EasySoC-Z7010 FPGA/SoC 설계 플랫폼

5. FPGA를 이용한 Text-LCD 제어 실습

개요

- Text-LCD 컨트롤러를 HDL로 설계한다.
  - 로직 설계 시 직면하게 되는 데이터시트의 타이밍 다이어그램을 이해.
  - Zynq7010의 FPGA를 이용하여 외부 디바이스를 제어.
  - 단순 논리에서 한 단계 높은 수준의 실습을 경험.

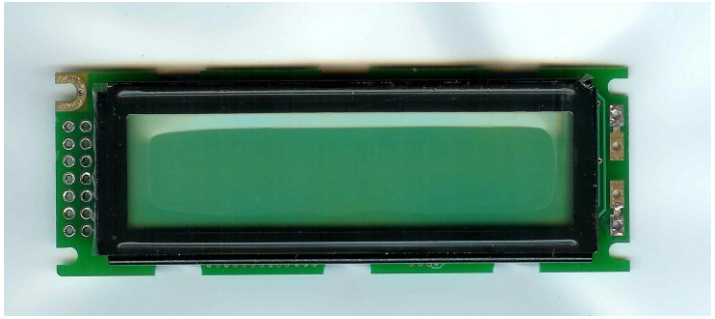
HUINS

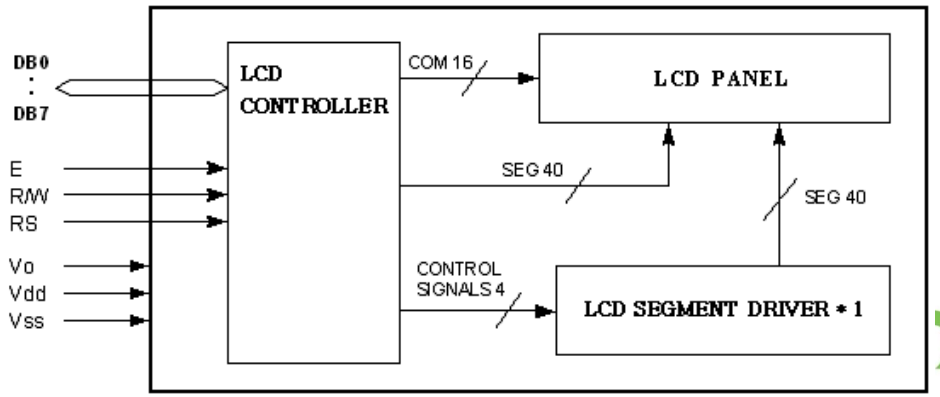
Text-LCD 구성 및 동작 설명

- LCD : 액정 디스플레이(Liquid Crystal Display)의 약자.
  - 전자계산기, 전자 시계 등 전자기기의 디스플레이 장치로 사용된다.
  - 원리
    - ✓ 액정은 전압이 걸리면 일정한 방향으로 늘어서는 성질이 있음.
    - ✓ 이 성질과 전기석에 의한 편광 성질을 이용 – 전압이 걸린 곳이 검게 표시됨.
  - 장점 : 사용하기 편하며 전력소모가 매우 적음.
  - 단점 : 동작속도가 느려 잔상이 발생할 수 있음.

Text-LCD 구성 및 동작 설명

- Text-LCD in EasySoC-Z7010 : LCD 패널과 제어기가 하나로 되어 있는 모듈.
  - 제어기에 데이터 버스 등을 통하여 데이터를 전송하기만 하면 원하는 표시를 얻을 수 있음.
    - ✓ LCD 제어기 : DB0~DB7의 8bit 데이터 버스를 통해 데이터를 주고받음.
      - RS : 명령어를 입출력과 내부의 메모리의 입출력 중 하나를 선택하기 위해 사용되는 핀.
      - R/W : 데이터 읽기와 데이터 쓰기를 선택하는 데 사용.
      - E : LCD에 제어 명령이나 데이터를 쓰거나 읽는 등의 동작을 제어함.
    - ✓ Vdd와 Vss 사이에 +5V 전원을 인가하면 모듈이 동작.





Text-LCD 구성 및 동작 설명

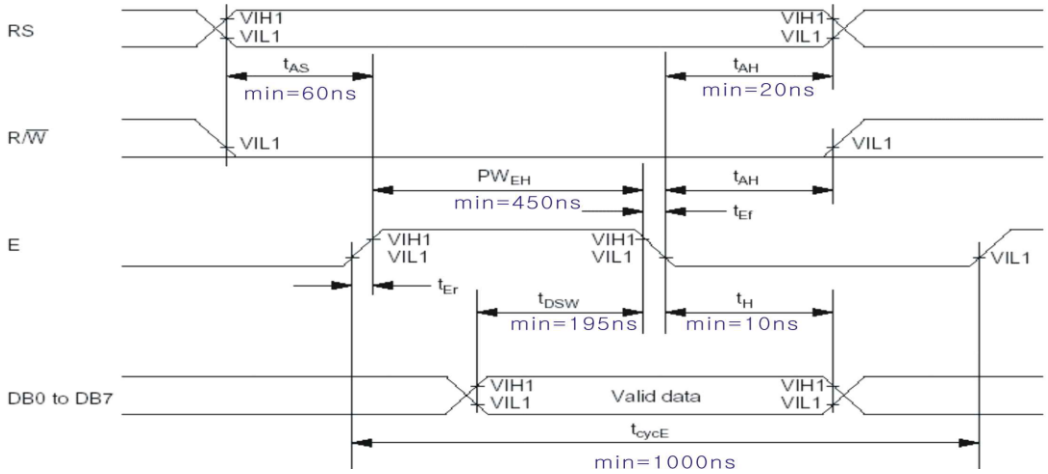
- LCD를 사용하기 위한 제어 신호와 그 기능
  - ① 화면을 지움.
  - ② 커서를 좌측 상단의 위치(Home)로 옮김.
  - ③ 출력되는 글자가 한 글자씩 뒤로 갈 것인가 앞으로 갈 것인가를 결정.
  - ④ 화면의 ON/OFF, 커서의 ON/OFF, 커서의 깜빡임 ON/OFF를 결정.
  - ⑤ LCD에 출력할 문자를 계속 전송할 경우 방향을 결정.
  - ⑥ LCD의 제어 설정. 4bit/8bit 데이터 설정과 폰트 크기 등을 결정.
  - ⑦ LCD에 표시할 위치에 해당하는 어드레스를 지정.

번호	기 능	코 드									
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	디스플레이 클리어	0	0	0	0	0	0	0	0	0	1
2	Home으로 돌아옴	0	0	0	0	0	0	0	0	1	×
3	입력 모드 설정	0	0	0	0	0	0	0	1	I/D	S
4	디스플레이 ON/OFF	0	0	0	0	0	0	1	D	C	B
5	커서/시프트	0	0	0	0	0	1	S/C	R/L	×	×
6	시스템 설정	0	0	0	0	1	IF	N	F	×	×
7	CG RAM 설정	0	0	0	1	ACG					
8	DD RAM 설정	0	0	1	ADD						
9	BF/어드레스 읽기	0	1	BF	AC						
10	CG나 DD 쓰기	1	0	출력 데이터							
11	CG나 DD 읽기	1	1	입력 데이터							

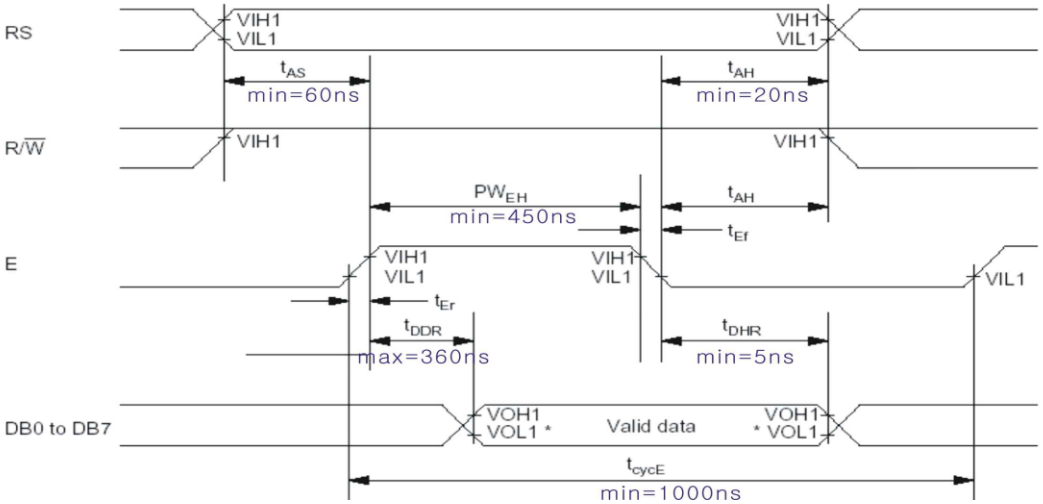
HUINS

Text-LCD 동작 방법

- Character-LCD의 제어코드를 Read/Write 하기 위해서 일정한 동작 타이밍에 신호를 주어야 한다.



Character-LCD Write 동작 타이밍

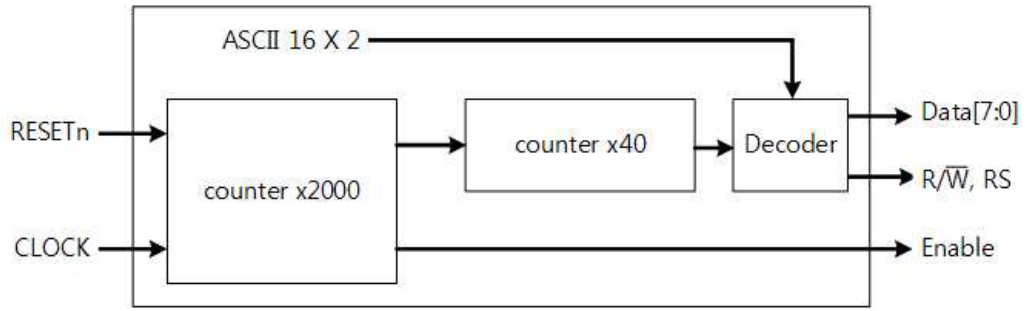


Character-LCD Read 동작 타이밍

HUINS

Text-LCD에 ASCII 문자 출력

- Text-LCD 구동 회로 블록 다이어그램
  - RESETn : Low : 전체 로직이 Reset. / High : Reset이 해제되어 회로가 동작함.
  - Clock : 25MHz 입력.
  - R/W : constant 0으로 출력, Text-LCD는 항상 쓰기모드로 동작함.
  - Counter x2000 : CLOCK Rising Edge마다 1씩 증가시키는 counter.
    - ✓ 출력 값을 0부터 1999까지 계속 반복.
  - Counter x40 : counter x2000의 출력 값을 받아 1씩 증가시키는 counter.
    - ✓ Counter x40의 출력 값은 처음에만 0~40으로 증가, 두 번째부터 반복할 때는 6~40으로 무한 반복함.
  - Decoder : counter x40의 출력 값에 따라 Text-LCD의 data로 값을 출력함.

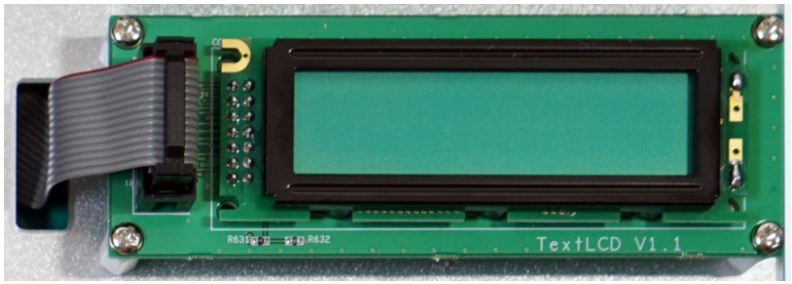


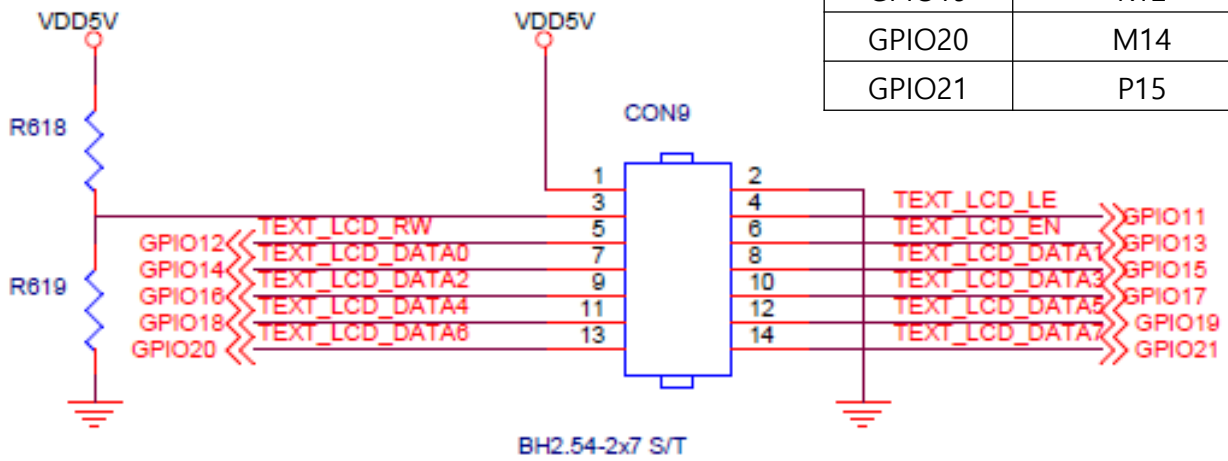
HUINS

Text-LCD 회로도(Ext. Module)

- Text-LCD 회로도

GPIO	Package Pin	External Module
GPIO11	J15	text_le
GPIO12	M15	text_rw
GPIO13	R13	text_en
GPIO14	M12	text_data0
GPIO15	N13	text_data1
GPIO16	L13	text_data2
GPIO17	G11	text_data3
GPIO18	H11	text_data4
GPIO19	R12	text_data5
GPIO20	M14	text_data6
GPIO21	P15	text_data7

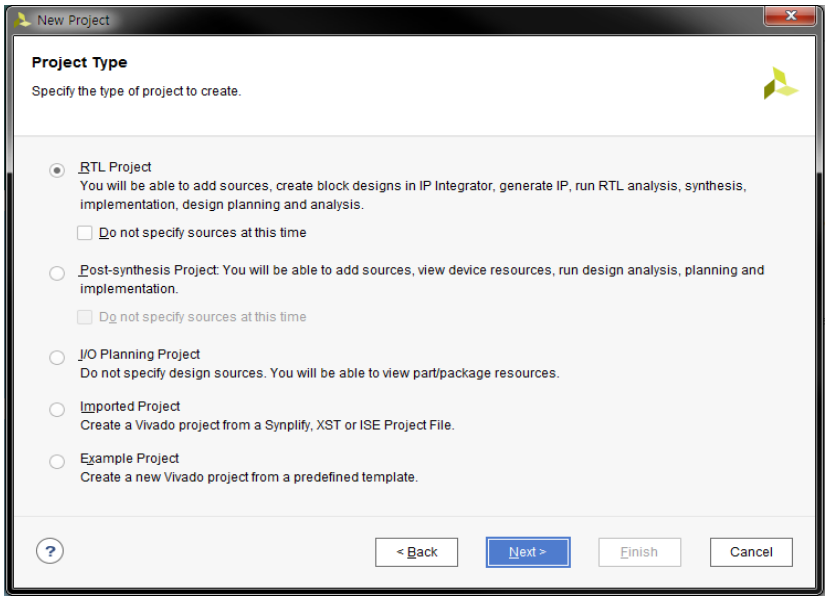


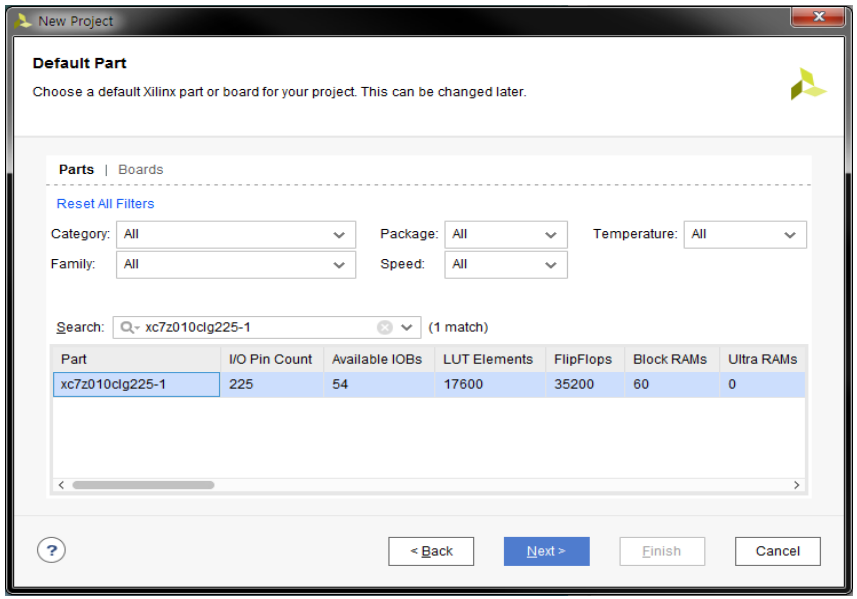


HUINS

Text-LCD 컨트롤러 설계 프로젝트 생성하기

- 프로젝트 생성
  - 이름과 경로를 지정하여 새 프로젝트를 생성한다.
  - 프로젝트의 타입은 RTL Project로, Xilinx part는 xc7z010clg225-1로 설정한다.

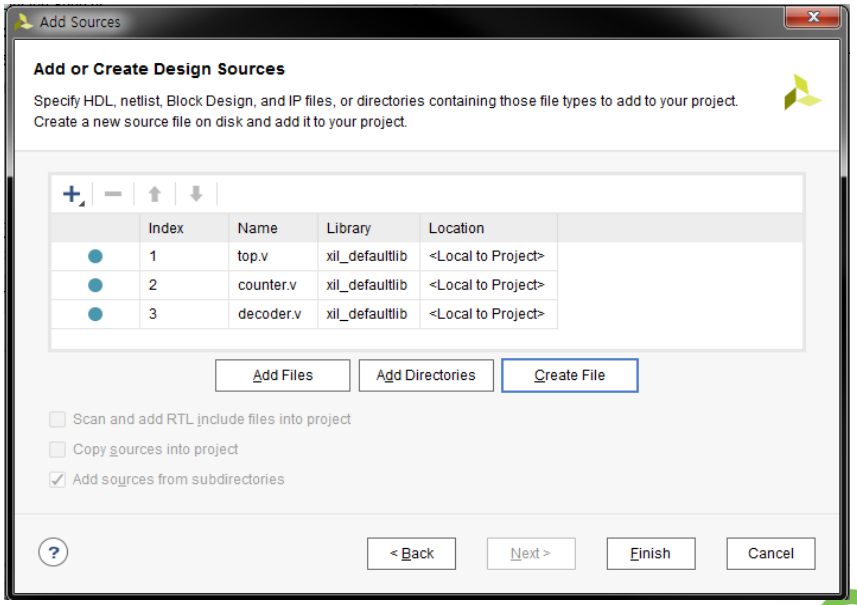




HUINS

Text-LCD에 ASCII 문자 출력

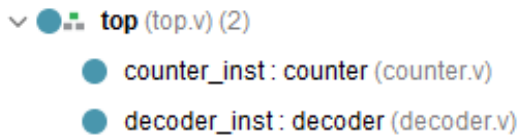
- Text-LCD 구동 회로를 설계하기 위한 RTL 코드 파일 생성하기
  - Project Manager – Add sources에서 "Add or Create Design Sources"를 선택한다.
  - Create Files 버튼을 클릭해 RTL 코드를 추가한다.
    - ✓ top.v, counter.v, decoder.v



HUINS

소스코드 작성하기

- 소스코드는 총 3개의 파일로 구성되어 있다.



✓ top.v 작성하기(top 모듈)

```
`timescale 1ns / 1ps
module top(
    input  wire clk,
    input  wire reset_n,
    output wire lcd_rs,
    output wire lcd_rw,
    output wire lcd_en,
    output wire [7:0] lcd_data
);

wire [7:0] count_lcd;

counter counter_inst (
    .clk(clk),
    .reset_n(reset_n),
    .lcd_en(lcd_en),
    .count_lcd(count_lcd)
);
```

```
decoder decoder_inst (
    .clk(clk),
    .reset_n(reset_n),
    .count_lcd(count_lcd),
    .lcd_rs(lcd_rs),
    .lcd_rw(lcd_rw),
    .lcd_data(lcd_data)
);

endmodule
```

HUINS

소스코드 작성하기

- counter.v 코드 작성

```
`timescale 1ns / 1ps

module counter(
    input clk,
    input reset_n,
    output lcd_en,
    output reg [7:0] count_lcd
    );
reg [11:0] delay_lcdclk;
wire tc;

// Counterx2000 & //Counterx40
    always @(posedge clk) begin
        if (reset_n == 0) begin
            delay_lcdclk <= 0;
        end
        else begin
            // Counterx2000
            if (delay_lcdclk < 1999)
                delay_lcdclk <=  delay_lcdclk + 1;
            else
                delay_lcdclk <= 0;
        end
    end
```

```
// Counterx40
    always @(posedge clk) begin
        if (reset_n == 0) begin
            count_lcd <= 0;
        end
        else if(tc) begin
            if (count_lcd < 40)
                count_lcd <= count_lcd + 1;
            else
                count_lcd <= 6;
            end
    end

    assign lcd_en = (delay_lcdclk >= 1600) ? 1:0;
    assign tc = (delay_lcdclk == 0)? 1:0;

endmodule
```

HUINS

소스코드 작성하기

- decoder.v 코드 작성

```
`timescale 1ns / 1ps

module decoder(
    input clk,
    input reset_n,
    input [7:0] count_lcd,
    output lcd_rs,
    output lcd_rw,
    output [7:0] lcd_data
);

localparam lcd_blank = 8'b00100000, lcd_dash = 8'b00101101,lcd_colon = 8'b00111010,
lcd_comma = 8'b00101100, lcd_Dot = 8'b00101110, lcd_0 = 8'b00110000, lcd_1 = 8'b00110001,
lcd_2 = 8'b00110010, lcd_3 = 8'b00110011, lcd_4 = 8'b00110100, lcd_5 = 8'b00110101,
lcd_6 = 8'b00110110, lcd_7 = 8'b00110111, lcd_8 = 8'b00111000, lcd_9 = 8'b00111001,
lcd_a = 8'b01000001, lcd_b = 8'b01000010, lcd_c = 8'b01000011, lcd_d = 8'b01000100,
lcd_e = 8'b01000101, lcd_f = 8'b01000110, lcd_g = 8'b01000111, lcd_h = 8'b01001000,
lcd_i = 8'b01001001, lcd_j = 8'b01001010, lcd_k = 8'b01001011, lcd_l = 8'b01001100,
lcd_m = 8'b01001101, lcd_n = 8'b01001110, lcd_o = 8'b01001111, lcd_p = 8'b01010000,
lcd_q = 8'b01010001, lcd_r = 8'b01010010, lcd_s = 8'b01010011, lcd_t = 8'b01010100,
lcd_u = 8'b01010101, lcd_v = 8'b01010110, lcd_w = 8'b01010111, lcd_x = 8'b01011000,
lcd_y = 8'b01011001, lcd_w = 8'b01010111, lcd_x = 8'b01011000, lcd_y = 8'b01011001,
lcd_z = 8'b01011010, lcd_under = 8'b01011111, lcd_s_a = 8'b01100001, lcd_s_b = 8'b01100010, lcd_s_c =
8'b01100011, lcd_s_d = 8'b01100100, lcd_s_e = 8'b01100101, lcd_s_f = 8'b01100110, lcd_s_g =
8'b01100111, lcd_s_h = 8'b01101000, lcd_s_i = 8'b01101001, lcd_s_j = 8'b01101010, lcd_s_k =
8'b01101011, lcd_s_l = 8'b01101100, lcd_s_m = 8'b01101101,
```

HUINS

소스코드 작성하기

```
lcd_s_n = 8'b01101110, lcd_s_o = 8'b01101111, lcd_s_p = 8'b01110000, lcd_s_q = 8'b01110001, lcd_s_r =
8'b01110010, lcd_s_s = 8'b01110011, lcd_s_t = 8'b01110100, lcd_s_u = 8'b01110101, lcd_s_v =
8'b01110110, lcd_s_w = 8'b01110111, lcd_s_x = 8'b01111000, lcd_s_y = 8'b01111001, lcd_s_z =
8'b01111010, add_line_1 = 8'b10000000, add_line_2 = 8'b11000000,
add_line_3 = 8'b10010100, add_line_4 = 8'b11010100;

localparam MODE_PWR_ON = 4'd1,      //power on
           MODE_FN_SET = 4'd2,      //function set
           MODE_ONOFF = 4'd3,       //display on/off control
           MODE_ENTR_1 = 4'd4,
           MODE_ENTR_2 = 4'd5,
           MODE_ENTR_3 = 4'd6,
           MODE_SET_ADDR_1 = 4'd7, //set addr 1st line
           MODE_WR_1 = 4'd8,        //write 1st line
           MODE_SET_ADDR_2 = 4'd9, //set addr 2nd line
           MODE_WR_2 = 4'd10,       //wirte 2nd line
           MODE_DELAY = 4'd11,      //delay
           MODE_ACTCMD = 4'd12;     //user command

wire [31:0] reg_a;
wire [31:0] reg_b;
wire [31:0] reg_c;
wire [31:0] reg_d;
wire [31:0] reg_e;
wire [31:0] reg_f;
wire [31:0] reg_g;
wire [31:0] reg_h;

reg [4:0] lcd_mode = 0;
reg [9:0] set_data;
```

HUINS

소스코드 작성하기

```
assign reg_a = {lcd_e, lcd_s_a, lcd_s_s, lcd_s_y} ;     // Easy
assign reg_b = {lcd_s, lcd_s_o, lcd_c, lcd_blank} ;     // SoC
assign reg_c = {lcd_dash, lcd_blank, lcd_z, lcd_7} ;     // - z7
assign reg_d = {lcd_0, lcd_1, lcd_0, lcd_Dot} ;          // 010,
assign reg_e = {lcd_h, lcd_s_u, lcd_s_i, lcd_s_n} ;     // Huin
assign reg_f = {lcd_s_s, lcd_blank, lcd_c, lcd_s_o} ;     // s Co
assign reg_g = {lcd_Dot,  lcd_comma, lcd_blank, lcd_s_l} ;// ., l
assign reg_h = {lcd_s_t, lcd_s_d, lcd_Dot, lcd_blank} ;// td.

// decoder switch
always @(posedge clk) begin
    if (reset_n == 0)
        lcd_mode <= MODE_PWR_ON;
    else begin
        case (count_lcd)
            0  : lcd_mode <= MODE_PWR_ON ;
            1  : lcd_mode <= MODE_FN_SET ;
            2  : lcd_mode <= MODE_ONOFF ;
            3  : lcd_mode <= MODE_ENTR_1 ;
            4  : lcd_mode <= MODE_ENTR_2 ;
            5  : lcd_mode <= MODE_ENTR_3 ;
            6  : lcd_mode <= MODE_SET_ADDR_1 ;
            7  : lcd_mode <= MODE_WR_1 ;
            23 : lcd_mode <= MODE_SET_ADDR_2 ;
            24 : lcd_mode <= MODE_WR_2 ;
            40 : lcd_mode <= MODE_DELAY ;
            41 : lcd_mode <= MODE_ACTCMD ;
            default : begin end
        endcase
    end
end

assign lcd_rs = set_data[9];
assign lcd_rw = set_data[8];
assign lcd_data = set_data[7:0];
```

HUINS

소스코드 작성하기

```
// decoder output
always @(clk or lcd_mode or count_lcd) begin
    if (reset_n == 0)
        set_data = 10'b0000000000;
    else begin
        case (lcd_mode)
                        MODE_PWR_ON : set_data = {2'b00, 8'h38};
                        MODE_FN_SET : set_data = {2'b00, 8'h38};
                        MODE_ONOFF : set_data = {2'b00, 8'h0e};
                        MODE_ENTR_1 : set_data = {2'b00, 8'h06};
                        MODE_ENTR_2 : set_data = {2'b00, 8'h02};
                        MODE_ENTR_3 : set_data = {2'b00, 8'h01};
MODE_SET_ADDR_1 : set_data = {2'b00, 8'h80};
                        MODE_WR_1 : begin
                    case (count_lcd)
                            7 : set_data = {1'b1, 1'b0, reg_a[31:24]};
                            8 : set_data = {1'b1, 1'b0, reg_a[23:16]};
                            9 : set_data = {1'b1, 1'b0, reg_a[15: 8]};
                           10 : set_data = {1'b1, 1'b0, reg_a[7 : 0]};
                           11 : set_data = {1'b1, 1'b0, reg_b[31:24]};
                           12 : set_data = {1'b1, 1'b0, reg_b[23:16]};
                           13 : set_data = {1'b1, 1'b0, reg_b[15: 8]};
                           14 : set_data = {1'b1, 1'b0, reg_b[7 : 0]};
                           15 : set_data = {1'b1, 1'b0, reg_c[31:24]};
                           16 : set_data = {1'b1, 1'b0, reg_c[23:16]};
                           17 : set_data = {1'b1, 1'b0, reg_c[15: 8]};
                           18 : set_data = {1'b1, 1'b0, reg_c[7 : 0]};
                           19 : set_data = {1'b1, 1'b0, reg_d[31:24]};
                           20 : set_data = {1'b1, 1'b0, reg_d[23:16]};
                           21 : set_data = {1'b1, 1'b0, reg_d[15: 8]};
                           22 : set_data = {1'b1, 1'b0, reg_d[7 : 0]};
                    endcase
                end
```

HUINS

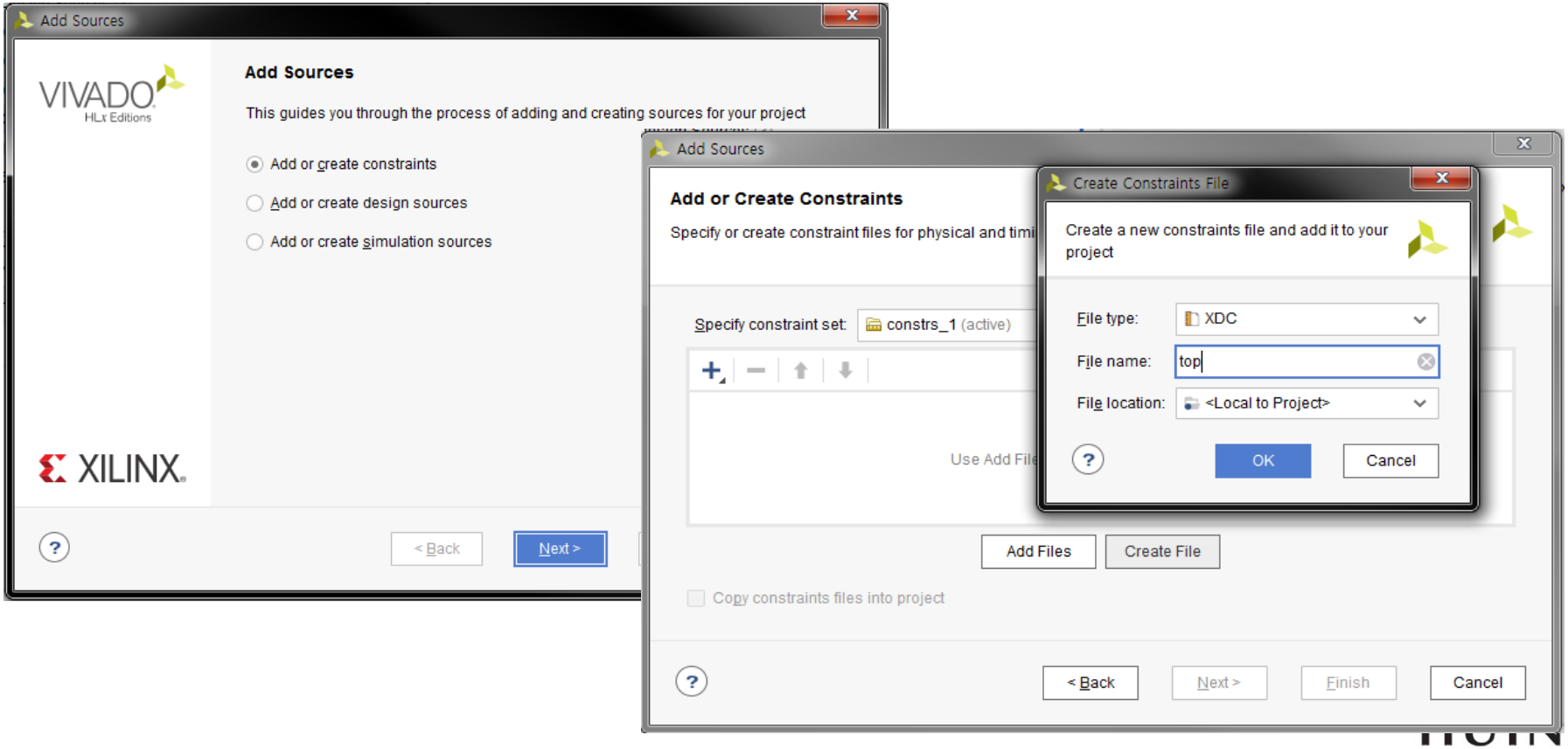
소스코드 작성하기

```
                    MODE_SET_ADDR_2 : set_data = {2'b00, 8'hc0};
                    MODE_WR_2 : begin
                  case (count_lcd)
                          24 : set_data = {1'b1, 1'b0, reg_e[31:24]};
                          25 : set_data = {1'b1, 1'b0, reg_e[23:16]};
                          26 : set_data = {1'b1, 1'b0, reg_e[15: 8]};
                          27 : set_data = {1'b1, 1'b0, reg_e[7 : 0]};
                          28 : set_data = {1'b1, 1'b0, reg_f[31:24]};
                          29 : set_data = {1'b1, 1'b0, reg_f[23:16]};
                          30 : set_data = {1'b1, 1'b0, reg_f[15: 8]};
                          31 : set_data = {1'b1, 1'b0, reg_f[7 : 0]};
                          32 : set_data = {1'b1, 1'b0, reg_g[31:24]};
                          33 : set_data = {1'b1, 1'b0, reg_g[23:16]};
                          34 : set_data = {1'b1, 1'b0, reg_g[15: 8]};
                          35 : set_data = {1'b1, 1'b0, reg_g[7 : 0]};
                          36 : set_data = {1'b1, 1'b0, reg_h[31:24]};
                          37 : set_data = {1'b1, 1'b0, reg_h[23:16]};
                          38 : set_data = {1'b1, 1'b0, reg_h[15: 8]};
                          39 : set_data = {1'b1, 1'b0, reg_h[7 : 0]};
                     endcase
                    end
                    MODE_DELAY : set_data = {2'b00, 8'h02};
                    MODE_ACTCMD : set_data = {2'b00, 8'h02};
                    default : begin end
                  endcase
    end
end

endmodule
```

HUINS

프로젝트 컴파일 하기

- Constraints 설정
  - Project Manager - Add Sources에서 "Add or create constrains" 항목을 선택한다.
  - Create File을 선택해 top.xdc 파일을 생성한다.



프로젝트 컴파일 하기

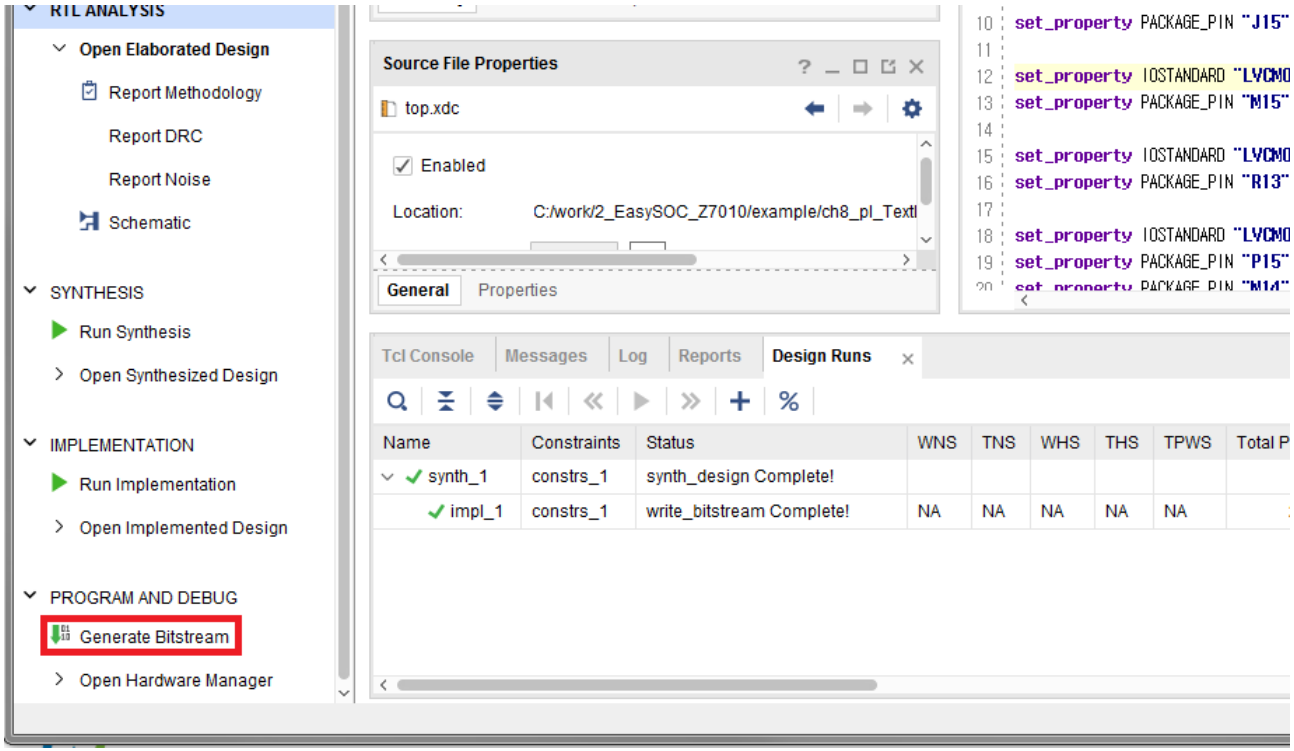
- top.xdc 파일의 내용을 다음과 같이 작성한다.

```
set_property IOSTANDARD "LVCMOS33"       [get_ports "clk"]
set_property PACKAGE_PIN "K15"           [get_ports "clk"]

set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets "clk"]

set_property IOSTANDARD "LVCMOS33"   [get_ports "reset_n"]
set_property PACKAGE_PIN "G12"       [get_ports "reset_n"]

set_property IOSTANDARD "LVCMOS33"   [get_ports "lcd_rs"]
set_property PACKAGE_PIN "J15"       [get_ports "lcd_rs"]

set_property IOSTANDARD "LVCMOS33"   [get_ports "lcd_rw"]
set_property PACKAGE_PIN "M15"       [get_ports "lcd_rw"]

set_property IOSTANDARD "LVCMOS33"   [get_ports "lcd_en"]
set_property PACKAGE_PIN "R13"       [get_ports "lcd_en"]

set_property IOSTANDARD "LVCMOS33"   [get_ports "lcd_data[*]"]
set_property PACKAGE_PIN "P15"       [get_ports "lcd_data[7]"]
set_property PACKAGE_PIN "M14"       [get_ports "lcd_data[6]"]
set_property PACKAGE_PIN "R12"       [get_ports "lcd_data[5]"]
set_property PACKAGE_PIN "H11"       [get_ports "lcd_data[4]"]
set_property PACKAGE_PIN "G11"       [get_ports "lcd_data[3]"]
set_property PACKAGE_PIN "L13"       [get_ports "lcd_data[2]"]
set_property PACKAGE_PIN "N13"       [get_ports "lcd_data[1]"]
set_property PACKAGE_PIN "M12"       [get_ports "lcd_data[0]"]
```

HUINS

프로젝트 컴파일 하기

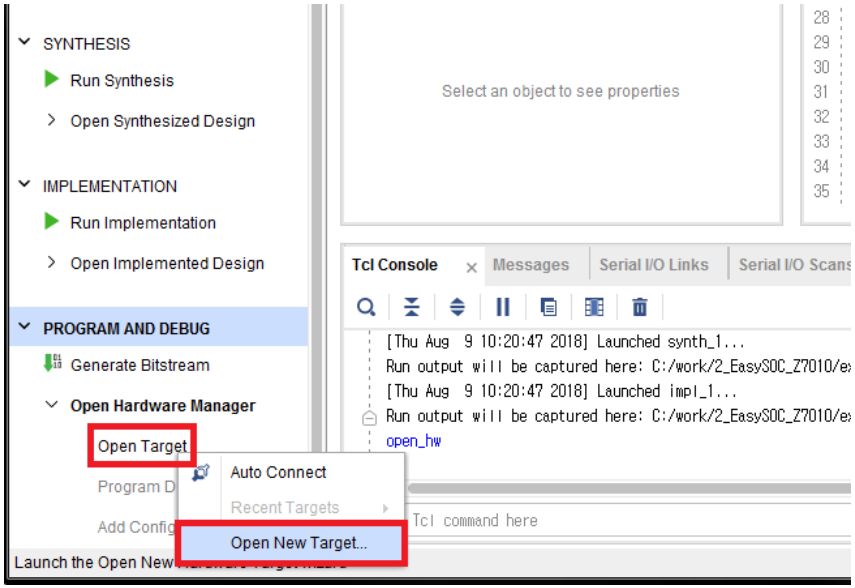
- 프로젝트 컴파일 하기
  - PROGRAM AND DEBUG – Generate Bitstream을 실행하여 PL영역에 프로그램 할 Bitstream파일을 생성한다.
  - Generate Bitstream을 실행할 경우 순서대로 Synthesis, Implementation, Generate Bitstream 순으로 진행이 된다.

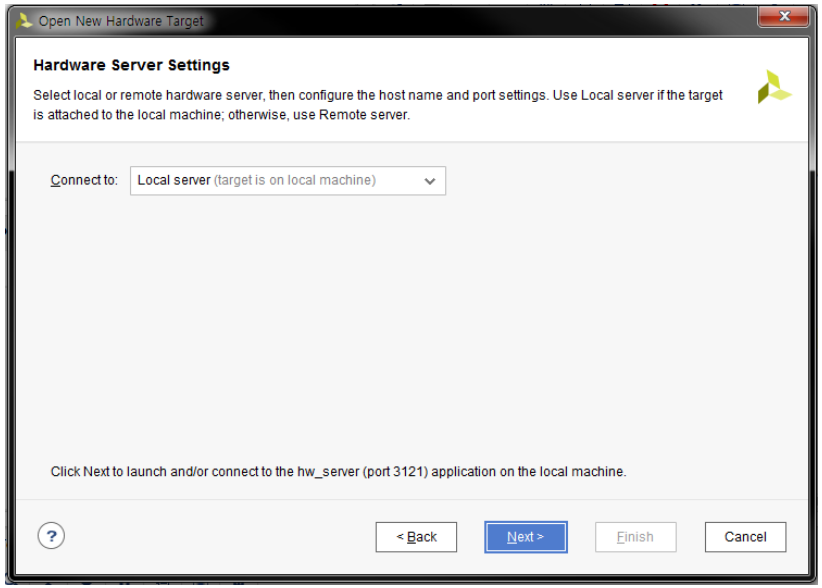


HUINS

EasySoC-Z7010에서 검증하기

- Host에 타겟 연결하기
  - PROGRAM AND DEBUG – Open Hardware Manger – Open Target에서 Open New Target을 클릭한다.
  - PC에 보드를 연결한 뒤 Local server로 연결한다.

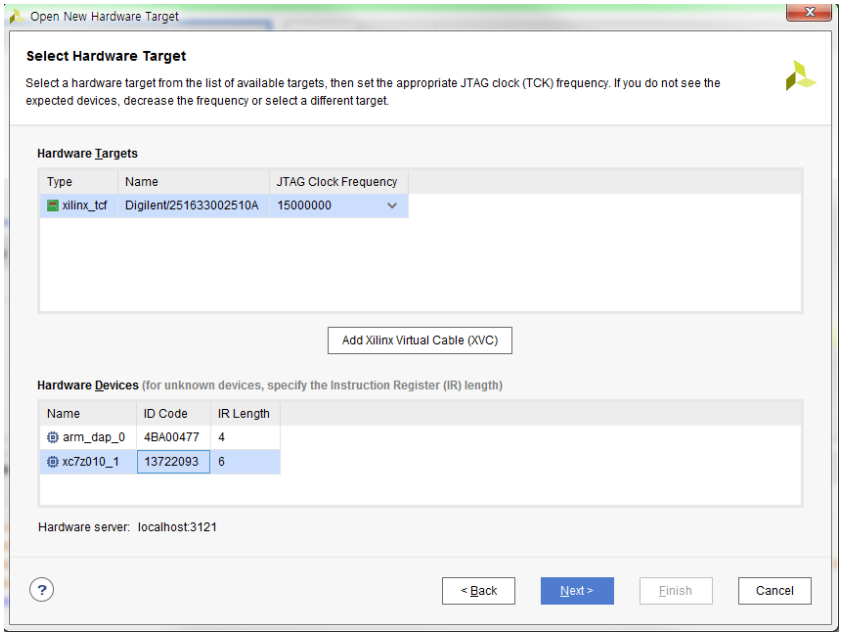


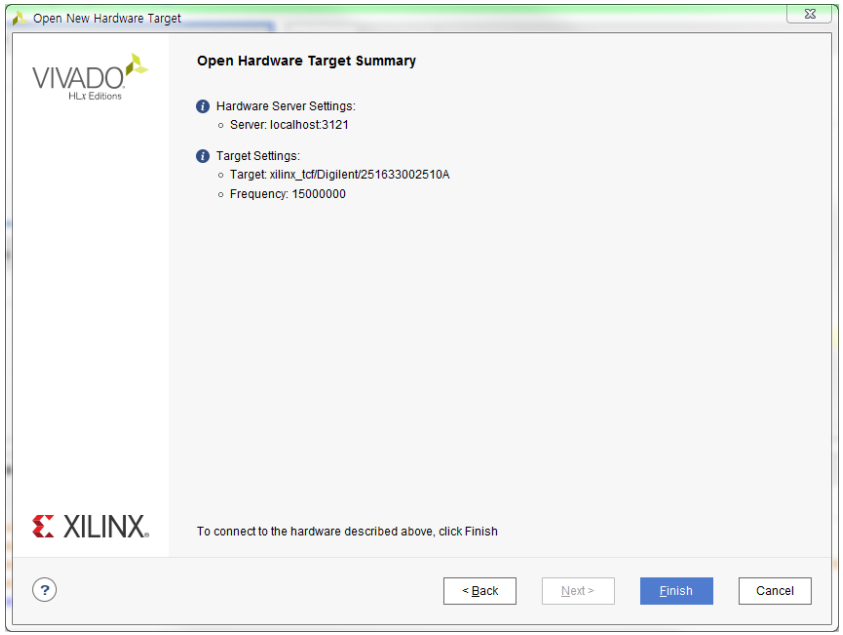


HUINS

EasySoC-Z7010에서 검증하기

- Host에 타겟 연결하기
  - Hardware Devices에 XC7Z010_1을 선택하고 next 버튼을 클릭한다.
  - ZYNQ-7000 PL에 Bitstream 파일을 다운로드 하기 위한 설정 내용을 알려주는 대화상자에서 Finish버튼을 클릭해 연결을 완료한다.

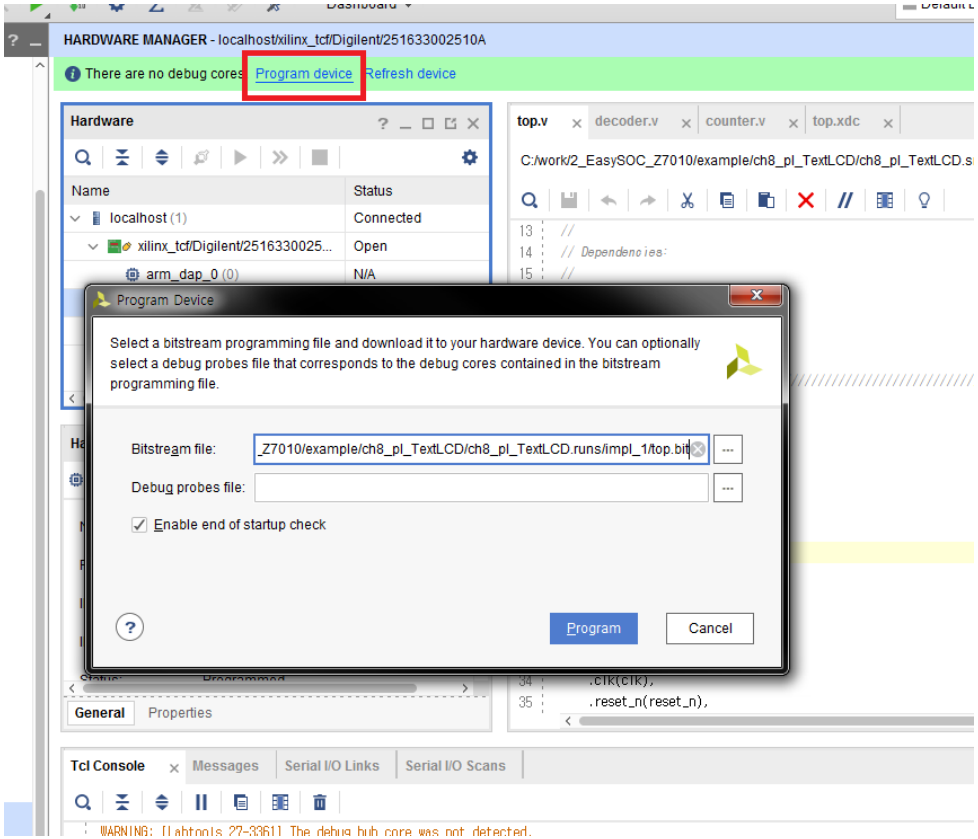




HUINS

EasySoC-Z7010에서 검증하기

- HARDWARE MANAGER에서 Program Device를 실행한다.
- Z7010의 PL 영역에 다운로드 할 Bitstream file을 선택하고 Program한다.



EasySoC-Z7010에서 검증하기

- 다운로드가 완료되면, Text-LCD가 동작한다.
  - ✓ Text-LCD는 다음과 같은 메시지를 출력한다.

EasySoC – Z7010
Huins Co., ltd.

HUINS

감사합니다.

HUINS