Objective:

The lab #4 had two parts to do. In part 1, the objective of the lab was to build a 4-bit CPU that executes the given set of instructions. The ACC and PC were displayed as shown in the given figure 7-i and each instruction was two bytes long in the given format. To test the CPU, one DIP switch was used to enter both bytes of the instruction. If the LSB of PC is 0 then DIP switch is the BYTE 0 or op-code byte and if the LSB is 1 then DIP switch is BYTE 1 or source/destination code. In part 2, the full schematic was to be provided and simulate of 4-bit CPU that executes the instruction set in part 1 but where the instructions to be executed are stored in RAM and then executed.

Components:

-One 74LS157 Quad 2/1 Data Selector (MUX)

-TWO 74LS00 Quad 2-input NAND gate

-One 7489 RAM

-TWO 74LS174 Hex D-Type Flip-Flop with Clear

-One 74LS181 Arithmetic Logic Unit/Function Generator

-Two 74LS247 BCD to 7-Segment Decoder/Driver

-One 17TOGSD-M SPDT (On)-(On) Toggle Switch

-Two 17DIP8SS 8-Switch DIP Switch Sets

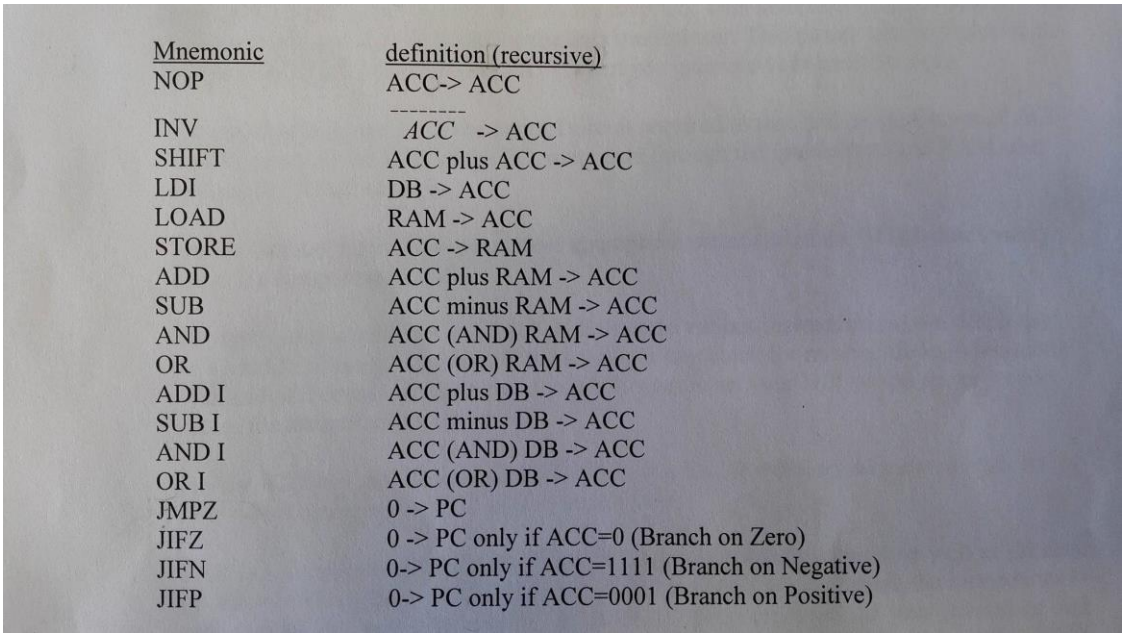
-One 74LS569 Four-Bit Up/Down Counter

-5V Power Supply

-Four 1000 Ohm Resistors

Experimental Approach:

The CPU was build should execute the following instruction:



Mnemonic	definition (recursive)
NOP	ACC-> ACC
INV	$\overline{ACC}$ -> ACC
SHIFT	ACC plus ACC -> ACC
LDI	DB -> ACC
LOAD	RAM -> ACC
STORE	ACC -> RAM
ADD	ACC plus RAM -> ACC
SUB	ACC minus RAM -> ACC
AND	ACC (AND) RAM -> ACC
OR	ACC (OR) RAM -> ACC
ADD I	ACC plus DB -> ACC
SUB I	ACC minus DB -> ACC
AND I	ACC (AND) DB -> ACC
OR I	ACC (OR) DB -> ACC
JMPZ	0 -> PC
JIFZ	0 -> PC only if ACC=0 (Branch on Zero)
JIFN	0-> PC only if ACC=1111 (Branch on Negative)
JIFP	0-> PC only if ACC=0001 (Branch on Positive)

There are 16 input , 8 for BYTE0(even)[MUX,W,M,Cn,S3,S2,S1,S0], 8 for BYTE1(odd) [DB3,DB2,DB1,DB0,RAM3,RAM2,RAM1,RAM0]. “MUX” determines execute RAM or the date bus (DB) has the access to the ALU. “W” is the write enable for RAM. “Cn, M, S” are the 74181 ALU control lines. “DB” is date bus. “RAM0-RAM3” is the addresses in the ram.

Function Table

Mode Select Inputs				Active LOW Operands & F_n Outputs		Active HIGH Operands & F_n Outputs	
				Logic	Arithmetic (Note 2)	Logic	Arithmetic (Note 2)
S3	S2	S1	S0	(M = H)	(M = L) (C_n = L)	(M = H)	(M = L) (C_n = H)
L	L	L	L	$\overline{A}$	A minus 1	$\overline{A}$	A
L	L	L	H	$\overline{AB}$	AB minus 1	$\overline{A}+\overline{B}$	A + B
L	L	H	L	$\overline{A}+B$	$A\overline{B}$ minus 1	$\overline{A}B$	$A+\overline{B}$
L	L	H	H	Logic 1	minus 1	Logic 0	minus 1
L	H	L	L	$\overline{A}+\overline{B}$	A plus $(A+\overline{B})$	$\overline{AB}$	A plus $A\overline{B}$
L	H	L	H	$\overline{B}$	AB plus $(A+\overline{B})$	$\overline{B}$	$(A+B)$ plus $A\overline{B}$
L	H	H	L	$\overline{A}\oplus\overline{B}$	A minus B minus 1	$A\oplus B$	A minus B minus 1
L	H	H	H	$A+\overline{B}$	$A+\overline{B}$	$A\overline{B}$	AB minus 1
H	L	L	L	$\overline{A}B$	A plus (A + B)	$\overline{A}+B$	A plus AB
H	L	L	H	$A\oplus B$	A plus B	$\overline{A}\oplus\overline{B}$	A plus B
H	L	H	L	B	$A\overline{B}$ plus (A + B)	B	$(A+\overline{B})$ plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logic 0	A plus A (Note 1)	Logic 1	A plus A (Note 1)
H	H	L	H	$A\overline{B}$	AB plus A	$A+\overline{B}$	$(A+B)$ plus A
H	H	H	L	AB	$A\overline{B}$ minus A	A + B	$(A+\overline{B})$ plus A
H	H	H	H	A	A	A	A minus 1

Note 1: Each bit is shifted to the next most significant position.

Note 2: Arithmetic operations expressed in 2s complement notation.

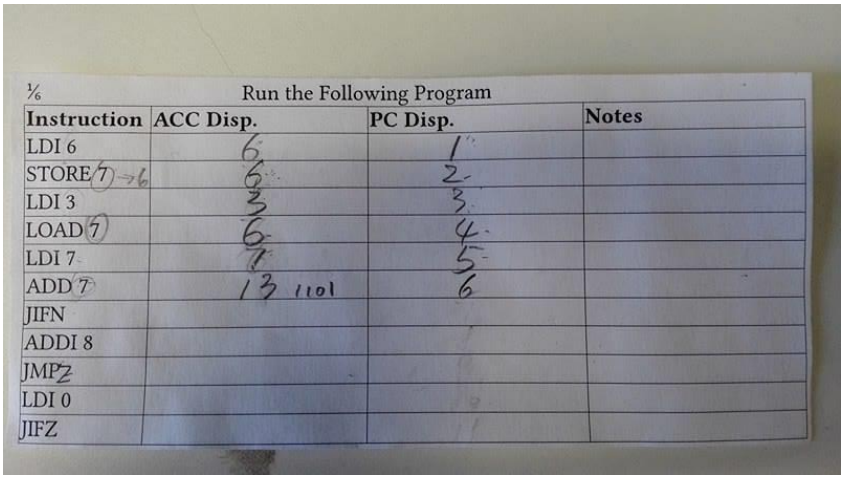
The definition is from the function table of ALU. For example, for Load instruction, if I load an 8 into address 4, the result is going to be:

MUX	W	Cn	M	S3	S2	S1	S0	DB3	DB2	DB1	DB0	RAM3	RAM2	RAM1	RAM0
1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0

Input these 16 values for the DIP switch and then use the clock one time, it will show the result of "8" "1" in the LED display.

Results:

A 4-bit CPU was built that executes the given set of instructions. The ACC and PC were displayed as shown in the given figure 7-i and each instruction was two bytes long. To test the CPU, one DIP switch was used to enter both bytes of the instruction. If the LSB of PC is 0 then DIP switch is the BYTE 0 or op-code byte and if the LSB is 1 then DIP switch is BYTE 1 or source/destination code. After testing from the instruction, there are 14 works correctly and 4 instructions don't work which are (JMPZ, JIFZ, JIFN, and JIFP). The PC works correctly.



Run the Following Program

Instruction	ACC Disp.	PC Disp.	Notes
LDI 6	6	1	
STORE 7	6	2	
LDI 3	3	3	
LOAD 7	6	4	
LDI 7	7	5	
ADD 7	13 1101	6	
JIFN			
ADDI 8			
JMPZ			
LDI 0			
JIFZ			

1st clock cycle: load (DB) a value 6 in ACC.

2nd clock cycle: store the value 6 in the ram address 7.

3rd clock cycle: load (DB) a value 3 in ACC.

4th clock cycle: load the value already store in address 7 (which is 6 according to second step)

5th clock cycle: load (DB) a value 7 in ACC.

6th clock cycle: add value 7 and the value in the ram (address 7whcih is 6) so 7+6=13.

Conclusion:

A 4-bit CPU was built that executes the given set of instructions. The ACC and PC were displayed as shown in the given figure 7-i and each instruction was two bytes long. To test the CPU, one DIP switch was used to enter both bytes of the instruction. If the LSB of PC is 0 then DIP switch is the BYTE 0 or op-code byte and if the LSB is 1 then DIP switch is BYTE 1 or source/destination code. After testing from the instruction, there are 14 works correctly and 4 instructions don't work which are (JMPZ, JIFZ, JIFN, and JIFP). The PC works correctly.

Circuit Schematic: