

# Unidade Lógica e Aritmética

J.L.R.B. e P.S.C./2001 (revisão)  
 E.T.M./2002 (revisão e adaptação)  
 E.T.M./2003 (revisão)  
 E.T.M./2005 (revisão)  
 E.T.M./2008 (revisão)  
 E.T.M./2011 (revisão)

## **RESUMO**

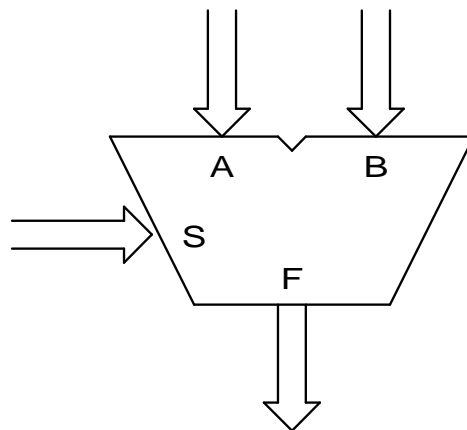
Esta experiência tem por objetivo a familiarização com o funcionamento da Unidade Lógica e Aritmética (ULA), especificamente o circuito integrado 74181, que executa 16 funções aritméticas e 16 funções lógicas.

A parte experimental inclui o projeto, montagem e documentação de um circuito baseado na ULA 74181, com a sequência de operações determinada pelo conteúdo de uma memória de acesso aleatório (RAM), o circuito integrado 74189.

## **1. PARTE TEÓRICA**

### **1.1. Unidade Lógica e Aritmética - ULA**

A Unidade Lógica e Aritmética (ULA) é um circuito combinatório responsável pela execução de somas, subtrações e funções lógicas, em um sistema digital. Na figura 1.1 é mostrado um esquema simplificado de uma ULA.



**Figura 1.1 - Esquema Simplificado de uma ULA.**

A operação que deve ser executada com os dados de entrada (A e B) é determinada pelos sinais de controle (S) e o resultado é obtido na saída (F). A complexidade da ULA é proporcional à complexidade do sistema em que será utilizada; assim, sistemas simples permitem o uso de ULAs simples e sistemas sofisticados exigem ULAs sofisticadas.

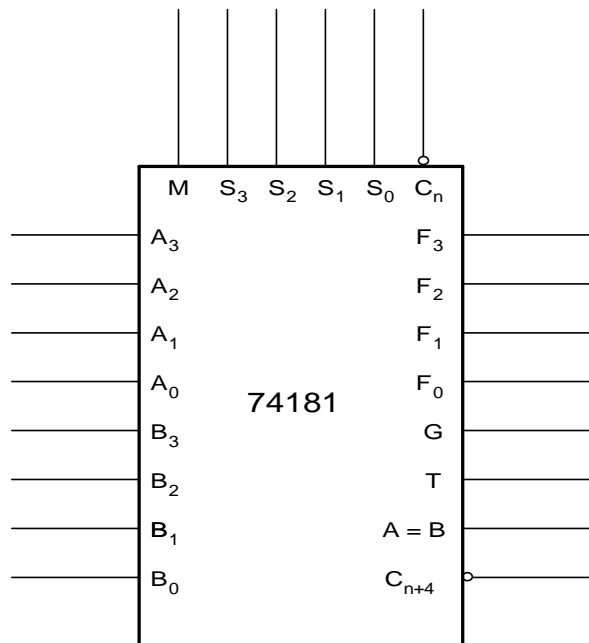
Uma vez estabelecido o porte do sistema, existe também o compromisso entre velocidade e preço. Por exemplo, as calculadoras eletrônicas exigem ULAs que permitem operações complexas, porém com velocidade de operação baixa, reduzindo-se o custo; já os computadores de grande porte exigem velocidade de operação elevada, aumentando o custo da ULA.

Atualmente têm-se várias alternativas de circuitos integrados que incluem uma ULA. Entre elas estão o 74181, 74381 e 74LS881.

A seguir será apresentada uma descrição de uma ULA integrada, o circuito integrado 74181.

## 1.2. Circuito Integrado 74181 – uma ULA de 4 bits

O circuito integrado MSI 74181 é uma ULA de 4 bits que tem possibilidade de executar 16 operações aritméticas binárias e 16 operações lógicas. A figura 1.2 apresenta um diagrama simplificado deste circuito integrado.



**Figura 1.2. - Diagrama Simplificado da ULA 74181.**

As tabelas I e II a seguir apresentam a descrição dos pinos e as operações da ULA, respectivamente.

**Tabela I - Descrição dos pinos da ULA 74181.**

| Pinos                         | Tipo    | Descrição                                                                             |
|-------------------------------|---------|---------------------------------------------------------------------------------------|
| $A_0$ a $A_3$ e $B_0$ a $B_3$ | Entrada | Dados de entrada                                                                      |
| $C_n$                         | Entrada | Bit de "vem um"                                                                       |
| $S_0$ a $S_3$                 | Entrada | Seleção da operação                                                                   |
| M                             | Entrada | Modo de operação:<br>M=0 - para operações aritméticas<br>M=1 - para operações lógicas |
| $F_0$ a $F_3$                 | Saída   | Dados de saída (resultado)                                                            |
| $C_{n+4}$                     | Saída   | Bit de "vai-um"                                                                       |
| G e T                         | Saídas  | Utilizadas para expansão utilizando <i>carry look-ahead</i>                           |
| A=B                           | Saída   | Indica igualdade das duas entradas                                                    |

Tabela II - Sinais e Operações da ULA 74181.

| Seleção        |                |                |                | Funções Lógicas ( M = 1 )          | Funções Aritméticas ( M = 0 )             |                                               |
|----------------|----------------|----------------|----------------|------------------------------------|-------------------------------------------|-----------------------------------------------|
| S <sub>3</sub> | S <sub>2</sub> | S <sub>1</sub> | S <sub>0</sub> |                                    | C <sub>n</sub> = 1 (sem carry)            | C <sub>n</sub> = 0 (com carry)                |
| 0              | 0              | 0              | 0              | $F = \bar{A}$                      | $F = A$                                   | $F = A + 1$                                   |
| 0              | 0              | 0              | 1              | $F = \overline{(A \text{ OR } B)}$ | $F = A \text{ OR } B$                     | $F = (A \text{ OR } B) + 1$                   |
| 0              | 0              | 1              | 0              | $F = \bar{A} \cdot B$              | $F = A \text{ OR } \bar{B}$               | $F = A \text{ OR } \bar{B} + 1$               |
| 0              | 0              | 1              | 1              | $F = 0$                            | $F = -1 (*)$                              | $F = 0$                                       |
| 0              | 1              | 0              | 0              | $F = \bar{A} \cdot \bar{B}$        | $F = A + A \cdot \bar{B}$                 | $F = A + A \cdot \bar{B} + 1$                 |
| 0              | 1              | 0              | 1              | $F = \bar{B}$                      | $F = (A \text{ OR } B) + A \cdot \bar{B}$ | $F = (A \text{ OR } B) + A \cdot \bar{B} + 1$ |
| 0              | 1              | 1              | 0              | $F = A \oplus B$                   | $F = A - B - 1$                           | $F = A - B$                                   |
| 0              | 1              | 1              | 1              | $F = A \cdot \bar{B}$              | $F = A \cdot \bar{B} - 1$                 | $F = A \cdot \bar{B}$                         |
| 1              | 0              | 0              | 0              | $F = \bar{A} \text{ OR } B$        | $F = A + A \cdot B$                       | $F = A + A \cdot B + 1$                       |
| 1              | 0              | 0              | 1              | $F = \overline{(A \oplus B)}$      | $F = A + B$                               | $F = A + B + 1$                               |
| 1              | 0              | 1              | 0              | $F = B$                            | $F = A \text{ OR } \bar{B} + A \cdot B$   | $F = A \text{ OR } \bar{B} + A \cdot B + 1$   |
| 1              | 0              | 1              | 1              | $F = A \cdot B$                    | $F = A \cdot B - 1$                       | $F = A \cdot B$                               |
| 1              | 1              | 0              | 0              | $F = 1$                            | $F = A + A$                               | $F = A + A + 1$                               |
| 1              | 1              | 0              | 1              | $F = A \text{ OR } \bar{B}$        | $F = (A \text{ OR } B) + A$               | $F = (A \text{ OR } B) + A + 1$               |
| 1              | 1              | 1              | 0              | $F = A \text{ OR } B$              | $F = A \text{ OR } \bar{B} + A$           | $F = A \text{ OR } \bar{B} + A + 1$           |
| 1              | 1              | 1              | 1              | $F = A$                            | $F = A - 1$                               | $F = A$                                       |

\*  $(-1)_{10}$  é representado por  $(1111)_2$  em Complemento de 2.

As saídas G e T são os sinais "gerador de vai-um", correspondentes ao bit mais significativo e, utilizando-se o circuito integrado 74182, *look-ahead carry generator*, permitem a expansão da largura da palavra a ser manipulada.

As operações de subtração são executadas em complemento de dois (C2). Por exemplo,  $(-1)$  é representado por  $(1111)$ . As operações lógicas são executadas bit a bit. Por exemplo, se a operação AND é aplicada às entradas  $A = 1011$  e  $B = 0110$ , resulta  $F = 0010$ .

O resultado de uma operação de comparação é apresentado na saída  $A = B$ . Para tanto, deve-se executar a operação  $A - B - 1$  com  $C_n = 1$ ; se as duas entradas são iguais, a saída  $A = B$  toma o valor 1. Esta saída tem a configuração *open-collector* para que seja possível implementar uma função *wired-and* dentre duas ou mais saídas deste tipo pertencentes a diferentes ULAs, quando se deseja expandir a largura da palavra a ser manipulada.

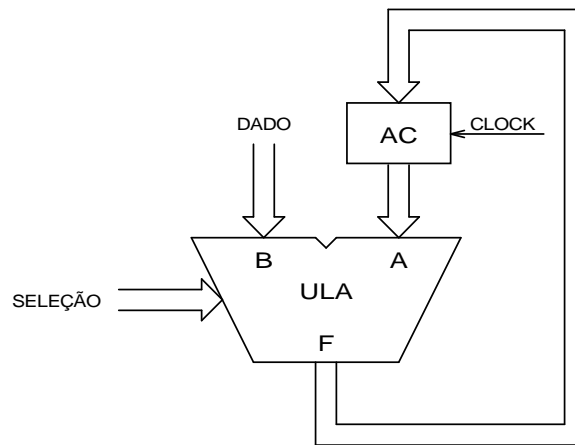
A saída  $C_{n+4}$  representa o sinal de vai-um do último bit da palavra. Ela pode ser usada para propagar o vai-um para o próximo estágio quando não há preocupação com a velocidade do circuito. O sinal  $C_{n+4}$  também pode ser usado em conjunto com a saída  $A = B$  para indicar as condições  $A > B$  e  $A < B$ .

OBSERVAÇÃO: Na realidade, em se tratando de um circuito combinatório, a ULA 74181 pode trabalhar com operandos representados em lógica positiva ou negativa. A Tabela II acima mostra o significado dos bits de seleção de operações quando se considera o uso de lógica positiva. Consulte o manual (*datasheet*) do componente para o caso do uso de lógica negativa.

Nesta experiência adotaremos a convenção de lógica positiva, portanto desconsidere eventuais referências ao uso de lógica negativa nos pinos da ULA no Quartus II.

### 1.3. Um Fluxo de Dados Simples utilizando a ULA 74181

A figura 1.3 mostra um fluxo de dados simples utilizando o circuito integrado 74181. De uma maneira geral, este circuito é encontrado como a base do fluxo de dados de vários microprocessadores.



**Figura 1.3 - Fluxo de dados de um circuito simples utilizando o 74181.**

Neste caso, todas as operações da ULA são executadas entre o conteúdo do acumulador (AC) e o conteúdo da via (DADO). Assim, por exemplo, uma operação de soma de dois números  $X + Y$  é executada seguindo-se os seguintes passos:

| Passo | $S_3$ | $S_2$ | $S_1$ | $S_0$ | M | $C_n$ | B  | Comentário            |
|-------|-------|-------|-------|-------|---|-------|----|-----------------------|
| 1     | 0     | 0     | 1     | 1     | 1 | d*    | d* | AC $\leftarrow$ 0     |
| 2     | 1     | 0     | 1     | 0     | 1 | d*    | X  | AC $\leftarrow$ X     |
| 3     | 1     | 0     | 0     | 1     | 0 | 1     | Y  | AC $\leftarrow$ X + Y |

\* OBS: **d** significa que o valor não importa (do inglês "don't care").

Após cada passo deve-se gerar um pulso no sinal CLOCK para forçar a cópia de saída da ULA no acumulador.

Operações mais complexas poderão ser executadas com outras sequências de operações simples como aquelas relacionadas na Tabela III. De uma maneira geral, estas operações simples podem ser classificadas em operações lógicas e aritméticas e operações de transferência de registradores.

## 1.4. Memória de Acesso Aleatório 74189 (RAM)

O circuito integrado 74189 é uma memória de 64 bits, organizados em 16 palavras de 4 bits. Os sinais de controle e dados são mostrados na figura 1.4, onde tem-se que:

- $D_0$  a  $D_3$  - dado de entrada (4 bits);
- $A_0$  a  $A_3$  - endereço (4 bits);
- $\overline{CE}$  - sinal de seleção da pastilha (*chip-enable*);
- $\overline{WE}$  - sinal de controle para armazenamento (*write-enable*);
- $\overline{O}_0$  a  $\overline{O}_3$  - saída de dados (4 bits, negada, *tri-state*);

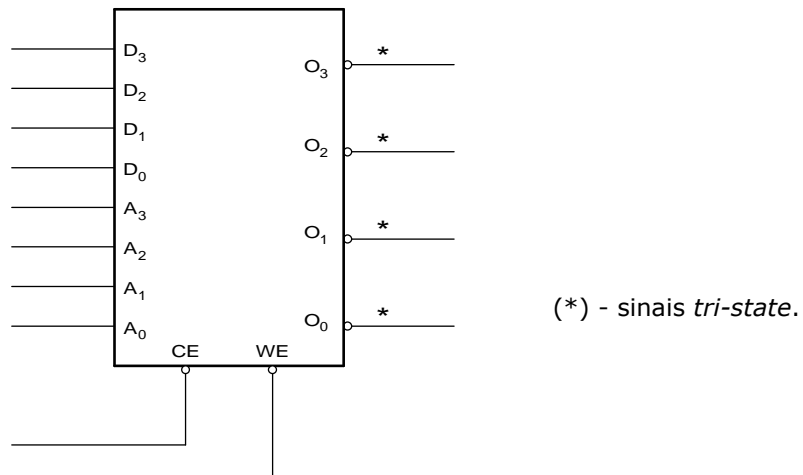


Figura 1.4 - Circuito Integrado 74189.

Para realizar uma **leitura** de uma posição de memória, deve-se primeiro colocar o endereço nas entradas  $A_0$  a  $A_3$  e depois colocar o sinal  $\overline{CE}$  em "0", mantendo  $\overline{WE}$  em "1".

Para realizar uma **escrita** em uma posição, além de primeiro endereçar a palavra e colocar o sinal  $\overline{CE}$  em "0", deve-se em seguida colocar o dado a ser escrito nas entradas  $D_0$  a  $D_3$  e depois gerar um pulso negativo na entrada  $\overline{WE}$ .

A figura 1.5 mostra o diagrama de tempos para um ciclo de leitura e um de escrita. Consulte o manual do circuito integrado para mais detalhes sobre a sequência de acionamento de sinais.

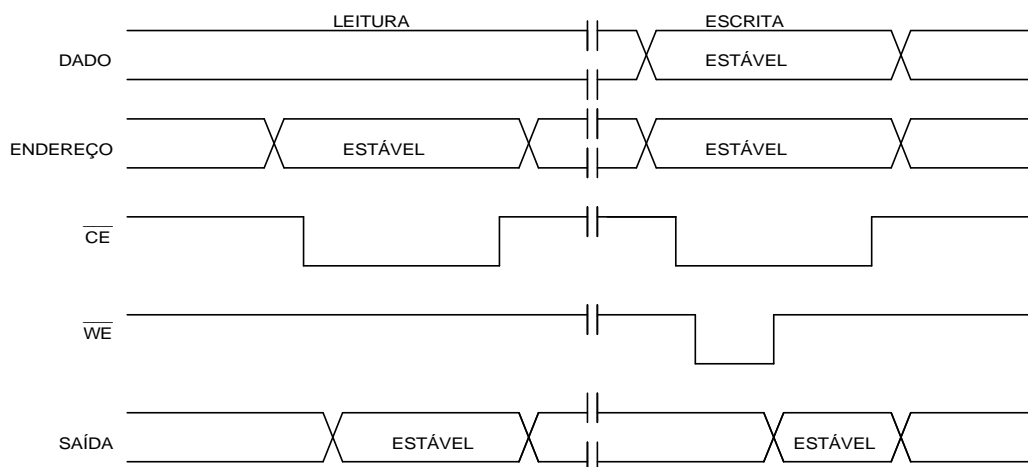


Figura 1.5 - Diagrama de Tempos dos Sinais do Circuito 74189 para os ciclos de leitura e escrita.

**OBS:** Existem diferenças técnicas entre as memórias 7489, 74189, 74289. Consulte os respectivos manuais.

## 2. PARTE EXPERIMENTAL

Na parte prática desta experiência utilizam-se extensamente os circuitos integrados 74181 e 74189. Recomenda-se analisar exaustivamente o funcionamento destes circuitos integrados.

### 2.1. Projeto: Circuito com a Unidade Lógica e Aritmética 74181 e a RAM 74189

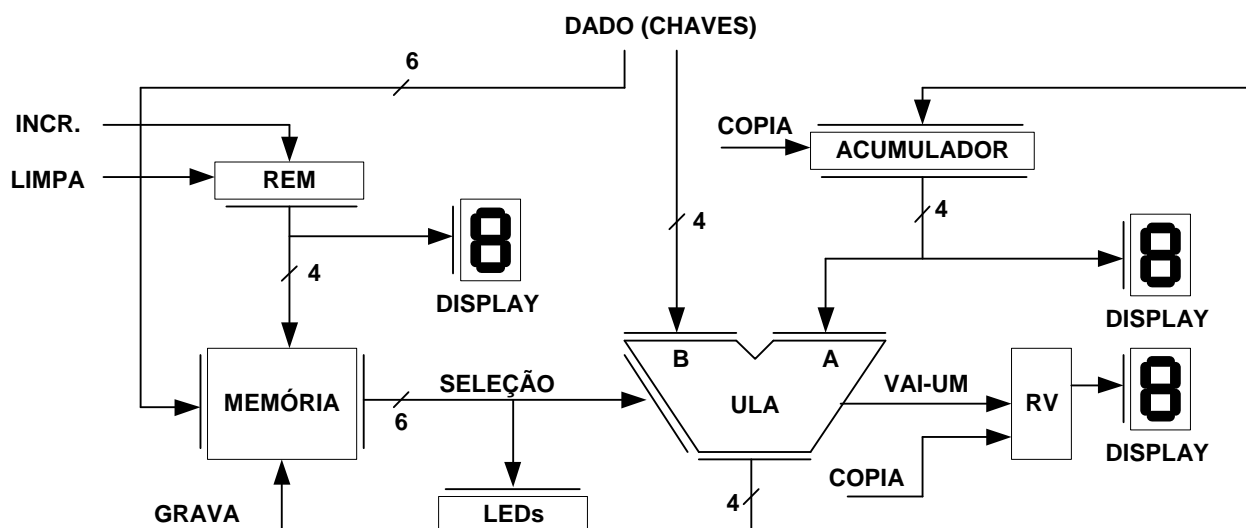
Utilizando a ULA 74181 e a memória 74189, projetar um circuito cujo fluxo de dados de 4 bits é esquematizado na figura 2.1 abaixo. Este fluxo de dados deve executar operações lógicas e aritméticas sobre os dados fornecidos pelas chaves, obedecendo a uma sequência dada pelas "instruções" contidas na memória. Cada "instrução" tem 6 bits e estes são utilizados para determinar a operação a ser executada pela ULA, pelos sinais  $S_0, S_1, S_2, S_3, M$  e  $C_n$ .

O fluxo de dados a ser projetado deve ter dois modos de operação (chave C7), descritos a seguir.

- a) **Modo "Programação"** ( $C7=0$ ): este modo de operação permite a carga de um "programa" na memória, que comandará a sequência de operações que se quer executar. Deve-se utilizar:
- 6 chaves ( $C0$  a  $C5$ ) para a entrada do conteúdo da memória;
  - 1 botão ( $B1$ ) para gerar o sinal GRAVA, que força o armazenamento do conteúdo das chaves na memória;
  - 1 botão ( $B2$ ) para gerar o sinal INCR, que tem como função incrementar o *Registrador de Endereços da Memória* (REM).
- b) **Modo "Execução"** ( $C7=1$ ): neste modo de operação, o fluxo de dados deve executar a sequência de operações dada pelo conteúdo da memória. Deve-se utilizar:
- 4 chaves ( $C0$  a  $C3$ ) para a entrada de dados;
  - 1 botão ( $B1$ ) para gerar o sinal LIMPA, que tem como função zerar o REM;
  - 1 botão ( $B2$ ) para comandar a execução de uma "instrução" gerando os sinais COPIA e INCR.

**Lembre-se que o planejamento deve incluir simulações com o Altera Quartus II.**

O modo de operação do circuito é selecionado pela chave C7. A unidade de controle deve ser projetada para gerar os sinais INCR, LIMPA, GRAVA e COPIA de acordo com o modo de operação selecionado.



**Figura 2.1 - Fluxo de Dados a ser Projetado.**

**OBS:** o grupo pode trazer etiquetas de papel auto-adesivas no dia da experiência para identificar os vários sinais da montagem. É proibido escrever diretamente no painel de montagem.

Assim, para se executar um "programa", deve-se seguir o seguinte procedimento:

- Colocar o circuito em modo "Execução" e limpar o REM;
- Mudar o modo de operação para o modo "Programação" e armazenar o "programa" na memória;
- Colocar o circuito em modo "Execução" e limpar o REM;
- Executar, passo a passo, as "instruções", colocando nas chaves o dado a ser manipulado, apertando o botão de execução (B2) e observando os resultados nos *leds* e *displays*.

## 2.2. Familiarização com os componentes

Antes de efetuar a montagem do projeto do circuito projetado, deverão ser executados certos procedimentos com o objetivo de se familiarizar com a ULA 74181 e a memória 74189.

### 2.2.1 Familiarização com a memória 74189

Este item tem com objetivo a familiarização com a operação do circuito de memória 74189.

- Monte um contador 74161 de 4 bits, ligando o botão B1 na entrada de *clock* e as quatro saídas em um display (D3). Ligue os demais sinais de entrada de acordo e teste o funcionamento desta parte.
- Efetuar em seguida a ligação dos seguintes sinais:
  - Ligar os 4 bits da saída do contador acima nas 4 entradas de endereço do 74189.
  - Ligar as chaves C0 a C3 nas entradas de dados do 74189 e as saídas D<sub>0</sub> a D<sub>3</sub> em *leds* (L8 a L11).
  - Ligar os sinais /CE e /WE nas chaves C6 e C7, respectivamente.
- Acionando o botão B1 e as chaves C6 e C7, execute operações de gravação de dados na memória e posterior leitura dos dados.  
 IMPORTANTE: Lembre-se de seguir a sequência apresentada nas formas de onda da figura 1.5 para o ciclo de gravação de dados na memória.
- Anote e comente como é o funcionamento da memória.

#### PERGUNTAS:

- Qual é a relação entre os dados de entrada da memória e os dados de saída?
- Se desejarmos que a saída de uma certa posição da memória seja "0110", como devemos proceder?
- O que acontece em um ciclo de escrita da memória se, enquanto o sinal /WE é mantido em 0, o valor das entradas de endereço é incrementado?

### 2.2.2 Familiarização com a ULA 74181

Este item tem como objetivo a familiarização com a operação da ULA 74181.

- No circuito do item anterior, acrescente uma segunda pastilha de memória, interligando os sinais de controle de forma especificada abaixo.
  - Ligar os 4 bits da saída do contador nas 4 entradas de endereço das pastilhas de memória.
  - Ligar as entradas de dados nas chaves C0 a C3 (pastilha 1) e C4 e C5 (pastilha 2).
  - Ligar os sinais /CE e /WE nas chaves C6 e C7, respectivamente.
  - Ligar os 4 bits das saídas da pastilha 1 de memória (que devem permanecer ligados nos *leds* L8 a L11) nos sinais de seleção S<sub>0</sub> a S<sub>3</sub>, respectivamente, da ULA 74181.
  - Ligar os 2 bits menos significativos da pastilha 2 nas entradas M e C<sub>n</sub>, respectivamente, da ULA 74181 e também nos *leds* L12 e L13.
  - Ajustar valores fixos com *jumpers* nas entradas A e B da ULA 74181.
  - Ligar a saída de dados da ULA no display D0 e o bit de "vai-um" no display D1.
- Programar nas primeiras posições de memória, o código de algumas operações selecionadas (veja Tabela II). Sugestão: use as operações da pergunta 3 abaixo.
- Acionando a memória para realizar operações de leitura, verifique a execução das operações programadas na memória, observando as saídas da ULA.

#### PERGUNTAS:

- Quais os valores dos sinais de entrada de controle da ULA para realizar as seguintes operações: A+B, A-B, 2×A, A and B, A or B, not B, A xor B?
- Quando há dois valores distintos para as entradas de controle da ULA que realizam a mesma operação, qual delas é a mais adequada? Por que?
- O que acontece com o valor da saída da ULA quando o sinal C<sub>n</sub> muda de 0 para 1?

### 2.3. Montagem do Circuito com a ULA e a RAM

Neste item, o circuito com a ULA deverá ser montado e testado de acordo com o planejamento previamente elaborado pelo grupo. O diagrama lógico deve apresentar todos os detalhes de montagem. Recomenda-se que o planejamento inclua simulações com o Quartus II.

h) Montar o circuito projetado anteriormente.

DICA: Para facilitar a depuração, monte inicialmente a memória e o REM; certifique-se do funcionamento correto destes e depois monte o resto do circuito.

i) Verifique o funcionamento de cada bloco do circuito, apresentando os testes realizados para verificar o funcionamento do subcircuito montado.

PERGUNTA:

7. Como garantir com um mínimo de operações o correto funcionamento do circuito? Justifique.

8. Como as montagens dos itens da seção 2.2 podem ser reaproveitados na montagem do circuito projetado?

### 2.4. Teste do Circuito com "Programas"

j) Verifique o funcionamento do circuito para várias sequências de operações, variando também os dados de entrada.

k) Escreva os "programas" para calcular as seguintes expressões:

i.  $X = (2 \times A) - B - C + D + 1$

ii.  $Y = (A \text{ OR NOT}(B) \text{ AND } C) \text{ XOR } D$

l) Verifique os resultados obtidos executando os "programas", para vários valores de A, B, C e D.

*Sugestão: para escrever os "programas", faça uso da Tabela II, a despeito da memória ter saídas ativas em zero.*

PERGUNTA:

9. Este projeto se assemelha com o núcleo de um processador de propósito geral, como os microprocessadores e microcontroladores. O registrador REM é equivalente ao registrador PC ("program counter" ou "instruction pointer"). Qual é a função do PC em um processador?

10. Qual é a função do ACUMULADOR em um processador?

### 2.5 Modificações no Circuito

O entendimento do funcionamento completo do circuito projetado é muito importante. O professor **poderá** solicitar uma pequena modificação no circuito projetado pelo grupo. Esta modificação deverá ser realizada e documentada de acordo com as normas do laboratório (documento "Documentação de aula" disponível na página Internet do curso).

PERGUNTA:

11. Seja a implementação da inclusão de uma instrução de salto incondicional (JUMP) para uma determinada posição de memória. A instrução JUMP em um processador faz com que a próxima instrução a ser executada seja especificada pelo dado de entrada que especifica o endereço de memória. Por exemplo, se o dado de entrada contiver o valor "0111", o circuito deve mudar a sequência de execução e iniciar a execução pela instrução presente no endereço "0111" da memória. Mostre um diagrama contendo as modificações necessárias para incluir esta instrução ao projeto.

## 3. BIBLIOGRAFIA

- Fairchild Semiconductor. **TTL Data Book**. Mountain View, California, 1978.
- FREGNI, E. e SARAIVA, A. M. **Engenharia do Projeto Lógico Digital: Conceitos e Prática**. Editora Edgard Blücher Ltda, 1995.
- TANENBAUM, A. S. **Organização estruturada de computadores**. 5ª edição, Prentice-Hall, 2009.
- TOCCI, R. J.; WIDMER, N.S.; MOSS, G.L. **Sistemas Digitais: Princípios e Aplicações**. Prentice-Hall, 10ª ed., 2007.
- WAKERLY, John F. **Digital Design Principles & Practices**. 4<sup>th</sup> edition, Prentice Hall, 2006.



#### **4. MATERIAL DISPONÍVEL**

- Circuitos Integrados TTL:  
7400, 7404, 7408, 7410, 7432, 7450, 7474, 7486, 74157, 74161, 74175, 74181, 74189, 74193.

ATENÇÃO: procure usar somente os componentes acima para o projeto. Recomenda-se não usar quaisquer outros componentes.

#### **5. EQUIPAMENTOS NECESSÁRIOS**

- 1 painel de montagens experimentais.
- 1 fonte de alimentação fixa,  $5V \pm 5\%$ , 4A.
- 1 osciloscópio digital.
- 1 multímetro digital.
- 1 gerador de pulsos.
- 1 computador compatível com IBM-PC com software Altera Quartus II.