

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN LẬP TRÌNH TRỰC QUAN**

**ĐỀ TÀI**

**XÂY DỰNG CHƯƠNG TRÌNH MÔ PHỎNG MẠCH LOGIC CƠ BẢN**

**Giảng viên hướng dẫn:** ThS. Phan Nguyệt Minh

**Nhóm thực hiện** : Nhóm 10

**Sinh viên 1** : Huỳnh Hạ Vy - 16521470

**Sinh viên 2** : Vũ Nguyên Đăng - 16521075

**Sinh viên 3** : Đỗ Ngọc Bích Trâm - 16521273

*TP. Hồ Chí Minh, 18 tháng 06 năm 2018*

# MỤC LỤC

MỤC LỤC .....	i
I. LỜI CẢM ƠN .....	1
II. GIỚI THIỆU .....	2
1. Giới thiệu đề tài .....	2
2. Các chức năng.....	3
III. PHÂN TÍCH THIẾT KẾ .....	3
1. Kiến trúc chương trình .....	3
2. Thao tác về cổng .....	4
2.1 Cấu trúc cơ bản của một cổng.....	4
2.2 Các thao tác trên cổng .....	5
3.Thao tác chuột.....	11
4. Logx .....	12
5. Save và Load file làm việc .....	13
5.1 Save .....	13
5.2 Load .....	13
IV. MINH HỌA CHƯƠNG TRÌNH .....	13
V. KẾT LUẬN.....	20
1 Kết quả đạt được .....	20
2 Hạn chế .....	20
3 Hướng phát triển .....	20
V. TÀI LIỆU THAM KHẢO .....	21

## I. LỜI CẢM ƠN

Để đồ án này hoàn thành đúng thời hạn và đạt kết quả tốt đẹp, chúng em đã nhận được sự hỗ trợ, giúp đỡ của nhiều thầy, cô khoa Công nghệ Phần Mềm – trường Đại học Công Nghệ Thông Tin. Với tình cảm sâu sắc, chân thành, cho phép em bày tỏ lòng biết ơn sâu sắc đến thầy, cô tạo điều kiện, giúp đỡ chúng em trong suốt quá trình học tập và hoàn thiện đề tài.

Trước hết, cho phép chúng em gửi lời cảm ơn sâu sắc cùng những lời chúc tốt đẹp nhất đến cô Phan Nguyệt Minh, thầy Phạm Thi Vương và cô Huỳnh Hồ Thị Mộng Trinh đã tận tình hướng dẫn cũng như giúp đỡ chúng em hoàn thành đề tài này một cách tốt đẹp nhất.

Với điều kiện thời gian cũng như kinh nghiệm còn nhiều hạn chế, đồ án này cũng không thể tránh gặp những thiếu sót. Nhóm chúng em rất mong nhận được những ý kiến góp ý từ thầy cô để có điều kiện bổ sung, nâng cao hiểu biết của mình để có thể phát triển đồ án và nhất là phát triển bản thân.

Chúng em chân thành cảm ơn !

## II. GIỚI THIỆU

### 1. Giới thiệu đề tài

Thiết kế logic là viên gạch nền của ngành Khoa học máy tính, đồng thời cũng là một trong những chủ đề quan trọng nhất của ngành này. Hầu như trong bất kỳ ngành giáo dục khoa học máy tính nào đều bắt buộc học các bài học về thiết kế logic để cung cấp cho sinh viên sự hiểu biết sâu sắc về nguyên tắc làm việc của máy tính một cách rất cơ bản để có thể hiểu được cách mà máy tính hoạt động. Bên cạnh đó, sinh viên có khả năng xây dựng thiết kế mạch riêng với sự giúp đỡ của giảng viên tại các phòng thí nghiệm thiết kế logic. Các trường đại học dành kinh phí đáng kể cung cấp cho sinh viên của họ có nhiều kinh nghiệm về thiết kế logic. Trong đồ án này chủ yếu tập trung về “Mô phỏng mạch logic cơ bản” giúp sinh viên ngành khoa học máy tính có thêm trải nghiệm và vận dụng trước khi thực hành bản thiết kế, góp phần làm giảm tỉ lệ tai nạn thường gặp trong khi thực hành và gián tiếp giúp giảm chi phí trong thực tế.

Đây là môn học cơ sở cho các môn học khác về sau, dù học kỹ lý thuyết, sinh viên có khả năng chưa định hình rõ nội dung môn học đó hoặc có thể vận dụng vào thực tiễn. Việc áp dụng lý thuyết của Thiết kế logic giúp sinh viên dễ dàng hiểu rõ hơn kiến thức của những môn học khác. Đó là ích lợi của việc học tập và nghiên cứu môn học này, điều này đòi hỏi sinh viên phải thực hành thí nghiệm. Ở đây, các mô phỏng đi vào thực nghiệm như những trợ lý hỗ trợ sinh viên trong các thí nghiệm trong không gian mô phỏng máy tính, sinh viên thoải mái phát huy ý tưởng sáng tạo mà không cần bận tâm thiết bị có thể bị hư hại hay không. Do đó, dự án này cung cấp cho sinh viên cơ hội chạy thử nghiệm những ý tưởng thú vị và thiết kế sáng tạo trong môi trường 2D mô phỏng một phòng thí nghiệm thiết kế logic thực tế. Hơn thế nữa, sinh viên không cần bỏ phí để chạy thử nghiệm các thiết kế mạch mà vẫn có thể phát triển các mạch của mình.

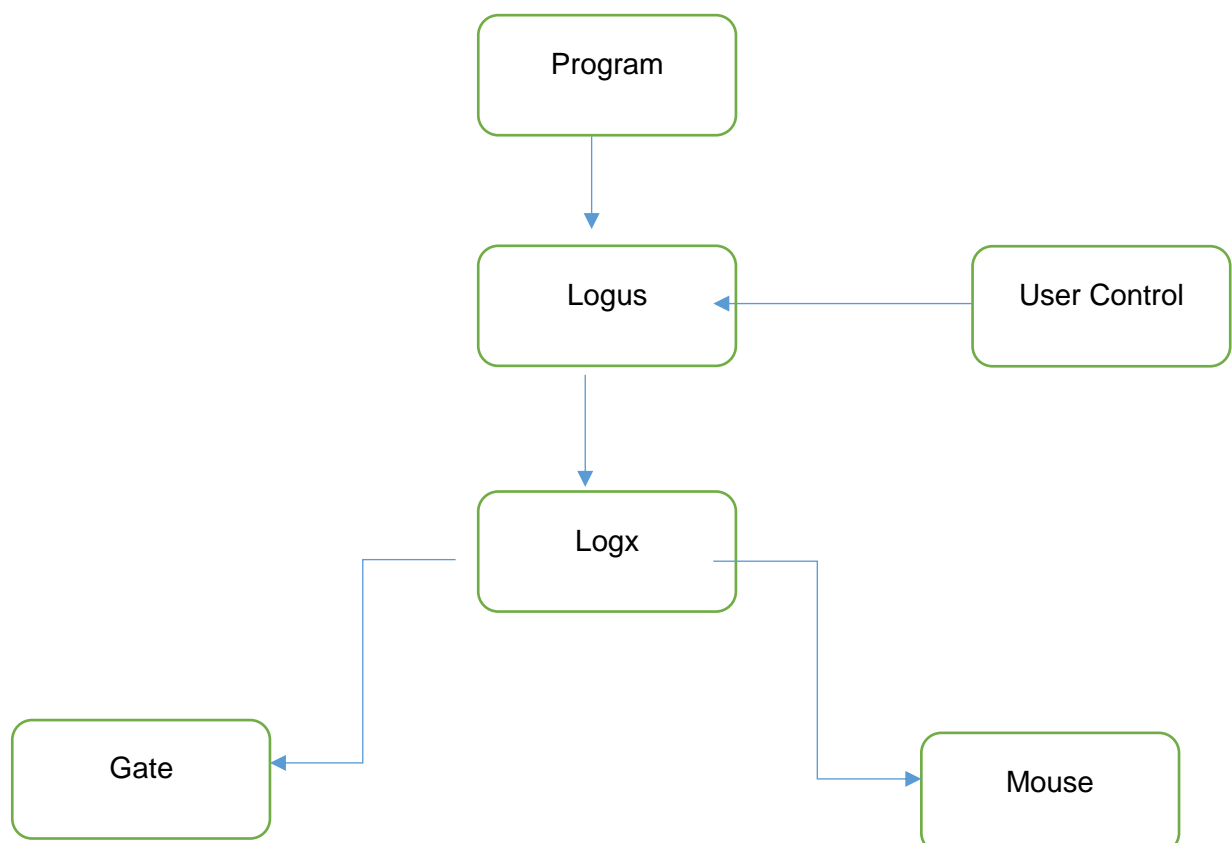
## 2. Các chức năng

Bao gồm các chức năng cơ bản như việc bạn thực hiện thiết kế logic trong phòng thí nghiệm như:

- Tạo mới, thêm, sửa, xóa, di chuyển một cách có chọn lọc ( do người dùng chọn ) các cổng logic, cổng nguồn, đèn led.
- Nối dây các cổng, đèn led, nguồn để tạo thành mạch logic cần thiết kế
- Bật / tắt công tắc nguồn để dễ theo dõi sự hoạt động của mạch
- Mô tả công thức của mạch vừa thiết kế ( người dùng chọn tùy ý một cổng thì công thức của mạch được nối cho đến cổng được chọn sẽ được hiển thị cho người dùng để theo dõi )
- Nhắc lại lý thuyết về các cổng logic cơ bản nhằm hỗ trợ một cách tối đa việc thiết kế mạch.

## III. PHÂN TÍCH VÀ THIẾT KẾ

### 1. Kiến trúc chương trình



## 2. Thao tác về cổng

Hiện tại ta chỉ tập trung vào 7 cổng cơ bản như trong môn học “Nhập Môn Mạch Số” : and, or, not, xor,...

Sử dụng kiểu dữ liệu “List” để quản lý cổng một cách dễ dàng.

### 2.1 Cấu trúc cơ bản của 1 cổng:

- + Tọa độ của cổng xét theo trục x,y
- + Thành phần output/input
- + Gate Text : dùng để hiển thị công thức của cổng
- + IsOn : biến bool dành cho 1 số cổng đặc biệt có chức năng bật/tắt
- + Render code : dùng để chọn cổng ra từ sprite
- + Số lượng input của cổng:

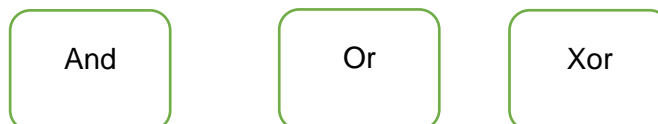
Cổng không có input: Các cổng nguồn



Cổng có 1 input:



Cổng có 2 input:



## 2.2 Các thao tác trên cổng

- + Tạo cổng
- + Xóa cổng
- + Di chuyển cổng
- + Tính output
- + Nối cổng với nhau

### Tạo cổng

- Vị trí ban đầu ở giữa màn hình :  $(\text{scene width} + \text{scene height}) / 2$
- Input (nếu có) : NULL
- Thêm cổng vào GateList
- Vẽ cổng lên màn hình làm việc theo render code

### Xóa cổng

- Tính toán lại công thức
- Loại bỏ cổng khỏi GateList
- Xóa cổng khỏi màn hình làm việc

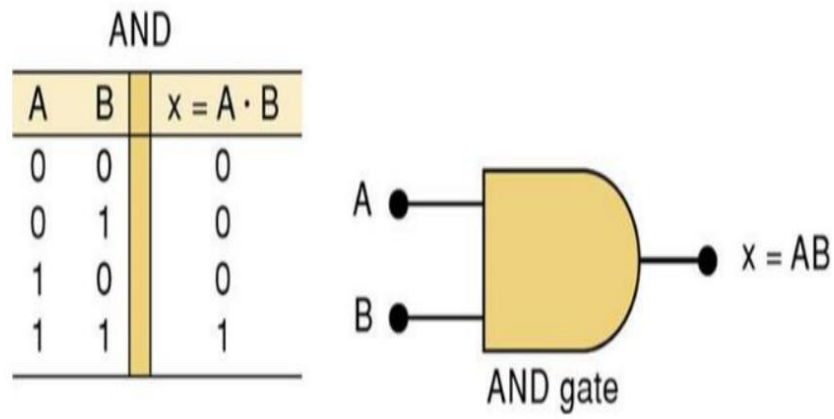
### Di chuyển cổng

- Bắt đầu khi người dùng nhấn và giữ chuột vào cổng
- Vẽ lại cổng theo vị trí của chuột theo màn hình làm việc theo từng frame
- Xóa cổng ở vị trí cũ trên màn hình làm việc theo từng frame
- Tiếp tục vẽ lại cho đến khi người dùng thả chuột

### Tính output

Để output được cập nhật thì cổng phải có output (trừ cổng đèn và cổng nguồn) và tất cả input của cổng phải khác NULL.

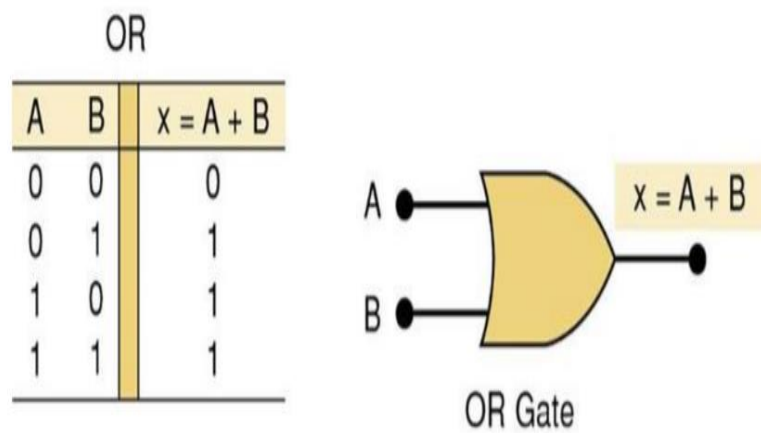
- **And**



Số lượng input : 2

Output = Input 1 && Input 2

○ **Or**

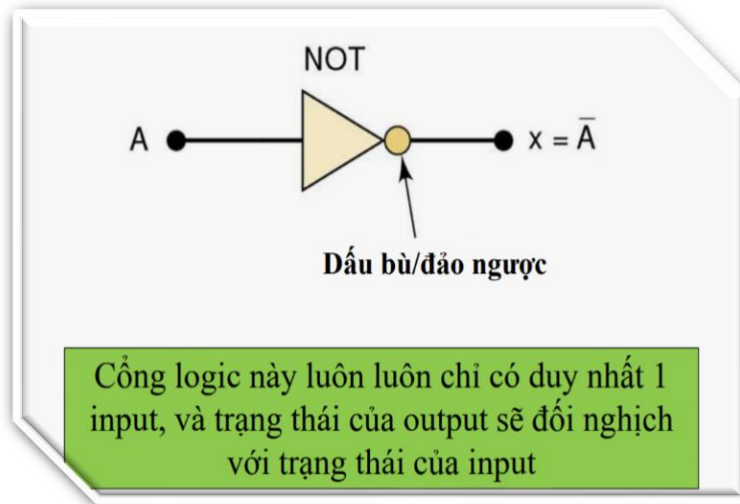


Số lượng input : 2

Output = Input 1 || Input 2

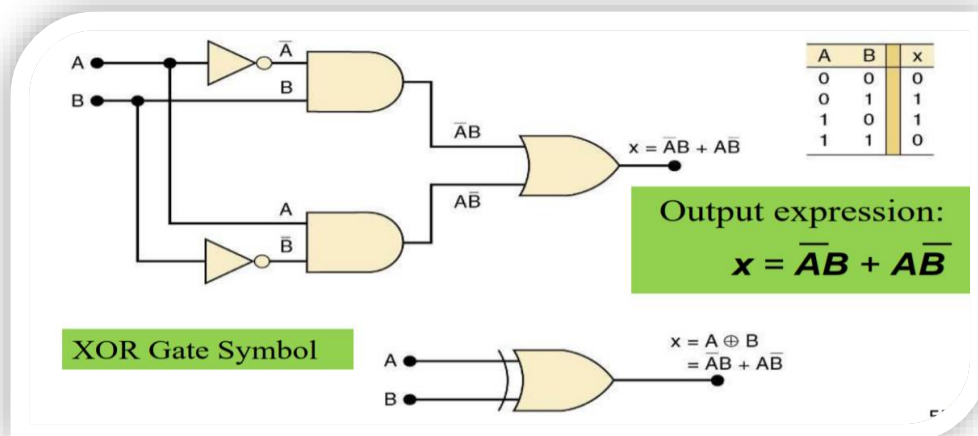
○ **Not**





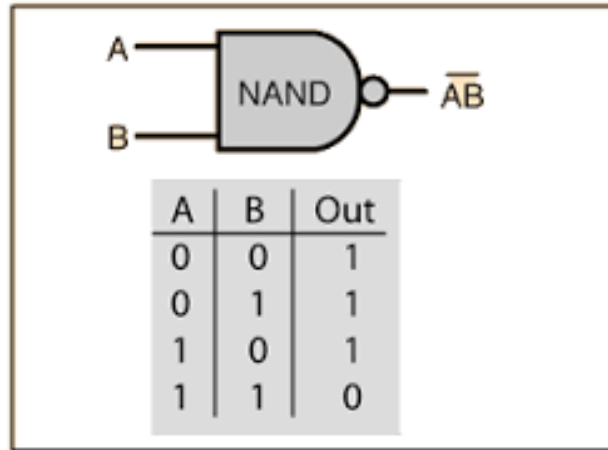
Số lượng input : 1  
Output = ! Input

### ○ Xor



Số lượng input : 2  
Output = Input 1 ^ Input 2

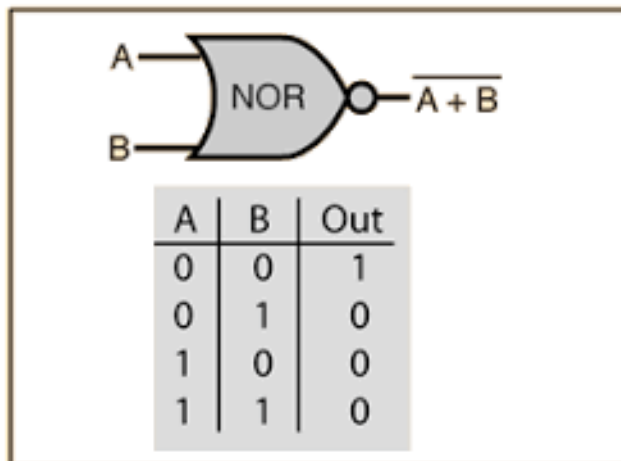
### ○ Nand



Số lượng input : 2

Output = !( Input 1 && Input 2 )

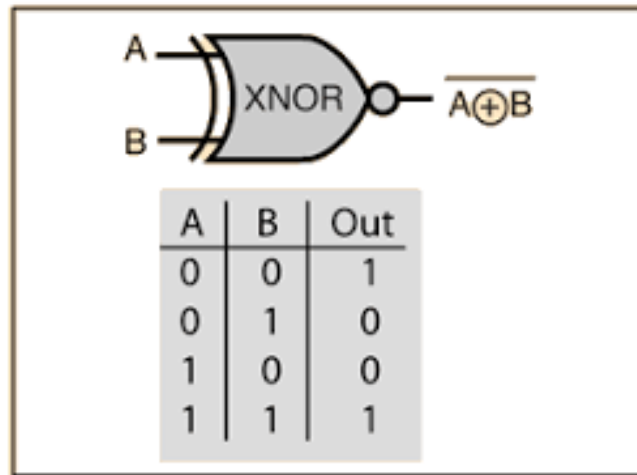
○ **Nor**



Số lượng input : 2

Output = !( Input 1 || Input 2 )

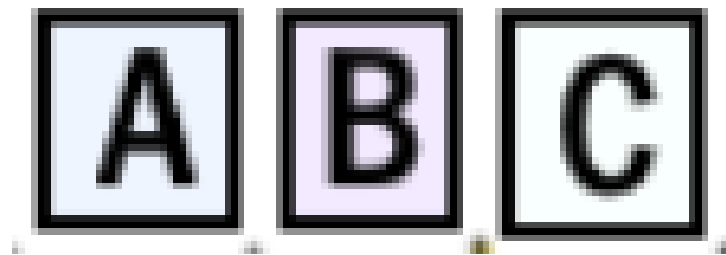
○ **Xnor**



Số lượng input : 2

Output =  $!(\text{Input 1} \wedge \text{Input 2})$

- Các cổng nguồn A, B, C



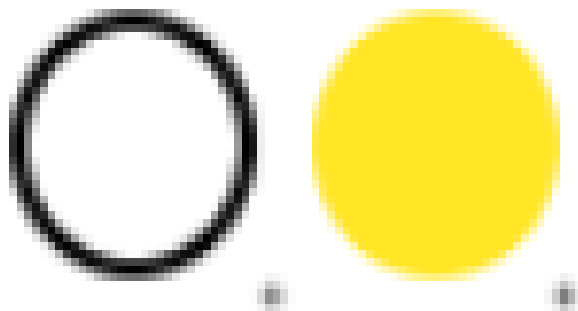
Số lượng input : 0

Giá trị output phụ thuộc vào trạng thái của công tắc được nối với cổng

Output = 1 khi công tắc ở trạng thái tích cực cao (ON)

Output = 0 khi công tắc ở trạng thái tích cực thấp (OFF)

- Đèn



Số lượng input : 1

Có 2 trạng thái cơ bản là sáng và tắt

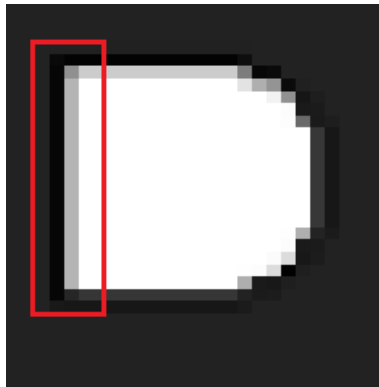
+ Đèn sáng khi tín hiệu đầu vào là 1

+ Đèn tắt khi tín hiệu đầu vào là 0

### **Nối cổng**

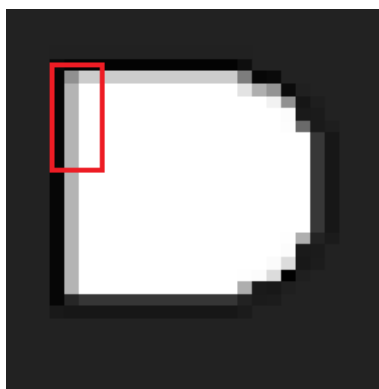
- Cổng sẽ được nối từ Input này tới Output kia

B1: Người dùng chọn vào vùng Input của cổng (Input Block)

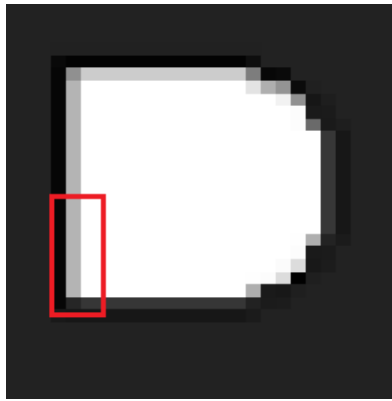


B2: Nếu cổng có input thì xác định xem người dùng chọn input trên hay input dưới

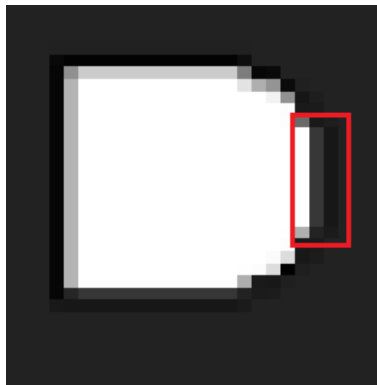
Vùng input trên :



Vùng input dưới:



B3: Người dùng chọn output cần nối đến



B4: Liên kết 2 cổng

- Cập nhập giá trị input của cổng dựa trên cổng được nối
- Cập nhập lại output của cổng dựa theo input vừa được nối
- Vẽ đường nối 2 cổng

### 3. Thao tác chuột

Dùng để quản lý các thao tác của chuột được dễ dàng và xử lý dc các sự kiện như “drag” chuột

#### a) Thành phần

- + Vị trí trên trục tọa độ x,y
- + Trạng thái chuột: tên nút chuột – trạng thái nút chuột (bool)

#### b) Chi tiết

Có ba nút chuột ta cần ghi nhận là chuột trái, chuột phải và chuột giữa

 Khởi tạo :

- Trạng thái của cả 3 nút chuột là false


 MouseDown

- Trạng thái của nút chuột được nhấn thành true


 MouseUp

- Trạng thái của nút chuột được thả từ true thành false

### c) Các sự kiện chuột chính

 Click chuột chọn cổng

- Vị trí chuột nằm trên cổng
- Trạng thái chuột hiện tại có nút chuột được nhấn
- Trạng thái chuột trước đó nút chuột được nhấn không được nhấn

 Click và drag chuột để di chuyển cổng

- Vị trí chuột nằm trên cổng
- Trạng thái chuột hiện tại có nút chuột (trái) được nhấn
- Trạng thái chuột trước đó nút chuột được nhấn không được nhấn
- Trong khi trạng thái nhấn của chuột (trái) vẫn giữ nguyên bằng true nghĩa là người dùng vẫn đang giữ và di chuyển chuột
  - > tiếp tục di chuyển cổng
- Khi trạng thái chuột (trái) chuyển sang false nghĩa là người dùng đã kết thúc công việc click and drag chuột
  - > Kết thúc sự kiện là ngừng di chuyển cổng

## 4. Logx

Là hàm điều khiển chính của chương trình

Gồm 3 thành phần chính

i) FrameUpate

Dùng để xử lý phần logic các thao tác được người dùng thực hiện trên màn hình là việc


- ii) Hàm điều chỉnh frame trên giấy
- iii) FrameRender

Dùng để xử lý các thay đổi về giao diện xảy ra do các thao tác nhười dùng thực hiệ trên màn hình làm việc

## 5. Save và Load file làm việc

### 5.1 Save

Do độ phức tạp không quá cao nên ta có thể lưu dữ liệu của chương trình thành rất nhiều loại file, miễn là ta có thể ghi và đọc dữ liệu dạng text từ loại file đó. Ở đây ta chọn file “.txt” vì độ thông dụng và dễ dàng cho người dùng

 Nội dung file save

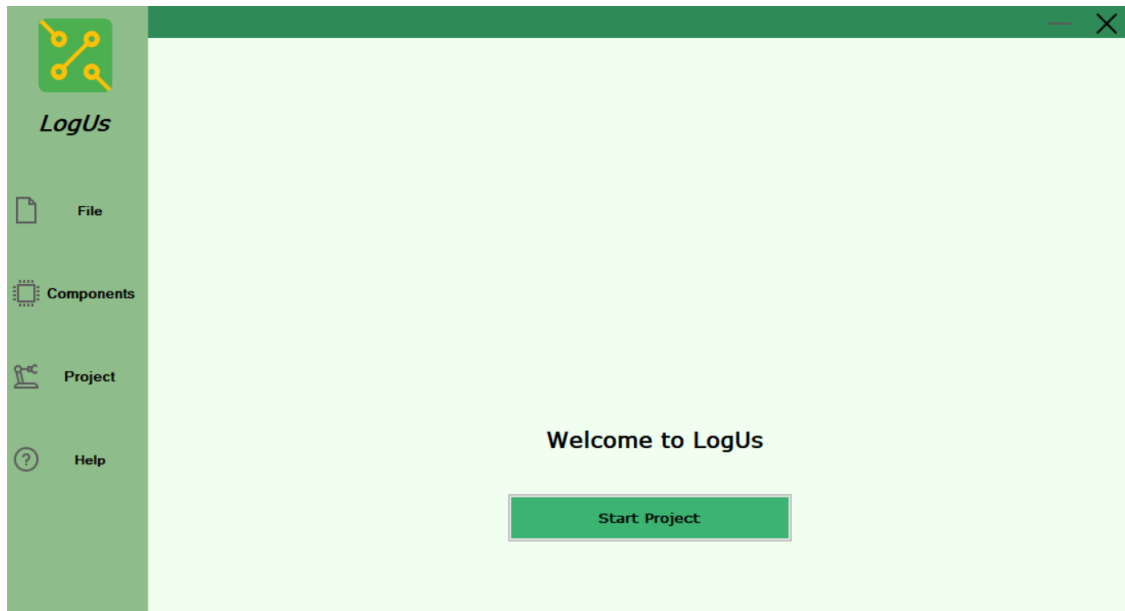
- i) Số lượng cổng
- ii) Danh sách cổng
  - a. Vị trí của cổng trên trục tọa độ
  - b. Render code của cổng để vẽ cổng từ file sprite
- iii) Các liên kết
  - a. Số thứ tự trong danh sách cổng của cổng có liên kết
  - b. Số thứ tự trong danh sách cổng của cổng được liên kết đến
  - c. Số thứ tự của liên kết trong Input (Dành cho cổng có hai Input)

### 5.2 Load

- B1: Lọc các file có định dạng phù hợp
- B2: Người dùng chọn file cần mở
- B3: Lưu thông số các cổng vào danh sách theo dữ liệu đọc được
- B4: Vẽ các cổng lên màn hình làm việc theo dữ liệu đọc được
- B5: Vẽ các liên kết lên màn hình làm việc theo dữ liệu đọc được
- B6: Tính toán lại output của các cổng

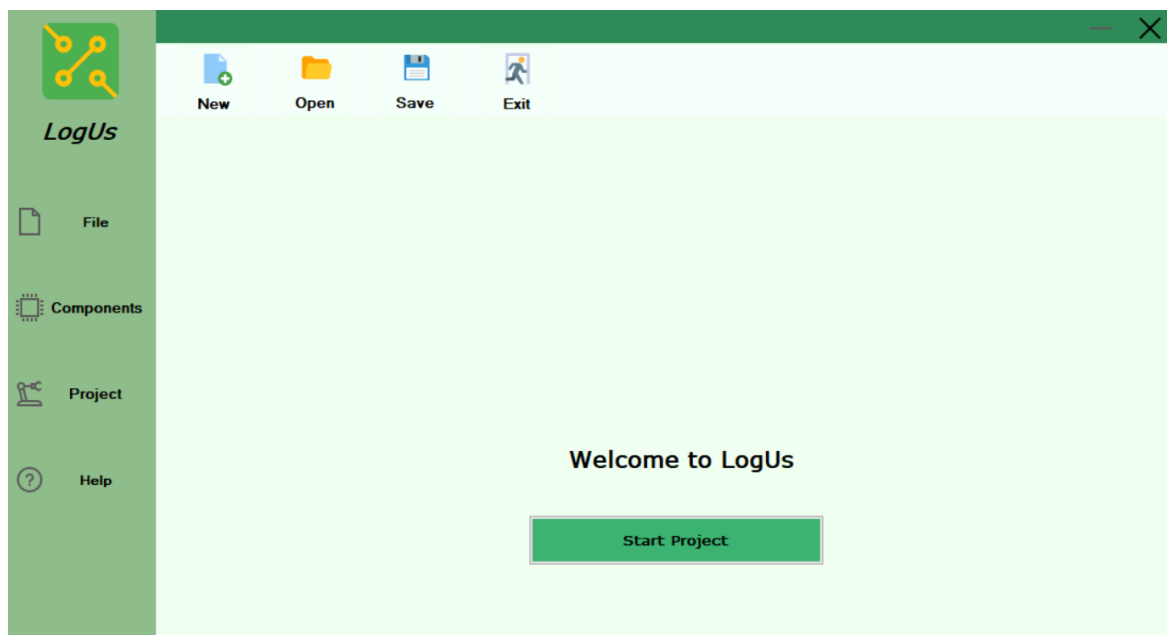
## IV. MINH HỌA CHƯƠNG TRÌNH

### 1. Màn hình khởi động



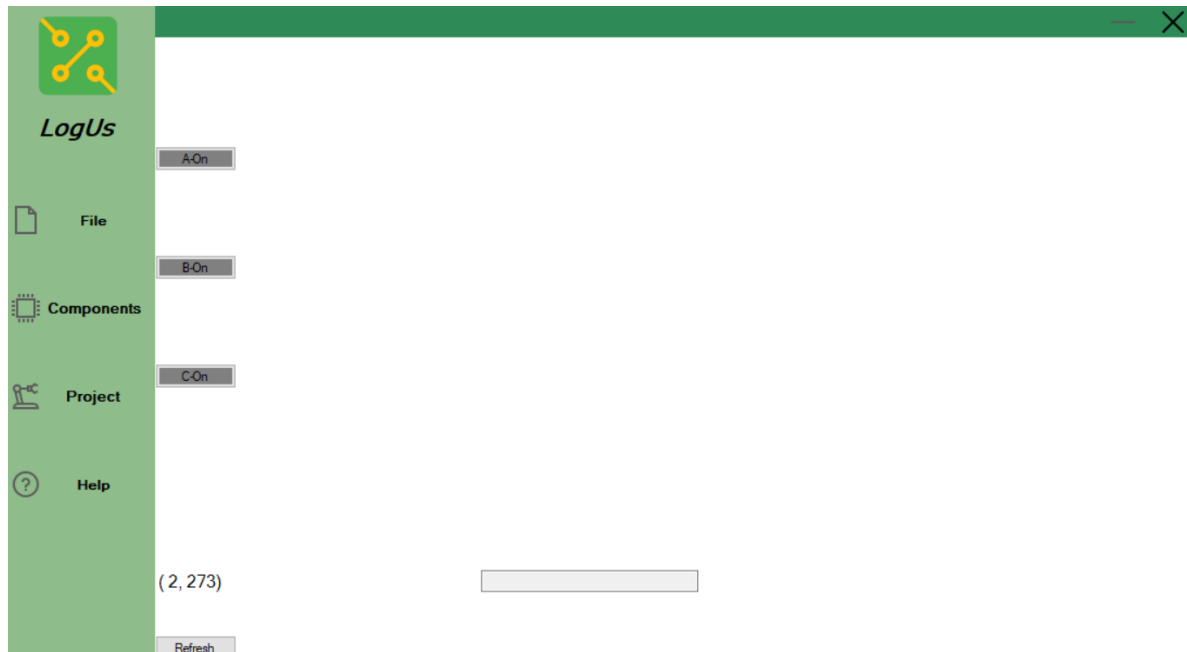
## 2. Màn hình các chức năng

### 2.1 Click chọn File

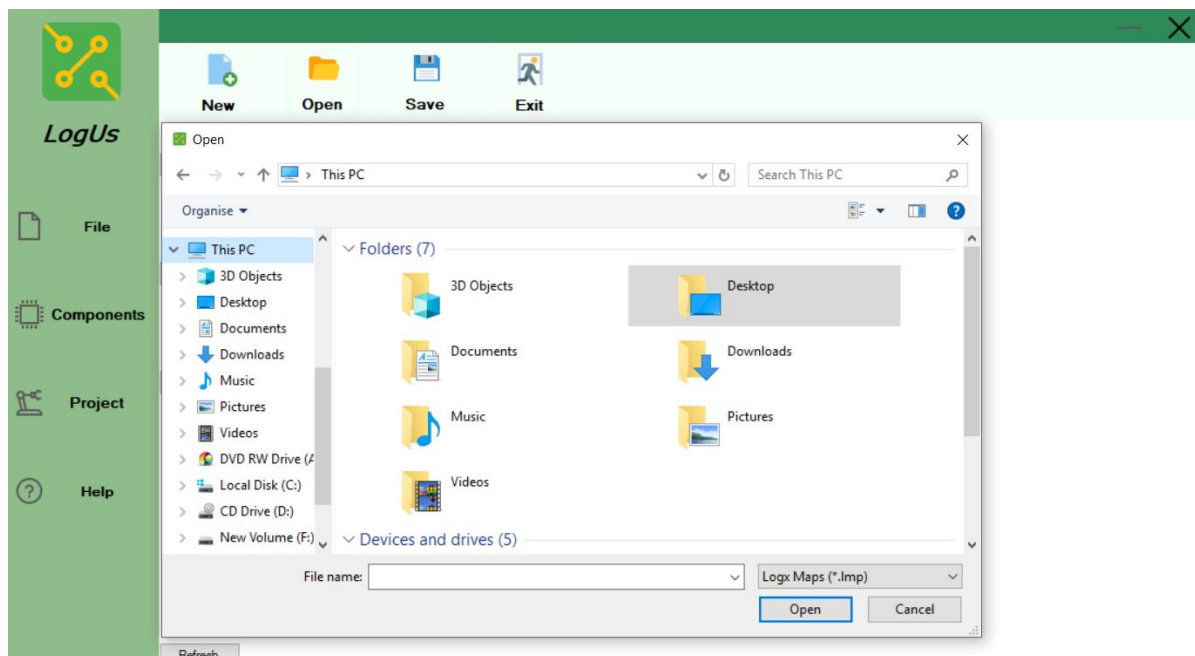


#### 2.1.1 Click chọn New : bắt đầu project mới

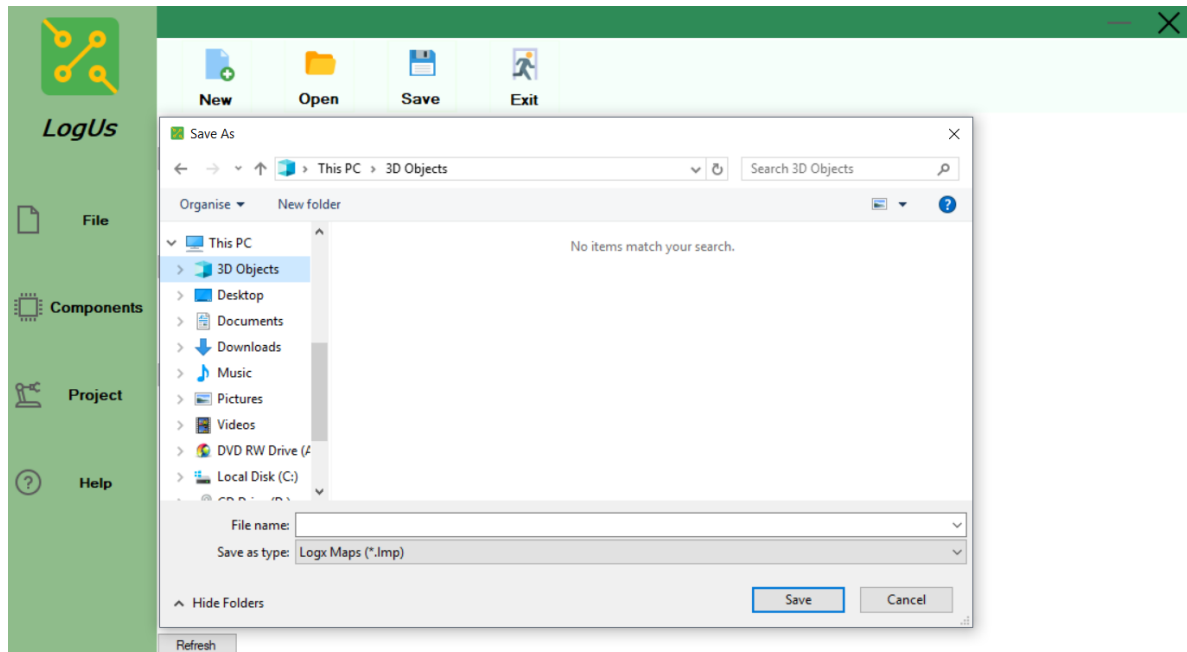




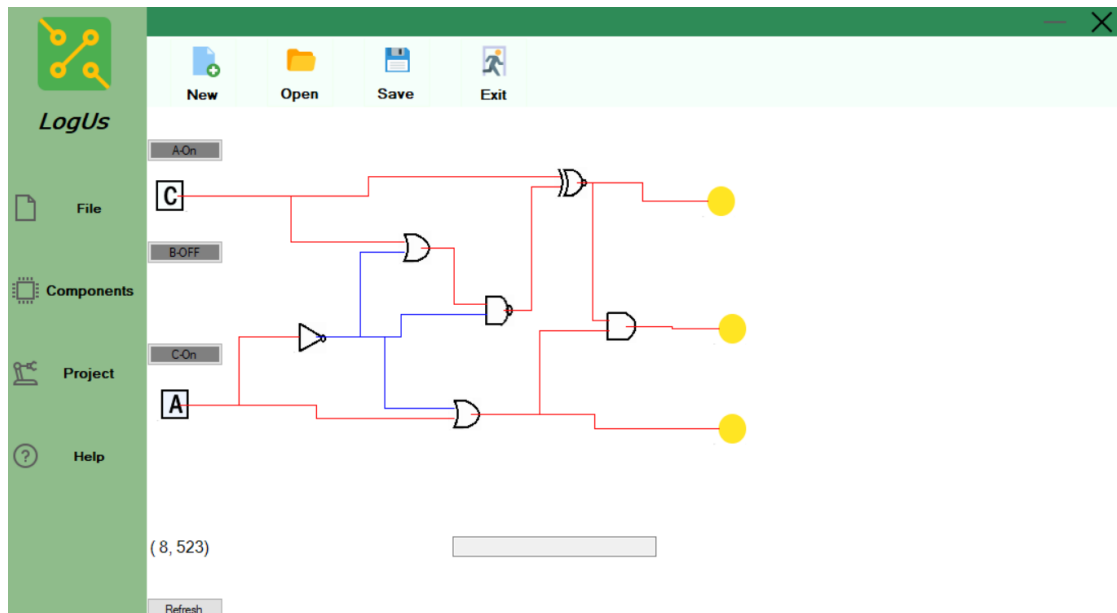
### 2.1.2 Click chọn Open : mở project có sẵn



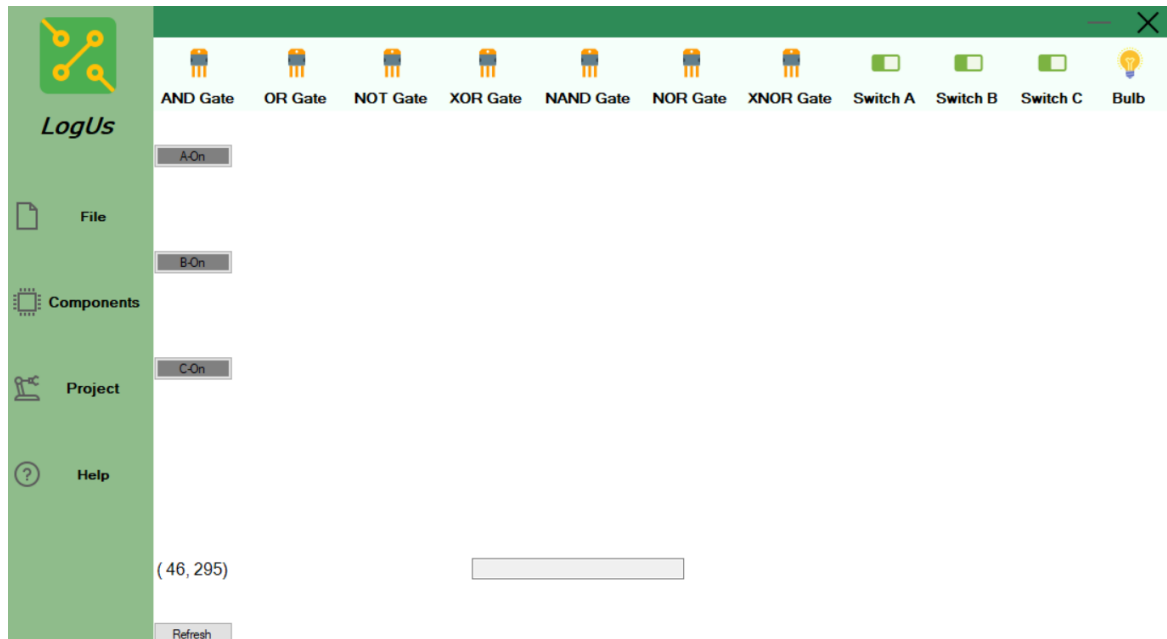
### 2.1.3 Click chọn Save : lưu project vừa thực hiện



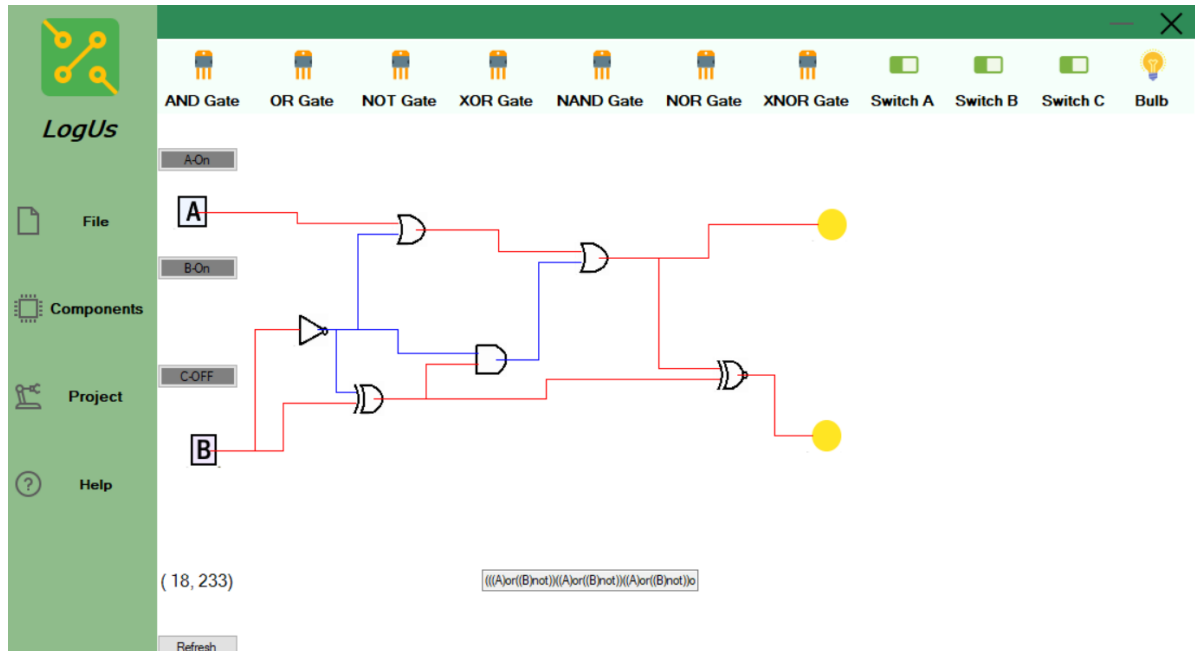
#### 2.1.4 Click chọn Exit : thoát chương trình



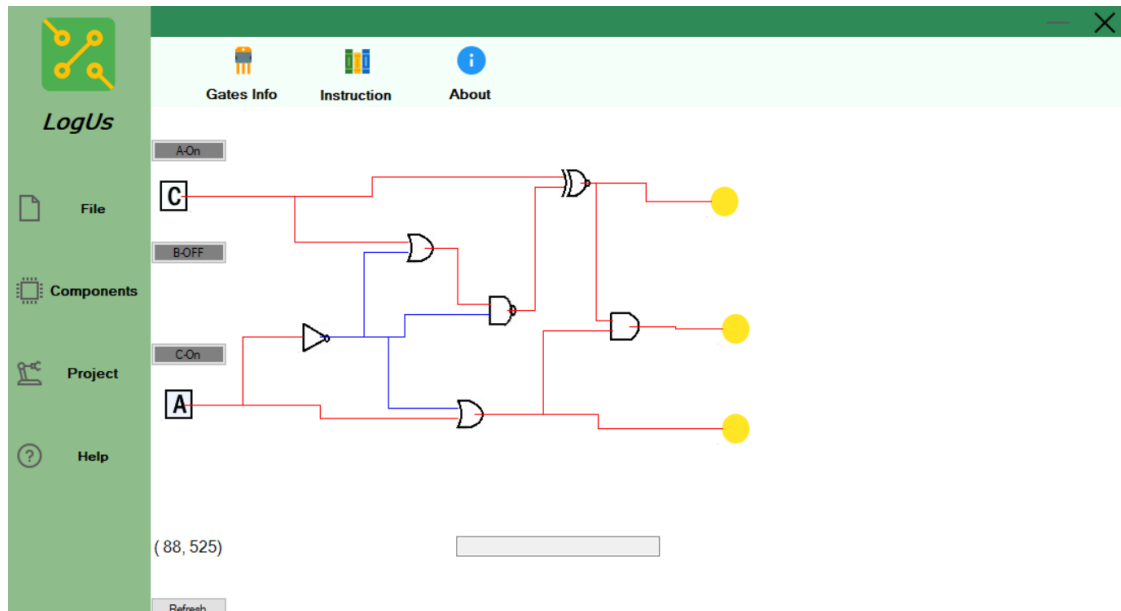
#### 2.2 Click chọn Components : hiển thị các button cổng, công tắc và đèn cho người dùng để thao tác



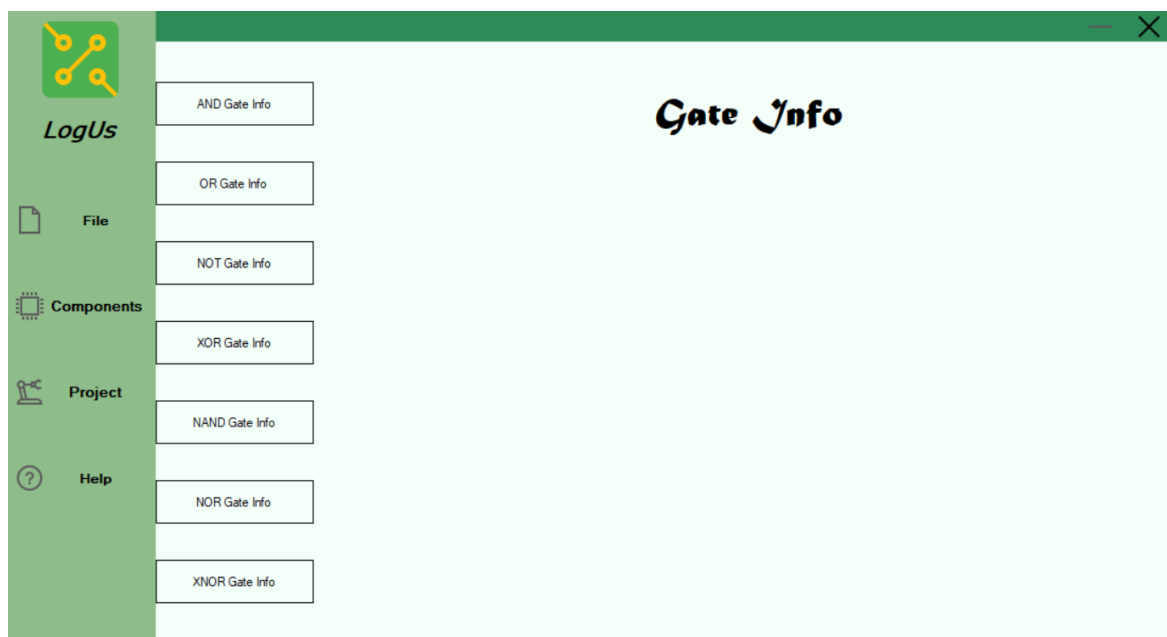
## 2.3 Click chọn Project : Quay lại project đang chạy



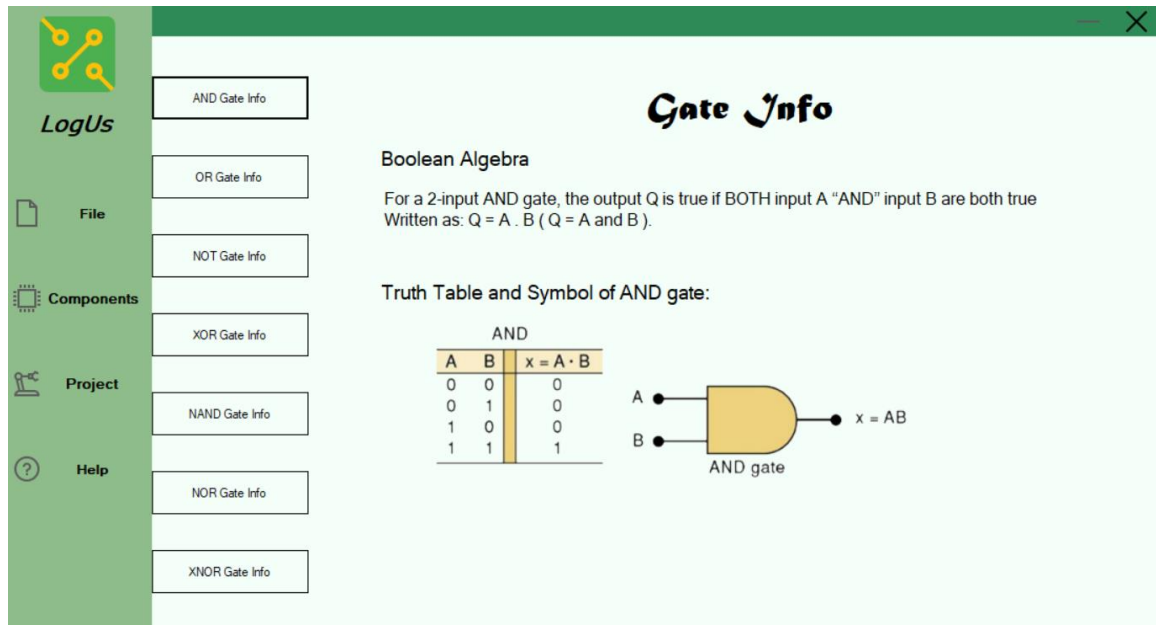
## 2.5 Click chọn help : hiển thị thông tin trợ giúp người dùng



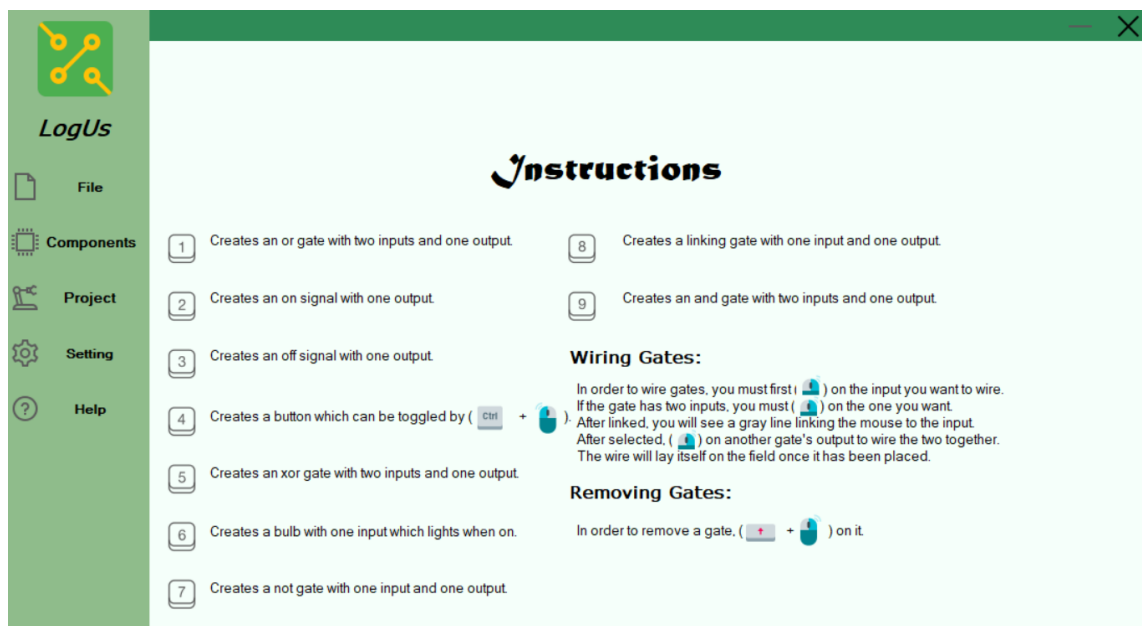
**2.5.1 Click chọn Gates Info :** hiển thị thông tin của cổng logic khi người dùng click chọn ngẫu nhiên một cổng



**Vd :** người dùng click chọn cổng AND



## 2.5.2 Click chọn Instruction : hiển thị thông tin hướng dẫn sử dụng phần mềm



## 2.5.3 Click chọn about : hiển thị thông tin về phần mềm



## V. KẾT LUẬN

### 1. Kết quả đạt được

- Hoàn tất tất cả các mục của phần mềm bao gồm: có hình ảnh mô phỏng lại từng thao tác xử lý, bên cạnh đó còn có phần xuất công thức của mạch sau khi được thiết kế nhằm giúp người dùng có thể thuận tiện hơn trong việc ghi lại công thức của mạch.
- Phân chia công việc giữa các thành viên rõ ràng đạt hiệu quả tốt .

### 2. Hạn chế

- Giao diện đồ họa chưa thực sự ấn tượng.
- Các tính năng chưa được cải tiến ở mức tốt nhất, chưa thực sự hoàn hảo như ý muốn.

### 3. Hướng phát triển

- Cải tiến giao diện cho bắt mắt hơn

- Cải tiến một số thuật toán, code để có thể xuất công thức đã được rút gọn từ mạch vừa thiết kế ( dùng bìa K ) đồng thời thể hiện được các kí hiệu toán học trên màn hình xuất công thức
- Không giới hạn số lượng input, output
- Cập nhập thêm chức năng làm việc với các linh kiện thực tế

## **VI. TÀI LIỆU THAM KHẢO**

- ✚ Slide môn học Nhập Môn Mạch Số ( trường đại học CNTT – DHQG TP.HCM )
- ✚ Slide môn học Lập Trình Trực Quan ( trường đại học CNTT – DHQG TP.HCM )