

**UNIVERSIDADE FEDERAL DO PARANÁ**

Leonardo H. Menezes  
André N. Makoski

## **Memória SRAM 64x8 bits**

Artigo elaborado como parte da avaliação da  
Disciplina de Circuitos Integrados Digitais,  
ministrada pelos Profs.: Marlio J. do C.  
Bonfin e Oscar da C. Gouveia Filho, do  
Curso de Engenharia Elétrica, Setor de  
Tecnologia, Universidade Federal do Paraná.

**CURITIBA**

**2010**

# Índice

1. Introdução .....	3
2. SRAM .....	3
2.1. Matriz de células básicas.....	3
2.2. Circuito entrada/saída .....	6
2.3. Decodificador de endereços .....	8
2.4. Lógica de controle.....	9
3. Simulação Elétrica .....	9
3.1. Célula básica .....	9
3.2. Circuito entrada/saída .....	11
3.3. Decodificador.....	12
3.4. Circuito completo da memória.....	12
3.4.1. Parâmetros Do circuito.....	13
4. Simulação Lógica.....	13
4.1. Célula básica .....	14
4.2. Circuito Entrada/Saída .....	15
4.3. Decodificador.....	15
5. Layout .....	16
6. Conclusão.....	23
7. Bibliografia .....	24

# 1. Introdução

As SRAM são memórias de acesso aleatório que mantêm os dados armazenados desde que seja mantida a sua alimentação, não necessitando que as células sejam atualizadas. São baseadas em circuitos Flip Flop SR, sendo relativamente grandes, as células básicas utilizam seis transistores.

Uma das aplicações mais comuns para a memória SRAM é a utilização como cachê de memória para os processadores, pois o acesso a memória SRAM é muito mais rápido que à memória DRAM. As memórias SRAM são o tipo de memória RAM mais rápidas da atualidade. Um dos maiores defeitos da memória SRAM é o fato de que a miniaturização é limitada e o custo é elevado quando comparado aos outros tipos de memória DRAM.

O funcionamento do circuito é baseado nos decodificadores para selecionar a célula de memória a ser lida ou gravada. Além disso, há também um circuito de entrada/saída composto de buffers tri-states em antiparalelo que determinam se será realizada uma gravação ou uma leitura. A célula de memória é composta por um latch de inversores e uma chave de passagem que é habilitada ou desabilitada pelo decodificador.

## 2. SRAM

A memória SRAM é composta dos seguintes blocos fundamentais: Matriz de células básicas, Circuito entrada/Saída, Decodificador de endereços e Lógica de controle. A seguir veremos cada bloco fundamental separadamente.

### 2.1. Matriz de células básicas

As células básicas são compostas por flip flop's SR e chaves de passagem. Os flip flop's são compostos por dois inversores conectados em paralelo. Os inversores são compostos de dois transistores, um N e um P, em série. Os canais foram dimensionados de maneira a fazer como que a resistência do transistor canal N e do transistor canal P seja igual. Abaixo vemos o esquemático da porta inversora:

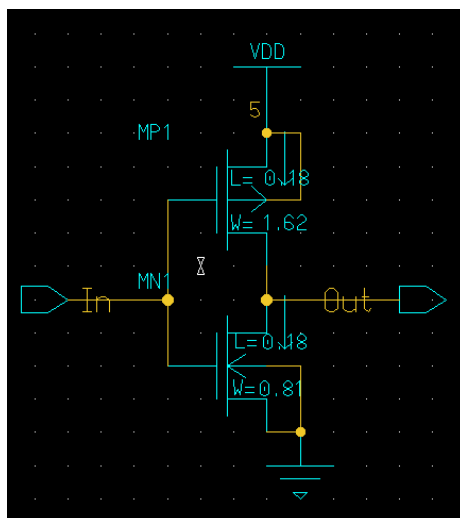


Figura 1: Porta inversora projetada.

As chaves de passagem são compostas de dois transistores N e P em paralelo. A chave de passagem é utilizada para fazer o controle do acesso ao bloco fundamental, ela é utilizada como lógica de controle. A porta inversora foi adicionada ao gate do transistor N, assim a célula é selecionada como o nível lógico zero, esta lógica foi escolhida para reduzir o numero de transistores no momento de realizar a lógica de controle. Abaixo podemos ver o esquemático da chave de passagem:

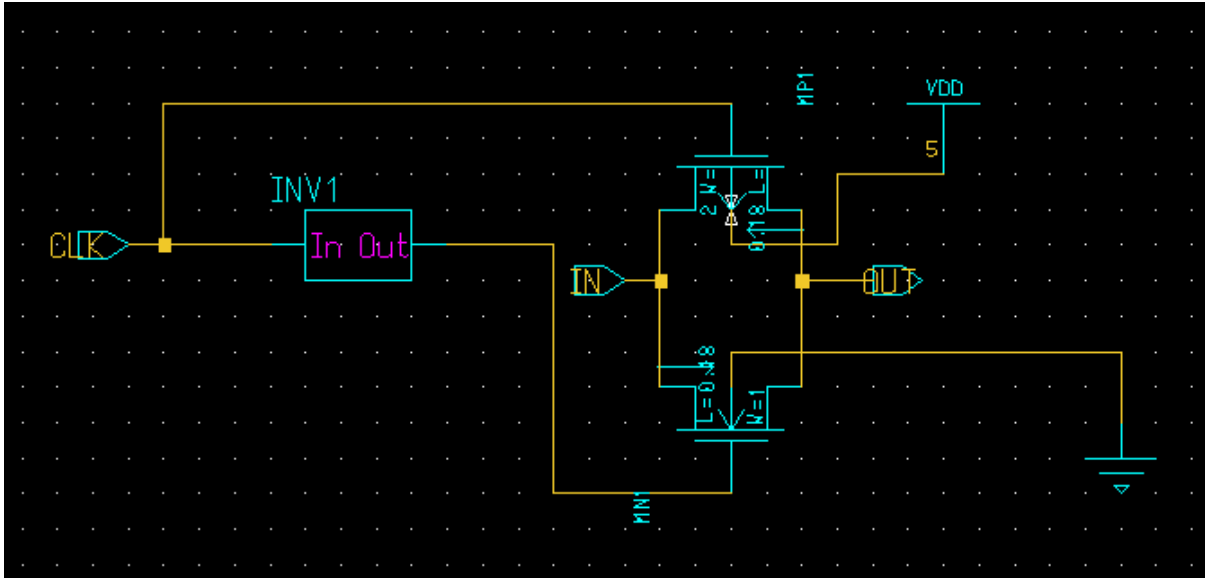


Figura 2: Chave de passagem.

Abaixo podemos ver o circuito completo da célula básica de um bit:

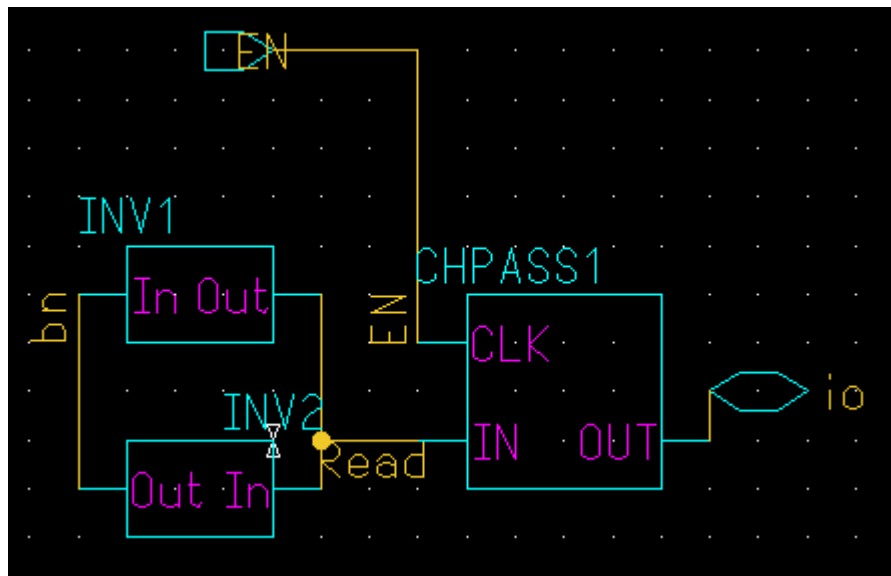


Figura 3: Circuito da célula básica com inversores em antiparalelo e a chave de passagem.

As células básicas armazenam um bit, sendo assim, elas foram agrupadas em oito células para formar a célula básica do byte. Uma porta NAND foi utilizada para realizar a lógica de controle, assim os dois decodificadores escolhem a linha e a coluna na matriz de células. A porta NAND foi utilizada para reduzir o numero de transistores do circuito, esse é o motivo pelo qual a lógica de ativação da célula básica foi invertida. Abaixo podemos ver o esquemático da célula básica de um byte:

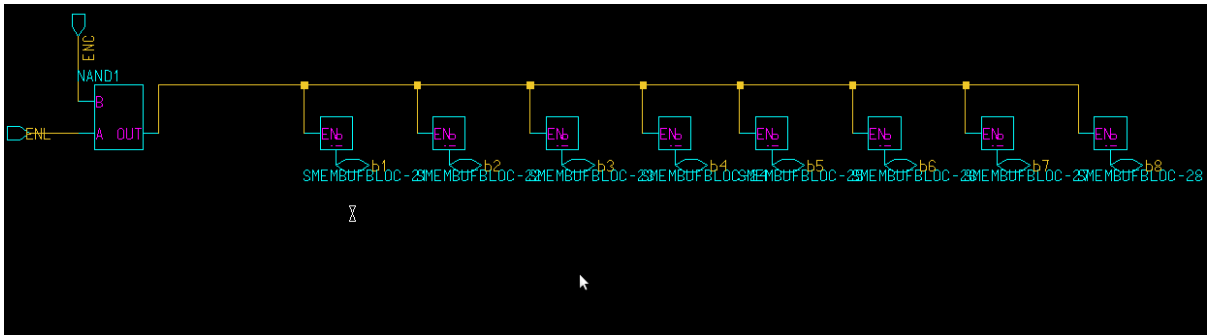


Figura 4: Célula básica de 8 bits.

Após o projeto da célula de um byte, a matriz de células foi montada, os pinos dos bits foram conectados, pois a lógica de controle permite selecionar somente uma célula. Abaixo podemos ver o circuito da matriz de células de 64 bytes:

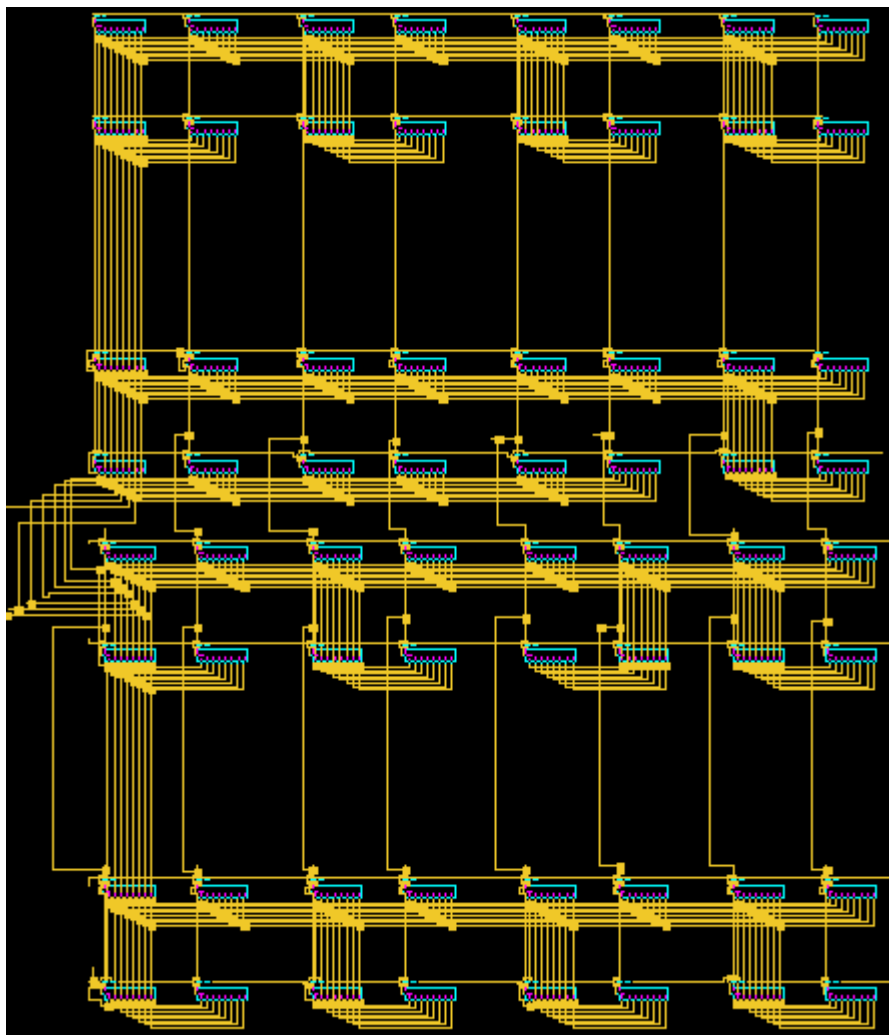


Figura 5: Célula básica de 64 bytes.

## 2.2. Circuito entrada/saída

O circuito de entrada/saída é composto de buffers tri-state ligados em antiparalelo com uma lógica de controle invertida. Os buffers tri-state são compostos por portas lógicas NAND, NOR e NOT em conjunto com dois transistores, um N e um P.

Abaixo o esquemático das portas lógicas NAND e NOR utilizadas, os canais dos transistores foram dimensionados de maneira que a resistência equivalente dos transistores P e dos transistores N seja igual, sendo assim os tempos de subida e descida seriam iguais.

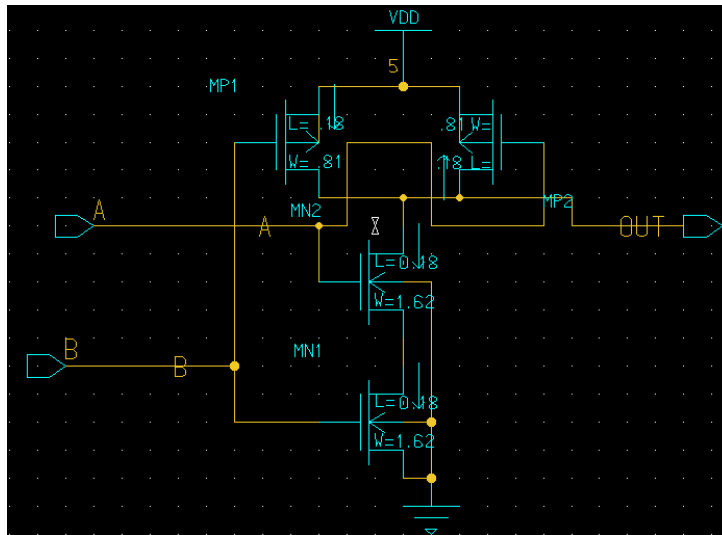


Figura 6: Porta NAND

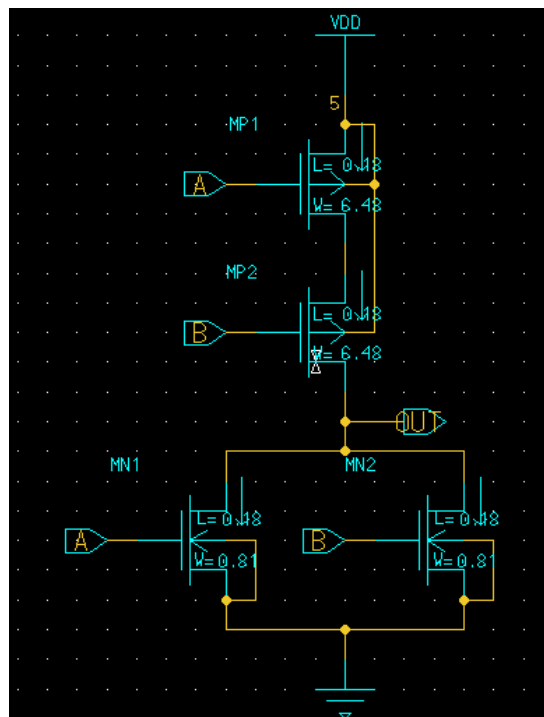


Figura 7: Porta NOR.

Os transistores da saída do buffer tri-state foram dimensionados de maneira que a corrente de escrita seja muito maior que a corrente do latch, para que assim a escrita possa ser realizada.

Abaixo podemos ver o circuito do buffer tri-state:

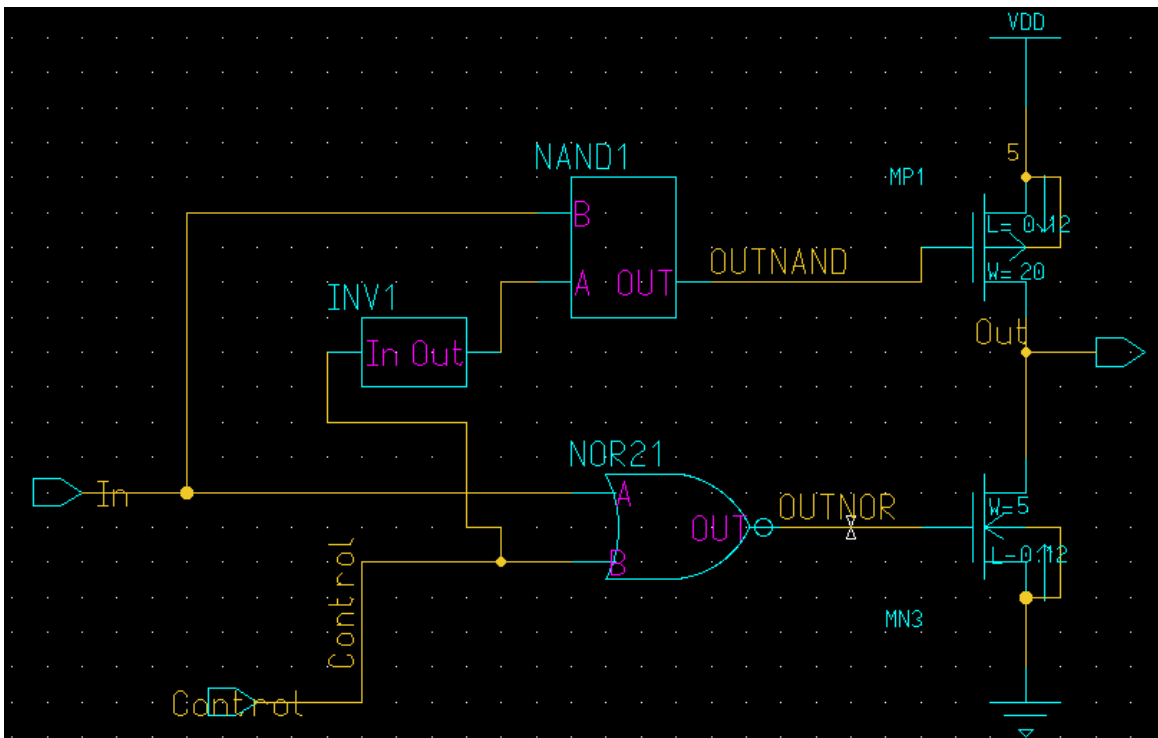


Figura 8: Circuito completo do Buffer Tri-State.

Os buffers tri-state foram ligados em antiparalelo de maneira, sendo que um dos buffers será utilizado para realizar a escrita e o outro será utilizado para realizar a leitura. Abaixo podemos ver o esquemático completo do circuito entrada/saída com os buffers tri-state conectados em antiparalelo:

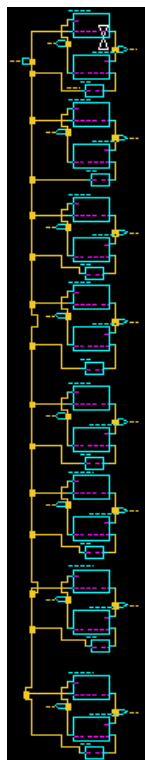


Figura 9: Circuito completo entrada/saída.

### 2.3. Decodificador de endereços

O decodificador de endereços é composto de três portas inversoras e oito portas AND de três entradas, ou seja, é um decodificador de três para oito. Dois decodificadores foram usados, um para as oito linhas e outro para as oito colunas. Abaixo podemos ver o circuito esquemático do circuito AND de três entradas.

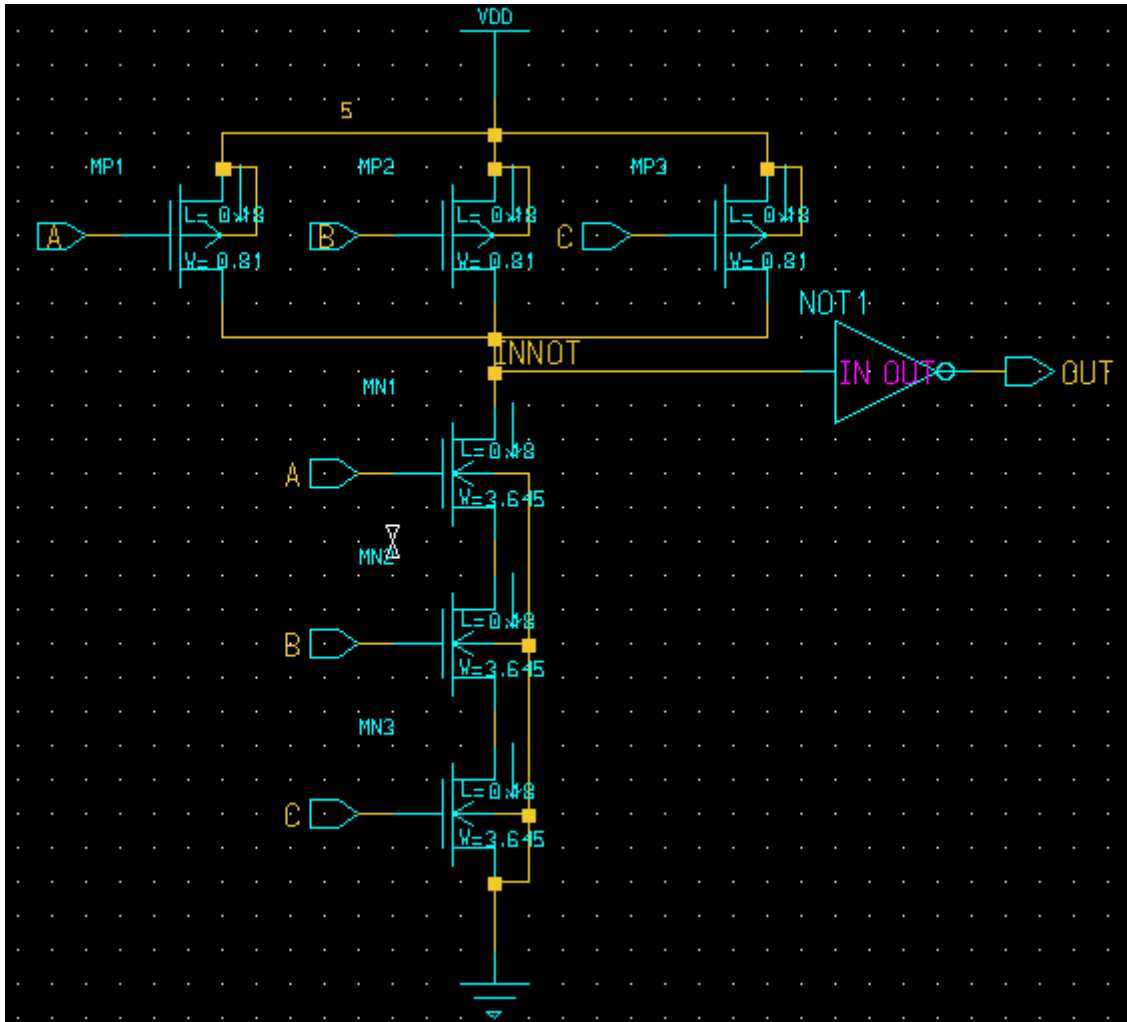


Figura 10: Circuito da porta.

Abaixo o esquemático do circuito do decodificador de três entradas para oito saídas:



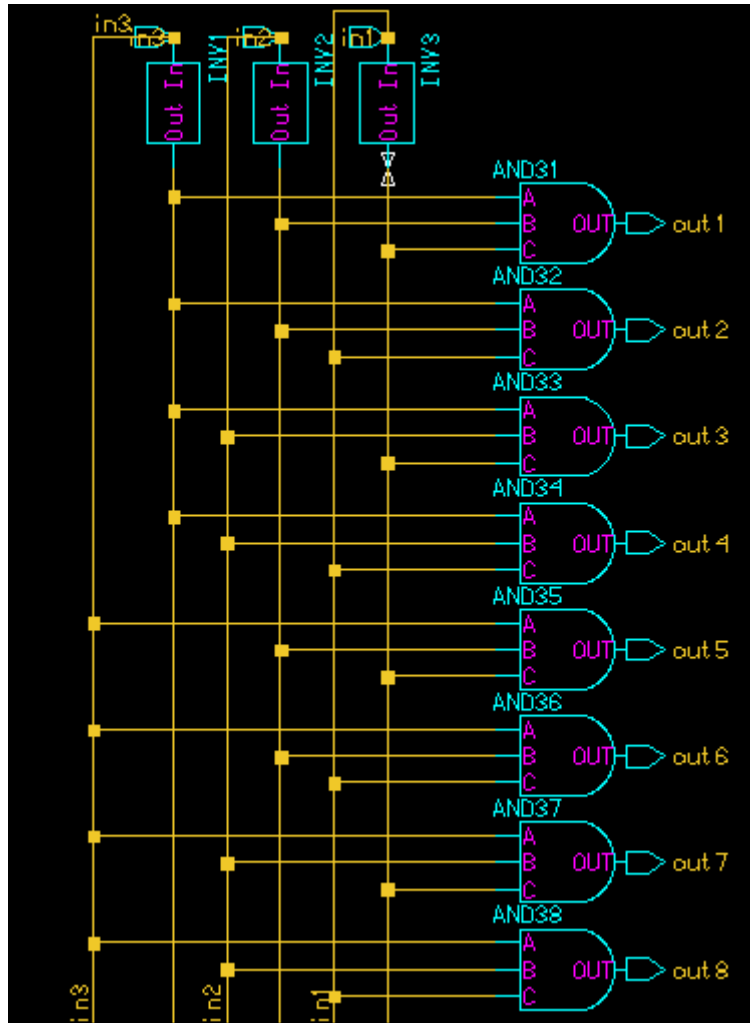


Figura 11: Circuito do decodificador.

## 2.4. Lógica de controle

O circuito de lógica de controle está no conjunto da porta NAND existente no circuito de enable da célula de um byte e do circuito do decodificador. O circuito decodificador determina a linha e a coluna, assim quando o nível lógico um é acionado tanto na linha quanto na coluna da porta NAND da célula de um byte, a chave de passagem da célula é ativada e o circuito do buffer está conectado a esta célula, logo, dependendo do nível lógico no pino RorW do buffer, a célula será lida ou gravada.

## 3. Simulação Elétrica

Os circuitos construídos foram simulados no software Mentor separadamente antes de serem colocados juntos. Abordaremos separadamente cada circuito e depois o circuito completo.

### 3.1. Célula básica

A simulação da célula básica de um bit foi realizada para verificar a capacidade do latch de armazenar o bit escrito pela entrada. Abaixo as formas de onda obtidas:

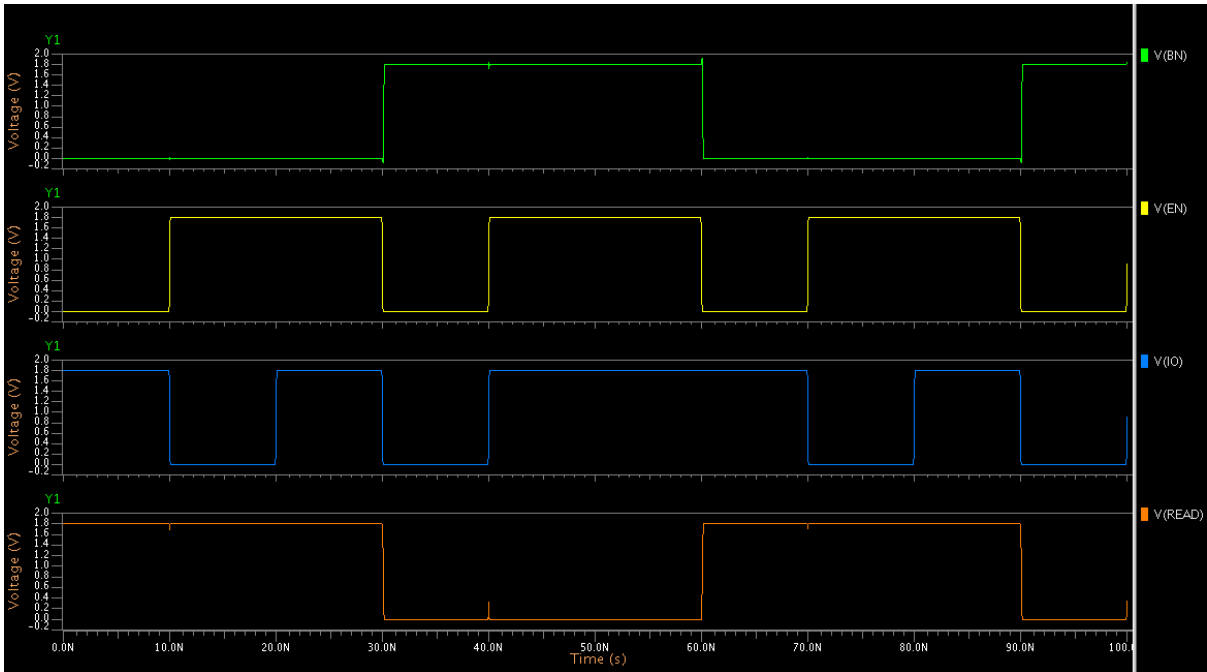


Figura 12: Formas de onda da simulação elétrica da célula básica. EN: Enable, Read: Bit escrito, IO: nível lógico na entrada da chave de passagem, BN: negação lógica de Read.

Conforme a figura 12 demonstra, o latch foi capaz de armazenar o bit escrito. No primeiro pulso nível lógico zero do enable, o bit 1 que estava na entrada foi armazenado no latch. No pulso seguinte o enable está no nível lógico 1 e o bit lido continua no nível lógico alto, demonstrando a capacidade do latch de armazenamento.

Após a simulação da célula básica de um bit, uma simulação da célula de um byte foi realizada para verificar a funcionalidade da lógica de controle. Devido ao atraso introduzido pela porta NAND existente na célula de um byte, é necessário manter a entrada no nível lógico desejado por dois ciclos. Por exemplo, se o período for de 10ns, a entrada tem de ser mantida por 20ns para salvar o bit no latch. Abaixo podemos ver as formas de onda obtidas:

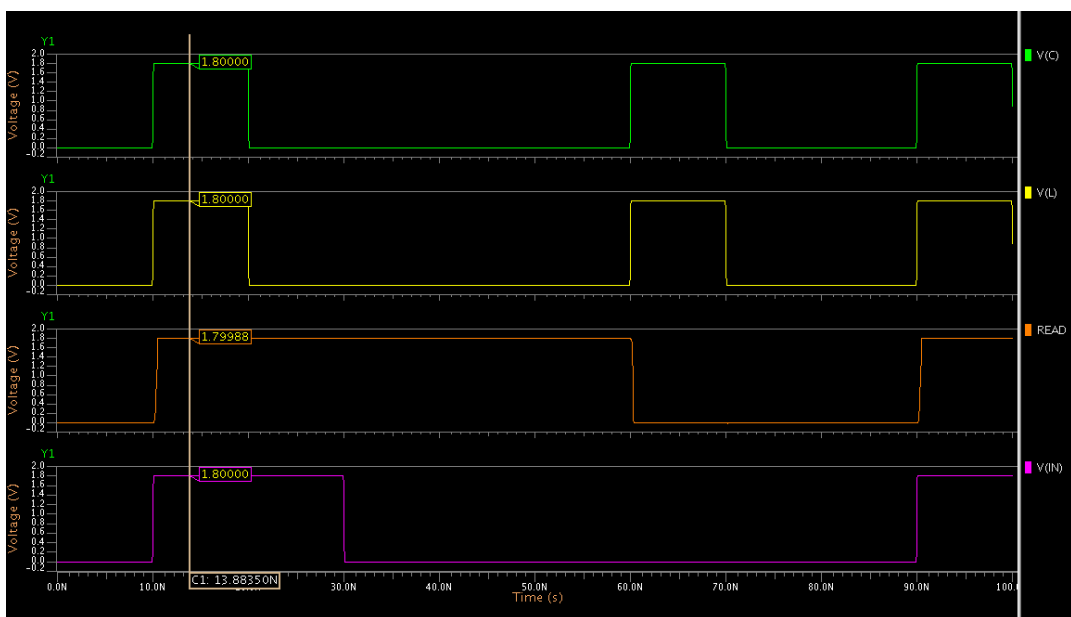


Figura 13: Formas de onda da simulação elétrica da célula de um byte. C: enable da coluna, L: enable da linha, Read: Bit escrito, IN: bit à ser escrito.

Conforme a figura 13 demonstra, o bit está sendo escrito e mantido no interior do byte. As formas de ondas dos outros bits não foram mostradas, pois são idênticas.

### 3.2. Circuito entrada/saída

Inicialmente, o circuito do buffer tri-state foi simulado. Quando o controle está no nível lógico 0, a entrada é passada para a saída. Quando o controle está no nível lógico 0, a saída fica em alta impedância. A figura 14 demonstra esse fato:

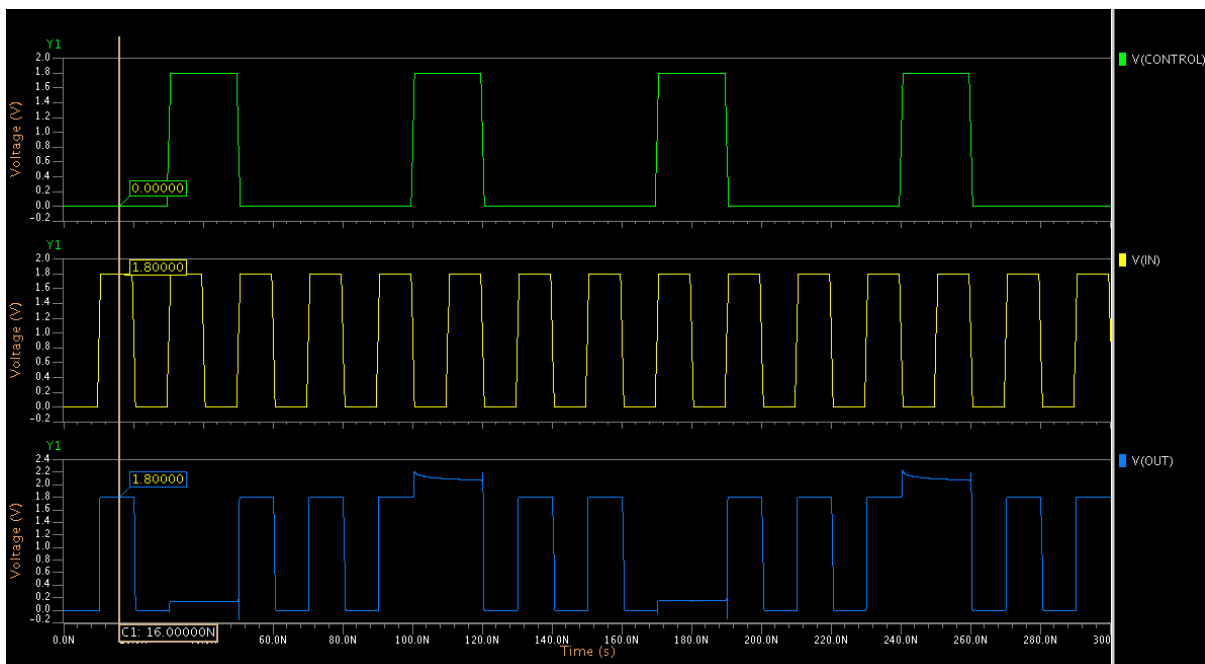


Figura 14: Formas de onda da simulação elétrica do buffer tri-state. Control: Bit que controla habilitação do buffer, IN: Entrada do buffer, OUT: Saída do buffer.

Após a simulação do buffer individual, um teste com o circuito completo foi realizado. Os dois buffers em paralelo foram simulados. Inicialmente a entrada foi colocada em OI, desse lado do buffer, a entrada é passada para a saída quando RorW está no nível lógico 1. Abaixo as formas de onda resultantes:

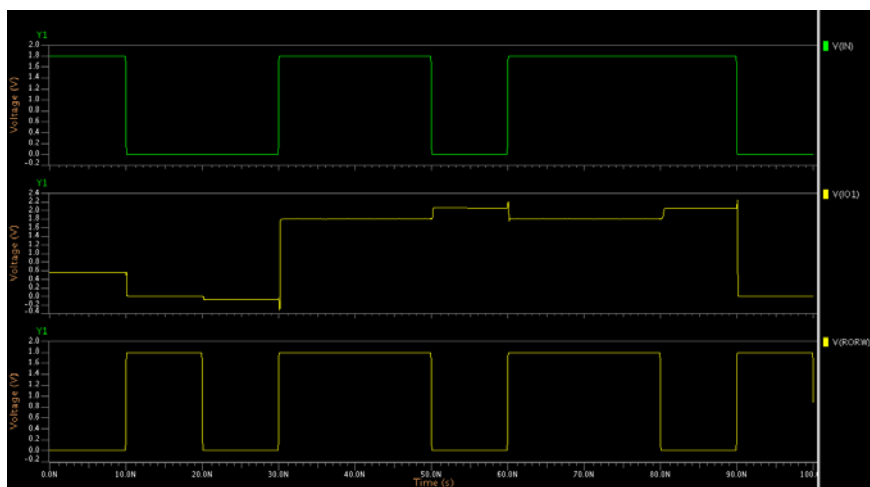


Figura 15: Formas de onda da simulação elétrica do circuito entrada/saída com entrada do lado OI. Control: Bit que controla habilitação do buffer, IN: Entrada do buffer, OUT: Saída do buffer.

Analogamente, para a entrada no lado IO, a entrada passa para a saída quando o RorW está no nível lógico 0. Como as formas de onda são análogas, elas não serão repetidas.

### 3.3. Decodificador

O circuito decodificador de três entradas para oito saídas foi testado uma única vez. Abaixo as formas de onda obtidas da simulação elétrica:

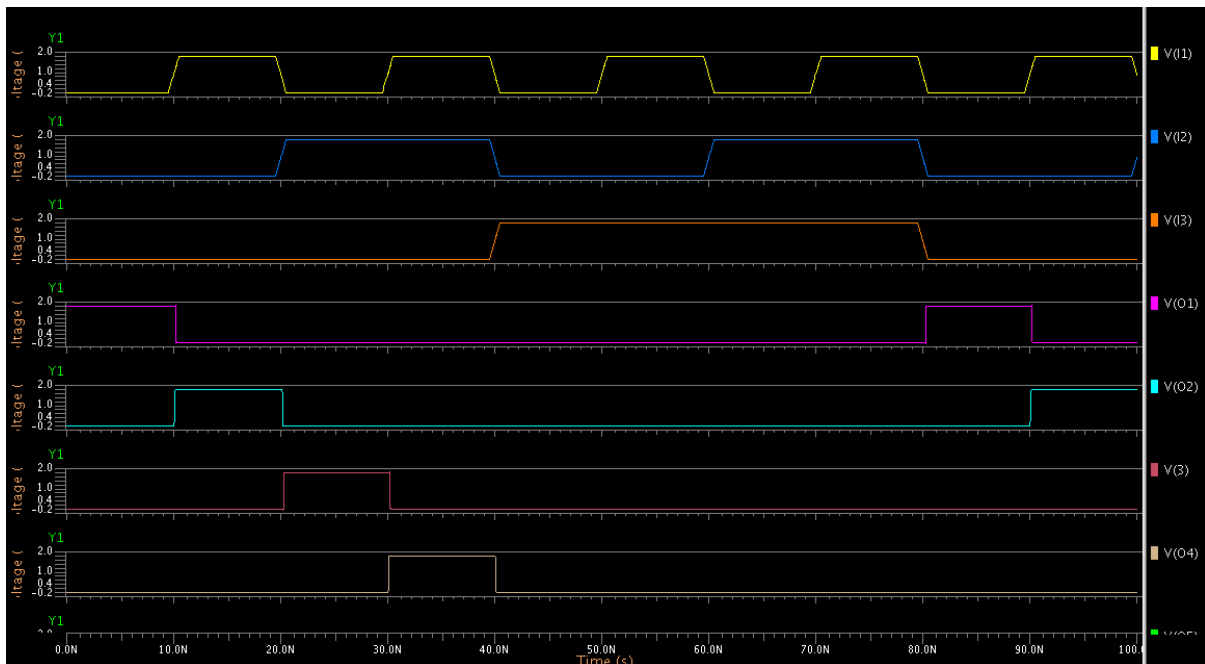


Figura 16: Formas de onda da simulação elétrica do decodificador. I1: bit menos significativo do endereço, I3: bit mais significativo do endereço, O1: saída 1, O2: saída 2, etc.

Como ficam constatadas pela figura 16, as saídas do decodificador são acionadas seqüencialmente conforme o endereço, ou seja, para entrada 000 a saída acionada é a O1 (Saída 1 do decodificador), conforme pode ser visto na figura 16.

### 3.4. Circuito completo da memória

Após a simulação do circuito completo, foi possível determinar o correto funcionamento do circuito. Além disso foi possível determinar os tempos de subida, descida, TPL e TPH. Abaixo se pode ver a forma de onda obtida para a gravação do endereço 1 da memória. O bit 1 foi gravado no primeiro ciclo e o bit 0 em  $t=30\text{ns}$ .

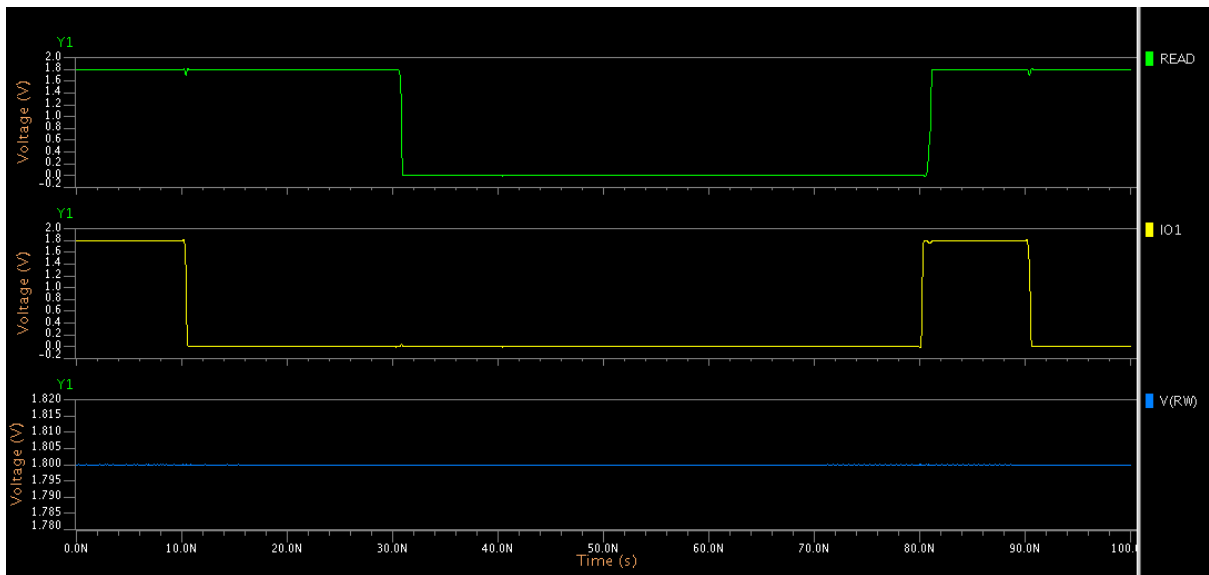


Figura 17: Formas de onda da simulação elétrica do circuito completo. Read: bit menos significativo do endereço, IO: entrada de escrita, RW: Seleção de leitura ou gravação, nível lógico 1=Gravação.

### 3.4.1. Parâmetros Do circuito

Após realizar a simulação completa do circuito foi possível determinar alguns parâmetros do circuito:

Parâmetro	Tempo (ns)
Tf	0,128
Tr	0,255
TPLH	0,778
TPHL	0,84

Tabela 1: Parâmetros do circuito.

Além dos parâmetros de tempo, foi obtido também a potência total dissipada pelo circuito durante a simulação:

Parametro	Potência (nW)
Power Dissipation	56,2621nW
Tempo de simulação:	100ns

Tabela 2: Potência dissipada pelo circuito.

## 4. Simulação Lógica

Além da simulação elétrica, uma simulação lógica do circuito foi realizada usando a linguagem verilog no software Mentor. Alguns problemas surgiram durante a simulação lógica e, devido as limitações do software, não foi possível simular o circuito completo, somente os subcircuitos foram simulados separadamente. Iremos analisá-los separadamente nas seções seguintes.

## 4.1. Célula básica

A simulação lógica da célula básica foi realizada separadamente, isso devido às limitações do software que não permitiram a simulação do latch com a chave de passagem. Abaixo as formas de onda do latch e da chave de passagem:

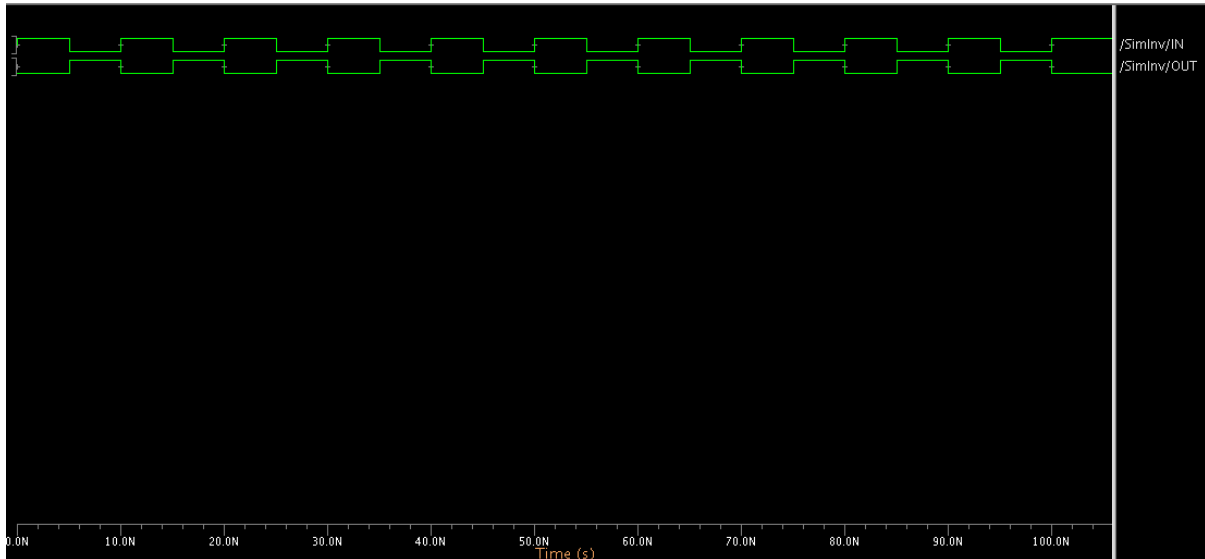


Figura 18: Formas de onda da simulação lógica do latch com inversores. IN: Entrada, OUT: Saída.

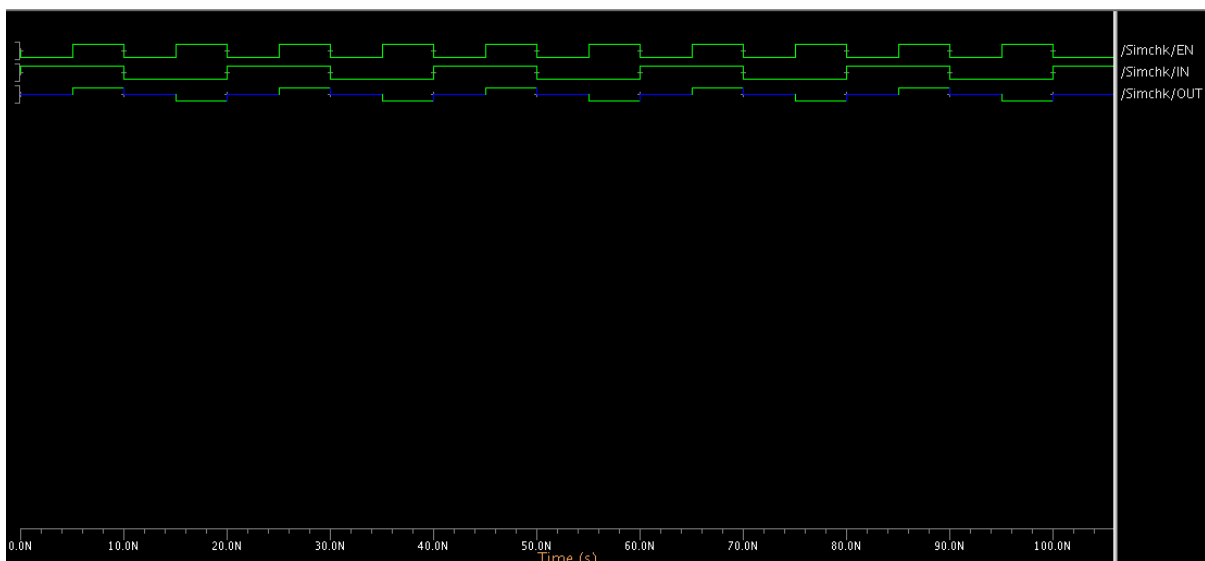


Figura 19: Formas de onda da simulação lógica da chave de passagem. IN: Entrada, OUT: Saída, EM: Enable.

Na figura 17 é possível ver que o latch está invertendo a entrada. Na figura 18 é possível verificar que quando o enable está no nível lógico alto, a entrada é passada para a saída, quando o enable está no nível lógico baixo, a saída fica em alta impedância.

## 4.2. Circuito Entrada/Saída

A simulação lógica do circuito entrada/saída foi realizada com o circuito completo, ou seja, com os 8 pares de buffers tri-state em antiparalelo. Abaixo as formas de onda obtidas:

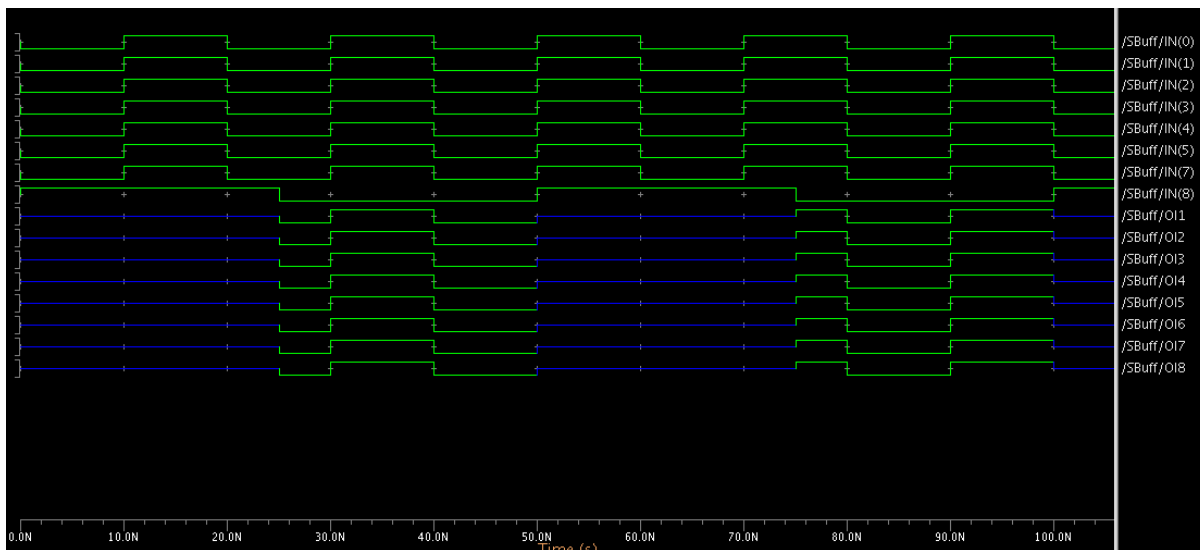


Figura 20: Formas de onda da simulação lógica do circuito entrada/saída. IN1...IN7: Entradas IO do buffer, IN8: RorW, OI1...OI8: saídas OI do buffer.

O circuito de entrada/saída está de acordo com o projeto, pois quando o RorW está no nível lógico alto, a saída fica em alta impedância, quando RorW está em nível lógico baixo, a entrada é transferida para a saída.

## 4.3. Decodificador

A simulação lógica do decodificador também foi realizada com o circuito completo. Abaixo as formas de onda obtidas:

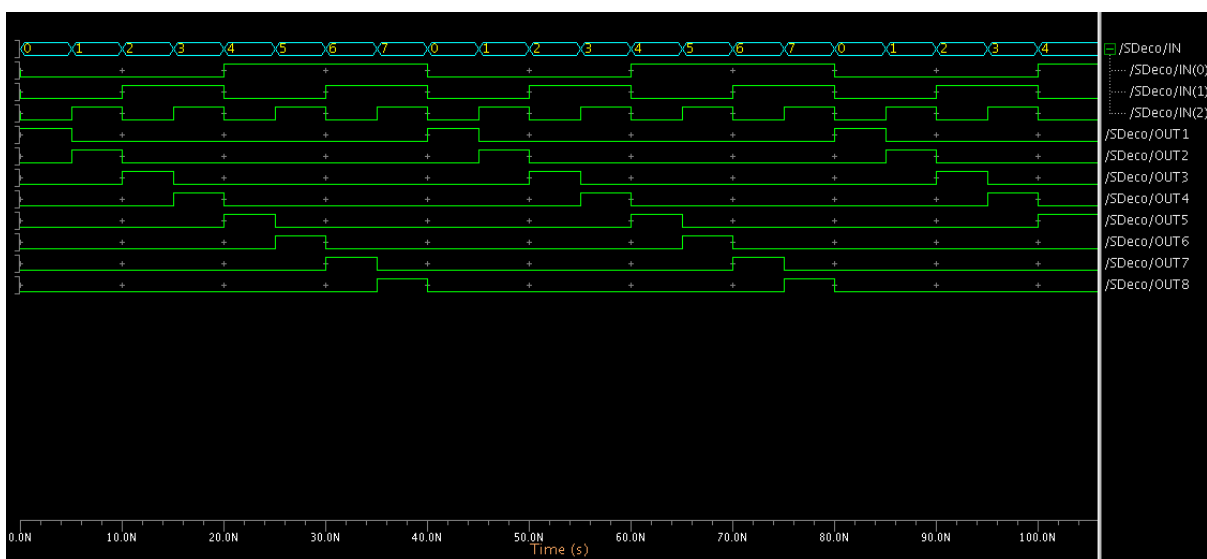


Figura 21: Formas de onda da simulação lógica do Decodificador. IN0...IN2: Endereços de entrada, OUT1...OUT8: Saídas do decodificador.

É possível verificar na figura 20 que o decodificador está funcionando conforme projetado, começando pelo endereço 000 a saída ativada é a saída OUT1 e assim sucessivamente.

## 5. Layout

O layout do circuito foi projetado a partir das portas lógicas, da chave de passagem e do buffer tri-state. Abaixo vemos os layouts das portas lógicas que foram obtidos de maneira automática a partir do software Mentor. Abaixo vemos os layouts de todas as portas lógicas, da chave de passagem e do buffer tri-state. Estes são os blocos fundamentais para a construção do restante do layout.

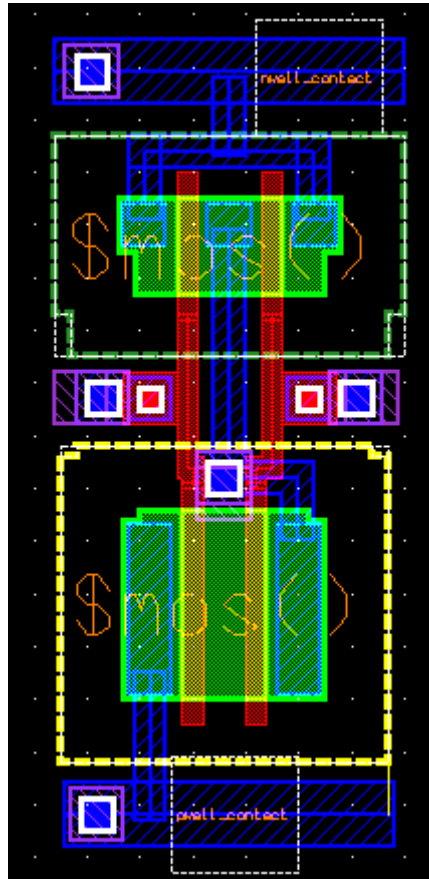


Figura 22: Layout Porta NAND



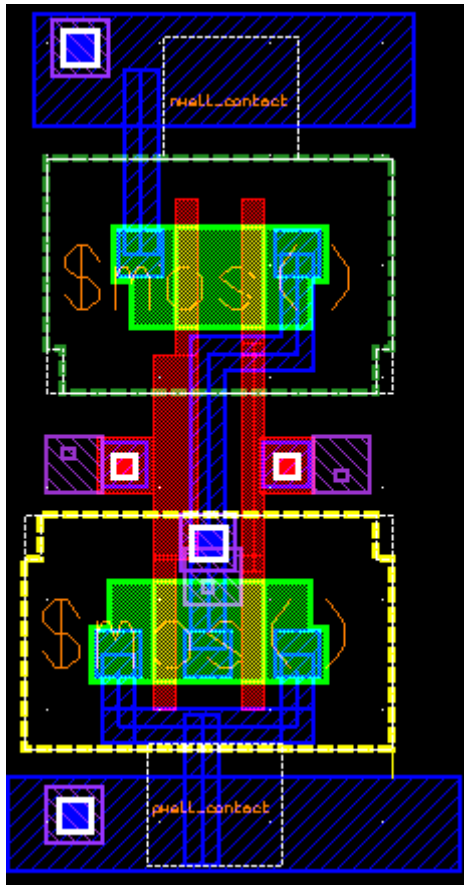


Figura 23: Layout Porta NOR

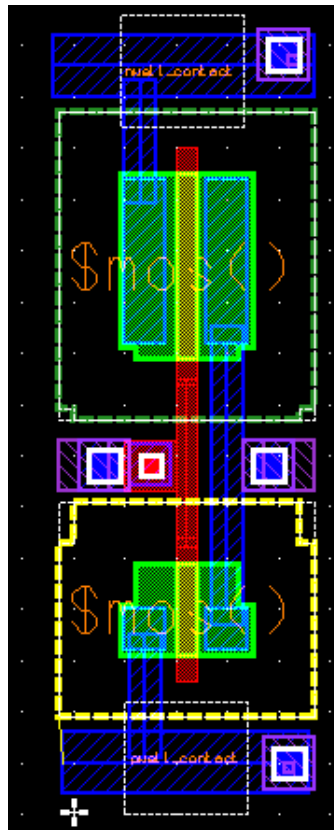


Figura 24: Layout Porta NOT.

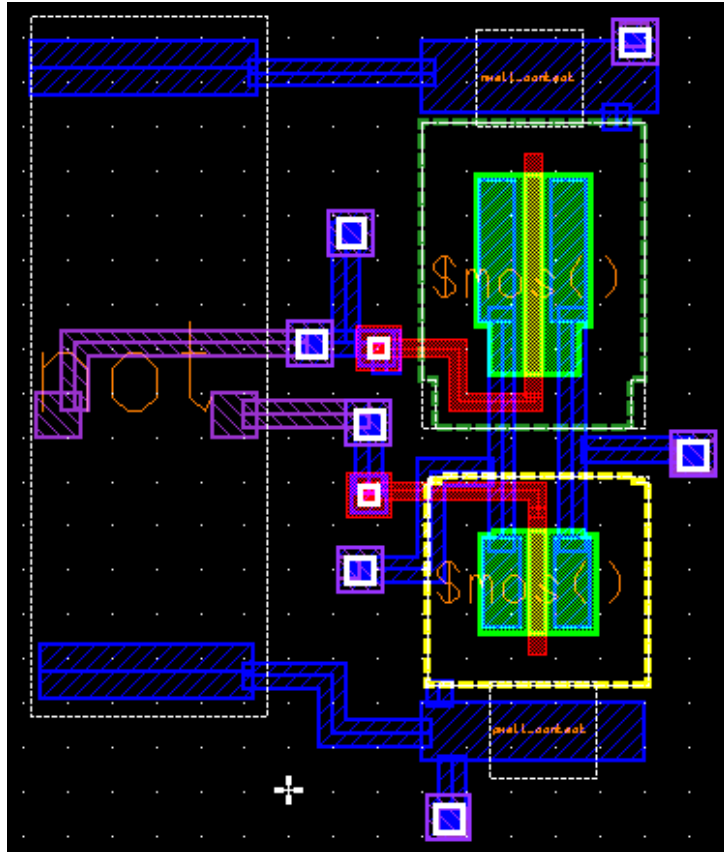


Figura 25: Layout Chave de passagem.

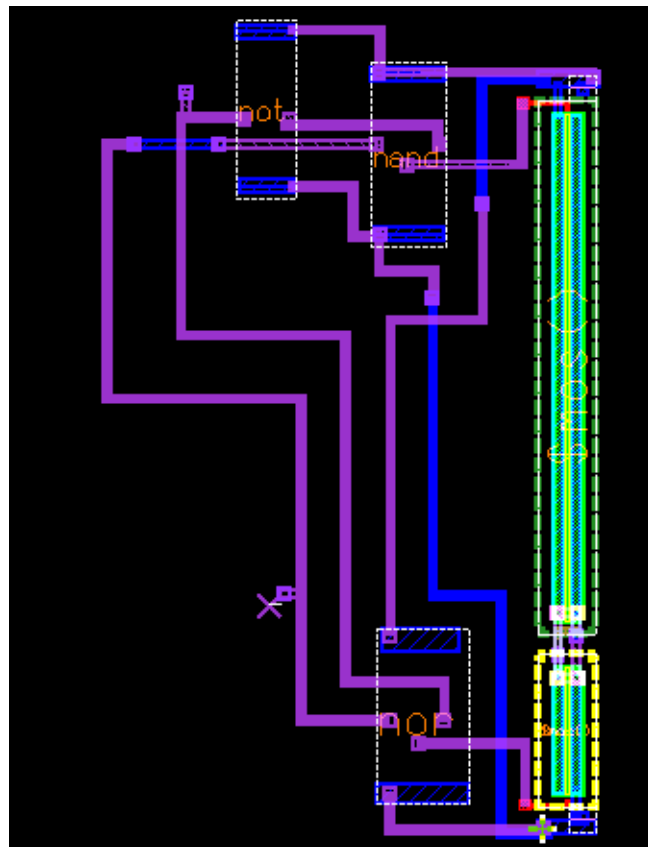


Figura 26: Layout buffer.

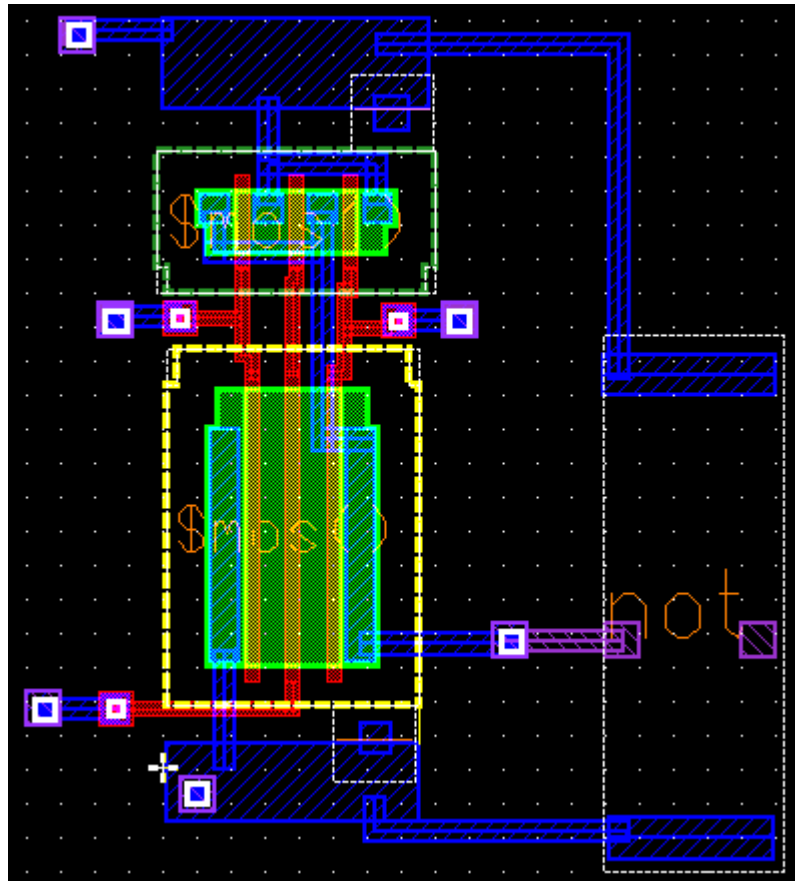


Figura 27: Layout AND de três entradas.

Com base nesses blocos fundamentais, o restante dos circuitos foi desenhado. Abaixo vemos os seguintes circuitos: Decodificador, Circuito entrada/saída (Buffers tri-state), célula de um bit, célula de um byte e circuito completo.

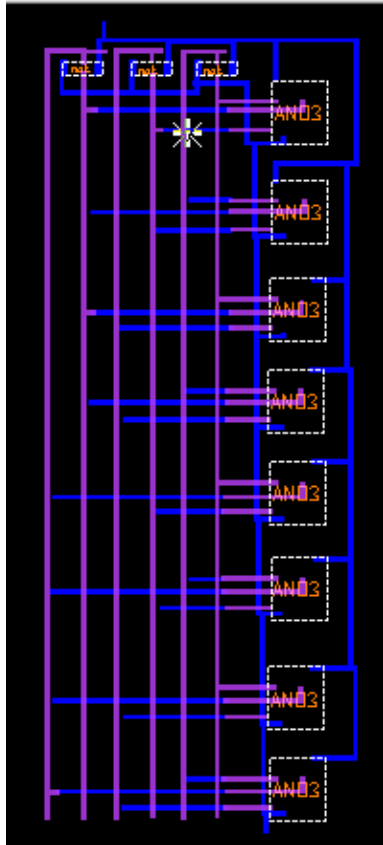


Figura 28: Layout decodificador.



Figura 29: Layout circuito entrada/saída.

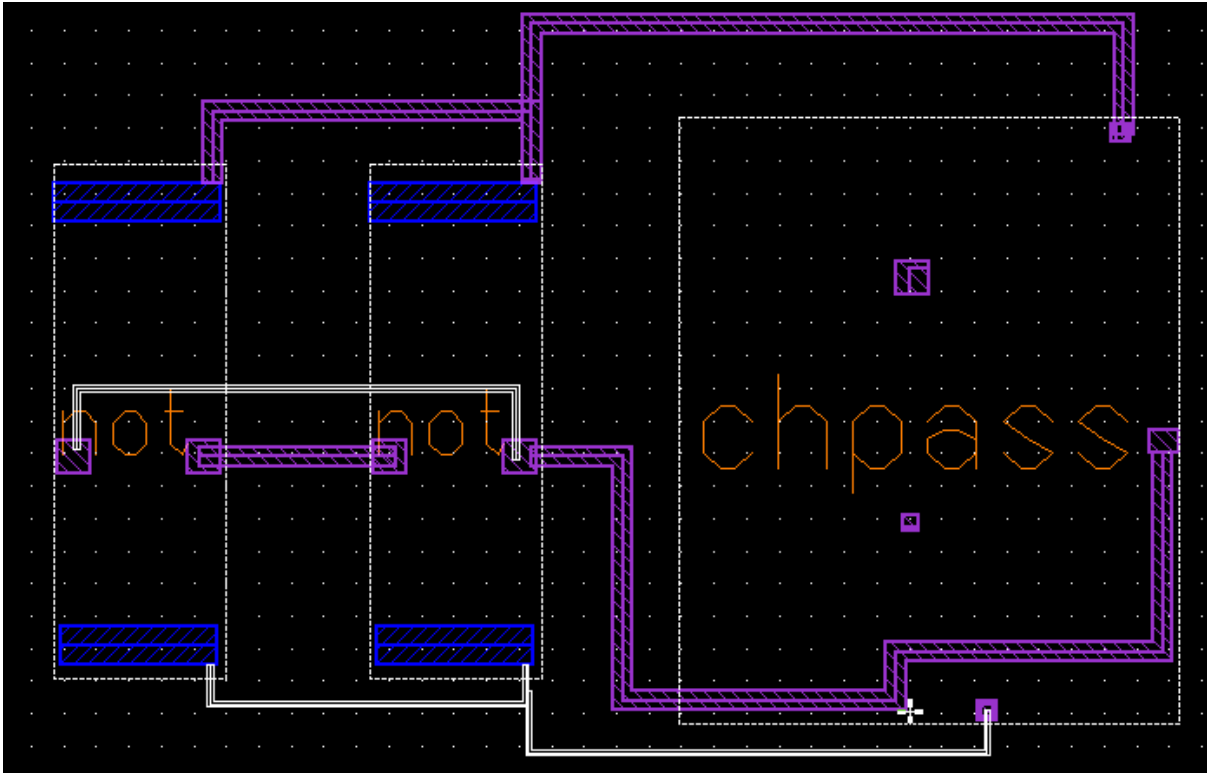


Figura 30: Layout circuito bit.

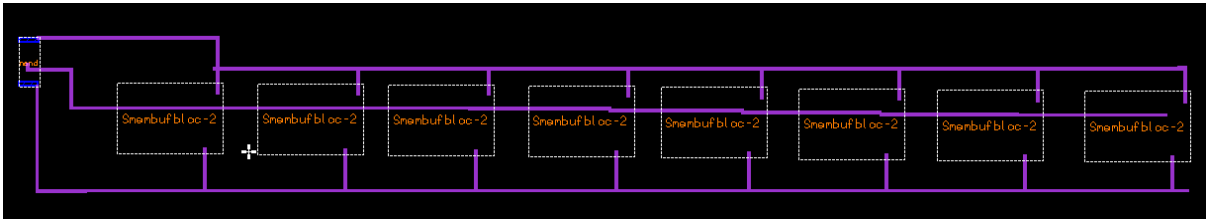


Figura 31: Layout circuito byte.

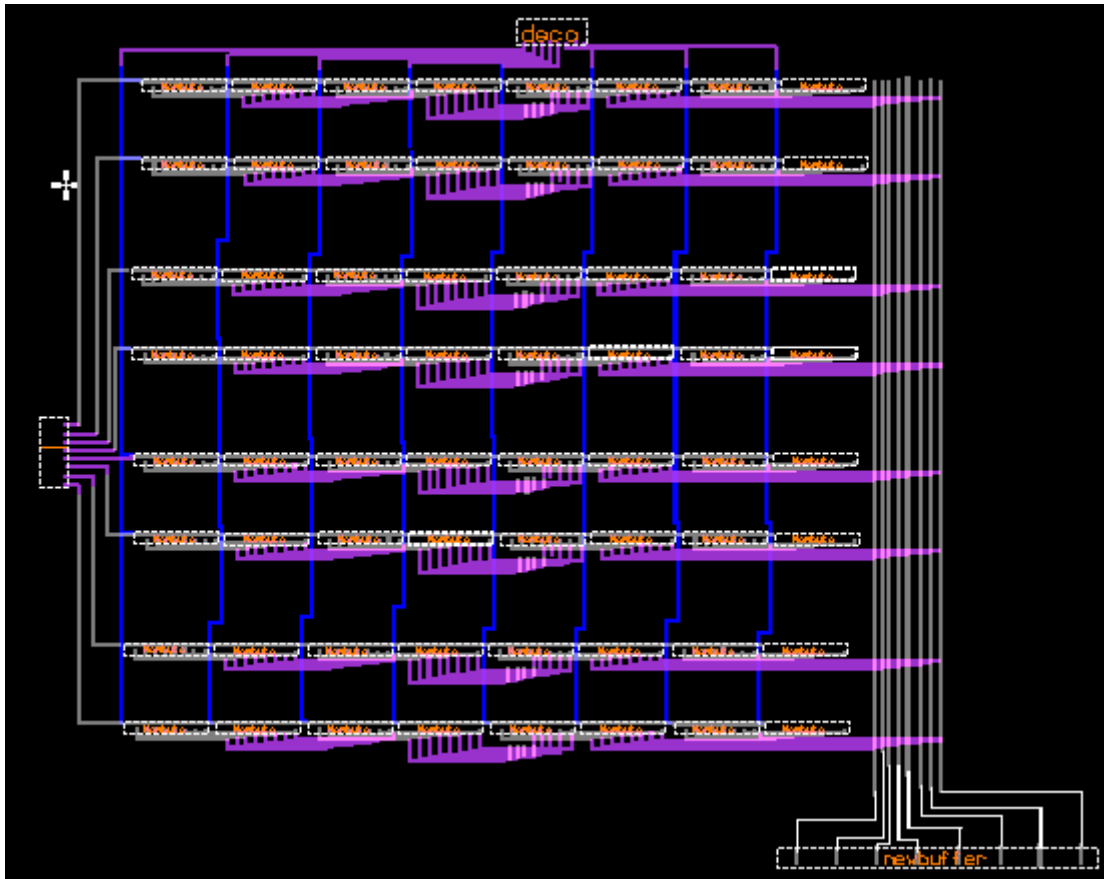


Figura 32: Layout completo da memória

## **6. Conclusão**

As memórias SRAM são fundamentais para a operação dos processadores atuais. Um dos maiores problemas do projeto é a limitada miniaturização do circuito, pode-se ver que as células básicas ocupam seis transistores por bit, enquanto a DRAM ocupa somente um transistor e um capacitor. Conseqüentemente à limitada miniaturização, segue-se o custo da memória, por ocupar mais espaço e possuir mais transistores que a memória DRAM, seu custo é elevado quando comparado à memória DRAM.

A partir da análise do circuito, pode-se concluir que a memória SRAM é muito rápida, o tempo de resposta para uma escrita de bit está em torno de 0,8 ns. Isso demonstra a grande velocidade da memória SRAM.

## 7. Bibliografia

1. **The MOSIS Service.** Rules for the MOSIS Scalable CMOS (SCMOS)  
<http://www.mosis.com/Technical/Designrules/scmos/scmosmain.html>
2. **Página da disciplina.** Circuitos Integrados Digitais:  
<http://www.eletr.ufpr.br/marlio/cidigital/cidigital.htm>
3. **Página da disciplina de Memórias.** Memórias <http://www.eletr.ufpr.br/marlio/te159/te159.htm>.