

# Open-Source nástroje pro práci s FPGA

Marek Vašut <marex@denx.de>

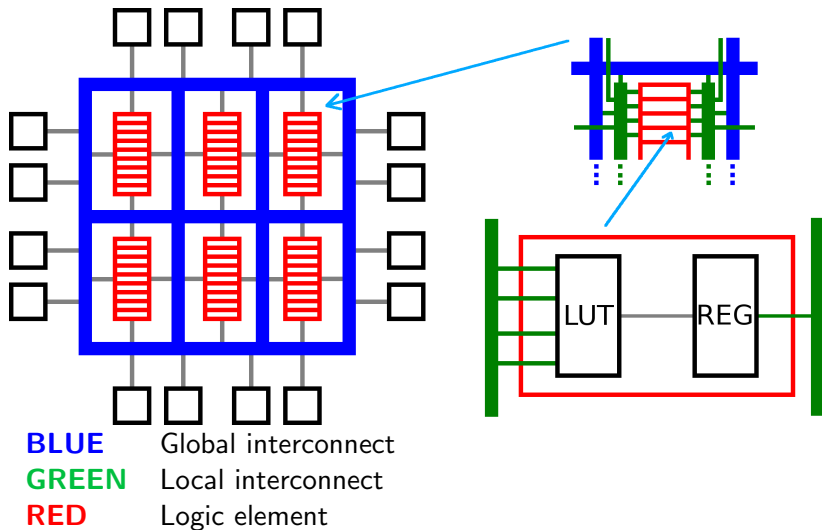
6. Listopad, 2016

- ▶ Software engineer at DENX S.E. since 2011
  - ▶ Embedded and Real-Time Systems Services, Linux kernel and driver development, U-Boot development, consulting, training
- ▶ Versatile Linux kernel hacker
- ▶ Custodian at U-Boot bootloader
- ▶ oe-core contributor

- ▶ Úvod do technologie FPGA
- ▶ Generování obsahu FPGA, od HDL k bitstreamu:
  - ▶ Nástroje pro Analýzu a Syntézu
  - ▶ Place and Route
  - ▶ Assembler
  - ▶ Simulace a vizualizace
- ▶ Ukázka
- ▶ Problém open-source nástrojů pro FPGA

- ▶ Field Programmable Gate Array
- ▶ Vysokorychlostní programovatelná logika
- ▶ Mnoho různých I/O
- ▶ Extrémně paralelní architektura
- ▶ Často se používá pro:
  - ▶ Digital Signal Processing (DSP)
  - ▶ Paralelní zpracování dat
  - ▶ Speciální hardwarová rozhraní
  - ▶ Prototypování ASICů
  - ▶ ...
- ▶ Výrobci – Xilinx, Altera, Lattice, Microsemi. . .

# Vnitřní struktura



- ▶ Každý výrobce má vlastní nástroje:  
Altera Quartus, Xilinx Vivado/ISE, Lattice Diamond, ...

- ▶ Nástroje jsou obecně closed-source

- ▶ Flow je velmi podobný:

Analysis and Synthesis  
Pack, Place and Route  
Assembler

HDL → Netlist

Netlist → Technology

Technology → Bitstream

---

Timing Analysis

Analýza časování designu

Test požadavků na časování

---

Simulation and Visualisation

Simulace a analýza  
designu na stanici

- ▶ HDL → Netlist
- ▶ Model chování → Schéma obvodu
- ▶ Analýza – Parsování HDL, validace, ...
- ▶ Synthesis – Naparsované HDL do Netlistu
- ▶ Nástroje:
  - ▶ Icarus Verilog
  - ▶ Odin II
  - ▶ Yosys

- ▶ Nástroj pro simulaci/překlad/syntézu HDL
- ▶ GPL licence (s výjimkou pro pluginy)
- ▶ Podpora pluginů
- ▶ Vstup:
  - ▶ Verilog 2005
    - ▶ Většina standardu implementována
    - ▶ Aktivně se používá
    - ▶ Aktivní vývoj
  - ▶ System Verilog – Podobná podpora jako Verilog 2005
  - ▶ VHDL – Omezená podpora
- ▶ Výstup:
  - ▶ VVP – Mezikód pro simulaci
  - ▶ Verilog – Minimalizace/zjednodušení
  - ▶ VHDL – Překlad
  - ▶ Gate-level netlist – odebráno ve verzi 0.9.1
- ▶ Website: <http://iverilog.icarus.com/>



- ▶ Framework pro syntézu HDL s podporou vizualizace
- ▶ MIT licence
- ▶ Vstup:
  - ▶ Verilog
  - ▶ BLIF netlist – z následujících fází
- ▶ Výstup: BLIF Netlist
  - ▶ Funguje přímo s VPR
  - ▶ Použitelné pro syntézu ASICů i FPGA
- ▶ Odkazy:
  - ▶ Website: <https://code.google.com/archive/p/odin-ii/>
  - ▶ Git: [https://github.com/verilog-to-routing/vtr-verilog-to-routing/tree/master/ODIN\\_II](https://github.com/verilog-to-routing/vtr-verilog-to-routing/tree/master/ODIN_II)

- ▶ Optimalizace/minimalizace množství logiky
- ▶ Většinou součást nástroje pro syntézu
- ▶ Vstup: BLIF netlist
- ▶ Výstup: BLIF netlist

- ▶ Soubor nástrojů pro syntézu HDL
- ▶ ISC licence
- ▶ Vstup:
  - ▶ Verilog 2005
  - ▶ BLIF netlist
- ▶ Výstup:
  - ▶ Zjednodušený Verilog
  - ▶ BLIF/EDIF/... netlist
- ▶ Vestavěná minimalizace/optimalizace s abc
- ▶ Podporuje namapování (překryv s PnR):
  - ▶ Knihovny ASICových buněk
  - ▶ Xilinx 7-series FPGA
  - ▶ Lattice iCE40 FPGA
- ▶ Website: <http://www.clifford.at/yosys/>

- ▶ Netlist → Technology-mapped netlist
- ▶ Skládá se z více podkroků:
  - ▶ Pack – Elementy netlistu sdruženy do větších bloků
  - ▶ Place – Bloky umístěny do FPGA
  - ▶ Route – Naroutování interconnectu mezi bloky
- ▶ Nástroje:
  - ▶ Arachne PnR
  - ▶ VPR

- ▶ Place and Route nástroj pro iCE40 FPGA
- ▶ Spolupracuje přímo s Yosys
- ▶ Vstup:
  - ▶ Technology mapped netlist z Yosys
- ▶ Výstup:
  - ▶ Textová reprezentace bitstreamu
- ▶ Website: <https://github.com/cseed/arachne-pnr>

- ▶ Versatile Placement and Routing
- ▶ Nástroj pro Pack, Place, Route
- ▶ Nyní součást VtR (Verilog to Routing)
- ▶ Velmi flexibilní
- ▶ Funguje s jakoukoli FPGA technologií
- ▶ Intenzivně se využívá ve výzkumu FPGA
- ▶ Funguje i s komerčními FPGA nástroji
- ▶ Website:  
<http://www.eecg.toronto.edu/~vaughn/vpr/vpr.html>

- ▶ Placed/Routed netlist → Bitstream
- ▶ Technologie nemá dostupnou dokumentaci
- ▶ Tento krok má nejméně nástrojů
- ▶ Nástroje:
  - ▶ IcePack

- ▶ Open-Source assembler pro iCE40 FPGA
- ▶ Součást projektu IceStorm
- ▶ Textová reprezentace bitstreamu → binární bitstream
- ▶ Website: <http://www.clifford.at/icestorm/>



- ▶ Zmíněné nástroje lze sestavit do kompletního flow
- ▶ Flow které přeloží HDL to bitstreamu:
  - ▶ IceStorm

- ▶ Flow z Verilogu do Bitstreamu
- ▶ Specifické pro Lattice iCE40 FPGA
- ▶ Nástroje:
  - ▶ Yosys – Analysis and Synthesis
  - ▶ Arachne PnR – Place and Route
  - ▶ IcePack – Bitstream generation
- ▶ Další nástroje:
  - ▶ IceProg – Programování FPGA
  - ▶ IceTime – Timing analysis
- ▶ Website: <http://www.clifford.at/icestorm/>

## Použití IceStorm, Gray counter, Top module

---

```
1 module top (  
2     input hwclk,  
3     output led1,  
4     output led2,  
5     output led3,  
6     output led4,  
7     output led5,  
8     output led6,  
9     output led7,  
10    output led8  
11 );
```

---

## Použití IceStorm, Gray counter, Top module

---

```
1  /* Counter register */
2  reg [7:0] count = 8'b0;
3  /* Grey counter implementation */
4  assign led1 = count[0] ^ count[1];
5  assign led2 = count[1] ^ count[2];
6  assign led3 = count[2] ^ count[3];
7  assign led4 = count[3] ^ count[4];
8  assign led5 = count[4] ^ count[5];
9  assign led6 = count[5] ^ count[6];
10 assign led7 = count[6] ^ count[7];
11 assign led8 = count[7];
12 /* Increment counter */
13 always @(posedge hwclk)
14     count <= count + 1;
15 endmodule
```

## Použití IceStorm, Gray counter, Mapování pinů

---

```
1 set_io --warn-no-port led1 B5
2 set_io --warn-no-port led2 B4
3 set_io --warn-no-port led3 A2
4 set_io --warn-no-port led4 A1
5 set_io --warn-no-port led5 C5
6 set_io --warn-no-port led6 C4
7 set_io --warn-no-port led7 B3
8 set_io --warn-no-port led8 C3
9 set_io --warn-no-port hwclk J3
```

---

## Použití IceStorm, Sestavení a programování

---

```
1 $ yosys -p "synth_ice40 -top top -blif top.blif" top.v
2 $ arachne-pnr -d 8k -P ct256 \  
3     -o top.txt -p pinmap.pcf top.blif
4 $ icepack top.txt top.bin
5 $ iceprog top.bin
```

---

- ▶ HDL je simulováno na vývojové stanici
- ▶ Lze aplikovat trigger pro různé podmínky
- ▶ Nástroje:
  - ▶ gHDL
  - ▶ Icarus Verilog
  - ▶ Verilator

- ▶ Simulátor VHDL
- ▶ Překládá VHDL do nativního kódu
- ▶ Používá GCC/LLVM/vestavěný backend pro generování kódu
- ▶ Rychlejší než interpretovaný simulátor
- ▶ Výstup:
  - ▶ VCD (Value Change Dump) – Vhodný pro Verilog
  - ▶ gHDL waveform – Nativní formát, vhodnější pro VHDL
- ▶ Website: <http://ghdl.free.fr/>



- ▶ Syntéza Verilogu do C++
- ▶ Verilator provádí optimalizace během syntézy
- ▶ Vstup:
  - ▶ Verilog
  - ▶ Verilog 2005 – Podporuje pouze podmnožinu
  - ▶ System Verilog – Podporuje pouze podmnožinu
- ▶ Website: <http://www.veripool.org/wiki/verilator>

- ▶ Primárně simulátor a překladač
- ▶ HDL je přeloženo do mezikódu VVP
- ▶ Nástroj vvp interpretuje VVP
- ▶ Extrémně užitečný pro psaní testbenchů
- ▶ Vizualizace výstupu: GTKWave
- ▶ Website: <http://iverilog.icarus.com/>

## Použití iVerilog, Gray counter, Testbench

---

```
1 module top_tb ();  
2  
3 reg clk;  
4 wire led1;  
5 wire led2;  
6 wire led3;  
7 wire led4;  
8 wire led5;  
9 wire led6;  
10 wire led7;  
11 wire led8;
```

---

## Použití iVerilog, Gray counter, Testbench

---

```
1 top top (  
2     .hwclk(clk),  
3     .led1(led1),  
4     .led2(led2),  
5     .led3(led3),  
6     .led4(led4),  
7     .led5(led5),  
8     .led6(led6),  
9     .led7(led7),  
10    .led8(led8)  
11 );
```

---

## Použití iVerilog, Gray counter, Testbench

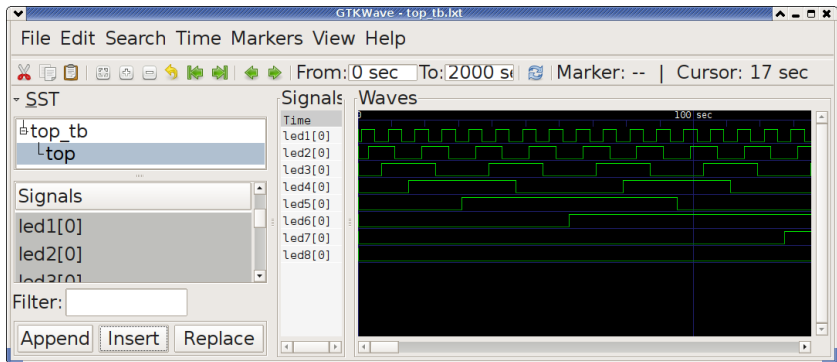
---

```
1 initial begin
2     $dumpfile("top_tb.lxt");
3     $dumpvars(0, top);
4     clk = 1'b0;
5     repeat(1000) begin
6         #1 clk = ~clk;
7         #1 clk = ~clk;
8     end
9 end
10
11 endmodule
```

---

Použití iVerilog, Gray counter, spuštění testu:

- 1 `iverilog -o top.vvp top_tb.v top.v`
- 2 `vvp top.vvp -lxt2`
- 3 `gtkwave top_tb.lxt`



- ▶ Nástroj pro vizualizaci
- ▶ Podpora mnoha formátů – VCD, LXT, FST, ...
- ▶ Funguje s gHDL, Icarus Verilog ...
- ▶ Website: <http://gtkwave.sourceforge.net/>

# Proč jsou open-source FPGA nástroj problém?

- ▶ Neexistující dokumentace
- ▶ Neochota vypustit proprietární algoritmy
- ▶ Nevole a strach ze strany výrobců IP



- ▶ Pokus o dokumentaci Altera (CII) a Xilinx FPGA
- ▶ Neaktivní
- ▶ Chybí textová dokumentace, pouze kód
- ▶ Mnoho kryptického C
- ▶ Podporuje stará FPGA
- ▶ Částečná podpora dumpování obsahu bitstreamu

# Děkuji za pozornost!

Kontakt: Marek Vasut <marex@denx.de>