دانشگاه اصفهان – دانشکده مهندسی کامپیوتر

آزمایشگاه معماری کامپیوتر – شعبه ۰۸

استاد درس: دکتر مهران رضایی

کنترل کننده چراغ راهنمایی و رانندگی

اعضای گروه:

متین اعظمی ۴۰۰۳۶۲۳۰۰۳

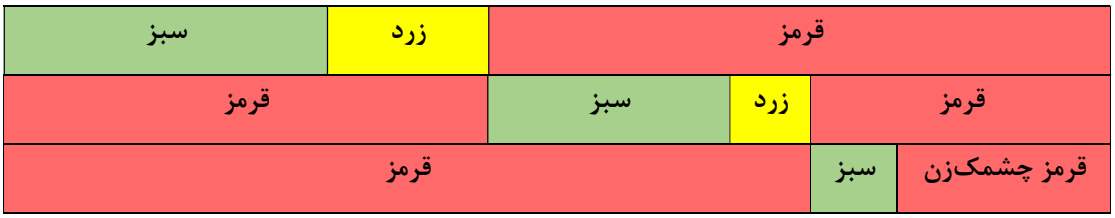
شیدا عابدپور ۴۰۰۳۶۲۳۰۲۵

پاییز ۱۴۰۲

هدف: طراحی یک کنترل کننده چراغ راهنمایی و رانندگی برای چهارراهی با یک خیابان اصلی، یک خیابان فرعی و محل عبور پیاده.

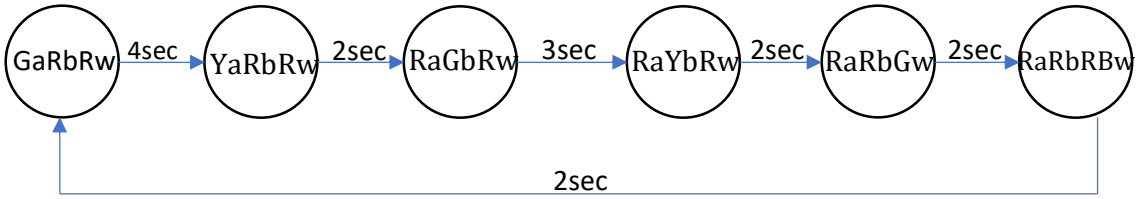
دیاگرام حالت:

ترکیب عملیاتی این چراغ‌ها به صورت زیر است: (به ترتیب خیابان اصلی، خیابان فرعی، محل عبور پیاده)



با توجه به ترتیب تغییرات چراغ‌ها، دیاگرام حالت این طراحی به این صورت است:

در صورتی که rst برابر با ۱ باشد:



در صورتی که rst برابر صفر باشد:

تمامی چراغ‌های قرمز به صورت چشمک‌زن با فرکانس ۱ هرتز روشن و خاموش می‌شوند.

پیاده‌سازی به صورت کد VHDL:

با توجه به دیاگرام حالت، در صورتی که rst برابر صفر باشد:

با هربار کلاک خوردن(هر کلاک ۱ ثانیه طول می‌کشد.) تعداد ثانیه‌های طی‌شده را با استفاده از timer ذخیره می‌کنیم و زمانی که زمان تغییر یک state فرا رسد، به state بعدی می‌رود.

```
if(rst = '0') then
        cur_state <= BLINK;
        timer <= "0000";

elsif(rst = '1') then

         case cur_state is
        when GaRbRw =>
                if(timer = "0100") then
                        cur_state <= YaRbRw;
                end if;
                timer <= std_logic_vector(unsigned(timer) + 1);
        when YaRbRw =>
                if(timer = "0110") then
                        cur_state <= RaGbRw;
                end if;
                timer <= std_logic_vector(unsigned(timer) + 1);
        when RaGbRw =>
                if(timer = "1001") then
                        cur_state <= RaYbRw;
                end if;
                timer <= std_logic_vector(unsigned(timer) + 1);
        when RaYbRw =>
                if(timer = "1010") then
                        cur_state <= RaRbGw;
                end if;
                timer <= std_logic_vector(unsigned(timer) + 1);
        when RaRbGw =>
                if(timer = "1100") then
                        cur_state <= RaRbRBw;
                end if;
                timer <= std_logic_vector(unsigned(timer) + 1);
        when RaRbRBw =>
                if(timer = "1110") then
                        timer <= "0000";
                        cur_state <= GaRbRw;
                else
                        cur_state <= RaRbRBw;
                        timer <= std_logic_vector(unsigned(timer) + 1);
                end if;
        when BLINK =>
                cur_state <= GaRbRw;
         end case;

end if;
```

حال زمانی که حالت ماشین مشخص شود با توجه به آن خروجی را تعیین میکنیم:

خیابان اصلی با main_street، خیابان فرعی با sub_street و محل عبور پیاده به صورت crosswalk مشخص شده‌اند. به این صورت که در main_street و sub_street بیت صفر نشان دهنده وضعیت چراغ سبز، بیت ۱ چراغ زرد و بیت ۲ چراغ قرمز است. در crosswalk بیت صفر چراغ سبز و بیت ۱ چراغ قرمز را نشان می‌دهد.

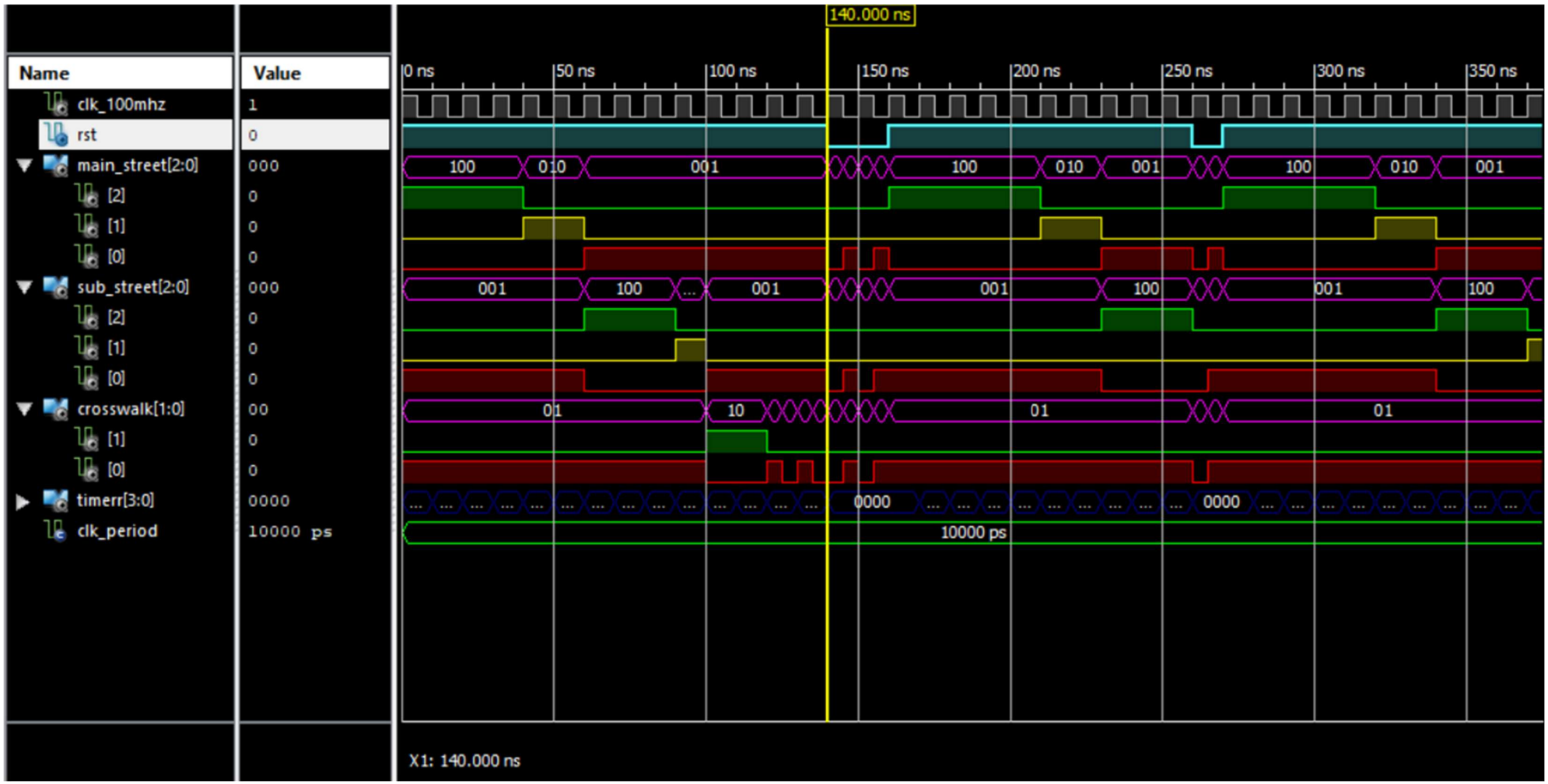
```
process(cur_state, clk_100Mhz) begin

                case cur_state is
        when GaRbRw =>
                main_street <= "100";
                sub_street <= "001";
                crosswalk <= "01";
        when YaRbRw =>
                main_street <= "010";
                sub_street <= "001";
                crosswalk <= "01";
        when RaGbRw =>
                main_street <= "001";
                sub_street <= "100";
                crosswalk <= "01";
        when RaYbRw =>
                main_street <= "001";
                sub_street <= "010";
                crosswalk <= "01";
        when RaRbGw =>
                main_street <= "001";
                sub_street <= "001";
                crosswalk <= "10";
        when RaRbRBw =>
                main_street <= "001";
                sub_street <= "001";
                if(clk_100MHz = '1') then
                        crosswalk <= "01";
                else
                        crosswalk <= "00";
                end if;
        when BLINK =>
                if(clk_100MHz = '0') then
                        main_street <= "001";
                        sub_street <= "001";
                        crosswalk <= "01";
                else
                        main_street <= "000";
                        sub_street <= "000";
                        crosswalk <= "00";
                end if;

        end case;

end process;
```

در حالت چشمک‌زن برای اینکه با فرکانس یک هرتز چشمک بزند، باید در هر پالس یک بار روشن و یک بار خاموش شود. به این منظور در هر کلاک در لبه بالارونده همه چراغ‌های قرمز روشن می‌شوند و در لبه پایین رونده همگی خاموش.

تست بنچ:

به منظور بررسی صحت عمکرد کد، تست بنچ به صورت زیر پیاده سازی شد.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity traffic_signal_light_tb is
end traffic_signal_light_tb;

architecture behavioral of traffic_signal_light_tb is
    signal clk_100MHz: std_logic := '0';
    signal rst: std_logic := '0';
    signal main_street: std_logic_vector(2 downto 0);
    signal sub_street: std_logic_vector(2 downto 0);
    signal crosswalk: std_logic_vector(1 downto 0);
    signal timerr: std_logic_vector(3 downto 0);

    constant clk_period: time := 10 ns;

begin
    dut: entity work.traffic_signal_light
        port map (
            clk_100MHz => clk_100MHz,
            rst => rst,
            main_street => main_street,
            sub_street => sub_street,
            crosswalk => crosswalk,
            timerr => timerr
        );

    -- Clock generation process
    clk_process: process
    begin
        while now < 1000 ns loop
            clk_100MHz <= '1';
            wait for clk_period / 2;
            clk_100MHz <= '0';
            wait for clk_period / 2;
        end loop;
        wait;
    end process;

    -- Reset process
    reset_process: process
    begin
        rst <= '1';
        wait for 140 ns;
        rst <= '0';
        wait for 20 ns;
        rst <= '1';
        wait for 100 ns;
        rst <= '0';
        wait for 10 ns;
        rst <= '1';
        wait;
    end process;


end behavioral;
```


140.000 ns
Name
Value
0 ns
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
350 ns
clk_100mhz
1
rst
0
main_street[2:0]
000
100
010
001
[2]
0
[1]
0
[0]
0
sub_street[2:0]
000
001
100
[2]
0
[1]
0
[0]
0
crosswalk[1:0]
00
01
10
[1]
0
[0]
0
timerr[3:0]
0000
0000
clk_period
10000 ps
10000 ps
X1: 140.000 ns

UCF flie:

```
NET "clk_100MHz"      LOC = P50;
NET "rst"             LOC = p133;
NET "main_street[2]"  LOC = P120;
NET "main_street[1]"  LOC = P121;
NET "main_street[0]"  LOC = P123;
NET "sub_street[2]"   LOC = P124;
NET "sub_street[1]"   LOC = P126;
NET "sub_street[0]"   LOC = P127;
NET "crosswalk[1]"    LOC = P131;
NET "crosswalk[0]"    LOC = P132;
```

synthesis report

خروجی نشان‌دهنده‌ی گزارش فرآیند سنتز (Synthesis) با ابزار Xilinx ISE (Integrated Software Environment)است که برای توسعه FPGA (Field-Programmable Gate Array) استفاده می‌شود. در زیر چند نکته از این گزارش آورده شده است:

۱. **اطلاعات مربوط به:Xilinx ISE**

- نسخه: ۱۴.۷
- حق تکثیر: از ۱۹۹۵ تا ۲۰۱۳
- مدت زمان واقعی به اتمام Xst: 0.00 ثانیه
- مدت زمان پردازنده به اتمام Xst: 0.06 ثانیه
- حجم کل حافظه استفاده شده: ۴۴۸۷۴۰۴ کیلوبایت

۲. **خلاصه سنتز:**

- انجام سنتز برای طراحی"traffic_signal_light"
- ورودی‌ها و خروجی‌ها: استفاده از فایل ورودی "traffic_signal_light.prj" و خروجی با فرمت NGCبا نام"traffic_signal_light"
- دستگاه هدفxc6slx9-3-tqg144 :

۳. **تنظیمات سنتز:**

- هدف بهینه‌سازی: سرعت
- تلاش بهینه‌سازی: ۱
- استفاده از تقریباً ۱٪ از منابعFPGA
- استفاده از ۲۸ رجیستر و ۱۰۸ LUT در تسخیر منطقFPGA
- استفاده از ۱۰) IO ورودی/خروجی(
- استفاده از ۱ بوفر ساعتBUFG

۴. **استفاده از منابع:**

- تعداد کل رجیسترها: ۲۸ از ۱۱۴۴۰) ٪(
- تعداد کلLUT ها: ۱۰۸ از ۱۵۷۲۰) ٪(

۵. **اطلاعات زمان‌بندی:**

- کلاک اصلیclk_100MHz :

- حداقل دوره: ۵.۴۸۴ نانوثانیه (فرکانس حداکثر: ۱۸۲.۳۵۹ مگاهرتز)
- حداکثر تاخیر مسیر ترکیبی: ۵.۴۵۶ نانوثانیه

۶. **استفاده از DSP و منابع دیگر:**

- هیچ واحد DSP مشخص نشده است.
- استفاده از ۰٪ ازBUFG/BUFGCTRLs

۷. **واحدهای شناسایی‌شده:**

- 2افزوده‌کننده/کم‌کننده با ۱ عدد ۲۷ بیتی و ۱ عدد ۴ بیتی
- 1مقایسه‌گر ۲۷ بیتی بزرگتر
- 1 FSMبا ۷ حالت و ۲۰ گذار

۸. **هشدارها و اطلاعات:**

- 8هشدار وجود دارد، از جمله یک هشدار درباره‌ی حساسیت یک سیگنال در یک فرآیند.

۹. **استفاده از حافظه:**

- استفاده از حداکثر ۴۴۸۷۴۰۴ کیلوبایت حافظه

۱۰. **گزارش تاخیر:**

- تحلیل تاخیر برای مسیرهای مختلف نشان داده شده است.

place and route report

اطلاعاتی که در این خروجی "Xilinx ISE "par نمایش داده شده‌اند، مربوط به فرآیند Place and Route (قرار دادن و مسیریابی) در محیط توسعه FPGA است. در اینجا چند نکته کلیدی آورده شده‌اند:

۱. **اطلاعات ابزار:**
   - نسخه ابزار: ۱۴.۷
   - حق تکثیر: کپی‌رایت از سال ۱۹۹۵ تا ۲۰۱۳ شرکتXilinx
   - پلتفرم: دسکتاپ با نام میزبان"DESKTOP-LO6OT5I"
   - تاریخ و زمان: ۱۷ دسامبر ۲۰۲۳، ساعت ۱۶:۳۸:۱۱

۲. **اطلاعات طراحی:**
   - نام طراحیtraffic_signal_light :
   - ورژنNCD: 3.2
   - دستگاهxc6slx9 :
   - پکیجtqg144 :
   - گرید سرعت: -۳

۳. **شروع:**
   - دما: ۸۵.۰۰۰ سلسیوس
   - ولتاژ: ۱.۱۴۰ ولت

۴. **تنظیمات:"par"**
   - گزینه‌هاoff -mt high -ol ise -intstyle -w :
   - فایل دستگاه: '۶'slx9.nph

۵. **خلاصه استفاده از منابع:**
   - استفاده از ۱٪ از ثبات‌ها و ۱٪ ازLUT ها (بلوک‌های منطقی) در حالتUtilization
   - استفاده از ۹٪ از IOBs (بلوک‌های ورودی/خروجی) در حالتUtilization

۶. **استفاده از ویژگی‌های خاص:**
   - درصد استفاده از منابع مختلف FPGA اعلام شده است.

۷. **سطوح تلاش:**
   - سطح کلی تلاشHigh :

- سطح تلاش مسیریابHigh :

۸. **مراحل مسیریابی:**

- ابزار از چند مرحله (فاز ۱ تا فاز ۱۰) برای کامل کردن مراحل مسیریابی استفاده کرده است.

۹. **تجزیه و تحلیل زمان:**

- امتیاز زمان: ۰Setup: 0) ، (Hold: 0
- همه‌ی محدودیت‌ها برآورده شده‌اند.

۱۰. **گزارش پد:**

- همه‌ی سیگنال‌ها کاملاً مسیریابی شده‌اند.

۱۱. **اتمام:"par"**

- زمان واقعی کل: ۲ ثانیه
- زمان پردازنده کل: ۲ ثانیه
- حداکثر استفاده از حافظه: ۴۴۵۸ مگابایت

۱۲. **اطلاعات اضافی:**

- گزارش اعلام کرده است که در حالت "Performance Evaluation Mode" اجرا شده است چرا که هیچ محدودیت زمان‌بندی کاربری یافت نشده یا گزینه برای نادیده گرفتن محدودیت‌های زمان‌بندی فعال شده است.

۱۳. **هشدارها/اطلاعات:**

- دو پیام اطلاعاتی گزارش شده‌اند و هیچ خطایی یا هشداری وجود ندارد.

static timing report

در گزارش زمانیFPGA ، تاخیرهای مختلف در مسیرهای مختلف مدار ذکر شده‌اند. با توجه به نوع گزارش و اطلاعات موجود، می‌توان تاخیرهای مربوط به مسیرهای مختلف را تشخیص داد. در ادامه، مشخصات تاخیرها بر اساس گروه‌های خواسته شده آورده شده‌اند:

الف .**مسیر ورودی به رجیسترها:**

- تاخیر از منبع (ورودی) به رجیستر:
  - نام منبع**rst** :
  - تاخیر حداکثر تنظیم به افت سرعت: ۲.۵۹۳ns (SLOW)
  - تاخیر حداکثر نگهداری به افت سرعت: ۰.۰۵۱ns (SLOW)
  - فرایندSLOW :
- مسیر **rst** :به **clk_100MHz_IBUF_BUFG**

ب .**مسیر رجیسترها به درگاه‌های خروجی:**

- تاخیر از رجیستر به مقصد:(PAD)
  - مقصد**crosswalk<0>**, **main_street<0>**, **sub_street<0>** :
  - تاخیر حداکثر clk به PAD: 4.138ns (FAST) تا ۹.۴۵۹ns (SLOW)
  - تاخیر حداقل clk به PAD: 4.138ns (FAST) تا ۹.۴۵۹ns (SLOW)
  - فرایند FAST :تاSLOW

ج .**مسیر بین رجیسترها:**

- تاخیر بین رجیسترها:
  - مسیر **clk_100MHz_IBUF_BUFG** :به **clk_100MHz_IBUF_BUFG**
  - تاخیرRise: 4.622ns

د .**مسیر درگاه‌های ورودی به درگاه‌های خروجی:**

- تاخیر از PAD به:PAD
  - مبدأ**clk_100MHz** :
  - مقصد**crosswalk<0>**, **main_street<0>**, **sub_street<0>** :
  - تاخیر: ۹.۹۶۹ns (crosswalk<0>), 10.344ns (main_street<0>), 10.332ns (sub_street<0>)

مسیر بحرانی (**Critical Path**) در طراحی مدار FPGA، مسیری است که تاخیر کل مدار بر اساس آن محاسبه می‌شود. این مسیر بیشترین تاخیر را داراست و هر تغییر در آن تاخیر کل مدار را تحت تأثیر قرار می‌دهد. برای تشخیص مسیر بحرانی از اطلاعات گزارش زمانی که ارائه داده‌اید، می‌توانید به تاخیرهای مربوط به مسیرهای مختلف توجه کنید.

با توجه به اطلاعات گزارش زمانی شما، می‌توان نقاط زیر را به عنوان مسیر بحرانی شناسایی کرد:

مسیر ورودی به رجیسترها (الف):

مسیر: rst به clk_100MHz_IBUF_BUFG

تاخیر از منبع به رجیستر: ۲.۵۹۳(SLOW) ns

تاخیر از رجیستر به clk_100MHz_IBUF_BUFG: 0.051ns (SLOW)

مسیر رجیسترها به درگاه‌های خروجی (ب):

مسیرهای <crosswalk<0>, main_street<0>, sub_street<0 به PAD

تاخیر از رجیستر به PAD: 4.138ns (FAST) تا ۹.۴۵۹(SLOW) ns

مسیر بین رجیسترها (ج):

مسیر: clk_100MHz_IBUF_BUFG به clk_100MHz_IBUF_BUFG

تاخیر بین رجیسترها: ۴.۶۲۲ns

مسیر درگاه‌های ورودی به درگاه‌های خروجی (د):

مسیرهای از clk_100MHz به <crosswalk<0>, main_street<0>, sub_street<0

تاخیر از PAD به PAD: 9.969ns (crosswalk<0>), 10.344ns (main_street<0>), 10.332ns (sub_street<0>)

با توجه به مقادیر تاخیرها، مسیری که تاخیر بیشتری دارد (حداقل تاخیر از منبع به مقصد) می‌تواند مسیر بحرانی باشد. در این مورد، مسیر rst به clk_100MHz_IBUF_BUFG به نظر می‌رسد که تاخیر حداکثری دارد و می‌تواند مسیر بحرانی باشد.