01 · Digital Logic & Number Systems

TutorialHub cheat sheet

Numbers

Bits	Unsigned range	Signed (two's complement)
4	0 … 15	−8 … +7
8	0 … 255	−128 … +127
16	0 … 65,535	−32,768 … +32,767
n	0 … 2^n-1	-2^{n-1} … $2^{n-1}-1$

- 1 hex digit = 4 bits. `8'hB6` = `1011_0110` = 182.
- Negate: invert all bits, add 1. `-10` = `1111_0110`.
- Sign-extend: copy the MSB. `1010` (−6) → `1111_1010`.
- Carry = unsigned result doesn't fit. Overflow = signed result doesn't fit.
- ADD overflow: `a[n-1]==b[n-1] && sum[n-1]!=a[n-1]`
- SUB overflow: `a[n-1]!=b[n-1] && diff[n-1]!=a[n-1]`
- Subtract with an adder: `a - b = a + ~b + 1`.

Boolean algebra

Rule	Form
Complement	a·ā = 0 · a + ā = 1
Absorption	a + a·b = a · a·(a + b) = a
De Morgan	$\overline{(a\cdot b)} = \bar{a} + \bar{b}$ · $\overline{(a + b)} = \bar{a}\cdot\bar{b}$
Consensus	a·b + ā·c + b·c = a·b + ā·c
XOR	a ⊕ b = a·b̄ + ā·b

NAND and NOR are universal. K-map groups: 1, 2, 4, 8 cells; wrap edges; use don't-cares.

Building blocks

Block	Behaviour
2:1 mux	`y = sel ? b : a`
2-to-4 decoder	one-hot output of the code
Priority encoder	index of the highest set bit
Half adder	s = a⊕b · c = a·b
Full adder	s = a⊕b⊕cin · cout = a·b + cin·(a⊕b)
Equality	AND of per-bit XNORs

Sequential

Term	Meaning
Latch	level-sensitive (avoid in RTL)
D flip-flop	samples D on the clock edge
t_su / t_h	stable before / after the edge
t_cq	clock edge → Q
Min period	t_cq + logic + t_su
Metastability	setup/hold violated → 2-FF synchronizer across clock domains
Moore	outputs from state only
Mealy	outputs from state + inputs (one cycle earlier)
Encodings	binary · one-hot · Gray

Verilog literals

`4'b1010` · `8'hFF` · `8'd45` · `-8'sd3` (signed) · `'0` / `'1` (fill all bits) · `_` separators allowed.

Checking recipe

1. Reference model in plain integer math, one bit wider than the DUT.
2. Exhaustive stimulus when inputs ≤ ~20 bits; otherwise constrained-random.
3. Compare every output with `!==` (catches X and Z).
4. Print the first few mismatches, then `PASS n/n` or `FAIL k of n`.

Quick methods

- Decimal → binary: divide by 2, read remainders bottom-up. 45 → `101101`.
- Binary → decimal: add the weights of the 1s. `1011` = 8 + 2 + 1 = 11.
- Clock: period = 1 / frequency. 1 GHz → 1 ns · 500 MHz → 2 ns · 100 MHz → 10 ns.
- Counter wraps: 4-bit 15 + 1 → 0. Trace FSMs input by input in a table.
- Sequential checks: reset values · every transition · corners (wrap, same-edge events, mid-sequence reset).

Remember it like this

Idea	Picture
Digital vs analog	light switch vs dimmer
Transistor	switch whose lever is the gate
Binary counting	odometer whose wheels have only 0 and 1
Two's complement	clock face: back 1 from 0 is the top value
AND / OR	switches in series / in parallel
XOR	staircase light: either switch flips it
Mux / decoder	railway points / hotel key card
Flip-flop	camera: edge = shutter click, Q = photo
Setup / hold	keep still while the shutter clicks
Metastability	ball balanced on a hill
FSM	traffic light