

RELATÓRIO TÉCNICO

**ANÁLISE DOS MICROPROCESSADORES
RISC DE 32 BITS**

Gabriel Pereira da Silva

**NCE-29/90
Novembro/90**

Universidade Federal do Rio de Janeiro
Núcleo de Computação Eletrônica
Caixa Postal 2324
20001 - Rio de Janeiro - RJ
BRASIL



**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA**

ANÁLISE DOS MICROPROCESSADORES RISC DE 32 BITS

SUMÁRIO

Este trabalho faz a análise das características de alguns microprocessadores RISC de 32 bits. Este estudo foi desenvolvido para permitir a escolha do microprocessador a ser utilizado no projeto MULTIPLUS, um sistema multiprocessador em desenvolvimento no NCE/UFRJ. Através da utilização de vários microprocessadores RISC acoplados a unidades aritméticas de ponto flutuante, pretende-se obter um sistema de processamento paralelo de alto desempenho. Este relatório analisa várias arquiteturas e mostra os critérios utilizados para a escolha realizada.

32 BIT RISC MICROPROCESSORS ANALYSIS

ABSTRACT

This work presents an analysis of some 32 bit RISC microprocessors. This study was developed to help the choice of a RISC microprocessor to the MULTIPLUS project. MULTIPLUS is a shared memory multiprocessor system under development at NCE/UFRJ. Through the utilization of parallel processing with RISC microprocessors coupled to floating point arithmetic units, it is intended to obtain a high performance parallel system. This paper analyze some architectures and shows the criteria used to choose one that best fits to this kind of application.

1 INTRODUÇÃO

O projeto MULTIPLUS [1] é um sistema multiprocessador com memória global compartilhada em desenvolvimento no NCE/UFRJ. Sua proposta é obter um sistema paralelo de alto desempenho com o uso de microprocessadores RISC acoplados a unidades aritméticas de ponto flutuante. Um dos objetivos embutidos neste projeto é o desenvolvimento de um microprocessador RISC compatível com o microprocessador comercial que venha a ser utilizado. Este trabalho trata da descrição das características dos microprocessadores estudados durante o processo de escolha do microprocessador comercial.

Optou-se pela utilização de microprocessadores RISC por causa da simplicidade de sua arquitetura interna, podendo ser implementados com uma tecnologia menos sofisticada e com menor área, além de possuírem uma relação custo/desempenho bem melhor que a dos microprocessadores convencionais. Além disto, se adaptam muito bem à linguagem "C" e aos ambientes do tipo UNIX, o que os torna de especial interesse, dado que o sistema operacional PLURIX¹ deverá ser transportado para a nova máquina.

Este trabalho está organizado em 5 seções: A seção 2 apresenta as características gerais dos microprocessadores RISC. Na seção 3 são avaliados diversos microprocessadores para possível utilização no MULTIPLUS. A seção 4 descreve os critérios utilizados para a escolha de um desses microprocessadores. E, finalmente, na seção 5 são apresentadas as conclusões e perspectivas de desdobramento deste trabalho.

¹Sistema Operacional com filosofia UNIX desenvolvido no NCE/UFRJ.

2. CARACTERÍSTICAS GERAIS

Primeiramente, seria interessante ressaltar algumas características normalmente presentes nas arquiteturas RISC:

- . Uso de pipeline: várias instruções são executadas simultaneamente para atingir a taxa de 1 ciclo/instrução na maioria dos casos.
- . Uso de circuitos de controle, e não de memória de microcódigo, para a execução e sequenciamento das instruções.
- . Arquitetura orientada a registrador: todas as operações aritméticas são realizadas entre registradores, e os acessos à memória existem apenas para transferências de dados de/para os registradores.
- . Uso de formato fixo das instruções (normalmente 32 bits). Isto facilita o controle do pipeline e simplifica a decodificação.
- . Uso intensivo de registradores internos e memória cache (interna ou externa), como forma de reduzir a utilização do barramento.
- . Migração de funções para o compilador. Somente o essencial é implementado na pastilha. Consequentemente, existe uma extrema dependência do compilador-otimizador, específico para cada arquitetura, para executar as tarefas com bom desempenho.
- . Da mesma maneira que para os processadores CISC, existe a necessidade do uso de processador de ponto flutuante para a execução de operações aritméticas mais complexas.

Com o uso destas características os microprocessadores RISC conseguem ter um desempenho equivalente aos processadores do tipo CISC, com um "hardware" bem mais simples [2].

Entre as diversas arquiteturas estudadas, as seguintes serão apresentadas em mais detalhes neste relatório: o SPARC, da SUN; o R3000, da MIPS; o Am29000, da AMD; o MC88100, da Motorola; e uma breve descrição do iAPX80860, da Intel.

3. DESCRIÇÃO DAS ARQUITETURAS

3.1 - SPARC

O SPARC se originou nos microprocessadores RISC desenvolvidos em Berkeley a partir de 1981 por alunos de graduação e pós-graduação: o RISC I, RISC II e SOAR [3,4,5].

O RISC I foi o primeiro destes projetos, tendo introduzido diversos conceitos novos. Teve várias dificuldades para sua fabricação, pois apenas se iniciava a utilização de "foundries" para este fim. Por exemplo, de quarenta unidades produzidas apenas quatro funcionaram, a uma frequência de "clock" bem abaixo daquela esperada. O RISC II foi desenvolvido em dois anos, sendo um projeto mais refinado, com um nível a mais de "pipeline", mais registradores, e uma área final menor que o RISC I. Estes dois projetos foram desenvolvidos com tecnologia NMOS, apresentando as seguintes características:

RISC I	RISC II
. N° total de instr. - 31	. N° total de instr. - 31
. Tamanho de instr. - 32 bits	. Tamanho de instr. - 32 bits
. N° de estag. pipeline - 2	. N° de estag. pipeline - 3
. N° de registradores - 78	. N° de registradores - 138
. $\lambda = 2 \mu\text{m}$	. $\lambda = 1.5 \mu\text{m}$

Fig 1 - Características do RISC I e II

Não possuíam previsão para multiprocessamento, nem para coprocessador de ponto flutuante, sendo o percentual de área gasto na lógica de controle de 5% e 10%, respectivamente. O RISC I apresentava barramentos distintos para dados e instruções, enquanto que o RISC II possuía apenas um barramento compartilhado.

O SOAR tinha o mesmo conjunto básico de instruções, com a diferença de possuir instruções específicas para programação orientada a objeto, em particular para o SMALLTALK. Estas instruções servem para facilitar o gerenciamento de dados e operações sobre variáveis, já que o tipo da variável só é conhecido em tempo de execução no SMALLTALK. Possuía as seguintes características:

SOAR	
. N° total de instr.	- 20
. Tamanho de instr.	- 32 bits
. N° de estag. pipeline	- 3
. N° de registradores	- 72
. λ	= 1.5 μm

Fig 2 - Características do SOAR

Além disto, ao contrário do RISC I e II, possuía instruções para carga e armazenamento de múltiplos registradores, e não possuía endereçamento a byte, pois este tipo de dado não existe em SMALLTALK.

Entre 1984 a 1987 a SUN Microsystems definiu a "Scalable Processor Architecture" (SPARC) [6,7], que tinha como objetivo principal a portabilidade, de modo que pudesse ser adaptada a diferentes tecnologias de fabricação. Pretendia-se obter uma arquitetura simples, cuja implementação pudesse ser realizada utilizando-se tecnologias tão distintas quanto CMOS ou ECL [8,9,10]. Assim, a arquitetura foi definida de modo a permitir uma unidade de ponto flutuante independente da unidade inteira, e sem uma definição específica para a forma de implementação da cache e gerência de memória.

O SPARC herdou do RISC I e II um esquema de janelas de registradores e a execução de desvios atrasados, e do SOAR instruções para suporte a linguagem orientada a objeto. Foram introduzidas extensões para suporte a multiprocessamento, para coprocessador de ponto flutuante, e para um coprocessador genérico. Possui as seguintes características:

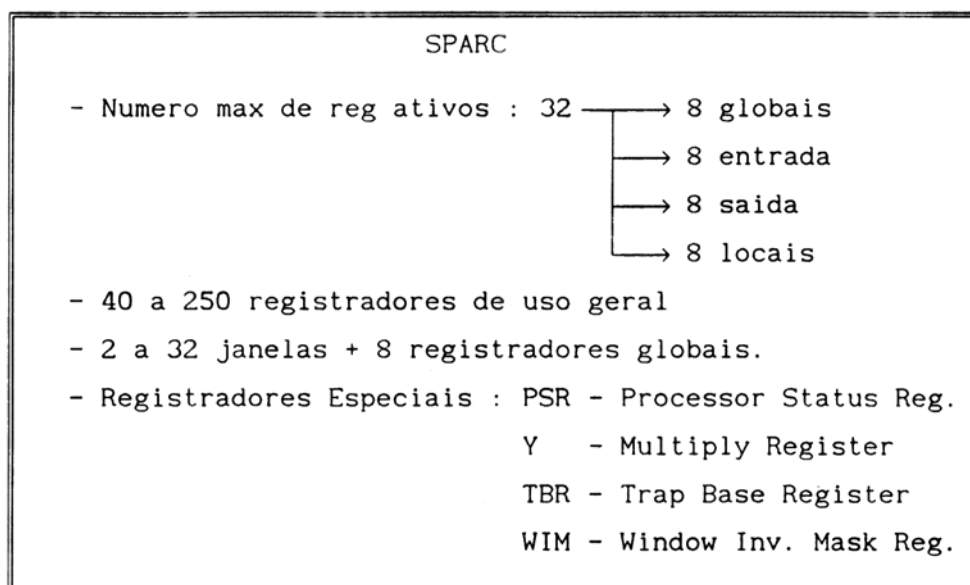


Fig. 3 - Características do SPARC

O SPARC utiliza esquema de janelas de registradores para otimização do tempo gasto com chamadas de subrotinas. Estudos [3] mostram que os tipos mais freqüentes de operandos são variáveis escalares locais, e uma forma de otimizar a utilização destes operandos é através do uso de múltiplos bancos de registradores para o armazenamento destes dados.

No esquema adotado no SPARC os conjuntos de registradores utilizados por cada rotina são sobrepostos parcialmente, permitindo que parâmetros sejam passados em registradores. A cada momento existe um conjunto de 32 registradores ativos, chamado de "Janela", que é dividido da seguinte maneira: 8 registradores globais (R0-R7); oito de entrada de parâmetros, passados pela rotina anterior (R24-R31); oito locais (R16-R23); e oito de saída de parâmetros, passados para a rotina a ser chamada (R8-R15). Quando é feita uma chamada de subrotina os registradores lógicos R8 a R31 são deslocados de 16 posições no conjunto físico de registradores. Desta maneira, os registradores de saída da rotina anterior passam a ser os registradores de entrada da rotina chamada, onde são lidos os parâmetros passados.

REGISTRADORES FÍSICOS

REGISTRADORES LÓGICOS

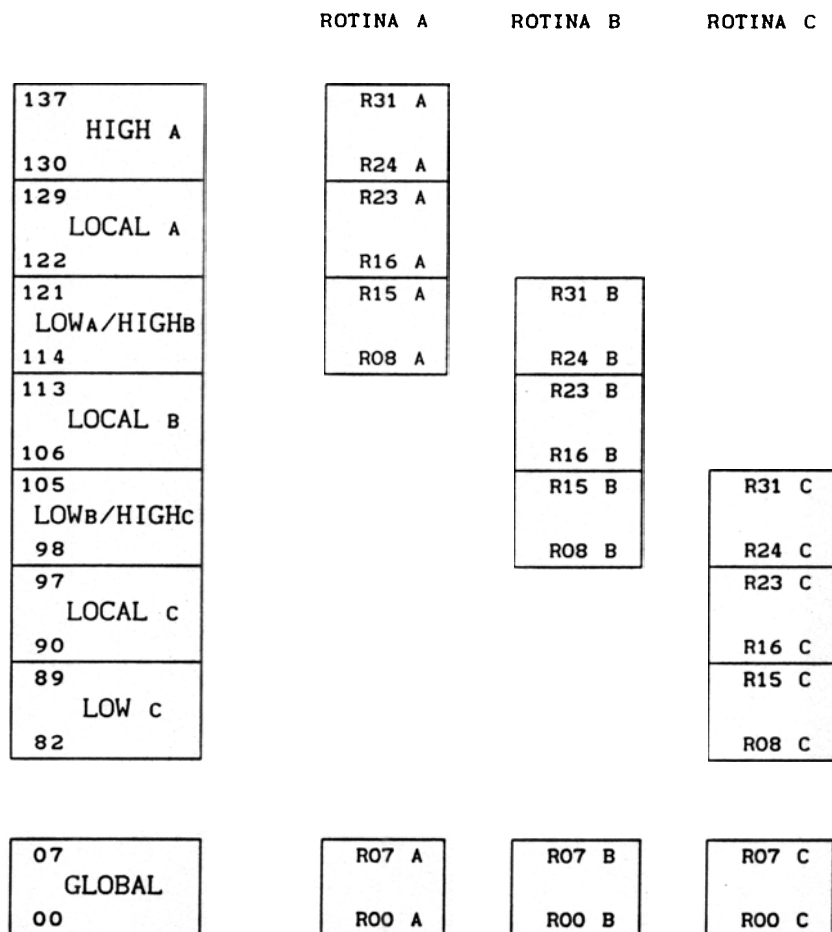


Fig. 4 - Esquema de janelas

Quando não há mais registradores físicos para serem alocados ocorre uma exceção, e o processador desvia para uma rotina de gerenciamento que salva alguns registradores na memória, deixando o espaço livre necessário para a nova rotina. Levantamentos [6] indicam um índice de 20% de acessos a dados na memória, contra 30 a 40% em microprocessadores sem uso da janela.

O uso de "pipeline" no SPARC melhora consideravelmente o desempenho da máquina, mas introduz algumas dificuldades, já que uma nova instrução é buscada em paralelo com a execução da instrução atual. Quando for executada uma instrução de desvio a instrução seguinte já terá sido buscada e estará pronta para ser executada. A solução normalmente adotada é executar a instrução que já foi buscada, caracterizando o que é chamado de "desvio

atrasado". O exemplo a seguir mostra como o aproveitamento desta instrução pode ser feito pelo compilador:

ENDEREÇO	DESVIO NORM.	DESVIO ATRAS.	DESVIO ATRAS. OTIMIZADO
100	LOAD X,A	LOAD X,A	LOAD X,A
101	ADD 1,A	ADD 1,A	JMP 105
102	JMP 105	JMP 106	ADD 1,A
103	ADD A,B	NOP	ADD A,B
104	SUB C,B	ADD A,B	SUB C,B
105	STORE A,Z	SUB C,B	STORE A,Z
106		STORE A,Z	

Fig 5 - Exemplo de Desvio Atrasado

No caso do SPARC a execução da instrução que se encontra no "delay slot" é opcional. Normalmente o compilador consegue introduzir código útil após estas instruções de uma maneira transparente ao programador, mas o fato desta execução ser opcional traz uma flexibilidade maior no aproveitamento desta instrução pelo compilador.

No momento, o SPARC está sendo licenciado para diversos fabricantes, entre os quais podemos destacar:

.CYPRESS - Realizou um projeto com tecnologia CMOS de 0.8 μ m, com desempenho estimado de 27 Mips VAX² em regime e 33 Mips de pico a 40 Mhz. Outros integrados da família são um controlador de cache e gerência de memória, e uma unidade de ponto flutuante, com desempenho estimado em 4.2 Mflops para o benchmark Linpack³ em precisão dupla a 33 MHz [11].

.FUJITSU - O seu primeiro projeto foi feito com um "gate-array" de 20000 portas, com tecnologia CMOS de 2 μ m [8]. O desempenho de pico estimado é de 10 Mips. Um novo

²Mips Vax- Medida de desempenho relativa ao número de instruções equivalentes ao VAX executadas em um segundo.

³Linpack - Benchmark para avaliação da capacidade de processamento de cálculo científico baseado no pacote LINPACK para solução de sistemas de equações lineares.

projeto com "clock" de 25 Mhz, desempenho de 15 Mips e feito com tecnologia "standard-cell" está sendo lançado. Versões futuras preveem um desempenho em torno de 40 Mips. O conjunto completo inclui gerência de memória, ponto flutuante e cache.

.LSI - Este projeto foi realizado com tecnologia CMOS de 0.9 μm , com "clock" de 25 Mhz e desempenho em torno de 15 Mips. Espera-se obter para breve versões de 40-50 Mips com uso de tecnologia BICMOS.

.BIT - Está lançando um projeto SPARC com tecnologia ECL com desempenho estimado entre 40 e 50 Mips [10]. Utilizará coprocessadores de ponto flutuante também em ECL que já constavam de sua linha de fabricação

3.2 - MIPS R3000

Assim como o SPARC, este projeto da MIPS Computer Systems teve origem em um projeto universitário: o projeto MIPS desenvolvido pelo Departamento de Engenharia Elétrica e Ciência da Computação da Universidade de Stanford [12]. O principal objetivo deste projeto foi obter alto desempenho na execução de código objeto compilado. A filosofia básica é apresentar um conjunto de instruções bem perto do que existiria a nível de microprogramação. Deste modo, o compilador fica responsável por gerar a codificação adequada para cada instrução.

O MIPS possui um "pipeline" de 5 estágios, mas é interessante observar a forma como trata os casos de dependências de dados no pipeline. Estes casos ocorrem quando uma determinada instrução deseja utilizar um registrador que foi modificado por uma instrução anterior, mas que devido ao processo inerente ao "pipeline", não foi atualizado. Desta maneira existem duas opções: o compilador garante que a próxima instrução não utiliza este registrador, ou o microprocessador congela o "pipeline" toda vez que um possível conflito deste tipo possa ocorrer. O MIPS transfere para o compilador a tarefa de lidar com as possíveis dependências de dado que surjam no "pipeline". Esta arquitetura foi batizada de "Microprocessor without Interlocked Pipeline Stages (MIPS)". Esta mudança de complexidade do "hardware" para o "software" possui as seguintes vantagens:

. A complexidade só influencia uma vez durante a compilação. Quando um usuário executa seu programa em uma arquitetura complexa, ele paga o preço deste custo cada vez que executa o programa.

. Isto permite a concentração de energias no "software", ao invés de construir uma máquina com "hardware" complexo, que é difícil de projetar, depurar e utilizar eficientemente. O "software" não é necessariamente mais fácil de construir, mas o ambiente de VLSI faz a simplicidade do "hardware" ser importante.

. O uso de "interlock por hardware" faz com que a lógica de controle do "pipeline" seja mais complexa e irregular, além de causar um pequeno atraso para todas instruções, independente de estarem ou não envolvidas no "interlock".

A primeira implementação do MIPS foi o R2000, tendo sido posteriormente atualizado para o R3000, anunciado com clock de 25 Mhz e um desempenho de 20 Mips VAX. O R3000 [13] possui ainda gerência de memória e o controle para uma cache externa integrados na mesma pastilha. O tamanho da cache de dados e instruções, pode variar desde 4K até 64 Kbytes cada.

- . ALU de 32 bits com lógica independente para multiplicação e divisão.
- . 32 registradores de 32 bits de uso geral.
- . Pipeline de 5 estágios.
- . "Translation Lookahead Buffer" com 64 posições.
- . Controle para cache de 4 até 64 Kbytes.
- . Dois barramentos internos de 32 bits bidirecionais.
- . Interface para processador de ponto flutuante.

Fig. 6 - Características do R3000

A MIPS Computer Systems também fabrica uma unidade de ponto flutuante, o R3010 que possui clock de 25 Mhz, com desempenho final em torno de 7 Mflops em precisão simples, executando o "benchmark" Linpack. Outras empresas que também fabricam o R3000 são as seguintes :

. INTEGRATED DEVICES TECHNOLOGY - Fabricado com CMOS de 1.2 μ m, possui como facilidade adicional o uso de um "buffer" de escrita. Este circuito, feito em uma pastilha separada, permite liberar o processador logo após uma escrita, enquanto que a verdadeira escrita é realizada posteriormente na memória. Com isto, é esperado um aumento de desempenho na ordem de 10% a 15% é esperado.

. PERFORMANCE SEMICONDUCTORS - Além do projeto original oferece uma cache de 256 Kbytes de dados e instruções.

Finalmente, gostaríamos de ressaltar as principais diferenças existentes entre o MIPS e o SPARC:

. O SPARC implementa "Interlock por Hardware" para as instruções de "load". O MIPS deixa esta tarefa para o compilador.

. O SPARC possui um "bit" que faz com que a execução de uma instrução após um desvio seja opcional, dando mais flexibilidade para o compilador. O MIPS sempre executa a instrução seguinte ao desvio.

. O SPARC possui uma tabela de vetores para desvio das rotinas de exceção. O MIPS possui apenas duas entradas e utiliza um registrador de "status" para verificar qual foi o tipo de exceção que ocorreu.

. O MIPS possui uma tabela interna de translação de endereços lógicos para endereços físicos. O SPARC depende de um controle de gerência externo, mas que em compensação já realiza a busca automática do descritor na tabela de páginas.

. O SPARC possui instruções de ciclo indivisível, o MIPS não possui este tipo de instrução. Este tipo de instrução é fundamental para aplicações com multiprocessadores.

. O SPARC utiliza registrador com códigos de condição. O MIPS possui instruções específicas de comparação entre registradores. Isto resulta em formas diferentes para teste da condição de desvio.

. O SPARC utiliza um esquema de janela para acesso a seus registradores, o MIPS utiliza o endereçamento direto.

3.3 - Am29000

Os processadores até agora descritos têm utilização principalmente em estações de trabalho com ambiente UNIX. Existe, entretanto, um mercado promissor que são os processadores dedicados de alto desempenho para uso em aplicações como automação industrial, processamento de imagem, robótica, controle de impressoras laser, entre outras. Dentro deste espírito, a AMD procurou realizar um projeto com características adequadas tanto ao uso em processadores dedicados, como em estações de trabalho.

O Am29000 [14] possui um desempenho em regime de 17 Mips trabalhando a 25 Mhz. Possui 192 registradores de uso geral organizados em dois grupos: 64 são globais e os 128 restantes são locais e podem ser endereçados em modo pilha. Existe ainda um esquema de proteção associado a cada banco de 16 registradores, que define a permissão de acesso aos mesmos.

No modo pilha os registradores são referenciados indiretamente por um "stack pointer", aparecendo como uma continuação da pilha existente na memória. Como existe uma acentuada localidade de referências ao topo da pilha, este esquema se mostra eficiente, reduzindo por um fator de três as referências externas quando comparado com alocação feita por compilador [15]. A pilha também permite que as chamadas de rotinas sejam agilizadas, pois não há necessidade de salvamento/recuperação de registradores se a quantidade de registradores alocados/liberados não exceder à capacidade física. Se isto ocorrer, ocorre um desvio para uma rotina do sistema operacional, que move um número adequado de registradores de/para a memória a fim de dar prosseguimento à execução.

Esta pastilha possui um "buffer" de busca antecipada de instruções e uma "cache" de endereço de desvio de 512 bytes. A lógica de busca antecipada procura buscar sempre 4 instruções em avanço. A cache de desvio tem como função evitar a perda de tempo em acessos a instruções não seqüenciais. Se a instrução de desvio estiver na cache, o desvio pode ser executado em apenas um ciclo. Esta cache está organizada em dois conjuntos

de 64 instruções, cada um dividido em 16 blocos de 4 instruções, e possui uma taxa de acerto de 60%, diminuindo consideravelmente a latência associada aos desvios.

Ainda embutida na pastilha está uma lógica de gerenciamento de memória, que consiste de uma cache de translação de endereço, e de uma lógica de proteção associada. Se o endereço desejado não estiver na cache, ou houver uma violação de proteção, é feito um desvio para uma rotina que realiza o tratamento adequado. Esta gerência é compartilhada tanto por dados como por instruções. As páginas possuem tamanhos de 1, 4 ou 8 Kbytes, e a cache de translação possui dois conjuntos com 32 entradas cada um.

O uso de uma unidade de ponto flutuante e de cache de dados é opcional, sendo disponíveis em pastilhas a parte (Am29027 e Am29062).

O conjunto de instruções possui instruções de ciclo indivisível e instruções de carga/armazenamento de múltiplos registradores. Possui ainda previsão para instruções de multiplicação e divisão, mas NÃO possui o hardware para as mesmas, que se forem executadas provocam o desvio para uma rotina de exceção. Na realidade existem implementadas apenas as instruções de "multiply step" e "divide step".

O barramento possui uma estrutura singular: existe um barramento de endereço compartilhado por dois barramentos distintos para dados e instrução, com capacidade de leitura de dados em modo rajada, com taxa de transferência de até 100 Mbytes/s. A busca de instruções é feita do seguinte modo: é colocado um endereço inicial no barramento de endereços, e a busca prossegue sequencialmente no barramento de instruções, até que haja uma instrução de desvio, seja ultrapassada uma fronteira de página, ou ocorra uma exceção de memória. Assim, o barramento de endereços pode ficar livre para endereçar as operações de leitura/escrita dos dados.

3.4 - MC88000

A MOTOROLA, apesar de já possuir uma bem sucedida linha CISC, resolveu também investir em um projeto do tipo RISC e lançou a linha 88000. O MC88000 é um conjunto composto por três pastilhas: duas MC88200 e uma MC88100. O MC88100 é um processador com barramentos separados para dados e

instruções, e o MC88200 é uma pastilha que integra um controlador de cache e uma gerência de memória. É necessário o uso de uma pastilha MC88200 para cada barramento. A descrição que segue refere-se apenas ao MC88100 [16].

O processador possui integrados na mesma pastilha uma Unidade Inteira e uma Unidade de Ponto Flutuante, padrão IEEE-754. Existe um conjunto de 32 registradores, referenciados através de um esquema de "scoreboarding" para otimizar o desempenho. Esta técnica introduzida pelo CLIPPER [17], permite o controle quando existe mais de uma unidade funcional na mesma pastilha. Esta técnica permite ainda a execução simultânea de instruções de LOAD com outras instruções. Quando a instrução de LOAD é iniciada, um bit é ligado em um registrador especial. Quando a operação de memória realmente termina o bit é desligado. Neste intervalo, qualquer referência a um registrador marcado faz o processamento parar até que a operação de carga tenha terminado. Um caso típico de interlock é mostrado a seguir

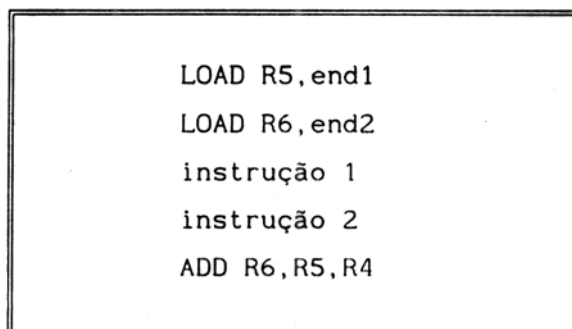


Fig 7 - Exemplo de Interlock

As instruções 1 e 2 são executadas em paralelo com as instruções de LOAD, sem nenhum gasto adicional de tempo. A instrução de ADD só será executada quando os operandos estiverem disponíveis nos registradores. De qualquer maneira, a ocorrência destas dependências de dado atrasa o processamento, devendo ser evitada através de um sequenciamento adequado das instruções em tempo de compilação.

O conjunto de instruções do MC88100 é o maior dos microprocessadores estudados, possuindo até instruções para manipulação de bits. Embora todo controle esteja implementado em hardware, isto foge um pouco ao conceito original de arquiteturas RISC, pois exige uma fatia maior

do espaço disponível na pastilha para este controle. Possui suporte para uso em multiprocessamento, inclusive com esquema para processamento tolerante a falhas, com processadores redundantes para verificação do funcionamento do processador principal. O seu conjunto de instruções foi otimizado para as instruções mais utilizadas por um compilador C.

Possui instruções específicas para multiplicação e divisão, que são executadas pela unidade de ponto flutuante, que possui somador e multiplicador independentes. Por isto, possui um alto grau de paralelismo na execução das instruções. Para exemplificar, em apenas um único ciclo podemos ter até cinco adições de ponto flutuante, quatro multiplicações de inteiros, três acessos à memória, e duas buscas de instrução em andamento. Como consequência possui um alto desempenho, mas o tratamento dos casos de exceção é bastante complexo, necessitando de um grande número de registradores auxiliares para a recuperação de contexto.

O seu desempenho é estimado em torno de 15 Mips VAX a 20 Mhz. A unidade integrada de ponto flutuante possui um desempenho estimado de 7 Mflops. Embora não estejam implementados, a arquitetura tem previsão para a integração de unidades especiais, tais como processadores vetoriais e gráficos. Atualmente, a MOTOROLA trabalha em conjunto com a DATA GENERAL para a produção de uma versão ECL.

3.5- IAPX80860

É uma arquitetura bastante interessante mas, devido à alta escala de integração requerida, não foi considerada para utilização em nosso projeto. A seguir é feita uma breve descrição de suas características

O IAPX 80860 é um RISC de alto desempenho com suporte para aplicações gráficas. Este processador possui integrados uma cache de dados e instruções com 12 Kbytes, gerência de memória e processador de ponto flutuante, além do processador gráfico. Na versão de 33 Mhz o seu desempenho estimado é de 33 Mips VAX para o processador inteiro, e de 10 Mflops para a unidade de ponto flutuante, executando o benchmark Linpack. Estas duas unidades podem operar em paralelo com a unidade de processamento gráfico, resultando em um desempenho significativo. Isto se torna mais relevante na medida que consideramos seu custo, que é bastante baixo se

comparado a outros sistemas com capacidade similar

Outras características marcantes são o uso de barramentos separados para dados e instruções, ambos com 64 bits de largura; um conjunto de 32 registradores de 32 bits para processamento inteiro, com três portas de acesso; um conjunto de registradores de ponto flutuante com 5 portas de acesso, que pode ser configurado com 8×128 , 16×64 ou 32×32 bits.

Estes registradores são gerenciados por um esquema de "scoreboarding". Aliado a isto, um esquema sofisticado de predição de desvios permite à máquina alcançar o desempenho de um ciclo por instrução na quase totalidade de seu conjunto de instruções.

4 - ARQUITETURA ESCOLHIDA

Foram determinados alguns requisitos básicos que a arquitetura ser utilizada deveria atender

- Simplicidade de código: Um conjunto de instruções mais extenso implica em um controle mais complexo, que poderia inviabilizar a implementação de um protótipo pelo tamanho final que o circuito teria com a tecnologia de implementação disponível.

Possibilidade de implementação de um subconjunto de registradores: Caso não seja possível implementar toda a arquitetura, a organização dos registradores deve permitir a utilização de um subconjunto dos registradores para tornar a implementação viável. Neste caso, a compatibilidade é mantida emulando-se os registradores em memória.

- Suporte para multiprocessamento: Atualmente, o melhor caminho para obtermos um sistema de alto desempenho é através da utilização de processamento paralelo, ou seja, dividir o programa em várias tarefas, cada uma delas executada por um microprocessador distinto, conseguindo-se um tempo total de execução muito menor. Para que isto seja possível, cada processador deve possuir instruções de sincronização para permitir a utilização concorrente dos recursos do sistema.

- A existência de uma forma simples de "interlock" por "hardware" e o uso do desvio atrasado opcional facilitariam a implementação do compilador. Isto implica em um gasto adicional de lógica de controle, e somente após uma melhor avaliação da relação custo/benefício deveremos definir alguma opção.

- Alguns pontos que não estão definidos são o tipo de barramento a ser utilizado, o uso ou não de cache interna, e qual o seu tipo. Assim, seria desejável que a arquitetura escolhida fosse o mais flexível possível em relação a estes itens.

Com base no exposto acima, a equipe de projeto optou pela utilização do SPARC pelos seguintes motivos:

. Apresenta um dos menores conjuntos de instrução, com apenas 68 instruções. Os demais microprocessadores apresentam tamanhos que variam entre 75 e 210. Foi um fator de peso na decisão final, pois o gasto de área do controle é proporcional ao número de instruções.

. O tamanho do conjunto de registradores não é fixo, podendo variar entre 40 e 520. Possui um esquema de janelas, que permite a emulação de registradores virtuais em memória. Nos demais microprocessadores estudados o tamanho do conjunto de registradores é fixo, e apenas o Am29000 apresenta uma organização do tipo pilha para seus registradores, mas que não é muito flexível.

. Possui suporte para multiprocessamento através de instruções de ciclo indivisível (SWAP, LDSTUB). Em alguns microprocessadores, como o MC88100 e Am29000, existe o mesmo tipo de suporte, mas no MIPS é simplesmente inexistente. Este foi um fator decisivo para a não utilização do MIPS.

. O SPARC possui um "annul bit" que faz com que a execução de uma instrução após um desvio seja opcional. Isto dá mais flexibilidade para o compilador.

. Implementa uma forma simples de interlock por hardware. Apesar disto, se for desejável, pode-se implementar este controle no compilador e, com o uso de filtros, manter a compatibilidade entre as versões com e sem "interlock".

. O SPARC possui uma tabela de vetores para desvio das rotinas de exceção, o que dá mais agilidade no tratamento de exceções.

. Não possui uma gerência de memória interna de translação de endereços, mas uma pastilha externa que já realiza a busca automática do descritor na tabela de páginas. O Motorola MC88100 utiliza o mesmo esquema, mas o MIPS R3000 e o AMD Am29000 possuem uma cache de endereços interna à pastilha, o que consome muito espaço, e obriga o processador a executar uma rotina para busca e carga dos descritores nesta cache.

. Possui implementações com desempenho de 33 Mips a 40 Mhz, e uma taxa de execução em torno de 1.5 ciclos para cada instrução. Em outros processadores com arquitetura mais sofisticada, esta taxa não é menor que 1.3.

. É uma arquitetura aberta, sendo produzida por diversos fabricantes, permitindo inclusive o uso de tecnologias diferentes, como CMOS ou ECL. Detalhes como a largura do barramento e a existência ou não de cache interna são deixados a cargo de cada implementação.

. Possui uma interface para a unidade de ponto flutuante que permite acoplar o processador que for escolhido pelo usuário, oferecendo maior flexibilidade para uso de um processador de ponto flutuante de maior desempenho.

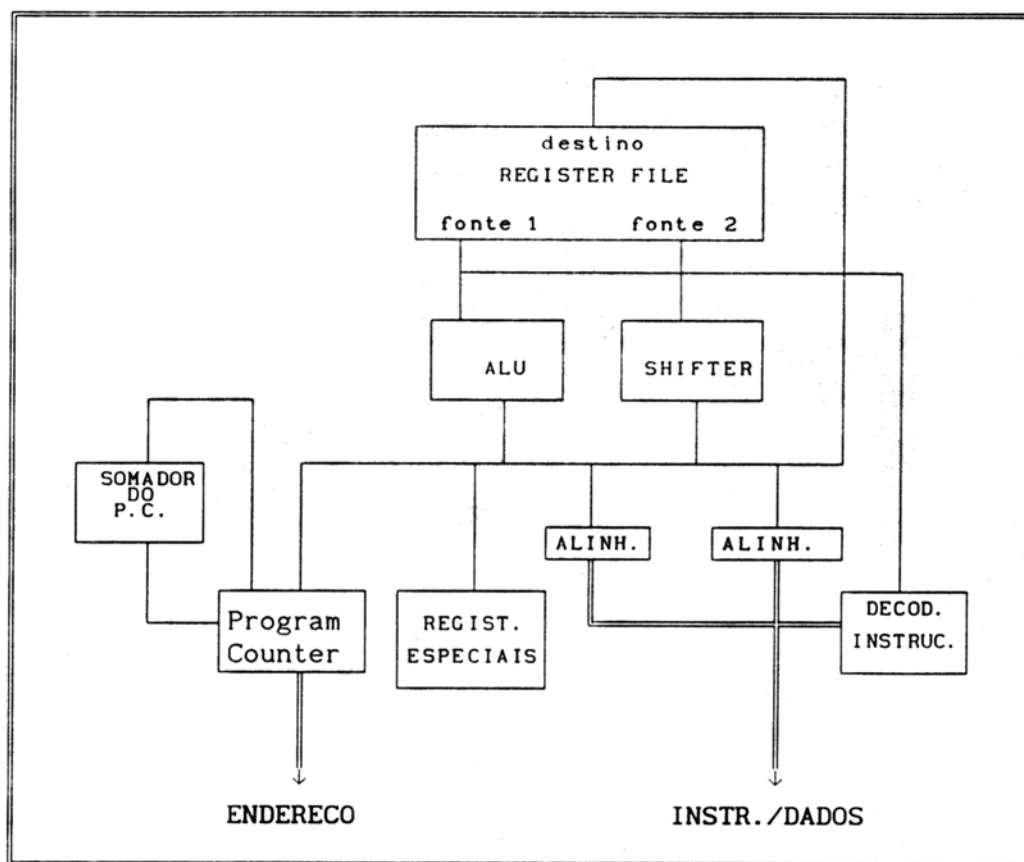


Fig 8 - Arquitetura do SPARC

5 - CONCLUSÃO E PERSPECTIVAS

Sem dúvida os microprocessadores RISC tiveram uma grande evolução nesta década, e hoje se destacam por conciliarem desempenho e baixo custo. O seu uso em sistemas multiprocesssores certamente é uma opção para obter uma alta capacidade de processamento, sem o custo e complexidade inerentes aos microprocessadores CISC e sistemas de alto desempenho convencionais.

Neste relatório apresentamos vários microprocessadores que foram considerados para utilização no sistema multiprocessador MULTIPLUS. Todas estas arquiteturas foram cuidadosamente estudadas, de forma a garantir o uso de um microprocessador que não apenas apresentasse desempenho, mas que também pudesse ter uma arquitetura compatível implementada, garantindo um maior domínio tecnológico sobre o projeto final do sistema

multiprocessador. Acreditamos que a escolha do SPARC seja um compromisso entre estes dois fatores, e que venha a atender plenamente aos requisitos necessários a este tipo de operação

A escolha até o momento se mostrou acertada, pois conseguimos elaborar uma definição inicial de arquitetura compatível com a definição SPARC e estamos iniciando o projeto lógico da arquitetura. Estão sendo elaborados simuladores funcionais para a arquitetura, a fim de verificar a qualidade e a funcionalidade da arquitetura proposta. A etapa seguinte a estas fases será a produção de um protótipo para testes que deverá ser elaborado com tecnologia CMOS de 1.2 micron.

6 - Bibliografia

- 1 - Aude, J. S. et alli "MULTIPLUS: Um Multiprocessador de Alto Desempenho" Anais do X Congresso da SBC, Vitoria, Julho de 1990
- 2 - Gimarc, C.E. & Milutinovic, U. "A Survey of RISC Processors and Computers of the MID 1980s." IEEE Computer, 20(9):59-69, Sep 1987
- 3 - Patterson D.A. & Sequin C.H. "RISC I: A Reduced Instruction Set VLSI Computer" ACM SIGARCH Newsletter, New York, NY, 9(3):443-57, May 1981
- 4 - Katevenis. M.G.H." "The RISC-II Micro-Architecture" Proceedings of the VLSI'83- IFIP 1983 - Elsevier Science Publishers, North-Holland, pp 349-355, 1983
- 5 - Ungar, D. et alli "Architecture of SOAR: Smalltalk on a RISC" Proceedings of the 1th International Symposium on Computer Architecture, pp 158-166, 1984
- 6 - Garner, R.B. et alli "The Scalable Processor Architecture " Proceedings of the IEEE COMPCON 88, IEEE, New York, NY, pp 278-283, 1988.
- 7 - Sun Microsystems Inc "The SPARC Architecture Manual", Mountain View CA 1987, 199 pp.
- 8 - Namjoo, M. et alli "CMOS Gate Array Implementation of the SPARC Architecture" Proceedings of the IEEE COMPCON 88, New York, IEEE, NY, pp 10-13 , 1988.
- 9 - Namjoo, M. et alli "CMOS Custom Implementation of the SPARC Architecture" Proceedings of the IEEE COMPCON 88, New York NY, IEEE, pp 18-20, 1988.
- 10- Agrawal, A. et alli "Design Considerations for a Bipolar Implementation of SPARC" Proceedings of the IEEE COMPCON 88, New York, NY, IEEE, pp 6-9, 1988.
- 11- Cypress Semiconductor "SPARC RISC USER'S GUIDE", Cypress, 2nd Edition, Feb 1990, 416 pp

- 12- Henessy, J. et alli "MIPS: A Microprocessor Architecture" ACM SIGMICRO NEWSLETTER, New York, NY, 13(4):17-2, Dec 1982
- 13- Kane, G. "MIPS Risc Architecture" Prentice-Hall, Englewood Cliffs, NJ, 1988, 334 pp
- 14- AMD "AM29000 Streamlined Instruction Processor". Johnson M, ed., Advanced Micro Devices, Sunnyvale, 1988, 440 pp
- 15- Berenbaum, A.D. et alli "CRISP: A pipelined 32-bit RISC microprocessor with 13 K-bit of Cache Memory" IEEE Journal of Solid State Circuits, New York, NY, SC-22(5):776-782, Oct 1987
- 16- MOTOROLA "MC88100 Reduced Instruction Set Computer Users Manual." Motorola, Austin, Texas, 1988, 235 pp.
- 17- Fairchild Semicondutores "CLIPPER MODULE 32-bit Microprocessor." Fairchild, 1986, 94 pp.

RESUMO DAS CONFIGURAÇÕES

MIPS :

- R3000 - Unidade de Processamento Inteiro, com Gerência de Memória e controle para uma cache externa.
- R3010 - Unidade de Ponto Flutuante

SPARC (CYPRESS) :

- CY7C601 - Unidade de Processamento Inteiro
- CY7C602 - Unidade de Ponto Flutuante
- CY7C604 - Unidade de Controle de Cache e Gerência de memória.
Exige pastilhas externas de tag e memória para a cache.

AMD :

- AM29000 - Unidade de Processamento Inteiro e Unidade de Gerência de Memória
- AM29027 - Unidade de Ponto Flutuante
- AM29062 - Unidade de Controle de Cache, inclui RAM de 8 Kbytes. É necessário uma para cada barramento.

MOTOROLA :

MC88100 - Unidade de Processamento Inteiro e Ponto Flutuante.

MC88200 - Unidade de Gerência de Memória e Controle da Cache. Possui 16 Kbytes de RAM. Deve haver uma para cada barramento.

INTEL

I80860 - Unidade de Processamento Inteiro, Unidade de Ponto Flutuante, Unidade de Processamento Gráfico, Gerência de Memória e Cache de 512 bytes.

TABELA COMPARATIVA

FABR.	MAQUINA	VELOC.	INST/CICLO	MIPS
MIPS	R2000	16.7	1.4	12.0 ¹
MIPS	R3000	25.0	1.3	20.0 ¹
CYPRESS	SPARC	40.0	1.6	27.0 ¹
AMD	29000	25.0	1.5	17.0 ³
INTEL	80860	33.0	1.0	33.0 ²
INTEL	80386	24.0	4.5	5.3 ³
MOTO	88100	20.0	1.3	15.0 ¹
MOTO	68020	25.0	5.0	5.0 ³

1 - MIPS VAX em regime de funcionamento

2 - MIPS VAX de pico.

3 - MIPS em regime.