

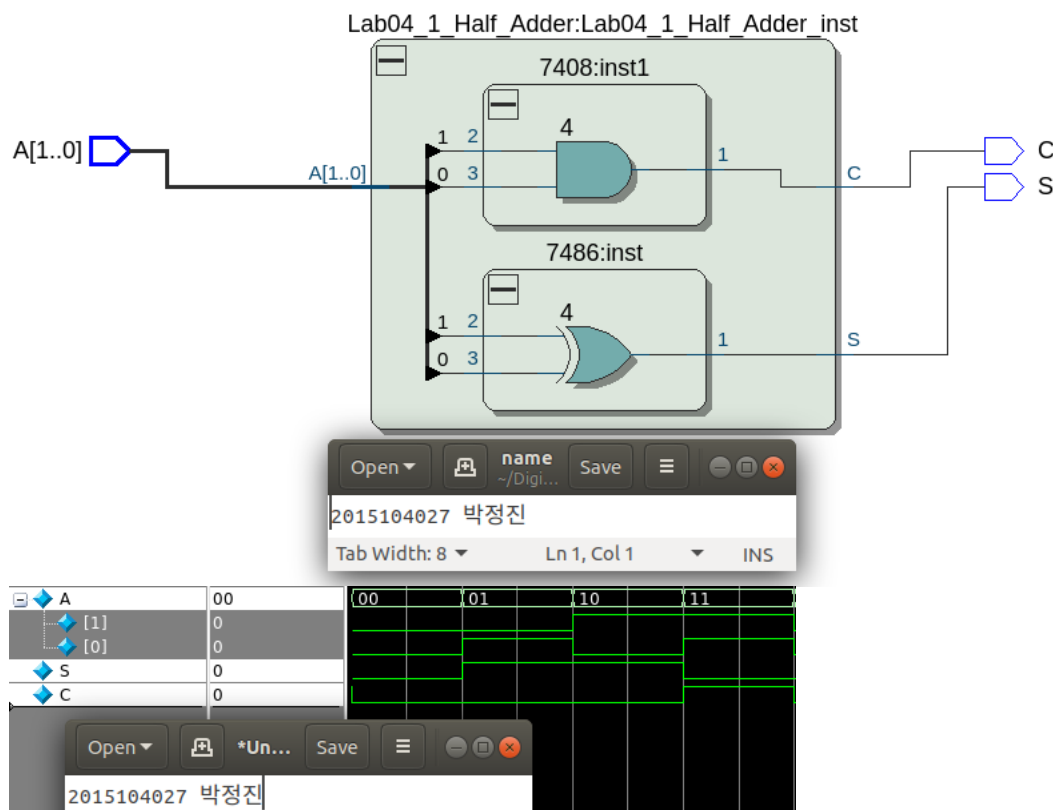
# 디지털회로실험 보고서

-3 주차-

전자공학과  
2015104027  
박정진

## 실험 결과

### Lab04 - 1 Half Adder

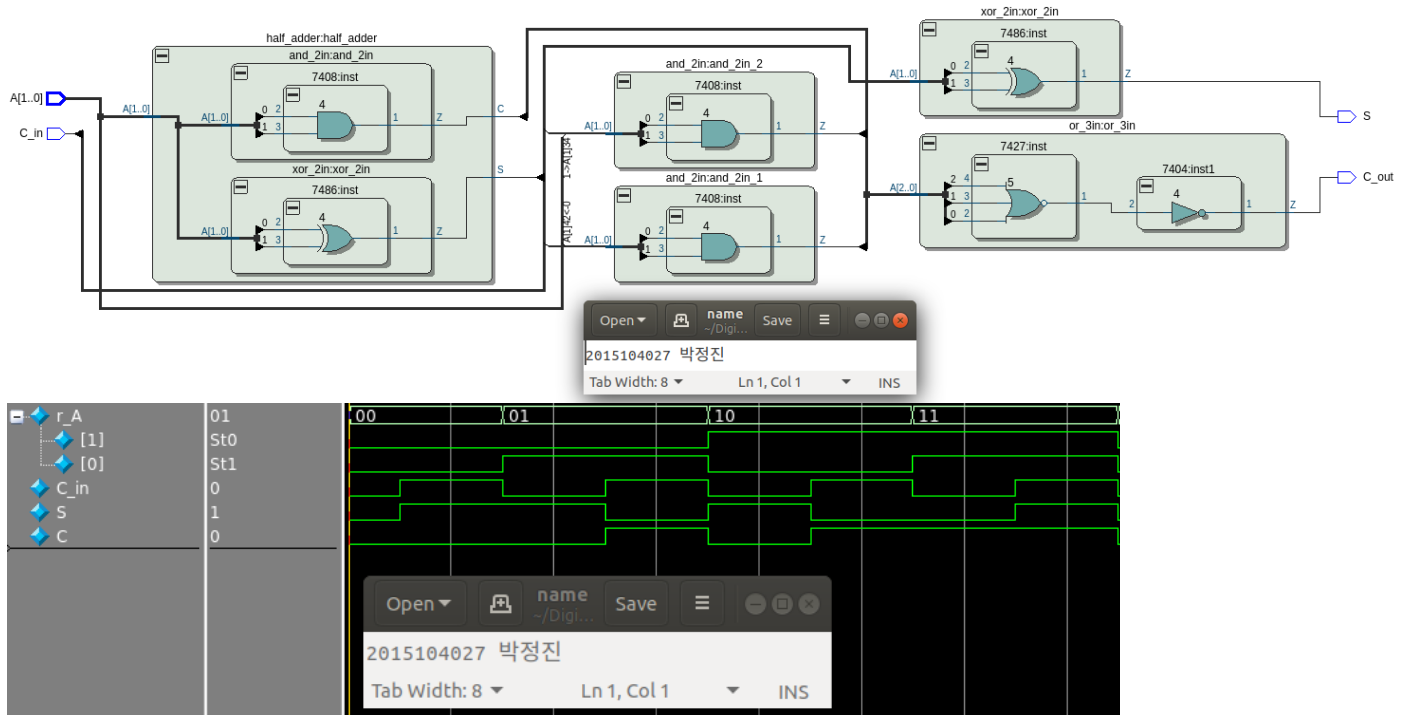


Truth Table

A[1](A)	A[0](B)	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half-Adder 이므로  $A+B$  결과로 Sum 과 Carry 가 알맞게 나오는것으로 알 수 있다.

## Lab04 - 2 Full Adder

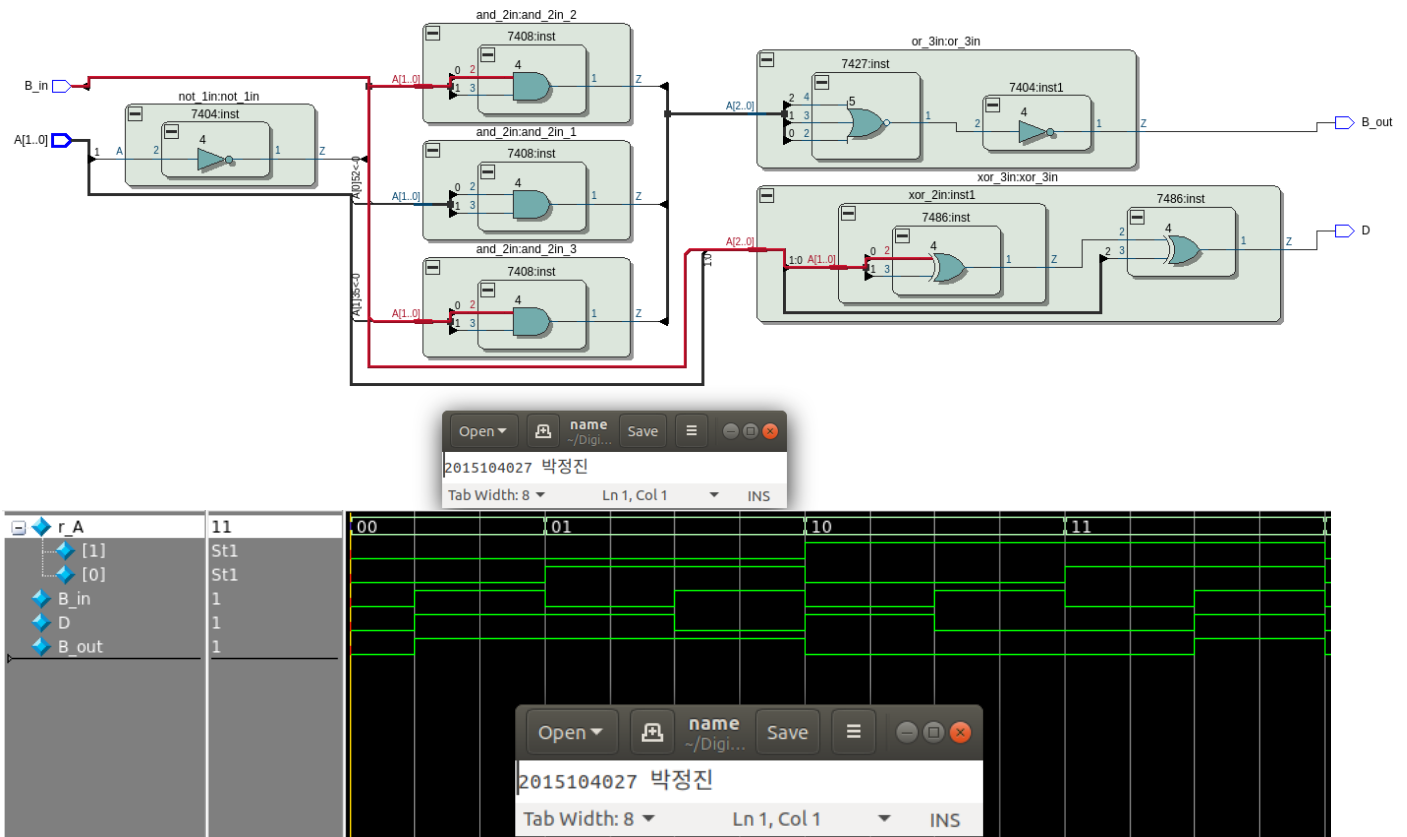


Truth Table

r_A[1](A)	r_A[0](B)	C_in	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full-Adder 이므로  $A + B + \text{Carry\_in}$  결과로 Sum 과 Carry\_Out 가 알맞게 나오는것으로 알 수 있다.

## Lab04 – 3 Full Subtractor

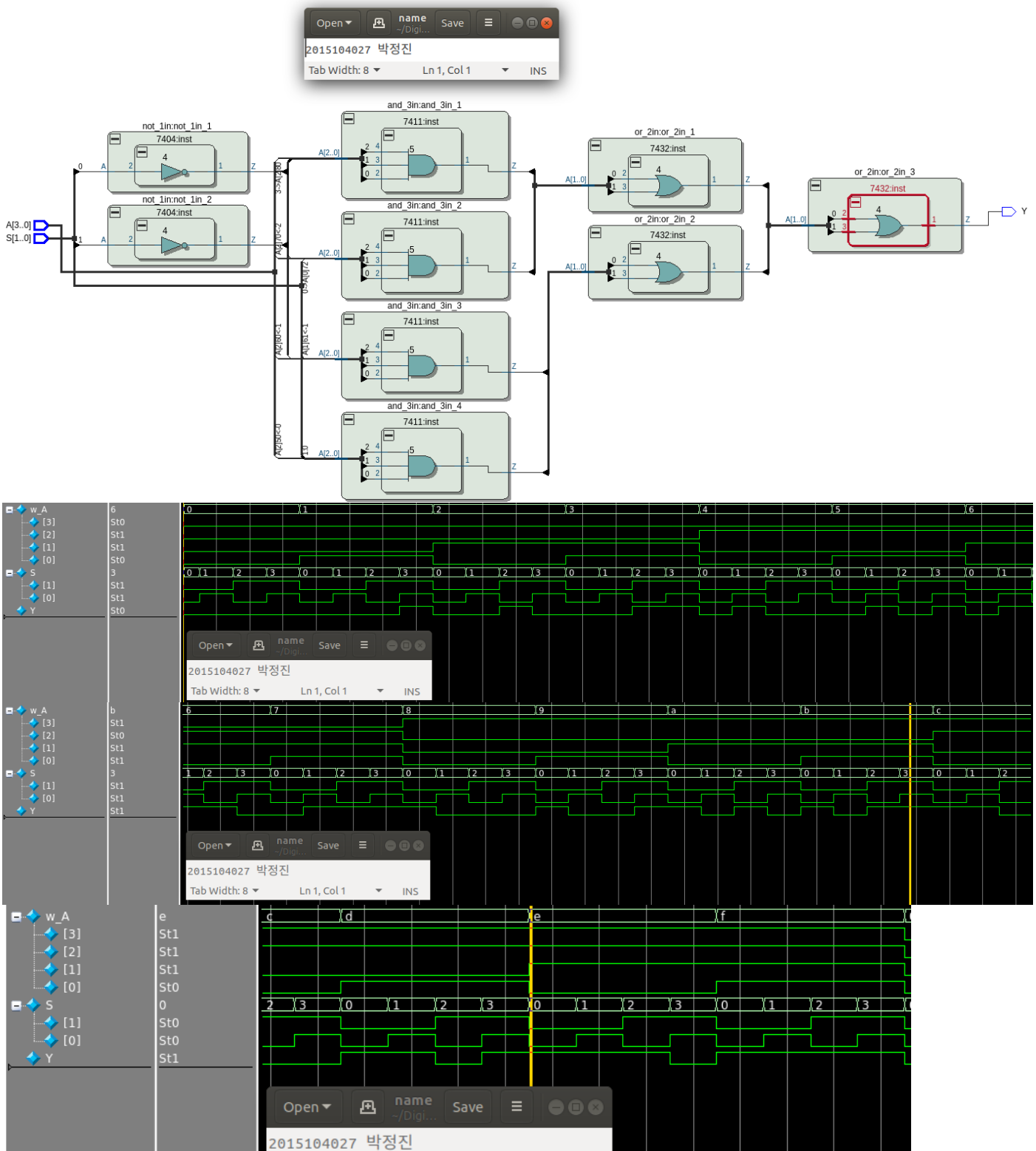


Truth Table

r_A[1](A)	r_A[0](B)	B_in	D	B_out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Full-Subtractor 이므로 A-B B\_in(자리 빌림) 결과로 차와 빌림 자리가 알맞게 나오는것으로 알 수 있다.

## Lab05 -1 4 to 1 Multiplexer



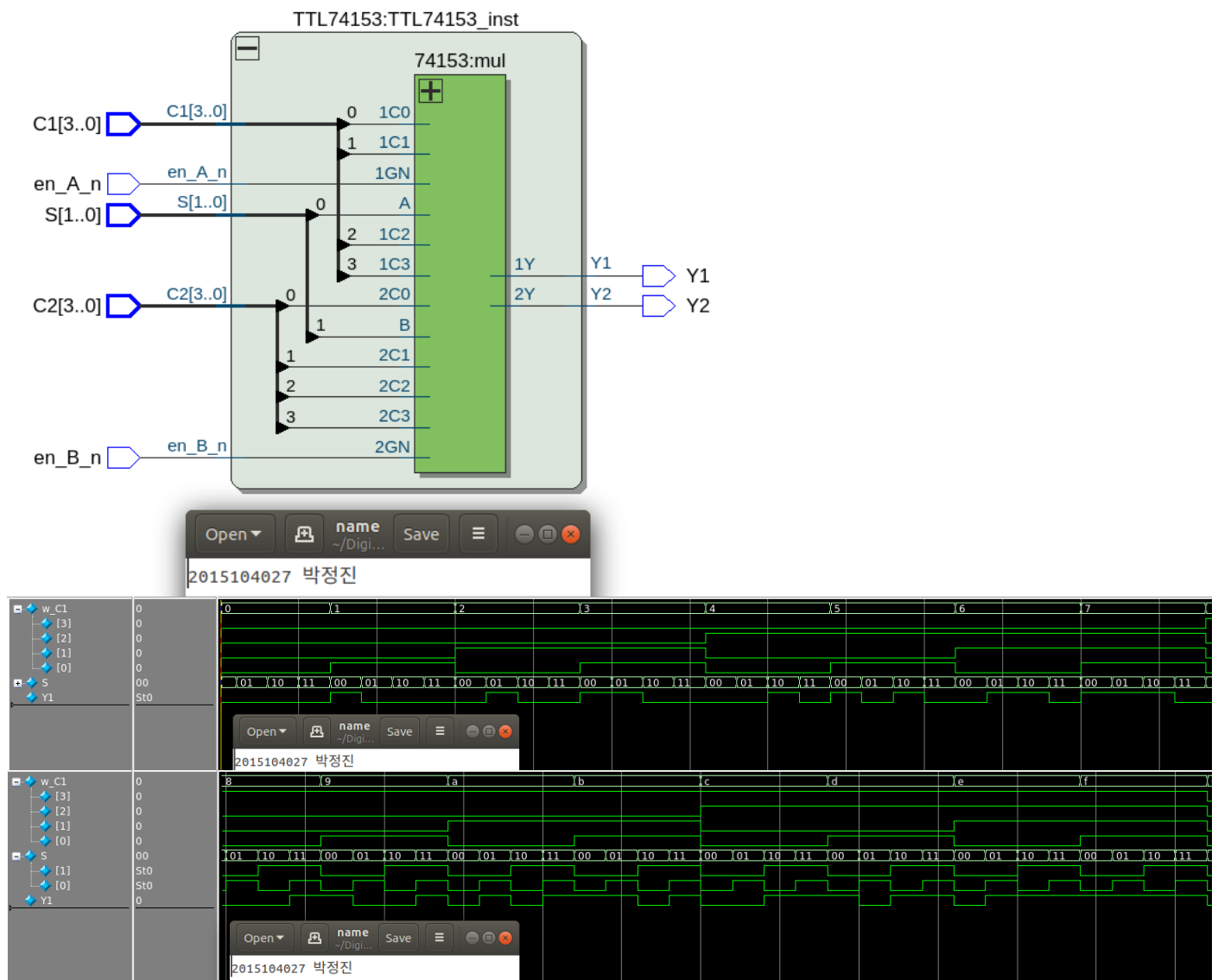
Truth Table

w_A[3](A)	w_A[2](B)	w_A[1](C)	w_A[0](D)	S[1](S1)	S[2](S2)	Y
0	x	x	x	0	0	0
1	x	x	x	0	0	1
x	0	x	x	0	1	0

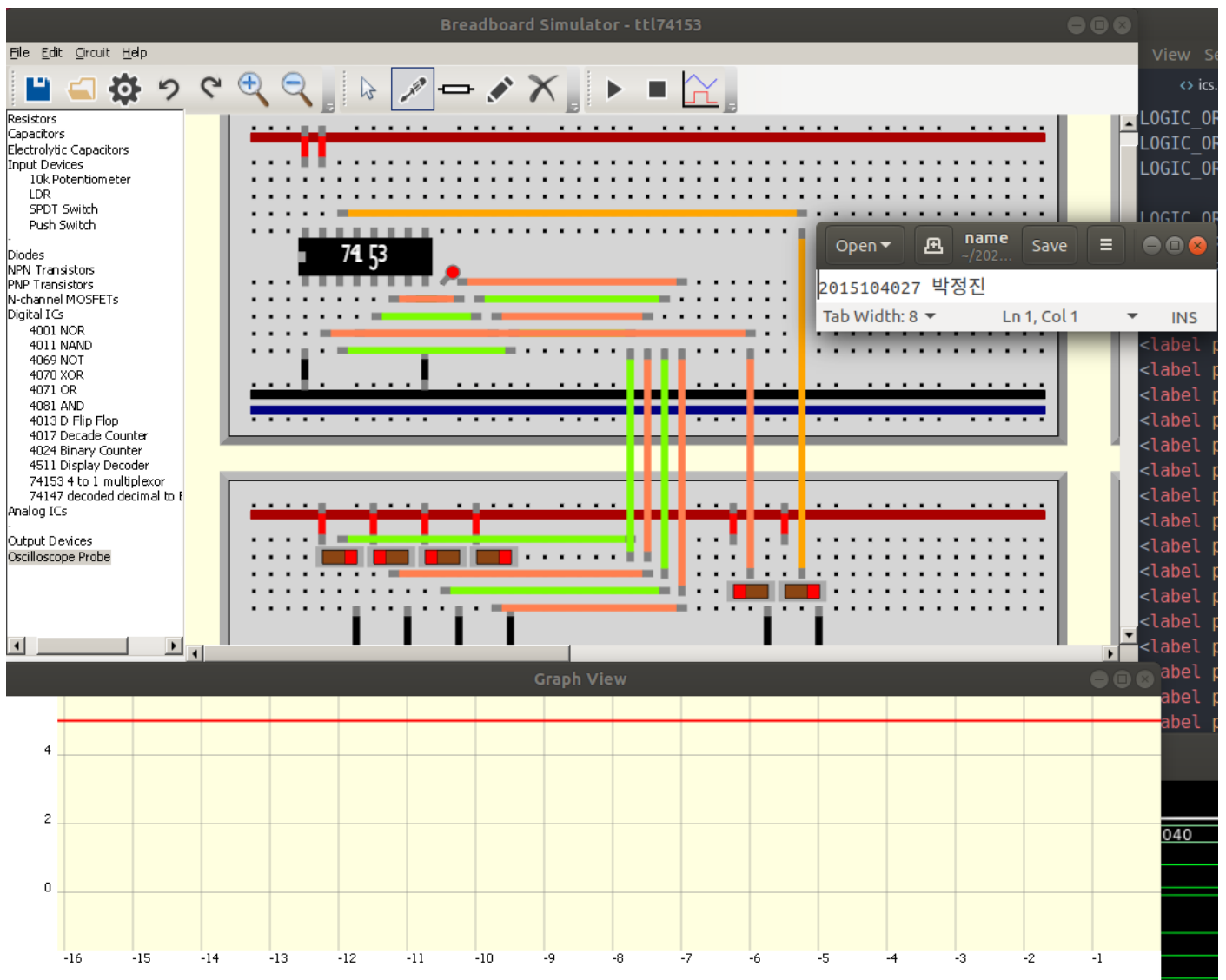
x	1	x	x	0	1	1
x	x	0	x	1	0	0
x	x	1	x	1	0	1
x	x	x	0	1	1	0
x	x	x	1	1	1	1
1	0	0	1	0	0	1
0	1	1	0	0	0	0
0	0	1	0	0	1	0
1	0	0	1	0	1	0
1	1	0	0	1	0	0
0	0	1	1	1	0	1
0	0	1	1	1	1	1
0	1	0	0	1	1	0

4to1 Mux 로 S 의 비트수에 따라 4 개의 Input 중 하나를 선택하는 회로이다. Select bits 는 Negative activation 이며 00 - 3 01 -2 10 -1 11 -0 을 뜻하게 된다.

## Lab05-2 TTL 74153



두개의 4to1 Mux 가 Enable 신호로 컨트롤 되는 칩이다. 실험책에서는 한 Mux 만 사용하므로 다른 쪽 연결(en\_B\_n 과 C2)의 신호는 인가하지 않았다.



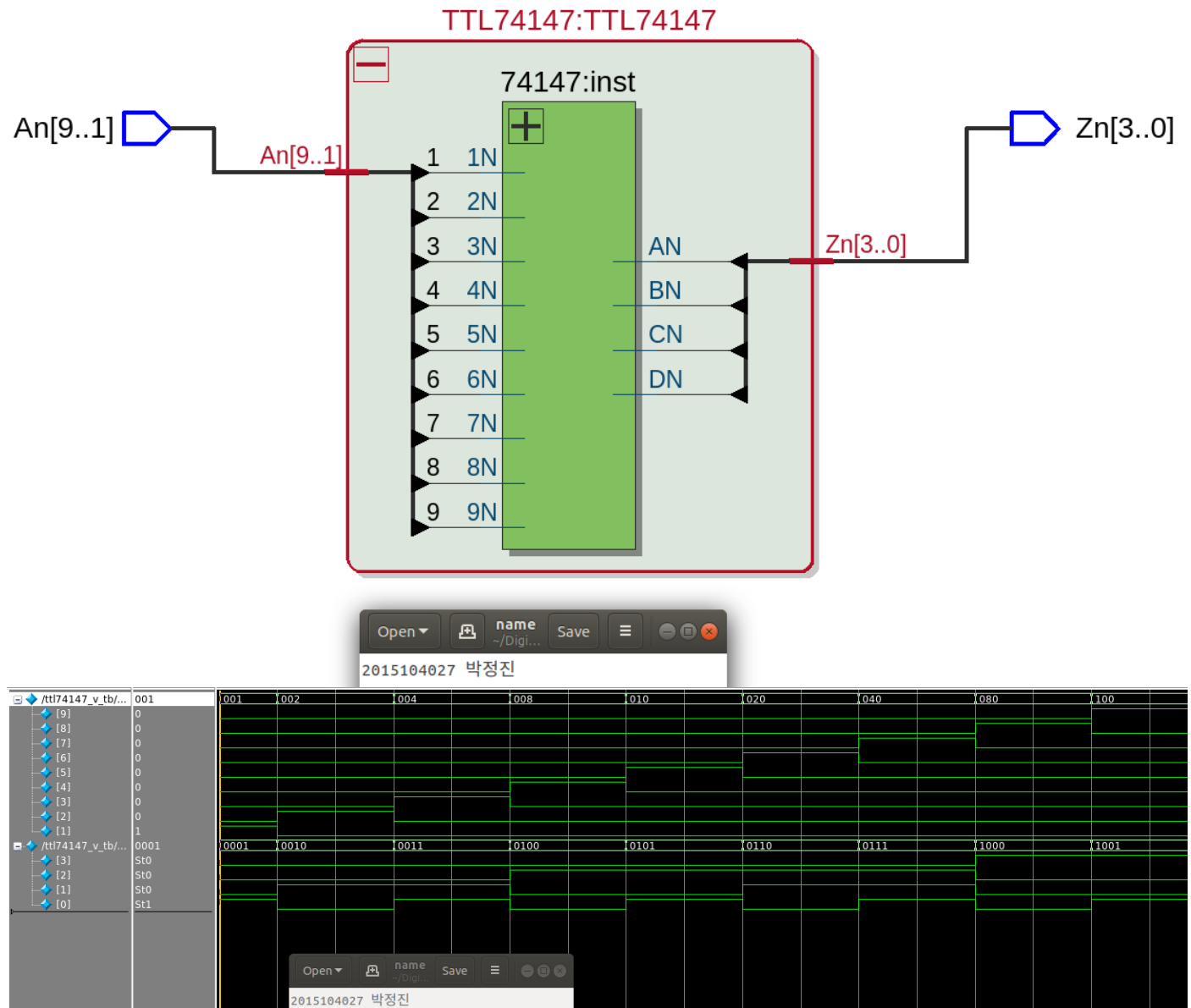
Truth Table

w_C1[3](A)	w_C1[2](B)	w_C1[1](C)	w_C1[0](D)	S[1](S1)	S[2](S2)	Y1
0	x	x	x	0	0	0
1	x	x	x	0	0	1
x	0	x	x	0	1	0
x	1	x	x	0	1	1
x	x	0	x	1	0	0
x	x	1	x	1	0	1
x	x	x	0	1	1	0
x	x	x	1	1	1	1
1	0	0	1	0	0	1
0	1	1	0	0	0	0
0	0	1	0	0	1	0
1	0	0	1	0	1	0
1	1	0	0	1	0	0
0	0	1	1	1	0	1
0	0	1	1	1	1	1
0	1	0	0	1	1	0

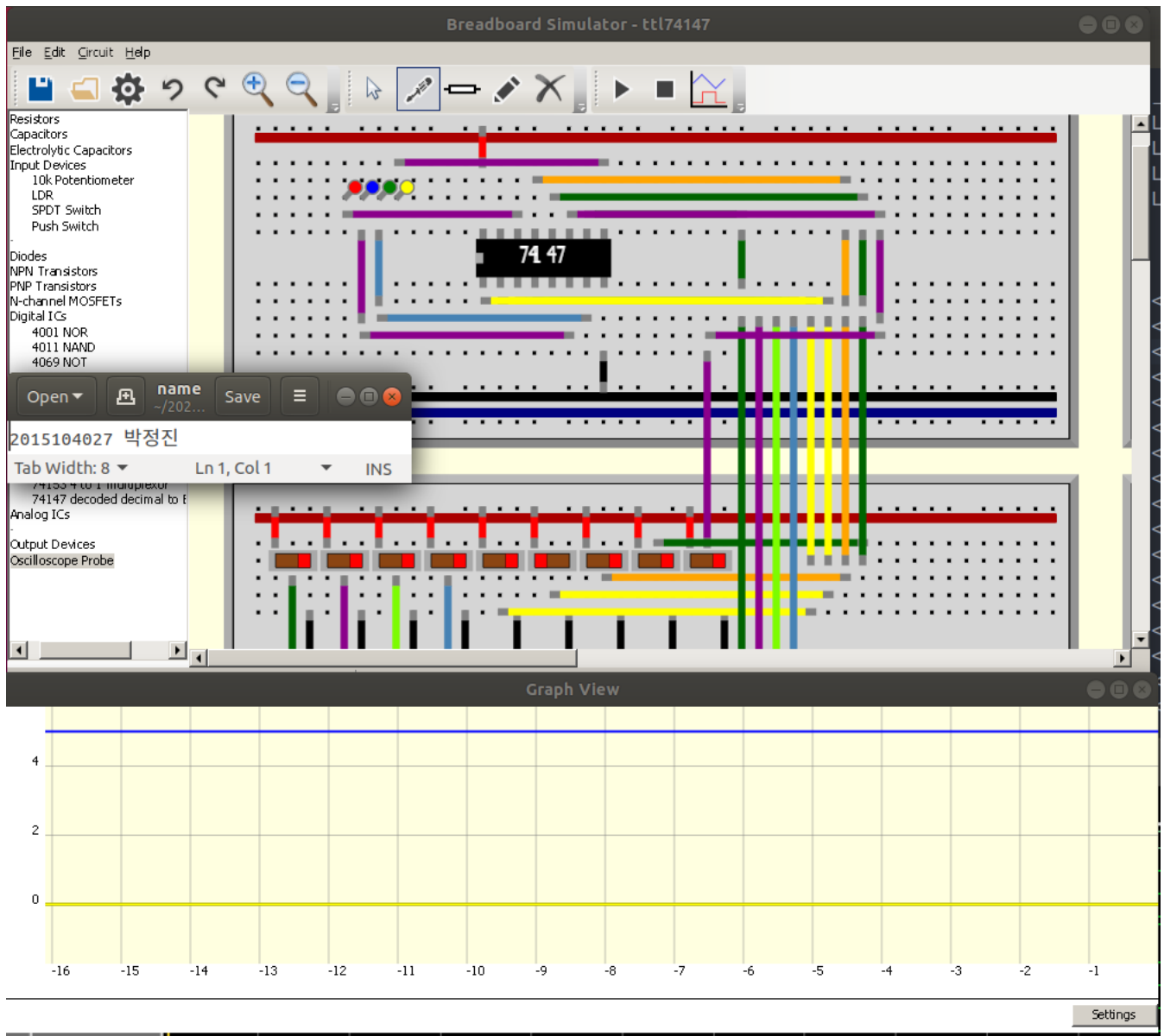
BreadBoard 상에서 실험은

w\_C1 = 4'b0110 S = 2'b10 으로 실험 하였고 Truth Table 처럼 Y1 을 출력이 High 임을 확인할 수 있다.

### Lab05-3 TTL 74147



쿼터스 의 모델 경우 Negative activation 을 표기하는 \*N 에 의해 원래는 제시된 벡터폼의 정확히 반대의 결과가 도출되어야 한다. 하지만 정확하고, 심플한 출력을 위해 74147 의 모델을 불러와 임의 수정했다. Input 을 BCD 코드로 바꿔주는 TTL Logic 이며 인풋의 인덱스 번호 가 BCD 코드가 나올 숫자이다.  
ex ) 1N input - 0001 8N - 1000



Truth Table

An[9](A8)	An[8](A7)	An[7](A6)	An[6](A5)	An[5](A4)	An[4](A3)	An[3](A2)
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
1	0	0	0	0	0	0
An[2](A1)	An[1](A0)	Zn[3](Y3)	Zn[2](Y2)	Zn[1](Y1)	Zn[0](Y0)	
0	1	0	0	0	1	
1	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
0	0	0	1	0	1	
0	0	0	1	1	0	



0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	0	1

일반 적인 쿼터스 모델, 브레드 보드 모듈에서는 위의 Truth Table 에서 반대로 하면 원하는 결과값을 얻을 수 있지만 보기 까다로워 임의로 쿼터스 모델, 브레드 보드 모듈의 Input,Output 에 수정을 조금 했다

ics.xml 파일에서 수정한 74147 코드이다.

```
<model name="74147 decoded decimal to BCD encoder" category="Digital" footprint="DIP16">
<data>LOGIC_AND {r}.and5_1 {r}.and5_1_i1 {r}.and5_1_i2 {r}.and5_1_o {8} {16}
LOGIC_AND {r}.and5_11 {r}.A0 {r}.A1b {r}.and5_1_i1 {8} {16}
LOGIC_AND {r}.and5_12 {r}.A3b {r}.and5_12_i2 {r}.and5_1_i2 {8} {16}
LOGIC_AND {r}.and5_13 {r}.A5b {r}.nor_1_o {r}.and5_12_i2 {8} {16}

LOGIC_AND {r}.and4_1 {r}.and4_1_i1 {r}.and4_1_i2 {r}.and4_1_o {8} {16}
LOGIC_AND {r}.and4_11 {r}.A2 {r}.A3b {r}.and4_1_i1 {8} {16}
LOGIC_AND {r}.and4_12 {r}.A5b {r}.nor_1_o {r}.and4_1_i2 {8} {16}

LOGIC_AND {r}.and3 {r}.A4 {r}.and3_i2 {r}.and3_o {8} {16}
LOGIC_AND {r}.and3_1 {r}.A5b {r}.nor_1_o {r}.and3_i2 {8} {16}

LOGIC_AND {r}.and_1 {r}.A6 {r}.nor_1_o {r}.and_1_o {8} {16}

LOGIC_AND {r}.and4_2 {r}.and4_2_i1 {r}.and4_2_i2 {r}.and4_2_o {8} {16}
LOGIC_AND {r}.and4_21 {r}.A1 {r}.A3b {r}.and4_2_i1 {8} {16}
LOGIC_AND {r}.and4_22 {r}.A4b {r}.nor_1_o {r}.and4_2_i2 {8} {16}

LOGIC_AND {r}.and4_3 {r}.and4_3_i1 {r}.and4_3_i2 {r}.and4_3_o {8} {16}
LOGIC_AND {r}.and4_31 {r}.A2 {r}.A3b {r}.and4_3_i1 {8} {16}
LOGIC_AND {r}.and4_32 {r}.A4b {r}.nor_1_o {r}.and4_3_i2 {8} {16}

LOGIC_AND {r}.and_2 {r}.A5 {r}.nor_1_o {r}.and_2_o {8} {16}

LOGIC_AND {r}.and_3 {r}.A6 {r}.nor_1_o {r}.and_3_o {8} {16}

LOGIC_AND {r}.and_4 {r}.A3 {r}.nor_1_o {r}.and_4_o {8} {16}
LOGIC_AND {r}.and_5 {r}.A4 {r}.nor_1_o {r}.and_5_o {8} {16}
LOGIC_AND {r}.and_6 {r}.A5 {r}.nor_1_o {r}.and_6_o {8} {16}
LOGIC_AND {r}.and_7 {r}.A6 {r}.nor_1_o {r}.and_7_o {8} {16}

LOGIC_NOR {r}.nor_1 {r}.A7 {r}.A8 {r}.nor_1_o {8} {16}
LOGIC_NOR {r}.nor_2 {r}.A7 {r}.A8 {14} {8} {16}

LOGIC_NOT {r}.nor5 {r}.nor5_i {9} {8} {16}
LOGIC_OR {r}.nor5_1 {r}.nor5_1_i1 {r}.nor5_1_i2 {r}.nor5_i {8} {16}
```

```

LOGIC_OR {r}.nor5_2 {r}.and5_1_o {r}.and4_1_o {r}.nor5_1_i1 {8} {16}
LOGIC_OR {r}.nor5_3 {r}.and3_o {r}.nor5_3_i2 {r}.nor5_1_i2 {8} {16}
LOGIC_OR {r}.nor5_4 {r}.and_1_o {r}.A8 {r}.nor5_3_i2 {8} {16}

LOGIC_NOT {r}.nor4_1 {r}.nor4_1_i {7} {8} {16}
LOGIC_OR {r}.nor4_11 {r}.nor4_11_i1 {r}.nor4_11_i2 {r}.nor4_1_i {8} {16}
LOGIC_OR {r}.nor4_12 {r}.and4_2_o {r}.and4_3_o {r}.nor4_11_i1 {8} {16}
LOGIC_OR {r}.nor4_13 {r}.and_2_o {r}.and_3_o {r}.nor4_11_i2 {8} {16}

LOGIC_NOT {r}.nor4_2 {r}.nor4_2_i {6} {8} {16}
LOGIC_OR {r}.nor4_21 {r}.nor4_21_i1 {r}.nor4_21_i2 {r}.nor4_2_i {8} {16}
LOGIC_OR {r}.nor4_22 {r}.and_4_o {r}.and_5_o {r}.nor4_21_i1 {8} {16}
LOGIC_OR {r}.nor4_23 {r}.and_6_o {r}.and_7_o {r}.nor4_21_i2 {8} {16}

LOGIC_NOT {r}.not1 {11} {r}.A0 {8} {16}
LOGIC_NOT {r}.not2 {12} {r}.A1 {8} {16}
LOGIC_NOT {r}.not3 {r}.A1 {r}.A1b {8} {16}
LOGIC_NOT {r}.not4 {13} {r}.A2 {8} {16}
LOGIC_NOT {r}.not5 {1} {r}.A3 {8} {16}
LOGIC_NOT {r}.not6 {r}.A3 {r}.A3b {8} {16}
LOGIC_NOT {r}.not7 {2} {r}.A4 {8} {16}
LOGIC_NOT {r}.not8 {r}.A4 {r}.A4b {8} {16}
LOGIC_NOT {r}.not9 {3} {r}.A5 {8} {16}
LOGIC_NOT {r}.not10 {r}.A5 {r}.A5b {8} {16}
LOGIC_NOT {r}.not11 {4} {r}.A6 {8} {16}
LOGIC_NOT {r}.not12 {5} {r}.A7 {8} {16}
LOGIC_NOT {r}.not13 {10} {r}.A8 {8} {16}

```

```

</data>
<label pin="1" name="A3b Input"/>
<label pin="2" name="A4b Input"/>
<label pin="3" name="A5b Input"/>
<label pin="4" name="A6b Input"/>
<label pin="5" name="A7b Input"/>
<label pin="6" name="Y2b Output"/>
<label pin="7" name="Y1b Output"/>
<label pin="8" name="Ground"/>
<label pin="9" name="Y0b Output"/>
<label pin="10" name="A8b Input"/>
<label pin="11" name="A0b Input"/>
<label pin="12" name="A1b Input"/>
<label pin="13" name="A2b Input"/>
<label pin="14" name="Y3b Output"/>
<label pin="15" name="NC Output"/>
<label pin="16" name="Supply"/>

</model>

```

위의 코드대로 진행하면 위의 Vector 결과와 같은 결과를 얻을 수 있다.