

Relatório Lab. AC II – ULA 4 bits + Arduino

Neste exercício, assim como o anterior, você deverá criar 2 programas. Um no hardware externo (Arduino) e outro no PC, que será a interface com o usuário. A idéia é ler um programa escrito pelo usuário, transformá-lo em mnemônicos gerando outro programa e finalmente passá-lo ao Hardware externo através da porta serial e realizar algum processamento nesse Hardware. O resultado será observado nos 4 Leds conectados no Hardware externo.

O Hardware externo

Você deverá projetar uma ULA com 4 bits para um dado A, 4 bits para um dado B e 4 bits para a instrução desejada. O funcionamento é similar à ULA anteriormente estudada.

Uma arquitetura do sistema proposto pode ser vista na Figura 1.

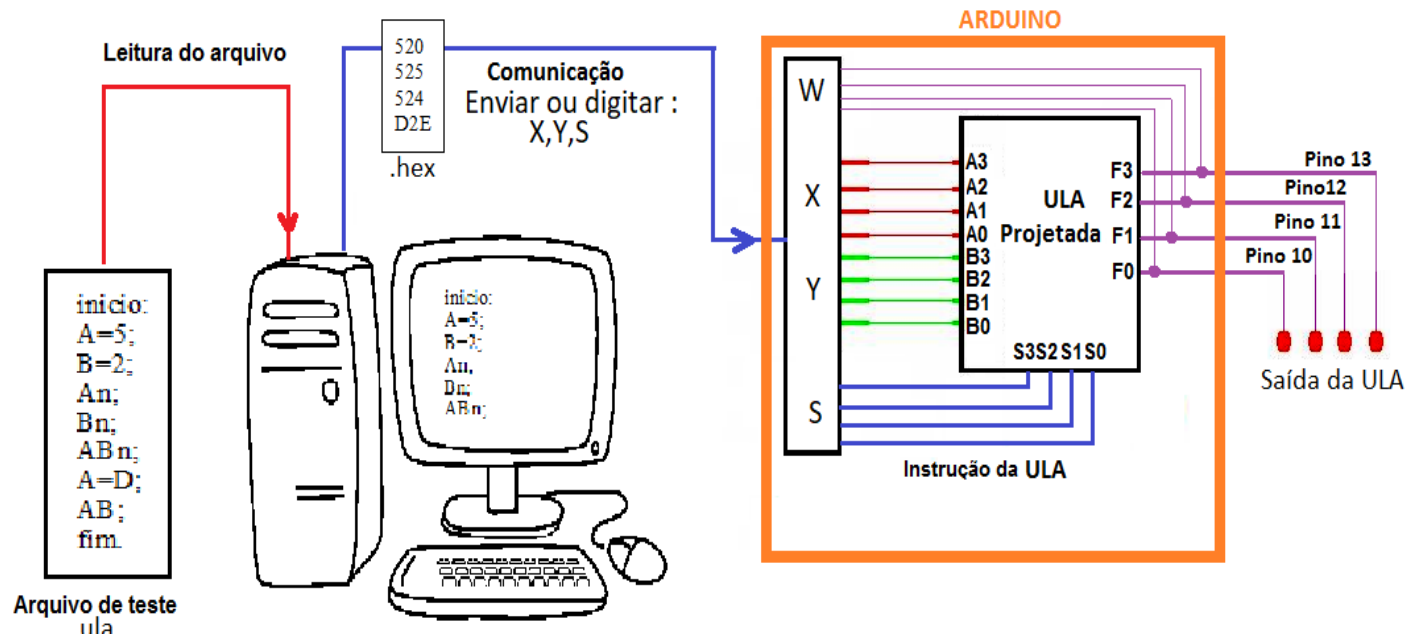


Figura 1: Arquitetura do sistema proposto

Você deverá elaborar um programa no Arduino que utilize a entrada serial para receber as entradas necessárias ao funcionamento da ULA (dados e instruções) e as saídas deverão ser 4 Leds ligados aos pinos 13, 12, 11 e 10 (o bit mais significativo no pino 13 e o menos significativo no pino 10).

A figura 2 a seguir mostra o conjunto de instruções da ULA e que você deverá inserir no Arduino.

| Função | Mnemônico | Código Hexa |
|-----------------|-----------|-------------|
| A' | An | 0 |
| (A+B)' | nAoB | 1 |
| A'B | AnB | 2 |
| 0 Lógico | zeroL | 3 |
| (AB)' | nAeB | 4 |
| B' | Bn | 5 |
| $A \oplus B$ | AxB | 6 |
| AB' | ABn | 7 |
| A' + B | AnoB | 8 |
| $(A \oplus B)'$ | nAxB | 9 |
| B | B | A |
| AB | AB | B |
| 1 lógico | umL | C |
| A+B' | AoBn | D |
| A+B | AoB | E |
| A | A | F |

Figura 2: Instruções e Mnemônicos
(ativos em 1- high)

O programa no Arduino

Seu programa no arduino deverá ser capaz de receber 3 dados da seguinte forma:

Um primeiro valor representando a entrada X (X0, X1, X2 e X3).

Um segundo valor representando a entrada Y (Y0, Y1, Y2 e Y3).

Um terceiro valor representando a instrução desejada S (S0, S1, S2 e S3).

Assim, se fornecermos pela comunicação serial na IDE do Arduino os seguintes 3 valores:

1 2 4, estaremos passando para a ULA as seguintes informações:

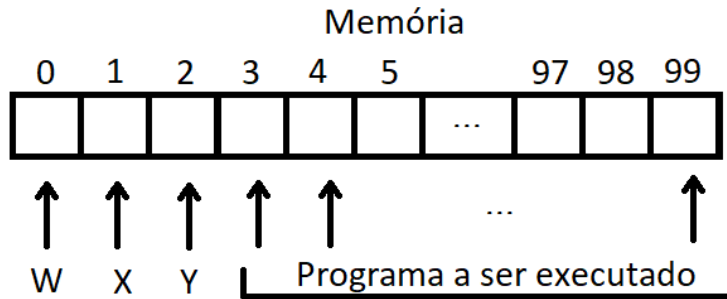
Valor de **X=1**, valor de **Y=2** e a instrução desejada=4 ou **S=4**. A ULA projetada no arduino deverá então realizar, conforme o conjunto de instruções da ULA (de acordo com a Fig. 2), a instrução (nAeB), ou seja (AB)' que sobre as variáveis X e Y ficaria (XY)'.

Observe que as operações sempre serão realizadas sobre as variáveis X e Y.

Observe que, para não haver confusão nos valores, deveremos usar os números em Hexadecimal, assim, se passarmos ao Arduino os seguintes dados A A A, o significado será:

Valor de X = 10, valor de Y=10 e a instrução desejada ou S=10. A ULA projetada no arduino deverá então realizar, conforme o conjunto de instruções da ULA (e de acordo com a Fig. 2) a instrução B, atenção que a instrução B apenas coloca o valor da entrada Y na saída (**não confunda a instrução B com a entrada tendo o valor de B**).

Deverá existir internamente no Arduino um vetor que será a memória da Unidade ou seja, deverá conter nos três primeiros campos os valores do resultado após a execução de uma instrução (W) e as duas variáveis (X e Y). Vamos considerar que iremos utilizar um espaço relativo a 100 posições (esta será a nossa área de memória) onde as três primeiras posições serão as variáveis e as 97 restantes o programa a ser executado.



Se, durante o programa a ser executado, houver uma alteração no valor das variáveis X ou Y, este valor deverá ser alterado na memória (nas respectivas posições do vetor). A alteração se dá através da atribuição de valores no programa fonte original.

Deveremos ainda acompanhar a evolução do programa observando 4 leds indicadores dos resultados ou seja, a saída da ULA deverá estar presente nos seguintes pinos:

- Pino 13 = F3
- Pino 12 = F2
- Pino 11 = F1
- Pino 10 = F0

Pergunta: Qual seria o significado de passarmos para o Arduino os seguintes valores “C 6 B”, como ficariam os LEDs ligados na saída e a memória antes e após a execução da instrução?

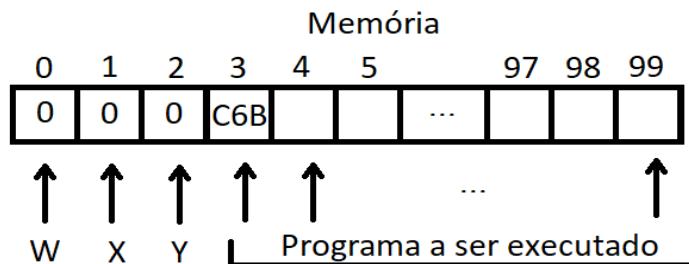
Resp:

- Entrada dos valores para X = 1100 ou C;
- Entrada para os valores de Y = 0110 ou 6;
- Entrada para os valores de S = 1011 ou B;

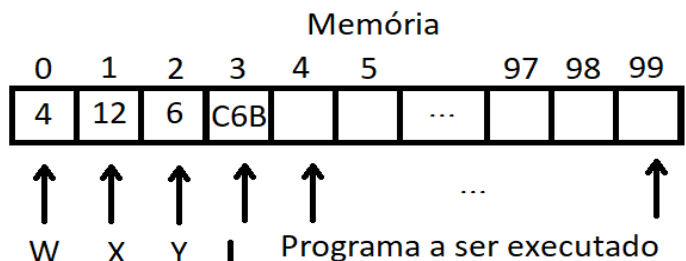
Como S=1011 indica que queremos a instrução AB (and das entradas), a saída F seria 0100 (4 em decimal) ou o led do pino 12 ligado.

Além disso a memória deverá conter os seguintes valores antes e após a execução do programa

Antes da execução da instrução:



Depois da execução da instrução:



Atenção (detalhe de projeto 1): Fica a critério do grupo a utilização do vetor para armazenar os valores, quer seja números na base decimal, binária ou mesmo String.

Funcionamento:

1)

Ao iniciarmos o Arduino, inicialmente deveremos fazer a carga do programa no vetor que representa a memória.

Atenção (detalhe de projeto 2): O grupo deverá propor uma forma de entrar com os dados que estarão no formato descrito pelo arquivo testeula.hex e que será descrito posteriormente nesta especificação. Eventualmente esta carga poderá ser realizada através de comunicação USB/serial (caso um Arduino real esteja presente) ou digitada, caso tenhamos uma versão virtual do Arduino.

2)

Ao iniciarmos a execução do programa, o Arduino deverá ler as instruções da memória a partir do início do programa a ser executado (posição 3 do vetor), decodificar a instrução, executar a instrução e escrever os valores das variáveis X e Y na memória e o resultado em W (posição 0 do vetor) assim como mostrar o valor do resultado nos leds. Posteriormente passar para a próxima instrução e assim sucessivamente. Vamos considerar inicialmente um intervalo de 2 segundos entre cada instrução, para podermos acompanhar as respostas nos LEDs.

Ao final da execução de cada instrução deveremos ter um DUMP da memória, ou seja, o Arduino mostrará todos os valores contidos na memória para acompanharmos a execução. Esta opção poderá estar continuamente ativa ou ser ativada caso o usuário deseje. Procure mostrar neste DUMP apenas as posições onde existem valores na memória e não toda ela.

O Software no PC

O software no PC poderá ser escrito em C, C++, C# , Java ou Python.

Você deverá criar um programa que transforme um texto lido de um arquivo nas instruções a serem executadas e permita a sua execução linha a linha através do console. Para isso, o programa deverá inicialmente ler um arquivo contendo um texto original com os mnemônicos (instruções a serem executadas) e gerar um segundo texto, onde cada linha seja transformada nos valores que serão disponibilizados para a porta USB/serial (ou digitados) no Arduino. Esse segundo texto deverá ser um arquivo gravado com os respectivos valores a serem enviados para a porta USB/serial (ou digitados) porém no formato hexadecimal.

Você deverá utilizar o conjunto de instruções que a ULA possui ilustrado na Figura 2. A Figura 3 ilustra um pequeno exemplo de código a ser transformado. A Figura 4 ilustra o programa a ser gerado. Os nomes dos arquivos indicados nas Figuras 3 e 4 correspondem aos nomes que você deverá utilizar no programa para leitura e escrita.

```
inicio:  
X=5;  
Y=2;  
An;  
Bn;  
nAeB;  
X=D;  
AoB;  
fim.
```

Figura 3: Exemplo do programa de teste
"testeula.ula"

```
520  
525  
524  
D2E
```

Figura 4: Programa gerado
"testeula.hex"

Como se vê, o programa a ser enviado(ou digitado) deverá ser a partir do arquivo contendo o programa gerado (Programa no formato .hex, Figura 4) e não o programa fonte original (programa no formato .ula, Figura 3).

O ciclo de execução da máquina pode ser entendido através da Figura 5 a seguir.

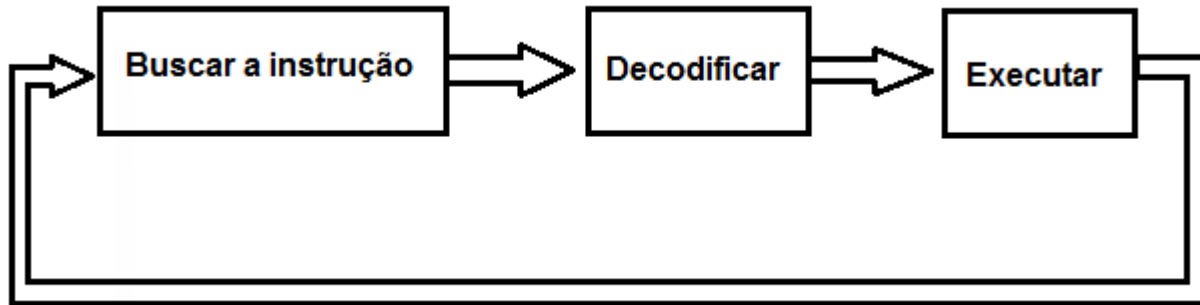


Figura 5: Ciclo de execução de uma instrução

2. O que apresentar ao final do projeto

- 1) Um programa no Arduino que simule uma ULA e receba os valores dos dados e instruções através da porta serial ou sejam digitados.
- 2) Um programa de acesso em C/C++/Java/Python (com muitos comentários!) que:
 - a) **leia um programa fonte (com os dados e os mnemônicos), você deverá criar um programa fonte de teste. Durante a aula um outro programa será utilizado para verificação do trabalho.**
 - b) **gere um arquivo hexa correspondente aos dados e instruções e**
 - c) **envie dados e instruções** através da porta Serial para a ULA (Caso um Arduino real esteja presente)

O programa de teste possuirá o nome "testeula.ula" e você só terá acesso a ele no momento do teste. O formato será o mesmo descrito na Figura 3 e os mnemônicos da Figura 2. O programa gerado deverá possuir o nome "testeula.hex". Para o seu teste crie seu próprio programa fonte, lembre-se de procurar testar todas as instruções possíveis.

Cada grupo fará a apresentação dos programas, (será a nota desse relatório, não haverá entrega pela internet, apenas a apresentação.)

Cada grupo deverá estar com os programas disponíveis e fazer a apresentação do trabalho virtualmente.

Eu irei avaliar/ testar individualmente os programas de cada grupo durante a apresentação, alunos que participaram do trabalho mas ausentes na apresentação, não terão nota.

Grupos que não estiverem com os programas não terão nota no relatório, o mesmo acontecendo com trabalhos copiados.

Comece já, você nunca terá tanto tempo!