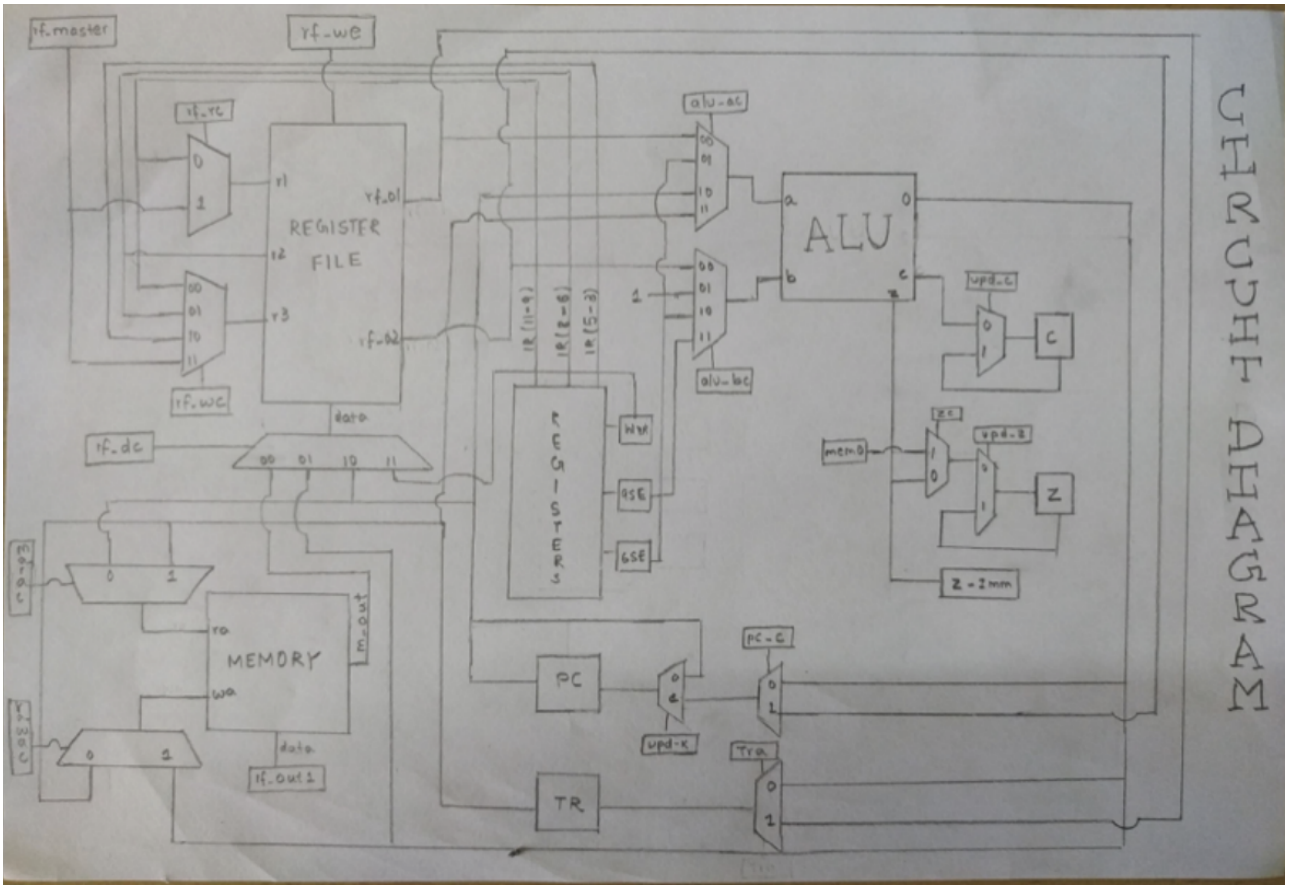
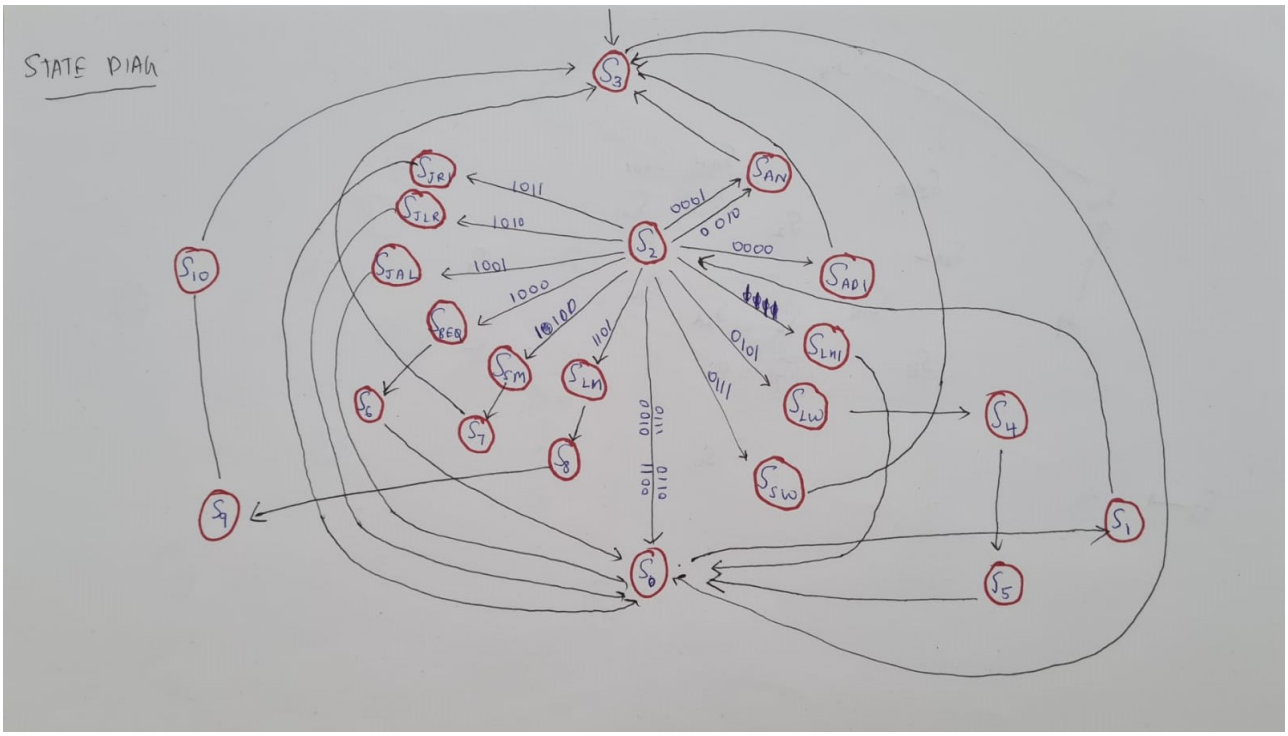
IITB-RISC

Aditya Jain, Ashwin Ramachandran
Pulkit Agarwal, Vaibhav Raj

1 Circuit diagram



2 State Diagram



3 States

State 0:- This is simply a synchronization state which is the starting state of all operations.

State 1 :- This state updates a control signal which causes the instruction fetch step (i.e., the instruction gets stored in the instruction register).

State 2 :- Another common state for all operations. This state updates different control signals depending upon the type of operation that we are trying to perform. These control signals in turn decide the inputs of various components such as the register file or the ALU.

For example, for ADD instruction we might want the two ALU inputs to be from the outputs of the register file. For some other instruction (such as the jump instructions) one of the input might have to be the Program Counter. In this case, we want to control the inputs of the component according to the instruction (which is defined by the operation code).

State 3 :- This state is responsible for incrementing the program counter by 1. The control signals are set such that ALU-a is equal to PC and ALU-b is equal to 1. The control signal to update program counter is also set to 1. This state is the last state of any operation as it leads to the zero state.

State 4:- This state defines "m_rac" which is a control signal to determine the address in the memory which has to be read. If this state is not a part of some instruction, then the only memory read operation to be done for that instruction would be the reading of the instruction. In instructions that involve memory read, the state sets "m_rac to '1', which leads to the read address for memory being changed to some other value depending upon the operation.

State 5:- This state is the successor of state 4. This state increments the Program Counter and also enables writing to the register file. State 4 and State 5 are a part of the load operation.

State 6:- This state is ran by the BEQ operation. It sets the control signals for updating the program counter and also for the inputs of the ALU. Note that the ALU inputs depend upon the zero bit. According to the equality condition, the ALU-b signal is set either to 1 or the value of the immediate.

State 7:- For the SM operation. Sets the control signals to increment the memory address for writing by 1. Also, this state causes the register file read address to the output of the priority encoder. The priority encoder takes the 8 bit signal (which is a part of the instruction), and outputs the register indices that need to be updated in a ascending order.

Note that this state is repeated multiple times. When running the SM operation, this state occurs as many times as we have to store into the memory, each time with the writing address being 1 more then the previous writing address.

State 8:- This state is responsible for incrementing the memory address from which has to be loaded into the register for the LM operation. The register which has to be written into is set according to the output of the priority encoder. The LM operation will need one state to run multiple times due to multiple write operations into the register file. This will be done in state 9, which is a successor of state 8.

State 9:- This state runs repetitively in the LM operation. Writing is done into the registers on the basis of addresses derived from the 8 bit selection signal.

State 10:- This state is a non repeating state for LM operation. This state does the last write operation into the register file and leads to state 3 for storing the update of program counter.

4 16 Bit Kogge Stone Adder

To design the adder, we have used a 16 bit Kogge Stone Network, with the process done as shown below.