

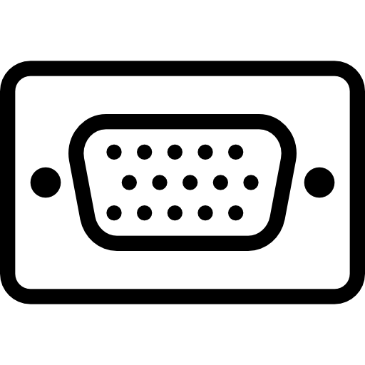
**Facultatea de automatica si calculatoare**

**Departamentul de calculatoare**

*Structura sistemelor de calcul*

**Utilizarea portului VGA al placii Nexys 4 DDR pentru afisarea**

**unor imagini**

****

Student : Gligor Mihai

Grupa : 302310

Profesor coordinator : Fati Daniela

Data : 12.01.2022

Cuprins

[1.Rezumat 3](#_Toc92817579)

[**2.Introducere** 3](#_Toc92817580)

[**3.Fundamentare teoretica** 4](#_Toc92817581)

[**4.Proiectare si implementare** 8](#_Toc92817582)

[**5**.**Rezultate experimentale** 10](#_Toc92817583)

[**6.Concluzii** 10](#_Toc92817584)

# **1.Rezumat**

Afisarea informatiei pe ecrane este o functionalitate necesara in orice domeniu de munca , de la entertainment pana la domeniile industriale ,astfel incat deprinderea abilitatilor necesare pentru a efectua aceasta actiune e o aditie binevenita in pachetul de cunostinte a oricarui inginer.

Acest proiect isi propune afisarea unor imagini de baza (ex cateva linii) pentru a demonstra conceptul si metodele de implementare ale acestuia . Una dintre acestre metode de implementare este prezentata aici prin intermediul unei placute programabile Nexys4 ddr , folosind limbajul VHDL , mediul de dezvoltare Vivado Design Suite 2016.4 si un monitor VGA.

Astfel sunt demonstrate principiile de baza ale afisarii imaginilor prin intermediul portului VGA.

## **2.Introducere**

Conectorul VGA

Dezvoltat in anul 1987 de catre compania IBM conectorul VGA (video graphic array) e un conector standard de tip D-sub (din cauza shield ului in forma de D) cu 15, pini pentru output ul video al calculatoarelor .

Desi portul VGA este folosit din ce in ce mai putin in detrimentul portului HDMI ,acesta este inca prezent in numeroase contexte tehnologice , principiul lui de functionare fiind inca standardul multor segmente industriale

Astfel , in cele ce urmeaza va fii prezentata o solutie simpla de afisare astfel incat sa fie clare principiile de functionare ale portului VGA.

In primul rand va exista o sectiune de fundament teoretic care vca explica principiile pe care functioneaza aceasta afisare , dupa care partea de implementare si proiectare care va intra in detalii cu privire la metoda folosita ,ca mai apoi sa apara sectiunile de rezultate experimentale si concluzii unde vor fi expuse efectele aplicarii metodei respective  
La final se afla bibliografia cu sursele de documentatie si Anexa care contine codul folosit si continutul fisierului de constrangeri.

### **3.Fundamentare teoretica**

Cablul VGA functioneaza ca o legatura intre computer si monitor transmitand semnale video analog ale culorilor standard RGB (red green blue ) si ale datelor horizontal vertical si VESA prin intermediul celor 15 pini asezati pe trei randuri .

O imagine care conține masă

Descriere generată automat

Placa Nexys4 ddr

Placa DDR Nexys 4 este o platformă completă, gata de utilizare pentru dezvoltarea circuitelor digitale, bazată pe cea mai recentă matrice de porți programabile (FPGA) Artix-7™ de la Xilinx®.Cu FPGA de mare capacitate (Xilinx număr de catalog XC7A100T-1CSG324C), memorii externe generoase, și o colectie de USB, Ethernet, și alte porturi, DDR Nexys4 poate găzdui modele variind de la circuite combinationale introductive la procesoare încorporate puternice.Mai multe periferice încorporate, inclusiv un accelerometru, un senzor de temperatură, un microfon digital MEMS, un amplificator de difuzor,Și mai multe dispozitive I/o permit utilizarea DDR-ului Nexys4 pentru o gamă largă de proiecte, fără a fi nevoie de alte componente

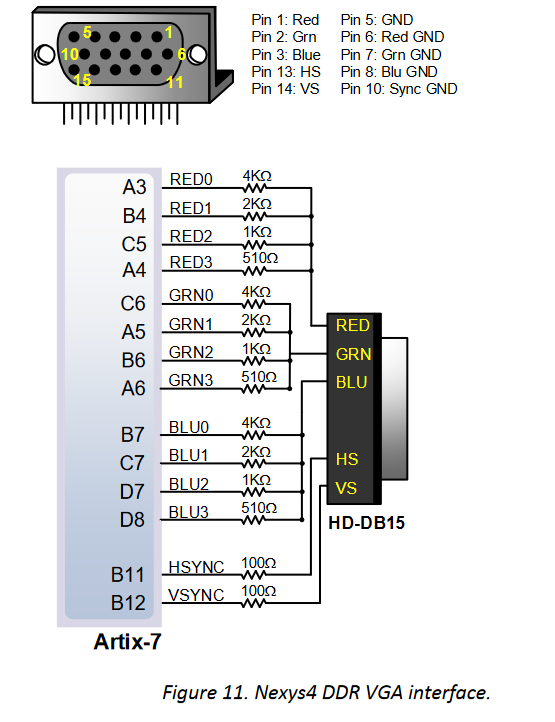
O imagine care conține text, electronice, circuit

Descriere generată automat

Portul VGA si conexiunea cu placa nexys 4 ddr

Placa nexys 4 ddr foloseste 14 semnale FPGA pentru a creea un port VGA cu 4 biti per culoare si doua semnale standard de sincronizare ( vertical si orizontal ) . Semnalele pentru culori folosesc circuite de divizare a rezistentelor care functiuoneaza in conjunctie cu rezistentele de 75 de ohmi ale display ului VGA pentru a creea 16 nivele de intensitate pentru fiecare culoare . Intensitatea creste in intremente egale intre 0 V(complet oprit) si 0,7 V (complet pornit) ,astfel fiind capabil sa afiseze 4096 de culori diferite pentru fiecare sir de 12 biti posibil.

Pentru a avea un display functional va trebui creeat un controller video care sa atribuie semnalelor de sync si culoare un timing corespunzator .



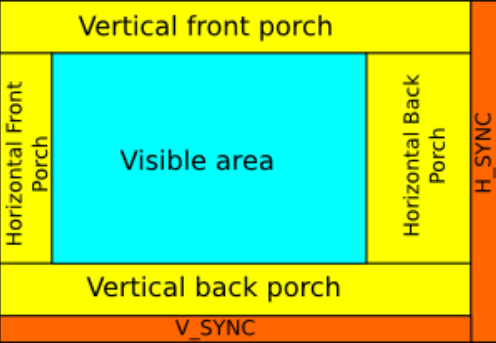
Fiecare linie incepe cu LOW pe semnalul horizontal sync si ficeare frame cu LOW pe semnalul vertical sync . Inainte si dupa fiecare puls de sincronizare semnalele rgb trebuia sa ia valoarea LOW pentru un anumit numar de cicluri de ceas , aceste perioade fiind denumite front porch respective back porch

Pentru o rezolutie de 640 x 480 linia orizontala consista in 640 de pixeli vizibili , 16 pixeli pentru front porch , 16 pentru back porch si 128 de pixeli pentru pulsul de sincronizare , in total insumand 800 de pixeli .Fiecare pixel e parcurs in 0.04 microsecunde ,deci vom executa o scanare orizontala complete a unei linii in 32 de microsecunde (800\*0.04) . Presupunem ca monitorul are 60 Hz deci are un output de 60 de frame uri pe secunda , de aici calculam 1/60 , deci un frame odata la 16,67 de milisecunde .O scanare orizontala tine 32 de milisecunde deci avem 16,67 milisecunde / 32 microsecunde , adica 521 de scanari orizontale per frame

Linia verticala contine 480 de pixeli vizibili , 29 de pixeli pentru back porch , 10 pixeli pentru front porch si 2 pentru pulsul de sincronizare , insumand 521 de scanari orizontale a cate 32 de microsecunde fiecare , deci un total de 16,67 milisecunde (un frame ).

Pentru o rezolutie de 1280 x 1024 linia orizontala consista in 1280 de pixeli vizibili , 48 pixeli pentru front porch , 248 pentru back porch si 112 de pixeli pentru pulsul de sincronizare , in total insumand 1688 de pixeli .Fiecare pixel e parcurs in 0.04 microsecunde ,deci vom executa o scanare orizontala complete a unei linii in 67,52 de microsecunde (1688\*0.04) . Presupunem ca monitorul are 60 Hz deci are un output de 60 de frame uri pe secunda , de aici calculam 1/60 , deci un frame odata la 16,67 de milisecunde .O scanare orizontala tine 67,52 de milisecunde deci avem 16,67 milisecunde / 67,52 microsecunde , adica 246 de scanari orizontale per frame

Linia verticala contine 1024 de pixeli vizibili , 38 de pixeli pentru back porch , 1 pixel pentru front porch si 3 pentru pulsul de sincronizare , insumand 246 de scanari orizontale a cate 67,56 de microsecunde fiecare , deci un total de 16,60 milisecunde (un frame ).



#### **4.Proiectare si implementare**

Pentru implementarea acestui proiect am folosit mediul de dezvoltare Vivado Design Suite al companiei Xilinx , versiunea 2016.4 .

In entitatea principala numita nexys4 avem doar port map ul pentru componenta principala numita VGA pentru a facilita conexiunea la placa Nexys4 ddr .

Entitatea VGA are ca intrare in port map doar clock ul , si ca iesiri semnalele VGA\_HS care reprezinta horizontal sync-ul , VGA\_VS care reprezinta vertical sync-ul si semnalele VGA\_R, VGA\_G, VGA\_B care sunt reprezentate de vectori de 4 biti , prin intermediul carora se vor transmite date cu privire la culoare. Aceasta entitate are in arhitectura sa doar componenta SYNC care sincronizeaza placa Nexys4 ddr si portul VGA .

Entitatea SYNC contine in port map semnalele de clock (singurul semnal de tip in), hsync, vsync (horizontal respective vertical sync) si vectorii pentru pinii de culoare r, g, b.

In arhitectura acestei entitati am declarant urmatoarele semnale :

***signal HPOS: integer range 0 to 1688:=0;***

***signal VPOS: integer range 0 to 1066:=0;***

care au valoriile respective deoarece hpos reprezinta linia orizontala formata din 1280 de pixeli vizibili + 48 de pixeli pentru front porch + 248 de pixeli pentru back porch + 112 pixeli pentru sync , insumand 1688 pixeli

vpos reprezinta linia verticala formata din 1024 de pixeli vizibili + 1 de pixel pentru front porch + 38 de pixeli pentru back porch + 3 pixeli pentru sync , insumand 1066 pixeli .Dupa begin avem un singur process format din mai multe if uri

***if hpos=1048 or vpos=554 or hpos=524 or vpos=277 then***

***r<=(others=>'1');***

***g<=(others=>'1');***

***b<=(others=>'1');***

***else***

***r<=(others=>'0');***

***g<=(others=>'0');***

***b<=(others=>'0');***

***end if;***

Primul selecteaza pixelii care vor fii colorati, in acest caz cu alb (r,g,b primesc valori de 1 logic)

Dupa care avem if ul in care se parcurge ecranul pe care vrem sa afisam, linie cu linie

***if hpos<1688 then***

***HPOS<=HPOS+1;***

***else***

***HPOS<=0;***

***if vpos<1066 then***

***vpos<=vpos+1; ---scanarea ecranului linie cu linie***

***else***

***vpos<=0;***

***end if;***

***end if;***

Acesta parcurge pixel cu pixel linia orizontala a ecranului si dupa ce ajunge la capat , reseteaza la o pozitia orizontala si incrementeaza positia verticala (trece la urmatoarea linie) cand ajunge la final ,resetand-o la 0 .

Pentru partea de sincronizare , documentatia legata de VGA ne indica faptul ca intre front porch si back porch avem nevoie de valori de 0 logic la semnalele de sincronizare orizontala si verticala, problema rezolvata prin intermediul urmatoarelor doua if uri

***if hpos>48 and hpos<160 then ---horizontal sync are low intre front porch si back porch***

***hsync<='0';***

***else***

***hsync<='1';***

***end if;***

***if vpos>0 and vpos<4 then --vertical sync are low intre front porch si back porch***

***vsync<='0';***

***else***

***vsync<='1';***

***end if;***

De asemenea , pinii care sunt responsabili de afisarea culorilor (r,g,b) au valori de 0 logic intre inceputul de la front porch si sfaritul de la back porch

***if ( hpos>0 and hpos<408 ) or ( vpos>0 and vpos<48 ) then --intre inceputul de la fp si sfarsitul de la bp --rgb primeste low***

***R<=(others=>'0');***

***G<=(others=>'0');***

***B<=(others=>'0');***

***end if;***

##### **5**.**Rezultate experimentale**

Rezultatele consta in afisarea a 4 linii (doua verticale , doua orizontale ) care arata la un nivel de baza functionalitatea metodei folosite . Aceste rezultate au fost obtinute prin utilizarea unui monitor VGA cu port VGA de rezolutie 1280 x 1024 ,proiectul fiind functional si pe rezolutii appropriate ( concept testat experimental ) , si o placuta programabila Nexys4 ddr a carui cod a fost scris si incarcat din mediul de dezvoltare Vivado Design Suite 2016.4

###### **6.Concluzii**

Afisarea imaginilor prin portul VGA este o tehnica folositoare oricarui inginer.

Dezavantajul acestui proiect este faptul ca imaginea afisata pentru demonstrarea conceptului este simpla si implementata manual in memorie , de aici putand fii extinsa functionalitatea lui pentru procesarea imaginilor mai complexe.

**Bibliografie**

<https://www.the-home-cinema-guide.com/vga-connector.html>

<https://components101.com/connectors/vga-connector-pinout-datasheet>

<https://en.wikipedia.org/wiki/VGA_connector>

<https://uk.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/vga-cable-guide>

<https://users.utcluj.ro/~baruch/resources/Digilent/nexys4ddr_rm.pdf>

<https://www.youtube.com/watch?v=WK5FT5RD1sU&t=1076s>

<https://www.youtube.com/watch?v=wzhDRIX2Ors>

**Anexa**

Codul folosit:

nexys4 :

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity nexys4 is

Port

(

signal clk:in std\_logic;

signal VGA\_R:out std\_logic\_vector(3 downto 0);

signal VGA\_G:out std\_logic\_vector(3 downto 0);

signal VGA\_B:out std\_logic\_vector(3 downto 0);

signal VGA\_HS:out std\_logic;

signal VGA\_VS:out std\_logic

);

end nexys4;

architecture Behavioral of nexys4 is

component VGA is

Port

(

signal clk:in std\_logic;

signal VGA\_HS:out std\_logic;

signal VGA\_VS:out std\_logic;

signal VGA\_R:out std\_logic\_vector(3 downto 0);

signal VGA\_G:out std\_logic\_vector(3 downto 0);

signal VGA\_B:out std\_logic\_vector(3 downto 0)

);

end component VGA;

begin

vgaPM: VGA port map(clk=>clk,VGA\_HS=>VGA\_HS,VGA\_VS=>VGA\_VS,VGA\_R=>VGA\_R,VGA\_G=>VGA\_G,VGA\_B=>VGA\_B);

end Behavioral;

VGA :

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

entity VGA is

Port

(

signal clk:in std\_logic; --24 mhz

signal VGA\_HS:out std\_logic;

signal VGA\_VS:out std\_logic;

signal VGA\_R:out std\_logic\_vector(3 downto 0);

signal VGA\_G:out std\_logic\_vector(3 downto 0);

signal VGA\_B:out std\_logic\_vector(3 downto 0)

);

end VGA;

architecture Behavioral of VGA is

component SYNC is

Port

(

signal clk: in std\_logic;

signal hsync:out std\_logic;

signal vsync:out std\_logic;

signal r:out std\_logic\_vector(3 downto 0);

signal g:out std\_logic\_vector(3 downto 0);

signal b:out std\_logic\_vector(3 downto 0)

);

end component;

begin

c1: SYNC port map(clk=>clk,hsync=>vga\_hs,vsync=>vga\_vs,r=>vga\_r,g=>vga\_g,b=>vga\_b);

end Behavioral;

SYNC :

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.std\_logic\_unsigned.ALL;

entity SYNC is

Port

(

signal clk : in std\_logic;

signal hsync,vsync : out std\_logic;

signal r,g,b : out std\_logic\_vector(3 downto 0)

);

end SYNC;

architecture Behavioral of SYNC is

signal HPOS: integer range 0 to 1688:=0;

signal VPOS: integer range 0 to 1066:=0;

--signal Xpos:integer range 0 to 1688:=0;

--signal Ypos:integer range 0 to 1066:=0;

begin

process(clk)

begin

if clk'event and clk='1' then

-------desenam o linie orizontala in mijlocul ecranului (mijloc h => offset plus suze/2 adica 160 + 640/2 = 480)

-------desenam o linie verticala pe mijlocul ecranului (mijloc v=> offset plus suze/2 adica 41 + 480/2 = 281)or hpos=240 or vpos=140 or vpos=420

if hpos=1048 or vpos=554 or hpos=524 or vpos=277 then

r<=(others=>'1');

g<=(others=>'1');

b<=(others=>'1');

else

r<=(others=>'0');

g<=(others=>'0');

b<=(others=>'0');

end if;

--------------------------------------------

if hpos<1688 then

HPOS<=HPOS+1;

else

HPOS<=0;

if vpos<1066 then

vpos<=vpos+1; ---scanarea ecranului linie cu linie

else

vpos<=0;

end if;

end if;

------------------------------------------------

if hpos>48 and hpos<160 then ---horizontal sync are low intre front porch si back porch

hsync<='0';

else

hsync<='1';

end if;

if vpos>0 and vpos<4 then --vertical sync are low intre front porch si back porch

vsync<='0';

else

vsync<='1';

end if;

-------------------------------------------------

if ( hpos>0 and hpos<408 ) or ( vpos>0 and vpos<48 ) then --intre inceputul de la fp si sfarsitul de la bp rgb primeste low

R<=(others=>'0');

G<=(others=>'0');

B<=(others=>'0');

end if;

--------------------------------------------------

end if;

end process;

end Behavioral;

Constrangeri Nexys4 ddr :

## This file is a general .xdc for the Nexys4 DDR Rev. C

## 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 { clk }]; #IO\_L12P\_T1\_MRCC\_35 Sch=clk100mhz

create\_clock -add -name sys\_clk\_pin -period 10.00 -waveform {0 5} [get\_ports {clk}];

##Switches

#set\_property -dict { PACKAGE\_PIN J15 IOSTANDARD LVCMOS33 } [get\_ports { sw[0] }]; #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]

#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 { cat[0] }]; #IO\_L24N\_T3\_A00\_D16\_14 Sch=ca

#set\_property -dict { PACKAGE\_PIN R10 IOSTANDARD LVCMOS33 } [get\_ports { cat[1] }]; #IO\_25\_14 Sch=cb

#set\_property -dict { PACKAGE\_PIN K16 IOSTANDARD LVCMOS33 } [get\_ports { cat[2] }]; #IO\_25\_15 Sch=cc

#set\_property -dict { PACKAGE\_PIN K13 IOSTANDARD LVCMOS33 } [get\_ports { cat[3] }]; #IO\_L17P\_T2\_A26\_15 Sch=cd

#set\_property -dict { PACKAGE\_PIN P15 IOSTANDARD LVCMOS33 } [get\_ports { cat[4] }]; #IO\_L13P\_T2\_MRCC\_14 Sch=ce

#set\_property -dict { PACKAGE\_PIN T11 IOSTANDARD LVCMOS33 } [get\_ports { cat[5] }]; #IO\_L19P\_T3\_A10\_D26\_14 Sch=cf

#set\_property -dict { PACKAGE\_PIN L18 IOSTANDARD LVCMOS33 } [get\_ports { cat[6] }]; #IO\_L4P\_T0\_D04\_14 Sch=cg

#set\_property -dict { PACKAGE\_PIN H15 IOSTANDARD LVCMOS33 } [get\_ports { cat[7] }]; #IO\_L19N\_T3\_A21\_VREF\_15 Sch=dp

#set\_property -dict { PACKAGE\_PIN J17 IOSTANDARD LVCMOS33 } [get\_ports { an[0] }]; #IO\_L23P\_T3\_FOE\_B\_15 Sch=an[0]

#set\_property -dict { PACKAGE\_PIN J18 IOSTANDARD LVCMOS33 } [get\_ports { an[1] }]; #IO\_L23N\_T3\_FWE\_B\_15 Sch=an[1]

#set\_property -dict { PACKAGE\_PIN T9 IOSTANDARD LVCMOS33 } [get\_ports { an[2] }]; #IO\_L24P\_T3\_A01\_D17\_14 Sch=an[2]

#set\_property -dict { PACKAGE\_PIN J14 IOSTANDARD LVCMOS33 } [get\_ports { an[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]

##Buttons

#set\_property -dict { PACKAGE\_PIN C12 IOSTANDARD LVCMOS33 } [get\_ports { CPU\_RESETN }]; #IO\_L3P\_T0\_DQS\_AD1P\_15 Sch=cpu\_resetn

#set\_property -dict { PACKAGE\_PIN N17 IOSTANDARD LVCMOS33 } [get\_ports { btn[0] }]; #IO\_L9P\_T1\_DQS\_14 Sch=btnc

#set\_property -dict { PACKAGE\_PIN M18 IOSTANDARD LVCMOS33 } [get\_ports { btn[1] }]; #IO\_L4N\_T0\_D05\_14 Sch=btnu

#set\_property -dict { PACKAGE\_PIN P17 IOSTANDARD LVCMOS33 } [get\_ports { btn[2] }]; #IO\_L12P\_T1\_MRCC\_14 Sch=btnl

#set\_property -dict { PACKAGE\_PIN M17 IOSTANDARD LVCMOS33 } [get\_ports { btn[3] }]; #IO\_L10N\_T1\_D15\_14 Sch=btnr

#set\_property -dict { PACKAGE\_PIN P18 IOSTANDARD LVCMOS33 } [get\_ports { btn[4] }]; #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 LVDS } [get\_ports { XA\_N[1] }]; #IO\_L9N\_T1\_DQS\_AD3N\_15 Sch=xa\_n[1]

#set\_property -dict { PACKAGE\_PIN A13 IOSTANDARD LVDS } [get\_ports { XA\_P[1] }]; #IO\_L9P\_T1\_DQS\_AD3P\_15 Sch=xa\_p[1]

#set\_property -dict { PACKAGE\_PIN A16 IOSTANDARD LVDS } [get\_ports { XA\_N[2] }]; #IO\_L8N\_T1\_AD10N\_15 Sch=xa\_n[2]

#set\_property -dict { PACKAGE\_PIN A15 IOSTANDARD LVDS } [get\_ports { XA\_P[2] }]; #IO\_L8P\_T1\_AD10P\_15 Sch=xa\_p[2]

#set\_property -dict { PACKAGE\_PIN B17 IOSTANDARD LVDS } [get\_ports { XA\_N[3] }]; #IO\_L7N\_T1\_AD2N\_15 Sch=xa\_n[3]

#set\_property -dict { PACKAGE\_PIN B16 IOSTANDARD LVDS } [get\_ports { XA\_P[3] }]; #IO\_L7P\_T1\_AD2P\_15 Sch=xa\_p[3]

#set\_property -dict { PACKAGE\_PIN A18 IOSTANDARD LVDS } [get\_ports { XA\_N[4] }]; #IO\_L10N\_T1\_AD11N\_15 Sch=xa\_n[4]

#set\_property -dict { PACKAGE\_PIN B18 IOSTANDARD LVDS } [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