

ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN-ĐIỆN TỬ

BỘ MÔN ĐIỆN TỬ

--- & ---



ĐỒ ÁN MÔN HỌC

**Đề tài : TÌM HIỂU VÀ THIẾT KẾ
TCAM 104-BIT DỰA TRÊN SRAM**

Giảng viên hướng dẫn: Trịnh Vũ Đăng Nguyên

<i>Họ và tên SVTH</i>	<i>MSSV</i>
Hoàng Nguyễn Bảo Long	1812861
Ngô Minh Nhân	1813327
Nguyễn Hoàng Minh Nhật	1810394
Hồ Bá Phước	1813638

TP Hồ Chí Minh, 03 Tháng 08 Năm 2021

MỤC LỤC

1. GIỚI THIỆU.....	1
1.1 Tổng quan.....	1
1.2 Mục tiêu đề tài.....	1
1.3 Nội dung đề tài.....	1
2. LÝ THUYẾT TỔNG QUAN.....	3
3. NỘI DUNG ĐỀ TÀI.....	6
3.1 Nguyên lý hoạt động của bộ nhớ TCAM.....	6
4. DỰ KIẾN KẾT QUẢ ĐẠT ĐƯỢC.....	12
4.1 Kết quả sơ khởi đã đạt được.....	12
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	17
6. KẾ HOẠCH THỰC HIỆN.....	18
TÀI LIỆU THAM KHẢO.....	19

DANH MỤC MINH HOẠ

Hình 2-1. Minh họa của một khối CAM.....	3
Hình 2-2. Minh họa một khối TCAM.....	4
Hình 3-1. Một khối ram đặc trưng.....	6
Hình 3-2. Ý tưởng cơ bản của định địa chỉ	6
Hình 3-3. Key được cắt ra thành nhiều phần và sử dụng để tạo ra match vector.....	7
Hình 3-4. Luồng dữ liệu cơ bản của cấu trúc TCAM.....	7
Hình 3-4. Sơ đồ giải thuật.....	10
Hình 4-1 Quá trình điền Rule vào RAM.....	12
Hình 4-2 Quá trình tăng Key để kiểm tra được những rule nào là hợp lệ với key đã được đưa vào trong TCAM.....	12
Hình 4-3 Kết quả tổng hợp phần cứng của RAM.....	13
Hình 4-4 Kết quả tổng hợp phần cứng trên FPGA.....	13
Hình 4-5: Kết quả mô phỏng tra key của TCAM.....	14

DANH MỤC BẢNG

Bảng 3-1. Các chú thích được sử dụng để giải thích giải thuật.....	5
Bảng 3-2. Basic Rule.....	7
Bảng 3-3. Nội dung khi được ghi trong RAM.....	8
Bảng 3-4. Nội dung của RAM (Word[1:0]).....	9
Bảng 3-5. Nội dung của RAM (Word[3:2]).....	10

1. GIỚI THIỆU

1.1 Tổng quan

Bộ nhớ định địa chỉ nội dung bậc ba (TCAM) là bộ nhớ cho phép tìm kiếm dữ liệu với tốc độ cao. Nó không chỉ được thừa nhận là bộ nhớ / lưu trữ kết hợp mà còn TCAM có thể so sánh nội dung tìm kiếm đầu vào (khóa) với tập hợp dữ liệu tích lũy và trả về địa chỉ phù hợp tương thích với dữ liệu tìm kiếm đầu vào này. TCAM dựa trên SRAM sử dụng và phân bổ các khối RAM để thực hiện ứng dụng TCAM trên phần cứng FPGA.

Thay vì chỉ lưu trữ dữ liệu nhị phân, TCAM có thể giải mã và lưu trữ trạng thái X (không cần quan tâm). Do đó, TCAM thực hiện đối sánh từng phần, được kích hoạt bởi trạng thái ký tự đại diện “x”, có thể dẫn đến đối sánh có nhiều từ. TCAM bao gồm mảng ô SRAM và bộ mã hóa ưu tiên. Với mã hóa ưu tiên, đầu ra của TCAM là kết quả được ưu tiên cao nhất. TCAM được sử dụng rộng rãi trong các bộ định tuyến mạng như bộ đệm dịch-nhìn-sang một bên (TLB) bộ nhớ đệm trong bộ vi xử lý, bộ tăng tốc cơ sở dữ liệu trong phân tích dữ liệu lớn và trong nhận dạng mẫu.

Không thể phủ nhận tính linh hoạt của phần mềm và hiệu suất gần ASIC của mảng cổng lập trình trường (FPGA) được gọi là phần cứng có thể cấu hình lại. Các thiết bị FPGA tối tân như Cyclone IV có một số lợi thế hiệu quả như tốc độ xung nhịp cao, tiêu hao điện năng thấp, tài nguyên trên chip phong phú và lượng lớn được nhúng với độ rộng từ có thể định cấu hình. Lý do vì nhu cầu về TCAM phải được tích hợp đơn giản, ngày càng có tầm quan trọng trong việc sử dụng các thiết bị FPGA để triển khai các công cụ tìm kiếm tương đương với TCAM.

1.2 Mục tiêu đề tài

Đề tài trình bày về thuật toán và nguyên lý hoạt động của TCAM. Theo sau đó là thiết kế phần cứng của một bộ nhớ TCAM, trình bày làm thế nào để ghi dữ liệu và tìm kiếm trong TCAM. Các thiết kế và nguyên lý hoạt động này sẽ được kiểm chứng bằng mô phỏng.

1.3 Nội dung đề tài

- Tìm hiểu tổng quan lý thuyết TCAM.
- Lập giải thuật cho thiết kế.

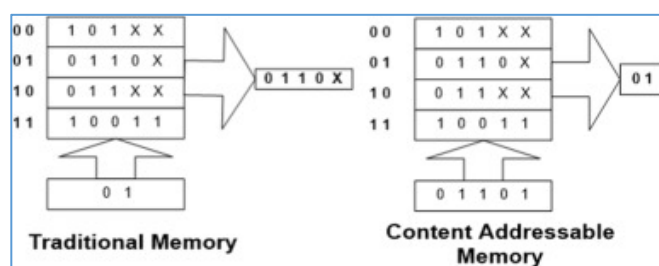
-
- Sử dụng ngôn ngữ lập trình Verilog viết chương trình cho giải thuật.
 - Sử dụng phần mềm kiểm tra chương trình và mô phỏng thiết kế. Từ đó, chỉnh sửa chương trình cho hoàn thiện và có được kết quả mong muốn.
 - Kết luận và đưa ra hướng phát triển cho đề tài.

2. LÝ THUYẾT TỔNG QUAN

2.1 Tổng quan về CAM/TCAM

2.1.1 CAM

Bộ nhớ tìm kiếm theo nội dung CAM (Content Addressable Memory) là bộ nhớ lưu trữ giá trị hoặc từ khóa trong khối RAM, cho phép truy cập tìm kiếm theo nội dung và trả về kết quả tìm kiếm dưới dạng địa chỉ ô nhớ trùng khớp trong một thời gian rất nhanh và cố định, thường là trong một vài chu kỳ hệ thống. Một từ tìm kiếm trong bộ nhớ CAM sẽ cho ra kết quả là trùng khớp (MATCH) hoặc không trùng khớp, có thể đưa ra cả địa chỉ trùng khớp nếu cần thiết. Bộ nhớ CAM có khả năng tìm kiếm song song cho một số lượng rất lớn các nội dung và có cơ chế tổ chức rất khác so với các phương pháp tìm kiếm theo dạng duyệt tuần tự. Trên thực tế, thời gian trễ của thuật toán tìm kiếm tuần tự trên bộ nhớ RAM tỷ lệ thuận với số lượng từ khóa và thường không cố định nên không thể áp dụng cho các thiết bị chuyển mạch thời gian thực.



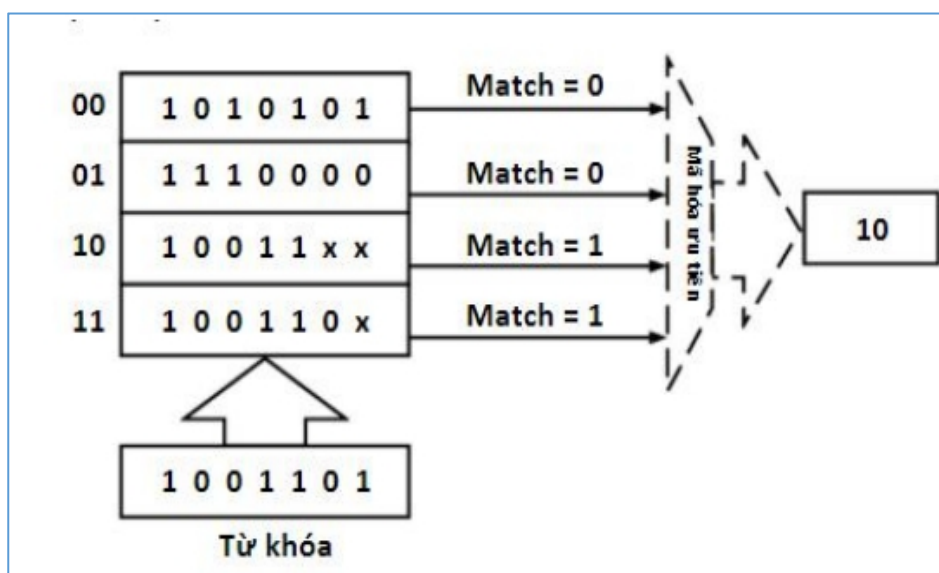
Hình 2-1. Minh họa của một khối CAM.

Như trình bày trên Hình 2-1, một từ khóa sẽ được tìm kiếm song song ở bốn vị trí đã lưu từ CAM. Tại các địa chỉ 00, 01, 10 sẽ cho ra kết quả MATCH = 0 do từ đã lưu khác với từ khóa tìm kiếm. Tại địa chỉ 11 sẽ cho ra kết quả MATCH = 1. Kết quả là địa chỉ của từ cho kết quả MATCH = 1.

2.1.2. TCAM

TCAM (Ternary Content Addressable Memory) là bộ nhớ kết hợp đặc biệt, trong đó có thể lưu các giá trị là 0, 1 hoặc X (don't care). Khác với CAM chỉ có thể biểu diễn một giá trị thì TCAM có thể biểu diễn một khoảng giá trị, khoảng giá trị này được biểu diễn bởi các giá trị X. Giá trị X có thể nhận 0 hoặc 1 nên bộ nhớ lưu trữ của TCAM có kèm thêm tham số mặt nạ thay vì chỉ có từ khóa hoặc giá trị. Mặt nạ là một

tham số đi kèm để xác định vị trí chứa bit X trong từ khóa hay giá trị. Khả năng khớp với một khoảng giá trị rất quan trọng và được ứng dụng trong ACL (Access Control List) của switch thông minh khi thực hiện lọc các gói tin theo dài IP (Internet Protocol), hoặc dài Port phần mềm, hoặc trong tìm kiếm địa chỉ IP trong thiết bị định tuyến... Tuy vậy thiết kế cũng như phần cứng sử dụng cho TCAM thường phức tạp và lớn hơn so với CAM. Giao diện tra cứu của TCAM nhận từ khóa tra cứu và đưa ra kết quả có chứa một tín hiệu cho biết từ khóa đó có khớp (MATCH) với bất kì một từ TCAM nào đã được lưu trong bộ nhớ hay không. Nếu có bất kì một vị trí khớp, bộ nhớ TCAM sẽ đưa ra địa chỉ của từ khóa đó, một từ khóa có thể được khớp với nhiều vị trí được lưu. Do có nhiều vị trí khớp nên bộ nhớ TCAM sẽ có thêm bộ mã hóa ưu tiên để lựa chọn vị trí cần tìm.



Hình 2-2. Minh họa một khối TCAM

Có thể thấy ở Hình 2-2, một từ khóa có giá trị 1001101 được đưa vào bộ nhớ TCAM để tìm kiếm, ở hai địa chỉ 00 và 01 khác với từ khóa nên sẽ cho kết quả MATCH = 0. Tại địa chỉ 10, do từ TCAM đã lưu có dạng 10011xx nên từ khóa tìm kiếm chỉ cần trùng khớp với năm bit bắt đầu, hai bit còn lại có thể nhận bất kì giá trị nào sẽ cho kết quả MATCH = 1. Tại địa chỉ 11, từ TCAM đã lưu có dạng 100110x nên từ khóa tìm kiếm chỉ cần trùng khớp với sáu bit đầu, bit còn lại có thể nhận bất kì giá trị nào sẽ cho kết quả MATCH = 1. Quá trình tìm kiếm có thể cho ra kết quả là địa chỉ của từ được tìm kiếm, tuy nhiên để đưa ra địa chỉ cuối cùng, kết quả MATCH sẽ

được đưa vào bộ mã hóa ưu tiên(Priority Encoder), chọn ưu tiên vị trí bit nhỏ nhất hay bit lớn nhất đầu tiên xuất hiện bit 1 thì tùy thiết kế của bộ mã hóa ưu tiên.

3. NỘI DUNG ĐỀ TÀI

3.1 Nguyên lý hoạt động của bộ nhớ TCAM

Phần này trình bày tổng quan về cách thức làm việc của bộ nhớ tìm kiếm theo nội dung và cách tổ chức bộ nhớ dựa trên bộ nhớ RAM truyền thống. Thiết kế CAM và TCAM sử dụng các khối SRAM tương tự như bộ nhớ truyền thống nhưng có sự khác biệt.

Đối với bộ nhớ truyền thống, người dùng cung cấp dữ liệu đầu vào và địa chỉ cho thao tác ghi, với thao tác đọc thì một địa chỉ sẽ được cung cấp và dữ liệu được lưu trữ trong địa chỉ đó sẽ được đọc ra. Còn đối với CAM và TCAM thì thao tác ghi giống với bộ nhớ truyền thống, thao tác đọc thì người dùng cung cấp đầu vào dữ liệu để tra cứu địa chỉ nơi dữ liệu đó được lưu trữ. Giống như một bộ nhớ truyền thống chỉ lưu trữ một từ dữ liệu tại một địa chỉ, mỗi địa chỉ CAM / TCAM chỉ có thể lưu trữ một dữ liệu duy nhất. Tuy nhiên, đối với TCAM một dữ liệu đó có thể được lưu tại nhiều địa chỉ khác nhau.

Một bộ nhớ TCAM sẽ được viết dưới dạng $N \times W$, trong đó N là số từ TCAM (độ sâu) và W là số bit TCAM (độ rộng). Một bộ nhớ TCAM sẽ được tổ chức dựa trên bộ nhớ RAM, khi đó W bit TCAM sẽ cần có 2^W ô nhớ địa chỉ của RAM tương ứng. Địa chỉ RAM khớp với từ TCAM thì ô nhớ sẽ ghi giá trị 1, ngược lại ô nhớ sẽ ghi giá trị 0.

3.2 Giải thuật thiết kế TCAM

3.2.1 Giải thuật

Ký hiệu	Mô tả
N	Độ sâu của TCAM(số từ)
W	Độ rộng của TCAM (số bit của Word TCAM)
S	Kích cỡ TCAM ($N * W$)
D	Số register của mỗi RAM (2^W)
K	Key đưa vào
Rule	Địa chỉ của bảng TCAM
T	Ternary word (giá trị của Rule)

Bảng 3-1. Các chú thích được sử dụng để giải thích giải thuật

Trong một TCAM $N \times W$, W bit nhị phân của key đưa vào được tra và được chuyển đổi chính xác thành một match vector N bit nhị phân. Khi sử dụng một $2^w \times N$ RAMs nó đạt được chức năng tương tự với TCAM gốc. Mỗi RAM lưu trữ từ TCAM và sử dụng giá trị của khóa đầu vào để tìm kiếm địa chỉ trong RAM

Thuật toán sử dụng RAM để thực hiện TCAM:

- Nguyên tắc 1: Ghi vào RAM

RAM [Địa chỉ][Quy tắc] = 1 nếu địa chỉ = T.

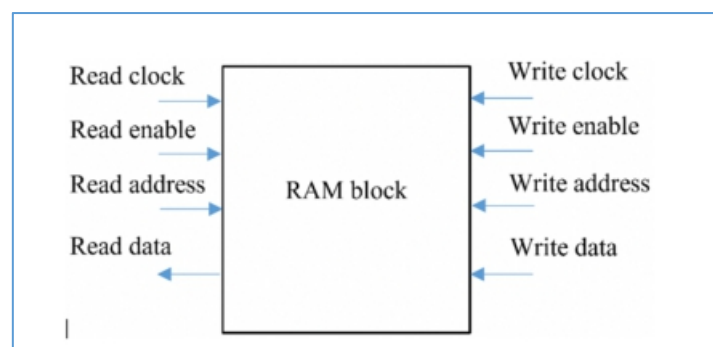
RAM [Địa chỉ][Quy tắc] = 0 nếu địa chỉ \neq T.

Với [địa chỉ]: địa chỉ trong RAM

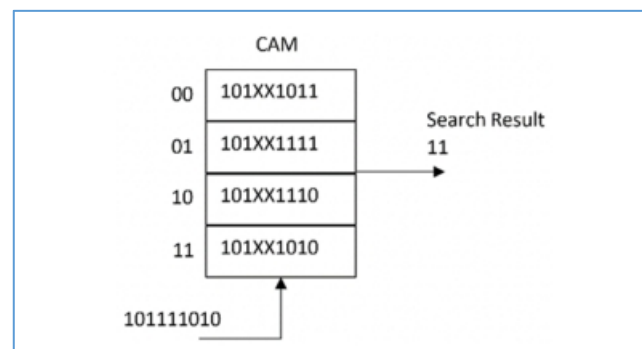
- Nguyên tắc 2: Đọc từ RAM

Đầu ra = RAM [địa chỉ] nếu địa chỉ = K.

Với [Địa chỉ]: địa chỉ ở trong RAM.



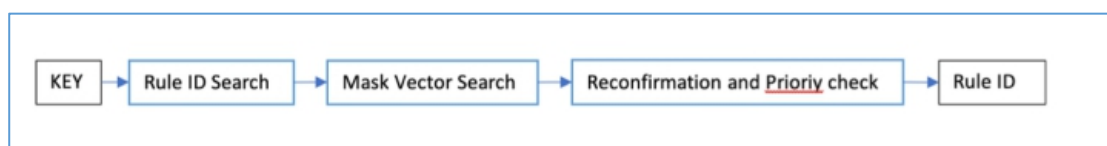
Hình 3-1. Một khối ram đặc trưng



Hình 3-2. Ý tưởng cơ bản của định địa chỉ



Hình 3-3. Key được cắt ra thành nhiều phần và sử dụng để tạo ra match vector



Hình 3-4. Luồng dữ liệu cơ bản của cấu trúc TCAM

Ví dụ : Sử dụng RAM để tìm KEY = 1100. Kích cỡ của RAM được thiết kế dựa trên số lượng rules và số lượng bit của mỗi rule. Với $W = 4$ và $Rule = 4$, ta cần $2^W = 24 = 16$ registers và số lượng bit của mỗi register = số lượng rules = 4. Với những rule ở trong bảng 2 dưới đây ta cần một RAM kích thước $2^4 \times 4 = 16 \times 4$ (16 registers với mỗi register 4 bit).

Rule	Word
0	1001
1	x100
2	01xx
3	xx00

Bảng 3-2. Basic Rule

Address	R0	R1	R2	R3
0000[0]	0	0	0	1
0001[1]	0	0	0	0
0010[2]	0	0	0	0
0011[3]	0	0	0	0
0100[4]	0	1	1	1
0101[5]	0	0	1	0
0110[6]	0	0	1	0

0111[7]	0	0	1	0
1000[8]	0	0	0	1
1001[9]	1	0	0	0
1010[10]	0	0	0	0
1011[11]	0	0	0	0
1100[12]	0	1	0	1
1101[13]	0	0	0	0
1110[14]	0	0	0	0
1111[15]	0	0	0	0

Bảng 3-3. Nội dung khi được ghi trong RAM

Giải thích :

- Với rule 0, word = T0 = 1001. Theo nguyên tắc 1, ta có: RAM[1001][0] = 1, RAM [Address ≠ T0][0] = 0.
- Với rule 1, word = x100 \diamond word = 0100, 1100. Theo nguyên tắc 1, ta có : RAM[0100][1] = 1 và RAM[1100][1] = 1, RAM [Address ≠ T1][1] = 0.
- Với rule 2 và 3, ta có thể điền vào RAM tương tự như trên

Sử dụng nguyên tắc 2 ở trên, kết quả đầu ra sẽ được truy xuất ra nếu [Address] = Key. Với ví dụ ở trên Key = 1100, Output = RAM[1100] = 0101 \rightarrow Chúng ta có kết quả là rule 1 và rule 3 phù hợp với kết quả tìm kiếm. Sử dụng bộ mã hóa ưu tiên – Priority Encoder để đưa ra một rule phù hợp cuối cùng. Tùy thuộc vào thiết kế của bộ mã hóa ưu tiên thì kết quả cuối cùng sẽ khác nhau.

Nhận xét:

- Ưu điểm: Đơn giản và dễ sử dụng.
- Nhược điểm: Cần một bộ nhớ dung lượng lớn để xử lý lượng dữ liệu lớn. Ví dụ với word = 100 bit, hoặc Key = 100 bit, chúng ta sẽ có một RAM chứa 2^{100} thanh ghi \rightarrow Không thể giải quyết được.

3.2.2 Ý tưởng chính.

Để tránh việc sử dụng RAM với dung lượng lớn sử dụng nhiều con RAM nhỏ để xử lý thay vì phải sử dụng một con RAM với dung lượng lớn.

- Phương pháp Depth Division:

Một TCAM có độ sâu N có thể được tách ra thành 2 TCAMs bằng cách chia nhỏ độ sâu $N = N1 + N2$. Ý tưởng hình thành từ việc nếu độ sâu N của TCAM quá lớn và không có một con RAM đơn hoặc TCAM có thể xử lý được.

$$N * WTCAM = (N1+N2)*WTCAM = N1*WTCAM_1 + N2*WTCAM_2 = N1 * 2^{w1}RAM_1 + N2 * 2^{w2}RAM_2$$

- Phương pháp Width Division:

Sử dụng chung ý tưởng với phương pháp Depth Division. Kỹ thuật này chia nhỏ độ rộng của TCAM hay chiều rộng Key tìm kiếm. Phương pháp này giải quyết vấn đề nếu số bit của Word TCAM quá lớn thì không một RAM đơn hay TCAM có thể xử lý được.

$$N*WTCAM = N*(W1+W2)TCAM = N*W1TCAM_1 + N*W2TCAM_2 = N*2^{w1}RAM_1 + N * 2^{w2}RAM_2$$

Sau khi chia chiều rộng của TCAM, kết nối tất cả đầu ra với cổng AND

để tìm vector khớp bit W cuối cùng. Với phương pháp này, chúng ta không chỉ tối ưu hóa bộ nhớ của từng RAM mà còn sử dụng một cách thích hợp một số RAM trong các thiết bị FPGA.

Ví dụ: Để chứng minh, chúng ta sử dụng lại ví dụ trên để chứng minh ý tưởng này. Giả sử chúng ta sử dụng RAM để giải mã Bảng quy tắc với nội dung sau trong bảng với giá trị khóa 1100. Với việc sử dụng một số RAM, chúng ta chia nhỏ từ quy tắc và Key thành 2 phần, mỗi phần 2 bit để xử lý. Với Nguyên tắc 1, chúng ta lần lượt điền vào 2 bit đầu tiên và 2 bit sau của các quy tắc thành 2 bảng như sau

Address	R0	R1	R2	R3
00	0	1	1	1
01	1	0	1	0
10	0	0	1	0
11	0	0	1	0

Bảng 3-4. Nội dung của RAM (Word[1:0])

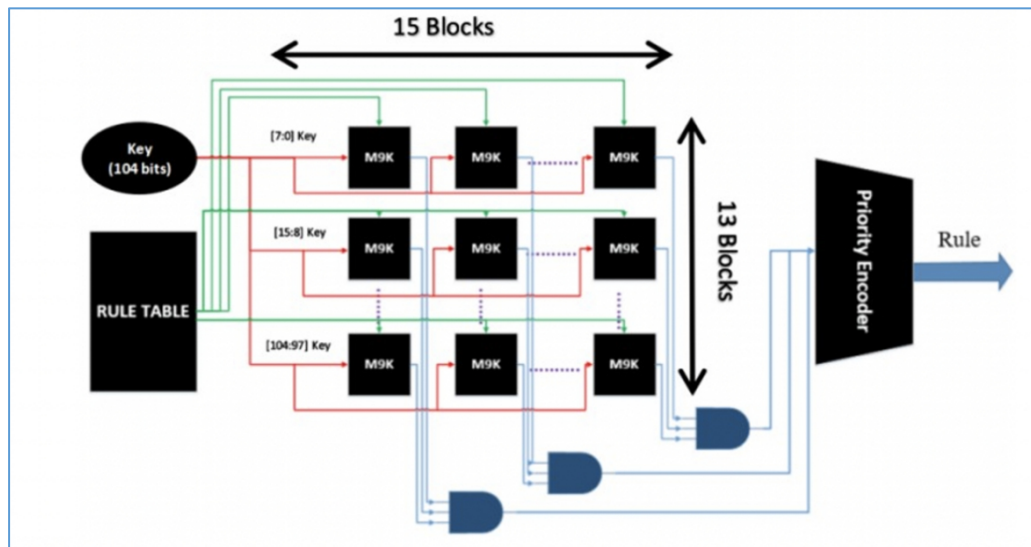
Address	R0	R1	R2	R3
00	0	0	0	1
01	0	1	1	1
10	1	0	0	1
11	0	1	0	1

Bảng 3-5. Nội dung của RAM (Word[3:2])

Với Nguyên tắc 2, chúng ta sẽ đọc 2 thanh ghi $\text{Output1} = \text{RAM}[\text{Key}[3:2]] = \text{RAM}[11]$ và $\text{Output2} = \text{RAM}[\text{Key}[1:0]] = \text{RAM}[00]$ từ các bảng. Sau khi đọc 2 đầu ra từ 2 RAM ở trên, chúng ta sẽ thực hiện phép toán AND để có được kết quả cuối cùng: $\text{OUTPUT} = \text{Output1} \& \text{Output2} = 0101 \& 0111 = 0101$. Có cùng giá trị với ví dụ trên.

Nhận xét:

- Ưu điểm: Sử dụng một lượng nhỏ RAM để giải mã.
- Nhược điểm: cần thêm RAM để sử dụng. Như vậy, sử dụng một số bộ nhớ RAMs nhỏ sẽ hiệu quả và khả thi hơn so với sử dụng RAM lớn.



Hình 3-5. Sơ đồ giải thuật

4. DỰ KIẾN KẾT QUẢ ĐẠT ĐƯỢC

4.1 Kết quả sơ khởi đã đạt được

Đối với những nội dung chưa hoàn thành, sinh viên trình bày kết quả dự kiến sẽ đạt được khi tiến hành thực thi

Mô phỏng tùy định của một khối TCAM 8 bit:

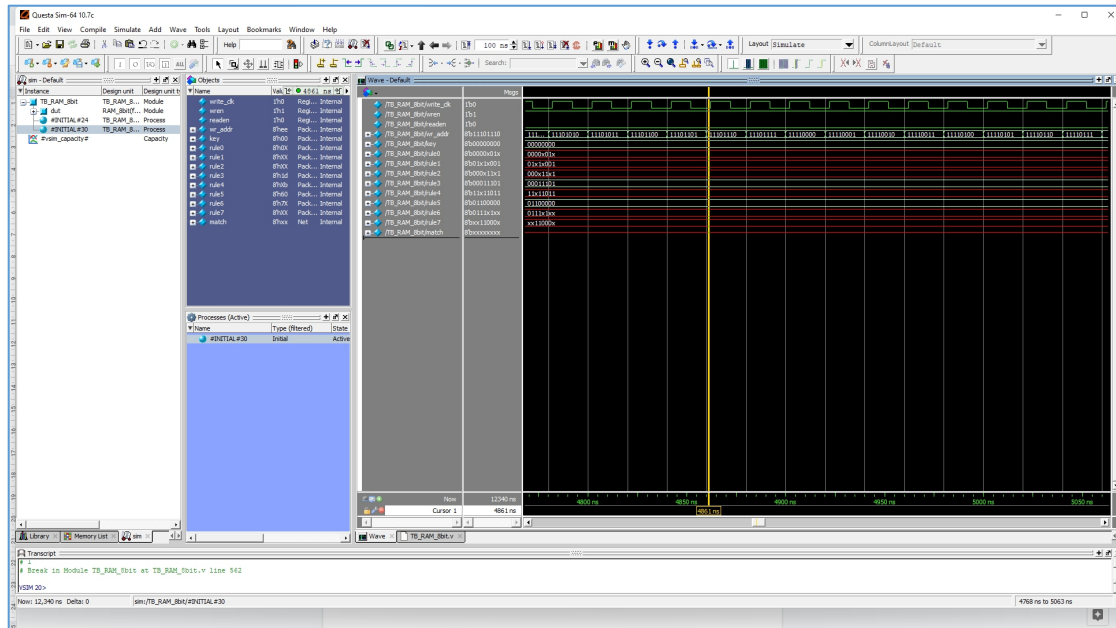
4.1.1. Cấu trúc một TCAM 8 bit

- Ngõ vào:
 - Write_clk, Wren, Readen.
 - Wr_addr, key, rule 0, rule1, rule2, rule3, rule4, rule5, rule6, rule7 có độ dài 8 bit.
- Ngõ ra: match có độ dài 8 bit.

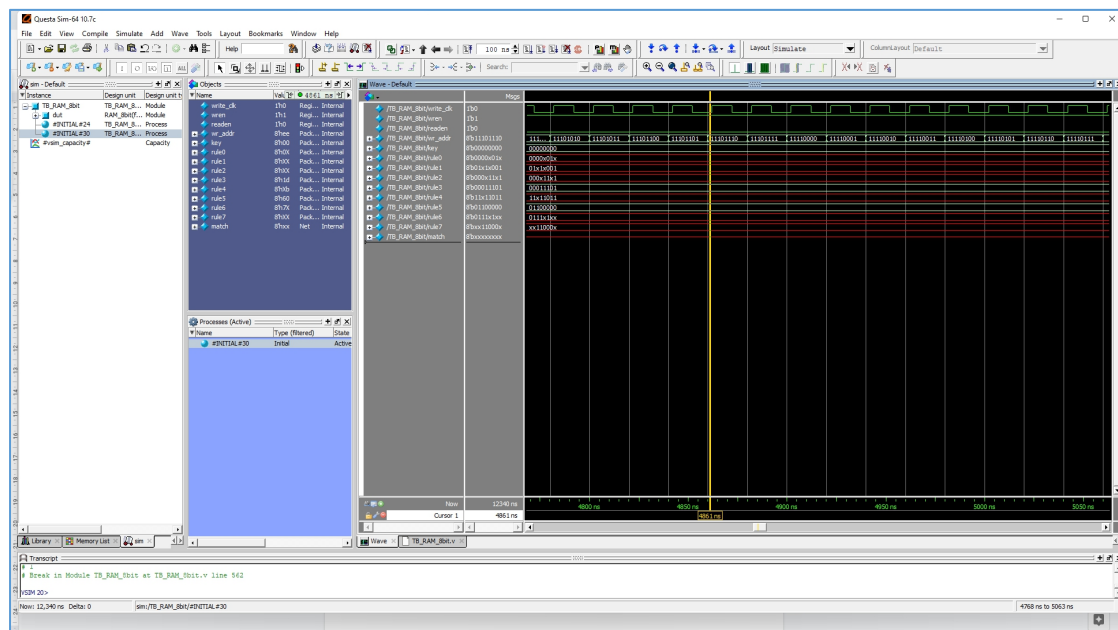
4.1.2. Phương pháp mô phỏng

- Tạo xung clk với chu kỳ 20ns.
- Reset hệ thống để đảm bảo tất cả các giá trị quay về mặc định.
- Cho rule là những số 8 bit bất kỳ trong đó bao gồm cả bit tùy định (x).
- Cho phép wren = 1'b1 để sau mỗi 20ns tăng wr_addr từ 0 đến 255 để so sánh và điền hết vào các địa chỉ.
- Cho phép readen = 1'b1 để sau mỗi 20ns tăng key từ 0 đến 255 để đọc được kết quả trong từng khay địa chỉ và truyền ra ngõ ra match.
- Kết quả của bit LSB cho đến MSB của match chính là những rule hợp lệ mà key được đưa vào TCAM để tìm kiếm (Tương ứng bit có thứ tự bằng mấy bằng 1 thì rule có thứ tự tương ứng sẽ là hợp lệ).

4.1.3. Kết quả mô phỏng



Hình 4-1 Quá trình điền Rule vào RAM.



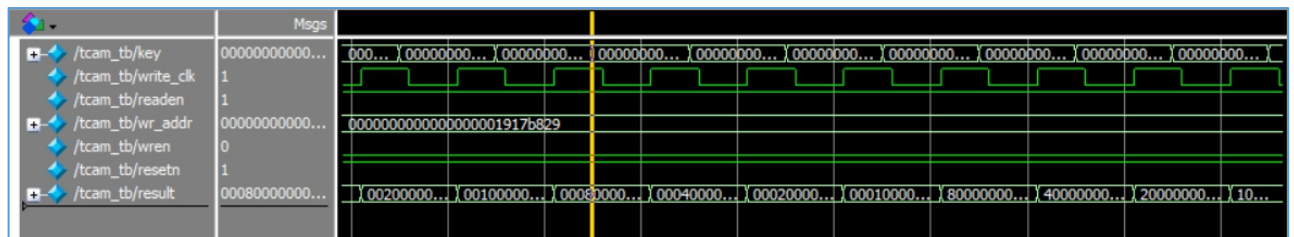
Hình 4-2 Quá trình tăng Key để kiểm tra được những rule nào là hợp lệ với key đã được đưa vào trong TCAM.

	Name	Type	Mode	Port A Depth	Port A Width	Port B
177	Row_of_RAM:Row_of_RAM8]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
178	Row_of_RAM:Row_of_RAM8]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
179	Row_of_RAM:Row_of_RAM8]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
180	Row_of_RAM:Row_of_RAM8]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
181	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
182	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
183	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
184	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
185	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
186	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
187	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
188	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
189	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
190	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
191	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
192	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
193	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
194	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256
195	Row_of_RAM:Row_of_RAM9]...uto_generated ALTSYNCRAM	AUTO	Simple Dual Port	256	8	256

Hình 4-3 Kết quả tổng hợp phần cứng của RAM

Analysis & Synthesis Status	Successful - Mon Jun 28 14:34:36 2021
Quartus Prime Version	16.1.0 Build 196 10/24/2016 SJ Lite Edition
Revision Name	Total_tcam
Top-level Entity Name	Total_tcam
Family	Cyclone V
Logic utilization (in ALMs)	N/A
Total registers	0
Total pins	12,812
Total virtual pins	0
Total block memory bits	399,360
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

Hình 4-4 Kết quả tổng hợp phần cứng trên FPGA



Hình 4-5: Kết quả mô phỏng tra key của TCAM

RULE	KEY	RULE MATCH
RULE 0= 8'b0000_x01x;	8'd0	Không có rule match
RULE 1= 8'b01x1_x001;	8'd1	Không có rule match
RULE 2= 8'b000x_11x1;	8'd2	RULE 0
RULE 3= 8'b0001_1101;	8'd3	RULE 0
RULE 4= 8'b11x1_1011;	8'd4	Không có rule match
RULE 5= 8'b0110_0000;	8'd5	Không có rule match
RULE 6= 8'b0111_x1xx;	8'd6	Không có rule match
RULE 7= 8'bx11_000x;	8'd7	Không có rule match
	8'd8	Không có rule match
	8'd9	Không có rule match
	8'd10	RULE 0
	8'd11	RULE 0
	8'd12	Không có rule match
	8'd13	RULE 2
	8'd14	Không có rule match
	8'd15	RULE 2
	8'd16	Không có rule match
	8'd17	Không có rule match
	8'd18	Không có rule match
	8'd19	Không có rule match
	8'd20	Không có rule match
	8'd21	Không có rule match
	8'd22	Không có rule match
	8'd23	Không có rule match
	8'd24	Không có rule match
	8'd25	Không có rule match
	8'd26	Không có rule match
	8'd27	Không có rule match
	8'd28	Không có rule match
	8'd29	RULE 2, RULE 3

Bảng 3-6. Kết quả mô phỏng tra key được thể hiện cụ thể qua bảng

4.2 Đánh giá:

- Tích cực:

-
- Chương trình hoạt động ổn định. Mô phỏng thực hiện được công việc viết RAM, đọc RAM và tìm kiếm khi có dữ liệu vào.
 - Kết quả output thực hiện mô phỏng khớp với kết quả mong muốn.
 - Mô phỏng cho thấy thiết kế TCAM trên RAM có chức năng có thể thực hiện tìm kiếm dữ liệu vào và cho ra output là Rule dùng để tra cứu dữ liệu có dữ liệu khớp với dữ liệu vào.
 - Hạn chế:
 - Thiết kế còn hạn chế về số Rule. Do chưa viết được khối cập nhật Rule. Nên còn nhiều trường hợp sẽ không có Rule match. Việc tìm kiếm dữ liệu sẽ bị hạn chế.
 - Với các kết quả mô phỏng, có thể thấy thiết kế TCAM có độ rộng bit càng lớn thì cần có bộ nhớ RAM càng lớn. Còn nếu chia độ rộng bit thành nhiều phần thiết kế thì cần nhiều RAM để thực hiện. Nên việc thiết kế TCAM có độ rộng bit càng lớn thì càng phức tạp.

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Trong đề tài em đã thực hiện được những công việc sau:

- Khảo sát phân lý thuyết:
 - Giới thiệu tổng quát về TCAM.
 - Giới thiệu phần mềm Quastus II, ModelSim dùng để lập trình, tổng hợp mạch và mô phỏng bằng ngôn ngữ Verilog.
- Nội dung:
 - Viết giải thuật tổ chức TCAM dựa trên RAM.
 - Viết chương trình thiết kế cho giải thuật.
 - Chạy mô phỏng cho chương trình.

Mặc dù đã rất cố gắng, nhưng do kiến thức và thời gian có hạn chế nên có những phần còn hạn chế như: còn bị giới hạn số Rule, chưa cập nhật được số Rule liên tục. Và đề tài cũng sẽ không tránh khỏi những thiếu sót, nhầm lẫn khác, kính mong quý thầy cô đóng góp ý kiến để đề tài có thể hoàn thiện tốt hơn.

5.2 Hướng phát triển

Để đề tài có thể phát triển hơn trong tương lai cần khắc phục các hạn chế trên. Thực hiện viết khối chương trình cập nhật Rule để có thể tra cứu với toàn bộ bộ nhớ TCAM, việc tra cứu sẽ dễ dàng tìm được dữ liệu khớp hơn. Sau đó thực hiện hoạt động và khảo sát trên kit với công nghệ FPGA để tổ chức trên các khối RAM, hiểu rõ hơn về tổ chức TCAM trên RAM với công nghệ FPGA. Có thể sử dụng thêm bộ nhớ TCAM vào các ứng dụng bộ nhớ Memory, nhằm cải tiến, phát triển thêm Memory cho các phần cứng.

6. KẾ HOẠCH THỰC HIỆN

Thời gian thực hiện đồ án: 4 tháng

Kế hoạch:

Thời gian	Tháng 3	Tháng 4	Tháng 5	Tháng 6
Nội dung	Tìm hiểu đề tài, đọc từ các bài báo quốc tế	Vẽ sơ đồ khối, tìm hiểu cấu trúc các phần	Code + Verify	Hoàn chỉnh báo cáo + chỉnh sửa 1 số lỗi còn lại

Phân chia công việc trong nhóm

Sinh viên mô tả cách thức phân chia công việc đồ án trong nhóm, chi tiết công việc cho mỗi sinh viên, thời hạn, và quy định riêng trong nhóm đồ án, thời gian họp nhóm.

	Nhật	Nhân	Phước	Long
Tìm hiểu đề tài	x	x	x	x
Vẽ sơ đồ khối		x		
Thiết kế các khối:				
Khối RAM	x			x
Khối so sánh rule với key		x	x	
Khối điền rule		x	x	
Khối so sánh		x		x
Khối hàng TCAM	x			
Khối TCAM	x			x
Viết file testbench	x			
Làm báo cáo			x	x
Làm Powerpoint			x	

TÀI LIỆU THAM KHẢO

- [1] Cornell University, “DE1-SoC FPGA memory examples ECE 5760 Cornell University”, <https://people.ece.cornell.edu/>.

- [2] Triet Nguyen, Kiet Ngo, Nguyen Trinh, Bao Bui, Linh Tran, Hoang Trang; “Efficient TCAM design based on dual port SRAM on FPGA”; Indonesian Journal of Electrical Engineering and Computer Science.

