



به نام خدا

طراحی کامپیوتری سیستم های دیجیتال

پاییز ۱۳۹۹

تمرین ۵

موعد تحویل:

۱- کد زیر را در نظر بگیرید

```
module Adder4to1(  
    input clk, reset,  
    input [9:0] A1, A2, A3, A4,  
    output reg [11:0] Sum  
);  
reg [9:0] A1L, A2L, A3L, A4L;  
reg [11:0] P1;  
always@ (posedge clk) begin  
    if (reset) begin  
        A1L <= 0;  
        A2L <= 0;  
        A3L <= 0;  
        A4L <= 0;  
        P1 <= 0;  
        Sum <= 0;  
    end  
    else begin  
        A1L <= A1;  
        A2L <= A2;  
        A3L <= A3;  
        A4L <= A4;  
        P1 <= A1L+A2L+A3L+A4L;  
        Sum <= P1;  
    end  
end  
end
```

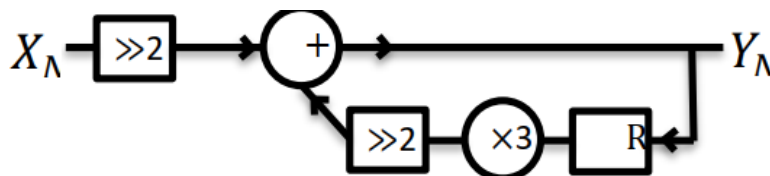
endmodule

با استفاده از تکنیک جابجایی رجیسترهای درونی (retiming) و بدون در نظر گرفتن قابلیت ابزار سنتز برای استفاده خودکار از این تکنیک، کد را به شکلی بازنویسی کنید که حداکثر فرکانس ماژول افزایش یابد. حداکثر فرکانس را در دو حالت روی Spartan3 XC3S400 FPGA بررسی کنید.

۲- فرمول بازگشتی زیر را در نظر بگیرید

$$Y_N = \frac{1}{4}X_N + \frac{3}{4}Y_{N-1} \quad N \geq 1, \quad Y_0 = 0$$

بلوک دیاگرام زیر یک شکل پیاده سازی این فرمول را نشان میدهد (R نماد رجیستر و بنابرین معادل یک واحد زمانی تاخیر، و >> نماد شیفت به راست است):



این شکل توسط کد زیر پیاده سازی شده است:

```
module RecursiveFormula(
    input clk, reset,
    input [5:0] Xn,
    output [9:0] Yn
);
    reg [9:0] Ynm1 = 0; //initial condition
    always @(posedge clk) begin
        if(reset) begin
            Ynm1 <= 0; //set to initial condition
        end
        else begin
            Ynm1 <= Yn;
        end
    end
end
```

```

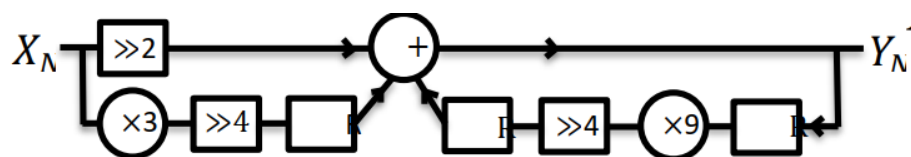
end
assign Yn = (Xn>>2)+((Ynm1*3)>>2);
endmodule

```

با جانشانی، Y_{N-1} فرمول را بر اساس Y_{N-2} بازنویسی میکنیم:

$$Y_N = \frac{1}{4}X_N + \frac{3}{16}X_{N-1} + \frac{9}{16}Y_{N-2} \quad N \geq 1, \quad X_0 = Y_0 = Y_{-1} = 0$$

و بلوک دیاگرام زیر را برای آن در نظر میگیریم.



کد این بلوک دیاگرام پیاده سازی کنید (توجه داشته باشید که ورودی ها و خروجی نسبت به ماژول قبل تغییری ندارند).

به منظور مقایسه حداکثر فرکانس این دو شکل پیاده سازی، یک ماژول سطح بالاتر بنویسید که ورودی ماژول RecursiveFormula را پس از ذخیره در یک رجیستر (از خروجی رجیستر) به آن تحویل دهد و خروجی ماژول را هم در یک رجیستر ذخیره کند (توجه داشته باشید علت این کار آن است که حداکثر فرکانس بر اساس منابع combinational میان دو رجیستر محاسبه میشود)، سپس حداکثر فرکانس دو شکل پیاده سازی را روی Spartan3 XC3S400 FPGA بررسی کنید.

Header (8'hCC)
Block Length (N)
D ₁
D ₂
D ₃
...
D _N
Parity Byte
Footer (8'hFF)

۳- در یک سیستم انتقال داده، مطابق شکل روبرو، هر بسته داده ارسال شده، از یک بایت Header ثابت (8'hCC)، یک بایت طول بسته (N)، تعداد N بایت داده، یک بایت parity که حاصل XOR داده های ارسالی است و یک بایت Footer ثابت (8'hFF) تشکیل میشود. پروسه تولید داده با آمدن پالس Start آغاز میشود (D₁, D₂, ..., D_N) و تا اتمام بسته و یا آمدن سیگنال Reset ادامه مییابد. میخواهیم یک ماژول تحت عنوان DataPacker را به زبان Verilog طراحی نماییم که با دریافت داده های ورودی بصورت سنکرون با سیگنال clock این بسته داده ها را تولید کند.

ورودی های ماژول عبارتند از:

Clock: کلیه عملیات با آن سنکرون میباشند

Reset: بصورت آسنکرون باعث reset شدن ماژول میشود

Start: پالس سنکرون با Clock که با یک شدن آن تولید بسته دادهها آغاز میشود

DataIn: رشته دادههای ورودی سنکرون با Clock

N: عدد 8بیتی معرف تعداد نمونه های هر بسته

خروجیهای ماژول عبارتند از:

DataOut: رشته 8بیتی خروجی سنکرون با Clock

تذکر 1: ماژول نوشته شده باید قابل سنتز بر روی سخت افزار باشد.

تذکر 2: نوشتن یک Test Bench مناسب جهت تست کد نوشته شده الزامی است.