Computer Architecture

(OrCAD)

Laboratory File

**Submitted By:**

**Digital Circuits**

**Procedure for simulation of circuits on ORCAD capture tool:**

1. Click the capture lite icon from desktop or from the start menu.
2. Click on file, then on new, and then click on project.
3. Write the name of the project and choose the location to store it, then click “OK” and make sure to choose analog or mixed A/D option and then again click “OK”.
4. Next pop-up window opens Pspice project on this choose a blank project option, then click “OK”.
5. Now realize the circuits using proper components.
6. Now choose Pspice option on top toolbar and click on new simulation profile.
7. Enter the name of the profile and then click on “OK”.
8. In the next window choose options and then choose gate level simulation. Choose the option and initialize all flip-flops to zero, then click on “OK”.

V place voltage where we need to see the outputs.

1. Lastly click the “Run” button in Pspice or Run option to see the resulting output.

**Analog Circuits**

**Procedure for simulation of circuits on ORCAD capture tool:**

**The use of resistors and capacitors:-**

1. Click on “place part” on the right hand side of the window.
2. Then click on analog library and select the required resistor and capacitor.

**The use of Transistor:-**

1. Click on “place part” on the right hand side of the window.
2. Click on the EVAL library and select the required transistor (E.G. Q2N2222).

**The use of source:-**

1. Click on “place part” on the right hand side of the window.
2. Then click on source library.

**The use of ground:-**

1. Click on ‘GND’ on the right hand side of the window.
2. Select CAPSYM library.
3. Select GND.
4. Click OK.
5. Edit properties to set “Name=0”.

**To select the arrow in op-amp as a filter:-**

1. Select the ‘(PWR/Q)’ from the right hand side of the window.
2. Select ‘Vcc\_arrow’.
3. Name = V+ or V-
4. While using a dc source always use a DC source.

**To change the value of any component:-**

1. Select the component.
2. Right click on it.
3. Click “edit properties “ .
4. Change the values of desired components.

**To change the name of any component:-**

1. Select NI on the right hand side of the window (not alias).
2. Change name.
3. Right click on it
4. Edit properties and click “OK”.
5. Box with the required name will be placed.

**To change the direction of any of selected component:-**

1. Place it on the required place and right click on it.
2. Select rotate and it will rotate by 90 degrees.
3. Select it as many times as rotation is required.
4. After placing all the components join them through the wire and you will obtain the required circuit.

**For Output:-**

1. Click Pspice simulation.
2. Select new simulation.
3. Select analysis type which can be different for different experiments.

INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| Sno. | **Experiment Name** | **Date** | **Remarks** |
| 1. | To design the circuit of half adder. |  |  |
| 2. | To design the circuit of full adder. |  |  |
| 3. | To design the circuit of half subtractor. |  |  |
| 4. | To design the circuit of full subtractor. |  |  |
| 5. | To design an 8×1 Multiplexer. |  |  |
| 6. | To design a 4 bit combinational shifter. |  |  |
| 7. | To Design a 3:8 Decoder |  |  |
| 8. | . To design an ALU. |  |  |
| 9. | To design a 4-bit adder subtractor |  |  |
| 10 | To design a BCD adder. |  |  |

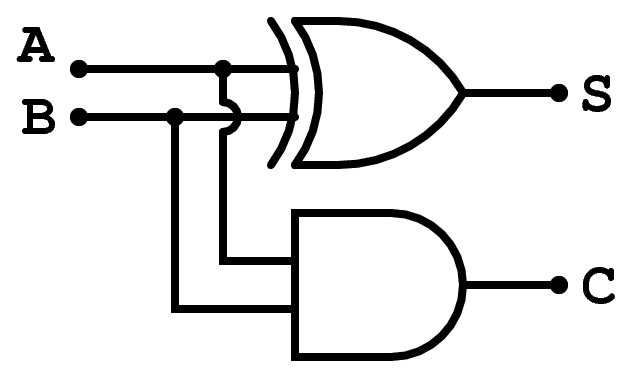
**EXPERIMENT NO.1**

AIM: To design the circuit of half adder.

IC USED: 7486(X-OR), 7408(AND).

THEORY: A half adder is a logical circuit that performs an additional operation on two binary digits. The half adder produces a sum and a carry value which are both binary digits.

A half adder circuit has two inputs A and B and two outputs – S representing sum and C representing carry.



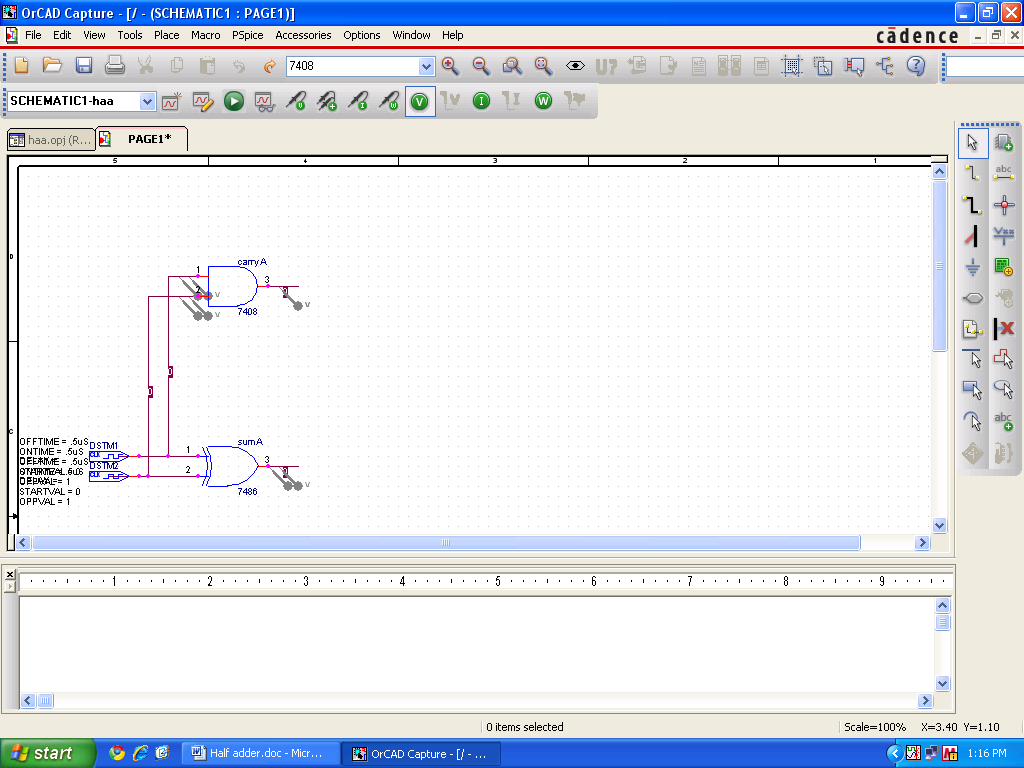
S = A xor B i.e. (A’B + AB’)

C = A and B i.e. (A.B)

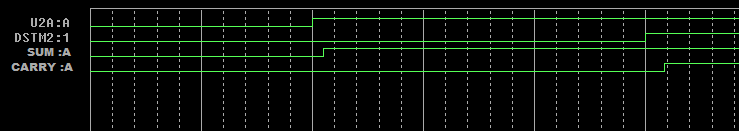
TRUTH TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **S** | **C** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

SCHEMATIC DIAGRAM:



WAVEFORM:



RESULT: The output waveform of half adder is verified.

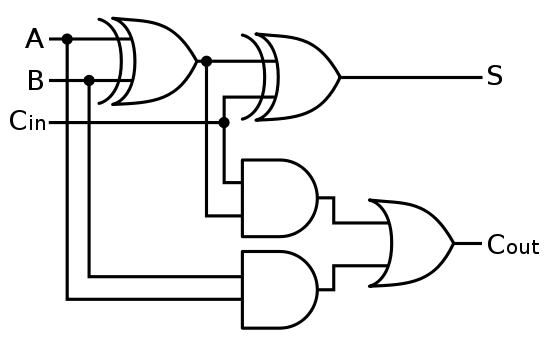
**EXPERIMENT NO.2**

AIM: To design the circuit of full adder.

IC USED: 7486(X-OR), 7408(AND), 7432(OR).

THEORY: A full adder is a logical circuit that performs an additional operation on three binary digits. The half adder produces a sum and a carry value which are both binary digits.

A full adder circuit has three inputs A,B and Cin and two outputs – S representing sum and Cout representing carry.



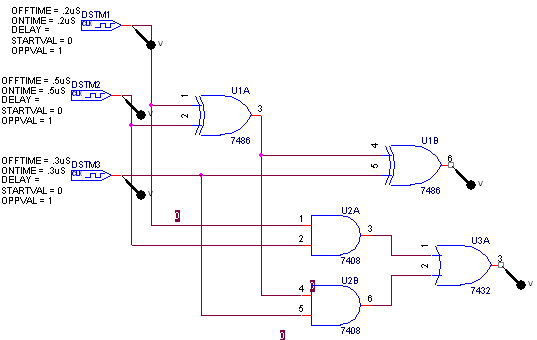
S = A xor B xor C

C = A.B +C(A xor B)

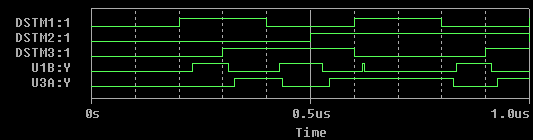
TRUTH TABLE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **Cin** | **S** | **Cout** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

SCHEMATIC DIAGRAM:



WAVEFORM:



RESULT: The output waveform of half adder and Full Adder is verified.

**EXPERIMENT NO.3**

AIM: To design the circuit of half subtractor.

IC USED: 7486(X-OR), 7408(AND), 7404(NOT).

THEORY: A half subtractor is a logical circuit that performs an subtraction operation on two binary digits. The half subtractor produces a Difference and a borrow value which are both binary digits.

A half subtractor circuit has two inputs X, Y and two outputs – D representing difference and B representing borrow.



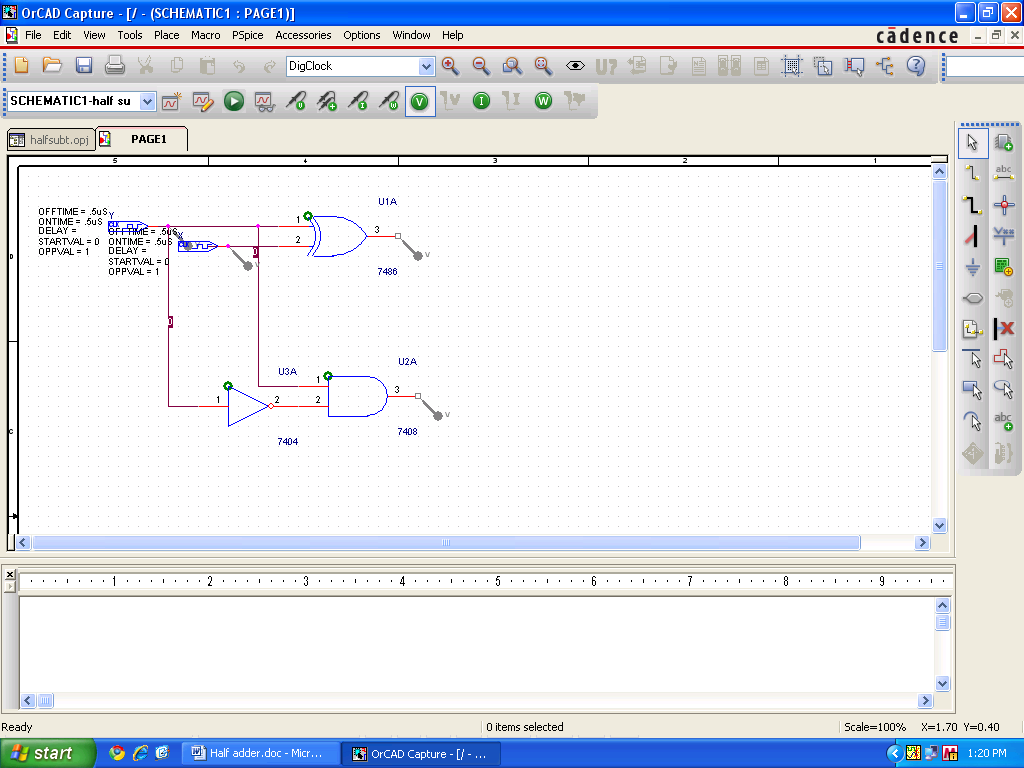
D = A xor B i.e. (A’B + AB’)

B = A’B

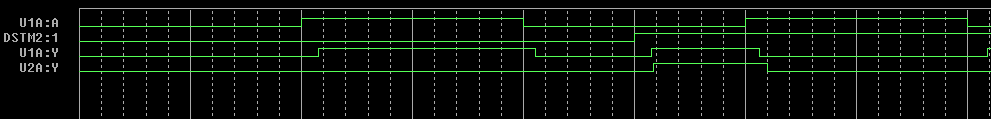
TRUTH TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **Y** | **D** | **B** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

SCHEMATIC DIAGRAM:



WAVEFORM:



RESULT: The output waveform of half subtractor is verified.

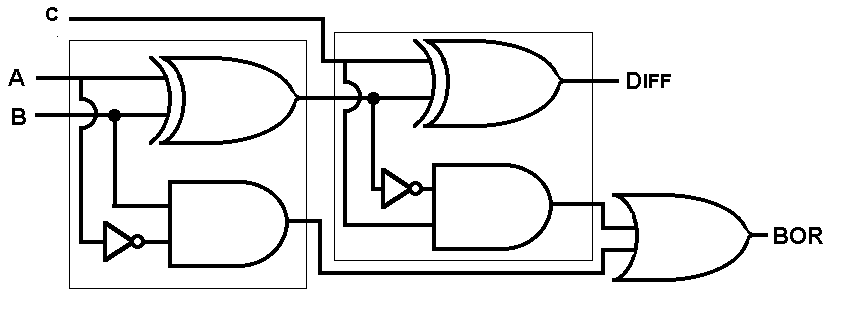
**EXPERIMENT NO.4**

AIM: To design the circuit of full subtractor.

IC USED: 7486(X-OR), 7408(AND), 7432(OR),7404(NOT).

THEORY: A full subtractor is a logical circuit that performs an subtraction operation on three binary digits. The full subtractor produces a difference and a borrow value which are both binary digits.

A Full adder circuit has three inputs A,B and C and two outputs – DIFF representing difference and BOR representing borrow.



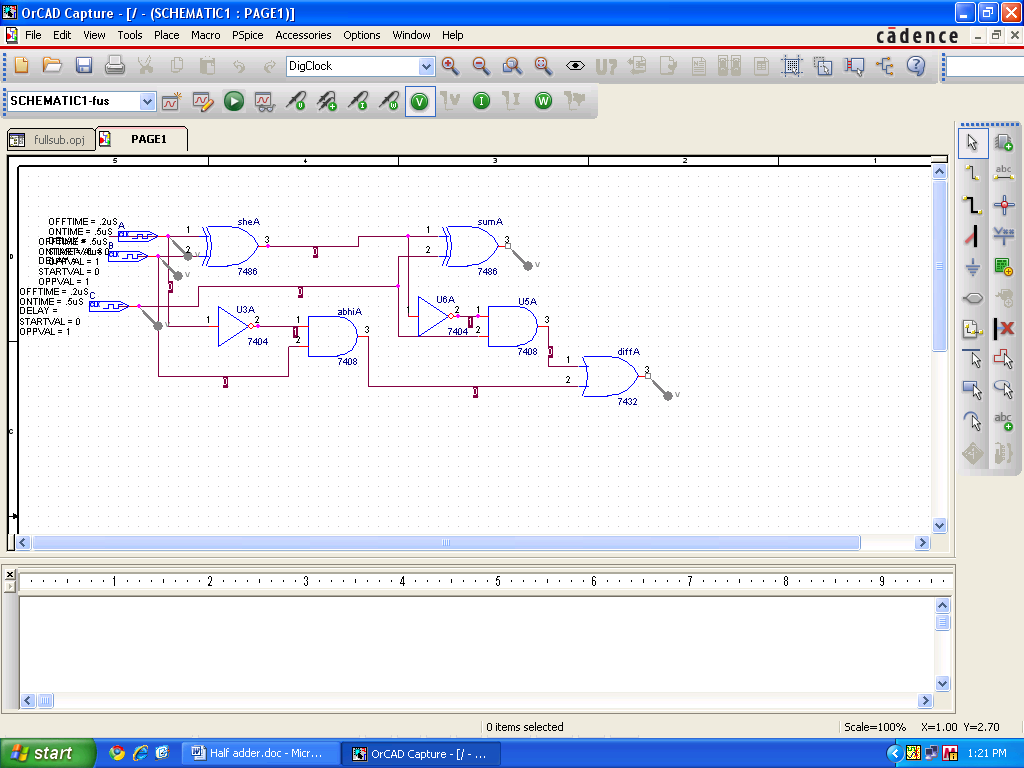
S = A xor B xor C

C = A’.B +C(A xnor B)

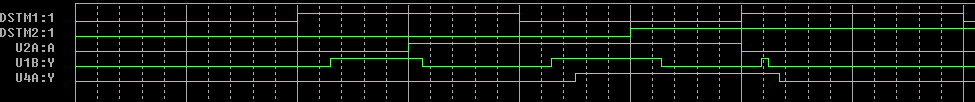
TRUTH TABLE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **DIFF** | **BOR** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

SCHEMATIC DIAGRAM:



WAVEFORM:



RESULT: The output waveform of full subtractor is verified.

**EXPERIMENT NO.5**

AIM: To design an 8×1 Multiplexer.

IC USED: 7486(X-OR), 7408(AND), 7432(OR), 7404(NOT), 74151A (8×1 MUX).

THEORY: A multiplexer is a device that performs multiplexing i.e. it selects one of many analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2n inputs has n select lines, which are used to select which input line to be sent to the output.

A Boolean equation for 8×1 Multiplexer is

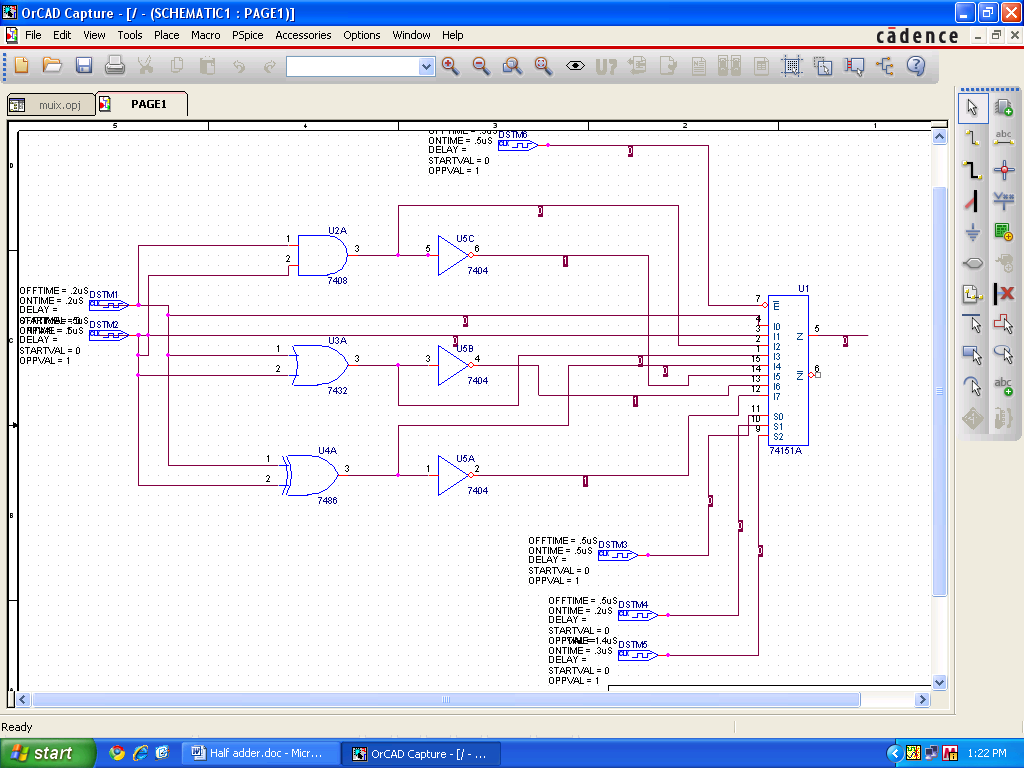
Z = A’.B’.C’ + A’.B’.C + A’.B.C’ + A’.B.C + A.B’.C’ + A.B’.C + A.B.C’ + A.B.C

|  |
| --- |
| 351px-Multiplexer_8-to-1 |

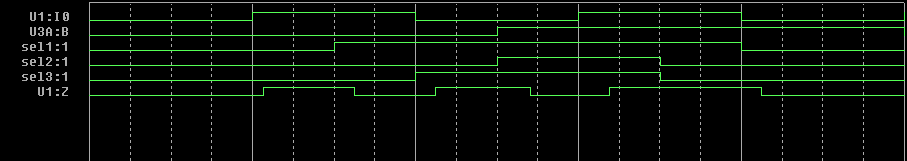
TRUTH TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| **S0** | **S1** | **S2** | **Z** |
| 0 | 0 | 0 | A |
| 0 | 0 | 1 | B |
| 0 | 1 | 0 | C |
| 0 | 1 | 1 | D |
| 1 | 0 | 0 | E |
| 1 | 0 | 1 | F |
| 1 | 1 | 0 | G |
| 1 | 1 | 1 | H |

SCHEMATIC DIAGRAM:



WAVEFORM:



RESULT: The output waveform of 8×1 Multiplexer is verified.

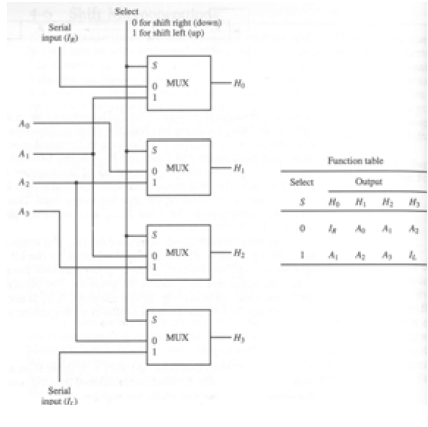
**EXPERIMENT NO.6**

AIM: To design a 4 bit combinational shifter.

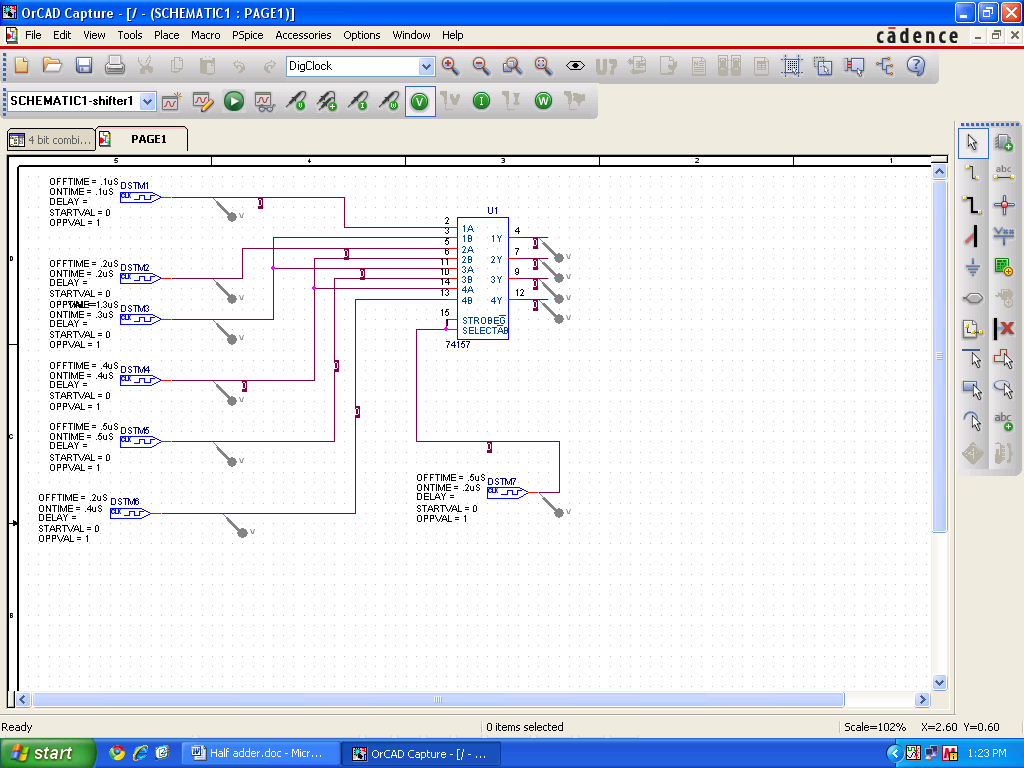
IC USED: 74157(quad 2×1 MUX).

THEORY: An n-bit shift circuit has a single n-bit data input A, and a single n-bit output R and a number of control inputs to determine the shift amount ( 0 to n-1).

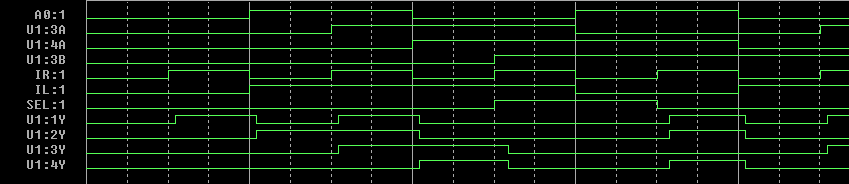
In this case the possible shift operations are SHIFT LEFT and SHIFT RIGHT. This shifting depends on selection line input S. If S = 0 then SHIFT RIGHT operation takes place , else SHIFT LEFT operation takes place.



SCHEMATIC:



WAVEFORM:



RESULT: The circuit of 4-bit combinational shifter is designed and output is verified.

**EXPERIMENT NO. 7**

AIM: To design a 3:8 decoder

IC USED: 7404(NOT), 7408(AND).

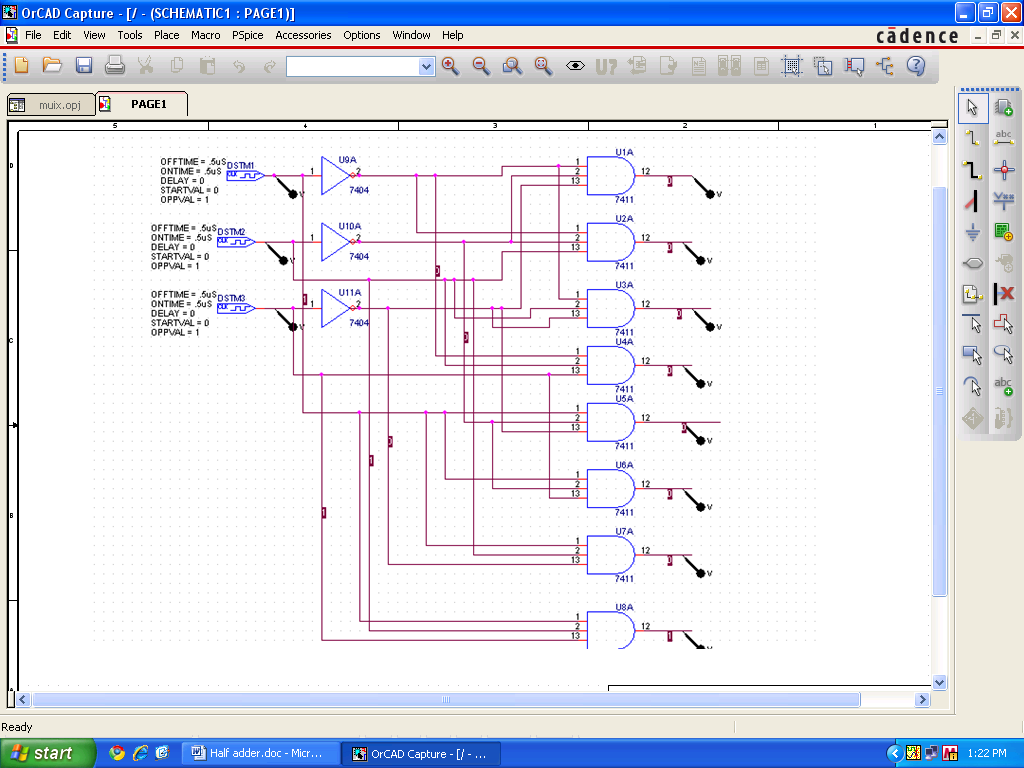
THEORY: A **decoder** is a device which does the reverse of an [encoder](http://en.wikipedia.org/wiki/Encoder), undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode.

In digital electronics, a decoder can take the form of a multiple-input, multiple-output [logic circuit](http://en.wikipedia.org/wiki/Logic_gate) that converts coded inputs into coded outputs, where the input and output codes are different.

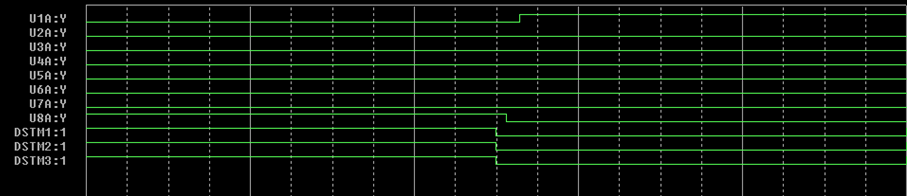
Decoders have 3 inputs and 8 outputs. The binary input fed at input will be decoded to provide either logical high or low on one of the 8 outputs, which is termed as octal equivalent for that binary input. For a binary input 100, which is 4 in octal number system, the output pin called O4 will go either high or low.

.

SCHEMATIC:



WAVEFORM:



RESULT: 3:8 decoder is designed and its output is verified.

**EXPERIMENT NO.8**

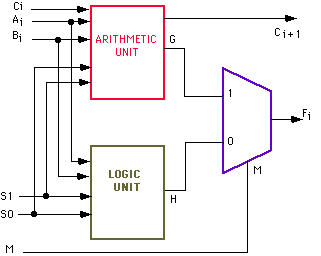
AIM: To design an ALU.

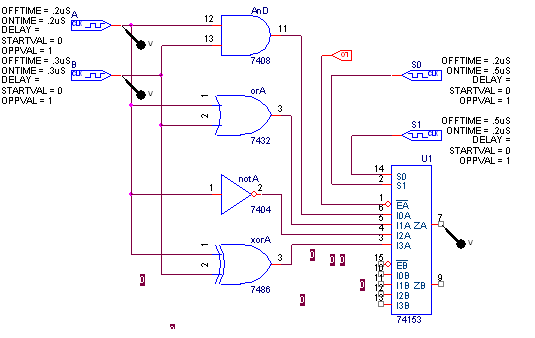
IC USED: 74153(4×1 MUX), 7432(OR), 7404(NOT), 7408(AND), 7486(XOR).

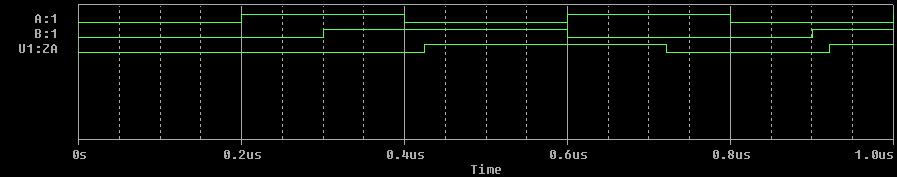
THEORY: In computing, an **arithmetic logic unit** (**ALU**) is a digital circuit that performs arithmetic and logical operations. The ALU is a fundamental building block of the central processing unit (CPU) of a computer, and even the simplest microprocessors contain one for purposes such as maintaining timers. The processors found inside modern CPUs and graphics processing units (GPUs) accommodate very powerful and very complex ALUs; a single component may contain a number of ALUs.

Mathematician John von Neumann proposed the ALU concept in 1945.

Below is a block diagram for an ALU:



SCHEMATIC: 

WAVEFORM: 

RESULT: The ALU is designed and its output is verified.

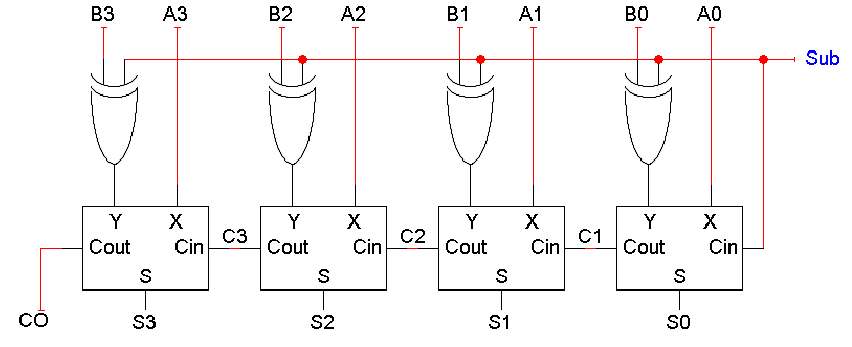
**EXPERIMENT NO.9**

AIM: To design a 4-bit adder subtractor.

IC USED: 7483A (4-bit adder), 7486(X-OR).

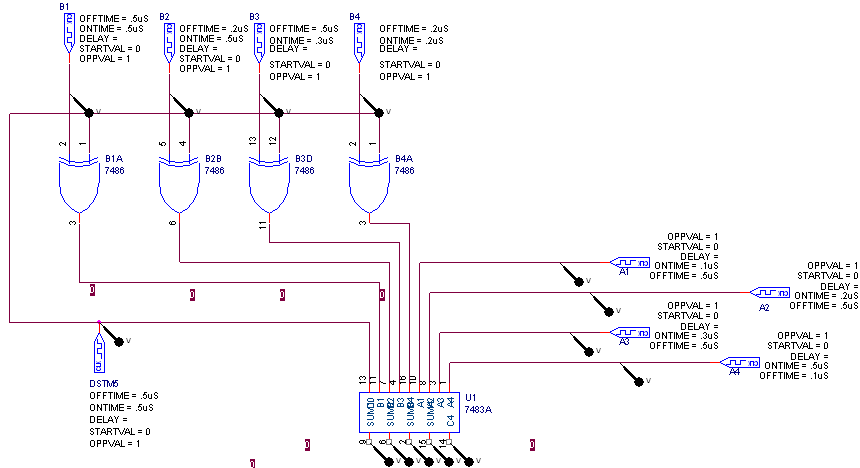
THEORY:

It is a combinational circuit which can act as both a binary adder and a binary subtractor. It is constructed using 4 full adders.Block diagram of the circuit is shown below:

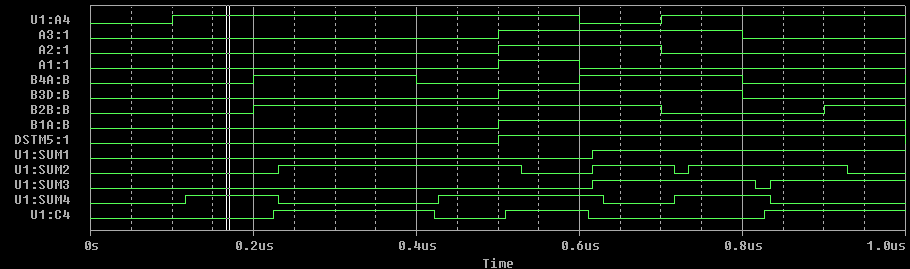


In this case if sub is 0, then the circuit acts as an adder,else subtractor. Subtraction is done by adding two’s compliment of B to A, which effectively results in A - B.

SCHEMATIC:



WAVEFORM:



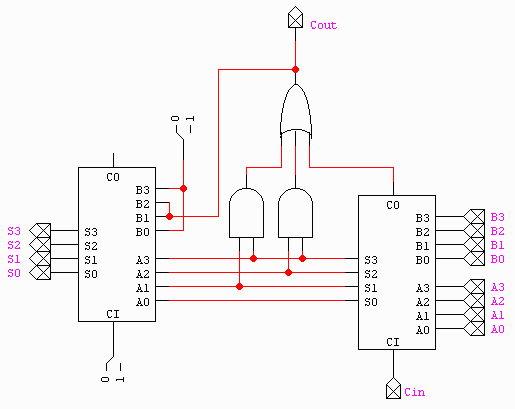
RESULT: The circuit of a 4-Bit adder subtractor is designed and verified.

**EXPERIMENT NO.10**

AIM: To design a BCD adder.

IC USED: 7483A (4-bit adder), 7408(AND), 7432(OR).

THEORY:

It is possible to perform addition in BCD by first adding in binary, and then converting to BCD afterwards. Conversion of the simple sum of two digits can be done by adding 6 (that is, 16 – 10) when the result has a value greater than 9. 

For example:

1001 + 1000 = 10001 = 0001 0001

9 + 8 = 17 = 1 1

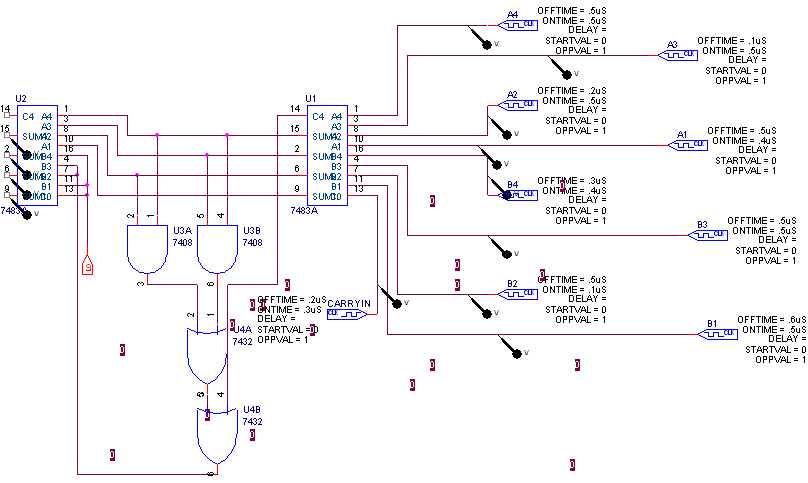
In BCD, there cannot exist a value greater than 9 (1001) per nibble. To correct this, 6 (0110) is added to that sum to get the correct first two digits:

0001 0001 + 0000 0110 = 0001 0111

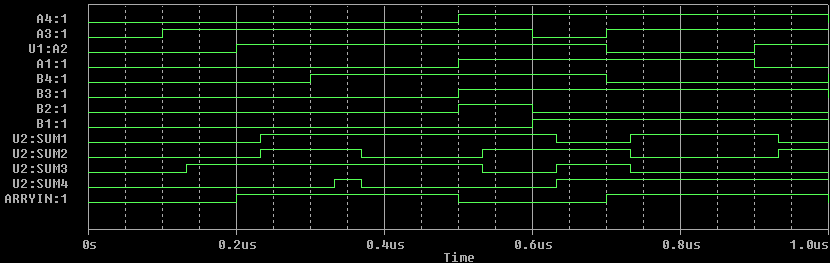
1 1 + 0 6 = 1 7

This gives two nibbles, 0001 and 0111, which correspond to the digits "1" and "7". This yields "17" in BCD, which is the correct result. This technique can be extended to adding multiple digits, by adding in groups from right to left, propagating the second digit as a carry, always comparing the 5-bit result of each digit-pair sum to 9.

SCHEMATIC:



WAVEFORM:



RESULT: BCD adder is designed and output is verified.