Hochschule
Augsburg University of
Applied Sciences

Hardwaresysteme
Kürprojekt

Fakultät für
Informatik

Studienrichtung
M.Sc. Informatik

Mathias Schoppe
Entwicklung eines integrierten Schaltkreises - Konzeptionierung und Umsetzung

Betreuer: Prof. Dr.-Ing. Gundolf Kiefer
Abgabe der Arbeit am: 09.07.2023

Hochschule für angewandte
Wissenschaften Augsburg
University of Applied Sciences

An der Hochschule 1
D-86161 Augsburg

Telefon +49 821 55 86-0
Fax +49 821 55 86-3222
www.hs-augsburg.de
info@hs-augsburg.de

Fakultät für Informatik
Telefon +49 821 5586-3450
Fax +49 821 5586-3499

Verfasser der Ausarbeitung:
Mathias Schoppe
Matr. Nr.: 000000
M. Sc. Informatik
Teamkollegen: Sascha Testname
und Timo Winklbauer

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

In dieser Ausarbeitung liegt der Fokus auf der Entwicklung eines eigenen integrierten Schaltkreises (IC) mithilfe des Tiny-Tapeout-Projekts[1]. Der Prozess des IC-Designs ist äußerst komplex und erfordert spezialisierte Kenntnisse sowie teure Ausrüstung und Ressourcen. Das Tiny-Tapeout-Projekt hat es geschafft, ein besonderes Konzept zu entwickeln, das es selbst Anfängern ermöglicht, ihre eigenen IC-Designs zu realisieren. Es bietet eine kostengünstige Möglichkeit, ein eigenen IC fertigen zu lassen und eröffnet somit neue Möglichkeiten für kreative Ideen und individuelle Schaltungen. Dieser Teil der Ausarbeitung behandelt Aspekte, welche für ein solches Vorhaben zu beachten sind und bezieht sich dabei auf die konkrete Umsetzung eines Reaktionsspiels als IC.

1.2 Zielsetzung

Das Ziel der Arbeit besteht darin, ein eigenes IC-Design für ein Reaktionsspiel zu entwerfen und zu simulieren. Hierbei wird darauf geachtet, dass maximal 500 Logikgatter und ausschließlich die Ein- und Ausgänge des Tiny-Tapeout Evaluationsboards für die Umsetzung benötigt werden. Durch die Realisierung dieses Projekts sollen grundlegende Kenntnisse im IC-Design erlangt und praktische Erfahrungen in der Entwicklung von elektronischen Schaltungen gesammelt werden.

1.3 Struktur der Arbeit

Die Struktur, Kapitel und die jeweiligen Unterpunkte sind im Inhaltsverzeichnis übersichtlich dargestellt. Dieser Abschnitt fasst kurz zusammen, was in den fünf Hauptpunkten beschrieben wird:

1 **Einleitung**
Die Einleitung verschafft einen Überblick über das zu behandelnde Thema und stellt die Motivation für die Verfassung der Arbeit dar.

2 **Grundlagen**
Das Kapitel Grundlagen legt die verwendeten Theorien und Konzepte dar, welche für die Umsetzung des Projektes unabdingbar sind.

2 Grundlagen

Das nachfolgende Kapitel befasst sich mit grundlegenden Themen, welche für die Umsetzung der Arbeit unabdingbar sind. Die Themenbereiche werden zusammengefasst und so erläutert, dass auch Lesende ohne Vorerfahrung im Bereich der Hardwareentwicklung alle Aspekte der Arbeit nachvollziehen können. Es wird ausschließlich auf Grundlagen, welche relevant für die konkrete Umsetzung des Projekts sind, eingegangen.

2.1 Linear rückgekoppeltes Schieberegister

Ein linear rückgekoppeltes Schieberegister (engl. linear feedback shift register (LFSR)) ist eine Schaltung, die aus einer Reihe von Flip-Flops (digitale Speicherelemente) besteht, die in einer Kette miteinander verbunden sind. Die Rückkopplung erfolgt, indem das Ausgangssignal eines Flip-Flops mit dem Eingang eines vorherigen Flip-Flops verbunden wird. Dadurch bildet sich ein geschlossener Rückkopplungs-Pfad.[2]
Die Anzahl der Flip-Flops in einem Schieberegister bestimmt die Anzahl der Speicherplätze oder Zustände, die es halten kann. Ein 4-Bit-Schieberegister kann beispielsweise 16 verschiedene Zustände einnehmen, während ein 8-Bit-Schieberegister 256 verschiedene Zustände einnehmen kann.

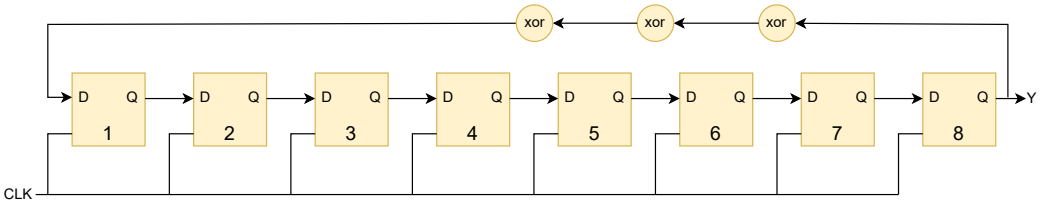

Abbildung 1: Abbildung eines Fibonacci-LFSR, angelehnt an [2]

In Abbildung 1 ist ein 8-Bit Fibonacci-LFSR dargestellt. Dabei repräsentiert CLK den Takteingang und Y den Ausgang des LFSR. Bei jedem Taktimpuls wird der gespeicherte Wert der Flip-Flops eine Stelle weiter geschoben. Bei jedem Taktimpuls wird das Rückkopplungs-Bit in das erste Flip-Flop des Schieberegisters eingespeist, alle anderen Bits werden zum nächsten Flip-Flop geschoben. Dies bedeutet, dass das Ausgangssignal des ersten Flip-Flops als Eingangssignal des zweiten Flip-Flops dient, das Ausgangssignal des zweiten Flip-Flops als Eingangssignal des dritten Flip-Flops und so weiter.

Das Fibonacci-LFSR zeichnet sich durch das in Gleichung 1 dargestellte primitive Generatorpolynom aus. Dies ermöglicht dem Schieberegister die Generierung von Pseudozufallszahlen.

$$pf(x) = x^8 + x^6 + x^5 + x^4 + 1 \tag{1}$$

2.2 Hardware Timer

Der Hardware Timer ist eine Schaltungskomponente, die verwendet wird, um Zeit zu messen oder Zeit basierte Ereignisse zu erzeugen. Er besteht aus einem internen Zähler, der bei jedem Taktimpuls inkrementiert oder dekrementiert wird und so die Grundlage für die Zeiterfassung bildet. Der Hardware Timer wird durch Steuersignale gesteuert, welche das starten, stoppen und zurücksetzen ermöglichen. Diese Steuersignale können entweder von einer externen Steuerung oder durch interne Logik generiert werden.
Damit durch den Hardware Timer eine bestimmte Zeit in Sekunden überbrückt wird, betrachten wir folgende Zusammenhänge. Die Taktfrequenz f gibt an, wie viele Taktzyklen pro Sekunde auftreten. Um eine bestimmte Zeit t in Sekunden zu warten, muss die Anzahl der Takte berechnet werden, die während dieser Zeit vergehen. Dazu wird in Gleichung 2 die Zeit t mit der Taktfrequenz f multipliziert.

$$\text{Anzahl der Takte} = t * f \tag{2}$$

Es ist wichtig zu beachten, dass diese Berechnung auf der idealisierten Annahme basiert, dass die Taktfrequenz und der Takt stabil und präzise sind. In der Praxis können jedoch Ungenauigkeiten auftreten, die die Zeitmessung beeinflussen.

3 Schaltkreis Entwurf

In diesem Kapitel wird der IC Entwurf unseres Reaktionsspiels entstehen. Der übliche Designprozess[**timoICDesign**] für ICs wird in Schritten abgearbeitet. Durch die Implementierung und Simulation des Designs werden wir die Funktionalität des Reaktionsspiels überprüfen und abschließend ein Layout erstellen, das für die Fertigung des ICs im Rahmen des Tiny-Tapeout-Projekts verwendet werden kann.

3.1 Das Reaktionsspiel

Um den Schaltkreis genau definieren zu können, muss vorab die exakte Funktionsweise des Reaktionsspiels definiert werden. Das Spiel wird ausgelegt für zwei Spieler. Jeder Spieler hat zwei Knöpfe als Eingabemöglichkeit und es gibt einen zusätzlichen Startknopf. Die insgesamt fünf Knöpfe werden als Eingangssignale in den von uns entwickelten IC übermittelt. Als Interface für die Kommunikation mit dem Benutzer wird die auf dem Evaluationsboard von Tiny-Tapeout verbaute 7-Segment-Anzeige verwendet.[**timoEvalboard**] Dadurch ergibt sich das in Abbildung 2 dargestellte Design für das Reaktionsspiel.

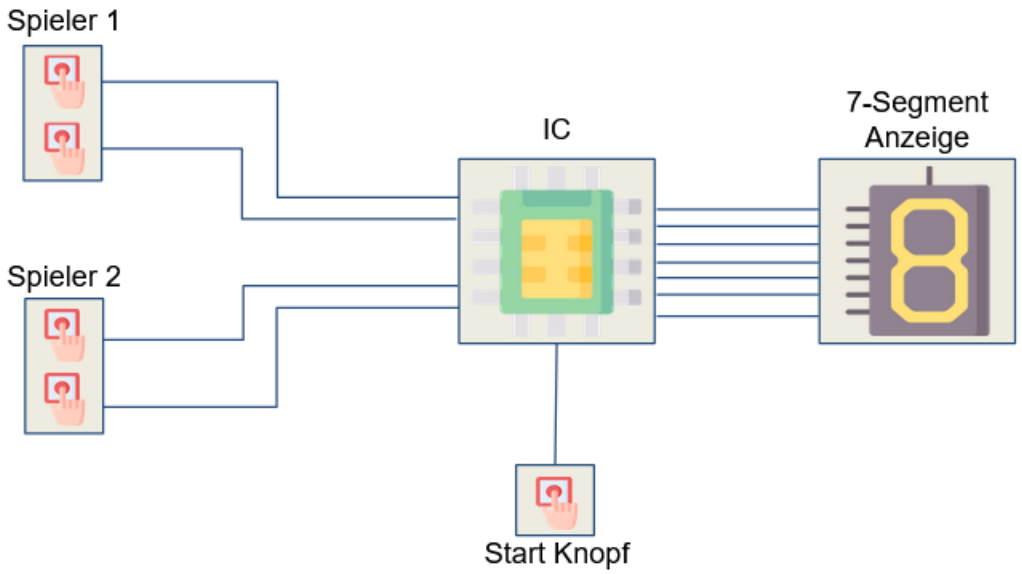

Abbildung 2: Design des Reaktionsspiels

Die Leuchtdioden der 7-Segment-Anzeige werden für die Visualisierung drei verschiedener Zustände verwendet, wie in 1 dargestellt.

Zustand	Beschreibung
Blinken	Wenn ein Spiel gestartet wird, blinken alle Segmente der Anzeige innerhalb von drei Sekunden drei mal auf
Zufälliges Aufleuchten	Nachdem eine pseudo zufällige Wartezeit verstrichen ist, leuchtet entweder das linke (bzw. obere) oder rechte (bzw. untere) Segment auf.
Gewinner Anzeigen	Je nachdem, ob ein Spieler zu früh, den richtigen Knopf oder den falschen Knopf drückt wird der Sieger ermittelt. Gewinnt Spieler 1, dann leuchtet die linke (bzw. obere) Hälfte der 7-Segment-Anzeige auf. Gewinnt Spieler 2, dann leuchtet die rechte (bzw. untere) Hälfte der 7-Segment-Anzeige auf.

Tabelle 1: Erklärung der Zustände der 7-Segment-Anzeige

Anhand der in Tabelle 1 definierten Zustände, lässt sich der Spielablauf für unser Reaktionsspiel ableiten. Das Spiel durchläuft die folgenden Schritte:

1. Warten bis der Start-Knopf betätigt wird
2. Sobald das Spiel Startet, blinkt die 7-Segment-Anzeige drei mal auf
3. Wenn das blinken beendet ist, warte eine (pseudo-)zufällige Zeit
4. Nach Ablauf der Zufallszeit, leuchtet das linke oder rechte Segment auf
5. Warte auf Knopfdruck der Spieler
6. Sieger wird ermittelt und die entsprechenden Segmente in der 7-Segment-Anzeige leuchten auf

3.2 Register-Transfer-Ebene

Nachdem die grundlegenden Funktionen des ICs im vorherigen Kapitel definiert wurden, kann der Entwurf auf Register-Transfer-Ebene beginnen. Der Prozess des Entwurfs durchläuft mehrere Schritte. Zuerst wird der Algorithmus anhand einer einfachen Beschreibungssprache definiert. Danach werden die Zustände des Algorithmus in einem Zustandsübergangsdiagramm modelliert, sodass die Überführung in eine Hardwarebeschreibungssprache (hier **V**ery High Speed Integrated Circuits **H**ardware **D**escription **L**anguage (VHDL)) im nächsten Schritt vereinfacht wird. Zuletzt wird das Programm synthetisiert und simuliert. Auf den Entwurf des Algorithmus wird in dieser Stelle nicht tiefer eingegangen, da die Funktionsweise des Schaltkreises in den Schritten des Zustandsübergangsdiagramms und der Überführung in VHDL klar wird.

3.2.1 Zufallsgenerierung

Für die Zufallsgenerierung wird, wie im Kapitel 2.1 eingeführt, ein 8-bit linear rückgekoppeltes Schieberegister verwendet. Als Bereich für die zufällige Wartezeit haben wir eine Zeit zwischen 0,5 und 5 Sekunden definiert. Nach Formel 2 kann abhängig von der Taktfrequenz des Evaluationsboards die Anzahl der Takte ausgerechnet werden, welche mindestens bzw. maximal gewartet werden dürfen um in diesem Bereich zu bleiben. Wie in Kapitel **??** eingeführt, läuft das Evaluationsboard mit einer Taktfrequenz von 25 Kilohertz. Daher ergibt sich für die zufällige Wartezeit von 0,5 Sekunden folgende Taktanzahl:

$$cnt_{min} = 25.000 * 0.5 = 12.500$$

Die maximale Anzahl der Takte lässt sich wie folgt berechnen:

$$cnt_{max} = 25.000 * 5 = 125.000$$

Für den Zähler werden also 17-Bit benötigt, um den Wertebereich bis 125.000 darstellen zu können. Damit mindestens eine halbe Sekunde Wartezeit entsteht, wird der Zähler mit der Bitfolge $00011000100000000 (12.544)$ initialisiert, was in etwa der halben Sekunde entspricht. Für die zufällige Wartezeit werden die drei niedrigsten Bits des LFSR für die drei höherwertigsten bits des Zählers verwendet, wie in Abbildung 3 dargestellt ist.

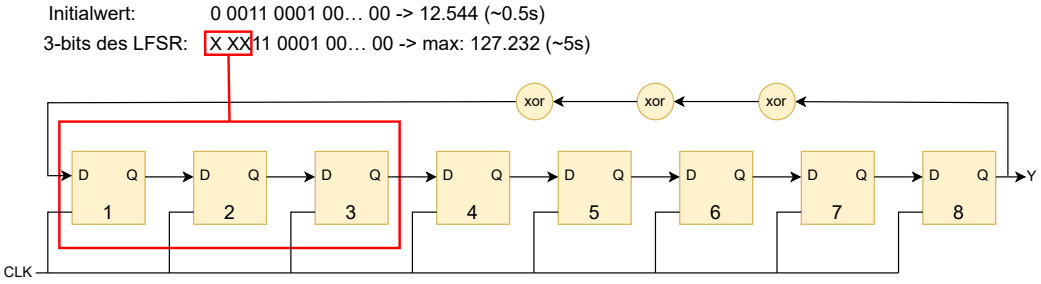

Abbildung 3: Wertevergabe des Zählers für die zufällige Wartezeit

Somit ergeben sich die in Tabelle 2 dargestellten möglichen Kombinationen für die Wartezeit.

Bitfolge	Wartezeit in Sekunden
0 0011 0001 00... 0	0,50 Sekunden
0 0111 0001 00... 0	1,16 Sekunden
0 1011 0001 00... 0	1,81 Sekunden
0 1111 0001 00... 0	2,47 Sekunden
1 0011 0001 00... 0	3,12 Sekunden
1 0111 0001 00... 0	3,78 Sekunden
1 1011 0001 00... 0	4,43 Sekunden
1 1111 0001 00... 0	5,09 Sekunden

Tabelle 2: Alle möglichen Bitfolgen des Zählers für die zufällige Wartezeit

3.2.2 Zustandsübergrangsdiagramm

In Abbildung 4 ist das aus dem Algorithmus überführte Zustandsübergangsdiagramm dargestellt. Das Diagramm modelliert die folgenden Zustände:

- **State_Reset:** In diesem Startzustand wird auf das drücken des Start-Knopfs gewartet. Sobald dieser gedrückt wurde, wird der Zustand verlassen
- **State_blink3Times:** Der Zustand fasst den Ablauf des dreifachen aufblinken der Segmentanzeige zusammen. Ist das Blinken beendet, wird die zufällige Wartezeit initialisiert
- **State_waitRndInit:** Setzt den Zähler des Timers auf einen zufälligen Wert, sodass nach Formel 2 eine Zeit zwischen 0.5-5 Sekunden gewartet wird
- **State_waitRnd:** Dekrementiert den Zähler bei jedem Clock-Zyklus. Währenddessen wird geprüft, ob ein Spieler einen seiner Knöpfe zu früh betätigt. Falls ja, wird direkt der Gewinner angezeigt. Sonst leuchtet nach Ablauf der Zeit entweder das linke oder rechte Segment auf
- **State_setRightLED:** Das rechte Segment leuchtet und es wird auf Knopfdruck von einem der Spieler gewartet, sodass der Gewinner ermittelt werden kann
- **State_setLeftLED:** Das link Segment leuchtet und es wird auf Knopfdruck von einem der Spieler gewartet, sodass der Gewinner ermittelt werden kann
- **State_p1Won:** Spieler 1 hat das Spiel gewonnen, das bedeutet die linke Hälfte der 7-Segment-Anzeige leuchtet auf. Anschließend wird auf ein erneutes drücken des Start-Knopfes gewartet
- **State_p2Won:** Spieler 2 hat das Spiel gewonnen, das bedeutet die rechte Hälfte der 7-Segment-Anzeige leuchtet auf. Anschließend wird auf ein erneutes drücken des Start-Knopfes gewartet

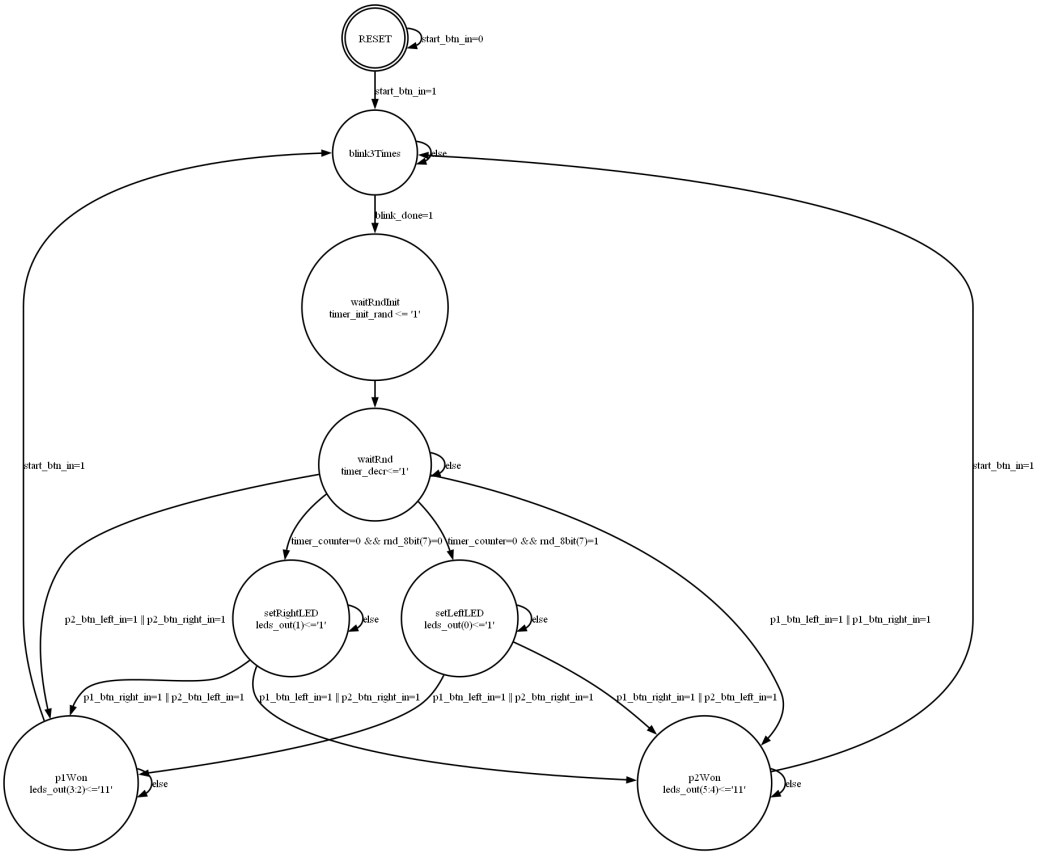

Abbildung 4: Zustandsübergangsdiagramm des ICs

IC integrierter Schaltkreis

LFSR linear rückgekoppeltes Schieberegister

VHDL **V**ery High Speed Integrated Circuits **H**ardware **D**escription **L**anguage

7 Literaturverzeichnis

Literatur

[1] Tiny Tapeout. *Tiny Tapeout Homepage*. online verfÃ¼gbar unter `https://tinytapeout.com`; zuletzt aufgerufen am 02.07.2023. 2023.

[2] Dirk Fox. „Linear Rueckgekoppelte Schieberegister“. In: *Datenschutz und Datensicherheit-DuD* 32.5 (2008), S. 351–351.