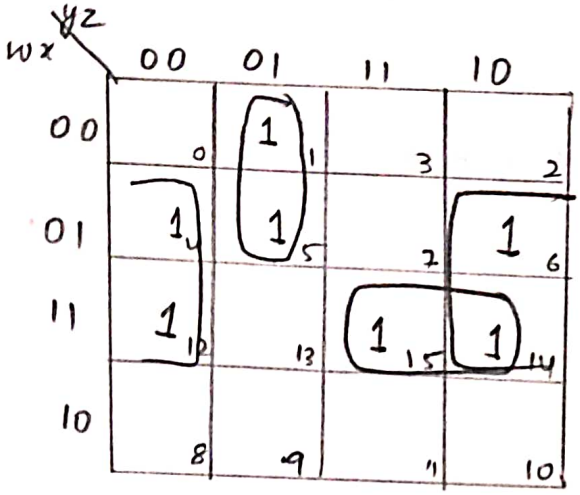
11 May 21

NAME : Ananya Prasad
Reg No : 20BCE10093
Semester : II

Subject : ECE2002
Slot : B11 + B12 + B13
FACULTY : Dr Jitendra Kumar

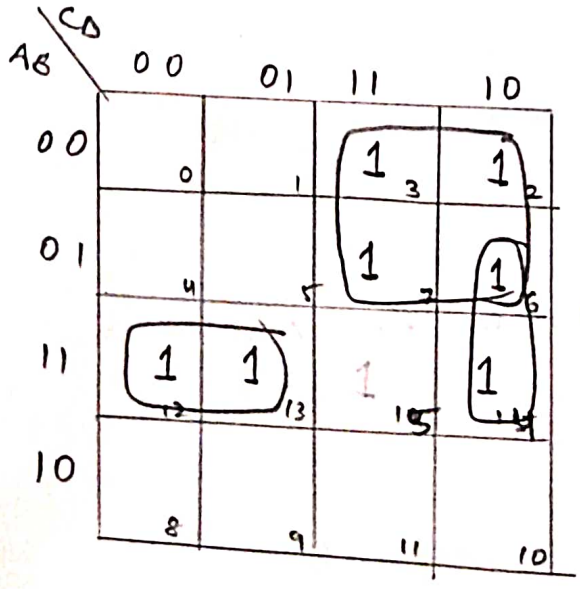
1(b)(i) $F(w,x,y,z) = \Sigma(1,4,5,6,12,14,15)$



2 pairs, 1 quad

$\Rightarrow \bar{w}\bar{y}z + x\bar{z} + wxy$

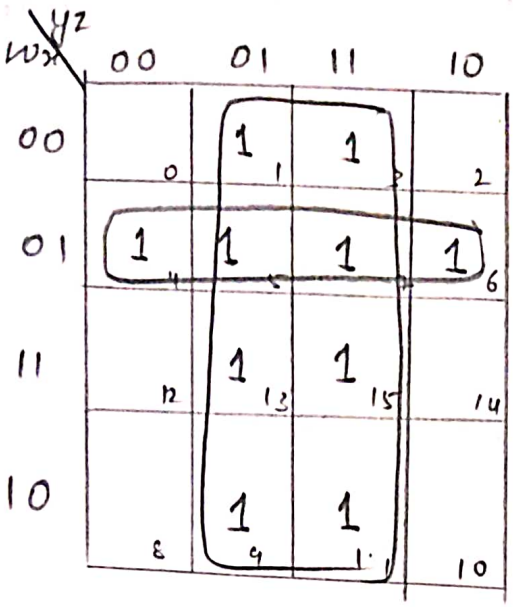
(ii) $F(A,B,C,D) = \Sigma(2,3,6,7,12,13,14)$



2 pairs, 1 quad

$\Rightarrow \bar{A}C + AB\bar{C} + BC\bar{D}$

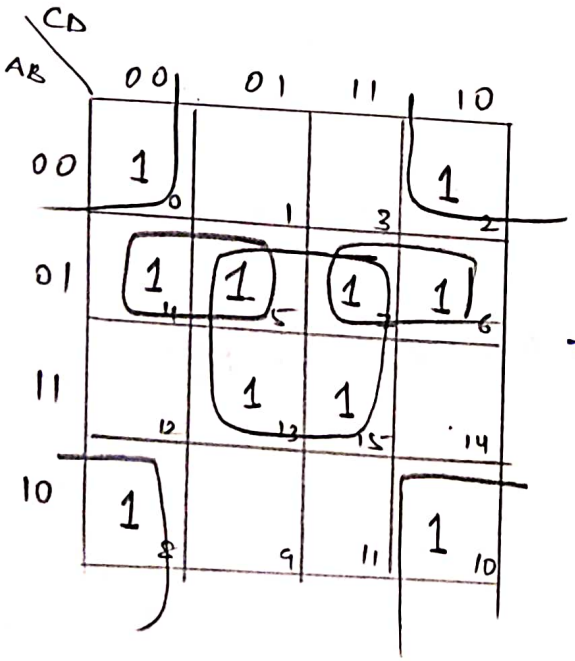
(iii) $F(w, x, y, z) = \Sigma(1, 3, 4, 5, 6, 7, 9, 11, 13, 15)$



1 oct, 1 quad

$\Rightarrow z + \bar{w}x$

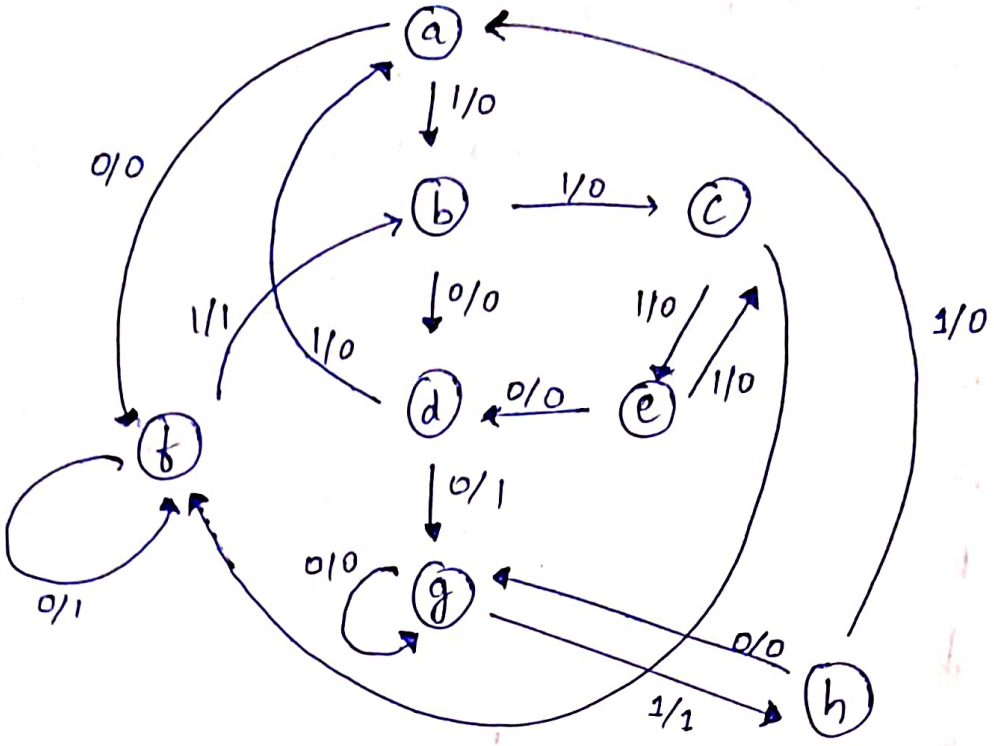
(iv) $F(A, B, C, D) = \Sigma(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$



2 quads, 2 pairs

$\rightarrow \bar{B}\bar{D} + BD + \bar{A}\bar{D}$

2(a)



There are 8 states ⇒

a = 0 0 0

b = 0 0 1 (state assignment)

c = 0 1 0

d = 0 1 1

e = 1 0 0

f = 1 0 1

g = 1 1 0

h = 1 1 1

(If next state & output of two present states are same, then we can eliminate one)

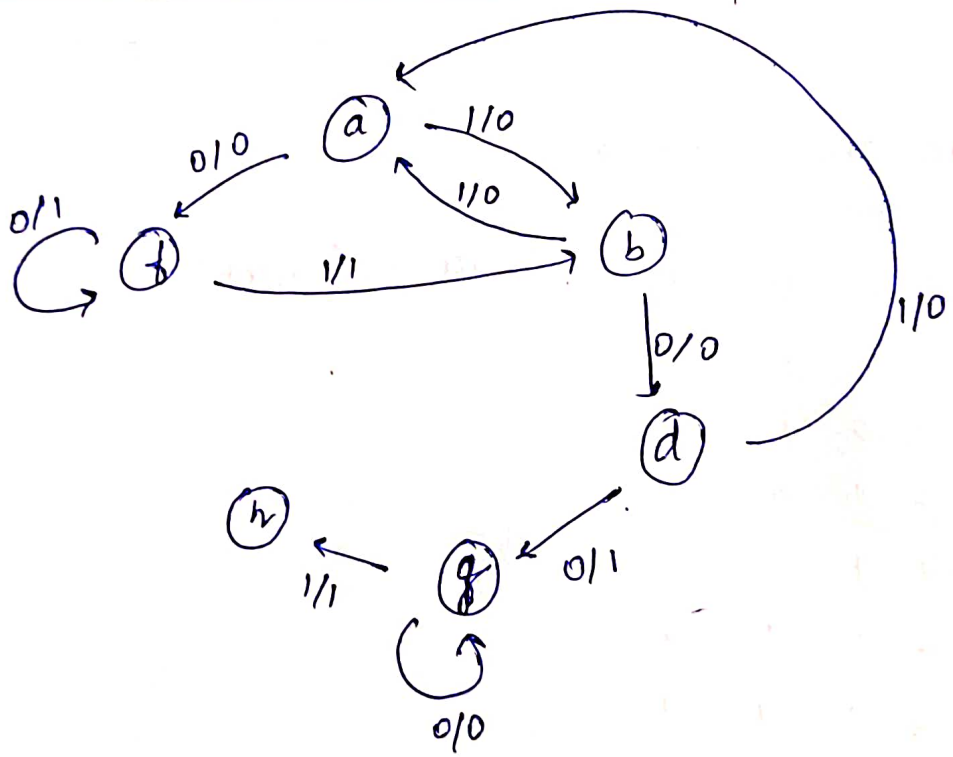
PRESENT STATE	NEXT STATE $x=0$	NEXT STATE $x=1$	OUTPUT $x=0$	OUTPUT $x=1$
⇒ a	f	b	0	0
→ b	d	c = a	0	0
⇒ c	f	c = b	0	0
⇝ d	g	a	1	0
→ e	d	c = a	0	0
f	f	b	1	1
g	g	h = a	0	1
⇝ h	g	a	0	0

$\therefore b = e$; $d = h$; $a = c$.

(ii) Reduced state table

Present state	Next State		Output	
	$x=0$	$x=1$	$x=0$	$x=1$
a	f	b	0	0
b	d	a	0	0
d	g	a	1	0
f	f	b	1	1
g	g	h	0	1

(iii)



3(a) PORT DECLARATION

Port declaration establishes the interface of the object to the outside word.

Each port in the port list is defined as input, output or inout based on the port signal's direction.

It can be named accordingly, but it should be any identifier which is not a reserved word.

It can have any declared or predefined datatype

If a port declaration includes the net or variable types, then that port is considered completely declared. It is illegal to declare the same port in a net or variable type declaration.

If the port declaration does not include a net or variable type, the port can be declared again in a net or variable type declaration.

EXAMPLE

For a full adder,

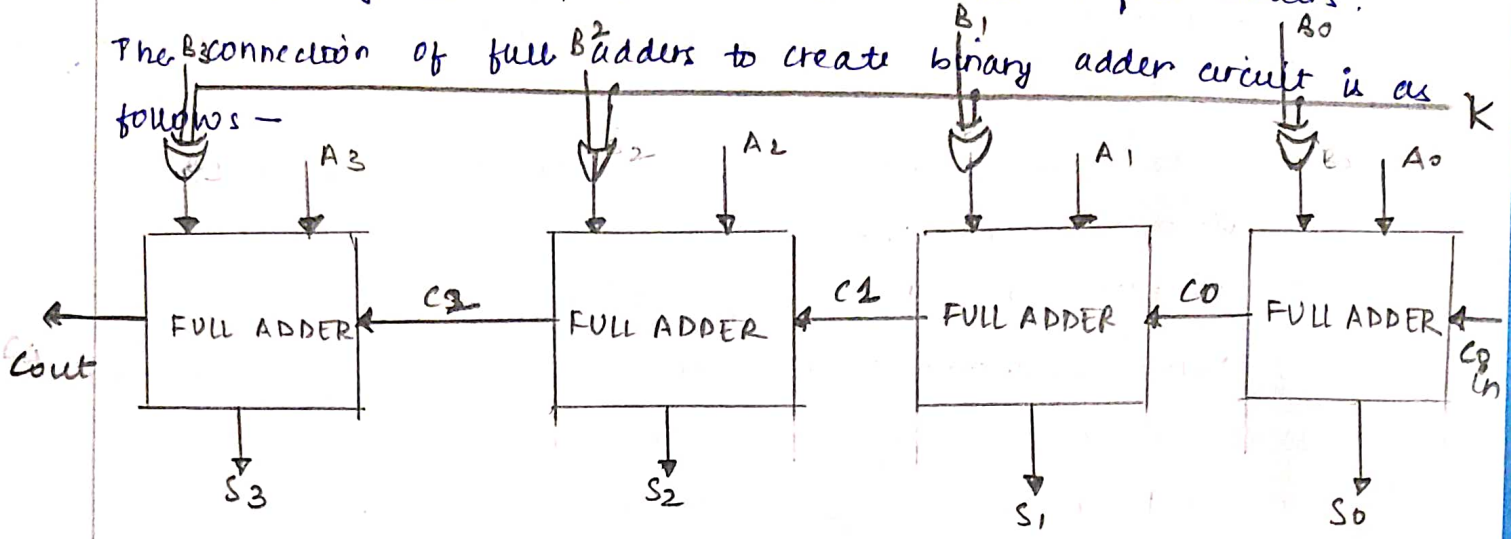
```
module full add 4C sum, c_out, a, b, c_in);  // port declaration starts
output [3:0] sum;
output c_out;
input [3:0] a, b;
input c_in;                                  // port declaration ends
<module internals>
end module
```

To declare a port small example: (syntax)

```
ENTITY test IS
        PORT (name : mode  data_type );
END test;
```

(4) 4-bit adder circuit :

Binary adders are implemented to add two binary numbers. To add 4-bit binary numbers, we will need to use 4-full adders. The connection of full adders to create binary adder circuit is as follows –



4-bit binary adder has a sequence of 4 full adders

First full adder has input (C_0); input A_0 is directly put in full adder. The third input is is $B_0 \oplus K$. The outputs are S_0 (sum/difference) and carry (C_1).

If $K = 1$, $B_0 \oplus K = \overline{B}_0$ $\therefore$ $A + \overline{B}_0$ is the operation

2's complement subtraction for A & B is $A + \overline{B}$ $\therefore$ for $K = 1$, subtraction takes place

similarly if $K = 0$, $A + B$ gives binary addition.

Then, C_0 is serially passed in second full adder. S_0 is recorded. Similarly, A_1, A_2, A_3 are added. Similarly, B_1, B_2, B_3 all exored with K are added successively. C_3 becomes total carry. S_1, S_2, S_3 become to record to form result with S_0.

ex. BINARY

$A = 010$; $B = 011$

1st adder → $A_0 + B_0 = 0 + 1 = 1$

$S_0 = 1$; $C_1 = 0$

∴ $S_1 = 0$ with $C_2 = 1$

$S_2 = 1$ with $C_2 = 0$

∴ $A = 010 = 2$

$B = 011 = 3$

Sum $= 0101 = 5$

For $k = 1$

$B_0 \oplus k = \bar{B}_0$ and $C_0 = k = 1$

for $S_0 = 1$ and $C_1 = 0$

Similarly,

$S_1 = 1$ and $C_2 = 0$

$S_3 = 1$ and $C_3 = 1$

here,

$A = 010 = (2)_{10}$

$B = 011 = (3)_{10}$

difference $= 1111 = -1$

=

Ex 2:

4-bit binary adder has a sequence of 4 full adders.

We are going to use RTL (Register Transfer level) implementation.

State	Previous Carry	Augend bits	Addend bits	Sum	Next Carry
0	$C_0 = 0$	$A_0 = 1$	$B_0 = 1$	$S_0 = 0$	$C_1 = 1$
1	$C_1 = 1$	$A_1 = 1$	$B_1 = 1$	$S_1 = 1$	$C_2 = 1$
2	$C_2 = 1$	$A_2 = 1$	$B_2 = 1$	$S_2 = 1$	$C_3 = 1$
3	$C_3 = 1$	$A_3 = 0$	$B_3 = 1$	$S_3 = 0$	$C_4 = 1$

For example we have considered two binary numbers 7 and 15.

both have previous carry = 0

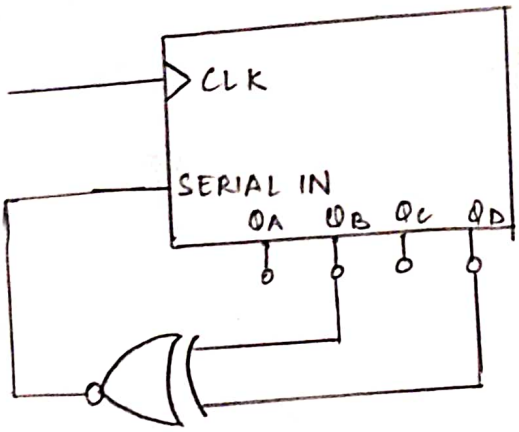
$\therefore$ Sum of two binary numbers 7 and 15 $\Rightarrow C_4 S_3 S_2 S_1 S_0$

$= (10110)_2$

$= 16 + 4 + 2 = (22)_{10}$

In the final result, we have appended C_4 in the front of the sum of digits to accomodate overflow bit from the binary addition flow.

5 SIPO ⇒ "0 0 0 0"



Let $Q_A = A$

$Q_B = B$

$Q_C = C$

$Q_D = D$

clk	I/P	A	B	C	D	
0	0	0	0	0	0	} 1 st
	1	1	0	0	0	} 2 nd
	1	1	1	0	0	} 3 rd
	0	0	1	1	0	} 4 th
	0	0	0	1	1	} 5 th
	0	0	0	0	1	} 6 th
	0	0	0	0	0	

At each clock pulse, $I/p = \overline{B \oplus D}$

∴ Output after 6 clock pulses is 0000.