LISTA – PROCESSADORES

Luan Henrique Medeiros do Nascimento

QUESTÃO 01

A produção de um processador começa de uma matéria prima retirado da areia: o silício. É a partir de um processo de derretimento da areia que se obtém o silício. O silício é um metal semicondutor. Por ser semicondutor, ele é perfeito para ser usado na construção de processadores, pois podemos ora transmitir energia e ora interromper a transmissão de energia. Depois do processo de retirada do silício da areia, esse material se torna um grande bloco. Depois, esse grande bloco de silício (uma grande salsicha de silício), é fatiado em discos, que são chamados de wafers. Esses wafers passam por cerca de 20 a 40 processos para ser dividido em partes bem pequenas chamadas de dies. Cada die será um chip de processador. Depois de dividir os wafers em dies, são realizados testes e retiradas as dies com falhas. Depois são feitas as junções dos dies em pacote/placas (por exemplo, para se ter processadores com mais de um núcleo). Logo após, são feitos mais testes, e, os que estiverem em bom estado, são finalizados e distribuídos para serem vendidos.

QUESTÃO 02

Para executar uma instrução, o processador realiza três ações: buscar, decodificação e execução. A busca é quando a CPU busca a instrução na memória RAM. Quando a instrução chega na CPU, ela é salva em um registrador chamado Registrador de Instrução (IR) e, então, é realizada a decodificação. A decodificação nada mais é do que entender (decodificar) o que a instrução quer fazer. Isso pode ser feito apenas lendo uma parte da instrução chamada código de operação (opcode) que indica a operação a ser realizada. Após a decodificação, é realizada a execução da instrução. Esse ciclo se repete indefinidamente até que o sistema seja desligado, ou ocorra algum tipo de erro, ou seja encontrada uma instrução de parada.

QUESTÃO 03

Os registradores armazenam informações vitais para o correto funcionamento dos programas em execução. É a memória mais rápido de um computador. São classificados em dois tipos: registradores visíveis ao usuário e registradores de controle de estado. Registradores visíveis ao usuário: são registradores que possibilitam que o programador de linguagem de máquina ou assembly minimize as referências à memória, pela otimização do uso de registradores. Ou seja, podem ser referenciados pelos recursos da linguagem de máquina que o processador executa. Os registradores visíveis ao usuário podem ser de quatro tipos: de uso (propósito) geral, de dados, de endereço e de códigos condicionais.

• De uso (propósito) geral: podem ser atribuídos para uma variedade de funções pelo programador, ou seja, podem salvar dados, endereços, etc. No entanto, frequentemente existem

restrições. Por exemplo, pode haver registradores dedicados para ponto flutuante e operações de pilha.

• De dados: podem ser usados apenas para guardar dados e não podem ser empregados para calcular o endereço de um operando.

• De endereços: podem ser, de certa forma, de uso geral ou podem ser dedicados para um modo de endereçamento em particular. Podem ser: ponteiros de segmento de memória, registradores de índice e ponteiros de pilha.

• De códigos condicionais: guarda códigos condicionais (também chamados de flags). Códigos condicionais são bits definidos pelo hardware do processador como resultado das operações. Por exemplo, uma operação aritmética pode produzir um resultado positivo, negativo, zero ou fora da capacidade. Registradores de controle de estado: são registradores usados pela unidade de controle para controlar a operação do processador e por programas privilegiados do Sistema Operacional para controlar a execução de programas. Existem quatro registradores de controle de estado que são muito usados: contador de programa (PC), registrador de instrução (IR), registrador de endereço de memória (MAR) e registrador de buffer de memória (MBR).

• Contador de programa: contém o endereço de uma instrução a ser lida.

• Registrador de instrução (IR): contém a instrução lida mais recentemente.

• Registrador de endereço de memória (MAR): contém o endereço de uma posição de memória.

• Registrador de buffer de memória (MBR): contém uma palavra de dados para ser escrita na memória ou a palavra lida mais recentemente.

QUESTÃO 04

Geralmente, a memória cache é dividida em nível, L1, L2 e L3. Veja na imagem uma possível arquitetura de memória cache, onde a L1 está mais próxima do núcleo do processador, a L2 é compartilhada por dois núcleos e a L3 compartilhada por todos os núcleos. Para o bom funcionamento da memória cache, é necessário ela seguir dois princípios: princípio da localidade espacial e princípio da localidade temporal.

• Princípio da localidade espacial: se um item é referenciado, provavelmente seus vizinhos também o sejam. Isso vai acontecer quando estamos usando funções, métodos, laços de repetições, contadores, variáveis etc.

• Princípio da localidade temporal: se um item é referenciado, provavelmente ele será referenciado novamente em um curto espaço de tempo, como por exemplo: vetores, matrizes, arrays etc.

QUESTÃO 05

Um processador CISC (Complex Instruction Set Computer, ou “computador com um conjunto complexo de instruções”), é capaz de executar várias centenas de instruções complexas diferentes, sendo extremamente versátil; Exemplos de processadores CISC são o 386 e o 486.

No começo da década de 80, a tendência era construir chips com conjuntos de instruções cada vez mais complexos. Alguns fabricantes, porém, resolveram seguir o caminho oposto, criando o padrão RISC (Reduced Instruction Set Computer, ou “computador com um conjunto reduzido de instruções”). Justamente por isso, os chips baseados nesta arquitetura são mais simples e muito mais baratos. Os processadores RISC são capazes de executar apenas algumas poucas instruções simples. Porém com uma velocidade mais alta que os processadores CISC. Um exemplo são os processadores Alpha, que em 97 já operavam a 600 MHz.

8 Características principais entre as duas arquiteturas:

A arquitetura CISC tinha as características:

• Muita interação com a memória;

• Requer múltiplos ciclos de clock para a execução ser completada;

• Pequena quantidade de registradores;

• Execução de uma instrução por vez.

A arquitetura RISC chegou trazendo as seguintes características:

• Execução de instrução por ciclo de clock;

• Instruções com formato fixo;

• Implementação por pipeline.

QUESTÃO 06

01001101.

QUESTÃO 10

V, V, V, F, V, F, F, F, F

RESTANTE

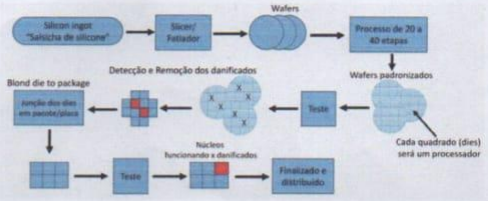
UEPB

UNIVERSIDADE ESTADUAL DA PARAÍBA – UEPB
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
DISCIPLINA: ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES
PROFESSOR: JOSÉ JANDILSON DE SOUSA ARRUDA

PROCESSADOR

QUESTÃO 01

Explique como funciona o processo de fabricação de um processador.



QUESTÃO 02

Quais e como funciona as etapas de ciclo de instrução.

QUESTÃO 03

Fale sobre os registradores, os tipos, classificação e os mais usados.

QUESTÃO 04

Explique como a memória cache funciona e quais os princípios de localidade que ela tem?

QUESTÃO 05

Diferencie arquitetura RISC e CISC e as principais diferenças.

QUESTÃO 06

Execute o programa mostrado na imagem abaixo.

PC = 218

REGISTRADORES	
MAR:	954
IR:	221
R1	218
R2	219
R3	220
R4	

MEMÓRIA RAM	
END.	CONTEÚDO
...	...
218	Ler 742
219	Ler 743
220	AND
221	Escreva em 954
...	...
742	1
743	0
...	...
954	0

QUESTÃO 07

Execute o programa mostrado na imagem abaixo.

PC	28
MAR	32
IR	33
R1	28
R2	29
R3	30

MEMÓRIA RAM	
END.	CONTEÚDO
...	...
28	00110110
29	01010011
30	Somar
31	Gravar em 32
32	10001001
...	...

QUESTÃO 08

Cinco instruções serão executadas usando pipeline de 5 estágios, como mostra na tabela abaixo. Sabendo que a instrução 3 depende da instrução 2 **para acessar a RAM**, preencha a tabela abaixo com as execuções.

Ciclos / Estágios	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°
Busca	I1	I2	I3	I4	I		I5	I6	I7	I8
Decodificação		I1	I2	I3			I4	I5	I6	I7
Acesso a RAM			I1	I2			I3	I4	I5	I6
Execução				I1	I2			I3	I4	I5
Salvar Resultado na RAM					I1	I_2			I3	I4

QUESTÃO 09

Observe o circuito lógico abaixo. Insira os bits "**0 1 0 0 1 1 0 1**" nas entradas **A, B, C, D,** respectivamente, e faça o circuito funcionar... Qual será o resultado?