



Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 01: NOT, AND, OR Gate

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify **NOT** gate operations using 74LS04 IC & Module KL-33001
- To implement and verify **AND** gate operations using 74LS08 IC & Module KL-33001
- To implement and verify **OR** gate operations using 74LS32 IC & Module KL-33001

Lab Hardware and Software Required:

1. 74LS04 IC (Inverter)
2. 74LS08 IC(AND gate)
3. 74LS32 IC(OR gate)
4. Module KL-33001
5. Breadboard
6. Connecting Wires

Background Theory:

NOT gate:

The NOT gate is an electronic circuit that produces an inverted version of input logic at its output. It is also known as inverter. If the input variable is A, the inverted output is known as NOT A. alternatively if the input is logic low (0), the output will be logic high (1) and if the input is logic high (1), the output will be logic low (0). NOT gate has only one input and one output, as shown in Fig 1.1 (a & b):

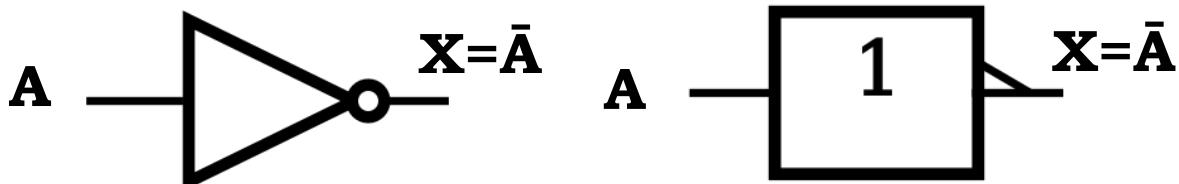


Fig 1.1 (a) Distinctive symbol of NOT gate

(b) Standard NOT gate symbol

The Boolean equation for an inverter is written $X = \bar{A}$ (which is read as “X equals NOT A”). The bar over the A is an inverter bar, used to represent the NOT operation.

74LS04 Inverted IC:

In order to implement the NOT gate operation using IC, the TTL 74LS04 IC can be used. This IC contains six inverters. It has 14pin Dual Inline Package (DIP) configuration as shown in Fig 1.2. The power supply connections are made to pin 7 and 14. This supply the operating voltage for all six NOT gates on the IC. Pin 1 is identified by a small indented circle next to it or by a notch cut out between pin 1 and 14.

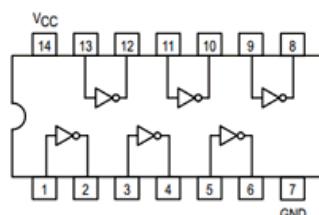


Fig 1.2: 74LS04 Inverter IC pin configuration

AND gate:

The AND gate is an electronic circuit that gives a high output (1) if all its inputs are high (1). When any one of its input is low (0), the output is low (0). The AND gates are composed of two or more inputs and a single output, as indicated by logic symbols in Fig 1.3 (a & b):



Fig 1.3: (a) Distinctive symbol of 2-input AND gate; (b) Standard 2-input AND gate symbol

There is no limit to the number of inputs that may be applied to an AND functions. However, for practical reasons, commercial AND gates are not most commonly manufactured with 2, 3 or 4 inputs. The Boolean expression for the AND operation is $X = A \cdot B$ (which is read as “X equals to A AND B”). The dot (.) sign is used to show the AND expression, however this sign is usually omitted. The key thing to remember is that the AND operation will produce a result of 1 only when all inputs (variables) are 1, just like ordinary multiplication.

74LS08 2-input AND gate IC:

In order to implement the AND operation using IC, the TTL 74LS08 2-input AND gate IC can be used. This IC contains four AND gates. It has 14 pin DIP configuration as shown in Fig 1.4:

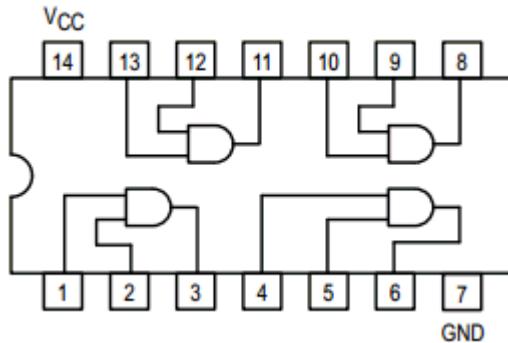


Fig 1.4: 74LS08 2-input AND gate IC pin configuration

OR Gate:

The OR gate produces output high when any one of inputs is high. The output is low when all of the inputs are low. The OR gate also has two or more inputs and a single output. The symbol for a two input OR gate is shown in Fig 1.5 (a & b):



Fig 1.5: (a) (a) Distinctive symbol of 2-input OR gate; (b) Standard 2-input OR gate symbol

The Boolean expression for the OR operation is written as $X = A + B$ (which is read as “X equals A OR B”). Notices the use of (+) symbol to represent the OR function.

74LS32 2-Input OR Gate IC:

In order to implement the OR operation using IC, the TTL 74LS32 2-input OR gate IC can be used. It has four OR gates with in the package. This IC has 14 pin DIP configuration as shown in Fig 1.6

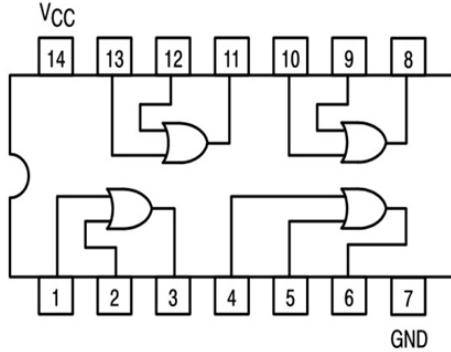


Fig 1.6: 74LS32 2-input OR gate IC pin configuration

Lab Examples:

Implementation of NOT gate

- To implement the NOT operations with the help of ICs, take the **74LS04**
- Pin Assignments for the ICs have been shown in fig 1.2
- Take 74LS04 IC and insert it in the breadboard present on the **KL-33001** training kit
- Connect its pin 14 to +5V and pin 7 to ground
- Use first gate (anyone can be taken) from IC by connecting input pin 1 to switch and output pin2 to LED and observing the NOT gate operation
- Fill the following observation table

Input	Output	
A	LED (on / off)	Level (1 / 0)
0	On	1
1	Off	0

Table 1.1: Observation table of NOT gate

Lab Activities:

1. Implementation of AND gate

- In order to implement the AND operations with the help of IC, take the **74LS08** IC
- Pin Assignments for the ICs have been shown in fig 1.4
- Take 74LS08 IC and insert it in the breadboard present on the **KL-33001** training kit
- Connect its pin 14 to +5V and pin 7 to ground
- Use first gate (any one can be taken) from IC by connecting input pin 1 & input pin 2 to SW0 & SW1 respectively, and output pin 3 to LED and observe the AND gate operation
- Fill the following observation table

Input		Output	
A	B	LED (on / off)	Level (1 / 0)
1	1	On	1
1	0	Off	0
0	1	Off	0
0	0	Off	0

Table 1.2: Observation table of AND gate

2. Implementation of OR gate

- In order to implement the OR operations with the help of IC, take the **74LS32** IC
- Pin Assignments for the ICs have been shown in fig 1.6
- Take 74LS08 IC and insert it in the breadboard present on the **KL-33001** training kit
- Connect its pin 14 to +5V and pin 7 to ground
- Use first gate (any one can be taken) from IC by connecting input pin 1 & input pin 2 to SW0 & SW1 respectively, and output pin 3 to LED and observe the OR gate operation
- Fill the following observation table

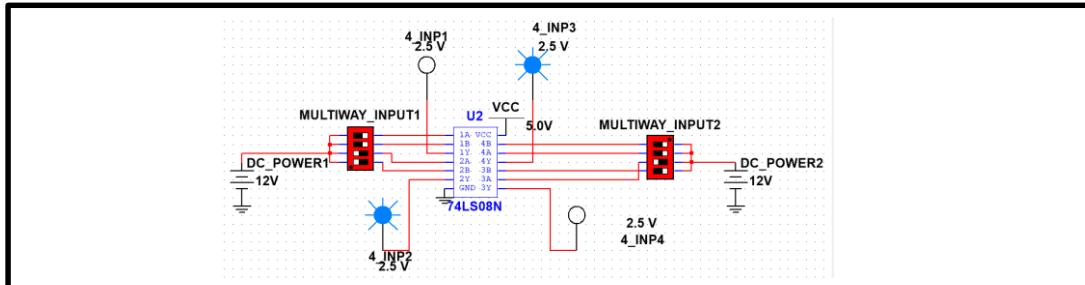
Input		Output	
A	B	LED (on / off)	Level (1 / 0)
0	0	Off	0
0	1	On	1
1	0	On	1
1	1	On	1

Table 1.3: Observation table of OR gate

Lab Exercises:

1. Design the circuit using 4 input AND Gate using one IC

- Draw the schematic Diagram
- Fill the Observation Table

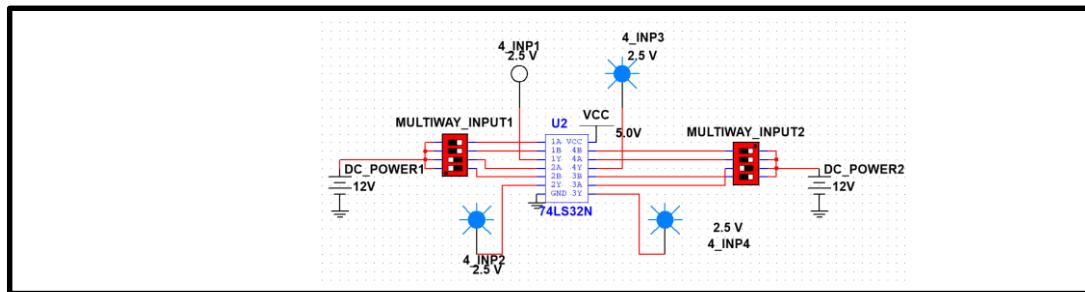


Input				Output	
A	B	C	D	LED (on / off)	Level (1 / 0)
0	0	0	0	Off	0
0	0	0	1	Off	0
0	0	1	0	Off	0
0	0	1	1	Off	0
0	1	0	0	Off	0
0	1	0	1	Off	0
0	1	1	0	Off	0
0	1	1	1	Off	0
1	0	0	0	Off	0
1	0	0	1	Off	0
1	0	1	0	Off	0
1	0	1	1	Off	0
1	1	0	0	Off	0
1	1	0	1	Off	0
1	1	1	0	Off	0
1	1	1	1	On	1

Table 1.4: Observation table of 4-Input AND gate

2. Design the circuit using 4 input OR Gate using one IC

- Draw the schematic Diagram
- Fill the Observation Table



Input				Output	
A	B	C	D	LED (on / off)	Level (1 / 0)
0	0	0	0	On	1
0	0	0	1	On	1
0	0	1	0	On	1
0	0	1	1	On	1
0	1	0	0	On	1
0	1	0	1	On	1
0	1	1	0	On	1
0	1	1	1	On	1
1	0	0	0	On	1
1	0	0	1	On	1
1	0	1	0	On	1
1	0	1	1	On	1
1	1	0	0	On	1
1	1	0	1	On	1
1	1	1	0	On	1
1	1	1	1	Off	0

Table 1.5: Observation table of 4-Input OR gate



Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 02: NAND & NOR Gates

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify **NAND** gate operations using 74LS00 IC & Module KL-33001
- To implement and verify **NOR** gate operations using 74LS02 IC & Module KL-33001

Lab Hardware and Software Required:

1. 74LS00 IC (NAND gate)
2. 74LS02 IC(NOR gate)
3. Module KL-33001
4. Breadboard
5. Connecting Wires

Background Theory:

NAND gate:

The logic symbols for NAND gate is shown in Fig 2.1 (a & b). It consists of AND symbol with inverter symbol added to output. The operation of NAND gate is same as the AND gate except that its output is inverted. NAND gates can also have two or more than two inputs and a single output. NAND gate produces a LOW output only when all the inputs are HIGH. When any of the inputs is LOW, the output will be HIGH.

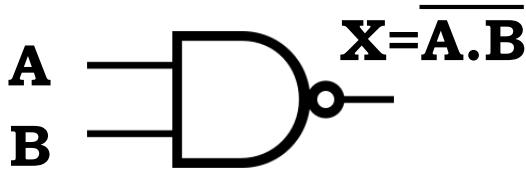
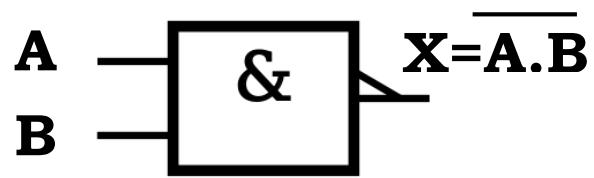


Fig 2.1: (a) Distinctive symbol of NAND gate



(b) Standard NAND gate symbol

74LS00 2-Input NAND gate IC:

In order to implement the NAND gate operation using IC, the TTL 74LS00 2-input NAND gate IC can be used. This IC has 14 pin Dual Inline Package (DIP) configuration as shown in Fig 2.2. The power supply connections are made to pin 7 and 14. Pin 1 is identified by a small indented circle next to it or by a notch cut out between pin 1 and 14.

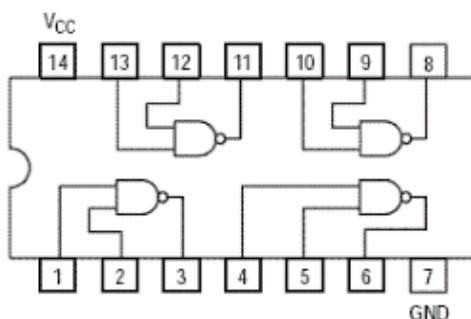


Fig 2.2: 74LS00 2-input NAND gate IC configuration

NOR gate:

The operation of the NOR gate is same as that of the OR gate except that its output is inverted. You can think of a NOR gate as an OR gate with an inverter at its output. The logic symbol for NOR gate is shown in Fig 2.3 (a & b). NOR gates can also have two or more than two inputs and a single output. The NOR gate produces a low output, When any one of input is high and produces a high output, when all inputs are low.

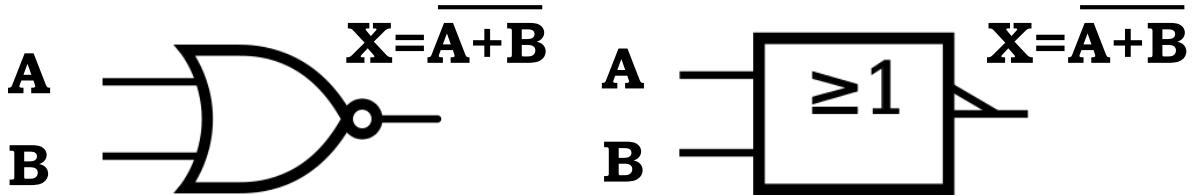


Fig 2.3: (a) Distinctive symbol of 2-input NOR gate (b) Standard 2-input NOR gate symbol

74LS02 2-input NOR gate IC:

In order to implement the NOR gate operation using IC, the TTL 74LS02 2-input NOR gate IC can be used. This IC has 14 pin Dual Inline Package (DIP) configuration as shown in Fig 2.4. The power supply connections are made to pin 7 and 14. Pin 1 is identified by a small indented circle next to it or by a notch cut out between pin 1 and 14.

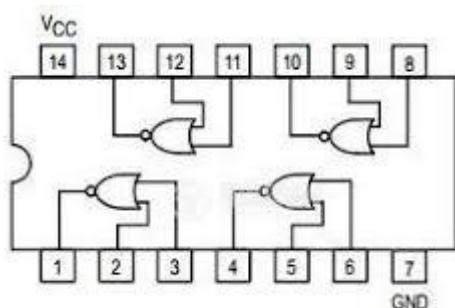


Fig 2.4: 74LS02 2-input NOR gate IC pin configuration

Lab Examples:

Implementation of NAND gate

- In order to implement the NAND gate operations with the help of ICs, take the **74LS00**
- Pin Assignments for the ICs have been shown in Fig 2.2
- Take 74LS00 IC and insert it in the breadboard present on the **KL-33001** training kit
- Connect its pin 14 to +5V and pin 7 to ground
- Use first gate (any one can be taken) from IC by connecting input pin 1 & input pin 2 to SW0 & SW1 respectively, and output pin 3 to LED and observe the AND gate operation
- Fill the following observation table

Input		Output	
A	B	LED (on / off)	Level (1 / 0)
0	0	On	1
0	1	On	1
1	0	On	1
1	1	Off	0

Table 2.1: Observation table of NAND gate

Lab Activities:

Implementation of NOR gate

- In order to implement the AND operations with the help of IC, take the **74LS02** IC
- Pin Assignments for the ICs have been shown in Fig 2.4
- Take 74LS02 IC and insert it in the breadboard present on the **KL-33001** training kit
- Connect its pin 14 to +5V and pin 7 to ground
- Use first gate (any one can be taken) from IC by connecting input pin 1 & input pin 2 to SW0 & SW1 respectively, and output pin 3 to LED and observe the AND gate operation
- Fill the following observation table

Input		Output	
A	B	LED (on / off)	Level (1 / 0)
0	0	On	1
0	1	Off	0
1	0	Off	0
1	1	Off	0

Table 2.2: Observation Table of NOR gate

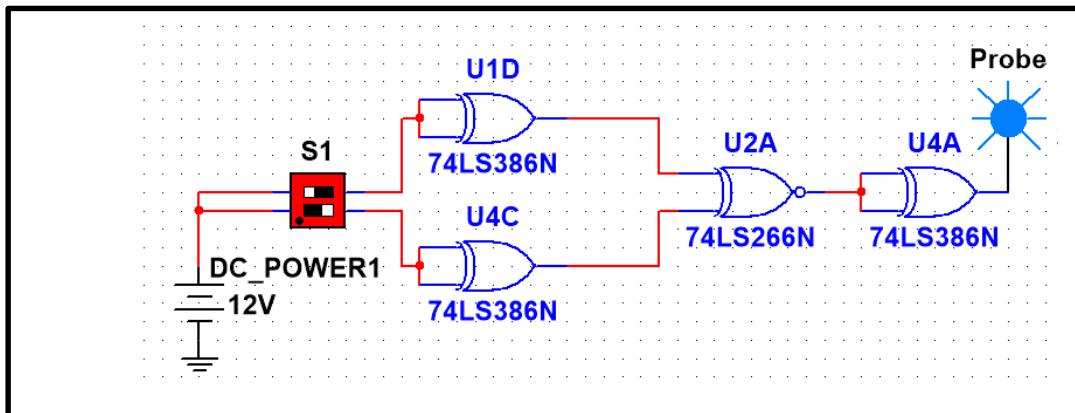
Lab Exercises:

NAND-NOR Universal Gates

NAND and NOR gates are known as Universal Gates as they can be used to construct an AND gate, an OR gate, an INVERTER, or any combination of these functions. The NAND Universal Gate can also be used to implement a NOR gate. Similarly, a NOR gate can be used to implement a NAND gate.

1. Implement NAND gate using NOR gates

- Draw the schematic Diagram
- Write the Boolean Expression
- Fill the Observation Table



Boolean Expression:

$$X = \overline{\overline{A + A} + \overline{B + B}} = \overline{\overline{A} + \overline{A} + \overline{B} + \overline{B}}$$

as $A + A = A$

$$X = \overline{\overline{A} + \overline{B}} + \overline{\overline{A} + \overline{B}}$$

$$\text{as } \overline{\overline{A} + \overline{B}} + \overline{\overline{A} + \overline{B}} = \overline{\overline{A} + \overline{B}}$$

$$X = \overline{\overline{A} + \overline{B}}$$

$$\text{as } \overline{\overline{A}} = A$$

$$X = \overline{A} + \overline{B}$$

$$\text{Demorgans law } \overline{A} + \overline{B} = \overline{A \cdot B}$$

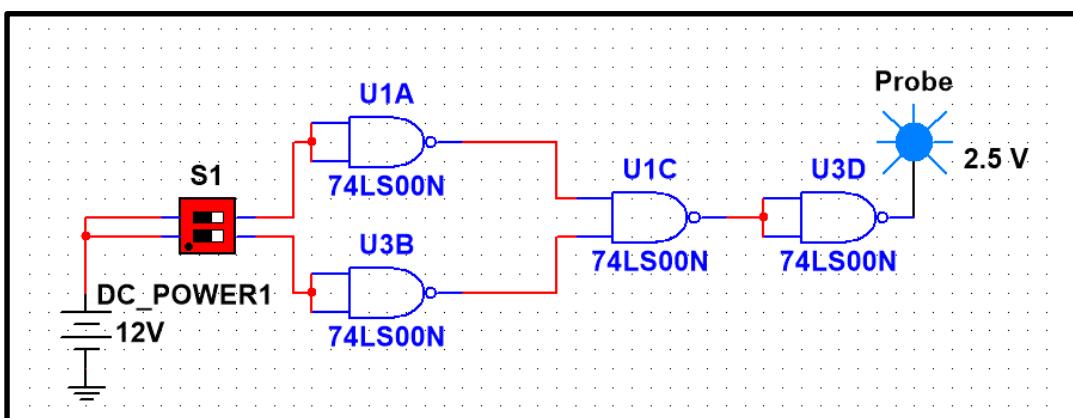
$$X = \overline{A \cdot B}$$

Input		Output	
A	B	LED (on / off)	Level (1 / 0)
0	0	On	1
0	1	On	1
1	0	On	1
1	1	Off	0

Fig 2.3: Observation table for 2-input NAND gate using NOR gates

2. Implement NOR gate using NAND gates

- Draw the schematic Diagram
- Write the Boolean Expression
- Fill the Observation Table



Boolean Expression:

$$X = \overline{\overline{A} \cdot \overline{B}} \cdot \overline{\overline{A} \cdot \overline{B}}$$

as $\overline{A} \cdot \overline{A} = \overline{A}$

$$X = \overline{\overline{A} \cdot \overline{B}}$$

as $\overline{A} \cdot \overline{B} = \overline{A} \cdot \overline{B}$

$$X = \overline{\overline{A} \cdot \overline{B}}$$

as $\overline{A} = \overline{A}$

$$X = \overline{\overline{A} \cdot \overline{B}}$$

Demorgans law $\overline{\overline{A} \cdot \overline{B}} = \overline{A + B}$

$$X = \overline{A + B}$$

Input		Output	
A	B	LED (on / off)	Level (1 / 0)
0	0	On	1
0	1	Off	0
1	0	Off	0
1	1	Off	0

Table 2.4: Observation table for 2-input NOR gate using NAND gates



Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Lab # 04: Half Adder

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify Half adder operations using 74LS83 IC & Module KL-33001
- To implement and verify Full adder operations using 74LS83 IC & Module KL-33001

Lab Hardware and Software Required:

1. 74LS83 IC (4-bit Adder)
2. Module KL-33001
3. Breadboard
4. Connecting Wires

Background Theory:

Half Adder:

Many logic circuits must be supplied with devices which can carry out the sum between two numbers, for this purpose the adder circuit is used. The basic rules for binary addition are:

Binary Addition:

$$\begin{aligned}0 + 0 &= 0 \\0 + 1 &= 1 \\1 + 0 &= 1 \\1 + 1 &= 0 \text{ and carry } 1\end{aligned}$$

These operations are performed by a logic circuit called a half adder. A half adder is a binary adder which adds on two bits, it accepts two binary digits on its inputs and produces two binary digits on its output, a sum bit and a carry bit. From the logical operation of the half adder as expressed by basic binary addition rules, we observe that the sum output is 1 only if the input bits are not equal. The sum can therefore be expressed as the exclusive-OR of the input variables. Notice that the carry output is 1 only when input bits are 1, therefore carry can be expressed as the AND of the input variables. A half adder is represented by the logic symbol and logic diagram as shown in Fig 4.1 (a & b).

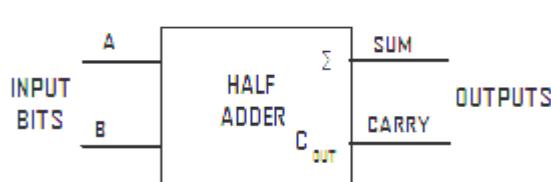
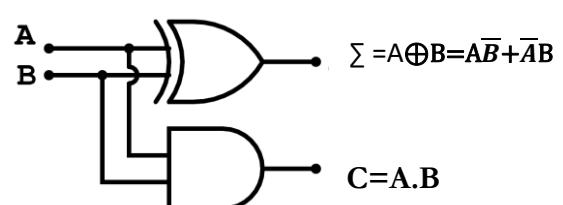


Fig 4.1: (a) Half Adder logic symbol



(b) Half Adder logic diagram

FULL ADDER:

The second basic category of adder is the full adder. The full adder accepts three inputs including an input carry and generates a sum output and an output carry. The basic difference between a full adder and half adder is that the full adder accepts an input carry. The full adder must add the two input bits and the input carry. From the half adder we know the sum of the bits A and B is the X-OR of those two variables A B. For the input carry (C_{in}) to be added to the input bits., it must be exclusive ORed of those variables AB , yielding the equation for the sum output of the full adder as $\Sigma = (A \oplus B) \oplus C_{in}$. This means that to implement the full adder sum functions, the X-OR gates can be used. The output carry is a 1 when both inputs to the second X-OR gate are 1. The equation for the carry output can be developed as $C_{out} = AB + (A \oplus B) C_{in}$. The logic symbol and logic diagram for a full adder are given in Fig 4.2(a & b).

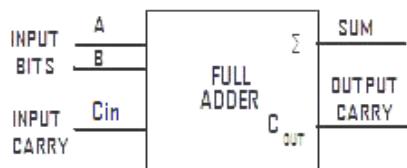
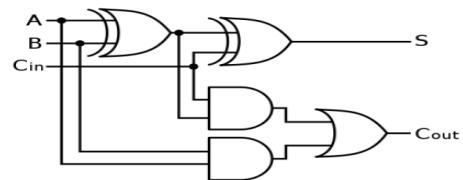


Fig 4.2: (a) Full Adder logic symbol



(b) Full Adder logic diagram

74LS83 4-bit ADDER IC:

4-bit adder IC present on the module is an integrate of TTL family. Fig 4.3 shows a pin diagram of TTL 74LS83 adder IC. This IC contains four full adders inside; carry out of each adder is connected to carry in of next full adder. The C_0 is basically a carry input for 4-bit full adder and C_4 is a carry out from 4-bit full adder. A_4, A_3, A_2, A_1 and B_4, B_3, B_2, B_1 are the inputs and S_4, S_3, S_2, S_1 are output of a 4-bit adder. This IC can be used as Half Adder, 2-bit, 3-bit and 4-bit full adder. It can be connected in cascade to other similar circuits, to form adders with more than 4-bits.

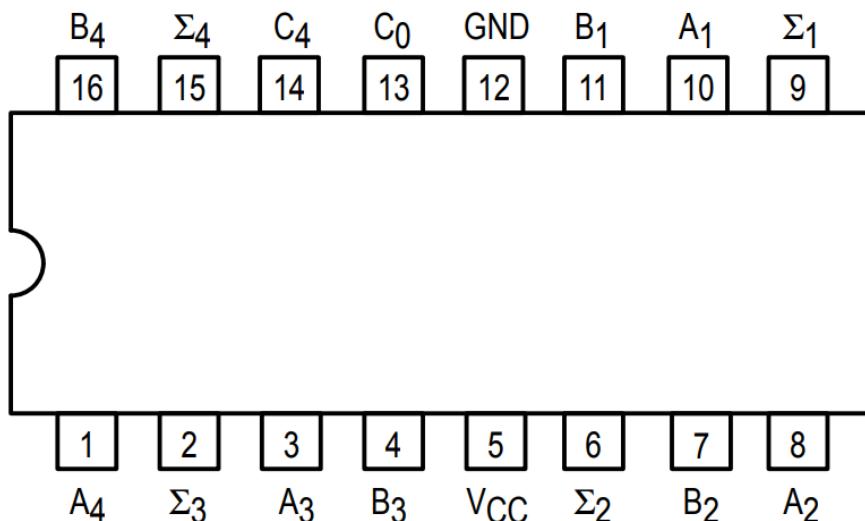


Fig 4.3: 74LS83 4-bit ADDER IC pin configuration

Lab Examples:

Implementation of Half Adder:

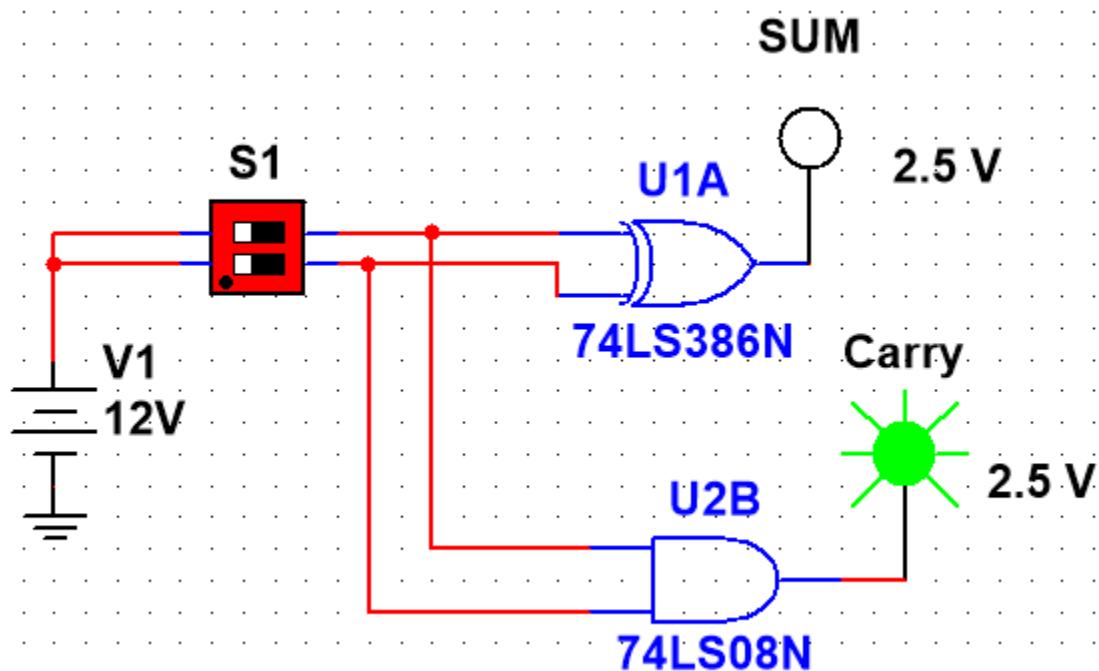
- Carry out the pin configuration of **74LS83** IC
- Take **74LS83** IC and insert it in the breadboard present on the **KL-33001** training kit
- Apply power supply to it
- Connect inputs A1 and B1 to the data switches SW0 and SW1 respectively
- Connect the output sum (Σ) and carry out (C0) to the LEDs
- Analyse the circuit behaviour by carrying out different combinations with the data switches and fill the observation table for half adder

Input		Output (Sum Σ)		Output (Carry Cout)	
A	B	LED (on / off)	Level (1 / 0)	LED (on / off)	Level (1 / 0)
0	0	Off	0	Off	0
0	1	On	1	Off	0
1	0	On	1	Off	0
1	1	Off	0	On	1

Table 4.1: Observation table of Half Adder

Lab Activities:

Half Adder Circuit





Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 05: Comparator

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify Comparator operations using 74LS85 IC & Module KL-33001

Lab Hardware and Software Required:

1. 74LS85IC (Comparator)
2. Module KL-33001
3. Breadboard
4. Connecting Wires

Background Theory:

Comparator:

Comparator is a combinational logic circuit that compares two input binary numbers and signals if the two numbers are equal, or if one is higher than the other. The basic function of comparator is to compare the magnitude of two quantities to determine the relationship of those quantities. A binary comparator usually has three outputs: $A=B$, $A>B$ and $A<B$. A 1-bit comparator compares two 1-bits (A & B) and determines whether they are “greater than” or “less than” or “equal to” each other by a high level on the appropriate output. As the outputs exclusively depend on the temporary inputs, the circuit is composed by logic gates. The logic symbol and logic diagram for 1-bit comparator as shown in Fig 5.1 (a & b).

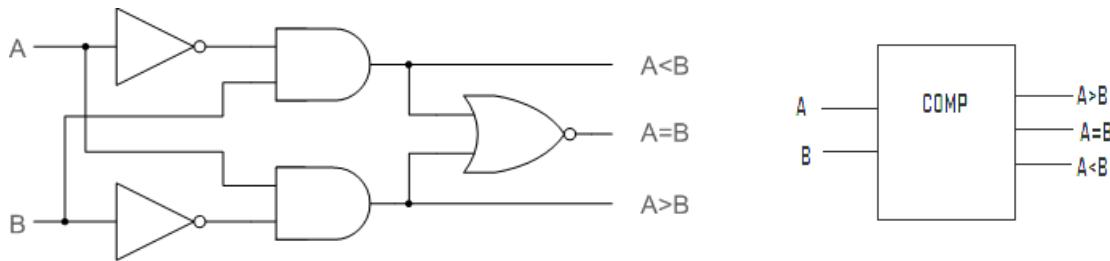


Fig 5.1: (a) Logic Symbol

(b) Logic symbol for 1-bit Comparator

74LS85 Comparator IC:

The TTL 74LS85 IC is a four bit magnitude comparator which will perform comparison of two 4-bit binary numbers. The pin configuration of IC is shown in Fig 5.2. This IC consists of eight comparing inputs ($A_0, A_1, A_2, A_3, B_0, B_1, B_2, B_3$), three cascading inputs ($I_{A<B}, I_{A=B}$ and $I_{A>B}$). This device compares two 4-bit words (A & B) and determines whether they are “greater than”, “less than” or “equal to” each other by a high level on the appropriate output. For words greater than 4-bits, units can be cascaded by connecting the outputs ($O_{A<B}, O_{A=B}$, and $O_{A>B}$) of the least significant stage to the cascaded inputs ($I_{A<B}, I_{A=B}$ and $I_{A>B}$) of the next significant stage. In addition the least significant stage must have a high level ($V_{in}(1)$) applied to the $A=B$ input and low level voltage ($V_{in}(0)$) applied to $A>B$ and $A<B$ inputs.

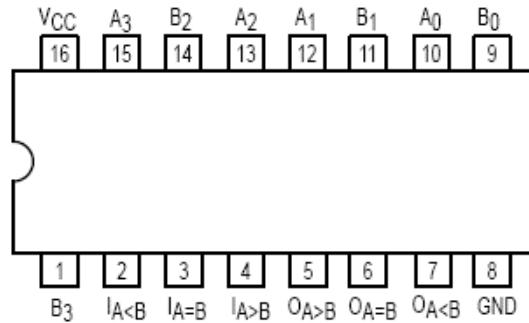


Fig 5.2: 74LS85 Magnitude Comparator IC pin configuration

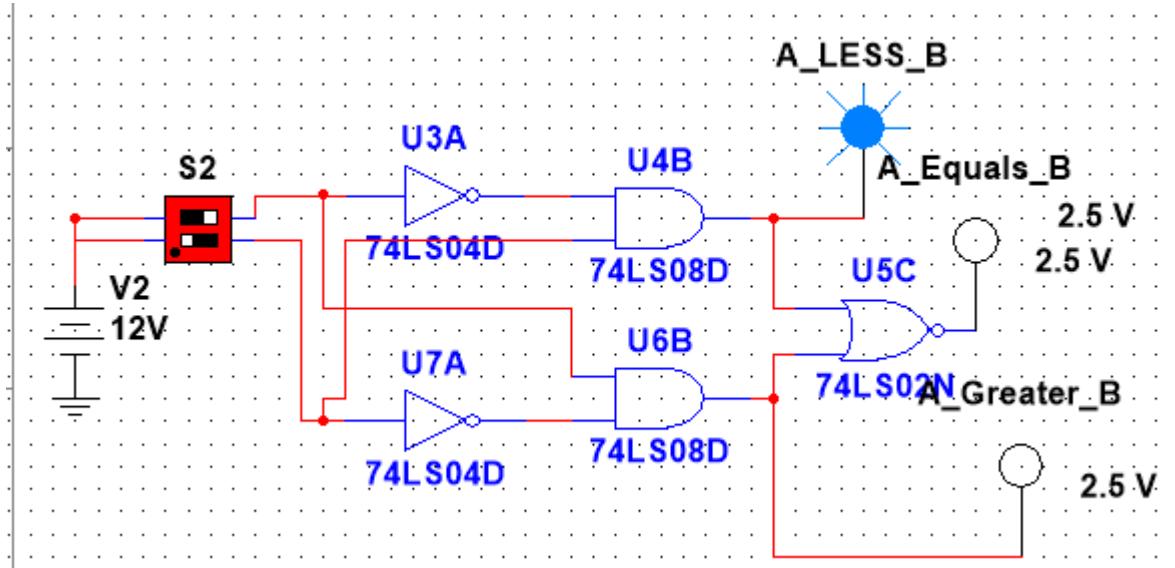
Lab Examples:

Implementation of 1-bit Comparator:

- Carry out the pin configuration of **74LS85** IC
- Take **74LS85** IC and insert it in the breadboard present on the **KL-33001** training kit
- Apply power supply to it
- Connect inputs A and B to the data switches SW0 and SW1 respectively
- Connect the three output to the LEDs
- Analyse the circuit behaviour by carrying out different combinations with the data switches and fill the observation table for comparator

Input		Output (A>B)		Output (A=B)		Output (A<B)	
A	B	LED (on / off)	Level (1 / 0)	LED (on / off)	Level (1 / 0)	LED (on / off)	Level (1 / 0)
0	0	Off	0	On	1	Off	0
0	1	Off	0	Off	0	On	1
1	0	On	1	Off	0	Off	0
1	1	Off	0	On	1	Off	0

Table 5.1: Observation table of 1-bit Comparator





Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 06: Decoder

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify Decoder operations using **74LS42** IC & Module KL-33001

Lab Hardware and Software Required:

1. 74LS42 IC (BCD-to-Decimal Decoder)
2. Module KL-33001
3. Breadboard
4. Connecting Wires

Background Theory:

Decoder:

Decoder is a logic circuit that accepts a set of inputs that represents a binary number and activates only the output that corresponds to that input number. In other words, a decoder circuit looks at its inputs, determines which number is present there, and activates one output that corresponds to that number; all other outputs remain inactive. Before the design of decoder, we must decide that whether we want an active high level output or an active low level output to indicate the value selected. For an active-high indication the required output is high and all other all other outputs are low; while in case of active-low, the required output is low and all other outputs are high. Fig 8.1 (a & b) shows the logic symbol and logic diagram for 2x 4 decoder. It uses all AND gates, so the outputs are active high. For a given input code, the only output that is active (high), is one corresponding to decimal equivalent of the binary input code.

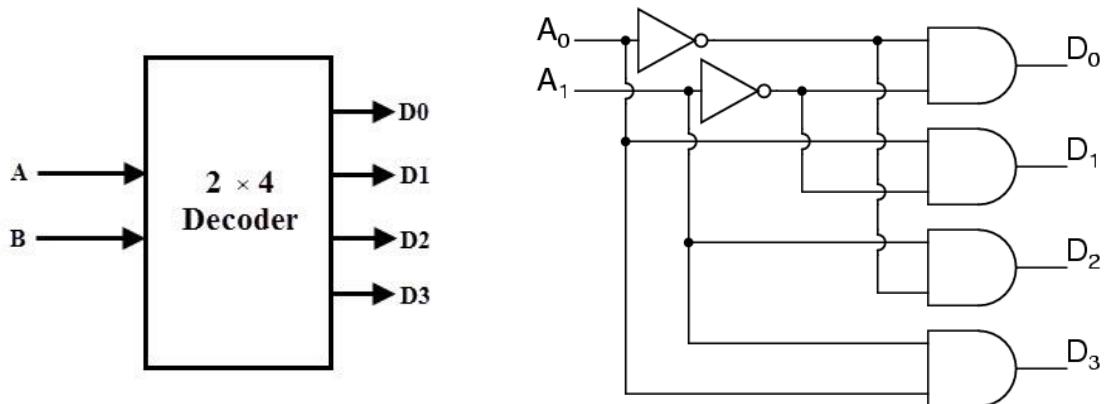


Fig 8.1: (a) Logic Symbol of 2 x 4 Decoder

(b) Logic Diagram of 2 x 4 Decoder

74LS42 Decoder IC:

The TTL 74LS42 accepts four lines of input data and it has ten active-low outputs. The inputs must be supplied by a coder type BCD, which causes the activation of the output line corresponding to the applied number. Each output goes low only when its corresponding BCD input is applied, while all other inputs remain high. Fig 8.2 shows the pin-out of 74LS42 IC.

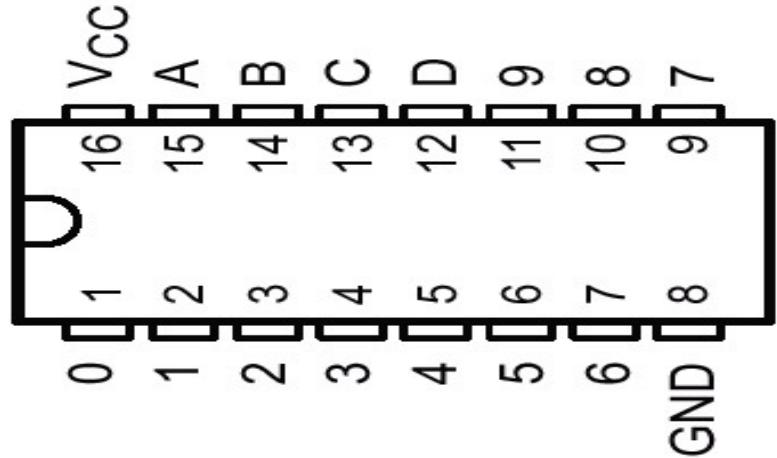


Fig 8.2: 74LS42 Decoder IC pin configuration

Lab Examples:

Implementation of 2 x 4 Decoder:

- Carry out the pin configuration of **74LS42** IC
- Take **74LS42** IC and insert it in the breadboard present on the **KL-33001** training kit
- Apply power supply to it
- Connect inputs A and B to two switches SW0 and SW1
- Connect the output Q0 to Q3 to the LEDs
- Set all the combinations of observation table with the help of switches and fill the observation table.

Input		Output (Y)			
B	A	Q0	Q1	Q2	Q3
		LED (on / off)			
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Table 8.1: Observation table of Decoder

Note: The output of the decoder (BCD to Decimal) is in negative logic, so a line results as active when the corresponding LED keeps off.

Lab Activities:

Implementation of BCD-to-Decimal Decoder using 74LS42:

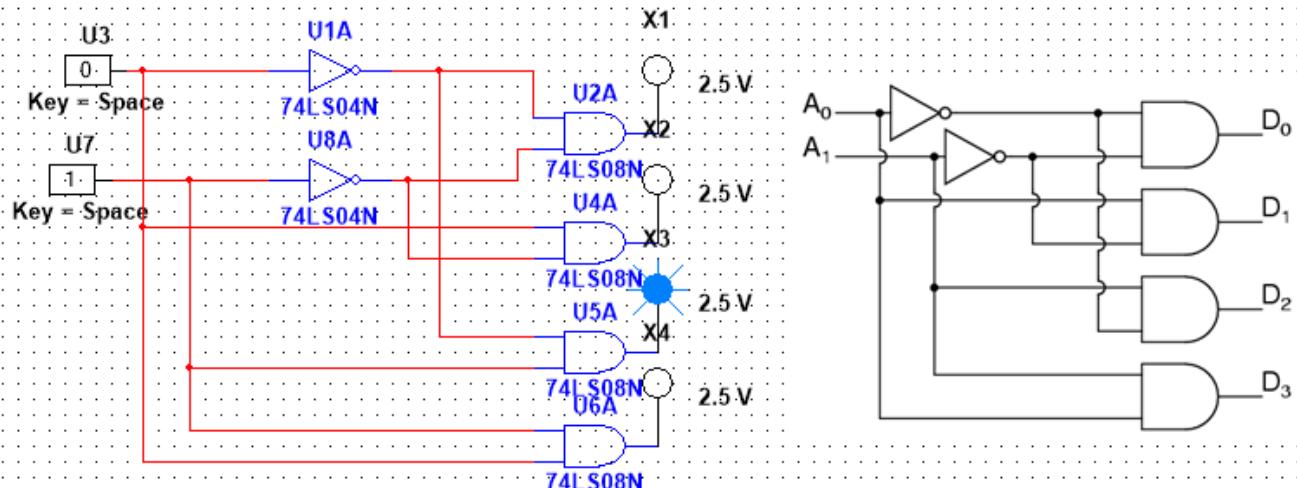
- Connect the inputs A, B, C and D to the switches
- Connect outputs 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 to the LEDs
- Compose all binary numbers from 0 to 9 and analyse the state of LEDs

Input				Output (Y)									
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1
1	0	1	0										
1	0	1	1										
1	1	0	0										
1	1	0	1										
1	1	1	0										
1	1	1	1										

Table 8.2: Observation table of BCD-to-Decimal Decoder

Lab Exercises:

Make BCD to decimal decoder using Multisim. **Input should be last digit of your CMS ID.**





Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 07: BCD-to-7-Segments Decoder

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify BCD-to-7 Segment Decoder operations using **74LS47** IC & Module KL-33001

Lab Hardware and Software Required:

1. 74LS47 IC (BCD-to-7 Segment Decoder)
2. Module KL-33001
3. Breadboard
4. Connecting Wires

Background Theory:

BCD-TO-7 SEGMENT DECODER:

Most digital equipment has some means for displaying information in a form that can be understood readily by user. This information is often numerical data but can be also alphanumeric (numbers and letters). One of the simplest and most popular methods for displaying numerical digits use a 7-segment configuration as shown in fig 9.1 (a) to form the decimal character 0 through 9 and sometimes the hexadecimal characters A through F. One common arrangement uses light emitting diodes (LEDs) for each segment. By controlling the current through each LED, some segment will lit up and others will be dark so that desired character pattern will be generated. Fig 9.1 (b) shows the segment patterns that are used to display the various digits. For example to display “6”, the segment A, C, D, E, F and G are made bright while B is dark.



Fig 9.1: (a) 7-Segment Arrangement (b) Active segment for each digit

A BCD-to-7 segment decoder is used to take a four bits BCD input and provide the output that will pass current through the appropriate segments to display the decimal digit. The logic for this decoder is more complicated than the logic of decoders because each output is active for more than one combination of input, for example the E segment must be activated for any of the digits 0, 2, 6 and 8, which means whenever any of the codes 0000, 0010, 0110 or 1000 occurs.

74LS47 BCD-TO-7 SEGMENT DECODER IC:

Fig 9.2 shows the pin-out for TTL 74LS47 IC. The 74LS47 accepts four lines of BCD input data, generates their complements internally and decodes the data with seven AND/OR gates having open collector outputs to drive indicator segments directly. RBI is active –low ripple bank input. BI/RBO is active low blanking input or ripple blanking output, A to G are active low segment outputs and A0-to-A3 are BCD inputs. The active low input terminal LT (lamp test) is

used to carry out a control of the display, turning all segments contemporarily ON, while the terminals RB (ripple blanking) are used to keep the display OFF, when they should display some non-significant 0.

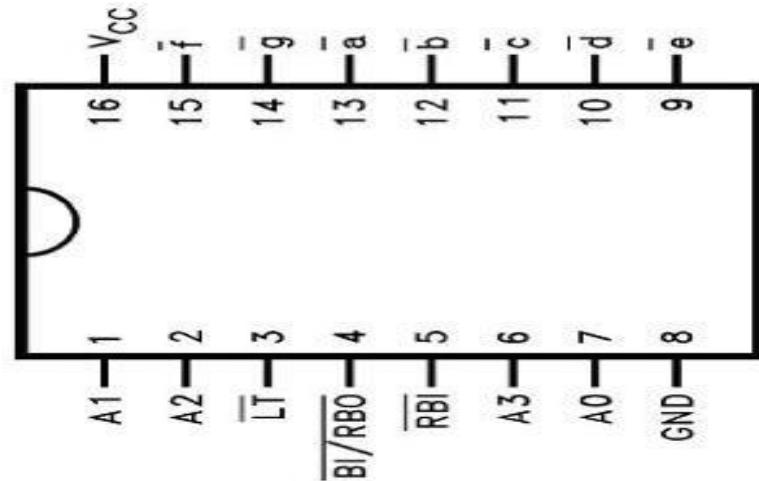


Fig 9.2: 74LS47 BCD-to-7 Segment Decoder IC

Fig 9.3 shows a TTL 74LS47 BCD-to-7 segment decoder being used to drive 7-segment common anode LED readout. Each segment consists of one or two LEDs. The anodes of the LEDs are all tied to VCC (+5V). The cathodes of LEDs are connected through current limiting resistors of value 330Ω to the appropriate output of the decoder. The value of limiting resistor can be found by knowing that the voltage drop across an LED is 1.7V and it takes approximately 10 mA to illuminate it. We need seven 330Ω resistors (eight if the decimal point is included) for current limiting. Dual-in-line package (DIP) resistor networks are available and simplify the wiring process because all seven (or eight) are in single DIP. The decoder has active low outputs.

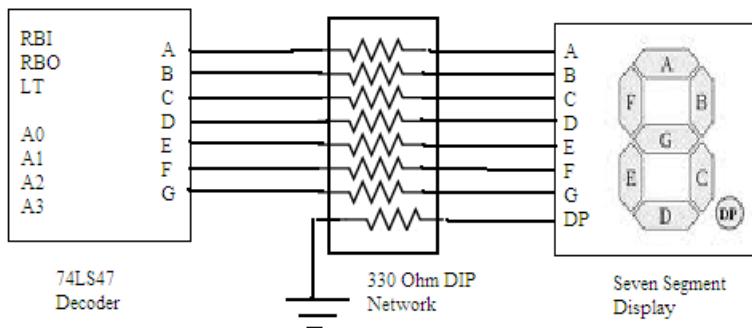


Fig 9.3: BCD to 7-segment decoder driving a common anode 7-segment LED display

To illustrate the operation of this circuit let us suppose that the BCD input is A3=0, A2=1, A1=0, A0=1, which is BCD for 5. With these inputs the decoder output A,F,G,C and D will be driven low (connected to ground) allowing the numeral 5. The B and E outputs will be HIGH (open), so that the LED segments B and E will not conduct.

Lab Examples:

Implementation of BCD-to-7-Segment decoder:

- Carry out the pin configuration of **74LS47** IC
- Take **74LS47** IC and insert it in the breadboard present on the **KL-33001** training kit

- Apply power supply to it
- Connect the inputs A0, A1, A2 and A3 of decoder to switches.
- Connect the outputs of decoder “A to G” to the corresponding inputs of LED display
- Set all combinations of inputs with the help of switches, check the LEDs and fill the observation table

Note: The **outputs of the decoder are negated logic**, so an LED ON means a LED of the 7-segment display off.

Input				Output						
A3	A2	A1	A0	A	B	C	D	E	F	G
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0

Table 9.1: Observation table of BCD-to-Segment decoder

Lab Activities:

Implement this circuit using Multisim and attach that circuit diagram showing last digit of your CMS ID.

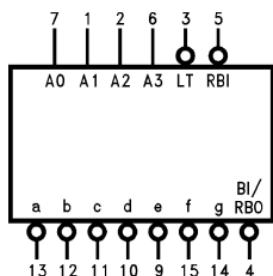
Lab Exercises:

Describe functions of LT, RBI, RBO and BI

Why do we use common anode seven segment display?

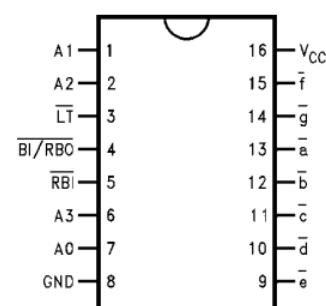
Draw pin configuration of seven segment common anode and common cathode display.

Logic Symbol



V_{CC} = Pin 16
GND = Pin 8

Connection Diagram



Pin Descriptions

Pin Names	Description
A0–A3	BCD Inputs
<u>RBI</u>	Ripple Blanking Input (Active LOW)
<u>LT</u>	Lamp Test Input (Active LOW)
<u>BI/RBO</u>	Blanking Input (Active LOW) or Ripple Blanking Output (Active LOW)
<u>a –g</u>	Segment Outputs (Active LOW) (Note 1)

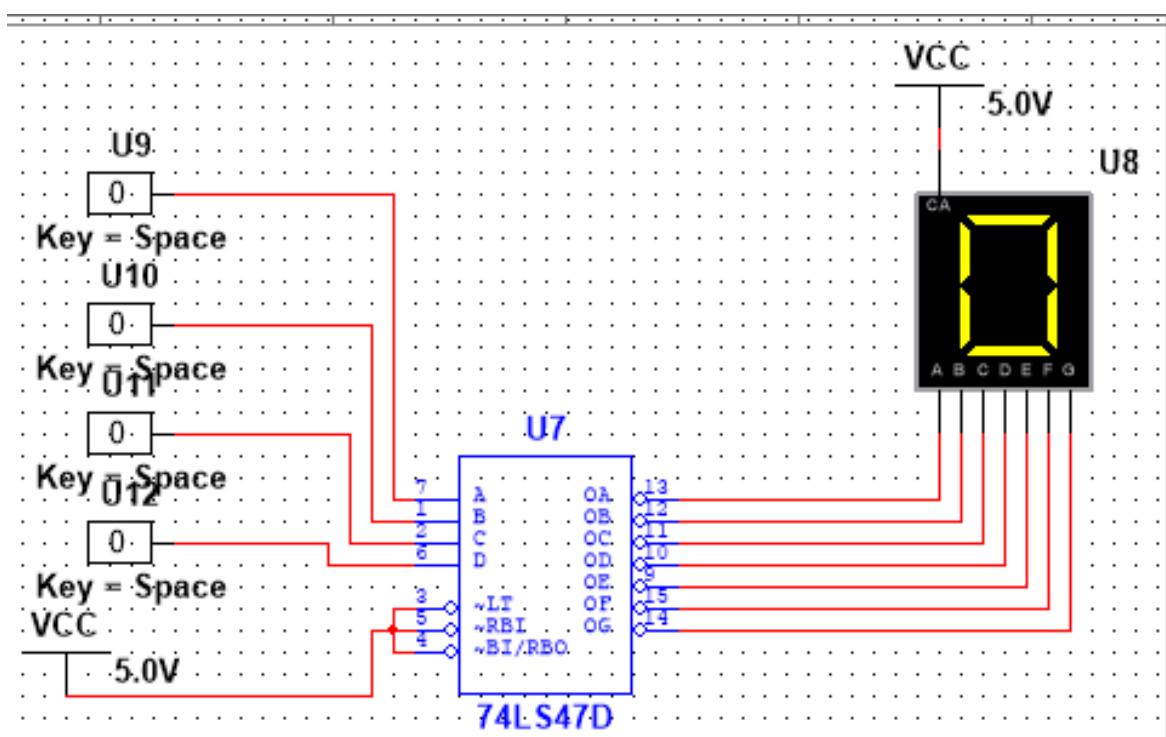
Note 1: OC—Open Collector

BI/RBO is wire-AND logic serving as blanking input (BI) and/or ripple-blanking output (RBO). The blanking out (BI) must be open or held at a HIGH level when output functions 0 through 15 are desired, and ripple-blanking input (RBI) must be open or at a HIGH level if blanking or a decimal 0 is not desired. X = input may be HIGH or LOW

When a LOW level is applied to the blanking input (forced condition) all segment outputs go to a HIGH level regardless of the state of any other input condition.

When ripple-blanking input (RBI) and inputs A0, A1, A2 and A3 are LOW level, with the lamp test input at HIGH level, all segment outputs go to a HIGH level and the ripple-blanking output (RBO) goes to a LOW level (response condition).

When the blanking input/ripple-blanking output (BI/RBO) is OPEN or held at a HIGH level, and a LOW level is applied to lamp test input, all segment outputs go to a LOW level.





Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 10: S-R Latch

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify S-R latch operations using two NOR or two NAND gates & Module KL-33001

Lab Hardware and Software Required:

1. 74LS02 IC
2. 74LS00 IC
3. Module KL-33001
4. Breadboard
5. Connecting Wires

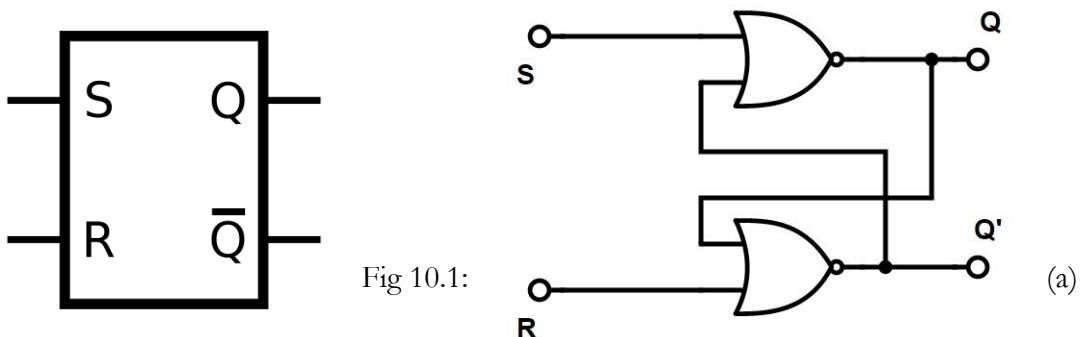
Background Theory:

Bi-stable Multivibrator:

The bi-stable multivibrators commonly called latches or latches, are the most common form of digital memory elements. A memory element is generally a device which can store the logic state 0 or 1, called information bit. The most basic latch circuit can be constructed from either two NOR gates or two NAND gates.

Using NOR Gates:

A main memory circuit can be carried out with the crossed coupling of two NOR gates, this type of connection is S-R latch as shown in fig 10.1 (a & b). The two NOR gates are cross coupled so that the output of NOR 1 gate is connected to one of the inputs of the NOR 2 gate and vice versa. The gates outputs Q and Q' respectively, are latch's outputs. Under the normal conditions, these outputs will always be the inverse of each other. There are two latch inputs: the set input (S) is the input that sets Q to the 1 state; the reset input (R) is the input that resets Q to the 0 state.



Simplified block diagram of S-R latch (b) NOR gate S-R latch

Suppose a data is to be inserted into the latch, the input levels are $S=R=0$. This is the normal resting state for the latch, and it has no effect on the output states. Q and \bar{Q} will remain in whatever state they were prior to the occurrence of this input condition. When $S=1$ and $R=0$, this will always set $Q=1$, where it will remain even after set returns to 0. $S=0$ and $R=1$ will always reset $Q=0$, where it will remain even after set returns to 0. $S=R=1$ condition tries to set and reset the latch at the same time, and it produces $Q=\bar{Q}=0$. If the inputs are returned to 0 simultaneously, the resulting output state is unpredictable. This input condition should not be used.

Using NAND Gates:

Two cross coupled NAND gates can be used to construct a NAND gate latch as shown in Fig: 10.2 (a & b). The arrangement 10.2 (b) is similar to NOR gate latch except that the Q and outputs have reversed positions.

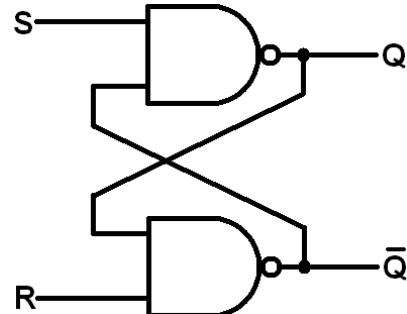
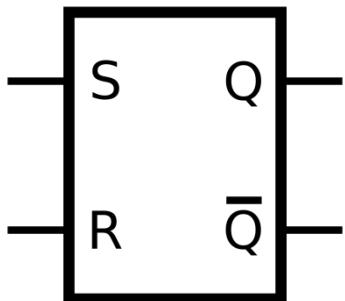


Fig 10.2: (a)

Simplified block diagram of S-R-latch (b) NAND gate latch

The analysis and the operation of the NAND gate latch can be performed is exactly the same manner as used for the NOR gate latch.

Lab Examples:

Implementation of S-R latch using NOR gates:

- Carry out the circuits of Fig 10.1 by using IC of NOR gates
- Make the circuit on breadboard present on the **KL-33001** training kit
- Apply power supply to it
- Connect the inputs S and R of both circuits to the switches
- Connect the outputs Q and \bar{Q} of both circuits to the LEDs
- Analyse the circuit's behaviour by carrying out different combinations with the switches and fill the observation tables

Inputs		Outputs	
S	R	Q	\bar{Q}
0	0	No Change	No Change
0	1	0	1
1	0	1	0
1	1	Invalid	Invalid

Table 10.1: Observation table of S-R latch using NOR gates

Lab Activities:

Implementation of S-R latch using NAND gates:

- Carry out the circuits of Fig 10.2 by using IC of NAND gates
- Make the circuit on breadboard present on the **KL-33001** training kit

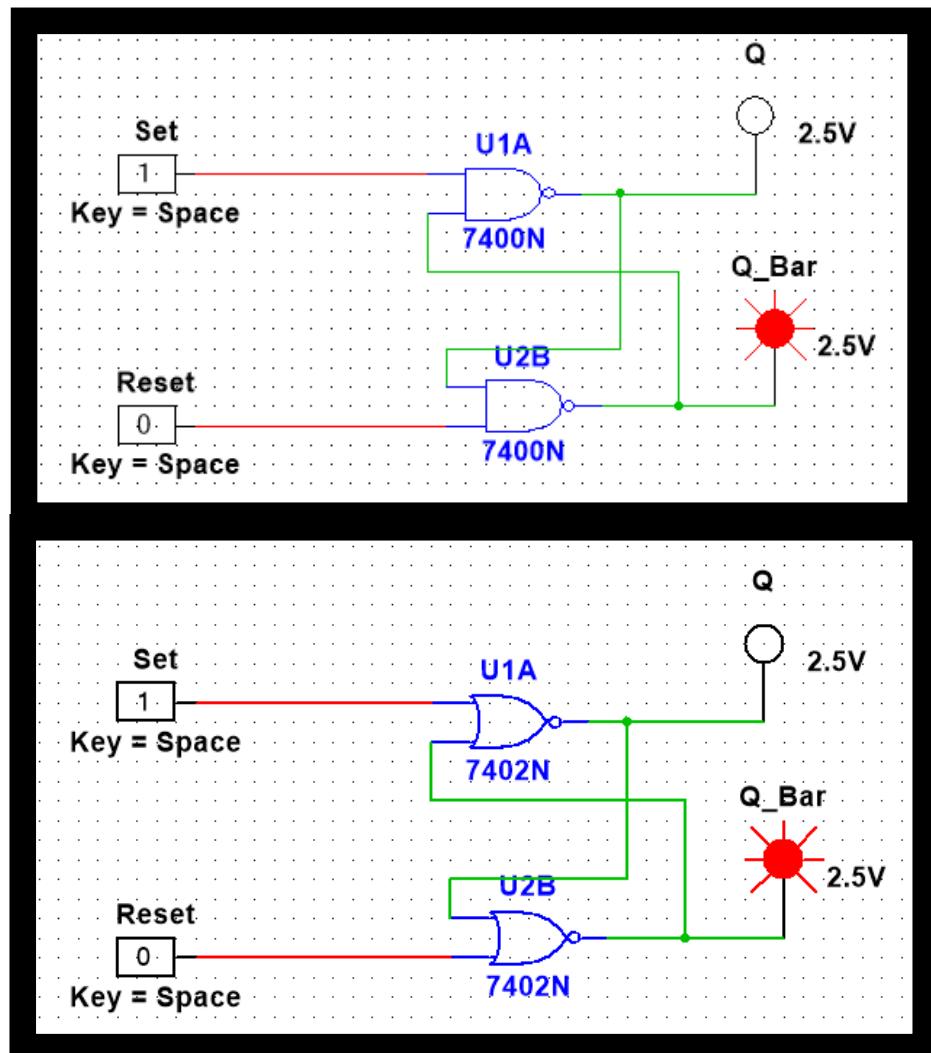
- Apply power supply to it
- Connect the inputs S and R of both circuits to the switches
- Connect the outputs Q and Q-of both circuits to the LEDs
- Analyse the circuit's behaviour by carrying out different combinations with the switches and filling the observation tables

Inputs		Outputs	
S	R	Q	Q
0	0	Invalid	Invalid
0	1	1	0
1	0	0	1
1	1	No Change	No Change

Table 10.2: Observation table of SR latch using NAND gates

Lab Exercises:

Implement S-R latch circuit in Multisim, attach circuit diagrams showing set and reset states.





Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 11: Clocked SR Latch

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify clocked RS latch operations using two NOR or two AND gates with clock & Module KL-33001

Lab Hardware and Software Required:

1. 74LS02 IC
2. 74LS08 IC
3. Module KL-33001
4. Breadboard
5. Connecting Wires

Background Theory:

Clocked RS latch:

In RS latch we saw that the output was changing by changing both inputs. In sequential systems, the change of state in the latch is often required to occur in synchronism with the clock pulse. Now the idea is that, a third input is introduced i.e. clock input (CK), which controls the output of the circuit. This is carried out by modifying the RS latch into the one of fig 11.1. In this case output will change only by changing the clock input.

This clock signal is generally a rectangular pulse train or a square wave. In most clocked latches the clock input is edge-triggered, which means that it is activated by a signal transition. This is indicated by the presence of a small triangle on the CK input. When clock changes from 0 to 1, this is called positive-going transition (PGT); when the clock changes from 1 to 0, this is called negative-going transition (NGT) of clock pulse. NGT is symbolized by presence of a small bubble on the clock input.

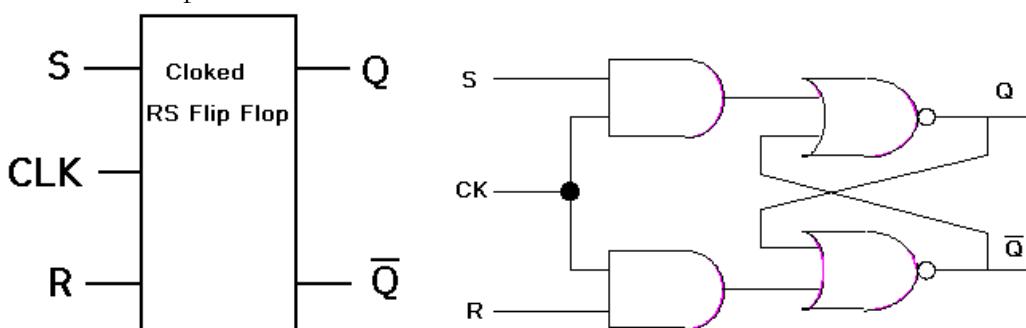


Fig 11.1: (a) Simplified block diagram of Clocked RS-latch (b) Clocked RS Latch

When the clock input is low ($CK=0$), the latch will not change its state and there would be no effect on the output Q. When clock goes high ($(CK=1)$), then depending on the states of R and S inputs, the latch can set or reset or there can be no change in state.

Lab Examples:

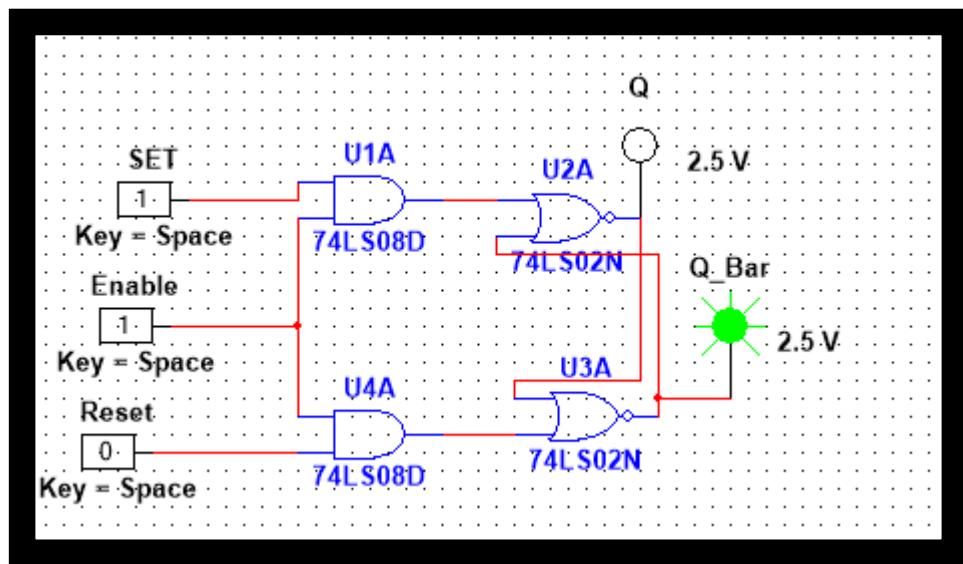
Implementation of clocked RS latch using NOR & AND gates:

- Carry out the circuits of Fig 11.1 by using ICs of NOR and AND gates
- Make the circuit on breadboard present on the **KL-33001** training kit
- Apply power supply to it

- Connect the inputs S and R of both circuits to the switches
- Connect the terminal of clock to the input CK of the latch
- Connect the outputs Q and \bar{Q} of both circuits to the LEDs
- Analyze the circuit's behaviour by carrying out different combinations with the switches and fill the observation tables

Inputs			Outputs	
CK	R	S	Q	\bar{Q}
1	0	0	0	0
1	0	1	0	1
1	1	0	0	0
1	1	1	0	0

Table 11.1: Observation table of clocked SR latch

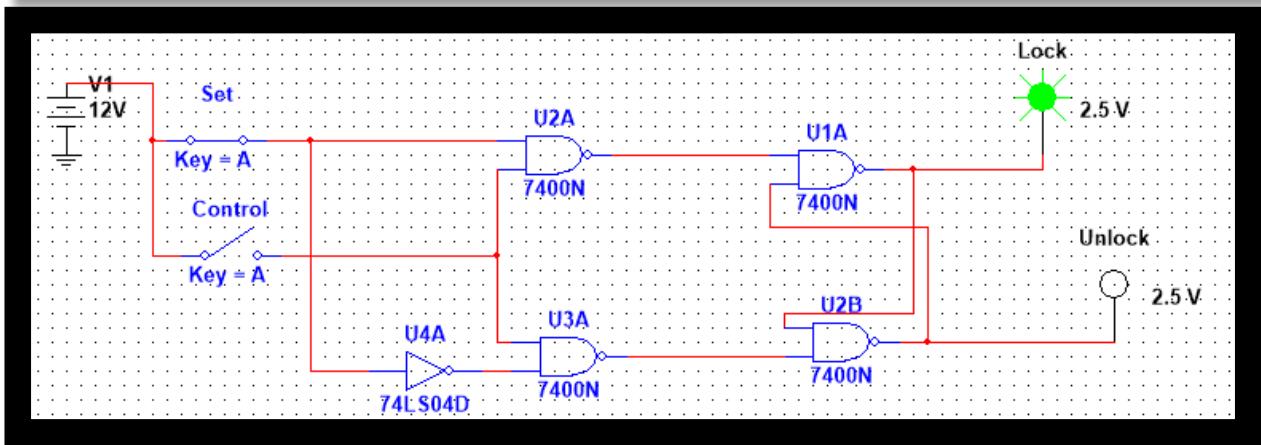
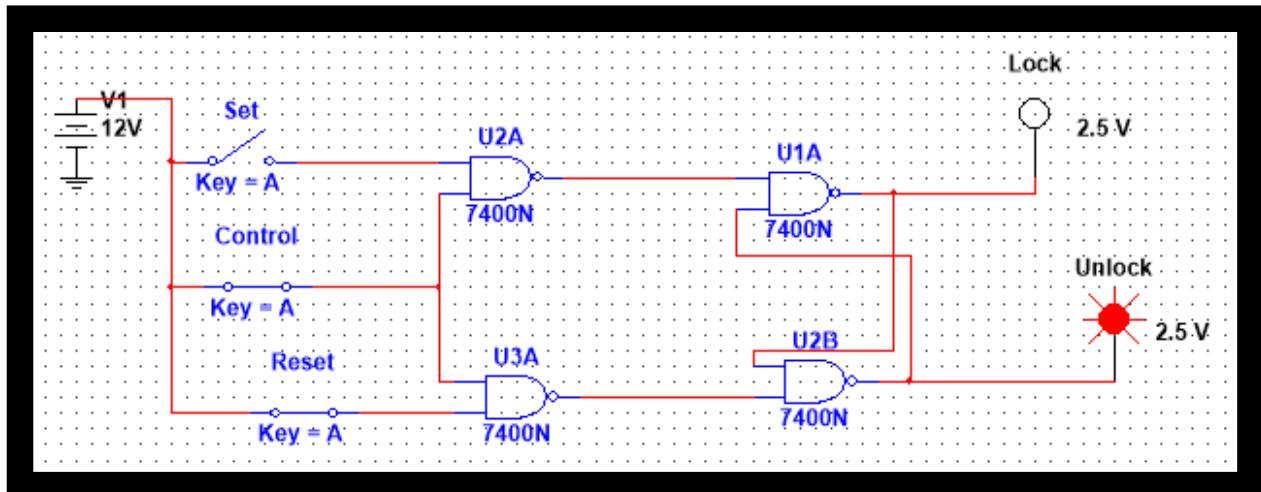


Lab Activities:

Change gated S-R latch to gated D latch and draw circuit diagram as well as truth table.

Lab Exercises:

Implement Gated S-R and gated D- latch in Multisim and attach circuit diagram showing set and reset states respectively.





Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 12: D Flip Flop

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify D flip flop operations using **74LS75 IC** & Module KL-33001

Lab Hardware and Software Required:

1. 74LS75 IC
2. Module KL-33001
3. Breadboard
4. Connecting Wires

Background Theory:

D-FLIP FLOP:

A clocked RS flip flop can be used to store a bit, but it needs proper logical states, and S and R inputs. In other words to store a bit, data must be supplied to on two lines via R and S. Another difficulty is that, whenever both R and S become high, an uncertain condition about the output state can occur. Such clocked flip flop can be modified to better version called as D-type flip flop. This flip flop possesses only one data input. Fig 12.1 (a & b) shows the logical symbol and logic diagram of flip flop.

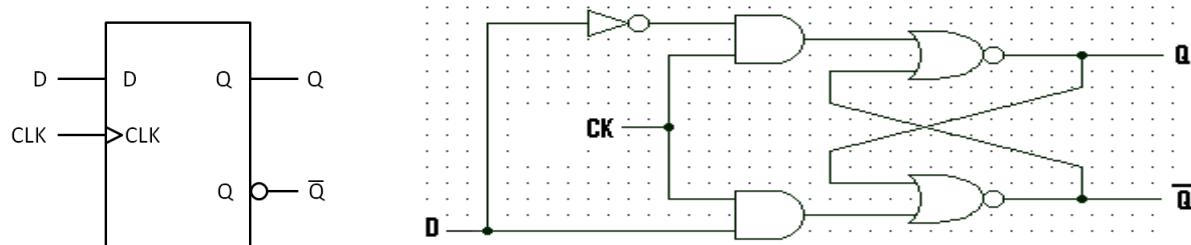


Fig 12.1: (a) logic symbol of D-flip flop

(b) logic diagram of D-flip flop

As seen from the Fig 12.1, an inverter is introduced between S and R inputs. With this arrangement, S and R can never become 1 at the same time. Thus the forbidden condition can be avoided. The operation is such that Q will be same as D while CK is high, and Q will remain latched when CK goes low(latched means that Q will remain constant regardless of changes in D).

74LS75 D Latch IC:

The 74LS75 is an example of an integrated circuit D-latch (also called a bi-stable latch). It contains four transparent D latches. Its pin configuration is given in Fig 12.2. Latches 1 and 2 share a common enable (E1-2), and latches 3 and 4 share a common enable (E3-4). The logic function developed by a circuit type latch is the one to transfer the logic value present across the input when the ENABLE signal is high logic value to its output, while the output signal keeps the same if the ENABLE signal is low.

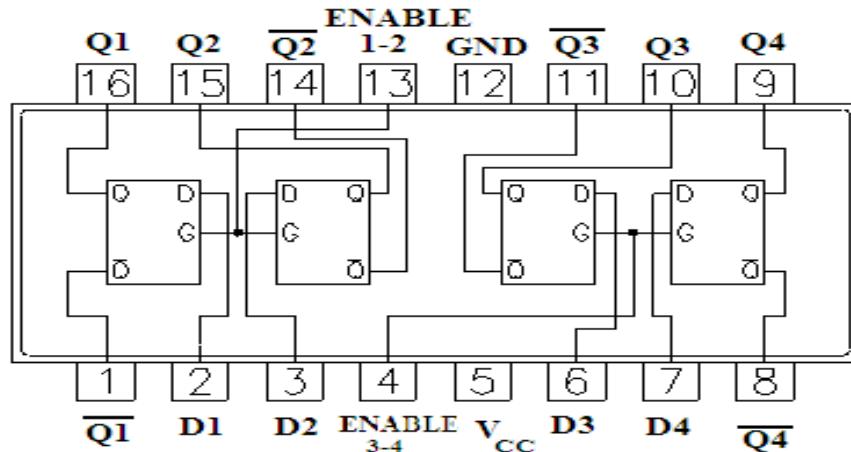


Fig 12.2: 74LS75 Pin Configuration

Lab Examples:

Implementation of D flip flop using 74LS75 IC:

- Carry out the circuits of Fig 12.1 by using IC 74LS75
- Make the circuit on breadboard present on the **KL-33001** training kit
- Apply power supply to it
- Connect the input D of circuit to the switch
- Connect the terminal of clock to the input CK of the flip flop
- Connect the outputs Q and \bar{Q} of both circuits to the LEDs
- Analyze the circuit's behaviour by giving clock signal and 0,1 data through the switch and fill the observation tables

Inputs		Outputs	
CK	D	Q	\bar{Q}
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

Table 12.1: Observation table of D flip flop

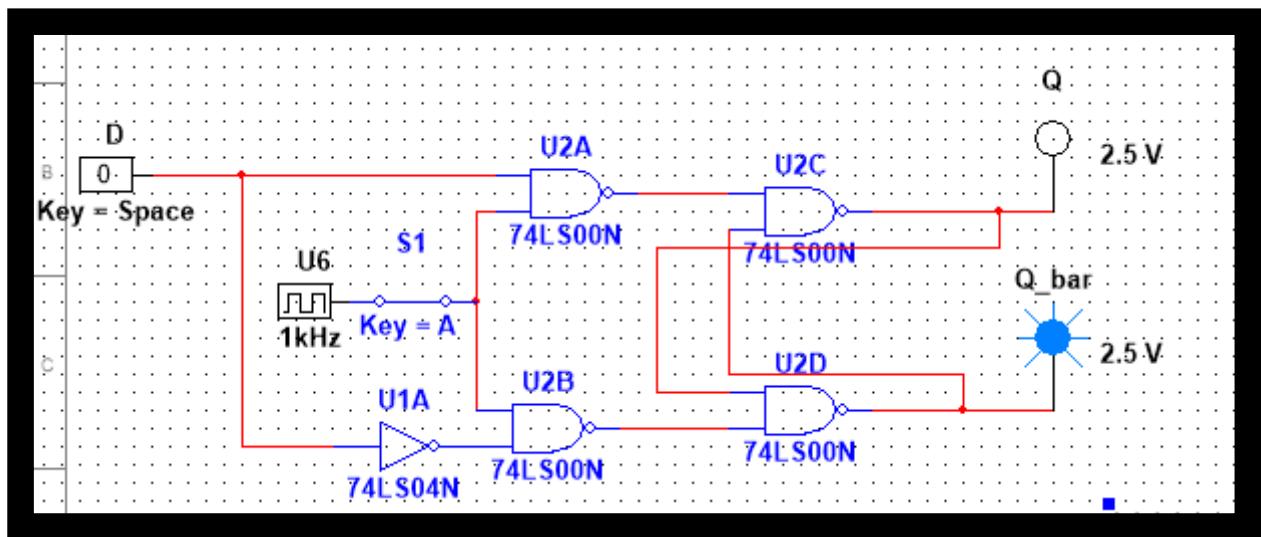
Lab Activities:

Analysis of 74LS75 IC:

- Connect the signals D and EN of one of the two latches shown in BISTABLE LATCH clock the two switches and Q & \bar{Q} to two LEDs
- Vary the state of the inputs and observe the circuit operation

Lab Exercises:

Implement D flip-flop in Multisim, attach circuit diagram and observed waveforms.





Sukkur Institute of Business Administration University

Department of Computer Science

ESE-201: Digital Logic Design Lab

Lab # 13: JK-Flip Flop

Submission Profile

Name: Sagar Chhabriya

Submission date (14/12/23):

CMS ID: 023-22-0310

Marks obtained:

Comments:

Receiving authority name and signature:

Instructor Signature

Note: Submit this lab hand-out in the next lab with attached solved activities and exercises

Lab Learning Objectives:

Upon successful completion of this experiment, the student will be able:

- To implement and verify JK flip flop operations using **74LS76 IC** & Module KL-33001

Lab Hardware and Software Required:

1. 74LS76 IC
2. Module KL-33001
3. Breadboard
4. Connecting Wires

Background Theory:

JK-FLIP FLOP:

The JK-flip flop is very versatile and is perhaps the most widely used type of flip flops. The J and K designation for the inputs have no known significance except that they are adjacent letters in the alphabet. The functioning of the JK flip flop is identical to that of RS flip flops in the SET, RESET and no change condition of operation. The difference is that the JK flip flop has no invalid state as does the RS flip flop. Therefore the JK flip flop is a very versatile device that finds application in digital system. Fig 13.1 (a & b) shows a logic symbol and logic diagram for the JK flip flop.

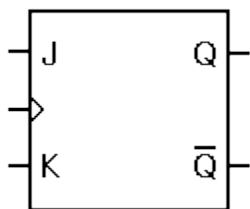
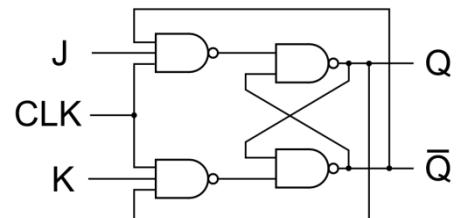


Fig 13.1: (a) logic symbol of JK-flipflop



(b) logic diagram of JK-flip flop

The data at the J and K inputs of the JK flip flop controls its output. Notice that it differs from the RS flip flop in the output of the flip flop is coupled back to input. J and K inputs control the state of the flip flop in the same way as the S and R inputs do for the clocked RS flip flop except for one major difference; the $J=K=1$ condition does not result in ambiguous output. For this 1, 1 condition, the flip flop always go to its opposite state upon the arrival of the clock signal. This is called the toggle mode of states (toggle) for each of the clock pulse.

74LS76 JK Flip Flop IC:

The 74LS76 is a popular JK flip flop IC; having dual flip flops (two flip flops in each IC package). Its pin configuration is given in Fig 13.2. It has only asynchronous inputs (RD and SD) as well as synchronous inputs (CP, J, K) and is negative-edge triggered flip flop. The asynchronous inputs RD and SD are active low, that is low on SD (set) will set the flip flop ($Q=1$) and low on RD will reset the flip flop ($Q=0$). The asynchronous input will cause the flip flop to respond immediately without regard to the clock trigger input CP. For synchronous operation using J,K, and CP, the asynchronous inputs must be disabled by putting a high level on both.

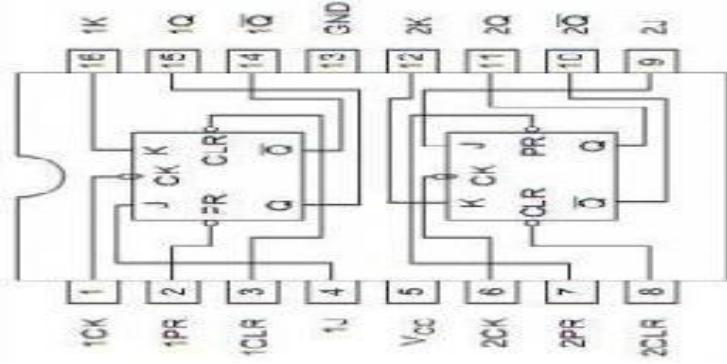


Fig 13.2 pin configuration of 74LS76 JK flip-flop IC

Lab Examples:

Implementation of JK flip flop using 74LS76 IC:

- Carry out the circuits of Fig 13.2 by using IC 74LS76
- Make the circuit on breadboard present on the **KL-33001** training kit
- Apply power supply to it
- Connect the inputs J & K of circuit to the switches
- Connect the terminal of clock to the input CK of the flip flop
- Connect the outputs Q and \bar{Q} of both circuits to the LEDs
- Analyze the circuit's behaviour by carrying out different combinations with the switches and fill the observation tables

Inputs			Outputs	
CK	K	J	Q	\bar{Q}
0	X	X		
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

Table 13.1: Observation table of JK flip flop

Lab Activities:

How can we convert J K flip into D flip flop?

Lab Exercises:

Implement JK flip flop in Multisim, attach circuit diagram and observed waveforms.

