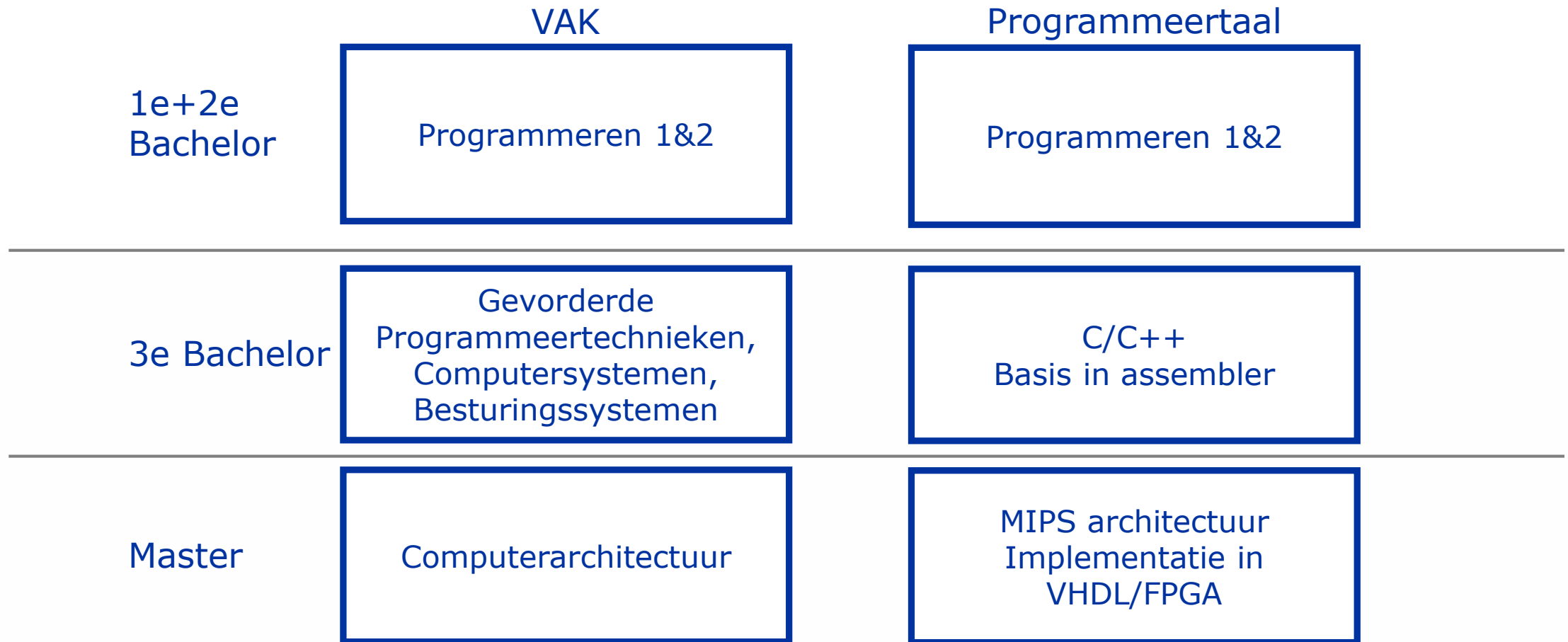


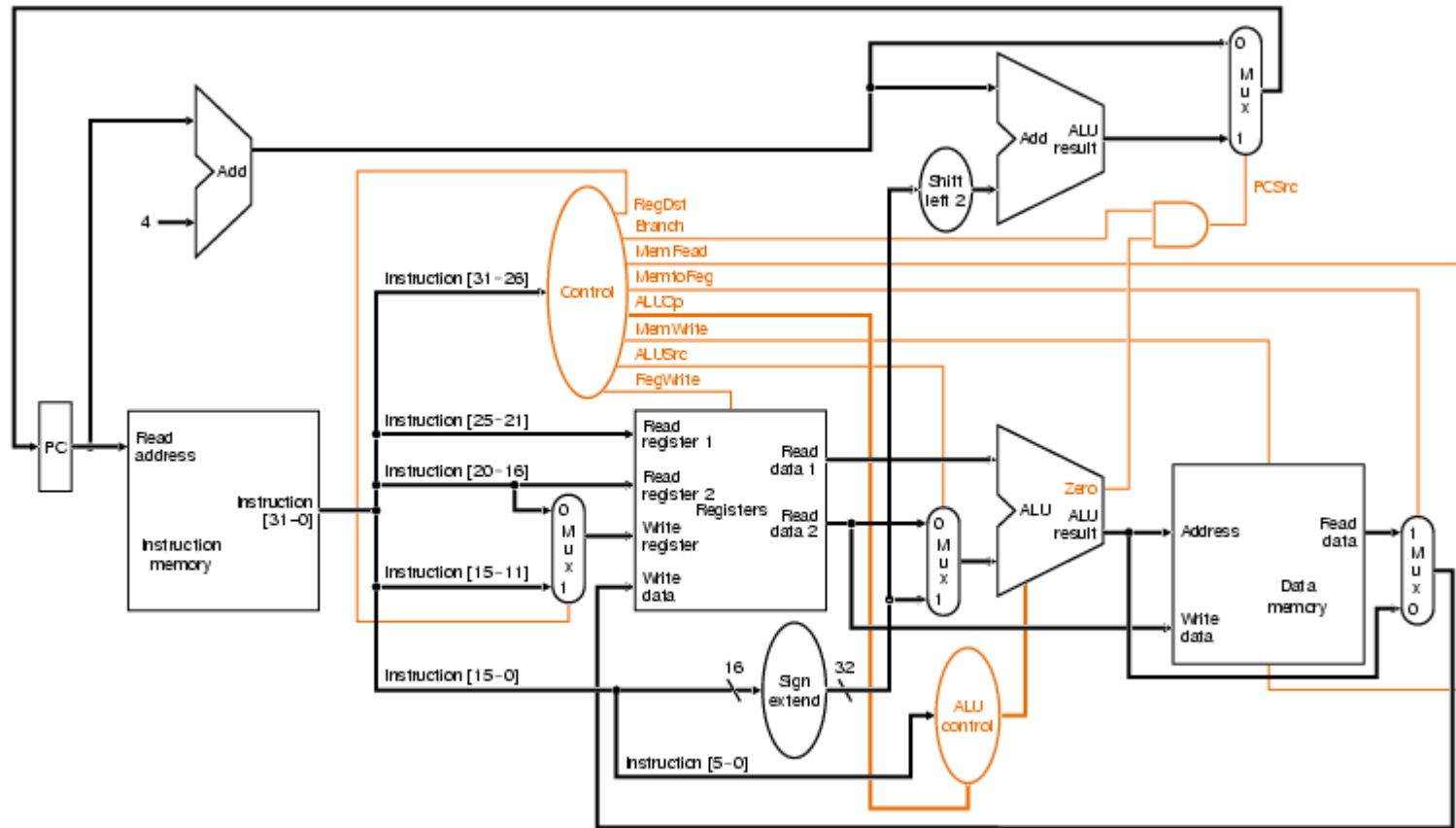
COMPUTERARCHITECTUUR

MIPS PROCESSOR ONTWERP & IMPLEMENTATIE (2019-2020)

Jurgen Vandendriessche
(jurgen.vandendriessche@vub.be)



BASISARCHITECTUUR MIPS



VHDL VS. ASSEMBLER

VHDL

Very High Speed Integrated Circuit **H**ardware
Description **L**anguage

Opbouw structuur MIPS Processor

Assembler code wordt in Instruction Memory
bewaard en uitgevoerd

Combinatorische logica & Finite State Machines

Assembler

Sequentieel programma dat door de MIPS
Processor uitgevoerd wordt

Wordt eerst naar binair/hex gecompileerd
alvorens in de Instruction Memory bewaard te
worden

ONTWERP EN IMPLEMENTATIE

VHDL/Ontwerp

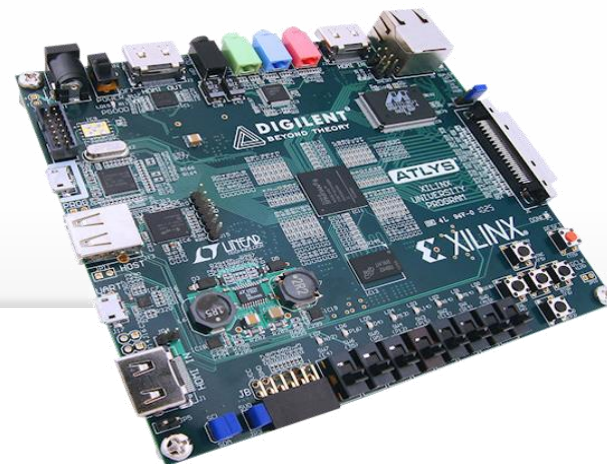
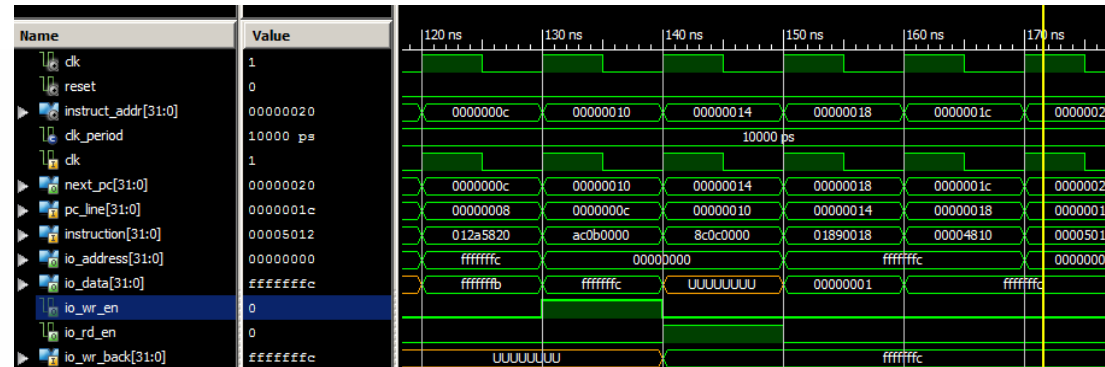


Simulatie/Verificatie



FPGA/Implementatie

```
architecture Behavioral of PC is |
  signal RESET_TRIGGER:STD_LOGIC;--<='0';
begin
  -- the setPC process
  setPC:process(Clk,PC_IN,RESET_TRIGGER)
  begin
    if (rising_edge(Clk)) then
      if (RESET_TRIGGER='1') then
        PC_OUT <= (others=>'0');
      else
        PC_OUT <= PC_IN;
      end if;
    end if;
  end process setPC;
```



FEATURES

DMA Controller

Verplaatsen van een aantal bytes van één geheugenlocatie naar een andere,
bv: van UART Controller naar Data geheugen

UART Communicatie

Eenvoudige communicatie tussen 2 devices
Memory Mapped: werken via adresseerbare registers

Instruction Memory 'herflashen'

Herprogrammeren van de controller zoals o.a. een PIC, MSP430,...
Nieuwe code komt in Instruction Memory

PROJECT

Groepswerk

Bouw de MIPS processor in groepen van 2 personen

Extra's

Implementeer enkele (minstens één) 'features'

Basiscomponenten

Enkele (VHDL)componenten worden ter beschikking gesteld

Doelstelling

Must have: werkende processor!

Korte presentatie waarop structuur en simulaties aan bod komen