

Arquiteturas de processadores avançadas

MAC 412- Organização de Computadores

- Siang W. Song

Superescalar

- Uma CPU superescalar possui múltiplas unidades funcionais no processador (e.g. várias ALUs e várias unidades para operações ponto flutuante).
- Pode assim executar várias instruções no mesmo ciclo.
- A execução de uma instrução está baseada na disponibilidade de seus operandos.
- Instruções podem ser executadas fora de ordem. O algoritmo de Tomasulo, implementado em hardware, pode ser usado para não violar a dependência de dados.
- O paralelismo implementado por uma CPU superescalar é do tipo ILP (*Instruction Level Parallelism*) dentro de um mesmo processador.
- Exemplo: PowerPC 970 (4 ALUs, 2 unidades de ponto-flutuante e 2 SIMD, com capacidade de até 8 instruções por ciclo). Alpha 21264 (até 6 instruções por ciclo).

VLIW - *Very Long Instruction Word*

- Para que várias instruções possam ser executadas em paralelo, ou fora de ordem, há um custo de verificação de dependência entre as instruções.
- VLIW executa instruções em paralelo com base num escalonamento feito pelo compilador, sem necessitar de um hardware de escalonamento.
- Uma instrução VLIW contém múltiplas instruções. Por exemplo, se há 5 unidades de execução, há 5 campos numa instrução VLIW, especificando cada um qual operação deve ser feita em cada unidade de execução.
- VLIW implementa o paralelismo do tipo ILP (*Instruction Level Parallelism*).
- A diferença entre VLIW e superescalar é que VLIW usa software para determinar quais instruções podem ser executadas em paralelo, ao contrário da superescalar que usa hardware.
- Exemplos: Itanium IA-64 (até 6 instruções por ciclo).

- Dois ou mais *cores* ou núcleos integrados numa mesma pastilha de silício. Cada núcleo atua como um processador independente.
- Cada núcleo em geral tem sua memória cache. Pode também haver caches compartilhados. O uso de cache fica cada vez mais importante para atenuar o problema de gargalo de acesso à memória.
- O aumento do número de *cores* justifica o aparecimento de *network on chip*. As interconexões típicas entre os *cores* incluem anel, 2-d grade e *cross-bar*.
- Há muitos exemplos, e.g. Core 2 Duo, Core 2 Quad, Xeon dual-quad- e hexa-core, Nvidia GeForce 200 (10 *cores*), Tesla multi-core GPGPU (10 *cores*), UltraSparc T2 (8 *cores*), Tiler TILE64 (64 *cores*).

- No evento SBAC-PAD 2009, Yale Patt deu uma palestra intitulada “The 1000 core microprocessor: Will we be ready for it?”

<http://regulus.pcs.usp.br/sbac2009/KeynoteTalks/>

- Em 6 a 8 anos (Lei de Moore) podemos ter um chip com 50 bilhões de transistores, o que daria para fabricar um chip com 1.000 cores.
- Yale Patt questiona se devemos projetar um chip desses e, caso sim, se estaremos preparados para enfrentar as dificuldades:
 - desenvolvimento de algoritmo paralelo
 - largura de banda de memória (on-chip e off-chip)
 - considerações de potência

- Destaca o papel crucial da educação:
 - “*Think in parallel is natural !*”
 - Ensinar programação paralela (*parallel thinking*) aos alunos do primeiro ano (e.g. como calcular fatorial de n com dois *cores*).
 - Enfatizar a abordagem *bottom-up* e livrar-se da *top-down*.
 - Entender todas as camadas: algoritmo, linguagem, compilador, micro-arquitetura.