# BAB IV
# CODING WITH CLOCKING-FUNCTION

## 3.1 Tujuan

1. Praktikan memahami konsep Clocking function.
2. Praktikan dapat mengimplementasikan clocking function pada papan FPGA Nexys A7.

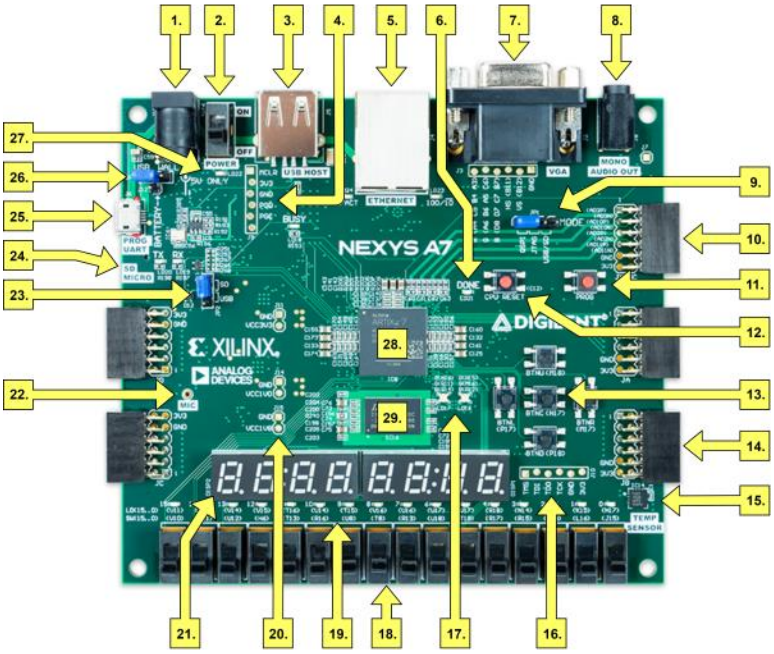
## 3.2 Alat dan Bahan

1. Laptop
2. Papan Nexys A7
3. Seven Segment
4. Switch

## 3.1 Dasar Teori

1. Nesxys A7

Nexys A7 merupakan platform papan pengembang sirkuit digital yang lengkap dan siap digunakan. Papan ini berbasis pada Artix-7™ Field Programmable Fate Array (FPGA) dari Xilinx®. Nexys A7 dapat menampung desain mulai dari sirkuit kombinasi pengantar hingga hingga prosesor tertanam dengan dukungan FPGA yang berkapasitas tinggi, memori eksternal yang besar, dan koleksi port seperti USB dan Ehternet. Beberapa perangkat bawaan seperti accelerometer, sensor suhu, mikrofon digital MEMs, amplifier speaker dan beberapa perangkat I/O memungkinkan Nexys A7 digunakan untuk berbagai desain tanpa memerlukan komponen lain.



| No | Komponen | No | Komponen |
|---|---|---|---|
| **1.** | Power jack | **16.** | JTAG port for (optional) external cable |
| **2.** | Power switch | **17.** | RGB LEDs |
| **3.** | USB host connector | **18.** | Slide switches (16) |
| **4.** | PIC24 programming port | **19.** | LEDs (16) |
| **5.** | Ethernet connector | **20.** | Power supply test point(s) |
| **6.** | FPGA Programming done LED | **21.** | 8 digit 7-seg display |
| **7.** | VGA connector | **22.** | Microphone |
| **8.** | Audio connector | **23.** | External configuration jumper (SD/USB) |
| **9.** | Programming mode jumper | **24.** | MicroSD card slot |
| **10.** | Analog signal Pmod port (XDAC) | **25.** | Shared UART/JTAG USB port |
| **11.** | FPGA configuration reset button | **26.** | Power select jumper and battery header |
| **12.** | CPU reset button | **27.** | Power-good LED |
| **13.** | Five pushbuttons | **28.** | Xilinx Artix-7 FPGA |
| **14.** | Pmod port | **29.** | DDR2 memory |
| **15.** | Temperature sensor | | |

(sumber : https://digilent.com/reference/programmable-logic/nexys-a7/reference-manual/)

2. Clocking (detak)

Clocking merupakan sinyal dalam sirkuit digital yang menentukan seberapa cepat flip flop berjalan. Sinyal clock terhubung ke semua flipflop dan blok RAM untuk mengaktifkannya sesuai frekuensi clock yang diatur. Semakin cepat clock, semakin cepat juga desain program berjalan. Oleh karena itu, FPGA dengan kecepatan clock yang tinggi akan melakukan fungsi yang diinginkan lebih cepat. FPGA biasanya terdiri dari bebrapa sinyal clock untuk memungkinkan area berbeda dalam FPGA beroperasi dalam kecepatan yang berbeda.

(sumber : https://hardwarebee.com/ultimate-guide-fpga-clock/)

3. Counter

Counter (pencacah) adalah rangkaian logika sekuensial yang digunakan untuk menghitung jumlah pulsa yang diberikan pada bagian masukan. *Counter* digunakan untuk berbagai operasi aritmatika, pembagi frekuensi, penghitung jarak (*odometer*), penghitung kecepatan (*speedometer*), yang pengembangannya digunakan luas dalam aplikasi perhitungan pada instrument ilmiah, kontrol industry, komputer, perlengkapan komunikasi, dan sebagainya.

Counter tersusun atas sederetan flip-flop yang dimanipulasi sedemikian rupa dengan menggunakan peta Karnaught sehingga pulsa yang masuk dapat dihitung sesuai rancangan.

Dalam perancangannya counter dapat tersusun atas semua jenis flip-flop, tergantung karakteristik masing-masing flip-flop.

Dilihat dari arah cacahan, rangkaian counter dibedakan menjadi pencacah naik (Up Counter) dan pencacah turun (Down Counter). Pencacah naik melakukan cacahan dari kecil ke besar, kemudian kembali ke cacahan awal secara otomatis. Pada pencacah menurun, pencacahan dari besar kea rah kecil hingga cacahan terakhir kemudian kembali ke cacahan awal.

(sumber : https://perpustakaan.poltektegal.ac.id/index.php?p=fstream-pdf&fid=5898&bid=4676/)

4. Papan FPGA

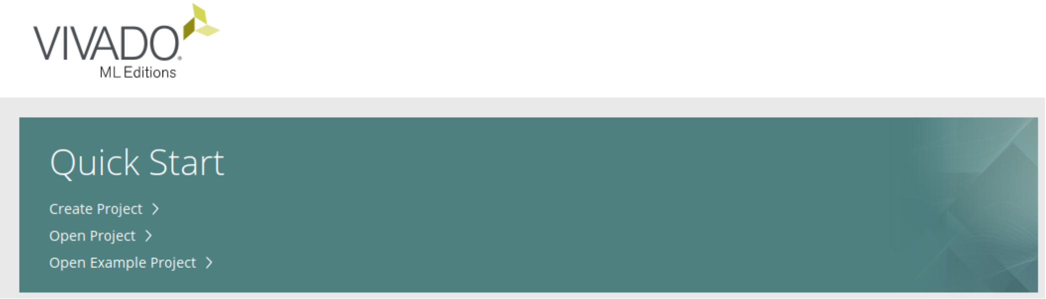
FPGA (Field Programmable Gate Arrays) adalah sirkuit terintegrasi yang perangkat kerasnya dapat dikonfigurasi untuk memenuhi kebutuhan spesifik dari pengguna setelah proses manufaktur. Hal ini membolehkan peningkatan fitur dan perbaikan kerusakan langsung di tempat.

(Sumber: https://www.arm.com/glossary/fpga#:~:text=Field%20Programmable%20Gate%20Arrays%20(FPGAs,requirements%20after%20the%20manufacturing%20process.)

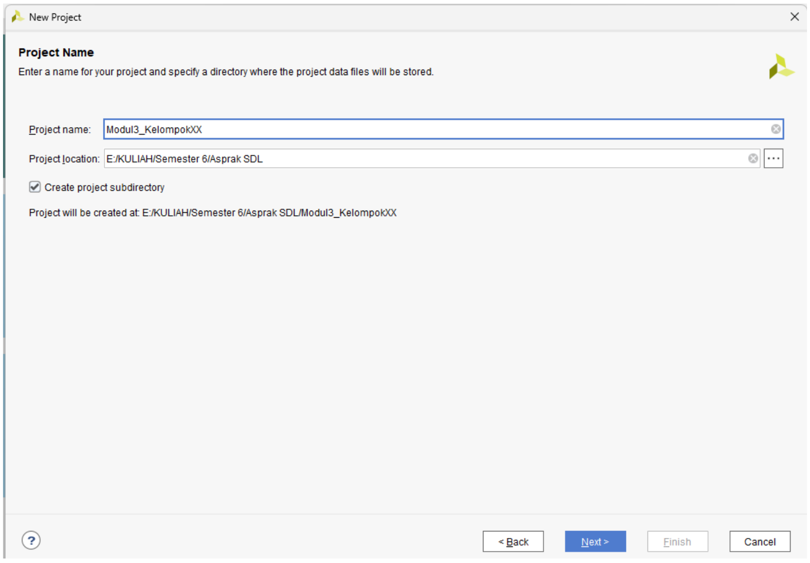
## 3.2 Langkah Kerja

### 3.4.1 Percobaan COUNTER UP

1. Pada halaman awal pilih create project



2. Klik next kemudian buat project dengan nama Modul3_KelompokXX



**Project Name**

Enter a name for your project and specify a directory where the project data files will be stored.

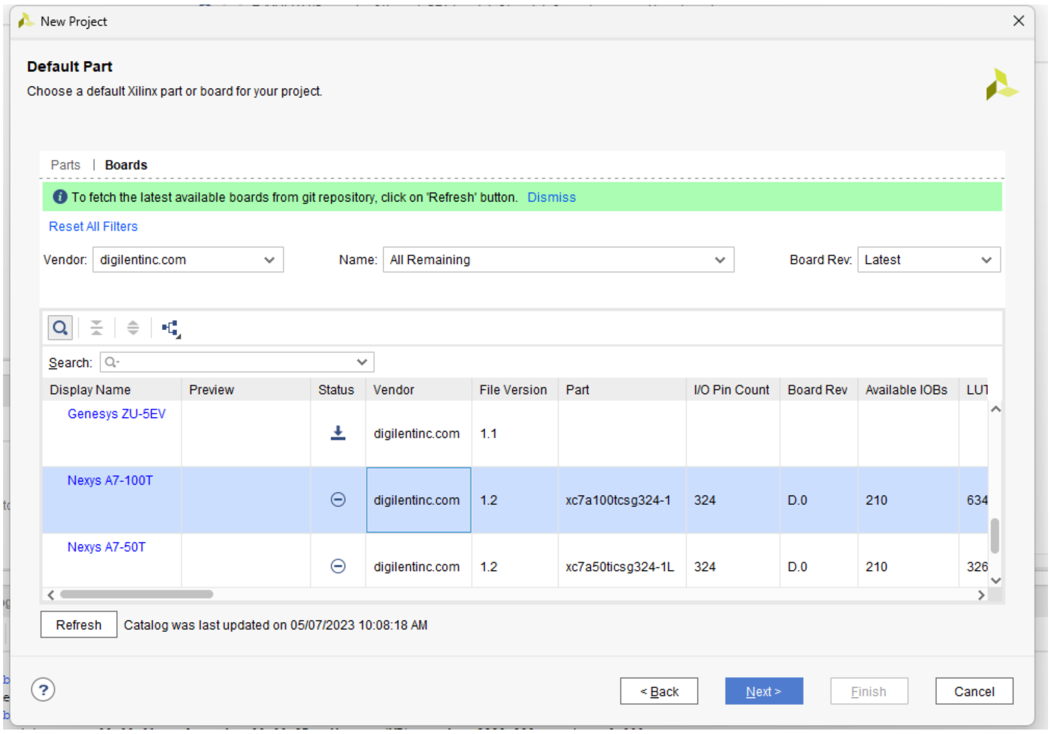
Project name: Modul3_KelompokXX

Project location: E:/KULIAH/Semester 6/Asprak SDL

☑ Create project subdirectory

Project will be created at: E:/KULIAH/Semester 6/Asprak SDL/Modul3_KelompokXX

3. Klik next hingga pada bagian default part, pilih boards Nexys A7-100T, lanjut klik Next dan klik finish.



**Default Part**

Choose a default Xilinx part or board for your project.

To fetch the latest available boards from git repository, click on 'Refresh' button. Dismiss

Vendor: digilentinc.com

Name: All Remaining

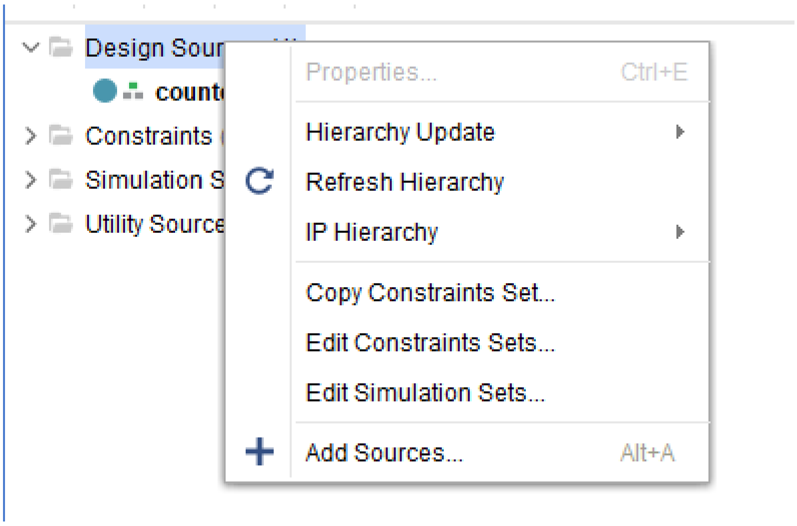
Board Rev: Latest

Search:

| Display Name | Preview | Status | Vendor | File Version | Part | I/O Pin Count | Board Rev | Available IOBs | LUT |
|---|---|---|---|---|---|---|---|---|---|
| Genesys ZU-5EV | | ⤓ | digilentinc.com | 1.1 | | | | | |
| Nexys A7-100T | | ⊖ | digilentinc.com | 1.2 | xc7a100tcsg324-1 | 324 | D.0 | 210 | 634 |
| Nexys A7-50T | | ⊖ | digilentinc.com | 1.2 | xc7a50ticsg324-1L | 324 | D.0 | 210 | 326 |

Refresh   Catalog was last updated on 05/07/2023 10:08:18 AM

4. Buat file design source dengan nama counter



5. Definisikan input dan output pada module dengan source code berikut

```
module counter(
 input clock_100Mhz,
 input reset,
 output reg [3:0] Anode_Activate,
 output reg [6:0] LED_out
    );
```

6. Buat kode Verilog untuk memulai dan me-refresh sinyal clocking, mengaktifkan counter aktivasi LED serta aktivasi Seven segmen output sebagai berikut.

```
reg [26:0] one_second_counter;
wire one_second_enable;
reg [15:0] displayed_number;
reg [3:0] LED_BCD;
reg [19:0] refresh_counter;
wire [1:0] LED_activating_counter;

always @(posedge clock_100Mhz or posedge reset)
   begin
       if(reset==1)
         one_second_counter <= 0;
       else begin
          if(one_second_counter>=99999999)
                 one_second_counter <= 0;
             else
                 one_second_counter <= one_second_counter + 1;
        end
    end
    assign one_second_enable = (one_second_counter==99999999)?1:0;
```

7. Buat fungsi refresh counter untuk pencacahan dan seven segmen sebagai berikut

```
always @(posedge clock_100Mhz or posedge reset)
    begin
        if(reset==1)
            displayed_number <= 0;
        else if(one_second_enable==1)
            displayed_number <= displayed_number + 1;
    end
    always @(posedge clock_100Mhz or posedge reset)
  begin
        if(reset==1)
            refresh_counter <= 0;
        else
            refresh_counter <= refresh_counter + 1;
    end
    assign LED_activating_counter = refresh_counter[19:18];
```

8. Berdasarkan counter aktivasi LED, buat sinyal anoda untuk 4 digit seven segmen sebagai berikut

```
    always @(*)
    begin
        case(LED_activating_counter)
        2'b00: begin
            Anode_Activate = 4'b0111;
            LED_BCD = displayed_number/1000;
              end
        2'b01: begin
            Anode_Activate = 4'b1011;
            LED_BCD = (displayed_number % 1000)/100;
              end
        2'b10: begin
            Anode_Activate = 4'b1101;
            LED_BCD = ((displayed_number % 1000)%100)/10;
                end
        2'b11: begin
            Anode_Activate = 4'b1110;
            LED_BCD = ((displayed_number % 1000)%100)%10;
               end
        endcase
    end
```

9. Selanjutnya, buat fungsi decoder BCD to 7-segment commond anode berdasarkan tabel berikut

| Angka | CA | CB | CC | CD | CE | CF | CG | Chatode [6:0] |
|---|---|---|---|---|---|---|---|---|
| 0 | low | low | low | low | low | low | high | 7’b0000001 |
| 1 | high | low | low | high | high | high | high | 7’b1001111 |
| 2 | low | low | high | low | low | high | low | 7’b0010010 |
| 3 | low | low | low | low | high | high | low | 7’b0000110 |
| 4 | high | low | low | high | high | low | low | 7’b1001100 |
| 5 | low | high | low | low | high | low | low | 7’b0100100 |
| 6 | low | high | low | low | low | low | low | 7’b0100000 |
| 7 | low | low | low | high | high | high | high | 7’b0001111 |
| 8 | low | low | low | low | low | low | low | 7’b0000000 |
| 9 | low | low | low | low | high | low | low | 7’b0000100 |

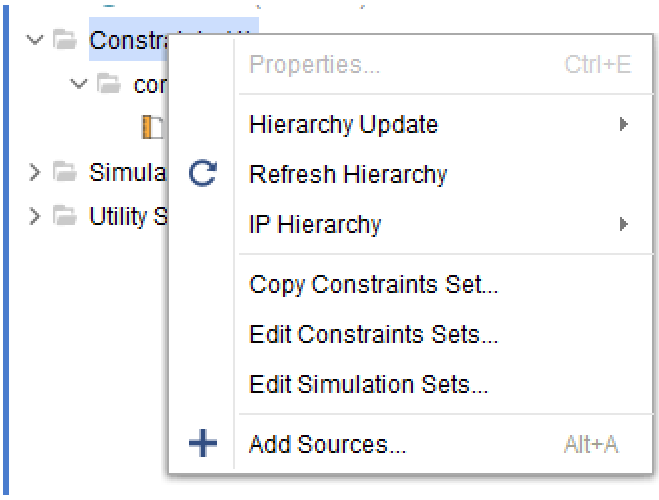
Sehingga diperoleh kode decoder berikut:

```
  always @(*)
    begin
```

```
        case(LED_BCD)
        4'b0000: LED_out = 7'b0000001; // "0"
        4'b0001: LED_out = 7'b1001111; // "1"
        4'b0010: LED_out = 7'b0010010; // "2"
        4'b0011: LED_out = 7'b0000110; // "3"
        4'b0100: LED_out = 7'b1001100; // "4"
        4'b0101: LED_out = 7'b0100100; // "5"
        4'b0110: LED_out = 7'b0100000; // "6"
        4'b0111: LED_out = 7'b0001111; // "7"
        4'b1000: LED_out = 7'b0000000; // "8"
        4'b1001: LED_out = 7'b0000100; // "9"
        default: LED_out = 7'b0000001; // "0"
        endcase
    end
```

10. Akhiri module program dengan kode endmodule di akhir program

11. Buatlah constraint dan memasukkan source code di bawah ini



```
## This file is a general .xdc for the Nexys A7-100T
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to
the top level signal names in the project

## Clock signal
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports {
clock_100Mhz }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
#create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5}
[get_ports {CLK100MHZ}];


##Switches
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports {
reset }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
#set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports
{ SW[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
#set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports
{ SW[2] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
#set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports
{ SW[3] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
#set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports
{ SW[4] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
```

```
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports
{ SW[5] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
#set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports
{ SW[6] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
#set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports
{ SW[7] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
#set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS18 } [get_ports
{ SW[8] }]; #IO_L24N_T3_34 Sch=sw[8]
#set_property -dict { PACKAGE_PIN U8    IOSTANDARD LVCMOS18 } [get_ports
{ SW[9] }]; #IO_25_34 Sch=sw[9]
#set_property -dict { PACKAGE_PIN R16   IOSTANDARD LVCMOS33 } [get_ports
{ SW[10] }]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
#set_property -dict { PACKAGE_PIN T13   IOSTANDARD LVCMOS33 } [get_ports
{ SW[11] }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
#set_property -dict { PACKAGE_PIN H6    IOSTANDARD LVCMOS33 } [get_ports
{ SW[12] }]; #IO_L24P_T3_35 Sch=sw[12]
#set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports
{ SW[13] }]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
#set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports
{ SW[14] }]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
#set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports
{ SW[15] }]; #IO_L21P_T3_DQS_14 Sch=sw[15]

## LEDs
#set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports
{ LED[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
#set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports
{ LED[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
#set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports
{ LED[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
#set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports
{ LED[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
#set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports
{ LED[4] }]; #IO_L7P_T1_D09_14 Sch=led[4]
#set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports
{ LED[5] }]; #IO_L18N_T2_A11_D27_14 Sch=led[5]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports
{ LED[6] }]; #IO_L17P_T2_A14_D30_14 Sch=led[6]
#set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports
{ LED[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7]
#set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports
{ LED[8] }]; #IO_L16N_T2_A15_D31_14 Sch=led[8]
#set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports
{ LED[9] }]; #IO_L14N_T2_SRCC_14 Sch=led[9]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports
{ LED[10] }]; #IO_L22P_T3_A05_D21_14 Sch=led[10]
#set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports
{ LED[11] }]; #IO_L15N_T2_DQS_DOUT_CSO_B_14 Sch=led[11]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports
{ LED[12] }]; #IO_L16P_T2_CSI_B_14 Sch=led[12]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports
{ LED[13] }]; #IO_L22N_T3_A04_D20_14 Sch=led[13]
#set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports
{ LED[14] }]; #IO_L20N_T3_A07_D23_14 Sch=led[14]
#set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports
{ LED[15] }]; #IO_L21N_T3_DQS_A06_D22_14 Sch=led[15]

## RGB LEDs
#set_property -dict { PACKAGE_PIN R12   IOSTANDARD LVCMOS33 } [get_ports
{ LED16_B }]; #IO_L5P_T0_D06_14 Sch=led16_b
```

```
#set_property -dict { PACKAGE_PIN M16   IOSTANDARD LVCMOS33 } [get_ports
{ LED16_G }]; #IO_L10P_T1_D14_14 Sch=led16_g
#set_property -dict { PACKAGE_PIN N15   IOSTANDARD LVCMOS33 } [get_ports
{ LED16_R }]; #IO_L11P_T1_SRCC_14 Sch=led16_r
#set_property -dict { PACKAGE_PIN G14   IOSTANDARD LVCMOS33 } [get_ports
{ LED17_B }]; #IO_L15N_T2_DQS_ADV_B_15 Sch=led17_b
#set_property -dict { PACKAGE_PIN R11   IOSTANDARD LVCMOS33 } [get_ports
{ LED17_G }]; #IO_0_14 Sch=led17_g
#set_property -dict { PACKAGE_PIN N16   IOSTANDARD LVCMOS33 } [get_ports
{ LED17_R }]; #IO_L11N_T1_SRCC_14 Sch=led17_r

##7 segment display
set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports {
LED_out[6] }]; #IO_L24N_T3_A00_D16_14 Sch=ca
set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports {
LED_out[5] }]; #IO_25_14 Sch=cb
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports {
LED_out[4] }]; #IO_25_15 Sch=cc
set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports {
LED_out[3] }]; #IO_L17P_T2_A26_15 Sch=cd
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports {
LED_out[2] }]; #IO_L13P_T2_MRCC_14 Sch=ce
set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports {
LED_out[1] }]; #IO_L19P_T3_A10_D26_14 Sch=cf
set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports {
LED_out[0] }]; #IO_L4P_T0_D04_14 Sch=cg
#set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports
{ DP }]; #IO_L19N_T3_A21_VREF_15 Sch=dp
set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports {
Anode_Activate[0] }]; #IO_L23P_T3_FOE_B_15 Sch=an[0]
set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports {
Anode_Activate[1] }]; #IO_L23N_T3_FWE_B_15 Sch=an[1]
set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports {
Anode_Activate[2] }]; #IO_L24P_T3_A01_D17_14 Sch=an[2]
set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports {
Anode_Activate[3] }]; #IO_L19P_T3_A22_15 Sch=an[3]
#set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 } [get_ports
{ AN[4] }]; #IO_L8N_T1_D12_14 Sch=an[4]
#set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports
{ AN[5] }]; #IO_L14P_T2_SRCC_14 Sch=an[5]
#set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVCMOS33 } [get_ports
{ AN[6] }]; #IO_L23P_T3_35 Sch=an[6]
#set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 } [get_ports
{ AN[7] }]; #IO_L23N_T3_A02_D18_14 Sch=an[7]

##CPU Reset Button
#set_property -dict { PACKAGE_PIN C12   IOSTANDARD LVCMOS33 } [get_ports
{ CPU_RESETN }]; #IO_L3P_T0_DQS_AD1P_15 Sch=cpu_resetn

##Buttons
#set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports
{ BTNC }]; #IO_L9P_T1_DQS_14 Sch=btnc
#set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 } [get_ports
{ BTNU }]; #IO_L4N_T0_D05_14 Sch=btnu
#set_property -dict { PACKAGE_PIN P17   IOSTANDARD LVCMOS33 } [get_ports
{ BTNL }]; #IO_L12P_T1_MRCC_14 Sch=btnl
#set_property -dict { PACKAGE_PIN M17   IOSTANDARD LVCMOS33 } [get_ports
{ BTNR }]; #IO_L10N_T1_D15_14 Sch=btnr
#set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVCMOS33 } [get_ports
{ BTND }]; #IO_L9N_T1_DQS_D13_14 Sch=btnd
```

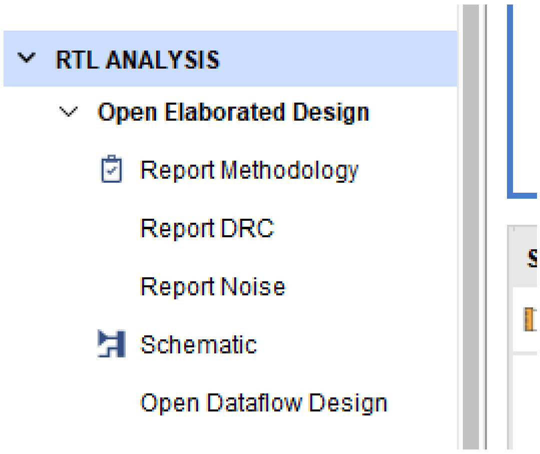
```
##Pmod Headers
##Pmod Header JA
#set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33 } [get_ports
{ JA[1] }]; #IO_L20N_T3_A19_15 Sch=ja[1]
#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports
{ JA[2] }]; #IO_L21N_T3_DQS_A18_15 Sch=ja[2]
#set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 } [get_ports
{ JA[3] }]; #IO_L21P_T3_DQS_15 Sch=ja[3]
#set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports
{ JA[4] }]; #IO_L18N_T2_A23_15 Sch=ja[4]
#set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports
{ JA[7] }]; #IO_L16N_T2_A27_15 Sch=ja[7]
#set_property -dict { PACKAGE_PIN E17   IOSTANDARD LVCMOS33 } [get_ports
{ JA[8] }]; #IO_L16P_T2_A28_15 Sch=ja[8]
#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports
{ JA[9] }]; #IO_L22N_T3_A16_15 Sch=ja[9]
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports
{ JA[10] }]; #IO_L22P_T3_A17_15 Sch=ja[10]

##Pmod Header JB
#set_property -dict { PACKAGE_PIN D14   IOSTANDARD LVCMOS33 } [get_ports
{ JB[1] }]; #IO_L1P_T0_AD0P_15 Sch=jb[1]
#set_property -dict { PACKAGE_PIN F16   IOSTANDARD LVCMOS33 } [get_ports
{ JB[2] }]; #IO_L14N_T2_SRCC_15 Sch=jb[2]
#set_property -dict { PACKAGE_PIN G16   IOSTANDARD LVCMOS33 } [get_ports
{ JB[3] }]; #IO_L13N_T2_MRCC_15 Sch=jb[3]
#set_property -dict { PACKAGE_PIN H14   IOSTANDARD LVCMOS33 } [get_ports
{ JB[4] }]; #IO_L15P_T2_DQS_15 Sch=jb[4]
#set_property -dict { PACKAGE_PIN E16   IOSTANDARD LVCMOS33 } [get_ports
{ JB[7] }]; #IO_L11N_T1_SRCC_15 Sch=jb[7]
#set_property -dict { PACKAGE_PIN F13   IOSTANDARD LVCMOS33 } [get_ports
{ JB[8] }]; #IO_L5P_T0_AD9P_15 Sch=jb[8]
#set_property -dict { PACKAGE_PIN G13   IOSTANDARD LVCMOS33 } [get_ports
{ JB[9] }]; #IO_0_15 Sch=jb[9]
#set_property -dict { PACKAGE_PIN H16   IOSTANDARD LVCMOS33 } [get_ports
{ JB[10] }]; #IO_L13P_T2_MRCC_15 Sch=jb[10]

##Pmod Header JC
#set_property -dict { PACKAGE_PIN K1    IOSTANDARD LVCMOS33 } [get_ports
{ JC[1] }]; #IO_L23N_T3_35 Sch=jc[1]
#set_property -dict { PACKAGE_PIN F6    IOSTANDARD LVCMOS33 } [get_ports
{ JC[2] }]; #IO_L19N_T3_VREF_35 Sch=jc[2]
#set_property -dict { PACKAGE_PIN J2    IOSTANDARD LVCMOS33 } [get_ports
{ JC[3] }]; #IO_L22N_T3_35 Sch=jc[3]
#set_property -dict { PACKAGE_PIN G6    IOSTANDARD LVCMOS33 } [get_ports
{ JC[4] }]; #IO_L19P_T3_35 Sch=jc[4]
#set_property -dict { PACKAGE_PIN E7    IOSTANDARD LVCMOS33 } [get_ports
{ JC[7] }]; #IO_L6P_T0_35 Sch=jc[7]
#set_property -dict { PACKAGE_PIN J3    IOSTANDARD LVCMOS33 } [get_ports
{ JC[8] }]; #IO_L22P_T3_35 Sch=jc[8]
#set_property -dict { PACKAGE_PIN J4    IOSTANDARD LVCMOS33 } [get_ports
{ JC[9] }]; #IO_L21P_T3_DQS_35 Sch=jc[9]
#set_property -dict { PACKAGE_PIN E6    IOSTANDARD LVCMOS33 } [get_ports
{ JC[10] }]; #IO_L5P_T0_AD13P_35 Sch=jc[10]

##Pmod Header JD
#set_property -dict { PACKAGE_PIN H4    IOSTANDARD LVCMOS33 } [get_ports
{ JD[1] }]; #IO_L21N_T3_DQS_35 Sch=jd[1]
#set_property -dict { PACKAGE_PIN H1    IOSTANDARD LVCMOS33 } [get_ports
{ JD[2] }]; #IO_L17P_T2_35 Sch=jd[2]
```

```
#set_property -dict { PACKAGE_PIN G1    IOSTANDARD LVCMOS33 } [get_ports
{ JD[3] }]; #IO_L17N_T2_35 Sch=jd[3]
#set_property -dict { PACKAGE_PIN G3    IOSTANDARD LVCMOS33 } [get_ports
{ JD[4] }]; #IO_L20N_T3_35 Sch=jd[4]
#set_property -dict { PACKAGE_PIN H2    IOSTANDARD LVCMOS33 } [get_ports
{ JD[7] }]; #IO_L15P_T2_DQS_35 Sch=jd[7]
#set_property -dict { PACKAGE_PIN G4    IOSTANDARD LVCMOS33 } [get_ports
{ JD[8] }]; #IO_L20P_T3_35 Sch=jd[8]
#set_property -dict { PACKAGE_PIN G2    IOSTANDARD LVCMOS33 } [get_ports
{ JD[9] }]; #IO_L15N_T2_DQS_35 Sch=jd[9]
#set_property -dict { PACKAGE_PIN F3    IOSTANDARD LVCMOS33 } [get_ports
{ JD[10] }]; #IO_L13N_T2_MRCC_35 Sch=jd[10]

##Pmod Header JXADC
#set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33 } [get_ports
{ XA_N[1] }]; #IO_L9N_T1_DQS_AD3N_15 Sch=xa_n[1]
#set_property -dict { PACKAGE_PIN A13   IOSTANDARD LVCMOS33 } [get_ports
{ XA_P[1] }]; #IO_L9P_T1_DQS_AD3P_15 Sch=xa_p[1]
#set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMOS33 } [get_ports
{ XA_N[2] }]; #IO_L8N_T1_AD10N_15 Sch=xa_n[2]
#set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVCMOS33 } [get_ports
{ XA_P[2] }]; #IO_L8P_T1_AD10P_15 Sch=xa_p[2]
#set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33 } [get_ports
{ XA_N[3] }]; #IO_L7N_T1_AD2N_15 Sch=xa_n[3]
#set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMOS33 } [get_ports
{ XA_P[3] }]; #IO_L7P_T1_AD2P_15 Sch=xa_p[3]
#set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports
{ XA_N[4] }]; #IO_L10N_T1_AD11N_15 Sch=xa_n[4]
#set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports
{ XA_P[4] }]; #IO_L10P_T1_AD11P_15 Sch=xa_p[4]

##VGA Connector
#set_property -dict { PACKAGE_PIN A3    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_R[0] }]; #IO_L8N_T1_AD14N_35 Sch=vga_r[0]
#set_property -dict { PACKAGE_PIN B4    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_R[1] }]; #IO_L7N_T1_AD6N_35 Sch=vga_r[1]
#set_property -dict { PACKAGE_PIN C5    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_R[2] }]; #IO_L1N_T0_AD4N_35 Sch=vga_r[2]
#set_property -dict { PACKAGE_PIN A4    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_R[3] }]; #IO_L8P_T1_AD14P_35 Sch=vga_r[3]
#set_property -dict { PACKAGE_PIN C6    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_G[0] }]; #IO_L1P_T0_AD4P_35 Sch=vga_g[0]
#set_property -dict { PACKAGE_PIN A5    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_G[1] }]; #IO_L3N_T0_DQS_AD5N_35 Sch=vga_g[1]
#set_property -dict { PACKAGE_PIN B6    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_G[2] }]; #IO_L2N_T0_AD12N_35 Sch=vga_g[2]
#set_property -dict { PACKAGE_PIN A6    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_G[3] }]; #IO_L3P_T0_DQS_AD5P_35 Sch=vga_g[3]
#set_property -dict { PACKAGE_PIN B7    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_B[0] }]; #IO_L2P_T0_AD12P_35 Sch=vga_b[0]
#set_property -dict { PACKAGE_PIN C7    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_B[1] }]; #IO_L4N_T0_35 Sch=vga_b[1]
#set_property -dict { PACKAGE_PIN D7    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_B[2] }]; #IO_L6N_T0_VREF_35 Sch=vga_b[2]
#set_property -dict { PACKAGE_PIN D8    IOSTANDARD LVCMOS33 } [get_ports
{ VGA_B[3] }]; #IO_L4P_T0_35 Sch=vga_b[3]
#set_property -dict { PACKAGE_PIN B11   IOSTANDARD LVCMOS33 } [get_ports
{ VGA_HS }]; #IO_L4P_T0_15 Sch=vga_hs
#set_property -dict { PACKAGE_PIN B12   IOSTANDARD LVCMOS33 } [get_ports
{ VGA_VS }]; #IO_L3N_T0_DQS_AD1N_15 Sch=vga_vs
```

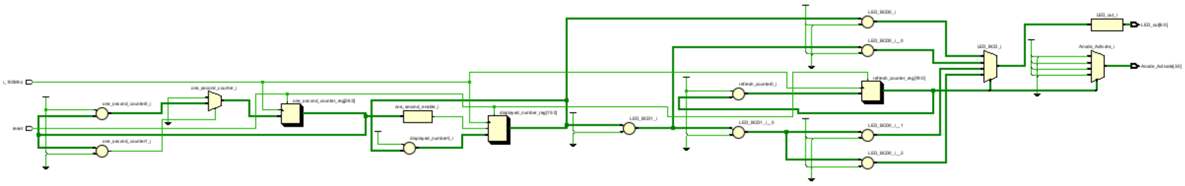
```
##Micro SD Connector
#set_property -dict { PACKAGE_PIN E2    IOSTANDARD LVCMOS33 } [get_ports
{ SD_RESET }]; #IO_L14P_T2_SRCC_35 Sch=sd_reset
#set_property -dict { PACKAGE_PIN A1    IOSTANDARD LVCMOS33 } [get_ports
{ SD_CD }]; #IO_L9N_T1_DQS_AD7N_35 Sch=sd_cd
#set_property -dict { PACKAGE_PIN B1    IOSTANDARD LVCMOS33 } [get_ports
{ SD_SCK }]; #IO_L9P_T1_DQS_AD7P_35 Sch=sd_sck
#set_property -dict { PACKAGE_PIN C1    IOSTANDARD LVCMOS33 } [get_ports
{ SD_CMD }]; #IO_L16N_T2_35 Sch=sd_cmd
#set_property -dict { PACKAGE_PIN C2    IOSTANDARD LVCMOS33 } [get_ports
{ SD_DAT[0] }]; #IO_L16P_T2_35 Sch=sd_dat[0]
#set_property -dict { PACKAGE_PIN E1    IOSTANDARD LVCMOS33 } [get_ports
{ SD_DAT[1] }]; #IO_L18N_T2_35 Sch=sd_dat[1]
#set_property -dict { PACKAGE_PIN F1    IOSTANDARD LVCMOS33 } [get_ports
{ SD_DAT[2] }]; #IO_L18P_T2_35 Sch=sd_dat[2]
#set_property -dict { PACKAGE_PIN D2    IOSTANDARD LVCMOS33 } [get_ports
{ SD_DAT[3] }]; #IO_L14N_T2_SRCC_35 Sch=sd_dat[3]

##Accelerometer
#set_property -dict { PACKAGE_PIN E15   IOSTANDARD LVCMOS33 } [get_ports
{ ACL_MISO }]; #IO_L11P_T1_SRCC_15 Sch=acl_miso
#set_property -dict { PACKAGE_PIN F14   IOSTANDARD LVCMOS33 } [get_ports
{ ACL_MOSI }]; #IO_L5N_T0_AD9N_15 Sch=acl_mosi
#set_property -dict { PACKAGE_PIN F15   IOSTANDARD LVCMOS33 } [get_ports
{ ACL_SCLK }]; #IO_L14P_T2_SRCC_15 Sch=acl_sclk
#set_property -dict { PACKAGE_PIN D15   IOSTANDARD LVCMOS33 } [get_ports
{ ACL_CSN }]; #IO_L12P_T1_MRCC_15 Sch=acl_csn
#set_property -dict { PACKAGE_PIN B13   IOSTANDARD LVCMOS33 } [get_ports
{ ACL_INT[1] }]; #IO_L2P_T0_AD8P_15 Sch=acl_int[1]
#set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVCMOS33 } [get_ports
{ ACL_INT[2] }]; #IO_L20P_T3_A20_15 Sch=acl_int[2]

##Temperature Sensor
#set_property -dict { PACKAGE_PIN C14   IOSTANDARD LVCMOS33 } [get_ports
{ TMP_SCL }]; #IO_L1N_T0_AD0N_15 Sch=tmp_scl
#set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVCMOS33 } [get_ports
{ TMP_SDA }]; #IO_L12N_T1_MRCC_15 Sch=tmp_sda
#set_property -dict { PACKAGE_PIN D13   IOSTANDARD LVCMOS33 } [get_ports
{ TMP_INT }]; #IO_L6N_T0_VREF_15 Sch=tmp_int
#set_property -dict { PACKAGE_PIN B14   IOSTANDARD LVCMOS33 } [get_ports
{ TMP_CT }]; #IO_L2N_T0_AD8N_15 Sch=tmp_ct

##Omnidirectional Microphone
#set_property -dict { PACKAGE_PIN J5    IOSTANDARD LVCMOS33 } [get_ports
{ M_CLK }]; #IO_25_35 Sch=m_clk
#set_property -dict { PACKAGE_PIN H5    IOSTANDARD LVCMOS33 } [get_ports
{ M_DATA }]; #IO_L24N_T3_35 Sch=m_data
#set_property -dict { PACKAGE_PIN F5    IOSTANDARD LVCMOS33 } [get_ports
{ M_LRSEL }]; #IO_0_35 Sch=m_lrsel

##PWM Audio Amplifier
#set_property -dict { PACKAGE_PIN A11   IOSTANDARD LVCMOS33 } [get_ports
{ AUD_PWM }]; #IO_L4N_T0_15 Sch=aud_pwm
#set_property -dict { PACKAGE_PIN D12   IOSTANDARD LVCMOS33 } [get_ports
{ AUD_SD }]; #IO_L6P_T0_15 Sch=aud_sd

##USB-RS232 Interface
#set_property -dict { PACKAGE_PIN C4    IOSTANDARD LVCMOS33 } [get_ports
{ UART_TXD_IN }]; #IO_L7P_T1_AD6P_35 Sch=uart_txd_in
#set_property -dict { PACKAGE_PIN D4    IOSTANDARD LVCMOS33 } [get_ports
{ UART_RXD_OUT }]; #IO_L11N_T1_SRCC_35 Sch=uart_rxd_out
```

```
#set_property -dict { PACKAGE_PIN D3    IOSTANDARD LVCMOS33 } [get_ports
{ UART_CTS }]; #IO_L12N_T1_MRCC_35 Sch=uart_cts
#set_property -dict { PACKAGE_PIN E5    IOSTANDARD LVCMOS33 } [get_ports
{ UART_RTS }]; #IO_L5N_T0_AD13N_35 Sch=uart_rts

##USB HID (PS/2)
#set_property -dict { PACKAGE_PIN F4    IOSTANDARD LVCMOS33 } [get_ports
{ PS2_CLK }]; #IO_L13P_T2_MRCC_35 Sch=ps2_clk
#set_property -dict { PACKAGE_PIN B2    IOSTANDARD LVCMOS33 } [get_ports
{ PS2_DATA }]; #IO_L10N_T1_AD15N_35 Sch=ps2_data

##SMSC Ethernet PHY
#set_property -dict { PACKAGE_PIN C9    IOSTANDARD LVCMOS33 } [get_ports
{ ETH_MDC }]; #IO_L11P_T1_SRCC_16 Sch=eth_mdc
#set_property -dict { PACKAGE_PIN A9    IOSTANDARD LVCMOS33 } [get_ports
{ ETH_MDIO }]; #IO_L14N_T2_SRCC_16 Sch=eth_mdio
#set_property -dict { PACKAGE_PIN B3    IOSTANDARD LVCMOS33 } [get_ports
{ ETH_RSTN }]; #IO_L10P_T1_AD15P_35 Sch=eth_rstn
#set_property -dict { PACKAGE_PIN D9    IOSTANDARD LVCMOS33 } [get_ports
{ ETH_CRSDV }]; #IO_L6N_T0_VREF_16 Sch=eth_crsdv
#set_property -dict { PACKAGE_PIN C10   IOSTANDARD LVCMOS33 } [get_ports
{ ETH_RXERR }]; #IO_L13N_T2_MRCC_16 Sch=eth_rxerr
#set_property -dict { PACKAGE_PIN C11   IOSTANDARD LVCMOS33 } [get_ports
{ ETH_RXD[0] }]; #IO_L13P_T2_MRCC_16 Sch=eth_rxd[0]
#set_property -dict { PACKAGE_PIN D10   IOSTANDARD LVCMOS33 } [get_ports
{ ETH_RXD[1] }]; #IO_L19N_T3_VREF_16 Sch=eth_rxd[1]
#set_property -dict { PACKAGE_PIN B9    IOSTANDARD LVCMOS33 } [get_ports
{ ETH_TXEN }]; #IO_L11N_T1_SRCC_16 Sch=eth_txen
#set_property -dict { PACKAGE_PIN A10   IOSTANDARD LVCMOS33 } [get_ports
{ ETH_TXD[0] }]; #IO_L14P_T2_SRCC_16 Sch=eth_txd[0]
#set_property -dict { PACKAGE_PIN A8    IOSTANDARD LVCMOS33 } [get_ports
{ ETH_TXD[1] }]; #IO_L12N_T1_MRCC_16 Sch=eth_txd[1]
#set_property -dict { PACKAGE_PIN D5    IOSTANDARD LVCMOS33 } [get_ports
{ ETH_REFCLK }]; #IO_L11P_T1_SRCC_35 Sch=eth_refclk
#set_property -dict { PACKAGE_PIN B8    IOSTANDARD LVCMOS33 } [get_ports
{ ETH_INTN }]; #IO_L12P_T1_MRCC_16 Sch=eth_intn

##Quad SPI Flash
#set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports
{ QSPI_DQ[0] }]; #IO_L1P_T0_D00_MOSI_14 Sch=qspi_dq[0]
#set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 } [get_ports
{ QSPI_DQ[1] }]; #IO_L1N_T0_D01_DIN_14 Sch=qspi_dq[1]
#set_property -dict { PACKAGE_PIN L14   IOSTANDARD LVCMOS33 } [get_ports
{ QSPI_DQ[2] }]; #IO_L2P_T0_D02_14 Sch=qspi_dq[2]
#set_property -dict { PACKAGE_PIN M14   IOSTANDARD LVCMOS33 } [get_ports
{ QSPI_DQ[3] }]; #IO_L2N_T0_D03_14 Sch=qspi_dq[3]
#set_property -dict { PACKAGE_PIN L13   IOSTANDARD LVCMOS33 } [get_ports
{ QSPI_CSN }]; #IO_L6P_T0_FCS_B_14 Sch=qspi_csn
```

12. Untuk melihat rangkaian, klik open elaborated design kemudian pilih schematic

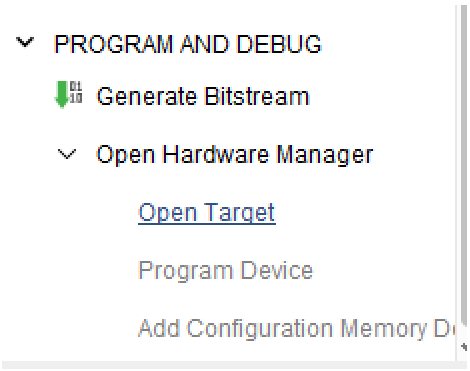


13.

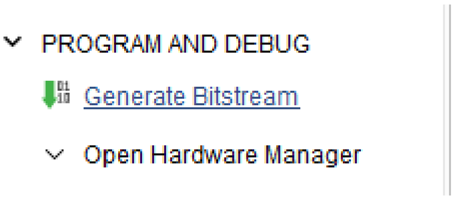


14. Sambungkan papan dengan laptop dan nyalakan switch power

15. sambungkan papan ke vivado



16. Generate Bitstream



17. Klik Program device

18. Aktifkan switch yang terhubung pada constrain dan lihat keluarannya