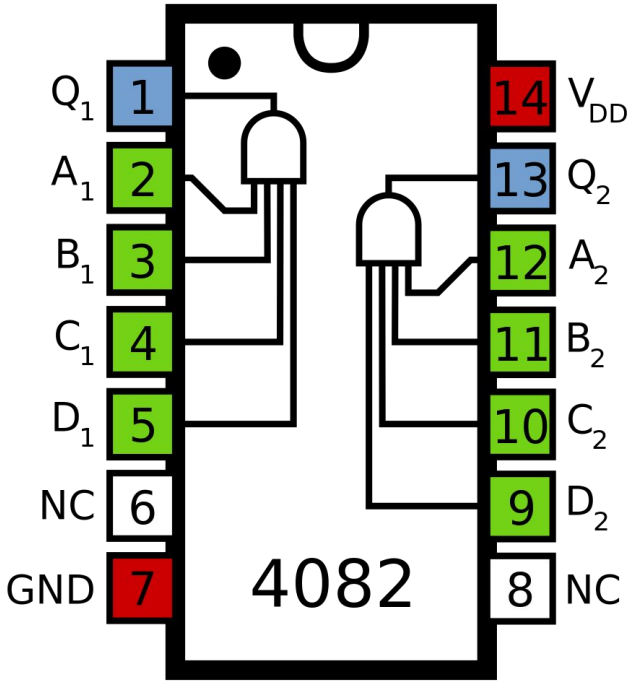
FOSS FPGA

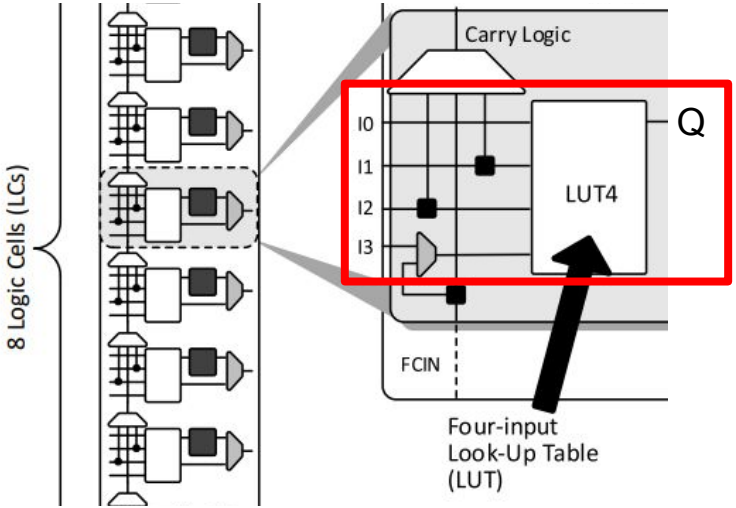
Martin Hubáček

@hubmartin
martinhubacek.cz
youtube.com/hubmartin

$$Q = I0 \& I1 \& I2 \& I3$$


Q1
1
A1
2
B1
3
C1
4
D1
5
NC
6
GND
7
14
VDD
13
Q2
12
A2
11
B2
10
C2
9
D2
8
NC
4082

Combinatorial logic



I0 - I3	Q
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	0
0111	0

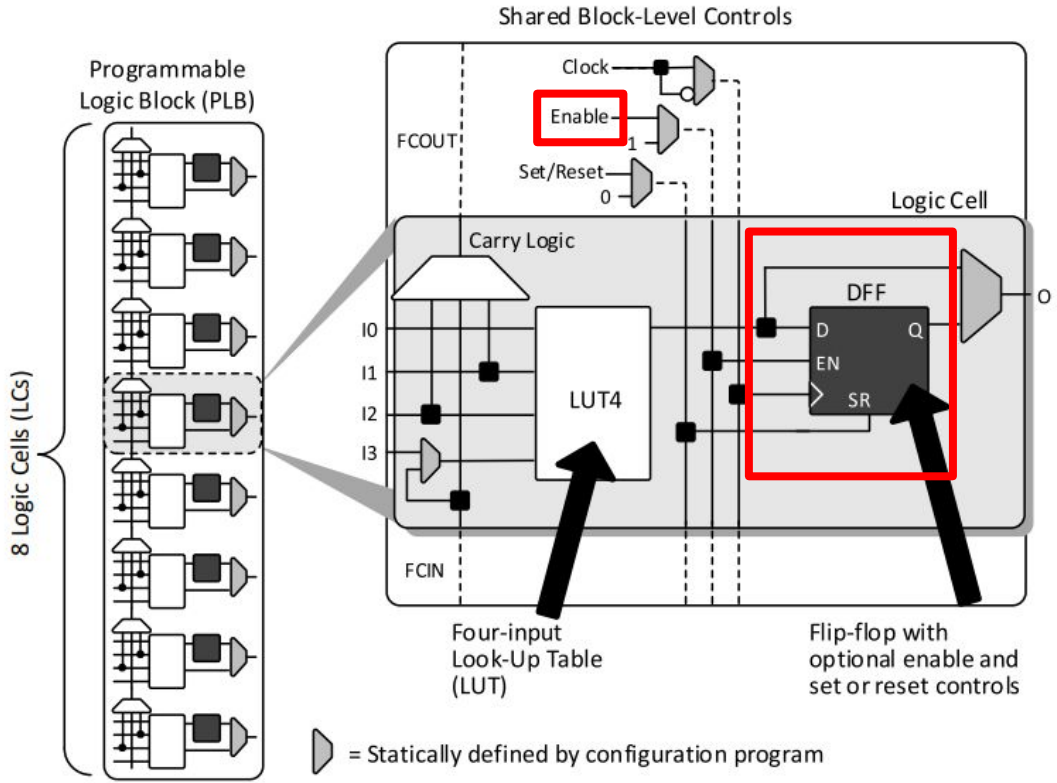
I0 - I3	Q
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	**1**

```
Q = I0 & I1 & I2 & I3

if(Q != Q_previous)
{ … }

Q_previous = Q
```

Sequential logic



Register

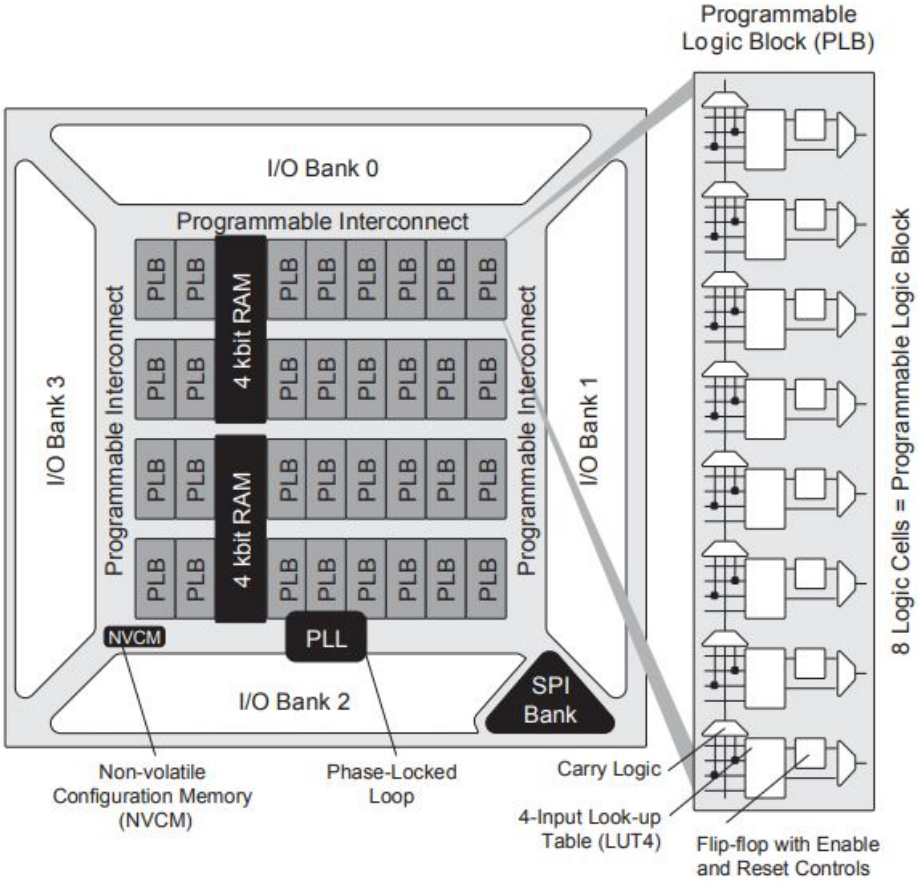
Memory

Counter

FPGA fabric

IO, BRAM, PLL, DSP

Hard-core/Soft-core peripherals or CPUs

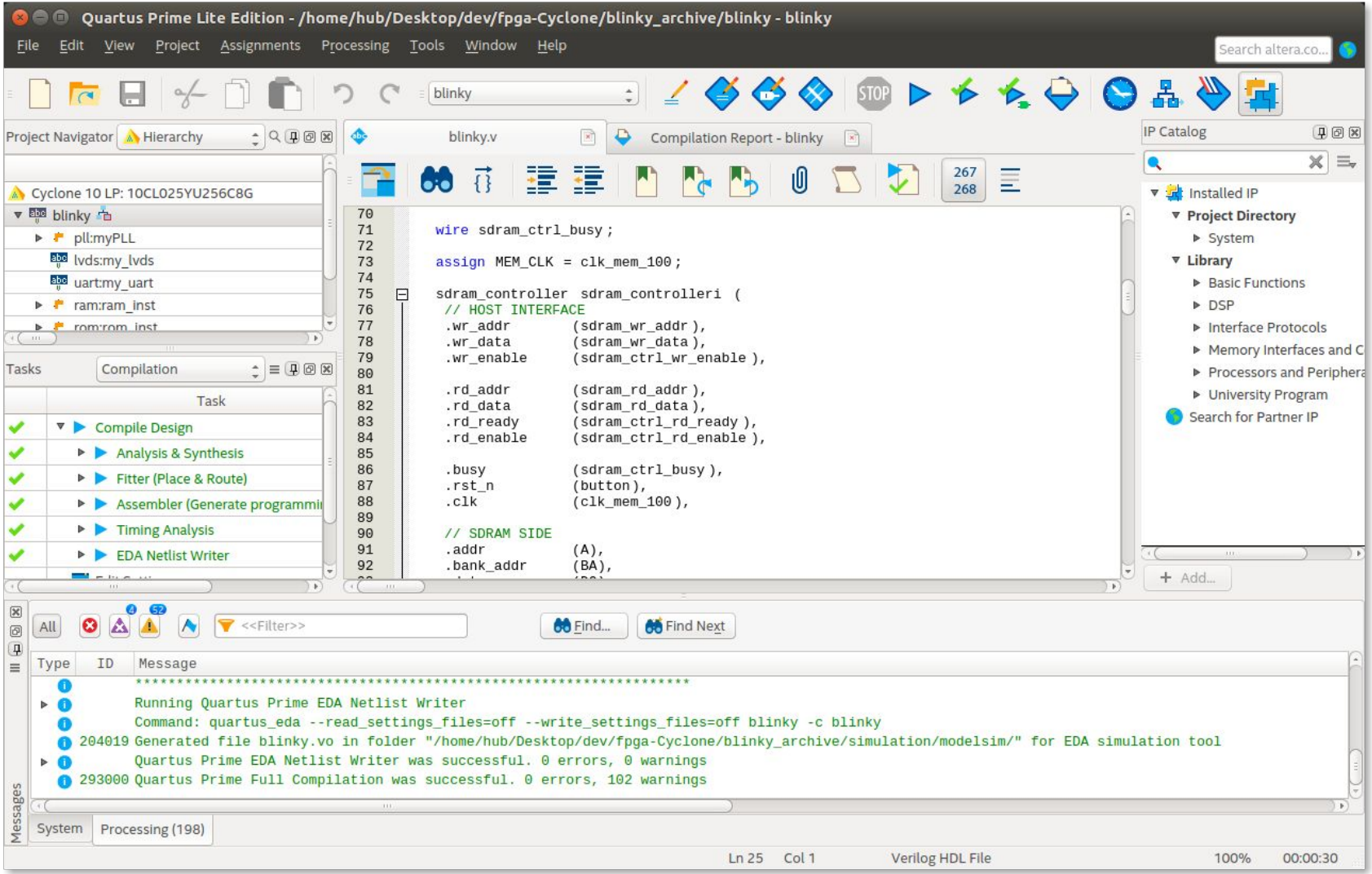


Official tools

Xilinx Vivado (ISE WebPACK)

Intel Quartus (Altera)

Lattice Diamond


Quartus Prime Lite Edition - /home/hub/Desktop/dev/fpga-Cyclone/blinky_archive/blinky - blinky
File Edit View Project Assignments Processing Tools Window Help
Search altera.co...
blinky
Project Navigator Hierarchy
Cyclone 10 LP: 10CL025YU256C8G
blinky
pll:myPLL
lvds:my_lvds
uart:my_uart
ram:ram_inst
rom:rom_inst
Tasks Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programmi
Timing Analysis
EDA Netlist Writer
blinky.v
Compilation Report - blinky
267
268
70
71 wire sdram_ctrl_busy;
72
73 assign MEM_CLK = clk_mem_100;
74
75 sdram_controller sdram_controlleri (
76 // HOST INTERFACE
77 .wr_addr (sdram_wr_addr),
78 .wr_data (sdram_wr_data),
79 .wr_enable (sdram_ctrl_wr_enable),
80
81 .rd_addr (sdram_rd_addr),
82 .rd_data (sdram_rd_data),
83 .rd_ready (sdram_ctrl_rd_ready),
84 .rd_enable (sdram_ctrl_rd_enable),
85
86 .busy (sdram_ctrl_busy),
87 .rst_n (button),
88 .clk (clk_mem_100),
89
90 // SDRAM SIDE
91 .addr (A),
92 .bank_addr (BA),
IP Catalog
Installed IP
Project Directory
System
Library
Basic Functions
DSP
Interface Protocols
Memory Interfaces and C
Processors and Peripher
University Program
Search for Partner IP
Add...
All
<<Filter>>
Find...
Find Next
Type ID Message
Running Quartus Prime EDA Netlist Writer
Command: quartus_eda --read_settings_files=off --write_settings_files=off blinky -c blinky
204019 Generated file blinky.vo in folder "/home/hub/Desktop/dev/fpga-Cyclone/blinky_archive/simulation/modelsim/" for EDA simulation tool
Quartus Prime EDA Netlist Writer was successful. 0 errors, 0 warnings
293000 Quartus Prime Full Compilation was successful. 0 errors, 102 warnings
Messages
System
Processing (198)
Ln 25 Col 1
Verilog HDL File
100%
00:00:30

Pin Planner - /home/hub/Desktop/dev/fpga-Cyclone/blinky_archive/blinky - blinky

File Edit View Processing Tools Window Help

Search altera.co...

Report

- Pin Resource Reports
  - I/O Banks
    - 1
    - 2
    - 3
    - 4
    - 5
    - 6
    - 7
    - 8
  - VREF Groups

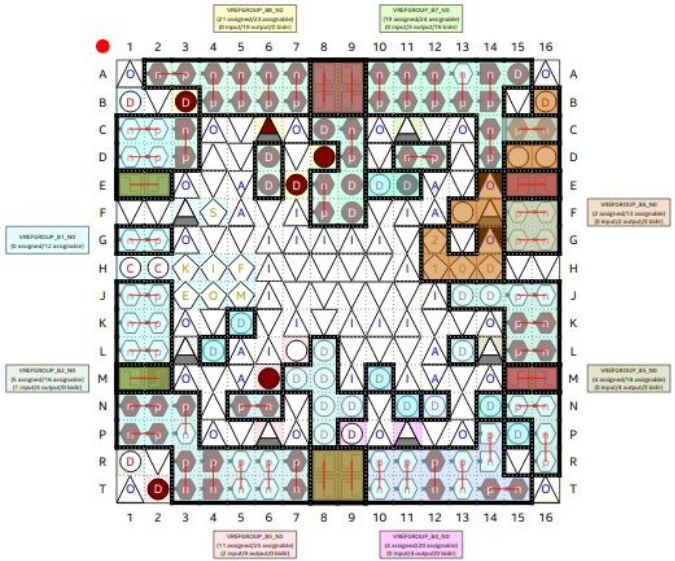
Groups | Report

Tasks

- Early Pin Planning
  - Early Pin Planning...
  - Run I/O Assignment Analysis
  - Export Pin Assignments...
- Pin Finder...
- Highlight Pins
  - I/O Banks
  - VREF Groups
  - Edges

Top View - Wire Bond

Cyclone 10 LP - 10CL025YU256C8G



Pin Legend

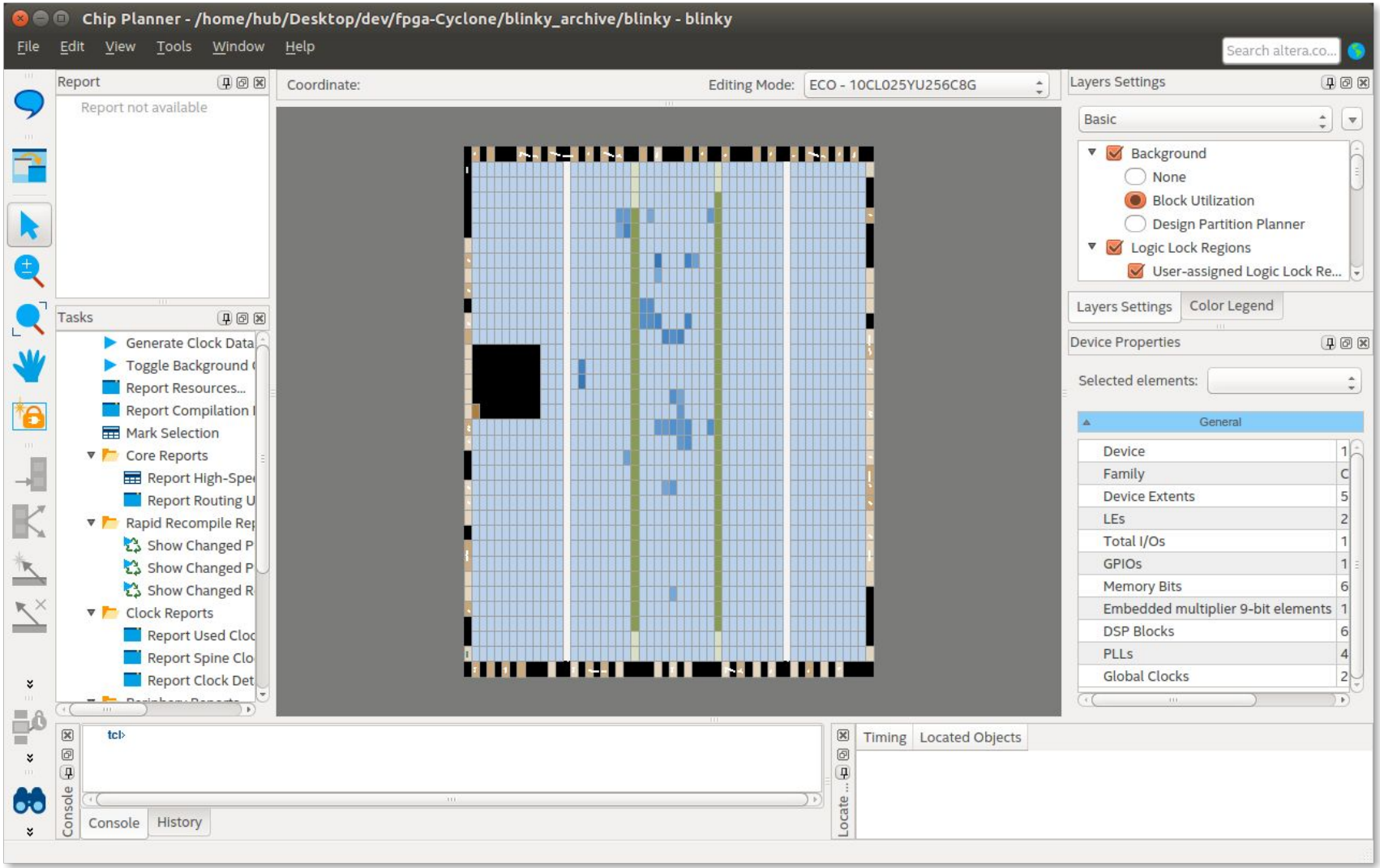
Symbol	Pin Type
	User I/O
	User assigned...
	Fitter assigne...
	Unbonded pad
	Reserved pin
C	Other configu...
n	DIFF_n
p	DIFF_p
n	DIFF_n output
p	DIFF_p output
	CLK_n
	CLK_p
D	Other dual pu...
0	MSEL0
1	MSEL1
2	MSEL2
D	CONF_DONE
E	nCE
F	nCONFIG
I	TDI
K	TCK
M	TMS
O	TDO

Named: * Edit: Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	trict Pr
led[3]	Output	PIN_R3	3	B3_N0	PIN_R3	2.5 V		8mA (default)	2 (default)		
led[2]	Output	PIN_T3	3	B3_N0	PIN_T3	2.5 V		8mA (default)	2 (default)		
led[1]	Output	PIN_T4	3	B3_N0	PIN_T4	2.5 V		8mA (default)	2 (default)		
led[0]	Output	PIN_M6	3	B3_N0	PIN_M6	2.5 V		8mA (default)	2 (default)		
lvds_clk	Output	PIN_N2	2	B2_N0	PIN_N2	LVDS_E_3R		Maximu...fault)	2 (default)	lvds_clk(n)	
lvds_even_rx[2]	Output	PIN_T14	4	B4_N0	PIN_T14	LVDS_E_3R		Maximu...fault)	2 (default)	lvds_ev...x[2](n)	
lvds_even_rx[1]	Output	PIN_R12	4	B4_N0	PIN_R12	LVDS_E_3R		Maximu...fault)	2 (default)	lvds_ev...x[1](n)	

All Pins

0% 00:00:00


Chip Planner - /home/hub/Desktop/dev/fpga-Cyclone/blinky_archive/blinky - blinky
File
Edit
View
Tools
Window
Help
Search altera.co...
Report
Report not available
Coordinate:
Editing Mode:
ECO - 10CL025YU256C8G
Layers Settings
Basic
Background
None
Block Utilization
Design Partition Planner
Logic Lock Regions
User-assigned Logic Lock Re...
Layers Settings
Color Legend
Tasks
Generate Clock Data
Toggle Background
Report Resources...
Report Compilation
Mark Selection
Core Reports
Report High-Spe
Report Routing U
Rapid Recompile Rep
Show Changed P
Show Changed P
Show Changed R
Clock Reports
Report Used Cloc
Report Spine Clo
Report Clock Det
Device Properties
Selected elements:
General
Device 1
Family C
Device Extents 5
LEs 2
Total I/Os 1
GPIOs 1
Memory Bits 6
Embedded multiplier 9-bit elements 1
DSP Blocks 6
PLLs 4
Global Clocks 2
tcl>
Console
History
Timing
Located Objects
Locate ...

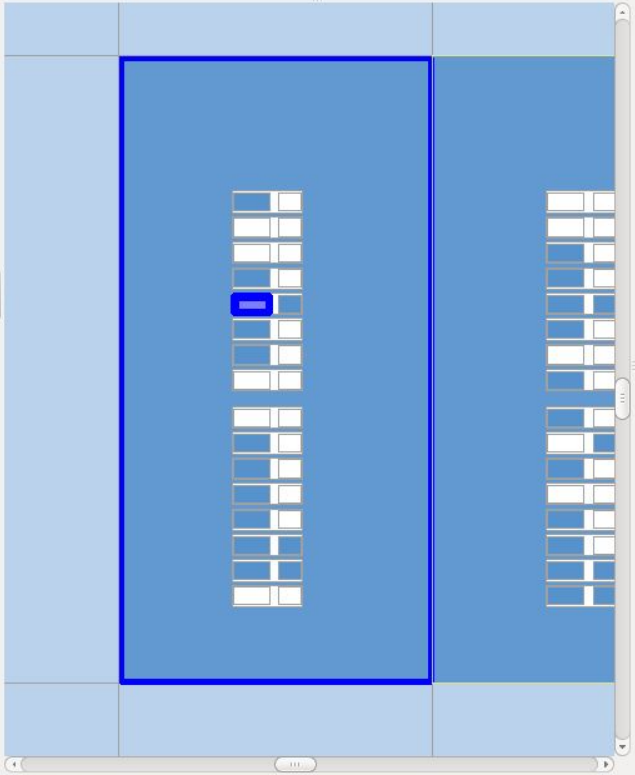
Chip Planner - /home/hub/Desktop/dev/fpga-Cyclone/blinky_archive/blinky - blinky

File Edit View Tools Window Help

Search altera.co...

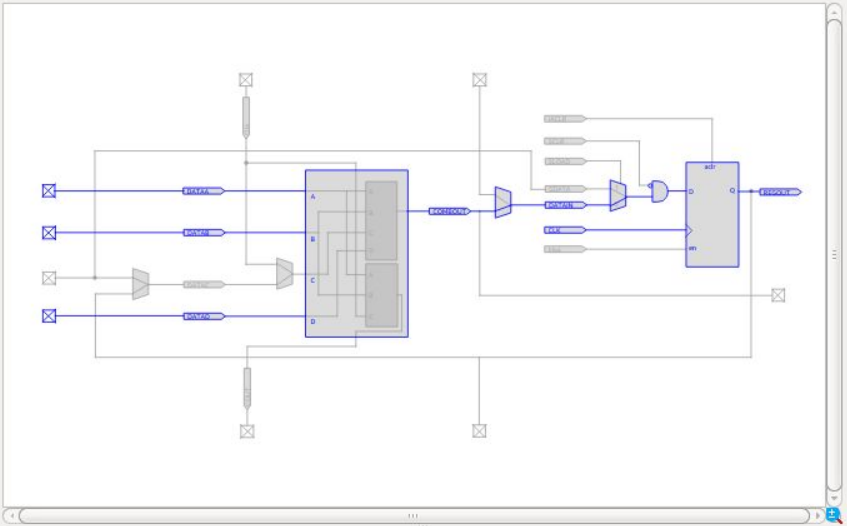
Report Tasks

Coordinate: (26, 16) Editing Mode: ECO - 10CL025YU256C8G



Node Properties

Selected elements: _~4



Properties/Modes	Values
Full Name	|blinky|rom_address[3]
Coordinate	(25 , 16)

Register Combinational

Properties Fan-in Fan-out

Layers Settings Color Legend

tcl>

Console History

Timing Located Objects

Console Locate ...

Official tools negatives

Complex many gigabyte tools

License installations

Non-effective IPs

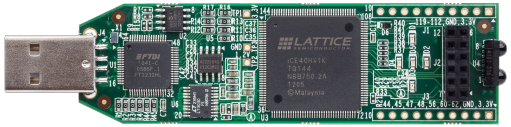
Highest FPGA lines needs licenses


Altium
NanoBoard
Altium
NanoBoard

*$$$

Generic low-cost FPGA boards

Altera MAX2 + JTAG $10

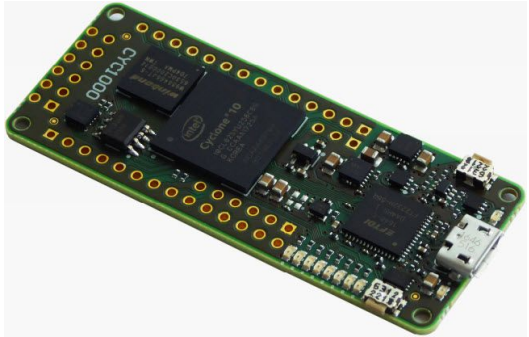


Lattice IceStick $25

Lattice MachXO2 (FLASH) $30

Altera Cyclone II + $15

Cyclone III / IV / V



Intel (Altera) CYC1000 / MAX $40

https://joelw.id.au/FPGA/CheapFPGADevelopmentBoards

How to reverse engineer an FPGA?

No details of programming internal logic and bitstream

Compare small gradual code changes (change > synthesize > compare bitstream)

Fuzzing

“de-synthethis”

Project IceStorm – iCE40 HX1K Overview

Project IceStorm aims at documenting the bitstream format of Lattice iCE40 FPGAs and providing simple to

This documentation is auto-generated by `icebox_html.py` from IceBox.
A machine-readable form of the database can be downloaded here.

The iCE40 FPGA fabric is organized into tiles. The configuration bits themself have the same

Click on a highlighted tile below to view the bitstream details for the tile. The highlighted tiles

	IO (1 17)	IO (2 17)	IO (3 17)	IO (4 17)	IO (5 17)	IO (6 17)	IO (7 17)	IO (8 17)	IO (9 17)	IO (10 17)	IO (11 17)	IO (12 17)
IO (0 16)	LOGIC (1 16)	LOGIC (2 16)	RAMT (3 16)	LOGIC (4 16)	LOGIC (5 16)	LOGIC (6 16)	LOGIC (7 16)	LOGIC (8 16)	LOGIC (9 16)	RAMT (10 16)	LOGIC (11 16)	LOGIC (12 16)
IO (0 15)	LOGIC (1 15)	LOGIC (2 15)	RAMB (3 15)	LOGIC (4 15)	LOGIC (5 15)	LOGIC (6 15)	LOGIC (7 15)	LOGIC (8 15)	LOGIC (9 15)	RAMB (10 15)	LOGIC (11 15)	LOGIC (12 15)
IO (0 14)	LOGIC (1 14)	LOGIC (2 14)	RAMT (3 14)	LOGIC (4 14)	LOGIC (5 14)	LOGIC (6 14)	LOGIC (7 14)	LOGIC (8 14)	LOGIC (9 14)	RAMT (10 14)	LOGIC (11 14)	LOGIC (12 14)
IO (0 13)	LOGIC (1 13)	LOGIC (2 13)	RAMB (3 13)	LOGIC (4 13)	LOGIC (5 13)	LOGIC (6 13)	LOGIC (7 13)	LOGIC (8 13)	LOGIC (9 13)	RAMB (10 13)	LOGIC (11 13)	LOGIC (12 13)
IO (0 12)	LOGIC (1 12)	LOGIC (2 12)	RAMT (3 12)	LOGIC (4 12)	LOGIC (5 12)	LOGIC (6 12)	LOGIC (7 12)	LOGIC (8 12)	LOGIC (9 12)	RAMT (10 12)	LOGIC (11 12)	LOGIC (12 12)
IO (0 11)	LOGIC (1 11)	LOGIC (2 11)	RAMB (3 11)	LOGIC (4 11)	LOGIC (5 11)	LOGIC (6 11)	LOGIC (7 11)	LOGIC (8 11)	LOGIC (9 11)	RAMB (10 11)	LOGIC (11 11)	LOGIC (12 11)
IO (0 10)	LOGIC (1 10)	LOGIC (2 10)	RAMT (3 10)	LOGIC (4 10)	LOGIC (5 10)	LOGIC (6 10)	LOGIC (7 10)	LOGIC (8 10)	LOGIC (9 10)	RAMT (10 10)	LOGIC (11 10)	LOGIC (12 10)
IO (0 9)	LOGIC (1 9)	LOGIC (2 9)	RAMB (3 9)	LOGIC (4 9)	LOGIC (5 9)	LOGIC (6 9)	LOGIC (7 9)	LOGIC (8 9)	LOGIC (9 9)	RAMB (10 9)	LOGIC (11 9)	LOGIC (12 9)
IO (0 8)	LOGIC (1 8)	LOGIC (2 8)	RAMT (3 8)	LOGIC (4 8)	LOGIC (5 8)	LOGIC (6 8)	LOGIC (7 8)	LOGIC (8 8)	LOGIC (9 8)	RAMT (10 8)	LOGIC (11 8)	LOGIC (12 8)
IO (0 7)	LOGIC (1 7)	LOGIC (2 7)	RAMB (3 7)	LOGIC (4 7)	LOGIC (5 7)	LOGIC (6 7)	LOGIC (7 7)	LOGIC (8 7)	LOGIC (9 7)	RAMB (10 7)	LOGIC (11 7)	LOGIC (12 7)
IO (0 6)	LOGIC (1 6)	LOGIC (2 6)	RAMT (3 6)	LOGIC (4 6)	LOGIC (5 6)	LOGIC (6 6)	LOGIC (7 6)	LOGIC (8 6)	LOGIC (9 6)	RAMT (10 6)	LOGIC (11 6)	LOGIC (12 6)
IO (0 5)	LOGIC (1 5)	LOGIC (2 5)	RAMB (3 5)	LOGIC (4 5)	LOGIC (5 5)	LOGIC (6 5)	LOGIC (7 5)	LOGIC (8 5)	LOGIC (9 5)	RAMB (10 5)	LOGIC (11 5)	LOGIC (12 5)
IO (0 4)	LOGIC (1 4)	LOGIC (2 4)	RAMT (3 4)	LOGIC (4 4)	LOGIC (5 4)	LOGIC (6 4)	LOGIC (7 4)	LOGIC (8 4)	LOGIC (9 4)	RAMT (10 4)	LOGIC (11 4)	LOGIC (12 4)
IO (0 3)	LOGIC (1 3)	LOGIC (2 3)	RAMB (3 3)	LOGIC (4 3)	LOGIC (5 3)	LOGIC (6 3)	LOGIC (7 3)	LOGIC (8 3)	LOGIC (9 3)	RAMB (10 3)	LOGIC (11 3)	LOGIC (12 3)
IO (0 2)	LOGIC (1 2)	LOGIC (2 2)	RAMT (3 2)	LOGIC (4 2)	LOGIC (5 2)	LOGIC (6 2)	LOGIC (7 2)	LOGIC (8 2)	LOGIC (9 2)	RAMT (10 2)	LOGIC (11 2)	LOGIC (12 2)
IO (0 1)	LOGIC (1 1)	LOGIC (2 1)	RAMB (3 1)	LOGIC (4 1)	LOGIC (5 1)	LOGIC (6 1)	LOGIC (7 1)	LOGIC (8 1)	LOGIC (9 1)	RAMB (10 1)	LOGIC (11 1)	LOGIC (12 1)
	IO (1 0)	IO (2 0)	IO (3 0)	IO (4 0)	IO (5 0)	IO (6 0)	IO (7 0)	IO (8 0)	IO (9 0)	IO (10 0)	IO (11 0)	IO (12 0)

Project IceStorm – iCE40 HX1K LOGIC Tile (7 9)

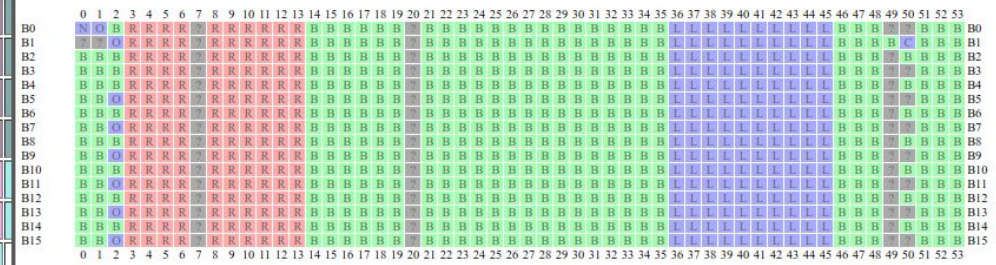
Project IceStorm aims at documenting the bitstream format of Lattice iCE40 FPGAs and providing simple tools for analyzing and creating bitstream files. This is work in progress.

This page describes the LOGIC Tile (7 9), what nets and configuration bits it has and how it is connected to its neighbourhood.

LOGIC Tile (5 11)	LOGIC Tile (6 11)	LOGIC Tile (7 11)	LOGIC Tile (8 11)	LOGIC Tile (9 11)
LOGIC Tile (5 10)	LOGIC Tile (6 10)	LOGIC Tile (7 10)	LOGIC Tile (8 10)	LOGIC Tile (9 10)
LOGIC Tile (5 9)	**LOGIC Tile (6 9)**	**LOGIC Tile (7 9)**	LOGIC Tile (8 9)	LOGIC Tile (9 9)
LOGIC Tile (5 8)	**LOGIC Tile (6 8)**	**LOGIC Tile (7 8)**	LOGIC Tile (8 8)	LOGIC Tile (9 8)
LOGIC Tile (5 7)	LOGIC Tile (6 7)	LOGIC Tile (7 7)	LOGIC Tile (8 7)	LOGIC Tile (9 7)

Configuration Bitmap

A LOGIC Tile has 864 config bits in 16 groups of 54 bits each:
`B0[53:0], B1[53:0], B2[53:0], B3[53:0], B4[53:0], B5[53:0], B6[53:0], B7[53:0],`
`B8[53:0], B9[53:0], B10[53:0], B11[53:0], B12[53:0], B13[53:0], B14[53:0], B15[53:0]`



https://knielsen.github.io/ice40_viewer/ice40_viewer.html

http://www.clifford.at/icestorm/bitdocs-1k/

FOSS FPGA chips/boards

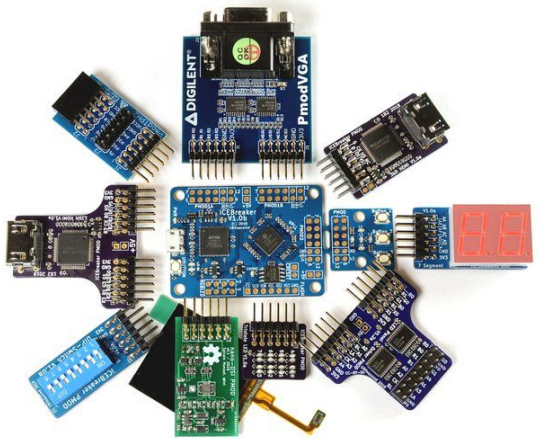
IceStorm - iCE40 (ICE40, 8k, 128kb, PLL)

- TinyFPGA BX $38
- iCEBreaker
- Glasgow
- Olimex iCE40HX8K-EVB
- ICOboard (Rpi HAT) OpenTechLab YouTube
- Alhambra (Arduino footprint)

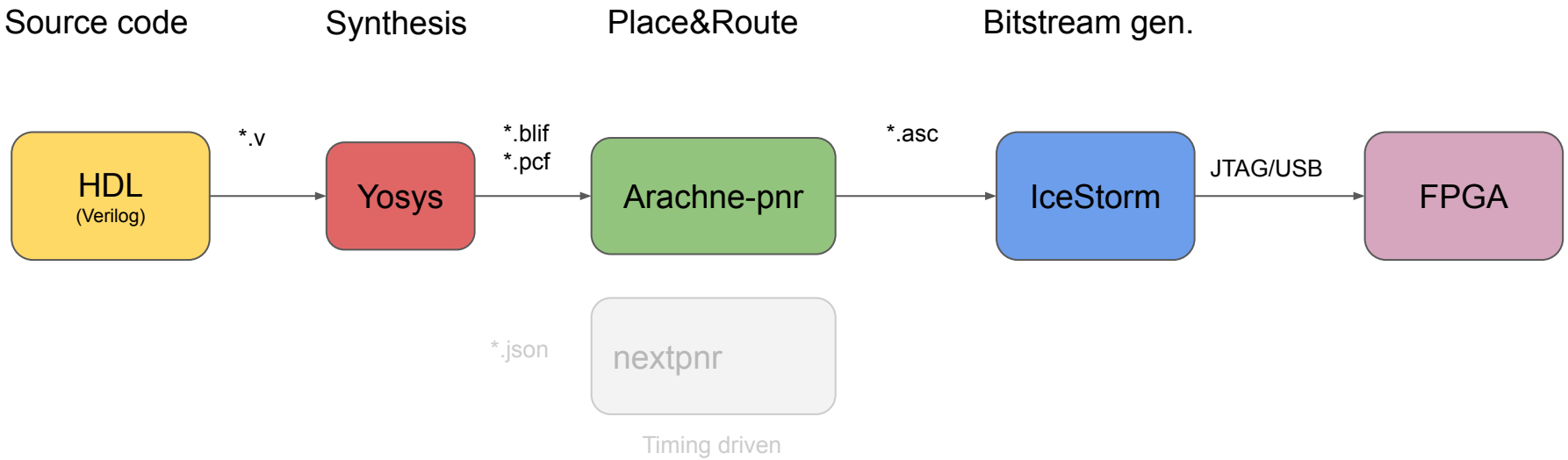
Trellis - ECP5

(85k, 3.7Mb BRAM, 156 18x18 DSPs, 5Gbps SERDES)

- TinyFPGA EX
- ULX3S

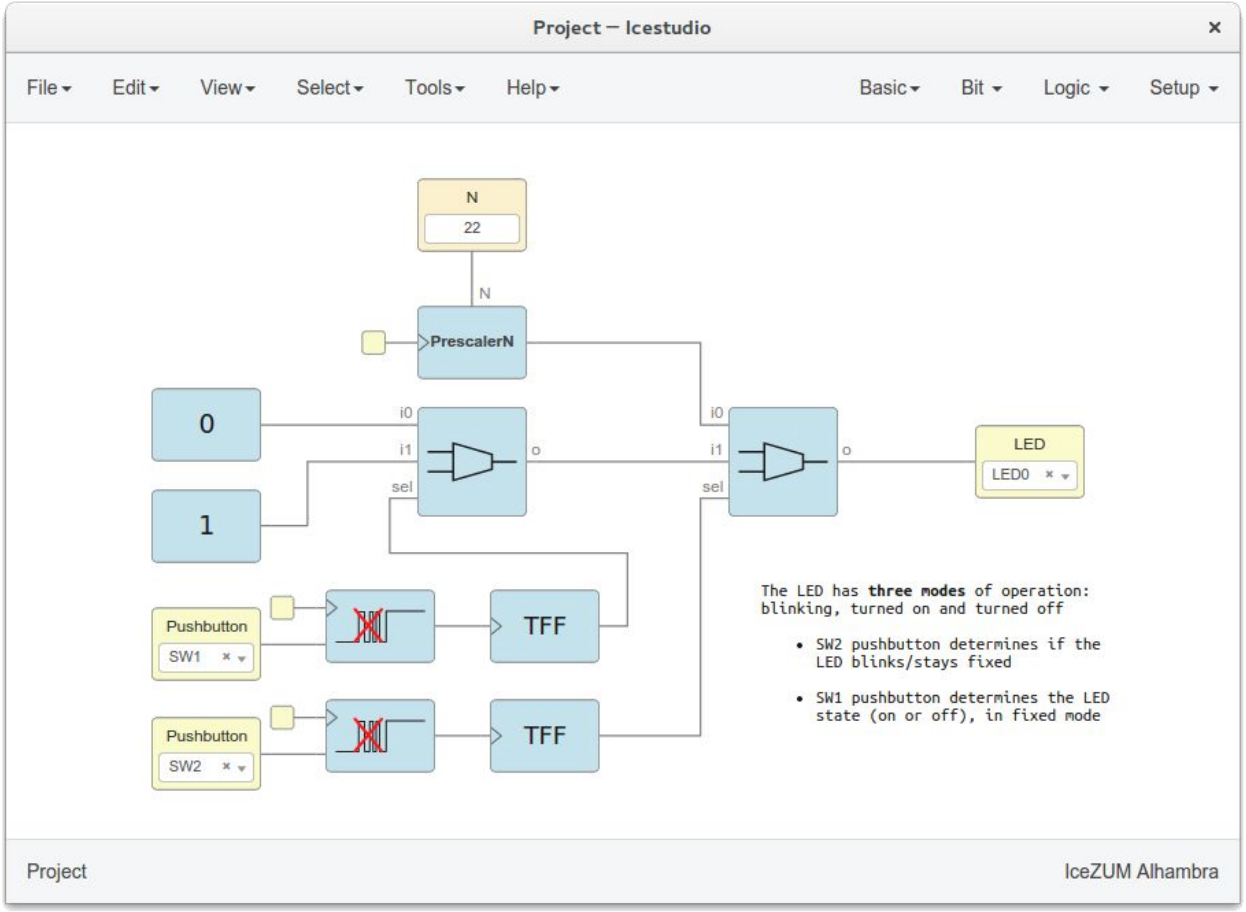


X-Ray - Xilinx 7-series



Berkeley Logic Interchange Format

Icestudio



FPGA wars

Demo 01 - Icestudio

APIO

Apio is a multiplatform toolbox, with static pre-built packages, project configuration tools and easy command interface to verify, synthesize, simulate and upload your verilog designs.

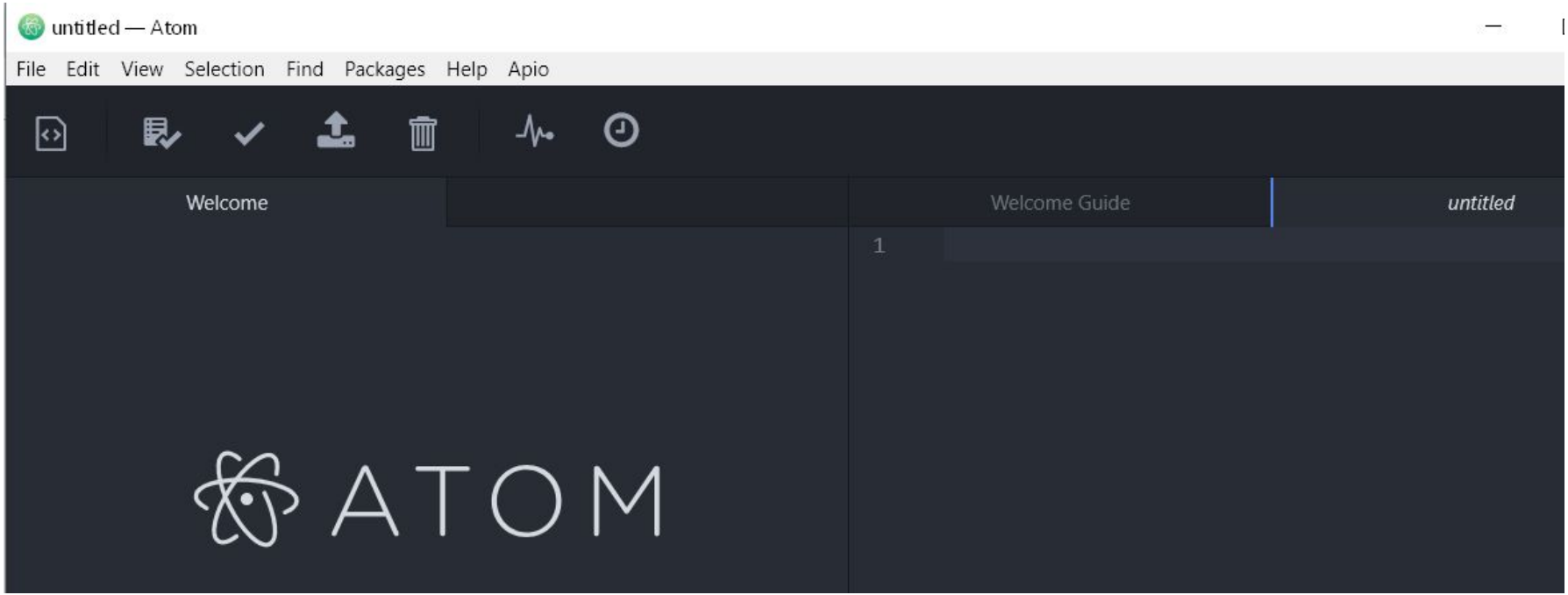
```
pip install apio==0.4.0b5 tinyprog

apio install system scons icestorm iverilog

apio drivers --serial-enable


apio [build|sim|upload]
```

ATOM IDE

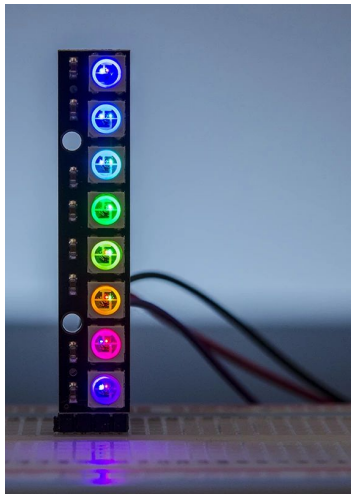

untitled — Atom
File Edit View Selection Find Packages Help Apio
Welcome
Welcome Guide
untitled
1
ATOM

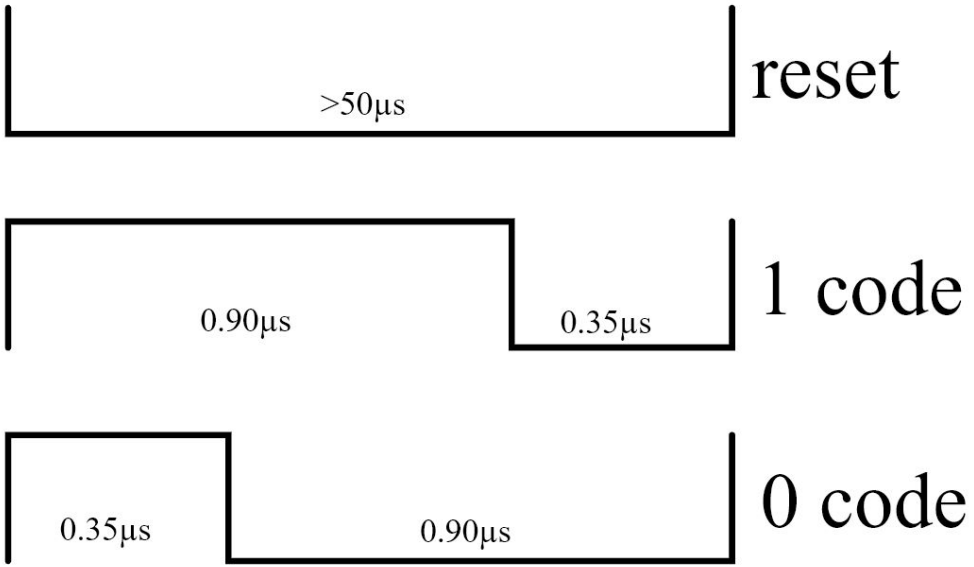
Demo 02 - APIO

WS2812B Module

Digital RGB(W) LED strips

Single wire, precise timing





Demo 03 - WS2812B

Simulation

Testbench (*_tb.v)

Test each module

Icarus Verilog - simulation and synthesis tool (iverilog)

Gtkview - Waveform view

Verilator - Compile Verilog to C++

EDA Playground https://www.edaplayground.com/x/2bRW

Demo 04 - Simulation

PicoSOC

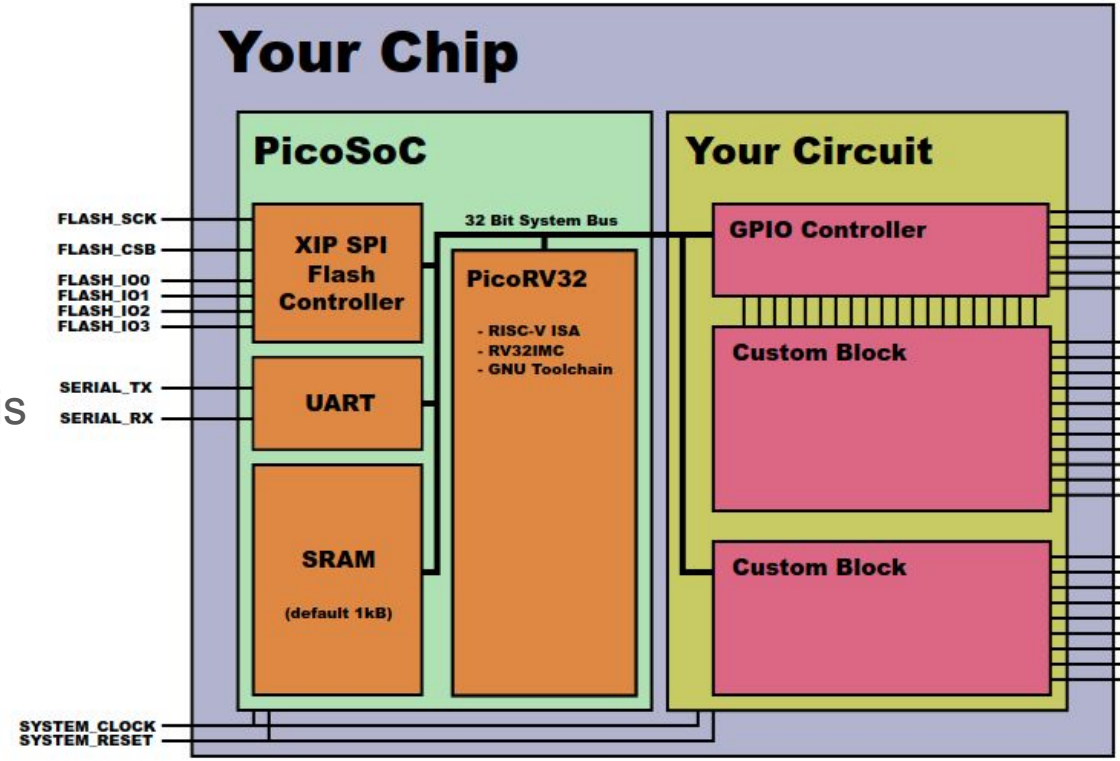
Soft-core RISC-V

When is soft-core useful

Compilation is faster than synthetis

Riscv-none-embed-gcc toolchain

Custom peripherals




```
$ xpm install --global @xpack-dev-tools/riscv-none-embed-gcc@latest
```

Two PicoSOCs place&route

https://twitter.com/q3k/status/1024623710165237760

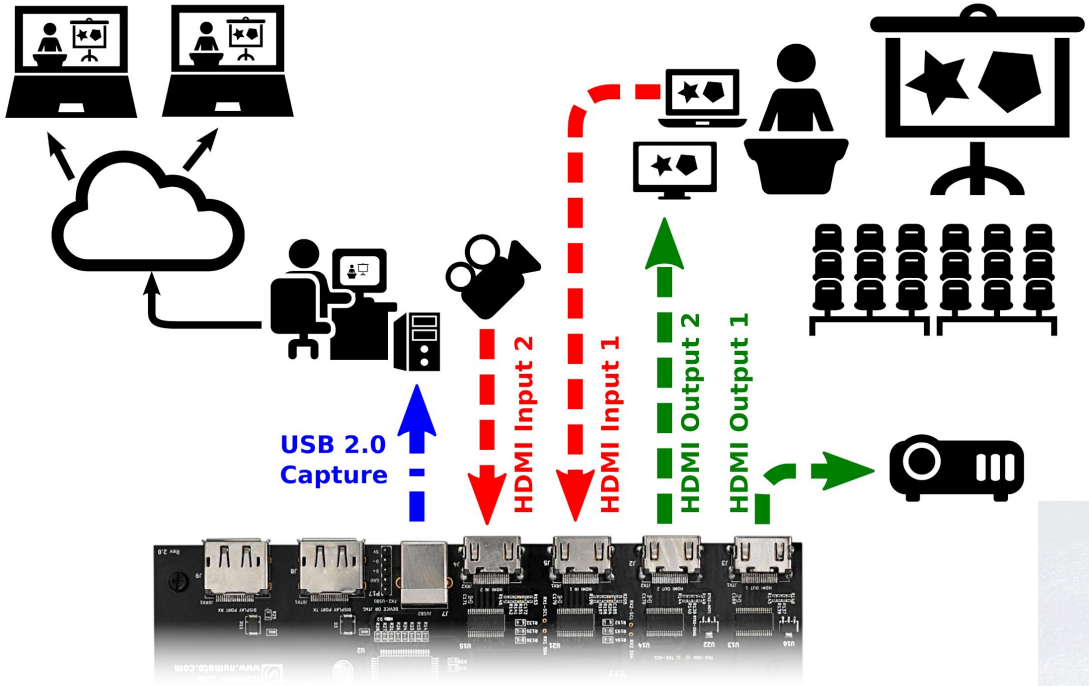
Demo 05 - RISC-V

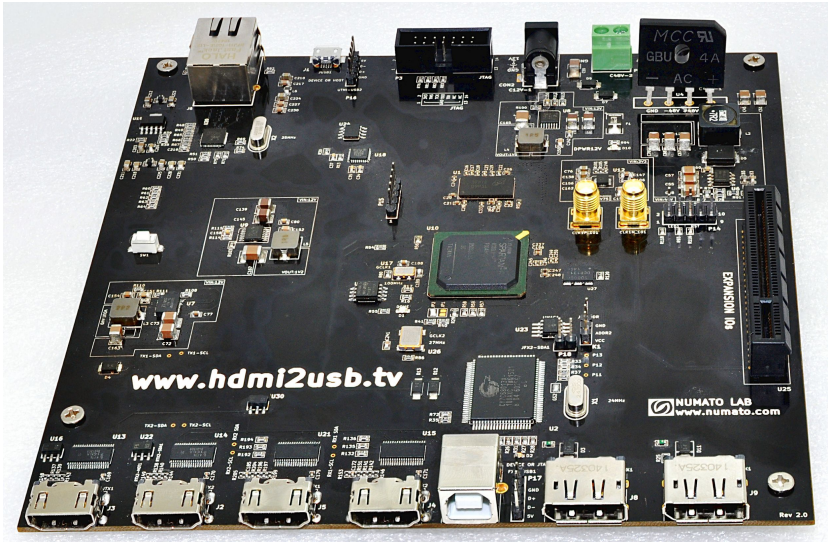
Migen, Litex

Migen, nMigen - python nástroj pro generování komplexního hardware

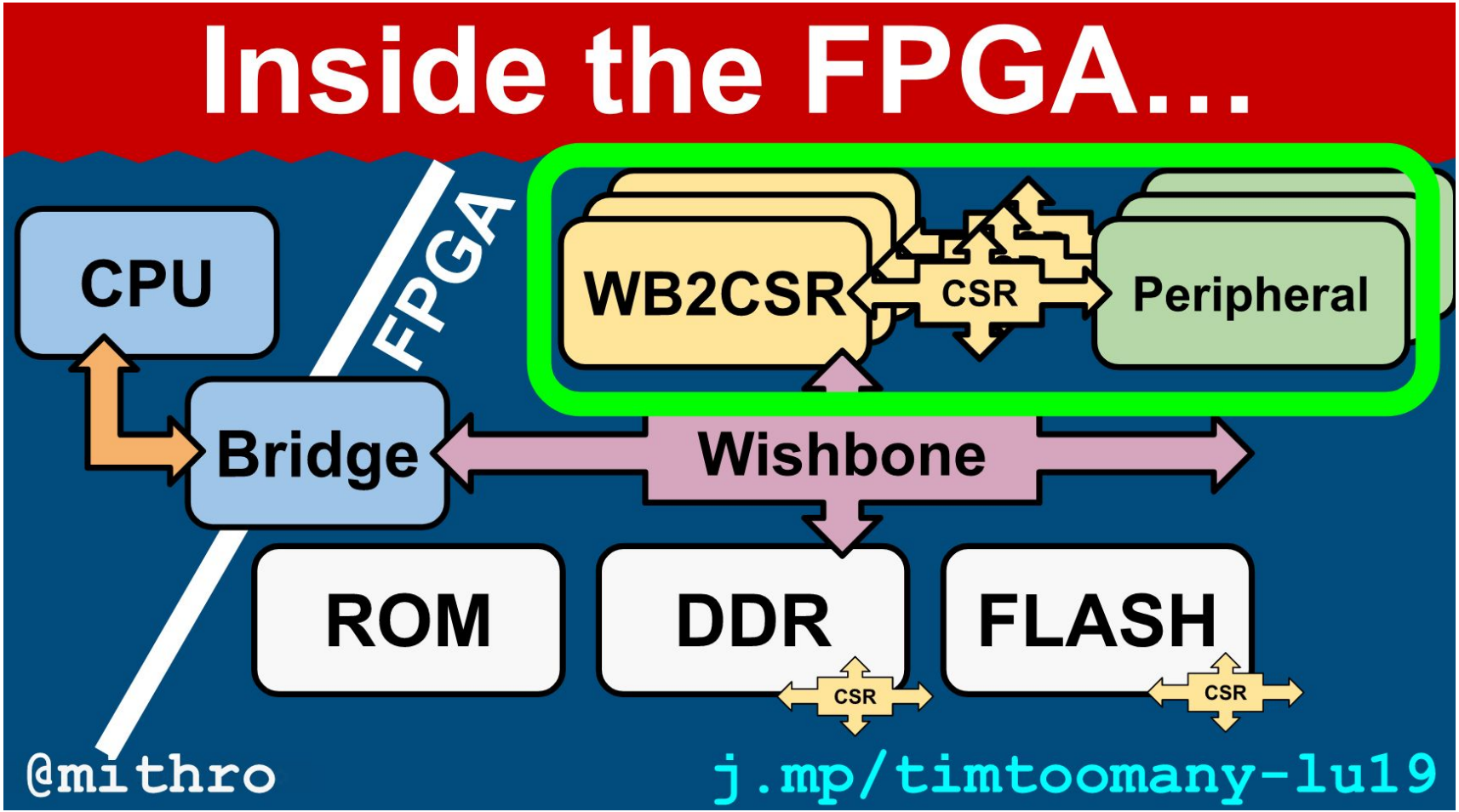
Nekompiluje python, pomáhá tvořit HDL kód

LiteX - LiteX is a MiSoC-based SoC builder using Migen as Python DSL that can be used to create SoCs and full FPGA designs.


USB 2.0
Capture
HDMI Input 2
HDMI Input 1
HDMI Output 2
HDMI Output 1


www.hdmi2usb.tv
NUMATO LAB
www.numato.com

HDMI2USB.tv



Tim has to many projects - LatchUp Edition https://www.youtube.com/watch?v=v7WrTmexod0

Future

SERDES 5G

Partial reconfiguration

Glasgow

...

Registrace zdarma
$7000 Prize money
22.-23.listopad
Brněnské výstaviště E

https://www.hackathons.cz/

HARDWARIO

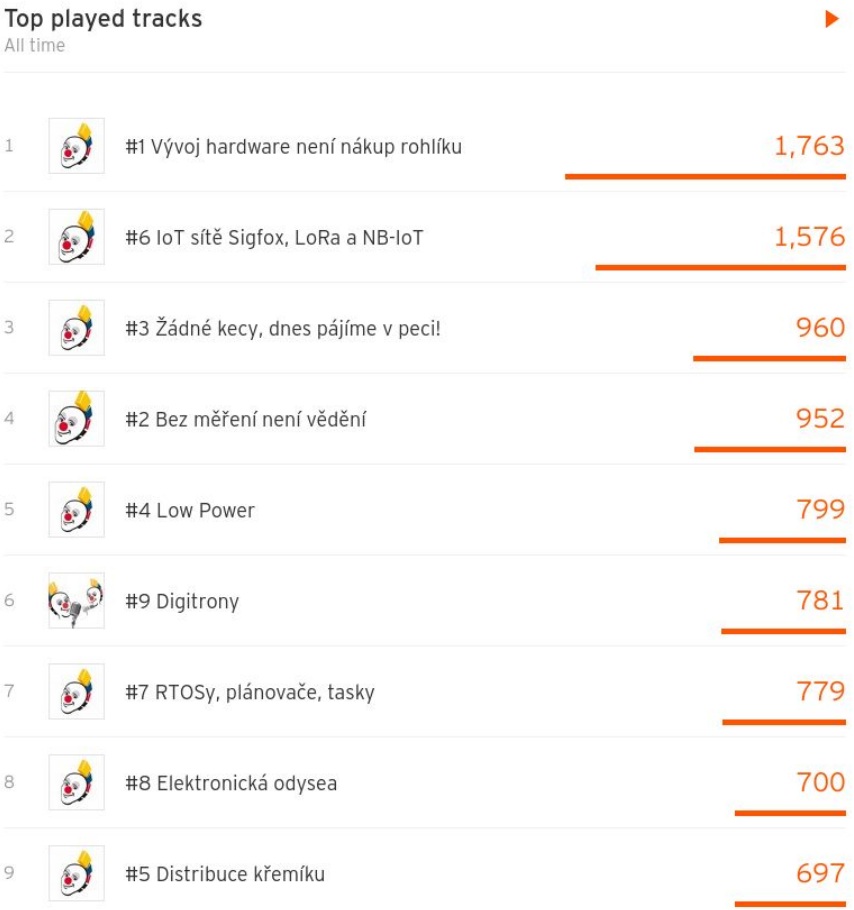
FAB BRNO

CZ.NIC

HWDEV Podcast

Hardware Development Podcast se zaměřuje na zajímavá témata z oblasti vývoje, výroby a oživování elektroniky.

https://soundcloud.com/hwdevpodcast



HWDEV Podcast

Hardware Development Podcast se zaměřuje na zajímavá témata z oblasti vývoje, výroby a oživování elektroniky.

https://soundcloud.com/hwdevpodcast

@hubmartin

martinhubacek.cz

www.youtube.com/hubmartin