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

- Logic
- DSP
- RAM
- IO
- Clock Taps
- Clock Muxes
- SERDES
### ECP5 Architecture

<table>
<thead>
<tr>
<th>Logic tiles contain both logic and interconnect</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIB tiles contain interconnect for non-logic functions</td>
</tr>
<tr>
<td>MIB tiles contain non-logic functionality (EBR, DSP, IO, etc)</td>
</tr>
</tbody>
</table>

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

**Mux driving** E3_H06E0003

<table>
<thead>
<tr>
<th>Source</th>
<th>F101B8</th>
<th>F101B9</th>
<th>F102B8</th>
<th>F103B8</th>
<th>F103B9</th>
<th>F104B8</th>
<th>F104B9</th>
<th>F105B9</th>
</tr>
</thead>
<tbody>
<tr>
<td>W3_H06E0003</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>S3_V06N0003</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>V06N0003</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>W3_H06E0303</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>V06S0003</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>N3_V06S0003</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>W1_H02E0001</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>N1_V01S0000</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>V01N0001</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>W1_H02E0301</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Q0</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Q3</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>H01E0001</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>F0</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>F3</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

Frame 104, bit 9 inside tile
# Database

**Configuration word SLICEA.K0.INIT**

Default value: 16'b1111111111111111

<table>
<thead>
<tr>
<th>SLICEA.K0.INIT[0]</th>
<th>!F25B10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLICEA.K0.INIT[1]</td>
<td>!F24B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[2]</td>
<td>!F23B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[3]</td>
<td>!F22B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[4]</td>
<td>!F21B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[5]</td>
<td>!F20B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[6]</td>
<td>!F19B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[7]</td>
<td>!F18B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[8]</td>
<td>!F17B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[9]</td>
<td>!F16B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[10]</td>
<td>!F15B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[12]</td>
<td>!F13B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[13]</td>
<td>!F12B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[14]</td>
<td>!F11B10</td>
</tr>
<tr>
<td>SLICEA.K0.INIT[15]</td>
<td>!F10B10</td>
</tr>
</tbody>
</table>
# Database

## Configuration Setting EBR0.DP16KD.DATA_WIDTH_A

Default value: 18

<table>
<thead>
<tr>
<th>Value</th>
<th>F40B0</th>
<th>F47B0</th>
<th>F51B0</th>
<th>F78B0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

## Configuration Setting EBR0.DP16KD.WRITEMODE_A

Default value: NORMAL

<table>
<thead>
<tr>
<th>Value</th>
<th>F7B0</th>
<th>F101B0</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMAL</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>READBEFOREWRITE</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>WRITETHROUGH</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
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
OpenRISC Linux -- http://openrisc.net
Built 1 zonelists in Zone order, mobility grouping off. Total pages: 4080
Kernel command line: console=serial,mIODevice,8/9/0/0/0,0/0,0,767600
earlycon: Early serial console at MTD 0x90000000 (options '767600')
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  ox table...
Memory: 21800K/32760K available (2694K kernel code, 88K rdata, 608K rdso, 7152K init, 85K bss, 108
88K reserved, 9K cma-resv)
mem_init done ..............................
NR_IRQS:32
clocksource: opentrace timer: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 764504170 NS
50.00 BogoMIPS (lpj=250000)
pid max: default: 32768 minimum: 301
Mount-cache hash table entries: 2948 (order: 0, 8192 bytes)
Mountpoint-cache hash table entries: 2948 (order: 0, 8192 bytes)
devtmpfs: initnlized
clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 19112604402750000 NS
NET: Registered protocol family 16
XGpio: /gpi0@0/10000000: 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
96000000.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.248:50 at MTD 0x90000000 (irq = 2, base_baud = 1562500) is a 16550A
console [tty0] enabled
console [tty0] enabled
bootconsole [uart0] disabled
bootconsole [uart0] disabled
NET: Registered protocol family 17
Freeing unused kernel memory: 7152K (c0352000 - c0a4e000)
init started: BusyBox v1.24.6-g0.e4f1 (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-uml3s-06643-gd2eaa2f2bef9 #7 Fri Oct 19 15:12:07 BST 2018 openrisc GNU/Linux
/ # echo 1 /sys/devices/platform/gpio0-0000/leds/led0/brightness
/ # echo "Hello World"
Hello World
/
Links

Trellis: https://github.com/SymbiFlow/prjtrellis
Data: https://symbiflow.github.io/prjtrellis-db/
Yosys: https://github.com/YosysHQ/yosys
nextpnr: https://github.com/YosysHQ/nextpnr
Slides: https://ds0.me/fosdem19.pdf