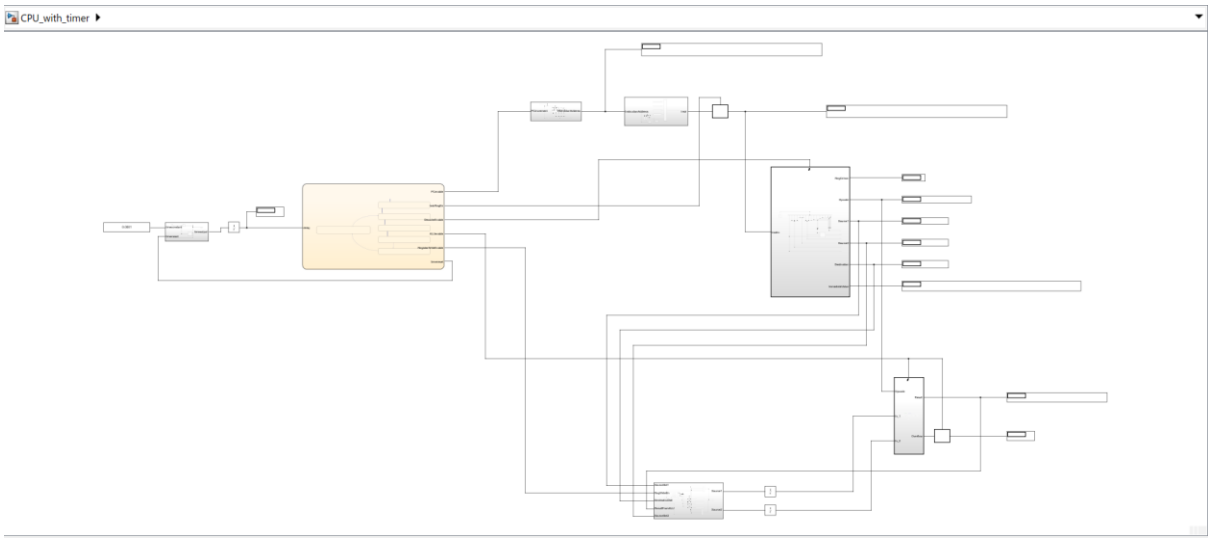
Design of a Single Cycle Processor

Bharat Kashyap Karri
2020AAPS0319H
Jan 2022

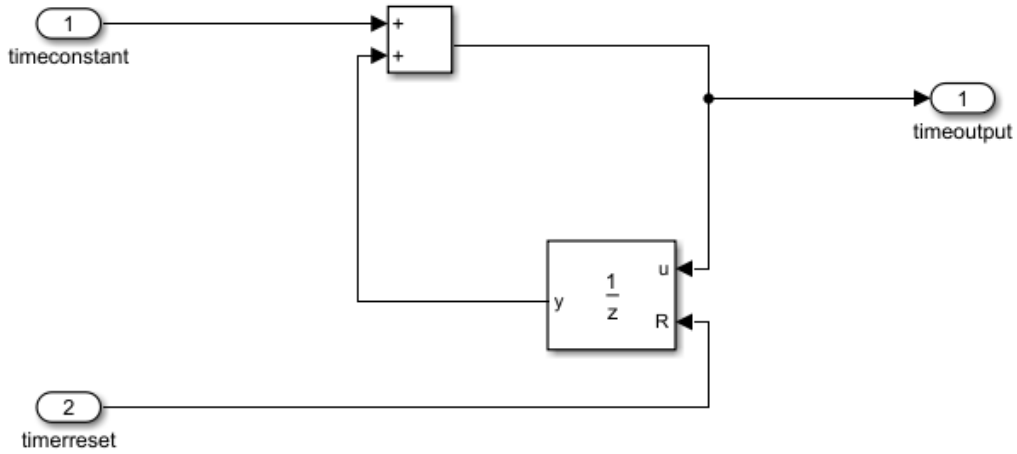
Overview

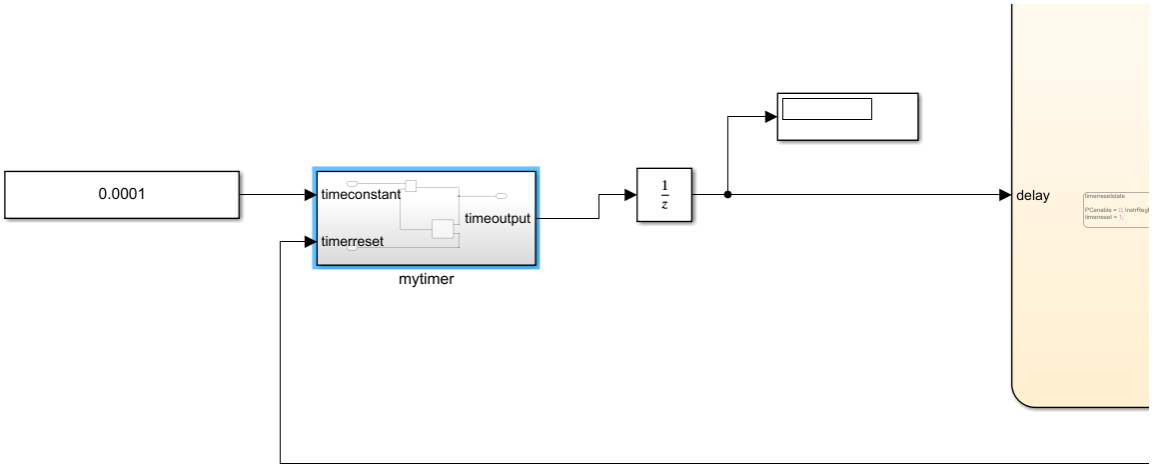
The project involved designing a single cycle processor that performs 4-bit arithmetic and logical instructions using 4 general purpose registers. Built a 4-bit ALU, register file, instruction decoder, program counter and a state machine to control the flow. Simulated in MATLAB Simulink.



Components

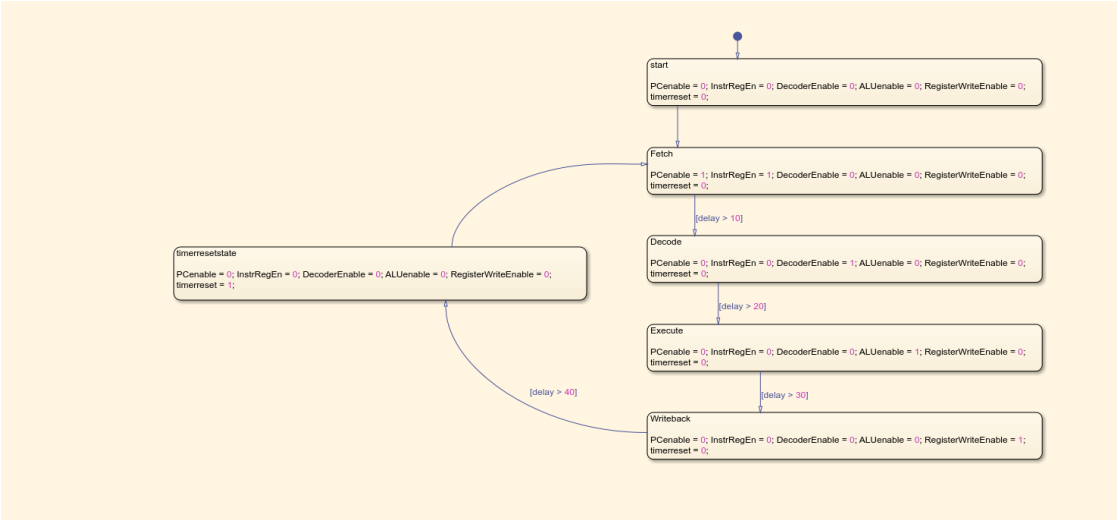
1. Timer
   This subsystem allows for the repeatability of the set of instructions.





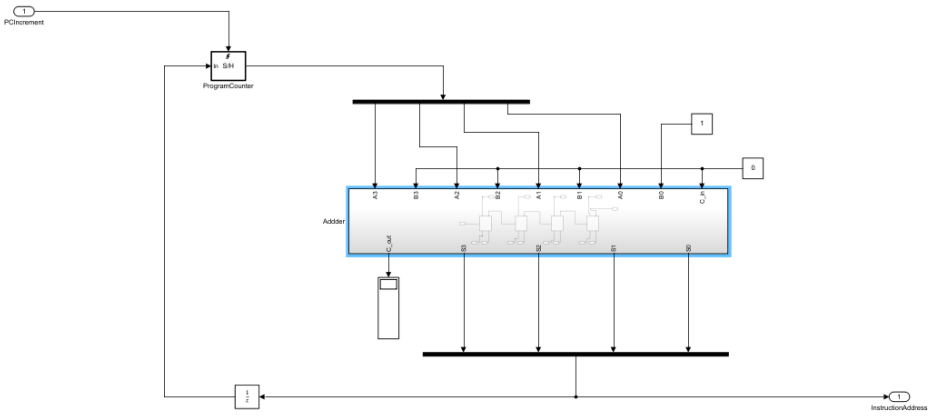
2. State Machine

   Controls the flow of the program with the trigger block in each subsystem. Each subsystem has an input from the state machine to set or reset the blocks in the subsystem.

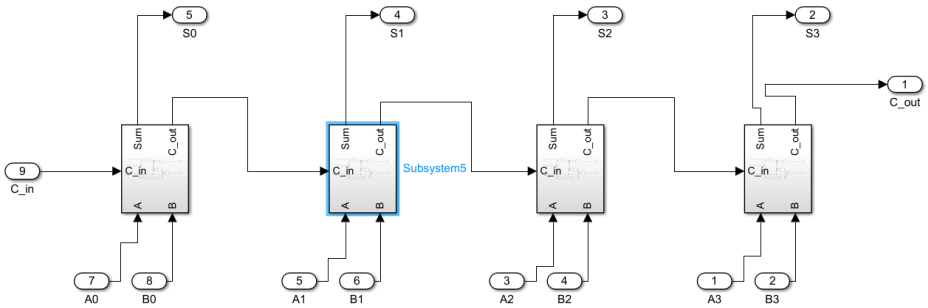


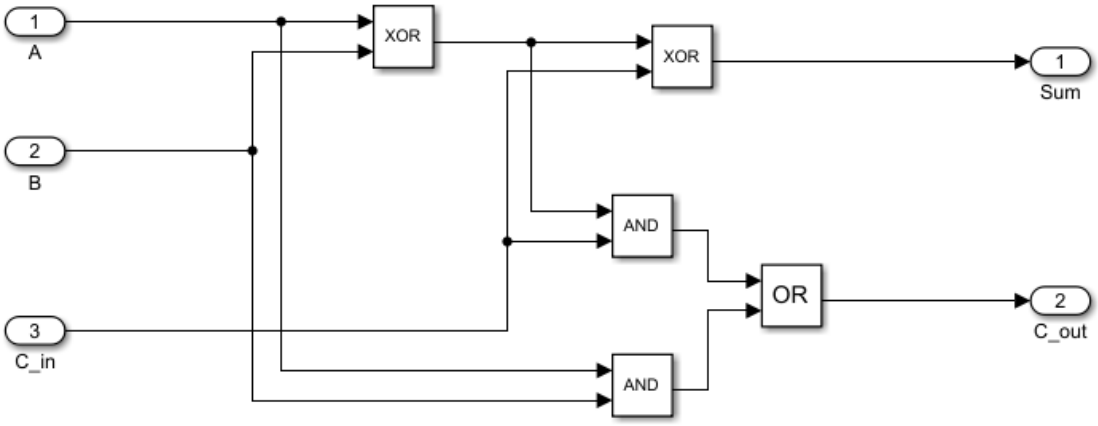
3. Program Counter Increment

   Increments the program counter as a substitute for the Instruction Address. In this implementation there are 4 instructions, and the locations of these instructions are given as numbers from the program counter. Contains a 4-bit Adder made from logic gates.



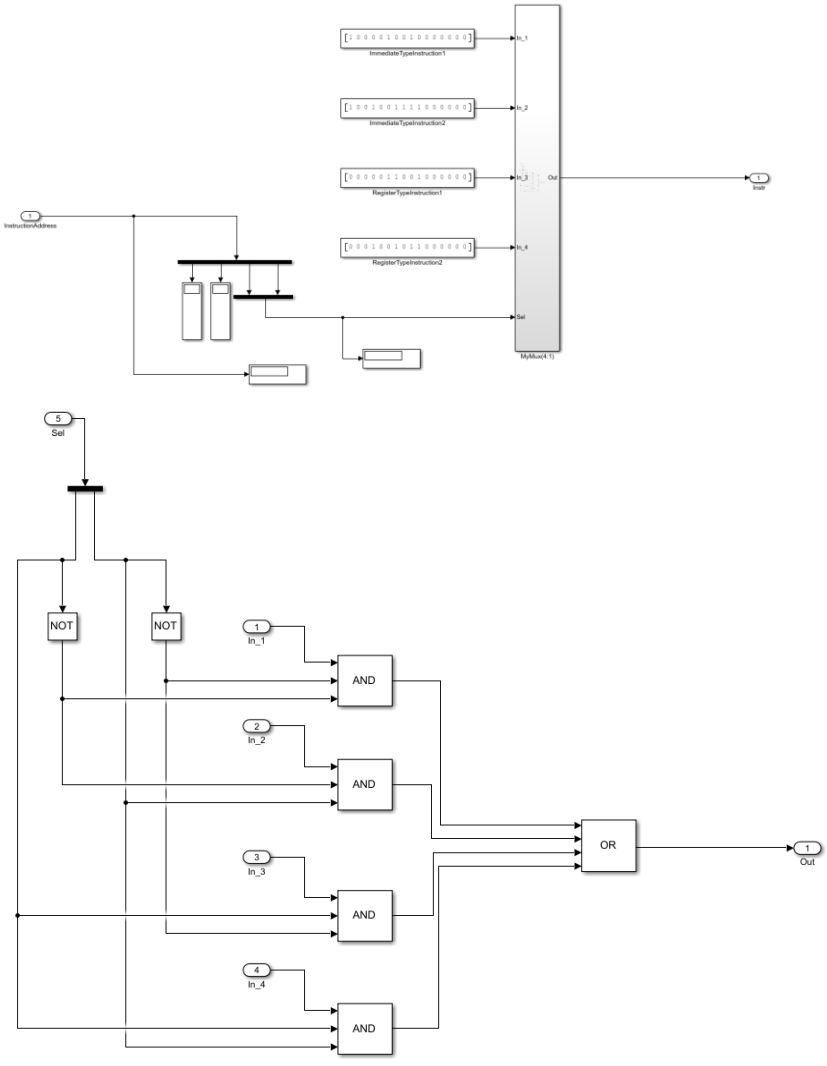
Adder:





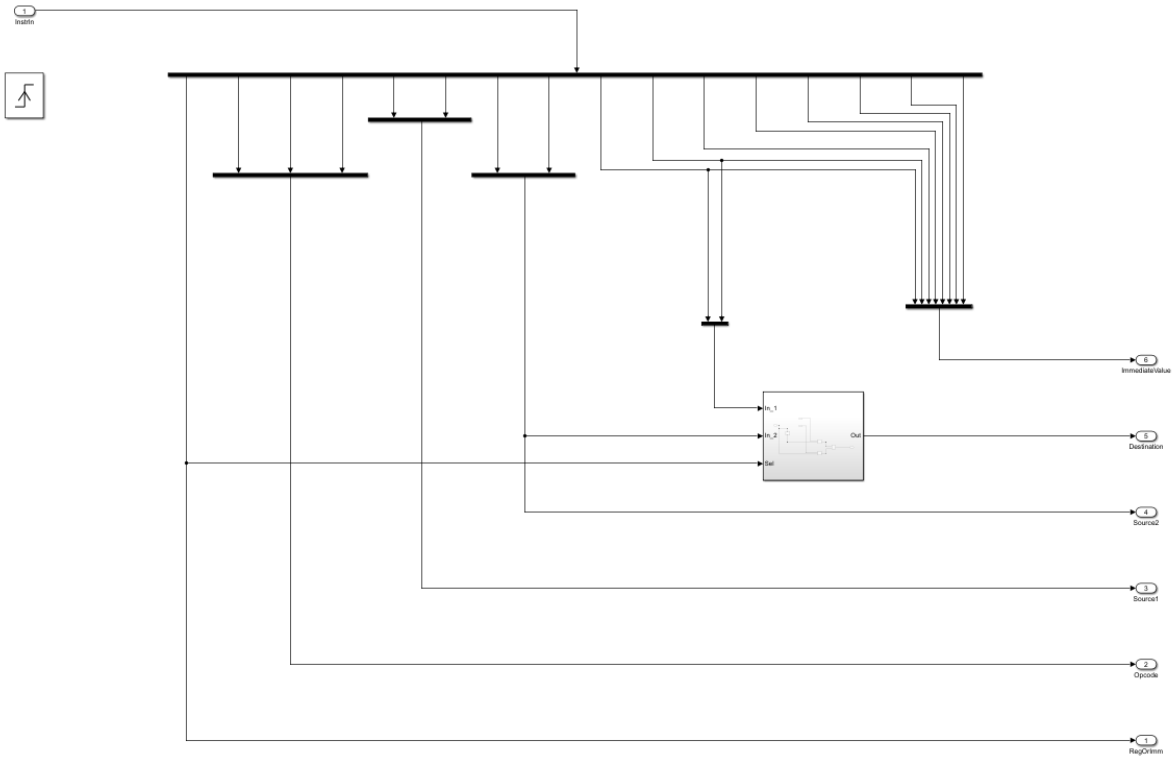
4. Instruction Memory

   Selects the instruction to output and send to the Instruction decode subsystem. This is done by using an inbuilt DEMUX and a MUX with select made from logic gates. In the instructions set there are 4 instructions: 2 immediate type and 2 register type.



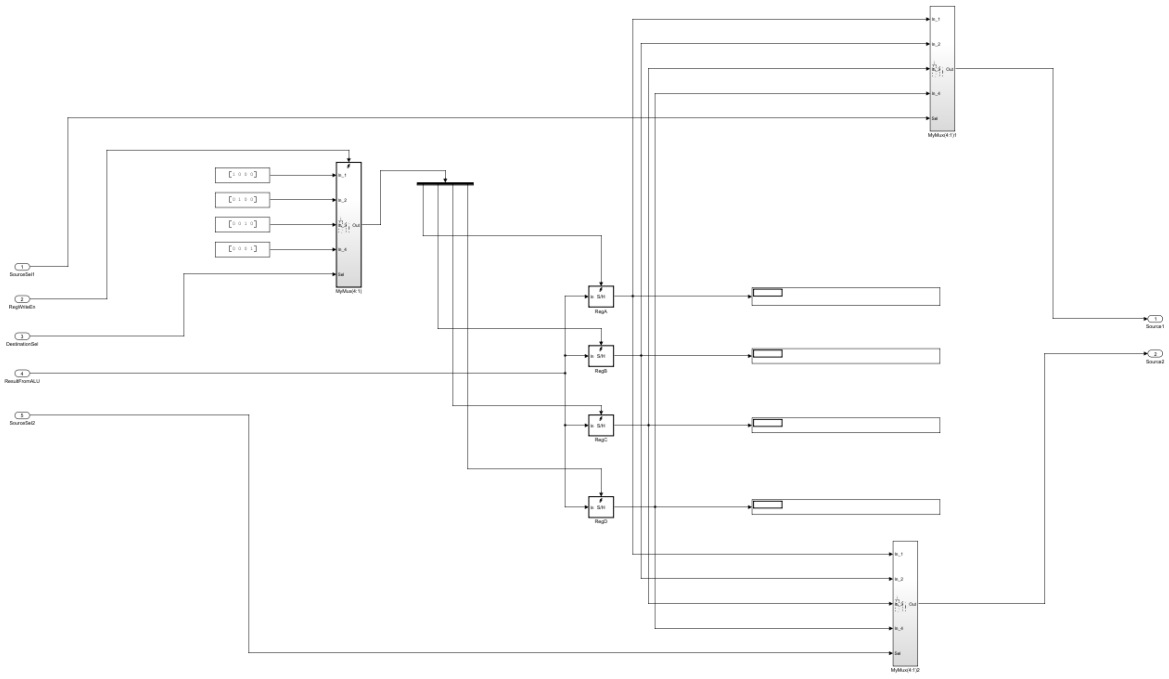
5. Instruction Decode

   The required source registers, destination register, opcode and immediate value (for immediate type instructions) are split and outputted using an inbuild DEMUX. The difference between immediate and register type instructions is also considered.



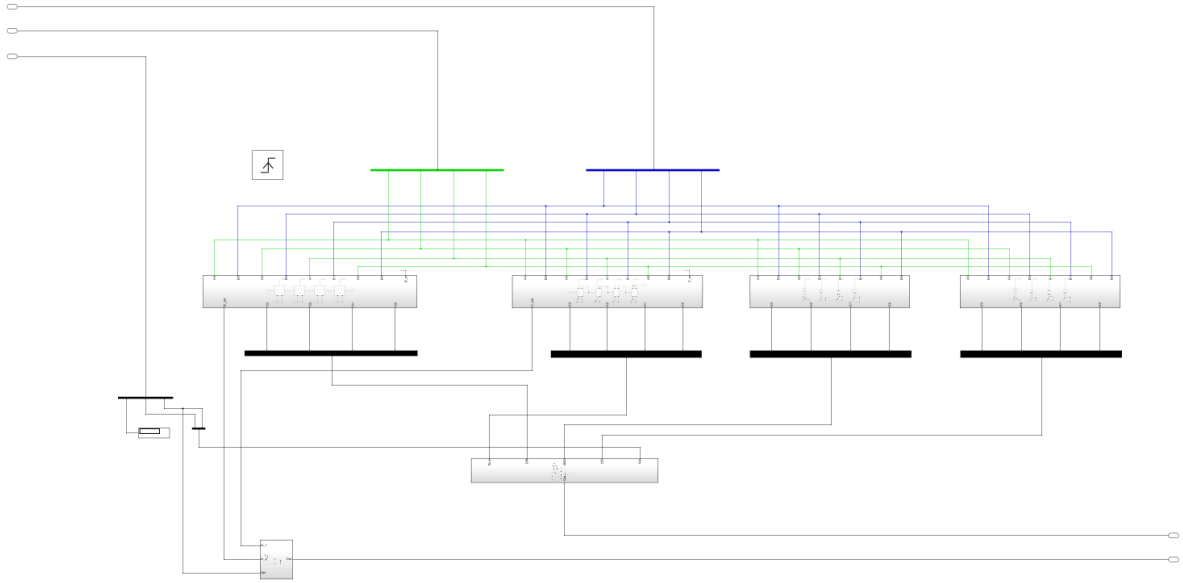
6. Register File

   The destination register is selected using one hot encoding. Source registers from instruction decode subsystem are used to select the correct registers using the MUX with select. There are a total of 4 registers each containing 4 bits.

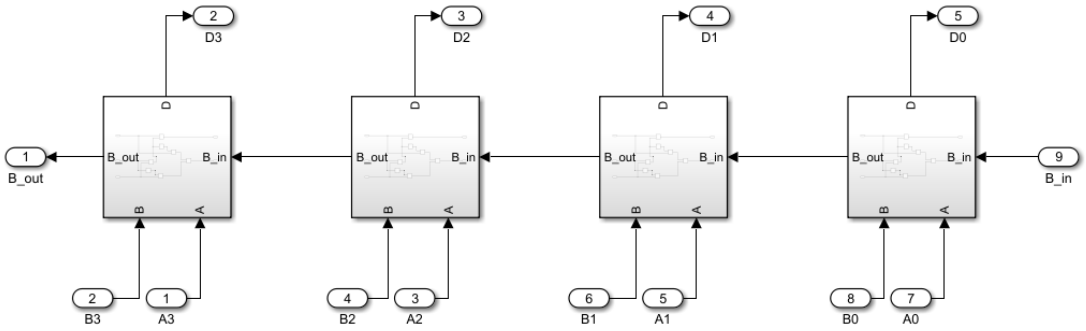


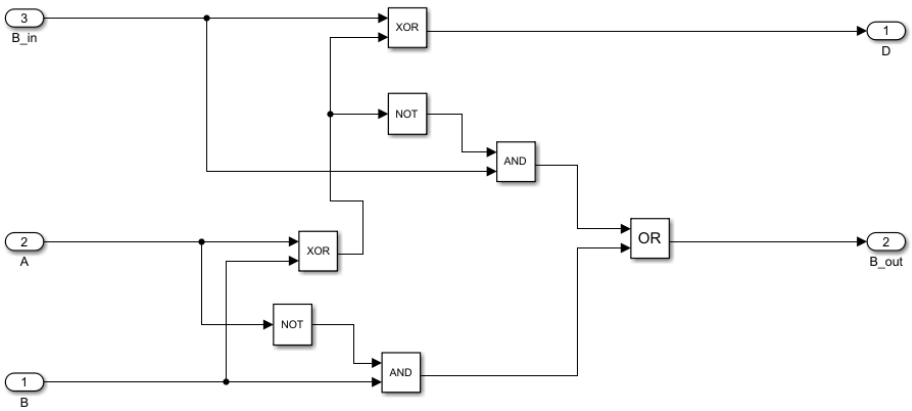
7. ALU

There are 4 operations in this ALU: Addition, Subtraction, Bitwise AND operation, Bitwise OR operation. One of these is selected using the opcode from the instruction decode subsystem. Each of the operations are written using logic gates. The output is then sent to the Register file for storage

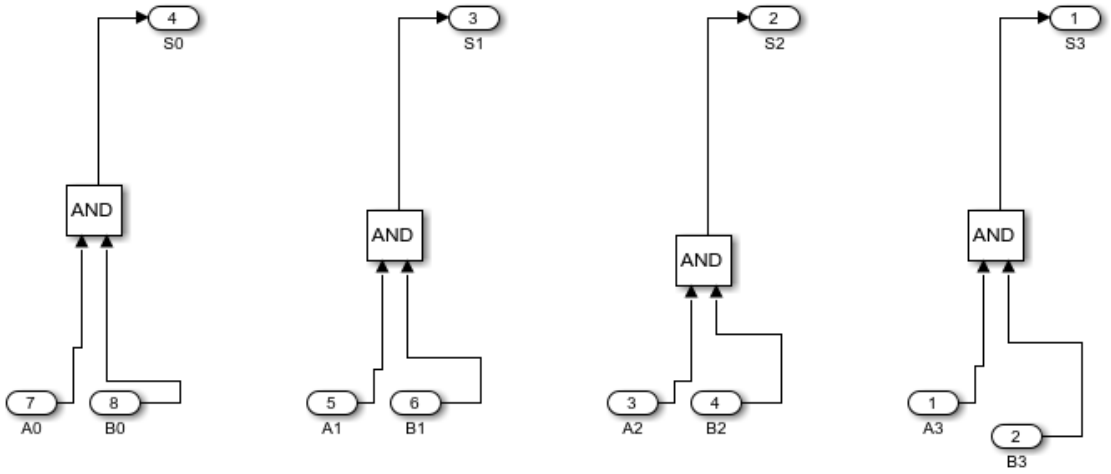


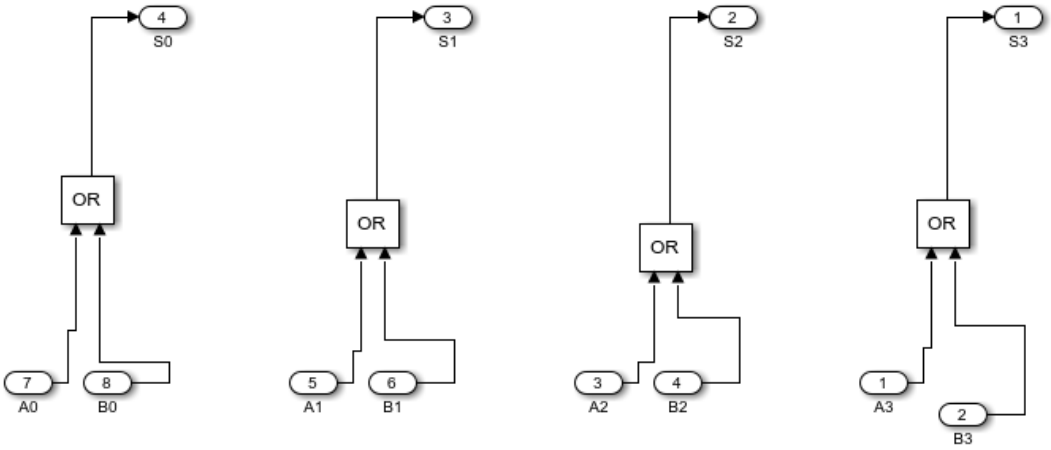
Subtractor:





Bitwise operations:





Source files and more sample images link - [Single Cycle Processor]