

Introdução à Arquitetura VLIW

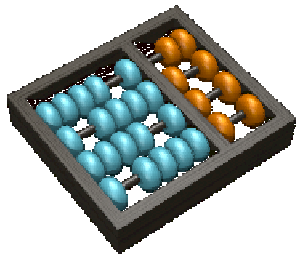
Fundamentos e Perspectivas

Autor:

André Augusto Ciré (015463)

Prof. Rodolfo Jardim Azevedo

MO401 – Arquitetura de Computadores I



UNICAMP

Tópicos da Apresentação

- Preliminares
- Perspectiva Histórica
- Fundamentos do VLIW
- Métodos de Escalonamento
- Exemplos de Máquinas VLIW
- O Futuro
- Conclusões

Preliminares

- Desempenho em máquinas monoprocesadas
- Limitações no modelo von Neumann
 - Seqüencialidade de instruções
- *Instruction Level Parallelism*
 - *Pipelining*
 - *Arquiteturas superescalares*
 - *Arquiteturas VLIW*

Preliminares

□ *Pipelining*

- Há anos no mercado.
- Poucas otimizações ainda são efetivas

□ Arquiteturas Superescalares

- Ganhos de desempenho significativos :-)
- Complexidade de hardware crescente :-(

□ Abordagem Alternativa: *VLWI*

- Simplificação no Hardware
- Complexidade no compilador (Software)

Perspectiva Histórica

- Surgimento da Microprogramação
 - Turing (1946) e Wilkes (1951)
- *Microprogramação Horizontal*
 - Base do VLIW
 - Microinstruções: operações associadas a unidades funcionais
 - Macroinstruções: palavras *longas*

Perspectiva Histórica

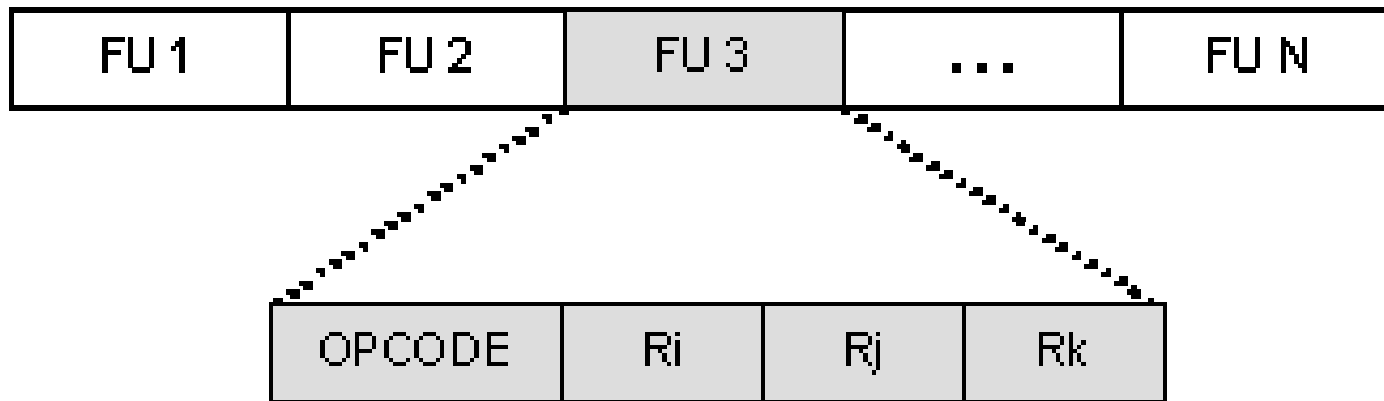
- Como gerar palavras longas a partir de instruções curtas?
 - *Trace Scheduling* (Fisher 1979)
- Surgimento da Arquitetura VLIW no mercado
 - Fisher funda a *Multiflow* (1984)
 - Bob Rau funda a *Cydrome* (1984)
- Dificuldades comerciais
 - Preço elevado de memória

Fundamentos do VLIW

- Redução do *número de instruções* da aplicação
- Uma instrução encapsula várias operações
 - *Microcódigo Horizontal*
 - Palavra VLIW contém diversos *opcodes*.
- Escalonamento **estático**
 - Compilador possui especificidades da arquitetura
 - Considera sempre pior caso das execuções

Fundamentos do VLIW

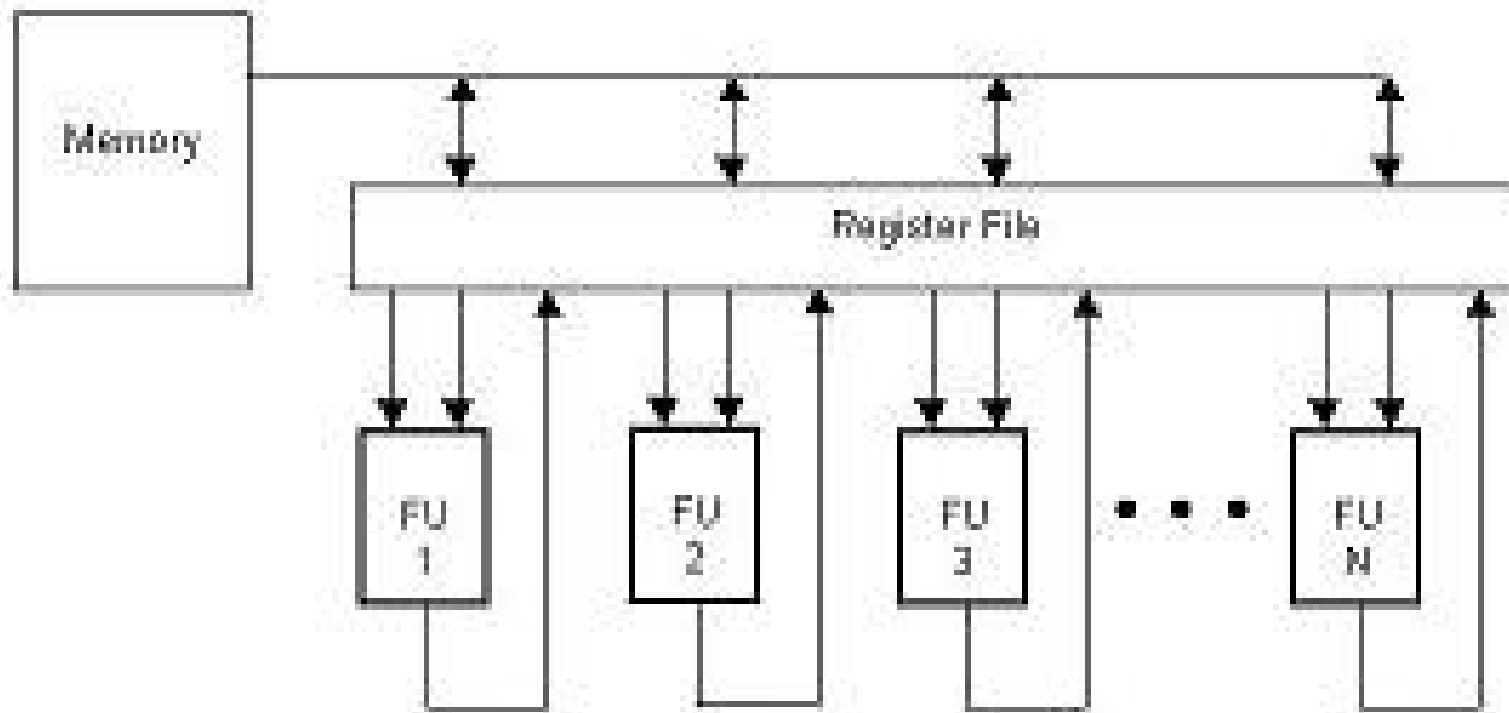
Palavra VLIW



Fonte: Sérgio Akira Ito

Fundamentos do VLIW

Arquitetura VLIW teórica



Fonte: Sérgio Akira Ito

Escalonamento Estático

- ❑ Crucial para o desempenho
- ❑ *Escalonamento por Blocos-Básicos*
 - Método tradicional, pouco eficaz
- ❑ *Escalonamento por Blocos-Básicos Estendido*
 - Compreende grande gama de algoritmos
 - Mais utilizados

Escalonamento Estático

□ *Trace Scheduling*

- Identificação de *traces* mais prováveis
- Portas de Entrada e Saída
- Necessidade de armazenar todos os *traces* já identificados

□ *Superblock Scheduling*

- *Superbloco*: *traces* com uma única entrada e várias saídas
- *Tail Duplication*

Escalonamento Estático

□ *Hyperblock Scheduling*

- Vários caminhos de execução considerados simultaneamente
- Presença de *Predicação* entre caminhos

□ *Tree Decision Scheduling*

- *Fluxo de Controle* representados por *árvores*
- Otimizações por predicações

Exemplos de Máquinas VLIW

□ *Multiflow TRACE*

- Conjunto de instruções *RISC-like*.
- Palavras de instrução com 1024 bits
- 28 operações por instrução
- Sem *cache* de dados

□ Cydra 5

- Palavras de instrução com 256 bits
- 6 operações por instrução
- *Mini-supercomputadores*

Exemplos de Máquinas VLIW

□ Intel Itanium

- *EPIC (Explicit Parallel Instruction Computing).*
- Arquitetura IA-64.
- Palavras como pacotes de 128 bits.
- 3 operações por instrução, 2 inst. por ciclo.
- Complexidade de Hardware (!!!)

Exemplos de Máquinas VLIW

- Transmeta Crusoe
 - Aplicações móveis
 - Baixo consumo de energia
 - Compatibilidade com máquinas a partir de emulação
 - *Code Morphing*
 - Palavra: *moléculas* com 4 operações (*átomos*)

O Futuro

- Foco das pesquisas:
 - *Compiladores*
 - *Arquitetura*
 - *Incompatibilidade de código-objeto*

- Centros de pesquisa ativos
 - Destaque para IBM

- Aplicabilidade em Sistemas Dedicados

Conclusões

- ❑ Idéia-chave: complexidade no compilador, não no hardware.
- ❑ Inviabilidade Comercial: compatibilidade do código-objeto
- ❑ Pesquisas trazem grandes avanços e promessas para a tecnologia.