

# Aufgabe 2

## Design-Flow

- Blinkfrequenz =  $7,692307$  Hz

Die Periodendauer des Blinkens des Rechtecks soll laut Angabe 0,13 Sekunden betragen.

Um die Blinkfrequenz zu verändern, mussten wir in der Datei `vga_control_arc.vhd` die Konstante `HALFPERIOD` anpassen. Die Taktfrequenz des FPGA beträgt 25,175 MHz, was 25 175 000 Clock-Ticks entspricht.

$$\Rightarrow 25175000 \cdot 0,13 = 3272750$$

Da in `HALFPERIOD` die halbe Periodendauer definiert werden muss, lautet das Ergebnis  $3272750/2 = 1636375$ , dies entspricht binär 110001111100000010111.

Source-Code-Änderung in Zeile 49 der Datei `vga_control_arc.vhd`:

```
43: -- signal and constant declarations
44: signal  r_next, g_next, b_next  : std_logic;
45: signal  toggle_sig             : std_logic;
46: signal  toggle_counter_sig     : std_logic_vector(TOG_CNT_WIDTH-1 downto 0);
47: signal  toggle_next           : std_logic;
48: signal  toggle_counter_next    : std_logic_vector(TOG_CNT_WIDTH-1 downto 0);
49: constant HALFPERIOD          : std_logic_vector(TOG_CNT_WIDTH-1 downto 0) := "0000110001111100000010111";
```

In der Behavioral-Simulation sieht man, dass unser berechnetes `HALFPERIOD` zum richtigen Ergebnis führt (siehe Abbildung 2.1: Periodendauer von `d.toggle` beträgt 129928254400 ps  $\approx$  129,928 ms).

Da wir eine kurze Periodendauer hatten, haben wir `HALFPERIOD` nicht skaliert und über die volle Periode von 130 ms simuliert.

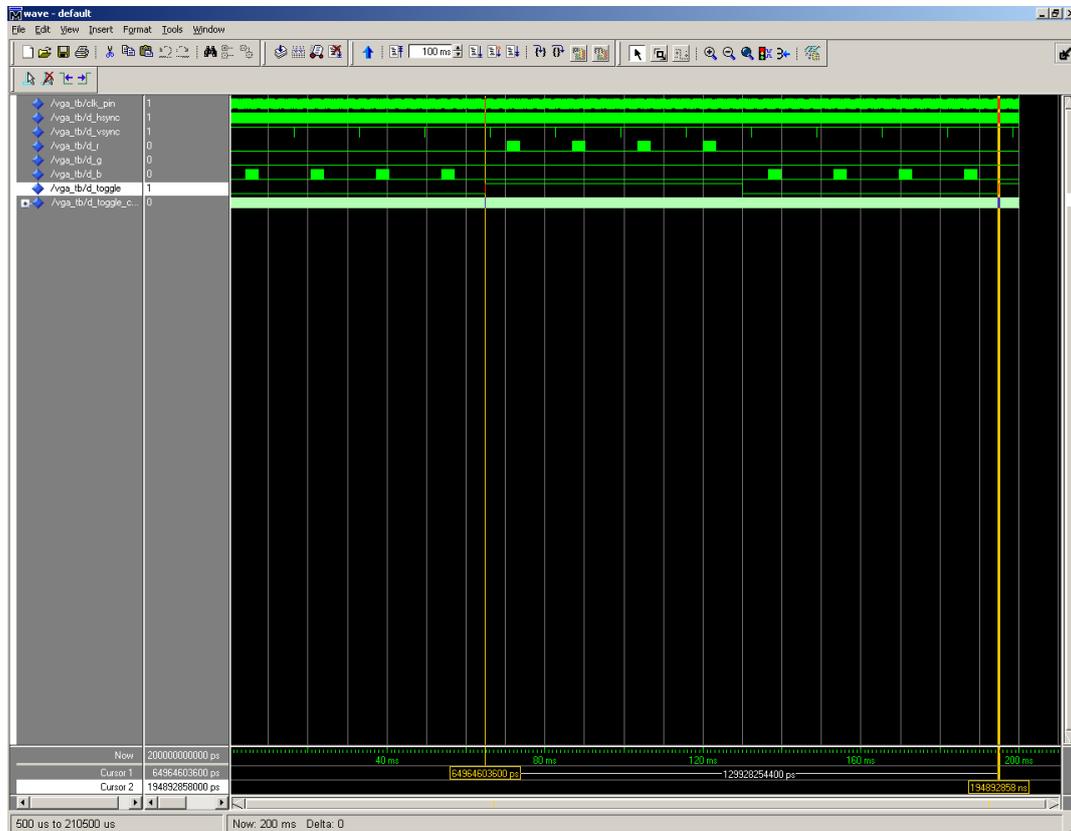


Abbildung 2.1: Behavioral-Simulation

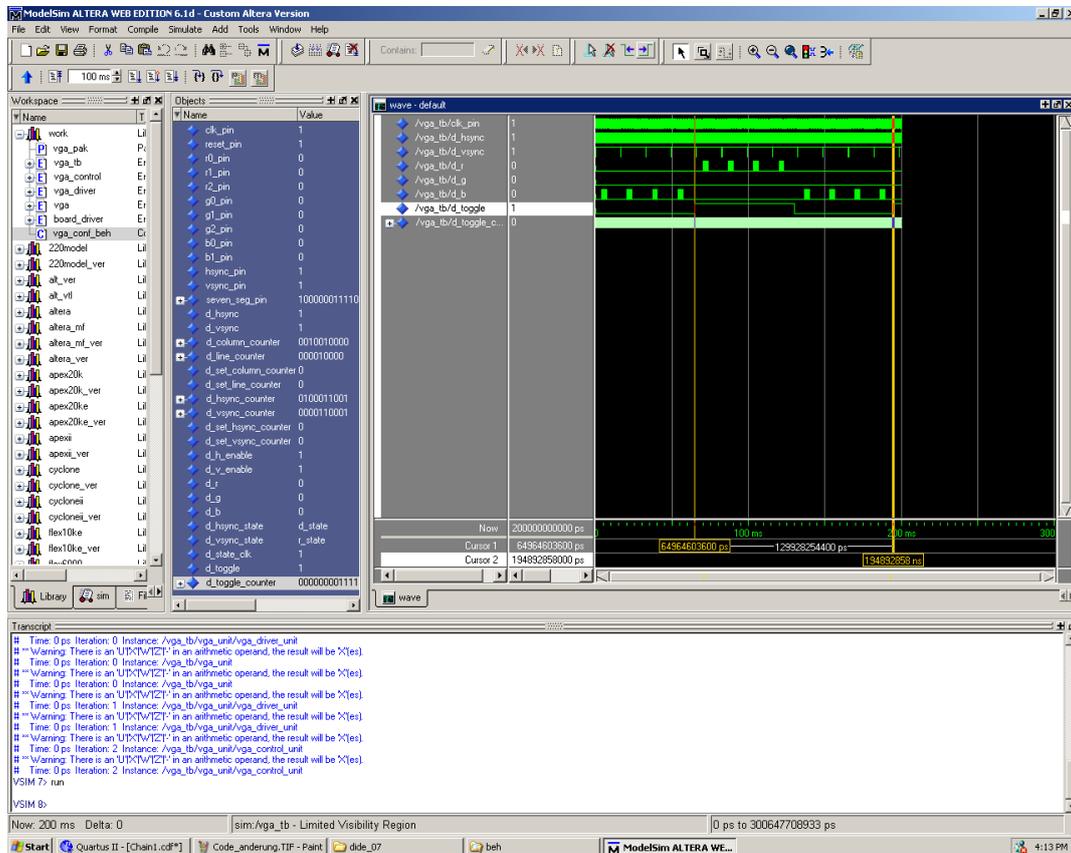


Abbildung 2.2: Behavioral-Simulation Übersicht

Nach erfolgreicher Behavioral-Simulation haben wir das Design synthetisiert. Da keine Errors oder Warnings auftraten, konnten wir mit der Pre-Layout-Simulation fortfahren.

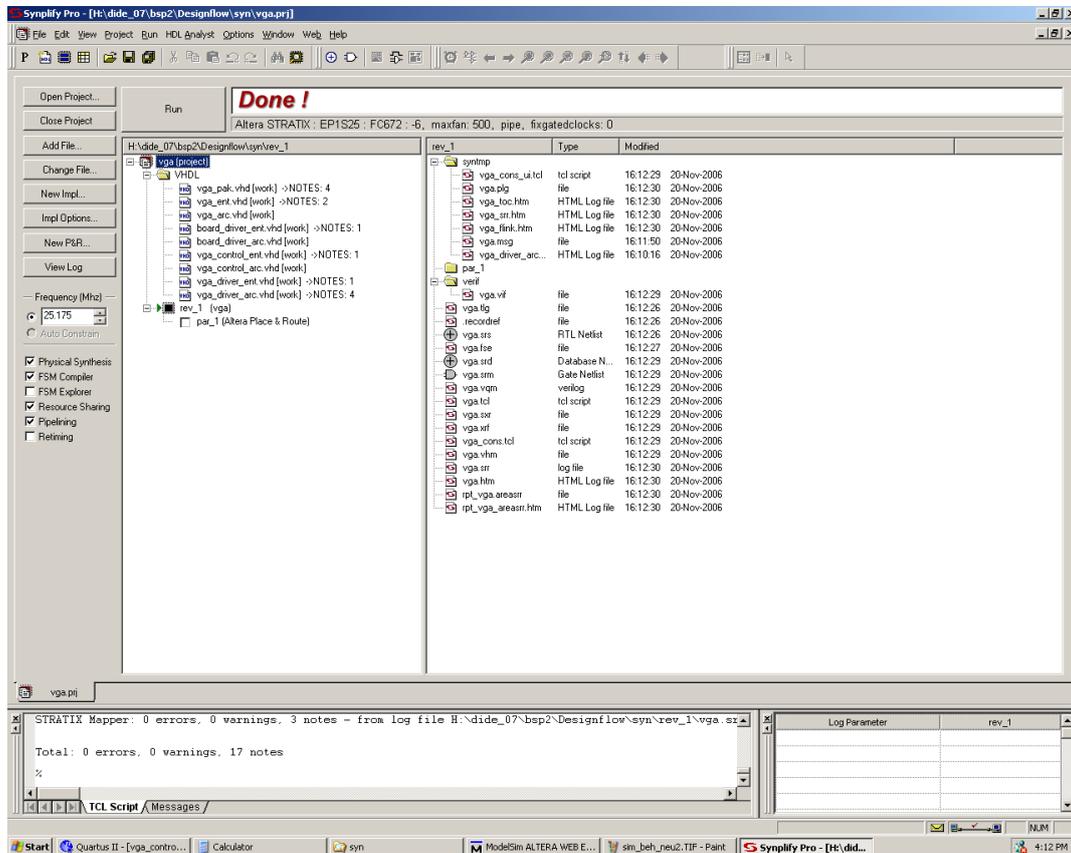


Abbildung 2.3: Synthese

Bei der Pre-Layout-Simulation mussten wir zeigen, dass der Toggle-Counter läuft. Somit mussten wir nur über einen Zeitraum von wenigen Millisekunden simulieren.

In Abbildung 2.4 sieht man, dass der Toggle-Counter (d.toggle.counter) zählt.

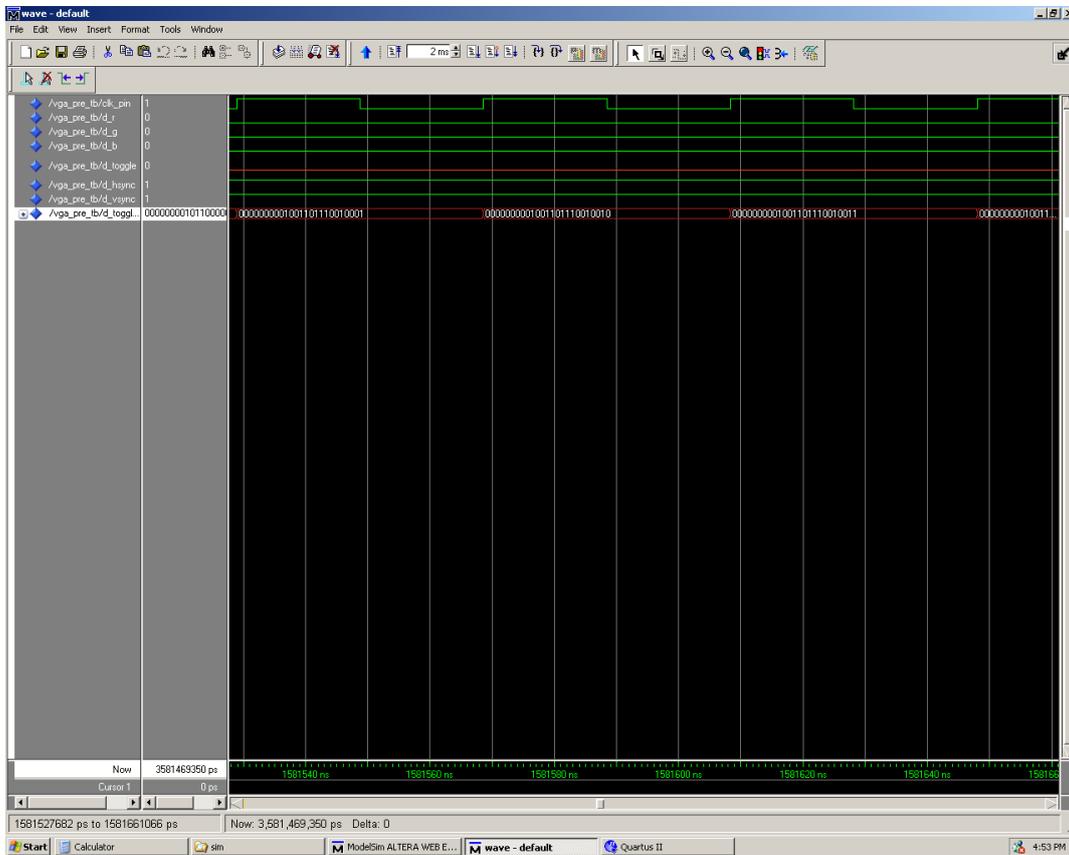


Abbildung 2.4: Pre-Layout-Simulation

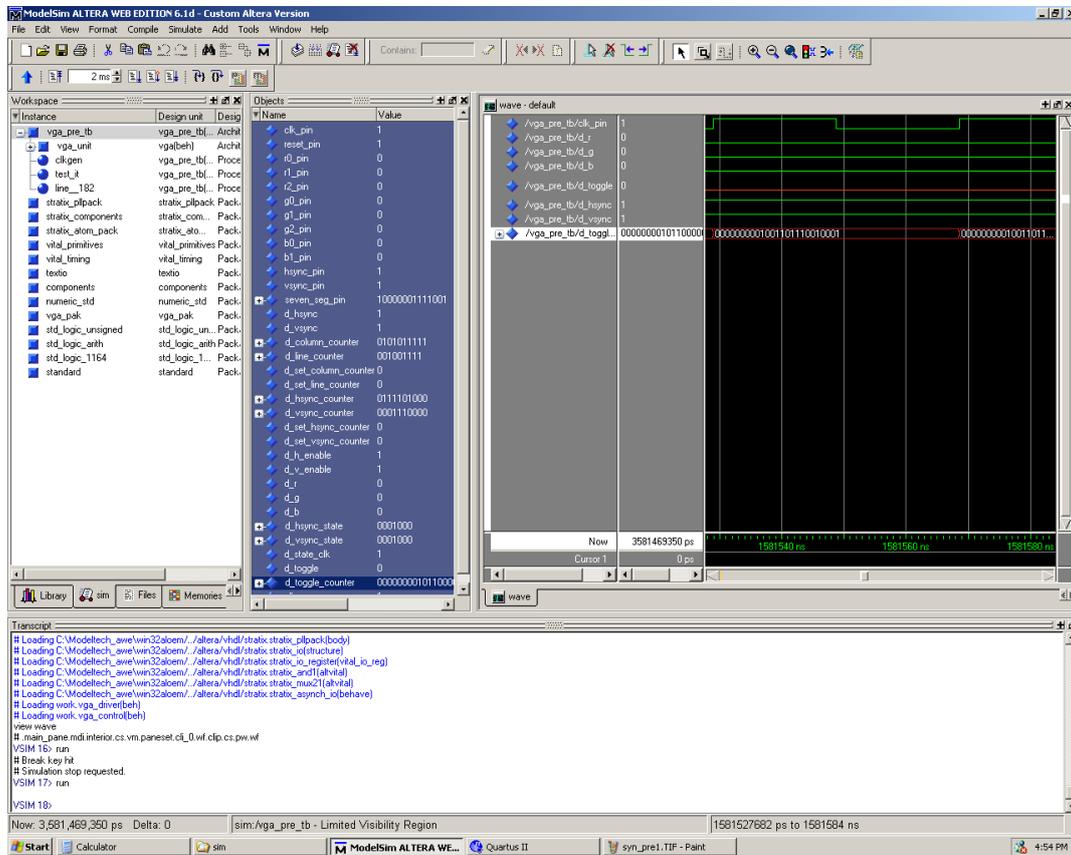


Abbildung 2.5: Pre-Layout-Simulation Übersicht

Bei der Post-Layout-Simulation mussten wir, wie bei der Pre-Layout-Simulation, nur zeigen, dass der Toggle-Counter läuft. Dass der Toggle-Counter zählt, sieht man in Abbildung 2.6

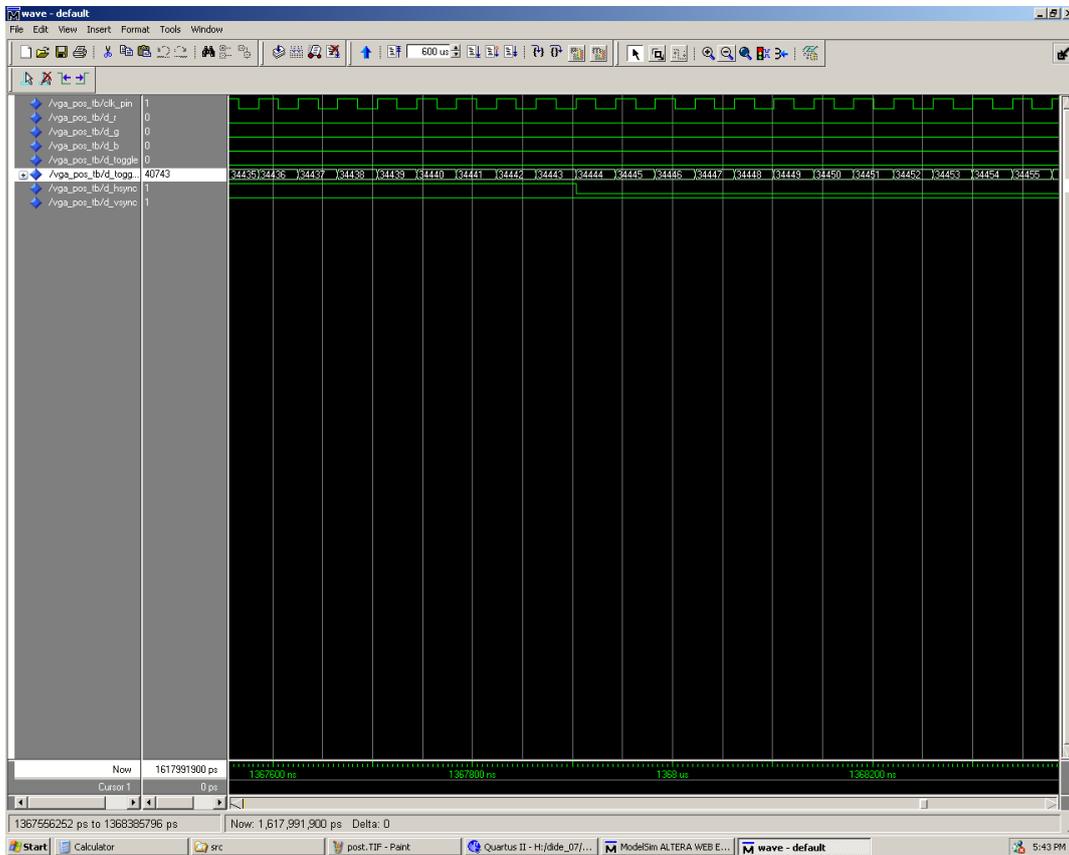


Abbildung 2.6: Post-Layout-Simulation

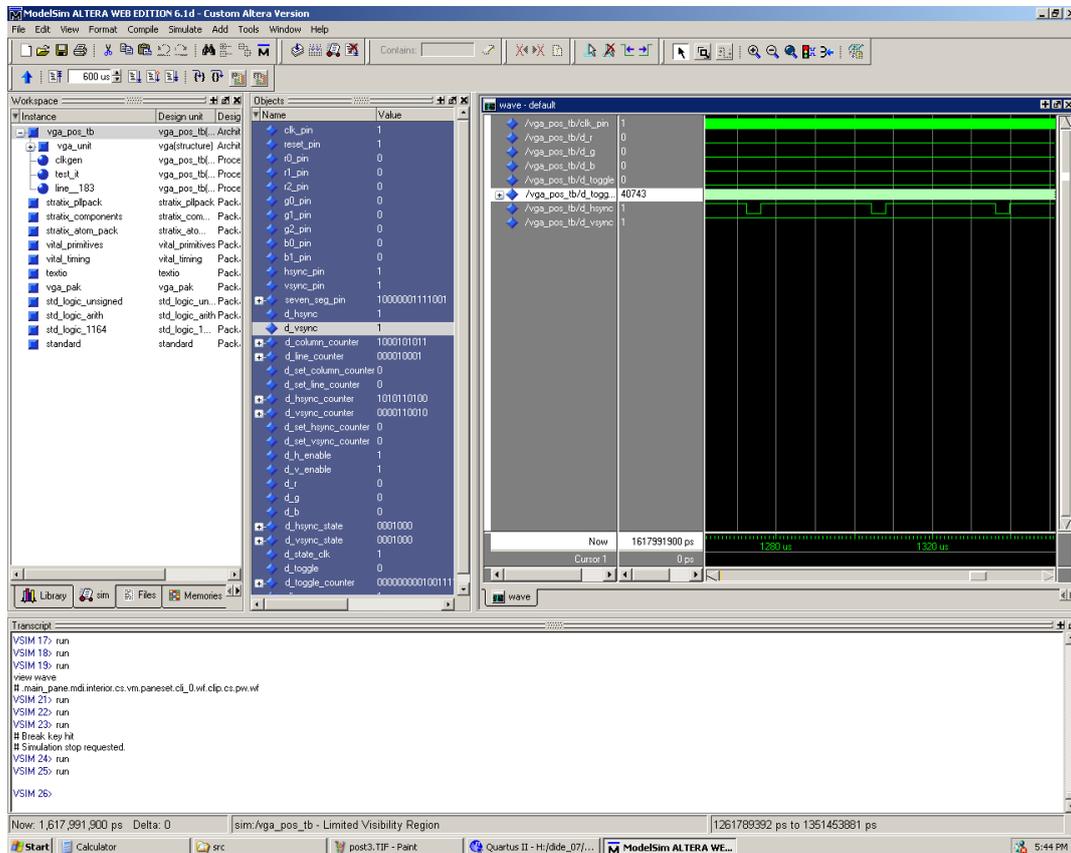


Abbildung 2.7: Post-Layout-Simulation Übersicht

Anschließend mussten wir das Place & Route noch einmal durchführen, um eine eigene Variante des Designs (mit PLL) für den Download zu erhalten. Nach diesem Vorgang wird ein Report angezeigt, in dem man die Auslastung des FPGA ablesen kann (siehe Abbildung 2.8). Die Auslastung beträgt 152/25660 (< 1%) logischen Elementen, 117/474 (25 %) Pins sowie 1/6 PLL.

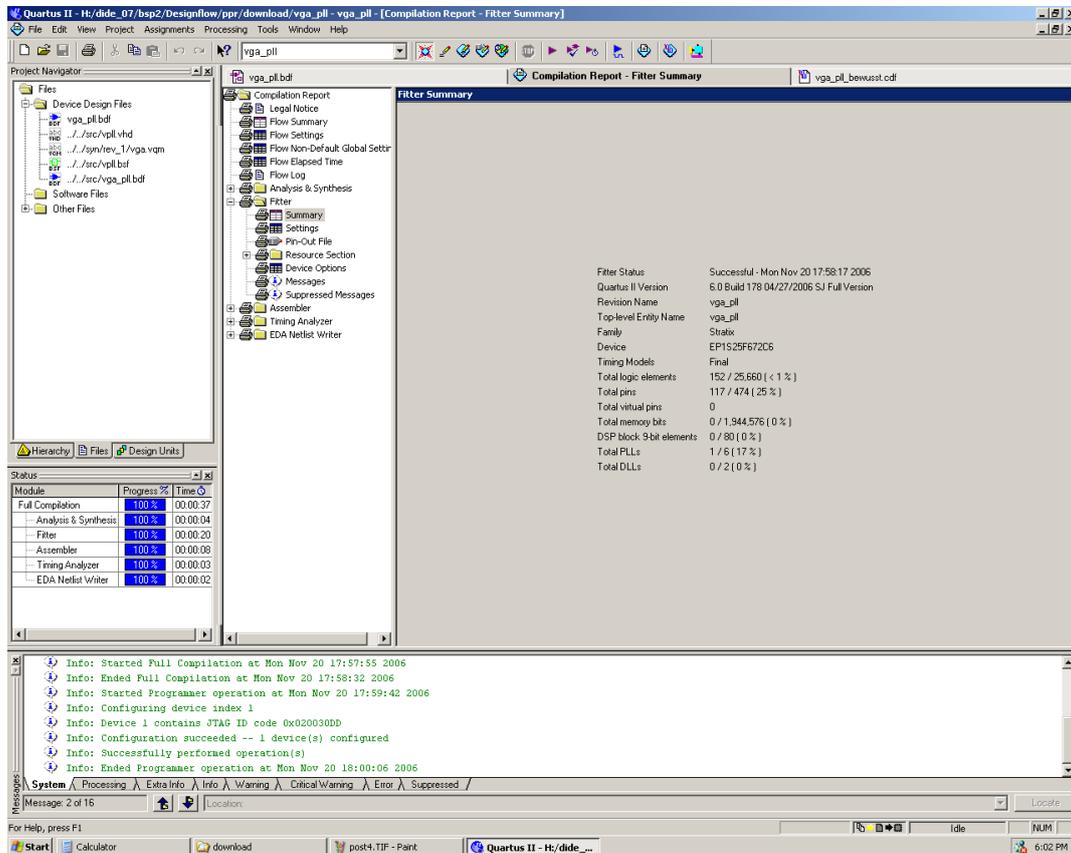


Abbildung 2.8: Auslastung des FPGA

Nun mussten wir unsere Blinkfrequenz am Logikanalysator überprüfen. Wir haben den Logikanalysator im State-Modus betrieben, weil wir Zustände messen wollten und in diesem Modus sehr einfache Trigger-Bedingungen formulieren konnten. Diese sind in Abbildung 2.9 ersichtlich. Auf dem Screenshot nicht ersichtlich sind die zu 1 und 2 analogen weiteren Bedingungen, sowie die abschließende Bedingung „Store no state“.

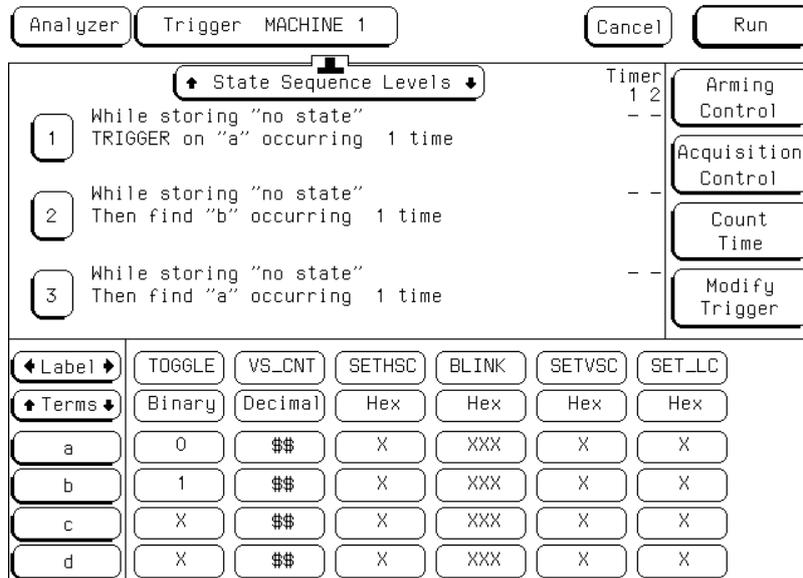


Abbildung 2.9: Logikanalysator - Trigger

In der Waveform sieht man, dass das Toggle-Signal arbeitet. Die anderen Signale sind hier wenig aussagekräftig, weil für sie die Abtastrate zu groß ist.

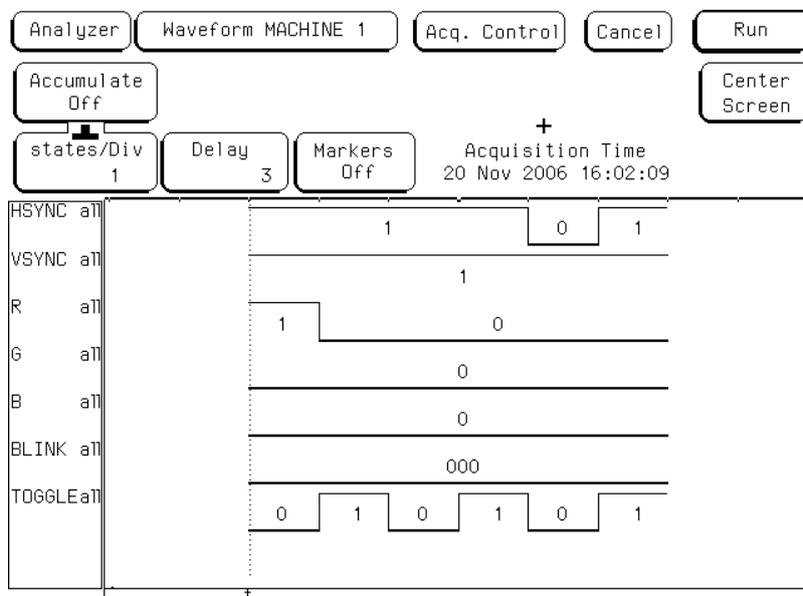


Abbildung 2.10: Logikanalysator - Waveform

In der Listenansicht sieht man, dass die Periodendauer 120,36 ms beträgt. Dies weicht vom errechneten und simulierten Wert von 130 ms ab, da das FPGA

mit einer Taktfrequenz von 27,175 MHz statt 25,175 MHz arbeitet.

Dies kann man durch nachrechnen verifizieren:  $\frac{2 \cdot 1636375}{27175000} = 0,12s$

Hätten wir den gesamten Prozess mit 27,175 MHz durchlaufen, wären wir auch bei den Simulationen auf die 120,36 ms gekommen. In diesem Fall hätten wir HALFPERIOD aber anders berechnen müssen, sodass wir auf die gewünschten 130 ms gekommen wären.

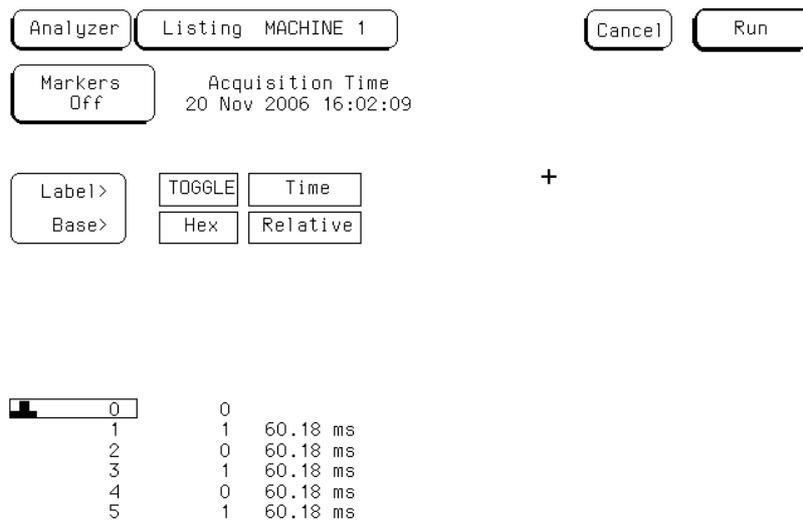


Abbildung 2.11: Logikanalysator - Liste