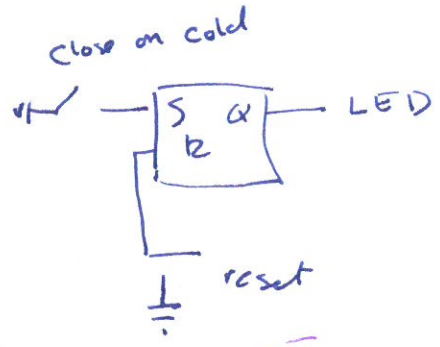
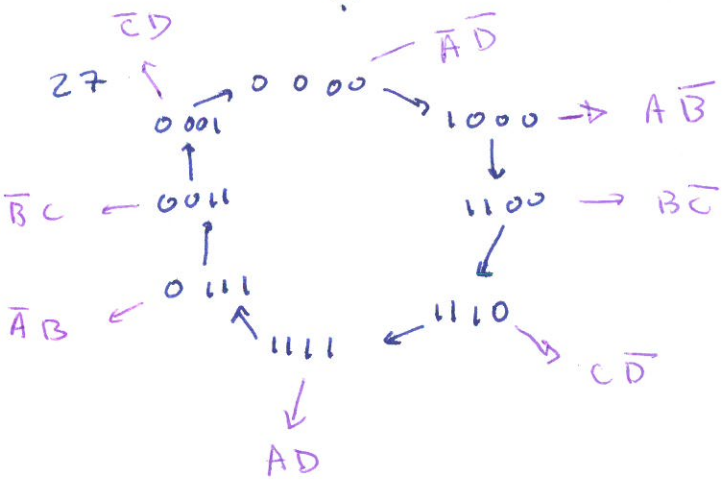
Problems.pdf : 21a, 27, 34, 31, 32, 35

computers.pdf: 1, 2

21a - use active low SK FF





34 a) Math: $r_1 = \frac{1}{7} r_3$

$r_2 = \frac{1}{2} r_3$

i say $r_3 = 100\,k$

$r_1 = 14.3\,k \Rightarrow 15$

$r_2 = 50\,k \quad 51$

b) Take $V_T = 1.25\,V \rightarrow \frac{1.25}{5} = \frac{r_1 \| r_2}{r_2}$

$\frac{1.25}{5} r_2 = r_1 \| r_2 = 12.5\,k$

$50k \rightarrow 51$

$\frac{\Delta V}{V_T} = \frac{.5}{5} = \frac{12.5k}{12.5k + R_3} \rightarrow .1(12.5 + R_3) = 12.5k$

$R_3 = 125 - 12.5$

$= 112.5\,k \rightarrow 110$

$\frac{1}{r_1} + \frac{1}{r_2} = \frac{1}{17.5}$

$r_1 = \frac{1}{\frac{1}{12.5} - \frac{1}{r_2}} = 16.7\,k \rightarrow 16$

$R_2 \| R_3 = \frac{51 \cdot 110}{161} = 34.85$

$R_1 \| R_3 = \frac{16 \cdot 110}{126} = 13.97$

$V_1 = V \frac{16}{16 + 34.85} = 1.53\,V$

$V_0 = V \frac{13.97}{51 + 13.97} = 1.08\,V$

$V_1 = V \frac{R_1}{R_1 + \frac{R_2 R_3}{R_2 + R_3}} = 5 \frac{15}{15 + 33.8} = 1.53\,V$

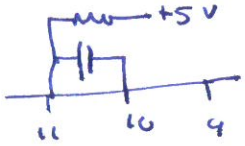
$V_0 = V \frac{\frac{R_2 \| R_3}{R_2 + R_2 \| R_3}}{} = 5 \cdot \frac{13}{51 + 13} = 1.02\,V$

$R_2 \| R_3 = \frac{100 \cdot 51}{151} = 33.8$

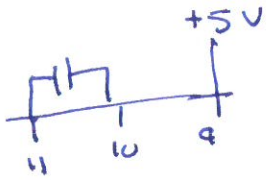
$R_1 \| R_3 = \frac{100 \cdot 15}{115} = 13.0$

31 — The spec sheet is a bit obscure on exactly how to connect the cap/resistor. HH Fig 7.59 (p 462) is accurate (even though it does not exactly apply to 121)
Figure 2 in SN74121B.pdf or in SN74121.pdf
In the notes: "1. an external capacitor may be connected between C_{ext} (positive) and R_{ext}/C_{ext}. 2. To use the internal resistor connect Rint to V_{CC}"

So: with external R:



with internal 2K:



The formula for pulse length is also not easy to find:

$\tau = .7\ RC$

I'm using internal 2k so $20\mu s = .7\ 2k\ C$

$.014\mu F = C$

↳ Note: not in HH p 1104 "5%" list so hand to purchaser — use $.015\mu F$

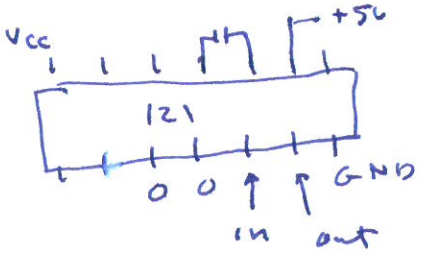
For 10 µs pulse needed for 50 kHz square wave need $\frac{1}{2}$ above ... 6.8 nF or 6800 pF

For 1ms delay $1 \times 10^{-3} = .7\ 2k\ C$

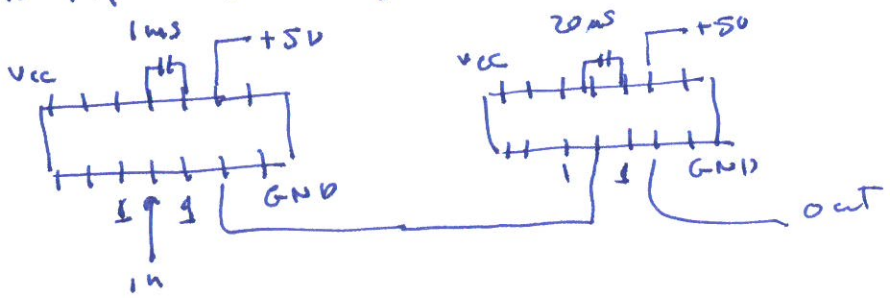
$.71\mu F = C$

↳ use .68 nF

9) B is good for positive edge
A1 & A2 = Low / pwr/GND at corners

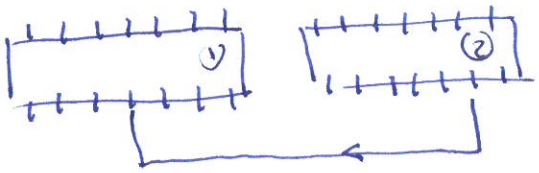


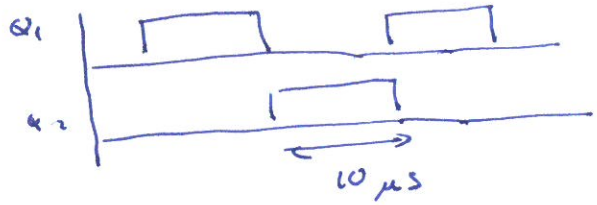
b) A inputs for negative edge: A1 = High A2 = Input B = High



c) For 50 kHz, Period = 20 μs so H & L are 10 μs each

As above





32) a) B b) C c) C d) A

36 See lecture 11_timers.pdf for words

mono shot
C: between (7) = discharge & gnd
R: between (7) & Vcc
) trigger goes low → pulse 1.1 RC

oscillate:
C: between (2) ← trigger & gnd
R_A: between (7) = discharge & Vcc
R_B: between (7) & (2)
connect (6) = threshold to (2) = trigger

```
format for :
64 bit ("double") float:
(sign)(11 bit exponent)(52 bit mantissa)
1.0= 0x3FF0000000000000 = 00 ten 1 rest 0
2.0= 0x4000000000000000 = 0 one 1 rest 0
0.5= 0x3FE0000000000000 = 00 nine 1 rest 0

find sqrt of a number stored in RAM
find 3 mistakes!

load r0 r1      1101x01 load opcode  #r1 holds S
address that holds number
mov r1,r2       0011x12 mov opcode
sub r2,r0,r2    0111202 sub opcode
2^52
clr r3          0001xx3 clr opcode
inc r3 r3       0011x33 inc opcode
asr r2 r3 r2    0111232 asr opcode
add r2 r0 r2    1111202 add opcode  #r2 holds the first approx: x
2^61
mov r0 r5       1011x05 mov opcode
2.0
mov r0 r6       1011x06 mov opcode
small
mov r0 r9       1011x09 mov opcode
2
mov rF rE       0011xFE mov opcode  #current PC=rF saved to rE
add r9 rE rE    01119EE add opcode  #note (rE) is address of following instruction
mov r2 r4       0011x24 mov opcode  #r4 is the starting x
fdiv r1 r2 r3   0111123 fdiv opcode
fadd r2 r3 r2   0111232 fadd opcode
fdiv r2 r5 r2   0111252 fdiv opcode #r2 is the updated x
fsub r2 r4 r4   0111232 fsub opcode
fdiv r4 r2 r4   0111123 fdiv opcode
fabs r4 r4      0011044 fabs opcode
fcmp r4 r6      011046x 2 fcmp opcode = fsub opcode
bpl rE          0011xEF conditional version of mov rE rF opcode
mov r0 rE       1011x0E mov opcode
address to output x
store r2 rE     0110CEx store opcode
halt



example: (all iterations shown)
take sqrt of 2.
bit pattern of 2.=4000000000000000 (hex)
bit pattern of x0=3FF8000000000000 (hex)
   1.5000000000000000
   1.4166666666666665
   1.4142156862745097
   1.4142135623746899
   1.4142135623730949

bit pattern of xN=3FF6A09E667F3BCC (hex)

   1.4142135623730951 (actual)

take sqrt of pi
3.14159265358979323d0
```