

Övningsuppgifter 2025-04-17

1. Konstruera en D-latch innehållande insignal D och *enable* samt utsignaler Q och Qn:

- Insignal *enable* ska användas för att låsa / låsa upp latchen; när *enable* = 1 är latchen öppen, annars är den låst.
- Insignal Q ska utgöra D-latchens ordinarie utsignal.
- Insignal Qn ska utgöra inversen av D-latchens ordinarie utsignal.

Därmed gäller följande:

$$enable = 1 \Rightarrow latchen \text{ är öppen} \Rightarrow Q = D, Qn = D'$$

$$enable = 0 \Rightarrow latchen \text{ är låst} \Rightarrow Q = Q, Qn = Qn$$

D-latchens utsignaler kan realiserars via följande ekvationer:

$$Q = (D' * Enable + Qn)'$$

samt

$$Qn = (D * Enable + Q)'$$

- Realisera motsvarande grindnät för hand och simulera i CircuitVerse.
- Testa att ändra insignal D när latchen är öppen respektive låst, notera utsignalerna.
- Testa kombinationer 00 – 11 av insignalerna två gånger, så att vi ser hur latchen beter sig när den blir låst efter att ha varit öppen och tilldelats D = 1, annars ser vi enbart när den låses när D = 0.

2. Konstruera en D-vippa innehållande insignaler *clock*, *reset_n*, D och *enable* samt utsignaler Q och Qn:

- Insignal *clock* ska utgöras av en systemklocka med godtycklig frekvens (dock 50 MHz på FPGA-kortet).
- Insignal *reset_n* ska utgöras av en inverterande reset-signal, när *reset_n* = 0 ska systemåterställning ske.
- Insignal *enable* ska användas för att låsa / låsa upp vippa; när *enable* = 1 är vippa öppen, annars är den låst.
- Utsignal Q ska utgöra D-vippans ordinarie utsignal.
- Utsignal Qn ska utgöra inversen av D-vippans ordinarie utsignal.

Därmed gäller följande:

$$reset_n = 0 \Rightarrow Systemåterställning \Rightarrow Q = 0, Qn = 1$$

$$reset_n = 1 \text{ och } enable = 1 \Rightarrow vippa \text{ är öppen} \Rightarrow Q = D, Qn = D' \text{ när klockan slår}$$

$$reset_n = 1 \text{ och } enable = 0 \Rightarrow vippa \text{ är låst} \Rightarrow Q = Q, Qn = Qn \text{ när klockan slår}$$

D-vippans utsignaler (utan asynkron reset) kan realiserars via följande ekvationer:

$$Q = (D' * enable * clock + Qn)'$$

samt

$$Qn = (D * enable * clock + Q)'$$

- Realisera motsvarande grindnät för hand och simulera i CircuitVerse. Sätt klockans periodtid till 1000 ms.
- Testa att ändra insignal D när vippa är öppen respektive låst, notera utsignalerna och hur de följer klockpulserna i normalfall. Testa sedan utsignalerna vid systemåterställning (då *reset_n* = 0).
- Om Q = 1 när systemåterställning sker, dröjer det tills klockan slår innan Q nollställs eller sker det direkt? Förklara varför/varför inte.

Tips:

- Lägg till en AND-grind med insignaler Q samt *reset_n* på D-vippans utgång för att enkelt implementera en asynkron inverterande reset så att vippans utsignaler återställs direkt utan att vänta på klockan vid
- Den inverterande utsignalen Qn kan enkelt sättas till inversen av den utsignalen Q via en NOT-grind.