

Universidade Católica de Brasília

Faculdade de Tecnologia (FT)

Departamento de Tecnologia

Curso: Ciência da Computação / Processamento de Dados

Disciplina: Organização de Computadores II

Prof^a: Suely

RISC versus CISC

Universidade Católica de Brasília

Faculdade de Tecnologia (FT)

Departamento de Tecnologia

Curso: Ciência da Computação / Processamento de Dados

Disciplina: Organização de Computadores II

Prof^a.: Suely

RISC versus CISC

Euclides Pereira do Lago Júnior

96/2607-7

Marcelo Cândido Brito

96/2394-9

Marcondes José de Freitas Bastos

96/2396-5

Renato Toledo da Silva

96/2602-6

ÍNDICE

| | |
|--|----|
| 1. Introdução | 4 |
| 2. CISC e RISC | |
| Qual a diferença? | 6 |
| 2.1. Diminui a diferença | 7 |
| 3. Plataformas CISC e RISC | |
| Uma coexistência pacífica | 8 |
| 3.1. Arquitetura CISC versus RISC | 9 |
| 4. Arquitetura de Pipeline | 14 |
| 5. Superpipelined versus Superescalar | 18 |
| 6. Software | |
| Uma barreira | 23 |
| 7. Tendências Recentes das Plataformas RISC e CISC | 25 |
| 8. A Arquitetura do CISC Intel x86 | 26 |
| 9. Os Processadores RISC e CISC da Motorola | 27 |
| 10. Os Chips SPARC da SUN Microsystems | 29 |
| 11. Os RISC PA e PRISMA da Hewlett-Packard | 31 |
| 12. O Consórcio ACE e o RISC da MIPS | 33 |
| 13. Uma Pequena História dos Processadores | 34 |
| 13.1. Vantagens e Desvantagens | 36 |
| 14. Conclusão | 39 |
| 15. Bibliografia | 42 |

1. Introdução

A arquitetura RISC foi tratada de modo abrangente e agressivo no final da década de 80. Seus defensores diziam que os microprocessadores com um conjunto reduzido de instruções logo apareceriam dentro dos processadores Intel x86. Até agora, a RISC ainda não colocou os pés seguramente no mercado de computadores pessoais. Há alguma razão para prever mudanças em breve? A resposta é sim e não, mas principalmente não.

Primeiro, os chamados microprocessadores RISC não apresenta, de fato, grandes benefícios na performance se comparados com os microprocessadores denominados CISC (Complex Instruction Set Computing). Essas vantagens teóricas foram muito ressaltadas a princípio. Um conjunto de instruções menor e mais rápido, por exemplo, também requer que mais instruções sejam executadas. Além disso, estratégias como conjuntos múltiplos de registradores e pipelining (encadeamento) não estão diretamente relacionadas a conjuntos reduzidos de instruções. Os projetos tradicionais dos processadores x86 poderiam empregar as mesmas estratégias.

Na verdade, os tais microprocessadores RISC não chegam perto dos projetos verdadeiramente RISC. Eles possuem conjuntos de instruções relativamente grandes e complexos, se comparados com conceitos RISC "puros". Eles seriam descritos mais precisamente com a expressão *influenciado pela arquitetura RISC*. O resultado prático de todos esses fatos é que os processadores x86 da Intel e de um número cada vez maior de concorrentes, como a Advanced Micro Devices (AMD) e a Cyrix, conseguiram garantir sua posição contra os novos projetos influenciados pela arquitetura RISC.

A segunda razão é a falta de compatibilidade com versões anteriores. Os usuários simplesmente não querem abrir mão de tudo que já possuem.

O sucesso do Windows 3.0 da Microsoft e do OS/2 2.0 ilustra bem essa idéia. Nem o Windows 1.x ou 2.x e nem o OS/2 1.x foram bem sucedidos. As primeiras versões de ambos os pacotes sofriam do mesmo mal: incapacidade para rodar aplicativos que os usuários já possuíam. Mas conseguiram chegar ao sucesso com a introdução de uma versão capaz de rodar também aplicativos antigos.

A arquitetura RISC possui o mesmo problema fundamental que o Windows e o OS/2 apresentavam: os sistemas não rodam de modo adequado os programas que os usuários já possuem. Mas a compatibilidade com versões anteriores existente no Windows e no OS/2 poderia ser corrigida. Os sistemas baseados em RISC nunca serão capazes de rodar software desenvolvidos para a arquitetura x86 tão bem quanto os sistemas para os quais esses aplicativos foram originalmente projetados.

Isso significa que os microprocessadores influenciados pela arquitetura RISC nunca poderão desafiar com sucesso o x86 no campo dos PCs de mesa? Não. A medida que as ferramentas baseadas em objetos e o desenvolvimento voltado para plataformas diferentes tornam-se mais comuns, fica mais fácil oferecer versões de software para a arquitetura RISC, podendo esta desafiar com sucesso a arquitetura x86 no campo dos PCs de mesa.

2. CISC E RISC: Qual a diferença?

A tecnologia RISC (Reduced Instruction Set Computing) foi desenvolvida pela IBM nos anos 70 e o primeiro *chip* surgiu em 1980. Sua proposta baseou-se em um conjunto reduzido de instruções, sendo definidas apenas as mais freqüentemente utilizadas e se evitando o uso de microcódigos. As instruções também seriam simplificadas, trabalhando com apenas um operando.

As operações enfatizavam o uso de registradores, sendo o acesso à memória limitado a instruções tipo *load/store*. Assim, o processador *g* gastaria apenas um ciclo por instrução. Porém, o que sucedeu não foi tão simples assim, pois havia muita dificuldade em se escrever programas complexos utilizando um conjunto muito reduzido de instruções. Então este conjunto foi incrementado com novas instruções, como as necessárias para trabalhar com memória virtual, multiprocessamento e assim por diante.

A tecnologia RISC começou a ser promovida no mercado com o surgimento das estações de trabalho científicas, pois sua atividade básica é "*CPU bound*". Os *chips* CISC (*Complex Instruction Set Computing*) de aplicação mais geral - típicas de ambientes comerciais - não ofereciam a velocidade necessária aos trabalhos com extensas manipulação de números e visualização gráfica. Em ambiente comercial, por seu lado, é necessário considerar todo o conjunto que compõe o sistema, como CPU, memória, velocidade de discos, sistema operacional e software de aplicação.

Uma comparação apenas a nível de processador e sua técnica não é correta. Muitos dos fatores que aumentam a velocidade de um processador RISC, não são inerentes a esta tecnologia (como uso de cache, *pipeline* de instruções e grande número de registradores na CPU), sendo que estes recursos estão disponíveis a qualquer projeto de computador, sendo usados também em máquinas CISC.

A análise dos processadores do mercado mostra que nos aspectos de *mips* ou operações aritméticas com números inteiros (*SPEC Integer*), o desempenho dos *chips* RISC e CISC são similares; em ponto flutuante (*SPEC Floating Point*) os RISC tendem a apresentar resultados melhores, embora o Pentium lhes seja equivalente.

2.1. Diminui a diferença

Cada vez mais as tecnologias RISC e CISC estão se aproximando: processadores RISC estão aumentando seu conjunto de instruções e os CISC estão adotando técnicas originalmente implementadas nos RISC. Por exemplo, o número de ciclos por instrução é bastante similar em ambos. O resultado prático é que o "*path length*" destes processadores para executar uma tarefa é praticamente igual.

As razões para isso são simples. Em aplicações comerciais, a CPU trabalha cerca de 30% do tempo com instruções de movimentação de cadeias de caracteres de um lugar a outro na memória; em programação Cobol é interessante dispor de instruções de aritmética decimal e trabalhar com operandos "não alinhados". Desta forma, será necessário adicionar-se instruções tipicamente CISC para trabalhar adequadamente com estas tarefas.

O Pentium, segundo a Intel, utiliza tecnologia CRISC, acoplamento das duas técnicas. Na verdade algumas máquinas RISC utilizam poucas de suas técnicas básicas, ao mesmo tempo que implementam conceitos típicos dos projetos CISC; entretanto, adotam o jargão RISC por questão de marketing.

Um ambiente comercial é caracterizado por processamento transacional, com muita manipulação de cadeias e inteiros além de alto fluxo de entrada/saída. Já o científico se caracteriza por grande número de instruções de ponto flutuante. A conclusão lógica é que as máquinas dos dois tipos tem diferentes concepções para diferentes utilizações.

3. Plataformas CISC e RISC: Uma coexistência pacífica

Agora que se estreitou a diferença entre o desempenho das arquiteturas RISC e CISC, a seleção de uma plataforma depende de fatores como viabilidade de custos, arquitetura do *bus*, suporte a periféricos, ambiente operacional, suporte a softwares e custos de manutenção. A partir de agora examinaremos as razões da convergência destas tecnologias de processadores, analisando as tendências atuais e identificando quais plataformas específicas irão sobreviver em uma época de aumento de competitividade e redução da diferenciação de produtos a nível do processador.

As workstations e PCs mais antigos baseavam-se em arquiteturas de processadores CISC (complex instruction set computing). Porém, nos últimos anos o mercado tornou-se estratificado. Workstations avançadas baseadas em processadores RISC (reduced instruction set computing) ganharam grande espaço por oferecerem desempenho superior. Neste meio tempo, as plataformas baseadas em CISC (especialmente aquelas baseadas nos microprocessadores Intel x86 e na série 68000 da Motorola) continuaram a dominar as faixas de médio e baixo desempenho devido aos baixos custos e disponibilidade de uma extensa base de aplicativos.

Mais recentemente, houve um aumento do desempenho da plataforma CISC, à medida que os projetistas lançaram mão de algumas estratégias RISC, tais como *pipelining*, *on-chip cache* (cache no chip) e *wide-instruction buses* (barramentos de largas instruções) e desde que se tornaram disponíveis os sistemas operacionais de 32 bits e compiladores otimizados para este tipo de plataforma. Ao mesmo tempo, os custos das plataformas RISC caíram devido ao aumento da competição no mercado, padronização, disponibilidade de várias fontes para

alguns chips, as economias de escala viabilizadas pelos grandes mercados e avanços na tecnologia de semicondutores.

Hoje, porém, as considerações na escolha entre as plataformas RISC e CISC não diferem muito. Com exceção dos níveis extremos de desempenho para cima e para baixo, a escolha de uma plataforma depende hoje de fatores tais como : ter um custo viável, arquitetura de *bus*, suporte a periféricos, ambiente operacional, suporte a softwares e custos de manutenção. Com tanta sobreposição entre a capacidade (e custos) das plataformas RISC e CISC, a arquitetura do processador deixa de ser uma consideração importante para o usuário final.

Nessa próxima parte examinaremos as razões da convergência da tecnologia de processadores e analisaremos as tendências mais recentes, a fim de identificar quais plataformas específicas têm chances de sobreviver em uma época de crescente competição e redução da diferenciação de produtos a nível do processador.

3.1 Arquitetura CISC versus RISC

As plataformas baseadas em CISC (como as que utilizam os microprocessadores x86 da Intel e a série 68000 da Motorola) ainda dominam o grupo de microcomputadores e superam as workstations RISC. Isto é devido às plataformas CISC custarem menos e oferecerem uma base muito maior de suporte de softwares. O número de pacotes de software para os PCs da IBM e seus compatíveis supera aqueles para as workstations UNIX mais utilizadas, e tem também um custo muito menor, devido às economias de escala e ao aumento da competição.

Comparado aos processadores RISC, os chips CISC são caracterizados por possuir conjuntos maiores de instruções (de algumas dúzias a algumas centenas de instruções). Se

considerarmos também os conjuntos de instruções dos processadores mais recentes e de maior potência da mesma família (por exemplo o Pentium 200 MHz da Intel versus o 88088 da Motorola) verificaremos que os mesmos são superconjuntos se comparados a seus antecessores. Isto significa que softwares aplicativos mais antigos geralmente operarão em equipamentos que utilizem processadores da mesma família. Um novo software pode ser criado para que utilize ou o conjunto de instruções do processador mais antigo (8088 no caso dos PCs IBM e seus compatíveis) ou *defaults* do mesmo se um conjunto de instruções de mais alto nível não estiver disponível.

Porém, existe uma tendência de criarem softwares que não operam em toda a base de PCs instalada. Isto porque os equipamentos mais antigos não possuem os recursos de sistema (por exemplo, resolução do monitor e capacidade de memória) necessários para rodar os atuais softwares gráficos ou orientados para redes. Mesmo quando os recursos destes equipamentos são adequados, a exigência do software para lidar com todas as diferentes configurações de equipamento e com o conjunto de instruções necessariamente diminuiria o seu desempenho. Esta tendência deve continuar com a emergência dos sistemas operacionais realmente de 32 bits que podem explorar as potentes instruções de 64 bits dos mais novos processadores CISC.

Nos processadores CISC, a lógica para a maioria das instruções, com exceção das mais primitivas (shift, add, etc...), é guardada em microcódigo no processador, enquanto que para o processador RISC, é fornecida por software. Considere, por exemplo, uma operação de multiplicação. A maioria dos chips CISC possui como parte de seu conjunto de instruções uma instrução de multiplicação (por exemplo, MUL ou IMUL para a série x86). Porém um chip RISC pode não possuir esta operação de multiplicação, devendo a mesma ser feita por uma seqüência de instruções primárias como "adds" e "shifts".

Os microprocessadores RISC (como aqueles utilizados nas workstations da Sun Microsystems) são baseados em um pequeno conjunto de instruções bastante básicas. A maioria das instruções de alto nível que estão embutidas nos processadores CISC teriam que ser escritas como algoritmos de software. Mas os processadores RISC possuem a vantagem

que com seus conjuntos de instruções menores possibilitam a decodificação e execução das instruções muito mais rapidamente do que seria possível ao processador CISC. Todas as instruções RISC também geralmente possuem palavras do mesmo tamanho e estão restritas a dois operandos, enquanto que o conjunto de instruções do processador CISC normalmente inclui instruções de 8, 16 e 32 bits e muitas operações necessitam de mais de dois operandos.

Ainda mais uma vantagem dos processadores RISC é que suas instruções pegam seus operandos somente dos registradores. Os operandos são transferidos da memória para os registradores embutidos no processador com instruções separadas. Estas instruções não fazem nada mais que carregar os registradores e guardar seu conteúdo na memória. Por esta razão, muitas vezes os processadores RISC são conhecidos como “processadores load/store”. Esta característica simplifica a lógica de controle do processador e elimina muito do *overhead* e da complexidade associada aos processadores CISC.

Um microprocessador consiste em uma unidade aritmética e lógica, registradores multiuso e registradores dedicados tais como o contador do programa. Porém, necessita também de um tipo de unidade de controle para decodificar as instruções e direcioná-las para o caminho a ser percorrido, afim de que o processamento ocorra de acordo com as instruções. Em um processador CISC, a unidade de controle tem que decodificar tantas instruções e realizar tantas operações que normalmente ela é um outro processador. Os processadores RISC por outro lado possuem conjuntos de operações muito mais simples e podem geralmente utilizar um controlador HARDWIRED menos complexo.

Um ponto controverso no debate entre CISC x RISC envolve o desenho do compilador. Os que advogam para o CISC argumentam que a disponibilidade de instruções de alto nível e a habilidade de especificar múltiplos operandos baseados na memória simplifica o desenho do compilador. Porém, a questão mais importante é qual é a arquitetura que melhora o desenho do compilador (permite que seja feito um compilador otimizador mais efetivo, que por sua vez, irá produzir códigos de execução mais rápidos).

Aparentemente o acesso do software a funções mais primárias embutidas no RISC (por exemplo, instruções de acesso a memória separadas) mais a padronização das instruções RISC (com um comprimento de palavra padrão, menos variáveis e menos instruções) facilitaria o desenho de um compilador otimizador. Como verificaremos mais tarde, estas vantagens do RISC se tornarão bastante significativas com o crescente paralelismo onde o software deve ser otimizado (pela reordenação das operações se necessário) para lidar com vários processadores executando instruções simultaneamente em um mesmo chip.

Teoricamente nem os processadores CISC nem os RISC possuem uma vantagem de desempenho inerente ou superior a nível do sistema. Essencialmente existe um compromisso entre software e hardware, com o CISC desempenhando mais operações no hardware e o RISC sendo mais intensivo no software. Portanto, a relativa atração a uma das abordagens vai depender do progresso tecnológico em curso para hardware e software.

Tanto o CISC como o RISC são casos especiais de arquitetura na qual os operandos são guardados em registradores multiuso, permitindo rápido acesso a registradores específicos. Os processadores mais antigos utilizavam arquiteturas baseadas em pilha (*stack*) e acumulador. Por exemplo, o 6502 (utilizado no computador Apple II) possuía uma arquitetura baseada em acumulador, assim como vários chips de calculadoras de mão. As arquiteturas *stack* eram largamente utilizadas em aplicações (tais como aquelas baseadas na linguagem de programação Forth) onde os operandos eram acessados em uma seqüência já fixa e o primeiro operando a ser acessado era o último a ser colocado no *stack*.

Ambas as arquiteturas RISC e CISC não são novas. Ambas eram utilizadas em *mainframes* e minicomputadores antes de serem introduzidas nas plataformas baseadas em microprocessadores. A arquitetura RISC foi desenvolvida em 1983 pela Hewlett - Packard, mas era originalmente uma implementação em múltiplos chips para superminicomputadores. Os chips RISC foram introduzidos nas workstations técnicas UNIX no final dos anos 80.

O progresso da tecnologia de semicondutores tornou a arquitetura RISC plausível e de custo viável para as workstations. Uma litografia mais densa nos chips (por exemplo, mais

transistores por chips) e *wafers* mais largos (mais chips por *wafers*) aumentaram o desempenho e a capacidade, enquanto abaixavam os custos dos chips de memória: e a arquitetura RISC é intensiva em memória (como sugerido na discussão acima sobre instruções separadas para buscas na memória). Acessos lentos à memória se tornariam um gargalo no desempenho de um processador RISC.

Estes mesmos avanços na tecnologia de semicondutores tornaram possível a colocação de mais circuitos no próprio chip RISC. Como resultado, a maioria dos chips RISC possui um gerenciador de memória embutido para dar suporte ao modelo de paginação por demanda, empregado na maioria das versões UNIX.

Não somente as RAMs dinâmicas como também as RAMs estáticas tornaram-se mais rápidas, densas e baratas. As SRAMs são por si só mais rápidas do que as DRAMs porque elas evitam o ciclo de resfriamento das versões dinâmicas. Isto permitiu que utilizassem estratégias de *cache*, nas quais uma pequena quantidade de memória SRAM suporta uma grande quantidade de DRAMs mais lentas afim de aumentar a velocidade de acesso à memória. Inicialmente as SRAMs para os processadores RISC ficavam alojadas em chips separados, mas os mais recentes avanços da tecnologia tornaram possível incluir a SRAM no próprio chip do processador, minimizando atrasos de propagação e permitindo *cache* das instruções, como se isto já fosse inerente à arquitetura do processador.

4. Arquiteturas de Pipeline

Além das já debatidas, a maioria das vantagens associadas aos processadores RISC, na realidade, advém de outras inovações em arquitetura que são possibilitadas por ela, e não pelo próprio número reduzido de instruções. A mais importante destas estratégias é o *pipelining*, apesar da memória *cache* embutida no chip também contribuir substancialmente para o desempenho como um todo.

Pipelining é a extensão de uma idéia conhecida como "prefetch", na qual a próxima instrução a ser processada é carregada da memória e decodificada ao mesmo tempo em que se executa a instrução que está sendo processada.

Um processador completamente *pipelined* utiliza a mesma estratégia de uma linha de montagem de automóveis (na qual vários veículos em vários estágios de montagem são trabalhados simultaneamente, porém cada um em uma operação específica diferente, a serem efetuadas em cada carro por vez). Portanto um *pipeline* de cinco estágios pode trabalhar com cinco instruções ao mesmo tempo. Enquanto a primeira instrução tem seus resultados guardados nos registradores de destino, a segunda pode estar executando uma operação aritmética, por exemplo. Após o *pipeline* ter sido preenchido inicialmente, a demora para completar as instruções é meramente o tempo de uma operação (idealmente um ciclo de *clock*). ao invés do tempo necessário para que todas as operações fossem processadas em seqüência.

A operação de um *pipeline* RISC hipotético de cinco estágios, está ilustrado no diagrama de tempo do Quadro 1. As instruções passam pelo *pipeline* da esquerda para a direita e são executadas em uma seqüência de cinco estágios de um ciclo. O tempo é medido de cima para baixo do diagrama em termos de ciclos que irão sendo completados.

Quadro 1

Esquema Simplificado da Arquitetura Pipelined de 5 Estágios

| | <i>Instrução</i> | <i>Registro</i> | <i>Execução</i> | <i>Acesso de Dados</i> | <i>Escreve o Resultado</i> |
|----------------|--------------------|--------------------|--------------------|------------------------|----------------------------|
| <i>Ciclo 1</i> | <i>Instrução 1</i> | | | | |
| <i>Ciclo 2</i> | <i>Instrução 2</i> | <i>Instrução 1</i> | | | |
| <i>Ciclo 3</i> | <i>Instrução 3</i> | <i>Instrução 2</i> | <i>Instrução 1</i> | | |
| <i>Ciclo 4</i> | <i>Instrução 4</i> | <i>Instrução 3</i> | <i>Instrução 2</i> | <i>Instrução 1</i> | |
| <i>Ciclo 5</i> | <i>Instrução 5</i> | <i>Instrução 4</i> | <i>Instrução 3</i> | <i>Instrução 2</i> | <i>Instrução 1</i> |

Fluxo da Instrução ⇒

No primeiro estágio, o processador pega um instrução da memória e a transfere para o registrador de instruções conforme indicado pelo contador do programa. No segundo estágio, as instruções decodificadas são movidas dos registradores e colocadas nos "source bus" como direcionado pela lógica de controle do processador. No terceiro estágio, a unidade aritmética e lógica (UAL executa a operação (por exemplo uma adição) utilizando o operando em seu "source bus". No quarto estágio, dados adicionais são carregados da memória ou do registrador temporário. De certa maneira, este é um segundo estágio de execução para aquelas operações que não foram completadas em um único ciclo (devido à precisão expandida ou operandos adicionais). Finalmente no quinto estágio, os resultados são transferidos dos *buses* de saída da UAL aos registradores de destino.

No *pipeline* do Quadro 1, cada instrução leva um total de cinco ciclos para completar sua execução. Mas à medida que as instruções se movem passo a passo através do *pipeline* e são executadas em paralelo, uma nova instrução é completada a cada um dos ciclos. Desta maneira, cinco instruções podem ser completadas em somente nove ciclos do relógio, opondo-se aos 25 ciclos que seriam necessários se elas fossem executadas separadamente em um processador que não fosse *pipelined*. O tempo que os quatro ciclos iniciais requerem para preencher o *pipeline* é conhecido como latência do processador. Em geral, *pipelines* mais lon-

gos oferecem menor granulação e, portanto, um processamento mais rápido, porém possuem uma latência mais alta.

Uma exigência importante da arquitetura *pipelined*, é que para colher os benefícios da mesma, as instruções enviadas ao *pipeline*, em qualquer momento, devem ser independentes. Se uma das instruções necessita do resultado de outra como seu operando (input), então estas duas instruções devem ser separadas de maneira a nunca estarem no mesmo *pipeline* simultaneamente, senão haverá travamento do *pipeline*. A segunda instrução teria que esperar até que a primeira completasse e gravasse todas as suas operações nos registradores de destino.

Entretanto, os processadores *pipelined* não podem oferecer uma performance além daquela de uma arquitetura comparável não-*pipelined*, a menos que o compilador (e/ou o programador) tenha evitado tais dependências entre as instruções. Esta é outra razão pela qual desenhar um compilador otimizado para processadores *pipelined* (e para outras arquiteturas complexas que exploram o paralelismo), pode ser considerada uma tarefa desafiadora. Por estas e outras razões, o desenvolvimento do software tendeu a se atrasar em relação aos rápidos melhoramentos na performance do hardware. Também os sistemas operacionais mais antigos e a maioria dos pacotes de aplicações falharam em explorar por completo o potencial das novas plataformas baseadas em RISC.

Os processadores CISC já se utilizam há bastante tempo da estratégia “prefetch”, porém não se imaginava possível um *pipelining* completo nestes processadores porque eles não possuíam o pequeno conjunto de instruções simplificadas do RISC. Portanto não haveria um conjunto fixo de operações comuns a todas as instruções. É claro que todas as mesmas necessitam de um operando, mesmo com a utilização da estratégia de “prefetch”.

Diferentemente dos processadores RISC, os CISC também não utilizavam de uma estratégia *load/store* uniforme para todas as instruções. Contudo, os últimos processadores CISC (por exemplo, o Intel Pentium e o 68050 da Motorola) possuem, de fato, *pipelines* e *cache* no chip. Isto foi conseguido com uma arquitetura híbrida, a qual a Intel chamou de

CRISC (Complex reduced instruction set computing). Na unidade de processador matemático de ponto flutuante no mesmo chip, algumas instruções são executadas em estilo RISC, sem microcódigos e em um ciclo do *clock*.

Parece provável que os futuros processadores Intel da série x86 (e seus conorrentes como a Chips and Technologies, Cyrix e Advance Micro Devices) irão estender esta técnica adicionando *pipelines* mais potentes, expandindo assim a porção RISC de seus conjuntos complexos de instruções. Em termos de arquitetura, os processadores serão realmente RISC, com chips CISC embutidos que executarão as exceções que não forem próprias para o *pipeline*.

5. Superpipelined versus Superescalar

Enquanto os processos CISC estão somente começando a se utilizar do *pipeline* e *cache* embutido, os processadores RISC já se utilizam de arquiteturas ainda mais avançadas e potentes. As arquiteturas que competem entre si não são mais a RISC e CISC, mas sim a RISC e a CRISC. Estas últimas empregam estratégias RISC para aumentar o desempenho. O novo objetivo será alcançar um crescente paralelismo e executar mais instruções por ciclos do *clock*. Duas arquiteturas, conhecidas como "Superpipelined" e "Superescalar", vem competindo nos laboratórios de pesquisa. A maioria dos novos chips tem dado ênfase à abordagem "Superescalar", apesar de também surgirem híbridos destas duas novas abordagens (muito parecido com a evolução do CRISC).

Os processadores *superpipelined* alcançam o paralelismo através da extensão do comprimento do *pipeline*, decrescendo assim a sua granulação e aumentando o número de instruções executadas por ciclo. Porém, devido a um número maior de instruções se encontrarem simultaneamente dentro do *pipeline*, problemas com interdependência entre as instruções e oscilações de dados (atrasos cumulativos devido a erros no *cache*, largura de faixa no *bus* e outros problemas de acesso a memória), aumentam o risco de travar o *pipeline*. Outra causa de possível travamento é a mudança do fluxo de instruções causado por interrupções e outros saltos ou ramificações na lógica do programa. Todos estes problemas podem ocorrer, em teoria, com qualquer *pipeline*, mas a sua probabilidade de ocorrência aumenta substancialmente com o *superpipelining*.

Um exemplo de um processador RISC *superpipelined* é o R5000 desenhado pela MIPS Computer (integrante da Silicon Graphics), que se utiliza de um *pipeline* de oito estágios. Os processadores RISC mais simples se utilizam de um *pipeline* de cinco estágios, portanto este *pipeline* mais longo do R5000 significa uma estratégia de arquitetura *superpipelined*.

Uma arquitetura superescalar aumenta o paralelismo por operar vários processadores *pipelined* separados simultaneamente. Esta estratégia e seus benefícios irão além da mera utilização de processadores adicionais (como a utilização de chip coprocessador matemático separado nos PCs). Em uma verdadeira arquitetura superescalar, as operações ocorrem ao mesmo tempo em todos os processadores aumentando o número de instruções executadas por ciclo. Atualmente, isto significa que individualmente os processadores *pipelined* deveriam estar em um mesmo chip (para minimizar atrasos de propagação), que todos os *buses* e acessos à memória deveriam ser "cached"(para favorecer a redução dos atrasos), e que os *buses* deveriam ser largos o suficiente para acomodar simultaneamente várias instruções.

Um exemplo de um verdadeiro processador superescalar é o chip utilizado para as workstations IBM RS/6000. Este chip possui três seções principais: um processador de inteiros (FXU), uma unidade de ponto flutuante (FPU) e uma unidade *cache* de instruções (ICU). A ICU carrega duas instruções ao mesmo tempo da memória através de seu cache de instruções. Os caminhos do *bus* para/da ICU têm a largura de duas palavras. Desta maneira a ICU pode alimentar simultaneamente os *pipelines* da FXU e da FPU com as novas instruções.

A arquitetura superescalar do processador RS/6000 é a base do chip PowerPC, atualmente sendo desenvolvido em conjunto pela IBM, Apple e Motorola. O objetivo desta aliança é desenvolver a estratégia para a próxima geração de plataformas workstations, com desempenho superior ao dos existentes IBM PS/2 e Macintosh da Apple.

As tendências da tecnologia de semicondutores em grande parte determinam as estratégias de arquitetura que serão mais eficazes nos microprocessadores. Se as velocidades dos chips semicondutores estivessem crescendo mais que sua densidade de circuitos, não haveria justificativa em arriscar oscilações do *clock* e os problemas resultantes para o compilador em arquiteturas avançadas de *pipeline*. Poderíamos simplesmente aumentar a frequência do *clock* de um processador mais simples. O PA (Precision Architecture) da Hewlett-Packard, não se utiliza nem das técnicas, de *superpipelining*, nem de superescalar. É somente uma bem projetada CPU *pipelined* com uma frequência alta de *clock*.

Porém, as novas versões do RISC PA incluem um processador de ponto flutuante no mesmo chip a fim de competir com outros chips que oferecem este recurso. Portanto, assim como o chip 486 da Intel que também possui um processador de ponto flutuante no mesmo chip que a unidade de processamento de inteiros, poderíamos argumentar que ele é superescalar. O ponto é que com a experiência desta empresa como pioneira no projeto de chips RISC e como fabricante de chips de alta velocidade utilizados em instrumentos de medições, a HP não necessitou imediatamente dos benefícios de aumento de desempenho utilizados nas arquiteturas mais recentes.

Matematicamente os processadores *superpipelined* e superescalar são duplicações um do outro, portanto nenhum oferece uma vantagem de desempenho mensurável sobre o outro. Ambos alcançam o paralelismo (permitir que duas instruções sejam processadas simultaneamente) utilizando-se de circuitos altamente densos, possíveis graças aos chips semicondutores de litografia submicron. Uma razão primária para a aceitação mais rápida das arquiteturas superescalar em relação a *superpipelined* é a sua modularidade.

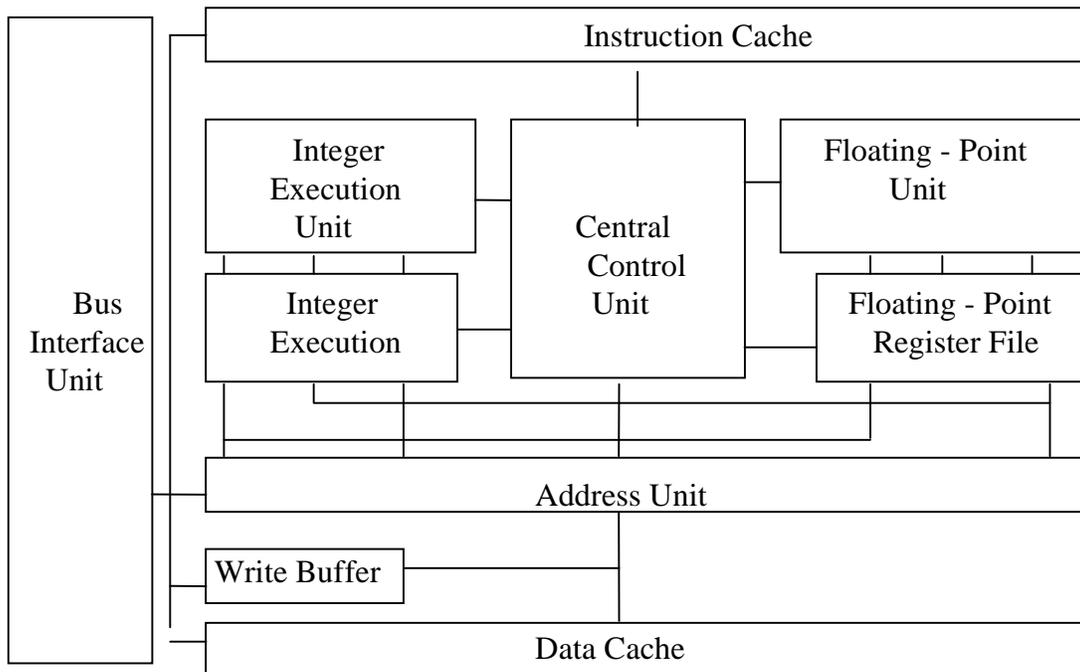
Com a arquitetura superescalar, os vários componentes (por exemplo, o processador de inteiros e o de ponto flutuante) podem ser projetados como células-padrão e podem ser utilizados em várias combinações para chips específicos. Assim sendo, a Intel oferece versões do 486 com o coprocessador matemático desativado. Também poderiam oferecer uma versão que inclui somente o processador de inteiros, com o processador matemático em um chip separado. Da mesma maneira, o RISC PA da HP encontra-se disponível em versões de um ou dois chips. Claramente a modularidade das arquiteturas superescalar permite uma maior customização dos microprocessadores a fim de se adaptarem às diversas estratégias de marketing.

Para os futuros chips de computação de alta performance e multiuso pode-se também equilibrar o uso das estratégias RISC e CISC. Por exemplo, o chip de mais alto-desempenho anunciado até agora são os utilizados nos equipamentos da série Alpha da Digital Equipment Corporation, os quais estão disponíveis em frequência de *clock* de até 300 MHz (Alpha XL 300). Os chips Alpha combinam tanto as técnicas de *pipelining* como de superescalar, ainda

assim eles não utilizam ambos os tipos de arquitetura aos limites extremos, onde os problemas de oscilação e do compilador poderiam se tornar excessivos. Esta série de processadores procura cobrir um grande espectro de plataformas de computadores, desde workstations até supercomputadores.

Um diagrama de bloco do chip Alpha é mostrado no Quadro 2 a seguir. Possui um *pipeline* de números inteiros de sete estágios e um *pipeline* de ponto flutuante de dez estágios, cada um com tempo de ciclo de 5ns (para a versão de 200 MHz). Os acessos à memória se ramificam na *interface* do *bus* e são aceleradas por um *cache* de instrução e um de dados de, ambos de 8 KB. Instruções duplas (duas instruções por *clock*) são então alimentadas via unidade de controle central para as unidades de execução de inteiros e de ponto flutuante. Os operandos são guardados nos arquivos de registradores de ponto flutuante e inteiros. As operações de *load/store* são feitas pela unidade de endereçamento.

Quadro 2 Arquitetura Alpha-Risc



A versão inicial dos processadores da série Alpha são chamados de chips 21064. Foram implementados com a tecnologia CMOS submicron (Complementary Metal-Oxide System) e encontram-se em versões com frequência de *clock* de 150 a 200 MHz. possuem uma arquitetura de 64 bits para os inteiros e de ponto flutuante. Foi projetado para trabalhar como uma máquina *load/store* de 64 bits e com instruções de 32 bits de comprimento fixo. Os tipos de dados manuseados incluem os inteiros de 8, 16, 32 e 64 bits e os formatos de ponto flutuante de 32 e 64 bits (Digital ou IEEE). Uma característica chamada PAL (Privileged Architecture Library) suporta estruturas complexas do sistema operacional tais como mudança de contexto e gerenciamento da memória. Isto torna a arquitetura mais flexível e poderia eventualmente permitir o uso de vários sistemas operacionais. (Open VMS, OS/2, UNIX, etc).

6. Software: Uma barreira

Os novos chips RISC, como os da série Alpha da Digital, sempre enfrentam o problema de não conseguir ganhar uma grande fatia do mercado até que possuam um grande número de software disponíveis. No mínimo eles precisam de um sistema operacional popular como o UNIX, e devem ter o suporte de um *assembler* e de compiladores para a maioria das linguagens de programação (tal como C ou C++), para que as aplicações possam ser portadas à arquitetura. Mesmo assim o software ainda pode ser um gargalo ao desempenho por não dar um suporte completo a todas as operações de 64 bits e outros recursos do novo chip.

No mercado de microcomputadores, portanto, estas dificuldades parecem estimular mais as extensões evolutivas das famílias de processadores mais populares (tal como a série SPARC da Sun Microsystems), do que novos chips revolucionários. Enquanto o desempenho dos novos chips RISC tem um gargalo devido à falta de um suporte de softwares de 64 bits para as aplicações de workstations, os chips CISC da série x86 da Intel ainda encontram-se bastante ligados à computação de 16 bits. É irônico que os chips CISC tenham alcançado originalmente a dominância do mercado devido à sua imensa base de suporte de softwares. Até bem recentemente, não havia suporte a sistemas operacionais de 32 bits para os cada vez mais potentes processadores CISC 38 e 486 de 32 bits que haviam surgido. Enquanto que a Intel e seus concorrentes (AMD, Chips & Technologies, Cyrix, etc) na fabricação de processadores x86 diminuíram bastante a diferença de desempenho do hardware, se comparado ao RISC, a distância ainda parecia grande no nível de sistemas práticos.

Mais recentemente a IBM anunciou o OS/2 Warp Versão 4.0. Diferente das versões anteriores de OS/2, as quais foram desenvolvidas em cooperação com a Microsoft, este novo esforço da IBM tem recursos impressionantes. Ele engloba todas as tendências atuais de projetos de sistemas operacionais, ou seja:

- é um sistema de 32 bits reais;

- possui uma GUI (banco de dados) orientada ao objeto;
- é preemptive e multithreaded;

- suporta um bus de 32 bits;

- dá suporte a uma arquitetura de memória *flat* (diferente do DOS);

- traz uma tecnologia de reconhecimento de voz (o que mais chama atenção no produto).

Ainda assim roda de maneira eficiente as aplicações existentes de DOS e Windows 3.x (com exceção do Windows95). A IBM está incentivando o esforço da Borland de escrever um compilador otimizador C++ para OS/2, o qual permitirá que outras aplicações de 32 bits sejam utilizadas no novo sistema operacional.

Enquanto isto, uma competição potencial vem surgindo de vários pontos. Talvez a maior ameaça seja o Windows NT (New Technology) da Microsoft. O NT apresenta muitas capacidades do OS/2. É o sistema operacional de preferência do antigo consórcio ACE (do qual a Microsoft era membro) e é voltado para as workstations RISC Mips R5000 assim como para as plataformas CISC x86.

Outra ameaça provém do sistema operacional Solaris proposto pela Sun Microsystems, um sistema UNIX que rodará nas plataformas SPARC e x86. A extensa base instalada de workstations SPARC da Sun (perdendo em participação de mercado somente para as plataformas x86), comparada com a escassez de equipamentos R4000, dá a este sistema operacional credibilidade versus o NT da Microsoft. Enquanto isto, a Sun estabeleceu uma participação inicial no mercado de sistemas operacionais para arquiteturas x86, através da aquisição da linha de produtos UNIX/80x86 da Interactive Systems Corp. Com uma estratégia

similar, a Microsoft adquiriu o sistema operacional SCO/UNIX (desenvolvido pela The Santa Cruz Operation) para as plataformas x86.

7. Tendências Recentes das Plataformas RISC e CISC

Como já mencionamos anteriormente, é improvável que a tecnologia de processadores, sozinha, tenha uma influência decisiva na seleção de plataformas de computação para microcomputadores. O desempenho dos processadores excedem as capacidades dos sistemas operacionais e outros softwares de suporte. Também, em vários casos, o baixo desempenho da CPU eclipsa o desempenho dos periféricos e a largura de faixa dos *buses* de *interface* de periféricos. Além disso, a existência de mercados de larga escala, aumento da competição e avanços na tecnologia de semicondutores baixou os custos dos processadores de maneira que eles não são mais o item mais significativo dentro do custo total do sistema.

Pelas razões citadas acima, a análise de arquiteturas de processadores será útil para identificar quais os fabricantes que não irão sobreviver (porque falharam em se manter atualizados com o progresso da tecnologia) ou na seleção de fabricantes e plataformas viáveis para aplicações de alto desempenho, ou aplicações verticais crescentes (onde o básico do desempenho dos processadores é necessário e onde os softwares e periféricos serão customizados para a aplicação).

8. A Arquitetura do CISC Intel x86

Os PCs IBM e seus compatíveis que utilizam a família de microprocessadores CISC da Intel dominam atualmente os microcomputadores, porque eles possuem uma enorme base de softwares aplicativos relativamente baratos. Os processadores operam na frequência do *clock* de até 200 MHz. A utilização de *pipelining*, *cache* e coprocessadores de ponto flutuante já embutidos no chip torna o desempenho dos microprocessadores i486 competitivos em relação a maioria dos chips RISC utilizados em workstations, e aproximadamente 45% mais rápido que os chips 386 operando nas mesmas frequências do *clock*.

Porém, o sistema DOS de 16 bits torna-se um gargalo no desempenho dos PCs. A utilização do UNIX ou de softwares avançados de redes, de 32 bits, como o Netware 386 da Novell aumenta o seu desempenho mas sacrifica a compatibilidade com o software originalmente escrito para as máquinas de 16 bits. Sistemas operacionais emergentes, notadamente o OS/2 4.0 da IBM, preserva a compatibilidade com o software de DOS enquanto oferece multitarefas de 32 bits, endereçamento de memória expandida, e GUI orientadas para o objeto.

A Intel encara uma crescente competição, não somente de fabricantes de chips RISC (tais como o MIPS R4000 e o Power pc da IBM/Apple), mas também de outros fabricantes de chips CISC compatíveis com o x86 (tais como a AMD, C&T e Cyrix). Como resultado, o custo dos processadores despencou e a Intel acelerou o desenvolvimento de processadores avançados aproveitando a tecnologia RISC. Seu processador Pentium se utiliza de uma arquitetura superescalar *pipeline* com processadores RISC duplos e um pré-processadores RISC duplos e um pré-processador CISC 386. Portanto, ele efetivamente une as tecnologias CISC e RISC.

No final de 1994, início de 1995 a Intel liberou o Pentium que constitui a família que originalmente seria chamada i586. A Intel optou por não utilizar a denominação “586” porque a legislação de patentes dos EUA não reconhece a patente de um número.

9. Os Processadores RISC e CISC da Motorola

Uma das únicas famílias de microprocessadores CISC (além da família x86 da Intel) que atingiram uma larga utilização em microcomputadores durante a última década foi a série 68000 da Motorola. As workstations técnicas mais antigas, como as da Sun Microsystems e Apollo (agora parte da Hewlett-Packard) utilizavam-se destes processadores CISC. E apesar de hoje a maioria das workstations estarem mudando para os chips RISC, os computadores Macintosh da Apple ainda se utilizam do Motorola 68030 e dos chips mais antigos da série 68000.

Como a Intel fez com o i486, a Motorola empregou uma arquitetura parcialmente *pipelined* para o 68040 em uma tentativa de estender o desempenho dos chips CISC e adiar a inevitável troca dos fabricantes de computadores para chips RISC. Ainda como a Intel, a Motorola preservou a compatibilidade dos conjuntos de instruções para os novos chips CISC, apesar de ir construindo gradualmente uma base de aplicações para os novos chips CISC, apesar de ir construindo gradualmente uma base de aplicações UNIX que poderia ser facilmente transferida aos novos chips da família.

Prevendo os acontecimentos, a Motorola também perseguiu um desenvolvimento paralelo de chips RISC enquanto estendia a série 68000. Porém a família de processadores RISC 88000, introduzidos em 1988, foram lançados muito tarde. Apesar de seu desempenho ser competitivo, provou-se incapaz de recapturar seu mercado em workstations. A maioria dos fabricantes de workstations já desenvolveram os seus próprios processadores RISC. Por exemplo, a Sun tem a sua família SPARC e a Hewlett-Packard estava utilizando os chips PRISMA. Mais recentemente, a HP anunciou o projeto de alto desempenho do RISC PA.

Mesmo assim, a Motorola continuou o desenvolvimento da série 88000 e recentemente anunciou suas últimas versões, designadas 88110. O código binário compatível da série 68000 foi transferido ao novo chip, estabelecendo um padrão de compatibilidade de arquitetura cruzada. Também um grupo independente de comercialização, o 880pen, foi formado afim de promover a utilização da série 88000 nas aplicações UNIX.

A base da estratégia da Motorola para a série 88000 era ter o 88110 aceito pela Apple como o processador para a próxima geração de computadores Macintosh. Porém, esta estratégia foi por água abaixo quando a Apple decidiu formar uma aliança estratégica com a IBM afim de desenvolver o processador RISC PowerPC. A Motorola juntou-se a esta aliança e passou a fabricar os chips PowerPC. Apesar do projeto da série 88000 ser bom, ele foi introduzido no mercado com atraso e por isso não foi bem aceito.

10. Os Chips SPARC da SUN Microsystems

Atualmente, a família de microprocessadores RISC mais utilizada é a SPARC (Scalable Processor Architecture) desenvolvida pela Sun e fabricada sob licença por pelo menos cinco companhias diferentes de semicondutores. Não somente a Sun é o principal fabricante das workstations UNIX, mas também a empresa optou por licenciar esta tecnologia a outros fabricantes de computadores e, portanto, aumentar a base instalada de equipamentos SPARC. Um consórcio independente, conhecido como SPARC International, que agora possui cerca de 150 membros, promove o desenvolvimento e a utilização desta arquitetura. Assim sendo, como os PCs IBM originais, as workstations da Sun têm sido muito copiadas. Esta competição, junto com a competição entre os fornecedores concorrentes de chips, tem garantido que os preços se mantenham competitivos. Como resultado, existe hoje uma sobreposição substancial em preço e desempenho entre as workstations RISC baseadas no SPARC e os micro-computadores CISC baseados no x86.

Como sugere o nome, a arquitetura SPARC é escalável. Portanto a competição entre os fornecedores de chips tem como resultado ganhos contínuos em desempenho para os processadores SPARC, possíveis graças a avanços na tecnologia de semicondutores como a fabricação submicron. A opção da Sun, entre os chips emergentes para a arquitetura lógica SPARC, parece ser um desenho de chip único altamente integrado (mas não superescalar) desenvolvido em conjunto com a Fujitsu (que fabrica os chips) e apelidado de Tsunami-plus. O novo inclui unidades de execução de ponto flutuante e inteiro, gerenciamento de memória e *cache*.

Outros chips SPARC emergentes incluem um desenho superescalar de 50 MHz (apelidado originalmente de Viking e hoje conhecido como Super SPARC) da Texas Instruments, e um desenho superescalar multichip (apelidado de Pinnacle) da Cyjpress Semiconductor, com desempenho esperado para um pouco abaixo de 100 SPECmarks. Ainda não está claro qual dos três novos processadores SPARC será o vencedor em termos de

aceitação do mercado. Acreditava-se originalmente que a versão da TI seria a mais aceita, porém, têm sido detectados supostos problemas para atingir altas proporções na razão instruções/*clock* na frequência do relógio desejada. Portanto as workstations baseadas no desenho não-superescalar da Fujitsu podem aparecer primeiro, ao menos porque esta arquitetura de implementação simplifica o desenho do sistema.

11. Os RISC PA e PRISMA da Hewlett-Packard

Após a Hewlett - Packard ter adquirido a Apollo Computer em 1989, ela tornou-se o segundo maior fabricante (depois da Sun) de workstations RISC. Apesar disto, a HP se viu às voltas com uma estranha composição de processadores, incluindo a série CISC 68000 e duas arquiteturas RISC diferentes. As workstations da série 9000 utilizaram-se da tecnologia RISC PA (precision Architecture) da casa, enquanto que as mais novas workstations Apollo utilizaram-se do PRISMA (Parallel Reduced Instruction Set Multiprocessor).

Utilizado nas workstations da série 1000 da HP, a arquitetura PRISMA inclui unidades independentes de inteiro e ponto flutuante. O uso de largos *caches* duplos para os dados (64KB) e instruções (128KB), mais *pipelining* e *buses* internos de 64 bits, permitem a execução paralela das operações do inteiro e do Ponto flutuante de dupla-precisão IEEE 754. Os chips PRISMA são em sua maioria baseados em CMOS VLSI de 1.5 micron, apesar da ALU de ponto flutuante do processador utilizar ECL.

A HP enfatizou sua arquitetura RISC PA, a qual surgiu do PA original introduzido em 1983. Os processadores RISC PA eram fabricados pela HP em duas versões. Uma versão de chip duplo opera com um *clock* de 66 MHz (hoje aproximadamente 400MHz - Intel Pentium Pro) e alcança 77 SPECmarks. Uma versão de chip único, mais recente, opera a 100 MHz (hoje 150Mhz a 180MHz) e alcança 120 SPECmarks. Os desenhos da HP são de alto desempenho operando em altas frequências de *clock*, apesar de evitar a necessidade de arquiteturas superescalares pesadas ou *superpipelined* exóticas com tendência a apresentarem problemas.

Originalmente, a HP manteve um rígido controle de sua arquitetura proprietária RISC PA. Mas a empresa aparentemente decidiu contra-atacar a SPARC, MIPS e outras alianças da indústria através do licenciamento da tecnologia para outras empresas, formando um novo

consórcio chamado PRO (Precision Architecture RISC Organization). Os membros deste consórcio incluem a Convex Computer, Sequoia Systems, Samsung e Oki.

A Samsung e a Oki estão desenvolvendo chips CMOS para uso em faixas de baixa a média performance, para seu próprio uso. A Hitachi esta trabalhando em dois chips RISC PA, um para comercialização no mercado e outro para sua nova estação de trabalho.

12. O Consórcio ACE e o RISC da MIPS

A MIPS Computer Systems (adquirida em 1992 pela Silicon Graphics Inc.) desenvolveu processadores RISC, os quais são fabricados em seis empresas de semicondutores diferentes (a empresa não fabrica seus próprios chips). A empresa licencia seus desenhos aos fabricantes de sistemas. Em 1989, a MIPS formou uma aliança técnica com a Digital Equipment Corp. Todas as atuais DECstations da Digital e os DECsystems estão baseados em elementos RISC da MIPS.

No início de 1991, a união dos fabricantes de sistemas, liderados pela Compaq Computer, anunciou a formação do consórcio ACE (Advanced Computing Environment). Este grupo planejava desenvolver sistemas padronizados (com um processador comum e uma *interface* gráfica) baseados no processador RISC MIPS R4000 e no sistema operacional Windows NT de 32 bits da Microsoft. O objetivo era ir de encontro à crescente dominação do mercado pelas arquiteturas x86 da Intel e SPARC da Sun. Em adição à Compaq, Digital Equipment, SGI, e Microsoft, o consórcio incluía a Tandem Computer e Siemens Nixdorf.

Apareceram sinais de que este consórcio estaria se desintegrando. A disponibilidade dos chips x86 avançados, tem melhorado a razão entre custo e desempenho de sua arquitetura. Isto torna menos urgente para empresas como a Compaq a procura de uma alternativa à extensão arquitetura CISC baseada em RISC. Sem muita surpresa, a Compaq deixou o consórcio. O desenvolvimento por parte da Digital dos Chips RISC Alpha de alto-desempenho podem sinalizar um direcionamento dos desenhos MIPS para futuras workstations e certamente para seus computadores mais potentes. Também outros usuários dos chips MIPS utilizam-se de chips RISC. Por exemplo a Prime Computer que é uma licenciada para chips RISC PA da HP.

Abalos na direção do consórcio ACE incluem a introdução pela IBM com sucesso, de seu OS/2 2.0 (enquanto que o do NT da Microsoft, na época, foi adiado), uma desaceleração do processo de desenvolvimento na MIPS, e a aliança IBM/Apple para desenvolver

processadores avançados e sistemas operacionais para equipamentos da classe das workstations.

13. Uma Pequena História dos Processadores

O primeiro processador do mundo nasceu juntamente com o primeiro computador, o Eniac (Eletronic Numerical Integrator and Calculator), construído por John Von Neumann em 1946. O processador é a unidade principal do computador, o cérebro - ele controla o fluxo dos programas, executa operações lógicas e aritméticas, acessa a memória, faz solicitações aos periféricos, confunde-se com a CPU. O processador do Eniac era discreto, ou seja, suas unidades funcionais eram fisicamente separadas e comunicavam-se através de fios, cuidadosamente soldados por mãos humanas. Em 1971, em uma firma chamada Intel Corporation, um homem de nome Ted Hoff construiu um processador que tinha todas as unidades reunidas em um chip: era o 4004, o primeiro microprocessador.

O 4004 foi possível porque, anos antes, em 1959, Bob Noyce (mais tarde fundador e o primeiro presidente da Intel) teve a idéia de reunir vários componentes eletrônicos em uma pastilha de silício. Noyce deu-lhe o nome de circuito integrado, ou C.I., mas o nome que ficou popular foi mesmo chip. A diferença básica entre o processador tradicional (discreto) e o microprocessador é o fato de este último poder ser produzido na linha de montagem, em larga escala, diminuindo drasticamente o custo. Por causa do preço e do pouco calor dissipado, os microprocessadores se espalharam pelo mundo, conquistaram o mercado, fizeram fortunas incalculáveis e quase levaram empresas como a IBM e a DEC à bancarrota.

Desde o Eniac até 1974, os processadores - discretos ou integrados - utilizavam conjuntos de instruções (instructions set) bastantes complexos. Este tipo de arquitetura, por sua difícil execução, exige que o processador analise as instruções e ececute pequenas sub-rotinas (ou microcódigos) fosse contraproducente, Jonh Cocke, da IBM, teve a idéia de construir um processador mais simples, que não necessitasse de microcódigo, deixando o trabalho pesado para os programas. Estava criada a filosofia do computador com conjunto

reduzido de instruções (Reduced Instruction Set Computer - RISC), um processador menor, mais barato, mais frio. Como era preciso identificar os outros computadores, não-RISC, foi cunhado o termo CISC (Complex Instruction Set Computer). Apesar de inventada em 1974, a filosofia RISC só chegou ao mercado em 1985, pelas mãos da Sun Microsystems, com seu Sparc.

Quando os RISC chegaram, passou a haver três vãos no mercado de informática: os mainframes, computadores corporativos, enormes, absurdamente caros; microcomputadores, individuais, baratos e lentos; e as novas máquinas, as workstations, para aplicações científicas, não muito caras mas muito, muito rápidas. O mercado mudou de lá pra cá. O mercado mudou muito de lá para cá. Os micros ficaram muitíssimo rápidos, os RISC ficaram absurdamente rápidos e os mainframes estão acabados. Mesmo os conceitos de computador pessoal e de workstation deixaram de existir. O aumento de performance dos micros e a exaustão vão de workstations forçaram os RISC a diminuir seus preços e desenvolver interfaces gráficas para disputar o mercado com os micros.

A Intel, ao vender microprocessadores como o 4004, pretendia que os fabricantes de computadores (que controlavam todas as etapas da produção) terceirizassem a produção de componentes, algo inédito e, aparentemente, impossível. Parece que o feitiço virou-se contra o feiticeiro, pois hoje há uma miríade de fabricantes de processadores (especialmente RISC) loucos para tomar-lhe o mercado. O número de diferentes processadores à venda é tamanho que é difícil entender o que se passa no mercado. O quer são estes chips, afinal? Quem vai vencer a batalha dos processadores? O que comprar?

Na verdade, a grande briga entre os fabricantes de processadores tem um único objetivo: roubar o mercado da Intel, responsável por mais de 70% dos processadores vendidos no mundo, tarefa nada fácil. Recentemente, a Intel lançou duas novas versões do seu topo de linha, o Pentium. Com o novo Pentium (versões de 200MHz e Pro, que utiliza wafer de 0.35u e tem tensão de 2.7 V) a Intel diz que garante ter resolvido um problema de superaquecimento e, de quebra, colocou no mercado chips com performance digna de um RISC. Uma observação curiosa a ser feita é que cada vez que os microprocessadores sofriam mudanças

em sua tecnologia (velocidade), suas tensões, ou seja, sua alimentação passava a diminuir, fazendo com que os microprocessadores esquentem mais, exigindo técnicas melhores de resfriamento. Em compensação, o consumo de energia diminuía também.

13.1. Vantagens e Desvantagens

A vantagem dos processadores da Intel é evidente: eles rodam código nativo DOS e Windows. Mas também há desvantagens, antes de mais nada: o preço! Estão entre um dos mais caros do mercado. Outro problema era a arquitetura, que já está exausta. Manter a compatibilidade com a família x86 (definida em 1981) obrigou o fabricante a renunciar uma série de recursos e técnicas mais modernas, sacrificando a performance. Manter a compatibilidade é sempre um problema para os produtores qualquer que seja. O Pentium apresenta ainda um outro problema: ele é muito diferente do 486 (principalmente do DX4 para baixo) e, para que se possa usufruir de todas as vantagens, exige recompilação dos programas. Acrescenta-se ainda sobre a grande diferença entre o Pentium e o 486, as principais alterações foram nos barramentos (desaparecimento do VESA no Pentium com a chegada do PCI), criação da IDE On-Board (recurso que facilitou a ligação de discos rígidos e flexíveis na placa-mãe) e a criação do sistema Plug-and-Play (sistema que facilitou bastante na configuração de periféricos instalados).

Para “tomar” o mercado da Intel, é preciso rodar o software que ela roda: programas DOS e Windows. Existem duas maneiras de se fazer isso:

- A primeira é oferecer um chip capaz de se comportar como se fosse um Intel, tão rápido quanto possível. O problema é que, além de não ser uma missão simples, exige que o fabricante se prenda às peculiaridades da família x86. Há dois fornecedores importantes nesta área. Um é a AMD (Advanced Micro Devices), que vem crescendo bastante. O outro, menor que a AMD, é a Cyrix, que impressiona por seu fôlego. A Cyrix pretende lançar uma família de microprocessadores baseados em arquitetura M1, que traz alguns avanços de projeto, como Data Forwarding e Register Renaming para a arquitetura x86 e pretende oferecer um desempenho similar ao de um Pentium sem exigir recompilação.
- A segunda maneira de se rodar o código x86 é através da emulação, o que consiste basicamente de um programa que leia o código binário de um processador Intel e interprete as instruções encontradas. Esta é a outra solução encontrada até agora por todas as outras empresas como a Sun, DEC, Mips, HP e a união Motorola/IBM/Apple (todas de

tecnologia RISC). A interpretação de instruções nativas de outro computador só é possível porque os processadores RISC são realmente muito rápidos. Não há RISC tão rápido que possa rodar num 486 quando se trata de rodar programas como DOS/Windows.

A Sun Microsystem foi a primeira em tecnologia RISC e é até hoje a maior fornecedora de workstations, mas não desfruta mais da vantagem tecnológica que costumava ter. Sua arquitetura Sparc é antiga, mas a Sun tem vantagens comerciais, é um excelente fornecedor possuindo um marketing bastante ofensivo (responsável pela informatização na Copa do Mundo de 1994), e além do mais, seus clientes são famosos e não querem mudar de fornecedor. De qualquer forma, a Sun precisa inovar o seu processador. A tecnologia RISC, por sua simplicidade, faz com que os avanços sejam inevitáveis do que no caso dos x86.

A DEC, segunda maior empresa de computadores do mundo (perdendo para a IBM) oferece o DECchip 21171, que equipa sua linha de computadores Alpha; o topo de linha Alpha XL 300 roda a 300 MHz e é simplesmente o chip mais rápido do mundo (excluindo os multiprocessadores). Os DECchips são processadores de 64 bits reais. Eles suportam os seguintes sistemas operacionais: Open VMS (uma versão do VMS), UNIX (DEC) e o Windows NT (Microsoft). Mais rápido que o Pentium/200 (mais de três vezes em operações com números reais), o Alpha XL 300 poderá se tornar a máquina mais rápida designada para a interface Windows se for julgada pela sua velocidade.

A Mips fabrica os processadores que equipam as máquinas da Silicon Graphics, estações gráficas muito populares. O topo de linha da Mips é o R5500, também de 64 bits, que tem o clock de 250 MHz e cuja performance se aproxima daquela do chip da DEC. O chip da Mips é um dos melhores que há até agora. Talvez só não seria o melhor por não ter o marketing da Sun ou o poder econômico da DEC (hoje a Mips tem processadores fabricados pela NEC). Outro processador não muito comentado é o Hewlett-Packard PA 7200, um bom chip, mas restrito no ponto de vista comercial. Com pouca penetração no mercado mundial, a HP lançou o HP Vectra XU 6/200C; dois processadores de Pentiums Pro, projetado especialmente para aplicativos de 32 bits, para que cresça no mercado.

A Motorola sempre foi a maior concorrente da Intel no campo de processadores CISC. Seus chips equipavam os micros da Apple. No tempo da Apple II Plus, a Motorola 6502 lutava contra o Z80, um clone do Intel 8088. Mais tarde, na era dos 16 e dos 32 bits, a família 680x0 sempre foi superior à família x86, pelo menos até a chegada do 486. A Motorola, a Apple e a IBM se uniram para desenvolverem um novo microprocessador para combater a Intel, o Power PC. É um chip que tem nome de computador que significa: Performance Optimization With Enhanced RISC.

14. Conclusão

Quando um novo microprocessador é criado, aumenta-se o seu conjunto de instruções, para que ele possa dispor de instruções mais poderosas. E até o presente momento é assim que um microprocessador evolui. E quando um outro microprocessador mais novo é criado, uma nova instrução substitui um grupo inteiro de instruções que antes eram necessárias para que um processador executasse uma determinada tarefa. Para o programador, escrever programas fica mais fácil: não há necessidade de escrever um pequeno programa (sub-rotina) para executar o que uma nova instrução faz sozinha.

Na verdade, o pequeno programa continua sendo necessário. Porém, ele estará escrito dentro do microprocessador cuja área encarregada de armazenar tais pequenos programas chama-se microcódigos. Desta forma, quando um novo microprocessador é lançado, ele possui um microcódigo maior, de modo a executar um número maior de instruções. Com um microcódigo maior, o microprocessador será fisicamente maior.

Por outro lado, quanto mais instruções o microprocessador contém, maior será o seu conjunto de instruções. Quando uma instrução é dada ao processador, uma parte existente de dentro dele (decodificador de instruções) trata de verificar se esta instrução é válida e, em caso positivo, trata de executar o subprograma referente a tal instrução dentro do microcódigo. Quanto maior o conjunto de instruções, mais lento ficará o microprocessador; ou seja, a procura pela existência e validade de uma determinada instrução e a sua efetiva execução dentro do microcódigo demorará mais tempo para finalizar (há mais dados a serem perquisados). E para compensar essa demora, passava-se a utilizar diversos novos recursos para o aumento da performance como por exemplo o método de *pipelined* (comentado no desenvolvimento do trabalho), fazendo com que acreditamos realmente que o microprocessador é mais avançado por ser mais rápido. A esta arquitetura denominamos de CISC (*Complex Instruction Set Computer* - Conjunto Reduzido de Instruções de Computador).

Apesar de existir um grande conjunto de instruções, apenas 20% delas são realmente usadas por programas e sistemas operacionais. O restante das instruções é raramente usado. Com isto, surgiu uma nova idéia: a construção de microprocessador com um conjunto de instruções reduzido. E mais do que isto: a completa eliminação do decodificador de instruções e, principalmente do microcódigo. A solução encontrada foi a completa padronização.

Assim sendo, microprocessadores construídos a partir dessa idéia (poucas instruções, padronizadas e sem microcódigos) são classificados de *RISC (Reduced Instruction Set Computer - Conjunto de Instruções Reduzidos de Computador)*. Todas as instruções RISC têm o mesmo tamanho e demoram o mesmo tempo dentro do microprocessador. Quando uma instrução é dada a um microprocessador RISC (sem decodificador de instruções nem microcódigos), cada bit da instrução é responsável por ativar e/ou desativar diretamente algum circuito lógico existente dentro do microprocessador. Logo, a tendência atual é a construção de microprocessadores RISC. Entretanto, os microprocessadores RISC são incompatíveis com os CISC, necessitando de programas e sistemas operacionais especiais (emulação).

Por outro lado, nota-se claramente, a partir do Pentium, que a própria Intel vem criando seus microprocessadores com uma ou outra característica RISC (uma arquitetura interna mais enxuta). O próprio Pentium Pro (P6) utiliza uma arquitetura híbrida muito interessante. Entretanto, enquanto a Intel reluta em utilizar mais recursos RISC em sua linha de microprocessadores CISC, alguns outros fabricantes não se limitaram tanto. Pelo contrário. Algumas empresas (como a Cyrix e AMD/NexGen) criaram famílias de microprocessadores Pentium-compatíveis, porém com uma arquitetura interna muito melhor do que o Pentium da Intel.

Para se ter uma idéia de performance, quando tais microprocessadores Pentium-compatíveis foram criados, dizia-se que os mesmos eram concorrentes do Pentium Pro (P6), com um preço acessível e com performance superior.

Enfim, ninguém precisa se assustar com o enorme poder dos RISC. Eles não vão tornar ilegíveis as imensas bases de dados acumulados até agora pelos usuários individuais ou pelas empresas. O caminho a ser seguido pelos estrategistas do RISC é fazer algo semelhante ao acontecido quando se introduziu a televisão em cores. Proprietários de aparelhos preto-e-branco continuaram recebendo seus programas normalmente enquanto a base de aparelhos novos pôde desfrutar, com o mesmo sinal, as imagens coloridas e com som.

15. Bibliografia

COMPUTER WORLD. CISC e RISC: vale a pena discutir? São Paulo. Jun.1993, p20.

BYTE BRASIL. São Paulo. Editora Rever, v.5, n° 09, set.1996. p19; p49. Edição mensal

TORRES, Gabriel. Curso completo Hardware. Editora Axcel Books do Brasil, Rio de Janeiro.
1996. 690pp.