

O Processador Intel Itanium II

Claudio Schepke

¹ Instituto de Informática – Universidade Federal do Rio Grande do Sul
Caixa Postal 15064 – 90501-970 Porto Alegre, RS

cschepke@inf.ufrgs.br

***Resumo.** A construção de máquinas eficientes, capazes de realizar simultaneamente um grande conjunto de operações em tempos cada vez menores dependem essencialmente de um bom projeto de arquitetura de computadores. Este trabalho visa descrever o microprocessador **Itanium II**, desenvolvido pela Intel e utilizado em sistemas de alto desempenho. Para tanto, serão apresentados a estrutura, os modelos e as diferentes tecnologias e características de implementação adotadas, oferecendo uma boa visão geral do funcionamento do microprocessador.*

1. Introdução

Apesar dos processadores continuarem dentro das expectativas da Lei de Moore [Hiremane, 2005], segundo a qual a capacidade de processamento dobra a cada 18 meses, mantendo os mesmos custos de produção, as necessidades de processamento são cada vez maiores, tanto para aplicações científicas como para aplicações comerciais.

Neste sentido, diversas tecnologias vem sendo utilizadas pelos fabricantes de processadores para que seja possível construir máquinas eficientes, destinadas a atender a um amplo conjunto de aplicações. Essas tecnologias incluem desde a melhoria na confecção física do *hardware*, passando pela estruturação dos componentes, até a utilização de múltiplas instruções em um mesmo ciclo de *clock* [Santos and Santos, 2005]. Em termos de mercado, existem diversos modelos de arquitetura e processadores disponíveis, cada qual tendo um enfoque específico.

Este trabalho tem como objetivo descrever o processador Itanium II. A próxima seção aborda algumas características gerais, bem como a descrição do primeiro Itanium e dos diversos modelos desenvolvidos de Itanium II, incluindo os modelos atuais. Uma segunda seção apresenta algumas características da arquitetura IA-64 utilizada no processador. Na seqüência são discutidas algumas idéias envolvendo projetos futuros decorrentes da evolução das implementações existentes. Por fim, são apresentadas as conclusões obtidas com a realização deste trabalho.

2. Itanium II

Itanium II é um microprocessador IA-64 (*Instruction Architecture-64*) desenvolvido por *Hewlett-Packard* (HP) e Intel. O primeiro processador da série foi apresentado em Julho de 2002, tendo por finalidade suprir as necessidades do mercado de servidores de alto desempenho. Processadores Itanium II também são utilizados no projeto Thunder e no projeto TeraGrid, sendo estes, atualmente, o 5^o e o 22^o maiores supercomputadores do mundo [Seager, 2004].

2.1. Evolução

Desde o lançamento de Itanium II diversas versões do processador foram implementadas. À seguir, será visto a evolução dos modelos, começando pela descrição do primeiro modelo Itanium.

- **Itanium:** O primeiro modelo da série foi apresentado em Junho de 2001. Desenvolvido em tecnologia de 180nm, o *chip* funcionava com frequências de 766 e 800 MHz, com a possibilidade de escolha de 2 ou 4 MB de *cache* L3. Quando ao desempenho, o processador conseguiu apresentar resultados melhores no modo IA-64 somente em relação aos processadores da família x86 com mesma frequência de *clock*. Já na utilização de instruções x86 o desempenho foi ainda pior: 1/8 de um x86 similar. Para estes casos o problema estava na alta latência do terceiro nível de *cache*, fazendo um mau uso do barramento da memória. A frequência do processador também desapontou em relação aos modelos semelhantes do mercado da época.
- **McKinley:** O modelo McKinley foi a primeira versão de Itanium II apresentado em Julho de 2002, sendo implementado em processo de 180 nm. Lançado com frequências de 900 MHz e 1 GHz e *caches* de 1.5 MB e 3 MB, a versão possuía suporte de *hardware* para as instruções de desvio longas do conjunto IA-64. Já o desempenho se apresentou muito mais lento do que os processadores da família x86 existentes. A performance de McKinley foi semelhante a um Pentium II com frequência equivalente a 2/3 da velocidade do *clock* do processador.
- **Madison:** Inicialmente apresentado em Junho de 2003 Madison possuía 3 versões: 1.3 GHz com 3 MB de *cache*, 1.4 GHz com 4 MB de *cache* e 1.5 GHz com 6 MB de *cache*. Implementado em um processo de 130 nm, o processador possui 374 mm² de dimensão. O consumo de energia permaneceu inalterado em relação ao modelo McKinley de 130 watts. Em Setembro de 2003, foram disponibilizadas as versões de 1.4 GHz com 1.5 MB de *cache*. Uma versão de 1.6 GHz com 6 MB de *cache* e um versão de 1.4 GHz com 3 MB de *cache* foi lançada em Abril de 2004. Já em Novembro de 2004 foi apresentada a versão de 1.6 GHz com 9 MB de *cache*.
- **Hondo:** Hondo foi anunciado pela HP em Fevereiro de 2003 e lançado no início de 2004, sendo caracterizado por um módulo dual conhecido como mx2 (dois processadores em um único módulo). O processador consiste em dois núcleos Madison com 32 MB de *cache* L4 numa área semelhante ao processador normal. Hondo está sendo disponibilizado apenas pela HP, processando, atualmente, numa frequência de 1.1 GHz com 4 MB de *cache* L3 para cada processador.
- **Deerfield:** Deerfield foi apresentado em Setembro de 2003. Ele possui 1.5 MB de *cache* e frequência de 1 GHz. Este é o primeiro processador Itanium de baixa voltagem, utilizando 62 watts.
- **Fanwood:** Fanwood foi apresentado em Novembro de 2004. Suas versões incluem modelos com 1.6 GHz e 3 MB de *cache* L3 e 200 MHz ou 266 MHz de FSB (*Front Side Bus* - barramento de dados que conecta a CPU aos demais dispositivos). Além disso ele possui uma baixa voltagem para o modelo com 1.3 GHz, 3 MB de *cache* L3 e 200 MHz de FSB.

2.2. Características dos modelos comerciais atuais

Os modelos atuais disponíveis para o Itanium II, segundo a Intel [Intel, 2005b] são divididos em 3 categorias: *Multi Processor* (MP)/Madison, *Dual Processor* (DP)/Hondo e *Low Voltage* (LV)/Fanwood. As principais características podem ser descritas na Tabela 1.

Tabela 1: Descrição do processador *Itanium II*

Característica	MP	DP	LV
Cache L1	32KB	32KB	32KB
Cache L2	256KB	256KB	256KB
Cache L3	3, 4, 6 e 9MB	1.5 e 3MB	1.5 e 3MB
Clock	1.30 a 1.60 GHz	1.40 a 1.60 GHz	1.00 a 1.30 GHz
Chipset	Intel E8870	Intel E8870	Intel E8870
Barramento - 128 bit	400 MHz	400 ou 533 MHz	400 MHz
Vazão E/S	6.4 GB/sec	6.4 GB/sec	6.4 GB/sec
Consumo	130W	99W	62W
Plataforma servidora	SR870BN4	SR870BH2	SR870BH2

Cada uma das versões possui determinadas características que permitem explorar um conjunto de aplicações [Intel, 2005a]. MP é indicada para o multiprocessamento de aplicações voltadas para banco de dados, aplicações inteligentes, ERP (*Enterprise Resource Planning*) e HPC (*High Performance Computing*). Já a implementação DP busca explorar uma computação mais técnica como HPC para *clusters*, servidor de rede e gerenciamento de sistemas de segurança. Para LV a preocupação maior está no baixo consumo e custo do servidor, possibilitando utilizar um sistema dual.

3. Arquitetura IA-64

Os modelos dos processadores Itanium estão baseados numa arquitetura de 64 *bits*. O objetivo desse modelo é gerar uma arquitetura que supere a era RISC, utilizando um conjunto de instruções longas (VLIM - *Very Long Instruction Word*). Isto significa que barramento, endereçamento, registradores, dados, instruções e operações podem ser feitas em um determinado momento sobre um conjunto de 64 bits. À seguir, serão vistas algumas características que definem melhor essa arquitetura.

3.1. EPIC

EPIC (*Explicitly Parallel Instruction Computing*) é um paradigma relacionado à forma como são executadas as instruções, onde é possível extrair um conjunto de operações paralelas de uma única instrução VLIW. Este paradigma também é conhecido como independência de arquitetura e busca tirar a complexidade de escalonamento do *hardware* para o *software*.

Em uma arquitetura onde as instruções podem ser executadas fora de ordem é necessário a existência de um decodificador complexo que examina cada instrução do fluxo de execução e verifica quais delas podem ser operadas em paralelo através das unidades de execução disponíveis. Por exemplo, uma série de instruções distintas cujo resultado não afeta nenhuma das outras instruções como $A = B + C$ e $D = F + G$ pode ser feita em duas unidades de execução diferentes de forma paralela. A possibilidade de extrair o paralelismo em nível de instrução (*Instruction Level Parallelism* - ILP) de um fluxo de instruções é essencial para um bom desempenho de um processador moderno.

Predizer que código pode ou não pode ser feito desta maneira é uma tarefa muito complexa. Em muitos casos as entradas de uma operação são dependentes da saída

da outra, mas somente se alguma outra circunstância é verdadeira. Por exemplo, considerando a modificação no exemplo anterior: $A = B + C$; SE $A == 5$ ENTÃO $D = F + G$. Neste caso, os cálculos permanecem independentes uns dos outros, mas a segunda instrução requer o resultado do primeiro cálculo a fim de saber se ela precisa ser executada.

Nestes casos, o processador pode tipicamente supor que a condição será verdadeira e executar a segunda instrução. Entretanto, uma suposição errada pode causar um desempenho ruim quando o resultado precisa ser descartado e o processador precisa re-executar um trecho para a obtenção da solução correta. Um dos motivos para um excelente desempenho das CPUs modernas se deve a utilização de uma lógica de predição melhor. No entanto, as melhorias que estão sendo introduzidas atualmente já não são mais tão significativas.

A arquitetura IA-64 aposta no compilador para esta tarefa. Antes que o programa seja executado, o compilador examina o código e faz as mesmas suposições que seriam feitas em tempo de execução. Uma vez decidido o fluxo de execução a ser executado, o compilador reúne as instruções que podem ser feitas em paralelo, empacota-as em uma instrução maior, e armazena-as no formato VLIW.

Deixar esta tarefa para o compilador ao invés de um componente específico do processador tem diversas vantagens. Primeiramente, o compilador pode gastar um tempo consideravelmente maior para examinar o código, o que geralmente não pode ocorrer em um processador devido à rapidez com que ele precisa executar. Assim, a versão do compilador pode também ser consideravelmente mais exata do que a versão gerada pelo processador. Segundo, os circuitos de predição são completamente complexos, assim a predição em tempo de compilação reduz enormemente essa complexidade na execução. Não é necessário examinar qualquer instrução, simplesmente ela é quebrada, distribuindo-se as partes para os componentes funcionais. Terceiro, fazer a predição no compilado não é um custo incluído na execução.

O aspecto negativo é de que o comportamento de um programa em tempo de execução nem sempre é óbvio em relação ao código gerado, e pode variar consideravelmente dependendo dos dados reais que estão sendo processados. A lógica de processar instruções fora de ordem pode não ser a mesma em um processador devido às decisões tomadas com base nos dados obtidos em tempo de execução e que o compilador simplesmente supõem. Isso significa que é possível para o compilador cometer um erro de predição mais frequentemente do que a lógica utilizada no processador. O projeto de VLIW confia assim pesadamente no desempenho dos compiladores, diminuindo a complexidade do processador e aumentando a complexidade do *software* do compilador.

3.2. Registradores

A arquitetura IA-64 possui um amplo conjunto de registradores. São 128 registradores de 82 bits para armazenar números em ponto flutuante e 128 registradores de 64 bits para inteiros. Aliado a isso, a arquitetura dispõe de um mecanismo de rotação que é controlado pela pilha de registradores. Além dos mecanismos típicos de *spill/fill* ou mecanismos de janela usados em outros processadores, o processador pode rotacionar um novo conjunto de registradores para parâmetros de uma nova função ou para variáveis temporárias. O mecanismo de rotação de registradores combinando com a atribuição é bastante eficiente para a execução de laços de repetição.

3.3. Conjunto de instruções

Em Itanium, uma instrução VLIW pode executar até três instruções de 41 bits. Os demais 5 bytes servem para indicar que tipo de instruções foram empacotadas. Enquanto um

típico VLIW atribui sub-instruções específicas relacionadas a uma unidade funcional a uma instrução, Itanium suporta diversos mapeamentos para permitir mais possibilidades de combinação em uma instrução, o que inclui um balanceamento entre as execuções seriais e paralelas. Das 32 combinações possíveis são utilizadas 24 combinações de instruções. Existe também a possibilidade de adicionar mais mapeamentos em futuras versões da arquitetura para as combinações não definidas ainda.

O Itanium tem individualmente registradores de predicado que podem ser ativados para causar um tipo de execução que não emite resultados para nenhuma instrução. Um Itanium novo, quando inicializado, não possui realmente todas as funcionalidades das instruções. Uma carregador-ROM conhecido como EFI (*Extensible Firmware Interface*) é carregado com código adicional na memória para definir estas instruções, e executando outras configurações, tais como escolher a modalidade da execução do processador (64-bit ou 32-bit.) Este projeto permite que um sistema de Itanium seja desdobrado com potencialidades diferentes dependendo dos índices do programa de EFI.

3.4. Suporte a instruções IA-32

As instruções IA-32 foram mapeadas para às unidades funcionais do Itanium. À fim de suportar essas instruções, Itanium pode escolher entre os dois modos (IA-32 ou IA-64) através de uma instrução de desvio. Entretanto, devido ao fato de que desde o início Itanium foi desenvolvido para obter desempenho nas instruções EPIC, e porque não há nenhuma vantagem para a execução fora de ordem das instruções de IA-32, o código apresenta uma severa queda de desempenho comparada à modalidade IA-64, ou em relação à série Pentium de processadores. Por exemplo, as unidades funcionais de Itanium não ativam automaticamente *flags* de desvio em operações com inteiros e não possuem suporte intrínseco para múltiplas cargas de memória desalinhas.

Há também emuladores em *software* que estão disponíveis para Windows e Linux. Estes emuladores de *hardware* obtém um desempenho aproximado 50%. O emulador IA-32 para Windows está sendo disponibilizado pela Microsoft, enquanto o emulador de Linux pode ser adquirido em alguns desenvolvedores como Novell. Dado o desempenho superior do emulador de software, houve algumas especulações de que a Intel iria remover a simulação IA-32 dos futuros processadores Itanium. Entretanto, o *hardware* de IA-32 representa menos de 1% dos transistores de um Itanium II. Assim, o ganho representa muito pouco para que esta estratégia seja adotada.

3.5. CPU Itanium

A Figura 1 apresenta a descrição da arquitetura do processador Itanium. Os blocos mais externos representam a *cache* L2 e L3 que se comunicam entre si e entre o barramento de memória. A *cache* L2 fornece informações para a *cache* L1 de instruções e dados. Antes das instruções irem para uma das unidades de inteiros, ponto-flutuantes, desvios, ou acessos a memória, representados respectivamente na figura pelas letras I, F, B e M, elas são armazenadas numa fila de instruções.

As unidades citadas anteriormente fazem uso da pilha de registradores com o qual é possível acessar os bancos de registradores e operações de predicado e desvio de registradores, e desta forma, realizar as operações determinadas pelas instruções. O tamanho do *pipeline* é de 10 estágios, podendo ser executadas até 6 instruções simultâneas [Berezaga, 2000]. A arquitetura oferece ainda uma unidade de controle e decodificação de instruções IA-32.

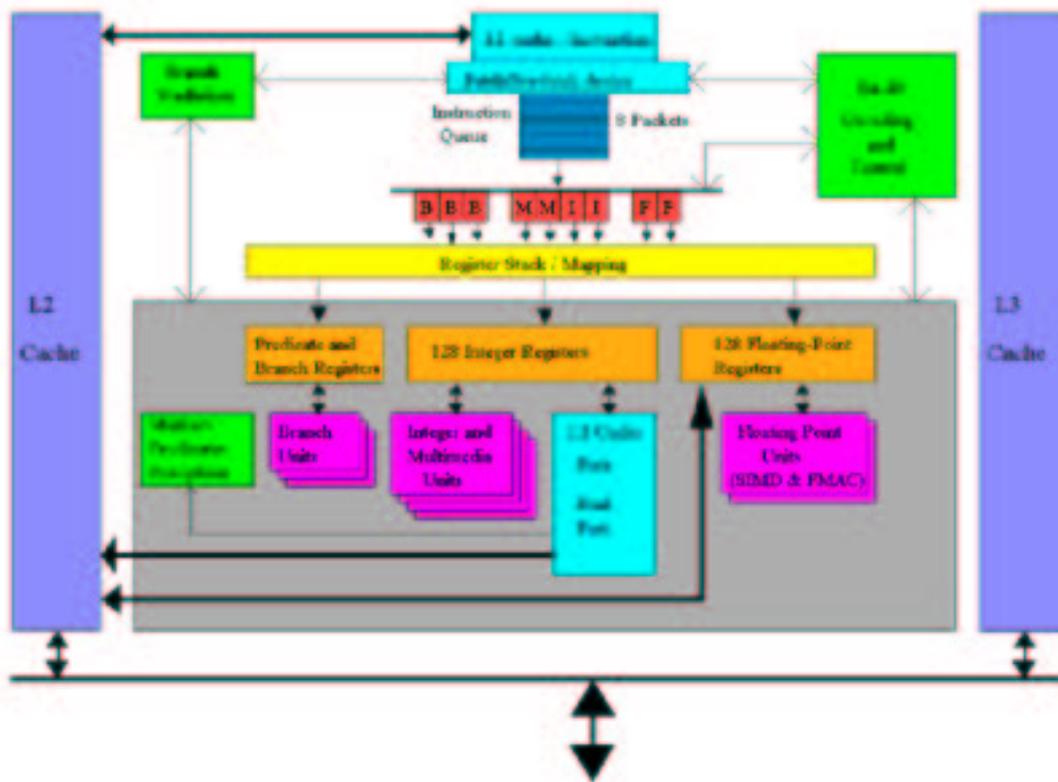


Figura 1: Descrição da arquitetura IA-64.

4. Implementações futuras

O futuro da família Itanium parece estar baseado em *chips* multiprocessados como demonstram os projetos Montecito/Montvale e Tukwila.

4.1. Montecito/Montvale

Montecito é o maior projeto de uma nova versão da família IPF (*Intel's Itanium Processor Family*). O projeto prevê:

- A utilização de dois núcleos por *chip*.
- Um barramento multi-*via* não simultâneo utilizado para a execução de múltiplos fluxos de execução para cada um dos núcleos. Desta forma, duas *threads* não poderão executar simultaneamente, mas o núcleo poderá trocar de *thread* no caso de um evento de alta latência, como uma falha no acesso a *cache* L3. Através dessa técnica a carga de trabalho deve aumentar entre 15 e 35 % em relação à utilização de um único fluxo.
- Um total de 4 fluxos de execução por *chip*.
- 1MB de *cache* L2 para instruções e 256 KB para dados por processador, mais 24 MB de *cache* L3 por *chip*.
- 1.72 bilhões de transistores por circuito.
- O tamanho do *chip* será de 596 mm²
- A frequência do clock ficará entre 2.0GHz e 2.5GHz para os modelos topo de linha. Apesar do modelo Montecito ter sido projetado para uma frequência de até 2.5 GHz, é provável que a frequência fique em torno de 2.2 GHz no máximo.
- Haverá um ajuste avançado de *clock* para o gerenciamento de energia. A frequência do relógio não será fixa, mas ajustada a fonte de energia, o que significa que sobre uma determinada condição, como baixa carga de trabalho, o que gera menos calor enquanto está sendo executado, o *chip* será capaz de aumentar

a frequência. Enquanto sob uma condição ruim, como uma temperatura ambiente excessivamente alta ou uma hélice do ventilador quebrado, ele pode diminuir a frequência e a voltagem. Uma baixa carga de trabalho pode ocorrer, também, quando há um intensivo cálculo envolvendo inteiros, ou em aplicações comerciais como as que envolvem banco de dados. Neste caso o *chip* poderá aumentar a frequência em torno de 10%.

- Baixo consumo de energia e dissipação térmica mais rápida dos capacitores. Apesar do consumo dos transistores e da alta frequência de *clock* a potência sugerida é de 100 W. Isto é possível graças a utilização de diferentes tipos de transistores.
- Correção avançada para erros de *cache*, tornando as operações em missões críticas mais seguras.
- Melhoramento do barramento de dados (FSB), com três vezes mais capacidade do que os barramentos existentes. Não se tem informações se isto se aplica a cada um dos nós ou ao conjunto dos nós. Todavia, a vazão do sistema por nó será de pelo menos 21 Gbytes por o segundo.
- Conectar dispositivos já existentes através do FSB.

4.2. Tukwila

Tukwila é uma futura geração da família IPF, que seguirá o Itanium II e o projeto Montecito. Existe uma expectativa de que ele chegue ao mercado comercial em 2007. Enquanto suas características não forem publicadas em detalhe, acredita-se que ele utilize múltiplos núcleos idênticos (*Symmetric MultiProcessing* - SMP) em um mesmo circuito integrado conhecidos como CMP (*Chip-level multiprocessing*). CMP permite que os processadores compartilhem o segundo e terceiros níveis de *cache*, o que possibilita uma utilização maior do paralelismo em nível de *threads* ou o paralelismo em nível de instruções. Os projetistas deste projeto baseiam-se nos esforços aplicados no modelo EV8 de DEC Alpha, que estava focado sobre SMT (*Simultaneous multithreading*).

É provável que o *chip* tenha 4 processadores. Ele poderá estar habilitado para uma execução sequencial tendo uma máxima frequência de *clock* enquanto os demais processadores permanecem desligados. Desta forma ele pode ser configurado para fornecer alta performance em múltiplos fluxos de execução ou alta performance em um único fluxo de execução, enquanto permanece dentro dos limites térmicos estabelecidos.

5. Considerações finais e conclusão

Evoluções de hardware são quase previsíveis, consequência direta da famosa lei de Moore. Porém cada vez há obstáculos maiores a serem superados. Como obter cada vez mais performance gastando menos energia? Como superar os problemas de superaquecimento?

O superaquecimento dos processadores é uma das maiores barreiras para o aumento da velocidade. A utilização de tecnologias cada vez melhores para a obtenção de frequências maiores está sendo revista, incluindo uma significativa preocupação com a dissipação do calor. Diversas arquiteturas estão enfrentando este tipo de problemas, enquanto a evolução dos componentes não apresenta soluções imediatas para este caso. Uma das possibilidades está na busca de novas soluções, completamente diferentes das utilizadas atualmente.

Neste trabalho foram apresentadas algumas características do processador Itanium II, especialmente as características evolutivas e conceitos aplicados. Apesar de todas as

dificuldades envolvidas, tanto em questões de projeto, como de mercado, existe uma expectativa de que a arquitetura possa ser melhorada com a incorporação de novas tecnologias.

Referências

Berezaga, M. D. (2000). Arquitetura ia-64 (epic). <http://www.inf.ufrgs.br/proctpar/disc/inf1191/trabs/992/IA64/ia64.html>.

Hiremane, R. (2005). From Moore's Law to Intel Innovation–Prediction to Reality. Technical report. <http://www.intel.com/technology/magazine/silicon/moores-law-0405.pdf>.

Intel (2005a). Intel Itanium 2 Processor. Technical report. <http://www.intel.com/design/itanium/itanium/Itanium2ProdBrief.pdf>.

Intel (2005b). Intel Itanium 2 Processor for Demanding Enterprise and Technical Applications. Technical report. http://www.intel.com/business/bss/products/server/itanium2/quick_reference.pdf.

Santos, R. R. and Santos, T. G. S. (2005). Projeto e Implementação de Arquiteturas Superescalares. In *Anais, Quinta Escola Regional de Alto Desempenho*, Canoas. Sociedade Brasileira de Computação - Unilasalle / UFPel / UCPel / UCS.

Seager, M. (2004). M&ic thunder. Technical report. <http://www.llnl.gov/linux/thunder/>.