Micro-programmed Control Unit

(Computer 2 described in Computer System Architecture-Morris Mano)

INDEX

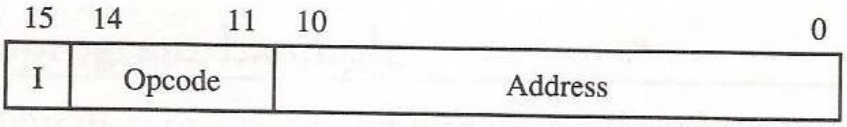
TOPIC	PAGE NO.
MICROINSTRUCTION	1
A L U	8
DATA REGISTER	10
PROGRAM COUNTER	11
ADDRESS REGISTER	12
CAR	12
SBR	13
Figure of computer 2	15

Tool used- logisim

1-MICROINSTRUCTION FORMAT

The 20 bits of the microinstruction are divided into four functional part the three fields F1,F2 and F3 specify micro-operation for computer the CD field select the status bit conditions.the BR field specifies the type of branch to be used.the AD field cotains the branch address.the address field is 7 bit wide,since they control memory has 128=2^7words.

Figure 7-5 Computer instructions.



(a) Instruction format

Symbol	Opcode	Description
ADD	0000	$AC \leftarrow AC + M\,[EA]$
BRANCH	0001	If $(AC < 0)$ then $(PC \leftarrow EA)$
STORE	0010	$M\,[EA] \leftarrow AC$
EXCHANGE	0011	$AC \leftarrow M[EA]$, $M[EA] \leftarrow AC$

EA is the effective address

(b) Four computer instructions

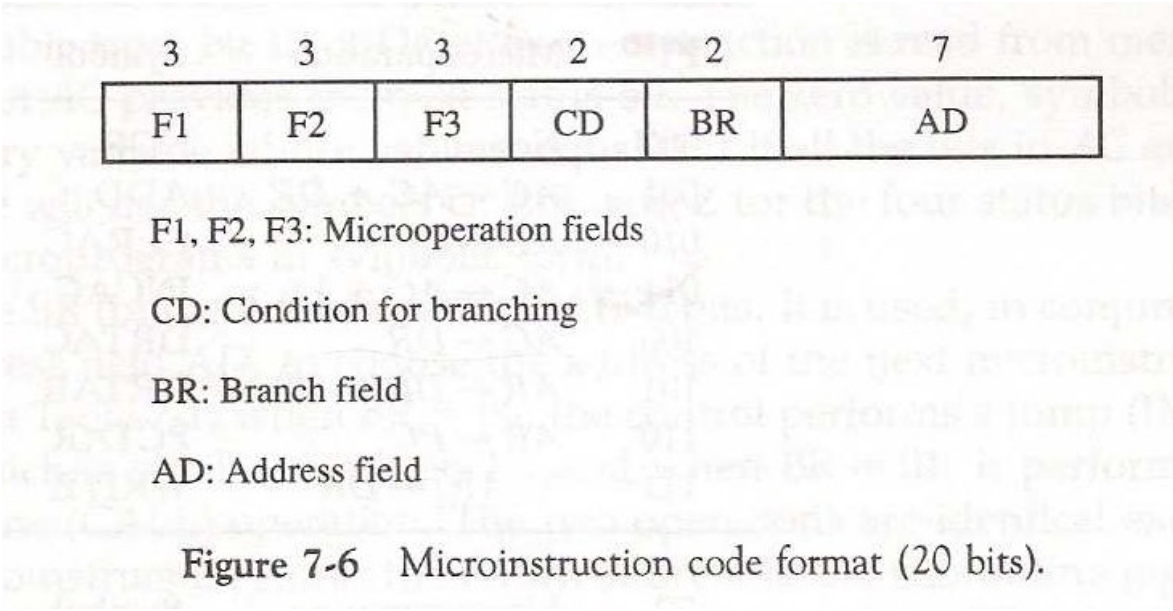


Figure 7-6 Microinstruction code format (20 bits).

16 bit input instruction given to the ram ,to perform any operation we write instruction code in ram. there is no change possible in rom, initially we fixed operation and write their sub routine in rom Each operation have atmost 4 memory space in rom , therefore we can perform 32 operation in computer 2 because the size of rom is 128 words.sub routine of fetch and indirect address is said to memory location 64-68,if we want to add any operation in our system simply write all there sub routine in rom.

Data size in rom is 20 bit. size of each of F1,F2 and F3 is 3 bit,so we can perform 7+7+7=21 operation atmost. because first bit of each of F1 ,F2 and F3 is nop.size of CD and BR is 2 bit,because there is four conditional bit and four branching statements.

If output of CD is true then branching statement executes otherwise control moves to next statement of sub routine .

We have four condition symbol (bit)

1-U is unconditional brancndih always 1

2-I is indirect address bit dr(15)

3-s is sign bit of ac ac(15)

4-z is zero value in ac(ac=0)

We have four branch conditional (bit)

1-JMP (car<-AD if condition=1

Car<- car+1,if condition=0)

2-CALL(car<-ad,sbr<-car+1 if condition=1

Car<-car+1 if condition=0)

3-RET-(car<-sbr(return from sub routine))

4-MAP-(car(2-5)<-dr(11-14),car(0,1,6)<-0)

TABLE 7-1 Symbols and Binary Code for Microinstruction Fields

F1	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC + DR$	ADD
010	$AC \leftarrow 0$	CLRAC
011	$AC \leftarrow AC + 1$	INCAC
100	$AC \leftarrow DR$	DRTAC
101	$AR \leftarrow DR(0\text{–}10)$	DRTAR
110	$AR \leftarrow PC$	PCTAR
111	$M[AR] \leftarrow DR$	WRITE

F2	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC - DR$	SUB
010	$AC \leftarrow AC \vee DR$	OR
011	$AC \leftarrow AC \wedge DR$	AND
100	$DR \leftarrow M[AR]$	READ
101	$DR \leftarrow AC$	ACTDR
110	$DR \leftarrow DR + 1$	INCDR
111	$DR(0\text{-}10) \leftarrow PC$	PCTDR

F3	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC \oplus DR$	XOR
010	$AC \leftarrow \overline{AC}$	COM
011	$AC \leftarrow$ shl AC	SHL
100	$AC \leftarrow$ shr AC	SHR
101	$PC \leftarrow PC + 1$	INCPC
110	$PC \leftarrow AR$	ARTPC
111	Reserved	

CD	Condition	Symbol	Comments
00	Always = 1	U	Unconditional branch
01	$DR(15)$	I	Indirect address bit
10	$AC(15)$	S	Sign bit of AC
11	$AC = 0$	Z	Zero value in AC

BR	Symbol	Function
00	JMP	$CAR \leftarrow AD$ if condition = 1
		$CAR \leftarrow CAR + 1$ if condition = 0
01	CALL	$CAR \leftarrow AD$, $SBR \leftarrow CAR + 1$ if condition = 1
		$CAR \leftarrow CAR + 1$ if condition = 0
10	RET	$CAR \leftarrow SBR$ (Return from subroutine)
11	MAP	$CAR(2\text{–}5) \leftarrow DR(11\text{–}14)$, $CAR(0,1,6) \leftarrow 0$

TABLE 7-2 Symbolic Microprogram (Partial)

Label	Microoperations	CD	BR	AD
	ORG 0			
ADD:	NOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	ADD	U	JMP	FETCH
	ORG 4			
BRANCH:	NOP	S	JMP	OVER
	NOP	U	JMP	FETCH
OVER:	NOP	I	CALL	INDRCT
	ARTPC	U	JMP	FETCH
	ORG 8			
STORE:	NOP	I	CALL	INDRCT
	ACTDR	U	JMP	NEXT
	WRITE	U	JMP	FETCH
	ORG 12			
EXCHANGE:	NOP	I	CALL	INDRCT
	READ	U	JMP	NEXT
	ACTDR, DRTAC	U	JMP	NEXT
	WRITE	U	JMP	FETCH
	ORG 64			
FETCH:	PCTAR	U	JMP	NEXT
	READ, INCPC	U	JMP	NEXT
	DRTAR	U	MAP	
INDRCT:	READ	U	JMP	NEXT
	DRTAR	U	RET	

LDA-

002C3 NOP I CALL INDRCT

10012 READ U JMP NEXT

80040 DRTAC U JMP FETCH

SUBTRACT-

002C3 NOP I CALL INDRCT

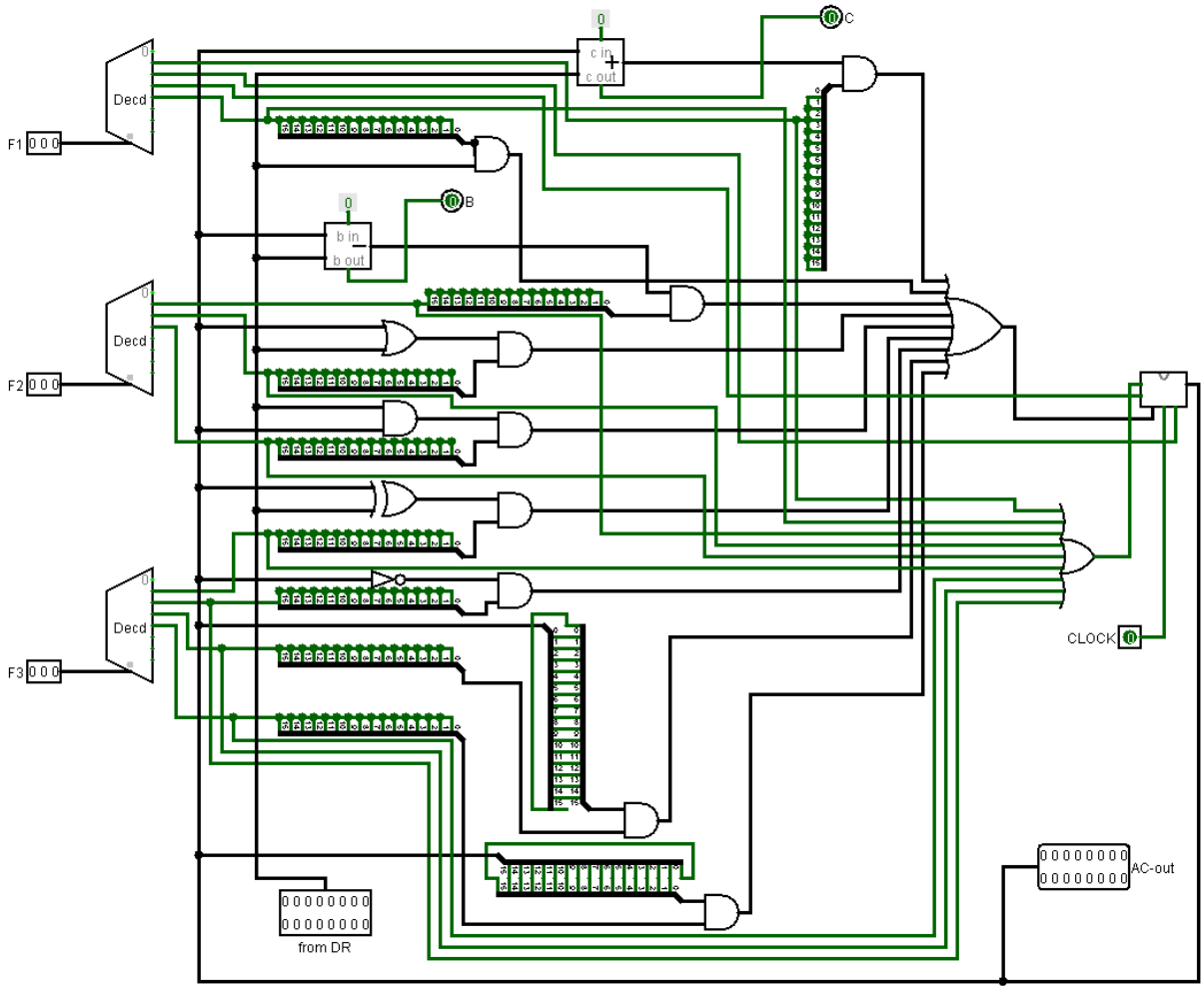
10016 READ U JMP NEXT

04040 SUB U JMP FETCH

TABLE 7-3 Binary Microprogram for Control Memory (Partial)

Micro Routine	Address		Binary Microinstruction					
	Decimal	Binary	F1	F2	F3	CD	BR	AD
ADD	0	0000000	000	000	000	01	01	1000011
	1	0000001	000	100	000	00	00	0000010
	2	0000010	001	000	000	00	00	1000000
	3	0000011	000	000	000	00	00	1000000
BRANCH	4	0000100	000	000	000	10	00	0000110
	5	0000101	000	000	000	00	00	1000000
	6	0000110	000	000	000	01	01	1000011
	7	0000111	000	000	110	00	00	1000000
STORE	8	0001000	000	000	000	01	01	1000011
	9	0001001	000	101	000	00	00	0001010
	10	0001010	111	000	000	00	00	1000000
	11	0001011	000	000	000	00	00	1000000
EXCHANGE	12	0001100	000	000	000	01	01	1000011
	13	0001101	001	000	000	00	00	0001110
	14	0001110	100	101	000	00	00	0001111
	15	0001111	111	000	000	00	00	1000000
FETCH	64	1000000	110	000	000	00	00	1000001
	65	1000001	000	100	101	00	00	1000010
	66	1000010	101	000	000	00	11	0000000
INDRCT	67	1000011	000	100	000	00	00	1000100
	68	1000100	101	000	000	00	10	0000000

ALU-



Load operations:

1-add- AC<-AC+DR

2-DRTAC-AC<-DR

3-SUB- AC<-AC-DR

4-OR-AC<-AC||DR

5-AND-AC<-AC^DR

6-XOR- AC<-AC (+) DR

7-COM-AC<-AC’

8-SHL-AC<-SHL(AC)

9-SHR-AC<-SHR(AC)

CLEAR OPERATION-

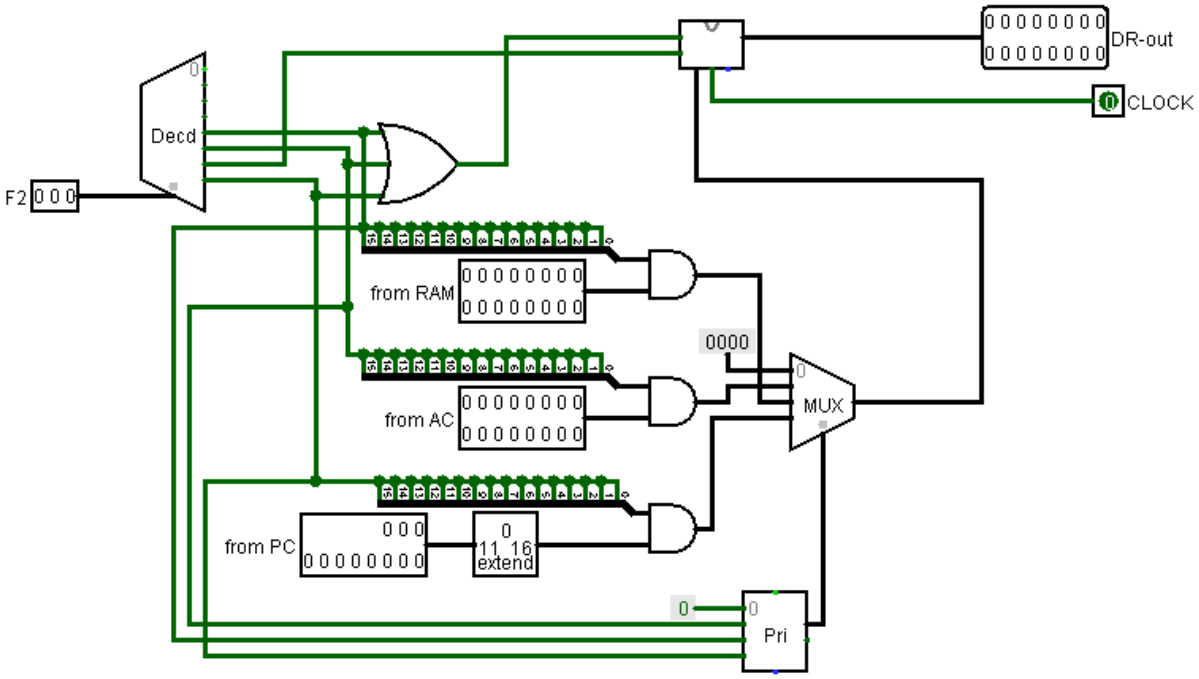
CLRAC –AC<-0

INCREMENT OPERATION-

INCAC- AC<-AC+1

ALU- the airthmetic logic shift unit is shown in above fig. ,instead of using gate to generate the control signal, operations like AND,ADD,and DR.these input now come from the output of the decoder associated by the decoders like AND,ADD etc operations.

DATA REGISTER-



LOAD OPERATION-

1-ACTDR-DR<-AC

2-PCTDR-DR(0-10)<-PC

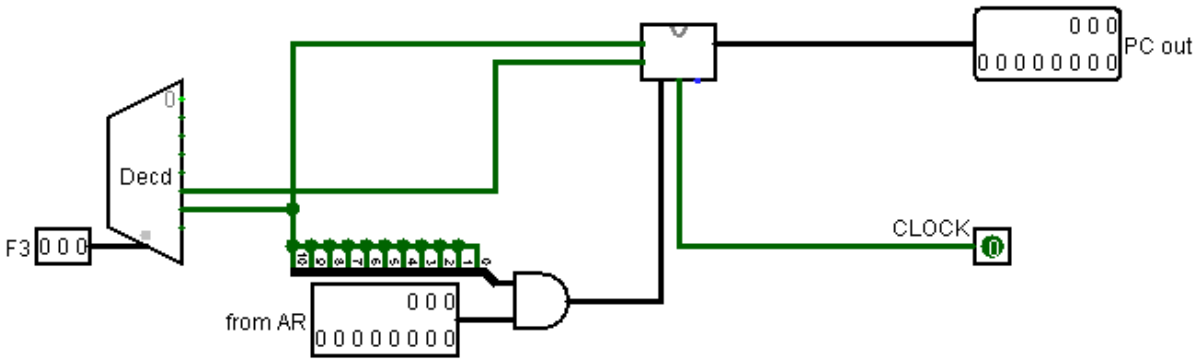
INCREMENT OPERATION

INCDR –DR<-DR+1

It is 16 bit data register which take input from multiplexer to select data of PC ,RAM and AC. Output is transferred to the AC through ALU to perform any arithmetic and logical operation

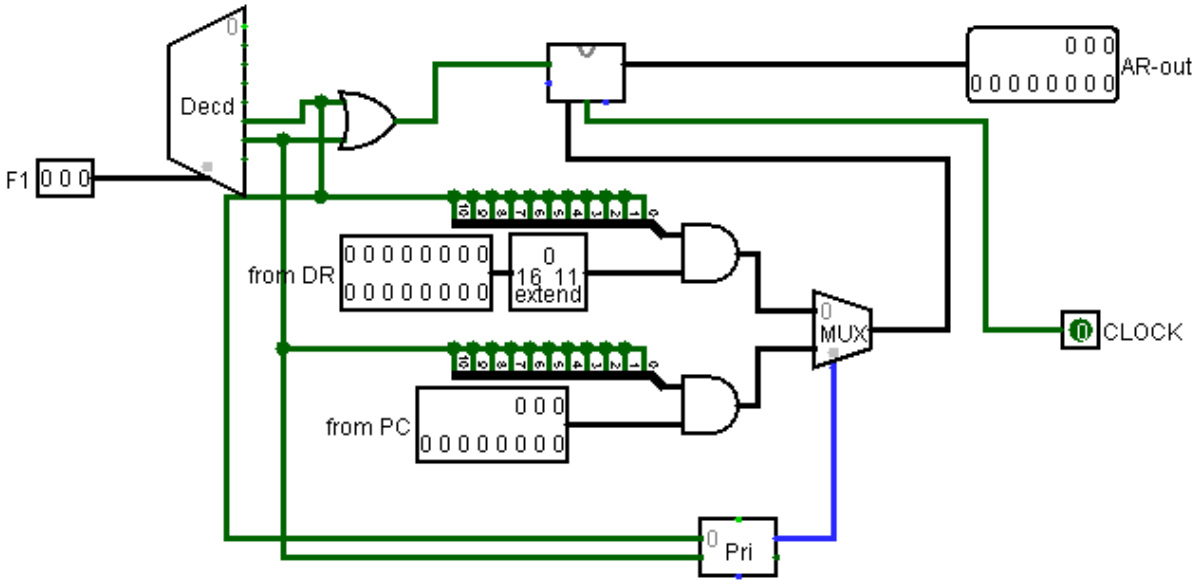
Between DR and AC.data from RAM can not be directly shifted to RAM,first it comes to DR then to AC.

PROGRAM COUNTER-



LOAD OPERATION-
ARTPC-PC<-AR
INCREMENT OPERATION
INCPC-PC<-PC+1
PC is used to store address of next instruction so size of PC is 11 bit alike AR.

ADDRESS REGISTER



LOAD OPERATION-
1-DRTAR-AR<-DR(0-10)
2-PCTAR-AR<-PC

AR is used to store address of current instruction during program execution of RAM ,size of AR is 11 bit input of AR is from either PC or DR output is always towards PC .it is always associated with RAM.

CAR- it is 7 bit control address register with specifies the address of micro instruction in control memory ROM.
It maintains the sequence of microinstruction tu be executed in RAM.input from CAR ,external map(RAM),incrementer, SBR or address part of CONTROL MEMORY and output is associated with CONTROL MEMORY.

SBR- 7 bit sub routine register which holds next address of CAR(incremented by 1).it is used for branching. Sub routine register can become source for transferring the address for the return to main sub routine

CONTROL MEMORY- it is non volatile memory,once instruction code is set in control memory then can never be change again,so it is necssary to write all instruction in ROM to perform operation .

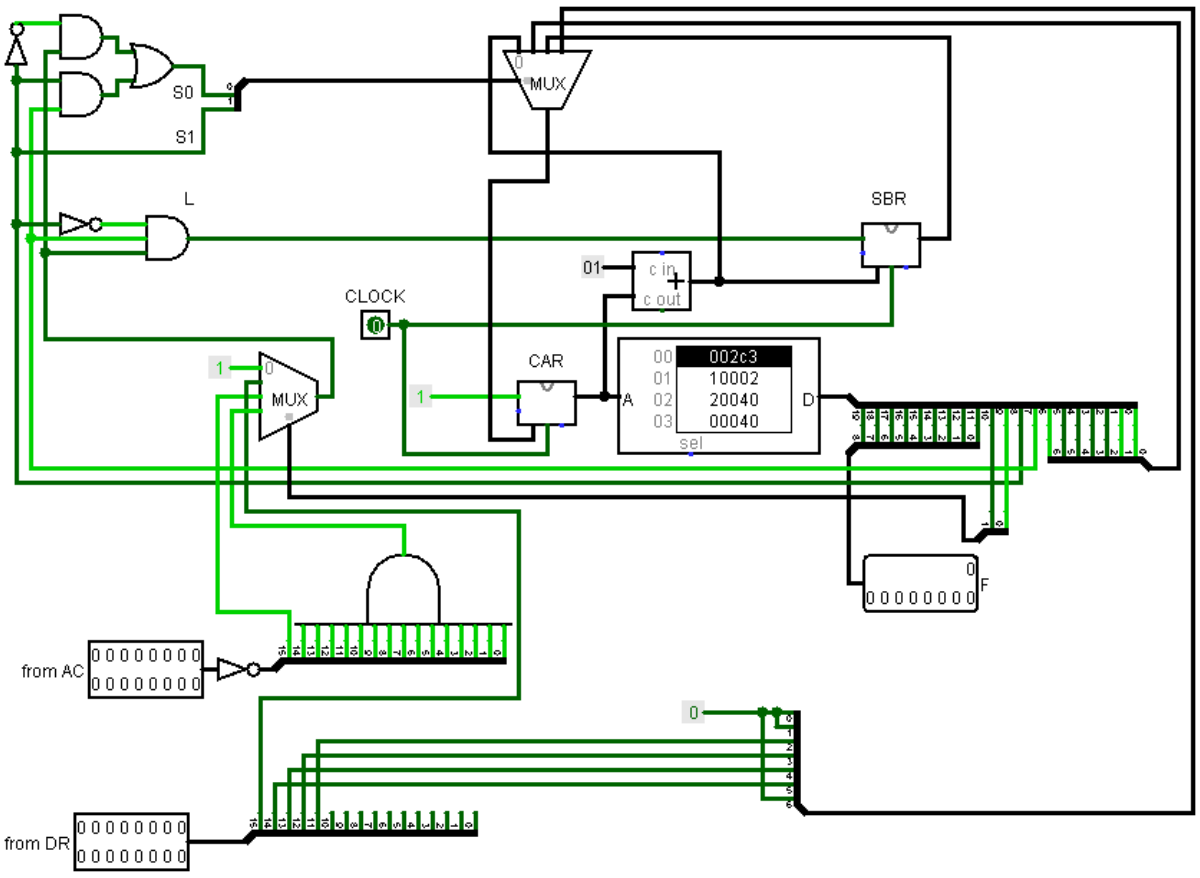


Fig.p2

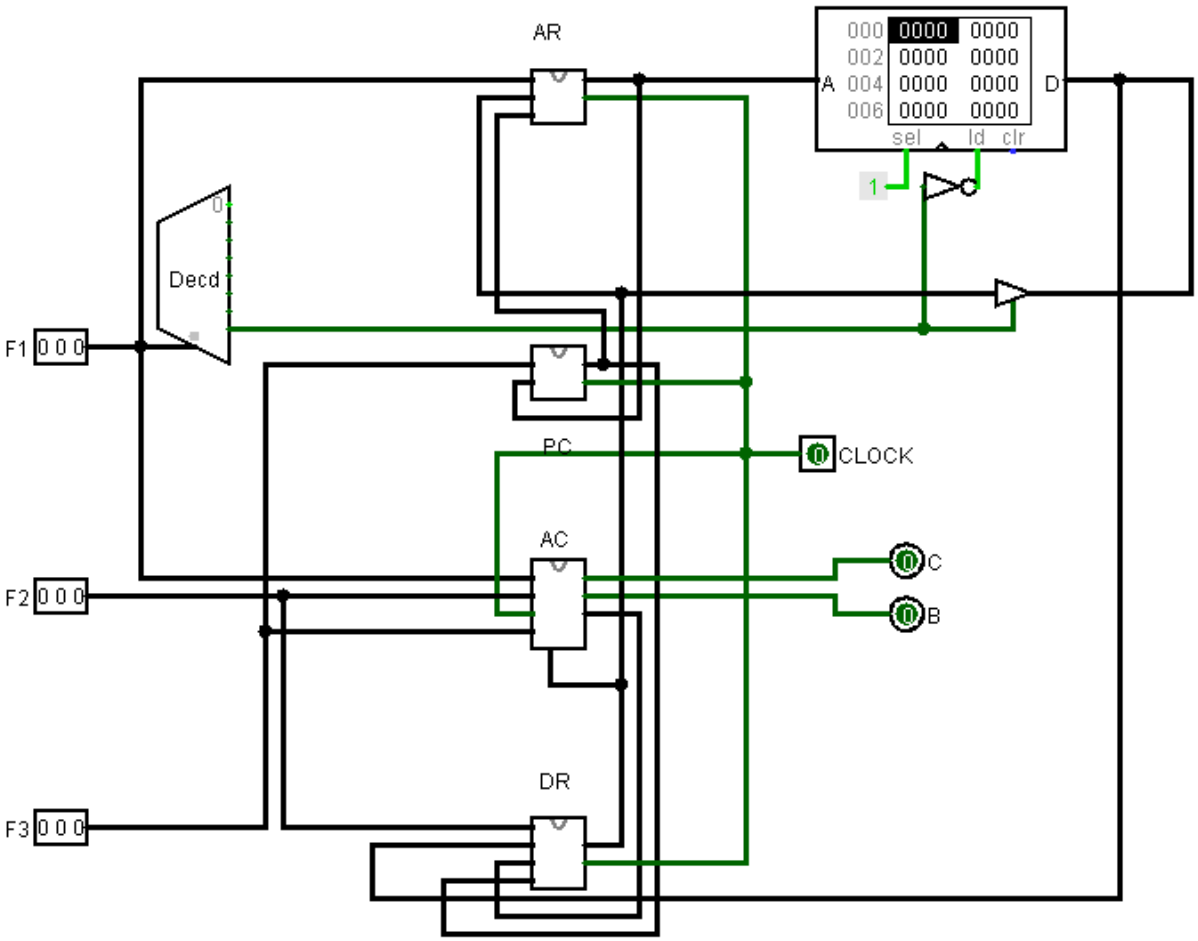


Fig. P1

Join both p1 and p2 to each other to get the computer 2.

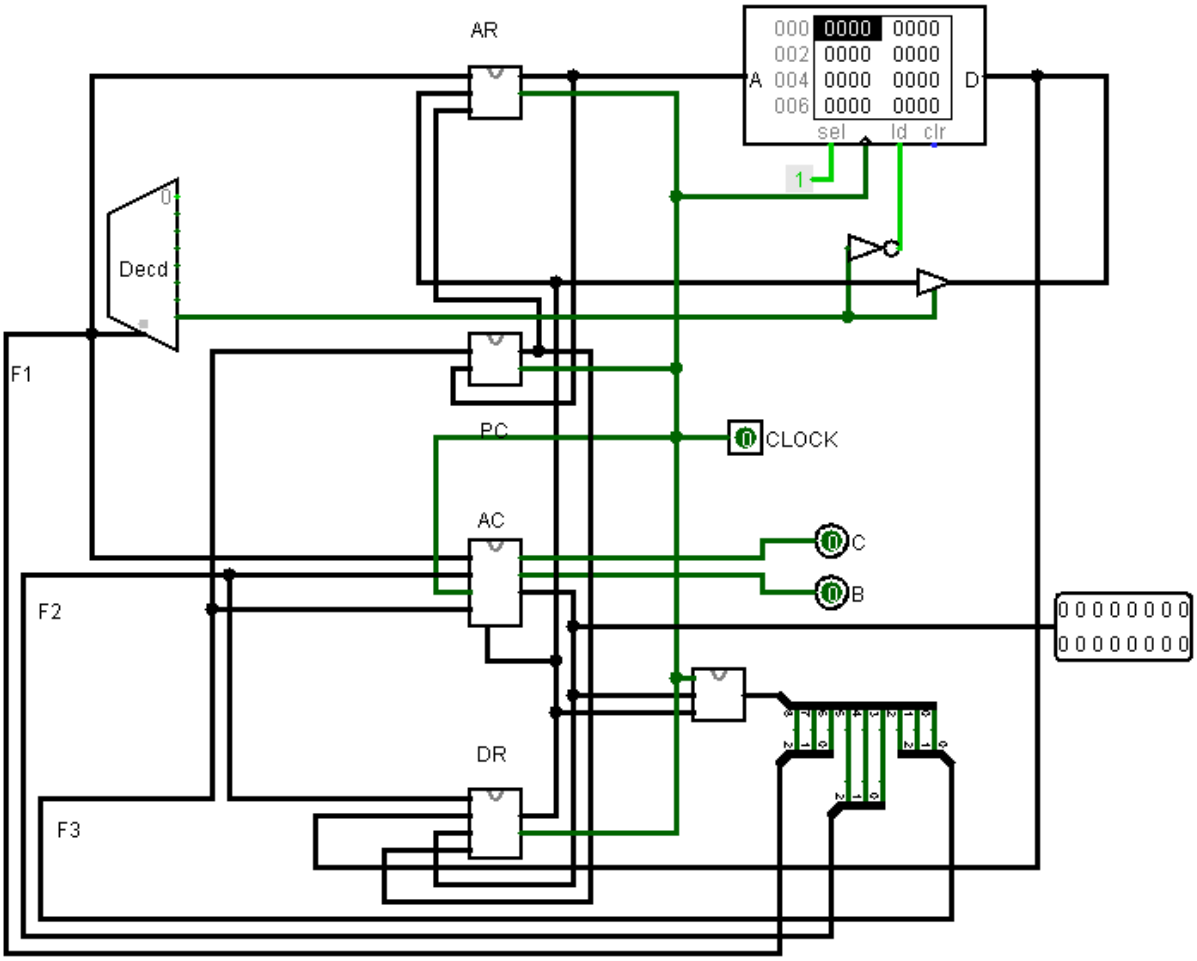


Figure for computer 2