DIGITAL SYSTEM DESIGN APPLICATION FINAL PROJECT

ÖMER FARUK SERT
040170244
İBRAHİM SAMED YALÇIN
040180055

CONTENTS

INTRODUCTION

SUBJECT: Calculate C =A/B where A is a 8-bit positive integer and B is a 4-bit positive integer. There will be two 8-bit positive integer outputs as quotient, Q, and remainder, R.

PROBLEM SOLUTION/ PREPARATION

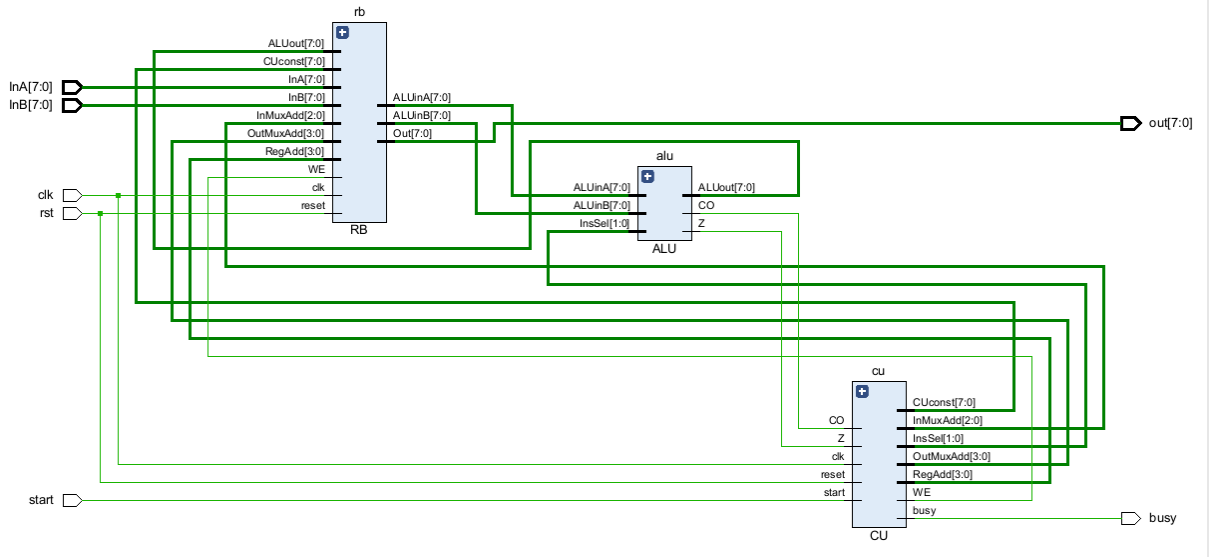
If we make the divider 2s complement and add it to the divident and count the number of addition then we found quotient and the what is left which can not be added to the dividers 2s complement is the remainder. In comments about results section, the logic we used is defined clearer.

ALGORTITHMIC STATE MACHINE

TOP MODULE

Verilog code was written in accordance with the schema of the given TOP file and the previously written reegister block, control unit, arithmetic and logic unit blocks were added and the RTL diagram of the figure is shown in the figure. Written based on TOP module in testbench and its results are shown

Top module RTL diagram

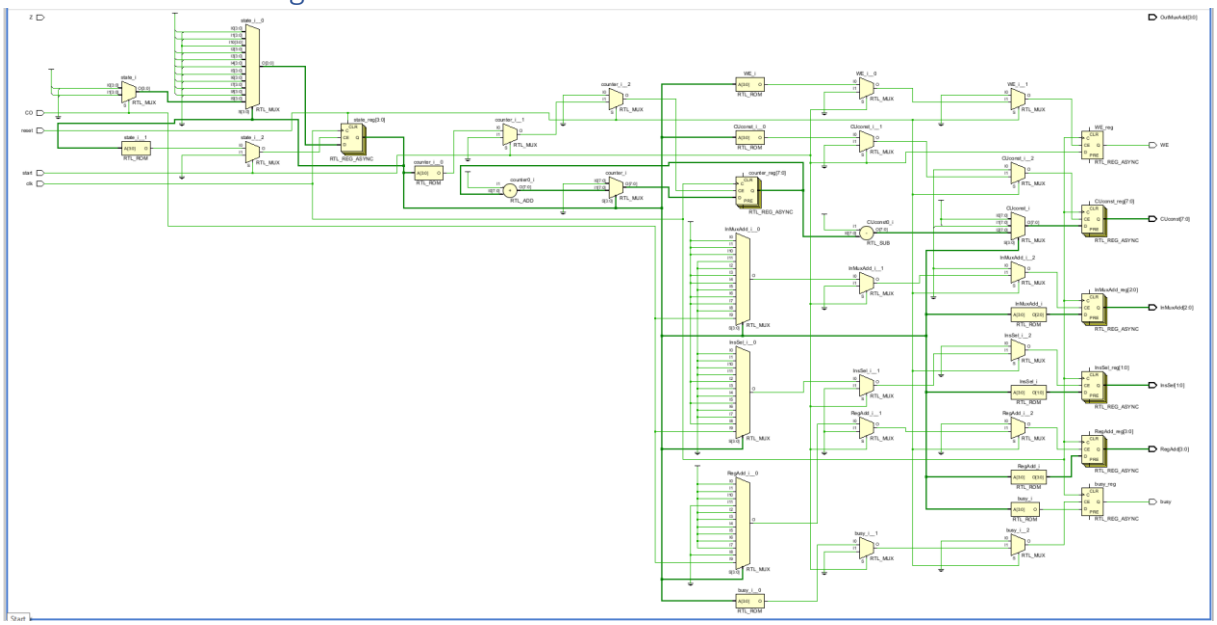


CONTROL UNIT

The control unit (CU) is a component of a computer's central processing unit (CPU) that directs the operations of the processor. It fetches instructions from memory, decodes them to determine the operation to be performed, and then coordinates the execution of that operation by activating the appropriate circuits within the CPU. The control unit acts as the "brain" of the CPU, constantly monitoring the state of the computer and making decisions about what actions to take next. It also manages the flow of data between the various components of the system, including the memory, the input-output devices, and the

arithmetic logic unit (ALU). The control unit reads the instruction from memory, interprets it and then generates the control signal that tells the other parts of the CPU what to do next

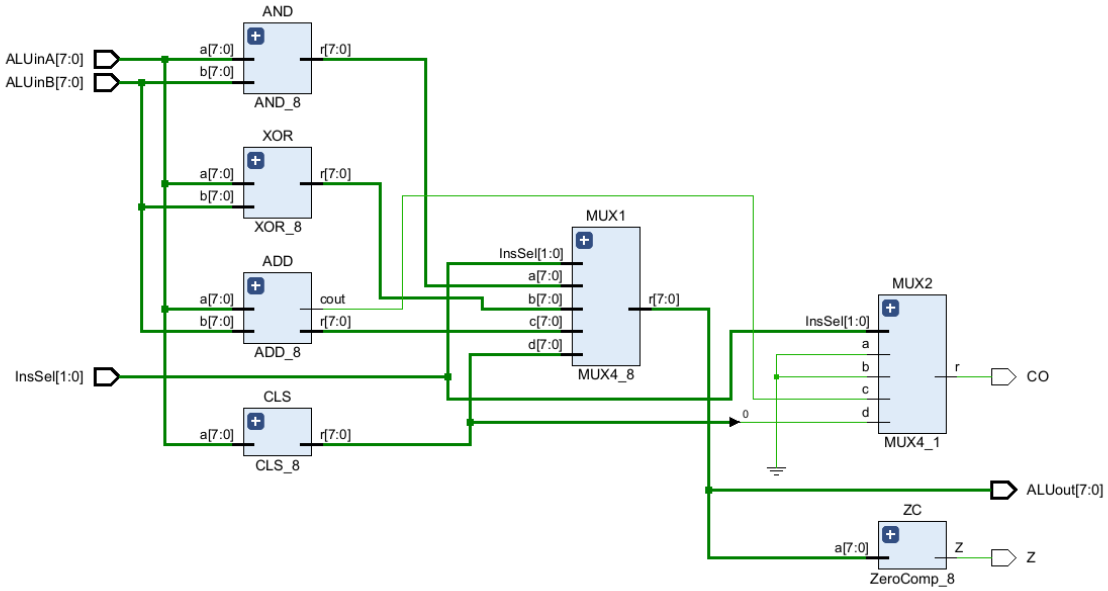
Control Unit RTL diagram



ALU

The arithmetic logic unit (ALU) is a fundamental component of a computer's central processing unit (CPU) that performs arithmetic and logical operations. The ALU is responsible for carrying out operations such as addition, subtraction, multiplication, division, and logical operations such as AND, OR, and NOT. The results of these operations are then stored in registers or memory. The ALU is typically made up of digital circuits that can perform basic mathematical operations, as well as circuits that can compare numbers and make logical decisions. The ALU is connected to the control unit, which sends it the instructions it needs to execute and then directs the ALU to send the results back to the registers or memory.

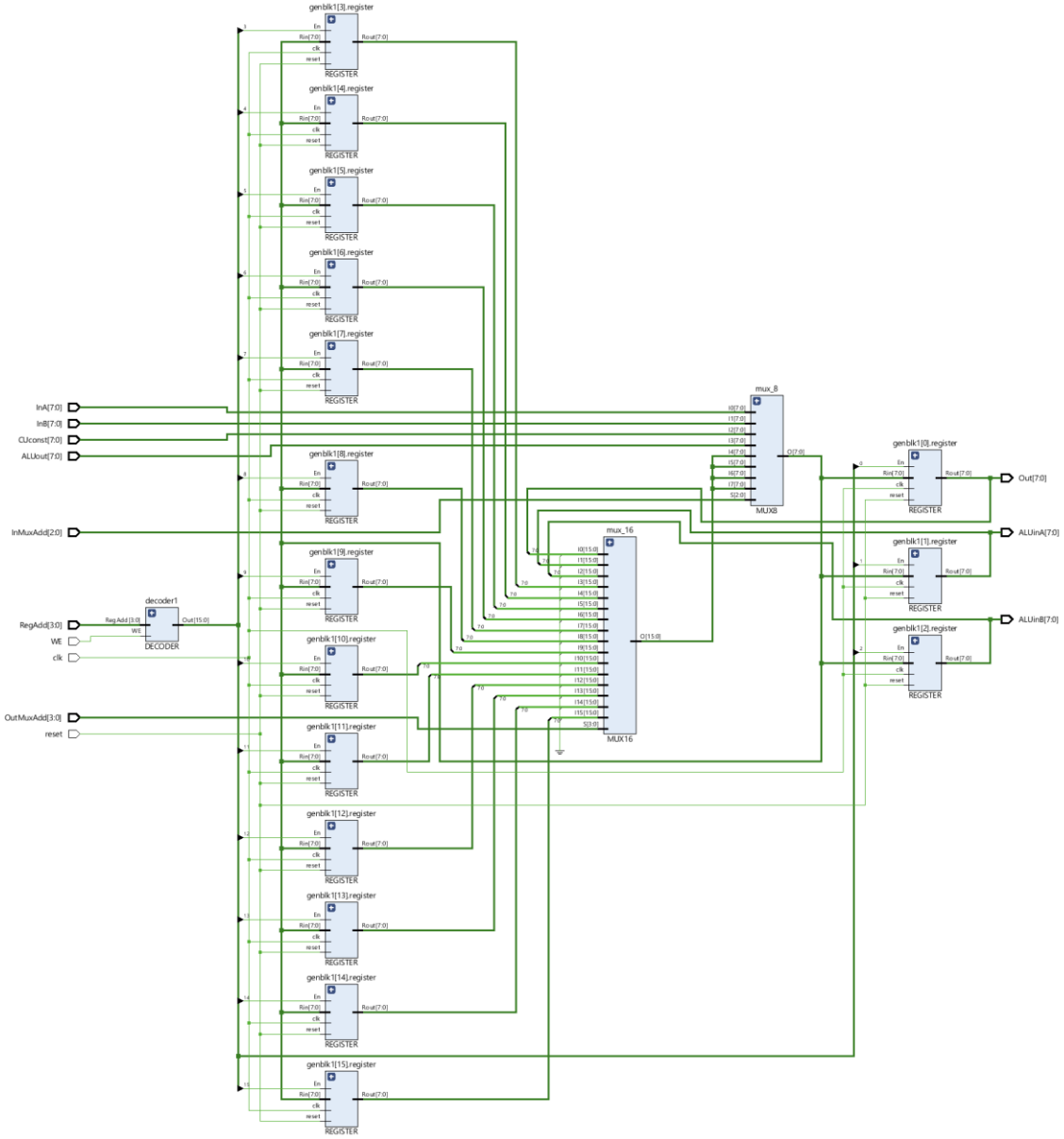
ALU RTL diagram



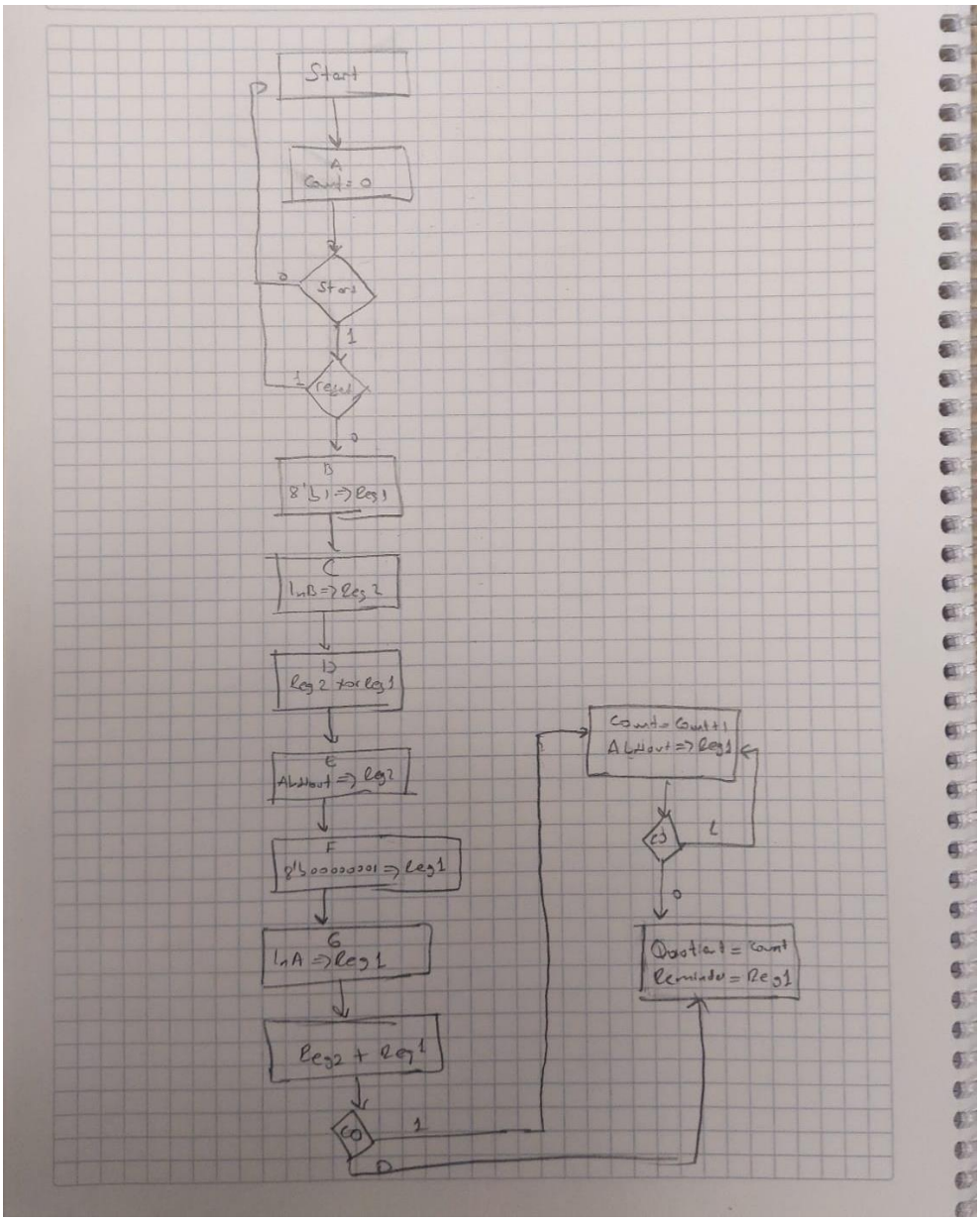
REGISTER BLOCK

A register block is a section of memory in a computer or other digital device that is used to store data that is frequently accessed or manipulated by the processor. Registers are typically located within the processor itself, and are used to store information such as the current instruction being executed, the current value of a particular calculation, or the location of data in memory. Because registers are located within the processor and can be accessed much more quickly than other types of memory, they are often used to speed up the execution of programs and improve overall system performance.

Register Block module RTL diagram

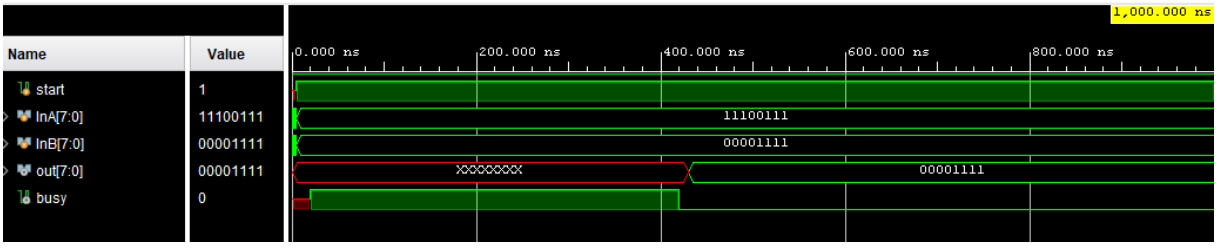

genblk1[3].register
genblk1[4].register
genblk1[5].register
genblk1[6].register
genblk1[7].register
genblk1[8].register
genblk1[9].register
genblk1[10].register
genblk1[11].register
genblk1[12].register
genblk1[13].register
genblk1[14].register
genblk1[15].register
REGISTER
InA[7:0]
InB[7:0]
CUconst[7:0]
ALUout[7:0]
InMuxAdd[2:0]
RegAdd[3:0]
WE
clk
OutMuxAdd[3:0]
reset
decoder1
DECODER
mux_8
MUX8
mux_16
MUX16
genblk1[0].register
genblk1[1].register
genblk1[2].register
Out[7:0]
ALUinA[7:0]
ALUinB[7:0]

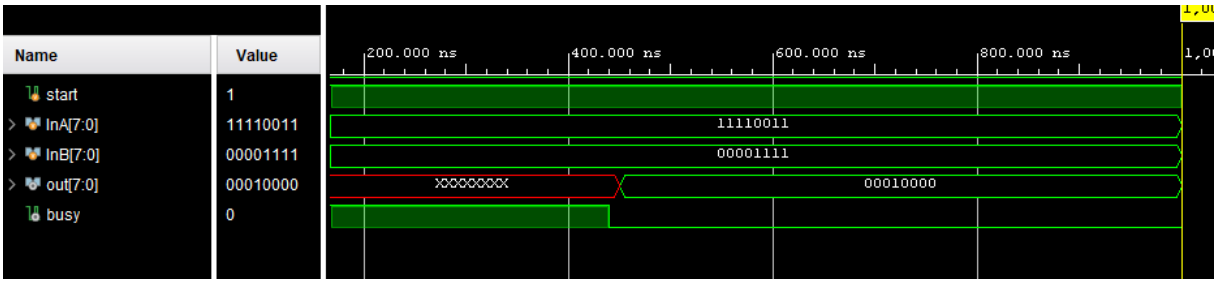
ASM

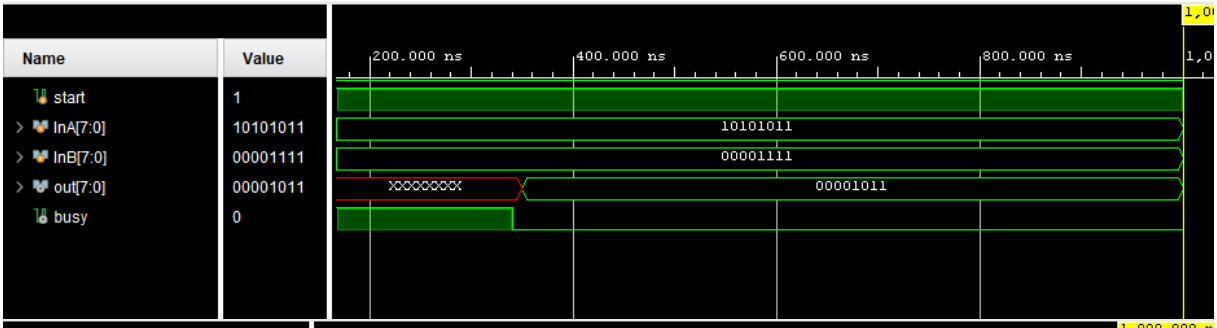


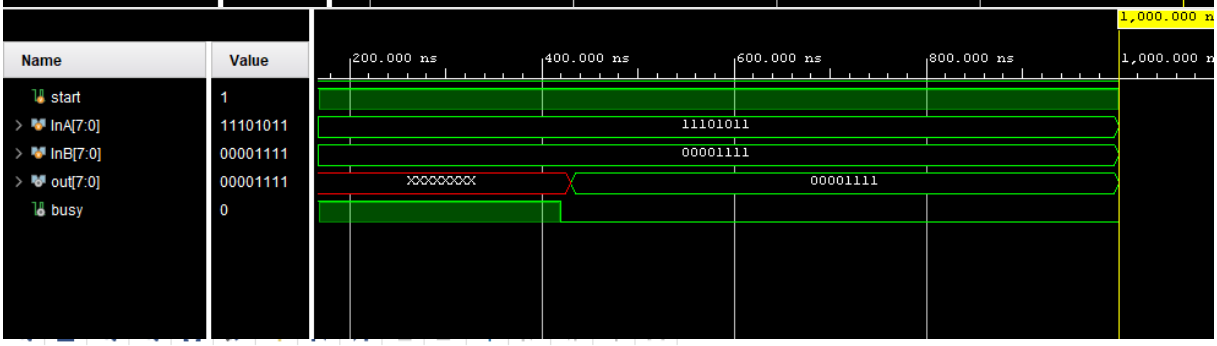
BEHAVIORAL SIMULATION

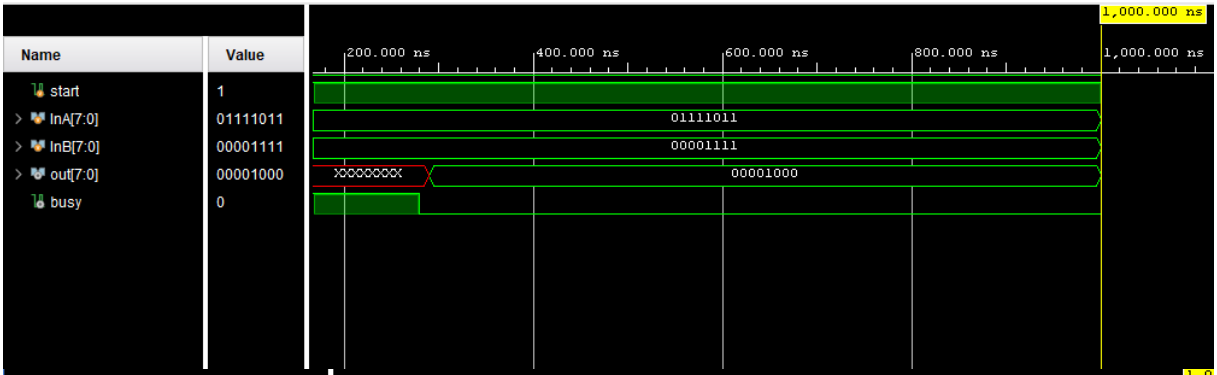
Wave form

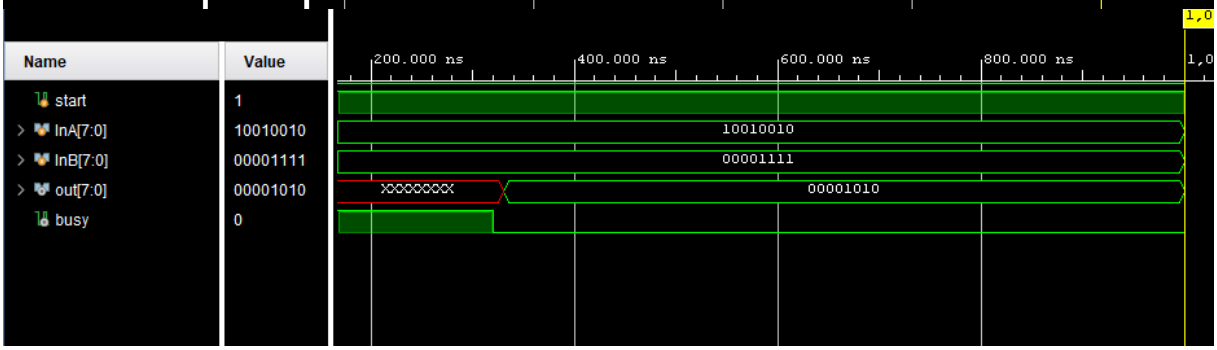


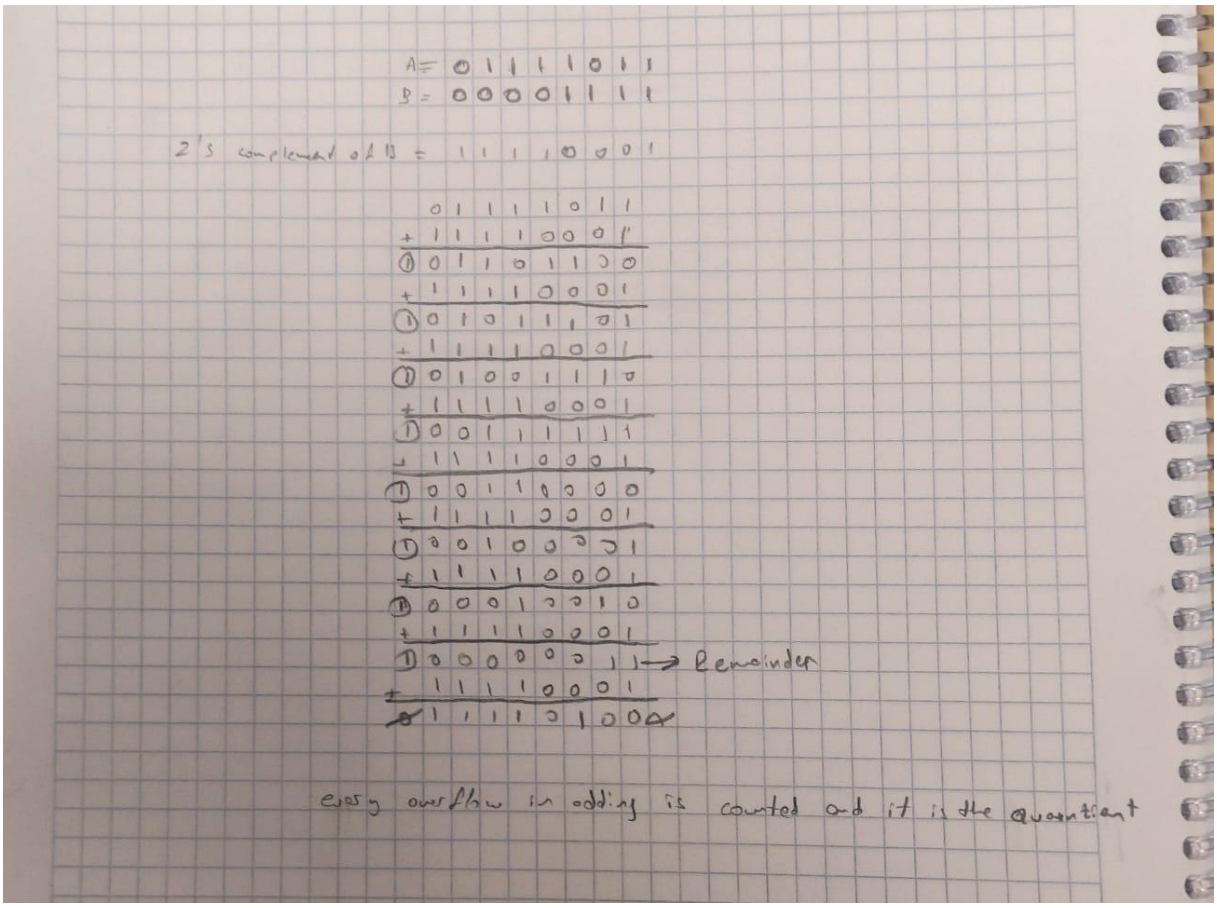

Name
Value
start
1
InA[7:0]
11110011
InB[7:0]
00001111
out[7:0]
00010000
busy
0
200.000 ns
400.000 ns
600.000 ns
800.000 ns
11110011
00001111
XXXXXXXX
00010000


Name
Value
start
1
InA[7:0]
10101011
InB[7:0]
00001111
out[7:0]
00001011
busy
0
200.000 ns
400.000 ns
600.000 ns
800.000 ns
10101011
00001111
XXXXXXXX
00001011


Name
Value
start
1
InA[7:0]
11101011
InB[7:0]
00001111
out[7:0]
00001111
busy
0
200.000 ns
400.000 ns
600.000 ns
800.000 ns
1,000.000 ns
11101011
00001111
XXXXXXXX
00001111


Name
Value
start
1
InA[7:0]
01111011
InB[7:0]
00001111
out[7:0]
00001000
busy
0
200.000 ns
400.000 ns
600.000 ns
800.000 ns
1,000.000 ns
01111011
00001111
XXXXXXXX
00001000


Name
Value
start
1
InA[7:0]
10010010
InB[7:0]
00001111
out[7:0]
00001010
busy
0
200.000 ns
400.000 ns
600.000 ns
800.000 ns
10010010
00001111
XXXXXXXX
00001010



Comment

This finite state machine works correctly but there is some situations that it shows one more then it should result. Remainder is stored in register1 but it is one step further so . I noticed too late that there should not be any register in control unit except state and output registers but it can be stored in register3 in register block instead of counter register in control block. The last wave form shows the incorrect result that ot results one more then it should be. After the wave i showed a hand calculated division logic that i used in this circuit.

TASK DISTRIBUTION

ÖMER FARUK SERT	İBRAHİM SAMED YALÇIN
SOLUTION AND ASM	WRITE TOP MODULE
WRITE CONTROL UNIT	WRITE TESTBENCH
WRITE ALU	WRITE REGISTER BLOCK
PREPARE REPORT	PREPARE REPOT