4-bit Computer Simulation Instruction Set

Roll XX1

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	MOV A,[ADDRESS]
5	OUT A
6	INC A
7	RCR A
8	MOV A,BYTE
9	JNZ ADDRESS
10	PUSH B
11	POP B
12	CALL ADDRESS
13	RET
14	OR A,[ADDRESS]
15	XOR A,[ADDRESS]
16	HLT

Roll XX4

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	MOV B,[ADDRESS]
5	OUT B
6	JNZ ADDRESS
7	RCR A
8	MOV B,BYTE
9	JMP ADDRESS
10	PUSH A
11	POP A
12	CALL ADDRESS
13	RET
14	XOR A,[ADDRESS]
15	TEST B,BYTE
16	HLT

Roll XX2

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	MOV B,BYTE
5	RCR B
6	JMP ADDRESS
7	JNZ ADDRESS
8	PUSHF
9	OR A,BYTE
10	PUSH B
11	POP B
12	OUT A
13	CALL ADDRESS
14	RET
15	AND A,[ADDRESS]
16	HLT

Roll XX5

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	IN A
5	RCR B
6	DEC B
7	JZ ADDRESS
8	JMP ADDRESS
9	OR B,BYTE
10	PUSH B
11	POP B
12	OUT A
13	CALL ADDRESS
14	RET
15	AND A,[ADDRESS]
16	HLT

Roll XX3

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	IN A
5	OUT A
6	INC A
7	MOV A,[ADDRESS]
8	MOV A,BYTE
9	JZ ADDRESS
10	PUSH B
11	POP B
12	RCL B
13	CALL ADDRESS
14	RET
15	AND A,[ADDRESS]
16	HLT

Roll XX6

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	RCL A
5	OUT A
6	INC A
7	MOV B,[ADDRESS]
8	MOV B,BYTE
9	JMP ADDRESS
10	PUSH B
11	POP B
12	NOT A
13	CALL ADDRESS
14	RET
15	TEST A,B
16	HLT

Roll XX7

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	MOV A,[ADDRESS]
5	MOV [ADDRESS],B
6	OUT A
7	TEST B,A
8	OR B,[ADDRESS]
9	JNZ ADDRESS
10	JMP ADDRESS
11	PUSHF
12	PUSH A
13	POP A
14	CALL ADDRESS
15	RET
16	HLT

Roll XX0

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	MOV A,[ADDRESS]
5	RCR B
6	IN A
7	OUT A
8	AND A,B
9	TEST B,BYTE
10	OR B,BYTE
11	XOR A,[ADDRESS]
12	PUSH B
13	POP B
14	CALL ADDRESS
15	RET
16	HLT

Roll XX9

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	MOV A,[ADDRESS]
5	MOV [ADDRESS],B
6	JNZ ADDRESS
7	XOR A,[ADDRESS]
8	PUSHF
9	IN B
10	OUT A
11	JMP ADDRESS
12	PUSH A
13	POP A
14	CALL ADDRESS
15	RET
16	HLT

Roll XX8

1	ADD A,B
2	SUB A,B
3	XCHG B,A
4	RCL B
5	SHR A
6	MOV [ADDRESS],A
7	XOR A,[ADDRESS]
8	AND A,B
9	OR B,[ADDRESS]
10	OUT A
11	JZ ADDRESS
12	PUSH B
13	POP B
14	CALL ADDRESS
15	RET
16	HLT

Note:
Code RAM size: 1024x8
Data RAM size: 1024x8
Input port consists of 8 switches.
Output port consists of 8 LEDs.
Code segment and Data Segment should be separate.
Write your assembly code, compile, and load the hex file/abs file into a ROM. When simulation starts, the data from the ROM will be written to the Code and Data RAM accordingly at the very beginning (you may need to write your own compiler to do this). Next, fetch cycle should start.
Flags: ZF,SF,CF, OF must be shown. Other flags optional
Implementation of Pipelining would be favorable for bonus.
Each student should submit their own program.
Presentation contest: Explanation of architecture and run a given program

Submission deadline: 6th Week (sharp)