

# HBE-COMBO II VHDL 실습

## 제 4주차 강의

(주) 한백전자 기술연구소





# 장비 구성 및 형상

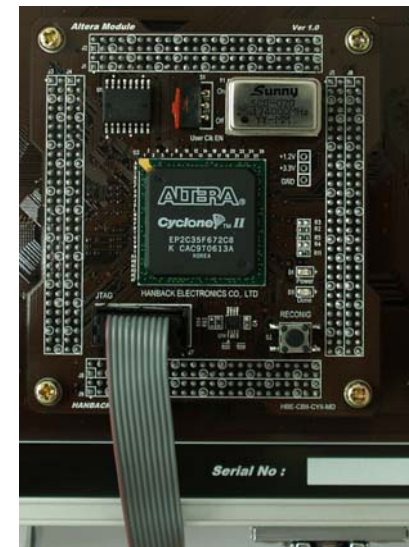




# 장비 구성 및 형상



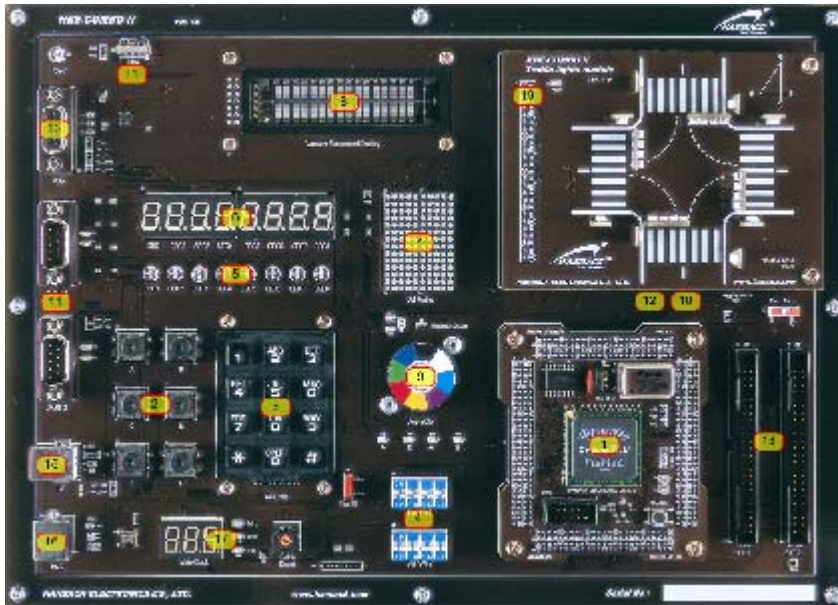
[전원케이블 연결]



[다운로드 케이블 연결]



# 장비 구성 및 형상



1. FPGA Module
2. Button S/W
3. Keypad
4. Bus S/W
5. LED
6. 7-Segment
7. DOT
8. VFD(Vacuum Fluorescent Display)
9. STEP MOTOR
10. SRAM
11. IrDA
11. IrDA
12. Piezo
13. VGA port
14. UART
15. USB to Serial
16. PS/2 port
17. Clock control block
18. Expansion port
19. Expansion port (Daughter)



# Contents

- EX\_3\_3. 8 X 3 ENCODER
- EX\_4\_1. 채널 선택회로(Multiplex)
- EX\_4\_2. 채널 분배회로



# EX\_3\_3. 8 X 3 ENCODER 기본 이론

E1	INPUTS								OUTPUTS				
	I(0)	I(1)	I(2)	I(3)	I(4)	I(5)	I(6)	I(7)	A(2)	A(1)	A(0)	GS	EO
1	X	X	X	X	X	X	X	X	X	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	X	X	X	X	X	X	X		0	0	0	0	1
0	X	X	X	X	X			0	1	0	0	1	0
0	X	X	X			0	1	1	1	0	1	1	0
0	X	X			0	1	1	1	1	1	0	0	1
0	X	X		0	1	1	1	1	1	1	0	1	0
0	X		0	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

8진-2진 인코더는 8개의 디지털들 각각에 대해 입력이 필요하므로 8개의 입력이 있어야 하며 그 대응되는 2진수를 산출하기 위해 3개의 출력들이 있어야 한다. 즉, 인코더는 코드화기이다. 출력 A(0)는 입력 8진 디지털이 홀수이면 "1"이 된다. 출력 A(1)는 입력 8진 디지털이 2, 3, 6, 7이면 "1"이 된다. 또, 출력 A(2)는 입력 8진 디지털이 4, 5, 6, 7일 때 "1"이 된다.



# EX\_3\_3. 8 X 3 ENCODER 구문

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY EX3_3 IS
PORT(
  I : IN STD_LOGIC_VECTOR(7 DOWNTO 0);
  EI : IN STD_LOGIC;
  EO, GS : OUT STD_LOGIC;
  A : OUT STD_LOGIC_VECTOR(2 DOWNTO 0)
);
END EX3_3;

ARCHITECTURE HB OF EX3_3 IS
BEGIN

  PROCESS(I, EI)
  BEGIN
    IF EI = '1' THEN
      A <= "111";
      GS <= '1';
      EO <= '1';
    ELSIF I = "11111111" THEN
      A <= "111";
      GS <= '1';
      EO <= '0';
    ELSIF I(7) = '0' THEN
      A <= "000";
      GS <= '0';
      EO <= '1';
    END IF;
  END PROCESS;
END;
```

- I(7 DOWNTO 0) : 8개의 디지트들 각각에 대해 입력이 필요하므로 8개의 입력 비트
- A : 그 대응되는 2진수를 산출하기 위해서 3개의 출력비트
- GS : ambiguity 1을 제거, GS=0 유효출력



# EX\_3\_3. 8 X 3 ENCODER

```
ELSIF I(7 DOWNT0 6) = "10" THEN
  A <= "001";
  GS <= '0';
  EO <= '1';
ELSIF I(7 DOWNT0 5) = "110" THEN
  A <= "010";
  GS <= '0';
  EO <= '1';
ELSIF I(7 DOWNT0 4) = "1110" THEN
  A <= "011";
  GS <= '0';
  EO <= '1';
ELSIF I(7 DOWNT0 3) = "11110" THEN
  A <= "100";
  GS <= '0';
  EO <= '1';
ELSIF I(7 DOWNT0 2) = "111110" THEN
  A <= "101";
  GS <= '0';
  EO <= '1';
ELSIF I(7 DOWNT0 1) = "1111110" THEN
  A <= "110";
  GS <= '0';
  EO <= '1';
ELSIF I(7 DOWNT0 0) = "11111110" THEN
  A <= "111";
  GS <= '0';
  EO <= '1';
END IF;
END PROCESS;
```

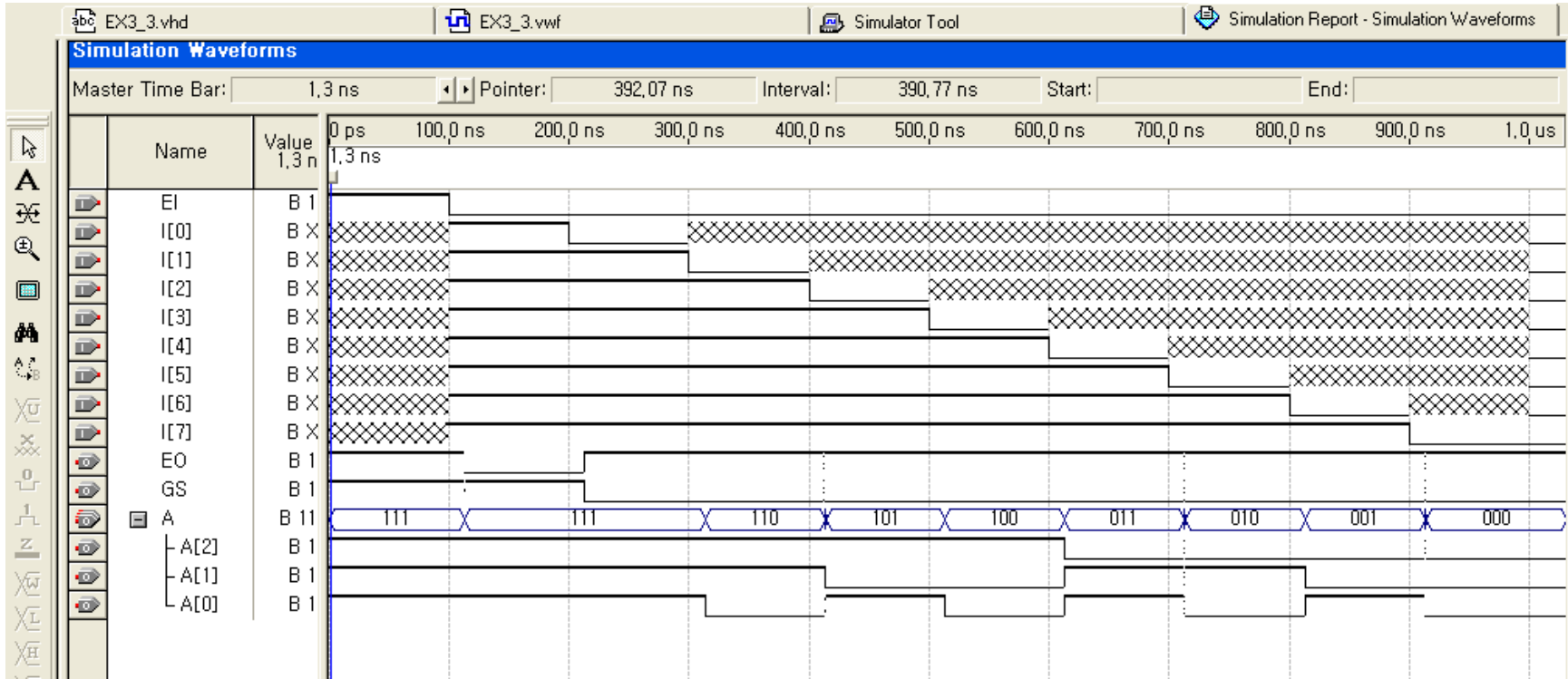
END HB;

- TI : 확장 연결시 사용
- EO : 확장 연결시 사용
- TI=H, 엔코딩 입력에 상관없이 출력 코드=111, EO=H
- TI=L, 엔코딩 입력의 유효 유무에 따라 GS와 EO 결정



















# EX\_3\_3. 8 X 3 ENCODER 시뮬레이션





# EX\_3\_3. 8 X 3 ENCODER 핀 맵

	To	Location	I/O Bank	I/O Standard	General Function
1	 A[0]	PIN_AB8	8	LVTTL	Column I/O
2	 A[1]	PIN_W8	8	LVTTL	Column I/O
3	 A[2]	PIN_AF6	8	LVTTL	Column I/O
4	 EI	PIN_Y10	8	LVTTL	Column I/O
5	 EO	PIN_AF7	8	LVTTL	Column I/O
6	 GS	PIN_AE7	8	LVTTL	Column I/O
7	 I[0]	PIN_Y12	8	LVTTL	Column I/O
8	 I[1]	PIN_AE11	8	LVTTL	Column I/O
9	 I[2]	PIN_U12	8	LVTTL	Column I/O
10	 I[3]	PIN_AC12	8	LVTTL	Column I/O
11	 I[4]	PIN_AD12	8	LVTTL	Column I/O
12	 I[5]	PIN_AA12	8	LVTTL	Column I/O
13	 I[6]	PIN_AB12	8	LVTTL	Column I/O
14	 I[7]	PIN_Y13	7	LVTTL	Column I/O
15	<<new>>	<<new>>			



# EX\_3\_3. 8 X 3 ENCODER 프로그래밍

EX3\_3.vhd | EX3\_3.vwf | Simulator Tool | Simulation Report - ... | Assignment Editor | Compilation Report - ... | EX3\_3.cdf

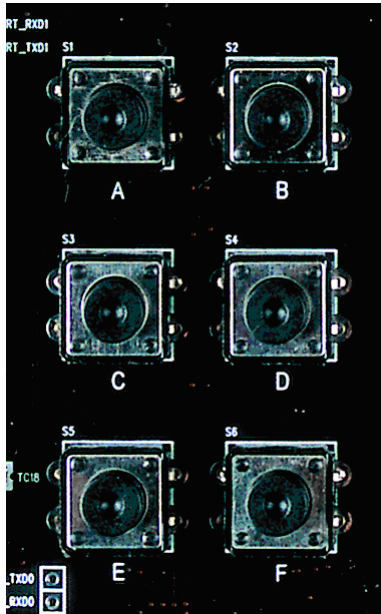
Hardware Setup... | ByteBlasterMV [LPT1] | Mode: JTAG | Progress: 100 %

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
EX3_3.sof	EP2C35F672	002E48D5	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Start | Stop | Auto Detect | Delete | Add File... | Change File... | Save File... | Add Device... | Up | Down



# EX\_3\_3. 8 X 3 ENCODER



## ➤ 입 출력 포트 관계

### ■ 버스 스위치1 : 입력

-I(0):1(Y12), I(1):2(AE11), I(2):3(U12),  
I(3):4(AC12), I(4):5(AD12), I(5):6(AA12),  
I(6):7(AB12), I(7):8(Y13)

### ■ 스위치 : 입력

- EI : A(Y10)

### ■ LED : 출력

-EO:LED1(AF7), GS:LED2(AE7),A(0):LED3(AB8)  
A(1): LED4(W8), A(2): LED5(AF6)





# Contents

- EX\_3\_3. 8 X 3 ENCODER
- EX\_4\_1. 채널 선택회로(Multiplexer)
- EX\_4\_2. 채널 분배회로



# EX\_4\_1. 채널 선택회로 기본 이론

선택신호		입력				출력
S(0)	S(1)	D(0)	D(1)	D(2)	D(3)	Y
0	0	0	X	X	X	0
0	0	1	X	X	X	1
0	1	X	0	X	X	0
0	1	X	1	X	X	1
1	0	X	X	0	X	0
1	0	X	X	1	X	1
1	1	X	X	X	0	0
1	1	X	X	X	1	1

멀티플렉서(multiplexer)

- n개의 선택입력에 따라  $2^n$ 개의 입력 중 하나를 선택하여 출력으로 연결시켜 주는 조합 회로
- 데이터 선택기(data selector)라고도 하며, 줄여서 MUX라 한다.



# EX\_4\_1. 채널 선택회로구문

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY EX_4_1A IS
PORT(
  D0 : IN STD_LOGIC;
  D1 : IN STD_LOGIC;
  D2 : IN STD_LOGIC;
  D3 : IN STD_LOGIC;
  S  : IN STD_LOGIC_VECTOR(1 downto 0);
  Y  : OUT STD_LOGIC
);
END EX_4_1A;

ARCHITECTURE HB OF EX_4_1A IS
BEGIN
PROCESS(D0,D1,D2,D3, S)
BEGIN
Y <= '0';

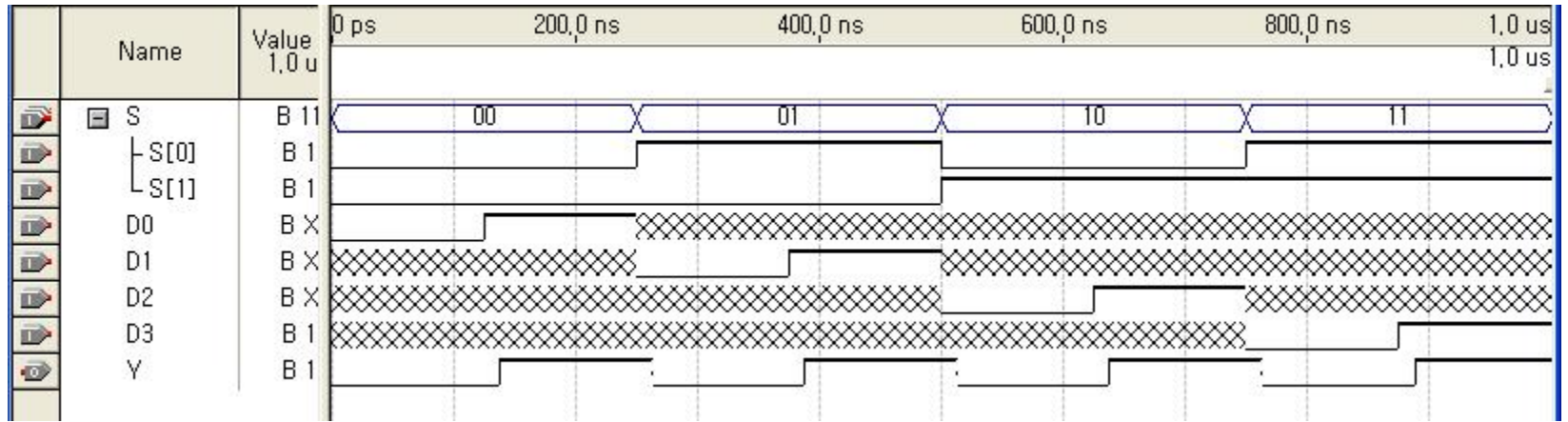
  IF S = "00" THEN
    Y <= D0;
  ELSIF S = "01" THEN
    Y <= D1;
  ELSIF S = "10" THEN
    Y <= D2;
  ELSE
    Y <= D3;
  END IF;
END PROCESS;

END HB;
```

- D0~3 : 입력 비트
- S(0), S(1) : 선택 비트
- Y : 출력 비트



# EX\_4\_1. 채널 선택회로 시뮬레이션







# EX\_4\_1. 채널 선택회로 핀 맵

	To	Location	I/O Bank
1	D0	PIN_Y10	8
2	D1	PIN_W10	8
3	D2	PIN_AA9	8
4	D3	PIN_V9	8
5	S[0]	PIN_Y13	7
6	S[1]	PIN_AB12	8
7	Y	PIN_AF7	8
8	<<new>>	<<new>>	



# EX\_4\_1. 채널 선택회로 프로그래밍

Simulator Tool | EX\_4\_1A.vhd | EX\_4\_1A.vwf | Assignment Editor | EX\_4\_1A.cdf

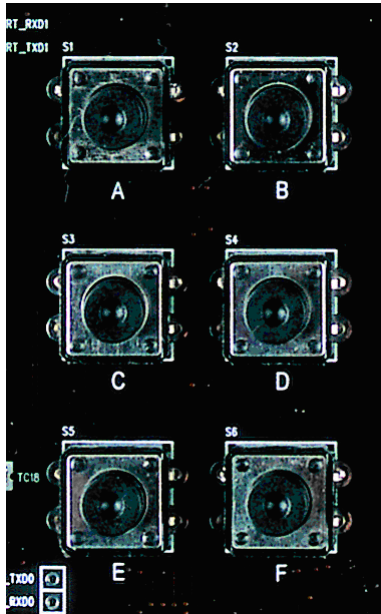
Hardware Setup... | ByteBlasterMV [LPT1] | Mode: JTAG | Progress: 100%

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
EX_4_1A.sof	EP2C35F672	002E2E83	FFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Start | Stop | Auto Detect | Delete | Add File... | Change File... | Save File... | Add Device... | Up | Down



# EX\_4\_1. 채널 선택회로



- 입 출력 포트 관계
  - 버스 스위치1 : 입력
    - S(1):SW7(AB12), S(0):SW8(Y13)
  - 스위치 : 입력
    - D(0):A(Y10), D(1):B(W10), D(2):C(AA9)
    - D(3):D(V9)
  - LED : 출력
    - Y : LED1(AF7)





# Contents

- EX\_3\_3. 8 X 3 ENCODER
- EX\_4\_1. 채널 선택회로
- EX\_4\_2. 채널 분배회로(Demultiplexer)



## EX\_4\_2. 채널 분배회로 기본 이론

선택신호		입력	출력			
S(0)	S(1)	I	Y(0)	Y(1)	Y(2)	Y(3)
X	X	0	0	X	X	X
0	0	1	1	X	X	X
0	1	1	X	1	X	X
1	0	1	X	X	1	X
1	1	1	X	X	X	1

DUMUX는 하나의 데이터 입력이 있고 제어입력 (control input)들의 값에 따라 출력 중의 하나에 데이터 입력이 연결된다. 디코더와 디멀티플렉서의 논리 회로는 모두 근본적으로 동일하다. 디멀티플렉서의 데이터 입력은 디코더에서 인에이블 입력이다.



## EX\_4\_2. 채널 분배회로 구문

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY EX_4_2B IS
PORT(
    I : IN STD_LOGIC;
    S : IN STD_LOGIC_VECTOR(0 TO 1);
    Y : OUT STD_LOGIC_VECTOR(0 TO 3)
);
END EX_4_2B;

ARCHITECTURE HB OF EX_4_2B IS
BEGIN

    PROCESS(I, S)
    BEGIN
        Y <= "0000";
        CASE S IS
            WHEN "00" => Y(0) <= I;
            WHEN "01" => Y(1) <= I;
            WHEN "10" => Y(2) <= I;
            WHEN "11" => Y(3) <= I;
            WHEN OTHERS => NULL;
        END CASE;
    END PROCESS;
END HB;
```

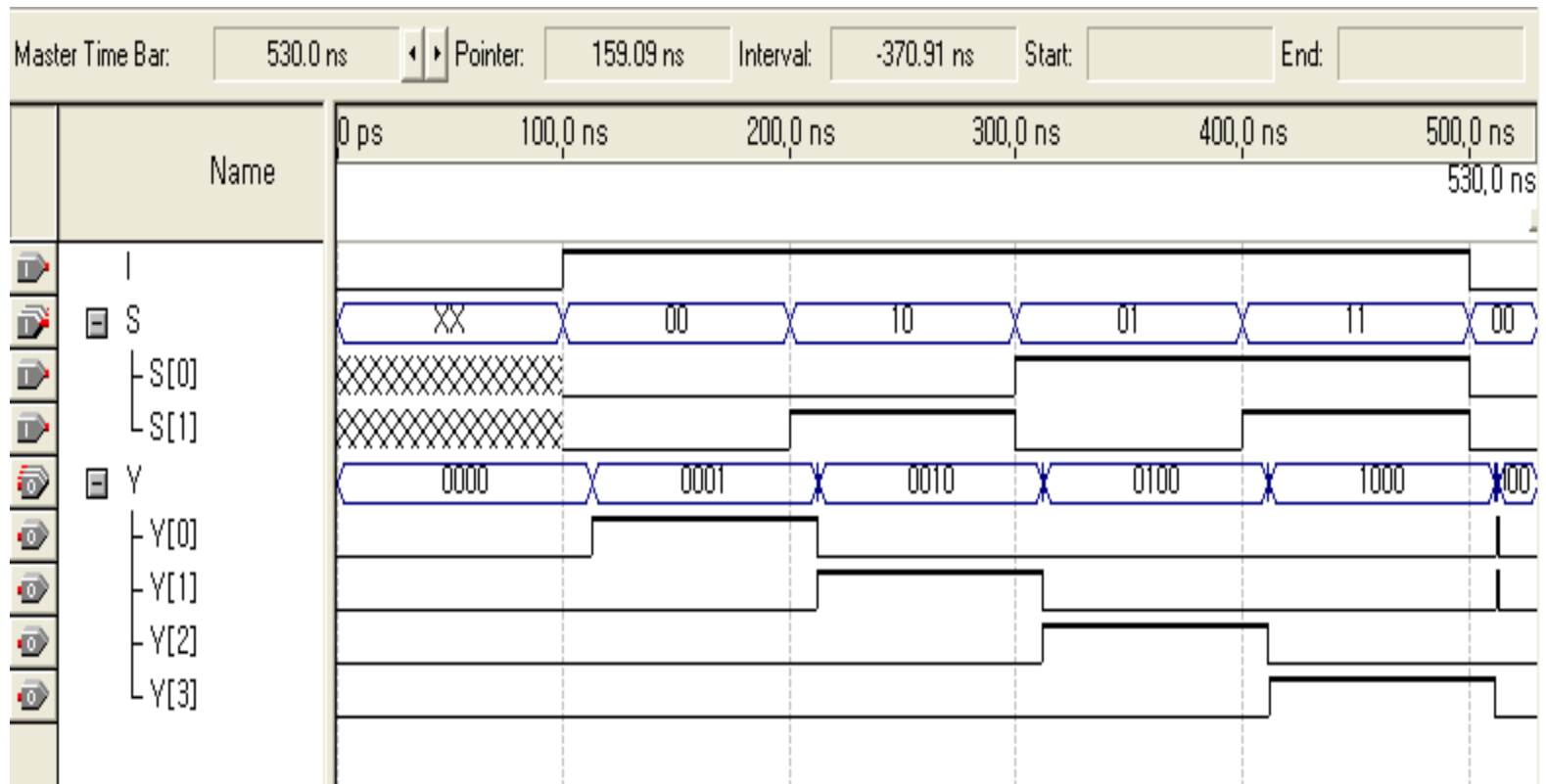
■ I : 입력 비트

■ S : 선택 비트

■ Y : 출력 비트



# EX\_4\_2. 채널 분배회로 시뮬레이션





# EX\_4\_2. 채널 분배회로 핀 맵

	To	Location	I/O Bank
1	Y[0]	PIN_AF7	8
2	Y[1]	PIN_AE7	8
3	Y[2]	PIN_AB8	8
4	Y[3]	PIN_W8	8
5	S[0]	PIN_Y10	8
6	S[1]	PIN_W10	8
7	I	PIN_Y13	7
8	<<new>>	<<new>>	





# EX\_4\_2. 채널 분배회로 프로그래밍

EX\_4\_2B.vhd | EX\_4\_2B.vwf | Simulator Tool | Simulation Report - Simul... | Assignment Editor | EX\_4\_2B.cdf

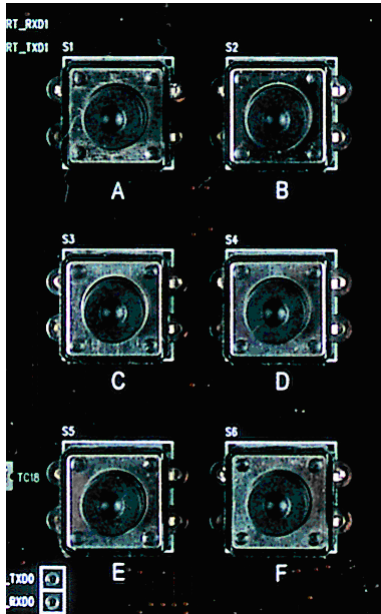
Hardware Setup... | ByteBlasterMV [LPT1] | Mode: JTAG | Progress: 100%

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
EX_4_2B.sof	EP2C35F672	002E3595	FFFFFFFF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Start | Stop | Auto Detect | Delete | Add File... | Change File... | Save File... | Add Device... | Up | Down



# EX\_4\_2. 채널 분배회로



- 입 출력 포트 관계
  - 버스 스위치1 : 입력
    - I : SW8(Y13)
  - 스위치 : 입력
    - S(0) : A(Y10), S(1) : B(W10)
  - LED : 출력
    - Y(0):LED1(AF7), Y(1):LED2(AE7),  
Y(2):LED3(AB8), Y(3):LED4(W8)

