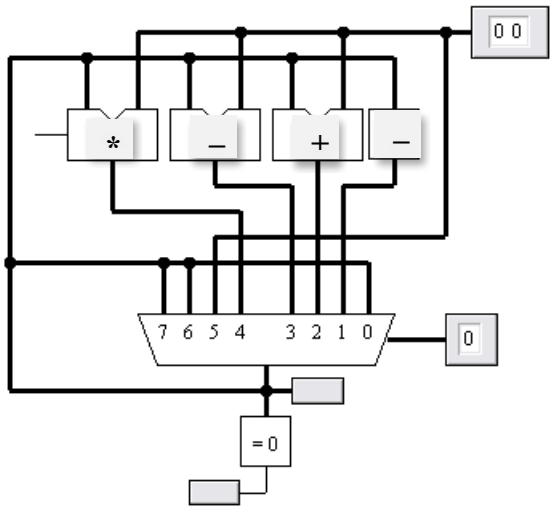
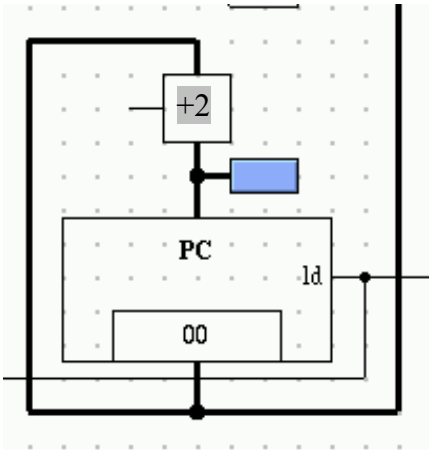
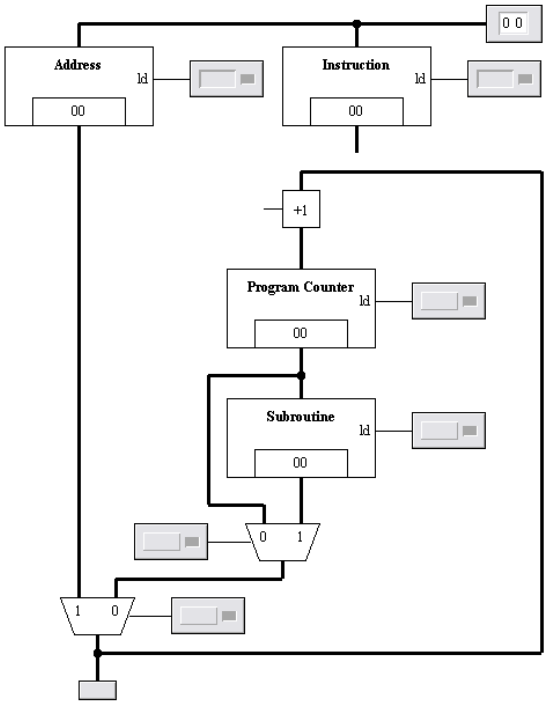
SOLUTIONS

ALU

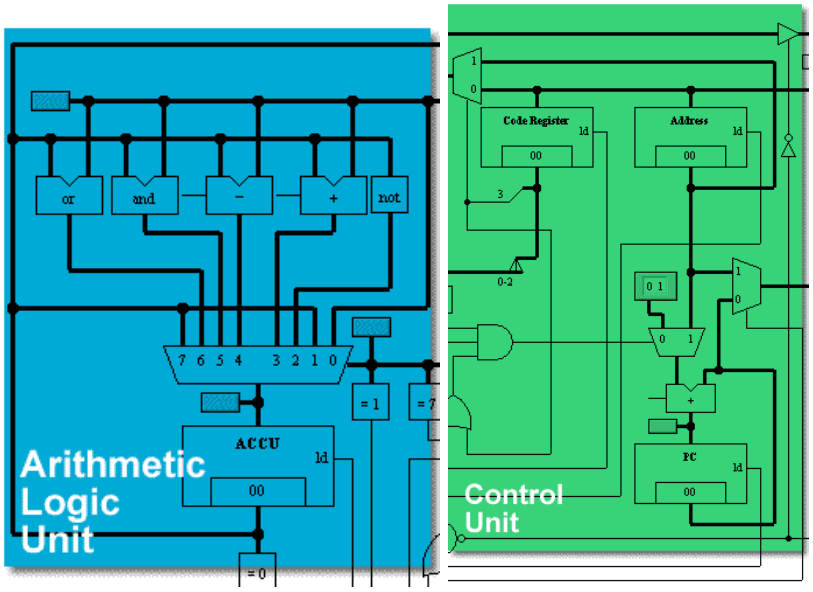


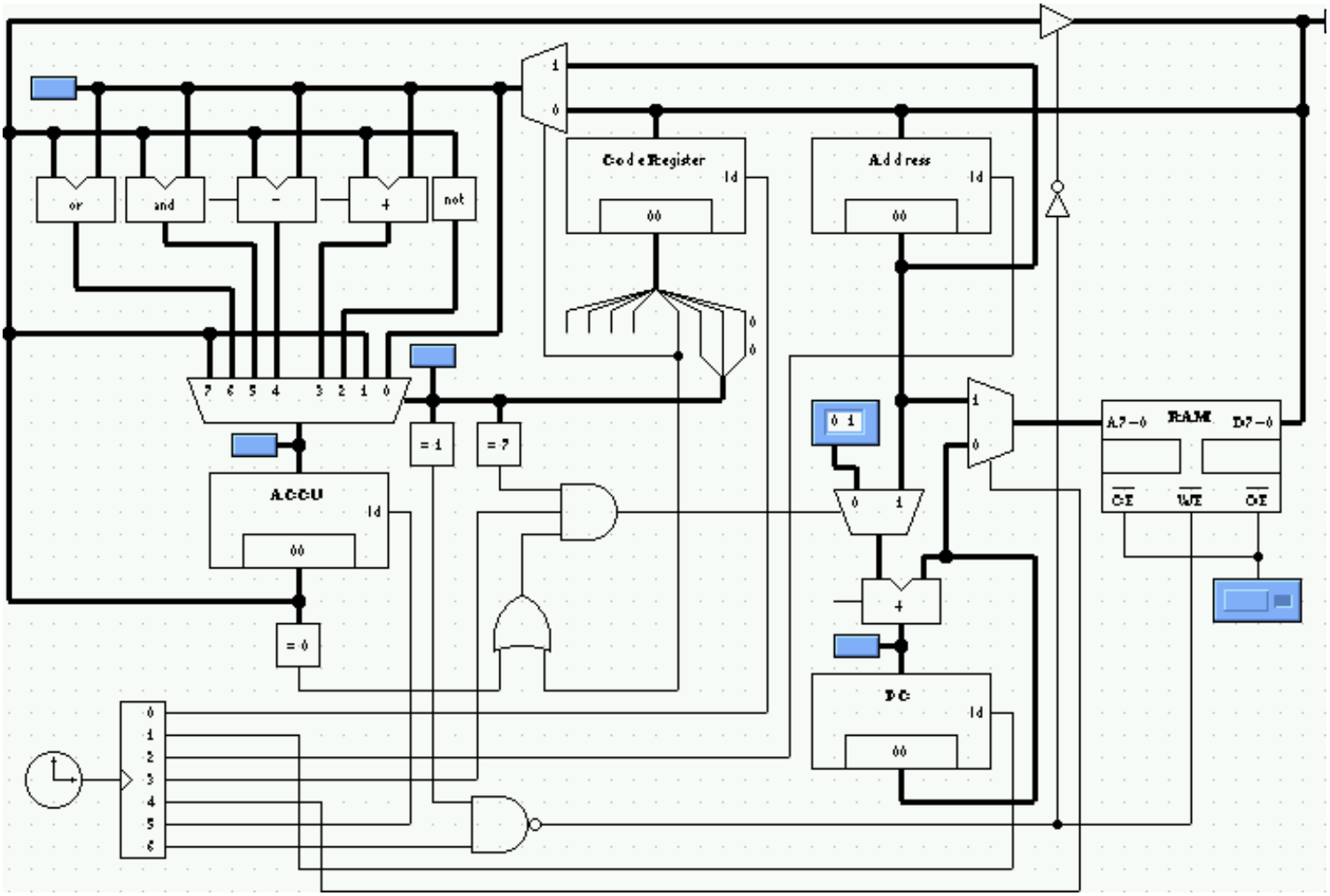
CU with +2 and CU with Jump and Subroutine call/return





CPU4

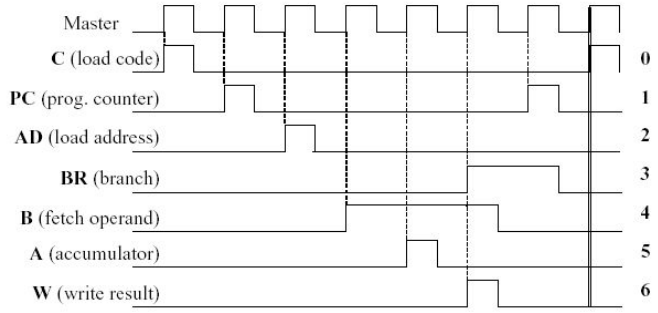

Arithmetic
Logic
Unit
Control
Unit
or
and
-
+
not
ACCU
Code Register
Address
PC
ld
00
= 0
= 1
= 7


or
and
-
+
not
Code Register
Address
ACCU
PC
RAM
ld
00
= 0
= 1
= 7

Opcodes

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