Capítulo 3

Sistema de Memória

Como vimos no capítulo 1 um processador está constantemente buscando e executando instruções armazenadas na Unidade de Memória, também conhecida como **memória principal**, cujo tempo de acesso é compatível com o da UCP. Por razões econômicas é comum adicionar nos (micro)computadores **memória secundária**, constituída por dispositivos de armazenamento de grande capacidade e de baixo custo, como disquetes, fitas e discos rígidos, para guardar informações de pouco uso. O objeto de estudo deste capítulo é a Unidade de Memória.

Uma Unidade de Memória contém um conjunto de células organizadas em um arranjo (forma matricial), onde cada linha é endereçável por um número inteiro. O espaço de endereços em cada dispositivo não é necessariamente igual ao da UCP; portanto, um circuito adicional é necessário para mapear os espaços (locais) de cada dispositivo no espaço endereçável (global) do processador, de 0 até 2^m sendo m a quantidade de linhas de endereços.

Os dispositivos mais utilizados para implementar uma memória principal são as pastilhas semicondutoras. Estas pastilhas podem ainda diferir na tecnologia, demandando distintos circuitos adicionais para garantir um correto armazenamento. Algumas delas são não-voláteis (conseguem manter a informação mesmo sem alimentação), outras voláteis (perdem o seu conteúdo quando não são devidamente alimentadas) e outras "pseudo-não-voláteis" (conseguem manter o seu conteúdo com baixo consumo de energia). Mesmo energizadas, ainda se distinguem memórias que necessitam (as dinâmicas) e não necessitam de regeneração periódica (as estáticas). A adequabilidade dessas memórias em um sistema computacional depende do balanço entre custo, desempenho, consumo, tamanho e confiabilidade.

Estes fatores podem tornar o projeto de um sistema de memória complexo, requerendo várias decisões como:

- diante da diversidade na capacidades de pastilhas semicondutoras de memória, quais critérios devem ser utilizados para decidirmos por um determinado tipo de memória?
- como se pode organizar um conjunto de pastilhas de memória em blocos de memória compatíveis com o espaço de endereços do processador?
- como se projeta um decodificador de endereços para mapear o endereço (global) gerado pela UCP em um endereço (local) dentro do dispositivo endereçado?
- quais dispositivos digitais podem ser utilizados para implementar um decodificador de endereços? quais critérios devem ser utilizados na sua escolha?
- quais tipos de memória são disponíveis no mercado? quais são as peculiaridades de cada um?
- quais cuidados devemos ter no projeto do sistema ao optarmos por um determinado tipo de memória?
- quando se deve utilizar memória *cache*?

3.1 Terminologia

Capacidade: a quantidade de bits que uma memória consegue armazenar. Normalmente, utiliza-se a notação $A \times p$ para designar a capacidade de uma memória, onde A é a linha de palavras e p, tamanho de cada palavra.

Densidade: relação entre os bits armazenados por área da pastilha.

Célula de mamória: é o dispositivo ou um circuito elétrico que armazena um bit (0 ou 1).

3.2 Projeto de Decodificadores de Endereços

O projeto de um decodificador pode ser dividido em quatro fases:

1. construir o mapa do espaço de endereços do processador;
2. decidir como as pastilhas devem ser organizadas em blocos;
3. definir a tabela de decodificação de endereços; e
4. escolher dispositivos digitais para implementar a tabela de decodificação.

Uma pastilha de capacidade $A \times p$ pode ter diferentes organizações ao variar o número de linhas A e o número de bits em cada linha p. É função do projetista organizá-las em blocos de memória com tamanho de palavra compatível com o do processador. Os **mapas se E/S** e **mapas de memória** são úteis para visualizar graficamente a ocupação do espaço de endereços de uma processador pelos dispositivos da Unidade de Memória e pelas interfaces E/S, respectivamente.

No caso de MC68000, embora ele tenha uma arquitetura de 32 bits e suporte instruções para três tamanhos de dados (*byte*, *word* e *long word*), os projetistas podem considerá-lo uma máquina de 16 bits e devem organizar as pastilhas de memória em blocos de memória de palavras de 16 bits. Figs. 5.30 (p. 346) e 5.31 (p. 347) do livro-texto ilustram a organização de 16 pastilhas de 4M × 1 bit e 16 pastilhas de 512K × 8 bits em um boloco de memória de 4M × 16 bits. A primeira dúvida que aparece é qual das duas organizações é preferível. Isso vai depender das restrições do projeto (custo, desempenho, manutenção, disponibilidade, etc). Grosso modo, a 1^a configuração é superior sobre a 2^a, pois

1. a 2^a configuração requer um circuito adicional de decodificação para selecionar o par de pastilhas que contém o endereço desejado;
2. cada pastilha da 1^a configuração possui 22 pinos de endereços e 1 pino de dados; enquanto na 2^a configuração, cada uma tem 19 pinos de endereços e 8 pinos de dados. A segunda pastilha é, portanto, fisicamente maior (maior número de pinos) e tem uma área de ocupação maior; e
3. a 2^a configuração é mais propícia a ruído, pois cada linha do barramento é conectada em paralelo a 8 distintos pinos, implicando uma carga 8 vezes maior do que a carga na 1^a configuração.

Após a organização das pastilhas de memória em blocos de palavras de tamanho compatível com o do processador, é necessário definir uma correspondência entre os endereços (locais) em cada bloco e os endereços (globais) gerenciados pelo processador. O princípio de correspondência utilizado se baseia essencialmente em dois fatos (Fig. 5.3 do livro-texto, p. 309):

- as linhas de palavra em uma pastilha de memória são referenciadas sequencialmente por valores inteiros, de 0 até $A - 1$; e
- os dados são colocados(no ciclo de leitura)/capturados(no ciclo de escrita) nos seus pinos em função do sinal de gatilho CS* (*chip select*).

Para acessar uma palavra localizada em uma pastilha específica, o processador precisa somente ativar o sinal CS* da pastilha e colocar no barramento de endereços o endereço (local) da palavra. Resta agora saber como se pode derivar estas duas informações a partir de um endereço (global) da UCP. A solução é bem intuitiva – dividir os bits que representam um endereço em dois campos disjuntos, um campo de bits (menos/mais significativos) para identificar o dispositivo endereçado, ou derivar os sinais CS*, e um campo de bits (mais/menos significativos) para referenciar os endereços (locais) deste dispositivo.

Dependendo da utilização dos bits de endereço, distinguem-se três estratégias de decodificação:

Endereço Completo: (*full address*) todos os bits do endereço são utilizados para endereçar uma palavra da memória (Figs. 5.4, 5.5 e Tab. 5.1 do livro-texto, pp. 311–312). Normalmente o circuito de decodificação é mais complexo, envolvendo muitas entradas (bits de endereços) e poucas saídas (sinais CS*).

Endereço Parcial: (*partial address*) nem todos os bits são utilizados para endereçar uma palavra da memória. Usualmente, a quantidade de bits mais significativos reservados para derivar sem ambiguidade os sinais de CS* é $log_2 M$, onde M é o número de blocos de memória (Fig. 5.8 e Tab. 5.3 do livro-texto, p. 315). Ela é a mais simples e barata, mas subutiliza o espaço de memória provido pelo microprocessador, o que dificulta uma expansão posterior.

Endereço por Bloco: (*block address*) os bits que derivam os sinais CS* são organizados em níveis. É um compromisso entre as duas estratégias anteriores: uma lógica mais simples porém com possibilidades de expansão (Fig. 5.14 e Tab. 5.4 do livro-texto, p. 322).

Note que como a arquitetura do MC68000 é de E/S mapeada em (o espaço de endereços de) memória, os periféricos também devem ser considerados no projeto de decodificadores.

Uma vez estabelecida a correspondência entre os espaços de endereços, representada em tabelas de decodificação, deve-se projetar um decodificador dos sinais CS* a partir dos bits de endereços. A tecnologia utilizada pode variar de simples portas lógicas até dispositivos com lógica programável.

Exercícios de Revisão

1. Resolver os exercícios 1, 2, 3, 4, 5, 6, 11, 14, 17, 18, 19, 20 e 27 do capítulo 5 do livro-texto.

Preparo para Próxima Aula

1. Leia atentamente a seção 5.2 do livro-texto (pp. 316 – 343) e descreva as funções lógicas suportadas pelos seguintes dispositivos digitais
   - portas lógicas;
   - decodificadores m–linhas–para–n–linhas;
   - PROMs (ROMs programáveis);
   - PLDs (dispositivos programáveis): PLA/FPLA (Arranjo lógico programável), PGA/FPGA (Arranjo de portas programável) e PAL (Lógica de arranjos programável).
2. Explique ainda como cada um pode ser utilizado na impelmentação das funções descritas em uma tabela de decodificação.

De modo geral, espera-se que um decodificador tenha as seguintes características:

- ser rápido: ele não deve introduzir atrasos que comprometam a compatibilidade temporal entre a UCP e a Unidade de Memória.

- minimizar o número de pastilhas: para ocupar menos espaço e ser menos susceptível a erros.
- ser flexível(expansível): ser fácilmente modificável para mapear novos componentes ao espaço de endereços do microprocessador.
- ser economicamente viável.

Dos dispositivos digitais mais utilizados para implementação de um decodificador de endereços é da função dos projetistas encontrar um ponto de equilíbrio entre as propriedades positivas e negativas de cada um deles:

- custo baixo das portas lógicas contra o custo relativamente alto dos dispositivos programáveis;
- flexibilidade (funcional) dos dispositivos programáveis (e, com certa restrição, dos PROMs) contra a rigidez das portas lógicas e dos decodificadores;
- alta área de ocupação das portas lógicas (vários CIs) contra a baixa área demandada pelos outros dispositivos;
- baixa velocidade de chaveamento dos decodificadores contra alta velocidade de chaveamento de algumas portas lógicas e dispositivos programáveis;
- flexibilidade (física) das portas lógicas contra a rigidez dos PROMs e dispositivos programáveis, nos quais o número de pinos de entrada e de saída é pré-definido.

A prática mais comum é a utilização combinada destes dispositivos para tirar o melhor proveito de cada um, como ilustram vários exemplos no livro-texto.

3.3 Memória Estática

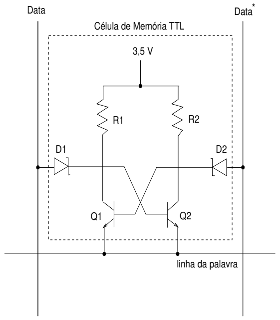
Uma memória estática se caracteriza por não necessitar de regenerações periódicas para manter o seu conteúdo enquanto o sistema estiver devidamente energizado.

Quanto à volatilidade do conteúdo da memória, podemos classificar as memórias estáticas em [S]ROMs (*read only memory*) e [S]RAMs (*random access memory*). O conteúdo das ROMs não é perdido se as desconectarmos da fonte de alimentação. O mesmo não ocorre com as RAMs.

3.3.1 SRAMs

Memórias estáticas RAMs são utilizadas em aplicações onde os programas e os dados são alterados frequentemente. Internamente, elas são constituídas por um arranjo de células interligadas pelas linhas de palavra e pelas linhas de bit. As linhas de palavra controlam o acesso simultâneo dos bits que compõem cada palavra da memória; enquanto as linhas de bits controlam o acesso ao conteúdo de cada célula. Quanto à tecnologia utilizada, distinguem-se duas classes de RAMs:

células bipolares (TTL): uma configuração simples consiste de dois transistores, dois (Schottky) diodos e duas resistências.



Assumindo que o conteúdo da célula seja 1 quando Q_1 conduz e Q_2 não conduz; e seja 0 quando Q_1 não conduz e Q_2 conduz,

- no ciclo de escrita: o nível de tensão na linha da palavra passa de 2.5V (estado inativo) para 0.3V ao ser selecionada. Para escrever 0 na célula, basta aplicar uma tensão de ordem 3V na linha DATA, fazendo com que Q_2 conduza (e consequentemente, Q_1 deixe de conduzir). E para armazenar 1 na célula, a tensão deve ser aplicada na linha DATA*.

- no ciclo de leitura: com o nível de tensão na linha da palavra em torno de 0.3V, o diodo do lado do transistor que conduz passa a ser polarizado no sentido direto e fluirá a corrente por ele e pela linha (DATA ou DATA*) na qual ele está ligado. Quando uma corrente é detectada na linha DATA, assume-se que o conteúdo da célula seja 0; senão, 1.

células MOS: uma configuração simples é constituída por 6 transistores (Fig. 5.28 do livro-texto, p. 345). De forma análoga às células bipolares, a condução num dos transistores T1 e T2 pode representar o conteúdo 1 ou 0. As memórias MOS, além de apresentarem uma densidade maior e um processo de fabricação mais simples em relação às bipolares, tem alta imunidade ao ruído e baixo consumo de potência. Um aspecto negativo destas memórias seria a sua velocidade de operação – elas são mais lentas em comparação com as bipolares.

Aspecto Funcional

Como já vimos no capítulo 2, além dos sinais de endereçamento e de dados, RAMs são providas de dois tipos de sinais de controle básicos (Figs. 4.17, 4.24, Tabs. 4.5, 4.8 nas p. 228, 229 e 238):

- sinais de seleção de pastilha: CS e
- sinal de tipo de acesso: R/W*.

Considerando que os sinais de endereços sejam válidos, os sinais de dados são capturados em relação ao sinal de gatilho, CS. O sentido do fluxo de dados (entrada ou saída) é determinado pelo sinal R/W*.

Algumas pastilhas dispõem ainda de um sinal de controle adicional, conhecido como habilitação de saída (OE*). Este sinal pode ser usado pelos projetistas de sistema para controlar explicitamente o estado dos pinos de dados em cada ciclo de acesso, podendo evitar o fenômeno de contenção (Figs. 4.26 e 4.28, p. 241 e 243). Normalmente, o tempo de chaveamento dos dados válidos para estado "flutuando" pelo OE* é menor do que pelos sinais CS.

Aspecto Temporal

O ciclo de barramento de memórias RAMs varia de alguns até dezenas de nanosegundos. A temporização dos sinais nos dois ciclos (de leitura e de escrita) é fornecida pelos fabricantes.

Modo de Operação Standby

Uma das características mais úteis das memórias RAMs de tecnologia CMOS é a sua capacidade em operar com um nível de potência baixíssima (0.1 mW) no modo denominado *standby*) – o modo no qual não ocorrem mudanças nos estados das células, os dados são somente retidos. Com isso, é possível manter o conteúdo das células dessas memórias com pequenas baterias na falta de força (elétrica), desde que seja projetado um circuito de chaveamento entre os dois modos de operação: modo normal e o modo de *standby*. Fig. 5.33 do livro-texto (p. 349) apresenta a relação funcional e temporal entre os níveis de tensão de alimentação e o sinal de controle CS para caracterizar os dois modos de operação.

Um sistema de memória que suporta os dois modos de operação demanda alguns cuidados. Dois problemas devem ser analisados no projeto, considerando que a bateria seja recarregável pela fonte de alimentação no modo de operação normal:

1. Como deve ser o chaveamento entre as fontes, isolando a bateria da fonte de alimentação normal no modo de operação *standby*?
   - Por um diodo (Fig. 5.34 do livro-texto);
   - Por dois diodos (Fig. 5.35 do livro-texto); ou
   - Por transistores (Fig. 5.36 do livro-texto).
2. Os dispositivos digitais de tecnologia CMOS são mais sensíveis a ruídos, devendo evitar que os seus pinos de entrada fiquem "flutuando". Portanto, deve-se "pull up" as entradas de CMOS para V_{CC} ou "pull down" para o terra. E no modo de operação *standby*, quando não há fonte de alimentação V_{CC} para "pull up", como se deve proceder?

   Usando a propriedade de que os dispositivos de interface LSTTL (*low power Schottky* TTL) apresentam baixa impedância quando a sua alimentação V_{CC} estiver no nível do terra, podemos utilizá-los para "aterrar" os pinos de endereços e de dados da memória no modo de operação *standby*

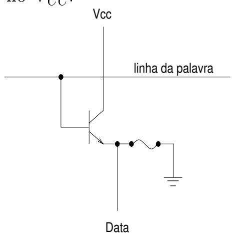
   Para os sinais de controle como CS* e R/W*, que não podem ser aterrados, podemos derivar o nível lógico "verdadeiro" a partir da bateria, como ilustra Fig. 5.38 do livro-texto (p. 354).

Exercícios de Revisão

1. Explique o funcionamento de uma célula de armazenamento de uma memória RAM. Por que o seu conteúdo é perdido na falta de alimentação?
2. O que são memórias voláteis e não-voláteis? Por que algumas memórias RAMs são consideradas "pseudo-não-voláteis"?
3. Qual é a caracterísitca essencial de uma memória estática?
4. O que é modo de operação *standby* de uma memória? Explique a temporização das transições entre os dois modos com uso de um diagrama de tempo.
5. Quais são os problemas que devem ser resolvidos no projeto de um sistema de memória que suporta o modo de operação *standby*?
6. Explique a função dos resistores R_p e das interfaces da família LSTTL no circuito da Fig. 5.38 do livro-texto.

3.3.2 ROMs

Estas memórias pertencem à classe de memórias não-voláteis, uma vez que o seu conteúdo não é perdido sob circunstâncias normais de operação mesmo sem alimentação, já que eletricamente as células com conteúdo 0 é ligado no terra e com conteúdo 1, no V_{CC}.



Dentre as aplicações usuais de ROMs podemos citar:

1. *firmware*, ou seja, armazenamento de instruções e dados permanentes. Em alguns sistemas, o sistema operacional e compiladores/interpretadores são implementados em *firmware*.
2. *bootstrap program* que consiste em um conjunto de instruções para inicializar o *hardware* do sistema e carregar o sistema operacional do disco para Unidade de Memória.
3. tabelas permanentes como tabelas trigonométricas, permitindo a geração de ondas das funções trigonométricas, de conversão de códigos, ou decodificação de endereços.

Das memórias ROMs podemos ainda distinguir as que são reprogramáveis das não-reprogramáveis. Pertencem às não-reprogramáveis:

MROM (*Mask-programmed ROM*) é programável através da máscara-negativa fotográfica. Esta máscara controla as interconexões elétricas na pastilha durante o processo de fabricação.

PROM (*programmable ROM*) é programável pelos usuários com programadores especiais. Estes programadores aplicam pulsos de tensão V_{pp} elevada no dreno de cada célula. Quando o dado for 0, a diferença de tensão aplicada no fusível é suficientemente elevada que leva à sua ruptura; se for 1, o fusível fique intato.

Preparo para Próxima Aula

1. Leia atentamente as páginas 355–375 do livro-texto e descreva as características elétricas e funcionais das seguintes memórias ROMs reprogramáveis:
   - EPROMs;
   - *flash* EEPROMs; e
   - E^2PROMs.
2. Quais são as diferenças entre estas memórias em termos de custo, densidade, velocidade, remoção do conteúdo e reprogrmação?

Existem essencialmente duas tecnologias de reprogramação:

- a tecnologia de injeção de elétrons quentes que requer um valor maior na alimentação V_{CC}/V_{pp}. Neste caso, um circuito adicional de chaveamento deve ser incluído no sistema para suprir os dois níveis de tensão, como mostra a Fig. 5.51 do livro-texto (p. 370).
- a tecnologia de canalização, que não necessita de condições elétricas diferenciadas para programação.

Aspecto Funcional

Os ciclos de leitura e de escrita das memórias ROMs são funcionalmente similares aos ciclos das memórias RAMs. Os dados válidos são capturados sob o controle dos sinais CS e R/W*.

Algumas ROMs reprogramáveis mais sofisticadas são providas de registradores de controle e podem suportar outros modos de operação especificados explicitamente pels usuários, como verificação do conteúdo da memória, verificação do conteúdo programado, etc.

Aspecto Temporal

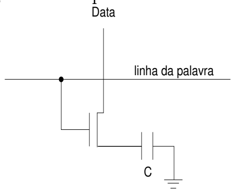
As relações temporais dos sinais são similares às das memórias RAMs. Vale ressaltar que os ciclos de escrita nas memórias reprogramáveis tem uma duração bem maior, na ordem de grandeza que varia de microsegundos a milisegundos. Isso se deve ao fato de que o ciclo de escrita inclue dois tempos, o tempo de escrita propriamente dito e o tempo de verificação.

Exercícios de Revisão

1. Resolver os exercícios 24, 25 e 27 do capítulo 5 do livro-texto.
2. Explique o funcionamento de uma célula de armazenamento de uma memória ROM. Por que o seu conteúdo não é perdido na falta de alimentação?
3. Por que o ciclo de escrita das ROMs programáveis é maior do que o ciclo de escrita das SRAMs?

3.4 Sistemas de Memória Dinâmica

As memórias dinâmicas, DRAMs, caracterizam-se por necessitarem de regeneração periódica para manter o seu conteúdo, uma vez que os dados são armazenados como carga de um capacitor.



A regeneração consiste essencialmente em ler o conteúdo de cada célula periodicamente para "recarregá-la". Ela pode ser feita por hardware ou por software. Quando é feita por hardware dedicado, o gerador deve ser um potencial mestre de barramento que compartilha com a UCP o uso do barramento para acessar o conteúdo da memória. A periodicidade de regeneração de uma memória DRAM é especificada pelos seus fabricantes. Usualmente, quanto menor a capacidade da memória, menor é o período de regeneração. Por exemplo, 64K (2 ms), 256K (4 ms) e 1M (8 ms).

Em comparação com as SRAMs, as DRAMs são mais baratas e tem uma densidade de armazenamento mais alta (requerem menos transistores por célula). Usualmente, dada uma área de silício, consegue-se armazenar em DRAMs 4 vezes mais dados do que em SRAMs. Porisso, são as preferidas para implementar a Unidade de Memória nos computadores de uso genérico.

O compromisso entre alta densidade e limitação física (número de pinos de endereçamento) é um fator que levou à adoção de **barramento de endereços multiplexado** nas memórias DRAMs. Para um projetista de um sistema de memória DRAM, é conveniente pensar que as suas palavras sejam organizadas em uma estrutura matricial, cada qual endereçável por dois endereços: um para coluna e um para linha. Com esta organização pode-se utilizar o mesmo conjunto de pinos de endereços para especificar a coluna e a linha de uma palavra em duas janelas distintas de tempo. A diferenciação entre um endereço de coluna e um endereço de linha é feita pelos sinais de *strobes* RAS* e CAS*.

DRAMs sofrem ainda o problema de partícula alpha (íons de Hélio), que em geral provém da contaminação do material de encapsulamento. Esta partícula, ao passar pela célula de memória, pode causar ionização suficiente a

corromper os dados armazenados, gerando **erros** nos dados. Para minimizar este problema, reforça-se o controle de qualidade na fabricação ou utiliza-se códigos detectores/corretores de erro. Há tendência de embutir circuitos de deteção/correção de erros nas pastilhas.

Outra propriedade das memórias DRAMs é que, durante a inicialização do sistema, elas requerem um tempo de polarização que leva em torno de 200 μs para entrar em "pleno funcionamento" e que entre dois acessos precisa executar algumas operações internas conhecidas como **pré-carga**.

3.4.1 Aspecto Funcional

Os pinos de endereços das DRAMs podem desempenhar três funções:

- prover o endereço da linha quando o sinal de strobe RAS* estiver ativo durante o ciclo de operação normal;
- prover o endereço da coluna quando o sinal de strobe CAS* estiver ativo durante o ciclo de operação normal; e
- prover o endereço da linha de regeneração quando o sinal de strobe RAS* estiver ativo durante o ciclo de regeneração.

No ciclo de acesso normal a uma palavra o endereço da linha é fornecido antes do endereço da coluna, ou seja, RAS* é ativado antes do CAS*. E no ciclo de regeneração? Quais são os sinais de controle utilizados para diferenciar o ciclo de regeneração do ciclo normal? A maioria das DRAMs adotam uma das seguintes quatro técnicas:

somente RAS* : somente RAS* é ativado durante o ciclo de acesso quando o endereço da linha a ser regenerado é fornecido (Fig. 5.80 do livro-texto, p. 403).

CAS* antes de RAS* : nas DRAMs de nova geração, o endereço da linha é gerada internamente (*on–chip*). Neste caso, pode-se distinguir um ciclo de operação normal de um ciclo de regeneração invertendo a sequência de ativação dos sinais CAS* e RAS* (Fig. 5.81 do liro-texto, p. 404).

implícita : o ciclo de regeneração ocorre quando os dados nos pinos de saída são ainda válidos. No fim de um ciclo de acesso normal, CAS* mantém-se ativo enquanto a borda de descida do RAS* engatilha o ciclo de regeneração (Fig. 5.82 do livro-texto, p. 405).

de pino 1 : algumas DRAMs de geração velha reservam o pino 1 ao controle do ciclo de regneração. Quando o sinal deste pino estiver ativo, o ciclo de regeneração é executado (Figs. 5.83 e 5.84 do livro-texto, p. 405 e 406).

Existem várias formas de introduzir o ciclo de regeneração na operação normal de um microprocessador: por rajada (de uma vez todas as linhas) ou por entrelaçamento (distribuído ao longo dos acessos normais).

A organização multiplexada dos endereços abre ainda um espaço para variar os modos de acesso na operação normal de uma DRAM a fim de melhorar o seu desempenho, como

modo paginado (FPM – fast page mode): permite sucessivos acessos em uma mesma linha, mantendo o mesmo endereço de linha no registrador (RAS* ativo) e variando os endereços de coluna (com um trem de pulsos do sinal CAS*) (Fig. 5.66 do livro-texto, p. 391). Este modo é particularmente útil para transferência de dados de posições contíguas (*bursts* de dados). Uma variação do modo paginado é o modo EDO (*extended data out*).

modo *nibble* : permite quatro acessos sequenciais em uma mesma linha (RAS* ativo) a partir do endereço de coluna fornecido pela UCP (Fig. 5.67 do livro-texto, p. 391).

modo de coluna estática : permite acessos sucessivos em uma mesma linha mantendo os sinais RAS* e CAS* ativados e variando somente os endereços da coluna (Fig. 5.68 do livro-texto, p. 392). Este modo é utilizado em sistemas de alto desempenho.

modo de ciclo de leitura-escrita : permite executar um ciclo de leitura seguido de um ciclo de escrita com um único acesso aos pares de endereços de linha e coluna.

Preparo para Próxima Aula

1. Leia atentamente as páginas 377–389 do livro-texto e descreva sucintamente com uso do diagrama de tempo os ciclos de leitura e de escrita de uma memória DRAM.

2. Responda ainda:
   - Qual é a sequência de ativação dos sinais RAS* e CAS*?
   - Por que o tempo t_{RCD} é um pseudo-máximo?
   - Quais são os tempos de *setup* e *hold* de dados no ciclo de leitura? Qual é o sinal de gatilho?
   - Quais são os tempos de *setup* e *hold* de dados no ciclo de escrita? Qual é o sinal de gatilho?

3.4.2 Aspecto Temporal

A média do tempo destinado para um ciclo de acesso a uma memória DRAM é maior do que a média de tempo de acesso a uma memória SRAM devido à operação de pré-carga entre os acessos.

O ciclo de regeneração é temporalmente igual ao ciclo de leitura no modo de operação normal. Eles diferem apenas no dispositivo que controla os acessos. No primeiro caso, o mestre de barramento é o "circuito de controle" de regeneração e no segundo, a UCP.

Exercícios de Revisão

1. Resolver os exercícios 28, 29, 30, 31, 32, 35, 36, 37, 38, 39, 42 e 44 do capítulo 5 do livro-texto.

3.5 Memória Cache

Memórias cache são, de fato, memória estáticas SRAMs de alta velocidade (na ordem de grandeza de dezenas de nanosegundos). Elas são utilizadas pelos projetistas para explorar melhor o desempenho de um processador sem elevar demasiadamente o custo do sistema de memória (lembrem-se de que as DRAMs são muito mais baratas do que as SRAMs), com base no **princípio de localidade**. Fig. 7.41 do livro-texto (p. 573) iluatra uma organização de um sistema de memória com cache. Note que um controlador é necessário para verificar a localização do dado referenciado pela UCP em cada acesso.

Se o dado estiver na memória cache, ele ativa o sinal *hit* indicando acesso à memória cache no lugar da memória principal; caso contrário, a memória principal é acessada e o conteúdo do cache atualizado.

A razão entre o tempo gasto num sistema sem memória cache e num sistema com memória cache é uma função da percentagem de acessos à memória cache (H) num total de N acessos:

$$S = \frac{Nt_m}{N(Ht_c + Mt_m)} \tag{3.1}$$

$$= \frac{t_m}{Ht_c + (1-H)t_m} \tag{3.2}$$

$$= \frac{\frac{t_m}{t_m}}{H\frac{t_c}{t_m} + (1-H)\frac{t_m}{t_m}}. \tag{3.3}$$

Denominando $\frac{t_c}{t_m} = k$ por **taxa de aceleração do acesso**, temos

$$S = \frac{1}{1 - H(1-k)}.$$

A memória cache tem normalmente uma capacidade bem menor do que a da memória principal. Portanto, a correspondência entre os blocos de dados (também conhecidos como linhas) na memória cache e na memória principal não é 1:1. Como é então organizada a memória cache para facilitar o mapeamento? São conhecidas três orgnizações, nas quais os dados tanto da memória principal quanto da cache são agrupados em blocos/linhas:

1. Mapeamento direto: a correspondência é feita através da função

   $$i = j \ mod \ m,$$

   com i, j e m denotando número de bloco na memória cache, número de bloco na memória principal e quantidade de blocos na memória cache, respectivamente. Como o par (i, j) é fixo, pode ocorrer trocas frequentes mesmo com blocos vazios no cache (Figs. 7.43 e 7.45 do livro-texto).

2. Mapeamento associativo: não existe nenhuma relação entre o número de um bloco no cache com o número de um bloco na memória principal. Memória associativa é utilizada na localização de um bloco no cache pelo seu rótulo. Quando um bloco não se encontra no cache em um acesso, um dos algoritmos de troca, como *FIFO*, *random*, *the least-recently used* e *the leas-frequently used*, é utilizado para decidir qual bloco a ser substituído (trocado) (Fig. 7.46 do livro-texto, p. 580).

3. Mapeamento ao conjunto associativo: o sistema é constituído por um conjunto de n caches de mesmo tamanho (mesmo número de blocos), isto é, no lugar de (i, j), temos (i_K, j) onde $i_1 = i_2 = \cdots = i_K = \cdots = i_n$. Com isso, n blocos distintos de memória principal, porém mapeados no mesmo valor i, podem coexistir nos caches, evitando trocas frequentes. Ë um compromisso entre o mapeamento direto e mapeamento associativo (Fig. 7.47 do livro-texto).

Exercícios de Revisão

1. Resolver os exercícios 29, 30, 32 e 35 do capítulo 7 do livro-texto.

3.6 Auto-avaliação

Após este capítulo, você deve ser capaz de:

- saber o princípio básico de decodificação e as diferentes estratégias.
- projetar um decodificador de endereços.
- saber as funções lógicas providas pelos diversos dispositivos lógicos e utilizá-los para implementar um decodificador.
- descrever as características das SRAMs e o seu funcionamento.
- explicar como se pode manter o conteúdo de uma memória SRAM de tecnologia CMOS sem a fonte de alimentação e quais cuidados devem ser tomados no projeto deste sistema de memória.
- descrever as características das ROMs e o seu princípio de funcionamento.

- saber os diferentes tipos de ROMs e as caracterísitcas de cada um que interessam ao projetista de um sistema de memória.
- descrever as peculiaridades das DRAMs e como estas podem influenciar no projeto de um sistema de memória.
- explicar por que os sinais BR*, BG* e BACK* são necessários no controle do ciclo de regeneração de uma DRAM.
- explicar o projeto apresentado nas pp. 408 – 414 do capítulo 5 do livro-texto.
- citar as aplicações dos diferentes tipos de memórias.
- projetar sistemas de memória com uso de RAMs, ROMs e DRAMs.
- descrever as memórias caches.
- estimar a taxa teórica de aceleração em um sistema de memória com cache.
- explicar as três organizações de memórias caches.

3.7 Exercícios

1. Resolver o exercício 9 do capítulo 5 do livro-texto.
2. Resolver o exercício 15 do capítulo 5 do livro-texto.
3. Resolver o exercício 41 do capítulo 5 do livro-texto.
4. Resolver o exercício 36 do capítulo 7 do livro-texto.
5. Procurar a descrição de uma memória cache (Uma referência útil: `www.kingston.com`).