

Universidade de São Paulo

Instituto de Ciências Matemáticas e Computação de São Carlos

SSC117 – Introdução à Lógica Digital

Relatório de desenvolvimento da sub prática 14

Docente:

Onofre Trindade Junior

Discentes:

Gustavo Mazzo 7238940 A

Wesley Tiozzo 8077925 A

1. Definição do Problema

Foi iniciado o desenvolvimento de uma ULA de 4 bits com somador, subtrator e complemento de 1 de 4 bits.

2. Descrição das Soluções

Tabela verdade para o somador

| SOMADOR | | | | | |
|---------|---|-----|---|------|---|
| A | B | CIN | S | COUT | |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Mapa de Karnaugh do somador com saída

| SOMADOR | | | | | |
|---------|----|----|---|----|--|
| | A' | A' | A | A | |
| CIN' | 0 | 1 | 0 | 1 | |
| CIN | 1 | 0 | 1 | 0 | |
| | B' | B | B | B' | |

Mapa de Karnaugh para o somador com a saída e o carry out

| SOMADOR | | | | | |
|---------|----|----|---|----|--|
| | A' | A' | A | A | |
| CIN' | 0 | 0 | 1 | 0 | |
| CIN | 0 | 1 | 1 | 1 | |
| | B' | B | B | B' | |

Equação do circuito do somador com saída

Resultado = $A*B'*\text{Carry_In}' + A'*B*\text{Carry_In}' + A'*B'*\text{Carry_In} + A*B*\text{Carry_In}$;

Equação do circuito do somador com saída e carry out

$$\text{Resto} = B \text{ Carry_In} + A \text{ Carry_In} + A B ;$$

Tabela verdade para o subtrator

| SUBTRATOR | | | | | |
|-----------|---|-----|---|------|---|
| A | B | CIN | S | COUT | |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Mapa de Karnaugh do subtrator com saída

| SUBTRATOR | | | | | |
|-----------|--|----|--|----|--|
| S | | A' | | A | |
| A' | | 0 | | 1 | |
| CIN' | | 1 | | 0 | |
| CIN | | 0 | | 1 | |
| B' | | B | | B' | |

Mapa de Karnaugh para o subtrator com a saída e o carry out

| SUBTRATOR | | | | | |
|-----------|--|----|--|----|--|
| COUT | | A' | | A | |
| A' | | 0 | | 1 | |
| CIN' | | 0 | | 1 | |
| CIN | | 0 | | 0 | |
| B' | | B | | B' | |

Equação do circuito do subtrator com saída

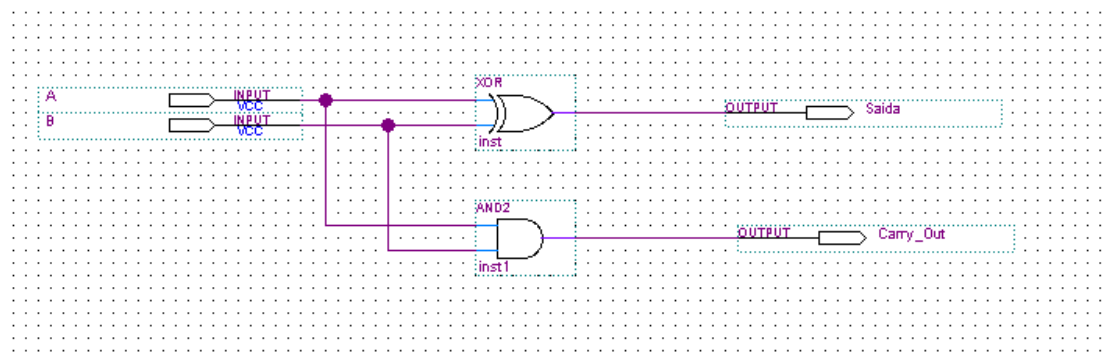
$$\text{Resultado} = A' \cdot B' \cdot \text{Carry_In} + A' \cdot B \cdot \text{Carry_In}' + A \cdot B' \cdot \text{Carry_In}' + A \cdot B \cdot \text{Carry_In};$$

Equação do circuito do subtrator com saída e carry out

$$\text{Resto} = A' \cdot B' \cdot \text{Carry_In} + A' \cdot B \cdot \text{Carry_In}' + A' \cdot B \cdot \text{Carry_In} + A \cdot B \cdot \text{Carry_In};$$

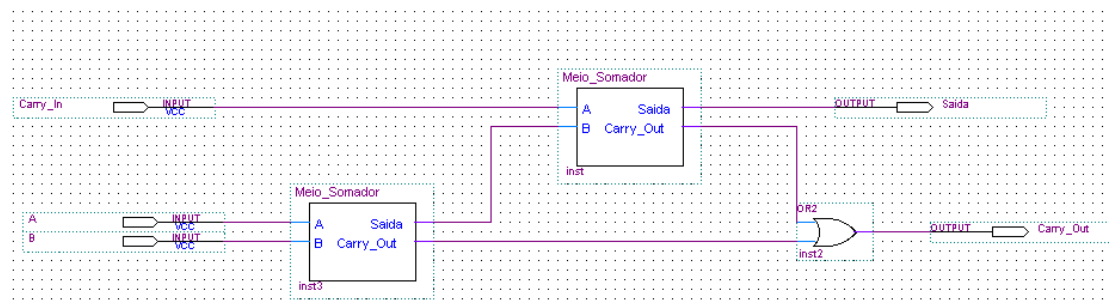
3. Circuitos Implementados

Meio Somador



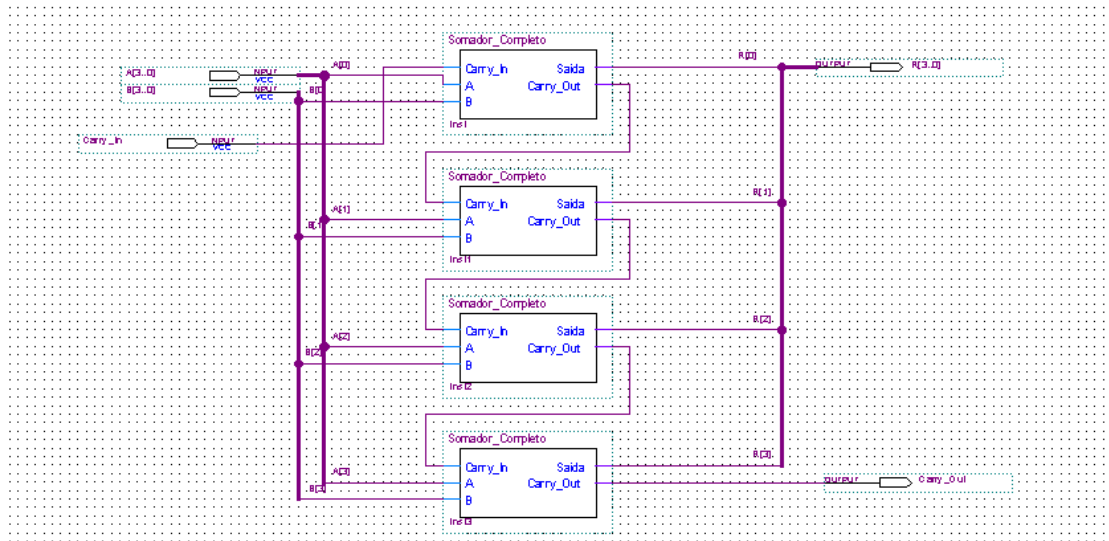
Como a tabela da porta XOR é similar a porta de soma, 0 e 0 resulta em 0, 0 e 1 resulta em 1, 1 e 0 resulta em 1, e 1 e 1 resulta em 0. No caso de 1 e 1, é necessário uma porta AND para assim então a saída ser equivalente a 0 e o Carry out equivalente a 1.

Somador Completo



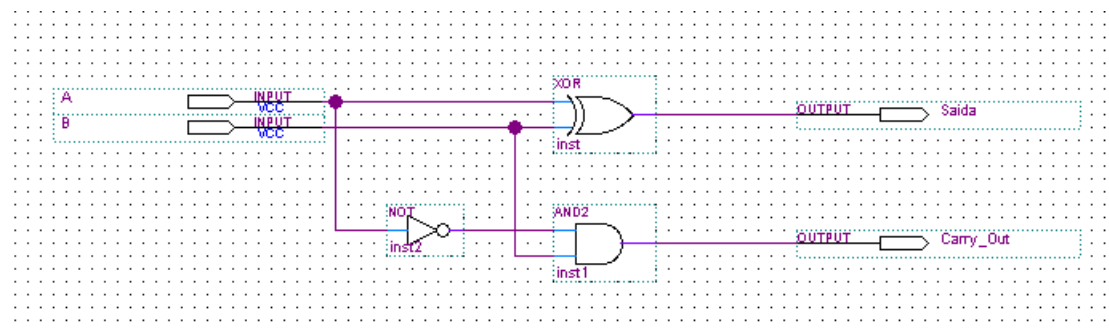
O circuito do somador completo seria uma soma de dois meio somadores, desse modo o resto da entrada ou carry in também será somado na operação.

Somador de 4 bits



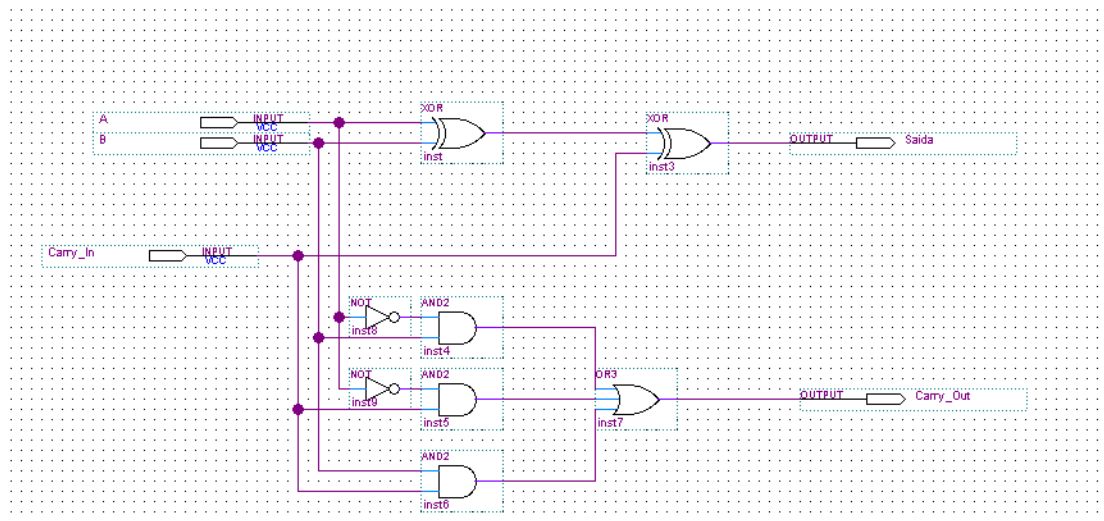
O Somador de 4 bits une 4 somadores completos usando barramento de entrada e saída.

Meio Subtrator



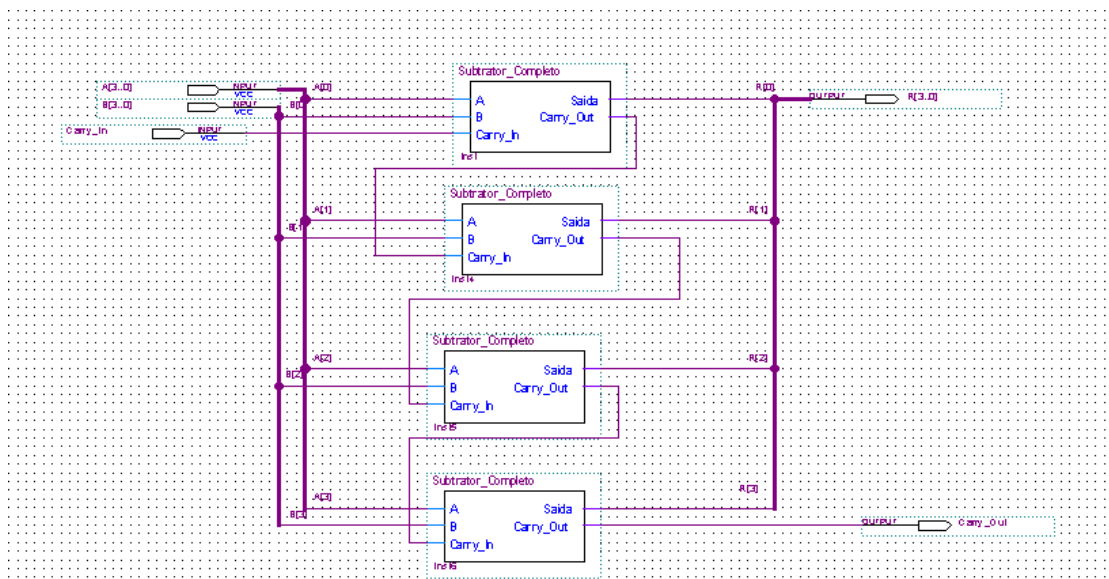
Com o uso da porta XOR, os valores para a subtração $0 - 0 = 0$, $1 - 0 = 1$, $1 - 1 = 0$ e $0 - 1 = 1$. Nesse caso se A for 1, a porta NOT resultará em 0, e se B for 1, o carry out será 1,

Subtrator Completo



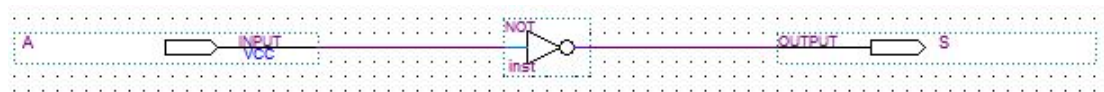
O subtrator completo representa um meio subtrator e mais um outro meio subtrator. Esse, subtrai-se os números e posteriormente subtrai o resultado com o carry in, então o carry out é calculado com as possíveis combinações de A, B e o carry in, caso do 0 - 1 que retorna 1.

Subtrator de 4 bits



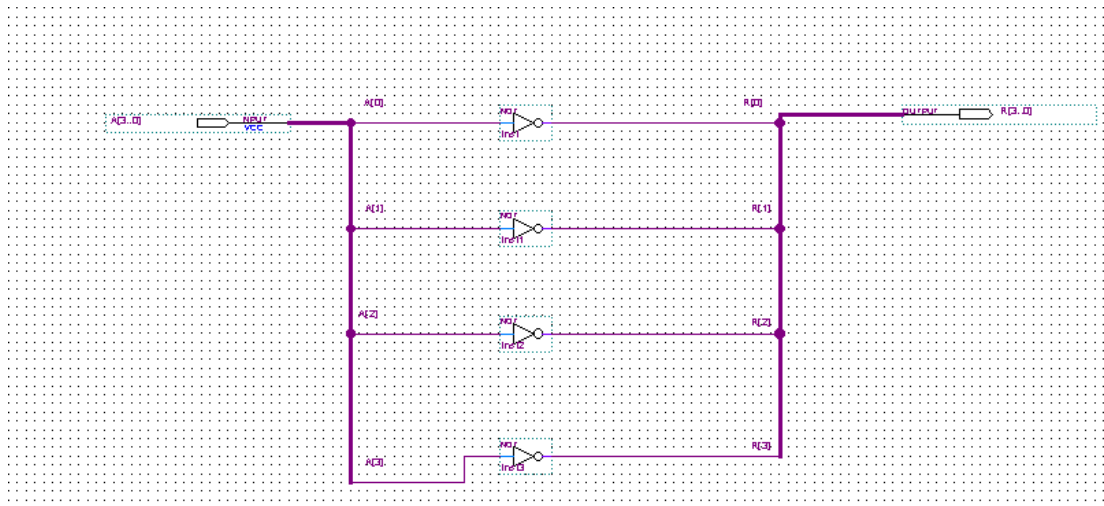
O Subtrator de 4 bits une 4 subtratores completos usando barramento de entrada e saída.

Complemento de 1



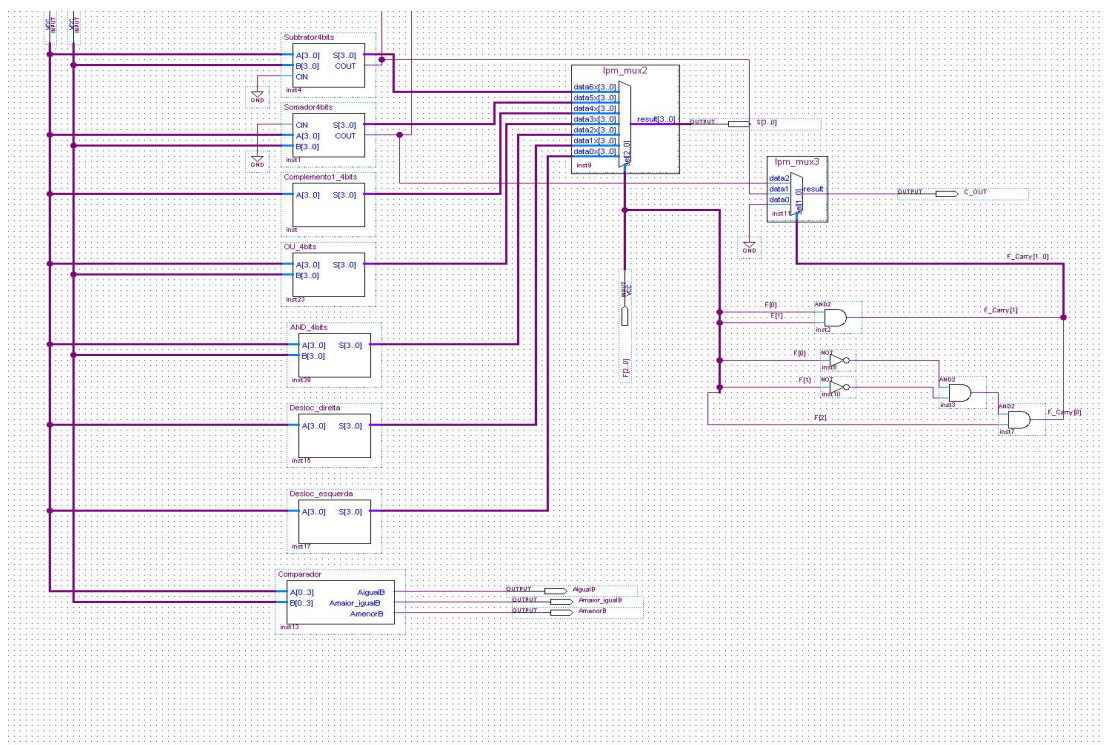
Com o uso da porta NOT, se A for 0, a saída será 1, caso contrário, a saída será 0.

Complemento de 1 de 4 bits

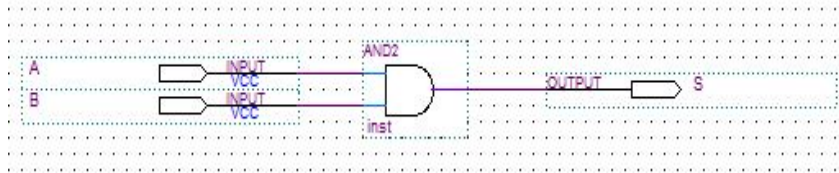


Para o complemento de 1 com 4 bits, utiliza-se a porta NOT quatro vezes.

ULA de 4 bits com somador, subtrator, complemento de 1, OR, AND, deslocamento a direita, deslocamento a esquerda e comparador.

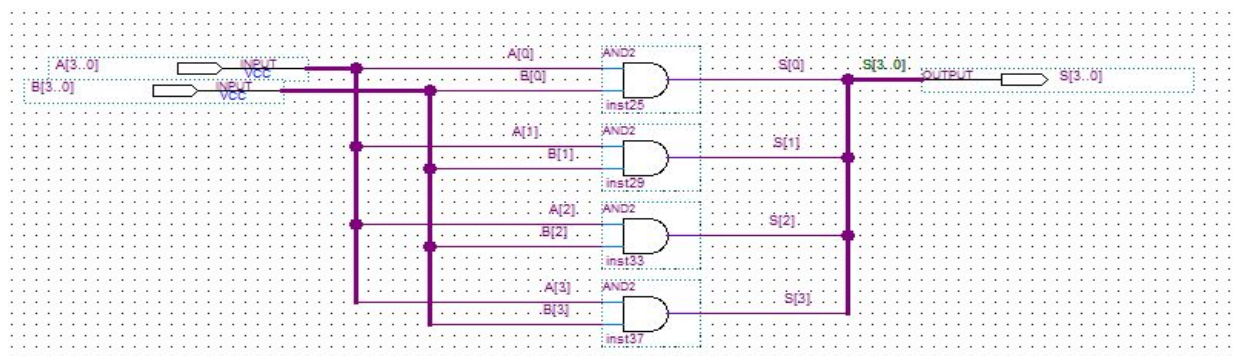


Operação AND



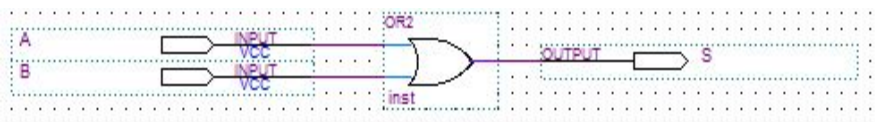
A operação da porta AND resulta em, 1 se A e B forem 1, e 0 nos demais casos que seguem sua tabela verdade.

Circuito da operação AND com 4 bits

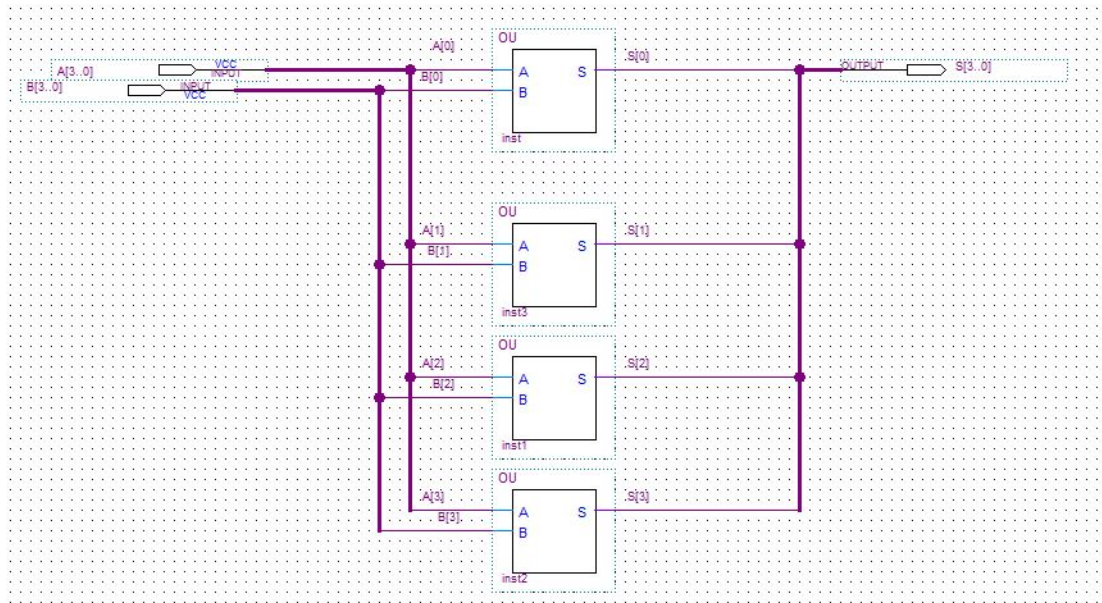


O circuito da operação AND com 4 bits segue o mesmo padrão de um circuito da operação AND para 1 bit, porém esse tem barramento para 4 bits.

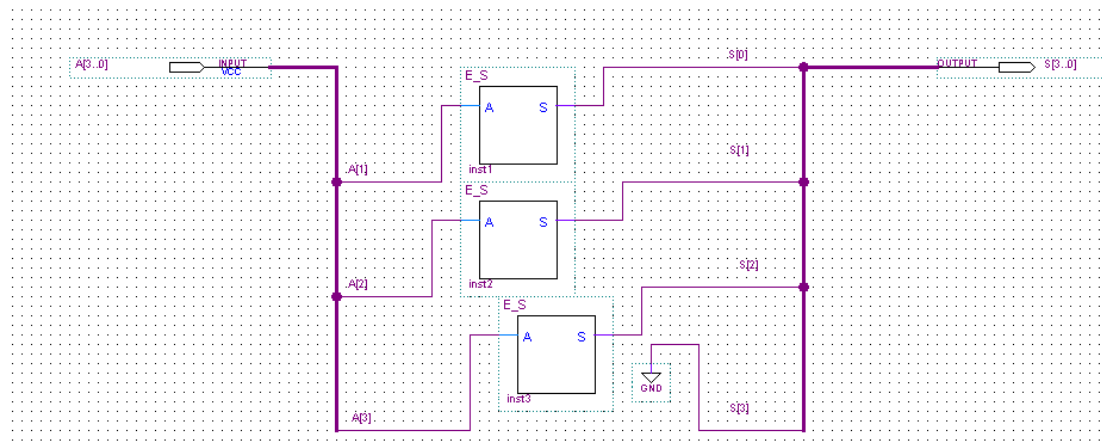
Operação OR



O circuito da operação OR segue o padrão de sua respectiva tabela verdade, 1 caso alguma entrada seja verdadeira, e 0 caso as duas entradas forem falsa.

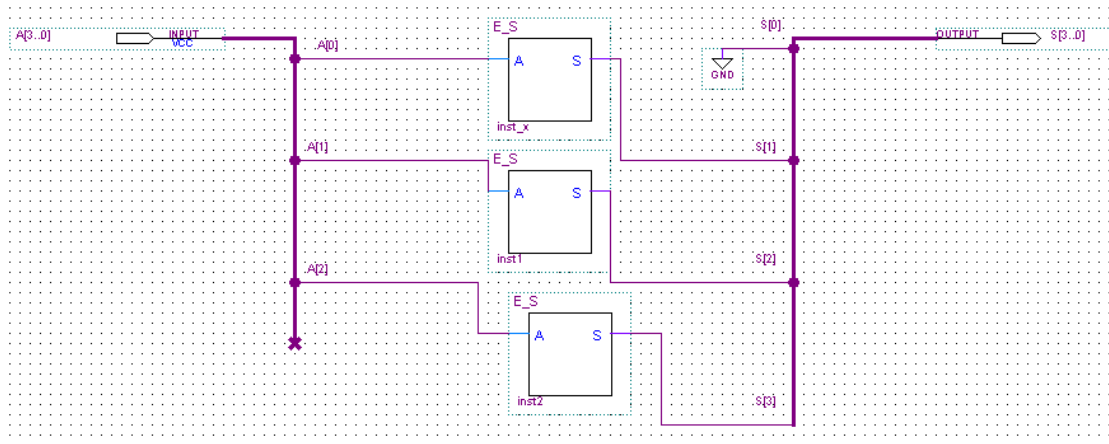


Deslocamento de bits a direita



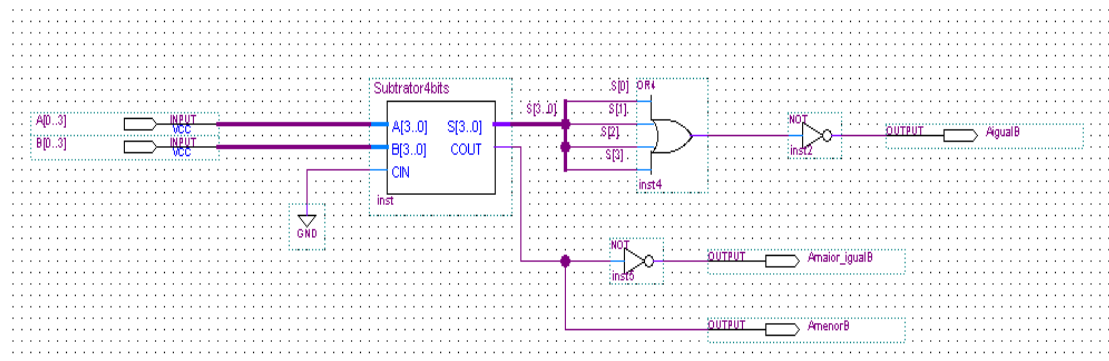
Desloca todos os bits para a direita preenchendo os novos com 0

Deslocamento de bits a esquerda



Desloca todos os bits para a esquerda preenchendo os novos com 0

Comparador de bits



O circuito acima representa um comparador de 4 bits

Através da subtração, podemos obter a comparação de 2 números nos seguintes casos:

$A = B$

$A \geq B$

$A < B$

No caso de A for igual a B, é feita uma subtração para análise da equivalência dos termos.

Se A for maior ou igual a B, a subtração retorna um carry out igual a 0 e é feita uma inversão do mesmo a fim de setar a saída.

Para A menor do que B, a subtração retorna um carry out igual a 1.

4. Simulação das condições de teste

Chave de Seleção: F[0], F[1], e F[2]:

000: deslocamento de bits para a esquerda

001: deslocamento de bits para a direita

010: AND

011: OR

100: complemento de 1

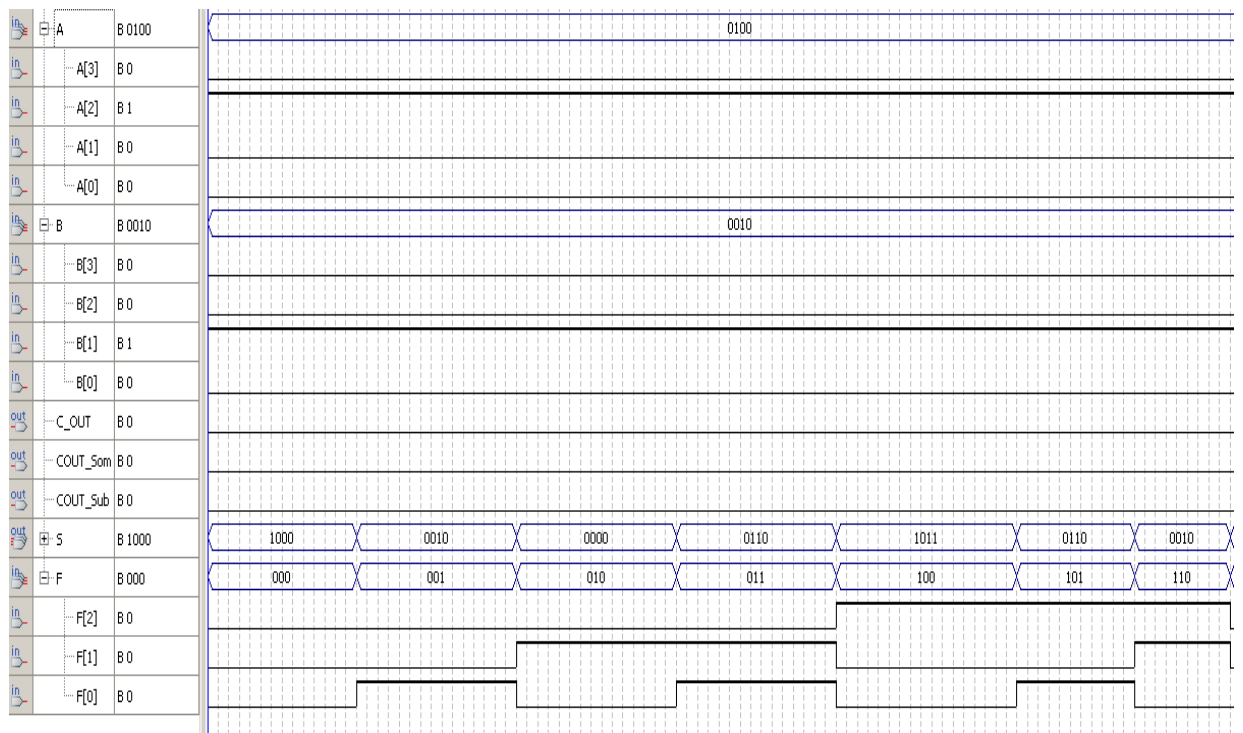
101: somador

110: subtrator

Comparador sem chave de seleção

Valores:

| | | | | |
|-----------|-----------|-----------|----------|--------|
| A[0] = 0, | A[1] = 0, | A[2] = 1, | A[3] = 0 | : 0100 |
| B[0] = 0, | B[1] = 1, | B[2] = 0, | B[3] = 0 | : 0010 |



Saídas:

AND: 0010 e 0100 = 0000

OR: 0010 ou 0100 = 0110

Complemento de 1: 0010 = 1101

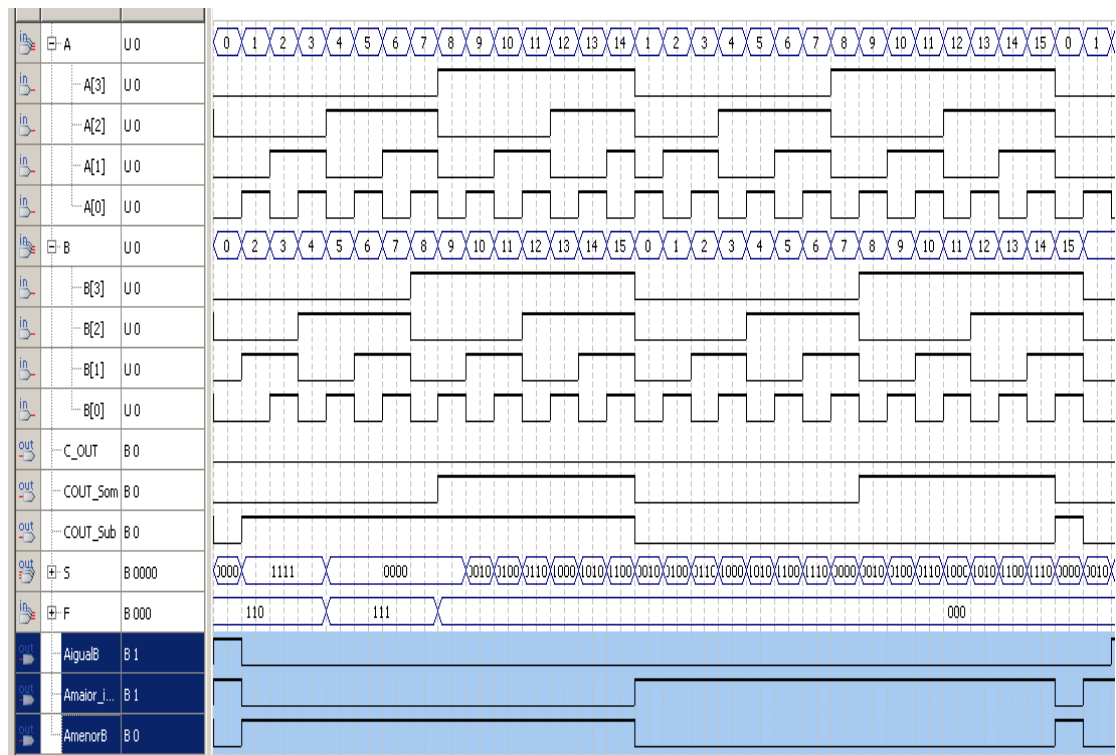
Soma: 0010 + 0100 = 0110

Subtração: 0100 - 0010 = 0010

Deslocamento de bit para a direita: 0100 = 0010

Deslocamento de bit para a esquerda: 0100 = 1000

Simulações do comparador



5. Conclusões

Através da implementação do somador, subtrator, complemento de 1, operações OR, AND, deslocamento de bits para direita e esquerda e comparador de 4 bits, foi possível iniciar o desenvolvimento de uma ULA de 4 bits com as respectivas operações.