

LABOR VHDL

Projekt 2: EierUhr

Author:
Steve WAGNER

Supervisor:
Prof. Elke MACKENSEN

8. Dezember 2015

Inhaltsverzeichnis

1	Einführung	1
2	Architektur	2
3	Zustandsdiagramme	4
3.1	Vorwärtzähler / Counter	4
3.1.1	Sekundeneinheit	4
3.1.2	Sekundenzehner	5
3.1.3	Minuteneinheit	6
3.1.4	Minutenzehner	7
3.2	Rückwärtzähler / Decounter	8
3.2.1	Sekundeneinheit	8
3.2.2	Sekundenzehner	9
3.2.3	Minuteneinheit	10
3.2.4	Minutenzehner	11
4	Simulation mit ModelSim	12
4.1	Debounce (für Drucktaste)	12
4.2	Frequenzteiler	13
4.3	Vorwärtzähler (Counter)	14
4.4	Rückwärtzähler (Decounter)	15
4.5	Audio / Ende der Zeit	16
4.6	Display	17
4.7	Ganze Architektur (Timer)	18
5	Konklusion	20

1 Einführung

Im Rahmen dieser Projektaufgabe soll ein Kurzzeitwecker (umgangssprachlich Eieruhr) entwickelt werden, deren Einsatz z.B. beim Kochen Verwendung finden kann. Es handelt sich hierbei um einen Timer/Zeitmesser der kurze Zeiteinheiten, z.B. 60 Minuten, misst und durch ein akustisches, optisches oder mechanisches Signal anzeigt.

Die Spezifikationen werden je nach der Test und Simulation in dem 3. Teil beschreiben.

2 Architektur

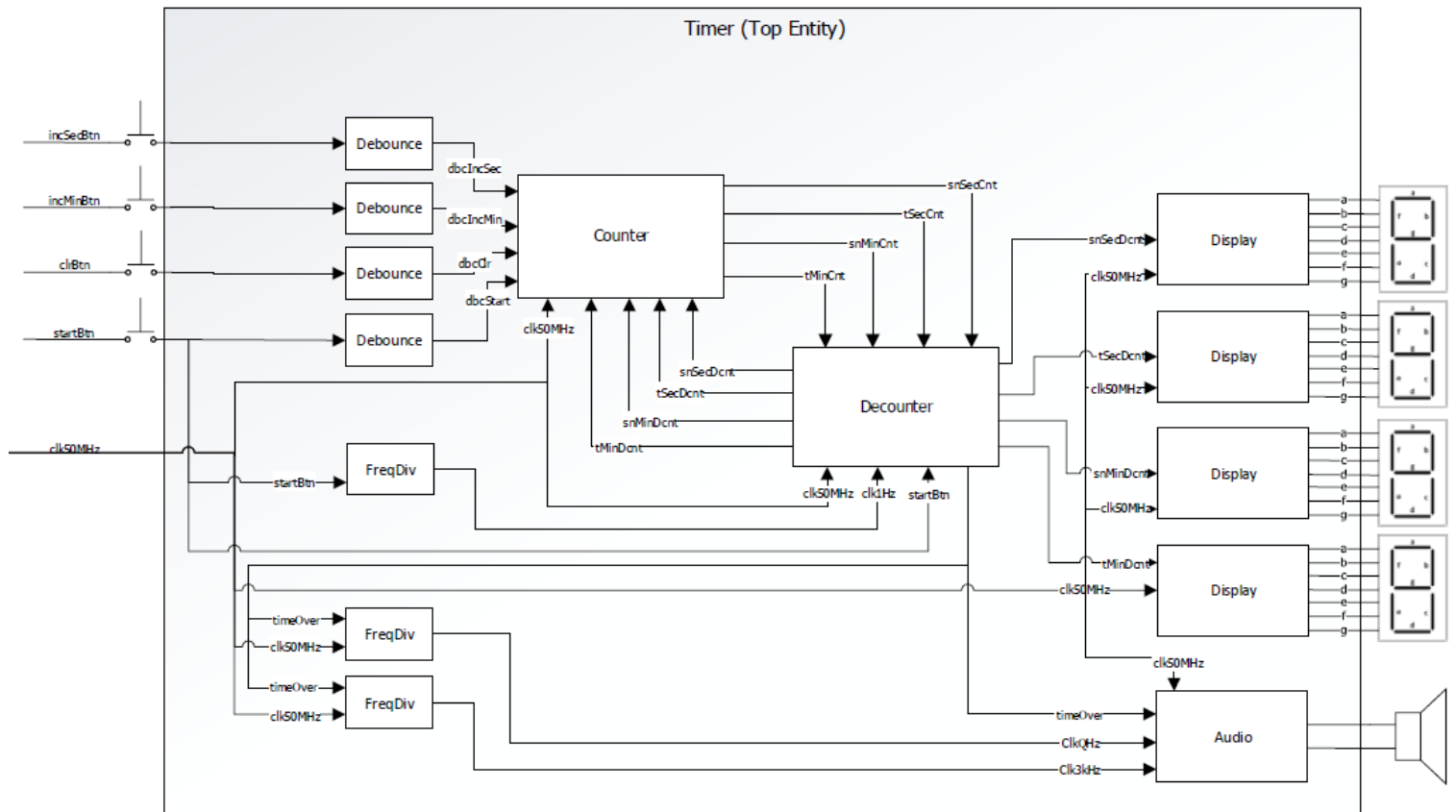


Abbildung 1: Hauptarchitektur des Projekts

Debounce

Der Debounce ist benutzt, um den Rückfall des Knopfs zu warten. Das erlaubt, um für jede Betätige nur einen Wert hochzuzahlen.

FreqDiv

Der Frequenzteiler ist benutzt, um einen spezifischen Frequenz zu erreichen, wie zb. den 1-Hz Frequenz, die für den Decounter benutzt ist.

Counter

Der Counter Teil erlaubt, um einen Wert nach jeder Betätige der Sekunde- oder Minute-Taste hochzuzählen. Diesen Werte können wieder, nach einer Betätige auf der Clear-Taste, zu Null anfangen.

Die Werte sind dann in der Decounter gesendet.

Es funktioniert nur wenn man in Stop mode ist (default mode am Anfang)

Decounter

Der Decounter Teil erlaubt, um die Werte von Counter jede Sekunde runterzählen. Diesen Werte sind in Display gesendet.

Wenn man den Null erreicht, ist ein timeOver-Signal zu dem Audio Teil gesendet.

Dieser Teil funktioniert nur wenn man in Start mode ist (nach eine Betätige auf der Start-Taste)

Display

Der Display stellt die Werte von Decounter dar.

Audio

Der Audio Teil erlaubt, um die Signal-Töte zu erreichen.

Es fängt in dem Moment der timeOver-Signal zu 1 ist an. Dann wird eine Signal-Töte jede 250ms (clkQHz), mit dem Frequenz des Signals, um etwas zu hören (clk3KHz), auf der dem Signalgeber senden. Diese Signal-Töte wird wieder nach einer Ruhe von 1 Sekunde senden. Dies während eine Minute oder bis man auf der Start-Taste betätigt, um wieder in Stop mode zu gehen.

3 Zustandsdiagramme

3.1 Vorwärtszähler / Counter

3.1.1 Sekundeneinheit

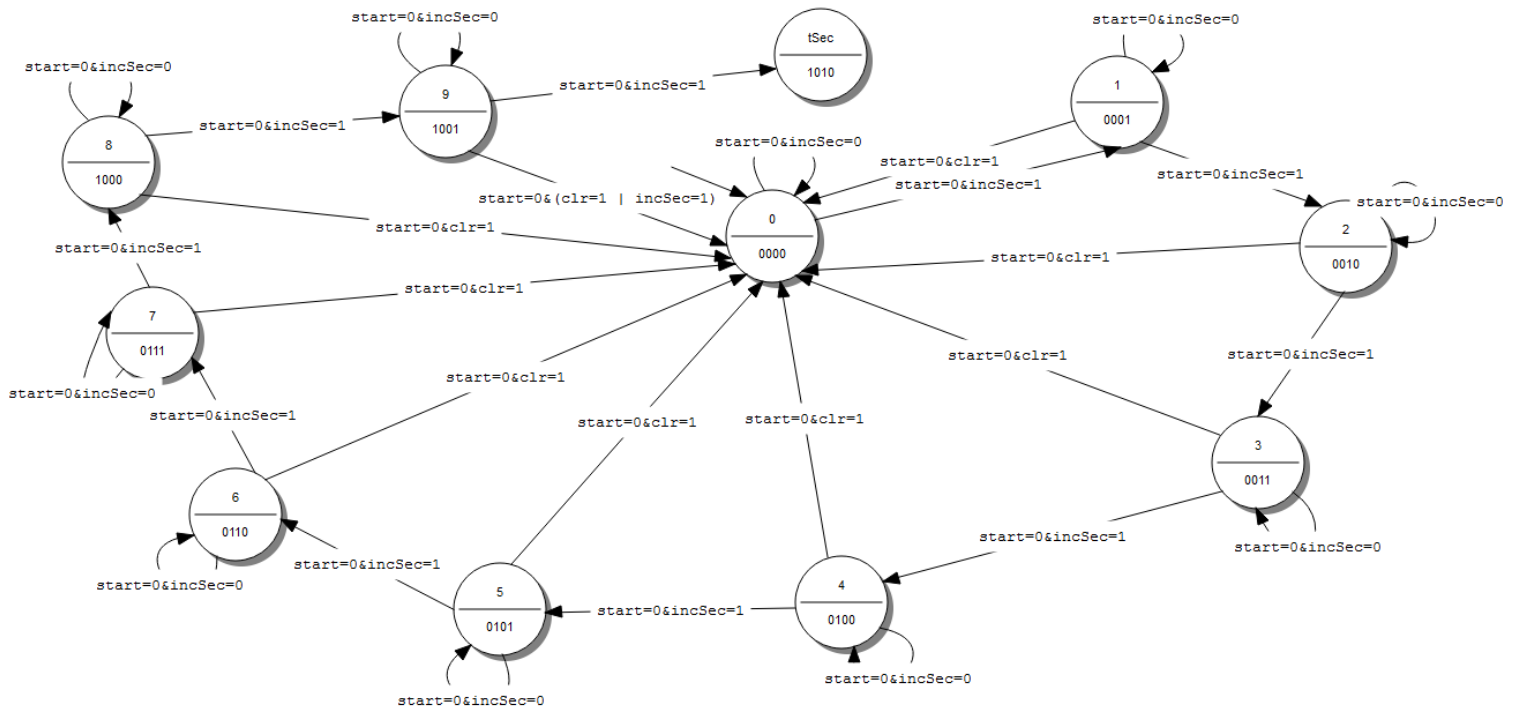


Abbildung 2: Zustandsdiagramm des Sekundeneinheit-Vorwärtszählers

Der Sekundeneinheit-Vorwärtszähler zählt nur wenn man in Stop mode ist (start=0) hoch und wenn man auf die Sekunde-Taste betätigt.

Wenn die 9 Wert erreichen ist, wird dann in der nächste Sekunde-Taste-Betätige, die Sekundenzehner hochzählen und wieder die Sekundeneinheit zu Null anfangen.

3.1.2 Sekundenzehner

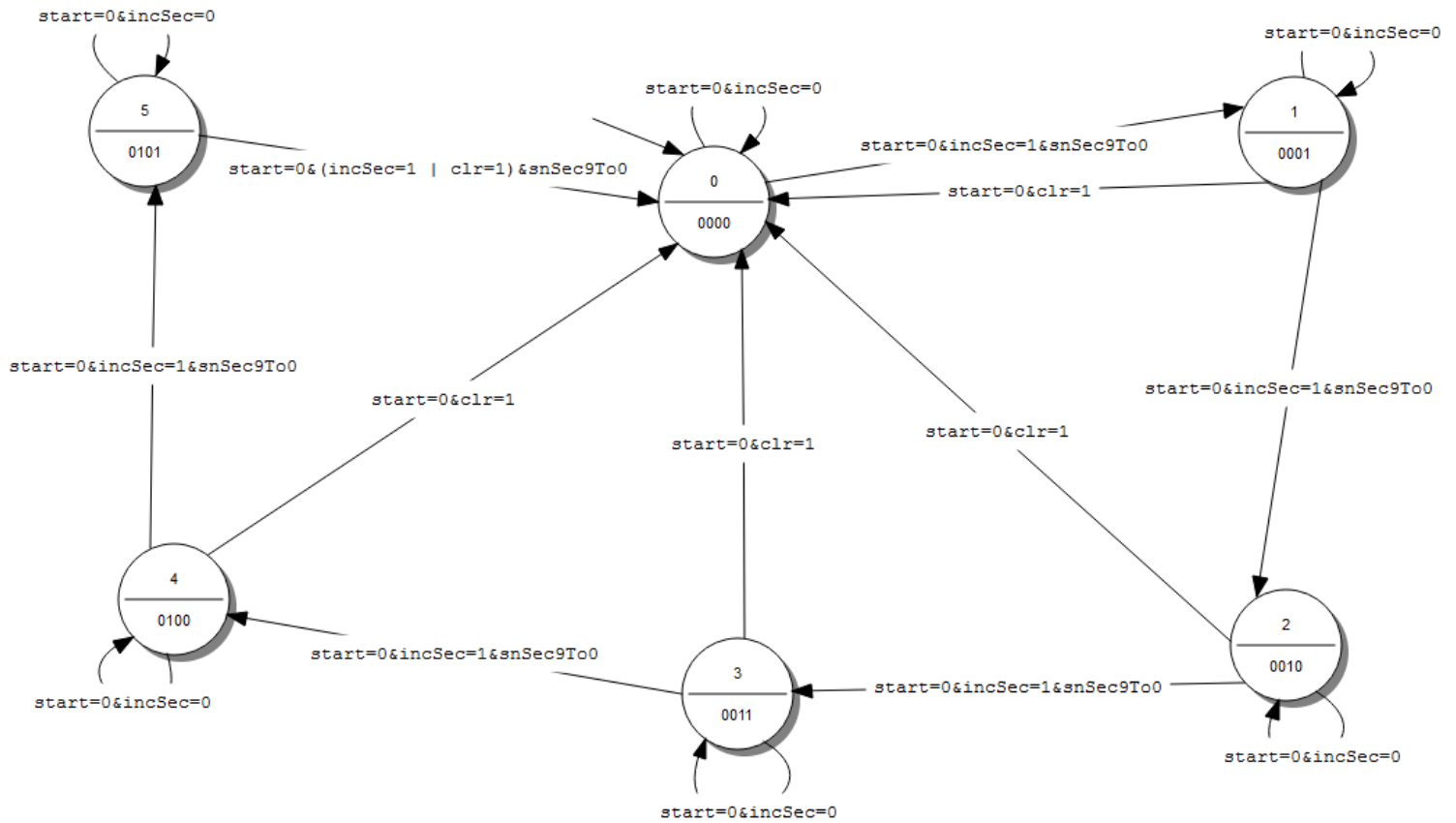


Abbildung 3: Zustandsdiagramm des Sekundenzehner-Vorwärtszählers

Der Sekundenzehner-Vorwärtszähler zählt nur wenn man in Stop mode ist hoch, wenn eine Betätigte auf der Sekunde-Taste gibt und wenn die Sekundeneinheit von 9 bis 0 geht. Wenn die 5 erreichen ist, fängt der Sekundenzehner wieder zu 0 an.

3.1.3 Minuteneinheit

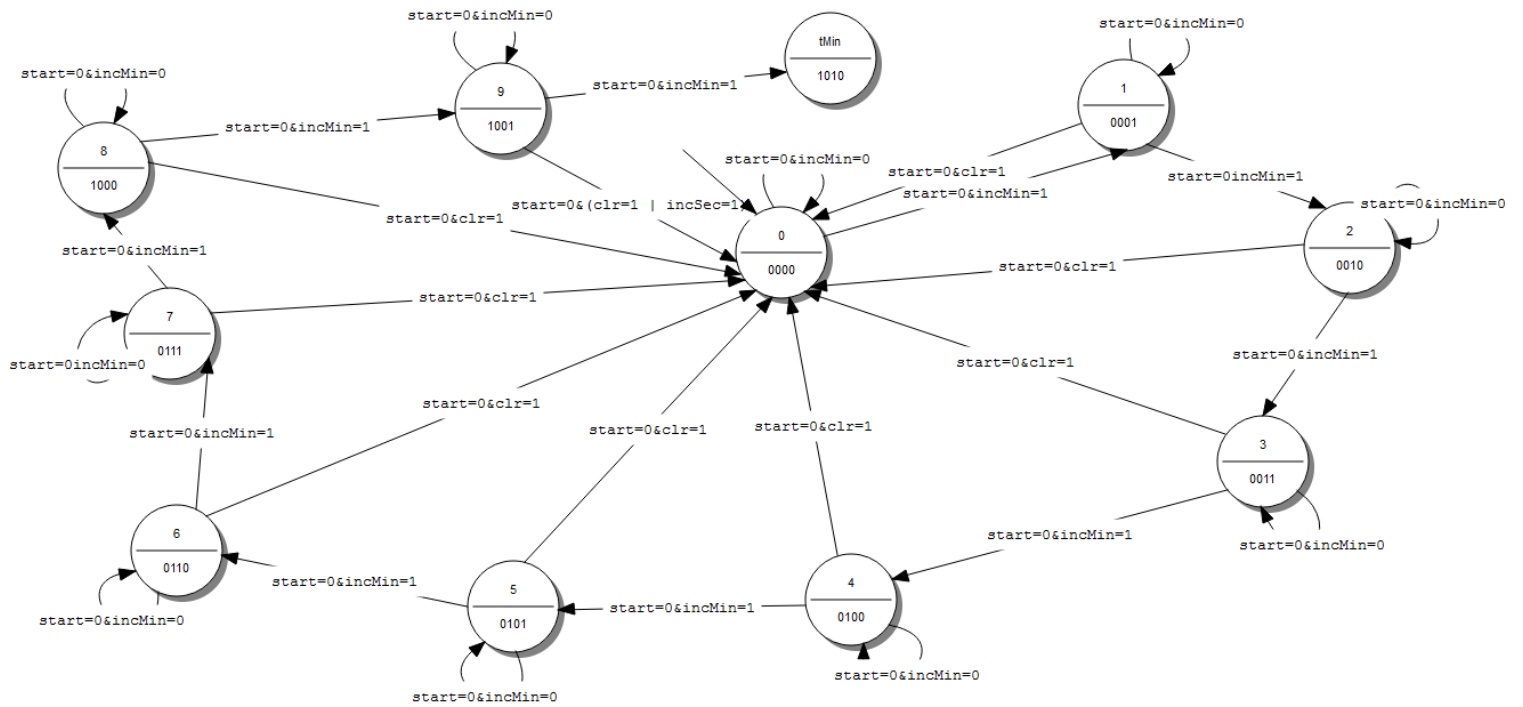


Abbildung 4: Zustandsdiagramm des Minuteneinheit-Vorwärtszählers

Wie in Sekundeneinheit, sondern für die Minuten.

3.1.4 Minutenzehner

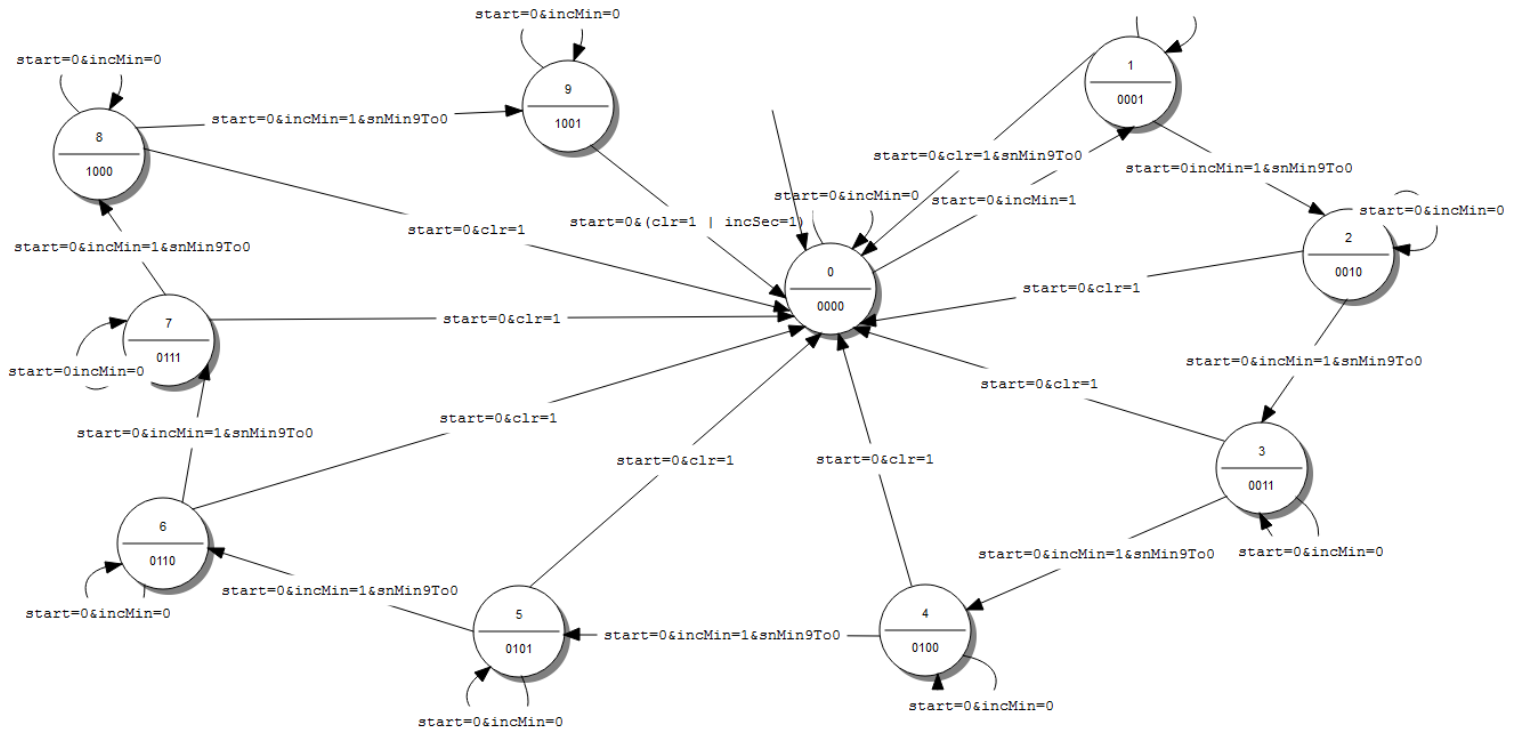


Abbildung 5: Zustandsdiagramm des Minutenzehner-Vorwärtszählers

Wie in Sekundenzehner, sondern für die Minuten.

3.2 Rückwärtszähler / Decounter

3.2.1 Sekundeneinheit

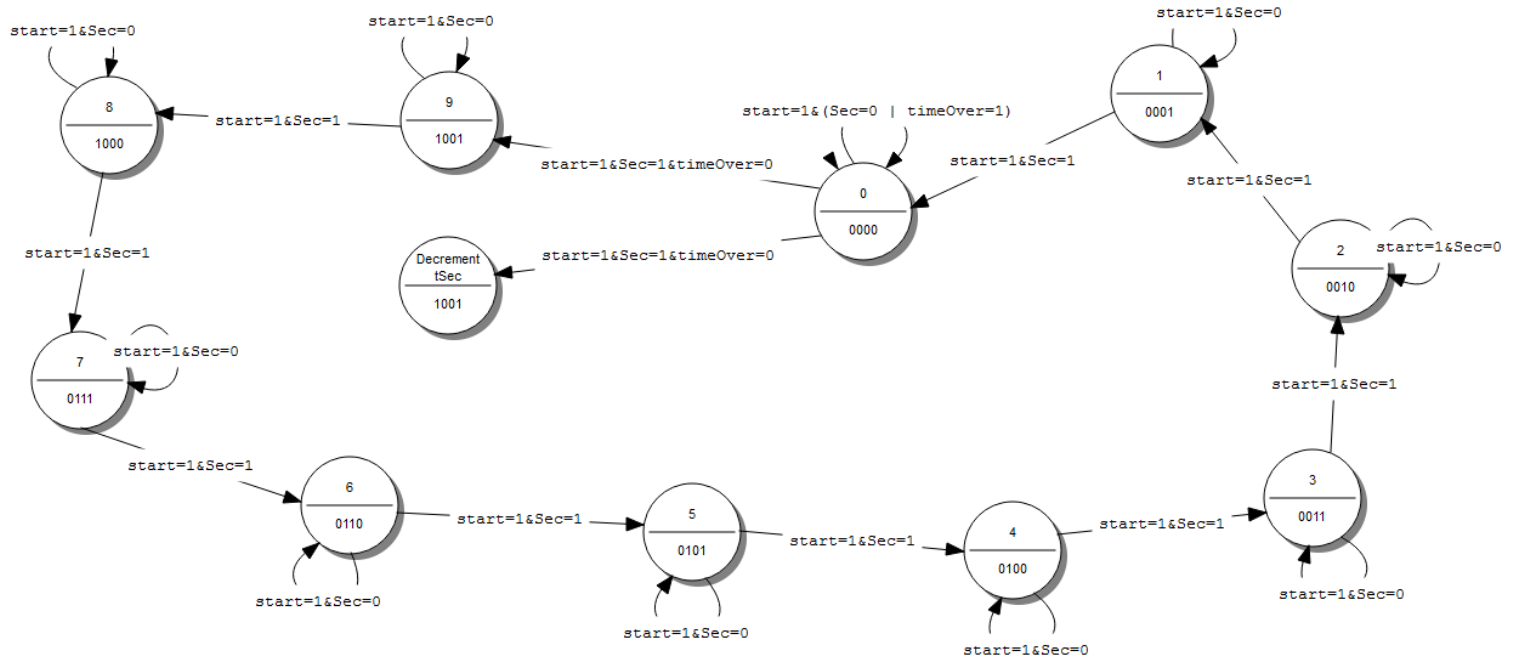


Abbildung 6: Zustandsdiagramm des Sekundeneinheit-Rückwärtszählers

Der Sekundeneinheit-Rückwärtszähler fängt mit dem letzten Wert von dem Sekundeneinheit-Vorwärtszähler, wenn man in Start mode ist und wenn eine Sekunde vergangen ist an.

Wenn der Wert 0 erreichen ist und wenn die anderen Werte noch nicht zu 0 sind, fängt der Wert wieder zu 9 an. In dem Fall die anderen Werte zu 0 sind, bleibt dieser Wert zu 0 und geht die timeOver-Signal zu 1.

3.2.2 Sekundenzehner

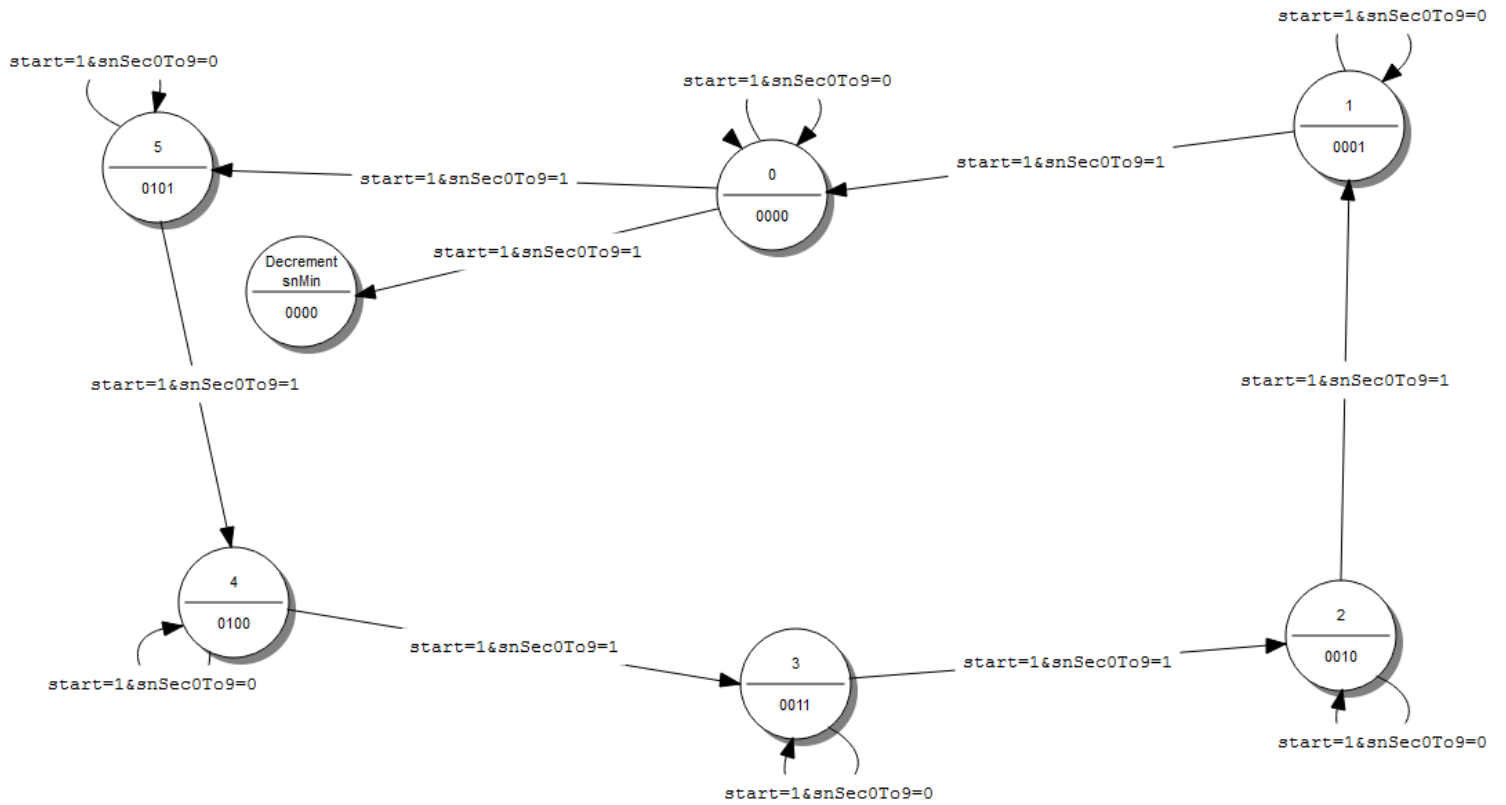


Abbildung 7: Zustandsdiagramm des Sekundenzehner-Rückwärtszählers

Der Sekundenzehner-Rückwärtszähler fängt mit dem letzten Wert von dem Sekundenzehner-Vorwärtszähler, wenn man in Start mode ist und wenn die Sekundeneinheit von 9 zu 0 geht.

3.2.3 Minuteneinheit

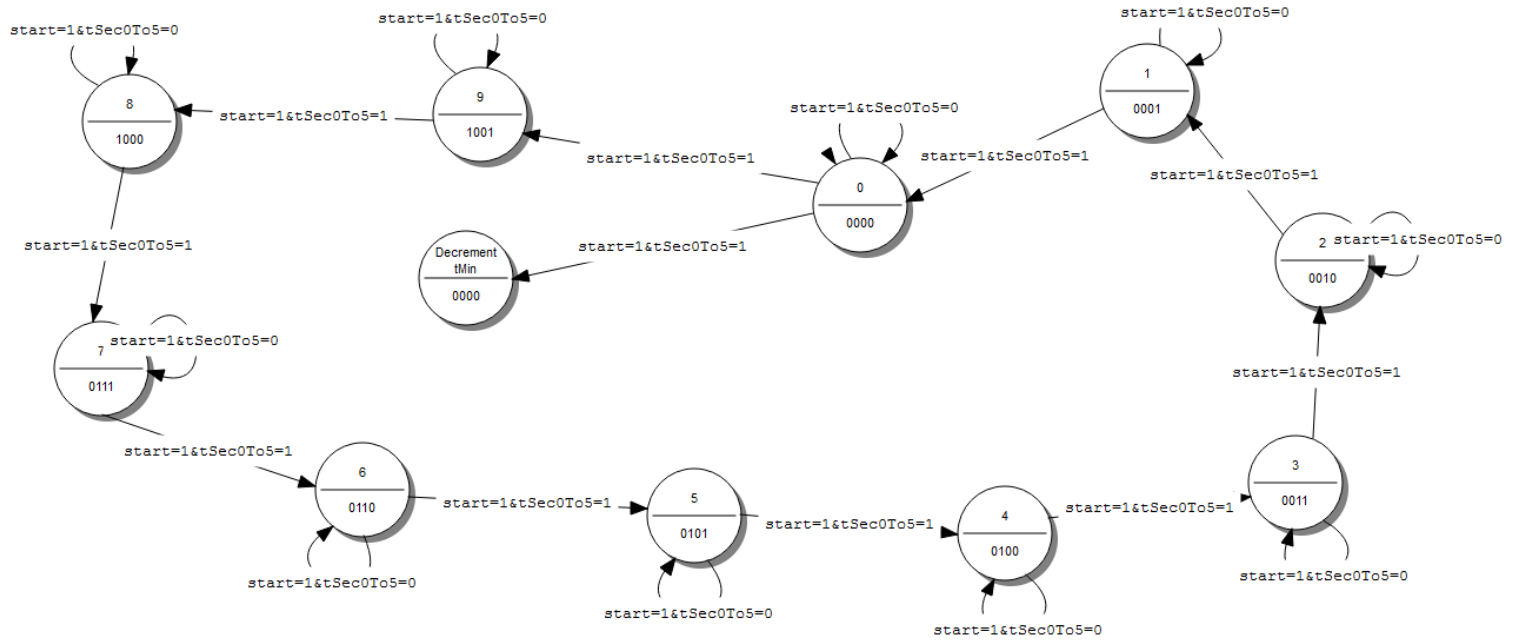


Abbildung 8: Zustandsdiagramm des Minuteneinheits-Rückwärtszählers

Der Minuteneinheits-Rückwärtszähler fängt mit dem letzten Wert von dem Minuteneinheit-Vorwärtszähler, wenn man in Start mode ist und wenn die Sekundenzehner von 0 zu 5 geht an.

3.2.4 Minutenzehner

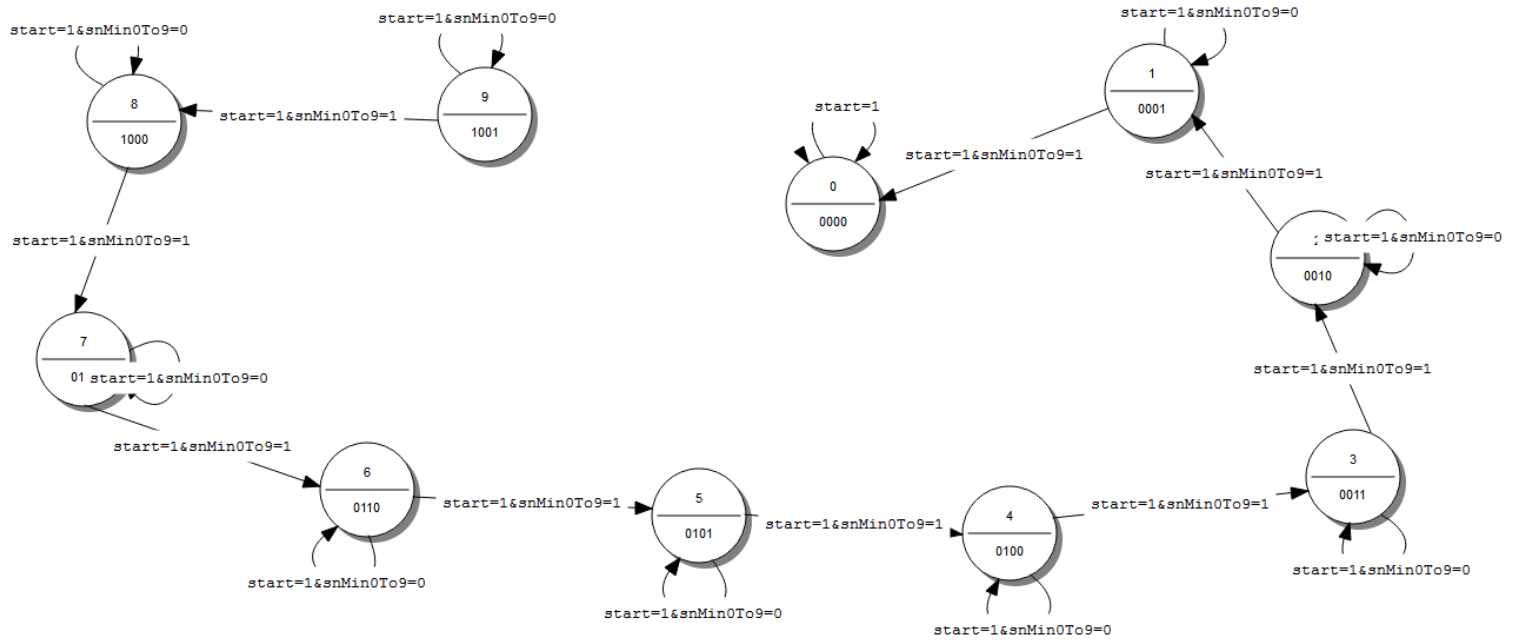


Abbildung 9: Zustandsdiagramm des Minutenzehner-Rückwärtszählers

Der Minutenzehner-Rückwärtszähler fängt mit dem letzten Wert von dem Minutenzehner-Vorwärtszähler, wenn man in Start mode ist und wenn die Minuteneinheit von 0 zu 9 geht an.

4 Simulation mit ModelSim

4.1 Debounce (für Drucktaste)

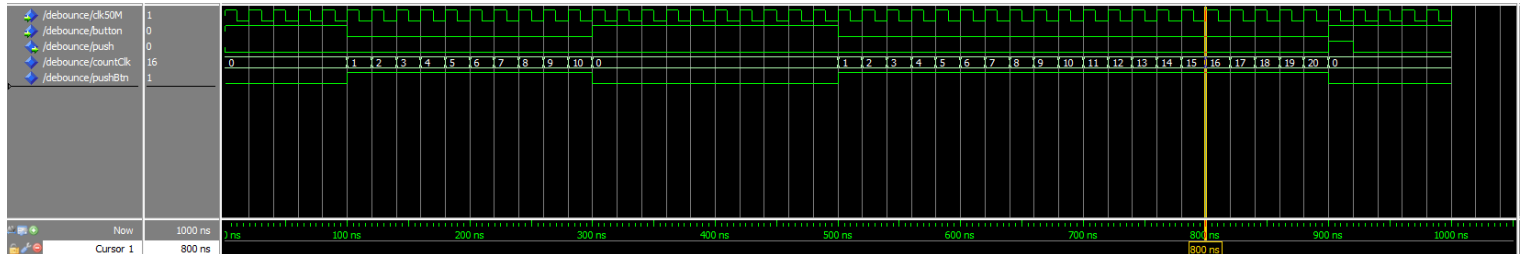


Abbildung 10: Simulation eine Debounce von 250ns

1. Problem

- Wenn Rauschen auf einer Taste gibt, kann ein unwünscher Wert stattfinden. Deswegen soll ein Debounce herstellen, um diese Situation zu vermeiden.
- Für diese Simulation ist eine Warte von 250ns benutzt. In praktisch ist eine Warte von 10ms benutzt.

2. Ergebnis (Siehe Abbildung 10)

- Diese Simulation ist mit `debouce.do` geprüft
- Ab 0 fäng eine Betätige während 200ns an. Es ist nicht genug, deswegen bleibt den push zu 0.
- Ab 500ns fäng wieder eine Betätige während 400ns an. Das ist genug. Man hat also ein push, die eine richtige Betätige darstellt.
- Diese Debounce ist für jede Taste benutzen.

4.2 Frequenzteiler

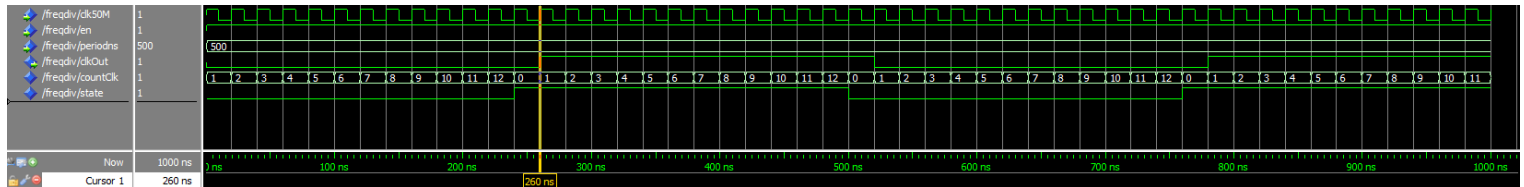


Abbildung 11: Simulation eine 500ns-Period-Signal mit dem Frequenzteiler

1. Spezifikation

- Das Vermessen der Zeit soll derart erfolgen, dass die zu vermessende Zeit in 1-Sekunden- Schritten heruntergezählt wird.
- Für diese Simulation ist ein Period von 500ns benutzt.

2. Ergebnis (Siehe Abbildung 11)

- Diese Simulation ist mit [freqDiv.do](#) geprüft
- Man sieht, dass eine Period 520ns dauert. Man hat diese Verspäterung, weil die Signale mit dem 50-MHz-Clock synchronisiert sind. Es ist nicht schlimm wegen 1 Sekunde in praktisch.

4.3 Vorwärtszähler (Counter)

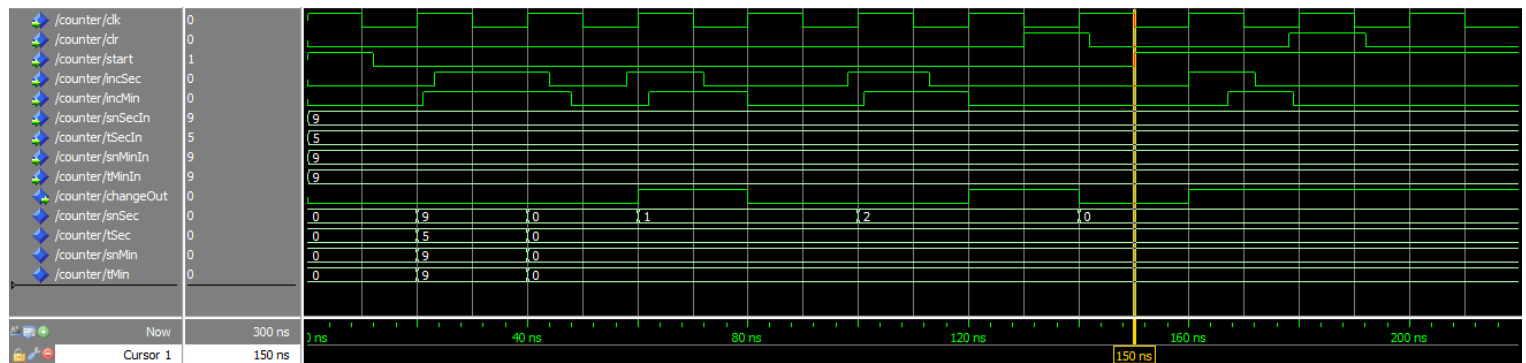


Abbildung 12: Simulation des Vorwärtszählers

1. Spezifikation

- Die Taste Start/Stop soll die Steuerung des Messvorgangs der zu messenden Zeit ermöglichen.
- Mit der Clear-Taste soll die messende Zeit auf 0 zurückgesetzt werden.
- Durch Betätigen der Min-Taste sollen die zu messenden Minuten um jeweils eine Minute erhöht werden.
- Durch Betätigen der Sec-Taste sollen die zu messenden Sekunden um jeweils eine Sekunde erhöht werden.
- Die zu messende Zeit soll maximal 99 Minuten und 59 Sekunden betragen können.

2. Ergebnis (Siehe Abbildung 12)

- Diese Simulation ist mit `counter.do` geprüft
- Man fängt diese Simulation in Stop Zustand an, um der Werte 99 Minuten 59 Sekunden zu haben, die in praktisch aus der Decounter kommt.
- Nach Betätigen die Sekunde- und Minute-Taste gehen die Sekunden und Minuten wieder zu null, weil der maximale Wert (99min59sec) erreichen ist.
- Ab 60ns kann man sehen, dass die Werte nach Betätigen der Sekunde- und Minute-Taste hochzählen (und die mit dem Clock synchronisiert sind).
- Ab 130ns gibt eine Clear-Betätige, die alle Werte wieder zu null macht.
- Ab 160ns geht man wieder zu den Start Mode. Man sieht dann, dass die Betätigen nicht mehr funktionieren.

4.4 Rückwärtszähler (Decounter)

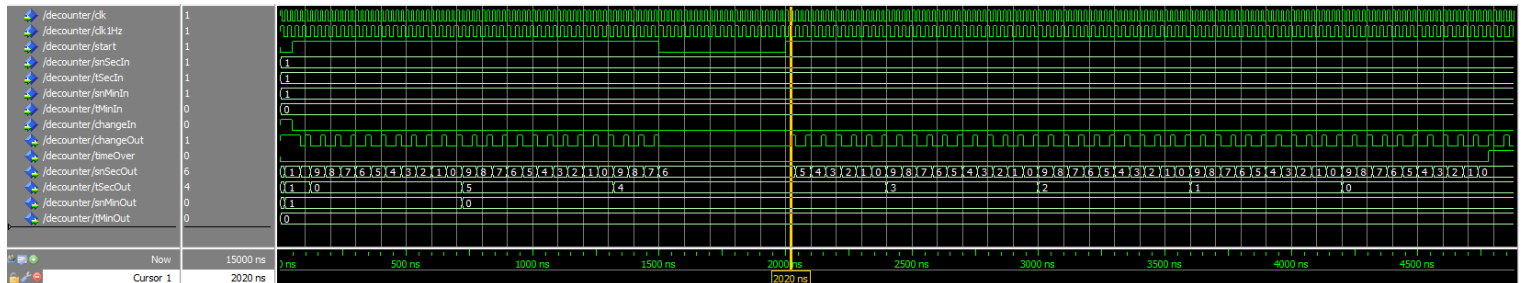


Abbildung 13: Simulation des Rückwärtszählers

1. Spezifikation

- Durch Betätigen der Start-/Stop-Taste soll die Vermessung der eingestellten, zu vermessende Zeit gestartet werden.
- Das Vermessen der Zeit soll derart erfolgen, dass die zu vermessende Zeit in 1-Sekunden- Schritten heruntergezählt wird.
- Wird während des Messvorgangs die Start/Stop-Taste betätigt, so soll der Messvorgang abgebrochen werden. Es soll in diesem Fall die noch verbleibende zu messende Zeit auf der 7-Segment-Anzeige dargestellt werden.
- Es soll möglich sein, durch Betätigung der Start/Stop-Taste der Messvorgang weiter durchzuführen.

2. Ergebnis (Siehe Abbildung 13)

- Diese Simulation ist mit `decounter.do` geprüft
- Man fängt diese Simulation in Start Zustand an, um mit der Werte 01 Minute 10 Sekunden zu haben, die praktisch aus der Counter kommt.
- Ab 50ns zählt der Decounter jede rising edge von den 1-Hz-Clock runter, bis 1500ns, wo man wieder in Start Zustand geht. Man sieht dann, dass der letzte Wert bleibt.
- Ab 2000ns zählt der Decounter bis zu Null runter. Dann kann man sehen, dass das timeOver-Signal zu 1 geht. Das zeigt das Ende der Zeit.

4.5 Audio / Ende der Zeit

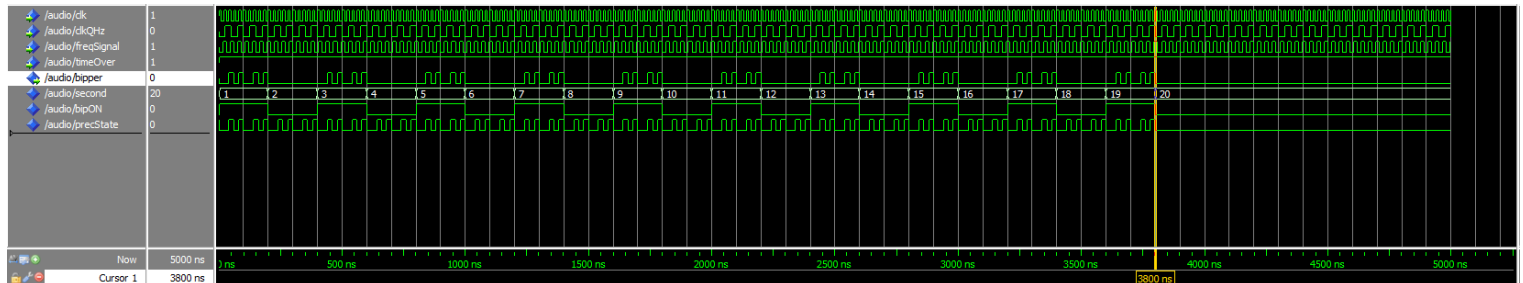


Abbildung 14: Simulation des Ende der Zeit mit dem Signalgeber und Leds

1. Spezifikation

- Wird der Messvorgang nicht durch Betätigen der Start/Stop-Taste abgebrochen, so soll nach Beendigung des Messvorgangs – also bei Erreichen der Zeit 0 – durch ein akustisches Signal signalisiert werden, dass der Messvorgang beendet wird.
- Das akustische Signal soll eine Minute lang ertönen. Der Signalgeber soll innerhalb dieser Minute jede Sekunde vier kurze Signal-Töne ausgeben. Zwischen diesen vier kurzen Tönen soll deutlich eine Signalfpause hörbar sein (also keine Ausgabe eines Dauertons).

2. Ergebnis (Siehe Abbildung 14)

- Diese Simulation ist mit [audio.do](#) geprüft
- In diese Simulation, dauert die Minute nur “20“ Sekunden
- Man sieht, dass man 4 Signal-Töne (die nochmalweise jede 250ms-Period dauert) hat, und dann eine Ruhe (die 1 Sekunde in praktisch dauert) gibt.
- Nach 20 Sekunden (die in praktisch 1 Minute ist) von Signal-Tönen gibt es eine Ruhe bis man auf Stop betätigt

4.6 Display

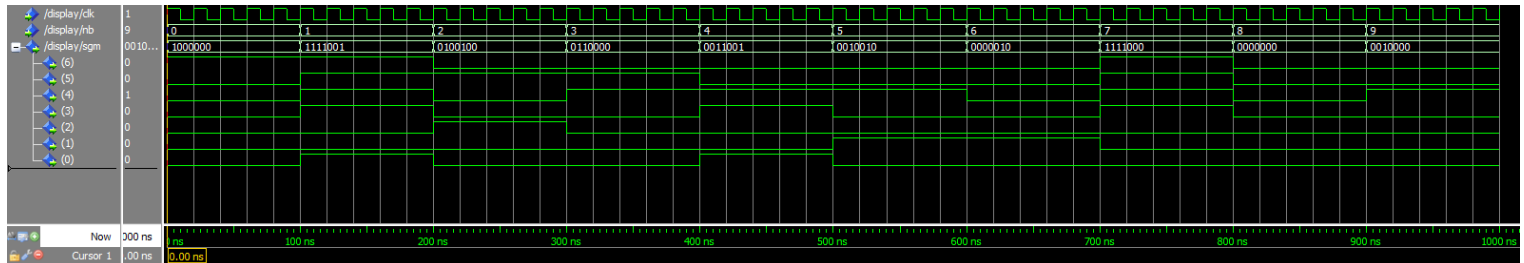


Abbildung 15: Simulation des Displays für jeden Wert (0-9)

Zahl	a(0)	b(1)	c(2)	d(3)	e(4)	f(5)	g(6)
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0

Tabelle 1: Darstellung Zahl - 7-Segment

1. Spezifikation

- Die zu messende bzw. die noch verbleibende, zu messende Zeit während des Messvorgangs soll auf den 7-Segment-Anzeigen dargestellt werden.

2. Ergebnis (Siehe Abbildung 15)

- Diese Simulation ist mit display.do geprüft
- Siehe Tabelle 1 und Abbildung 15, um die beide zusammen zu vergleichen

4.7 Ganze Architektur (Timer)

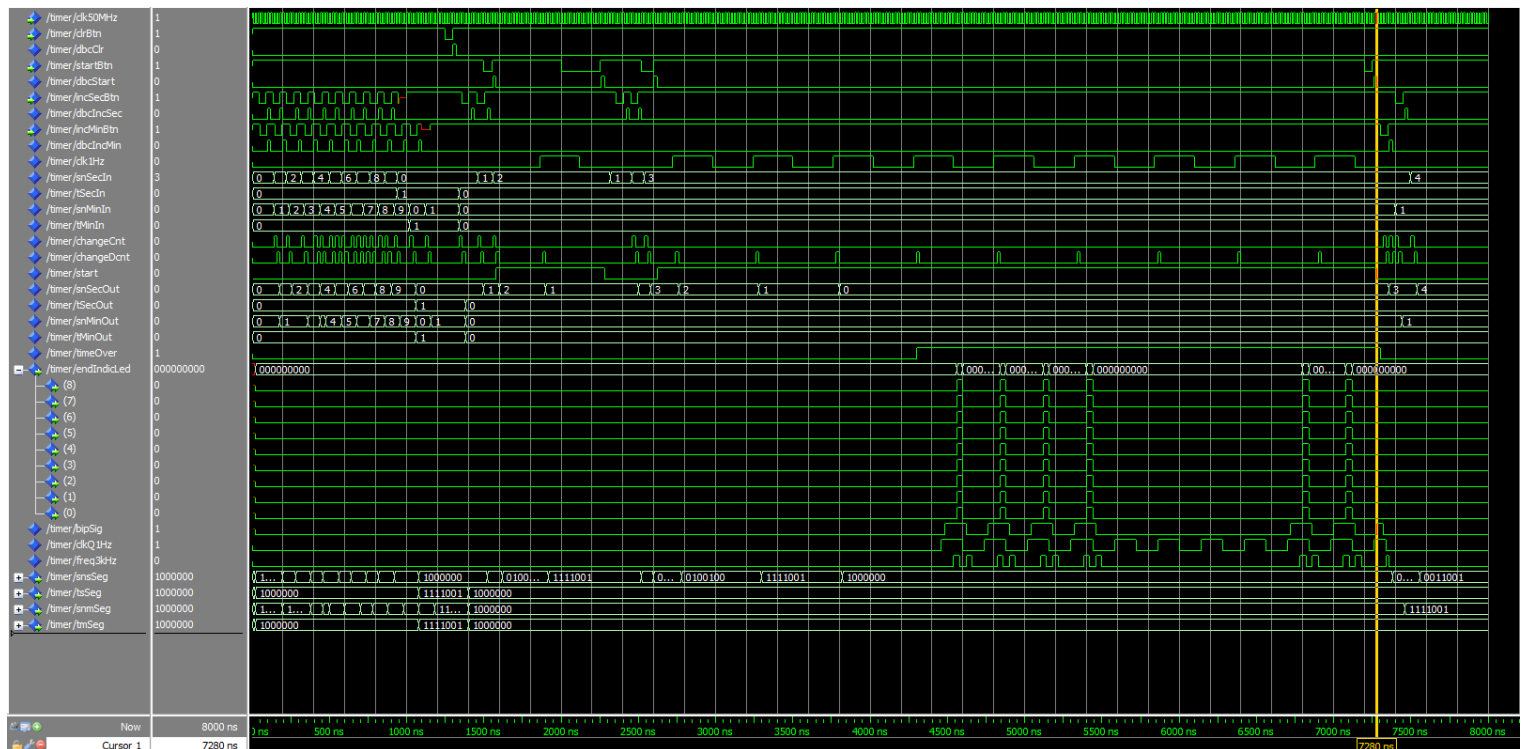


Abbildung 16: Simulation des Rückwärtszählers

1. Spezifikation, die noch kein Ergebnis gibt

- Nach Abbruch des Messvorgangs soll prinzipiell wieder ein Einstellen, der zu messenden Zeit möglich sein.
- Wird der Messvorgang nicht durch Betätigen der Start/Stop-Taste abgebrochen, so soll nach Beendigung des Messvorgangs – also bei Erreichen der Zeit 0 – durch ein akustisches Signal signalisiert werden, dass der Messvorgang beendet wird.
- Nach Ablauf der akustischen Signalisierung soll auf der 7-Segment-Anzeige die ursprünglich während des Einstellvorgangs eingestellte, zu vermessende Zeit dargestellt werden.
- Die über eine Minute andauernde Ausgabe des akustischen Signaltons, soll durch Betätigen der Start/Stop-Taste abgebrochen werden können. Auch in diesem Fall soll auf der 7- Segment-Anzeige die ursprünglich während des Einstellvorgangs eingestellte, zu vermessende Zeit dargestellt werden.

2. Ergebnis (Siehe Abbildung 16)

- Diese Simulation ist mit [timer.do](#) geprüft
- Am Anfang, zählt man, bei Betätige die Sekunde- und Minute-Taste, der Counter hoch.
- Ab 1300ns gibt es eine Betätige auf der Clear-Taste, die die Werte von Counter zu Null macht.
- Man zählt die Counter wieder hoch (bis 2 Sekunden).
- Ab 1550ns gibt es ein Betätige auf der Start-Taste.
- Dann zählt der Decounter runter, bis man auf der Start-Taste betätigt (ab 2250ns).
- Wir sind in diesem Moment in Stop mode. Man zieht, dass der letzte Wert von Decounter bleibt.
- Wir zählen diesen Wert wieder hoch, bis 3 Sekunden und gehen wieder in Start mode (ab 2600ns) bis man den 0 Wert erreichen ist (ab 4250ns)
- Dann gibt es die Signal-Töne und eine Ruhe, bis man auf den Start-Taste wieder betätigt (7250ns).
- Man sieht in diesem Fall, dass der letzte Wert programmiert (vor wir in Start mode gehen) wieder dargestellt ist (auf snSecOut).
- Man kann wieder diesen Wert hochzählen, und wieder in Start mode gehen.

5 Konklusion

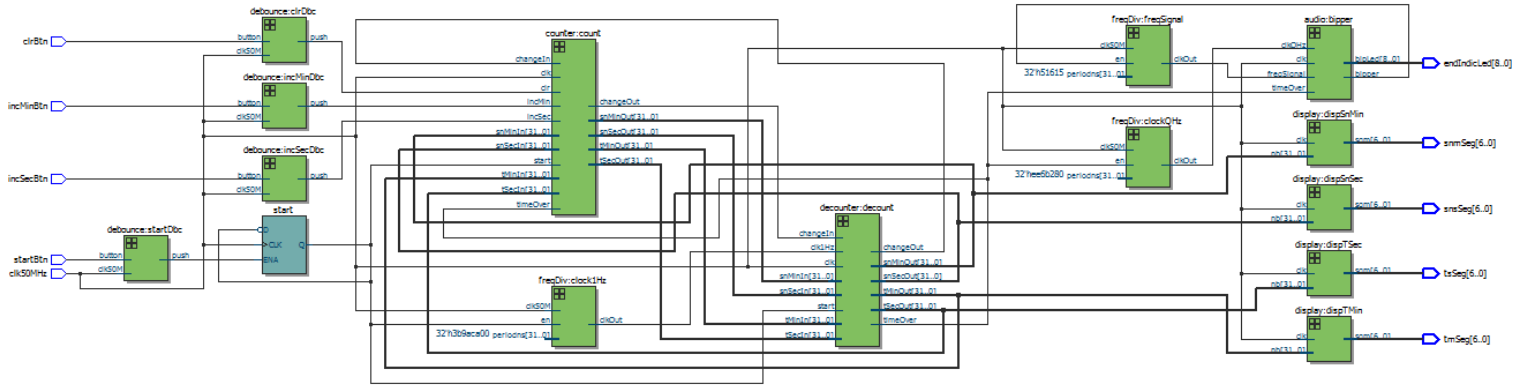


Abbildung 17: Darstellung des Projekts mit RTL-Viewer von Quartus

Abschließend kann man die Abbildung 1, die am Anfang des Projekts hergestellt ist, mit der Abbildung 17, die am Ende des Projekts hergestellt ist, vergleichen. Man sieht dass, was man am Anfang möchte, ist am Ende hergestellt. Das zeigt, dass die Architektur am Anfang sehr wichtig um etwas Komplex zu herstellen ist.

Reihenweise sind jede Spezifikation mit einem Do-File geprüft (Siehe jede Schritt in 3. Simulation mit ModelSim). Alle Spezifikationen sind einhalten. Außerdem, war diese Projekt am Anfang mit einigen Modulus und Teilunge gemacht (für die Decoupler). Aus Neugier, möchte ich wissen, ob das optimisch ist. Mit dem RTL-Viewer hatte ich ein Architektur mit Modulus und Teilunge mit eine mit IF-Bedingunge, anstatt die beide, vergleichen. Die Ergebnisse sind Abbildung 18 und 19 dargestellt. Man sieht, dass die Modulus und Teilunge viele Logik benutzt. Deswegen habe ich letztendlich diese Decoupler nur mit IF-Bedingunge hergestellt.

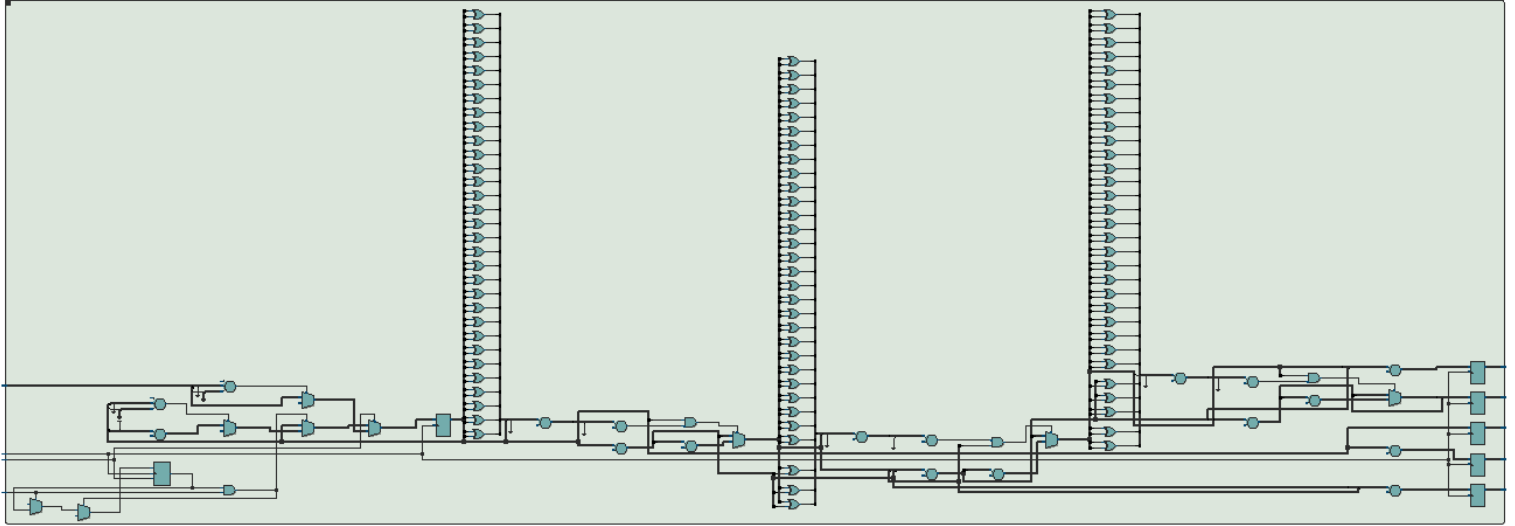


Abbildung 18: Darstellung der Decounter mit Modulus und Teilung

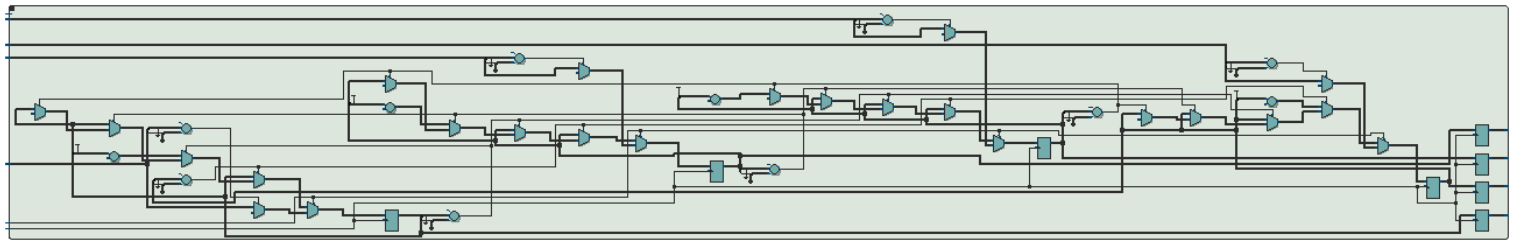


Abbildung 19: Darstellung der Decounter mit IF-Bedingunge