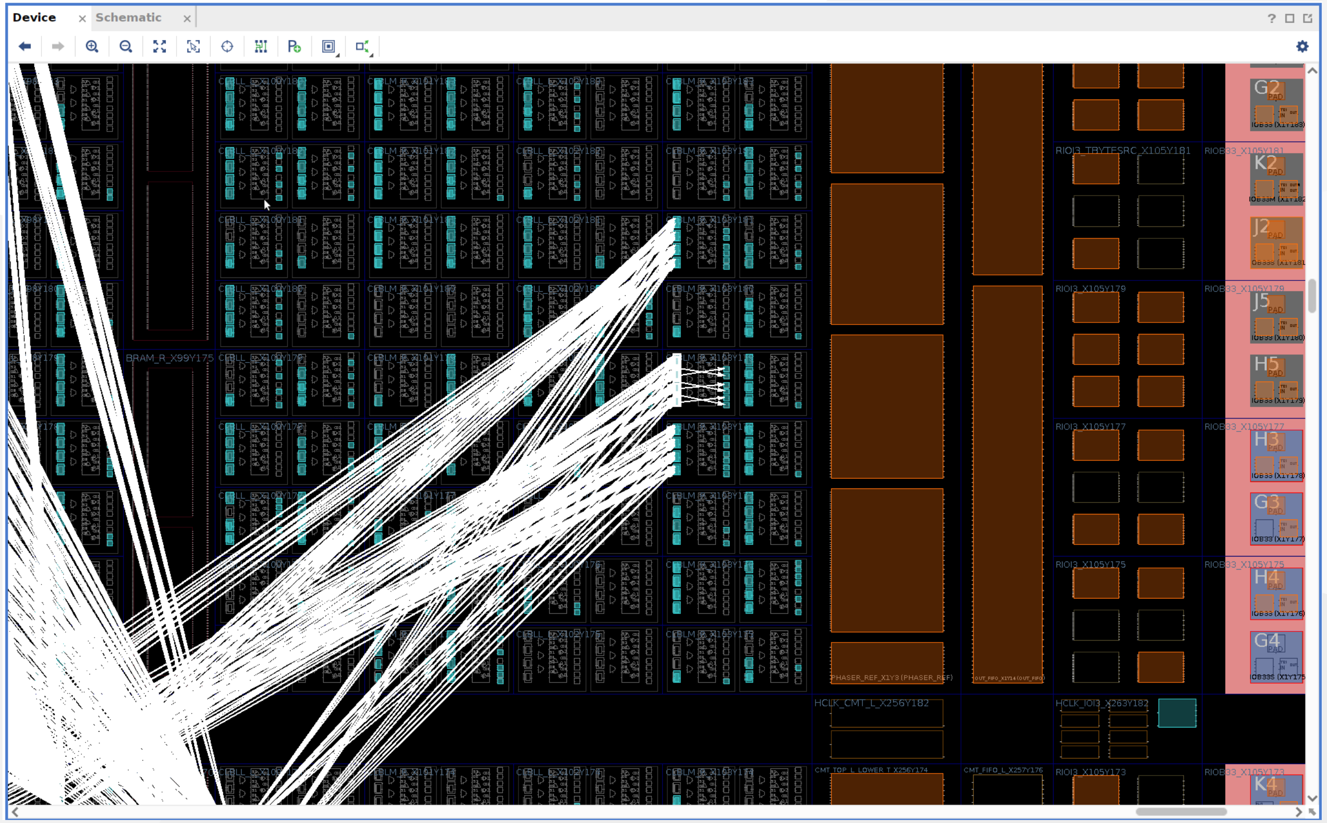
RISC-V Lab

Ex2: FPGA Protoyping



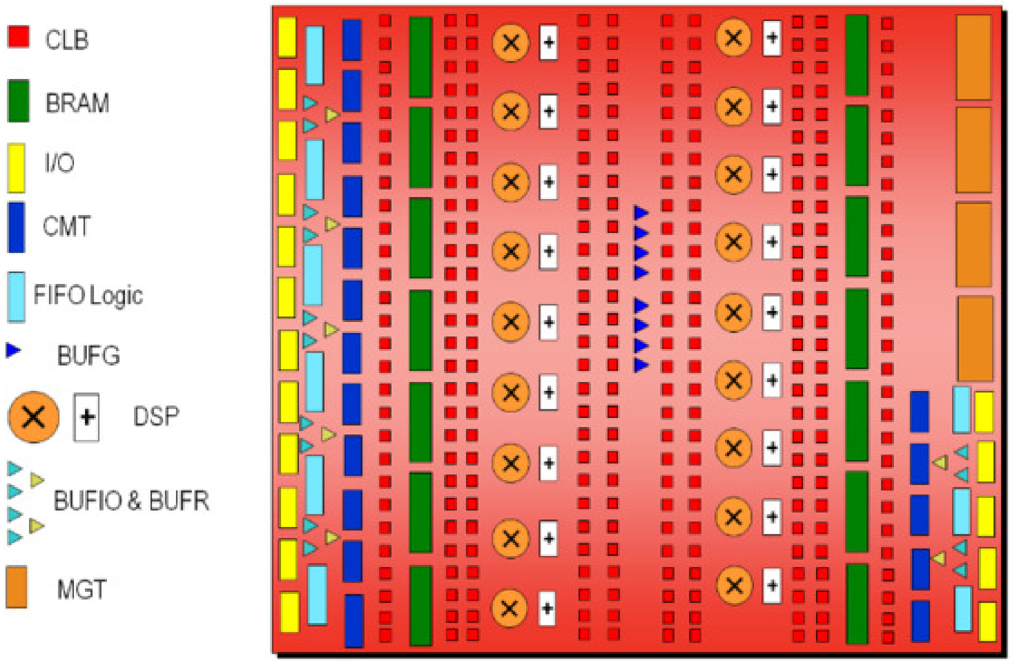
Content

1. Xilinx Series 7 (condensed)
2. FPGA design flow
3. Netlist simulation

Xilinx Series 7 (condensed)

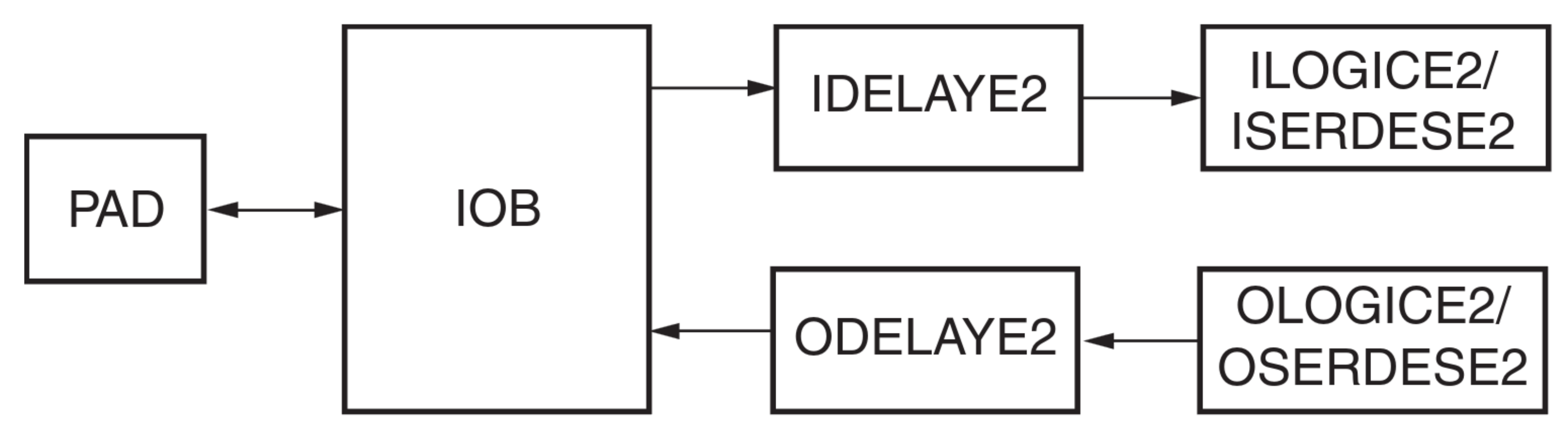
Overview

- common architecture for all 7 series
- columns with
  - IO
  - slice / CLB
  - Hard macros: DSP, BRAM, ...



Artix-7 Architecture Overview

IO Tile



- PAD: metal window
- IOB (IO Buffer): physical driver / receiver (logic <-> world)
- IDELAY2 / ODLEAY2 : programmable delays
  e.g. for data line delay matching
- I/OLOGIC I/OSERDES: fast shift registers, ...

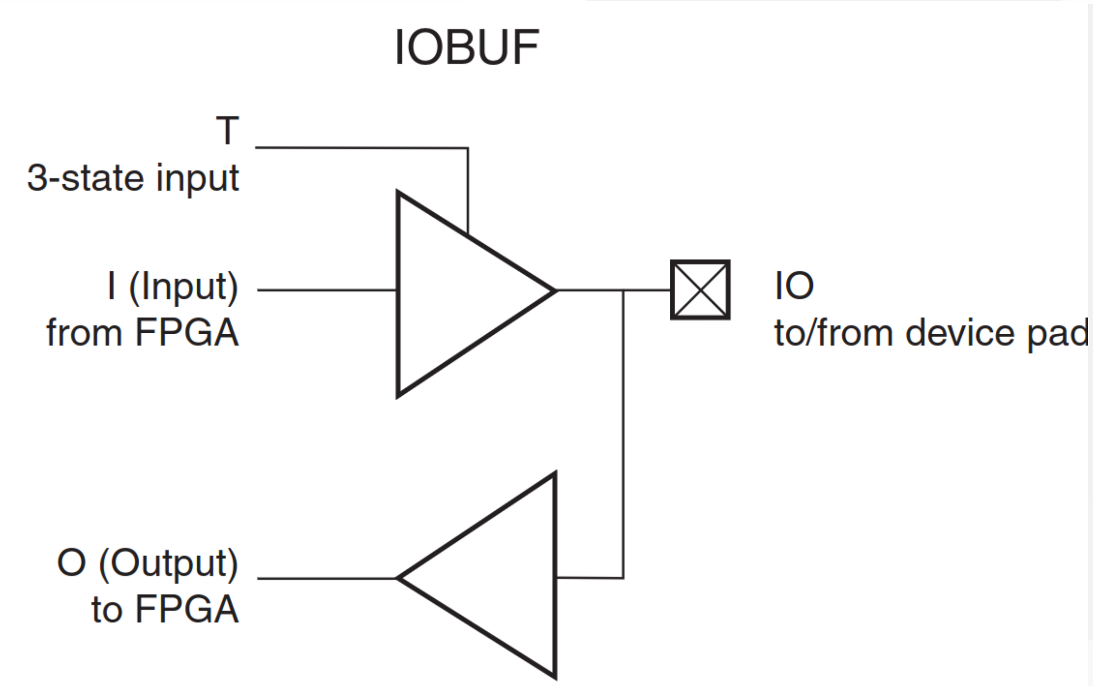
IO Tile: IO Buffers

- IOBUF
  - on PCB set to 3.3V (!)

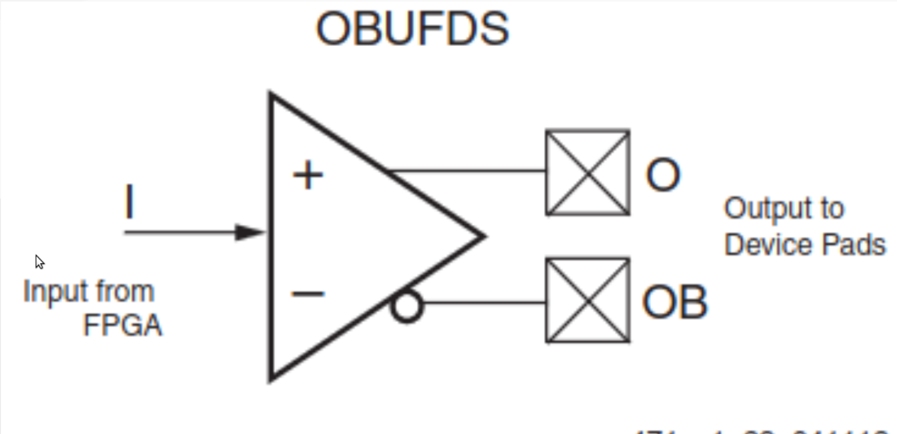
```
T  I  O      IO
0  0  IO     0
0  1  IO     1
1  *  Z/IO   Z/IO
Note: rvlab wrapper inverts T !
```

- IBUFDS / OBUFDS (HDMI)

from/details: "7 Series FPGAs SelectIO Resources User Guide (UG471)"



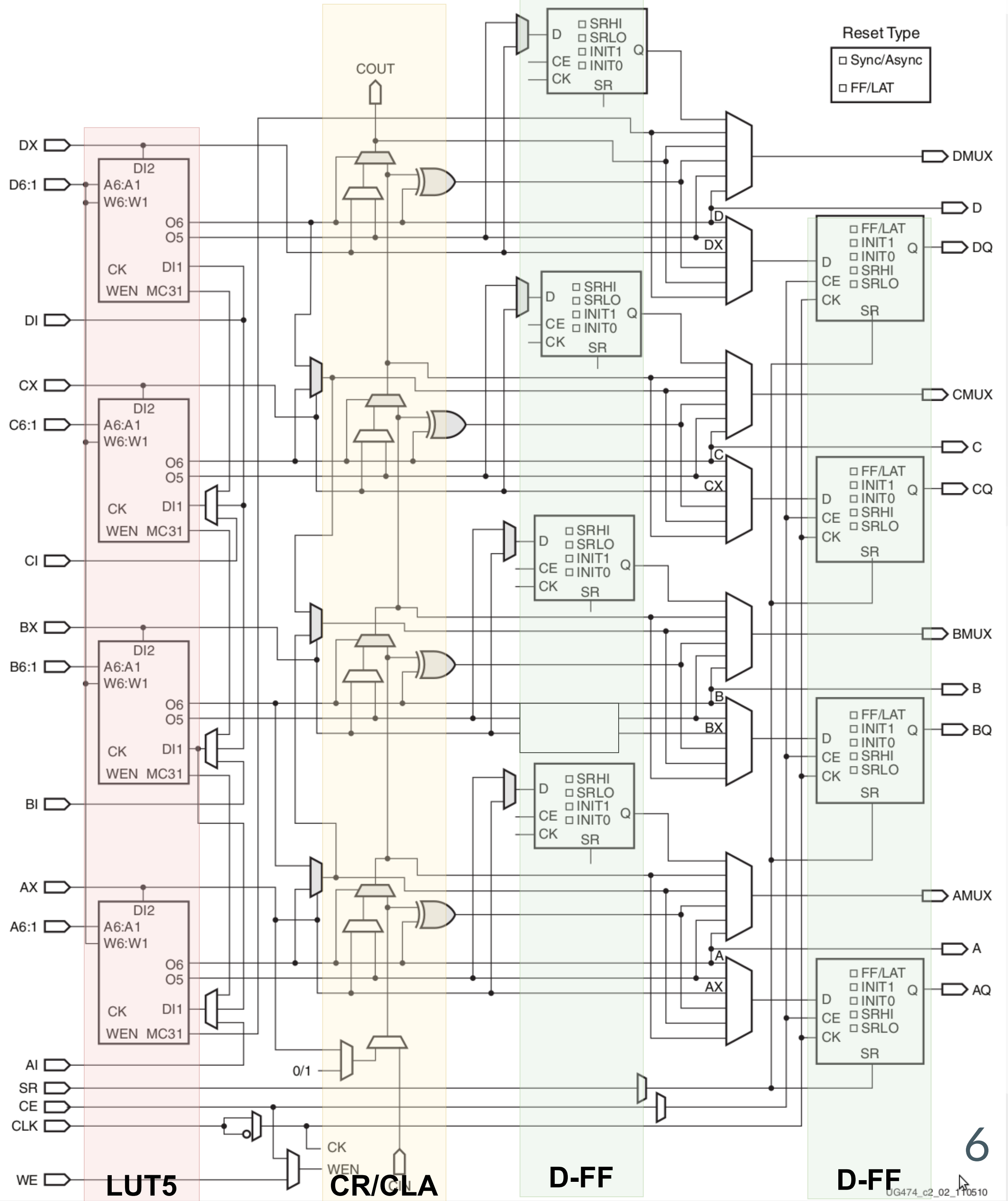
ug471_c1_20_041112



ug471_c1_22_041112

Slice

- basic block for all logic!
- 4x 6 input or 8x 5 input LUT
- 8x D-FF
- add/sub: CR / 4 bit CLA
- (output) muxing
- alternatively: 32 / 2x16 bit shift register or 64bit RAM
- 1 CLB (configurable logic block) = 2 * slices



Slice: consequences

=> efficient logic should fit into a slice (-> no slow routing)
(add / sub can span multiple slices due to dedicated CR lines)

Questions

- Q1: What is the maximum 1:N Mux for a **4** input LUT ?
- Q2: How many % of a 6 input LUT does the function ~(a & ~b | c) occupy ?

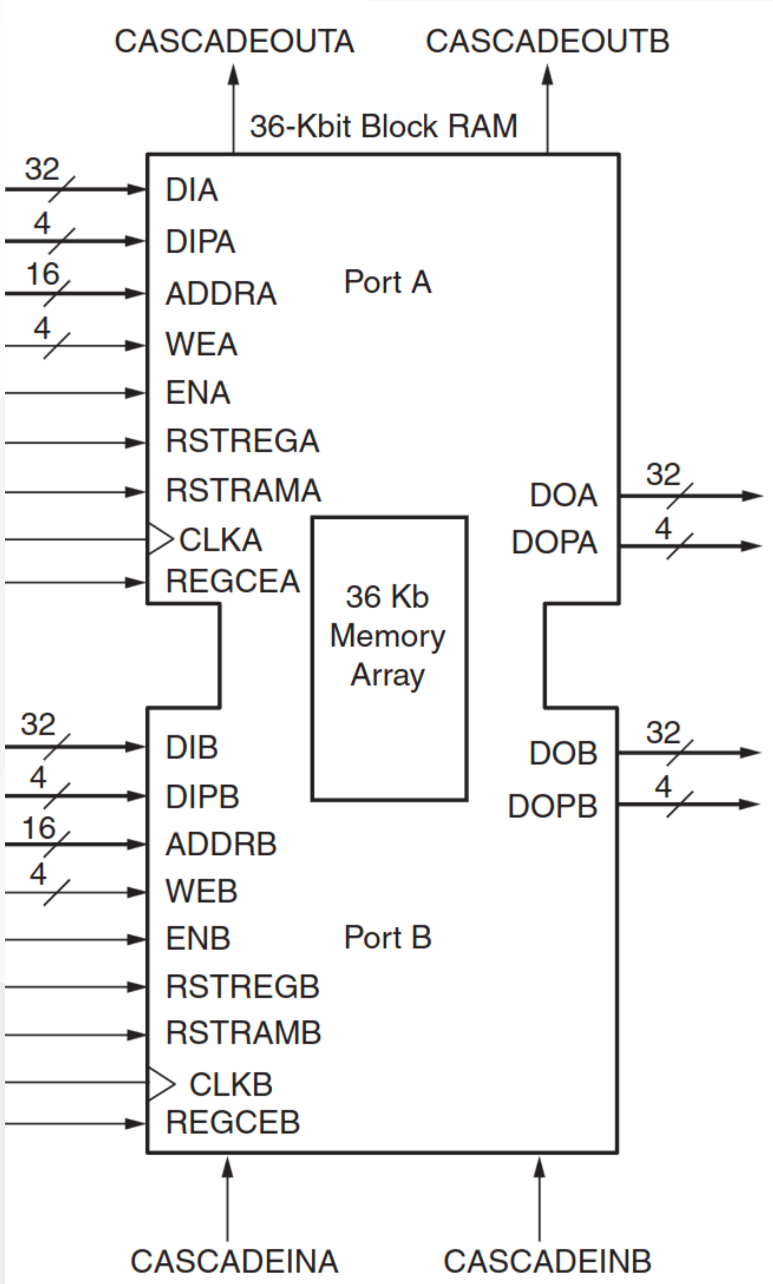
from / details: "7 Series FPGAs Configurable Logic Block User Guide UG474 (v1.8) September 27, 2016"

Hardcores

- DSP
  - 25x18 multiplier ---
  - use cases: (adaptive) filters (FIR), FFT, ...
  - XC7A200T: 740
- PLLs / Mixed-Mode Clock Manager
  - frequency multiplication M/N
  - phase shift
  - N phase output (e.g. for high resolution PWMs)
- GBit Transceiver, PCIe, ...

Hardcores: Block rams (1)

- synchronous read & write
- true dual port
- 1x 36kb or 2x 18k
- aspect ratio configurable
- cascade: 2x32kx1 => 64kx1
- XC7A200T: 365 (13140 Kbit)



Hardcores: Block rams (2)

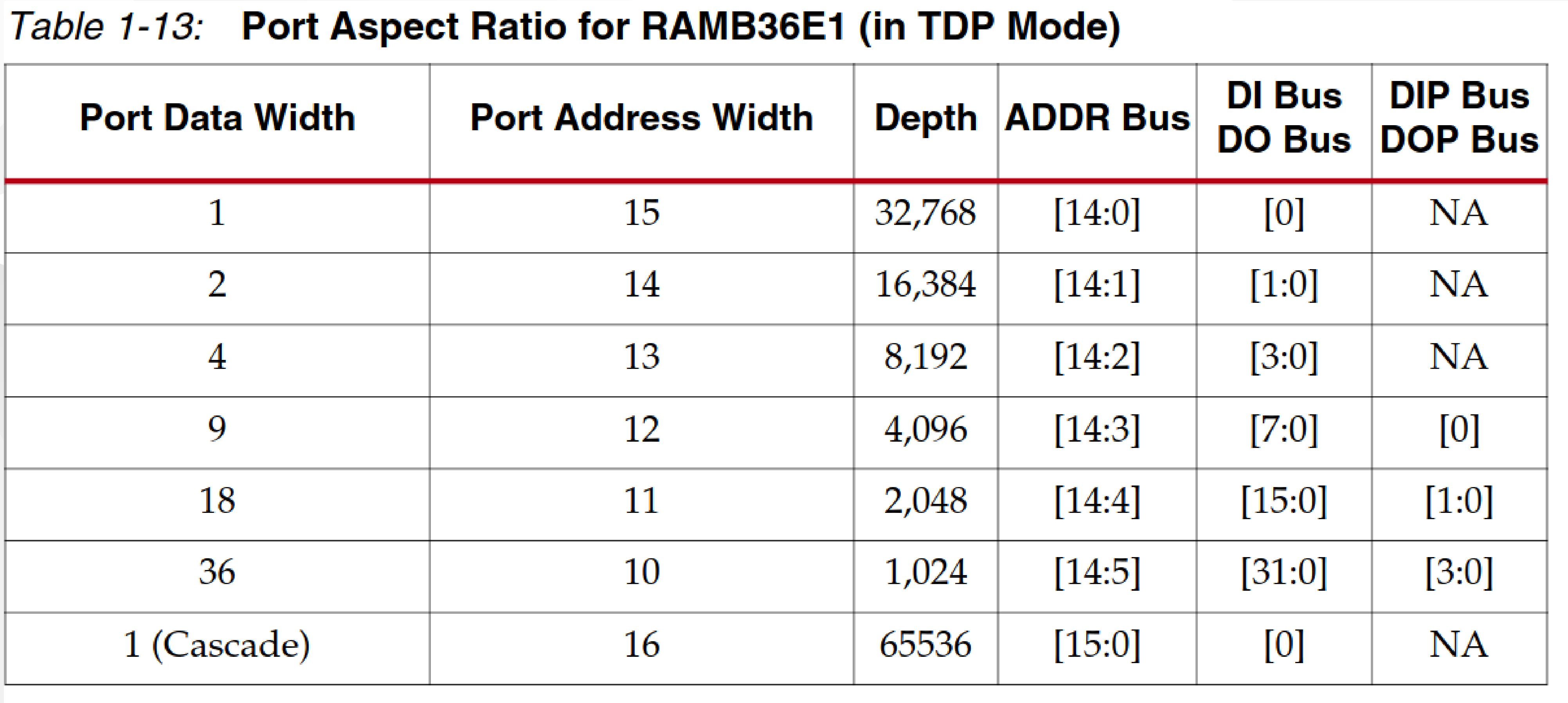


Table 1-13: **Port Aspect Ratio for RAMB36E1 (in TDP Mode)**

Port Data Width	Port Address Width	Depth	ADDR Bus	DI Bus DO Bus	DIP Bus DOP Bus
1	15	32,768	[14:0]	[0]	NA
2	14	16,384	[14:1]	[1:0]	NA
4	13	8,192	[14:2]	[3:0]	NA
9	12	4,096	[14:3]	[7:0]	[0]
18	11	2,048	[14:4]	[15:0]	[1:0]
36	10	1,024	[14:5]	[31:0]	[3:0]
1 (Cascade)	16	65536	[15:0]	[0]	NA

from / details: "7 Series FPGAs Memory Resources User Guide UG473 (v1.14) July 3, 2019"

How to Use

1. automatic inference by synthesis (preferred)
2. if synthesis fails instantiate "primitive" from library

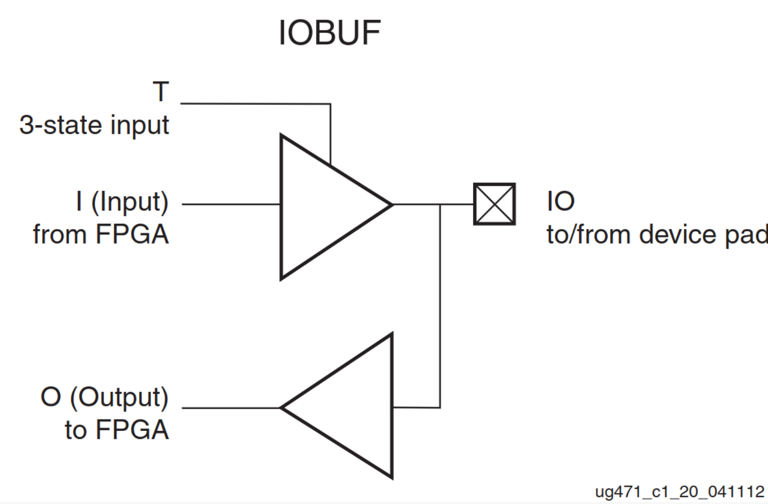
- DSP slices (esp. MAC, mul always works)

3. instantiate "primitives" from library

- Block rams (try synthesis, but keep checking !)
- (global) buffers / routing ressources (BUGF, BUGH, ...)
- IOs (already provided in rvlab)
- PLLs

How to Use: Example

```
module iocell_bidir #(
    parameter int Width = 1
) (
  inout  wire  [Width-1:0] pad,
  input  logic [Width-1:0] oe,
  input  logic [Width-1:0] out,
  output logic [Width-1:0] in
);
  generate
    genvar i;
    for (i = 0; i < Width; i++) begin : cellarray
      IOBUF #(
        .DRIVE       (4),
        .IBUF_LOW_PWR("TRUE"),
        .IOSTANDARD  ("DEFAULT"),
        .SLEW        ("SLOW")
      ) iocell (
        .O (in[i]),    // Buffer output
        .IO(pad[i]),   // Buffer inout port (connect directly to top-level port)
        .I (out[i]),   // Buffer input
        .T (~oe[i])    // 3-state enable input, high=input, low=output
      );
    end
  endgenerate
endmodule
```



How to Use (2)

Instance templates are in the: *"Vivado Design Suite 7 Series FPGA and Zynq-7000 SoC Libraries Guide UG953 (v2021.2) October 22, 2021"* (see Resources)

Hardware O(N)

+C, -C, +1, -1, +,-,*, /, <<, >>, |, &, ~, 1:N mux, M:N mux, sin,cos, >=, <=, ==

O(k):

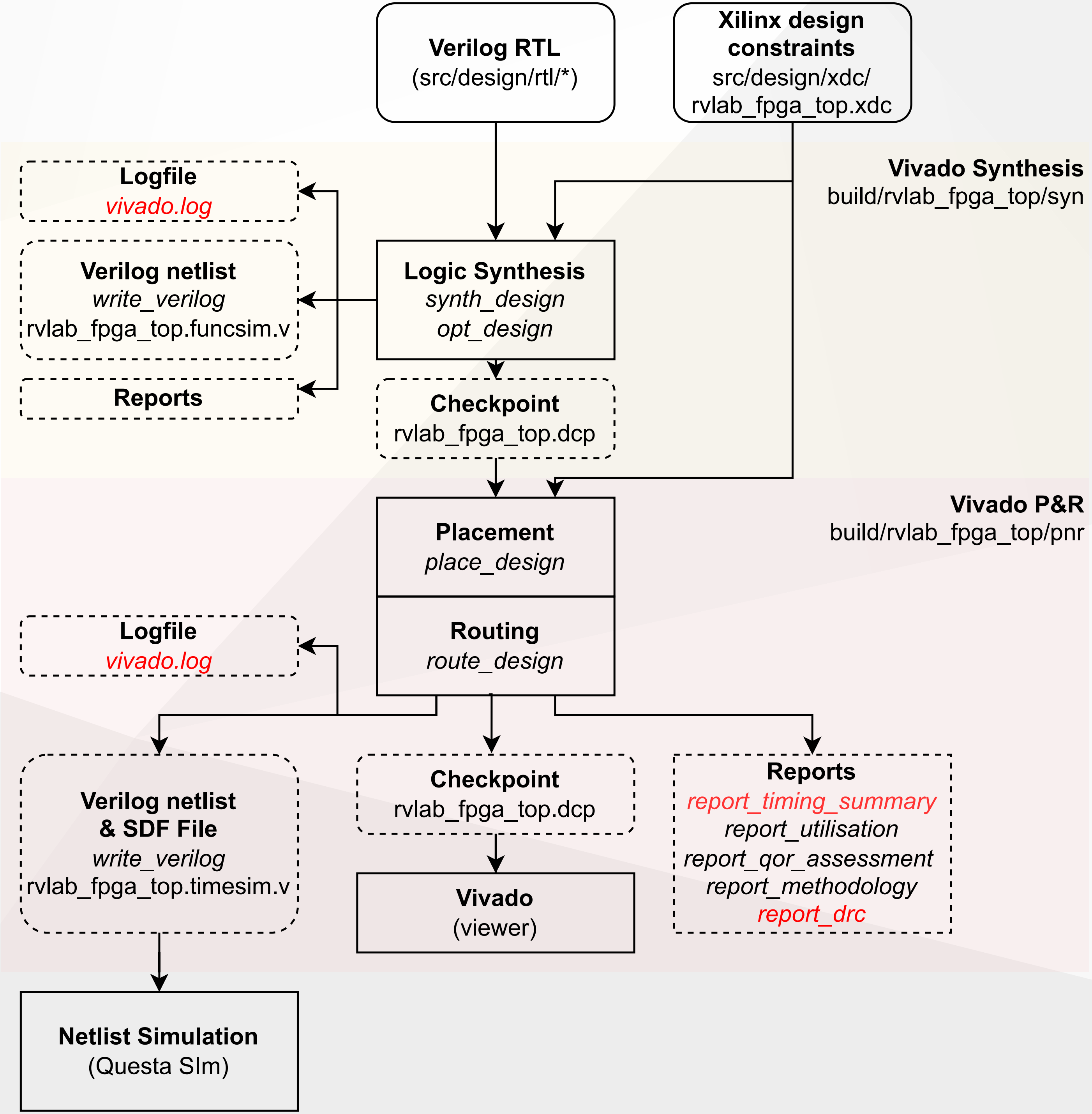
O(n):

O(N log N):

O(N^2):

don't use:

FPGA Design Flow



Verilog RTL
(src/design/rtl/*)

Xilinx design constraints
src/design/xdc/
rvlab_fpga_top.xdc

Vivado Synthesis
build/rvlab_fpga_top/syn

Logfile
vivado.log

Verilog netlist
write_verilog
rvlab_fpga_top.funcsim.v

Reports

Logic Synthesis
synth_design
opt_design

Checkpoint
rvlab_fpga_top.dcp

Vivado P&R
build/rvlab_fpga_top/pnr

Placement
place_design

Routing
route_design

Logfile
vivado.log

Verilog netlist
& SDF File
write_verilog
rvlab_fpga_top.timesim.v

Checkpoint
rvlab_fpga_top.dcp

Reports
report_timing_summary
report_utilisation
report_qor_assessment
report_methodology
report_drc

Vivado
(viewer)

Netlist Simulation
(Questa SIm)

XDC (excerpt!)

```
# IO Placement
set_property -dict { PACKAGE_PIN R4    IOSTANDARD LVCMOS33 } [get_ports { clk_100mhz_i }];
# Clock input
create_clock -add -name clk_100mhz -period 10.00 -waveform {0 5} [get_ports clk_100mhz_i]
create_generated_clock -name sys_clk [get_pin clkmgr_i/mmcm_i/CLKOUT0]

# JTAG via FT2232H
set_property -dict ...
...
create_clock -add -name tck -period 20.00 -waveform {0 10} [get_ports jtag_tck_i]
set_output_delay -clock tck 0.0 [get_ports "jtag_tdo_o"]
set_input_delay -clock tck 10.0 [get_ports "jtag_tdi_i jtag_tms_i"]
set_input_delay -clock tck 0.0 [get_ports "jtag_trst_ni"]

# Inputs without timing constraints; the following commands remove the warnings:
set_false_path -from [get_ports "jtag_trst_ni"]

# Inter-clock paths: tck, sys_clk and their derived clocks are asynchronous from all others
set_clock_groups -group [get_clocks tck] -asynchronous
set_clock_groups -group [get_clocks clk] -asynchronous

... (lots of IO placements)
```

Netlist simulation with Questa Sim

- Inputs
  - verilog netlist of the design (from P&R)
  - standard delay file (SDF) (from P&R)
  - verilog models of FPGA primitives (LUT, FFs, BRAM, ...)
  - (same as RTL simulation: test bench & tcl scripts)
- Outputs
  - value trace of all signals showing actual timing
  - timing (setup / hold) warnings of the FPGA primitives

```
flow systb_rlight.sim_pnrtime_questa
```