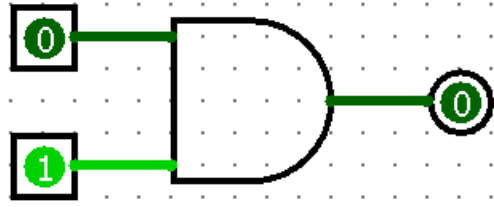
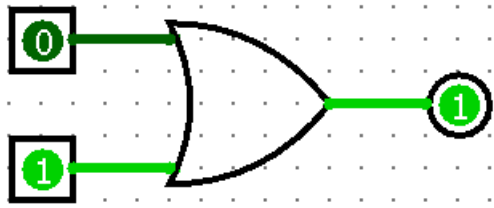


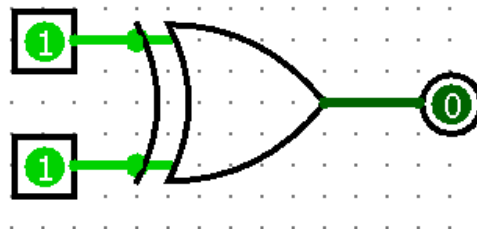
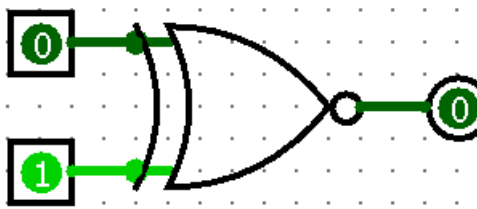
## MỤC LỤC

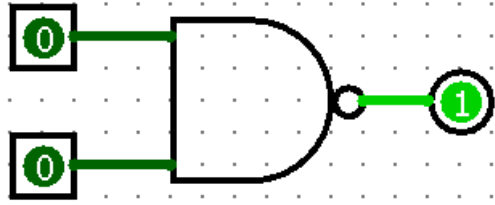
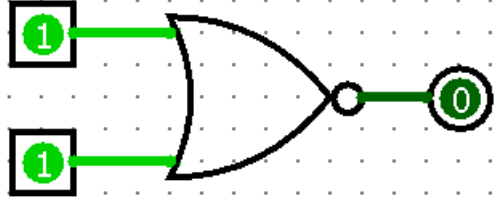
<b>1.1. Mô phỏng các cổng luận lý và thiết bị lưu trữ .....</b>	<b>1</b>
<i>1.1.1. Mô phỏng chức năng của các cổng luận lý .....</i>	<i>1</i>
<i>1.1.2. Mô phỏng các thiết bị lưu trữ .....</i>	<i>4</i>
<b>1.2. Mô phỏng mạch .....</b>	<b>7</b>
<i>1.2.1. Mô phỏng mạch tổ hợp .....</i>	<i>7</i>
<i>1.2.2. Mô phỏng mạch tuần tự .....</i>	<i>9</i>

## 1.1. Mô phỏng các cổng luận lý và thiết bị lưu trữ

### 1.1.1. Mô phỏng chức năng của các cổng luận lý

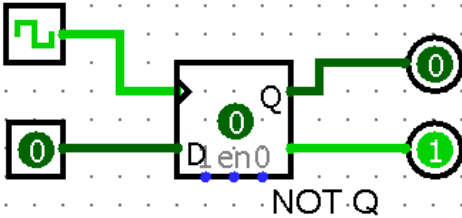
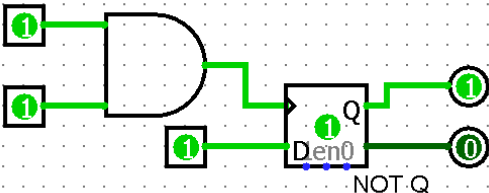
Tên cổng	Chức năng	Tính chất	Biểu thức logic	Ký hiệu	Bảng chân lý																		
AND	Cổng AND là một cổng logic dùng để thực hiện biểu thức logic hàm AND cho hai hay nhiều biến.	<ul style="list-style-type: none"><li>Số ngõ vào tùy thuộc vào số biến và có một ngõ ra là kết quả thực hiện hàm AND của các biến.</li><li>Giá trị của ngõ ra có thể là giá trị thấp/sai/false (0) hoặc cao/đúng/true (1) phụ thuộc vào các giá trị ngõ vào.</li><li>Ngõ ra là 1 khi tất cả giá trị ngõ vào là 1 và 0 khi có một ngõ vào là 0, bất chấp giá trị của các ngõ vào còn lại.</li></ul>	$F = A \cdot B$ $F = A \& B$	 <p>Hình 1: Cổng AND Ký hiệu theo chuẩn ANSI Ngõ vào 0 AND 1, cho ra kết quả 0</p>	<table><tr><th colspan="2">Input</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Input		Output	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
Input		Output																					
A	B	F																					
0	0	0																					
0	1	0																					
1	0	0																					
1	1	1																					
OR	Cổng OR là một cổng logic dùng để thực hiện biểu thức logic hàm OR cho hai hay nhiều biến.	<ul style="list-style-type: none"><li>Số ngõ vào tùy thuộc vào số biến và có một ngõ ra là kết quả thực hiện hàm OR của các biến.</li><li>Giá trị của ngõ ra có thể là giá trị thấp/sai/false (0) hoặc cao/đúng/true (1) phụ thuộc vào các giá trị ngõ vào.</li><li>Ngõ ra là 0 khi tất cả giá trị ngõ vào là 0 và 1 khi có một</li></ul>	$F = A + B$ $F = A \mid B$	 <p>Hình 2: Cổng OR Ký hiệu theo chuẩn ANSI Ngõ vào 0 OR 1, ngõ ra trả kết quả 1</p>	<table><tr><th colspan="2">Input</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Input		Output	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
Input		Output																					
A	B	F																					
0	0	0																					
0	1	1																					
1	0	1																					
1	1	1																					

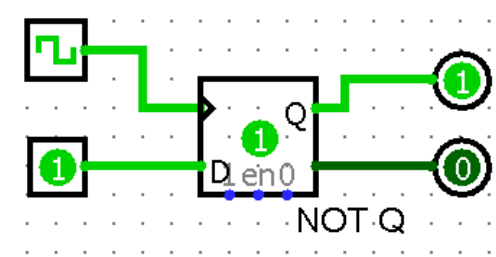
		ngõ vào là 1, bất chấp giá trị của các ngõ vào còn lại.																					
XOR	Cổng XNOR là một cổng logic dùng để thực biểu thức logic hàm XNOR cho hai hay nhiều biến.	<ul style="list-style-type: none"><li>▪ Cổng XOR chỉ có 2 ngõ vào và 1 ngõ ra.</li><li>▪ Giá trị của ngõ ra có thể là giá trị thấp/sai/false (0) hoặc cao/đúng/true (1) phụ thuộc vào các giá trị ngõ vào.</li><li>▪ Ngõ ra là 0 khi tất cả giá trị ngõ vào cùng là 0 hoặc cùng là 1; và là 1 khi có một giá trị ngõ vào khác các giá trị còn lại.</li></ul>	$F = A \oplus B$	<div></div> <p>Hình 3: Cổng XOR Ký hiệu theo chuẩn ANSI Ngõ vào 1 XOR 1, ngõ ra trả kết quả 0</p>	<table><tr><th colspan="2">Input</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Input		Output	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
Input		Output																					
A	B	F																					
0	0	0																					
0	1	1																					
1	0	1																					
1	1	0																					
XNOR	Cổng XNOR là một cổng logic dùng để thực biểu thức logic hàm XNOR cho hai hay nhiều biến.	<ul style="list-style-type: none"><li>▪ Cổng XNOR là sự kết hợp của cổng XOR và NOT.</li><li>▪ Cổng XNOR chỉ có 2 ngõ vào và 1 ngõ ra.</li><li>▪ Giá trị của ngõ ra có thể là giá trị thấp/sai/false (0) hoặc cao/đúng/true (1) phụ thuộc vào các giá trị ngõ vào.</li><li>▪ Ngược lại với cổng XNOR, ngõ ra là 1 khi tất cả giá trị ngõ vào cùng là 0 hoặc cùng là 1; và là 0 khi có một giá trị ngõ vào khác các giá trị còn lại.</li></ul>	$F = A \odot B$	<div></div> <p>Hình 4: Cổng XNOR Ký hiệu theo chuẩn ANSI Ngõ vào 0 XNOR 1, ngõ ra trả kết quả 0</p>	<table><tr><th colspan="2">Input</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	Input		Output	A	B	F	0	0	1	0	1	0	1	0	0	1	1	1
Input		Output																					
A	B	F																					
0	0	1																					
0	1	0																					
1	0	0																					
1	1	1																					

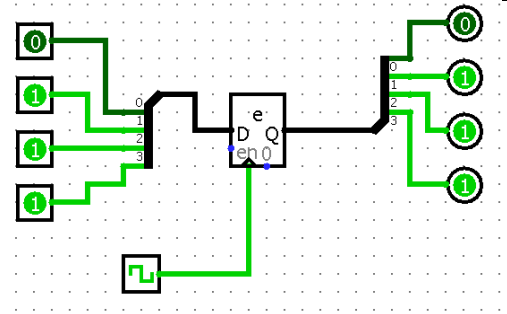
NAND	<p>Cổng NAND là một cổng logic dùng để thực biểu thức logic hàm NAND cho hai hay nhiều biến.</p>	<ul style="list-style-type: none"><li>▪ Cổng NAND là sự kết hợp của cổng AND và NOT.</li><li>▪ Số ngõ vào tùy thuộc vào số biến và có một ngõ ra là kết quả thực hiện hàm OR của các biến.</li><li>▪ Giá trị của ngõ ra có thể là giá trị thấp/sai/false (0) hoặc cao/đúng/true (1) phụ thuộc vào các giá trị ngõ vào.</li><li>▪ Ngược lại với cổng AND, ngõ ra của NAND là 0 khi tất cả giá trị ngõ vào là 1 và 1 khi có một ngõ vào là 0, bất chấp giá trị của các ngõ vào còn lại.</li></ul>	$F = \overline{A \cdot B}$	<div></div> <p>Hình 5: Cổng NAND Ký hiệu theo chuẩn ANSI Ngõ vào 0 NAND 0, ngõ ra trả kết quả 1</p>	<table><tr><th colspan="2">Input</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Input		Output	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
Input		Output																					
A	B	F																					
0	0	1																					
0	1	1																					
1	0	1																					
1	1	0																					
NOR	<p>Cổng NOR là một cổng logic dùng để thực biểu thức logic hàm NOR cho hai hay nhiều biến.</p>	<ul style="list-style-type: none"><li>▪ Cổng NOR là sự kết hợp của cổng OR và NOT.</li><li>▪ Số ngõ vào tùy thuộc vào số biến và có một ngõ ra là kết quả thực hiện hàm OR của các biến.</li><li>▪ Giá trị của ngõ ra có thể là giá trị thấp/sai/false (0) hoặc cao/đúng/true (1) phụ thuộc vào các giá trị ngõ vào.</li><li>▪ Ngược lại với cổng OR, ngõ ra của NOR là 1 khi tất cả giá trị ngõ vào là 0 và 0</li></ul>	$F = \overline{A + B}$	<div></div> <p>Hình 6: Cổng NOR Ký hiệu theo chuẩn ANSI Ngõ vào 1 NOR 1, ngõ ra trả kết quả 0</p>	<table><tr><th colspan="2">Input</th><th>Output</th></tr><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	Input		Output	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
Input		Output																					
A	B	F																					
0	0	1																					
0	1	0																					
1	0	0																					
1	1	0																					

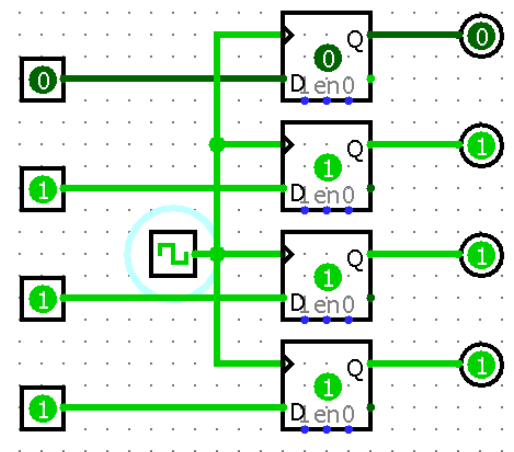
		khi có một ngõ vào là 1, bắt chấp giá trị của các ngõ vào còn lại.			
--	--	--	--	--	--

### 1.1.1. Mô phỏng các thiết bị lưu trữ

Tên thiết bị	Chức năng	Tính chất	Ký hiệu	Bảng chân lý																																														
D Latch	Thiết bị lưu trữ thông tin, dữ liệu, truyền dữ liệu, thiết kế các mạch tuần tự (chủ yếu với dữ liệu thấp).	<ul style="list-style-type: none"><li>Có khả năng lưu trữ 1 bit thông tin.</li><li>Lưu trữ thông tin trạng thái tích cực (active) theo mức (level) (cao/thấp tùy thiết lập, thường là mức cao).</li><li>Ngõ ra mang giá trị mà nó lưu trữ (thuận là Q và đảo là <math>\bar{Q}</math>) và ngõ vào để thu nhận dữ liệu hoặc điều khiển (D).</li><li>Có một ngõ nạp để điều khiển việc nhập giá trị (E/Enable) (có thể là Clock/CLK hoặc cổng Logic).</li><li>Bất cứ khi nào E ở trạng thái tích cực (mức cao/high level (1)) hoặc ở trạng thái tích cực (mức cao (1)) cuối cùng thì ngõ ra là giá trị ngõ vào D.</li><li>Khi thay đổi giá trị ngõ vào thì ngõ ra có thể thay đổi ngay lập tức miễn là E ở trạng thái tích cực. Do đó, Latch không có tính đồng bộ.</li><li>D Latch không thể được sử dụng như một thanh ghi/register.</li></ul>	<div></div> <p>Hình 7: D Latch (Trigger: High level, sử dụng CLK) E tích cực (1), ngõ ra trả kết quả theo D (0), ngõ ra <math>\bar{Q}</math> trả kết quả ngược lại (1)</p> <div></div> <p>Hình 8: D Latch (Trigger: High level, sử dụng Cổng AND)</p>	<table><tr><th colspan="3">Input</th><th>Output</th></tr><tr><th>E</th><th>D</th><th>Q</th><th>Q<sup>+</sup></th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><th>E</th><th>Q<sup>+</sup></th></tr><tr><td>0</td><td>Q</td></tr><tr><td>1</td><td>D</td></tr></table>	Input			Output	E	D	Q	Q <sup>+</sup>	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	E	Q <sup>+</sup>	0	Q	1	D
Input			Output																																															
E	D	Q	Q <sup>+</sup>																																															
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																																															
E	Q <sup>+</sup>																																																	
0	Q																																																	
1	D																																																	

		<ul style="list-style-type: none"><li>▪ Latch đơn giản hơn. Do đó tốc độ vận hành của D Latch nhanh và cần ít dung lượng để vận hành hơn D Flip-flop.</li><li>▪ Latch dạng D là Latch dạng Data, có một đầu vào duy nhất là dữ liệu. Ngoài ra còn có S-R Latch, J-K Latch, T Latch.</li></ul>																																																
D Flip-flop	Thiết bị lưu trữ thông tin, dữ liệu, truyền dữ liệu, bộ đếm...	<ul style="list-style-type: none"><li>▪ Có khả năng lưu trữ 1 bit thông tin.</li><li>▪ Lưu trữ thông tin trạng tích cực (active) theo cạnh (edge) (cạnh lên hoặc cạnh xuống tùy thiết lập, thường là cạnh lên).</li><li>▪ Ngõ ra mang giá trị mà nó lưu trữ (thuận là Q và đảo là <math>\bar{Q}</math>) và ngõ vào để thu nhận dữ liệu hoặc điều khiển (D).</li><li>▪ Có một ngõ nạp để điều khiển việc nhập giá trị (CLK/clock).</li><li>▪ Bất cứ khi nào CLK ở trạng thái tích cực (cạnh lên/ rising edge(<math>\uparrow</math>)) thì ngõ ra là giá trị ngõ vào D.</li><li>▪ Khi thay đổi giá trị ngõ vào thì ngõ ra sẽ không thay đổi cho đến khi thay trạng thái CLK (từ - tới <math>\uparrow</math> hoặc từ <math>\uparrow</math> tới - tới <math>\uparrow</math>). Do đó, Flip-flop có tính đồng bộ.</li><li>▪ D Flip-flop có thể được sử dụng như một thanh ghi/register.</li></ul>	 <p>Hình 9: D Flip-flop (Trigger: Rising edge) CLK tích cực (1), ngõ ra Q trả kết quả theo D (1), ngõ ra <math>\bar{Q}</math> trả kết quả ngược lại (0)</p>	<table><tr><th colspan="3">Input</th><th>Output</th></tr><tr><th>CLK</th><th>D</th><th>Q</th><th>Q<sup>+</sup></th></tr><tr><td>-</td><td>0</td><td>0</td><td>0</td></tr><tr><td>-</td><td>0</td><td>1</td><td>1</td></tr><tr><td>-</td><td>1</td><td>0</td><td>0</td></tr><tr><td>-</td><td>1</td><td>1</td><td>1</td></tr><tr><td><math>\uparrow</math></td><td>0</td><td>0</td><td>0</td></tr><tr><td><math>\uparrow</math></td><td>0</td><td>1</td><td>0</td></tr><tr><td><math>\uparrow</math></td><td>1</td><td>0</td><td>1</td></tr><tr><td><math>\uparrow</math></td><td>1</td><td>1</td><td>1</td></tr></table> <table><tr><th>CLK</th><th>Q<sup>+</sup></th></tr><tr><td>-</td><td>Q</td></tr><tr><td><math>\uparrow</math></td><td>D</td></tr></table>	Input			Output	CLK	D	Q	Q <sup>+</sup>	-	0	0	0	-	0	1	1	-	1	0	0	-	1	1	1	$\uparrow$	0	0	0	$\uparrow$	0	1	0	$\uparrow$	1	0	1	$\uparrow$	1	1	1	CLK	Q <sup>+</sup>	-	Q	$\uparrow$	D
Input			Output																																															
CLK	D	Q	Q <sup>+</sup>																																															
-	0	0	0																																															
-	0	1	1																																															
-	1	0	0																																															
-	1	1	1																																															
$\uparrow$	0	0	0																																															
$\uparrow$	0	1	0																																															
$\uparrow$	1	0	1																																															
$\uparrow$	1	1	1																																															
CLK	Q <sup>+</sup>																																																	
-	Q																																																	
$\uparrow$	D																																																	

		<ul style="list-style-type: none"> <li>▪ Flip-flop phức tạp hơn. Do đó tốc độ vận hành của D Flip-flop chậm và cần nhiều dung lượng để vận hành hơn D Latch.</li> <li>▪ Flip-flop dạng D là flip-flop dạng Data, có một đầu vào duy nhất là dữ liệu. Ngoài ra còn có T Flip-flop, J-K Flip-flop, S-R Flip-flop.</li> </ul>		
Thanh ghi (Register)	Thiết bị lưu trữ thông tin, dữ liệu, truyền dữ liệu, làm tăng dung lượng lưu trữ, tăng tốc độ xử lý của các chương trình máy tính.	<ul style="list-style-type: none"> <li>▪ Được cấu tạo bởi n flip-flop (từ 4 Flip-flop trở lên) nối chung ngõ vào CLK, lưu trữ n bit dữ liệu.</li> <li>▪ Dữ liệu từ bộ nhớ chính được chuyển các thanh ghi, tính toán trên chúng, sau đó chuyển kết quả vào bộ nhớ chính.</li> <li>▪ Có các tính chất tương tự như Flip-flop.</li> </ul>	 <p>Hình 10: Thanh ghi (Register) (Trigger: Rising Edge Data Bits: 4) Thanh ghi dạng gộp</p>	



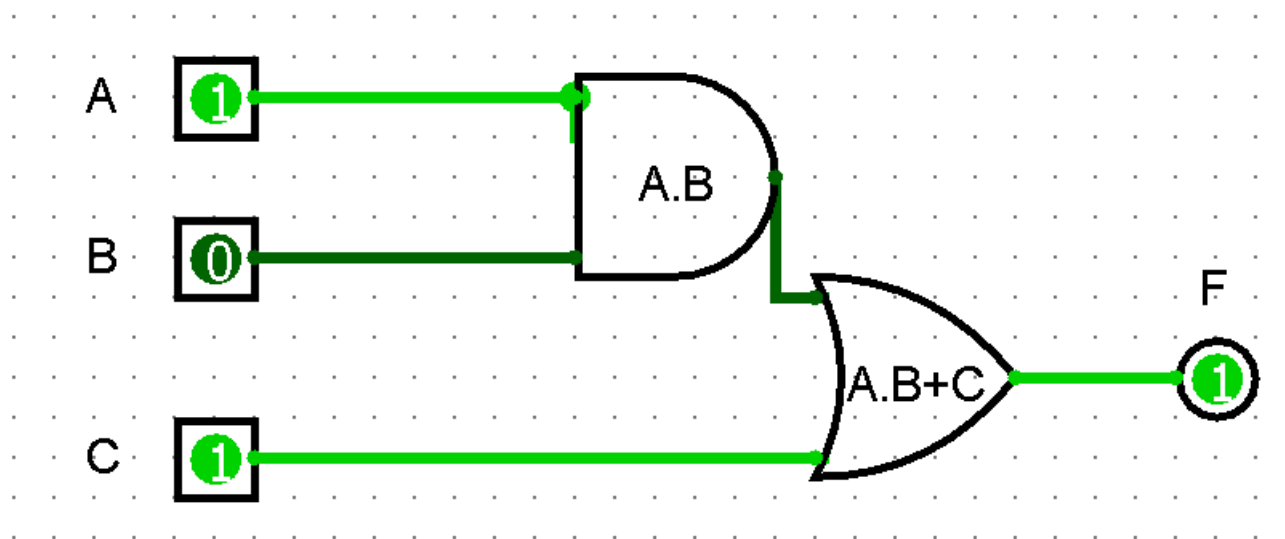
Hình 11: 4 D Flip-flop  
(Trigger: Rising Edge)  
Thanh ghi được tạo từ 4 D Flip-flop

### 1.1. Mô phỏng mạch

#### 1.2.1. Mô phỏng mạch tổ hợp

- Kết quả:





Hình 12: Mạch  $F = A.B + C$

$$A=1, B=0, C=1 \Rightarrow A.B=1 \Rightarrow F=A.B+C=1$$

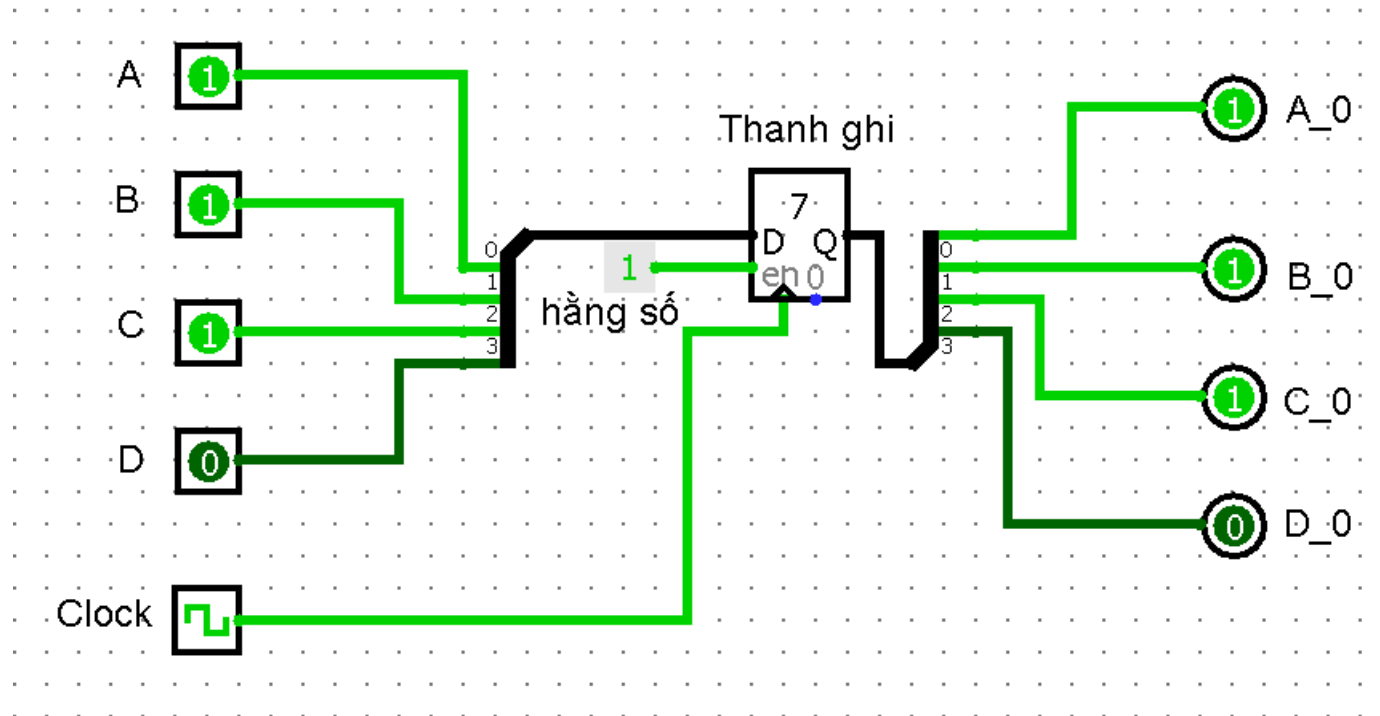
▪ Bảng chân lý:

Input			Output
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

### 1.2.2. Mô phỏng mạch tuần tự

#### ▪ Kết quả

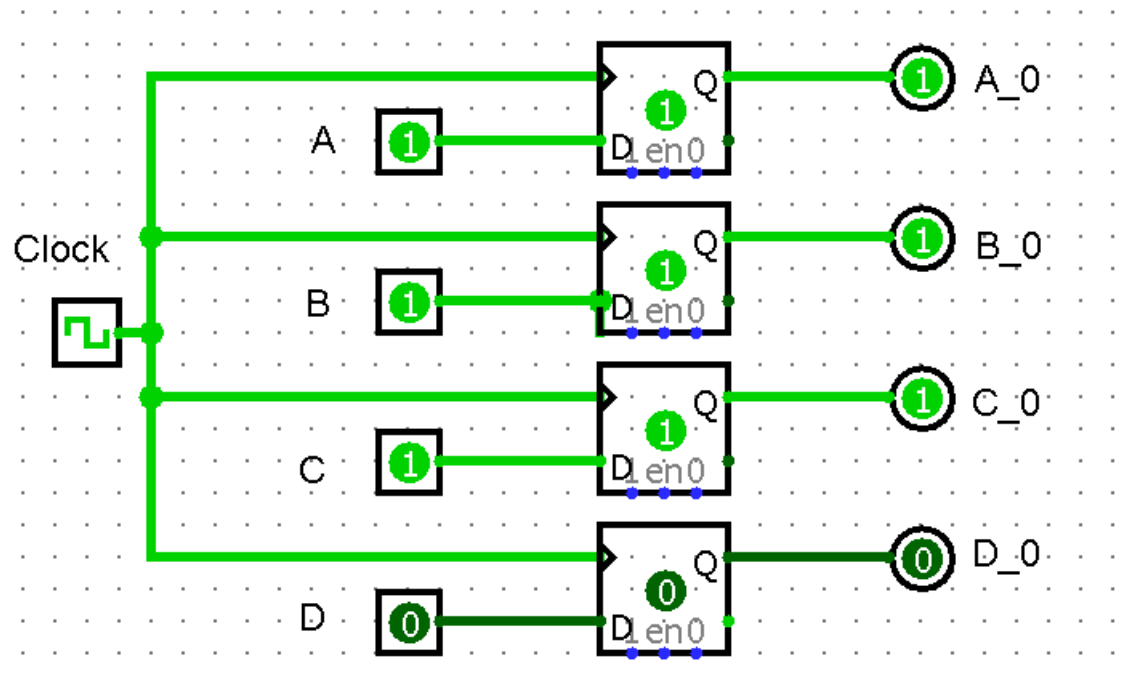


Hình 13: Mạch tuần tự gồm 4 bit ngõ vào truyền dữ liệu qua 1 thanh ghi (dạng gộp).

#### ▪ Trong đó:

- Bộ tách bus và gộp bus có các thuộc tính Fan Out: 4 và Bit Width In: 4;
- Thanh ghi có thuộc tính Data Bits: 4.
- CLK ở trạng thái tích cực nên ngõ ra là giá trị ngõ vào ( $A=A\_0=1, B=B\_0=1, C=C\_0=1, D=D\_0=0$ ).
- Khi thay đổi giá trị ngõ vào thì ngõ ra sẽ không thay đổi cho đến khi thay trạng thái CLK (từ unactive tới active hoặc từ active-unactive-active).

Có thể thay thanh ghi dạng gộp bằng 4 D Flip-flop và cho kết quả tương tự:



Hình 13: Mạch tuần tự gồm 4 bit ngõ vào truyền dữ liệu qua 4 D Flip-flop.