

TSEA83 : Datorkonstruktion, 8hp

Fö1

Introduktion

Fö1 : Agenda

- Inledning
 - Mål, Resultat från tidigare år, Förändringar till årets kurs
- Formalia
 - Planering, Föreläsningar, Laborationer, Kurslitteratur
- VHDL
 - VHDL, Simulering, FPGA-kort
- Projekt
 - Exempel på tidigare projekt
- Distansarbete
 - Hur kan man arbeta hemifrån?

Inledning

Mål, Resultat från tidigare år, Förändringar till årets kurs

Mål

- Ni ska i grupper om 4 teknologer konstruera en inbyggd dator



Resultat från tidigare år

- 90 studenter gick kursen 2023
- 16 studenter svarade på utvärderingen i Evaluate
- Helhetsbetyget var 4,69

Förändringar till årets kurs

- Ny FPGA-hårdvara : Basys3 (tidigare Nexys3)
- Ny programvara : Vivado
- Nya datorer

Formalia

Planering, Föreläsningar, Laborationer, Kurslitteratur

Planering 2023, VT1 (2hp)

vecka	föreläsning	laboration	leverans
3	1. Intro 2. Mikroprog I 3. Mikroprog II		
4	4. Pipelining 5. Cache	1. Mikroprog	
5	6. Minnen +Bussar 7. Grafik +Proj	2. Pipelining	
6	8. VHDL I 9. VHDL II		Bilda 4-grupp klar ti kl 12
7	10. VHDL III	3. UART (VHDL)	Kravspec 0.1 inlämn. ti kl 12
8	11. ALU	4. VGA (VHDL)	Kravspec 1.0 klar ti kl 12
9			Designspec 0.1 inlämn. ti kl 12
10			Designspec 1.0 klar ti kl 12

Planering 2023, VT2 (6hp)

vecka	händelse	projekt	leverans
11	Omtenta-P ht2		
12	Tenta-P VT1		
13	(Påsk: to-fr)	Konstruktion	
14	(Påsk: må)	Konstruktion	Statusrapp. ti kl 12
15		Konstruktion	Statusrapp. ti kl 12
16		Konstruktion	Statusrapp. ti kl 12
17		Konstruktion	Statusrapp. ti kl 12 Milstolpe
18	(Valb.+1Maj: ti+on)	Konstruktion	Statusrapp. ti kl 12
19		Konstruktion	Statusrapp. ti kl 12
20		Konstruktion	Statusrapp. ti kl 12
21			Presentation ti + on, Tek.dok + återlämning fr kl 12

1. Intro
2. Mikroprogrammering 1
3. Mikroprogrammering 2
4. Pipelining
5. Cache
6. Minnen + Bussar
7. Grafik + Projekt
8. VHDL 1
9. VHDL 2
10. VHDL 3
11. ALU/Projektstart

} CPU

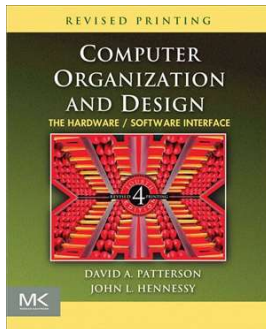
} VHDL är språket
i projektet

Alla PP-slides på hemsidan

- 1. Mikroprogrammering** (2 + 2 tim)
 - Kan göras på egen hand och endast redovisas på labben
- 2. Pipelining** (2 + 2 tim)
 - Kan göras på egen hand och endast redovisas på labben
- 3. UART** (seriell kommunikation) (2 + 2 tim)
 - Kodning och simulering kan göras på egen hand.
Syntetisering kräver FPGA-kort i labsal
- 4. VGA** (2 + 2 tim)
 - Kodning och simulering kan göras på egen hand.
Syntetisering kräver FPGA-kort i labsal

Alla lab-pM på hemsidan

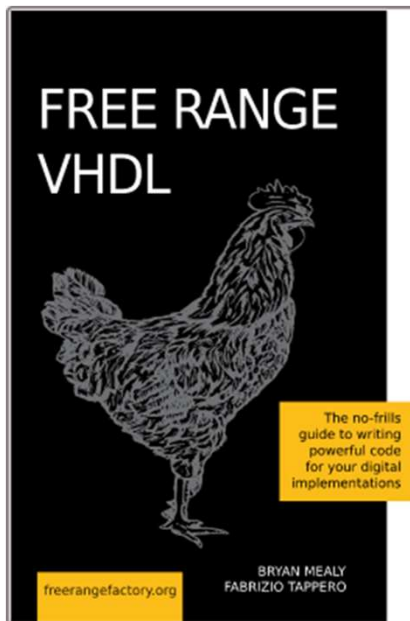
Kurslitteratur - Datorteknikdelen



Patterson, Hennessy: Computer Organization and design
→ pipelining, **e-bok**



Olle Roos: Grundläggande datorteknik
→ mikroprogrammering



Mealy, Tappero
Free Range VHDL
→ pdf

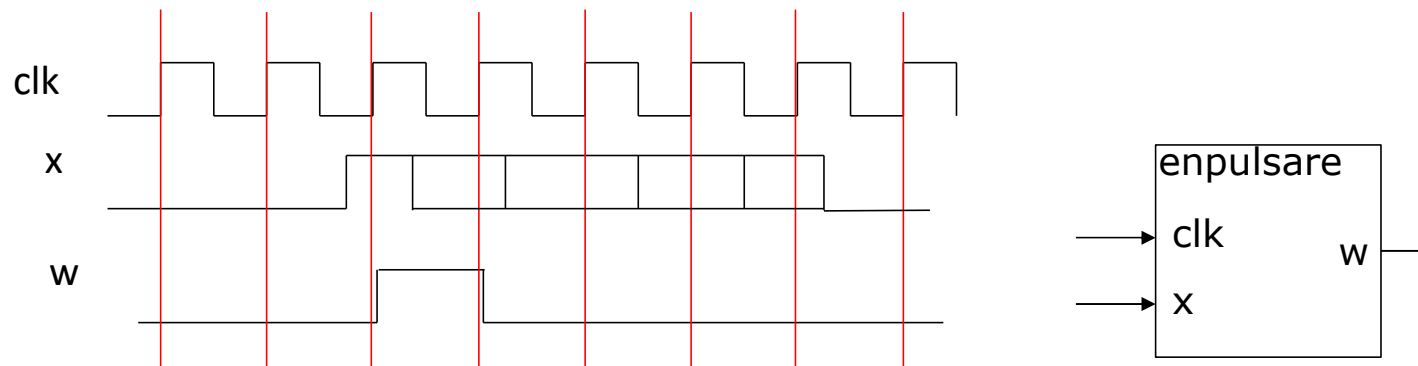
VHDL

VHDL, Simulering, FPGA-kort

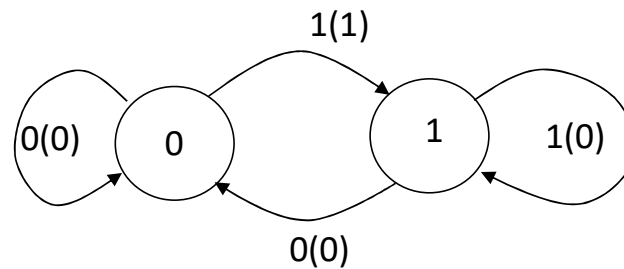
VHDL : 1.Problembeskrivning

Jag vill bygga en ”synkronisering och enpulsare”.
När asynkron insignal går hög (0->1) ska en synkroniserad
utpuls produceras. Pulsens längd ska vara 1 CP. Inpulsen måste
vara hög vid minst en klockflank.

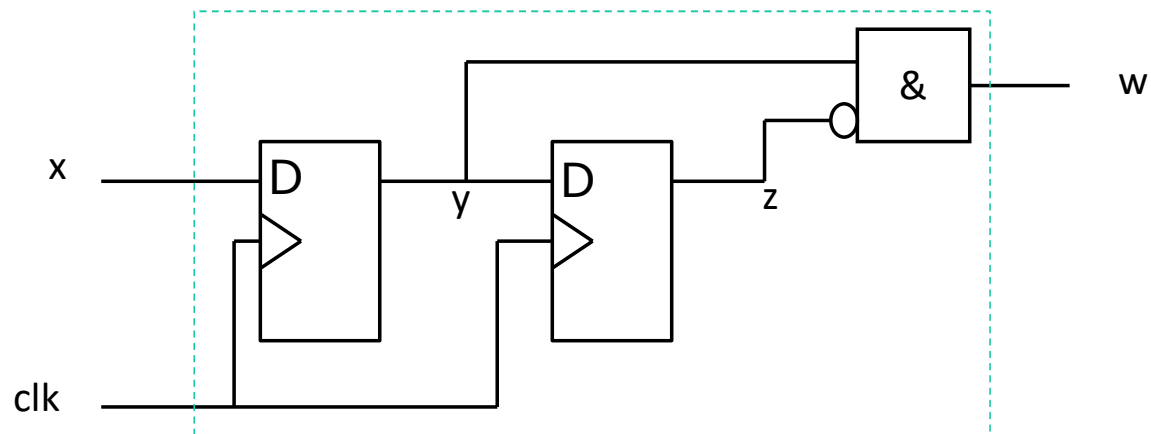
Tidsdiagram



a) Synkvippa + tillståndsgraf



b) Hårdvaruschema direkt



VHDL : 3.VHDL-kod för b)

```
entity enpulsare is
  port(x,clk : in std_logic;
        w : out std_logic);
end enpulsare;
```

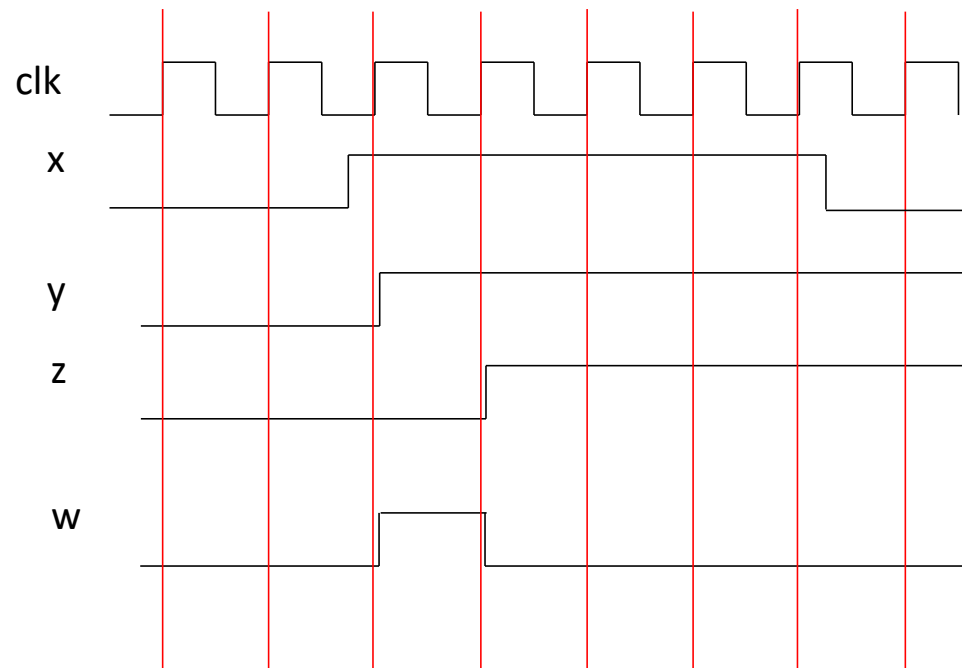
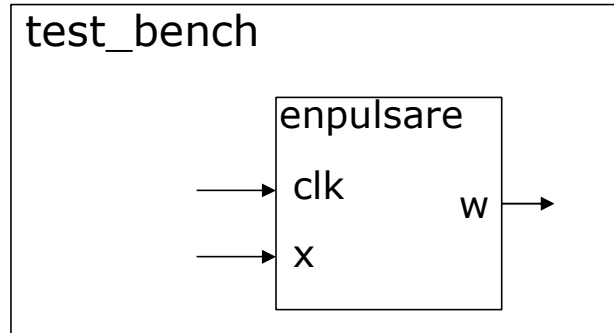
entity:
Gränssnitt mot
omvärlden

```
architecture func of enpulsare is
  signal y,z : std_logic;
begin
  w <= y and not z;          -- kombinatoriken

  process(clk)
  begin
    if rising_edge(clk) then
      y <= x;                -- synk-vippan
      z <= y;                -- tillståndsvippan
    end if;
  end process;
end architecture;
```

architecture:
Innehåll i kretsen

VHDL : 4.Simulering med ModelSim



Klocka

Insignal

Interna signaler

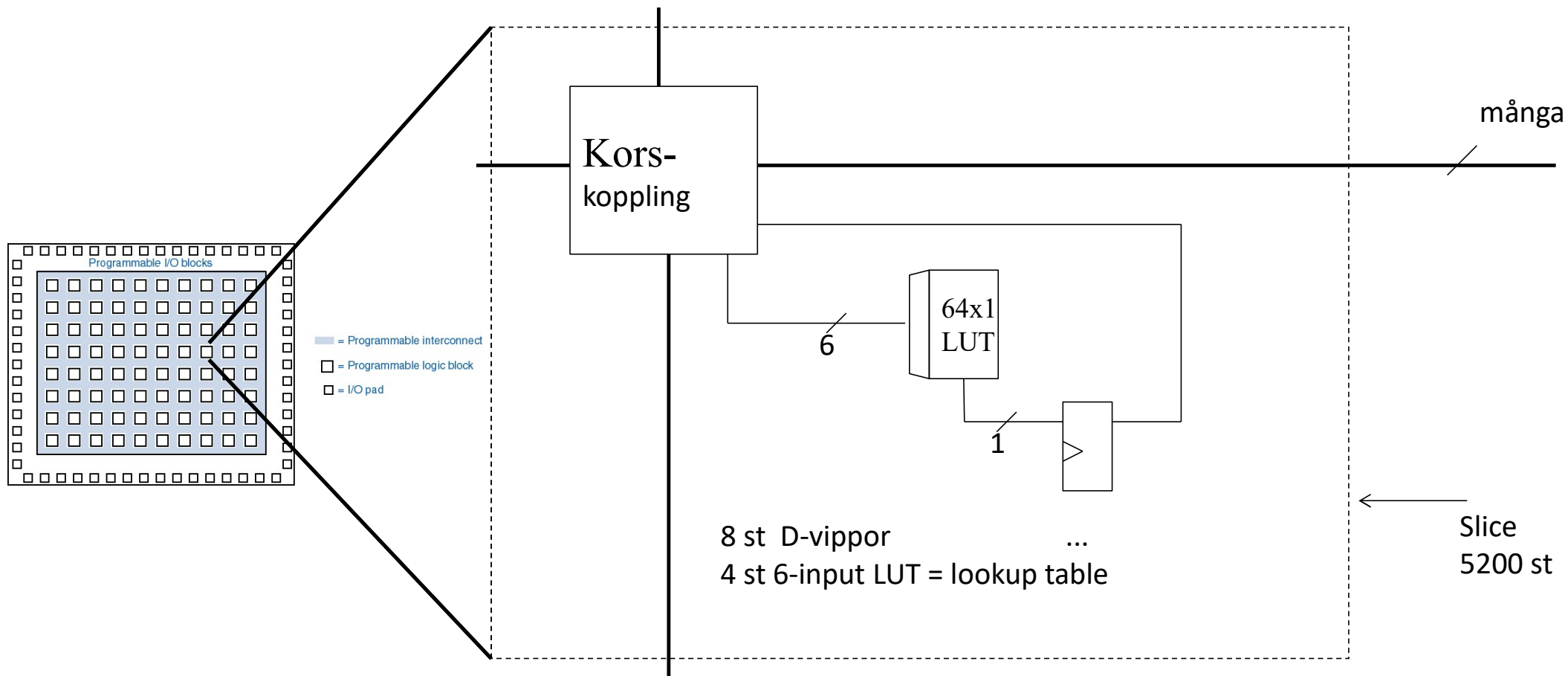
Utsignal

VHDL : 5.Syntetisering till Basys 3



FPGA (Field Programmable Gate Array)

- 41600 vippor
- 20800 LUTs (Look Up Table)(64x1 RAM)
- 100 st 2kB blockRAM (1800Kbit)
- 90 st 18x18 multiplikatorer + (MAC)

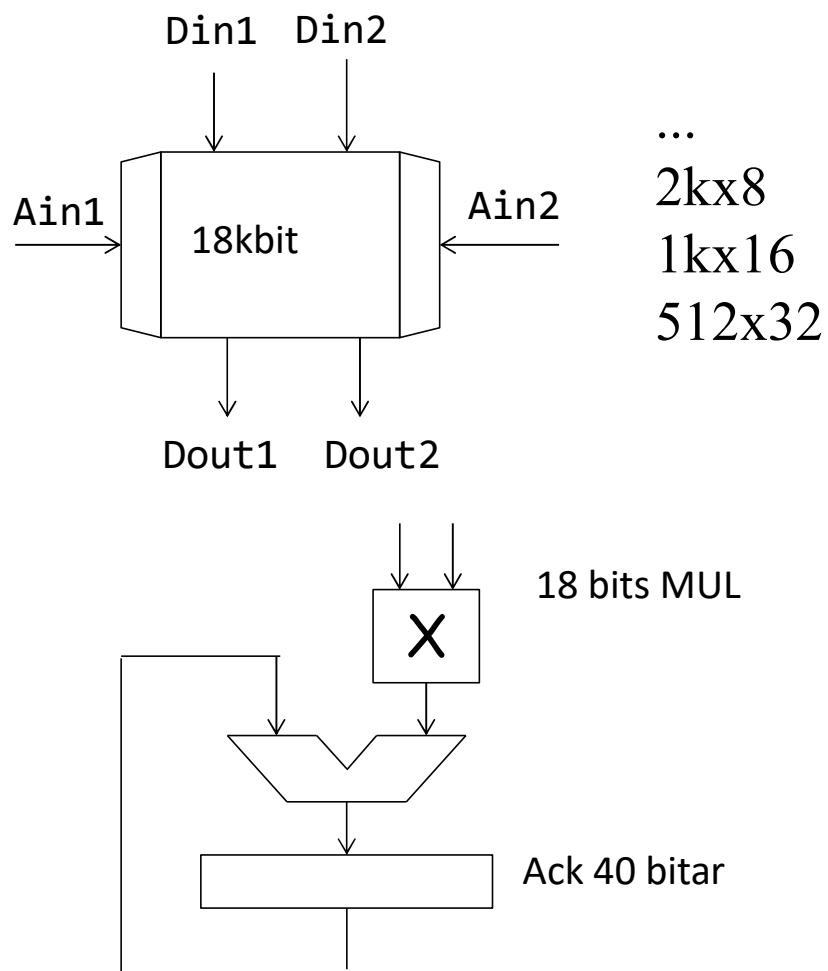


VHDL : FPGA Artix-7 35T

Innehåller också

100 st block RAM
a 2kB a 9 bit

90 st DSP slices



VHDL : Hur går det till?

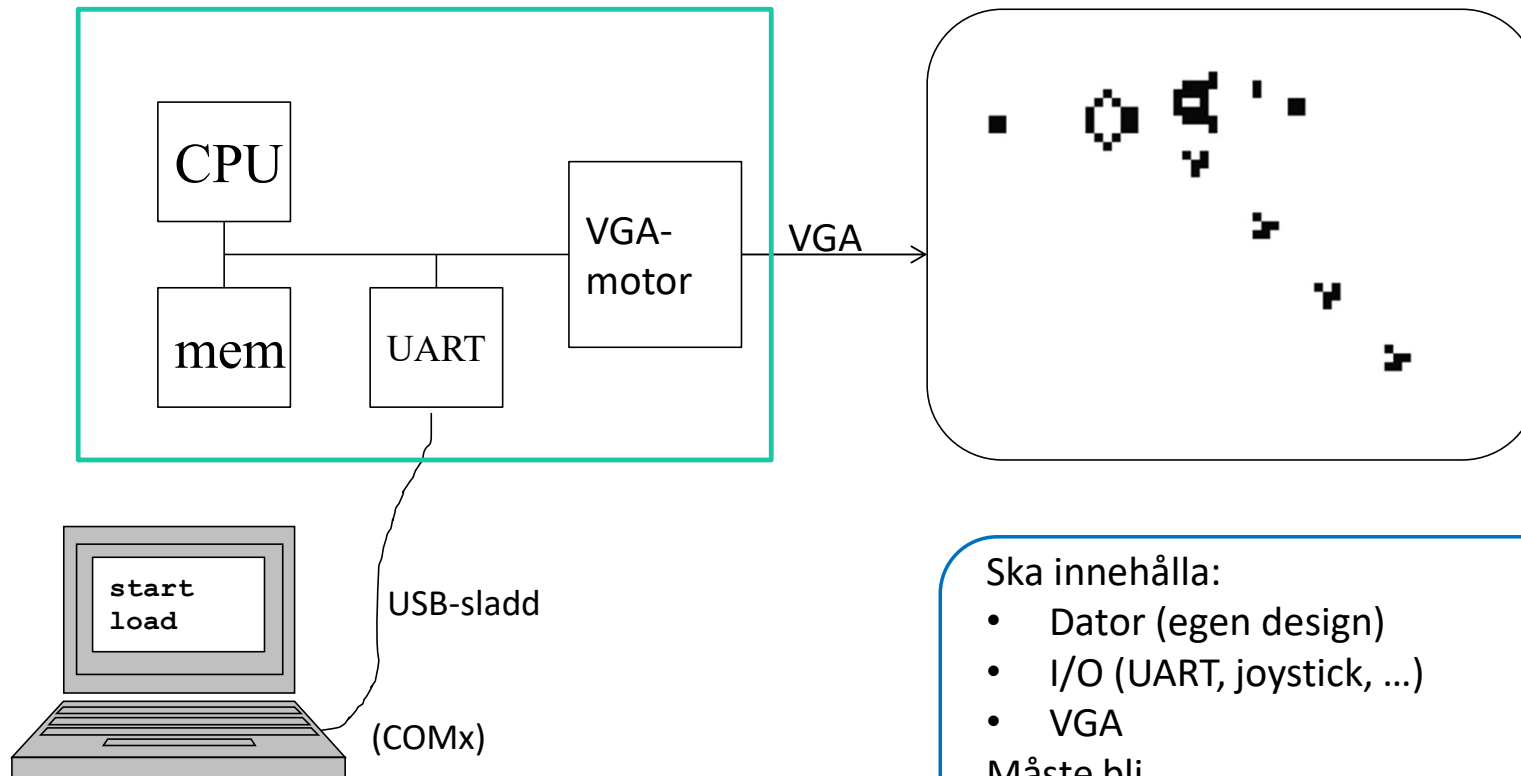
- Skriv VHDL med Emacs el dyl. Vi kör Linux.
- **make sim**
 - ModelSim hoppar igång
- **make bit**
 - synthesize = "booleska ekvationer"
 - translate = "LUTtar och vippor"
 - place = "placera allting i schema"
 - route = "koppla ihop allting"
- **make prog**
 - filen design.bit programmeras i FPGA'n

Projekt

Exempel på tidigare projekt

Vad kan man bygga?

"en grafikdator, som gör Game of Life"



Ska innehålla:

- Dator (egen design)
- I/O (UART, joystick, ...)
- VGA

Måste bli

- demonstrerbar

Vad har dom gjort förut?

Tetris
Drag race
Snake
2048
Music Mania
Mandelbrot
Ritprogram
Grafisk räknare
Pong
Pac-man
Minesweeper

Achtung die kurve!
Breakout
Frogger
Othello
Synth
Bomberman
Space invaders
Asteroids
Tron
[Counter Strike 2D](#)
[GameBoy](#)
[MUX-OS](#)

<https://www.youtube.com/watch?v=HThvHjRpshE>

<https://www.youtube.com/watch?v=EjhpCFiEj40>

<https://www.youtube.com/watch?v=9t4dSRWNqOM&feature=youtu.be>

Distansarbete

Hur kan man arbeta hemifrån?
Se menyvalet **Distansarbete** under Lisam

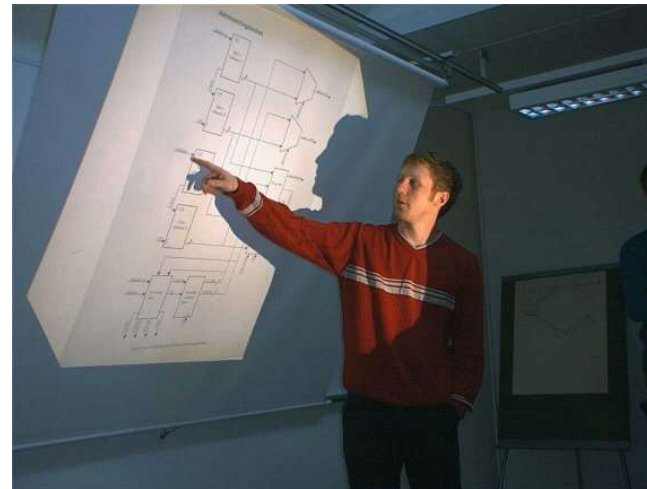
Sammanfattning

VT1

- Föreläsningar
- Laborationer
- Bilda 4-grupp (ti v.6)
- Skriv en kravspecifikation
 - **Vad** ska datorn göra
- Skriv en designspecifikation
 - **Hur** ska datorn konstrueras

VT2

- Konstruktion i 8 veckor
- Demo av fungerande dator (enl. kravspec)
- Föredrag v.21 (15 min)
- Skriv en Teknisk rapport



Vad behöver ni göra nu?

- Registrera er på kursen TSEA83
- Anmäl er till laborationerna, i Lisam
 - 2 tillfällen per lab, i grupper om 2 studenter

Anders Nilsson

www.liu.se