
  - Sistemas Embarcados (Computação Reconfigurável, High-Level Synthesis)
  - Computação Gráfica (Algoritmos de Renderização)

Quem sou eu?

- Bacharel em Ciência da Computação – Universidade Católica de Brasília (2006)
  - Maior índice VIDA (Valor Indicativo de Desempenho Acadêmico)
- Mestre (2010) e Doutor (2014) PESC/COPPE
  - Linha: Arquitetura de Computadores e Sistemas Operacionais (ASO)
  - Dissertação Mestrado: *"GridRT: Uma Arquitetura Paralela para Ray Tracing Utilizando Volumes Uniformes"*
    - Orientadores: Felipe M.G. França (PESC) e Nadia Nedjah (UERJ)
  - Tese de Doutorado: *"Automatic Complex Instruction Identification with Hardware Sharing for Efficient Application Mapping onto ASIPs"* – *Prêmio Gilberto Velho*
    - Sanduíche: Universidade Técnica de Eindhoven (TU/e) – quase 2 anos
    - Orientadores: Felipe M.G. França (PESC), Nadia Nedjah (UERJ), Henk Corporaal (TU/e), Lech Józwiak (TU/e)
- Prof. Instituto de Matemática e Estatística – UERJ/DICC
  - Prof. Permanente Mestrado Ciências Computacionais – UERJ/CCOMP
- Áreas de interesse:
  - Computação de Alto Desempenho (Programação e Arquiteturas Paralelas)
  - Sistemas Embarcados (Computação Reconfigurável, High-Level Synthesis)
  - Computação Gráfica (Algoritmos de Renderização)

Quem sou eu?

- Bacharel em Ciência da Computação – Universidade Católica de Brasília (2006)
  - Maior índice VIDA (Valor Indicativo de Desempenho Acadêmico)
- Mestre (2010) e Doutor (2014) PESC/COPPE
  - Linha: Arquitetura de Computadores e Sistemas Operacionais (ASO)
  - Dissertação Mestrado: *"GridRT: Uma Arquitetura Paralela para Ray Tracing Utilizando Volumes Uniformes"*
    - Orientadores: Felipe M.G. França (PESC) e Nadia Nedjah (UERJ)
  - Tese de Doutorado: *"Automatic Complex Instruction Identification with Hardware Sharing for Efficient Application Mapping onto ASIPs"* – *Prêmio Gilberto Velho*
    - Sanduíche: Universidade Técnica de Eindhoven (TU/e) – quase 2 anos
    - Orientadores: Felipe M.G. França (PESC), Nadia Nedjah (UERJ), Henk Corporaal (TU/e), Lech Józwiak (TU/e)
- Prof. Instituto de Matemática e Estatística – UERJ/DICC
  - Prof. Permanente Mestrado Ciências Computacionais – UERJ/CCOMP
- Áreas de interesse:
  - Computação de Alto Desempenho (Programação e Arquiteturas Paralelas)
  - Sistemas Embarcados (Computação Reconfigurável, High-Level Synthesis)
  - Computação Gráfica (Algoritmos de Renderização)

Quem sou eu?

- Bacharel em Ciência da Computação – Universidade Católica de Brasília (2006)
  - Maior índice VIDA (Valor Indicativo de Desempenho Acadêmico)
- Mestre (2010) e Doutor (2014) PESC/COPPE
  - Linha: Arquitetura de Computadores e Sistemas Operacionais (ASO)
  - Dissertação Mestrado: *"GridRT: Uma Arquitetura Paralela para Ray Tracing Utilizando Volumes Uniformes"*
    - Orientadores: Felipe M.G. França (PESC) e Nadia Nedjah (UERJ)
  - Tese de Doutorado: *"Automatic Complex Instruction Identification with Hardware Sharing for Efficient Application Mapping onto ASIPs"* – *Prêmio Gilberto Velho*
    - Sanduíche: Universidade Técnica de Eindhoven (TU/e) – quase 2 anos
    - Orientadores: Felipe M.G. França (PESC), Nadia Nedjah (UERJ), Henk Corporaal (TU/e), Lech Józwiak (TU/e)
- Prof. Instituto de Matemática e Estatística – UERJ/DICC
  - Prof. Permanente Mestrado Ciências Computacionais – UERJ/CCOMP
- Áreas de interesse:
  - Computação de Alto Desempenho (Programação e Arquiteturas Paralelas)
  - Sistemas Embarcados (Computação Reconfigurável, High-Level Synthesis)
  - Computação Gráfica (Algoritmos de Renderização)

Quem sou eu?

- Bacharel em Ciência da Computação – Universidade Católica de Brasília (2006)
  - Maior índice VIDA (Valor Indicativo de Desempenho Acadêmico)
- Mestre (2010) e Doutor (2014) PESC/COPPE
  - Linha: Arquitetura de Computadores e Sistemas Operacionais (ASO)
  - Dissertação Mestrado: *"GridRT: Uma Arquitetura Paralela para Ray Tracing Utilizando Volumes Uniformes"*
    - Orientadores: Felipe M.G. França (PESC) e Nadia Nedjah (UERJ)
  - Tese de Doutorado: *"Automatic Complex Instruction Identification with Hardware Sharing for Efficient Application Mapping onto ASIPs"* – *Prêmio Gilberto Velho*
    - Sanduíche: Universidade Técnica de Eindhoven (TU/e) – quase 2 anos
    - Orientadores: Felipe M.G. França (PESC), Nadia Nedjah (UERJ), Henk Corporaal (TU/e), Lech Józwiak (TU/e)
- Prof. Instituto de Matemática e Estatística – UERJ/DICC
  - Prof. Permanente Mestrado Ciências Computacionais – UERJ/CCOMP
- Áreas de interesse:
  - Computação de Alto Desempenho (Programação e Arquiteturas Paralelas)
  - Sistemas Embarcados (Computação Reconfigurável, High-Level Synthesis)
  - Computação Gráfica (Algoritmos de Renderização)

Quem sou eu?

- Bacharel em Ciência da Computação – Universidade Católica de Brasília (2006)
  - Maior índice VIDA (Valor Indicativo de Desempenho Acadêmico)
- Mestre (2010) e Doutor (2014) PESC/COPPE
  - Linha: Arquitetura de Computadores e Sistemas Operacionais (ASO)
  - Dissertação Mestrado: *"GridRT: Uma Arquitetura Paralela para Ray Tracing Utilizando Volumes Uniformes"*
    - Orientadores: Felipe M.G. França (PESC) e Nadia Nedjah (UERJ)
  - Tese de Doutorado: *"Automatic Complex Instruction Identification with Hardware Sharing for Efficient Application Mapping onto ASIPs"* – *Prêmio Gilberto Velho*
    - Sanduíche: Universidade Técnica de Eindhoven (TU/e) – quase 2 anos
    - Orientadores: Felipe M.G. França (PESC), Nadia Nedjah (UERJ), Henk Corporaal (TU/e), Lech Józwiak (TU/e)
- Prof. Instituto de Matemática e Estatística – UERJ/DICC
  - Prof. Permanente Mestrado Ciências Computacionais – UERJ/CCOMP
- Áreas de interesse:
  - Computação de Alto Desempenho (Programação e Arquiteturas Paralelas)
  - Sistemas Embarcados (Computação Reconfigurável, High-Level Synthesis)
  - Computação Gráfica (Algoritmos de Renderização)

Quem sou eu?

- Bacharel em Ciência da Computação – Universidade Católica de Brasília (2006)
  - Maior índice VIDA (Valor Indicativo de Desempenho Acadêmico)
- Mestre (2010) e Doutor (2014) PESC/COPPE
  - Linha: Arquitetura de Computadores e Sistemas Operacionais (ASO)
  - Dissertação Mestrado: *"GridRT: Uma Arquitetura Paralela para Ray Tracing Utilizando Volumes Uniformes"*
    - Orientadores: Felipe M.G. França (PESC) e Nadia Nedjah (UERJ)
  - Tese de Doutorado: *"Automatic Complex Instruction Identification with Hardware Sharing for Efficient Application Mapping onto ASIPs"* – *Prêmio Gilberto Velho*
    - Sanduíche: Universidade Técnica de Eindhoven (TU/e) – quase 2 anos
    - Orientadores: Felipe M.G. França (PESC), Nadia Nedjah (UERJ), Henk Corporaal (TU/e), Lech Józwiak (TU/e)
- Prof. Instituto de Matemática e Estatística – UERJ/DICC
  - Prof. Permanente Mestrado Ciências Computacionais – UERJ/CCOMP
- Áreas de interesse:
  - Computação de Alto Desempenho (Programação e Arquiteturas Paralelas)
  - Sistemas Embarcados (Computação Reconfigurável, High-Level Synthesis)
  - Computação Gráfica (Algoritmos de Renderização)

Projetos

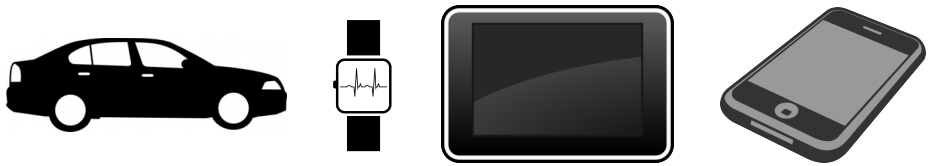
- Colaboração:
  - Laboratório LAM/PESC
  - Electronic Systems Group – TU/e (Holanda)
- Alguns projetos:
  - Arquiteturas paralelas em FPGA
    - GridRT – Arquitetura paralela para Ray-Tracing
    - MPSoC para Ray-Tracing e Volume Ray-Casting
  - Microprogramação
    - Instrução complexa microprogramada para Ray-Tracing
  - Extensão do conjunto de instruções de ASIPs
    - Identificação automática de instruções complexas
    - MPSoC para Ray-Tracing e Volume Ray-Casting com instruções complexas
  - Síntese de Circuitos
    - Compartilhamento de Hardware
  - Arquiteturas e Modelos Dataflow
    - JSucuri - Versão java da Sucuri (A Minimalistic Dataflow Programming Library for Python)

Projetos

- Colaboração:
  - Laboratório LAM/PESC
  - Electronic Systems Group – TU/e (Holanda)
- Alguns projetos:
  - Arquiteturas paralelas em FPGA
    - GridRT – Arquitetura paralela para Ray-Tracing
    - MPSoC para Ray-Tracing e Volume Ray-Casting
  - Microprogramação
    - Instrução complexa microprogramada para Ray-Tracing
  - Extensão do conjunto de instruções de ASIPs
    - Identificação automática de instruções complexas
    - MPSoC para Ray-Tracing e Volume Ray-Casting com instruções complexas
  - Síntese de Circuitos
    - Compartilhamento de Hardware
  - Arquiteturas e Modelos Dataflow
    - JSucuri - Versão java da Sucuri (A Minimalistic Dataflow Programming Library for Python)

Motivação

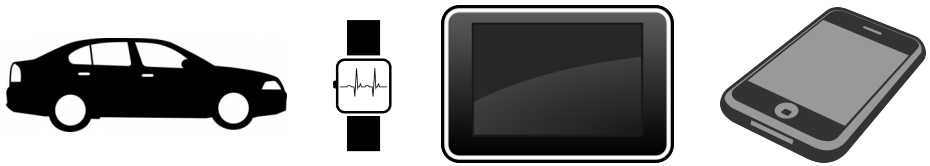
- Nos últimos anos:
  - Processadores de Propósito Geral → Processadores de Propósito Específico
    - Application-Specific Instruction Set Processors (ASIPs)
  - Sistemas Embarcados: carros, smartphones, tablets, câmeras, etc.



- Projetar e desenvolver ASIPs costuma ser demorado e caro
- Automatizar etapas do desenvolvimento pode ajudar a:
  - Reduzir o tempo de produção (*Time-to-market*)
  - Aumentar a qualidade e eficiência do ASIP
- Uma destas etapas é a Extensão do Conjunto de Instruções

Motivação

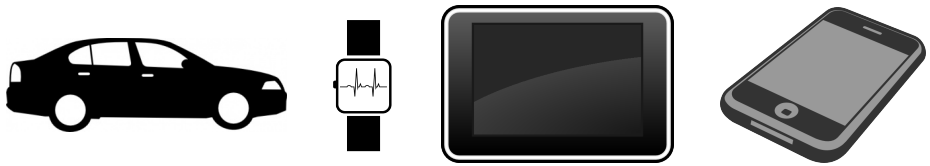
- Nos últimos anos:
  - Processadores de Propósito Geral → Processadores de Propósito Específico
    - Application-Specific Instruction Set Processors (ASIPs)
  - Sistemas Embarcados: carros, smartphones, tablets, câmeras, etc.



- Projetar e desenvolver ASIPs costuma ser demorado e caro
- Automatizar etapas do desenvolvimento pode ajudar a:
  - Reduzir o tempo de produção (*Time-to-market*)
  - Aumentar a qualidade e eficiência do ASIP
- Uma destas etapas é a Extensão do Conjunto de Instruções

Motivação

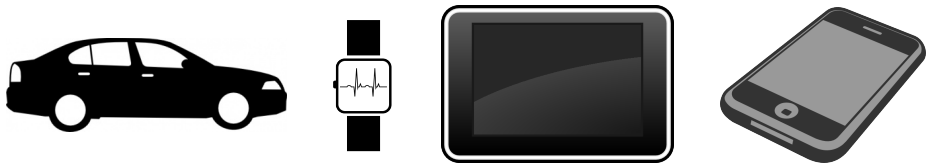
- Nos últimos anos:
  - Processadores de Propósito Geral → Processadores de Propósito Específico
    - Application-Specific Instruction Set Processors (ASIPs)
  - Sistemas Embarcados: carros, smartphones, tablets, câmeras, etc.



- Projetar e desenvolver ASIPs costuma ser demorado e caro
- Automatizar etapas do desenvolvimento pode ajudar a:
  - Reduzir o tempo de produção (*Time-to-market*)
  - Aumentar a qualidade e eficiência do ASIP
- Uma destas etapas é a Extensão do Conjunto de Instruções

Motivação

- Nos últimos anos:
  - Processadores de Propósito Geral → Processadores de Propósito Específico
    - Application-Specific Instruction Set Processors (ASIPs)
  - Sistemas Embarcados: carros, smartphones, tablets, câmeras, etc.



- Projetar e desenvolver ASIPs costuma ser demorado e caro
- Automatizar etapas do desenvolvimento pode ajudar a:
  - Reduzir o tempo de produção (*Time-to-market*)
  - Aumentar a qualidade e eficiência do ASIP
- Uma destas etapas é a Extensão do Conjunto de Instruções

Extensão do Conjunto de Instruções

- Incluir no caminho de dados do processador unidades funcionais especializadas: **instruções complexas**
- Trade-off: área do circuito × atraso do caminho crítico × consumo de energia
- Exemplo clássico: multiply-add (ou multiply-accumulate)
  - Operação muito usada em aplicações de computação gráfica
  - Muito comum em GPUs

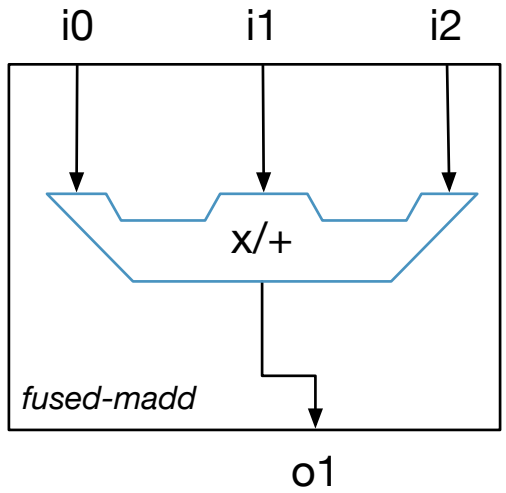


Figure: Unidade funcional Fused-multiply-add.

Identificação Automática de Instruções Complexas

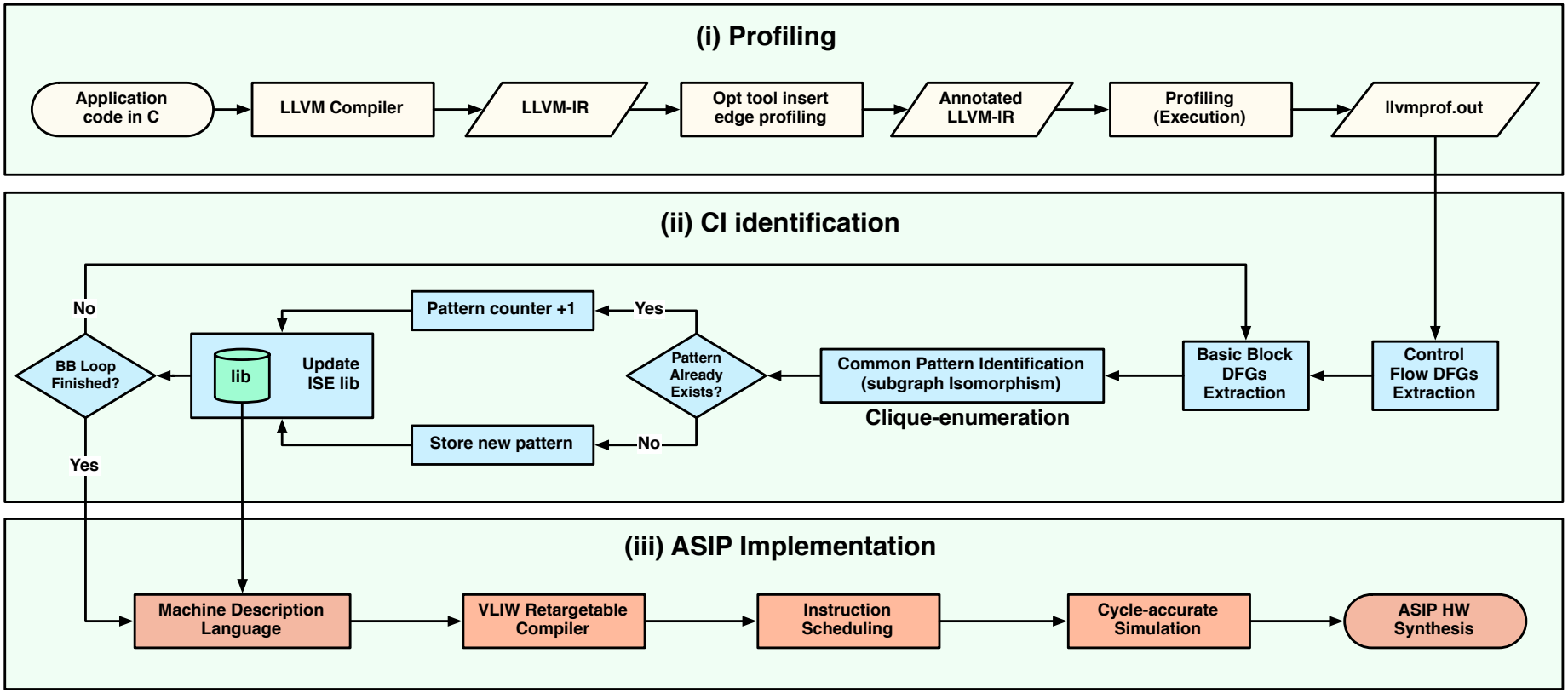
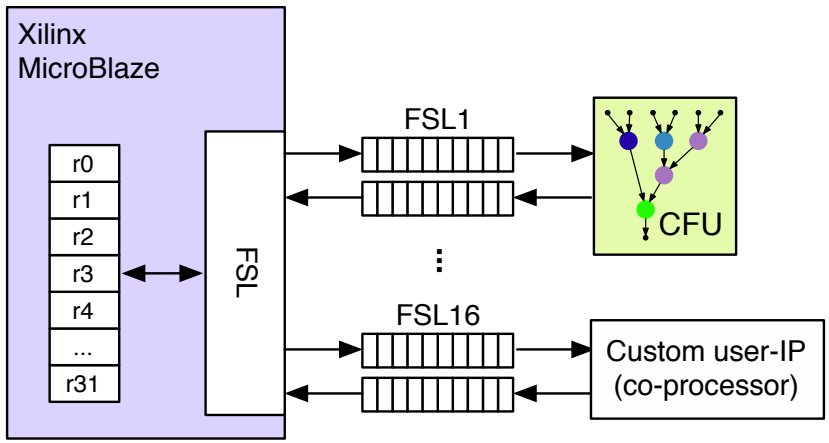


Figure: Dinâmica da Identificação Automática de Instruções Complexas.

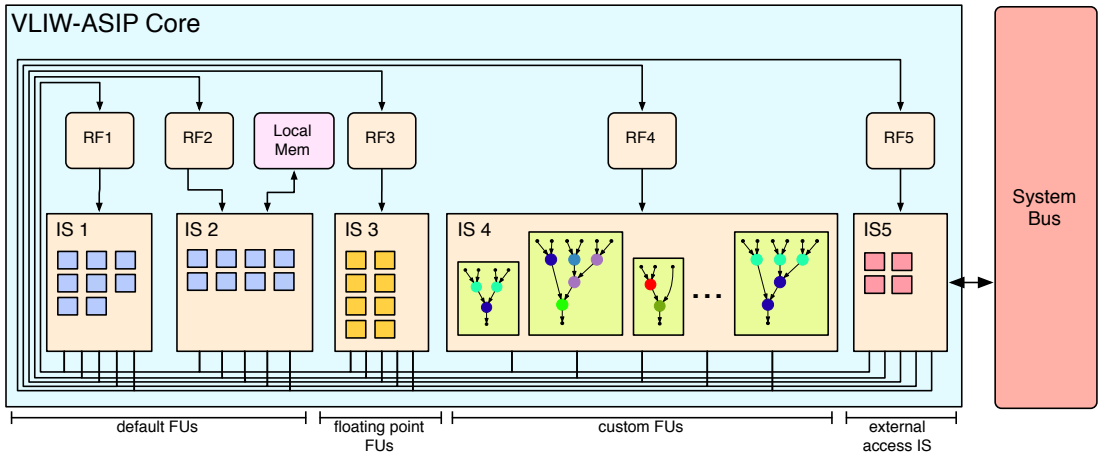
Identificação Automática de Instruções Complexas

1. Profiling
   - Etapa para "conhecer" a aplicação
   - Inclusão de código de instrumentação para *profiling* de Blocos Básicos
   - Produz DFGs dos Blocos Básicos mais frequentemente executados
2. Identificação de instruções complexas
   - Comparação dos grafos dataflow de cada bloco básico
   - Isomorfismo de subgrafos (via enumeração de cliques máximas)
3. Implementação do ASIP
   - Inclusão das unidades funcionais das instruções complexas no caminho de dados
   - Execução da aplicação no processador reconfigurado com as instruções complexas



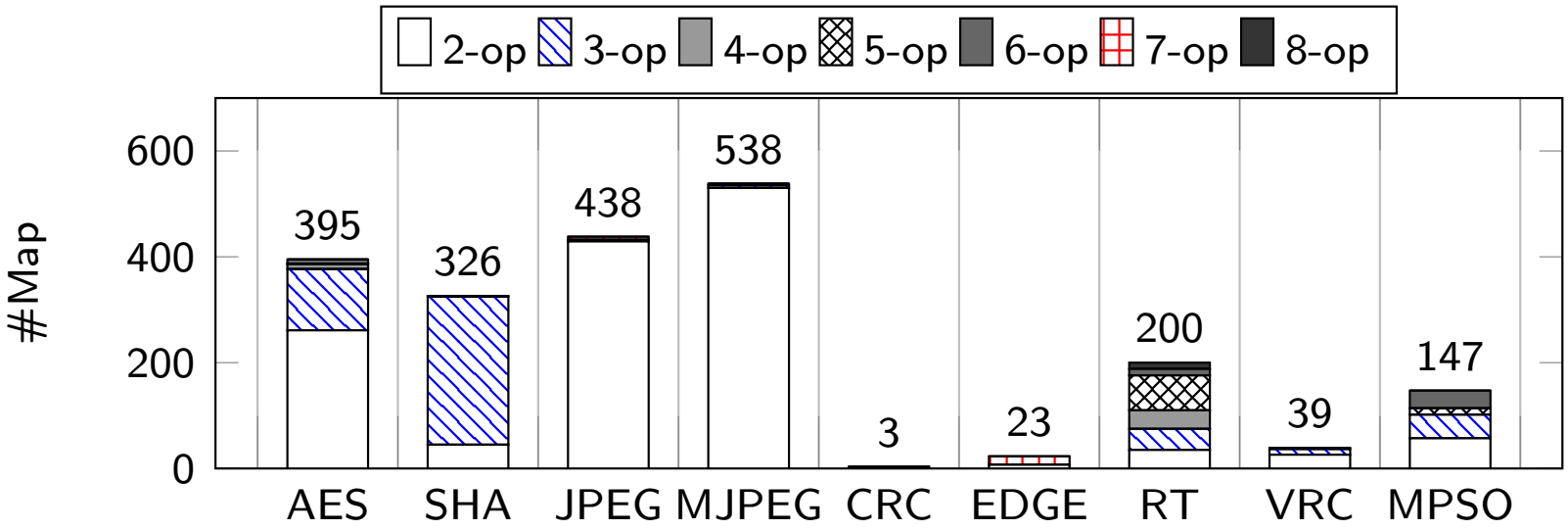
Identificação Automática de Instruções Complexas

1. Profiling
   - Etapa para "conhecer" a aplicação
   - Inclusão de código de instrumentação para *profiling* de Blocos Básicos
   - Produz DFGs dos Blocos Básicos mais frequentemente executados
2. Identificação de instruções complexas
   - Comparação dos grafos dataflow de cada bloco básico
   - Isomorfismo de subgrafos (via enumeração de cliques máximas)
3. Implementação do ASIP
   - Inclusão das unidades funcionais das instruções complexas no caminho de dados
   - Execução da aplicação no processador reconfigurado com as instruções complexas



Resultados Interessantes

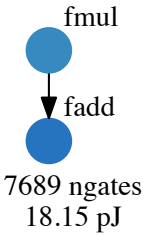
- Total de 70 instruções complexas foram identificadas e mapeadas (para todas as aplicações)
  - aplicações: criptografia, processamento de imagens, renderização, otimização, etc.



Resultados Interessantes

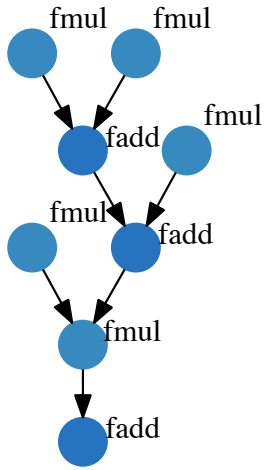
- Instruções de granularidade fina (menos operações) foram mapeadas (usadas) com mais frequência
- Ou seja, instruções complexas menores são re-utilizadas com mais frequência
  - "Encaixam com mais facilidade"

multiply-add:



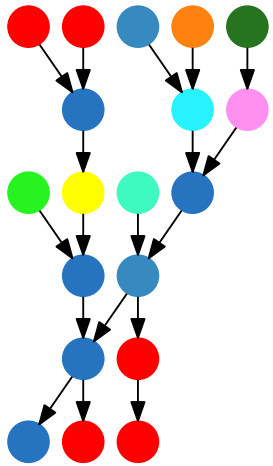
7689 ngates
18.15 pJ

dot-product:

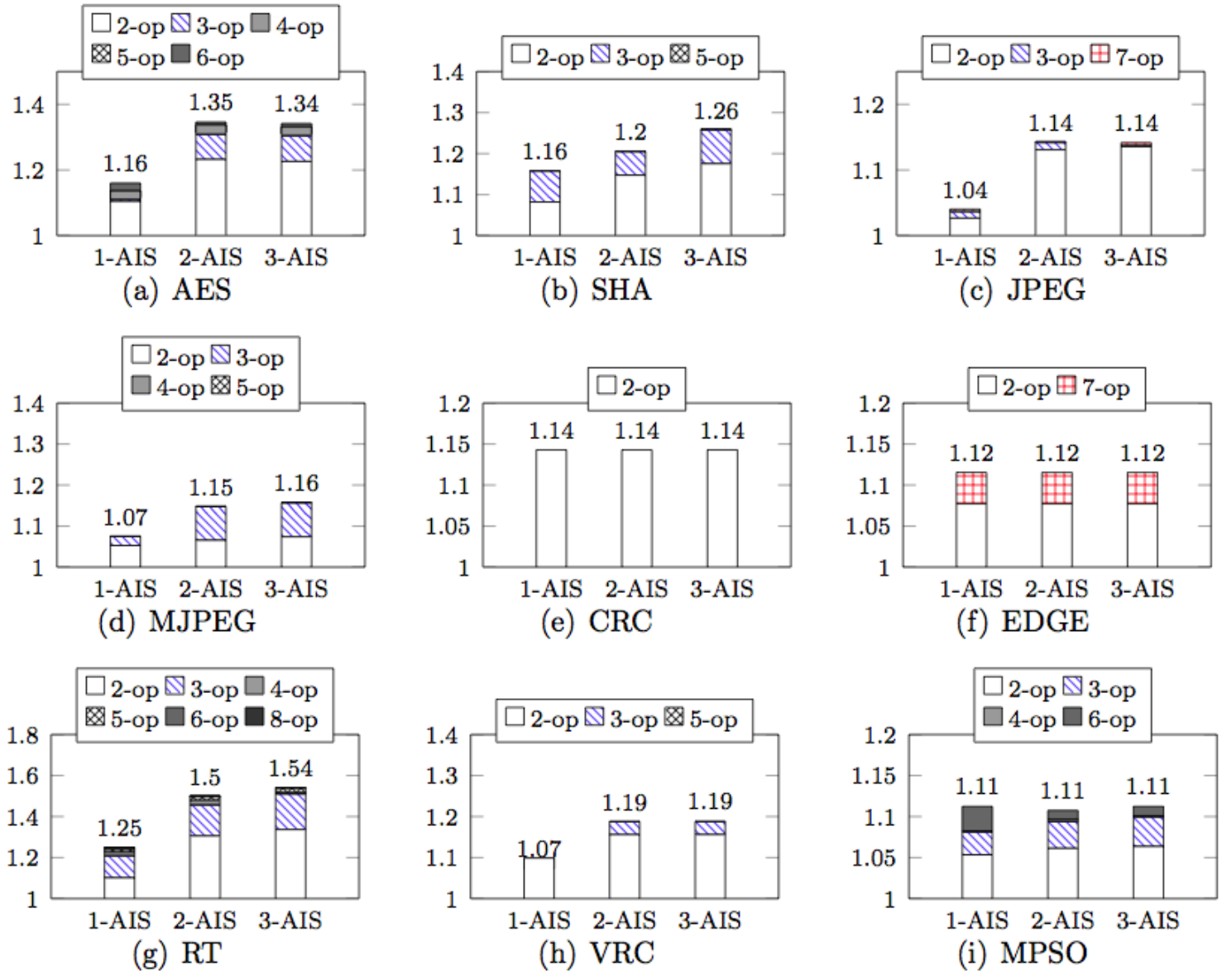


32721 ngates
79.51 pJ

trecho programa:



Logo, as instruções complexas de grão-fino contribuíram mais para acelerar a execução das aplicações



Resultados Interessantes

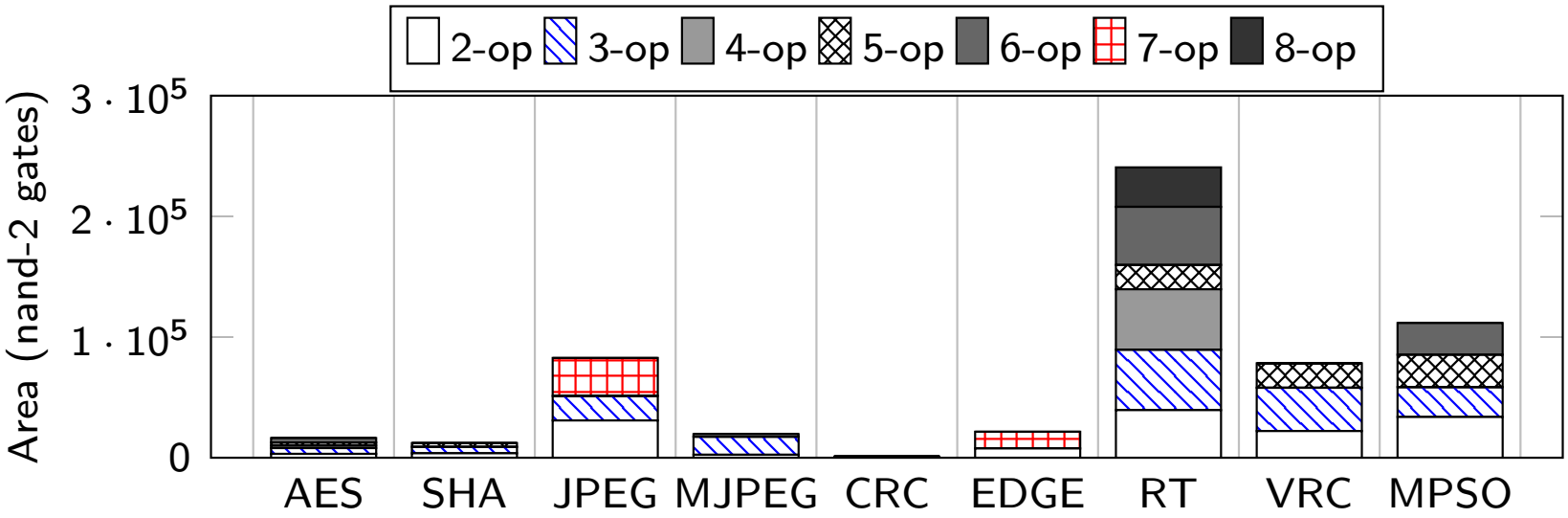


Figure: Estimativa de consumo de área das instruções complexas para diferentes granularidades. TSMC 65nm @200MHz.

Resultados Interessantes

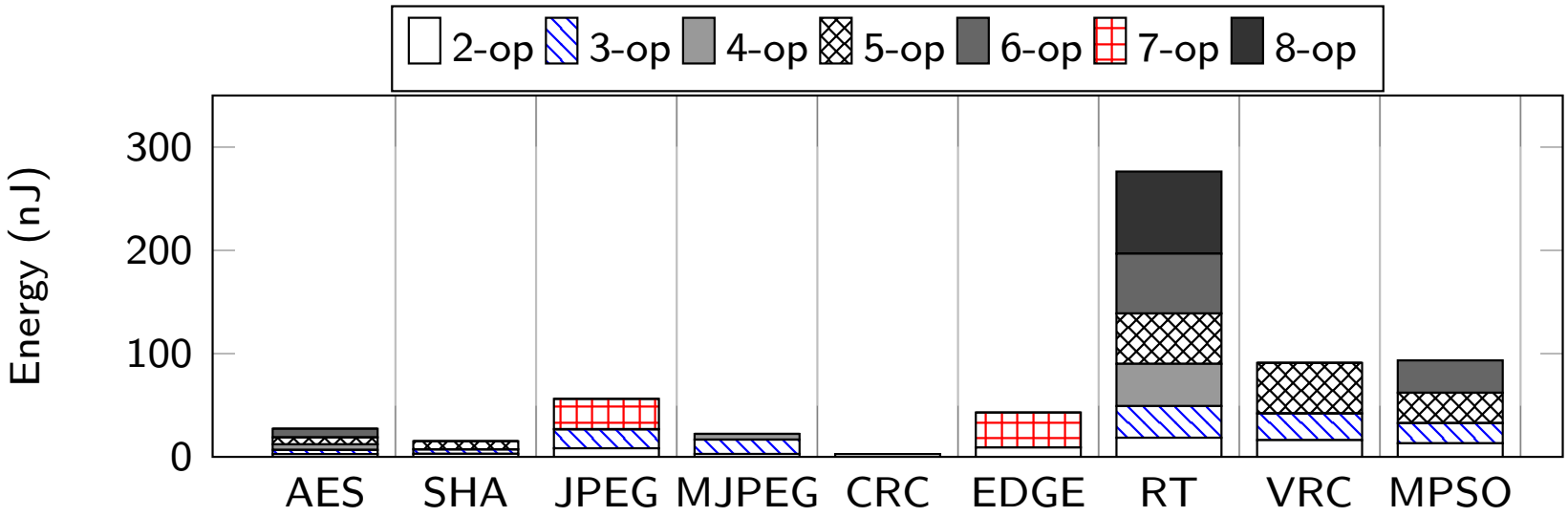


Figure: Estimativa de consumo de energia das instruções complexas para diferentes granularidades. TSMC 65nm @200MHz.

Aplicações

MPSoCs reconfiguráveis baseado em ASIPs:

- Paralelização do algoritmo de Ray-Tracing
  - Distribuir os raios entre os processadores
- Projeto & Implementação de dois MPSoCs com instruções complexas
- Instruções complexas identificadas automaticamente
  - Baseado em RISC:
    - Até 5 microprocessadores Microblaze 125MHz cada
    - Cada um com até 3 instruções complexas (limitação da FPGA)
    - Xilinx Virtex-5 FPGA
  - Baseado em VLIW:
    - Até 8 microprocessadores Intel VLIW
    - Cada um com até 14 instruções complexas
    - Solução ASIC: TSMC 65nm technology 200MHz

Aplicações – Baseado em RISC

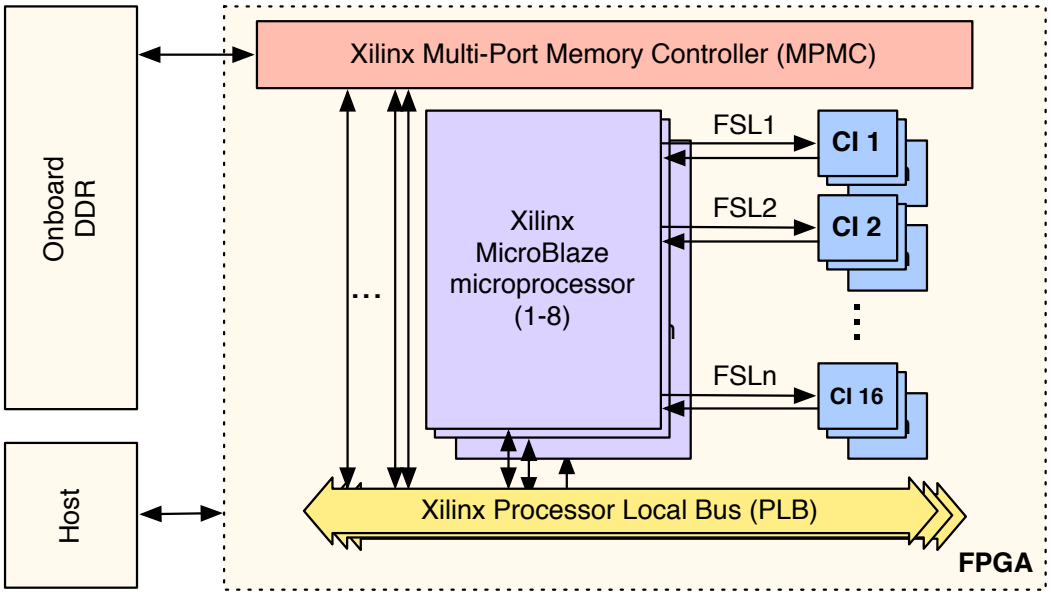


Figure: RT Reconfigurable MPSoC.

1. Processamento paralelo de raios
2. Com instruções complexas
3. Escalável*
4. FSL (comunicação ponto-a-ponto de baixa latência)
5. Imagem de 320×240 (pouca memória)

Escalabilidade*

Mais de 8 microprocessadores requer um novo controlador de memória

Aplicações – Baseado em VLIW

- Processadores VLIW com instruções complexas
  - Em paralelo

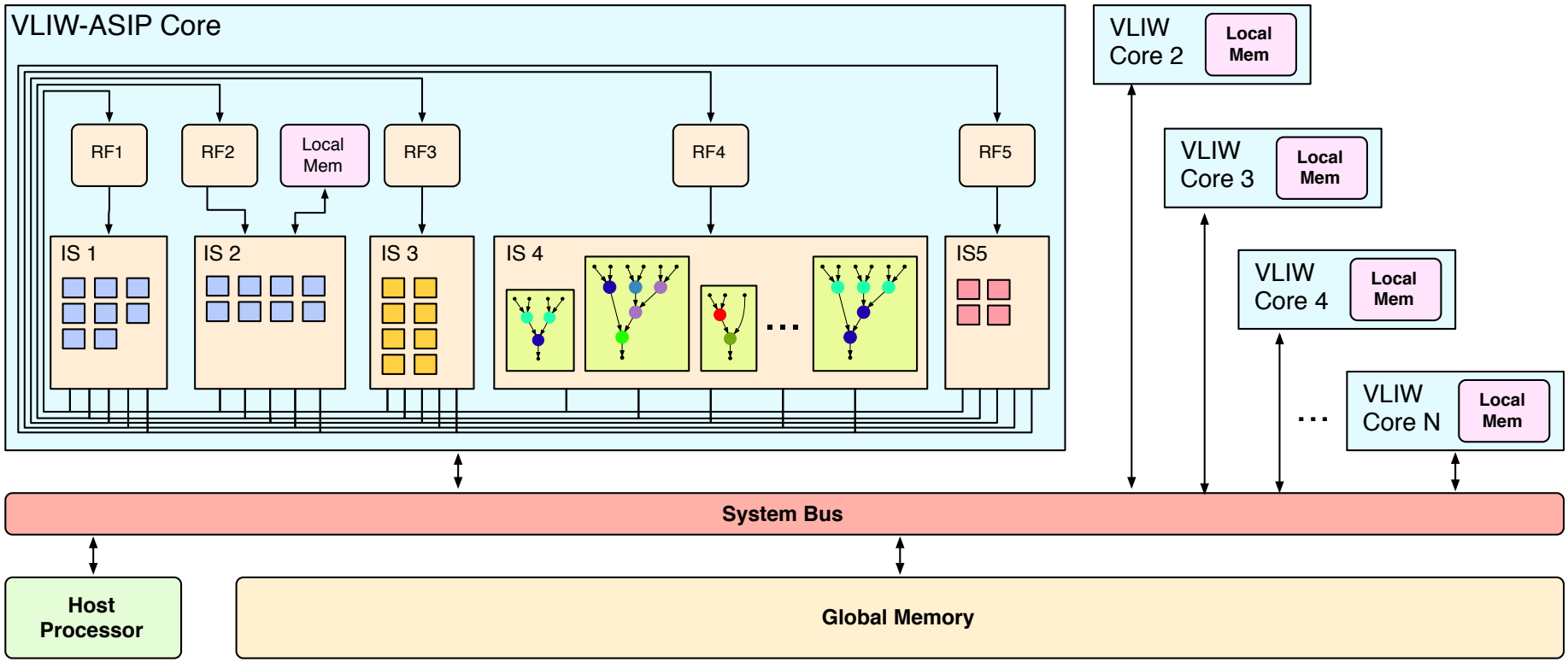
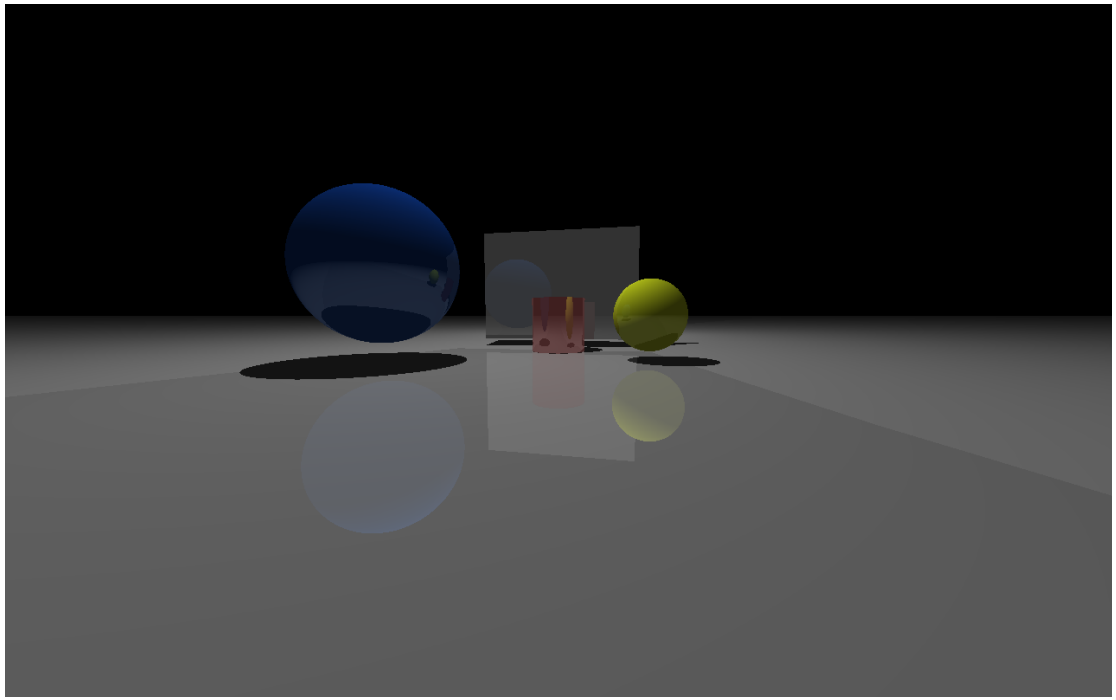


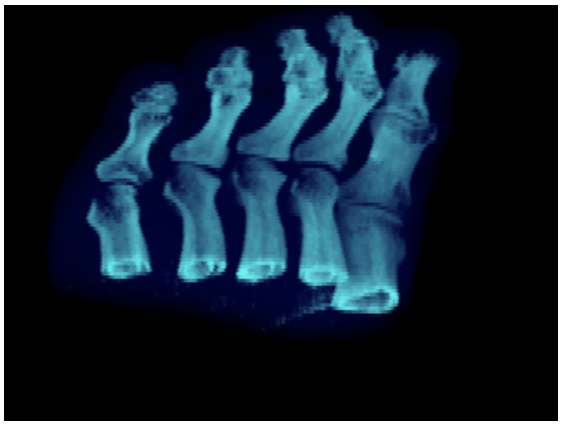
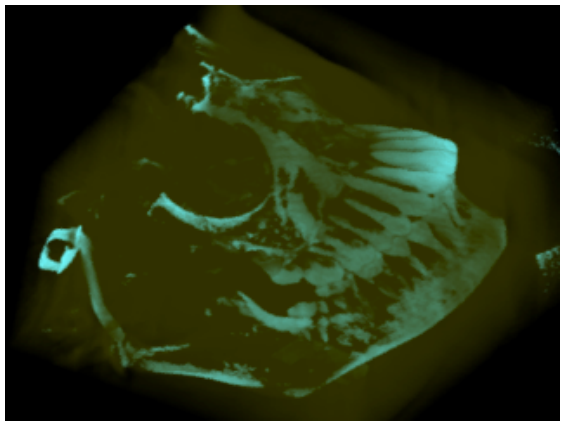
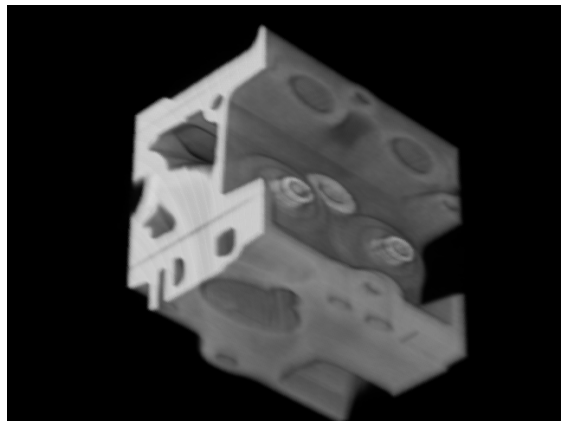
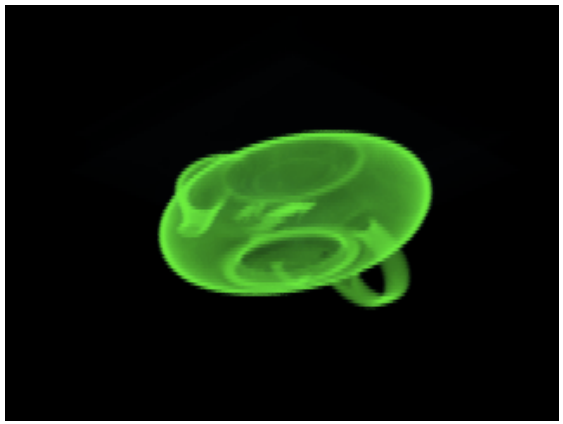
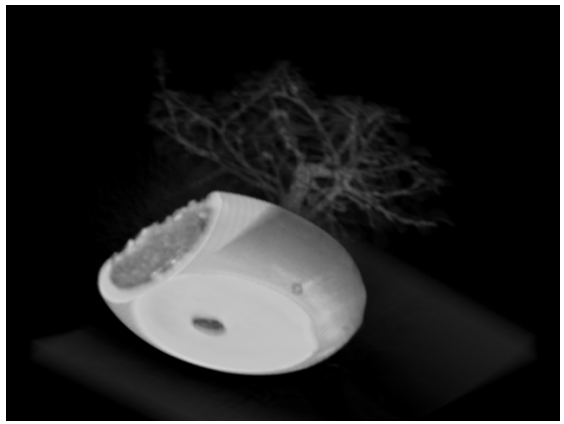
Figure: A arquitetura VLIW Multi-core com instruções complexas.

Saída Ray-Tracing



- 8 microprocessadores VLIW @200MHz, 1280x800, com instruções complexas: "tempo real" (2fps)
- 5 microprocessadores RISC @125MHz, 800x600, com instruções complexas: 1 frame a cada 7.8s

Saída Volume Ray-Casting

- 4 microprocessadores RISC @125MHz, sem instruções complexas: 1 frame / 40s.

Considerações finais

- Conjuntos de métodos e ferramentas para extensão do conjunto de instruções
  - Menor tempo de desenvolvimento
  - Permite o desenvolvimento de processadores mais eficientes
  - Conjunto de 70 instruções complexas para diferentes aplicações
  - Aceleração de até 54% (Ray-Tracing)
  - Instruções "baratas" para AES, SHA, MJPEG, CRC, EDGE
  - Instruções "caras" para aplicações que usam Ponto Flutuante
  - Instruções de grão-fino contribuem em grande parte para o aumento do desempenho
- Arquiteturas MPSoC
  - Aceleração quase linear dos algoritmos Ray-Tracing e Volume Ray-Casting
  - Imagens de alta resolução
  - Instruções complexas contribuíram para melhorar o desempenho
    - RISC-based: por volta de 7%
    - VLIW-based: por volta de 50%

Fim...

Obrigado! Perguntas?

Contato:

anery@ime.uerj.br

alexandre.solon@gmail.com