



**“FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”  
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA  
UNIVEM**

**Bacharelado em Ciência da Computação**

Disciplina

**Organização e Arquitetura de Computadores**

**Capítulo II**

**Subsistemas de Memória**

**(Tecnologias de Memória)**

**(Classificação e Organização de Memórias)**

**(Projetos de Subsistemas de Memória Semicondutoras)**

**Prof. - Ildeberto de Genova Bugatti**

[bugatti.univem@gmail.com](mailto:bugatti.univem@gmail.com)

## **Identificação dos pré-requisitos**

São considerados pré-requisitos para o efetivo acompanhamento do tema os seguintes conceitos ou técnicas: bases numéricas, representação de números nas bases dois e hexadecimal, álgebra de boole, lógica digital; síntese de circuitos lógicos digitais combinacionais e sequenciais; elementos multiplexadores; elementos decodificadores; técnicas de associação de componentes eletrônicos em série e paralelo; subsistemas básicos de uma arquitetura de computador.

## **Objetivos**

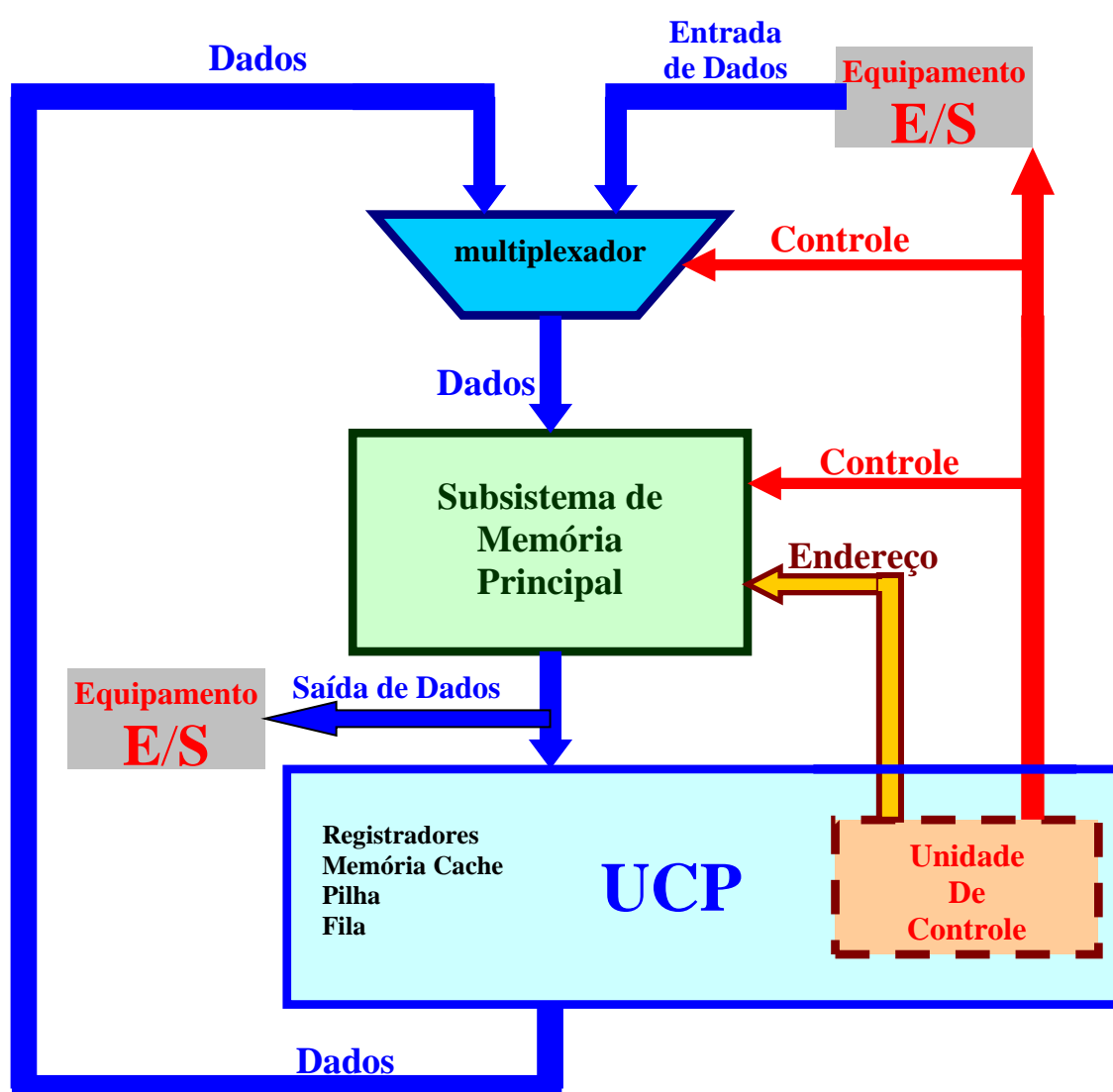
Transferir aos leitores conceitos e técnicas de classificação e organização de memórias, identificando esses conceitos nas principais tecnologias de integração de componentes de memórias e capacitá-los para realizar as seguintes atividades: sintetizar, dimensionar, projetar, validar e avaliar o impacto de diferentes mecanismos e estruturas no desempenho de subsistemas de memória em diferentes aplicações.

## **7- Subsistemas de Memória**

Para descrever os tipos de memória de acordo com as suas tecnologias e características será utilizado para exemplificar os subsistemas de memória existentes em uma arquitetura de computador.

Um Subsistema de Memória tem a função de armazenar (conter) informações durante um período. Existe uma diversidade de dispositivos de memória que diferem quanto a suas características de: armazenamento de informações, capacidade de armazenamento, velocidade de acesso e manutenção da informação. Os conceitos, características e formas de organização dessas memórias são apresentados nos itens que seguem. Para tanto, será realizada discussão quanto as seguintes medidas: custo, capacidade de armazenamento e velocidade de acesso à informação; para cada uma das formas existentes de classificação e organização de subsistemas de memória.

Será utilizada, para exemplificar e caracterizar os diversos subsistemas de memória existente uma arquitetura clássica de computador. A Figura 2.1 mostra os três principais subsistemas de uma arquitetura genérica (memória principal, Unidade central de processamento e Unidades de Entrada e Saída). No bloco UCP existe além da ULA diversos sistemas, grande parte deles caracterizam-se como subsistema de memória, tais como: registradores de uso geral, registradores de uso específico (acumulador, Contador de programa, registrador de instrução, e outros), sistema de pilha, sistema de fila, sistema de memória cache, entre outros. Grande parte dos equipamentos de entrada de saída também se caracterizam como subsistemas de memória, tais como: disco rígido, DVD, Pen Drives, entre outros.



*Figura 2.1 – Diagrama de Blocos da Arquitetura de Computador Genérica*

O byte (conjunto de oito bits) será utilizado como unidade de medida de capacidade de armazenamento para qualquer tipo ou classificação de memória. Assim, o byte será a unidade utilizada como referencial para comparar custos entre os diferentes tipos de subsistemas de memória existentes.

## **2.1- Critérios Utilizados na Classificação de Memória.**

Os sistemas de memória são classificados de acordo com duas características principais: formas de acesso à informação e formas de manutenção da informação. Quanto à manutenção da informação existem dois critérios para classificação: manutenção da informação em relação à disponibilidade de energia (alimentação) e manutenção da informação em relação ao tempo.

### **2.1.1- Classificação quanto ao Acesso à Informação.**

Quanto ao acesso existem basicamente dois tipos de memória: memórias acesso sequencial e memórias de acesso aleatório.

Em um sistema de acesso sequencial o tempo gasto para acessar palavras contidas em diferentes posições da memória não é homogêneo. Ou seja, o tempo necessário para acessar um registro localizado fisicamente no início da memória é menor do que o tempo necessário para acessar um registro localizado fisicamente no meio da memória e muito menor do que o tempo necessário para acessar um registro localizado fisicamente no final da memória. Geralmente, memórias de acesso sequencial possuem baixa velocidade de acesso. No entanto, possuem enorme capacidade de armazenamento e baixo custo por byte.

Como exemplos de memória de acesso sequencial temos a maioria dos sistemas de armazenamento secundário dos computadores. A memória secundária geralmente é controlada por subsistemas de Entrada e Saída. Podemos citar como exemplo: discos rígidos, disquetes; CDs, DVDs, fitas magnéticas entre outros (fitas DAT).

Em sistemas de acesso aleatório, o tempo necessário para acessar qualquer posição de memória é sempre o mesmo. Ou seja, o tempo de acesso a qualquer registro de um sistema de acesso aleatório é constante. O tempo para acessar um registro que está posicionado no início da memória é igual tanto ao tempo de acesso de um registro localizado fisicamente no meio da memória quanto no final da mesma. Geralmente, os sistemas de acesso aleatório possuem grande velocidade de acesso. Como exemplo de memória de acesso aleatório temos as memórias semicondutoras, utilizadas na memória principal de um computador.

A Tabela 2.1 mostra as principais diferenças entre memórias de acesso sequencial e memória de acesso aleatório.

<b>Características e Classificação de Memórias quanto à forma de Acesso aos dados</b>				
<b>Classificação</b>	<b>Custo</b>	<b>Capacidade</b>	<b>Velocidade</b>	<b>Exemplos</b>
<b>Sequencial</b>	Baixo	Alta	Baixa ( $\mu$ s)	Fitas magnéticas, CDs, DVDs Discos Rígidos (HD), Disquetes
<b>Aleatório</b>	Alto	Baixa	Alta (ns)	Memórias Semicondutoras, Memória Principal, Memória CACHE

**Tabela 2.1- Memórias Sequenciais e Aleatórias.**

As memórias sequenciais geralmente são muito mais lentas que as memórias aleatórias por necessitarem, geralmente, de sistemas de controle mecânicos para atingir o endereço físico (posição) do registrador que armazena a informação. Nos discos rígidos e nos disquetes existem: tanto o movimento giratório para girar os discos, quanto o movimento de movimentação linear dos cabeçotes de escrita e leitura. As características de alguns desses dispositivos serão apresentadas no item 2.1.3.

Existem também sistemas de acesso mistos, conjugando a potencialidade de armazenamento dos sistemas sequenciais com a velocidade de acesso dos sistemas aleatórios.

## **2.1.2- Classificação de Memória quanto à manutenção da Informação.**

Uma característica importante de subsistemas de armazenamento de dados está relacionada à manutenção da informação no seu interior. Existem dois parâmetros bastante utilizados para gerar essa classificação que são: manutenção da informação em relação à disponibilidade de energia (alimentação) e manutenção da informação em relação ao tempo.

### **2.1.2.1- Manutenção da Informação quanto à disponibilidade de Energia.**

Quanto à manutenção da informação em relação à disponibilidade de energia, podemos caracterizar os sistemas em memórias Voláteis e memórias Não Voláteis.

Os sistemas voláteis perdem a informação na inexistência de energia. Exemplos de aplicação: todas memórias semicondutoras implementadas com Flip-Flops, tal como os subsistemas de memória CACHE dos computadores; memórias semicondutoras implementadas com elementos capacitivos, tal como a maioria dos sistemas de memória principal dos computadores; memórias de dados temporários de calculadoras, entre outros. A organização, aplicação e características de subsistemas de memória CACHE será apresentado no item 2.2.

Os sistemas não voláteis, mantêm a informação mesmo na ausência de energia. Exemplos de aplicação: discos rígidos, CDs, DVDs, fitas magnéticas, disquetes, memórias semicondutoras apenas de leitura (ROMs, PROMs, EPROMs, EEPROMs, memória FLASH, memórias dos PEN Drivers, entre outras). Os termos, utilizados para denominar as memórias apenas de leitura acima relacionados, serão descritos de forma mais detalhada no item 2.1.3.

A Tabela 2.2 mostra as principais características e diferenças entre memórias Voláteis e memórias Não-Voláteis.

<b>Características e Classificação de Memória - Manutenção da Informação com Energia</b>					
<b>Classificação</b>	<b>Manutenção Sem Energia</b>	<b>Custo</b>	<b>Capacidade</b>	<b>Velocidade</b>	<b>Exemplos</b>
<b>Volátil</b>	Perde Informação	Alto	Média	Média/alta (ns)	Memória principal (capacitivas)
			Baixa	Alta (ns)	Memória CACHE (Flip-Flops)
<b>Não Volátil</b>	Mantém Informação	Baixo	Alta	Baixa ( $\mu$ s)	Fitas magnéticas, CD, DVD, Disco Rígido, Disquete, ...
		Alto	Média	Média/alta (ns)	ROMs, PROMs, EPROMs, EEPROMs (FLASH), ...

**Tabela 2.2- Memórias Voláteis e Não-Voláteis**

### **2.1.2.2- Manutenção da Informação no Tempo**

Quanto à manutenção da informação no tempo, os sistemas são classificados como memórias estáticas e memórias dinâmicas.

As memórias estáticas, mantêm a informação por tempo indeterminado na existência de energia. Exemplos: sistemas de memória CACHE, todas memórias semicondutoras implementadas com Flip-Flops.

As memórias dinâmicas, mesmo com existência de energia, podem perder a informação, isso acontece com o decorrer do tempo. Ou seja, a informação deteriora-se após um período determinado, geralmente medidos em microssegundos. Para que a informação não seja perdida, são realizadas operações periódicas e cíclicas que recuperam as informações (registros) contidas em todas as palavras do sistema. Essas operações são denominadas de “REFRESH” (refrescamento). O procedimento de “REFRESH” é realizado periodicamente e consiste na realização de um ciclo de leitura seguido de um ciclo de escrita em todas as palavras do sistema de memória dinâmica. Ou seja, antes que um registro seja perdido no tempo, ele é lido e novamente armazenado no mesmo endereço da memória, dessa forma a

integridade da informação é recuperada. Esse procedimento é batizado de sistema de refrescamento de memória.

A Tabela 2.3 mostra a classificação, características e principais diferenças entre memória estática e memória dinâmica.

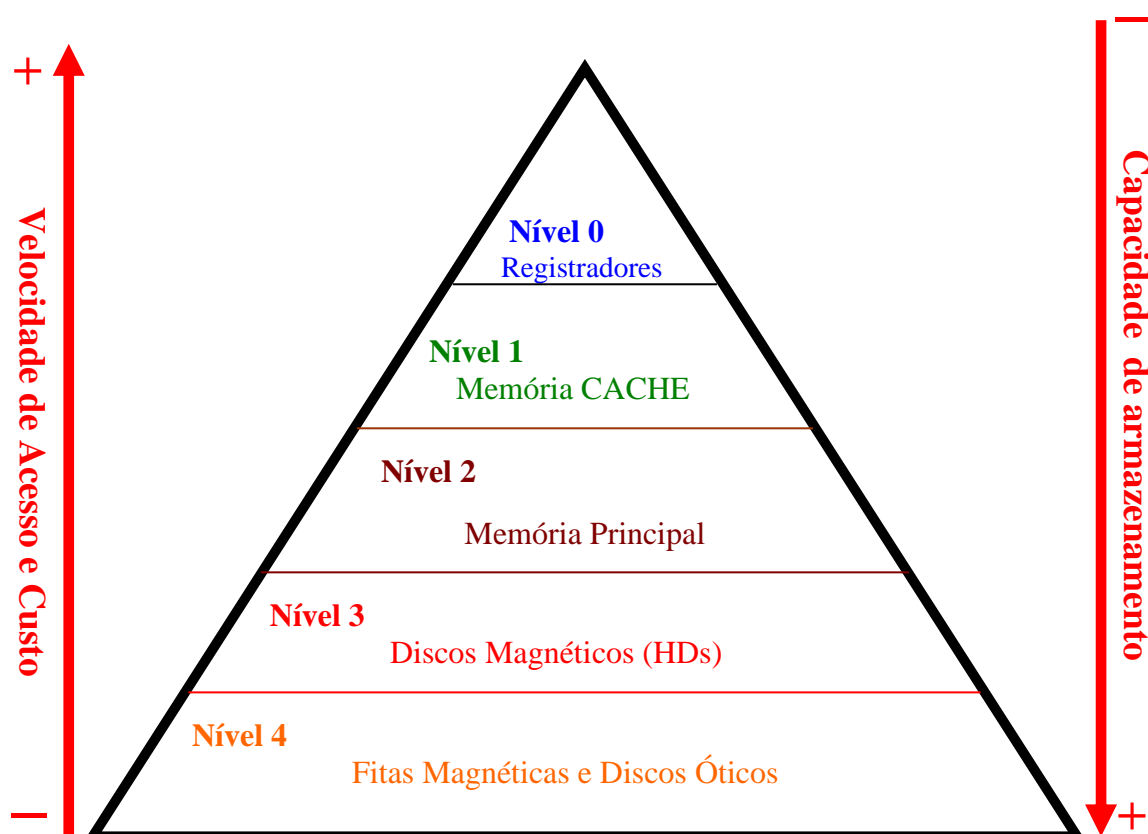
Características e Classificação de Memória - Manutenção da Informação no Tempo					
Classificação	Informação no Tempo	Custo	Capacidade	Velocidade	Exemplos
<b>Estática</b>	Mantém Informação	Alto	Baixa	Alta	Memória CACHE (Flip-Flops)
<b>Dinâmica</b>	Perde Informação	Alto	Média	Média/Alta	Memória Principal (sistemas capacitivos)

**Tabela 2.3- Memórias Estáticas e Dinâmicas**

Os conceitos, formas de organização e técnicas de projeto e implementação de sistemas de memórias estáticas e aleatórias serão apresentados no item 2.2.

- **Nível 0** (topo): os Registradores da CPU que são subsistemas que armazenam apenas um registro, no entanto, numa velocidade muito alta, compatível ou maior que a velocidade da CPU. É classificada da seguinte forma: volátil, acesso aleatório e estática;
- **Nível 1**: Memória CACHE que possui capacidade relativamente baixa de armazenamento de dados, atualmente na faixa de megabytes podendo atingir dezenas de megabytes. A capacidade de armazenamento da memória CACHE está diretamente relacionada ao desempenho do equipamento. Quanto maior a capacidade de armazenamento da memória CACHE, maior será o desempenho do computador. Seu custo é alto e a velocidade de acesso é alta. Além disso, devem ser discutidos outros aspectos inerentes a uma memória CACHE que são: organização (comprimento da palavra e tamanho das páginas); organização e gerenciamento. A memória CACHE e a memória principal trabalham de forma simultânea, a memória CACHE contém uma cópia (xerox) de um subconjunto de dados existentes da memória principal. É classificada da seguinte forma: volátil, acesso aleatório e estática;
- **Nível 2**: Memória Principal, possui capacidade média de armazenamento de dados, na faixa de Megabytes a dezenas de Gigabytes, tem a função de conter dados de forma temporária. A memória principal é geralmente implementada com memórias semicondutoras com células capacitivas. É classificada da seguinte forma: volátil, acesso aleatório e dinâmica;

- **Nível 3:** Discos Magnéticos, existem discos flexíveis (disquetes), já em desuso, e discos rígidos que possui grande capacidade de armazenamento de informação, na faixa de centenas ou milhares de Gigabytes. Os discos rígidos são de baixo custo e tempo de acesso lento, pois funcionam com sistema mecânicos para introduzir e retirar dados em e de seu interior, tais como: rotação do disco e deslocamento de cabeçotes de leitura e escrita. É classificado da seguinte forma: não-volátil, acesso seqüencial e estático;
- **Nível 4:** Fitas magnéticas e discos óticos, com capacidade muito grande de armazenamento, na faixa de terabytes, a um custo muito baixo. Geralmente, esses sistemas de memória são utilizados para armazenamento de arquivos e sistemas de backups.



**Figura 2.2- Organização Clássica de Hierarquia de Memória**

Com o desenvolvimento contínuo da tecnologia de semicondutores e microeletrônica através de pesquisas de novos processos e de novos materiais, visualizam-se os seguintes cenários:

- As memórias dinâmicas (memória principal) estão ficando cada vez mais rápidas, e as memórias estáticas (CACHE) estão aumentando sua capacidade de armazenamento



a um menor custo. Assim, num determinado ponto desse desenvolvimento essas memórias irão apresentar capacidade de armazenamento, velocidade de acesso e custos muito semelhantes, podendo ser eliminados os sistemas de memória CACHE, pois a memória principal poderá ser implementada em sua totalidade por memória estática;

- as memórias semicondutoras não voláteis, similares às memórias FLASH, estão aumentando rapidamente sua capacidade de armazenamento a um custo cada vez menor, assim, de forma natural, sistemas de memórias semicondutoras irão, num futuro próximo, substituir os discos rígidos que possuem tempo de acesso centenas a milhares de vezes maior que as memórias semicondutoras, pois funcionam com sistema mecânicos (rotação do disco e posicionamento de cabeçotes) para introduzir e retirar dados em e de seu interior.

Caso confirme-se essa previsão, serão eliminados um ou dois níveis nessa hierarquia clássica de organização de memória.

### **2.1.3- Características de Dispositivos de Armazenamento de Informação**

Nos itens anteriores foram discutidos os critérios utilizados para classificar subsistemas de memória e foram relacionados como exemplos alguns dispositivos ou equipamentos existentes nas diversas formas de classificação. Nesse item serão discutidos principalmente os dispositivos e equipamentos mais comumente utilizados na implementação de memórias secundárias dos computadores, níveis 3 e 4 da organização hierárquica de memória apresentada na figura 2.2. Para cada dispositivo será apresentado: um breve histórico, a tecnologia utilizada e formas de organização da informação.

#### **2.1.3.1- Discos Rígidos e Discos Flexíveis**

Conforme descritos anteriormente, os discos rígidos e flexíveis são utilizados para armazenamento secundário de dados. Eles possuem grande capacidade de armazenamento e baixa velocidade de acesso à informação e baixo custo, são classificados como memórias: não-voláteis, acesso seqüencial e estáticas. Na organização hierárquica (Figura 2.2) eles estão presentes no nível 3.

A denominação “disco” é oriunda da forma circular de construção dos elementos que contém revestimento magnetizável. Os discos são constituídos por um ou mais discos (pratos) cujo diâmetro está sempre diminuindo com o desenvolvimento de novas tecnologias e utilização de novos materiais. Atualmente existem discos rígidos com diâmetro menor que três centímetros.

Os discos flexíveis já foram muito utilizados para armazenagem de arquivos e realizar a troca de arquivos entre equipamentos, no entanto, com o advento de novas tecnologias, foram disponibilizados novos equipamentos, tais como: CD-RW, Pen-drivers entre outros; que substituem os disquetes com muitas vantagens.

Os discos rígidos são construídos com diversos “pratos” empilhados, geralmente de alumínio, e por cabeçotes de leitura e escrita, um ou dois para cada prato. Os pratos possuem diversas trilhas cilíndricas e concêntricas. Ao conjunto de trilhas que estão a mesma distância do centro dos discos é denominado cilindro. As trilhas são formadas por diversos setores. Os setores podem conter um conjunto uma determinada quantidade de bytes, geralmente 512 bytes. A Figura 2.3 ilustra organização de um disco (prato).

Existem duas formas de organizar um disco, inserir em cada trilha a mesma quantidade de setores ou inserir maior quantidade de setores quanto maior for a circunferência da trilha (mais distante do centro).

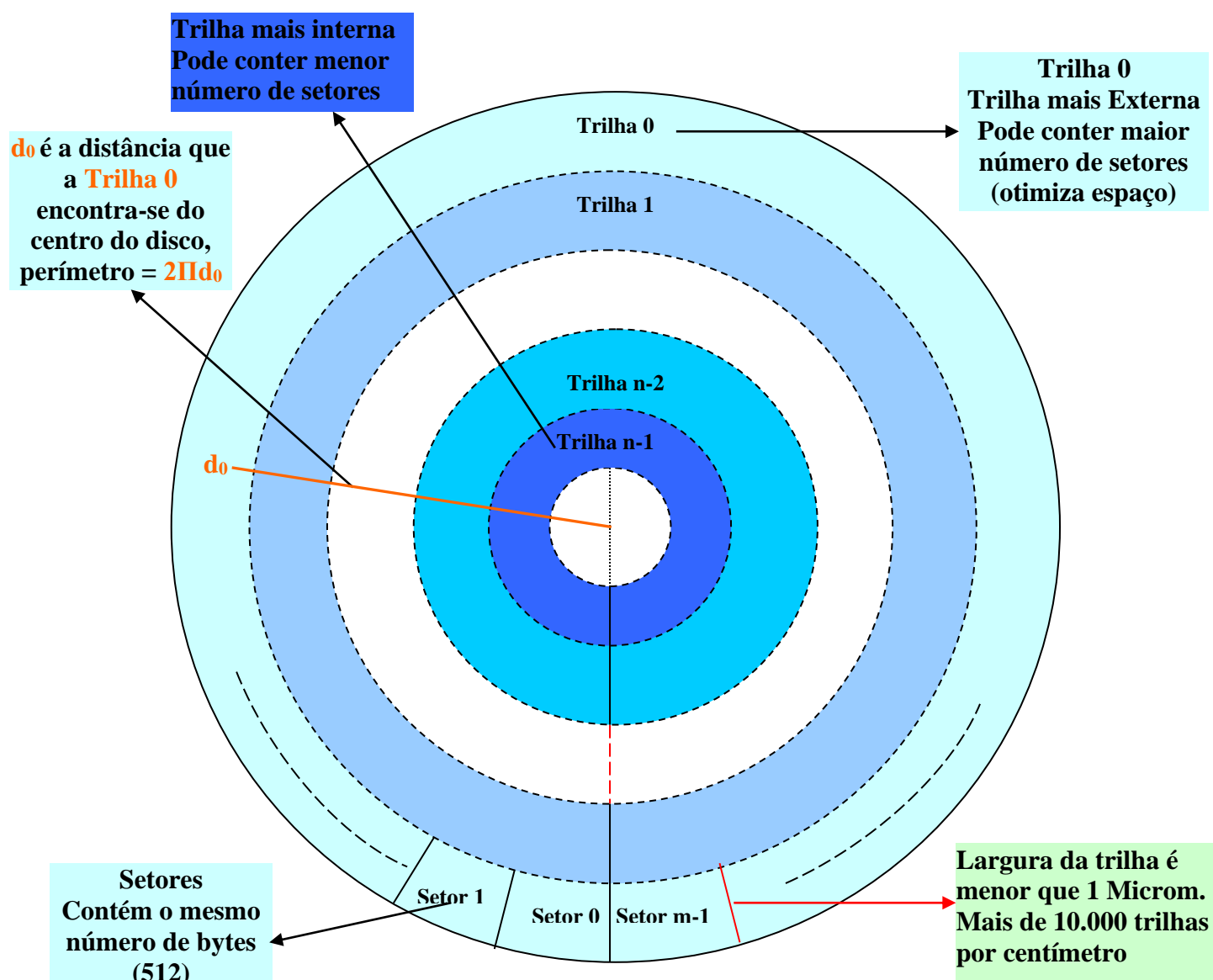
No entanto, o perímetro das trilhas mais próximas do centro é menor do que o perímetro das trilhas mais distantes do centro, pois o perímetro de uma circunferência é dado por  $2\pi d$ . Onde “d” é a distância que a trilha se encontra do centro.

Assim na formatação onde são inseridos a mesma quantidade de setores por trilha, como os setores possuem dimensões idênticas, nas trilhas mais externas existe perda da capacidade de armazenagem, pois a quantidade de setores por trilha é determinada pela trilha mais interna, de menor perímetro. No entanto, essa forma de formatar o disco rígido apresenta maior velocidade de acesso, porque a localização dos setores nas trilhas fica simplificada.

Quando as trilhas são constituídas por quantidades diferentes de setores é otimizado a capacidade de armazenagem do disco. No entanto, a localização dos setores em cada trilha é realizada de forma diferente, acarretando um controle mais elaborado e, conseqüentemente, aumenta o tempo de acesso a informação. Ou seja, nessa forma de formatar o disco quanto mais distante do centro, mais setores uma trilha pode conter. Essa organização dificulta a localização dos setores, no entanto aumenta a capacidade de armazenagem.

O desempenho de um disco rígido está diretamente relacionado à taxa de transferência. Atualmente existem discos com taxas de transferência próxima de 80 MB/s (Megabytes por segundo). A taxa de transferência depende da velocidade de rotação dos discos, atingindo mais que 10.800 RPM (rotações por minuto). Com essa faixa de transferência, um setor de 512 bytes demora quase  $7\mu s$  (microsegundos,  $1\mu = 10^{-6}$ ). No entanto, essa taxa de

transferência é medida após o posicionamento correto dos cabeçotes de leitura e escrita. O tempo de posicionamento dos cabeçotes depende: tanto da velocidade de rotação, quanto da velocidade de deslocamento dos cabeçotes. O tempo de posicionamento correto dos cabeçotes é denominado de tempo de busca (seek).



**Figura 2.3- Organização de cada “prato” de um disco rígido**

O tempo gasto no posicionamento dos cabeçotes (busca) é muito maior que tempo de transferência de dados, medido em unidades de milissegundos ( $1\text{ms} = 10^{-3}$  segundos). Portanto, ler setores de forma aleatória é uma forma muito ineficiente de operar discos rígidos.

O tempo gasto para posicionar os cabeçotes em trilhas vizinhas é sempre menor, portanto, organizar os arquivos e setores de uma mesma trilha e em trilhas vizinhas aumenta consideravelmente a taxa de transferência de dados.

A capacidade de armazenamento dos discos está diretamente relacionada à quantidade de trilhas que podem ser inseridas em uma área do disco e na quantidade de setores presentes em cada trilha. A tecnologia disponível permite a inserção de mais de 10.000 trilhas por centímetro, o que resulta numa largura de trilha menor que 1 microm ( $1 \text{ microm} = 10^{-4} \text{ mm}$ ). A densidade linear de bits dentro de uma trilha pode ser maior que 100.000 bits/cm e cada trilha mede “211d” centímetros. A figura 2.3 ilustra a distribuição das trilhas em todos os “pratos” de um disco e a distribuição dos setores em uma trilha.

Deve-se observar que a capacidade de armazenamento de um disco formatado é 15% menor do que sua capacidade máxima. Pois após a formatação, algumas pequenas regiões (áreas) do disco são utilizadas para introduzir informações adicionais que auxiliam o controle de posicionamento de cabeçotes e informações que contribuem para aumentar a consistência dos dados através da verificação e correção de possíveis erros de leitura e escrita no disco, através da utilização de códigos corretores de erros, tais como, códigos de Hamming ou códigos Reed-Solomon.

### **2.1.3.2- Discos óticos (Compact Disk – CD, CDR, CD-RW e DVD)**

Os CDs óticos são utilizados para armazenar dados, música, imagem e vídeo. São classificados como memórias: não-voláteis, acesso seqüencial e estáticas. Na organização clássica de hierarquia de memória (Figura 2.2) eles estão presentes no nível 4, e geralmente implementam sistemas de memória secundária de computadores.

Os primeiros discos óticos surgiram nos anos 80 e foram utilizados comercialmente com muito sucesso para armazenar e comercializar músicas substituindo rapidamente os discos de vinil, daí surgiu a denominação Compact Disk (CD). Os primeiros discos óticos eram utilizados para gravar programas televisivos.

Os CDs de músicas são denominados CD-ROMs pois permitem apenas a leitura de dados. O processo de escrita em um CD-ROM envolvia a construção de discos matrizes que eram depois reproduzidos em equipamentos especiais para gerar cópias desse CD matriz. Tanto o processo quanto equipamentos de escrita eram muito caros para viabilizar o uso comercial. Assim, o uso comercial só era viabilizado através da produção e de uma quantidade muito grande de cópias de um mesmo conteúdo para diluir o custo de cada CD-ROM, característica muito adequada à indústria fonográfica.

No entanto, a evolução da indústria de computadores também viabilizou a evolução dos CR-ROMs surgiu nos anos 90 os CDRs (CD – Recordables), CDs graváveis. Os equipamentos para gravar dados nos CDs ficaram com custo muito acessível assim como a

tecnologia tornou-os tão pequenos que foi possível inseri-los nos gabinetes dos computadores de mesa e até mesmo nos computadores portáteis (notebooks). A capacidade de armazenamento de dados em um CDR é compatível a capacidade de um CD-ROM, essa limitação é consequência da intenção de manter compatibilidade entre CD-ROMs e CDRs, pois a tecnologia permite processos de gravação e leitura em discos óticos de capacidade muito maior. Após o processo de gravação o conteúdo do CDR não pode mais ser modificado.

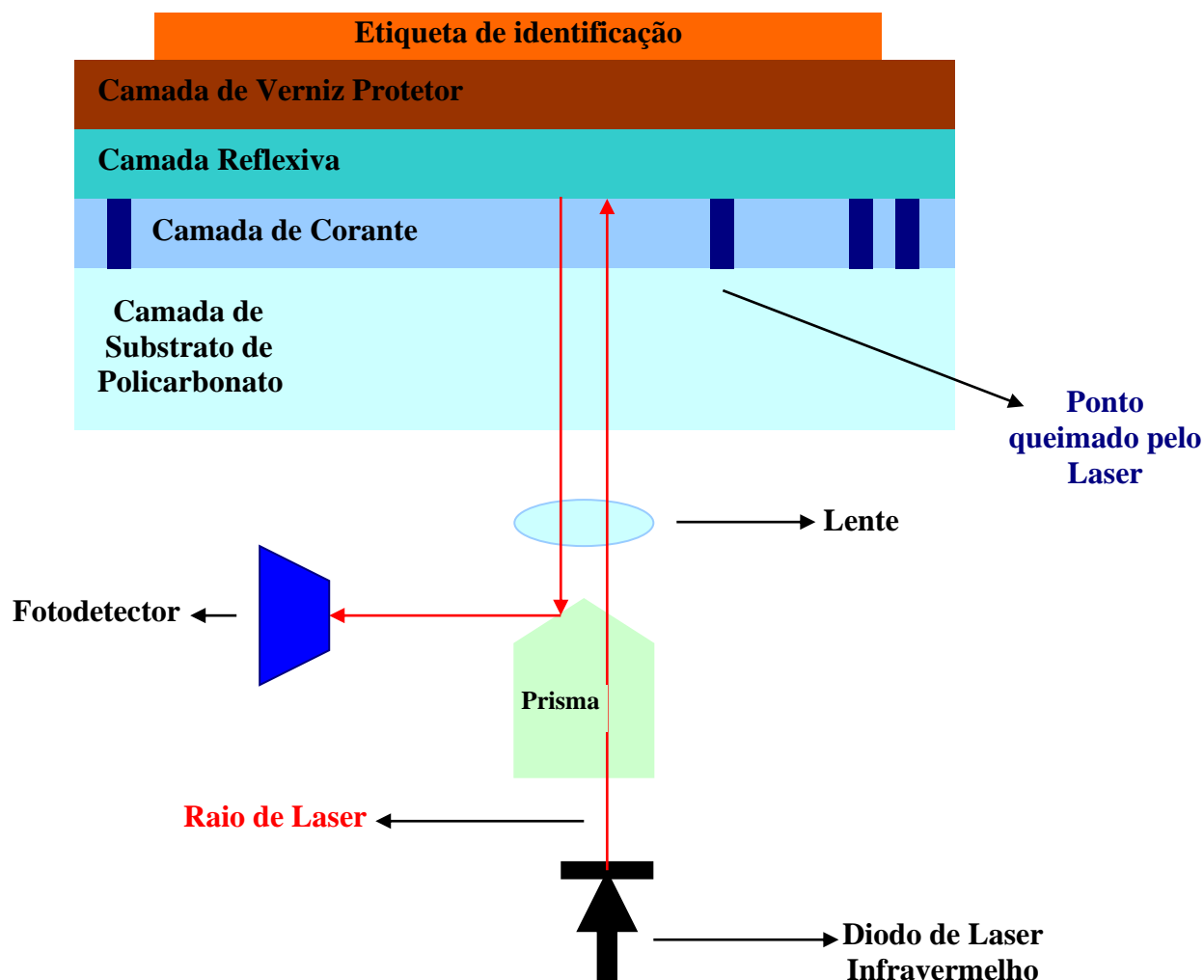
Um CDR tem as mesmas dimensões de um CD-ROM, 120mm de diâmetro e 1,2mm de espessura. Em sua espessura existem 4 camadas de materiais diferentes e necessários para viabilizar os processos de gravação e leitura de dados. As camadas são organizadas da seguinte maneira: camada de substrato de Policarbonato, camada de corante, camada reflexiva e camada de verniz protetor. Acima do verniz protetor pode ser colada uma etiqueta para identificar o CDR. A figura 2.4 mostra essa organização e o processo de escrita e leitura em um CDR.

A evolução natural dos CDRs foi o CD-RW (CD – ReWritable) que possui as mesmas dimensões dos CDRs. Para conseguir apagar o conteúdo de um disco regravável, a camada de corante mostrada na figura 2.4 foi trocada por uma camada contendo uma liga de prata, índio, antimônio e telúrio. Essa liga possui a seguinte característica, pode permanecer em dois estados considerados estáveis, o estado cristalino e o estado amorfo. Além disso, os *drivers* de CD-RW usam lasers de três potências diferentes, alta, média e baixa. Os lasers de alta e baixa potência são utilizados no processo de escrita. A alta potência muda o estado da liga de amorfo para cristalino e a potência média realiza a transição do estado amorfo para o estado cristalino, e dessa forma realiza o processo de escrita no CD-RW. A potência baixa é utilizada no processo de leitura.

Com a evolução tecnológica aliada a necessidade de adicionar características de multimídia aos softwares foi criado o DVD, originalmente significando Digital Vídeo Disk e evoluindo para Vídeo Versatile Disk, mantendo a sigla.

Os DVDs possuem as mesmas dimensões dos CDs, no entanto, a utilização de avanços tecnológicos dos mecanismos de leitura e escrita e de materiais utilizados na construção dos discos resultou em maior densidade de dados numa mesma área e uma velocidade de gravação e leitura muito maior. Atualmente os DVDs podem ser disponibilizados em 4 formatos diferentes. Os formatos foram consequência de duas características: a primeira muito simples, trata-se de construir discos com dupla face, basta construir dois discos e grudá-los através de uma camada adesiva; a segunda foi consequência de melhorias tecnológicas nos

processos de gravação leitura que permitiram a inclusão de duas camadas de dados em uma única face. A combinação de quantidade de faces com quantidade de camadas gerou discos com os 4 formatos distintos e com capacidade de armazenamento muito maior que as disponíveis nos formatos comerciais de CDRs. A Tabela 2.4 mostra as características desses formatos.

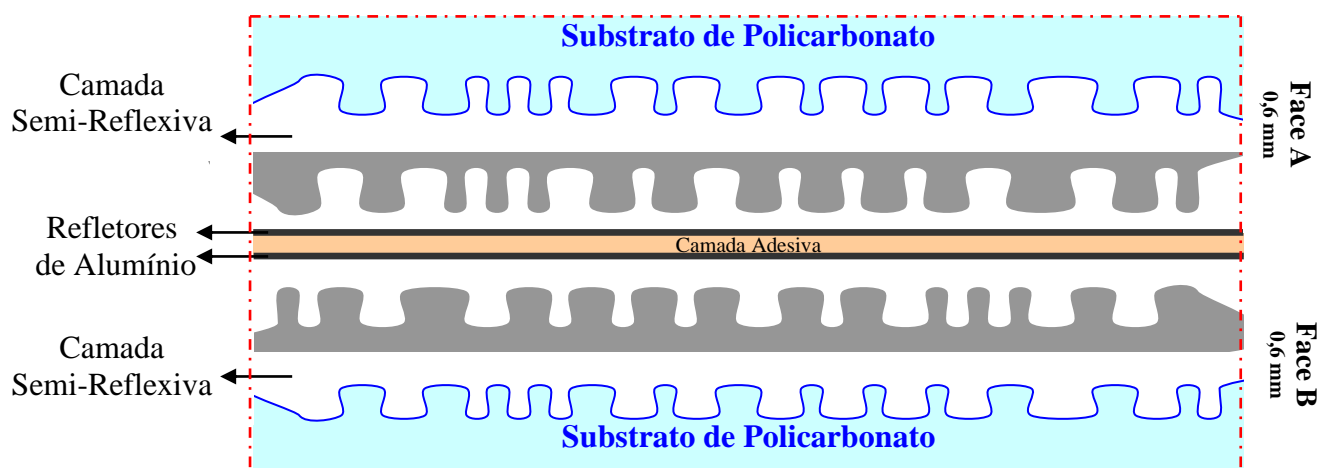


**Figura 2.4- Constituição e processo de leitura e gravação de um CDR**

Formatos de DVDs			
Formato	Faces	Camadas	Capacidade
1	Uma	Uma	4,7 GB
2	Uma	Duas	8,5 GB
3	Duas	Uma	9,4 GB
4	Duas	Duas	17 GB

**Tabela 2.4- Relação e características dos formatos comerciais de DVDs**

A figura 2.5 mostra a organização de um DVD com dupla face e dupla camada formada por dois discos de dupla camada com 0,6mm sobrepostos e unidos através de uma camada adesiva.



**Figura 2.5- Constituição de um DVD Dupla face Dupla camada**

## 2.2- Organização de Memória Semicondutora

Através da união de um conjunto de registradores, podem ser construídos sistemas com grande capacidade de armazenamento para conter qualquer tipo de informações, tais como: dados numéricos, textos, operações, som, imagem, animação, programas, entre outros. Os sistemas utilizados para armazenamento de grande volume de informações são denominados de subsistemas de memória.

A capacidade de armazenamento de bits em um subsistema (pente, chip) de memória semicondutora é informada através da forma como eles estão organizadas. Um subsistema de memória é constituído de um conjunto de palavras e cada uma dessas palavras é formada por um mesmo número de bits. O número de bits em uma palavra é definido como o comprimento da palavra.

A capacidade de armazenamento de palavras na memória principal de um computador é limitada pelo barramento de endereço e a quantidade de bits de cada palavra é definida pelo barramento de dados. Um subsistema de memória sempre é referenciado pela sua capacidade de armazenamento de bits que está diretamente relacionado a quantidade de palavras existentes no subsistema de memória e com o comprimento de cada palavra. A quantidade de palavras de uma memória e o comprimento de cada palavra define a forma de organização da mesma.

Para assimilar de forma mais consistente as funções de: sintetizar, dimensionar e organizar e implementar subsistemas de memórias semicondutoras é necessário dominar unidades as métricas utilizadas para quantificar, medir e ou projetar subsistemas de memórias semicondutoras. No entanto, essas unidades métricas são utilizadas para medir subsistemas de

memória implementados em qualquer tecnologia (magnética, ótica ou semicondutora). Convencionou-se o byte, um conjunto de oito dígitos binários (0 ou 1) com sendo unidade de medida de memória. A Tabela 2.5 mostra algumas unidades métricas utilizadas para denotar e medir capacidade de memória.

A Tabela 2.5 mostra algumas unidades métricas utilizadas para denotar e medir capacidade de memória.

Sigla	Valor	denominação
K	$2^{10} = 1.024$	Kilo
M	$2^{20} = 1.048.576$	Mega
G	$2^{30} = 1.073.741.824$	Giga
T	$2^{40} = 1.099.511.627.776$	Tera
byte	Um conjunto de oito bits	Byte

**Tabela 2.5- Unidades Métricas**

Por exemplo, uma memória contendo 2.147.483.648 bits ou 2 Gigabits, pode estar organizada de várias formas.

A forma clássica de medir ou informar a capacidade de um subsistema de memória é informar a sua quantidade de palavras seguido do comprimento da palavra.

No exemplo acima podemos ter os seguintes exemplos de organização:

- 256MB (Megabytes) ou 256Mx8bits, essa notação clássica mais utilizada de informar a capacidade de um subsistema de memória e que deve ser lida: 256 Mega palavras de oito bits;
- 128Mx16, 128 mega palavras de 16 bits ou 128 Mega por 16;
- 64Mx32, 64 Mega palavras de 32 bits ou 64 Mega por 32;
- 32Mx64, 32 Mega palavras de 64 bits ou 32 Mega por 64.

As unidades de memória principal dos computadores podem armazenar tanto programas quanto dados, como consequência, a Memória Principal geralmente é dividida em duas grandes áreas de armazenamento, área de programa e área de dados.

A área de programa, por sua vez, pode estar subdividida em 3 outras repartições: memória BIOS (não volátil); área do Sistema Operacional e área de programas de usuários.

### **2.2.1- Associação Série Paralelo de Componentes de Memória.**

A memória principal de um computador é constituída por um ou mais componentes (chips / pentes) de memória semicondutora, com células de armazenamento capacitivas. Ela é classificada como sendo de acesso aleatório, volátil e dinâmico. Quando o componente (chip /



circuito integrado) é composto por células constituídas por flip-flops o subsistema de memória e classificado como sendo de acesso aleatório, volátil e estático. Componentes com essas características são, geralmente, utilizados para implementar subsistemas de memória CACHE e registradores da CPU.

O processo de integração de células de memória, capacitivas ou flip flops, permite construir componentes físicos (circuitos integrados) constituídos por uma grande quantidade de registradores (palavras), cada um contendo o mesmo número de bits. Esses registradores possuem entrada paralela e saída paralela de dados.

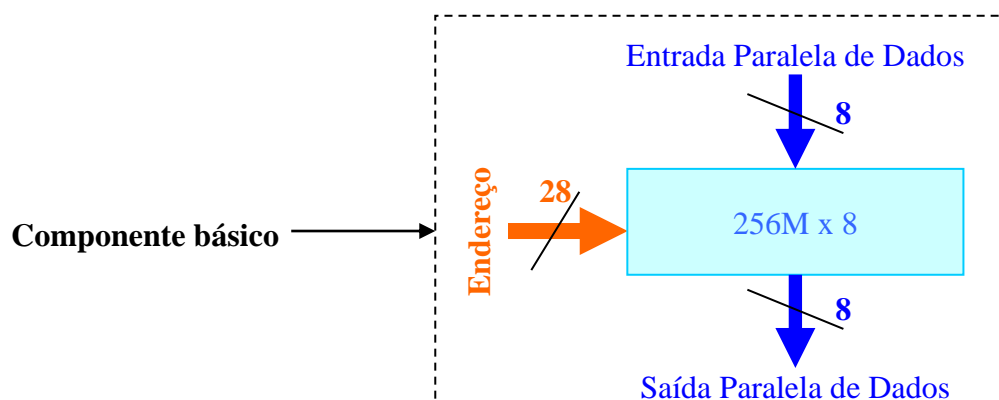
Com a disponibilidade de circuitos integrados com capacidade de armazenamento de  $M$  palavras de  $N$  bits é possível projetar subsistemas de memória com quantidade de palavras sendo múltiplo de  $M$  e comprimento de palavra sendo múltiplo de  $N$ .

Isso é possível através da associação em série ou em paralelo de circuitos integrados de memória. A associação em série aumenta a quantidade de palavras e mantém o comprimento das palavras. A associação em paralelo de componentes de memória mantém a quantidade de palavras e aumenta o comprimento da palavra (quantidade de bits das palavras).

A associação em paralelo de um componente de memória contendo “ $M$ ” palavras de “ $N$ ” bits possibilita a construção de subsistemas de memória com mesma quantidade de palavras, mas com palavras com comprimento múltiplo de “ $N$ ”. Aumenta o comprimento da palavra. Se for utilizado a associação de “ $K$ ” componentes em paralelo, será obtida uma memória de “ $M$ ” com palavras de comprimento “ $K*N$ ”.

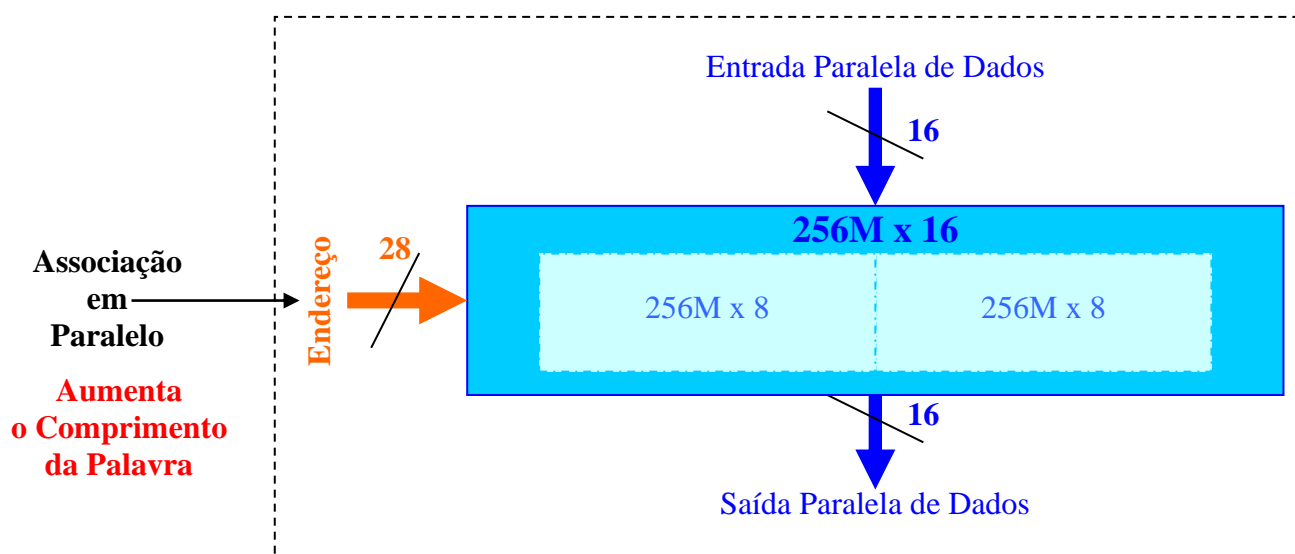
A associação em série de componentes de memória contendo “ $M$ ” palavras de “ $N$ ” bits em possibilita a construção de subsistemas de memória com quantidade de palavras sendo múltiplo de “ $M$ ” e com mesmo comprimento “ $N$ ” de palavra. Aumenta a quantidade de palavras. Se for utilizado a associação de “ $K$ ” componentes em série, será obtida uma memória com “ $K*M$ ” palavras de comprimento “ $N$ ”. A quantidade de bits de endereços aumenta de forma proporcional

A figura 2.6 mostra um componente básico de memória com capacidade de 256 Mega palavras de 8 bits cada, ou seja 256 MB (Megabytes)



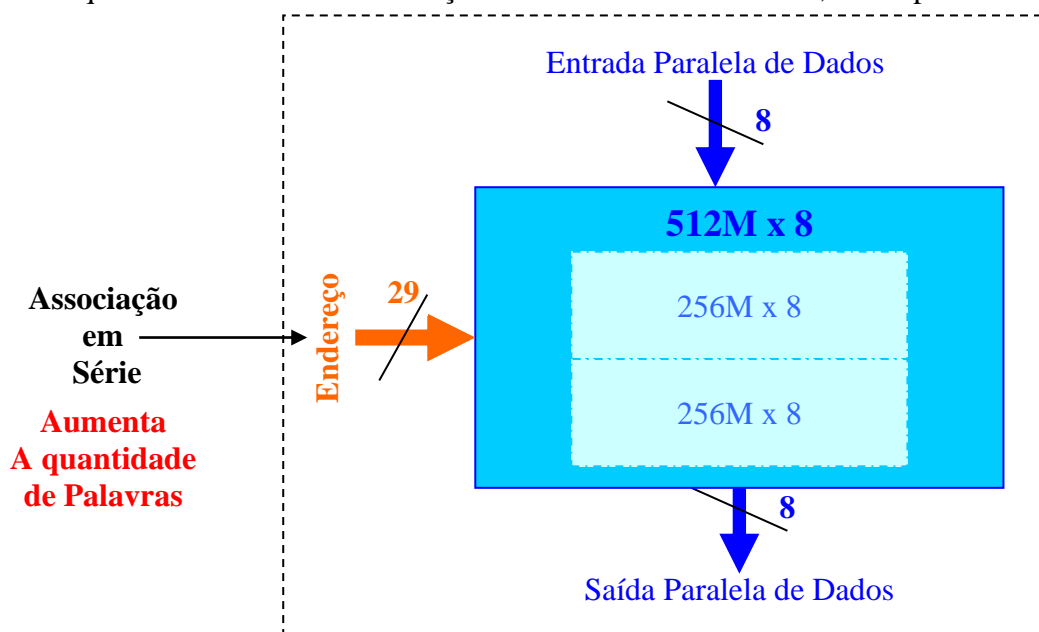
### Figura 2.6- Componente de Memória de 256 Megabytes

A figura 2.7, mostra a associação em paralelo de dois componentes de memória de **256M** palavras de 8 bits da figura 2.6, para construir um subsistema de memória de **256M** palavras de 16 bits. Ou seja, a associação em paralelo manteve a quantidade de palavras em **256 Mega** e dobrou o comprimento das palavras para **16** bits, obtendo um subsistema de memória de **256M x 16**.



**Figura 2.7- Associação em Paralelo componentes em paralelo**

A figura 2.8, mostra associação em série de dois componentes de memória de **256M** palavras de 8 bits da figura 2.6, para construir um subsistema de memória de **512M** palavras de 8 bits. Ou seja, a associação em série dobrou a quantidade de palavras em **256 Mega** e manteve o comprimento das palavras com 8 bits, obtendo um subsistema de memória de **512M x 8**. A quantidade de bits de endereços deve aumentar em um bit, de 28 para 29 bits.



### **Figura 2.8- Associação em Série de componentes de memória.**

Conforme mostram as figuras 2.8 e 2.9 através da associação de componentes de memória nas formas serial e paralelo possibilitam o projeto e implementação de subsistemas de memória de quaisquer dimensões ou capacidade. Esse fato permite a construção de subsistemas de memória que satisfaçam as necessidades de qualquer aplicação ou necessidades de uma arquitetura computacional.

Na figura 2.8 deve também ser verificado que a quantidade de bits de endereçamento aumentou de 28 para 29. Esse bit acrescido ao barramento de endereço é necessário para endereçar todas as palavras de memória existente no novo subsistema obtido através da associação série dos componentes. Como a figura mostra a associação de apenas dois componentes o bit acrescido ao barramento de endereço é utilizado para identificar um dos componentes de memória. Quando o bit acrescido assume o valor lógico “0” é endereçada uma palavra contida no primeiro componente de memória, ou seja, aquele que contém as primeiras 256 MB. Quando o bit acrescido assume o valor lógico “1” é endereçada uma palavra contida no segundo componente de memória, ou seja, aquele que contém as restantes 256 MB de palavras.

O bit acrescido é, portanto, utilizado por um circuito denominado decodificador de endereço. O decodificador de endereço é um circuito combinacional que possui como entrada o bit adicionado ao barramento de endereço e como saídas, duas linhas que selecionarão um dos componentes contidos na figura. Geralmente, para efeito didático e organizacional do subsistema de memória gerado, considera-se que o bit mais significativo do barramento de endereços como sendo o bit adicionado. A identificação das linhas (bits) contidas em um barramento de endereço é realizada através da letra “A” de “Address” (Endereço) seguidas de números que diferenciam cada uma das linhas. Onde o bit menos significativo do barramento é identificado pelo número zero (0) gerando o símbolo “A0”. Como na figura 2.8 mostra um barramento de endereço contendo 29 bits, o bit mais significativo é identificado pelo símbolo “A28” ( $28=29-1$ ).

Os projetos de subsistemas de memória podem exigir qualquer quantidade de componentes de memória associados em série e/ou paralelo. O item 2.2.2 que segue contém os conceitos e técnicas para obter qualquer subsistema de memória a através da técnica de

associação em série e/ou paralelo de componentes de memória. A tabela 2.6 e a figura 2.10 mostram a síntese e implementação de um decodificador de endereço genérico.

A tabela verdade que segue descreve essas características e permitem sintetizar o circuito decodificador de endereço. Na tabela o bit mais significativo do barramento de endereço (A28) é a entrada do circuito e as saídas denominadas: “Enable 1 e “Enable 0”, representadas respectivamente por: “E1” e “E0”, serão utilizadas para habilitar “escolher” apenas um dos componentes contidos na associação série de dois componentes de memória para realizar uma operação de escrita ou leitura no subsistema de memória obtido. O componente selecionado/habilitado pela saída “E0” é batizado de banco de memória zero (banco 0) e o componente selecionado/habilitado pela saída “E1” é batizado de banco de memória um (banco 1).

Entrada	Saídas		Comentário
A28	E1	E0	
0	0	1	Seleciona Banco 0
1	1	0	Seleciona Banco 1

### Decodificador de endereço de apenas um bit

As figuras: 2.12 e 2.13 mostram um exemplo de associação série e paralelo de um componente de memória real mostrado na figura 2.11.

## 2.2.2- Projeto de Um Subsistema de Memória

O projeto de uma memória deve considerar os seguintes parâmetros: a quantidade de palavras de memória; a quantidade de bits das palavras. Além disso, um sistema de memória deve contemplar os seguintes subsistemas: **Registrador de Entrada de Dados; Registrador de Saída de Dados Registrador de Endereço; Decodificador de Endereço, Multiplexadores de Saída de Dados e do Sistema de Controle** da memória, que é responsável por realizar as operações leitura e escrita na memória. A figura 2.9 mostra um subsistema de memória genérico. A organização mostrada na figura 2.9 pode ser utilizada tanto em memórias RAMs (dinâmicas e estáticas) quanto em memórias ROMs.

O **Registrador de Entrada Dados** é um registrador com entrada paralela e saída paralela de dados. Sua função é conter os dados que serão armazenados em uma determinada palavra durante uma operação de escrita na memória.

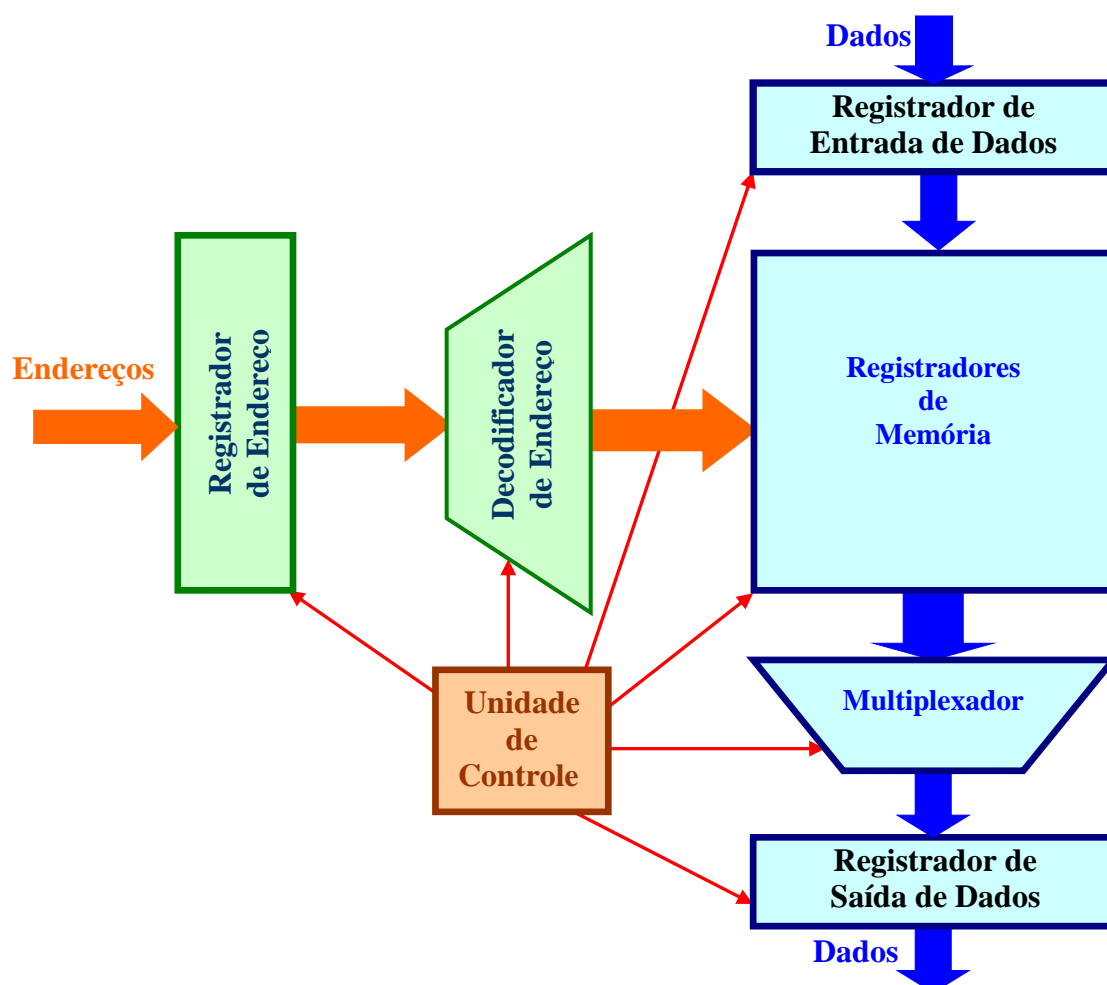
O **Registrador de Saída de Dados**, assim como o registrador de Entrada de dados, possui entrada e saída paralela de dados e pode ser implementado da mesma forma como a proposta de implementação de um registrador de uso geral. No entanto, sua função é a de

conter temporariamente o conteúdo de uma determinada palavra da memória, obtido através de uma operação de leitura na memória.

O **Multiplexador de Saída de Dados** é formado por um conjunto de elementos multiplexadores. Sua função é selecionar o dado correto que deve ser extraído da memória. Essa seleção é realizada através do conteúdo contido no Registrador de Endereço. Ou seja, utiliza o endereço da palavra (registrador) que está sendo lida da memória.

A função do **Registrador de Endereço** é conter o endereço físico da palavra que irá receber dados (operação de escrita) ou disponibilizar dados (operação de leitura). A quantidade de bits contida no registrador de endereço determina a capacidade máxima de palavras que deve conter o sistema de memória. Se o registrador de endereço possuir “ $n$ ” bits o sistema de memória pode conter no máximo “ $2^n$ ” palavras.

O **Decodificador de Endereço** é formado por um circuito combinacional que tem a capacidade de localizar e selecionar o registrador identificado pelo endereço contido no Registrador de Endereço da memória, para que seja realizada uma operação de leitura ou escrita na mesma.



**Figura 2.9- Organização Clássica de um Subsistema de Memória**

Um decodificador genérico é organizado da seguinte forma: possui “n” entradas de seleção, “2<sup>n</sup>” saídas e uma entrada de controle denominada Enable, utilizada para habilitar ou desabilitar o funcionamento normal do decodificador. Quando as entradas de seleção são endereços de memória, o decodificador para a ter a função de decodificar o endereço. Daí a denominação Decodificador de Endereço.

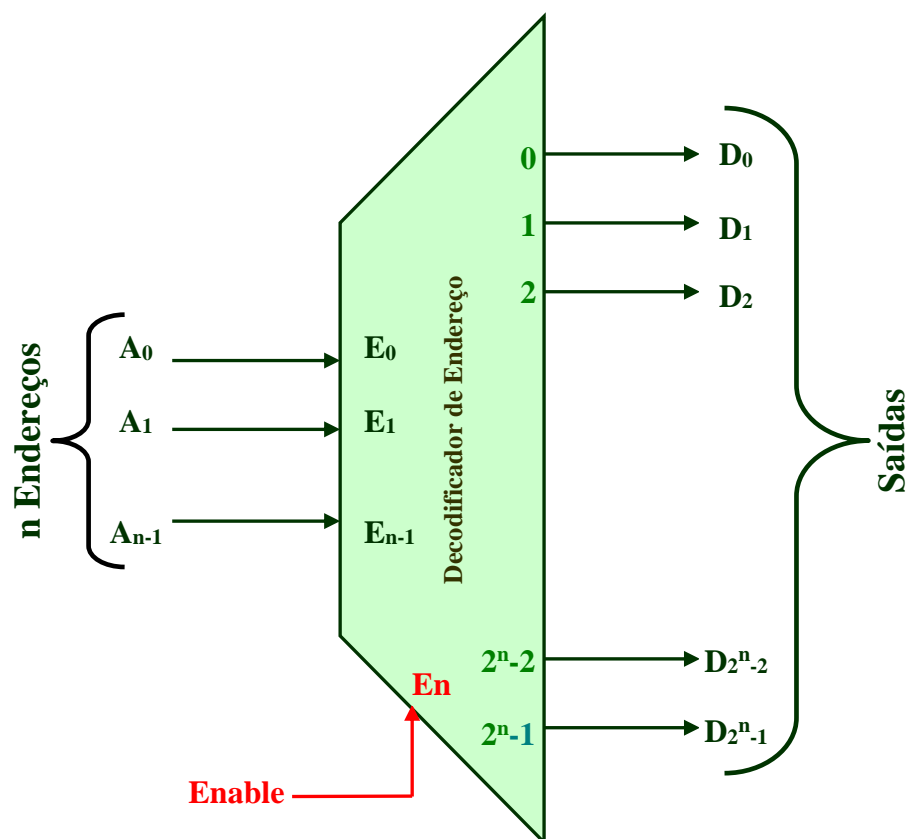
Um decodificador de endereço genérico, contendo “n” entradas de seleção, funciona logicamente da seguinte forma: insere o valor lógico “1” (High), em uma única saída, dentre as “2<sup>n</sup>” existentes, de acordo com o valor do endereço contido em suas entradas de seleção. Todas as demais “2<sup>n</sup>-1” saídas permanecem com o nível lógico “0” (Low). Em um decodificador de endereço, as entradas de seleção são endereços de memória contidos no Registrador de Endereço.

O funcionamento de um decodificador de endereço genérico, contendo “n” entradas de seleção, “2<sup>n</sup>” saídas e uma de controle de “Enable” está detalhado na Tabela 2.6. Quando a entrada Enable assumir o valor lógico “1” (High) todas as “2<sup>n</sup>” saídas permanecem no valor lógico “0”. Na Tabela 2.6, as entradas de seleção são identificadas pelas letras “A” originária da palavra “Adress” (Endereço). As saídas estão identificadas pela letra D.

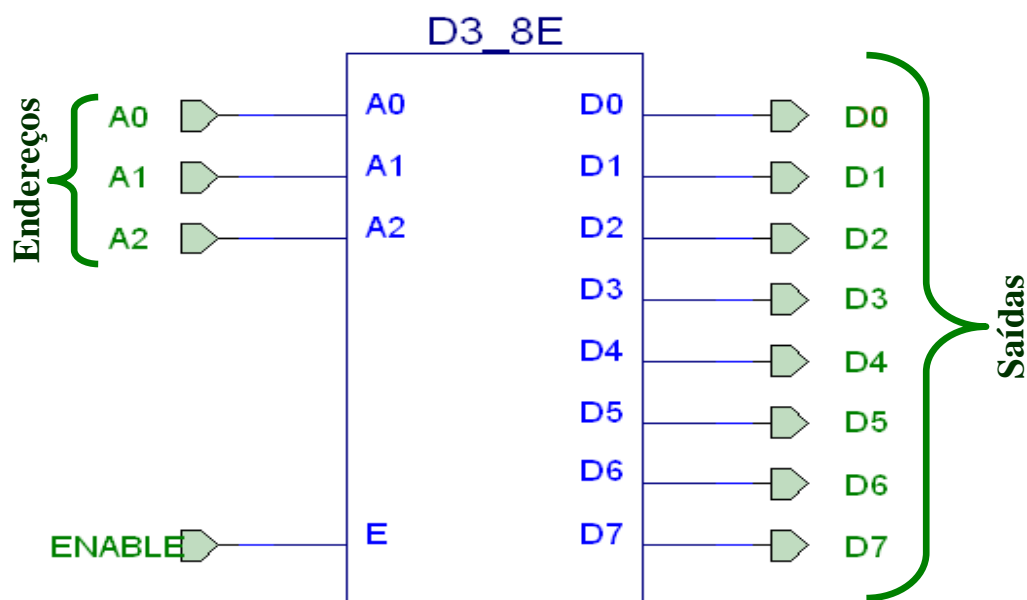
Decodificador com “n” entradas de Endereço													
Entradas de Endereço						Entrada de Controle	Saídas						
Valor	Valores binários												
Decimal	A <sub>n-1</sub>	A <sub>n-2</sub>	---	A <sub>1</sub>	A <sub>0</sub>	Enable	D <sub>2<sup>n</sup>-1</sub>	D <sub>2<sup>n</sup>-2</sub>	---	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	---	0	0	1	0	0	---	0	0	0	1
1	0	0	---	0	1	1	0	0	---	0	0	1	0
2	0	0	---	1	0	1	0	0	---	0	1	0	0
3	0	0	---	1	1	1	0	0	---	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---
2 <sup>n</sup> -2	1	1	---	1	0	1	0	1	---	0	0	0	0
2 <sup>n</sup> -1	1	1	---	1	1	1	1	0	---	0	0	0	0
X	X	X	X	X	X	0	0	0	---	0	0	0	0

**Tabela 2.6- Tabela Verdade de um Decodificador de Endereço Genérico**

A figura 2.10a mostra o diagrama de blocos de um Decodificador de Endereço Genérico e a Figura 2.10b mostra a implementação de um Decodificador de Endereço contendo três bits de seleção de endereço e oito saídas, implementado no software Project Manager, utilizando o elemento “D3x8”.



(a) diagrama de blocos de um decodificador de endereços genérico.



(b) Uma Implementação do Registrador Acumulador no software Project Manager

### Figura 2.10- Decodificador de Endereço

Com a descrição e detalhamento do funcionamento de um decodificador de endereço, estão identificados todos os elementos necessários para a implementação de um subsistema de

memória. Para iniciar a implementação de um subsistema de memória real, temos que unir todos esses elementos e controlá-los de forma apropriada a forma que eles irão interagir e trocar informações (dados, endereços e códigos de operações).

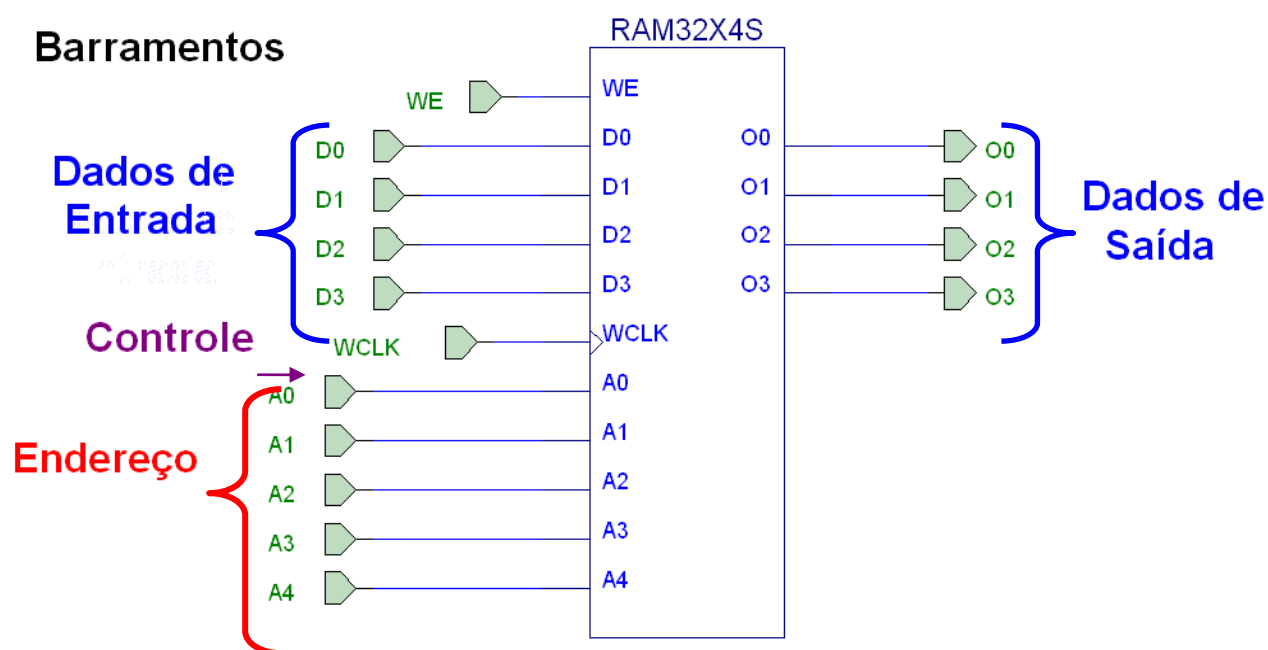
A tarefa de realizar o controle de todo o subsistema será realizada pelo elemento denominado Unidade de Controle. A Unidade de Controle, têm a função de controlar a sequência de microordens necessárias para a implementação de uma operação de leitura e/ou escrita na memória.

A figura 2.9, mostra em diagrama de blocos, um subsistema de memória genérico contendo seus elementos básicos e os barramentos de dados e de endereço, que servem para interligá-los. As linhas de controle, oriundas da Unidade de Controle, estabelece os caminhos e os instantes adequados para permitir que as informações fluam entre os elementos e através dos barramentos. A Unidade de Controle está localizada fisicamente na Unidade central de Processamento.

Para implementar o subsistema de memória real foi utilizado o software Project Manager. E dentre os elementos de memória presentes na biblioteca de componentes do software será utilizado, no exemplo real, o elemento “RAM32X4S” da figura 2.11. Ele contém 32 palavras de 4 bits e será utilizado para projetar e implementar um subsistema de memória volátil, aleatória e estática; contendo 128 palavras de 8 bits. Dessa forma serão também consolidadas as técnicas de associação série/paralelo de componentes de memórias apresentadas no item 2.2.1.

Um circuito elemento de memória é um circuito sequencial, portanto, não basta estar projetado corretamente para funcionar de forma adequada. Ele também precisa ser operado corretamente para gerar os resultados esperados. Assim, os valores das linhas de controle e a ordem em que são disponibilizados são essenciais para que sejam realizadas operações de escrita e leitura na memória. A tarefa de disponibilizar esses valores nos instantes adequados é realizada pela Unidade de Controle.





**Figura 2.11- Elemento de Memória “RAM32X4S” do Project Manager**

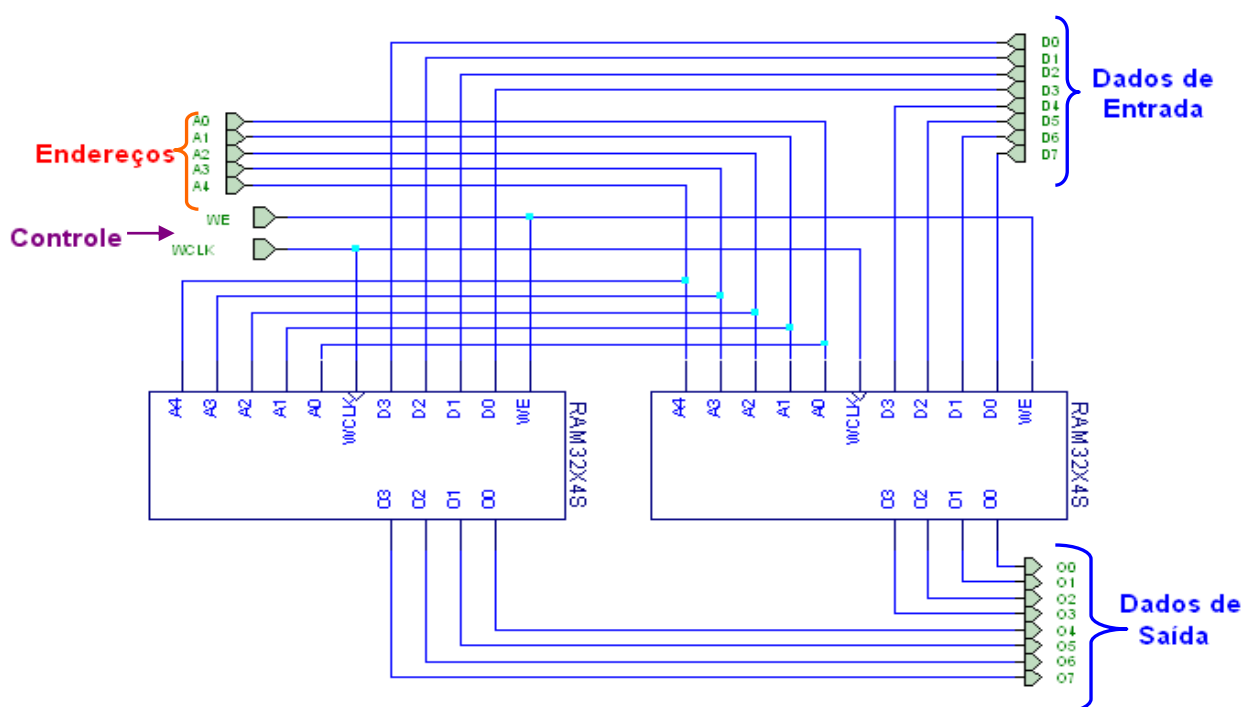
A tabela 2.7, mostra a forma de funcionamento do componente “RAM32X4S”. A entrada de controle WE (Write Enable) define qual operação é realizada no componente, leitura ou escrita. (WE=0 leitura, WE=1 escrita). Para realizar uma operação de leitura no componente basta fazer WE=0 e disponibilizar o endereço e o dado é amostrado nas saídas de dados (Barramento de Saída de Dados). Para realizar um escrita no componente é necessário fazer WE=1, disponibilizar o valor a ser escrito nas entradas de Dados (barramento de Entrada de Dados), disponibilizar o endereço onde o dado será armazenado (Barramento de Endereço) e acionar a entrada WCLK (clock de escrita) com uma borda de subida.

Endereço	Dado	Controles		Comentário
		WE	WCLK	
A <sub>5</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	0	X	Leitura do dado contido no endereço A <sub>5</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub> (barramento de endereço)
A <sub>5</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	1	X	
A <sub>5</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	1	↑	Escrita no dado contido nas entradas D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub> (barramento de Dados) no endereço contido nas entradas A <sub>5</sub> A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub> (barramento de endereço)

**Tabela 2.7- Tabela verdade do componente “RAM32X4S”**

A figura 2.12 mostra dois componentes de memória “RAM32X4S” associado em paralelo. Observe que a quantidade de palavras foi mantida e, conseqüentemente, o barramento de

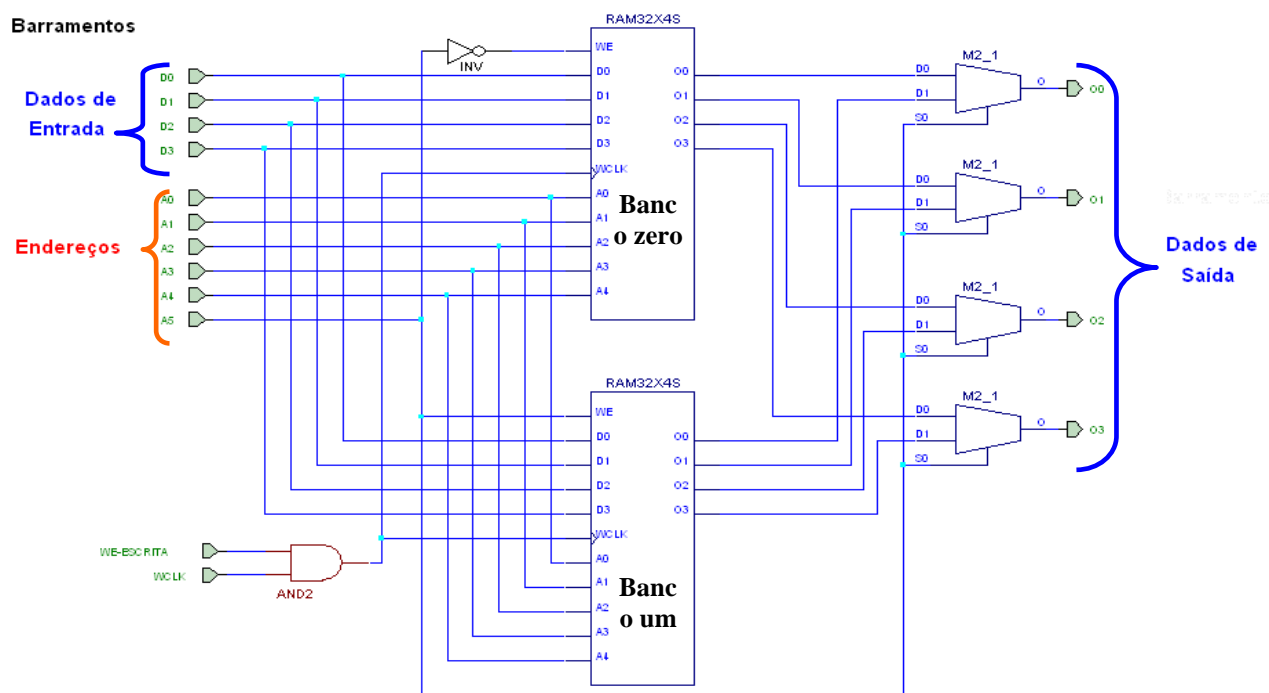
endereços permanece com 5 bits, necessários e suficientes para endereçar todas as 32 palavras da memória. O comprimento da palavra dobrou para 8 bits, formando uma memória 32x8.



**Figura 2.12- Memória 32x8 - Associação em Paralelo de duas RAM32X4S**

A figura 2.13 mostra dois componentes de memória da figura 2.11, associado em série, formando uma memória com 64 palavras de 4 bits (64x4). Dobrando a quantidade de palavras e mantendo a quantidade de bits das palavras do elemento “RAM32X4S”. Verifique que aumentou apenas um bit no barramento de endereços. E que foi inserido um conjunto de 4 elementos multiplex “M2\_1”, um para cada bit da palavra. As entradas de seleção do elemento multiplex são realizadas pelo bit mais significativo do barramento de endereço e é responsável por disponibilizar o dado correto na saída de dados, escolhendo o entre o dado proveniente do banco “zero” da memória ou o dado proveniente do banco “um” da memória.

A Figura 2.14 mostra o projeto de uma memória de acesso aleatório, volátil e estática contendo 128 palavras de 8 bits, obtida através da associação em série e paralelo do elemento “RAM32X4S”. Para endereçar as 128 palavras são necessários 7 bits de endereços ( $7 = \log_2 128$ ). Note que a memória 128x8 foi montada através da associação em série de 4 (quatro) bancos de memória 32x8 da figura 2.11.



**Figura 2.13- Memória 64x4 - Associação em Série de duas “RAM32X4S”**

Para selecionar um entre os 4 bancos da memória 128x8 são necessários os dois bits mais significativos do barramento de endereço, pois  $2^2 = 4$ . Assim, serão utilizadas as linhas “ADREES-6” e “ADRESS-5”. Essas linhas de endereços também serão utilizadas no controle das operações de leitura, pois realizam a seleção dos 8 elementos Multiplex “M4\_1E”.

O circuito da figura 2.14 é um circuito seqüencial, então, não basta que o circuito esteja funcionando corretamente para obter-se um resultado esperado quando ele ser utilizado. Para isso, é necessário operá-lo controlá-lo de forma adequada. Num circuito seqüencial, a ordem de execução de atividades e geração das microordens é essencial para que o circuito opere corretamente e gere as respostas esperadas.

Para realizar de forma correta as operações de escrita (write) de leitura (Read) disponíveis no subsistema de memória da figura 2.14 é necessário utilizar corretamente todos os elementos utilizados na sua implementação que são: Registrador de Entrada de Dados; Registrador de Saída de Dados; Registrador de Endereços; Decodificador de Endereços e os elementos Multiplex responsáveis pela seleção de dados de saída.



A seqüência de microordens e atividades que deve ser respeitada para realizar uma operação de **Escrita** (Write) na Memória da figura 2.14 é mostrada na Tabela 2.8

<b>Seqüência de Controle da Operação de Escrita na Memória (Write)</b>		
<b>Passo</b>	<b>Comentário / Atividade</b>	<b>Microordem</b>
1	<ul style="list-style-type: none"> <li>- Disponibilizar Endereço</li> <li>- Ativar Enable do Registrador de Endereço;</li> <li>- Ativar LOAD do Registrador de Endereço;</li> <li>- Disponibilizar Dado</li> <li>- Ativar Enable do Registrador de Entrada de Dados</li> </ul>	<ul style="list-style-type: none"> <li>- E_R_END = 1;</li> <li>- L_R_END= 1;</li> <li>- E_R_D_I = 1;</li> </ul>
2	- Gerar/ aguardar Borda de subida no relógio (clock)	- Clock = ↑
3	<ul style="list-style-type: none"> <li>- Ativar Enable da Memória;</li> <li>- Selecionar operação de escrita</li> </ul>	<ul style="list-style-type: none"> <li>- E_MEM = 1;</li> <li>- WE = 1;</li> </ul>
4	-Gerar/ aguardar Borda de subida no relógio (clock)	- Clock = ↑

**Tabela 2.8- Operação de Escrita na Memória – Seqüência de Atividades**

A proposta de seqüência de microordens apresentada na Tabela 2.8, mostra que são necessários quatro (4) passos para realizar operação de **Escrita** (Write) na Memória da figura 2.14.

A seqüência de microordens e atividades que deve ser respeitada para realizar uma operação de **Leitura** (Read) na Memória da figura 2.14 é mostrada na Tabela 2.9.

A proposta de seqüência de microordens apresentada na Tabela 2.9, mostra que são necessários quatro (4) passos para realizar operação de **Leitura** na Memória da figura 2.14. A mesma quantidade de passos necessária para realizar a operação de Escrita. Essa informação será útil na implementação da Unidade de Controle.

<b>Seqüência de Controle da Operação de Leitura na Memória (Read)</b>		
<b>Passo</b>	<b>Comentário / Atividade</b>	<b>Microordem</b>
1	<ul style="list-style-type: none"> <li>- Disponibilizar Endereço;</li> <li>- Ativar Enable do Registrador de Endereço;</li> <li>- Ativar LOAD do Registrador de Endereço;</li> </ul>	<ul style="list-style-type: none"> <li>- E_R_END = 1;</li> <li>- L_R_END= 1;</li> </ul>
2	- Aguardar Borda de subida no relógio (clock);	- Clock = ↑;
3	<ul style="list-style-type: none"> <li>- Ativar Enable da Memória;</li> <li>- Selecionar operação de Leitura;</li> <li>- Ativar Enable do Reg. de Saída de Dados da Mem.;</li> </ul>	<ul style="list-style-type: none"> <li>- E_MEM = 1;</li> <li>- WE = 0;</li> <li>- E_R_D_O = 1;</li> </ul>
4	- Aguardar Borda de subida no relógio (clock)	- Clock = ↑;

**Tabela 2.9- Operação de Leitura na Memória – Seqüência de Atividades**

O Exemplo de Aplicação 1 que segue será utilizado para consolidar os conceitos inerentes a síntese, dimensionamento, organização e projeto de subsistemas de memória. Os principais

conceitos e técnicas utilizadas em projeto de subsistemas de memória são: dimensionamento dos barramentos de endereço e dados; associação em série e paralelo dos componentes de memória disponíveis para obter o subsistema de memória desejado. Organizar as associações em série e paralelo para otimizar tanto o tempo de projeto quanto o tempo de acesso a informação do subsistema de memória projetado. A forma ou organização das associações em série e paralelo dos componentes de memória influem nos bancos físicos de memória do subsistema final influenciando diretamente na quantidade e mapeamento dos bancos físicos de memória.

### **2.2.2.1- Exemplo de aplicação –**

#### **Exemplo de aplicação –**

#### **Síntese e projeto de um subsistema de memória de 2 Giga palavras de 16 bits, utilizando componentes de 512M Bytes.**

Projetar um subsistema de memória de acesso aleatório contendo dois Giga-palavras de 16 bits, utilizando componentes de memória de 512 Mega-bytes. Descrever e mostrar o mapa da memória considerando que um banco de memória contém 512 Mega-palavras. Detalhar o barramento de endereços de acordo com os bancos de memórias. Dimensionar o barramento de dados.

#### **Resolução:**

A síntese e projeto de subsistemas de memória está diretamente relacionado aos seguintes conceitos ou técnicas: medidas de memória, dimensionamento de barramento de dados, dimensionamento do barramento de endereços, associação de componentes de memória em paralelo e associação de componentes de memória em série.

O objetivo do exercício é projetar um subsistema de memória de 2 Giga palavras com comprimento de 16 bits (2Gx16), utilizando componentes de memória de 512 Mega palavras de comprimento 8 (512Mx8 ou 512 Mega Bytes).

Para projetar o subsistema de memória solicitado é necessário identificar e dimensionar os seguintes fatores:

- A quantidade de componentes de memória (512Mx8) que devem ser utilizados para obter as dimensões da memória solicitada;
- Dimensionar o barramento de dados, esse dimensionamento determina a quantidade de componentes de memória (512Mx8) que devem ser associados em paralelo para

obter palavras de comprimento 16. Gerando como consequência um banco físico de memória com o comprimento de palavra desejado (512Mx16);

- Dimensionar o barramento de endereços, esse dimensionamento define a quantidade de bancos físicos de memória (512x16) que devem ser associados em série para obter a quantidade de palavras solicitadas (2 Giga);
- Sintetizar o decodificador de endereços para endereçar todos os bancos físicos de memória obtidos.

#### **2.2.2.1.1- Características do componente de memória de 512 M x 8.**

A figura 2.15 mostra as características básicas de um componente de memória de 512Mx8. O seu barramento de dados é composto por 8 bits e o seu barramento de endereços por 29 bits. O texto que segue descreve as principais características do componente que também estão contidos na figura 1.

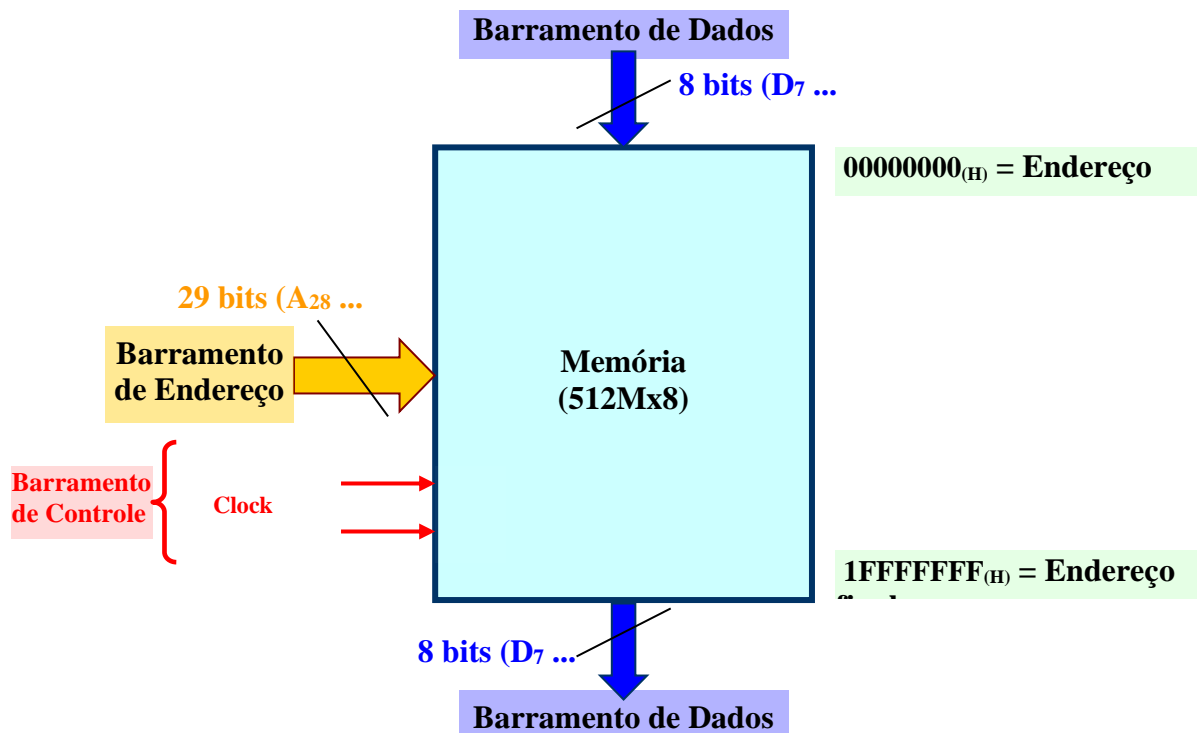
O barramento de dados do componente disponibilizado é de 8 bits pois o barramento de dados é definido pelo comprimento da palavra do componente de memória.

O barramento de endereços do componente é composto por 29 bits, pois o comprimento do barramento de endereços está diretamente relacionado à quantidade (número) de palavras (registradores) contidas (existentes) no componente de memória e obedece a seguinte relação:  $2^x = \text{quantidade de palavras}$ , onde x representa ao comprimento do barramento de endereços. Como a quantidade de palavras é de 512 Mega e  $512 = 2^9$  e  $1\text{Mega} = 2^{20}$ , então:  $2^x = 2^9 * 2^{20} = 2^{29}$ , portanto  $x = 29$ . Ou seja, o barramento de endereços que o componente de memória disponibilizado possui é formado por 29 bits e os endereços das palavras (registradores) é expresso na base Hexadecimal. Para tanto é necessário transformar a representação binária em bits para a base Hexadecimal, da seguinte forma:

Posição zero = **0000000000000000000000000000**<sub>(2)</sub> = **0000000**<sub>(H)</sub>

Última posição = **1111111111111111111111111111**<sub>(2)</sub> = **1FFFFFF**<sub>(H)</sub>

A quantidade de componentes de memória que deve ser utilizada é obtida através da determinação da quantidade de componentes de memória que dever ser associado em paralelo para gerar o comprimento de palavra solicitado multiplicado pela quantidade de bancos físicos de memória necessários para obter 2 Mega palavras. Para gerar uma palavra de 16 bits são necessários dois componentes de memória ( $16 \text{ DIV } 8 = 2$ ) e para gerar 2 Mega palavras de 16 bits é preciso associar em série 4 bancos de memória ( $4 * 512 \text{ M} = 2048 \text{ M} = 2 * 1024 \text{ M} = 2 \text{ G}$ ). Portanto são necessários 8 componentes de memória de 512Mx8 para projetar um subsistema de memória de 2Gx16.



**Figura 2.15- Características principais de um componente de memória de 512Mx8**

#### **2.2.2.1.2- Determinação da quantidade de componentes:**

A quantidade de componentes de memória que deve ser utilizada é obtida através da determinação da quantidade de componentes de memória que dever ser associado em paralelo para gerar o comprimento de palavra solicitado multiplicado pela quantidade de bancos físicos de memória necessários para obter 2 Mega palavras. Para gerar uma palavra de 16 bits são necessários dois componentes de memória ( $16 \text{ DIV } 8 = 2$ ) e para gerar 2 Mega palavras de 16 bits é preciso associar em série 4 bancos de memória ( $4 * 512 \text{ M} = 2048 \text{ M} = 2 * 1024 \text{ M} = 2 \text{ G}$ ). Portanto são necessários 8 componentes de memória de 512Mx8 para projetar um subsistema de memória de 2Gx16.

#### **2.2.2.1.3- Dimensionamento do barramento de Dados:**

- O componente de memória disponibilizado possui palavras (registradores) de 8 bits e para obter palavras de comprimento 16 (palavras de 16 bits) devem ser utilizados dois componentes de memória de 512 Megabytes (512Mx8) associados em paralelo. A associação em paralelo de componentes de memória aumenta o comprimento da palavra. Para cada componente de memória de 512Mx8 associado em paralelo aumenta-se o barramento de dados em 8 bits. A figura 2.16 mostra o resultado da associação de dois componentes de memória de 512Mx8 em paralelo. Verifica-se que a memória obtida na figura 2, a única característica modificada foi o comprimento do barramento de dados que agora possui 16 bits.



Para obter um subsistema de memória de 2Gx16 é necessário associar 4 componentes de memória de 512Mx16 em série. No item anterior o componente de memória contendo 512Mx16 foi sintetizado e projetado. Assim vamos utilizá-lo para sintetizar e projetar o subsistema de memória solicitado 2Gx16.

Na associação em série o comprimento da palavra não é alterado. Nessa forma de associação de componentes é modificada a quantidade de palavras. Para cada componente associado em série a memória resultante recebe a mesma quantidade de palavras contida no componente associado. Nesse caso, para cada componente de 512Mx16 associado em série são anexados no projeto da memória mais 512 Mega palavras. Esse fato gera como consequência o aumento de bits no barramento de endereço. Pois com 4 componentes de 512Mx16 associado em série é obtido um subsistema de memória de 2 Giga palavras e para endereçar 2 Giga palavras é exigido um barramento de endereço com 31 bits. Pois  $1\text{G} = 2^{30}$ , portanto  $2\text{G} = 2 * 2^{30} = 2^{31}$ . Os 31 bits do barramento de endereços serão representados pela letra A (Adress) seguido de um índice para indicar a sua posição:  $A_{30}, A_{29}, \dots, A_1, A_0$ .

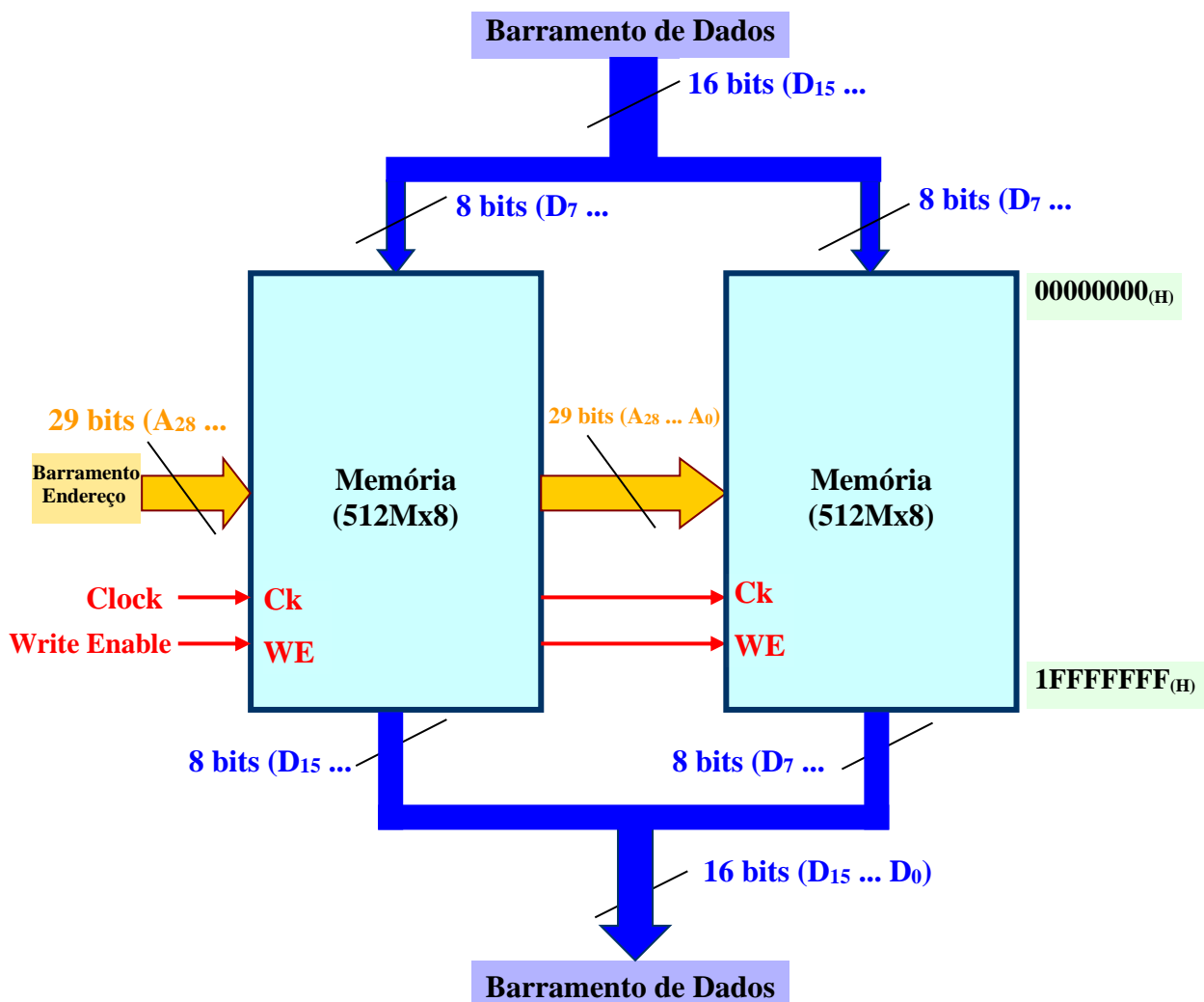


Figura 2.16- Resultado da associação em paralelo de componentes de memória 512x8

Última posição = **11111111111111111111111111111111**<sub>(2)</sub> = **7FFFFFFF**<sub>(H)</sub> =  $(2^{31} - 1)_{(10)}$

No entanto, o subsistema de memória é formado por 4 bancos que geram os respectivos bancos físicos de memória. Cada banco físico ou componente de memória é identificado pelos quatro valores binários que os bits de endereços  $A_{30}$  e  $A_{29}$  podem assumir:  $00_{(2)}=0_{(F)}$ ,  $01_{(2)}=1_{(F)}$ ,  $10_{(2)}=2_{(F)}$  e  $11_{(2)}=3_{(F)}$ , respectivamente.

Assim a mapa de endereços do subsistema de memória de  $2G \times 16$  assume a seguinte configuração:

#### Banco 0

Endereço Inicial =  $00000000000000000000000000000000_{(2)} = 00000000_{(H)}$   
 Endereço Final =  $00111111111111111111111111111111_{(2)} = 1FFFFFFF_{(H)}$

#### Banco 1

Endereço Inicial =  $01000000000000000000000000000000_{(2)} = 20000000_{(H)}$   
 Endereço Final =  $01111111111111111111111111111111_{(2)} = 3FFFFFFF_{(H)}$

#### Banco 2

Endereço Inicial =  $10000000000000000000000000000000_{(2)} = 40000000_{(H)}$   
 Endereço Final =  $10111111111111111111111111111111_{(2)} = 5FFFFFFF_{(H)}$

#### Banco 3

Endereço Inicial =  $11000000000000000000000000000000_{(2)} = 60000000_{(H)}$   
 Endereço Final =  $11111111111111111111111111111111_{(2)} = 7FFFFFFF_{(H)}$

Na operação de escrita (Write) as linhas de endereço  $A_{30}$  e  $A_{29}$  atuam no decodificador de endereço e conseqüentemente nas linhas de controle denominadas Write Enable (WE) dos componentes de memória para determinar o banco de memória onde está localizada a palavra na qual será inserida uma informação (dado). Na operação de leitura (Read), as linhas de endereços  $A_{30}$  e  $A_{29}$  atuam no elemento multiplex para determinar de qual banco de memória será selecionada a informação contida na palavra determinada pelo endereço. A figura 2.18 mostra o diagrama de blocos e a tabela verdade de um multiplex de duas entradas de seleção, quatro entradas de dados e uma saída de dados. Esse elemento multiplex é geralmente denominado Multiplex 4 para 1 ( $M4 \times 1$ ).

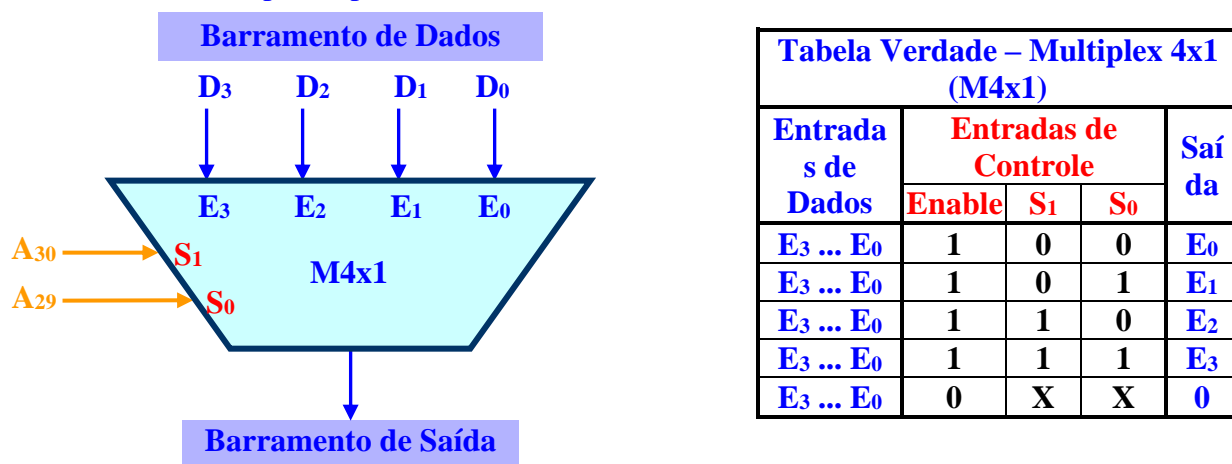


Figura 2.18- Diagrama de Blocos e Tabela Verdade de um Elemento Multiplex 4x1

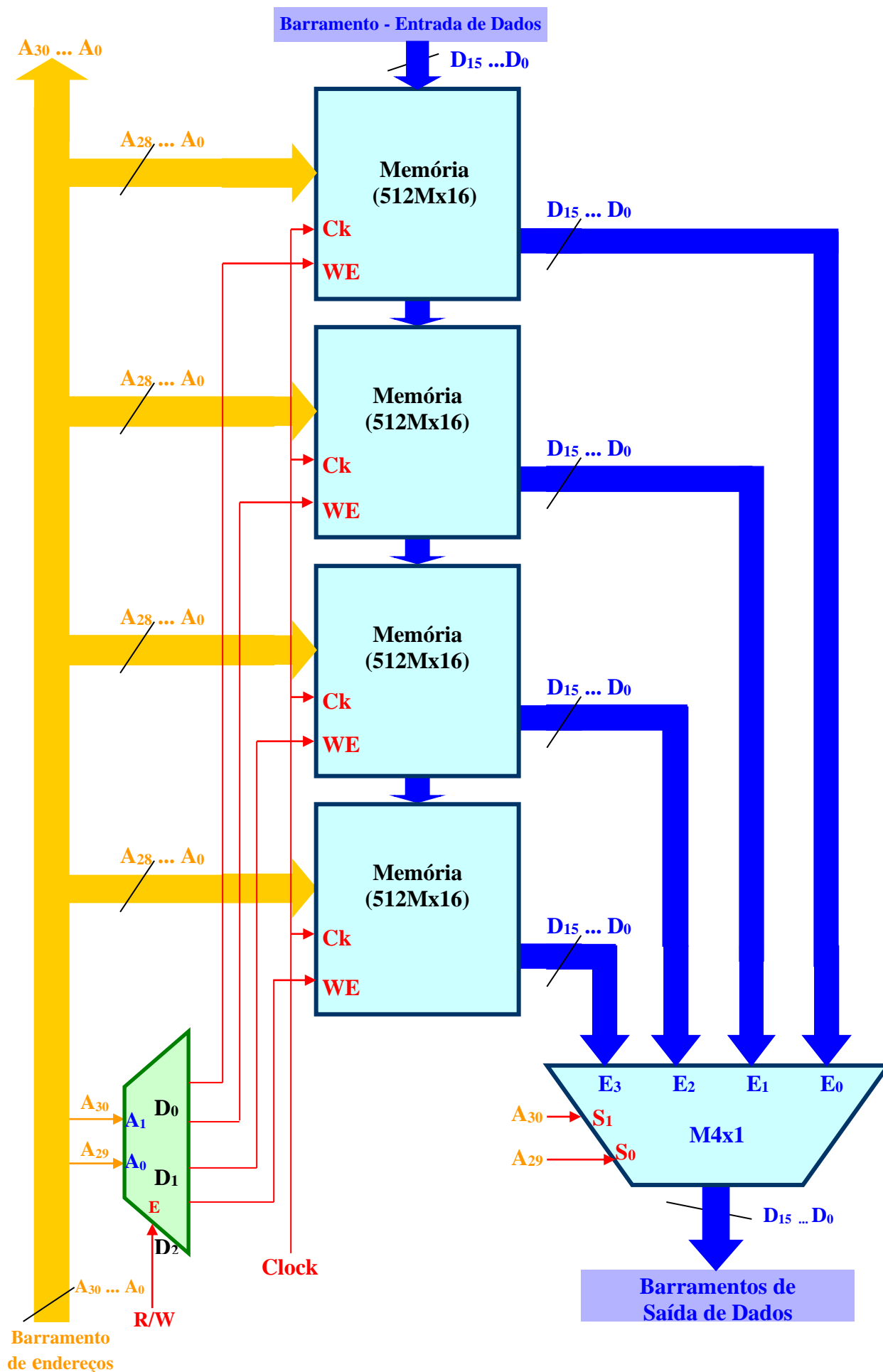


Figura 2.19 – Subsistema de Memória 2Gx16

A figura 2.19, mostra, em diagrama de blocos, o subsistema de memória de 2 Giga palavras de 16 bits. Verifique que o decodificador de endereços está selecionando, através de suas saídas, um único componente de memória dentre os quatro interligados em série, e esse componente é determinado pelo valor dos dois bits mais significativos do barramento de endereço ( $A_{30}$  e  $A_{29}$ ). Assim, em uma operação de escrita ou leitura, um único banco de memória é acessado no subsistema todo. O endereço da palavra dentro de banco de memória é determinado pelos 29 bits menos significativos ( $A_{28} \dots A_0$ ) do barramento de endereços.

A operação de escrita em um subsistema de memória deve obedecer a seguinte sequência:

- Disponibilizar endereço e dado;
- Definir operação de escrita;
- Gerar borda de subida na entrada de controle de Clock.

A operação de leitura em um subsistema de memória necessita da seguinte sequência:

- Disponibilizar o endereço da palavra desejada que o subsistema coloca em sua saída o conteúdo armazenado no endereço definido.

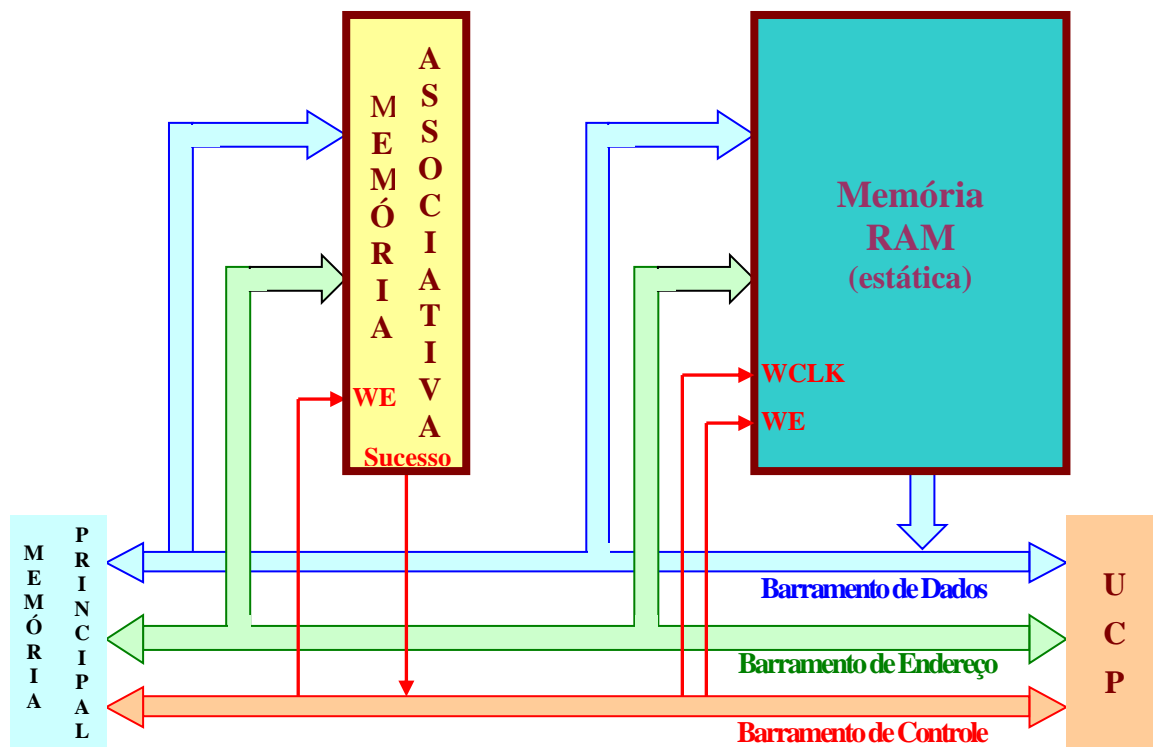
## 2.3- Subsistema de Memória Cache

O objetivo da memória CACHE aumentar o desempenho das máquinas através da disponibilização de um sistema de memória de alta velocidade de acesso.

A figura 2.20, mostra em diagrama de blocos os componentes existentes em um subsistema de memória CACHE. Ele é composto basicamente por uma memória estática com palavras de mesmo comprimento da Memória Principal (memória dinâmica) e uma memória associativa.

A figura 2.21 mostra a localização e forma de desempenho do subsistema de memória CACHE. Ela está localizada entre a CPU e a memória principal de uma máquina. A CPU, durante o ciclo de busca de instruções ou operandos realiza a busca das informações desejadas nos dois subsistemas concomitantemente, tanto na Memória Principal quanto na Memória CACHE.

A memória estática do subsistema CACHE deve ter tempos de acesso de operações de leitura e escrita muito menores que a memória principal. No entanto a sua capacidade de armazenamento (quantidade de palavras) é significativamente inferior a capacidade da memória principal. Ela irá conter sempre a cópia de um subconjunto de informações contidas na Memória Principal



**Figura 2.20– Diagrama de Blocos da constituição de um Subsistema de memória CACHE**

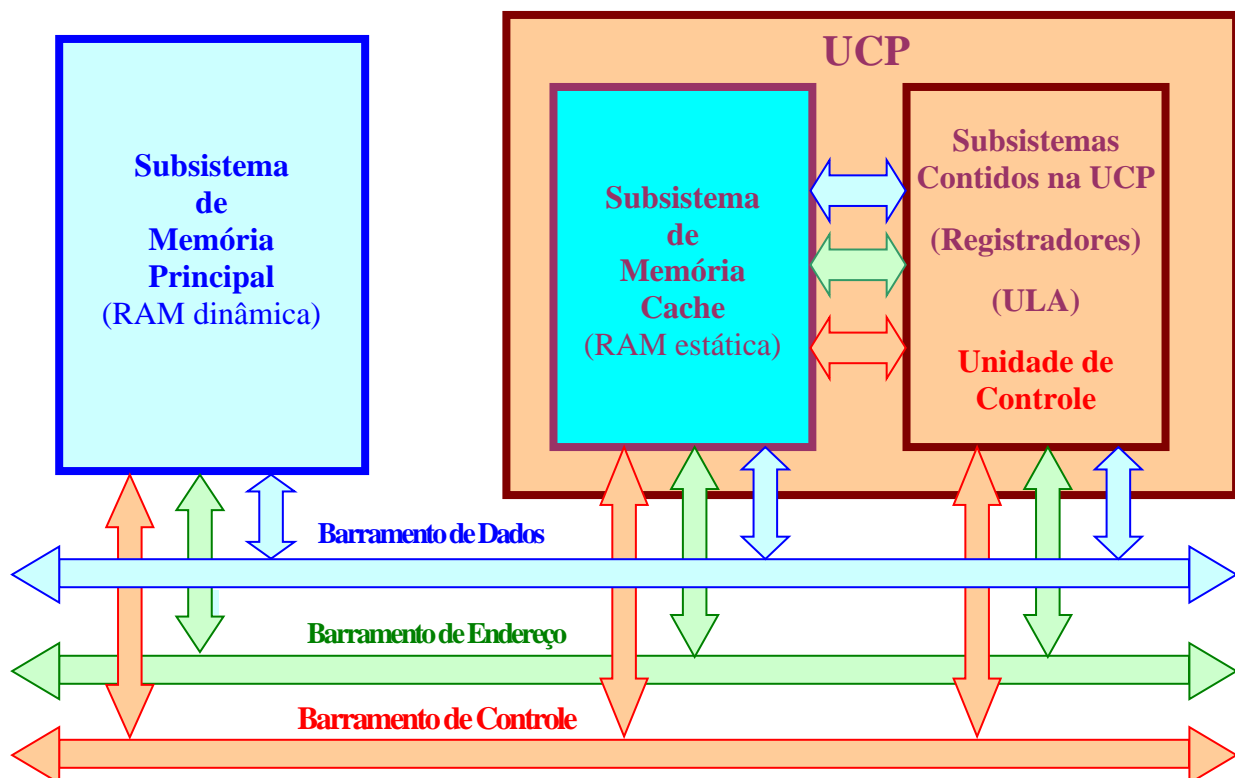
A memória associativa do subsistema CACHE é responsável pela verificação da disponibilidade de operandos e/ou instruções dentro do subsistema nas operações de busca (leitura) de instruções ou operandos realizados pela unidade de controle da UCP.

As características da memória CACHED aumentam a eficiência de execução de instrução através da diminuição do tempo de resposta na busca ou obtenção de instruções ou operandos (tempo de latência).

Para que a busca de operandos ou instruções na memória CACHE seja realizada com sucesso é necessário que as instruções e/ou operandos buscados estejam armazenados na CACHE.

Para tanto a memória CACHE deve ser preenchida com operações e instruções de um programa em execução. Como a capacidade de armazenamento da memória CACHE é significativamente menor que a capacidade de armazenamento da Memória Principal, ela só pode conter um subconjunto de instruções de um programa que está sendo executado. Assim, para que o subsistema CACHE aumente a eficiência da máquina, são necessárias metodologias adequadas de gerenciamento de CACHE. A eficiência dessas metodologias está diretamente relacionada às características do programa em execução ou de maior utilização do equipamento. Essas metodologias devem contribuir para que o conjunto de instruções

armazenado no subsistema de memória CACHE contenha um subconjunto de instruções com alta probabilidade serem as próximas instruções a serem executadas pela CPU.



**Figura 2.21 – Diagrama de Blocos da utilização de um sistema de memória Cache**

São duas as formas mais utilizadas para o preenchimento da memória estática, do subsistema de memória CACHE. Essas formas estão relacionadas ou às características de sequencialização de execução de instruções, ou pela localização de um conjunto de instruções.

Conseqüentemente são denominadas de forma temporal ou forma espacial.

A forma temporal realiza o organiza o preenchimento da memória estática da CACHE concomitantemente a primeira busca e execução da instrução pela UCP. Uma após a outra, uma cópia da instrução buscada pela CPU na memória principal é armazenada na CACHE. Essa forma é altamente recomendável quando a quantidade de iterações contidas em um programa é grande. Pois em um comando iterativo as instruções são executadas muitas vezes em uma pequena faixa de tempo. Essa forma garante que após a primeira iteração de uma estrutura repetitiva, uma cópia de todas as instruções que serão reexecutadas, estará totalmente contida (armazenada) no subsistema CACHE.

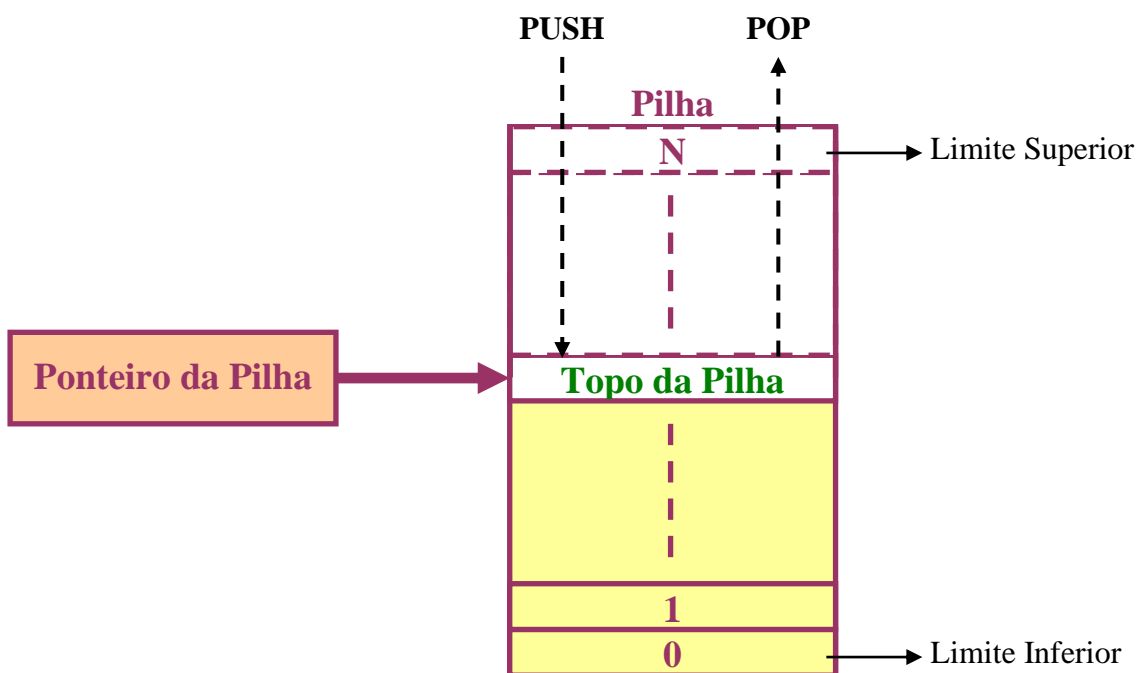
A forma espacial preenche o subsistema CACHE com um subconjunto de instruções armazenadas sequencialmente (endereços consecutivos) na memória principal. Assim, após a busca de uma instrução na Memória Principal acarreta a realização de uma cópia

(transferência) de um subconjunto de instruções contíguas no subsistema CACHE. A eficiência nessa forma de preenchimento da CAHE é obtida do princípio de que elaboração de um programa é realizada de forma seqüencial pelo programador e consequentemente carregadas de forma sequencial na Memória Principal, portanto com grande probabilidade de também serem executadas de forma seqüencial. Assim a busca de instruções no subsistema CACHE terá sucesso na maioria dos ciclos de busca de instruções realizadas pela CPU.

## 2.4- Subsistema de Memória de Pilha (LIFO)

Uma estrutura de Pilha, também denominada “LIFO” é um subsistema de armazenamento de informação com características peculiares para ser utilizada em determinadas aplicações.

A sigla LIFO é gerada pela forma como os dados são inseridos e retirados de seu interior. O mnemônico LIFO é o mnemônico de “Last In First Out” (“Último a Entrar (chegar) é o Primeiro a Sair”).



**Figura 2.22- Diagrama de blocos de uma Pilha (LIFO)**

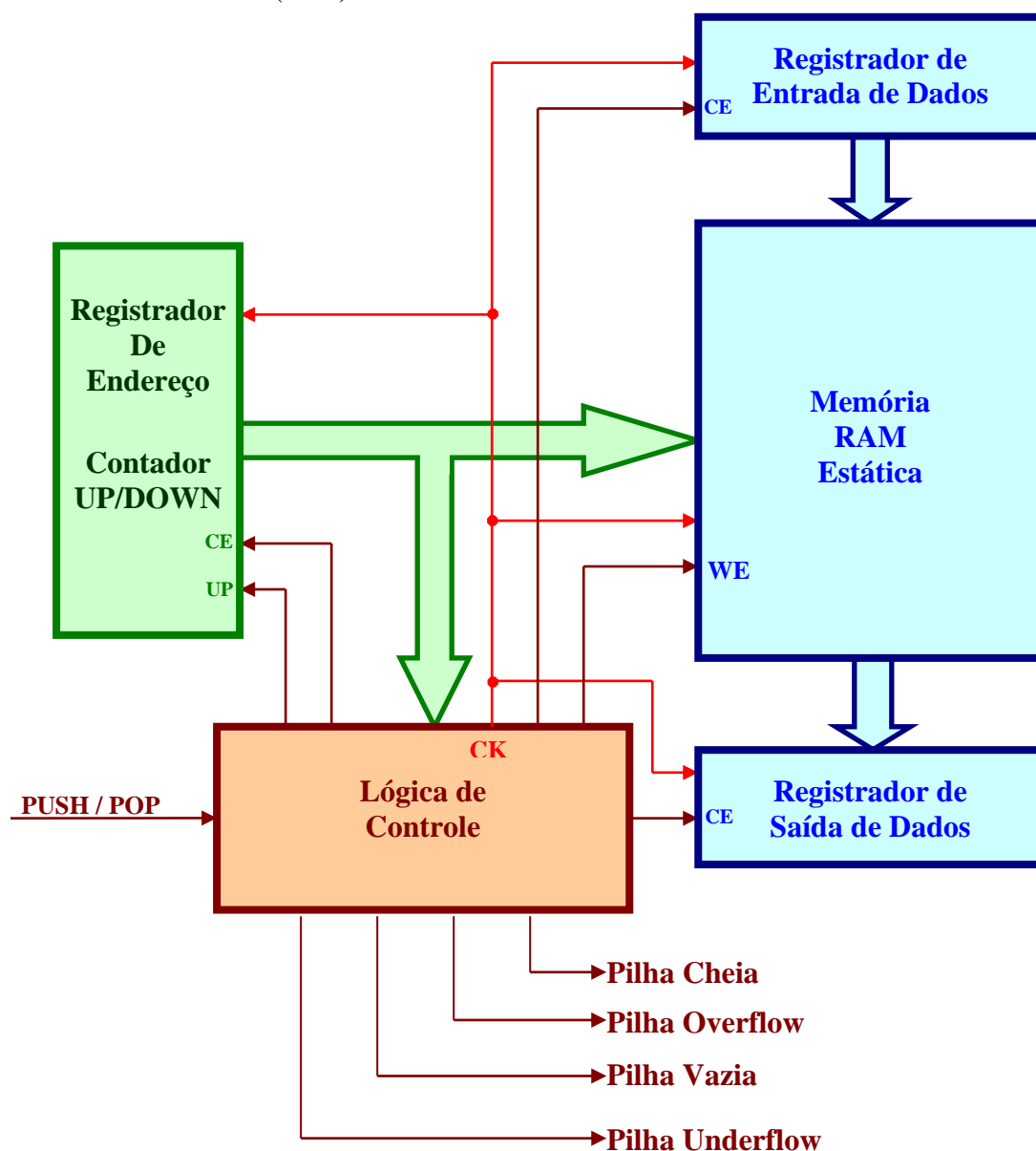
Uma estrutura de Pilha clássica, mostrada na Figura 2.22, é composta pela seguinte estrutura física:

- Possui um sistema de memória com um limite inferior e um limite superior bem definido;
- Possui um ponteiro (registrador de endereço) que aponta sempre para a primeira posição vazia da estrutura de memória da pilha, denominado “Topo da Pilha”;



- Contém duas operações denominadas PUSH e POP;
- A operação PUSH, insere uma nova informação (elementos) na Pilha e incrementa o ponteiro;
- A operação POP retira uma informação da Pilha e decrementa o ponteiro
- Um elemento é sempre inserido no Topo da Pilha. Um elemento é sempre retirado do Topo da Pilha.

O diagrama de blocos contido na figura 2.22 mostra os principais componentes que o projeto deve conter: Registrador de Entrada de Dados (registrador com entrada e saída paralela), Registrador de Saída de Dados (registrador com entrada e saída paralela); registrador de Endereço (contador Up-Down); memória RAM de 256 palavras de 32 bits e, uma lógica de controle para implementar as operações de inserção de elementos (PUSH) e retirada de elementos (POP) na Pilha.



**Figura 2.22- Diagrama de Blocos de uma Estrutura de Pilha (LIFO)**

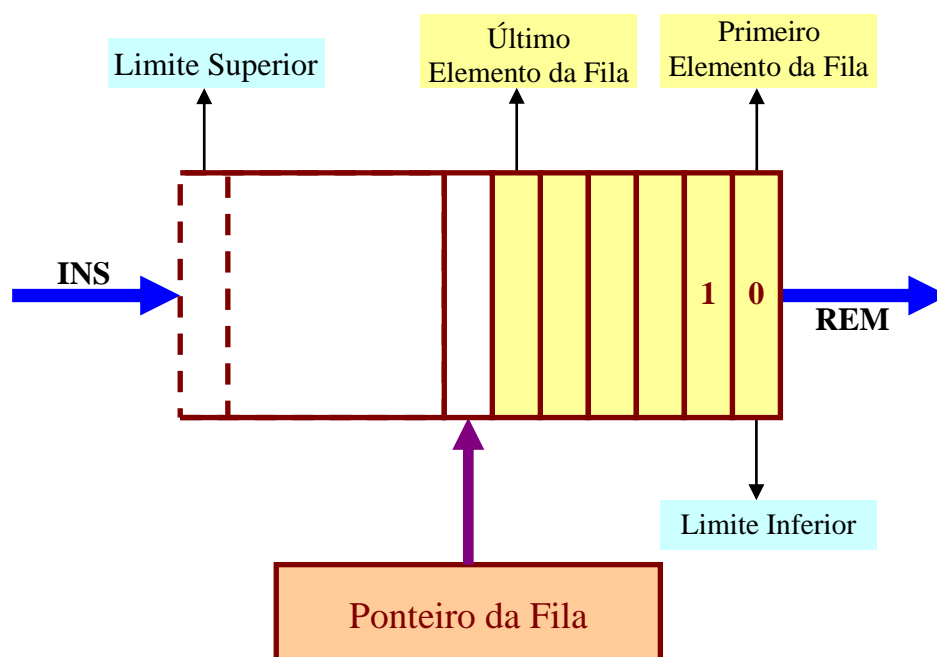
## 2.5- Subsistema de Memória de Fila (FIFO)

Uma estrutura de Fila, também denominada “FIFO” é um subsistema de armazenamento de informação com características peculiares para ser utilizada em determinadas aplicações.

A sigla FIFO é gerada pela forma como os dados são inseridos e retirados de seu interior. O mnemônico FIFO é originado pela expressão “First In First Out” (“Primeiro a Entrar (chegar) é o Primeiro a Sair”). Essa forma de funcionamento é útil para inúmeros processos e aplicações em um sistema computacional enfatizando o acesso a recursos compartilhados.

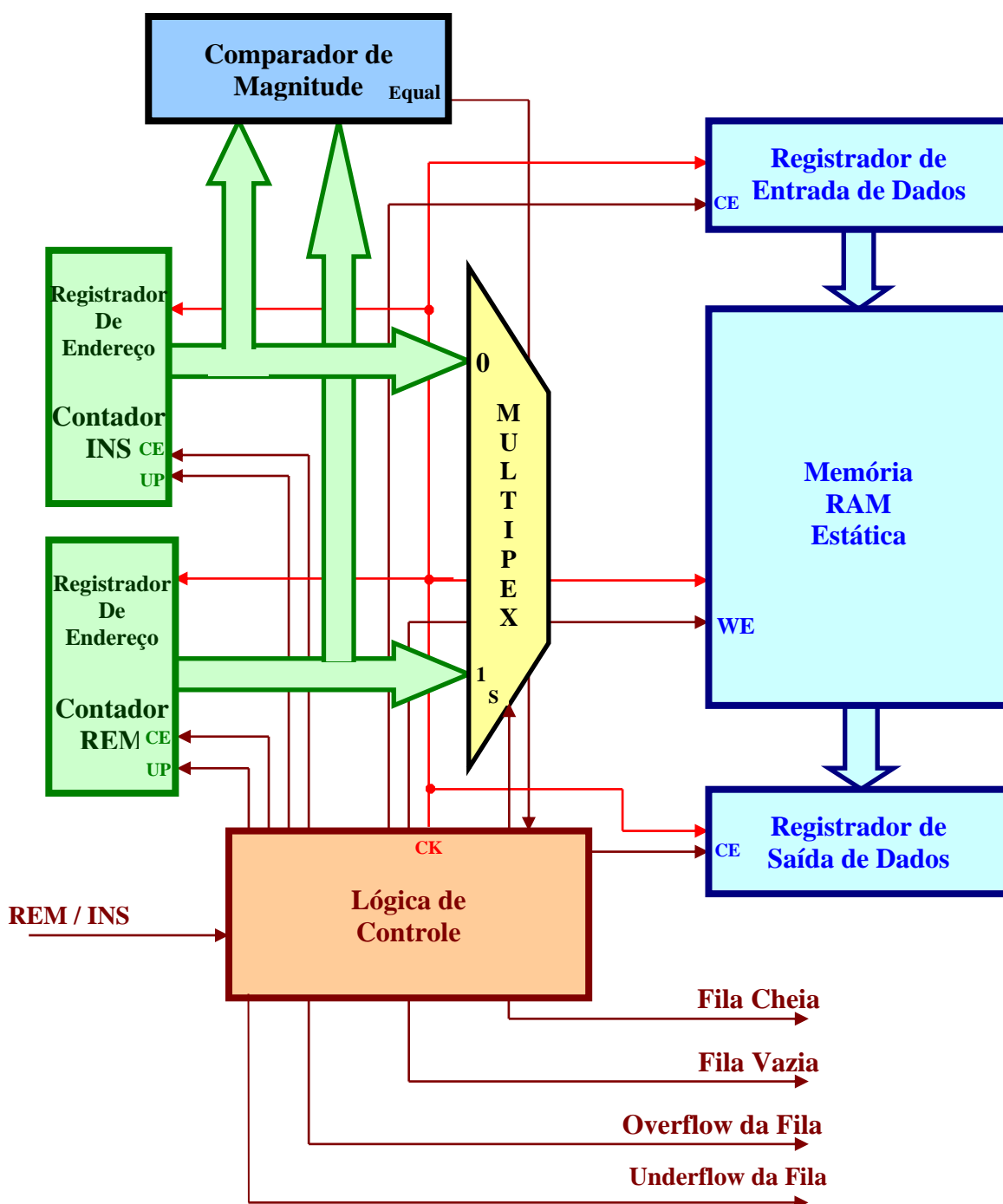
Uma estrutura de Fila clássica, mostrada na figura 2.23, é composta pela seguinte estrutura física:

- Um sistema de memória com um limite inferior e um limite superior bem definido;
- Possui um ponteiro que aponta sempre para o registrador que deve receber uma nova informação. Quando a Fila está vazia ele aponta para o registrador posicionado no limite inferior da Fila. Quando a Fila está não vazia ele aponta para o primeiro registrador disponível após a inserção do último elemento na Fila.
- Contém duas operações denominadas **Insert (INS)** e **Remove (REM)**;
- A operação **INS** insere uma informação no final da fila;
- A operação **REM** retira da Fila a informação armazenada no seu início.



**Figura 2.23-- Diagrama de blocos de uma estrutura de Fila (FIFO)**

Uma estrutura de Fila pode ser implementada por hardware e anexada a arquitetura de um processador. A figura. 2.24, mostra, em diagrama de blocos, a implementação de uma Fila utilizando uma memória RAM estática. Uma RAM possui como principal característica a forma de acesso aleatório e, em uma Fila, o acesso é realizado de forma sequencial. Assim deverão ser inseridos no projeto estruturas de hardware que viabilizem a forma de desempenho (funcionamento) de uma Fila.



*Figura 2.24- Diagrama de Blocos de uma Fila implementado com memória RAM*

O diagrama de blocos contido na figura 2.24 mostra os principais componentes que o projeto deve conter:

- Registrador de Entrada de Dados (registrador com entrada e saída paralela);
- Registrador de Saída de Dados (registrador com entrada e saída paralela);
- Dois registradores de Endereço (contadores Up-Down), um para a operação **INS** e outro para a operação **REM**. O registrador de endereço denominado “**Contador INS**” é utilizado para inserir um elemento na Fila, ele aponta sempre para o primeiro registrador livre quando a fila não está cheia. O registrador de endereço denominado “**Contador REM**” é utilizado para remover uma informação (dado) da fila. Ele aponta sempre para o primeiro elemento da Fila;
- Memória RAM estática;
- Uma lógica de controle para implementar: as operações de inserção de elementos e remoção de elementos de sua estrutura e gerar e verificar o status da fila (fila cheia, overflow da fila, fila vazia, underflow da fila) após a execução de cada operação de inserção e remoção.

## Anexo VII.1

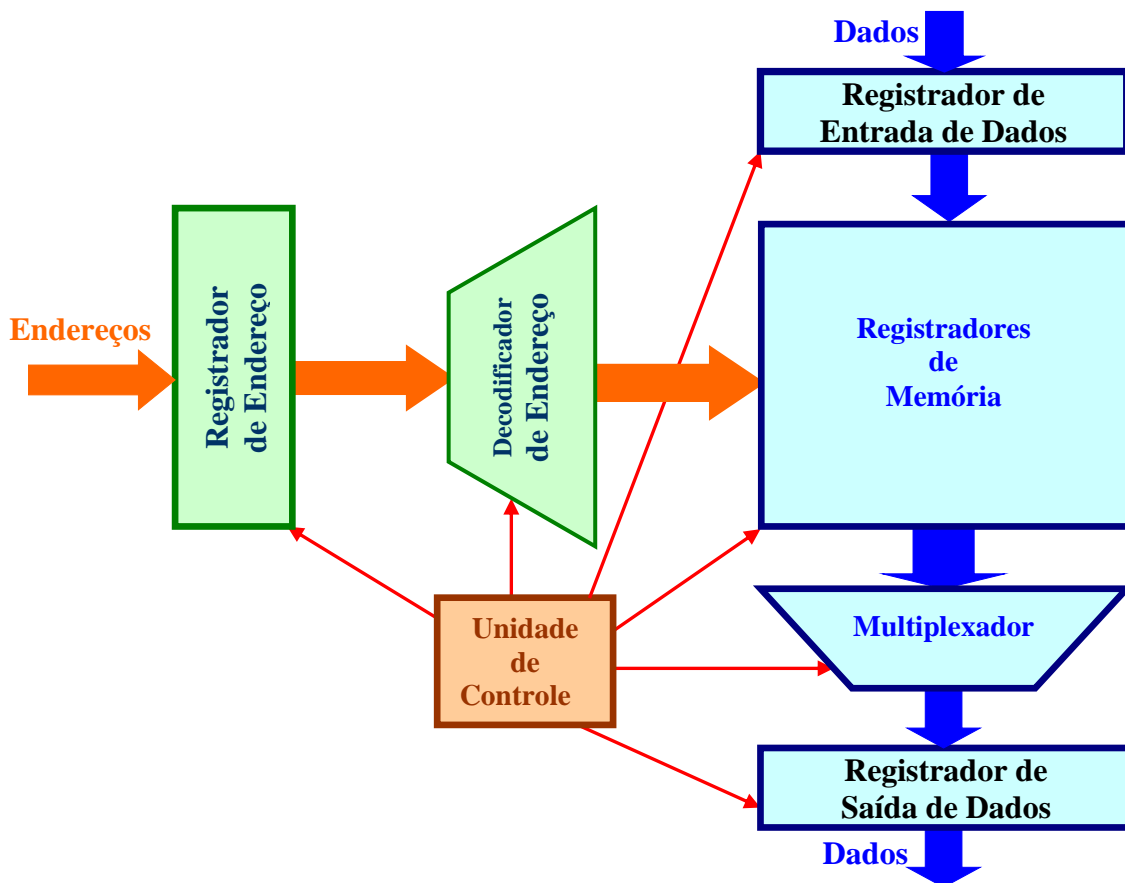
### Projetos Propostos

#### Projeto 2.1

#### Subsistemas de Memória (RAM Estática de 2Kx32)

Sintetizar, projetar, implementar e validar através de simulação um subsistema de memória com as seguintes características: volátil, estática e acesso aleatório; contendo 2048 palavras de 32 bits. (2Kx32).

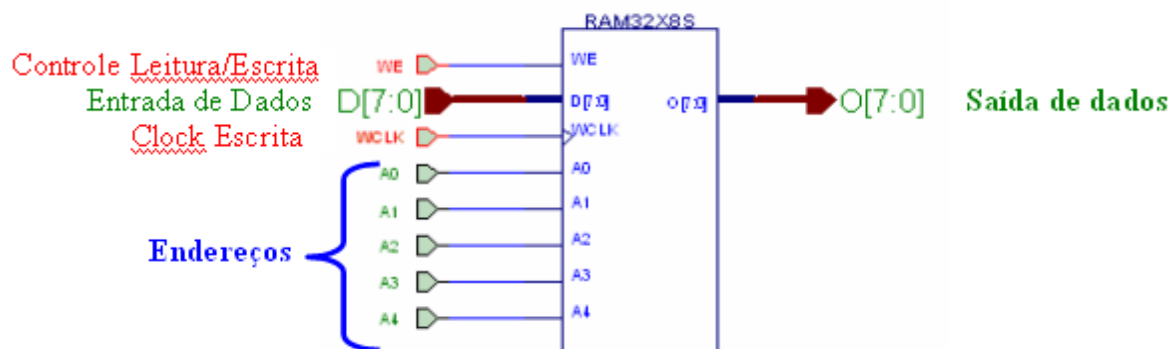
O subsistema solicitado deverá conter todos circuitos que realizem todas as funções contidas no diagrama de bloco da figura P2.1.1 que segue.



**Figura P2.1.1- Organização clássica de um Subsistema de Memória**

Na implementação do subsistema de memória deve ser utilizado o componente de memória “RAM32X8S” contido na biblioteca software Project Manager e na figura P2.1.2. O componente “RAM32X8S” contém 32 palavras de 8 bits. Assim, para a obtenção de um subsistema de memória de 1K palavras de 32 bits deverá ser utilizada técnica de associação em série e paralelo de componentes eletrônicos. Descreva como foi realizado o

dimensionamento e a organização dos componentes “RAM32X8S” para implementar o subsistema de memória solicitado (1Kx32).



**Figura P2.1.2-Características do Elemento de Memória RAM32X8S do Project Manager**

O relatório e documentação do projeto deverá seguir o formato sugerido no documento “Roteiro para a Geração de Relatórios de Projetos” (Anexo VII-2) e conter, entre outras, as seguintes informações:

- Conceitos sobre organização e classificação de memória considerados relevantes para a implementação do projeto;
- Dimensionamento dos barramentos de endereço e de dados;
- Mapa da memória especificando a quantidade de bancos de memória e os endereços inicial e final de cada um dos bancos de dados existentes na implementação realizada;
- Os algoritmos que realizam as operações de leitura (inserção) e escrita (remoção) de informações no subsistema de memória;
- Os microprogramas que implementam os algoritmos de escrita e leitura, contendo toda a sequência de microordens;
- A simulação deverá conter informações que comprovem o perfeito funcionamento de todos os bancos de memória;
- O projeto de todos os demais blocos contidos na figura P71.1, exceto a unidade de controle;

## Projeto 2.2

### Subsistema de Memória de Pilha (LIFO 256x32)

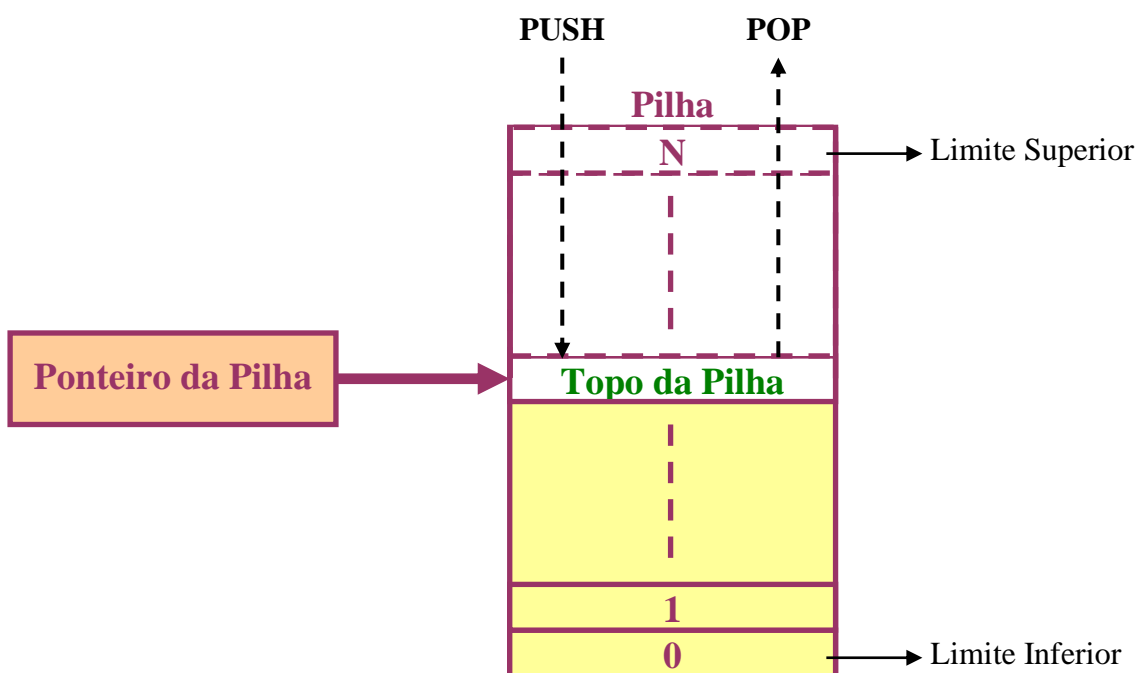
Sintetizar, projetar, validar e simular uma estrutura de pilha contendo 256 palavras de 32 bits, utilizando componentes de memória RAM Estática e o software Project Manager.

Uma estrutura de Pilha, também denominada “LIFO” é um subsistema de armazenamento de informação com características peculiares para ser utilizada em determinadas aplicações.

A sigla LIFO é gerada pela forma como os dados são inseridos e retirados de seu interior. O mnemônico LIFO é o mnemônico de “Last In First Out” (“Último a Entrar (chegar) é o Primeiro a Sair”).

Uma estrutura de Pilha clássica, mostrada na figura P2.2.1, é composta pela seguinte estrutura física:

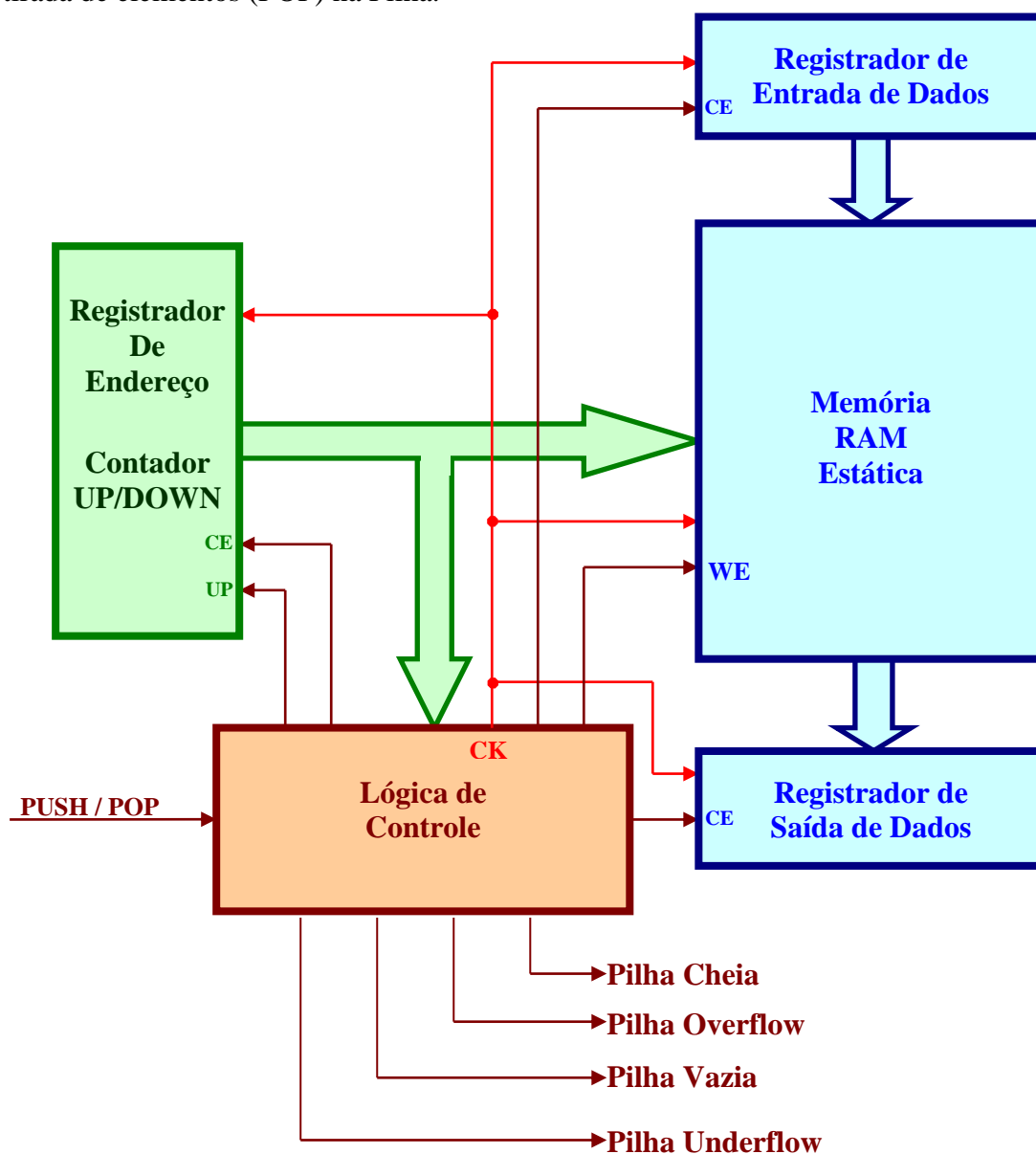
- Possui um sistema de memória com um limite inferior e um limite superior bem definido;
- Possui um ponteiro (registrador de endereço) que aponta sempre para a primeira posição vazia da estrutura de memória da pilha, denominado “Topo da Pilha”;
- Contém duas operações denominadas PUSH e POP. A operação PUSH insere uma nova informação (elementos) na Pilha. A operação POP retira uma informação da Pilha. Um elemento é sempre inserido no Topo da Pilha. Um elemento é sempre retirado do Topo da Pilha.



*Figura P2.2.1- Diagrama de blocos de uma Pilha (LIFO)*

O projeto prevê a implementação de uma Pilha utilizando uma memória RAM estática. Uma RAM possui como principal característica a forma de acesso aleatório e, em uma Pilha, o acesso é realizado de forma seqüencial. Assim deverão ser inseridos no projeto estruturas de hardware que viabilizem a forma de desempenho (funcionamento) de uma Pilha.

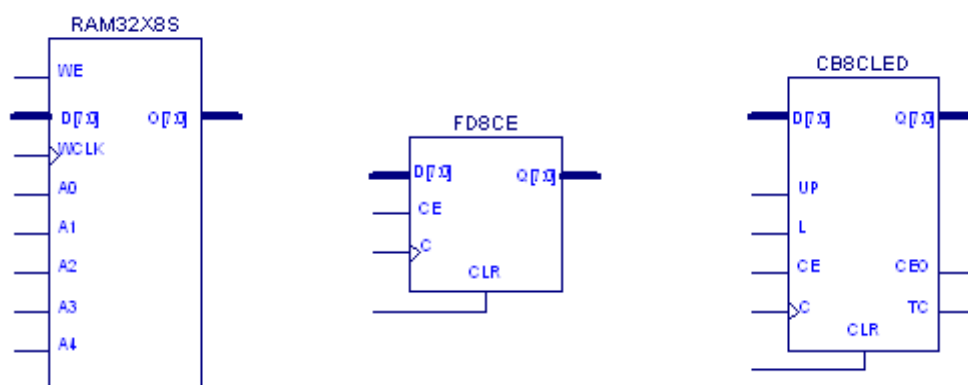
O diagrama de blocos contido na figura P2.2.2 mostra os principais componentes que o projeto deve conter: Registrador de Entrada de Dados (registrador com entrada e saída paralela), Registrador de Saída de Dados (registrador com entrada e saída paralela); registrador de Endereço (contador Up-Down); memória RAM de 256 palavras de 32 bits e, uma lógica de controle para implementar as operações de inserção de elementos (PUSH) e retirada de elementos (POP) na Pilha.



*Figura P2.2.2- Diagrama de Blocos de uma Estrutura de Pilha (LIFO)*



Podem ser utilizados componentes contidos na biblioteca do software Project Manager na implementação do projeto. A figura P2.2.3 mostra sugestões de componentes dessa biblioteca. O componente RAM 32X8S é uma memória RAM de 32 palavras de 8 bits com Chip Enable e Clock. O componente FD8CE é um registrador de 8 bits e entrada e saída paralelas contendo Chip Enable e Clock. O componente CB8CLED é um contador bidirecional UP/DOWN de 8 bits.



**Figura P2.2.3- Componentes da Biblioteca do software Project Manager**

A descrição, funcionamento e tabela verdade de todos os componentes sugeridos está disponibilizada de forma detalhada nos itens no capítulo VIII desse livro, denominado “Sistemas de Uma UCP”. O componente FD8CE na seção 8.2.1, o componente CB8CLED no item 8.2.3 e o componente COMP8 no item 8.2.5.

### Observações:

O relatório e documentação do projeto deverá seguir o formato sugerido no documento “Roteiro para a Geração de Relatórios de Projetos” (Anexo VII-2) e conter, entre outras, as seguintes informações:

- A Pilha deverá conter 256 registradores de 32 bits;
- O projeto deverá conter a verificação dos estados de pilha vazia e pilha cheia e overflow da pilha;
- Detalhar a sequência de microordens necessária para inserir um elemento na pilha (operação PUSH);
- Detalhar a sequência de microordens necessária para retirar um elemento da pilha (operação POP);
- O relatório do projeto deve conter os itens relacionados no Roteiro para Geração de Relatórios (Anexo VII-2).

## Projeto 2.3

# Subsistema de Memória de Fila (FIFO 256x32)

Sintetizar, projetar e validar através de simulação uma estrutura de Fila contendo 256 palavras de 32 bits, utilizando componentes de memória RAM Estática e o software Project Manager.

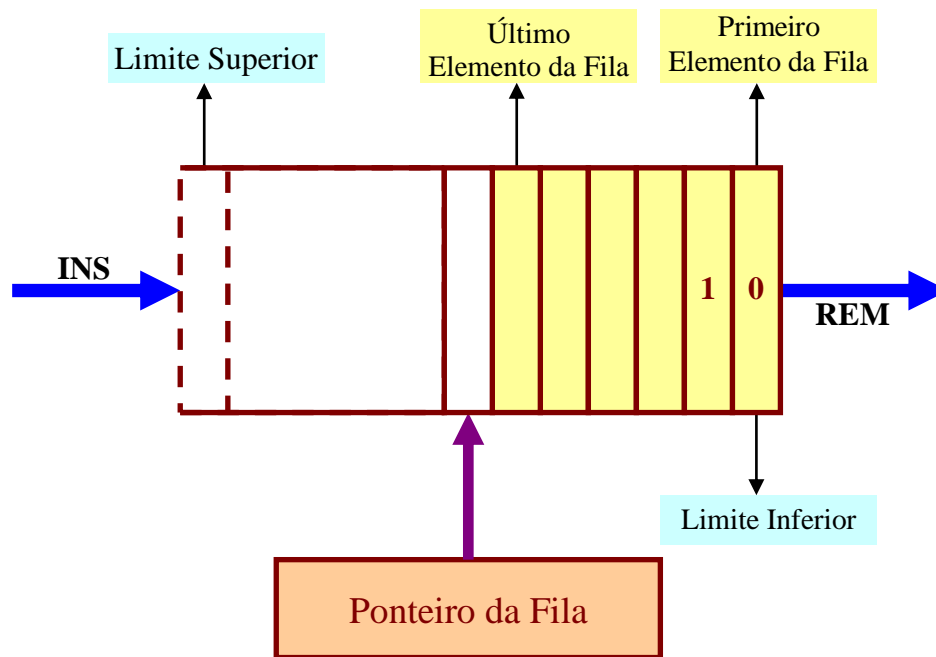
Uma estrutura de Fila, também denominada “FIFO” é um subsistema de armazenamento de informação com características peculiares para ser utilizada em determinadas aplicações.

A sigla FIFO é gerada pela forma como os dados são inseridos e retirados de seu interior. O mnemônico FIFO é originado pela expressão “First In First Out” (“Primeiro a Entrar (chegar) é o Primeiro a Sair”). Essa forma de funcionamento é útil para inúmeros processos e aplicações em um sistema computacional enfatizando o acesso a recursos compartilhados.

Uma estrutura de Fila clássica, mostrada na Figura P73.1, é composta pela seguinte estrutura física:

- Um sistema de memória com um limite inferior e um limite superior bem definido;
- Possui um ponteiro que aponta sempre para o registrador que deve receber uma nova informação. Quando a Fila está vazia ele aponta para o registrador posicionado no limite inferior da Fila. Quando a Fila está não vazia ele aponta para o primeiro registrador disponível após a inserção do último elemento na Fila.
- Contém duas operações denominadas **Insert (INS)** e **Remove (REM)**. A operação **INS** insere uma informação no final da fila. A operação **REM** retira da Fila a informação armazenada no seu início.

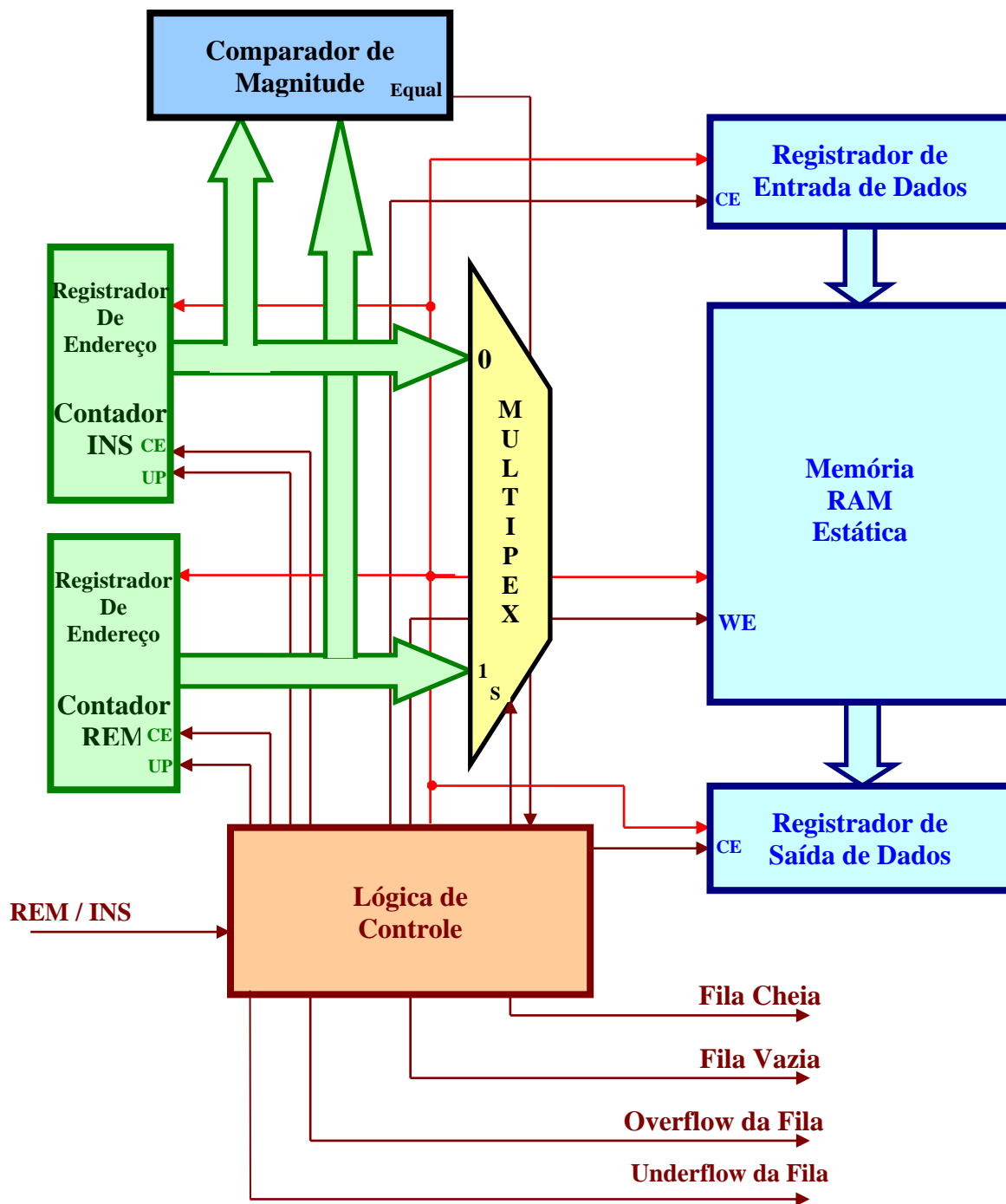
O projeto prevê a implementação de uma Fila utilizando uma memória RAM estática. Uma RAM possui como principal característica a forma de acesso aleatório e, em uma Fila, o acesso é realizado de forma seqüencial. Assim deverão ser inseridos no projeto estruturas de hardware que viabilizem a forma de desempenho (funcionamento) de uma Fila.



**Figura P2.3.1-- Diagrama de blocos de uma estrutura de Fila (FIFO)**

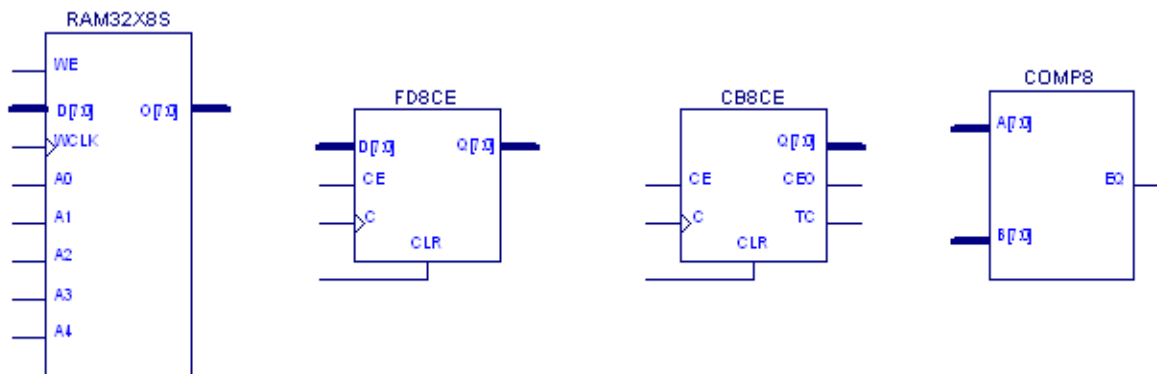
O diagrama de blocos contido na figura P73.2 mostra os principais componentes que o projeto deve conter:

- Registrador de Entrada de Dados (registrador com entrada e saída paralela);
- Registrador de Saída de Dados (registrador com entrada e saída paralela);
- Dois registradores de Endereço (contadores Up-Down), um para a operação **INS** e outro para a operação **REM**. O registrador de endereço denominado “*Contador INS*” é utilizado para inserir um elemento na Fila, ele aponta sempre para o primeiro registrador livre quando a fila não está cheia. O registrador de endereço denominado “*Contador REM*” é utilizado para remover uma informação (dado) da fila. Ele aponta sempre para o primeiro elemento da Fila;
- Memória RAM estática de 256 palavras de 32 bits e;
- Uma lógica de controle para implementar: as operações de inserção de elementos e remoção de elementos de sua estrutura e gerar e verificar o status da fila (fila cheia, overflow da fila, fila vazia, underflow da fila) após a execução de cada operação de inserção e remoção.



**Figura P2.3.2- Diagrama de Blocos de uma Fila implementado com memória RAM**

Podem ser utilizados componentes contidos na biblioteca do software Project Manager na implementação do projeto. A figura P2.3.3, mostra sugestões de componentes dessa biblioteca. O componente RAM 32X8S é uma memória RAM estática de 32 palavras de 8 bits contendo Clock para a operação de escrita e Chip Enable. O componente FD8CE é um registrador de 8 bits que possui entradas e saídas paralelas e Chip Enable. O componente CB8CE é um contador crescente unidirecional de 8 bits com Chip Enable. O componente COMP8 é um comparador de magnitude de 8 bits que verifica a igualdade dos valores inseridos nas suas entradas.



**Figura P2.3.3- Componentes da Biblioteca do software Project Manager**

A descrição, funcionamento e tabela verdade de todos os componentes sugeridos está disponibilizada de forma detalhada nos itens no capítulo VIII desse livro, denominado “Subsistemas de Uma UCP”. O componente FD8CE na seção 8.2.1, o componente CB8CELED no item 8.2.3 e o componente COMP8 no item 8.2.5.

#### **Observações:**

O relatório e documentação do projeto deverá seguir o formato sugerido no documento “Roteiro para a Geração de Relatórios de Projetos” (Anexo VII-2) e conter, entre outras, as seguintes informações:

- A Fila deverá conter 256 registradores de 32 bits;
- O projeto deverá conter a verificação dos estados de fila vazia e fila cheia e overflow da fila;
- Detalhar a sequência de microordens necessária para realizar a inserção de um elemento na fila (operação INS);
- Detalhar a sequência de microordens necessária para realizar a remoção de um elemento da fila (operação REM).

## Anexo VII.2

### Roteiro para a Geração de Relatórios de Projetos

Os relatórios relativos a cada projeto solicitado devem conter os seguintes itens:

**Capa de rosto com as seguintes informações:**

- Identificação do curso;
- Identificação da disciplina;
- Título do Projeto;
- Identificação do(s) aluno(s);
- Identificação do professor.

**1- Objetivos do trabalho.**

**2- Descrição Geral do subsistema implementado:**

Descrição macroscópica e sintética do subsistema.

Relacionar e descrever conceitos importantes e características relevantes utilizados durante na implementação do projeto.

Para auxiliar a descrição do subsistema pode ser inserida figura contendo o diagrama lógico do mesmo.

**3- Descrição dos módulos do subsistema:**

Descrever a forma de funcionamento e as funções exercidas pelos blocos contidos no diagrama lógico do circuito, relacionando e descrevendo as suas entradas e saídas.

Documentar o circuito de forma a auxiliar a sua interpretação e utilização.

Inserir circuito(s) lógico(s) e/ou seqüencial (ais) implementado(s) no Project Manager.

Para cada circuito implementado mostre:

- A forma de síntese e o funcionamento do componente;
- A respectiva tabela verdade;
- A função exercida pelo componente no projeto em tela.

**4- Simulação do circuito (Project Manager).**

A simulação do circuito deve conter testes que possibilitem verificar o funcionamento correto de todas as funções exercidas pelo subsistema implementado, identificando e comentando os pontos (instantes) mais importantes da simulação.

Para cada passo de simulação relacionar as microordens ativadas.

Inserir microprogramas (seqüência de microrordens) utilizados na simulação.

**5- Aplicações.**

Relacionar e descrever aplicações do subsistema implementado.

**6- Conclusões.**

Descrever a forma como os objetivos foram alcançados relacionando prováveis problemas encontrados.

Descrever: a contribuição do projeto para a sua formação; a importância do mesmo na área de organização e arquitetura de computadores.

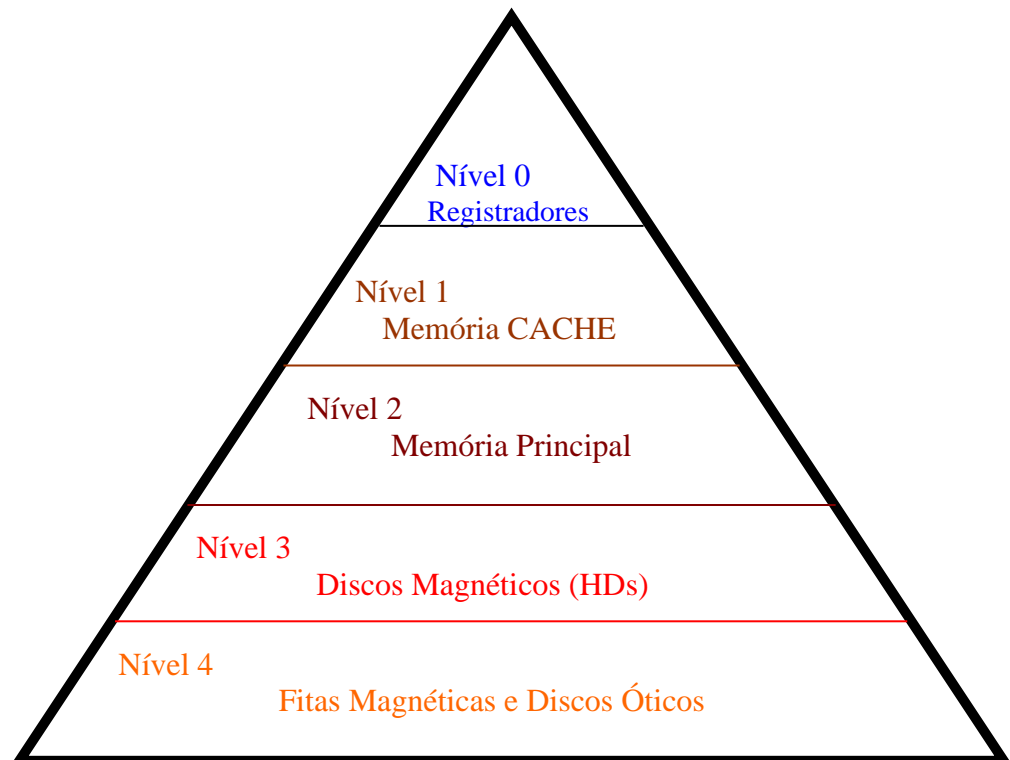
**7- Bibliografia utilizada e/ou consultada.**

Inserir relação de livros, manuais, artigos e páginas consultadas. Os itens deverão conter as seguintes informações: autor, título, ano de publicação, editora. A consulta a páginas na Internet deverá conter a identificação (nome e autor) da página, endereço e data de acesso.

## Anexo VII.3

### Questões de Múltipla Escolha

1- Utilizando a figura 2.2 que segue mostra a organização clássica de hierarquia de memória, classifique os elementos ou sistemas de memória destacados em cada um dos níveis hierárquicos, de acordo com os seguintes critérios: manutenção da informação com energia; manutenção da informação no tempo e forma de acesso. As alternativas mostram a classificação de cada um dos níveis na sequência de critérios solicitada. Assinale a alternativa correta



**Figura 2.2- Organização Clássica de Hierarquia de Memória**

- (A) 0- Volátil, Estática, Sequencial; 1- Volátil, Estática, Aleatório; 2-Volátil, Dinâmica, Aleatório; 3- Não Volátil, Estática, Sequencial; 4- Não Volátil, Estática, Sequencial.
- (B) 0- Volátil, Estática, Aleatório; 1- Volátil, Dinâmica, Aleatório; 2-Volátil, Dinâmica, Aleatório; 3- Não Volátil, Estática, Sequencial; 4- Não Volátil, Estática, Sequencial.
- (C) 0- Volátil, Estática, Aleatório; 1- Volátil, Estática, Aleatório; 2-Volátil, Dinâmica, Aleatório; 3- Não Volátil, Estática, Sequencial; 4- Não Volátil, Estática, Sequencial.
- (D) 0- Volátil, Estática, Aleatório; 1- Volátil, Estática, Aleatório; 2-Volátil, Estática, Aleatório; 3- Não Volátil, Estática, Sequencial; 4- Não Volátil, Estática, Sequencial.
- (E) 0- Volátil, Estática, Aleatório; 1- Volátil, Estática, Aleatório; 2-Volátil, Dinâmica, Aleatório; 3- Não Volátil, Estática, Sequencial; 4- Não Volátil, Estática, Aleatório.



2- Uma Máquina possui capacidade máxima de memória principal = 8 GBytes palavras de 64 bits. Qual deve ser o comprimento dos barramentos de endereço e de dados da arquitetura dessa máquina?

(Obs:- 1GByte = 1.073.741.824 bytes)

- (A) Endereço = 33 bits; Dados = 8 bits;
- (B) Endereço = 33 bits; Dados = 64 bits;
- (C) Endereço = 32 bits; Dados = 32 bits;
- (D) Endereço = 8 bits; Dados = 8 bits;
- (E) Endereço = 16 bits; Dados = 32 bits

3- Existem duas formas tradicionais de formatar as trilhas contidas em disco que são:

I- Inserir a mesma quantidade de setores em cada trilha;

II- Inserir quantidade variável de setores em cada trilha, sendo que as trilhas mais internas irão conter a menor quantidade de setores.

Compare as diferenças entre as duas formas de formatação, enfatizando custos e benefícios (vantagens e desvantagens) de cada forma, de acordo com as características clássicas de acesso (leitura e escrita) de dados em um Disco Rígido (HD) e assinale a alternativa correta

- (A) I- Aumenta velocidade de leitura; facilita controle; diminui capacidade de armazenamento  
II- Aumenta velocidade de leitura; facilita controle; diminui capacidade de armazenamento
- (B) I- Aumenta velocidade de leitura; controle mais elaborado; diminui capacidade de armazenamento  
II- Diminui velocidade de leitura; facilita controle; aumenta capacidade de armazenamento
- (C) I- Aumenta velocidade de leitura; facilita controle; aumenta capacidade de armazenamento  
II- Diminui velocidade de leitura; controle mais elaborado; diminui capacidade de armazenamento
- (D) I- Diminui velocidade de leitura; facilita controle; diminui capacidade de armazenamento  
II- Aumenta velocidade de leitura; controle mais elaborado; aumenta capacidade de armazenamento
- (E) I- Aumenta velocidade de leitura; facilita controle; diminui capacidade de armazenamento  
II- Diminui velocidade de leitura; controle mais elaborado; aumenta capacidade de armazenamento

Respostas	
Questão	Alternativa Correta
1	C
2	B
3	E

## Anexo VII.4

### Exercícios Propostos

**1-** Os discos rígidos, utilizados em sistemas de armazenamento secundário de informações na arquitetura de Von Neumann, podem ser formatados de forma a otimizar seu desempenho de acordo com os requisitos e características das mais diversas aplicações. De acordo com essas características, responda os itens que seguem:

- a-** Mostre e comente a organização clássica de um disco contido em uma unidade de Disco Rígido;
- b-** comente e descreva os benefícios e custos de inserir na formatação de um disco rígido a mesma quantidade de setores em cada trilha;
- c-** Comente e descreva os benefícios de inserir na formatação de um disco rígido uma quantidade diferente de setores por trilha;
- d-** Compare as diferenças entre as duas formas de formatação descritas nos itens “b” e “c”, enfatizando custos e benefícios (vantagens e desvantagens) de cada uma das formas.

**2-** Um sistema computacional pode conter diversos dispositivos de memória que se diferenciam de acordo com suas características e aplicações. Para uma arquitetura convencional, responda aos itens abaixo:

- a- Considerando a organização dos subsistemas de memória em níveis hierárquicos, relacione todos os níveis de hierarquia, classificando-os, comentando-os e comparando-os de acordo com os seguintes critérios: custo, velocidade e capacidade de armazenamento;
- b- como são classificados os diferentes dispositivos de armazenamento de informação utilizados em um sistema computacional de acordo com os seguintes critérios: manutenção da informação com energia; manutenção da informação no tempo e formas de acesso. Exemplifique, classificando os subsistemas de memória descritos no item a.

**3-** Projetar um subsistema de memória de acesso aleatório contendo dois Giga-bytes, utilizando componentes de memória de 256 Mega-bytes. Descrever e mostrar o mapa da memória considerando que um banco de memória contém 256 Mega-bytes. Detalhar o barramento de endereços de acordo com os bancos de memórias. Dimensionar o barramento de dados.

**4-** Projetar um subsistema de memória de acesso aleatório contendo dois Giga-palavras de 16 bits, utilizando componentes de memória de 512 Mega-bytes. Descrever e mostrar o mapa da memória considerando que um banco de memória contém 512 Mega-palavras. Detalhar o barramento de endereços de acordo com os bancos de memórias. Dimensionar o barramento de dados.

**5-** Projetar um subsistema de memória de acesso aleatório contendo 1,5 Giga-palavras de 64 bits, utilizando componentes de memória de 512 Mega-bytes. Descrever e mostrar o mapa da memória considerando que um banco de memória contém 512 Mega-palavras. Detalhar o barramento de endereços de acordo com os bancos de memórias. Dimensionar o barramento de dados.

**6-** Projetar um subsistema de memória RAM, com capacidade de armazenamento de 2,5 Giga palavras de 24 bits, utilizando componentes de memória de 512 Mega-bytes. Descrever e mostrar o mapa da memória considerando que um banco de memória contém 512 Mega-palavras de 24 bits (endereço inicial e final). Dimensionar e detalhar o barramento de endereços e formas de endereçamento de acordo com a quantidade de bancos de memórias contidos no subsistema solicitado. Detalhar e comentar a organização e dimensionamento do barramento de dados.

**7-** Descreva a utilidade e forma de funcionamento de um sistema de memória CACHE.

**8-** Ao acrescentar um sistema de memória CACHE em uma arquitetura, é necessário aumentar o barramento de endereço? Por quê?

**9-** Os itens que seguem referem-se a um sistema de memória Cache.

- a)** Classifique sistema de memória Cache de acordo com os critérios clássicos de classificação de memória;
- b)** Qual deve ser a organização da memória do sistema de memória Cache, quando comparado com a organização da memória principal de uma máquina?
- c)** A capacidade de armazenamento de informações de uma máquina aumenta quando é anexado, nessa máquina, um sistema de memória Cache? Por quê?
- d)** Relacione e comente duas formas de preenchimento da memória Cache;

**10-** A Figura que segue mostra, em diagrama de blocos, a estrutura de uma Fila (FIFO) composta por 8 registradores de 16 bits. Utilizando as condições iniciais da Fila que a figura mostra, realizar as operações indicadas nos itens “a”, “b” e “c”, mostrando os resultados obtidos da seguinte forma:

- Mapa da fila,
- Valor dos ponteiros,
- Estados de: Fila Vazia, Underflow da Fila, Fila Cheia e Overflow da Fila.

Mapa Inicial			
Ponteiro de Inserção	Endereço da RAM	Conteúdo	Ponteiro de Remoção
	7		
	6		
	5		
	4	EEEE	
	3	DDDD	
	2	CCCC	
	1	BBBB	
→	0		

Fila Vazia	Underflow da Fila	Fila Cheia	Overflow da Fila
0	0	0	0

a) Mostre, utilizando o diagrama abaixo, a situação da Fila no final da execução do seguinte conjunto de instruções:



REM  
INS EEEE  
REM  
REM  
REM  
REM

Mapa após a execução das Instruções do item a			
Ponteiro de Inserção	Endereço da RAM	Conteúdo	Ponteiro de Remoção
	7		
	6		
	5		
	4		
	3		
	2		
	1		
	0		

Fila Vazia	Underflow da Fila	Fila Cheia	Overflow da Fila

b) Considerando o estado final da Fila, obtido no item a, mostre no diagrama abaixo, a situação da Fila no final do seguinte conjunto de instruções:

```

INS  FFFF
INS  AAAA
INS  BBBB
REM
INS  CCCC
REM
INS  DDDD
INS  EEEE
INS  FFFF
INS  AAAA
INS  BBBB
REM
INS  CCCC
INS  DDDD

```

Mapa após a execução das Instruções do item a			
Ponteiro de Inserção	Endereço da RAM	Conteúdo	Ponteiro de Remoção
	7		
	6		
	5		
	4		
	3		
	2		
	1		
	0		

Fila Vazia	Underflow da Fila	Fila Cheia	Overflow da Fila

c) Considerando o resultado obtido no item b, mostre no diagrama abaixo, a situação da Fila no final do seguinte conjunto de instruções:

```

REM
INS  EEEE
REM
INS  FFFF
INS  AAAA

```

Mapa após a execução das Instruções do item b			
Ponteiro de Inserção	Endereço da RAM	Conteúdo	Ponteiro de Remoção
	7		
	6		
	5		
	4		
	3		
	2		
	1		
	0		

Fila Vazia	Underflow da Fila	Fila Cheia	Overflow da Fila

### Observações referentes ao Exercício 10.

1- A operação “**INS XXXX**”, realiza as seguintes tarefas:

- Insere na Fila o valor XXXX na posição indicada pelo Ponteiro de Inserção;
- Se Ponteiro de Inserção = 7 então Zera Ponteiro de Inserção;  
senão Incrementa o Ponteiro de Inserção.

2- A operação “**REM**”, realiza as seguintes tarefas:

- Remove da Fila o valor contido na posição indicada pelo Ponteiro de remoção;
- Se Ponteiro de Remoção = 7 então Zera Ponteiro de Remoção;  
senão Incrementa Ponteiro de Remoção.

**11-** Um subsistema de memória cachê, quando inserido em uma arquitetura modifica algumas características de desempenho e controle de uma arquitetura. Responda aos itens abaixo:

- a- Qual a utilidade e o principal benefício auferido por uma arquitetura quando é inserido um subsistema cachê em seu sistema de memória?
- b- Relacione e descreva as formas ou políticas de preenchimento de um sistema de memória “Cache” de acordo com o princípio da localidade;
- c- Descreva as principais políticas utilizadas em um sistema de memória “Cache” para realizar a substituição de palavras ou blocos nele contidos?

**12-** No anexo VII-1 foi sugerido a implementação de um Subsistema de Memória de Pilha (Projeto 2.2). Utilizando esse projeto, responda aos itens que seguem: Descrever e comentar as principais características de uma estrutura de pilha;

- a) Descrever e comentar as principais características do circuito seqüencial implementado;
- b) Relacionar e comentar três aplicações da estrutura de pilha;
- c) Descrever os principais benefícios obtidos ao inserir uma estrutura de pilha implementada em hardware em uma arquitetura de computador.

**13-** No anexo VII-1 foi sugerido a implementação de um Subsistema de Memória de Fila (Projeto 2.3). Utilizando esse projeto, responda aos itens que seguem:

- a) Descrever e comentar as principais características de uma estrutura de fila;
- b) Descrever e comentar as principais características do circuito seqüencial implementado;
- c) Relacionar e comentar aplicações da estrutura de fila;
- d) Descrever os principais benefícios obtidos ao inserir uma estrutura de fila implementada em hardware em uma arquitetura de computador.

'A mente que se abre a uma nova idéia  
jamais volta ao seu tamanho original'.  
(Albert Einstein)