

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG ĐIỆN – ĐIỆN TỬ**



**THIẾT KẾ VLSI**

**Đề tài:**

**THIẾT KẾ BỘ GIẢI MÃ SỬ DỤNG THUẬT TOÁN VITERBI**

Sinh viên thực hiện:      Ngô Minh Đắc              20213870

   Vũ Tuấn Minh              20214015

   Nguyễn Việt Thành      20211016

Nhóm:                              20

Ngành:                              Điện tử Viễn thông – K66

Giảng viên hướng dẫn:      TS. Nguyễn Vũ Thắng

Hà Nội, 1-2025

# MỤC LỤC

MỤC LỤC.....	2
DANH MỤC HÌNH ẢNH .....	iv
DANH MỤC BẢNG BIỂU .....	v
1. Cơ sở lý thuyết .....	1
1.1 Điều chế .....	1
1.2 Giải điều chế .....	1
2. Khối top module.....	4
2.1. Sơ đồ khối .....	4
2.2. Chức năng .....	4
2.3. Đầu vào đầu ra .....	4
3. Module khối control.....	5
3.1. Sơ đồ khối .....	5
3.2. Chức năng .....	5
3.3. Đầu vào đầu ra .....	5
3.4. Sơ đồ máy trạng thái FSM .....	6
4. Module khối extract_bit .....	7
4.1. Sơ đồ khối .....	7
4.2. Chức năng .....	7
4.3. Đầu vào đầu ra .....	7
4.4. Lưu đồ thuật toán .....	8
5. Module khối branch_metric .....	9
5.1. Sơ đồ khối branch_metric .....	9
5.2. Chức năng .....	9
5.3. Đầu vào, đầu ra .....	9
5.4. Lưu đồ thuật toán .....	10
6. Module khối add_compare_select .....	11
6.1. Sơ đồ khối .....	11
6.2. Chức năng .....	11

6.3. Đầu vào, đầu ra .....	12
6.4. Lưu đồ thuật toán .....	13
7. Module memory .....	14
7.1. Sơ đồ khối .....	14
7.2. Chức năng .....	14
7.3. Đầu vào, đầu ra .....	14
8. Module traceback_output_decision .....	15
8.1. Sơ đồ khối .....	15
8.2. Chức năng .....	15
8.3 Đầu vào, đầu ra .....	16
8.4. Máy trạng thái FSM .....	17
9. Kết quả đạt được .....	17

## DANH MỤC HÌNH ẢNH

Hình 1. Sơ đồ khối hệ thống Điều chế .....	1
Hình 2. Sơ đồ FSM.....	2
Hình 3. Sơ đồ FSM dạng lưới.....	2
Hình 4. Lưu đồ Hamming Distance .....	3
Hình 5. Sơ đồ khối top .....	4
Hình 6. Sơ đồ khối control.....	5
Hình 7. Sơ đồ máy trạng thái cho khối control.....	6
Hình 8. Sơ đồ khối extract_bit .....	7
Hình 9. Lưu đồ thuật toán khối extract_bit .....	8
Hình 10. Sơ đồ khối branch_metric.....	9
Hình 11. Lưu đồ thuật toán khối branch_metric .....	10
Hình 12. Sơ đồ khối add_compare_select .....	11
Hình 13. Lưu đồ thuật toán khối add_compare_select.....	13
Hình 14. Sơ đồ khối memory.....	14
Hình 15. Sơ đồ khối traceback_output_decision .....	15
Hình 16. Máy trạng thái FSM khối traceback_output_decision .....	17
Hình 17. Kết quả Direct Test.....	18
Hình 18. Kết quả Synthesis .....	19
Hình 19. Kết quả Layout.....	20
Hình 20. Kết quả DRC Check.....	21
Hình 21. Kết quả Connectivity Check.....	21

## DANH MỤC BẢNG BIỂU

<i>Bảng 1. Bảng trạng thái của hệ thống.....</i>	<i>1</i>
<i>Bảng 2. Bảng Hamming Distance .....</i>	<i>3</i>
<i>Bảng 3. Đầu vào, đầu ra khối top .....</i>	<i>4</i>
<i>Bảng 4. Đầu vào, đầu ra khối control.....</i>	<i>5</i>
<i>Bảng 5. Đầu vào, đầu ra khối extract_bit .....</i>	<i>7</i>
<i>Bảng 6. Khoảng cách Hamming giữa các nút.....</i>	<i>9</i>
<i>Bảng 7. Đầu vào, đầu ra khối branch_metric .....</i>	<i>9</i>
<i>Bảng 8. Đầu vào, đầu ra khối add_compare_select .....</i>	<i>12</i>
<i>Bảng 9. Đầu vào, đầu ra khối memory.....</i>	<i>14</i>
<i>Bảng 10. Đầu vào, đầu ra khối traceback_output_decision .....</i>	<i>16</i>

# 1. Cơ sở lý thuyết

## 1.1 Điều chế

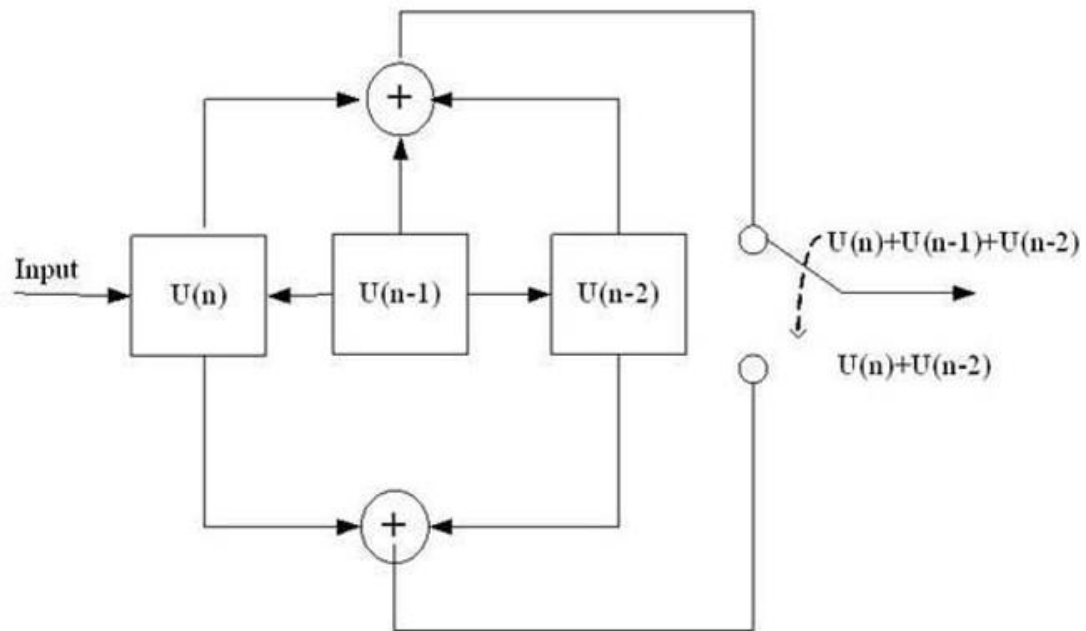


Figure 1: Shift Register in Convolutional code

Hình 1. Sơ đồ khối hệ thống Điều chế

## 1.2 Giải điều chế

$$C1 = 1 + x + x^2$$

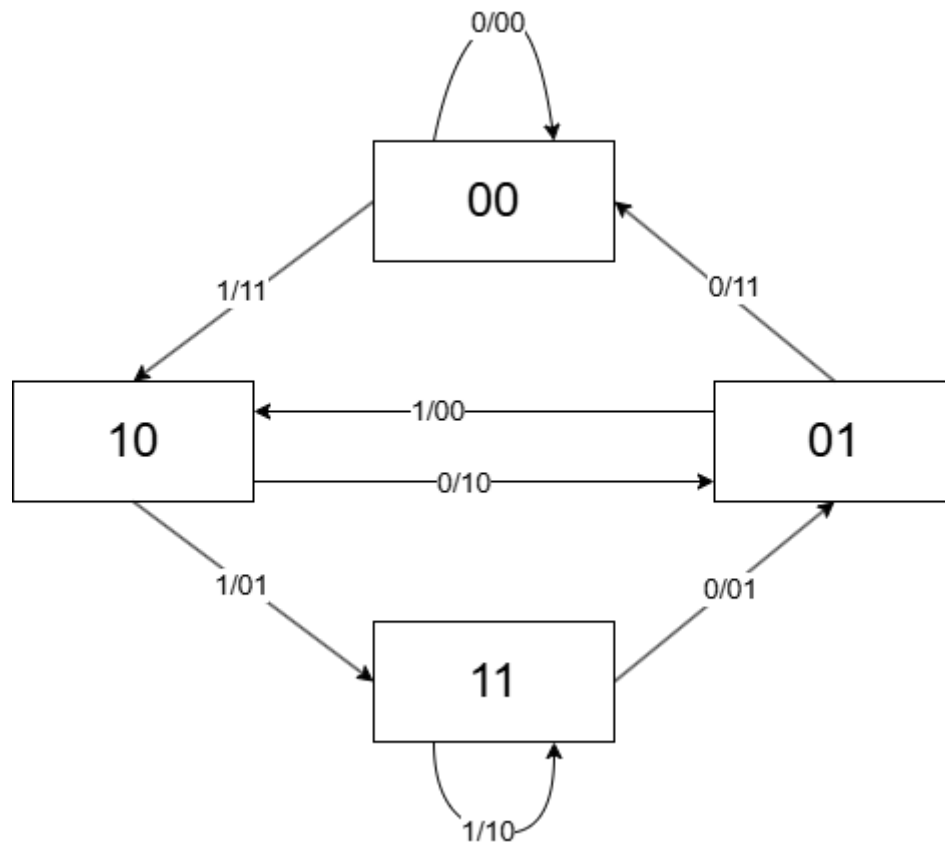
$$C2 = 1 + x^2$$

Trạng thái của hệ thống là gộp của 2 thanh ghi  $U(n-1)$  và  $U(n-2)$  tương ứng S1S2

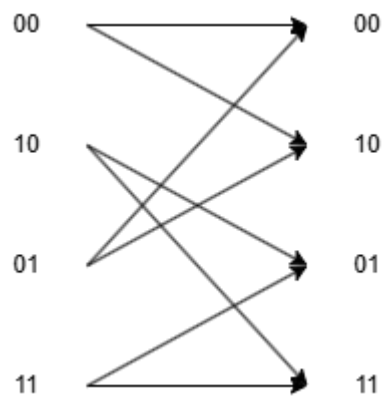
Input	S1	S2	C1	C2	S1'	S2'
0	0	0	0	0	0	0
1	0	0	1	1	1	0
0	0	1	1	1	0	0
1	0	1	0	0	1	0
0	1	0	1	0	0	1
1	1	0	0	1	1	1
0	1	1	0	1	0	1
1	1	1	1	0	1	1

Bảng 1: Bảng trạng thái của hệ thống

- State Diagram:



Hình 2. Sơ đồ FSM



Hình 3. Sơ đồ FSM dạng lưới

- Qua sơ đồ dịch chuyển trạng thái, ta có thể đưa ra nhận xét sau:

+ Để dịch chuyển tới trạng thái 00, 01 bit đầu vào phải có giá trị là 0 (khác trạng thái trước)

+ Để dịch chuyển tới trạng thái 10, 11 bit đầu vào phải có giá trị là 1 (khác trạng thái trước)

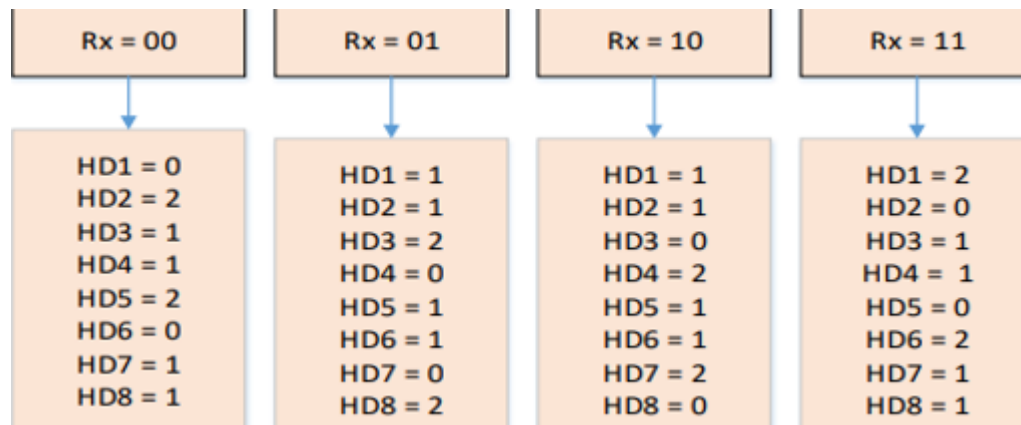
=> Không thể phân biệt dịch chuyển trạng thái thông qua bit đầu vào mà phải lưu lại từng trạng thái dẫn dịch chuyển đến trạng thái đang xét

\* Ký hiệu Hamming Distance cho các dịch chuyển:

Trước\Sau	00	10	01	11
00	HD1 (00)	HD2 (11)		
10			HD3 (10)	HD4 (01)
01	HD5 (11)	HD6 (00)		
11			HD7 (01)	HD8 (10)

*Bảng 2: Bảng Hamming Distance*

- Từ đó có thể dễ dàng tính sẵn độ sai khác bit (Hamming Distance) của các dịch chuyển với các trường hợp 2 bit đầu tương ứng 00, 01, 10, 11:

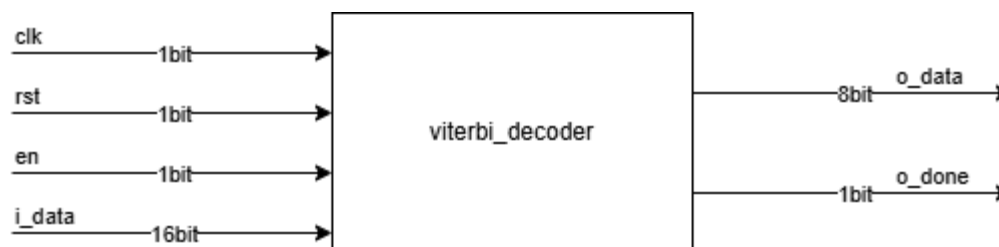


*Hình 4. Lưu đồ Hamming Distance*



## 2. Khối top module

### 2.1. Sơ đồ khối



Hình 5. Sơ đồ khối top

### 2.2. Chức năng

Module Viterbi\_decoder đóng vai trò là top module trong khối thiết kế, có chức năng khởi tạo các khối module bé hơn và thực hiện kết nối tín hiệu giữa các khối này với nhau. Việc sử dụng khối top module giúp tách biệt các phần khác nhau của hệ thống, làm cho việc thiết kế và kiểm tra trở nên dễ dàng hơn, và cho phép các phần tử khác nhau của hệ thống được phát triển và kiểm tra độc lập.

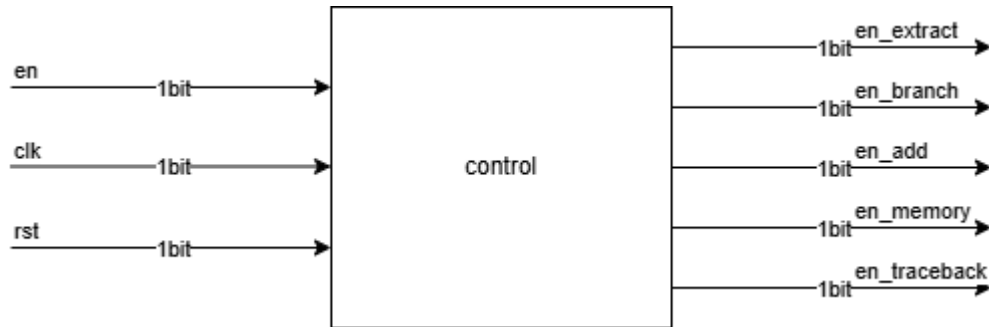
### 2.3. Đầu vào đầu ra

Bảng 3: Đầu vào, đầu ra khối Top

Tên chân	Số bit	I/O	Chức năng
clk	1	I	Sườn lên của tín hiệu clk điều khiển và thực hiện lệnh trong mạch tuần tự
rst	1	I	Tín hiệu thiết lập lại các giá trị ban đầu
en	1	I	Là tín hiệu báo hiệu bắt đầu quá trình giải mã
i_data	16	I	Data đầu vào 16bit cần giải mã
o_data	8	O	Data đầu ra 8bit đã giải mã xong
o_done	1	O	Tín hiệu báo hiệu giải mã xong

### 3. Module khối control

#### 3.1. Sơ đồ khối



Hình 6. Sơ đồ khối control

#### 3.2. Chức năng

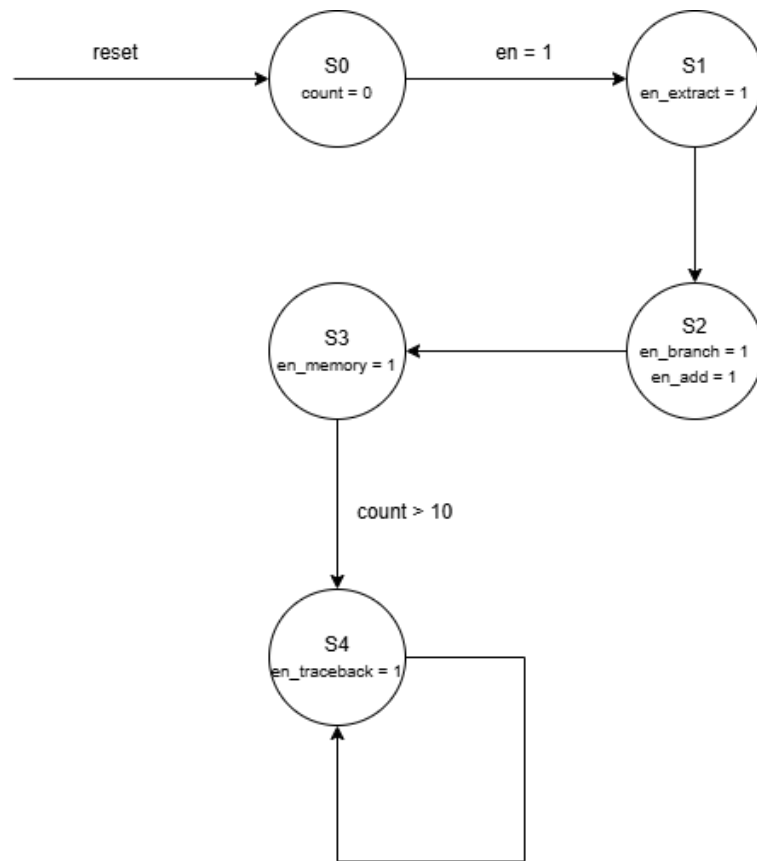
Khối control trong viterbi decoder chứa logic và các trạng thái để thực hiện các chức năng điều khiển hoạt động của các khối đã nêu ở trên thông qua đầu ra là tín hiệu điều khiển, nó giúp cho bộ giải mã có thể hoạt động được đúng với chức năng đã đề ra.

#### 3.3. Đầu vào đầu ra

Bảng 4: Đầu vào đầu ra khối control

Tên chân	Số bit	I/O	Reg	Chức năng
en	1	I		Tín hiệu để báo hiệu một đầu vào nhận được mới đã sẵn sàng để được giải mã.
clk	1	I		Suôn dương của tín hiệu clk điều khiển hoạt động mạch tuần tự
rst	1	I		Tín hiệu thiết lập lại trạng thái ban đầu
en_extract	1	O	reg	Tín hiệu điều khiển khối extract_bit hoạt động
en_branch	1	O	reg	Tín hiệu điều khiển khối branch_metric hoạt động
en_add	1	O	reg	Tín hiệu điều khiển khối add_compare_select hoạt động
en_memory	1	O	reg	Tín hiệu điều khiển khối memory hoạt động
en_traceback	1	O	reg	Tín hiệu điều khiển khối traceback_output_decision hoạt động

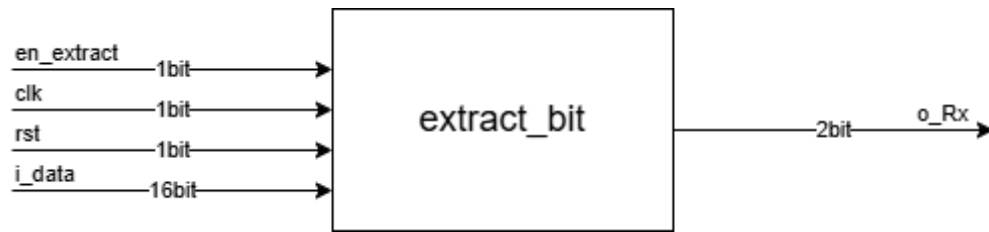
### 3.4. Sơ đồ máy trạng thái FSM



Hình 7. Sơ đồ máy trạng thái cho khối control

## 4. Module khối extract\_bit

### 4.1. Sơ đồ khối



Hình 8. Sơ đồ khối extract\_bit

### 4.2. Chức năng

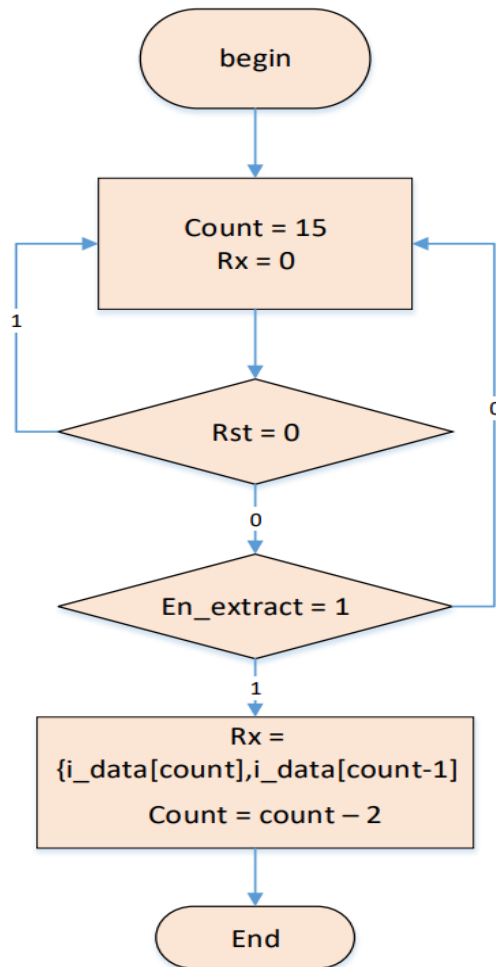
Khối extract\_bit có chức năng chia data đầu vào 16 bit thành từng cặp 2 bit một để gửi đi cho quá trình giải mã tiếp theo. Việc sử dụng một đầu ra 2 bit giúp giảm số lượng đầu ra và logic cần thiết trong mạch, dẫn đến tiết kiệm diện tích mạch và tài nguyên phần cứng. Đồng thời số lượng đầu ra ít hơn có thể giảm độ phức tạp của mạch, làm cho việc thiết kế, kiểm tra và bảo trì dễ dàng hơn.

### 4.3. Đầu vào đầu ra

Bảng 5: Đầu vào, đầu ra khối extract\_bit

Tên chân	Số bit	I/O	Reg	Chức năng
en_extract	1	I		Tín hiệu điều khiển khối extract_bit hoạt động
clk	1	I		Sườn dương của tín hiệu clk điều khiển hoạt động mạch tuần tự
rst	1	I		Tín hiệu thiết lập trạng thái ban đầu
i_data	16	I		Data đầu vào 16bit cần giải mã
o_Rx	2	O	reg	Hai bit đưa vào các khối tiếp theo để giải mã

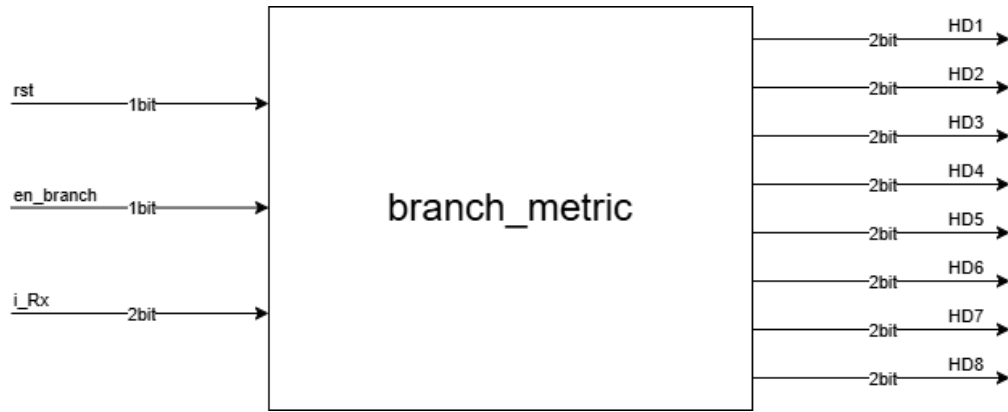
#### 4.4. Lưu đồ thuật toán



Hình 9. Lưu đồ thuật toán khối extract\_bit

## 5. Module khối branch\_metric

### 5.1. Sơ đồ khối branch\_metric



Hình 10. Sơ đồ khối branch\_metric

### 5.2. Chức năng

Khối branch\_metric có chức năng nhận tín hiệu từ khối extract\_bit, chứa các logic để tính độ sai khác của 2 bit đầu vào với đầu ra giữa các nút(Hamming) và cho ra 6 đầu ra 2 bit để sử dụng cho khối Add\_compare\_select.

Bảng 6: Khoảng cách hamming giữa các nút

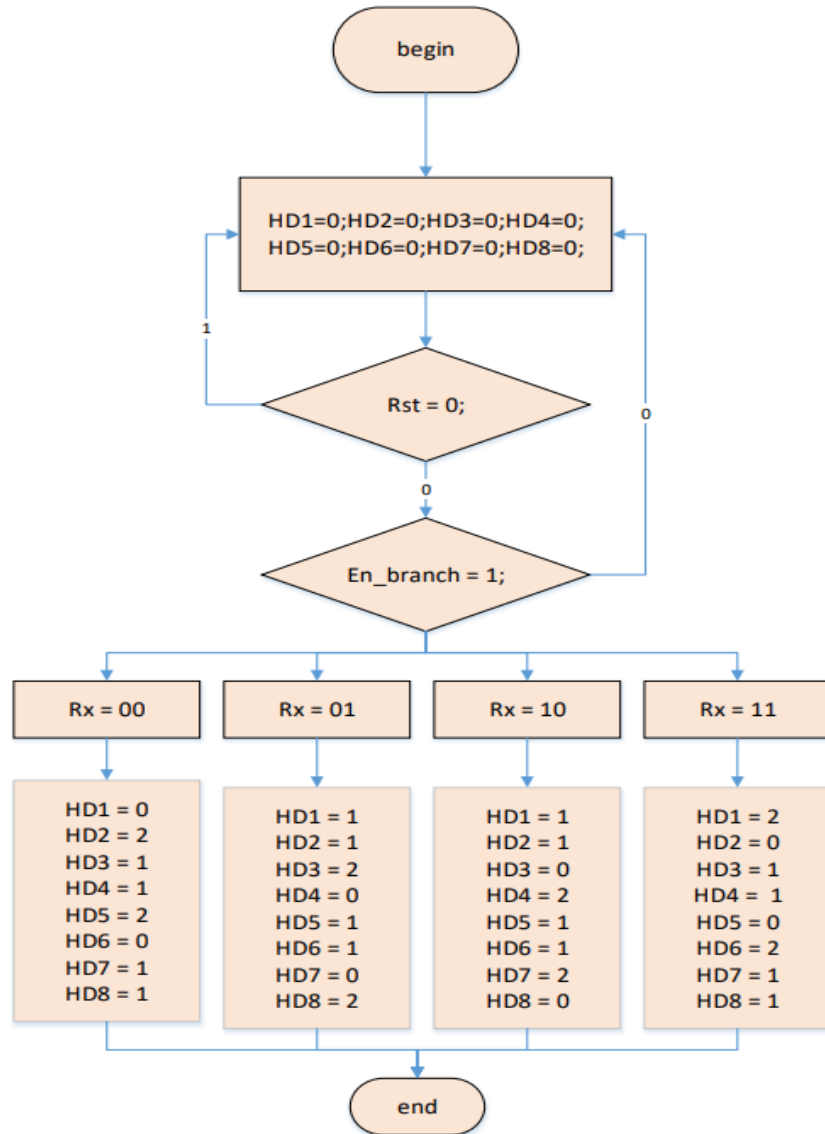
Nút	00	10	01	11
00	HD1	HD2	-	-
10	-	-	HD3	HD4
01	HD5	HD6	-	-
11	-	-	HD7	HD8

### 5.3. Đầu vào, đầu ra

Bảng 7. Đầu vào, đầu ra khối Branch\_metric

Tên chân	Số bit	I/O	Reg	Chức năng
rst	1	I		Tín hiệu thiết lập lại trạng thái ban đầu
en_branch	1	I		Tín hiệu điều khiển khối branch_metric hoạt động
i_Rx	2	I		Hai bit đầu vào thực hiện giải mã
HD1-HD5	2	O	Reg	Độ sai khác giữa đầu vào và đầu ra tại nút 00
HD2-HD6	2	O	Reg	Độ sai khác giữa đầu vào và đầu ra tại nút 10
HD3-HD7	2	O	Reg	Độ sai khác giữa đầu vào và đầu ra tại nút 01
HD4-HD8	2	O	Reg	Độ sai khác giữa đầu vào và đầu ra tại nút 11

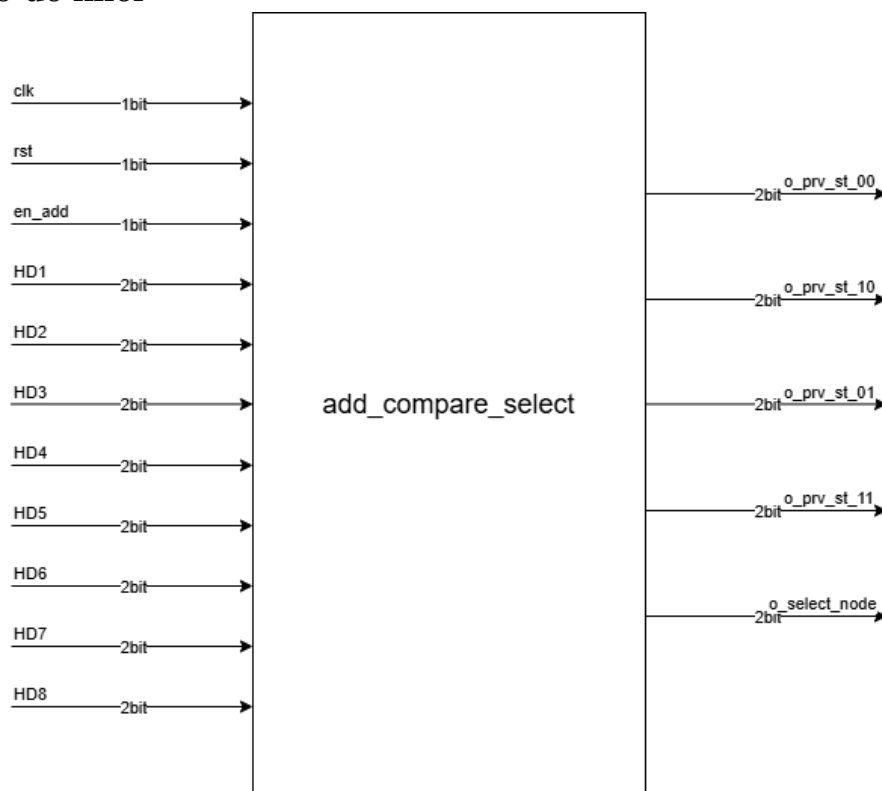
#### 5.4. Lưu đồ thuật toán



Hình 11. Lưu đồ thuật toán khối *branch\_metric*

## 6. Module khối add\_compare\_select

### 6.1. Sơ đồ khối



Hình 12. Sơ đồ khối `add_compare_select`

### 6.2. Chức năng

Khối `add_compare_select` nhận các tín hiệu khoảng cách hamming từ khối `branch_metric`, có chức năng tính toán so sánh và lựa chọn ra tổng hamming nhỏ nhất tại các nút và các nút dịch chuyển tới nút đó, đồng thời gửi các giá trị đó tới khối `memory` và giá trị nút có hamming nhỏ nhất tới khối `traceback_output_decision`.

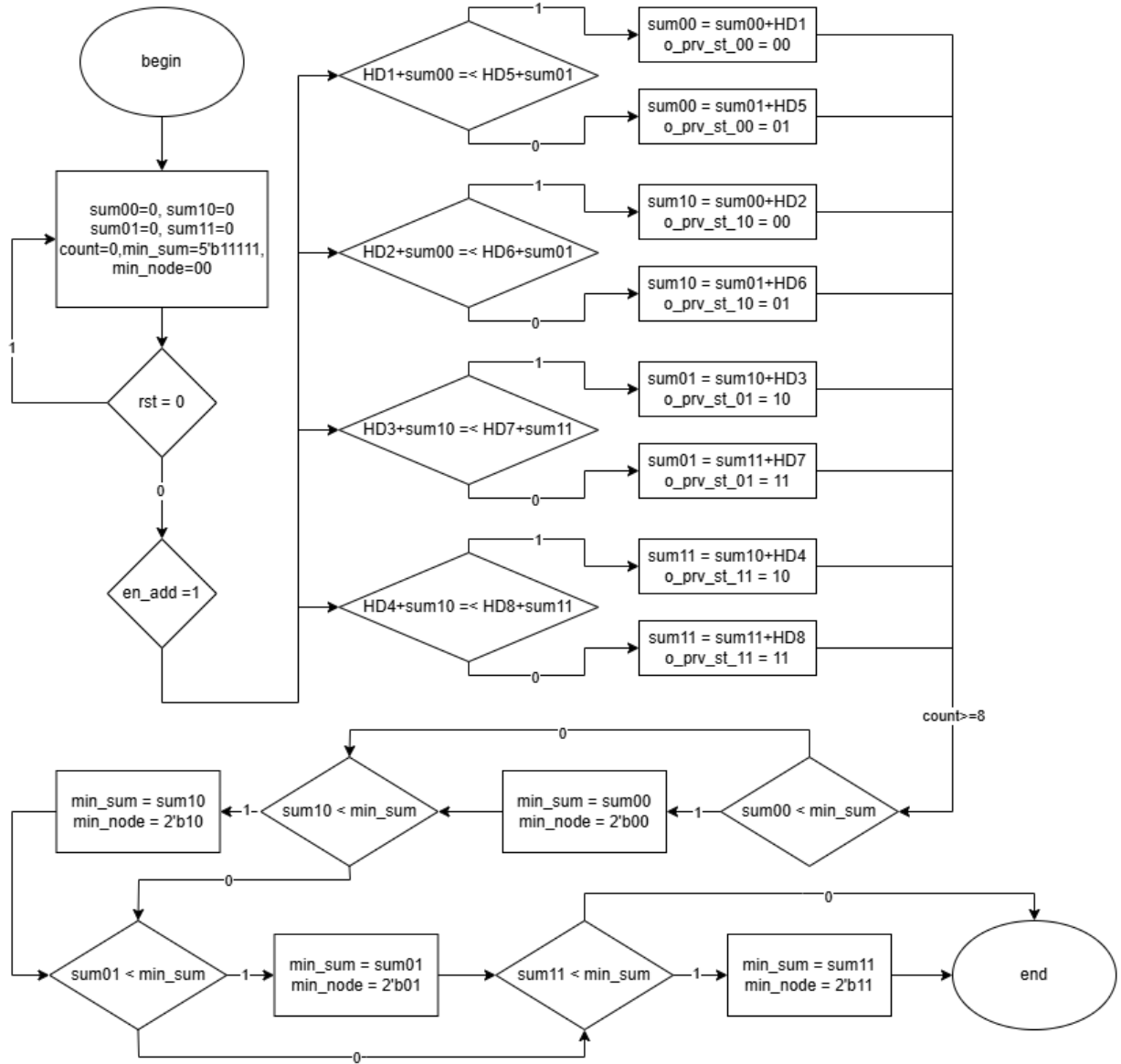


### 6.3. Đầu vào, đầu ra

*Bảng 8. Đầu vào, đầu ra khối add\_compare\_select*

Tên chân	Số bit	I/O	reg	Chức năng
clk	1	I		Sườn dương của tín hiệu clk điều khiển hoạt động mạch tuần tự
rst	1	I		Tín hiệu thiết lập lại trạng thái ban đầu
en_add	1	I		Tín hiệu điều khiển khối add_compare_select
HD1-HD5	2	I		Khoảng cách hamming tại nút 00
HD2-HD6	2	I		Khoảng cách hamming tại nút 10
HD3-HD7	2	I		Khoảng cách hamming tại nút 01
HD4-HD8	2	I		Khoảng cách hamming tại nút 11
o_prv_st_00	2	O	reg	Nút trước khi chuyển sang nút 00
o_prv_st_10	2	O	reg	Nút trước khi chuyển sang nút 10
o_prv_st_01	2	O	reg	Nút trước khi chuyển sang nút 01
o_prv_st_11	2	O	reg	Nút trước khi chuyển sang nút 11
o_select_node	2	O	reg	Nút có tổng khoảng cách hamming nhỏ nhất

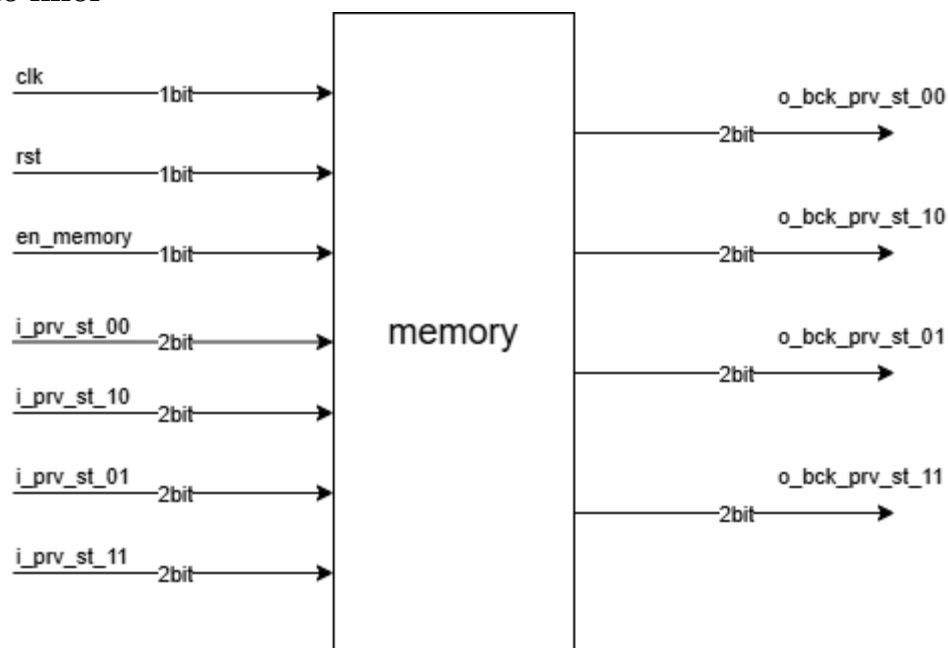
## 6.4. Lưu đồ thuật toán



Hình 13. Lưu đồ thuật toán khối *add\_compare\_select*

## 7. Module memory

### 7.1. Sơ đồ khối



Hình 14. Sơ đồ khối memory

### 7.2. Chức năng

Khối memory có chức năng lưu trữ sự dịch chuyển của các nút tại từng thời điểm khác nhau và lưu thành tập hợp 4 thanh ghi với 8 phần tử để sử dụng cho khối traceback\_output\_decision.

### 7.3. Đầu vào, đầu ra

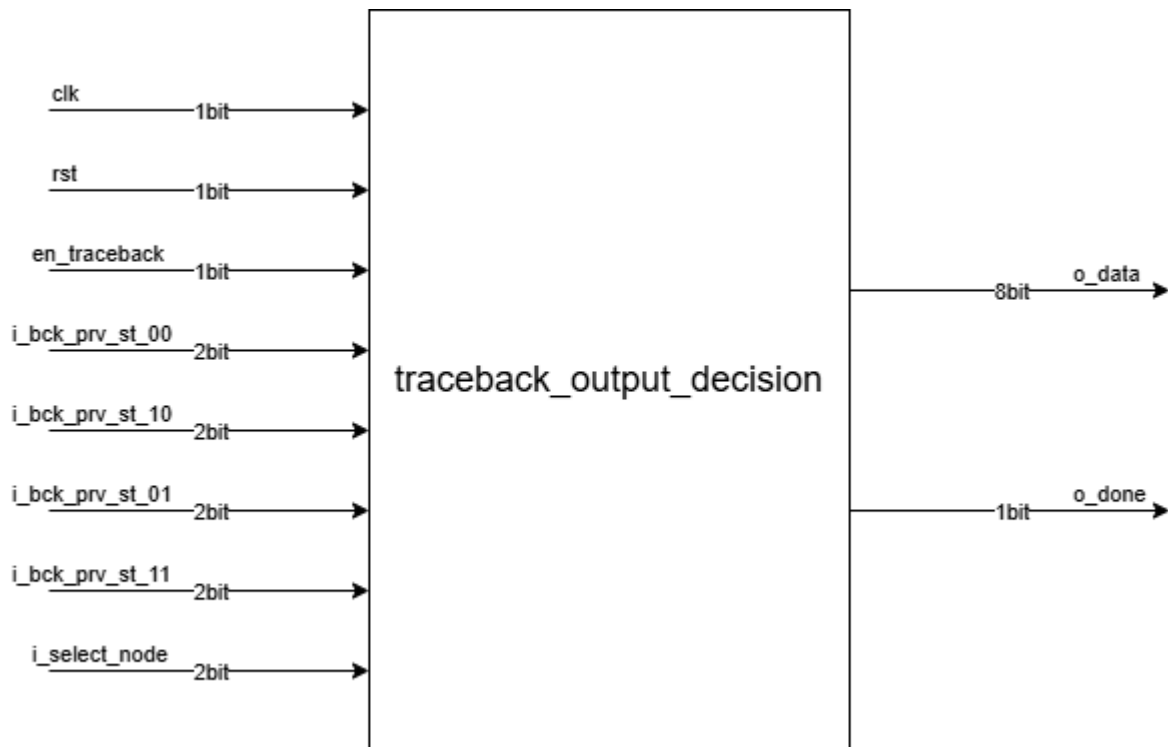
Bảng 9. Đầu vào, đầu ra khối memory

Tên chân	Số bit	I/O	reg	Chức năng
clk	1	I		Sườn dương của tín hiệu clk điều khiển hoạt động mạch tuần tự
rst	1	I		Tín hiệu thiết lập lại trạng thái ban đầu
en_memory	1	I		Tín hiệu điều khiển khối memory hoạt động
i_prv_st_00	2	I		Nút đi đến nút 00
i_prv_st_10	2	I		Nút đi đến nút 10
i_prv_st_01	2	I		Nút đi đến nút 01
i_prv_st_11	2	I		Nút đi đến nút 11
o_bck_prv_st_00	2	O	reg	Nút đi đến nút 00

o_bck_prv_st_10	2	O	reg	Nút đi đến nút 10
o_bck_prv_st_01	2	O	reg	Nút đi đến nút 01
o_bck_prv_st_11	2	O	reg	Nút đi đến nút 11

## 8. Module traceback\_output\_decision

### 8.1. Sơ đồ khối



Hình 15. Sơ đồ khối traceback\_output\_decision

### 8.2. Chức năng

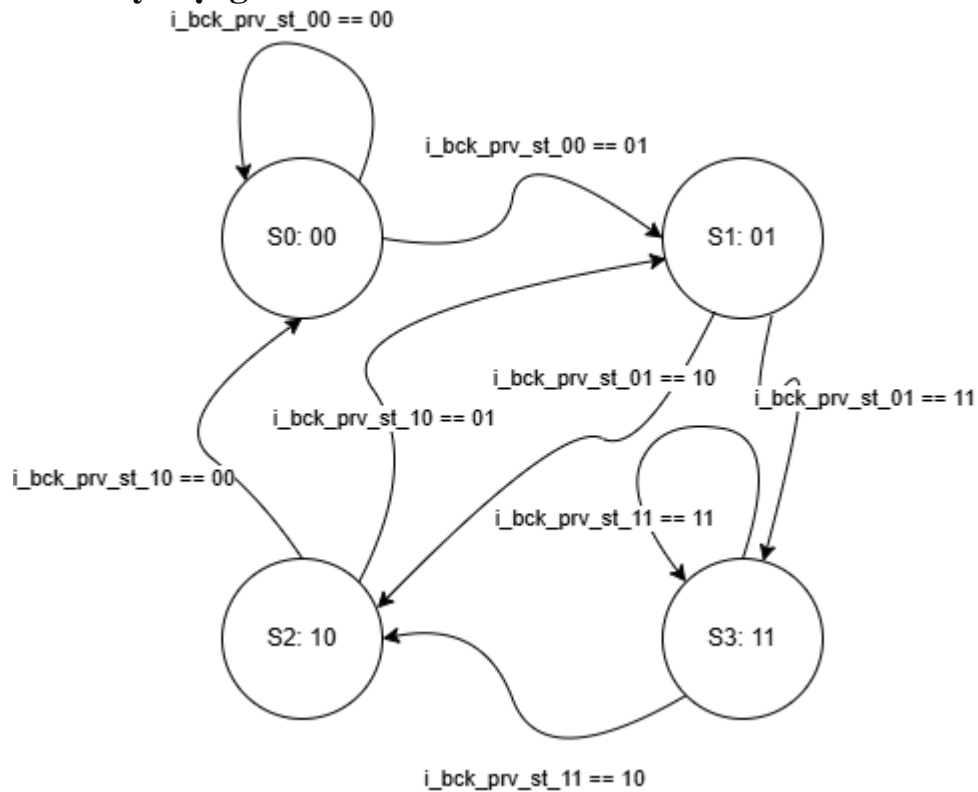
Khối traceback\_output\_decision nhận các tín hiệu trạng thái trước đây của từng nút từ khối memory và tín hiệu chọn nút có hamming nhỏ nhất của khối add\_compare\_select, có chức năng truy suất lại đường đi tối ưu nhất để đưa ra 8 bit giải mã và tín hiệu đã hoàn thành giải mã.

### 8.3 Đầu vào, đầu ra

Bảng 10. Đầu vào, đầu ra khối traceback\_output\_decision

Tên chân	Số bit	I/O	reg	Chức năng
clk	1	I		Suồng dương của tín hiệu clk điều khiển hoạt động mạch tuần tự
rst	1	I		Tín hiệu thiết lập lại trạng thái ban đầu
en_traceback	1	I		Tín hiệu điều khiển khối Traceback hoạt động
i_bck_prv_st_00	2	I		Nút đi đến nút 00 tại một thời điểm
i_bck_prv_st_10	2	I		Nút đi đến nút 10 tại một thời điểm
i_bck_prv_st_01	2	I		Nút đi đến nút 01 tại một thời điểm
i_bck_prv_st_11	2	I		Nút đi đến nút 11 tại một thời điểm
i_select_node	2	I		Nút có khoảng cách hamming nhỏ nhất
o_data	8	O	reg	Data đã giải mã xong
o_done	1	O	reg	Tín hiệu báo hiệu đã traceback xong 8bit

## 8.4. Máy trạng thái FSM

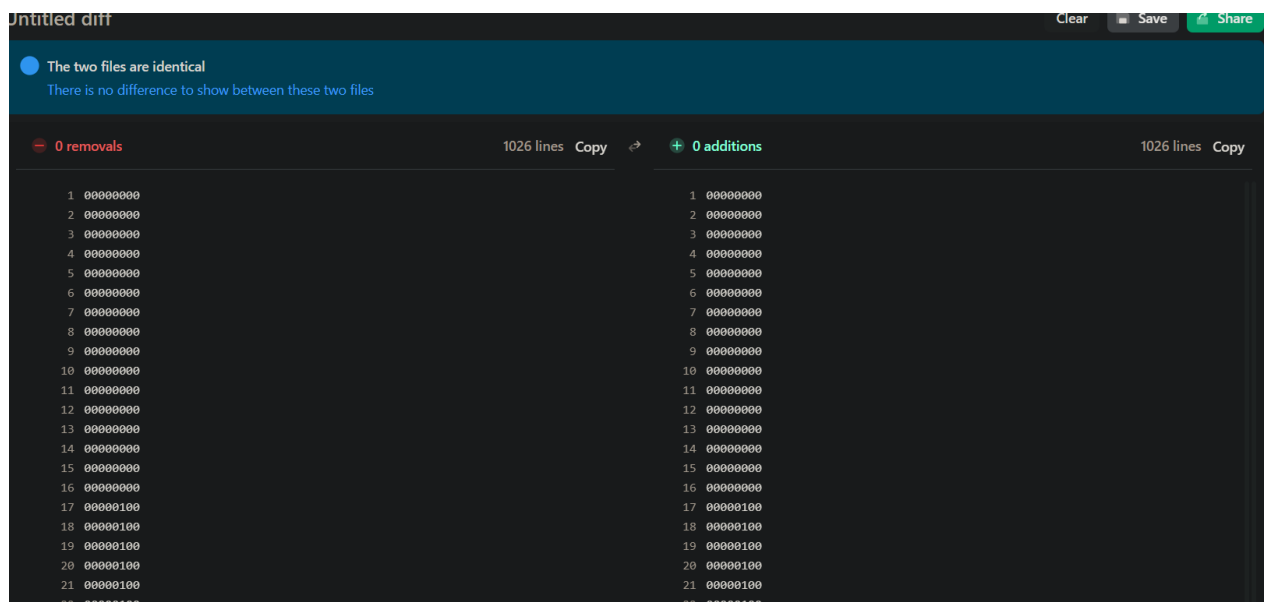


Hình 16. Máy trạng thái FSM khối `traceback_output_decision`

## 9. Kết quả đạt được

Hệ thống được triển khai, tổng hợp, bố trí sử dụng EDA tools của Cadence

- **Gedit:** RTL Design
- **Genus:** Synthesis
- **Innovus:** Physical Design



*Hình 17. So sánh Golden Output*

```

#-----#
# Timing Point      Flags      Arc      Edge      Cell      Fanout Load Trans Delay Arrival Instance
#                  (fF) (ps) (ps) (ps) (ps) Location
#-----#
Ex1_o Rx_reg[0]/CK -      -      R      (arrival) 109      -      0      0      0      0      (-,-)
Ex1_o Rx_reg[0]/Q -      CK->Q R      DFFRHQX1  4 11.3    79     95     95     95     (-,-)
g6626__2346/Y      -      AN->Y R      NAND2BX1  2 6.7      61     51     146    146     (-,-)
g6621__4733/Y      -      A->Y  F      NAND2X1   2 6.5      91     71     217    217     (-,-)
g6615__6131/Y      -      A->Y  R      NOR2X1   2 6.9      122    83     300    300     (-,-)
g6589__4733/C0     -      A->C0 R      ADDHX1   3 8.9      64     77     377    377     (-,-)
g6585/Y            -      A->Y  F      INVX1    3 9.0      71     56     433    433     (-,-)
g6583__7098/Y      -      A->Y  R      NOR2X1   2 6.7      96     76     509    509     (-,-)
g6580/Y            -      A->Y  F      INVX1    1 4.3      47     49     558    558     (-,-)
g6559__7410/Y      -      B0->Y R      OAI31X1  3 8.7      184    51     609    609     (-,-)
g6536__6783/Y      -      B0->Y F      AOI2BB1X1 1 4.3      66     80     690    690     (-,-)
g6528__6417/Y      -      C0->Y R      AOI221X1  1 4.2      106    83     773    773     (-,-)
g6516__1881/Y      -      B->Y  R      OR2X1    2 6.5      47     54     827    827     (-,-)
g6487__4733/Y      -      A1N->Y R      AOI2BB1X1 1 4.4      67     54     881    881     (-,-)
g6478__2802/Y      -      B0->Y F      AOI21X1   1 4.3      68     41     922    922     (-,-)
g6472__4319/Y      -      A1->Y R      OAI22X1   1 4.4      85     67     989    989     (-,-)
g6468__5477/Y      -      B0->Y F      AOI2BB1X1 2 6.4      90     78    1067    1067     (-,-)
g6769/Y            -      B->Y  R      NOR2BX1  6 16.1     220    151    1218    1218     (-,-)
g5767__6161/Y      -      A1->Y F      AOI22X1   1 4.3      99     119    1337    1337     (-,-)
g5721__7098/Y      -      B0->Y R      AOI2BB1X1 1 4.3      51     52    1389    1389     (-,-)
Add1_sum11_reg[3]/D <<< -      R      DFFRHQX1  1 -        -      0     1389    1389     (-,-)
#-----#

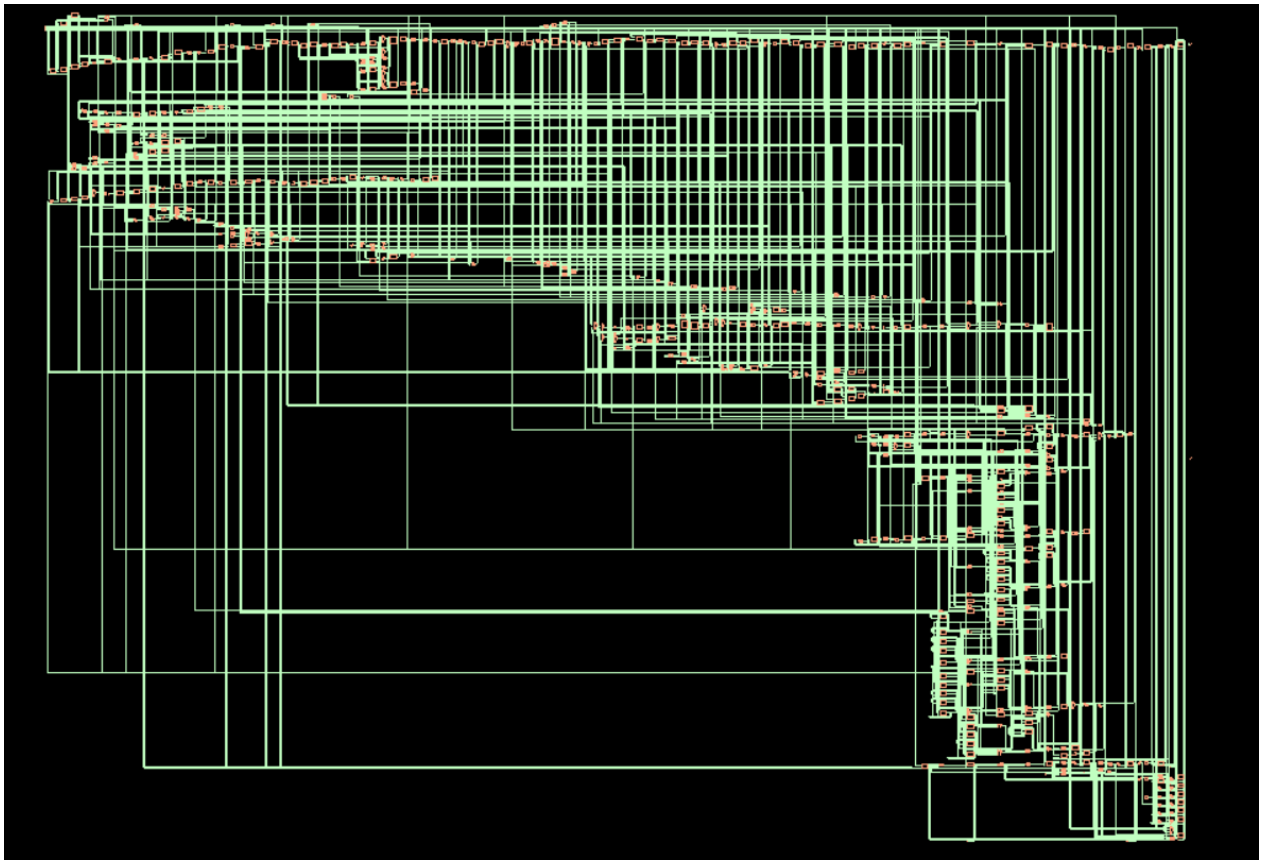
@genus:root: 12> report_area
=====
Generated by:      Genus(TM) Synthesis Solution 23.10-p004_1
Generated on:      Jan 16 2025 04:02:57 pm
Module:            viterbi_decoder
Operating conditions: PVT_1P1V_0C
Interconnect mode: global
Area mode:         physical library
=====

Instance  Module  Cell-Count  Cell-Area  Net-Area  Total-Area
-----
viterbi_decoder NA          547    1607.742    696.869    2304.611
@genus:root: 13>

```

Hình 18. Kết quả Synthesis





*Hình 19. Kết quả Synthesis*

```

@innovus 19> check_drc
#-check_ndr_spacing auto           # enums={true false auto}, default=auto, user setting
#-check_same_via_cell true         # bool, default=false, user setting
#-report viterbi_decoder.drc.rpt   # string, default="", user setting
*** Starting Verify DRC (MEM: 3105.4) ***

VERIFY DRC ..... Starting Verification
VERIFY DRC ..... Initializing
VERIFY DRC ..... Deleting Existing Violations
VERIFY DRC ..... Creating Sub-Areas
VERIFY DRC ..... Using new threading
VERIFY DRC ..... Sub-Area: {0.000 0.000 146.175 146.160} 1 of 1
VERIFY DRC ..... Sub-Area : 1 complete 0 Viols.

Verification Complete : 0 Viols.

*** End Verify DRC (CPU TIME: 0:00:00.2  ELAPSED TIME: 0:00:00.0  MEM: 272.1M) ***

@innovus 20> set_db check_drc_area {0 0 0 0}

```

*Hình 20. Kết quả DRC Check*

```

@innovus 15> check_connectivity -type all -check_geometry_loops -ignore_soft_pg_connects -error 1000 -warning 50
VERIFY_CONNECTIVITY use new engine.

***** Start: VERIFY CONNECTIVITY *****
Start Time: Thu Jan 16 19:21:26 2025

Design Name: viterbi_decoder
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (146.1750, 146.1600)
Error Limit = 1000; Warning Limit = 50
Check all nets

Begin Summary
  Found no problems or warnings.
End Summary

End Time: Thu Jan 16 19:21:26 2025
Time Elapsed: 0:00:00.0

***** End: VERIFY CONNECTIVITY *****
Verification Complete : 0 Viols. 0 Wrngs.
(CPU Time: 0:00:00.1  MEM: 0.000M)

```

*Hình 21. Kết quả Connectivity Check*