



FH Salzburg

Informationstechnik & System-Management

Laborübung 5

Ziel:

- Verständnis von synchronen Schaltwerken
- Umsetzung einer Schaltung mit einem FPGA.

Vorbereitung:

- Studieren Sie die vorherigen Laborübungen und die Kapitel „Asynchrone Schaltwerke“ und „Synchrone Schaltwerke“ aus der Vorlesung.
- Studieren Sie die Übungsanleitung, den Versuchsaufbau und die Projektvorlage aus Moodle.
- Bereiten Sie die für die Übung notwendigen Vorbereitungsfragen (siehe Moodle) vor.

Abgabe pro Aufgabenpunkt:

- Beantworten Sie die Fragen aus diesem Dokument präzise
- Erstellen Sie Notizen / Berechnungen
- Erstellen Sie Skizzen / Schaltungen
- Fertigen Sie Fotos vom Aufbau am Steckbrett bzw. einen Screenshot von Quartus Prime File mit Compilation-Anzeige (grüne Hakerl) an

Nachbereitung:

- Schreiben Sie einen kurzen Laborbericht mit den in „Abgabe pro Aufgabenpunkt“ genannten Anforderungen

Labordurchführung

1. Einführungsaufgabe: Kennenlernen des Arduino MKR Vidor 4000, Quartus Prime und der Arduino IDE

Mit der Einführungsaufgabe soll der Umgang mit dem Arduino MKR Vidor 4000 veranschaulicht werden. Ebenso soll der Umgang mit Quartus Prime gezeigt werden und die Übertragung einer Schaltung mittels Arduino IDE auf das FPGA.

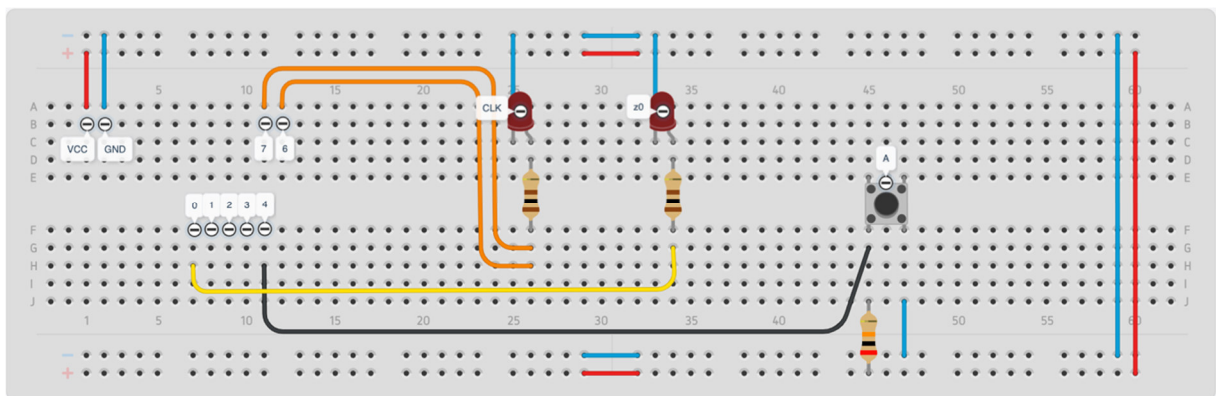
1.1. Schaltungsaufbau auf dem Steckbrett

Bauen Sie die nachfolgend dargestellte Schaltung auf dem Steckbrett auf.




Der Arduino MKR Vidor 4000 ist nicht dargestellt, die Anschlussbezeichnungen links, VCC, GND und 0 bis 7 beziehen sich aber auf dessen Anschlussnummern.



An Anschluss 7 wird vom Microcontroller ein Clock-Signal (CLK) zur Verfügung gestellt welches im Sekundentakt schaltet. Dieses ist in der Projektvorlage definiert. Die CLK wird über Anschluss 6 in das FPGA eingeschliffen.

- Welche Ausgangsspannung liefert das FPGA / der Arduino an VCC?
- Wählen Sie die LEDs (CLK, z0) aus und berechnen den Vorwiderstand.
- Achten Sie auf die Einbaurichtung der LED! (Anode & Kathode)
- Verwenden Sie einen Schließer für A und wählen einen geeigneten Pull-Up-Widerstand lt. dem Datenblatt für den Arduino, speziell für das FPGA (Intel Cyclone 10 LP)




1.2. Schaltungsentwurf in Quartus Prime

- Kopieren Sie den Zielordner zur Verfügung gestellten Ordner „05-Projektvorlage“ auf Ihren Computer und öffnen Sie unter \05_Projektvorlage\Quartus\project\MKRVIDOR4000 die Datei MKRVIDOR4000.qpf per Doppelklick.
- Bestätigen Sie das überschreiben der Datenbank mit ja wenn Sie dazu aufgefordert werden.
- Führen Sie das IP upgrade mit „Launch IP Upgrade Tool...“ und „ Perform Automatic Upgrade“ durch.
- Danach doppelklicken Sie auf „FHS_DIG_Schematic“ im Projekt Navigator im linken Bereich von Quartus Prime, daraufhin bekommen Sie ein Blockschaubild des FPGA und eine Beispielschaltung im Arbeitsbereich angezeigt.
- Unter „View“ können Sie die Zoomstufe verändern oder Sie verwenden das Zoom-Symbol über dem Arbeitsbereich , mit  können sie den Fensterinhalt verschieben.
- Am FPGA (links im Bild) selbst ist nichts zu tun.
- Für unsere Schaltungen verwenden wir die bidirektionalen digitalen Anschlüsse bMKR_D[0] bis bMKR_D[6], die Nummern in [...] beziehen sich direkt auf die Anschlussnummern am Arduino.
- Sehen Sie sich die Beispielschaltung an – was macht diese Schaltung in Zusammenhang mit dem Aufbau auf dem Steckbrett?
- Mit  können Sie Elemente der Schaltung auswählen, mit einem Rechtsklick können Sie in „Properties“ z.B. die Benennung eines Elements verändern.

- Die benötigten Logikbausteine finden sich unter , im Unterordner c:/intelfpga_lite.../primitives. Sie können diese Elemente mittels  verbinden.
- Um ein *.tff-file zu erstellen, dass Sie nach weiteren Konvertierungsschritten auf den Arduino übertragen können, speichern Sie die Lösung und klicken danach links unter „Tasks“ auf „Compile Design“ und „Start“. Wenn alles erfolgreich kompiliert wurde, sollte die letzte Nachricht im Fenster „Processing“ unten lauten: „Quartus Prime Full Compilation was successful. 0 errors, ... warnings“. Bei Fehlern doppelklicken sie auf diese im Blockdiagramm direkt dort hinzukommen.

1.3. Übertragung des Schaltungsentwurfs mittels Arduino IDE

- Schließen Sie den Arduino mittels USB-Kabel an Ihren Computer an.
- Doppelklicken Sie im Ordner unter \05_Projektvorlage auf die Datei CreateSketchFile.cmd. Damit wird das *.tff-file aus Quartus Prime so konvertiert, dass es auf den Arduino übertragen werden kann. Die Arduino IDE öffnet sich am Ende des Vorgangs automatisch.
- Wählen Sie ggf. Ihr Arduino-Board am der entsprechenden Schnittstelle unter „Select Board“ aus, danach klicken Sie auf „Upload“ .
- Nach einem erfolgreichen Upload sollte sich am Arduino das gewünschte Verhalten einstellen¹.

Hinweis: Für jede erneute Übertragung schließen Sie bitte die Arduino IDE!

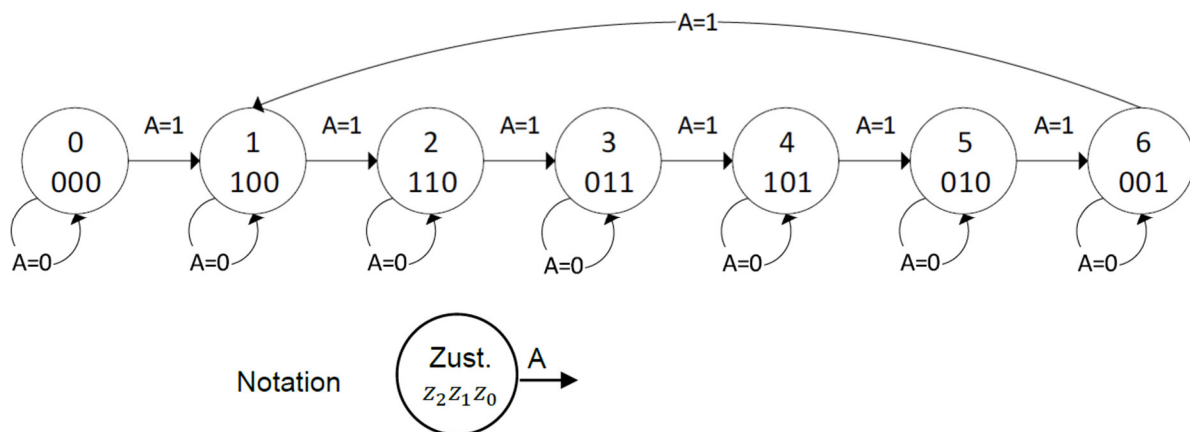
2. Grundaufgabe: Pseudo-Zufallszahlengenerator

Mit Hilfe eines Moore-Automaten soll ein Zahlengenerator entworfen werden, der die Folge

0b100, 0b110, 0b011, 0b101, 0b010, 0b001

periodisch am Ausgang liefert. Die Zahlenfolge wird gewechselt, solange ein Schalter A auf einer logische 1 geschaltet ist. Der Schalter wird nach einer zufälligen Zeitspanne auf eine logische 0 gesetzt. Man könnte sich diesen Zahlengenerator wie eine Walze eines Casino-Automaten vorstellen:

Nach reiflicher Überlegung wurde folgendes Zustandsdiagramm entworfen. Der Zustandswechsel findet nur statt, wenn der Eingang **A = 1** ist. Bei **A = 0** verbleibt der Generator im jeweiligen Zustand.



2.1. Zustandsfolgetabelle mit Ausgabetabelle

Erstellen Sie eine Zustandsfolgetabelle mit Ausgabetabelle.

Hinweis: Um sicherzustellen, dass das Schaltwerk nach dem Einschaltvorgang in den gewünschten Zyklus übergeht, kann für nicht benutzte Zustandskombinationen ein Übergang in einen definierten Zustand (z.B. den Initialzustand) definiert werden.

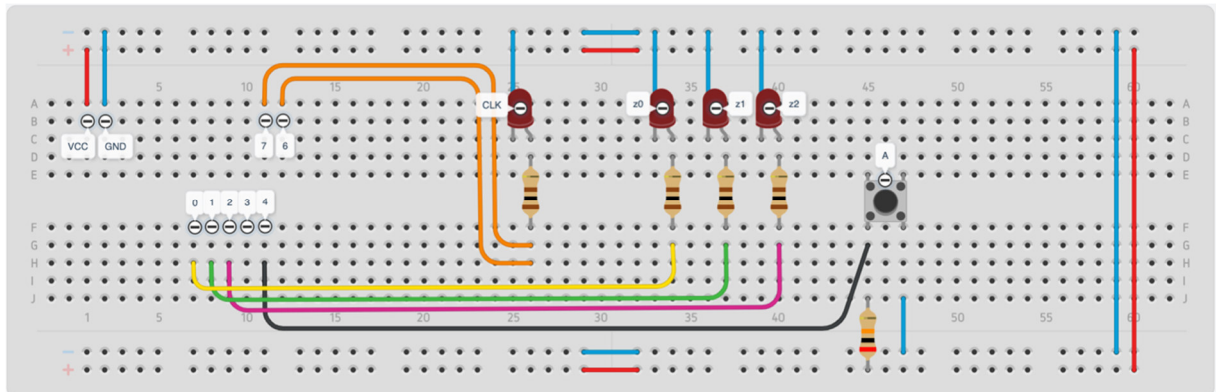
¹ LEDs CLK und z0 blinken synchron, bei Betätigung von A blinkt z0 nicht mehr

2.2. KV-Diagramme, Zustandsübergangs- und Ausgabefunktion

Ermitteln Sie die vereinfachten Zustandsübergangsfunktionen und die Ausgabefunktion mit KV-Diagrammen.


2.3. Schaltungsaufbau auf dem Steckbrett

Folgende Abbildung soll Ihnen als Hilfestellung für Ihren Aufbau auf dem Steckbrett dienen:



2.4. Schaltungsdesign in Quartus Prime

Erstellen Sie ein Schaltbild für ein entsprechendes Schaltwerk mit D-Flipflops (positive Flanken-steuerung) in Quartus Prime.

Hinweis: Die benötigten Logikbausteine finden sich unter , im Unterordner `c:/intelfpga_lite.../primitives`. Ein D-Flipflop heißt dort dff, ein 2-fach-UND and2, usw.

2.5. Übertragung auf den Arduino

Übertragen Sie Ihre Schaltung auf den Arduino und prüfen Sie diese auf Richtigkeit.

Könnte etwas falsch sein? Wie könnte sich der Fehler beheben lassen²?

3. Erweiterungsaufgabe: Rücksetzfunktionalität

Erweitern Sie die Grundaufgabe dahingehend, dass mittels eines Reset-Tasters wieder auf den Ausgangszustand **0b000** zurückgestellt werden kann.

Nachfolgend eine Hilfestellung für Ihren Schaltungsaufbau:

Hinweis: Der einfachste Weg besteht darin, die CLRN-Eingänge der D-Flipflops zu verwenden. Wird der Eingang auf GND gelegt wird das D-Flipflop zurückgesetzt.

² Fehler: die LEDs wechseln im Zyklus von CLK. Um manuell weiterschalten zu können, kann z.B. die Pull-Up-Schaltung an A gegen eine Pull-Down-Schaltung getauscht werden.

