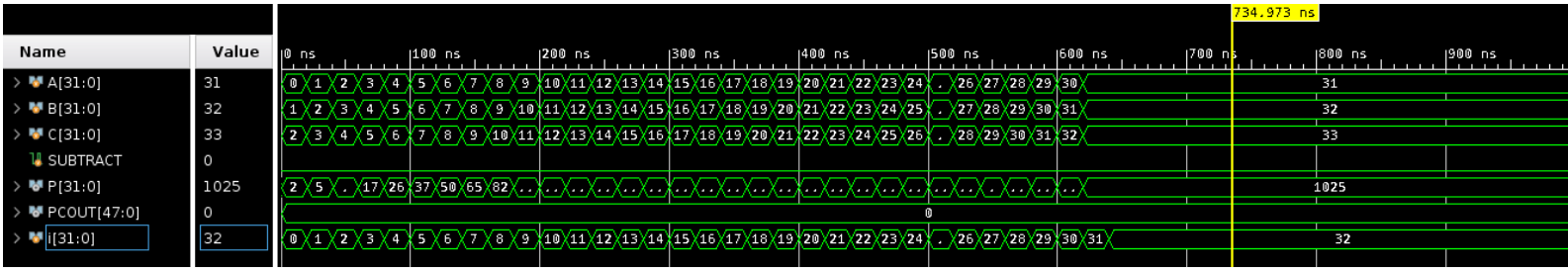
Alihan Mahanoglu 2019-81292 2019.04.03

Janghoon Han 2008-12407

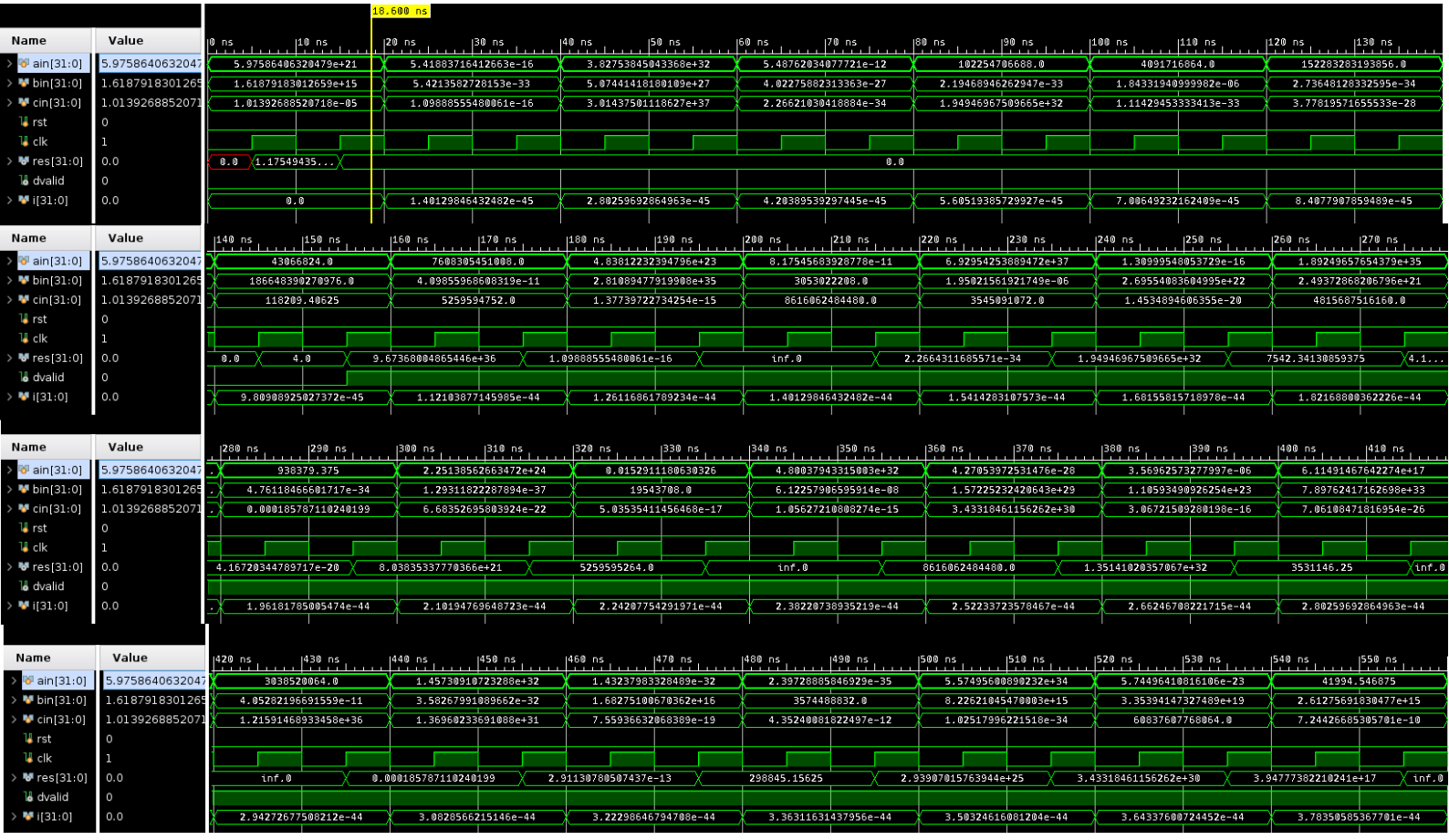
Hardware System Design Lab#4

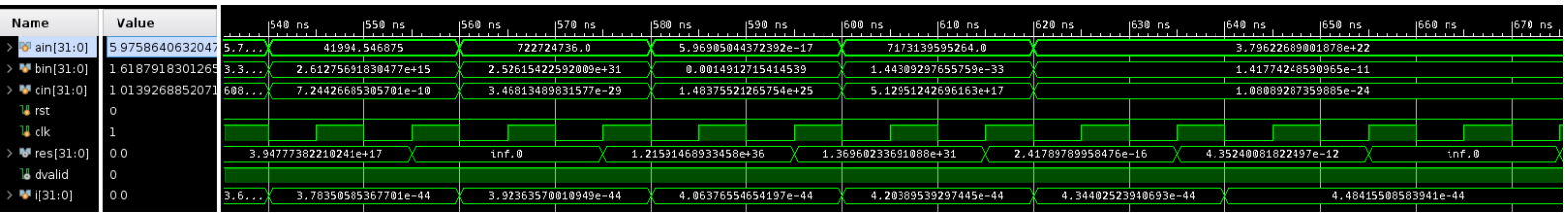
- Test Bench Simulation Result for 32Bit Integer Multiply Adder



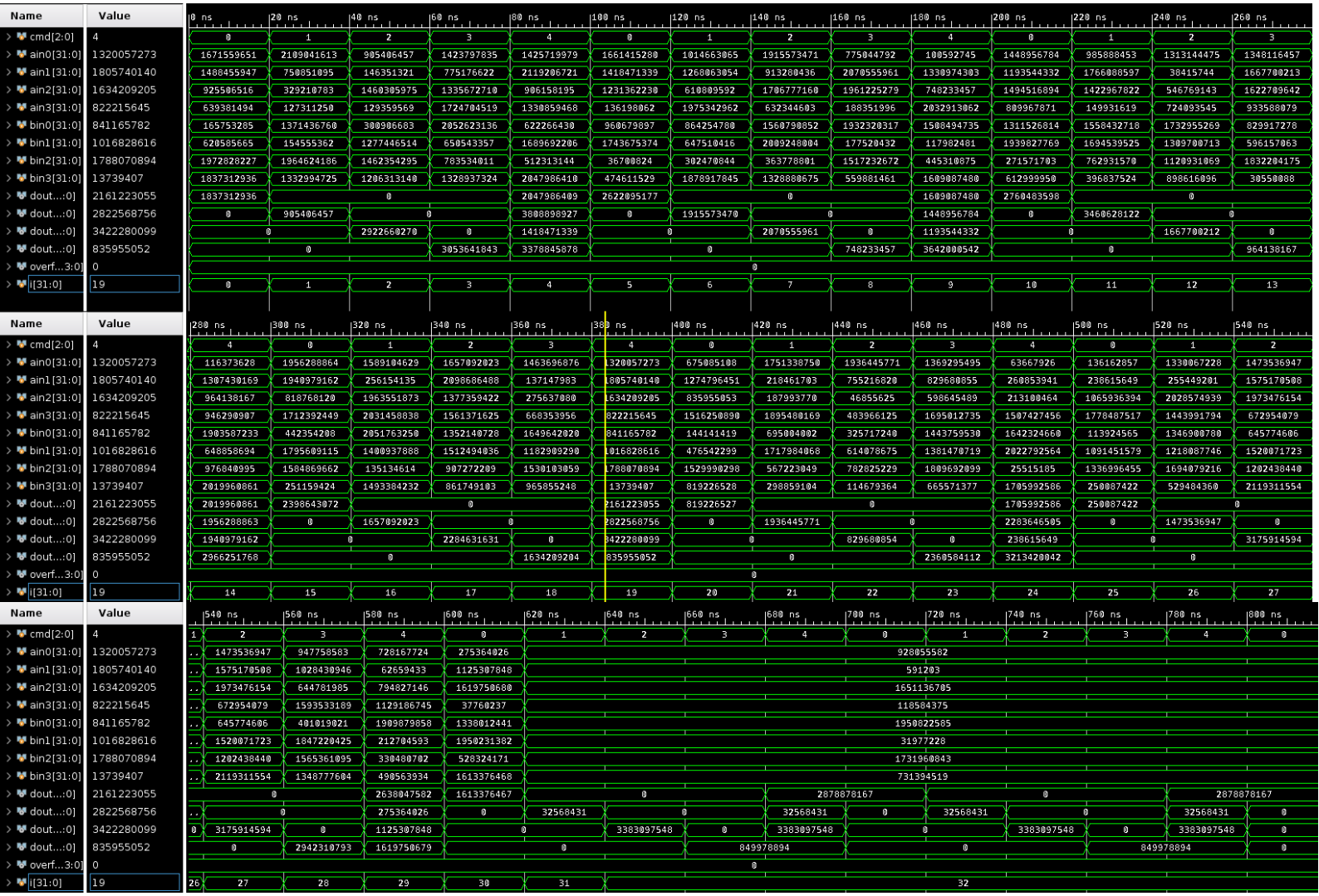
Please note that the values given for A, B and C are changed (assigned as i, i+, i+2 in 32 step for loop) from random assignment because it was easier to check.

- Test Bench Simulation Result for 32Bit Floating Point Multiply Adder





- Test Bench Simulation Result for Adder Array



Code explanations

As it was already provided in the 4th lab manual pdf, “floating point” was selected from IP catalog with given settings. In the manual, the code for testbench was also provided. Therefore for the 32Bit floating point multiply adder nothing much was done else than checking the testbench result of the module. Please note that there is around 140 ns delay in the test bench result.

For 32Bit integer multiply adder provided test bench code was modified in a way that it will work the multiply adder module for integers. At the first run of simulation it was seen that there was an internal delay, so module was reconfigured to not include any delay.

For adder array, an adder module was created the same way it was done in previous lab.

```
module adder(ain, bin, dout, overflow);
    input [31:0] ain;
    input [31:0] bin;
    output [31:0] dout;
    output overflow;

    assign {overflow, dout} = ain+bin;
endmodule
```

Image 1 The Code for adder module

In order to tie up the inputs in the adder array to the outputs an intermediate my_ain, my_bin and my_dout variables are initiated as 4 pairs of 32bit wires (for each ain, bin inputs and dout outputs). Then these variables were connected to actual input and output of adder array with assign function.

The process of assigning “dout”s to “my_dout”s were dependent on cmd, therefore conditional operator as in PowerPoint presentation of 3rd week 42th slide.

After using the generate function 4 different adder modules created with the intermediate variables (my_ain, my_bin, my_dout).