

# FPGA pro BFU

Marek Vašut <marex@denx.de>

October 10, 2015

- ▶ Custodian at U-Boot bootloader
- ▶ Linux kernel hacker
- ▶ oe-core contributor (Yocto/OE/Poky)
- ▶ FPGA enthusiast

- ▶ Co jsou PLD, historie PLD
- ▶ Dostupné vývojové kity
- ▶ Jak programovat PLD a demo

Programovatelný obvod umožňující implementovat uživatelskou logickou funkci

▶  $Y = f(A_0, \dots, A_n) \quad A_0, \dots, A_n \in \{0, 1\}; Y \in \{0, 1\}^*$

## Co lze s PLD dělat?

- ▶ Blikat LED :-)
- ▶ Samplování rychlých dějů (např. z ADC)
- ▶ Generování rychlých dějů (např. pro DAC)
- ▶ Implementace obskurních sběrnic a protokolů
- ▶ Rychlé paralelní transformace (např. obrazu)
- ▶ ...
- ▶ Syntetizovat CPU a nabootovat např. Linux
  - ▶ Existuje model ARM (arm7tdmi, arm9, ...)
  - ▶ Existuje model SuperH2 (J-core, J2)
  - ▶ Existuje model OpenRISC, RISC-V, ...
  - ▶ Existuje model m68k, Z80, MOS6502
  - ▶ Existuje model i486SX (ao486), Pentium (v586) ...

```
Initializing cgroup subsys cpu
Linux version 4.3.0-rc1-next-20150918-dirty (marex@bfu)(gcc version 5.2.0 (GCC) ) #86 Sat Sep ...
bootconsole [early0] enabled
early_console initialized at 0xe8004d50
[...]
Memory: 21140K/294912K available (2604K kernel code, 88K rwdata, 496K rodata, 5584K init, 102K...
NR_IRQS:64 nr_irqs:64 0
clocksource: nios2-clksrc: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 38225208935 ns
Console: colour dummy device 80x25
Calibrating delay loop (skipped), value calculated using timer frequency.. 100.00 BogoMIPS (lp...
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 1024 (order: 0, 4096 bytes)
Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes)
devtmpfs: initialized
clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 7645041785100000 ns
NET: Registered protocol family 16
clocksource: Switched to clocksource nios2-clksrc
[...]
Poky (Yocto Project Reference Distro) 1.8+snapshot-20150918 generic-nios2 /dev/ttyJ0
```

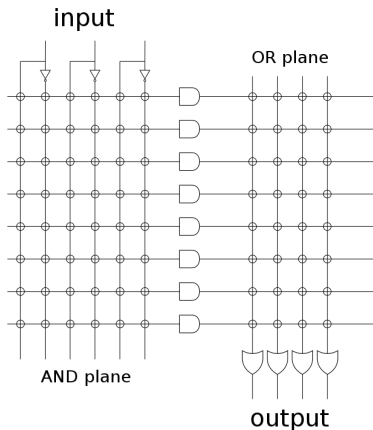
```
generic-nios2 login: root
root@generic-nios2:~# cat /proc/cpuinfo
CPU:           Nios II/fast
MMU:          present
FPU:          none
Clocking:     50.00 MHz
BogoMips:    100.00
Calibration:  50000000 loops
HW:
  MUL:        yes
  MULX:       no
  DIV:        yes
Icache:      4kB, line length: 32
Dcache:      4kB, line length: 32
TLB:         16 ways, 128 entries, 8 PID bits
```

Rok ~1970

- ▶ Užití PROM pro implementaci logické funkce
- ▶ PLA:
  - ▶ Programmable Logic Array
  - ▶ 1970: TMS2000 – programování při výrobě pomoci masky
  - ▶ Později: PLA s programovatelnou AND a OR maticí

## První polovina 198x

- ▶ PAL  
Programmable Array of Logic
  - ▶ 1978: MMI PAL
  - ▶ Programovatelný AND  
Fixní OR
  - ▶ Programoval se pomocí  
OTP fuses
- ▶ GAL  
Generic Array of Logic
  - ▶ 1985: Lattice GAL
  - ▶ Fuses se dají přeprogramovat



PLA – Ilija Kr.

CC BY-SA 3.0:

<https://creativecommons.org/licenses/by-sa/3.0/deed.en>

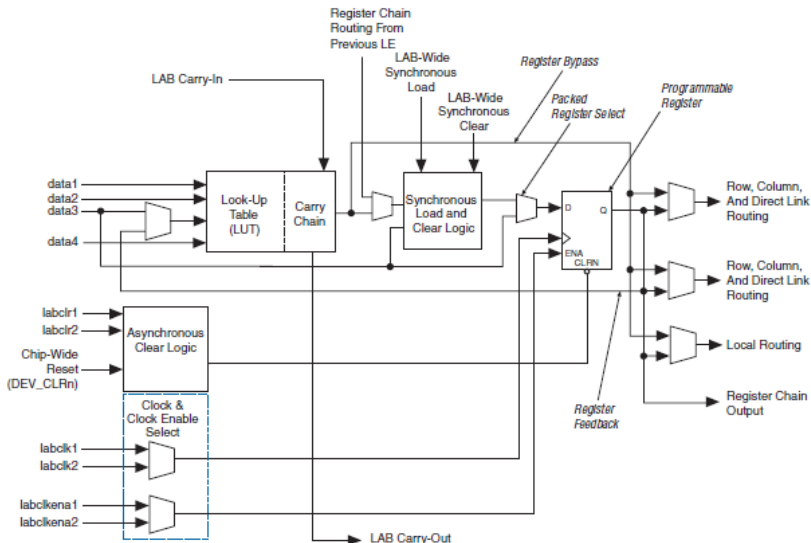


## Druhá polovina 1980x

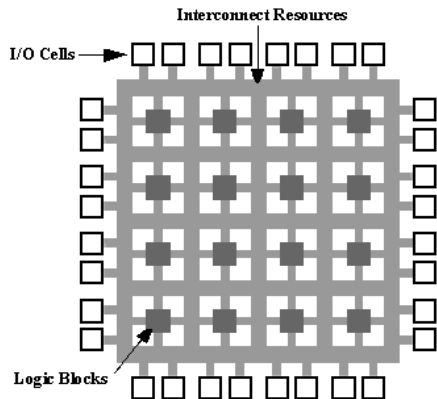
- ▶ Complex Programmable Logic Device
- ▶ Složeno z Macrocells (architektura podobná GAL)
- ▶ Macrocells propojené přes "programmable interconnect"
- ▶ CPLD si drží naprogramovanou funkci i po odpojení energie
  - ▶ CPLD fungují okamžitě po připojení napájení
  - ▶ ⇒ Vhodné pro distribuci hodin na desce
  - ▶ CPLD obsahují paměť FLASH
- ▶ Problém "Sea of gates"

## Druhá polovina 197x

- ▶ 197x: Signetics 82S100
- ▶ 1984: Altera EP300 / 1985: Xilinx XC2064
- ▶ Řeší problém "Sea of gates"
- ▶ Založeno na LUT tabulkách a SRAM:
  - ▶ Nejmenší blok je LE (logic element)
  - ▶ Sdruženo do LAB (logic array block)
  - ▶ LAB propojeny přes "programmable interconnect"
- ▶ FPGA obsahují i speciální bloky – paměti, násobičky, PLL, ...



Altera Cyclone II LE



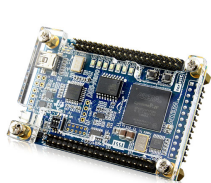
W.T.Freeman

<http://www.vision.caltech.edu/CNS248/Fpga/fpga1a.gif>

CC BY 2.5: <http://creativecommons.org/licenses/by/2.5/>

- ▶ Terasic DE0-Nano – Cyclone IV / E
- ▶ Terasic DE0-Nano-SoC – Cyclone V / SE
- ▶ WaveShare CoreEP4CE6 – Cyclone IV / E

Pozor, Cyclone III a starší nebrat, nový Quartus II je nepodporuje.



DE0-Nano



DE0-Nano-SoC



CoreEP4CE6

- ▶ Avnet Microboard – Spartan 6
- ▶ Papilio Duo Pro – Spartan 6
- ▶ Numato Mimas V2 – Spartan 6



Microboard



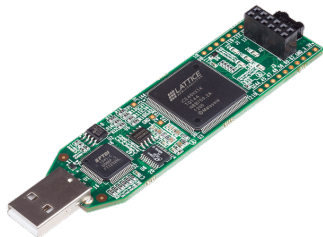
Papilio Duo



Mimas V2

## iCE40 IceStick

- ▶ Open-Source toolchain – Project IceStorm



- ▶ Každý výrobce má vlastní nástroje:
  - ▶ Xilinx Vivado / Xilinx ISE
  - ▶ Altera Quartus II
- ▶ Nástroje jsou obecně closed-source :-)
- ▶ Formát souboru s obsahem FPGA je proprietární :-)
- ▶ Základní nástroje jsou dostupné zdarma i pro Linux
- ▶ Obsahují dostatek funkcionality



- ▶ Obsah FPGA se modeluje v HDL
- ▶ HDL – Hardware Description Language
- ▶ Dnes dva hlavní HDL – Verilog a VHDL
- ▶ Nastroje umožňují HDL v projektech kombinovat
- ▶ Repositáře projektu se svobodnou licenci:
  - OpenCores <http://opencores.org/>
  - OpenCores projects <http://opencores.org/projects>
  - CERN Open HW Repo <http://www.ohwr.org/>

Modelování v HDL vs. psaní kódu pro CPU:

- ▶ Velmi odlišné a matoucí pro programátory :-)
- ▶ CPU: Programátor implementuje sekvenci popis algoritmu
- ▶ FPGA: Programátor implementuje hardware který vykonává funkci

- ▶ Kombinatorická – Vystup je okamžitý produkt vstupu

```
assign Z = X ^ Y;
```



- ▶ Sekvenční logika – Vystup je synchronizován hodinami

```
always @(posedge clk)  
    Z <= DAT;
```



- ▶ Vše se spouští paralelně
- ▶ DEMO: podmíněný výraz s přiřazením:

---

```
1 if      (foo && bar)      quux <= 2'b00;  
2 else if (foo && !bar)    quux <= 2'b10;  
3 else                                     quux <= 2'b01;
```

---

## DEMO: Blikání LED

---

```
1 module blink (  
2     input          clk,  
3     input          reset,  
4  
5     output reg [7:0] led  
6 );  
7  
8 always @(posedge clk) begin  
9     if (reset)  
10        led <= 8'h00;  
11    else  
12        led <= led + 1;  
13 end  
14  
15 endmodule
```

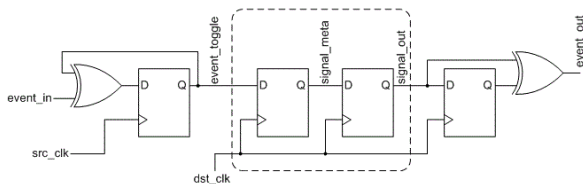
## Instanciacie modulu

---

```
1 module top (  
2     input          CLOCK50,      /* 50 MHz clock */  
3     input  [7:0]  KEY,  
4     output [7:0]  LED  
5 );  
6  
7 blink b (  
8     .clk(CLOCK50),  
9     .reset(KEY[0]),      /* Debouncing problem */  
10    .led(LED)  
11 );  
12  
13 endmodule
```

---

- ▶ FPGA je synchronní chip
- ▶ Lze mít více hodinových domén
- ▶ Lze samplovat externí signály
- ▶ Přechod mezi doménami je potřeba ošetřit
- ▶ Existuje více možností:
  - ▶ Jeden signál – Rada registru
  - ▶ Citace – Aplikace Grayových kódů
  - ▶ Rychlá data – Dual-port RAM jako FIFO



## Synchronizace tlačítka

---

```
1 module top (  
2     input          CLOCK50,      /* 50 MHz clock */  
3     input  [7:0]  KEY  
4 );  
5  
6 reg  [2:0] sync;  
7 always @(posedge CLOCK50)  
8     sync <= { sync[1:0], KEY[0] };  
9 wire risingedge = (sync[2:1] == 2'b01);  
10  
11 ...  
12  
13 endmodule
```

---



- ▶ Simulace (na počítači)
- ▶ Probing (na FPGA)

- ▶ Simulační nástroje:
  - Icarus Verilog <http://iverilog.icarus.com/>
  - ghdl <http://home.gna.org/ghdl/>
  - ModelSim <http://en.wikipedia.org/wiki/ModelSim/>
- ▶ Implementuje se testcase v rozšířeném HDL
- ▶ Po spuštění testcase lze studovat výsledek
  - ▶ Zobrazit waveformy různých signálů
  - ▶ Dekódovat a analyzovat sběrnice
  - ▶ Provádět triggerování na komplexních podmínkách
  - ▶ ...

- ▶ Používá se ke sledování chování FPGA
- ▶ Lze si představit jako bus analyzer v FPGA
- ▶ Nastroje pro probing jsou např. Altera SignalTap
- ▶ FPGA design se rozšíří o IP core pro probing
- ▶ Probing core se řídí pomocí debugovací sondy (např. přes JTAG)
- ▶ Pozor, probe core ukládá data do FPGA a FPGA má omezené množství interní paměti

- ▶ HDL soubory – základní stavební kamen
- ▶ IP block – kolekce HDL souboru
- ▶ FPGA design – kolekce IP bloku
- ▶ Pro usnadnění práce s komplexními designy dodávají výrobci FPGA high-level tooly, např. Altera QSys.

- ▶ Srozumitelný článek o FPGA (CZ)
- ▶ FPGA4FUN: Vysvětlení detailu FPGA (EN)
- ▶ ASIC-WORLD: Verilog detailně (EN)
- ▶ Exploring Arrow SoCkit (EN)
- ▶ OpenCores:IP cores se svobodnou licenci (EN)
- ▶ CERN OHWR (EN)
- ▶ Icarus: Překladač HDL Verilog (EN)

# Děkuji za pozornost!

Kontakt: Marek Vašut <marex@denx.de>