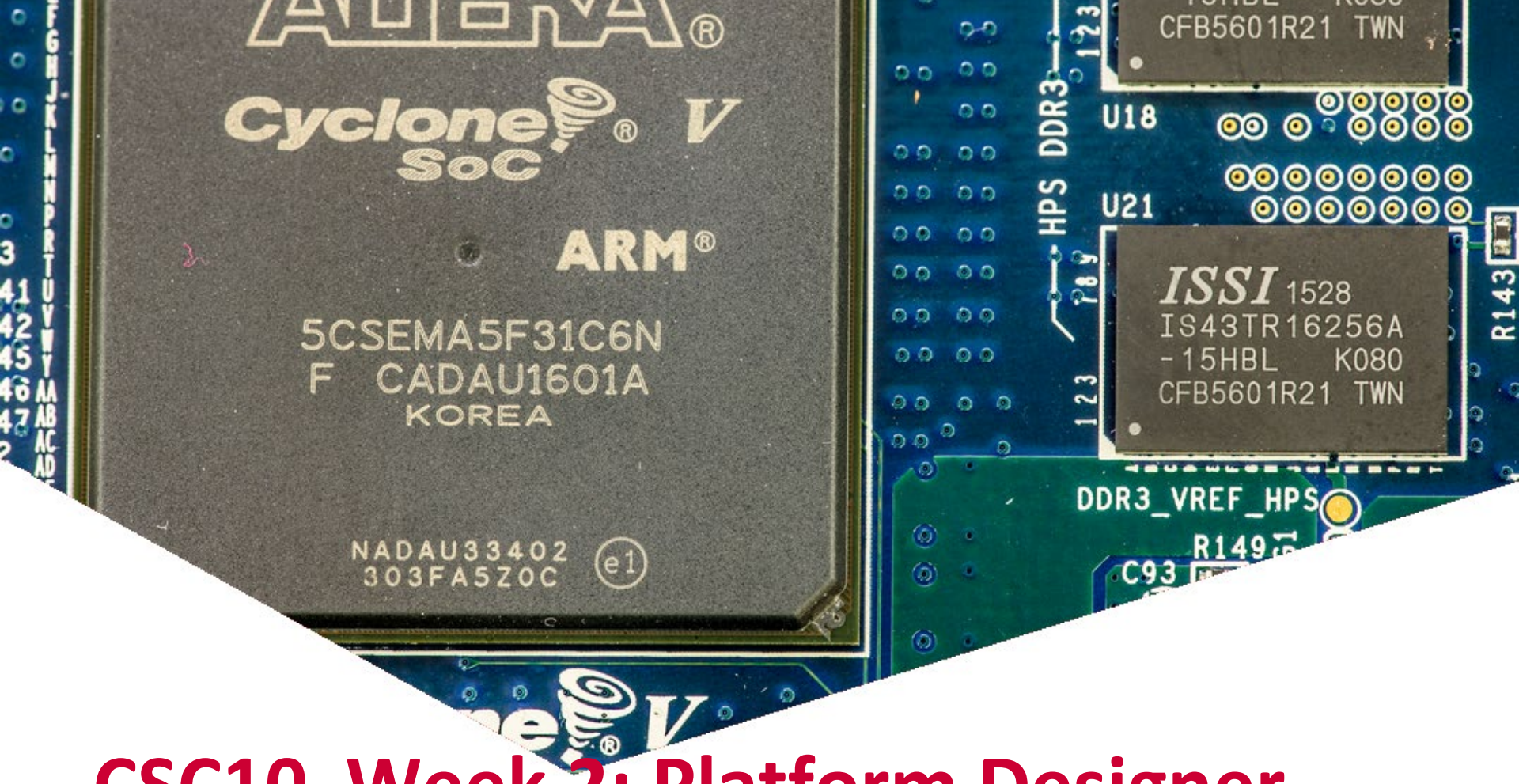


CSC10 Week 2: Platform Designer component en RTOS

Leerdoelen CSC10

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 CSC10

- We maken gebruik van door Intel beschikbaar gestelde tutorials.
- Het uitvoeren van deze tutorials 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 2

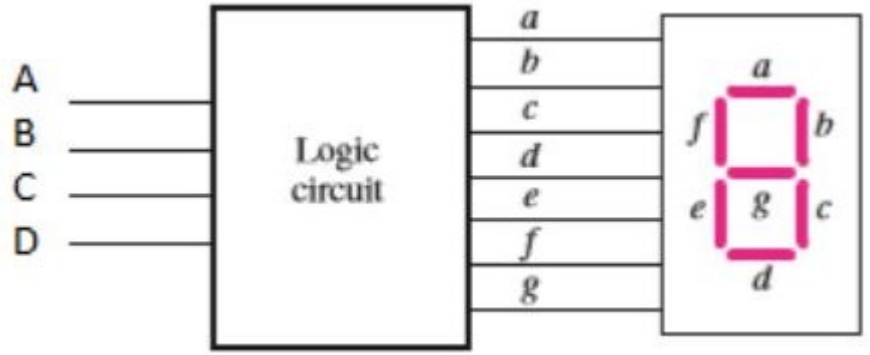
Leerdoelen week 2. Je leert hoe je:

- een eigen Platform Designer component kan definiëren met behulp van VHDL;
- deze component kan opnemen in een te bouwen systeem dat geïmplementeerd kan worden in een FPGA;
- hoe je deze component kan aansturen vanuit software met behulp van een RTOS.

Software versus Hardware

7-segment decoder

```
int hex_to_7_seg(int hex_digit) {
    if (hex_digit == 0x0) return 0x40;
    if (hex_digit == 0x1) return 0x79;
    if (hex_digit == 0x2) return 0x24;
    if (hex_digit == 0x3) return 0x30;
    if (hex_digit == 0x4) return 0x19;
    if (hex_digit == 0x5) return 0x12;
    if (hex_digit == 0x6) return 0x02;
    if (hex_digit == 0x7) return 0x78;
    if (hex_digit == 0x8) return 0x00;
    if (hex_digit == 0x9) return 0x18;
    if (hex_digit == 0xA) return 0x08;
    if (hex_digit == 0xB) return 0x03;
    if (hex_digit == 0xC) return 0x46;
    if (hex_digit == 0xD) return 0x21;
    if (hex_digit == 0xE) return 0x06;
    if (hex_digit == 0xF) return 0x0E;
    return 0x7F;
}
```

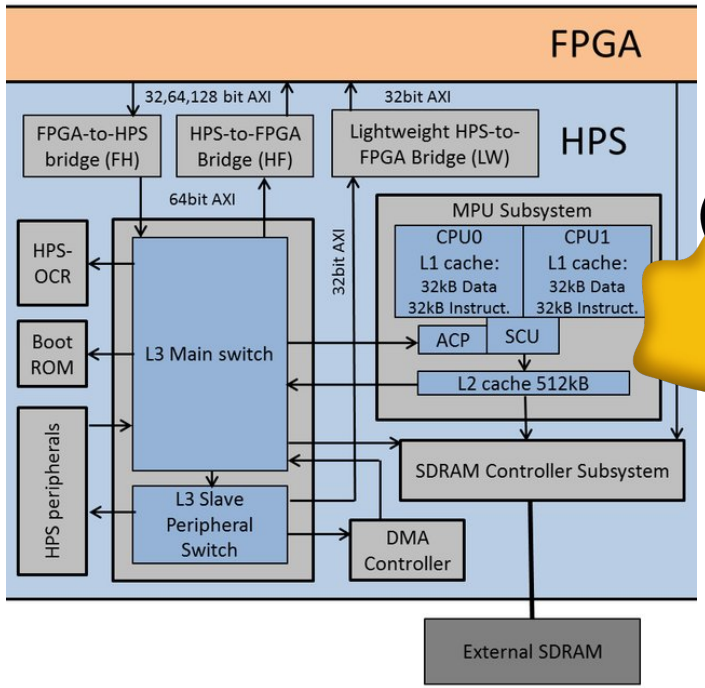


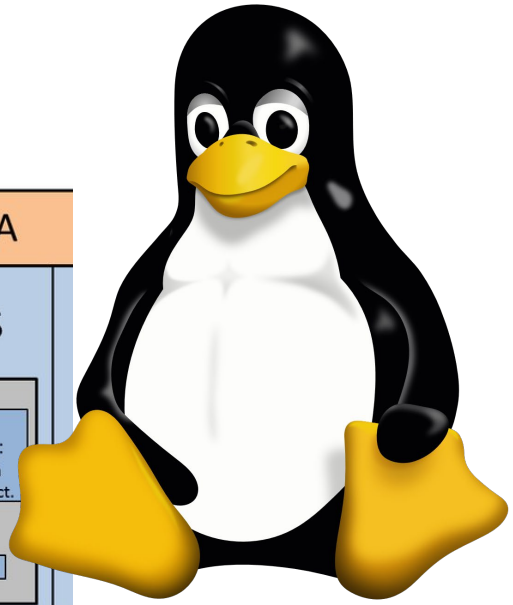

```
library ieee;
use ieee.std_logic_1164.all;

entity seven_segment_decoder is
    port (
        hex : in std_ulogic_vector (3 downto 0);
        ss : out std_ulogic_vector (6 downto 0)
    );
end entity;

architecture combinatorial of seven_segment_decoder is
begin
    with hex select ss <=
        not "0111111" when "0000", -- 0
        not "0000110" when "0001", -- 1
        not "1011011" when "0010", -- 2
        not "1001111" when "0011", -- 3
        not "1100110" when "0100", -- 4
        not "1101101" when "0101", -- 5
        not "1111101" when "0110", -- 6
        not "0000111" when "0111", -- 7
        not "1111111" when "1000", -- 8
        not "1101111" when "1001", -- 9
        not "1110111" when "1010", -- A
        not "1111100" when "1011", -- b
        not "0111001" when "1100", -- C
        not "1011110" when "1101", -- d
        not "1111001" when "1110", -- E
        not "1110001" when "1111", -- F
        not "0000000" when others;
end architecture;
```

Voor- en nadelen?

Volgende week…

De dual core ARM-Cortex A9 hard core gebruiken en (o.a.) programmeren met Linux.





Aan de slag!

Aan de slag met [Opdrachten_Week_2.pdf]