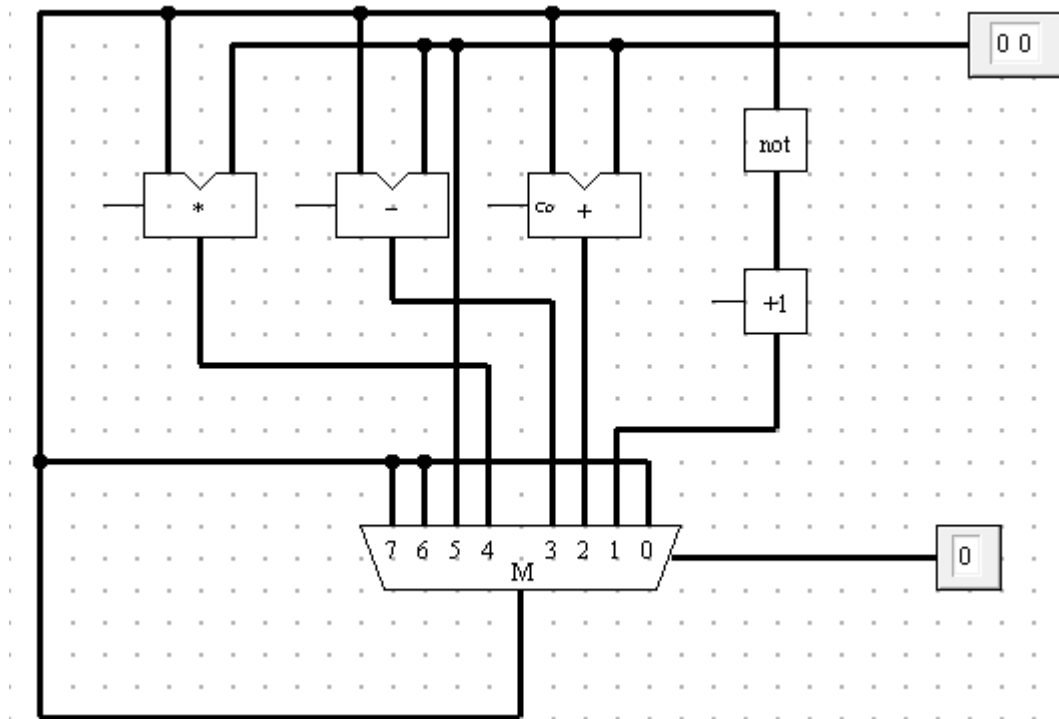
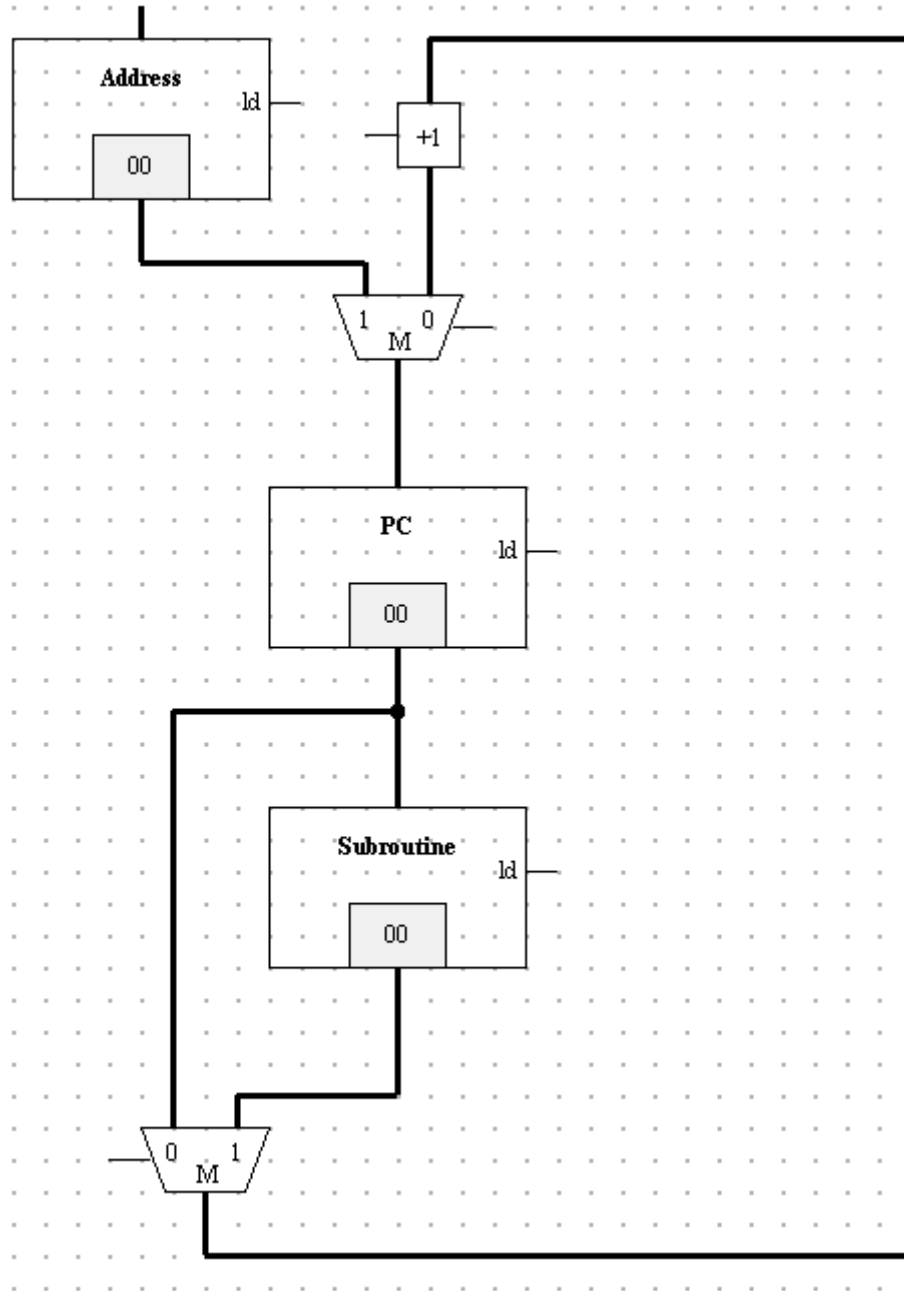


**Tutorial 2 – ALU + CU**

2.

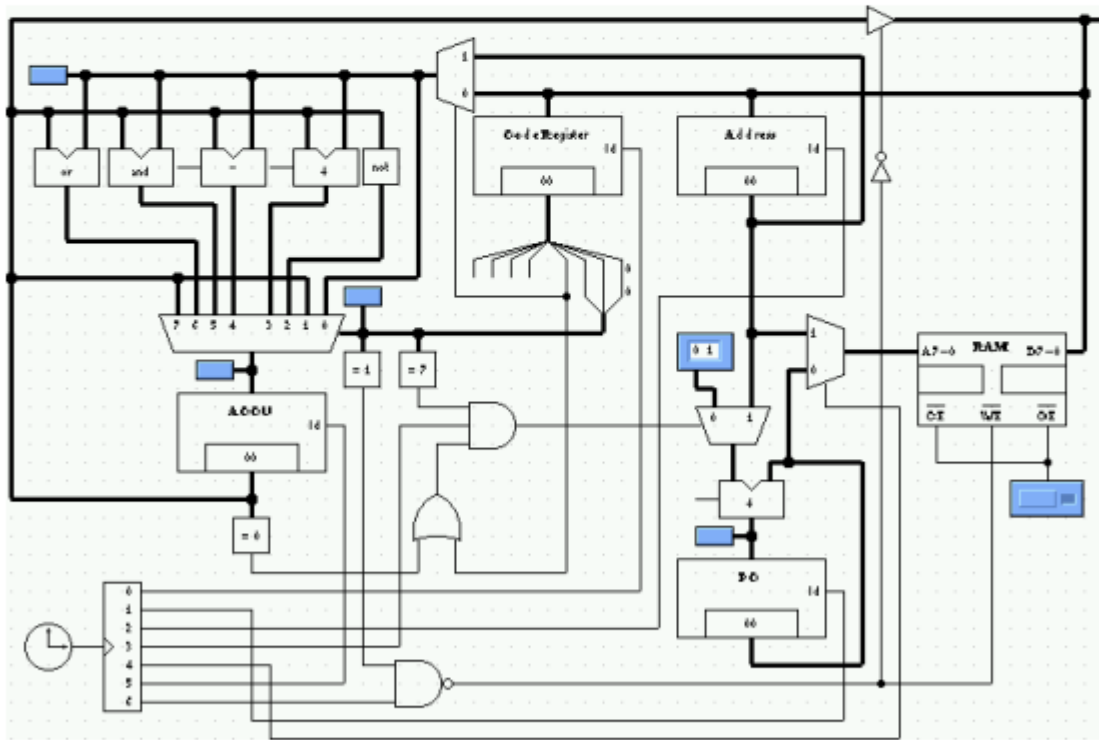
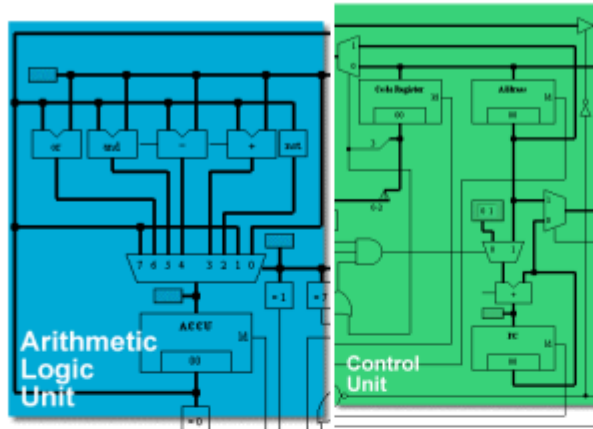


3.



4.

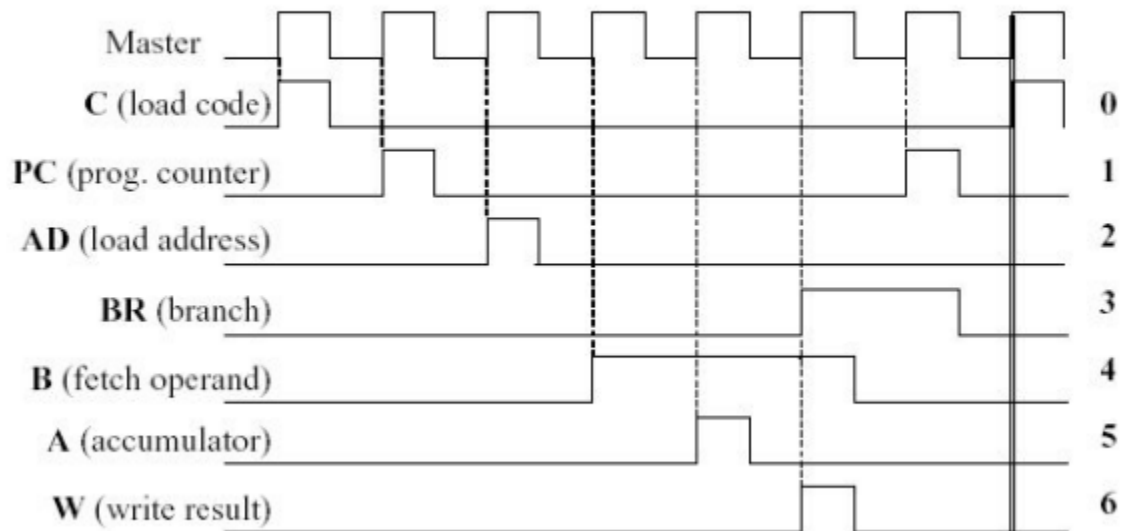
# CPU4



## OpCodes

Opcode	Description	Abbreviation
0	acc ← memory pc ← pc + 2	<b>LOAD</b> memory
1	memory ← acc pc ← pc + 2	<b>STORE</b> memory
2	acc ← NOT acc pc ← pc + 2	<b>NOT</b>
3	acc ← acc + memory pc ← pc + 2	<b>ADD</b> memory
4	acc ← acc - memory pc ← pc + 2	<b>SUB</b> memory
5	acc ← acc AND memory pc ← pc + 2	<b>AND</b> memory
6	acc ← acc OR memory pc ← pc + 2	<b>OR</b> memory
7	(* acc unchanged *) if acc = 0 then pc ← pc + address else pc ← pc + 2	<b>BEQ</b> address
8	acc ← constant pc ← pc + 2	<b>LOAD</b> constant
9		
10		
11	acc ← acc + constant pc ← pc + 2	<b>ADD</b> constant
12	acc ← acc - constant pc ← pc + 2	<b>SUB</b> constant
13	acc ← acc AND constant pc ← pc + 2	<b>AND</b> constant
14	acc ← acc OR constant pc ← pc + 2	<b>OR</b> constant
15	(* acc unchanged *) pc ← pc + address	<b>BRA</b> address

# Timing Diagram



## Multiplication Program

Program to multiply two numbers

Address	Opcode Data	Mnemonic	Comment
00	08 00	LOAD #0	Clear result memory cell (\$FF)
02	01 FF	STORE FF	
04	00 FD	LOAD FD	Load first operand (\$FD) ..
06	07 FF	BEQ -1	.. done if 0 (BEQ -1 equiv. to dynamic HALT)
08	0C 01	SUB #1	Subtract 1 from first operand
0A	01 FD	STORE FD	
0C	00 FE	LOAD FE	Load second operand (\$FE) and add to result
0E	03 FF	ADD FF	
10	01 FF	STORE FF	
12	0F F1	BRA -15	Branch to loop (address 4)