

Questão chamada subrotina (2,0p)

Para responder esta questão, considere que você é um desenvolvedor de um compilador para o processador MIPS estudado em sala de aula. Explique os passos necessários para se executar uma subrotina no MIPS, desde a chamada da subrotina até o retorno dela. Na explicação, detalhe quais registradores e instruções do MIPS dão suporte a chamadas de subrotinas.

São 6 passos

PASSO 1

Rotina que faz a chamada (caller) **coloca argumentos** em um **lugar onde a subrotina** chamada (callee) **pode acessá-los**

- **Passagem de argumentos**

No MIPS, 4 registradores são destinados para armazenar argumentos

\$a0 - \$a3 – números 4 a 7

PASSO 2

Caller transfere controle para o callee

MIPS oferece uma instrução para fazer a chamada a subrotina

Jump And Link

Instrução para chamar a subrotina possui um operando: **Label** da subrotina

`jal label`

Instrução **pula para endereço inicial da subrotina** e **salva endereço de retorno**

(instrução após chamada)

\$ra – return address (número 31) registrador que armazena endereço de retorno

Armazena PC + 4

PASSO 3

Callee **adquire os recursos de armazenamento** necessários

PASSO 4

Callee **executa** suas instruções

PASSO 5

Callee (quando é o caso) coloca o valor do resultado em um lugar que caller pode acessá-lo

Variáveis podem ser salvas em registradores disponíveis

No MIPS, 2 registradores para **valores retornados**

\$v0 - \$v1 – números 2 a 3

PASSO 6

Callee retorna controle ao caller

MIPS oferece uma instrução que pode ser utilizado para retornar da subrotina

Jump Register

Instrução para retornar da subrotina possui um operando: Registrador que contém um endereço

`jr registrador`

Instrução pula para endereço armazenado no registrador

No caso de retorno de subrotina, o registrador deve ser o \$ra

Questão conflito pipeline (2,0p)

Para resolver conflitos de controle em um pipeline, a CPU pode-se utilizar de execução especulativa estática ou dinâmica, e se as previsões derem erradas, a CPU dá um flush nas instruções executadas de forma errônea. Explique:

a) Em linhas gerais, como pode ser implementada a execução especulativa dinâmica.

Hardware mede comportamento dos desvios

Registra a **história recente de todos os desvios** em uma tabela

Assume que o **comportamento futuro** dos desvios continuará o **mesmo**

Quando errado, **anula instruções executadas erroneamente** (coloca os valores de controle para zero), **busca as instruções corretas** e **atualiza a história** do comportamento do desvio

b) Como é dado o flush na CPU? Detalhe quais são os registradores que são afetados pelo flush, considerando a CPU pipeline de cinco estágios vistos em sala de aula.

Zera os sinais de controle relativos às instruções subsequentes

Tem-se que fazer isto para instruções que estão nos **estágios anteriores a detecção** do branch

Zera os sinais de controle relativos às instruções executadas erroneamente, e faz-se isto para instruções que estão nos estágios IF, ID e EX (pois branch só é corretamente avaliado no estágio MEM).

Questão mapeamento cache (2,0p)

Considere que um computador possua uma cache de 512KB, que use endereços de 32 bits e que cada bloco tenha 4 palavras. Descreva o layout do endereço, indicando quais e quantos bits são utilizados para representar a tag, o índice (ou conjunto) e o offset, para as seguintes configurações de associatividade:

a) Associativa por conjunto grau 2 (2 - way set).

Offset:

Indica cada byte, de cada palavra do bloco: deve representar os 2^4 Bytes de cada bloco Logo, será necessário 4 bits de Offset.

Index (ou conjunto):

Como a cache possui 512 K B de dados, ou seja $2^9 \times 2^{10} \text{ B} = 2^{19} \text{ B}$

cada bloco tem 2^4 Bytes, logo tem $2^{19} \div 2^4$ Blocos (2^{15} Blocos)

Por ser de Grau 2, devemos 2^{15} dividir por 2, ou seja 2^{14}

Logo, Conjunto deverá ter 14 bits

a Tag será o que falta para 32, $32 - 4 - 14 = 14$ bits

Tag: 14 bits	Set: 14 bits	Offset: 4 bits
--------------	--------------	----------------

b) Associativa por conjunto grau 8 (8 - way set).

Offset:

Indica cada byte, de cada palavra do bloco: deve representar os 2^4 Bytes de cada bloco
Logo, será necessário 4 bits de Offset.

Index (ou conjunto):

Como a cache possui 512 K B de dados, ou seja $2^9 \times 2^{10} \text{ B} = 2^{19} \text{ B}$

cada bloco tem 2^4 Bytes, logo tem $2^{19} \div 2^4$ Blocos (2^{15} Blocos)

Por ser de Grau 8, devemos 2^{15} dividir por 8 (2^3), ou seja 2^{12}

Logo, Conjunto deverá ter 12 bits

a Tag será o que falta para 32, $32 - 4 - 12 = 16$ bits

Tag: 16 bits	Set: 12 bits	Offset: 4 bits
--------------	--------------	----------------

Observação: Não esqueça de justificar as respostas.

Questão Memória virtual (2,0p)

Paginação é um mecanismo de gerenciamento de memória virtual muito utilizado nos sistemas atuais. Tabela de páginas armazenadas em memória são utilizadas para auxiliar o processo de tradução de endereços virtuais em endereços reais. Um problema decorrente do armazenamento de tabelas de páginas na memória é que ela pode se **tornar grande demais (ter muitas entradas)**, ocupando muito espaço na memória. Cite uma solução para este problema e descreva como a partir de um endereço virtual podemos achar a página desejada utilizando esta solução (mostre de maneira geral o layout do endereço virtual na descrição).

Uma solução seria as tabelas multi níveis, que consiste em **paginar a tabela de páginas**.

Quebra-se espaço de endereçamento virtual em **múltiplas tabelas de página**.

Existe uma **tabela de tabela de páginas** (a de maior nível) que **contém os índices (localização) para as demais tabelas** de páginas.

Tabelas de páginas são carregadas na memória de acordo com a necessidade.

E aí o endereço vai ser

<número da tabela de página><número da página><offset>