



## Atividade Prática 07 – Unidade Lógica e Aritmética



### OBJETIVOS:

- Utilizar o CI 74181 para executar operações lógicas e aritméticas.
- Aplicar a notação complemento de 2.
- Programar as entradas de controle da ULA para executar operações.



### MATERIAIS:

- 01 placa Arduino Uno
- 01 protoboard de 830 furos
- 14 resistores de  $330\ \Omega$  (laranja – laranja – marrom)
- 01 resistor de  $1\ k\Omega$  (marrom – preto – vermelho)
- 14 LEDs difusos
- Fios jumper macho-macho
- CI 74181



### ATENÇÃO:

- Para usar o pino 0 do Arduino como saída, desconecte o fio jumper dele antes de carregar o código no Arduino. Só depois você deve encaixar o fio jumper no pino 0.
- Conecte ao GND o terminal do LED alinhado ao chanfro.

O CI 74181 implementa uma Unidade Lógica e Aritmética (ULA) de 4 bits. Seu diagrama lógico e pinagem estão no Apêndice. Seus pinos de entrada, de saída e de controle são os seguintes:

#### Entradas

$A, B$	Entradas de dados, de 4 bits cada
$C_n$	<i>carry-input</i> , projetada em lógica invertida. Ou seja, ela é ativada pelo nível 0 (LOW) e desativada pelo nível 1 (HIGH)

#### Saídas

$F$	Resultado da operação, de 4 bits
$C_n + 4$	<i>carry-output</i> . Pode ser usada para detectar <i>overflow</i> . Seu resultado é exibido em lógica invertida, ou seja, o nível 0 (LOW) indica que houve <i>carry-out</i> e o nível 1 (HIGH) indica que não houve <i>carry-out</i> (transbordamento)
$A = B$	Saída de igualdade. O nível 1 indica que as palavras $A$ e $B$ são iguais
$G$ e $P$	Saídas de geração e propagação rápida de <i>carry</i> (não serão usadas nesta atividade)

#### Controle

$S_i$	Selecionam a operação a ser desempenhada pela ULA
$M$	Determina o modo de operação da ULA: lógico ( $M=1$ ) ou aritmético ( $M=0$ )

A Tabela Verdade da Figura 1 informa quais operações lógicas ou aritméticas são realizadas entre as duas entradas  $A$  e  $B$  para produzir o resultado  $F$ , todos de 4 bits de largura.

O Código 1 é um *modelo* de como testar o resultado de uma operação, determinada por uma combinação de seletores, a partir de um conjunto de entradas  $A$  e  $B$ . Nesse exemplo específico, o código configura os seletores da ULA para realizar a operação  $F = \overline{A+B}$  (linhas 5 e 8), sobre as entradas  $A = 0101$  e  $B = 1100$  (linhas 11 e 12).

Selection				Active High Data		
				M = H Logic Functions	M = L; Arithmetic Operations	
S3	S2	S1	S0		C <sub>n</sub> = H (no carry)	C <sub>n</sub> = L (with carry)
L	L	L	L	$F = \overline{A}$	$F = A$	$F = A \text{ Plus } 1$
L	L	L	H	$F = \overline{A+B}$	$F = A + B$	$F = (A + B) \text{ Plus } 1$
L	L	H	L	$F = \overline{AB}$	$F = A + \overline{B}$	$F = (A + \overline{B}) \text{ Plus } 1$
L	L	H	H	$F = 0$	$F = \text{Minus } 1 \text{ (2's Compl)}$	$F = \text{Zero}$
L	H	L	L	$F = \overline{AB}$	$F = A \text{ Plus } \overline{AB}$	$F = A \text{ Plus } \overline{AB} \text{ Plus } 1$
L	H	L	H	$F = \overline{B}$	$F = (A + B) \text{ Plus } \overline{AB}$	$F = (A + B) \text{ Plus } \overline{AB} \text{ Plus } 1$
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	$F = \overline{AB}$	$F = \overline{AB} \text{ Minus } 1$	$F = \overline{AB}$
H	L	L	L	$F = \overline{A+B}$	$F = A \text{ Plus } AB$	$F = A \text{ Plus } AB \text{ Plus } 1$
H	L	L	H	$F = \overline{A \oplus B}$	$F = A \text{ Plus } B$	$F = A \text{ Plus } B \text{ Plus } 1$
H	L	H	L	$F = B$	$F = (A + \overline{B}) \text{ Plus } AB$	$F = (A + \overline{B}) \text{ Plus } AB \text{ Plus } 1$
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	$F = A + \overline{B}$	$F = (A + B) \text{ Plus } A$	$F = (A + B) \text{ Plus } A \text{ Plus } 1$
H	H	H	L	$F = A + B$	$F = (A + \overline{B}) \text{ Plus } A$	$F = (A + \overline{B}) \text{ Plus } A \text{ Plus } 1$
H	H	H	H	$F = A$	$F = A \text{ Minus } 1$	$F = A$

Figura 1: Tabela Verdade da ULA implementada pelo CI 74181.



### ATENÇÃO:

- Não confunda a operação lógica **OU**, indicada pelo sinal + na Figura 1, com a operação aritmética da **adição**, indicada por Plus.
- A subtração aritmética é indicada por Minus.

O circuito da Figura 2 mostra como a ULA deve ser conectada ao Arduino. Usando o Código 1, você fará com que o Arduino controle os seguintes sinais:

- Entradas de dados  $A$  (pinos 0–3) e  $B$  (pinos 4–7).
- Modo de operação  $M$  (pino 8).
- *Carry-in* (pino 9).
- Seletores de função  $S_3$ – $S_0$  (pinos 10–13).

Além da saída de dados  $F$ , o CI 74181 exibe duas saídas de sinalização:

- *Carry-out* ( $C_n + 4$ ): indica se houve transbordamento de bits (nível LOW); e
- $A = B$ : indica se as duas palavras de entrada são iguais (nível HIGH), mas *só é válido quando a operação de comparação está selecionada*.

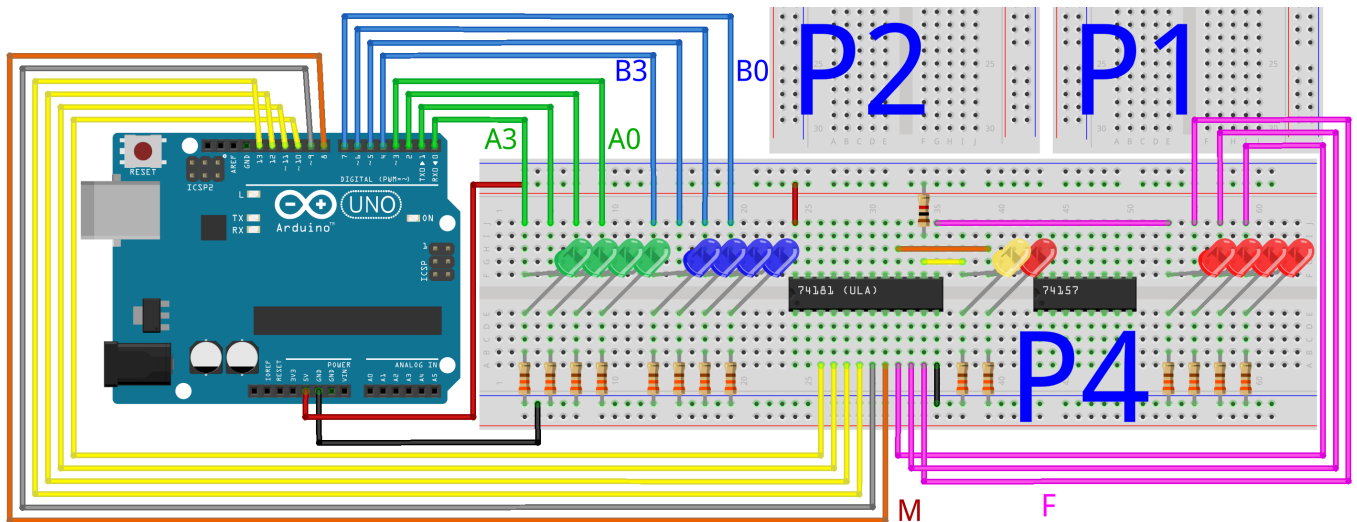


Figura 2: Conexão *parcial* do Arduino com o CI 74181. *Complete a ligação com fios jumper entre os pinos 0–7 do Arduino e as entradas A e B da ULA.*

## TAREFAS

- Monte o circuito da Figura 2 na **protoboard P4**, consultando a pinagem do CI 74181, no Apêndice. Dicas:
  - Conecte o CI 74181 com cuidado para não entortar os seus terminais.
  - Conecte os fios de alimentação, depois os resistores de 330 k $\Omega$  e os LEDs.
  - Use um resistor de 1 k $\Omega$  para conectar a saída  $A = B$  da ULA ao Vcc.
  - Conecte os pinos 0–3 do Arduino às entradas  $A_3$ – $A_0$  da ULA.
  - Conecte os pinos 4–7 do Arduino às entradas  $B_3$ – $B_0$  da ULA.
  - Conecte o pino 8 do Arduino ao seletor de modo  $M$  da ULA.
  - Conecte o pino 9 do Arduino ao *carry-in*  $C_n$ .
  - Conecte os pinos 10–13 do Arduino aos seletores  $S_0$ – $S_3$  da ULA.
  - Conecte as saídas  $F_0$ – $F_3$  da ULA ao polo *positivo* dos LEDs vermelhos.
  - Os LEDs azuis e verdes servem para indicar, em dado momento, que nível de tensão (HIGH ou LOW) o Arduino está enviando às entradas da ULA.



### ATENÇÃO:

O CI 74181 exige da fonte de alimentação uma maior intensidade de corrente do que outros CIs. Por isso, **use as trilhas horizontais da protoboard P4 para alimentar a ULA.**

- Para cada Experimento identificado por 01 a 06 na **Folha de Respostas**, proceda assim:
  - Identifique qual a operação da ULA a ser examinada.
  - Consultando a tabela verdade da ULA (Figura 1), determine quais valores devem ser injetados nos seletores  $S_i$ ,  $M$  e  $C_n$  a fim de executar a operação examinada. Em alguns casos, pode haver mais de uma resposta correta.

- (c) Modifique o Código 1 de acordo com os valores determinados no passo anterior. Carregue o novo código no Arduino e observe os resultados.
- (d) Responda as perguntas contidas na **Folha de Respostas**, conforme os valores *exibidos* nas saídas  $A = B$ ,  $C_n + 4$ , e  $F$ . No caso da saída  $C_n + 4$ , informe o valor exibido no LED, sem se importar que ela utiliza lógica invertida.
3. Desenhe um *diagrama lógico*, usando o bloco básico exibido no Apêndice, mostrando como uma ou mais unidades da ULA 74181 devem ser usadas para implementar os seguintes circuitos. **Ignore as saídas G, P e  $A = B$ , e as entradas de alimentação do CI ( $V_{cc}$  e GND).**
- (a) Um MUX  $8 \times 4$ .
- (b) Uma ULA de 8 bits.

Código 1: Código que realiza a função  $F = \overline{A + B}$ , para as entradas  $A = 0101$  e  $B = 1100$ .

```

1  /* Seletor de Modo: HIGH - Logico; LOW - Aritmetico */
2  byte M = HIGH;
3
4  /* Seletor de Modo: HIGH - sem carry-in; LOW - tem carry-in */
5  byte Cin = LOW;
6
7  /* Seletores de funcao: S3, S2, S1, S0 */
8  byte S[] = {LOW, LOW, LOW, HIGH};
9
10 /* Entradas de dados */
11 byte A[] = {LOW, HIGH, LOW, HIGH}; /* Entrada A, de 4 bits */
12 byte B[] = {HIGH, HIGH, LOW, LOW}; /* Entrada B, de 4 bits */
13
14 /* a funcao 'setup' eh executada apenas uma vez */
15 void setup() {
16     /* ajusta os pinos 0 a 13 como saida */
17     for (byte i = 0; i <= 13; i++)
18         pinMode(i, OUTPUT);
19 }
20
21 /* a funcao 'loop' eh executada indefinidamente */
22 void loop() {
23     /* Ajusta o modo da ULA (pino 8) */
24     digitalWrite(8, M);
25
26     /* Ajusta o Cin (pino 9) */
27     digitalWrite(9, Cin);
28
29     /* Ajusta os valores de:
30     - palavra de dados A0 A1 A2 A3: pinos 0-3
31     - palavra de dados B0 B1 B2 B3: pinos 4-7
32     - seletores S3 S2 S1 S0: pinos 10-13
33     */
34     for (byte i = 0; i < 4; i++) {
35         digitalWrite(0 + i, A[i]);
36         digitalWrite(4 + i, B[i]);
37         digitalWrite(10 + i, S[i]);
38     }
39 }

```

## Apêndice – Pinagem dos CIs utilizados

### ULA 74181

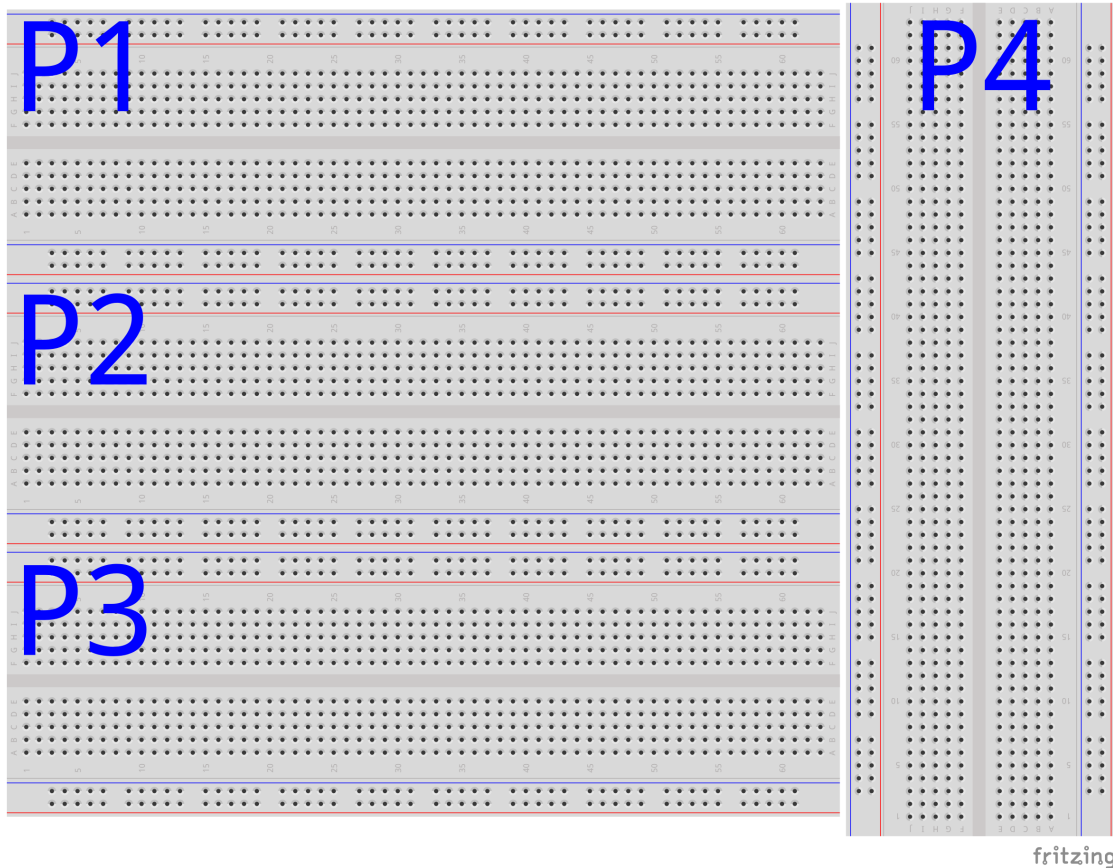
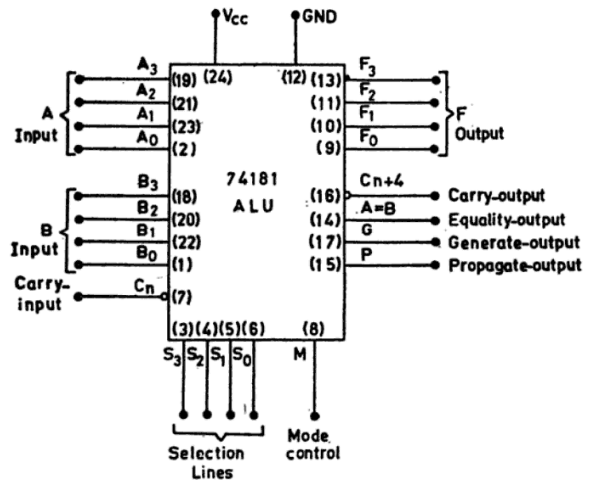
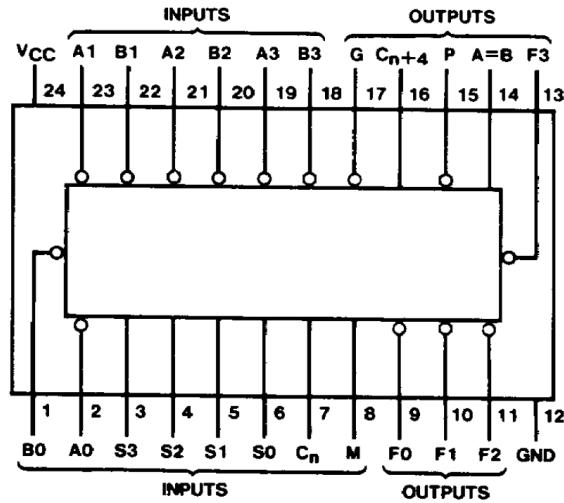


Figura 3: Orientação das protoboards.