

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN ĐIỆN TỬ



BÀI TẬP LỚN MÔN VLSI
THIẾT KẾ KHỐI MẠCH ĐẾM SỐ LẦN SỐ 5 XUẤT
HIỆN TỪ TÍN HIỆU ĐẦU VÀO, GIAO TIẾP HIỂN THỊ
TRÊN LED 7 THANH

GVHD: TS. Nguyễn Vũ Thắng
Công ty CoAsia Semi VN

Sinh viên thực hiện: Nhóm 8

Phí Mạnh Toàn	20193142
Vũ Mạnh Hà	20192817
Bùi Việt Đức	20182427

Hà Nội, 08/2023

MỤC LỤC

Nội dung	
REVISION HISTORY	3
DANH MỤC HÌNH VẼ	4
DANH MỤC BẢNG BIỂU	5
I. Design Specification	6
1. Overview	6
2. Functional Requirement	6
3. Block Diagram	6
3.1. Top module block diagram.....	6
3.2. Sub-module block diagram	7
3.3. Flowchart.....	8
4. Functional Specification.....	9
4.1. Khối FSM.....	9
4.1.1. Block Diagram	9
4.1.2. Pins and Parameters Description	9
4.1.3. Implementation Specification	9
4.2. Khối counter.....	10
4.2.1. Block Diagram	10
4.2.2. Pins and Parameters Description	11
4.2.3. Implementation Specification	11
4.3. Khối extract_bit.....	11
4.3.1. Block Diagram	11
4.3.2. Pins and Parameters Description	12
4.3.3. Implementation Specification	12
4.4. Khối LED7SEG	12
4.4.1. Block Diagram	12
4.4.2. Pins and Parameters Description	13
4.4.3. Implementation Specification	13
5. Timing Waveforms	14
II. RTL Coding & Simulation	14
1. RTL Coding & Compilation.....	14

2. Simulation	15
III. Synthesis.....	17
IV. CONCLUSION	21
V. REFERENCES	22

REVISION HISTORY

Revision	Author	Date	Description of changes
0.1	Vũ Mạnh Hà	17/05/23	Phân tích yêu cầu, lưu đồ thuật toán
0.2	Vũ Mạnh Hà, Bùi Việt Đức	24/5/23	Thiết kế chức năng khối giải mã trạng thái FSM
0.3	All	27/5/23	Hoàn thiện Spec
0.4	Phí Mạnh Toàn, Bùi Việt Đức	31/5/23	RTL Coding, Simulation
0.5	Phí Mạnh Toàn	13/8/23	Synthesis
0.6	All	15/8/23	Hoàn thiện project, viết báo cáo, slide

DANH MỤC HÌNH VẼ

Hình 1 Sơ đồ khối Top Module.....	6
Hình 2 Kiến trúc chi tiết khối mạch.....	7
Hình 3 Lưu đồ thuật toán	8
Hình 4 Sơ đồ khối mạch FSM	9
Hình 5 Sơ đồ dịch chuyển trạng thái.....	10
Hình 6 Sơ đồ khối counter	10
Hình 7 Sơ đồ khối extract_bit	11
Hình 8 Sơ đồ khối LED7SEG	12
Hình 9 Nguyên lý led 7 thanh	13
Hình 10 Timing Waveform	14
Hình 11 Giao diện phần mềm Altera Quartus ii.....	14
Hình 12 Kết quả sơ đồ dịch chuyển trạng thái.....	15
Hình 13 Kiến trúc sơ đồ khối mạch.....	15
Hình 14 Testbench khối counter.....	16
Hình 15 Testbench khối FSM.....	16
Hình 16 Testbench khối top_module.....	17
Hình 17 Giao diện Synthesis cùng Yosys.....	17
Hình 18 Chọn top_module làm top module để thực hiện synthesis	18
Hình 19 Thống kê kết quả synthesis.....	18
Hình 20 File định dạng json của kết quả synthesis	19
Hình 21 Mapping với thư viện sky130nm.....	20

DANH MỤC BẢNG BIỂU

Bảng 1 Bảng các chân tín hiệu khối Top Module.....	7
Bảng 2 Bảng các chân tín hiệu khối FSM	9
Bảng 3 Bảng các chân tín hiệu khối counter	11
Bảng 4 Bảng chân tín hiệu khối extract_bit.....	12
Bảng 5 Bảng các chân tín hiệu khối Led 7 thanh.....	13
Bảng 6 Bảng mã hóa led 7 thanh mắc anode chung.....	13

I. Design Specification

1. Overview

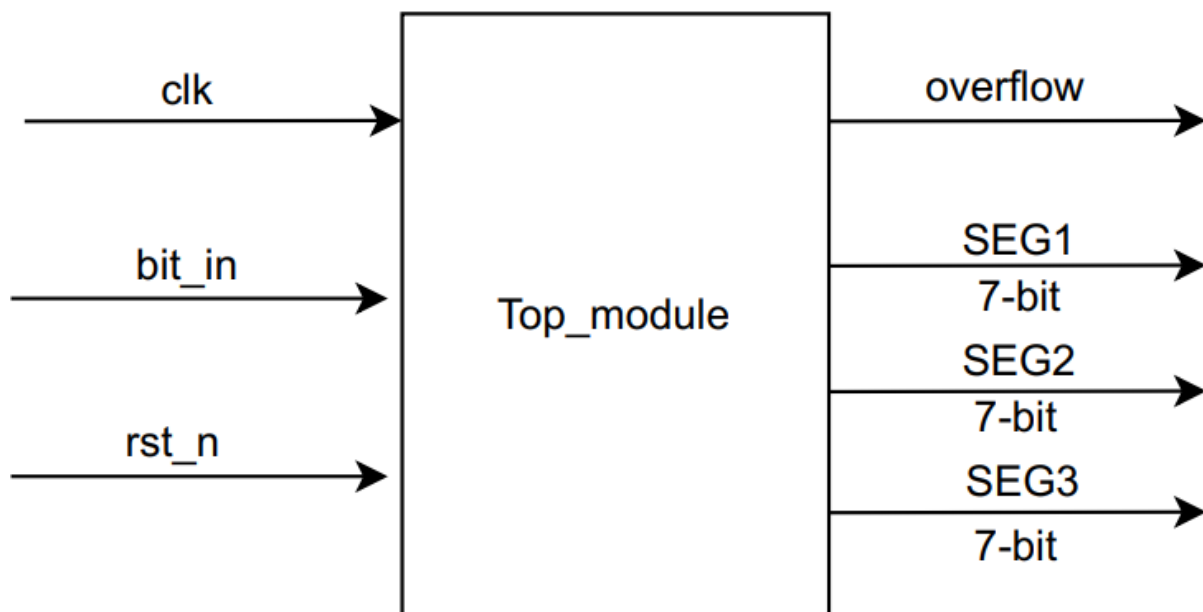
Xây dựng khối mạch điều khiển, đếm số lần số 5 (chuỗi bit “101”) xuất hiện. Đầu ra gồm cờ overflow báo hiệu khi bộ đếm bị tràn, các chân tín hiệu giao tiếp với led 7 thanh hiển thị giá trị bộ đếm.

2. Functional Requirement

- Đếm số lần xuất hiện chuỗi bit 101 từ chuỗi bit đầu vào, tín hiệu đếm count có giá trị 8 bits.
- Cờ overflow lên mức 1 khi bộ đếm bị tràn, giữ trạng thái ở mức 1 khi có tín hiệu reset.
- Đầu ra giao tiếp với Led 7 thanh hiển thị giá trị bộ đếm.
- Tín hiệu reset tích cực mức thấp.

3. Block Diagram

3.1. Top module block diagram

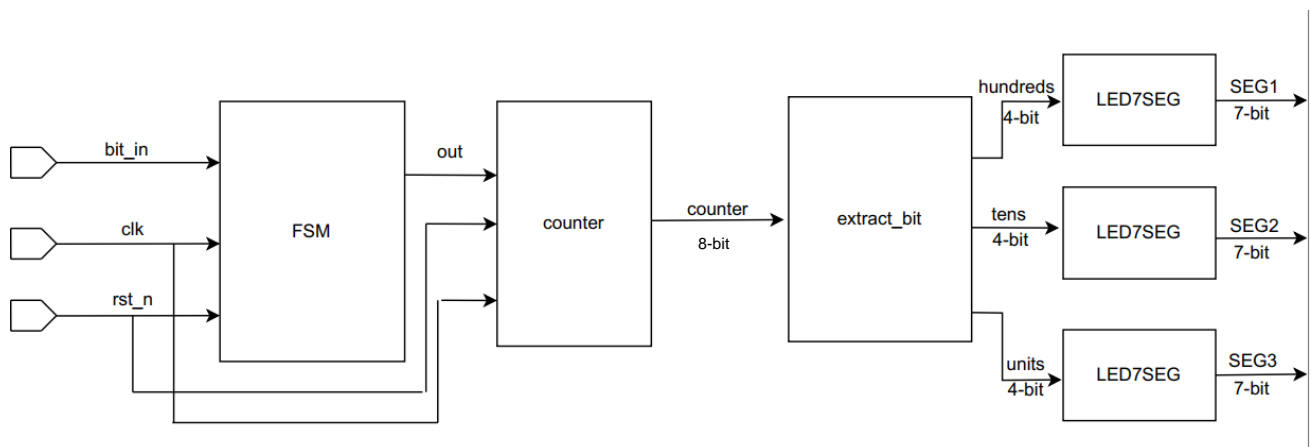


Hình 1 Sơ đồ khối Top Module

Tên tín hiệu	Số bit	Hướng tín hiệu	Mô tả
clk	1	Đầu vào	Xung đồng hồ
rst_n	1	Đầu vào	Tín hiệu reset bộ đếm, tích cực mức thấp
bit_in	1	Đầu vào	Tín hiệu vào, 1 bit dữ liệu.
overflow	1	Đầu ra	Cờ báo tràn, đưa tín hiệu này lên mức 1 khi bộ đếm overflow, giữ mức trạng thái 1 cho đến khi có tín hiệu reset.
SEG1	7	Đầu ra	Điều khiển led 7 thanh hiển thị giá trị hàng trăm của bộ đếm
SEG2	7	Đầu ra	Điều khiển led 7 thanh hiển thị giá trị hàng chục của bộ đếm
SEG3	7	Đầu ra	Điều khiển led 7 thanh hiển thị giá trị hàng đơn vị của bộ đếm

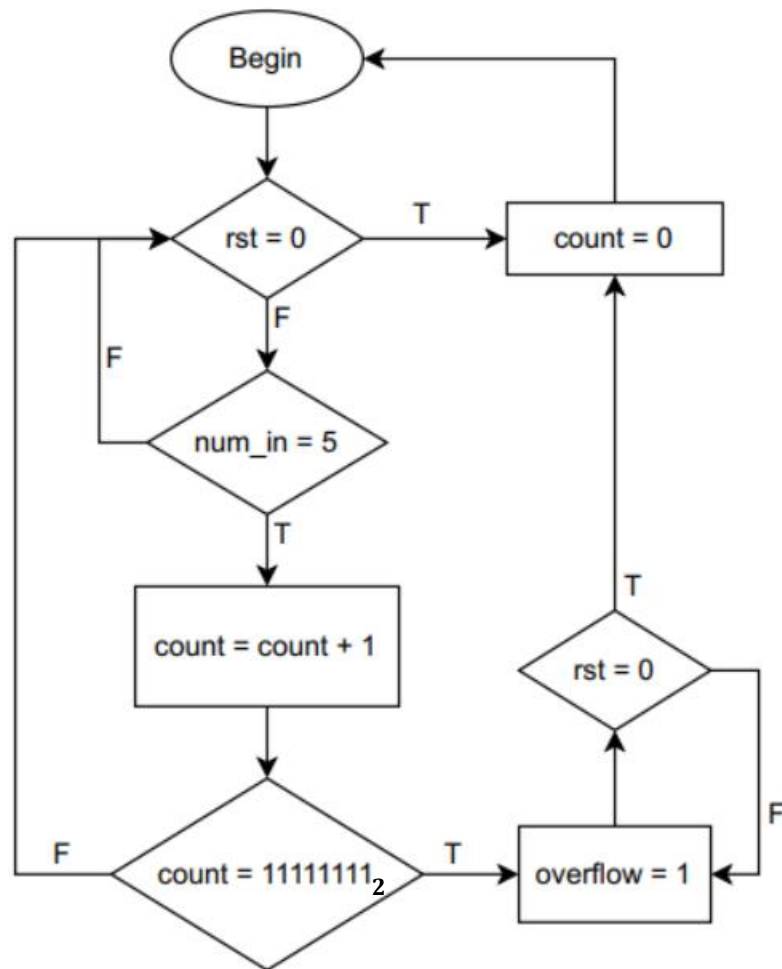
Bảng 1 Bảng các chân tín hiệu khối Top Module

3.2. Sub-module block diagram



Hình 2 Kiến trúc chi tiết khối mạch

3.3. Flowchart

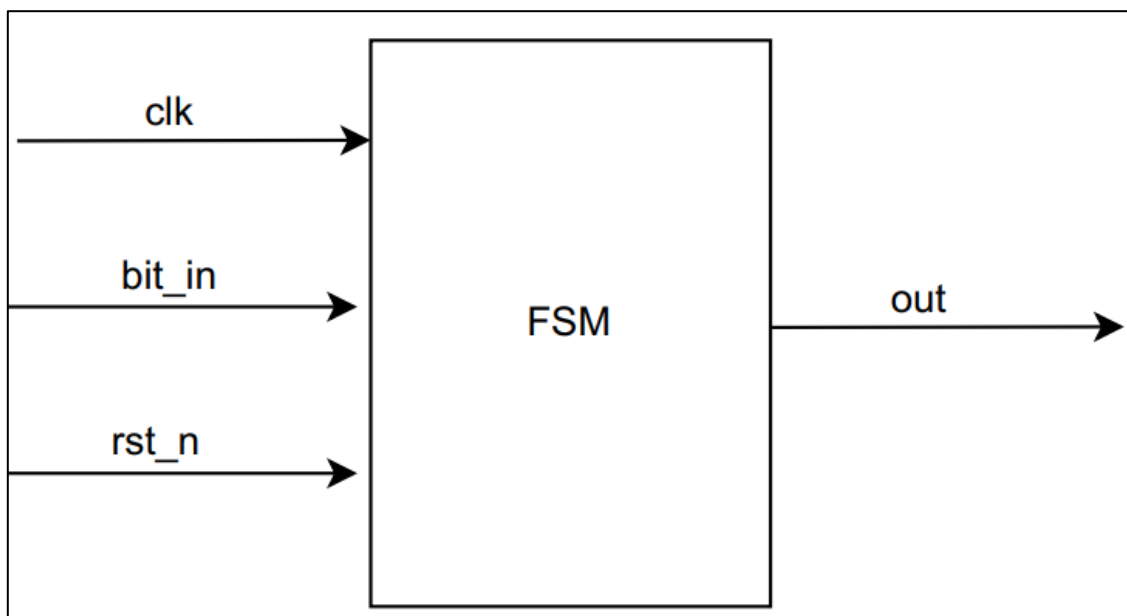


Hình 3 Lưu đồ thuật toán

4. Functional Specification

4.1. Khối FSM

4.1.1. Block Diagram



Hình 4 Sơ đồ khối mạch FSM

4.1.2. Pins and Parameters Description

Tên tín hiệu	Số bit	Hướng tín hiệu	Mô tả
bit_in	1	Đầu vào	Tín hiệu đầu vào
clk	1	Đầu vào	Xung đồng hồ
rst_n	1	Đầu vào	Tín hiệu reset bộ đếm, tích cực mức thấp
out	1	Đầu ra	Tín hiệu đầu ra

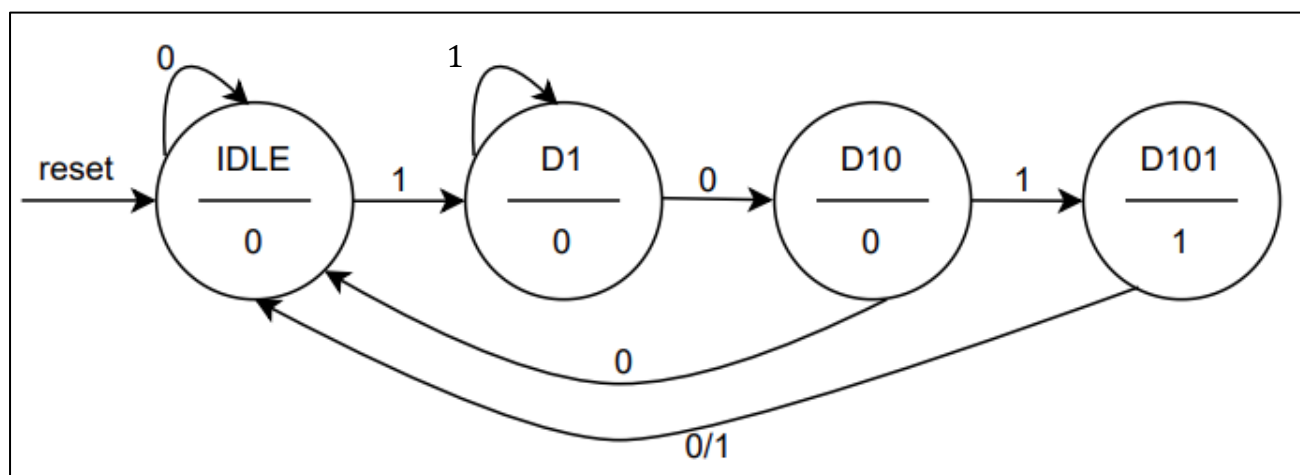
Bảng 2 Bảng các chân tín hiệu khối FSM

4.1.3. Implementation Specification

Khối mạch FSM thực hiện chức năng nhận bit đầu vào, mô hình hóa việc phát hiện chuỗi bit “101” bằng các trạng thái tương ứng. Khi nhận được chuỗi bit “101” tín hiệu đầu ra mức 1, tương ứng với đếm được 1 lần xuất hiện của chuỗi bit yêu cầu.

Sơ đồ dịch chuyển trạng thái được mô tả trong mục 4.1.3.1.

4.1.3.1. State machine diagram



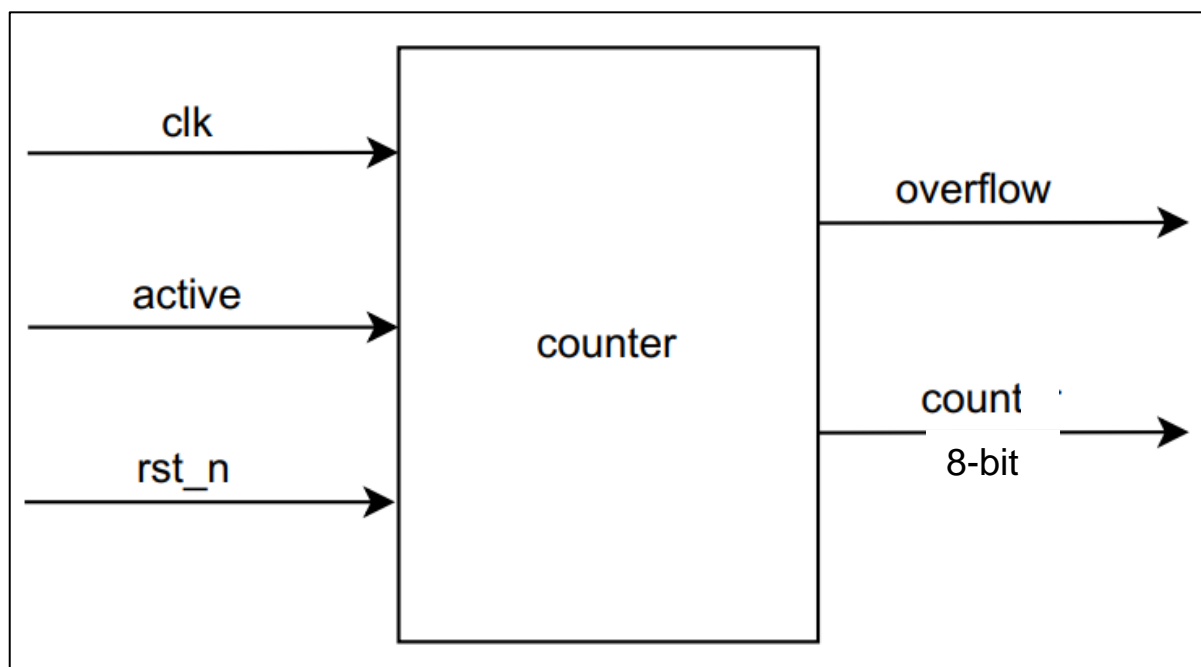
Hình 5 Sơ đồ dịch chuyển trạng thái

Sơ đồ dịch chuyển trạng thái kiểu Moore bao gồm 4 trạng thái tương ứng:

- IDLE: Trạng thái chờ, đầu ra bằng 0.
- D1: Phát hiện được bit “1”, đầu ra bằng 0.
- D10: Phát hiện được chuỗi bit “10”, đầu ra bằng 0.
- D101: Phát hiện được chuỗi bit “101”, đầu ra bằng 1.

4.2. Khối counter

4.2.1. Block Diagram



Hình 6 Sơ đồ khối counter

4.2.2. Pins and Parameters Description

Tên tín hiệu	Số bit	Hướng tín hiệu	Mô tả
clk	1	Đầu vào	Xung đồng hồ
rst_n	1	Đầu vào	Tín hiệu reset bộ đếm, tích cực mức thấp
active	1	Đầu vào	Tín hiệu vào 1 bit dữ liệu, lên mức 1 khi phát hiện được chuỗi bit 101 từ khối FSM
overflow	1	Đầu ra	Cờ báo tràn, đưa tín hiệu này lên mức 1 khi bộ đếm overflow, giữ mức trạng thái 1 cho đến khi có tín hiệu reset.
count	8	Đầu ra	8-bits không dấu thể hiện số lần chuỗi 101 xuất hiện

Bảng 3 Bảng các chân tín hiệu khối counter

4.2.3. Implementation Specification

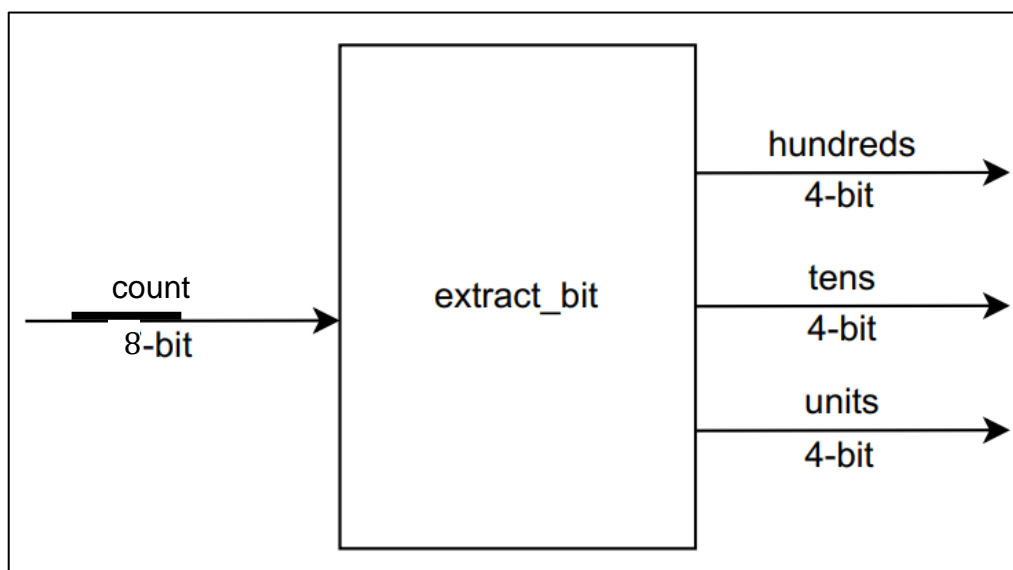
Khối counter nhận tín hiệu active từ khối mạch FSM, khi phát hiện được chuỗi bit từ khối FSM, tín hiệu active lên mức cao, tín hiệu count được cộng thêm 1 đơn vị, cập nhật theo sườn dương của xung clock.

Khi có tín hiệu reset, biến count và cờ overflow được thiết lập lại giá trị 0, thực hiện đếm lại từ đầu.

Khi tín hiệu count tràn (đạt giá trị 1111111_2) cờ overflow lên mức cao, giữ cho đến khi có tín hiệu reset.

4.3. Khối extract_bit

4.3.1. Block Diagram



Hình 7 Sơ đồ khối extract_bit

4.3.2. Pins and Parameters Description

Tên tín hiệu	Số bit	Hướng tín hiệu	Mô tả
count	8	Đầu vào	8 bits không dấu thể hiện số lần chuỗi 101 xuất hiện.
hundreds	4	Đầu ra	4 bits đầu ra điều khiển hoạt động của led 7 thanh hiển thị giá trị đếm hàng trăm.
tens	4	Đầu ra	4 bits đầu ra điều khiển hoạt động của led 7 thanh hiển thị giá trị đếm hàng chục.
units	4	Đầu ra	4 bits đầu ra điều khiển hoạt động của led 7 thanh hiển thị giá trị đếm hàng đơn vị.

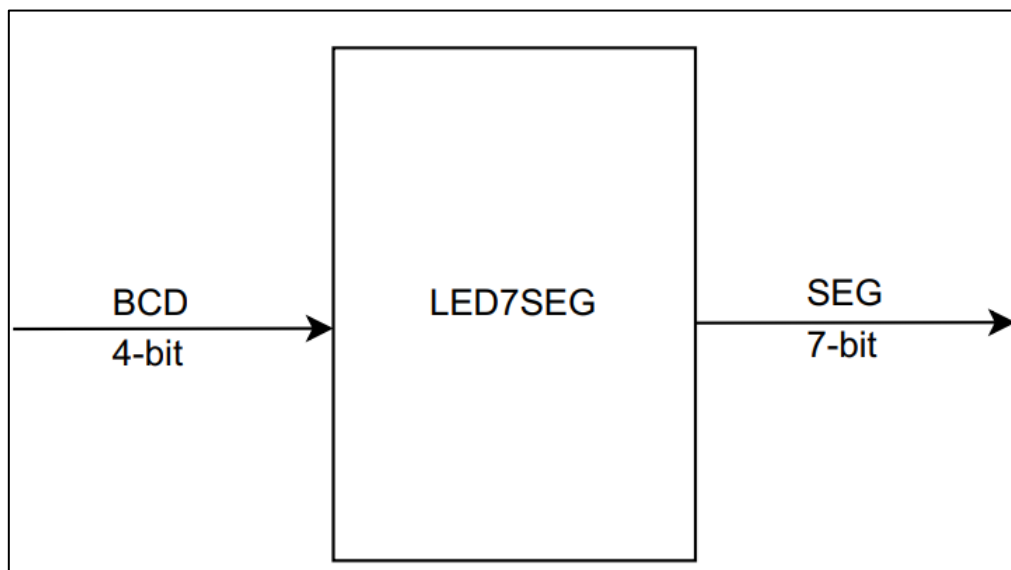
Bảng 4 Bảng chân tín hiệu khối extract_bit

4.3.3. Implementation Specification

Khối extract_bit có chức năng chuyển biến đếm count (8 bits) thành các chữ số (4 bits) hàng trăm, hàng chục, hàng đơn vị riêng biệt để hiển thị lên Led 7 thanh.

4.4. Khối LED7SEG

4.4.1. Block Diagram



Hình 8 Sơ đồ khối LED7SEG

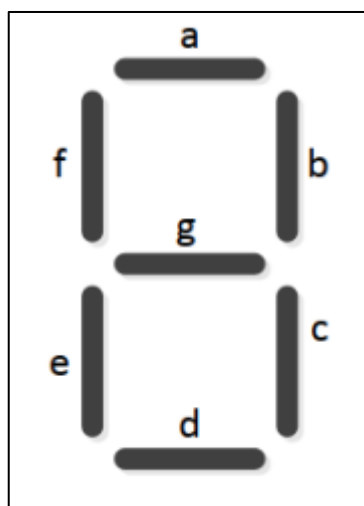
4.4.2. Pins and Parameters Description

Tên tín hiệu	Số bit	Hướng tín hiệu	Mô tả
BCD	4	Đầu vào	Số thập phân 4 bits đầu vào (0-9)
SEG	7	Đầu ra	7 bits đầu ra tương ứng điều khiển các chân tín hiệu của led 7 thanh (mắc anode chung).

Bảng 5 Bảng các chân tín hiệu khối Led 7 thanh

4.4.3. Implementation Specification

Khối LED7SEG thực hiện giao tiếp với Led 7 thanh, chuyển đổi từ số thập phân đầu vào thành tín hiệu 7 bits tương ứng điều khiển led 7 thanh hiển thị các số tương ứng.

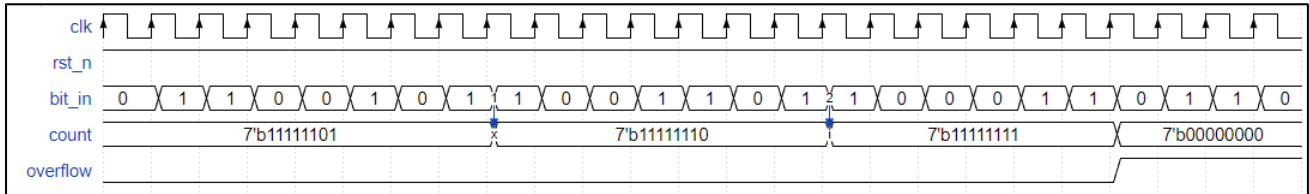


Hình 9 Nguyên lý led 7 thanh

Đầu vào	Đầu ra
4'd0	7'b00000001
4'd1	7'b1001111
4'd2	7'b0010010
4'd3	7'b1001100
4'd4	7'b0100100
4'd5	7'b1100001
4'd6	7'b1100000
4'd7	7'b0001111
4'd8	7'b0000000
4'd9	7'b0001100
default	7'b00000001

Bảng 6 Bảng mã hóa led 7 thanh mắc anode chung.

5. Timing Waveforms

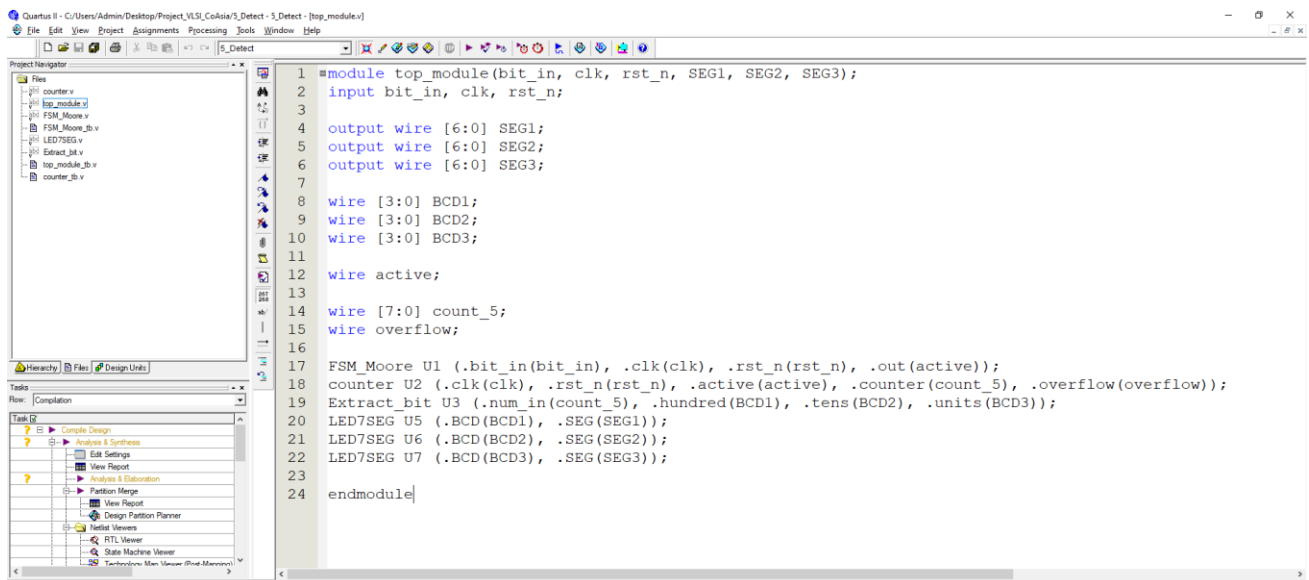


Hình 10 Timing Waveform

II. RTL Coding & Simulation

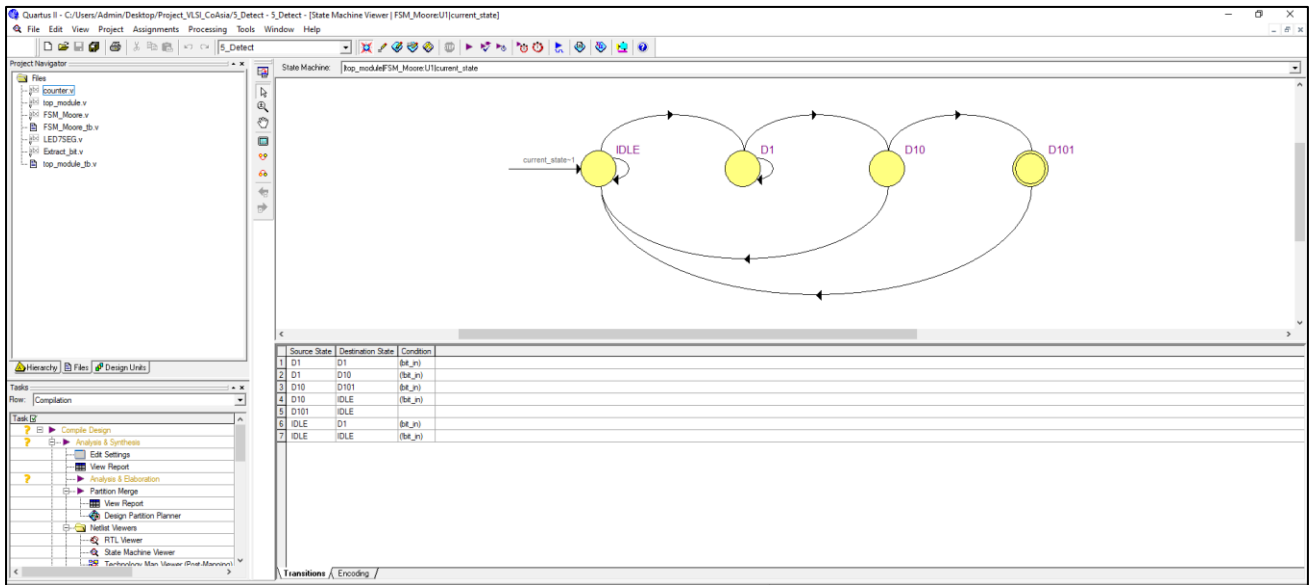
1. RTL Coding & Compilation

Tool: Altera Quartus II 9.1sp2 Web Edition

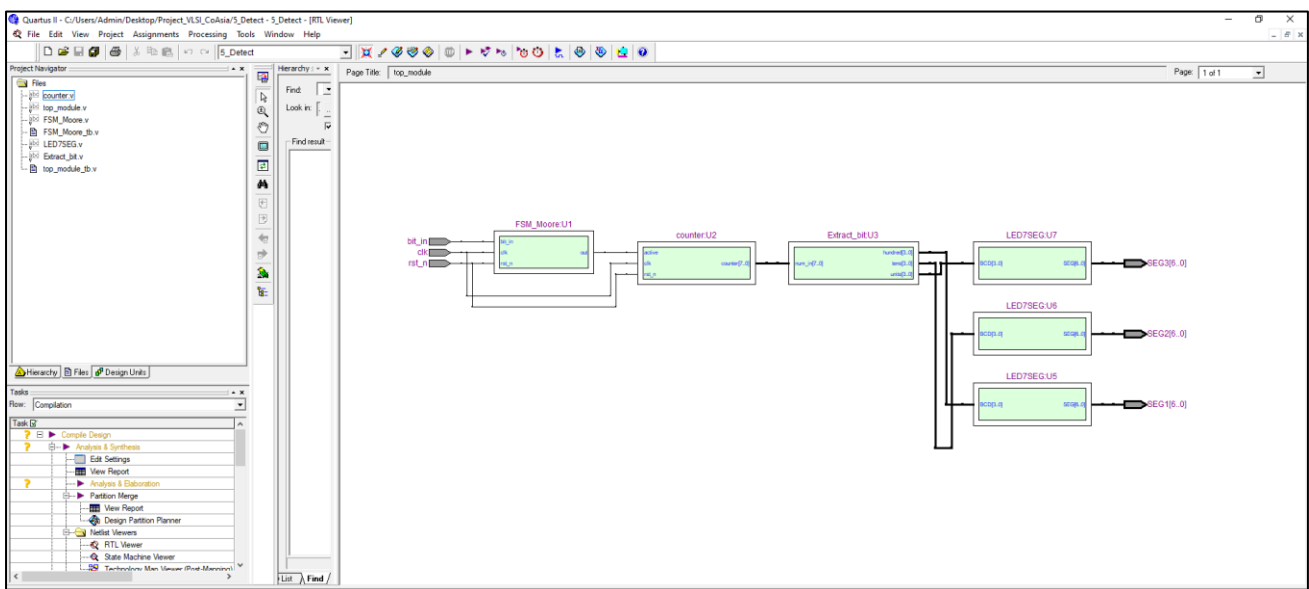


Hình 11 Giao diện phần mềm Altera Quartus ii

Kết quả:



Hình 12 Kết quả sơ đồ dịch chuyển trạng thái

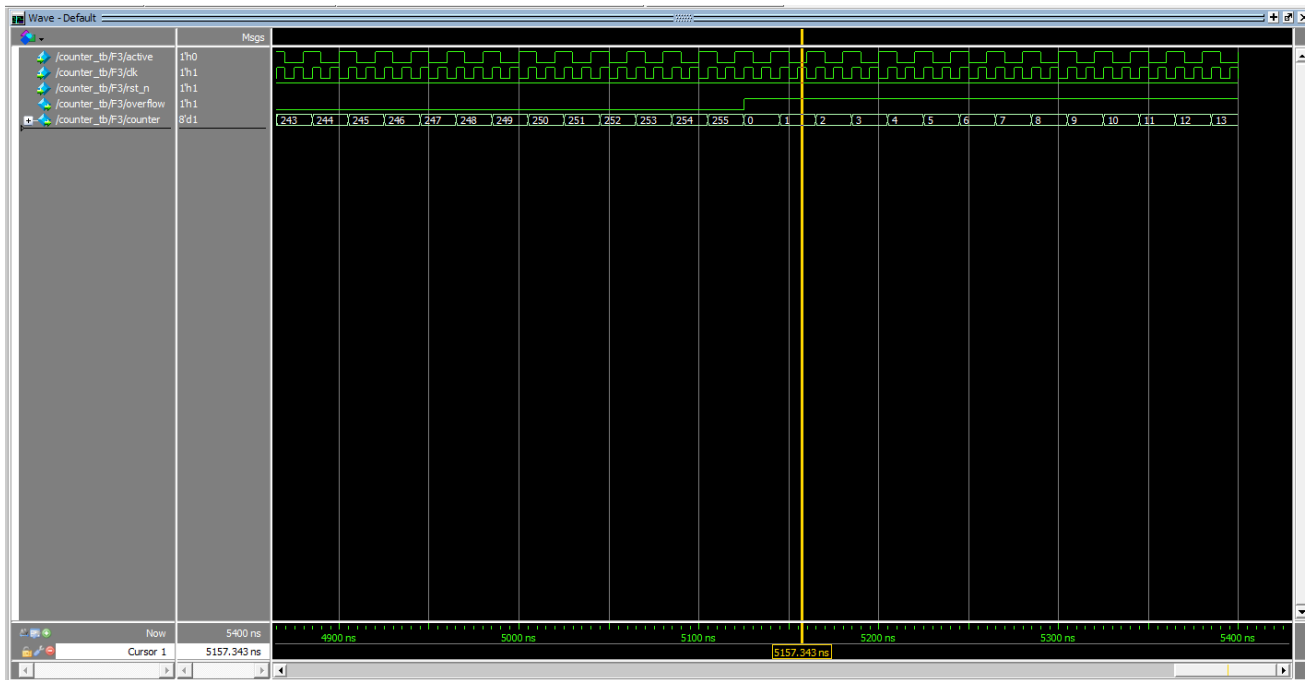


Hình 13 Kiến trúc sơ đồ khối mạch

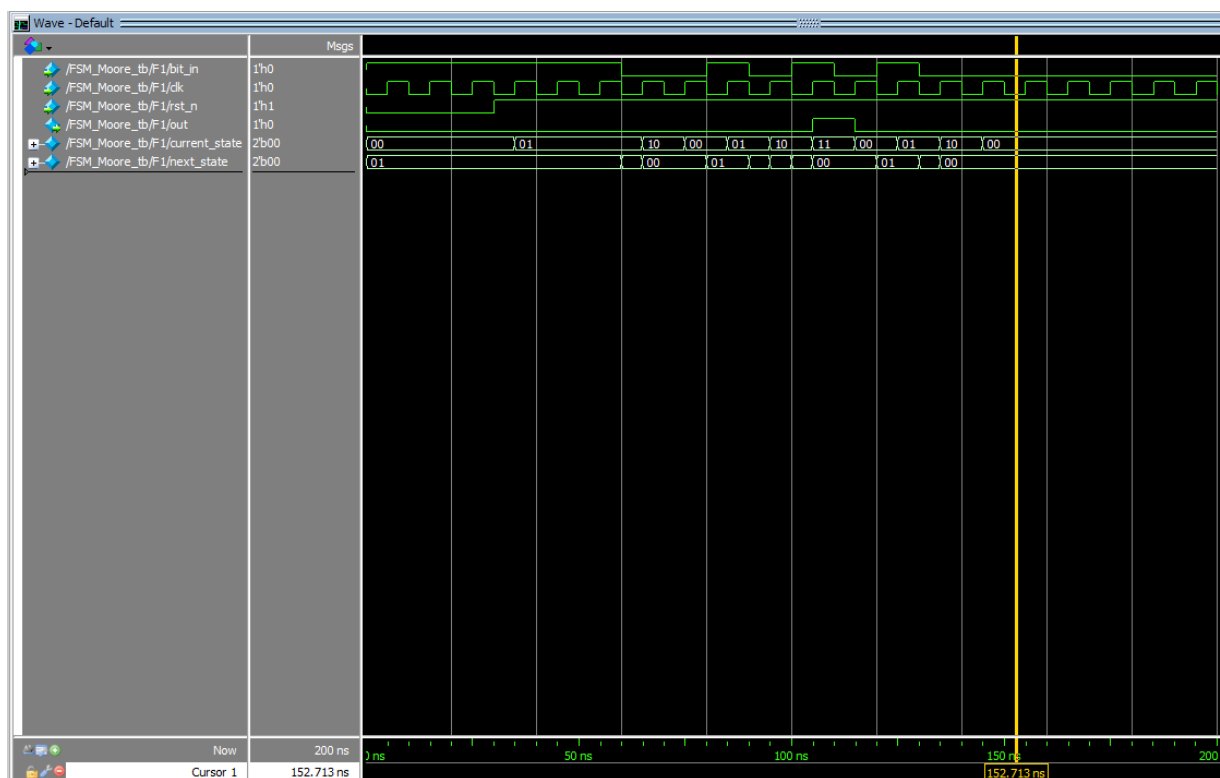
2. Simulation

Tool: Modelsim SE-64 2020.4

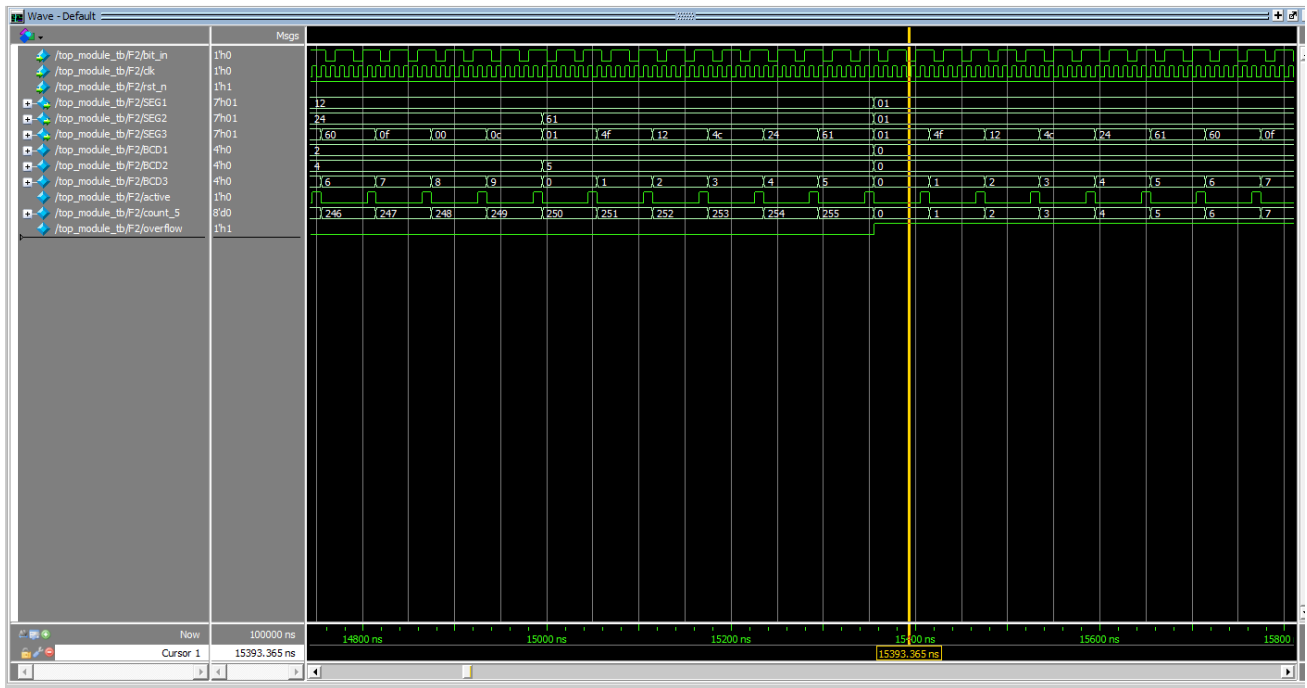
Thực hiện viết các file testbench, mô phỏng trên phần mềm modelsim, kiểm tra chức năng lần lượt của các khối counter, FSM, top_module.



Hình 14 Testbench khối counter



Hình 15 Testbench khối FSM



Hình 16 Testbench khối top_module

Nhận xét: Chức năng của các khối module chính thực hiện chính xác, đạt các yêu cầu về thiết kế như đã đề ra.

III. Synthesis

Thực hiện quá trình Synthesis (Tổng hợp) trên hệ điều hành Linux, công cụ tổng hợp: Yosys.

Mapping với thư viện công nghệ SKY130nm (Google x Skywater130nm PDK)

```

toanpm1011@toanpm1011: ~/$ cd VLSI/Project_VLSI_20222/Synthesis
toanpm1011@toanpm1011:~/VLSI/Project_VLSI_20222/Synthesis$ yosys

-----
yosys -- Yosys Open SYnthesis Suite

Copyright (C) 2012 - 2019 Clifford Wolf <clifford@clifford.at>

Permission to use, copy, modify, and/or distribute this software for any
purpose with or without fee is hereby granted, provided that the above
copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
-----

Yosys 0.9 (git sha1 1979e0b)

yosys>

```

Hình 17 Giao diện Synthesis cùng Yosys

```
toanpm1011@toanpm1011: ~/VLSI/Project_VLSI_2022/Synt...
Warning: wire '\counter' is assigned in a block at top_module.v:166.
Warning: wire '\counter' is assigned in a block at top_module.v:171.
Successfully finished Verilog frontend.

yosys> hierarchy -top top_module

2. Executing HIERARCHY pass (managing design hierarchy).

2.1. Analyzing design hierarchy..
Top module: \top_module
Used module: \LED7SEG
Used module: \Extract_bit
Used module: \counter
Used module: \FSM_Moore

2.2. Analyzing design hierarchy..
Top module: \top_module
Used module: \LED7SEG
Used module: \Extract_bit
Used module: \counter
Used module: \FSM_Moore
Removed 0 unused modules.

yosys>
```

Hình 18 Chọn top_module làm top module để thực hiện synthesis

```
=== design hierarchy ===

top_module          1
  Extract_bit        1
    FSM_Moore        1
    LED7SEG          3
    counter          1

Number of wires:      5287
Number of wire bits:  5390
Number of public wires:  40
Number of public wire bits: 127
Number of memories:    0
Number of memory bits:  0
Number of processes:    0
Number of cells:      5284
  $_ANDNOT_           918
  $_AND_               103
  $_AOI3_              344
  $_AOI4_               1
  $_DFF_PN0_           22
  $_DFF_PN1_            1
  $_MUX_               911
  $_NAND_              228
  $_NOR_               411
  $_NOT_               365
  $_OAI3_              420
  $_ORNOT_             218
  $_OR_                477
  $_XNOR_              172
  $_XOR_               693

3.27. Executing CHECK pass (checking for obvious problems).
checking module Extract_bit..
checking module FSM_Moore..
checking module LED7SEG..
checking module counter..
checking module top_module..
found and reported 0 problems.

yosys>
```

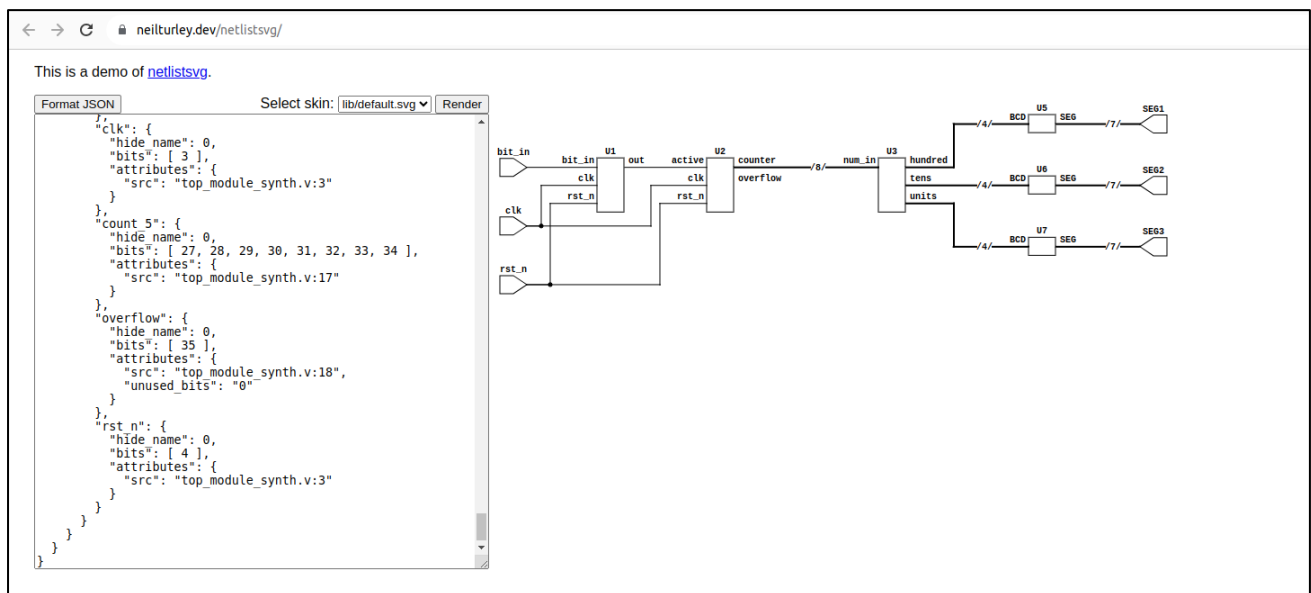
Hình 19 Thống kê kết quả synthesis

```

testing.v x testing_post_synth.json x counter.v x Extract_bit.v x FSM_Moore.v x LED7SEG.v x top_module.v x top_module_synth.v x top_module_synth.post_synth.json x top_module_synth.post_synth.v x
1 {
2   "creator": "Yosys 0.9 (git sha1 1979e6b)",
3   "modules": {
4     "Extract_bit": {
5       "attributes": {
6         "src": "top_module_synth.v:122"
7       },
8       "ports": {
9         "num_in": {
10          "direction": "input",
11          "bits": [ 2, 3, 4, 5, 6, 7, 8, 9 ]
12        },
13        "hundred": {
14          "direction": "output",
15          "bits": [ 10, 11, "0", "0" ]
16        },
17        "tens": {
18          "direction": "output",
19          "bits": [ 12, 13, 14, 15 ]
20        },
21        "units": {
22          "direction": "output",
23          "bits": [ 16, 17, 18, 19 ]
24        }
25      },
26      "cells": {
27        "$abc$147398$auto$blifparse.cc:371:parse_blif$147399": {
28          "hide_name": 1,
29          "type": "$S_AND_",
30          "parameters": {
31            "A": [ 9 ],
32            "B": [ 8 ],
33            "Y": [ 20 ]
34          },
35          "port_directions": {
36            "A": "input",
37            "B": "input",
38            "Y": "output"
39          },
40          "connections": {
41            "A": [ 9 ],
42            "B": [ 8 ],
43            "Y": [ 20 ]
44          }
45        },
46        "$abc$147398$auto$blifparse.cc:371:parse_blif$147400": {
47          "hide_name": 1,
48          "type": "$S_OR_",
49          "parameters": {
50            "A": [ 9 ],
51            "B": [ 8 ],
52            "Y": [ 20 ]
53          },
54          "port_directions": {
55            "A": "input",
56            "B": "input",
57            "Y": "output"
58          },
59          "connections": {
60            "A": [ 9 ],
61            "B": [ 8 ],
62            "Y": [ 20 ]
63          }
64        }
65      }
66    }
67  }
68 }

```

Hình 20 File định dạng json của kết quả synthesis



Hình 21. Kết quả sau khi tổng hợp

```
toanpm1011@toanpm1011: ~/Project_VLSI_20222/Synthesis
sky130_fd_sc_hd__dfbbs_1 _DFFSR_NNP_ (.CLK_N( C ), .D( D ), .Q( Q ), .Q_N(~Q),
.RESET_B(~R), .SET_B( S));
sky130_fd_sc_hd__dfbbs_1 _DFFSR_NPN_ (.CLK_N( C ), .D( D ), .Q( Q ), .Q_N(~Q),
.RESET_B( R ), .SET_B(~S));
sky130_fd_sc_hd__dfbbs_1 _DFFSR_NPP_ (.CLK_N( C ), .D( D ), .Q( Q ), .Q_N(~Q),
.RESET_B(~R), .SET_B(~S));
sky130_fd_sc_hd__dfbbs_1 _DFFSR_PNN_ (.CLK( C ), .D( D ), .Q( Q ), .Q_N(~Q), .R
ESET_B( R ), .SET_B( S));
sky130_fd_sc_hd__dfbbs_1 _DFFSR_PNP_ (.CLK( C ), .D( D ), .Q( Q ), .Q_N(~Q), .R
ESET_B(~R), .SET_B( S));
sky130_fd_sc_hd__dfbbs_1 _DFFSR_PPN_ (.CLK( C ), .D( D ), .Q( Q ), .Q_N(~Q), .R
ESET_B( R ), .SET_B(~S));
sky130_fd_sc_hd__dfbbs_1 _DFFSR_PPP_ (.CLK( C ), .D( D ), .Q( Q ), .Q_N(~Q), .R
ESET_B(~R), .SET_B(~S));
Mapping DFF cells in module '\Extract_bit':
Mapping DFF cells in module '\FSM_Moore':
  mapped 3 $_DFF_PN0_ cells to \sky130_fd_sc_hd__dfbbs_1 cells.
  mapped 1 $_DFF_PN1_ cells to \sky130_fd_sc_hd__dfbbs_2 cells.
Mapping DFF cells in module '\LED7SEG':
Mapping DFF cells in module '\counter':
  mapped 9 $_DFF_PN0_ cells to \sky130_fd_sc_hd__dfbbs_1 cells.
Mapping DFF cells in module '\top_module':

yosys>
```

Hình 21 Mapping với thư viện sky130nm

Các file mã nguồn, báo cáo, slide tổng hợp tại:

https://github.com/toanpm1011/Project_VLSI_20222.git

IV. CONCLUSION

Sau quá trình học tập trên lớp và quá trình thực hiện đề tài cùng công ty CoAsia Semi VN, nhóm chúng em đã hoàn thành các yêu cầu đề ra của đề tài này. Không chỉ trong quy mô một project môn học, nhóm chúng em đã tiếp thu được nhiều kiến thức về ngành vi mạch, về quy trình làm việc chuyên nghiệp của công ty.

Trong quá trình làm việc nhóm, các thành viên đều có ý thức hoàn thành công việc được phân chia, tham gia đầy đủ các buổi hướng dẫn từ phía công ty của anh Phạm Minh Đức và anh Nguyễn Ngọc Trung.

Lời đầu tiên chúng em xin cảm ơn thầy, TS. Nguyễn Vũ Thắng, người trực tiếp giảng dạy và kết nối chúng em với công ty. Đây là một cơ hội rất hữu ích để chúng em có thể tiếp cận, được học tập và làm việc với doanh nghiệp. Từ đó, chúng em thu được rất nhiều kinh nghiệm hữu ích cho quá trình làm việc thực tế sau này.

Chúng em xin gửi lời cảm ơn tới công ty CoAsia Semi VN đã tạo điều kiện hỗ trợ chúng em, đặc biệt là anh Phạm Minh Đức và anh Nguyễn Ngọc Trung, đã thay mặt công ty hướng dẫn chúng em trong quá trình thực hiện đề tài này.

V. REFERENCES

- [1] ASIC Desin: Design Synthesis- CoAsia SEMI VN, DucPM.
- [2] User Guide: Installing/ Deploying Open-Source EDA Tools, For VLSI Course, 2022.2, Dr. Nguyen Vu Thang, HUST- DucPM, CoAsia SEMI VN.
- [3] Design Template- CoAsia SEMI VN.
- [3] FPGA Propotyping by Verilog Examples- Pong P.Chu, Xilinx Spartan™-3 Version.