Paper 7, TDC Part-3
Chapter– 4, Combinational Logic Design
Lecture - 15

By:

Mayank Mausam

Assistant Professor (Guest Faculty)

Department of Electronics

L.S. College, BRA Bihar University,

Muzaffarpur, Bihar

Combinational Logic Design

The subtraction process is explained in detail as below.

Case 1 :-> $A_n = 0$, $B_n = 0$ & $Bw_{n-1} = 0$,

$B_n + Bw_{n-1} = 0 + 0 = 0$

So, $A_n (0) - \{B_n + Bw_{n-1}\}(0) = 0$

i.e. $0 - 0 = 0$

Case 2 :-> $A_n = 0$, $B_n = 0$ & $Bw_{n-1} = 1$

$B_n + Bw_{n-1} = 0 + 1 = 1$

So, $A_n (0) - \{B_n + Bw_{n-1}\}(1)$

$= 0 - 1$

will give $D_n = 1$ with $Bw_n = 1$

Combinational Logic Design

borrowing 1 from higher bit make A

$A = (0) + (10)_2 = (10)_2 =$

Now, $(10)_2 - (1)_2 = (2)_{10} - (1)_{10} = (1)_{10}$

$= (1)_{10} = (1)_2$

So, Case 2 results in $D_n = 1$ & $Bw_n = 1$.

Case 3: Same as the result for case 3 i.e.

$A_n = 0$, $B_n = 1$ & $Bw_{n-1} = 0$; $B_n + Bw_{n-1} = 1$

$\Rightarrow A_n(0) - \{B_n + Bw_{n-1}\}(1)$

$\Rightarrow 0 - 1$ will results $D_n = 1$ and

$Bw_n = 1$

Combinational Logic Design

Case 4 :→ $A_n = 0$, $B_n = 1$ & $Bw_{n-1} = 1$

then $B_n + Bw_{n-1} = 1 + 1 = (10)_2$

Now, $A_n(0) - \{B_n + Bw_{n-1}\}(10)_2$

Here again A_n has to borrow 1 from higher bit. On borrowing '1' from higher bit A_n will become $(10)_2$ then

$A_n (10)_2 - \{B_n + Bw_{n-1}\}(10)_2$

will results $D_n = 0$ & $Bw_n = 1$

Case 5 :→ $A_n = 1$, $B_n = 0$ & $Bw_{n-1} = 0$

then $B_n + Bw_{n-1} = 0 + 0 = 0$

So, $A_n(1) - \{B_n + Bw_{n-1}\}(0)$

$\Rightarrow 1 - 0 = 1$

i.e. $D_n = 1$ & $Bw_n = 0$

Combinational Logic Design

Case 6 :- $A_n = 1$, $B_n = 0$ & $Bw_{n-1} = 1$

~~Will be same as in case 5.~~

then, $B_n + Bw_{n-1} = 0 + 1 = 1$

So, $A_n(1) - \{B_n + Bw_{n-1}\}(1)$

$: 0$ i.e. $D_n = 0$ & $B_n = 0$

Case 7: $A_n = 1$, $B_n = 1$ & $Bw_{n-1} = 0$

Will be same as in case 6.

i.e. $B_n + Bw_{n-1} = 1 + 0 = 1$

So, $A_n(1) - \{B_n + Bw_{n-1}\}(1)$

$= 0$ i.e. $D_n = 0$ & $B_n = 0$

Combinational Logic Design

Case 8: $A_n = 1$, $B_n = 1$ & $B_{w(n-1)} = 1$

i.e. $B_n + B_{w(n-1)} = 1 + 1 = (10)_2$

So, $A_n(1) - \{B_n + B_{w(n-1)}\}(10)_2$

Here A_n will borrow 1 from higher bit and on borrowing 1 from higher bit A_n will become $[1 + (10)_2] = (11)_2$

Now, $A_n(11)_2 - \{B_n + B_{w(n-1)}\}(10)_2$

$= (11)_2 - (10)_2$

$= (3)_{10} - (2)_{10}$

$= (1)_{10}$

$= (01)_2$

So $D_n = 1$ & $B_n = 1$.

Combinational Logic Design

For realization of Full Subtractor using gates let us plot K-Map for D_n & K-Map for B_w.

So from truth table of Full Subtractor

D_n

A_n \ $B_n B_{wn-1}$	00	01	11	10
0		1		1
1	1		1	

So, $D_n = \bar{A}_n \bar{B}_n B_{wn-1} + \bar{A}_n B_n \bar{B}_{wn-1} + A_n \bar{B}_n \bar{B}_{wn-1} + A_n B_n B_{wn-1}$

$D_n = A_n \oplus B_n \oplus B_{wn-1}$

Bwn

Combinational Logic Design

Bw_n

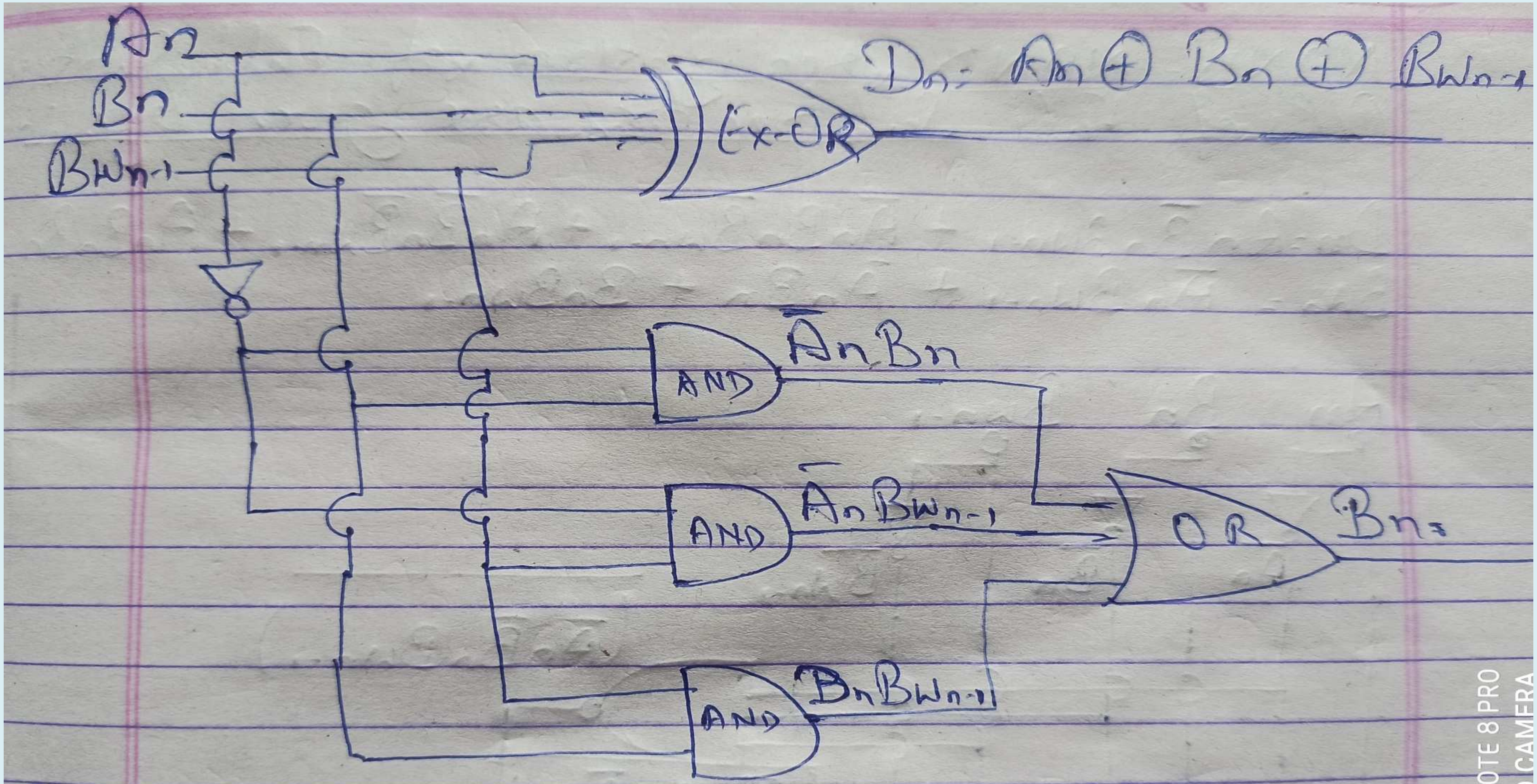
A_n \ $B_n Bw_{n-1}$	00	01	11	10
0		1	1	1
1			1	

$$Bw_n = \bar{A}_n Bw_{n-1} + \bar{A}_n B_n + B_n Bw_{n-1}$$

So full subtractor can be realized using Ex-OR gates, AND gates, NOT gates & OR gate as shown in figure FS·1

NAND-NAND realization of full subtractor is shown in FS.2.

Combinational Logic Design



FS.1 Realization of Full Subtractor using gates.

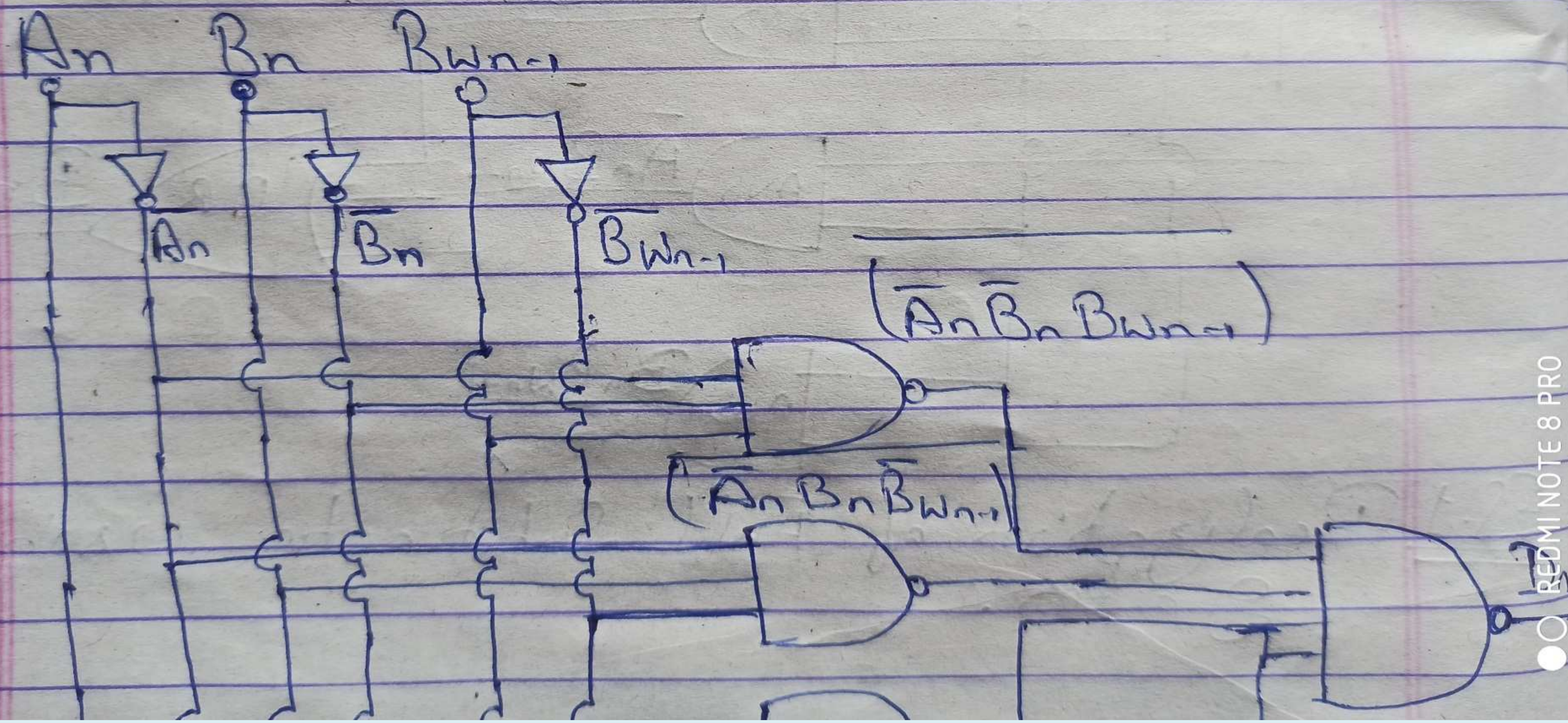
Combinational Logic Design

NAND - NAND realization of Full Subtractor

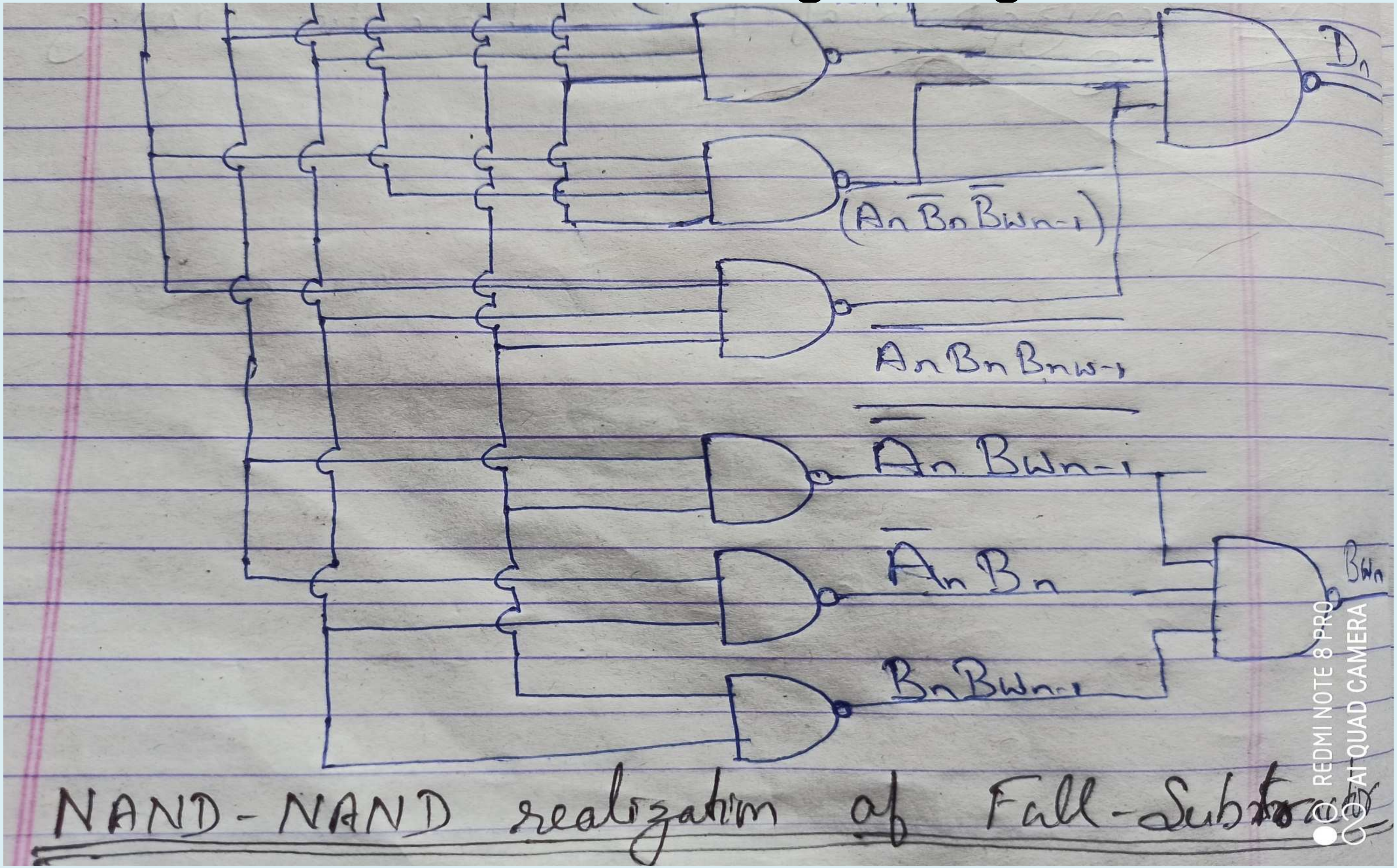
O/P's of Full Subtractor are given by :-

$$D_n = \bar{A}_n \bar{B}_n B_{wn-1} + \bar{A}_n B_n \bar{B}_{wn-1} + A_n \bar{B}_n \bar{B}_{wn-1} + A_n B_n B_{wn-1}$$

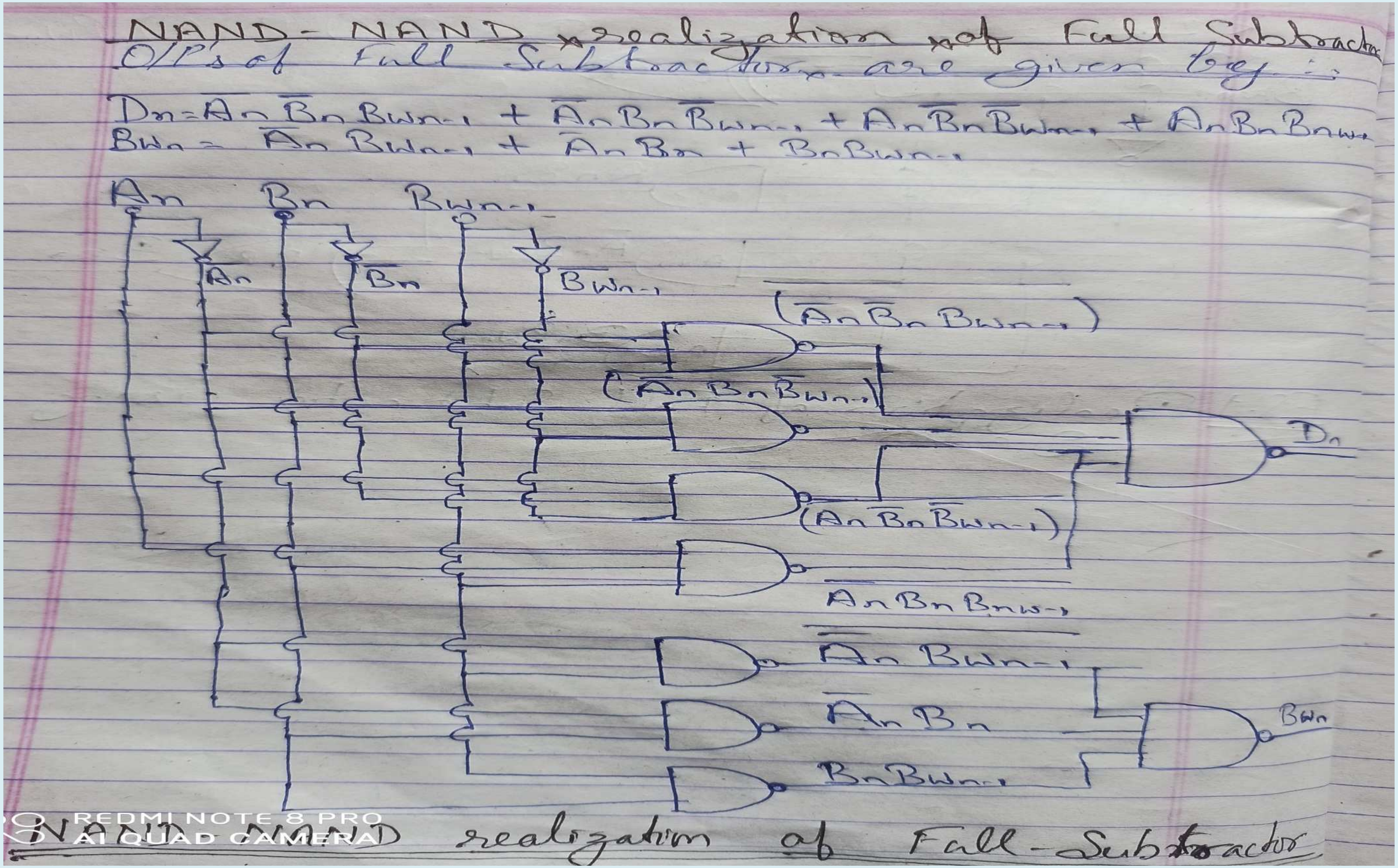
$$B_{wn} = \bar{A}_n B_{wn-1} + \bar{A}_n B_n + B_n B_{wn-1}$$



Combinational Logic Design



Combinational Logic Design



Combinational Logic Design

Refer book- Modern Digital Electronics by RP Jain.

Thank You