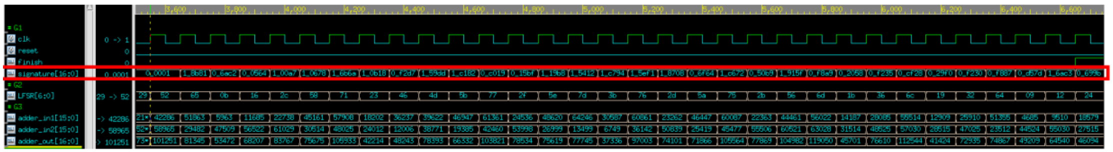
Final project

COSR 111501502 劉俊廷

Introduction:

Insert BIST circuit into your adder, by using a 7-stage Linear Feedback Shift-Register (LFSR), as the automatic pattern generator of 127 pseudo random patterns, and a Multi-Input Shift Register (MISR) as the signature analyzer. Find its golden signature when the circuit is fault free. Also, Set up the simulation environment to verify if your adder can operate at the speed of 1GHz.

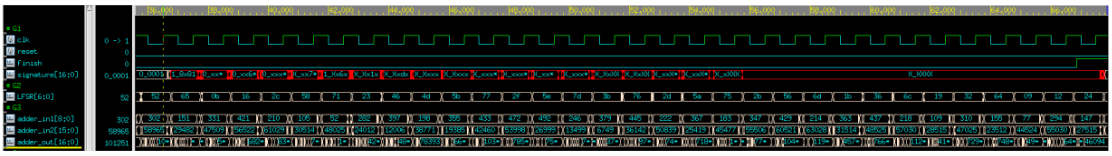
1.The golden signature of fault free circuit is “0699b”.

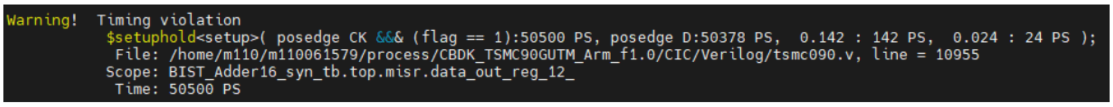


Initial	Pattern0	Pattern1	Pattern2	Pattern3	Pattern4	Pattern5	Pattern6
00001	18b81	06ac2	00564	100a7	10678	16b6a	10b18
Pattern7	Pattern8	Pattern9	Pattern10	Pattern11	Pattern12	Pattern13	Pattern14
0f2d7	159dd	1c182	0c019	015bf	119b8	15412	1c794
Pattern15	Pattern16	Pattern17	Pattern18	Pattern19	Pattern20	Pattern21	Pattern22
15ef1	18708	06f64	1c672	050b9	1915f	0f8a9	02058
Pattern23	Pattern24	Pattern25	Pattern26	Pattern27	Pattern28	Pattern29	Pattern30
0f235	0cf28	029f0	0f230	0f887	0d57d	16ac3	0699b

2. 1GHz Verification

In Design Complier, I set the clock period to 1ns. In this condition, the signature is “xxxxx”, but the golden signature is “0699b”. From the execution window, we can know that timing violation happens. So, my adder can’t operate at the speed of 1GHz.

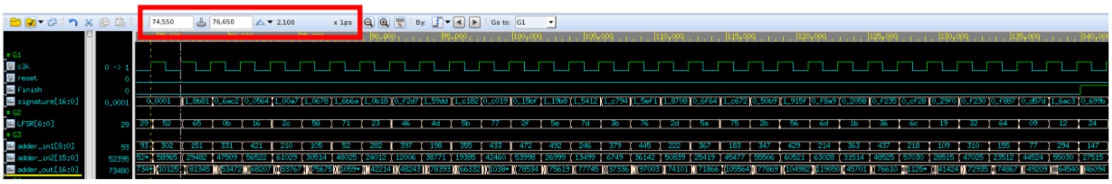




```
Warning!  Timing violation
           $setuphold<setup>( posedge CK &&& (flag == 1):50500 PS, posedge D:50378 PS,  0.142 : 142 PS,  0.024 : 24 PS );
            File: /home/m110/m110061579/process/CBDK_TSMC90GUTM_Arm_f1.0/CIC/Verilog/tsmc090.v, line = 10955
           Scope: BIST_Adder16_syn_tb.top.misr.data_out_reg_12_
            Time: 50500 PS
```

3. maximum operating speed

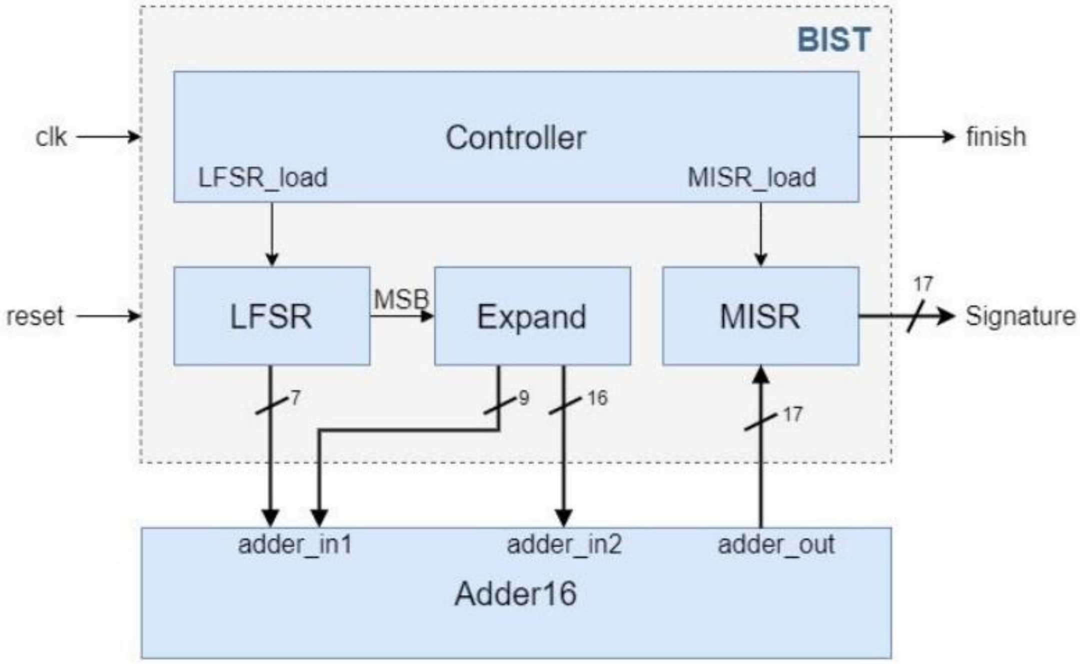
The maximum operating speed of my circuit is 476MHz (clock period = 2.1ns). The simulation waveform is shown below. From the cursor measurement, we can clearly see the clock period is 2.1ns.



Procedure :

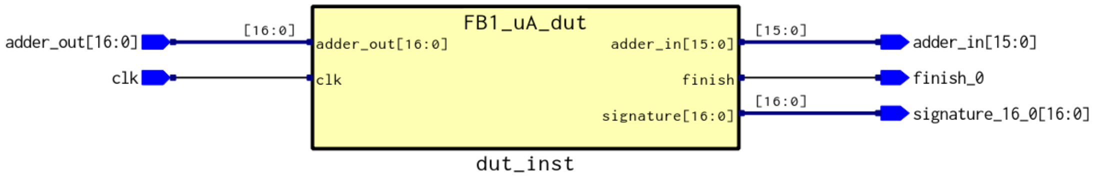
First, I set the clock period to 1ns. From the timing report of design complier shown below, “slack” is -1.08. I guess that my circuit can roughly operate at the clock period of 2.08ns (1+1.08=2.08). Then, I set the clock period to 2.1ns and 2.0ns respectively. Only at clock period 2.1ns, the signature is at the same as golden signature. So the maximum operating speed of my circuit is 476MHz

Block diagram:

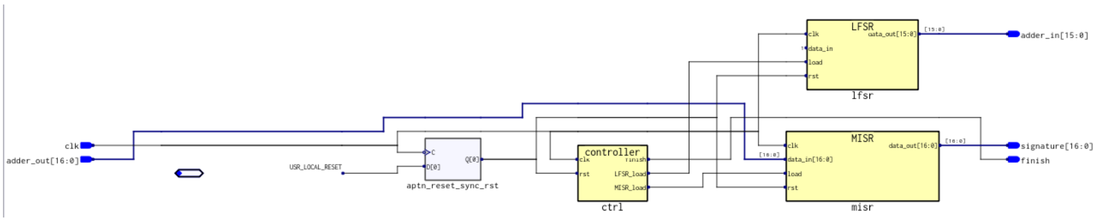


The schematics of FPGA_A and FPGA_B:

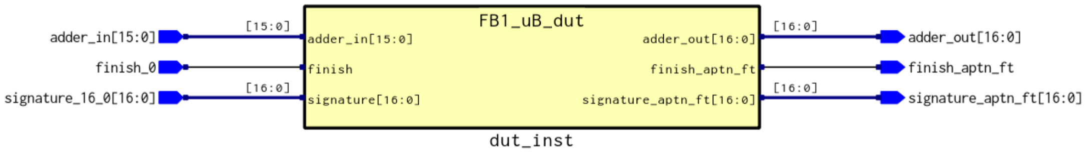
A:



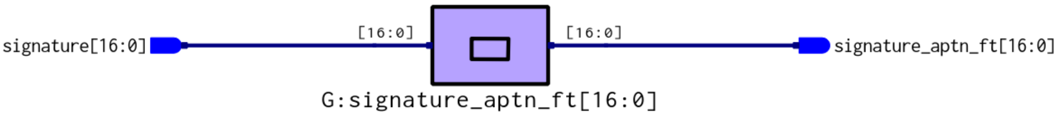
More

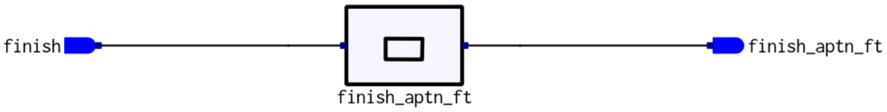


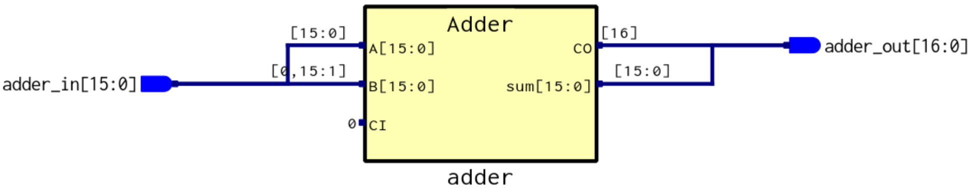
B:



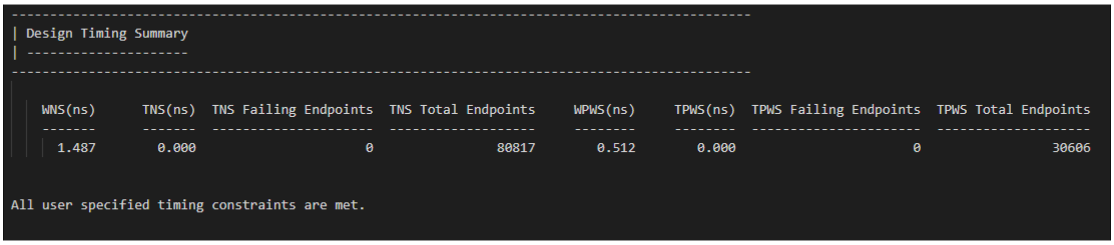
More







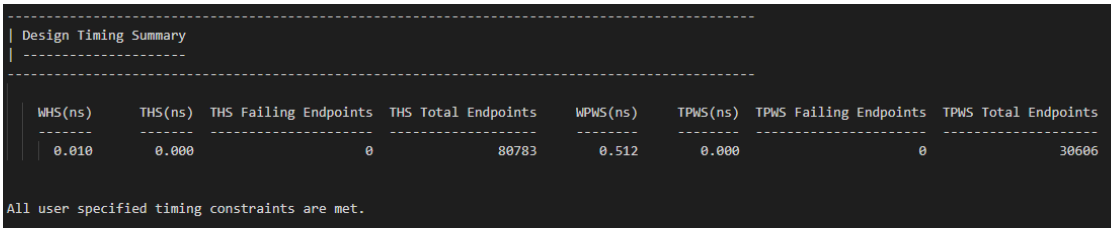
Timing report:



```
------------------------------------------------------------------------------------------------
| Design Timing Summary
| --------------------
------------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WPWS(ns)     TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoints
    -------      -------  ---------------------  -------------------      --------     --------  ----------------------  --------------------
      1.487        0.000                      0                80817         0.512        0.000                       0                 30606


All user specified timing constraints are met.
```



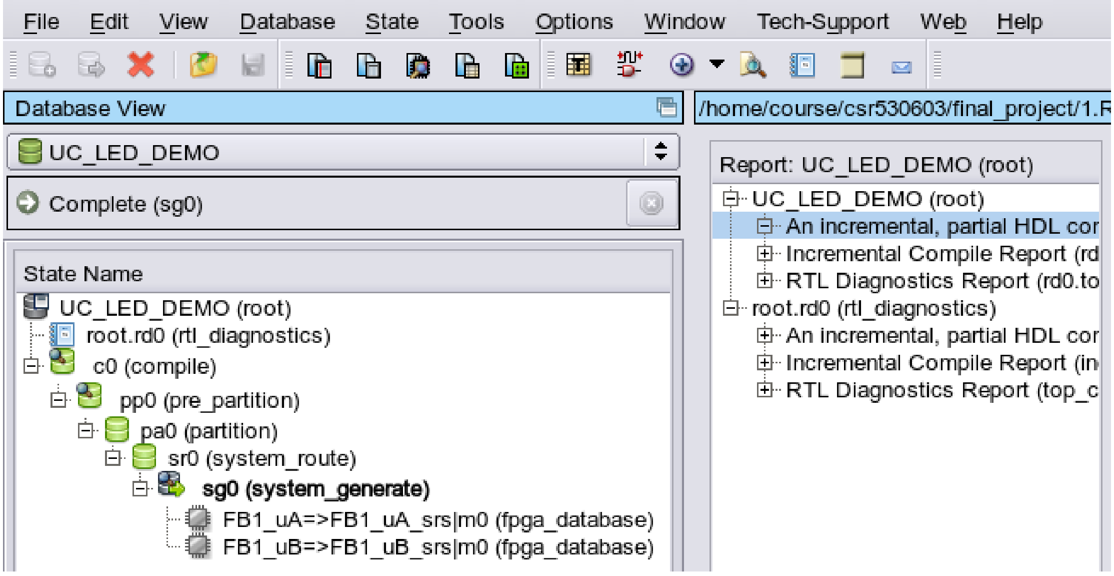
```
------------------------------------------------------------------------------------------------
| Design Timing Summary
| --------------------
------------------------------------------------------------------------------------------------

    WHS(ns)      THS(ns)  THS Failing Endpoints  THS Total Endpoints      WPWS(ns)     TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoints
    -------      -------  ---------------------  -------------------      --------     --------  ----------------------  --------------------
      0.010        0.000                      0                80783         0.512        0.000                       0                 30606


All user specified timing constraints are met.
```

Now, we use haps to realize “FPGA Design and Implementation Flow”, “FPGA Verification Flow” , “FPGA Prototyping Flow” and “improve Debug Capability”

Using HAPS complete screenshot:



/home/course/csr530603/final_project/1.R

UC_LED_DEMO

Complete (sg0)

State Name

```
UC_LED_DEMO (root)
  root.rd0 (rtl_diagnostics)
  c0 (compile)
    pp0 (pre_partition)
      pa0 (partition)
        sr0 (system_route)
          sg0 (system_generate)
            FB1_uA=>FB1_uA_srs|m0 (fpga_database)
            FB1_uB=>FB1_uB_srs|m0 (fpga_database)
```

Report: UC_LED_DEMO (root)

```
UC_LED_DEMO (root)
  An incremental, partial HDL cor
  Incremental Compile Report (rd
  RTL Diagnostics Report (rd0.to
root.rd0 (rtl_diagnostics)
  An incremental, partial HDL cor
  Incremental Compile Report (in
  RTL Diagnostics Report (top_c
```

Conclusion:

HAPS prototyping system provides a more reliable verification method that can help design teams identify and solve problems more quickly, thereby improving the quality and efficiency of the design.

Also, they have some advantages.

1. Faster verification speed: The HAPS prototyping system can run hardware prototypes in real-time, making it possible to verify the correctness of designs more quickly and detect errors earlier.
2. Higher accuracy: Since the HAPS prototyping system is based on an FPGA hardware prototype, it can simulate real hardware behavior more precisely.
   Higher reusability: Since the HAPS prototyping system is programmable, it can be reused in different design projects, saving time and cost.

Finally, I want to take a moment to thank our HAPS and one-element teachers for all the effort they put into teaching us. Without their hard work and dedication, we wouldn't have learned as much as we did. They really went above and beyond to make sure we understood the material and could apply it to real-life situations. It's clear that they genuinely care about our success, and we're grateful for all the time and energy they've invested in us. We've gained so much knowledge and confidence under their guidance, and we're excited to see where it takes us in the future. Thank you so much!