

# Diskussion Design-Flow

---

## Erstellung abstrakter VP

Das Erstellen der EmuCpu an sich, fanden wir von der Idee her gut. Das gemeinsame Simulieren von Hardware- und Software-Komponenten kann damit gut nachvollzogen werden. Allerdings finden wir, dass dies (in dieser Übung) etwas besser umgesetzt werden könnte (siehe nachfolgende Punkte).

## Verfeinerung des Hardware-Modells für HLS

Dass das CC-Modell von HLS nicht 1:1 übernommen werden konnte, war schade, aber auch zu erwarten. Vielleicht wäre es besser gewesen, kein SystemC-CC-Modell zu entwickeln und stattdessen den Algorithmus selbst in (HLS) C++ zu entwickeln und diesen für die Simulation in ein LT-SystemC-Modell zu kapseln?

## Erweiterung der Firmware um Teile der API der Zielplattform und asynchrone Interrupts

Funktionen zu implementieren, deren Schnittstelle man vorher so noch nie gesehen hatte, war eher mühsam.

Ein "Zurückportieren" vom tatsächlichen Treiber auf leere Treiber für die Simulationsumgebung, würde sich für uns sinnvoller anhören.

Dass die Interrupts im endgültigen Design nicht vorhanden waren, war schade...

## High-Level-Synthese mit SystemC, Schnittstellen-Synthese und erzeugen eines IP-Cores

Es fühlt sich nicht so an, als würde SystemC WIRKLICH unterstützt werden. Solange das nicht der Fall ist, sehen wir keine Vorteile gegenüber "normalem" HLS mit C++, bei dem die Schnittstellensynthese funktioniert, es keine Reset-Probleme gibt und wobei das entwickeln auch wesentlich schneller von der Hand geht als mit SystemC.

## Migration auf die Zielplattform

Die Integration der fertigen HLS-IP in ein Block-Design verlief relativ reibungslos. Auch die Migration der Firmware lief ohne allzu große Probleme. Dass man erst in Hardware das fehlerhafte Reset-Verhalten des IP-Cores beobachten konnte, war nicht einfach herauszufinden.

Ob das vorausimplementieren von Firmware-Teilen viel Zeit sparte, ist fraglich.