Hardwired Control Unit System

(Computer 1 described in Computer System Architecture-Morris Mano)

INDEX

The organisation of the computer is defined by its internal registers, the timing and control structure,and the set of instructions and that it uses. The design of the computer is then carried out in detail. Although the basic computer presented in this computer 1 is very small compared to commercial computers, it has the advantage of being simple enough so we can demonstrate the design process without too many complications.

The internal organization of a digital system is defined by the sequence of microoperations it performs on data stored in its registers. The general purpose digital computer is capable of executing various microoperations and, in addition,can be instructed as to what specific sequence of operations it must perform. The user of a computer can control the process by means of a program. A program is a set of instructions that specify the operations, operands, and the sequence by which processing task may be altered by specifying a new program with different instructions or specifying the same instructions with different data.

A computer instruction is a binary code that specifies a sequence of microoperations for the computer. Instruction codes together with data are stored in memory. The computer reads each instruction from memory and places it in a control register. The control then interprets the binary code of the instruction and proceeds to execute it by issuing instruction set. The ability to store and execute instructions, the stored program concept, is the most important property of a general purpose computer.

1. BUS

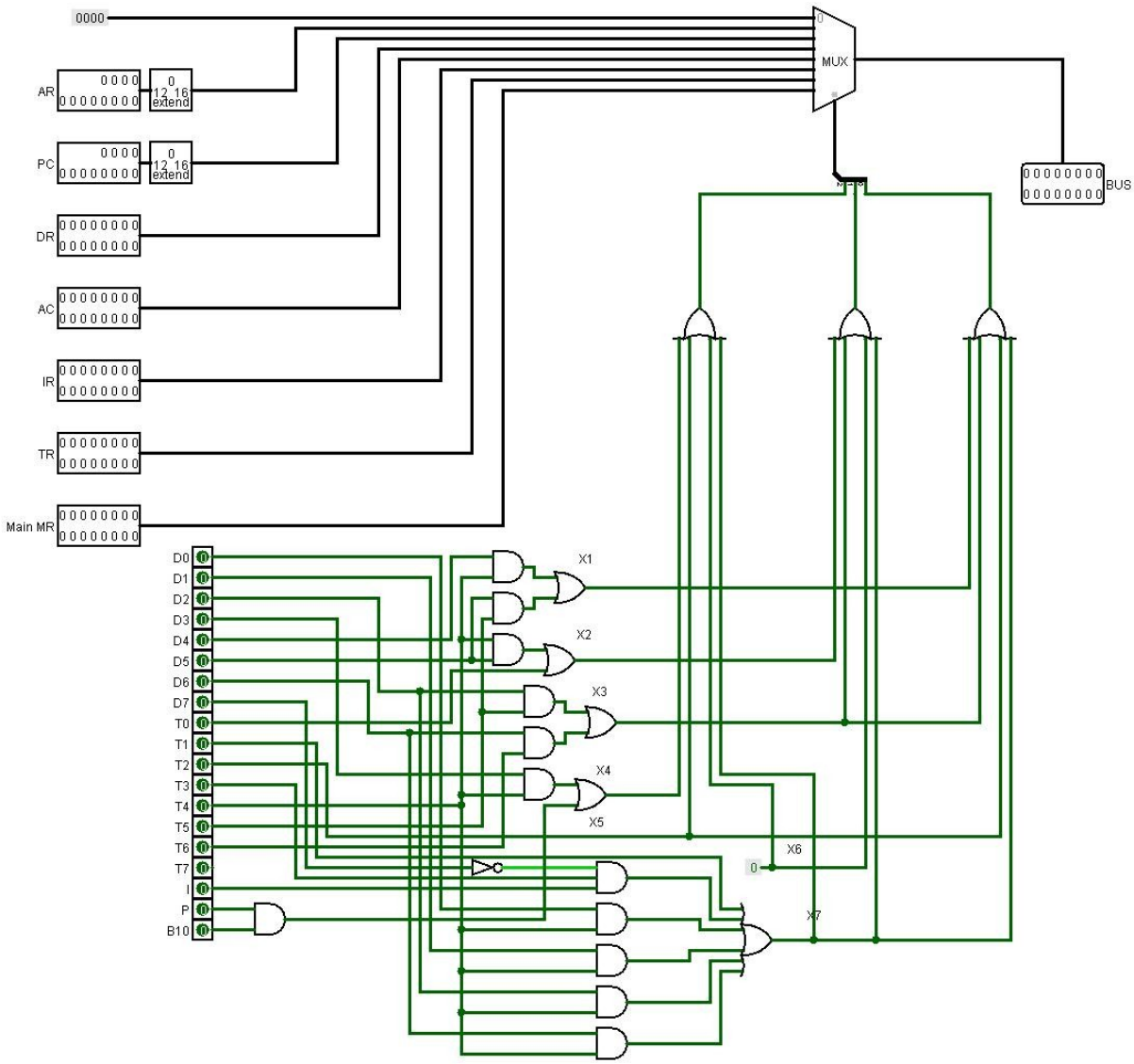


FIG-1.1

Here we used 8*1 multiplexer to select data of 6 register and main S0,S1,S2 is used to select data from register to bus(in other words data transfer into bus).

Suppose x1 (AR to BUS), x2 (PC to BUS), x3(DR to BUS),

x4(AC to BUS), x5(IR to BUS), x6 (TR to BUS),

x7 (MAIN MEMORY to BUS) without using interrupt in computer 1.

x1 = D4T4 + D5T5

x2 = T0 + D5T4

x3 = D2T5 + D6T6

x4 = D3T4

x5 = T2

x6= 0 (because there is no special use of TR without interrupt in computer 1)

x7 = T1 +D7'IT3 +(D0+D1+D2+D6)T4

As shown in fig 1.1

(X1-x7)-> (**ENCODER**)->(S2-S0)

1. Output of all registers behave as a input for bus.
2. To select a specific register S0,S1,S2 are employed.
3. Output of bus are connected to input of all registers(except AC) and main memory.
4. Input of AC is always connected to ALU.
5. To control S0, S1,S2 we have used priority encoder (As shown above).
6. For AR and PC 12 bit LSB data and for OUTR 8 bit LSB data is only used.

2. AR

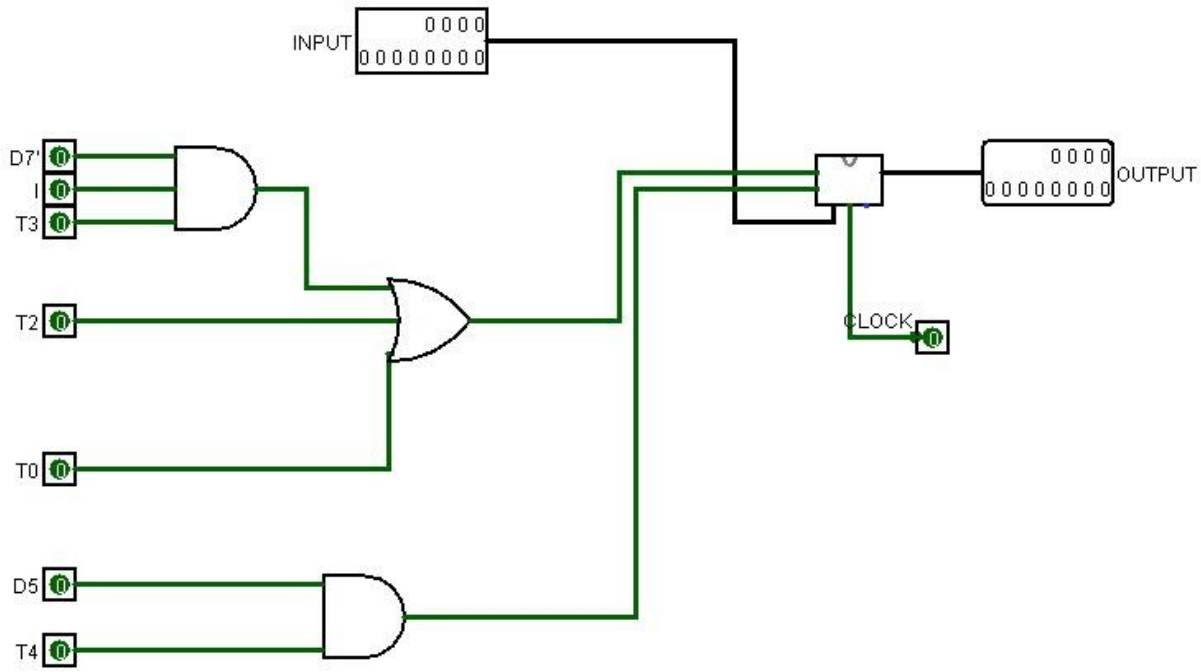


FIG-1.2

1. AR is 12 bit address register which holds address of memory location.
2. Initially data is transferred from PC to AR.
3. Function of AR for computer 1 –
   LD – (D7'IT3 +T2 + T0)
   INR – (D5T4)
   Input - from BUS
   Output – to BUS
4. AR is always associated with Input of address part of main memory.

3. PC

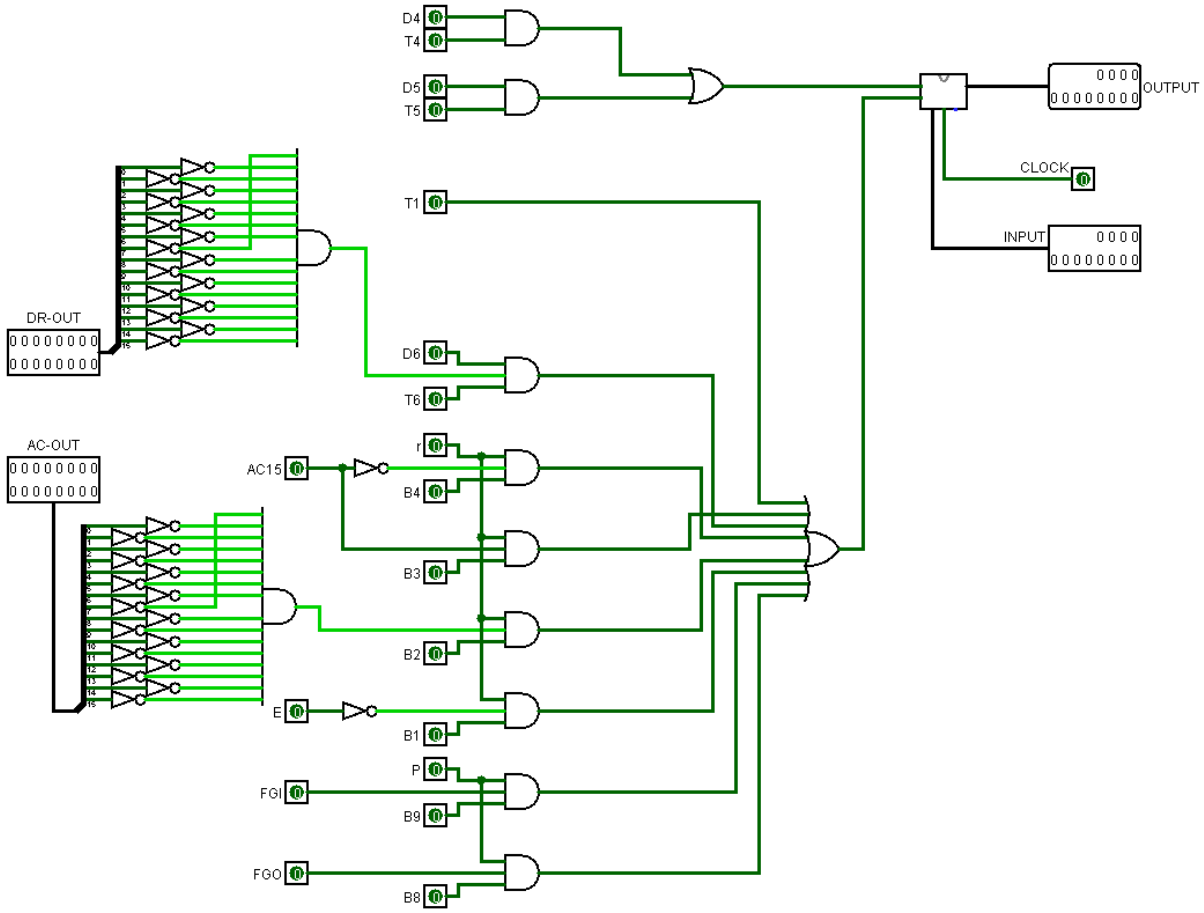


FIG – 1.3

1. It is a 12 bit program counter register, which holds address of next instruction to be read from memory after current instruction is executed.
2. Functions of PC –
   LD – (D4T4 +D5T5)
   INR – (T1 + D6DR'T6 + rAC(15)'B4 + rAC(15)B3 + rAC'B2 + rE'B1 + P(FGIB9 + FGOB8))
3. Input – from BUS
4. Output – to BUS

4. DR

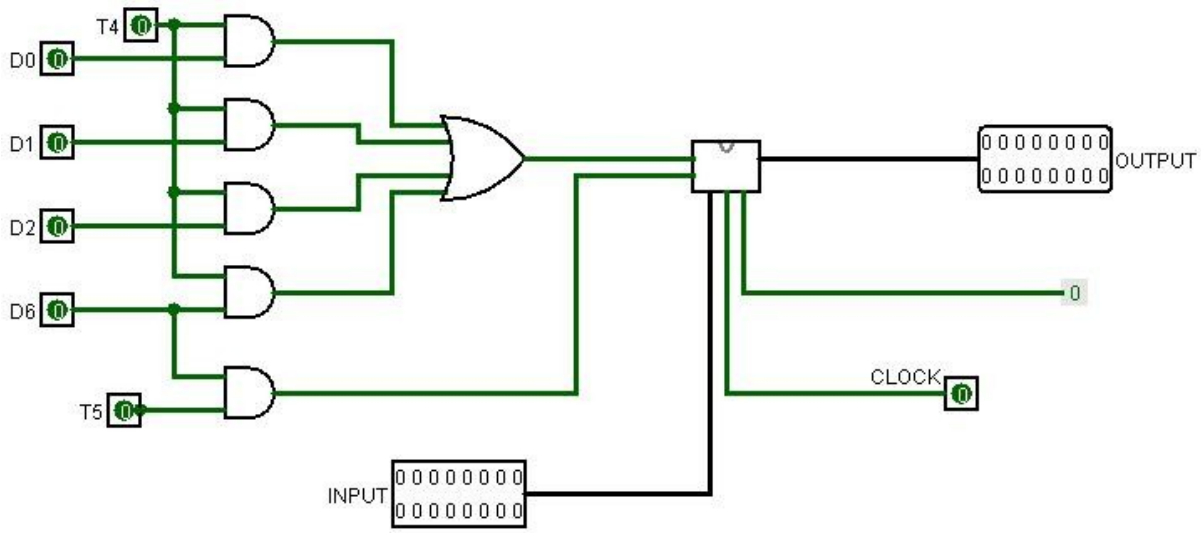


FIG -1.4

1. It is a 16 bit Data Register which holds memory operand to perform arithmetic and logical operations between AC and DR through ALU.
2. Input of DR is always from BUS and Output to BUS and also to ALU.
3. Function of DR –
   LD – (D0 + D1 + D2 + D6)T4
   INR – D6T5

5. IR

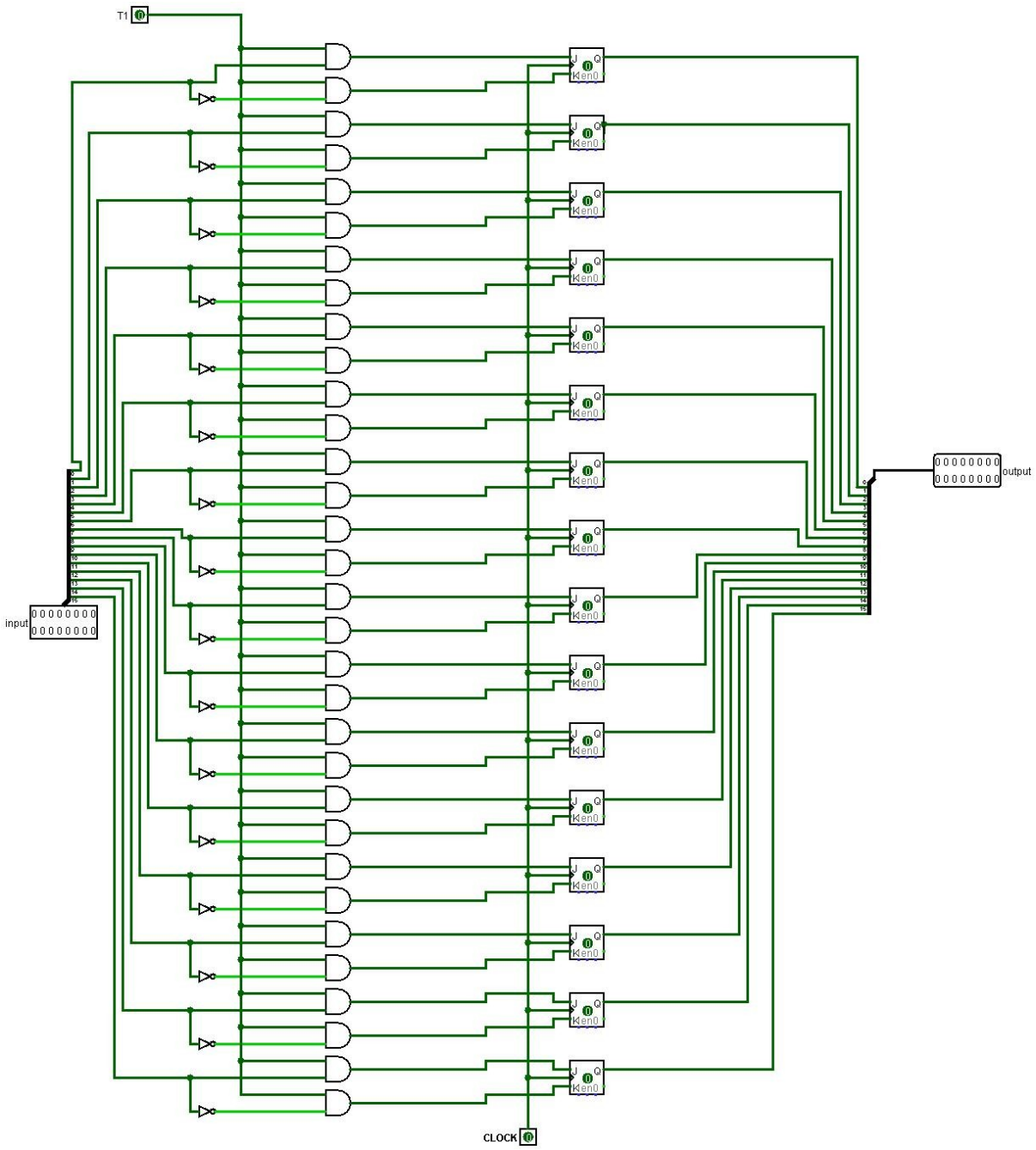


FIG – 1.5

1. It is a 16 bit Instruction Register which hold instruction code to perform specific operation.
2. LSB 12 bit data is transferred to AR

Bit 12,13 and 14 are connected to decoder as input and output of decoder is transferred to control logic gate.

3. Bit 15 is transferred to 1 bit flag flip flop(I) which is used to identify direct or indirect address during any operation.
4. Function of IR –
   LD – T1
5. INPUT-from BUS
   OUTPUT-to BUS

6. TR

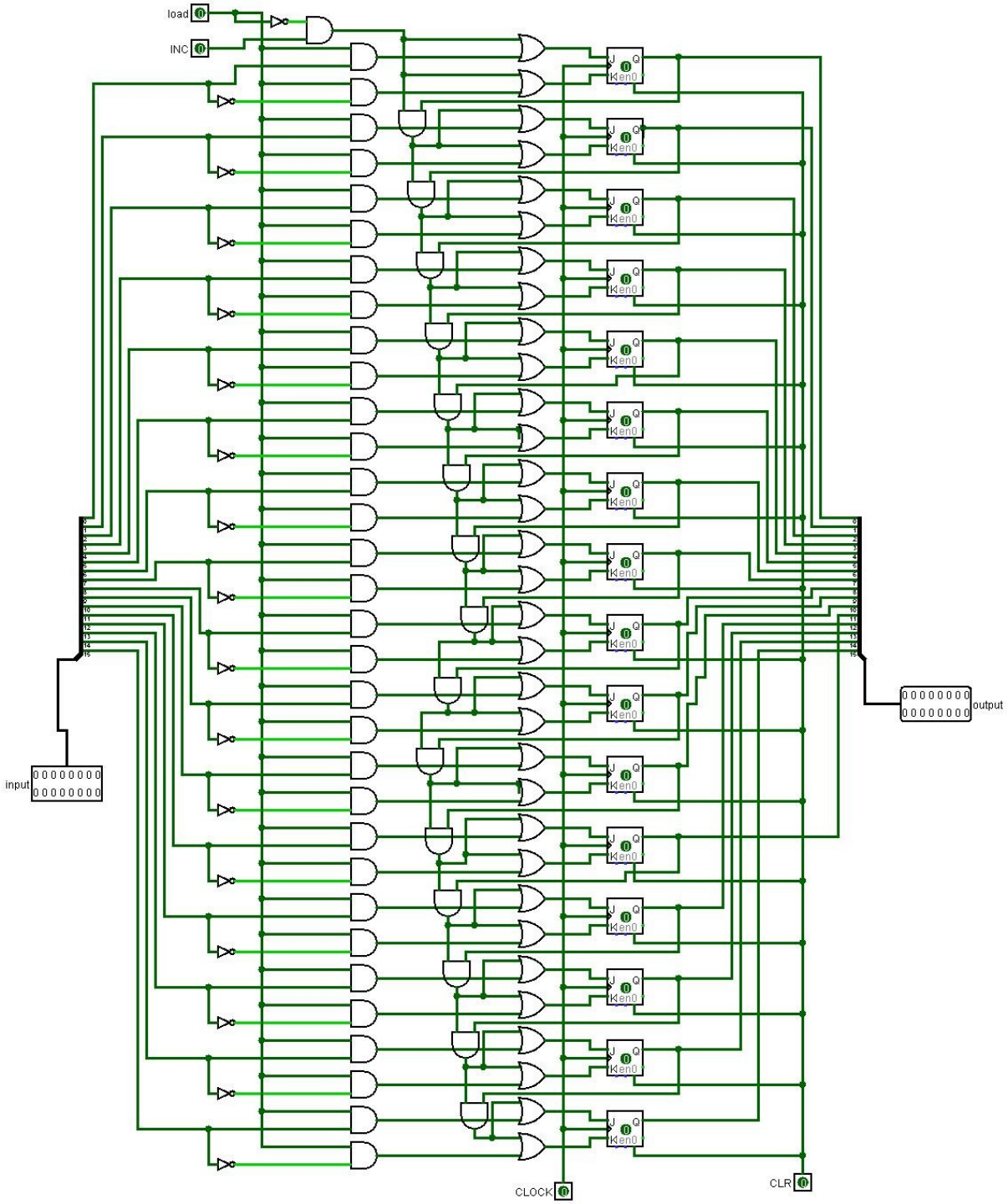


FIG – 1.6

1.It is 16 bit temporary register used to hold temporary data program execution.

2.There is no special use of TR in circuitory of computer 1 without interrupt.

3.INPUT-from BUS

OUTPUT-to BUS

7. INPR

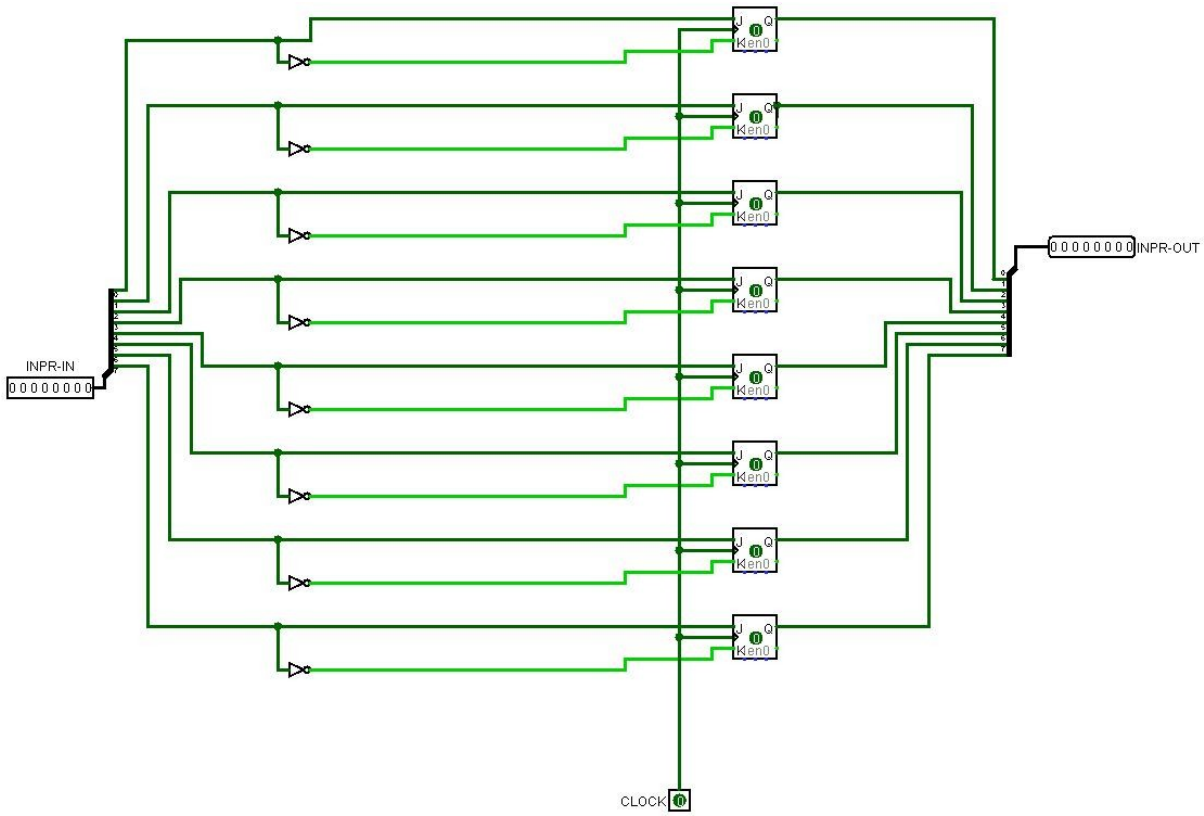


FIG-1.

1.It is 8 bit input register to hold input character from keyboard only when FGI=0.

2.Input is from keyboard and output is to ALU.

3.INPR is used only during I/O reference instruction and I/O interrupt.

8. OUTR

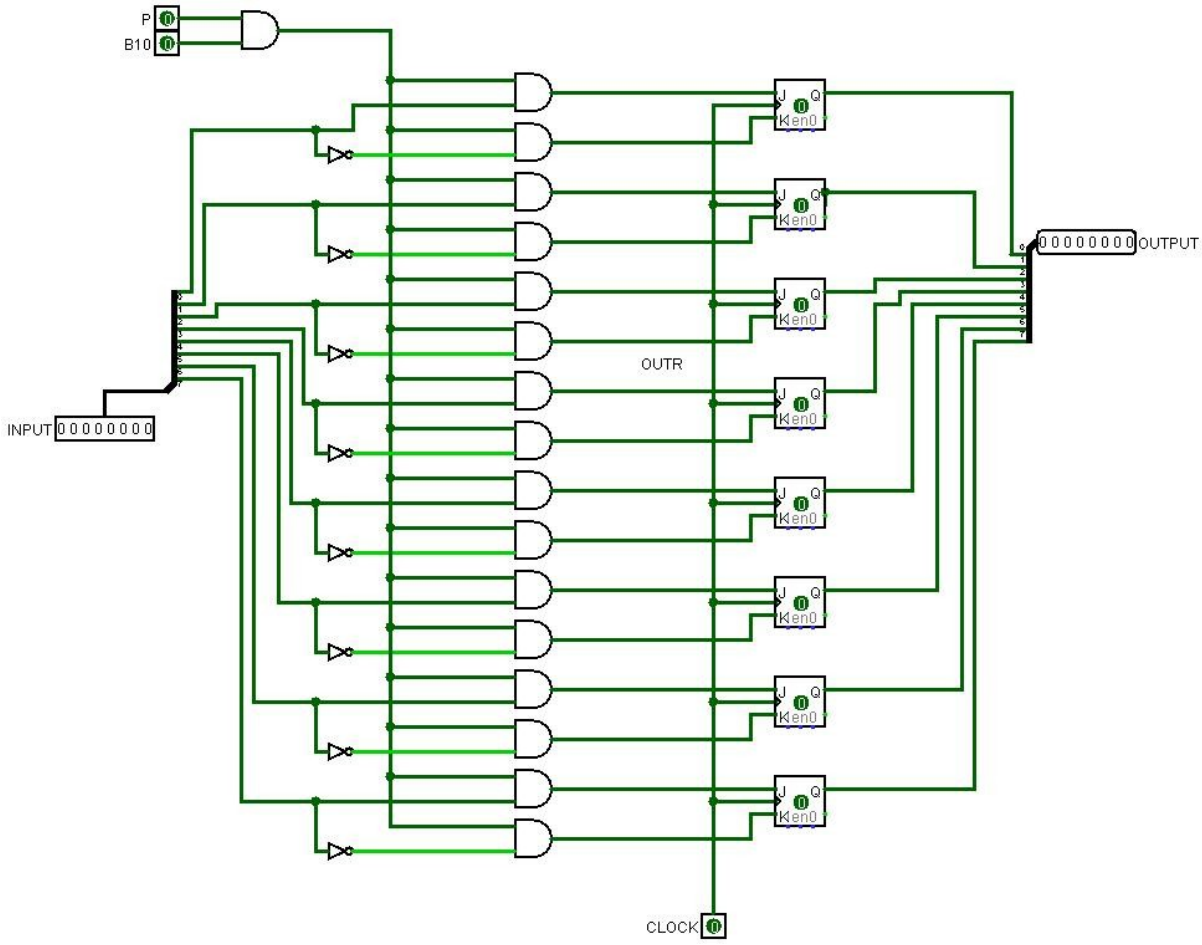


FIG-1.8

1.It is 8 bit output register which holds output character which is shifted from bus.

2.OUTR doesn’t have any output.

3.INPUT-8 bit LSB data from bus

4.Only used in I/O reference instruction and I/O interrupt.

5.Function- LD-PB10

9. ALU

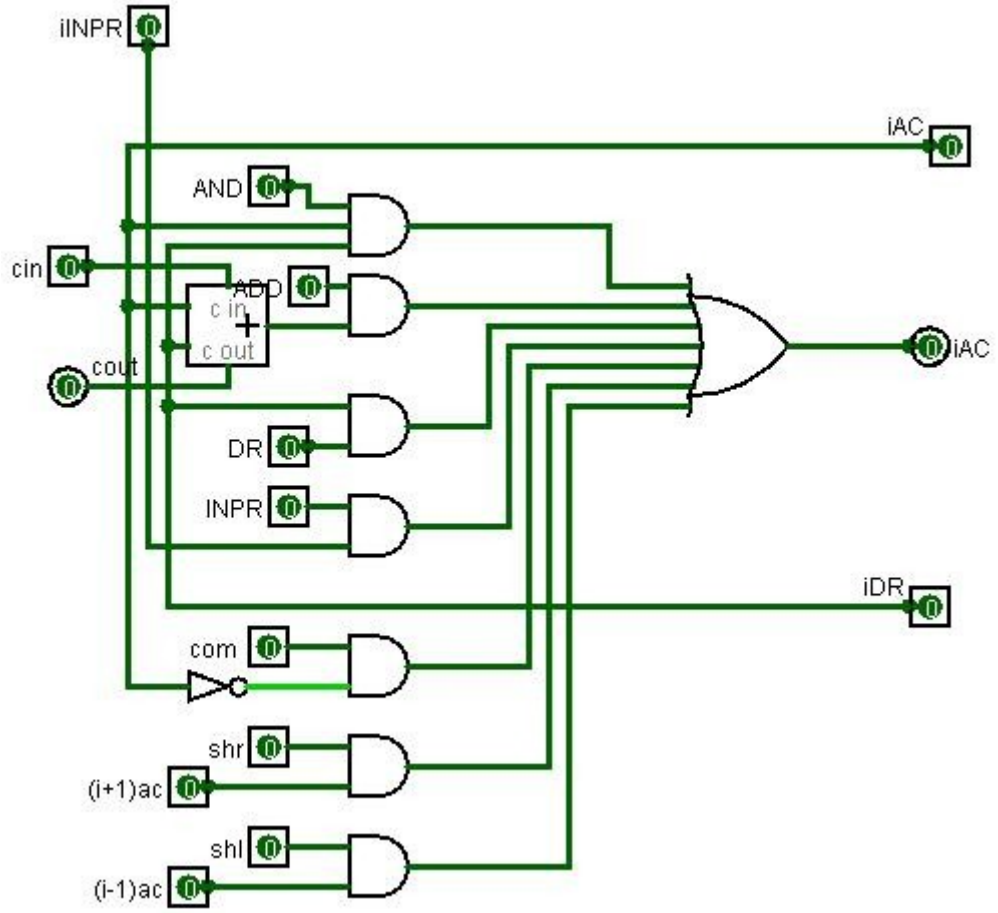


FIG-1.9.1

1bit adder and logic circuit then we changed it to 16 bit adder and logic circuit.

1.Adder and logic circuit can be divided into 16 stages corresponding bit of AC register .

2.Output of ALU is input for AC register and input for ALU is AC,DR and INPR.

3.Extra 1 bit flag register is used to hold carry of add operation and also used to perform left shift and right shift operations.

4.Operations performed in ALU :

i) AND b/w AC and DR.

ii) ADD b/w AC and DR.

iii) Data transfer from DR and INPR.

iv) To find complement of AC.

v) To perform operation of left shift and right shift.

vi)To perform inc and clr operation for accumulator.

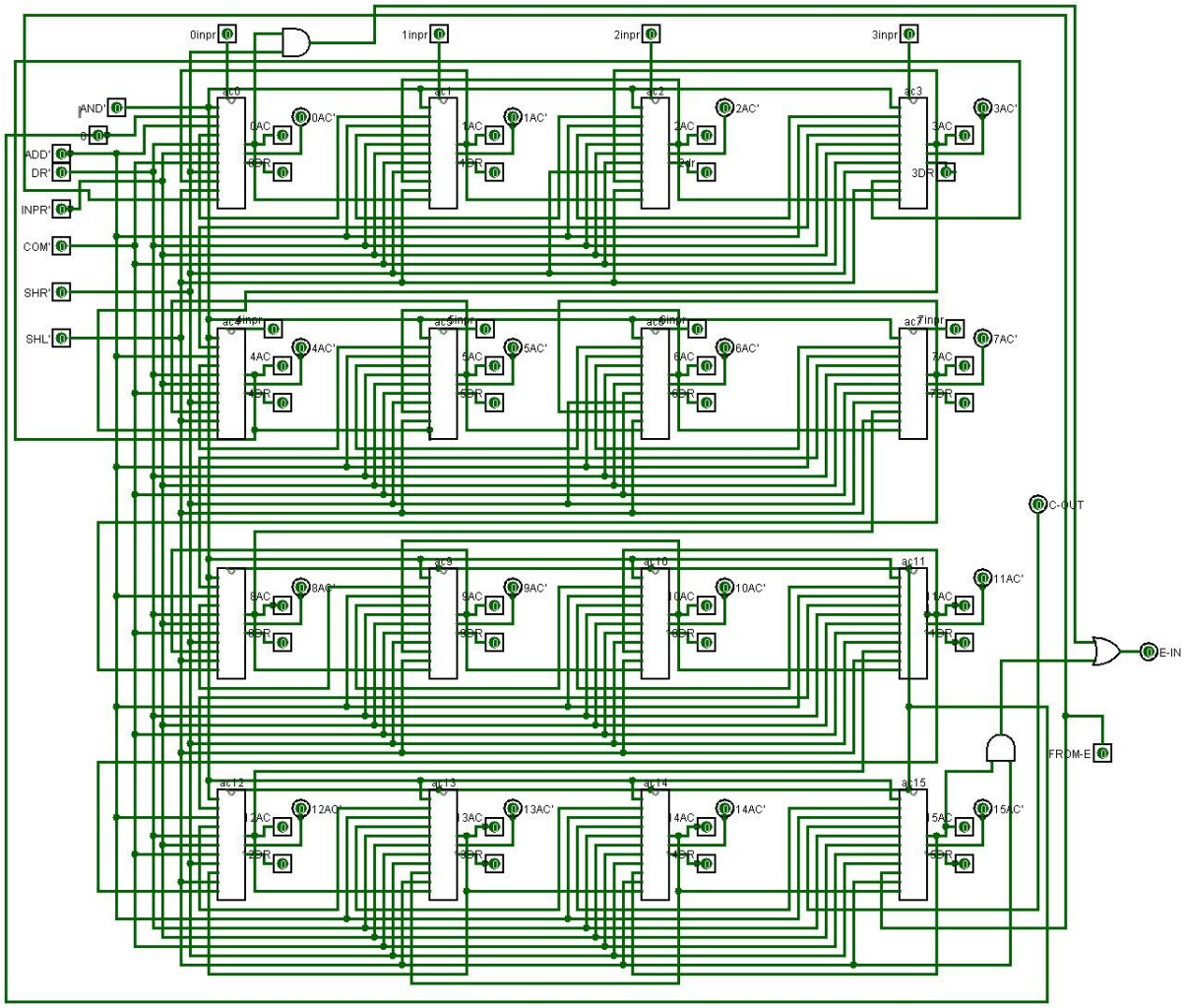
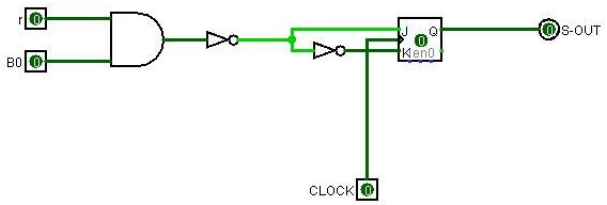


FIG-1.9.2

S F/F



1.Throughout in program execution S F/F Remains set. At the end of program it becomes reset.

2. It is only to identify halt condition of program.

HLT – rB0 (S <- 0)

10. AC

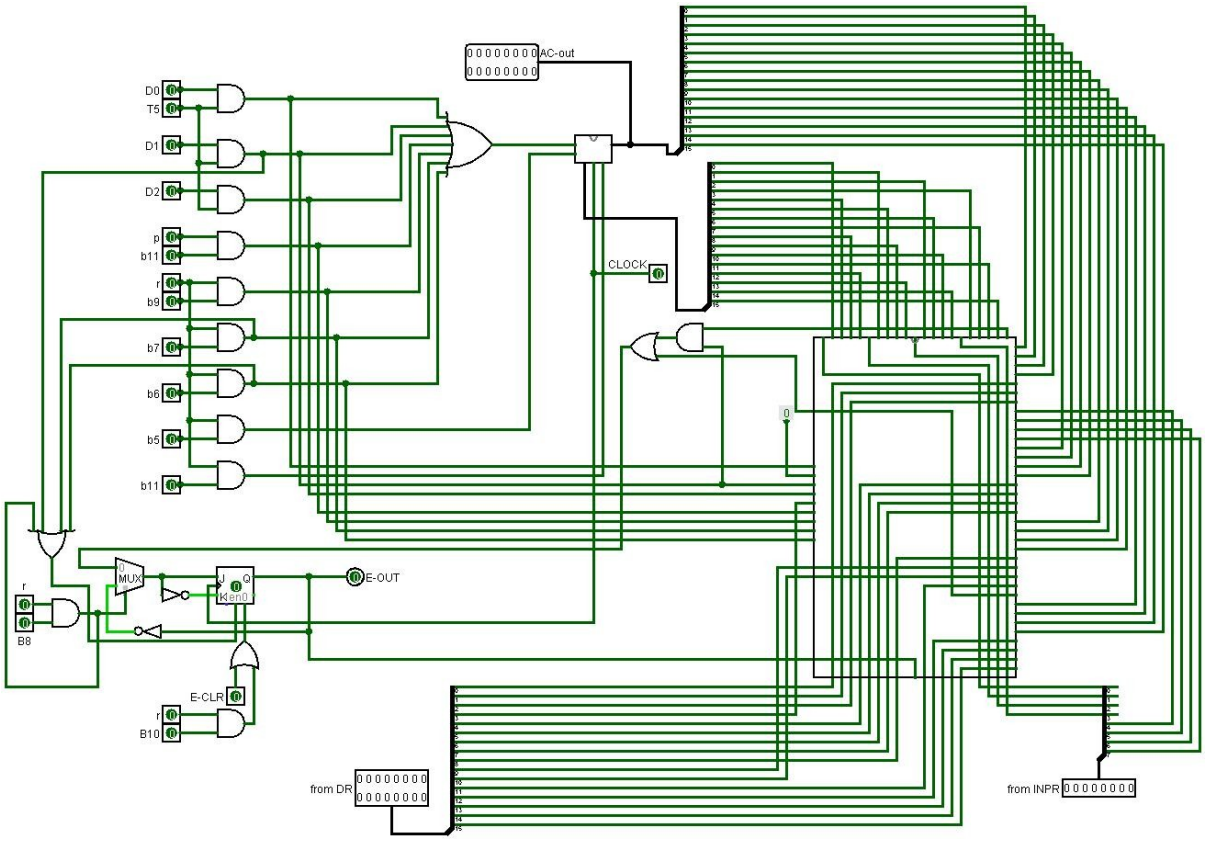


FIG – 1.10

1. It is 16 bit general purpose register(accumulator).
2. It is used to store data of performed operations in ALU.
3. INPUT-always from ALU
   OUTPUT-to ALU and bus
4. Function of ALU:
   LD – (D0 + D1 + D2)T5 +PB11 + (B9 + B7 + B6)r
   INR – rB5
   CLR – rB11

E F/F

LD – D1T5 + r(B6 + B7 + B8)

CLR – rB10

(As shown in FIG – 1.10)

11. RAM

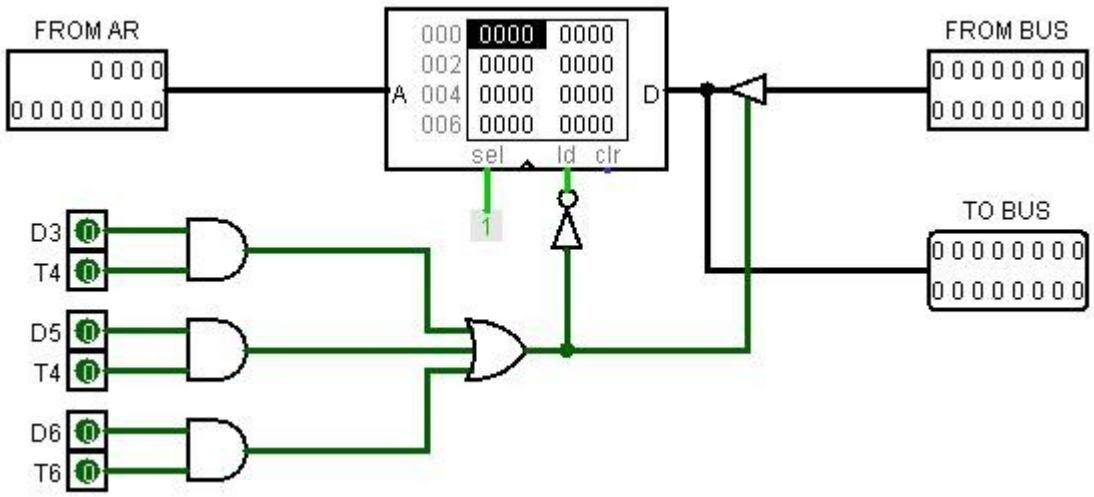


FIG – 1.11

1. RAM Size - 4096 x 16

Data size -16 bits

Address size -12 bits

2. Chip select input is always constant 1 because there is only one main memory unit.

3. INPUT-from bus

OUTPUT-to bus

RAM is always associated with AR register to find memory reference.

4. Functions of RAM : LD – (D3T4 + D5T4 + D6T6)

(As shown in FIG – 1.11)

12. I F/F

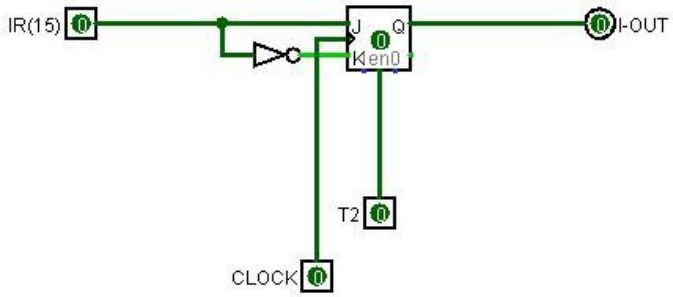


FIG - 1.12

1. It is 1 bit flag F/F used to store 15th bit of IR register.
2. Functions of I:

   LD – T2

3. INPUT – IR(15)

(As shown in FIG – 1.12)

13. SC

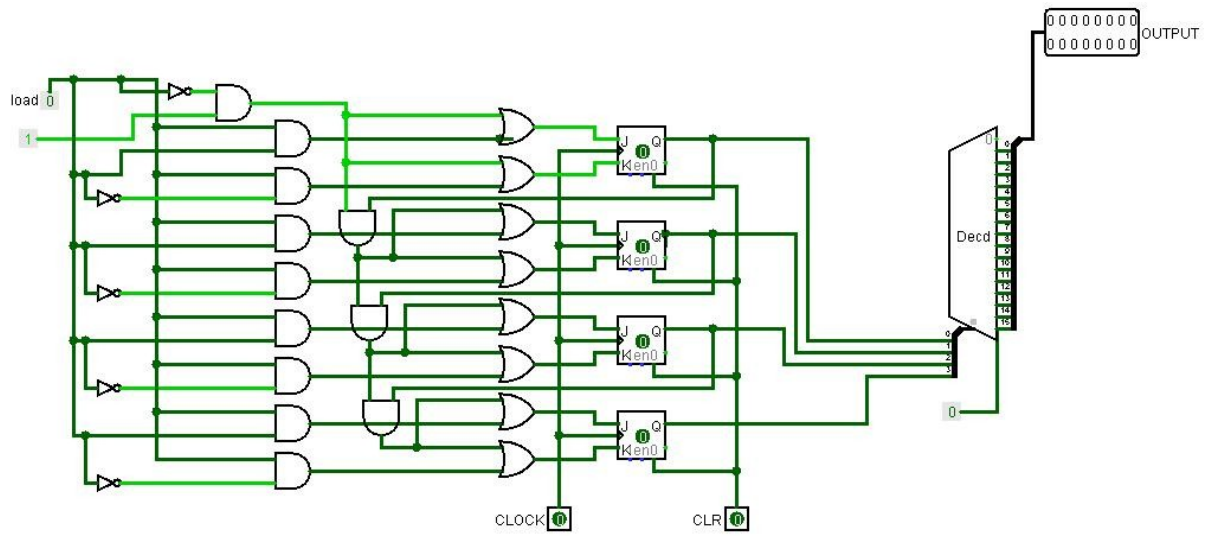


FIG – 1.13.1

LD always disabled.

INR always enabled.

Remove 1st bit of decoder to start time at T0 and add extra last bit(16th).

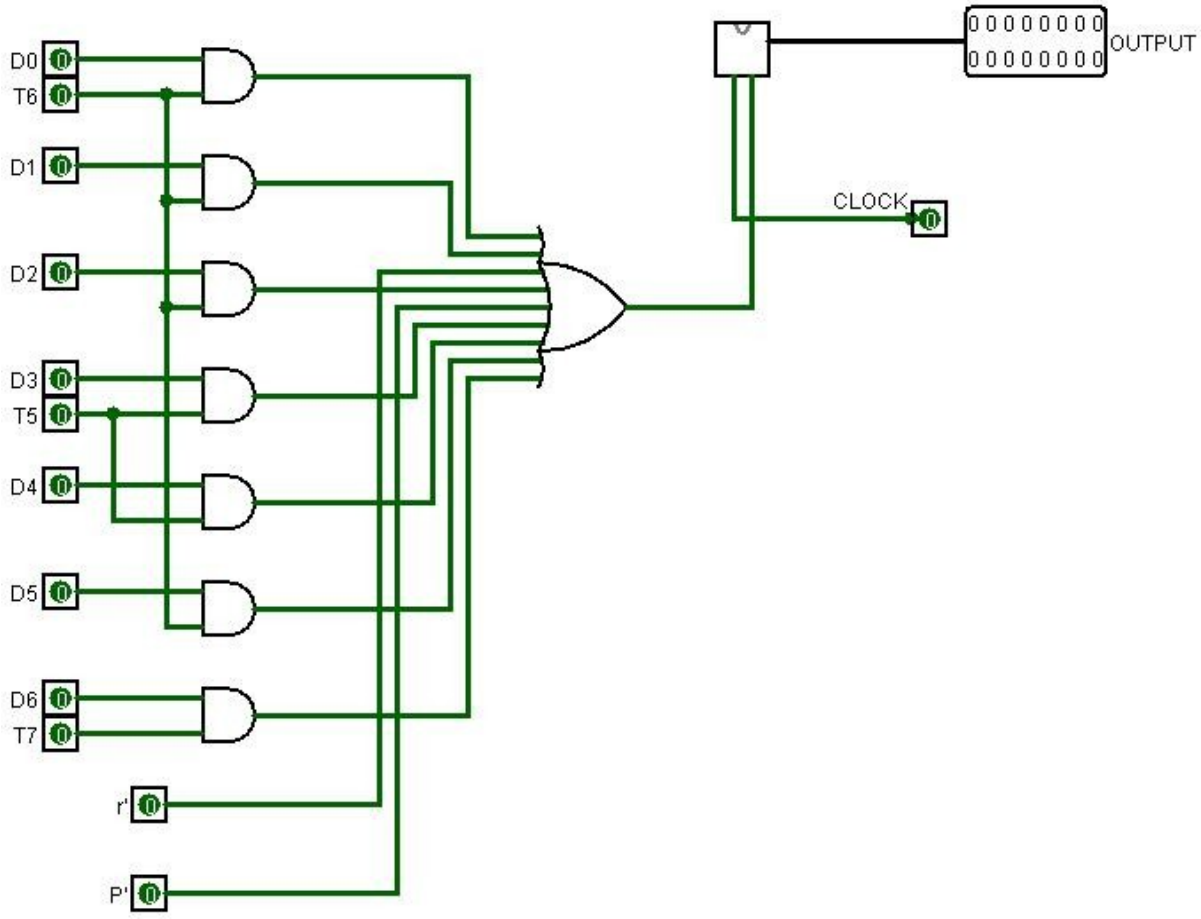


FIG – 1.13.2

CLR – (D0 + D1 + D2 + D5)T6 + (D3 + D4)T5 + D6T7 + D7I’T4 + D7IT4

Connect all register and f/f to each other as requirement,output of all registers is connected to common bus,selection line of common bus select one of all the register where selection line is controlled by priority encoder.Output of bus is connected to all register (except AC and memory unit).When any register has load enabled, data is transferred from bus to register.When write is enabled of memory unit then data is written to the given memory location.Clock time input of all register is connected from decoder of sequence counter.

Input of D0-D7 of all registers is from 3x8 decoder which is associated with bit no. 12,13,14 of IR register.

Connect all registers and f/f by common clock.

When S f/f becomes reset stop pressing the clock (end of program).

COMPUTER 1

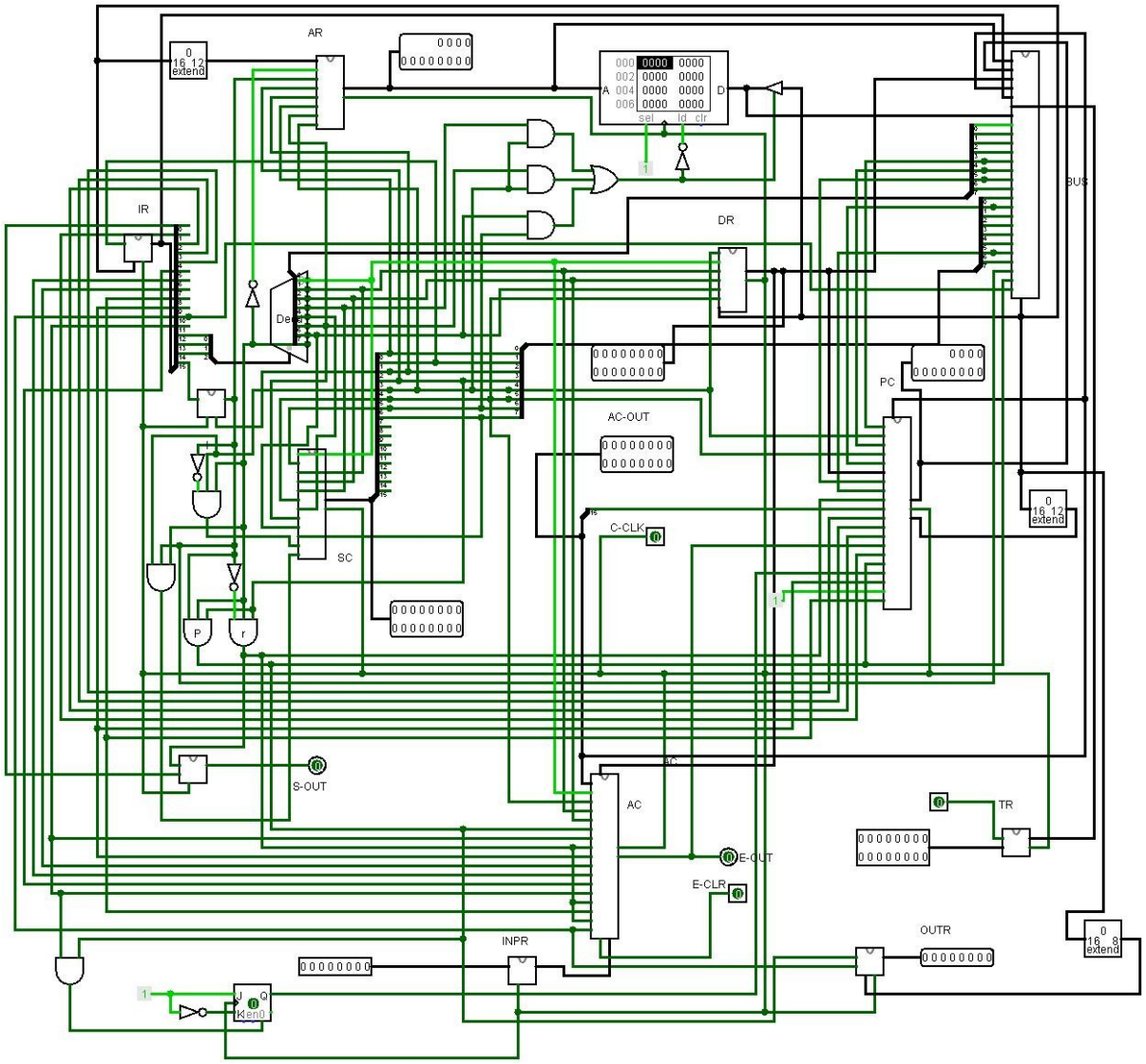


TABLE 5-2 Basic Computer Instructions

Symbol	Hexadecimal code $I = 0$	Hexadecimal code $I = 1$	Description
AND	0xxx	8xxx	AND memory word to *AC*
ADD	1xxx	9xxx	Add memory word to *AC*
LDA	2xxx	Axxx	Load memory word to *AC*
STA	3xxx	Bxxx	Store content of *AC* in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear *AC*
CLE	7400		Clear *E*
CMA	7200		Complement *AC*
CME	7100		Complement *E*
CIR	7080		Circulate right *AC* and *E*
CIL	7040		Circulate left *AC* and *E*
INC	7020		Increment *AC*
SPA	7010		Skip next instruction if *AC* positive
SNA	7008		Skip next instruction if *AC* negative
SZA	7004		Skip next instruction if *AC* zero
SZE	7002		Skip next instruction if *E* is 0
HLT	7001		Halt computer
INP	F800		Input character to *AC*
OUT	F400		Output character from *AC*
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

An assembly language program for computer 1 to subtract content of memory location (1023)10 from content of memory location (1024)10 and store result at memory location (1025)10.

We want to perform C=A-B where A is content of memory location 1024 and B is of memory location 1023. For it we move B in Accumulator and compute its 1's complement i.e. B' by CMA and increment content of accumulator by 1 i.e. B'+1 and add to A. therefore we are computing (B'+1)+A and it is equal to A-B and finally store result at memory location 1025.

1. 0 LDA 1023
2. CMA
3. INC
4. 0 ADD 1024
5. 0 STA 1025
6. HLT

Conversion of above assembly language program into machine language

I	OPCODE	OPERAND
0	0 0 1	0 0 1 1 1 1 1 1 1 1 1 1
0	1 1 1	0 0 1 0 0 0 0 0 0 0 0 0
0	1 1 1	0 0 0 0 0 0 1 0 0 0 0 0
0	0 0 1	0 1 0 0 0 0 0 0 0 0 0 0
0	0 1 1	0 1 0 0 0 0 0 0 0 0 0 1
0	1 1 1	0 0 0 0 0 0 0 0 0 0 0 1

Initially PC is set to 000(hexadecimal code).

Instruction on memory location

Memory Location	Instruction code
000	23FF
001	7200
010	7020
011	1400
100	3401

101	7001
3FF	DATA 1
400	DATA 2

Keep pressing clock till halt is not reset.

When halt is reset stop pressing key , reset of halt indicates end of program thats why we write last instruction of halt.