Number Systems and Codes

Converting ( )$_r$ → ( )$_{10}$

$(2F.8)16 = 2 \times 16^1 + F \times 16^0 + 8 \times 16^{-1}$

Converting ( )$_{10}$ → ( )$_2$

Repeated Radix Division

$(74)_{10}$ ( )$_8$

74/8 = 9 r 2
9/8 = 1 r 1
1/8 = 0 r 1

↑ Collect Digits Bottom to Top

Repeated Radix Multiplication

$(0.2)_{10}$ ( )$_8$

.2 x 8 = 1.6 | 1
.6 x 8 = 4.8 | 4
.8 x 8 = 6.4 | 6
.4 x 8 = 3.2 | 3
.2 x 8 = 1.6 | 1
.6 x 8 = 4.8 | 4

↓ Collect Digits Top to Bottom

SM • MSB = Sign

0 = + 1 = –

-1 in 5-bit = 1 0 0 0 1
(neg) $2^3\ 2^2\ 2^1\ 2^0$

2's Complement

Flip L → R till first 1

(-10) → ( )$_{2CNS}$

0000 1010
1111 0110
Ignore carry bit

Grouping

( )$_2$ ↔ ( )$_8$ = Groups of 3

( )$_2$ ↔ ( )$_{16}$ = Groups of 4

(001)(011)(100)(101).(111)

Addition/Subtraction

Line up radix points

Multiplication

```
  1011.10
x   10.01
---------
  1011.11
 0000
00
000
10111000
---------
11001.1110
```

Subtraction

Overflow = Incorrect Sign
Carry In = Carry Out

```
  11100111 - 01011000
  10101000
+ 11100111   Ignore
----------   Carry
 10001111
```

Number Ranges

2CNS: $[-2^{n-1}, 2^{n-1} - 1]$

SM/1CNS: $[-(2^{n-1} - 1), 2^{n-1} - 1]$

BCD Code

Each Decimal = 4 bits`

Only counts up to 9

Add 6 to numbers >9 or that have carry out of 4-bit

Gray Code

Only one bit changes from one code to next

( )$_2$ → Gray

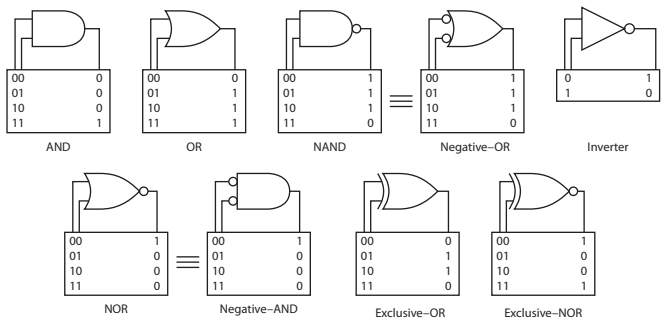
L → R: XOR each pair of bits

1 0 1 1 0 1 0 0
1 1 1 0 1 1 1 0

Gray → ( )$_2$

L → R: XOR each pair of bits

1 1 1 1 1 1 1 0
1 0 1 1 0 1 0 0



Boolean Algebra

Commutative:	$A + B = B + A$	$A * B = B * A$
Associative:	$A + (B + C) = (A + B) + C$	$A * (B * C) = (A * B) * C$
Distributive:	$A * (B + C) = A * B + A * C$	$A + (B * C) = (A + B) * (A + C)$
Distributive:	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$

1) $A + 0 = A$
2) $A + 1 = 1$
3) $A \cdot 0 = 0$
4) $A \cdot 1 = A$
5) $A + A = A$
6) $A + \overline{A} = 1$
7) $A \cdot A = A$
8) $A + \overline{A} = 0$
9) $\overline{\overline{A}} = A$
10) $A + AB = A$
11) $A + \overline{A}B = A + B$
12) $(A + B)(A + C) = A + BC$

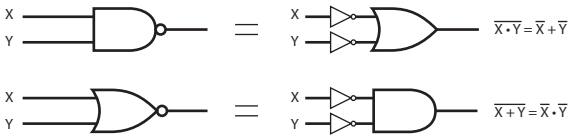
Logic Gates

Standard Form of Functions

SOP = minterms (1s) → Complement = 0

POS = maxterms (0s) → Complement = 1

Negation Indication → De Morgan's maintains equivalency - Flip complements, operators, use ( )



Getting Minterms/Maxterms from Logic Expression

AB ($\overline{BC}$ + BD)
$AB\overline{B}C + ABBD$
ABD (C + $\overline{C}$)
$\sum m(13, 15)$

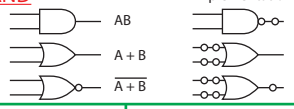
Distributive Law
Rule 7

(A + $\overline{B}$)(A + C) C
(A + $\overline{B}$ + 0)(A + 0 + C) C
(A + B + $C\overline{C}$)(A + $B\overline{B}$ + C) C

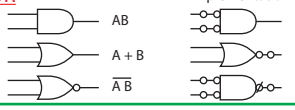
For Maxterms: + 0 = $A\overline{A}$

For Maxterms: x 1 = (A + $\overline{A}$)

NAND

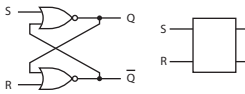


NOR



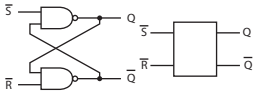
Latches: S – R Latch

Active High = 2 NOR Gates



S	R	Q^+	
0	0	Q → NC	No Change
0	1	0	Reset to 0
1	0	1	Set to 1
1	1	0*	Instability

Active Low= 2 NAND Gates



$\overline{S}$	$\overline{R}$	Q^+	
0	0	1*	
0	1	1	Set to 1
1	0	0	Set to 0
1	1	Q → NC	No Change

Gated S – R Latch

EN	S	R	Q^+	
0	x	x	Q	
1	0	0	Q	EN = 1: Function
1	0	1	0	EN = 0: No Change
1	1	0	1	
1	1	1	0*	

Gated D Latch

S	D	Q^+
0	X	Q
1	0	0
1	1	1

Synchronous with Clock (CLK)

D – Flip-Flop

Inputs D	Inputs CLK	Outputs Q	Outputs $\overline{Q}$	Comments
0	↑	0	1	Reset
1	↑	1	0	Set

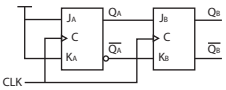
J–K – Flip Flop

J	K	CLK	Q	$\overline{Q}$	Comments
0	0	↑	Q_0	$\overline{Q_0}$	No Change
0	1	↑	0	1	R eset
1	0	↑	1	0	Set
1	1	↑	$\overline{Q_0}$	Q_0	Toggle

Asynchronous Inputs

PRE = 1 → FF = 1

CLR = 1 → FF = 0



Characteristic Equations for Flip-Flops

SR: $Q^+ = S + \overline{R}Q \longrightarrow Q_n^+ = S_n + \overline{R_n}Q_n$

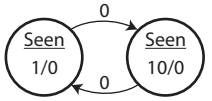
JK: $Q^+ = J\overline{Q} + \overline{K}Q \longrightarrow Q_n^+ = J_n\overline{Q_n} + \overline{K_n}Q_n$

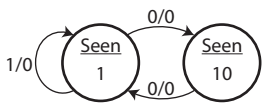
D: $Q^+ = D \longrightarrow Q_n^+ = D_n$

T: $Q^+ = T\overline{Q} + \overline{T}Q \longrightarrow Q_n^+ = T_n\overline{Q_n} + \overline{T_n}Q_n$

Application / Excitation Table for Flip-Flops

Q	Q^+	D	J	K	S	R	T
0	0	0	0	X	0	Z	0
0	1	1	1	X	1	0	1
1	0	0	X	1	0	1	1
1	1	1	X	0	X	0	0





Mealy = Output funciton of present state and input. Outputs active immediately on path

Moore = Output function of present state only → Active for full clock cycle (on state circle)

Synchronous Sequential Circuit Analysis

1) Logic Diagram: # inputs / outputs, + / -, model
2) Assign state / excitation variables: Label inputs / outputs for flip-flops
3) Excitation and Output Equations: Write logic equations for each input flip-flop and design output
4a) Transition Equations: (Obtained by applying flip-flop characteristic equation) Use characteristic equation to write flip-flop output equation.
4b) Excitation Table: Simplify equation to minimal POS / SOP
5) Transition Table: Create and fill K-Map per next state Qn+ using 4b equations (# of variables = # of flip-flops + number of inputs)
6) State Table: Use K-Maps together to determine next state

Present State Q1, Q2, . . .	Next State Q1, Q2, . . .	Output
0 0 . . .	One column/input combo	Moore: 1 column output
0 1 . . .	X Y = 0 0, 0 1, 1 0, 1 1	Mealy: 1 column / input combo

7) State Diagram: Create bubble diagram

Synchronous Sequential Circuit Design

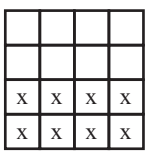
1) Determine # of inputs/outputs. "n" state variables, states, Mealy vs. Moore Model
2) Draw state table (#6 on left)
3) From state table, draw n next state K-Maps for Q_1^+, Q_2^+, Q_3^+,
4) Using each next tate K-Map from above and the application table, draw K-Maps for flip-flop inputs and repeat for all next state variables
5) Find SOP equation using flip-flop input K-Maps in terms of input variables and state variables (FF inputs). Simplify
6) Determine simplified equation in terms of input variables and state variables either by K-Map or inspecting logic in truth table.
7) Draw Logic (flip-flop) diagram using #5 and #6

K-Maps

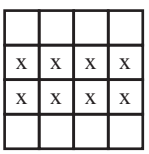
000	001	011	010	110	111	101	100
m_0	m_1	m_3	m_2	m_6	m_7	m_5	m_4
m_8	m_9	m_{11}	m_{10}	m_{14}	m_{15}	m_{13}	m_{12}
m_{24}	m_{25}	m_{27}	m_{26}	m_{30}	m_{31}	m_{29}	m_{28}
m_{16}	m_{17}	m_{19}	m_{18}	m_{22}	m_{23}	m_{21}	m_{20}

	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{16}
10	m_8	m_9	m_{11}	m_{10}

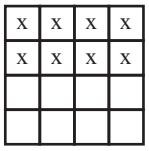
Flip-Flop K-Maps



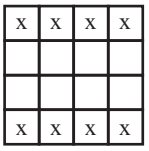
J_1^+ use Q_1^+,Q_1



J_2^+ use Q_2^+,Q_2



K_1^+ use Q_1^+,Q_1



K_2^+ use Q_2^+,Q_2

Word Problems

1) What needs to be remembered
2) Name & Number inputs / outputs
3) Determine Mealy / Moore
4) Draw state diagram for active case
5) Complete state diagram by including transition path for all inputs
6) Reread & adjust

Loading K-Maps

SOP loading K-Map
each expression is a 1

POS loading K-Map
each expression is a 0

SOP = Put 1s in

POS = Put 0s in