Simple General-Purpose Processor

Harshraj Parmar

Introduction

This simple general purpose processor was simulated using the VHDL design environment Quartus II and tested on the Altera Cyclone II EP2C35F672C6 FPGA board. The processor is composed of 4 main components, two storage units, an arithmetic logic unit, a seven segment display unit and a control unit.

The storage units are latches which will be used as temporary memory for the input data, 2 sets of data will be inputted so there are 2 latches. The control unit is composed of a finite state machine and a 4 to 16 decoder. The finite state machine will cycle through states 0 to 8, and pass the current state to the 4 to 16 decoder. The decoder will then decode the input and output it as a microcode instruction which is passed into the arithmetic logic unit. The arithmetic logic unit takes the microcode input from the 4 to 16 decoder and selects a function which runs operations on the inputted data from the latches and will send out the output of this data. Lastly the seven segment display will be used to display the results of the arithmetic logic unit.

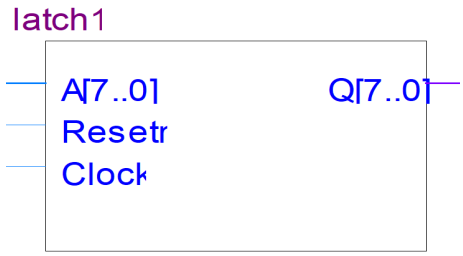
Components

Latch

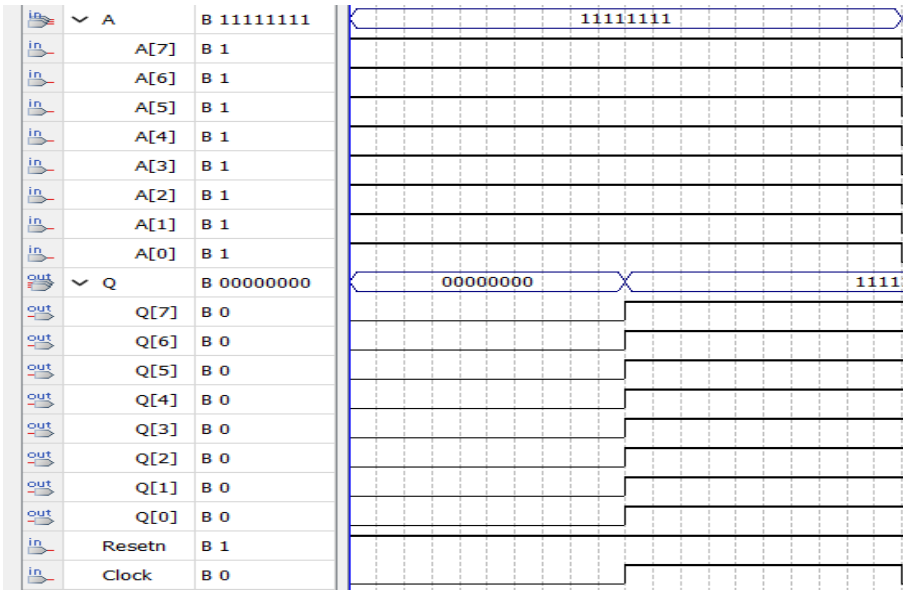
A latch is a component which is able to store bits of data until the latch is updated. For the purposes of the processor we are using a D latch as a temporary storage. It has 3 inputs and 1 output, the A input is used to input data, the clock input is used to update the data, the reset input is used to reset the data and the Q output will output the data. Below is the waveform of the latch. Latch 1 is used for input data A[7..0] and Latch 2 is used for input data B[7..0].

Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1

Truth Table



Block Diagram

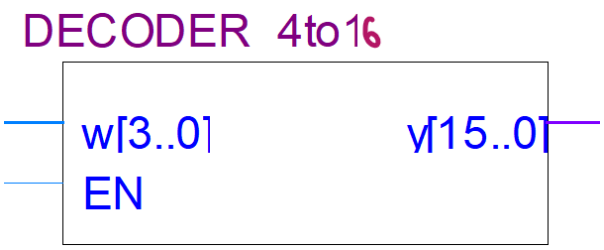


Waveform Latch

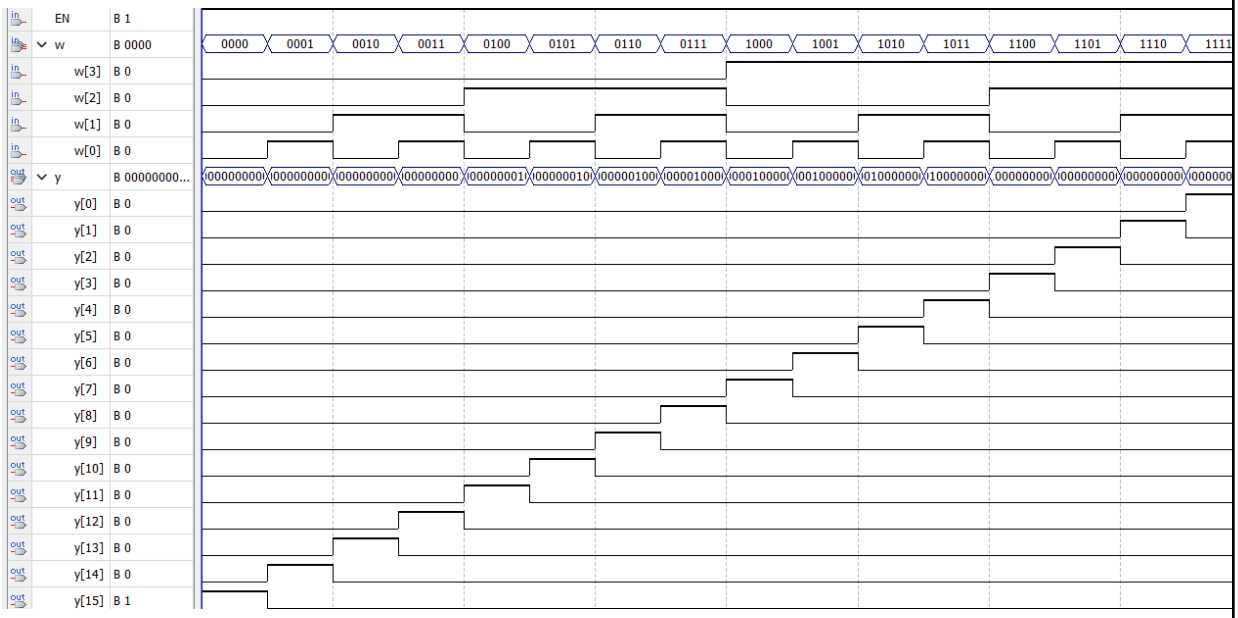
4 To 16 Decoder

A 4 to 16 decoder is a component which will take a 4 bit input and output a 16 bit output. With 4 bits you have a possibility of having 2 to the power of 4 possible outputs, so 16 different possible outputs. Depending on the 4 bit input it will select 1 of the 16 outputs.

On the component itself there are 2 inputs and 1 output, the decoder component has an enabler input which is used to turn the component on and off, it also has the w input which is then decoded into a 16 bit output.



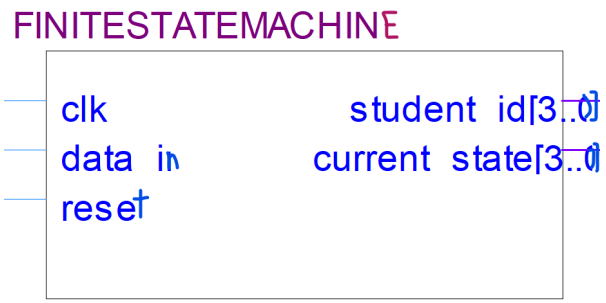
Block Diagram



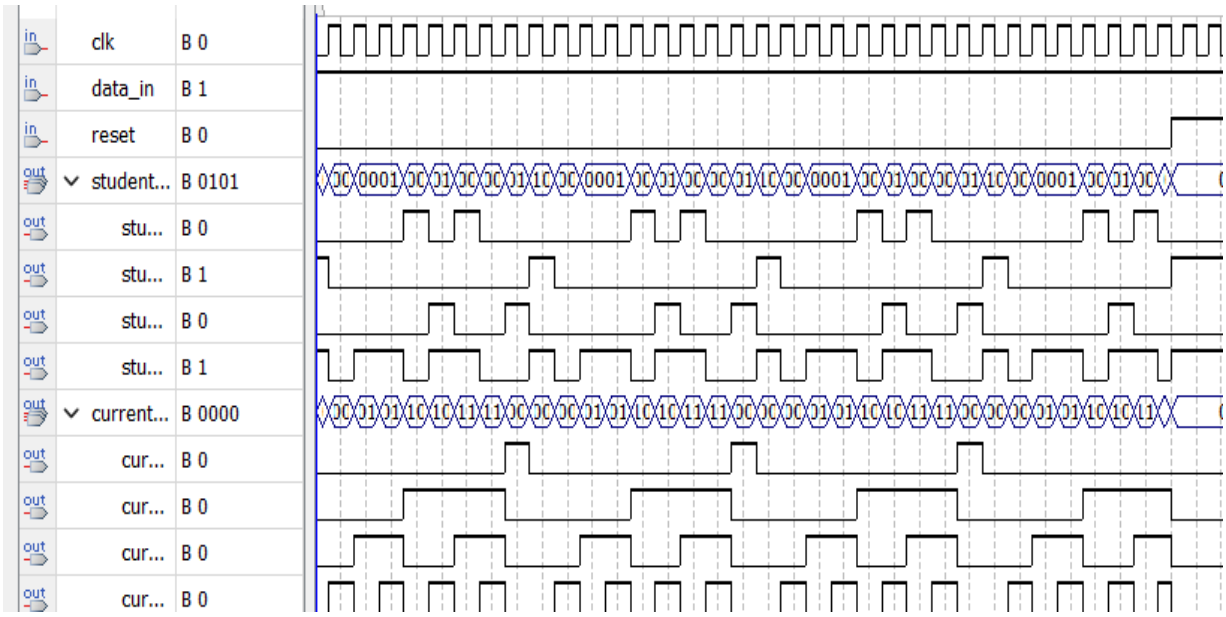
Waveform Decoder

Finite State Machine

A finite state machine is a component which cycles through states depending on its input, and as it cycles through the state each state will have an output. In the case of this processor each state will go from 0 to 8 and will output digits of my student id at each state. In order to change states you must clock the finite state machine. There is a clock input to change to the next state, a data in and reset input, data in is used to cycle through the states and the reset input is used to reset back to state 0.

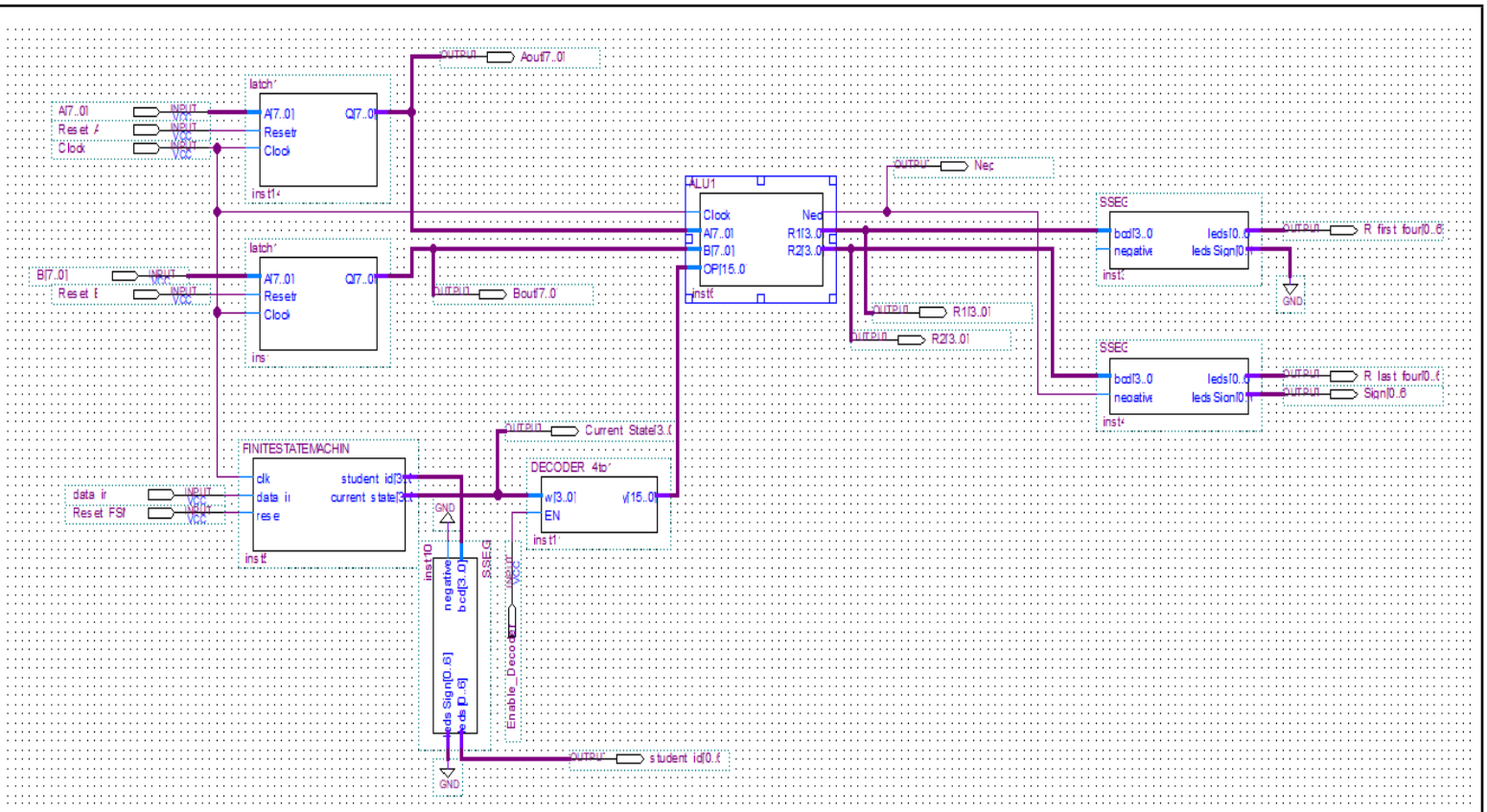


Block Diagram



Waveform FSM

ALU Problem Set 1: A=(39) B=(02)



Block Diagram of Problem Set 1

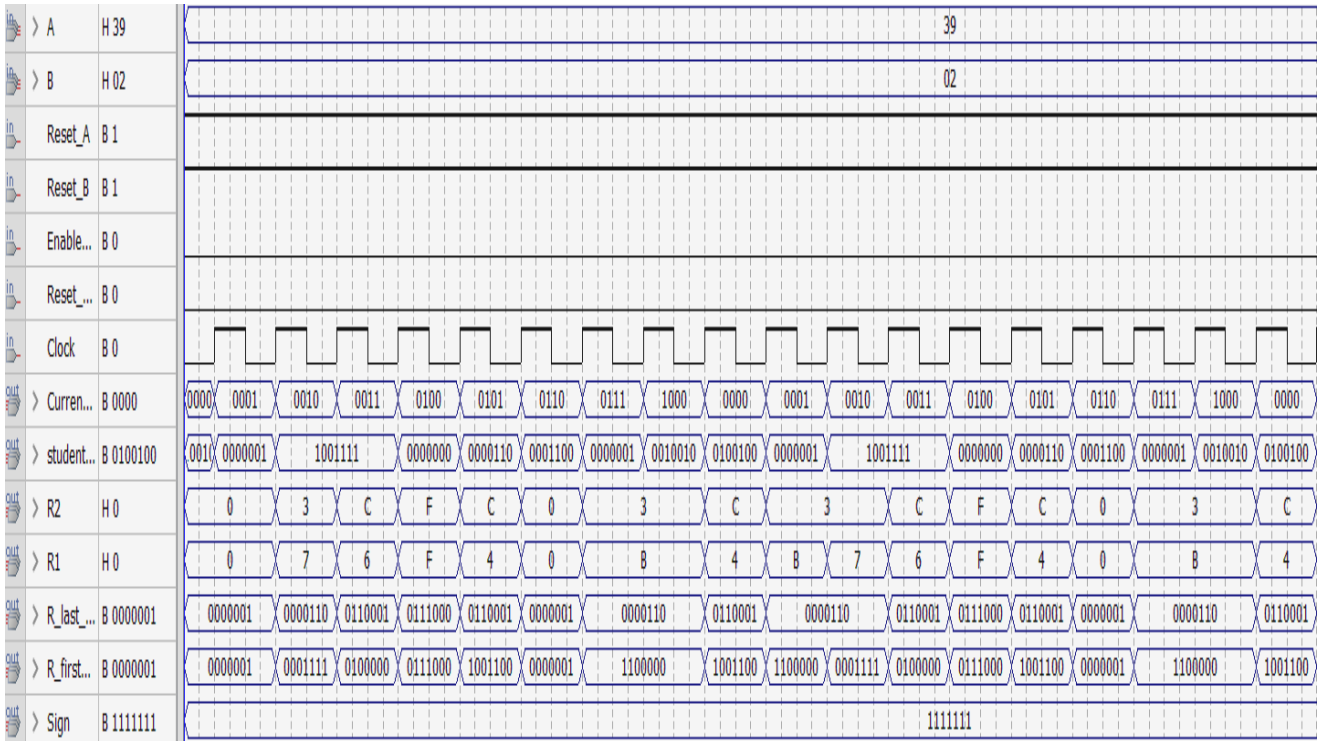
Description:

The purpose is to create an arithmetic logic unit which will perform a selected operation on input data based on the microcode selector. There are 2 latches which store the input data, this data is then input into the ALU. There is a finite state machine which will cycle through its states and will send a 4 bit current state output to the input of the 4 to 16 decoder which will then decode that input into the microcode selector and send it into the ALU. Seven segment displays are then used to display the current state of the FSM, as well as the results of the ALU.

Purpose of Input/Output:

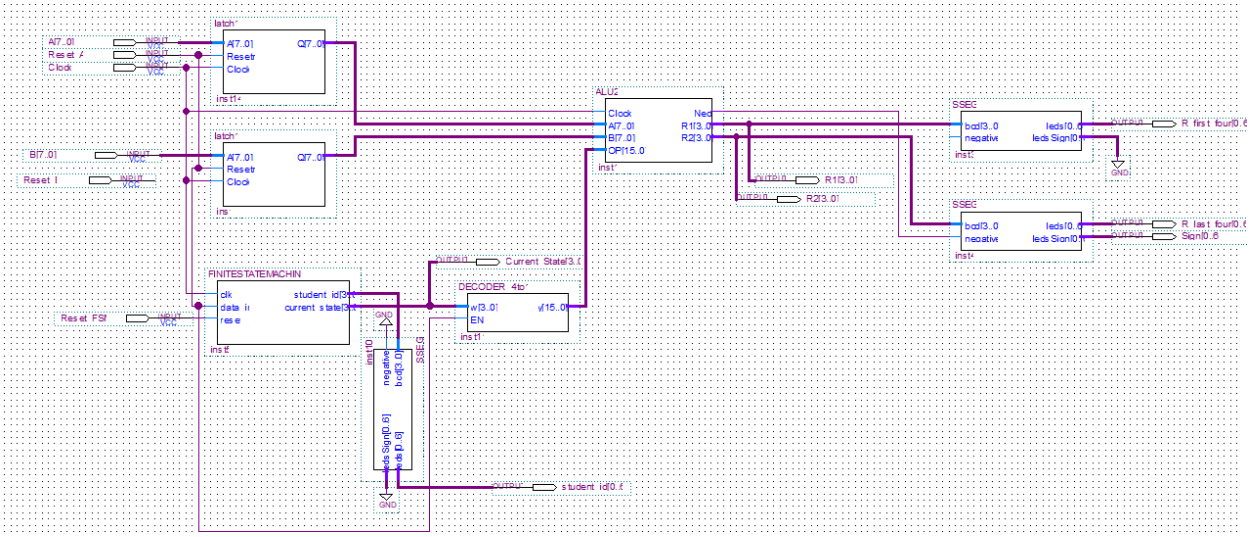
A/B[7..0] input	- Input data with length of 8 bits
Reset_A/B input	- Resets input data to (00000000)
Q[7..0] output	- Will output the input data from latch
Clock input	- This is the input which clocks the schematic
Data_In input	- This input is used to cycle the FSM
Reset_FSM input	- Resets FSM to state 0
student_id [3..0] output	- Outputs student id digits
Current_state[3..0] output	- outputs the current state
W[3..0] input	- Used to select microcode in decoder
EN input	- Used to enable decoder
Y[15..0] output	- Output which is sent to ALU (Microcode)
OP[15..0] input	- Input used to take in the microcode
R1[3..0] output	- first four bit of result from ALU
R2[3..0] output	- last four bit of result from ALU
Leds[3..0] output	- Seven Segment display for digit
Leds_Sign[3..0] output	- Seven segment display for sign

Function #	Microcode ALU_1	Operation
1	0000000000000001	sum(A,B)
2	0000000000000010	diff(A,B)
3	0000000000000100	A inverse
4	0000000000001000	Nand
5	0000000000010000	Nor
6	0000000000100000	And
7	0000000001000000	Xor
8	0000000010000000	Or
9	0000000100000000	Xnor



Waveform for ALU1

ALU Problem Set 2: A=(39) B=(02)



Block Diagram of Problem Set 2

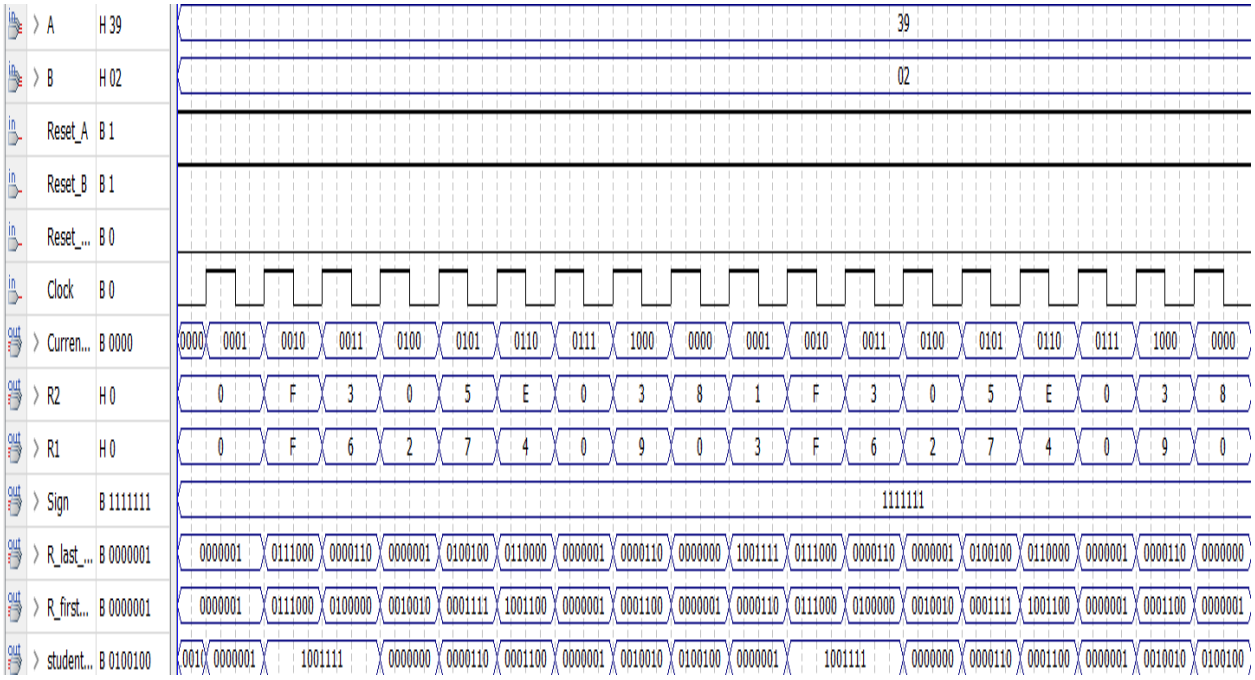
Description:

The purpose is to create an arithmetic logic unit which will perform a selected operation on input data based on the microcode selector. There are 2 latches which store the input data, this data is then input into the ALU. Inside of the ALU for problem set 2 there are custom functions I created which allow it to take input and perform specific operations and then to display it onto the seven segment display.

Purpose of Input/Output:

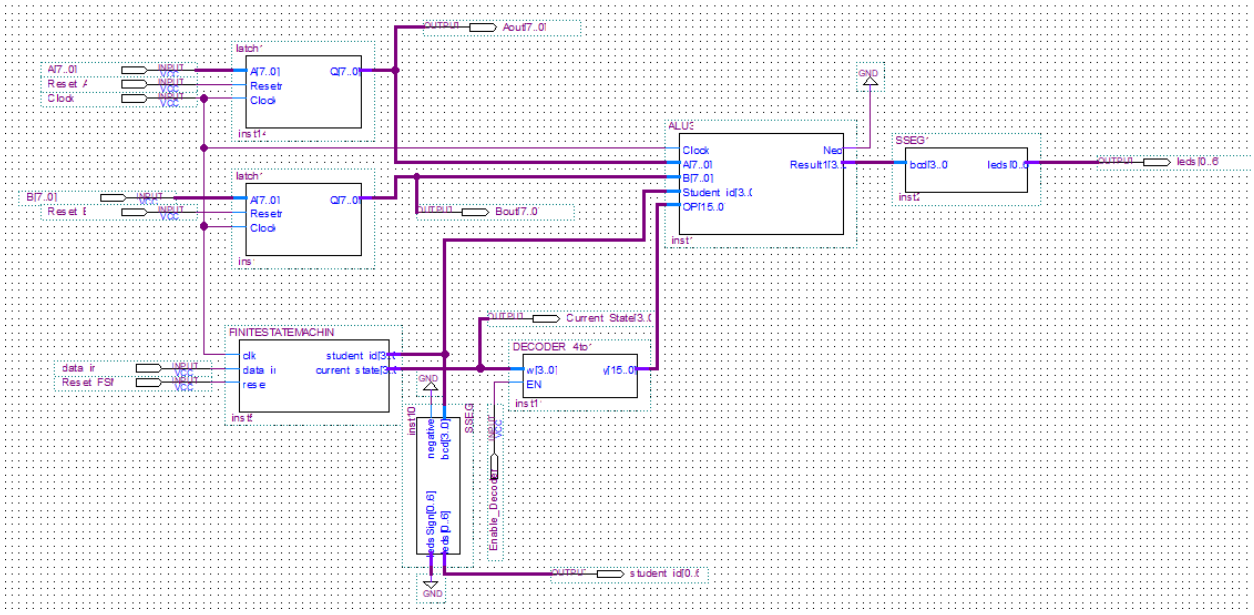
A/B[7..0] input	- Input data with length of 8 bits
Reset_A/B input	- Resets input data to (00000000)
Q[7..0] output	- Will output the input data from latch
Clock input	- This is the input which clocks the schematic
Data_In input	- This input is used to cycle the FSM
Reset_FSM input	- Resets FSM to state 0
student_id [3..0] output	- Outputs student id digits
Current_state[3..0] output	- outputs the current state
W[3..0] input	- Used to select microcode in decoder
EN input	- Used to enable decoder
Y[15..0] output (Microcode)	- Output which is sent to ALU
OP[15..0] input	- Input used to take in the microcode
R1[3..0] output	- first four bit of result from ALU
R2[3..0] output	- last four bit of result from ALU
Leds[3..0] output	- Seven Segment display for digit
Leds_Sign[3..0] output	- Seven segment display for sign

Function #	Microcode ALU_2	Operation
1	0000000000000001	Odd A replaced with Odd B
2	0000000000000010	Nand(A,B)
3	0000000000000100	Sum(A,B) - 5
4	0000000000001000	2’s Complement of B
5	0000000000010000	Invert Even B
6	0000000000100000	Shift A to left by 2
7	0000000001000000	Null
8	0000000010000000	2’s Complement of A
9	0000000100000000	Rotate B by 2 bits to right



Waveform for ALU2

ALU Problem Set 3: A=(39) B=(02)



Block Diagram of Problem Set 3

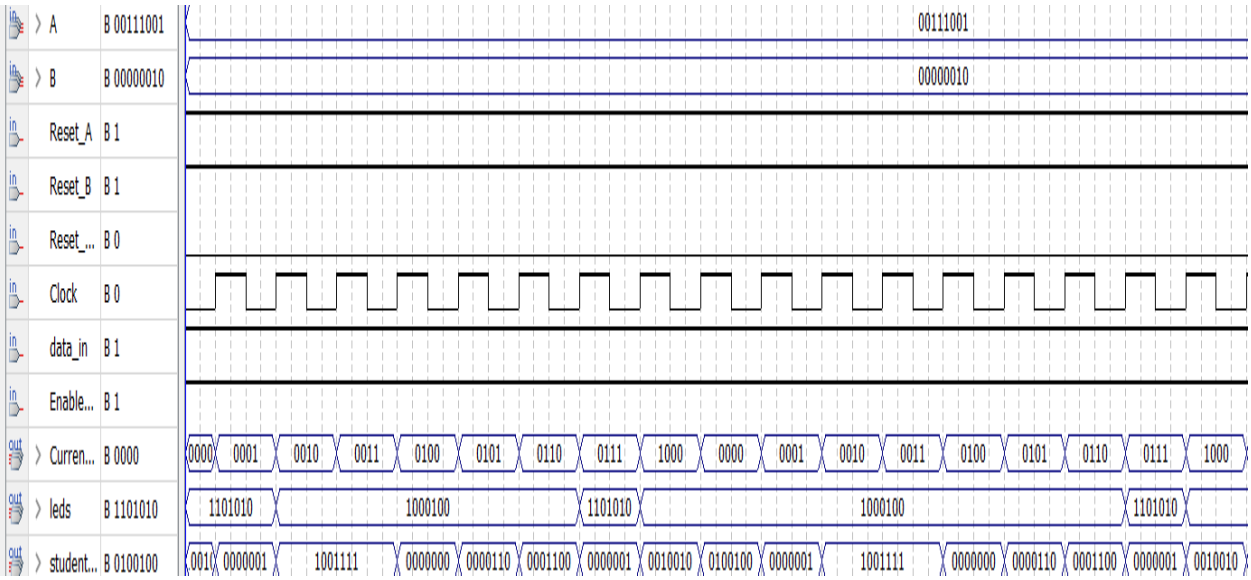
Description:

The purpose is to create an arithmetic logic unit which will perform a selected operation on input data based on the microcode selector. There are 2 latches which store the input data, this data is then input into the ALU. In problem set 3 the seven segment display will display either a Y or an N depending on if a digit from the A is greater than the student number, which is why we need to add an additional input student number into the ALU for this problem set.

Purpose of Input/Output:

A/B[7..0] input	- Input data with length of 8 bits
Reset_A/B input	- Resets input data to (00000000)
Q[7..0] output	- Will output the input data from latch
Clock input	- This is the input which clocks the schematic
Data_In input	- This input is used to cycle the FSM
Reset_FSM input	- Resets FSM to state 0
student_id [3..0] input	- inputs student id digits to ALU
Current_state[3..0] output	- outputs the current state
W[3..0] input	- Used to select microcode in decoder
EN input	- Used to enable decoder
Y[15..0] output (Microcode)	- Output which is sent to ALU
OP[15..0] input	- Input used to take in the microcode
Result[3..0] output	- first four bit of result from ALU
Leds[3..0] output	- Seven Segment display for Y/N

Function #	Microcode ALU_3	Operation
1	0000000000000001	Display Y if greater than student_id, N otherwise
2	0000000000000010	Display Y if greater than student_id, N otherwise
3	0000000000000100	Display Y if greater than student_id, N otherwise
4	0000000000001000	Display Y if greater than student_id, N otherwise
5	0000000000010000	Display Y if greater than student_id, N otherwise
6	0000000000100000	Display Y if greater than student_id, N otherwise
7	0000000001000000	Display Y if greater than student_id, N otherwise
8	0000000010000000	Display Y if greater than student_id, N otherwise
9	0000000100000000	Display Y if greater than student_id, N otherwise



Waveform for ALU3

Conclusion

In conclusion, using various different components such as a finite state machine, latches, decoder, seven segments and an arithmetic logic unit. Each component by itself performs a specific task that is important. The finite state machine is necessary to cycle through states which has the effect of cycling through the operations performed by the ALU. The latches are important as they will store the input data which is necessary information that is needed in order to perform an operation. The seven segment display is important as it is a way to visualize what is actually happening inside of the ALU and lastly the ALU is important as it is able to perform different tasks. In the end each unique unit all comes together to form a working processor.

References

[1] Toronto Metropolitan University COE328: Lab 6 - lab manual