

HRA-MPSoC: Arquitetura Hierarquicamente Reconfigurável de Sistemas Multiprocessados em Chip

Alexandre Marques Amaral e Carlos Augusto Paiva da Silva Martins
Grupo de Sistemas Digitais e Computacionais (GSDC)
Pontifícia Universidade Católica de Minas Gerais
alexmarques@ieee.org, capsam@pucminas.br

Resumo

Nos últimos anos, os sistemas computacionais embutidos (SCE) têm evoluído bastante. Além disso, a diversidade de requisitos das aplicações estimula o projeto e desenvolvimento de uma grande variedade de SCEs. Entretanto, os SCEs convencionais normalmente não suportam adequação, considerando as mudanças dinâmicas da carga de trabalho e ou dos requisitos. O objetivo deste artigo é apresentar uma arquitetura hierarquicamente reconfigurável de sistemas multiprocessados em chip (HRA-MPSoC). A HRA-MPSoC é reconfigurável, em suas diferentes camadas, para adequar a arquitetura e ou a implementação dos SCEs às variações dos requisitos e da carga de trabalho da aplicação. Resultados de desempenho, flexibilidade e escalabilidade verificam as vantagens da HRA-MPSoC no projeto de SoCs e MPSoCs, caracterizando-a como a principal contribuição.

1. Introdução

Nos últimos anos, a utilização de sistemas computacionais embutidos (SCEs) tem aumentado bastante, não restringindo mais a simples controladores [1]. Conseqüentemente, os SCEs têm evoluído tanto em suas arquiteturas como nas implementações [2]. A partir dessa evolução surgiu a necessidade do desenvolvimento de sistemas complexos com requisitos funcionais de execução de funções complexas e requisitos não funcionais críticos, como consumo de potência, desempenho, dentre outros. Com isso, muitos SCEs passaram a ser desenvolvidos como sistemas em chip (SoCs) [1]. Atualmente, aplicações com requisitos mais críticos e novas tecnologias de projeto e fabricação de circuitos integrados, incentivam o desenvolvimento de sistemas multiprocessados em chip (MPSoCs) para aplicação em SCEs.

Considerando as diferentes aplicações de SCEs, existe uma grande diversidade de requisitos que estimulam o projeto e desenvolvimento de uma grande variedade de SCEs [2]. Os diferentes tipos de projetos de SCEs podem ser representados como elementos no espaço de projeto. Em um extremo deste espaço existem requisitos de execução de operações simples e diminuição do consumo de potência, com restrições de mínimo desempenho computacional permitido (ex. um sistema em chip – SoC – utilizado em aplicações móveis). Em outro extremo do espaço de projeto, existem requisitos de execução de operações complexas e aumento do desempenho computacional, com restrições de máximo consumo de potência permitido (ex. um sistema multiprocessado em chip – MPSoC – para aplicações em tempo real).

Além da grande diversidade de requisitos, existem aplicações que podem variar sua carga de trabalho para atender novos requisitos. Estes requisitos podem ser funcionais ou não funcionais. Entretanto, os SCEs convencionais, baseados em microcontroladores, normalmente utilizam soluções computacionais que não permitem a adequação da arquitetura e da sua implementação considerando as mudanças dinâmicas da carga de trabalho e ou dos seus requisitos. Com isso, quando a carga de trabalho da aplicação é modificada, os SCEs convencionais podem não atender a um sub-conjunto dos requisitos não funcionais da aplicação, como desempenho, flexibilidade, tolerância a falhas, dentre outros. Portanto, a falta de adequação das arquiteturas e ou implementações à variedade das aplicações, variações da carga de trabalho e ou dos requisitos é um problema dos SCEs convencionais.

Considerando a ausência de adequação dos SCEs, a meta desta pesquisa é propor uma arquitetura hierarquicamente reconfigurável de sistemas multiprocessados em chip (*Hierarchical-Reconfigurable-Architecture for MultiProcessor-System-on-Chip* – HRA-MPSoC). Esta arquitetura é reconfigurável nas suas camadas hierárquicas

estruturais, permitindo a reconfiguração das arquiteturas e implementações de SoCs e MPSoCs para atender diferentes requisitos de diferentes aplicações.

O objetivo deste artigo é apresentar a HRA-MPSoC e algumas implementações de SoCs e MPSoCs para verificar as vantagens da utilização desta arquitetura no projeto, desenvolvimento e implementação de sistemas em chip.

O artigo está organizado da seguinte maneira: a seção 2 apresenta uma análise dos principais trabalhos relacionados; a seção 3 apresenta a HRA-MPSoC, suas características e as implementações desenvolvidas para a sua verificação; a seção 4 apresenta os resultados de verificação, seguido da análise e discussão; na seção 5, as conclusões, contribuições e trabalhos futuros.

2. Trabalhos Relacionados

Alguns trabalhos têm sido desenvolvidos utilizando SoCs e MPSoCs em SCEs [3,4]. Outros trabalhos aplicam reconfiguração em SoCs e MPSoCs [5,6]. Um terceiro grupo de trabalhos aplicam reconfiguração em SoCs e MPSoCs para aplicações em SCEs [7,8]. Estes últimos são mais relacionados com esta pesquisa.

Herkersdorf e Stechele [7] apresentam uma Unidade Eletrônica de Controle (ECU) aplicada em automóveis, para executar tarefas de autodireção a partir de processamento de imagens e tomada de decisões. Essa função possui alta demanda computacional para o processamento das imagens detectadas por múltiplos sensores. Portanto, para atender aos requisitos de desempenho e custo, os autores desenvolveram um sistema multiprocessado reconfigurável em chip, proposto para ser o núcleo de processamento. Neste MPSoC, os módulos de processamento de imagens são reconfigurados dinamicamente no FPGA. A implementação desses módulos em hardware garante a resposta em tempo real. Além disso, a reconfiguração possibilita a multiplexação no tempo dos módulos de cada etapa do processamento (ex. MPEG-7), reduzindo a ocupação de recursos e os custos de implementação.

Mignolet e outros [8] apresentam uma plataforma de computação reconfigurável para aplicações multimídia de aparelhos portáteis. Apresenta-se também um sistema multitarefa utilizando esta plataforma. A característica de reconfiguração da plataforma foi desenvolvida para atender diferentes requisitos das aplicações. Resultados funcionais do sistema e alguns resultados de desempenho são apresentados.

Ressalta-se que não foram encontrados trabalhos que apresentem arquiteturas reconfiguráveis hierarquicamente. Os trabalhos citados nesta seção apresentam SoCs e MPSoCs reconfiguráveis, não possuindo qualquer tipo de contribuição arquitetural com reconfiguração nas camadas hierárquicas nas fases

de projeto e implementação de SoCs e MPSoCs fixos ou reconfiguráveis, ao contrário deste artigo.

3. HRA-MPSoC

A Arquitetura Hierarquicamente Reconfigurável de Sistemas Multiprocessados em Chip (HRA-MPSoC) foi desenvolvida no contexto de uma pesquisa em arquiteturas hierarquicamente reconfiguráveis de sistemas computacionais (HRA-CS). A pesquisa HRA-CS tem como objetivo desenvolver arquiteturas sistêmicas e hierárquicas de sistemas computacionais, que sejam reconfiguráveis nas suas diferentes camadas hierárquicas estruturais [9]. A HRA-MPSoC explora as camadas inferiores da HRA-CS que sejam implementadas em um único chip.

3.1. Arquitetura

Como apresentado, existem muitos problemas gerados pela falta de adequação dos SCEs, de acordo com as mudanças dinâmicas dos requisitos ou da carga de trabalho das aplicações. Esta falta de adequação se deve às arquiteturas e implementações fixas dos SCEs.

Considerando esses problemas, a arquitetura HRA-MPSoC foi proposta, projetada e desenvolvida para permitir implementações de SoCs e MPSoCs que suportem modificações tanto nos requisitos da carga de trabalho das aplicações como mudança de carga de trabalho. Além disso, essas modificações devem ocorrer sem perda do atendimento aos requisitos não modificados. Para isso, a HRA-MPSoC utiliza conceitos de reconfiguração para adequar a arquitetura e ou a implementação do SCE às diferentes cargas de trabalho das aplicações e atender aos requisitos.

A figura 1 apresenta o diagrama em blocos da camada sistêmica (camada 1), a representação da camada de elementos (camada 2) e da camada de componentes (camada 3) da HRA-MPSoC. A camada 1 possui os seguintes blocos funcionais: Elemento de Processamento (EP), Elemento de Armazenamento (EA), Elemento de Interconexão (EI), Interface de Entrada (IE) e Interface de Saída (IS).

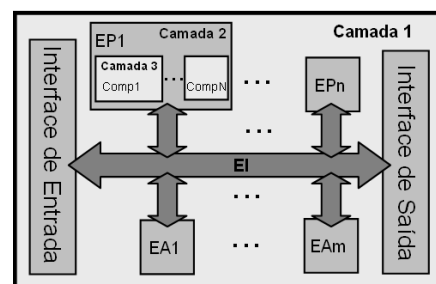


Figura 1. Diagrama da HRA-MPSoC.

Os EPs são responsáveis por todo o processamento a ser realizado no interior do SCE. Esse processamento pode ser paralelo, com a divisão da carga de trabalho da aplicação. Dependendo da aplicação, ela pode ser executada com um mesmo fluxo de instruções e mais de um fluxo de dados entre os EPs ou com múltiplos fluxos de instruções e de dados entre os EPs.

Os EAs podem implementar diferentes modelos de armazenamento, como: hierarquia de memória, distribuídas ou compartilhadas entre os EPs. Os blocos de Interface de Entrada (IE) e Interface de Saída (IS) estabelecem a comunicação dos elementos no interior do chip com o exterior. O Elemento de Interconexão (EI) realiza a comunicação dos EPs, EAs, IE e IS.

A reconfiguração nas diferentes camadas hierárquicas da HRA-MPSoC facilita a adequação às cargas de trabalho das aplicações. A reconfiguração na camada sistêmica (camada 1) permite que a arquitetura seja implementada como um circuito específico para aplicação em chip (ASIC), um processador convencional de propósito geral ou aplicação específica (GPP ou ASIP), um sistema monoprocessoado em chip (SoC) ou como um sistema multiprocessoado em chip (MPSoC). Portanto a reconfiguração na camada 1 permite a adequação sistêmica de acordo com as características gerais da aplicação, variando a organização dos elementos EP, EA, EI, IE e IS. A reconfiguração na camada 2 permite que a arquitetura de cada elemento seja adequada de acordo com o subconjunto de operações executado naquele elemento. A reconfiguração na camada 3 permite a adequação dos componentes dos elementos. Os blocos EA, EI, IE, IS não são abordados com detalhes neste artigo, pois o seu escopo é a apresentação da HRA-MPSoC utilizada no desenvolvimento de SoCs e MPSoCs com reconfiguração nas camadas 2 e 3 dos EPs.

Além das camadas 1, 2 e 3 de reconfiguração apresentadas na figura 1, podem existir outras camadas no interior dos elementos (camadas 4, 5, etc.). Destaca-se que a reconfiguração em uma camada superior (ex. camada 1) pode ou não implicar a reconfiguração das camadas inferiores (camada 2, 3, etc.). A quantidade de camadas é um parâmetro da HRA-MPSoC e pode ser reconfigurado de acordo com a estruturação da arquitetura para executar uma aplicação específica. Outros parâmetros são: a quantidade de elementos reconfiguráveis ou fixos na camada 1 e a largura de bits dos elementos (camada 2) ou componentes (camada 3).

A reconfiguração na camada dos elementos de processamento (EPs) (camada 2) permite que os EPs sejam implementados como diferentes classes, variando as suas características de programabilidade e flexibilidade, como apresentado na figura 2. Segundo a característica de programabilidade, os elementos são classificados como EP programável ou não-

programável através de software (EP-P e EP-NP). Segundo a característica de flexibilidade, os elementos são classificados como EP fixo (EP-P-F e EP-NP-F) e EP não-fixo (EP-P-NF e EP-NP-NF). Detalhes dessa classificação estão apresentados em Amaral e outros [9]. No caso da HRA-CS e da HRA-MPSoC, é utilizada a reconfiguração para implementar o atributo *não-fixo* da característica de flexibilidade. Destaca-se que as características de flexibilidade (*F* e *NF*) e programabilidade (*P* e *NP*) são independentes.

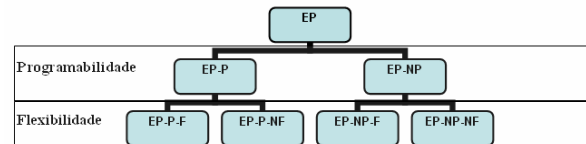


Figura 2. Diagrama de classificação dos EPs.

As diferentes classes de EPs e a possibilidade de reconfiguração nas camadas hierárquicas, permitem que a HRA-MPSoC seja totalmente adequada (reconfigurada) para executar uma determinada aplicação. Ou seja, a quantidade de EPs do sistema e a maneira de comunicação podem ser modificados através da reconfiguração na camada 1. Além disso, cada EP pode ser adequado para um subconjunto de operações da aplicação executada, através da reconfiguração das camadas internas (camadas 2, 3, etc.). Portanto, a reconfiguração hierárquica possibilita o maior desempenho possível da implementação.

Suponha que uma aplicação possui uma carga de trabalho heterogênea, sendo que parte dela é composta por operações complexas e a outra parte é composta por muitas operações simples. As operações complexas poderiam ser implementadas em EP-Ps e as simples em EP-NPs. Esta arquitetura heterogênea (na camada 1) poderia apresentar ganhos de desempenho bastante significativos quando comparada a arquiteturas homogêneas com todos os EPs configurados como EP-NP ou EP-P. Esse ganho ocorre devido à adequação da arquitetura do EP ao modelo de execução de cada subconjunto de operações da carga de trabalho.

No exemplo anterior, suponha que a carga de trabalho da aplicação é modificada, de maneira que ela possua muitas operações aritméticas simples, porém com alto grau de paralelismo entre elas, ex. Convolução de Imagens. Neste caso, a HRA-MPSoC possibilita que um sistema com múltiplos EP-NPs idênticos seja implementado para executar a aplicação com o máximo grau de paralelismo possível. A HRA-MPSoC não possui limites de grau máximo de paralelismo. Este limite é determinado pela aplicação, ou pelas restrições tecnológicas da implementação ou por restrições de projeto (ex. custo). Esta arquitetura paralela homogênea (na camada 1) poderia apresentar

ganhos de desempenho quando comparada a arquiteturas heterogêneas, devido à melhor adequação ao modelo de execução do subconjunto de operações e máxima exploração do paralelismo.

Destaca-se que os MPSoCs podem ser homogêneos ou heterogêneos nas suas diversas camadas hierárquicas [10]. Por exemplo, a homogeneidade na camada 1 implica que todos os EPs possuem comportamentos idênticos. A homogeneidade na camada 2 implica que todos os EPs possuem estruturas e comportamentos idênticos, ex. nas suas arquiteturas ISA, microarquiteturas e nas características de funcionamento. Caso contrário, considerando a camada analisada, o sistema é caracterizado como um MPSoC heterogêneo. Generalizando, a HRA-MPSoC permite a implementação de MPSoCs homogêneos ou heterogêneos, em suas diversas camadas hierárquicas.

Portanto, com a característica de reconfiguração hierárquica, a HRA-MPSoC é totalmente flexível e escalável. Essa característica permite que ela seja reconfigurada para atender a diferentes tipos de cargas de trabalho, atendendo aos requisitos funcionais e não funcionais de diferentes aplicações.

3.2. Estudos de Caso

Na verificação da HRA-MPSoC, diferentes SoCs e MPSoCs (reconfiguração na camada 1) foram desenvolvidos para dois tipos de aplicações bastante distintas. Essas aplicações são de: controle digital e filtragem de imagens digitais. O EP gerenciador tanto dos SoCs como dos MPSoCs desenvolvidos é um EP-P-F chamado *Intellectual-Property Processor Component* (IPPC) [11,12]. O EP que implementa as operações do controlador digital é um EP-NP-F chamado *Reconfigurable Electronic Control Unit* (RECU)¹. O EP que implementa as operações do filtro é um EP-NP-F chamado *Image Convolution Component* (ICC) [13,14].

A organização de SoC projetada é composta por uma única instância de IPPC e mais de uma instância de EP-NP-F (ICC ou RECU). A organização de MPSoC projetada possui múltiplas instâncias de IPPC e de EP-NP-F (ICC ou RECU). A reconfiguração na camada 1 da HRA-MPSoC possibilita a variação do número de EP-P-Fs e EP-NP-Fs, suportando implementações de diferentes SoCs e MPSoCs. A reconfiguração na camada 2 permite a implementação de IPPCs, RECU ou ICCs, com diferentes quantidades

de componentes. A reconfiguração na camada 3 possibilita a modificação da arquitetura interna dos componentes (ex. Unidade Lógica Aritmética dos IPPCs). Portanto, os SoCs e MPSoCs projetados e implementados comprovam as vantagens da reconfiguração hierárquica da HRA-MPSoC, através da parametrização, da flexibilidade e da escalabilidade no desenvolvimento e funcionamento dos SCEs.

A figura 3 apresenta o diagrama de blocos parametrizado do MPSoC projetado utilizando ICC ou RECU, representando a organização de um controlador multiprocessado para controle paralelo de diferentes sistemas ou de um filtro multiprocessado para filtragem paralela de imagens. Os SoCs e MPSoCs desenvolvidos foram classificados pelas quantidades de IPPCs e de ICCs ou RECU (n-IPPC_m-EP-NP-F). Nesta classificação, n é o número de IPPCs e m é o número de ICCs ou RECU. Nos SoCs, $n=1$.

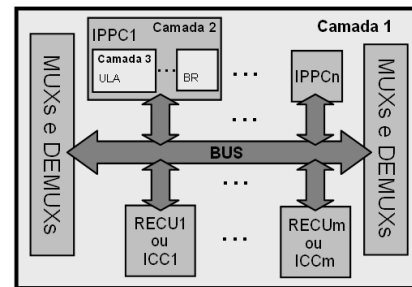


Figura 3. Organização dos MPSoCs.

As instâncias do IPPC podem funcionar como fonte de dados ou de comandos para os EPs aceleradores de aplicação ou como elementos de processamento executando paralelamente parte da aplicação em software. Nas implementações do filtro as instâncias do IPPC funcionam como fontes de comandos, enquanto nas implementações do controlador elas funcionam como elementos de processamento. Além disso, as versões atuais das implementações foram projetadas e desenvolvidas utilizando elemento de interconexão (EI), interfaces de entrada e de saída (IE e IS) compostos por MUXs e DEMUXs. Entretanto, um trabalho em desenvolvimento é a implementação do EI utilizando uma chave *crossbar* reconfigurável [15].

No caso do filtro de imagens, o ICC foi integrado a uma interface de manipulação de pixels para suportar acesso à *buffers* de pixels. Esses *buffers* armazenam os próximos pixels a serem processados. Além disso, essas interfaces permitem acesso à memória onde a imagem está armazenada, com manipulação de palavras de $k \times 8$ bits, sendo k é um parâmetro da arquitetura. Portanto, com pixels de resolução de 8 bits, essa interface permite ao ICC receber e enviar até k pixels a cada ciclo de clock. Apesar de sua arquitetura permitir a vazão de um único pixel de saída, a interface melhora a

¹ O projeto RECU (*Reconfigurable Electronic Control Unit*) está sendo desenvolvido pelo Grupo de Sistemas Digitais e Computacionais e ganhou o segundo lugar no *Xilinx Student Contest*, ocorrido no *Chip on the Island 2005*, promovido pela Sociedade Brasileira de Computação e Sociedade Brasileira de Microeletrônica.

vazão na entrada de pixels, quando o acesso a mais de um pixel, que não esteja nos *buffers*, for necessário.

Utilizando o ICC integrado à interface de manipulação de pixels, foram projetados nove configurações de SoCs e MPSoCs. Três configurações possuem restrição de acoplamento de dois IPPCs para cada ICC, sem adequação da interface entre IPPC e ICC. As outras seis configurações foram projetadas com o desacoplamento da quantidade de ICCs e IPPCs, sendo que a quantidade de ICCs é independente da quantidade de IPPCs. Portanto, como os IPPCs funcionam como fonte de comandos e sua arquitetura é parametrizável [11,12], foi definido que para cada IPPC poderia existir qualquer quantidade de ICCs. Isso é possível a partir da adequação da interface de comunicação entre o IPPC e os ICCs.

No caso do controlador digital, duas configurações foram projetadas, sendo uma de SoC e outra de MPSoC. Em ambas as configurações, devido a requisitos de baixo custo e pela ausência de requisitos críticos de desempenho, utilizou-se apenas uma RECU. Este controlador foi utilizado para controlar funções de conforto de um automóvel, como: luz de cortesia, limpadores de pára-brisa, vidro elétrico e outros. Os IPPCs do SoC e do MPSoC são processadores paralelos e auxiliam a execução da RECU.

4. Resultados de Verificação

Esta seção apresenta os resultados obtidos das configurações de SoCs e MPSoCs desenvolvidas com a HRA-MPSoC. Esses resultados são de ocupação de recursos dos FPGAs e de desempenho.

4.1. Implementações

Os sistemas projetados foram descritos em VHDL sintetizável e os elementos de processamento RECU e ICC, previamente codificados, foram integrados em instâncias do IPPC [11,12] para construir os SoCs e os MPSoCs. As descrições VHDL foram sintetizadas e implementadas (*mapped, placed and routed*) para dois FPGAs de duas famílias distintas da Xilinx utilizando o pacote de ferramentas ISE. Esses FPGAs são um Spartan-3 XC3S200-4ft256 e um Virtex-II XC2V1500-5ff896. Estes dispositivos foram selecionados, pois possuem características bastante diferentes, como: quantidade de recursos em silício; preços e desempenhos. Com isso, verificou-se os recursos utilizados, o desempenho e o custo da implementação para FPGAs bastante diferentes.

A figura 4 apresenta a ocupação de recursos no FPGA da família Virtex-II para um dos SoCs projetados e implementados.

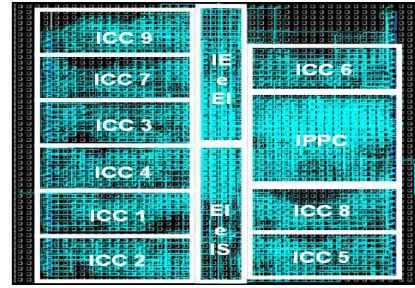


Figura 4. Floorplanning do 1-IPPC_9-ICCs no XC2V1500-5FF896.

Este SoC possui 1 IPPC e 9 ICCs de máscara 3x3 (1-IPPC_9-ICC). Essa é a nomenclatura utilizada para classificar os SoCs e MPSoCs desenvolvidos. Dentre as configurações projetadas, essa configuração possui a quantidade máxima de ICCs que o dispositivo suporta, considerando a quantidade de recursos disponíveis.

A partir da análise da figura 4, observa-se que, mesmo para a configuração máxima, existem muitos recursos do FPGA não utilizados. Porém, os recursos de memória (BRAMs) estão quase totalmente ocupados. A tabela 1 apresenta a quantidade de recursos utilizados para as diferentes configurações tanto do controlador como do filtro (máscara 3x3), para os dispositivos Spartan-3 (Sp) e Virtex-II (Vx).

Tabela 1. Quantidade de recursos utilizados.

Sistema	Regs	Slices	LUT	I/Os	BRAM
1-IPPC 1-RECU Sp	14,6%	57,8%	37,8%	42,8%	16,7%
2-IPPC 1-RECU Sp	27,9%	71,7%	54,5%	43,4%	33,3%
2-IPPC 1-ICC Vx	7,8%	29,2%	18,6%	14,2%	27,1%
1-IPPC 1-ICC Vx	4,5%	16,2%	10,3%	14,0%	22,9%
4-IPPC 2-ICC Vx	15,6%	59,5%	38,3%	15,3%	54,2%
1-IPPC 2-ICC Vx	5,7%	19,4%	12,3%	14,2%	41,7%
6-IPPC 3-ICC Vx	23,4%	88,8%	56,6%	16,5%	81,3%
1-IPPC 3-ICC Vx	6,9%	23,3%	14,7%	14,6%	60,4%
1-IPPC 4-ICC Vx	8,1%	26,7%	17,0%	15,2%	79,2%
1-IPPC 5-ICC Vx	9,3%	30,2%	19,1%	15,3%	97,9%
1-IPPC 9-ICC Vx	14,0%	49,8%	30,8%	16,7%	97,9%

Analisando a tabela 1, observa-se um aumento da utilização dos recursos de registradores (Regs), *Slices* e *Look-up-Tables* (LUTs) de aproximadamente duas e três vezes quando a configuração é variada entre 2-IPPC_1-ICC para 4-IPPC_2-ICC e para 6-IPPC_3-ICC, respectivamente. Entretanto, esta última configuração é a máxima quando projetada com acoplamento de quantidades de IPPCs e ICCs. Este limite se deve pela utilização de recursos lógicos (*Slices*) e de memória (BRAMs). Por outro lado, quando projetada com desacoplamento de quantidades, o dispositivo suporta até a configuração 1-IPPC_9-ICC. Portanto, no caso da filtragem ser realizada apenas nas instâncias de ICC, a configuração com acoplamento de quantidades apresenta uma ineficiência bastante significativa quando comparada com a configuração com desacoplamento. Esta ineficiência ocorre, pois com menor quantidade de recursos lógicos (Regs,

Slices, LUTs), a configuração máxima desacoplada possui 9 ICCs, enquanto que a acoplada possui apenas 3 ICCs. Esta diferença caracteriza um desempenho aproximadamente três vezes maior da desacoplada em relação à acoplada. Esse ganho caracteriza a vantagem da reconfiguração hierárquica nas camadas 1 e 2, aproveitando de modo eficiente os recursos do FPGA. Detalhes de desempenho são apresentados na seção 4.2.

Além disso, comparando as configurações acoplada e desacoplada, que possuem um único ICC (2-IPPC_1-ICC e 1-IPPC_1-ICC), verifica-se que para o mesmo potencial de processamento, a configuração acoplada utiliza quantidades maiores de recursos lógicos, com diferença variando entre 0,2% e 13,0% do total de recursos. Isso ocorre devido aos recursos utilizados para implementar os IPPCs adicionais.

No caso das configurações com desacoplamento implementadas no dispositivo Virtex-II, os recursos são subutilizados, com exceção dos recursos de memória (BRAMs). Observa-se ainda que a configuração máxima (1-IPPC_9-ICC) não utiliza nem a metade dos recursos lógicos. Entretanto, os recursos de memória são quase que totalmente utilizados (97,9%). A diferença insignificante de quantidade de BRAMs utilizadas pelas configurações 1-IPPC_5-ICC_Vx e 1-IPPC_9-ICC_Vx e a pouca diferença desta em relação às demais configurações de filtro, se devem a melhorias na utilização dos recursos de memória.

Com exceção da configuração 1-IPPC_9-ICC, todas as demais configurações utilizaram memórias de porta-única (*single-port*) para implementar multiplicadores como LUTs. Entretanto, verificou-se que esta opção de projeto é ineficiente quando a máscara de convolução possui pesos iguais. Isto porque o conteúdo das LUTs representa os valores de todas as combinações de pixel multiplicados por um determinado peso. Além disso, cada BRAM permite a implementação de porta-dupla (*dual-port*). Nesse caso, para maior eficiência na utilização das BRAMs, memórias de porta-dupla foram utilizadas para implementar dois multiplicadores com mesmo peso em uma única BRAM.

4.2. Análise do Desempenho

Esta seção apresenta análises dos desempenhos dos SoCs e MPSoCs desenvolvidos como configurações da HRA-MPSoC utilizando ICCs, para as aplicações de filtros de imagem. Esses resultados são comparados com outros tipos de implementação tanto comerciais como acadêmicas. A tabela 2 apresenta a frequência máxima de operação de cada configuração de SoC, MPSoC e de um processador comercial. O software executado no Pentium-IV possui melhorias de codificação e de compilação, como inversão de laço de repetição e utilização do compilador da Intel.

Tabela 2. Frequências máximas de operação.

Sistema	Frequência	Sistema	Frequência
Pentium-IV	3,8 GHz	6-IPPC_3-ICC	53,1 MHz
1-ICC	76,5 MHz	1-IPPC_3-ICC	62,2 MHz
2-IPPC_1-ICC	61,9 MHz	1-IPPC 4-ICC	61,5 MHz
1-IPPC_1-ICC	64,4 MHz	1-IPPC 5-ICC	60,2 MHz
4-IPPC_2-ICC	61,4 MHz	1-IPPC 9-ICC	55,5 MHz
1-IPPC_2-ICC	64,6 MHz		

Considerando a tabela 2, observa-se que as configurações desacopladas possuem frequência máxima de operação um pouco maior que as configurações acopladas que possuem a mesma quantidade de ICCs. Essa diferença se deve à maior quantidade de blocos lógicos utilizados e de sinais roteados no FPGA, aumentando o tempo de atraso de comunicação entre os blocos lógicos configuráveis.

A figura 5 apresenta os tempos de resposta para execução da filtragem de uma imagem 2048x2048 com diferentes máscaras. Os tempos apresentados são de uma configuração acoplada da HRA-MPSoC no Virtex-II, de um ICC isolado e de um Pentium-IV.

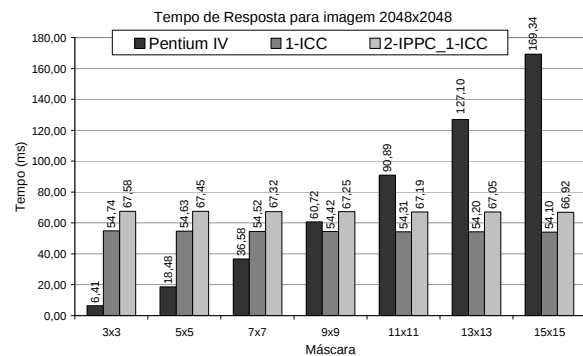


Figura 5. Tempos de resposta (ms).

Analisando o gráfico da figura 5, observa-se que a filtragem no Pentium-IV de uma imagem 2048x2048 possui tempos de resposta que aumentam bastante em função do aumento do tamanho da máscara. Esta tendência ocorre com a filtragem de qualquer tamanho de imagem, pois apesar de possuir paralelismo na execução de instruções, a sua arquitetura não explora o alto grau de paralelismo (espacial e temporal) implícito da convolução de imagens. No caso da execução com máscaras menores (3x3, 5x5 e 7x7), como o Pentium-IV possui arquitetura bastante avançada e elevada frequência de operação, seu desempenho é superior ao das configurações 1-ICC e 2-IPPC_1-ICC.

Por outro lado, ao contrário dos tempos do Pentium-IV, os tempos de resposta dos sistemas baseados em ICCs não sofrem aumentos significativos com o aumento do tamanho da máscara de convolução. Isso ocorre devido ao suporte da HRA-MPSoC à exploração do paralelismo espacial e temporal da operação de convolução, através da reconfiguração hierárquica. Neste caso, as implementações do ICC, utilizadas no

projeto e desenvolvimento dos SoCs e MPSoCs, exploram o paralelismo espacial e temporal existente em uma iteração da máscara de convolução. Esse paralelismo ocorre devido à reconfiguração na camada 2 da HRA-MPSoC. Ao contrário, os tempos sofrem uma pequena redução, quando o tamanho da máscara é aumentado e se as bordas da imagem não são processadas, como apresentado nas figuras 5 e 6. Essa pequena melhoria no desempenho é devido à diminuição da quantidade de pixels processados. Resumindo, devido à adequação da arquitetura às características da aplicação, os ganhos de desempenho em relação ao Pentium-IV são de até 2,53 para o 2-IPPC_1-ICC, com frequência 61,4 vezes menor.

Na figura 5, considerando as configurações que possuem um único ICC (1-ICC), a configuração com 1-ICC isolado possui o melhor desempenho, para uma mesma máscara, devido à sua maior frequência máxima de operação (tabela 2). Entretanto, esta configuração é um circuito dedicado de filtragem, sendo que o acesso à memória de dados deve ser realizado por alguma entidade externa ao circuito. No caso dos sistemas em chip, é desejável que os sistemas sejam autônomos [2]. Portanto, as configurações projetadas acessam a memória de dados sem o auxílio de entidades externas.

A figura 6 apresenta tempos de resposta de duas configurações com 1 ICC (acoplada e desacoplada) e uma configuração desacoplada com 2 ICCs no Vx-II.

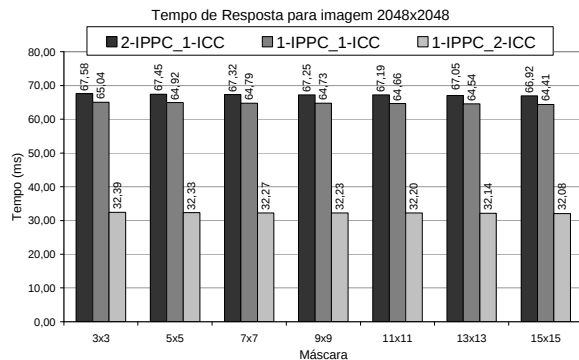


Figura 6. Tempos de resposta (ms).

Na figura 6, considerando as configurações da HRA-MPSoC projetadas com ICC único (n-IPPC_1-ICC), observa-se que os tempos da desacoplada são menores devido à diferença de frequência máxima de operação (tabela 2). Além disso, comparando os tempos das configurações 1-IPPC_1-ICC e 1-IPPC_2-ICC, observa-se que os tempos da segunda são aproximadamente a metade dos tempos da primeira. Como a diferença entre as frequências máximas de operação não é significativa (da tabela 2: 64,4 MHz e 64,6 MHz), esse ganho de desempenho ocorre devido ao paralelismo proporcionado pela duplicação da quantidade de ICCs (EP-NP-Fs). Esta duplicação

possibilita a divisão da imagem em dois subconjuntos de pixels e o processamento paralelo em cada ICC.

A figura 7 apresenta os tempos de resposta da filtragem, com máscaras 3x3 e 15x15, para sete configurações de MPSoC configuradas no Virtex-II. Dessas, duas são acopladas e cinco são desacopladas.

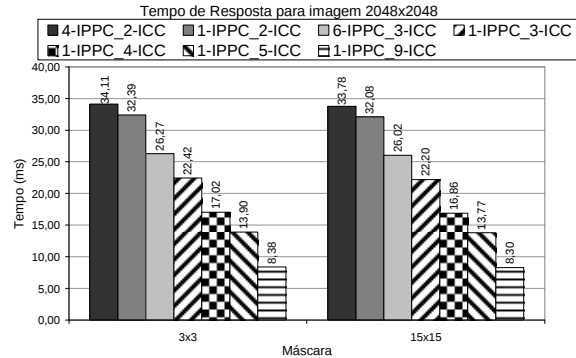


Figura 7. Tempos de resposta (ms).

Analisando a figura 7, observa-se que as configurações acopladas possuem tempos de resposta maiores que as desacopladas, quando possuem o mesmo número de ICCs. Como mencionado anteriormente, esta diferença de desempenho se deve à diferença de frequência máxima de operação (tabela 2). Além disso, quando se compara duas configurações acopladas (ex. 4-IPPC_2-ICC e 6-IPPC_3-ICC) ou duas desacopladas (1-IPPC_2-ICC e 1-IPPC_4-ICC), o ganho de desempenho não é proporcional com o inverso da quantidade de ICCs. O motivo disso é a diferença das frequências de operação das acopladas (61,4 MHz e 53,1 MHz) e das desacopladas (64,6 MHz e 61,5 MHz), como apresentado na tabela 2.

Analisando as configurações desacopladas (1-IPPC_m-ICC), observa-se que as configurações com 2, 3, 4, 5 e 9 ICCs possuem ganhos de desempenho variando de 2,01 a 7,76 em relação à configuração desacoplada com apenas um ICC (1-IPPC_1-ICC). Os ganhos de desempenho dos SoCs e MPSoCs projetados e implementados, em relação ao Pentium-IV ou ao SoC com um ICC, se devem às implementações com paralelismo na execução das operações da carga de trabalho. Esse paralelismo foi possível devido às características de flexibilidade e escalabilidade da HRA-MPSoC. Os ganhos de desempenho obtidos não seriam possíveis se a HRA-MPSoC não permitisse a adequação da arquitetura do sistema implementado, com a reconfiguração das suas camadas 1, 2 e 3, de acordo com o paralelismo da operação de convolução.

Amaral e outros [13] apresentam ganhos de desempenho bastante significativos do circuito com 1-ICC isolado, comparado com trabalhos acadêmicos e comerciais da literatura [16]. Como apresentado nesta seção, as configurações de SoCs e MPSoCs, possuindo

mais de 1 ICC, apresentaram ganhos de desempenho significativos comparados com o desempenho de 1-ICC. Portanto, conclui-se que os SoCs e MPSoCs desenvolvidos possuem ganhos ainda maiores que os obtidos em [13], quando seus desempenhos são comparados com os trabalhos acadêmicos e comerciais. Por exemplo, para convolução com máscara 3x3 de imagem 512x512, os ganhos em relação aos trabalhos acadêmicos e comerciais variam de 3,47 a 9,13 para 1-ICC e entre 14,12 e 37,25 para 1-IPPC_9-ICC.

5. Conclusões e Trabalhos Futuros

Após a análise dos resultados apresentados na seção anterior, conclui-se que os objetivos desta pesquisa foram alcançados. Neste artigo, apresenta-se e verifica-se a arquitetura hierarquicamente reconfigurável de sistemas multiprocessados em chip (HRA-MPSoC). Sendo parte da pesquisa em arquiteturas hierarquicamente reconfiguráveis de sistemas computacionais (HRA-CS), este artigo apresentou os resultados da utilização e reconfiguração das camadas que são implementadas como sistemas em chip.

Os resultados de flexibilidade e desempenho apresentados verificaram as vantagens da utilização da HRA-MPSoC no projeto e desenvolvimento de SoCs e MPSoCs. Com a HRA-MPSoC e sua reconfiguração hierárquica, foi possível o projeto e desenvolvimento de sistemas em chip para duas aplicações bem distintas, utilizando uma única arquitetura. Além disso, para cada aplicação foram desenvolvidos SoCs e MPSoCs com diferentes quantidades de EPs (camada 1) com diferentes características internas (camadas 2 e 3), para atender aos requisitos da carga de trabalho.

A principal contribuição desta pesquisa é a proposta de uma arquitetura que permita o desenvolvimento de sistemas em chip flexíveis tanto em tempo de projeto como em tempo de execução. Essa flexibilidade é viabilizada a partir da reconfiguração das diferentes camadas hierárquicas da arquitetura e da implementação. Com isso, os SCEs desenvolvidos podem ser adequados às características das cargas de trabalho e aos requisitos de diferentes aplicações.

Os principais trabalhos futuros são: projeto e implementação de sistemas em chip com as demais classes de EPs (figura 2) e o projeto e implementação de um sistema hierarquicamente reconfigurável em tempo de execução, utilizando a HRA-MPSoC.

Agradecimentos

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil.

Referências

- [1] W. Wolf, “Embedded Computer Architectures in the MPSoC Age”, *32th International Symposium on Computer Architecture*, IEEE Computer Society, Madison, 2005.
- [2] J. Becker, “Configurable Systems-on-Chip (CSoC)”. *15th Symposium on Integrated Circuits and Systems Design (SBCCI)*. IEEE C.S., Washington, DC, 2002, pp. 379.
- [3] W. Wolf, “Multimedia applications of systems-on-chips”, *Conference on Design, Automation and Test in Europe - Designers’ Forum*, ACM Press, 2005, pp. 86-89.
- [4] T. Austin et al, “Mobile Supercomputers”, *IEEE Computer*, 37(5), IEEE Computer Society, 2004, pp. 81-83.
- [5] G.J.M. Smit et. al, “The Chameleon Architecture for Streaming DSP Applications”, *EURASIP Journal on Embedded Systems*, 2007, pp. 1-10.
- [6] M.B. Taylor et al, “The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs”, *IEEE Micro* 22, 2, 2002, pp. 25-35.
- [7] A. Herkersdorf, W. Stechele. “AutoVision: flexible processor architecture for video-assisted driving”, *Conference on Design, Automation and Test in Europe*, European Design and Automation Association, Belgium, 2006, pp. 556-556.
- [8] J.-Y. Mignolet et al, “Enabling hardware-software multitasking on a reconfigurable computing platform for networked portable multimedia appliances”, *Int. Conf. on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, USA, 2002, pp. 25-27.
- [9] A.M. Amaral, M.O.S. Souza, C.A.P.S. Martins, “HRA-CS: Hierarchical Reconfigurable Architecture for Computer Systems” (Relatório Técnico), PUC Minas, Belo Horizonte, 2007.
- [10] W. Wolf, *Computers as Components: Principles of Embedded Computing System Design*, M. Kaufman, San Francisco, 2000.
- [11] A.M. Amaral, “R2NPC (Reconfigurable RISC Network Processor Core): Projeto e Implementação do Núcleo do Processador de Rede R2NP” (Monografia), Departamento de Engenharia Eletrônica, PUC Minas, Belo Horizonte, 2006.
- [12] A.M. Amaral, C.A.P.S. Martins, “IPPC: Intellectual Property Processor Component applied in Embedded Computer Systems”, *VII Student Forum on Microelectronics*, SBC, Rio de Janeiro, 2007.
- [13] A.M. Amaral, M.B. Carvalho, C.A.P.S. Martins, “Arquitetura Paralela e Parametrizada para Convolução de Imagens”, *VII Workshop de Sistemas Computacionais de Alto Desempenho*, SBC, Ouro Preto, 2006, pp. 195-198.
- [14] A.M. Amaral, C.A.P.S. Martins, “Image Convolution Circuit: Parallel and Parameterized Architecture and FPGA Implementation”, *V Student Forum on Microelectronics*, SBC, Florianópolis, 2005.
- [15] H.C. Freitas, M.B. Carvalho, A.M. Amaral et al, “Reconfigurable Crossbar Switch Architecture for Network Processors”, *IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE CAS Society, Kos, 2006. pp. 42-45.
- [16] C.T. Huitzil, M. A. Estrada, “Real-time Image Processing with a Compact FPGA-based Systolic Architecture”, *Journal of Real-time Imaging*, Elsevier, 2004, pp. 177-187.