Project Trellis & nextpnr

FOSS Tools for ECP5 FPGAs

David Shah
@fpga_dave
Symbiotic EDA || Imperial College London

FPGA?

- Programmable digital logic
- Typical basic elements are look-up-tables and D-flipflops; connected by programmable switches
- Configured by a **bitstream** which sets all this up
- Most FPGA development uses closed-source tools, FPGA vendors don’t document bitstreams

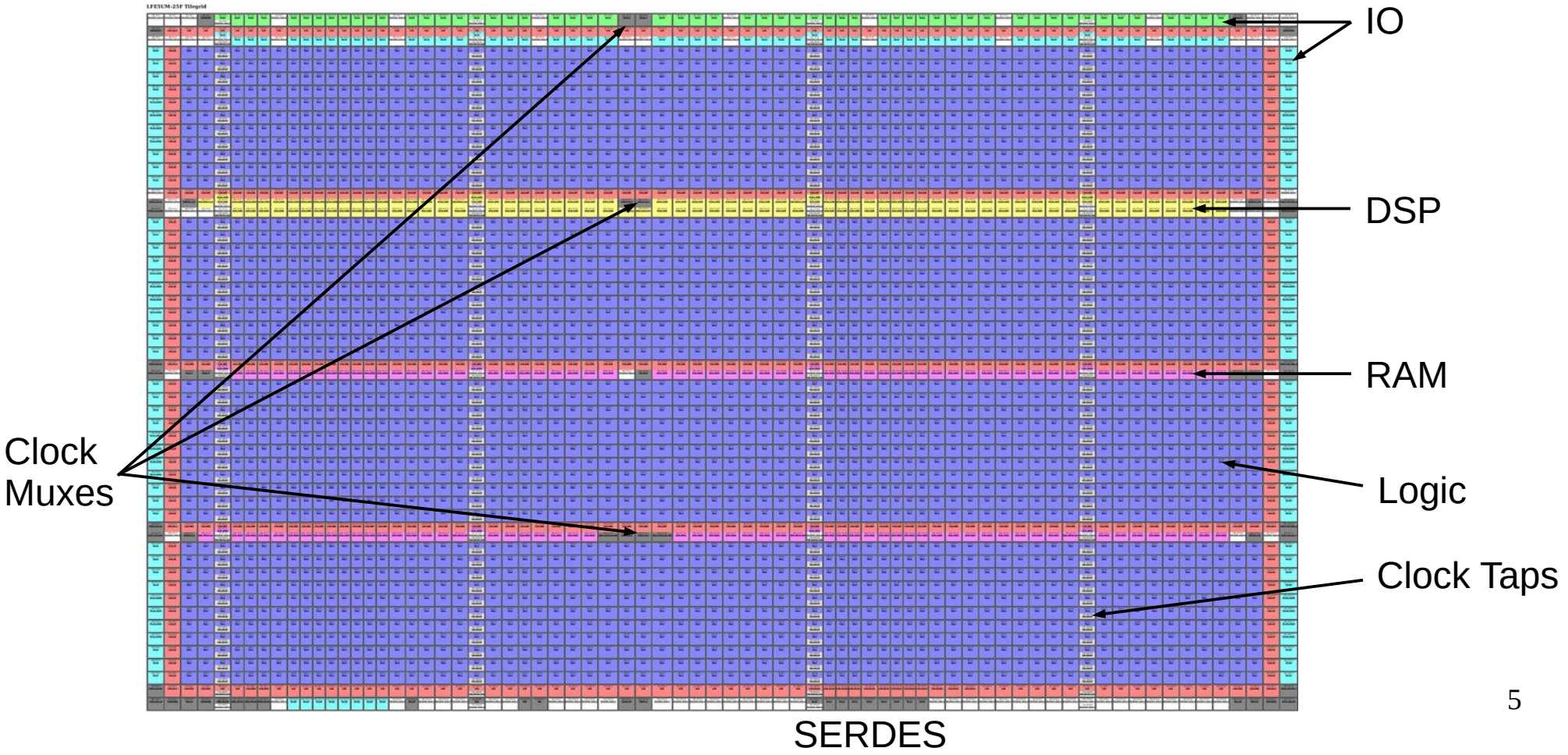
ECP5 FPGA

- Up to 85k logic cells (LUT4+carry+FF)
- Up to 3.7Mb block RAM (in 18Kb blocks), 156 18x18 DSPs
- Available with 3Gbps or 5Gbps SERDES for PCIe, USB 3.0, etc
- Single-quantity pricing starts from $5 (“12k” LE)

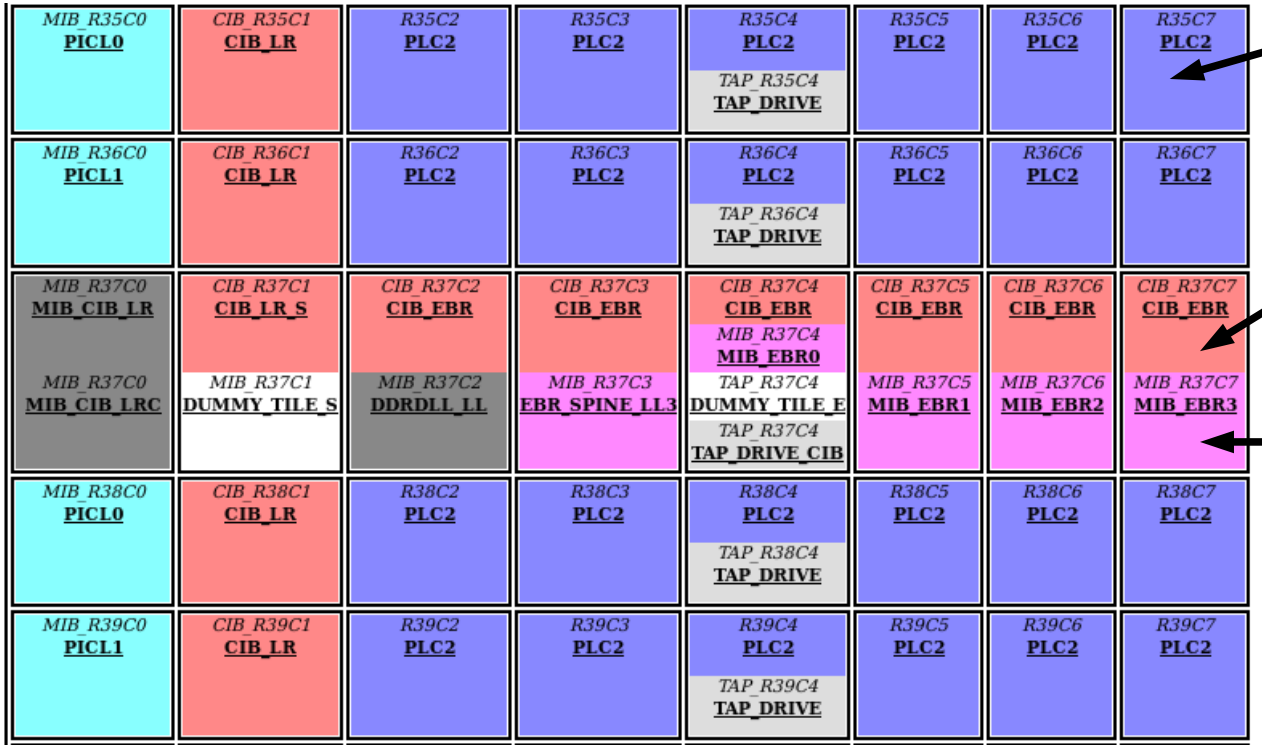
ECP5 Architecture

- Split up into **tiles** of different types. Logic tiles split into 4 **slices**
- **Slice:** 2 LUT + 2FF; carry + 2FF; 16x2 RAM + 2FF; also cascade muxes
- Fixed interconnect **wires**
- **Arcs** connect **wires** together and are configurable or fixed (aka **pip**)
- All arcs and wires are unidirectional – mux topology
- Dedicated global clock network connects to all tiles

ECP5 Architecture



ECP5 Architecture



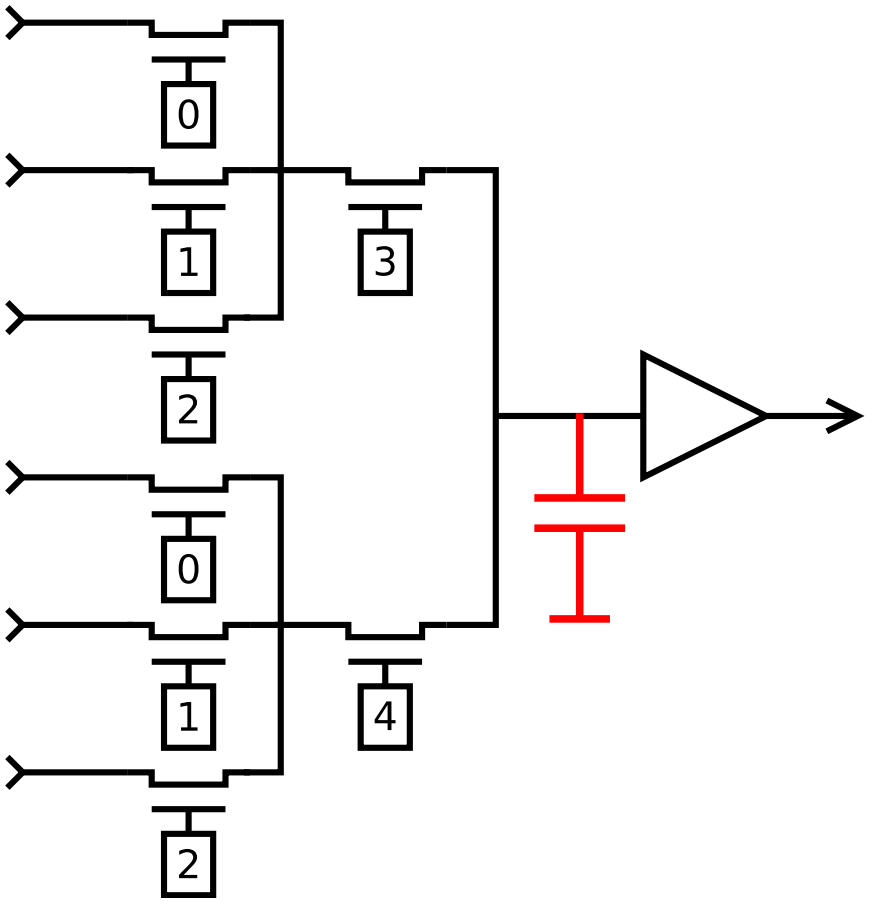
Logic tiles contain both logic and interconnect

CIB tiles contain interconnect for non-logic functions

MIB tiles contain non-logic functionality (EBR, DSP, IO, etc)

More than one tile at a location is possible!

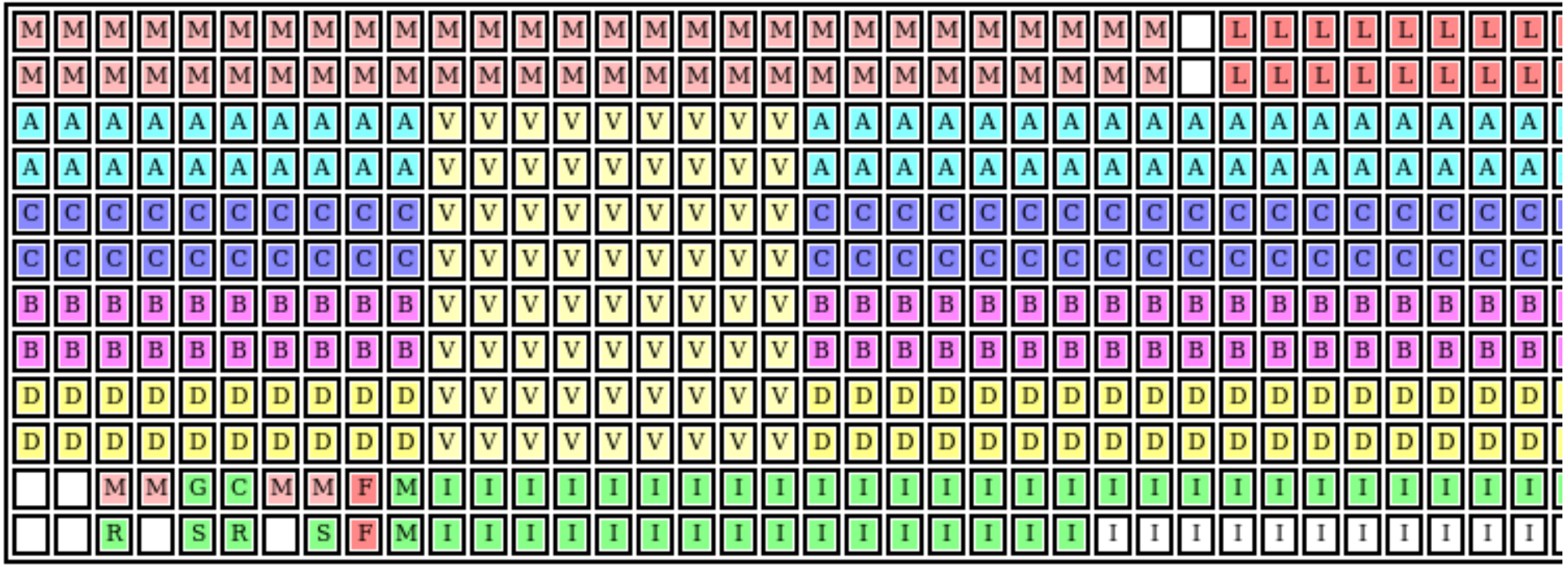
ECP5 Architecture



Current Status

- Bit and routing documentation for almost all functionality (missing: obscure DSP modes)
- Timing documentation for fabric, logic cells, IO and BRAM
- Timing-driven Yosys & nextpnr flow supporting majority of functionality

Database



Database

Normalised netname
Nominal position is x+3

Frame 104, bit 9
inside tile

Mux driving E3_H06E0003

Source	F101B8	F101B9	F102B8	F103B8	F103B9	F104B8	F104B9	F105B9
W3_H06E0003	1	-	-	1	-	-	-	-
S3_V06N0003	-	1	-	1	-	-	-	-
V06N0003	-	-	1	1	-	-	-	-
W3_H06E0303	1	-	-	-	1	-	-	-
V06S0003	-	1	-	-	1	-	-	-
N3_V06S0003	-	-	1	-	1	-	-	-
W1_H02E0001	1	-	-	-	-	1	-	-
N1_V01S0000	-	1	-	-	-	1	-	-
V01N0001	-	-	1	-	-	1	-	-
W1_H02E0301	1	-	-	-	-	-	1	-
Q0	-	1	-	-	-	-	1	-
Q3	-	-	1	-	-	-	1	-
H01E0001	1	-	-	-	-	-	-	1
F0	-	1	-	-	-	-	-	1
F3	-	-	1	-	-	-	-	1

Database

Configuration word SLICEA.K0.INIT

Default value: 16'b1111111111111111

SLICEA.K0.INIT[0]	!F25B10
SLICEA.K0.INIT[1]	!F24B10
SLICEA.K0.INIT[2]	!F23B10
SLICEA.K0.INIT[3]	!F22B10
SLICEA.K0.INIT[4]	!F21B10
SLICEA.K0.INIT[5]	!F20B10
SLICEA.K0.INIT[6]	!F19B10
SLICEA.K0.INIT[7]	!F18B10
SLICEA.K0.INIT[8]	!F17B10
SLICEA.K0.INIT[9]	!F16B10
SLICEA.K0.INIT[10]	!F15B10
SLICEA.K0.INIT[11]	!F14B10
SLICEA.K0.INIT[12]	!F13B10
SLICEA.K0.INIT[13]	!F12B10
SLICEA.K0.INIT[14]	!F11B10
SLICEA.K0.INIT[15]	!F10B10

Database

Configuration Setting EBR0.DP16KD.DATA_WIDTH_A

Default value: 18

Value	F40B0	F47B0	F51B0	F78B0
1	1	1	1	1
2	1	0	1	1
4	1	0	1	0
9	0	0	1	0
18	0	0	0	0

Configuration Setting EBR0.DP16KD.WRITEMODE_A

Default value: NORMAL

Value	F7B0	F101B0
NORMAL	0	0
READBEFOREWRITE	0	1
WRITETHROUGH	1	0

Text Configuration

- Need to make use of & test fuzz results
- Tools to convert bitstreams to/from a text config format
- Check that output is logical for simple designs
- Check for unknown bits in larger designs

Text Configuration

```
.tile R53C71:PLC2
arc: A1 W1_H02E0701
arc: A3 H02E0701
arc: A4 H02E0501
arc: A5 V00B0000
arc: A7 W1_H02E0501
arc: B0 S1_V02N0301
arc: S3_V06S0303 W3_H06E0303
arc: W1_H02W0401 V02S0401
word: SLICEA.K0.INIT 1100110000000000
word: SLICEA.K1.INIT 1010101000000000
enum: SLICEA.CCU2.INJECT1_0 NO
enum: SLICEA.CCU2.INJECT1_1 NO
enum: SLICEA.D0MUX 1
enum: SLICEA.D1MUX 1
enum: SLICEA.MODE CCU2
```

Timing

- Need to know how large internal delays are to determine if a design can work at a given frequency
- Like bitstream format, not enough vendor documentation
- Delays for cells (LUTs, etc) extracted from SDF files
- Interconnect delays determined using least-squares linear fit

Yosys

- Verilog RTL Synthesis Framework
- Support for multiple FPGA families (ECP5, iCE40, Xilinx, ...) and ASIC synthesis
- Uses Berkley ABC for logic optimisation
- Formal equivalence checking and assertion verification
- Plus much more!

nextpnr

- New open source multi-architecture place and route tool
- Development started early May
- Aimed primarily at real silicon (unlike VPR)
- Timing driven throughout

nextpnr

- Architectures in nextpnr implement an API rather than providing fixed data files
- Choose how you store the device database based on device size and external constraints
- Custom packer and other functions can be architecture-provided

nextpnr Arch API

- Blackbox ID types: BelId, WireId, PipId
- getBels(), getPips(), getWires(), getPipsUphill(wire): return “some kind of range” of BelId, PipId, etc
- Range must implement begin(), end()
- Iterators must implement ++, *, !=
- Could be anything from a std::vector to custom walker of a deduplicated database!

nextpnr Arch API

- Arch code stored in its own folder, different binary for each arch built
- Enables heavy compile-time optimisation and arch-specific types compared to virtual functions
- Avoids n^2 build complexity of C++ templates

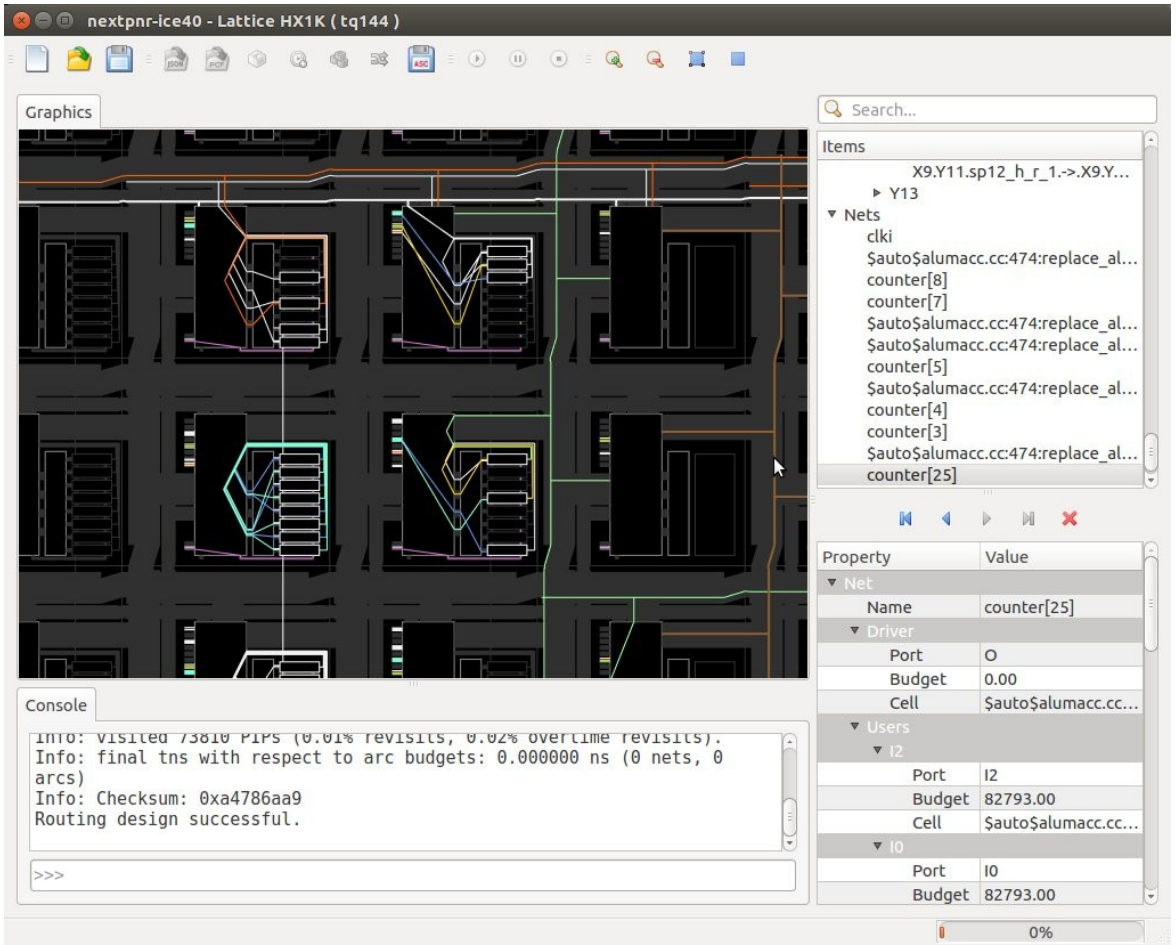
nextpnr

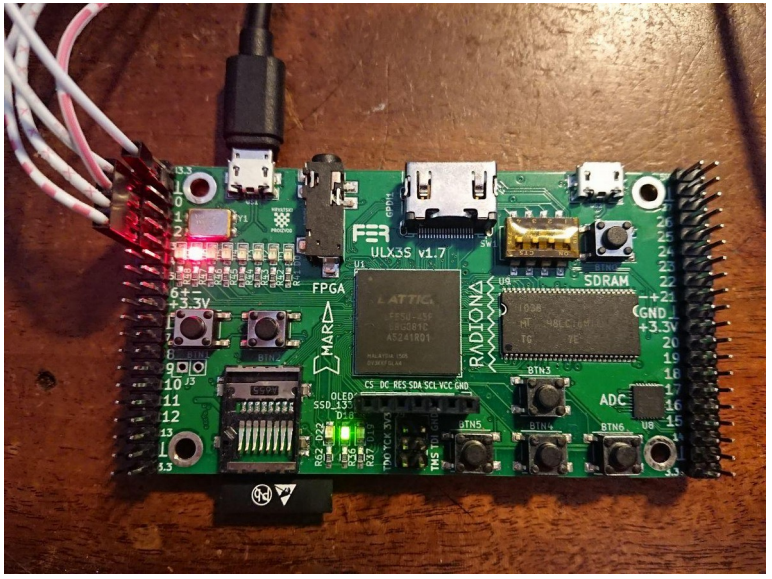
- Support for iCE40 and ECP5 FPGAs
- *Very experimental* 7-series support with Torc/XDL
- Future “generic” architecture will allow building FPGA using Python API

nextpnr

- Started over the summer with “blank canvas” PnR – SA placer and vaguely A*+ripup router
- Now working on improvements including path-based timing-driven detail placement; analytical placer; SAT-based packing/initial placer….
- Python API for extensions, constraints, custom manipulations, algorithm prototyping

nextpnr


nextpnr-ice40 - Lattice HX1K (tq144)
Graphics
Search...
Items
X9.Y11.sp12_h_r_1.->.X9.Y...
Y13
Nets
clki
$auto$alumacc.cc:474:replace_al...
counter[8]
counter[7]
$auto$alumacc.cc:474:replace_al...
$auto$alumacc.cc:474:replace_al...
counter[5]
$auto$alumacc.cc:474:replace_al...
counter[4]
counter[3]
$auto$alumacc.cc:474:replace_al...
counter[25]
Property
Value
Net
Name
counter[25]
Driver
Port
O
Budget
0.00
Cell
$auto$alumacc.cc...
Users
I2
Port
I2
Budget
82793.00
Cell
$auto$alumacc.cc...
I0
Port
I0
Budget
82793.00
Console
Info: final tns with respect to arc budgets: 0.000000 ns (0 nets, 0 arcs)
Info: Checksum: 0xa4786aa9
Routing design successful.
>>>
0%



david@archlinux: cu -l /dev/ttyUSB0 -s 57600

File Edit View Search Terminal Help

```
OpenRISC Linux -- http://openrisc.net
Built 1 zonelists in Zone order, mobility grouping off.  Total pages: 4080
Kernel command line: console=uart,mmio,0x90000000,57600
earlycon: Early serial console at MMIO 0x90000000 (options '57600')
bootconsole [uart0] enabled
PID hash table entries: 128 (order: -4, 512 bytes)
Dentry cache hash table entries: 4096 (order: 1, 16384 bytes)
Inode-cache hash table entries: 2048 (order: 0, 8192 bytes)
Sorting __ex_table...
Memory: 21880K/32768K available (2694K kernel code, 88K rwdata, 608K rodata, 7152K init, 85K bss, 108
88K reserved, 0K cma-reserved)
mem_init_done .....................................
NR_IRQS:32
clocksource: openrisc_timer: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 76450417870 ns
50.00 BogoMIPS (lpj=250000)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 2048 (order: 0, 8192 bytes)
Mountpoint-cache hash table entries: 2048 (order: 0, 8192 bytes)
devtmpfs: initialized
clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 19112604462750000 ns
NET: Registered protocol family 16
XGpio: /gpio@91000000: registered
clocksource: Switched to clocksource openrisc_timer
NET: Registered protocol family 2
TCP established hash table entries: 2048 (order: 0, 8192 bytes)
TCP bind hash table entries: 2048 (order: 0, 8192 bytes)
TCP: Hash tables configured (established 2048 bind 2048)
UDP hash table entries: 512 (order: 0, 8192 bytes)
UDP-Lite hash table entries: 512 (order: 0, 8192 bytes)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
futex hash table entries: 256 (order: -2, 3072 bytes)
Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
90000000.serial: ttyS0 at MMIO 0x90000000 (irq = 2, base_baud = 1562500) is a 16550A
console [ttyS0] enabled
console [ttyS0] enabled
bootconsole [uart0] disabled
bootconsole [uart0] disabled
NET: Registered protocol family 17
Freeing unused kernel memory: 7152K (c0352000 - c0a4e000)
init started: BusyBox v1.24.0.git (2015-02-21 02:22:51 EET)
Configuring loopback device

Please press Enter to activate this console. ifconfig: SIOCSIFADDR: No such device

/ # uname -a
Linux openrisc 4.4.0-ulx3s-66943-gd2ea1a2fbef9 #7 Fri Oct 19 15:12:07 BST 2018 openrisc GNU/Linux
/ # echo 1 > /sys/devices/platform/gpio-leds/leds/led6/brightness
/ # echo "Hello World"
Hello World
/ #
```

OSDA 2019

Workshop on Open Source Design Automation

Friday March 29, 2019

DATE 2019, Florence

https://osda.gitlab.io/

Links

Trellis: https://github.com/SymbiFlow/prjtrellis

Yosys: https://github.com/YosysHQ/yosys

nextpnr: https://github.com/YosysHQ/nextpnr

Slides: https://ds0.me/fosdem19.pdf

IRC: Freenode #yosys or ##openfpga

Email: dave@ds0.me Twitter: @fpga_dave