

Software Pipeline

Sergio Ricardo Souza Leal de Queiroz

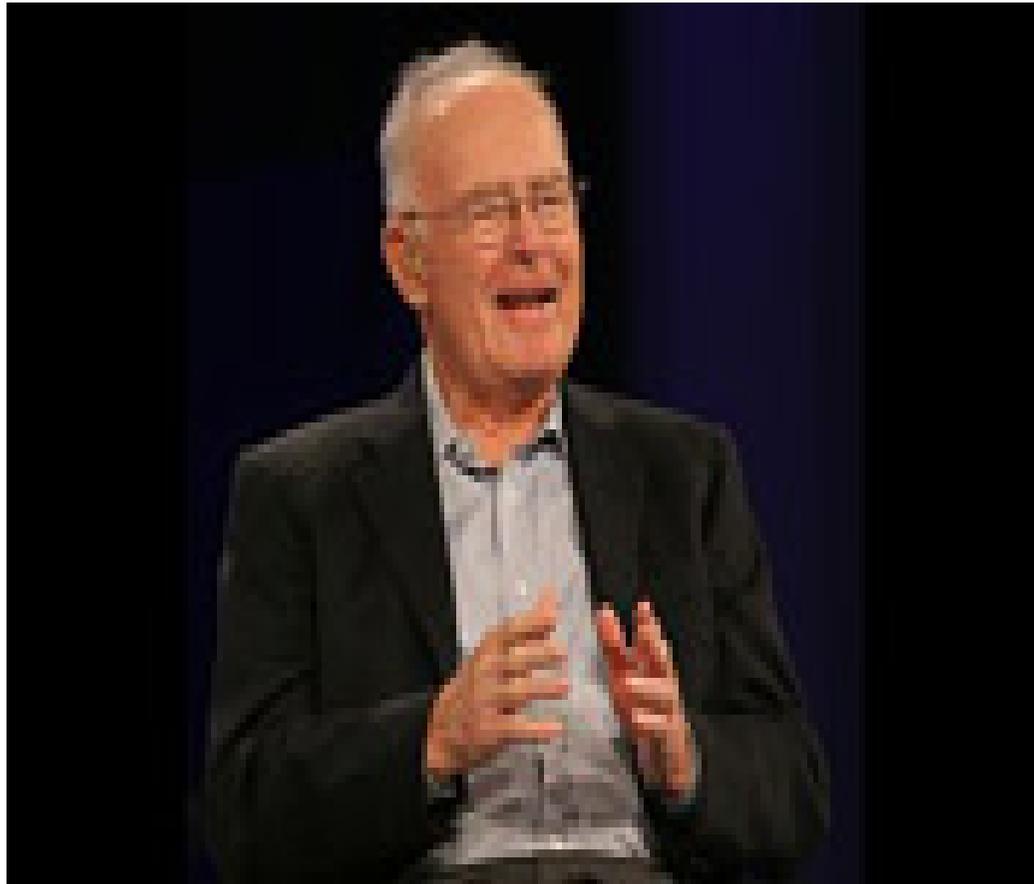
ra: 107070

Arquitetura de Computadores I
MO401 - 1s2010

Prof. Paulo Cesar Centoducatte

Lei de *Moore*

***Gap* entre processadores e memórias**



Software Pipelining - Unrolling Loops

- . Identifica o *Loop* -> *Kernel*
- . Rearranja o *Loop* -> *Loop Unrolling* -> Escalona e Desdobra
- . Novo *Kernel*
- . Calcula o Intervalo de Inicialização (ii)

Intervalo de Inicialização (ii)

Principal elemento de eficiência

- . Pior ii - maior a restrição
- . Melhor ii - relação direta com o nr. de registradores

Restrições e Conflitos

- . Não Modificar a Semântica do Programa
- . Dependência de Dados
- . Número de Registradores
- . Tamanho do código do novo *Kernel*

Escalonamento do Módulo

- . A partir do *Kernel*, um novo *Kernel* é gerado, garantindo que não haja restrições de recursos ou dependência de dados. O início de cada iteração deve seguir o ii.
- . Vantagem: *loop* não precisa ser desenrolado.
- . Desvantagem: instruções devem estar posicionadas de tal forma que todas as iterações sejam identicamente escalonadas.
- . Principal questão: cálculo do ii

Escalonamento do Módulo - Variações

- . **Via redução hierárquica:** escalona separadamente os componentes fortemente conectados.
- . **Módulo Predicado:** gera caminhos diferentes apenas para as instruções dependentes do desvio condicional.
- . Evita a expansão de código pelo renomeamento de registradores.
- . Reduz a quantidade de recursos.
- . Precisa de suporte de *hardware*.

Escalonamento do Módulo - Variações

- . **Enhanced Módulo:** utiliza *if-conversion* para posicionar linearmente as instruções de código predicado. Após o escalonamento renomeia os registradores para as diferentes iterações, e depois restaura a estrutura de desvio, *reverse if-conversion*, inserindo instrução de desvio no código.
- . Simplicidade do método pode ser uma vantagem, mas eleva a complexidade do algoritmo, explodindo o código.
- . **Single-dimension software pipeline (SSP):** transforma um loop multidimensional (Nest= n) em um loop unidimensional (Nest=1). É uma extensão do escalonamento de módulo tradicional.

Identificação do *Kernel*

- . Feita através do desdobramento do loop.
- . Não precisa repetir o escalonamento para diferentes intervalos de inicialização (ii).
- . Elevação do custo do algoritmo.

Primeiro: desenrola o *loop* e verifica dependências.

Depois: Escalona instruções conforme a dependência de dados.

Por último: identifica os blocos idênticos, definindo um novo corpo de *loop* que forma um novo *loop*.

Identificação do *Kernel* - Variações

- . ***Perfect Pipeline***: usado em arquiteturas com modelos mais gerais, resolvendo o problema de arquitetura através de mudança de parâmetros.
- . **Modelos de rede de *Petri***: baseado na teoria de grafos para identificar o *Kernel*, utiliza a renomeação de registradores.
- . **Técnica de *Vegdahl's***: explora todas as soluções possíveis e seleciona a melhor. Impraticável nos códigos reais.
- . ***Multithread***: Modelo mais utilizado. *Threads* concorrentes comunicam-se através de leitura e escrita em variáveis concorrentes. Podem progredir em paralelo ou em um único processador.

Conclusão

- . Paralelização é o novo paradigma da arquitetura.
- . Técnica de *software pipeline* é um desafio para arquitetos e programadores.
- . Irrelevância da lei de *Moore*
- . Programadores e arquitetos devem trabalhar colaborativamente.
- . Desafios nas restrições e conflitos