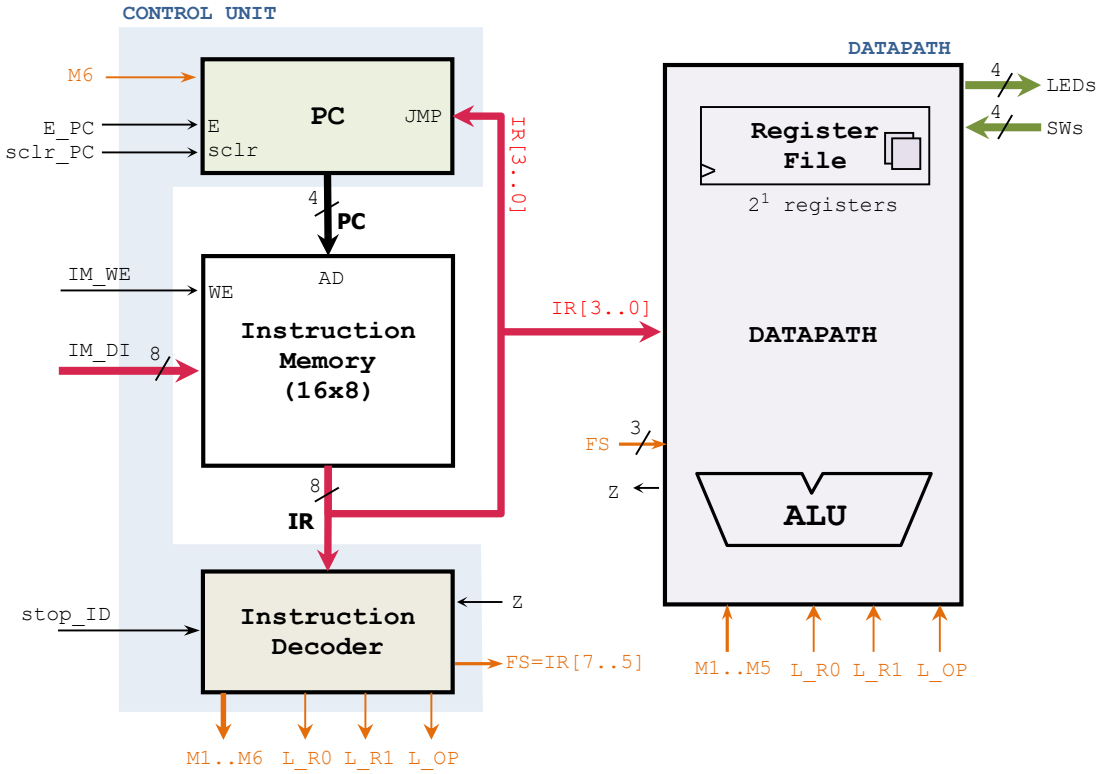
Solutions - Homework 4

(Due date: April 4th @ 7:30 pm)
Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (40 PTS)

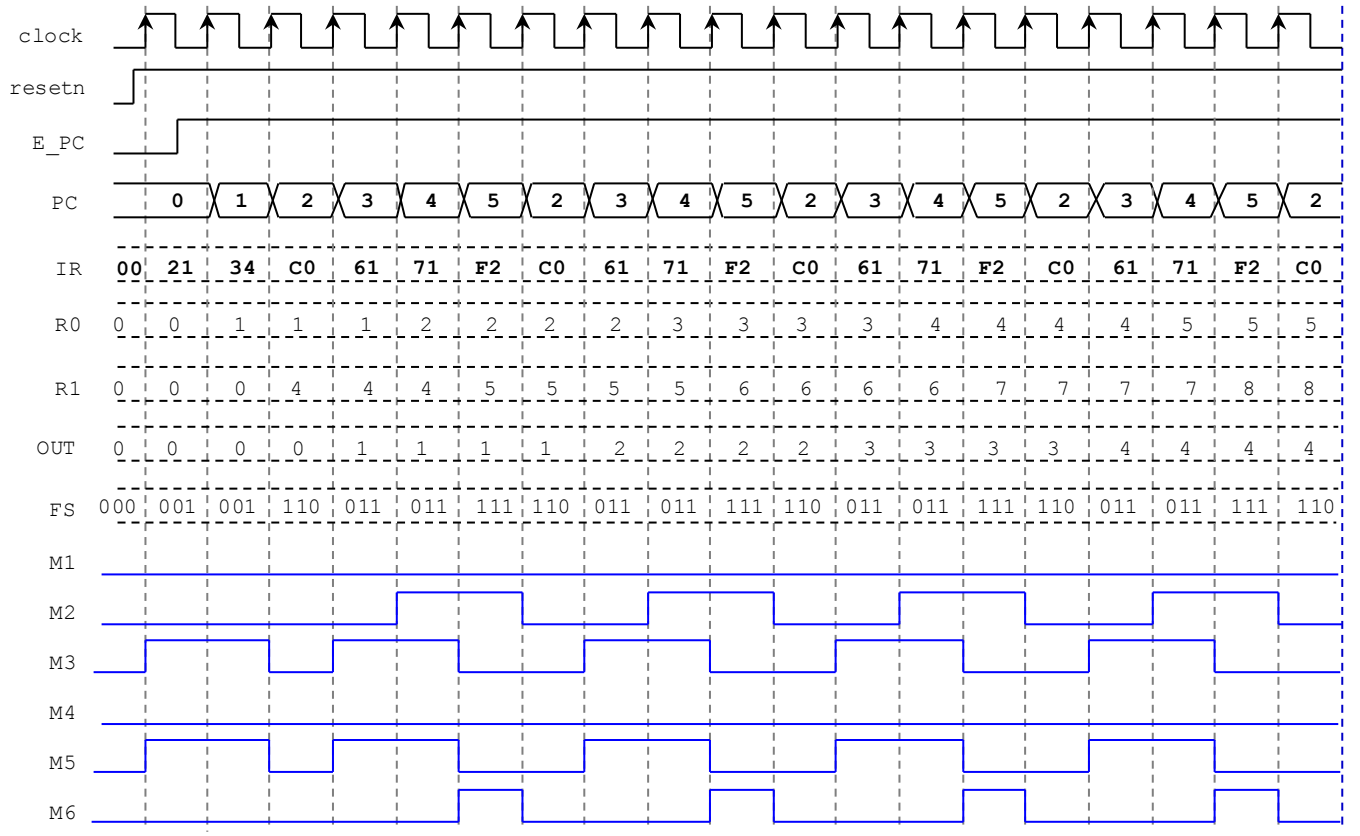
- **"VBC (Very Basic Computer)"**: 2 registers, 16-word instruction memory, 8 bits per instruction. (See Notes – Unit 6).

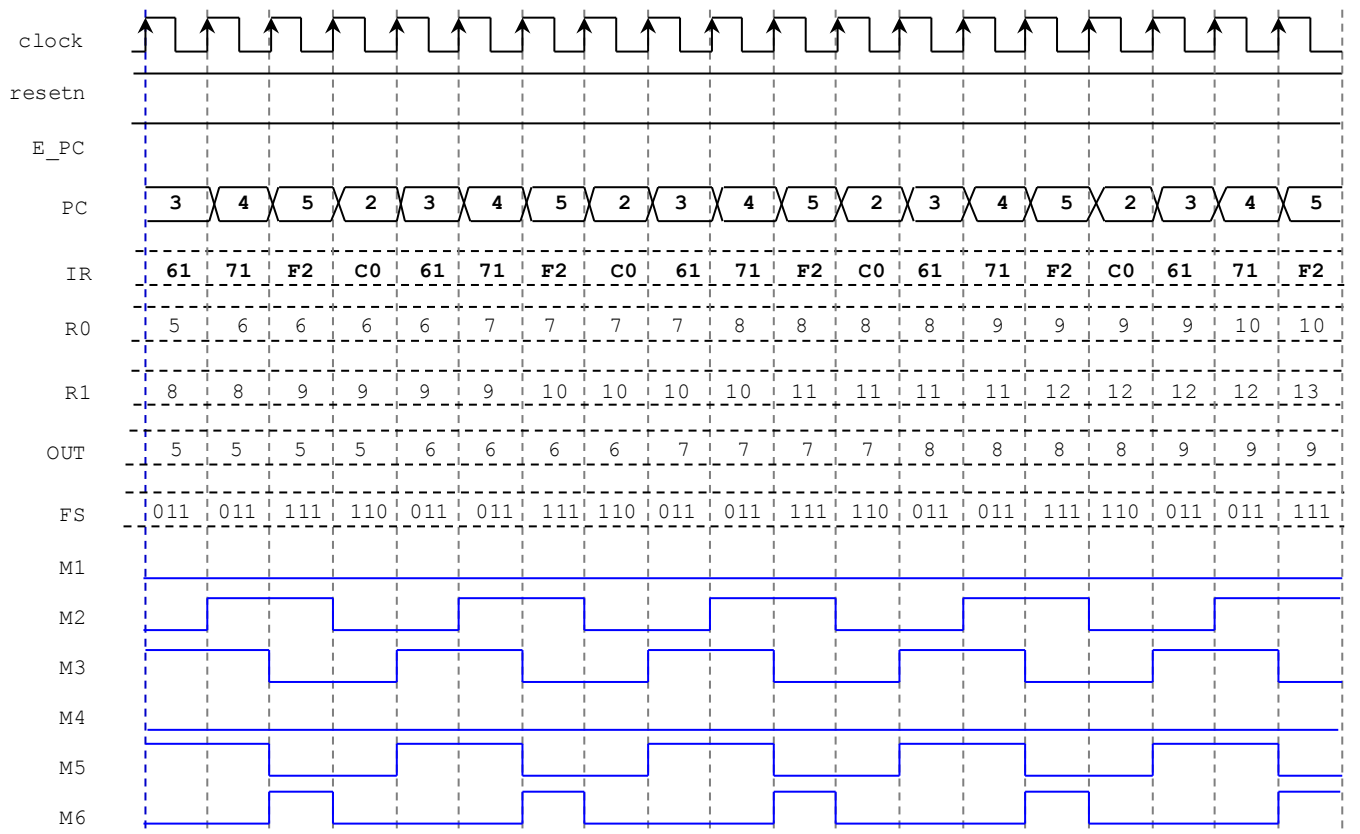


- ✓ Write an assembly program for a counter from 1 to 12: 1, 2, 3, …, 12, 1, 2, 3, … The count must be shown on the output register (**OUT**). Use labels to specify any address that an instruction may jump to. You can only have up to 16 instructions.
- ✓ Provide the contents of the Instruction Memory. If some instruction bits are unused, you can assign 0's.

Address	Instruction Memory	Assembly Instruction
0000	00100001	start: loadi R0,1
0001	00110100	loadi R1,4
0010	11000000	loop: out R0
0011	01100001	addi R0,1
0100	01110001	addi R1,1
0101	11110010	jnz R1,loop
0110	00110001	loadi R1,1
0111	11110000	jnz R1, start
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. (25 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`.





PROBLEM 2 (25 PTS)

- "**Simple Computer**": 8 registers, 64-word instruction memory, 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).

Address	Instruction	OPCODE	DR	SA	SB	OP	AD	Instruction Memory contents
0	LDI R0,5	1001100	000	---	NA	101	NA	1001100000---101
1	LDI R3,4	1001100	011	---	NA	100	NA	1001100011---100
2	LDI R2,7	1001100	010	---	NA	110	NA	1001100010---111
3	SUB R7,R3,R0	0000101	111	011	000	NA	NA	0000101111011000
4	ADI R5,R7,7	1000010	101	111	NA	111	NA	1000010101111111
5	ST R3,R5	0100000	---	011	101	NA	NA	0100000---011101
6	XOR R0,R5,R2	0001010	000	101	010	NA	NA	0001010000101010
7	LD R6, R3	0010000	110	011	---	NA	NA	0010000110011---
8	BRN R7,-5	1100001	NA	111	NA	NA	111011	1100001111111011
9	SHL R7,R2	0001110	111	---	010	NA	NA	0001110111---010
10	JMP R2	1110000	NA	010	NA	NA	------	1110000---010---

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

PC	R0	R2	R3	R5	R6	R7
000000	0005	0000	0000	0000	0000	0000
000001	0005	0000	0004	0000	0000	0000
000010	0005	0007	0004	0000	0000	0000
000011	0005	0007	0004	0000	0000	FFFF
000100	0005	0007	0004	0006	0000	FFFF
000101	0005	0007	0004	0006	0000	FFFF
000110	0001	0007	0004	0006	0000	FFFF
000111	0001	0007	0004	0006	0006	FFFF
001000	0001	0007	0004	0006	0006	FFFF
000011	0001	0007	0004	0006	0006	0003
000100	0001	0007	0004	000A	0006	0003
000101	0001	0007	0004	000A	0006	0003
000110	000D	0007	0004	000A	0006	0003
000111	000D	0007	0004	000A	000A	0003
001000	000D	0007	0004	000A	000A	0003
001001	000D	0007	0004	000A	000A	000E
001010	000D	0007	0004	000A	000A	000E

PROBLEM 3 (20 PTS)

- "**Simple Computer**": 8 registers, 64-word instruction memory, 16 bits per instruction (see Notes – Unit 6)
  - ✓ Write an assembly program for a counter from 4 to 13: 4, 5, 6, 7, 8, …, 13, 4, 5, … The count must appear on **R3**. Use labels to specify any address that an instruction may jump to. Note that you can only have up to 64 instructions. Note that the ALU treats data as signed numbers. Thus, data on the registers and Data Memory (16 bits) is signed.
  - ✓ Provide the contents of the Instruction Memory and its addresses.

```
* 4 to 13 ≡ 9 downto 0
```

Address	Instruction Memory	Assembly Instructions
000000	1001100000---100	start: ldi R0, 4
000001	1001100011---100	ldi R3, 4
000010	1001100001---111	ldi R1, 7
000011	1000010001001010	adi R1,R1, 2
000100	1000010011011001	loop: adi R3,R3, 1
000101	0000110001001---	dec R1,R1
000110	1100000111001011	brz R1, -5
000111	1110000---000---	jmp R0

PROBLEM 4 (15 PTS)

- Attach a printout of your Project Status Report (no more than three pages, single-spaced, 2 columns). This report should contain the current status of the project. You **MUST** use the provided template (`Final Project – Report Template.docx`).