Asynchronous FIFO Design

KARNATI PARANJAI

IEC2021097

Final year student, B-Tech ECE, IIIT ALLAHABAD

SAMPLE OUTPUTS

1) Sample testbench and corresponding output when **freq write_clock > freq read_clock**

Testbench:

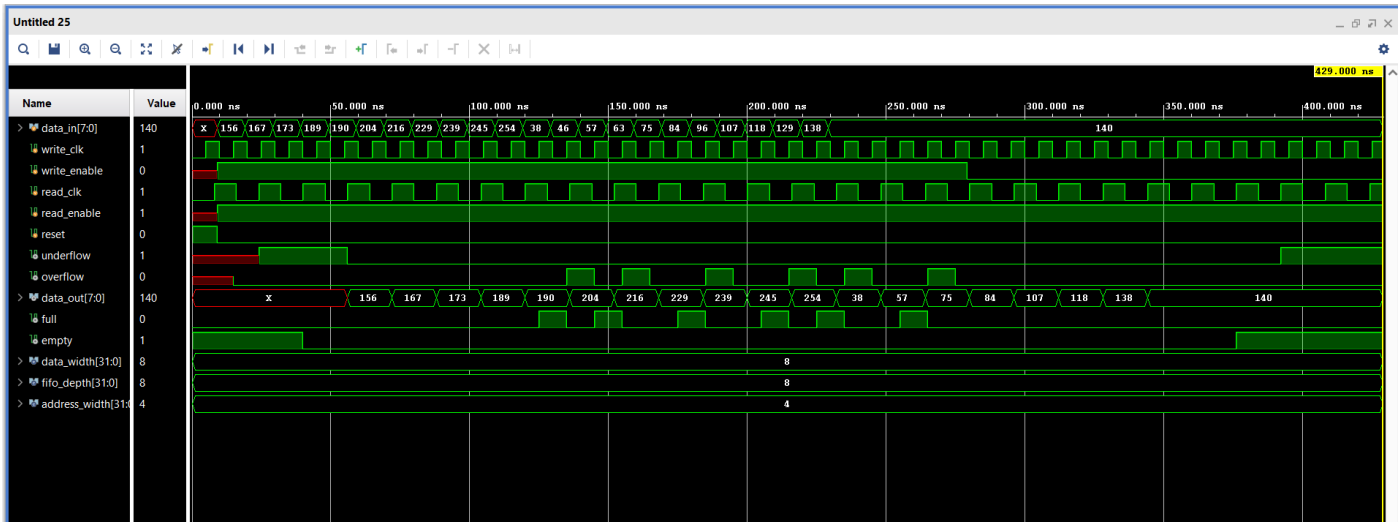
```
always #5 write_clk = ~write_clk;
always #8 read_clk = ~read_clk;

initial begin
write_clk = 0;
read_clk = 0;
reset = 1;
#9 write_enable = 1;reset =0; read_enable=1;data_in = 8'd156;
#10 data_in = 8'd167;
#10 data_in = 8'd173;
#10 data_in = 8'd189;
#10 data_in = 8'd190;
#10 data_in = 8'd204;
#10 data_in = 8'd216;
#10 data_in = 8'd229;
#10 data_in = 8'd239;
#10 data_in = 8'd245;
#10 data_in = 8'd254;
#10 data_in = 8'd38;
#10 data_in = 8'd46;
#10 data_in = 8'd57;
#10 data_in = 8'd63;
#10 data_in = 8'd75;
#10 data_in = 8'd84;
#10 data_in = 8'd96;
#10 data_in = 8'd107;
#10 data_in = 8'd118;
#10 data_in = 8'd129;
#10 data_in = 8'd138;
#10 data_in = 8'd140;
#50 write_enable = 0;
#150 $finish;
end
endmodule
```

Observation :

Since the write clock is faster than the read clock, consequently the FIFO gets full frequently and we observe frequent overflows during the transfer of data and the data is transferred across the clock domain successfully.

Output:



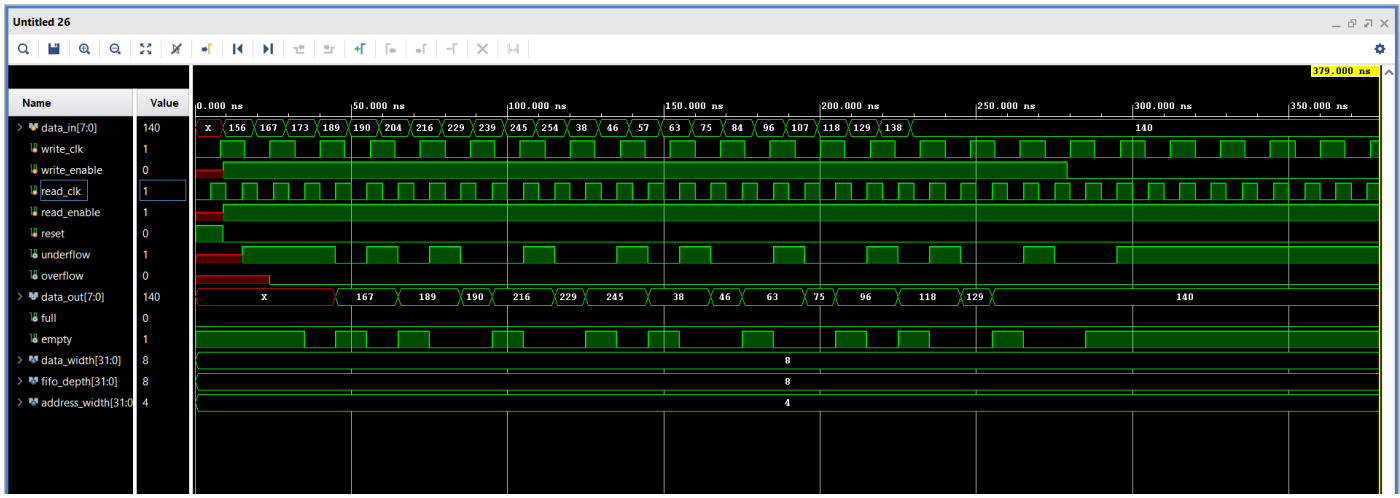
2) Sample testbench and corresponding output for when **freq write_clock < freq read_clock:**

Testbench:

```
always #8 write_clk = ~write_clk;
always #5 read_clk = ~read_clk;

initial begin
write_clk = 0;
read_clk = 0;
reset = 1;
#9 write_enable = 1;reset =0; read_enable=1;data_in = 8'd156;
#10 data_in = 8'd167;
#10 data_in = 8'd173;
#10 data_in = 8'd189;
#10 data_in = 8'd190;
#10 data_in = 8'd204;
#10 data_in = 8'd216;
#10 data_in = 8'd229;
#10 data_in = 8'd239;
#10 data_in = 8'd245;
#10 data_in = 8'd254;
#10 data_in = 8'd38;
#10 data_in = 8'd46;
#10 data_in = 8'd57;
#10 data_in = 8'd63;
#10 data_in = 8'd75;
#10 data_in = 8'd84;
#10 data_in = 8'd96;
#10 data_in = 8'd107;
#10 data_in = 8'd118;
#10 data_in = 8'd129;
#10 data_in = 8'd138;
#10 data_in = 8'd140;
#50 write_enable = 0;
#100 $finish;
end
endmodule
```

Output:



Observation:

Since the read clock is faster than the write clock, consequently the FIFO gets empty frequently and hence we observe frequent underflows during the transfer of data and the data is transferred across the clock domain successfully.