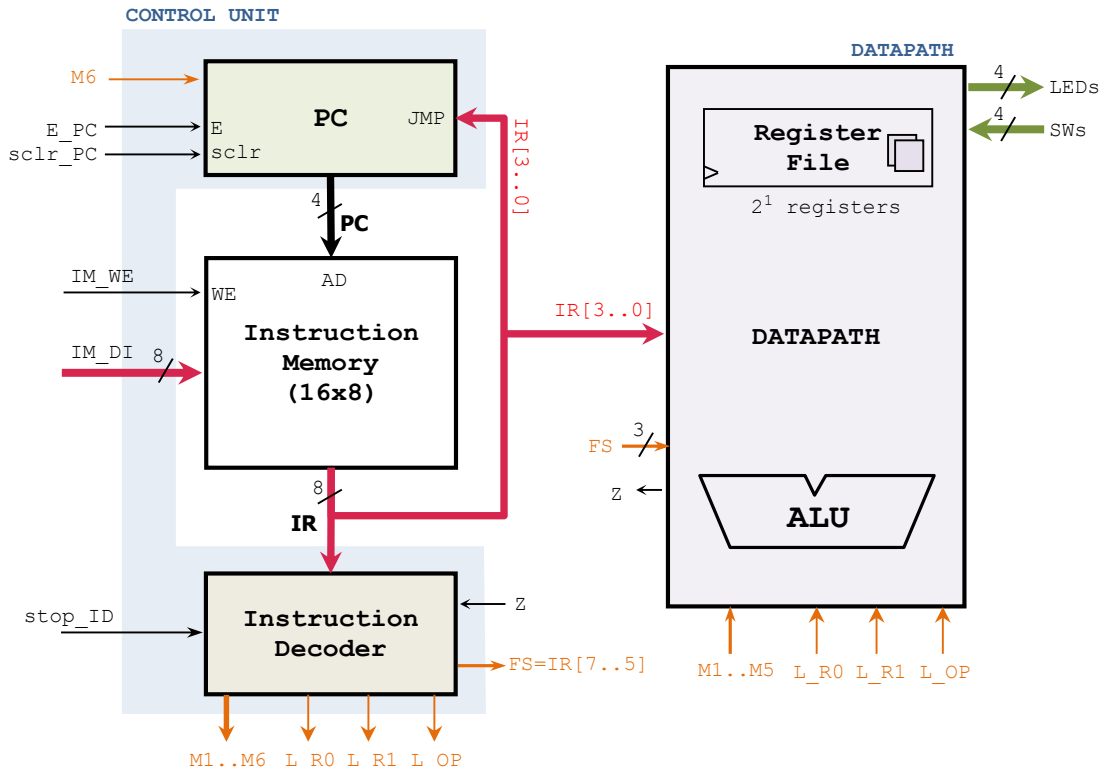# Homework 4

(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.



- ✓ 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 | | |
| 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. (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 | | | | | | | |
| 1 | LDI R3,4 | | | | | | | |
| 2 | LDI R2,7 | | | | | | | |
| 3 | SUB R7,R3,R0 | | | | | | | |
| 4 | ADI R5,R7,7 | | | | | | | |
| 5 | ST R3,R5 | | | | | | | |
| 6 | XOR R0,R5,R2 | | | | | | | |
| 7 | LD R6, R3 | | | | | | | |
| 8 | BRN R7,-5 | | | | | | | |
| 9 | SHL R7,R2 | | | | | | | |
| 10 | JMP R2 | | | | | | | |

- ✓ 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 | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

## 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.

## 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).