Lab 05: Synthesizable Asynchronous FIFO

Specification

1. Design a 256X16 asynchronous FIFO transferring data across two clock domains. Before you proceed, read the following two papers (hint: synthesizable asynchronous FIFO is detailed in the second paper):
   - Clifford E. Cummings, "Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs," SNUG 2001, San Jose.
   - Clifford E. Cummings, "Simulation and Synthesis Techniques for Asynchronous FIFO Design," SNUG 2002, San Jose.
2. Primary IOs (you should know the meaning of signals with no description):
   (a) rst_n_w: write reset
   (b) clk_w: write clock
   (c) push_w: push (or write enable)
   (d) datain_w
   (e) fifo_full_w
   (f) fifo_almost_full_w
   (g) fifo_error_w
   (h) rst_n_r: read reset
   (i) clk_r: read clock
   (j) pop_r: pop (or read enable)
   (k) dataout_r
   (l) fifo_empty_r
   (m) fifo_almost_empty_r
   (n) fifo_error_r
3. Use memory compiler to generate the memory needed in the asynchronous FIFO.
4. Avoid using Verilog **`assign`** statement, except the direct assignment of two signals, e.g., `assign data = memory[addr];`
5. Perform the synthesis and post-synthesis simulation.
6. Verify your design.
   - In your testbench, you should consider the operations between clock domains, e.g., PUSH and POP should be operated at different clock domains, respectively. How do you verify the functionality of FIFO comprehensively? You are required to design your own test patterns.
   - At least the two test cases should be considered:
     a. clk_w > clk_r
     b. clk_w < clk_r
   - Show the PUSH/POP commands, empty/full/error/almost flags and the corresponding results to TA and verify the correctness of your design. When doubt, ask your TA!

The Usage of Memory Compiler

➔ Use Memory Compiler for 0.18um process. The executables are in

/theda21_2/CBDK_TSMC018_Arm_f1.0/CIC/Memory/

➔ To open it, make sure you are in the working directory, execute the compiler by using full path. E.g., to open the memory compiler of dual-port SRAM:
~/hw5$ /theda21_2/CBDK_TSMC018_Arm_f1.0/CIC/Memory/ra2sh

➔ You CANNOT execute the memory compiler under
/theda21_2/CBDK_TSMC018_Arm_f1.0/CIC/Memory/
Otherwise, you will get the error of "Unable to write to logfile, ACI.log."

Other Reference

1. Clifford E. Cummings, "Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons," SNUG 2002, San Jose.
2. Mohit Arora, "Chapter 3: Handling Multiple Clocks in The Art of Hardware Architecture," Springer, 2012. (The e-book can be accessed via NTHU Library.)

Assignment due on Jan 3, 2017 11:59PM

➔ Submit your code with a summary report.
➔ Make a demo scheduled by our TAs.
➔ Demonstrate the simulation and revision history based on TAs' requirement.
➔ Questions about inter-clock domains will be asked in the demo.