

Inhaltsverzeichnis

1. Protokoll	2
1.1. Aufgabenstellung.....	2
1.2. Einrichtungen und Geräte	2
1.3. Versuchsvorbereitung.....	3
1.3.1. Schaltungen zur Generierung und Prüfung von CRC-Werten.....	3
1.3.2. Modifikation für ein CRC-16-Polynom.....	4
1.3.3. Pseudo-Zufallsgenerator	4
1.4. Versuchsdurchführung	5
1.4.1. Simulation der CRC-Generierung und -Prüfung.....	5
1.4.2. Pseudo-Zufallsgenerator	7
2. Fazit	8
3. Anhang.....	9
4. Bildquellen:.....	9

1. Protokoll

1.1. Aufgabenstellung

In diesem Versuch wird die Kanalcodierung von Nutzinformationen mittels zyklischer Codes untersucht. Dabei wird jeweils eine Schieberegisterschaltung für die Generierung und die Prüfung eingesetzt, die einerseits eine Polynom-Multiplikation und bei der Prüfung eine Polynom-Division durchführen.

Im zweiten Teil wird das linear Rückgekoppelte Schieberegister von der Prüfung als Pseudo-Zufallsgenerator verwendet und dessen Periodizität gemessen.

Alle Schaltungen werden dabei auf einem CPLD realisiert.

1.2. Einrichtungen und Geräte

Für den Versuch standen folgende Geräte und Einrichtungen zur Verfügung:

- PC mit Xilinx ISE Synthese Tool
- ModSys-System mit:
 - XILINX Coolrunner CPLD XC2C256-PQ208-7
 - 2 Stk. ModSys IOM Board

1.3. Versuchsvorbereitung

1.3.1. Schaltungen zur Generierung und Prüfung von CRC-Werten

Es wurde je eine Schaltung zur Generierung des zyklischen Codes und eine zur Prüfung erstellt. Das verwendete Generator-Polynom lautet: $x^6 + x^5 + x^3 + x^2 + 1$.

CRC-Generation:

Die Schaltung (Abb.1) besteht aus einem rückgekoppelten Schieberegister, das eine Polynom-Multiplikation mit dem Generator-Polynom vornimmt. Der VHDL-Code ist in Anhang 1 zu finden.

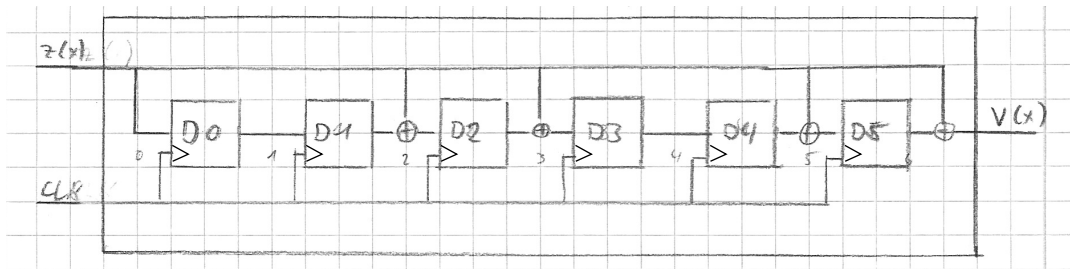


Abb.1: Blockschaltbild Generator

CRC-Prüfung:

Die Schaltung (Abb.2) besteht aus einem linear rückgekoppelten Schieberegister, das eine Polynom-Division durch das Generator-Polynom vornimmt. Der VHDL-Code ist in Anhang 2 zu finden.

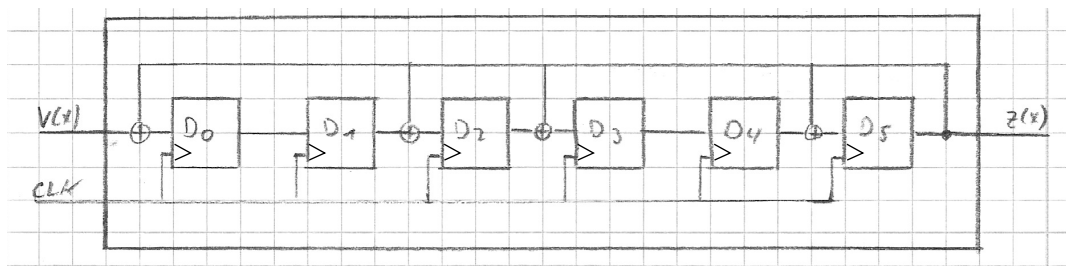


Abb.2: Blockschaltbild Prüfung

Testmuster/ Fehlermuster:

Zur Verifikation der Schaltungen wurden folgende Testmuster verwendet:

- Aus der Nutzinformation ergibt sich durch Polynom-Multiplikation der CRC-Wert. Wenn die folgenden Nutzinformation seriell (MSB-first, taktsynchron) in den CRC-Generator eingegeben wird, soll der folgende CRC-Wert seriell (MSB-first, taktsynchron) ausgegeben werden.

Nutzinformation:	$x^7 + x^5 + x^3 + x^2 + 1$	1010.1101
CRC:	$x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^4 + 1$	11.1001.1001.0001

- Die CRC-Prüfung wird getestet, indem das folgende CRC-Polynom seriell (MSB-first, taktsynchron) eingegeben wird. Dann soll die folgende Nutzinformation seriell (MSB-first, taktsynchron) ausgegeben werden und das Schieberegister mit Nullen belegt sein.

CRC:	$x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^4 + 1$	11.1001.1001.0001
Nutzinformation:	$x^7 + x^5 + x^3 + x^2 + 1$	1010.1101

- Für das Testen einer fehlerhaften Übertragung wurde das folgende CRC-Polynom gewählt. Die Fehler sind fett markiert. Diesmal wird erwartet, dass ein Rest im Schieberegister verbleibt.

CRC:	$x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^4 + x^2 + x^1 + 1$	11. 01 01.1001. 0111
------	--	------------------------------------

1.3.2. Modifikation für ein CRC-16-Polynom

Um ein CRC-16-Polynom zu benutzen, müsste der VHDL-Code um weitere FlipFlops und XOR-Gatter ergänzt werden, sodass der Code dem entsprechenden Blockschaltbild entspricht. Außerdem müssten natürlich die internen Signale, die für die Ein- und Ausgänge der FlipFlops verwendet werden, auf eine Länge von 16 Bit erweitert werden.

Diese Schaltung könnte dann Bündelfehler von maximal 16 Bit erkennen.

1.3.3. Pseudo-Zufallsgenerator

Der Pseudo-Zufallsgenerator besteht erst einmal aus der gleichen Schaltung, wie die Schaltung zur CRC-Prüfung. Diese Mal werden jedoch die Inhalte (Eingänge) der Schieberegister benutzt, indem diese eine parallel gespeicherte binäre Zahl darstellen. Der Ausgang der Schieberegister-Schaltung wird nicht mehr benötigt.

Zu Beginn wird ein Mal eine 1 an den Eingang angelegt, das die Schaltung anregt. Danach wird mit jedem Takt eine neue Zahl quasi zufällig erzeugt, wobei sich die Zahlen nach einer gewissen Periodendauer wiederholen.

Die Periodizität wird mit Hilfe eines binären Zählers ermittelt. Dieser zählt mit jedem Takt eine Zahl weiter hoch und wird beim Auftreten einer bestimmten Zahl im Schieberegister zurückgesetzt. Vor jedem Zurücksetzen wird jedoch der Inhalt des Zählers auf einem Ausgang ausgegeben, sodass immer der maximale Wert des Zählers ausgegeben wird.

Die Schaltung (Abb.3) besteht aus einem linear Rückgekoppelten Schieberegister und einem binären Zähler von 7 Bit Breite. Außerdem gibt es ein Modul, dass das Schieberegister ausliest und bei einem gegebenen Wert den Zähler zurücksetzt, sowie den Zählerstand zu diesem Zeitpunkt speichert und ausgibt. Der VHDL-Code ist in Anhang 3 zu finden.

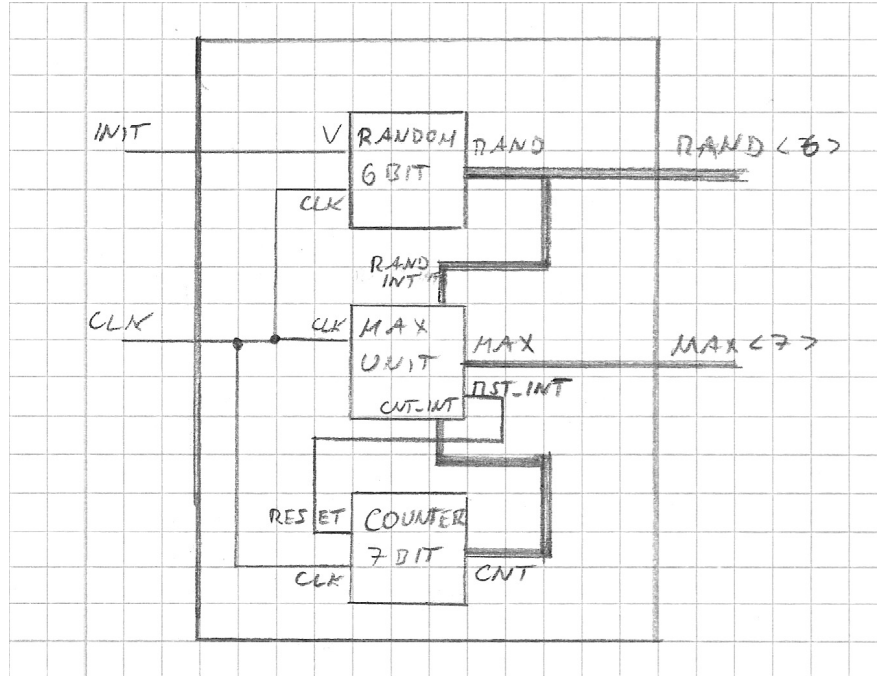


Abb.3: Blockschaltbild Pseudo-Zufall

Die maximale Periodizität beträgt $2^6 - 1 = 63$ (oder auch 0x3F)

1.4. Versuchsdurchführung

1.4.1. Simulation der CRC-Generierung und -Prüfung

Es wurden die Testmuster aus 1.3.1 angewendet.

Test A:

Die Schaltung wurde mit der Nutzinformation über den Eingang Z gespeist, der korrekte CRC-Wert wurde über den Ausgang V ausgegeben, wie in Abb.4 zu sehen ist. Test erfolgreich.

Nutzinformation: $x^7 + x^5 + x^3 + x^2 + 1$ 1010.1101
 CRC: $x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^4 + 1$ 11.1001.1001.0001

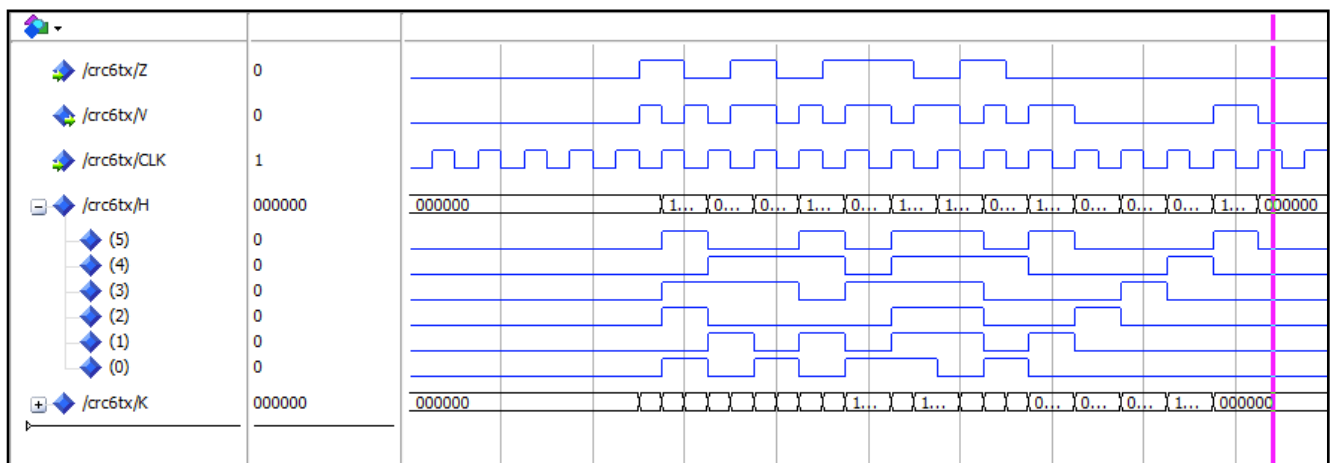


Abb.4: Simulation der CRC-Generierung

Test B:

Die Schaltung wurde mit dem CRC-Wert über den Eingang V gespeist, die korrekte Nutzinformation wurde über den Ausgang ZOUT ausgegeben, wie in Abb.5 zu sehen ist. Test erfolgreich.

CRC: $x^{13} + x^{12} + x^{11} + x^8 + x^7 + x^4 + 1$ 11.1001.1001.0001
 Nutzinformation: $x^7 + x^5 + x^3 + x^2 + 1$ 1010.1101

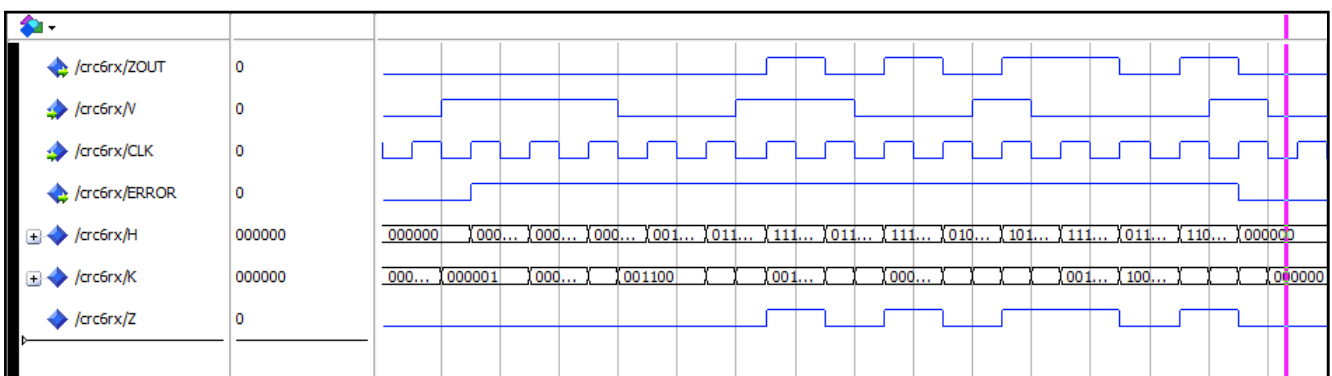


Abb.5: Simulation der CRC-Prüfung mit korrektem CRC-Wert

Test C:

Die Schaltung wurde mit dem CRC-Wert über den Eingang V gespeist, im Schieberegister (Signal H) verbleibt ein Rest, wie in Abb.6 zu sehen ist. Test erfolgreich.

CRC: $x^{13}+x^{12}+x^{10}+x^8+x^7+x^4+x^2+x^1+1$ 11.0101.1001.0111

Rest: 01.0010

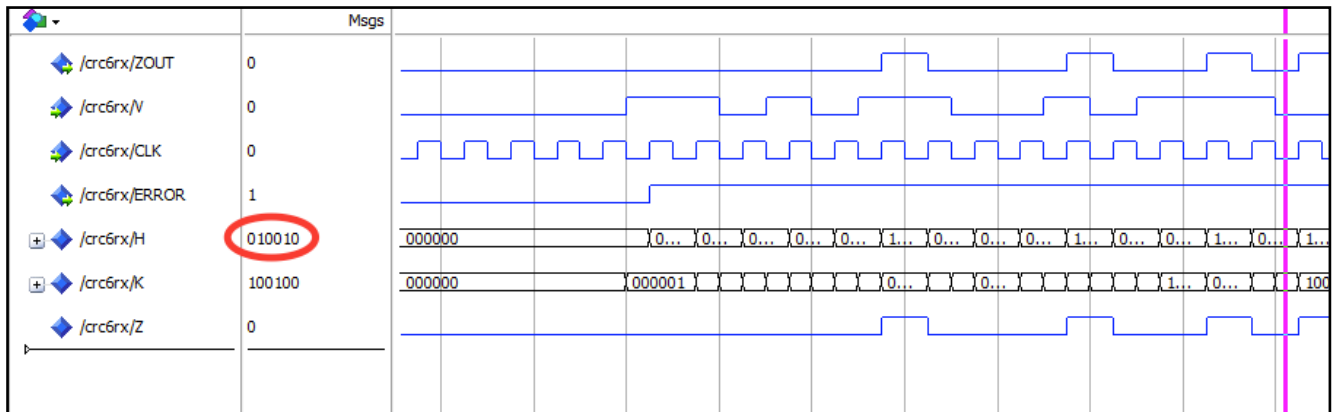


Abb.6: Simulation der CRC-Prüfung mit fehlerhaftem CRC-Wert

1.4.2. Pseudo-Zufallsgenerator

Simulation:

Der Pseudo-Zufallsgenerator wurde mit modelsim simuliert. Der Counter zählt jeden Takt eins hoch und wird vom Reset auf 1 zurückgesetzt.

Die Periodizität beträgt 63 (0x3F), wie in Abb.7 zu sehen ist. Der Pseudo-Zufallsgenerator funktioniert in der Simulation also wie vorgesehen.

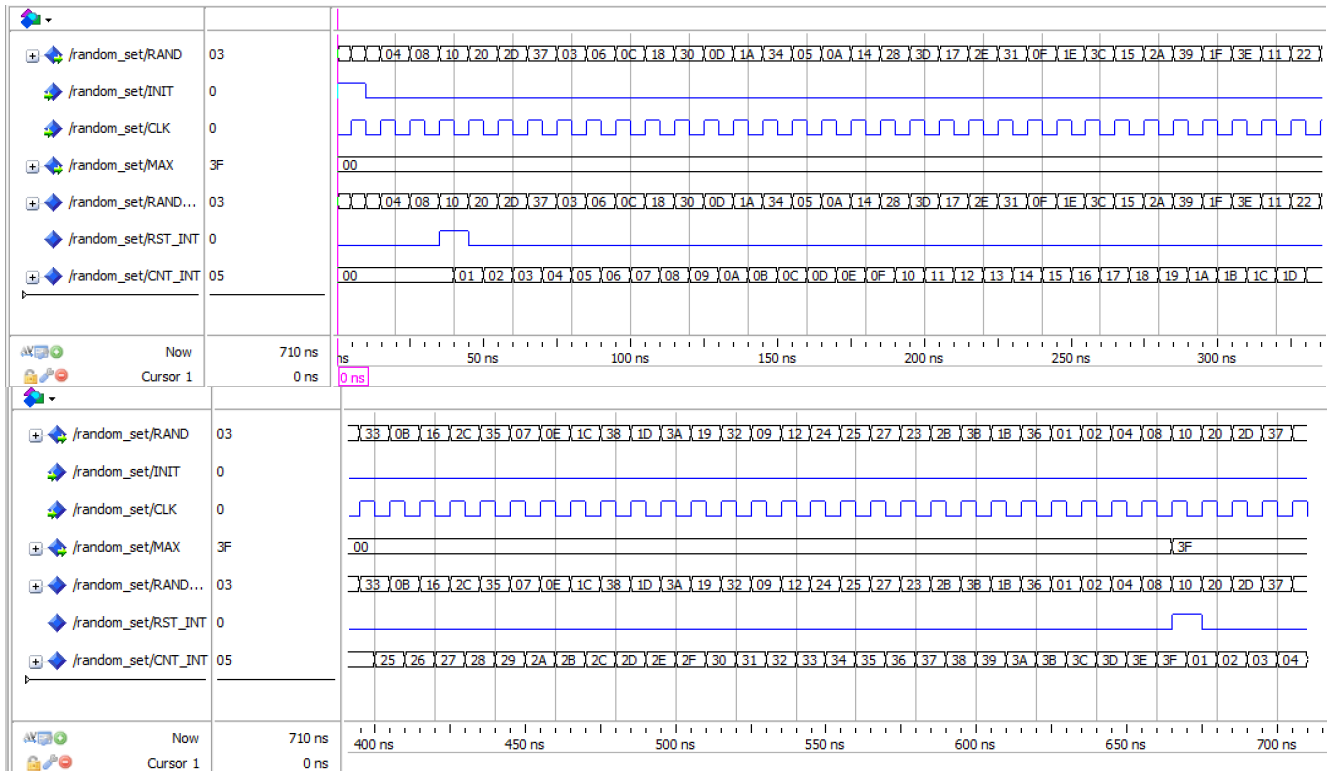


Abb.7: Simulation des Pseudo-Zufallsgenerators

Implementierung auf CPLD:

Der verwendete CPLD wurde auf über das Programm ISE programmiert, indem dieses den VHDL-Code synthetisiert hat und die Logik entsprechend auf den CPLD übertragen hat.

Für die Zuweisung der Anschlüsse des CPLD wurden entsprechende UCF-Dateien (siehe Anlagen) verwendet.

Auf der Versuchsplattform ModSys wurden, nach einem initialen Löschen des CPLD, je ein IOM-Board an die Erweiterungsanschlüsse 3 und 4 angesteckt, um die Zufallswerte, sowie den Maximalwert des Zählers auszugeben. Außerdem wurden die prellfreien Taster dafür genutzt den Eingang der Schaltung zu initialisieren, sowie den Takt einzugeben.

Auch der auf der Hardware implementierte Pseudo-Zufallsgenerator zeigte das gewünschte Verhalten. Die maximale Periodizität wurde auch hier als 63 ausgegeben.

2. Fazit

CRC-Generator und CRC-Prüfung:

Die simulierten Schaltungen für die CRC-Generierung und -Prüfung haben die Tests mit je einer zulässigen und einem unzulässigen Information erfolgreich bestanden.

Der vom Generator aus der Nutzinformation erzeugte Wert wird von der Schaltung zur CRC-Prüfung wieder in die ursprüngliche Nutzinformation übersetzt.

Wenn auf der Übertragungsstrecke Fehler auftreten und somit ein unzulässiges Codewort entsteht, verbleibt im Register ein Wert ungleich 0, was bedeutet, dass ein Fehler bei der Übertragung aufgetreten ist. Dieser Fehler wird erkannt, aber in dieser Schaltung nicht behoben.

Pseudo-Zufallsgenerator

Der Pseudo-Zufallsgenerator hat eine maximale Periodizität von 63, welche auch der theoretisch erreichbaren Periodizität entspricht. Dies bedeutet, dass das verwendete Generatorpolynom irreduzibel ist.

3. Anhang

- Anlage 1: CRC-Generation: VHDL-Code
- Anlage 2: CRC-Prüfung: VHDL-Code
- Anlage 3: Pseudo-Zufallsgenerator: VHDL-Code
- Anlage 4: Pseudo-Zufallsgenerator: UCF-Datei

4. Bildquellen:

- Abb.1: Blockschaltbild Generator
Jannik Beyerstedt
- Abb.2: Blockschaltbild Prüfung
Jannik Beyerstedt
- Abb.3: Blockschaltbild Pseudo-Zufall
Jannik Beyerstedt
- Abb.4: Simulation der CRC-Generierung
Jannik Beyerstedt
- Abb.5: Simulation der CRC-Prüfung mit korrektem CRC-Wert
Jannik Beyerstedt
- Abb.6: Simulation der CRC-Prüfung mit fehlerhaftem CRC-Wert
Jannik Beyerstedt
- Abb.7: Simulation des Pseudo-Zufallsgenerators
Jannik Beyerstedt