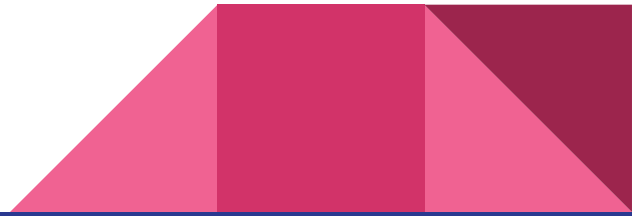
Arquitetura e Organização de Computadores II

Limitações do Paralelismo em nível de instrução e introdução ao multithreading

Prof. Nilton Luiz Queiroz Jr.

Limitações do ILP

- Para explorarmos os limites do ILP é interessante definir um processador ideal;
  - Removem-se todas as restrições sobre o ILP;
  - Se mantém apenas limites quanto ao fluxo de dados;
    - Registradores;
    - Memória;



Limitações do ILP

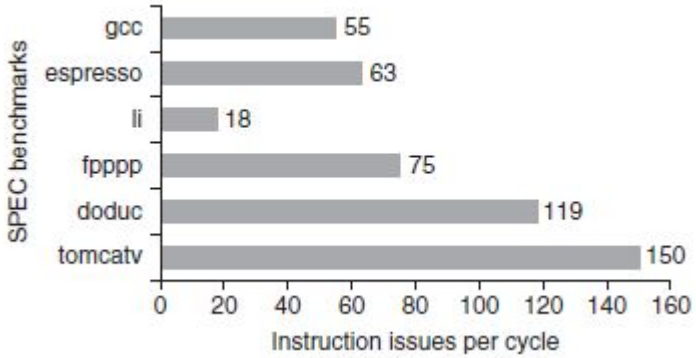
- Suponha um processador com as seguintes características:
  - Renomeação infinita de registradores;
    - Infinitos registradores virtuais para resolver conflitos WAW e WAR;
  - Previsão de desvio perfeita;
    - Todos desvios são previstos com exatidão;
  - Previsão perfeita de salto;
    - Todos saltos saltos previstos perfeitamente;
      - Inclusive os por registradores;
    - Junto a previsão de desvio perfeita é formada uma especulação perfeita;
  - Análise perfeita de alias de endereço de memória;
    - Todos endereços de memória são conhecidos exatamente;
      - Loads podem ser passados à frente do stores quando não tem o mesmo endereço (não evita Hazards WAW e WAR em memória);
  - Caches perfeitas;
    - Todos endereços de memória utilizam um ciclo de clock;

Limitações do ILP

- Examinando esse modelo de hardware com:
  - Despacho de instruções ilimitado;
  - Todas latências das unidades funcionais com 1 ciclo;
  - Sem restrições de quais tipos de instruções podem ser executadas em um ciclo;
    - Infinitas unidades funcionais;

Limitações do ILP

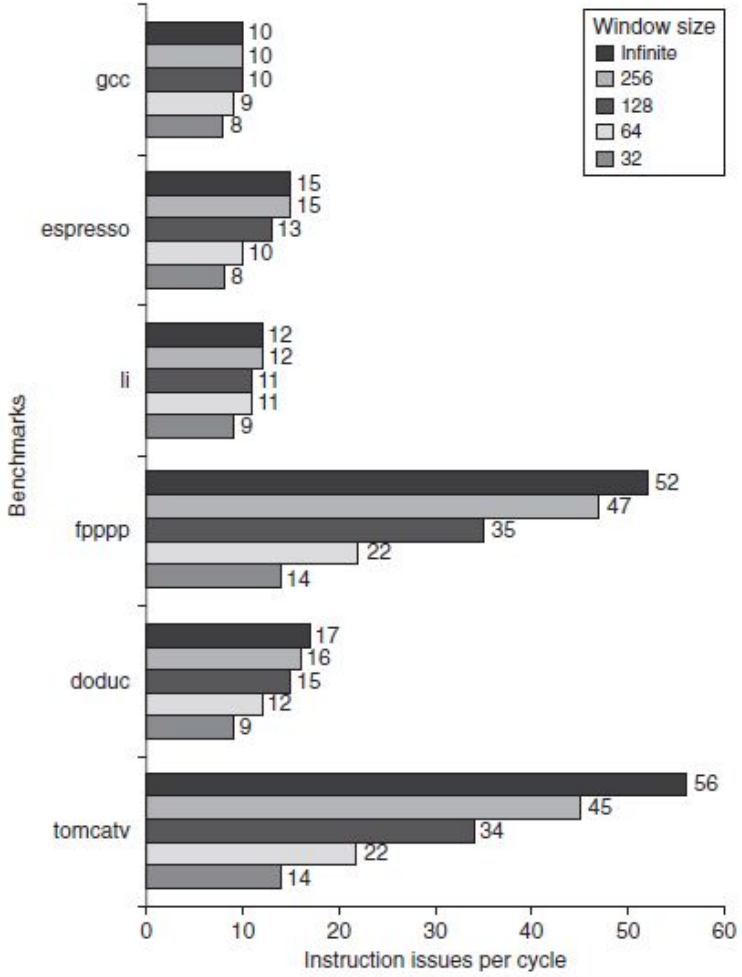
- Experimentos realizados para esse hardware mostram o nível de paralelismo que pode ser alcançado por aplicações reais:



Obs: Os programas fpppp, doduc e tomcatv são programas com grande quantidade de laços, o que faz com que tenham um alto nível de paralelismo a nível de laços

Limitações do ILP

- Agora supondo um processador com níveis ambiciosos de hardware:
  - Até 64 despachos de instruções por ciclo sem restrição de despacho;
  - Previsor de torneio com 1k de entradas e um previsor de retorno com 16 entradas;
  - Desambiguidade perfeita de referências a memória feita dinamicamente;
  - Renomeação de registradores com 64 registradores adicionais de inteiros e PF;
    - Assumindo que as instruções levam 1 ciclo na fase de execução, isso é o suficiente, pois as instruções irão ficar pouco tempo nos reorder buffers;


Window size
Infinite
256
128
64
32
gcc
10
10
10
9
8
espresso
15
15
13
10
8
li
12
12
11
11
9
fpppp
52
47
35
22
14
doduc
17
16
15
12
9
tomcatv
56
45
34
22
14
Benchmarks
0
10
20
30
40
50
60
Instruction issues per cycle

Limitações do ILP

- É possível observar que o tamanho da janela influência mais em programas baseados em ponto flutuante;
  - Janela: quantidade de instruções avaliadas como candidatas para execução;
- Programas baseados em dados inteiros não são tão influenciados pelo tamanho da janela;
  - São as aplicações que têm mais limitações pela previsão de desvios, renomeação de registradores e a menor quantidade de paralelismo;
  - Aplicações mais comuns da atualidade dependem de performance de inteiros;
    - Processamento de transações;
    - Servidores web;

Limitações do ILP

- As principais limitações do ILP são:
  - Hazards WAW e WAR em memória;
    - A renomeação de registradores não elimina os hazards em memória;
  - Dependências desnecessárias;
    - Introduzidas por recorrências ou convenções de geração de código
      - Por exemplo:
        - incremento de um registrador em um loop irá gerar uma dependência ao laço anterior (o registrador depende dele mesmo);
  - Contornar o limite do fluxo de dados;
    - Se a previsão de valores funcionasse com alta precisão, o limite de fluxo de dados poderia ser superado;

Limitações do ILP

- Esses limites são barreiras para exploração do ILP;
- Tentativas de rompê-los geraram frustrações;
  - Pequenas melhorias com grandes aumentos de complexidade, de clock e aumentos desproporcionais na potência;
- Tentar extrair mais ILP se tornou ineficiente;

Multithreading

- O multithreading é uma técnica primária usada para expor mais paralelismo em nível de thread;
- Alternativa encontrada para aproveitar melhor as unidades funcionais de um único processador;

Multithreading

- Compartilha a maior parte do núcleo do processador entre diferentes; threads;
  - Duplica somente os status privados;
    - Program Counter;
    - Registradores;
    - Tabela de páginas;
  - Diferentes das abordagens de multiprocessamento que duplicam o processador como um todo;
- Necessita de mudança de contexto (entre as threads) com alta velocidade;

Multithreading

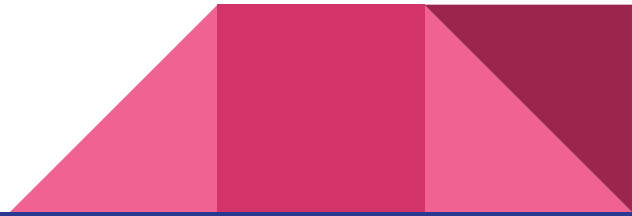
- Três principais técnicas:
  - Granularidade fina;
  - Granularidade Grossa;
  - Simultâneo;

Multithreading

- Granularidade Fina:
  - Alterna os threads a cada clock;
  - Execução intercalada;
    - Troca de thread a cada ciclo;
  - Geralmente faz-se um round-robin, pulando threads em stall;
    - Troca de contexto feita de maneira simples;
  - Principal vantagem:
    - Permite execução de outros threads enquanto uma está parada;
      - Esconde atrasos de throughput;
  - Principal desvantagem:
    - Atrasa a execução dos threads individuais;
      - Desempenho single thread é ruim
  - Exemplos:
    - Sun T1;
    - GPUs Nvidia;

Multithreading

- Granularidade Grossa:
  - Troca de thread somente em stalls custosos;
    - Por exemplo:
      - Falhas no nível 2 ou 3 da cache;
  - Vantagem:
    - Menor atraso para cada thread;
  - Desvantagem:
    - Limitação de contornar as falhas de throughput;
      - Principalmente stalls menores;
    - Quando ocorrem stalls curtos existirão bolhas no pipeline;



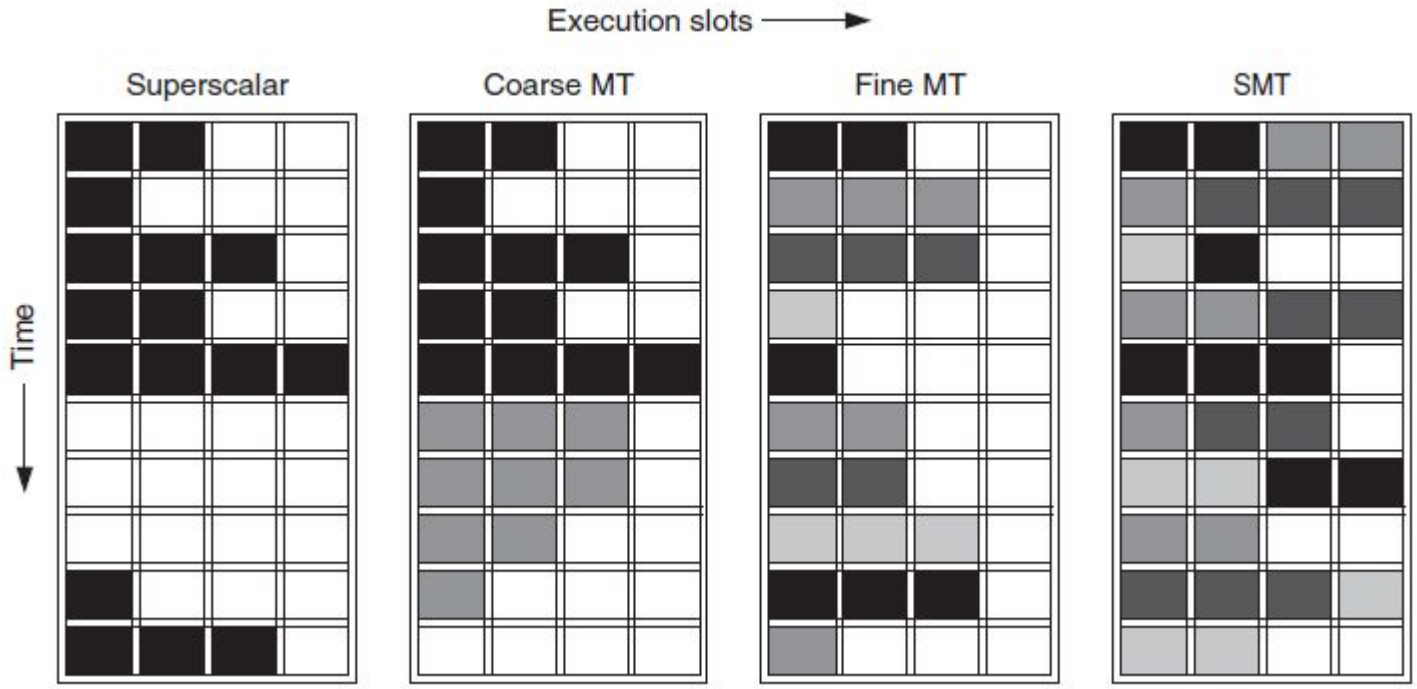
Multithreading

- Multithreading Simultâneo (Simultaneous multithreading -SMT):
  - Mais comumente implementado;
  - É basicamente um multithreading de granularidade fina implementada em um processador com múltipla emissão e dinamicamente escalonado;
  - Explora paralelismo em nível de thread e também paralelismo em nível de instrução;
  - Sua principal motivação é que processadores com despacho múltiplo geralmente tem mais hardware do que o necessário que uma única thread pode usar;
  - A renomeação de registradores e escalonamento dinâmico permitem que múltiplas instruções de threads independentes sejam emitidas sem considerar as dependências entre elas;
  - Não troca os recursos a cada ciclo;
    - Deixa que o hardware cuide da associação das instruções a seus slots e da renomeação de registradores com suas próprias threads;

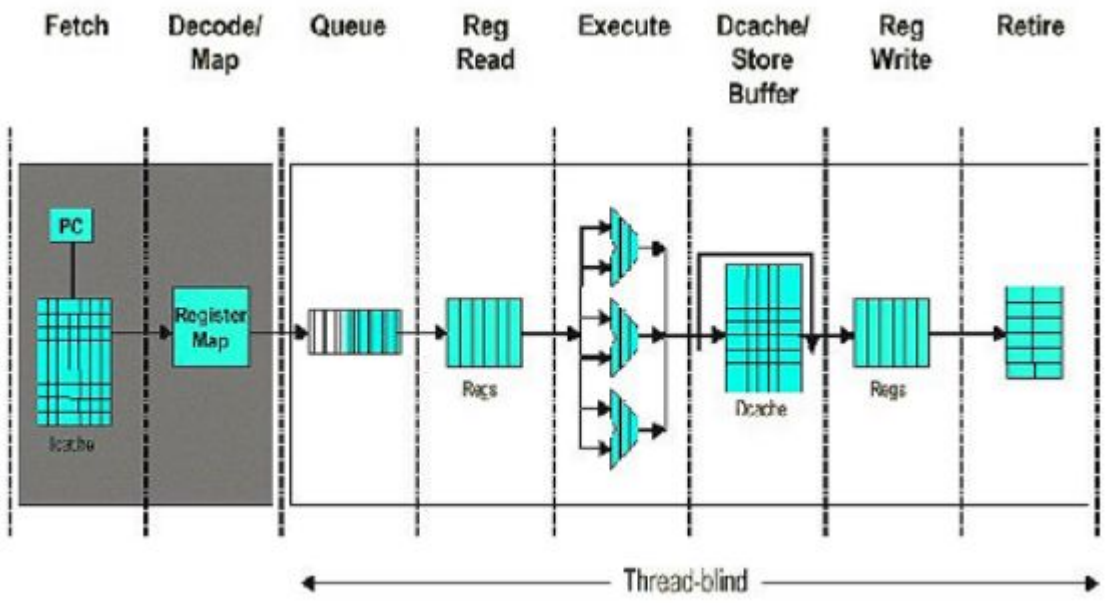
Multithreading

- Requer alterações como:
  - Múltiplos PCs;
  - Pilhas de retornos separadas para cada thread
  - Identificação da thread em cada entrada do branch target buffer (BTAC)
  - Um banco de registradores grande, com registradores para as threads e registradores adicionais para renomeação
  - Dois estágios no pipeline para acesso aos registradores
    - Necessidade de mais um nível de bypass;
    - Aumento da distância entre busca e execução
      - Falhas na previsão de desvio se tornam mais caras (1 ciclo a mais)
    - Aumento do tempo minimo que um regsitrador está ocupado;

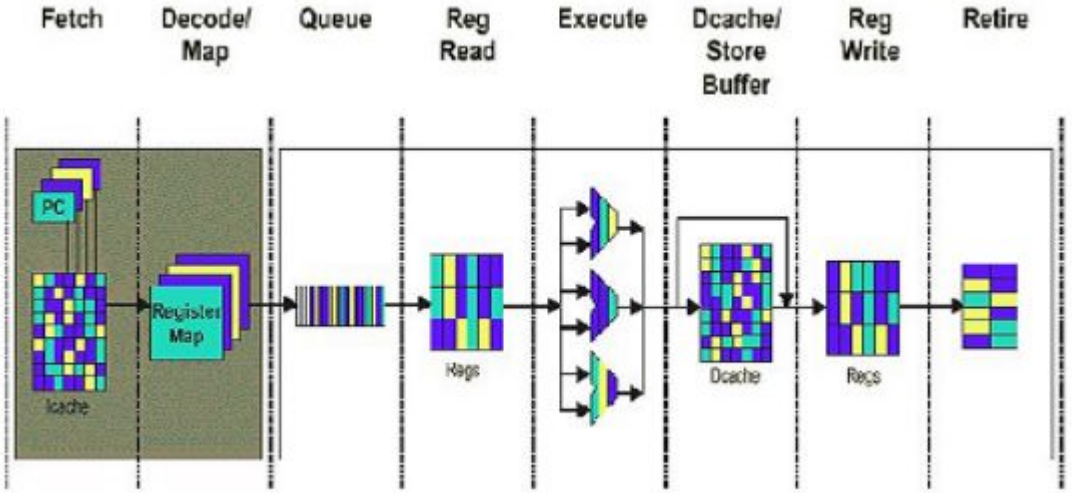
Multithreading



Processador Superescalar



Processador com SMT



Multithreading

- Multithreading Simultâneos podem ser implementados em processadores com execução fora de ordem usando uma tabela de renomeação de registradores pro thread;
- Algumas vezes threads compartilham recursos

Eficiência do multithreading

- Os principais fatores que determinam a eficiência de um multithreading são:
  - Número de contextos suportados;
    - Mais contextos:
      - Reduzem chances do processador ficar em stall;
      - Introduzem maior custo;
      - Efetivos para aplicações com alto grau de paralelismo;
  - Overhead para realização da troca de contexto;
    - Em multithreading de granularidade grossa a troca de contexto deve ser feita em baixas quantidade de ciclos;
      - Menor que a quantidade de ciclos gastos nas operações cujo a latência deve ser ocultada;

Referências

HENNESSY, John L.;PATTERSON, David A. Computer architecture: a quantitative approach. Elsevier, 2011.

SILVA, Gabriel P. Arquitetura de computadores II: Multithreading. Universidade federal do Rio de Janeiro (Notas de Aula), [2008].