CAPÍTULO I

INTRODUÇÃO

Em algumas áreas, como por exemplo: processamento de imagens, meteorologia e compressão de dados em tempo real; a capacidade de processamento requerida é bem maior do que os sistemas atuais podem prover. Todas estas áreas têm em comum o fato de tratarem de problemas envolvendo grande quantidade de dados estruturados em forma de matrizes ou vetores.

Para atender a esta demanda existem dois caminhos. O primeiro é projetar computadores utilizando arquiteturas convencionais, mas que são implementadas com os circuitos integrados mais rápidos disponíveis. Estes projetos requerem técnicas de custo bastante elevado, só viáveis para grandes companhias. Outra alternativa é empregar o paralelismo e novas arquiteturas para obter desempenho a baixo custo.

Entre as arquiteturas paralelas existem dois modelos bem distintos: o modelo MIMD (*Multiple Instruction Stream Multiple Data Stream*) e o modelo SIMD (*Single Instruction Stream Multiple Data Stream*). Um sistema MIMD geralmente é constituído de vários processadores independentes que se comunicam através de uma rede de interconexões, cada um executando sua própria parte do programa. Um sistema SIMD é composto por um conjunto de Elementos Processadores (EPs), geralmente organizados em forma de matriz, atuando sob o controle de uma única Unidade de Controle (UC). Num sistema SIMD todos os processadores ativos executam a mesma instrução ao mesmo tempo, porém sobre dados diferentes.

Uma vantagem do modelo SIMD sobre o modelo MIMD é que o primeiro possui um único fluxo de instruções, dispensando novas técnicas de programação. Isto facilita a adição de construções paralelas às linguagens convencionais, como PASCAL, C e FORTRAN e, conseqüentemente, possibilita uma migração suave dos programas já existentes e dos usuários dessas linguagens. Além disso, os nós de processamento das máquinas MIMD são bem mais complexos, o que limita por problemas de custo o número de nós que se pode construir neste tipo de arquitetura. Essa complexidade se deve, primeiramente, ao fato de os nós das máquinas MIMD serem processadores completos, enquanto que os nós das máquinas SIMD são basicamente unidades lógicas e aritméticas de estrutura simples. Em segundo lugar porque, em máquinas MIMD, cada processador deve possuir memória suficiente para armazenar não só

seus dados, como também sua própria cópia da parte que lhe interessa do programa, o que não ocorre nas máquinas SIMD, onde a memória de cada processador armazena exclusivamente seus dados.

Este trabalho visa desenvolver o Elemento Processador (EP) de uma arquitetura do tipo SIMD, que atenda eficientemente a aplicações que envolvam grande quantidade de dados estruturados na forma de matrizes ou vetores, como as citados anteriormente. Ele faz parte do projeto PRIMATA (Processador de Imagens e Matrizes), em desenvolvimento no Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro. A escolha deste tipo de arquitetura se deve principalmente às aplicações visadas (as máquinas SIMD são apropriadas para aplicações que exibem estruturas de dados regulares, como matrizes) e ao custo de desenvolvimento.

Nesta arquitetura, o paralelismo advém da matriz de processadores, que é construída replicando-se um Elemento Processador simples tantas vezes quanto for necessário e possível. Deste modo, o projeto e a implementação da arquitetura em circuitos integrados são enormemente facilitados.

A arquitetura do PRIMATA não visa ser, por si só, um computador completo, mas sim atuar como um "coprocessador" de uma estação de trabalho (*host*). A comunicação entre o *host* e a matriz de processadores é executada por uma interface. Essa interface é encarregada de enviar comandos para a UC da matriz de processadores, que por sua vez providencia a execução dos mesmos.

Por se tratarem de dispositivos simples, os EPs têm um custo individual baixo. O desempenho da arquitetura depende basicamente do produto do número de EPs pela velocidade de cada EP. Sendo assim, o desempenho individual não é crítico. O importante é manter um compromisso entre o custo e o desempenho do EP, a fim de viabilizar o fator "número de EPs".

Antes de iniciar o desenvolvimento do EP este trabalho apresenta no segundo capítulo um estudo histórico e comparativo das arquiteturas SIMD mais representativas. São introduzidas as principais características e inovações de cada arquitetura.

Para desenvolver o Elemento Processador procurou-se otimizar a arquitetura para a execução das operações aritméticas mais importantes, tendo como restrições o tamanho e o custo da área de silício e o número de processadores que se pode construir. A metodologia utilizada para a otimização foi a análise das operações aritméticas mais representativas em ordem crescente de complexidade. Para isto foram escolhidas as operações de soma, subtração, multiplicação e divisão de inteiros e soma e multiplicação com ponto-flutuante.

No terceiro capítulo são analisadas as operações aritméticas sobre números inteiros. Inicialmente é proposta uma arquitetura extremamente simples. A partir da análise do desempenho desta arquitetura vão sendo propostas modificações, gerando novas arquiteturas mais evoluídas. Além de arquiteturas, também são apresentados e comparados diversos algoritmos. Como resultado da análise, ao final do capítulo é eleita uma arquitetura base, sobre a qual serão estudadas as operações aritméticas sobre números reais apresentadas no quarto capítulo. Neste capítulo são sugeridas algumas modificações na arquitetura base, visando melhorar o seu desempenho nas operações sobre reais.

No quinto capítulo são definidos os mecanismos de comunicação entre EPs e entre os EPs e a Unidade de Controle. É apresentada também a definição do microcódigo que controla a matriz de processadores. Além disto, são discutidos a simulação dos elementos processadores e algoritmos propostos e aspectos da implementação da arquitetura.

Por último, no sexto capítulo são apresentadas as conclusões e perspectivas futuras deste trabalho.