3/1/2020

Frequency Reader V1.0

Design from Scratch

Ahmed Gamal Ahmed Hussein Haroon

To start with this design we will know first the simplest design for it and we'll continue adding features to it until it reaches what we have now.

First we start with reading a frequency from 1 to 9 Hz Range, so how we can do it?

First we'll need counter that counts from 0 to 9 and we can do that using 74192 IC.

SGS-THOMSON
MICROELECTRONICS

M54/M74HC192
M54/M74HC193

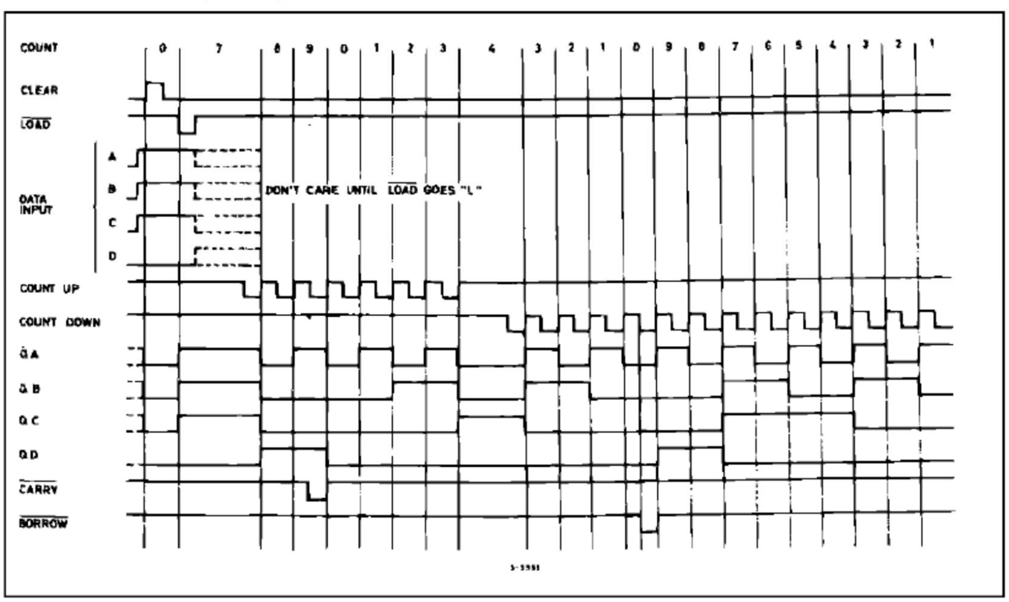
HC192 - SYNCHRONOUS UP/DOWN DECADE COUNTER
HC193 - SYNCHRONOUS UP/DOWN BINARY COUNTER

TRUTH TABLE

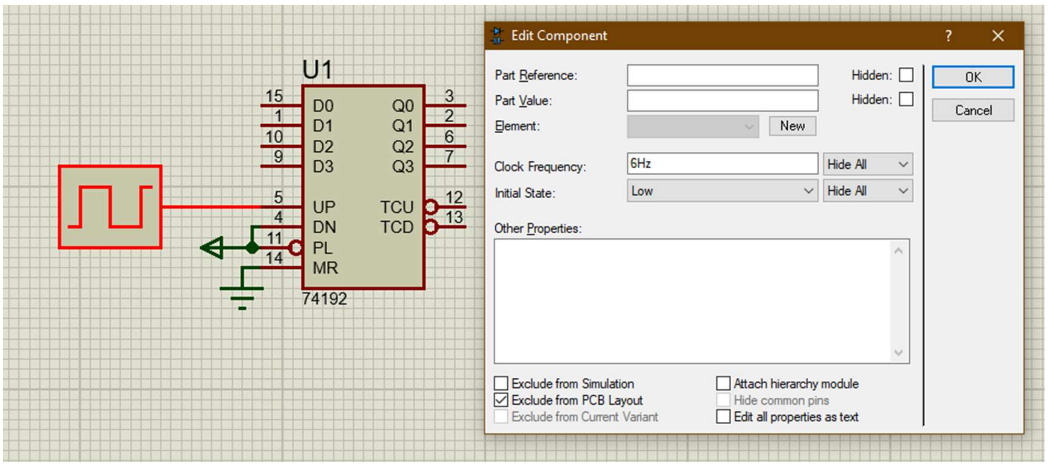
COUNT UP	COUNT DOWN	$\overline{LOAD}$	CLEAR	FUNCTION
↑	H	H	L	COUNT UP
↓	H	H	L	NO COUNT
H	↑	H	L	COUNT DOWN
H	↓	H	L	NO COUNT
X	X	L	L	PRESET
X	X	X	H	RESET

X: Don't Care

TIMING DIAGRAM (HC192)



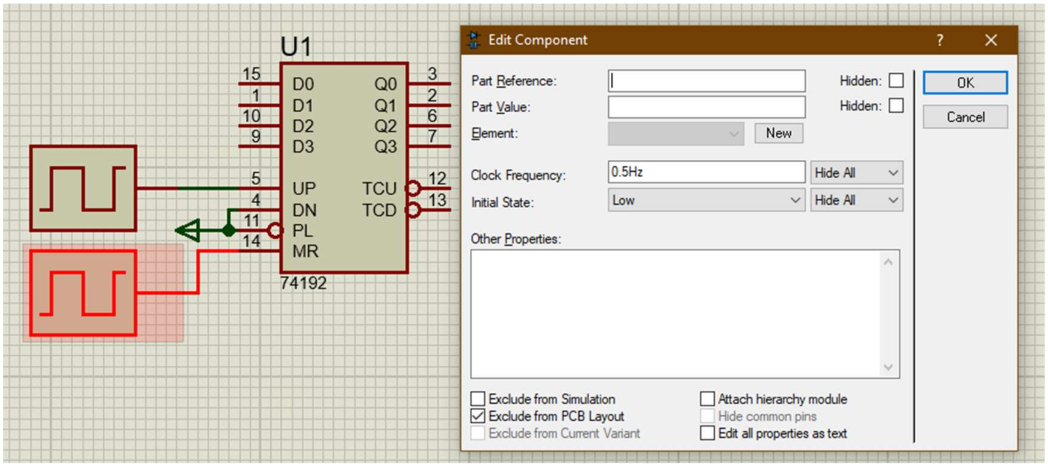
Reading truth table and time diagram we see how this IC works, it counts in binary system from 0 to 9 with every rising edge on the PIN 'UP', until it reaches the number 9 then the TCU signal turns into 0 in the next falling edge then it turns back into one with the next rising edge making the counter count from 0 again, so we can say that after every full counting from 0 to 9 the TCU signal gives a pulse, and to use it in our application we will connect the circuit as below.



With this connection when we run it, it will count with every pulse but in this case wouldn't be any restrictions, so as it reaches the number 6 it will continue non stopping, and by that we can't get a reading.

From the definition of the frequency, that is the number of cycles per second, we want the counter to stop counting after one second. We could do that using the PIN "MR", when it's 1 the counter resets and when it's 0 it proceeds counting.

So we want it to be 0 for a second then becomes 1 to stop counting, we're going to do that by putting a 0.5 Hz frequency on the "MR" PIN, that's mean it will be 1 for a second and it'll be 0 for the other second. So the circuit becomes like this.



When we run this circuit we would run into another problem and it's that after one second of counting the counter reset not allowing us to latch the output number, so we need to use registers using 74194 IC.

SGS-THOMSON MICROELECTRONICS — **M54HC194 M74HC194**

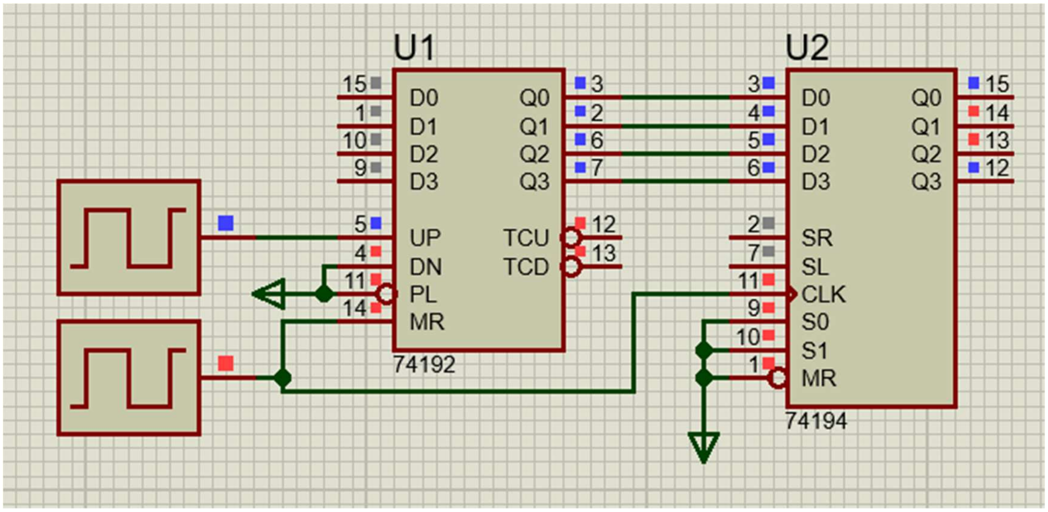
4 BIT PIPO SHIFT REGISTER

TRUTH TABLE

INPUTS											OUTPUS			
CLEAR	MODE		CLOCK	SERIAL		PARALLEL					QA	QB	QC	QD
	S1	S0		LEFT	RIGHT	A	B	C	D					
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	X	X	↓	X	X	X	X	X	X	QA0	QB0	QC0	QD0
H	H	H	↑	X	X	a	b	c	d	a	b	c	d
H	L	H	↑	X	H	X	X	X	X	H	QAn	QBn	QCn
H	L	H	↑	X	L	X	X	X	X	L	QAn	QBn	QCn
H	H	L	↑	H	X	X	X	X	X	QBn	QCn	QDn	H
H	H	L	↑	L	X	X	X	X	X	QBn	QCn	QDn	L
H	L	L	X	X	X	X	X	X	X	QA0	QB0	QC0	QD0

X: Don't Care : Don't Care
a ~ d : The level of steady state input voltage at input A ~ D respactively
QA0 ~ QD0 : No charge
QAn ~ QDn : The level of QA, QB, QC, respectively, before the mst recent positive transition of the clock.

We'll use the case in the black frame from the truth table in our application, so the circuit becomes like that.

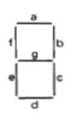


And by this way we could latching our value 6 even after the second passing.

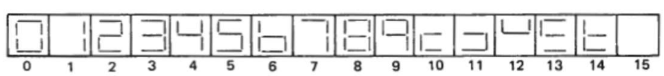
To turn the binary into 7-Segement display we will use 7448 IC.

SN5446A, ’47A, ’48, SN54LS47, ’LS48, ’LS49
SN7446A, ’47A, ’48, SN74LS47, ’LS48, ’LS49
BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS

SDLS111 – MARCH 1974 – REVISED MARCH 1988



SEGMENT IDENTIFICATION



NUMERICAL DESIGNATIONS AND RESULTANT DISPLAYS

’46A, ’47A, ’LS47 FUNCTION TABLE (T1)

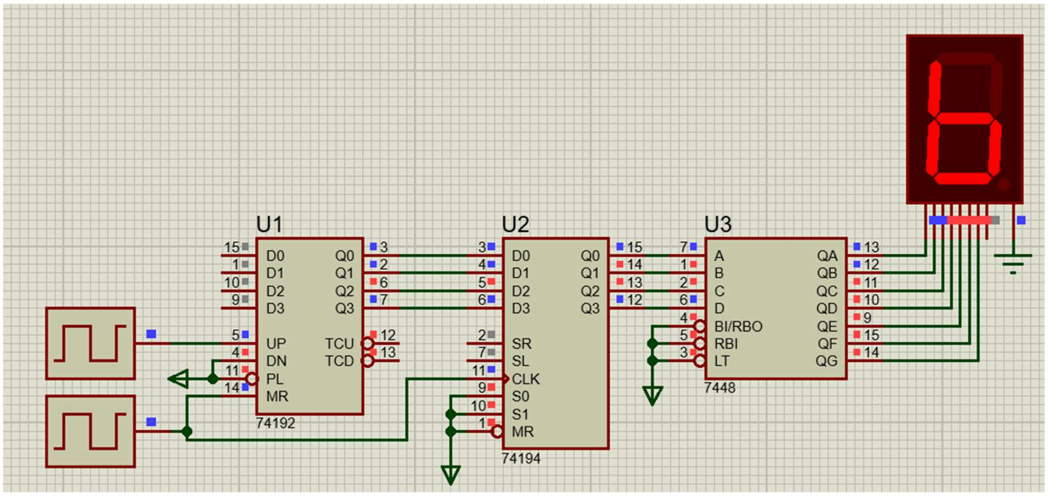
DECIMAL OR FUNCTION	INPUTS						$\overline{BI}/\overline{RBO}$†	OUTPUTS							NOTE
	$\overline{LT}$	$\overline{RBI}$	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	1
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON	
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON	
10	H	X	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON	
11	H	X	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON	
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON	
13	H	X	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON	
14	H	X	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON	
15	H	X	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4

H = high level, L = low level, X = irrelevant

NOTES:
1. The blanking input ($\overline{BI}$) must be open or held at a high logic level when output functions 0 through 15 are desired. The ripple-blanking input ($\overline{RBI}$) must be open or high if blanking of a decimal zero is not desired.
2. When a low logic level is applied directly to the blanking input ($\overline{BI}$), all segment outputs are off regardless of the level of any other input.
3. When ripple-blanking input ($\overline{RBI}$) and inputs A, B, C, and D are at a low level with the lamp test input high, all segment outputs go off and the ripple-blanking output ($\overline{RBO}$) goes to a low level (response condition).
4. When the blanking input/ripple blanking output ($\overline{BI}/\overline{RBO}$) is open or held high and a low is applied to the lamp-test input, all segment outputs are on.

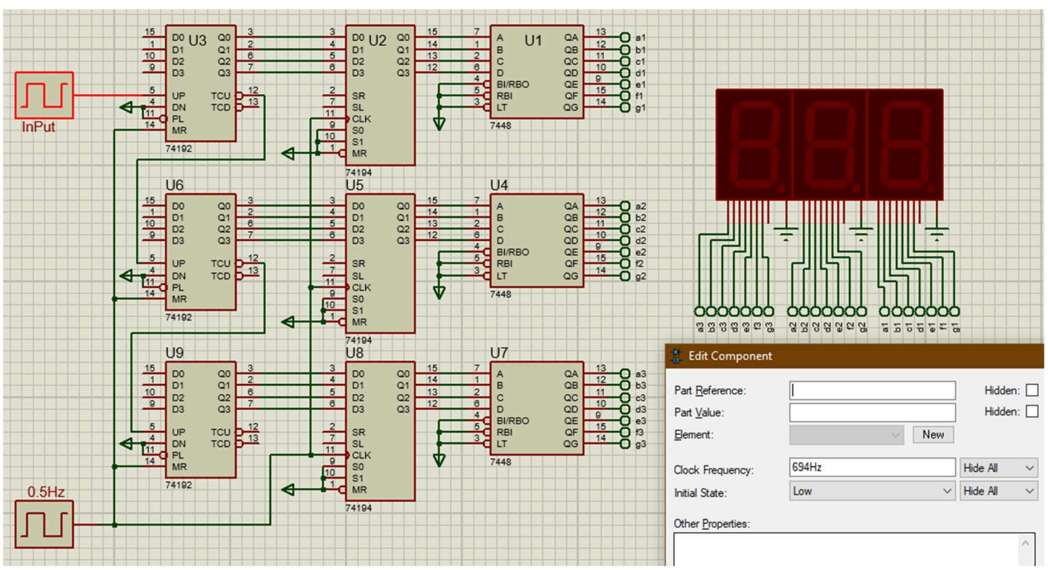
[1] $\overline{BI}/\overline{RBO}$ is wire AND logic serving as blanking input ($\overline{BI}$) and/or ripple-blanking output ($\overline{RBO}$).

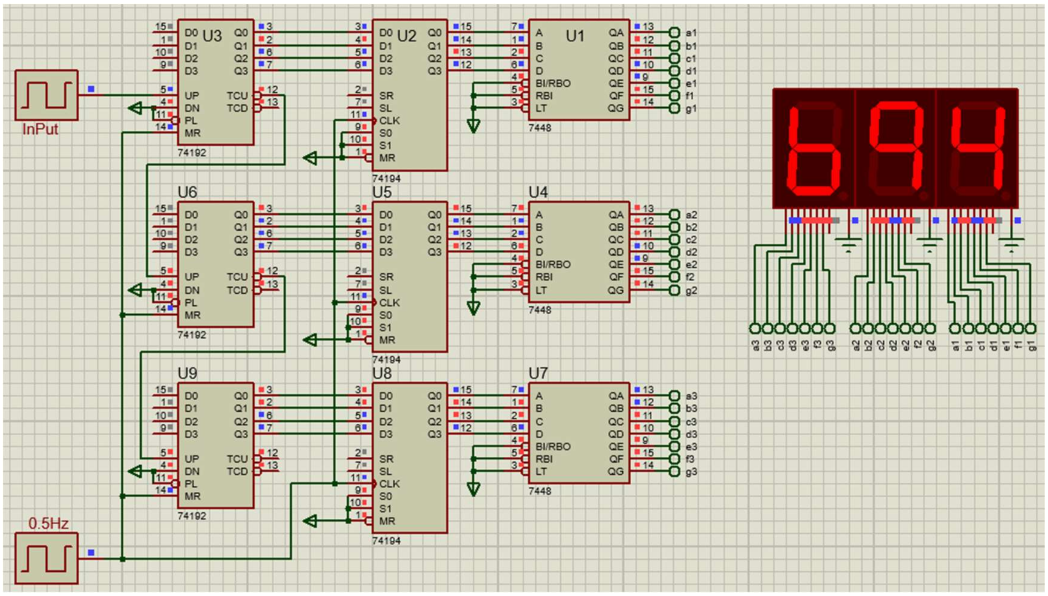
And by that the circuit becomes like that.



And by that we made a frequency reader with range from 0 to 9 Hz.

If we want to increase the range to become 0 to 999 Hz, we would use the PIN “TCU” by cascading 3 counters connecting the first TCU with the second PIN “UP”, so the circuit becomes like that.

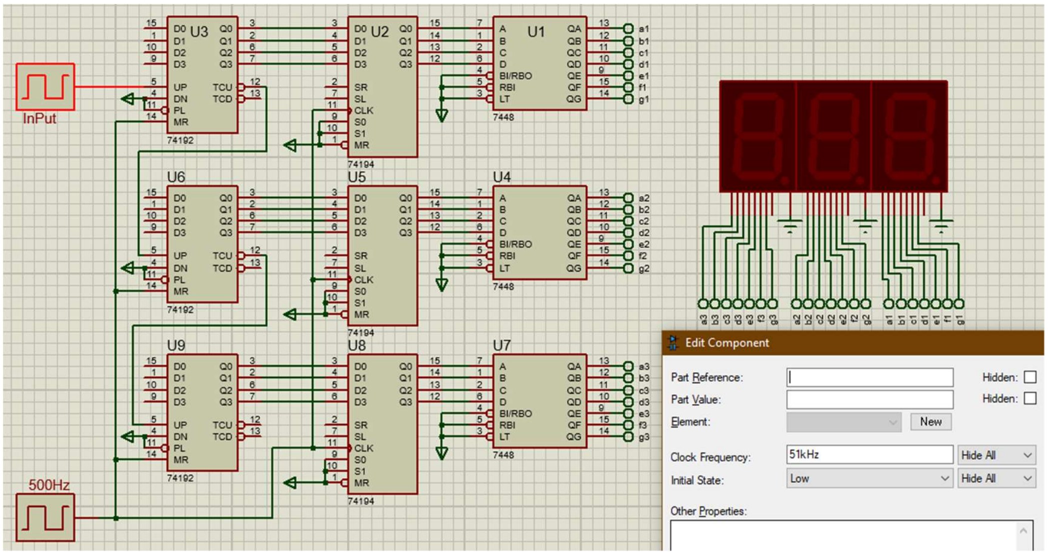


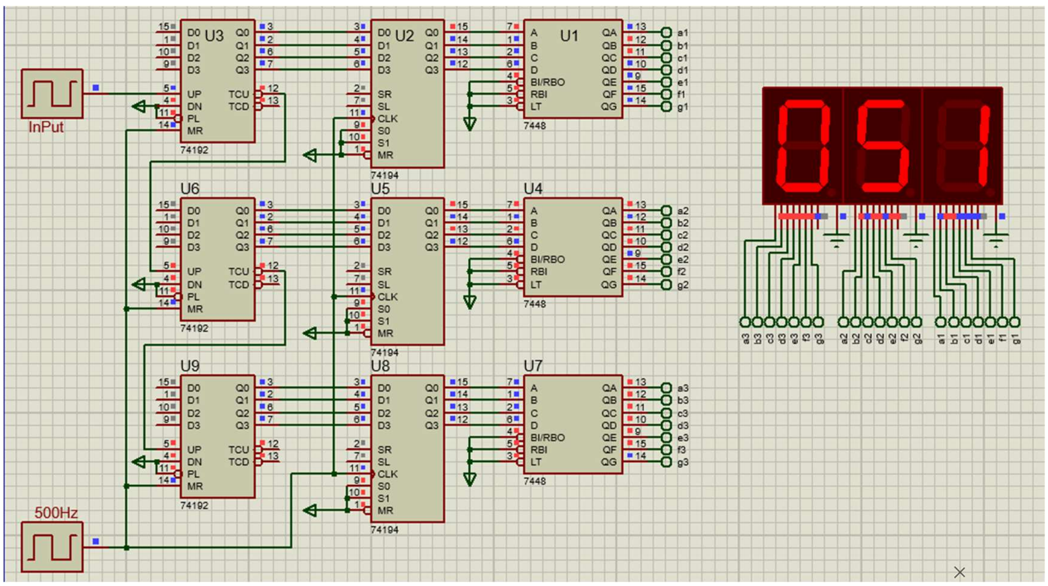


So what do we do if we wanted to increase our range even further than that, meaning we will entering the kilo range, so we want to display in the same 3 digits knowing that it's kilo range, actually the answer to that problem is easy and it comes from the same definition of the frequency, we mentioned it earlier, but now we are going to put it in the form of law so it becomes

$f = \frac{number\ of\ cycles\ in\ kHz}{1\ s}$, so if we wanted to read in kHz all we have to do is dividing the numerator and denominator by 1000 so the law becomes$f = \frac{number\ of\ cycles\ in\ Hz}{1\ ms}$.

By that all what we have to do is multiplying the MR Clock by 1000 turning it into 500 Hz instead of 0.5 Hz.





And by that we read in the kHz range.

M54/M74HC192/193

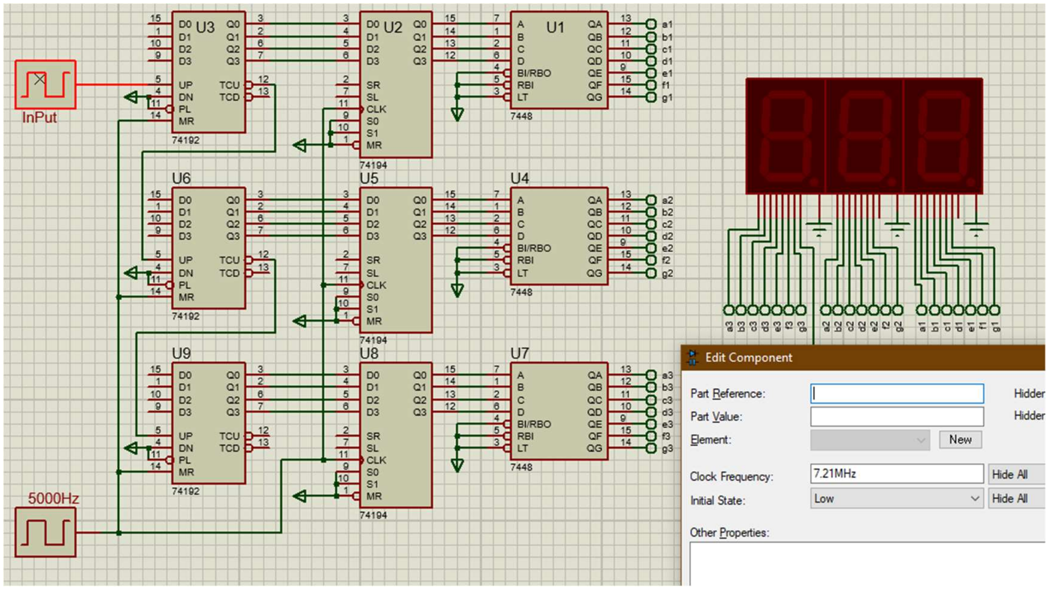
AC ELECTRICAL CHARACTERISTICS (C_L = 50 pF, Input t_r = t_f = 6 ns)

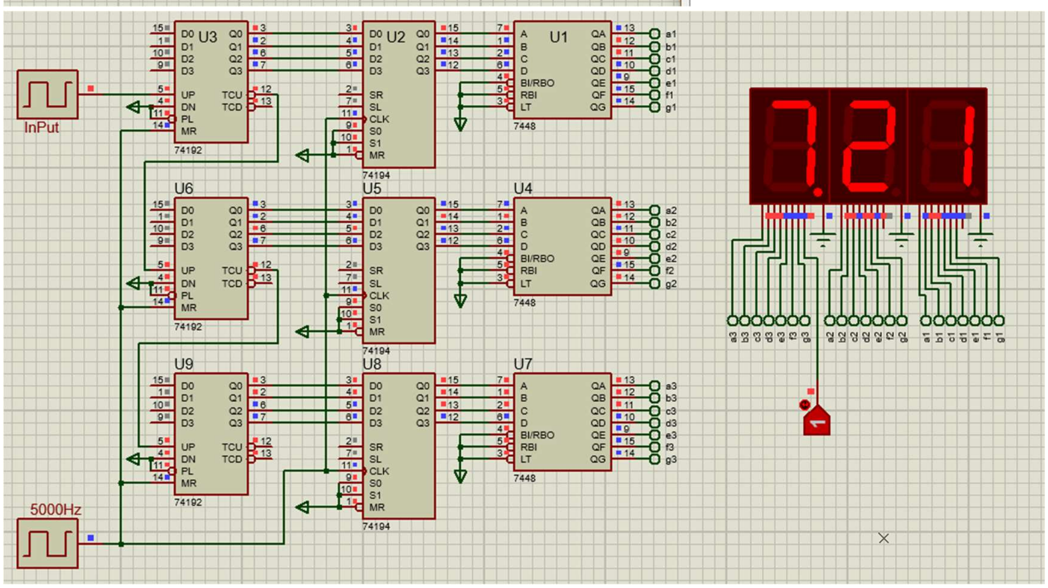
Symbol	Parameter	Test Conditions		Value							Unit
		V_{CC} (V)		T_A = 25 °C 54HC and 74HC			-40 to 85 °C 74HC		-55 to 125 °C 54HC		
				Min.	Typ.	Max.	Min.	Max.	Min.	Max.	
f_{MAX}	Maximum Clock Frequency	2.0		5	12		4		3.4		MHz
		4.5		25	48		$\boxed{20}$		17		
		6.0		30	55		24		20		

From 74192 Data Sheet we can see that it has a maximum limit of the clock frequency in different conditions considering the worst probability, which is in the black frame, we shouldn't get over 20 MHz, by that the IC ad restrictions, so in order to be in the safe zone I'll make my maximum limit is 10 MHz, which bring us to the next problem, how can I use all the three digits to display a number from 1 to 10 in MHz range? The answer is the decimal point, using the decimal point would allow me to use this range in most accurate possible. But to do that we need to change the MR Clock but this time the decimal point interfered, so how can we solve this problem?

The solution is in ignoring the exiting of the decimal point and make the law, which we used earlier to transfer into kilo range, more general. So if the numerator has an integer from 'n' digits in Hz and we want to display it in 3 digits only, we divide both numerator and denominator by 10^{n-3} and the denominator becomes the new time which is needed after it resetting the counter and start over, and as the denominator of the original law was 1 sec so we can say that the new time which needed after it resetting the counter equals 10^{3-n}, and by that the new MR Clock frequency equals $1/_{2*10^\wedge(3-n)}$ = 5 * 10^{n-4} Hz.

To apply this rule on our case: the required is projecting the MHz range using 1 digit for integer and 2 digits for floating like 2.15 MHz or 8.02 MHz, by turning into Hz range it becomes 2150000 Hz or 8020000 Hz, so the total digits are 7 digits and by that the new MR Clock frequency = 5 * 10^{7-4} = 5000 Hz.





And by that we can read in MHz range.

And now we go to the next problem which is how can we read in range from 1 Hz to 10 MHz? Here we should choose between the frequencies 0.5 Hz, 500 Hz, and 5000 Hz for the MR Clock which we used them earlier for the Hz, kHz, and MHz ranges, and by that we need something enable us to select between the three frequencies only one, and that will be by using multiplexers, but the problem would be in the function for the selected lines. With that we have to ask what happens to the circuit if the input frequency in out of the available range for it? The answer contains in the last TCU signal, being out of range meaning that I need more than 3 digits which makes the last TCU signal gives pulses which can be used to travel from range to another, then the function would be like next: if I made the default is the Hz “00” then the MR Clock equals 0.5 Hz then I watch the TCU signal, if it gave a pulse I move to the next range which is the kHz “01” then the MR Clock equals 500 Hz then I watch the TCU signal, if it gave a pulse I move to the next range which is the MHz “10” then the MR Clock equals 5000 Hz then I watch the TCU signal, if it gave a pulse then I’m out of range and bigger than 10 MHz “11” and the MR Clock becomes constant 1, high voltage, so to create this function I’ll connect a counter to the select lines of the multiplexer, and its input would be TCU signal.

We’re going to use 74153 IC to be our multiplexer.

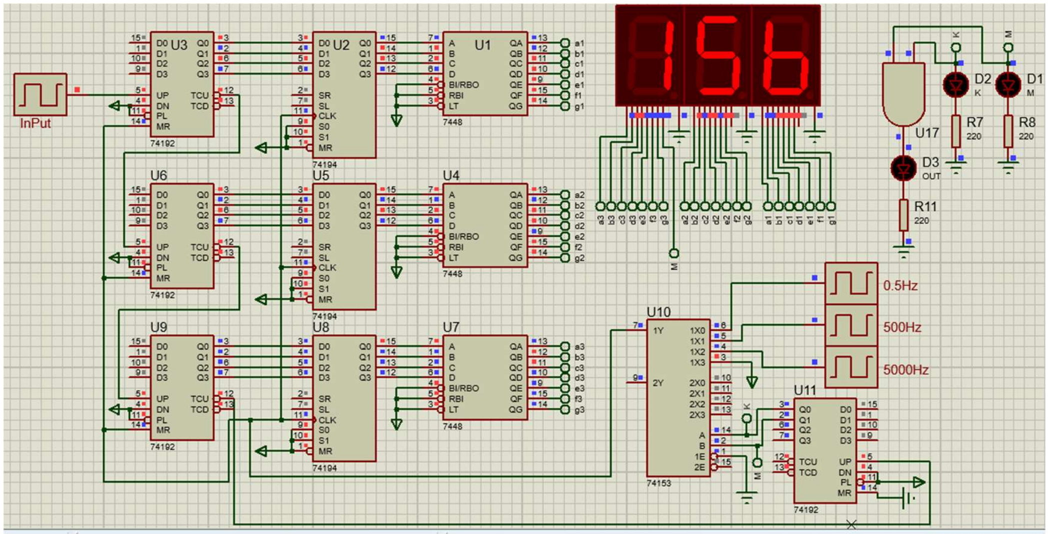
DM74LS153
Dual 1-of-4 Line Data Selectors/Multiplexers

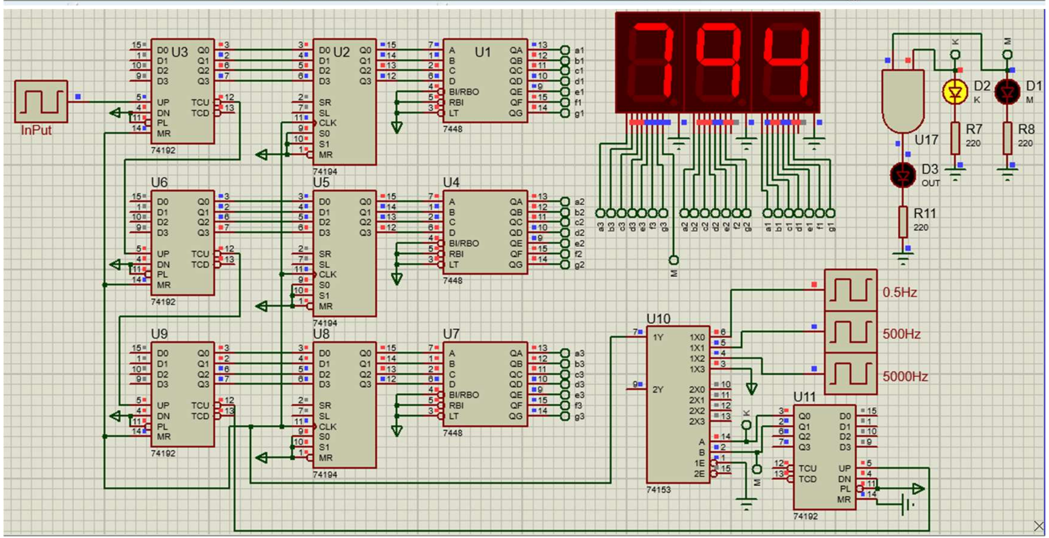
Function Table

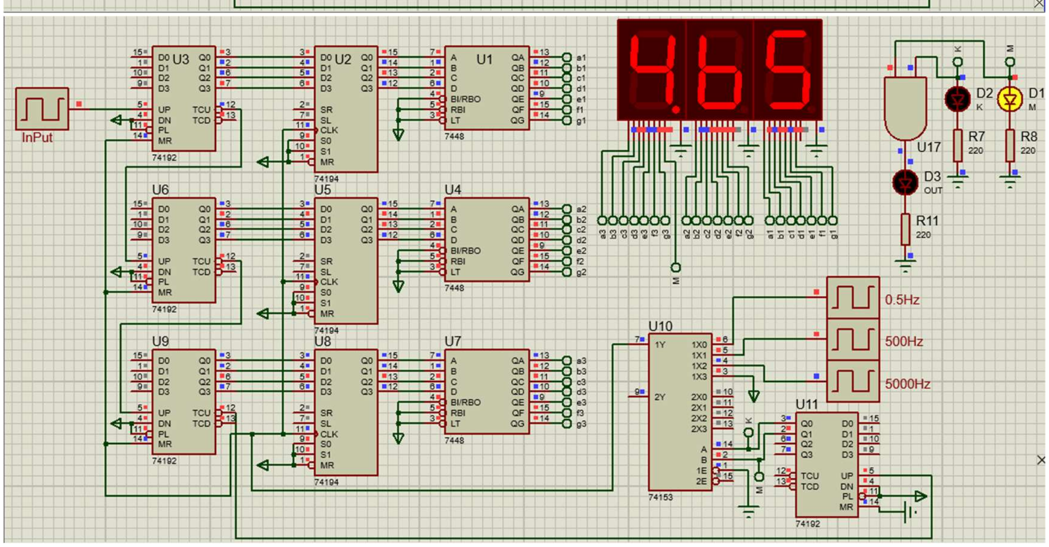
Select Inputs		Data Inputs				Strobe	Output
B	A	C0	C1	C2	C3	G	Y
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

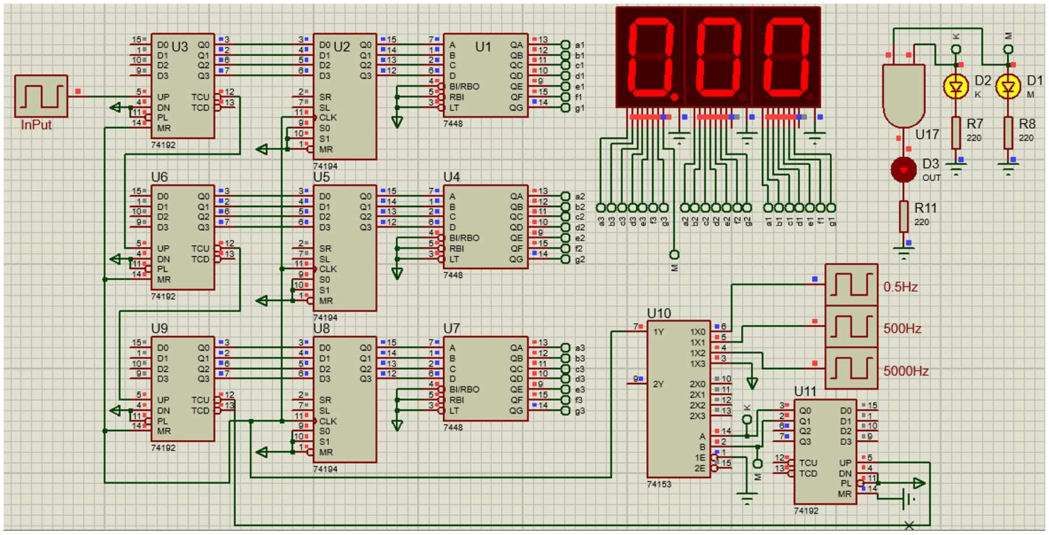
Select inputs A and B are common to both sections.
H = HIGH Level
L = LOW Level
X = Don't Care

We will take the select line “A” as indication for kHz, and the select line “B” as indication for MHz.


InPut
U3
U2
U1
U6
U5
U4
U9
U8
U7
U10
U11
U17
D1
D2
D3
R7
R8
R11
0.5Hz
500Hz
5000Hz
74192
74194
7448
74153


InPut
U3
U2
U1
U6
U5
U4
U9
U8
U7
U10
U11
U17
D1
D2
D3
R7
R8
R11
0.5Hz
500Hz
5000Hz
74192
74194
7448
74153

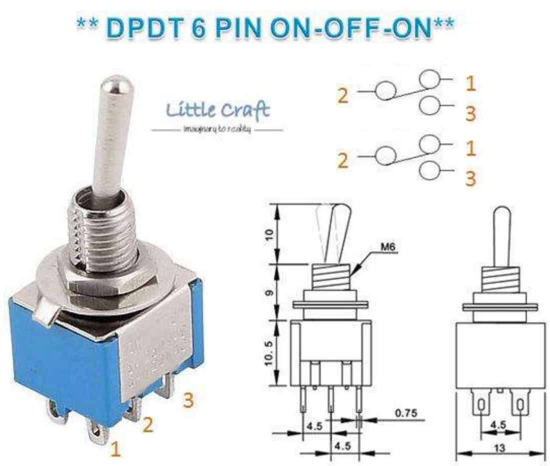

InPut
U3
U2
U1
U6
U5
U4
U9
U8
U7
U10
U11
U17
D1
D2
D3
R7
R8
R11
0.5Hz
500Hz
5000Hz
74192
74194
7448
74153



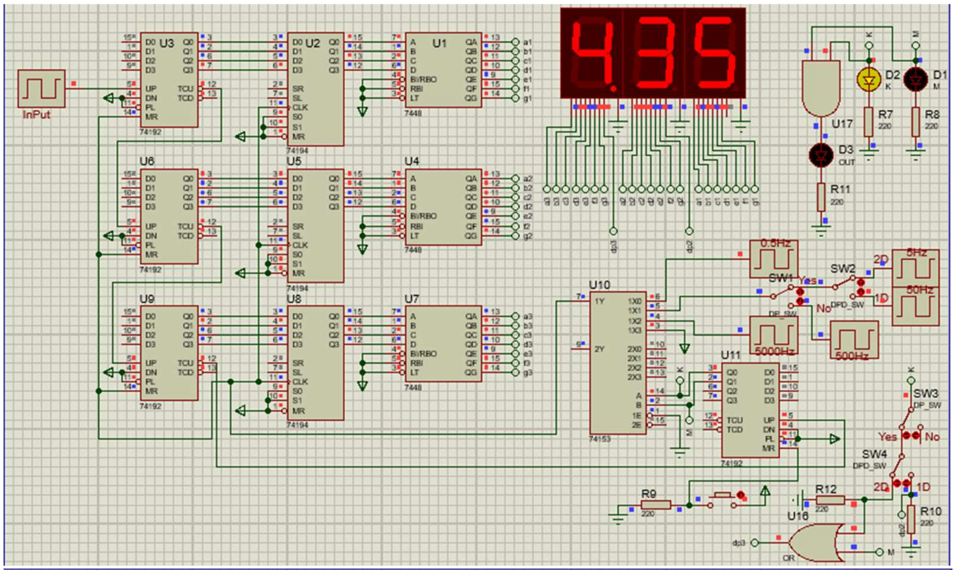
And by that we are able to read in range from 1 Hz to 10 MHz.

We are close to finish, we want to add some ranges in the kHz range itself, we want to divide it into three sections: 1_{st} from 1 to 10 kHz, 2_{nd} from 10 to 100 kHz, 3_{rd} from 100 kHz to 1 MHz. By that we have in the 1_{st} range 4 digits making the MR Clock frequency = 5 * 10^{4-4} = 5 Hz, in the 2_{nd} range we have 5 digits making the MR Clock frequency = 5 * 10^{5-4} = 50 Hz, and the 3_{rd} range has 500 Hz for the MR Clock.

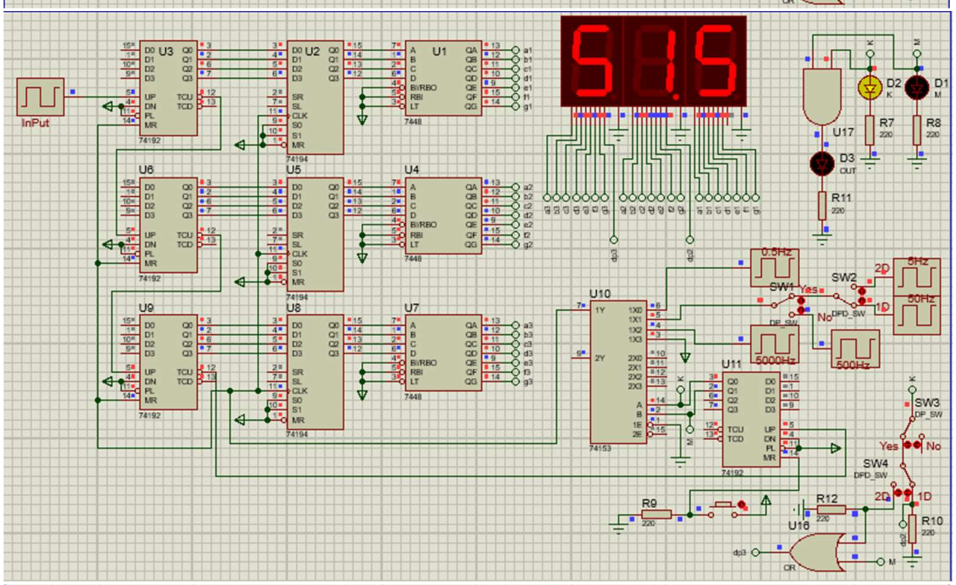
For choosing a range inside the kHz range we will do it manually not automatically, so we will use two of the toggle switch 6 PIN to be able to control the range and the decimal point in the same movement.



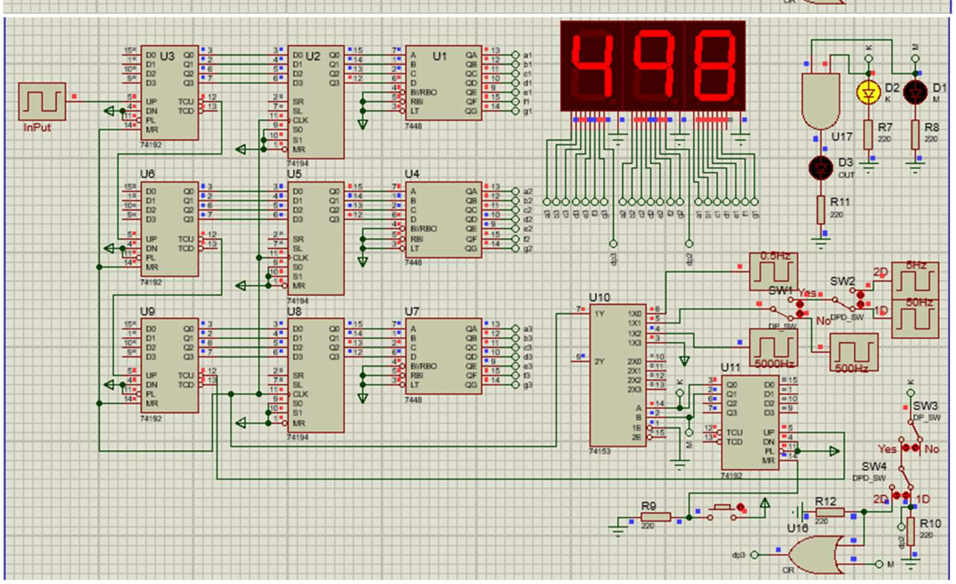
And we will adding a push button allows me to reset the circuit in case I chose a wrong range or an unsuitable range for my reading.



InPut
U3
U2
U1
U6
U5
U4
U9
U8
U7
U10
U11
U17
D1
D2
D3
R7
R8
R9
R10
R11
R12
SW1
SW2
SW3
SW4
0.5Hz
5Hz
50Hz
500Hz
5000Hz



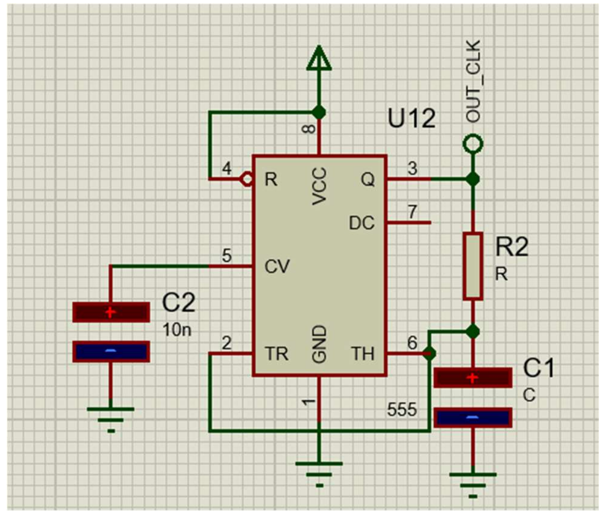
InPut
U3
U2
U1
U6
U5
U4
U9
U8
U7
U10
U11
U17
D1
D2
D3
R7
R8
R9
R10
R11
R12
SW1
SW2
SW3
SW4
0.5Hz
5Hz
50Hz
500Hz
5000Hz



InPut
U3
U2
U1
U6
U5
U4
U9
U8
U7
U10
U11
U17
D1
D2
D3
R7
R8
R9
R10
R11
R12
SW1
SW2
SW3
SW4
0.5Hz
5Hz
50Hz
500Hz
5000Hz

It only remains making the clock generator circuits, I'll choose the 555 IC method.

We will connect the circuit as it's shown to gain 50% duty cycle.



The output frequency can be calculated from the law

$$f = \frac{1}{\ln(2) * 2 * RC}$$

And we're going to use buffer to increase the strength of the output clock.

We're going to use 74245 IC.

SN74LS245

Octal Bus Transceiver

The SN74LS245 is an Octal Bus Transmitter/Receiver designed for 8-line asynchronous 2-way data communication between data buses. Direction Input (DR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable input ($\overline{E}$) can be used to isolate the buses.

- Hysteresis Inputs to Improve Noise Immunity
- 2-Way Asynchronous Data Bus Communication
- Input Diodes Limit High-Speed Termination Effects
- ESD > 3500 Volts

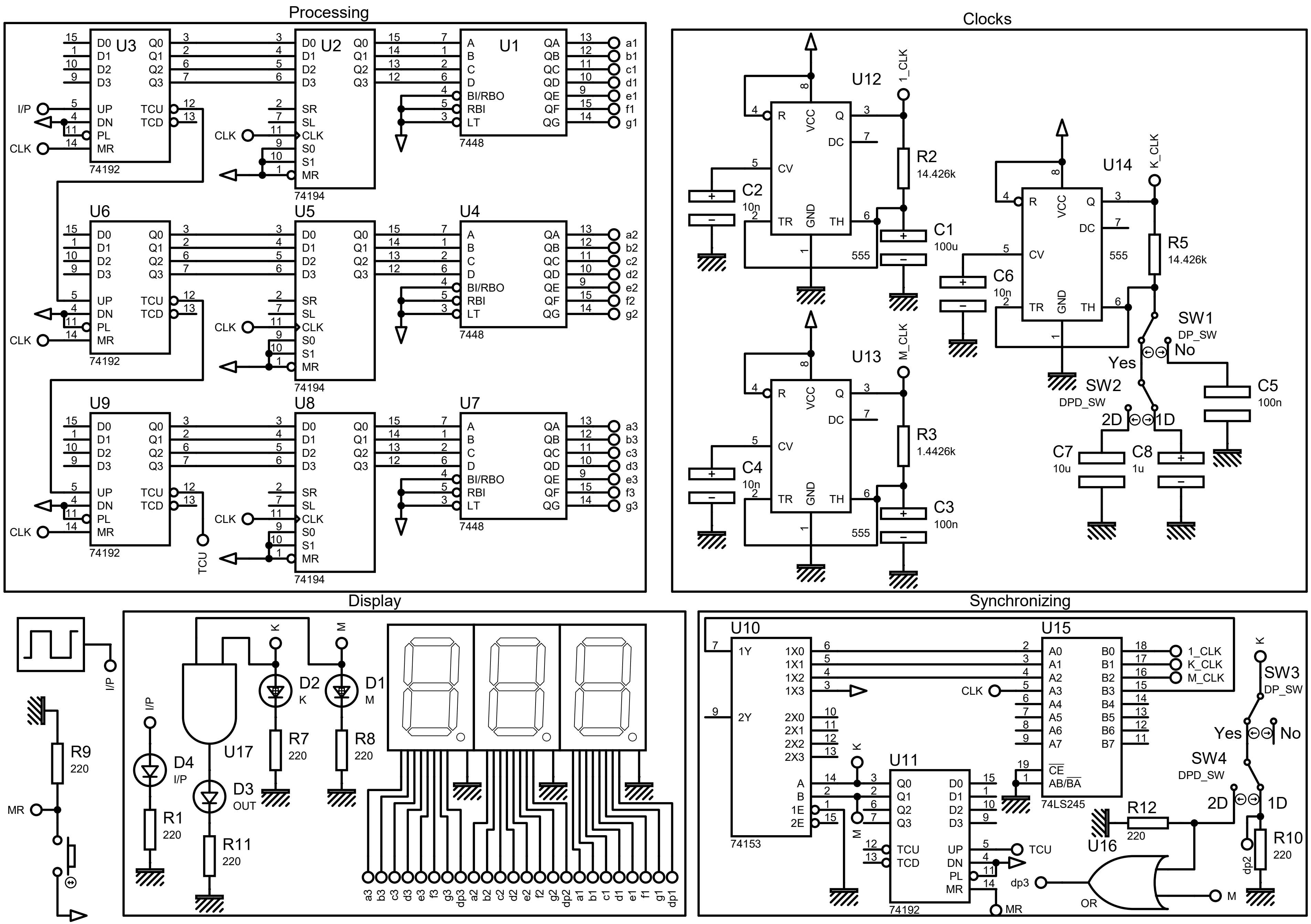
TRUTH TABLE

INPUTS		OUTPUT
$\overline{E}$	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	Isolation

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

And doing some arrangement in the circuit.

And we're done.


Processing
Clocks
Display
Synchronizing
U1 U2 U3 U4 U5 U6 U7 U8 U9
74192
74194
7448
U12 U13 U14
555
1_CLK
M_CLK
K_CLK
R2 14.426k
R3 1.4426k
R5 14.426k
C1 100u
C2 10n
C3 100n
C4 10n
C5 100n
C6 10n
C7 10u
C8 1u
SW1 DP_SW
SW2 DPD_SW
Yes
No
2D
1D
D1 M
D2 K
D3 OUT
D4 I/P
U17
R1 220
R7 220
R8 220
R11 220
R9 220
MR
I/P
CLK
TCU
U10 74153
U11 74192
U15 74LS245
U16 OR
R12 220
R10 220
SW3 DP_SW
SW4 DPD_SW
dp2
dp3
M
K