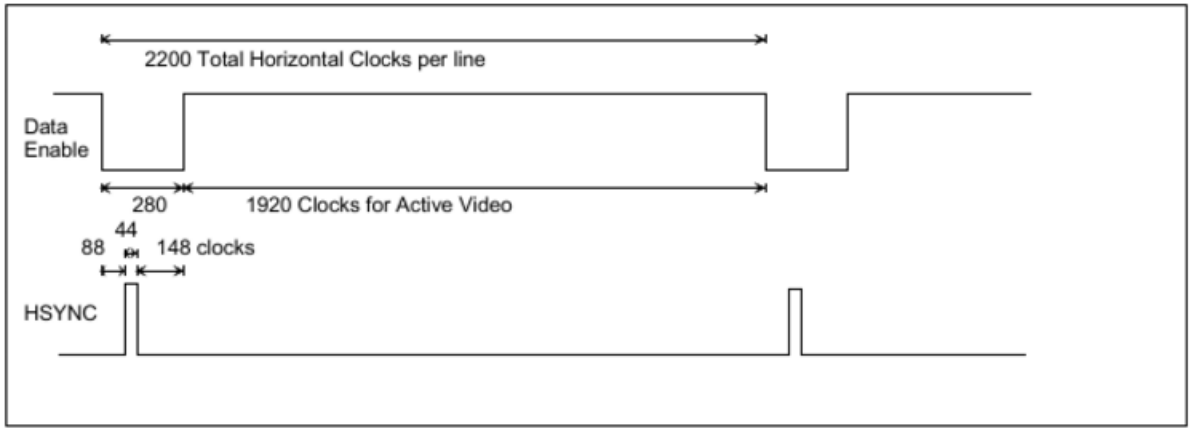
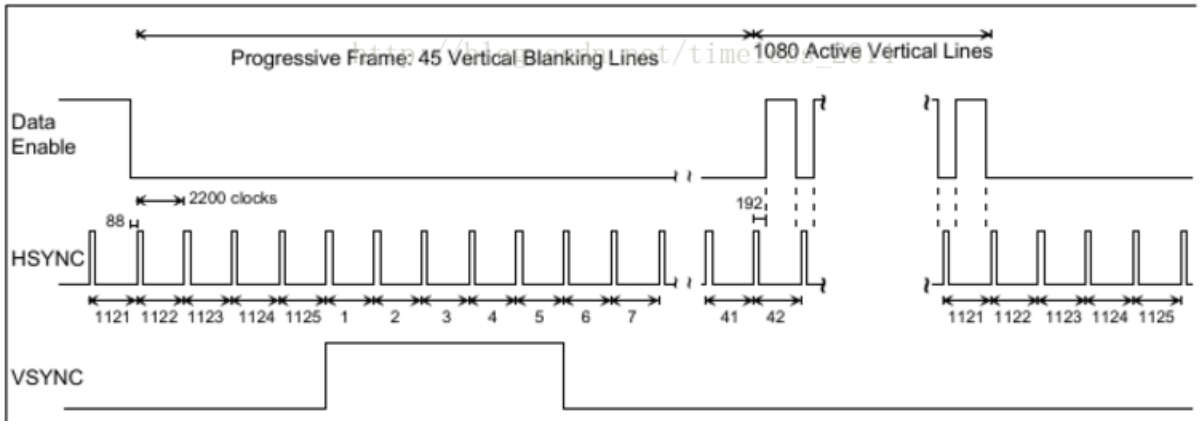
HDMI/DVI的传输协议

HDMI/DVI虽然已有多个版本，但其调用其IP核后的传输协议是相互兼容的，现有传输协议主要有CEA861和BT1120两种

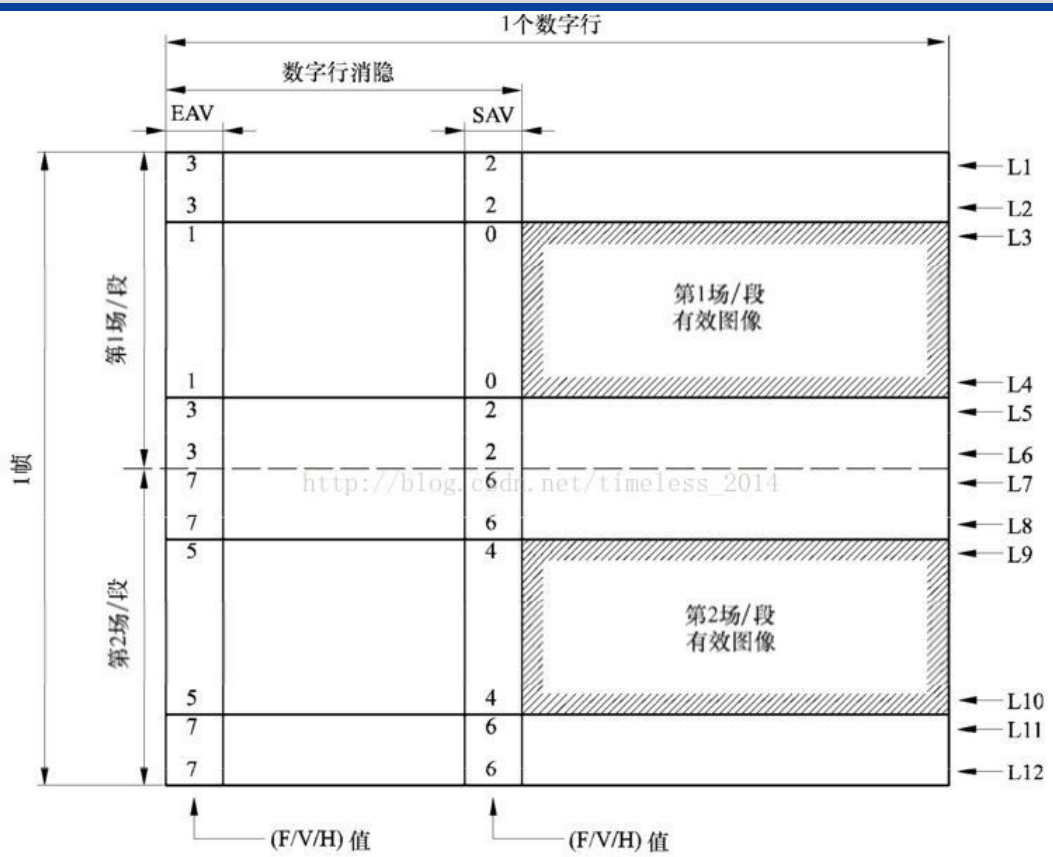
CEA861 时序：





HDMI/DVI的传输协议

BT1120 时序：

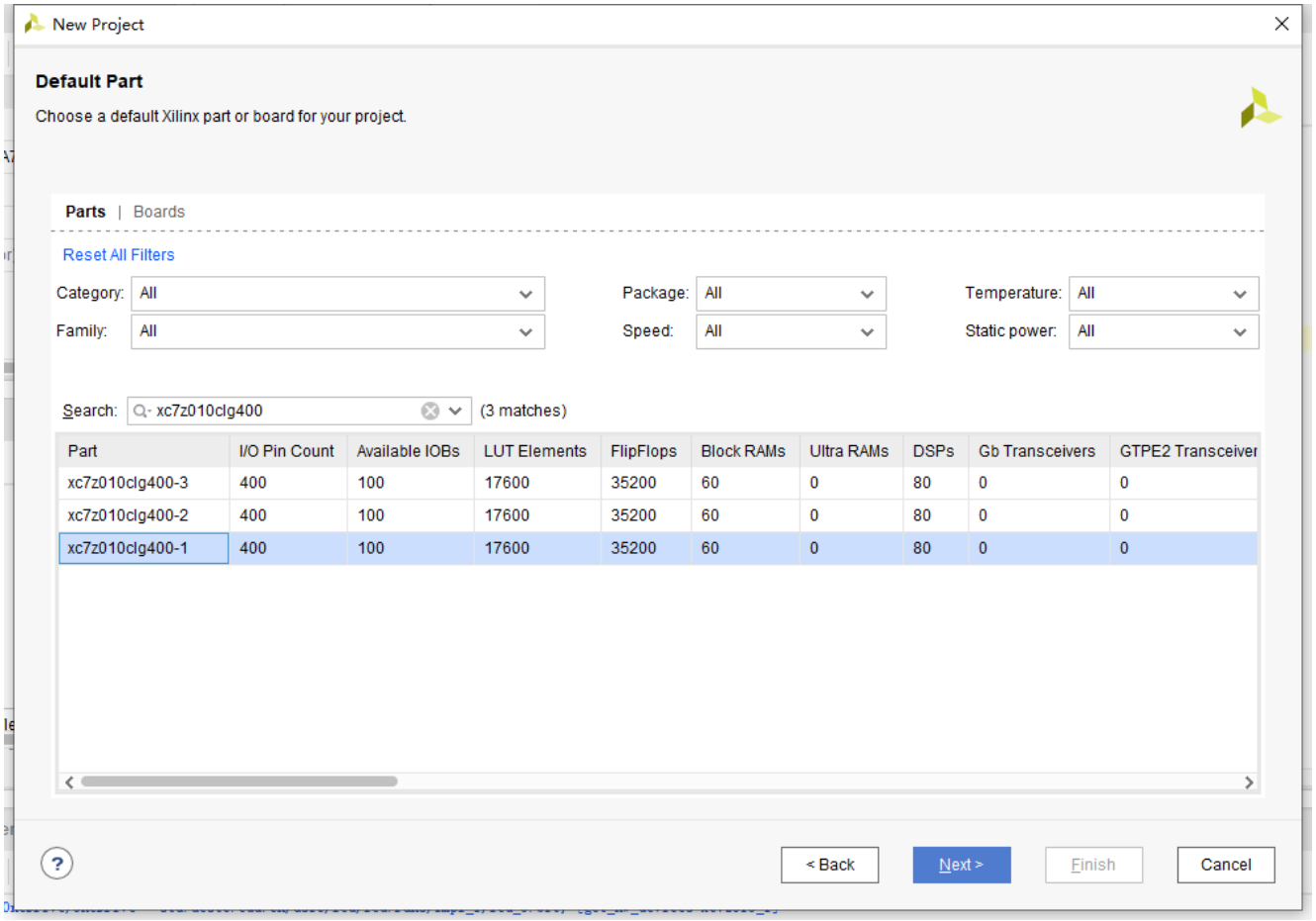


CEA-861是美国电子消费品制造商协会（Consumer Electronics Association，CEA）制定的“未压缩数字视频标准

BT1120是高清晰度电视演播室信号数字接口协议

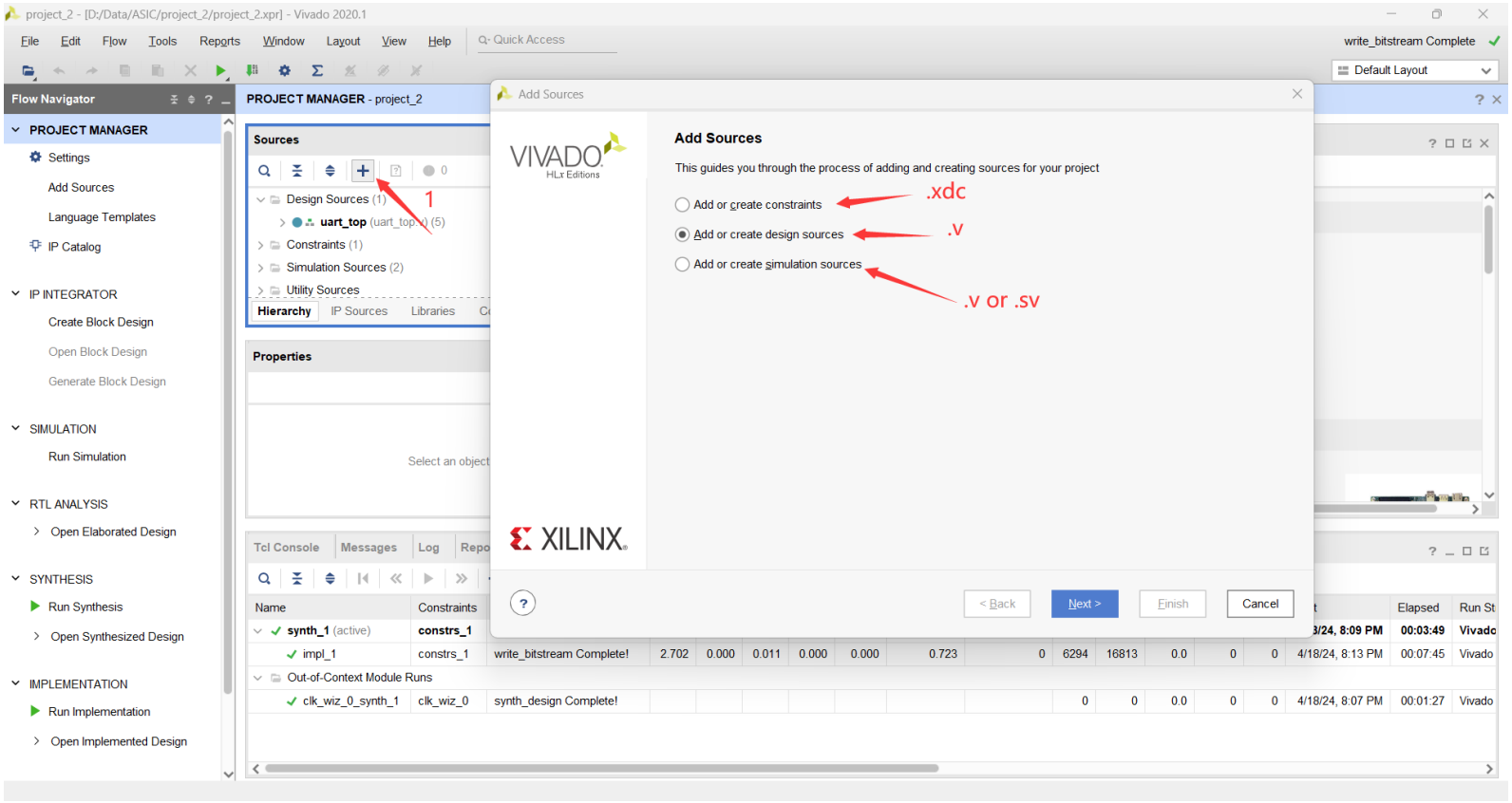
作业内容

器件型号：


New Project
Default Part
Choose a default Xilinx part or board for your project.
Parts | Boards
Reset All Filters
Category: All
Family: All
Package: All
Speed: All
Temperature: All
Static power: All
Search: xc7z010clg400
(3 matches)
Part | I/O Pin Count | Available IOBs | LUT Elements | FlipFlops | Block RAMs | Ultra RAMs | DSPs | Gb Transceivers | GTPE2 Transceiver
xc7z010clg400-3 | 400 | 100 | 17600 | 35200 | 60 | 0 | 80 | 0 | 0
xc7z010clg400-2 | 400 | 100 | 17600 | 35200 | 60 | 0 | 80 | 0 | 0
xc7z010clg400-1 | 400 | 100 | 17600 | 35200 | 60 | 0 | 80 | 0 | 0
< Back
Next >
Finish
Cancel

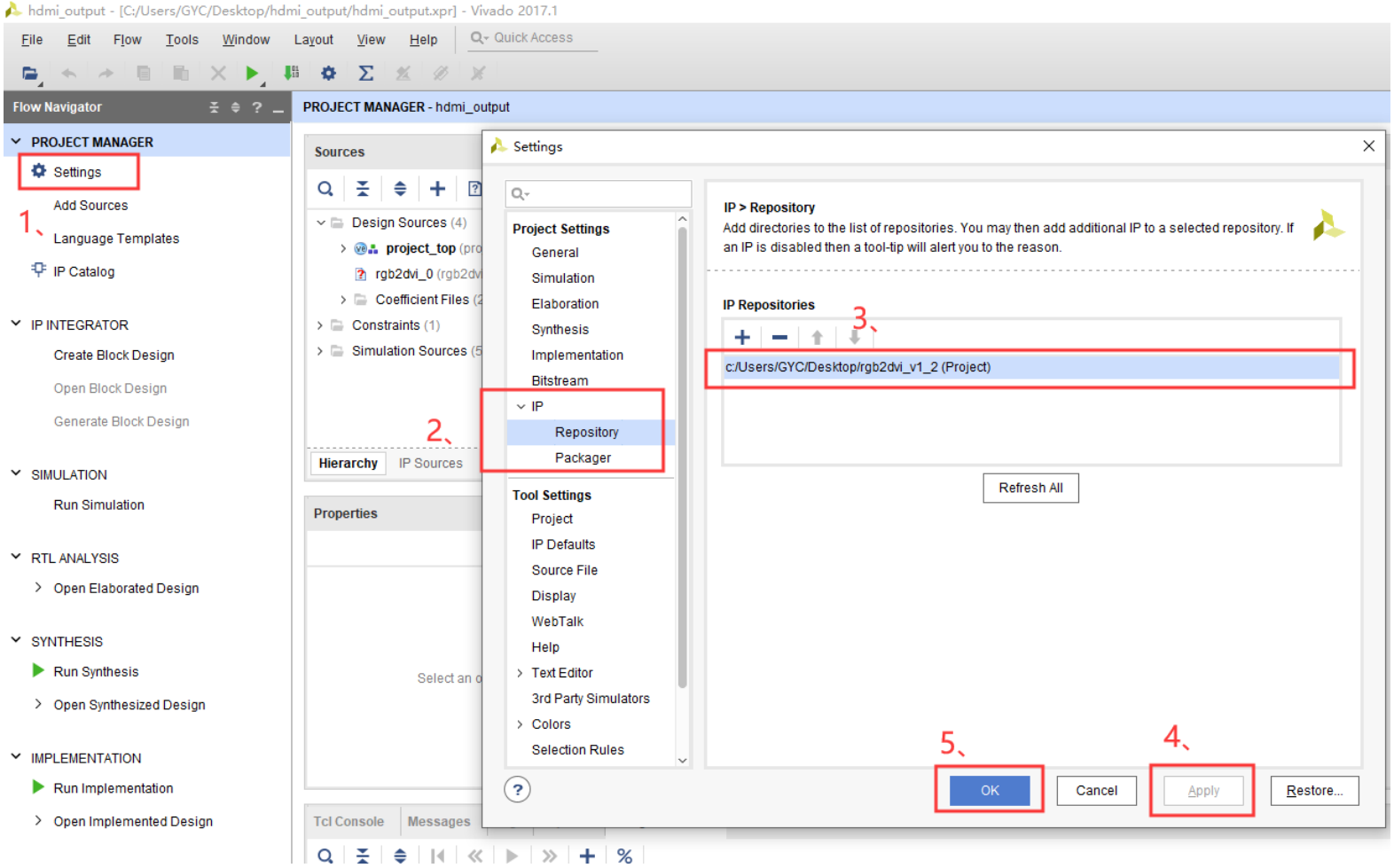
作业内容

导入源文件：


project_2 - [D:/Data/ASIC/project_2/project_2.xpr] - Vivado 2020.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
write_bitstream Complete
Default Layout
Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
SYNTHESIS
Run Synthesis
Open Synthesized Design
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROJECT MANAGER - project_2
Sources
Design Sources (1)
uart_top (uart_top.v) (5)
Constraints (1)
Simulation Sources (2)
Utility Sources
Hierarchy IP Sources Libraries
1
Properties
Select an object
Add Sources
VIVADO HLx Editions
Add Sources
This guides you through the process of adding and creating sources for your project
Add or create constraints
.xdc
Add or create design sources
.v
Add or create simulation sources
.v or .sv
XILINX
< Back
Next >
Finish
Cancel
Tcl Console Messages Log
Name Constraints
synth_1 (active) constrs_1
impl_1 constrs_1 write_bitstream Complete! 2.702 0.000 0.011 0.000 0.000 0.723 0 6294 16813 0.0 0 0 4/18/24, 8:13 PM 00:07:45 Vivado
Out-of-Context Module Runs
clk_wiz_0_synth_1 clk_wiz_0 synth_design Complete! 0 0 0.0 0 0 4/18/24, 8:07 PM 00:01:27 Vivado
Elapsed
Run St

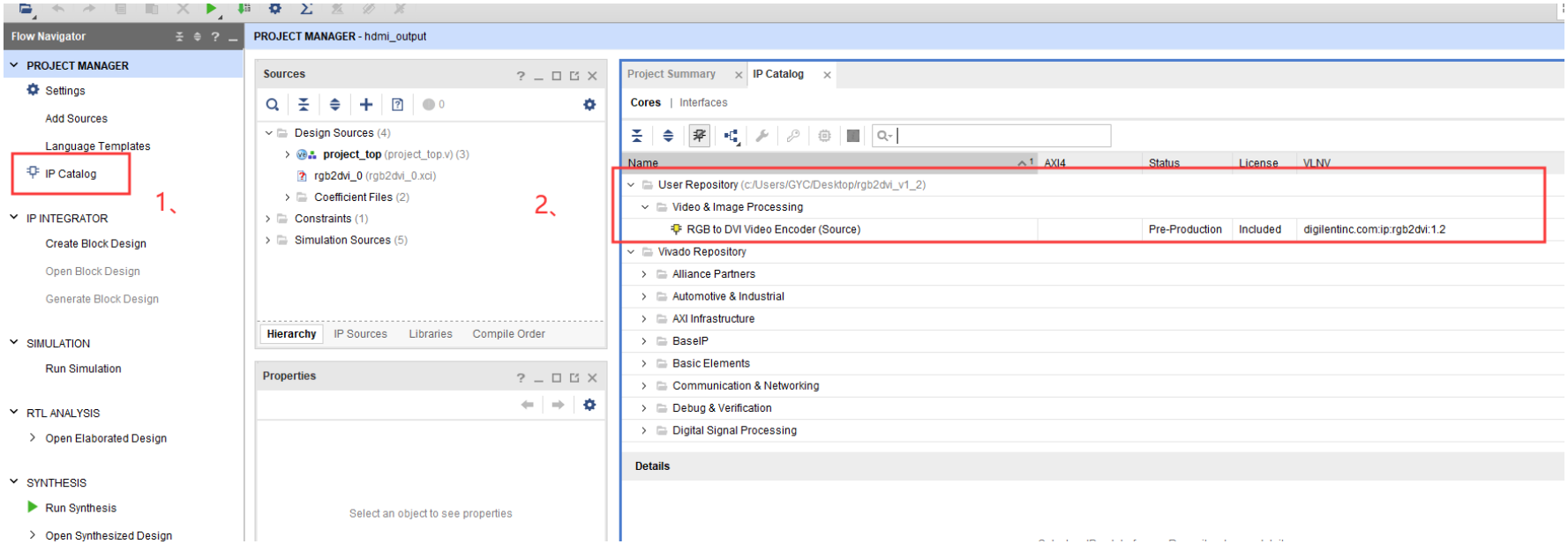
作业内容

添加外部IP核的方式：



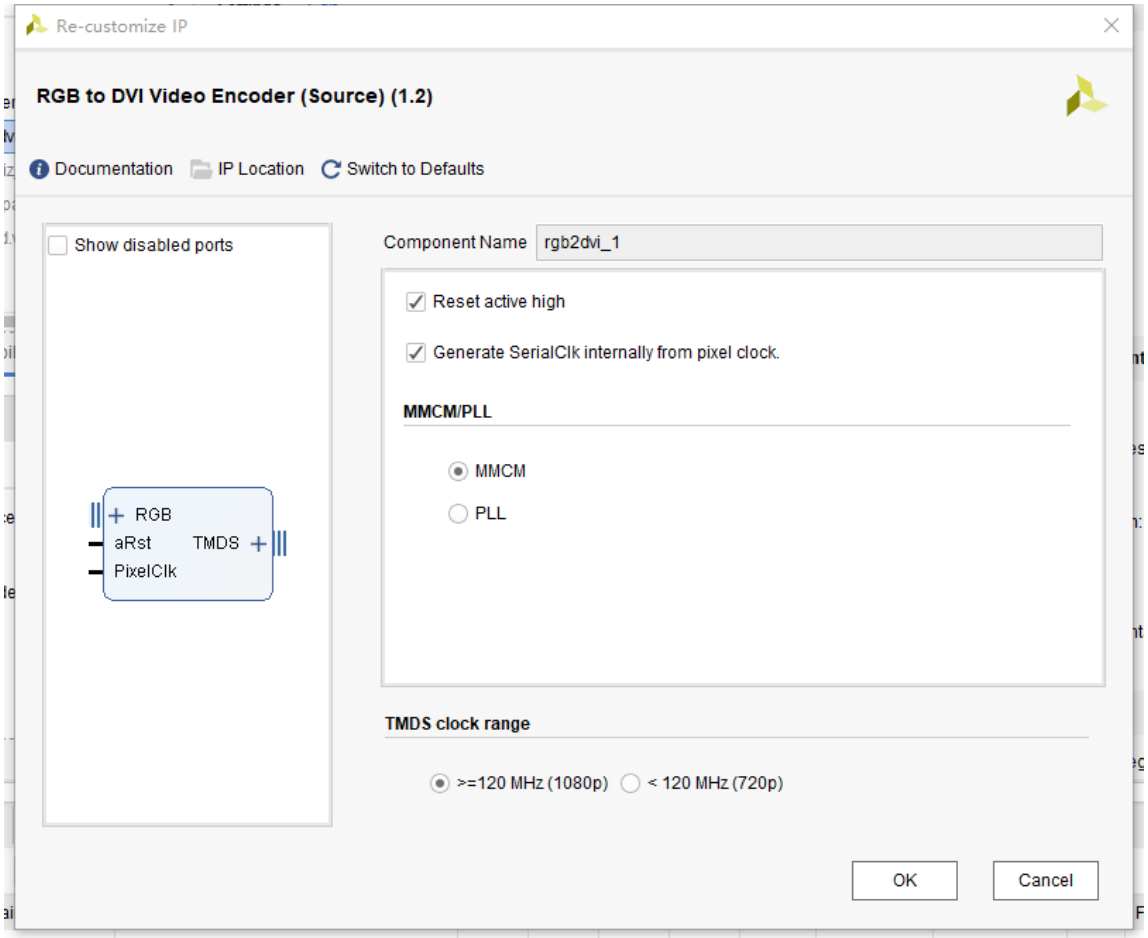
作业内容

添加外部IP核的方式：



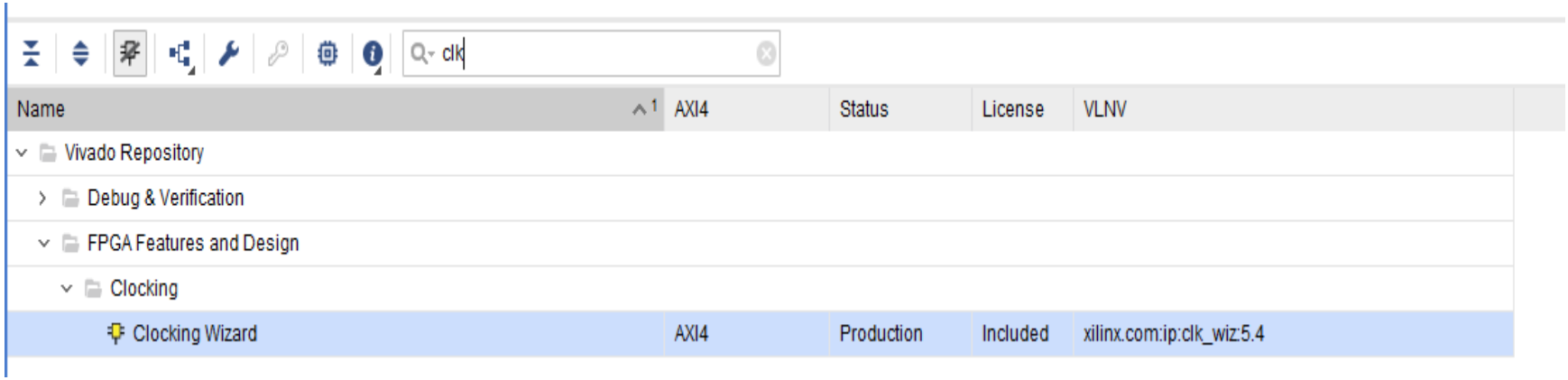
作业内容

HDMI/DVI 两个IP核的参数：



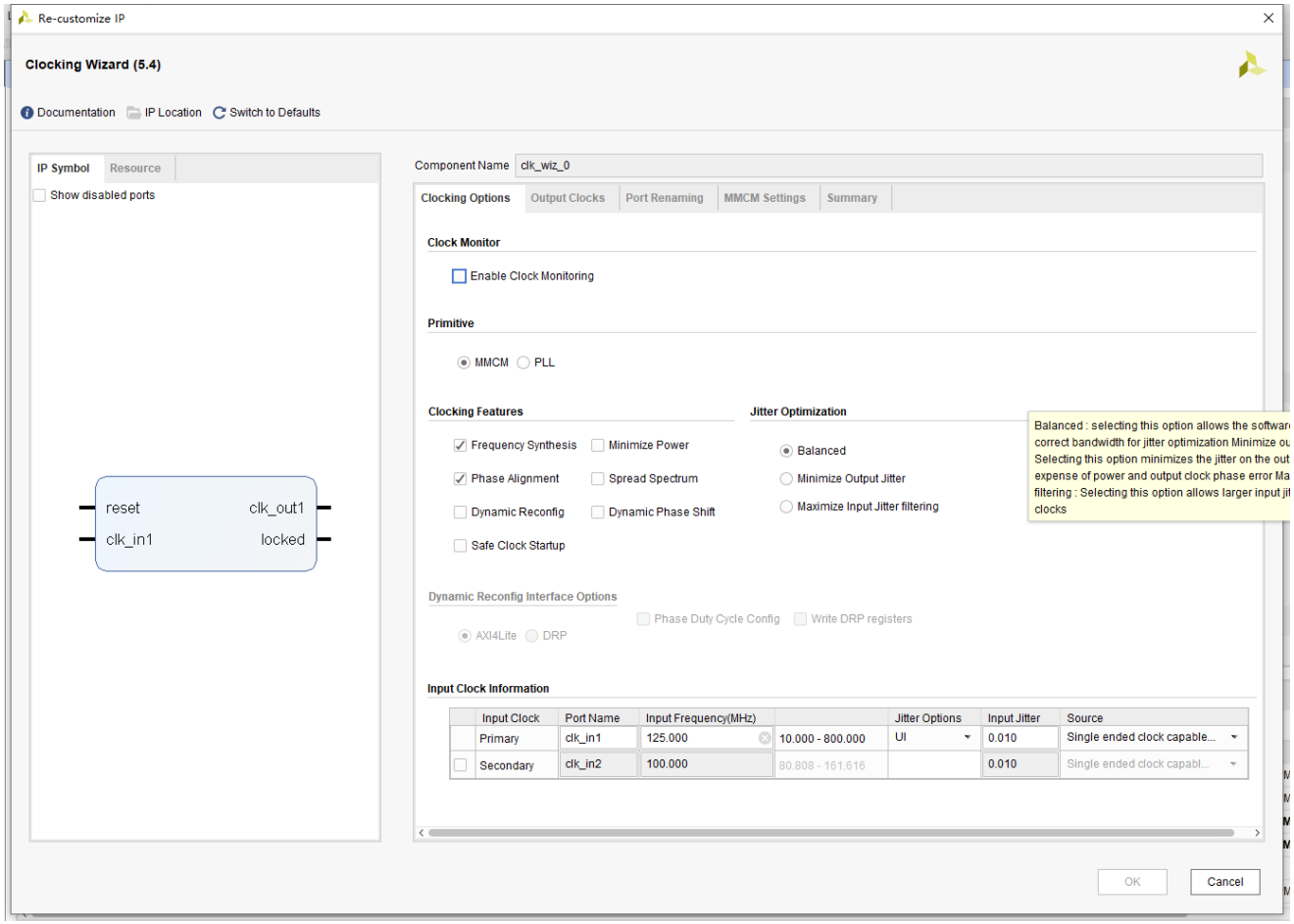
作业内容

添加外部IP核的方式：



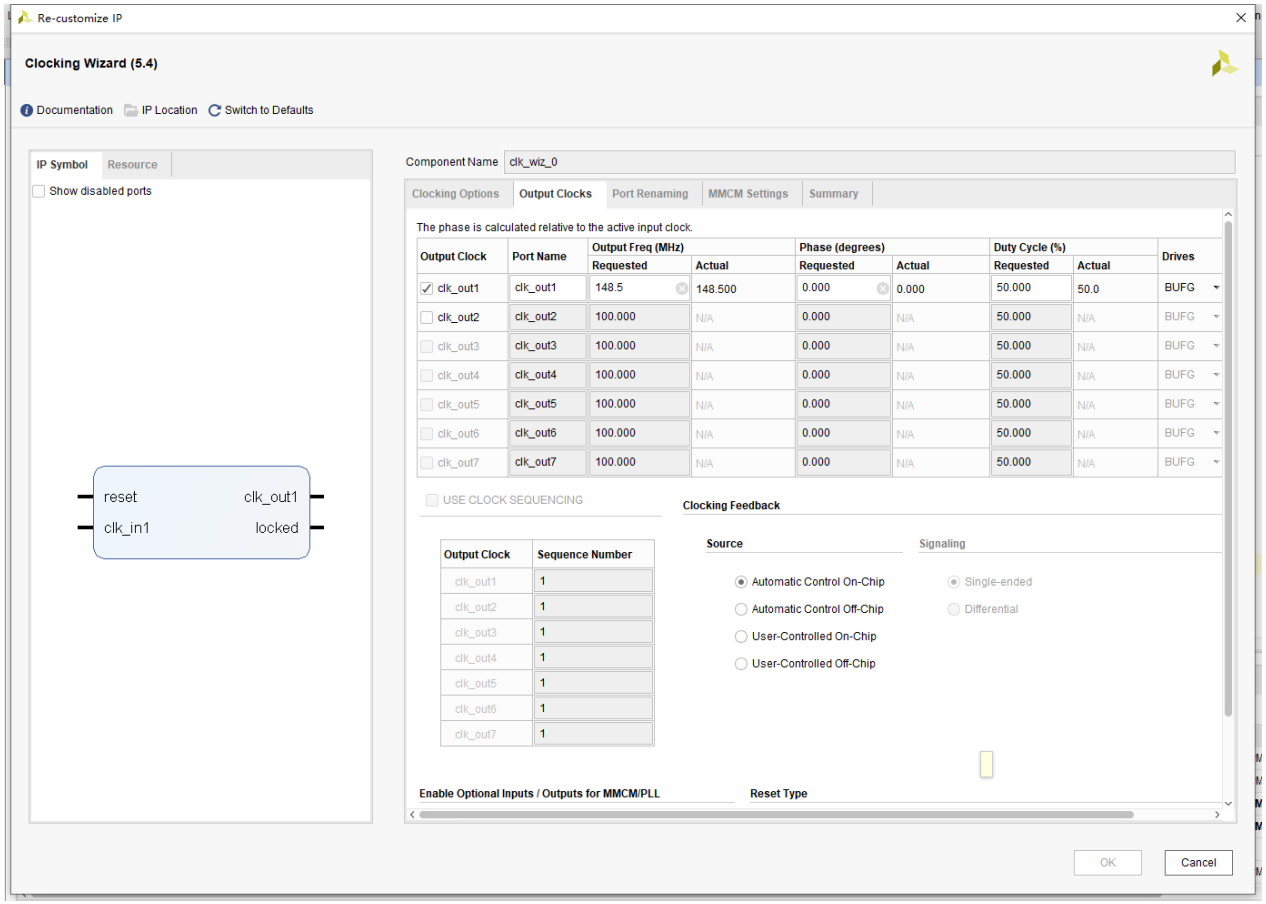
作业内容

HDMI/DVI 两个IP核的参数：



作业内容

HDMI/DVI 两个IP核的参数：



代码讲解

hdmi_controller.v

```
input           sys_clk      ;
input           rst_n        ;
input [23:0]    data         ;
output          clk_out      ;
output [2:0]    hdmi_out_data_n ;
output [2:0]    hdmi_out_data_p ;
output          hdmi_out_clk_p  ;
output          hdmi_out_clk_n  ;
output          hdmi_out_cec    ;
output [11:0]   col_pixel_number;
output [10:0]   row_pixel_number;
```

data为24位，
前8位为红色
中间8位为蓝色
后8位为绿色
如要打印纯红色
data=24’dFF_00_00

代码讲解

三段式状态机：**traffic_led.v**
三个**always**块

同步时序描述状态转移：

```
44      always@(posedge clk or posedge rst) begin
45          if(rst) begin
46              c_state <= GREEN;
47          end
48          else begin
49              c_state <= n_state;
50          end
51      end
```

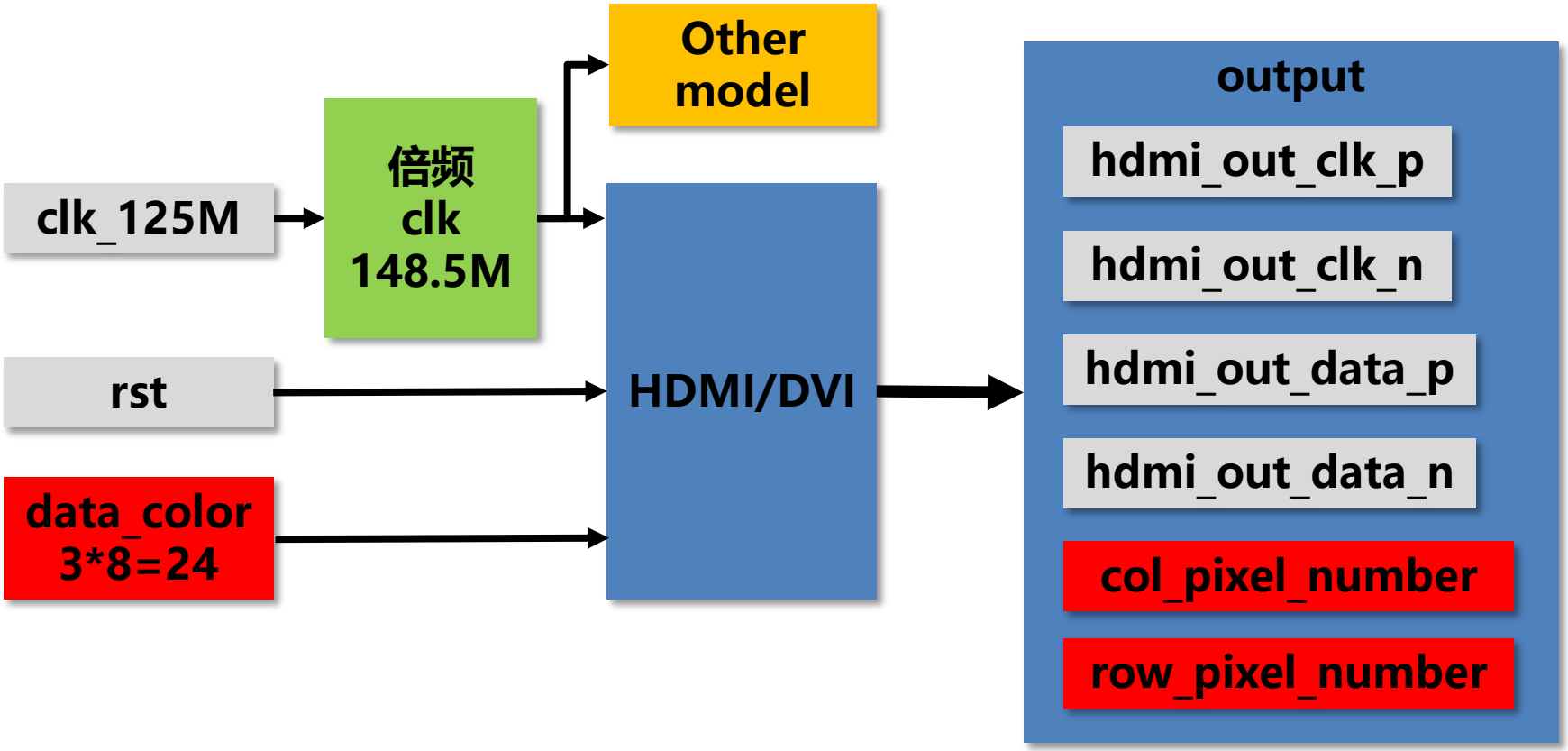
组合逻辑判断转移条件：

```
100     always @(*) begin
101         case(c_state)
102             GREEN: begin
103                 n_state <= (light_cnt == GREEN_CNT) ? YELLOW : GREEN;
104             end
105             YELLOW: begin
106                 n_state <= (light_cnt == YELLOW_CNT)  ? RED : YELLOW;
107             end
108             RED: begin
109                 n_state <= (light_cnt == RED_CNT) ? GREEN : RED;
110             end
111             default: begin
112                 n_state <= GREEN;
113             end
114         endcase
115     end
```

同步时序描述每个状态的输出：

```
53      always@(posedge clk or posedge rst) begin
54          if(rst) begin
55              green <= 1'b0;
56              red   <= 1'b0;
57              yellow<= 1'b0;
58              light_cnt <= 6'b0;
59          end
60          else begin
61              case(c_state)
62                  GREEN: begin
63                      if(light_cnt == GREEN_CNT) begin
64                          light_cnt <= 6'b0;
65                      end
66                      else begin
67                          light_cnt <= light_cnt + 6'b1;
68                      end
69                      green <= 1'b1;
70                      red   <= 1'b0;
71                      yellow<= 1'b0;
72                  end
73                  YELLOW: begin
74                      if(light_cnt == YELLOW_CNT) begin
75                          light_cnt <= 6'b0;
76                      end
77                      else begin
78                          light_cnt <= light_cnt + 6'b1;
79                      end
80                      green <= 1'b0;
81                      red   <= 1'b0;
82                      yellow<= 1'b1;
83                  end
84                  RED: begin
85                      if(light_cnt == RED_CNT) begin
86                          light_cnt <= 6'b0;
87                      end
88                      else begin
89                          light_cnt <= light_cnt + 6'b1;
90                      end
91                      green <= 1'b0;
92                      red   <= 1'b1;
93                      yellow<= 1'b0;
94                  end
95                  default:;
96              endcase
97          end
98      end
```

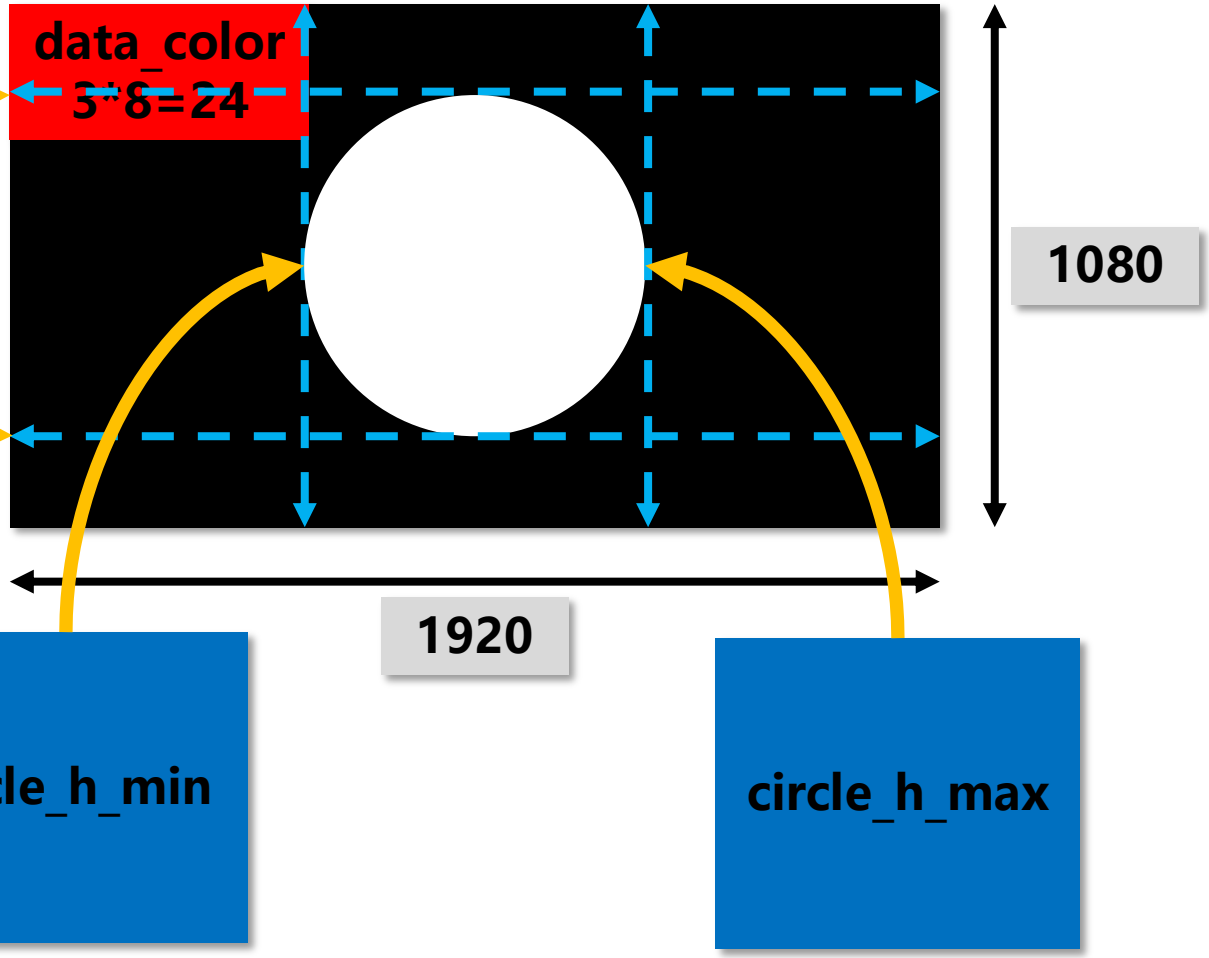
模块讲解


clk_125M
倍频
clk
148.5M
Other
model
rst
data_color
3*8=24
HDMI/DVI
output
hdmi_out_clk_p
hdmi_out_clk_n
hdmi_out_data_p
hdmi_out_data_n
col_pixel_number
row_pixel_number

模块讲解

1.col_pixel_number

2.row_pixel_number

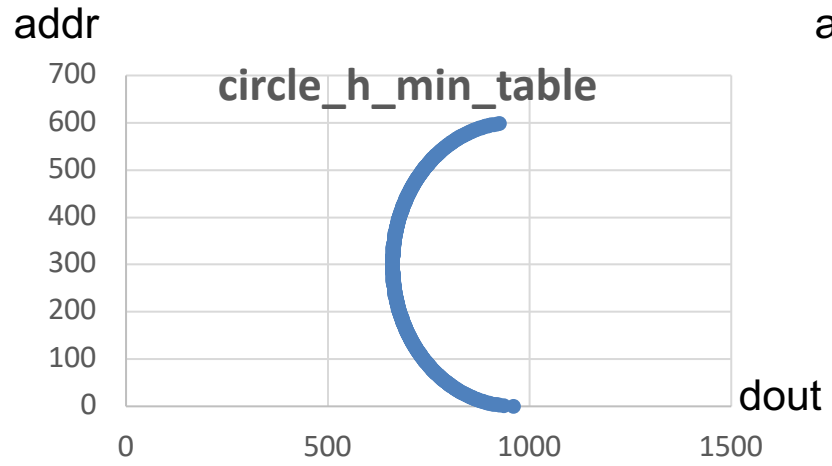


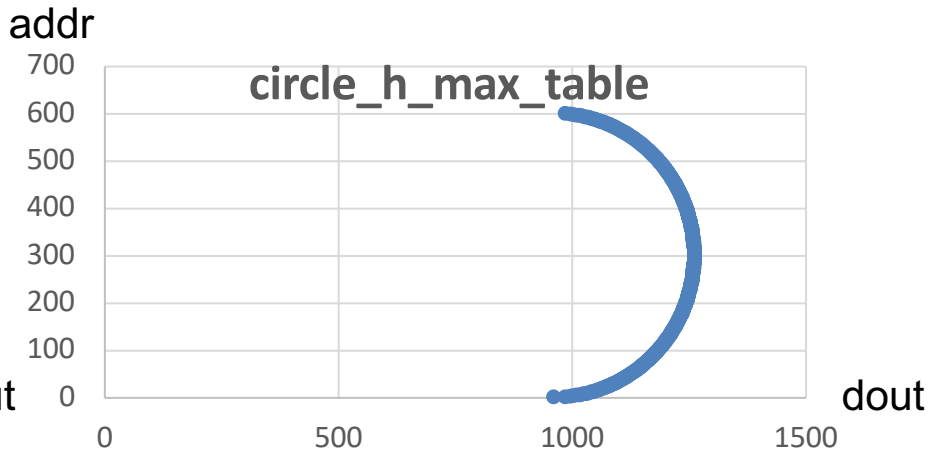
row:
(1080-600)/2=240

白:
24’ hFF_FF_FF
黑:
24’ h00_00_00
红:
24’ hFF_00_00
绿:
24’ h00_00_FF
黄:
24’ hFF_00_FF

模块讲解

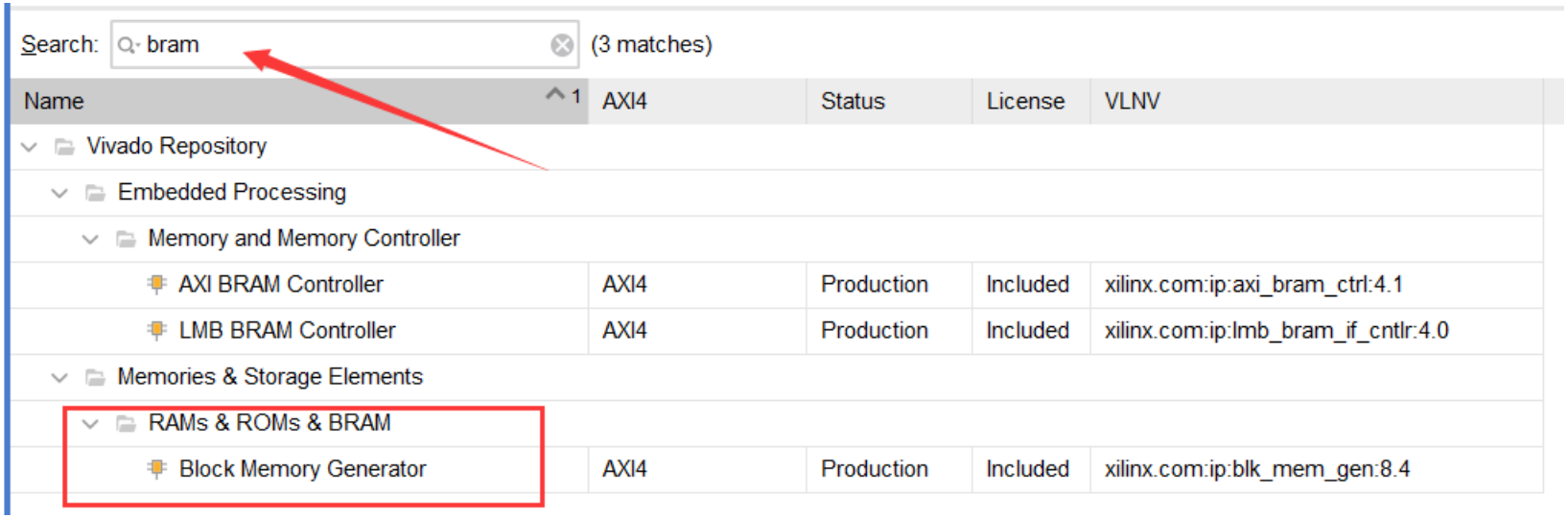
circle_h_max_table.coe和circle_h_min_table.coe



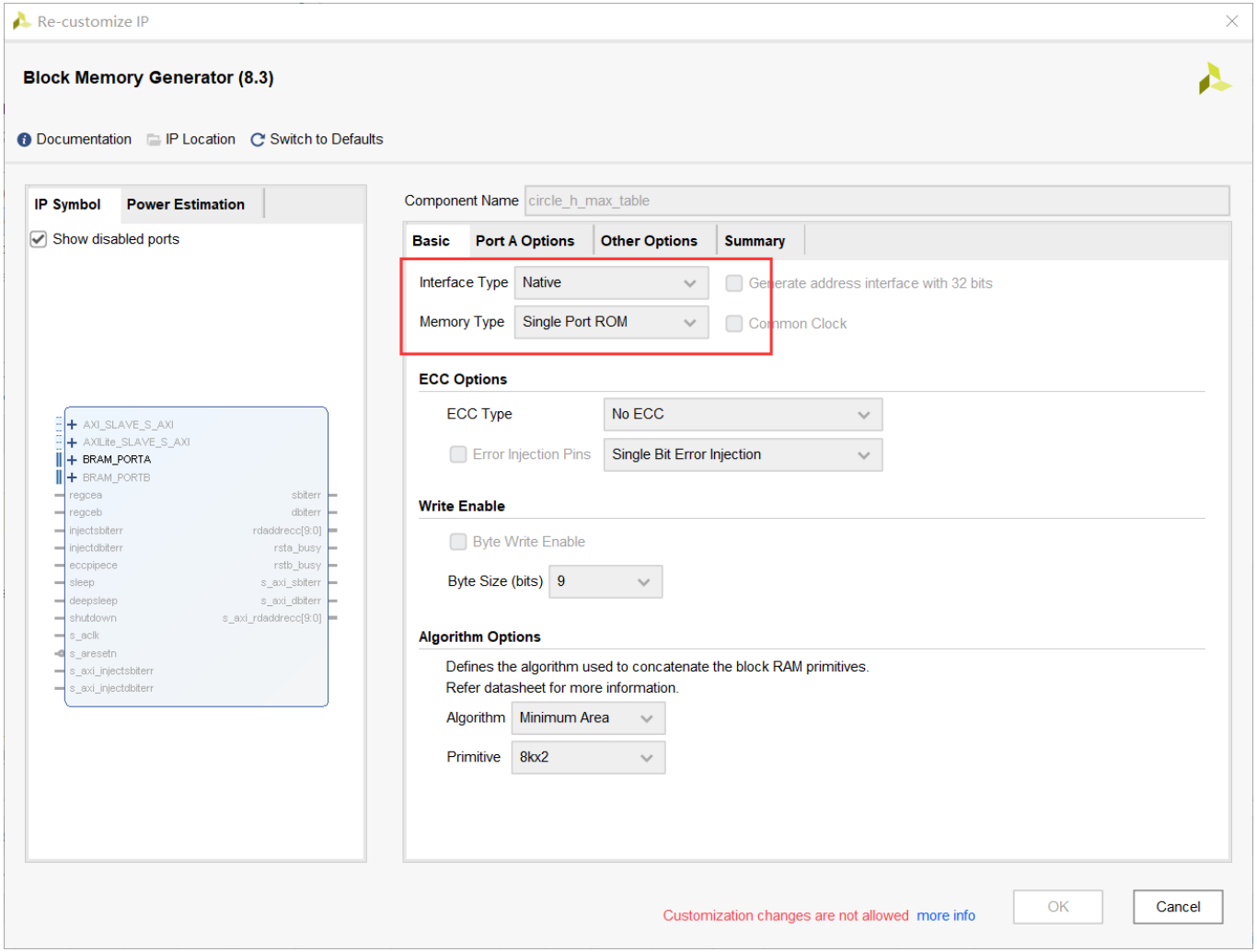



```
circle_h_max_table u_circle_h_max_table(
    .addra(addr[9:0]),
    .clka(clk),
    .douta(circle_h_max)
);
circle_h_min_table u_circle_h_min_table(
    .addra(addr[9:0]),
    .clka(clk),
    .douta(circle_h_min)
);
```

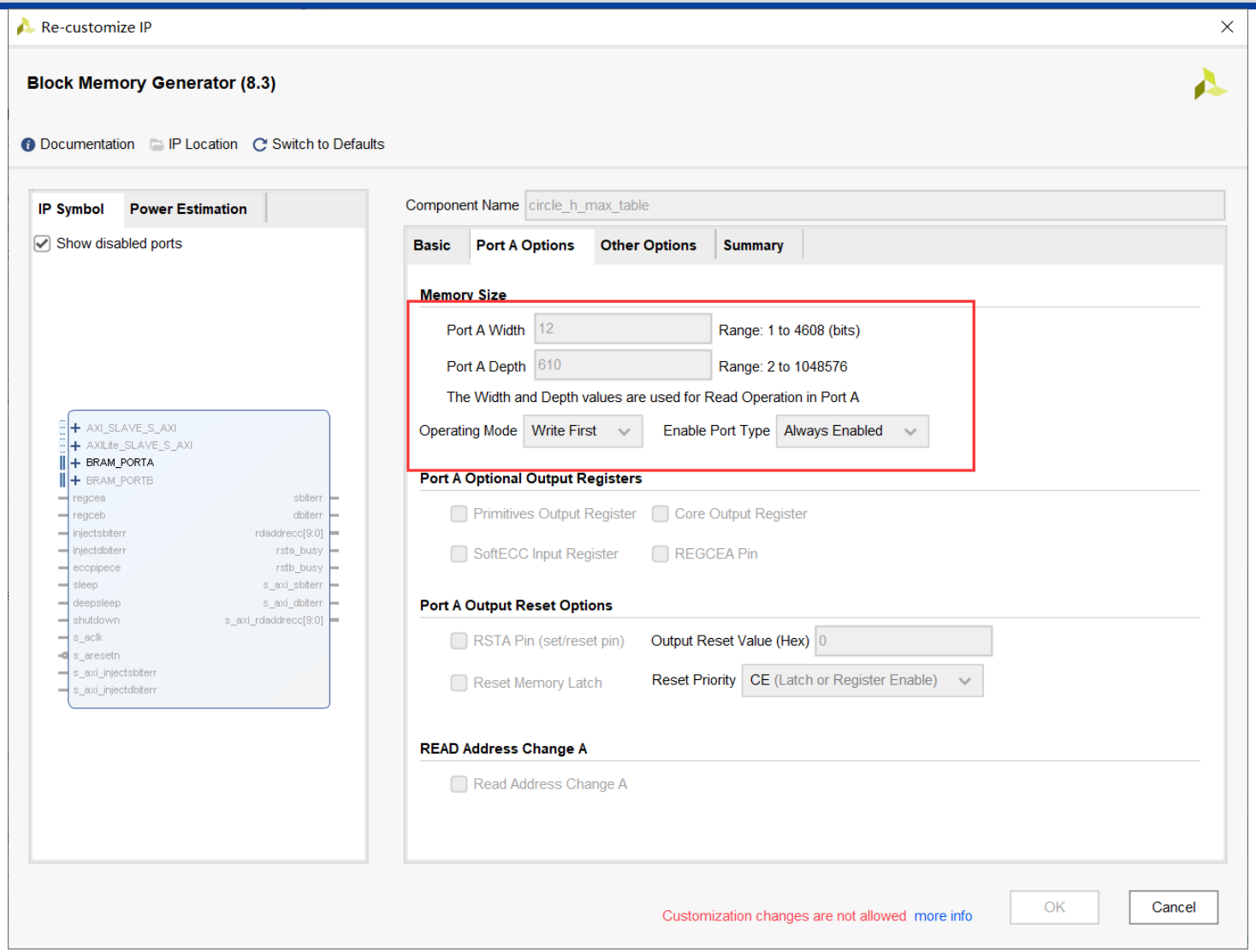
调用coe


Search: bram
(3 matches)
Name
AXI4
Status
License
VLNV
Vivado Repository
Embedded Processing
Memory and Memory Controller
AXI BRAM Controller
AXI4
Production
Included
xilinx.com:ip:axi_bram_ctrl:4.1
LMB BRAM Controller
AXI4
Production
Included
xilinx.com:ip:lmb_bram_if_cntlr:4.0
Memories & Storage Elements
RAMs & ROMs & BRAM
Block Memory Generator
AXI4
Production
Included
xilinx.com:ip:blk_mem_gen:8.4

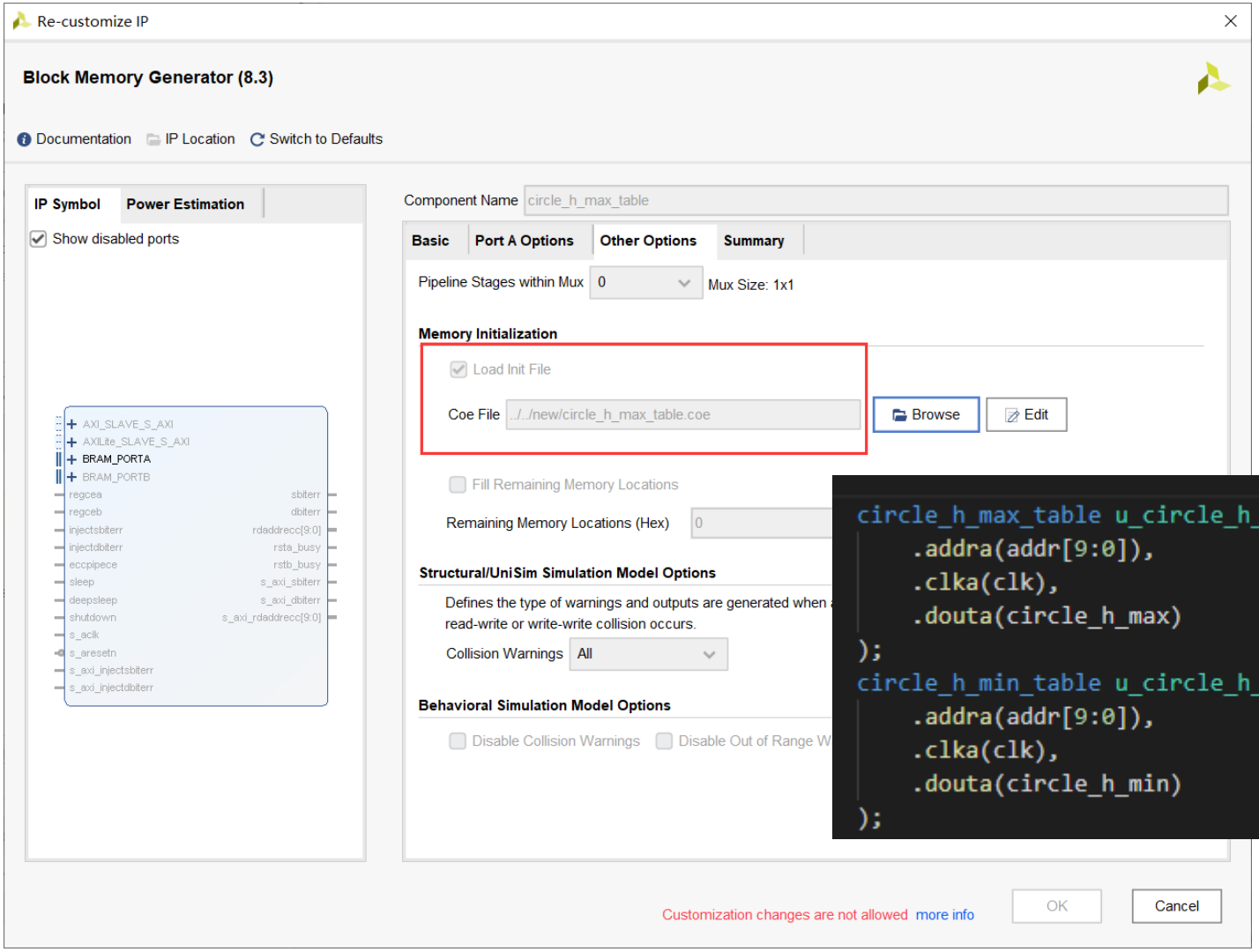
调用coe


Re-customize IP
Block Memory Generator (8.3)
Documentation
IP Location
Switch to Defaults
IP Symbol
Power Estimation
Show disabled ports
AXI_SLAVE_S_AXI
AXILite_SLAVE_S_AXI
BRAM_PORTA
BRAM_PORTB
regcea
regceb
injectsbiterr
injectdbiterr
eccpipece
sleep
deepsleep
shutdown
s_aclk
s_aresetn
s_axi_injectsbiterr
s_axi_injectdbiterr
sbiterr
dbiterr
rdaddrecc[9:0]
rsta_busy
rstb_busy
s_axi_sbiterr
s_axi_dbiterr
s_axi_rdaddrecc[9:0]
Component Name
circle_h_max_table
Basic
Port A Options
Other Options
Summary
Interface Type
Native
Generate address interface with 32 bits
Memory Type
Single Port ROM
Common Clock
ECC Options
ECC Type
No ECC
Error Injection Pins
Single Bit Error Injection
Write Enable
Byte Write Enable
Byte Size (bits)
9
Algorithm Options
Defines the algorithm used to concatenate the block RAM primitives.
Refer datasheet for more information.
Algorithm
Minimum Area
Primitive
8kx2
Customization changes are not allowed
more info
OK
Cancel

调用coe


Re-customize IP
Block Memory Generator (8.3)
Documentation
IP Location
Switch to Defaults
IP Symbol
Power Estimation
Show disabled ports
AXI_SLAVE_S_AXI
AXILite_SLAVE_S_AXI
BRAM_PORTA
BRAM_PORTB
regcea
regceb
injectsbiterr
injectdbiterr
eccpipece
sleep
deepsleep
shutdown
s_aclk
s_aresetn
s_axi_injectsbiterr
s_axi_injectdbiterr
sbiterr
dbiterr
rdaddrecc[9:0]
rsta_busy
rstb_busy
s_axi_sbiterr
s_axi_dbiterr
s_axi_rdaddrecc[9:0]
Component Name
circle_h_max_table
Basic
Port A Options
Other Options
Summary
Memory Size
Port A Width
12
Range: 1 to 4608 (bits)
Port A Depth
610
Range: 2 to 1048576
The Width and Depth values are used for Read Operation in Port A
Operating Mode
Write First
Enable Port Type
Always Enabled
Port A Optional Output Registers
Primitives Output Register
Core Output Register
SoftECC Input Register
REGCEA Pin
Port A Output Reset Options
RSTA Pin (set/reset pin)
Output Reset Value (Hex)
0
Reset Memory Latch
Reset Priority
CE (Latch or Register Enable)
READ Address Change A
Read Address Change A
Customization changes are not allowed
more info
OK
Cancel

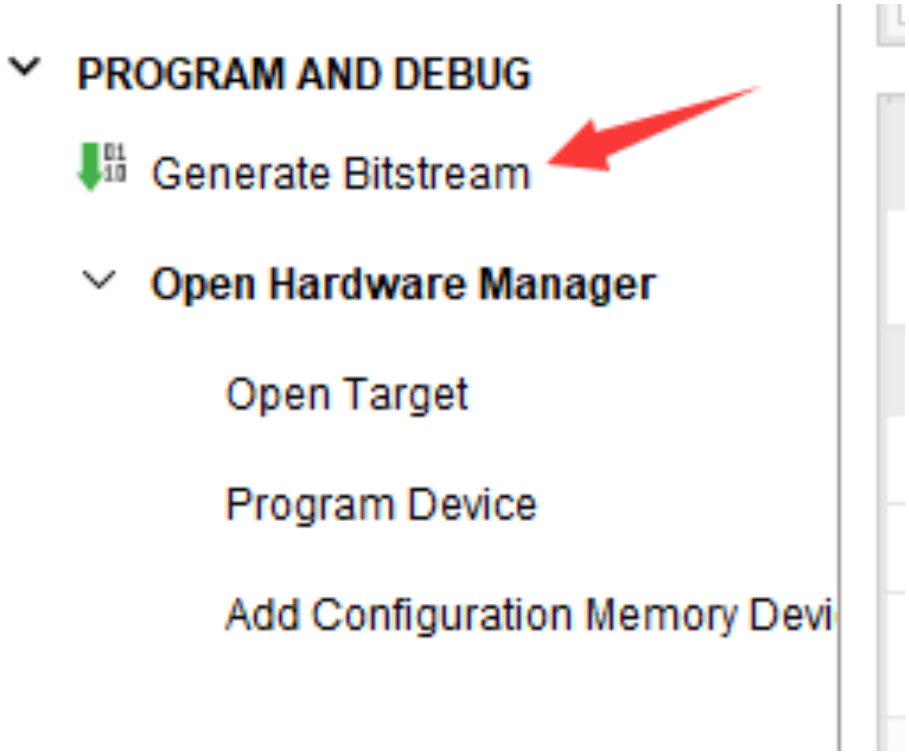
调用coe




```
circle_h_max_table u_circle_h_max_table(
    .addra(addr[9:0]),
    .clka(clk),
    .douta(circle_h_max)
);
circle_h_min_table u_circle_h_min_table(
    .addra(addr[9:0]),
    .clka(clk),
    .douta(circle_h_min)
);
```

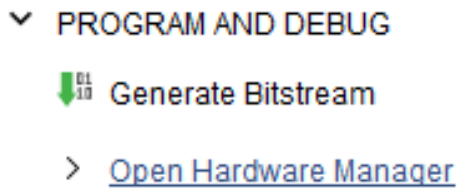
作业内容

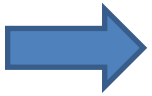
生成比特流文件：

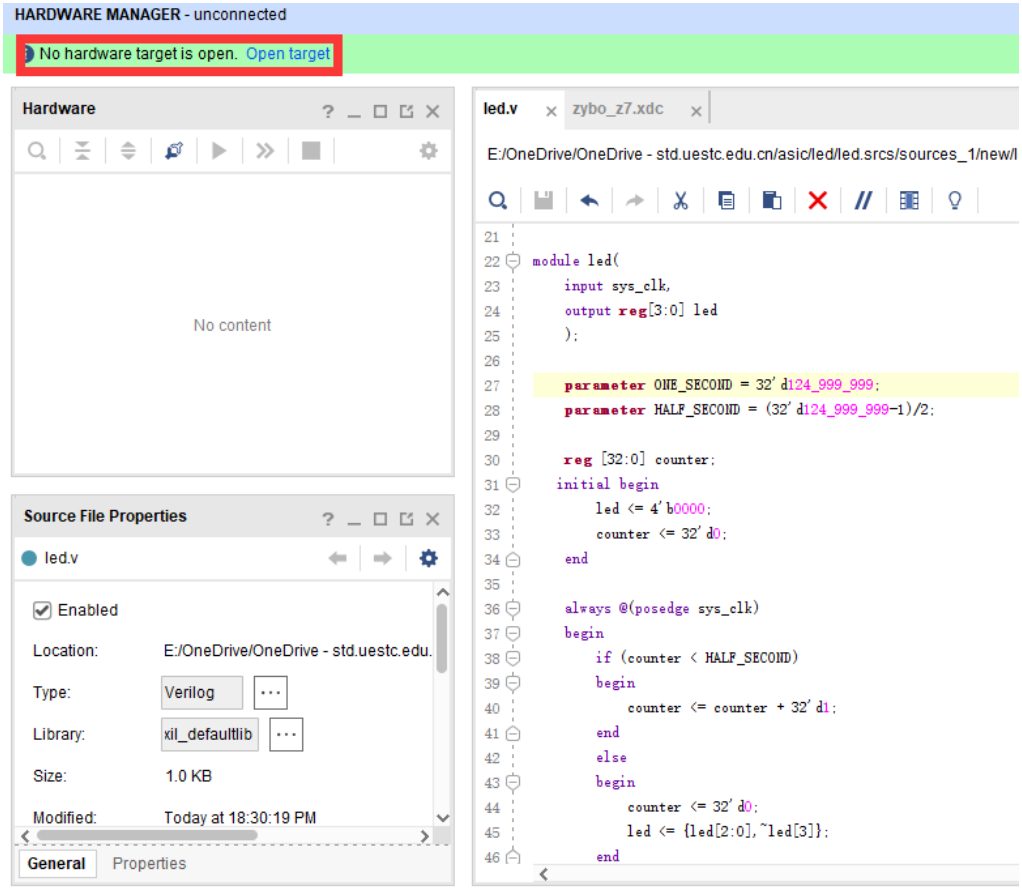


作业内容

编程下载：


PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager

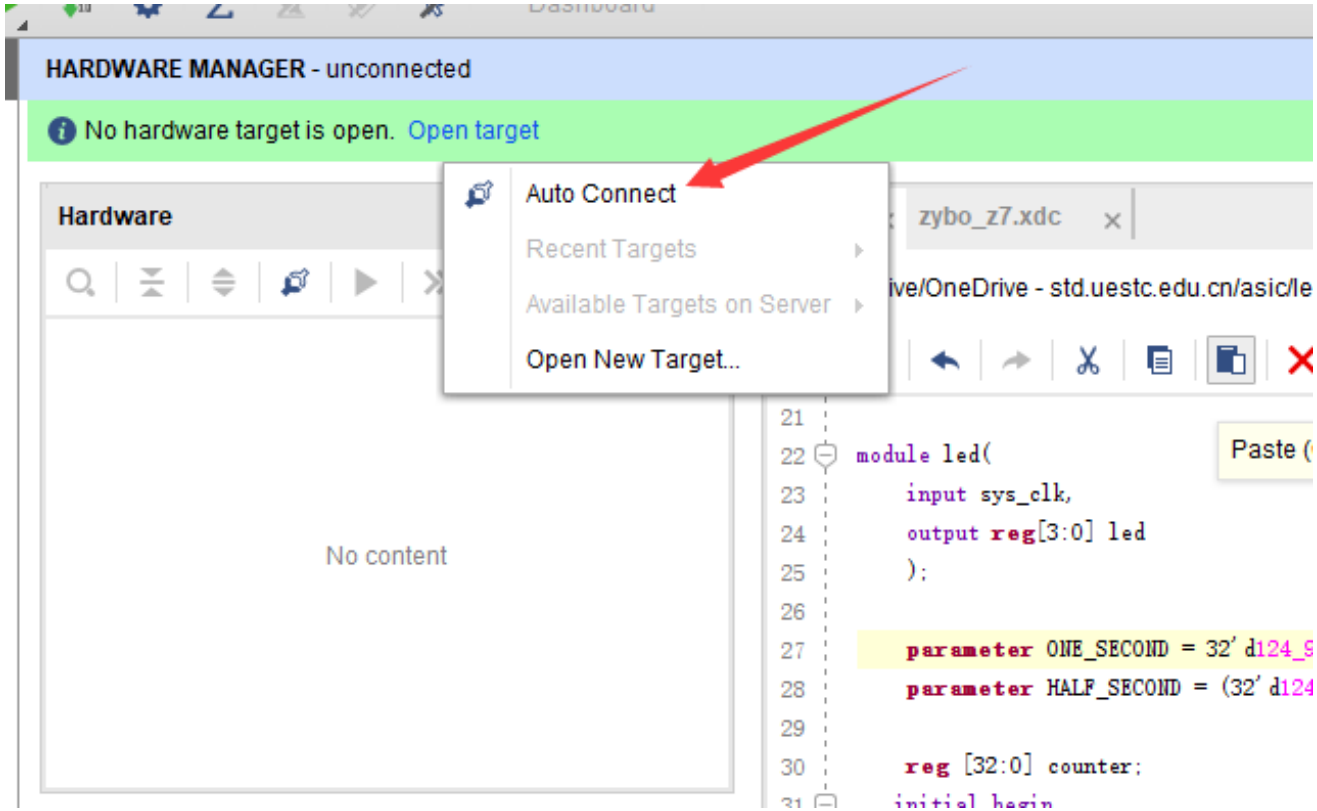



HARDWARE MANAGER - unconnected
No hardware target is open. Open target
Hardware
No content
Source File Properties
led.v
Enabled
Location: E:/OneDrive/OneDrive - std.uestc.edu.
Type: Verilog
Library: xil_defaultlib
Size: 1.0 KB
Modified: Today at 18:30:19 PM
General
Properties
led.v
zybo_z7.xdc
E:/OneDrive/OneDrive - std.uestc.edu.cn/asic/led/led.srcs/sources_1/new/l
21
22 module led(
23 input sys_clk,
24 output reg[3:0] led
25);
26
27 parameter ONE_SECOND = 32'd124_999_999;
28 parameter HALF_SECOND = (32'd124_999_999-1)/2;
29
30 reg [32:0] counter;
31 initial begin
32 led <= 4'b0000;
33 counter <= 32'd0;
34 end
35
36 always @(posedge sys_clk)
37 begin
38 if (counter < HALF_SECOND)
39 begin
40 counter <= counter + 32'd1;
41 end
42 else
43 begin
44 counter <= 32'd0;
45 led <= {led[2:0],~led[3]};
46 end

作业内容

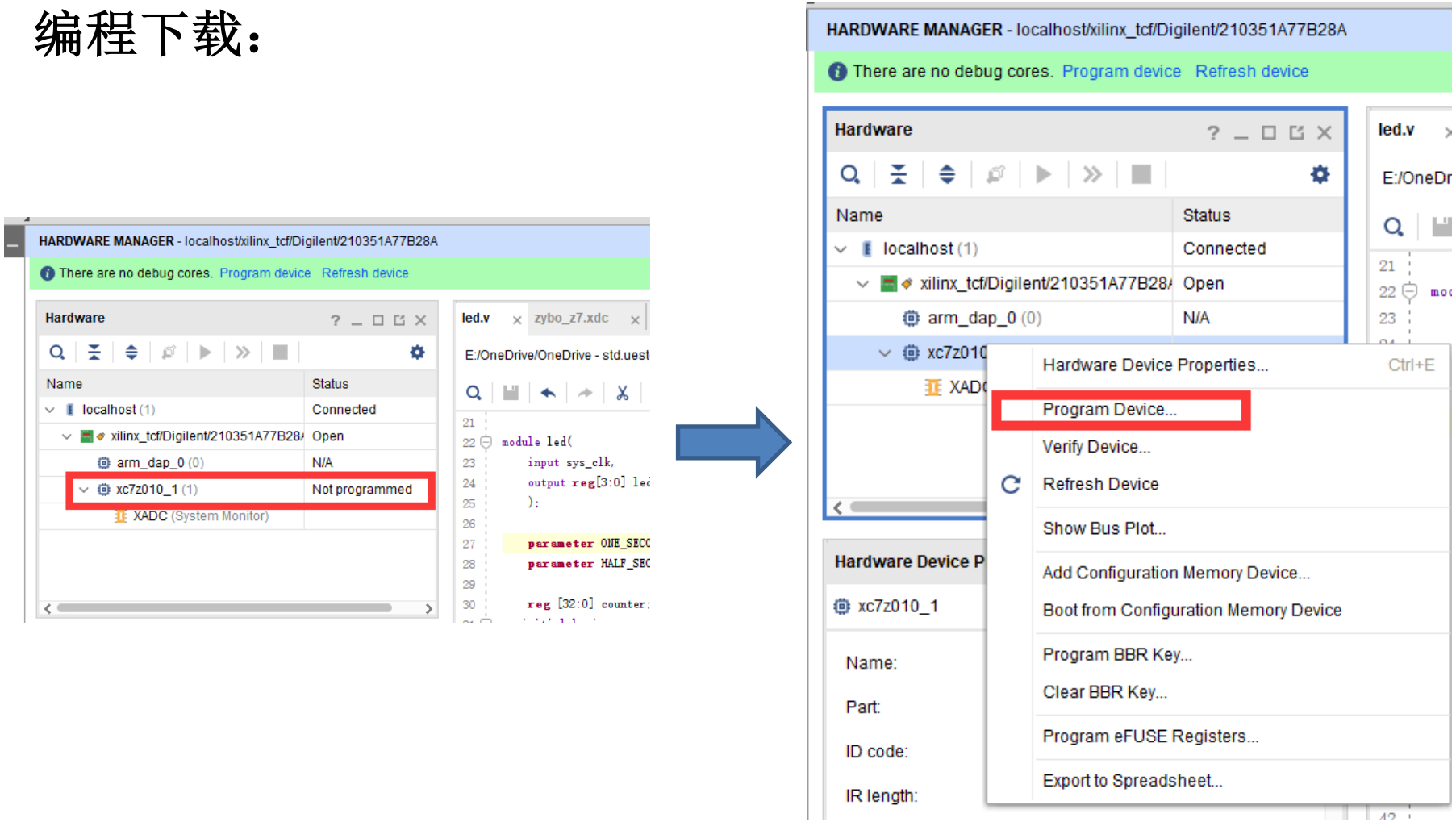
编程下载：

注意，请接通板卡电源并开机！



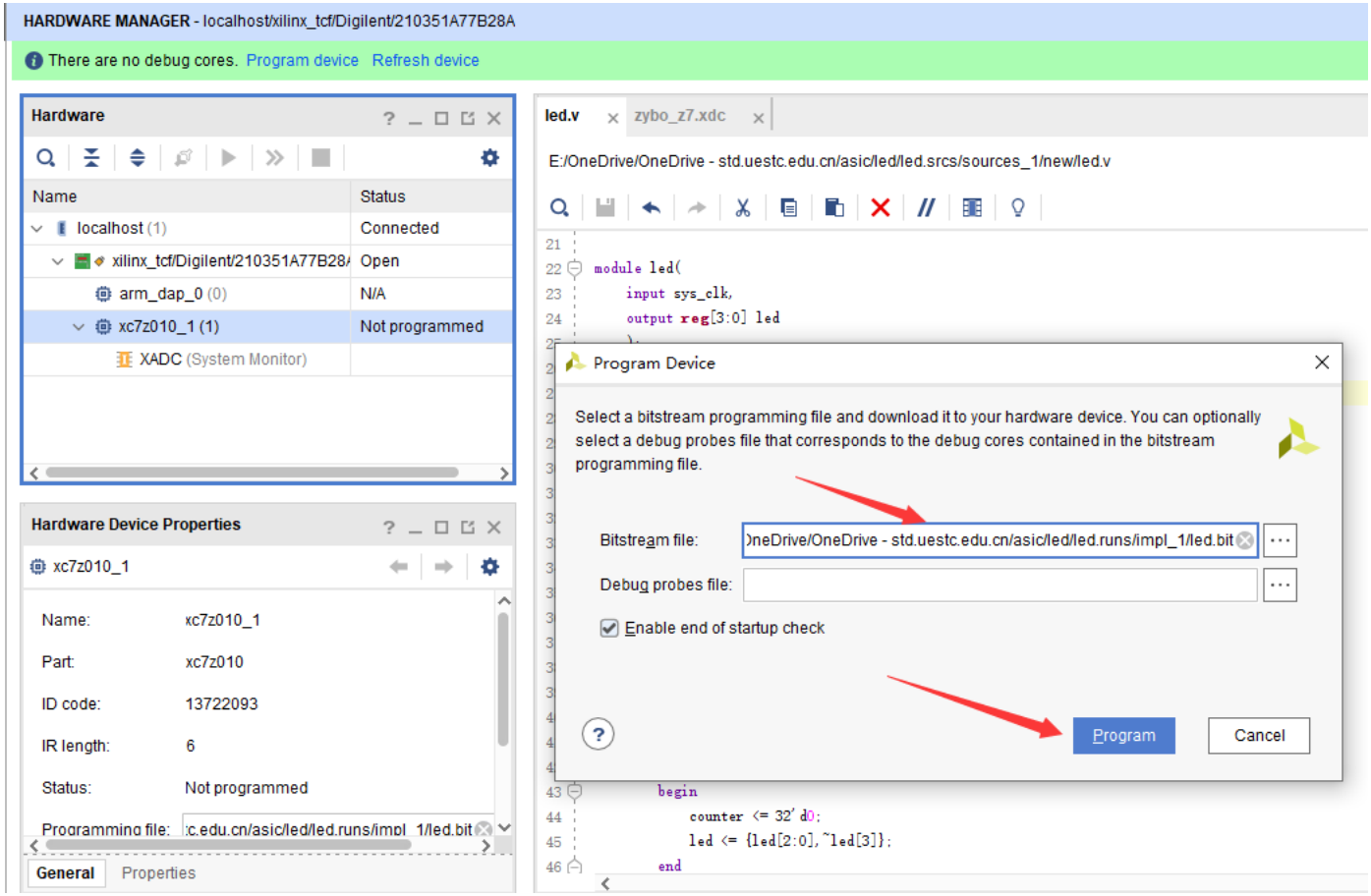
作业内容

编程下载：

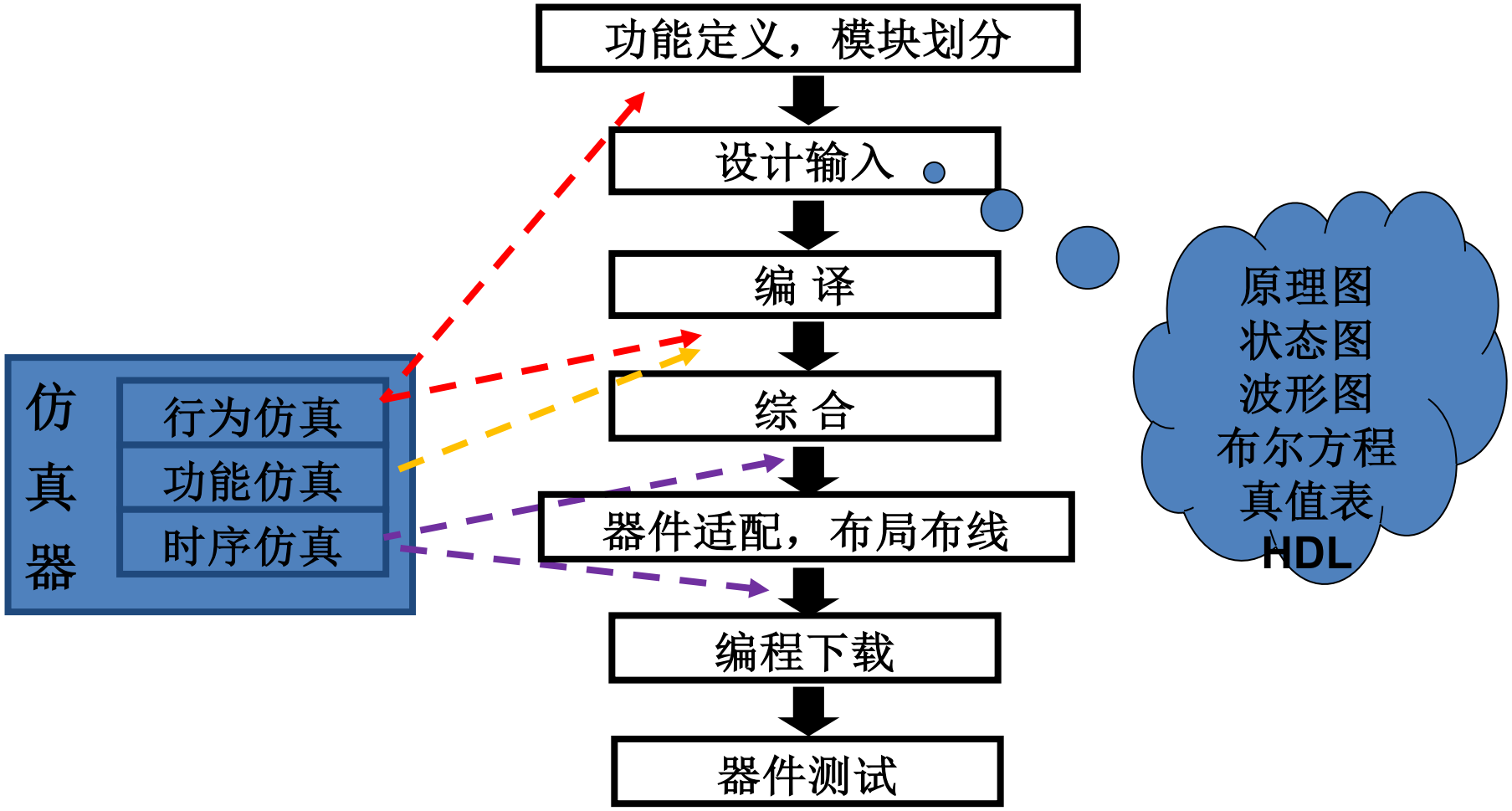

HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210351A77B28A
There are no debug cores. Program device Refresh device
Hardware
Name
Status
localhost (1)
Connected
xilinx_tcf/Digilent/210351A77B28A
Open
arm_dap_0 (0)
N/A
xc7z010_1 (1)
Not programmed
XADC (System Monitor)
led.v
zybo_z7.xdc
E:/OneDrive/OneDrive - std.uest
module led(
input sys_clk,
output reg[3:0] led
);
parameter ONE_SEC
parameter HALF_SEC
reg [32:0] counter;
HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210351A77B28A
There are no debug cores. Program device Refresh device
Hardware
Name
Status
localhost (1)
Connected
xilinx_tcf/Digilent/210351A77B28A
Open
arm_dap_0 (0)
N/A
Hardware Device P
xc7z010_1
Name:
Part:
ID code:
IR length:
Hardware Device Properties... Ctrl+E
Program Device...
Verify Device...
Refresh Device
Show Bus Plot...
Add Configuration Memory Device...
Boot from Configuration Memory Device
Program BBR Key...
Clear BBR Key...
Program eFUSE Registers...
Export to Spreadsheet...

作业内容

编程下载：

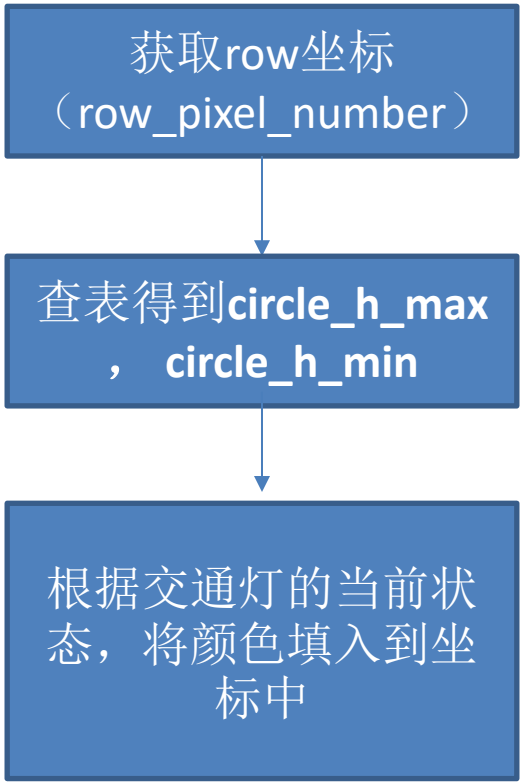

HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210351A77B28A
There are no debug cores. Program device Refresh device
Hardware
Name
Status
localhost (1)
Connected
xilinx_tcf/Digilent/210351A77B28A
Open
arm_dap_0 (0)
N/A
xc7z010_1 (1)
Not programmed
XADC (System Monitor)
Hardware Device Properties
xc7z010_1
Name: xc7z010_1
Part: xc7z010
ID code: 13722093
IR length: 6
Status: Not programmed
Programming file: :c.edu.cn/asic/led/led.runs/impl_1/led.bit
General
Properties
led.v
zybo_z7.xdc
E:/OneDrive/OneDrive - std.uestc.edu.cn/asic/led/led.srcs/sources_1/new/led.v
module led(
input sys_clk,
output reg[3:0] led
begin
counter <= 32'd0;
led <= {led[2:0],~led[3]};
end
Program Device
Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.
Bitstream file:
)neDrive/OneDrive - std.uestc.edu.cn/asic/led/led.runs/impl_1/led.bit
Debug probes file:
Enable end of startup check
Program
Cancel

FPGA设计流程

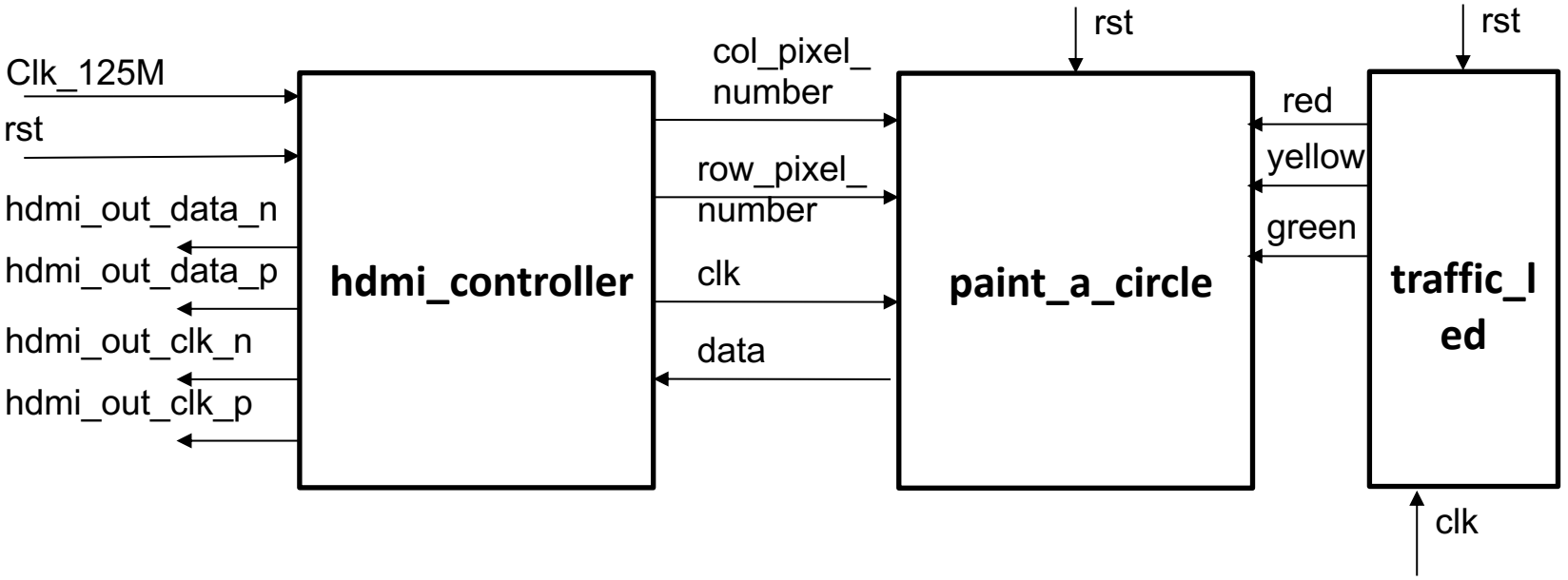

功能定义，模块划分
设计输入
编 译
综 合
器件适配，布局布线
编程下载
器件测试
仿真器
行为仿真
功能仿真
时序仿真
原理图
状态图
波形图
布尔方程
真值表
HDL

模块思路

paint_a_circle



整体架构


Clk_125M
rst
hdmi_out_data_n
hdmi_out_data_p
hdmi_out_clk_n
hdmi_out_clk_p
hdmi_controller
col_pixel_
number
row_pixel_
number
clk
data
rst
paint_a_circle
red
yellow
green
rst
traffic_l
ed
clk

实验任务&评分标准

利用提供IP核及hdmi_controller.v代码完成HDMI/DVI传输协议的通信，达到如下效果：

1、交通灯输出的信号接入HDMI接口，将红绿灯结果显示在屏幕上，时序如下，分值60：

状态	绿	黄	红
延时（s）	10	5	10

2、交通灯输出的信号接入HDMI接口，将红绿灯结果显示在屏幕上，时序如下，分值25：

状态	绿	黄闪	红	黄不闪	绿
延时（s）	10	5	10	5	10

3、加入按键消抖模块，xdc文件加上按键输入，实现如下时序，分值15：

初始状态：绿灯亮10秒，黄闪5秒，红灯亮10秒，黄不闪5秒，绿灯亮10秒

按键按下：一直为红灯

再次按下：绿灯亮10秒，黄闪5秒，红灯亮10秒，黄不闪5秒，绿灯亮10秒

再次按下：一直为红灯

……

第一次大作业

- ✓任务：
  1) 完成实验任务
  2) 5月9日下课之前找助教现场验收
  3) 5月9日下课之前提交实验报告
- ✓报告格式要求：电子档，pdf，文件命名“第N次大作业_学号_姓名”，如：第1次大作业_202252012031_魏浩杰；
- ✓提交方式: 5月9日下课之前提交至助教邮箱：3136302362@qq.com，逾期酌情扣分，请同学们注意按时提交作业。