Arquiteturas Superescalares*

Marcelo Fontes Santana
RA:100602†
Instituto de Computação
Unicamp
Campinas, Brasil
marcelo.santana@students.ic.unicamp.br

ABSTRACT
Neste artigo são apresentadas diferentes abordagens de arquitetura de computadores. Faz-se uma distinção de cada uma delas com as arquiteturas superescalares. Além disso, são apresentas as principais características utilizadas pelos processadores superescalares. Como forma de proporcionar um conhecimento extra sobre o adquirido na sala de aula, são apresentados alguns processadores superescalares com suas principais características.

Keywords
Arquitetura de Computadores, Arquiteturas Superescalares, Processadores Superescalares

∧

1. INTRODUÇÃO
As aplicações computacionais tem evoluído consideravelmente. Assim, para que este desenvolvimento fosse possível, foi necessária também a evolução dos processadores, pois eles são os responsáveis por executar as tarefas delas com a maior eficiência, tornando-as viáveis computacionalmente.
Assim, em meados da década de 1980 os processadores superescalares começaram a ser projetados [2]. Os projetistas buscavam romper a barreira do pipeline de uma única instrução executada por ciclo de clock [2].
Fora observado que aumentar a frequência de clock é uma alternativa para melhorar o desempenho dos processadores, no entanto, com o passar do tempo, também fora verificado que existia um limite de frequência, a qual dependia do tipo de tecnologia que era utilizada para construí-lo. O aumento excessivo da frequência do clock trazia vários outros problemas que acabavam afetando o seu desempenho, como por exemplo o aumento do consumo.

*A versão completa deste documento pode ser obtida na página da disciplina de Arquitetura de Computadores 2010. `http://www.ic.unicamp.br/ ducatte/mo401/mo401.html`

†Msc. Marcelo Fontes Santana.

Para solucionar esse problema foram desenvolvida algumas técnicas para a exploração do paralelismo. Elas melhorariam o desempenho do processador ao realizar mais de uma instrução de cada vez. Assim, essas técnicas foram cada vez mais evoluindo e a sua implementação nos processadores tem sido uma das principais razões dos altos desempenhos obtidos pelos computadores que utilizam essas tecnologias.
Neste trabalho serão estudadas essas técnicas paralelas utilizadas pelo processadores superescalares, as suas características e aplicações. Primeiramente será feita uma distinção entre as arquiteturas dos processadores, e depois serão introduzidos alguns tipos técnicas de paralelismo, tanto no nível das instruções, como no nível dos processadores. Após isso, como forma diferencial das técnicas apresentadas em sala, será dada ênfase a algumas arquiteturas superescalares, de uma forma bem detalhada, explorando a maioria das técnicas e tecnologias utilizadas para implementar essa arquitetura.

2. ARQUITETURA DOS PROCESSADORES
2.1 Microarquitetura dos processadores
A forma como estão dispostas e utilizadas as estruturas e os componentes do processador define o modelo da arquitetura de um processador. Assim, há diversas classificações de arquiteturas de processadores baseadas nas suas políticas e nos caminhos de execução dos dados. Na subseção seguinte é dada a distinção entre as principais arquiteturas em relação às arquiteturas superescalares, tomando como referência o sequenciamente e o fluxo entre as operações.

2.2 ARQUITETURAS SUPERESCALARES X Outras Arquiteturas
Uma arquitetura superescalar é aquela na qual várias instruções podem ser iniciadas simultaneamente e executadas independentemente umas das outras. Na arquitetura *pipeline* é permitido que diversas instruções sejam executadas ao mesmo tempo, mas elas devem estar, obrigatoriamente, em estágios diferentes do pipeline, num dado momento. A Figura 1 apresenta o fluxo de execução em pipeline de quatro instruções. Nela pode ser observado que cada instrução é processada num processo sequencial, onde a cada unidade de tempo uma instrução passa do seu estado atual no *pipeline* para o próximo, liberando assim o estágio para uma próxima instrução que vem em seguida.

Assim, conforme a Figura 1, a execução em *pipeline* aumenta o desempenho dos computadores de uma forma geral. No

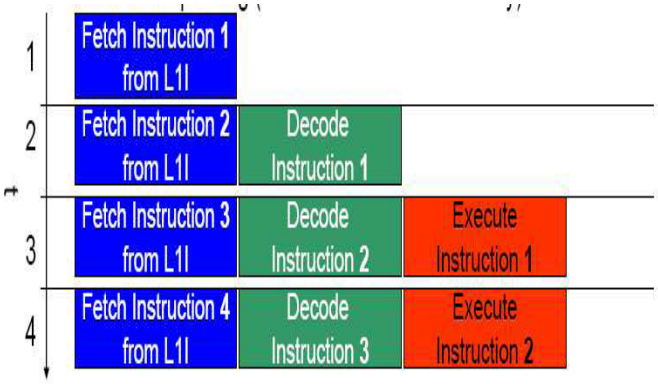

Figure 1: Fluxo de Execução em *Pipeline*.

entanto, ele só aumenta o *throuput* da execução, não altera a latência de cada instrução.
As arquiteturas superescalares incluem todos os aspectos do *pipeline* e ainda acrescentam o fato de as instruções poderem estar executando no mesmo estágio do *pipelining* (em linhas *pipelining* diferentes). Assim, elas tem a habilidade de iniciarem múltiplas instruções no mesmo ciclo de clock. Na Figura 2, pode ser observado que é realizado o *fetch* de quatro instruções ao mesmo tempo em L1-I.

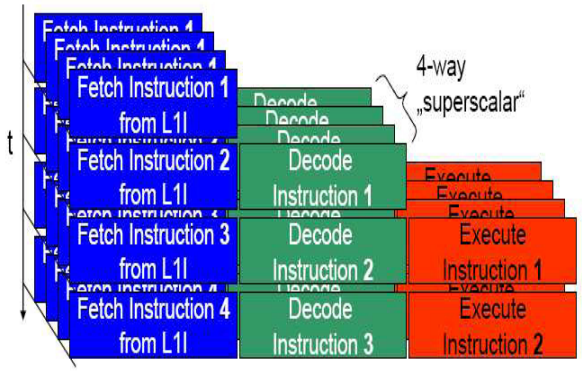

Figure 2: Fluxo de Execução Superescalar 4-way.

Num processador superpipeline, os estágios do processador *pipeline* são divididos em sub-estágios conforme a Figura 3.

Segundo [7], *Very Long Instruction Word* ou VLIW, é uma arquitetura de CPU que executa um grupo de instruções ao mesmo tempo. Um compilador garante que as instruções a serem processadas não sejam dependentes entre si para que possam ser executadas ao mesmo tempo sem perda de lógica do processamento, em alguns casos o compilador acrescenta instruções em branco a fim de garantir a não dependência das instruções.

Sendo assim, a grande diferença do processador VLIW é que o seu escalonamento das intruções é realizado por software,

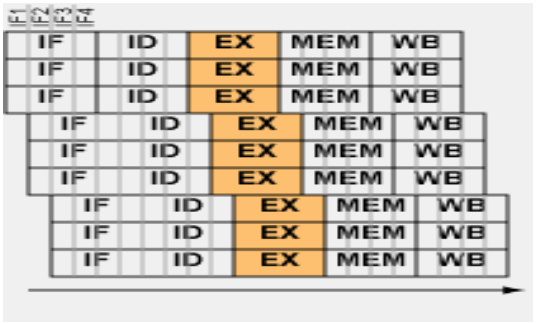

Figure 3: Fluxo de Execução Super Pipeline.

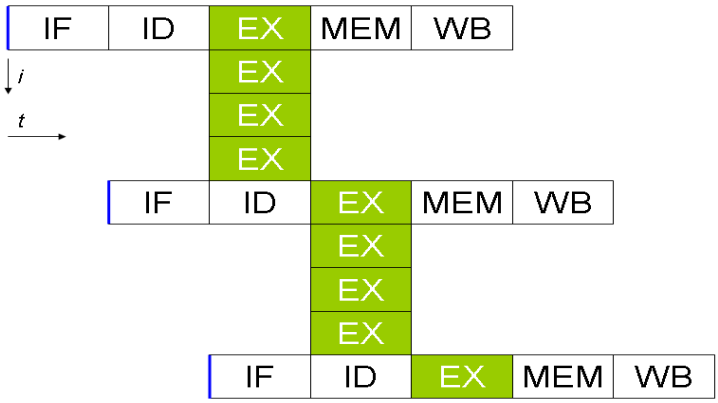

Figure 4: Fluxo de execução vliw.

ou seja, o compilador é o responsável por entregar as instruções de certa forma que a arquitetura seja aproveitada ao máximo. Desta forma, o VLIW não dispõe de escalonamento dinâmico.

Assim, existem algumas diferenças entre os processadores *pipeline*, superpipeline, superescalar, VLIW. Além deles, existem os *hyperthreading* e *multicore*. Aquele está relacionado com a forma de comunicação dos vários cores num mesmo processador. E este, multicore, está relacionado com a combinação de vários processadores em qualquer uma das arquiteturas descritas aqui.

3. ARQUITETURAS SUPERESCALARES

Nesta seção são abordadadas algumas das principais características dos processadores superescalares. Muitas das técnicas utilizadas são utilizadas em outras arquiteturas, no entanto, busca-se uma breve visão sobre estas técnicas. Uma melhor explanação sobre as técnicas aqui citadas podem ser encontradas em [2].

3.1 Dependências

Devido ao pipeline e ao despacho de várias instruções que devem ser executadas independentemente, a arquitetura superescalar tem que resolver todos os tipos de dependências: dependência de controle, dependência de dados e dependência estrutural. As dependências de dados são do tipo

RAW (Read-After-Write), WAR(Write-After-Read) e WAW (Write-After-Write). A resolução das dependências deve ocorrer para que a semântica do código alvo seja mantida.

3.2 Busca de Instruções

Responsável pelo fornecimento de instruções,onde as instruções são buscadas na cache e inseridas na fila de intruções, quando são feitas as previsões de desvios. Assim, antes da fase de busca existe uma fase de pré-busca *pre-fetching* visando eliminar ocorrências de falhas.
Em processadores superescalares a fase de busca traz várias instruções por ciclo da cache. Para suportar a largura de busca torna-se necessário separar a cache de dados da cache de instruções. Assim, o número de instruções buscadas deve atender a taxa de decodificação e execução.

3.3 Detecção do paralelismo

Podemos classificar as máquinas superescalares em duas classes: maquinas que implementam o algoritmo de detecção do paralelismo em hardware e a classe sem esse mecanismo de detecção.
No primeiro caso, compete ao algoritmo a tarefa de determinar se um ou mais comandos podem ser executados em paralelo. Ele é denominado mecanismo de despacho. Este mecanismo combinado com um sofisticado esquema de decodificação viabilizam a exploração do paralelismo, logo, tornam as unidades de controle dessa arquitetura mais complexas. Independente da estruturação do código paralelo, o hardware tentará explorar ao máximo os recursos da máquina.

3.4 Despacho de Múltiplas Instruções

Um dos grandes diferenciais das arquiteturas superescalares é o despacho de múltiplas instruções. O mecanismo de despacho procura na janela de instruções aquelas que podem ser executadas, (incluindo as fora de ordem). Um dos grandes desafios do despacho é identificar as dependências de dados As etapas são:

- identificar as instruções;
- selecionar as que serão encaminhadas;
- encaminhar as instruções para as filas de remessas das unidades funcionais;
- desalocar as instruções já encaminhadas.

Assim, precisam ser verificadas as disponibilidades dos operandos, das unidades funcionais, das interconexões, dos barramentos e das portas para acessar o buffer de ordenamento.

3.5 Previsão de Desvios

Para despachar multiplas instruções por ciclo é necessário minimizar as perdas por desvios. Nas arquiteturas superescalares, as perdas por desvios causam sérias consequências no desempenho: 25% a 75% das perdas são devido a erros de desvios [5]. Existem três técnicas de desvio: desvio com retardo; previsão de desvio; e múltiplos fluxos.
A primeira técnica desloca intruções que não afetam o resultado para depois do desvio. É uma forma de adiantar a execução do desvio diminuindo o número de ciclos perdidos quando do descarte. Ela funciona desde que haja instruções independentes.
Já a segunda técnica pode ser feita por previsão estática ou previsão dinâmica. Na previsão estática, a previsão é definida antes da execução do programa, muitas vezes por compilação. Sua principal vantagem é a simplicidade. Com ela se obtêm acertos de 70%, facilmente, estabelencendo que os desvios são sempre tomados. Para obter melhores taxas de acerto, é necessário o emprego do compilador com informações do perfil do programa. Já a previsão dinâmica, ela viabiliza a execução especulativa, na qual o resultado do desvio é previsto dinamicamente. As instruções são executadas condicionalmente, aguardando a previsão: correta - o programa continua normalmente; incorreta - os resultados são descartados, recomeçando a execução com o desvio correto [5].
Além dessas técnicas há também as que que empregam informações dos desvios condicionais anteriormente executados, como forma de aprendizado. O armazenamento do histórico pode ser feito através do BTB - (Branch Target Buffer), proposta por Lee e Smith em 84. Outra forma é empregando contadores saturados que implementam autômatos finitos com poucos bits.
Maiores detalhes sobre estas técnicas aqui apresentadas podem ser encontradas em [2].

3.6 Execução de Instruções

Nesta etapa ocorre a verdadeira execução da instrução. Os processadores superescalares possuem várias unidades funcionais para executar as várias intruções no estágio de execução simultãneamente. No entanto, elas ainda podem limitar o desempenho do processador, dependendo do programa que estiver sendo executado e da política de despacho utilizada pelo processador. Além disso, o desempenho de cada unidade funcional depende da latência, pois algumas instruções são complexas.

3.7 Commit

Como nas arquiteturas superescalares há a emissão e execução de várias instruções simultãneamente, a fase de *commit* permite manter os efeitos das instruções como se a execução fosse sequencial. Duas técnicas são comumente usadas para recuperar estados precisos. Ambas mantêm: um estado enquanto a operação executa e outro estado para a recuperação.
A primeira técnica usa *checkpoints*. o estado da máquina é salvo em determinados pontos enquanto instruções executam e, também, quando um estado preciso é necessário. Os estados precisos são recuperados de um *history buffer*. Na fase *commit* são eliminados estados do *history buffer* que não são mais necessários.
A segunda técnica divide em dois o estado da máquina: estado físico e estado lógico. O estado físico é atualizado assim que as instruções completam. O estado lógico é atualizado na ordem sequencial do programa, assim que os estados especulativos são conhecidos. O estado especulativo é mantido em um *reorder buffer* que, após o *commit* de uma instrução,

é movido para os registradores ou para a memória. A técnica com *reorder buffer* é mais popular, pois, além de proporcionar estados precisos, ela ajuda a implementar a renomeação de registradores.

4. PROCESSADORES SUPERESCALARES

4.1 AMD AthlonTM MP

Esta arquitetura QuantiSpeed possui quatro características que melhoraram o desempenho dos processadores AMD Athlon MP. Entre elas estão a arquitetura superescalar com 9 despachos, unidade de ponto flutuande em pipeline, pré-busca de dados em hardware e melhorias em seu TLB (*Translation Look-aside Buffers*)[3].
O processador AMD Athlon MP possui uma grande quantidade de execução com 9 canais (pipelines). Os canais de execução são três unidades de cálculo de endereços, três unidades de endereço e três unidades de ponto flutuante. A Figura 5 apresenta a arquitetura do AMD Athlon TM MP.

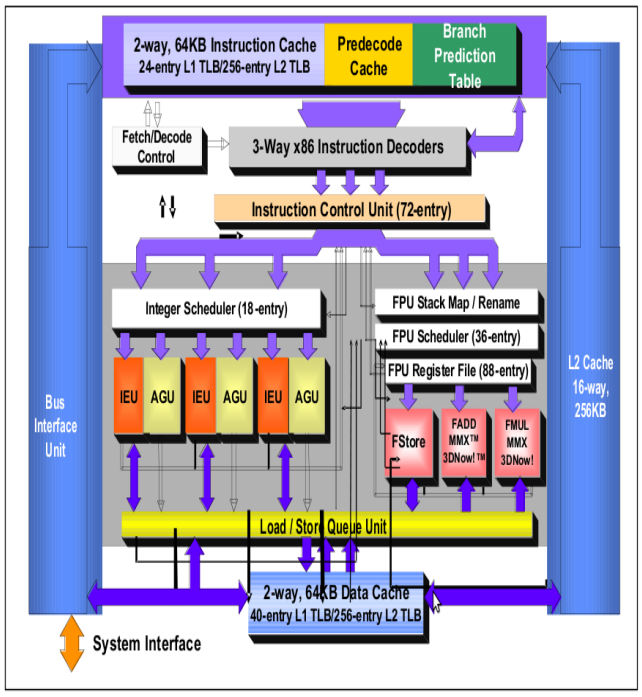

Figure 5: Diagrama de Blocos da Microarquitetura do AMD Athlon TM MP[3].

O processador AMD Athlon MP tem alta performance porque possui no chip uma cache dual-ported de 128KB (duas separadas de 64KB) dividindo a cache L1 uma integrada em alta velocidade, com conjunto de associatividade 16-way. Além disso, possui 512 KB de cache L2 usando uma interface de 72-bits (64 bits de dados + 8 de ECC).
As aplicações exploram o tamanho da cache pelo benefício do princípio da localidade. A cache de dados possui oito bancos para permitirem paralelismo máximo para execução de multiplas aplicações. Ela suporta acessos concorrentes por dois loads ou stores de 64 bits. A cache de instruções contém dados pré-decodificados para permitirem multiplas instruções, decodificação de instruções de alta performance. Ambas as caches possuem dual-ported e contém portas snoop dedicadas, projetadas para melhorar todo o tráfego do sistema de coerência[3].
Para melhorar o CPI do processador e, portanto a performance, o processador Athlon MP também usa pré-busca de dados em hardware. Este hardware busca antecipadamente os dados observando os acessos à memória, procurando por padrões de acessos regulares e especulando buscar a linha da cache com o dado na cache L2 de dados em avanço ao acesso dos dados, portanto reduzindo a latência média vista pelo processador no acesso à memória[3].

4.1.1 TLBs Exclusivo e Especulativo

O processador AMD Athlon MP possui dois níveis de TLBs (*Translation Look-aside Buffers*). Estas estruturas são para traduções de instruções e de dados. A TLB de instruções (I-TLB) de nível 1 (L1) possui 24 entradas. A TLB de dados (D-TLB) de nível 1 (L1) possui 40 entradas, e a (I-TLB) e (D-TLB) de nível 2 (L2) possuem 256 entradas cada[3].
Para reduzir a incidência de clonflitos de TLB, as estruturas de L1 e L2 adotam um projeto de arquitetura exclusiva. Com uma arquitetura de TLB exclusiva, as TLBs de nível 1 contém entradas que não são duplicadas na TLBs de nível 2, permitindo a combinação dos tamanhos das TLBs L1 e L2 para um espaço de entradas disponíveis total maior nas TLBs de instruções e dados [3].
As estruturas TLBs do processador AMD Athlon MP também permitem que as entradas das TLBs sejamm escritas especulativamente antes da primeira instrução ser completada, enquanto estiverem preservando corretamente a ordem de execução da instrução que removem o efeito da serialização e o resultado na peformance do sistema melhorado.

4.2 Arquitetura dos processadores superscalar PA-RISC

Os processadores HP PA 8000 e o PA 8200 PA-RISC foram algumas das primeiras implementações de uma nova geração de processadores da Hewlett-Packard [8]. O PA 80001-3 estava entre os melhores e mais avançados processadores do mundo. O PA 8200 continuou sua performance liderando com sua alta frequencia, caches maiores e diversas outras melhorias [1]. Ambos os processadores caracterizam uma implementação de um processador superescalar de 4 execuções, combinando execução especulativa com reordenamento de instruções.

Uma das características mais importantes deste processador é ilustrada no centro na Figura 6, o buffer reordenador de instruções com 56 entradas, o qual serve como uma unidade de controle central. Este bloco suporta renomeação completa de registradores para todas as intruções no buffer, e caminhos interdependente entre intruções para permitir fluxo de execução através de uma janela completa.

4.3 MIPS 10000

O MIPS 10000, Figura 7, possui 64 registradores de 64 bits (inteiros), 64 registradores de 64 bits (ponto flutuante) e 31 registradores de controle.Ele pode buscar até quatro ins-

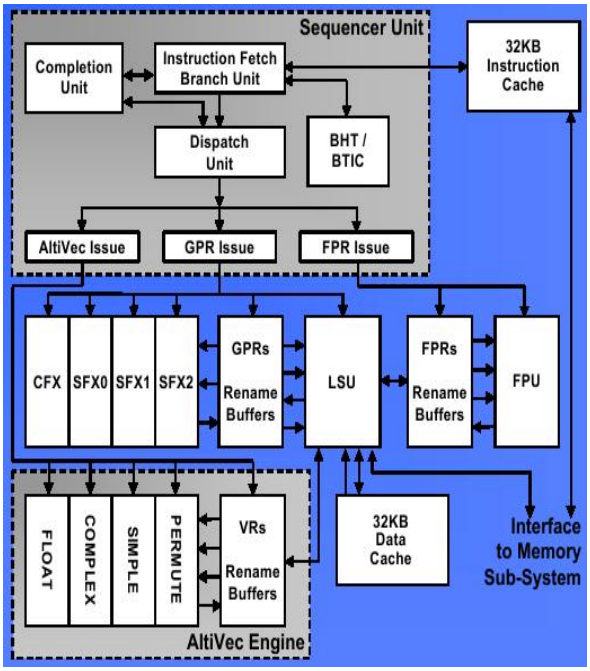

Figure 6: Diagrama de blocos funcional do processador HP PA 8000 [1].

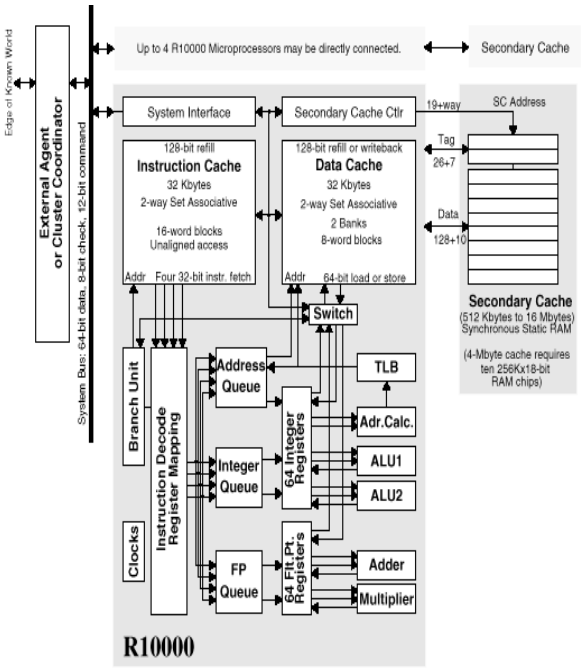

Figure 7: Diagrama de blocos do MIPS 10000 [4].

truções independente do seu alinhamento, as quais são pré-decodificadas (quatro bits são usados para identificar o tipo da instrução) antes da inserção na cache de 512 linhas. A cache de instruções, two-way set associative, contém uma tag de endereços e um campo de dados. Uma pequena TLB de oito entradas mantém um subconjunto das traduções da TLB principal. Logo após a busca, são calculados os endereços de jumps e branches, que são, então, preditos. A tabela de predição, com 512 entradas de 2 bits, está localizada no mecanismo de busca de intruções. A janela do processador considera até 32 instruções em busca de paralelismo [4].
Ao tomar um desvio, um ciclo é gasto no redirecionamento da busca de intruções. Durante o ciclo, as instruções para um caminho não tomado do desvio são buscadas e postam em uma resume cache, para 4 blocos de instruções, o que permite que até 4 desvios sejam considerados em qualquer momento.
Quando um branch é decodificado, o processador salva seu estudo numa pilha de branch com 4 entradas. O processador para de decodificar se um branch chega e a pilha está cheia. Se um branch é determinado incorreto, o processador aborta todas as instruções do caminho errado e restabelece o estado a partir da pilha de branch [4].

Após a busca, as instruções são decodificadas e seus operandos são renomeados. O despacho para a fila apropriada (memória, inteiros ou ponto-flutuante) é feito com base nos bits da pré-decodificação. O despacho é parado se as filas estiverem cheias. No despacho, um busy-bit para cada registrador físico de resultado é estabelecido como ocupado. O bit volta ao estado não ocupado quando uma unidade de execução escreve no registrador. Todos registradores lógicos de 32 bits são renomeados para registradores físicos de 64 bits usando free lists (Figure). As free lists de inteiros e ponto-flutuante são quatro listas circulares, paralelas, de profundidade oito. Isso permite até 32 instruções ativas [4]. Cada instrução, nas filas, monitora os busy-bits relacionados com seus operandos até que os registradores não estejam ocupados. Filas de inteiros e ponto-flutuante não seguem uma regra de FIFO, funcionam de forma similar a estações de reserva. A fila de endereço é uma FIFO circular que mantém a ordem do programa.
Existem cinco unidades funionais: um somador de endereçoes, duas ULAs, uma unidade de ponto flutuante (multiplicação, divisão e raiz quadrada) e um somador de ponto flutuante. Os pipelines de inteiros ocupam um estágio, os de load ocupam dois e os de ponto flutuante ocupam três estados. O resultado é escrito nos registradores do estágio seguinte. Estados precisos são mantidos no momento de exceções com um reoder buffer. Até quatro instruções recebem commit na ordem original do programa.
A hierarqueia de memória implementada de modo não bloqueante com dois níveis de cache set-associative. Todas as caches usam um algoritmo de realocação aproximado ao LRU. Endereçoes de memória virtual são calculados com a soma de dois registradores de 64 bits ou soma de um registrador e um campo imediato de 16 bits. A TLB traduz esses endereços virtuais em endereços físicos [4].
O R10000 tem três modos de operação e dois modos de endereçamento. Os três modos de operação estão listados em ordem decrescente de privilégios ao sistema:

- Modo Kernel (maior privilégios ao sistema) - pode acessar e modificar qualquer registrador. O núcleo mais interno do sistema operacional roda em modo kernel.
- Modo Supervisor - tem privilégios menores e é usado para seções menos críticas do sistema operacional.
- Modo Usuário (menos privilégios ao sistema) - previne

que usuários interfiram entre si.

5. POWERPC

PowerPC é uma família muito grande de processadores, mas todos seguem a mesma arquitetura básica, composta dos seguintes itens:

- ULA (unidade lógica aritmética) de inteiros de dois tipos: simples e complexa;
- unidade de ponto flutuante;
- unidade de carga/descarga (load/store);
- unidade de execução de desvio (branches);
- unidade de controle;
- cache de dados
- cache de instruções.

Especificamente o PowerPC 604 possui seis unidades de execução independentes, Unidade de execução de desvio, Unidade de Load/Store, 3 unidades de inteiros, unidade de ponto flutuante e despacho em ordem. O PowerPC620 possui o diferencial em relação ao PowerPC 604 por possuir despacho fora de ordem [6]. A Figura 8 apresenta a arquitetura do PowerPC 604.

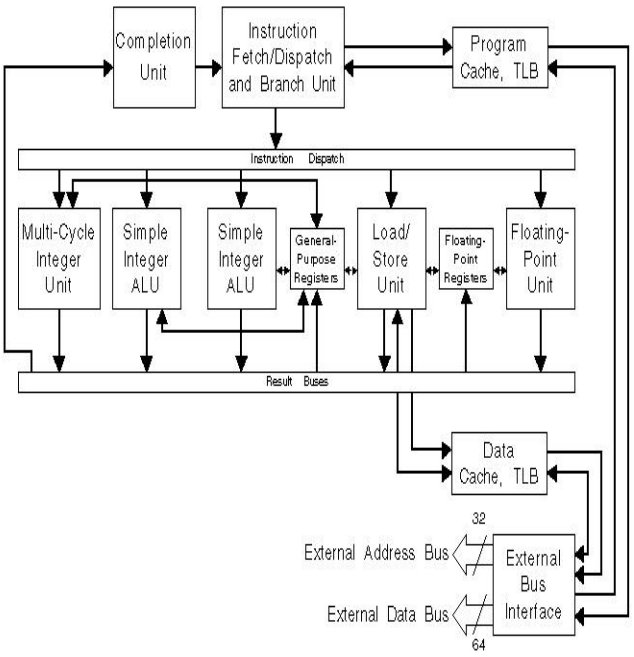

Figure 8: Arquitetura do processador PowerPC604 [6].

6. CONCLUSÃO

Entre várias características das arquiteturas apresentadas, podemos observar que os processadores superescalares apresentam algumas vantagens que continuam sendo utilizadas nos processadores modernos. Como exemplos, podem ser citadas as que o hardware detecta um paralelismo potencial entre as instruções, tenta despachar algumas instruções assim que possível em paralelo e resolve o renomeamento de registradores, diminuindo bastante a carga no banco de registradores. Além disso nestes processadores há a compatibilidade binária, pois, quando são adicionadas novas unidades funcionais numa nova versão da arquitetura ou outras melhorias, elas devem ser feitas apenas na arquitetura (sem mudar o as instruções), assim os programas antigos podem se beneficiar do potencial de paralelismo adicionado.Desta forma o novo hardware poderá despachar a sequência antiga num modo muito mais eficiente sem alteração do código.
No entanto, cabe observar que a arquitetura superescalar possui uma complexidade maior, além de muito hardware necessário para detecção de desvios em tempo de execução, pois existe um limite na distância da janela que pode ser feita com esta arquitetura. Além disso o consumo de energia pode ser muito maior devido a grande complexidade das estruturas inseridas.

7. REFERENCES

[1] E. DeLano, W. Walker, J. Yetter, and M. Forsyth. A high speed superscalar pa-risc processor. In *COMPCON '92: Proceedings of the thirty-seventh international conference on COMPCON*, pages 116–121, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press.

[2] J. Hennessy and D. Patterson. *Computer Architecture - A Quantitative Approach*. Morgan Kaufmann, 2003.

[3] J. Huynh. The amd athlon tm mp processor with 512kb l2 cache - technology and performance leadership for x86 microprocessors. Technical report, AMD, 2003.

[4] R. Martin, Y.-C. Chen, and K. Yeager. Mips r10000 microprocessor user's manual. Technical report, 2011 North Shoreline, 1997.

[5] P. Navaux. Arquitetura superescalares. Arquiteturas Avançadas, Aug 2009.

[6] S. P. Song, M. Denman, and J. Chang. The powerpc 604 risc microprocessor. *IEEE Micro*, 14(5):8–17, 1994.

[7] Vliw - very long instruction word, maio 2010.

[8] F. way Superscalar Pa-risc, K. P. Burkhart, and A. P. Scott. Four-way superscalar pa-risc processors: The hp pa 8000 and pa 8200 pa-risc cpus feature an aggressive four-way superscalar implementation, speculative execution, and on-the-fly instruction reordering, 1997.