

<b>Disciplina:</b> PCS 3335 – Laboratório Digital A	
<b>Prof.:</b> <i>Glauber De Bona</i>	<b>Data:</b> 22/03
<b>Turma:</b> <i>Glauber - T04</i>	<b>Bancada:</b> 08
<b>Membros:</b>	
<i>11261531 - Enzo Bustos Da Silva</i>	
<i>10379694 - Davi Augusto Bandeira</i>	



## ***Experiência 03***

### ***Decodificadores e Multiplexadores***

## 1. Introdução

*Este experimento visa continuar os estudos com Circuitos Integrados e implementação de funções lógicas, desta vez utilizando de decodificadores e multiplexadores para a criação de circuitos lógicos, implementando funções lógicas baseadas em seleção e decisão.*

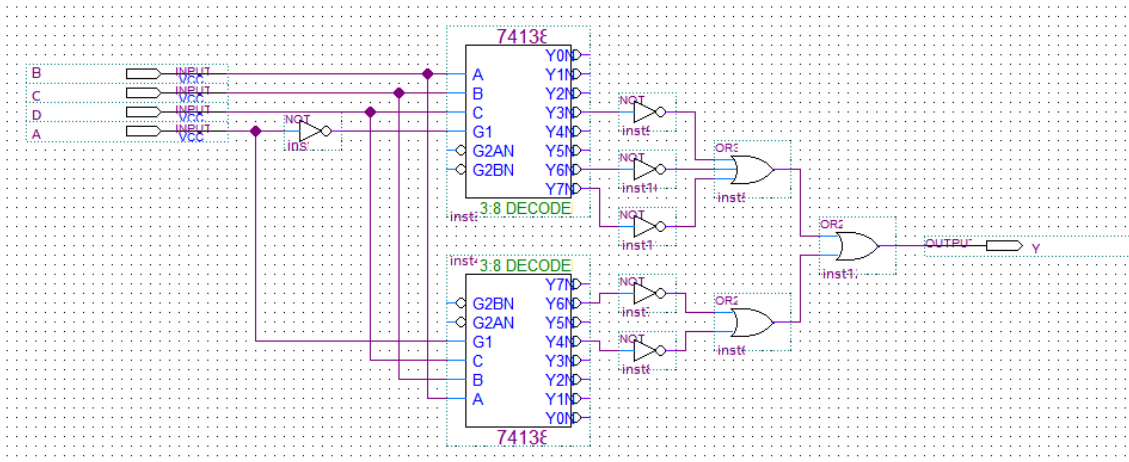
## 2. Objetivo

*O objetivo deste experimento é o de implementar funções lógicas utilizando componentes CIs mais elaborados (decodificadores e multiplexadores) de modo a criar um circuito baseado em seleção e decisão.*

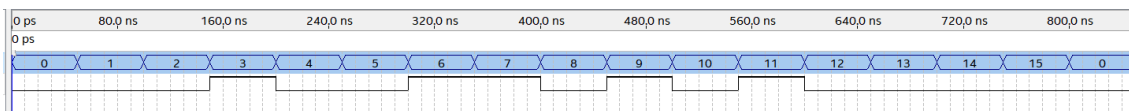
## 3. Planejamento

1. *A ideia do multiplexador é o de ser um componente que consegue transmitir diversas informações por meio de um único componente físico, agindo assim como uma chave seletora das entradas disponíveis. Já o processo de demultiplexação feito pelo decodificador consiste em selecionar qual a saída deve receber a informação contida da entrada, executando o inverso do multiplexador.*
  - a. CI 74151 → Multiplexador: [Datasheet](#)
  - b. CI 74138 → Decodificador: [Decoder](#)

2. Como o exercício pede um circuito com 4 entradas, foi realizado um cascadeamento de dois decodificadores 3x8 para obter um decodificador expandido de 4x16. O circuito final é apresentado abaixo:



3. A carta de Tempos do Circuito mencionado acima é apresentada a seguir:

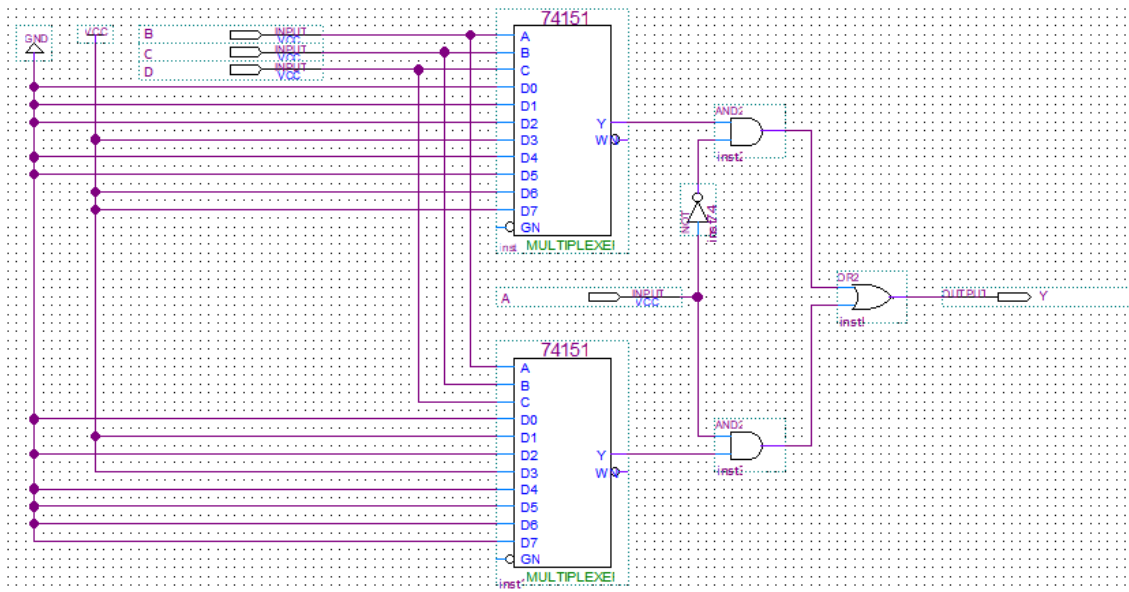


Podemos constatar que o circuito funciona como o esperado, tendo apenas valores altos nas entradas 0011 (3), 0110 (6), 0111 (7), 1001 (9) e 1011 (11).

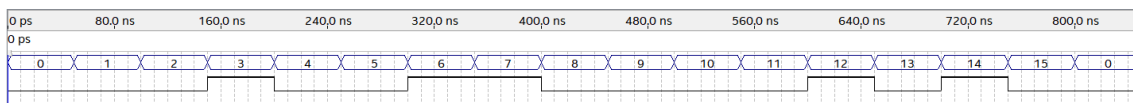
4. A Tabela de Testes do Circuito segue abaixo

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>Y esperado</b>	<b>Y obtido</b>
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

5. Como o exercício pede um circuito com 4 entradas, foi realizado um cascadeamento de dois multiplexadores para obter um multiplexador expandido. O circuito final é apresentado abaixo:



6. A carta de Tempos do Circuito mencionado acima é apresentada a seguir:



Podemos constatar que o circuito funciona como o esperado, tendo apenas valores altos nas entradas 0011 (3), 0110 (6), 0111 (7), 1001 (9) e 1011 (11).

7. A Tabela de Testes do Circuito segue abaixo

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>Y esperado</b>	<b>Y obtido</b>
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

## 4. Relatório

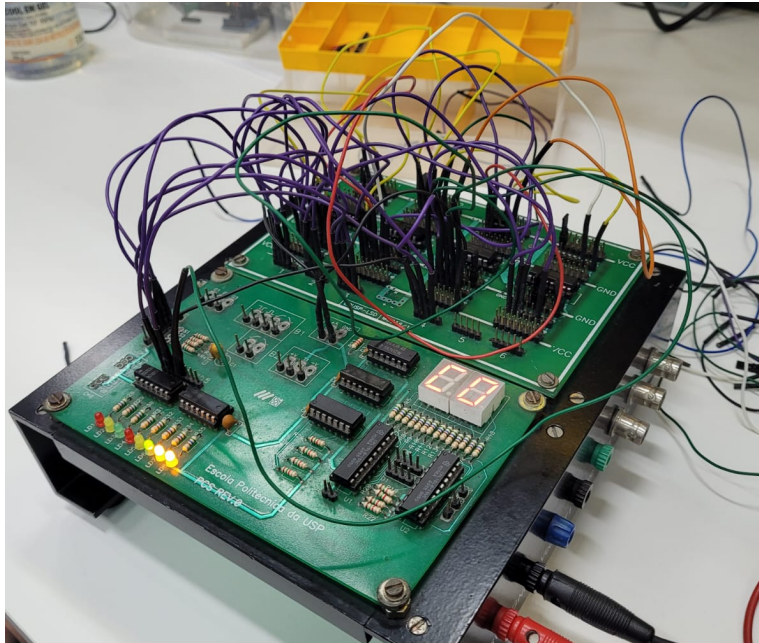
*Durante o período do Laboratório ambas as montagens foram realizadas com êxito.*

*Para a primeira, que utiliza o decodificador foram necessários 2 TTL 74138 (decodificador), 1 TTL 7404 (NOT) e 1 TTL 7432 (OR). Depois das primeiras conexões-padrão (alimentação dos CIs, GND e VCC) terem sido realizadas, as entradas foram conectadas aos LEDs e iniciou-se a montagem do Circuito dividindo-o em 4 quadrantes, cada qual que foi testado individualmente para garantir a integridade do circuito:*

- 1. Decoder Superior:*
  - a. O primeiro quadrante corresponde ao decoder superior, responsável pelo que chamamos de “entradas baixas” quando o valor de A é 0, ou seja, corresponde às entradas equivalente de 0 até 7 (0000 até 0111), essa lógica funciona, pois esse componente recebe A negado como sinal de Enable.*
  - b. Após essa montagem, constatamos que o nosso CI TTL 74138 não estava funcionando corretamente, então o trocamos por um novo e testamos novamente se estava funcionando adequadamente.*
  - c. Após esses testes, separamos as saídas de interesse (0011, 0110 e 0111) e reservamos.*
- 2. NOTs Superior + OR Superior:*
  - a. O segundo quadrante corresponde ao bloco de NOTs + OR que existe após a saída do Decodificador Superior.*
  - b. Cada uma das saídas do Decoder foi então negada usando o TTL 7404, essa operação faz com que o nosso Decoder tenha um comportamento de lógica alta (observe que o TTL 74138 apresenta a saída de interesse em 0, e as outras em 1, com os NOTs invertemos essa lógica)*
  - c. Essas saídas dos NOTs foram então passadas por 2 ORs do TTL 7432 (para realizar a operação OU3) e essa saída intermediária corresponde ao Y das entradas baixas. Que finalmente foi testado.*
- 3. Decoder Inferior:*
  - a. Montagem similar ao primeiro Decoder, porém o A entra diretamente na porta de Enable, dessa forma só ativando esse componente para as “entradas altas” equivalente de 8 até 15 (1000 até 1111).*
  - b. A montagem desse circuito correu bem e o componente funcionava de forma ideal. Após os testes, então separamos as saídas de interesse (1001 e 1011) e reservamos.*
- 4. NOTs Inferior + OR Inferior:*
  - a. Analogamente a parte 2, negamos as saídas reservadas para inverter a lógica do decoder.*
  - b. Cada uma dessas saídas negadas foi passada por um OR, dessa vez apenas foi utilizada 1 das portas do TTL 7432.*
  - c. Essa saída do OR corresponde ao Y das entradas altas. Que finalmente foi testado*

5. OR final:

- a. Após os testes em cada um dos quadrantes anteriores, ligamos as duas saídas intermediárias  $Y_{baixas}$  e  $Y_{altas}$  à última porta OR disponível no nosso circuito, essa saída é o Y do nosso sistema.



Sobre essa primeira montagem, correu sem muitos problemas. Porém vale ressaltar o problema no nosso componente que foi solucionado rapidamente devido aos testes.

Para a segunda, que utiliza o multiplexador, foram necessários 2 TTL 74151 (mux), 1 TTL 7404 (NOT) e 1 TTL 7432 (OR). Depois das primeiras conexões-padrão (alimentação dos CIs, GND e VCC) terem sido realizadas, as entradas foram conectadas aos LEDs e iniciou-se a montagem do Circuito dividindo-o em 3 fases, cada qual que foi testada individualmente para garantir a integridade do circuito:

1. Adaptação do Circuito:

- a. Como nossa Placa de Montagem só admite espaço para 4 tipos de CIs diferentes, percebemos que deveríamos alterar o circuito para remover as portas ANDs adicionais que foi usada na montagem inicial.
- b. Isso foi simples de ser resolvido, ligando a porta A diretamente no Strobe do Mux superior (que funciona como o Mux responsável pelas "entradas baixas") e o A negado foi ligado ao Strobe do Mux inferior ("entradas altas").

2. Ligação dos Mux:

- a. Após a adaptação começamos a ligar nossas entradas ao Mux, o maior problema nessa etapa foi que o Mux usado utiliza uma notação diferente do usual para o bit mais e menos significativo (geralmente ABC, enquanto este componente utiliza CBA). O que causou certa demora na montagem.



- b. Assim que corrigimos esse problema de montagem e as novas conexões foram feitas, testamos as saídas do Mux para confirmar que os componentes estavam íntegros.
- 3. Aterramento e Alimentação
  - a. Finalmente, a última fase foi ligar as saídas de interesse aos valores que devem ser refletidos no Y do circuito.
  - b. Desse modo, as saídas de interesse (0011, 0110, 0111, 1001 e 1011) foram ligadas diretamente ao VCC.
  - c. Todas as demais saídas foram aterradas.

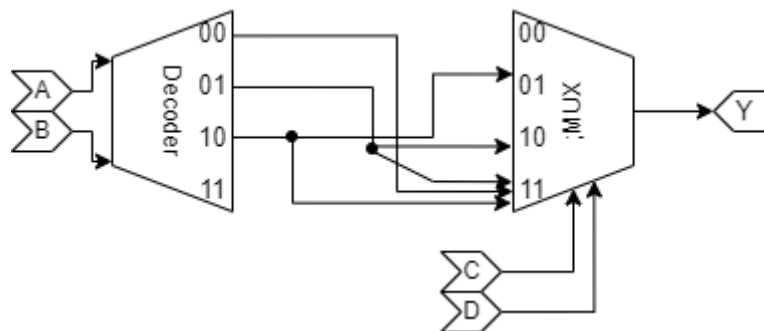
Devido ao tempo tomado, a dupla esqueceu de tirar foto da segunda montagem.

Sobre essa montagem vale ressaltar que foi necessitado certo auxílio dos professores e técnicos, primeiramente pelo Strobe, que era um conceito simples, mas que ainda não tínhamos conhecimento e também com a questão do padrão diferente das entradas do multiplexador que assumidos que era o convencional utilizado em outros componentes, sem checar propriamente o Datasheet.

## 5. Desafio

A ideia do desafio é implementar a mesma função lógica que temos, utilizando apenas um decodificador 2x4 e um multiplexador 4x1.

O design inicial de um possível circuito que poderia resolver essa questão está mostrado abaixo:



## Apêndices

Não constam para esse experimento.

## Referências

Não constam para esse experimento.