Digital Logic Design

Short notes for B.Tech CSE / IT / ECE · BCA · MCA · GATE CS

Uncodemy Handwritten Notes · free for students

1. Number Systems

- Common bases: binary (2), octal (8), decimal (10), hexadecimal (16).
- **Decimal to binary**: divide by 2 again and again, read remainders from bottom to top.
- **Binary to octal**: group 3 bits from the right. **Binary to hex**: group 4 bits.

EXAMPLE 45 in binary: 45/2 = 22 r1, 22/2 = 11 r0, 11/2 = 5 r1, 5/2 = 2 r1, 2/2 = 1 r0, 1/2 = 0 r1. Answer = 101101.

EXAMPLE (101101)2 = 32 + 8 + 4 + 1 = 45. Octal = 101 101 = (55)8. Hex = 0010 1101 = (2D)16.

Signed numbers (n bits)

Method	Range	Zeros
Sign-magnitude	−(2^(n−1) − 1) to +(2^(n−1) − 1)	Two (+0, −0)
1's complement	−(2^(n−1) − 1) to +(2^(n−1) − 1)	Two
2's complement	−2^(n−1) to +(2^(n−1) − 1)	One

- **1's complement**: flip all bits. **2's complement** = 1's complement + 1.
- −5 in 8-bit 2's complement: 5 = 00000101, flip = 11111010, +1 = 11111011.
- **Overflow** in 2's complement addition: when two numbers of the same sign give a result of the opposite sign.

Codes

- **BCD (8421)**: each decimal digit in 4 bits. 59 = 0101 1001.
- **Excess-3**: BCD + 3. It is self-complementing.
- **Gray code**: only one bit changes between consecutive numbers.

FORMULA Binary to Gray: g(MSB) = b(MSB), g(i) = b(i+1) XOR b(i). Example 1011 → 1110.

2. Boolean Algebra and Logic Gates

Law	Form
Identity	A + 0 = A, A · 1 = A
Null	A + 1 = 1, A · 0 = 0
Idempotent	A + A = A, A · A = A
Complement	A + A' = 1, A · A' = 0
Absorption	A + AB = A, A(A + B) = A
Distributive	A + BC = (A + B)(A + C)
De Morgan	(A + B)' = A'B', (AB)' = A' + B'
Consensus	AB + A'C + BC = AB + A'C

REMEMBER A + A'B = A + B. This one simplifies many GATE expressions.

➤ Gates

Gate	Output	Note
AND	A·B	1 only if all inputs are 1
OR	A + B	1 if any input is 1
NOT	A'	Inverts
NAND	(AB)'	Universal gate
NOR	(A + B)'	Universal gate
XOR	A'B + AB'	1 if inputs differ (odd number of 1s)
XNOR	AB + A'B'	1 if inputs are equal

- With n variables, number of distinct Boolean functions = 2^(2^n).
- NAND only: NOT A = A NAND A. AND = NOT of NAND. OR = (A NAND A) NAND (B NAND B).

3. Minimization: SOP, POS and K-maps

- **SOP** (sum of products) uses minterms Σm. **POS** (product of sums) uses maxterms ΠM.

- Minterm: product where each variable appears once (1 in truth table). Maxterm: sum (0 in truth table).
- **K-map** order uses Gray code: 00, 01, 11, 10.
- Group 1s in sizes 1, 2, 4, 8 (powers of 2). Bigger groups = simpler terms. Corners and edges wrap around.
- **Don't care (X)**: can be used as 1 or 0 to make bigger groups.

➤ Solved example

- $F(A, B, C) = \Sigma m(1, 3, 5, 7)$.

```
        BC=00  01  11  10
A=0       0    1   1   0
A=1       0    1   1   0
```

- All four 1s are in columns BC = 01 and 11, so C = 1 in all. Answer: $F = C$.

EXAMPLE $F(A,B,C,D) = \Sigma m(0, 2, 8, 10)$ = the four corners of a 4-var K-map, so $F = B'D'$.

- **Prime implicant**: a group that cannot be made bigger. **Essential prime implicant**: covers some 1 that no other prime implicant covers.
- **Quine–McCluskey** (tabular method) works for many variables.

4. Combinational Circuits

- Output depends only on present inputs. No memory.

➤ Adders

Circuit	Sum	Carry
Half adder	A XOR B	AB
Full adder	A XOR B XOR Cin	AB + Cin(A XOR B)
Half subtractor	A XOR B	Borrow = A'B

- A full adder = 2 half adders + 1 OR gate.
- **Ripple carry adder**: n full adders. Delay grows with n.
- **Carry look-ahead adder**: uses G = AB (generate) and P = A XOR B (propagate) to find carries fast.

FORMULA Carry look-ahead: $C(i+1) = G(i) + P(i)\cdot C(i)$.

➤ Other circuits

Circuit	Inputs → Outputs	Job
Multiplexer (MUX)	2^n data + n select → 1	Selects one input
Demultiplexer	1 + n select → 2^n	Sends input to one line
Decoder	n → 2^n	Activates one output line
Encoder	2^n → n	Gives binary code of active input
Priority encoder	2^n → n	Highest-priority input wins
Comparator	Two n-bit → 3	A > B, A = B, A < B

- A 2^n : 1 MUX can implement any function of n variables (or n + 1 variables with one variable at the data inputs).
- 4:1 MUX output: Y = S1'S0'I0 + S1'S0 I1 + S1 S0' I2 + S1 S0 I3.

> **TIP** To build a 16:1 MUX from 4:1 MUXes, you need 4 + 1 = 5 MUXes.

5. Sequential Circuits: Flip-Flops

- Output depends on present inputs and past state (has memory).
- **Latch**: level-triggered. **Flip-flop**: edge-triggered (clock edge).
- SR latch with NOR gates: S = R = 1 is invalid.

Flip-flop	Characteristic equation	Note
SR	Q+ = S + R'Q (SR = 0)	S = R = 1 not allowed
JK	Q+ = JQ' + K'Q	J = K = 1 toggles
D	Q+ = D	Used in registers
T	Q+ = T XOR Q	T = 1 toggles

➤ Excitation table (what input is needed)

Q → Q+	J K	D	T
0 → 0	0 X	0	0
0 → 1	1 X	1	1
1 → 0	X 1	0	1
1 → 1	X 0	1	0

- **Race-around condition**: in a level-triggered JK with J = K = 1, output toggles many times in one clock pulse. Fixed by **master–slave** JK or edge triggering.

6. Registers and Counters

- **Register** = group of flip-flops to store bits. n-bit register = n flip-flops.
- Shift registers: SISO, SIPO, PISO, PIPO.

➤ Counters

Type	Clock	Speed
Asynchronous (ripple)	Output of one FF clocks the next	Slow, delay adds up
Synchronous	Same clock to all FFs	Fast

- A mod-N counter has N states. Flip-flops needed = $\lceil \log_2 N \rceil$.
- **Ring counter** with n FFs: n states. **Johnson (twisted ring) counter** with n FFs: 2n states.

FORMULA n-bit ripple counter max frequency = 1 / (n x t_{pd}), where t_{pd} = delay of one flip-flop.

EXAMPLE A mod-10 (decade) counter needs $\lceil \log_2 10 \rceil = 4$ flip-flops. It resets when the count reaches 1010.

EXAMPLE A 4-bit ripple counter with 10 ns flip-flop delay: max frequency = 1 / (4 x 10 ns) = 25 MHz.

7. Finite State Machines

Mealy machine	Moore machine
Output depends on state and input	Output depends on state only
Usually fewer states	Usually more states
Output changes with input	Output changes only at clock

- Design steps: state diagram → state table → state assignment → flip-flop input equations → circuit.

8. Memory and Logic Families (short)

- **ROM**: non-volatile. Types PROM, EPROM (UV erase), EEPROM (electrical erase), Flash.
- **RAM**: volatile. **SRAM** (flip-flops, fast, cache) vs **DRAM** (capacitor, needs refresh, main memory).
- A memory with 2^k words of n bits has k address lines and n data lines.
- **PLA**: programmable AND and OR arrays. **PAL**: programmable AND, fixed OR.

Family	Note
TTL	Bipolar transistors, fast, more power
CMOS	Very low static power, used in most modern chips
ECL	Fastest, high power

FORMULA Fan-out = how many gate inputs one output can drive. Noise margin = $V_{OH(min)} - V_{IH(min)}$ for high level, $V_{IL(max)} - V_{OL(max)}$ for low level.

9. Most Asked Questions (practice)

1. Convert numbers between binary, octal, decimal and hex. Find 2's complement.
2. Simplify a Boolean expression using laws and De Morgan's theorem.
3. Minimize a 4-variable function with don't cares using a K-map.
4. Design a full adder using two half adders. Explain carry look-ahead.
5. Implement a function using an 8:1 MUX.
6. Explain SR, JK, D and T flip-flops with characteristic and excitation tables.
7. What is race-around condition? How does master-slave JK solve it?
8. Design a mod-6 synchronous counter using JK flip-flops.
9. Compare ring and Johnson counters.
10. Mealy vs Moore machine with an example.

TIP GATE asks K-map minimisation, MUX-based functions, 2's complement overflow, counter states and flip-flop conversions. Always verify a K-map answer with one minterm.