

# آزمایش نهم

گزارش آزمایش نهم

محمدپیام تائبی 400104867

امیرحسین علمدار 400105144

علیرضا سلیمیان 400105036

در این آزمایش قصد داریم TCAM را پیاده سازی کنیم

```
module TCAM
#(
    parameter data_size = 4,
    parameter number_of_address_lines = 3
)
(
    input clk,
    input reset,
    input write,
    input read,
    input [number_of_address_lines - 1 : 0] input_addr,
    input [data_size - 1 : 0] input_data,
    input [data_size - 1 : 0] dont_care_input,
    output reg [(1 << number_of_address_lines) - 1 : 0] match,
    output reg [(1 << number_of_address_lines) - 1 : 0] is_data_valid
);
```

همچنین ماژولی را خواهیم داشت که یک ماژول رم است که خانه های آن علاوه بر 0 و 1 دونت کیر نیز دارند و هنگام رایت مثل رم عادی ولی هنگام رید کردن آنهایی که دونت کیر هستند را نیز تطابق میزند:

ورودی ها:

کلاک ریست رید رایت مثل همه رم ها –

آدرس مشخص شده برای رایت -مقدار ورودی مشخص شده به شکل دو باس (دونت کیر ها را با 1 در dont\_care\_input مشخص میکنیم)

مچ و ولید دیتا را هم برای رید خروجی میدهیم که هر خونه حافظه آیا معتبر است و در کنار آن آیا با داده ورودی مچ میشود یا خیر

```

reg [data_size - 1 : 0] file [(1 << number_of_address_lines) - 1 : 0];
reg [data_size - 1 : 0] dont_care_file[(1 << number_of_address_lines) - 1 : 0];

reg [data_size - 1 : 0] data;
reg [data_size - 1 : 0] dont_care_data;

integer i, j, k , q;

```

برای این مموری مشابه ورودی دیتا دو آرایه نگه میداریم یکی فایل اصلی است و یکی طوری است که اگر دیتای آن 1 باشد یعنی مقدار با معنی دارد و اگر 0 باشد یعنی بی معنی و X است و باید با هم 0 هم 1 میچ شود.

اینترهای تعریف شده هم برای لوپ های بخش بعدی هستند

```

always @(posedge clk or posedge reset)
begin
    if(reset) begin
        is_data_valid = 0;
        match = 0;
        for(q = 0 ; q <= (1 << number_of_address_lines) - 1; q = q + 1)
        begin
            dont_care_file[q] = 0;
            file[q] = 0 ;
        end
    end
    else

```

ابتدا اگر ریست فعال شود کل حافظه را دونت گیر و همه را not valid میکنیم

```

    if(read)
begin
    match = 0;
    for(i = 0 ; i <= (1 << number_of_address_lines) - 1; i = i + 1)
begin
    if(is_data_valid [i] == 1)
begin
        k = 0;
        data = file [i];
        dont_care_data = dont_care_file[i];
        for(j = 0 ; j <= data_size - 1 ; j = j + 1)
begin
            if(!dont_care_data[j]) begin // yani bi mani
                k = k + 1;
            end
            else begin
                if(data[j] == input_data[j]) begin
                    k = k + 1;
                end
            end
        end
    end
    if(k == data_size )
        match[i] = 1;
    end
end
end

```

برای رید مشابه بالا ابتدا اگر ولید باشد و دونت کیر 0 باشد یعنی با معنی باشد حالا تازه با دیتای اصلی چک میشود و اگر یکی باشد مچ مورد نظر را 1 میکند

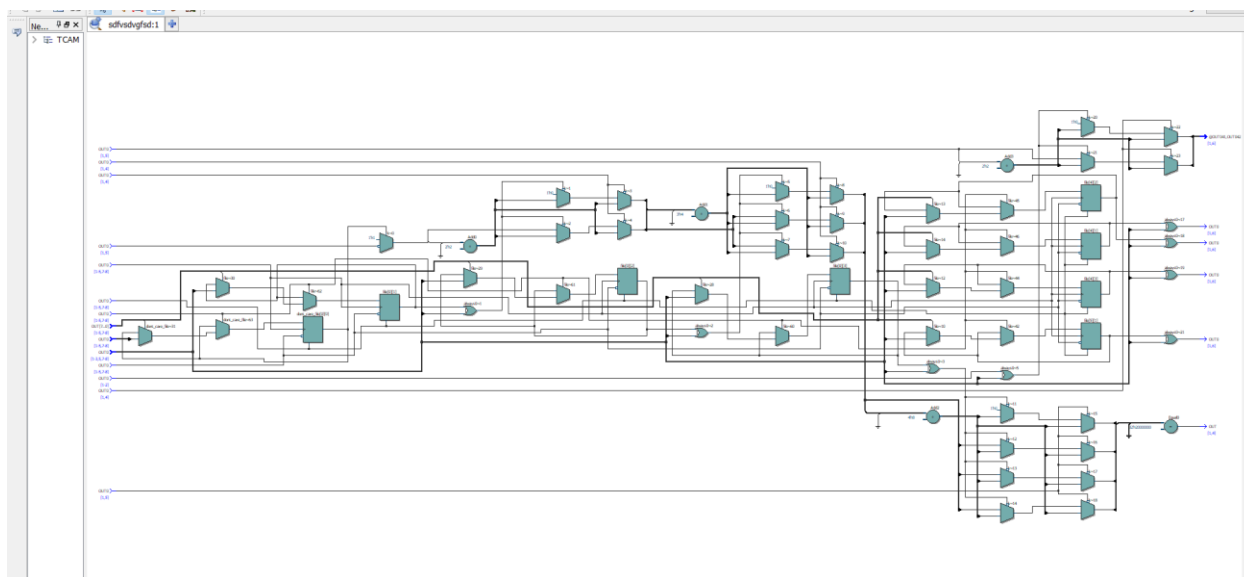
```

    else begin
        if(write) begin
            match = 0;
            file [input_addr] = input_data;
            dont_care_file [input_addr] = dont_care_input;
            is_data_valid [input_addr] = 1;
        end
    end
end
end

```

در آخر برای رایت نیز مثل رم عادی رایت میکنیم صرفا جای یک دیتا و یک آرایه دو دیتا و دو آرایه داریم.

چالش هایی که حین پیاپی سازی داشتیم چند باگ موقع رید بود به همین خاطر به ماژول ولید بودن دیتا هم خروجی دادم وگرنه نیازی نبود و جهت دیباگ گذاشته شده است.



عکس مدار سنتز شده :