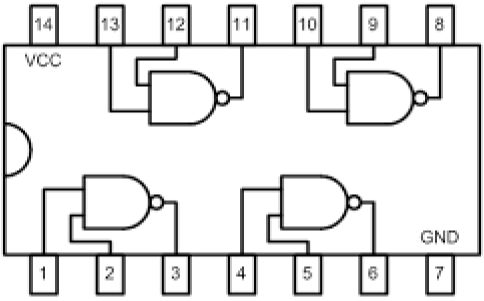
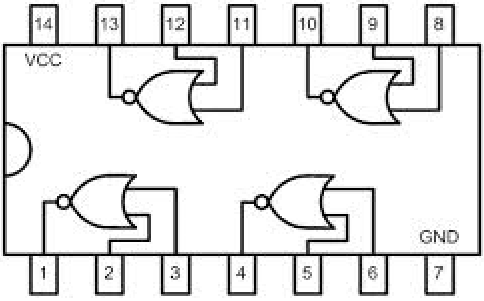
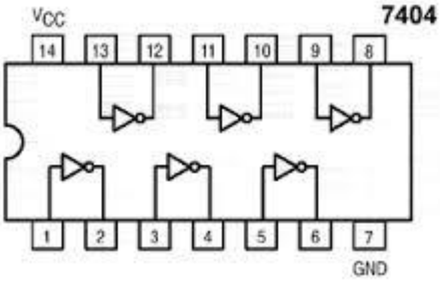


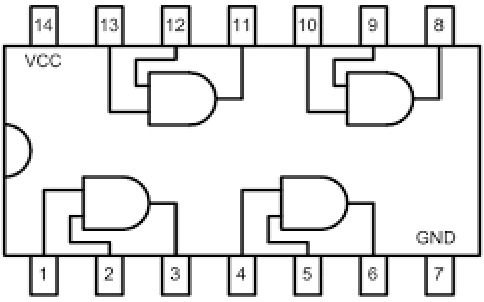
7400 Quad 2 Input NAND



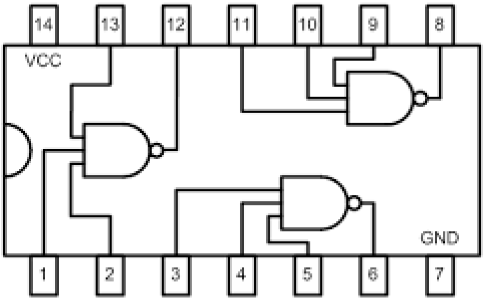
7402 Quad 2 Input NOR



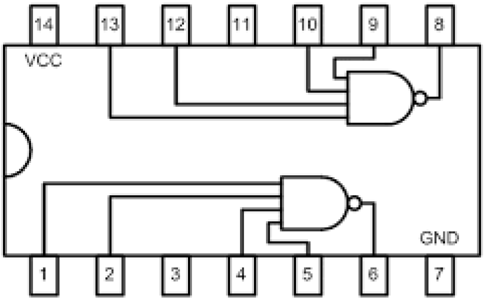
7404



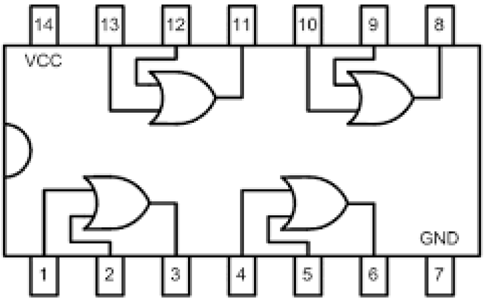
7408 Quad 2 Input AND



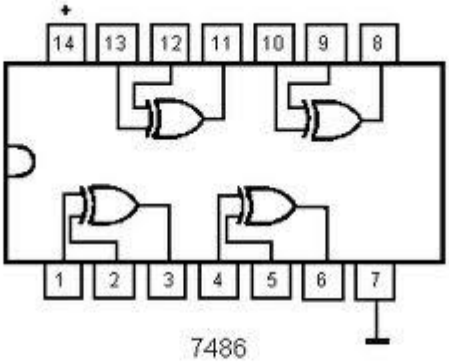
7410 Triple 3 Input NAND



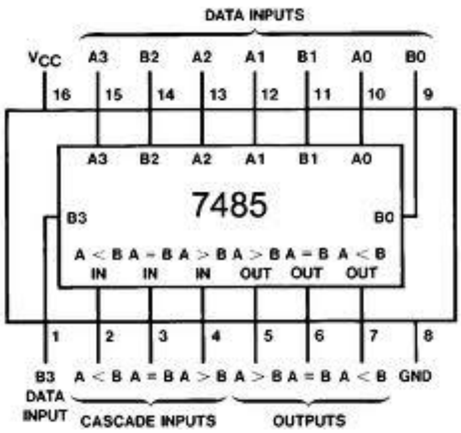
7420 Dual 4 Input NAND

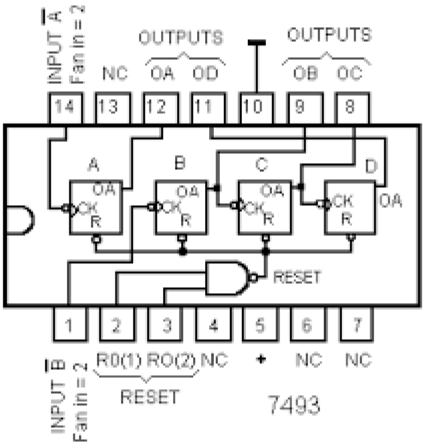


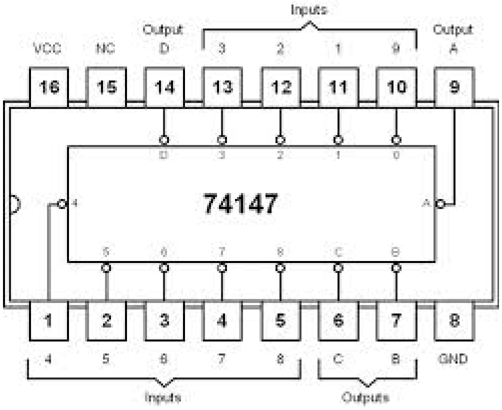
7432 Quad 2 Input OR

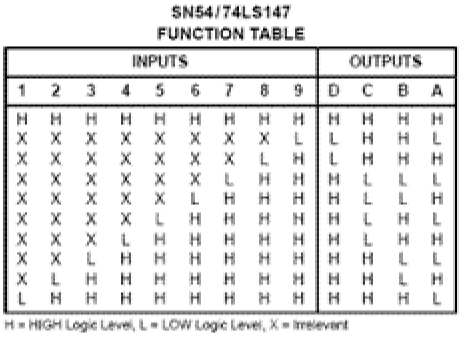


7486



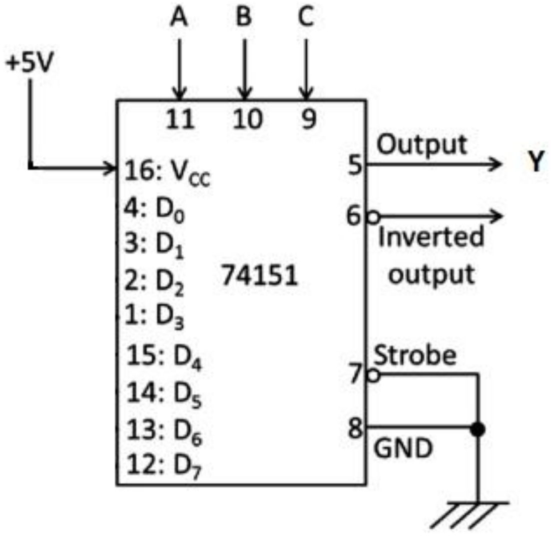


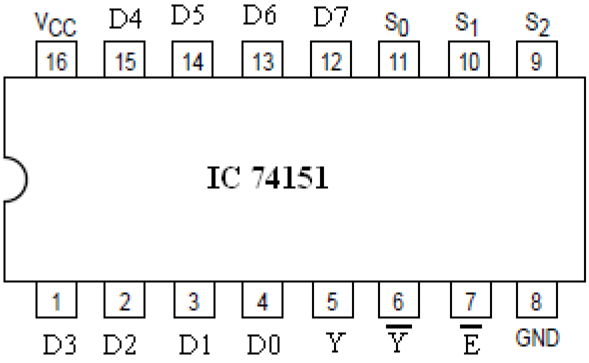


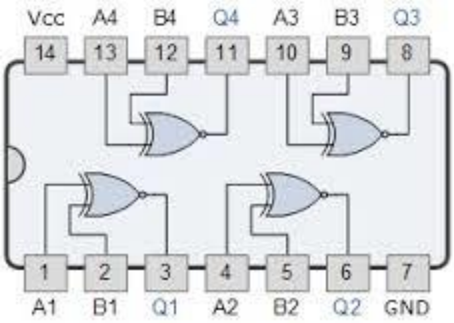

SN54/74LS147
FUNCTION TABLE

INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

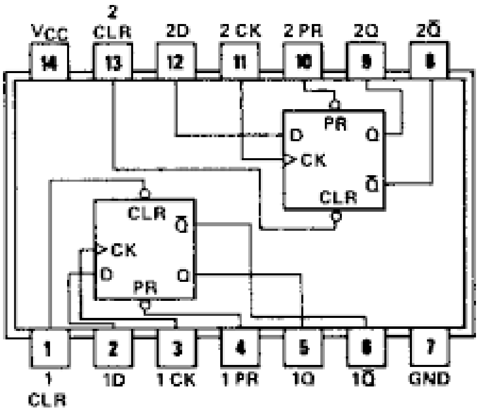
H = HIGH Logic Level, L = LOW Logic Level, X = Irrelevant



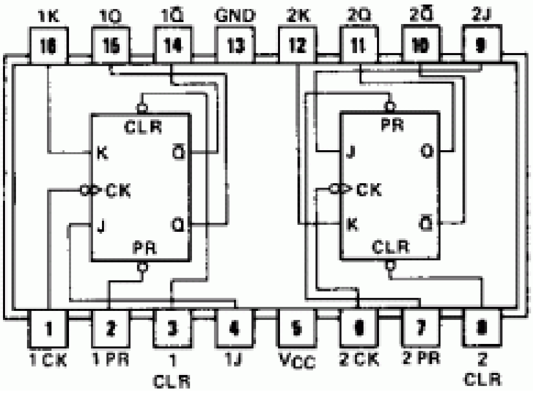


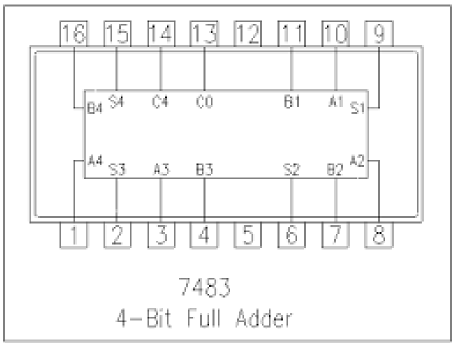


7474

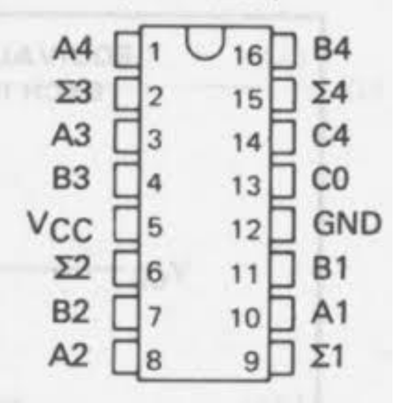


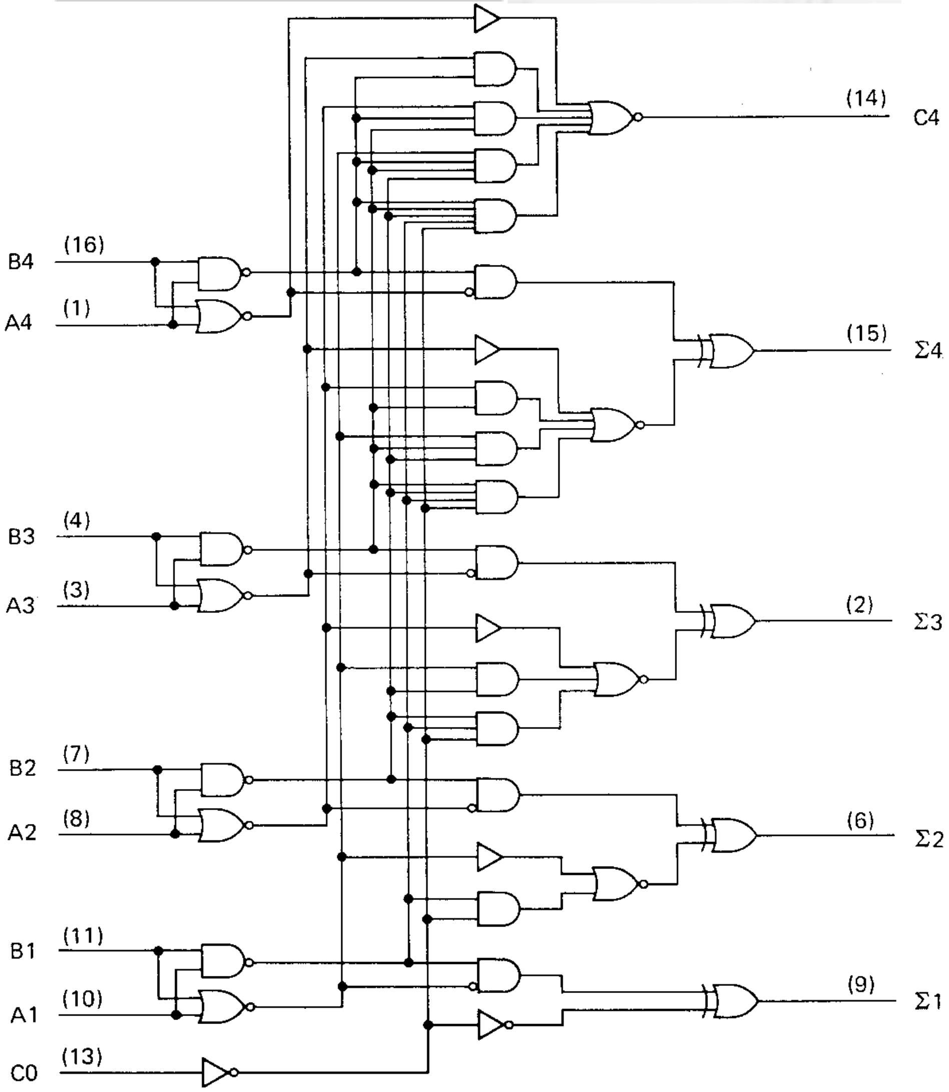
7476





(TOP VIEW)





Lab 01 Basic Gates

Gates are used to implement combinational circuits such as mathematical circuits, control circuits, etc. In this experiment, basic gates, i.e. OR, AND, NOT, NOR, NAND, XOR, XNOR and universality of NAND and NOR gates will be studied.

Objectives

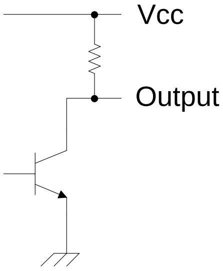
- To learn how to use basic equipments of digital experiments.
- To learn how to use IC data sheet.
- To perform and to understand the operation of basic gates.
- To study the universality of NAND and NOR gates.

Component lists

- 7400 7402 7404 7408 7432 7486 74266
  4001 4011 4071 4081
- Ω 220

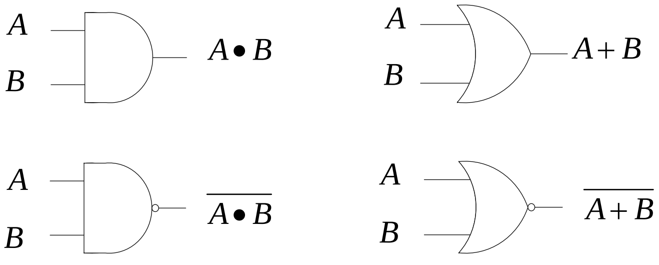
Note

- Study the pin layout diagram of all ICs used in the experiment from the data sheet and save them for future usage.
- IC 74266 has open-collector outputs, a resistor is needed or the output can't be display. Connect the circuit as shown below.



Experiment 1.1: AND, OR, NAND and NOR gates experiments

- Connect each gate as shown below and record the result in the table.

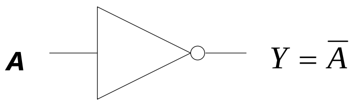


Gate	IC	Output Y, when AB =			
		00	01	10	11
AND	7408				
AND	4081				
OR	7432				
OR	4071				
NAND	7400				
NAND	4011				
NOR	7402				
NOR	4001				

Table 1.1 AND, OR, NAND and NOR gates experiment

Experiment 1.2: NOT gate experiment

- Connect each gate as shown below and record the result in the table.

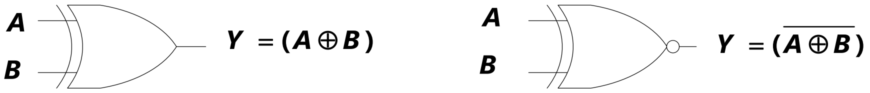


Gate	IC	Output Y, when A =	
		0	1
NOT	7404		

Table 1.2 NOT gate experiment

Experiment 1.3: XOR and XNOR gates experiment

- Connect each gate as shown below and record the result in the table.

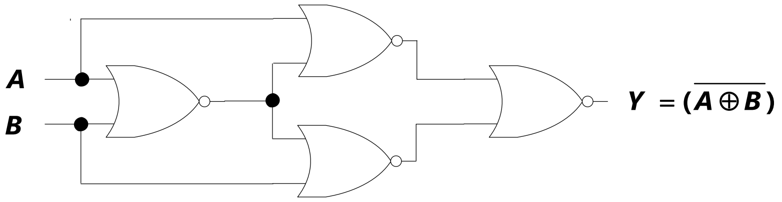


Gate	IC	Output Y, when AB =			
		00	01	10	11
XOR	7486				
XNOR	74266				

Table 1.3 XOR and XNOR gates experiment

Experiment 1.4: Implementing XNOR gate from NOR gates

- Construct an XNOR gate from NOR gates using the circuit below and record the result in the table.



Gate	Using	Output Y, when AB =			
		00	01	10	11
XNOR	NOR gates				

Table 1.4 Implementing XNOR gate from NOR gates

Questions

1.1 Summarize the operations of OR, AND, NOT, NOR, NAND, XOR and XNOR gates.
1.2 Design an OR gate from NOR gates.
1.3 Implement XNOR gate of experiment 1.4 using only NAND gates.

Lab 02 Boolean Algebra, De Morgan's Theorem, and K-map

Boolean algebra is a mathematical tool used in the analysis and design of digital circuits. It can be used to simplify the expression of logic circuit and can lead to a simpler way of implementing the circuit. Both NAND and NOR gates can be used to implement any of the basic Boolean operations (i.e. universality from lab 01). Two of the most important theorems of Boolean algebra are De Morgan's theorems. The other technique, called Karnaugh-map, can also be used to simplify the logic expression.

Boolean Algebra

Commutative law

a) $A+B = B+A$

b) $A\cdot B = B\cdot A$

Distributive law

a) $A \cdot (B+C) = (A\cdot B) + (A\cdot C)$

b) $A + (B\cdot C) = (A+B) \cdot (A+C)$

Complementary law

a) $(A)' = A'$

b) $(A')' = A$ (Involution law)

Principle of duality

a) $0\cdot A = 0$

b) $1\cdot A = A$

Associative law

a) $A+(B+C) = (A+B)+C$

b) $A\cdot(B\cdot C) = (A\cdot B)\cdot C$

Idempotent law

a) $A+A = A$

b) $A\cdot A = A$

Redundant law

a) $0+A = A$

b) $1+A = 1$

Absorption law

a) $A+A'.B = A+B$

b) $A\cdot(A'+B) = A\cdot B$

c) $A+AB = A\cdot(1+B) = A$

DeMorgan's Theorems

a) $(A+B)' = A'\cdot B'$

b) $(A\cdot B)' = A'+ B'$

Objectives

- To study the usage of Boolean algebra.
- To study DeMorgan's Theorems.
- To study K-map as a tool to simplify and design logic circuits.

Component lists

- 7400 7402 7404 7408 7410 7432

Note

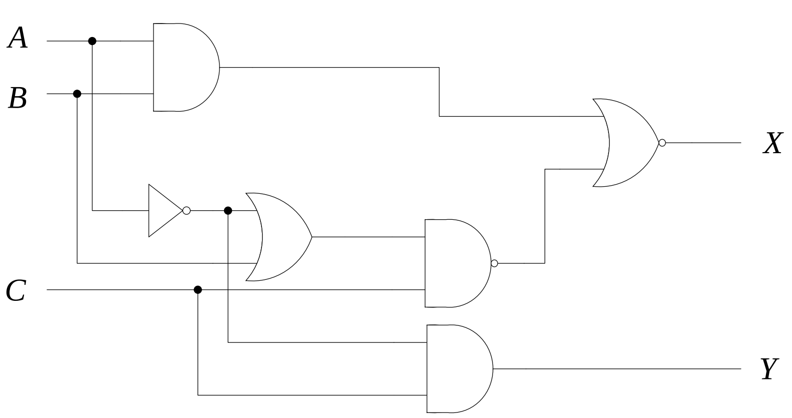
- Study k-map simplification process from your digital systems textbook.

Experiment 2.1: Simplification using Boolean Algebra

- Let $X = A \cdot B + ((A' + B) \cdot C)')'$ and $Y = A' \cdot C$, the following proves that we can use Boolean algebra to simplify the Boolean expression. Hence, typically, a simpler (i.e., faster, less expensive, and less complex) circuit can be obtained using Boolean algebra.
- Prove

$$
\begin{aligned}
(A \cdot B + ((A'+B) \cdot C)')' &= (A \cdot B)' \cdot (A'+B) \cdot C \\
&= (A'+B') \cdot (A'+B) \cdot C \\
&= (A' + A'B' + A'B + BB') \cdot C \\
&= (A' + BB') \cdot C \\
&= A'C
\end{aligned}
$$

- Construct a circuit as shown below and record the result in the table.



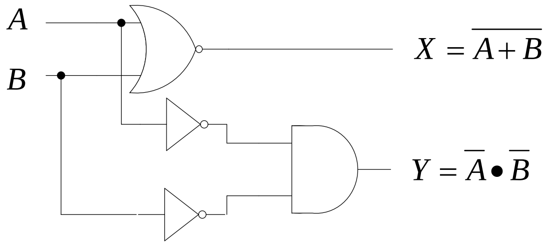
Input			Output	
A	B	C	X	Y
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table 2.1 Simplification using Boolean Algebra

Experiment 2.2: DeMorgan's Theorems

- Construct circuits as shown below and record the result in the tables.

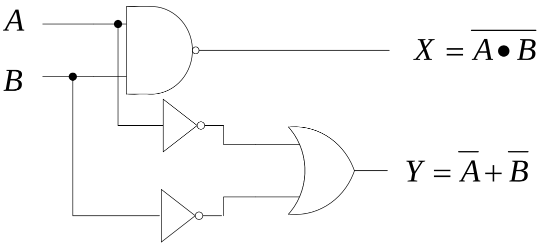
DeMorgan's Theorems (1)



Input		Output	
A	B	X	Y
0	0		
0	1		
1	0		
1	1		

Table 2.2A DeMorgan's Theorems: (A+B)' = A'·B'

DeMorgan's Theorems (2)


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

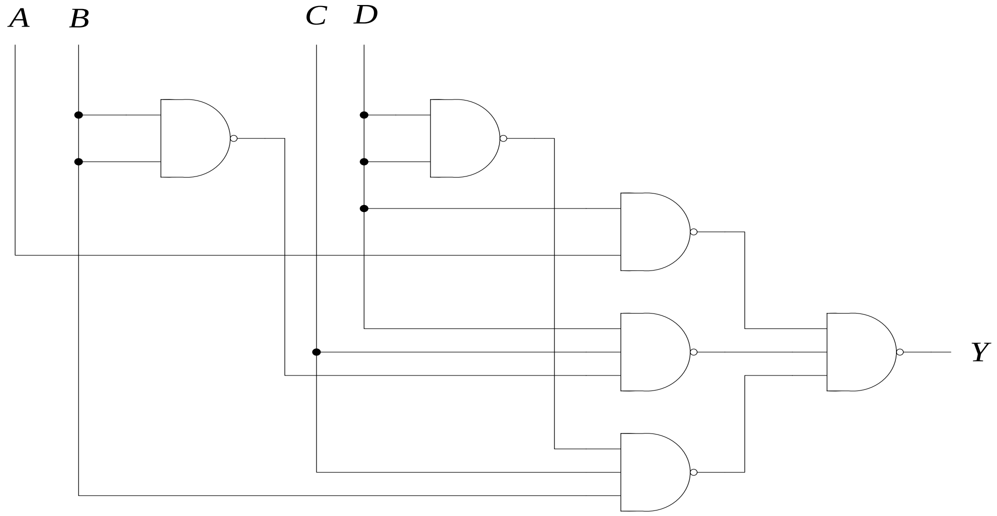
Input		Output	
A	B	X	Y
0	0		
0	1		
1	0		
1	1		

Table 2.2B DeMorgan's Theorems (cont): (A·B)' = A'+B'

Experiment 2.3: K-map simplification

The following circuit has as its input an integer 0..9 in BCD*, and outputs 1 whenever an input is greater than 0 and divisible by 3, i.e. Y = Σ m(3,6,9)+d(10,11,12,13,14,15). The NOR gates implementation of the circuit is as follow:

* see textbook for BCD code



Input (BCD-8421)				Output
A	B	C	D	Y
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	

Table 2.3 K-map simplification

Questions

2.1 Explain Product of Sum (POS) and Sum of Product (SOP) respectively.

2.2 Give examples how to change from POS to SOP using DeMorgan's Theorems and vice versa.

2.3 Show the K-map table of experiment 2.3 and show how to simplify using K-map.

Lab 03 Encoder & Decoder

An encoder is a device that generates a unique binary code in response to the activation of individual input. Encoder is very useful for code conversions, e.g. converts numbers from base-10 to base-8.

A traditional 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. Decoders are widely used in the memory system of a computer where they respond to the address code generated by the CPU to activate a particular memory location. Many types of decoders are available.

Objectives

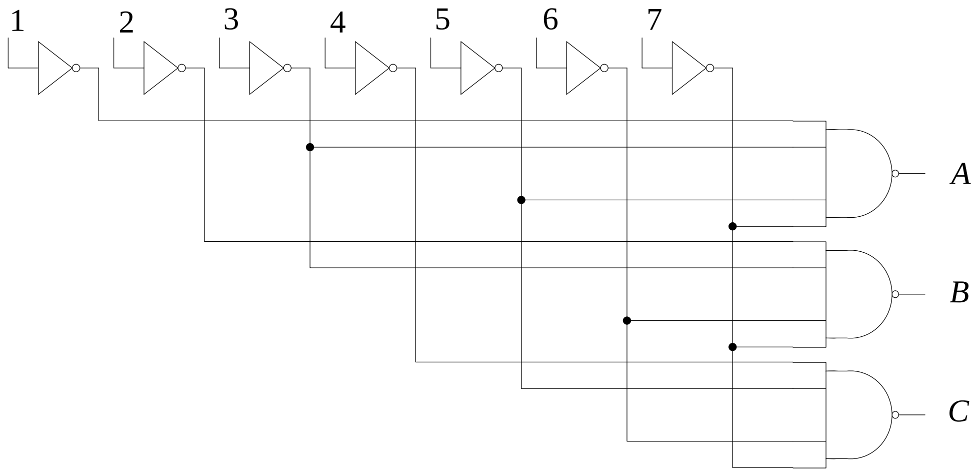
- To study encoder and decoder circuits.

Component lists

- 7404 7420 74147

Experiment 3.1: Encoder circuits

3.1A Decimal to Octal encoder: After a base 10 digit [0..7] is entered into the circuit, its equivalent octal number is produced.



Input							Output		
1	**2**	**3**	**4**	**5**	**6**	**7**	**C (msb)**	**B**	**A (lsb)**
0	0	0	0	0	0	0 *			
1	0	0	0	0	0	0			
0	1	0	0	0	0	0			
0	0	1	0	0	0	0			
0	0	0	1	0	0	0			
0	0	0	0	1	0	0			
0	0	0	0	0	1	0			
0	0	0	0	0	0	1			

Table 3.1A Decimal to Octal encoder

* equivalent to a '0' is entered

3.1B Design a 2:4 Priority encoder that produces the following function table:

Input (Active High)			Output (Active Low)	
1	**2**	**3**	**B**	**A (lsb)**
x	x	1	1	1
x	1	0	1	0
1	0	0	0	1
0	0	0	0	0

Table 3.1B Priority encoder

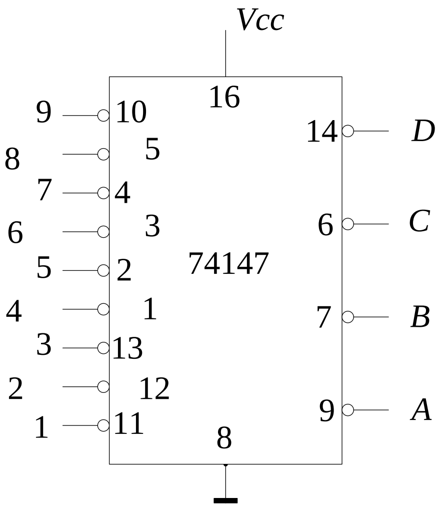
* The last row is equivalent to a '0' is entered

Show expressions of B and A in the report.

3.1C Decimal to BCD encoder: After a base 10 digit is entered into the circuit, its equivalent BCD number is produced.

Notes

- Study BCD from digital systems textbook and IC 74147 from the data sheet.
- DCBA should connect to the cathode pins of LEDs.
- IC 74147 has both active 'LOW' inputs and outputs.



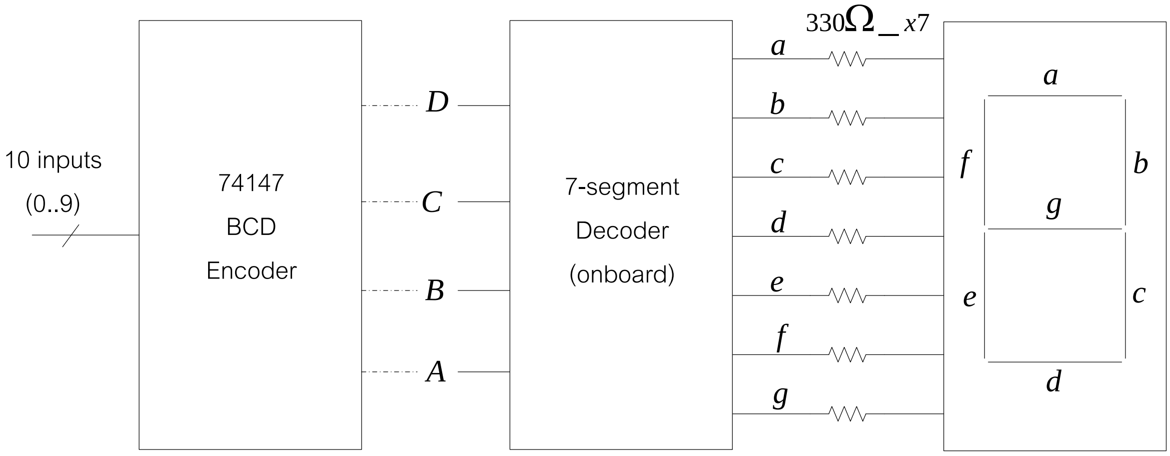
Input									Output			
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**D(msb)**	**C**	**B**	**A(lsb)**
1	1	1	1	1	1	1	1	1				
1	1	1	1	1	1	1	1	0				
1	1	1	1	1	1	1	0	1				
1	1	1	1	1	1	0	1	1				
1	1	1	1	1	0	1	1	1				
1	1	1	1	0	1	1	1	1				
1	1	1	0	1	1	1	1	1				
1	1	0	1	1	1	1	1	1				
1	0	1	1	1	1	1	1	1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

Table 3.1C Decimal to BCD encoder

Experiment 3.2: Decoder circuits

- First, try using a 7-segment decoder available onboard, i.e. we can connect input switches directly to DCBA inputs.
- Using circuit of experiment 3.1B, connect output DCBA to 7-segment decoder inputs (which connect directly to the 7-segment decoder IC) on the experiment board.
- Note that, because IC 74147 has active low inputs and outputs, its DCBA should connect to cathode pins of LED. However, the 7-segment displays onboard are common cathode. Therefore, we have to modify the circuit.

Note: Study how to use a 7-segment display on your experiment board.



Input of 7-segment decoder				Output						
D(msb)	C	B	A(lsb)	a	b	c	d	e	f	g
0	0	0	0							
0	0	0	1							
0	0	1	0							
0	0	1	1							
0	1	0	0							
0	1	0	1							
0	1	1	0							
0	1	1	1							
1	0	0	0							
1	0	0	1							

Table 3.2 7-segment Decoder

Questions

3.1 Typically there are two types of 7-segment displays, i.e. common cathode and common anode. Find information about these two 7-segment types.

3.2 CPU generates a 4-bit address to access word in a 16-word memory system. Show the decoding circuit used for address decoding.

Lab 04 Multiplexer, Demultiplexer & Arithmetic Circuits I

A multiplexer (mux) can be used to select one of several input signals and passes it on to the output. Its applications include data selection, operation sequencing, parallel to serial conversion, waveform generation, and login function generation. Nevertheless, in computer system a mux is typically used as a data routing device. By taking turns, many different data signals can share the same data path.

Demultiplexer (demux) is then used at the other end of the data path to separate the signals that are sharing a data path and distribute them to their respective destinations.

Arithmetic circuits are widely used in computer system. They allow us to perform arithmetic calculations. A half adder (HA) circuit can be used to add 2 one-bit numbers and cannot accept any carry-in. However, it can produce a carry out. A full adder (FA) circuit can add 2 one-bit numbers and a carry-in. Therefore, we can cascade them to produce an adder that can add 2 m-bit 2's complement numbers. We can typically find an IC parallel adder available to be used as a building block for more advanced arithmetic circuits. Moreover, an adder circuit constructed this way can be easily modified to handle a subtraction of 2 m-bit 2's complement numbers.

Objectives

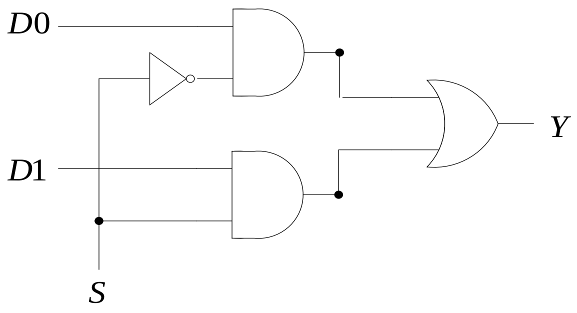
- To study multiplexer, demultiplexer and arithmetic circuits.

Component lists

- 7400 7404 7408 7410 7420 7421 7432 7486 74151

Experiment 4.1: Multiplexer circuits

4.1A 2:1 Mux

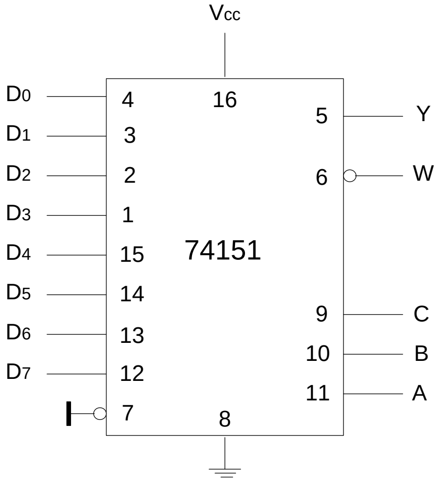


Data		Selector S	Output Y
D0	D1		
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Table 4.1A 2:1 Multiplexer

4.1B 8:1 Mux

Note: Study how 74151 IC works from the data sheet.

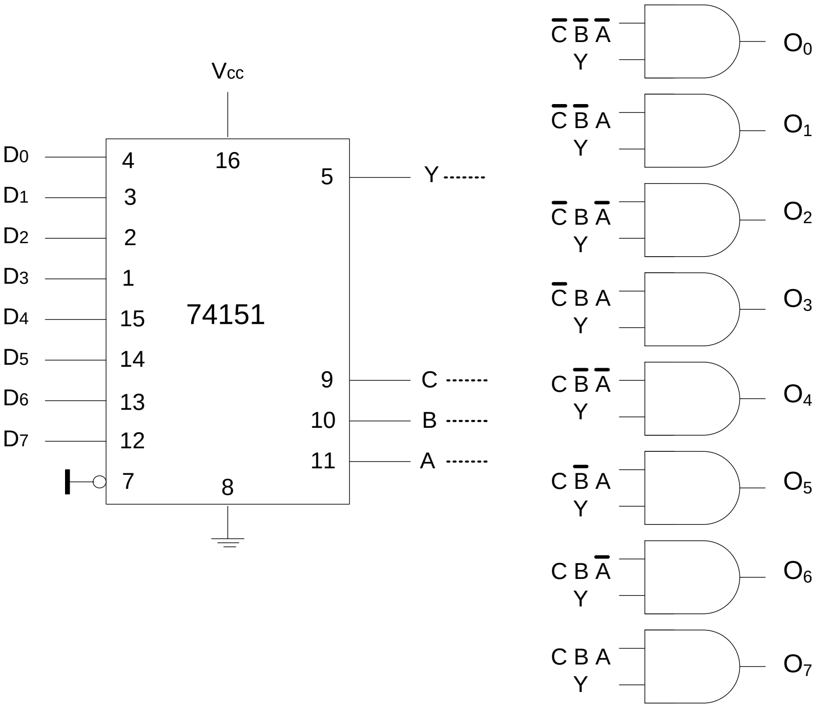


Data ($D_7..D_0$)	Address/Data Selector			Output Y
	C (msb)	B	A (lsb)	
11011010	0	0	0	
11011010	0	0	1	
11011010	0	1	0	
11011010	0	1	1	
11011010	1	0	0	
11011010	1	0	1	
11011010	1	1	0	
11011010	1	1	1	

Table 4.1B 8:1 Multiplexer

Experiment 4.2: Demultiplexer circuits

4.2 8:1 Mux working with 1:8 Demux. Connect 74151 with 8 of 4-input AND gates (or using 3x8 decoder and 2-input AND gates/ or other relevant gates) and record outputs $O_7..O_0$ in table 4.2B.

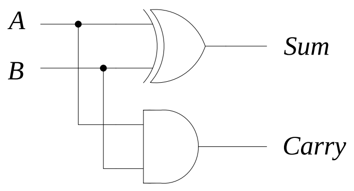


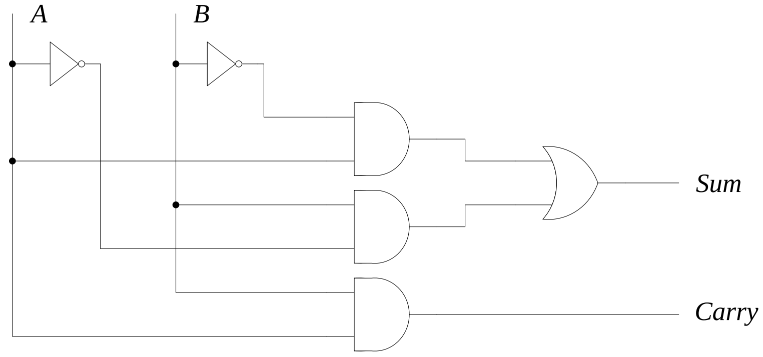
Data ($D_7..D_0$)	Address/Data Selector			Output ($O_7..O_0$)
	C (msb)	B	A (lsb)	
11011010	0	0	0	
11011010	0	0	1	
11011010	0	1	0	
11011010	0	1	1	
11011010	1	0	0	
11011010	1	0	1	
11011010	1	1	0	
11011010	1	1	1	

Table 4.2 8:1 Multiplexer and Demultiplexer

Experiment 4.3: Adder circuits

4.3A: Half adder (HA) circuits:

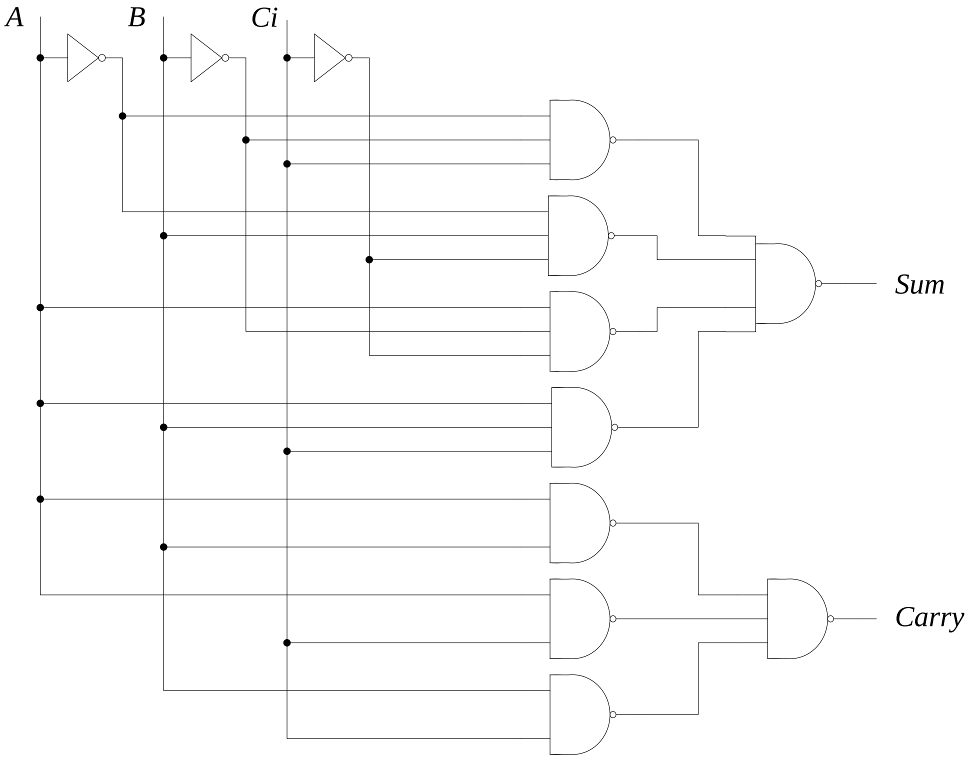




Input AB	Sum		Carry	
	Left Circuit	Right Circuit	Left Circuit	Right Circuit
00				
01				
10				
11				

Table 4.3A Half adder (HA) circuits

4.3B: Full adder (FA) circuit:

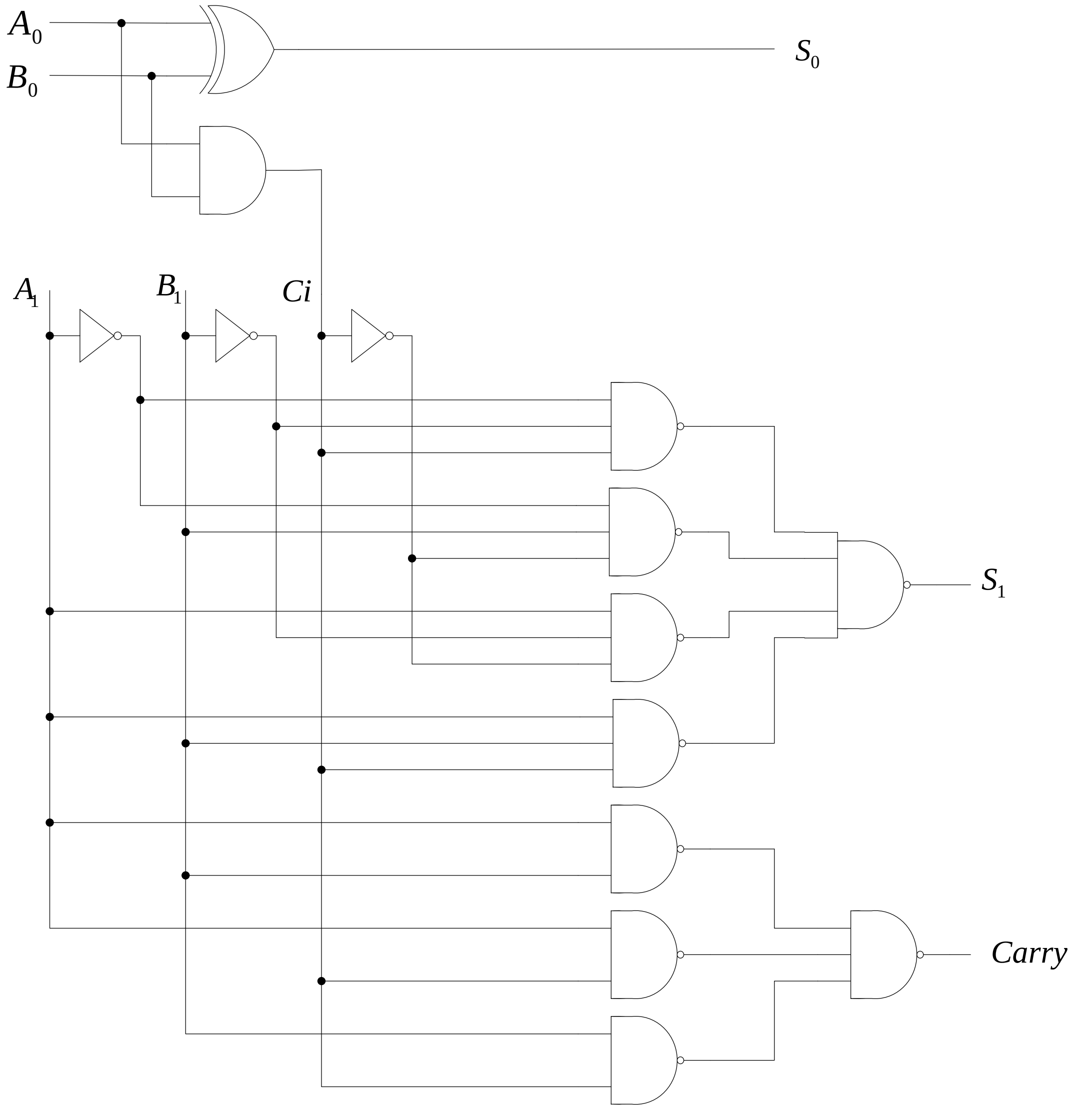


Input			Output	
A	B	Ci	Sum	Carry
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

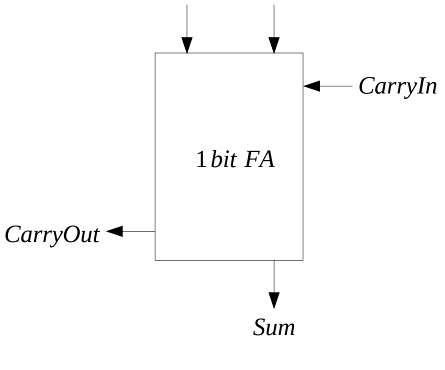
Table 4.3B Full adder (FA) circuit (NAND gates and Inverters Imp^n)

Questions

4.1 Explain the given 2-bit adder circuit below:



4.2 Using the building block below, show how to construct a 4-bit full adder that can add two 4-bit 2's complement numbers.



Lab 05 Arithmetic Circuits II

Arithmetic circuits are widely used in computer system. They allow us to perform arithmetic calculations. A half adder (HA – lab 04) circuit can be used to add 2 one-bit numbers and cannot accept any carry-in. However, it can produce a carry out. A full adder (FA – lab 04) circuit can add 2 one-bit numbers and a carry-in. Therefore, we can cascade them to produce an adder that can add 2 m-bit 2's complement numbers. We can typically find an IC parallel adder available to be used as a building block for more advanced arithmetic circuits. Moreover, an adder circuit constructed this way can be easily modified to handle a subtraction of 2 m-bit 2's complement numbers.

However, in the experiment, we will construct a full subtractor (FS) directly. A half subtractor (HS) circuit can be used to subtract 2 one-bit numbers (i.e., a minuend and a subtrahend) and cannot accept any borrow-in. The HS produces a borrow-out, though. A full subtractor (FS) circuit can handle 2 one-bit numbers and a borrow-in. it produces two outputs, i.e. a difference and a borrow-out. Like FAs, we can cascade m FSs to obtain an m-bit subtractor circuit. If numbers are in 2's complement system then an adder circuit can be modified to also perform subtraction.

More advanced (and more complex) arithmetic circuits will be studied in computer architectures course (EGCI 333).

Objectives

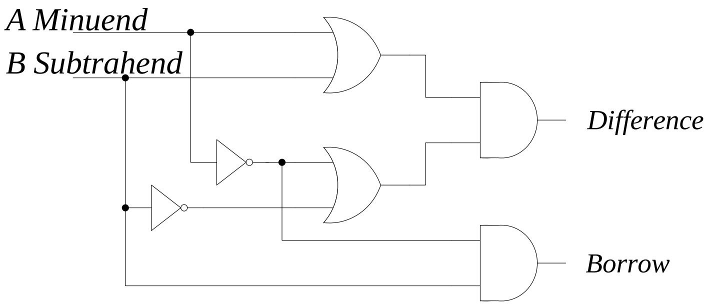
- To study the operations of basic arithmetic circuits.

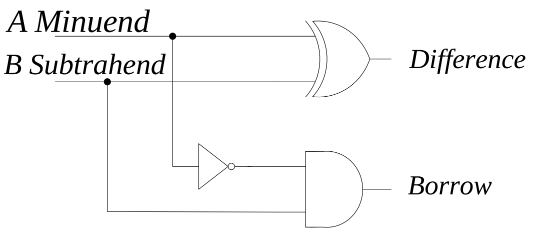
Component lists

- 7400 7404 7408 7410 7420 7432 7483 7486 ++

Experiment 5.1: Subtractor circuits

5.1A Half subtractor circuits:

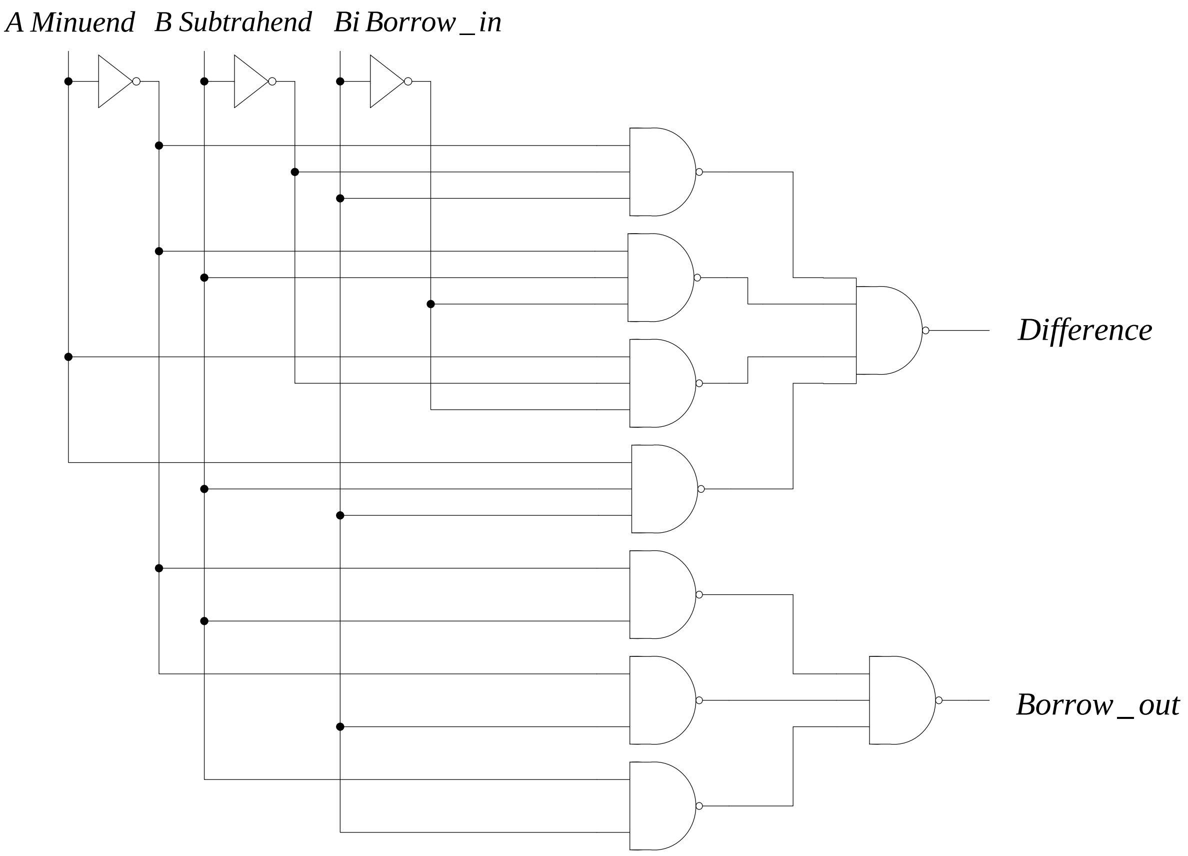




Input AB	Difference		Borrow	
	Left Circuit	Right Circuit	Left Circuit	Right Circuit
00				
01				
10				
11				

Table 5.1A Half subtractor (HS) circuits

5.1B Full subtractor circuit:

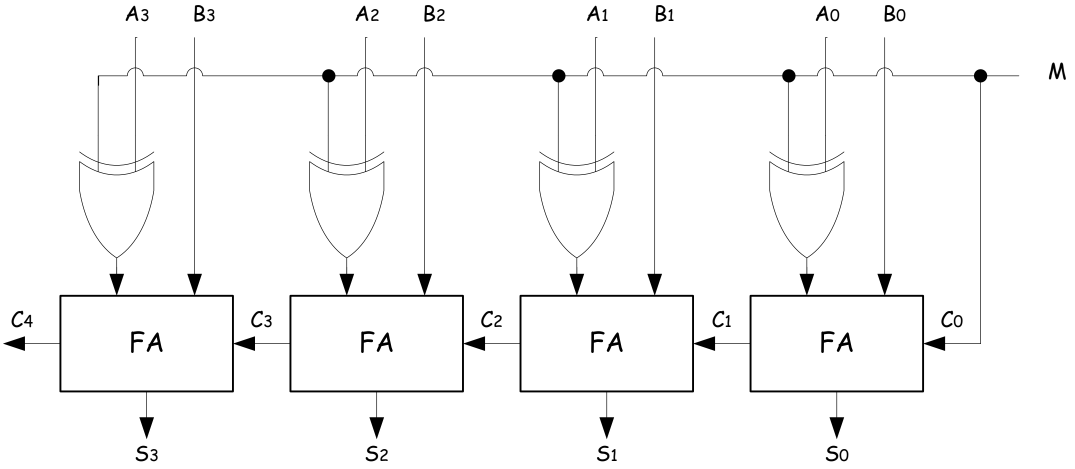


Input			Output	
A	**B**	**Bi**	**Difference**	**Borrow_out**
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Table 5.1B Full subtractor (FS) circuit (NAND gates and Inverters Imp^n)

Experiment 5.2: Adder-Subtractor Circuit

5.2 Adder-Subtractor Circuit: The circuit can be used to add or subtract two 2's complement numbers. The following circuit performs $B_{3..0}$ (Minuend) ± $A_{3..0}$ (Subtrahend). C_3 can't be accessed from 7483 so a separate circuit is needed.



Input					Output				
M	$B_{3..0}$	$B_{3..0}$ in base-10	$A_{3..0}$	$A_{3..0}$ in base-10	C_4	C_3	$C_4 \oplus C_3$	$S_{3..0}$	$S_{3..0}$ in base-10
0	0000		0101						
0	0110		0010						
0	1000		0010						
0	1101		0110						

Table 5.2A Circuit acts as an Adder

Input					Output				
M	$B_{3..0}$	$B_{3..0}$ in base-10	$A_{3..0}$	$A_{3..0}$ in base-10	C_4	C_3	$C_4 \oplus C_3$	$S_{3..0}$	$S_{3..0}$ in base-10
1	0000		0101						
1	0110		0010						
1	1000		0010						
1	1101		0110						

Table 5.2B Circuit acts as a Subtractor

Questions

5.1 What is the expression of C_3?

5.2 Explain the working of Adder-Subtractor Circuit in experiment 5.2.

5.3 Explain the role of $C_4 \oplus C_3$ in tables 5.2 A and B.

Lab 06 Flip-Flops and Registers

A flip-flop (FF) is a logic circuit with a memory characteristic such that its Q and Q' output will go to a new state in response to an input pulse and will remain in that new state after the input pulse is terminated. FFs can be used as memory units for many applications. There are many types of FF, i.e. D, RS or SC, JK and T.

Register is a group of FFs that work together as a single unit. Registers are used mainly within CPU for storing most active data.

Objectives

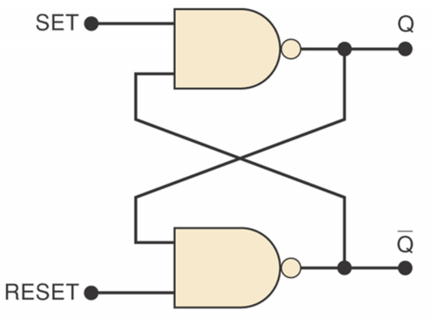
- To study the working of D, JK and T FFs.
- To study how to construct a register from FFs.

Component lists

- 7400 7402 7474 7476

Experiment 6.1: Latches

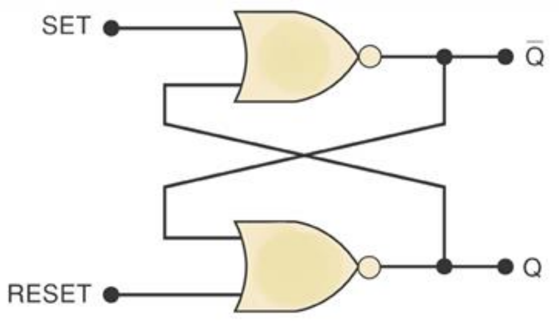
6.1A NAND latch



Set	Reset	Output	
		Q	Q'
0	0		
0	1		
1	0		
1	1		

Table 6.1A Working of NAND latch

6.1B NOR latch

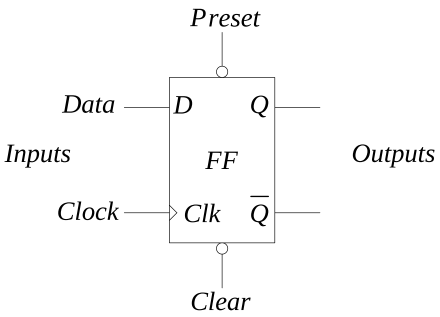


Set	**Reset**	**Output**	
		Q	**Q'**
0	0		
0	1		
1	0		
1	1		

Table 6.1B Working of NOR latch

Experiment 6.2: D FFs

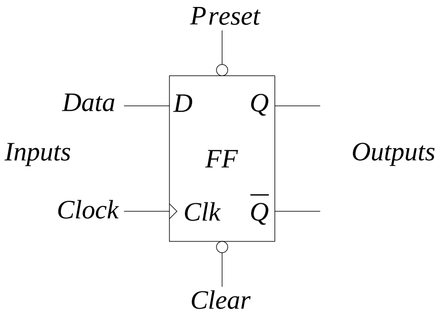
6.2A Workings of D FF



Data	**Clock Pulse**	**Output (after clock pulse)**	
		Q	**Q'**
-	Before 1st clock		
1	↑		
0	↑		
1	↑		
0	↑		

Table 6.2A Working of D FF

6.2B Asynchronous inputs of D FF

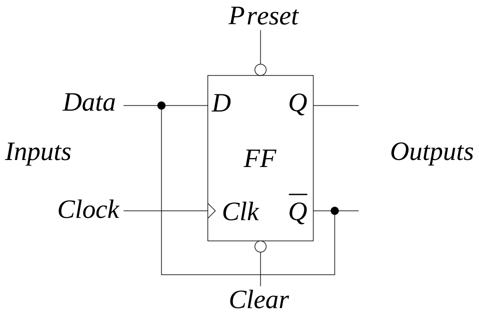


Inputs				Output	
Asynchronous		Synchronous			
Preset	Clear	Clock	Data	Q	Q'
0	1	x	x		
1	0	x	x		
*0	0	x	x		
1	1	↑	1		
1	1	↑	0		
1	1	0	1		
1	1	0	0		

Table 6.2B Asynchronous Inputs of D FF

* Not used
x = don't care

6.2C Construct T FF from D FF

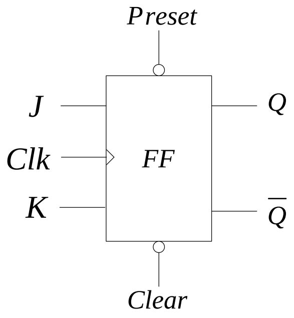


Clock Pulse	Output	
	Q	Q'
Before 1st clock		
↑		
↑		
↑		
↑		

Table 6.2C Construct T FF from D FF

Experiment 6.3: JK FFs

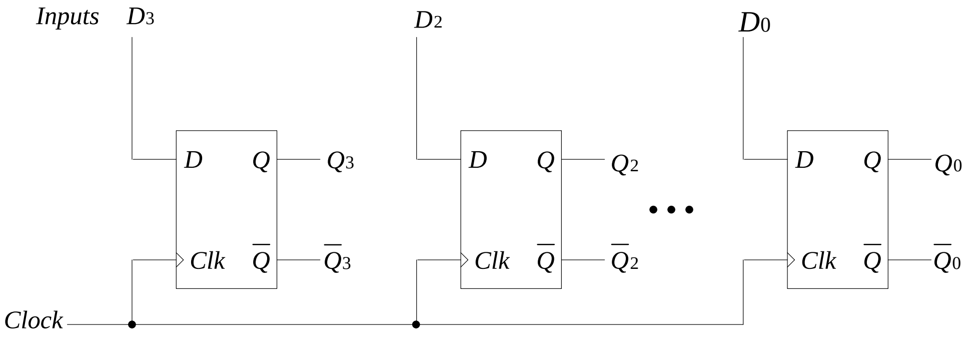
6.3A Workings of JK FF



Mode of Operations	Inputs					Output	
	Asynchronous		Synchronous				
	Preset	Clear	Clock	J	K	Q	Q'
Asynchronous set	0	1	x	x	x		
Asynchronous clear	1	0	x	x	x		
Prohibited	0	0	x	x	x		
No change	1	1	↑	0	0		
Clear	1	1	↑	0	1		
Set	1	1	↑	1	0		
Toggle	1	1	↑	1	1		

Table 6.3 Working of JK FF

Experiment 6.4: 4-bit Register implemented from 4 D-FFs



Input				clock	Output			
D_3	D_2	D_1	D_0		Q_3	Q_2	Q_1	Q_0
0	0	1	1	↑				
0	1	0	1	↑				
0	1	0	0	↑				
1	1	1	1	↑				

Table 6.4 4-bit Register implemented from 4 D-FFs

Questions

6.1 Show how to modify a JK FF to work as a D FF.

6.2 Construct 4-bit register from JK FFs.

Lab 07 Counters I

Counters are very useful in digital circuits. In asynchronous counters, the clock signal is applied to the LSB FF, and all other FFs are clocked by the output of preceding FF. Therefore, the maximum operating speed of asynchronous counter decreases as number of FFs increases. Whereas, in synchronous counters, all FFs are triggered by the same clock signal, therefore, the maximum operating speed is independent of the number of FFs used.

Counters come in many types. A ring counter is an N-bit shift register that recirculates a single '1' continuously, thereby acting as a MOD-N counter. A Johnson counter is a modified ring counter that operates as MOD-2N counter.

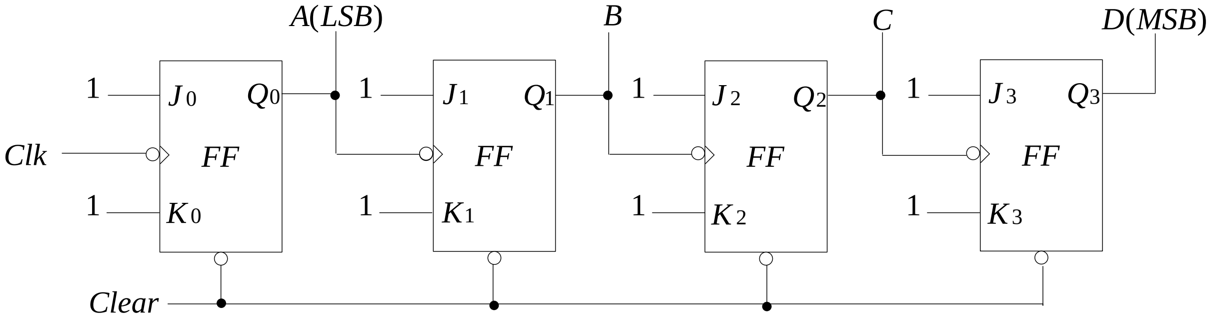
Objectives

- To study different types of counters.

Component lists

- 7408 7474 7476 **(Connect asyn. preset and clear of all FFs to Vcc)**

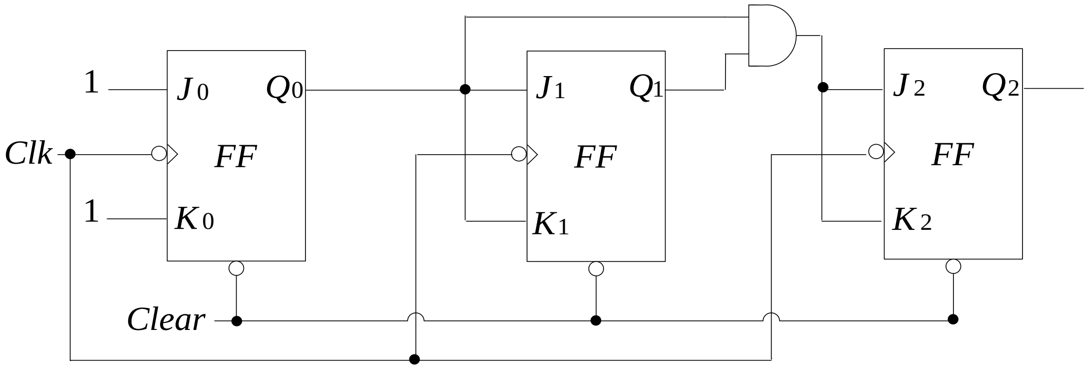
Experiment 7.1: Ripple counter circuits



Clear	Clock pulse #	Output			
		D	C	B	A
0	x	0	0	0	0
1	1				
1	2				
1	3				
1	4				
1	5				
1	6				
1	7				
1	8				
1	9				
1	10				
1	11				
1	12				
1	13				
1	14				
1	15				
1	16				

Table 7.1 Ripple counter circuits

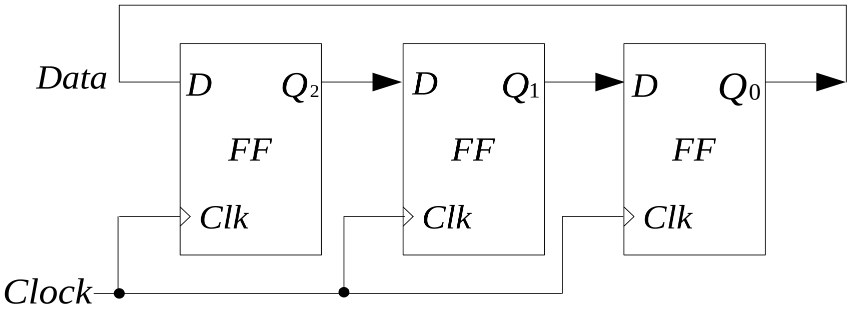
Experiment 7.2: Mod-8 synchronous counter circuits



Clear	Clock Pulse #	Output		
		Q_2 (msb)	Q_1	Q_0
0	x	0	0	0
1	1			
1	2			
1	3			
1	4			
1	5			
1	6			
1	7			
1	8			

Table 7.2 Mod-8 synchronous counter circuits

Experiment 7.3: Ring counter



Clock Pulse	Q_2 (msb)	Q_1	Q_0
Preset/Clear*	1	0	0
1			
2			
3			
4			

Table 7.3 Ring counter

* (asynchronous) preset Q_2/ (asynchronous) clear Q_1 and Q_0

Questions

7.1 Show how to design a synchronous Mod-6 counter using JK FFs.

7.2 Show the circuit of 3-bit Johnson counter.

Lab 08 Counters II

Counters are very useful in digital circuits. In asynchronous counters, the clock signal is applied to the LSB FF, and all other FFs are clocked by the output of preceding FF. Therefore, the maximum operating speed of asynchronous counter decreases as number of FFs increases. Whereas, in synchronous counters, all FFs are triggered by the same clock signal, therefore, the maximum operating speed is independent of the number of FFs used.

Counters come in many types. A MOD-N counter is a counter that counts in the following manner:
-> 0 -> 1 -> 2 -> .. N-1 -> 0 … . A bidirectional counter can also be constructed.

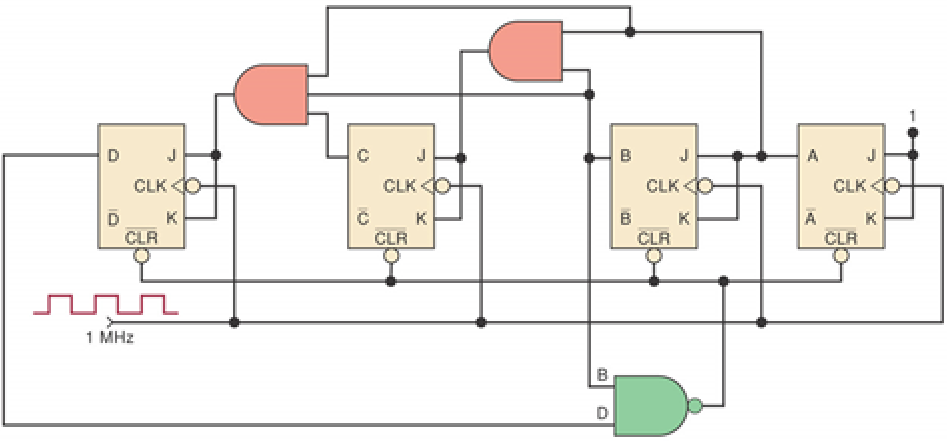
Objectives

- To study different types of counters.

Component lists

- 7400 7404 7408 7411 7432 7476 ++

Experiment 8.1: Asynchronous MOD-10 counter

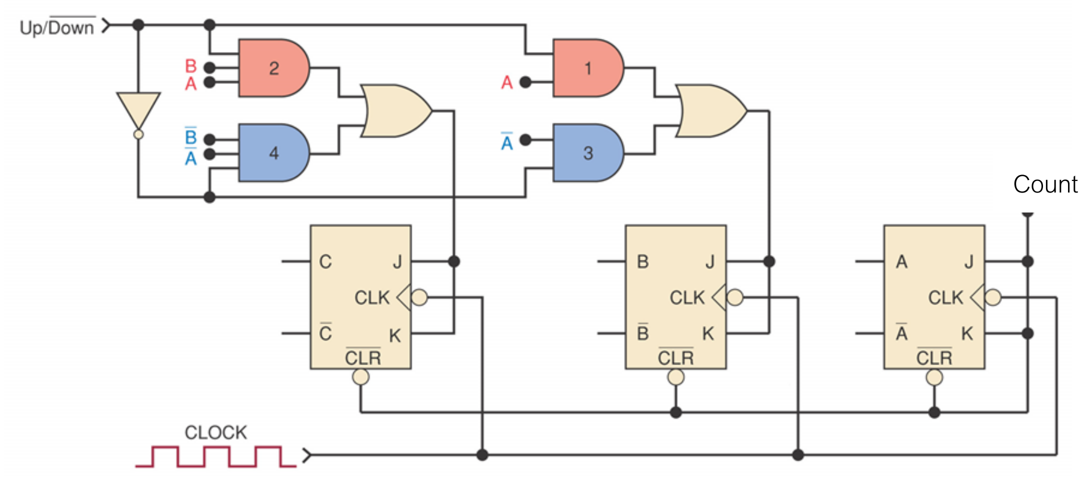


Clock pulse #	Output			
	D	C	B	A
Clear FFs				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

Table 7.1 Ripple counter circuits

Experiment 8.2: Mod-8 synchronous up-down counter

The following circuit can both count up and down (controlled by Up/$\overline{\text{Down}}$ signal). Count signal will first clear all FFs' outputs. Then the circuit is controlled by clock and Up/Down signal.



Count	Up/$\overline{\text{Down}}$	Clock Pulse #	C (msb)	B	A
0	x	x			
1	1	1			
1	1	2			
1	1	3			
1	1	4			
1	0	5			
1	0	6			
1	1	7			
1	1	8			

Table 8.2 Mod-8 synchronous up/down counter

Experiment 8.3: Presettable counter

Modify circuit 8.2 so we can define the starting state of the counter. The signal Count becomes $\overline{\text{Preset}}$/Count. When $\overline{\text{Preset}}$, the external inputs will either asynchronously preset or clear the FFs.

$\overline{\text{Preset}}$/Count	Up/$\overline{\text{Down}}$	Clock Pulse #	C (msb)	B	A
0*	x	x	1	0	1
1	1	1			
1	1	2			
1	1	3			
1	1	4			
1	0	5			
1	0	6			
1	1	7			
1	1	8			

Table 8.3 Presettable counter

* (asynchronous) preset C and A/ (asynchronous) clear B

Questions

8.1 Show the circuit of experiment 8.3.

Lab 09 Shifters

Shifter is a group of FFs arranged so that the binary numbers stored in the FFs are shifted from one FF to the next for every clock cycle. Input into the shifter can be either serial or parallel in. Similarly, output from shifter can be either serial or parallel out. Many ICs provide both types of input methods or output methods. Shift registers are very useful in digital equipments.

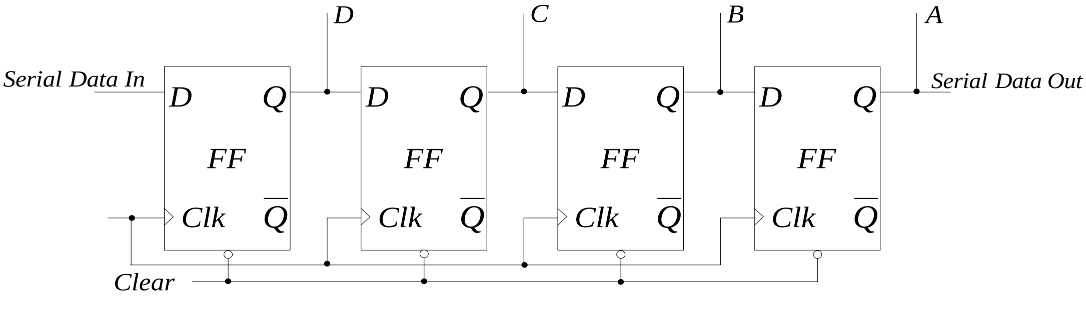
Objectives

- To study the working of shifters.

Component lists

- 7400 7408 7474

Experiment 9.1: Serial-In-Serial-Out (SISO) shift register

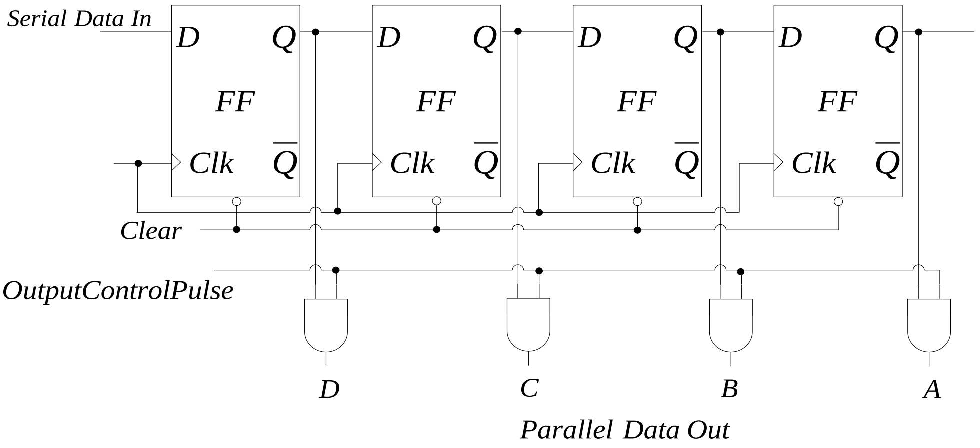


(Asynchronous) Clear	Serial Data In	Clock	Data in Register			
			D	C	B	A
0	x	-	0	0	0	0
1	1*	↑				
1	0	↑				
1	1	↑				
1	0	↑				

Table 9.1 Serial-In-Serial-Out (SISO) Shift register

* Supply 1 at Serial Data In and then apply clock signal

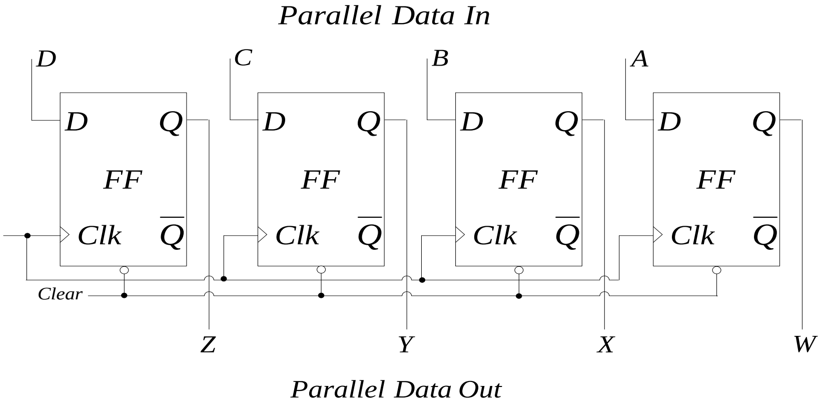
Experiment 9.2: Serial-In-Parallel-Out (SIPO) shift register



(Asynchronous) Clear	Serial Data In	Output Control Pulse	Clock	Data in Register			
				D	C	B	A
0	x	0	-	0	0	0	0
1	1	0	↑				
1	1	0	↑				
1	1	0	↑				
1	1	0	↑				
1	x	1	-				

Table 9.2 Serial-In-Parallel-Out (SIPO) Shift register

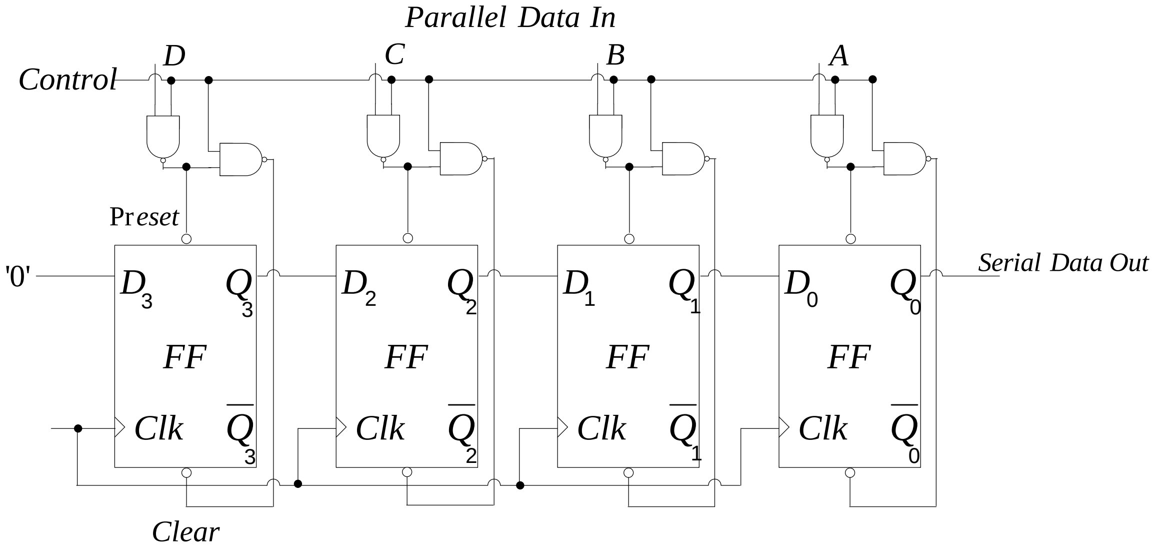
Experiment 9.3: Parallel-In- Parallel-Out (PIPO) shift register



(Asynchronous) Clear	Parallel Input				Clock	Output			
	D	C	B	A		Z	Y	X	W
0	x	x	x	x	-	0	0	0	0
1	0	0	1	1	↑				
1	0	1	1	0	↑				
1	1	0	1	0	↑				
1	1	1	0	1	↑				

Table 9.3 Parallel-In-Parallel-Out (PIPO) Shift register

Experiment 9.4: Parallel-In-Serial-Out (PISO) shift register



Control	Parallel Input (DCBA)	Clock	Data in register			
			Q_3 (MSB)	Q_2	Q_1	Q_0
1	Parallel Data In (=1101)	-	1	1	0	1
0	-	↑				
0	-	↑				
0	-	↑				
0	-	↑				

Table 9.4 Parallel-In-Serial-Out (PISO) Shift register

Questions

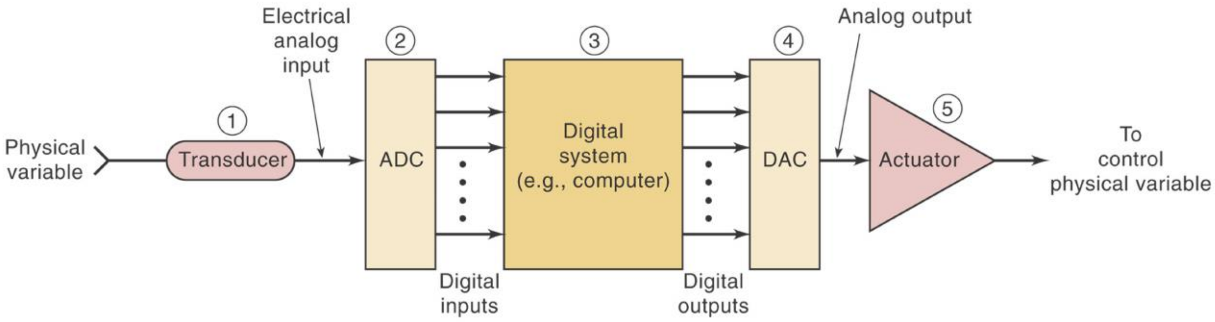
9.1 Explain the usage of shifter in computer communication. Which circuit(s), you think are used? Give some examples.

Lab 10 Analog to Digital and Digital to Analog

Physical variables that we want to measure, such as temperature, pressure, humidity, and so on, are continuously variable quantities. A transducer can be used to translate these quantities into an electrical signal of voltage or current that fluctuates in proportion to the physical variable. These continuously variable voltage or current signals are called analog signals.

A device that takes an analog input voltage and produces a digital output that represents the analog input is called an analog to digital converter (A/D converter). It usually uses a digital to analog converter (D/A converter) as its major component. Therefore, the A/D conversion process is generally more complex and time consuming than the D/A process, and many different methods have been developed and used, e.g. digital ramp A/D, successive approximation A/D, flash A/D, tracking A/D. Digital outputs from an A/D device can now be processed by a digital system (e.g. a computer).

After processing, the digital system produces digital outputs corresponding to the given inputs. These outputs will typically be used to control a monitored system. However, these systems can accept analog control signals. Therefore, a device that takes digital inputs and produces analog outputs that represent the digital inputs must be available. The device is called a digital to analog converter (D/A converter).



The above figure shows that Analog-to-digital converter (ADC) and digital-to-analog converter (DAC) are used to interface a computer to the analog world so that the computer can monitor and control a physical variable.

Objectives

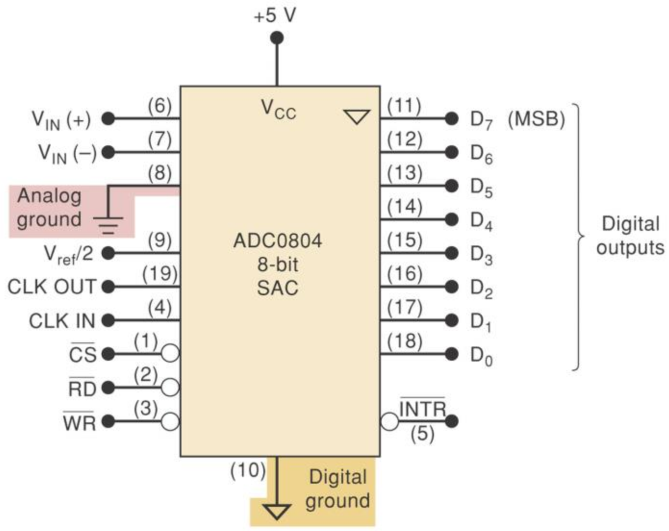
- To study the process and circuits of Analog to Digital conversion (ADC) and Digital to Analog conversion (DAC).

Component lists

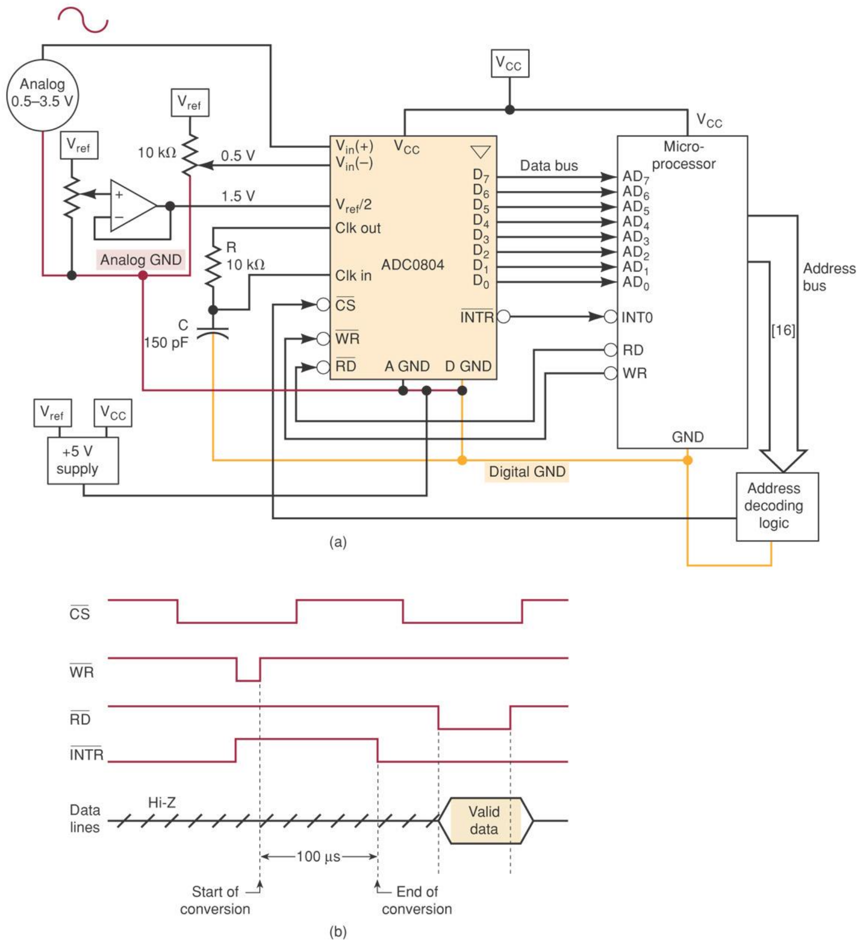
ADC 080X, DAC 0808
VR 2K, R10K(x3), 2.5K, and 1K
C 150pF and 15pF
IC 741 (OpAmp)
Power Supply DC ±15V and ±5V
2-CH Oscilloscope

ADC

Pin Layout of ADC0804 eight-bit successive-approximation ADC with tristate outputs. The numbers in parentheses are in the IC's pin number



Example of an application of an ADC0804 (a); typical timing signals during data acquisition (b).



(a)

(b)

Experiment 10.1: ADC

Try how the DAC works before the experiment. Supply Vin by turning the knob in clockwise direction (notice the markers). Record the result in the table.

Vin (volt)	Output							
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
16.67% (= 0.83 volt)								
33.33% (= 1.67 volt)								
50.00% (= 2.50 volt)								
66.67% (= 3.33 volt)								
83.33% (= 4.17 volt)								
100.00% (= 5.00 volt)								

Table 10.1 ADC

Explain the working of the above ADC circuit ____

____

____

____

____

____

____

Experiment 10.2: DAC

Try how the DAC works before the experiment. Then follow these steps ..

1. Connect inputs (on-off switches) to the DAC (D7..D0), i.e. switch 7 to D7 and so on.
2. Connect the A-out and ground from DAC unit to A-in and ground (respectively) of the ADC unit.
3. Connect data outputs of the DAC (D7..D0) to the logic LED display 7..0, i.e. D7 to LED 7 and so on.
4. Record the result in the table.

Input (D7..D0)	Output (Volts)
0000 0000	
0111 0000	
1000 0000	
1000 1000	
1111 1111	

Table 10.2 DAC

Explain the working of the above DAC circuit ____

____

____

____

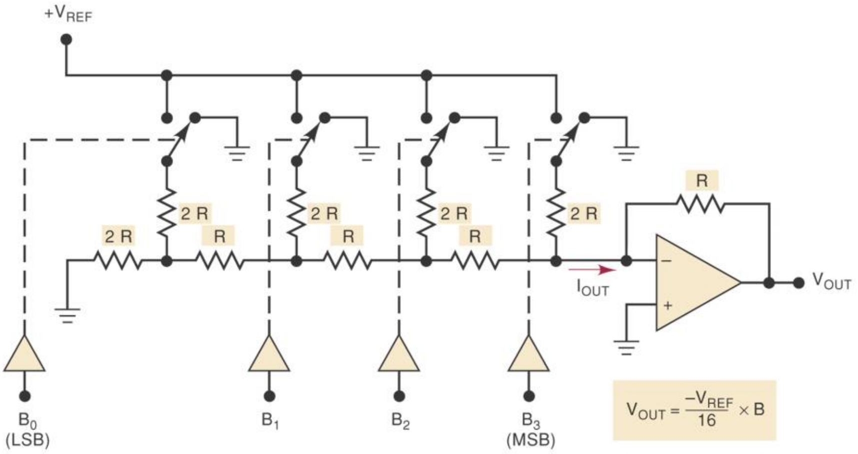
____

____

____

Questions

10.1 Explain R-2R ladder DAC.



10.2 Explain SAC ADC

10.3 What the step-size value of ADC in the experiment 10.1?