

3.7 EXERCÍCIOS PROPOSTOS

1. Com relação à organização histórica do computador, responda:

a) Quais os componentes do computador propostos no modelo de Von Neumann?

Resposta:

Os componentes do computador propostos no modelo de Von Neumann incluem uma unidade lógica e aritmética, uma unidade de controle, uma memória que armazena dados e instruções, e mecanismos de entrada e saída. O detalhamento desses componentes pode ser visto a seguir:

- Unidade Aritmética e Lógica: É a unidade que executa as operações aritméticas e lógicas tais como: soma, subtração, multiplicação, divisão, raiz quadrada, movimentação entre a unidade aritmética e a memória, verificação do sinal do resultado, conversão de decimal para binário e vice-versa. Um total de 10 operações fundamentais foram definidas por Von Neumann no artigo original.
- Unidade de Controle: É a unidade responsável pelo sequenciamento das operações, transferência dos dados e instruções e pelo controle das demais unidades do computador.
- Memória: É a unidade onde as instruções, os dados de entrada, as tabelas de referência, e os resultados intermediários são armazenados para permitir a execução de um programa.
- Entrada: É a unidade que transfere a informação (numérica ou não) do meio externo. Todas as transferências devem ser feitas para a memória e nunca diretamente para a unidade de controle.
- Saída: É a unidade que transfere a informação (numérica ou não) para o meio externo. Todas as transferências devem ser feitas da memória para o meio externo, e nunca diretamente da unidade de controle.

b) Qual a sua principal contribuição em termos da organização de um computador? Resposta:

A principal contribuição de Von Neumann em termos da organização de um computador foi a proposta de computador controlado pelo programa armazenado em memória. Ou seja, uma proposta de arquitetura para um computador digital eletrônico que armazena seus programas na memória, permitindo assim que um mesmo *hardware* realize funções diferentes de acordo com as instruções e dados que são carregados na sua memória.

c) Qual a forma de representação interna dos números utilizada na proposta elaborada por Von Neumann?

Resposta:

Von Neumann, em uma analogia com o comportamento dos neurônios, sugere o uso da numeração binária para a representação interna dos números, ao invés da numeração decimal, pela evidente economia que isso proporciona no tempo gasto nos cálculos e na complexidade dos circuitos.

d) O que era e como era implementado o E-element? Qual a sua analogia com a biologia?

Resposta: O componente eletrônico básico da máquina EDVAC, com comportamento aná-

logo ao neurônio biológico, era o E-element, que poderia ser implementado com 1 ou 2 tubos de vácuo (válvulas) e combinado para criar circuitos mais complexos para a unidade aritmética, memória e unidade de controle.

- e) Qual o componente eletrônico principal proposto para a construção do computador?

Resposta: O componente eletrônico principal proposto por Von Neumann para a construção do computador foi o tubo de vácuo (válvula).

- f) Quais as características da arquitetura Harvard?

Resposta: A arquitetura Harvard é uma arquitetura de computador que usa conjuntos separados de barramentos de endereço e dados para ler e escrever dados na memória e para buscar instruções. O uso de memórias e barramentos separados permite que uma operação de busca de instrução e uma operação de leitura/escrita de dados ocorram ao mesmo tempo.

- g) Quais as diferenças do modelo de Von Neumann para o modelo de barramento de sistema?

Resposta:

Nesse modelo, a unidade de controle e a unidade aritmética, e também novos dispositivos de memória chamados de registradores, são agregados em um único elemento que recebe o nome de processador. A unidade de entrada e a unidade de saída são apresentadas agora também como um único elemento, chamado de unidade de entrada/saída. A memória continua sendo vista com uma unidade independente, com as mesmas funções da arquitetura de Von Neumann, ou seja, armazenamento de dados e instruções dos programas em execução. Outro elemento novo que surge neste modelo é o próprio barramento de sistema, que faz a interligação entre o processador, a memória e a unidade de entrada/saída.

- h) Como se subdivide o barramento de sistema?

Resposta:

O barramento do sistema é subdividido em barramento de endereço, barramento de dados e barramento de controle.

- i) Existe o barramento do sistema nos modernos computadores? Como são organizadas as suas placas?

Resposta:

Nos computadores pessoais modernos não há um barramento de sistema explícito, mas apenas um controlador (chipset em inglês) de E/S que faz a intermediação entre o processador e os dispositivos de entrada e saída. Na mesma pastilha do processador estão embutidos também o controlador e o barramento de acesso à memória principal; o controlador e/ou a interface de interconexão à placa de vídeo; e ainda um terceiro barramento que faz a interface com o controlador de E/S externo.

2. O processador é talvez o elemento mais importante do computador. Com relação a isso, responda:

- a) Quais são os principais componentes do processador?

Resposta:

Os principais componentes do processador incluem a unidade de controle, a unidade lógica e aritmética (ULA), registradores e, modernamente, a memória cache, além das inter-

faces com a memória e o controlador de vídeo.

b) Quais são as funções realizadas pelo processador?

Resposta: As funções realizadas pelo processador incluem:

- Buscar instruções e dados na memória;
- Programar a transferência de dados entre a memória e os dispositivos de entrada/saída;
- Decodificar as instruções;
- Executar as operações aritméticas e lógicas;
- Realizar o tratamento das exceções na execução das instruções;
- Responder aos sinais enviados por dispositivos de entrada/saída, tais como interrupções e sinais de erro.

c) Qual a função do relógio? Quais são os fatores limitantes da frequência máxima do processador?

Resposta:

A função do relógio é sincronizar todas as operações do processador. É um sinal elétrico periódico que controla o funcionamento de todos os componentes do processador. Os fatores limitantes da frequência máxima do processador incluem o tempo de propagação dos sinais através dos componentes do processador, e a potência máxima dissipada pelo processador. Quanto maior a frequência, maior o calor gerado pela operação do processador, o que pode levar ao colapso do seu funcionamento.

d) Quais são os passos do ciclo de instrução que o processador realiza?

Resposta:

Os passos do ciclo de instrução, que é executado pelo processador, incluem:

- i. Buscar a próxima instrução que está localizada na memória para o registrador de instrução;
- ii. Atualizar o apontador de instruções (PC) para o endereço da próxima instrução a ser executada;
- iii. Determinar o tipo de instrução e o número de operandos;
- iv. Buscar os operandos, se houver, para os registradores do processador;
- v. Executar as operações indicadas na instrução;
- vi. Armazenar os resultados;
- vii. Voltar ao passo 1 para executar a próxima instrução.

3. O desempenho de um processador está associado a diversas variáveis, entre elas o número médio de instruções executadas por ciclo e a frequência do relógio. Com relação a isso responda:

a) Qual a equação que define o tempo total de execução de um programa?

Resposta:

O tempo total de execução de um programa é definido pela equação:

Tempo de Execução = Número de Instruções * Ciclos por Instrução * Tempo de Ciclo do Relógio.

b) Qual a diferença de estratégia entre as arquiteturas RISC e CISC para diminuir o tempo de execução de um programa?

Resposta:

A principal diferença de estratégia entre as arquiteturas RISC e CISC para diminuir o tempo de execução de um programa é que a arquitetura RISC utiliza um conjunto reduzido de instruções simples e rápidas, enquanto a arquitetura CISC utiliza um conjunto mais complexo de instruções que podem realizar várias operações em uma única instrução. A arquitetura RISC tenta diminuir o tempo de execução de um programa aumentando o número de instruções executadas por ciclo e reduzindo o tempo de ciclo com uma arquitetura mais simples, enquanto a arquitetura CISC tenta diminuir o tempo de execução de um programa reduzindo o número total de instruções necessárias para executar um programa.

- c) Se um processador gasta 1.200.000 de ciclos para executar um total de 1.000.000 instruções, qual o IPC (instruções por ciclo) médio obtido?

Resposta:

Se um processador gasta 1.200.000 ciclos para executar um total de 1.000.000 instruções, então o IPC (instruções por ciclo) médio obtido é $1.000.000 / 1.200.000 = 0,83$.

- d) E o CPI (ciclos por instrução) médio?

Resposta:

O CPI (ciclos por instrução) médio é o inverso do IPC, então neste caso seria $1.200.000 / 1.000.000 = 1,2$.

- e) Qual o tempo total de execução deste programa em um processador com frequência de 1,2 GHz?

Resposta:

O tempo total de execução deste programa em um processador com frequência de 1,2 GHz seria:

Tempo = 1.200.000 ciclos * Tempo de Ciclo = 1.200.000 ciclos * (1 / 1,2 GHz) = 0,001 segundos ou 1 milissegundo.

4. As arquiteturas de processador podem se diferenciar bastante quanto à sua forma de construção, o que vai influenciar em diversas características finais desses processadores. Com relação a isso responda:

- a) O que define a largura em bits de uma arquitetura?

Resposta:

A largura em bits de uma arquitetura é definida pela largura em bits do maior operando inteiro que pode ser processado de uma única vez (em um ciclo de máquina) pela unidade aritmética e lógica.

- b) O que diferencia a chamada de procedimentos de um desvio comum no processador?

Resposta:

A principal diferença entre a chamada de procedimentos e um desvio comum no processador é que a chamada de procedimentos salva o endereço de retorno na pilha antes de desviar para o endereço do procedimento, enquanto um desvio comum simplesmente altera o apontador de instruções para o novo endereço sem salvar o endereço de retorno.

- c) Qual a principal vantagem da máquina de pilha? Onde encontramos exemplos de sua aplicação?

Resposta:

Na arquitetura de pilha, todos os operandos das instruções estão no topo da pilha, o que significa que os operandos são implicitamente definidos. Isso resulta em um número menor de bits necessários para codificar as instruções, em comparação com as arquiteturas com registradores, que exigem especificar a origem e o destino das operações no código das instruções. Atualmente, essa arquitetura é amplamente utilizada em máquinas virtuais que precisam enviar seu código pela Internet, como nas máquinas virtuais Java ou WebAssembly.

- d) Qual a principal característica das arquiteturas registrador-registrador? Onde encontramos exemplos de sua aplicação?

Resposta:

A principal característica das arquiteturas registrador-registrador é que todas as operações são realizadas entre registradores, sem acesso direto à memória durante a execução da instrução. Isso permite que as instruções sejam mais simples e rápidas, pois apenas as instruções de load e store fazem acesso à memória. Isso simplifica a decodificação das instruções e a lógica da unidade de controle, permitindo um hardware mais rápido e eficiente. Exemplos de sua aplicação incluem as arquiteturas RISC, como ARM e MIPS.

- e) As arquiteturas memória-memória são características de qual tipo de processador?

Resposta: As arquiteturas memória-memória são características dos processadores CISC, como o VAX780. Nesse tipo de arquitetura, as instruções podem acessar todos os seus operandos diretamente na memória. Apesar disso resultar em um código de programa mais compacto, a complexidade resultante torna o processador mais lento e ineficiente.

- f) Quais as diferenças e similaridades entre arquiteturas de pilha e acumulador?

Resposta:

As arquiteturas de pilha e acumulador são semelhantes no sentido em que ambas usam operandos implícitos na codificação das instruções. No entanto, enquanto a arquitetura de pilha os operandos estão no topo da pilha, a arquitetura do acumulador usa um único registrador, chamado acumulador, para esse propósito. Isso significa que na arquitetura do acumulador, todas as operações são realizadas com o acumulador e outro operando, enquanto na arquitetura da pilha, as operações são realizadas com os dois operandos no topo da pilha.

5. A unidade de controle é responsável pela coordenação do funcionamento do processador. Com relação a isso, responda:

- a) Quais as funções realizadas pela unidade de controle do processador?

Resposta:

A unidade de controle é responsável por gerar todos os sinais que controlam as operações dos componentes do processador e coordenar a movimentação de dados para garantir o correto funcionamento interno do processador. Além disso, ela realiza a busca das instruções na memória, transferindo-as para o registrador de instruções (RI) e, em seguida, decodifica essas instruções.

- b) O que é a palavra de controle da unidade de controle?

Resposta:

A palavra de controle da unidade de controle é um conjunto de bits que especifica quais

operações devem ser realizadas pelo processador em um determinado ciclo de instrução. Cada bit da palavra de controle corresponde a um sinal de controle específico que é enviado para os componentes do processador para realizar uma determinada operação. A operação desses circuitos é coordenada pelo relógio do processador que, junto com a palavra de controle, faz o acionamento de registradores, unidades funcionais, multiplexadores e outros circuitos para a correta execução das instruções.

- c) Qual a diferença entre a unidade de controle microprogramada e por *hardware*.

Resposta: A principal diferença entre a unidade de controle microprogramada e por *hardware* é a forma como os sinais de controle são gerados. Em uma unidade de controle microprogramada, os sinais de controle são gerados por um microprograma armazenado em uma memória especial chamada memória de controle. Em uma unidade de controle por *hardware*, os sinais de controle são gerados diretamente por circuitos lógicos combinacionais, normalmente uma PLA (matriz programável de ANDs e ORs para gerar funções lógicas), além de uma pequena máquina de estados.

- d) Quais as principais diferenças entre a microprogramação vertical e horizontal?

Resposta: A principal diferença entre a microprogramação vertical e horizontal é a forma como as palavras de controle são armazenadas na memória de controle. A microprogramação vertical tem um conjunto não muito extenso de microinstruções, tornando muito simples a definição de um microassembly para a especificação desses microprogramas. Cada microinstrução do conjunto realiza, em geral, poucas micro-operações, sendo, portanto, bastante simples. Portanto, se o conjunto total de instruções é igual a N , são necessários apenas $\lceil \log_2(N) \rceil$ bits para cada microinstrução. Como consequência, a memória de microprograma se torna mais estreita e, em geral, com maior comprimento, já que o número de microinstruções associado a execução de cada instrução do processador tende a crescer. Na microprogramação horizontal, cada palavra de controle contém todos os bits necessários para especificar todos os sinais de controle para uma determinada microinstrução. A lógica de controle é mais rápida, pois cada bit da microinstrução controla diretamente um ponto de controle específico do processador, mas o uso de um grande número de microinstruções diferentes pode dificultar a definição de um micro-assembly com funcionalidades diversas.

6. Um processador pode buscar os operandos de suas instruções de diversas maneiras. Com relação a isso responda:

- a) Enumere os principais modos de endereçamento das instruções do processador. Eles estão presentes em todos os processadores?

Resposta:

Os principais modos de endereçamento das instruções do processador incluem: endereçamento imediato, endereçamento direto ou absoluto, endereçamento por registrador ou acumulador, endereçamento indireto via memória, endereçamento indireto via registrador, endereçamento por deslocamento, endereçamento indexado e endereçamento de pilha. Nem todos os modos de endereçamento estão presentes em todos os processadores, pois isso depende da arquitetura do processador.

- b) Quais modos de endereçamento podem ter os operandos implícitos, ou seja, não estão

especificados claramente na instrução?

Resposta: Os modos de endereçamento que podem ter os operandos implícitos incluem o endereçamento de pilha e o endereçamento por acumulador. Nesses modos de endereçamento, o operando é armazenado no acumulador ou no topo da pilha e não precisa ser especificado na instrução.

c) Qual a utilização do modo indireto de endereçamento? Resposta:

O modo indireto de endereçamento é empregado quando o endereço do operando não é conhecido no momento da compilação do programa. Essa abordagem é especialmente útil para a manipulação de ponteiros especificados em linguagens de alto nível.

d) Qual a utilização do modo imediato de endereçamento?

Resposta:

O modo imediato de endereçamento é utilizado para representar valores ou constantes diretamente nas instruções do programa, em vez de fazer referência a um endereço de memória ou registrador. Nesse modo, o valor é codificado diretamente na instrução, sendo usado para carregar valores iniciais em variáveis ou comparação com valores fixos, como o valor final da variável de iteração em laços, definidos em linguagens de alto nível.

7. Os dados maiores que um byte podem ser armazenados com diferentes ordenações na memória do computador. Com relação a isso responda:

a) Quais são as ordenações possíveis desses dados? Detalhe e exemplifique.

Resposta:

Os dados maiores que um byte podem ser armazenados na memória do computador em duas ordenações diferentes: big-endian e little-endian. No formato big-endian, o byte mais significativo é armazenado primeiro (no endereço de memória mais baixo), enquanto no formato little-endian, o byte menos significativo é armazenado primeiro (no endereço de memória mais baixo). Por exemplo, considere o número hexadecimal 0x12345678. No formato big-endian, ele seria armazenado na memória como 12 34 56 78, enquanto no formato little-endian, ele seria armazenado como 78 56 34 12, considerando o endereço de memória mais alto aquele mais à esquerda.

b) Exemplifique pelo menos uma arquitetura para cada um dos modos de ordenação.

Resposta: Algumas arquiteturas que utilizam o formato big-endian incluem SPARC, Motorola 68000 e PowerPC. Algumas arquiteturas que utilizam o formato "little-endian" incluem x86, RISC-V e as primeiras versões da arquitetura ARM.

c) Existem arquiteturas que podem utilizar os 2 modos? Em caso positivo, exemplifique.

Resposta:

Sim, existem arquiteturas que podem utilizar ambos os modos de ordenação. Essas arquiteturas são chamadas de bi-endian e permitem que o modo de ordenação seja selecionado por meio de um bit de configuração. Um exemplo de arquitetura bi-endian é a MIPS, as versões mais recentes do ARM, e SPARC V9 de 64 bits (Sun/Oracle).

d) Computadores com ordenação diferente podem trocar dados diretamente? Como eles fazem para se comunicar, por exemplo, pela internet?

Resposta:

Computadores com ordenação diferente podem trocar dados desde que seja feita uma

conversão entre os formatos de ordenação para garantir que os dados sejam interpretados corretamente. Isso pode ser feito por meio de software ou hardware. Quando os computadores se comunicam pela internet, eles geralmente usam um formato padrão para troca de dados, como o formato big-endian usado pelo protocolo TCP/IP. Isso garante que os dados sejam interpretados corretamente independentemente da ordenação utilizada pelos computadores envolvidos na comunicação.

8. A arquitetura dois processadores se dividem em duas grandes categorias: CISC e RISC. Com relação a isso responda:

a) Apresente um resumo das principais características dos processadores CISC.

Resposta:

Os processadores CISC (Complex Instruction Set Computing) possuem um conjunto de instruções complexas, que podem realizar várias operações em uma única instrução. Esses processadores suportam diversos modos de endereçamento de operandos. Outra particularidade é que as instruções possuem formatos muito variáveis, de acordo com a quantidade de operações que deverão ser executadas, tornando o decodificador do processador mais complexo. Diferentes tipos de instruções podem referenciar operandos diretamente na memória principal. Por fim, cada fase do processamento das instruções pode ter duração variável, dependendo da complexidade da instrução em questão. Para executar uma única instrução, um processador CISC pode exigir vários ciclos de relógio. Como consequência, a implementação com uso de pipeline era difícil, e a taxa média de execução das instruções por ciclo tendia a ser superior a 1 ciclo por instrução (CPI). A unidade de controle era em geral microprogramada, e os compiladores podem gerar códigos compactos para essas arquiteturas.

b) Apresente um resumo das principais características dos processadores RISC.

Resposta: Os processadores RISC (Reduced Instruction Set Computing) apresentam características distintas em comparação às arquiteturas CISC. Elas são caracterizadas por instruções mais simples, usualmente com o mesmo tamanho, que demandam um número fixo de ciclos de máquina para sua execução. Além disso, utilizam poucos modos simples de endereçamento de operandos e possuem poucos formatos diferentes de instruções. Nas arquiteturas RISC, apenas as instruções de load (carregamento) e store (armazenamento) referenciam operandos na memória principal, e cada fase do processamento da instrução tem duração fixa, correspondente a um ciclo de máquina. Essas características têm diversas consequências. As arquiteturas RISC são implementadas com o uso do pipeline, permitindo a execução de várias instruções em paralelo e aumentando a eficiência do processamento. A taxa média de execução de instruções por ciclo de máquina é igual ou inferior a um ciclo por instrução (CPI), o que torna o processamento mais eficiente. Em geral, a unidade de controle das arquiteturas RISC é implementada em *hardware* (*hardwired*). O processo de compilação para essas arquiteturas é complexo e requer cuidados especiais para otimização do desempenho do código gerado, visando aproveitar ao máximo o potencial de execução rápida das instruções.

c) Enumere duas arquiteturas do tipo CISC.

Resposta: Duas arquiteturas do tipo CISC incluem a família x86 da Intel e a família VAX

da Digital Equipment Corporation.

- d) Enumere duas arquiteturas do tipo RISC.

Resposta:

Duas arquiteturas do tipo RISC incluem ARM e MIPS.

- e) Como são organizados atualmente os processadores Intel x86_64? Como arquiteturas RISC ou CISC?

Resposta: Atualmente, os processadores Intel x86_64 são organizados com uma abordagem híbrida. Externamente, eles seguem o padrão CISC (Complex Instruction Set Computing) para garantir a compatibilidade com o vasto conjunto de software legado desenvolvido para a arquitetura x86 ao longo dos anos. No entanto, internamente, os processadores x86_64 são projetados com uma arquitetura RISC (Reduced Instruction Set Computing) mais moderna. O funcionamento interno dos processadores Intel x86_64 envolve a conversão e decodificação das instruções x86 em micro-operações, chamadas de μ ops. Essas μ ops são operações mais básicas e simples, que o núcleo de execução do processador, que é essencialmente uma arquitetura RISC, pode otimizar para uma execução mais eficiente. Essa abordagem híbrida permite que os processadores x86_64 mantenham a compatibilidade com o software legado, enquanto aproveitam os benefícios de uma arquitetura interna mais otimizada. Dessa forma, os processadores podem alcançar um desempenho melhor e uma maior eficiência de execução, apesar da complexidade externa da arquitetura CISC.

9. O processador Sapiens é uma arquitetura hipotética, bem simples, utilizada para o ensino de arquitetura de computadores. Com relação a isso, responda:

- a) Qual o tamanho da arquitetura em bits?

Resposta:

A arquitetura do processador Sapiens é de 8 bits. Possui unidade lógica aritmética de 8 bits e acumulador, o único registrador do processador, também de 8 bits.

- b) Entre os diversos tipos, como você classifica a arquitetura do processador Sapiens? Justifique.

Resposta:

O processador Sapiens é uma arquitetura baseada em acumulador. Isso significa que ele possui um registrador especial, chamado acumulador, que é usado como operando implícito para a maioria das instruções.

- c) Você classificaria o processador Sapiens como um processador RISC ou CISC? Justifique.

Resposta:

O processador Sapiens pode ser classificado como um processador CISC (Complex Instruction Set Computing), pois apesar de possuir um conjunto bem simples de instruções complexas, pode acessar os operandos das instruções diretamente na memória e o tamanho das instruções é variável, dependendo da sua complexidade.

- d) Quais os modos de endereçamento utilizados pelo processador Sapiens? Exemplifique cada um dos casos.

Resposta:

O processador Sapiens utiliza cinco modos de endereçamento: acumulador, direto, imediato, indireto e pilha. O acumulador é um operando implícito em diversas instruções. No

modo direto, a palavra que segue o código da instrução contém o endereço de memória do operando. No modo imediato, a palavra que segue o código da instrução contém o próprio operando. No modo indireto, a palavra que segue o código da instrução contém o endereço de memória do ponteiro para o operando. A pilha também é um operando implícito em algumas instruções de manipulação de pilha e de chamada e retorno de procedimento.

- e) Relacione as instruções do Sapiens que não possuem nenhum modo de endereçamento.

Resposta:

As instruções que não utilizam nenhum dos modos de endereçamento apresentados são aquelas que não têm o acumulador e nenhum outro operando implícito ou explícito: NOP, HLT

- f) Qual o tamanho em bytes das suas instruções?

Resposta:

As instruções do processador Sapiens podem ter um, dois ou três bytes (8 bits).

- g) Como os modos de endereçamento são codificados em formato binário na instrução?

Resposta:

Os modos de endereçamento são codificados em formato binário nos 2 bits menos significativos do primeiro byte da instrução. 00 - modo direto 01 - modo indireto 10 - modo imediato 8 bits 11- modo imediato 16 bits.

- h) Quais são e para que servem os códigos de condição?

Resposta:

Os códigos de condição são bits armazenados no registrador de código de condição que indicam o resultado da última operação realizada pelo processador. O processador Sapiens possui três códigos de condição: negativo (N), zero (Z) e carry (C). Esses códigos são usados para controlar o fluxo de execução do programa através das instruções de desvio condicional.

- i) Relacione as instruções que alteram o código de condição C.

Resposta:

As instruções lógicas e aritméticas (ADD, ADC, SUB, SBC, NOT, AND, OR, XOR, SHL, SHR, SRA) e as instruções de transferência LDA, LDS e POP afetam apenas os códigos de condição N e Z de acordo com o resultado produzido. As instruções lógicas e aritméticas (ADD, ADC, SUB, SBC, SHL, SHR, SRA) afetam também o código de condição Carry de acordo com o resultado produzido.

- j) Relacione as instruções que NÃO alteram o código de condição C.

Resposta:

As demais instruções (STA, STS, JMP, JN, JP, JZ, JNZ, JC, JNC, JSR, RET, PUSH, IN, OUT, NOP, TRAP e HLT) não alteram os códigos de condição.

- k) Relacione as instruções que podem alterar o valor do apontador de instruções (PC) para um endereço diferente da instrução seguinte.

Resposta:

As instruções que podem alterar o valor do apontador de instruções (PC) para um endereço diferente da instrução seguinte são as instruções de desvio, as instruções de chamada e retorno de sub-rotina.

- l) Relacione as instruções que podem alterar o valor do apontador de pilha (SP).

Resposta:

As instruções que podem alterar o valor do apontador de pilha (SP) são as instruções PUSH e POP e as instruções CALL e RET. Adicionalmente, o valor do SP pode ser alterado pela instrução LDS.

- m) Quais são as instruções utilizadas nas operações de E/S?

Resposta:

As instruções utilizadas nas operações de E/S são as instruções IN e OUT, além da instrução especial TRAP.

- n) Quais são os dispositivos de E/S emulados pelo simulador SimuS?

Resposta:

Os dispositivos de E/S emulados pelo simulador SimuS incluem um painel de chaves, um display de sete segmentos, um teclado de 12 teclas, e um banner 1 x 6.

- o) Qual a finalidade da instrução de TRAP? Quais as funções implementadas no simulador Simus?

Resposta:

A finalidade da instrução TRAP é chamar o simulador para realizar operações mais elaboradas de E/S. As funções implementadas no simulador Simus incluem leitura e escrita em console, temporização, geração de números aleatórios, entre outras.

- p) Na sua arquitetura, qual a função dos registradores REM, RDM e RI?

Resposta:

Na arquitetura do processador Sapiens, os registradores REM (Registrador de Endereço de Memória), RDM (Registrador de Dados da Memória) e RI (Registrador de Instrução) são usados pela unidade de controle para armazenar temporariamente informações durante a execução das instruções. O REM armazena o endereço da memória onde os dados serão lidos ou escritos. O RDM armazena os dados lidos ou escritos na memória. O RI armazena a instrução atualmente em execução.

- q) Quais são as fontes possíveis para os endereços armazenados no REM?

Resposta:

As fontes possíveis para os endereços armazenados no REM incluem o apontador de instruções (PC), o apontador de pilha (SP), e o operando direto ou indireto das instruções, quando houver.

- r) Quais os destinos possíveis para os dados armazenados no RDM?

Resposta:

Os destinos possíveis para os dados armazenados no RDM incluem o acumulador (AC), o registrador de instruções (RI), a memória principal e os dispositivos de E/S.

- s) Relacione os registradores que possuem 8 ou 16 bits na sua arquitetura.

Resposta:

Na arquitetura do processador Sapiens, os registradores que possuem 8 bits são o acumulador (AC). Os registradores que possuem 16 bits são o contador de programa (PC) e o apontador de pilha (SP).

10. No trecho em linguagem de montagem a seguir, identifique os diversos modos de endereçamento (imediate, registrador/acumulador, direto, indireto) existentes nas instruções do processador Sapiens. Note que uma instrução pode ter MAIS de um modo de endereçamento.
- a) IN BANNER
 - b) JSR @ROTINA
 - c) LDA #100
 - d) SUB @PONTEIRO
 - e) SHL
 - f) TRAP @X
 - g) RET
 - h) ADC UM
 - i) AND @A
 - j) PUSH

Resposta:

Aqui estão os diversos modos de endereçamento existentes nas instruções do processador Sapiens no trecho de código fornecido:

- 'IN BANNER': Modo direto e acumulador.
- 'JSR @ROTINA': Modo indireto e pilha.
- 'LDA #100': Modo imediato e acumulador.
- 'SUB @PONTEIRO': Modo indireto e acumulador.
- 'SHL': acumulador.
- 'TRAP @X': Modo indireto.
- 'RET': Modo pilha.
- 'ADC UM': Modo direto e acumulador.
- 'AND @A': Modo indireto e acumulador.
- 'PUSH': Pilha e acumulador.

Note que uma instrução pode ter mais de um modo de endereçamento, dependendo do número de operandos que ela possui. Por exemplo, a instrução 'ADC UM' possui um operando e utiliza o modo direto para acessá-lo. Mas possui o acumulador como modo implícito de endereçamento.

11. Qual será o conteúdo do apontador de instruções (PC), apontador de pilha (SP), do acumulador e da variável B após a execução do seguinte trecho de programa? Justifique.

```
ORG 100
A: DW Y
Y: DB 20, 30, 40, 50
B: DS 1
ORG 0
INICIO:
    LDA #200
    LDS A
    POP
    POP
    STS A
    ADC @A
```

STA B
HLT

Resposta:

PC: 10H, ou 16 em decimal. É o endereço da instrução HLT. SP: 0068H ou 104 em decimal, pois foi carregado com o endereço armazenado em A (Y ou 102 em decimal) e incrementado com as duas instruções de POP. B: 46H ou 70 em decimal, que é a soma de Y[1] (30), colocado no acumulador com o segundo POP, e Y[2] (40), apontado pela variável A.

DRAFT