

ESTUDO COMPARATIVO E SIMULAÇÃO DE MEMÓRIAS DIGITAIS: EEPROM, FLASH, RAM¹

Bruno Malinowski Lutz²

Mateus da Silva Zuccoli³

Carlos Alberto Bertuol⁴

Gabriel de Souza Philipus⁵

Prof. MSc. Juliano De Pellegrin Pacheco⁶

RESUMO

A hierarquia de memória é um pilar da arquitetura de computadores, definida pelo trade-off entre memórias voláteis (RAM) e não-voláteis (NVM). Este artigo apresentou uma análise das tecnologias SRAM, DRAM, EEPROM e FLASH, com foco em suas arquiteturas e trade-offs de aplicação. O objetivo foi revisar os princípios operacionais que diferenciam essas memórias e validar experimentalmente a arquitetura da SRAM. A metodologia foi um projeto prático e simulatório de uma memória SRAM 16x8 no simulador lógico Logisim. Os resultados experimentais validaram a funcionalidade de leitura e escrita da SRAM simulada e demonstraram sua alta complexidade lógica, o que justifica seu alto custo e baixa densidade. O estudo conclui que a simulação validou com sucesso os princípios teóricos da SRAM, reforçando que a seleção da memória é um trade-off fundamental entre granularidade (EEPROM), densidade (FLASH) e velocidade (SRAM).

I. INTRODUÇÃO

A arquitetura de computadores moderna fundamenta-se em uma hierarquia de memória projetada para equilibrar a disparidade entre a velocidade de processamento da CPU e a capacidade de armazenamento. O desafio central dessa estrutura reside no trade-off entre memórias voláteis (RAM), essenciais pela velocidade mas dependentes de energia, e memórias não-voláteis (NVM), que garantem a persistência dos dados. Este trabalho analisa e compara as tecnologias fundamentais — RAM, EEPROM e FLASH — e apresenta o desenvolvimento prático de uma memória SRAM funcional de 16 palavras por 8 bits. O projeto, realizado no simulador lógico Logisim, abrange desde a modelagem da célula básica de 1 bit até a implementação do sistema completo, integrando decodificadores e multiplexadores para validar os princípios de funcionamento dessas arquiteturas.

¹ Artigo apresentado na disciplina de Circuitos e Sistemas Digitais da Faculdade de Ciências da Computação da Universidade Tuiuti do Paraná. 2º Sem. 2025.

² Graduando Ciência da Computação da Universidade Tuiuti do Paraná. Curitiba, Paraná. E-mail: bruno.lutz@utp.edu.br

³ Graduando Ciência da Computação da Universidade Tuiuti do Paraná. Curitiba, Paraná. E-mail: mateus.zuccoli@utp.edu.br

⁴ Graduando Ciência da Computação da Universidade Tuiuti do Paraná. Curitiba, Paraná. E-mail: carlos.bertuol@utp.edu.br

⁵ Graduando Ciência da Computação da Universidade Tuiuti do Paraná. Curitiba, Paraná. E-mail: gabriel.philipus@utp.edu.br

⁶ Orientador. Universidade Tuiuti do Paraná. Curitiba, Paraná. E-mail: juliano.pacheco@utp.br

II. REFERENCIAL TEÓRICO

A. Memória RAM (Random Access Memory)

A Memória de Acesso Aleatório (RAM) constitui a principal memória de trabalho volátil em um sistema computacional. Sua designação deriva da característica de que qualquer localização de memória pode ser lida ou escrita em um tempo constante, independentemente de sua posição física, um conceito conhecido como “acesso aleatório” [1]. Como uma memória de Leitura e Escrita (RWM), ela armazena temporariamente os dados e instruções que a CPU necessita para a execução de programas.

A característica fundamental de toda RAM semicondutora moderna é a volatilidade. Conforme detalhado por Stallings [1] e Tocci [2], os dados em uma RAM são mantidos apenas enquanto a alimentação elétrica do circuito é mantida. Uma vez que a energia é interrompida, toda a informação armazenada é perdida. A tecnologia RAM divide-se em duas categorias principais, com arquiteturas de célula e, consequentemente, trade-offs de desempenho e custo fundamentalmente distintos: Estática (SRAM) e Dinâmica (DRAM).

1) SRAM (Static RAM)

A Memória RAM Estática (SRAM) armazena cada bit de informação em um circuito biestável, comumente um latch ou Flip-Flop. A célula de SRAM mais comum é construída com seis transistores (6T) [1].

A célula 6T utiliza dois inversores CMOS em retroalimentação (cross-coupled) para criar dois estados estáveis (0 e 1). Enquanto a célula estiver energizada, ela manterá seu estado “estaticamente”, sem a necessidade de intervenção externa. Os outros dois transistores atuam como portas de acesso, controlados pela linha de palavra (Word Line), para permitir operações de leitura ou escrita [2].

A natureza “estática” da célula elimina a necessidade de ciclos de refresh. Isso, aliado à sua arquitetura baseada em transistores, resulta em tempos de acesso extremamente baixos (alta velocidade).

No entanto, a complexidade da célula (6 transistores por bit) implica em uma baixa densidade de armazenamento e um alto custo por bit. Devido à sua alta velocidade, a SRAM é a tecnologia de escolha para memórias cache (L1, L2, L3), que servem como um buffer de alta velocidade entre a CPU e a memória principal (DRAM) [1].

2) DRAM (Dynamic RAM)

A Memória RAM Dinâmica (DRAM) é a tecnologia de mais alta densidade e menor custo por bit, sendo a base para a memória principal da maioria dos sistemas computacionais.

A DRAM adota uma arquitetura de célula radicalmente mais simples, composta por apenas um transistor e um capacitor (1T1C) [1]. O bit de dados é armazenado como uma carga elétrica no capacitor: a presença de carga representa um '1' e a ausência de carga representa um '0'. O transistor atua como uma chave para conectar o capacitor à linha de bit (Bit Line) para leitura ou escrita. A simplicidade da célula 1T1C permite uma densidade de armazenamento altíssima e um custo por bit muito baixo.

O capacitor não é ideal e sua carga vaza naturalmente. Para evitar a perda de dados, a DRAM requer um circuito de controle externo que, a cada poucos milissegundos, executa um ciclo de refresh para ler e reescrever (recarregar) cada célula. É essa necessidade de atualização constante que lhe confere o nome "Dinâmica" [1], [2].

O processo de leitura é destrutivo. Para ler o bit, o transistor conecta o capacitor à linha de bit, esvaziando sua carga. Um amplificador de detecção lê a pequena variação de tensão, e o próprio circuito deve então reescrever o valor que acabou de ser lido de volta na célula.

A troca de menor velocidade e maior complexidade de controle é compensado pelo baixo custo e alta capacidade. A DRAM é, portanto, a tecnologia universalmente utilizada para a memória principal (de trabalho) em computadores, servidores e dispositivos móveis [1].

B. Memórias Não-Voláteis (NVM):

1) ROM e PROM

O referencial teórico das memórias não-voláteis (NVM) começa com a ROM (Memória exclusivamente de Leitura). Conforme Stallings [1], a ROM "contém um padrão permanente de dados... gravados fisicamente no chip como parte do processo de fabricação". Isso apresenta dois problemas: um alto custo fixo de produção e a total falta de flexibilidade, já que "se um bit estiver errado, o lote inteiro de ROM precisará ser descartado".

A primeira solução para a rigidez da ROM foi a PROM (ROM Programável). Esta variação permitiu que a gravação ("programação") fosse "realizada eletricamente por um fornecedor ou cliente", tornando-a viável para volumes menores [1]. No entanto, a PROM mantém uma limitação crítica: ela só pode ser gravada uma vez. Para aplicações que exigem atualizações de firmware ou correção de bugs, a PROM ainda é inadequada, pois o chip precisaria ser descartado.

2) O Transistor de Porta Flutuante

A demanda por uma memória não-volátil que pudesse ser apagada e regravada eletricamente levou ao desenvolvimento da EPROM (PROM Apagável, apagável por luz UV) e, subsequentemente, da EEPROM (PROM Eletricamente Apagável) e da Memória Flash [1], [2].

A base tecnológica para todas essas memórias é o transistor de porta flutuante. Este é um tipo especial de transistor que, diferentemente de um transistor padrão, possui duas portas:

- Porta de Controle A porta externa padrão, conectada ao resto do circuito.
- Porta Flutuante: Uma camada condutora interna, completamente isolada eletricamente por um material dielétrico (óxido).

O armazenamento de um bit é alcançado através do aprisionamento de elétrons nesta porta flutuante isolada. A célula é “programada” (um bit ‘0’ é escrito) aplicando-se uma alta tensão à Porta de Controle. O campo elétrico gerado é forte o suficiente para forçar elétrons (através de tunelamento quântico ou injeção de portador quente) a atravessar a barreira de óxido e ficarem presos na Porta Flutuante [1]. Como a Porta Flutuante é totalmente isolada, os elétrons ali aprisionados não têm para onde escapar, mesmo quando a energia do sistema é desligada. Essa carga pode ser mantida por anos [2]. A presença dessa carga negativa (elétrons) na Porta Flutuante age como um “escudo”, alterando a tensão de limiar do transistor.

C. EEPROM (Electrically Erasable PROM)

Enquanto a ROM e a PROM resolvem o problema de armazenamento não-volátil, a gravação única é uma limitação severa [1]. Para solucionar a necessidade de regravação no sistema, foi feita a Memória Somente de Leitura Programável e Apagável Eletricamente (EEPROM). Esta é uma forma “mais atraente” de memória que age principalmente para leitura [1], pois pode ser programada e apagada eletricamente, sem necessidade de remoção do circuito ou equipamentos especiais.

A característica que define a EEPROM é a sua granularidade. Diferente de outras tecnologias que exigem apagamento total ou em bloco, a EEPROM permite que “somente o byte ou os bytes endereçados sejam atualizados” [1].

Esta capacidade de alteração em nível de byte é uma consequência direta da arquitetura de sua célula de memória. A EEPROM utiliza o fenômeno de Tunelamento Fowler-Nordheim (F-N) para mover elétrons para dentro ou para fora de uma porta flutuante, onde ficam aprisionados para armazenar o bit de dados [3]. Para permitir que este processo ocorra em um único byte, a célula de EEPROM tradicional é composta por dois transistores: um transistor de armazenamento (a porta flutuante) e um transistor de seleção [3]. É este transistor de seleção que permite o circuito isolar um único byte para uma operação de escrita ou apagamento, sem afetar os dados em outros endereços.

Contudo, esta arquitetura de “dois transistores por célula” [3] é a causa direta do trade-off causado pela EEPROM. A necessidade de transistores adicionais “requer uma área maior” [3] de silício, o que explica por que a EEPROM é “mais cara” e “menos densa, admitindo menos bits por chip” [1] quando comparada a outras tecnologias NVM. Além disso, o processo de tunelamento F-N é lento, resultando em uma “operação de gravação que leva muito mais tempo do que a operação de leitura” [1], tipicamente na ordem de centenas de microssegundos ou até milissegundos por byte.

Devido a essa troca de flexibilidade de byte por custo e densidade, a EEPROM é a tecnologia ideal para armazenar pequenas quantidades de dados de configuração que mudam com pouca frequência, como o BIOS de um sistema, configurações de rede ou parâmetros em microcontroladores.

D. Memória FLASH: Densidade e Arquitetura

A Memória Flash é um tipo de memória não volátil e de alta velocidade, cujo conceito foi apresentado por Fujio Masuoka, da Toshiba, no início da década de 1980 [3]. Foi projetada para superar as limitações de densidade das EPROMs e o alto custo por bit das EEPROMs tradicionais [2], [3]. Essa tecnologia permitiu escalar a capacidade de armazenamento significativamente, representando uma evolução crucial das memórias programáveis [1].

A partir da década de 1990, a Memória Flash tornou-se fundamental para dispositivos de armazenamento, culminando nas unidades de estado sólido (SSDs). Estes dispositivos gradualmente substituem as tecnologias magnéticas tradicionais, como os discos rígidos (HDs), em aplicações que exigem desempenho, oferecendo soluções mais rápidas, compactas e resistentes a impactos mecânicos [1], [2].

Seu funcionamento baseia-se em células de transistores MOSFET com porta flutuante, organizadas em matrizes [3]. Cada célula armazena um ou mais bits, representados pela carga de elétrons aprisionada na porta flutuante; essa carga altera a tensão de limiar do transistor, definindo o valor lógico armazenado (0 ou 1) [2], [3].

Diferentemente da EEPROM, que permite o apagamento byte a byte, a Memória Flash executa essa operação em blocos inteiros [1]. Essa característica aumenta drasticamente a densidade e a eficiência em gravações de grandes volumes de dados. Por outro lado, esse comportamento de bloco torna o processo menos eficiente para alterações pequenas e frequentes (granularidade fina), nicho onde a EEPROM ainda mantém superioridade técnica [1], [2].

A Memória Flash é dividida em duas arquiteturas principais: NOR Flash e NAND Flash, que diferem na organização das células e no método de acesso [1], [3].

1) Arquitetura NOR FLASH:

A NOR Flash possui células conectadas em paralelo, o que permite acesso aleatório direto e rápido a qualquer endereço de memória, similar à RAM [2]. Essa arquitetura possibilita a funcionalidade Execute-In-Place, permitindo que o processador execute código diretamente do chip. Por essa razão, é amplamente utilizada para armazenar firmwares, BIOS/UEFI e carregadores de inicialização de sistemas operacionais, onde a confiabilidade e o acesso imediato ao código são essenciais [1]. Contudo, apresenta velocidades de escrita e apagamento inferiores à NAND e possui um custo por bit maior devido à menor densidade [2].

2) Arquitetura NAND FLASH:

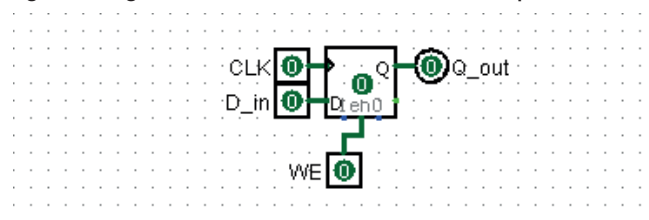
A NAND Flash, por sua vez, organiza suas células em série, o que proporciona maior densidade de armazenamento possível e o menor custo por bit entre as memórias semicondutoras [1], [3]. Seu acesso ocorre de forma sequencial, tornando-a ideal para armazenamento em massa de arquivos, como pendrives, SSDs, cartões de memória e armazenamento interno de smartphones [2]. Devido ao acesso sequencial, a NAND não permite a execução direta de código; portanto, os dados e programas nela armazenados precisam ser copiados para a memória RAM antes de serem processados pela CPU [1].

III. METODOLOGIA

A. Ambiente de Simulação e Projeto da Célula

O ambiente de simulação utilizado foi o Logisim. O bloco fundamental, a Célula de 1-Bit de SRAM, foi projetado utilizando um Flip-Flop tipo D síncrono, utilizado para simplificação de simulação. Esta escolha de componente implementa logicamente o comportamento biestável da célula de memória estática, garantindo a retenção de dados sem a necessidade de refresh (característico das DRAMs). O circuito possui uma entrada de dados (D), uma entrada de Clock (CLK) e uma entrada de habilitação (WE - Write Enable), conforme ilustrado na Fig. 1.

Fig. 1. Diagrama lógico da Célula de 1-Bit de SRAM implementada no Logisim.

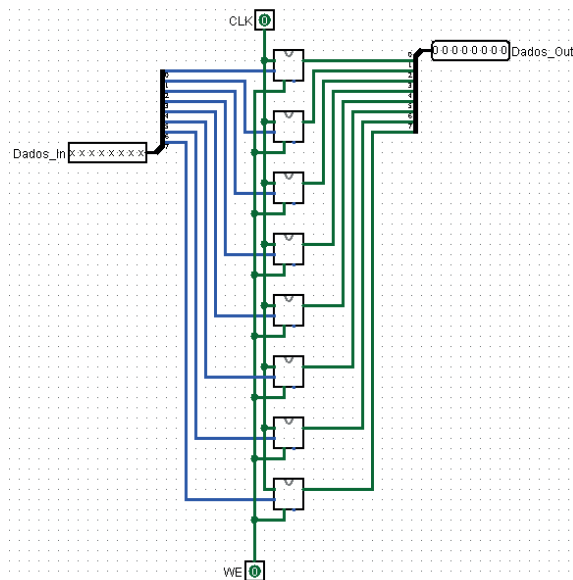


Fonte: Os autores.

B. Projeto da Arquitetura RAM 16x8

A arquitetura da RAM foi construída de forma hierárquica para simular a organização de matrizes de memória reais. Primeiramente, oito Células de 1-Bit foram encapsuladas para formar um Registrador de 8 bits (Fig. 2), representando uma “palavra” de memória. Este bloco utiliza Splitters para gerenciar o barramento de dados. Em seguida, 16 destes Registradores foram instanciados para formar o banco de memória principal.

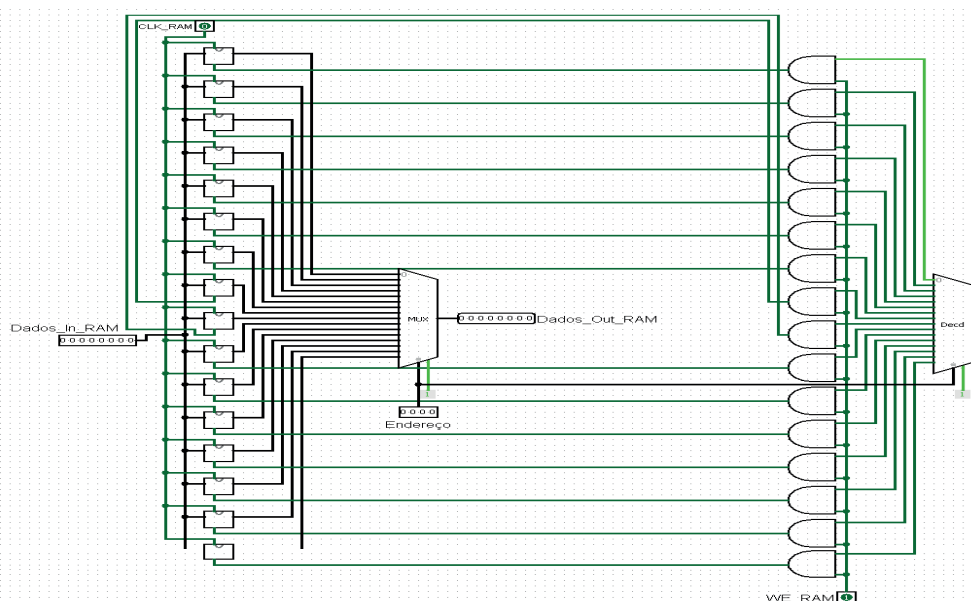
Fig. 2. Sub-circuito do Registrador de 8 bits, encapsulando 8 células de 1 bit.



Fonte: Os autores.

A lógica de endereçamento (Fig. 3) materializa o conceito de decodificação de linha e coluna. Foi utilizado um Decodificador 4-para-16 para a seleção de escrita (ativando apenas o pino WE da palavra endereçada) e um Multiplexador (MUX) 16-para-1 para direcionar o dado da palavra selecionada para o barramento de saída.

Fig. 3. Arquitetura completa da RAM 16x8, exibindo os 16 registradores e a lógica de seleção com Decoder (esquerda) e MUX (direita).



Fonte: Os autores.

C. Procedimentos de Validação

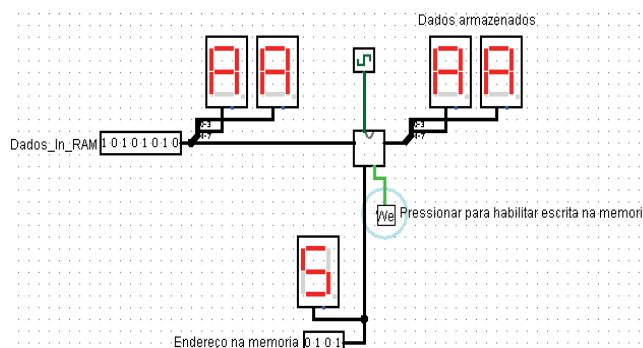
Para validar a funcionalidade da RAM frente aos requisitos teóricos, os seguintes casos de teste foram definidos: (1) Teste de Escrita e Leitura, (2) Teste de Isolamento de Endereço (para garantir que a escrita em um endereço não corrompa outro) e (3) Teste de Sobrescrita.

RESULTADOS EXPERIMENTAIS

Ambiente de Simulação e Projeto da Célula

A bancada de testes foi implementada no circuito main do Logisim (Fig. 4). A simulação foi executada estritamente conforme os casos de teste definidos na metodologia. A Tabela I resume os resultados obtidos, comparando a saída esperada com a observada.

Fig. 4. Bancada de testes (main) validando a RAM. A figura captura o Teste de Leitura: após escrever o dado 'AA' no Endereço (5), o dado é corretamente exibido na saída 'Dados_Out_RAM'.



Fonte: Os autores.

TABELA I.

Casos de Teste	Endereço	Entrada	Write Enable	Saída esperada	Saída Observada
Escrita	5	AA	Pressionado	AA	AA
Leitura	5	FF	Não Pressionado	AA	AA
Isolamento	6	FF	Não Pressionado	00	00
Sobrescrita	5	BB	Pressionado	BB	BB

D. Análise dos Resultados

Os resultados apresentados na Tabela I e ilustrados na Fig. 4 validam experimentalmente o funcionamento lógico da arquitetura proposta. Todos os casos de teste definidos na metodologia

(escrita, leitura, sobrescrita e isolamento de endereço) foram aprovados, confirmando que a lógica de seleção com MUX e Decoder operou como esperado.

A análise aprofundada revela conexões diretas com o referencial teórico, primeiramente no que tange à validação da volatilidade. Durante a simulação, observou-se que os dados (como 'AA' e 'BB') permaneceram estáveis indefinidamente enquanto o circuito estava energizado. No entanto, a execução do comando de reinicialização (Reset Simulation), que simula o corte de energia, resultou na perda instantânea de todas as informações armazenadas, retornando as saídas 00. Este comportamento comprova empiricamente a natureza volátil da SRAM diferenciando-a fundamentalmente das memórias não-voláteis EEPROM e Flash, cuja retenção de dados é garantida fisicamente por portas flutuantes.

Além do aspecto da volatilidade, o experimento evidenciou a crítica relação entre complexidade lógica e densidade. A construção de uma memória modesta de apenas 16 palavras (128 bits totais) exigiu uma contagem massiva de componentes, totalizando 128 Flip-Flops, 16 portas AND, além de decodificadores e multiplexadores complexos. Isso demonstra na prática a baixa densidade da tecnologia SRAM apontada por Stallings [1]. Enquanto a célula de Flash exige apenas um transistor por bit, a simulação evidencia que a SRAM demanda múltiplos portões lógicos (equivalentes a seis transistores físicos) para armazenar a mesma unidade de informação. Essa complexidade experimental justifica o alto custo por área de silício discutido na teoria, confirmando por que a SRAM é inviável para armazenamento principal e restrita a aplicações de Cache.

CONCLUSÃO

Este artigo realizou uma análise comparativa das arquiteturas de memória RAM, EEPROM e FLASH, demonstrando os trade-offs fundamentais de cada tecnologia. O referencial teórico estabeleceu que a escolha da memória é definida por um balanço entre velocidade, custo, densidade e granularidade de escrita. Foi demonstrado que a arquitetura de dois transistores da EEPROM permite a granularidade de byte, ao custo de uma menor densidade, enquanto a arquitetura de transistor único da FLASH otimiza a densidade ao custo da granularidade, forçando o apagamento em bloco.

O experimento prático não apenas validou o funcionamento lógico de uma memória SRAM, mas também serviu como prova empírica dos conceitos teóricos. A simulação no Logisim demonstrou com sucesso as operações de leitura, escrita e sobreposição. Mais importante, a análise dos resultados confirmou duas verdades sobre a SRAM: sua volatilidade, observada com a perda de dados ao reiniciar a simulação, e sua baixa densidade. A elevada complexidade lógica observada para construir apenas 128 bits de memória valida experimentalmente por que a SRAM, apesar de sua velocidade, é restrita a aplicações de alto custo e baixa capacidade, como caches de CPU. O trabalho conclui, portanto, que não existe uma tecnologia de memória

“superior”, mas sim uma solução de engenharia otimizada para cada aplicação específica, seja ela a velocidade da SRAM, a granularidade da EEPROM ou a densidade da FLASH.

Em reflexão sobre este trabalho, algumas dificuldades foram encontradas no entendimento da ferramenta Logisim, tendo em vista que se trata de um aplicativo de ensino escolar e superior, e não um aplicativo otimizado e prático para o âmbito profissional. Entretanto, depois de horas de prática, foi possível compreender melhor como funciona cada componente que se estava confeccionando, assim como seus parâmetros de entrada e saída. Ademais, a utilização de blocos personalizados para a montagem de circuitos mais complexos, como a própria RAM 16x8 que é a união de outros circuitos de forma abstraída, provou-se desafiadora.

REFERÊNCIAS

- [1] W. Stallings, *Arquitetura e Organização de Computadores*, 10ª ed. São Paulo: Pearson Prentice Hall, 2010.
- [2] R. J. Tocci, N. S. Widmer, G. L. Moss, *Sistemas Digitais: Princípios e Aplicações*, 11ª ed. São Paulo: Pearson Prentice Hall, 2011.
- [3] P. Pavan, R. Bez, P. Olivo, E. Zanoni, “Flash memory cells: An overview on design, reliability and scaling,” *Proceedings of the IEEE*, vol. 85, no. 8, pp. 1248–1271, Aug. 1997.