**Lab report 2**

**ESE 461: Design Automation for Integrated Circuit Systems**

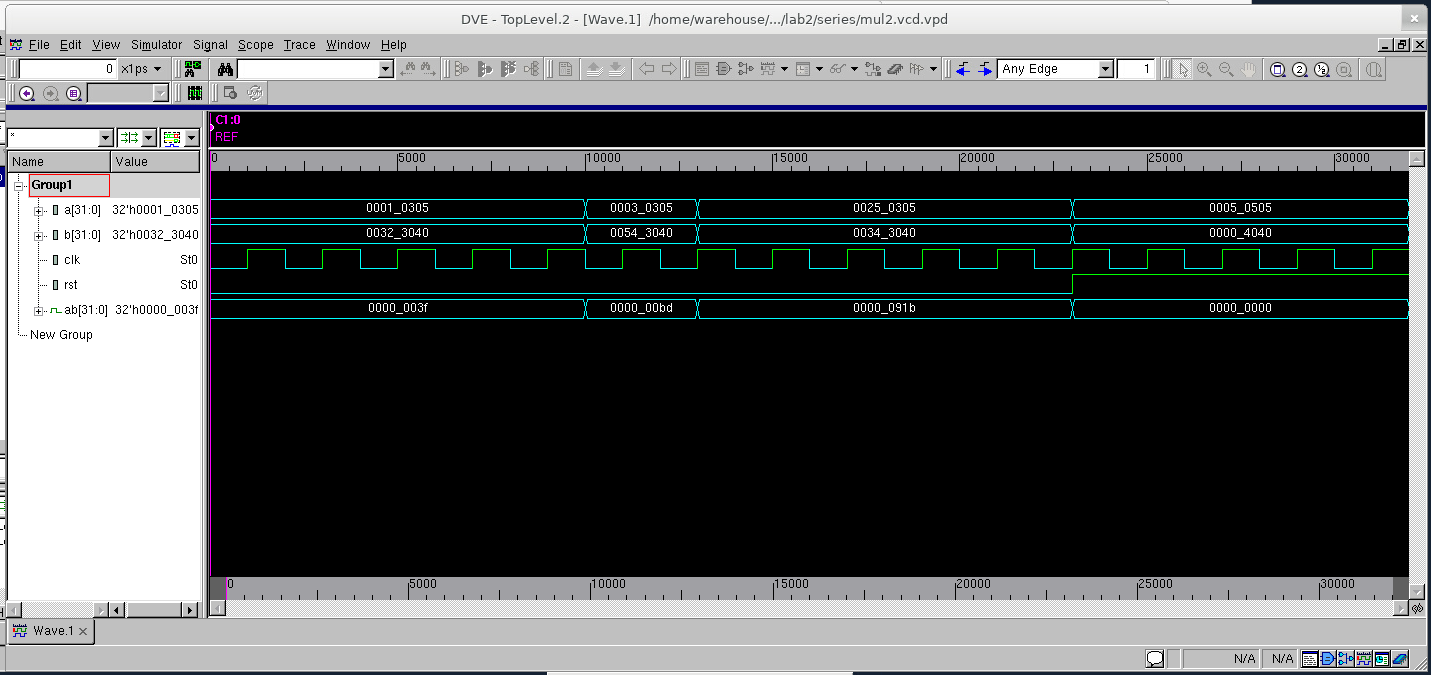
**Zhiliang Peng**

**Question 1**

Design the fixed-multiplier in the sequential processed way (accumulate only one shifted a to the 64-bit product) in the following figure. Simulate your Verilog code in VCS and analyzes your simulation result.

**Design method:**

To achieve the sequential processed design, I use a FSM to do the control module. There are two state inside the FSM (though you won’t see from result screen snap, you will see inside my code). One is ADD, and the other is SFT. ADD means when b [0] == 1, add a to abtemp and shift b to the right for 1 bit. SFT means shift b for 1 bit. Each time when b[0] is 1, abtemp adds up to form a 64 bits accurate product. Since we want only 32 bits of product, we truncate abtemp[47:16] into ab, which is the final result. Notice that, all the numbers in the design is fixed point number in Q44 format, and we ignored potential overflow.

**Result:**

At each positive edge of clock ab = a\*b when reset is 0. And when reset is true, pull ab down to 0000\_0000. (Hex number)

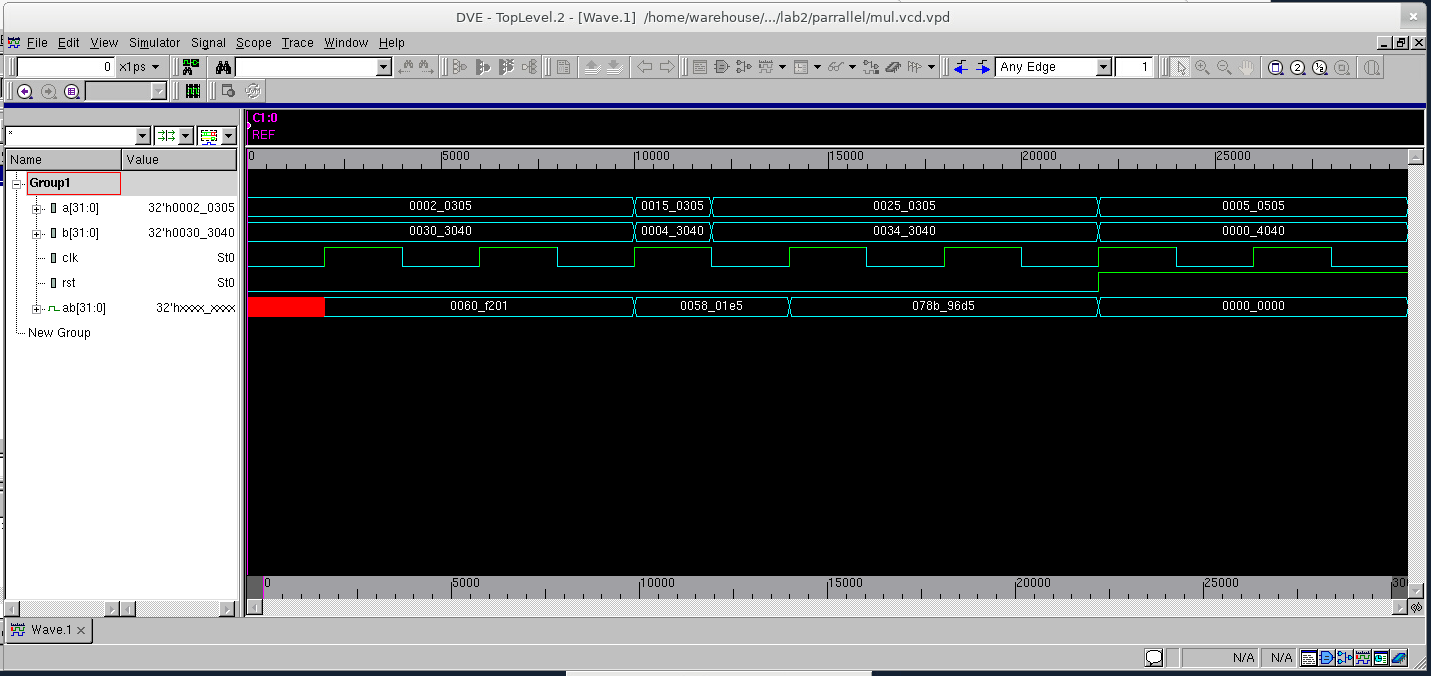
**Question 2**

Design the fixed-multiplier in the parallel processed way (accumulate all shifted a to the 64-bit product in one clock cycle) in the following figure. Simulate your Verilog code in VCS and analyzes your simulation result.

**Design method:**

To achieve parallel processed design. I use 32 mux to select either a itself or 0000\_0000 based on b[i] bit (i=1:32) at the same time. And then, at the same time, shift a for i bits and add up all a. The difference between question2 and question1 is that question2 select process and add process happen at once rather in procedures.

**Result:**



At each positive edge of clock ab = a\*b when reset is 0. And when reset is true, pull ab down to 0000\_0000. (Hex number)