CS211 Lab#3 Pre-Report

20160410 윤석훈

1. What is multiplexer?

여러 개의 데이터 입력 중 한 개의 상태를 그대로 출력하는 회로로, 어떤 선택용 제어신호를 이용한다.

2.

\* Truth Table of 8-to-1 MUX

|  |  |  |  |
| --- | --- | --- | --- |
| S0 | S1 | S2 | Output |
| 0 | 0 | 0 | Input0 |
| 0 | 0 | 1 | Input1 |
| 0 | 1 | 0 | Input2 |
| 0 | 1 | 1 | Input3 |
| 1 | 0 | 0 | Input4 |
| 1 | 0 | 1 | Input5 |
| 1 | 1 | 0 | Input6 |
| 1 | 1 | 1 | Input7 |

\*pseudo-code in combinational statements

Input: I0~I7 : value

Sel : 3bit vector

Output: Out : value

Architecture Combi of Combinational

begin

Out <= I0 when Sel=”000” else

<= I1 when Sel=”001” else

<= I2 when Sel=”010” else

<= I3 when Sel=”011” else

<= I4 when Sel=”100” else

<= I5 when Sel=”101” else

<= I6 when Sel=”110” else

<= I7 when Sel=”111”

End architecture

\*Pseudo-code in sequential statements

Architecture Seq of Sequential

begin

Process(Sel)

Begin

Case Sel is

When “000” : Out <= I0

When “001” : Out <= I1

When “010” : Out <= I2

When “011” : Out <= I3

When “100” : Out <= I4

When “101” : Out <= I5

When “110” : Out <= I6

When “111” : Out <= I7

end process

end architecture

3. Pseudo-code for 7-segment

\*in function

Input: Num : 4 bit vector (which represents 0~15)

Output: Seg : 7 bit vector

Architecture seg of segfun is

Function segf (Num : in std\_logic\_vector(4))

Begin

if Num=”0000” then return “1000000”

else if Num=”0001” then return “1111001”

else if Num=”0010” then return “0100100”

else if Num=”0011” then return “0110000”

else if Num=”0100” then return “0011001”

else if Num=”0101” then return “0010010”

else if Num=”0110” then return “0000010”

else if Num=”0111” then return “1111000”

else if Num=”1000” then return “0000000”

else if Num=”1001” then return “0011000”

else if Num=”1010” then return “0001000”

else if Num=”1011” then return “0000000”

else if Num=”1100” then return “1000110”

else if Num=”1101” then return “1000000”

else if Num=”1110” then return “0000110”

else if Num=”1111” then return “1001110”

end

end

\*in component

ENTITY COMPO IS

PORT (inputval : IN std\_logic\_vector(4);

Outputval: std\_logic(7));

ARCHITECTURE REDBOX of COMPO is

begin

outputval <= “1000000 “ when inputval=”0000“ else

outputval <= “1111001 “ when inputval=”0001“ else

outputval <= “0100100 “ when inputval=”0010“ else

outputval <= “0110000 “ when inputval=”0011“ else

outputval <= “0011001 “ when inputval=”0100“ else

outputval <= “0010010 “ when inputval=”0101“ else

outputval <= “0000010 “ when inputval=”0110“ else

outputval <= “1111000 “ when inputval=”0111“ else

outputval <= “0000000 “ when inputval=”1000“ else

outputval <= “0011000 “ when inputval=”1001“ else

outputval <= “0001000 “ when inputval=”1010“ else

outputval <= “0000000 “ when inputval=”1011“ else

outputval <= “1000110 “ when inputval=”1100“ else

outputval <= “1000000 “ when inputval=”1101“ else

outputval <= “0000110 “ when inputval=”1110“ else

outputval <= “1001110 “ when inputval=”1111“ else

end COMPO

나중에 다른 아키텍쳐에서 컴포넌트로 불러온다.