

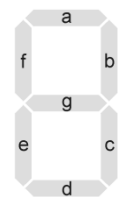
## Übung 1: Kombinatorische Logik

### Aufgabe: Binär-Hexadezimal-Konverter

Beschreiben Sie einen VHDL-Block, der einen vierstelligen Binärwert auf einer 7-Segment-Anzeige darstellt. Dabei sind die Ausgänge aktiv low, d.h. das entsprechende Segment leuchtet, wenn eine '0' anliegt.

Der Eingang der ENTITY ist der Bus "data" vom Typ "std\_ulogic\_vector(3 DOWNT0 0)", die Ausgänge "a" bis "g" sind Einzelsignale vom Typ "std\_ulogic". Diese Datentypen sind in der LIBRARY ieee.std\_logic\_1164 definiert.

Character	Input				Output						
	data(3)	data(2)	data(1)	data(0)	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0							
9	1	0	0	1							
A	1	0	1	0							
b	1	0	1	1							
C	1	1	0	0							
d	1	1	0	1							
E	1	1	1	0							
F	1	1	1	1							



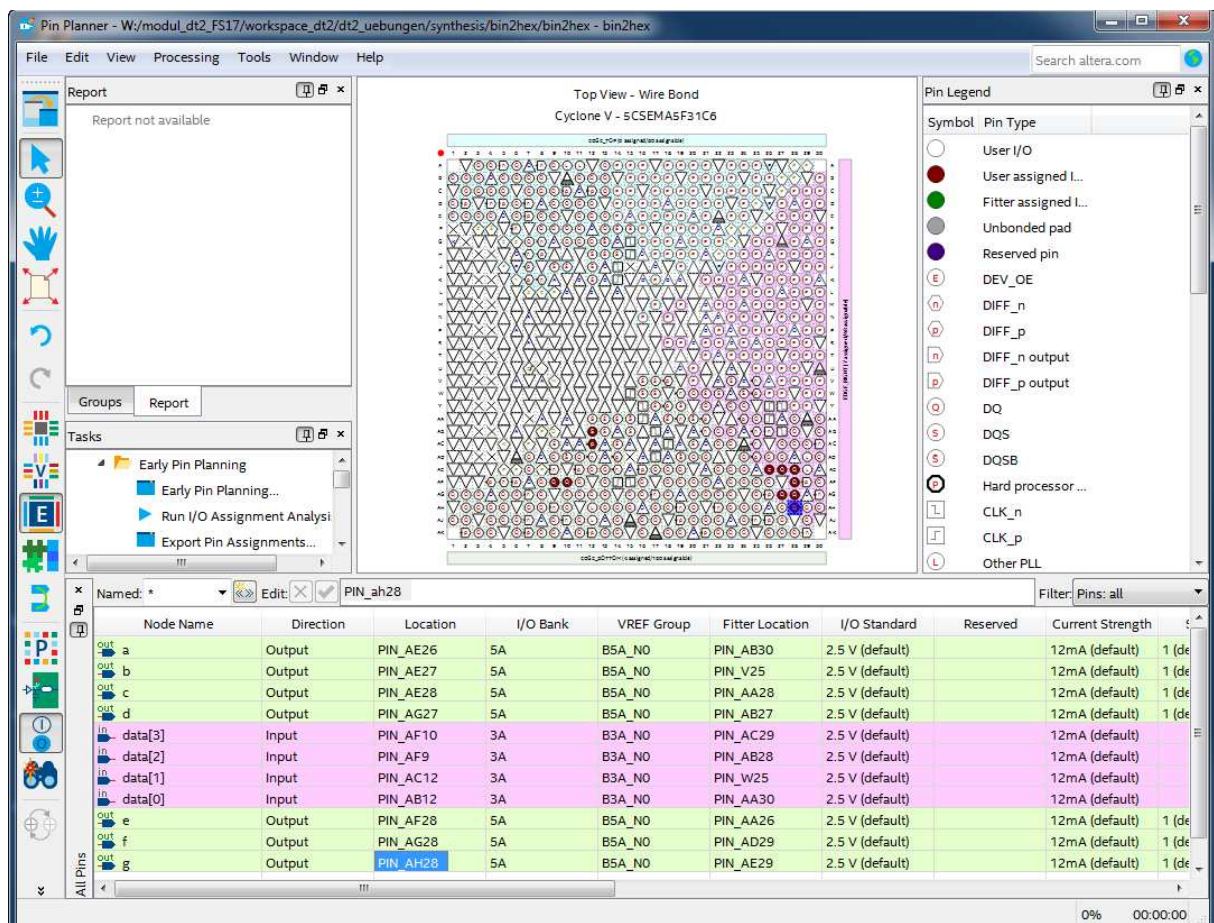
1. Ergänzen Sie die Wahrheitstabelle
2. Beschreiben Sie die bool'sche Funktionsgleichung für Ausgang a (aus der Wahrheitstabelle mit disjunktiver / ODER Normalform, dt1, S.22)
3. Beschreiben Sie für Ausgang b die reduzierte bool'sche Funktionsgleichung mithilfe eines Karnaughdiagrammes (dt1, S.23)
4. Beschreiben Sie den Binär-Hexadezimal-Konverter in VHDL (mit Sigasi, siehe nächste Seite, Design Entry)
5. Optional: Kontrollieren Sie das Syntheseresultat von Quartus mit Ihrer Lösung von 3 (unter Tools > Netlist Viewer > Technology Map Viewer)



Bevor Sie das Design auf des DE1-SoC-Board laden können, müssen Sie noch die Eingänge mappen. Ergänzen Sie also die Signalnamen und FPGA-Pin-Nummern entsprechend Ihren VHDL-Portnamen: (DE1-Soc User Manual, S. 25-27)

VHDL-Entity	Signalname Board	FPGA-Pin-Nummer	Beschreibung
data(3)	SW[3]	PIN_AF10	Slide Switch[3]
data(2)	SW[2]	PIN_AF9	Slide Switch[2]
data(1)	SW[1]	PIN_AC12	Slide Switch[1]
data(0)	SW[0]	PIN_AB12	Slide Switch[0]
a	HEX0_D[0]	PIN_AE26	Seven Segment Digit 0[0]
b	HEX0_D[1]	PIN_AE27	Seven Segment Digit 0[1]
c	HEX0_D[2]	PIN_AE28	Seven Segment Digit 0[2]
d	HEX0_D[3]	PIN_AG27	Seven Segment Digit 0[3]
e	HEX0_D[4]	PIN_AF28	Seven Segment Digit 0[4]
f	HEX0_D[5]	PIN_AG28	Seven Segment Digit 0[5]
g	HEX0_D[6]	PIN_AH28	Seven Segment Digit 0[6]

### Assignments > Pin Planner

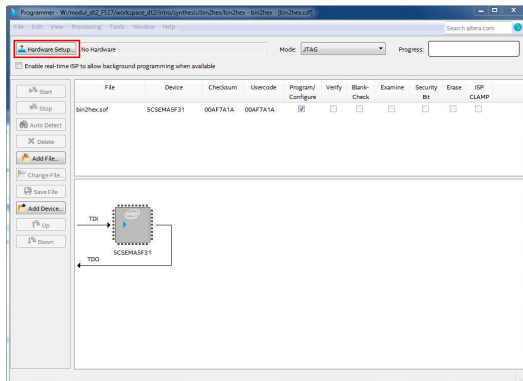


Nun müssen Sie die Synthese nochmals laufen lassen mit

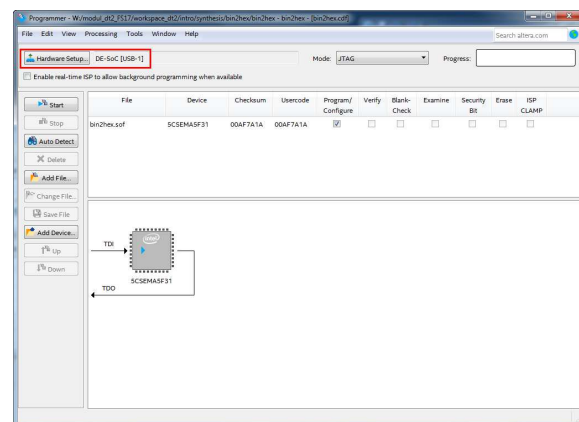
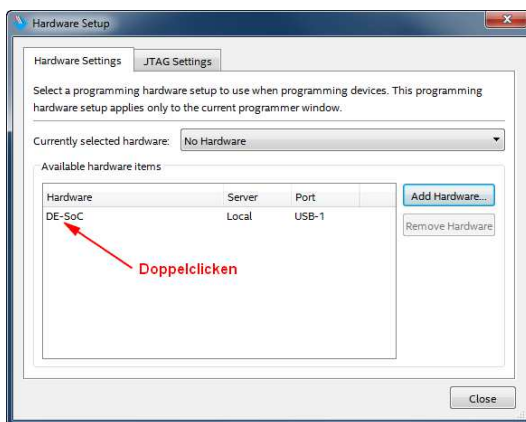
### Processing > Start Compilation

Schliessen Sie ein DE1-SoC-Board an die USB-Schnittstelle an und schalten Sie es ein. Laden Ihr Design auf den Altera-Baustein mit:

*Tools > Programmer*



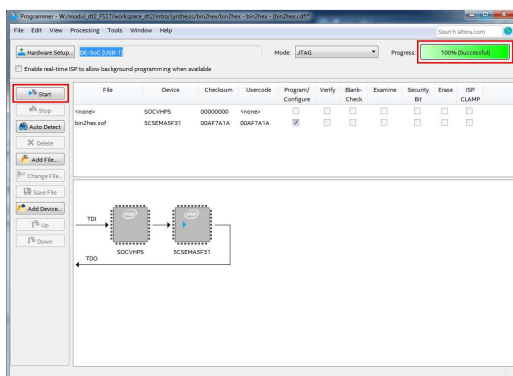
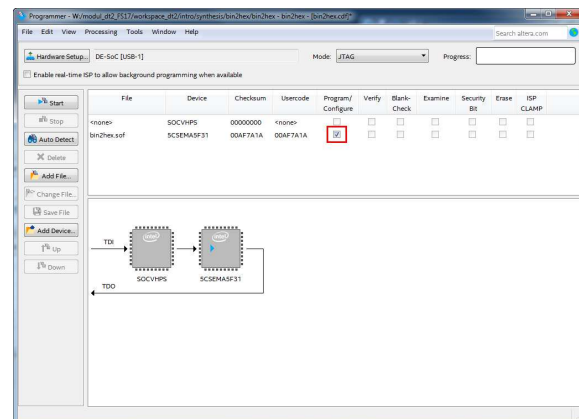
Da das DE1-SoC-Board noch nicht erkannt wurde ("No Hardware"), stimmt auch die Programmierkette noch nicht...



Schliessen Sie den Programmer und starten Sie in erneut mit

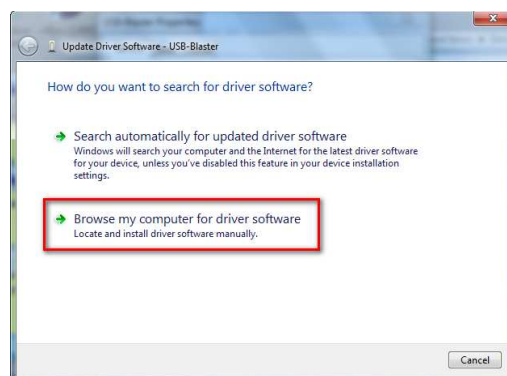
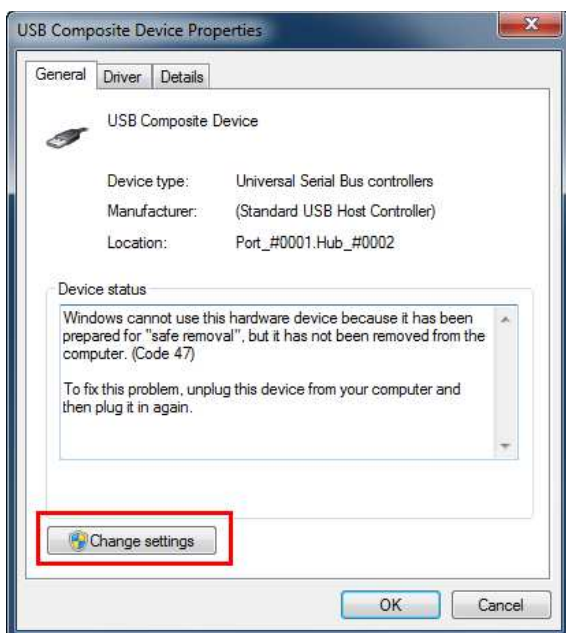
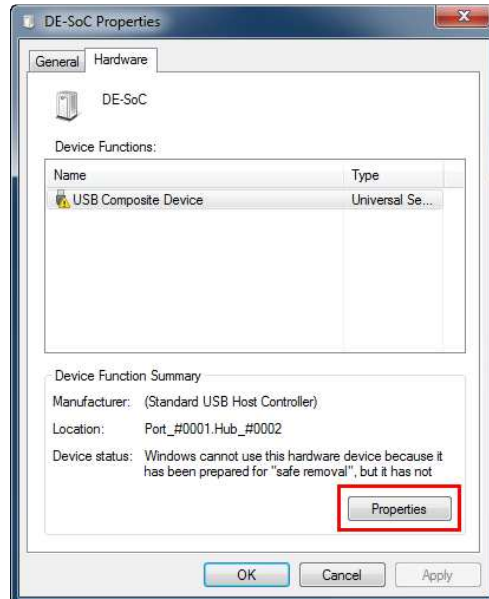
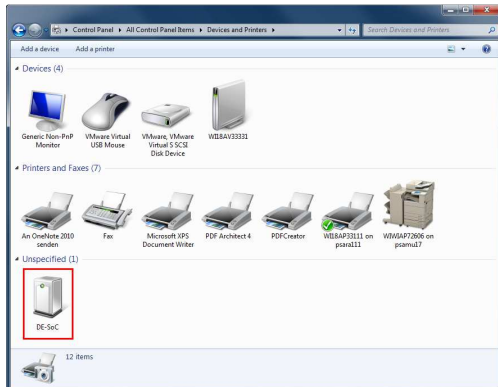
*Tools > Programmer*

Über die DE-SoC-USB Hardware wird jetzt das DE1-SoC-Board erkannt und die richtige Programmierkette definiert...



Nun können Sie mit den Schaltern SW0 - SW3 eine Binärkombination einstellen, die auf HEX0 hexadezimal dargestellt wird.

Falls der USB-Treiber noch nicht installiert ist, gehen Sie wie folgt vor:



Drivers sind unter

C:\intelFPGA\_lite\16.1\quartus\drivers