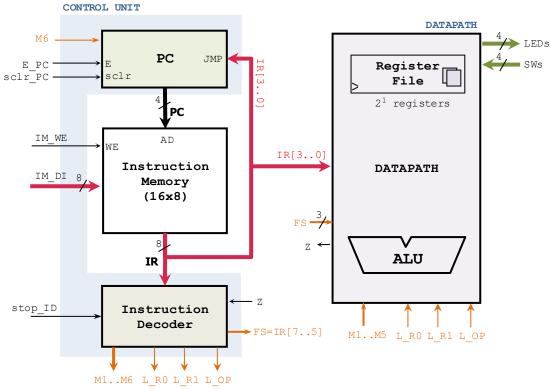
Homework 4

(Due date: April 13th @ 11:59 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (30 PTS)

• "VBC (Very Basic Computer)": 2 registers, 16-word Instruction Memory (IM), 8 bits per instruction (see Notes – Unit 6).



- ✓ Write an assembly program for a counter from 3 to 9: 3, 4, ... 9, 3, 4, ... The count must be shown on the output register (OUT). Use labels to specify any address that an instruction may jump to. You can only have up to 16 instructions.
- ✓ Provide the contents of the Instruction Memory. If some instruction bits are unused, you can assign 0's.

Address	Instruction Memory	Assembly Instruction
0000		
0001		
0010		
0011		
0100		
0101		
0110		
0111		
1000		
1001		
1010		
1011		
1100		
1101		
1110		
1111		

✓ Complete the timing diagram for the execution of the previous program. Use unsigned decimal values for R0, R1, OUT, and PC. Specify IR in hexadecimal. (18 pts) IM: Array of registers. In this case, when reading, output data appears as soon as address is ready. Assumptions: the program is already in IM. Also: sclr PC=0, IM WE=0, stop ID=0. clock resetn E_PC PC 0 IR 00 R0 0 R1 0 OUT 0 FS 000 Μ1 М2 MЗ Μ4 М5 Μ6 clock resetn E PC PC IR R0 R1 OUT FS Μ1 М2 MЗ Μ4 Μ5 Μ6

PROBLEM 2 (18 PTS)

- "Simple Computer": 8 registers, 64-word Instruction Memory (IM), 16 bits per instruction (see Notes Unit 6).
 - ✓ For each of the following instructions, indicate the bits for the instruction fields (if they do not apply, write 'NA'), and the instruction memory contents (if some bits are unused, you can assign 0's). (8 pts)

Address	Instruction	OPCODE	DR	SA	SB	OP	AD	Instruction Memory contents
0	LDI R0,3							
1	LDI R3,2							
2	LDI R4,6							
3	SUB R7,R0,R4							
4	ADI R5,R4,7							
5	ST R5, R7							
6	XOR R4,R7,R5							
7	LD R6, R5							
8	BRN R7,-5							
9	ST R7,R4							
10	JMP RO							

✓ Emulate the execution of the program by completing the state of the registers (use hexadecimal values) after the instruction pointed by PC is executed. Then, complete the PC (use binary values) of the next instruction. Complete it until the JMP instruction is reached (execute this instruction as well). (10 pts)

PC	R0	R3	R4	R5	R6	R7
000000						

PROBLEM 3 (17 PTS)

- "Simple Computer": 8 registers, 64-word IM, 16 bits per instruction (see Notes Unit 6)
 - ✓ Write an assembly program for a counter from 3 to 11: 3,4, 5, 6, 7, ..., 11, 3, 4, ... The count must appear on **RO**. Use labels to specify any address that an instruction may jump to.
 - Note that you can only have up to 64 instructions.
 - The ALU treats data as signed numbers. Thus, data on the registers (R0 R7) and Data Memory (16 bits) is signed.
 - \checkmark Provide the contents of the Instruction Memory (IM) and its addresses.

PROBLEM 4 (20 PTS)

- "PicoBlaze MicroProcessor": 16 registers, 1024-word Instruction Memory, 18 bits per instruction (see Notes Unit 6).
 - Emulate the execution of this program by completing the state of the registers (hexadecimal values) and the status flags (C, Z) after the instruction pointed by PC is executed. Then, complete the PC (hexadecimal values) of the next instruction and update SP (binary values). Complete it until the jump instruction (003) is reached (execute this instruction as well).

Address	Assembly Program
000	main: load s3,04
001	load s4,05
002	call umult
003	jump main
067	umult: load s5,00
068	load s7,08
069	loop: sr0 s4
06A	jump nc, sh p
06B	add s5, s3
06C	sh p: sra s5
06D	
06E	sub s7,01
06F	jump nz, loop
070	return
<u> </u>	•••

PC	SP	s3	s4	s5	S6	s7	С	Z

-				
-				

PROBLEM 5 (15 PTS)

 Attach a printout of your Project Status Report (no more than 3 pages, single-spaced, 2 columns). This report should contain the current status of the project, including more details about the design and its components. You <u>MUST</u> use the provided template (Final Project - Report Template.docx).