SS
Thursday, June 2, 2022 7:47 AM

A 30
lab 50
quiz 20
fnl 30
WE 70

transmitting lines vs t

serial // data transfer

duty cycle $(\frac{t_w}{T})_{100\%}$ 50% t_w t_r t_f 10%–90%

logic signals

overflow impossible signed carry ⊕ 1's comp 2's comp carry BCD GRAY

XOR (odd parity detector) Inverter Buffer

logic gates

Kmaps

boolean arithmatic

$XY + YZ \cdot 1 \cdot X'Z = XY + X'Z$

$A + A'B = A + B$

$A \cdot AB = A$

$A(A+B) = A$ tricks $(A+A')$ $(1+A)$ $(1 \cdot A')$

duality vs demorgans

Numb sys

logic families

unipolar: CMOS (PMOS, NMOS); PMOS; NMOS

bipolar: saturated (DTL, TTL, DL, RTL); non saturated (Schotty TTL, ECL)

hybrid (BICMOS)

fast switching

terminology

- fanin/out
- propagation delay: diff 50% transitions pts
- power dissapation = $V_{drain} \times I_{avg}$

timing hazards (when ckt are not ideal) glitches

static: 1 (output supposed to be 1 but momentary 0); 0

dynamic: output switches from 0 to 1 multiple times → when var's connected through

Shorthand notation

$\sum m(1,2,3) = \prod M(4,5,6)$

canonical SoP: minterms — n literals (contain every var); normal product term with ABC (1 that contain one var more than once); on-set

canonical PoS: maxterms ⇒ off sets

Corg & dig des

logic elements (logic ckts)

combinational (O depends only on current I)

implementation: lvl 1 → com; lvl 2 → AN; lvl 3 → O

universal gates: NOR, NAND (can implement AND, OR + NOT)

arch components: decoder →

sequential logic (O depends on current I + past O)

stored in state vars ⇒ bin (need to record time + val) finite # (2^n)

finite state machines

keep track using free running clk

t_H active high; t_L low — state changes

types

built using: feedback seq. ckt; latches & flip flops

clocked synchronous state machines

controlled by clock

flip flop

edge triggered (rising edge): changes at specific clk signals

latch (uncontrolled): cts watch on I/O independent of clk signals

types

S-R: set reset

S-R with EN

D

D ⇒ keep the val (1-bit store)

S	R	Q	$\bar{Q}$N
0	0	last Q	last $\bar{Q}$N
0	1	0	1
1	0	1	0

complement

both S & R

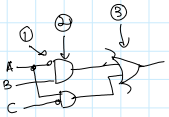
when var & complement connected through same OR gate } sol^n: K-maps adjacent terms circle

} 3 levels of OR & AND
} 3 or more paths } sol^n: use 2 lvl design

fast, slow, fast

synchronous ccts → clock can be tuned to eliminate timing hazards

asynchronous ccts → 2 lvl design K-maps



D for SOP

OR } n-input

implementation
- NAND based — double negation: $\overline{A \cdot B}$ = NAND, $\overline{A+B}$ = NOR
- OR based → turn every gate ⇒ NAND, reverse effects with NOT; double not cancels out

n to m, $m \leq 2^n$

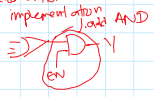
2 to 4

3 to 8

enable pin (EN)

Output when only EN=1 can add after implementation, add AND

Y_0 becomes 1 sequentially ↓ Y_3



edge triggered flipflops } computer mem used this type

(-)ve

ve edge trig gered

D	CLK	Q	QN
0	↑	0	1
1	↑	1	0
x	0	last Q	last QN
x	1	last Q	last QN

J-K flipflop

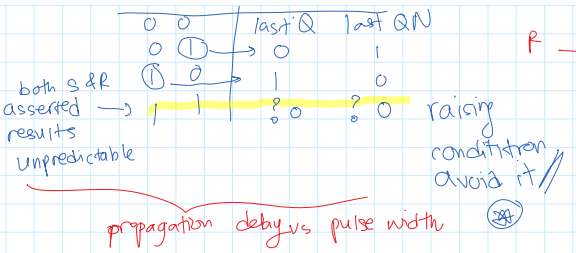
J	K	CLK	Q	QN
1	1	↑	last QN	Q

~ previous Q

solv S-R flipflop prob

T-flipflop

negates every clk cycle


last Q
last QN
both S&R asserted
results unpredictable
raising condition
avoid it
propagation delay vs pulse width


R
Q
N

↓	1	1	0
x	0	last Q	$\overline{Q}$N
x	1	last Q	$\overline{Q}$N