

[View on GitHub](#)

## Circuitos Digitais

### Repo of Digital Circuits course - CRT0384

### PRÁTICA 08 - UNIDADE LÓGICA ARITMÉTICA

[Voltar à home](#)

#### OBJETIVOS

- Realizar a montagem de uma ULA de 4 bits

#### Material Necessário:

- 01 TTL 74F181
- Kit Digital

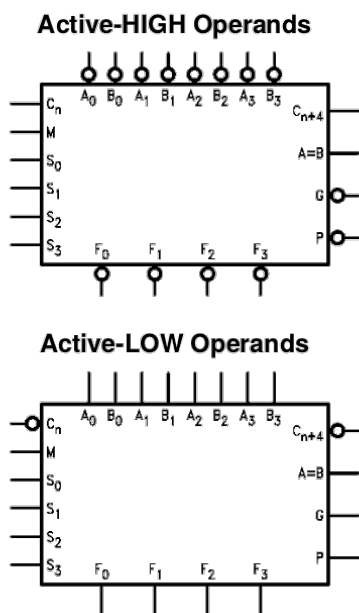
Como se pode observar, à medida que a complexidade das operações matemáticas é maior os circuitos necessários aumentam. Isso ocorre também com o aumento do número de bits envolvidos na operação. Para solucionar esses problemas foram desenvolvidos circuito integrados capazes de realizar diversas operações lógicas e aritméticas, envolvendo palavras de 4 ou 8 bits. Esse circuito é chamado de ULA - Unidade Lógica Aritmética (em inglês ALU - Arithmetic Logic Unit). Por esse nome também se designa o bloco interno responsável por operações lógicas e aritméticas em processadores e microcontroladores.

Uma ULA tipicamente tem duas palavras de entrada (4 ou 8 bits) e uma palavra de saída (4 ou 8 bits, respectivamente). A seleção da operação a ser realizada é feita através de entradas com esses fins. Adicionalmente, podem ser encontrados saídas que indicam se o resultado é igual a zero, se houve estouro da capacidade de representação, comparação se os valores de entrada são iguais, qual o maior, etc.

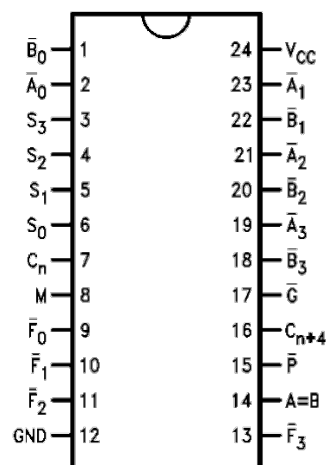
Quando se trata de operações aritméticas, as palavras de entradas são consideradas como valores inteiros, isto é, internamente existe carry que o resultado de um bit influencie o resultado do seguinte. No caso das operações lógicas, os bits são tratados individualmente, respeitando-se apenas a posição dos bits nas duas palavras.

O circuito 74LS181N implementa uma ULA de 4 bits para duas entradas e uma saída. A operação e modo de funcionamento são selecionados com as entradas S, como representado a seguir:

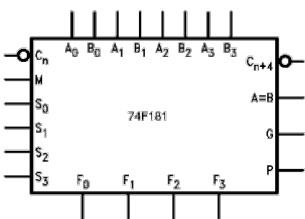
## Logic Symbols



## Connection Diagram



Pin Names	Description	U.L. HIGH/LOW	Input $I_{IH}/I_{IL}$ Output $I_{OH}/I_{OL}$
$\overline{A}_0\text{--}\overline{A}_3$	A Operand Inputs (Active LOW)	1.0/3.0	20 $\mu$ A/–1.8 mA
$\overline{B}_0\text{--}\overline{B}_3$	B Operand Inputs (Active LOW)	1.0/3.0	20 $\mu$ A/–1.8 mA
$S_0\text{--}S_3$	Function Select Inputs	1.0/4.0	20 $\mu$ A/–2.4 mA
M	Mode Control Input	1.0/1.0	20 $\mu$ A/–0.6 mA
$C_n$	Carry Input	1.0/5.0	20 $\mu$ A/–3.0 mA
$\overline{F}_0\text{--}\overline{F}_3$	Function Outputs (Active LOW)	50/33.3	–1 mA/20 mA
A = B	Comparator Output	OC (Note 1)/33.3	(Note 1)/20 mA
$\overline{G}$	Carry Generate Output (Active LOW)	50/33.3	–1 mA/20 mA
$\overline{P}$	Carry Propagate Output (Active LOW)	50/33.3	–1 mA/20 mA
$C_{n+4}$	Carry Output	50/33.3	–1 mA/20 mA

	$S_0$	$S_1$	$S_2$	$S_3$	Logic (M=H)	Arithmetic (M=L, $C_0$ =Inactive)	Arithmetic (M=L, $C_0$ =Active)
 <p><b>b. All Input Data True</b></p>	L	L	L	L	$\overline{A}$	A	A plus 1
	H	L	L	L	$\overline{A} + \overline{B}$	A + B	A + B plus 1
	L	H	L	L	$\overline{A} \cdot B$	A + $\overline{B}$	A + $\overline{B}$ plus 1
	H	H	L	L	Logic "0"	minus 1 (2s comp.)	Zero
	L	L	H	L	$\overline{A} \cdot \overline{B}$	A plus (A $\cdot$ $\overline{B}$ )	A plus A $\cdot$ $\overline{B}$ plus 1
	H	L	H	L	$\overline{B}$	A $\cdot$ $\overline{B}$ plus (A + B)	A $\cdot$ B plus (A + B) plus 1
	L	H	H	L	A $\oplus$ B	A minus B minus 1	A minus B
	H	H	H	L	A $\cdot$ $\overline{B}$	A $\cdot$ $\overline{B}$ minus 1	A $\cdot$ $\overline{B}$
	L	L	L	H	$\overline{A} + B$	A plus A $\cdot$ B	A plus A $\cdot$ B plus 1
	H	L	L	H	$\overline{A} \oplus \overline{B}$	A plus B	A plus B plus 1
	L	H	L	H	B	A $\cdot$ B plus (A + $\overline{B}$ )	A $\cdot$ B plus (A + $\overline{B}$ ) plus 1
	H	H	L	H	A $\cdot$ B	A $\cdot$ B minus 1	A $\cdot$ B
	L	L	H	H	Logic "1"	A plus A (2 $\times$ A)	A plus A (2 $\times$ A) plus 1
	H	L	H	H	A + $\overline{B}$	A plus (A + B)	A plus (A+B) plus 1
	L	H	H	H	A + B	A plus (A + $\overline{B}$ )	A plus (A+ $\overline{B}$ ) plus 1
	H	H	H	H	A	A minus 1	A

Observe o exemplo de configuração apresentada:

$s_3 \sim s_2 \sim s_1 \sim s_0 \sim = 1001$  - o circuito funciona como um somador

$s_3 \sim s_2 \sim s_1 \sim s_0 \sim = 1000$  - o circuito funciona como uma operação lógica ou "bitwise" (aplicada em cada bit)

OBS: o Datasheet apresenta mais configurações de funcionamento da ULA.

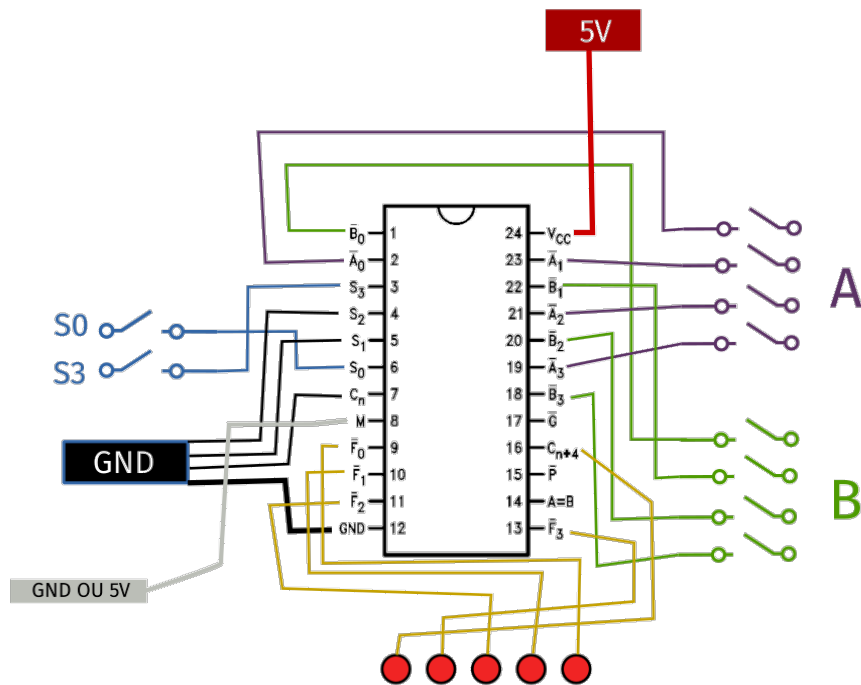
#### PARTE 1 - CONEXÃO DA ULA

Conecte a ULA com 4 bits em cada operando (A e B) ligadas nas chaves, e as entradas seletoras S de forma a representar as palavras 0000, 0001, 1000 e 1001. Portanto, iremos implementar as funções matemáticas e lógicas a seguir: (+ = ou lógico, plus = soma matemática)

L	L	L	L	$\overline{A}$	A
H	L	L	L	$\overline{A} + \overline{B}$	A + B
L	L	L	H	$\overline{A} + B$	A plus A $\cdot$ B
H	L	L	H	$\overline{A} \oplus \overline{B}$	A plus B

Por sua vez, Ligue  $s_1$  e  $s_2$  no GND e  $C_0$  em GND. Conecte o M (mode control) em GND, por enquanto.

Represente, como saída, as variáveis F3F2F1F0 e o bit de carry out  $C_{n+4}$



Escreva uma tabela verdade para cada configuração de entrada representada a seguir. Represente também a variável C(4)

Input A Input B M=0/ S=0000 M=0/ S=0001 M=0/ S=1000 M=0/ S=1001 ———— 0101 1100  
 1111 0011  
 1011 1011  
 0100 0101  
 1000 0111  
 1010 0010  
 1100 1110  
 0001 1011  
 1001 1111  
 1011 0000  
 0101 0010  
 0100 1010  
 0011 0010  
 0010 1010  
 1011 1110  
 0101 0101

Agora mude a configuração M para alto (M=1) e represente a nova tabela:

Input A Input B M=1/ S=0000 M=1/ S=0001 M=1/ S=1000 M=1/ S=1001 ———— 0101 1100  
 1111 0011  
 1011 1011  
 0100 0101  
 1000 0111  
 1010 0010  
 1100 1110  
 0001 1011  
 1001 1111  
 1011 0000  
 0101 0010  
 0100 1010  
 0011 0010  
 0010 1010  
 1011 1110  
 0101 0101

Circuitos Digitais maintained by [marcielbp](#)

Published with [GitHub Pages](#)