#1 Given: With VMLAB, write a code file that starts by putting ten numbers, the values +11.0 to +20.0, into the EEPROM using directives. Each number should be in IEEE 754 format with the bytes arranged big-endian. Then a loop should replace each value with the value plus +1.0. For example, the bytes for +11.0 will change to the bytes for +12.0.

Find: AVR code for the design described above

Solution: (this is the farthest that I got. I ran out of time with project / labs in ENGR 223 that need finishing and our term project here) I was able to get the proper numbers loaded into the EEPROM (i did 10-19 instead of 11-20) and have it point to the first position but it just keeps copying the number in the first position to the next. Everyone else in class is using this as their drop and I had to use HW#2 as my drop because of personal things that came up, so I am just turning in what I was able to finish.

```
.include "C:\VMLAB\include\m168def.inc"
.eseg
.dw 0x1110
.dw 0x1312
.dw 0x1514
.dw 0x1716
.dw 0x1918
.cseg
ldi r16, 0x000
out eearh, r16
ldi r16, 0x001
out eearl, r16
sbi eecr, eere
wait1: sbic eecr, eere
rjmp wait1
subi r16, 1
out eearl, r16
sbi eecr, eempe
sbi eecr, eepe
wait2: sbic eecr, eepe
rjmp wait2
subi r16, -1
out eearl, r16
rjmp wait1
```

#2 Given: For the ATmega168, write AVR code that just sets the hardware SPI to mode 2. A description of ATmega168 SPI mode is found in the SPI section of the full ATmega168 datasheet. (Note: the posted solution only has 4 lines of code.)

Find: AVR code to set SPI to mode 2

Solution:

```
.include "C:\VMLAB\include\m168def.inc"
LDI R16, 10110000
OUT DDRB, R16
LDI R16, 00001000
OUT SPCR, R16
```

<u>#3 Given:</u> Two flip-flops are configured to form a simple synchronizer. Each flip-flop has the following characteristics:

τ = 200 ps; T0 = 150 ps; ts (FF setup time) = 500 ps; asynchronous frequency = 0.2 Hz; Tc is the clock period

<u>Find:</u> For tr = Tc - ts, which synchronizer clock frequency results in an MTBF of about 1 year?

<u>Solution:</u>

$$MTBF = e^{(t_r / \tau)} * (\frac{1}{f_a * f_c * T_0})$$

1 year = 3.15576*10^7 seconds
f_a = 0.2 Hz
T0 = 150*10^-12
τ = 200*10^-12
ts = 500*10^-12
f_c = 1/T_c
tr = (1/T_c) - ts

Plugging in our given values and solving for f_c via TI-89 solver function, we have

T_c = 3.036*10^-9

Then we can convert to frequency

f_c = 1/T_c = 0.329 Ghz

#4 Given: Two flip-flops are configured to form a simple synchronizer with τ = 33 ps and T0 = 20 ps.

Find: By how much do you need to increase the clock period (Tc) to increase the MTBF by an order of magnitude (a factor of 10)?

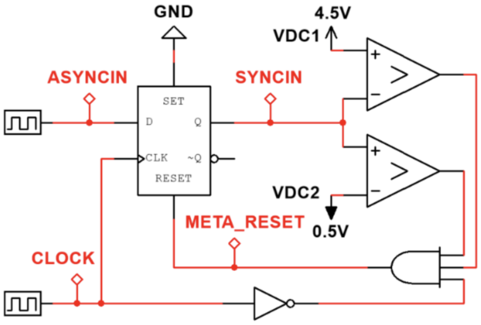
Solution:

τ = 33*10^-12
T0 = 20*10^-12

$$MTBF = e^{(t_r / \tau)} * (\frac{1}{f_a * f_c * T_0})$$

Considering T_c = 1 / f_c , to increase the MTBF by a factor of 10, we need to increase T_c period by 10 times

#5 Given: The circuit shown is designed to solve problems with metastability. If a metastable signal is detected with the components connected to the RESET signal, the Q output clears. The only problem is: it doesn't work.



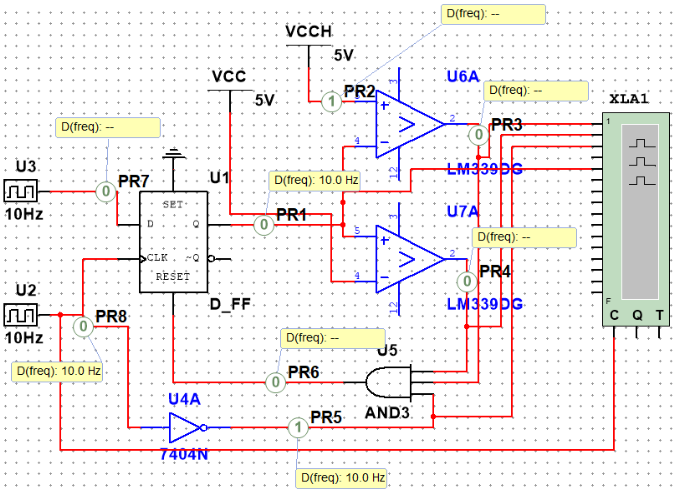
Find:

a) Does the detector circuit (which is the comparators, NAND gate and inverter) used to reset the FF have any logical errors? If not, explain your reasoning. If yes, identify the errors and suggest a correction.

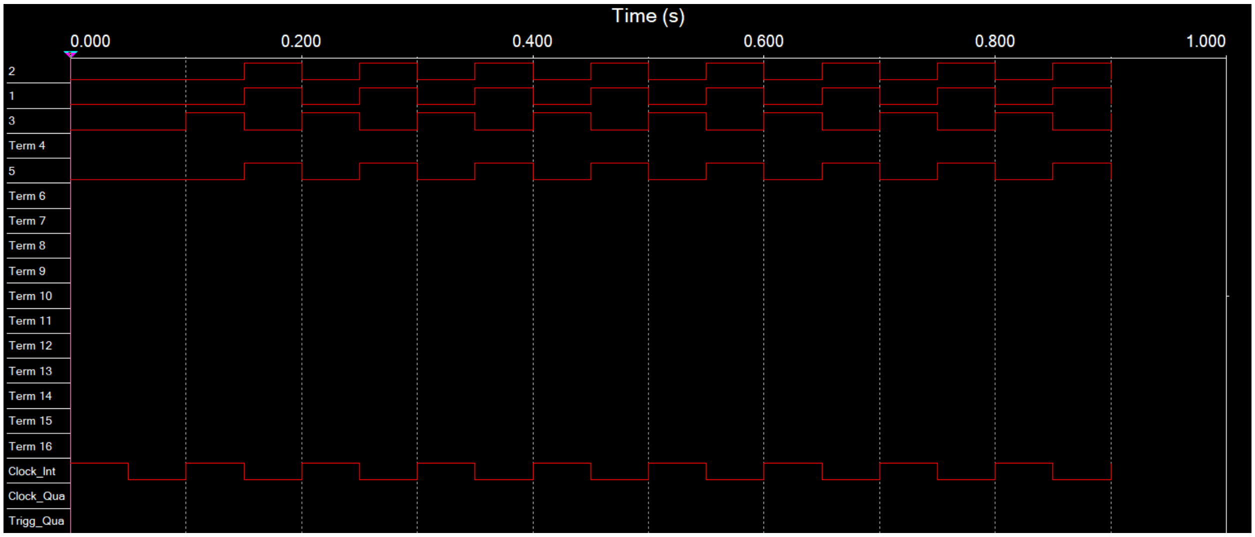
b) Even with a perfect detector circuit, describe a condition where this circuit could still fail. Include a timing diagram with your solution.

Solution:

a) There are no logical errors. I built this circuit in Multisim and ran through at 10Hz with digital probes on all wires, and saw no logical errors. After further analysis, it does not work if you have the VCC / VCCH at 0.5 and 4.5 respectively. The comparator output does not change at all. The only values that change are the CLKs, as expected. The rest of the digital probes do not change at all. Once we change both of the source voltages to 5, it seems to work properly.



b)



1+2 are the outputs of the comparator, 3 is the output of the inverter and 5 is the input of the comparators. I changed the input voltage on the comparators to 5V for both and this seems to produce a working metastability fixer, but the timing diagram proposes some issues. Unless the timing delay on the inverter is slow enough to match the comparators so you have 3 Highs going into the AND3 gate, you'll never yield a High output on the AND3 gate, as we can see from this timing diagram.