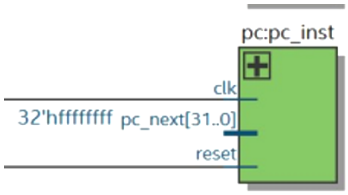
#ITERATION 2

In iteration 1 all modules were created keeping in mind the flow of data and paths through them, although as an individual module their working was as expected, we came across some critical errors while synthesizing the design.
The main reason for such failures was the way synthesizing tools look at different modules and connections. Our iteration 1 top module lacked this critical point.
Though functionally correct, the Iteration 1 module is not readily synthesizable. Hence the aim of this module is to- understand , document and correct these details so that iteration 2 is a completely synthesizable RISC one cycle processor. We will also add waveform outputs and timing diagrams of synthesized RISC at the end along with the RTL and other reports.

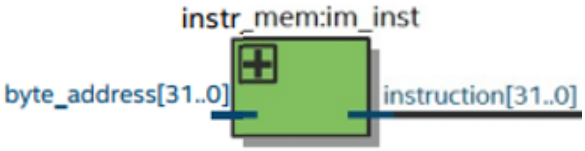
Let's start with errors and their analysis, ill analyse each module-

PC-
3 inputs as expected- clk, pc_next and reset.
Error- no output reg PC [31:0];
This reg PC was supposed to be an input to instruction memory where it would be named as input byte address.
Solution--It appears that we did not connect them by a wire so the synthesis tool can not identify this connection. Although the input appears as expected at instruction memory, we might have to define this connection explicitly in the top module.
Correction1- mention output pc wire and assign to byte_address
instr_mem im_inst(.byte_address(pc), .instruction(instruction));

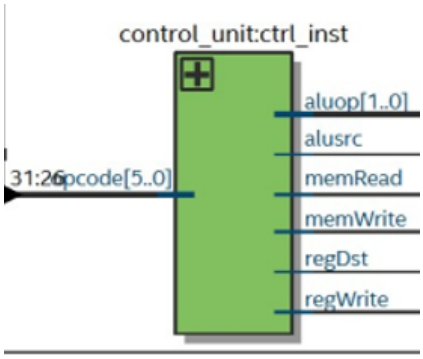


Instruction memory-
Error- no input from PC
Correction1- already corrected in the PC module.

Sign Extender-
For proper connections of modules we will need to create a separate module as sign extender, because internal connections in top module will not appear in RTL.



Muxes-
Only 2 out of 4 muxes have appeared in the RTL, this is due to not explicitly defining their inputs from other modules. Internal connections directly in the Top module will not appear as connections but as directly a new wire.



Adders and muxes

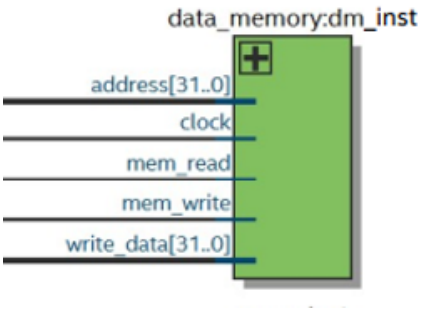
Many of the connecting wires do not show up in the RTL due to implicit definitions of adders and muxes. Although this is not a problem, we have decided to have a completely connected RTL design so we will also add the adder module and shift_left module separately.

Data_memory-

Error- No output wire

The output from data memory should be an input to a mux which along with alu result and MemToreg will decide what gets written in the regfile.

Correction- Made these connections by creating a separate mux module and instance.



Control unit-

Error- 2 of the output are missing.

Correction- Created a separate ‘AND’ gate which intakes these outputs so that they are visible in RTL.

These are the major problems that synthesis faced. With improvement in these we will conclude our Single cycle RISC processors iteration 2, and move on to more advanced concepts like pipelining in next iterations.