# Simulation Results

1. ADD –operation: Code 1030, at 48th location of memory, I put 5 to read.

Figure 1: Fetch and decode processes are done for addition operation

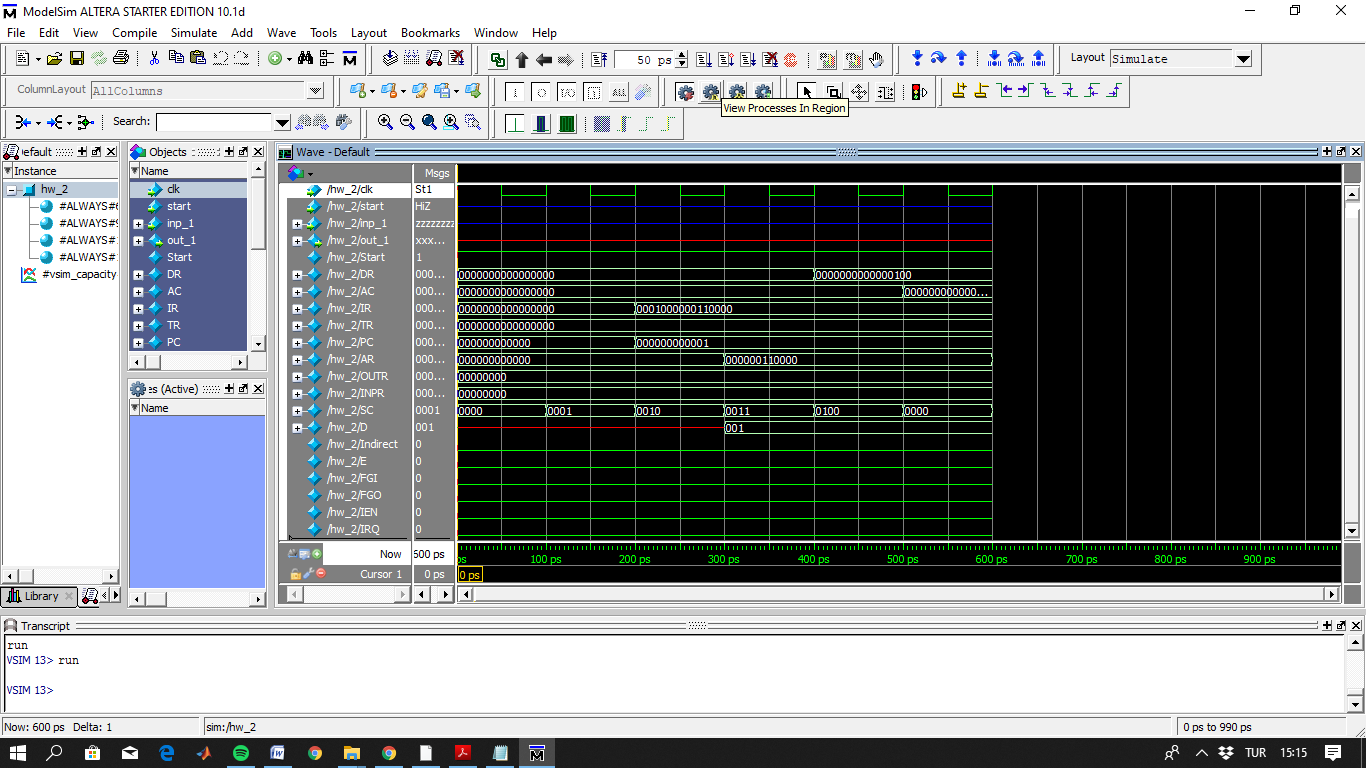
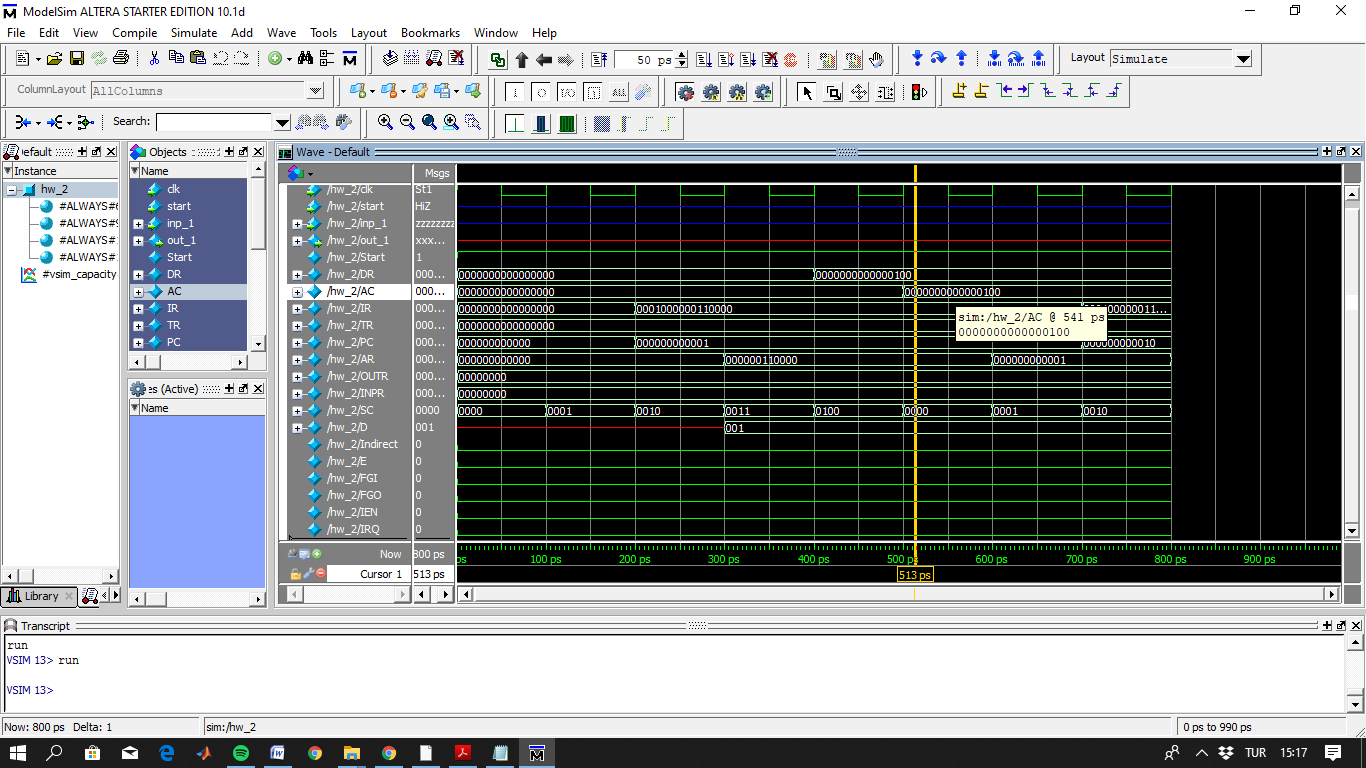


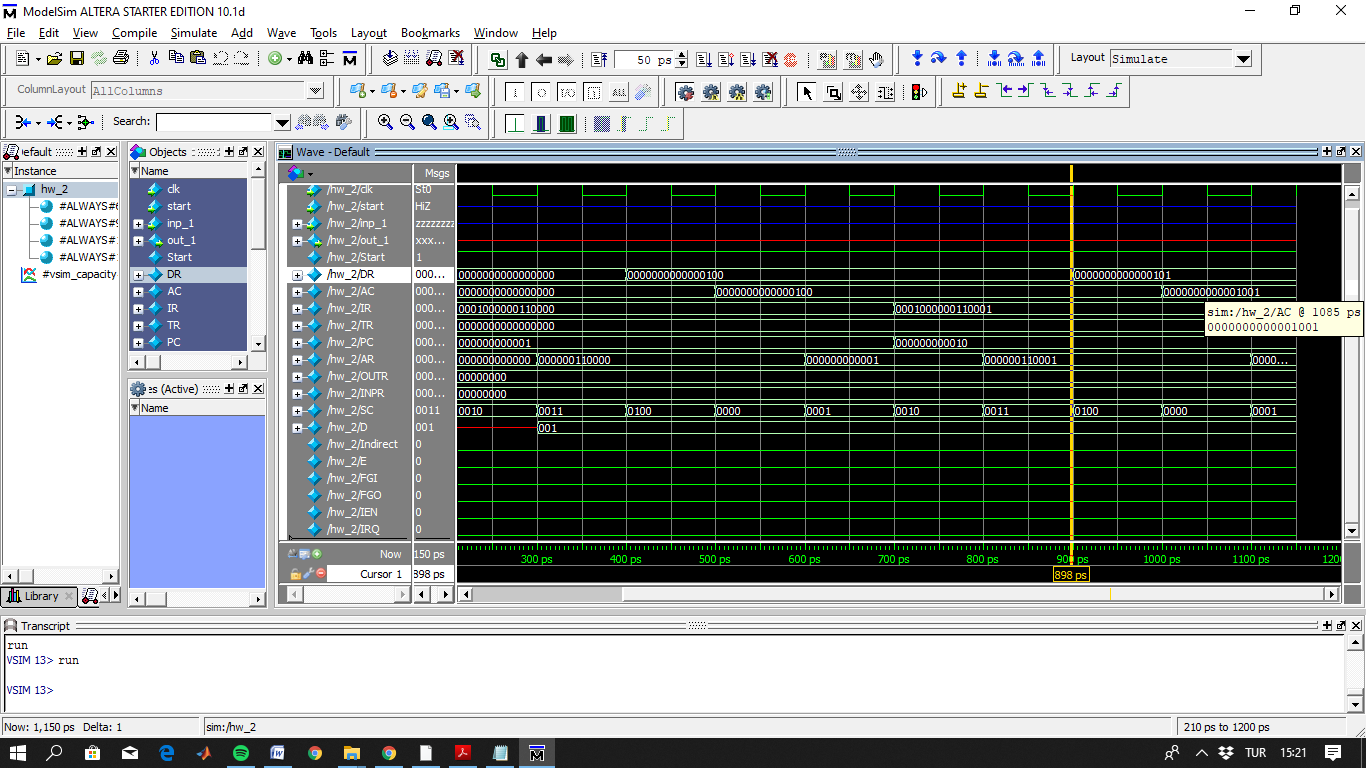
Figure 2: Addition occur and DR register holds 4 after that micro-operation



1. Another ADD operation: AC has already stored 4. ADD 5 to AC. Result is 9.

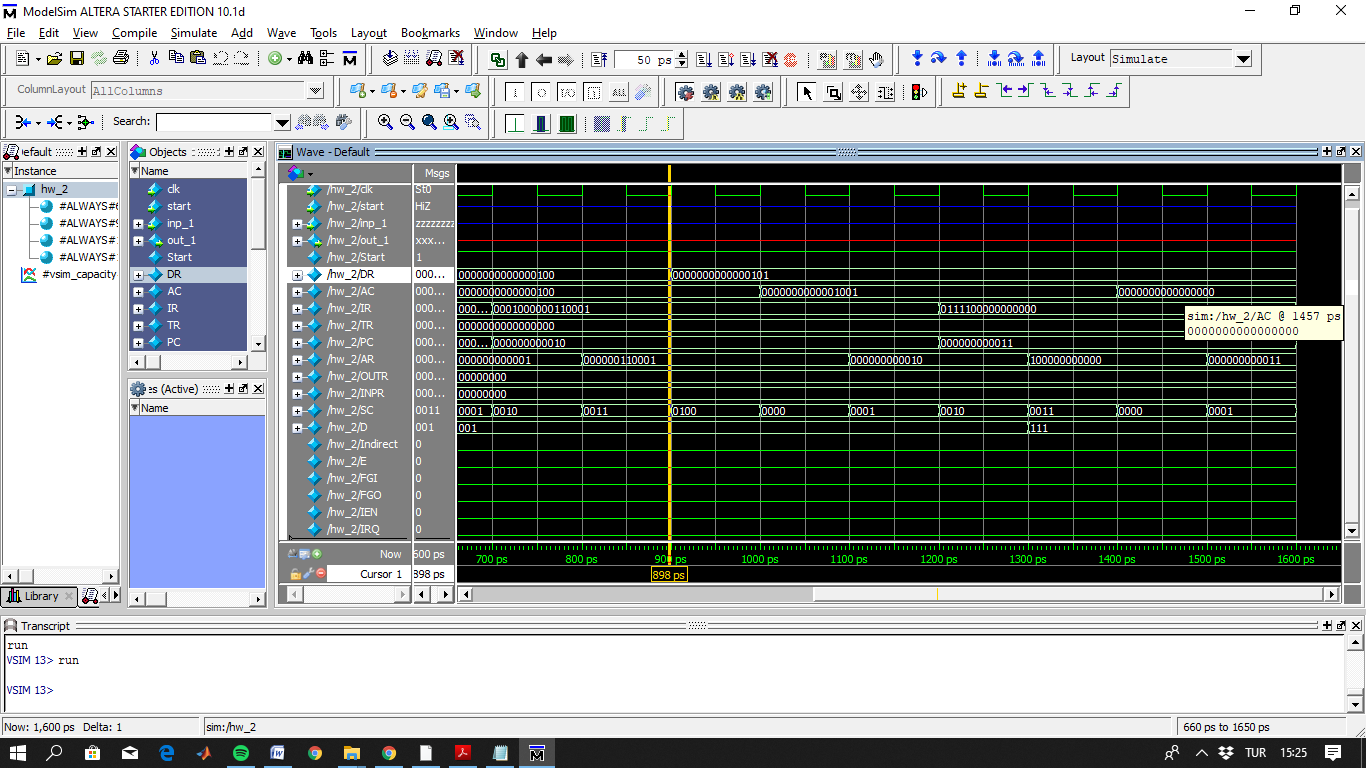
(Hex code 1031, first 1 for addition and at 49th location I put 5 to memory to read.)

Figure 3: 4 + 5 result stored into accumulator register



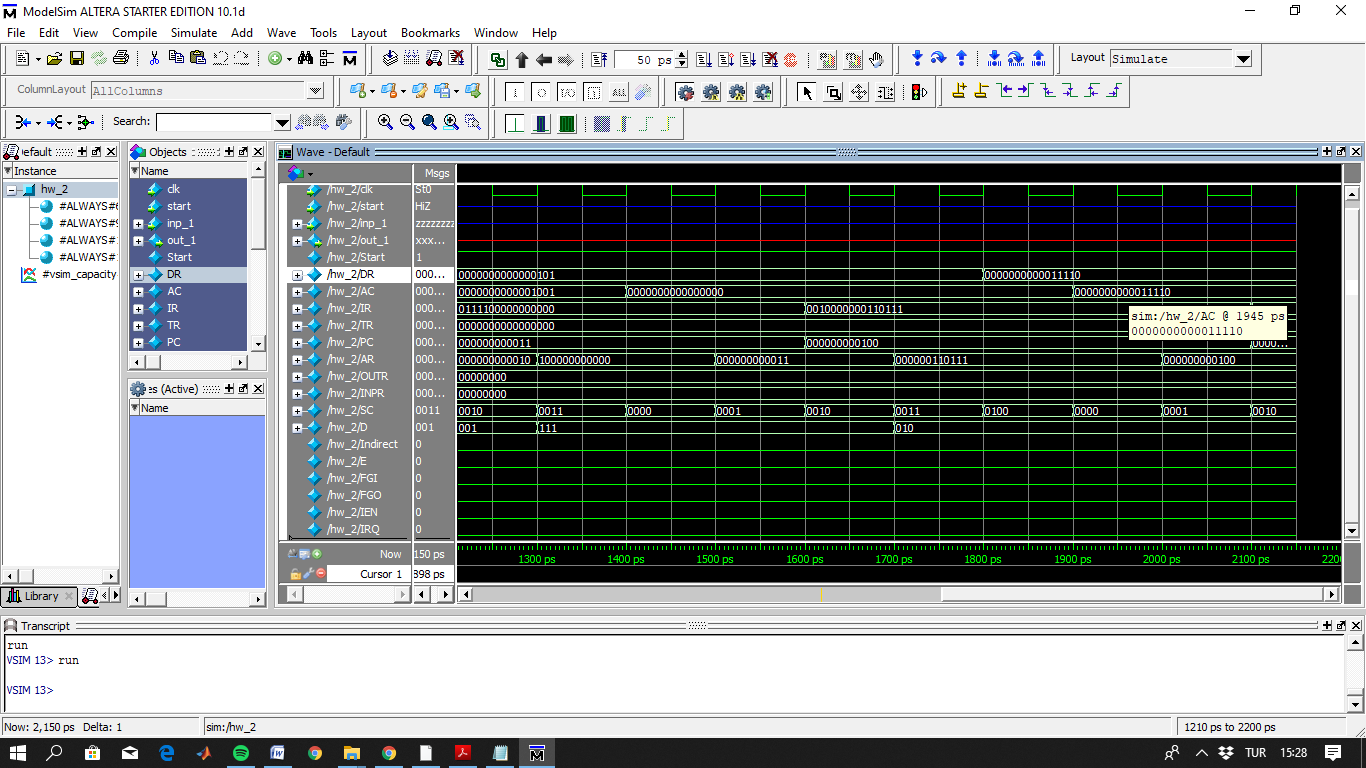
1. CLA operation: will clear accumulator, and code is 7800.

Figure 4: Accumulator is cleared



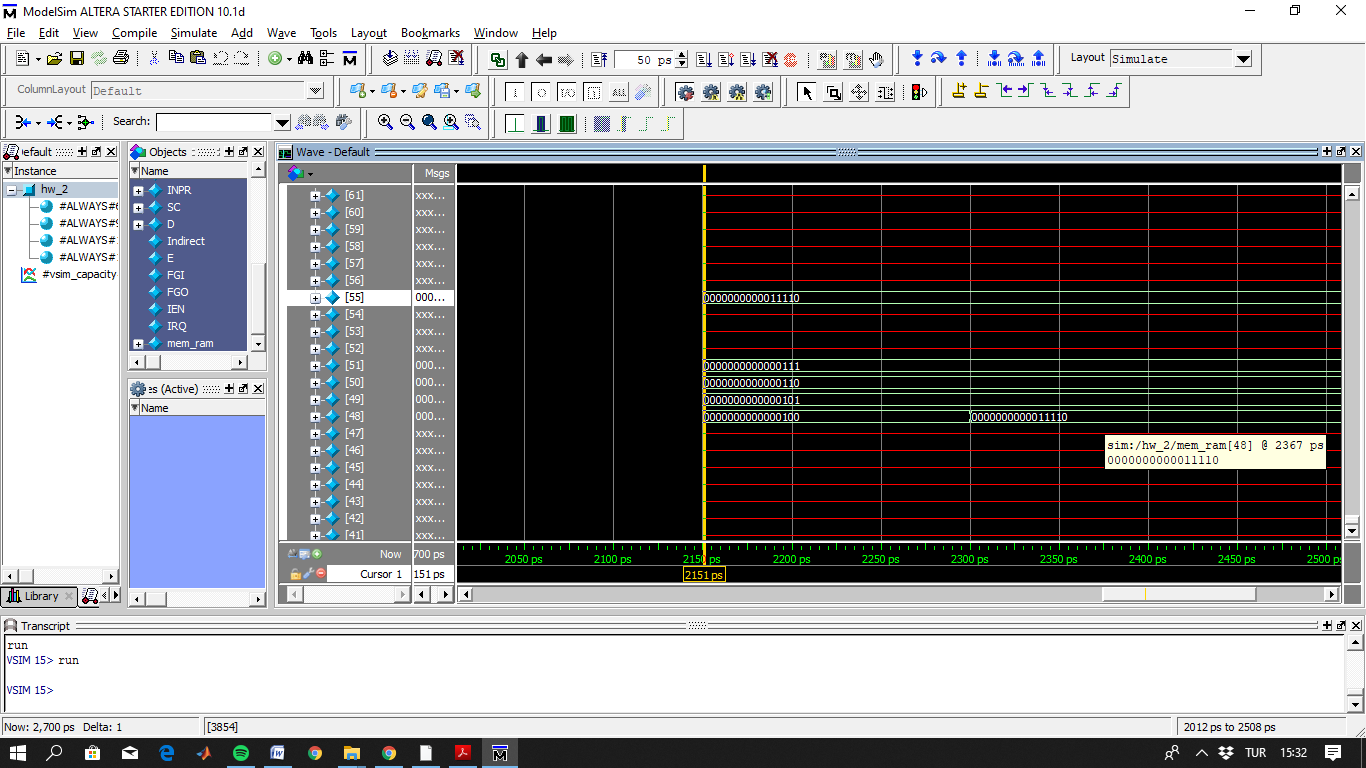
1. LDA (Load) operation: Hex code, h2037, 2 for load and, loaded data has been already stored in to 55th location of memory.

Figure 5: 30 is loaded to the accumulator



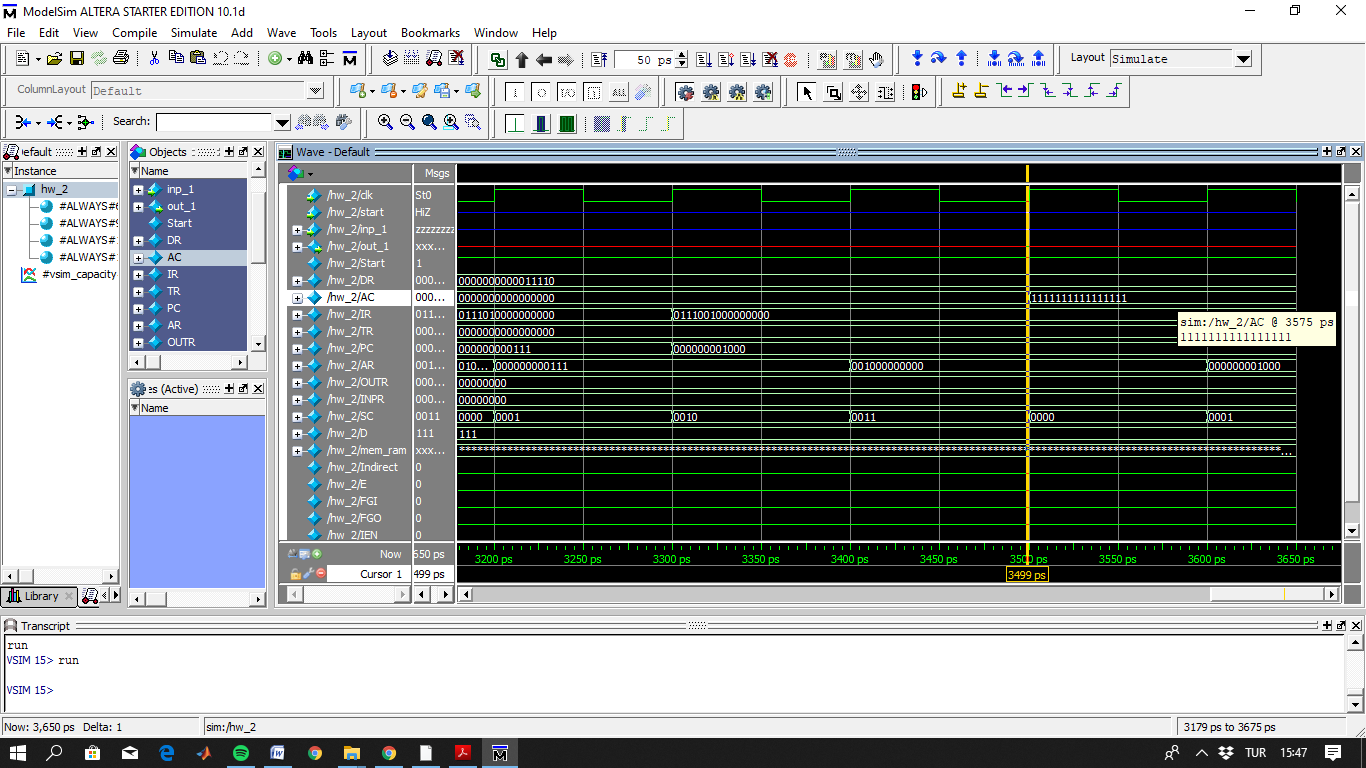
1. STR (store) operation: Hex code is h3030. 30 stored into 48th location of memory as shown below.

Figure 6: At 48th location of memory, 30 is stored



1. BUN, BSA and ISZ instructions are working well. One who wants to check these operations, are just delete comments and run since I put their hexadecimal codes into my main verilog code.
2. CLE (h7400) is implemented as similar to the CLA.
3. CMA operation: Hex code is h7200. That will complement of the value of AC.

Figure 7: AC is cleared before CMP operation. CMP of 0 is 16 bit 1 as shown.



1. CIR operation: Hex code is h7080. It will circulate the AC to the right, and E will hold least significant bit of AC.

Figure 8: Firstly AC is 29

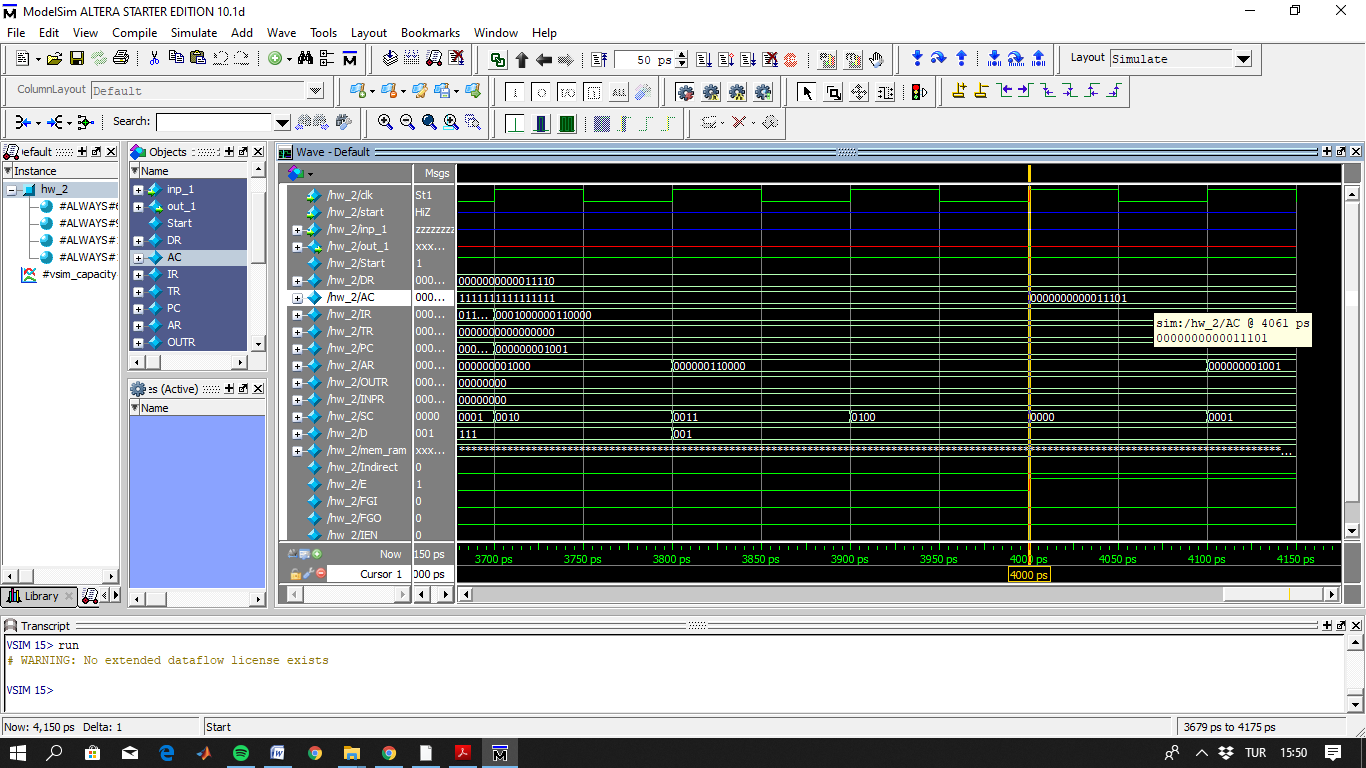
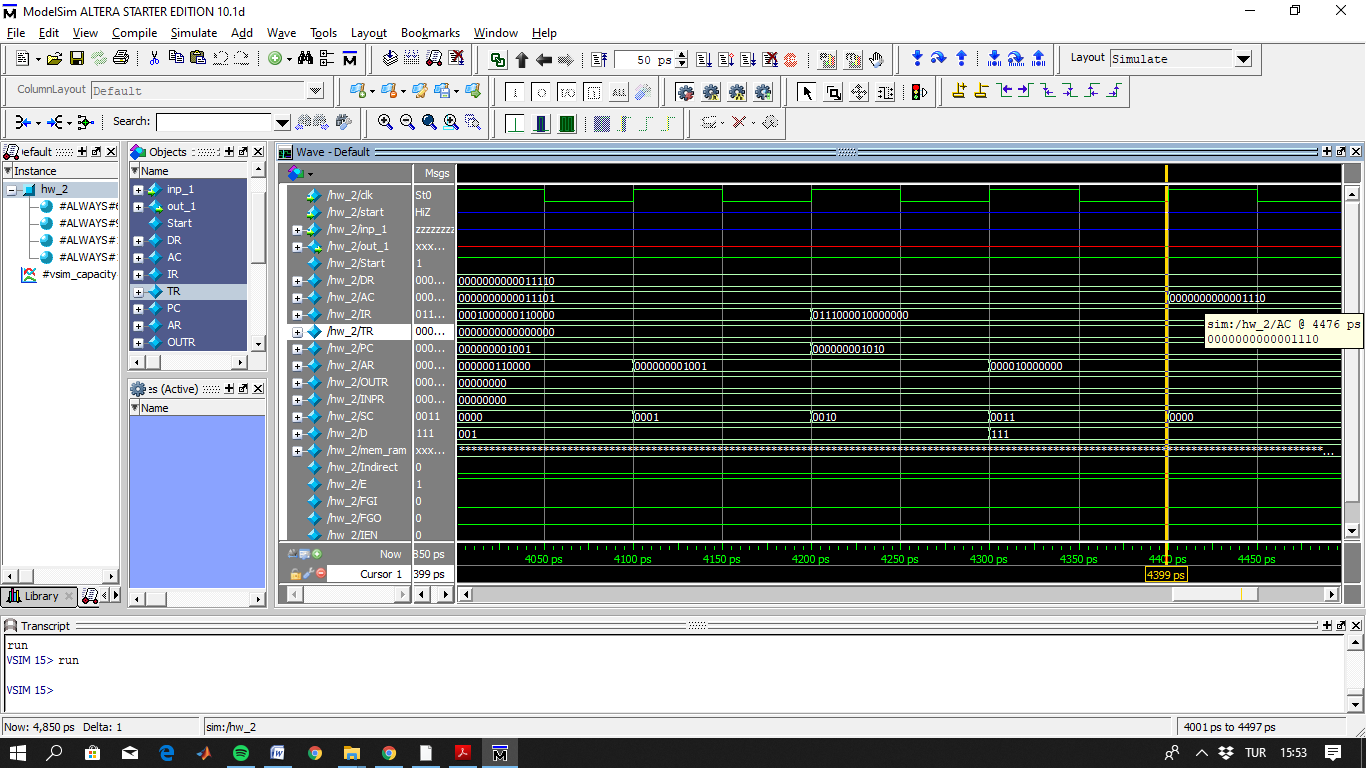
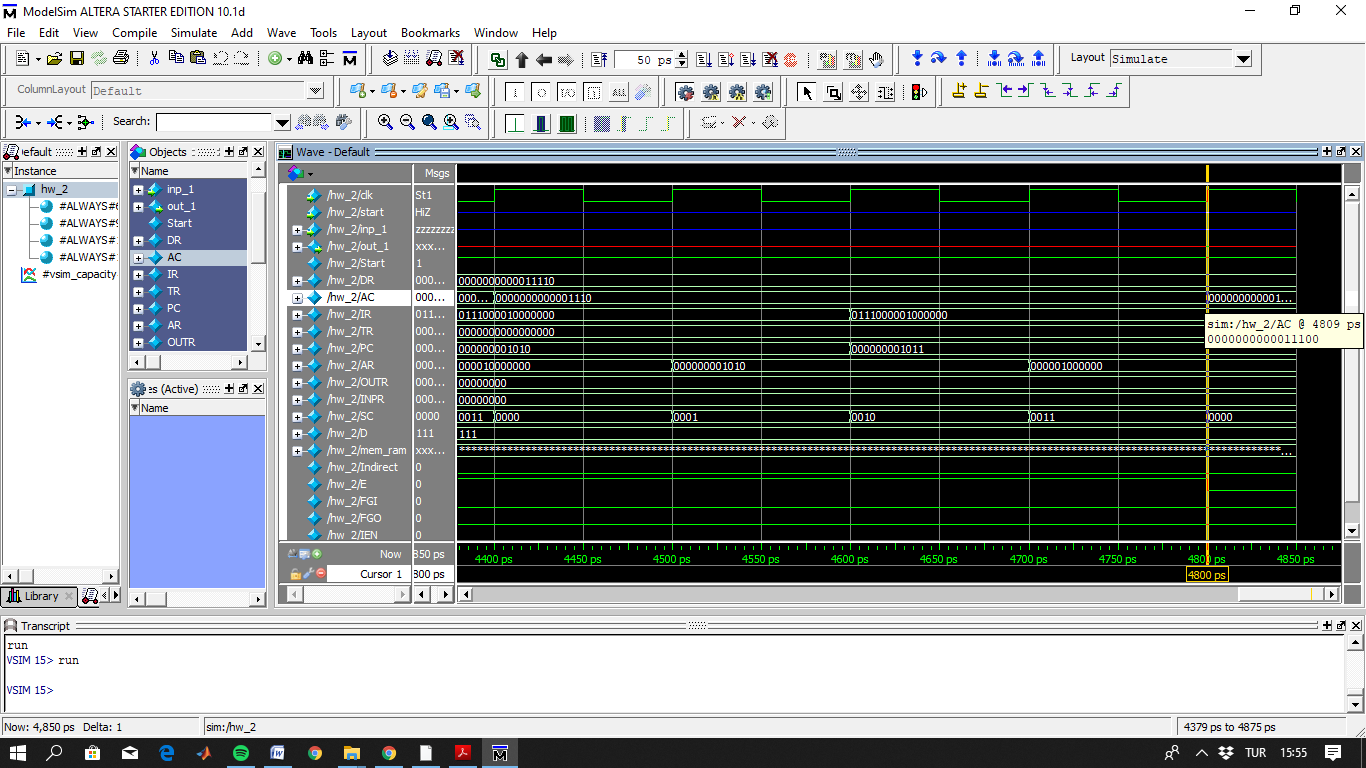


Figure 9: After circulating AC to the right (new value is 1110, first value was 11100 shown in figure 8)



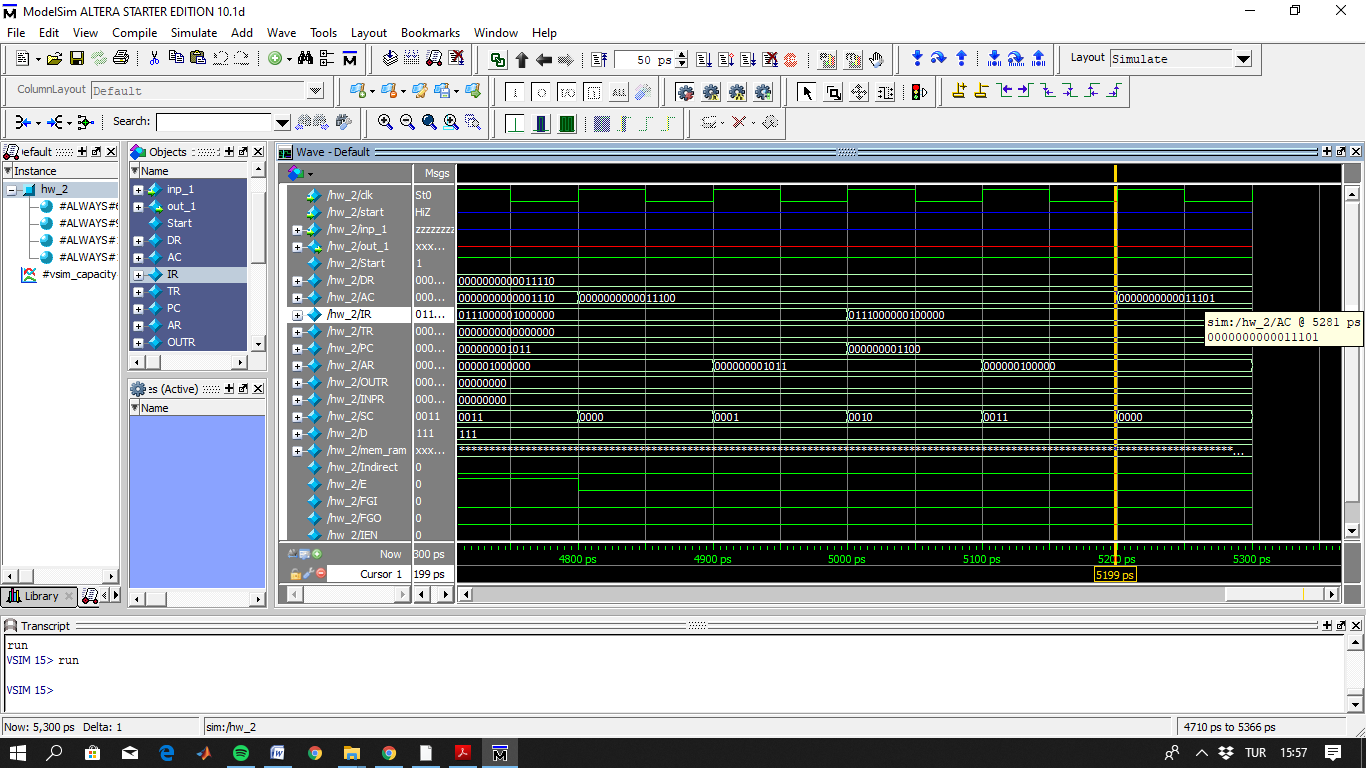
1. CIL operation: Hex code is h7040. It will circulate the AC to the left, and E will hold most significant bit of AC.

Figure 10: After circulate 1110 to the left, 11100 was observed as expected



1. INC operation: Hex code is h7020.

Figure 11: 11100 is incremented



1. SPA, SZA, SNA, SZE and HLT instructions are implemented in the main code, but since these are skip instructions, results will increment just program counter register. They are working well. One, who wants to test these micro-operations, can add related inputs to the memory. Afterwards, by providing shared hexadecimal codes to the memory, results can be observed.