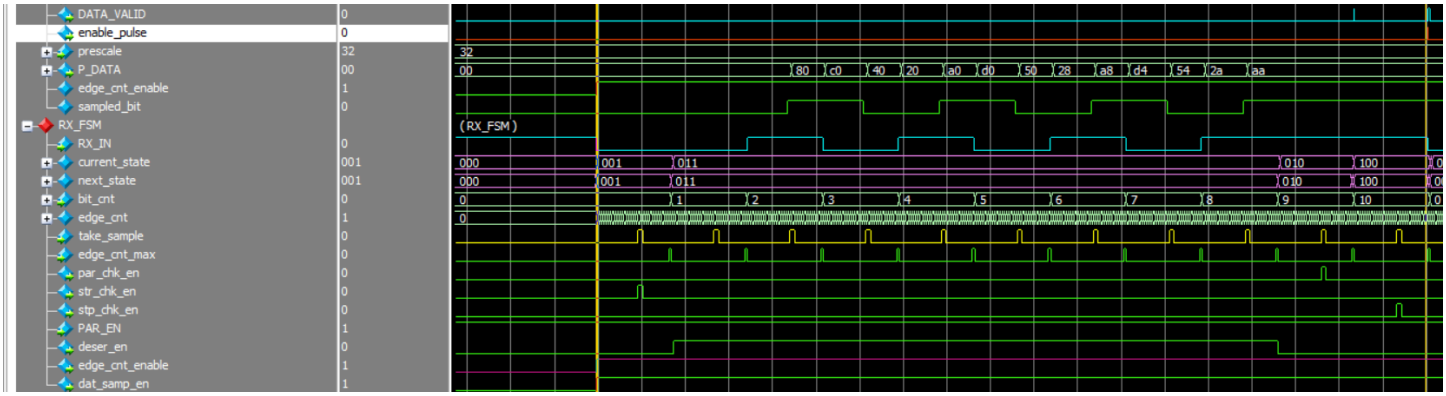
FINAL SYSTEM

System report

Hint: I added WrInc pulse generator to my system

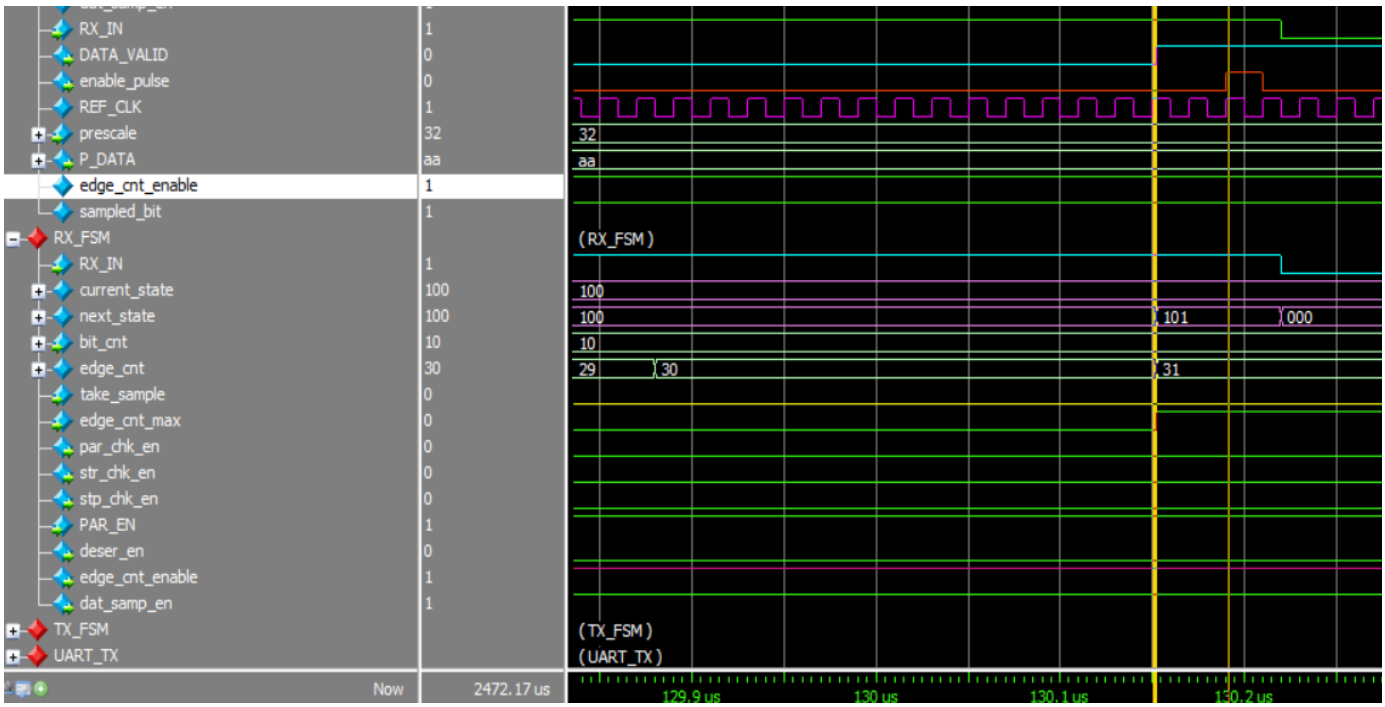
1) **Test Case 1:** testing writing 0xAA in address 0x44

1- UART_RX : receiving first command : 0xAA

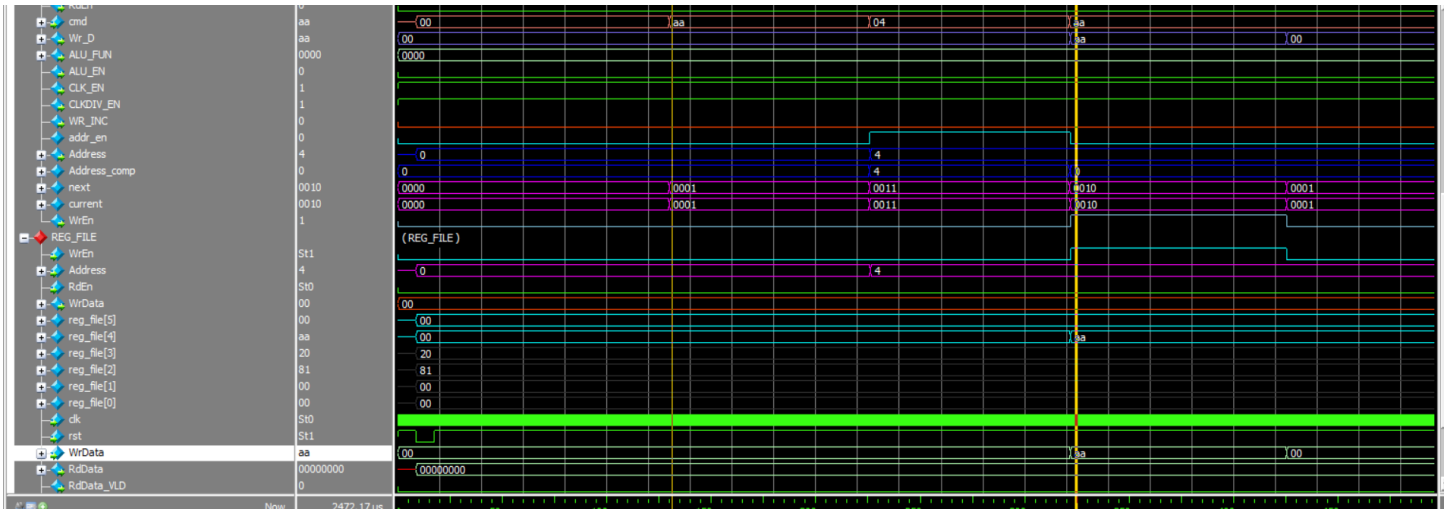


2- Data_valid pulse from UART_RX to SYS_CTRL :

Here we can see that enable_pulse came after Data_valid by 2 REF_CLK_CYCLES

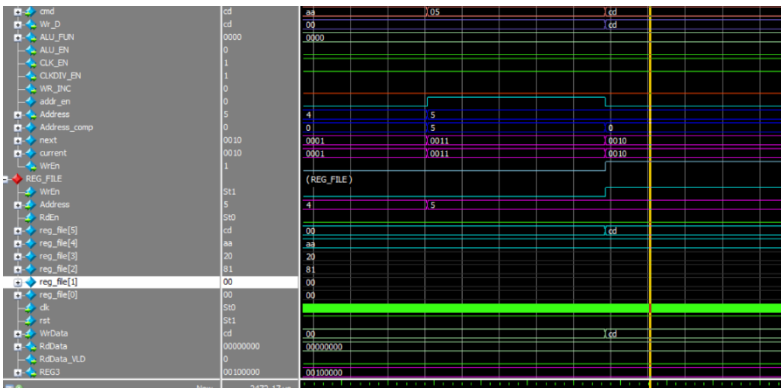


3- SYS_CTRL BLOCK → REG_FILE:
After decoding and many FSM states… output from SYS_CTRL to REG_FILE is show as:
SYS_CTRL FSM States:
0000→ idle
0001→ Wr_cmd
0011→wr_address
0010→ wr_data



That process were repeated again to write 0xCD in address 0x5

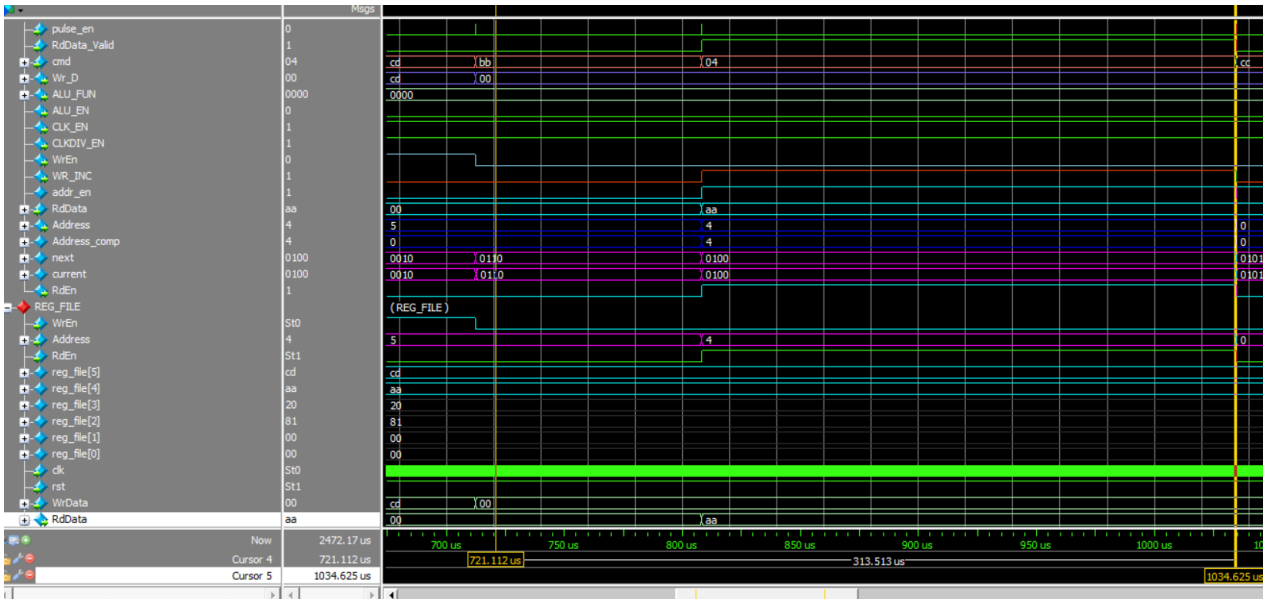
```
$display( "=============TEST CASE 1====================");
$display( "=============================================");
$display( "witing 0xAA in address 0x4");
#(TX_CLK_PERIOD);
DATA_IN('b11_1010_1010_0); // Write cmd = 0XAA
DATA_IN('b10_0000_0100_0); // Address = 0x4
DATA_IN('b11_1010_1010_0); // Write data = 0xAA

$display( "witing 0xAA in address 0x5");
#(TX_CLK_PERIOD);
DATA_IN('b11_1010_1010_0); // Write cmd = 0XAA
DATA_IN('b11_0000_0101_0); // Address = 0x5
DATA_IN('b10_1100_1101_0); // Write data = 0xCD
```



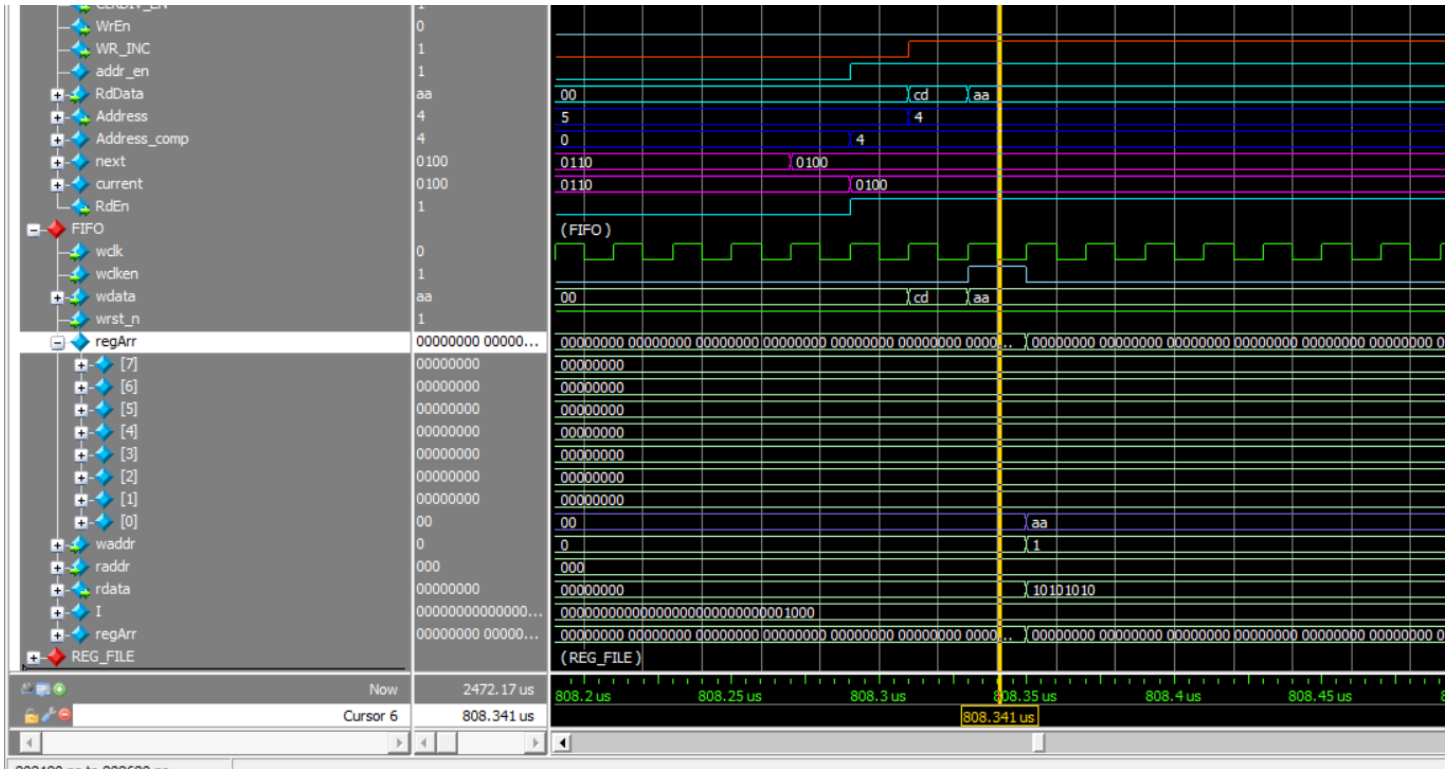
2)Test Case 2: test reading data were written before:

-Reading command received (0xbb) : FSM → 0110=rd_cmd state

-Then address received (0x4) : FSM → 0100=rd_data state that rise the RdEn and aallow us reading

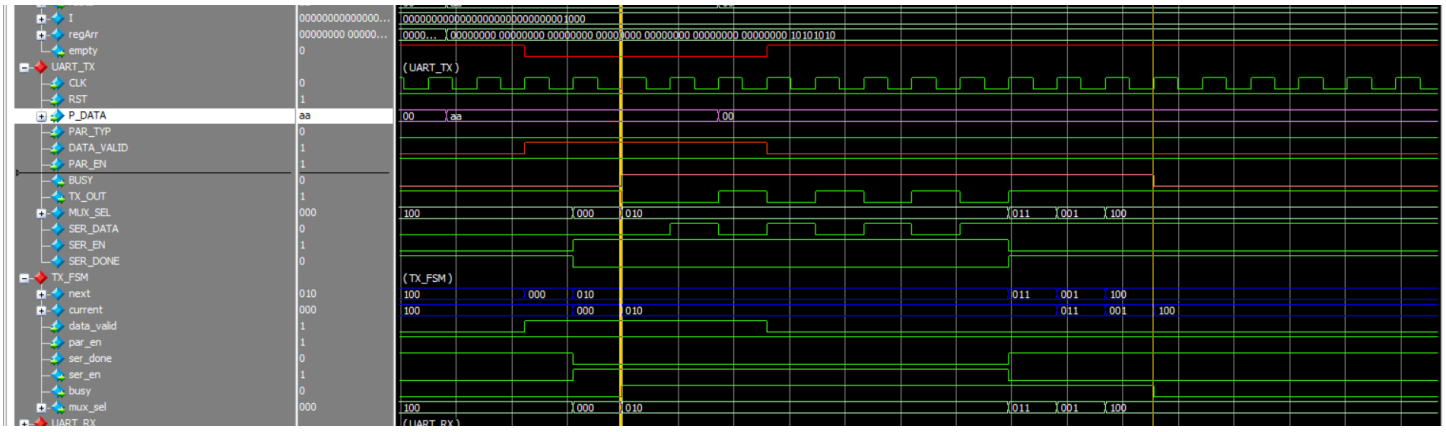


- Then read data go through SYS_CTRL to FIFO
- FSM:0100→ rd_data
- WrInc raised → WCLKEN had an enable pulse as I've added pulse generator from SYS_CTRL/WrInc to FIFO/WCLKEN
- 0xaa written in regArr[0]



-when embty flag down→ Data valid up after 2 UART_TX clock cylecs :

-P_Data in TX_serializer = 0xaa (required data to send)



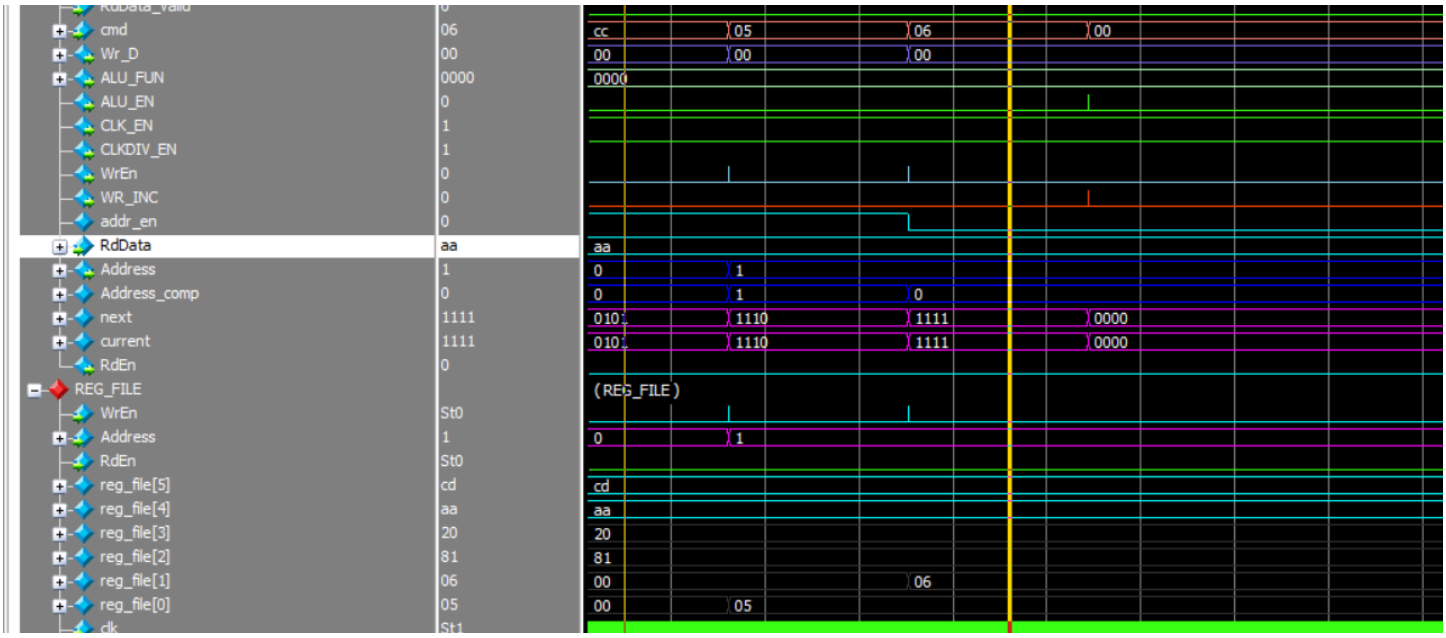
- Sent Data was correct:

```
#
#
# ================================================
# =============TEST CASE 1=======================
# ================================================
# witing 0xAA in address 0x4
#  Serial RX_IN = 11101010100
#  Serial RX_IN = 10000001000
#  Serial RX_IN = 11101010100
#  Serial RX_IN = 11101010100
#  Serial RX_IN = 11000001010
#  Serial RX_IN = 10110011010
# ================================================
# =============TEST CASE 2=======================
# ================================================
# Reading 0xAA that were written before from
#  Serial RX_IN = 11101110110
#  Serial RX_IN = 10000001000
# TX output = aa
# TEST 2 PASSED
# ================================================
```

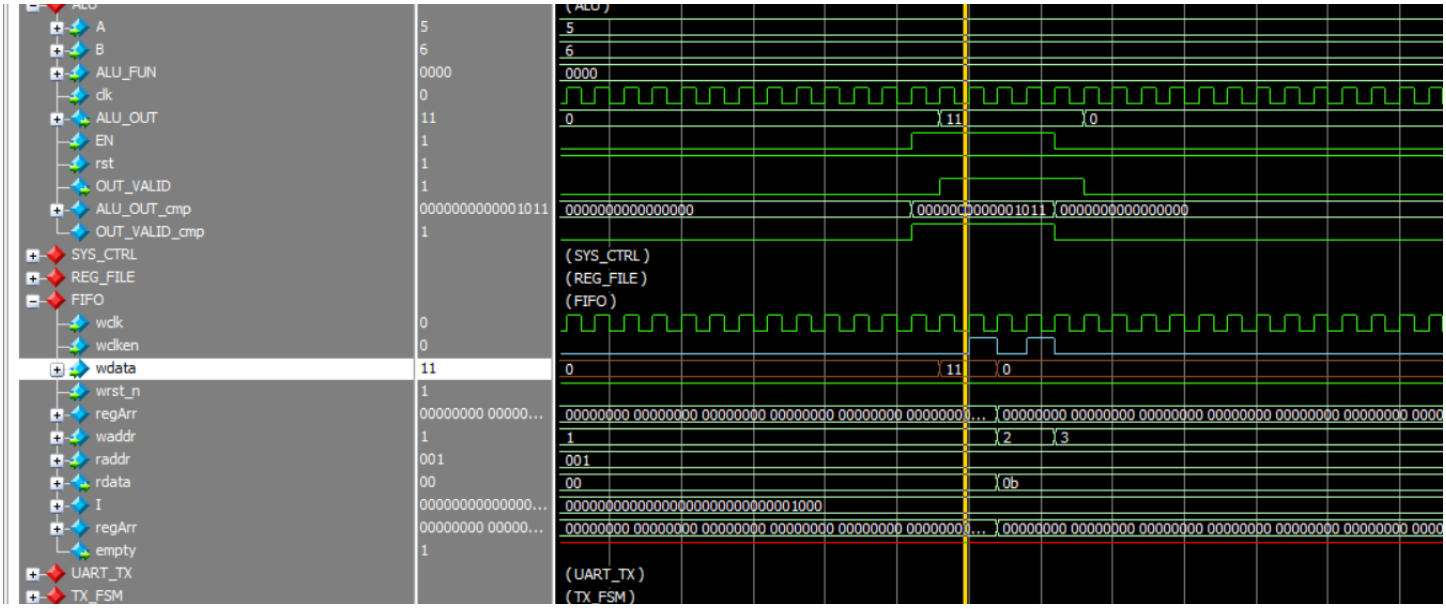
3) ALU_W_OP: command = 0xCC

- operand A = 5, B=6 , operation (pluse)

-As done before... same steps were made to write in regfile: regfile[0]=5 , regfile[1]=6



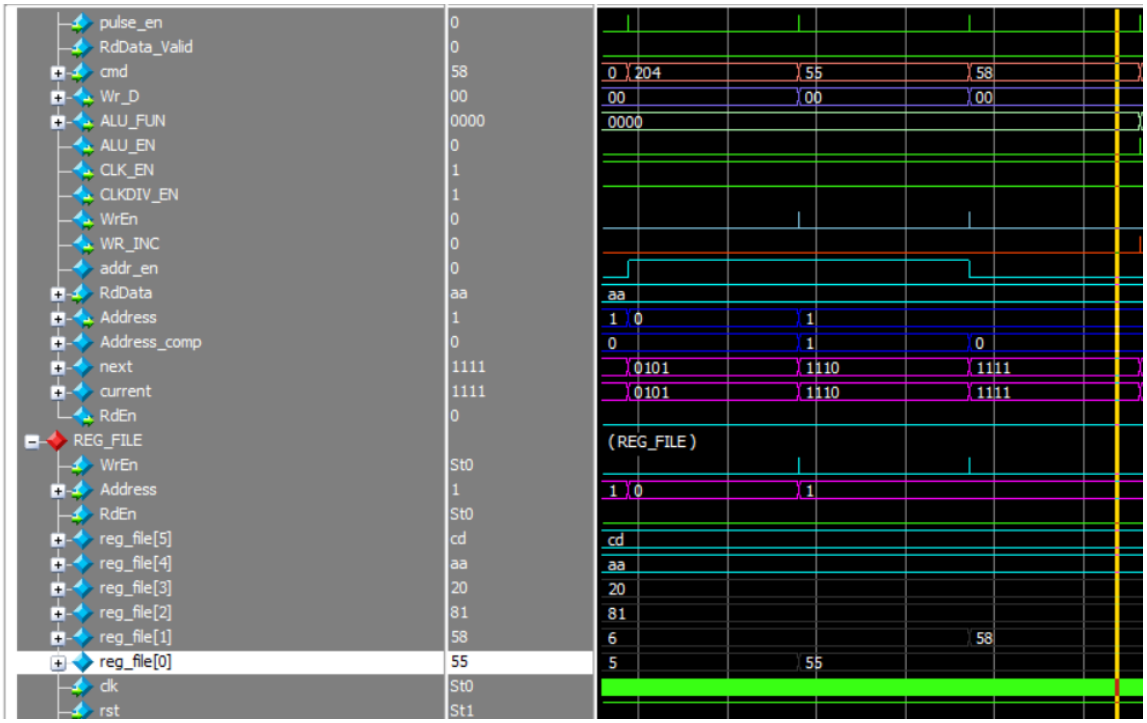
-after writing operands in regfile, ALU_FUN command were send and ALU_OPERATION start:



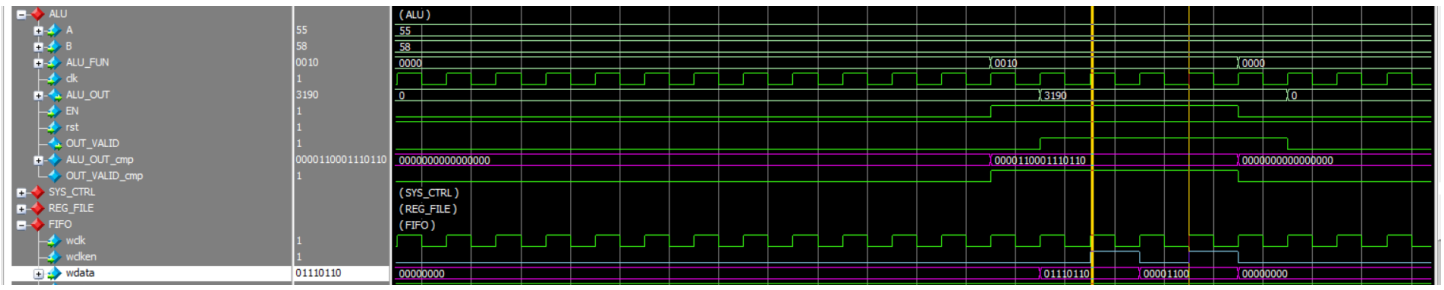
- AlU_OUT sent to FIFO in two frames and then to UART_TX
- But, to test sending the ALU_OUT 16-bits: I made another test case that multiply 55*58

```
////////////////////////////
//////// TEST 4 ////////
////////////////////////////
// Testing ALU operation with operand
// A = 55
// B = 58
$display( "=========================================");
$display( "============TEST CASE 4==================");
$display( "=========================================");
$display( "ALU_W_OP: A= 55 , B=58 , multiplication operation ");
DATA_IN('b11_1100_1100_0); // ALU_W_OP = 0xCC
DATA_IN('b10_0011_0111_0); // A = 55
DATA_IN('b11_0011_1010_0); // B=58
DATA_IN('b10_0000_0010_0); // multiplication

out_check_with_parity('b0111_0110,4);
out_check_with_parity('b1100,4);
```

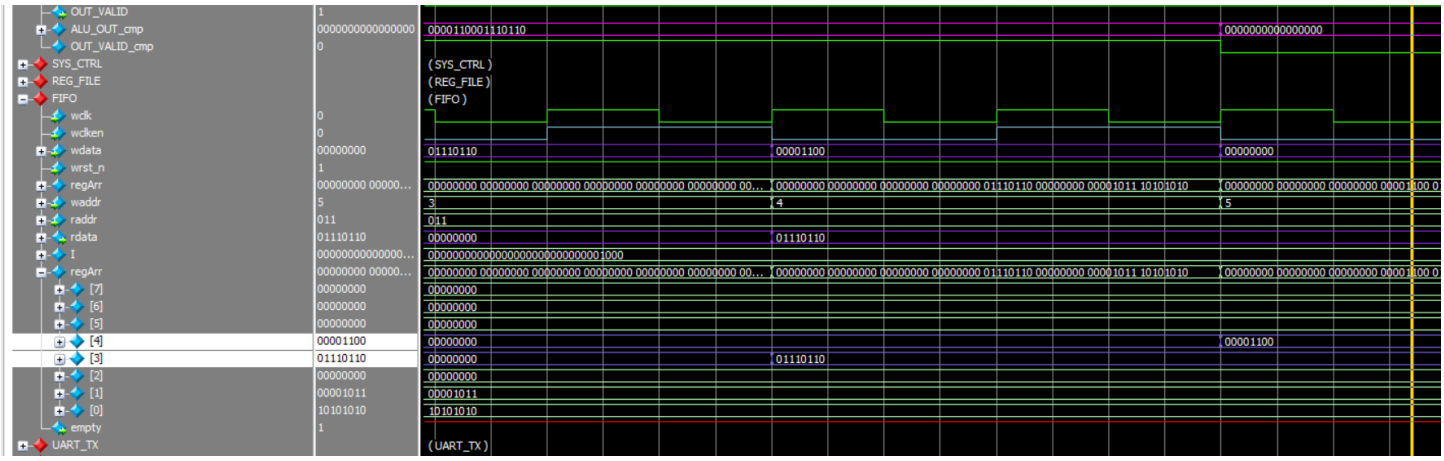
-



-ALU:



- Here you can see WCLEN raised twice and the 16-bit frame send in two 8-bit concatenated frames
- Those frames were written successfully in FIFO as shown in next figure:



-then those two frames arrive to UART_TX and sent successfully to TOP level

-Transept results :

```
# ================================================
# =============TEST CASE 3=======================
# ================================================
# ALU_W_OP: A= 5 , B=6 , pluse operation
#  Serial RX_IN = 11110011000
#  Serial RX_IN = 11000001010
#  Serial RX_IN = 11000001100
#  Serial RX_IN = 11000000000
# TX output = b
# TEST 3 PASSED
# TX output = 0
# TEST 3 PASSED
# ================================================
# =============TEST CASE 4=======================
# ================================================
# ALU_W_OP: A= 55 , B=58 , multiplication operation
#  Serial RX_IN = 11110011000
#  Serial RX_IN = 10001101110
#  Serial RX_IN = 11001110100
#  Serial RX_IN = 10000000100
# TX output = 76
# TEST 4 PASSED
# TX output = c
# TEST 4 PASSED
```

4)ALU operation without operand: Command: 0xDD

-this test where made and passed successfully

-system operations were done

-Transept results:

```
# ================================================
# =============TEST CASE 5=======================
# ================================================
# ALU_W_OP: A= 55 , B=58 , pluse operation
#  Serial RX_IN = 11110111010
#  Serial RX_IN = 11000000000
# TX output = 71
# TEST 5 PASSED
```

```
#
# ===========================================
# ============TEST CASE 1====================
# ===========================================
# witing 0xAA in address 0x4
#  Serial RX_IN = 11101010100
#  Serial RX_IN = 10000001000
#  Serial RX_IN = 11101010100
# witing 0xAA in address 0x5
#  Serial RX_IN = 11101010100
#  Serial RX_IN = 11000001010
#  Serial RX_IN = 10110011010
# ===========================================
# ============TEST CASE 2====================
# ===========================================
# Reading 0xAA that were written before from
#  Serial RX_IN = 11101110110
#  Serial RX_IN = 10000001000
# TX output = aa
# TEST 2 PASSED
# ===========================================
# ============TEST CASE 3====================
# ===========================================
# ALU_W_OP: A= 5 , B=6 , pluse operation
#  Serial RX_IN = 11110011000
#  Serial RX_IN = 11000001010
#  Serial RX_IN = 11000001100
#  Serial RX_IN = 11000000000
# TX output = b
# TEST 3 PASSED
# TX output = 0
# TEST 3 PASSED
# ===========================================
# ============TEST CASE 4====================
# ===========================================
# ALU_W_OP: A= 55 , B=58 , multiplication operation
#  Serial RX_IN = 11110011000
#  Serial RX_IN = 10001101110
#  Serial RX_IN = 11001110100
#  Serial RX_IN = 10000000100
# TX output = 76
# TEST 4 PASSED
# TX output = c
# TEST 4 PASSED
# ===========================================
```

```
# ===========================================
# ============TEST CASE 5====================
# ===========================================
# ALU_W_OP: A= 55 , B=58 , pluse operation
#  Serial RX_IN = 11110111010
#  Serial RX_IN = 11000000000
# TX output = 71
# TEST 5 PASSED
# ===========================================
# ============ All Test cases done ============
# ===========================================
```