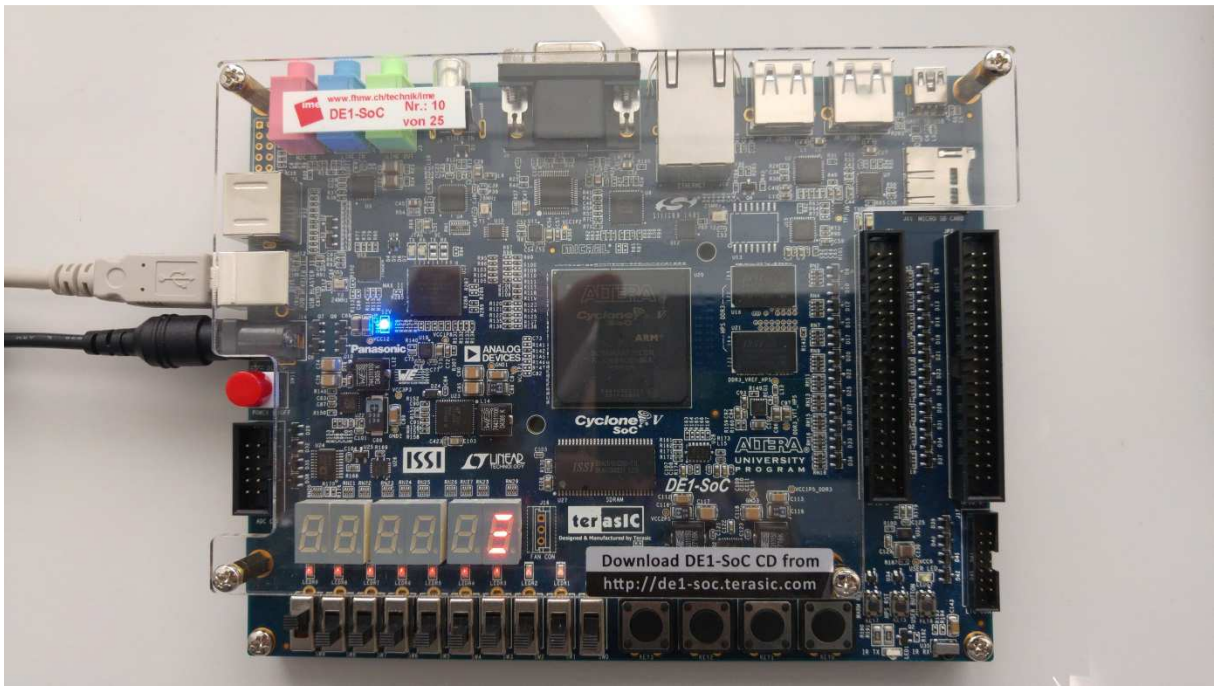
PROJEKT "DICE"

(elektronischer Würfel)



Realisierung eines elektronischen Würfels

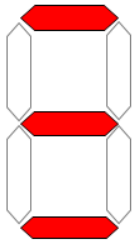
1. System Spezifikation

Sie wollen einen elektronischen Würfel entwickeln, der folgendermassen funktioniert:

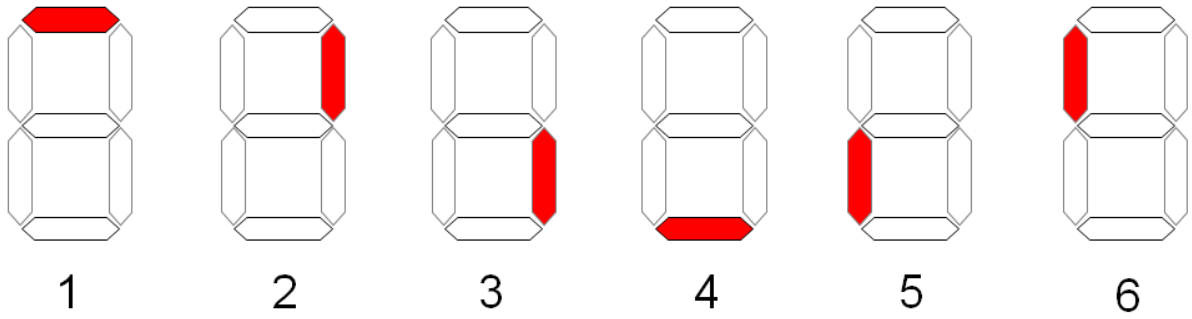
Als Interface dienen eine Taste und eine 7-Segment-Anzeige. Folgende Punkte sollen realisiert werden:

- Zu Beginn (und nach einem Reset) wird das Resetmuster angezeigt.
- Durch Drücken der Taste wird intern ein Zufallsgenerator aktiviert. Auf der Anzeige sollen die Segmente der 7-Segment-Anzeige im Uhrzeigersinn rotieren.
- Wenn die Taste losgelassen wird, sollen die Segmente in die andere Richtung rotieren, langsamer werden und anschliessend eine Zufallszahl zwischen 1 und 6 auf der 7-Segment-Anzeige darstellen.

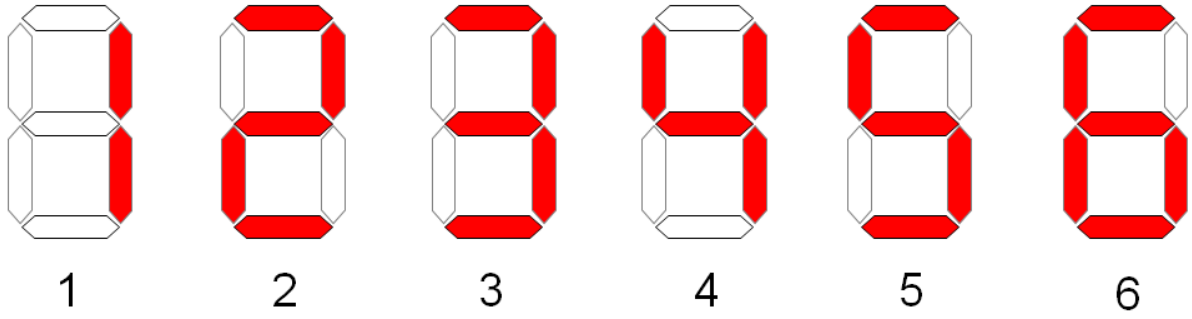
Resetmuster:



Animationsmuster (Uhrzeigersinn):



Zahlenmuster:



2. Strukturierte Analyse

2.1 Context Diagramm

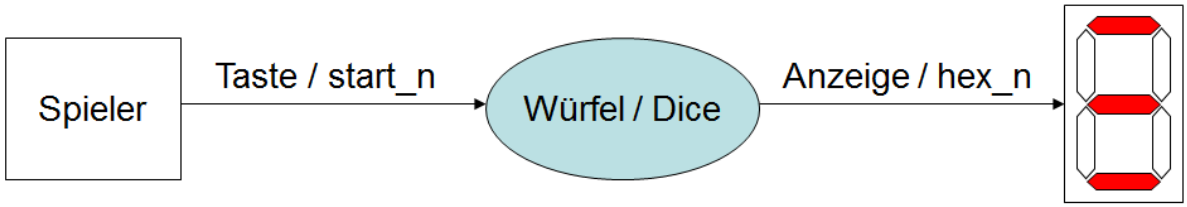


Abbildung 1: CD

Data Dictionary:

Signal	Richtung	Beschreibung
start_n	input	Start-Taste: Taste gedrückt: Schnelles Rotationsmuster zeigen (Zufallszahl generieren) Taste loslassen: Langsamer werdendes Rotationsmuster anzeigen, dann Zufallszahl anzeigen
hex_n (6:0)	output	Anzeige des Rotationsmusters bzw. der Zufallszahl (7-Segment-Anzeige)

2.2 Datenfluss Diagramm 0

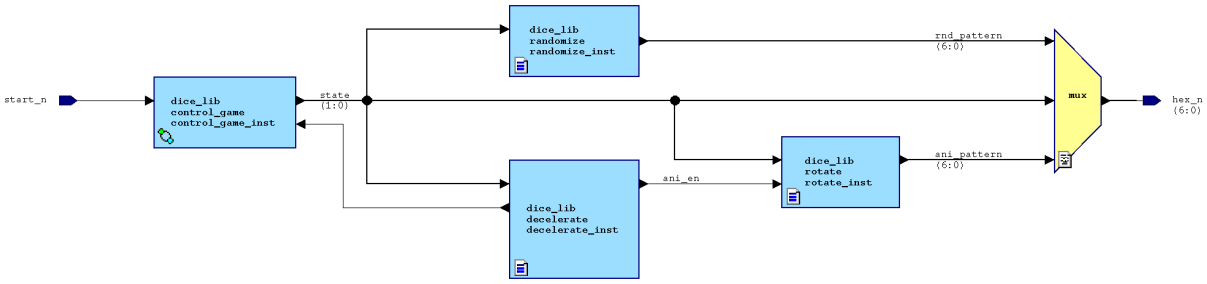


Abbildung 2: DFD0

Data Dictionary:

Signal	Richtung	Quelle	Beschreibung
reset_n	input	extern	System Reset, active low
clock	input	extern	50 MHz System Clock
start_n	input	input	Steuert das Würfeln, active low 0 : Taste gedrückt: schnelle Rotation anzeigen, Zufallszahl generieren 1 : Taste loslassen: langsamer werdende Rotation anzeigen, Zufallszahl anzeigen
state	intern	control_game	Spielzustand 00_{bin} = show_number 01_{bin} = show_animation 10_{bin} = slow_down_animation
ani_en	intern	decelerate	Rotation / Animation enable: Puls der Breite eines clock-cycles, Frequenz abh. von state
rnd_pattern	intern	randomize	7-Segment-Anzeige-Pattern für die Zufallszahl
ani_pattern	intern	rotate	7-Segment-Anzeige-Pattern für das Animationsmuster
hex_n (6:0)	output	output	Ansteuerung der einzelnen Segmente, active low

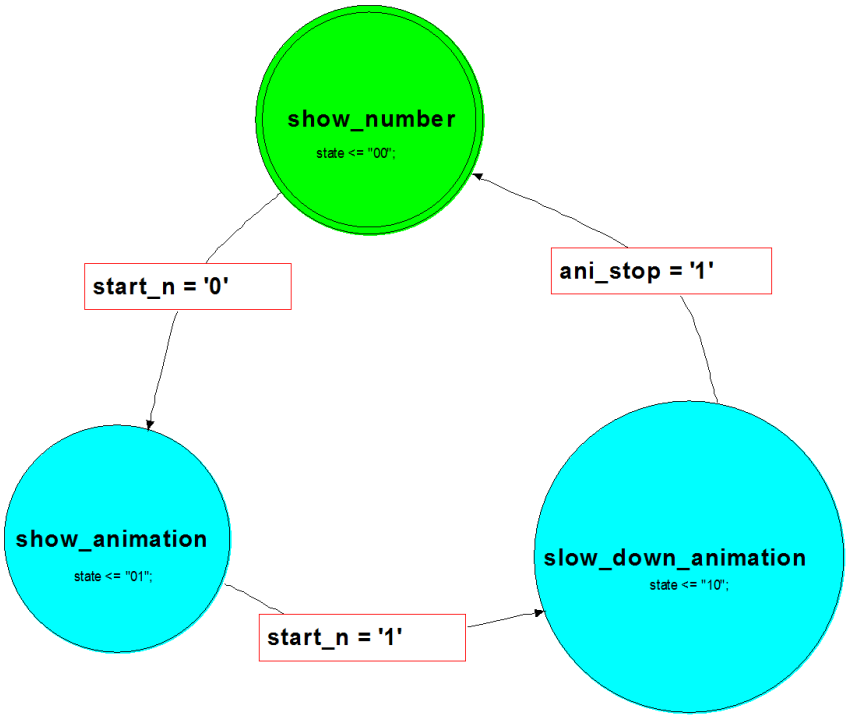
2.3 PSpec "control_game"

Das Spiel unterscheidet drei Zustände:

1. **show_animation**
   Segmente rotieren im Uhrzeigersinn. Wenn start_n losgelassen wird, Wechsel zu slow_down_animation
2. **slow_down_animation**
   Segmente rotieren im Gegenuhrzeigersinn und werden immer langsamer. Beim Stillstand Wechsel zu show_number.
3. **show_number**
   Segmente zeigen Zufallszahl. Wenn start_n gedrückt wird, Wechsel zu show_animation

Data Dictionary:

Signal	Richtung	Datentyp	Beschreibung
reset_n	input	std_ulogic	System Reset, active low
clock	input	std_ulogic	50 MHz System Clock
start_n	input	std_ulogic	Steuert das Würfeln, active low: 0: Animation anzeigen, Zufallszahl generieren 1: Animation auslaufen, Zufallszahl anzeigen
ani_stop	input	std_ulogic	Stop Puls für die Animation
state	output	std_ulogic_vector(1:0)	Spielzustand 00_{bin} = show_number 01_{bin} = show_animation 10_{bin} = slow_down_animation



2.4 PSpec "decelerate"

Das Modul "Decelerate" ist ein IP (Intellectual Property), d.h. ein Modul, dessen Funktionalität zur Verfügung steht und benutzt werden kann. Damit dieses Modul richtig eingebunden werden kann, muss das Interface genau beschrieben sein:

Data Dictionary:

Signal	Richtung	Datentyp	Beschreibung
reset_n	input	std_ulogic	System Reset, active low
clock	input	std_ulogic	50 MHz System Clock
state	input	std_ulogic_vector(1:0)	Steuert die Ausgabe: 00_{bin} : Keine Animation: ani_en = 0, nötig für einen Neustart 01_{bin} : Schnelle Animation: f(ani_en) = 24Hz 10_{bin} : Verlangsamende Animation: f(ani_en) von 24 Hz auf 0 Hz in 10 Schritten 11_{bin} : nicht definiert: gleich wie 10_{bin} behandeln
ani_en	output	std_ulogic	Animation enable: Puls der Breite eines clock-cycles, Frequenz abh. von state
ani_stop	output	std_ulogic	Animation stop: statisches Signal, wird erst mit state = 00_{bin} zurückgesetzt

2.5 PSpec "randomize"

Solange die Animation läuft, wird eine Zufallszahl berechnet. Ist die Animation beendet, wird das entsprechende Bitmuster ausgegeben.

Eine zufällige Zahl zwischen 1 und 6 wird erreicht, indem man einen Zähler mit der Systemfrequenz von 50 MHz über eine zufällige Zeitspanne laufen lässt. Wenn die Würfel-Taste gedrückt wird (Signal start_n = '0'), beginnt der Zähler zu laufen und stoppt an einer quasi zufälligen Position beim Loslassen der Taste.

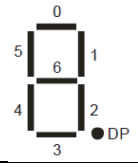
Data Dictionary:

Signal	Richtung	Datentyp	Beschreibung
reset_n	input	std_ulogic	System Reset, active low
clock	input	std_ulogic	50 MHz System Clock
state	input	std_ulogic_vector(1:0)	Spielzustand: 00_{bin} = show_number 01_{bin} = show_animation 10_{bin} = slow_down_animation
rnd_pattern	output	std_ulogic_vector(6:0)	7-Segment-Pattern für die Zufallszahl

2.6 PSpec "rotate"

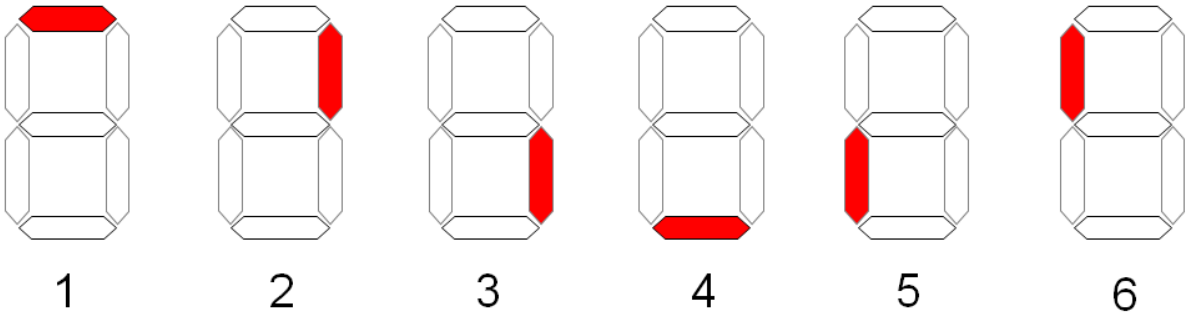
Bei der Animation soll immer ein Segment der 7-Segment-Anzeige in die entsprechende Richtung rotieren. Die Geschwindigkeit und die Drehrichtung wird durch Steuersignale bestimmt.

Data Dictionary:

Signal	Richtung	Datentyp	Beschreibung
reset_n	input	std_ulogic	System Reset, active low
clock	input	std_ulogic	50 MHz System Clock
state	input	std_ulogic_vector(1:0)	Spielzustand: 00_{bin} = show_number 01_{bin} = show_animation / Rotation im Uhrzeigersinn 10_{bin} = slow_down_animation / Rotation im Gegenuhrzeigersinn
ani_en	input	std_ulogic	Enable für die Animation
ani_pattern	output	std_ulogic_vector(6:0)	7-Segment-Pattern für das Animationsmuster Bit 0 = Segment 'A' Bit 1 = Segment 'B' … … Bit 6 = Segment 'G' 

Überlegen Sie sich einen einfachen Algorithmus, um die LED im und entgegen dem Uhrzeigersinn rotieren zu lassen.

Animationsmuster (Uhrzeigersinn):



2.7 PSpec "mux"

Während der Animation wird das 7-Segment-Pattern mit den Animationsmustern an die Ausgänge geschaltet, danach das 7-Segment-Pattern mit der Zufallszahl.

Data Dictionary:

Signal	Richtung	Datentyp	Beschreibung
state	input	std_ulogic_vector(1:0)	Spielzustand: 00_{bin} = show_number 01_{bin} = show_animation 10_{bin} = slow_down_animation
ani_pattern	input	std_ulogic_vector(6:0)	7-Segment-Pattern für die Zufallszahl
rnd_pattern	input	std_ulogic_vector(6:0)	7-Segment-Pattern für das Animationsmuster
hex_n	output	std_ulogic_vector(6:0)	Ansteuerung der sieben Segmente, active low

3. DE1-SoC Quartus

FAMILY "Cyclone V SE"

DEVICE 5CSEMA5F31C6

3.1 Pinning Liste

VHDL Name (dice_top)	FPGA Pin (DE0-Board)
reset_n	PIN_AA14 (KEY0)
clock	PIN_AF14
start_n	PIN_Y16 (KEY3)
hex_n[0]	PIN_AE26
hex_n[1]	PIN_AE27
hex_n[2]	PIN_AE28
hex_n[3]	PIN_AG27
hex_n[4]	PIN_AF28
hex_n[5]	PIN_AG28
hex_n[6]	PIN_AH28