CS2310

VERILOG LAB-4 (CPU DESIGN)

Submitted by: **Kumar Kshitiz Singh**
CS21B044

Introduction

Simulating a CPU (CU+ALU) using verilog code by sending 3 part input bus (3-bit instruction, 8-bit operand1, 8-bit operand2), and receiving an 8-bit bus as output.

Problem overview

Input- I [18:0]

Output- o [7:0]

First 3 bits(I [18:16]) consists of the instruction bits, which commands the type of operation to perform.

Mapping:

001 - ADD, 010 - SUBTRACT, 011 - INCREMENT, 100 - DECREMENT

101 - BITWISE AND, 110 - BITWISE OR, 111 - NOT

Operands:

Operand1 - I [15:8]

Operand2 - I [7:0]

APPROACH

CU (Control Unit)

The control unit module takes in the input and decodes the instruction bits, and passes the operands to relevant operator modules to give the final output.

The output is calculated by taking a bitwise AND of the computed result with the respective decoder output bit, so that only the desired result remains, and the rest become 0.

The final output can be computed by taking Bitwise OR of all the new (post-and) results.

DECODER MODULE

Takes in a 3-bit input and returns an 8-bit bus output, with only one of the bits being High.

ADD MODULE

Receives 2 8-bit inputs and calculates their sum by calling the 8-bit ripple carry adder module. Returns the output.

SUBTRACT MODULE

Modifies the second operand to its 2's complement, then calls the RCA module to add the two operands.

INCREMENT MODULE

Takes in the first operand and ignores the second. Increments the first operand by 1, by calling the RCA module and adding 1.

DECREMENT MODULE

Takes in the first operand and ignores the second. Decrements the first operand by 1, by calling the RCA module and adding 11111111.

AND MODULE

Receives 2 8-bit inputs and calculates their bitwise AND and returns the output.

OR MODULE

Receives 2 8-bit inputs and calculates their bitwise ORand returns the output.

NOT MODULE

Takes in the first operand and ignores the second. Takes the not of each bit and returns the 8-bit bus.

TESTBENCH

GIVEN INPUTS:

001-00100011-00010100

010-00100011-00010100

011-00100011-00010100

100-00100011-00010100

101-00100011-00010100

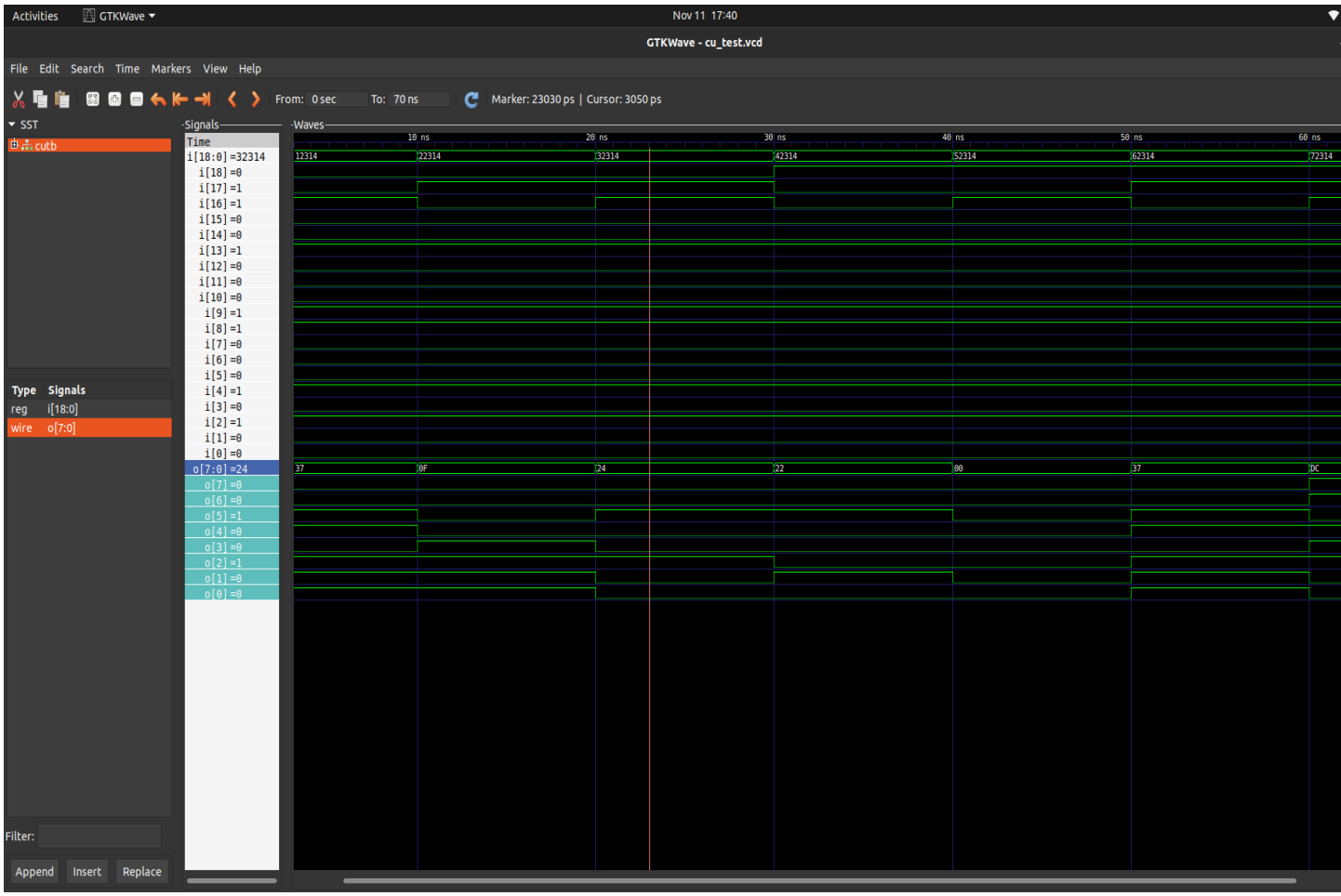
110-00100011-00010100

111-00100011-00010100

OUTPUT STATUS: Matching

GTKWAVE SCREENSHOT:

https://drive.google.com/file/d/1tloBtzCgHzfBRyNqKpISSLhBQoeJBIRg/view?usp=share_link


Activities
GTKWave
Nov 11 17:40
GTKWave - cu_test.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 70 ns
Marker: 23030 ps | Cursor: 3050 ps
SST
cutb
Type Signals
reg i[18:0]
wire o[7:0]
Filter:
Append
Insert
Replace
Signals
Waves
Time
i[18:0] =32314
o[7:0] =24
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
12314
22314
32314
42314
52314
62314
72314
37
0F
24
22
00
37
DC

THANK YOU :)