1 Asynchronous Counter

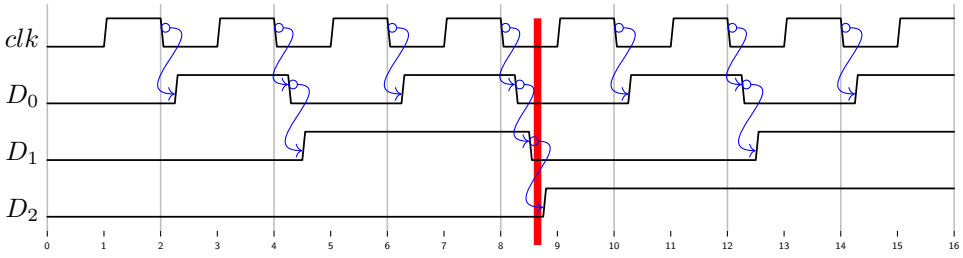

Figure 1: Asynchronous Counter (w/Fast Clock)

The daisy-chain shown in Fig 1 causes D_0 to change *in response to* the falling edges of *clk*, D_1 to change as a result of the falling edges of D_0, and D_2 to change as a result of the falling edges of D_1.

This design results in the more significant bits to update later than the least significant bits as a result of the effect of the "ticking" *clk*.

This delay is cumulative as the number of output bits in this type of counter increases. In some system designs this is OK. However, as the clock period is reduced (frequency increases) to the point where the propagation delays through all the flip-flops starts to become a significant percentage of it, things can become indeterminate. For example, what is the value of the counter $[D_0..D_2]$ at the point in time indicated by the red line? What *should* it be?

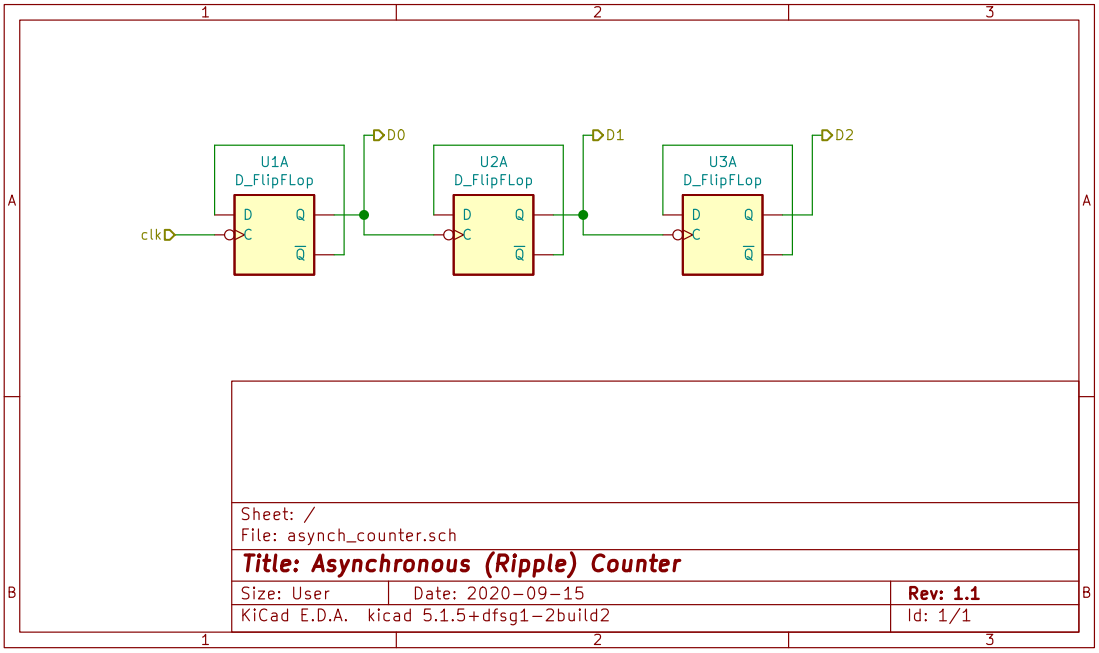

Figure 2: Asynchronous Counter Schematic

~/NIU/courses/463/2020-fa/notes/SyncAsyncCounters/counters.tex
jwinans@niu.edu 2020-09-18 09:41:21 -0500 v2.0-679-g93629c3

2 Synchronous Counter

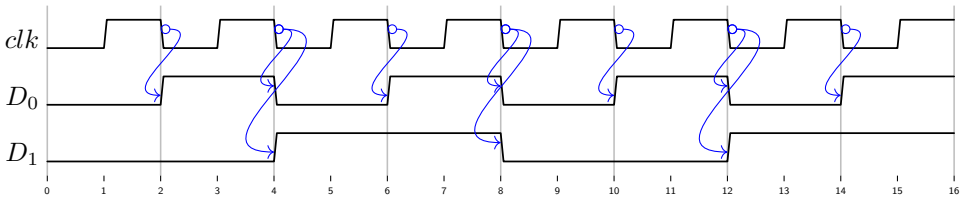

Figure 3: Synchronous Counter

In a synchronous counter the state of *all* the output bits (D_0 and D_1 in Fig!3) change at the same time in response to the *clk* signal.

One way to design a circuit that can count synchronously is to determine the *next state* of each of the output signal values by using a combinational circuit whose inputs are connected to the current output signal values. The *next state* signals may then be fed back into the inputs of latches that are used to store the *current state* of each of the output signals.

Expressing the next state in the form of a truth table can help illustrate this idea.

Current State		Next State	
D_1	D_0	N_1	N_0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Simplifying the output values...

$$
\begin{aligned}
N_1 &= (\overline{D_1} \wedge D_0) \vee (D_1 \wedge \overline{D_0}) && (1)\\
&= D_1 \oplus D_0 && (2)\\
N_0 &= (\overline{D_1} \wedge \overline{D_0}) \vee (D_1 \wedge \overline{D_0}) && (3)\\
&= (\overline{D_1} \vee D_1) \wedge \overline{D_0} && (4)\\
&= 1 \wedge \overline{D_0} && (5)\\
&= \overline{D_0} && (6)
\end{aligned}
$$

We can now add the N_0 and N_1 signals to our waveform diagram in Fig 4 and schematic in Fig 5 to see how they are used.

3 Observations

- Synchronous sequential circuits have the clock inputs to all latches connected to the same source-signal.
- Asynchronous sequential circuits have the clock inputs to their latches connected to different (or additional) signals.
- The MSb in a ripple counter changes after the LSb.
- Given a wide (many bits) enough ripple counter, the MSb can be delayed so long that it is not always obvious which clock cycle is causing it to change!

~/NIU/courses/463/2020-fa/notes/SyncAsyncCounters/counters.tex
jwinans@niu.edu 2020-09-18 09:41:21 -0500 v2.0-679-g93629c3

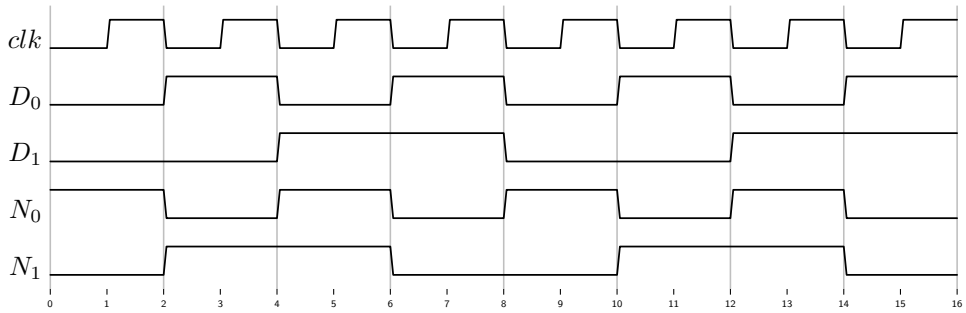

Figure 4: Synchronous Counter Waveform

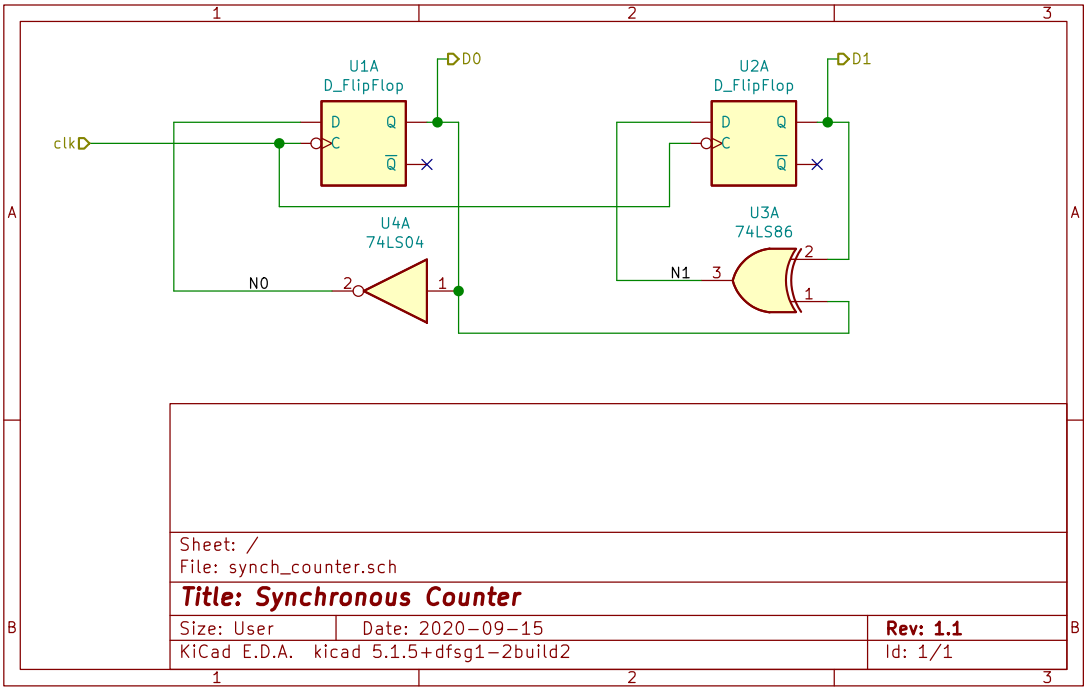

Figure 5: Synchronous Counter Schematic

~/NIU/courses/463/2020-fa/notes/SyncAsyncCounters/counters.tex
jwinans@niu.edu 2020-09-18 09:41:21 -0500 v2.0-679-g93629c3