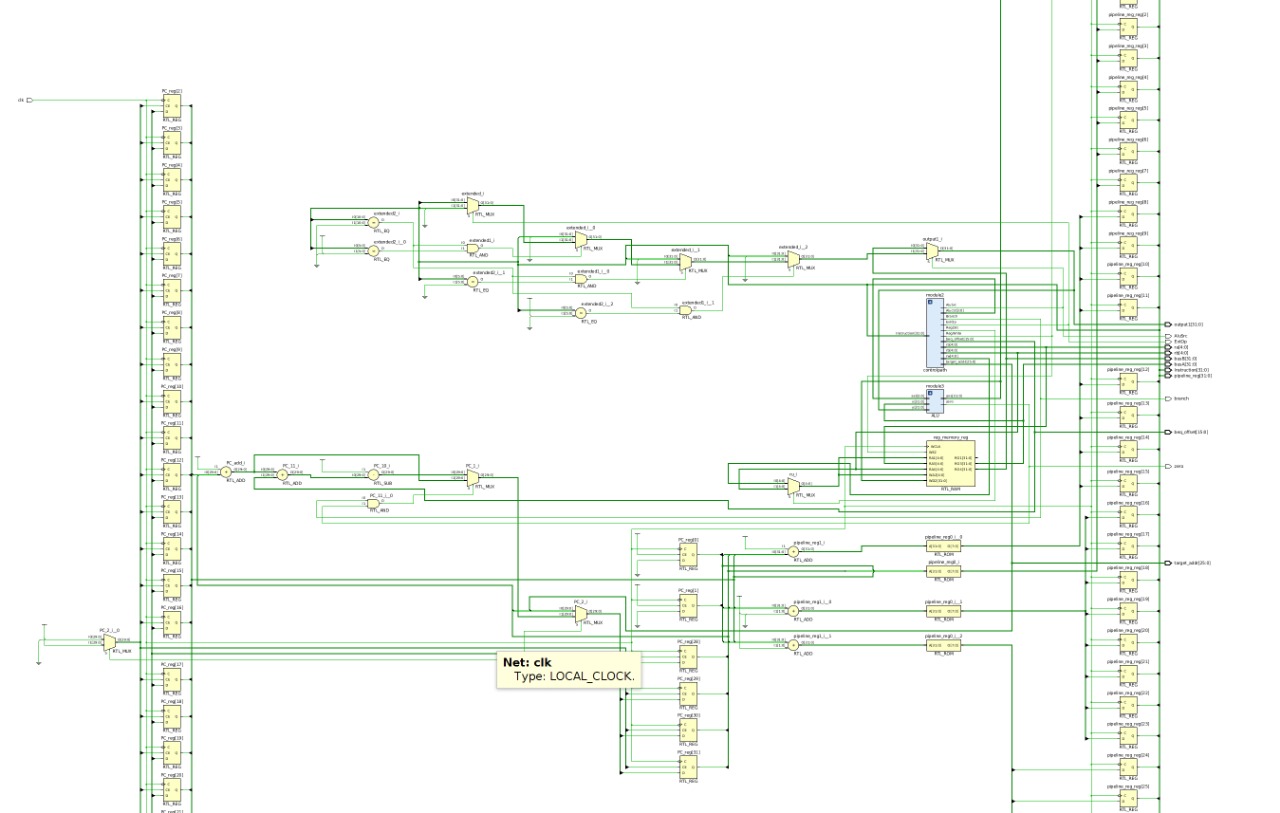
FINAL MIPS PROCESSOR REPORT –

1.Architectural Design:

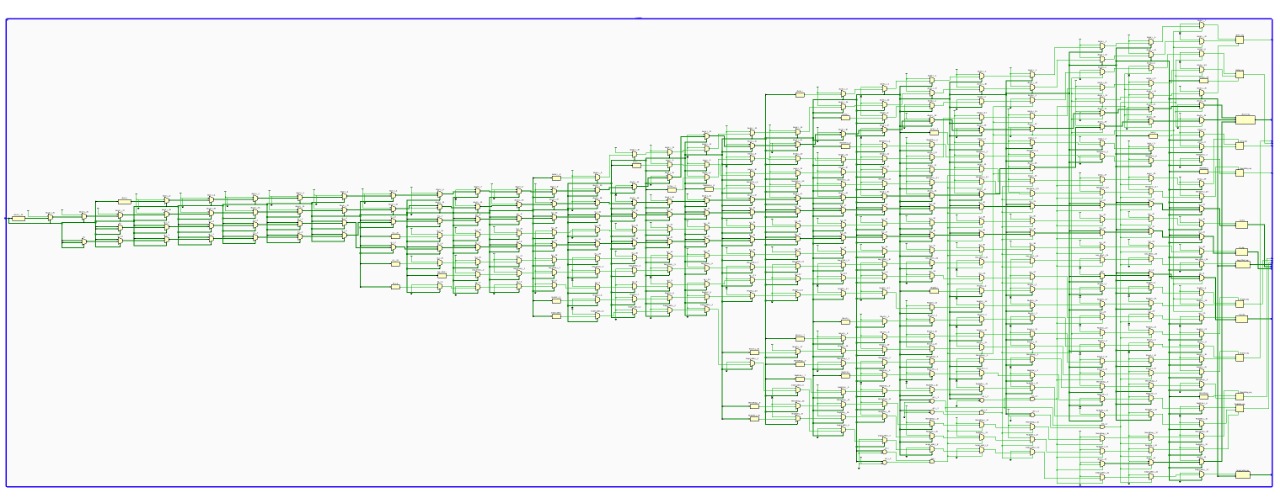
• 2–stage pipelined processor with separate blocks for Fetch unit and the Decode and Execute unit.

• ALU implemented with 16 operations including arithmetic, logical, shifting and relational.

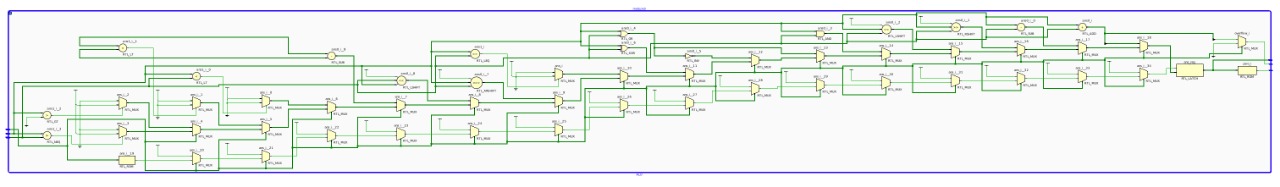
• Decode unit implemented with 22 instructions which gives the control signals, register addresses, branch offset, jump target address, immediate value and ALU operation control.



Control path-



ALU-



Timing of implemented instructions-

2. Main-Memory Design –

Main memory is realized using 2D array in Verilog. The 2D array used is has dimensions 32X32 (Since not many instructions are used in test bench). The memory is word based (= 4 bytes). Main memory and Instruction memory are separately realized on the basis of Harvard architecture.

Instruction memory is also realized using 2D array. It’s size is 128X8. This instruction memory is byte-based and is initialized in fetch unit.

Total Main memory size – 1 KB.

Total instruction memory size – 128 Bytes.

3. Instructions supported –

* 14 R-type Instructions, 11 I-type Instructions and 2 jump instructions have been implemented -
* ADD
* ADDI
* AND
* ANDI
* NOR
* OR
* ORI
* SLL
* SLLV
* SRA
* SRAV
* SRL
* SRLV
* SUB
* XOR
* XORI
* SLT
* SLTU
* SLTI
* BEQ
* BGTZ
* BLEZ
* BNE
* J
* JAL
* LB
* SB

4. Instruction not supported -

* SLTIU

5. Processor Clock Frequency –

The maximum clock frequency supported is 1.2GHz and hence the minimum clock period is 0.833ns.

6. Any other interesting things about your processor design. Both pros and cons

* Pros-
* RTL makes it easy to visualize the flow of the MIPS processor.
* The addition of pipeline register makes it time efficient.
* 2 – stage pipelined architecture.
* Simple realization of Data-memory, and registers files.
* Harvard architecture-based memory management implemented which is ideal for embedded computer systems.
* Cons-
* Despite of a perfect RTL , all modules in a single code makes it difficult to debug the Verilog code.
* Overflow flag doesn’t raises any exception in writing to registers.
* The Instructions to be implemented should be converted to binary format and entered manually to the register files.