

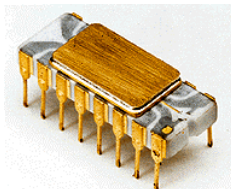
Introduktion till konstruktion av digitala elektroniksystem

Innehåll

- Historik
- Motivation – hårdvarubeskrivande språk
- Översikt av integrerade kretsar
- Konstruktionsflöde
- Konstruktionsverktyg

Historik –den integrerade kretsen

- ◆ 1958: Den första integrerade kretsen presenteras
 - Ej användbar teknik eftersom den var för dyr
- ◆ 60-talet:
 - En stor drivkraft till utvecklingen var NASAs rymdprogram
- ◆ 70-talet:
 - Kretsarna blev all mindre, snabbare och billigare
 - 1971 kom Intels första mikroprocessor – 4004

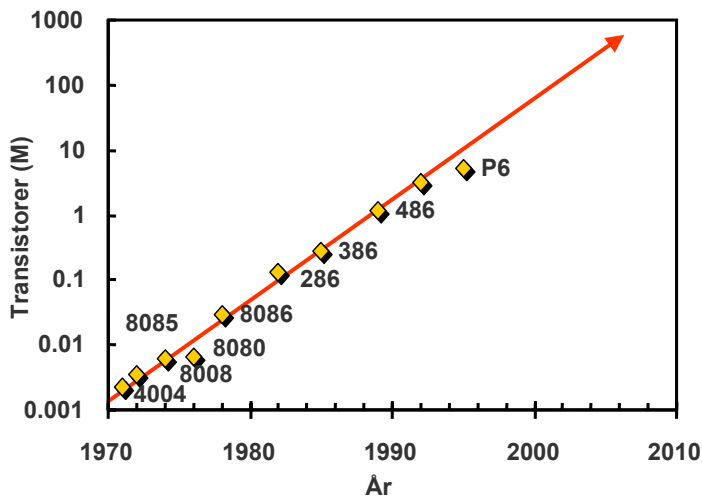


Historik – den integrerade kretsen

- ◆ 1958: Första analoga integrerade kretsen
 - Jack Kilby (Nobelpristagare 2000)
- ◆ 60-talet
 - Digital elektronik börjar komma i flera elektronikprodukter
- ◆ Fortsättningen
 - 70-talet: 100 tr./IC
 - 80-talet: 10 000 tr./IC
 - 90-talet: 100 000 tr./IC
 - Nu: 10-tals miljoner tr./IC

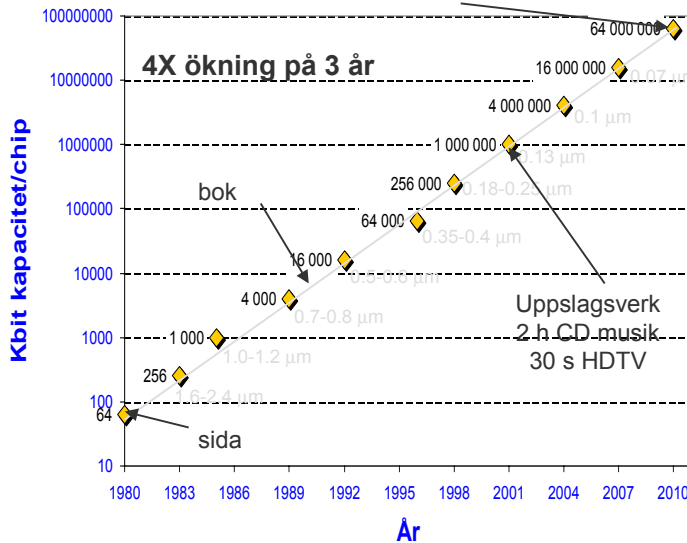


Utveckling – Mikroprocessor



Utveckling – Minnen (DRAM)

Mänskligt DNA



Bengt Oelmann -- copyright 2002

5

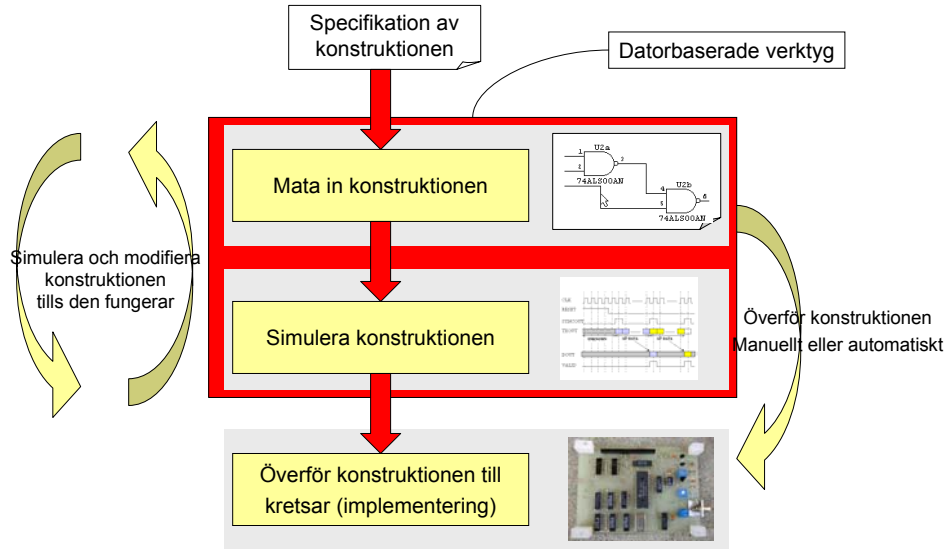
Vad används integrerade kretsar till ?

- ◆ Generella integrerade kretsar
 - Mikroprocessor
 - Minnen
 - AD-omvandlare
 - Speciella programmerbara kretsar
 - T.ex för digital signalbehandling (DSP)
- ◆ Speciella integrerade kretsar
 - Tillämpningsspecifik (t.ex krets för TV-mottagare)
 - Specifik för en viss produkt
 - Kontrollera speciella maskiner (hiss, traktor, diskmaskin)

Bengt Oelmann -- copyright 2002

6

Konstruktionsflöde för digitala system

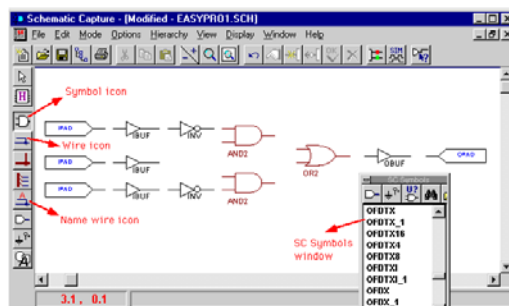


Bengt Oelmann -- copyright 2002

7

Schemainmatning

- ◆ Grafisk metod
 - Placera ut färdiga komponenter på en rityta
 - Koppla ihop in- och utgångar på dessa komponenter för att bygga en ny funktion



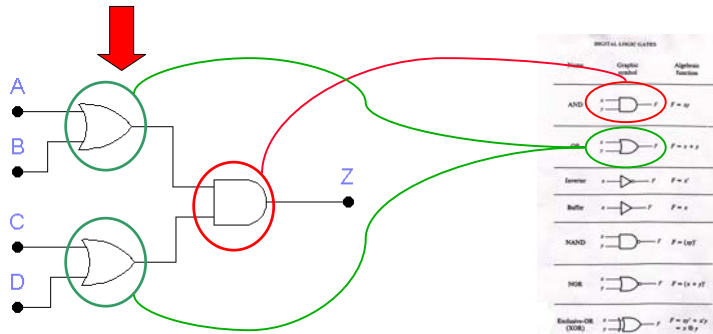
Bengt Oelmann -- copyright 2002

8

Exempel 1: Syntes av kombinatorisk logik

- ◆ Implementera funktionen m.h.a tillgängliga grindar

- $Z = (A+B) \cdot (C+D)$

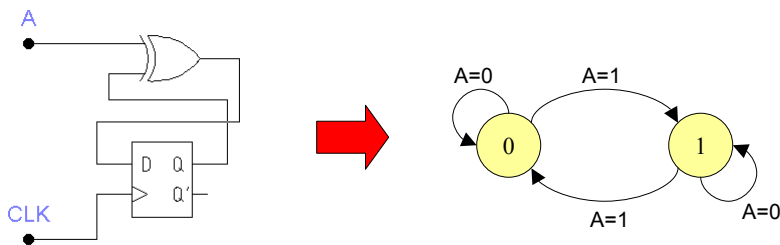


Bengt Oelmann -- copyright 2002

9

Exempel 2: Analys av tillståndsmaskin

Hur fungerar tillståndsmaskinen ?



Strukturell beskrivning
 -Sammankoppling av grundläggande komponenter
 -Nätlista (grind-nivå)
 -Eng. gate-level netlist

Beteendebeskrivning
 -Beskriver hur den fungerar
 -Säger inget om hur den är gjord

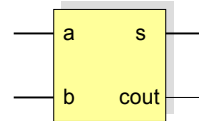
Bengt Oelmann -- copyright 2002

10

Exempel 3: Konstruktion med hårdvarubeskrivande språk

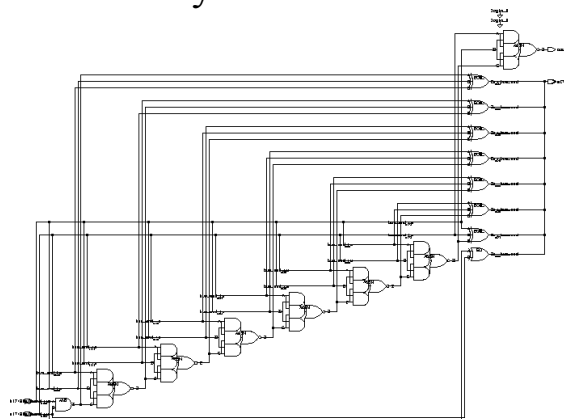
◆ 8-bitars adderare

```
architecture rtl of add8 is
begin
  process (a,b)
    variable s_var : std_logic_vector(8 downto 0);
  begin
    s_var := a + ('0' & b);
    s <= s_var(7 downto 0);
    cout <= s_var(8);
  end process add;
end rtl;
```



Forts. Exempel 3

◆ Automatisk syntes av koden för adderaren

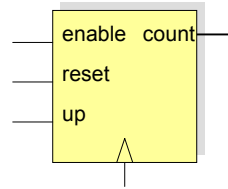


Design: add8	Design: Bengt Oelmann	Date: 01/2002
Technology: VHDL	Company: Högskolan i Gäddede	Sheet: 1 of 1

Exempel 4: Tillståndsmaskin

◆ Modulo-8 upp/ner-räknare med 'enable'

```
architecture rtl of fsm is
    signal count : std_logic_vector(2 downto 0);
begin -- rtl
    process (clk, reset)
    begin -- process
        if reset = '0' then
            count <= (others=>'0');
        elsif clk'event and clk = '1' then
            if enable='1' then
                case up is
                    when '1' => count <= count + 1;
                    when others => count <= count -1;
                end case;
            end if;
        end process;
        q <= count;
    end rtl;
```

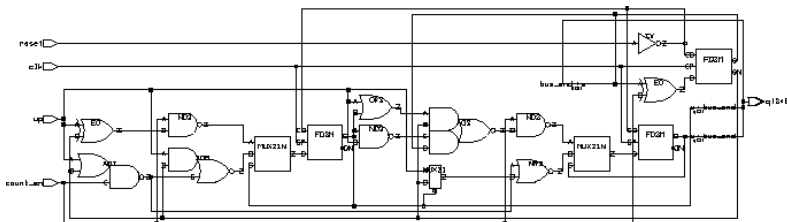


Bengt Oelmann -- copyright 2002

13

Forts. Exempel 4

◆ Automatisk syntes av koden för räknaren



design: up-down	designer: Bengt Oelmann	date: 3/8/2000
technology: HYCMOS00	company: MTHH gøteborg AB	sheet: 1 of 1

Bengt Oelmann -- copyright 2002

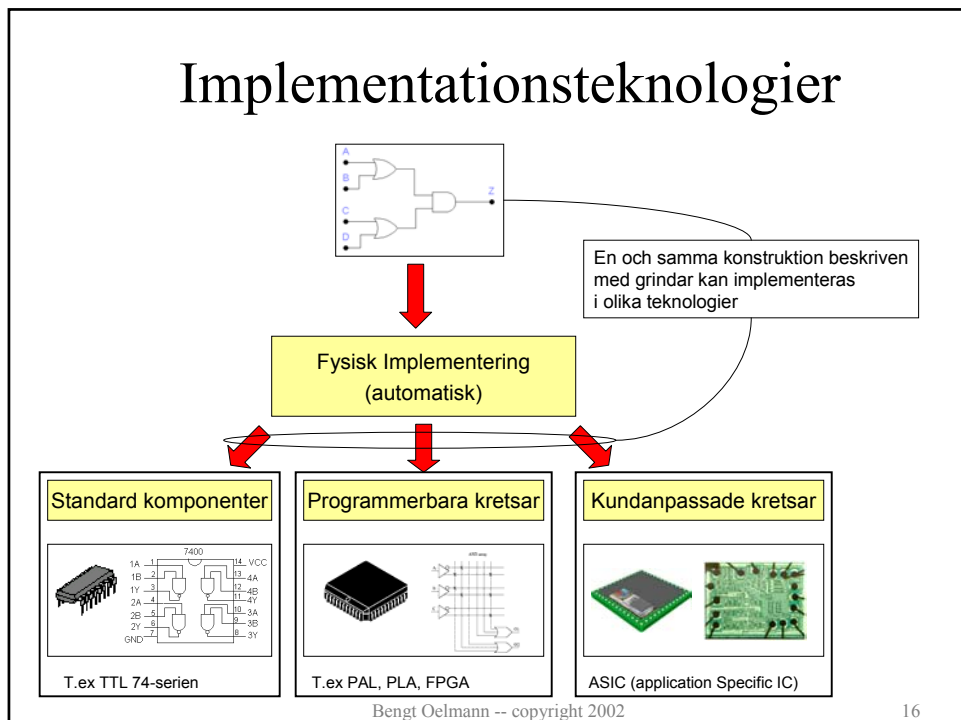
14

Hårdvarubeskrivande språk

◆ Fördelar

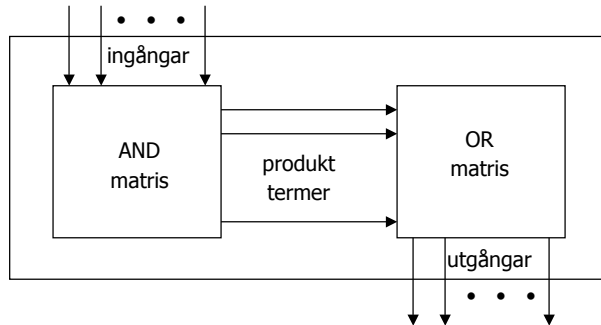
- Jämfört med schemabaserad konstruktion så blir det lättare att beskriva och läsa konstruktionernas funktion
- Koden är oberoende av implementationsteknologi
 - Den läser inte konstruktionen till en enda teknologi
- Koden kan återanvändas
 - Parameteriserbar kod

Implementationsteknologier



Programmerbara logiska matriser – PAL

- ◆ Pre-fabricerade byggblock med många AND/OR grindar
 - Består egentligen av NOR eller NAND grindar
 - Kretsen kan konfigureras (programmeras) genom att bryta kopplingar mellan grindar
 - Blockdiagram för programmerbar krets som skapar uttryck på summa-av-produktform

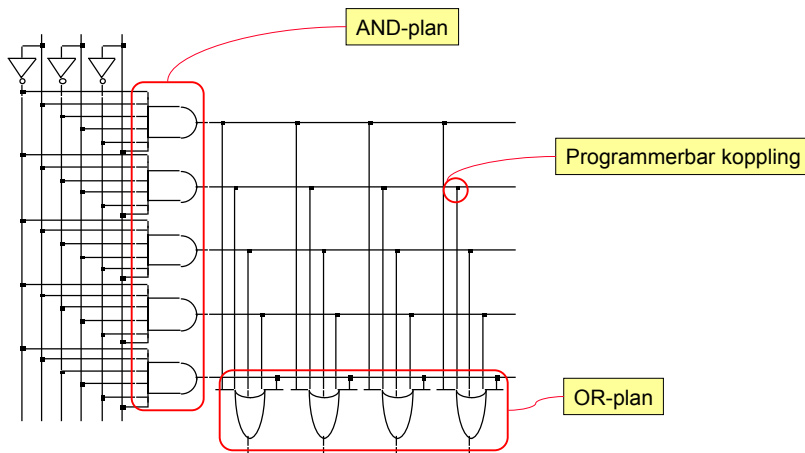


Bengt Oelmann -- copyright 2002

17

Före programmering

- ◆ Alla möjliga kopplingar finns tillgängliga före programmering

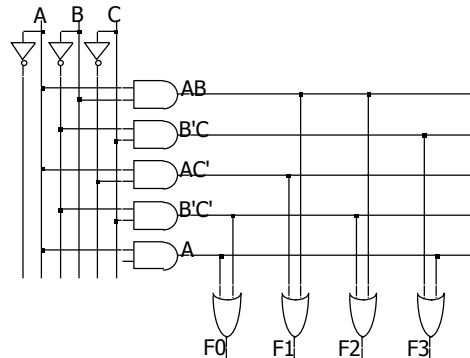


Bengt Oelmann -- copyright 2002

18

Efter programmering

- ◆ Ej önskade kopplingar tas bort
 - 'Fuse' normalt kopplade, bryt oönskade
 - 'Anti-fuse' (normalt okopplade, koppla de önskade)

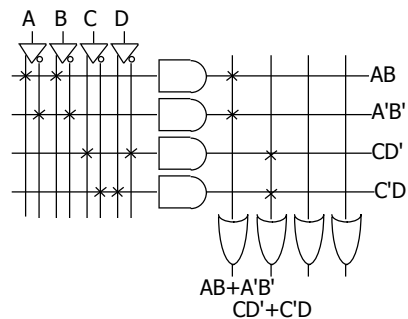
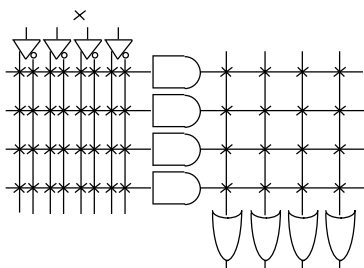


Bengt Oelmann -- copyright 2002

19

Exempel

- ◆ Förenklad notation – alla ledningar ritas ej
 - Visar att en koppling finns och att signalen är en ingång till grinden
 - Implementera $F0=AB + A'B'$ samt $F1=CD' + C'D$

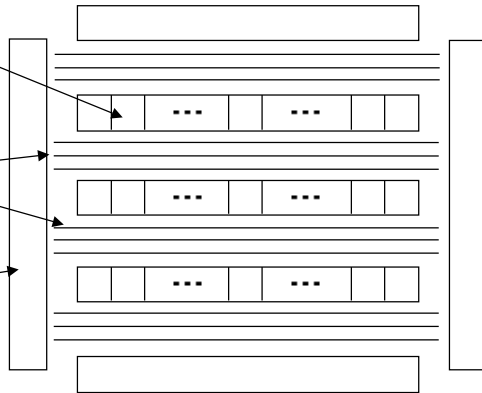


Bengt Oelmann -- copyright 2002

20

Grindmatriser (Gate-arrays)

- ◆ Enkla logiska grindar
 - Transistorer implementerar kombinatorisk och sekventiell logik
- ◆ Sammankopplingar
 - Ledningar för att koppla samman in- och utgångar till logiska block
- ◆ I/O block
 - Speciella block för att koppla signalerna externt chippet
- ◆ Lägg till ledningar för sammankoppling
 - Metalledningar läggs till efter chippet är tillverkat
 - “mask-programmable”



Bengt Oelmann -- copyright 2002

21

Att göra stora programmerbara kretsar

- ◆ Alternativ 1: ”CPLD”
 - Lägg massor av PAL-kretsar på samma chip
 - Lägg till ledningar mellan dem och vars kopplingar kan programmeras
- ◆ Alternativ 2: ”FPGA”
 - Härma Gate-array teknologin
 - Kallas för *Field Programmable Gate Array* (FPGA)
 - För det krävs
 - Ett sätt att implementera logiska grindar som kan konfigureras
 - Ett sätt att koppla samman dem

Bengt Oelmann -- copyright 2002

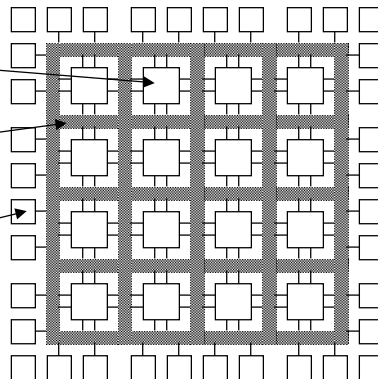
22

Field-Programmable Gate Arrays

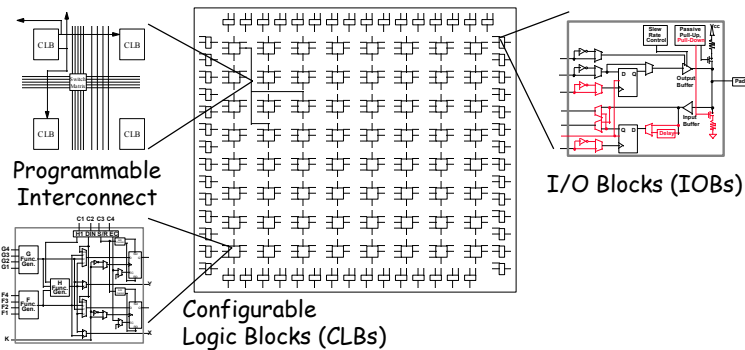
- ◆ PAL-kretsar
 - 10 – 100 grindar per chip
- ◆ Field Programmable Gate Arrays
 - Altera MAX Family
 - Actel Programmable Gate Array
 - Xilinx Logical Cell Array
 - 10000 – 4 miljoner grindar per chip

FPGA

- ◆ Logiska block
 - Implementerar kombinatorisk och sekventiell logik
- ◆ Sammankopplingar
 - Ledningar kopplar in- och utgångar till logiska block
- ◆ I/O block
 - Speciella block för att koppla signalerna externt chippet



FPGA från Xilinx



Bengt Oelmann -- copyright 2002

25

Datorstödd konstruktion

- ◆ Det är inte möjligt att konstruera FPGA för hand
 - Alldeles för mycket logik som ska hanteras
- ◆ Hårdvarubeskrivande språk (HDL)
 - Specificera logikens funktion på en hög nivå
- ◆ Validering: högnivå simulering
 - För att tidigt finna fel i konstruktionen
- ◆ Logiksyntes
 - Kompilera HDL program till logiska grindar
- ◆ Överföring till implementationsteknologi (FPGA)
 - Logiken ska föras över till de element som finns tillgängliga i den implementationsteknologi som finns tillgänglig

Bengt Oelmann -- copyright 2002

26

VHDL – ett hårdvarubeskrivande språk

- ◆ Det finns två vanliga språk för att beskriva hårdvara
 - VHDL
 - Verilog HDL
- ◆ VHDL
 - Very High Speed Integrated Circuit Hardware Description Language
- ◆ Historik
 - Skapat av amerikanska försvarsdepartementet (DoD) för att dokumentera militära konstruktioner
 - Standardiserat av IEEE (1076 VHDL) 1993

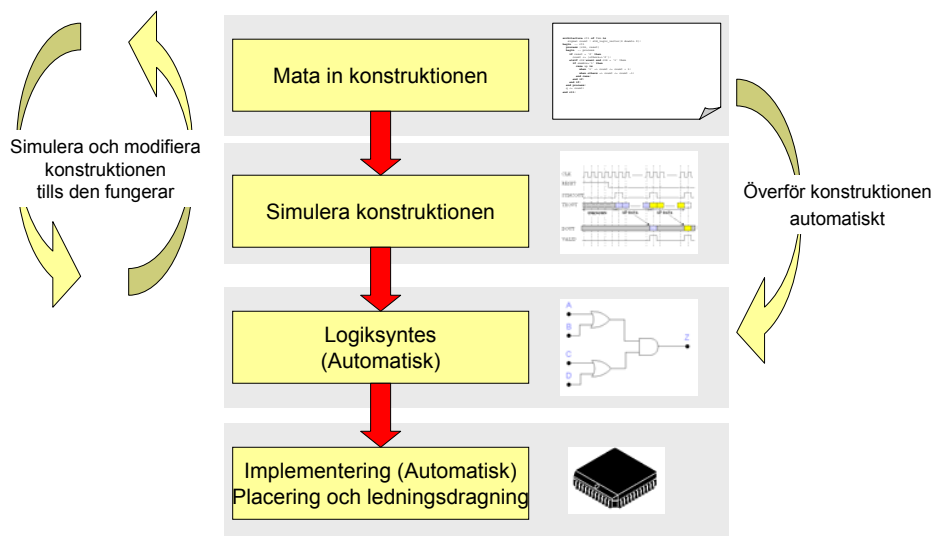
Bakgrund till VHDL

- ◆ Problem
 - Ökande behov för att kunna konstruera, implementera, test och dokumentera allt komplexare system på allt kortare tid
- ◆ Lösning (?)
 - Ett högnivåspråk för att beskriva hårdvara (HDL)
 - HDL tillsammans med datorstödd konstruktion (CAD) för automatisk syntes och simulering
 - Programmerbar logik för snabb implementering

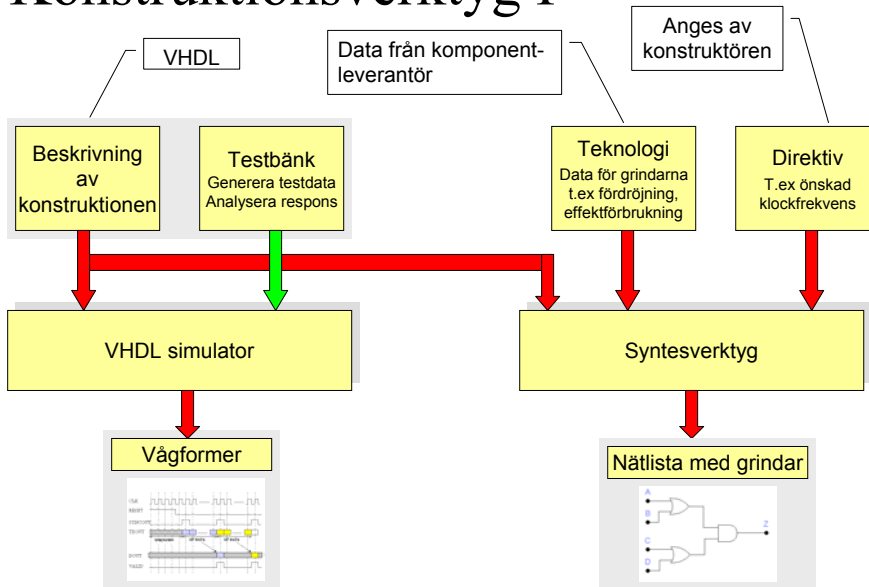
Bakgrund till VHDL

- ◆ Anledningar till att använda HDL
 - Högnivåspråk gör det kraftfullt och flexibelt
 - Konstruktionen görs oberoende av den komponent eller teknologi man ska använda
 - Standardiserat språk gör det möjligt att använda verktyg från olika leverantörer (simulatorer etc.)
 - Gör det möjligt att gå från idé till produkt snabbt med programmerbara kretsar

Konstruktionsflöde för FPGA



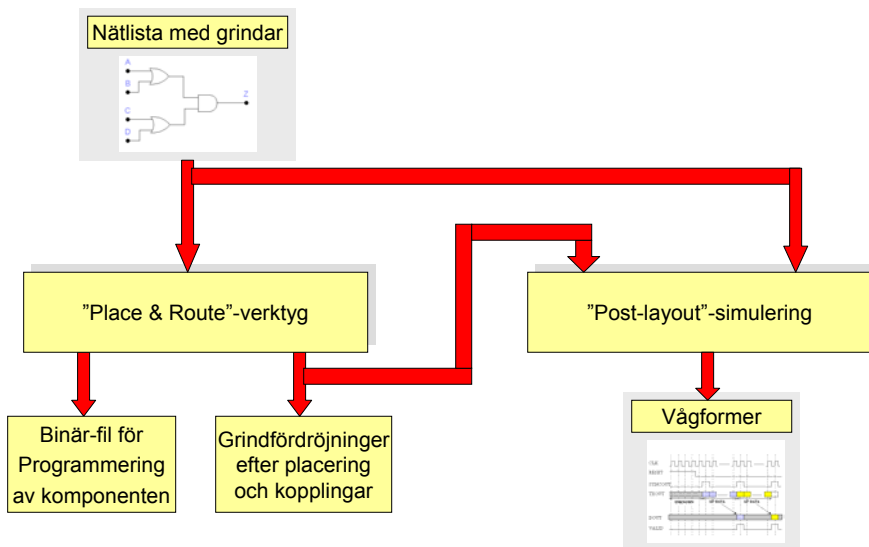
Konstruktionsverktyg I



Bengt Oelmann -- copyright 2002

31

Konstruktionsverktyg II



Bengt Oelmann -- copyright 2002

32

SLUT på Föreläsning 1

◆ Innehåll

- Historik
- Motivation – hårdvarubeskrivande språk
- Översikt av integrerade kretsar
- Konstruktionsflöde
- Konstruktionsverktyg