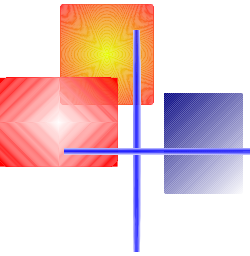
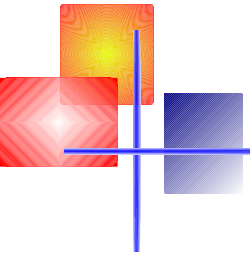


– Software Pipelining –

Uma técnica para paralelização de Loops

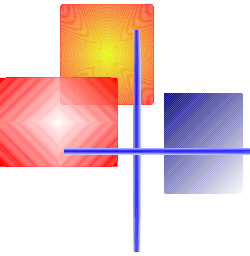
Carla Geovana Macário

Junho, 2006



Roteiro

- Visão Geral
- Suporte necessário
- Algoritmos existentes
- Estado Atual
- Conclusões



O que é?

- Técnica para prover paralelismo à execução de loops
- Busca sobrepor operações de diferentes iterações aumentando o paralelismo

Loop Unrolling X Software Pipeling

- Loop Unrolling
  - Desenrolamento do loop: programa seqüencial

```
for i from 1 to 100 do
    a[i] := a[i] + b[i]
    i := i + 1
```

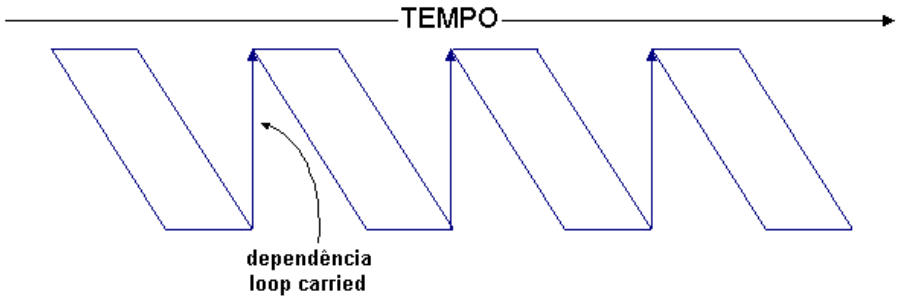
Loop Original

```
for i from 1 to 100 do
    a[i] := a[i] + b[i]
    a[i+1] := a[i+1] + b[i+1]
    a[i+2] := a[i+2] + b[i+2]
    a[i+3] := a[i+3] + b[i+3]
    i := i + 4
```

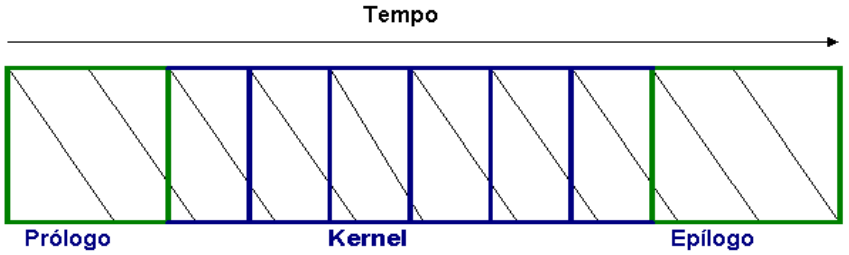
Loop Unrolling

Loop unrolling X Software Pipeling

- Software Pipeling
  - Reforma do loop para agrupar operações que possam ser executadas em paralelo
  - $\{ABC\}_n$ é semanticamente igual a $A\{BCA\}_{n-1}BC$
  - BCA é o novo corpo do loop (kernel)

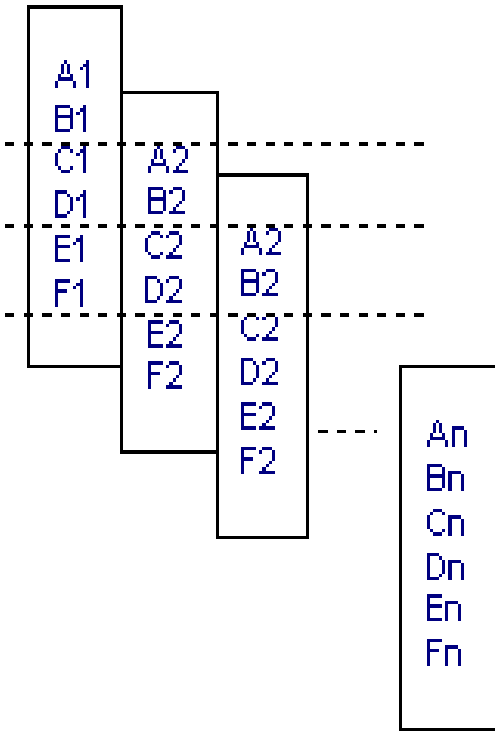


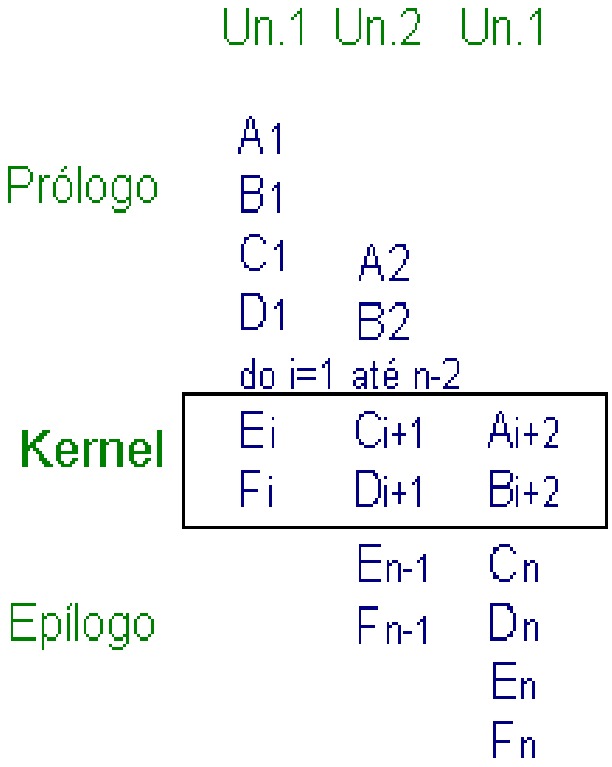
Loop sem software pipelining



Loop com software pipelining

Software Pipeling





Visão Geral

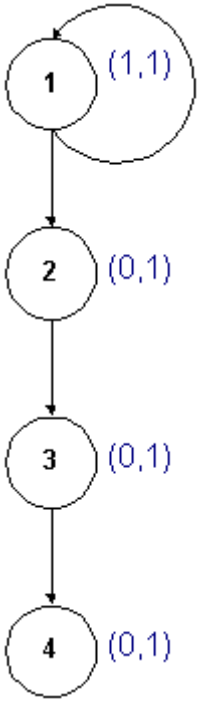
- Derivado do trabalho de [Patel & Davidson 1976]
- Rau e Aiken desenvolveram o 1o. Compilador que tirava proveito das arquiteturas paralelas.

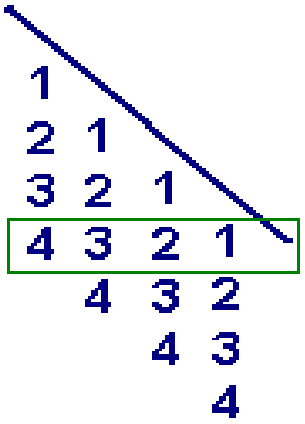
- **Objetivo principal:**
  - Identificar o novo corpo do Loop (kernel)
-
- Restrições a serem respeitadas nesta atividade:
  - Dependência de dados entre iterações
  - Uso de recursos

Suporte ao Sw Pipelining

- Grafo de dependência de dados (DDG)
  - Fornece a dependência entre operações e as latências existentes entre elas

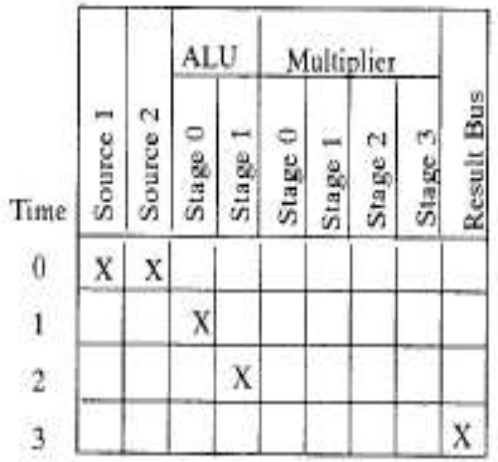
```
for i from 1 to n do
        a[i+1] := a[i] + 1
    b[i] := a[i+1] /2
    c[i] := b[i] +3
    d[i] := c[i]
    i := i + 1
```



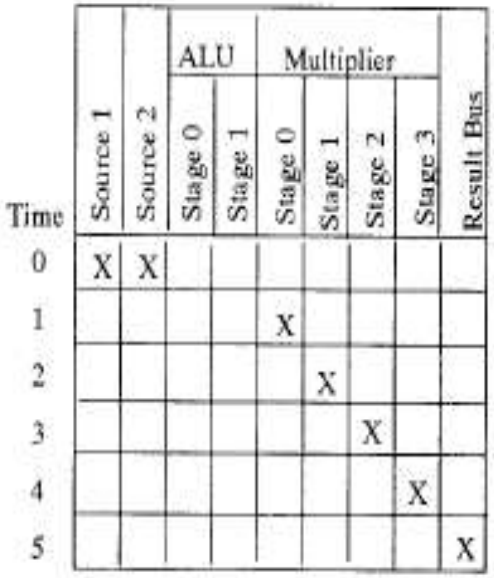


Restrição de Recursos

- 2 operações O1 e O2 estão em conflito se requerem o mesmo recurso ao mesmo tempo
- Tabela de reserva de recursos:
  - usada para checar conflitos de recursos durante o



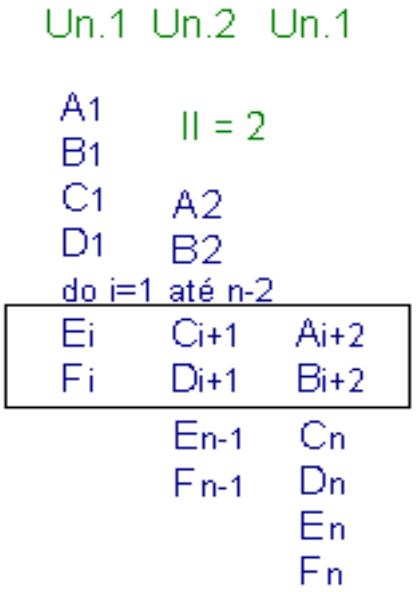
Time	Source 1	Source 2	ALU		Multiplier				Result Bus
			Stage 0	Stage 1	Stage 0	Stage 1	Stage 2	Stage 3	
0	X	X							
1			X						
2				X					
3									X

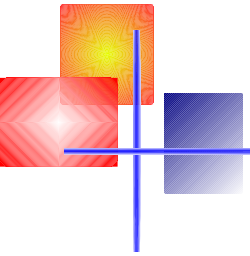


Time	Source 1	Source 2	ALU		Multiplier				Result Bus
			Stage 0	Stage 1	Stage 0	Stage 1	Stage 2	Stage 3	
0	X	X							
1					X				
2						X			
3							X		
4								X	
5									X

Intervalo de Iniciação (II)

- Intervalo de tempo necessário entre cada início de iteração
- Quanto menor o II, maior o throughput (maior paralelização)
- Dependente da restri ; e da dependência de dados.





Calculando o *II mínimo (MII)*

- Determinar a restrição de recursos é simples
- Determinar o escalonamento considerando a restrição de dependência de dados é um problema NP-Completo
- Técnicas para isso:
  - Enumeração exaustiva de todos os ciclos simples
  - Todos os pares do algoritmo de menor caminho
  - Algoritmo iterativo do menor caminho: fecho transitivo
  - Programação linear

Classificação dos Algoritmos

Escalonamento Modulo:

- faz um escalonamento e usa movimentação de código para melhorar o kernel encontrado

Identificação do Kernel:

- Promove um desenrolamento do loop para identificar o conjunto de instruções que podem compor o kernel

Escalonamento Modulo

- Proposto por [Rau & Glaeser 1981]
- Principais algoritmos desenvolvidos:
  - Redução Hierárquica [Lam 1988]
    - Variável de expansão
    - Bastante difundido, sendo uma referência até hoje
  - Escalonamento Predicado
    - Arquivo rotacional

Identificação do Kernel

- Perfect Pipeling [Aiken & Nicolau 1988]
  - combina movimentação de código com escalonamento para melhorar o paralelismo, buscando adequar o problema à arquitetura em uso
- PetriNet model
  - uma variação do anterior, usando redes de petri para resolver o problema de reconhecimento do kernel
- Vegdahl's
  - um método exaustivo que considera todas as soluções possíveis de escalonamento para escolher a melhor delas.

Identificação do Kernel

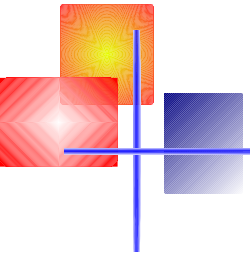
- Enhaced Pipeline
  - extensão do Perfect Pipelining, desenvolvido para tirar proveito de arquiteturas que suportam a execução multi-predicadas
  - usa movimentaçao de código, mas retém o corpo do loop, deixando de ser necessária a identificação do kernel

Comparação dos Algoritmos

Algoritmo	*Classe*	*Contribuição*	*Vantagem*	*Desvantagem*
Reduçao hierárquica.	ME	- aplicável a todos os loops - expansão de variável	Gera escal. próximos de ótimo	Expansão de código
Escalonamento Predicado	ME	- uso de arquivos rotivos para alocação de registradores	Escalonamento após análise de todas as operações	Execução de todas as instruções
Perfect Pipeline	IK	- adequar o escalonamento aos recursos disponíveis	Gera escalonamentos ótimos após movimentação do código	Usa técnicas ad hoc que limitam sua eficiência
Rede de Petri	IK	- usa rede de Petri para identificar as operações que estão prontas para serem executadas	provê maior formalismo ao Perfetc Pipeling	Dificuldade em encontrar o kernel
Técnica de Vergdahl	IK	- fornecer heurísticas para outros algoritmos	Considera todas as soluções possíveis para o escalonamento	Complexidade exponencial para escalonamento
Enhanced Pipeline	IK	- desenvolvido para tirar proveito de arquiteturas que suportam a execução multi-predicada	Retém o corpo do loop, sem necessidade de identificação do kernel	Difícil entendimento

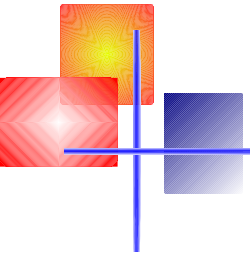
Estado Atual

- Nenhum novo algoritmo de impacto foi proposto.
- Observa-se apenas trabalhos buscando melhorar pontos específicos nos algoritmos existentes
- Exemplo:
  - alocação de registros entre iterações: [Chabin et al. 2005] e [Rong et al. 2005]
  - Redução do tamanho do código: [Zhuge et al. 2003] e o de [Kim et al. 2003]
- Retrospectiva de Lam [Lam 2004]: software pipelining é efetivo em máquinas VLIW sem exigir suporte complicado de hardware .



Conclusão

- Software pipelining é uma técnica importante para melhorar desempenho de programas via paralelismo
- Não é simples promover este paralelismo: problema NP-Completo
- Diversos algoritmos para sw pipelining, poucos de impacto
- Software pipelining dirige o projeto de algumas arquiteturas, mas não exige hardware muito sofisticado
- Área estratégica para investir: Lei Amdahl



Dúvidas?