

6 Versuch Nr. 5

6.1 Anmerkungen zum Versuch Nr. 5

In den bisherigen Versuchen haben Sie sich mit kombinatorischen Schaltkreisen beschäftigt, in denen die Ausgänge bisher nicht auf die Eingänge zurückgeführt wurden. Im folgenden Versuch werden Sie sich mit rückgekoppelten Schaltnetzen, d.h. mit Schaltnetzen beschäftigen, bei denen die Ausgänge ganz oder teilweise wieder auf die Eingänge zurückgeführt sind. Sie lernen Flip-Flops kennen, den wichtigsten Vertreter der rückgekoppelten Schaltkreise, und Sie werden mit Flip-Flops eine Anwendung mit der Altera-Software MAX+PlusII programmieren.

6.2 Literaturhinweis zu Versuch Nr. 5

- Tietze, Schenk: Halbleiterschaltungstechnik (Kap. 10)
- Keller / Paul: Hardwaredesign (Kap. 5)
- L. Borucki: Digitaltechnik (Kap. 7, 8)

6.3 Aufgabenstellung vor Versuchsdurchführung

6.3.1

- Was ist die wichtigste Eigenschaft von Flip-Flops ?
- Wo finden Flip-Flops ihre Anwendung ?

6.3.2

In Abbildung 1 ist der Aufbau eines RS-Flip-Flops mit NAND-Gattern zu sehen.

- Erläutern Sie das logische Verhalten dieser Schaltung.
- Was versteht man unter der reduzierten Wahrheitstabelle, bzw. unter verbotenen Eingangszuständen ? Erklären Sie warum es verbotene Eingangszustände gibt, und weshalb dies so ist.

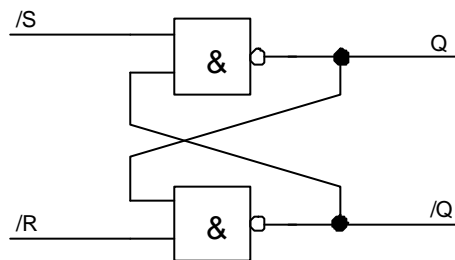


Abbildung 1: RS-Flip-Flap mit NAND-Gattern

6.3.3

In Abbildung 2 ist der Aufbau eines RS-Flip-Flops mit NOR-Gattern zu sehen.

- Erläutern Sie den Unterschied zu dem aus NAND-Gattern bestehendem RS-Flip-Flop aus Abschnitt 6.3.2.

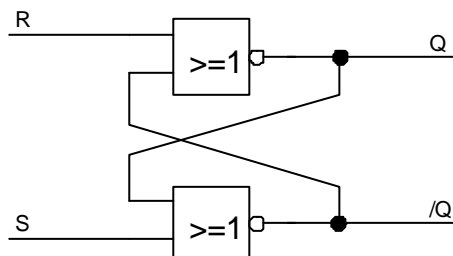


Abbildung 2: RS-Flip-Flop aus NOR-Gattern

6.3.4

In Abbildung 3 ist der Aufbau eines D-Flip-Flops mit Set- und Clear-Eingang zu sehen.

- Was ist der Vorteil des D-Flip-Flops gegenüber einem einfachen RS-Flip-Flop ?
- Erläutern Sie das logische Verhalten dieser Schaltung unter der Voraussetzung, dass Set und Clear = 0 sind.
- Welche Funktion haben Set und Clear ? Erläutern Sie nun das logische Verhalten der Schaltung.

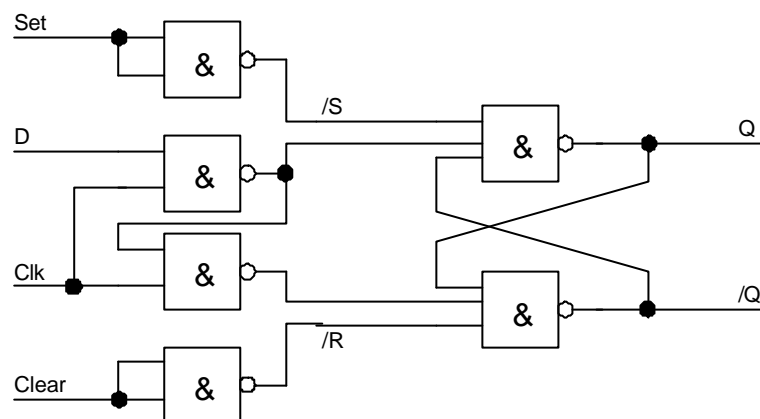


Abbildung 3: D-Flip-Flop

6.3.5

In Abbildung 4 ist das Blockschaltbild eines 8-Bit-Zählers skizziert.

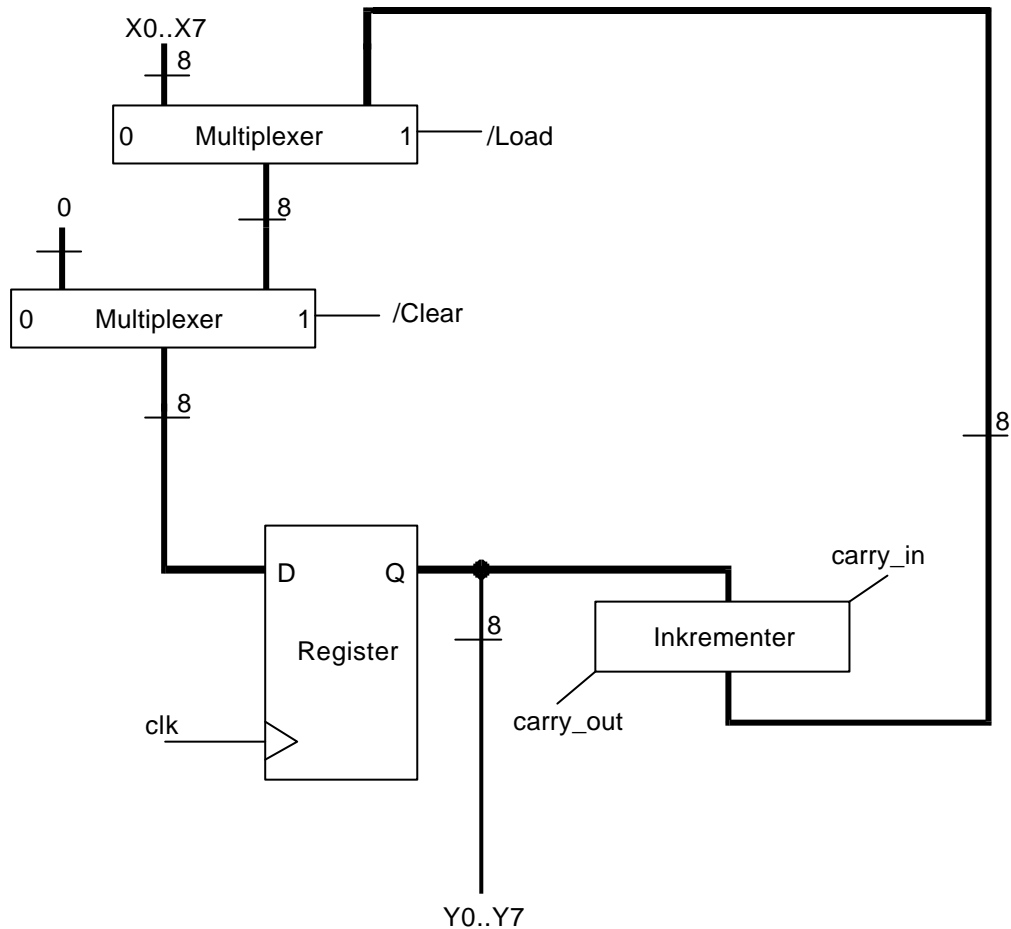


Abbildung 4: 8-Bit-Zähler

Der 8-Bit-Zähler besteht aus zwei 8-Bit-Multiplexern, einem 8-Bit-Register und einem 8-Bit-Inkrementier, und besitzt die folgenden Ein- und Ausgänge:

- Dateneingänge $X = (X0 \dots X7)$
- Datenausgänge $Y = (Y0 \dots Y7)$
- Dateneingang $carry_in$ für einen Eingangsübertrag
- Datenausgang $carry_out$ für einen Ausgangsübertrag
- Clockeingang clk
- Kontrollsignal $Clear (/Clear)$ und $Load (/Load)$

6.3.5.1

- Erläutern Sie die Funktion dieses 8-Bit-Zählers an Hand des Blockschaltbildes.

6.3.5.2

- Erläutern Sie das logische Verhalten eines Multiplexers.
- Entwerfen Sie einen 8-Bit-Multiplexer, bestehend aus AND-, OR-Gattern und Invertern. Skizzieren Sie den entsprechenden Schaltkreis.
- Skizzieren Sie ein mögliches Zeitdiagramm (Vorlage für ein SCF-File), um den Multiplexer mit der Altera MAX+PlusII-Software später simulieren zu können.

6.3.5.3

- Erläutern Sie das logische Verhalten eines Registers.
- Entwerfen Sie ein 8-Bit-Register bestehend aus D-Flip-Flops. Skizzieren Sie den entsprechenden Schaltkreis.
- Skizzieren Sie ein mögliches Zeitdiagramm (Vorlage für ein SCF-File), um das 8-Bit-Register mit der Altera MAX+PlusII-Software später simulieren zu können.

6.3.5.4

- Erläutern Sie das logische Verhalten eines Inkrementers.
- Wie können Sie einen 8-Bit-Inkrementer aufbauen, wenn Sie Ihren in Versuch 4 entworfenen 8-Bit-Addierer verwenden möchten ?

6.3.5.5

- Skizzieren Sie ein mögliches Zeitdiagramm (Vorlage für ein SCF-File), um den 8-Bit-Zähler mit der Altera MAX+PlusII-Software später simulieren zu können.

6.4 Versuchsdurchführung

6.4.1

Die in Kapitel 6.3.2 dargestellte RS-Flip-Flop-Schaltung soll mit dem IC-Steckkasten aufgebaut werden.

- Nehmen Sie das logische Verhalten der Schaltung auf, in dem Sie über das Schalterfeld die Eingangsvariablen $/R$ und $/S$ anlegen. Versuchen Sie Ihre theoretischen Erwartungen nachzuvollziehen.
- Geforderte Dokumentation:
 - Messtabelle der Aufnahme des logischen Verhaltens der Schaltung.
 - Kennzeichnung der unzulässigen Eingangsbedingung.

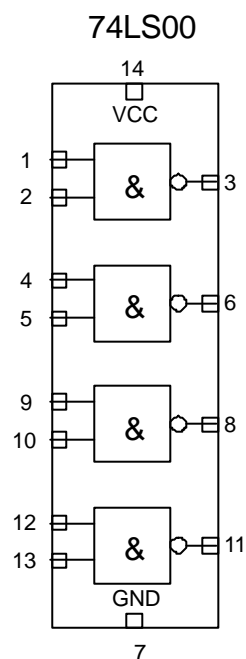


Abbildung 5: Zu verwendender Baustein zum Aufbau des RS-Flip-Flops

6.4.2

Die in Kapitel 6.3.4 dargestellte D-Flip-Flop-Schaltung soll mit dem IC-Steckkasten aufgebaut werden.

- Nehmen Sie zunächst das logische Verhalten der Schaltung auf, in dem Sie über das Schalterfeld die Eingangsvariablen `D` und `Clk` anlegen, und `Set` und `Clear` auf `0` legen. Versuchen Sie Ihre theoretischen Erwartungen nachzuvollziehen.
- Nehmen Sie nun noch das logische Verhalten der Schaltung unter Verwendung der Signal `Set` und `Clear` auf. Versuchen Sie auch hier Ihre theoretischen Erwartungen nachzuvollziehen.
- Geforderte Dokumentation:
 - Gesamt-Messtabelle der Aufnahme des logischen Verhaltens der Schaltung.
 - Kennzeichnung der unzulässigen Eingangsbedingungen.

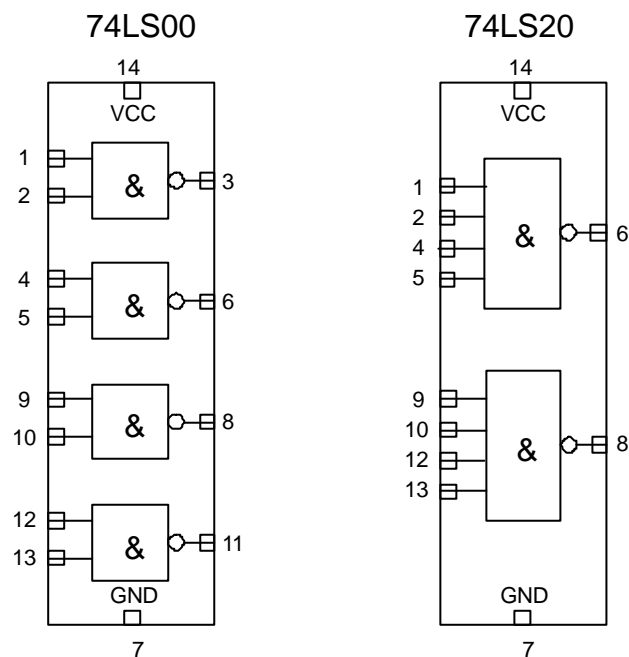


Abbildung 6: Zu verwendende Bausteine zum Aufbau des D-Flip-Flops

6.4.3

Der in Kapitel 6.3.5 abgebildete 8-Bit-Zähler soll nun mit der Altera MAX+PlusII-Software eingegeben werden, und - wie auch schon im letzten Versuch durchgeführt - in den EPM7128SLC-10-Baustein programmiert und danach getestet werden. Gehen Sie dazu wie in den folgenden Aufgaben beschrieben vor.

6.4.3.1

- Geben Sie den von Ihnen entworfenen 8-Bit-Multiplexer als GDF-File ein. Erzeugen Sie sich von dem Multiplexer ein Symbol, um diesen hinterher in weiteren Schaltkreisentwürfen einsetzen zu können.
- Kompilieren Sie Ihr Design für das Device EPM7128SLC-10.
- Erstellen Sie ein SCF-File, um den Multiplexer simulieren zu können. Simulieren Sie schließlich den Multiplexer, und prüfen Sie Ihr Ergebnis.
- Geforderte Dokumentation:
 - Eingegebener Schaltkreis (GDF-File)
 - Das SCF-File vor der Simulation
 - Das SCF-File nach der Simulation

6.4.3.2

- Geben Sie das von Ihnen entworfene 8-Bit-Register als GDF-File ein. Verwenden Sie die in der Bibliothek vorhandenen D-Flip-Flops (DFF). Diese D-Flips-Flops verfügen über eine Set- und Clear-Eingang (PRN und CLRN), die Sie in Ihrem Design auf VCC legen. Erzeugen Sie sich von dem Register ein Symbol, um diesen hinterher in weiteren Schaltkreisentwürfen einsetzen zu können.
- Kompilieren Sie Ihr Design für das Device EPM7128SLC-10.
- Erstellen Sie ein SCF-File, um das Register simulieren zu können. Simulieren Sie schließlich das Register, und prüfen Sie Ihr Ergebnis.
- Geforderte Dokumentation:
 - Eingegebener Schaltkreis (GDF-File)
 - Das SCF-File vor der Simulation
 - Das SCF-File nach der Simulation

- Kompilieren Sie Ihr Design für das Device EPM7128SLC-10.
- Erstellen Sie ein SCF-File, um den Inkrementer simulieren zu können. Simulieren Sie Inkrementer, und prüfen Sie Ihr Ergebnis.
- Geforderte Dokumentation:
 - Eingebener Schaltkreis (GDF-File)
 - Das SCF-File vor der Simulation
 - Das SCF-File nach der Simulation

6.4.3.4

- Geben Sie nun abschließend den 8-Bit-Zähler aus Abbildung 4 als GDF-File unter der Verwendung der zuvor erzeugten Symbole Multiplexer, Inkrements und Register ein.
- Kompilieren Sie Ihr Design für das Device EPM7128SLC-10, und betrachten Sie sich das Ergebnis im Floorplan-Editor.
- Die willkürlich getroffenen I/O-Belegungen des Compilers sollen wie folgt im Floorplan-Editor abgeändert werden:

Pin	IO-Port / Pin-Nr.
Eingangübertrag <code>carry_in</code>	11
Ausgangsübertrag <code>carry_out</code>	74
Clock <code>clk</code>	83
Kontrollsignal / <code>Clear</code>	33
Kontrollsignal / <code>Load</code>	34
Eingänge X0	15
X1	16
X2	17
X3	18
X4	20
X5	21
X66	22
X7	24
Ausgänge Y0	65
Y1	64
Y2	63
Y3	58
Y4	57
Y5	56
Y6	55
Y7	54

Vergessen Sie nicht, das Design nochmals zu kompilieren.

Anmerkung: Als Clock (`clk`) soll der auf dem Baustein vorhandene Takt des 50-MHz-Oszillators verwendet werden, der an Pin 83 (Global Clock `GCLK1` des EPM7128SLC-10) angeschlossen ist.

- Erstellen Sie ein SCF-File, um den 8-Bit-Zähler simulieren zu können. Simulieren Sie diesen, und prüfen Sie Ihr Ergebnis.

Anmerkung:

- Da Ihnen ein 50MHz-Takt zur Verfügung steht (20ns-Periodendauer) müssen Sie Ihr Grid-Size im Options-Menü auf 10ns einstellen.
 - Um eine Clock-Waveform (clk) zu erzeugen, müssen Sie die gesamte clk-Node selektieren, und Overwrite Clock im Edit-Menü auswählen. Wählen Sie OK, um die Default-Werte zu akzeptieren.
- Ist das Ergebnis der Simulation korrekt, so programmieren Sie den Baustein auf dem HWPRAK-Altera-Board, und überprüfen Sie Ihren 8-Bit-Zähler, in dem Sie an die entsprechenden IO-Ports des Bausteines mit kleinen Verbindungsleitungen Schalter zur Eingabe bzw. LEDs zur Ausgabe anschließen.
- Geforderte Dokumentation:
 - Eingegebener Schaltkreis (GDF-File)
 - Der Device View des gerouteten 8-Bit-Addierers
 - Das SCF-File vor der Simulation
 - Das SCF-File nach der Simulation