

Design e simulação de layouts dos circuitos meio-somador e somador-completo através das ferramentas *Magic* e *Microwind*

Pedro A. Tavares, UCPEL
E-mail: pedro.tavares@sou.ucpel.edu.br

Resumo—Este trabalho referente à disciplina de Introdução ao Projeto de Circuitos Integrados do curso de Mestrado em Engenharia Eletrônica e Computação da UCPEL se propõe a apresentar layouts e simulações para o circuito meio-somador de 1 bit utilizando as ferramentas *Magic* e *Microwind* (em conjunto com o software DSCH2). Também é mostrado o layout do circuito somador completo mas sem simulações, apenas para fins de comparação.

Palavras-Chave—half-adder, meio somador, full-adder, somador completo, layout, magic, microwind, CMOS, microeletrônica.

1 Introdução

Este trabalho tem como objetivo específico demonstrar o desenvolvimento de um layout para o circuito de meio-somador, amplamente utilizado em eletrônica digital. Foram feitos dois layouts, um desenhado inteiramente na ferramenta *Magic*, onde é possível exportar os arquivos de simulação *spice*. E o outro foi gerado automaticamente, através da compilação de um arquivo Verilog no software *Microwind*, onde também é possível fazer simulações alterando algumas características do circuito.

Este artigo foi estruturado conforme o enunciado do trabalho da disciplina. Na seção 2 é mostrado o layout a nível de transistores do circuito somador completo CMOS com redes *pull-up* e *pull-down*. Também é demonstrado o layout simbólico gerado no software *Microwind*. Na seção 3 são apresentados o layout do circuito meio-somador implementado na ferramenta *Magic* e as simulações executadas no software *SpiceOpus*. Na seção 4 é apresentado o circuito meio-somador gerado pelo software *Microwind* e alguns gráficos de simulações feitos nesta mesma ferramenta.

2 Layout do circuito somador completo

O circuito somador completo binário, soma dois bits de entrada e também adiciona ao cálculo o bit de *carry* da soma anterior. Uma das diversas maneiras que ele pode ser construído é através de duas portas XOR, duas ANDs e uma OR, como no esquemático da figura 1.

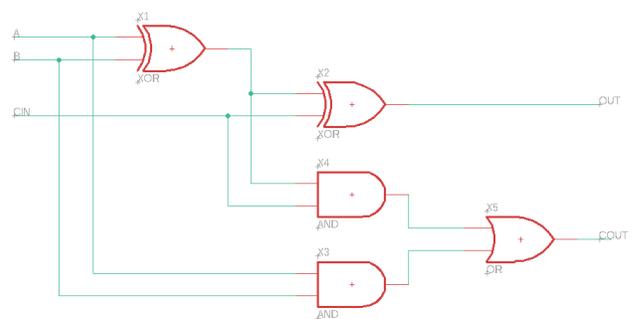


Figura 1: Esquemático do somador completo

A tabela-verdade do somador completo de 1-bit é mostrado na tabela 1, sendo que a entrada C é o *carry* de entrada:

Através do esquemático de portas lógicas é possível construir o circuito em nível de transistores de tecnologia CMOS. Existem diversas maneiras de implementar tal circuito e, para este trabalho, utilizou-se a lógica complementar (C-

A	B	C	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Tabela 1: Tabela verdade do somador completo de 1 bit

CMOS) baseada em [1], [2] e [3]. O circuito é mostrado na figura 2.

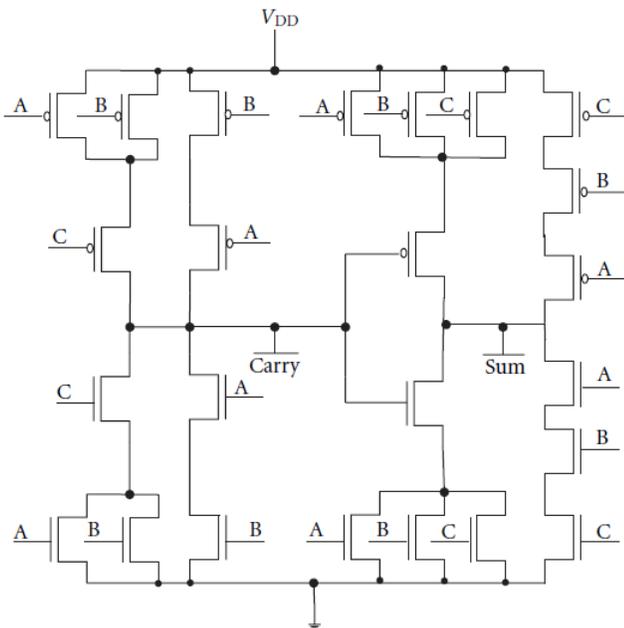


Figura 2: Layout CMOS do somador completo

Para montar o layout simbólico do circuito de somador completo, foi utilizado primeiramente a ferramenta DSCH2 que permite a implementação e simulação de circuitos em nível de portas lógicas. Foi criado um somador completo utilizando a ferramenta, conforme mostrado no esquemático da figura 3.

A partir dessa ferramenta é exportado um arquivo contendo a descrição do hardware em linguagem Verilog (figura 4). Esse arquivo pode então ser utilizado no software *Microwind*, que compila o arquivo Verilog e gera automaticamente o layout do somador completo, conforme mostrado na figura 5.

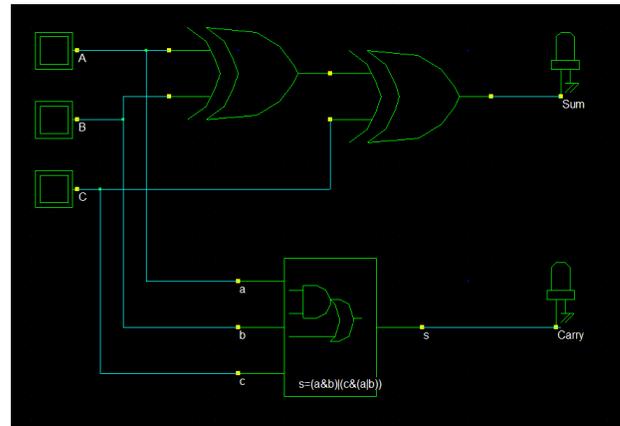


Figura 3: Esquemático do somador completo na ferramenta DSCH2

```
// DSCH 2.7a
// 29/07/2020 23:57:24
// By C:\Users\Pedro\dev\dSCH2\full_adder.sch

module full_adder( C,B,A,Carry,Sum);
  input C,B,A;
  output Carry,Sum;
  xor #(16) xor2(w4,A,B);
  assign Carry=(A&B)|(C&(A|B));
  xor #(16) xor2(Sum,w4,C);
endmodule

// Simulation parameters in Verilog Format
always
#1000 C=~C;
#2000 B=~B;
#4000 A=~A;

// Simulation parameters
// C CLK 10 10
// B CLK 20 20
// A CLK 40 40
```

Figura 4: Arquivo Verilog do somador completo

3 Layout do circuito meio somador utilizando a ferramenta Magic

O circuito meio somador também calcula a soma de dois bits mas de forma mais simples pois não considera o bit de *carry* como entrada. A figura 6 mostra o circuito lógico e a tabela 2 a representação da tabela-verdade do meio-somador.

Para desenhar o layout na ferramenta *Magic* é necessário representar o circuito em nível de transistores (CMOS). Essa representação com redes *pull-up* e *pull-down* é mostrada na figura 7.

A partir desta representação foi desenhado todo o layout do circuito impresso no *Magic*, utilizando

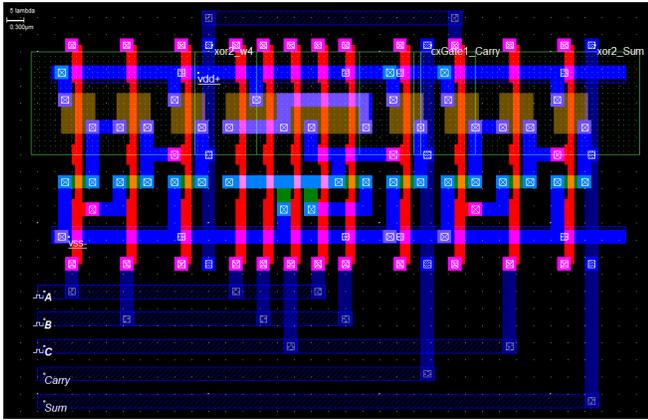


Figura 5: Layout do somador completo gerado pelo Microwind

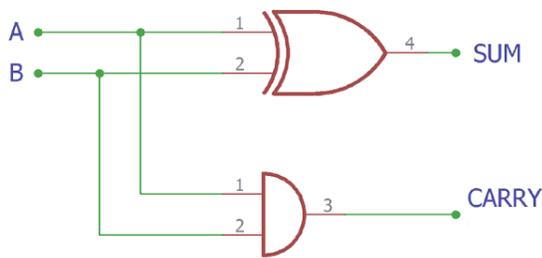


Figura 6: Esquemático do meio somador

A	B	Carry	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Tabela 2: Tabela-verdade do meio somador

os componentes e *layers* da biblioteca *scmos.lib*. Para carregar a biblioteca deve-se passar essa informação como parâmetro na linha de comando ao executar o *Magic* como no exemplo: (`$>magic -T scmos`). Desta maneira o *software* sabe que a tecnologia utilizada para o design do circuito será o sCMOS (*scientific Complementary metal-oxide-semiconductor*). O layout final é mostrado na figura 8.

Depois da montagem do layout, para realizar as simulações foi gerado o arquivo `".spice"` através da execução dos comandos `ext` e `ext2spice`, dentro do console do *Magic*.

A partir desse arquivo que descreve o circuito e, utilizando a biblioteca *amisc5n* que contém as definições da tecnologia, foi possível simular o comportamento do circuito através do software

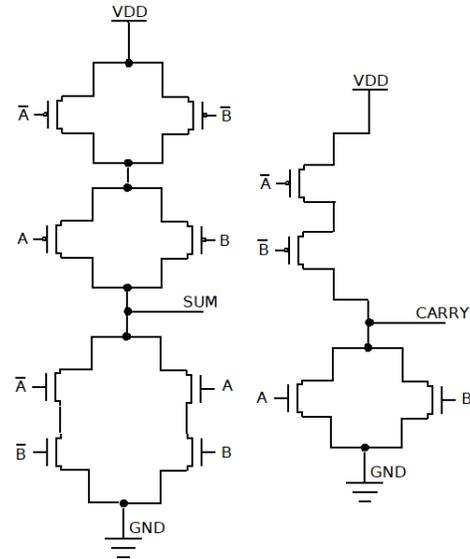


Figura 7: Layout CMOS do meio somador

SpiceOpus.

O sub-circuito do modelo *spice* está representado na figura 9.

Utilizando o sub-circuito, foi criado o arquivo de simulação mostrado na figura 10 para plotar os gráficos de saída através do software *SpiceOpus*.

Os gráficos 11 e 12 mostram sucessivamente a simulação do circuito sem carga e com carga capacitiva de 800fF.

Já os gráficos a seguir mostram em uma escala de tempo mais reduzida, ou seja, com maior zoom, [13] o circuito sem carga, [14] o circuito com carga capacitiva de 800fF, [15] o circuito com a mesma carga capacitiva mas com largura $W = 18$ e, finalmente, o gráfico [16] que mostra o consumo do circuito.

4 Layout do circuito meio somador utilizando as ferramentas DSCH2 e Microwind

Similarmente ao que se foi feito na seção 2 para o somador completo, aqui foi gerado o layout do meio somador a partir da compilação do arquivo Verilog exportado pelo software *DSCH2*.

Primeiramente foi montado o circuito no *DSCH2* conforme o esquemático mostrado na figura 17.

Exportou-se o arquivo Verilog (figura 18) que, quando compilado no *Microwind*, gerou o layout mostrado na figura 19.

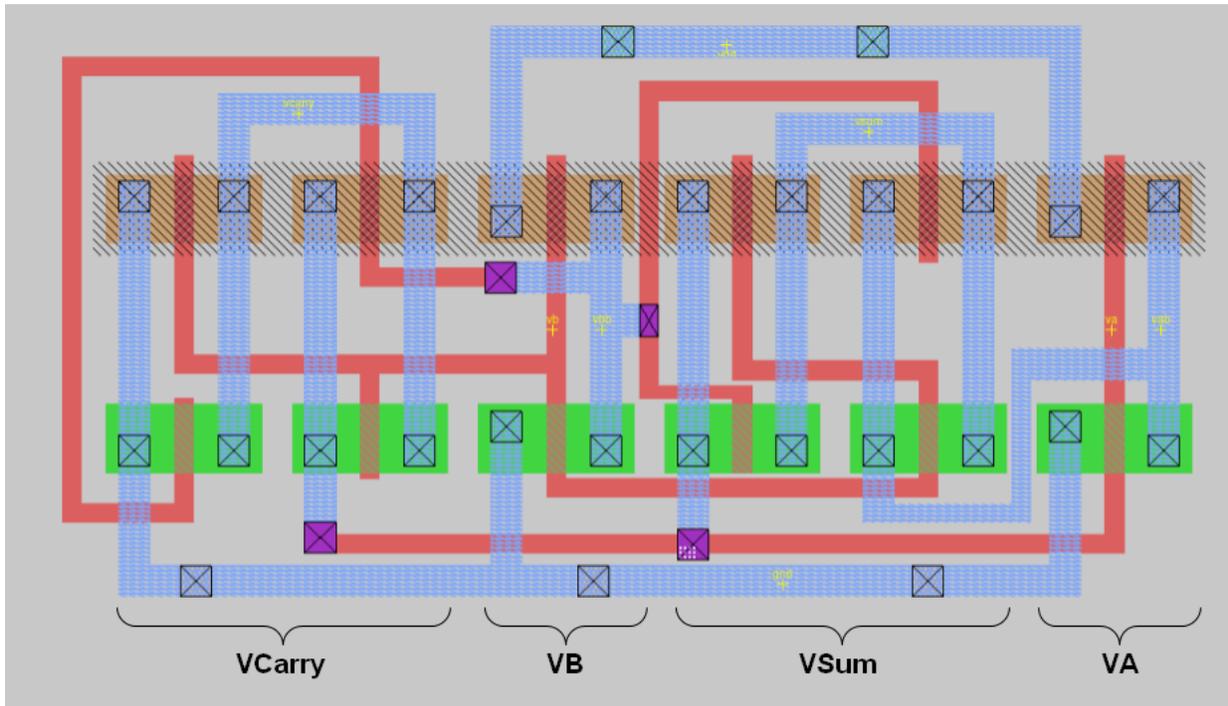


Figura 8: Layout do meio somador na ferramenta *Magic*

```
* SPICE3 file created from halfadder_layout.ext - technology: scmos
.option scale=1u
.SUBCKT ha va vb vcarry vsum vdd Gnd

M1000 vcarry vbb va w_n49_n1 cmosp w=11 l=3 ad=242 pd=88 as=242 ps=88
M1001 vsum vbb vab w_n49_n1 cmosp w=11 l=3 ad=242 pd=88 as=242 ps=88
M1002 vsum vb vab Gnd cmosn w=11 l=3 ad=242 pd=88 as=242 ps=88
M1003 vcarry vb va Gnd cmosn w=11 l=3 ad=242 pd=88 as=242 ps=88
M1004 vab va gnd Gnd cmosn w=11 l=3 ad=0 pd=0 as=363 ps=132
M1005 vab va vdd w_n49_n1 cmosp w=11 l=3 ad=0 pd=0 as=242 ps=88
M1006 vcarry vbb gnd Gnd cmosn w=11 l=3 ad=0 pd=0 as=0 ps=0
M1007 vsum vb va w_n49_n1 cmosp w=11 l=3 ad=0 pd=0 as=0 ps=0
M1008 vcarry vb gnd w_n49_n1 cmosp w=11 l=3 ad=0 pd=0 as=196 ps=104
M1009 vbb vb gnd Gnd cmosn w=11 l=3 ad=121 pd=44 as=0 ps=0
M1010 vsum vbb va Gnd cmosn w=11 l=3 ad=0 pd=0 as=0 ps=0
M1011 vbb vb vdd w_n49_n1 cmosp w=11 l=3 ad=121 pd=44 as=0 ps=0

C0 vbb w_n49_n1 7.80fF
C1 va w_n49_n1 2.53fF
C2 vb w_n49_n1 4.76fF
C3 vab Gnd 18.75fF
C4 vsum Gnd 19.50fF
C5 va Gnd 62.03fF
C6 vcarry Gnd 19.50fF
C7 gnd Gnd 47.00fF
C8 vb Gnd 65.08fF
C9 vbb Gnd 92.02fF
.ENDS ha
```

Figura 9: Arquivo do sub-circuito no modelo *spice*

As figuras 20 e 21 mostram o comportamento do circuito gerado, confirmando seu correto funcionamento através da comparação com sua tabela-verdade 2. Foi utilizada a escala de tempo em 5ns para mostrar os efeitos dos atrasos nas trocas de estado nas saídas do circuito, principalmente na

saída SUM.

5 Considerações Finais

Neste trabalho foi explicado como desenvolver e simular os layouts de circuitos somadores através das ferramentas *Magic*, *DSCH2*, *Microwind* e *SpiceOpus*. Além disso, também foi demonstrado de maneira gráfica como fatores como a capacitância de carga e largura dos barramentos dos transistores CMOS podem afetar o tempo de resposta na saída destes circuitos.

Em trabalhos futuros sugere-se melhorar o layout do meio-somador desenhado na ferramenta *Magic* com o objetivo de reduzir os ruídos apresentados nos gráficos. Isso é possível aplicando-se boas práticas para design de layouts, como manter distâncias mínimas entre canais e utilizar larguras mínimas ao longo dos barramentos.

Referências

- [1] S.-M.Kang and Y. Leblebici. Cmos digital integrated circuits: Analysis and design. *Tata McGraw-Hill, New York, NY, USA*, 2003.
- [2] N. Weste and D. Harris. Cmos vlsi design. *Pearson Wesley*, 2005.
- [3] S. R. Whitaker D. Radhakrishnan and G. K. Maki. Formal design procedures for pass-transistor switching circuits. *IEEE Journal of Solid-State Circuits*, 20:531–536, 1984.

```

Titulo: Meio Somador de 1 bit
* -----
* Includes
.include amis_c5n.txt
.include ha.spice
* -----
* Fontes de alimentação e entradas

v1 gnd 0 dc 0
v2 vdd 0 dc 5
V3 va 0 pulse=(5 0 0 1p 1p 1u 2u)
V4 vb 0 pulse=(5 0 0 1p 1p 2u 4u)
* -----
* Saídas (Capacitor de saída)

c1 vsum gnd 800ff
* -----
* Declaração do circuito

x1 va vb vcarry vsum vdd Gnd ha
* -----
* Tipo de simulação

.control
tran 1n 16u
plot va vb+6 vsum+18 vcarry+24
.endc
.end

```

Figura 10: Arquivo de simulação do *SpiceOpus*

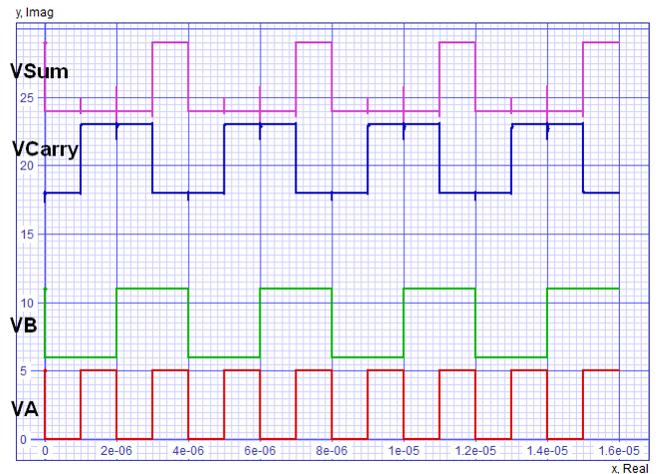


Figura 12: Simulação do meio-somador com carga capacitiva de 800fF

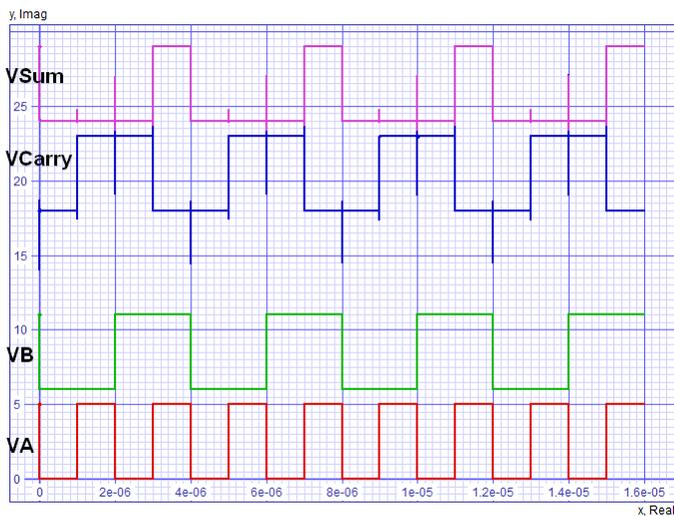


Figura 11: Simulação do meio-somador sem carga

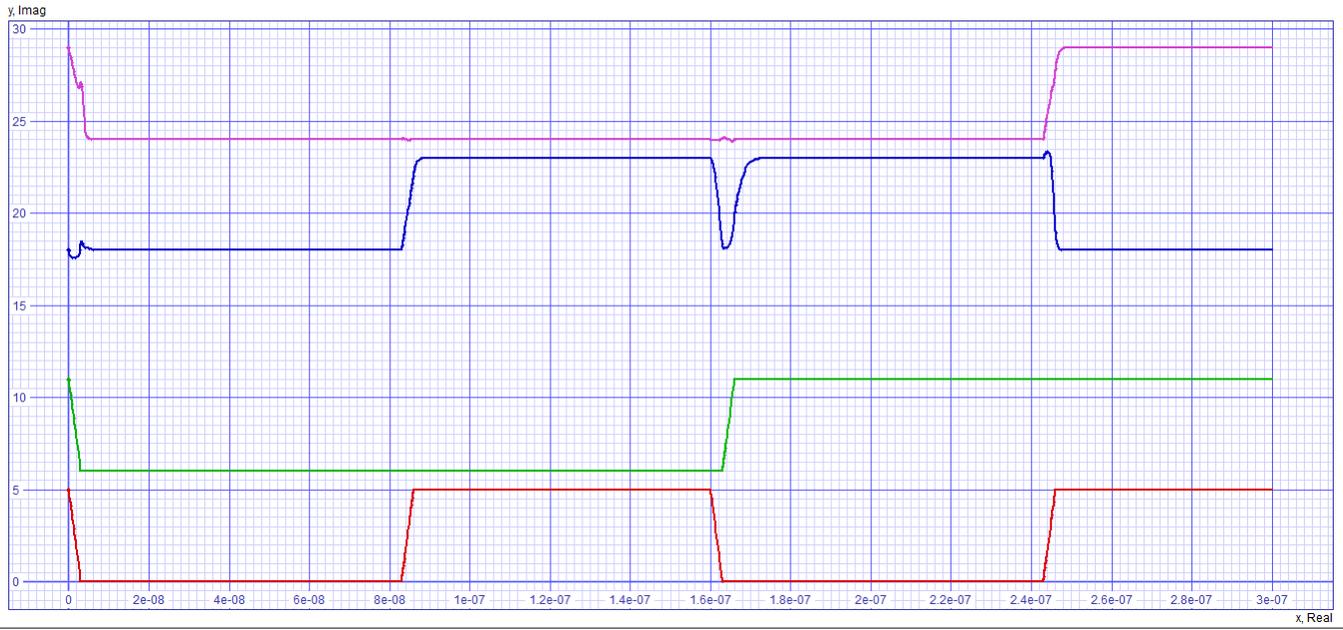


Figura 13: Meio-somador sem carga



Figura 14: Meio-somador com carga capacitiva de 800fF



Figura 15: Meio-somador com carga capacitiva de 800fF e $W=18 \lambda$

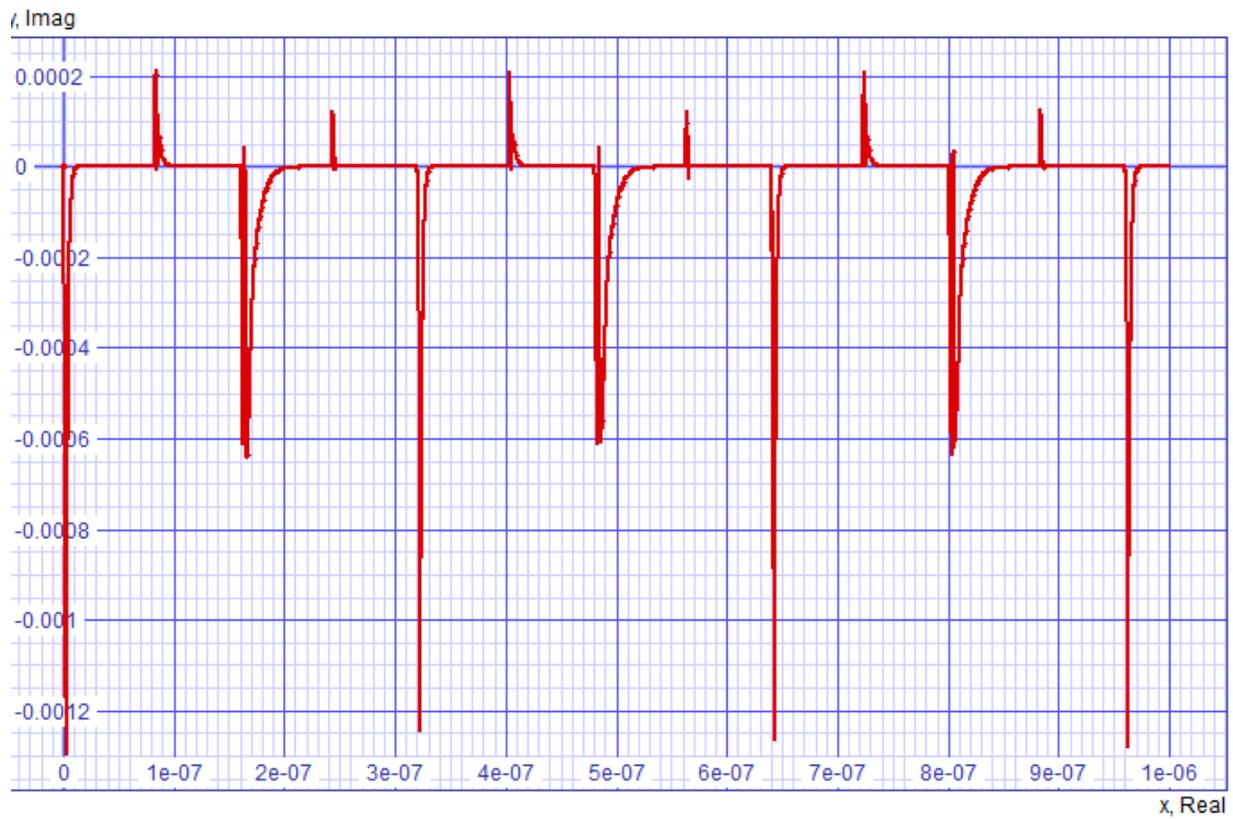


Figura 16: Consumo do circuito meio-somador

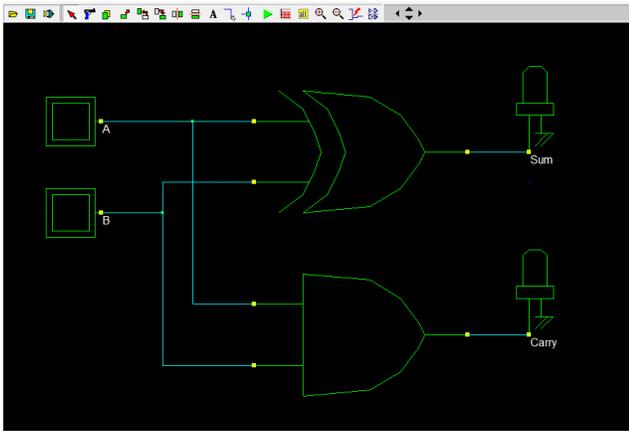


Figura 17: Esquemático do meio somador na ferramenta DSCH2

```
// DSCH 2.7a
// 30/07/2020 01:24:21
// C:\Users\Pedro\dev\dsch2\half_adder.sch

module half_adder( B,A,Carry,Sum);
  input B,A;
  output Carry,Sum;
  and #(16) and2(Carry,B,A);
  xor #(16) xor2(Sum,A,B);
endmodule

// Simulation parameters in Verilog Format
always
#1000 B=~B;
#2000 A=~A;

// Simulation parameters
// B CLK 10 10
// A CLK 20 20
```

Figura 18: Arquivo Verilog do meio somador

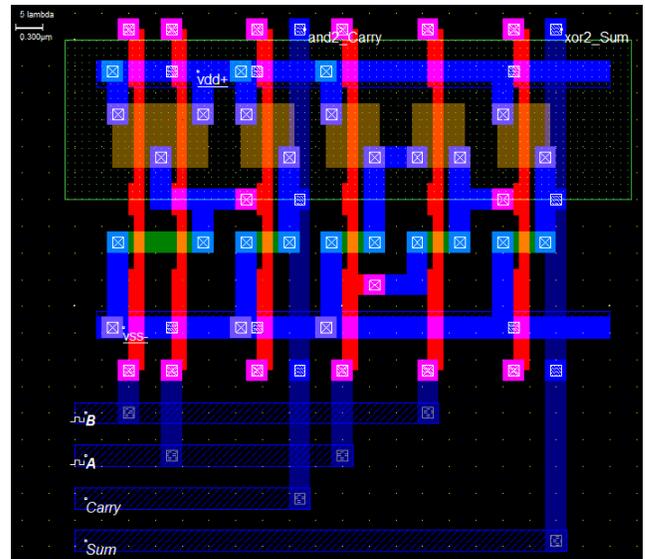


Figura 19: Layout do meio somador gerado pelo Microwind

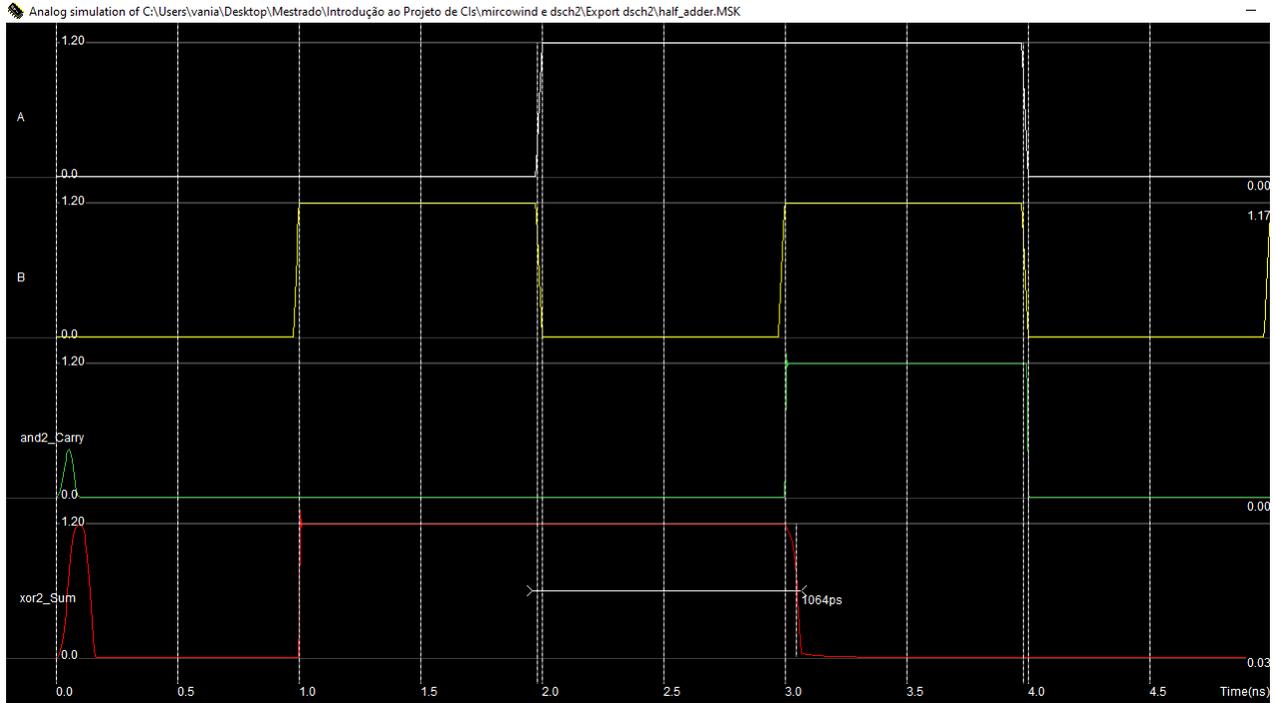


Figura 20: Simulação do meio somador no Microwind

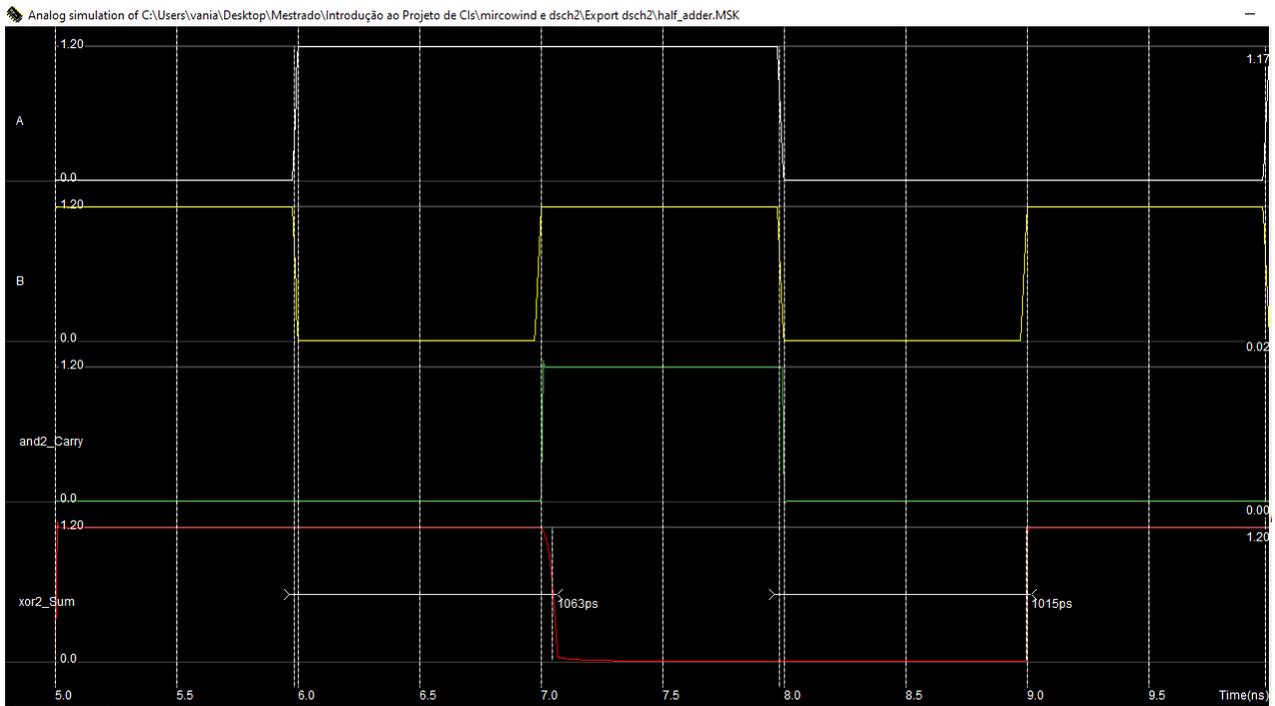


Figura 21: Simulação do meio somador no Microwind (cont.)