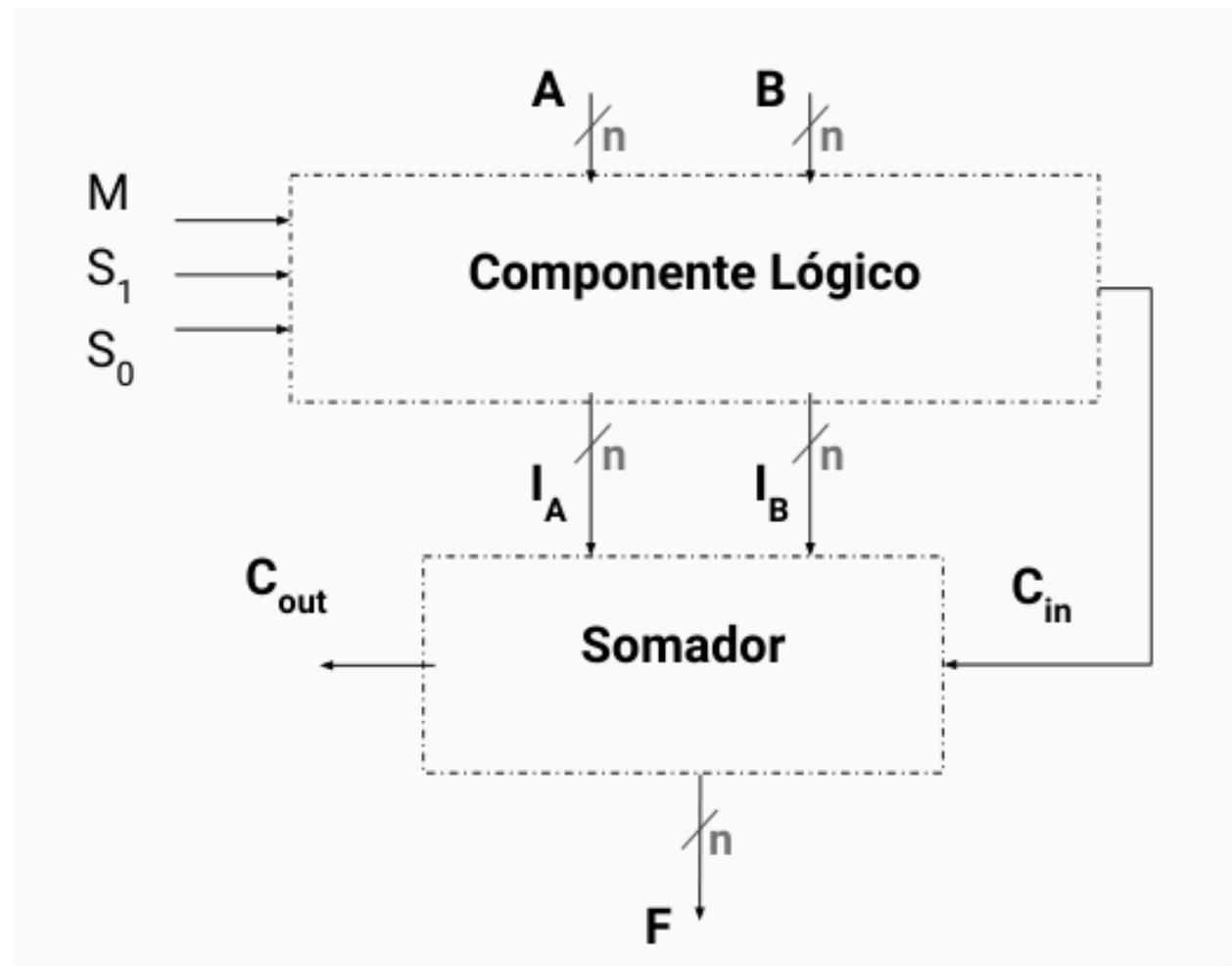


## ULA de 16 bits

O diagrama de blocos da implementação da nossa ULA é idêntico ao apresentado no slide de referência da disciplina, este é:



A única diferença notável é a nomeação de algumas variáveis que serão esclarecidas em seguida nesse documento.

### Tabela verdade do componente lógico

M	S1	S0	Operação	Ia	Ib	Cin
0	0	0	$A + B$	$A$	$B$	0
0	0	1	$A - B$	$A$	$\overline{B}$	1
0	1	0	$A \ll 1$	$A(14 \text{ downto } 0) \& '0'$	(others => '0')	0
0	1	1	$A \gg 1$	$'0' \& A(14 \text{ downto } 0)$	(others => '0')	0
1	0	0	$A \wedge B$	$A \wedge B$	(others => '0')	0
1	0	1	$A \vee B$	$A \vee B$	(others => '0')	0
1	1	0	$A \oplus B$	$A \oplus B$	(others => '0')	0
1	1	1	$A \text{ XNOR } B$	$A \text{ XNOR } B$	(others => '0')	0

## Expressões de entrada do somador

O nosso somador de 16 bit é implementado a partir de vários somadores de 1 bit, tal somador tem as seguintes entradas e saídas:

- Entradas:
  - **A**: bit do operando  $A$ .
  - **B**: bit do operando  $B$ .
  - **Cin** carry-in (bit de transporte da última posição).
- Saídas:
  - **Result**: soma dos bits  $A, B$  e **Cin**.
  - **Cout**: carry-out (bit de transporte para a próxima posição).

**Expressões lógicas** Além disso, as expressões das saídas do somador de 1 bit são:

- **Soma**

$$F = A \oplus B \oplus \text{Cin}$$

- **Carry-out:**

$$\text{Cout} = (A \cdot B) + (\text{Cin} \cdot (A \oplus B))$$

**Mapa de karnaugh para as saídas**

- **Para  $F$  (Result):**

O mapa de karnaugh de  $F$  para as combinações das entradas  $A, B$  e **Cin** é:

$A \setminus B, \text{Cin}$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

- **Para Cout:**

E por fim, o mapa de karnaugh para **Cout** com as combinações das entradas  $A, B$ , e **Cin** é:

$A \setminus B, \text{Cin}$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

## Somador 16 bits

Uma vez definido o funcionamento do somador de 1 bit, implementamos o somador de 16 utilizando a propagação de carry, isto é:

- Cada carry (**Cout**) é passado como **Cin** para o próximo somador de 1 bit.
- O resultador (**Result**) é calculado bit a bit usando os somadores de 1 bit.

## Plano de Simulação

Para averiguar o funcionamento da ULA, realizaremos os seguintes testes no Quartus:

## Cenários de Teste para a ULA de 16 bits

Caso de Teste	Entrada	Saída Esperada	Descrição
Teste 1	M=0, S1=0, S0=0, A=0001, B=0001	0000 0000 0000 0010	Soma dos valores A e B ( $1 + 1 = 2$ ).
Teste 2	M=0, S1=0, S0=1, A=0005, B=0003	0000 0000 0000 0010 e Cout=1	Subtração de A por B ( $5 - 3 = 2$ ).
Teste 3	M=0, S1=1, S0=0, A=0001	0000 0000 0000 0010	Deslocamento de A um bit à esquerda.
Teste 4	M=0, S1=1, S0=1, A=0002	0000 0000 0000 0001	Deslocamento de A um bit à direita.
Teste 5	M=1, S1=0, S0=0, A=0101, B=0011	0000 0000 0000 0001	Operação lógica AND entre A e B.
Teste 6	M=1, S1=0, S0=1, A=0101, B=0011	0000 0000 0000 0111	Operação lógica OR entre A e B.
Teste 7	M=1, S1=1, S0=0, A=0101, B=0011	0000 0000 0000 0110	Operação lógica XOR entre A e B.
Teste 8	M=1, S1=1, S0=1, A=0101, B=0011	1111 1111 1111 1001	Operação lógica XNOR entre A e B.
Teste 9	M=0, S1=0, S0=0, A=1111, B=0001	0000 0000 0000 0000 e Cout=1	Soma com <b>overflow</b> .
Teste 10	M=0, S1=0, S0=1, A=0000, B=0001	0000 0000 0000 0000	Subtração negativa resultando em 0.

Devido ao número de testes, seus resultados foram colocados na pasta **assets/**. De antemão, assertamos que o resultado foi bem sucedido!