



Laboratório 4: Síntese e Análise de Unidades Lógicas Aritméticas (ULAs).

Bernardo Hoffmann da Silva
Marcos Vinicius Pereira Veloso

4. Tarefas

4.1. Simulação de ULA de 1 bit

Feita a montagem especificada para a ULA de 1 bit dada, pode-se obter o diagrama de temporização conforme simulação, no qual o resultado é dado pela Figura 1.

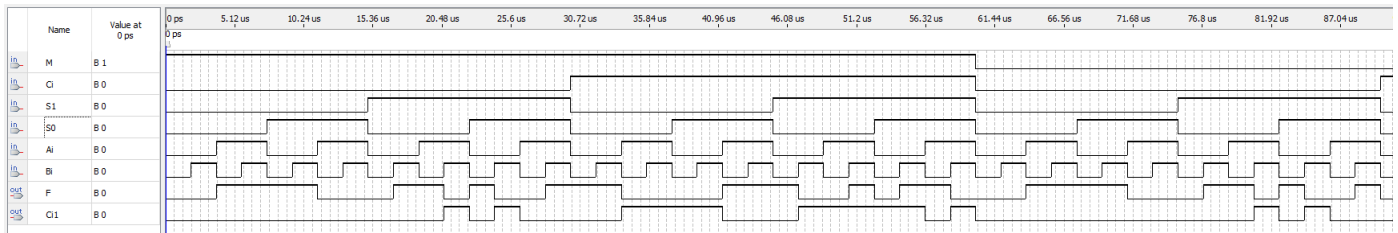


Figura 1: Diagrama de temporização obtido para a ULA especificada no item 4.1.

Por inspeção, pode-se denotar que, para cada valor de entrada M , S_1 , S_0 e C_i , o resultado de simulação é condizente com a Tabela de operações especificada.

4.2. Projeto e Simulação de uma ULA de 4 bits utilizando o circuito integrado 74181

Note-se, para a implementação das operações requeridas para a ULA, basta que sejam utilizadas as funções pertinentes e necessárias para efetuar as operações desejadas.

Nesse sentido, feita a inserção das entradas de 4 bits A e B na ULA, deve-se especificar a operação a ser feita a partir da variável de seleção S. Para $S = 1$, a operação especificada é A mais B, i.e., a que corresponde às variáveis de seleção $S_3S_2S_1S_0\bar{C}_n = 10011$, enquanto que para $S = 0$ tem-se a seleção $S_3S_2S_1S_0\bar{C}_n = 01100$, para que ocorra A menos B. Dado que o valor de M é o mesmo para ambas as operações (valor zero), basta associarmos a tal entrada o parâmetro *ground*. Com as duas expressões de seleção dadas, visto que o valor de cada variável alterna quando S se alterna, uma possível implementação da lógica é que seja associado a cada entrada

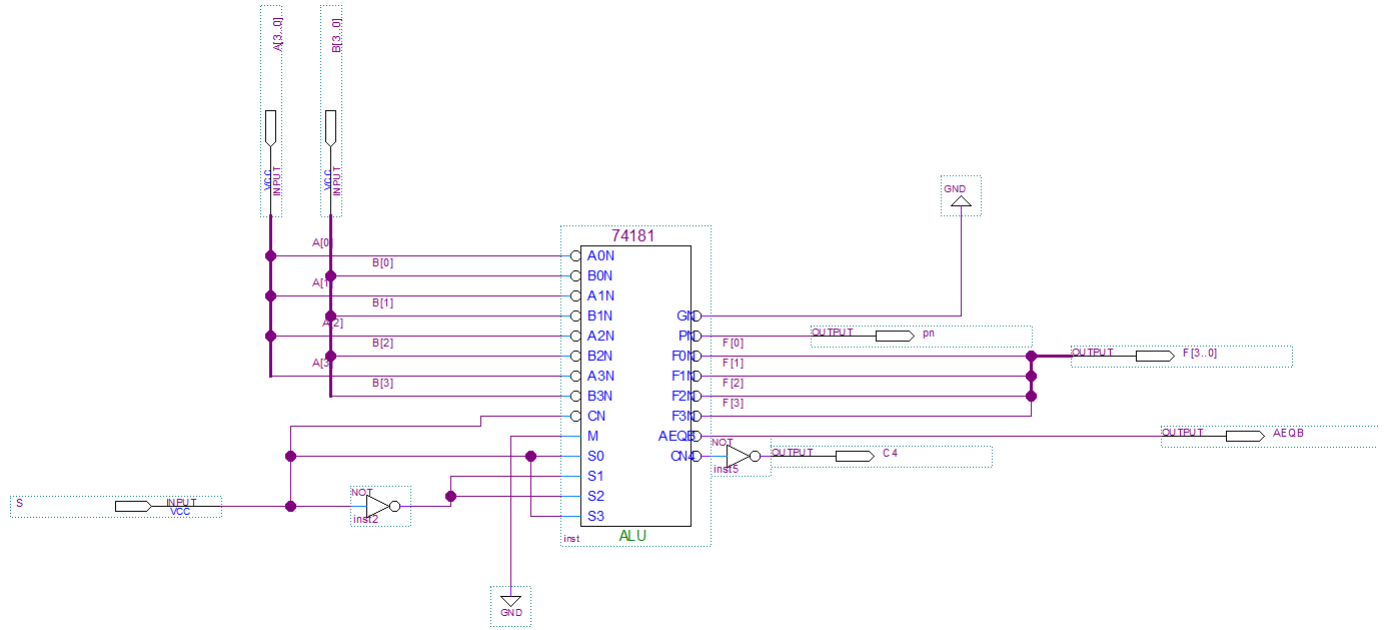


Figura 2: Montagem do circuito lógico desejado usando a ULA 74181, conforme projeto do item 4.2.

S_i e \bar{C}_n o valor lógico S ou \bar{S} . Com isso em mente, pode-se montar o circuito conforme segue na Figura 2.

Feito o circuito, pode-se simular os resultados de saída conforme os valores de S , para a obtenção dos diagramas de temporização das operações soma e diferença, conforme seguem nas Figuras 3 e 4.

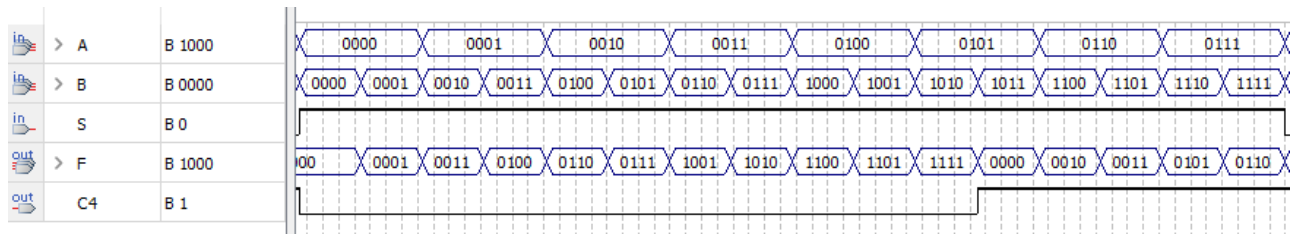


Figura 3: Diagrama de temporização do operador soma da ULA especificada no item 4.2. Nota-se que as somas são realizadas como o esperado.

Pode-se denotar, por inspeção de cada coluna, que as saídas condizem com o as saídas de resultado e de carry/borrow dos operadores soma e subtração. Para fins de consulta, pode-se comparar os resultados obtidos em tais diagramas com as respectivas colunas que fazem a operação soma de A, B e C das Tabelas 1 e 3, relativas ao tópico que segue.

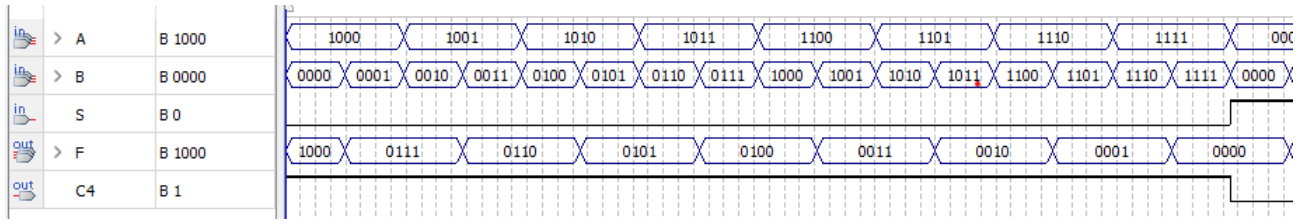


Figura 4: Diagrama de temporização do operador subtração da ULA especificada no item 4.2. Nota-se que as subtrações são feitas como o eseprado.

4.3: Projeto e Simulação de uma célula básica ULA de 1 bit

Com as funções da ULA especificada, pode-se elaborar a Tabela verdade para cada uma das operações, tanto para a saída de resultado quanto para a saída de carry. Tais resultados estão sintetizados nas Tabelas 1, 2 e 3.

S_1S_0			00	01	10	11
A	B	C	A mais C	A menos C	A mais B mais C	A menos B menos C
0	0	0	0	0	0	0
0	0	1	1	1	1	1
0	1	0	0	0	1	1
0	1	1	1	1	0	0
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	1	0	1	1	0	0
1	1	1	0	0	1	1

Tabela 1: Esquema de saída F da Tabela Verdade para a ULA especificada, para as operações com $M = 0$.

S_1S_0		00	01	10	11
A	B	\bar{A}	$\bar{A}\bar{B}$	$A + B$	$A \oplus B$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	0	0	0

Tabela 2: Esquema de saída F da Tabela Verdade para a ULA especificada, para as operações com $M = 1$.

S_1S_0			00	01	10	11
A	B	C	A mais C	A menos C	A mais B mais C	A menos B menos C
0	0	0	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	0	0	1
0	1	1	0	1	1	1
1	0	0	0	0	0	0
1	0	1	1	0	1	0
1	1	0	0	0	1	0
1	1	1	1	0	1	1

Tabela 3: Esquema de saída C da Tabela Verdade para a ULA especificada, para as operações com $M = 0$.

Pelo mapa de Karnaugh reduzido, pode-se encontrar uma expressão de saída F a partir dos termos de seleção M , S_1 e S_0 . Como há três termos de seleção, há $2^3 = 8$ modos de seleção mutuamente exclusivos, que podem por sua vez ser implementada com um multiplexador 8x1. A expressão de F , conforme mencionada, é dada na Relação (1).

$$\begin{aligned}
F = & S_1S_0M(A \oplus B) + S_1S_0\bar{M}(A - B - C) + S_1\bar{S}_0\overline{MA + B} \\
& + \bar{S}_1S_0M(\overline{AB}) + \bar{S}_1\bar{S}_0M(\bar{A}) + \bar{S}_1S_0\bar{M}(A - C) + S_1\bar{S}_0\bar{M}(A \text{ mais } B \text{ mais } C) + \bar{S}_1\bar{S}_0\bar{M}(A \text{ mais } C)
\end{aligned} \tag{1}$$

Como as operações soma e subtração possuem mesma tabela verdade de saída (i. e., saída de resultado da forma $A \oplus B$), podemos implemenar tais operações com portas XOR, fazendo as devidas substituições dos operadores $+$ e $-$ destacados por \oplus .

Posto isso, para a implementação da saída C da ULA especificada, tem-se que, para cada par S_1S_0 especificado, a saída carry é tal que é gerada por cada uma das quatro operações de soma/diferença definidos na ULA especificada quando $M = 0$. Dado que:

- Para uma operação soma entre A e B com carry inicial C , o carry de saída é dado por $C = A(B + C) + BC$;
- Para uma diferença entre A e B com borrow inicial C , o borrow de saída é dado por $C = \bar{A}(B + C) + BC$

Nesse sentido, pode-se usar as duas relações acima para definir o carry de saída conforme uma das 4 operações possíveis para $M = 0$. Note-se que, como para $M = 1$ não são feitas operações de soma ou subtração, o resultado de carry de saída não é um dado pertinente (don't care). Pode-se então, tal como especificado para a saída F conforme relação (1), usar um multiplexador de três variáveis de seleção (S_1 , S_0 e M) para selecionar cada um dos 8 casos possíveis de saída. Para o Carry, pode-se utilizar outro multiplexador 8x1, de variáveis de seleção S_1 , S_0 e M , no qual é convencionado o valor zero para as 4 entradas com $M = 1$ (já que tal valor não é relevante) e o uso das funções booleanas especificadas pelas duas relações dadas no parágrafo anterior para o caso em que $M = 0$. A figura 5 apresenta o circuito simulado para as especificações dadas.

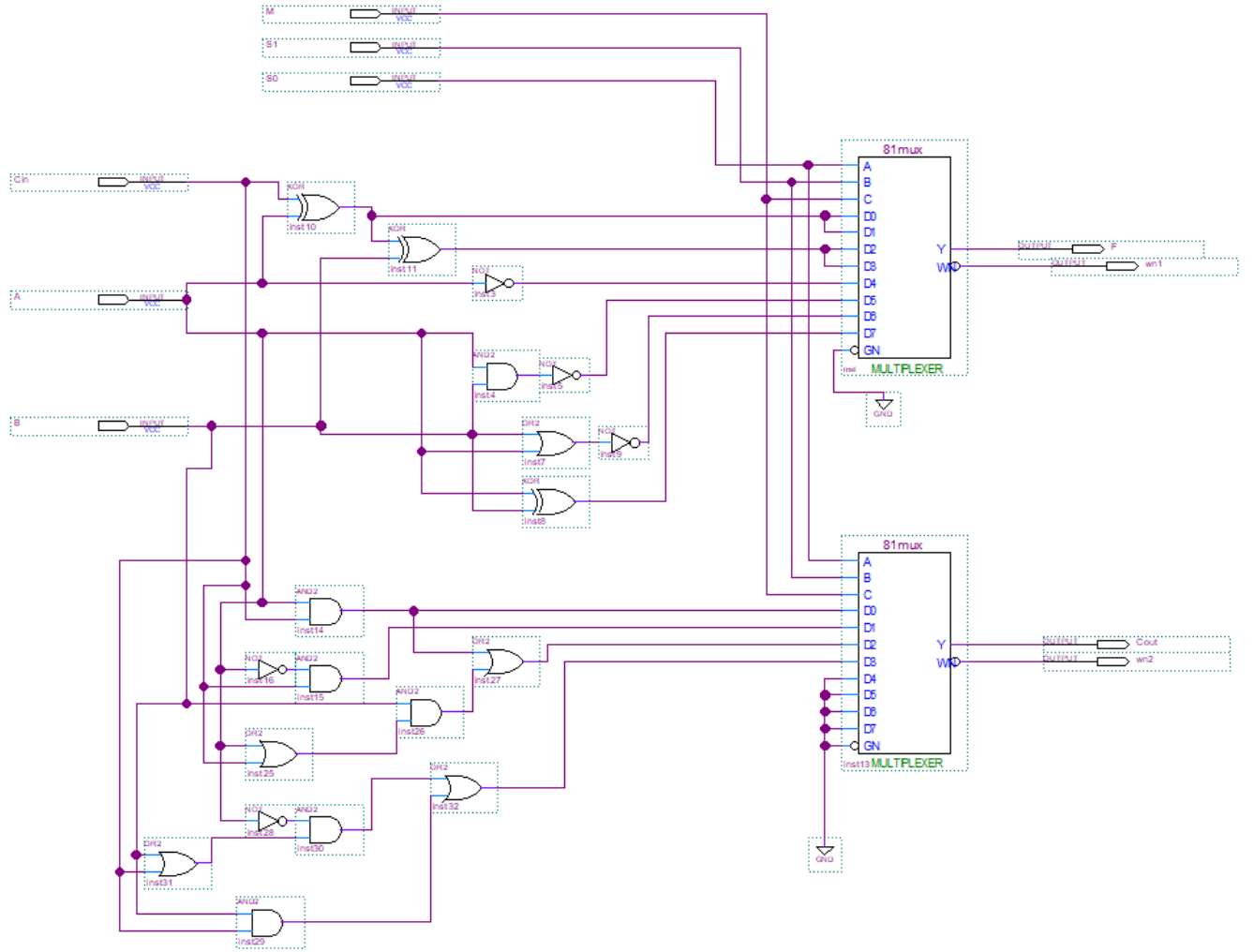


Figura 5: Circuito montado utilizando dois mux 8x1 para as saídas F_i e C_i especificadas em 4.3

Em seguida, foram feitos testes para 12 operações - as funções booleanas, as somas e subtrações (essas sem e com Carry-in, respectivamente). As 12 operações foram testadas para os valores $A_i = 1$ e $B_i = 0$, cujos resultados são mostrados na figura 6, $A_i = 0$ e $B_i = 1$, cujos resultados são mostrados na figura 7 e $A_i = 1$ e $B_i = 1$, cujos resultados são mostrados na figura 8.

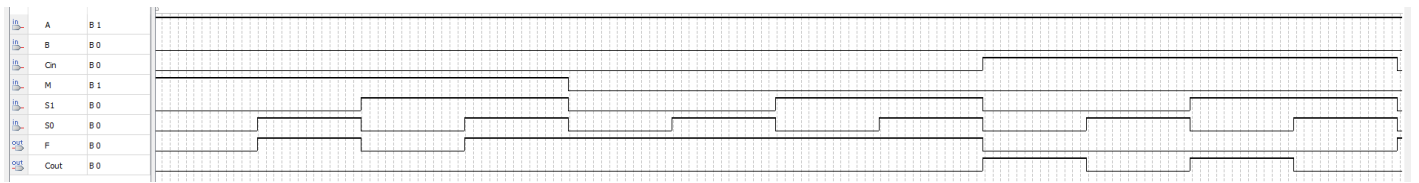


Figura 6: Resultados das operações da ULA para $A_i = 1$ e $B_i = 0$

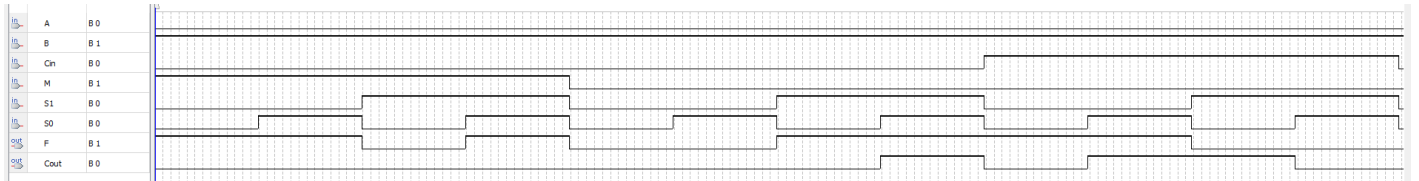


Figura 7: Resultados das operações da ULA para $A_i = 0$ e $B_i = 1$

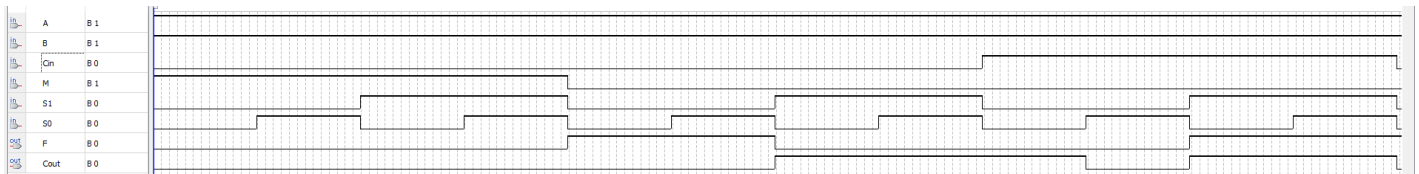


Figura 8: Resultados das operações da ULA para $A_i = 1$ e $B_i = 1$

Nota-se que os resultados encontrados pela simulação seguem o esperado das Tabelas Verdade construídas nas tabelas 1, 2 e 3. Com isso, conclui-se o projeto e a simulação da ULA de 1 bit com as operações enunciadas em 4.3.