



## Duplicate Flip-Flops

Um das Timing zu verbessern, dupliziert Quartus häufig Flip-Flops. Diese Duplikate verbessern zwar tatsächlich das Timing (was innerhalb von CHD3 nicht weiter wichtig ist) stören aber die Übersicht (was innerhalb von CHD3 durchaus wichtig ist). Der Einbau von Flip-Flop-Duplikaten lässt sich mit folgender Anweisung in der .qsf-Datei unterbinden:

```
set_global_assignment -name ALLOW_REGISTER_DUPLICATION OFF
```

Alternativ kann man dies auch mit der graphischen Oberfläche erreichen: Quartus — Assignments — Settings... — Category: Compiler Settings — Advanced Settings (Synthesis)... — Allow Register Duplication : Off.

### 1. Aufgabe Latches und Flip-Flops

Im Verzeichnis `unitFForLatch` finden Sie die VHDL-Beschreibungen (`Rtl_A` bis `Rtl_H`).

Das Synthesetool erzeugt Technologieschaltungen aus ein oder mehreren D-Latches und/oder ein oder mehreren D-Flip-Flops.

Untersuchen Sie den Zusammenhang der unterschiedlichen Beschreibungen mit dem Syntheseergebnis. Erläutern Sie, warum das jeweils zu beobachtende Syntheseergebnis entsteht. Insbesondere interessant ist die Frage, warum das Synthesetool bei manchen Beschreibungen Latches bei anderen dagegen Flip-Flops erzeugt. Wie erklären Sie sich den Unterschied, der sich ergibt, je nachdem, ob `iD` in der *Sensitivity-List* enthalten ist oder nicht.

Welche *Warnings* oder *Critical Warnings* erhalten Sie in Quartus und wie interpretieren Sie diese?

Gibt es einen Unterschied zwischen `(iClk'event and iClk = '1')` und `rising_edge(iClk)`? Suchen Sie hierzu die Definition von `rising_edge()` im *Source-Code* des *Package* `ieee.std_logic_1164`. Diesen finden Sie neben einigen Quellen im Internet auch im Installationsverzeichnis des Simulators Questa- oder Modelsim.

### 2. Aufgabe D-Flipflop

Erstellen Sie eine VHDL-Beschreibung für ein positiv-flankengesteuertes D-Flip-Flop mit asynchronem Rücksetzeingang `inResetAsync` (negative Logik!). Der Dateneingang heißt `iD`, der Takteingang `iClk`. Das Flipflop stellt zwei Ausgänge zur Verfügung: Der normale Datenausgang trägt den Namen `oQ`, der invertierte Datenausgang den Namen `onQ`.

Erweitern Sie Ihre VHDL Beschreibung um einen *synchronen* Setzeingang `iSet` und einen Enable-Eingang `iEnable` (Datenübernahme findet nur bei aktivem `Enable` statt, das synchrone Setzen erfolgt dagegen unabhängig von `Enable`).

Weisen Sie die korrekte Funktion des Flip-Flops mittels einer entsprechenden *Testbench* nach. Achten Sie dabei auf die Vollständigkeit der Anregungen in der Simulation.

Synthetisieren Sie das Flipflop und prüfen Sie, ob das Ergebnis Ihren Erwartungen entspricht.

Das Flipflop soll nun auf dem Board realisiert werden. Hierzu wird ein *Testbed* verwendet.

Für diese Aufgabe wird der Takt `iClk` mit einem Taster/Schalter von Hand erzeugt, verwenden Sie daher einen entsprechenden Taster/Schalter. Jeder Tastendruck soll eine steigende Flanke erzeugen. Falls beim Drücken eine fallende Flanke erzeugt wird, negieren Sie den von der Taste kommenden Wert innerhalb des *Testbeds*. Ein Loslassen der Taste führt zu einer fallenden Flanke. Wie hoch ist der maximale Takt, den Sie auf diese Weise vormittags/nachmittags/abends erzeugen können?

Um das Flipflop auf dem Board testen zu können, schliessen Sie bitte die restlichen Eingänge `inResetAsync`, `iSet`, `iEnable` und `iD` an geeignete Tasten/Schalter an.

Bitte stets alle Signale innerhalb des *Testbeds* in positive Logik übersetzen, nie jedoch das Reset-Signal. Dieses wird durchgängig in negativer Logik ausgeführt. Die Datenausgänge des Flipflops visualisieren Sie bitte mit Hilfe zweier LEDs.

Stellen Sie nun das Flip-Flop auf Steuerung durch die fallende Taktflanke um und Probieren Sie bei sonst unverändertem Entwurf, ob nun tatsächlich das Loslassen der Taktaste neue Werte einspeichert.

Die Tasten sind auf dem Board in aufwändiger Weise mittels einer analogen Schaltung entprellt. Verwenden Sie in einem weiteren Versuch einen Schiebeschalter als Taktquelle. Können Sie mittels dieser Schaltung zeigen, dass die Schalter prellen?

### **3. Aufgabe Allflankengesteuertes D-FlipFlop**

Versuchen Sie, ein D-FlipFlop (mit oder ohne Reset) zu beschreiben, das sowohl auf die steigende als auch die fallende Flanke Daten übernimmt. Synthetisieren Sie Ihren Entwurf. Welches Ergebnis erhalten Sie?

### **4. Aufgabe Schieberegister**

Ausgehend von der Beschreibung eines D-Flip-Flops mit einem asynchronen Reset-Eingang und einem (nichtinvertierten) Datenausgang `oQ` erweitern Sie nun auf ein Schieberegister mit der Länge (=Anzahl hintereinandergeschalteter Flip-Flops) von 9. Es soll also die Beschreibung des D-Flip-Flops zu einem Schieberegister umgearbeitet werden. Das Schieberegister soll nicht durch Instantiieren von D-Flip-Flops aufgebaut werden.

Angezeigt werden `iD` und alle `oQ` der 9 Flip-Flops auf den LEDs unmittelbar vor den Schaltern des Boards. Die Flip-Flops reagieren auf die steigende Taktflanke, welche wieder durch Tastendruck erzeugt wird. Auch `iD` und `inResetAsync` werden mittels Tasten erzeugt.

Wäre es möglich, die Flip-Flops des Schieberegisters durch Latches zu ersetzen?