

CSC10 Week 4: Design Space Exploration

Planning week 4 en 5

Week 4

- **Weekopdracht** 4: Design Space Exploration

Week 5

- **Weekopdracht** 5: High Level Synthesis
- Weekopdrachten 1 t/m 4 afmaken
- **Gastcollege**: 7-12 10:30 AP.B03.072 (zie mededeling Teams)
- Bedenken eindopdracht, zie:
 - https://bitbucket.org/HR_ELEKTRO/csc10/wiki/Opdrachten/Eindopdracht.pdf
 - https://bitbucket.org/HR_ELEKTRO/csc10/wiki/Opdrachten/Nakijkmodel.pdf
- Uiterlijk **dinsdag 12-12** goedgekeurd door docenten

Planning week 6 t/m T2

- Werken aan **eindopdracht**
- Docenten zijn beschikbaar om te helpen (in ieder geval tijdens de ingeroosterde uren)

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.

Manier van werken week 4 CSC10

- We maken gebruik van een **deelopdrachten**.
- Het uitvoeren van deze deelopdrachten is **geen doel** op zich!
- Zorg dat je **begrijpt** wat je doet en hou de **leerdoelen** in het oog.
- **Vraag** indien nodig de docent om extra uitleg.
- Hou een **logboek** bij, zodat je e.e.a. snel terug kunt zoeken.

Leerdoelen week 4. Je leert hoe je:

- een **audio-sigitaal** kunt **filteren** m.b.v. het DE1-SoC board door een FIR-filter te implementeren in C-code dat uitgevoerd wordt op een **Nios II processor**;
- een audio-sigitaal kunt filteren m.b.v. het DE1-SoC board door een FIR-filter te implementeren in hardware d.m.v. **VHDL**;
- een audio-sigitaal kunt filteren m.b.v. het DE1-SoC board door een FIR-filter te implementeren in de hard core **ARM processor** m.b.v. een programmeertaal onder Linux;
- de bovenstaande implementaties met elkaar kunt **vergelijken** op **uitvoersnelheid**.

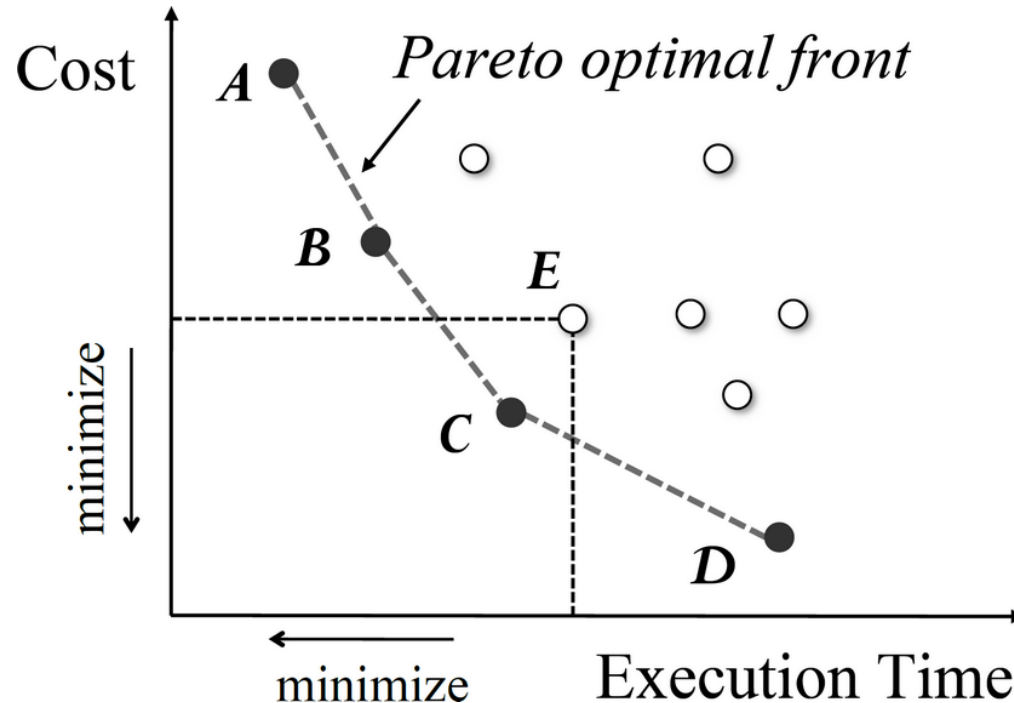
Verschillende **implementaties** van een embedded systeem kunnen vergeleken worden op relevante **eigenschappen**:

- prestatie (snelheid)
- energieverbruik
- kosten (productie- en/of ontwikkelkosten).
- ontwikkeltijd
- afmetingen
- gewicht
- aanpasbaarheid / uitbreidbaarheid

Design Space

Als de verschillende implementaties op n eigenschappen met elkaar vergeleken worden dan kunnen deze implementaties weergegeven worden in een n -dimensionale ruimte.

Voorbeeld $n = 2$



Aan de slag!

Aan de slag met [Opdrachten_Week_4.pdf](#)

