Report

32-Bit RISC_V Microprocessor Design

Introduction

This project requires the design of a 32-bit RISC-V architecture that perform 10 different instructions. All the conventional digial design procedures have been thoroughly followed throughout this project work, beginning from high-level simulation and HDL coding. The RTL and Testbenches were coded using verilog (SystemVerilog) and were simulated using both modelSim and Vivado. Additionally, the gate-level netlis and layout design were performed using Synopsys Design Compiler (DC Compiler for short) and IC compiler II (ICC2), respectively.

Design Structure and Setup

The RISC-V architectural design on this project consists of six different modules (excluding the top module) as shown in Figure-1. The module under the name “program_counter” fetches instructions from the intruction memory and feeds them to the decoder. The decoder interprets (decrypts) all the received instruction and generates corresponding control signals based on th instruction format given in table-1. These controls signals then drive the associated modules to perform a particular operation.

32-bit RISC-V Instruction Formats

Instruction Formats	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Register/register	funct7							rs2					rs1					funct3			rd					opcode						
Immediate	imm[11:0]												rs1					funct3			rd					opcode						
Upper Immediate	imm[31:12]																				rd					opcode						
Store	imm[11:5]							rs2					rs1					funct3			imm[4:0]					opcode						
Branch	[12]	imm[10:5]						rs2					rs1					funct3			imm[4:1]				[11]	opcode						
Jump	[20]	imm[10:1]										[11]	imm[19:12]								rd					opcode						

- **opcode (7 bit):** partially specifies which of the 6 types of *instruction formats*
- **funct7 + funct3 (10 bit):** combined with **opcode**, these two fields describe what operation to perform
- **rs1 (5 bit):** specifies register containing first operand
- **rs2 (5 bit):** specifies second register operand
- **rd (5 bit):**: Destination register specifies register which will receive result of computation

Table-1: RISC-V Instruction Format schematic

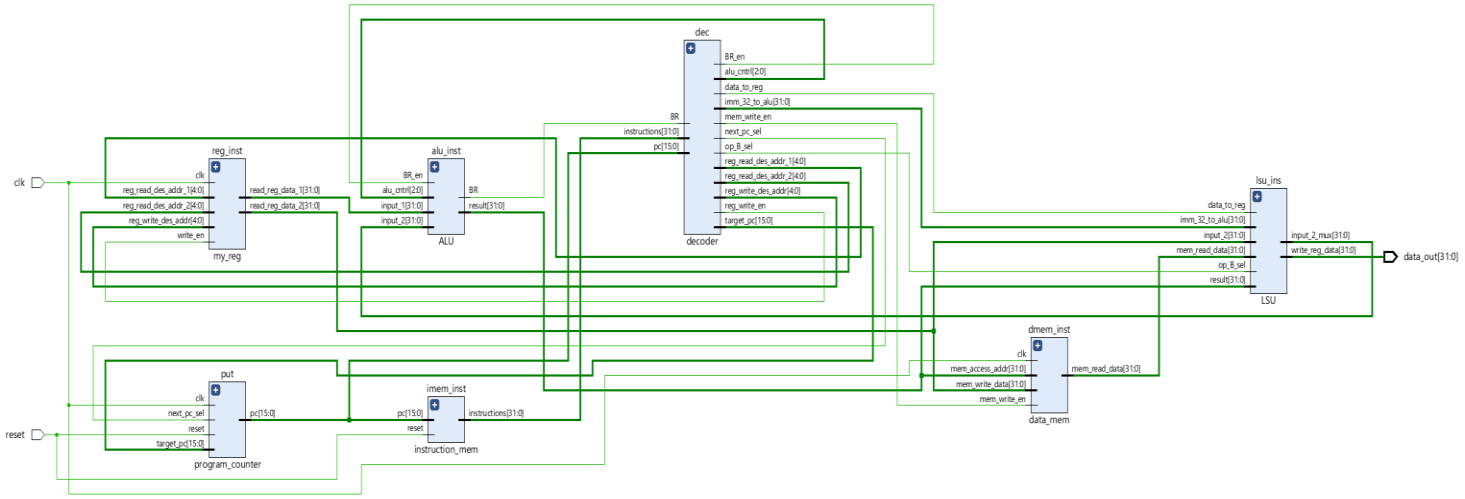

Figure-1: The schematic view of verilog RTL

The ALU Unit is responsible for arithmetic and logical operations. Additionally, the branching operation has been embedd inside the ALU unit. This was done with the mere reason that as we have only two branching operations we prefered to include them inside the ALU unit instead of creating a separate module. Doing so would also minimize the uneccesary power and resource utilization associated with the wiring. Moreover, the design consists of three different memory units each of them with a unique purpose and functionality. These are, the Register File, Data Memory and the Instruction Memory. Another important point to note here is that the bit size of the alu control signal that goes to the ALU unit from the decoder has been reduced

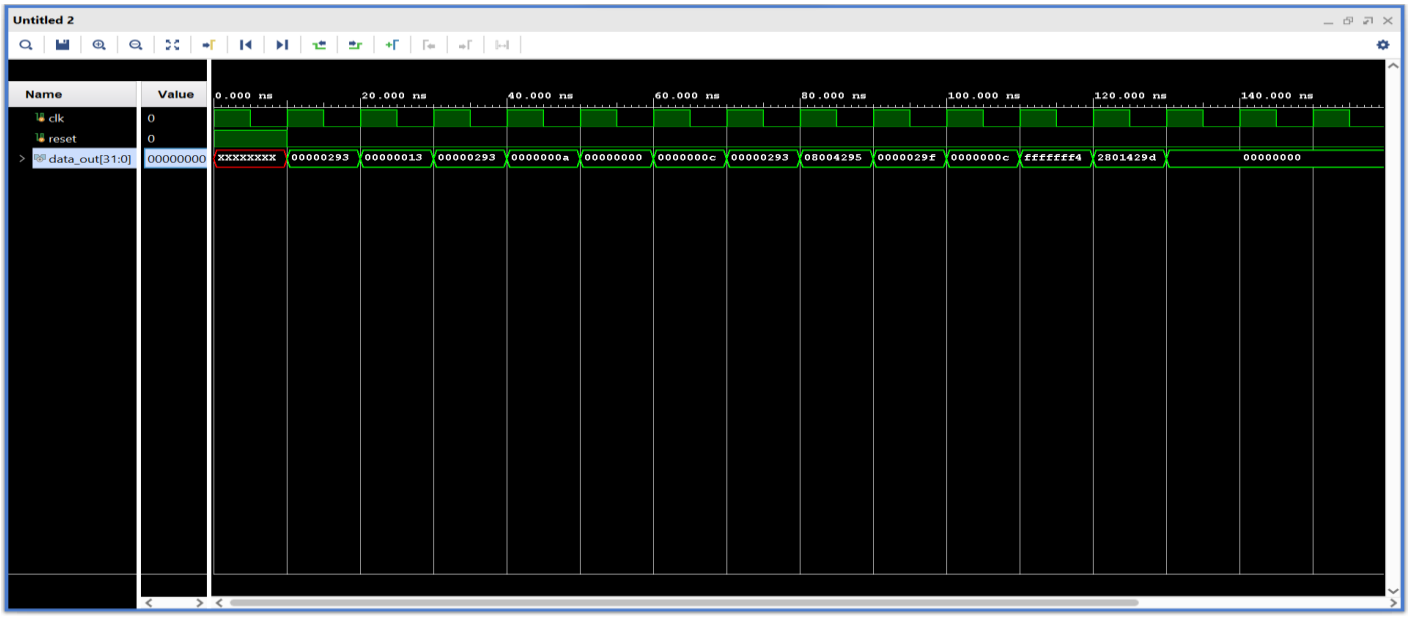

Figure-2: Simulation Results from Vivado

to three because it sufficiently satisfies the requirements for this project, and using a 6-bit or higher would only lead to the wastage of resources and power (particularly for this project).

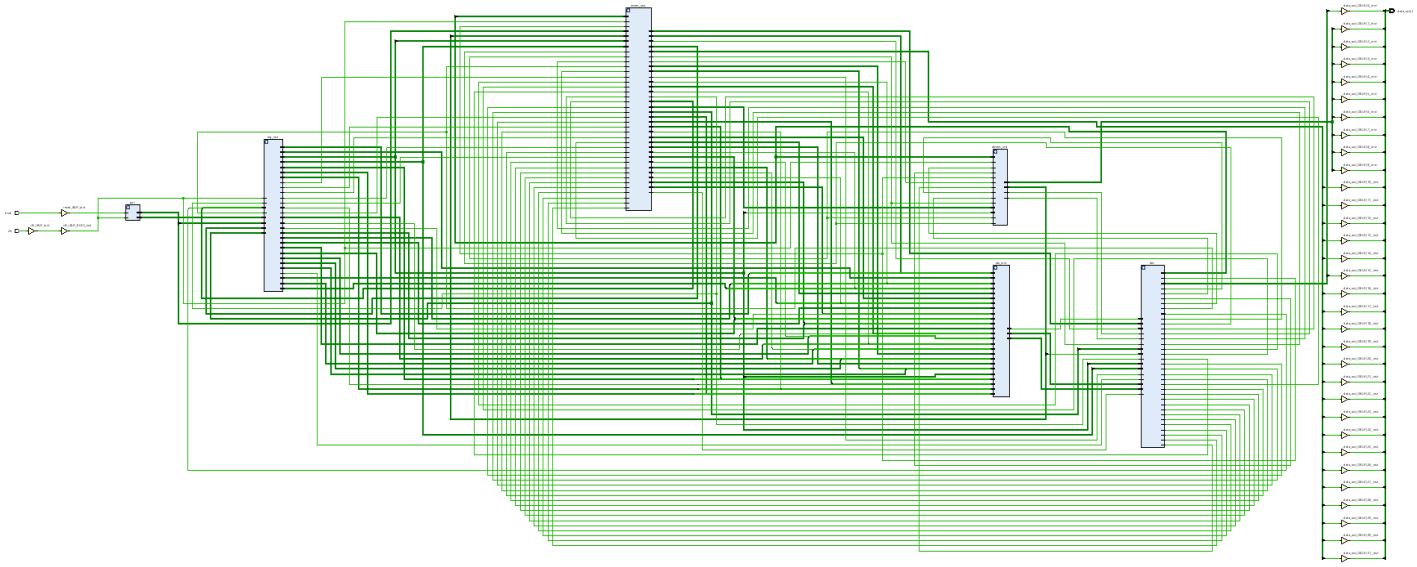

Figure-3: The Schematic of RIC-V when synthesized on vivado for Artix 7 FPGA

Synthesis Using Synopsys DC Compiler

The RTL design of this project has also been synthesized into a gate-level netlist using DC compiler. The schematic diagram shown in figure-4 is obtained after running synthesis on the design using Global Foundry's 22nm standard cell library. The design on this project capable of running at a maximum frequency of 2MHz (clock period of 500 ps) with out violations. Details about the timinig constraints and report is given in the below illustration.

```
clock clk (rise edge)                                  500.00     500.00
clock network delay (ideal)                              1.00     501.00
clock uncertainty                                       -1.80     499.20
output external delay                                  -50.00     449.20
data required time                                                449.20
--------------------------------------------------------------------------
data required time                                                449.20
data arrival time                                                -448.90
--------------------------------------------------------------------------
slack (MET)                                                         0.30
```

Listing-1: Timing report obtained from synthesi using DC compiler

```
Number of ports:                          992
Number of nets:                         96600
Number of cells:                        95697
Number of combinational cells:          61876
Number of sequential cells:             33811
Number of macros/black boxes:               0
Number of buf/inv:                       1041
Number of references:                       8

Combinational area:               28880.983093
Buf/Inv area:                       500.997103
Noncombinational area:            56253.384837
Macro/Black Box area:                 0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                  85134.367930
Total area:                 undefined
```

Listing-2: Area report obtained from synthesi using DC compiler

```
Global Operating Voltage = 0.9
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ps
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1uW


  Cell Internal Power  = 202.0429 mW   (100%)
  Net Switching Power  =   4.4386 uW     (0%)
                         ---------
Total Dynamic Power    = 202.0473 mW   (100%)

Cell Leakage Power     = 209.9193 uW


                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000              0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000              0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000              0.0000             0.0000  (   0.00%)
clock_network      0.0000           0.0000              0.0000             0.0000  (   0.00%)
register         202.1000        2.4991e-03           141.8064           202.2443  (  99.96%)
sequential         0.0000           0.0000          1.1851e-02         1.1851e-05  (   0.00%)
combinational    4.7029e-03       1.9394e-03            68.1009         7.4743e-02  (   0.04%)
--------------------------------------------------------------------------------------------------
Total            202.1047 mW     4.4386e-03 mW       209.9192 uW        202.3191 mW
```

Listing-3: power report obtained from synthesi using DC compiler

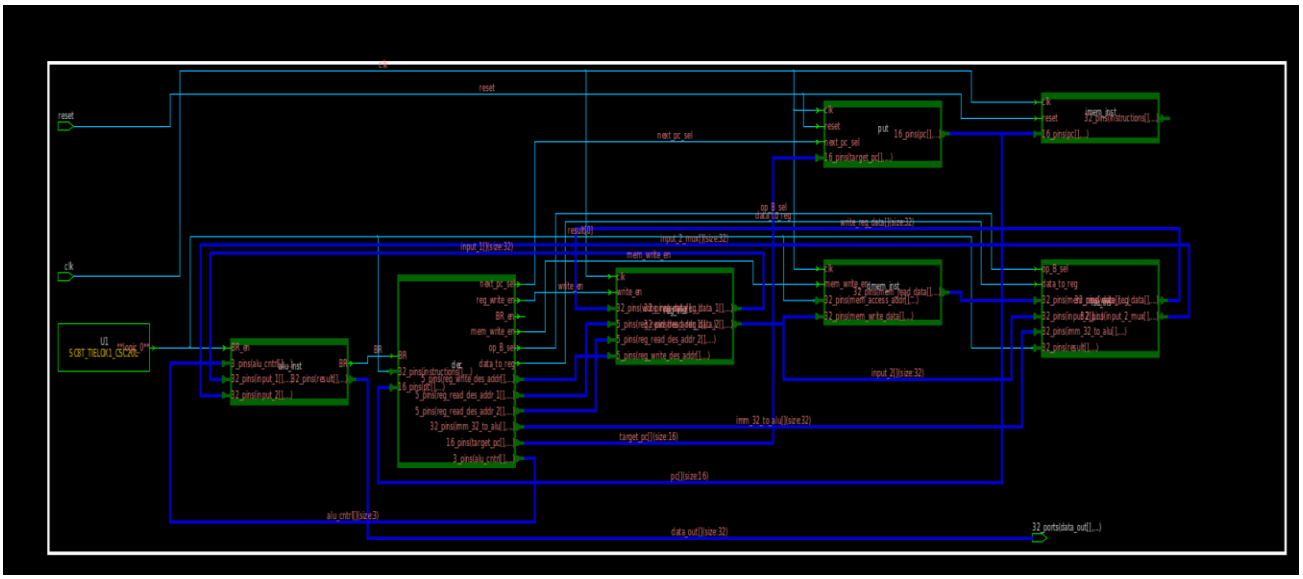

Figure-4: The Schematic diagram of RIC-V when synthesized on DC compiler

Place and Route

The figure below is obtained after place and route has also been carried out using IC Compiler II (ICC-II for short)

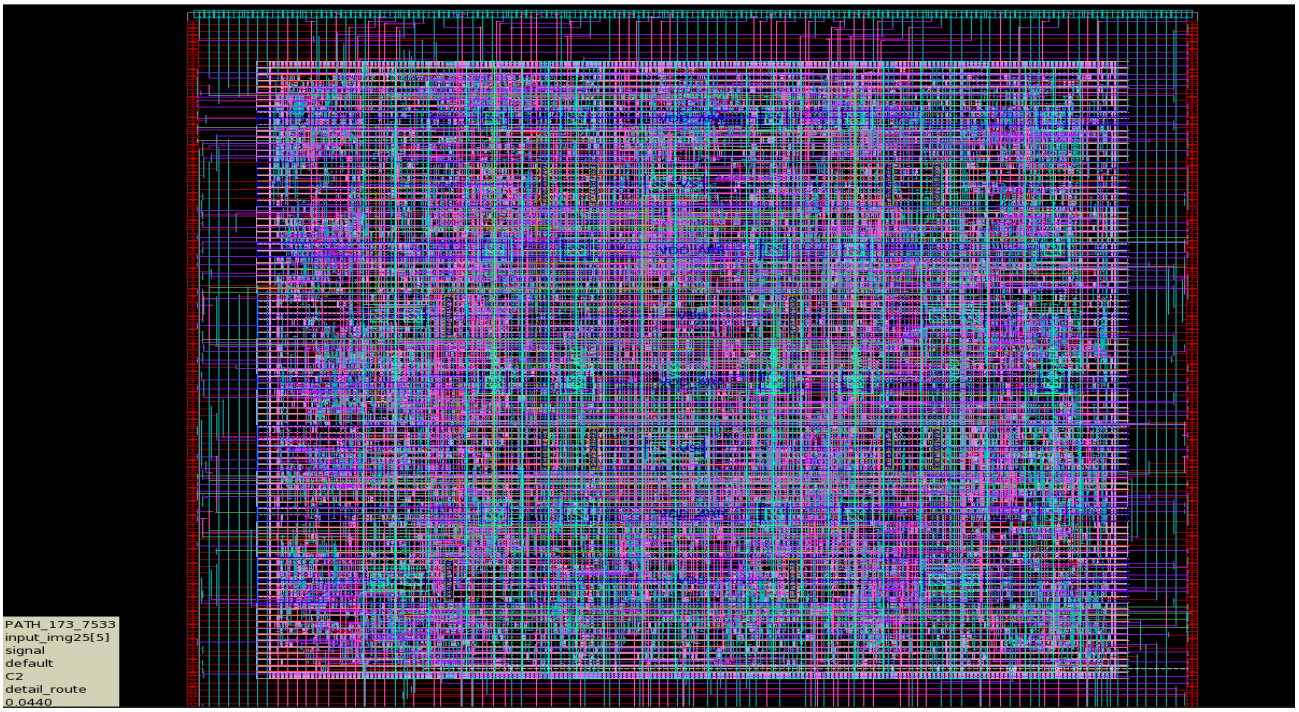

Figure-5: The Schematic diagram of RIC-V, obtained from ICC-II

Conclusion

The flow of the design of RISC-V architecture of this project was thoroughly monitored both for FPGA and ASIC design flow, and decent results were observed in both cases. Generally, the processor is capable of running at a maximum frequency of 2.0 GHz (the synthesis was undertaken with clock period of 500ns, assuming the default unit of time in DC compiler is in nanoseconds) without any timing violations. This chip occupies an area of 0.085mm^2 assuming 22nm GF technology, and a total power consumption of 202.3191mW. It is also worth mentioning that this architecture is designed for only a limited number of instructions and operations. However, this work can be expanded in the future to accommodate all the remaining instruction.