Lista 2 – Arquitetura e Organização de Computadores

Victor Rocha

1. Quais as vantagens de um processador multiciclo em relação a um uniciclo?

Devido ao fato do multiciclo poder ter tamanhos de clock diferentes para diferentes instruções, ganha-se no tempo. Um ciclo no multiciclo não é determinado pela instrução mais demorada, como no caso do uniciclo. Um load pode demorar 5 cliclos e um branch 3 por exemplo.

Outra vantagem é a reutilização de unidades funcionais, como a memória e a unidade lógica-aritmética por exemplo.

1. Quais as modificações necessárias em um processador multiciclo simples para que se introduza a função de pipeline?

Fazer a sobreposição de instruções, nesse caso também ajustando o clock para a instrução mais demorada;

1. Considerando o pipeline do MIPS (simples com **MEM compartilhada** para instrução e dados) e uma iteração de loop conforme o trecho de programa abaixo, relacione os conflitos que podem ocorrer e seus consequentes stalls. Qual o speedup (por iteração) para o programa em relação à versão sem pipeline?

Loop: subi $t2, $t2, 4

lw $t1, 0($t2)

add $t3, $t1, $t4

add $t4, $t3, $t3

sw $t4, 0($t2)

beq $t2, $0, loop

Conflito estrutural, no caso do store acessar a memória ao mesmo tempo que o load, no quarto ciclo.

Conflito de Dados, pois terão registradores que dependem do resultado de instruções prévias.

1. Na questão anterior, assuma que a memória de instruções e dados podem ser segmentadas e que o processador aplica a técnica de bypassing. Como ficarão os conflitos e seus stalls, e o speedup?

Com o bypassing, o resultado vai direto para a ULA, não precisando passar primeiro no banco de registradores, então resolveria a problema do conflito de dados.

O bypassing ainda não resolveria o conflito estrutural, teria um stall com o load e o store acessando a memória ao mesmo tempo.

O speedup seria de 2,72. No caso, 30 ciclos para o procedimento sem bypassing e 11 para com bypassing.

1. No programa abaixo, relacione as dependências (dados, WAR, WAW e outros) existentes.

div.d F1, F2, F3

sub.d F4, F5, F1

s.d F4, 4(F10)

add.d F5, F6, F7

div.d F4, F5, F6

(ordem de cima para baixo)

Dependência de dados: Div.d com sub.d; sub.d com s.d,

RAW: div.d com sub.d; add.d com div.d

WAW: add.d com div.d (f5 ainda não atualizado);

1. Apresente 5 processadores diferentes que trabalham paralelismo. Adicionalmente, descreva o funcionamento e o tipo de paralelismo de cada um dos processadores, bem como suas vantagens e desvantagens.

Whirlwind I foi um computador desenvolvido na guerra fria e trabalhava com *Bit-level paralelism,* esse tipo de paralelismo se baseia em aumentar o tamanho da palavra, porque ao fazer isso, reduz-se o número de instruções que o processador terá que executar em operações cujas variáveis têm um tamanho maior do que o cumprimento da palavra.

O Pentium trabalha com paralelismo em nível de instrução dinâmico, em que o processador decide quais instruções desempenhar em paralelo. Por exemplo, se temos 3 instruções em que só uma depende das anteriores, as que são independentes são executadas simultaneamente e depois a terceira. Reduzindo de 3 ciclos para 2. O problema dessa técnica se dá pela crescente disparidade entre a frequência de operação do processador e o tempo de acesso à memória. Embora seja possível tolerar essa latência na memória, o preço por dissipação de energia seria desproporcional.

O Itanium também apresenta paralelismo em nível de instrução, só que estático (que conta como outro tipo). Nesse o compilador é quem decide quais instruções desempenhar em paralelo.

GPUs, tipo Nvidia, apresentam paralelismo de dados. É a execução simultânea da mesma função em múltiplos núcleos nos elementos de um conjunto de dados.

Intel cores, família i. paralelismo de tarefas. Distribui diversas tarefas para serem desempenhadas simultaneamente e sobre os mesmos dados. Uma vantagem seria o forward scaling que adapta o número de tasks com o tamanho do problema a ser resolvido.

1. Em relação a memória cache. Um computador tem CPI 1 quando todos os acessos à memória acertam no cache. Loads e Stores totalizam 50% das instruções. Se a penalidade por miss é de 25 ciclos e o miss rate é 2%, qual o desempenho relativo se o computador acertar todos os acessos?

Memory stall cycles = Num\_misses x Penalidade

= IC x (1 +0.5) x 0.02 x 25

= IC x 0.75

Tempo de execução da CPU (cache) = (IC x 1.0 + IC x 0.75) x Ciclo de clock

= 1.75 x IC x ciclo de clock

Tempo de execução da CPU (cache) / tempo de execução da CPU = 1.75;

8) Descreva os seguintes conceitos:

A) Write through

É um método de armazenamento em que o dado é escrito na cache e na localidade correspondente da memória principal ao mesmo tempo.

b) Write back

É o método de armazenamento em que o dado é escrito na cache toda vez que uma mudança acontece, mas só é escrito na memória principal em certos intervalos.

c) Localidade Temporal

Está ligado ao intervalo de tempo em que o processador faz acesso à Palavra da Memória, ou seja, se um Bloco foi acessado recentemente, há grandes chances probabilidades de que ele seja novamente acessado em breve, durante a execução de um programa (loop).

d) Localidade Espacial

Tem a ver com o acesso de recursos próximos em um curto espaço de tempo, no caso, a possibilidade de acessar um programa ou recurso seguinte.