# BÀI 3: LẬP TRÌNH ĐIỀU KHIỂN MẠCH TUẦN TỰ

## Mã bài: MĐ ĐTCN 25 - 03

***Giới thiệu:***

Hiện nay các mạch tuần tự có thể lập trình được. Do đó, người học phải được trang bị những kiến thức cơ bản về lập trình điều khiển các mạch tuần tự như: mạch Flip-Flop RS, mạch Flip-Flop JK, mạch Flip-Flop D, mạch Flip-Flop T…

**Mục tiêu:**

**Sau khi học xong bài học này người học có khả năng:**

**+ *Kiến thức:***

- Phân tích được cấu trúc của các mạch điều khiển mạch tuần tự;
- Trình bày được trình tự thực hiện lập trình các mạch điều khiển mạch tuần tự;
- Phân tích được những lỗi thường gặp, nguyên nhân và biện pháp xử lý, phòng tránh.

**+ *Kỹ năng:***

- Lập trình được các mạch điều khiển mạch tuần tự theo yêu cầu kỹ thuật;
- Phòng tránh và sửa được các lỗi khi lập trình điều khiển;
- Kiểm tra chính xác điều kiện hoạt động của thiết bị.

**+ *Năng lực tự chủ và trách nhiệm:***

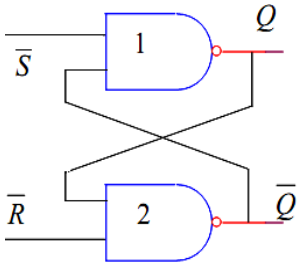
- Rèn luyện tính tỷ mỉ, chính xác và an toàn vệ sinh công nghiệp;
- Tự chịu trách nhiệm khi thực hiện các việc được giao.

**Nội dung**:

### 1. Lập trình mạch Flip-Flop RS

#### *1.1. Flip-Flop RS*

##### *1.1.1. FF R-S sử dụng cổng NAND*




| $\overline{S}$ | $\overline{R}$ | Q |
|---|---|---|
| 0 | 0 | X (cấm) |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | $Q^0$ |

Hình 3.1: Sơ đồ mạch và bảng trạng thái FF R-S sử dụng cổng NAND

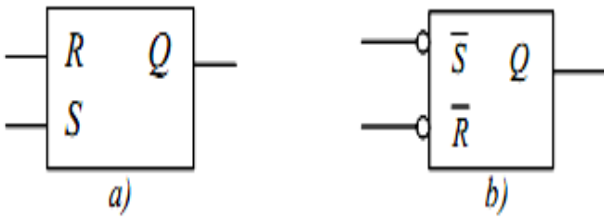
- Dựa vào bảng trạng thái của cổng NAND, ta có:

*+ $\overline{S}$ =0, $\overline{R}$ = 1 ⇒ Q=1. Khi Q=1 hồi tiếp về cổng NAND 2 nên cổng NAND 2 có 2 ngõ vào bằng 1, vậy $\overline{Q}$ = 0.*

*+ $\overline{S}$ =0, $\overline{R}$ = 1 ⇒ $\overline{Q}$ =1. Khi $\overline{Q}$ =1 hồi tiếp về cổng NAND 1 nên cổng NAND 1 có 2 ngõ vào bằng 1, vậy Q= 0.*

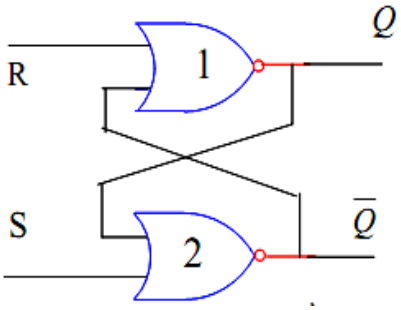
*+ $\overline{S}$ = $\overline{R}$ =0 ⇒ $\overline{Q}$ = Q =1 đây là trạng thái cấm.*

+ $\overline{S}$ = $\overline{R}$ =1, Giả sử trạng thái trước đó có Q =1, $\overline{Q}$ = 0 ⇒ hồi tiếp về cổng NAND 1 nên cổng NAND 1 có một ngõ vào bằng 0, vậy Q = 1 ⇒ FF R-S giữ nguyên trạng thái cũ. Như vậy gọi là FF không đồng bộ bởi vì chỉ cần một trong hai ngõ vào S hay R thay đổi thì ngõ ra cũng thay đổi theo. Về mặt kí hiệu, các FF R-S không đồng bộ được kí hiệu như hình 3.2:




Hình 3.2 : a>. R,S tác động mức 1 – b>. R,S tác động mức 0

### *1.1.2 Mạch FF R-S sử dụng cổng NOR*




| S | R | Q |
|---|---|---|
| 0 | 0 | $Q^0$ |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | X |

*Hình 3.3: FF R-S không đồng bộ sử dụng cổng NOR và bảng trạng thái.*

- Dựa vào bảng trạng thái của cổng NOR, ta có:

*+ S=0, R= 1 ⇒ Q = 0. Khi Q=0 hồi tiếp về cổng NOR 2 nên cổng NOR 2 có 2 ngõ vào bằng 0 ⇒ $\overline{Q}$ = 1. Vậy Q= 0 và $\overline{Q}$ = 1.*

*+ S=0, R= 1 ⇒ $\overline{Q}$ = 0. Khi $\overline{Q}$ = 0 hồi tiếp về cổng NOR 1 nên cổng NOR 1 có 2 ngõ vào bằng 0 ⇒ Q= 1. Vậy Q= 1và $\overline{Q}$ = 0.*

+ Giả sử trạng thái trước đó có S =0, R = 1 ⇒ Q =0, $\overline{Q}$ = 1.

✓ Nếu tín hiệu ngõ vào thay đổi thành : S = 0, R = 0 ( R chuyển từ 1→ 0 ) ta có :

- S =0 và Q = 0 $\Rightarrow \overline{Q} = 1$.
- R = 0 và $\overline{Q} = 1 \Rightarrow Q = 0 \Rightarrow$ *FF R-S giữ nguyên trạng thái trước đó.*

+ Giả sử trạng thái trước đó có S = 1, R = 0 $\Rightarrow$ Q = 1, $\overline{Q}$ = 0.

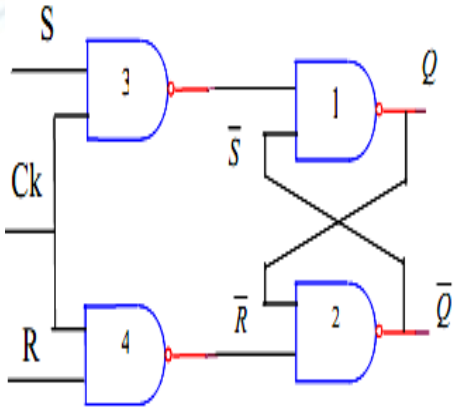
✓ Nếu tín hiệu ngõ vào thay đổi thành : R = 0, S = 0 ( S chuyển từ 1 → 0 ) ta có :

- R =0 và $\overline{Q}$Q = 0 $\Rightarrow$ Q = *1*.
- S= 0 và $Q = 1 \Rightarrow \overline{Q} = 0 \Rightarrow$ *FF R-S giữ nguyên trạng thái trước đó.*

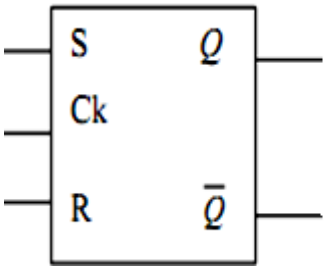
### *1.1.3. FF R-S tác động theo xung lệnh*

Xét sơ đồ FF R-S đồng bộ với sơ đồ mạch, ký hiệu và bảng trạng thái hoạt động như hình 3.4a,b.

Trong đó : Ck là tín hiệu điều khiển đồng bộ hay tín hiệu xung Clock ( tín hiệu xung đồng hồ).




Hình 3.4a: Sơ đồ logic của FF R-S tác động theo xung lệnh




| S | R | Ck | Q |
|---|---|---|---|
| X | X | 0 | $\overline{Q}$ |
| 0 | 0 | 1 | $\overline{Q}$ |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | X |

*Hình 3.4b : Ký hiệu và bảng trạng thái của FF R-S tác động theo xung lệnh*

- CK = 0: cổng NAND 3 và 4 khóa không cho dữ liệu đưa vào, vì cổng NAND 3 và 4 đều có ít nhất một ngõ vào CK = 0 $\Rightarrow \overline{S} = \overline{R} = 1 \Rightarrow$ Q = $\overline{Q}$*: FF R-S giữ nguyên trạng thái cũ.*

- *CK =1:* cổng NAND 3 và 4 mở. Ngõ ra Q sẽ thay đổi tùy thuộc vào trạng thái của S và R.

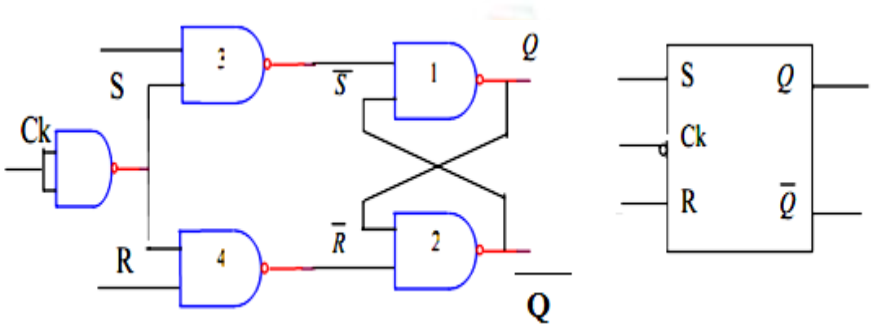
+ S= 0, R = 0 $\Rightarrow \overline{S} = 1, \overline{R} = 1 \Rightarrow \overline{Q} = Q$

+ S= 0, R = 1 $\Rightarrow \overline{S} = 1, \overline{R} = 0 \Rightarrow Q = 0$

+ S= 1, R = 0 $\Rightarrow \overline{S} = 0, \overline{R} = 1 \Rightarrow Q = 1$

+ S= 1, R = 1 $\Rightarrow \overline{S} = 0, \overline{R} = 0 \Rightarrow Q = X$

Trong trường hợp này tín hiệu đồng bộ Ck tác động mức 1, nếu tín hiệu Ck tác động mức 0 ta mắc thêm cổng đảo như hình 3.5




Hình 3.5: Sơ đồ logic và ký hiệu FF R-S của mức 0

❖ ***Xung Clock và các tác động của xung Clock***

Theo trên ta thấy các ngõ ra của FF chỉ thay đổi khi C = 1. Tuy nhiên sự thay đổi ở ngõ vào là liên tục thì không thể xác định trạng thái ngõ ra tại thời điểm bất kỳ. Để tránh điều này này lệnh C được thay bằng các xung điện tuần tự theo thời gian và mỗi khi xuất hiện một xung ngõ ra của các FF thay đổi trạng thái một lần.

Các xung điện như vậy gọi là xung nhịp hay xung đồng hồ ký hiệu là $C_K$. Xung Clock thường là một chuỗi xung hình chữ nhật hoặc sóng hình vuông. Xung Clock được phân phối đến tất cả các bộ phận của hệ thống. Và hầu hết ngõ ra của hệ thống chỉ thay đổi trạng thái khi có một xung Clock thực hiện một bước chuyển tiếp.

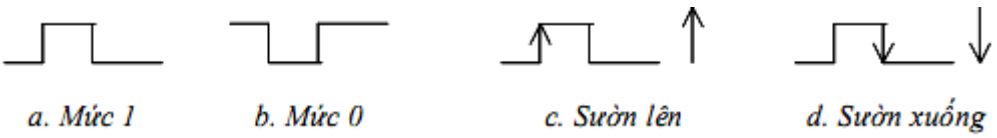
Tùy thuộc vào mức tích cực của tín hiệu đồng bộ $C_k$ , chúng ta có các loại tín hiệu điều khiển như hình 3.6.

+ $C_k$ điều khiển theo mức 1

+ $C_k$ điều khiển theo mức 0

+ $C_k$ điều khiển theo sườn lên (sườn trước)

+ $C_k$ điều khiển theo sườn xuống (sườn sau)




Hình 3.6: Các loại tín hiệu điều khiển của $C_k$

### *1.2. Trình tự thực hiện*

***Lập trình FF R-S sử dụng cổng NAND ( R, S tác động mức 0)***

***Với*** $\overline{S}$ ***=SB,*** $\overline{R}$ ***= RB, Q = Q ,*** $\overline{Q}$ ***= QB***

***Các tín hiệu đầu vào SB, RB kết nối với V16, V17 của board mạch Basys 3***

***Các tín hiệu đầu ra Q, QB kết nối với U16, E19 của board mạch Basys 3***

#### *1.2.1. Điều kiện thực hiện*

- **Chuẩn bị thiết bị, dụng cụ, vật tư.**

| THIẾT BỊ, DỤNG CỤ | VẬT TƯ |
|---|---|
| - Máy tính có cài phần mềm Vivado<br>- BASYS 3 Hãng XILINX | - Dây kết nối máy tính với BASYS 3 |

- **An toàn lao động.**

+ Đảm bảo an toàn điện.

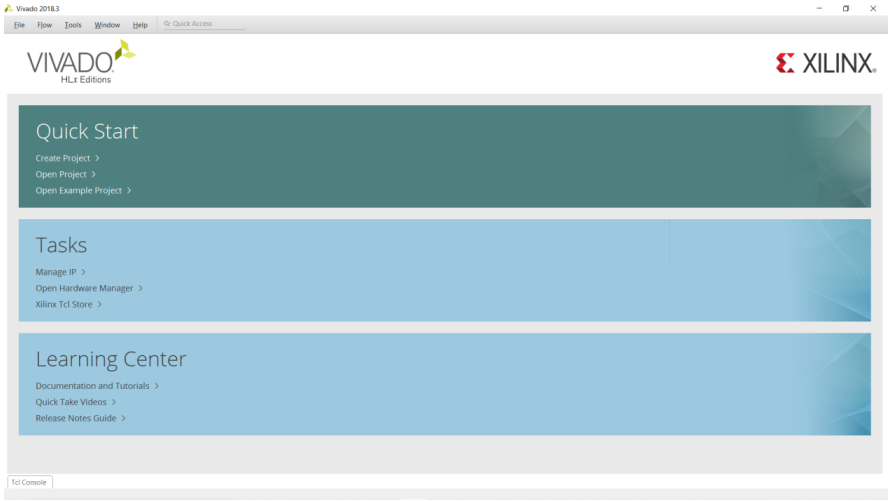
+ Đảm bảo an toàn cho người và thiết bị, dụng cụ.

#### *1.2.2. Các bước thực hiện*

**Bước 1: Khởi động phần mềm Vivado**

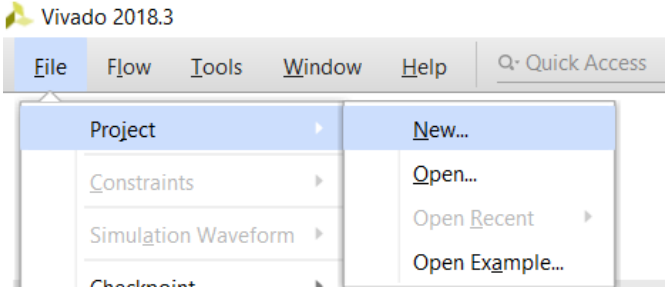
Kích vào biểu tượng 

trên màn hình Desktop ta có giao diện phần mềm như sau

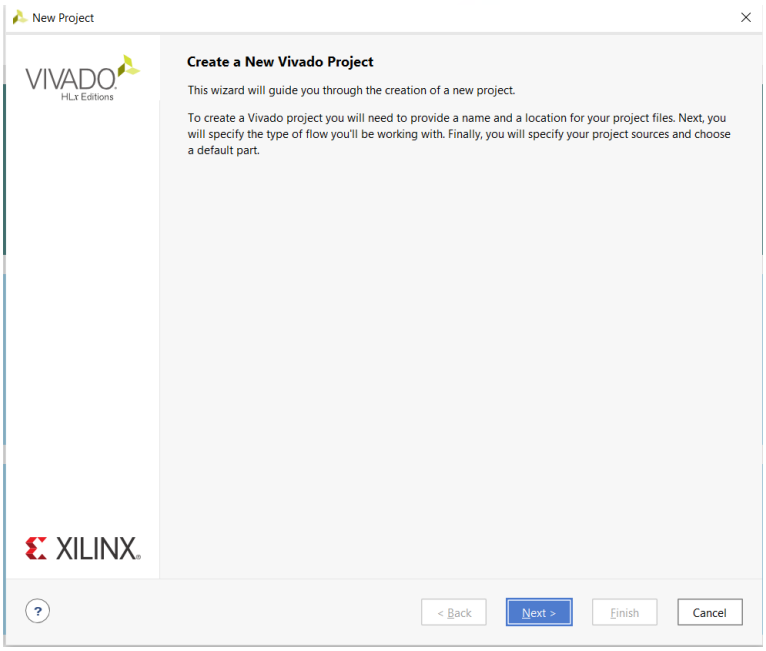


**Bước 2: Tạo project mới và đặt tên cho project là *FF_RS_NAND***

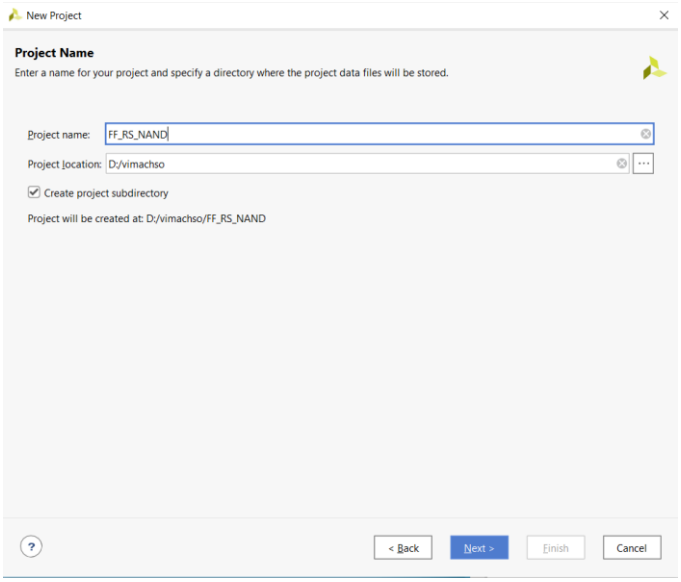
Vào File chọn Project sau đó chọn New




Hiển thị ra hộp thoại



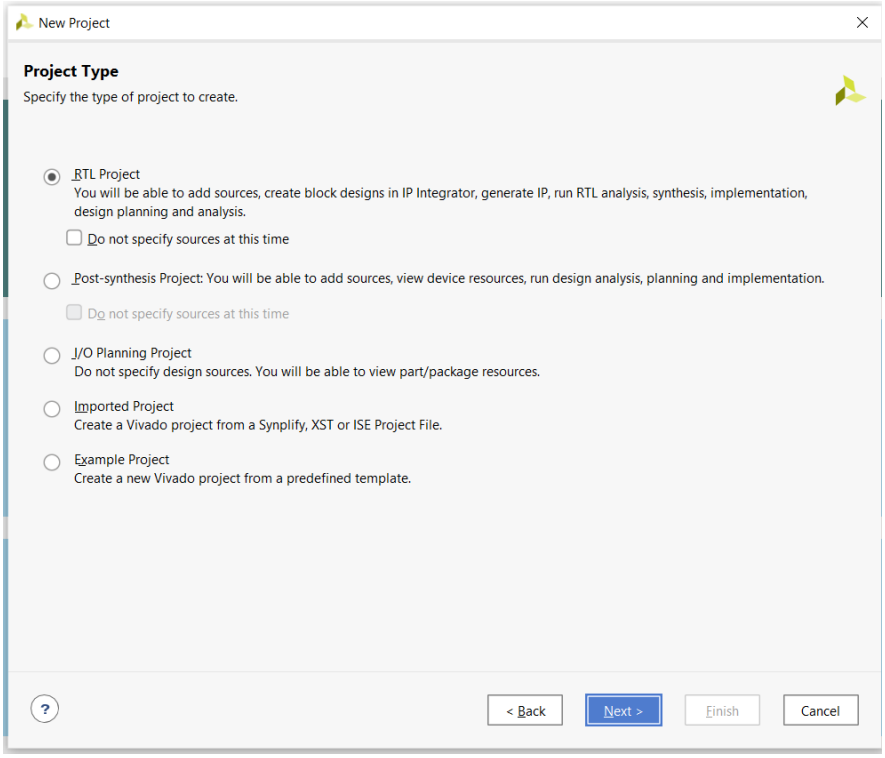

Nhấn Next hiển thị ra Hộp thoại



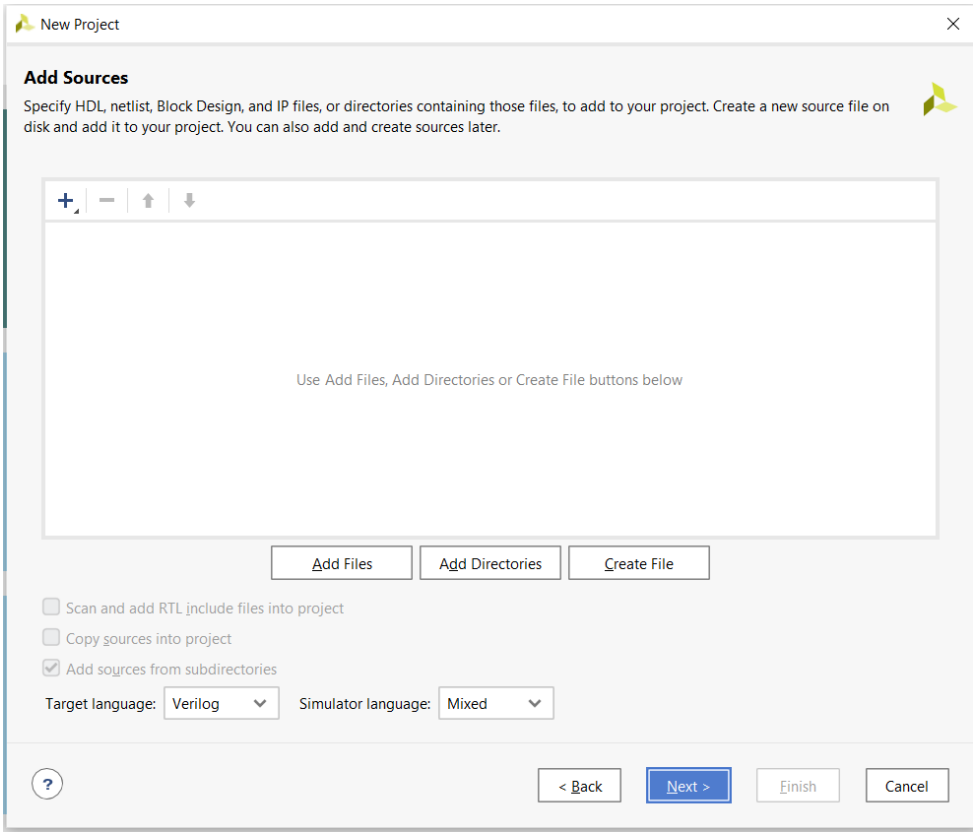

Project name ta đặt tên là ***FF_RS_NAND***

Project location chọn đường dẫn tới D:/vimachso sau đó nhấn Next

Hiển thị ra hộp thoại




Nhấn Next hiển thị ra hộp thoại

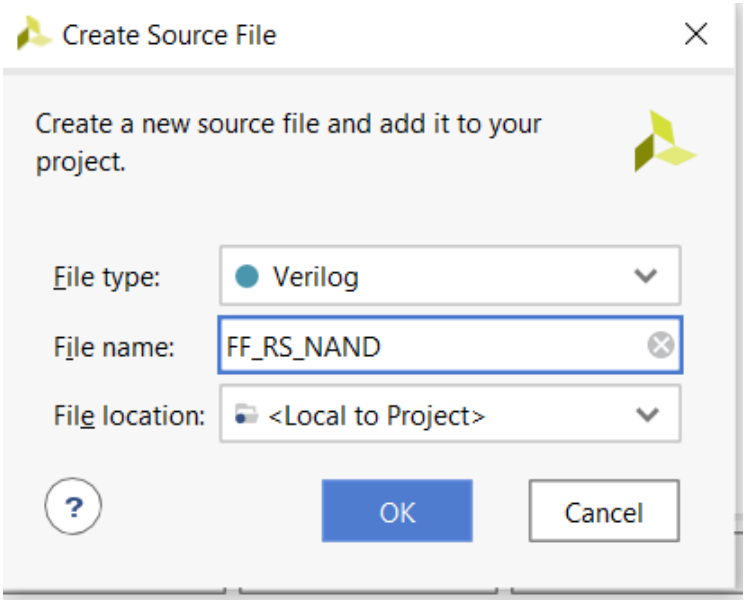


**Bước 3: Tạo file *FF_RS_NAND*.v và *FF_RS_NAND*.xdc**

Target langguage ta chọn Verilog

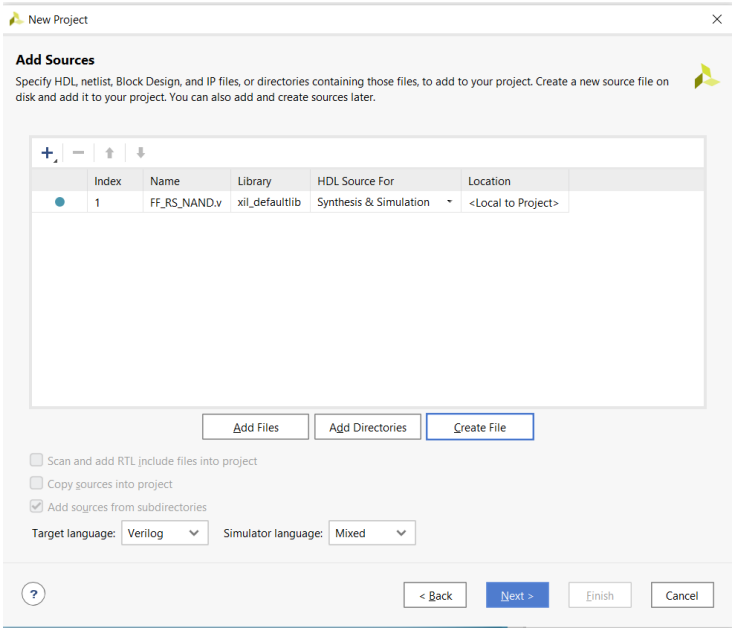
Simulator langguage ta chọn Mixed

Nhấn vào Create file hiển thị ra hộp thoại

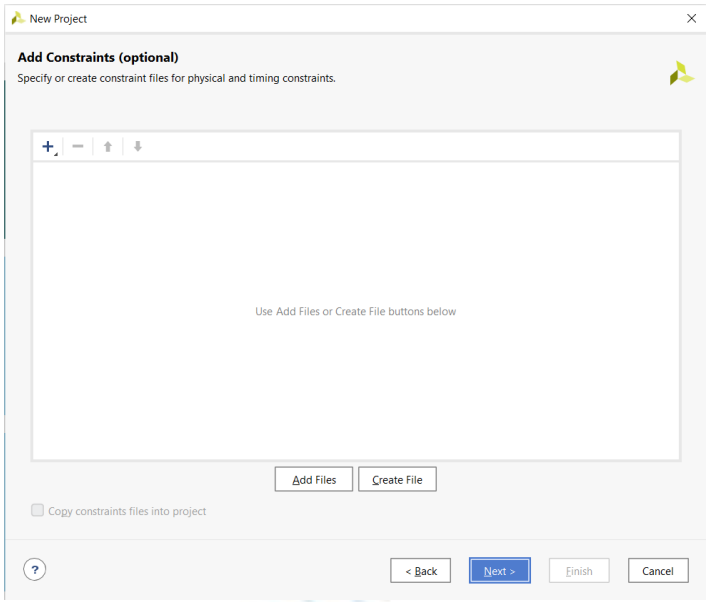



File name ta gõ ***FF_RS_NAND*** sau đó nhấn OK

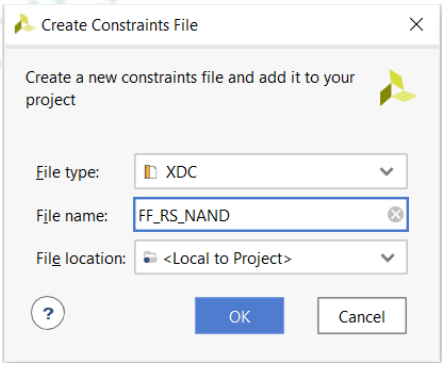
Hiển thị ra hộp thoại



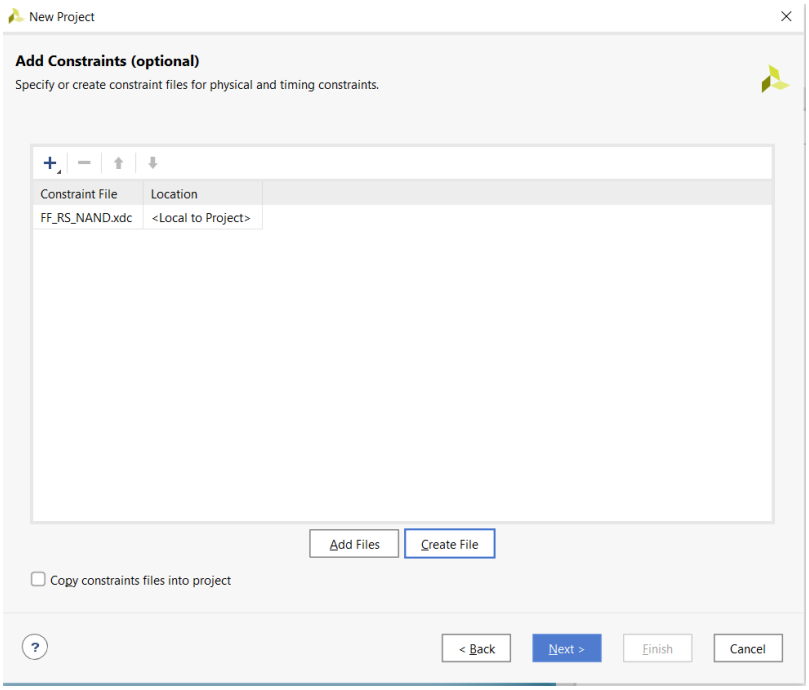

Nhấn Next hiển thị ra hộp thoại



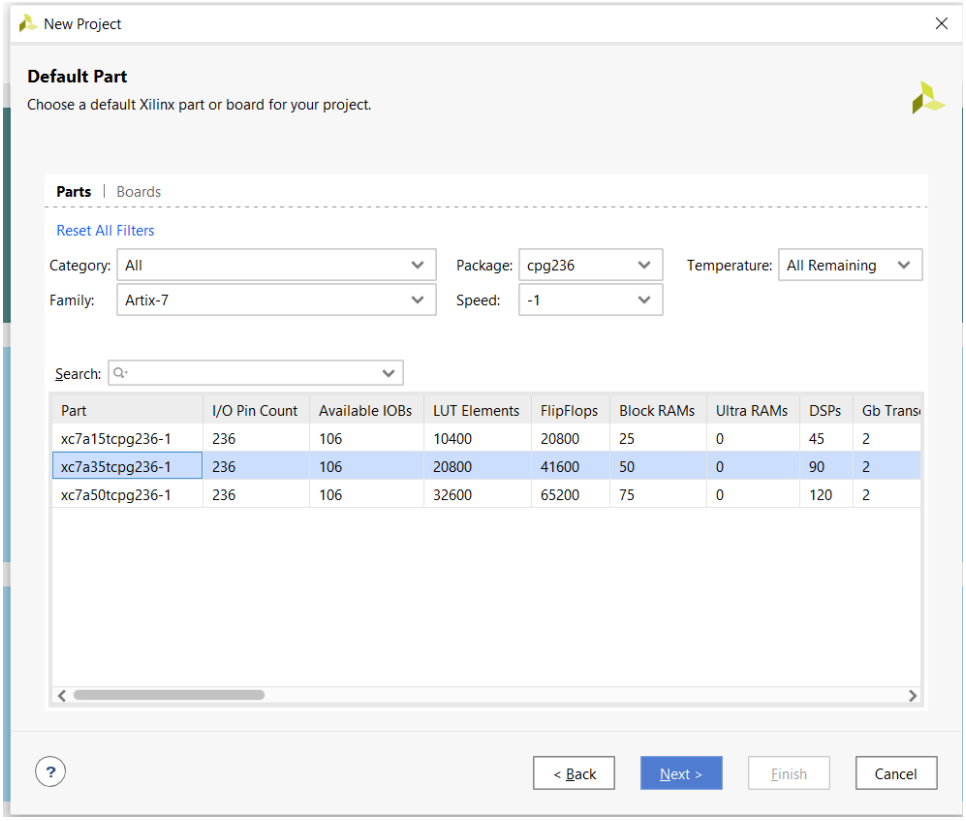

Nhấn vào Create file hiển thị ra hộp thoại




File name ta gõ ***FF_RS_NAND*** sau đó nhấn OK

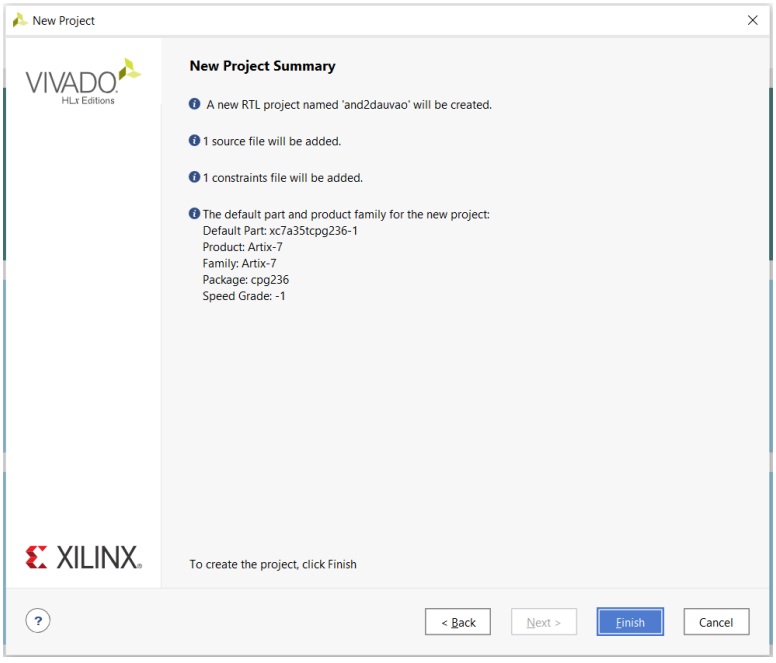



Nhấn Next hiển thị ra hộp thoại

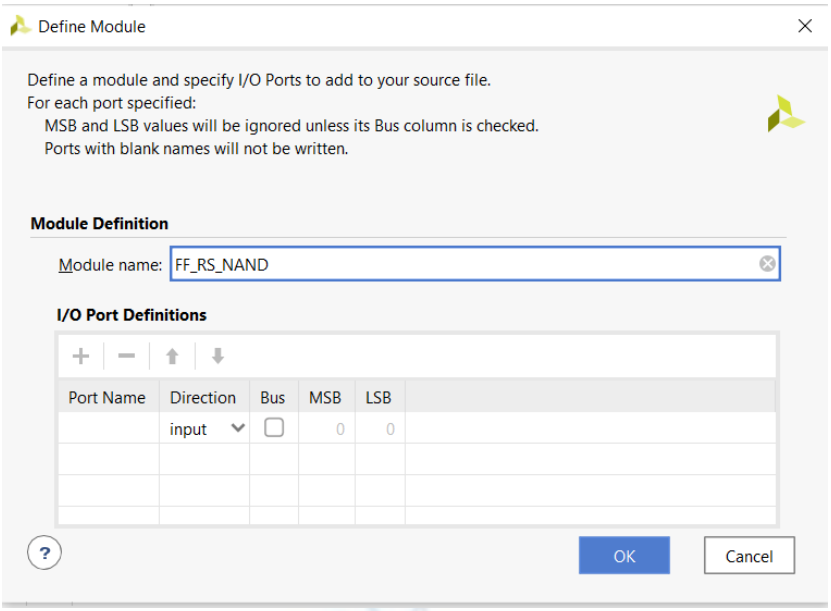



**Bước 4: Chọn chọn dòng IC lập trình**

Family ta chọn Artix-7, Package chọn cpg236, Speed chọn -1 sau đó ta chọn chíp xc7a35tcpg236-1 sau đó nhấn Next sẽ hiển thị ra hộp thoại




Nhấn Finish hiển thị ra hộp thoại




**Bước 5: Ghi các thông tin đầu vào và đầu ra**

Theo bài toán lập trình FF R-S sử dụng cổng NAND ( R, S tác động mức 0)

| Port Name | Direction | Bus | MSB | LSB |
|---|---|---|---|---|
| SB | input | ☐ | 0 | 0 |
| RB | input | ☐ | 0 | 0 |
| Q | inout | ☐ | 0 | 0 |
| QB | inout | ☐ | 0 | 0 |

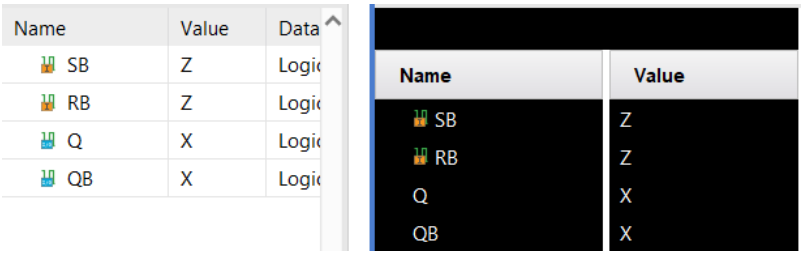
Sau đó nhấn OK

**Bước 6: Viết Chương trình trong file *FF_RS_NAND*.v**

```
`timescale 1ns / 1ps
module FF_RS_NAND(
    input SB,
    input RB,
    inout Q,
    inout QB
    );
    assign Q = ~(SB & QB);
    assign QB = ~(RB & Q);
endmodule
```

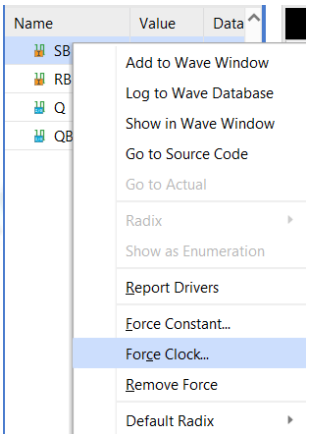
**Bước 7: Mô phỏng dạng sóng trên phần mềm**

Click vào Run Simulation Chọn Run Behavioral Simulation

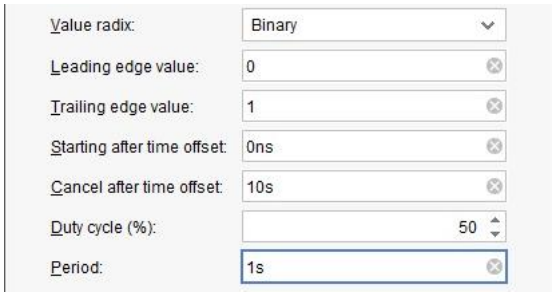



Để mô phỏng dạng sóng trên phần mềm, ta đặt các thông số cho các đầu vào và quan sát dạng sóng đầu ra

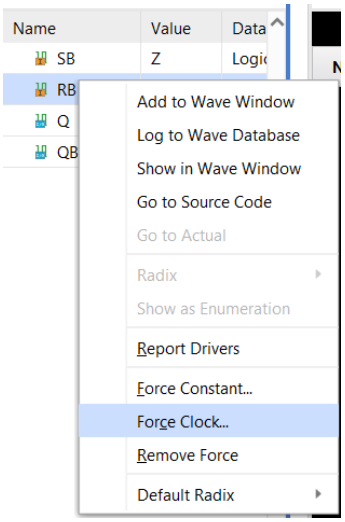
❖ Đặt thống số cho đầu vào SB, ta click chuột phải vào SB



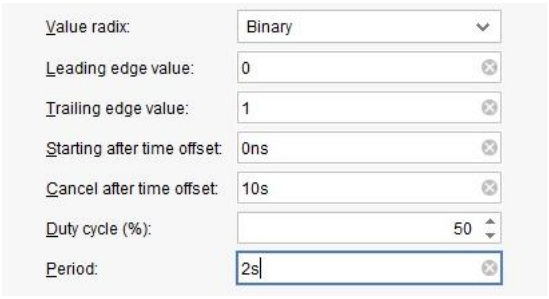

Chọn Force Clock

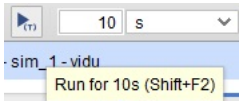



❖ Đặt thống số cho đầu vào SB, ta click chuột phải vào SB

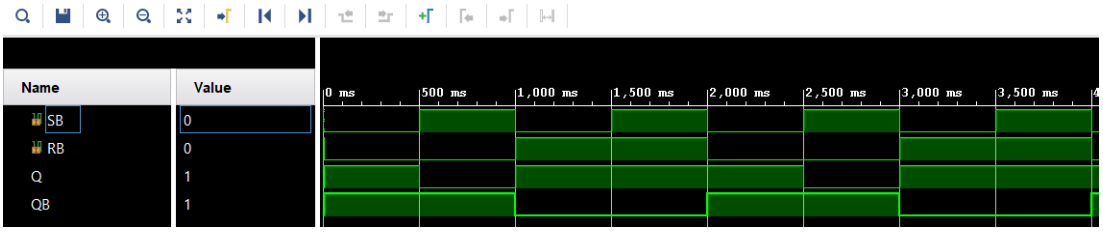


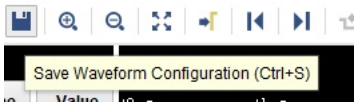
Chọn Force Clock




Nhấn vào biểu tượng 

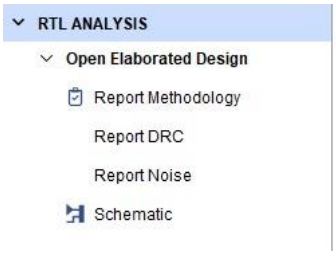
để mô phỏng dạng sóng trên phần mềm



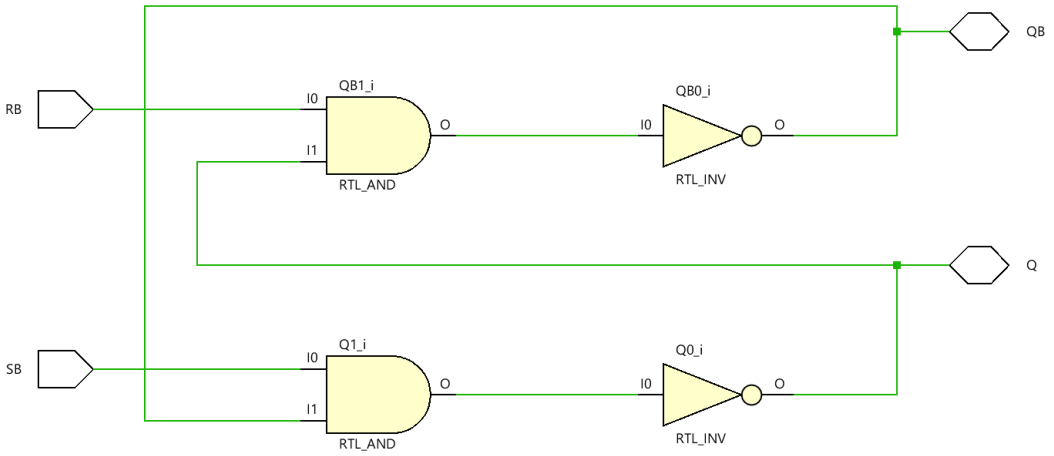

Nhấn vào biểu tượng 

để lưu dạng sóng

**Bước 8: Gán chân thích ứng với thiết bị**



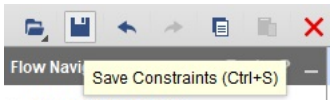

Trong phần RTL ANALYSIS ta click vào Schematic



Click vào I/O Ports để gán chân linh kiện phù hợp với thiết bị. Trong bài toán này ta Click vào 4 I/O Ports

Ta đặt các đầu vào và đầu ra tương ứng với các chân của Chip trong cột Package Pin và dạng dữ liệu I/O Std như sau:

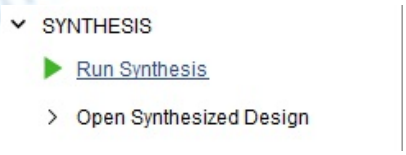
| Name | Direction | Interface | Neg Diff Pair | Package Pin | Fixed | Bank | I/O Std |
|---|---|---|---|---|---|---|---|
| SB | IN | | | V16 | ✓ | 14 | LVCMOS33* |
| RB | IN | | | V17 | ✓ | 14 | LVCMOS33* |
| QB | INOUT | | | U16 | ✓ | 14 | LVCMOS33* |
| Q | INOUT | | | E19 | ✓ | 14 | LVCMOS33* |

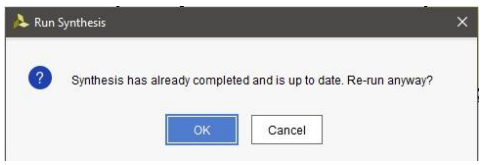
Nhấn vào 

để lưu toàn bộ dữ liệu

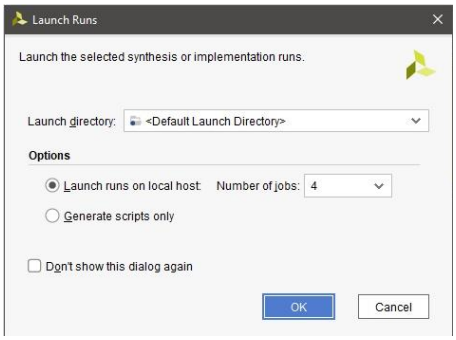
**Bước 9: Biên dịch tổng hợp chương trình**

Trong mục SYNTHESIS ta click vào Run Synthesis

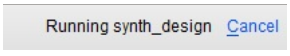


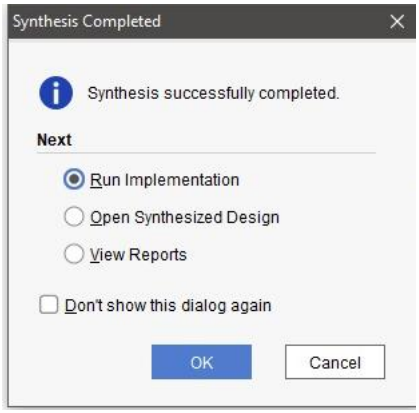

Nhấn OK



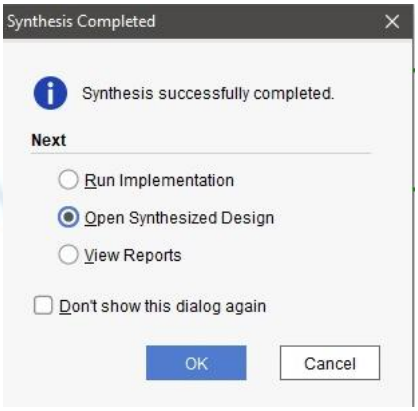

Chọn Launch runs on local host number of jobs 4 sau đó nhấn OK

Quan sát quá trình chạy 

đến khi hoàn thành



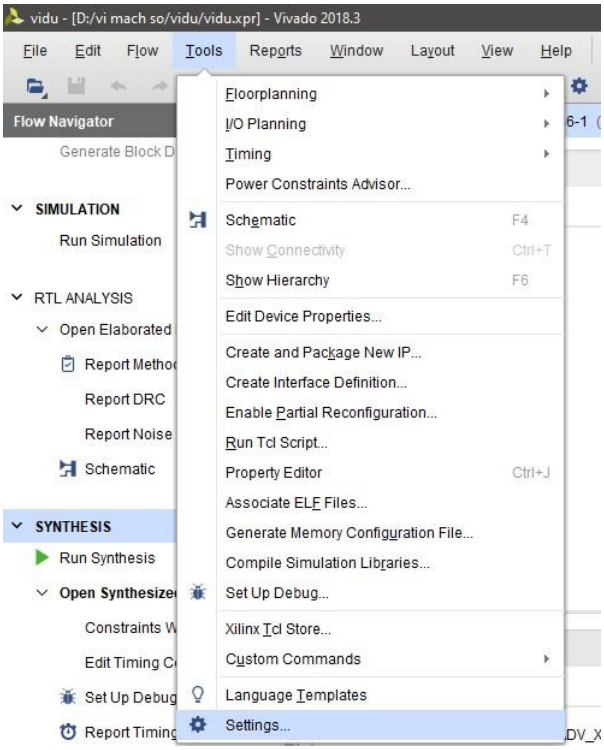

Click vào Open Synthesized Design để quan sát thiết kế



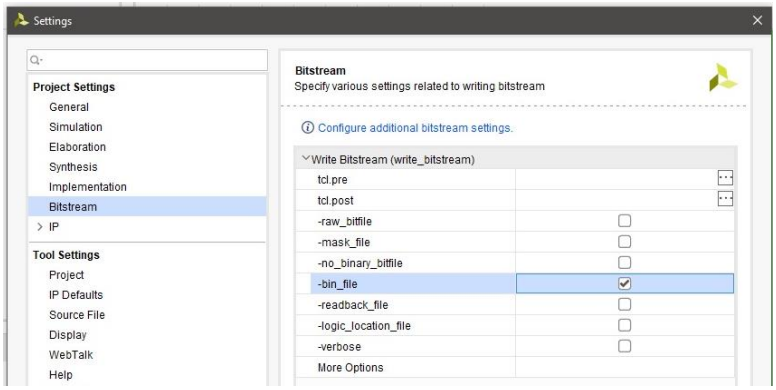

Nhấn OK

**Bước 10: Download chương trình vào phần cứng của thiết bị**

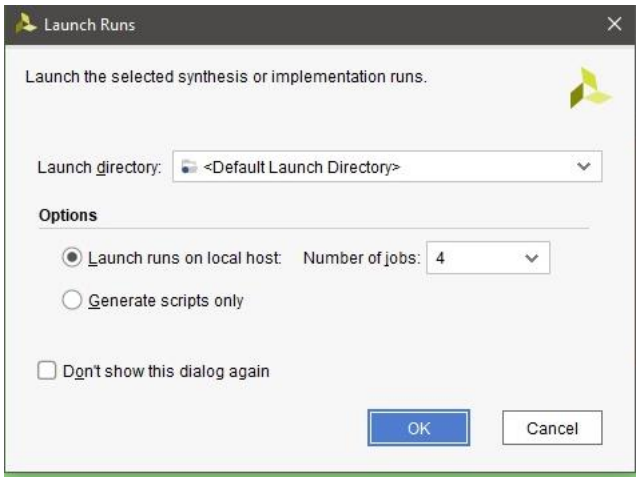
Để Download chương trình vào phần cứng của thiết bị ta vào Tool chọn Settings…



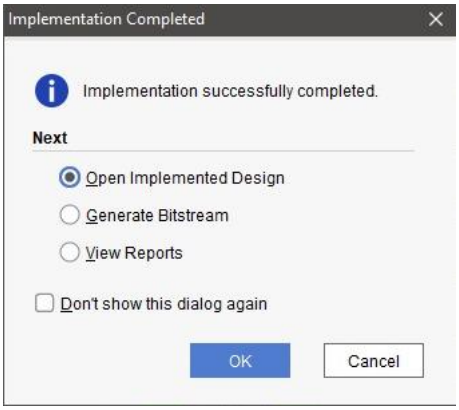
Click vào Bitstream, chọn dạng xuất dữ liệu là -bin_file



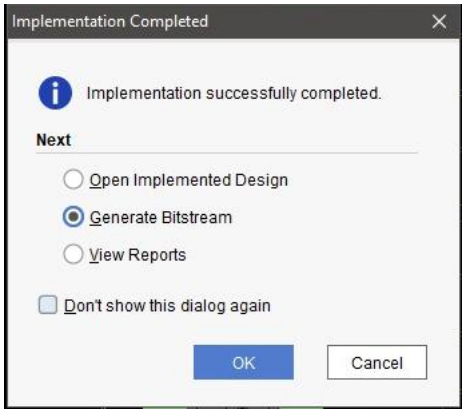

Trong phần IMPLEMENTATION ta click vào Run Implementation

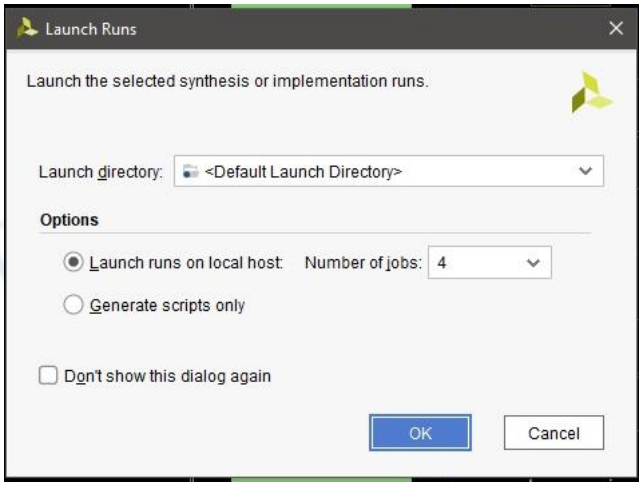

Chọn Launch runs on local host Number or jobs 4 sau đó nhấn OK




Sau khi chạy thành công ta chọn Generate Bitstream

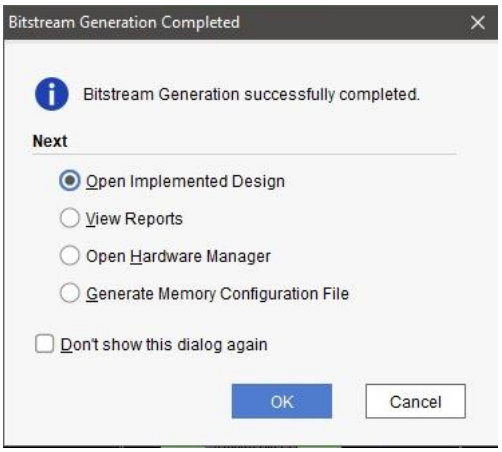



Nhấn OK

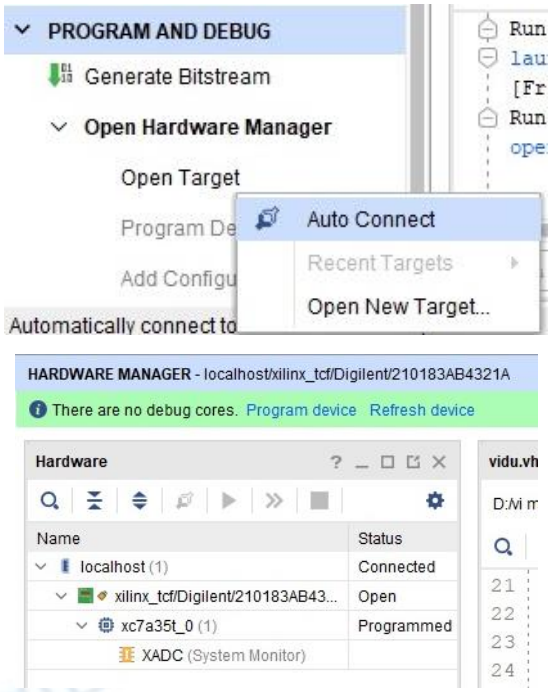



Chọn Launch runs on local host Number or jobs 4 sau đó nhấn OK

Đợi quá trình hoàn thành 

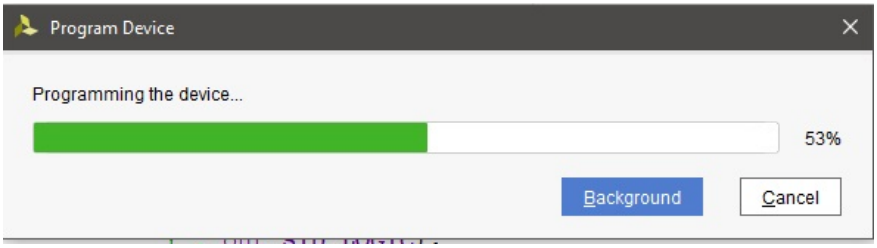




Để đổ chương trình vào phần cứng của thiết bị ta cắm thiết bị vào cổng USB của máy tính sau đó trong phần PROGRAM AND DEBUG ta click vào Open Target chọn Auto Connect



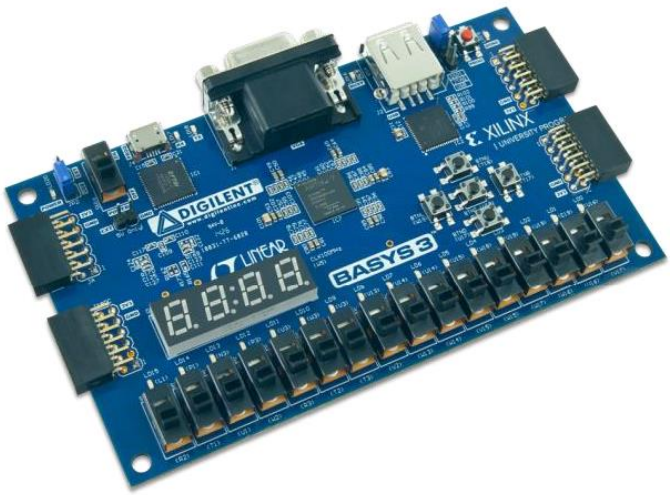

Click vào Program device

Chọn đường dẫn đến ***FF_RS_NAND***.bit sau đó nhấn vào Program




Quá trình ghi dữ liệu đến khi hoàn thành 100% là xong.

**Bước 11: Chạy chương trình trên thiết bị**




Gạt các công tắc V16 và V17 theo bảng trạng thái ***FF_RS_NAND*** và quan sát đèn U16, E19

*1.3. Một số lỗi thường gặp, nguyên nhân và biện pháp xử lý, phòng tránh.*

| TT | Sai hỏng | Nguyên nhân | Biện pháp xử lý, phòng tránh |
|---|---|---|---|
| 1 | Sai chân đầu vào/đầu ra | Gán sai chân đầu vào/đầu ra trên thiết bị | Xác định đúng chân đầu vào/đầu ra trên thiết bị.<br>Gán đúng chân đầu vào/đầu ra trên thiết bị |
| 2 | Sai lệnh cấu trúc lệnh | Lập trình sai cấu trúc | Cần lập trình đúng theo cấu trúc lệnh:<br>ví dụ:<br>assign Q = ~(SB & QB);<br>assign QB = ~(RB & Q); |
| 3 | Không download được chương trình xuống thiết bị | Do kết nối thiết bị với máy tính | Kiểm tra kết nối giữa thiết bị với máy tính |

*1.4. Bài tập áp dụng*

**Bài 1:**

Lập trình FF R-S sử dụng cổng NAND ( R, S tác động mức 0)

Với $\overline{S}$ =SB, $\overline{R}$ = RB, Q = Q , $\overline{Q}$ = QB

Các tín hiệu đầu vào SB, RB kết nối với V16, V17 của board mạch Basys 3

Các tín hiệu đầu ra Q, QB kết nối với U16, E19 của board mạch Basys 3

**Bài 2:**

Lập trình FF R-S sử dụng cổng NOR

Với Q = Q , $\overline{Q}$ = QB

Các tín hiệu đầu vào S, R kết nối với V16, V17 của board mạch Basys 3

Các tín hiệu đầu ra Q, Q kết nối với U16, E19 của board mạch Basys 3

Gợi ý: Để thực hiện lậm trình FF R-S sử dụng cổng NOR ta viết các hàm đầu ra theo cấu trúc lệnh:

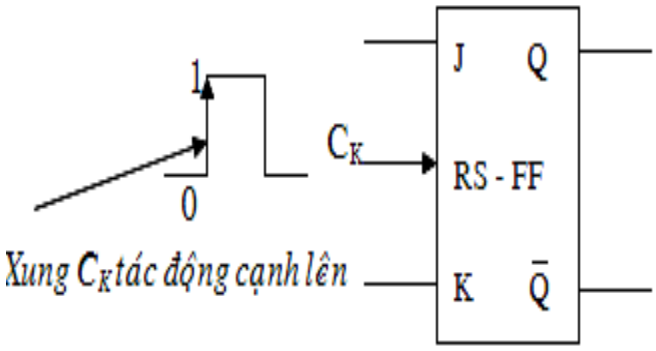
assign Q = ~(R + QB);

assign QB = ~(S + Q);

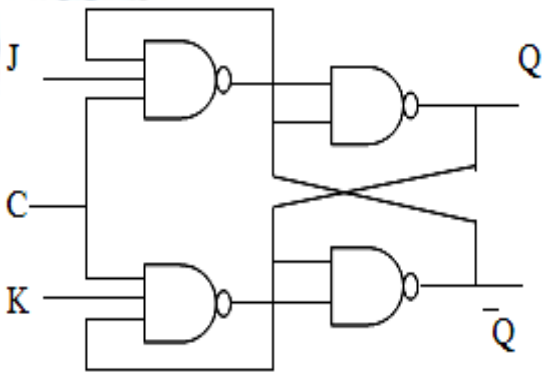
## 2. Lập trình mạch Flip-Flop JK

### *2.1. Flip-Flop JK*

#### *2.1.1.Cấu trúc mạch logic*




Hình 3.7a: Ký hiệu FF J-K




Hình 3.7b: *Cấu trúc mạch logic FF* J –K

#### *2.1.2.Bảng trạng thái FF J-K*

| $C_k$ | J | K | $Q^K$ |
|---|---|---|---|
| ↑ | 0 | 0 | Q ( nhớ) |
| ↑ | 0 | 1 | 0 ( xóa) |
| ↑ | 1 | 0 | 1( lập) |
| ↑ | 1 | 1 | $\overline{Q}$(thay đổi trạng thái theo mỗi xung nhịp) |

Hình 3.8 : Bảng trạng thái FF J-K

Trong đó: