```
        if (Valid) begin
            P1<=In;
            P2<=P1;
            P3<= if((P1+P3)>TH)
                    P3<=100;
                 else
                    P3<=0;
            Out<=P3;
            ValidOut<=1;
            end
        else
            ValidOut<=0;
endmodule
```

تمرین: سؤال ۵ تمرین ۲

تمرین خطی: `module Car(Out, In, Clk, ResetL, Start);`

```
    output [3:0] Out;
    input In, Clk, ResetL, Start;
    reg [3:0] Out;
    reg r;
    always @ (posedge Clk or negedge ResetL)
        if (!ResetL) begin
            r<=0;
            Out<=0;
            end
        else begin
```

PAYCO

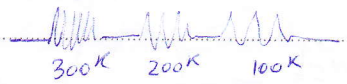
Subject: ۳۹

Year : Month : Day : ( )

```
if (Start) begin
    r<=1;
    Out<=0;
    end
else begin
    if (!In && r)
        Out<=Out+1;
    r<=In;
    end
end
endmodule
```

تمرین: سوال ۳ تمرین ۲

فرستنده، گیرنده:



1 0 0 1

100 KHz → 10000 ns

cycle: 50 ns → $\frac{10000}{50} = 200$ cycle

← اگر 200 cycle یا کمتر تغییری نکرد ← ورودی صحیح است.

← اگر در یک 200 cycle تغییر در ورودی 100K، اگر در یک 100 cycle تغییر در ورودی 200K و اگر در یک 66 cycle تغییر در ورودی 300K ثبت

← جواب بدی و نسبت

PAYCO

Subject:

Year : Month : Day : ( )

نکته: مبنای ۱۶ (مبنای عدد مورد نظر)

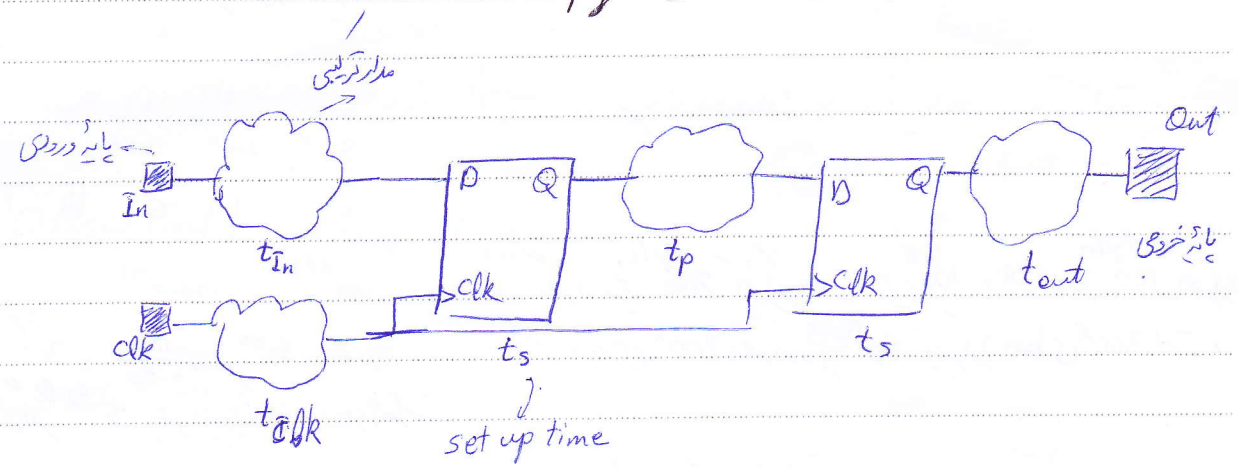
در نرم افزار coregen برای مقدار اولیه دادن به حافظه‌ها باید از یک فایل با پسوند .coe که با syntax مربوط نوشته می‌شود استفاده کرد

```
MEMORY_INITIALIZATION_RADIX = 16;
MEMORY_INITIALIZATION_VECTOR=
ffffffab,
12345678;
:
11112888;
```

سپس فایل core تولید شده با پسوند .vco را داخل ماژول کپی می‌کنیم.

فایل دیتری که برای Implement کردن استفاده می‌شود پسوند .ngc دارد. (به جای .edf)

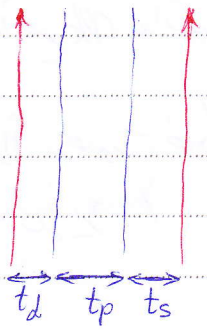
offset_Out , offset_In : fpga



اگر تأخیر ورودی تا خروجی فلیپ فلاپ t_d باشد:

Subject: ۴۱

Year : Month : Day : ( )



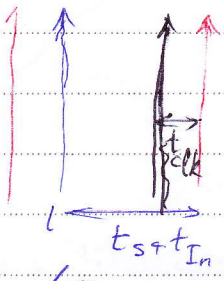
$$\min period = t_d + t_p + t_s$$

مینیمم پریود برابر حداقل زمانی است که داده از خروجی فلیپ فلاپ اول به ورودی فلیپ فلاپ دوم برسد.

Offset In: fpga

حداقل و حداکثر مقدار وقت که قبل از آنکه لبه بالارونده ساعت وارد fpga شود لازم است داده روی ورودی In، Stable شده باشد؟

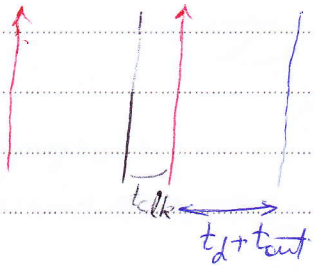
$$Offset\ In = t_s + t_{In} - t_{clk}$$



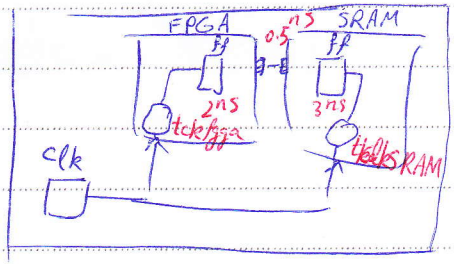
Offset Out:

حداقل و حداکثر مقدار وقت بعد از آنکه لبه بالارونده ساعت وارد fpga شود داده روی خروجی Out، معتبر می‌باشد.

$$Offset\ Out = t_d + t_{out} + t_{clk}$$



مثال:



$$\min Period = 2 + 0.5 + 3 + t_{clk_{fpga}} - t_{clk_{SRAM}}$$

PAYCO

Subject: ٤٤

Year : Month : Day : ( )

برای اینکه تأخیر clk نداشته باشیم

(این عمل معمولاً به صورت خودکار در fpga صورت می‌پذیرد.)

CPU در خانوادهٔ Xilinx :

Xilinx
- Pico Blaze (8bit - Soft Core)
- Micro Blaze (32 bit - Soft Core)
- Power PC (64 bit - Hard Core) — Virtex II pro / Virtex 4 FX / Virtex 5,6 FX

CPU های Micro Blaze ، Power PC احتیاج به یک محیط نرم افزاری به نام EDK (Embeded Development Kit) دارند.

PAYCO