

Homework 3

Alunos:

Loredana Romano Devico – 11911BCC001

Lucas Marçal Coutinho – 11911BCC012

Lucas Mattos Vieira – 11911BCC015

Questões:

Sistema de Interconexão:

P1) Os barramentos podem ser caracterizados por diversos fatores tais como:

- Tipos de Barramentos;
- Técnicas de Comunicação em Barramento;
- Métodos para a Mudança do Controlador do Barramento;
- Convenções para Transferência de Informação;
- Características Elétricas.

Quais são os dois esquemas básicos para temporização da comunicação no barramento? E como estes *esquemas* operam?

Os dois tipos de esquemas básicos são o síncrono e assíncrono.

No esquema síncrono, a ocorrência dos eventos é determinada pelo clock. O barramento contém uma linha de clock, por meio do qual um transmissor transmite uma sequência de 0s e 1s. O processador emite um sinal de leitura e coloca um endereço de memória no barramento de endereços. Ele também emite um sinal de iniciar para marcar a presença do endereço e informação de controle no barramento. Um módulo de memória reconhece o endereço e depois do atraso de um ciclo (neste caso uma sequência de um 0 e um 1), coloca os dados e um sinal de confirmação no barramento.

No esquema assíncrono, a ocorrência de um evento no barramento depende de um evento ocorrido anteriormente.

O processador coloca sinais de endereço e de leitura no barramento. Depois de uma pausa para estabilização dos sinais, o processador emite um sinal MSYN (sincronismo mestre), indicando a presença de sinais válidos de endereço e de controle. A memória responde enviando os dados e um sinal SSYN (sincronismo escravo), que indica o envio de uma resposta. Depois do processador ler as linhas de dados, o processador retira os sinais MSYN do barramento. Isso faz com que a memória retire os dados e o sinal do barramento. Uma vez retirada a linha do sinal, o processador remove o sinal de leitura e a informação de endereço.

P2) Um barramento assíncrono que, por não depender de sinal de *clock*, pode conviver com dispositivos de velocidades distintas e diferentes tecnologias. Em contrapartida requer sinais adicionais para prover a devida sincronização dos eventos. A figura abaixo descreve a transferência iniciada pelo destino usando linhas de ***handshake***.

a) Descreva sequência de eventos na transferência.

Nesse método de controle, a unidade mestre usa uma linha de request para começar a transferência e a unidade de fontes usa uma linha de acknowledge para indicar que o dado está pronto.

1. A unidade mestre (receptora) confirma a solicitação para receber dados.
2. A unidade de fonte de dados coloca dados no bus e confirma ACK.
3. A unidade fonte recebe o dado e desativa o pedido.
4. Unidade receptora está pronta para próximo pedido.

b) Ilustre a troca de sinais envolvidos na transferência.

A unidade mestre envia um sinal requisitando dados (Master Ready), a fonte confirma com um sinal ACK que colocou no bus data, a fonte desativa o sinal e a unidade receptora sinaliza que está pronta para o próximo pedido (Slave Ready).

P3) Considere os dados a seguir.

- Barramento síncrono:

- Período do clock: 10 ns;
- Cada transmissão gasta 1 ciclo de clock;
- Tempo de acesso à memória: 100 ns.

- Barramento assíncrono:

- 15 ns para cada passo do protocolo handshake.

- Considerar dados com 32 bits em ambos casos (barramento com 32 bits)

a) Qual o tempo total para a leitura de uma palavra em cada um dos tipos de barramento?

- Barramento síncrono:

- Período do clock: 10 ns;
- Cada transmissão gasta 1 ciclo de clock;
- Tempo de acesso à memória: 100 ns.
- largura do barramento = 4 bytes (32 bits)
- envio do endereço para a memória: 10 ns
- leitura da memória: 100 ns
- envio do dado para o dispositivo: 10 ns

- Tempo total para a leitura de uma palavra:

$$10 \text{ ns} + 100 \text{ ns} + 10 \text{ ns} = 120 \text{ ns}$$

- Barramento assíncrono:
 - 15 ns para cada passo do protocolo handshake.
 - largura do barramento = 4 bytes (32 bits)
 - cada passo: 15 ns
 - passo da memória: 100 ns
- Tempo total para a leitura de uma palavra:
 - Passo 1: 15 ns
 - Passos 2,3,4: máximo (3 x 15 ns , 100 ns) = 100 ns
 - Passos 5,6,7: 3 x 15 ns = 45 ns
 - 15 ns + 100 ns + 45 ns = 160 ns

b) Qual a largura de banda (bandwidth) máxima?

- Barramento síncrono:

$$4 \text{ bytes} / 120 \text{ ns} = 33 \text{ MB/s}$$
- Barramento assíncrono:

$$4 \text{ bytes} / 160 \text{ ns} = 25 \text{ MB/s}$$

P4) Dado o projeto de um sistema embarcado, cujo processador acessa um dispositivo de entrada/saída por polling a uma taxa média de 1 acesso a cada 20 ms. Considerando uma comunicação assíncrona P82 (paridade par, 8 bits de dados e dois stop bits além do start bit), com 144.000 bps e uma UART com um registrador de buffer e um de deslocamento, sendo que em cada polling a CPU pode fazer uma ou duas escritas na UART, calcule:

UART é o acrônimo de Universal Asynchronous Receiver/Transmitter ou Receptor/Transmissor Universal Assíncrono. Sua finalidade é possibilitar a transmissão e a recepção de dados originalmente disponíveis na forma paralela, conforme mostra a figura a seguir.

Acesso (polling) = 1/20 ms

Comunicação = P82

STT – Start

BitP – Paridade

STP – Stop Bit

Taxa = 144.000 bps (bits por segundo)

UART = 1 buffer + 1 shift register

Quantidade = 100 Kbytes (KB)

a) Qual a taxa máxima de transmissão ideal (não deve ser considerado o tempo de polling do processador)?

Taxa máxima = 144.000 bps / 12 b = 12.000 Bps (bytes por segundo)

Tempo para transmitir na taxa máxima = 100KB / 12.000 Bps = 8,333 seg

b) Qual a taxa efetiva de transmissão em caracteres?

Taxa efetiva:

$1 / 12.000 = 83,33 \mu s$

20ms

Comunicação CPU dispositivo de E/S não fornecida.

c) Qual o tempo necessário para transmitir 100 Kbytes para cada caso acima?

Taxa efetiva = 2 B / 20 ms = 1 B / 10 ms = 100 B/seg (desconsiderando o tempo de comunicação da CPU com dispositivos de E/S)

Tempo = 100KB / 100 Bps = 1K seg = 1000 seg

d) Faça um esboço para ilustrar os instantes de tempo onde ocorre transmissão do dado, a comunicação CPU com dispositivo de E/S, e o tempo em que o processador está em atividades entre polling.

X	X	...	X	X
$1/12000 = 83.33 \mu s$		20ms		Não fornecido

P5) A figura abaixo representa a transferência de dados de um dispositivo de E/S (I/O device) para uma CPU por meio de uma interface. Descreva a transferência de dados, passo-a-passo, e descreva o significado dos sinais envolvidos.

I. A CPU verifica o status de E/S pelo registrador de status da interface;

II. O módulo de E/S retoma o status prévio e a CPU verifica se este está disponível para uso. (Caso seja utilizado Interrupt Driven, criar-se-á uma rotina de interrupção quando o dispositivo estiver disponível; caso use Programmed I/O, a checagem de disponibilidade será dentro de um loop).

III. Se estiver "ready", a CPU solicita para o módulo a transferência da informação, colocando os endereços específicos nos registradores e solicitando leitura.

IV. O módulo de E/S obtém os dados

V. O módulo de E/S em seguida transfere os dados para a CPU, armazenando no registrador definido pela CPU.

- No caso do DMA, a própria CPU adiciona o dado na memória.

P6) Suponha uma linha serial assíncrona com taxa de transmissão de 33,6 kbit/s. Quanto tempo leva para enviar um arquivo de 2MB? Suponha a serial configurada como 8 bits de dados, 1 stop bit e paridade ímpar.

- Taxa de transmissão de 33,6 kbit/s

1 startbit

8 bits de dados (1 byte)

1 bit de paridade ímpar

1 stopbit

$2 * 1024 \text{ bytes} = 2048 \text{ bytes (2MB)}$

Para cada byte se adiciona mais 3 bits

$3 * 2048 = 6144 \text{ bits} / 8 = 768 \text{ bytes}$

Somando isso a quantidade de informações

$2048 \text{ bytes} + 768 \text{ bytes} = 2816 \text{ bytes}$

Convertendo a taxa de transferência para Kbytes

$33,6 / 8 = 4.2 \text{ Kbytes/s}$

$\text{Tempo} = 2816 \text{ bytes} / 4.2 = 670,48 \text{ seg}$

Sistema de Entrada e Saída:

P7) [Floyd 13.30] Explique a diferença entre portas de E/S dedicadas e mapeadas em memória. Qual delas não pode utilizar instruções de IN e OUT? Por quê?

- A Memória mapeada de E/S é mapeada para o mesmo espaço de endereçamento como memória de programa ou usuário, sendo acessada da mesma forma. As portas dedicadas usam um endereço separado e dedicado a isso, sendo acessado por um determinado conjunto de instruções do microprocessador. Assim, vê-se que o método das portas dedicadas é a que usa IN e OUT, porque o método de mapeamento de memória não utiliza a classe de instruções do processador que permite a escrita de instruções de IN e OUT, feita no espaço de endereçamento separado da memória principal.

P8) Brevemente descreva a diferença entre isolated I/O e memory mapped I/O. Qual é o modo utilizado no MIPS? Explique. Quais são as vantagens do memory mapped I/O?

- O método de mapeamento de E/S em memória utiliza via de endereços para endereçar a memória e os dispositivos de E/S - as instruções da CPU usadas para acessar a memória também são usadas para acessar os dispositivos. Para acomodar os dispositivos de E/S, áreas do espaço endereçável da CPU devem ser reservadas para estes, podendo ser temporária ou permanente. Cada dispositivo de E/S monitora a via de endereçamento da CPU e responde a qualquer acesso da CPU ao espaço de endereço destinado ao dispositivo, conectando a via de dados ao registrador do hardware do dispositivo desejado.

As vantagens desse método são a simplicidade e o uso de comandos sem a necessidade de linhas de controle. Além disso, é suportado por todas as instruções e modos de endereçamento, tornando a programação algo mais simples.

O outro método de E/S utiliza uma classe especial de instruções do processador, especificamente as instruções IN e OUT, que escrevem em um dispositivo de E/S. Estes dispositivos possuem um espaço de endereçamento separado da memória geral, implementado por um pino extra de E/S na interface da CPU ou por um barramento inteiro dedicado a entrada/saída. Por esta área de memória, dedicada a E/S, ser isolada da memória principal ela é chamada de E/S isolada.

O MIPS é um exemplo de E/S isolado, pois possui uma coleção de instruções especiais nas operações de E/S.

P9) A arquitetura de E/S (entrada/saída) de um computador constitui sua interface com o mundo exterior. Ela é projetada para permitir um controle sistemático de interação com o mundo exterior e fornecer ao sistema operacional as informações que ele necessita para gerenciar as atividades de E/S de maneira efetiva. Estudamos três técnicas para a realização de operações de E/S que são: programmed I/O; interrupt-driven e DMA (direct memory access). Descreva cada um destes métodos, apresentando as vantagens e desvantagens de cada um.

1. Programmed I/O:

Neste modo, as instruções escritas em um programa de computador iniciam a transferência. Uma instrução de entrada é necessária para armazenar os dados do dispositivo na CPU e outra, de armazenamento, é necessária para transferir os dados para o dispositivo. Essa transferência requer monitoramento constante do dispositivo periférico pela CPU, que também verifica constantemente a possibilidade de nova transferência subsequente. Portanto, a CPU permanece em um loop até que o dispositivo de E / S indique que está pronto para a nova transferência. Vê-se que esse modo é um processo demorado, ocupando desnecessariamente o processador e leva ao desperdício dos ciclos da CPU.

2. Interrupt-driven:

Interrupções e comandos especiais são os responsáveis por informar a interface para emitir o comando de interrupção quando os dados se tornam disponíveis e a interface está pronta para a transferência. Evitando desperdício, enquanto isso, a CPU continua executando outras tarefas e não precisa verificar o sinalizador. Quando o flag é acionado, a interface é informada e uma interrupção é iniciada, fazendo com que a CPU se desvie do que está fazendo para responder à transferência de E/S e então se ramifica para o serviço que processa a solicitação de E/S. Após a conclusão da transferência, a CPU retorna à tarefa anterior que estava executando.

3. Acesso direto à memória (DMA):

A transferência de dados entre uma mídia de armazenamento rápido e a unidade de memória é limitada pela velocidade da CPU. Pensando nisso, foi desenvolvido o que permite que os periféricos se comuniquem diretamente uns com os outros usando os barramentos de memória, ou seja, sem a intervenção da CPU. Esse tipo de técnica de transferência de dados é conhecido como DMA, no qual o controlador DMA assume os barramentos e gerencia a transferência diretamente entre os dispositivos e a unidade de memória.

P10) Brevemente descreva a diferença entre blocking I/O e non-blocking I/O. Quais são os cenários de uso de cada uma em programas que realizam operações de E/S?

Non-blocking I/O é a capacidade de fazer operações de entrada e saída sem bloquear a execução de aplicações em paralelo. Comandos de E/S podem demorar devido ao controle do hardware, então algumas técnicas podem ser aplicadas para paralelizar a execução. A mais popular atualmente é a assincronicidade. Aplicações bloqueantes são especialmente ruins quando está fazendo interação com o usuário, sendo então uma técnica vantajosa para a experiência do usuário.

Com o blocking I/O, quando o cliente faz uma solicitação de conexão ao servidor, o soquete que processa essa conexão e o thread correspondente que lê a partir dele é bloqueado até que alguns dados lidos apareçam. Esses dados são colocados no buffer da rede até que estejam todos lidos e prontos para processamento. Até que a operação seja concluída, o servidor não pode fazer mais nada além de esperar.

P11) Descreva o método Daisy-chaining para gerenciamento de interrupções?

O método daisy-chaining envolve a conexão de todos os dispositivos que podem solicitar uma interrupção de maneira serial. Esta configuração é influenciada pela prioridade dos dispositivos. O dispositivo com a prioridade mais alta é colocado na primeira posição, seguido pelos dispositivos de prioridade mais baixa até o dispositivo com a prioridade mais baixa, que é colocado por último na cadeia. Se algum dispositivo tiver seu sinal de interrupção no estado de baixo nível, a linha de interrupção vai para o estado de baixo nível e habilita a entrada de interrupção na UCP. Quando nenhuma interrupção está pendente, a linha de interrupção permanece no estado de alto nível e nenhuma interrupção é reconhecida pela CPU.

P12) Um sistema de arbitragem tipo daisy-chain contém dispositivos que levam 10ns para propagar o sinal de arbitragem. Considerando que o período de clock é de 155ns, quantos dispositivos podem ser colocados no daisy-chain? Explique por quê.

Nesse caso 15 dispositivos podem ser colocados no sistema de arbitragem daisy-chain. Isso se dá porque o tempo de propagação do sinal do primeiro dispositivo ao do último dispositivo tem que ser menor que 1 período de clock. E, nesse caso,

$15 \text{ (dispositivos)} * 10 \text{ ns (tempo de propagação do dispositivo)} = 150 \text{ ns} < 155 \text{ ns (tempo de clock)}$.

P13) Em quase todos os sistemas que incluem módulos de DMA (direct memory access), o acesso do módulo de DMA à memória principal tem prioridade mais alta do que o acesso da CPU. Por quê? Dica: Pense na natureza das operações de acesso de memória do processador e dos dispositivos de E/S.

Os computadores que têm os canais de acesso direto à memória podem transferir dados aos dispositivos com muito menos perdas gerais de processamento do que computadores sem uma via de acesso direto à memória. Similarmente um elemento de processamento dentro de um processador multi-core pode transferir dados para e de sua memória local sem ocupar seu tempo de processamento e permitir a simultaneidade de transferência de dados. Sem acesso direto da memória, usando a modalidade programada de entrada/saída (E/S) para uma comunicação com os dispositivos periféricos, ou as instruções da troca no caso dos núcleos multi-core, o processador central é ocupado inteiramente para a leitura ou escrita da operação, e assim não se torna possível executar o outro afazer. Com acesso direto da memória, o processador central executa transferências, faz outras operações enquanto alguma transferência estiver em andamento, recebe uma interrupção do controlador de acesso direto da memória uma vez que a operação foi feita.

P14) [Tanenbaum 3.37] Calcule a largura de banda necessária em um barramento para mostrar um filme VGA (640×480), true-color (24 bits) a uma taxa de 30 quadros/s. Considere que os dados devem passar pelo barramento duas vezes: uma para ir do drive de DVD para a memória e outra para ir da memória para a placa de vídeo.

Do enunciado, retiramos os seguintes dados:

$$307200 \text{ pixels (por quadro)} \leftarrow 640 * 480$$

$$7.3728 \text{ Mbits} \leftarrow 307200 * 24$$

$$0.9216 \text{ MB (por quadro)} \leftarrow 7.3728 / 8$$

$$27.648 \text{ MB/s (30 quadros por segundo)} \leftarrow 0.9216 * 30$$

Porém, os dados passarão duas vezes pelo barramento. Assim, devemos transportar:

$$27.648 * 2 = 55.296 \text{ MB/s}$$

R: A largura necessária no barramento é de pelo menos 56MB.

P15) Durante um evento de interrupção CPU, primeiramente, recebe um sinal $IRQ=1$. Descreva os passos subsequentes que a CPU realiza após o recebimento desse sinal, assumindo que as interrupções estão habilitadas. Para elaborar sua resposta, assuma a existência de um par de linhas separadas de $IRQ/IACK$ para cada dispositivo de I/O.

- 1 – CPU recebe o pedido de interrupção (IRQ);
- 2 – CPU termina a execução da instrução corrente;
- 3 – CPU reconhece sinal de interrupção (IACK);
- 4 – CPU coloca PSW e PC na pilha de controle;
- 5 – CPU carrega novo valor do PC, baseado na interrupção;
- 6 – Salva informações remanescentes sobre o estado do processo;
- 7 – Processa a interrupção;
- 8 – Restaura a informação do estado do processo;
- 9 – Restaura os PSW e PC anteriores.

P16) Suponha que uma unidade de disco tenha as seguintes características:

- 4 superfícies.
- 1024 trilhas por superfície.
- 128 setores por trilha.
- 512 bytes / setor.
- Tempo de busca (tempo de busca) trilha-a-trilha de 5 milissegundos.
- Velocidade de rotação de 5000 RPM.

a) Qual é a capacidade da unidade?

$$4 * 1024 * 128 * 512 = 268435456 \text{ Bytes}$$

$$268435456 : (2^{20}) = 256\text{MB}$$

b) Qual é o tempo de acesso?

$$5000 \text{ RPM} = 83.333 \text{ RPS}$$

$$t_A = t_s + 1/(2r) + b/(rN)$$

$$t_A = 0.005 + 1/(2*83.333) + 512/(83.333*512*128)$$

$$t_A = 0.005 + 1/166.666 + 512/5461311.488$$

$$t_A = 0.005 + 0.006 + 0.0001 = 0.0111 \text{ seg} = 11.1 \text{ ms}$$