



# **Računalniško podprto načrtovanje digitalnih struktur**



# Računalniško podprto načrtovanje dig. struktur

## Pregled programskih orodij

- **minimizator** (*angl. minimizer*) je programsko orodje za avtomatizirano poenostavljanje preklopnih funkcij
- z **urejevalnikom shematskih prikazov** (*angl. schematic editor*) izrišemo simbolno shemo vezja
- **simulator vezij** (*angl. circuit simulator*) omogoča simulacijo in analizo delovanja načrtovanega vezja
- v **strojno opisnem jeziku** (*angl. hardware description language, HDL*) opišemo gradnike vezja in povezave med njimi v obliki, ki omogoča realizacijo vezja s programirljivo makrostrukturo
- **sintetizator geometrije** (*angl. layout designer*) izdelava načrt postavitve elementov in povezav na nivoju tiskanega vezja (postavitve integriranih vezij in ostalih komponent na tiskani plošči, *angl. PCB layout*) ali na nivoju integriranega vezja (postavitve tranzistorjev in ostalih elementov v čipu, *angl. IC layout*)



# Računalniško podprto načrtovanje dig. struktur

## Minimizatorji

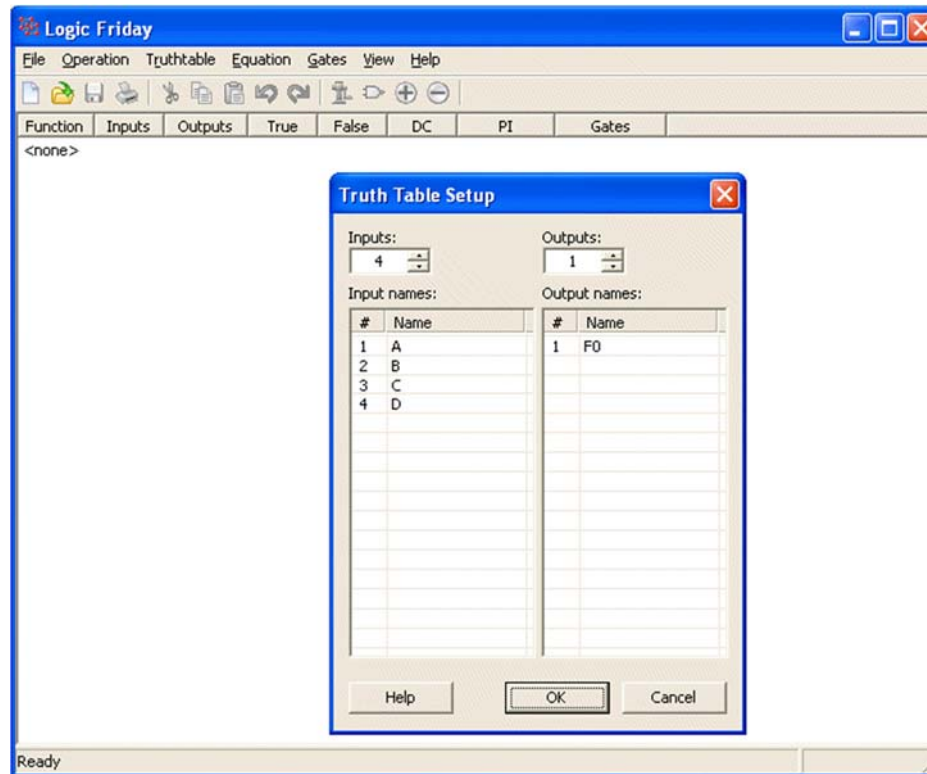
- minimizatorji omogočajo poenostavljanje preklopnih funkcij z različnimi metodami minimizacije (Quine-McCluskeyev algoritem, Petrickova metoda, algoritem Espresso, ...), prevedbe operatorjev (AND-OR  $\leftrightarrow$  OR-AND  $\leftrightarrow$  XOR  $\leftrightarrow$  NAND  $\leftrightarrow$  NOR ...) in realizacije funkcij (z MUX, PROM, PAL ...):
  - *Logic Friday*\* ([http://sontrak.com/download\\_lf.aspx](http://sontrak.com/download_lf.aspx))
  - *Minilog*\* (<http://www.brothersoft.com/minilog-download-26547.html>)
  - ...
- mnoga programska orodja za simulacijo in sintezo že vsebujejo algoritme za minimizacijo in prevedbo funkcij; če imamo na razpolago takšno orodje, ne potrebujemo ločenega minimizatorja

\* brezplačen



# Računalniško podprto načrtovanje dig. struktur

## Minimizatorji: Logic Friday

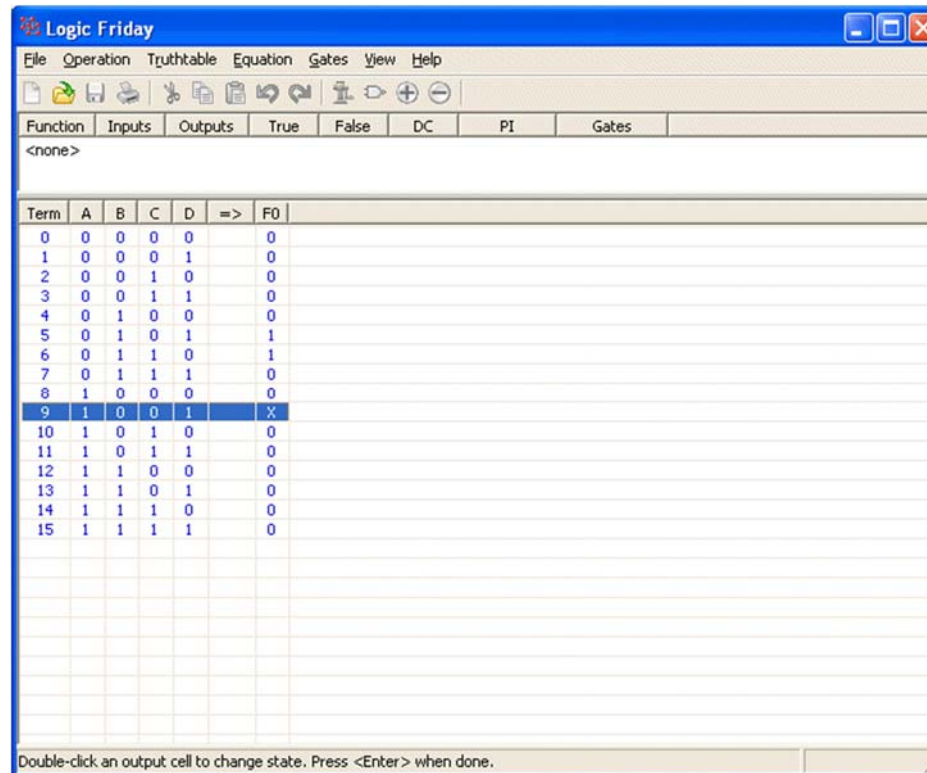


določitev vhodnih in izhodnih spremenljivk



# Računalniško podprto načrtovanje dig. struktur

## Minimizatorji: Logic Friday



Term	A	B	C	D	=>	F0
0	0	0	0	0		0
1	0	0	0	1		0
2	0	0	1	0		0
3	0	0	1	1		0
4	0	1	0	0		0
5	0	1	0	1		1
6	0	1	1	0		1
7	0	1	1	1		0
8	1	0	0	0		0
9	1	0	0	1		X
10	1	0	1	0		0
11	1	0	1	1		0
12	1	1	0	0		0
13	1	1	0	1		0
14	1	1	1	0		0
15	1	1	1	1		0

vnos podatkov v pravilnostno tabelo



# Računalniško podprto načrtovanje dig. struktur

## Minimizatorji: Logic Friday

The screenshot shows the Logic Friday interface with the following components:

- Function Table:**

Function	Inputs	Outputs	True	False	DC	PI	Gates
F0	4	1	4	12	0	Unminimized	Not mapped
F1-F2	4	2	2, 4	14, 12	0, 0	3	5
- Truth Table:**

A	B	C	D	=>	F1	F2
0	1	1	0		1	
0	1	0	1		1	
0	0	X	X			1
- Entered by truth table:**
$$F1 = A' B C' D + A' B C D';$$
$$F2 = A' B' C' D' + A' B' C' D + A' B' C D' + A' B' C D;$$
- Minimized:**
$$F1 = A' B C D' + A' B C' D;$$
$$F2 = A' B';$$
- Logic Circuit Diagram:** Shows the implementation of F1 and F2. F2 is implemented as a simple NOR gate (labeled [5]). F1 is implemented using a 2-to-1 Multiplexer (MUX, labeled [2]) with inputs from inverters (labeled [3] and [11]) and a 4-input AND gate (labeled [4]).

- (1) minimizacija v DNO  
(2) realizacija  $F_1$  z MUX, prevedba  $F_2$  v NOR



# Računalniško podprto načrtovanje dig. struktur

## Urejevalniki shematskih prikazov

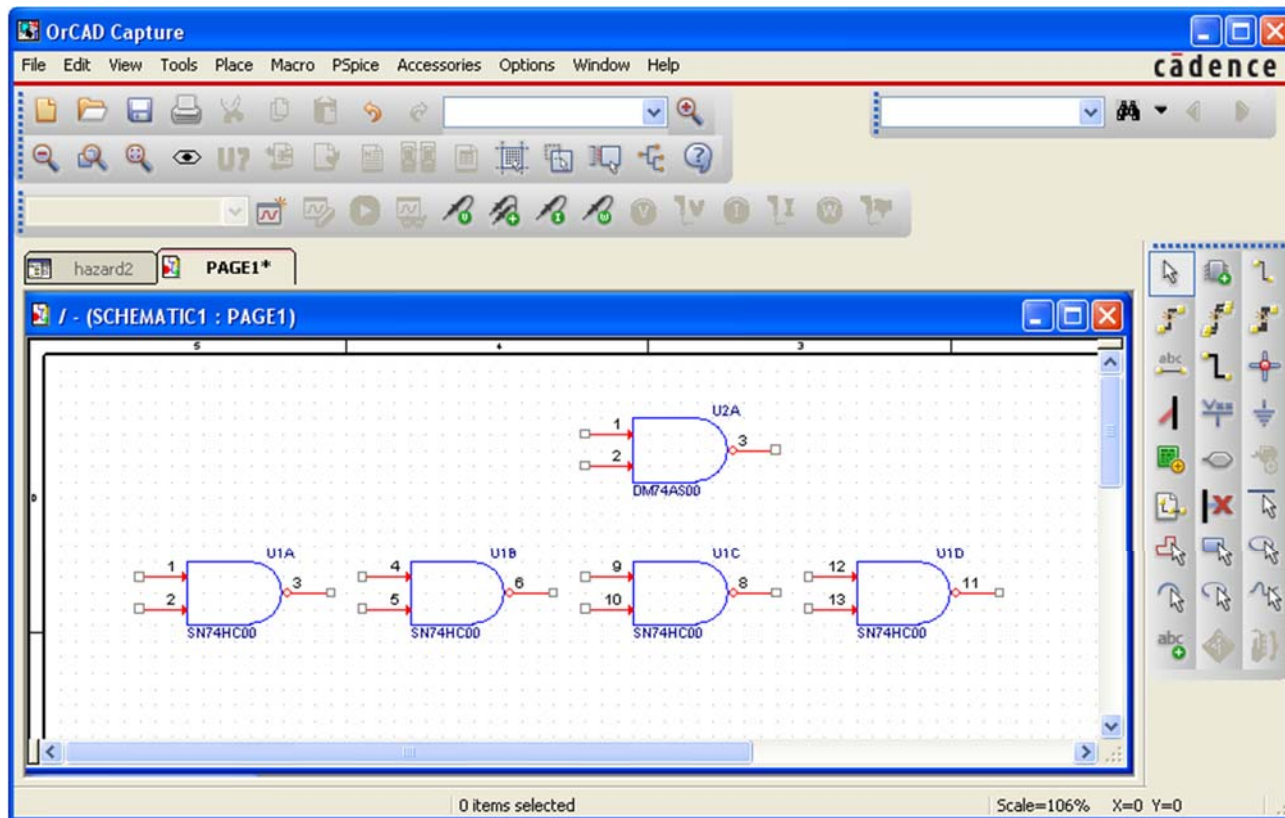
- v urejevalniku shematskih prikazov izrišemo simbolno shemo vezja, ki ga načrtujemo, kar je uporabno za dokumentiranje, predstavlja pa tudi izhodišče tako za simulacijo in analizo delovanja vezja kot za končno sintezo geometrije vezja:
  - *CadSoft Eagle*\* (<http://www.cadsoftusa.com/download.htm.en>)
  - *gEDA*\* (prej *gschem*, <http://geda.seul.org/wiki/geda:download>)
  - *NI Multisim* (prej *Electr. Workbench*; <http://www.ni.com/multisim/>)
  - *OrCAD Capture* (<http://www.cadence.com/orcad/>)
  - *TARGET 3001!*\* (<http://server.ibfriedrich.com/wiki/ibfwikien/>)
  - ...
- paketi za simulacijo in sintezo geometrije vezij imajo pogosto že vgrajene tudi tovrstne urejevalnike

\* obstaja tudi brezplačna verzija



# Računalniško podprto načrtovanje dig. struktur

## Urejevalniki shematskih prikazov: OrCAD Capture



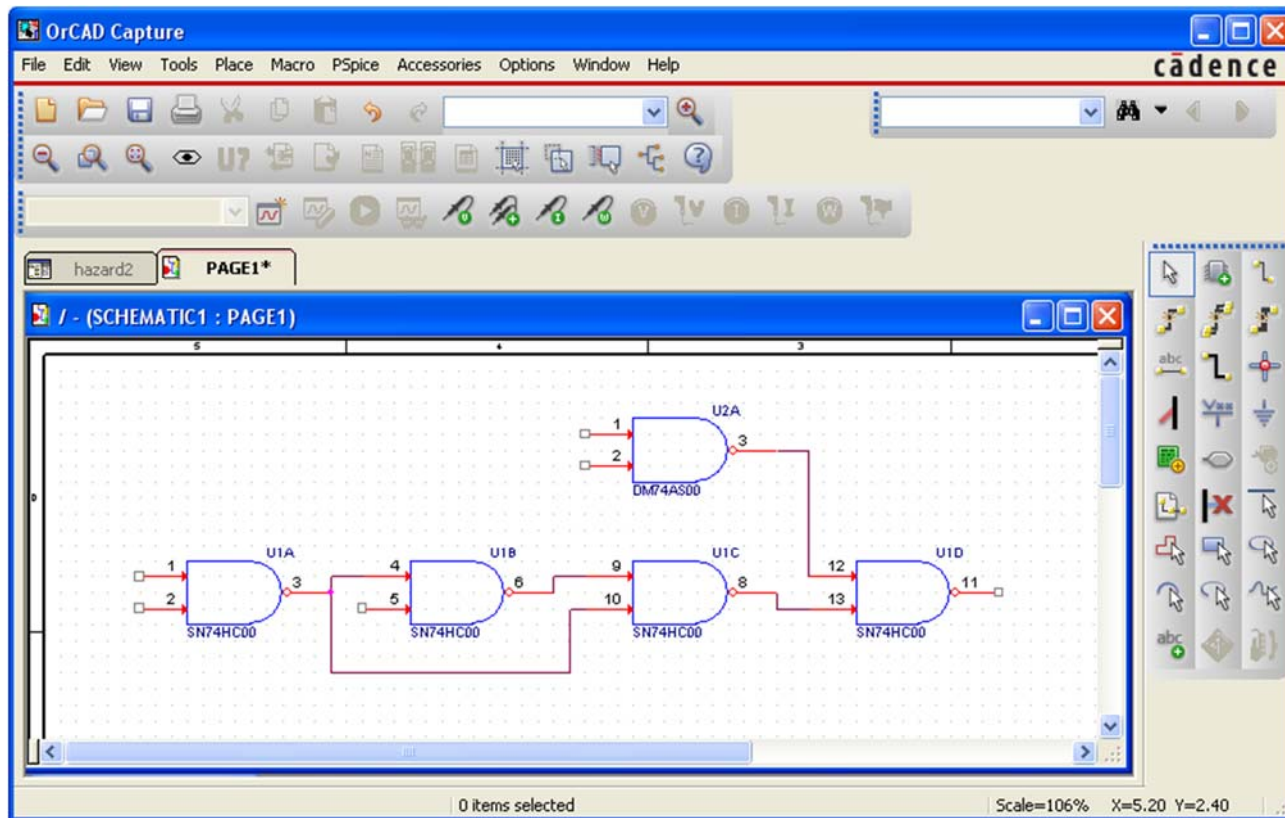
razporedimo logična vrata vezja, v katerem smo v poglavju [Preklopne funkcije in logična vrata](#) (str. 46) ugotovili pojav dinamičnega hazarda





# Računalniško podprto načrtovanje dig. struktur

## Urejevalniki shematskih prikazov: OrCAD Capture

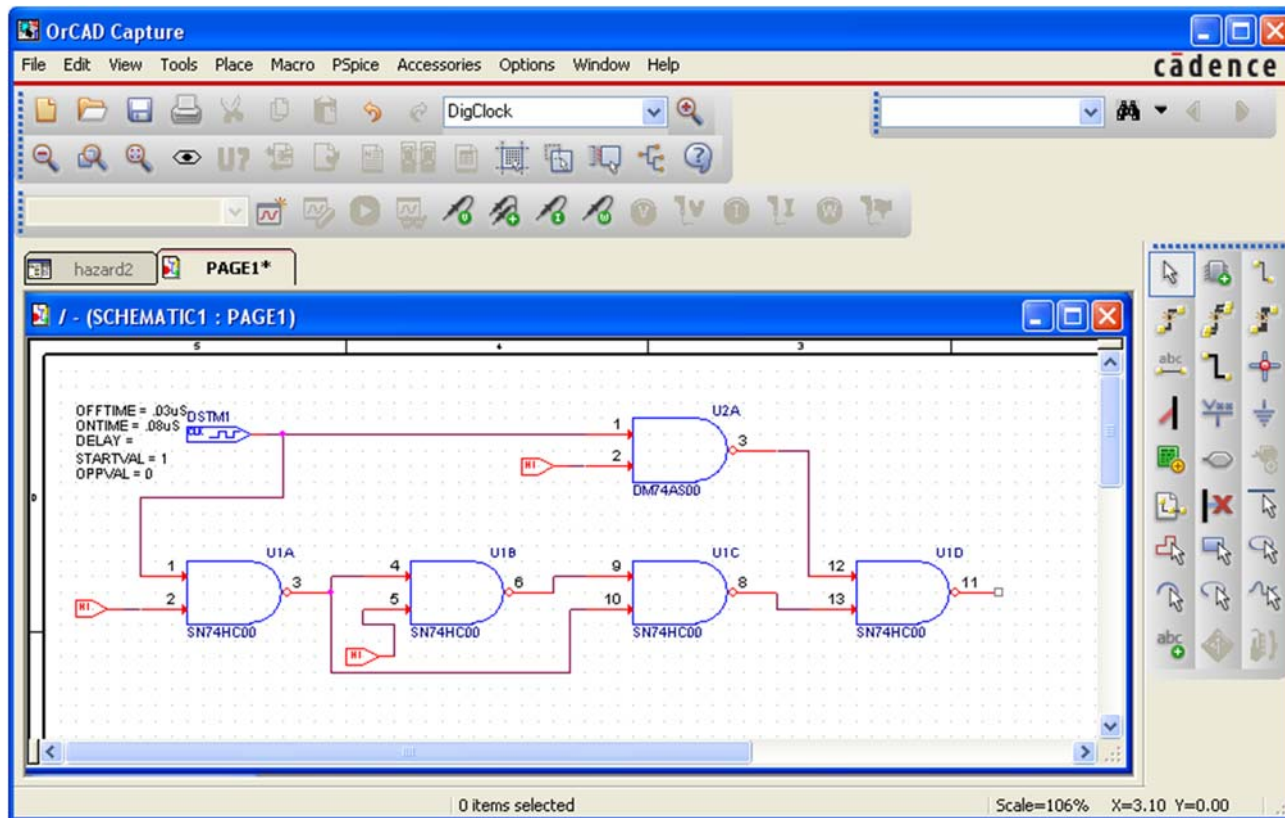


vrata povežemo med seboj v preučevano vezje



# Računalniško podprto načrtovanje dig. struktur

## Urejevalniki shematskih prikazov: OrCAD Capture



dodamo še vhodne signale



# Računalniško podprto načrtovanje dig. struktur

## Simulatorji vezij

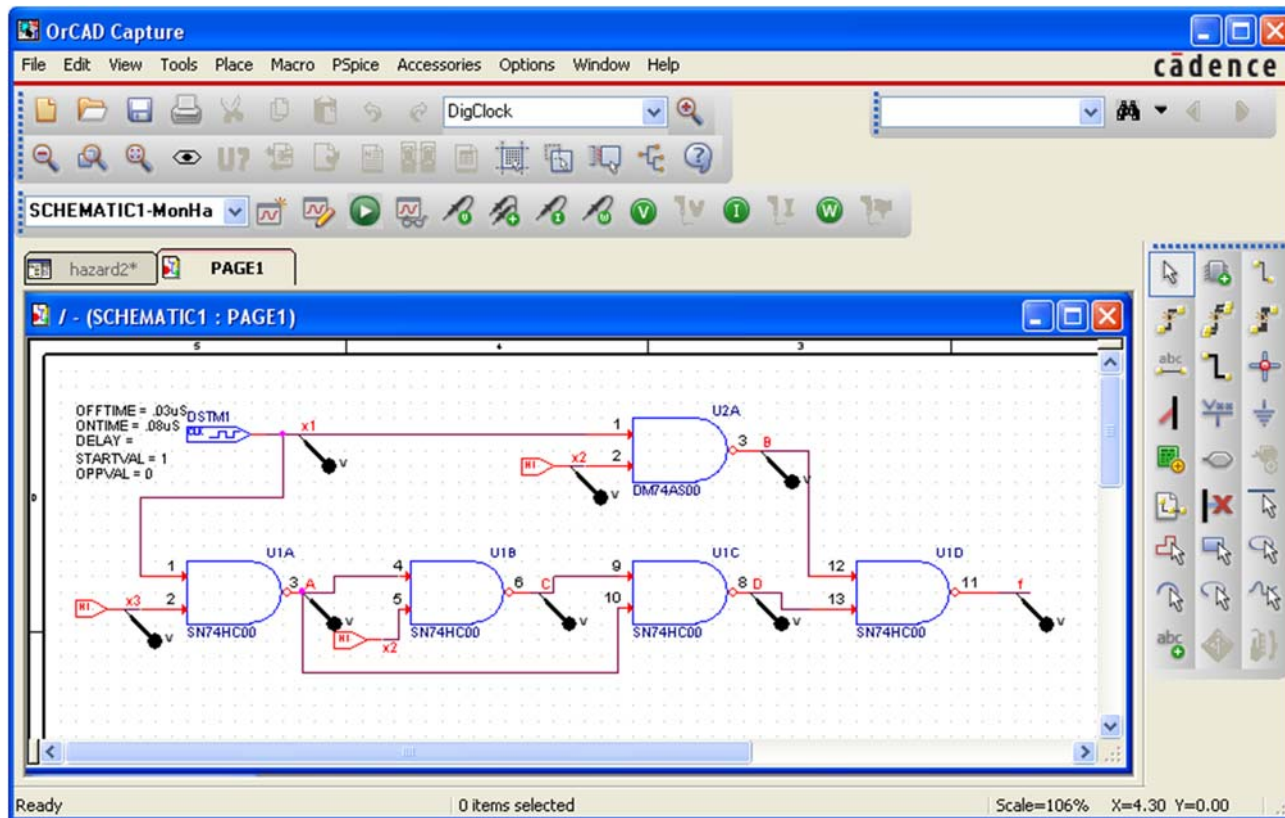
- s simulatorjem vezij preučujemo časovne poteke notranjih in izhodnih spremenljivk (preklopnih funkcij) v vezjih:
  - *Logisim*\* (<http://ozark.hendrix.edu/~burch/logisim/>)
  - *OrCAD PSpice A/D* (<http://www.cadence.com/orcad/>)
  - *Simulo*\* (<http://simulo.codeplex.com/>)
  - *Synopsis HSPICE* (<http://www.hspice.com>)
  - *Virtuoso Spectre*  
([http://www.cadence.com/products/cic/spectre\\_circuit/](http://www.cadence.com/products/cic/spectre_circuit/))
  - ...
- profesionalne izvedbe vsebujejo knjižnice s podatki o dejanskih integriranih vezjih različnih proizvajalcev in tehnologij, kar omogoča analizo zakasnitev, padcev napetosti, tokov, porabe moči ipd.


\* brezplačen



# Računalniško podprto načrtovanje dig. struktur

## Simulatorji vezij: OrCAD PSpice A/D

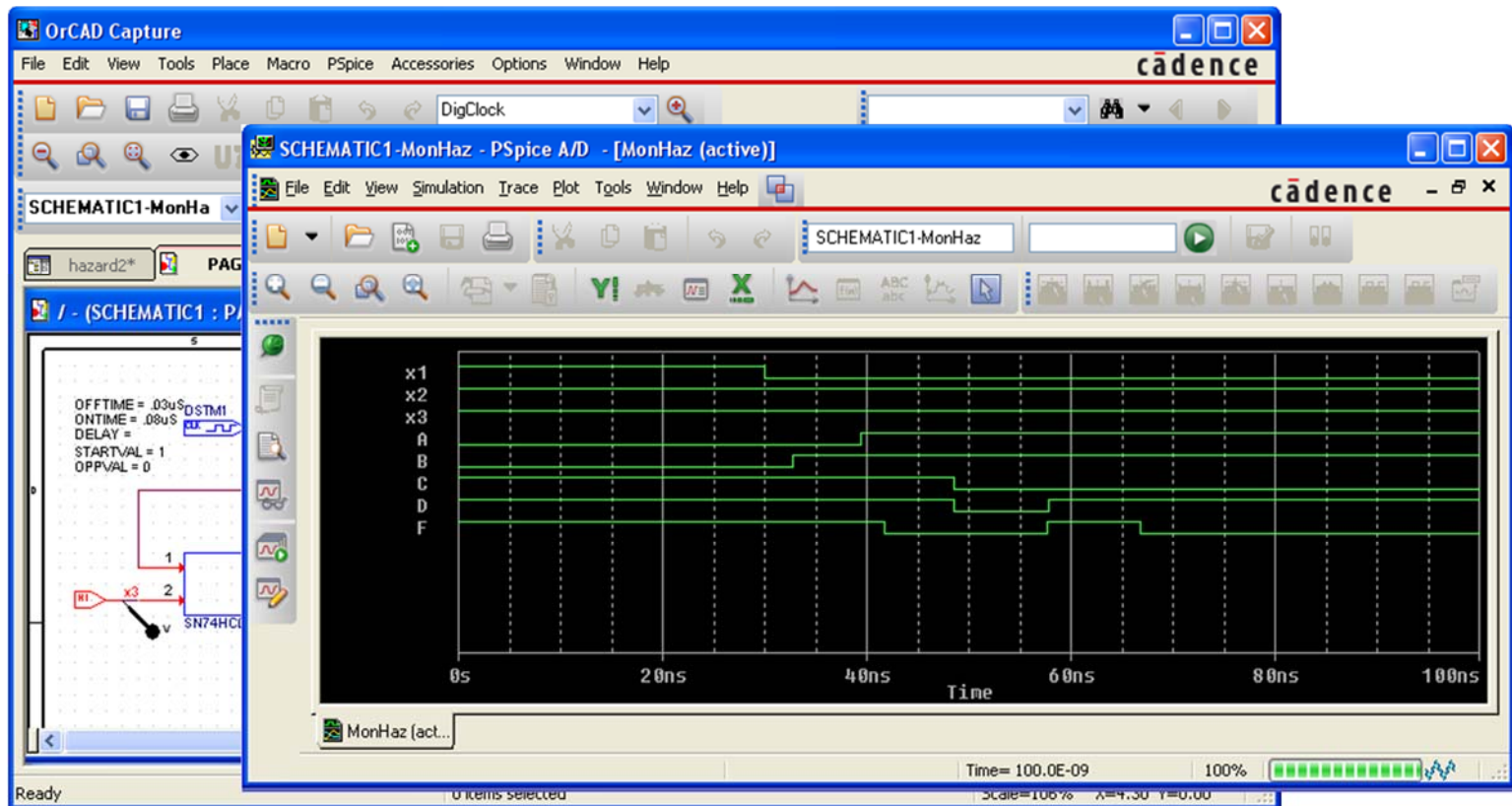


v programu OrCAD Capture poimenujemo spremenljivke ( $x_1, x_2, \dots, A, B, \dots$ ), dodamo sonde zanjihov zajem, nato pa s klikom na ikono  aktiviramo simulacijo



# Računalniško podprto načrtovanje dig. struktur

## Simulatorji vezij: OrCAD PSpice A/D



s tem se zažene program OrCAD PSpice A/D, ki izvede simulacijo delovanja vezja in prikaže časovne poteke spremenljivk, ki jih zajemamo s sondami



# Računalniško podprto načrtovanje dig. struktur

## Strojno opisni jeziki

- strojno opisni jeziki (HDL), med katerimi sta danes najbolj razširjena *VHDL* in *Verilog*, omogočajo realizacijo najrazličnejših digitalnih vezij s programirljivimi makrostrukturami (CPLD, FPGA);
- takšna realizacija poteka v treh korakih:
  - v **urejevalniku besedila** (*angl. text editor*) napišemo kodo (program), ki v sintaksi izbranega HDL opisuje gradnike vezja in povezave med njimi,
  - s **prevajalnikom** (*angl. HDL compiler*) kodo prevedemo v splošno obliko, primerno za programiranje makrostruktur
  - s **sintetizatorjem** (*angl. HDL synthesizer*) priredimo prevedeno kodo specifični makrostrukturi, v kateri nato s **programatorjem** (*angl. CPLD/FPGA programmer*) vzpostavimo povezave, ki zagotovijo njeno delovanje skladno z opisom delovanja vezja v HDL



# Računalniško podprto načrtovanje dig. struktur VHDL

The screenshot displays the VisualHDL environment. The main editor window shows the VHDL code for an entity named LEDBlinkDemo. The code includes a conditional compilation block for a simulator, a port declaration for a clock and LEDs, and a signal declaration for a counter and a divided clock. A clock divider component is instantiated at the bottom of the code. A red circle with the number (1) is placed next to the entity declaration line.

```
1 entity LEDBlinkDemo
2 {
3   #ifdef __SIMULATOR__
4     const int ClockPredividerSteps = 0;
5   #else
6     const int ClockPredividerSteps = 16;
7   #endif
8
9   port
10  {
11    in logic clk;
12    out logic[8] LEDs;
13  }
14
15  signal logic[16] Counter = 0;
16  signal logic DividedClock;
17
18  signal logic UnusedSignal; //Just to demonstrat
19
20  ClockDivider<ClockPredividerSteps> divider(
21    src = clk,
22    divided = DividedClock
```

The Project Explorer on the right shows the project structure, including the source files and generated VHDL files. The Object tree on the right shows the hierarchy of the simulation, including the entity LEDBlinkDemo and its components and signals. A red circle with the number (2) is placed next to the output window.

Output

```
Waiting for 3 sub-compilation(s) to finish...
Compiled 9 VHDL Units
Built simulation executable Project3.exe
Fuse Memory Usage: 108444 KB
Fuse CPU Usage: 202 ms
```

(1) pisanje in (2) prevajanje kode VHDL v orodju VisualHDL



# Računalniško podprto načrtovanje dig. struktur

## Programatorji



univerzalni programator vezij, ki omogoča tudi programiranje makrostruktur





# Računalniško podprto načrtovanje dig. struktur

## Sintetizatorji geometrije vezij

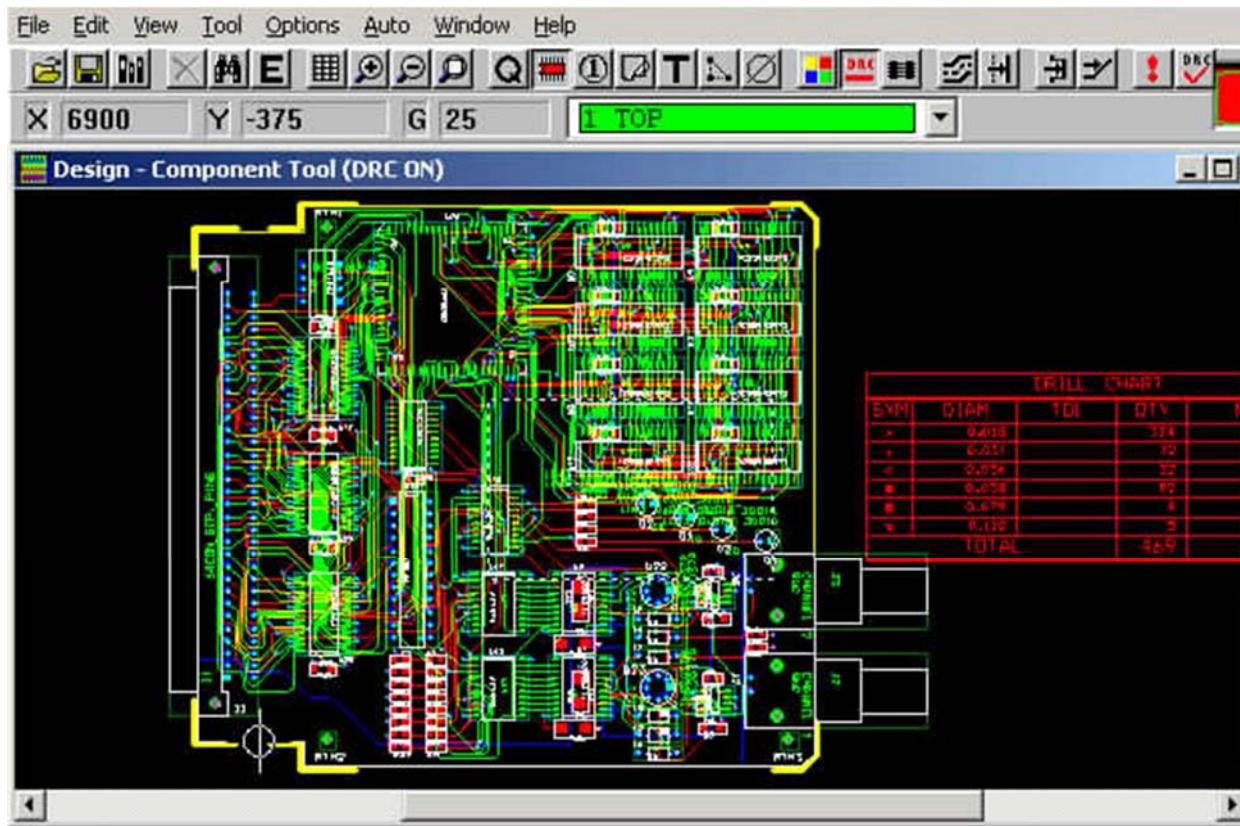
- s sintetizatorjem geometrije vezja izdelamo načrt postavitve elementov in povezav na nivoju tiskanega vezja:
  - *Allegro PCB* ([http://www.cadence.com/products/pcb/pcb\\_design/](http://www.cadence.com/products/pcb/pcb_design/))
  - *Altium Designer* (<http://www.altium.com/products/altium-designer/>)
  - *CadSoft Eagle\** (<http://www.cadsoftusa.com/download.htm.en>)
  - *OrCAD PCB Designer* (prej *OrCAD Layout*,  
[http://www.cadence.com/products/orcad/orcad\\_pcb\\_designer/](http://www.cadence.com/products/orcad/orcad_pcb_designer/))
  - ...
- ali na nivoju integriranega vezja:
  - *Encounter* ([http://www.cadence.com/products/di/edi\\_system/](http://www.cadence.com/products/di/edi_system/))
  - *Magma Hydra/Talus* (<http://www.magma-da.com/products-solutions/>)
  - *Synopsis Galaxy Custom Designer LE*
  - ...

\* obstaja tudi brezplačna verzija



# Računalniško podprto načrtovanje dig. struktur

## Sintetizatorji geometrije tiskanih vezij: OrCAD PCB Designer

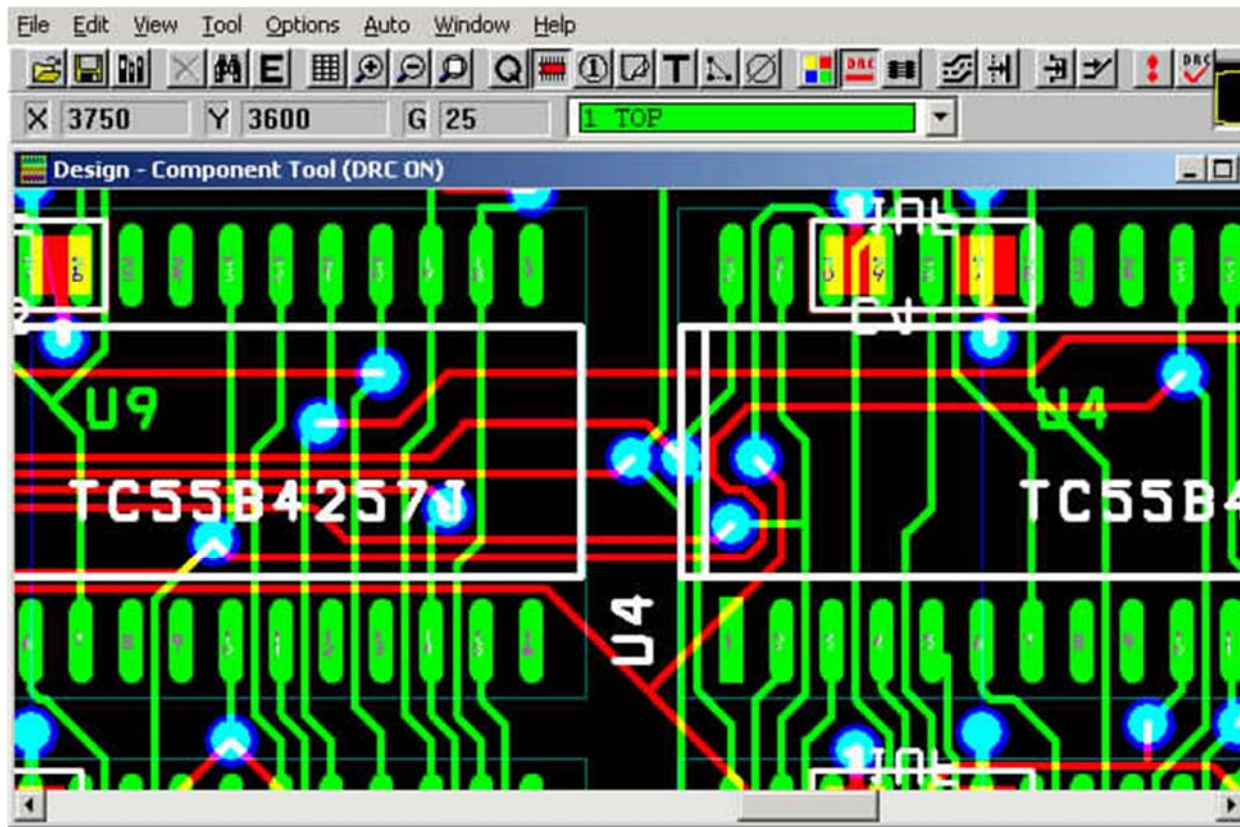


sinteza geometrije tiskanega vezja matične plošče osebnega računalnika



# Računalniško podprto načrtovanje dig. struktur

## Sintetizatorji geometrije tiskanih vezij: OrCAD PCB Designer

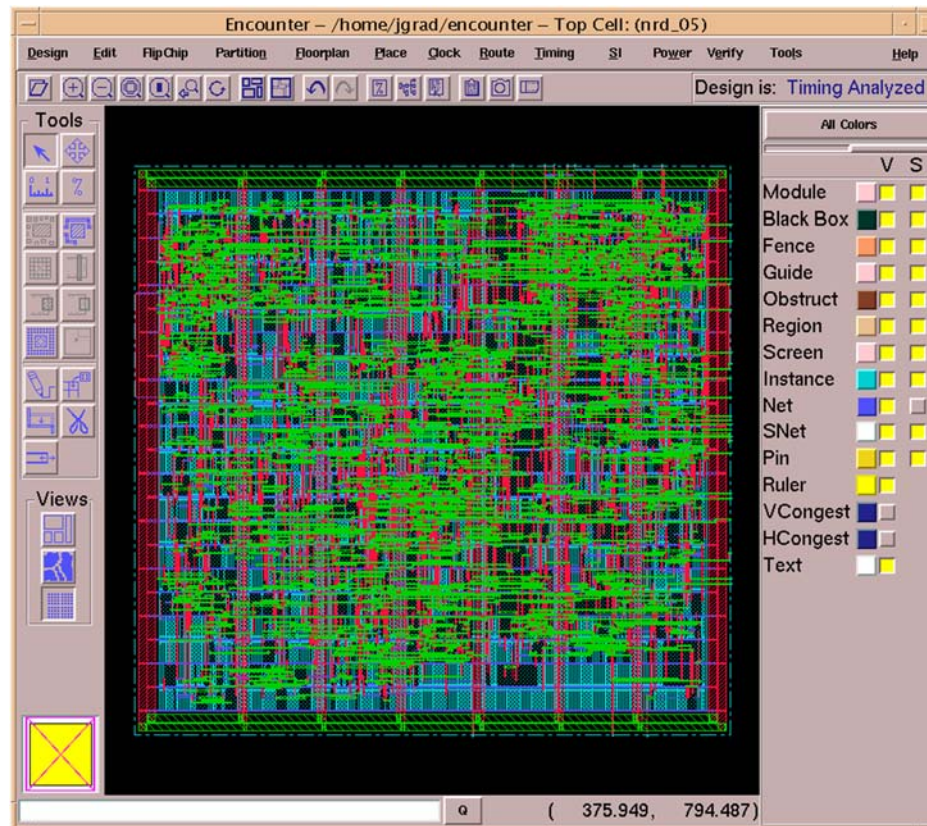


sinteza geometrije tiskanega vezja matične plošče osebnega računalnika (povečava)



# Računalniško podprto načrtovanje dig. struktur

## Sintetizatorji geometrije integriranih vezij: Encounter



sinteza geometrije integriranega vezja