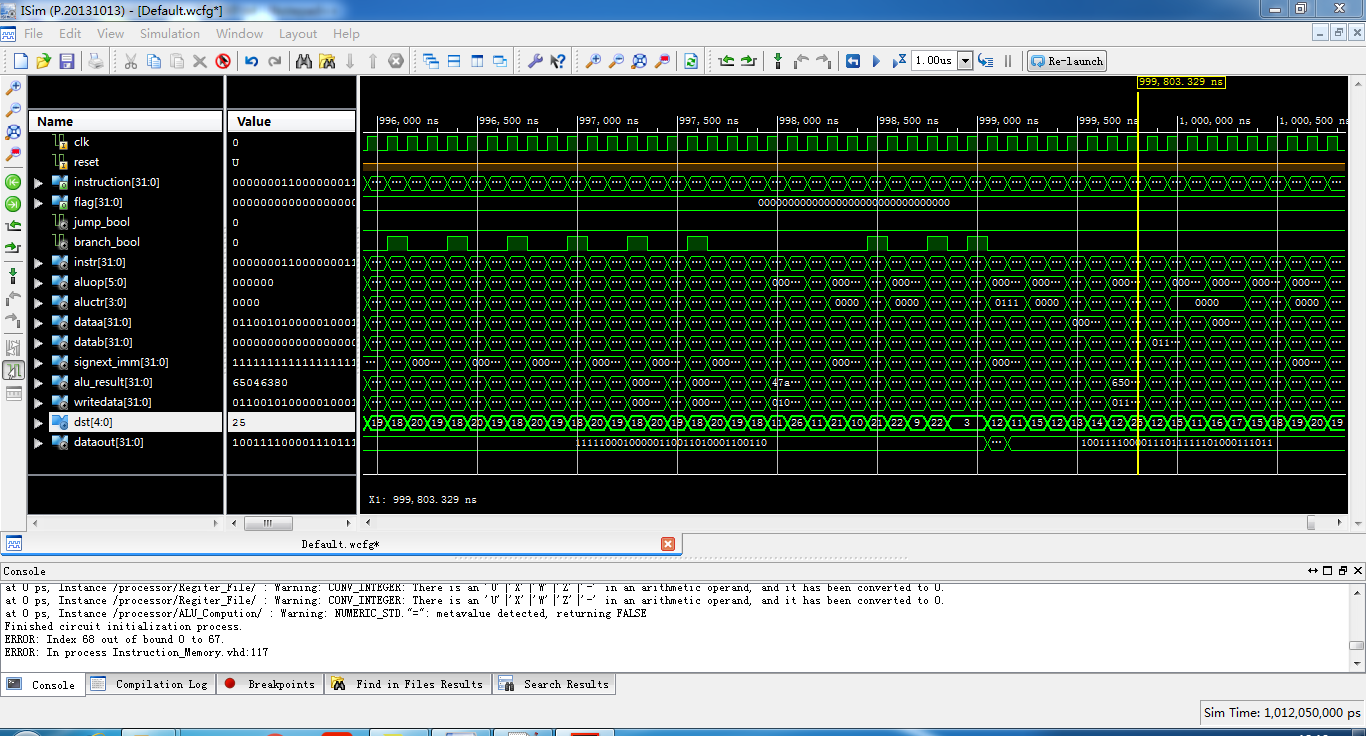
1. Simulation:
   1. Key expansion:

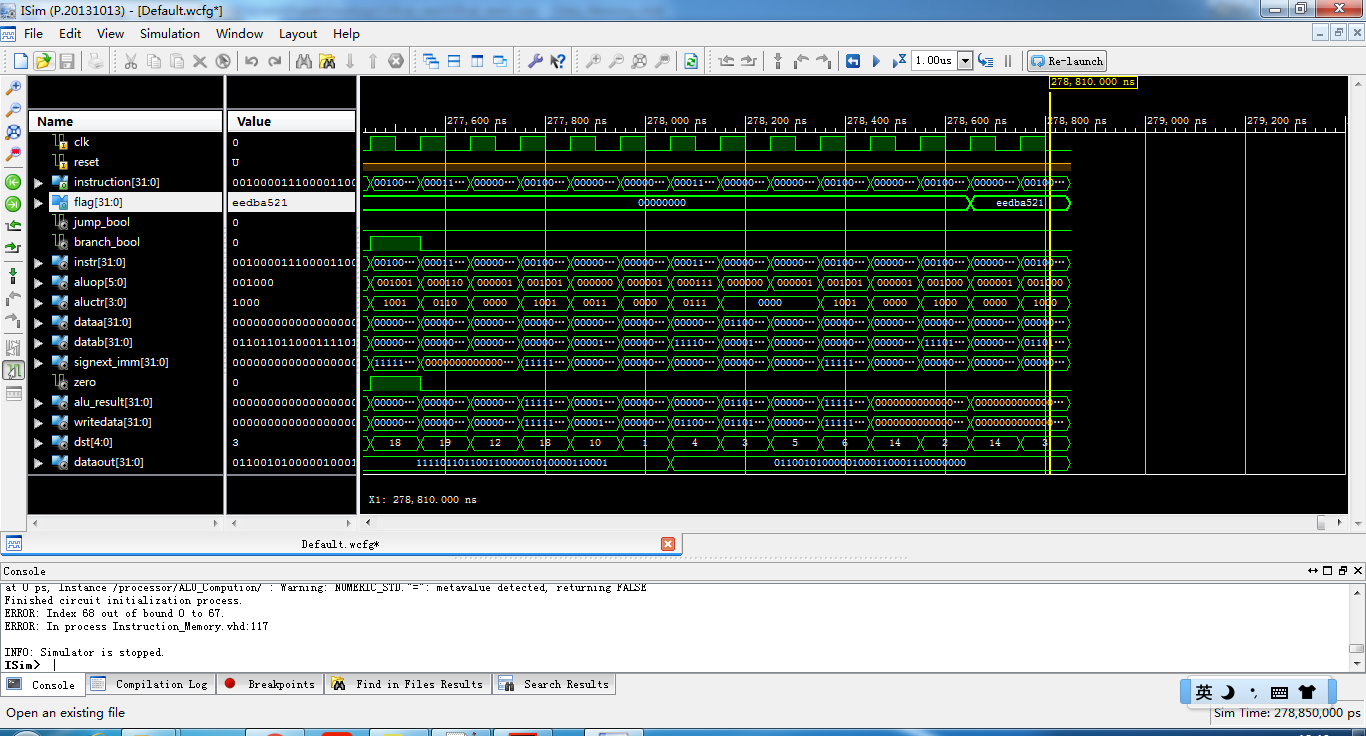


Above is the function simulation result of key expansion. I put key expansion codes into instruction memory file, and hard-coded Ukey(0x00000000000000000000000000000000) in the data memory file, to see if a correct Skey could be generated.

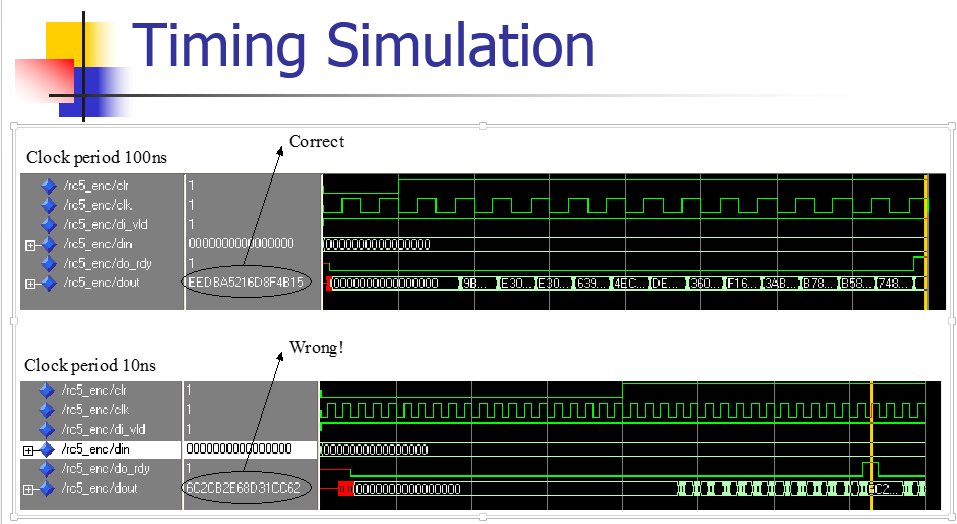
On the screenshot, dst[25] means R25 which stores Skey[25], and the value inside the alu\_result is the actual result, 0x65046380. This is correspond with the reference result from previous lab.



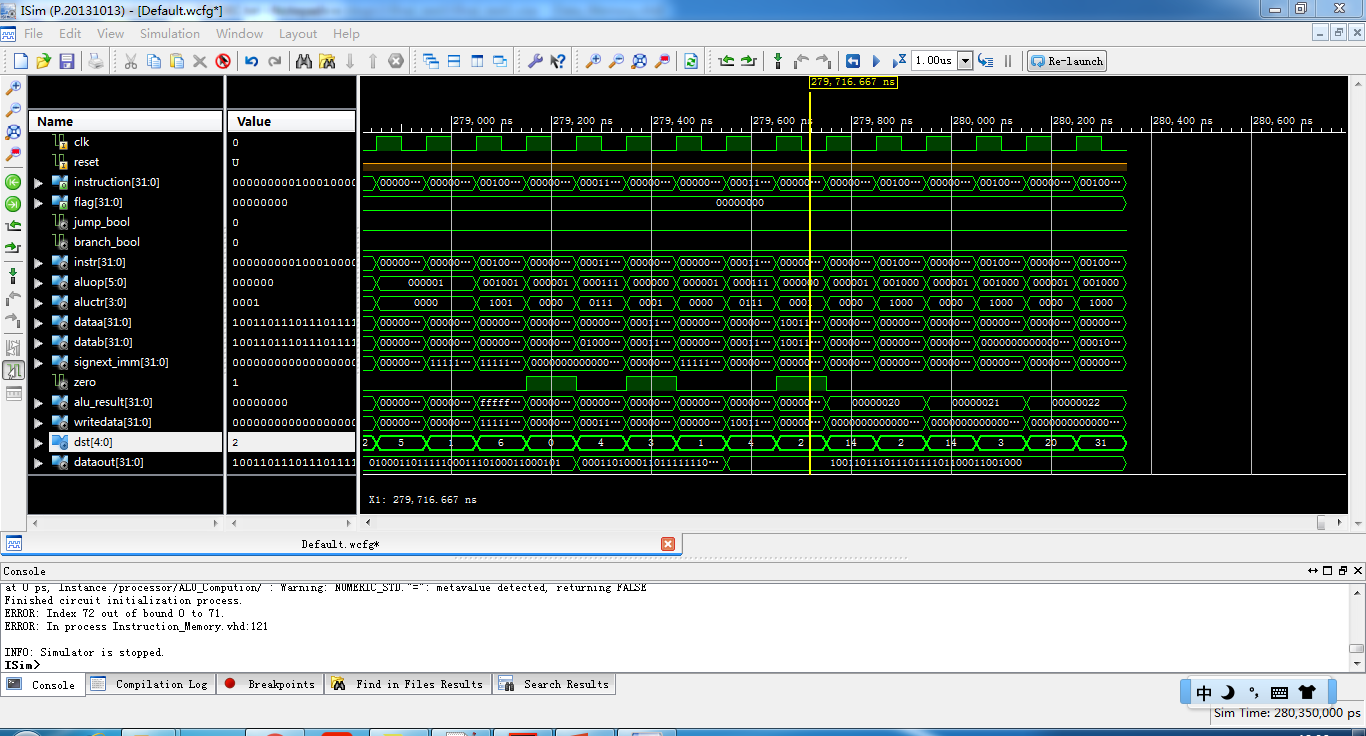
* 1. RC5 encryption:



This screenshot shows the result of RC5 encryption with Din(0x0000000000000000). We put codes into instruction memory file and hard-coded the generated Skeys into data memory file. From the screenshot, “flag” points to the memory block that stores the higher 32 bits of the result. It is correspond with the reference result in the lecture provided by professor.

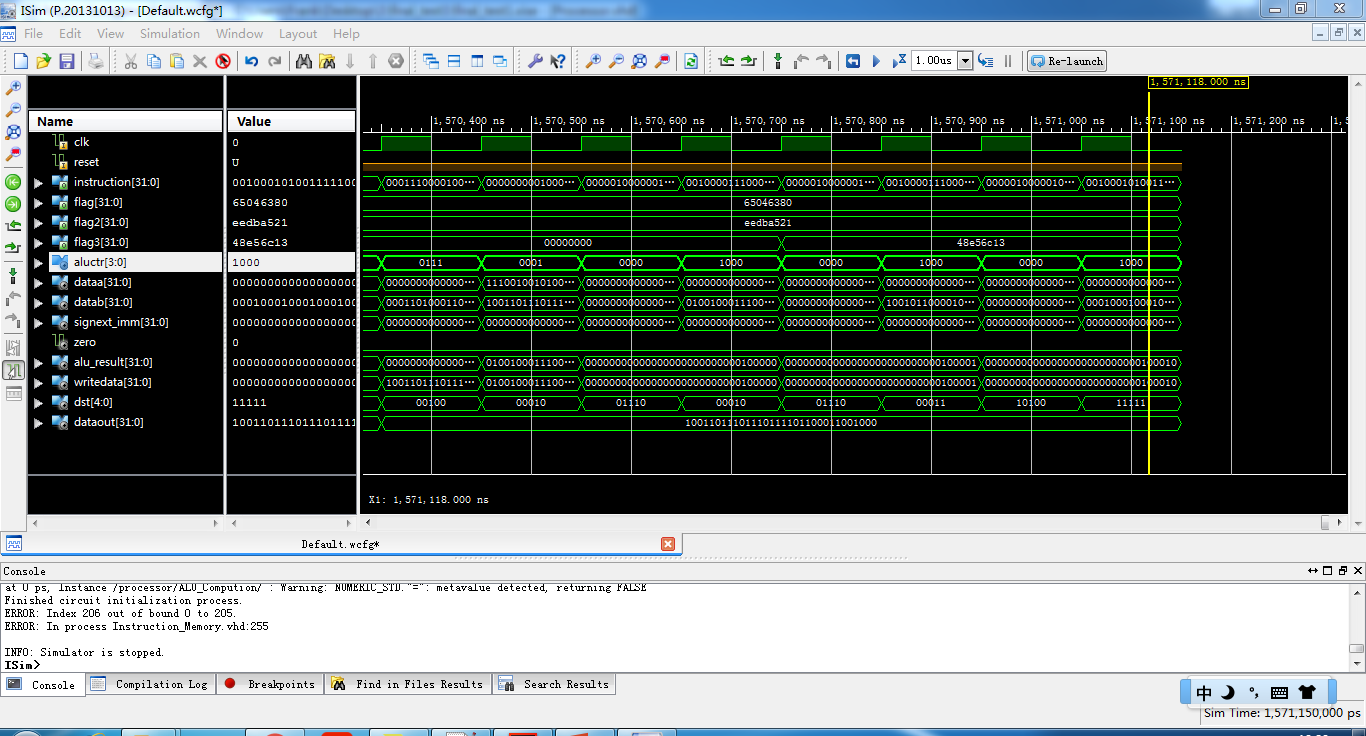


* 1. RC5 Decryption:



This screenshot shows the result of RC5 Decryption using the output from Encryption as input. So the result should be the input of RC5 Encryption, which is all 0s’. We put codes into instruction memory file and hard-coded the generated Skeys into data memory file. From the screenshot, “flag” points to the memory block that stores the higher 32 bits of the result. It is exactly the input we used for Encryption.

* 1. RC5 Key expansion, encryption, and decryption:



After separate module tests, we put all codes together to see if it works. “Flag”, “Flag2”, “Flag3” on the above screenshots stand for the results of key expansion, encryption, and decryption respectively. Clearly, “flag” and “flag2” show the identical results as separate ones. And since we used 0x0000000000000000 for both encryption and decryption, “flag2” are different from the previous one, but we proved that it is also the correct output.