3.1

Truth Table NOR SC Latch

S	C		O/P
0	0		No Change
0	1		Clear
1	0		Set
1	1		Invalid

Truth Table NAND SC Latch

S	C		O/P
0	0		Invalid
0	1		Set
1	0		Clear
1	1		No Change

These latch designs are inverse, I it can be said that NOR is active HI, with NAND as active LO

Sequences of S C gave Q !Q:

11 → 10 → 00 :⇒ 00 → 10 → 10
11 → 01 → 00 :⇒ 00 → 01 → 01

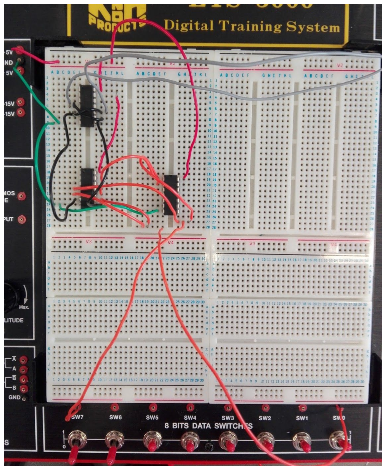
Attempting to switch between the no-change and the clash state does not render reproducible or consistent results, as the result is determined not by intent or intended pattern but by the small variation is initial conditions.

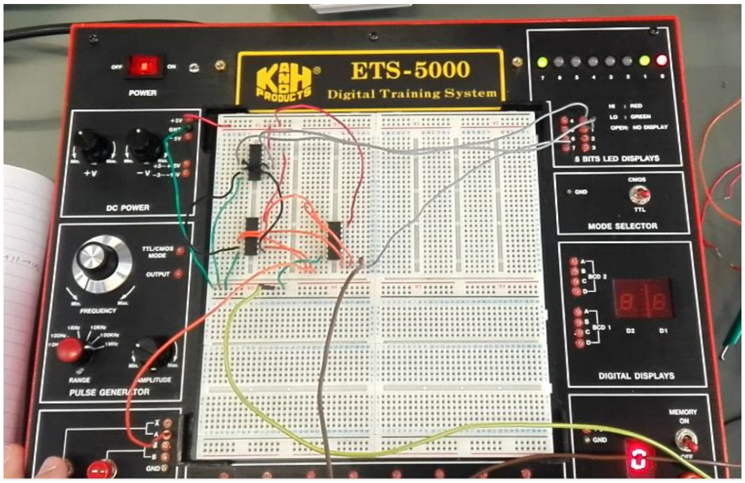
3.2

Implementing the AND steering circuit, with the enable line functions as intended, with the O/P only reactive on a E = 1

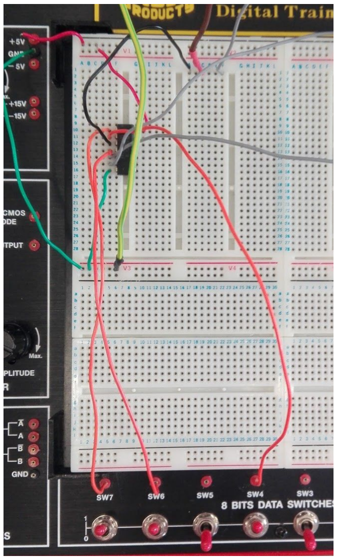
3.3

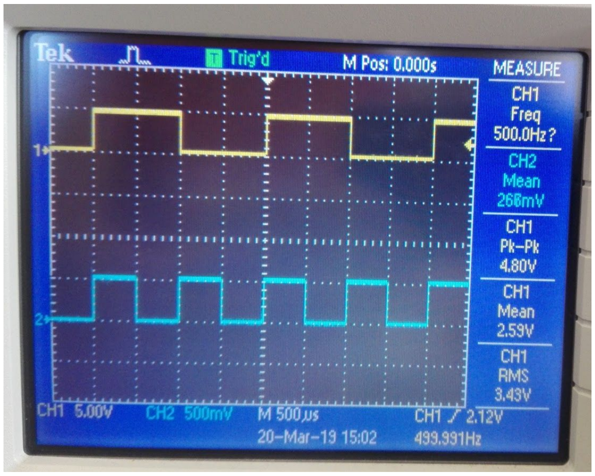
This allowed of the oscillating input to be sampled and temporality stored upon a pulse signal, and until the next pulse. Holding samples every state change.





3.4





Using the enable line to investigate state change response, we concluded that Q!Q output **only** change LO → HI CLK transition, not any specific state.

The !SET !RESET lines are complete asynchronous O/P overrides and have immediate effect no matter the CLK transition.

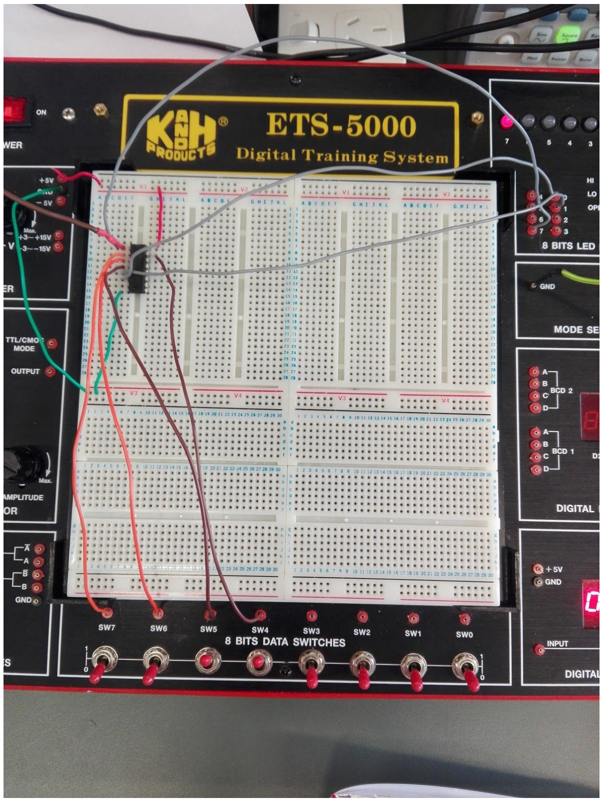
When connected to a steady signal CLK and in the toggle state;
You can observe the o/p Q (yellow) oscillates at ½ the frequency of the CLK i/p. Ie a frequency divider.

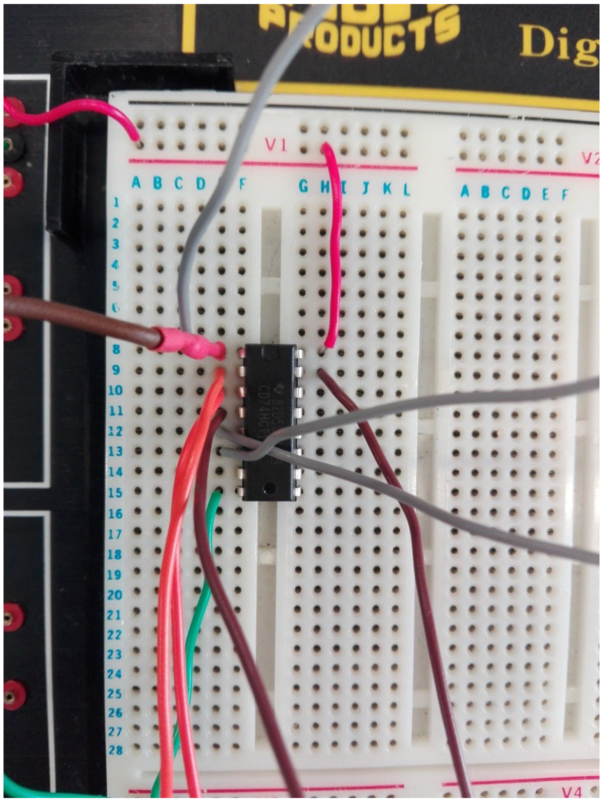
3.5

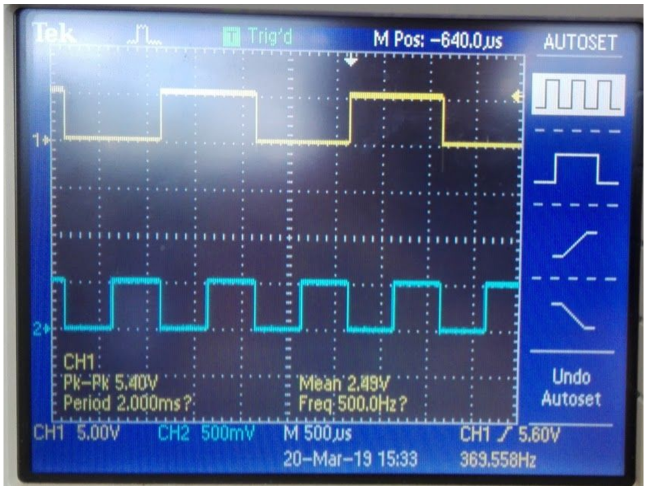
J	K	Q	!Q
0	0	1	0
0	1	0	1
0	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Table left shows the O/P state for a given I/P after the next CLK transition

Shown below is the circuit implemented and the oscilloscope showing the toggle state output dividing the clock IP in 2. Also showing the active transition as HI → LO



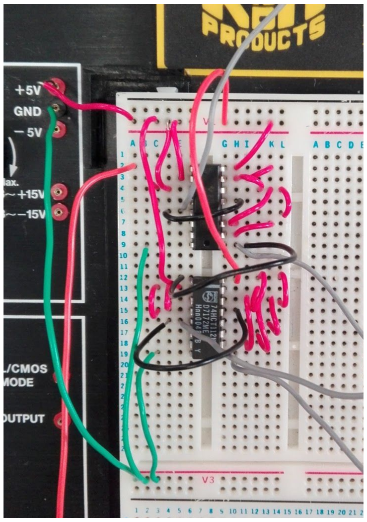




3.6 4bit ripple

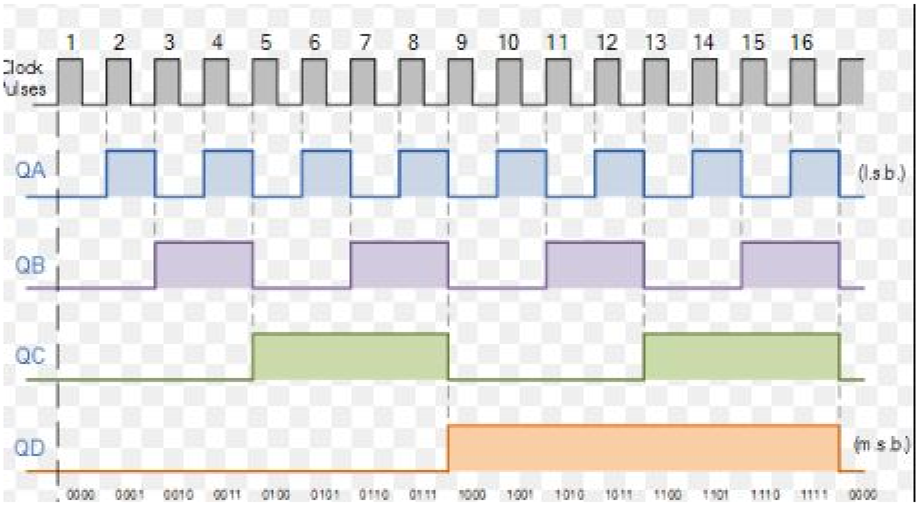
Using to duel JK chips; with the asynchronous overrides, inputs all wired HI and the SET output of the previous FF fed into the clk of the next was used to make a MOD16 counter. As each sequential FF's O/P toggles at ½ the frequency of the last the 8bit progression sequentially counts

Pulses in	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0



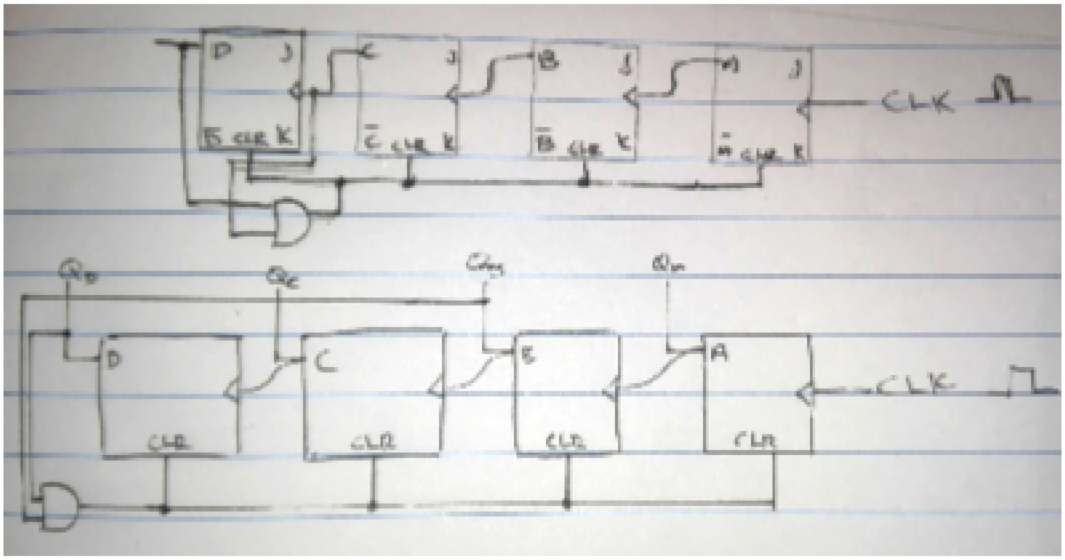
Using the LEDs and a pulse input, circuit successfully did a sequential count and reset

Outputs connected to the oscilloscope show each subsequent Q toggling at half that FF's input CLKs frequency ie:

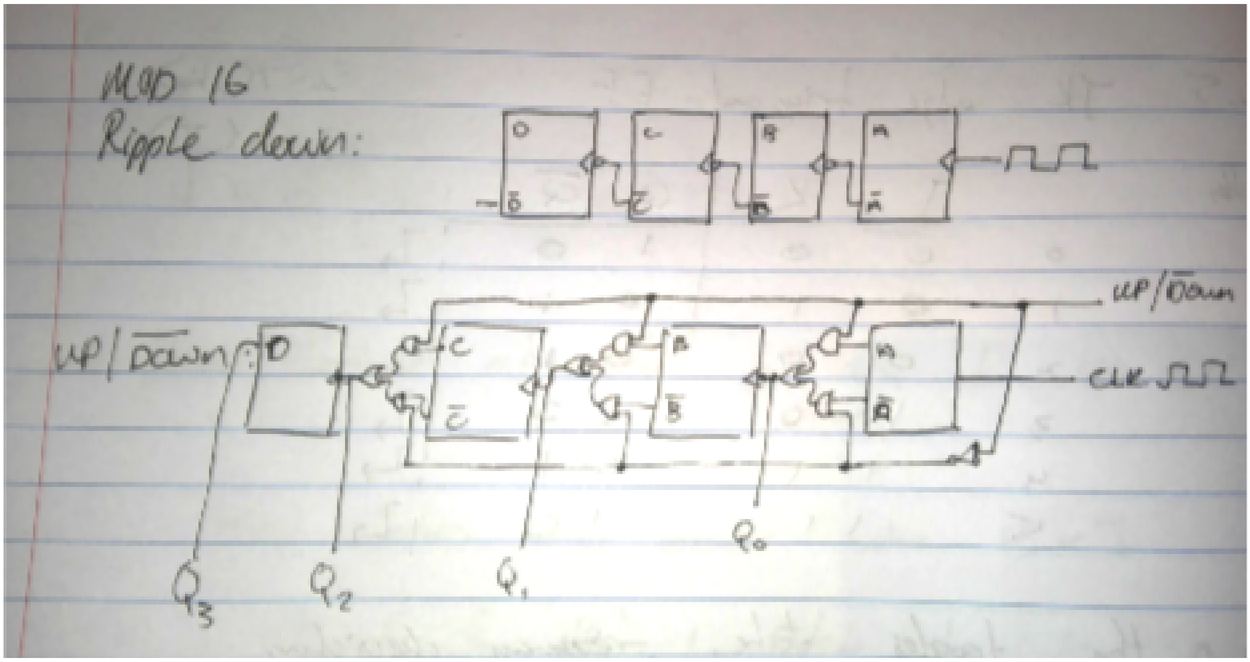


POST LAB

4.1

MOD 12 1100	 D J D̄ clk K C J C̄ clk K B J B̄ clk K A J Ā clk K CLK QD QC QB QA D C B A CLR CLR CLR CLR CLK
MOD 10 1010	

4.2


MOD 16
Ripple down:
D
D̄
C
C̄
B
B̄
A
Ā
up/$\overline{\text{Down}}$
up/$\overline{\text{Down}}$
D
C
C̄
B
B̄
A
Ā
CLK
Q3
Q2
Q1
Q0