

1 Timing 1 - Maximale Frequenz

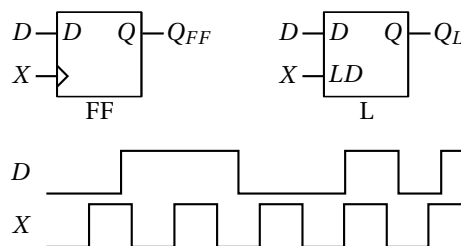
Implementieren Sie einen einfachen 3-Bit ripple-carry Binärzähler ohne reset, der nur FFs, Inverter und NANDs mit den unten angegebenen Eigenschaften enthält. Mit welcher Taktfrequenz kann der Zähler maximal betrieben werden?

$$\begin{aligned} \text{FF:} & \quad t_{\text{Setup}} = 3\text{ns} \quad t_{\text{hold}} = 2.5\text{ns} \quad t_{\text{CKQ}} = 9.5\text{ns} \\ \text{NAND:} & \quad t_p = 8\text{ns} \\ \text{INV:} & \quad t_p = 3.5\text{ns} \end{aligned}$$

2 Timing 2 - FF und Latch

Gegeben sei folgendes Timing-Diagramm und die Speicherelemente FF und L. Im Diagramm beschreibt D das Signal, das am Dateneingang der Speicherelemente anliegt und X das Steuersignal.

Zeichnen Sie den Ausgang Q_{FF} und Q_L in das Timing Diagramm ein. Worin unterscheiden sich beide Elemente?



3 Stoppuhr

Implementieren Sie eine Stoppuhr auf der Nexys2-Platine:

Es soll einen Start- und einen Stopp-Knopf geben, die die Zeitnahme aktivieren bzw. unterbrechen. Durch einen eigenen Knopf soll die Zeitanzeige wieder genullt werden (d. h. durch „Stopp“ und anschließenden „Start“ wird die Anzeige nicht zurückgesetzt, sondern läuft vom aktuellen Stand aus weiter). Außerdem soll es einen weiteren Knopf geben, durch den die Zeitanzeige angehalten wird, die Uhr währenddessen aber weiterläuft (*Zwischenzeit*). Durch erneuten Knopfdruck soll die Anzeige wieder in den fortlaufenden Modus zurückkehren.

Die Stoppuhr soll die Zeit in Schritten von Zehntelsekunden messen. Sie können zwischen mehreren Schwierigkeitsgraden auswählen:

- Anzeige der Zehntelsekunden als vierstellige Hexadezimalzahl
- Anzeige der Zehntelsekunden als vierstellige Dezimalzahl
- Anzeige von Minuten, Sekunden und Zehntelsekunden

Entwerfen Sie dazu zwei State-Machines in Verilog mittels case statements: Die erste soll zwischen start- und stop-Zuständen wechseln, die zweite soll die Zwischenzeit realisieren.