Lab5 Pre-report

20160410 윤석훈

1. Half Adder, Full Adder

Half-Adder 는 오직 2개의 비트를 계산하는 회로이다. Full-Adder 는 Carry(Ci)를 포함하는 다음 논리식을 구현하는 회로이다. Si=(not Xi)Yi(not Ci)+Xi(not Yi)(not Ci)+(not Xi)(not Yi)Ci+XiYiCi,

when C(i+1)=XiYi+XiCi+YiCi.

2. Two’s Complement

어떤 n-bit의 숫자 B=BnB(n-1)B(n-2)..B1는 2의 보수법에서 다음과 같이 계산된다. V(B)=B1+B2\*2+B2\*(2^2)+…+B(n-1)\*(2^(n-2))-Bn\*(2^(n-1)).

이러한 2의 보수법으로 뺄셈을 할 때는 모든 음수를 2보수 비트로 표현한 뒤 더하면 된다(자릿수를 넘어간 Carry는 무시해준다). 음수를 n비트의 2보수비트로 표현하기 위해서는, 음수의 절댓값을 n비트로 표현한 후, 그것을 2^n에서 빼주면 된다.

3. (1) Half-adder component

Entity HA is

Port (x,y: in std\_logic;

S,c : out std\_logic);

End HA

Architecture logicfunc of HA is

Begin

S<=x Xor y;

C<=x And y;

End logicfunc

(2) Full-adder component

Entity FA is

Port( Carryin, X, Y : In std\_logic;

S, Carryout : out std\_logic);

End FA

Architecture LogicFunc of FA is

Begin

S<= x XOR y XOR Cin;

Cout<= (x And y) OR (Cin And x) OR (Cin And y);

End logicfunc

(3) 2’s complement component

Entity twos is

Port (x: in std\_logic\_vector(7 downto 0);

twox: out std\_logic\_vector(7 downto 0));

End twos;

Architecture logicfun of twos is

Begin

Process(x(7))

Begin

if(x(7)=’1’) then

y<= “11111111”-x+1;

else

y<=x;

end if;

end process;

end logicfun

(4) Comparison component

Entity compare is

Port(x,y : in std\_logic\_vector (8 downto 0);

4. 8 bit ripple-carry adder는 8개의 full adder를 연결하여 각 비트마다 sum signal을 출력하고 carry signal은 마지막 full adder까지 이동(ripple)시켜 출력하는 방법이다. 8비트 덧셈을 회로로 구현할 수 있는 것이다.