HW/SW-Codesign LU SS2011
Optimierungskriterien

Nachfolgende Kostenfunktion soll als Leitfaden bei der Optimierung der gegebenen Applikation dienen. Der Verbrauch von Ressourcen bringt Negetiv-Punkte, das Erreichen der Performance-Ziele bringt Plus-Punkte. Je höher also die Gesamtpunkte-Zahl, desto besser.

Zusatzfläche durch Extension Module

- LUT : -0,25 Pkt/Lut
- Register : -1 Pkt/Registerbit
- Memory : -0,1 Pkt/Memorybit
- Embedded Multiplier (9 bit element): -25 Pkt
- Embedded Multiplier (18 bit element): -50 Pkt

Die Zusatzfläche betrifft dabei nur die Module die zusätzlich zur von uns gelieferten Implementierung geschrieben wurden (d.h. exklusive SPEAR2 und bereits vorhandenen Modulen). Anzahl für Register, LUTs und Embedded Multiplier findet man in Quartus Compilation Report unter "*Analysis & Synthesis*" => "*Resource Utilization by Entity*" in folgenden Spalten aufgelistet:

LC Combinationals , LC Registers , Memory Bits , DSP 9x9 , DSP 18x18

Größe der SW-Applikation

- Codesize: -1 Pkt/Byte

Die Codesize der SW kann mittels im Disassemlby (main.txt) abgelesen werden. Dort werden die Basisdaten über die einzelnen Sections im Binary aufgelistet. Für die Größe des Programms interessiert uns die Text-Section:

```
Sections:
Idx Name          Size      VMA       LMA       File off  Algn  Flags
  0 .text         000049f8  00000000  80000000  00000658  2**0  ....
```

Performance

Die Punkteverteilung für die Laufzeit der Applikation ist so gestaltet, dass zumindest 0.10 fps erreicht werden müssen, um Plus-Punkte zu erhalten. Im Bereich zwischen 0.10 und 30 fps steigt die erreichbare Punktezahl dann linear. Ab 30 fps können durch Performance-Steigerungen keine Plus-Punkte mehr lukriert werden. In diesem Fall kann die Lösung nur mehr durch Vermeidung von Negativ-Punkten (d.h. geringerer Ressourcenverbrauch) verbessert werden. Die Punkte können gemäß folgender Funktion berechnet werden:

- f < 0.10 fps: 0 Pkt
- 0.10 <= f <= 30 fps: 3344.5* (f – 0.10) + 100000 Pkt
- f > 30 fps: 200000 Pkt

Genauigkeit

Eine höhere Performance der Applikation kann natürlich auch durch eine ungenauere Berechnung erzielt werden. Es gibt jedoch Punktabzüge, je ungenauer das Gesicht erkannt wird. Um die Genauigkeit zu ermitteln, verwenden wir drei Referenzbilder (Armstrong, Aldrin, Collins). Auf der LVA-Webseite findet sich einen Python-Skript (precisionCheck), mit dem der Grad der Überlappung zwischen dem Rechteck des Referenzbildes und dem Rechteck, welches euere Applikation berechnet hat, bestimmt werden kann.

Beispiel: SW-Lösung

Unsere Softwarelösung hat demnach folgende Punkteanzahl:

Zusätzliche Extension Module	0 LEs, 0 Membits, 0 Regbits, 0 Multiplier	0 Punkte
Codesize	62896 Bytes	-62896 Punkte
Geschwindigkeit	0,009 fps	0 Punkte
Genauigkeit	0.853399	-1466 Punkte
Gesamt		**-64362 Punkte**