TSN1101 Computer Architecture and Organisation

00 } Shift
11
10 A←A-M & Shift
01 A←A+M & Shift

① Number Systems and Codes

Number Systems

Decimal ⟶ (repeated ÷) Binary / Octal / Hexadecimal
⟵ expand

Fraction – repeated multiplication ↑↓

Binary: + − × ÷

Octal – Binary – Hexadecimal
+ − + −

1 Octal digit = 3 Binary digit
1 Hex digit = 4 Binary digit

Codes

① Decimal – NBCD (8421)
15 – 0001 0101

② Decimal – 2421
15 – 0001 1011

③ Decimal – XS3
Decimal + 3 (0011) → 8421
Decimal ← 8421 − 3 (0011)

④ Binary – Gray code

B 0100 → G 0010
G 0110 → B 0011
B 0110 → G 1101
G 0101 → B 1001

Error Detection

Parity Protocol	1's	Parity Bit
Odd	Odd	0
	Even	1
Even	Odd	1
	Even	0

② Data Representation and Arithmetic

Sign-Magnitude

LSB	
0	+
1	−

+0 and −0.

Ones Complement

Complement to get −ve. +0 and −0.

Twos Complement

Complement and add 1 to get −ve.

Overflow Rule: two numbers of same sign added; result of opposite sign.

Subtraction Rule: A − B = A + (−B)

Floating Point:

1 bit	8 bits	23 bits
Sign Bit	Biased Exponent	Significand
	Actual Exponent + 127 → Binary	

+ − × ÷ (make them have same exponent)

M = 0101, Q = 1010, −M = 1011

Booth Algorithm: 5 × (−6) = −30

M Register	A Register	Q Register	Q−1	
0101	0000	1010	0	
	0000 + 1011	0101	0	Shift
	1011	0101	0	A←A-M
	1101 + 0101	1010	1	Shift
	0010	1010	1	A←A+M
	0001 + 1011	0101	0	Shift
	1100	0101	0	A←A-M
	1110	0010	1	Shift

Product

Finding Product

1110 0010
1's Complement
→ 0001 1101
2's Complement
→ 0001 1110 (30)
∵ Sign bit is 1
∴ −30

0010 1010
∵ Sign bit is 0
∴ 42

③ Introduction to Digital Logic and Boolean Algebra

Logic Gates

NOT		
AND		
OR		
NAND	$\overline{AB}$	[00, 01, 10]
NOR	$\overline{A+B}$	[00]
XOR	$A \oplus B$	[10, 01]
XNOR	$\overline{A \oplus B}$	[00, 11]

Boolean Algebra

$A+0=A$ $\quad A \cdot A = A$
$A+1=1$ $\quad A \cdot \overline{A} = 0$
$A \cdot 0 = 0$ $\quad \overline{\overline{A}} = A$
$A \cdot 1 = A$ $\quad A + AB = A$
$A+A=A$ $\quad A + \overline{A}B = A+B$
$A+\overline{A}=1$ $\quad (A+B)(A+C) = A+BC$

DeMorgan's Theorem

$\overline{X \cdot Y} = \overline{X} + \overline{Y}$

$\overline{X+Y} = \overline{X} \cdot \overline{Y}$

④ Design of Combinational Logic Circuits

Product of Sums & Sum of Products

		Output	A	$\bar{A}$	Term
SOP	AND-OR	1	1	0	Product/Min
POS	OR-AND	0	0	1	Sum/Max

Minimal to Standard Form:

$F = (A+B+C')(A'+C)$ ×

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

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

$\overline{\overline{SOP}}$ → NAND-NAND

$\overline{POS}$ → NOR-NOR

Karnaugh Map

① Grouping 1's (SOP)/0's (POS) - 1, 2, 4, 8

② OR (SOP) / AND (POS) the terms.

③ Don't care: X could be 1/0.

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

Design Logic Circuit

① Truth Table

② SOP (AND-OR) / POS (OR-AND).

③ Simplify - Boolean Algebra / K-Map.

④ Implement with logic gates.

⑤ Standard Combinational Logic Circuits

Adder / Subtractor

Half-Adder

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$S = \bar{A}B + A\bar{B} = A \oplus B$

$C = AB$

Full-Adder

A	B	C	S	C
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

A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$S = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$

$= A \oplus B \oplus C$

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	0

$C = AC + BC + AB$

Half-Subtractor

A	B	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = \bar{A}B + A\bar{B} = A \oplus B$

$B = \bar{A}B$

Full-Subtractor

A	B	C	D	B
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

A \ BC	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$D = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$

$= A \oplus B \oplus C$

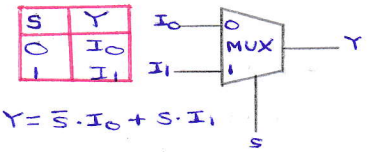
A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$B = \bar{A}C + BC + \bar{A}B$

Multiplexer / Demultiplexer

2:1 MUX

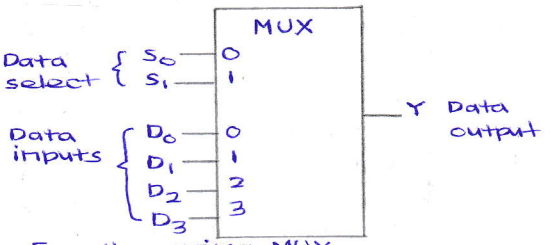
S	Y
0	I_0
1	I_1



$Y = \bar{S}\cdot I_0 + S\cdot I_1$

4:1 MUX

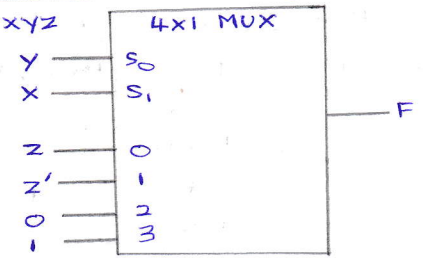
S_1	S_2	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Implementation of Boolean Function using MUX

$F = x'y'z + x'yz' + xyz' + xyz$

x	y	z	F	
0	0	0	0	F = z
0	0	1	1	
0	1	0	1	F = z'
0	1	1	0	
1	0	0	0	F = 0
1	0	1	0	
1	1	0	1	F = 1
1	1	1	1	



1:4 DEMUX

S_1	S_2	Y_0	Y_1	Y_2	Y_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

BCD Addition

① If 4-bit sum > 9, add 6 (0110).

② If a carry results after added 6, add carry to next 4-bit.

478	0100	0111	1000
+137	0001	0011	0111
	0110	1011	1111
	0000	0110	0110
615	0110	0001	0101

Decoder / Encoder

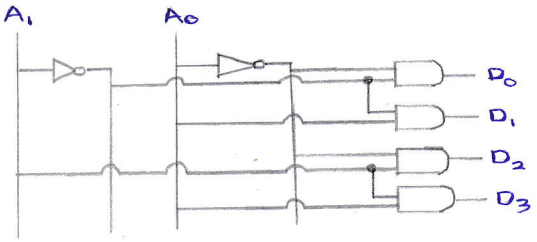
① Active-LOW output
- only one low output
- use NAND and NOT

② Active-HIGH output
- only one high output
- use AND and NOT

2x4 Decoder

① Active-HIGH output

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



② Active-LOW output

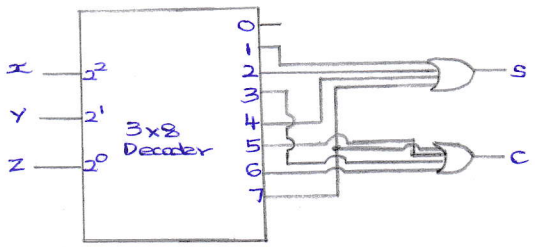
A_1	A_0	D_0	D_1	D_2	D_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Decoder with Logic Gates

Full Adder:

$S(x,y,z) = \Sigma m(1,2,4,7)$

$C(x,y,z) = \Sigma m(3,5,6,7)$



8x3 Encoder (Octal-to-Binary Encoder)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

4x2 Priority Encoder

D_0	D_1	D_2	D_3	x	Y
0	0	0	0	X	X
1	0	0	0	0	0
X	1	0	0	0	1
X	X	1	0	1	0
X	X	X	1	1	1

$D_0 D_1$ \ $D_2 D_3$	00	01	11	10
00	X	1	1	1
01		1	1	1
11		1	1	1
10		1	1	1

$x = D_2 + D_3$

$D_0 D_1$ \ $D_2 D_3$	00	01	11	10
00	X	1	1	1
01	1	1	1	1
11	1	1	1	1
10		1	1	1

$Y = D_3 + D_1 D_2'$