

Exercício Prático 2

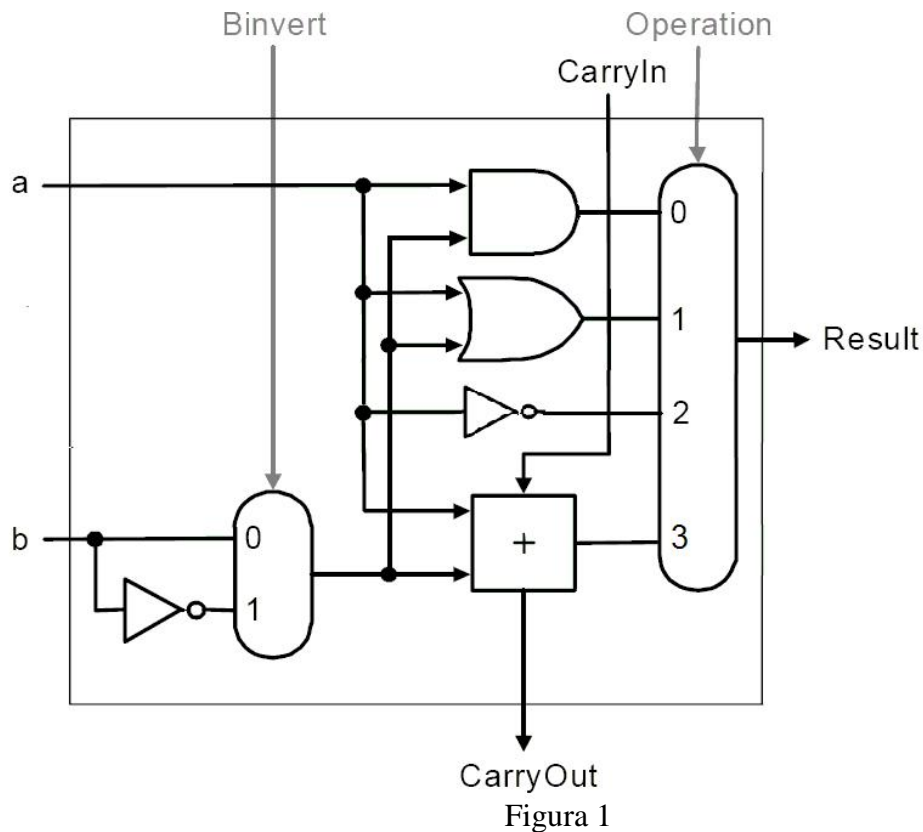
Laboratório de ac2

Objetivo:

Cosntruir uma Unidade Lógica e Aritmética (ULA) de 1 bit, 4 bits e implementar no Logisim.

Parte 1 (estudo da ULA usando Logisim):

1. Considere a Unidade Lógica e Aritmética de 1 bit ilustrada na Figura 1 a seguir:



2. Procure entender o esquema, principalmente a subtração.
3. Sua ULA possui a seguinte tabela de opcodes:

Op. Code (Operation)	Instrução (Result)
0	AND (a,b)
1	OR (a,b)
2	NOT (a)
3	SOMA(a,b)

4. Teste a sua ULA de acordo com o seguinte roteiro:

Início:

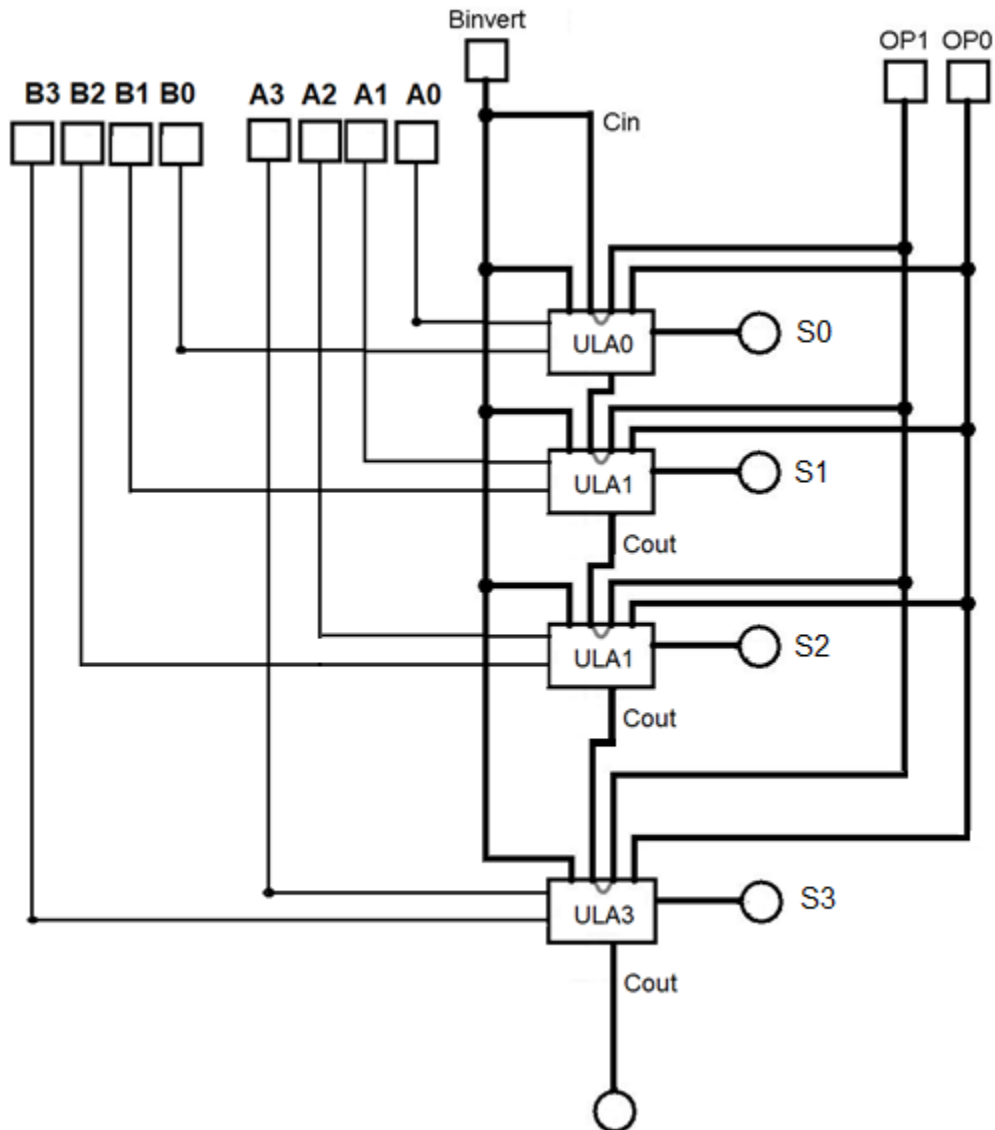
A=0;

B=1;

AND(A,B);

A=1;
 B=1;
 OR(A,B);
 SOMA(A,B);
 NOT(A);
 SOMA (A,-B);
Fim.

5. Usando essa ula de 1 bit, construa essa ULA para **4 bits** no Logisim e verifique o seu funcionamento. Veja como funciona o barramento de instruções (operation) e o barramento de dados (a e b). Observe a ligação do Binvert ao Carry_in da primeira ULA. Procure usar subcircuitos, seu circuito deverá estar como a figura a seguir:



6. Teste a sua ULA de acordo com o seguinte roteiro (considerando os números de 4 bits):

Início:

A=2; (ou A=0010)
B=1; (ou B=0001)
AND(A,B);
B=3; (ou B=0011)
OR(A,B);
SOMA(A,B);
A=12; (ou A=1100)
NOT(A);
B=13; (ou B=1101)
AND(B,A);

Fim.

Para o programa de teste acima, preencher a tabela a seguir considerando que cada linha corresponderá à execução de uma instrução (a primeira linha já foi realizada, observe que a palavra deverá conter 10 bits, para escrevermos em hexa completamos os dois bits à esquerda com zero):

Instrução realizada	Binário (A,B,Op.code)	Valor em Hexa (0x ...)	Resultado em binário
AND(A,B)	0010 0001 00	(0000 1000 0100) = 0x084	0000
OR(A,B)			
SOMA(A,B)			
NOT(A)			
AND(B,A)			

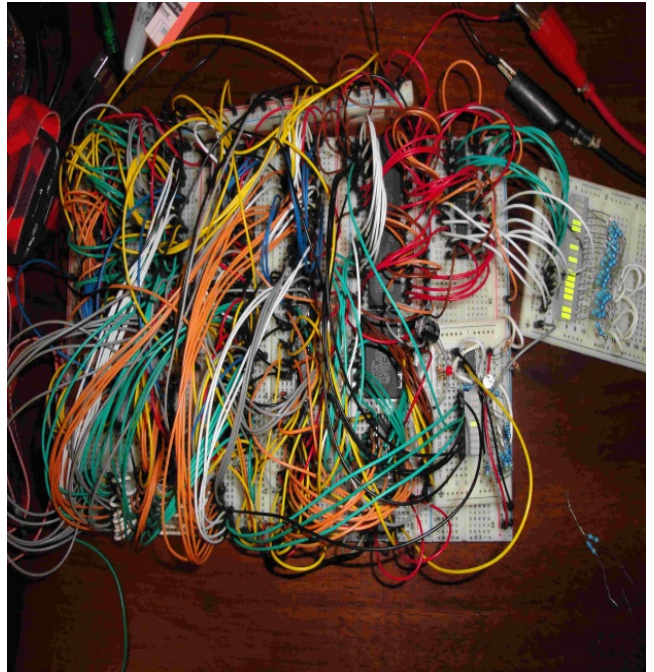
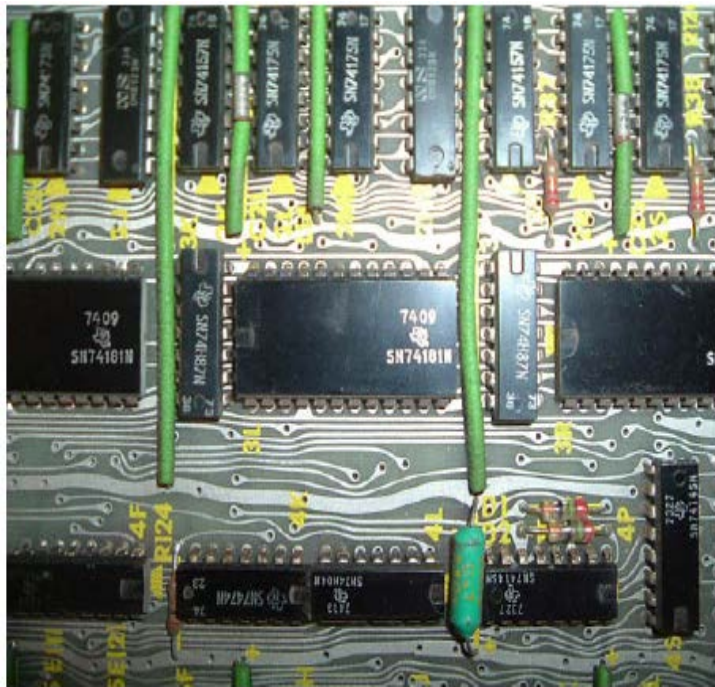
O que apresentar nesse exercício:

Parte 1 (Logisim)

- Os prints de todos os circuitos elaborados (ULA de 1 bit e ULA de 4 bits).
- Preencher a tabela de resultados para a unidade de 4 bits.
- Um print de cada um dos testes realizados (use o roteiro indicado no item 6).

Parte 2

Nesta experiência você irá projetar no logisim o circuito 74181, que foi inicialmente utilizado para a construção de computadores de 8 e 16 bits (conforme as figuras abaixo). Posteriormente iremos implementar uma ULA semelhante dentro do Arduino, por isso é importante conhecê-la.

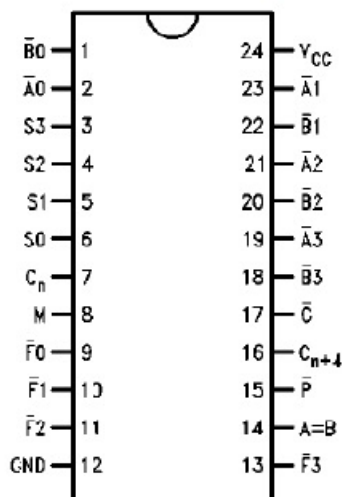


Como a ULA funciona.

A ULA a ser utilizada é a 74LS181, que possui 4 bits de controle e é uma ULA de 4 bits (saída). Portanto, opera sobre duas entradas de 4 bits. A distribuição dos pinos pode ser vista a seguir:

SELECTION				M = H LOGIC FUNCTIONS	ACTIVE-HIGH DATA	
S3	S2	S1	S0		M = L; ARITHMETIC OPERATIONS	
					$\bar{C}_n = H$ (no carry)	$\bar{C}_n = L$ (with carry)
L	L	L	L	$F = \bar{A}$	$F = A$	$F = A \text{ PLUS } 1$
L	L	L	H	$F = A + B$	$F = A + B$	$F = (A + B) \text{ PLUS } 1$
L	L	H	L	$F = \bar{A}B$	$F = A + \bar{B}$	$F = (A + \bar{B}) \text{ PLUS } 1$
L	L	H	H	$F = 0$	$F = \text{MINUS } 1 \text{ (2's COMPL)}$	$F = \text{ZERO}$
L	H	L	L	$F = \bar{A}B$	$F = A \text{ PLUS } \bar{A}B$	$F = A \text{ PLUS } \bar{A}B \text{ PLUS } 1$
L	H	L	H	$F = \bar{B}$	$F = (A + B) \text{ PLUS } \bar{A}B$	$F = (A + B) \text{ PLUS } \bar{A}B \text{ PLUS } 1$
L	H	H	L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$
L	H	H	H	$F = \bar{A}B$	$F = \bar{A}B \text{ MINUS } 1$	$F = \bar{A}B$
H	L	L	L	$F = \bar{A} + B$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$
H	L	L	H	$F = A \oplus B$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$
H	L	H	L	$F = B$	$F = (A + \bar{B}) \text{ PLUS } AB$	$F = (A + \bar{B}) \text{ PLUS } AB \text{ PLUS } 1$
H	L	H	H	$F = AB$	$F = AB \text{ MINUS } 1$	$F = AB$
H	H	L	L	$F = 1$	$F = A \text{ PLUS } A$	$F = A \text{ PLUS } A \text{ PLUS } 1$
H	H	L	H	$F = A + \bar{B}$	$F = (A + B) \text{ PLUS } A$	$F = (A + B) \text{ PLUS } A \text{ PLUS } 1$
H	H	H	L	$F = A + B$	$F = (A + \bar{B}) \text{ PLUS } A$	$F = (A + \bar{B}) \text{ PLUS } A \text{ PLUS } 1$
H	H	H	H	$F = A$	$F = A \text{ MINUS } 1$	$F = A$

Connection Diagram



Pin Descriptions

Pin Names	Description
$\bar{A}0-\bar{A}3$	Operand Inputs (Active LOW)
$\bar{B}0-\bar{B}3$	Operand Inputs (Active LOW)
$S0-S3$	Function Select Inputs
M	Mode Control Input
C_n	Carry Input
$\bar{F}0-\bar{F}3$	Function Outputs (Active LOW)
$A = B$	Comparator Output
G	Carry Generate Output (Active LOW)
\bar{P}	Carry Propagate Output (Active LOW)
C_{n+4}	Carry Output

Nessa primeira parte do experimento você deverá implementar toda a ULA no Logisim. Esta ULA permite a execução de instruções lógicas e aritméticas e permite que usemos entradas ativas em nível alto e em nível baixo. **Atenção:** Usaremos entradas em **nível alto**, conforme a tabela de funções ilustrada e iremos utilizar **apenas as instruções lógicas**.

O próximo passo será um projeto no LOGISIM dessa mesma ULA.

Agora você poderá utilizar os componentes presentes no Logisim (MUX, somadores, portas de múltiplas entradas, etc). Você também deverá utilizar o conceito de barramento para cada entrada e/ou saída, isso evitará um número muito grande de conexões.

Para um teste, você deverá dar um valor para A, um valor para B e executar todas as funções que a ALU permite através de S0, S1, S2 e S3. A saída da ALU deverá ser verificada nos pinos F0, F1, F2 e F3.

Iremos testar todas as funções lógicas da ULA da seguinte forma:

- criaremos uma palavra de 12 bits (os primeiros 4 bits para A0, A1, A2 e A3), os próximos 4 bits para B (B0, B1, B2 e B3) e os 4 bits finais para a operação desejada (S0, S2 e S3). O valor a ser preenchido da tabela será o resultado da operação.

Exemplo:

Instrução	Binário	Resultado da operação
4CB	010011001011	4

O significado da instrução é o seguinte (observe que escrevemos os valores em Hexadecimal para simplificar):

- O valor de A = 4 (ou **0100** em binário)
- O valor de B = C (ou **1100** em binário)
- A operação será B (ou **1011** em binário)

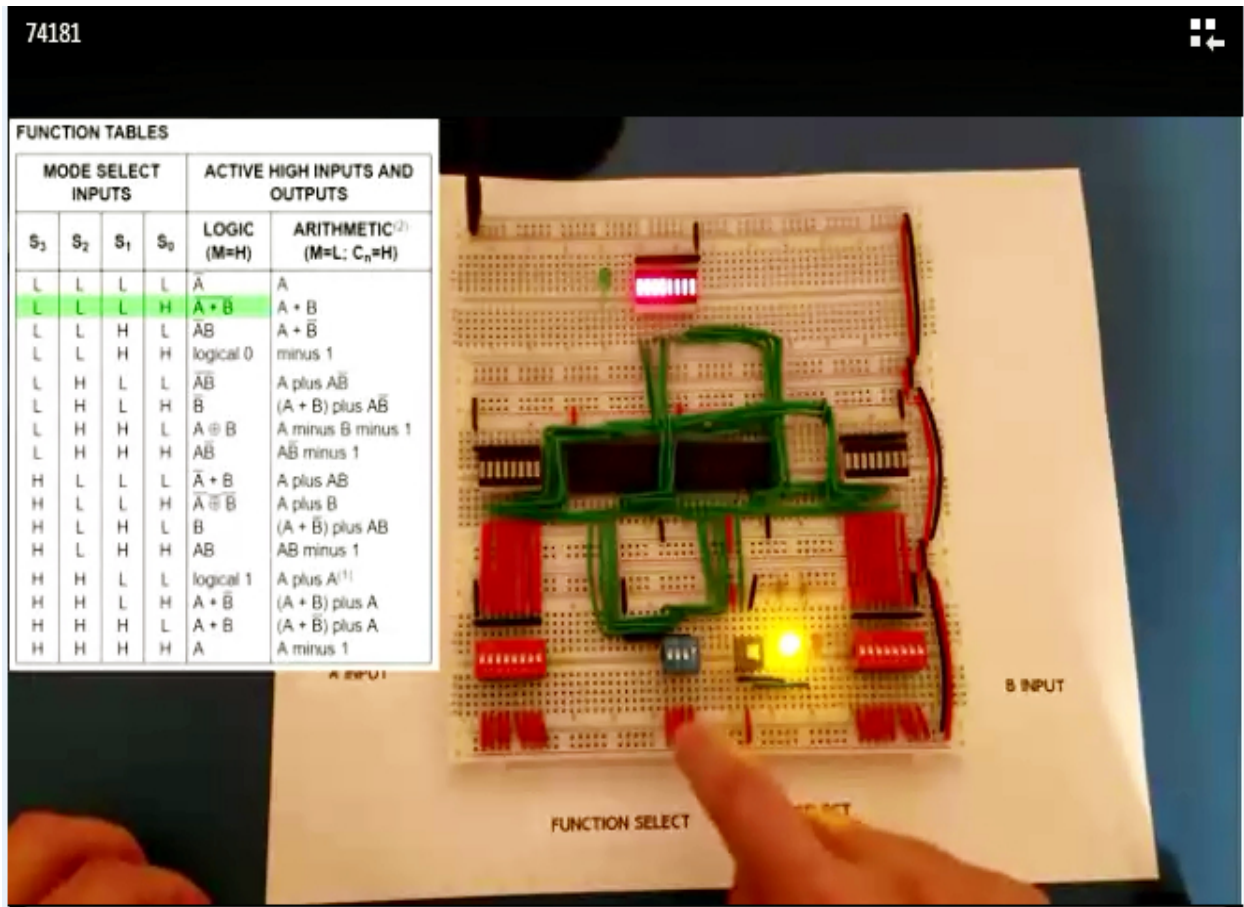
O que deveremos então fazer será a operação **1011** sobre os dados **0100** (que é o valor de A ou o primeiro operando) sobre **1100** (que é o valor de B ou o segundo operando).

Quando olhamos na tabela da ULA, a operação 1011 (ou H L H H) corresponde a $F = AB$, ou seja, a saída da ULA será o AND de A com B. Como $A=0100$ e $B=1100$, o AND de A e B será 0100, que é o resultado da operação e que deverá ser colocado na tabela ($0100 = 4$).

Complete agora a tabela a seguir onde todas as instruções que a ULA pode fazer serão testadas.

Instruções	Binário	Resultado da operação
450		
CB1		
A32		
C43		
124		
785		
9B6		
CD7		
FE8		
649		
D9A		
FCB		
63C		
98D		
76E		
23F		

A figura a seguir ilustra, apenas como exemplo, uma montagem real de uma ULA de 8 bits utilizando 2 circuitos 74181 e a implementação de uma função lógica $(A+B)'$, cujo opcode é LLLH ou 0001.



O que apresentar nesse relatório:

- 1) A tabela de teste com as funções da ULA completamente preenchida.
- 2) O projeto da ULA no Logisim com um printscreen de alguma instrução da tabela sendo executada.

Responder:

- Se o objetivo fosse realmente testar esta ULA, quantas linhas a nossa tabela verdade deveria ter, ou seja a tabela que você preencheu deveria ter quantas linhas?