UNIVERSITY OF BUEA

P.O. Box 63.

Buea, South West Region

CAMEROON

Tel: (273)33322134/33322690

Fax: (273)33322272

REPUBLIC OF CAMEROON

Peace-Work-Fatherland

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTEMENT OF COMPUTER ENGINEERING

COURSE : COMPUTER ARCHITECTURE. **CEF 250**

COURSE COORDINATOR: DR VALERY NKEMENI

PROJECT TITLE :

BUILDING A 16 BIT MICRO-COMPUTER

Group 27 Members:

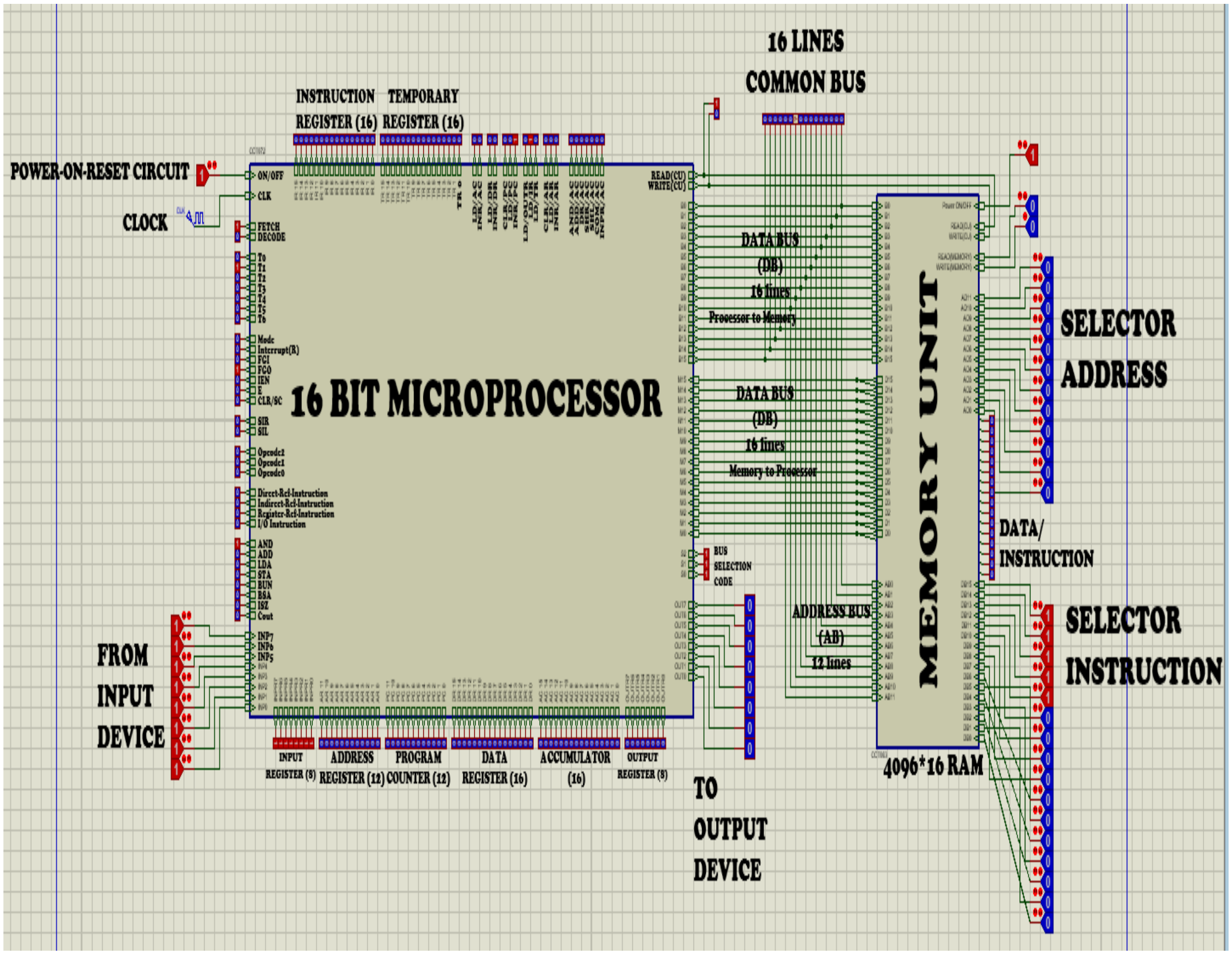
1. **KAMCHE YANN ARNAUD-FE21A208**
2. **AHOUMO TEMATEU ROXANE PHILIPINE-FE21A128**
3. **NTUI RAOUL NTUI NJOCK-FE21A288**
4. **CHO WESLEY MUNGO-FE21A160**
5. **DJIALEU TANKOUA KEANE RANDY -FE21A170**
6. **KOUOTOU AHMAD BILAL-FE21A221**
7. **NJOBAM ROSINE DZELAMONYUY-FE21A396**
8. **DJUIKO METSANOU MERIEM TRESOR-FE21A355**
9. **JULIE ENANGA NJOMBE-FE21A371**

Second Semester Continuous Assessment report, First year (B. ENG) Degree in Computer Engineering.

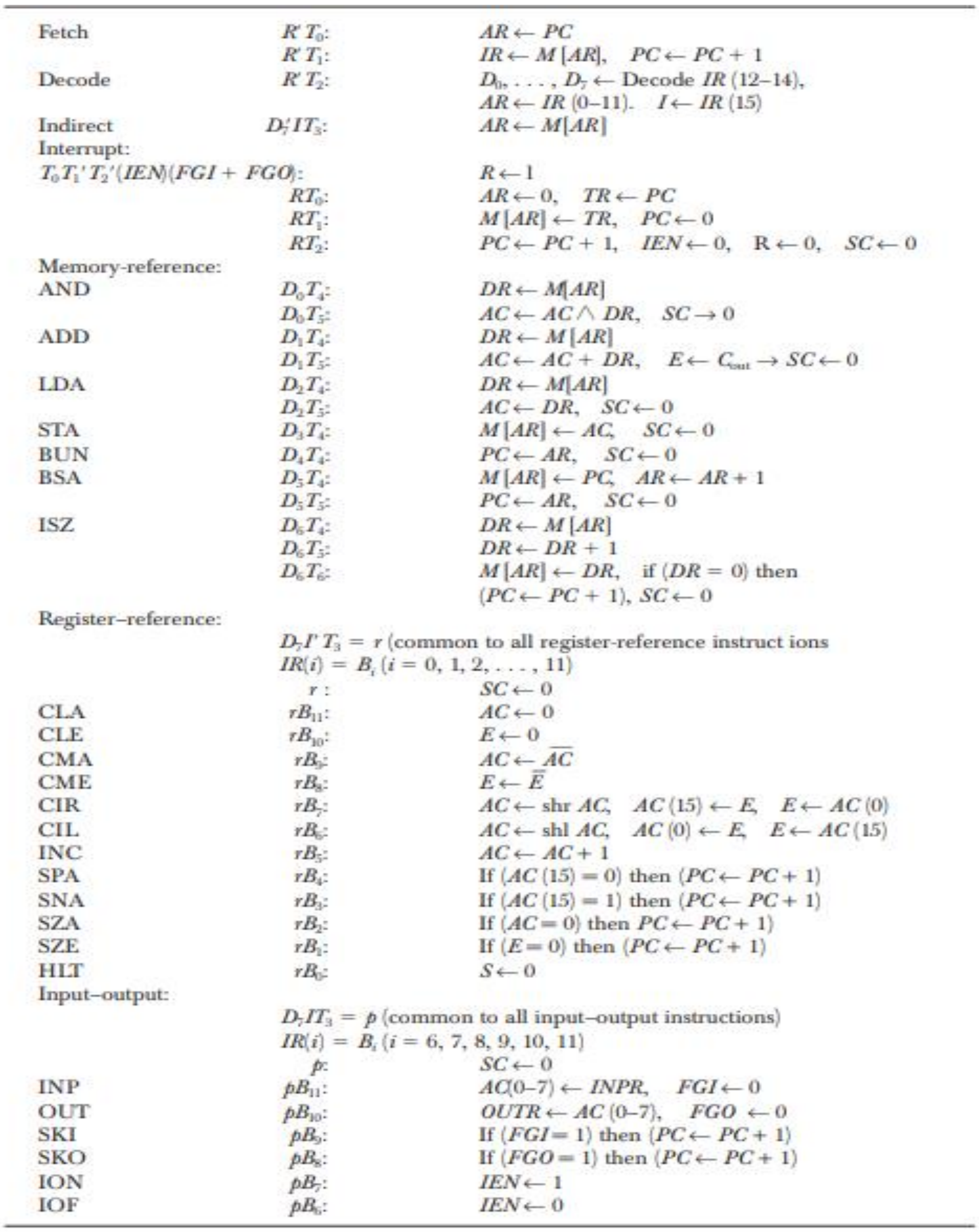
MAY
2022

TABLE OF CONTENTS:

THE 16 BIT COMPUTER



INSTRUCTION SET



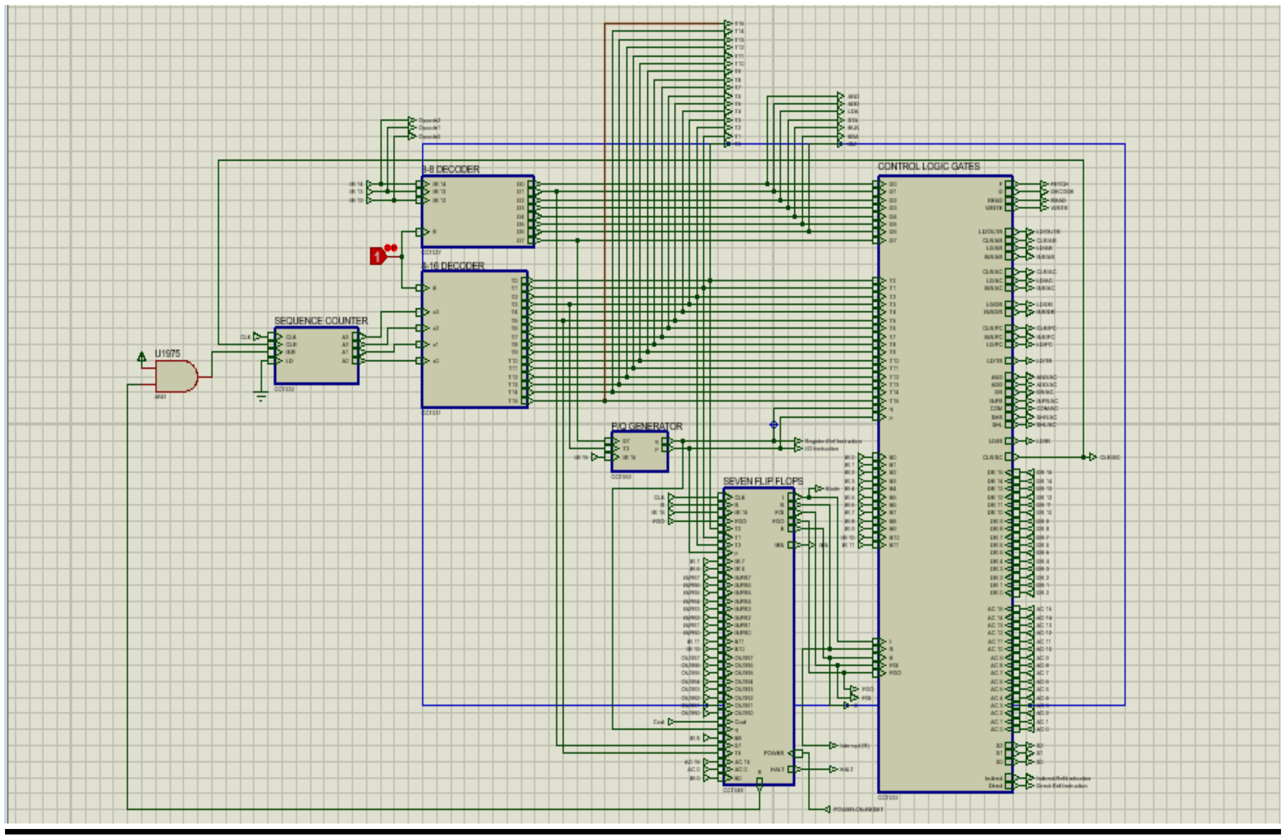
Fetch	$R'T_0$:	$AR \leftarrow PC$
	$R'T_1$:	$IR \leftarrow M[AR]$, $PC \leftarrow PC + 1$
Decode	$R'T_2$:	$D_0, \ldots, D_7 \leftarrow$ Decode $IR(12\text{–}14)$,
		$AR \leftarrow IR(0\text{–}11)$. $I \leftarrow IR(15)$
Indirect	$D_7'IT_3$:	$AR \leftarrow M[AR]$
Interrupt:		
$T_0'T_1'T_2'(IEN)(FGI + FGO)$:		$R \leftarrow 1$
	RT_0:	$AR \leftarrow 0$, $TR \leftarrow PC$
	RT_1:	$M[AR] \leftarrow TR$, $PC \leftarrow 0$
	RT_2:	$PC \leftarrow PC + 1$, $IEN \leftarrow 0$, $R \leftarrow 0$, $SC \leftarrow 0$
Memory-reference:		
AND	D_0T_4:	$DR \leftarrow M[AR]$
	D_0T_5:	$AC \leftarrow AC \wedge DR$, $SC \rightarrow 0$
ADD	D_1T_4:	$DR \leftarrow M[AR]$
	D_1T_5:	$AC \leftarrow AC + DR$, $E \leftarrow C_{out} \rightarrow SC \leftarrow 0$
LDA	D_2T_4:	$DR \leftarrow M[AR]$
	D_2T_5:	$AC \leftarrow DR$, $SC \leftarrow 0$
STA	D_3T_4:	$M[AR] \leftarrow AC$, $SC \leftarrow 0$
BUN	D_4T_4:	$PC \leftarrow AR$, $SC \leftarrow 0$
BSA	D_5T_4:	$M[AR] \leftarrow PC$, $AR \leftarrow AR + 1$
	D_5T_5:	$PC \leftarrow AR$, $SC \leftarrow 0$
ISZ	D_6T_4:	$DR \leftarrow M[AR]$
	D_6T_5:	$DR \leftarrow DR + 1$
	D_6T_6:	$M[AR] \leftarrow DR$, if $(DR = 0)$ then
		$(PC \leftarrow PC + 1)$, $SC \leftarrow 0$
Register–reference:		
	$D_7I'T_3 = r$ (common to all register-reference instruct ions	
	$IR(i) = B_i$ $(i = 0, 1, 2, \ldots, 11)$	
	r:	$SC \leftarrow 0$
CLA	rB_{11}:	$AC \leftarrow 0$
CLE	rB_{10}:	$E \leftarrow 0$
CMA	rB_9:	$AC \leftarrow \overline{AC}$
CME	rB_8:	$E \leftarrow \overline{E}$
CIR	rB_7:	$AC \leftarrow$ shr AC, $AC(15) \leftarrow E$, $E \leftarrow AC(0)$
CIL	rB_6:	$AC \leftarrow$ shl AC, $AC(0) \leftarrow E$, $E \leftarrow AC(15)$
INC	rB_5:	$AC \leftarrow AC + 1$
SPA	rB_4:	If $(AC(15) = 0)$ then $(PC \leftarrow PC + 1)$
SNA	rB_3:	If $(AC(15) = 1)$ then $(PC \leftarrow PC + 1)$
SZA	rB_2:	If $(AC = 0)$ then $PC \leftarrow PC + 1)$
SZE	rB_1:	If $(E = 0)$ then $(PC \leftarrow PC + 1)$
HLT	rB_0:	$S \leftarrow 0$
Input–output:		
	$D_7IT_3 = p$ (common to all input–output instructions)	
	$IR(i) = B_i$ $(i = 6, 7, 8, 9, 10, 11)$	
	p:	$SC \leftarrow 0$
INP	pB_{11}:	$AC(0\text{–}7) \leftarrow INPR$, $FGI \leftarrow 0$
OUT	pB_{10}:	$OUTR \leftarrow AC(0\text{–}7)$, $FGO \leftarrow 0$
SKI	pB_9:	If $(FGI = 1)$ then $(PC \leftarrow PC + 1)$
SKO	pB_8:	If $(FGO = 1)$ then $(PC \leftarrow PC + 1)$
ION	pB_7:	$IEN \leftarrow 1$
IOF	pB_6:	$IEN \leftarrow 0$

<u>THE CONTROL UNIT</u>

This is sometimes referred as the master of the computer processes because it lets the computer’s logic unit, memory and both input and output devices know how to respond to instructions received from a program.

Instructions are fetched into the control unit which in turn decodes the instructions so that the **adder and logic circuit** can understand.

As it can be seen below our control unit is made up of the control logic gates, 7-flipflops, PQ-generator, sequence counter, 4x16-decoder and the 3x8-decoder.

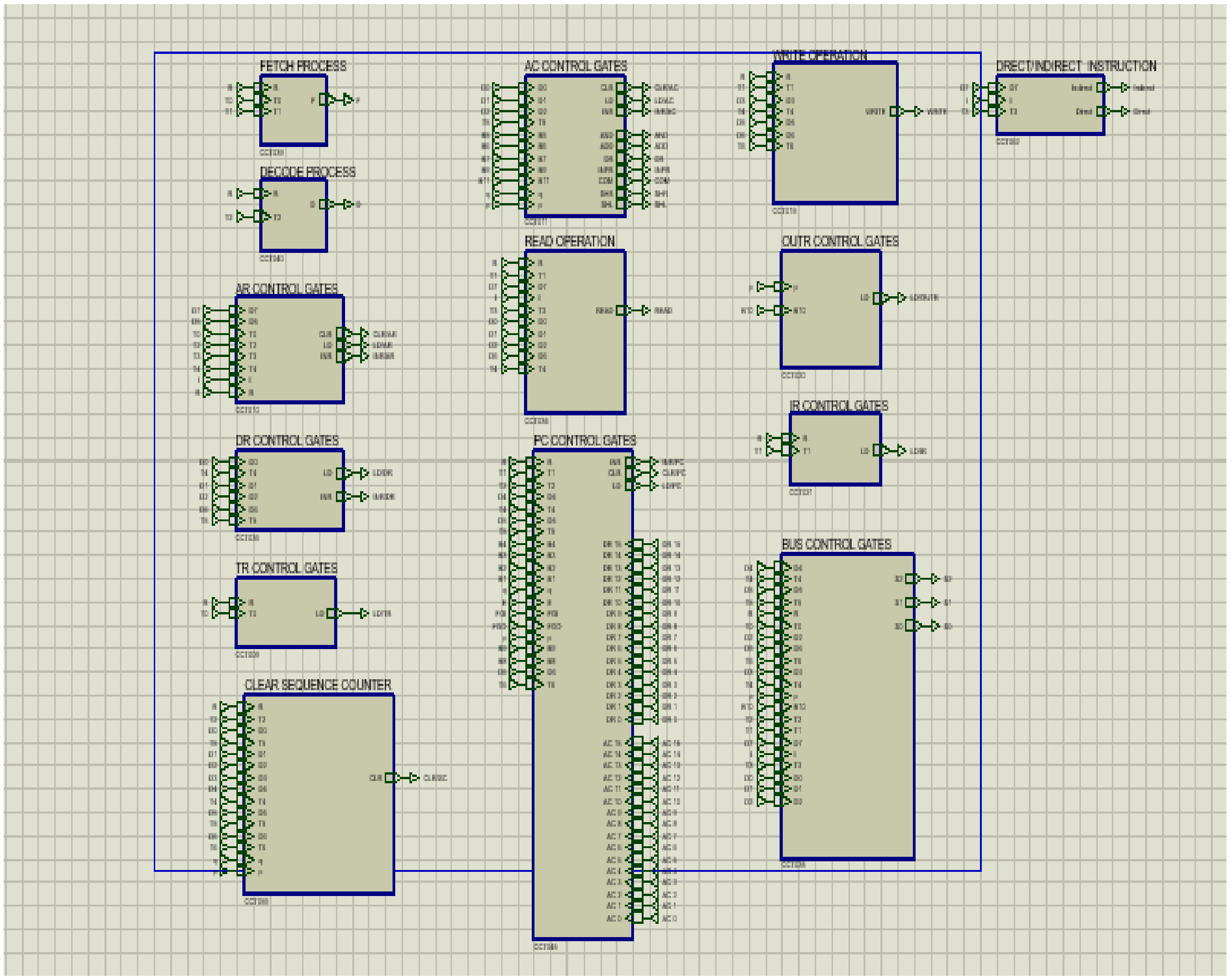


The Control Logic Gates

This is a hardware component of a basic computer system, it comprises of inputs and outputs the diagram below shows hard wiring of the control organization.

Here, we have the 9 control logic gates for the 9registers, fetch and decode process, the bus counter gates, the 7-flipflop as you can clearly observe below.

The control logic gates get its inputs from the 3x8 decoder and the 4x16 decoder, the I-flip-flop, bits 0 through 11 of the instruction register.

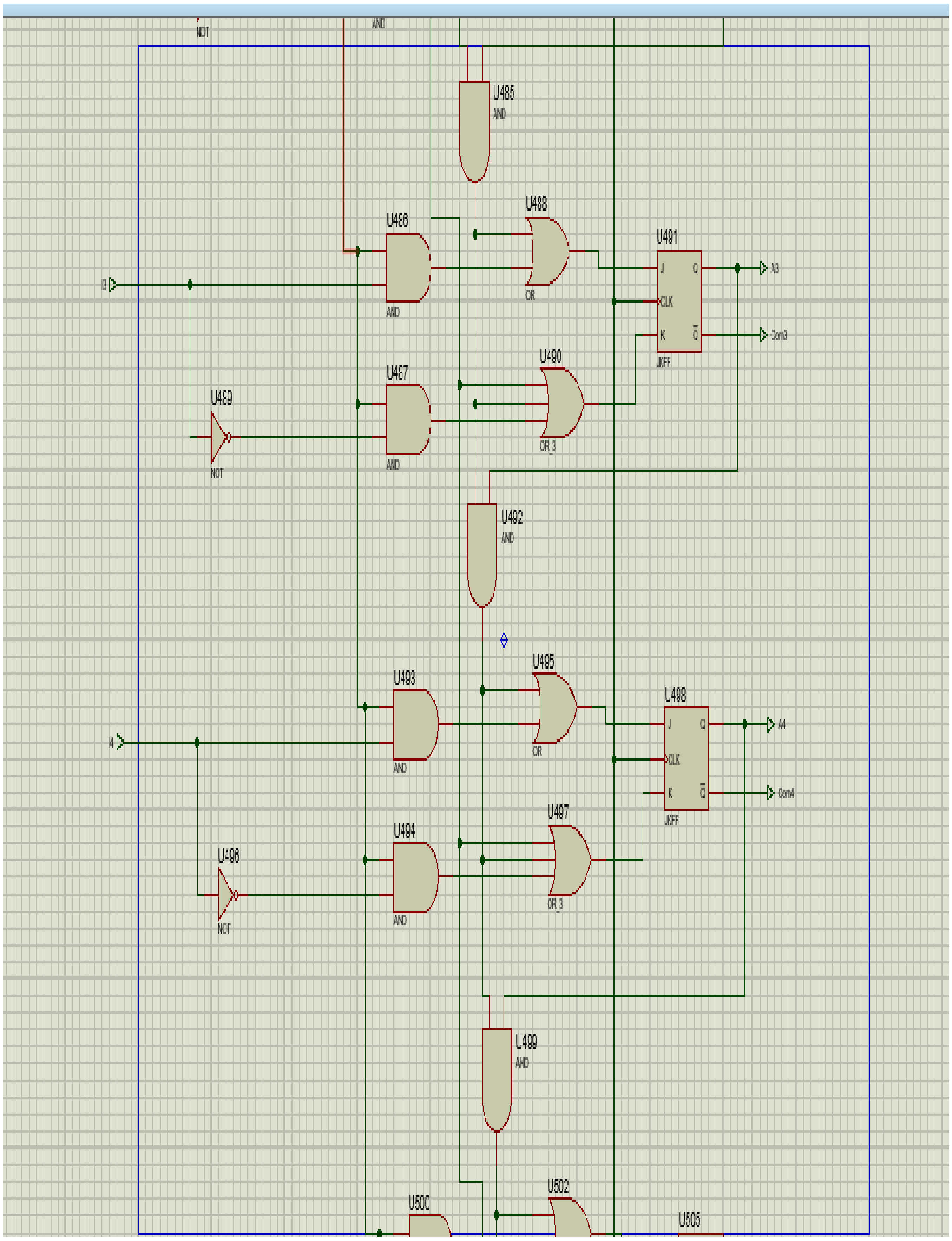


THE REGISTERS

This refers to a group of flip-flops with each flip-flop capable of storing one bit of information, this mean that an n-bit register has n-flip-flops and store n-bit of information. In this project, we shall focus on 9- registers which store their own amount of data and have different number of bits.

The memory address register (AR) has 12 bits since this is the width of a memory address. The program counter (PC) also has 12 bits and it holds the address of the next instruction to be read from memory after the current instruction is executed. The PC goes through a counting sequence and causes the computer to read sequential instructions previously stored in memory. Instruction words are read and executed in sequence unless a branch instruction is encountered. A branch instruction calls for a transfer to a nonconsecutive instruction in the program. The address part of a branch instruction is transferred to PC to become the address of the next instruction. To read an instruction, the content of PC is taken as the address for memory and a memory read cycle is initiated. PC is then incremented by one, so it holds the address of the next instruction in sequence. Two registers are used for input and output. The input register (INPR) receives an 8-bit character from an input device. The output register (OUTR) holds an 8-bit character for an output device.

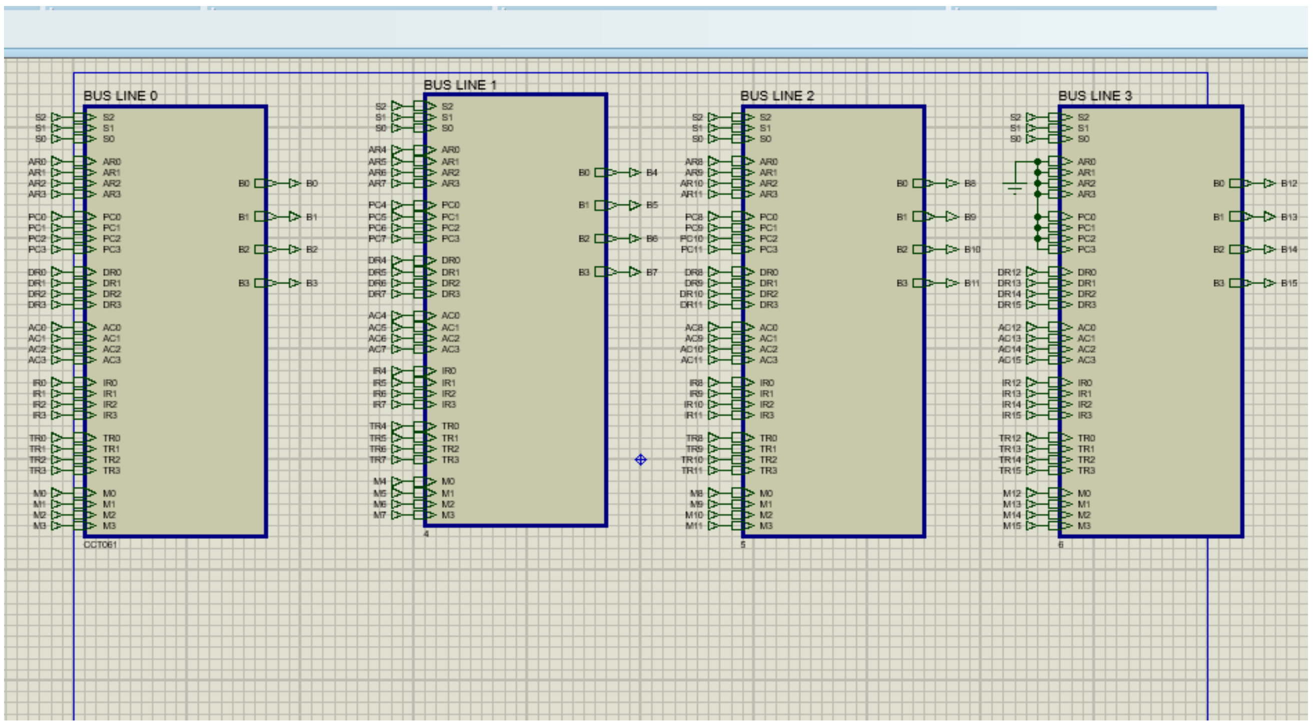
As you can see in the diagram below which illustrates the 9 registers which all have the same circuit diagram but now differ in the number of bits which determines the number of times circuit will be repeated.


NOT
AND
U485
AND
U486
U488
U491
I3
J
Q
A3
CLK
K
Q̄
Com3
AND
OR
JKFF
U487
U490
U489
NOT
AND
OR_3
U482
AND
U483
U495
U498
I4
J
Q
A4
CLK
K
Q̄
Com4
AND
OR
JKFF
U484
U497
U496
NOT
AND
OR_3
U499
AND
U500
U502
U505

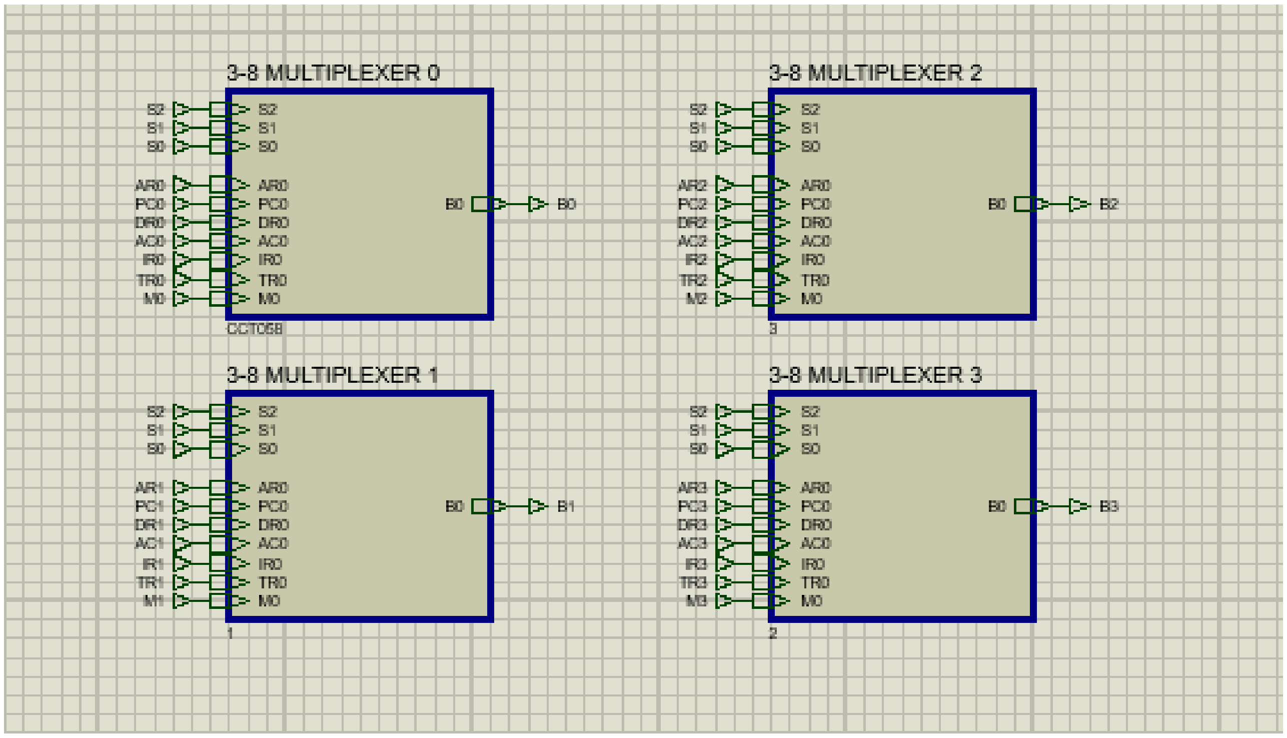
THE BUS

A bus structure consists of a set of common lines, one for each bit of a register, through which binary information is transferred one at a time. Control signals determine which register is selected by the bus during each particular register transfer. One way of constructing a common bus system is with multiplexers. The multiplexers select the source register whose binary information is then placed on the bus. The construction of a bus system for eight registers is shown below. Each register has its own number of bits which ranges from 8-16 bits. The bus consists of sixteen 8x1 multiplexers with 3 selection inputs, S2, S1 and S0. In order not to complicate the diagram with 16 lines crossing each other, we use labels to show the connections from the outputs of the registers to the inputs of the multiplexers. The diagram shows that the bits in the same significant position in each register are connected to the data inputs of one multiplexer to form one line of the bus. Thus MUX 0 multiplexes the sixteen 0 bits of the registers, MUX 1 multiplexes the sixteen 1 bit of the registers, and similarly for the other two bits.

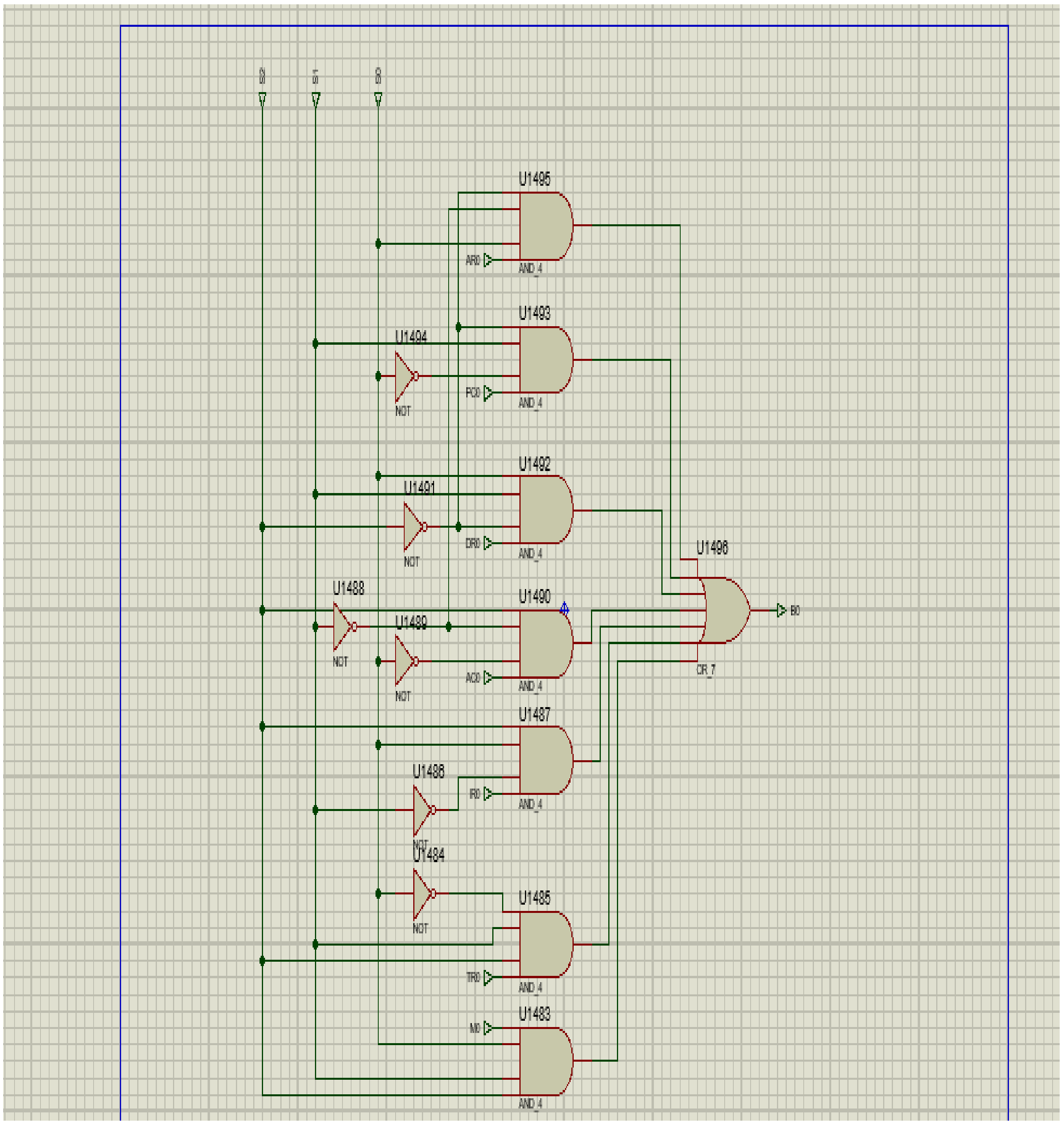
we have four bus which are bus line 0,1,2 and 3. In each bus line you fine four 8x1 MUX giving us a total of sixteen 8x1 MUX as shown in the diagram



Content of one bus line



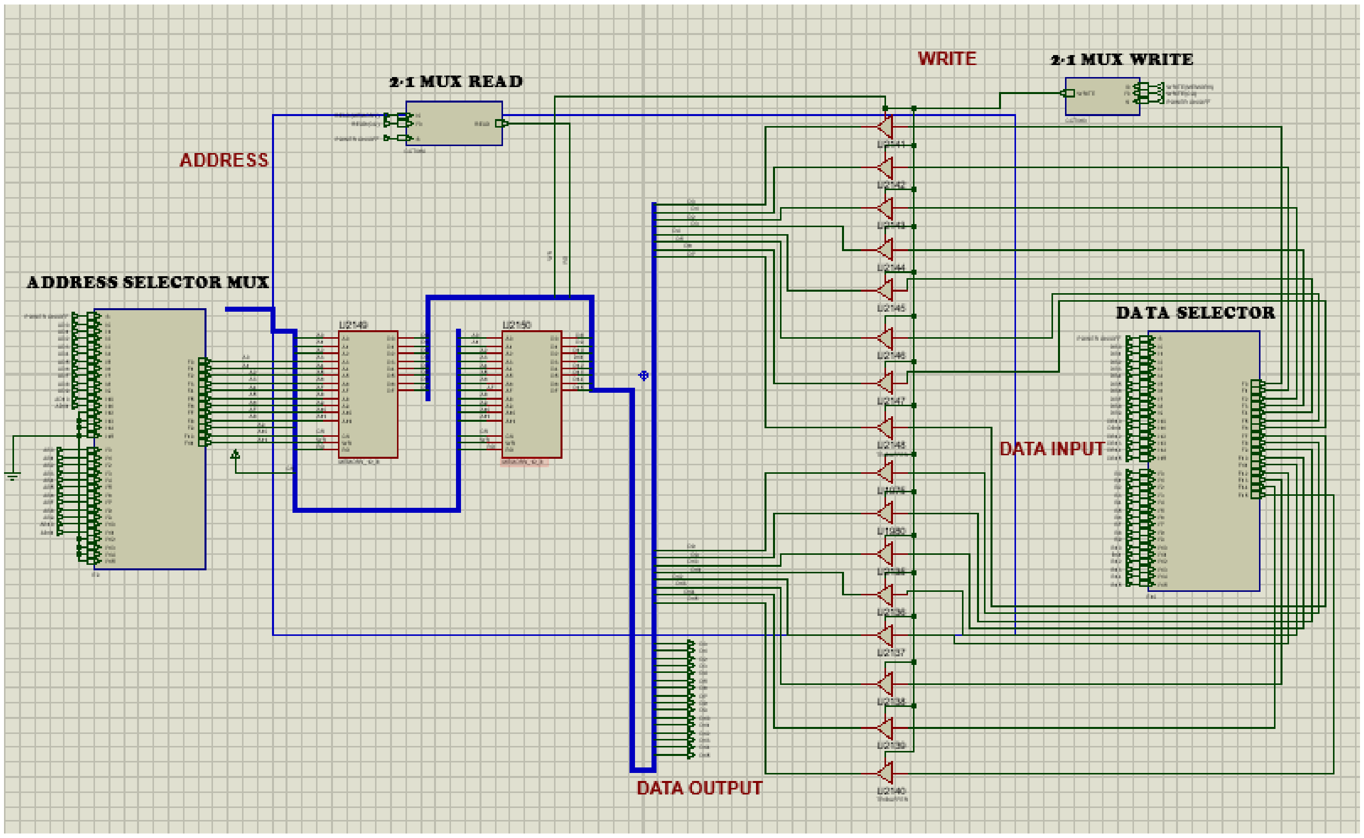
8x1-MUX


S2
S1
S0
U1495
AR0
AND_4
U1493
U1494
NOT
PC0
AND_4
U1492
U1491
NOT
DR0
AND_4
U1498
U1488
U1490
U1489
NOT
NOT
AC0
AND_4
B0
OR_7
U1487
U1486
IR0
AND_4
NOT
U1484
NOT
U1485
TR0
AND_4
U1483
M0
AND_4

THE MEMORY

The memory sometimes known as the storage unit of the computer is divided into the RAM and ROM in our case.

RAM and ROM chips are connected to a CPU through the data and address buses. The low-order lines in the address bus select the byte within the chips and other lines in the address bus select a particular chip through its chip select inputs. The connection of memory chips to the CPU. This configuration gives a memory capacity of 512 bytes of RAM and 512 bytes of ROM. Each RAM receives the seven low-order bits of the address bus to select one of 128 possible bytes. The particular RAM chip selected is determined from lines 8 and 9 in the address bus. This is done through a 2 4 decoder whose outputs go to the CS1 inputs in each RAM chip. Thus, when address lines 8 and 9 are equal to 00, the first RAM chip is selected. When 01, the second RAM chip is selected, and so on. The RD and WR outputs from the microprocessor are applied to the inputs of each RAM chip. The selection between RAM and ROM is achieved through bus line 10. The RAMs are selected when the bit in this line is 0, and the ROM when the bit is 1. The other chip select input in the ROM is connected to the RD control line for the ROM chip to be enabled only during a read operation. Address bus lines 1 to 9 are applied to the input address of ROM without going through the decoder. This assign addresses 0 to 511 to RAM and 512 to 1023 to ROM. The data bus of the ROM has only an output capability, whereas the data bus connected to the RAMs can transfer information in both.

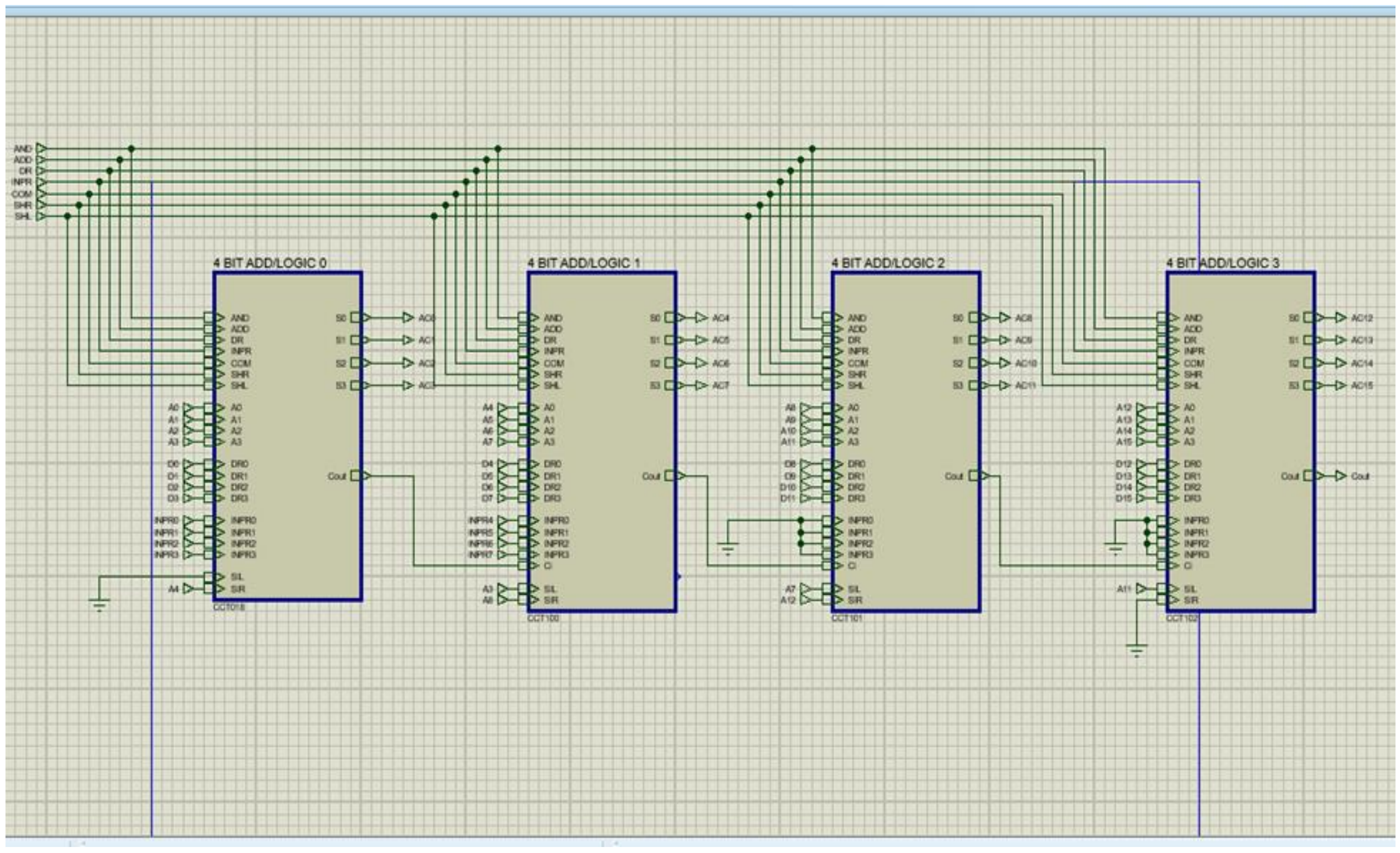

WRITE
2-1 MUX WRITE
2-1 MUX READ
ADDRESS
ADDRESS SELECTOR MUX
U2149
U2150
DATA SELECTOR
DATA INPUT
DATA OUTPUT

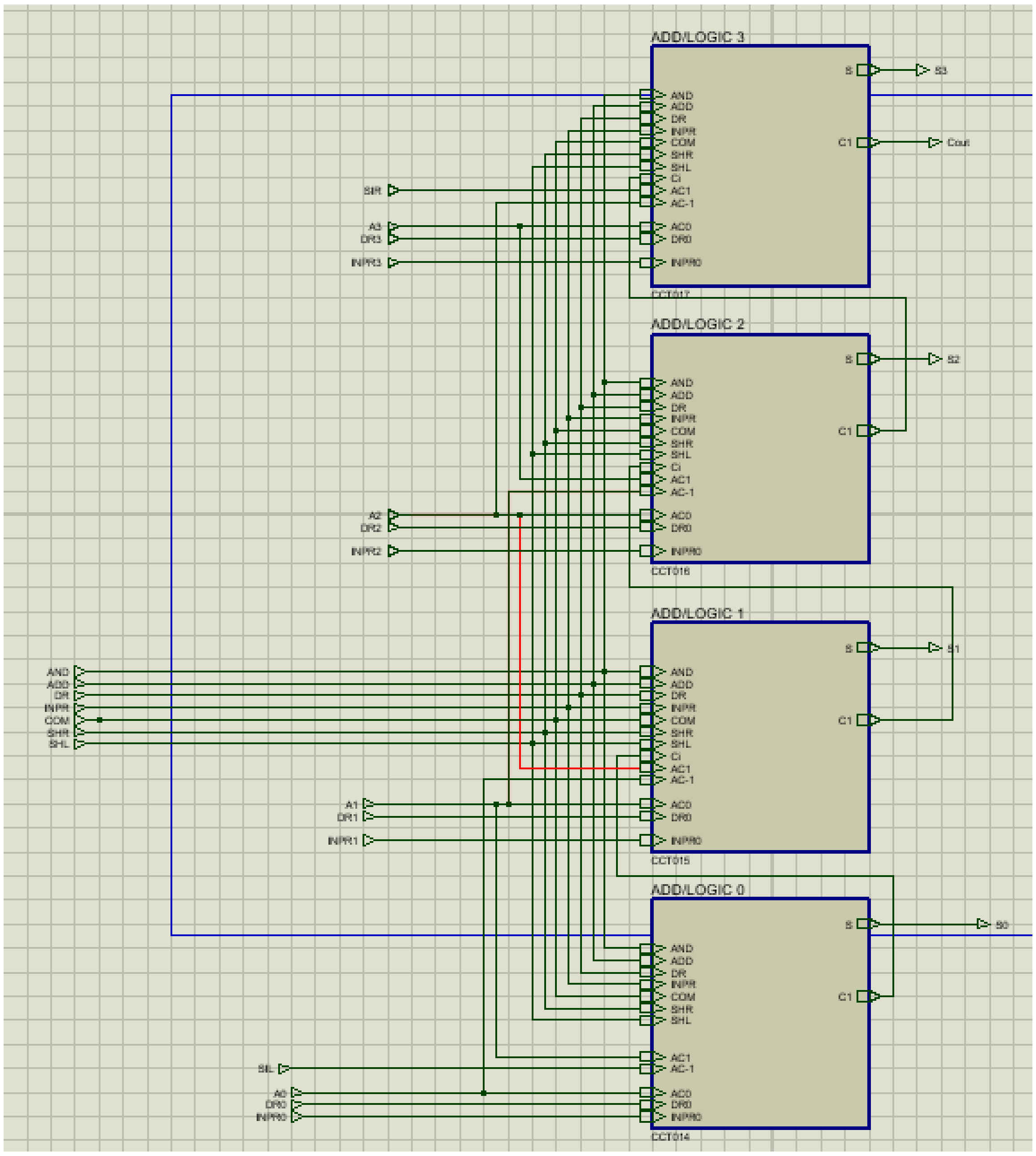
The adder and logic circuit can be subdivided into 16 stages, with each stage corresponding to one bit of AC. The internal construction of the register is as shown in Fig. 2-11. Looking back at that figure we note that each stage has a JK flip-flop, two OR gates, and two AND gates. The load (LD) input is connected to the inputs of the AND gates. Figure 5-22 shows one such AC register stage (with the OR gates removed). The input is labeled Ii and the output AC(i). When the LD input is enabled, the 16 inputs Ii for i 0, 1, 2, . . ., 15 are transferred to AC (0–15). One stage of the adder and logic circuit consists of seven AND gates, one OR gate and a full-adder (FA), as shown in Fig. 5-22. The inputs of the gates with symbolic names come from the outputs of gates marked with the same symbolic name in Fig. 5-21. For example, the input marked ADD in Fig. 5-21 is connected to the output marked ADD in Fig. 5-21. The AND operation is achieved by ANDing AC(i) with the corresponding bit in the data register DR(i). The ADD operation is obtained using a binary adder similar to the one shown in Fig. 4-6. One stage of the adder uses a full-adder with the corresponding input and output carries. The transfer from INPR to AC is only for bits 0 through 7. The complement microoperation is obtained by inverting the bit value in AC. The shift-right operation transfers the bit from AC (i + 1), and the shift-left operation transfers the bit from AC (i

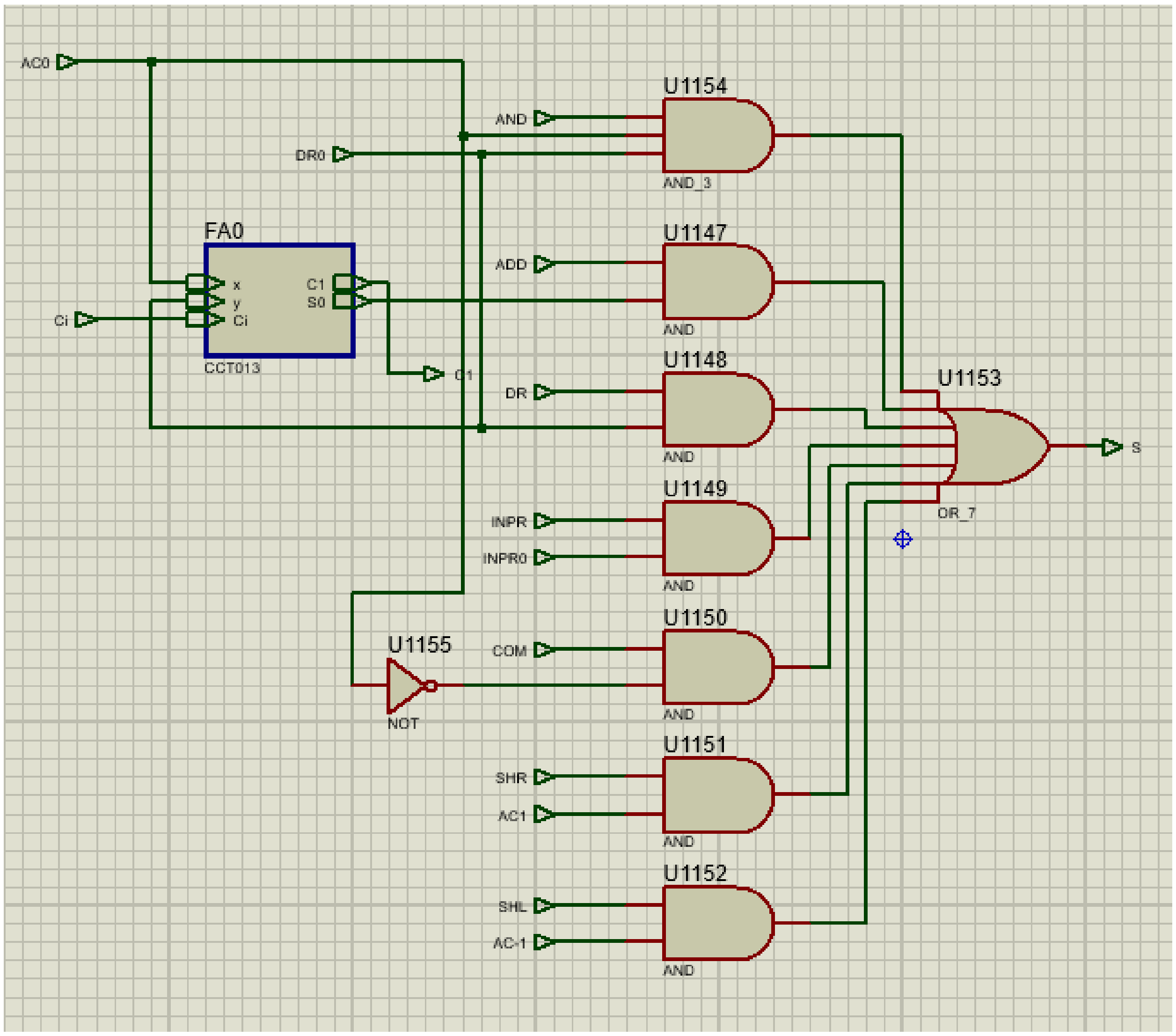
---

1). The complete adder and logic circuit consists of 16 stages connected together.

The diagram below illustrates the circuit diagram of the adder and logic circuit.


AND
ADD
DR
INPR
COM
SHR
SHL
4 BIT ADD/LOGIC 0
4 BIT ADD/LOGIC 1
4 BIT ADD/LOGIC 2
4 BIT ADD/LOGIC 3
A0
A1
A2
A3
DR0
DR1
DR2
DR3
INPR0
INPR1
INPR2
INPR3
SIL
SIR
CI
S0
S1
S2
S3
Cout
AC0
AC1
AC2
AC3
AC4
AC5
AC6
AC7
AC8
AC9
AC10
AC11
AC12
AC13
AC14
AC15
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
INPR4
INPR5
INPR6
INPR7
CCT018
CCT100
CCT101
CCT102


ADD/LOGIC 3
S
S3
C1
Cout
AND
ADD
DR
INPR
COM
SHR
SHL
Ci
AC1
AC-1
AC0
DR0
INPR0
SIR
A3
DR3
INPR3
CCT017
ADD/LOGIC 2
S
S2
C1
AND
ADD
DR
INPR
COM
SHR
SHL
Ci
AC1
AC-1
AC0
DR0
INPR0
A2
DR2
INPR2
CCT016
ADD/LOGIC 1
S
S1
C1
AND
ADD
DR
INPR
COM
SHR
SHL
Ci
AC1
AC-1
AC0
DR0
INPR0
A1
DR1
INPR1
CCT015
ADD/LOGIC 0
S
S0
C1
AND
ADD
DR
INPR
COM
SHR
SHL
AC1
AC-1
AC0
DR0
INPR0
SIL
A0
DR0
INPR0
CCT014


AC0
DR0
AND
U1154
AND_3
FA0
x
y
Ci
C1
S0
CCT013
Ci
C1
ADD
U1147
AND
DR
U1148
AND
U1153
S
OR_7
INPR
INPR0
U1149
AND
U1155
NOT
COM
U1150
AND
SHR
AC1
U1151
AND
SHL
AC-1
U1152
AND