The goal of the assignment is to develop an understanding for the different Verilog concepts regarding full system level design. The project is split into two parts, with this first part focusing on the DataPath and Controller implementation of the processing unit.

***Project Part 1-A***

The Verilog code for this section can be found in the Pt A folder of the ZIP file submitted:

*AC.v*

*ALU.v*

*Multiplier.v*

*DataPath.v*

*DataPathTester.v*

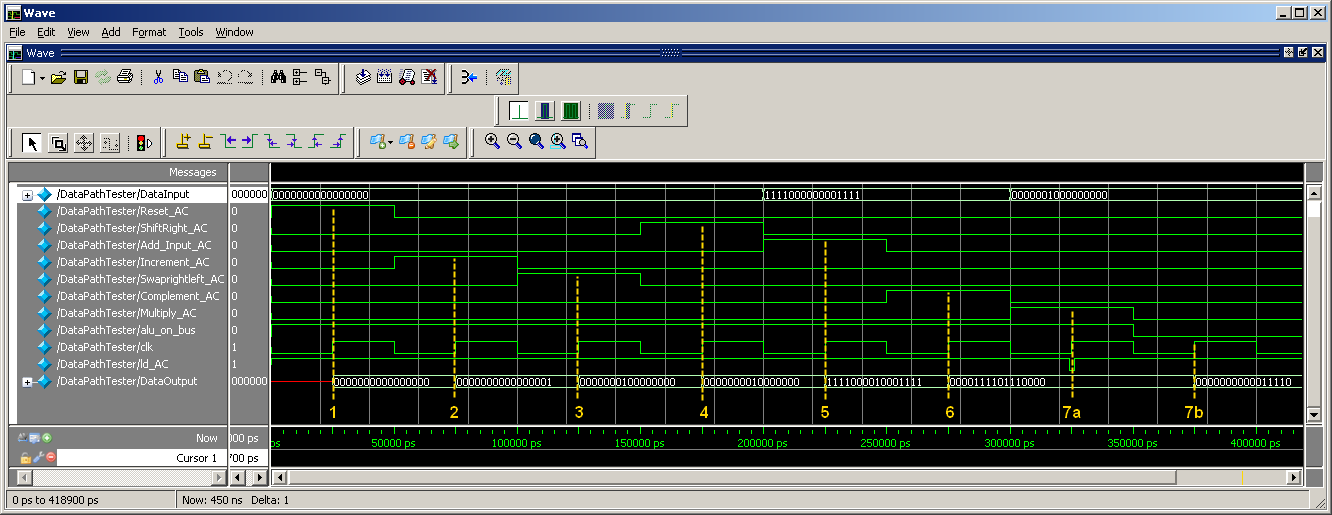
The following diagram is a visual representation of the DataPath created for the processing unit:

**Processing Unit – DataPath Components and Interconnections**



After building each of the individual modules – the accumulator, the multiplier, and the arithmetic logic unit (ALU) – a testbench was created to ensure that all signals were being passed and that the ALU was performing operations correctly. Without the implementation of a Controller, all operations happen on the positive edge of the clock when the control signal is seen as asserted. Once the Controller is implemented, state changes will result in different clock delays.

**Simulation Waveform Output of DataPathTester**



@ 1 : *Reset\_AC* = 1 🡺 DataOutput = 0000\_0000\_0000\_0000 **CORRECT**

@ 2 : *Increment\_AC* = 1 🡺 DataOutput = 0000\_0000\_0000\_0001 **CORRECT**

@ 3 : *Swaprightleft\_AC* = 1 🡺 DataOutput = 0000\_0001\_0000\_0000 **CORRECT**

@ 4 : *ShiftRight\_AC* = 1 🡺 DataOutput = 0000\_0000\_1000\_0000 **CORRECT**

@ 5 : *Add\_Input\_AC* = 1 🡺 DataOutput = 1111\_0000\_0000\_1111 + 0000\_0000\_1000\_000

DataOutput = 1111\_0000\_1000\_1111 **CORRECT**

@ 6 : *Complement\_AC* = 1 🡺 DataOutput = 0000\_1111\_0111\_0000 **CORRECT**

@ 7a: *Multiply\_AC* = 1 🡺 DataOutput = <previous value> = 0000\_1111\_0111\_0000

@ 7b: *Reset\_AC* = 1 🡺 DataOutput = 0000\_0010 \* 0000\_1111

DataOutput = 0000\_0000\_0001\_1110 **CORRECT**

Through a simple testbench, the handshaking between all components (without a Controller) works as intended. This concludes the analysis for the Project Part 1-A.

***Project Part 1-B***

The Verilog code for this section can be found in the Pt B folder of the ZIP file submitted:

*Controller.v*

The following diagram is a visual representation of the Controller created for the processing unit:

**Processing Unit – Controller State Diagram and Interconnections**

Both the Controller and the DataPath are tested in the following section in preparation for the second part of the project. This concludes the analysis for the Project Part 1-B.

***Project Part 1-C***

The Verilog code for this section can be found in the Pt C folder of the ZIP file submitted:

*AC.v*

*ALU.v*

*Multiplier.v*

*DataPath.v*

*Controller.v*

*aluCPU.v*

*aluCPUTester.v*

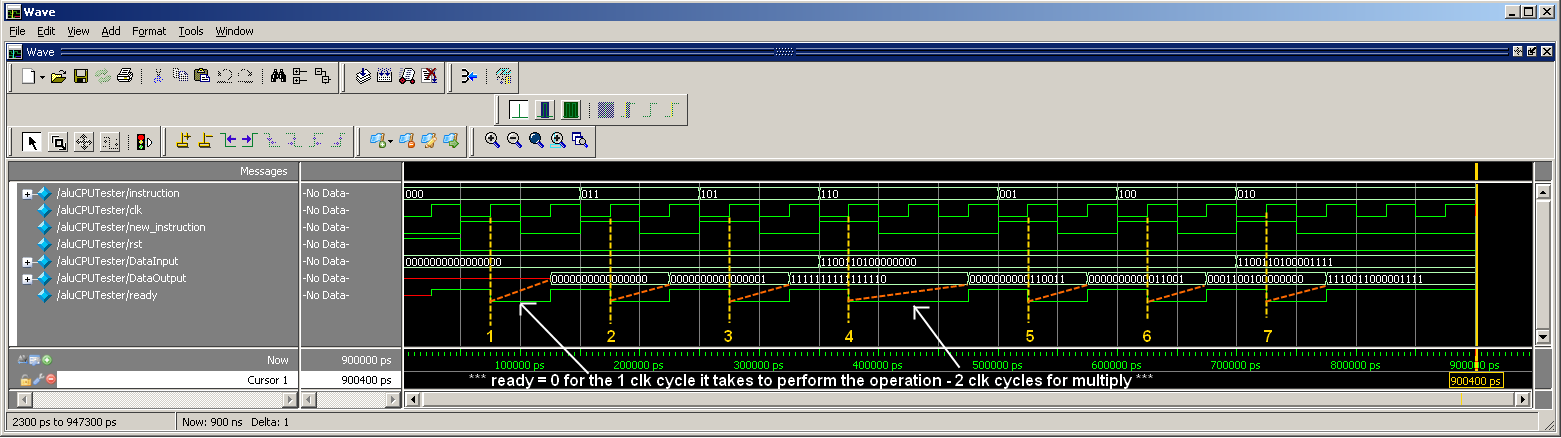
The following diagram is a visual representation of the full processing unit, with all interconnections between the DataPath and the Controller:

**ALU Processing Unit – Overall System Diagram**



All of these interconnections and correct assertions can be seen by combing through all the comments in my Verilog code. In preparation for Pt 2 of this project, I created a simple Testbench to ensure functionality of all instructions and both the Controller and DataPath in one fully integrated ALU Processing Unit:

**Simulation Waveform Output of aluCPUTester**



\* Start by having *rst = 1* to set the initial state of the aluCPU.

@ 1 : *instruction* = *Reset\_AC*

🡺 DataOutput = 0000\_0000\_0000\_0000 **after 1 Clk** **CORRECT**

@ 2 : *instruction* = *Increment\_AC*

🡺 DataOutput = 0000\_0000\_0000\_0001 **after 1 Clk** **CORRECT**

@ 3 : *instruction* = *Complement\_AC*

🡺 DataOutput = 1111\_1111\_1111\_1110 **after 1 Clk** **CORRECT**

@ 4 : *instruction = Multiply\_AC*

🡺 DataInput = 1100\_1101\_0000\_0000

>> MultiplicationOp1 = DataInput [15:8] = 1100\_1101 🡪 (D) -51

>> MultiplicationOp2 = Accumulator [15:8] = 1111\_1111 🡪 (D) -1

🡺 DataOutput = 0000\_0000\_0011\_0011 🡪 (D) 51 **after 2 Clk** **CORRECT**

@ 5 : *instruction = ShiftRight\_AC*

🡺 DataOutput = 0000\_0000\_0001\_1001 **after 1 Clk CORRECT**

@ 6 : *instruction = Swaprightleft\_AC*

🡺 DataOutput = 0001\_1001\_0000\_0000 **after 1 Clk**  **CORRECT**

@ 7 : *instruction = Add\_Input\_AC*

🡺 DataInput = 1100\_1101\_0000\_1111

🡺 DataOutput = 0001\_1001\_0000\_0000 +

1100\_1101\_0000\_1111

DataOutput = 1110\_0110\_0000\_1111 **after 1 Clk CORRECT**

Through a simple testbench, the handshaking between the full system (DataPath and Controller) works as intended. This concludes the analysis for the Project Part 1-C.