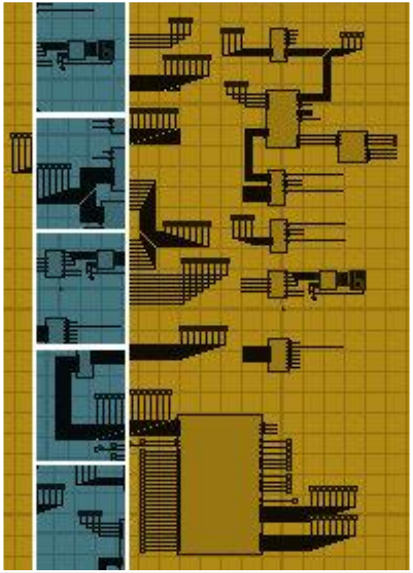


Very Simple Computer Design and Simulation

CSE 404 Digital System Design

Bangladesh University of Engineering and Technology
Department of Computer Science and Engineering

Submitted by

Section:B1 Groupe:03

Ahsan Ali 1105083

Al Habib Rahman 1105064

Khondokar Solaiman1105082

Golam Saklain Ratul 1105069

Table of Contents

➢ Introduction:

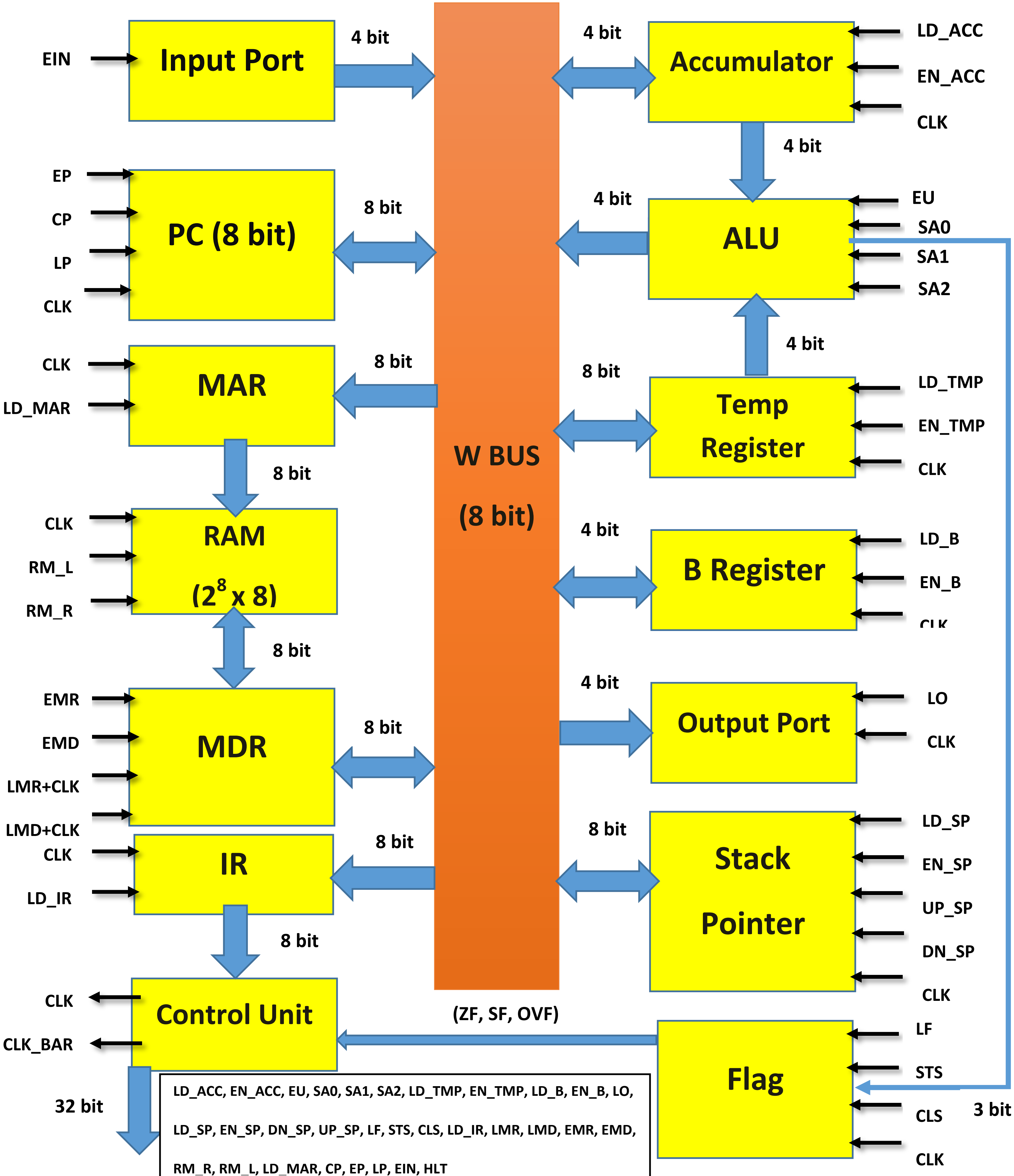
Our 4 bit computer system can execute 28 instructions. Each instruction requires only 3.2875 clock cycles on average. Our Control unit is micro programmed. Both 8 bit address bus and 4 bit data bus share the same 8 bit W-BUS of our 4 bit PC. All memory address length is 8 bit. All components of 4 bit PC share this 8 bit W-BUS for transferring data. All outputs therefore from PC components that are connected to the shared-BUS through tri-state buffer so that one component does not load data from W-BUS while others are using it. RAM contains the instruction data and memory data. RAM also contains the program stack. Our RAM size is 256 X 8 bit. All instructions are either 1 or 2 bytes long. All data values are 4 bit. Although immediate values take out 1 byte of RAM (it is because RAM is byte addressable), the higher nibble contains all zeros, and lower nibble contains the actual 4 bit data. Some instructions require 4~6 clock cycle on average and others (CALL, RET) require 10~12 clock cycles on average. The computer can also communicate with the external devices through the input and output port registers. The instruction set contains all types of instructions which generalize the computer to execute any kind of program. However the 8 bit address bus restricts the program length to be within 256 Bytes. The 4 bit arithmetic and logic unit can perform addition and subtraction and several logical operations. Each instruction has fixed 3 clock cycles for fetching instruction code in control unit. After this, execution cycle starts. We used LSI and digital primitive components of Proteus to simulate our 4 bit PC. Our 4 bit is fully automated, after loading program instruction hex values in ROM of RAM block, PC starts executing instructions if START switch is ON. The main components of our 4 bit PC architecture are-

- Program Counter
- Memory Address Register
- Random Access Memory
- Memory Data Register
- Instruction Register
- Accumulator
- Arithmetic Logic Unit
- Temporary Register
- B Register
- Input Port
- Output Port
- Stack Pointer Register
- Flag Register
- Control Unit

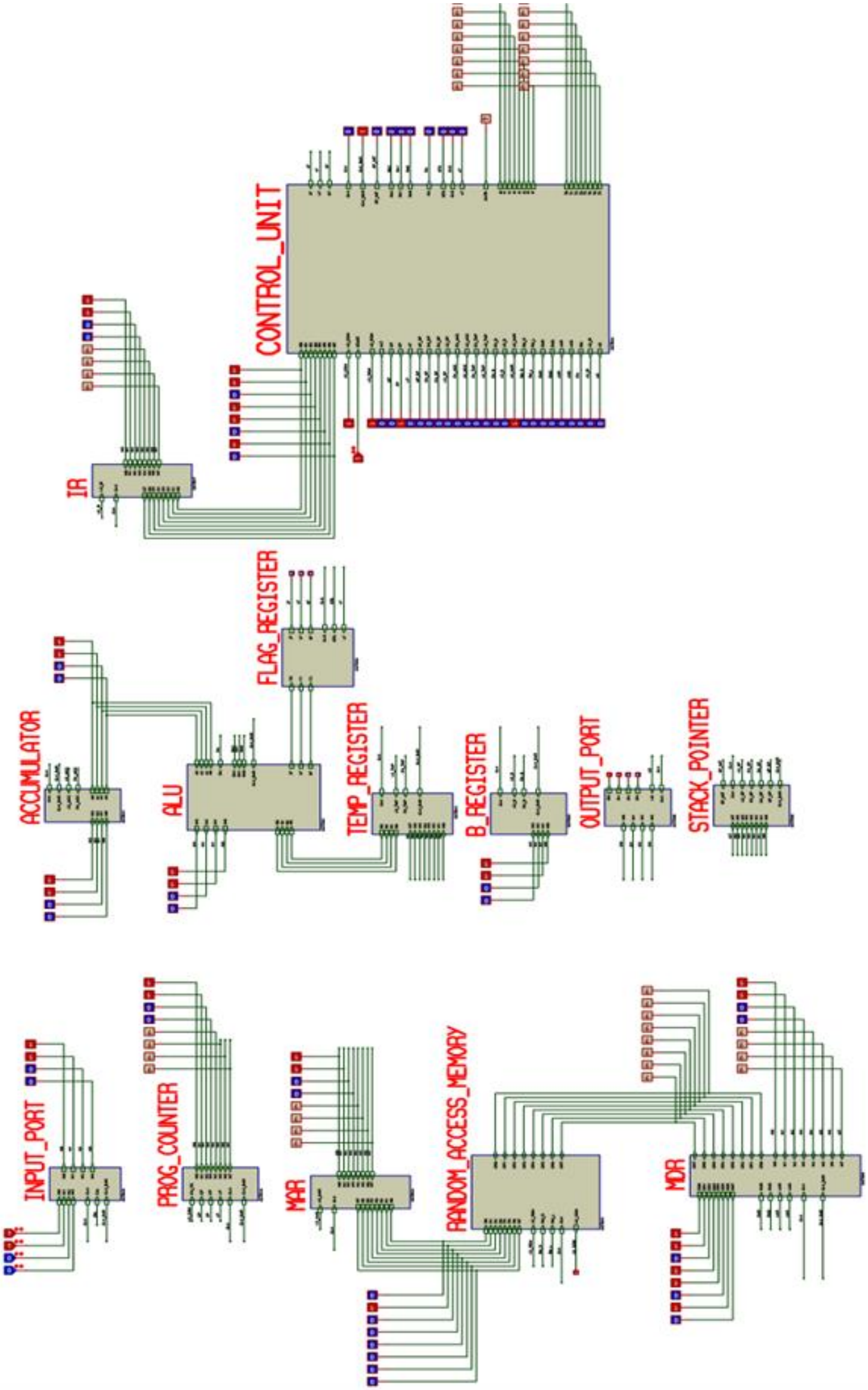
➢ Instruction Set:

No	Instruction	Description	Op-Code(HEX)
1	LDA address;	Acc ← Memory [address]	03
2	STA address;	Memory [address] ← Acc	08
3	MOV Acc, B;	Acc ← B	0D
4	MOV B, Acc;	B ← Acc	0E
5	MOV Acc,immediate;	Acc ← Immediate	0F
6	IN;	Acc ← Input_port	12
7	OUT;	Output_port ← Acc	13
8	ADD B;	Acc ← Acc + B	14
9	ADC B;	Acc ← Acc + B + C (Content of Carry Flag)	16
10	SUB B;	Acc ← Acc – B	18
11	SBB B;	Acc ← Acc – B – Bo(Content of Carry Flag)	1A
12	ADC immediate;	Acc ← Acc + Immediate + C (Content of Carry Flag)	1C
13	SBB immediate;	Acc ← Acc – Immediate – Bo (Content of Carry Flag)	20
14	CMP B;	Accumulator will be unchanged. Set flags according to (Acc – B)	24
15	TEST B;	Accumulator will be unchanged. Set flags according to (Acc . B)	26
16	JNZ address;	Jumps to the address if zero flag is not set	32
17	JL address;	Jump if less	35
18	PUSH;	Pushes the content of Accumulator to the stack	38
19	POP;	Pops off stack to Accumulator	3B
20	CALL address;	Calls a subroutine (at the specified address) unconditionally	3E
21	RET;	Returns from current subroutine to the caller unconditionally	4E
22	JMP;	Jumps unconditionally to the address	5A
23	HLT;	Halts execution	5D
24	NOP;	No Operation	5E
25	STS;	Sets the sign flag	30
26	CLS;	Clears the zero flag	31
27	AND [address];	Acc ← Acc . Memory [address]	28
28	OR B;	Acc ← Acc | B	2E

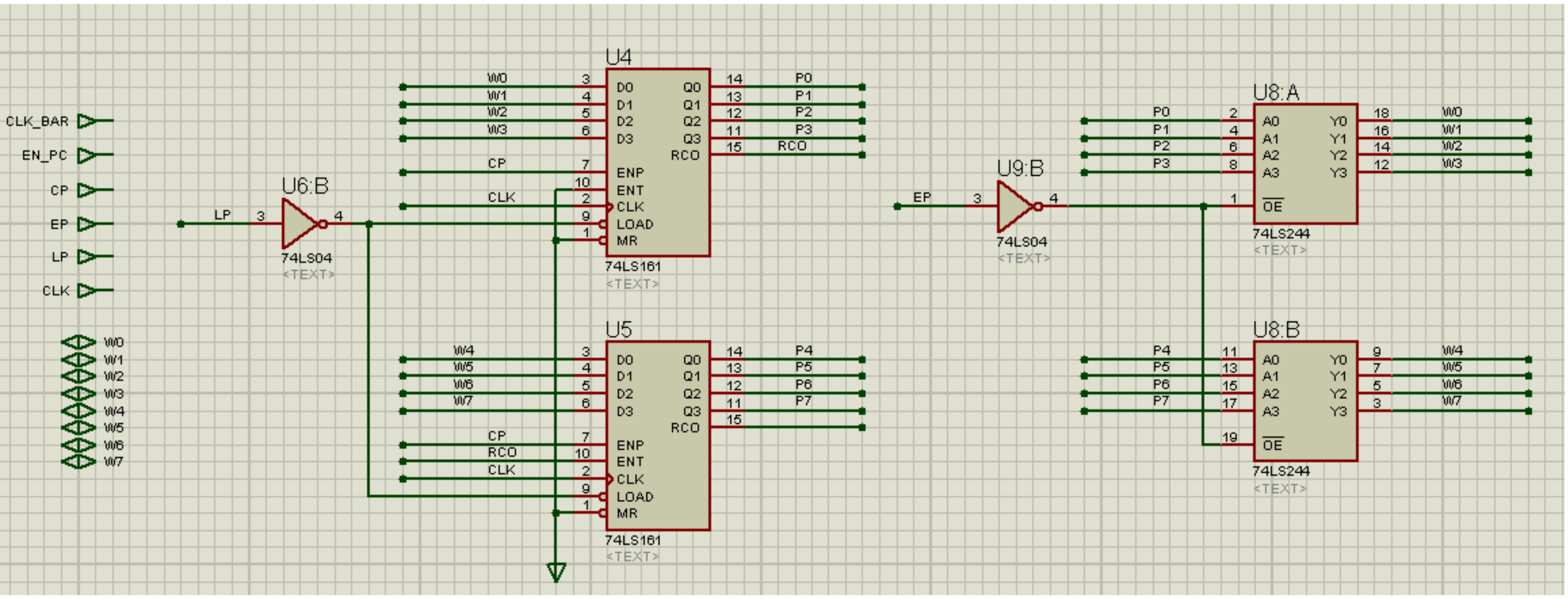
➢ Block Diagram of 4bit PC Architecture:



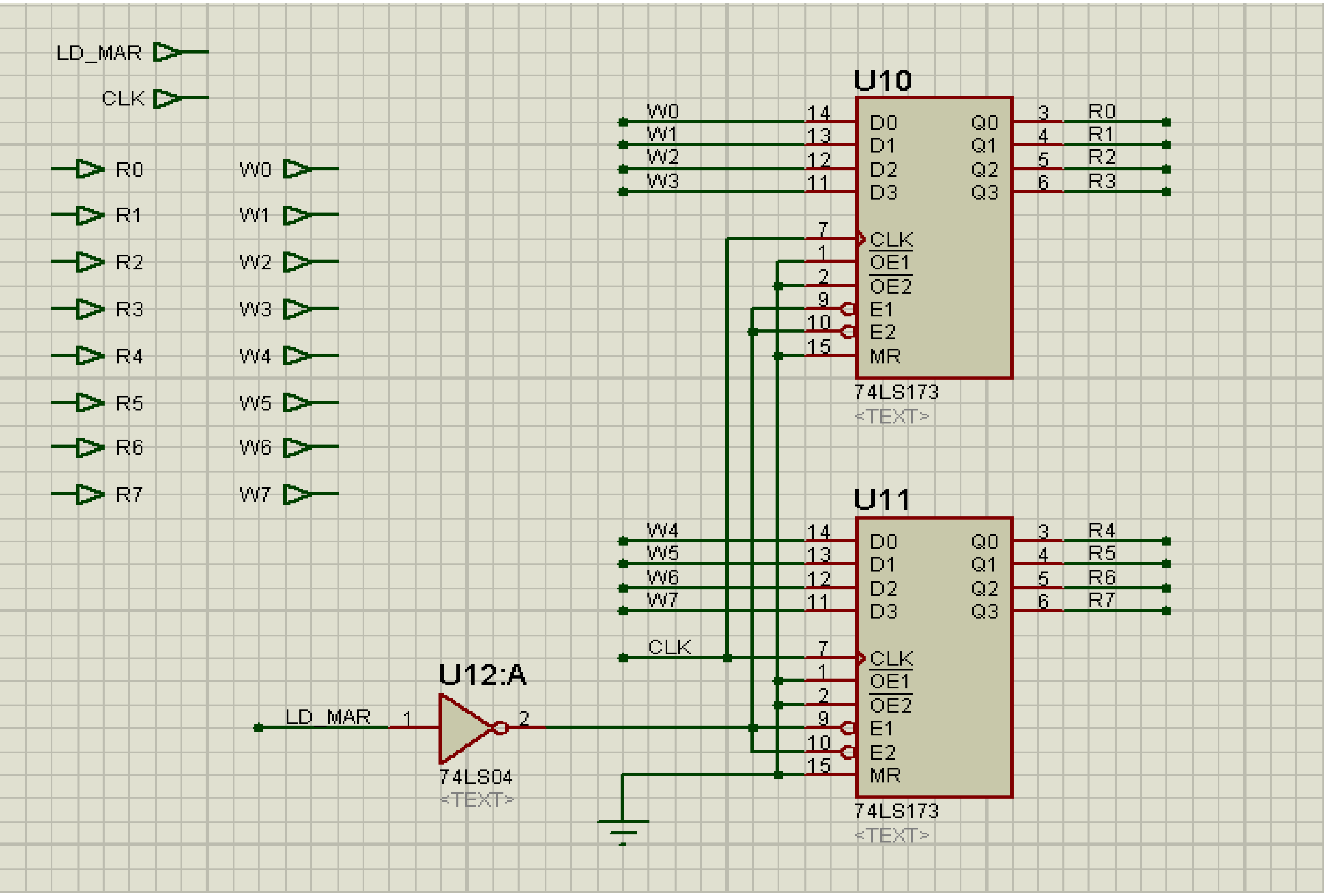
➢ Complete Circuit Diagram of 4bit PC :



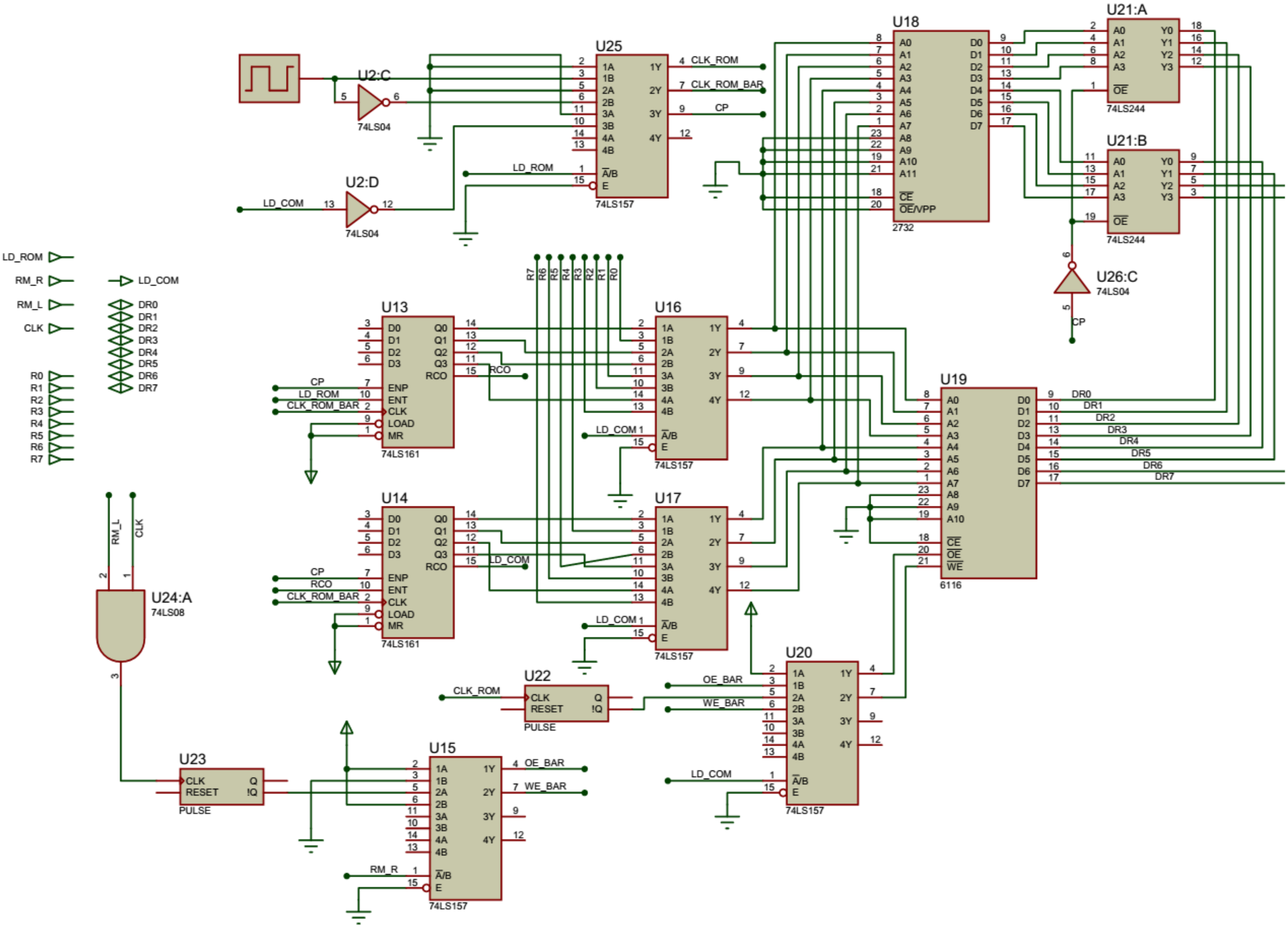
❖ Sub Circuit Diagram of Program Counter :



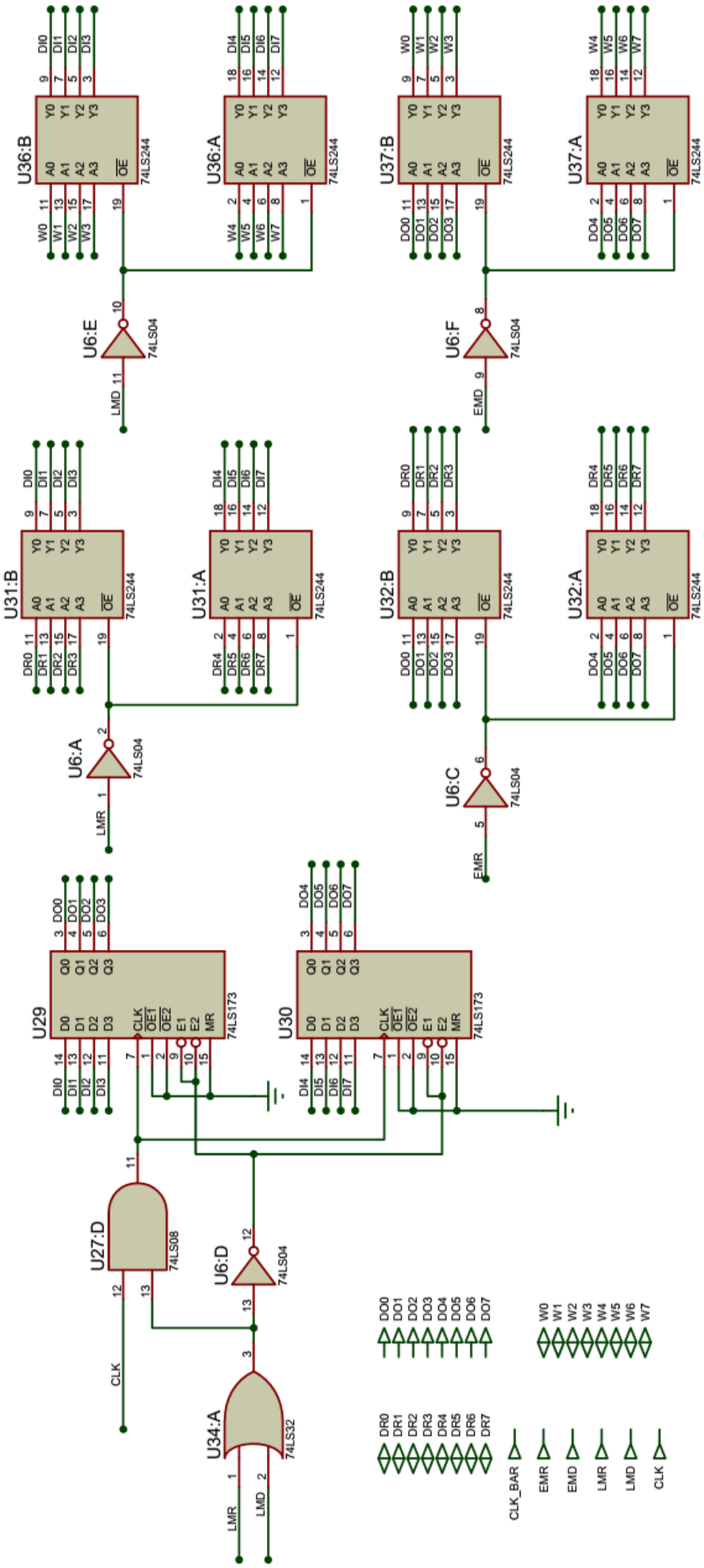
❖ Sub Circuit Diagram of Memory Address Register(MAR) :



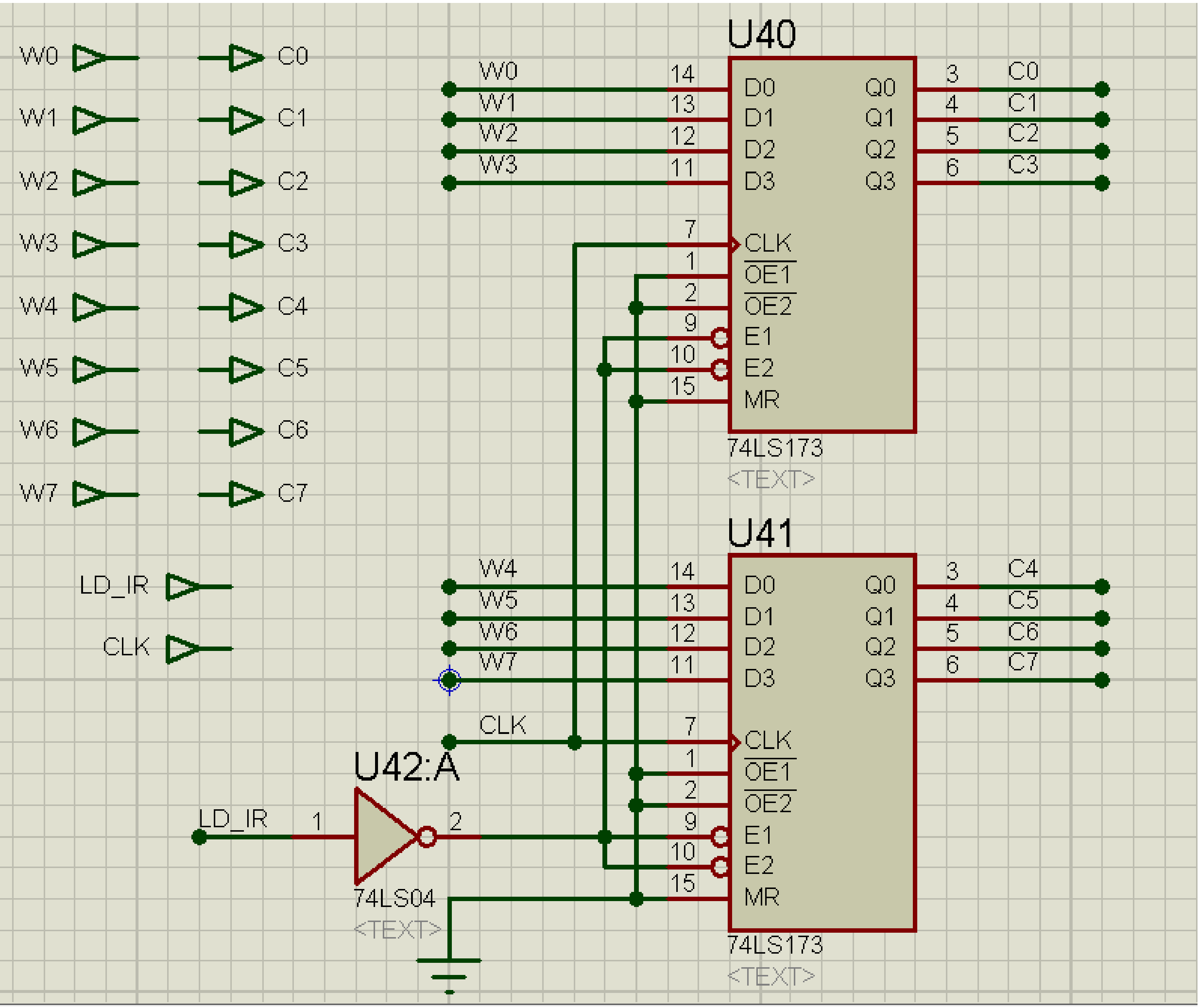
❖Sub Circuit Diagram of Random Access Memory(RAM) :



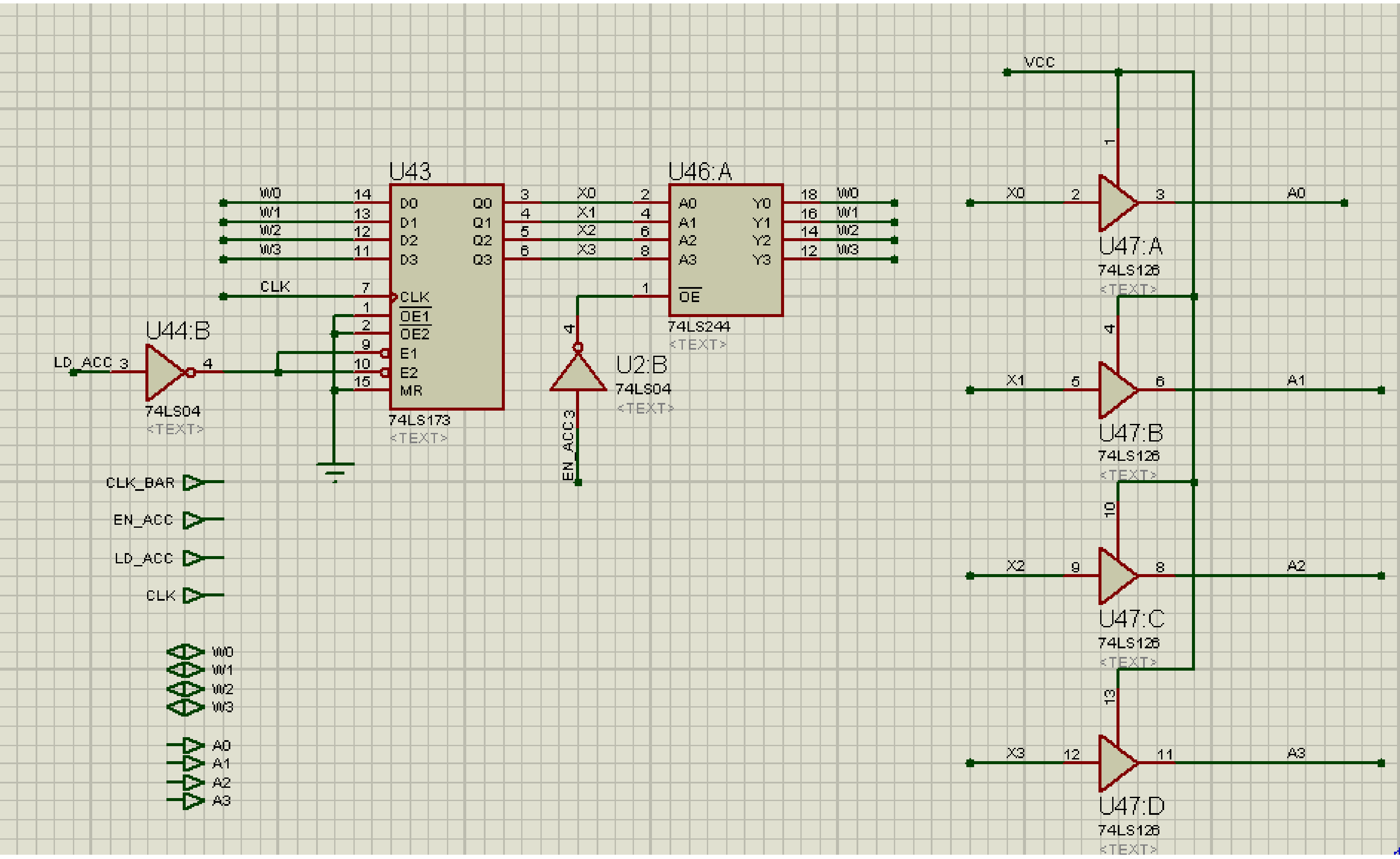
❖ Sub Circuit Diagram of Memory Data Register(MDR) :



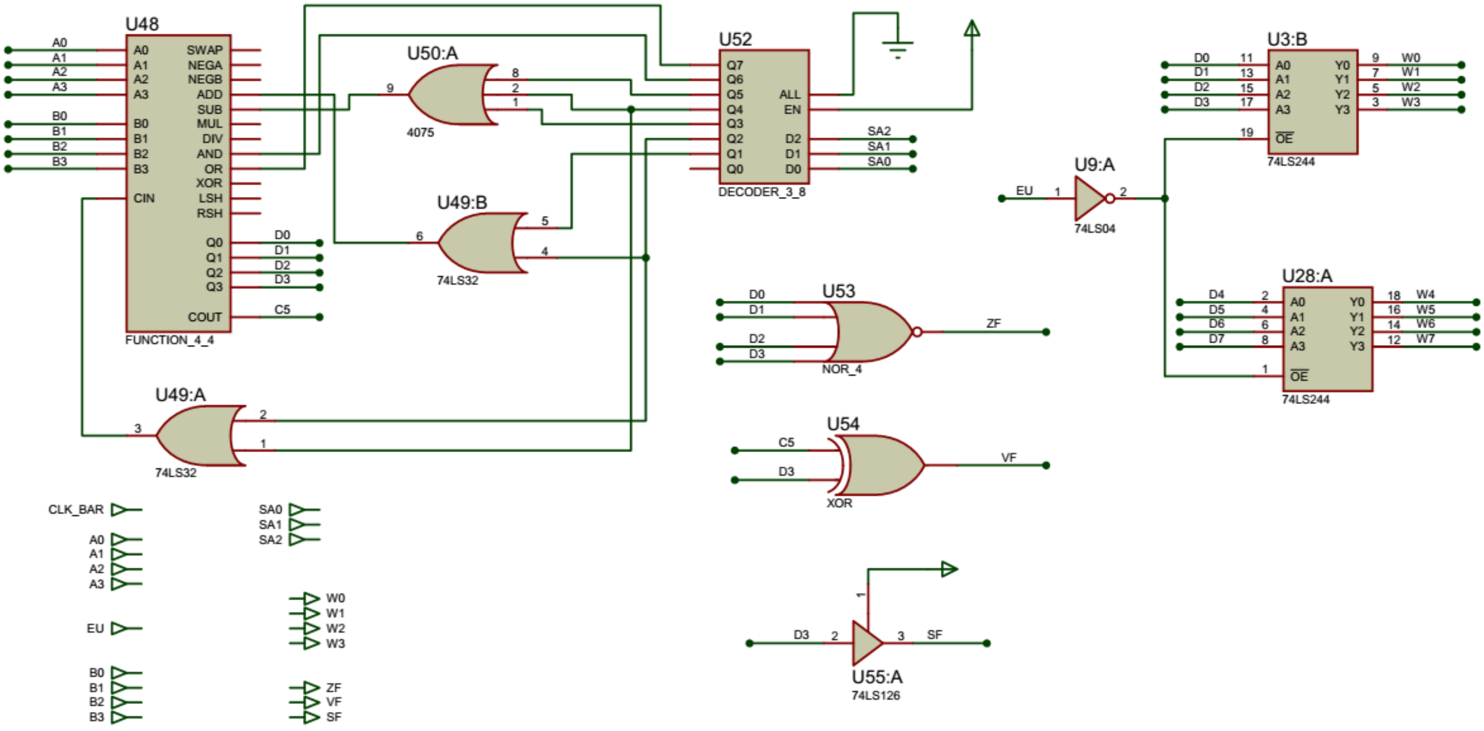
❖ Sub Circuit Diagram of Instruction Register(IR) :



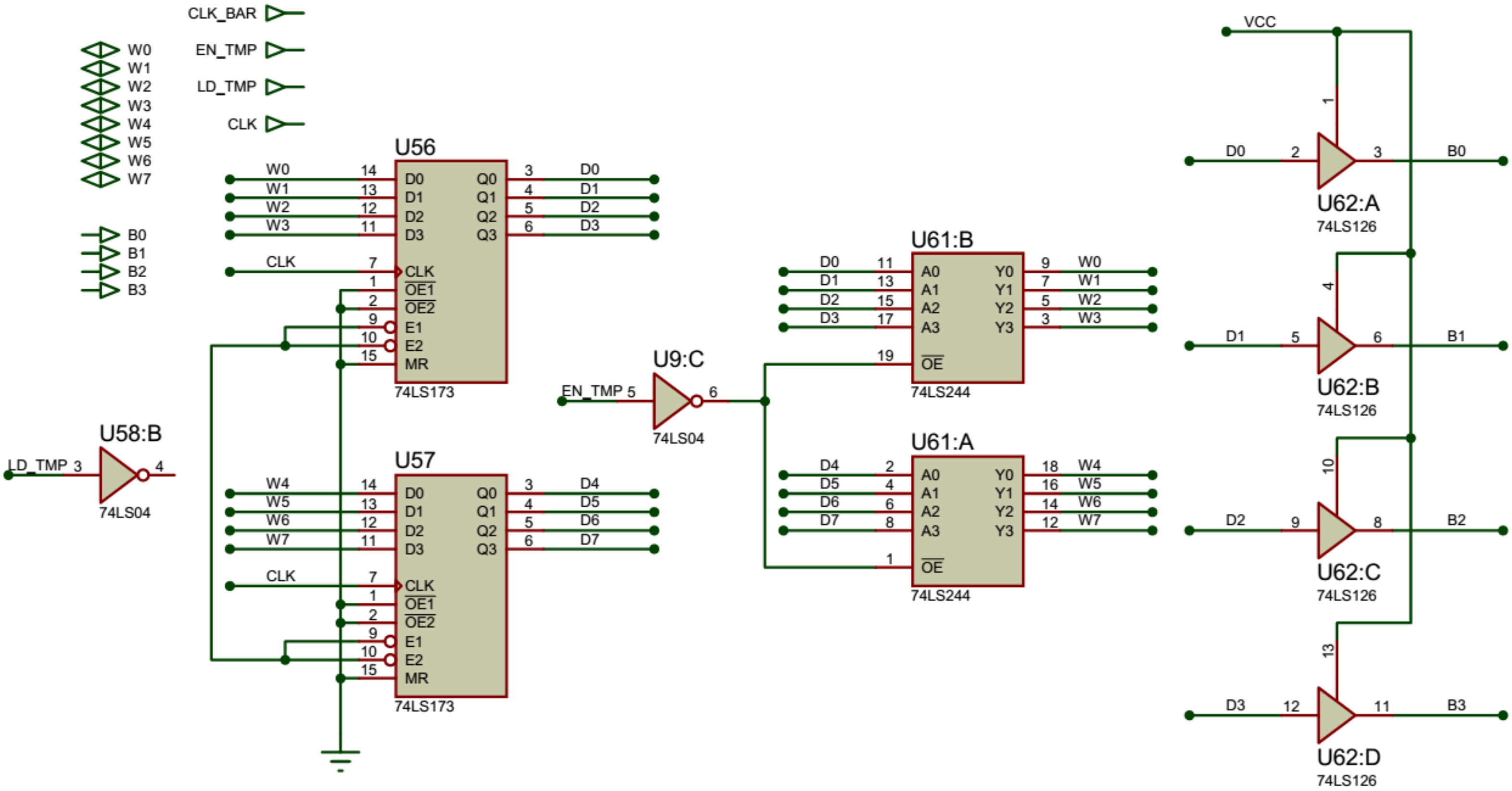
❖ Sub Circuit Diagram of Accumulator :



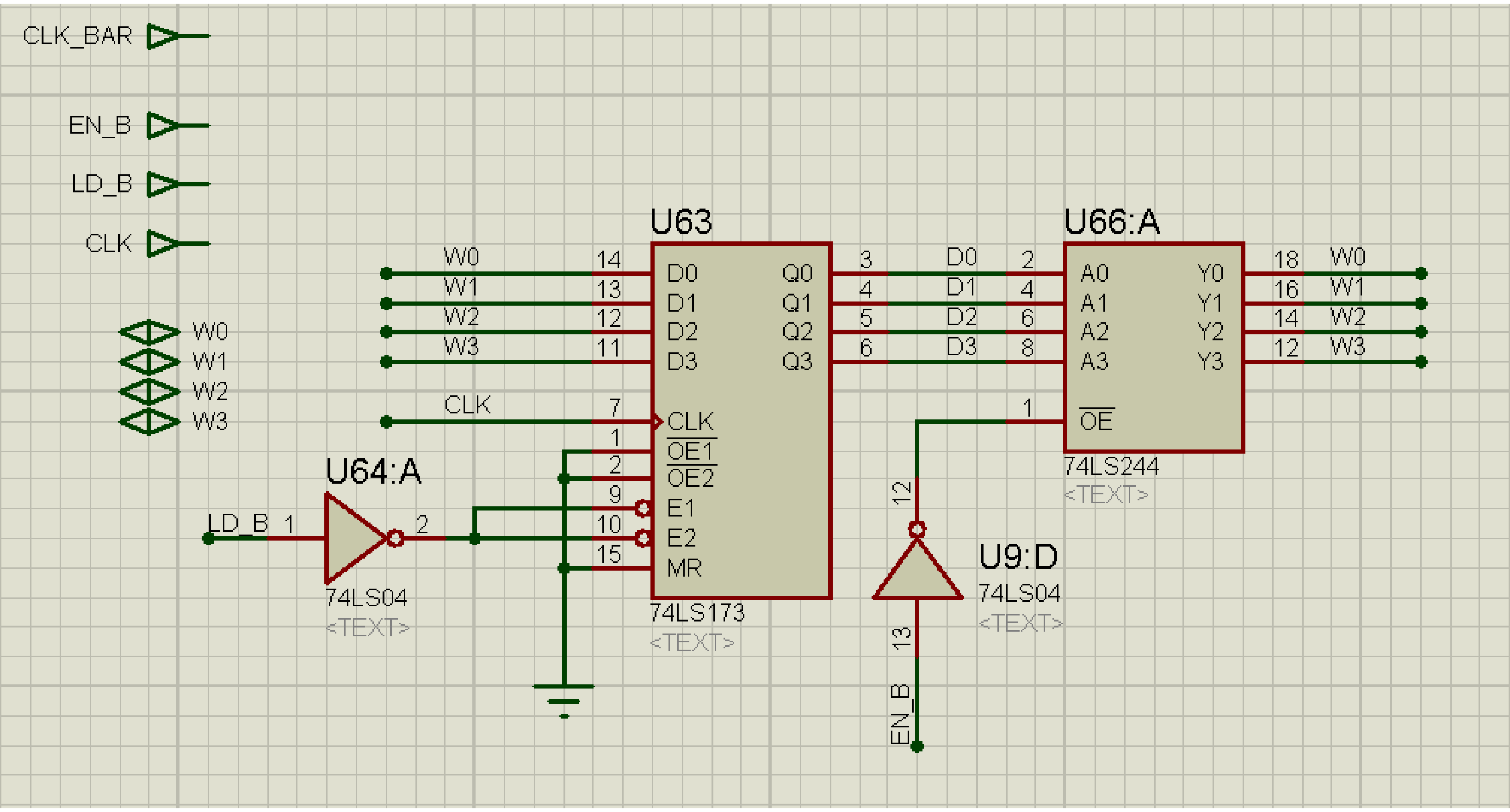
❖ Sub Circuit Diagram of ALU:



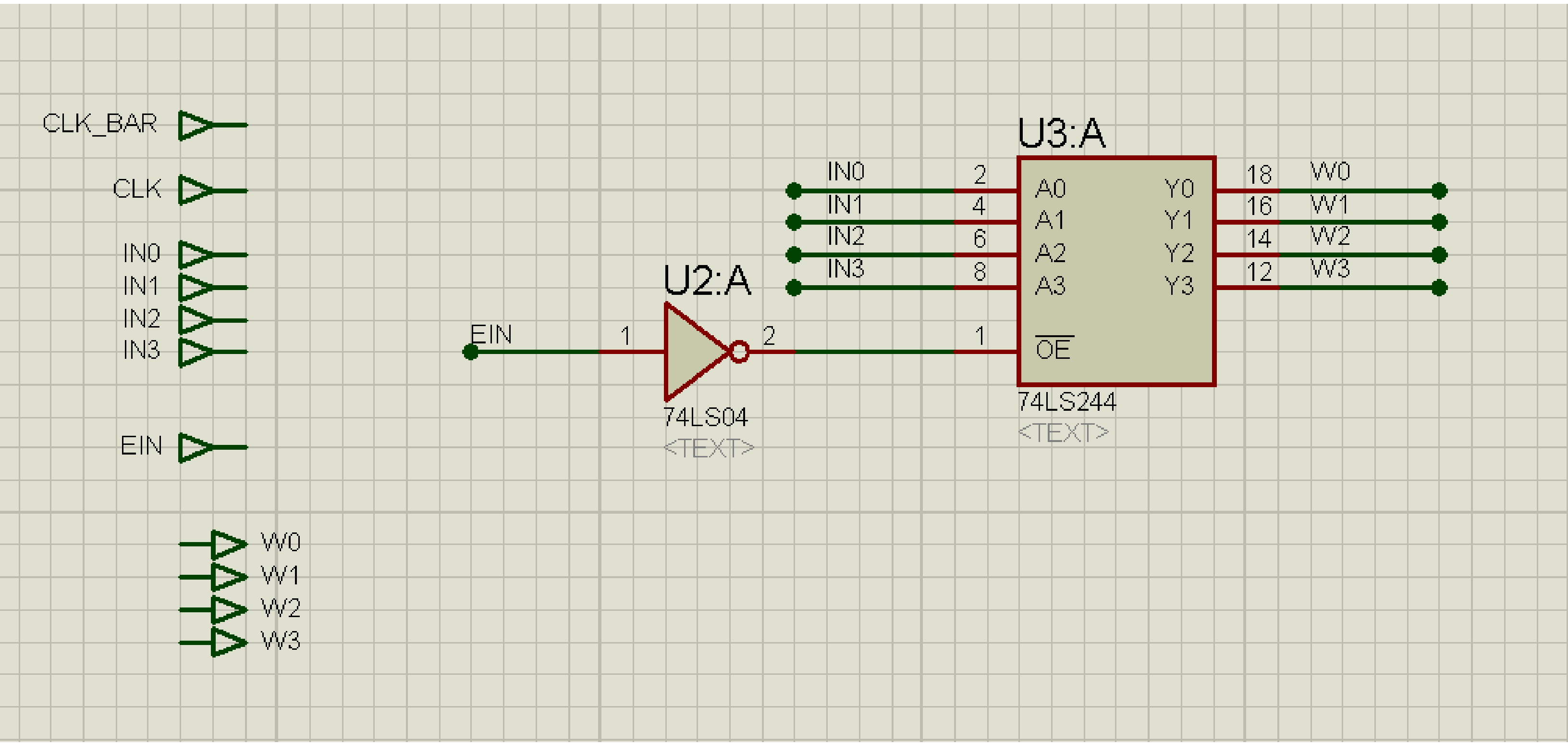
❖ Sub Circuit Diagram of Temporary Register:



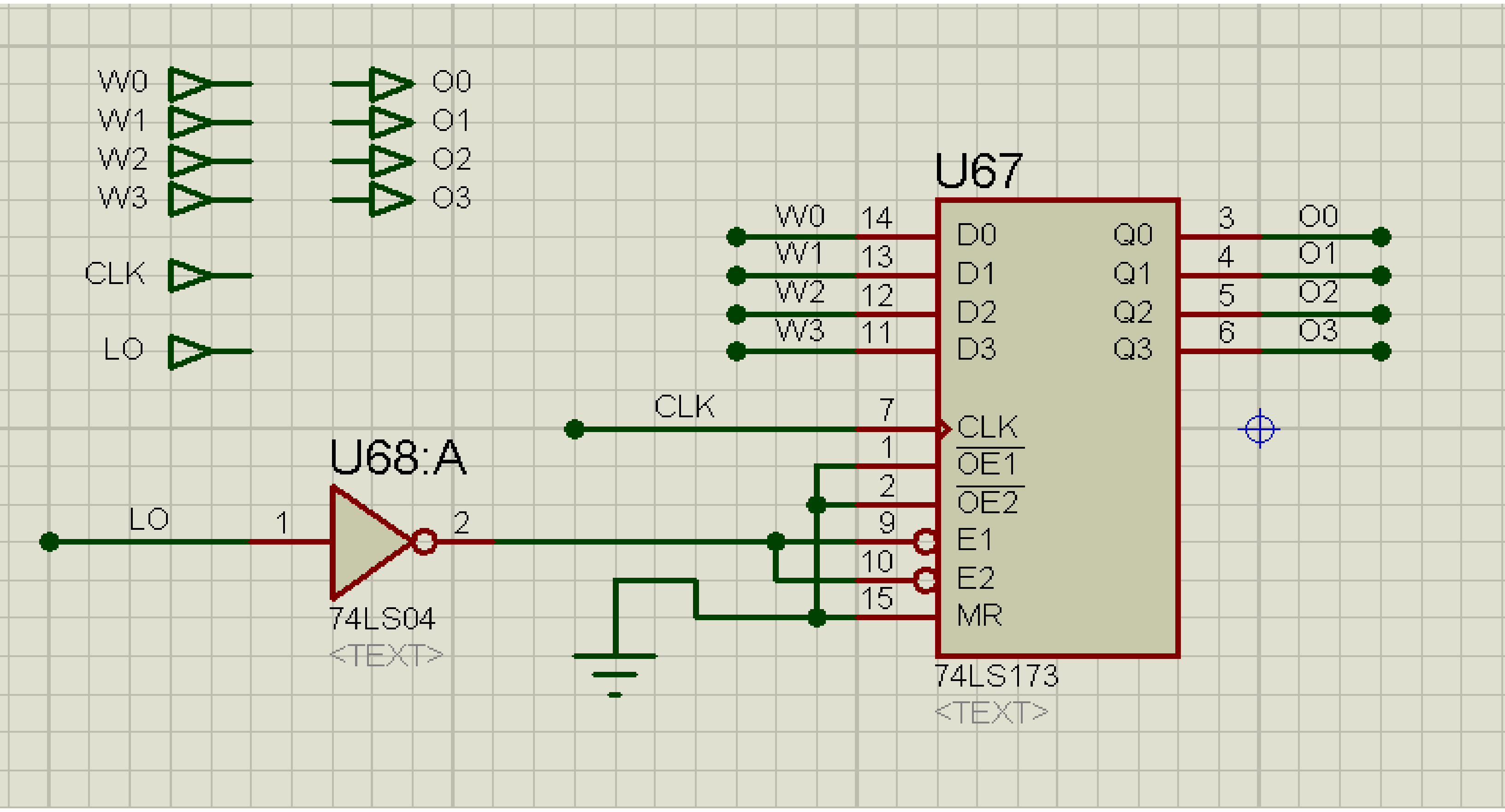
❖ Sub Circuit Diagram of B Register:



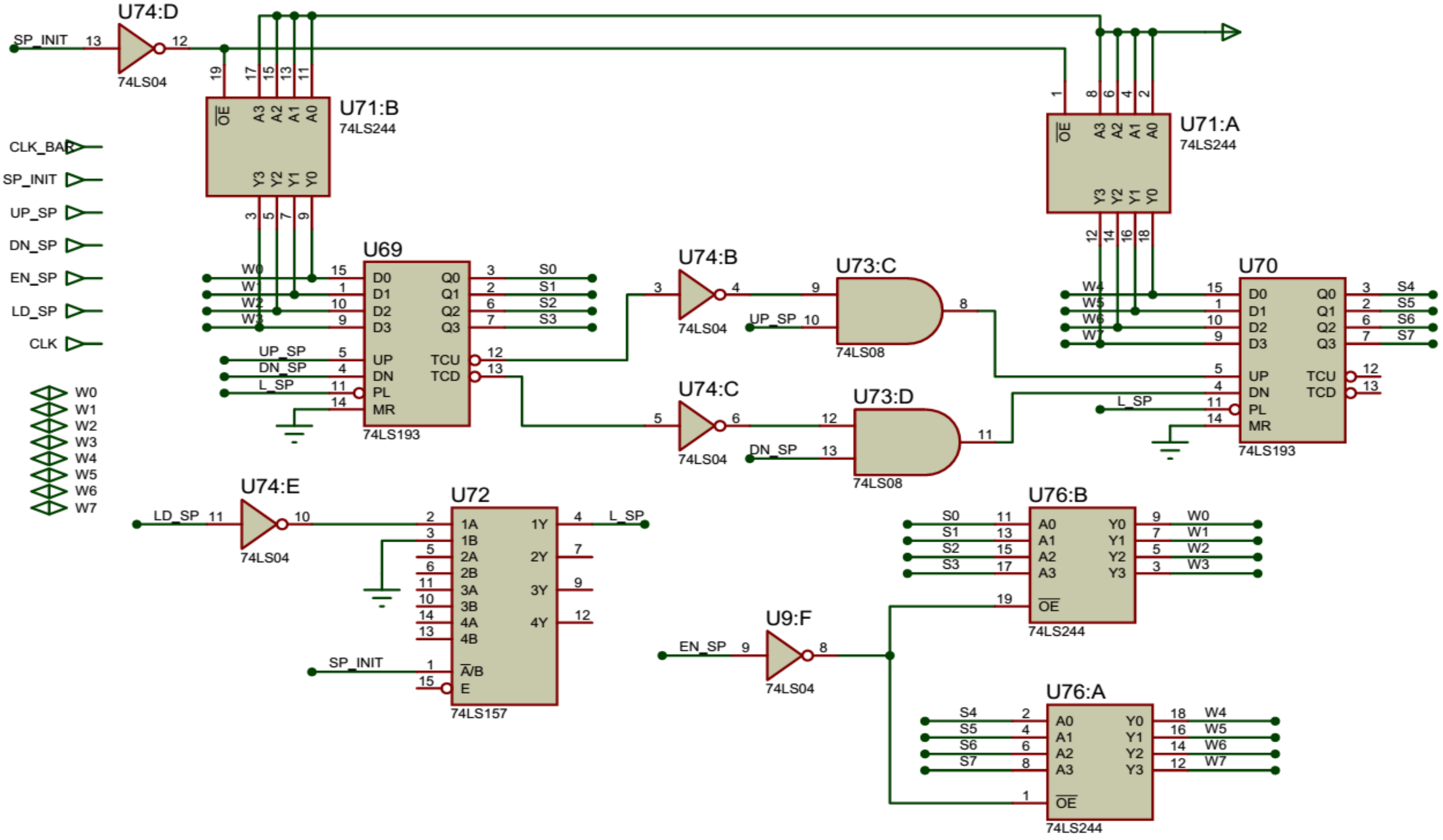
❖ Sub Circuit Diagram of Input Port:



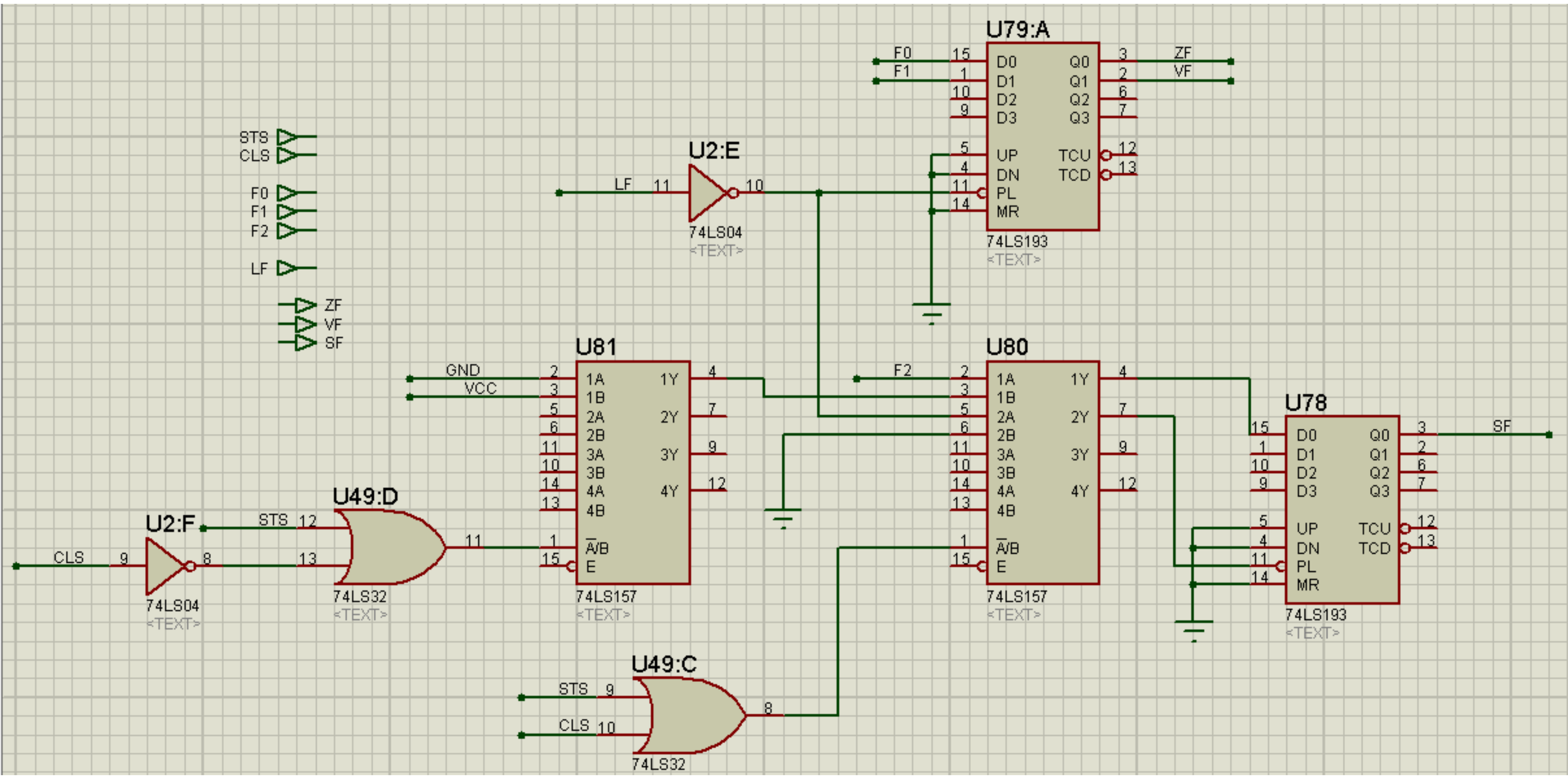
❖ Sub Circuit Diagram of Output Port:



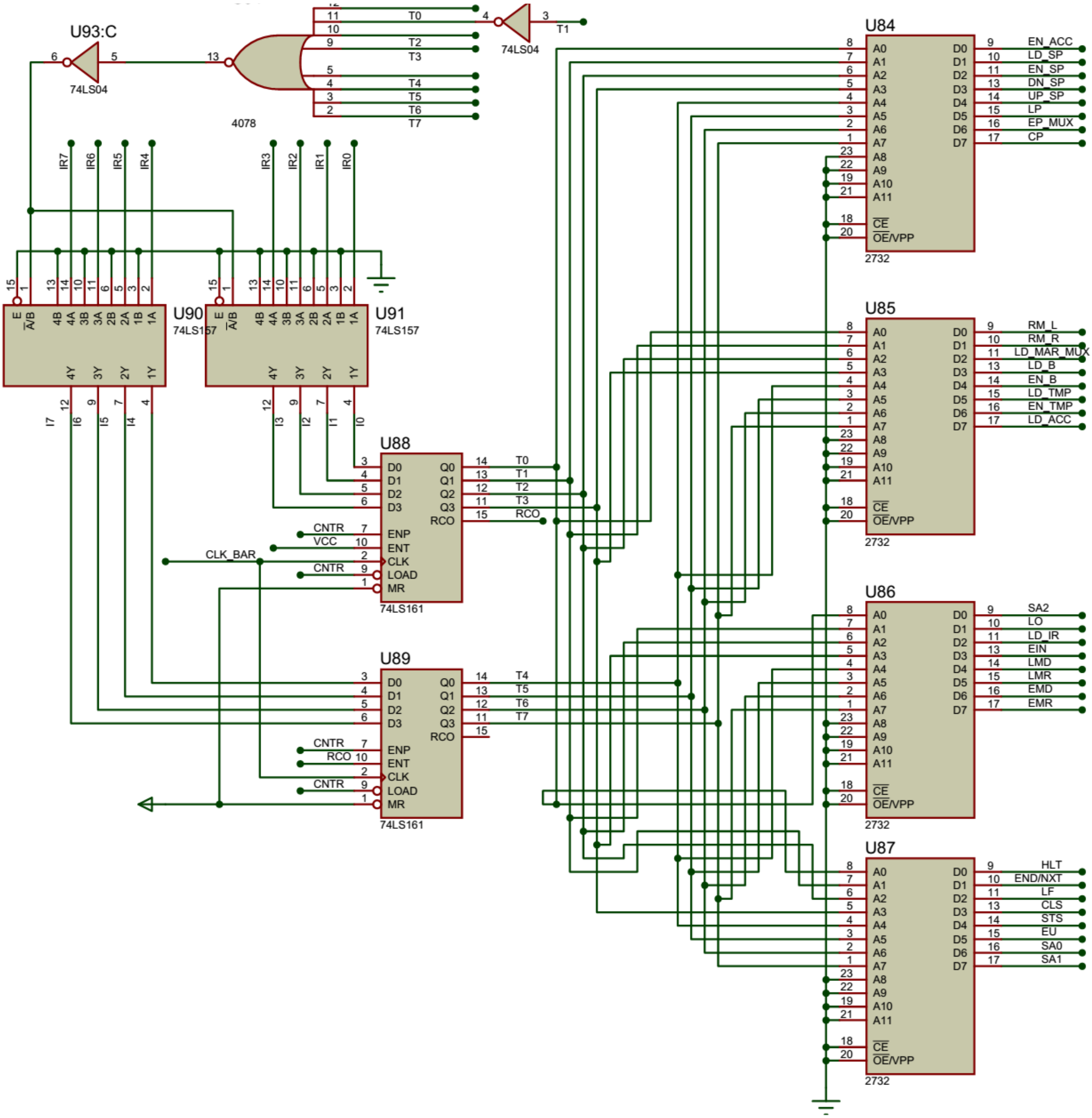
❖ Sub Circuit Diagram of Stack Pointer Register:



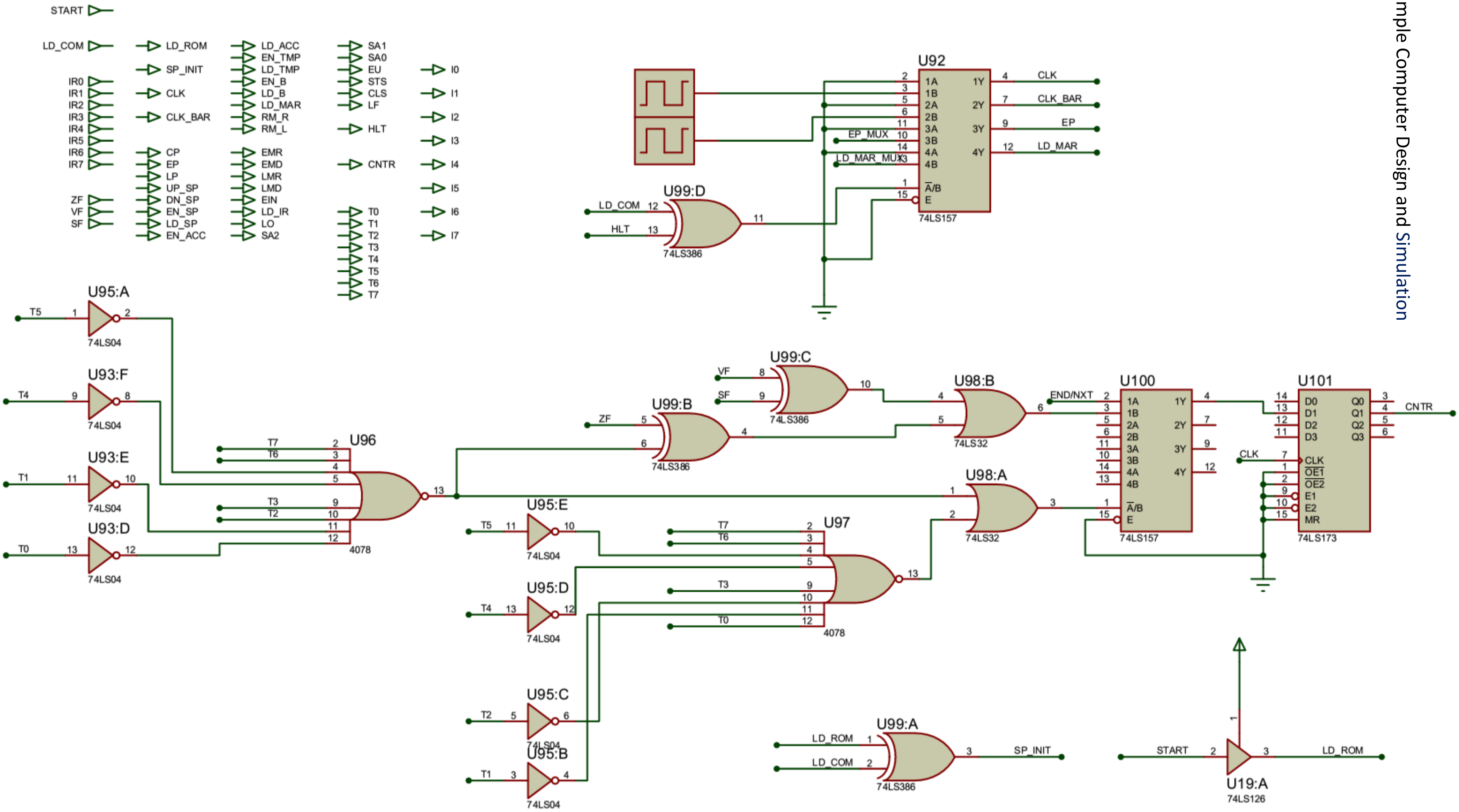
❖ Sub Circuit Diagram of Flag Register:



❖ Sub Circuit Diagram of Control Unit(Part-1):

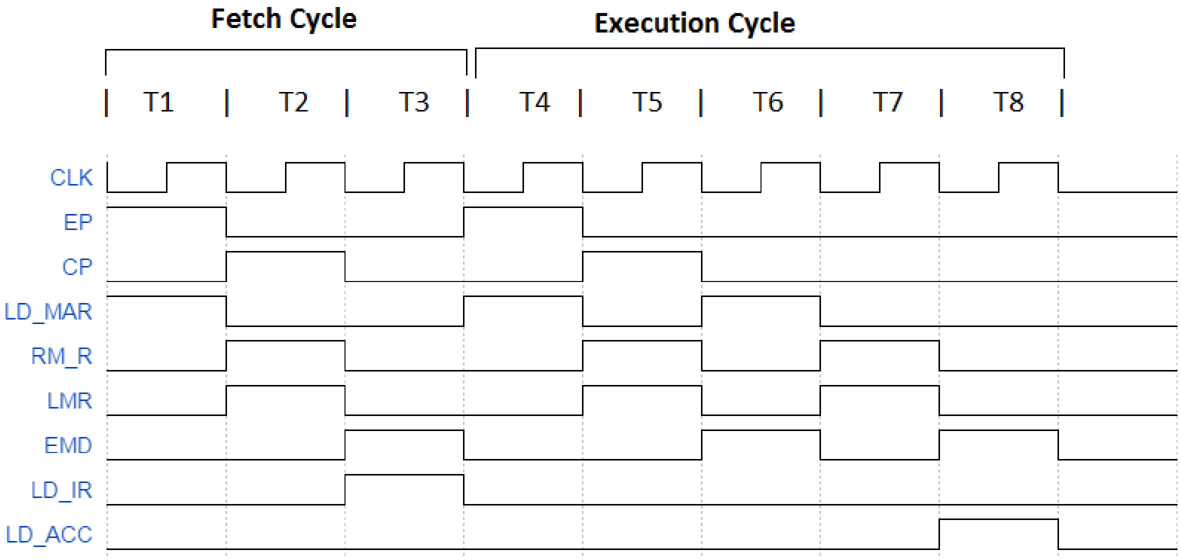


❖ Sub Circuit Diagram of Control Unit(Part-2):

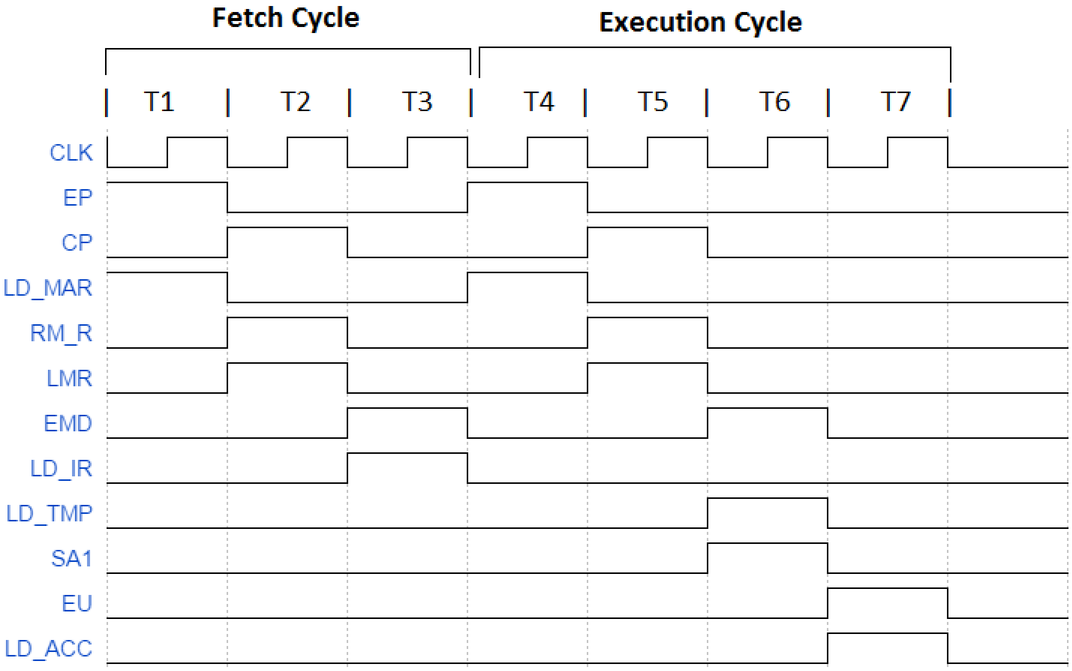


Timing Diagrams:

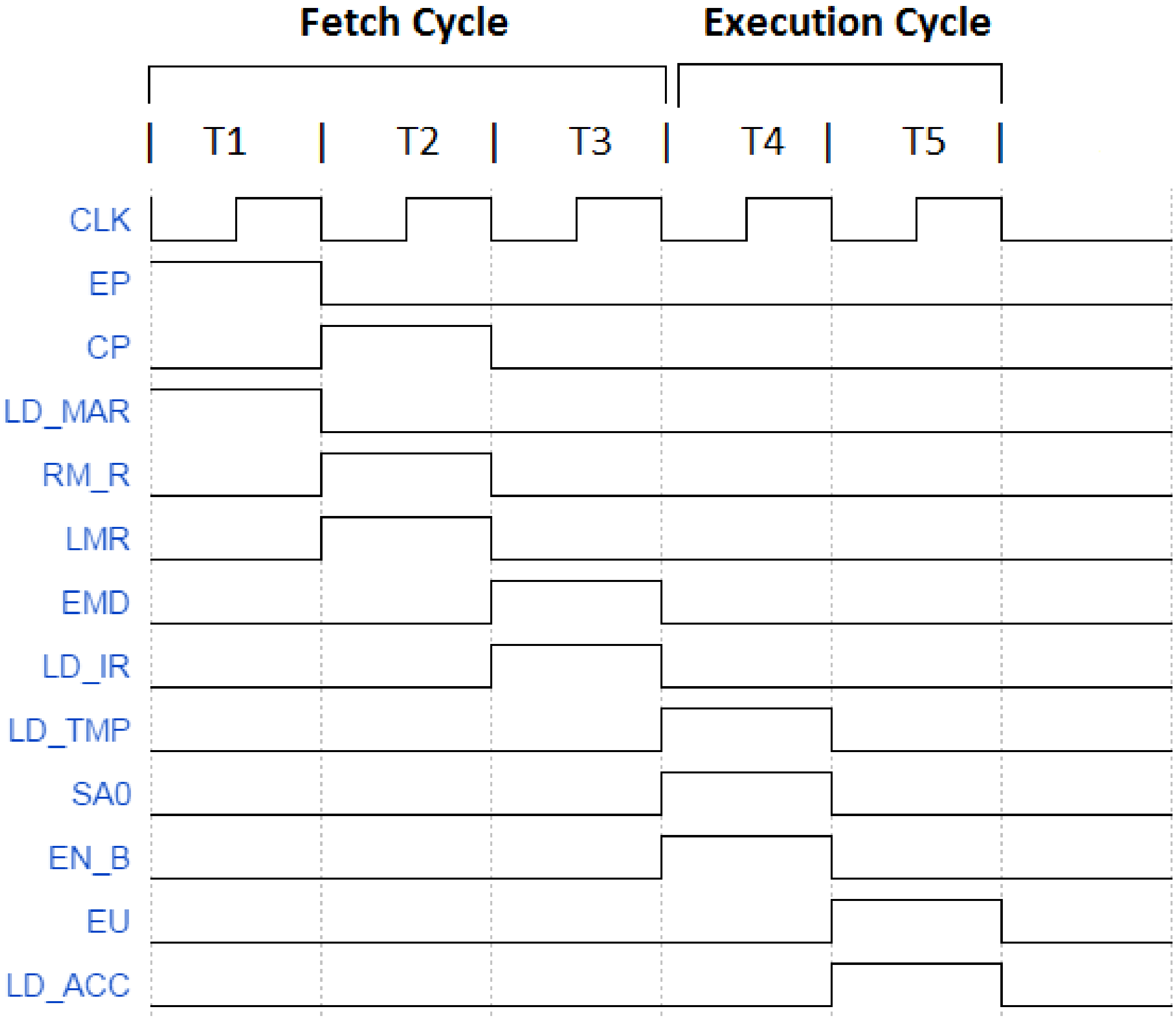
1. LDA[Address]: (8 Clock cycles)



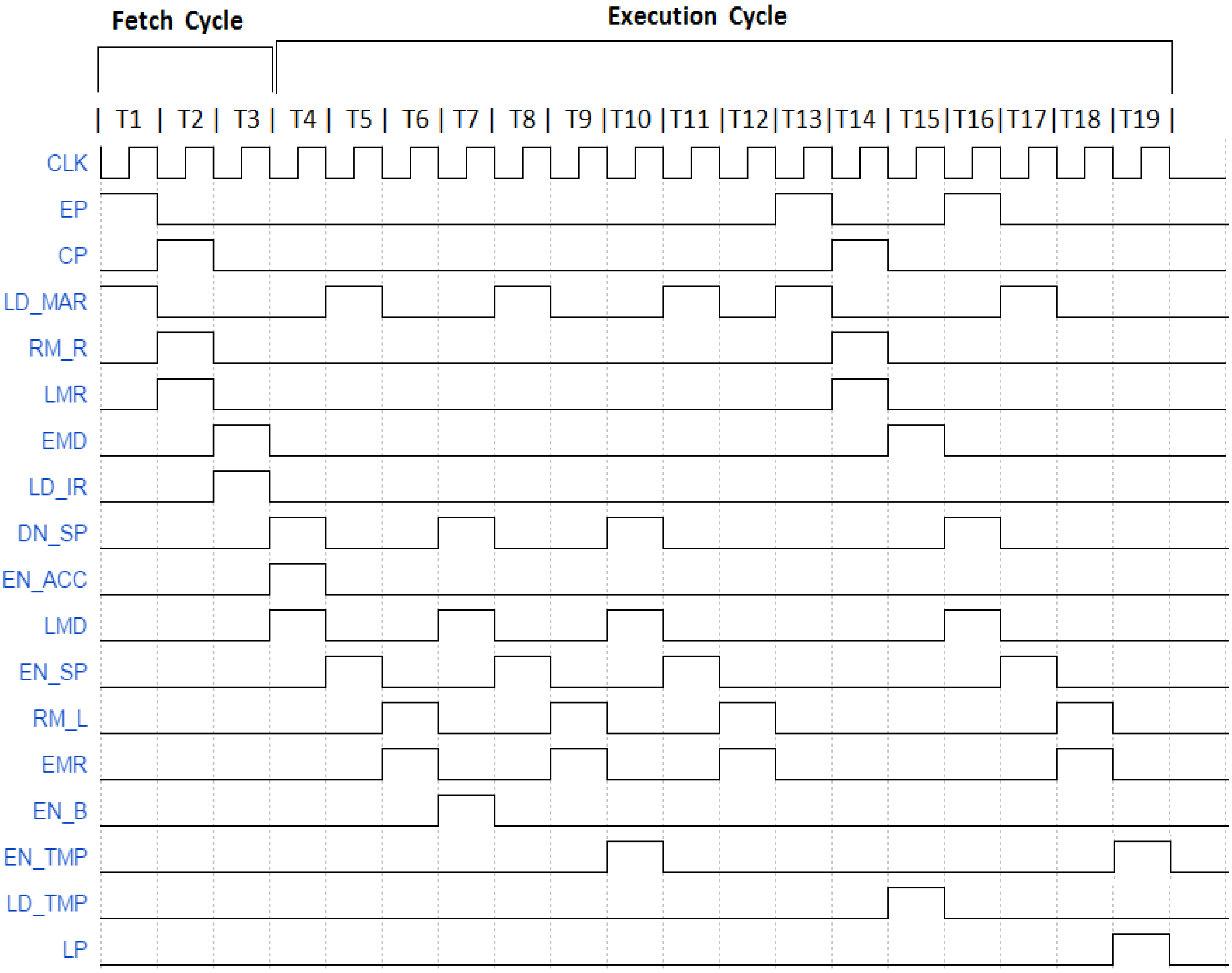
2. ADC Immediate: (7 Clock cycles)



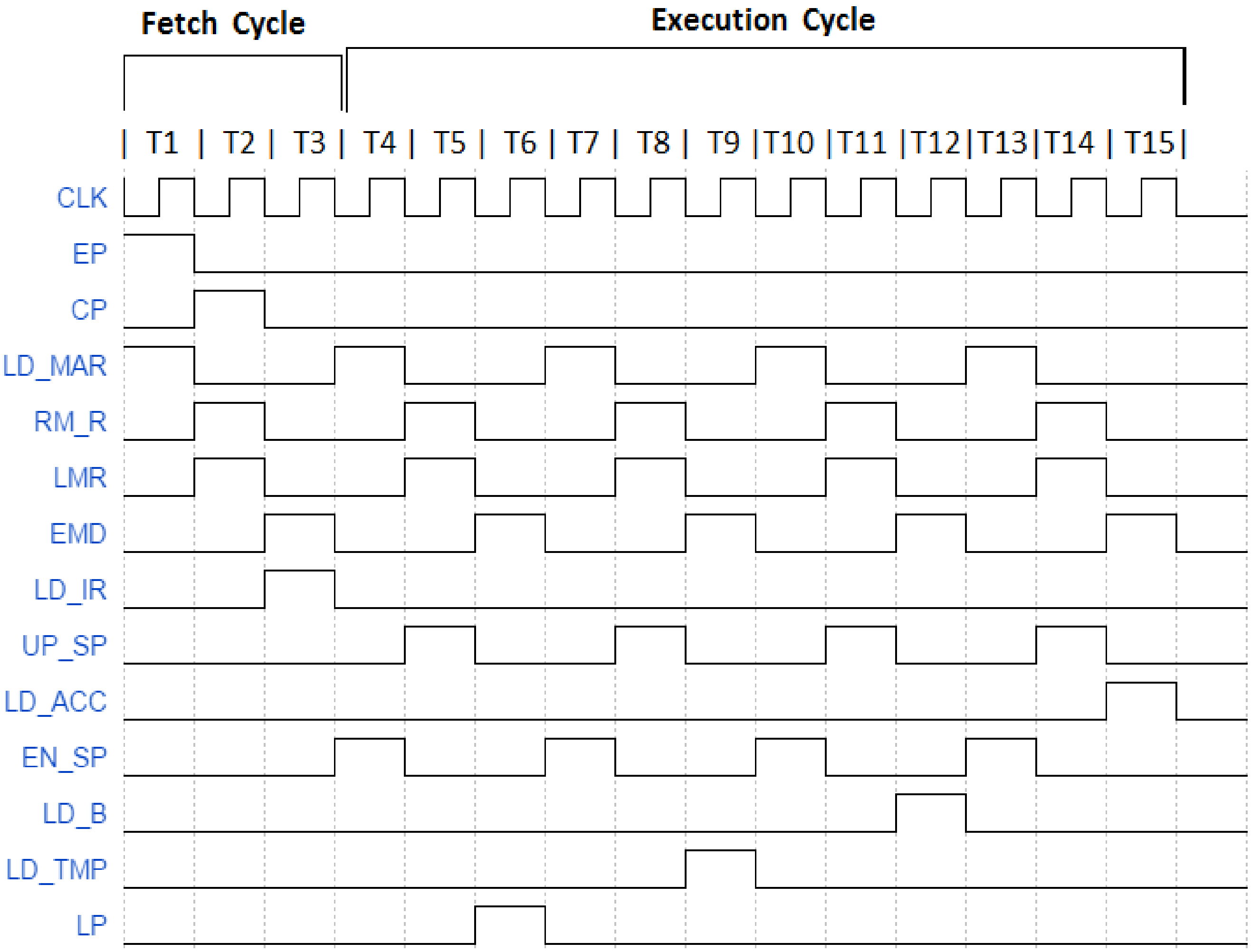
3. ADD B: (5 Clock cycles)



4. CALL [Address]: (19 Clock cycles)

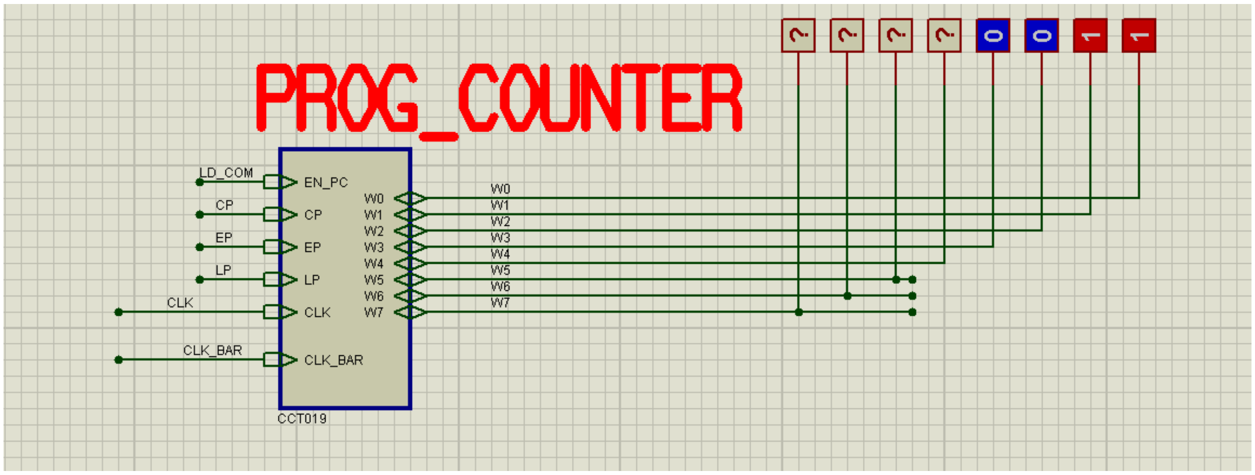


5. RET: (15 Clock cycles)



➢ Explanation of all Blocks:

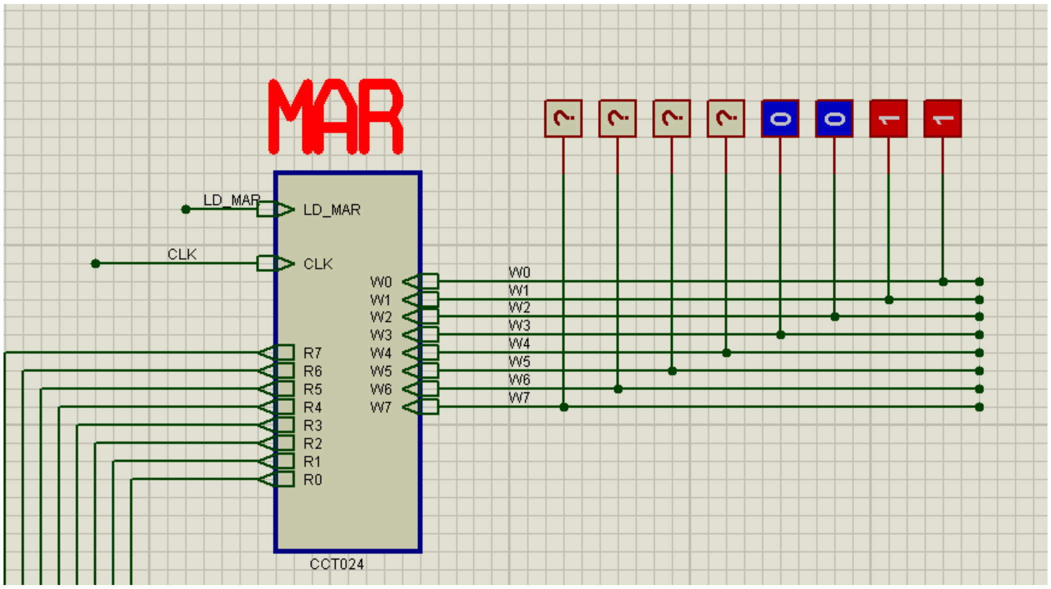
1. Description of Program Counter Register Block:



The Program counter register block contains the address of the next instruction to be executed. It has 6 input pins EN_PC, CP, EP, LP, CLK, CLK_BAR and 8 bidirectional pins W0-to-W7 (address bus). When ROM is loaded to RAM, then EN_PC pin is HIGH which enables program counter. Program counter is incremented at CLK signal when CP pin is HIGH. Program counter output goes to 8 bit address bus (W-BUS) when EP pin is HIGH. Program counter loads JMP, CALL, JNZ, JL, RET addresses from 8 bit address bus at CLK signal when LP pin is HIGH. Program counter's 8bit address (output) is connected to 8 bit address bus (W0-W7) with 74LS244 buffer. Program counter is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS161 (Synchronous 4 bit binary counter)	2
74LS04 (NOT GATE)	2 Gates
74LS244 (Octal Buffer with tri-state output)	1

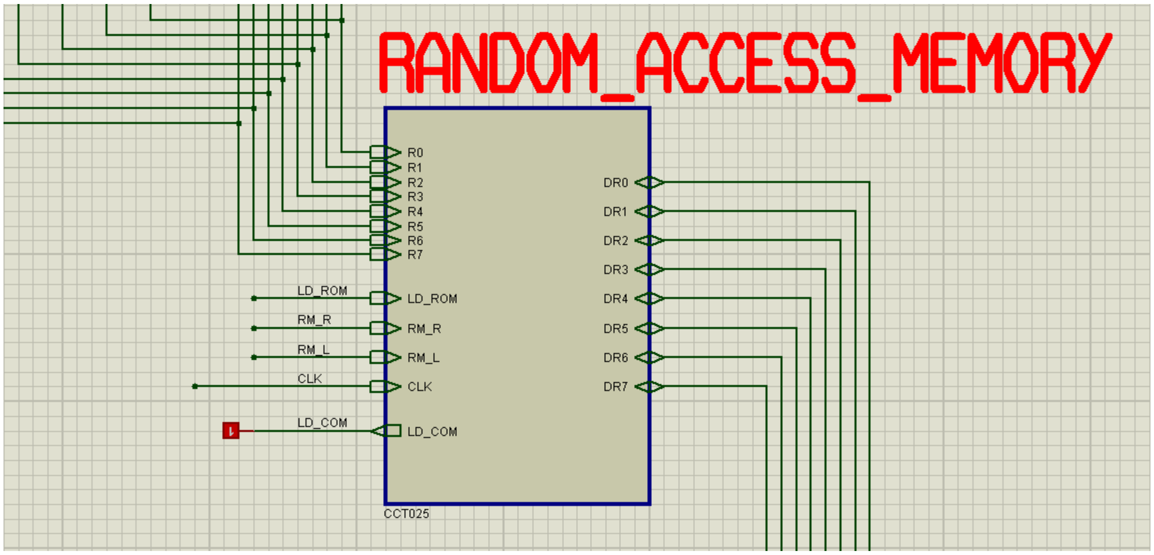
2. Description of Memory Address Register (MAR) Block:



The Memory address register (MAR) block stores the 8 bit address of instruction or data which are placed in RAM. It has 10 input pins LD_MAR, CLK, W0-to-W7 (address bus 8 pins) and 8 output pins R0-to-R7 which are fed into RAM input. Memory address register loads 8 bit address W0-W7 from W-BUS and passes it to output R0-R7 at CLK signal when LD_MAR is HIGH. MAR is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS173 (Quad D-type Flip-Flop with tri-state output)	2
74LS04 (NOT GATE)	2 Gates

3. Description of Random Access Memory (RAM) Block:



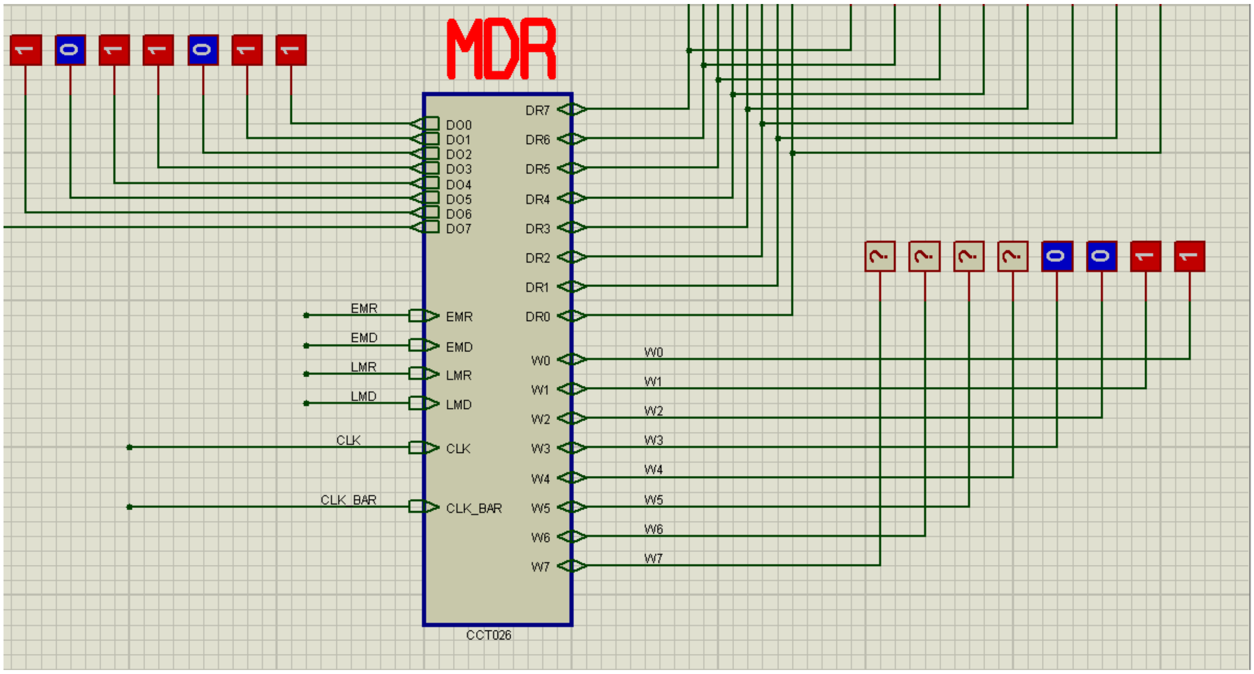
The Random access memory block contains the program instructions and data. We also used it as Stack in our 4 bit PC. It has 12 input pins LD_ROM, RM_R, RM_L, CLK, 8 bit address R0-to-R7 pins and 8 bidirectional pins DR0-to-DR7 which are fed into Memory data register and 1 output pin LD_COM which is fed into Control Unit. RAM block contains RAM and ROM (contains executable instructions). RAM has 8 address pin and 8 output pins, so RAM size is 256X8 bit. When LD_ROM pin is HIGH instructions are loaded from ROM to RAM and after load is complete LD_COM pin outputs HIGH. Input R0-R7 is the address of RAM and DR0-DR7 is the output/input data in case of read/write.

RAM mode	RM_R	RM_L
HALT	0	0
READ	0	1
WRITE	1	X

RAM is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS161 (Synchronous 4 bit binary counter)	2
74LS04 (NOT GATE)	3 Gates
74LS244 (Octal Buffer with tri-state output)	1
74LS157 (Quadruple 1-of-2 Data Multiplexers)	5
74LS08 (AND GATE)	1 Gate
PULSE (Monostable pulse generator-Digital)	2
2732 (4K X 8 EPROM Memory)	1
6116 (2K X 8 Static RAM Memory)	1
CLOCK (Square wave source-Digital)	1

4. Description of Memory Data Register (MDR) Block:



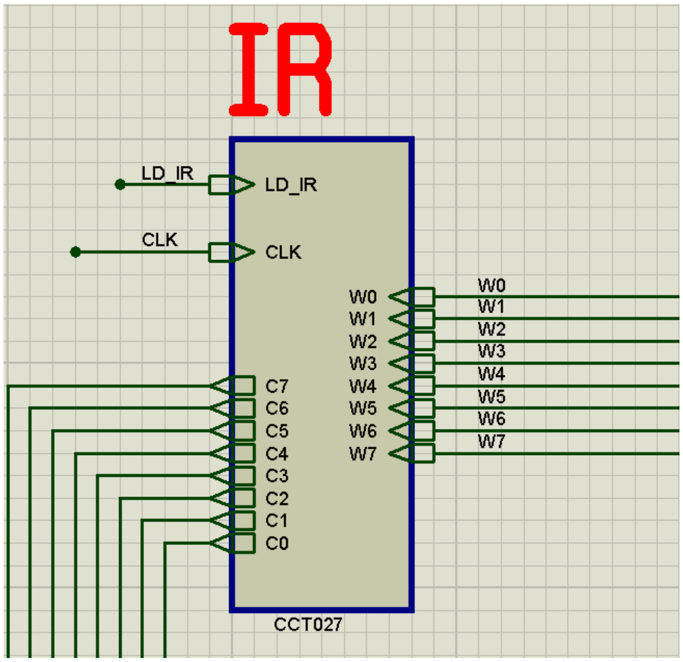
The Memory data register block stores the data that is received from RAM and pass the data to 8 bit address bus (W-BUS) in case of RAM-READ operation and also stores data that is passed to RAM coming from W-BUS in case of RAM-WRITE operation. It has 6 input pins EMR, EMD, LMR, LMD, CLK, CLK_BAR and 16 bidirectional pins where 8 pins DR0-to-DR7 connected to RAM and 8 pins W0-toW7 connected to W-BUS. DR0-DR7 and W0-W7 pins are connected to MDR with 74LS244 buffer. Its Operations are-

MDR Operation	EMR,EMD,LMR,LMD	CLK
Load data from W-BUS to MDR	LMD = HIGH	HIGH
Load data from RAM to MDR	LMR = HIGH	HIGH
Release data from MDR to W-BUS	EMD = HIGH	X
Release data from MDR to RAM	EMR = HIGH	X

MDR is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS173 (Quad D-type Flip-Flop with tri-state output)	2
74LS04 (NOT GATE)	5 Gates
74LS244 (Octal Buffer with tri-state output)	4
74LS08 (AND GATE)	1 Gate
74LS32 (OR GATE)	1 Gate

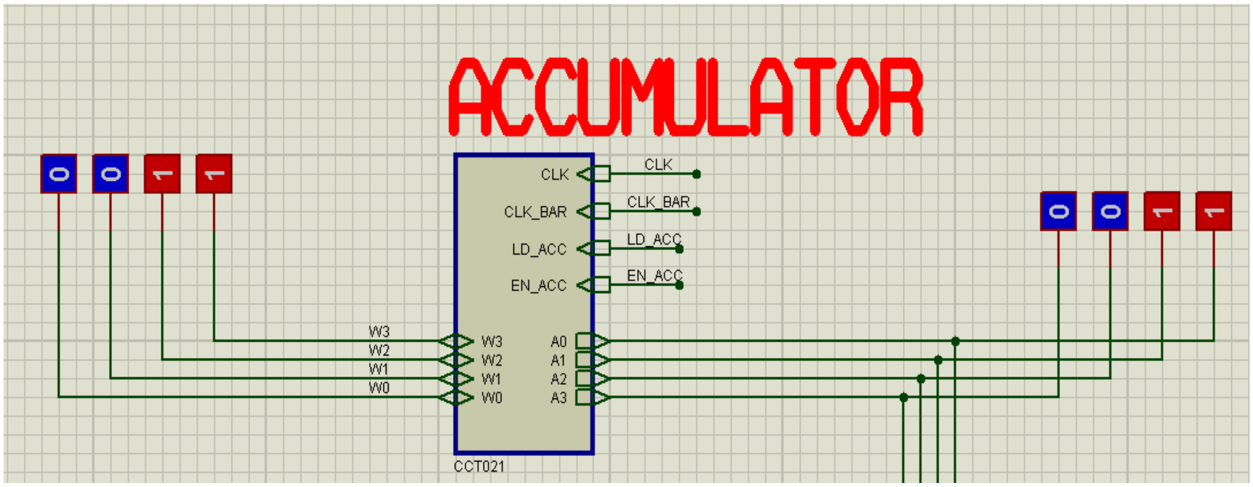
5. Description of Instruction Register (IR) Block:



The Instruction register block stores the 8 bit instruction from W-BUS when it is released from RAM. It has 10 input pins LD_IR, CLK, 8 pins W0-to-W7 (W-BUS) and 8 output pins C0-to-C7 are fed into Control Unit. It loads instruction from W-BUS at CLK signal when LD_IR pin is HIGH and it passes instruction to Control Unit through C0-C7. IR is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS173 (Quad D-type Flip-Flop with tri-state output)	2
74LS04 (NOT GATE)	1 Gate

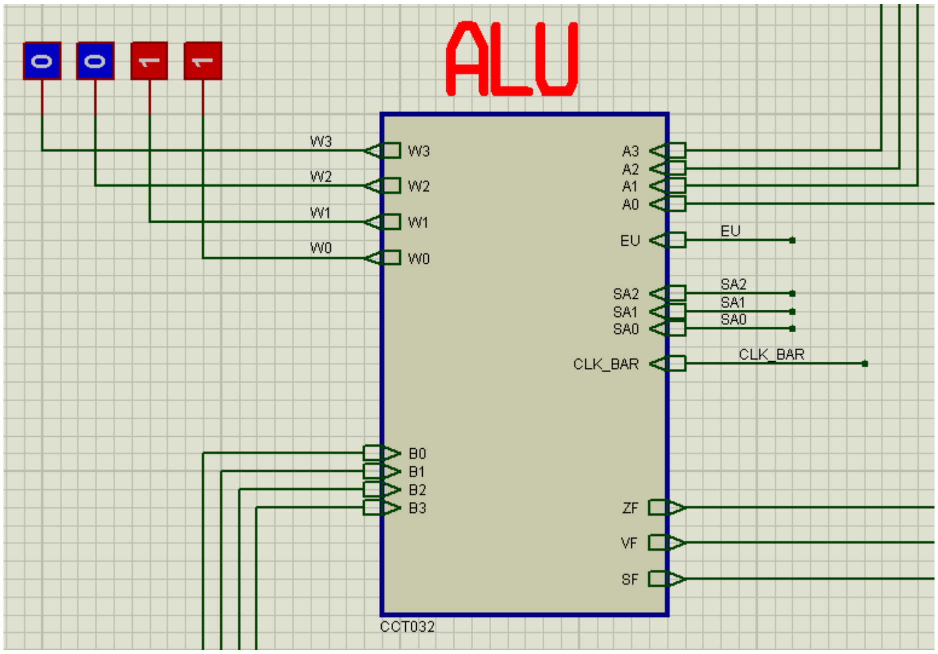
6. Description of Accumulator Register Block:



The Accumulator register block stores the 4bit data for arithmetic and logic operations in ALU. It has 4 input pins EN_ACC, LD_ACC, CLK, CLK_BAR, 4 bidirectional pins W0-to-W3 and 4 output pins A0-toA3. Accumulator data (X0-X3) are connected to W-BUS with 74LS244 buffer. A0-A3 pins are fed into ALU. It loads 4 bit data from W-BUS at CLK signal when LD_ACC pin is HIGH. It releases 4 bit data to W-BUS when EN_ACC is HIGH. When Accumulator loads data output A0-A3 pins directly passes data to ALU. Accumulator is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS173 (Quad D-type Flip-Flop with tri-state output)	1
74LS04 (NOT GATE)	2 Gates
74LS244 (Octal Buffer with tri-state output)	1
74LS126 (Quadruple bus buffer gate with tri-state output)	1

7. Description of Arithmetic Logic Unit (ALU) Block:

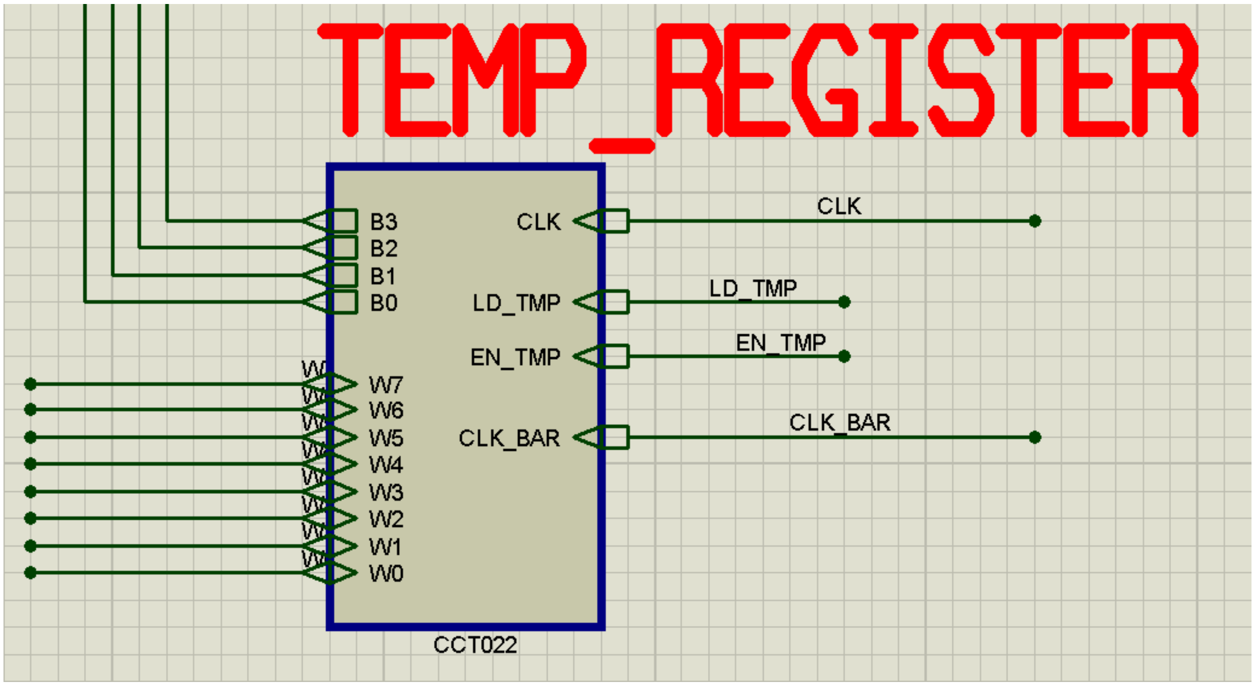


The ALU block performs arithmetic and logic operations on 4 bit data coming from Accumulator and Temporary register. It has 13 input pins EU, CLK_BAR, SA0, SA1, SA2, 4 pins A0-A3, 4 pins B0-B3 and 7 output pins ZF, VF, SF, 4 pins W0-W7. 4 bit data A0-A3 comes from Accumulator and 4 bit data B0-B3 comes from Temporary register. ALU results D0-D3 are connected to W-BUS (W0-W3 output pins) through 74LS244 buffer. The 3 input SA0-to-SA2 pins are selector pins which selects the ALU operations to perform. ZF, VF, SF are zero, overflow and sign flag output that is fed to flag register. ALU releases result to W-BUS when EU pin is HIGH.

ALU is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS126 (Quadruple bus buffer gate with tri-state output)	1 Gate
74LS04 (NOT GATE)	1 Gate
74LS244 (Octal Buffer with tri-state output)	1
74LS32 (OR GATE)	2 Gate
4075 (3-OR GATE)	1 Gate
NOR-4 (4 input NOR GATE- Digital)	1 Gate
XOR (2 input XOR GATE)	1 Gate
Decoder_3_8 (3 to 8 Decoder-Digital)	1
Function_4_4 (N bit ALU-Digital)	1

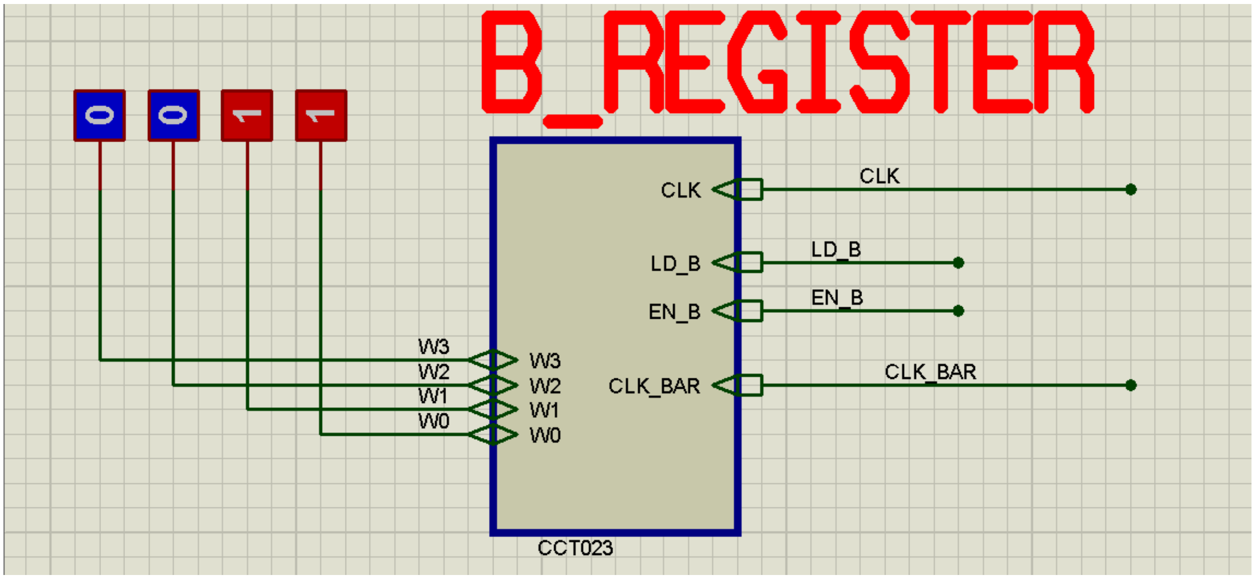
8. Description of Temporary Register Block:



The Temporary register block stores the 4 bit data of B register before any ALU operation and also store program counter 8 bit address in CALL and RET instruction cycle. It has 4 input pins LD_TMP, EN_TMP, CLK, CLK_BAR, 8 bidirectional pins W0-to-W7 and 4 output pins B0-B3. Temporary register 8 bit data (D0-D7) is connected to W-BUS through 74LS244 buffer and 4 bit data (D0-D3) is directly fed to ALU through 4 output pins B0-B3. Temporary register loads data from W-BUS at CLK signal when LD_TMP pin is HIGH and releases data to W-BUS when EN_TMP pin is HIGH. Temporary register is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS173 (Quad D-type Flip-Flop with tri-state output)	2
74LS04 (NOT GATE)	2 Gates
74LS244 (Octal Buffer with tri-state output)	1
74LS126 (Quadruple bus buffer gate with tri-state output)	1 Gate

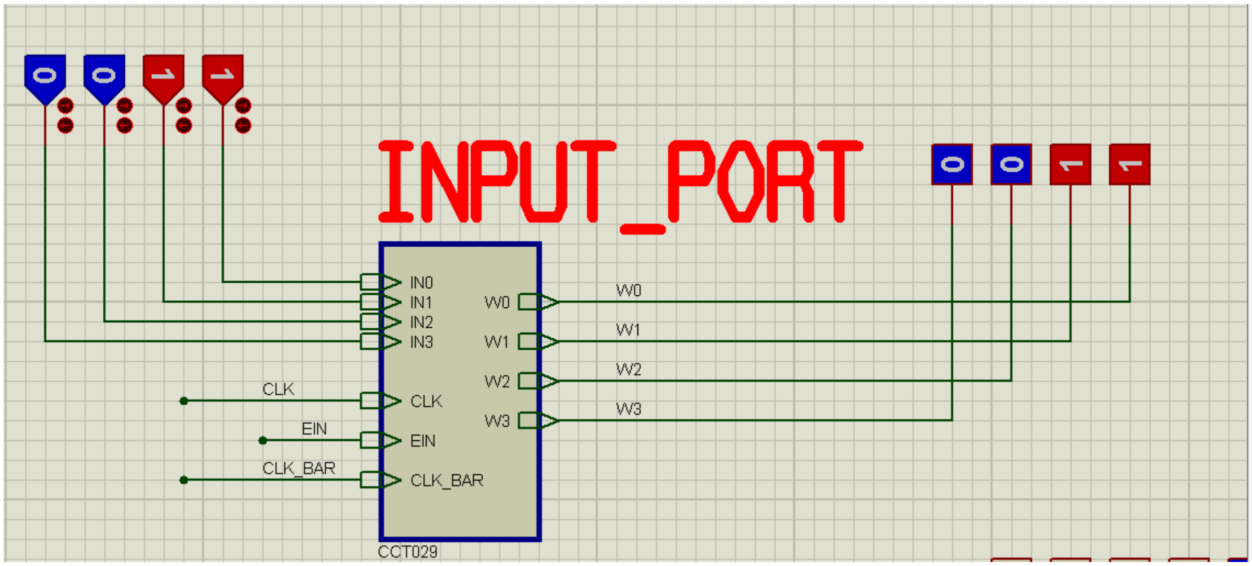
9. Description of B Register Block:



The B register block stores the 4 bit data of B register (supplies data to be added/subtracted from accumulator). It has 4 input pins LD_B, EN_B, CLK, CLK_BAR and 4 output pins W0-W3.B register data (D0-D3) is connected to W-BUS through 74LS244 buffer. It loads 4 bit data from W-BUS at CLK signal when LD_B pin is HIGH and releases 4 bit data to W-BUS when EN_B pin is HIGH. B register is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS173 (Quad D-type Flip-Flop with tri-state output)	1
74LS04 (NOT GATE)	2 Gates
74LS244 (Octal Buffer with tri-state output)	1 Gate

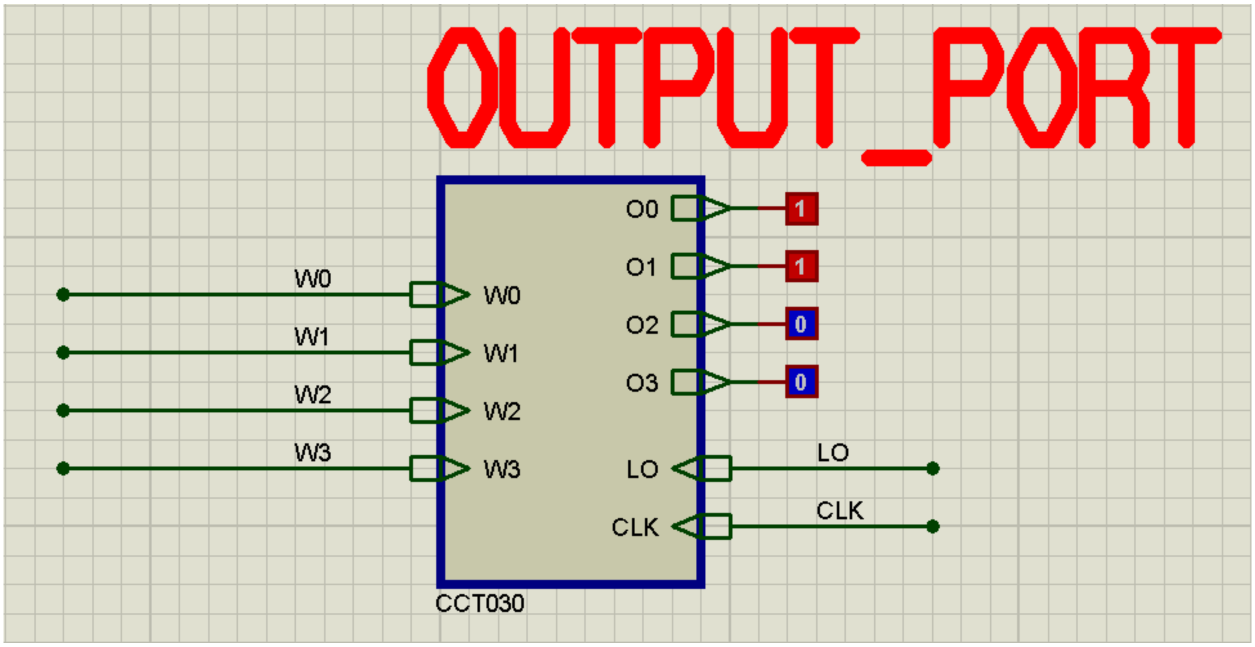
10. Description of Input Port Block:



The Input port block stores the 4 bit data from input switch. It has 7 input pins EIN, CLK, CLK_BAR, 4 pins IN0-to-IN3 (input data) and 4 output pins W0-to-W3. It loads data from input switches IN0-IN3 when EIN pin is HIGH and passes it to W-BUS (W0-W3). Input port block is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS04 (NOT GATE)	1 Gate
74LS244 (Octal Buffer with tri-state output)	1 Gate

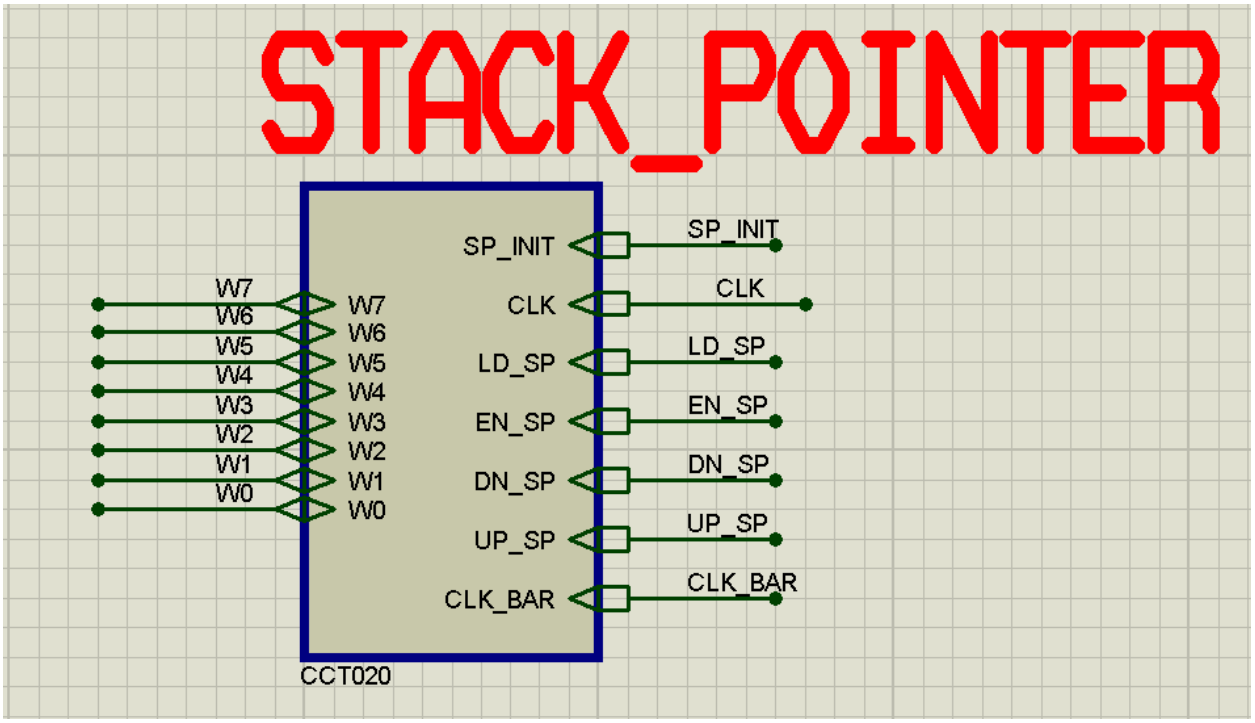
11. Description of Output Port Block:



The Output port block passes the 4 bit data from W-BUS to output probe. It has 6 input pins LO, CLK, 4 pins W0-to-W3 (W-BUS data) and 4 output pins O0-to-O3. It loads data from W-BUS W0-W3 at CLK signal when LO pin is HIGH and passes it to output O0-O3. Output port block is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS04 (NOT GATE)	1 Gate
74LS173 (Quad D-type Flip-Flop with tri-state output)	1

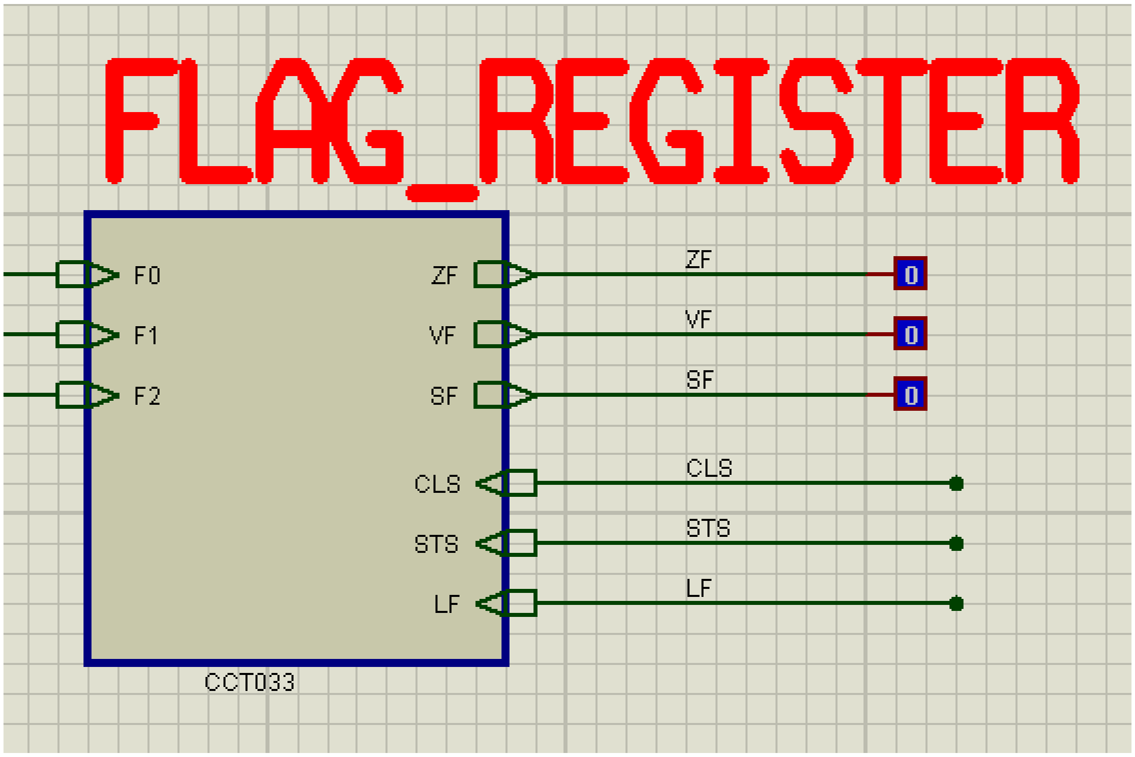
12. Description of Stack Pointer Block:



The Stack pointer register block contains the address (pointer) of stack memory. It has 7 input pins SP_INIT, CLK, CLK_BAR, UP_SP, DN_SP, LD_SP, EN_SP and 8 bidirectional pins W0-to-W7. It loads address bit from W-BUS at CLK signal when LD_SP pin is HIGH and releases stack pointer data bits to W-BUS when EN_SP is HIGH. It increments stack pointer address when UP_SP is HIGH and decrements address when DN_SP is HIGH. Stack pointer register data S0-S7 is connected to W-BUS through 74LS244 buffer. Initially stack pointer address is loaded to FF when SP_INIT pin is HIGH. Stack pointer block is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS193 (4 bit Synchronous up-down binary counter)	2
74LS157 (Quadruple 1-of-2 Data Multiplexers)	1
74LS04 (NOT GATE)	5 Gates
74LS244 (Octal Buffer with tri-state output)	2
74LS08 (AND GATE)	2 Gates

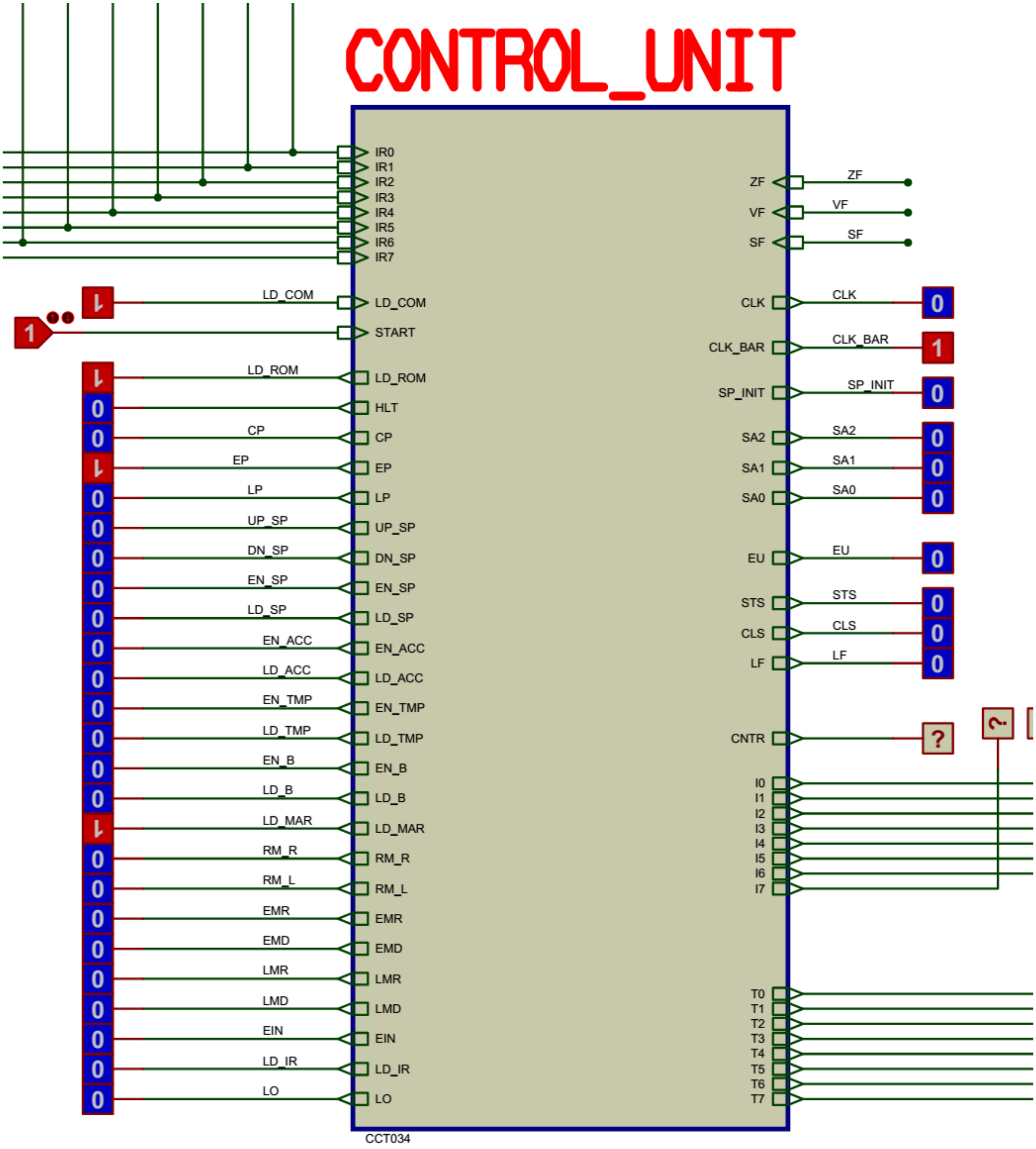
13. Description of Flag Register Block:



The Flag register block stores the flag bit after any ALU operation. It has 6 input pins F0, F1, F2, CLS, STS, LF and 3 output pins ZF, SF, VF. F0, F1, F2 pins carry the zero, overflow, sign flag data and come from ALU. ZF, SF, VF pins data are fed into Control Unit. It loads data F0, F1, F2 into register when LF pin is HIGH. It clears sign flag bit when CLS pin is HIGH. It sets sign flag bit when STS pin is HIGH. Flag register block is in HALT state when all input pins are LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS193 (4 bit Synchronous up-down binary counter)	2
74LS157 (Quadruple 1-of-2 Data Multiplexers)	2
74LS04 (NOT GATE)	2 Gates
74LS32 (OR GATE)	2 Gates

14. Description of Control Unit Block:



Control unit block controls the execution of instructions of 4 bit PC by sending CLK pulse and necessary signals to other circuit blocks. It has 34 output pins which contains 30 bit control word of 4 ROM (except control bit HLT and END/NXT) and 4 other signals CLK, CLK_BAR, LD_ROM, SP_INIT. It has 13 input pins IR0-to-IR7 (8 bit instruction code), LD_COM, START, ZF, VF, SF. PC starts when control unit START pin is HIGH, then LD_ROM pin becomes HIGH (also in RAM block) which starts loading instructions from ROM to RAM in RAM block and SP_INIT pin becomes HIGH (also in SP block) which starts loading Stack pointer block register to FF. After instruction loading is complete in RAM block it sends a LD_COM signal and so LD_COM input pin becomes HIGH which starts the CLOCK pulse of Control Unit (or PC). In our control unit CLK and CLK_BAR are sent to each block along with signal bits. Control Unit executes instructions in T states or clock cycles. Every instruction needs 3 clock cycles (T0, T1, T2) to fetch from RAM to control unit through IR0-to-IR7. In each clock cycle (or T state) 32 control bits are generated from 4 ROM where 2 bit HLT and END/NXT controls the control unit. After T2 clock cycle control unit loads the 8 bit instruction code from IR block and execution starts. When execution of an instruction is complete it goes back to initial T0 state of fetch cycle except HLT bit is HIGH. If HLT bit is high then execution stops and all blocks get LOW in their input pins along with CLK and CLK_BAR pin. The input pin ZF, VF, SF is from flag register block which is needed to make decisions for JNZ, JL instructions. The pin designation of 4 ROM is-

Signal->	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ROM 1	CP	EP	LP	UP_SP	DN_SP	EN_SP	LD_SP	EN_ACC
ROM 2	LD_ACC	EN_TMP	LD_TMP	EN_B	LD_B	LD_MAR	RM_R	RM_L
ROM 3	EMR	EMD	LMR	LMD	EIN	LD_IR	LO	SA2
ROM 4	SA1	SA0	EU	STS	CLS	LF	END/NXT	HLT

When control unit is in HLT state all output pins become LOW. Its sub-circuit contains-

IC Name	IC Counts
74LS161 (Synchronous 4 bit binary counter)	2
74LS04 (NOT GATE)	10 Gates
74LS173 (Quad D-type Flip-Flop with tri-state output)	1
74LS157 (Quadruple 1-of-2 Data Multiplexers)	4
74LS32 (OR GATE)	2 Gates
CLOCK (Square wave source-Digital)	2
2732 (4K X 8 EPROM Memory)	4
4078 (8-input NOR GATE)	3 Gates
74LS386 (XOR GATE)	4 Gates
74LS126 (Quadruple bus buffer gate with tri-state output)	1 Gate

➢ Explanation of all Instructions:

❖ Instructions, Op-codes, Clock cycles:

Instruction	Length(in Byte)	Op-code(8/16 bit) (HEX + Addr/Imm)	Clock Cycle (Execution)
LDA address;	2	**03+** Address	**5**
STA address;	2	**08+** Address	**5**
MOV Acc, B;	1	**0D**	**1**
MOV B, Acc;	1	**0E**	**1**
MOV Acc,immediate;	2	**0F+** Immediate	**3**
IN;	1	**12**	**1**
OUT;	1	**13**	**1**
ADD B;	1	**14**	**2**
ADC B;	1	**16**	**2**
SUB B;	1	**18**	**2**
SBB B;	1	**1A**	**2**
ADC immediate;	2	**1C+** Immediate	**4**
SBB immediate;	2	**20+** Immediate	**4**
CMP B;	1	**24**	**2**
TEST B;	1	**26**	**2**
JNZ address;	2	**32+** Address	**3**
JL address;	2	**35+** Address	**3**
PUSH;	1	**38**	**3**
POP;	1	**3B**	**3**
CALL address;	2	**3E+** Address	**16**
RET;	1	**4E**	**12**
JMP address;	2	**5A+** Address	**3**
HLT;	1	**5D**	**1**
NOP;	1	**5E**	**1**
STS;	1	**30**	**1**
CLS;	1	**31**	**1**
AND [address];	2	**28+** Address	**6**
OR B;	1	**2E**	**2**

❖ Description of Instructions:

No	Instruction	Description
1	LDA address;	It reads data from RAM address (of instruction) and store that data to the Accumulator.
2	STA address;	It stores data of Accumulator to RAM address (of instruction).
3	MOV Acc, B;	It stores B register data to Accumulator.
4	MOV B, Acc;	It stores Accumulator data to B register.
5	MOV Acc,immediate;	It stores immediate data of instruction to Accumulator.
6	IN;	It reads data from input port and stores it to Accumulator.
7	OUT;	It reads data from Accumulator and stores it to output port.
8	ADD B;	It adds B register data with Accumulator data and stores the resulted data to Accumulator and Load Flag register.
9	ADC B;	It adds B register data with Accumulator data and carry bit, and stores the resulted data to Accumulator and Load Flag register.
10	SUB B;	It subtracts B register data from Accumulator data and stores the resulted data to Accumulator and Load Flag register.
11	SBB B;	It subtracts B register data from Accumulator data and borrow bit and stores the resulted data to Accumulator and Load Flag register.
12	ADC immediate;	It adds immediate data (of instruction) with Accumulator data and carry bit, and stores the resulted data to Accumulator and Load Flag register.
13	SBB immediate;	It subtracts immediate data (of instruction) from Accumulator data and borrow bit and stores the resulted data to Accumulator and Load Flag register.
14	CMP B;	It subtracts B register data from Accumulator data and stores the resulted data to Accumulator and Load Flag register.

No	Instruction	Description
15	TEST B;	It performs bitwise AND operation between B register data and Accumulator data stores the resulted data to Accumulator and Load Flag register.
16	JNZ address;	It jumps to instruction address (of instruction) if zero flag is not set and doesn't jump otherwise.
17	JL address;	It jumps to instruction address (of instruction) if VF != SF and doesn't jump otherwise.
18	PUSH;	It writes the data of Accumulator to RAM at the address of Stack pointer register.
19	POP;	It reads data from RAM at the address of Stack pointer register and stores it to Accumulator.
20	CALL address;	It calls a subroutine and jumps to instruction address (of instruction) unconditionally and start executing from that instruction address's instructions.
21	RET;	It returns from current subroutine to caller subroutine unconditionally and loads previous instruction address from Stack.
22	JMP address;	It jumps to instruction address (of instruction) unconditionally and start executing from that instruction address's instructions.
23	HLT;	It halts all execution and clock cycle.
24	NOP;	No operation in this cycle.
25	STS;	It sets the Sign flag in Flag register.
26	CLS;	It clears the Sign flag in Flag register.
27	AND [address];	It does bitwise AND operation between the data of RAM address (in instruction) and Accumulator data and stores the resulted data to Accumulator and Load Flag register.
28	OR B;	It does bitwise OR operation between B register data and Accumulator data and stores the resulted data to Accumulator and Load Flag register.

➢ Explanation of Control Signals:

Our Control Unit generates 32 bit control signals from 4 ROM in each clock cycle. The control signals are-

Signal->	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ROM 1	CP	EP	LP	UP_SP	DN_SP	EN_SP	LD_SP	EN_ACC
ROM 2	LD_ACC	EN_TMP	LD_TMP	EN_B	LD_B	LD_MAR	RM_R	RM_L
ROM 3	EMR	EMD	LMR	LMD	EIN	LD_IR	LO	SA2
ROM 4	SA1	SA0	EU	STS	CLS	LF	END/NXT	HLT

The Control signals are active at HIGH state except (RM_R, RM_L, SA0, SA1, SA2). The task of control signals are-

No	Signal	Task
1	**CP**	Increments Program counter register's instruction address.
2	**EP**	Releases Program counter register's instruction address to W-BUS.
3	**LP**	Loads instruction address from W-BUS to Program counter.
4	**UP_SP**	Increments Stack Pointer register's address.
5	**DN_SP**	Decrements Stack Pointer register's address.
6	**EN_SP**	Releases Stack Pointer register's address to W-BUS.
7	**LD_SP**	Loads address data from W-BUS to Stack Pointer register.
8	**EN_ACC**	Releases Accumulator data to W-BUS.
9	**LD_ACC**	Loads Accumulator with data of W-BUS.
10	**EN_TMP**	Releases Temporary register's data to W-BUS.
11	**LD_TMP**	Loads Temporary register with data of W-BUS.
12	**EN_B**	Releases B register's address to W-BUS.
13	**LD_B**	Loads B register with data of W-BUS.
14	**LD_MAR**	Loads address from W-BUS to MAR register.
15	**RM_R**	Controls read/write operation in RAM.
16	**RM_L**	Controls read/write operation in RAM.
17	**EMR**	Releases MDR register's data to RAM input.
18	**EMD**	Releases MDR register's data to W-BUS.
19	**LMR**	Loads data from RAM output to MDR register.

No	Signal	Task
20	LMD	Loads data from W-BUS to MDR register.
21	EIN	Loads Input port's data in W-BUS.
22	LD_IR	Loads data from W-BUS to Instruction register.
23	LO	Loads data from W-BUS to Output port.
24	SA2	Selection bit for ALU operation.
25	SA1	Selection bit for ALU operation.
26	SA0	Selection bit for ALU operation.
27	EU	Releases resulted data to Accumulator after any ALU operation.
28	STS	Sets Sign flag in Flag register.
29	CLS	Clears Sign flag in Flag register.
30	LF	Loads flag bits from ALU output in Flag register.
31	END/NXT	Increment T state if NXT = 1 and Load instruction address from IR if NXT = 0 and start again from T0 state.
32	HLT	HALT all executions and operation in all registers.

RAM read write controls are-

RAM mode	RM_R	RM_L
HALT	0	0
READ	0	1
WRITE	1	X

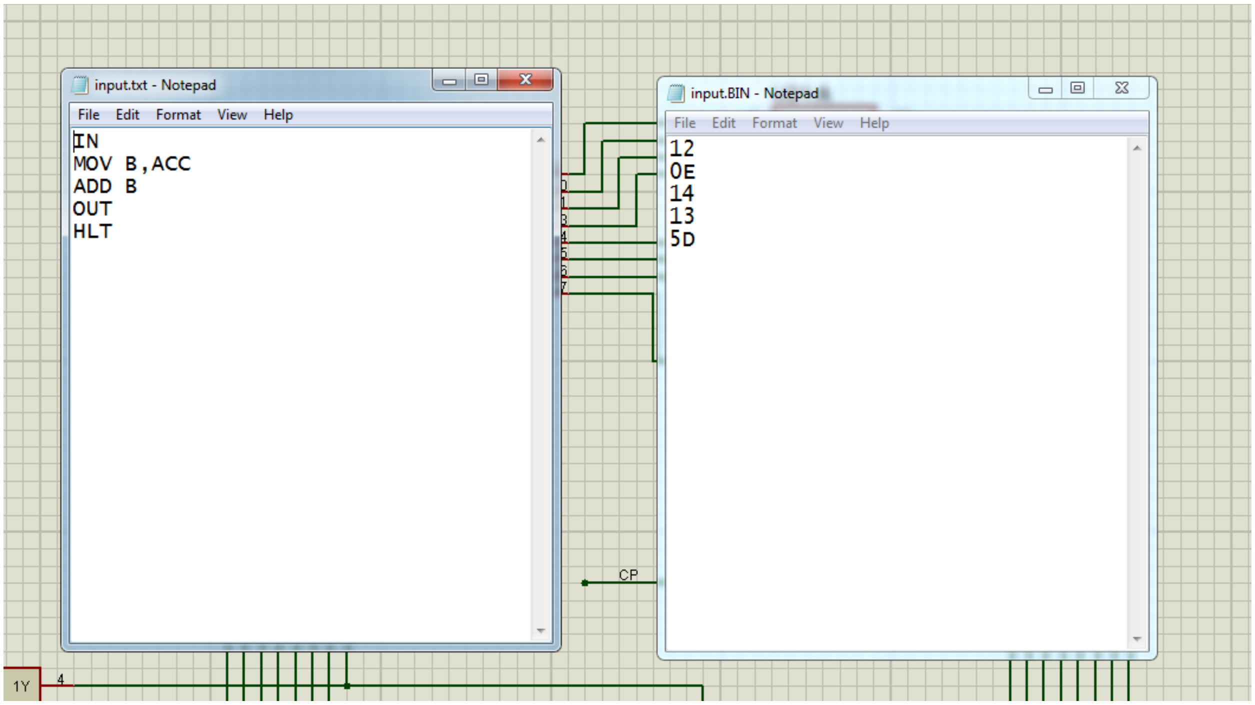
ALU Operation Selections are-

SA2	SA1	SA0	Operation
0	0	0	Do Nothing
0	0	1	ADD
0	1	0	ADC
0	1	1	SUB
1	0	0	SBB
1	0	1	CMP
1	1	0	TEST
1	1	1	OR

➢ How to execute program in 4bit PC:

We have used Proteus Design Suite [©Labcenter Electronics 1989-2009] Release 7.7 SP2 (Build 9089) with Advanced Simulation Tool for designing the 4-bit PC. The steps for executing programs are-

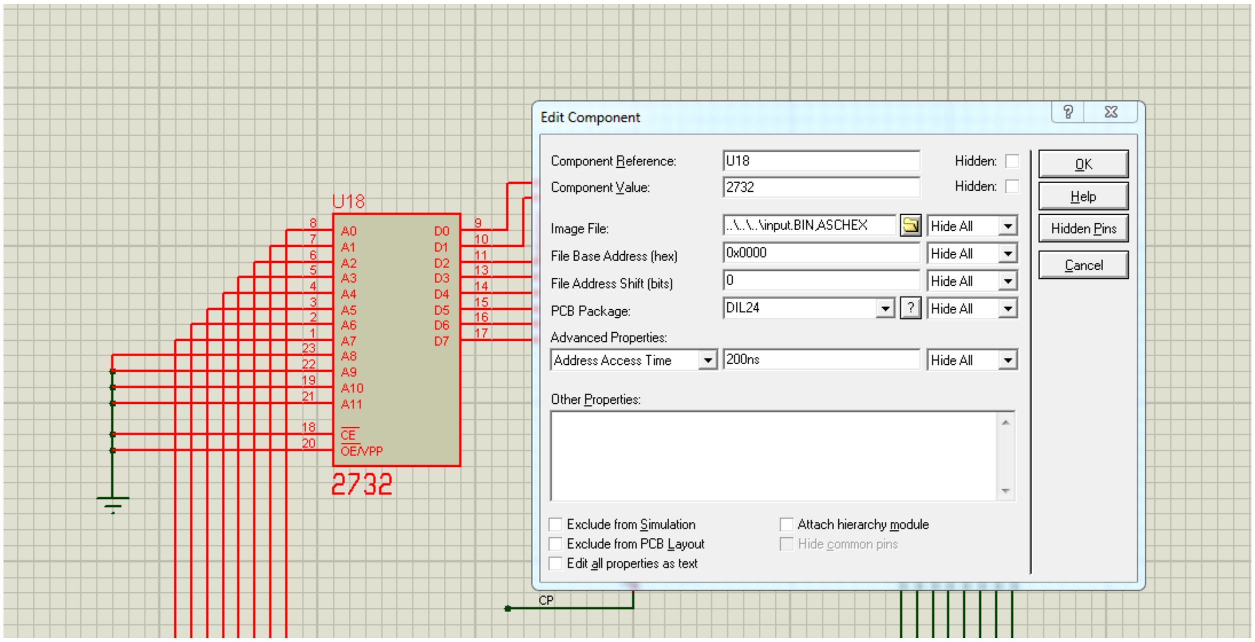
- At first we have to write a program in 'input.txt' file containing executable instructions of this PC. Then a binary file 'input.bin' containing HEX value (Op-code) has to be written from our Instruction set look-up table (Page -X) for 'input.txt' program.



```
IN
MOV B,ACC
ADD B
OUT
HLT
```

```
12
0E
14
13
5D
```

- After writing file ‘input.bin’, it has to be loaded to ROM 2732 of RAM block.



Edit Component

Component Reference: U18 Hidden: □
Component Value: 2732 Hidden: □

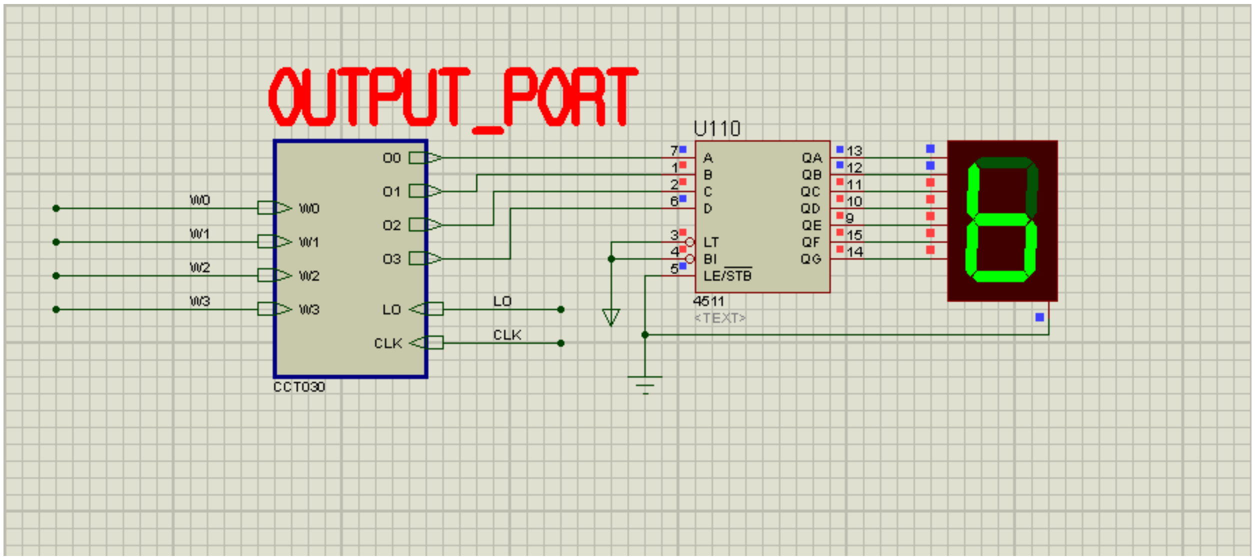
Image File: ..\..\..\input.BIN,ASCHEX Hide All
File Base Address (hex) 0x0000 Hide All
File Address Shift (bits) 0 Hide All
PCB Package: DIL24 ? Hide All

Advanced Properties:
Address Access Time 200ns Hide All

Other Properties:

□ Exclude from Simulation
□ Exclude from PCB Layout
□ Edit all properties as text
□ Attach hierarchy module
□ Hide common pins

- Now if we set START pin ON/HIGH and RUN simulation then our instructions would be executed and we can see output in output port.



➢ IC used for 4 bit PC Design:

No	IC Number	IC Name	IC Count
1	**74LS161**	Synchronous 4 bit binary counter	6
2	**74LS04**	2-input NOT GATE	7
3	**74LS173**	Quad D-type Flip-Flop with tri-state output	12
4	**74LS193**	4 bit Synchronous up-down binary counter	4
5	**74LS244**	Octal Buffer with tri-state output	12
6	**74LS08**	2-input AND GATE	1
7	**PULSE**	Monostable pulse generator-Digital	2
8	**6116**	2K X 8 Static RAM Memory	1
9	**74LS157**	Quadruple 1-of-2 Data Multiplexers	12
10	**74LS32**	2-input OR GATE	2
12	**CLOCK**	Square wave source-Digital	3
13	**2732**	4K X 8 EPROM Memory	5
14	**4078**	8-input NOR GATE	3
15	**74LS386**	2-input XOR GATE	1
16	**74LS126**	Quadruple bus buffer gate with tri-state output	3
17	**Function_4_4**	N bit ALU-Digital	1
18	**NOR-4**	4 input NOR GATE- Digital	1
19	**XOR**	2 input XOR GATE-Digital	1
20	**Decoder_3_8**	3 to 8 Decoder-Digital	1
21	**4075**	3-input OR GATE	1
		Total IC used	79

Discussion:

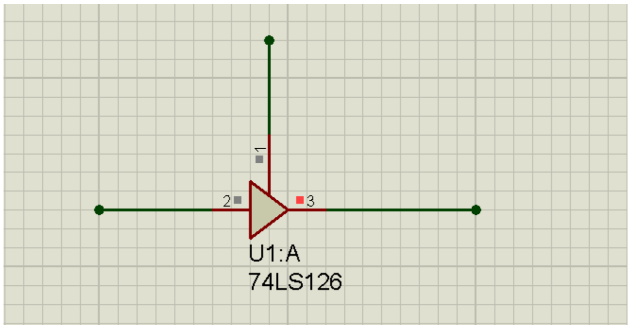
- We used Proteus Design Suite which is very handy tool to design 4 bit PC architecture. We labeled all necessary wires and used logic probes to see all necessary wires state for debugging. We also marked our sub-circuits with large labels so that anyone can understand easily.

- We developed fully automated 4 bit PC, that means no manual input is needed for loading ROM data into RAM. After simulation starts if we ON START switch then ROM will start loading into RAM and then Control Unit will start executing instructions.

- Our design contains 4 bit data bus (lower 4 bit of W-BUS) and 8 bit address bus (W-BUS). Address and data are not sent to W-BUS at the same time. We used RAM as our Stack cause RAM size is 256 X 8.

- While designing in Proteus we faced a lot of errors –

1. While using 74LS126 (tri-state-buffer) we see that its output HIGH when input and selector are garbage. But 74LS244 (buffer) don't act like that, its output LOW when selector is garbage. So it took time to figure out this problem and avoided using 74LS126 IC.



2. While designing Flag register, for STS and CLS instruction we feed back ZF and VF bit to input which caused a race condition error and simulation didn't start. Later we found out that problem and change it using MUX.

- We successfully executed 26 instructions out of 28. RET and SBB instruction didn't give the desired output. We think there may be some mistakes in theory or circuit implementation which caused malfunction. As ADC instruction worked well so we can do SBB instruction by Result = A+B_BAR+1 in ALU, that is by doing Addition with B inverse and carry rather than Subtraction. RET instruction had 12 clock cycles in execution stage, so it was hard to debug.