

Projeto de um ULA de 4-bits (8 Operações)

Prof. Fernando Passold
Eng. Elétrica
Universidade de Passo Fundo
fpassold@upf.br

22 de novembro de 2016

Resumo

Este material demonstra uma forma de realizar o projeto de uma ULA de 4-bits capaz de realizar 8 operações diferentes. É ressaltado a forma de realizar o projeto mais do que o diagrama elétrico final.

1 ULA Desejada

Se pede o projeto completo de uma ULA para palavras de 4-bits capaz de realizar as operações mostradas na tabela 1. Notar que as operações de subtração devem obedecer o padrão Complemento-2. A ULA além das palavras de entrada A e B e da saída F que são de 4-bits, ainda deve disponibilizar 1 bit de entrada para o *carry-in*: C_0 e o bit de saída relacionado com o *carry-out*: C_4 .

2 Solução

2.1 Primeira Parte

Analisando-se a tabela de operações da ULA se percebe que metade da mesma realiza operações aritméticas (reparar que $S_2 = 0$) e a outra metade realiza apenas operações lógicas ($S_2 = 1$). Então o estágio de saída da ULA pode ser finalizada por um simples MUX de 2 canais de entrada conforme mostra o esboço mostrado na figura 1.

Nota-se ainda que para as operações aritméticas, um único CI Somador de 4-bits será o “coração”(parte central) deste bloco. Já para a parte lógica, como 4 operações lógicas diferentes são executadas, não há como aproveitar um único componente o que implica em usar um MUX de 4 canais de entradas para selecionar que operação lógica resultará do bloco que tratará exclusivamente das operações lógicas desta ULA. Notar ainda que os bits menos significativos de seleção da ULA: S_1S_0 são os que selecionam a operação lógica à ser executada (enquanto $S_2 = 1$; mas as operações lógicas podem ser executadas internamente independente do estado lógico de S_2).

Desta forma, podemos esboçar um primeiro diagrama de blocos relacionado com a forma como a ULA provavelmente executará de maneira separada as operações aritméticas e lógicas – ver figura 2.

Seleção da Operação $S_2S_1S_0$	Operação	Observações
0 0 0	$F = A + B$	Aritmético, requer $C_0 = 0$
0 0 1	$F = A - B$	Aritmético, requer $C_0 = 1$
0 1 0	$F = B - A$	Aritmético, requer $C_0 = 1$
0 1 1	$F = 2A$	Aritmético, requer $C_0 = 0$
1 0 0	$F = \overline{A}$	Lógico, $C_0 = X$
1 0 1	$F = \overline{B}$	Lógico, $C_0 = X$
1 1 0	$F = A \cdot B$	Lógico, $C_0 = X$
1 1 1	$F = A \oplus B$	Lógico, $C_0 = X$

Tabela 1: Operações previstas para a ULA.

$S_2 S_1 S_0$	Operação	Obs.
0'0'0	$F = A + B$	Aritmético, Requer $C_0 = 0$
0'0'1	$F = A - B$	Requer $C_0 = 1$
0'1'0	$F = B - A$	Requer $C_0 = 1$
0'1'1	$F = 2A$	Requer $C_0 = 0$
1'0'0	$F = \bar{A}$	$C_0 = X$
1'0'1	$\bar{F} = \bar{B}$	$C_0 = X$
1'1'0	$F = A \cdot B$	Lógico, $C_0 = X$
1'1'1	$F = A \oplus B$	$C_0 = X$

Figura 1: Operações aritméticas e lógicas executadas pela ULA.

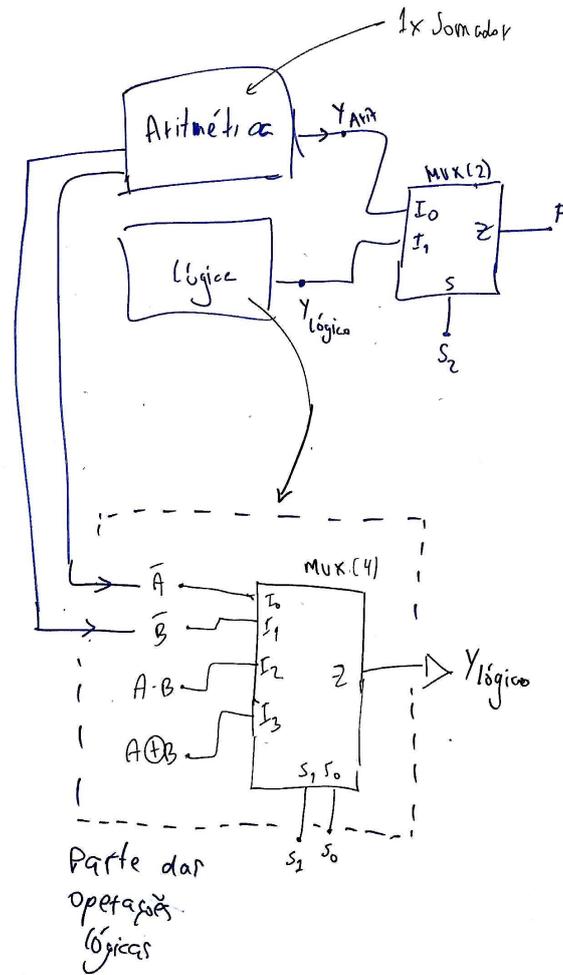


Figura 2: Blocos básicos da ULA (um primeiro diagrama em blocos).

ULA de 4-bits: $F = A \pm A$

Rej	$S_2 S_1 S_0$	Operação	Obs.
0	0'0 0	$F = A + B$	Aritmético, Requer $C_0 = 0$
1	0'0 1	$F = A - B$	Requer $C_0 = 1$
2	0'1 0	$F = B - A$	Requer $C_0 = 1$
3	0'1 1	$F = \overline{2A}$	Requer $C_0 = 0$
4	1'0 0	$F = \overline{A}$	$C_0 = X$
5	1'0 1	$F = B$	$C_0 = X$
6	1'1 0	$F = A \cdot B$	Lógico, $C_0 = X$
7	1'1 1	$F = A \oplus B$	$C_0 = X$

Arrows from the table point to X_B (from S_2) and X_A (from S_1).

Figura 3: Análise das operações de complementação de bits necessárias nesta ULA.

2.2 Segunda Parte

Deduzida esta primeira parte, podemos nos “focar” em resolver cada parte em separado da ULA. Nos concentrando em separado na parte aritmética e depois na parte lógica.

2.2.1 Parte Aritmética

Concentrando-nos apenas nas operações aritméticas da ULA percebemos algumas detalhes – ver figura 3.

Analisando a figura 3 notamos que:

- Os bits da palavra A devem ser invertidos (ou complementados) quando estamos realizando as operações: $F = B - A$ ($S_2 S_1 S_0 = 010_{(2)} = 2_{(10)}$) e a operação: $F = \overline{A}$ ($S_2 S_1 S_0 = 100_{(2)} = 4_{(10)}$). Podemos identificar esta situação como a variável X_A que identifica (em ativo alto) quanto necessitamos \overline{A} :

$$\begin{aligned}
 X_A &= \sum_m \{2, 4\} \\
 &= \underbrace{S_2 \cdot S_1 \cdot S_0}_{m_2} + \underbrace{S_2 \cdot \overline{S_1} \cdot \overline{S_0}}_{m_4} \\
 &= \overline{S_0} \cdot (S_2 \oplus S_1) \quad \therefore \text{ Solução SEM usar DEC} \\
 \text{ou} \\
 X_A &= \overline{O_2} + \overline{O_4} \quad [\text{Saída de um DEC (Ativo ALTO)}] \\
 &= \overline{O_2} + \overline{O_4} \\
 &= \overline{O_2 \cdot O_4} \quad (\text{Porta NAND}(2))
 \end{aligned}$$

- Os bits da palavra B devem ser invertidos (ou complementados) quando estamos realizando as operações: $F = A - B$ ($S_2 S_1 S_0 = 001_{(2)} = 1_{(10)}$) e a operação: $F = \overline{B}$ ($S_2 S_1 S_0 = 101_{(2)} = 5_{(10)}$). Podemos identificar esta

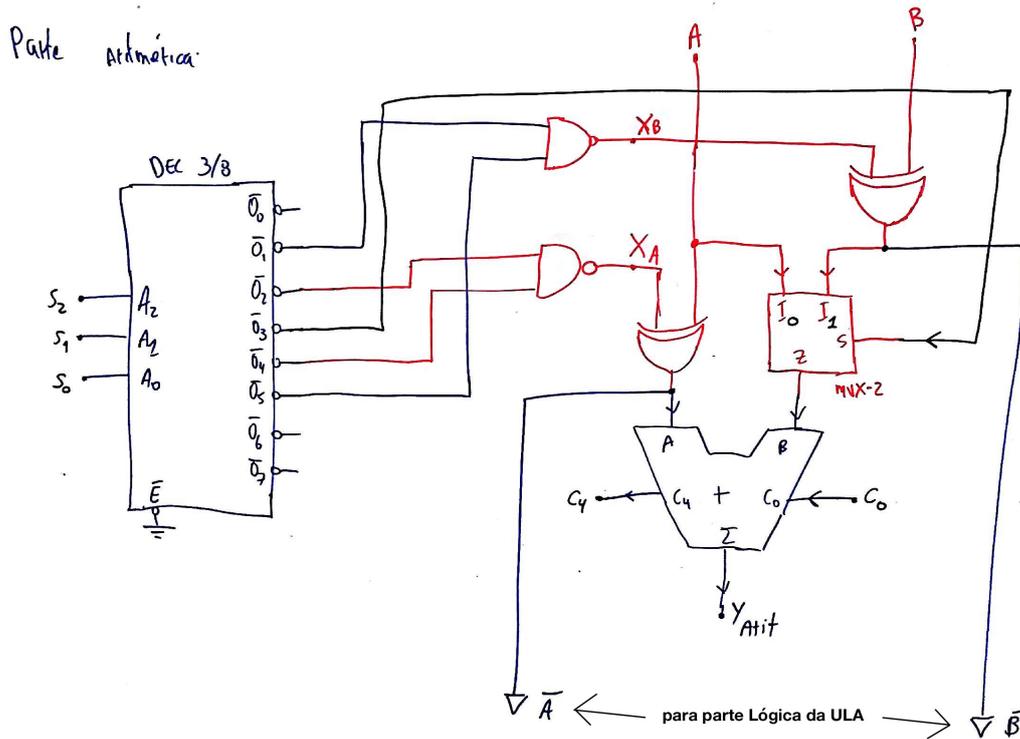


Figura 4: Parte aritmética da ULA.

situação como a variável X_B que identifica (em ativo alto) quanto necessitamos \bar{B} :

$$\begin{aligned}
 X_B &= \sum_m \{1, 5\} \\
 &= \underbrace{\bar{S}_2 \cdot \bar{S}_1 \cdot S_0}_{m_1} + \underbrace{S_2 \cdot \bar{S}_1 \cdot S_0}_{m_5} \\
 &= \bar{S}_1 \cdot S_0 \cdot (\underbrace{\bar{S}_2 + S_2}_{=1}) \\
 &= \bar{S}_1 \cdot S + 0 \quad \therefore \text{ Solução SEM usar DEC} \\
 \text{ou} \\
 X_B &= O_1 + O_5 \quad [\text{Saída de um DEC (Ativo ALTO)}] \\
 &= \overline{O_1 + O_5} \\
 &= \overline{O_1 \cdot O_5} \quad (\text{Porta NAND}(2))
 \end{aligned}$$

- E notamos ainda que há um caso em que na entrada B do Somador Binário não deve entrar a palavra externa B e sim a palavra externa A . Este “detalhe” pode ser solucionado através de MUX de 2 canais que controla que palavra externa (A ou B) passam para a entrada B do Somador. Mas neste caso, resta estabelecer como será realizado o controle sobre a entrada de Seleção deste MUX. Analisando-se a tabela percebe-se que quando estamos na linha de $S_2 S_1 S_0 = 011_{(2)} = 3_{(10)}$, se faz necessário passar a palavra externa A para a entrada B do MUX. Ficaria mais prático associar o próprio nível lógico (BAIXO) da saída \bar{O}_3 do DEC (previsto para os itens anteriores) com a entrada $Select$ do MUX; neste caso, quando a entrada $Select$ deste MUX for igual à '0' queremos que o mesmo passe para sua saída a entrada A , motivo pelo qual, a palavra externa A vai conectada à entrada I_0 deste MUX.

A figura 4 permite acompanhar um primeiro esboço para esta solução preliminar.

2.2.2 Parte Lógica

Já o projeto da parte lógica é mais fácil no caso desta ULA e podemos incluí-lo já no diagrama em blocos completo da ULA.

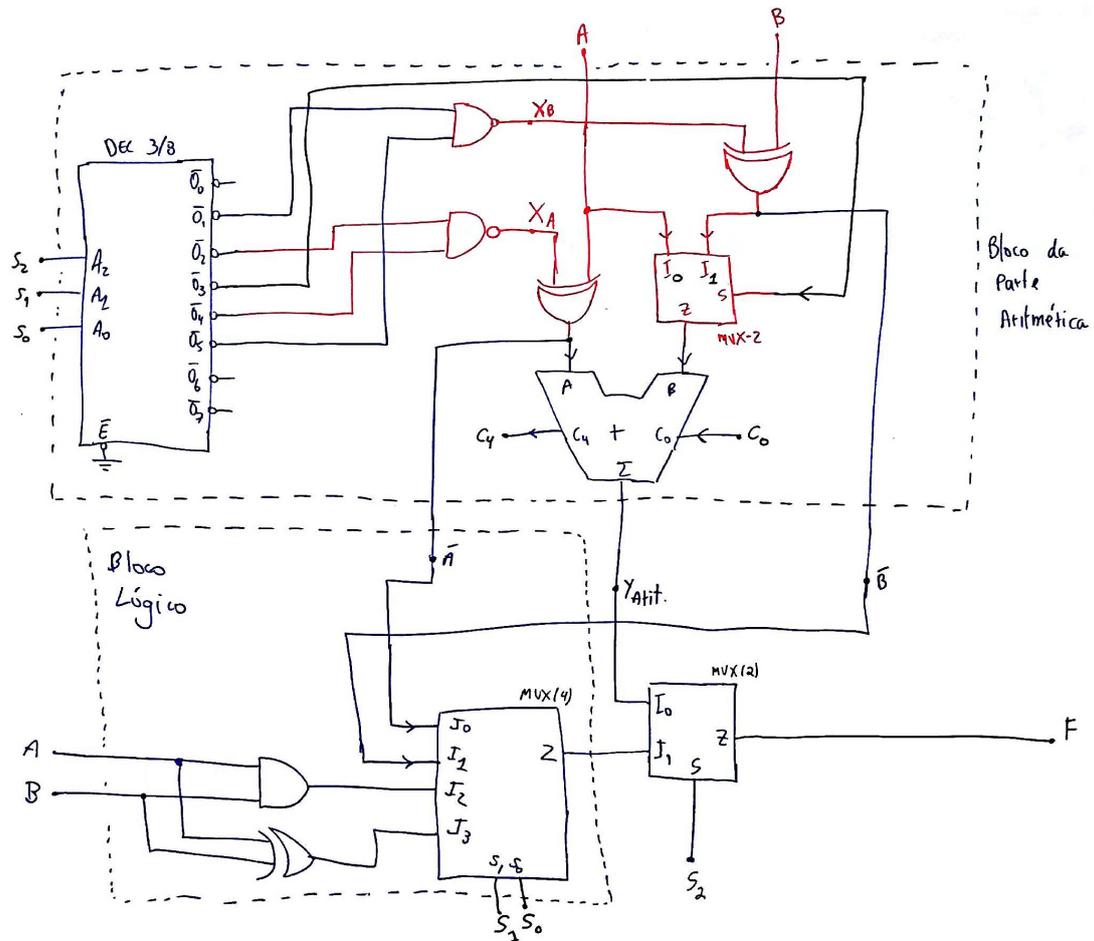


Figura 5: Diagrama em bloco (final) da ULA – opção 1.

2.3 Diagrama em Blocos Final

O diagrama em blocos completo da ULA é mostrado na figura 5.

Note que a entrada *Select* do MUX que decide se passa A ou B para a entrada B do Somador, poderia ser conectada à uma simples porta NAND conectada às entradas $S_1 S_0$. Desta forma, quando se faz necessário passar a palavra A para a entrada B do Somador ($S_2 S_1 S_0 = 011_{(2)} = 3_{(2)}$), $\overline{S_1 \cdot S_0} = 0$ (NAND(2)), conforme mostra a figura XX (note que o nível lógico de S_2 não importa para esta operação).

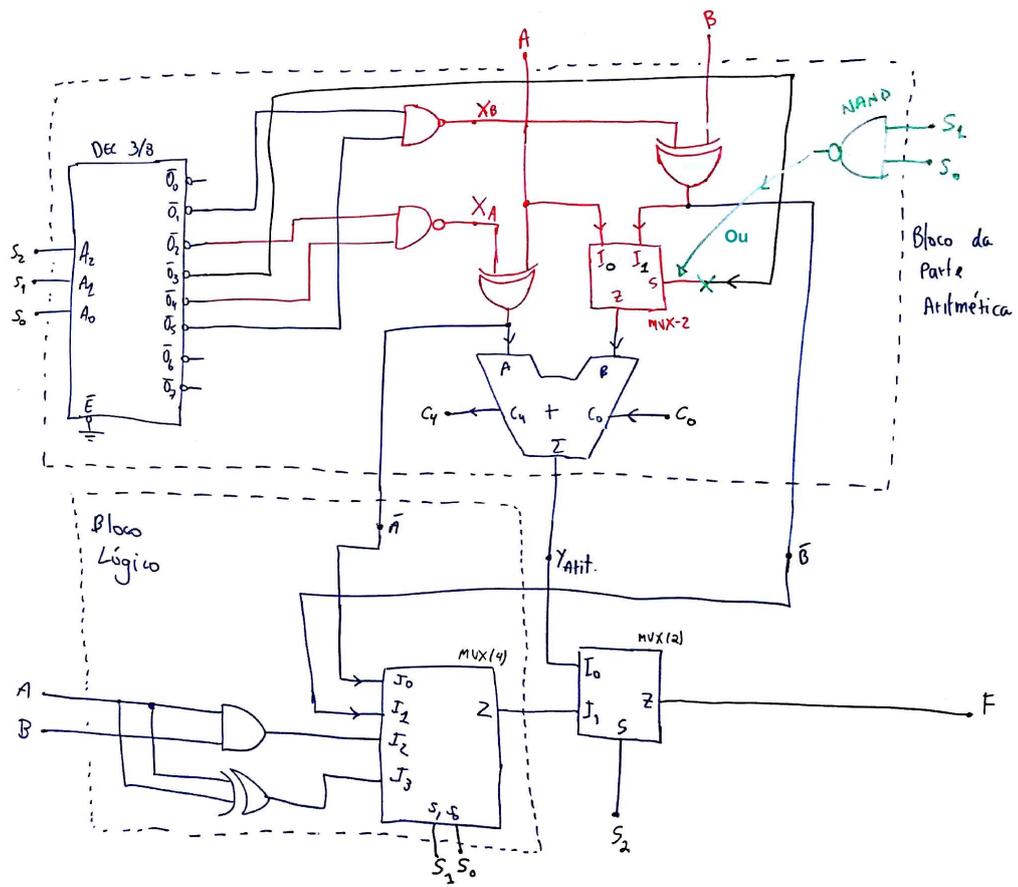


Figura 6: Diagrama final para a ULA, apresentando outra opção de solução associada com o acionamento do MUX atrelado à palavra B de entrada do Somador.

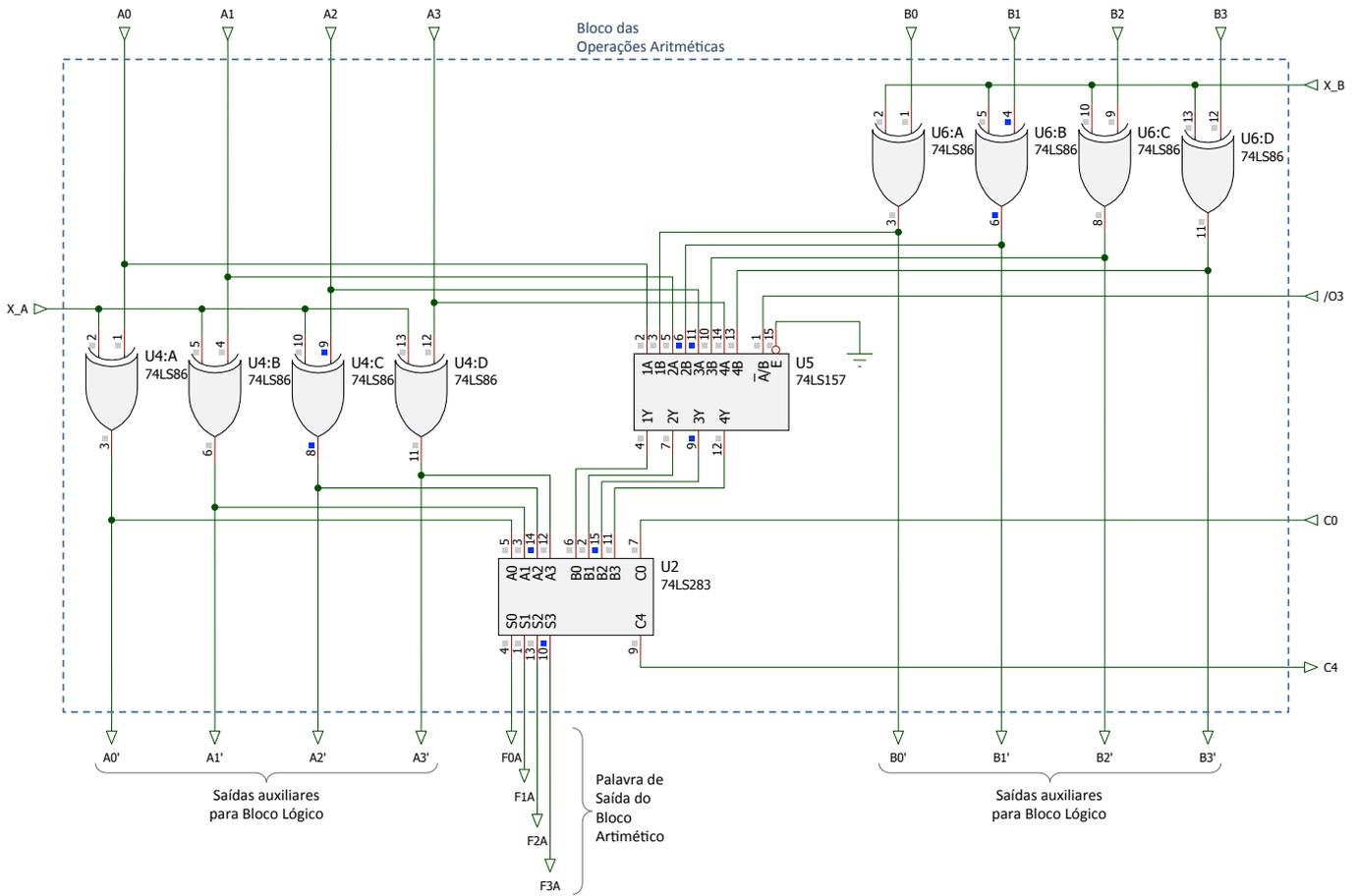


Figura 7: Diagrama elétrico correspondente ao bloco aritmético da ULA,

3 Diagrama Elétrico

A “tradução” para o diagrama em blocos apresentado na figura 5 aparece dividido em diferentes esquemas elétricos correspondentes à cada bloco básico (funcional) da ULA. Notar que fica difícil mostrar num único diagrama elétrico 100% do circuito de forma que o esquema ainda permaneça legível, motivo pelo qual, por motivo de clareza, o mesmo é dividido em diferentes blocos, mas notar que as conexões entre os blocos estão claramente determinadas.

A figura 7 mostra o diagrama elétrico que corresponde ao **Bloco Aritmético** desta ULA.

A figura 8 mostra o diagrama elétrico referente ao **Bloco Lógico** da ULA.

A figura 9 mostra os diagramas elétricos referente ao bloco que gera linhas de controle interna (a) e ao estágio de saída da ULA.

4 Resultados

A figura 10 mostra o caso de um simulação com: $S_2S_1S_0 = 011_{(2)} = 3_{(10)}$ que corresponde à operação: $F = 2A$. Notar que C_0 deve ser forçado à '0'.

Considerações Finais

Note que a solução apresentada não é a única possível. Mas tenta ser a mais ordenada e “limpa” possível.

Obs.: Este projeto foi desenvolvido entre os alunos de Circuitos Digitais I por ocasião do segundo semestre de 2016.

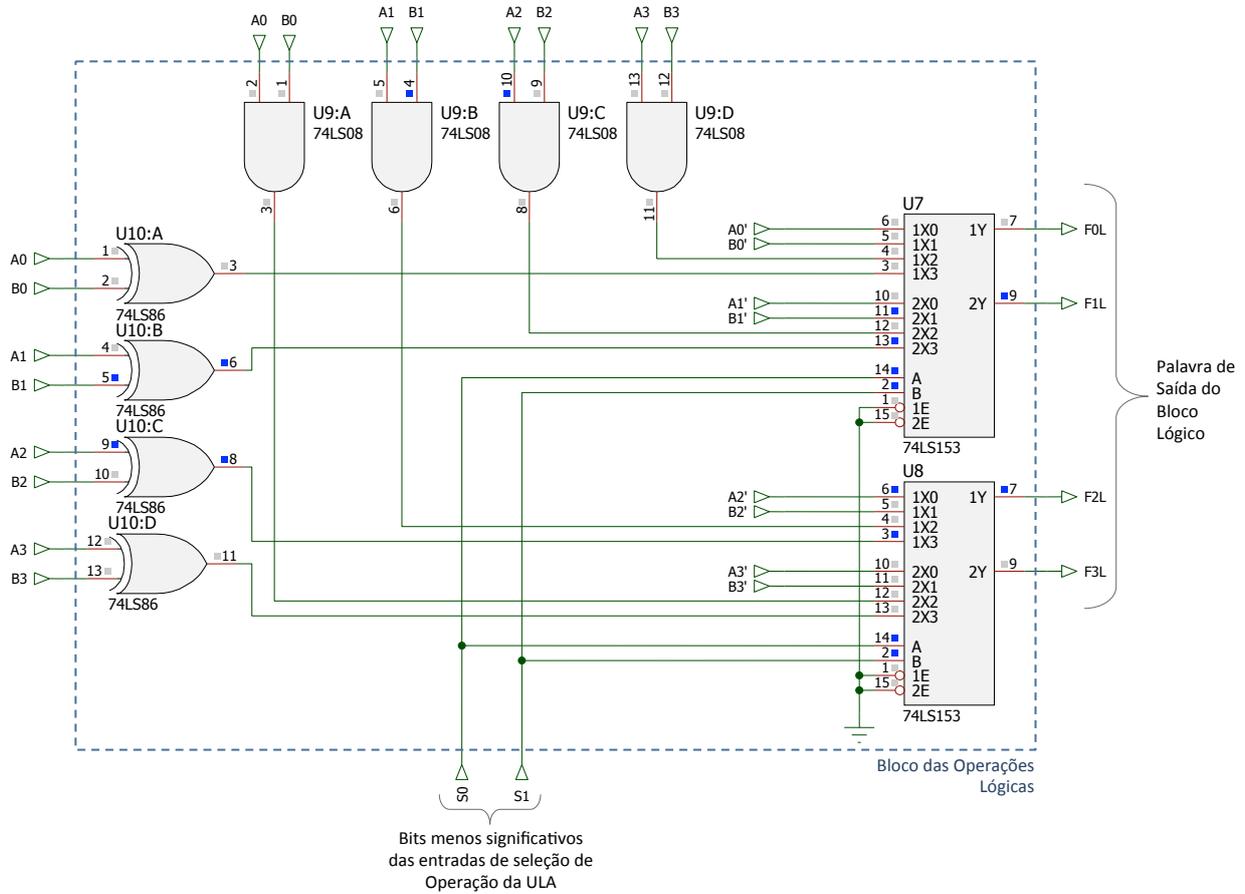
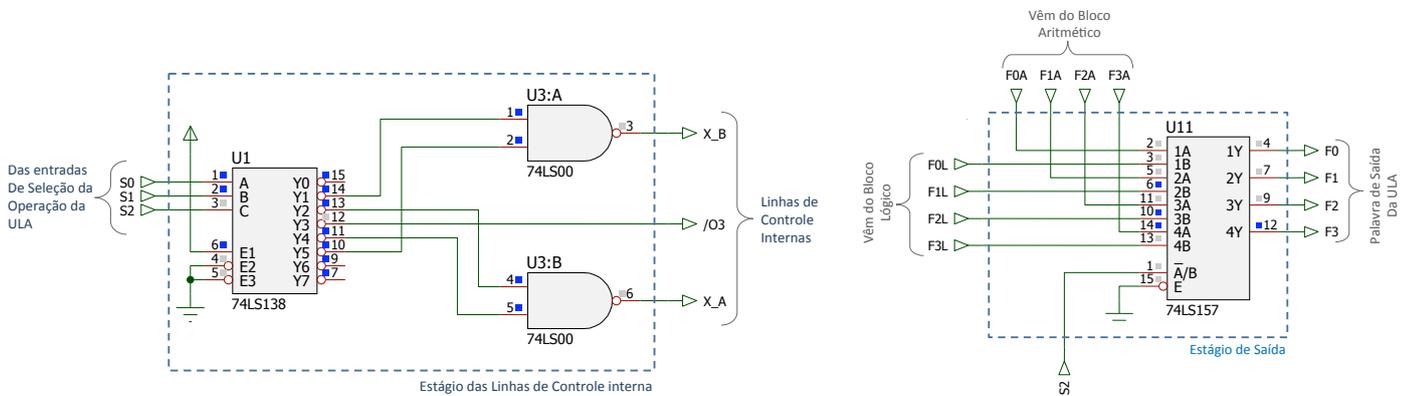


Figura 8: Diagrama elétrico referente à parte das operações lógicas da ULA.



(a) Parte referente à lógica de controle interna da ULA.

(b) Parte referente ao estágio de saída da ULA.

Figura 9: Outros estágios (blocos) da ULA.

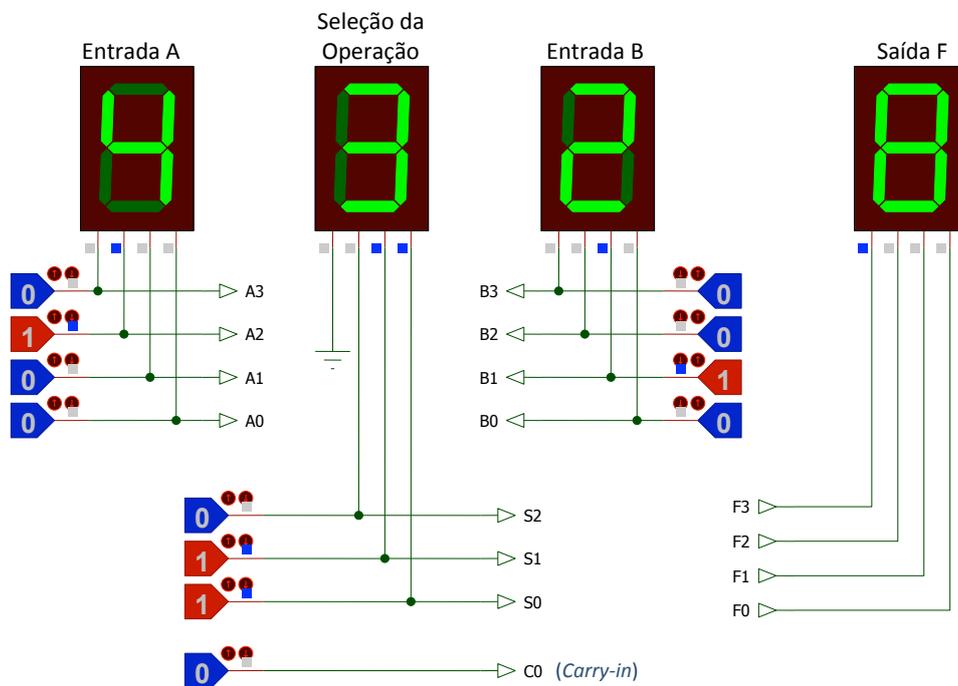


Figura 10: Simulação de uma das operações da ULA.