EEL 3701C
PI: Keith Khadar
Section: 11195
April 9, 2025

Lab 6 Quiz

Time Limit: 60 minutes

Part 1

Modify your Lab 6 Part 2 CPU to start program execution from address $4200. Next you will write a short program based on the pseudocode below. On your scratch paper, hand assemble your program using the template table given in Lab 6 (check the documentation section below). You should make the necessary hardware modifications, create a MIF file, and simulate. Submit the Quartus archive for your updated Lab 6 Prelab (including your new assembly program through Canvas).

```
A = 6
B = 9
loop
{
    A = A * 4
    A = A + B
}
```

Part 2

Imagine you want to add a new instruction to your CPU to occupy opcode 111 which will branch if the contents of RegA are negative when interpreted as a two's complement value. This new instruction is "BNEG". To implement this instruction, you may begin by modifying your ASM. On paper, draw the necessary modifications/new branches to your ASM. You DO NOT need to implement the instruction, just show the new ASM paths. This instruction works very similar to an JMP as it is followed by a branch location at the next address. You should only branch if the condition is true, otherwise, increment the PC to continue past the instruction. Here is an example:

$1230: BNEG
$1231: {address to take if A is neg}
$1232: {instruction to execute next is A is not neg}

EEL 3701C
PI: Keith Khadar
Section: 11195
April 9, 2025

Documentation:

Table 5: Program to assemble.

Addr		Mach Codes	A	B	A	B	A	B	A	B
$2B70	LDAA #7									
	TAB									
	LDAA #3									
	ABA									
	SAR									
	ABA									
	ABA									
	JMP 5									
	ABA									
	TAB									

Table 1: Input source MUXs for Registers A and B.

MSA/ MSB1	MSA/ MSB0	Bus Selected as Input to REGA/REGB
0	0	INPUT Bus
0	1	REGA Bus
1	0	REGB Bus
1	1	OUTPUT Bus

MSC2:0	Action
000	REGA Bus to OUTPUT Bus
001	REGB Bus to OUTPUT Bus
010	complement of REGA Bus to OUTPUT Bus
011	bit wise AND REGA/REGB Bus to OUTPUT Bus
100	bit wise OR REGA/REGB Bus to OUTPUT Bus
101	sum of REGA Bus & REGB Bus to OUTPUT Bus
110	shift REGA Bus left one bit to OUTPUT Bus
111	shift REGA Bus right one bit to OUTPUT Bus **without** sign extension

Table 4. Part 2 instructions.

IR2:0	Instruction	Function
000	TAB	Copy A to B (transfer A to B)
001	ABA	Load A with A plus B plus Cin; update Cout
010	LDAA #data	Load A with input data
011	SAR	Shift A right 1 bit, store in A (**logical**, not arithmetic shift)
100	SAL	Shift A left 1 bit, store in A
101	JMP Addr	Load PC with input address
110	Future use	
111	Future use	

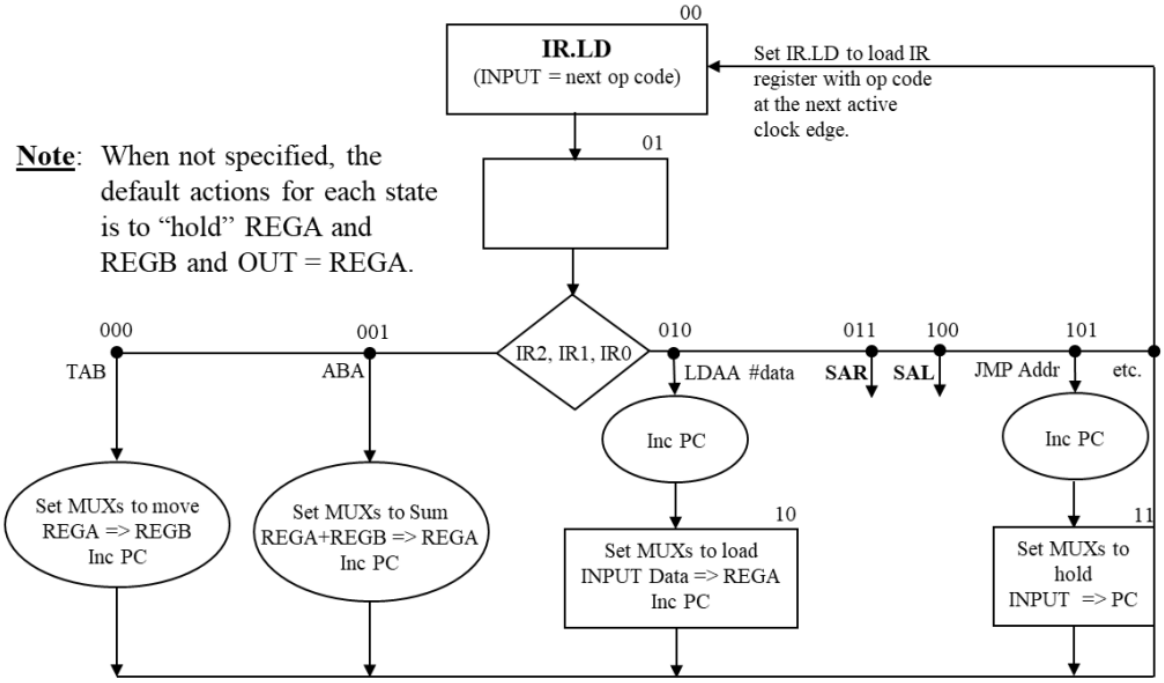

Figure 4. Controller flowchart (**not** an ASM).