

Electronic Design Automation (EDA)

Entwurfseingabe

Entwurfseingabe

Systemebene

...SystemC

...SystemVerilog

RT-Ebene

...Verilog

...VHDL

Gatterebene

...Gatternetzliste

...Schematic Entry

Elektrische Ebene/ Strukturelle Sicht

...Bauelementenetzliste

...Schematic Entry

Elektrische Ebene/ Physikalische Sicht

...CIF

...GDSII

...Layout Entry

Entwurfseingabe: Entwurfseingabe

Entwurfseingabe ...

- ... findet auf verschiedenen Ebenen statt (siehe Y-Diagramm).
- ... kann funktionell erfolgen: $C = A + B$
- ... kann strukturell erfolgen: $\text{ADD}(A, B, C)$
- ... wird durch graphische Benutzeroberflächen erleichtert.
- ... häufig werden auch Sprachen verwendet

Datenaustauschsprachen ...

- ... dienen dem Datenaustausch zwischen Programmen auf verschiedenen Ebenen.
- ... sind für maschinelle / automatisierte Verarbeitung konzipiert.

Beschreibungssprachen ...

- ... beschreiben Daten/Information auf höherer Ebenen.
- ... kommen menschlicher Denkweise entgegen

Der Entwickler kann den Entwurf auf verschiedenen Entwurfsebenen und in verschiedenen Sichten eingeben. Viele Entwurfsschritte können durch ein EDA-Tool vorgenommen werden, ein vollständig automatisierter Ablauf (Silicon Compiler) ist nicht realisierbar, deshalb muss der Entwickler in verschiedenen Stadien des Entwurfs Eingriffe von Hand vornehmen. Der Entwurf kann strukturell ($\text{ADD}(A, B, C)$) oder funktionell ($C=A+B$) eingegeben werden. Die Entwurfseingabe basiert immer auf Verwendung einer Eingabesprache. Häufig wird dem Entwickler die Eingabe durch eine graphische Benutzeroberfläche vereinfacht. Man unterscheidet bei den Eingabesprachen:

1) Datenaustauschsprachen

Diese Sprachen sind für den Austausch von Daten gedacht. Dabei handelt es sich hauptsächlich um einen maschinellen Austausch.

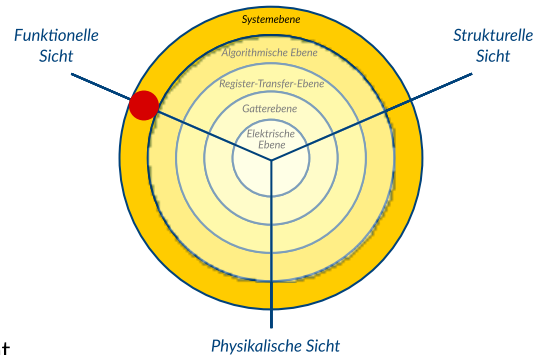
2) Beschreibungssprachen

Diese Sprachen dienen zur Beschreibung von Daten/Informationen auf einer Ebene mit einem höheren Abstraktionsgrad. Sie sind für die Benutzung durch den Menschen ausgelegt.

Entwurfseingabe: Systemebene

Systemebene / Funktionelle Sicht

- Oberste Eingabeebene
- Spezifikation als Eingabe
- Hardware- / Software-Funktionalität
- Eingabesprachen: SystemC, SystemVerilog, UML, SysML, Marte
- "Alternativen": C, C++, Mathematiksysteme wie Maple, Mathematica, MATLAB (inkl. SimuLink) u.a.



Die Systemebene ist die oberste Ebene für die Entwurfseingabe. Eine ideale Eingabe wäre die Systemspezifikation (z.B. 8 bit ALU bei 800 MHz lauffähig und mit folgendem Befehlssatz sowie weiteren spezifizierten Randbedingungen). Dies ist aber noch eine Vision. Wichtig ist, dass Sprachen auf dieser Ebene sowohl die Funktionalität von Hardware als auch von Software beschreiben können müssen, da Hardware/Software Co-Design immer wichtiger wird. Derzeitig unterstützte Eingabesprachen sind SystemC und System-Verilog für die Hardware. Alternative Eingaben auf Systemebene sind möglich mit C, C++, VML. Häufig wird für die Systembeschreibung auch mathematische Software wie Matlab, MAPLE oder Mathematica genutzt.

Eingabesprache SystemC

- C++ Klassen-Bibliothek
- Kompatibel und kompilierbar mit Standard-C-Compilern
- HW/SW-Co-Simulation / Co-Entwicklung möglich

```
#include "systemc.h"  
SC_MODULE(counter)  
{
```

```
// counter.cc  
#include "counter.h"  
void counter::onetwothree()  
{  
    if (clear) {  
        countval = 0 ;  
    } else if (load) {  
        countval = din.read() ;  
    } else {  
        countval ++ ;  
    }  
    dot = countval ;  
}
```

SystemC ist eine C++ Klassen-Bibliothek. SystemC ist voll kompatibel und kompilierbar mit Standard-C++-Compilern und bietet eine einfache HW/SW-Co-Entwicklungsumgebung.

Eingabesprache SystemVerilog

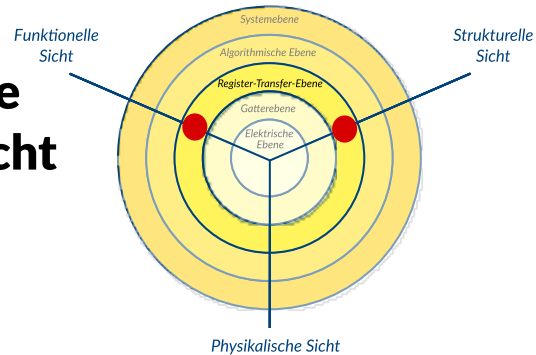
- Erweiterung von Verilog:
 - Objektorientiert (Klassen)
 - Dynamische Fehler
 - Assoziative Fehler
 - String-Operationen
 - Interprozesssynchronisierung
 - Direct Programming Interface (DPI zum Aufruf von C-, C++-Funktionen)
- Kompatibel mit Verilog-Standard
- Annäherung an C++

SystemVerilog ist eine Erweiterung von Verilog-2001 und kompatibel zum Verilog-Standard.
SystemVerilog ist stark angelehnt an C++.

Entwurfseingabe: RT-Ebene

Register-Transfer- Ebene / Funktionelle bzw. Strukturelle Sicht

- Beschreibung durch Register und Operationen
- Eingabe durch Hardwarebeschreibungssprache (VHDL, Verilog)
- Spracherweiterung für analoge Blöcke



Auf der Register-Transfer-Ebene existieren Register und Operationen. Die Eingabe auf dieser Ebene erfolgt mit Hilfe einer Hardwarebeschreibungssprache (Hardware Description Language, HDL). Die am weitesten verbreiteten Sprachen sind Verilog und VHDL. Für die eigentlich digitalen HDLs existieren Erweiterungen, um analoge Blöcke zu beschreiben. HDLs sind in der Lage, sowohl Funktion als auch Struktur zu beschreiben.

Eingabesprache Verilog

- 1984 / 1985 entwickelt von Moorby
- 1991 von Cadence übernommen
- 1995 IEEE Standard 1364 (Update 2001)
- Verilog-A zur Beschreibung analoger Schaltungen
- Code-Beispiel: Ampelsteuerung

```
...
always @(LA or FG or TimeOut
        or current_state)

begin
  case (current_state)
    HS_gruen :
      begin
        hsa = gruen;
        lsa = rot;
        nsa = rot;
        fga_as = rt_fg
        if (TimeOut==1)
          next_state = HS_gelb;
        else
          next_state = HS_gruen;
      end
  end
...

```

Verilog wurde 1984/1985 von Moorby und anderen entwickelt und 1991 von der EDA-Firma Cadence übernommen. Seit 1995 existiert der IEEE Standard 1364, der im Jahr 2001 aktualisiert wurde. Es existiert eine Spracherweiterung zu Verilog (Verilog-A) zur Beschreibung analoger Schaltungen.

Eingabesprache VHDL

- VHDL = VHSIC-HDL =
" Very High Speed Integrated Circuit -
Hardware Descripton Language "
- Entwicklung initiiert 1980 durch
US Department of Defense
- Design-unabhängig und
Simulator-unabhängig
- 1987 IEEE-Standart 1076
(Update 1993)
- VHDL-AMS-Erweiterung für
Mixed-Signal-Schaltungen

```
entity NANDXOR is
  port (
    A, B : in  bit;
    C    : in  bit;
    D    : out bit;
  end NANDXOR;

  architecture RTL1 of NANDXOR is
  begin
    D <= (A nand B) xor C;
  end RTL1;

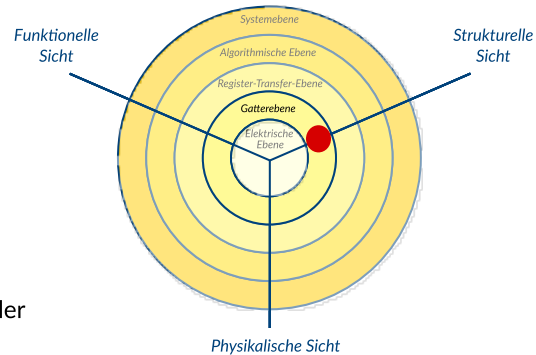
  architecture RTL2 of NANDXOR is
  begin
    process (A, B, C)
    begin
      if (C = '0') then
        D <= A nand B;
      else
        D <= A and B;
      end if;
    end process;
  end RTL2;

  achitecture RTL3 of NANDXOR is
  signal T : bit;
  begin
    T <= A nand B;
    p1 : process(T, C)
    begin
      D <= T xor C;
    end process p1;
  end RTL3
```

Entwurfseingabe: Gatterebene

Gatterebene / Strukturelle Sicht

- Prozesstechnologie kann berücksichtigt werden
- Gatterbibliothek der Chip-Hersteller
- Eingangskapazität und Treiberstärke können beachtet werden
- Gatterverzögerungszeiten können als Schätzung zur Verfügung stehen



Die Eingabe auf Gatterebene muss die zu verwendende Prozesstechnologie berücksichtigen. Gatter sind in einer vom Chip-Hersteller zur Verfügung gestellten Bibliothek beschrieben. Nur diese Grundgatter können verwendet werden. Eingangskapazitäten und Treiberstärke müssen beachtet werden. Gatterverzögerungszeiten stehen als Schätzung zur Verfügung.

Entwurfseingabe: ...Gatternetzliste

Gatternetzliste

- Beispiel Verilog-Netzliste:
- Signalübergangszeiten können mit $\#(T_{\text{fall}}, T_{\text{rise}})$ definiert werden.

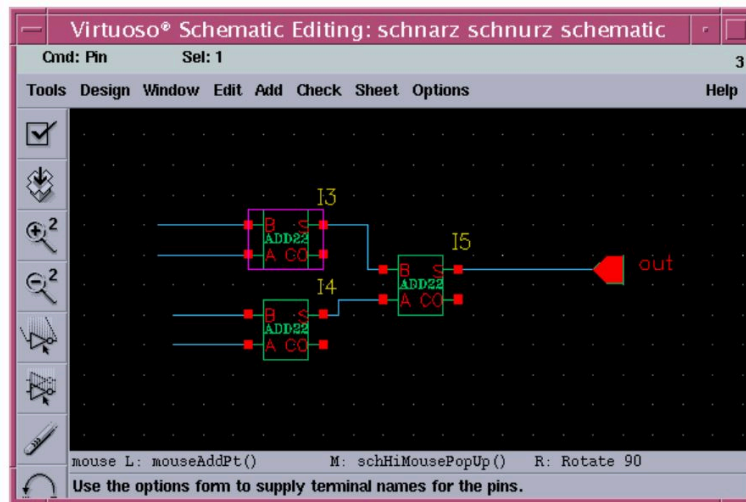
```
module 4bit_compare (A, B, equal);  
  
    input [3:0] A;  
    input [3:0] B;  
    output equal;  
  
    wire w1, w2, w3, w4, w5, w6;  
  
    XNOR # (0.2n, 0.3n)  G1 (A[0],B[0],w1);  
    XNOR # (0.2n, 0.3n)  G2 (A[1],B[1],w2);  
    XNOR # (0.2n, 0.3n)  G3 (A[2],B[2],w3);  
    XNOR # (0.2n, 0.3n)  G4 (A[3],B[3],w4);  
  
    NAND # (0.1n, 0.1n)  G5 (w1, w2, w5);  
    NAND # (0.1n, 0.1n)  G6 (w3, w4, w6);  
  
    NOR   # (0.1n, 0.2n)  G7 (w5, w6, equal);  
  
endmodule
```

Gatternetzlisten können beispielsweise mit Verilog beschrieben werden. Dort treten die Gatter als Bibliotheksaufrufe in Erscheinung. Signalübergangszeiten können mit $\#(T_{\text{fall}}, T_{\text{rise}})$ definiert werden.

Entwurfseingabe: ...Schematic Entry

Schematic Entry auf Gatterebene

- Gatter in symbolischer Darstellung aus Gatterbibliothek
- Ideale Verbindungsleitungen

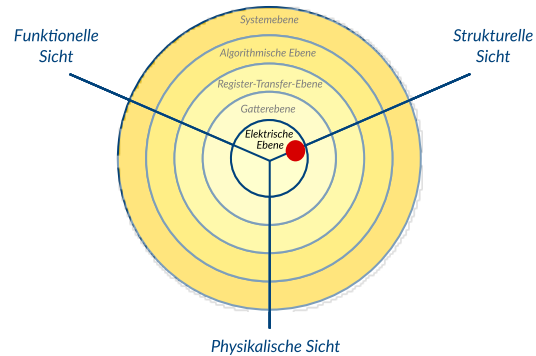


Eine Eingabe auf der Gatterebene kann auch mit Hilfe eines speziellen grafischen Editors (Schematic Entry) erfolgen. Die symbolische Darstellung der Gatter wird der Bibliothek entnommen. Die Verbindungsleitungen werden als ideal leitend angenommen.

Entwurfseingabe: Elektrische Ebene/ Strukturelle Sicht

Elektrische Ebene / Strukturelle Sicht

- Festlegung der Schaltungsstruktur (Topologie)
- Festlegung der Schaltungsdimensionierung
- Alles ist Analog!



Auf der elektrischen Ebene muss die Schaltungsstruktur (Topologie) mit den Parametern der Schaltungselemente (R, C, W, L, ...) definiert werden. Es wird nicht mehr zwischen digitalen und analogen Schaltungen unterschieden. Die eingegebenen Schaltungen bestehen aus Grundelementen (Widerstände, Kapazitäten, Induktivitäten, Transistoren, Dioden, Quellen). Die Eingabe kann in Form einer Bauelementenetzliste oder wiederum graphisch durch Schematic Entry erfolgen.

Entwurfseingabe: ...Bauelementenetzliste

Bauelementenetzliste

- Wichtigstes Format ist SPICE
(Simulation Program with Integrated Circuit Emphasis).

```
V_Vcc      6  0  DC  10
Q_Q1       4  3      5  Q2N2222
R_R1       6  3      100k
R_RC       6  4      2.2k
C_Cout     4  2      1u
C_Cin      1  3      0.1u
R_R2       3  0      47k
R_RL       2  0      47k
R_RE       5  0      1k
V_VIN      1  0  DC  -10 AC 1 sin ( 0 0.01 1000hz )
```

Eine Bauelementenetzliste ist eine Schaltungsbeschreibung auf elektrischer Ebene, die als Text in einem vom Rechner lesbaren Format eingegeben oder rechnerintern erzeugt wird. Es gibt unterschiedliche Formate von Netzlisten, wie z.B. SPICE-Format, EDIF. Am ältesten und am meisten verbreitet ist das Format des Schaltungssimulators SPICE (Simulation Program with Integrated Circuit Emphasis), zum ersten Mal vorgestellt 1972. SPICE war nur auf Großrechnern lauffähig, konnte aber Entwicklungszeiten und -kosten stark reduzieren.

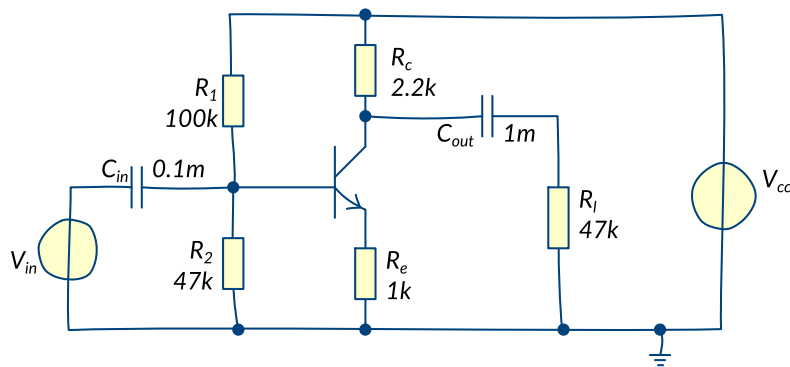
Später wurde das Simulationsprogramm SPICE von der Firma MicroSim in einer PC-Version mit dem Namen PSPICE veröffentlicht. Um PSPICE benutzerfreundlicher zu gestalten wurden ein Schaltplaneditor und ein "Software - Oszilloskop" (PROBE), was die Darstellung der Simulationsergebnisse am Monitor ermöglichte, hinzugefügt. Das Programm wurde weiterhin um Zusätze erweitert, die das Erstellen von Platinen-Layouts ermöglichen. PSPICE ist als Testversion (die Anzahl der Halbleitertypen sowie die Anzahl der Bauelemente in einer simulationsfähigen Schaltung sind eingeschränkt) kostenlos zugänglich.

Ein Nachteil des Programms ist, dass ausschließlich US-Schaltzeichen für analoge und digitale Bauelemente benutzt werden.

Entwurfseingabe: ...Schematic Entry

Schematic Entry auf elektrischer Ebene

- Graphisch unterstützte Eingabe von, aus Grundelementen aufgebauten, Schaltungen
- Bauelementebibliothek erforderlich
- Netzlister zur automatischen Erzeugung von Bauelementenetzlisten

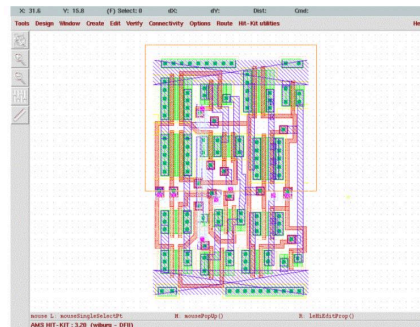
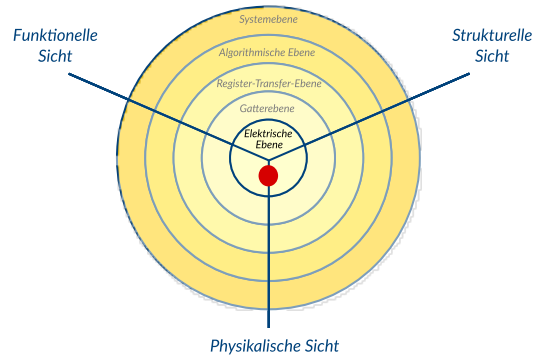


Mit Hilfe eines Schematic-Entry-Werkzeugs können Schaltungen auf elektrischer Ebene auch eingegeben werden. Für die Eingabe wird eine Bauelementebibliothek benötigt. Mit einem Netzlister können aus dem Schaltbild automatisch Bauelementenetzlisten erzeugt werden.

Entwurfseingabe: Elektrische Ebene/ Physikalische Sicht

Elektrische Ebene / Physikalische Sicht

- Layout des Chips besteht aus geometrischen Grundformen (Rechtecke, Linien, Polygone)
- Layout beschreibt Masken für die Chipfertigung
- Formate: CIF, GDSII



Auf der elektrischen Ebene und der physikalischen Sicht besteht die Schaltung aus geometrischen Grundmustern (Rechtecke, Linien, Polygone) mit denen Leiterbahnen und aktive Bauelemente beschrieben werden. Das Layout wird für die Herstellung der Fertigungsmasken benötigt. Auch in dieser Form kann eine Schaltungseingabe erfolgen. Dies ist mit einem graphischen Editor oder mit entsprechender Beschreibungssprache (-format) wie z.B. CIF (Caltech Intermediate Format) oder GDSII (Graphical Design Station II) möglich.

Caltech Intermediate Format

- 1980 publiziert von Carver Mead und Lynn Conway in "Introduction to VLSI Systems" (Technical University of California)
- ASCII-Format (lesbarer Klartext)
- Standard für den Layout-Datenaustausch an Hochschulen

```
(nmos transistor)
DS 1 1 1;
L CONT;
B 90 90 0,0;
L NPLUS;
B 290 290 0,0;
L DIFF;
B 150 150 0,0;
L MET1;
B 190 190 0,0;
DF;
DS 2 1 1;
L MET1;
W 110 -1015,310 175,310;
W 110 -1025,-290 155,-290;
L POLY1;
W 80 -1040, -5 190,-5;
L DIFF;
B 680 1120 -480,0;
C1 T-480, 310;
C1 T-480,-290;
DF;
C 2;
```

Das Caltech Intermediate Format wurde 1980 von Carver Mead und Lynn Conway ("Introduction to VLSI Systems") vorgestellt. Es ist lesbarer Klartext im ASCII-Format. CIF ist eine Art Makrosprache, die sogar Kreisstrukturen erlaubt. Dieses Format wird hauptsächlich an Hochschulen verwendet.

Entwurfseingabe: ...GDSII

Graphic Design Station II

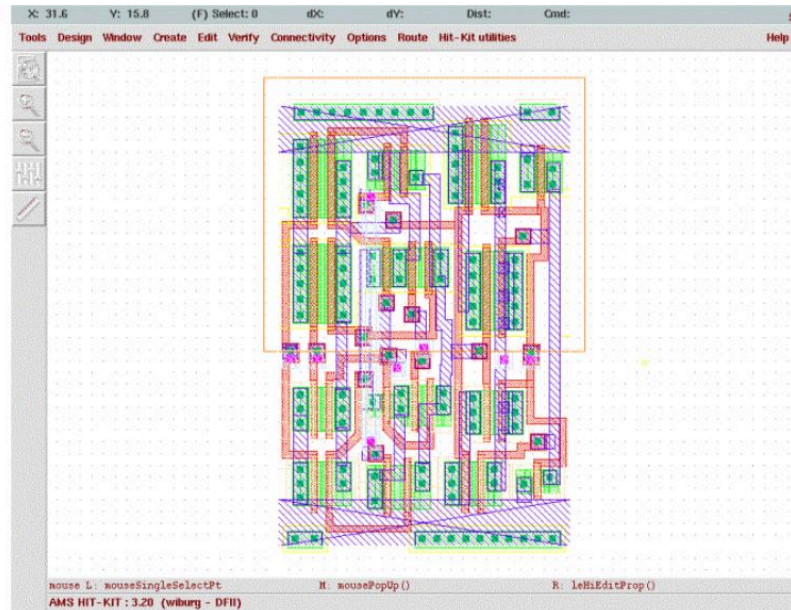
- In den 1970ern von der Firma Calma Co. eingeführt
- Eigentliches Format: STREAM
(Binärformat)
- Standard für den Layout-Daten-
austausch in der Industrie

0100011101
0011011
01111110
111100000
00110011111
11000000010
00111100
011101110
1111000
1111111110
0001
0111
11100

Was heute als GDSII-Format bezeichnet wird, hieß ursprünglich STREAM. Es war die Ausgabe der Calma Graphical Design Station II (entwickelt in den 1970ern), daher der Name GDSII. GDSII ist ein Binärformat und in der Industrie eine Art Standard für den Datenaustausch zwischen EDA-Werkzeugen oder auch zwischen Design-Teams.

Entwurfseingabe: ...Layout Entry

Layout Entry



Eine grafische Layouteingabe ist mit Ebenen-orientierten Graphikprogrammen möglich. Auf den Ebenen werden Polygone angelegt. Häufige Operationen sind: copy, mirror, rotate, stretch.