Homework 7: VGA Circle Report
December 11, 2017
Sabbir Ahmed

1 Background

In this project students will explicitly implement a computational finite-state machine, utilize rescheduling and resource sharing, and become familiar with the concept of using an on-chip clock multiplier. Students will leverage the faster clock to implement computations in a serial fashion. In this project, students will display a circle on the screen, examine analysis reports, and modify synthesis options.

2 Implementation

Multiple designs were implemented to analyze their effects on resource sharing and timing constraints.

2.1 Single Cycle Computation Design

The initial design implemented the entire inequality in a single cycle. Since the design emphasized on the computation being performed in a single cycle, explicitly generating several registers to hold the constant state value was unnecessary. An additional state was included for the synthesizer to consider encoding the FSM. The single-state module successfully generated the circle on the VGA screen using the formula $(x - x_c)^2 + (y - y_c)^2 < 10000$. The module is initialized with an asynchronous reset.

2.1.1 Testbench

Figure 1 provides the waveforms generated by sample coordinates (x, y) to the module.

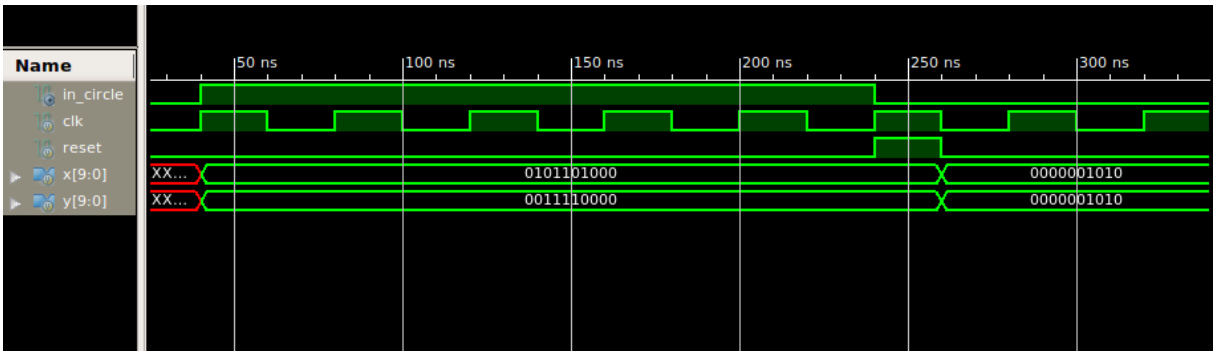

Figure 1: Single cycle computation design demonstrating the circle flag (`in_circle`) activated at $(360, 240)$ and deactivated at $(10, 10)$.

2.1.2 Synthesis

As expected, the design did not meet the timing constraints. Figure 2 provides the summary of the time constraints report, where the constraint `TS_uut_CLK0_BUF` was not met. Table **??** lists its timing slacks:

Table 1: FSM state encoding generated by the synthesizer for the single cycle design.

Check	Worst Case Slack
SETUP	-6.736 ns
HOLD	1.003 ns

Table 2 provides the macro statistics generated by the synthesizer.

```
Derived Constraint Report
Review Timing Report for more details on the following derived constraints.
To create a Timing Report, run "trce -v 12 -fastpaths -o design_timing_report design.ncd design.pcf"
or "Run Timing Analysis" from Timing Analyzer (timingan).
Derived Constraints for TS_CLK_50MHZ
+-------------------------------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
|                               |   Period    |       Actual Period       |      Timing Errors        |      Paths Analyzed       |
|          Constraint           | Requirement |-------------+-------------|-------------+-------------|-------------+-------------|
|                               |             |   Direct    | Derivative  |   Direct    | Derivative  |   Direct    | Derivative  |
+-------------------------------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+
|TS_CLK_50MHZ                   |     20.000ns|      7.500ns|     33.472ns|            0|            2|            0|       115653|
| TS_dcm_uut_CLK0_BUF           |     20.000ns|     33.472ns|          N/A|            2|            0|       114964|            0|
| TS_dcm_uut_CLK2X_BUF          |     10.000ns|      9.263ns|          N/A|            0|            0|          689|            0|
+-------------------------------+-------------+-------------+-------------+-------------+-------------+-------------+-------------+

1 constraint not met.
```

Figure 2: Screen capture of the timing constraint report showing failure of `TS_uut_CLK0_BUF` of the DCM.

The multipliers are used to multiply the two 21-bit squared coordinates. Several adders and subtractors are used in the design to handle `pos_v`, `pos_h` and the centers coordinates.

Table 2: Macro statistics generated by the synthesizer for the single cycle design.

# Multipliers	**2**
11x11-bit multiplier	2
# Adders/Subtractors	**4**
10-bit adder	1
11-bit subtractor	2
23-bit adder	1
# Counters	**2**
10-bit up counter	2
# Registers	**8**
1-bit register	8
# Comparators	**1**
24-bit comparator less	1

2.1.3 States

Table 3 provides the FSM states encoded by the synthesizer. The automatic-encoding encoded states do not differ from the values assigned to them during initialization because of the small number of states. The states were intentionally assigned with 2-bit values to alert the synthesizer of the FSM.

Table 3: FSM state encoding generated by the synthesizer for the single cycle design.

State	Encoding
00	00
01	01

1. **INIT (00):** Serves as a buffer to the computational state. This state serves no other purpose.
2. **COMPUTE (01):** Computes the entire circle inequality.