

# *FPGA's* *Field Programmable Gate Array*

Wat is het?

Wat kan je ermee?

Rob Alblas  
Werkgroep Kunstmanen

# *Overzicht*

- FPGA versus processor
- Interne opbouw
- Hoe te ontwerpen
- Benodigde ontwerpsoftware

# *Een FPGA is NIET...*

- Iets als een microprocessor.



# *FPGA versus processor*

- Processor
  - draait een programma af
  - Seriële verwerking

# *FPGA versus processor*

- Processor
  - draait een programma af
  - Seriële verwerking
- FPGA:
  - Na initialiseren (“spanning erop”): gedraagt zich als een “bak digitale IC's”
  - Parallele verwerking

## *FPGA: ontstaan*

- ASIC: Application Specific IC
  - Alle transistoren/logische poorten vrij te plaatsen
  - Start-kosten: heel duur (orde grootte M\$)
  - Heel grote aantallen: goedkoop



## *FPGA: ontstaan*

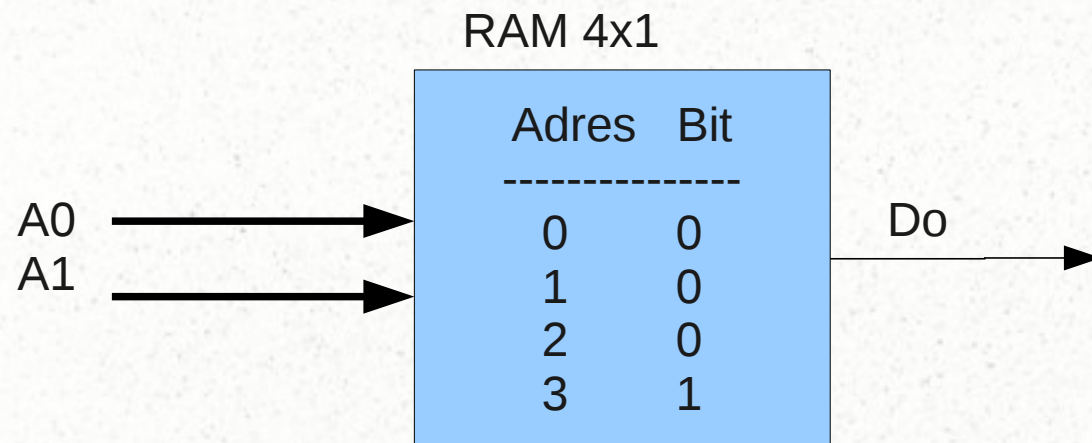
- ASIC: Application Specific IC
  - Alle transistoren/logische poorten vrij te plaatsen
  - Start-kosten: heel duur (orde grootte M\$)
  - Heel grote aantallen: goedkoop
- Gate Array:
  - Transistoren liggen vast
  - Doorverbindingen worden door ontwerp bepaald
  - Start-kosten: veel lager
  - Heel grote aantallen: duurder dan ASIC

# *FPGA*

- Field Programmable Gate Array
  - Als Gate Array, maar geheel “in het veld” te programmeren
  - Startkosten: nihil
  - Grote aantallen duur

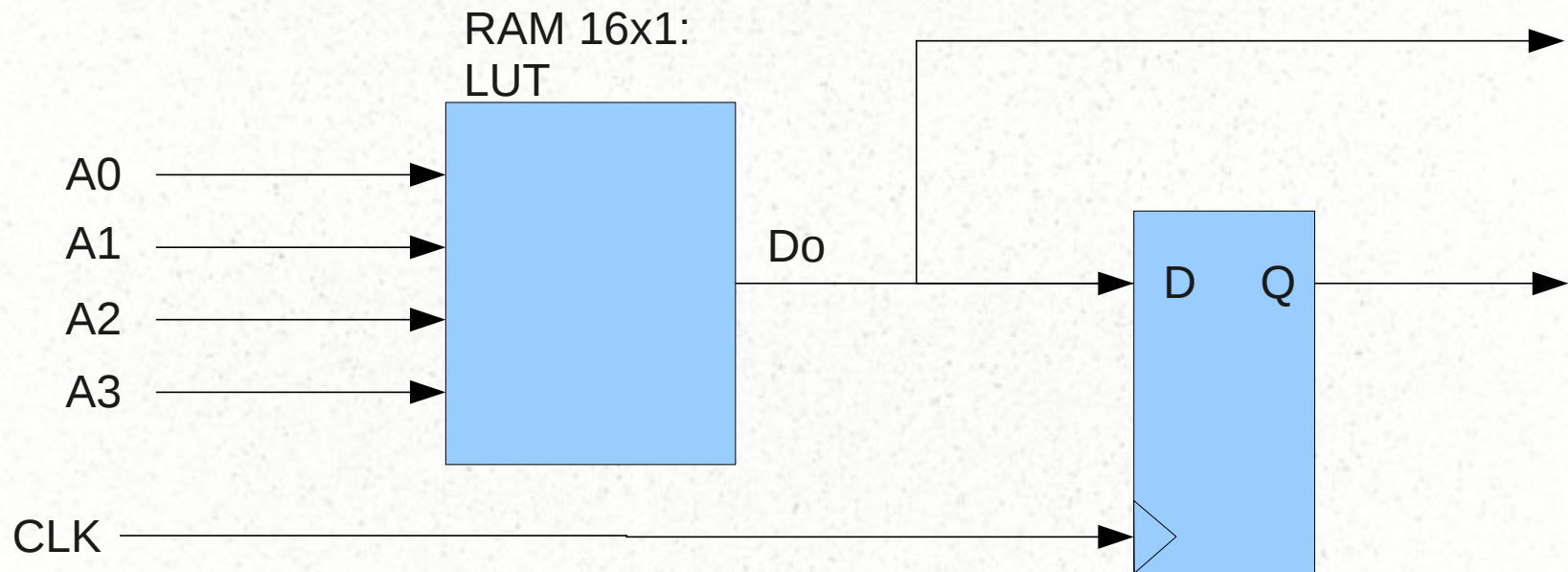


# *FPGA: een programmeerbare cel*

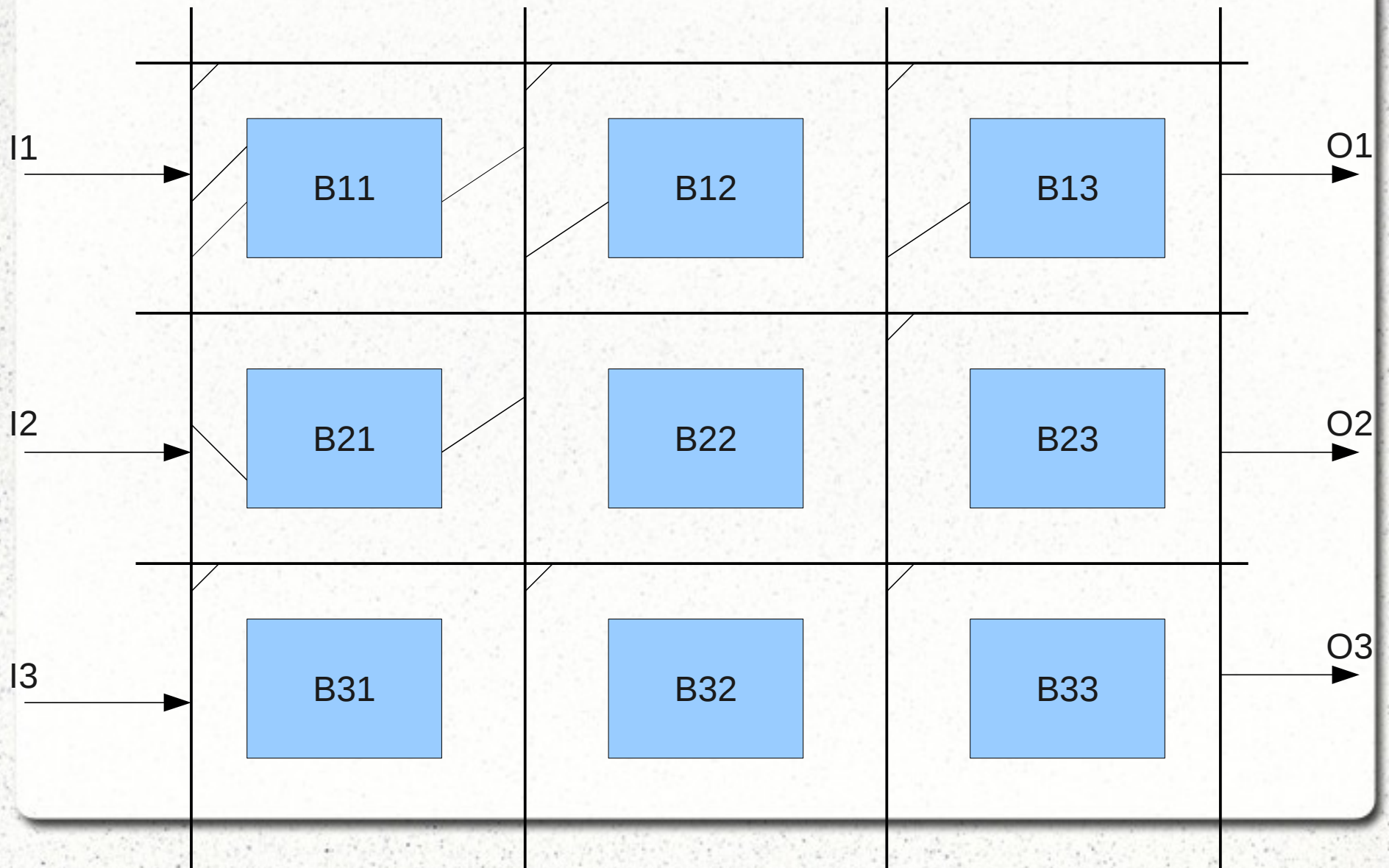


# *FPGA: een programmeerbare cel*

CLB



# *FPGA: opbouw*





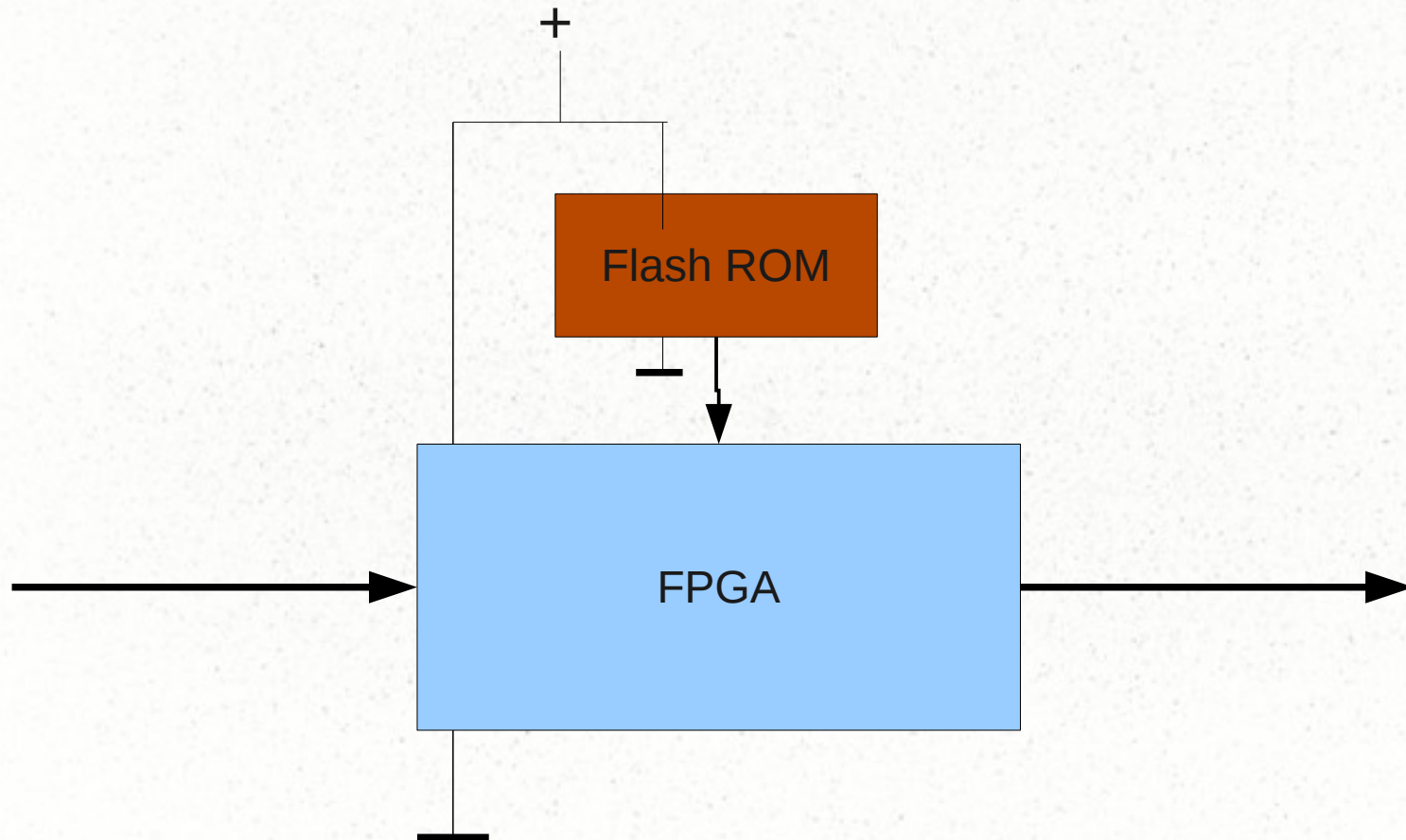
# *FPGA: IO*

- Pinnen programmeerbaar als in/uitgang, en:
  - pull-up
  - TTL-niveau
  - Differentieel
  - Enz.

# *FPGA: speciale blokken*

- PLL
- DCM:ingangsklok x2...32 /1...32
- Multipliers
- RAM
- SERDES; Serieel => parallel en omgekeerd
  - Gb/s
  - clockrecovery

# *FPGA: hardware*



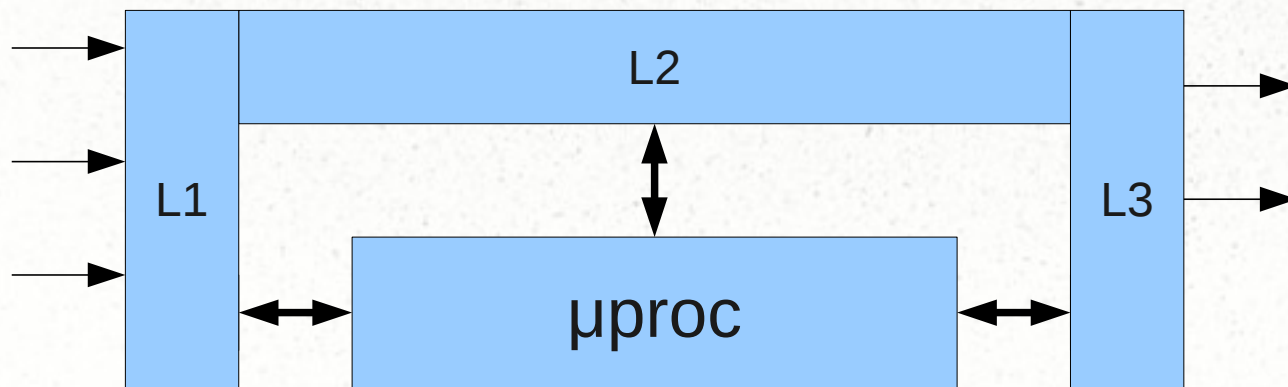


# *Xilinx FPGA's*

- Spartan3: XCS500
  - 10.000 CLB's (LUT's + flipflops)
  - 60 pinnen
- Kintex:
  - Tot 1,2 M CLB's
- Virtex:
  - 2 M CLB's
  - 2,8 Tb/s

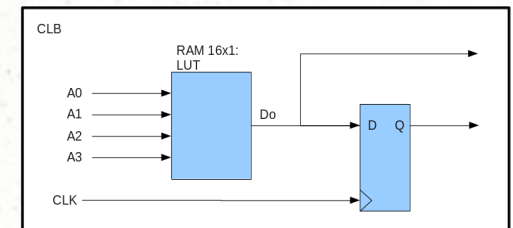
# *FPGA: embedded cores*

- Een FPGA met bv.:
  - 8051 core
  - Digitale logica



# *FPGA: ontwerpen*

- Schematische invoer
- Synthesizer: vertaling naar CLB's
- Mapper:
  - bepalen welke CLB's
  - onderlinge doorverbinding
- Vertaling naar bit-file
- Programmeren van de PROM





# *FPGA: invoer*

- Schematische invoer... 10000 FF's?
  - Beter van niet...
- Hogere programmeertaal:
  - VHDL
  - Verilog
  - (C++)

# *Een teller (1)*

```
IF cnt < 15 THEN
  cnt  <= cnt+1;
  puls <= '0';
ELSE
  cnt  <= 0;
  puls <= '1';
END IF;
```

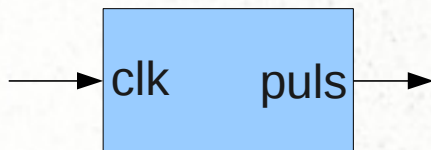
# *Een teller (2)*

```
PROCESS
BEGIN
    WAIT UNTIL clk='1';
    IF cnt < 15 THEN
        cnt <= cnt+1;
        puls <= '0';
    ELSE
        cnt <= 0;
        puls <= '1';
    END IF;
END PROCESS;
```



# Een teller (3)

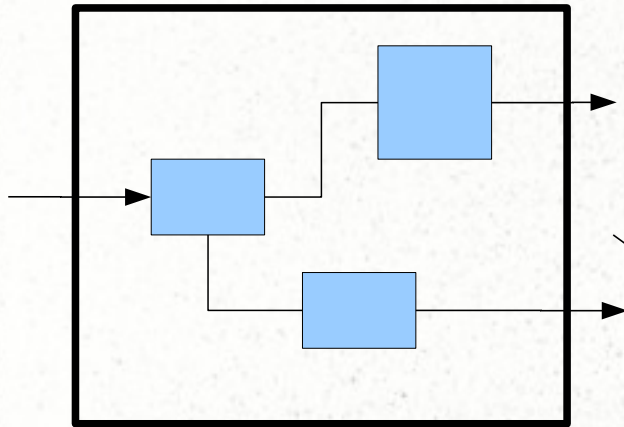
```
ENTITY tel IS
  PORT
  (
    clk   : in  std_logic;
    puls  : out std_logic
  );
END ENTITY tel;
```



```
ARCHITECTURE rtl OF tel IS
  SIGNAL cnt: integer RANGE 0 to 15;
BEGIN
  PROCESS
  BEGIN
    WAIT UNTIL clk='1';

    IF cnt < 15 THEN
      cnt <= cnt+1;
      puls <= '0';
    ELSE
      cnt <= 0;
      puls <= '1';
    END IF;
  END PROCESS;
END rtl;
```

# *Hiearchische opbouw*



```
BEGIN MODEL test
```

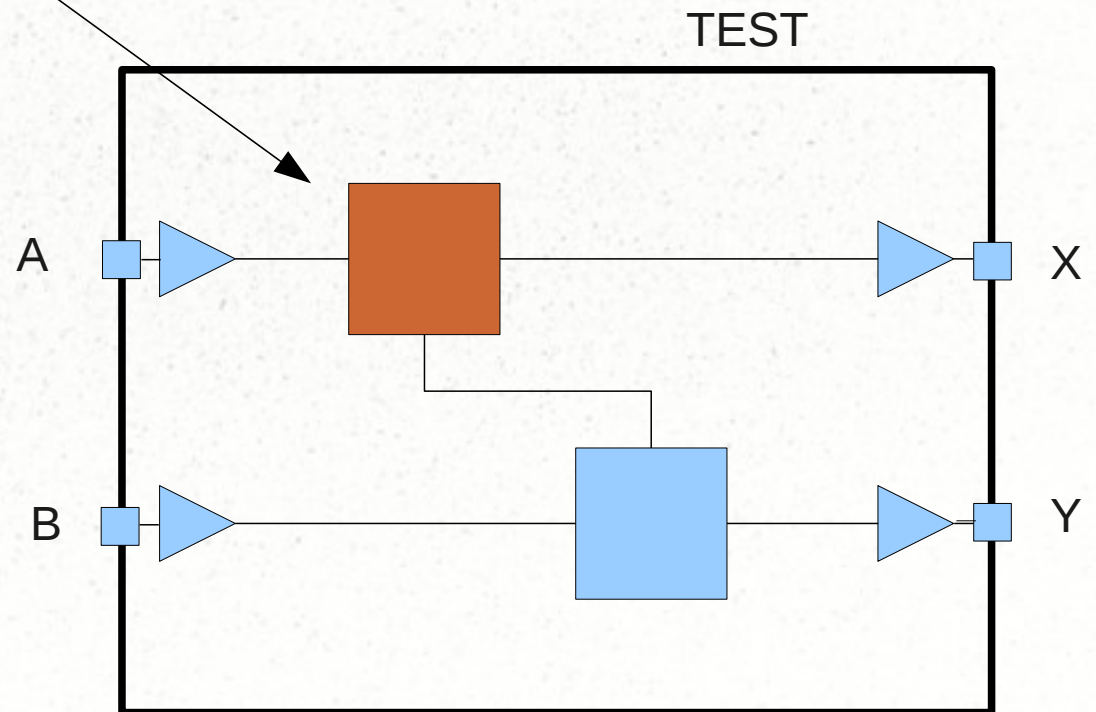
```
PIN "A" loc=P36
```

```
PIN "B" loc=P83
```

```
PIN "X" loc=P12
```

```
PIN "Y" loc=P44
```

```
END
```



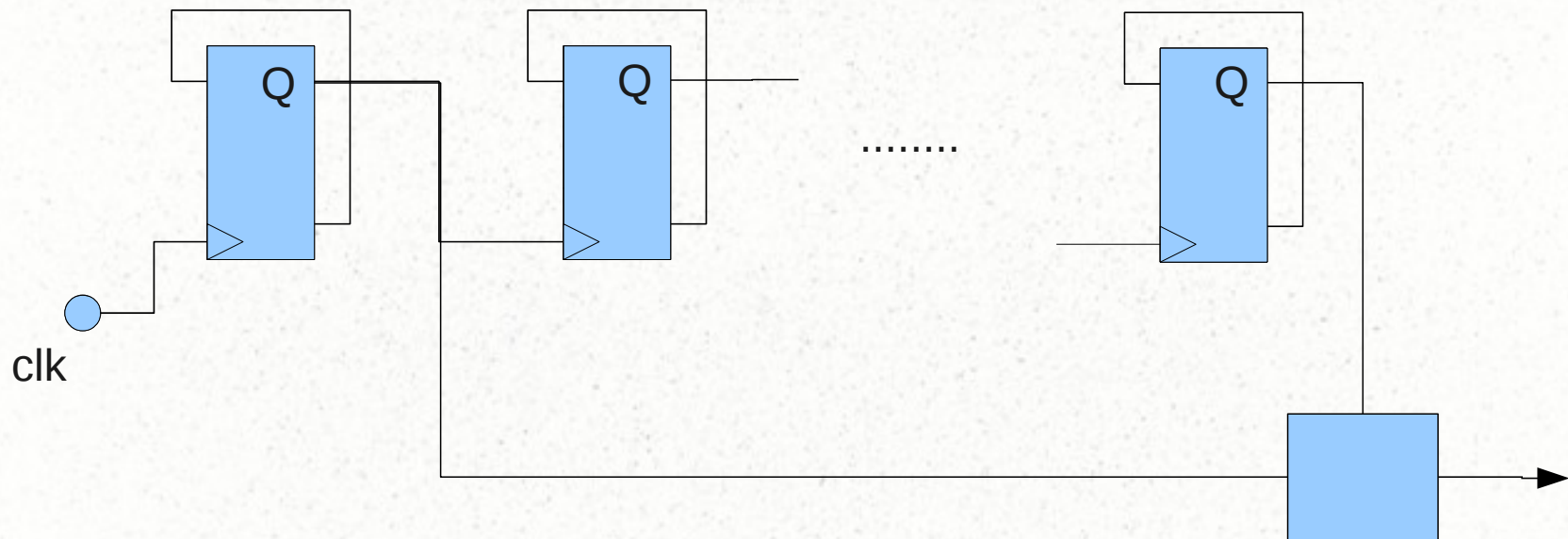
# *Benodigde software*

- Complete omgeving:
  - Vivado (Xilinx)
- Of: Aparte programma's
  - Tekst editor (plat ASCII)
  - Simulator: modelsim
  - Synthese: afhankelijk van FPGA:
    - Xilinx: ISE, Vivado
    - Altera/Intel: Quartus
  - Bit-programmer; upload-kabel



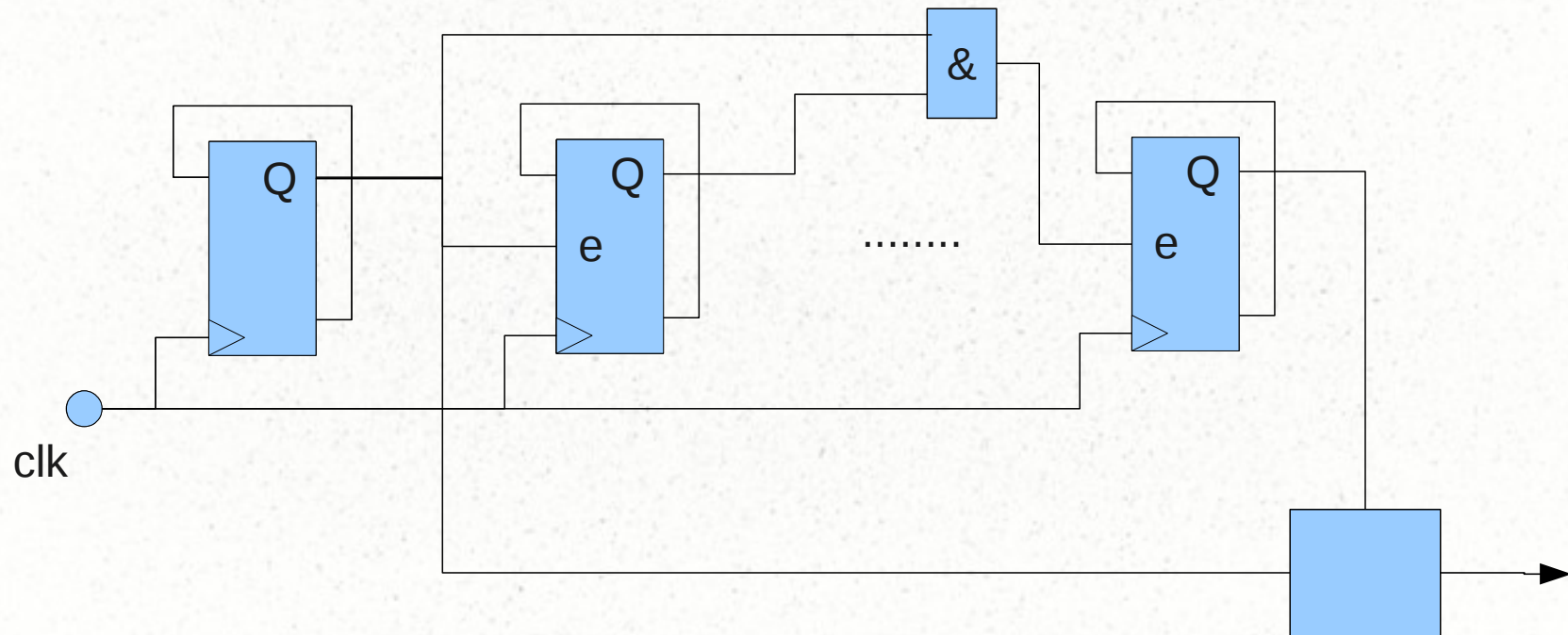
# Ontwerpen

- Asynchroon?



# Ontwerpen

- Synchronon!



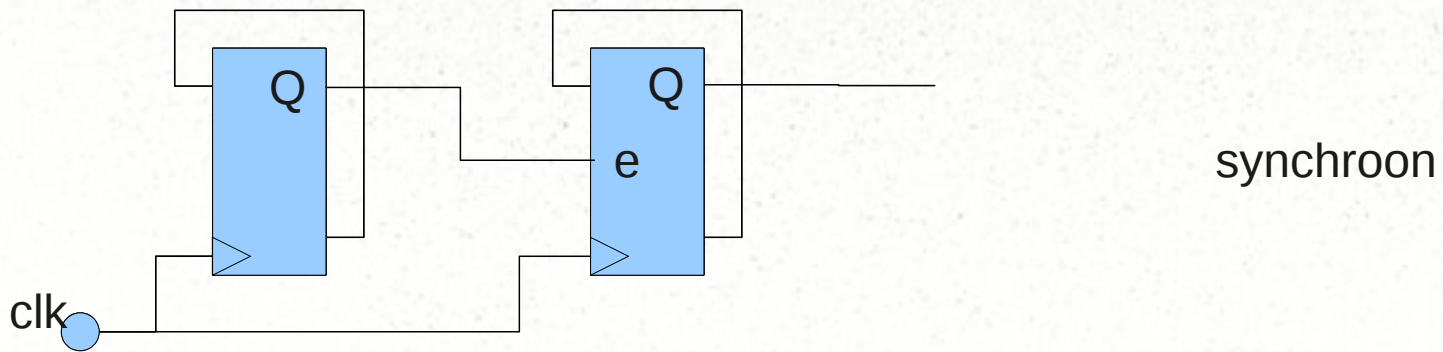
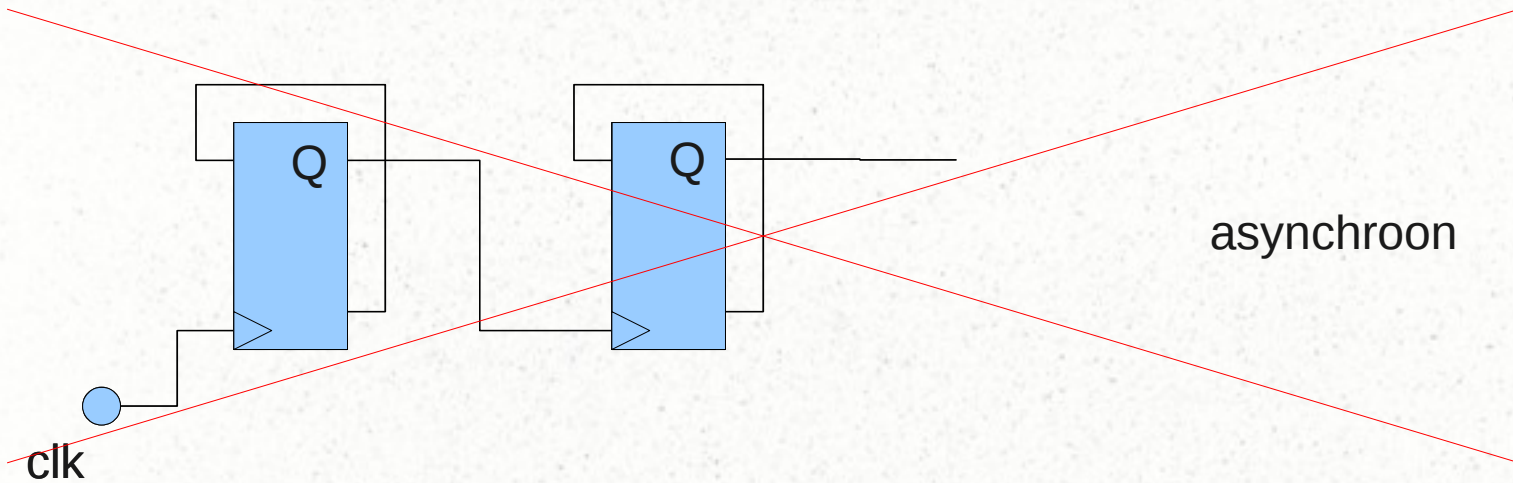
# *Een teller (2)*

```
PROCESS
BEGIN
    WAIT UNTIL clk='1';
    IF cnt < 15 THEN
        cnt <= cnt+1;
        puls <= '0';
    ELSE
        cnt <= 0;
        puls <= '1';
    END IF;
END PROCESS;
```



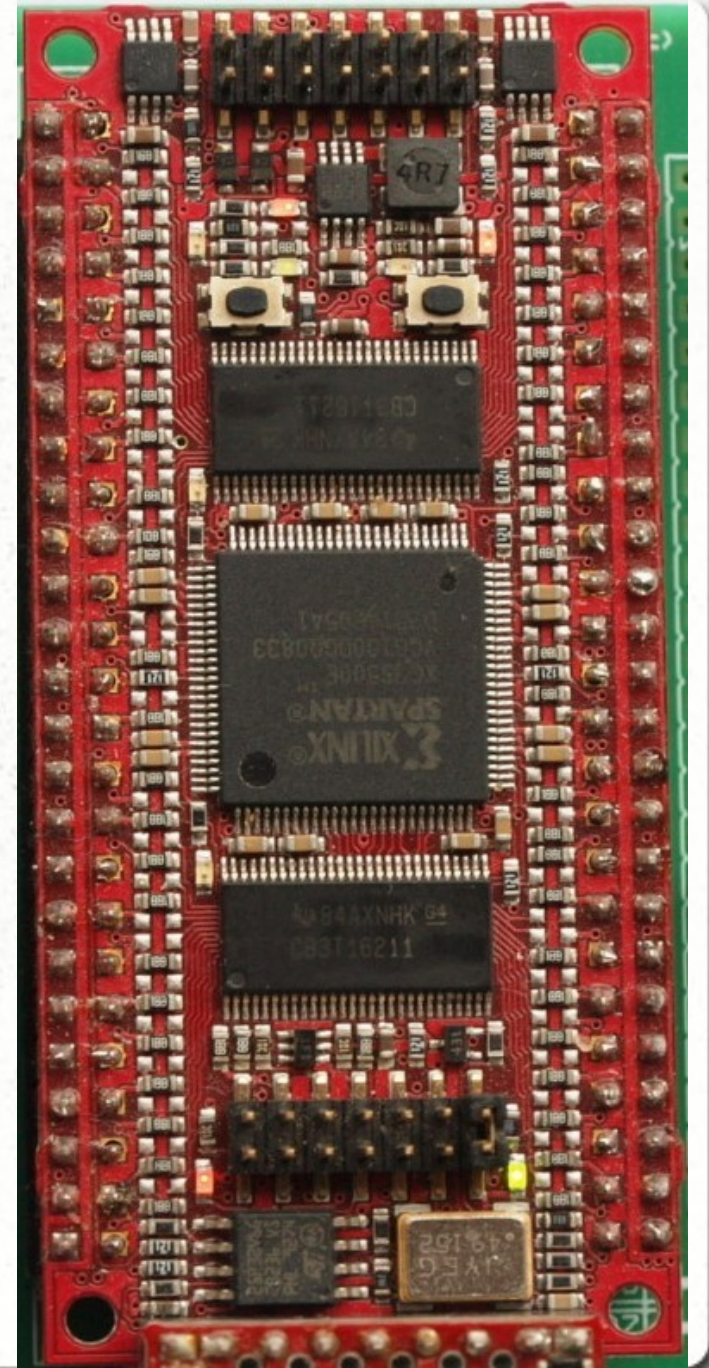
# Ontwerpen

- Synchron



## *(A) HRPT decoder*

- Flipflops: 2500
- LUT's: 8700
- IO: 50





# *Conclusie*

- Met een FPGA kan je relatief eenvoudig complexe digitale schakelingen ontwerpen, simuleren en realiseren.
- Kosten: bijna alleen de FPGA zelf (software gratis)
- Vhdl/verilog leren vereist wel aandacht



## *Links*

- Modelsim (simulieren):

[https://www.mentor.com/company/higher\\_ed/  
modelsim-student-edition](https://www.mentor.com/company/higher_ed/modelsim-student-edition)

- Xilinx, Vivado:

<https://www.xilinx.com/support.html>

- 8051 VHDL code:

<http://www.cs.ucr.edu/~dalton/i8051/i8051syn/>