Universidade Federal do Amazonas
Instituto de Computação
ICC060 – Sistemas Lógicos (SL) – 2018/2
Prof. Leandro Galvão

Atividade Prática 06 – Circuitos Multiplexadores

OBJETIVOS:

- Empregar um MUX 8×4 como seletor de dados de 4 bits de largura.
- Aplicar o princípio de cascateamento de multiplexadores.

MATERIAIS:

- 01 placa Arduino Uno
- 04 protoboards de 830 furos
- 20 resistores de 330 Ω (laranja – laranja – marrom)
- 20 LEDs difusos
- 03 CIs 74157
- Fios jumper macho-macho

ATENÇÃO:

- Não troque os terminais de alimentação dos CIs (**Vcc** e **GND**).
- Conecte o terminal do lado chanfrado do LED ao **GND** da fonte.
- As cores das trilhas da protoboard são apenas um *referencial*. Você tem que fornecer alimentação para injetar algum sinal nelas.

Experimento 1: O multiplexador 8×4 (74157)

Considere duas palavras de dados, com 4 bits de largura: *RA* e *RB*. O objetivo deste Experimento é utilizar um MUX 8×4 (CI 74157) para selecionar qual palavra (*RA* ou *RB*) será entregue na saída *Y*.

Este cenário é uma simplificação do que acontece em um processador real, que opera sobre palavras de 64 bits de largura. No processador, um MUX é utilizado, por exemplo, para selecionar qual das seguintes instruções será executada:

- `x = a + b`: adição de duas variáveis, `a` e `b`, ambas armazenadas na memória de dados.
- `x = a + 42`: adição de uma variável `a`, armazenada na memória de *dados*, com uma constante de valor `42`, armazenada na memória de *instruções*. No jargão da Arquitetura de Computadores, constantes são referidas como *imediato*.

Como mostra a Figura 1, o MUX é responsável por selecionar qual dos dois tipos de adição será realizado (com variável ou com imediato): uma das entradas do MUX vem da memória de dados (variável `b`) e a outra entrada vem da memória de instruções (imediato 42). O seletor do MUX é comandado pela Unidade de Controle do processador, que verifica qual das duas instruções será realizada. A partir daí, Unidade de Controle determina qual será o valor do seletor do MUX (0 ou 1) e do seletor da Unidade Lógica e Aritmética (operação de adição).

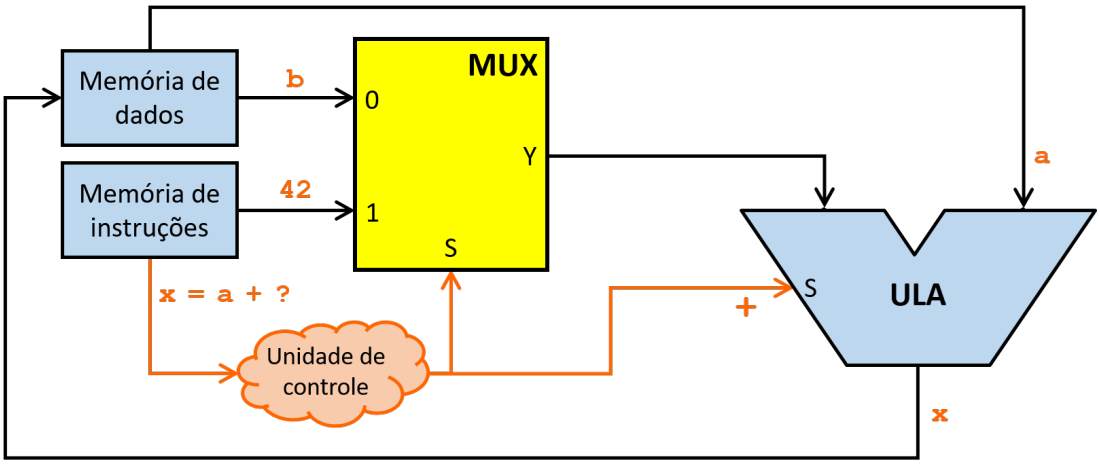


Figura 1: Exemplo simplificado de operação conjunta entre MUX e ULA, para determinar qual entre dois tipos de adição será realizado pelo processador.

TAREFAS

1. Monte o circuito ilustrado na Figura 3 segundo a disposição da Figura 4:

   (a) Siga rigorosamente a disposição sugerida pela Figura 4. A partir desta Atividade Prática, você começará a montar o Processador Básico do Trabalho Final. **Se você dispor os componentes de forma diferente, terá que refazer todas as conexões quando for fazer o Trabalho Final.**

   (b) Por enquanto, **não** conecte as entradas RA_i e RB_i, nem o seletor S do MUX ao Arduino.

   (c) Carregue o Código 1. Ele gera dois sinais de 4 bits de largura:

      i. Nos pinos 10–13 do Arduino, são injetados os vetores E_0–E_3 do Código 1 (linhas 5 a 8). Portanto, eles produzem uma contagem em ordem *crescente*, conforme se vê na Figura 2, observando-se coluna por coluna.

      ii. Já nos pinos 6–9 do Arduino, os elementos dos vetores E_0–E_3 são injetados em ordem inversa, do último para o primeiro. Portanto, eles produzem uma contagem em ordem *decrescente*, como se as colunas da Figura 2 fossem lidas da direita para a esquerda.

   (d) Por meio dos LEDs azul (RA) e verde (RB), verifique se o Arduino realiza corretamente as duas contagens: crescente de 0 a 15 nos pinos 10–13, e decrescente de 15 a 0 nos pinos 6–9.

2. Agora, consulte a pinagem e a tabela verdade do CI 74157 no Apêndice:

   (a) No MUX 74157, a entrada E (*enable*), no pino 15, serve para habilitar ($E = 0$) ou desabilitar ($E = 1$) o CI, conforme tabela verdade. Por isso, conecte-a no GND, como mostra a Figura 4.

   (b) Use fios jumper macho-macho para conectar as saídas 10–13 do Arduino nas entradas RA_0 a RA_3 do MUX 74157, e as saídas 6–9 do Arduino nas entradas RB_0 a RB_3 do MUX. Para isso, bifurque o sinal disponível no anodo (terminal positivo) de cada LED da protoboard P1 em direção às entradas de dados do MUX, localizado na protoboard P3. Observe a *ordem de significância dos bits*: RA_0 e RB_0 são os bits *menos* significativos.

   (c) Use um fio jumper para ajustar o valor do seletor S do MUX 74157:

   - Se $S = 0$, então $Y = RA$.
   - Se $S = 1$, então $Y = RB$.

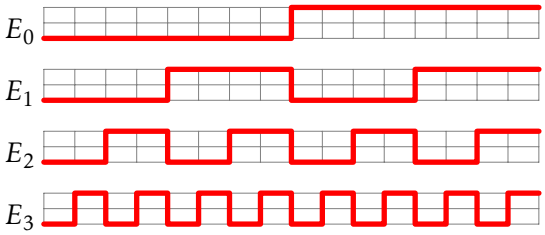


Figura 2: Quatro ondas quadradas de períodos distintos.

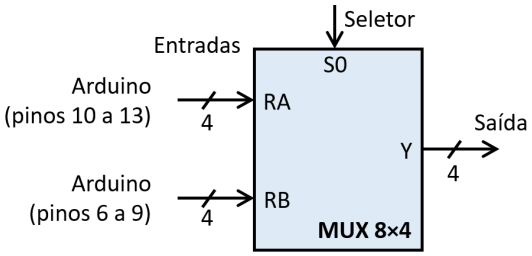


Figura 3: Diagrama lógico do MUX 8×4.

Código 1: Gera duas sequências de contagem.

3. **Mostre o seu circuito ao instrutor.**

```
#define TEMP 1000    /* Semi-periodo (em ms) da onda E0 (LSB) */
#define SLOTS 16     /* No. de slots de tempo em 4 ondas quadradas */

/* Vetores com os sinais E0, E1, E2, E3, em frequencias diferentes */
byte e3[] = {LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,HIGH,HIGH,HIGH,HIGH,HIGH,HIGH,HIGH,
    HIGH};
byte e2[] = {LOW,LOW,LOW,LOW,HIGH,HIGH,HIGH,HIGH,LOW,LOW,LOW,LOW,HIGH,HIGH,HIGH,
    HIGH};
byte e1[] = {LOW,LOW,HIGH,HIGH,LOW,LOW,HIGH,HIGH,LOW,LOW,HIGH,HIGH,LOW,LOW,HIGH,
    HIGH};
byte e0[] = {LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,
    HIGH};

/* a funcao 'setup' eh executada apenas uma vez */
void setup() {
  /* ajusta todos os digitais como saida */
  for (byte i = 0; i <= 13; i++)
    pinMode(i, OUTPUT);
}

/* a funcao 'loop' eh executada indefinidamente */
void loop() {
  byte j;   /* variavel auxiliar */

  for(byte i = 0; i < SLOTS; i++) {
    /* RA (contagem progressiva 0--15) */
    digitalWrite(10, e0[i]);
    digitalWrite(11, e1[i]);
    digitalWrite(12, e2[i]);
    digitalWrite(13, e3[i]);

    /* RB (contagem regressiva 15--0) */
    j = SLOTS - i - 1;
    digitalWrite(6, e0[j]);
    digitalWrite(7, e1[j]);
```

```
32        digitalWrite(8, e2[j]);
33        digitalWrite(9, e3[j]);
34
35        delay(TEMP);
36    }
37 }
```

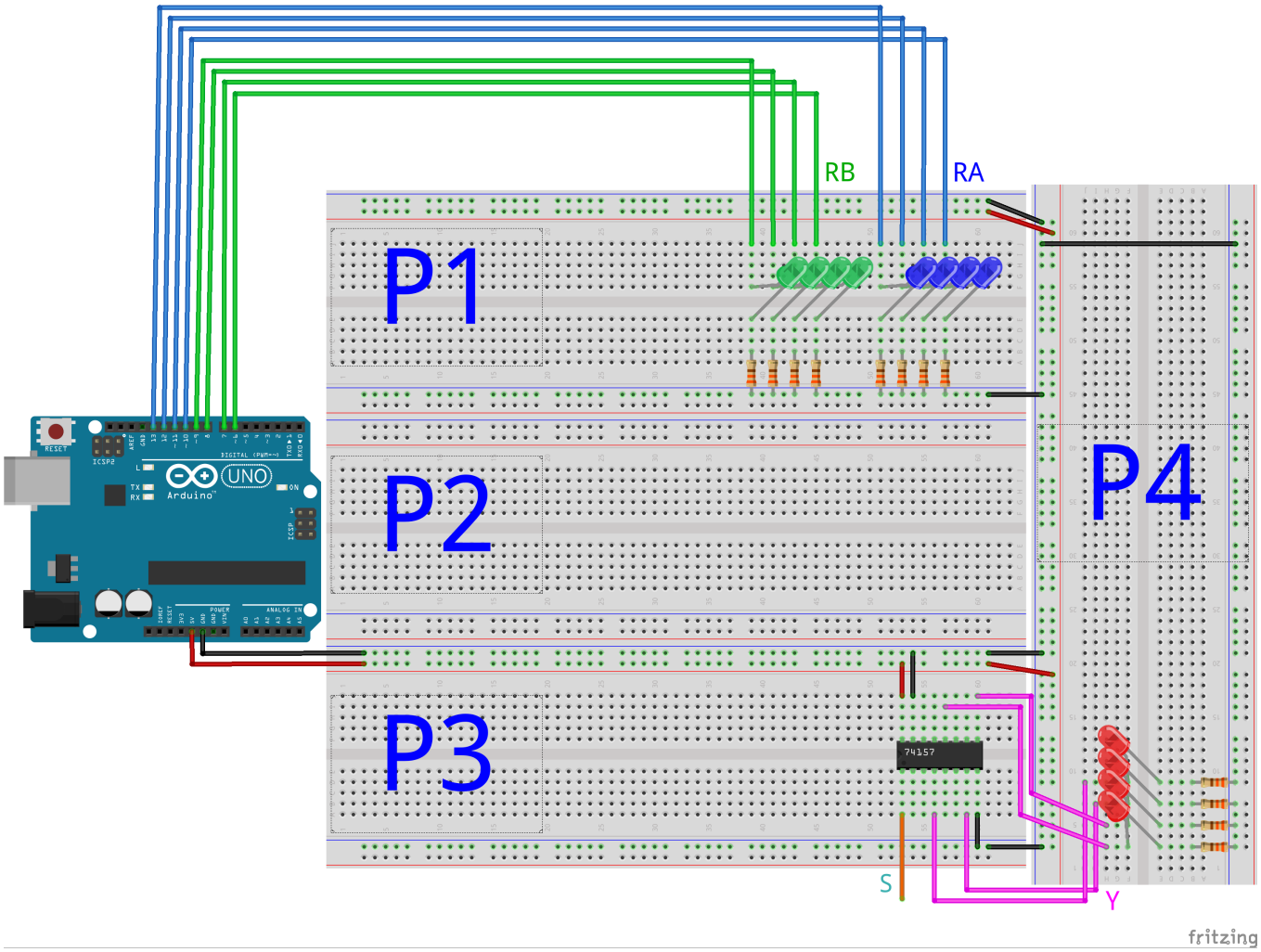


Figura 4: Disposição dos componentes no Experimento 1.

Experimento 2: Cascateamento do MUX 8 × 4 (74157)

Durante a execução de um programa, o processador mantém uma cópia das variáveis mais utilizadas recentemente em uma memória interna chamada *banco de registradores*. Cada registrador do banco armazena uma variável. Voltando ao exemplo da instrução `x = a + 42`, o processador deverá *selecionar* o registrador que contém o valor da variável `a`, a fim de executar essa instrução.

Neste Experimento, vamos considerar um exemplo bem simples de processador, que contém apenas 04 registradores. Cada registrador armazena palavras dedados com quatro bits de largura. O objetivo do experimento é selecionar qual dos quatro registradores terá seus dados encaminhados à saída Y do MUX. Portanto, você terá que cascatear três MUX 8×4 para implementar um MUX 16×4.

O Arduino será utilizado para emular o valor armazenado nos quatro registradores, designados por $R0$–$R3$:

- O registrador `R3` faz uma contagem progressiva de 0–15.
- O registrador `R2` faz uma contagem regressiva de 15–0.
- O registrador `R1` faz uma contagem progressiva de 0–14, apenas dos números pares, com o dobro da frequência de $R3$ e $R2$.
- O registrador `R0` faz uma contagem regressiva de 15–8, com o dobro da frequência de $R3$ e $R2$.

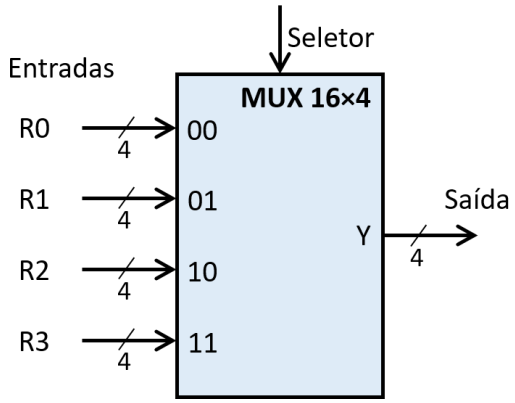


Figura 5: Diagrama lógico do MUX 16×4.

TAREFAS

1. Monte o circuito correspondente ao diagrama lógico da Figura 5, tomando o circuito da Figura 6 como *ponto de partida*:

   (a) Primeiro, teste se os registradores R0–R3 estão fazendo a contagem conforme descrito acima.

   (b) Depois, conecte os sinais R0–R3 do Arduino às entradas dos MUX. Para isso, bifurque o sinal disponível no anodo (terminal positivo) de cada LED da protoboard P1 em direção às entradas de dados dos MUX, na protoboard P3.

   (c) Interligue os CI 74157 entre si, implementando o cascateamento.

2. Desconecte o fio jumper da saída 0 do Arduino. Carregue o Código **??** no Arduino. Reconecte o fio jumper à saída 0 do Arduino.

3. **Mostre o seu circuito ao instrutor.**

4. Desenhe, na **Folha de Respostas**, um diagrama lógico descrevendo como três multiplexadores 8×4 devem ser cascateados para implementar um MUX 16×4. Explicite os nomes das entradas e saídas, e indique a ordem de *significância* das entradas e dos seletores. Ou seja, você deve explicitar as ligações internas ao retângulo da Figura 5.

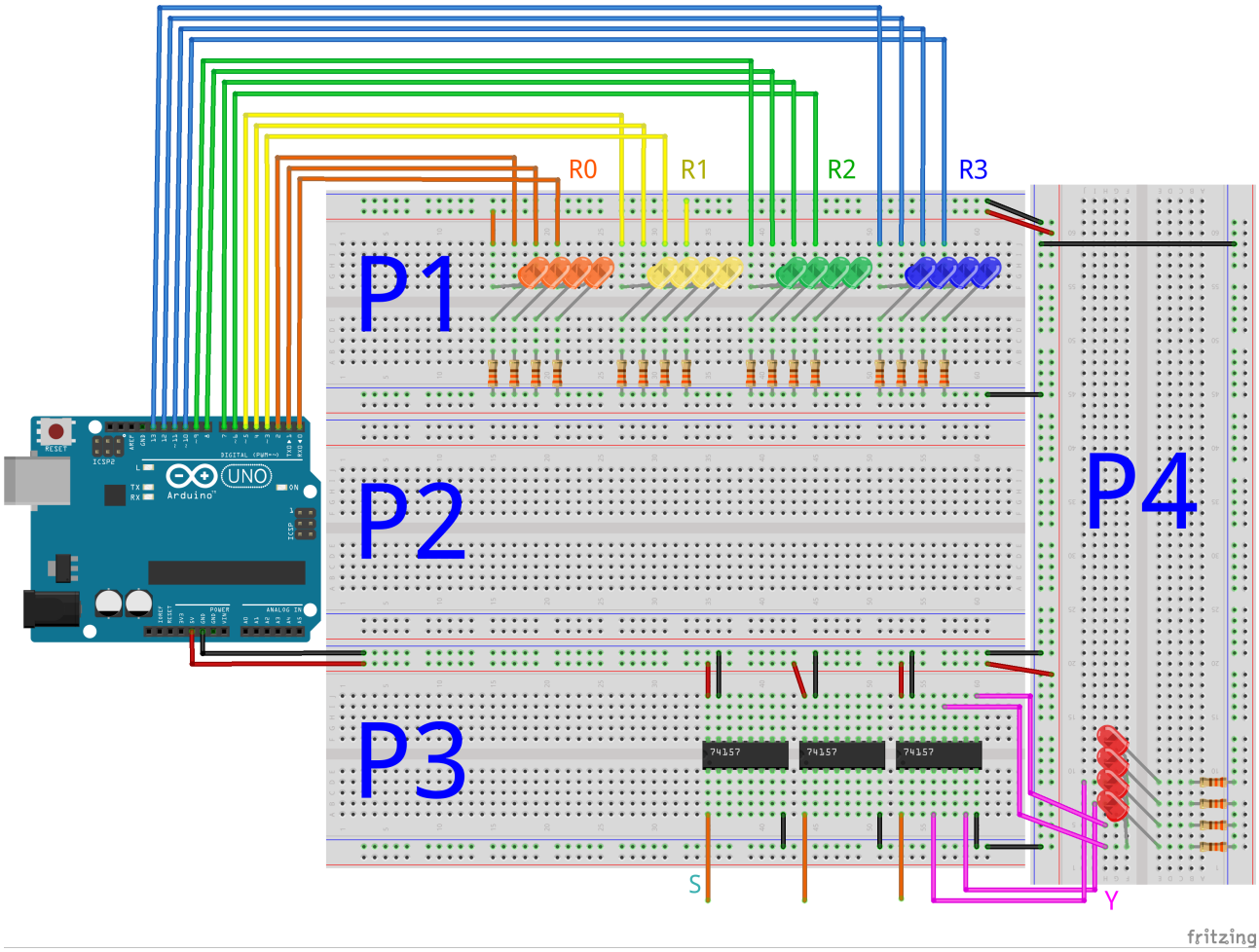


Figura 6: Disposição dos componentes para *iniciar a montagem* de um MUX 16×4. Não se esqueça de ligar os sinais R0–R3 às entradas do MUX, e de interligar os MUX entre si.

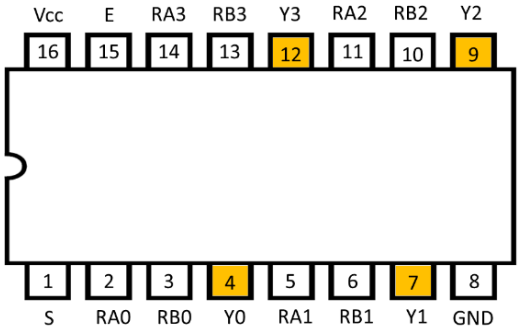
Código 2: Gera quatro sequências de contagem.

```
#define TEMP 1000    /* Semi-periodo (em ms) da onda E0 (LSB) */
#define SLOTS 16     /* No. de slots de tempo em 4 ondas quadradas */

/* Vetores com os sinais E0, E1, E2, E3, em frequencias diferentes */
byte e3[] = {LOW,LOW,LOW,LOW,LOW,LOW,LOW,LOW,HIGH,HIGH,HIGH,HIGH,HIGH,HIGH,HIGH,
    HIGH};
byte e2[] = {LOW,LOW,LOW,LOW,HIGH,HIGH,HIGH,HIGH,LOW,LOW,LOW,LOW,HIGH,HIGH,HIGH,
    HIGH};
byte e1[] = {LOW,LOW,HIGH,HIGH,LOW,LOW,HIGH,HIGH,LOW,LOW,HIGH,HIGH,LOW,LOW,HIGH,
    HIGH};
byte e0[] = {LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,HIGH,LOW,
    HIGH};

```

```
/* a funcao 'setup' eh executada apenas uma vez */
void setup() {
  /* ajusta todos os digitais como saida */
  for (byte i = 0; i <= 13; i++)
    pinMode(i, OUTPUT);
}

/* a funcao 'loop' eh executada indefinidamente */
void loop() {
  byte j;   /* variavel auxiliar */

  for(byte i = 0; i < SLOTS; i++) {
    /* R3 (contagem progressiva 0--15) */
    digitalWrite(10, e0[i]);
    digitalWrite(11, e1[i]);
    digitalWrite(12, e2[i]);
    digitalWrite(13, e3[i]);

    /* R2 (contagem regressiva 15--0) */
    j = SLOTS - i - 1;
    digitalWrite(6, e0[j]);
    digitalWrite(7, e1[j]);
    digitalWrite(8, e2[j]);
    digitalWrite(9, e3[j]);

    /* R1 (contagem progressiva 0--14, pares) */
    digitalWrite(3, e1[i]);
    digitalWrite(4, e2[i]);
    digitalWrite(5, e3[i]);

    /* R0 (contagem regressiva 15--8) */
    digitalWrite(0, e0[j]);
    digitalWrite(1, e1[j]);
    digitalWrite(2, e2[j]);

    delay(TEMP);
  }
}
```

Apêndice – Pinagem dos CIs utilizados

74157 (MUX 8 × 4)



ENABLE	SELECT INPUT	DATA INPUTS		OUTPUT
$\bar{E}$	S	RA	RB	Y
H	X	X	X	L
L	L	L	X	L
L	L	H	X	H
L	H	X	L	L
L	H	X	H	H

H = High Voltage Level, L = Low Voltage Level, X = Don't Care