**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA Kỹ Thuật Máy Tính**

**NGUYỄN PHƯỚC HƯNG – 21520252**

**Tổ chức và Cấu trúc Máy tính II/ IT012.M11**

**GIẢNG VIÊN HƯỚNG DẪN**

**<TÊN GIẢNG VIÊN HƯỚNG DẪN>**

**TP. HỒ CHÍ MINH, <NĂM>**

MỤC LỤC

[Chương 1. BÁO CÁO LAB01 1](#_Toc91018178)

[1.1. Báo cáo phần thực hành 1](#_Toc91018179)

[1.1.1. Mô phỏng chức năng các cổng luận lý 1](#_Toc91018180)

[1.1.2. Mô phỏng chức năng các thiết bị lưu trữ 4](#_Toc91018181)

[1.2. Báo cáo phần bài tập 6](#_Toc91018182)

[1.2.1. Mô phỏng mạch tổ hợp 6](#_Toc91018183)

[1.2.2. Mô phỏng mạch tuần tự 7](#_Toc91018184)

[Chương 2. TÊN CHƯƠNG 2 11](#_Toc91018185)

[2.1. Báo cáo phần thực hành 11](#_Toc91018186)

[2.1.1. Mô phỏng ALU 11](#_Toc91018187)

[2.1.2. Mô phỏng Register Files gồm 4 thanh ghi 8 bit 12](#_Toc91018188)

[2.2. Báo cáo phần bài tập 12](#_Toc91018189)

[2.2.1. Cải tiến ALU 12](#_Toc91018190)

[2.2.2. Thiết kế lại Register Files 14](#_Toc91018191)

[TÊN CHƯƠNG 3 19](#_Toc91018192)

[2.3. Chủ đề cấp độ 2 19](#_Toc91018193)

[2.3.1. Chủ đề cấp độ 3 19](#_Toc91018194)

[2.4. Chủ đề cấp độ 2 19](#_Toc91018195)

DANH MỤC HÌNH

[Hình 1.1.1.1 AND 1](#_Toc91018335)

[Hình 1.1.1.2 OR 1](#_Toc91018336)

[Hình 1.1.1.3 NOT 2](#_Toc91018337)

[Hình 1.1.1.4 XOR 2](#_Toc91018338)

[Hình 1.1.1.5 XNOR 3](#_Toc91018339)

[Hình 1.1.1.6 NAND 3](#_Toc91018340)

[Hình 1.1.1.7 NOR 4](#_Toc91018341)

[Hình 1.1.2.1 D flipflop 4](#_Toc91018342)

[Hình 1.1.2.2 D latch 5](#_Toc91018343)

[Hình 1.1.2.3 Thanh ghi 6](#_Toc91018344)

[Hình 1.2.1.1 6](#_Toc91018345)

[Hình 1.2.2.1 Mạch tuần tự 7](#_Toc91018346)

[Hình 1.2.2.2 9](#_Toc91018347)

[Hình 1.2.2.3 9](#_Toc91018348)

[Hình 1.2.2.4 Thanh ghi 16 bit( 4 thanh 4 bit) 10](#_Toc91018349)

[Hình 2.1.1.1 Cấu tạo ALU 11](#_Toc91018350)

[Hình 2.1.2.1 Cấu tạo Register Files 12](#_Toc91018351)

[Hình 2.2.1.1 Cấu tạo ALU cải tiến 13](#_Toc91018352)

[Hình 2.2.2.1 Cấu tạo Register Files(2) 14](#_Toc91018353)

[Hình 2.2.2.2 Cấu tạo Mux4to1 15](#_Toc91018354)

[Hình 2.2.2.3 Cấu tạo Decode2to4 16](#_Toc91018355)

[Hình 2.2.2.4 Bộ cộng 8bit 17](#_Toc91018356)

[Hình 2.2.2.5 Mạch so sánh 4bit 17](#_Toc91018357)

DANH MỤC BẢNG

[Bảng 1.1.1.1 AND 1](#_Toc91018219)

[Bảng 1.1.1.2 OR 1](#_Toc91018220)

[Bảng 1.1.1.3 NOT 2](#_Toc91018221)

[Bảng 1.1.1.4 XOR 2](#_Toc91018222)

[Bảng 1.1.1.5 XNOR 3](#_Toc91018223)

[Bảng 1.1.1.6 NAND 3](#_Toc91018224)

[Bảng 1.1.1.7 NOR 4](#_Toc91018225)

[Bảng 1.2.1.1 Bảng chân trị 7](#_Toc91018226)

[Bảng 1.2.2.1 7](#_Toc91018227)

[Bảng 1.2.2.2 9](#_Toc91018228)

[Bảng 1.2.2.3 10](#_Toc91018229)

[Bảng 2.1.1.1 Cấu tạo ALU 11](#_Toc91018230)

[Bảng 2.1.2.1 Bảng chân trị Register Files 12](#_Toc91018231)

[Bảng 2.2.1.1 Bảng chân trị ALU cải tiến 13](file:///D:\IT012\maubaocao%20(1).docx#_Toc91018232)

[Bảng 2.2.2.1 Bảng chân trị Mux4to1 15](#_Toc91018233)

[Bảng 2.2.2.2 Bảng chân trị Decode2to4 16](#_Toc91018234)

DANH MỤC TỪ VIẾT TẮT

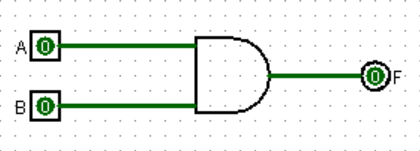
Chương 1. BÁO CÁO LAB01

## Báo cáo phần thực hành

### Mô phỏng chức năng các cổng luận lý

Cổng AND

Hình 1.1.1.1 AND



Bảng 1.1.1.1 AND

|  |  |  |
| --- | --- | --- |
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Cổng OR

Hình 1.1.1.2 OR

Diagram

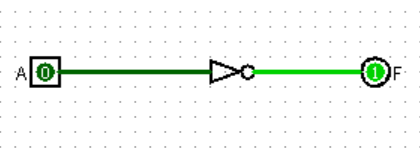
Description automatically generated

Bảng 1.1.1.2 OR

|  |  |  |
| --- | --- | --- |
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Cổng NOT

Hình 1.1.1.3 NOT

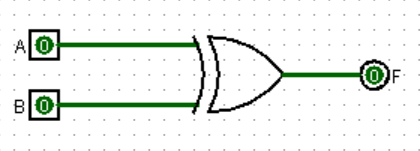


Bảng 1.1.1.3 NOT

|  |  |
| --- | --- |
| A | F |
| 0 | 1 |
| 1 | 0 |

Cổng XOR

Hình 1.1.1.4 XOR

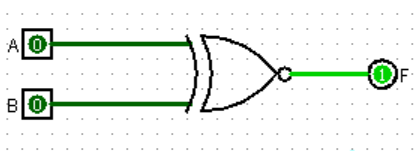


Bảng 1.1.1.4 XOR

|  |  |  |
| --- | --- | --- |
| A | B | F |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Cổng XNOR

Hình 1.1.1.5 XNOR

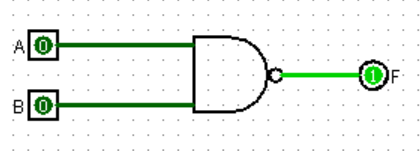


Bảng 1.1.1.5 XNOR

|  |  |  |
| --- | --- | --- |
| A | B | F |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Cổng NAND

Hình 1.1.1.6 NAND

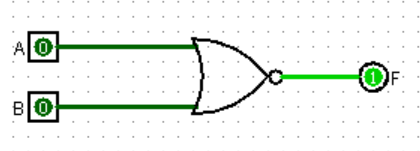


Bảng 1.1.1.6 NAND

|  |  |  |
| --- | --- | --- |
| A | B | F |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Cổng NOR

Hình 1.1.1.7 NOR



Bảng 1.1.1.7 NOR

|  |  |  |
| --- | --- | --- |
| A | B | F |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

### Mô phỏng chức năng các thiết bị lưu trữ

* D flipflop

Hình 1.1.2.1 D flipflop

A screenshot of a computer

Description automatically generated with low confidence

Bảng .1 D flipflop

|  |  |  |  |
| --- | --- | --- | --- |
| CLK | D | Q | Q+ |
| - | 0 | 0 | 0 |
| - | 0 | 1 | 1 |
| - | 1 | 0 | 0 |
| - | 1 | 1 | 1 |
| ↑ | 0 | 0 | 0 |
| ↑ | 0 | 1 | 0 |
| ↑ | 1 | 0 | 1 |
| ↑ | 1 | 1 | 1 |

* D latch

Hình 1.1.2.2 D latch

A screenshot of a computer

Description automatically generated with low confidence

Bảng .2 D latch

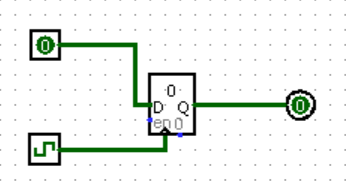
|  |  |  |  |
| --- | --- | --- | --- |
| E | D | Q | Q+ |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Khi sữ dụng D latch trong logisim, ta phải chỉnh attribute trigger sang High level để mô tả đúng cách hoạt động của D latch



* Thanh ghi

Hình 1.1.2.3 Thanh ghi



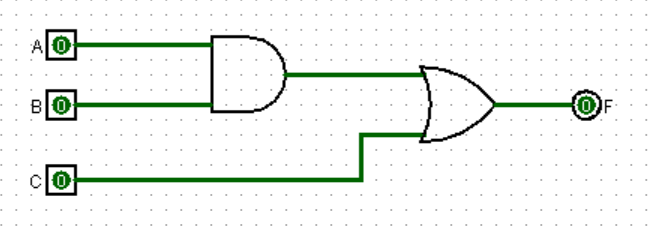
Bảng 1.1.2.3 Bảng giá trị của thanh ghi

|  |  |  |  |
| --- | --- | --- | --- |
| CLK | D | Q | Q+ |
| - | 0 | 0 | 0 |
| - | 0 | 1 | 1 |
| - | 1 | 0 | 0 |
| - | 1 | 1 | 1 |
| ↑ | 0 | 0 | 0 |
| ↑ | 0 | 1 | 0 |
| ↑ | 1 | 0 | 1 |
| ↑ | 1 | 1 | 1 |

## Báo cáo phần bài tập

### Mô phỏng mạch tổ hợp

Hình 1.2.1.1



Bảng 1.2.1.1 Bảng chân trị

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | F |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

### Mô phỏng mạch tuần tự

Hình 1.2.2.1 Mạch tuần tự

Diagram

Description automatically generated

Bảng 1.2.2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLK | A\_o\* | B\_o\* | C\_o\* | D\_o\* |
| 0 | A\_o | B\_o | C\_o | D\_o |
| 1 | A | B | C | D |

**Bài tập bổ sung**

1. Trình bày ngắn gọn chức năng và nguyên lý hoạt động D-Flipflop, Thanh ghi.

Trả lời:

* D-Flipflop là thiết bị điện tử có khả năng lưu trữ một bit nhị phân và có thể thay đổi được nhờ CLK(clock)
* Trong khi ngõ vào quyết định ngõ ra sẽ là gì thì, CLK(clock) sẽ quyết định khi nào có sự thay đổi đó, cụ thể khi xung CLK tác động theo cạnh lên thì ngõ ra sẽ bằng ngõ vào và khi xung tác động theo cạnh xuống thì ngõ ra không thay đổi.

1. Phân biệt sự khác nhau giữa mạch tổ hợp và mạch tuần tự.

Trả lời:

Mạch tổ hợp là mạch mà đầu ra chỉ phụ thuộc vào đầu vào hiện tại

Mạch tuần tự là mạch mà đầu ra phụ thuộc vào đầu vào hiện tại và đầu vào quá khứ

1. Clock( xung nhịp) CPU là gì, các trạng thái của clock.

Trả lời:

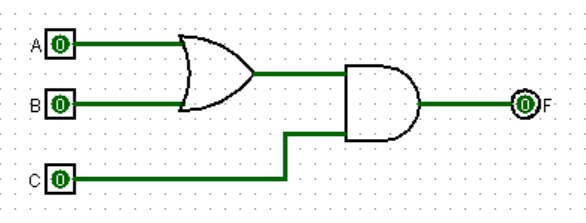
Clock(xung nhịp) CPU là một mạch tạo xung nhằm tạo mối quan hệ thời gian để cho phép các tác vụ có thể thực hiện một cách tuần tự, trước sau hoặc đồng thời cùng lúc, nó như một “đồng hồ” cho phép máy tính có thể tham chiếu thời gian để hoạt động chính xác

Clock có hai trạng thái là cạnh lên và cạnh xuống.

1. Mô phỏng mạch bằng logisim

Biến đổi

Hình 1.2.2.2



Bảng 1.2.2.2

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | F |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Biến đổi

=

Hình 1.2.2.3

Diagram

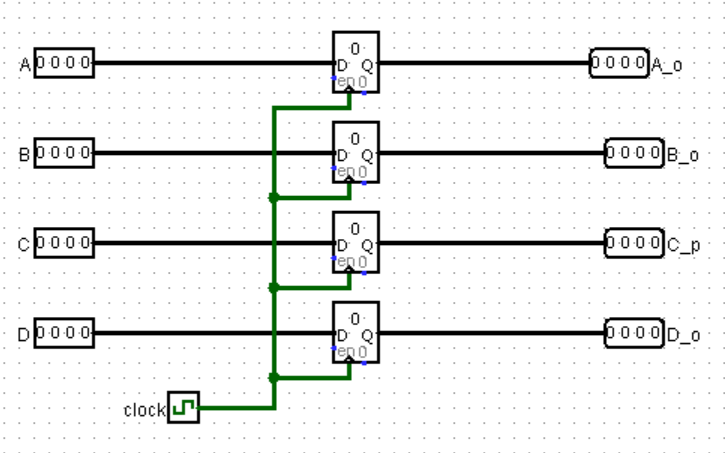
Description automatically generated

Bảng 1.2.2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | F |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

1. Thiết kế lại mạch tuần tự ở 1.2.1 với 16bit dữ liệu

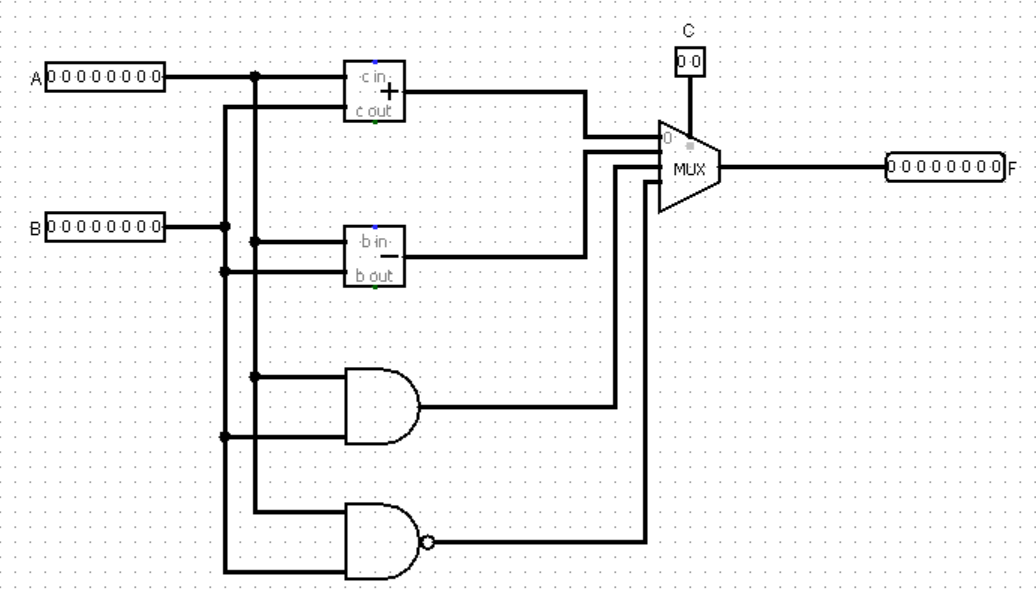
Hình 1.2.2.4 Thanh ghi 16 bit( 4 thanh 4 bit)



# BÁO CÁO LAB02

## Báo cáo phần thực hành

### Mô phỏng ALU

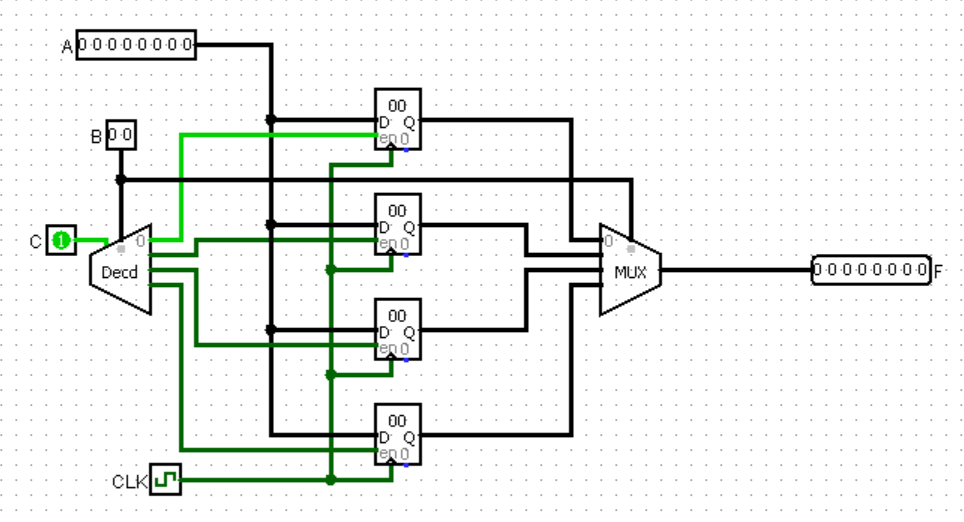


Hình 2.1.1.1 Cấu tạo ALU

Bảng 2.1.1.1 Cấu tạo ALU

|  |  |
| --- | --- |
| C | F |
| 00 | A+B |
| 01 | A-B |
| 10 | A AND B |
| 11 | NOT (A AND B) |

### Mô phỏng Register Files gồm 4 thanh ghi 8 bit



Hình 2.1.2.1 Cấu tạo Register Files

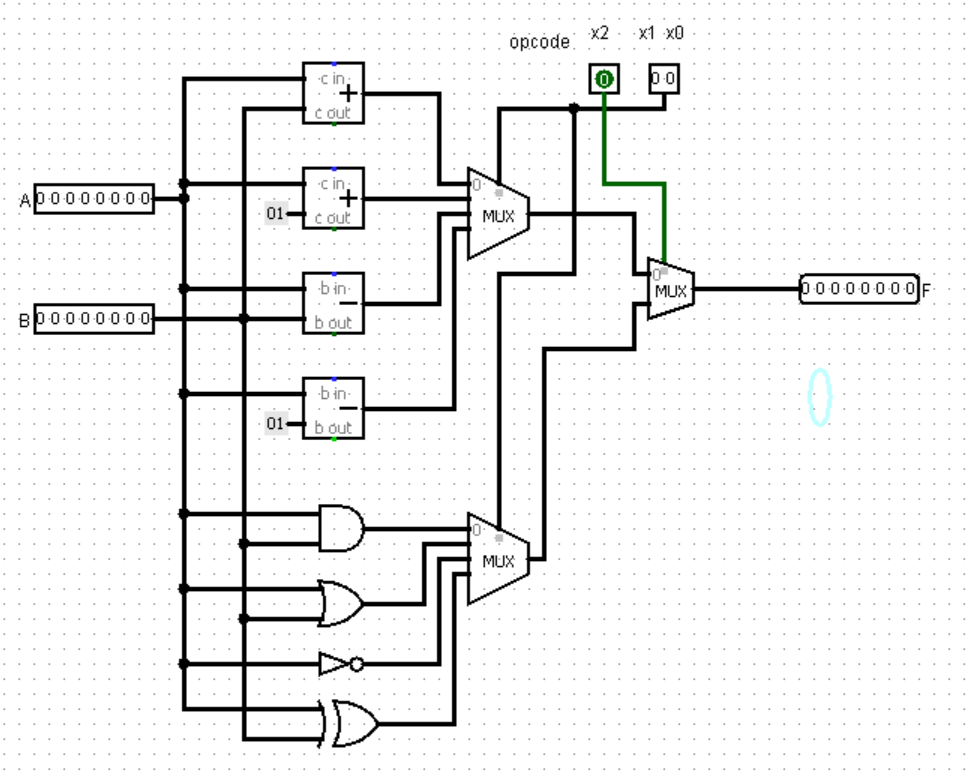
|  |  |  |
| --- | --- | --- |
| C | CLK | F\* |
| 0 | 0 | F\* |
| 0 | 1 | F\* |
| 1 | 0 | F\* |
| 1 | 1 | A |

Bảng 2.1.2.1 Bảng chân trị Register Files

## Báo cáo phần bài tập

### Cải tiến ALU

Cải tiến ALU với các phép toán: A + B, A + 1, A – B, A – 1, A AND B, A OR B, NOT A, A XOR B

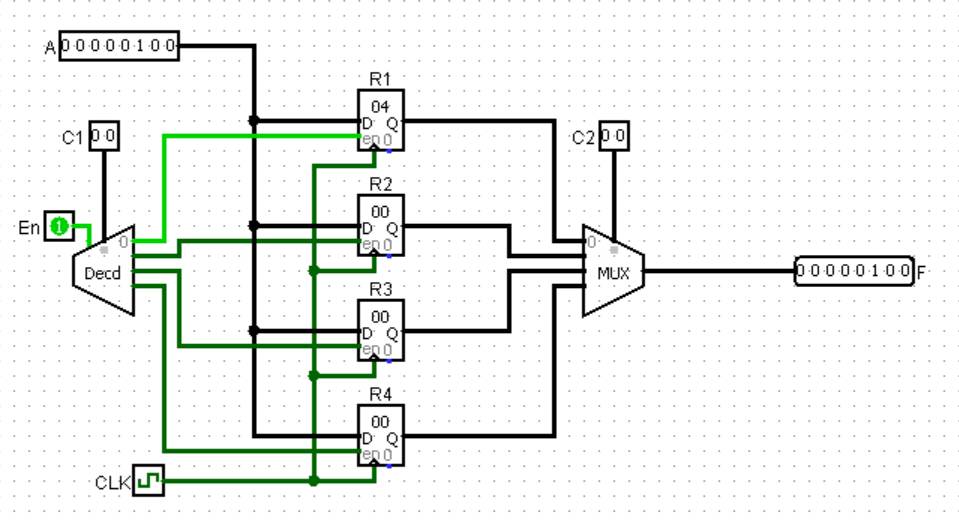


Hình 2.2.1.1 Cấu tạo ALU cải tiến



Bảng 2.2.1.1 Bảng chân trị ALU cải tiến

### Thiết kế lại Register Files

Thiết kế và mô phỏng lại Register Files với địa chỉ xuất riêng với địa chỉ ghi

Hình 2.2.2.1 Cấu tạo Register Files(2)

**Bài tập Bổ sung**

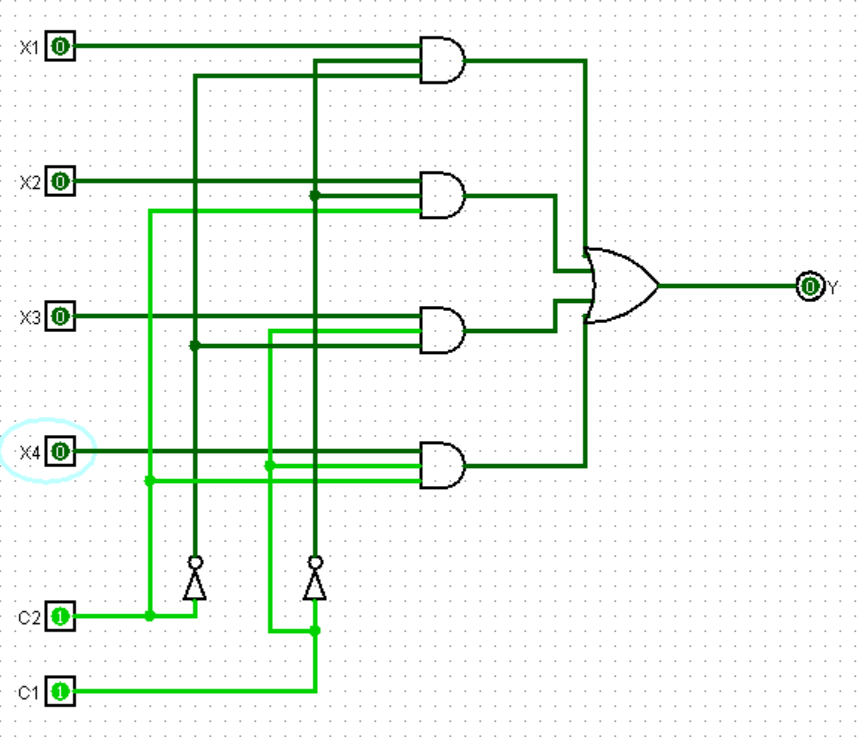
1. Phân biết Mux và Decode? Thiết kế mux 4to1 và decode 2to4 bằng các cổng luận lý.

Trả lời:

Mux là mạch có chức năng chọn lần lượt 1 trong N kênh vào để đưa đến ngõ ra duy nhất.

Decode là mạch biến đổi tín hiệu đầu vào nhị phân “n” thành mã tương đương sử dụng 2n đầu ra.

Thiết kế Mux4to1:

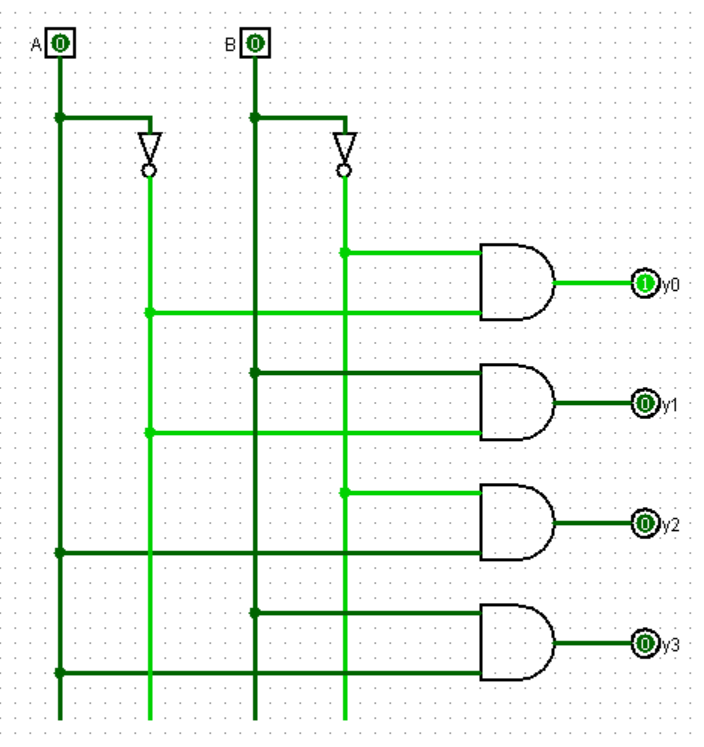


Hình 2.2.2.2 Cấu tạo Mux4to1

|  |  |  |
| --- | --- | --- |
| C1 | C2 | F |
| 0 | 0 | X1 |
| 0 | 1 | X2 |
| 1 | 0 | X3 |
| 1 | 1 | X4 |

Bảng 2.2.2.1 Bảng chân trị Mux4to1

Thiết kế decode2to4:



Hình 2.2.2.3 Cấu tạo Decode2to4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| B | A | x0 | x1 | x2 | x3 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

Bảng 2.2.2.2 Bảng chân trị Decode2to4

1. Thiết kế lại bộ cộng có chức năng cộng 2 số 8 bit.

Trả lời:

Diagram

Description automatically generated

Hình 2.2.2.4 Bộ cộng 8bit

1. Sinh viên thiết kế mạch có chức năng so sánh hai input 4 bit có bằng nhau hay không? Trường hợp bằng nhau, output bằng 1 ngược lại output bằng 0.

Trả lời

Diagram

Description automatically generatedHình 2.2.2.5 Mạch so sánh 4bitBÁO CÁO LAB03

## Báo cáo phần thực hành

|  |  |
| --- | --- |
| Mô phỏng | Chức Năng |
| add $s1,$s2,$s3  addi $s1,$s2,100  addu $s1,$s2,$s3  addiu,  sub,  subu,  and,  andi,  or,  nor,  lw,  sw,  slt,  slti,  sltu,  sltiu,  syscall | Cộng giá trị hai thanh ghi s2, s3 và lưu kết quả vào s1  Cộng giá trị thanh ghi s2 với một số nguyên và lưu kết quả vào s1 |

### Mô phỏng việc thực thi các câu lệnh và cho biết chức năng của chúng

TÀI LIỆU THAM KHẢO