



CSC1 Week 3

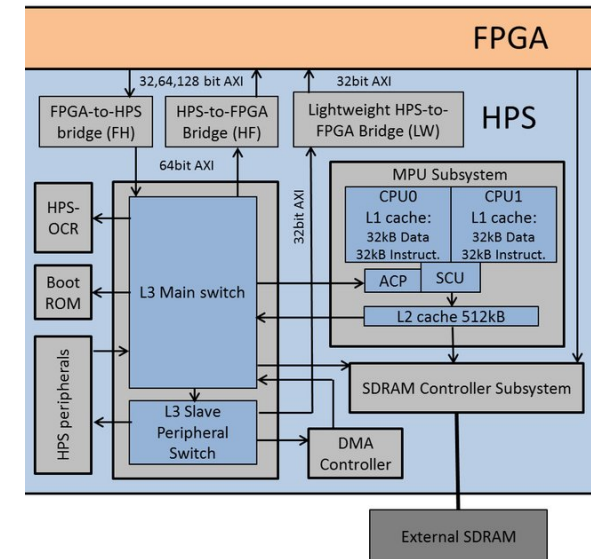
Je bent na het volgen van deze cursus in staat om:

- in VHDL een hardware module te ontwerpen en implementeren met een memory bus interface zodat deze module, vanuit een soft of hard core processor, memory mapped te programmeren is;
- een embedded systeem op een FPGA te ontwerpen en implementeren bestaande uit een soft core, software, bestaande hardware modules en zelf in VHDL geïmplementeerde hardware modules;
- op dit embedded systeem een RTOS toe te passen;
- een embedded systeem op een FPGA te ontwerpen en implementeren bestaande uit een **hard core**, software die draait onder Linux, bestaande hardware modules en zelf in VHDL geïmplementeerde hardware modules;
- te beslissen of bepaalde functionaliteit van een embedded applicatie beter op een soft core, op een hard core of in hardware geïmplementeerd kan worden;
- verschillende vormen van High Level Syntheses met de voor- en nadelen van deze vormen te benoemen.

Leerdoelen Week 3

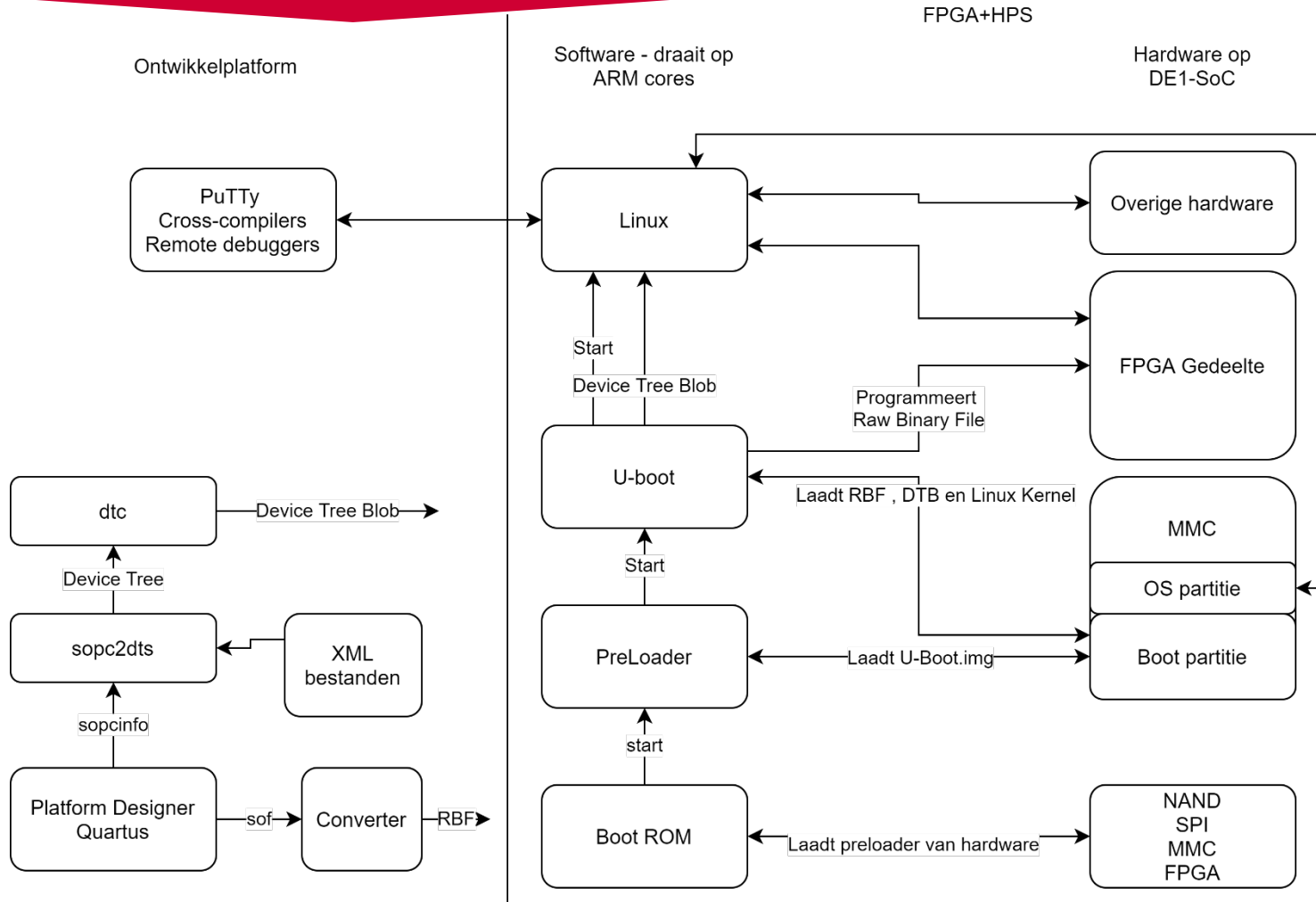
Leerdoelen week 3. Je leert hoe je:

- De ARM-Cortex A9 bare-metal kunt programmeren
- Een HPS systeem configureert
- Linux laat samenwerken met jouw HPS systeem
- Vanuit Linux met de FPGA communiceert
 - Kernel module
 - Character device



Overzicht processen

EMBEDDED SYSTEMS



1. Genereer HPS systeem in platform designer
2. Converteer .sof bestand naar .rbf bestand
3. Genereer device-tree-blob
 1. Converteer .sopcinfo en xml naar device-tree
 2. Compileer device-tree naar device-tree blob
4. Vervang .dtb en .rbf bestanden op bootpartitie
5. Boot het bordje

- Stap 1
 - User-space code om registers te beschrijven/lezen
 - adres = `mmap`(address lightweight bus,...)
 - `*(adres + pio_offset + register_offset)`
- Stap 2
 - Kernel-space module voor abstractie hardware
 - Character device
 - Benaderbaar vanuit user-space

Design space Exploration

Implementatie op Softcore of FPGA of Hardcore?

Aan de slag!

EMBEDDED SYSTEMS

Aan de slag met [Opdrachten_Week_3.pdf](#)

