**디지털회로실험 보고서**

-3주차-

전자공학과

2015104027

박정진

**실험 결과**

**Lab04 - 1 Half Adder**

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

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 Sum과 Carry 가 알맞게 나오는것으로 알 수 있다.

**Lab04 - 2 Full Adder**

A picture containing screenshot, clock

Description automatically generatedA picture containing clock, monitor

Description automatically generated

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+Carry\_in 결과로 Sum과 Carry\_Out 가 알맞게 나오는것으로 알 수 있다.

**Lab04 - 3 Full Subtractor**

A picture containing screenshot, clock

Description automatically generatedA picture containing monitor

Description automatically generated

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**

A close up of a device

Description automatically generatedA picture containing clock, green, white

Description automatically generatedA close up of a screen

Description automatically generatedA picture containing clock, monitor, sitting, mounted

Description automatically generated

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 로S의 비트수에 따라 4 개의 Input 중 하나를 선택하는 회로이다. Select bits는 Negative activation 이며 00 - 3 01 -2 10 -1 11 -0 을 뜻하게 된다.

**Lab05-2 TTL 74153**

A picture containing clock

Description automatically generated

A close up of a device

Description automatically generated

A picture containing clock

Description automatically generated

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

A picture containing screenshot

Description automatically generated

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**

A picture containing clock

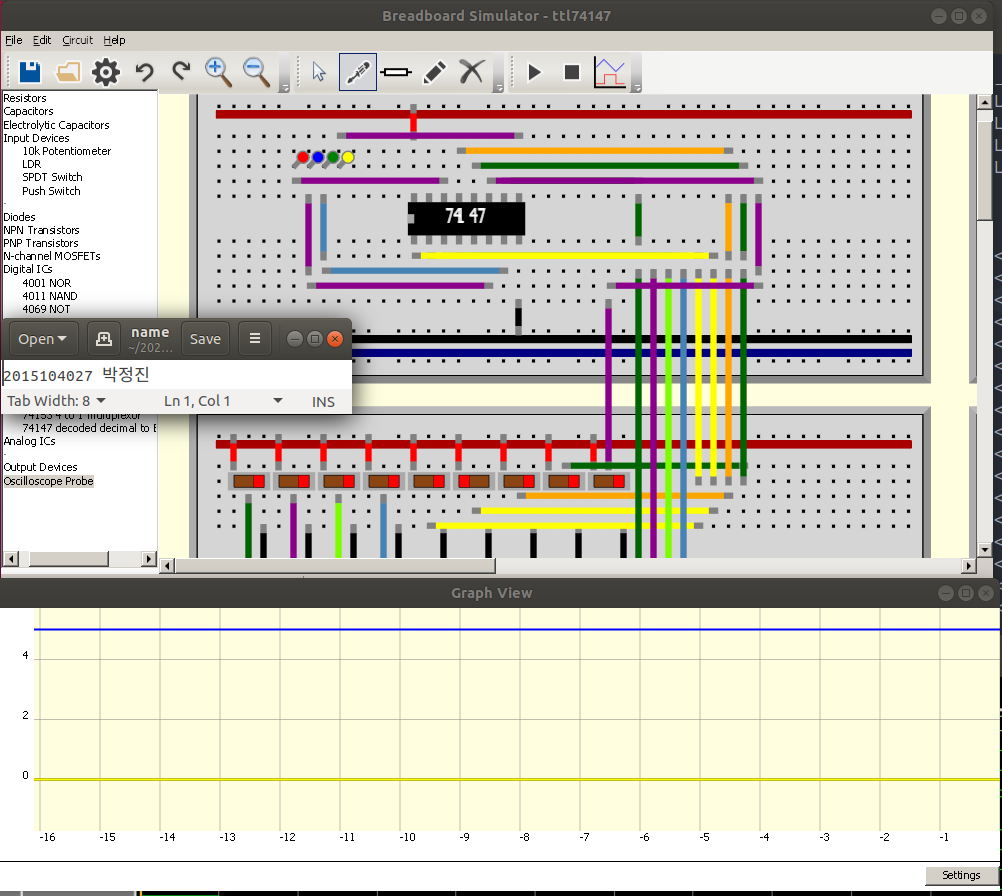
Description automatically generatedA close up of a computer

Description automatically generated

쿼터스 의 모델 경우 Negative activation 을 표기하는 \*N에 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 BCD코드가 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>

위 의 코드대로 진행하면 위의 Vector 결과와 같은 결과 을 얻을 수 있다.