DLX Processor

Aaryan Darad (21110001)
Abhay Kumar Upparwal (21110004)
Vaibhavi Sharma (21110231)

1) Introduction:

This report presents the design and implementation of a DLX Processor in Verilog. The instruction set of the processor is based on the RISC architecture, which is a reduced instruction set computer. The DLX is capable of performing R-type Triadic, R-I type Triadic and R-type dyadic instructions. The processor has been designed using the Verilog hardware description language and has been verified through simulation. We have added upon the R-Type instructions from the first assignment and made it an even better processor.

2) Design:

The DLX Processor consists of three main components: the ALU, reg file, and the top module.

ALU consists of all the operation such as

1) **R Type Triadic:** ADD, SUB, AND, OR, XOR, SLL (shift left logical), SRL (shift right logical), SRA (shift right arithmetic), ROL (rotate left), ROR (rotate right), SLT (signed less than comparison), SGT (signed greater than), SLE (signed less than or equal to comparison), SGE, UGT, ULT, ULE, UGE.
2) **R-I Type Triadic:** ADDI, SUBI, ANDI, ORI, XORI, SLLI, SRLI, SRAI, ROLI, RORI, SLTI, SGTI, SLEI, SGEI, UGTI, ULTI, ULEI, UGEI, LHI.
3) **R-I Type Dyadic:** BEQZ, BNEZ, JR, JALR.

The register file is typically accessed by two inputs: the register address, which specifies the register to be read or written, and the register data, which specifies the value to be written to the register or the value to be read from the register. It also outputs if RS1 is zero or not for BEQZ and BNEZ.

Top module integrates all the components and it receives a 32 bit instruction and gives a 32 bit output value. It also has an instype input for the type of instruction to be performed.

The Processor has been verified using simulation. A testbench was created to verify the functionality of the processor. The testbench consists of a set of test cases that cover all the possible scenarios of the processor. The test cases include simple arithmetic and logical operations.

The simulation results show that the processor is able to execute all the instructions correctly. The processor was able to perform the operations on the operands as specified by the instructions. The register file and the ALU were able to store and perform the operations correctly. We have also added the PC register to show the R type Dyadic Instructions, we show the output of the ALU which takes PC as input.

3) Work Distribution:

1. Aaryan Darad:

- Coded Arithmetic and Comparator operations of R-Type in ALU.
- Added the isRS1zero and modified the Top Module.

2. Abhay Kumar Upparwal:

- Coded the Logical R- Type Instructions in ALU.
- Coded the Top Module.

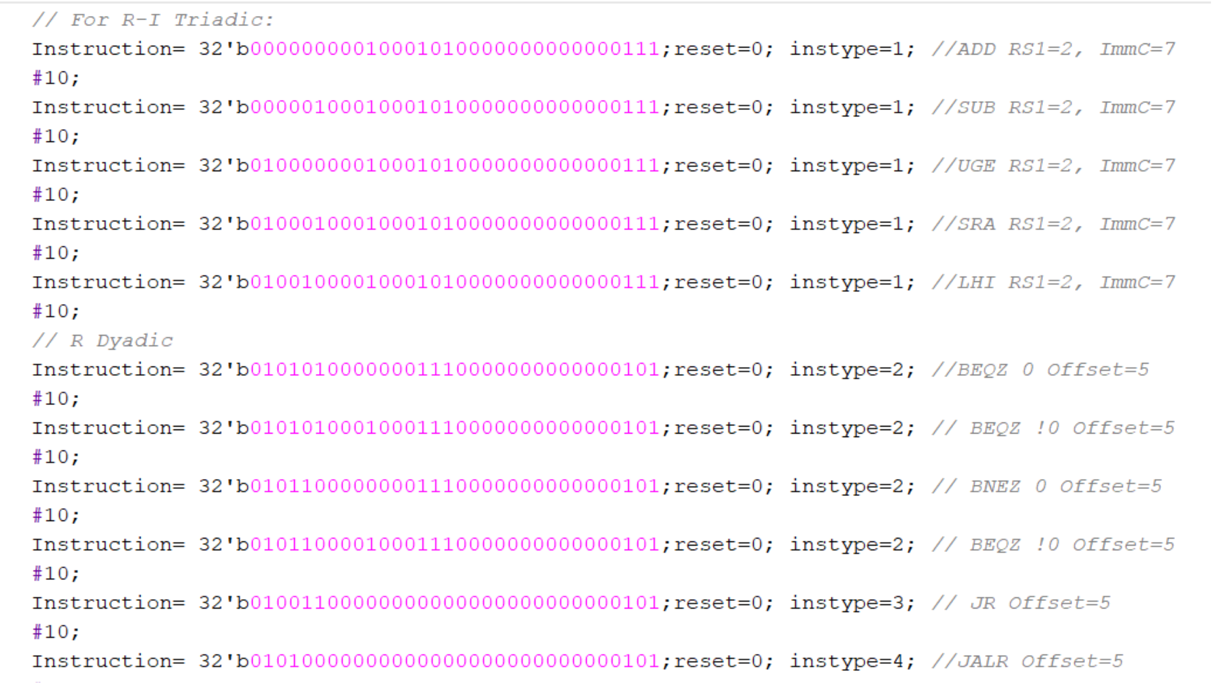
3. Vaibhavi Sharma:

- Coded the Register File and Top Module.
- Added the R-type Dyadic instructions in ALU and modified the Top Module.

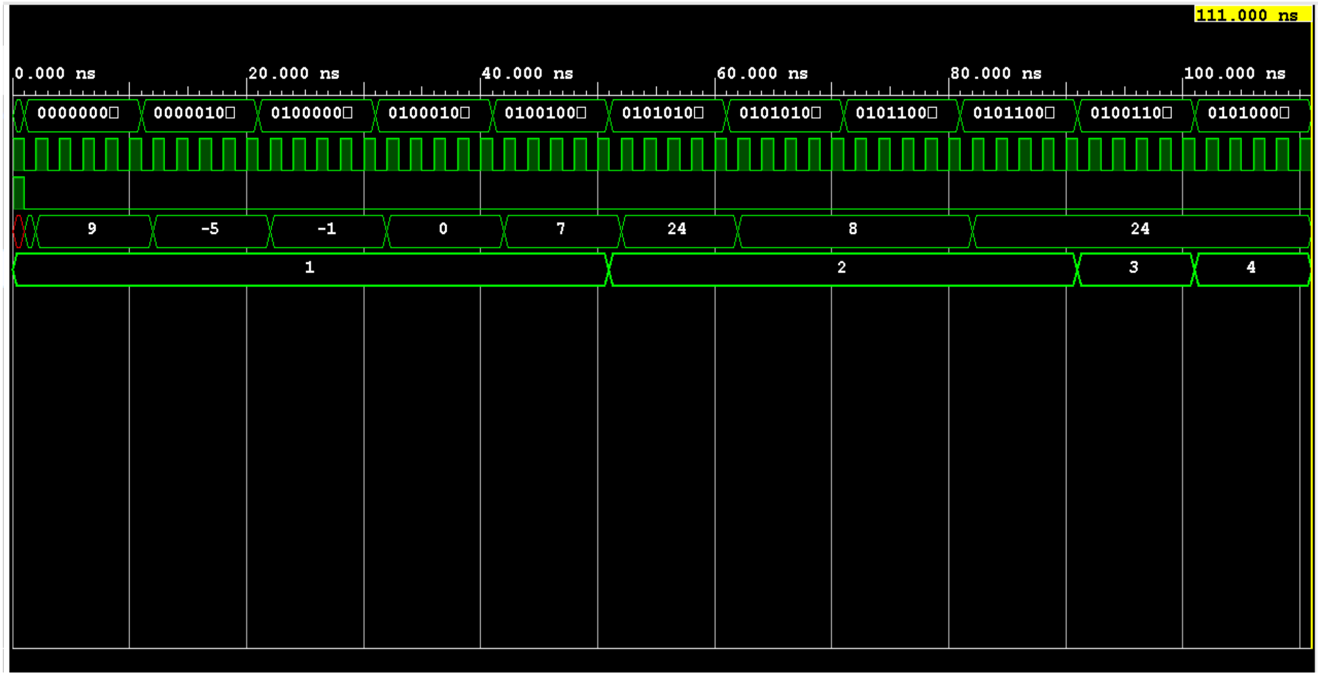
4) Verilog Codes

All the Verilog codes are uploaded on the github repository:
https://github.com/daradaaryan/ES215-COA-Aaryan-Abhay-Vaibhavi

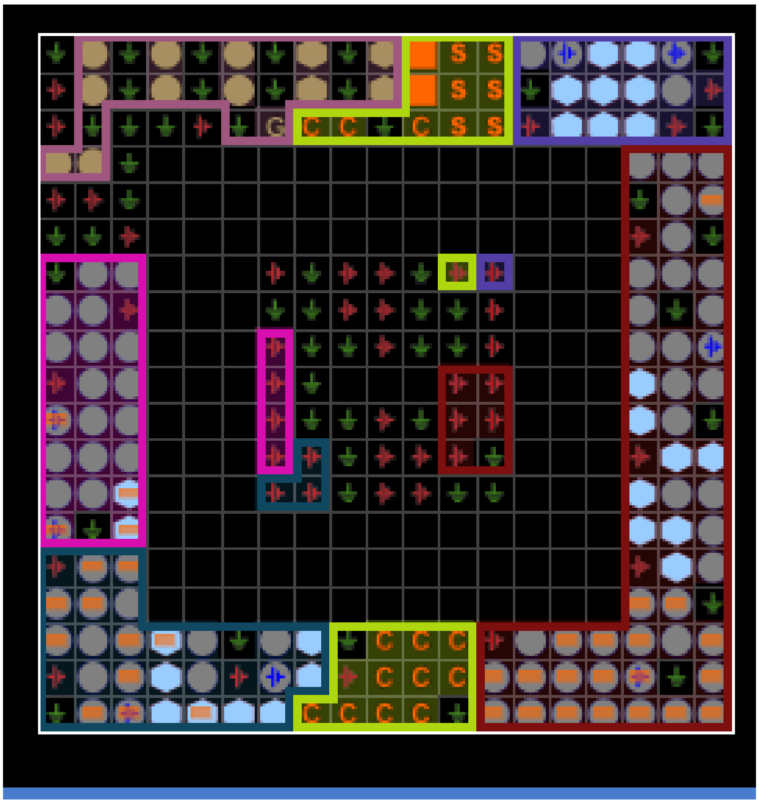
5) Simulation Waveforms:

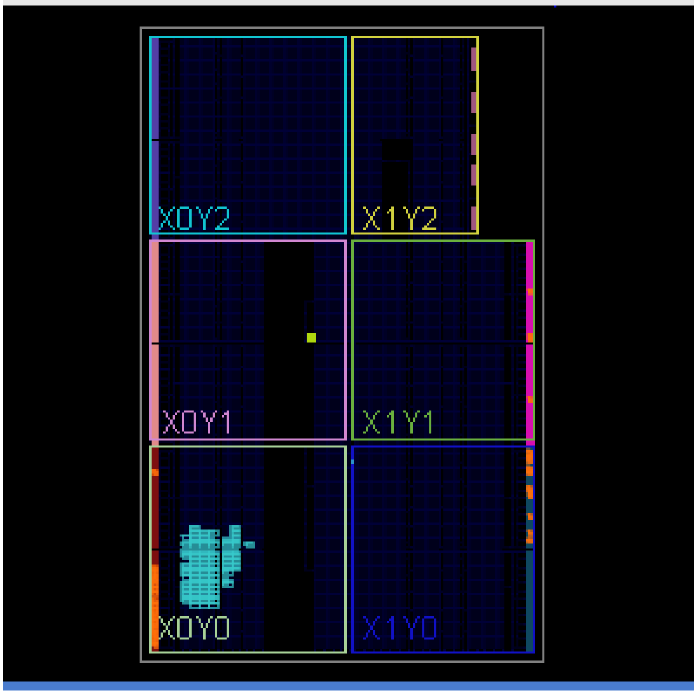


```
// For R-I Triadic:
Instruction= 32'b00000000010001010000000000000111;reset=0; instype=1; //ADD RS1=2, ImmC=7
#10;
Instruction= 32'b00000100010001010000000000000111;reset=0; instype=1; //SUB RS1=2, ImmC=7
#10;
Instruction= 32'b01000000010001010000000000000111;reset=0; instype=1; //UGE RS1=2, ImmC=7
#10;
Instruction= 32'b01000100010001010000000000000111;reset=0; instype=1; //SRA RS1=2, ImmC=7
#10;
Instruction= 32'b01001000010001010000000000000111;reset=0; instype=1; //LHI RS1=2, ImmC=7
#10;
// R Dyadic
Instruction= 32'b01010100000000111000000000000101;reset=0; instype=2; //BEQZ 0 Offset=5
#10;
Instruction= 32'b01010100010001110000000000000101;reset=0; instype=2; // BEQZ !0 Offset=5
#10;
Instruction= 32'b01011000000000111000000000000101;reset=0; instype=2; // BNEZ 0 Offset=5
#10;
Instruction= 32'b01011000010001110000000000000101;reset=0; instype=2; // BEQZ !0 Offset=5
#10;
Instruction= 32'b01001100000000000000000000000101;reset=0; instype=3; // JR Offset=5
#10;
Instruction= 32'b01010000000000000000000000000101;reset=0; instype=4; //JALR Offset=5
```

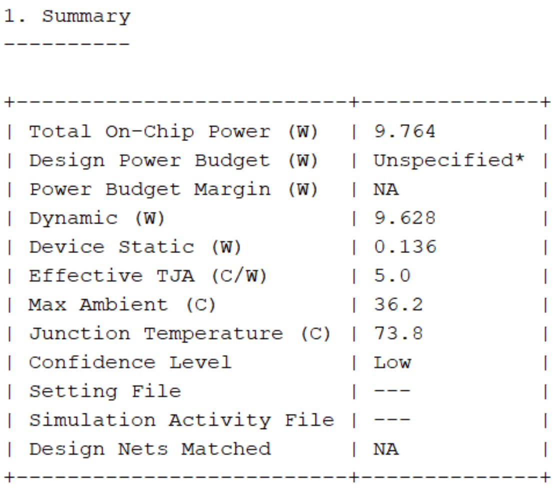


6) Synthesis and Implementation



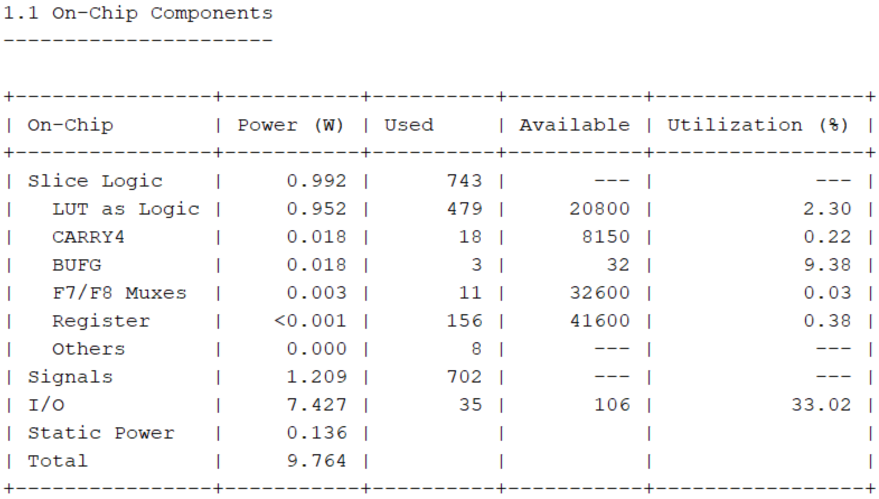


7) Performance Report



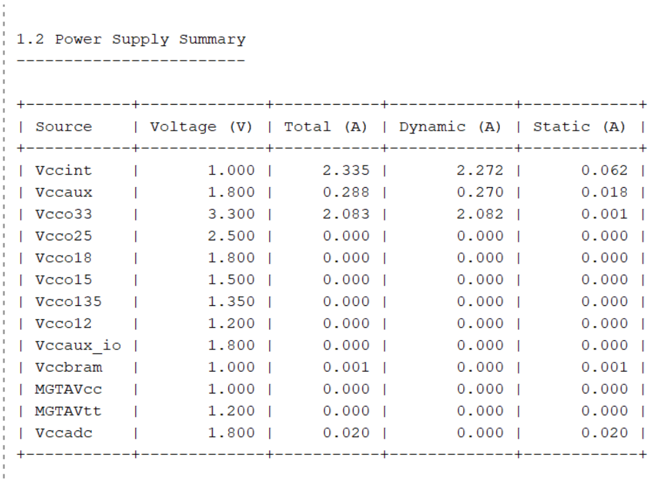
```
1. Summary
----------

+--------------------------+--------------+
| Total On-Chip Power (W)  | 9.764        |
| Design Power Budget (W)  | Unspecified* |
| Power Budget Margin (W)  | NA           |
| Dynamic (W)              | 9.628        |
| Device Static (W)        | 0.136        |
| Effective TJA (C/W)      | 5.0          |
| Max Ambient (C)          | 36.2         |
| Junction Temperature (C) | 73.8         |
| Confidence Level         | Low          |
| Setting File             | ---          |
| Simulation Activity File | ---          |
| Design Nets Matched      | NA           |
+--------------------------+--------------+
```



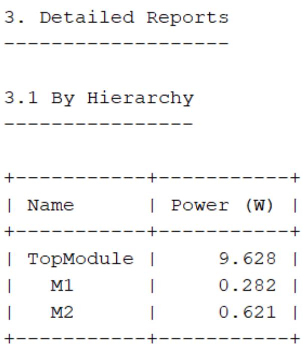
```
1.1 On-Chip Components
----------------------

+----------------+-----------+----------+-----------+-----------------+
| On-Chip        | Power (W) | Used     | Available | Utilization (%) |
+----------------+-----------+----------+-----------+-----------------+
| Slice Logic    |     0.992 |      743 |       --- |             --- |
|   LUT as Logic |     0.952 |      479 |     20800 |            2.30 |
|   CARRY4       |     0.018 |       18 |      8150 |            0.22 |
|   BUFG         |     0.018 |        3 |        32 |            9.38 |
|   F7/F8 Muxes  |     0.003 |       11 |     32600 |            0.03 |
|   Register     |    <0.001 |      156 |     41600 |            0.38 |
|   Others       |     0.000 |        8 |       --- |             --- |
| Signals        |     1.209 |      702 |       --- |             --- |
| I/O            |     7.427 |       35 |       106 |           33.02 |
| Static Power   |     0.136 |          |           |                 |
| Total          |     9.764 |          |           |                 |
+----------------+-----------+----------+-----------+-----------------+
```



```
1.2 Power Supply Summary
------------------------

+-----------+-------------+-----------+-------------+------------+
| Source    | Voltage (V) | Total (A) | Dynamic (A) | Static (A) |
+-----------+-------------+-----------+-------------+------------+
| Vccint    |       1.000 |     2.335 |       2.272 |      0.062 |
| Vccaux    |       1.800 |     0.288 |       0.270 |      0.018 |
| Vcco33    |       3.300 |     2.083 |       2.082 |      0.001 |
| Vcco25    |       2.500 |     0.000 |       0.000 |      0.000 |
| Vcco18    |       1.800 |     0.000 |       0.000 |      0.000 |
| Vcco15    |       1.500 |     0.000 |       0.000 |      0.000 |
| Vcco135   |       1.350 |     0.000 |       0.000 |      0.000 |
| Vcco12    |       1.200 |     0.000 |       0.000 |      0.000 |
| Vccaux_io |       1.800 |     0.000 |       0.000 |      0.000 |
| Vccbram   |       1.000 |     0.001 |       0.000 |      0.001 |
| MGTAVcc   |       1.000 |     0.000 |       0.000 |      0.000 |
| MGTAVtt   |       1.200 |     0.000 |       0.000 |      0.000 |
| Vccadc    |       1.800 |     0.020 |       0.000 |      0.020 |
+-----------+-------------+-----------+-------------+------------+
```



```
3. Detailed Reports
-------------------

3.1 By Hierarchy
----------------

+-----------+-----------+
| Name      | Power (W) |
+-----------+-----------+
| TopModule |     9.628 |
|   M1      |     0.282 |
|   M2      |     0.621 |
+-----------+-----------+
```

8) Summary

The Assignment 2 of our project was an even greater challenge than the first assignment, but we enjoyed every moment of it. We had to make modifications to our previous implementation and it was amazing to see the effects of our changes in real-time on the FPGA. We gained a deeper understanding of Computer Architecture and Organization and we're excited to explore this field even further. Overall, this experience has been extremely rewarding and we can't wait to see where our learning takes us next.
Thanks to Prof. Rajat Moona and Prof. Sameer Kulkarni for giving us this opportunity.