



PROPOSTA DE UMA ARQUITETURA MULTIPROCESSADA BASEADA NO PROCESSADOR SOFTCORE NIOS II DA ALTERA

Henrique Rodrigo Terneiro da Silva *
e-mail: henrique@icmc.usp.br

Maria Stela Veludo de Paiva*

Leandro Carlos Fernandes *

Eduardo Marques **

Rodrigo Fernandes de Mello **

e-mail: mstela@sel.eesc.usp.br
e-mail: {lnd, emarques, mello}@icmc.usp.br

* USP/EESC, São Carlos, Brasil

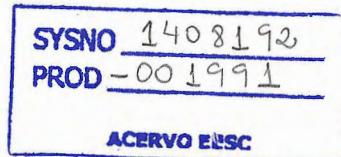
** USP/ICMC, São Carlos, Brasil

Abstract: Now a days, we can see an expansion of reconfigurable computing area. In fact, this growth if gives mainly for the increase of the FPGA capacity and the researches using this kind of technology. The increase of logic cells number into a FPGA allow, for the first time, a designer put much more than one processor inside a single-FPGA enabling the new architectures development. A new multiprocessed architecture based in many instances of Altera's Nios II softcore processor putting together into only one FPGA chip is proposed in this paper. Thus, this architecture is multiprocessed because it aims to run parallel process using various processors at same time. The tiled-architecture is used to insure that the architecture will be scalable. A propose to the load balance is suggested in order to advance the distribution of the processes running.

Palavras Chaves: FPGA, computação paralela, Nios, SOPC, Tiled-Architecture.

Introdução

A resolução de problemas que podem ser solucionados de forma paralela tem sido um grande desafio para as plataformas de computação, especialmente quando as operações tem que ser realizadas em uma linha de tempo. Sistemas como cluster de supercomputadores ou de computadores pessoais, multicomputadores de memória compartilhada distribuída utilizando barramento cruzado e redes de interconexão multi-estágio, e cluster de sistema escalar de uni- e multiprocessador domina o campo de alta performance [1]. Estes sistemas de computação conseguem resolver os problemas de intensivo paralelismo, porém o alto custo financeiro, seu alto custo de projeto e de ciclos de desenvolvimento, as dificuldades de



implementação de sistemas, são algumas das dificuldades enfrentadas na implantação dos mesmos.

Por outro lado, temos o contínuo desenvolvimento da indústria de semicondutores e das vantagens no desenvolvimento de projeto de arquitetura. Um desses projetos são as Field-Programmable Gate Arrays (FPGAs), as quais tem crescido de tal forma que permitem formar plataforma de computação System-on-a-Programmable-Chip (SOPC ou SOC). Novas gerações de FPGA tem possibilitado integrar um grande número de módulos de computação e construir máquinas paralelas em um simples dispositivo de FPGA [2]. Para trabalhar com as FPGAs temos os processadores configuráveis Nios II da Altera [3]. O processador embarcado Nios II é otimizado para lógica programável Altera e soluções SOPC. A Altera fornece uma poderosa ferramenta de desenvolvimento integrado de sistema, o SOPC Builder, que suporta implementação de sistemas de processador embarcado baseado em Nios II.

Neste artigo, é apresentado o projeto de uma máquina multiprocessada com memória compartilhada baseada no processador configurável Nios II da Altera como nós de elementos de processamento [3]. Sua arquitetura inicialmente será projetada com um número fixo de CPUs (Central Process Units) e posteriormente serão adicionados mais processadores Nios II de forma que o sistema reconheça automaticamente a inserção de novos processadores e se reconfigure de acordo com a nova topologia. Além disto, será apresentado uma proposta para o balanceamento de carga em nível de sistema.

Plataformas Paralelas

Depois de vários anos de pesquisa em sistemas como cluster de supercomputadores ou de computadores pessoais, multi-computadores de memória compartilhada distribuída utilizando barramento cruzado e redes de interconexão multi-estágio, e cluster de sistema escalar de uni- e multi-processador no campo de alta performance [1], conseguiu-se desenvolver sistemas de computação que conseguem resolver os problemas de intensivo paralelismo. Porém o alto custo financeiro, seu alto custo de projeto e de ciclos de desenvolvimento, as dificuldades de implementação dos sistemas, são algumas das dificuldades enfrentadas na implantação dos sistemas paralelos.

Durante anos os centros de supercomputadores foram utilizados para computação distribuída, sendo atualmente também usados como nós de computação peer-to-peer [4]. A fim de tornar a computação paralela disponível a maioria das pessoas, todos os nós da Internet em “grid computing” são candidatos a resolver problemas de larga escala em um meio de computação distribuída [2]. Entretanto, estas melhorias para computação de alta performance não são viáveis para sistemas dedicados para uma simples aplicação ou para soluções de baixo custo.

Objetivando computação paralela de custo efetivo, pesquisadores tem deslocado suas atenções para a adoção de clusters contendo nós de PC (Personal Computer). Estes sistemas são adequados para problemas de grande escala, tal como mineração de dados (data mining) em grandes banco de dados [5]. Cluster de PCs são sistemas descentralizados. Cada nó é um PC independente executando seu próprio sistema operacional. A comunicação de dados entre os PCs são controlados pelo software de camada de aplicação de preferência que pelo sistema de software de baixo nível ou hardware. Entretanto, a latencia das comunicações de dados é usualmente mais longa do que de computadores paralelos e supercomputadores que contém hardware especializado para implementar redes de comunicação. Além disso, os modeos de programação para executar aplicações sobre cluster de PCs difere daquelas realizadas em

máquinas paralelas ou supercomputadores. Outro problema é o balanceamento de carga entre PCs, o qual se torna um assunto crítico na tentativa de conseguir alta performance.

Um outro segmento de pesquisadores estão realizando pesquisas na solução de problemas que envolvem paralelismo utilizando FPGA e processadores Nios II. Isto ocorre porque a indústria de chips e o projeto de arquitetura estão se desenvolvendo continuamente. Pode-se destacar como vantagens: o barateamento dos custos, já que não é necessário comprar um supercomputador ou vários computadores; a questão do espaço físico, pois as FPGAs são pequenas; o aumento das células lógicas dentro de uma FPGA, permitindo que um sistema com várias instruções possam ser colocadas nela; e, podem formar uma plataforma de computação de um sistema dentro de um chip programável (conceito de SOPC). A utilização de um processador Nios II, ou vários, permite que um sistema possa ser escrito em uma linguagem de alto nível (linguagem C), o que facilita o desenvolvimento de sistemas.

Cada componente ou parte de um componente que pertence a uma arquitetura pode ser denominado de *tile*. Este é um conceito novo que está surgindo para denominar parte de um sistema ou arquitetura. Em [6], por exemplo, é usado o conceito de *tile* para denominar uma quantidade de pixel da tela, usado para o desenvolvimento do método AHV (Adaptive Hierarchical Visibility). O projeto Raw [7] conecta um arranjo de *tiles* de processadores simples, usando o conceito de *tile* para denominar um processador.

Nesta proposta é usado o conceito de *tile* para denominar um processador Nios II que pode ser anexado ao sistema, que ajudará os outros processadores na execução das aplicações paralelas.

FPGAs e Processadores Nios II

As FPGAs são dispositivos programáveis que surgiram da tecnologia dos MPGAs (Mask-Programmable Gate Arrays). A principal diferença é que nos FPGAs as interconexões são programadas via comutadores programáveis eletricamente, assim como nos PLDs [8], já nos MPGAs, a interconexão é feita durante o processo de fabricação, como em circuitos integrados.

A computação reconfigurável combina a velocidade do hardware com a flexibilidade do software. Velocidade do hardware porque o programa final residirá em uma FPGA. Como a programação de um FPGA é realizada pelo próprio usuário final, permite ao mesmo criar uma solução para suas próprias necessidades, igualando à flexibilidade que se tem quando se desenvolve um sistema através de software.

A tecnologia da computação reconfigurável consiste na habilidade de se modificar o hardware da arquitetura para esse se adequar à aplicação [9]. Essa reconfiguração do hardware pode ser estática ou dinâmica. Na reprogramação estática, a utilização do chip deve ser suspensa até que o processo de reprogramação seja concluído. Na reconfiguração dinâmica algumas partes do chip podem ser utilizados sem a necessidade de interrupção do processo de reprogramação. As regiões que não são alteradas, e somente as regiões que não estão sendo usadas no momento da reprogramação são reprogramadas.

O desenvolvimento de sistemas para as FPGAs é utilizado as linguagens de descrição de hardware VHDL ((Very High Speed Integrated Circuit) Hardware Description Language) e Verilog. VHDL é uma forma de se descrever, através de um programa, o comportamento de um circuito ou componente digital. Verilog é uma Linguagem de Descrição de Hardware (HDL – Hardware Description Language) usada por desenvolvedores de hardware na indústria e universidades, sendo esta muito semelhante a linguagem de programação C, ensinada na maioria das universidades.

A Altera e a Xilinx são as duas principais indústrias de fabricação de FPGA. A Altera possui uma linha de chips chamado Stratix II, como mostrado na Figura 1(a) [3]. O chip fabricado pela Xilinx é denominado Virtex II, e pode ser visto na Figura 1(b) [10]. A Altera fornece um ambiente de desenvolvimento de sistemas chamado Quattus, o qual trabalha tanto com VHDL como Verilog, além de um ambiente gráfico que permite fazer esquemáticos de módulos do sistema. O ambiente ISE (Integrated Software Environment) é o nome do ambiente de desenvolvimento da Xilinx, o qual também possui os recursos citados para o ambiente da Altera.

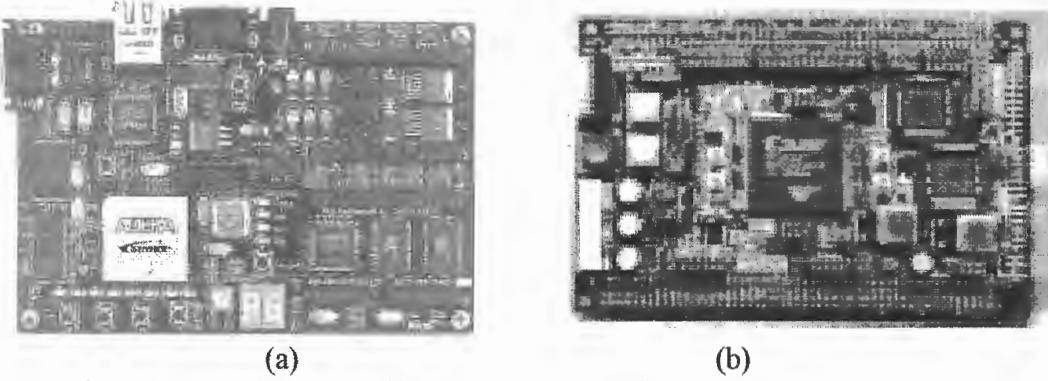
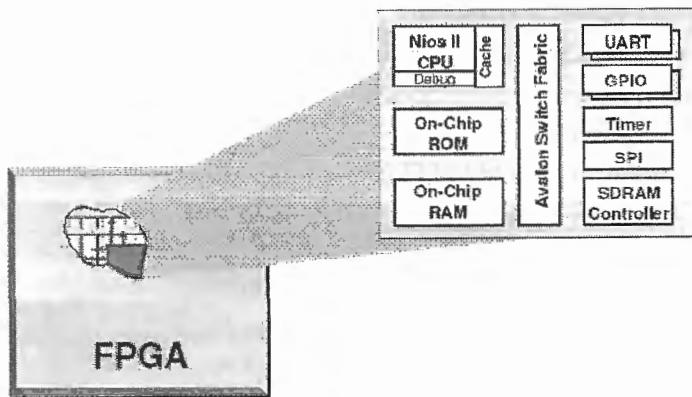


Figura 1: (a) Placa da Altera Stratix II; (b) Placa da Xilinx Virtex II

As placas Stratix da Altera podem usar uma instância do processador configurável chamado Nios II, o qual é baixado para dentro da FPGA através do SOPC Builder. O processador RISC (Reduced Instruction Set Computing) Nios II é totalmente flexível e configurável que oferece 125 MHz dentro da placa. A CPU trabalha com palavras de 16 ou 32 bits, velocidade de clock, arquivo de registrado, SDK (Standard Development Kit) espaço de endereço, memória RAM (Random Access Memory) ou ROM (Read Only Memory) on-chip, disponibilidade de multiplicados hardware/software e vários outros periféricos on-chip podem ser usados para especificações de usuários.

Utilizando um processador Nios II permite uma melhor comunicação com os periféricos da placa Stratix, como a utilização de um monitor de vídeo para saída do processamento, utilização de um pequeno display e outros periféricos que podem ser acoplados à placa. A Figura 2 apresenta uma visão do que pode conter uma FPGA. Além do Nios II, temos uma memória ROM, uma memória RAM, uma chave de barramento Avalon, um timer, um controlador SDRAM (Synchronous Dynamic Random Access Memory), um SPI (Serial Peripheral Interface), um GPIO (General Purpose Input and Output), um UART (Universal Asynchronous Receiver Transmitter).

Figura 2: Visão do processador Nios II e outros componentes dentro de uma FPGA



Para a escrita de sistemas que serão executados pelo processador Nios II é utilizado a linguagem de programação C, conhecida por vários programadores. A linguagem C é uma linguagem de alto nível, que permite fazer dentre outras coisas comunicação com o hardware e escrever programas paralelos.

Um ambiente de desenvolvimento de sistemas para o processador Nios II é disponibilizado pela Altera. Este ambiente possui duas ótimas ferramentas: a Plataforma Eclipse e o SOPC Buider. A Plataforma Eclipse permite a escrita de programas utilizando várias linguagens de programação, tais como: C, Verilog e Java.

SOPC Buider é uma poderosa plataforma para composição de sistemas definidos no nível de blocos ou de componentes. O SOPC Buider tem um biblioteca de componentes que vai desde simples blocos de lógica fixa até componentes complexos, parametrizados e subsistemas dinamicamente gerados. Esses componentes podem ser comprados como componente de propriedade intelectual (IP – Intellectual Property) da Altera ou de terceiros, alguns podem ser baixados e usados livremente e os usuários podem também facilmente criar seus próprios componentes SOPC Buider.

Utilização de paralelismo usando mais de um processador Nios II

Dentro de uma única FPGA pode-se ter várias instâncias do processador softcore Nios II da Altera. Cada processador tem sua própria memória local onde instruções e dados podem ser armazenados. Existe uma memória adicional que pode ser acessada globalmente. Através dessa memória é possível compartilhar dados, pois todos os processadores podem acessá-la. As conexões entre os vários componentes são feitos usando o sistema de barramento Avalon.

A Altera define Nios II como sendo um “microprocessador RISC de propósito geral” [3]. Nios II é configurável, isto é, suas características podem ser selecionadas pelo usuário dependendo da memória e requisitos de lógica, tais como suporte de ponto flutuante, suporte a interrupções de hardware, etc. Nios II também suporta instruções configuráveis escritas pelo usuário armazenada na região chamada de *custom logic* (lógica configurável). Estas instruções podem ser adicionadas pelo usuário ao projeto do Nios para executar tarefas específicas que podem ser suportadas pelo Nios II. Estas instruções, nas quais são implementadas como blocos de hardware projetado usando uma linguagem de descrição de Hardware, tal como VHDL ou Verilog, são adicionadas ao Nios II usando o ambiente SOPC Builder da Altera. Esta opção oferece melhoramento significante em termos de temporização, eficiência, etc.

O sistema de barramento Avalon é disponibilizado pela Altera, o qual permite conectar processadores e periféricos, criando conexões ponto-a-ponto entre eles. Assim, pode-se trocar informações entre os processadores utilizando o barramento Avalon e a memória global

compartilhada por todos os processadores. A Figura 3 mostra os vários processadores Nios II que podem estar presentes em uma FPGA, o que cada CPU contém, incluindo o barramento Avalon, e onde fica a *custom logic*.

Um sistema paralelo pode ser implementado usando este tipo de arquitetura. Cada componente do sistema pode ser executado por um dos processadores Nios II e as estruturas globais podem ser armazenadas na memória global, a qual é compartilhada por todas as CPUs. Os processadores podem se comunicar através de várias técnicas de comunicação de processos. Como exemplo, podemos citar a técnica de variável de comporta e a troca de mensagens [11]. A técnica de variável de comporta é mais simples de implementar e pode dizer quando uma determinada informação está disponível na memória global para o outro processador. Já a troca de mensagens poderia ser implementada dentro do próprio sistema, simplesmente fazendo a troca de informação entre as CPUs.

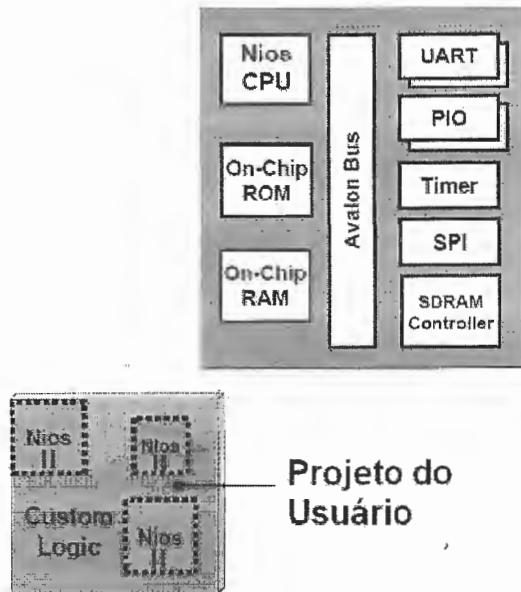


Figura 3: FPGA com vários Nios II

A proposta deste trabalho é propor uma arquitetura que utilize vários processadores Nios II, distribuindo o paralelismo existente para um sistema paralelo denominado Armosh (Figura 4), que faça o controle da comunicação existente entre as partes paralelas desse sistema e faça o balanceamento de carga dos processadores, para que haja uma igualdade e agilidade na execução das tarefas.

Proposta

A microeletrônica fornece-nos uma oportunidade para projetar máquinas paralelas em um simples chip no qual pode-se aumentar a eficiência incrementando o número de processadores para muitas aplicações. Enquanto a quantidade de transistores dentro de um simples chip tem aumentado constantemente, somente agora é que o problema da realização de microarquitetura paralela de propósito geral sobre um simples chip tem surgido como grande desafio. Protótipos de pesquisas tais como *Trips* [12], *Smart Memories* [13], e *Raw* [14] representam os primeiros passos dentro do espaço de projeto de arquiteturas *tiled*.

Em [14] é apresentado a exploração no espaço de projeto para o Microprocessador Raw como exemplo para uma arquitetura *tiled*. O Microprocessador Raw é reminiscente de FPGA e engloba um conjunto replicado de *tiles* acoplados juntos por um conjunto de compiladores, fios, chaves. Cada *tile* contém um componente de processamento como RISC e memória DRAM para instruções e dados (Figura 5).

O primeiro desafio é determinar a melhor utilização dos recursos entre computação, memória e comunicação, isto é, o problema do balanceamento de carga. O segundo é determinar o número de granularidade dos *tiles*, o problema da granularidade.

Para resolver o problema da granularidade propõe-se a utilização de processadores Nios II simétricos, ou seja, com os mesmos elementos de processamento. Sendo iguais, o projeto de implementação dos *tiles* fica mais previsível e de fácil expansão quando surgir a necessidade de um aumento no número de processadores.

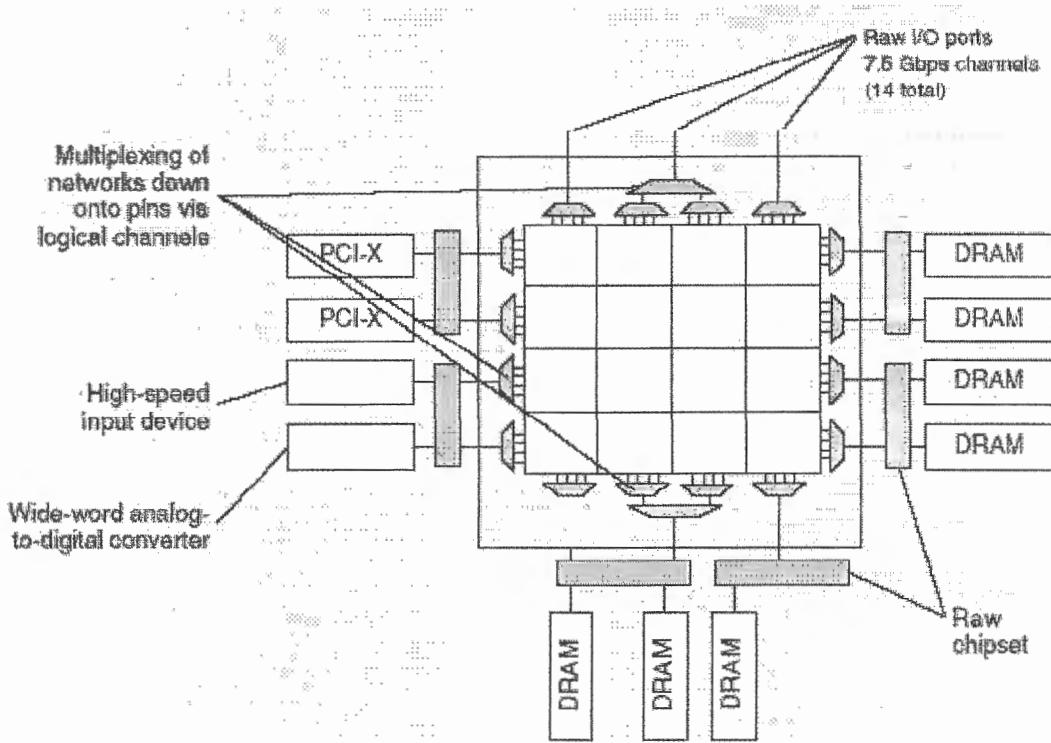


Figura 5: Multiplexagem dos pinos e uso dos dispositivos em Raw.

O problema para o balanceamento de carga será resolvido inicialmente utilizando o algoritmo de balanceamento de carga para cluster de computadores chamado TLBA (*Tree Load Balancing Algorithm*). Para executar as operações de balanceamento de carga, os computadores são organizados na forma de árvore (apenas uma topologia lógica) [15].

Similarmente a esta técnica, os processadores serão colocados de forma a formarem uma topologia lógica de uma árvore. Quando um novo processador é inserido no sistema, o processador localizado na raiz da árvore determina em qual ramo o novo processador irá ser adicionado.

Quando surgir a necessidade de migração de um processo para outra CPU, faz-se uma análise da carga do sistema, ou seja, uma análise de processamento das CPUs pertencentes à árvore de balanceamento. Para essa análise é utilizado a carga de cada CPU e de cada memória local pertencente a um *tile* em um determinado instante, como apresentado em [15].

Conforme apresentado na Figura 4, um robô autônomo será desenvolvido utilizando FPGAs. Esse robô terá um sistema operacional µCLinux para a execução de processos. Essa proposta de trabalho visa implementar uma arquitetura que permite um melhor aproveitamento dos recursos de FPGA do robô. Pode-se por exemplo, distribuir a tarefa de aquisição e processamento da imagem com outros dispositivos que por ventura não estejam sendo utilizados em um dado instante. Assim, a distribuição dos processos e balanceamento de carga se torna altamente necessário para uma maior eficiência do sistema como um todo. A Figura 4 ilustra os componentes do robô, onde cada FPGA controlará uma função específica do robô, como por exemplo, uma FPGA responsável pelo controle dos sensores do robô, outra pela aquisição da imagem, etc.

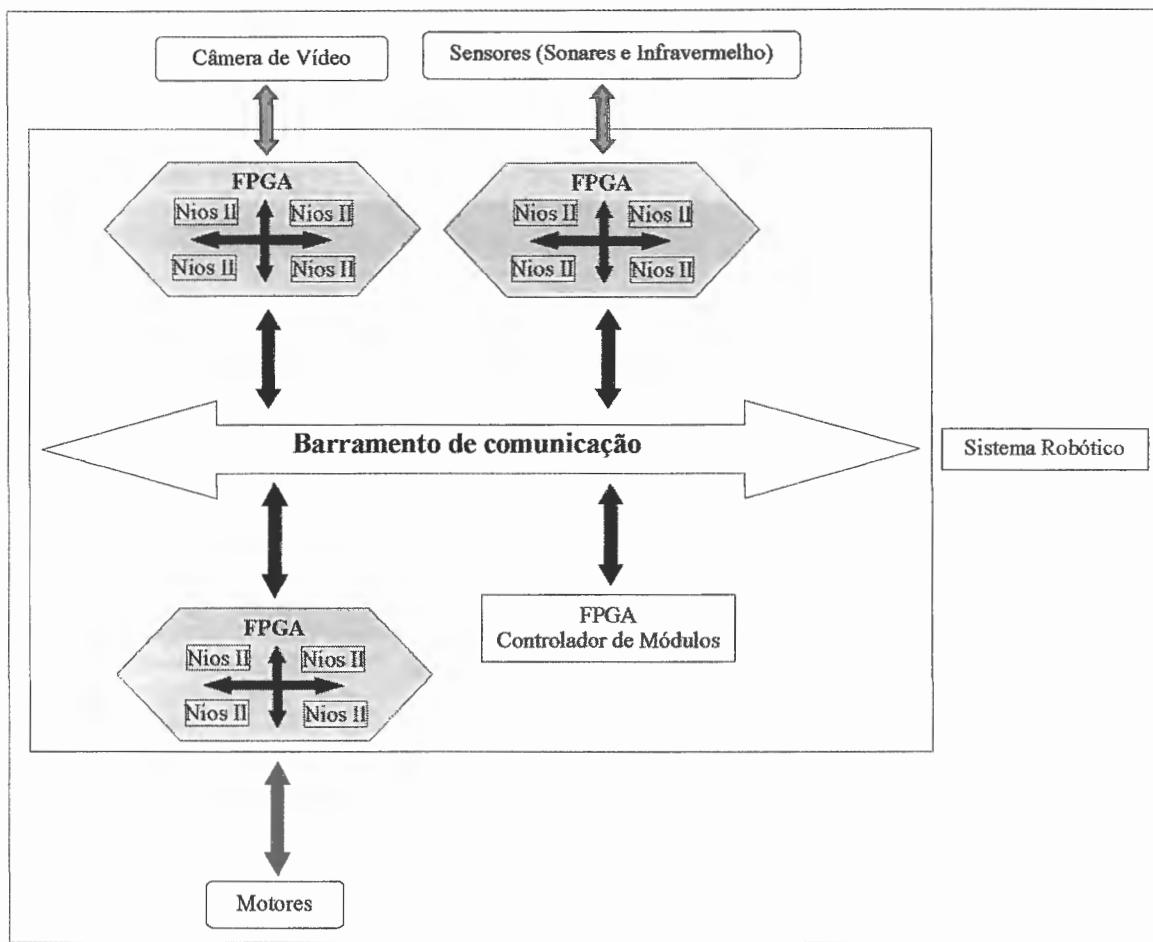


Figura 4: Visão dos componentes do robô – Projeto Armosh

Conclusão

Este artigo apresentou uma proposta para utilização de uma arquitetura multiprocessada que utiliza várias instâncias de processadores softcore Nios II dentro de um único chip de FPGA. Isto permite a execução de sistemas paralelos de forma mais rápida, possibilitando um ganho de tempo de execução dos processos.

A evolução das arquiteturas paralelas é apresentada para se ter um acompanhamento da evolução dessa tecnologia que permite a execução de processos de forma mais rápida e

distribuída. Dentro dessa arquitetura paralela encontra-se o parallismo proposto para as arquiteturas *tiled*.

O modelo de arquitetura *tiled* é usado para permitir que a arquitetura proposta possa ser expandida sem preuízo para o sistema. Isso se deve ao fato da arquitetura *tiled* definir que todas as instâncias de processadores dentro da FPGA terão a mesma configuração.

Uma proposta para o balanceamento de carga dos processadores pertencentes à arquitetura é proposta para tornar o paralelismo mais eficiente e permitir uma melhor distribuição do processamento entre os processadores.

E por fim, este trabalho apresenta o projeto de um robô autônomo que possui vários componentes FPGA e podem ter vários processadores Nios II. Dentro dessas FPGA pretende se ter vários processadores Nios II que permitia a execução de uma tarefa de forma paralela, como por exemplo a aquisição das imagens obtidas pelas câmeras do robô.

Referências

- [1] Li, X., Ziavras, S. G., Manikopoulos, C. N., “*Paralell DSP algorithms on TurboNet: an experimental hybrid message-passing/shared-memory architecture*”, Concurrency and Computation: Practice and Experience, pp. 387-411, 1996.
- [2] Wang, X., Ziavras, S. G., “*Paralell LU Factorization of Sparse Matrices on FPGA-Based Configurable Computing Engines*”, Concurrency and Computation: Practice and Experience, V. 16, N. 4, p. 319-343, 2003.
- [3] Altera Corporation, Internet site address: <http://www.altera.com> acessado em 25/09/2004.
- [4] Bell, G., Gray, J., “*High performance computing: Grays, clusters and centers. What next?*”, Technical Report MSR-TR-2001-76, Microsoft Research, San Francisco, CA, 2001.
- [5] Oguchi, M., Shintani, T., Tamura, T., and Kitsuregawa, M., “*Optimizing Protocol Parameters to Large Scale PC Cluster and Evaluation of its Effectiveness with Parallel Data Mining*”, Proc 7th Internation Symposium High Performance Distributed Computation, pp. 34-41, 1998.
- [6] Xie, F., Shantz, M., “*Adaptive Hierarchical Visibility in a Tile Architecture*”, Scientific American, 1997.
- [7] Taylor, M. B., Lee, W., et. al., “*Evaluation of the Raw Microprocessor: An Exposed-Wire-Delay Architecture for ILP and Streams*”, The 31st Annual International Symposium on Computer Architecture, 2004.
- [8] Aragão, A. C. O. S., “*Uma Arquitetura Reconfigurável Dinamicamente Dedicada ao Controle de Robôs Móveis*”, Dissertação de Mestrado – ICMC USP, 2000.
- [9] Villasenor, J., Mangione-Smith, W. H., “*Configurable Computing*”, Scientific American, 1997.
- [10] Xilinx Corporation, Internet site address: <http://www.xilinx.com> acessado em 25/09/2004.

- [11] Tanenbaum, A. S., “*Sistemas Operacionais Moderno*”, Ed. Makon Books, São Paulo, 2003.
- [12] Nagarajan, R., Sankaralingam, K., Burger, D. C. Keckler, S. W., “*A Design Space Evaluation of Grid Processor Architectures*”, In 34th Annual International Symposium on Microarchitecture, pp. 40-51, 2001.
- [13] Mai, K., Paaske, T., Jayasena, N., Ho, R., Dally, W. J., Horowitz, M., “*Smart Memories: A Modular Reconfigurable Architecture*”, In 28th Annual International Symposium on Computer Architecture, pp. 161-171, 2000.
- [14] Taylor, M. B., Kim, J., Miller, J., Wentzla, D., Ghodrat, F., Greenwald, B., Homann, H., Johnson, P., Lee, J., Lee, W., Ma, A., Saraf, A., Seneski, M., Shnidman, N., Strumpen, V., Frank, M., Amarasinghe, S., Agarwal, A., “*The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs*”, IEEE Micro, pp. 25-36, 2002.
- [15] Mello, R. F., “*Proposta de avaliação de desempenho de um algoritmo de balanceamento de carga para ambientes distribuídos heterogêniros escaláveis*”, EESC – USP, 2003.