Introduction to FPGA

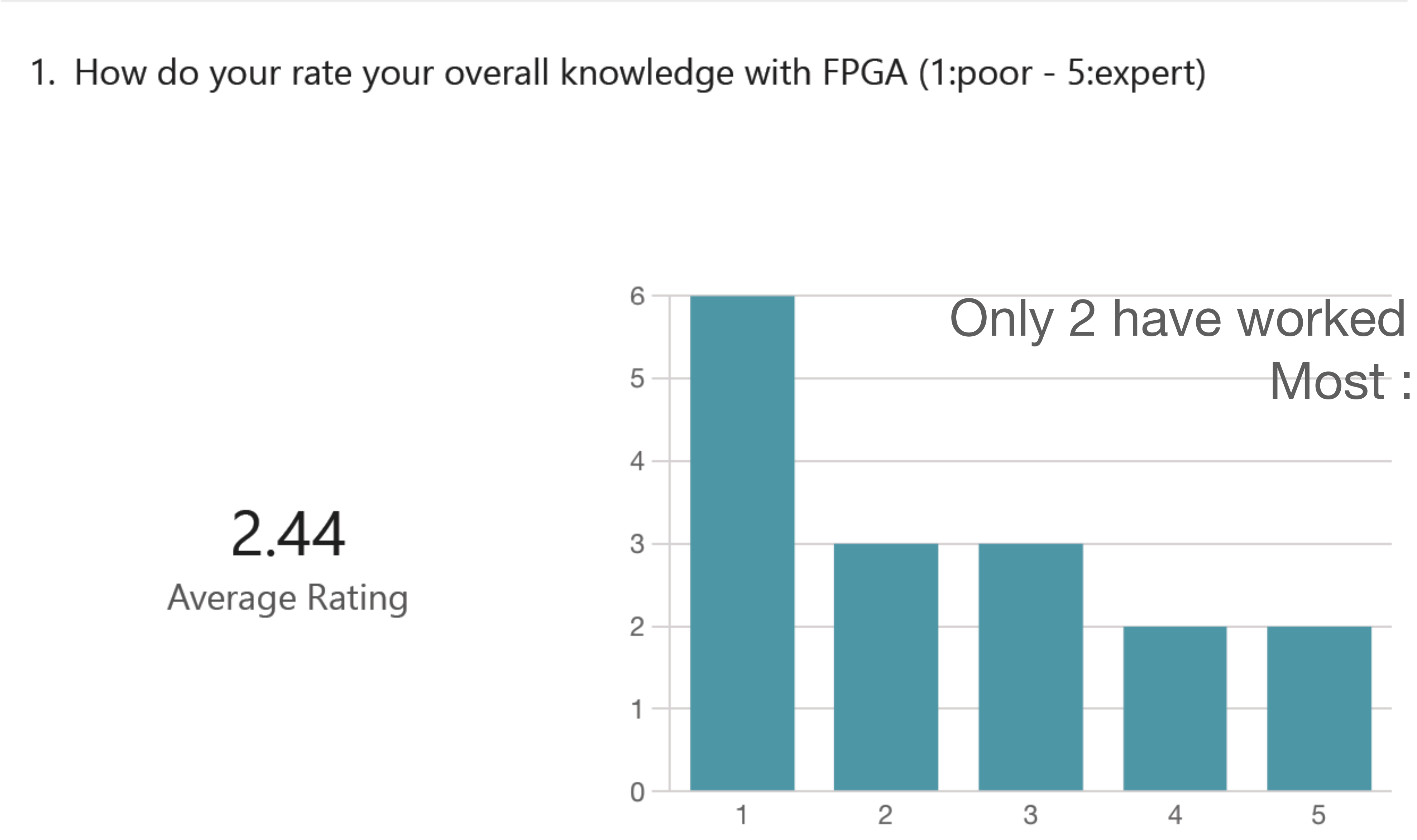
Corso INFN “Tecniche Di Machine Learning Con Dispositivi FPGA per Gli Esperimenti Di Fisica Delle Particelle”

Riccardo Travaglini - 02/11/2022

Disclaimer:
This presentation is intended only for personal use of this course participants;
Do not distribute - dot not modify
Use only for education purposes

Outline

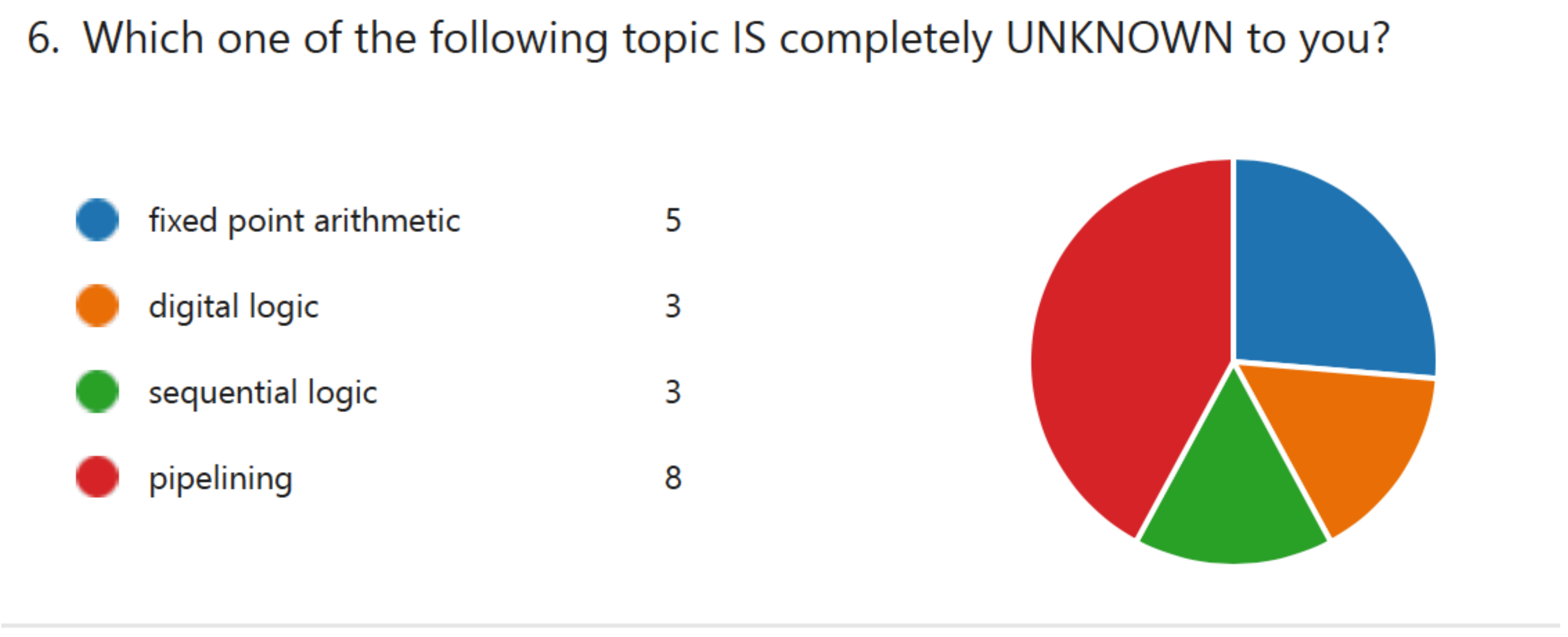
- Digital circuits: combinational, sequential and synchronous logic
- FPGA hardware - Xilinx devices as an example
- How to design



1. How do your rate your overall knowledge with FPGA (1:poor - 5:expert)

2.44
Average Rating

Only 2 have worked on > 10 FPGA project
Most : 0 and 1



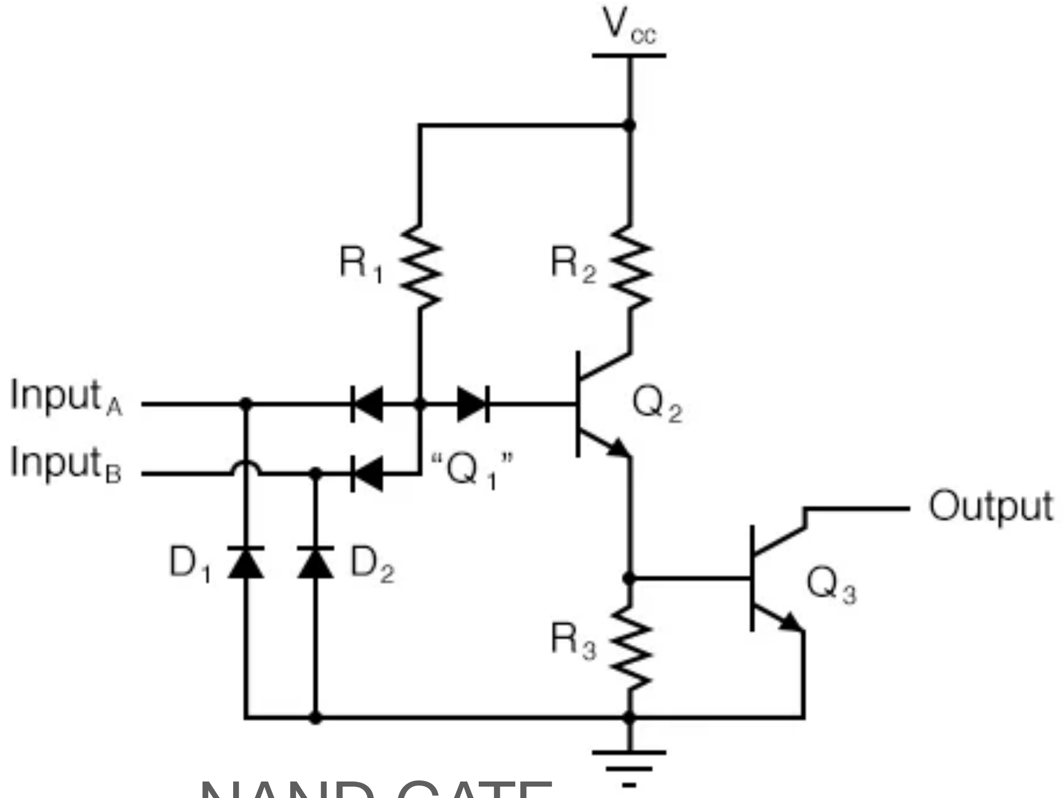
6. Which one of the following topic IS completely UNKNOWN to you?

●	fixed point arithmetic	5
●	digital logic	3
●	sequential logic	3
●	pipelining	8

Field Programmable Gate Array

Gate? Gate Array? Field Programmable?

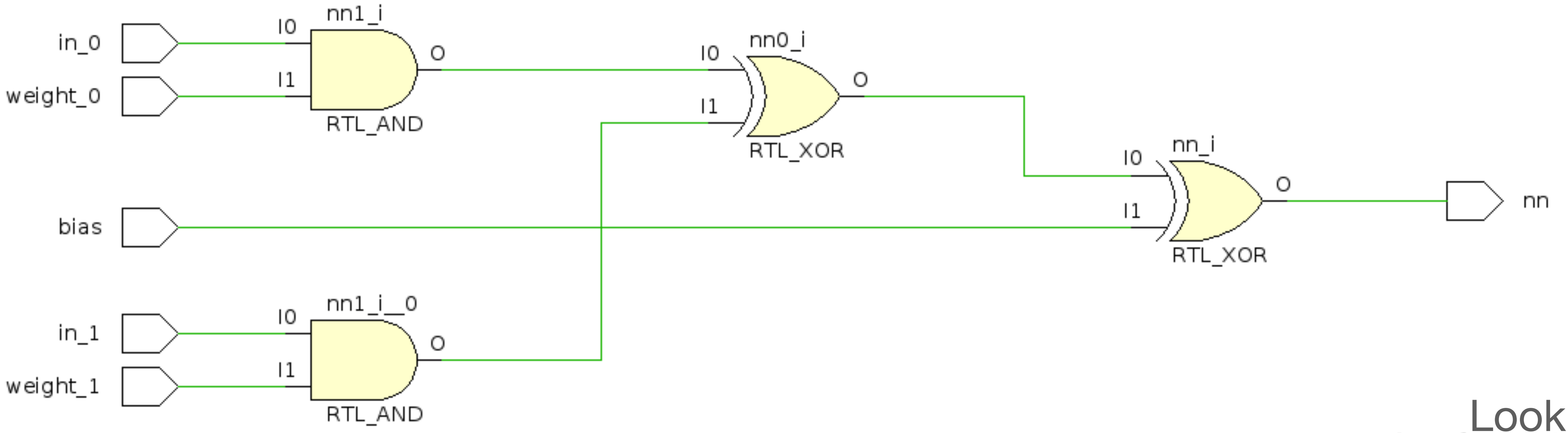
- Data => bits; Boolean algebra => bit operations
- Logic gates are basic build blocks of Boolean expressions (AND, OR, NAND, XOR….)
- Digital circuits: electrical circuits where I/O voltages (rarely current) take only 2 values (high and low) ,so representing bits
- Gates are circuits implementing logic gates
- Digital circuits + gates => combinational logic (electric/electronic circuits implementing boolean logic)
- FPGA: plenty of gates (array in a broader sense)
- No function at the power up: FPGA must be *programmed*
- *Program* vs*. Configure*



NAND GATE

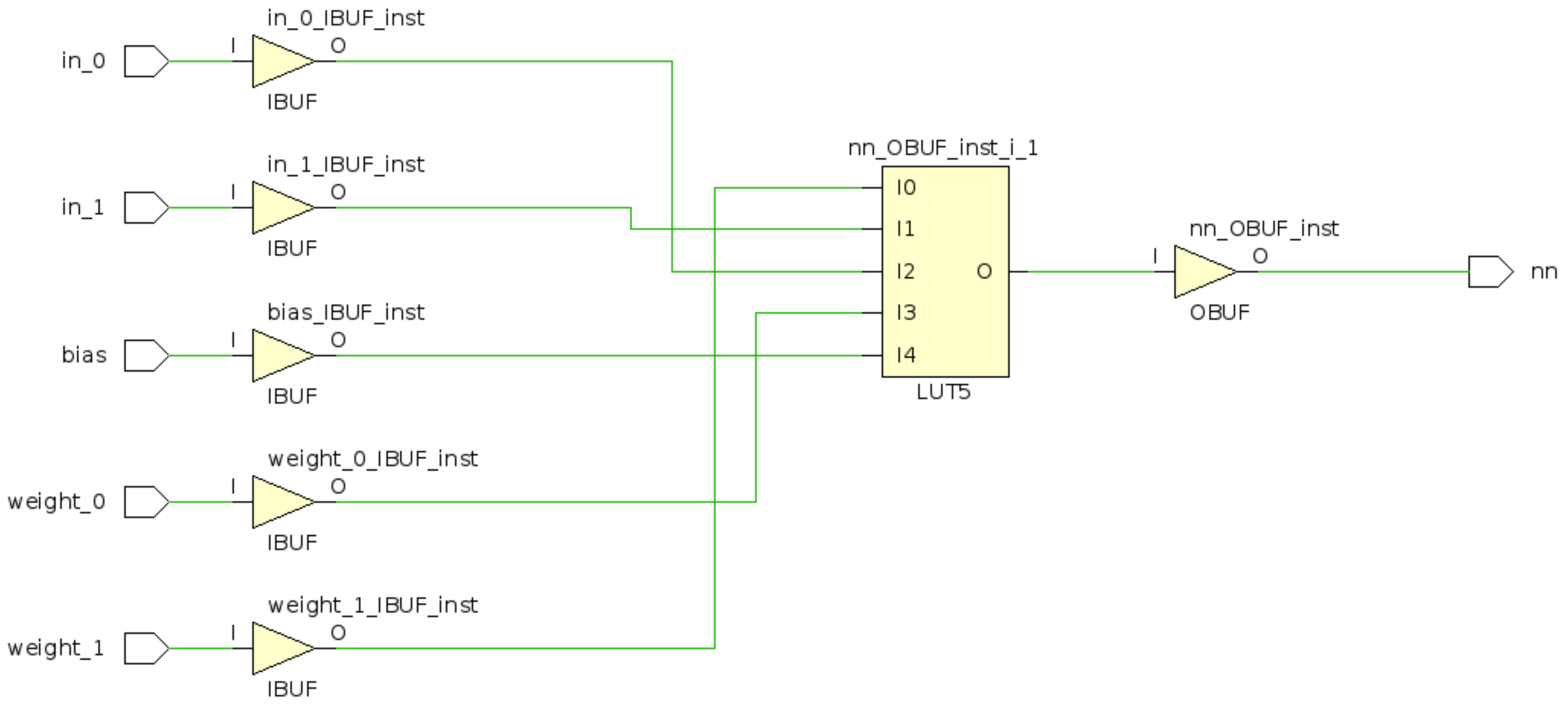
Combinational Logic

Basic gates in modern FPGA



nn = in_0 * weight_0 + in_1 * weight_1 + bias

Output is only function of inputs (memory-less)



Look-up Tables

The ~~function generators~~ can implement:

- Any arbitrarily defined six-input Boolean function.
- Two arbitrarily defined five-input Boolean functions, as long as these two functions share common inputs.
- Two arbitrarily defined Boolean functions of three and two inputs or less.

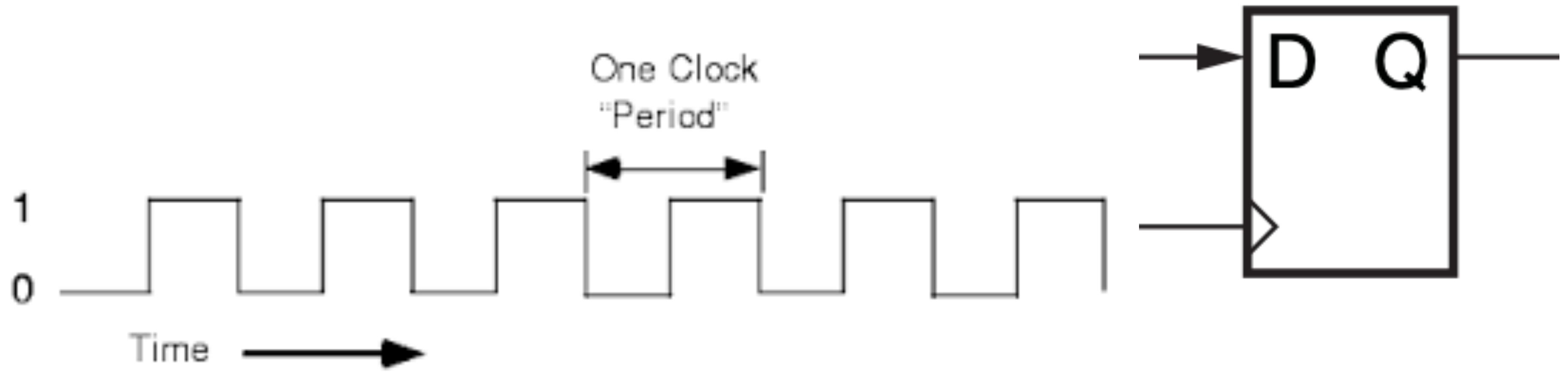
Xilinx - UltraScale Architecture Configurable Logic Block User Guide (UG574)

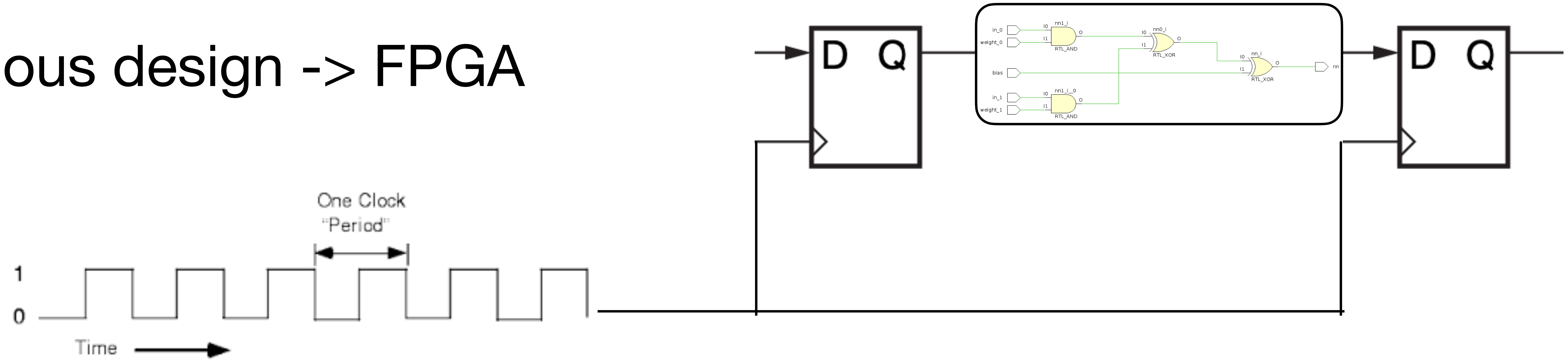
Others combinational building blocks: MUX and Carry Logic

Sequential logic and synchronous design

Add memory

- Sequential logic: output function of inputs as internal state (~memory of past input)
- Memory component: flip flop
- Clock signal
- Full synchronous design -> FPGA

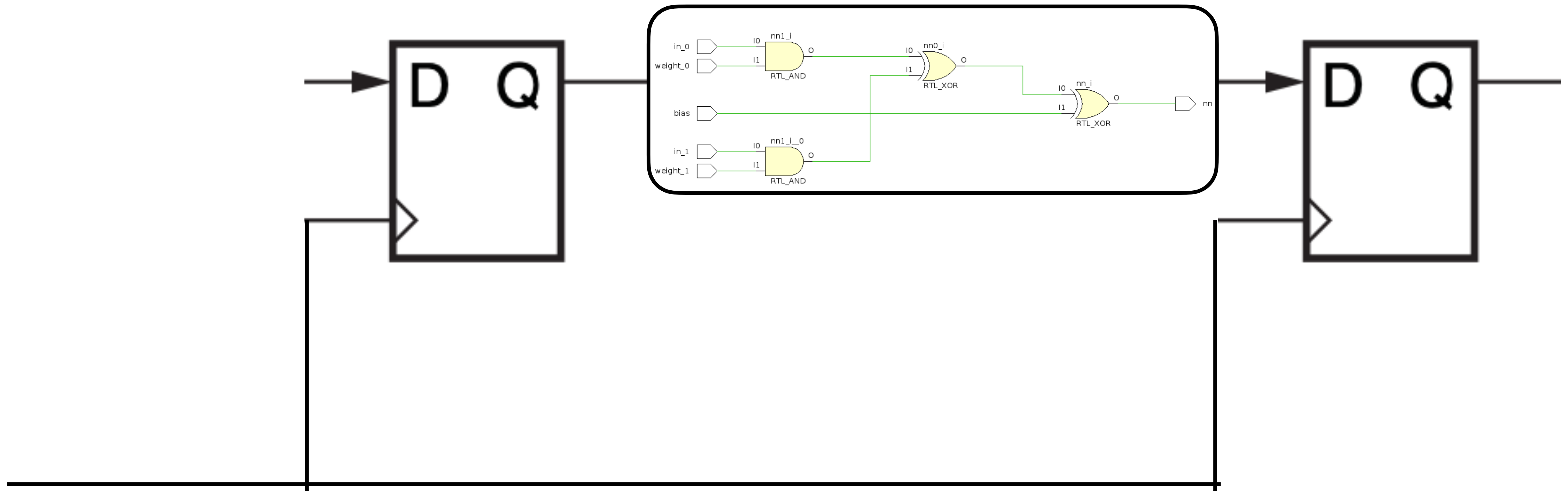




Latency - Throughput

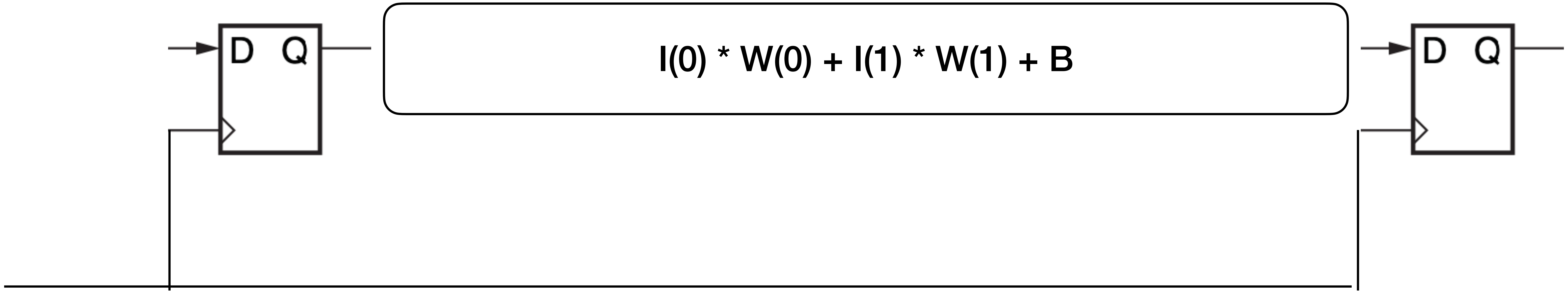
- Latency: timing from inputs to outputs
  - In clock cycles
  - Absolute time (clock-cycles * clock-period)

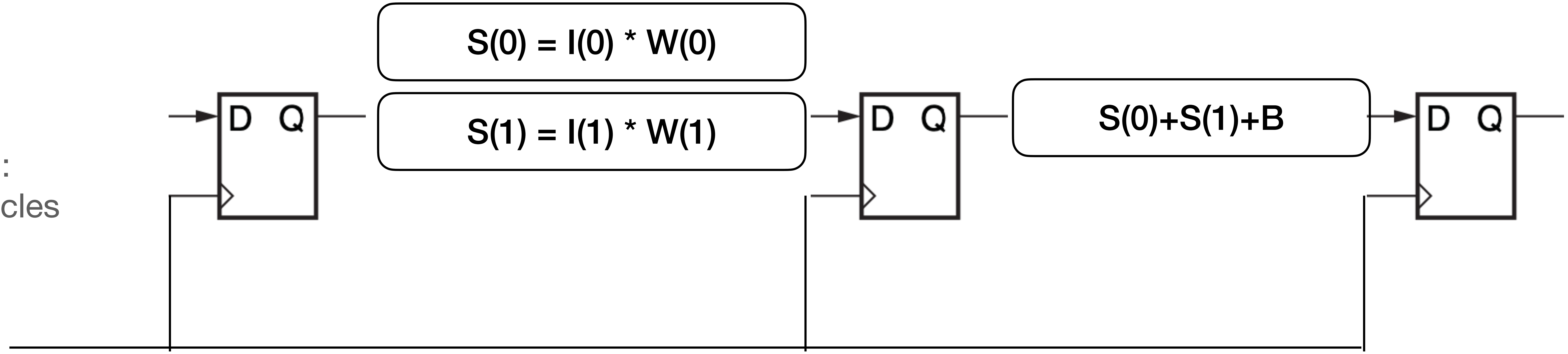
- Throughput: outputs in the time unit
  - number of output data per clock-cycles



Pipelining

What about latency and throughput?

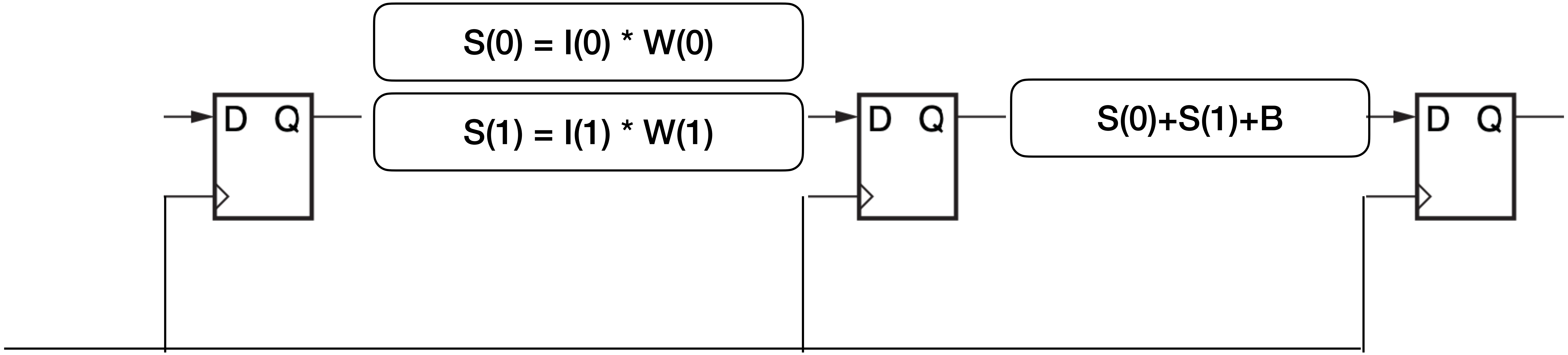




Initiation Interval:
How many clock cycles
between 2
distinct inputs?

Feedback and parallelising

Initiation interval - (loop unrolling)

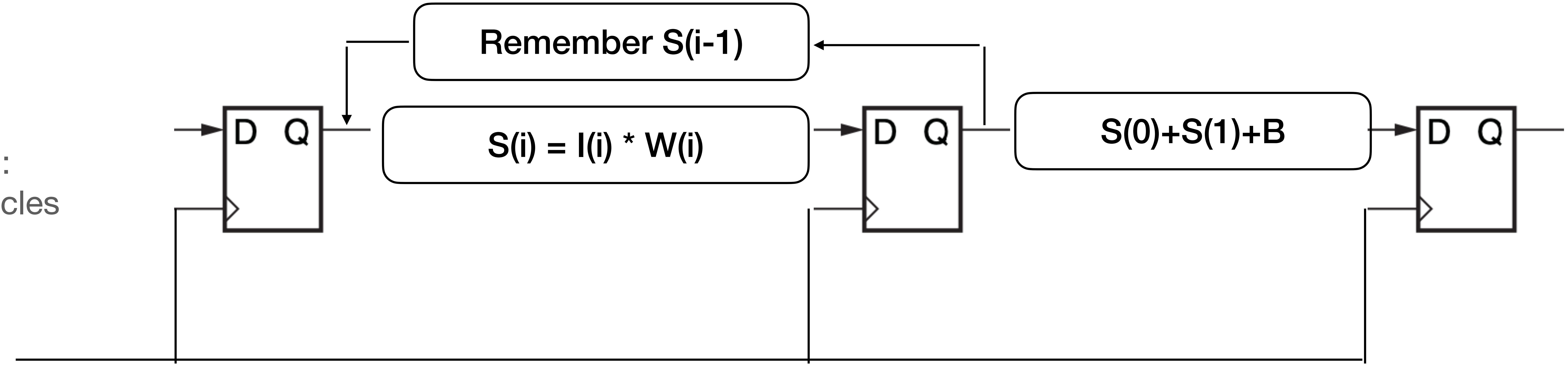


S(0) = I(0) * W(0)

S(1) = I(1) * W(1)

S(0)+S(1)+B

Initiation Interval:
How many clock cycles
between 2
distinct inputs?



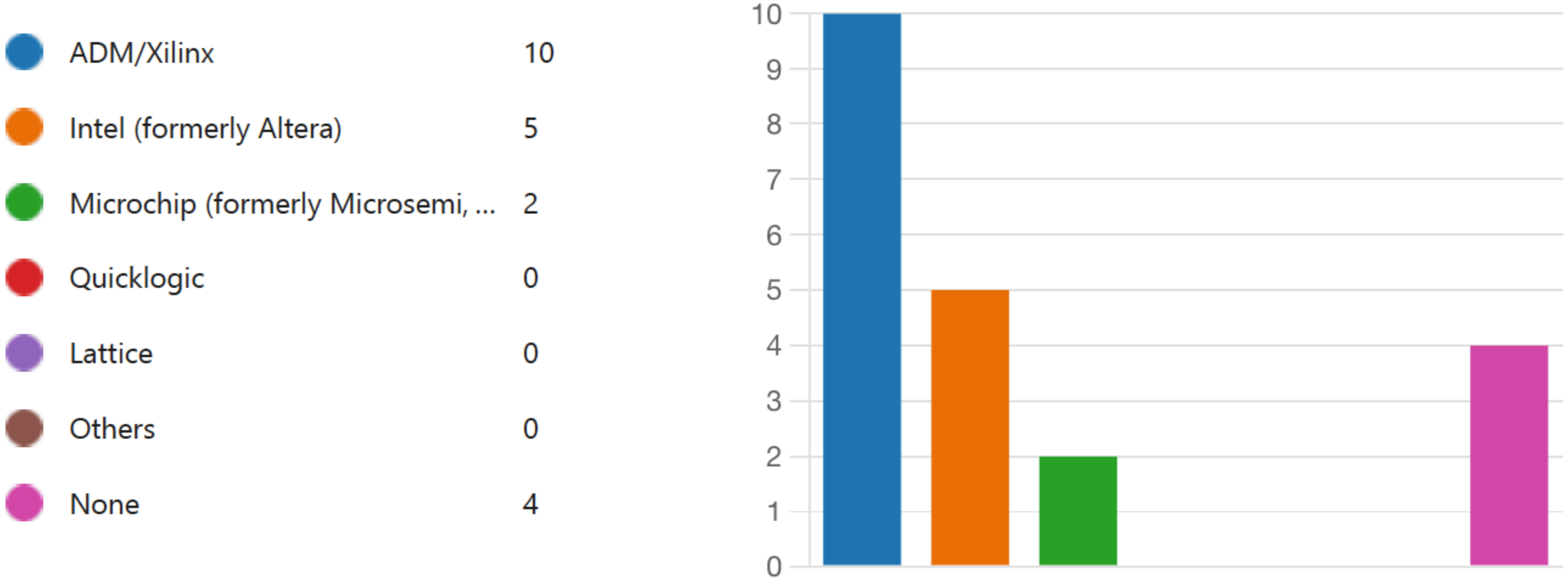
Remember S(i-1)

S(i) = I(i) * W(i)

S(0)+S(1)+B

Outline

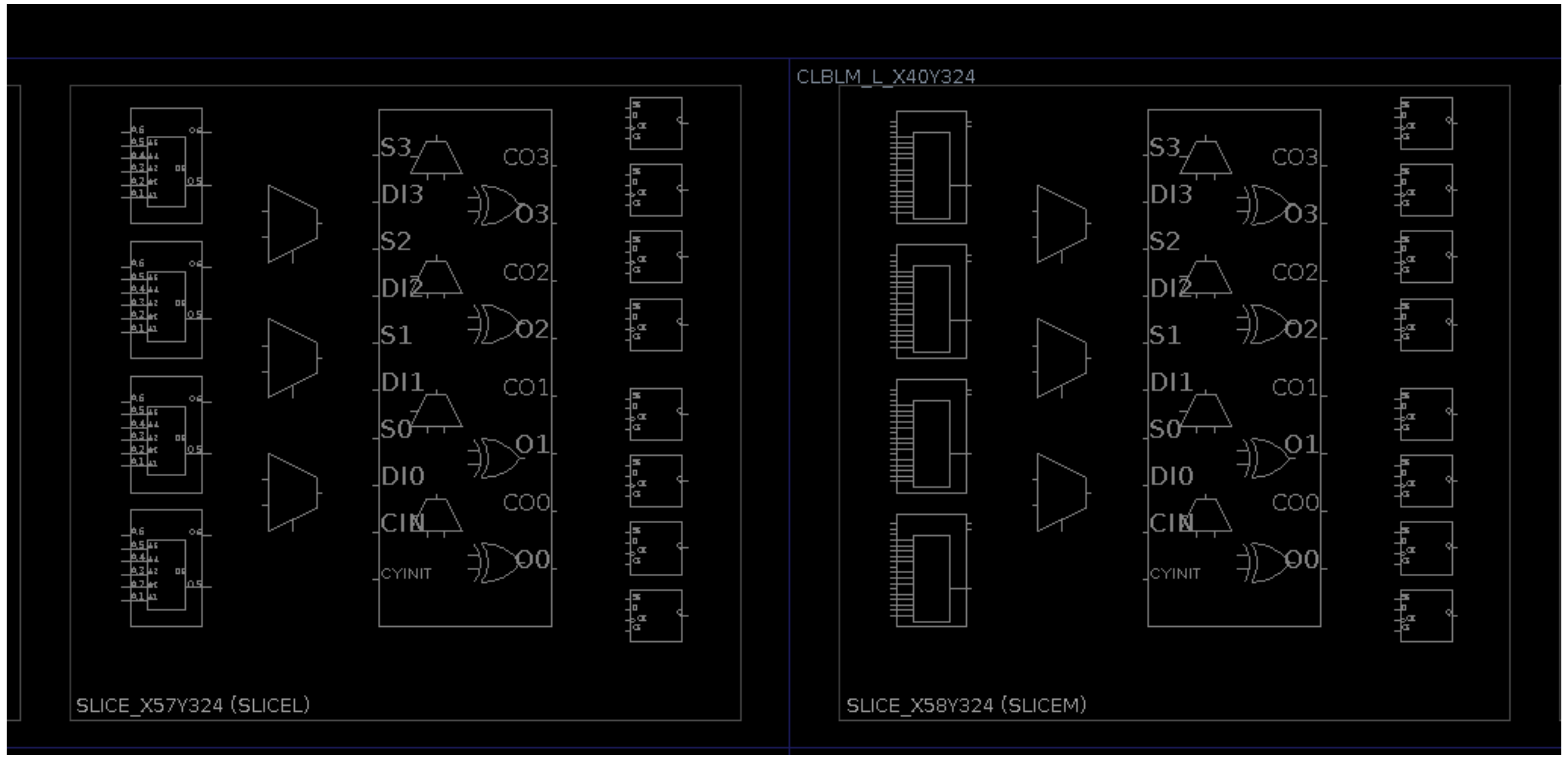
- Digital circuits: combinational, sequential and synchronous logic
- FPGA hardware - Xilinx devices as an example
- How to design

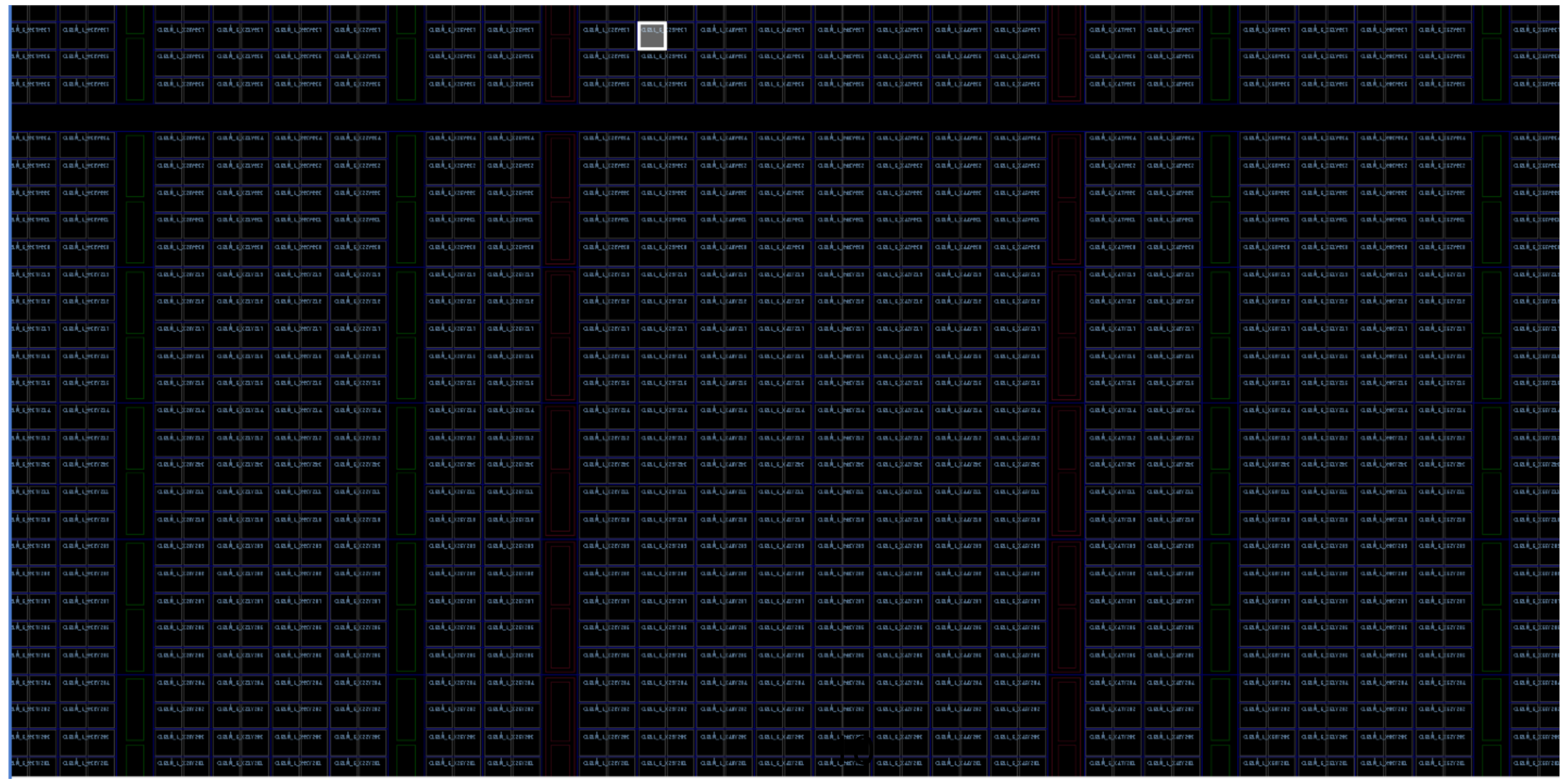


●	ADM/Xilinx	10
●	Intel (formerly Altera)	5
●	Microchip (formerly Microsemi, ...	2
●	Quicklogic	0
●	Lattice	0
●	Others	0
●	None	4

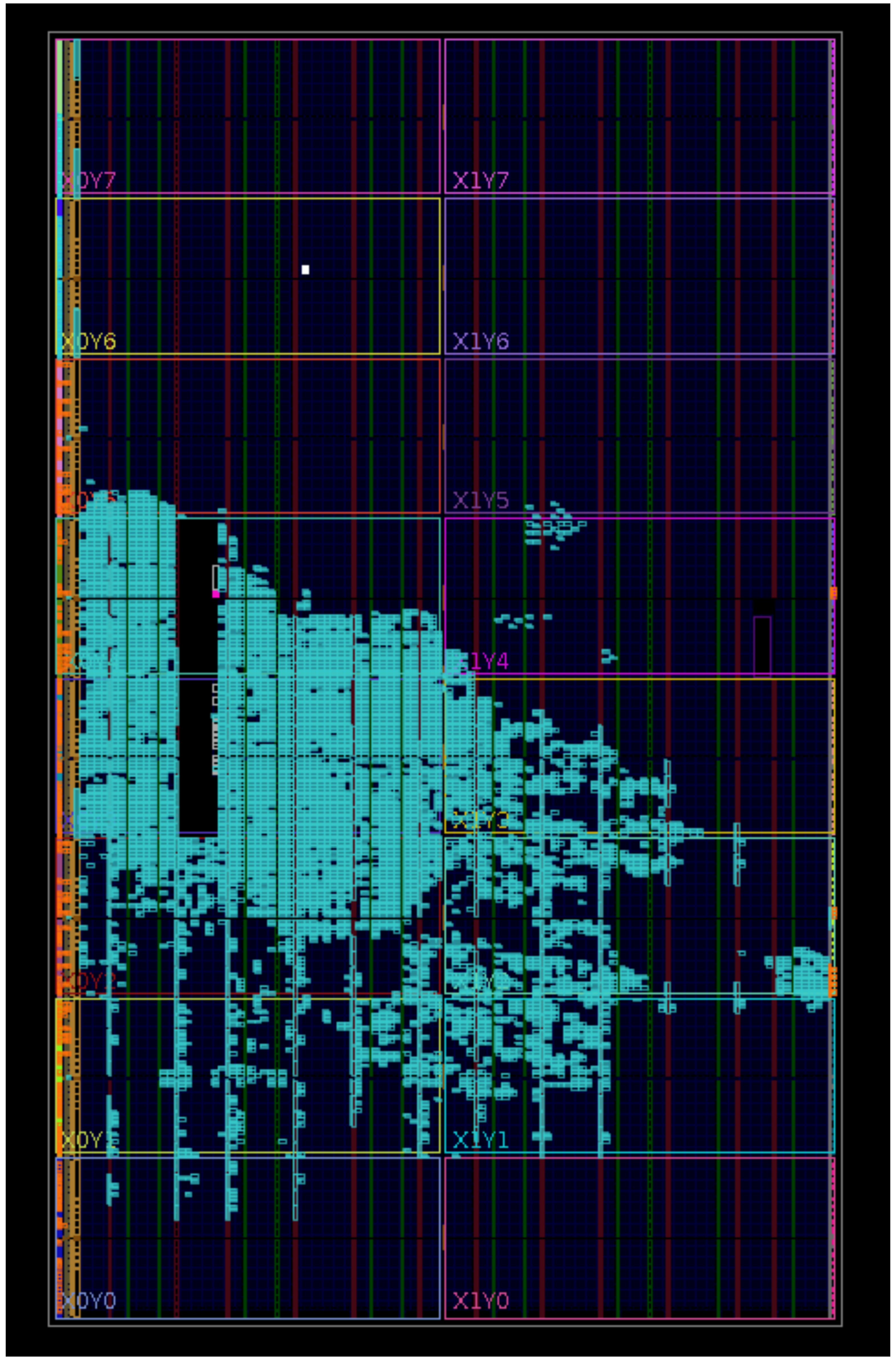
Configurable Logic Block

LUT, MUX, Carry Logic, Flipflops





Program FPGA
means
configure gates (f.i. LUTs) and
enable selected connections
among gates and elements



Memory and processing

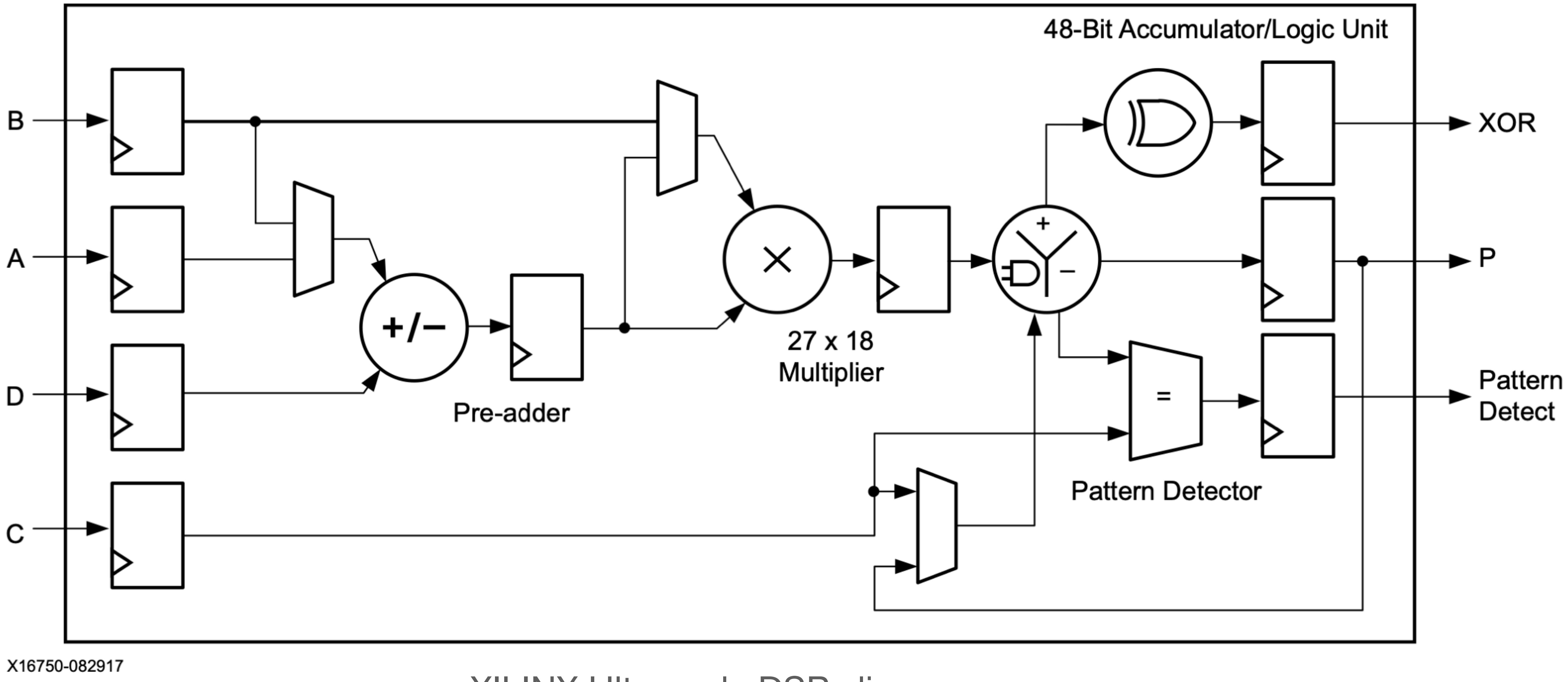
Hard silicon resources

Several kind of RAMs;

- BRAM
- Distributed RAM
- Ultra RAM

- FIFOs

Digital signal processing slices

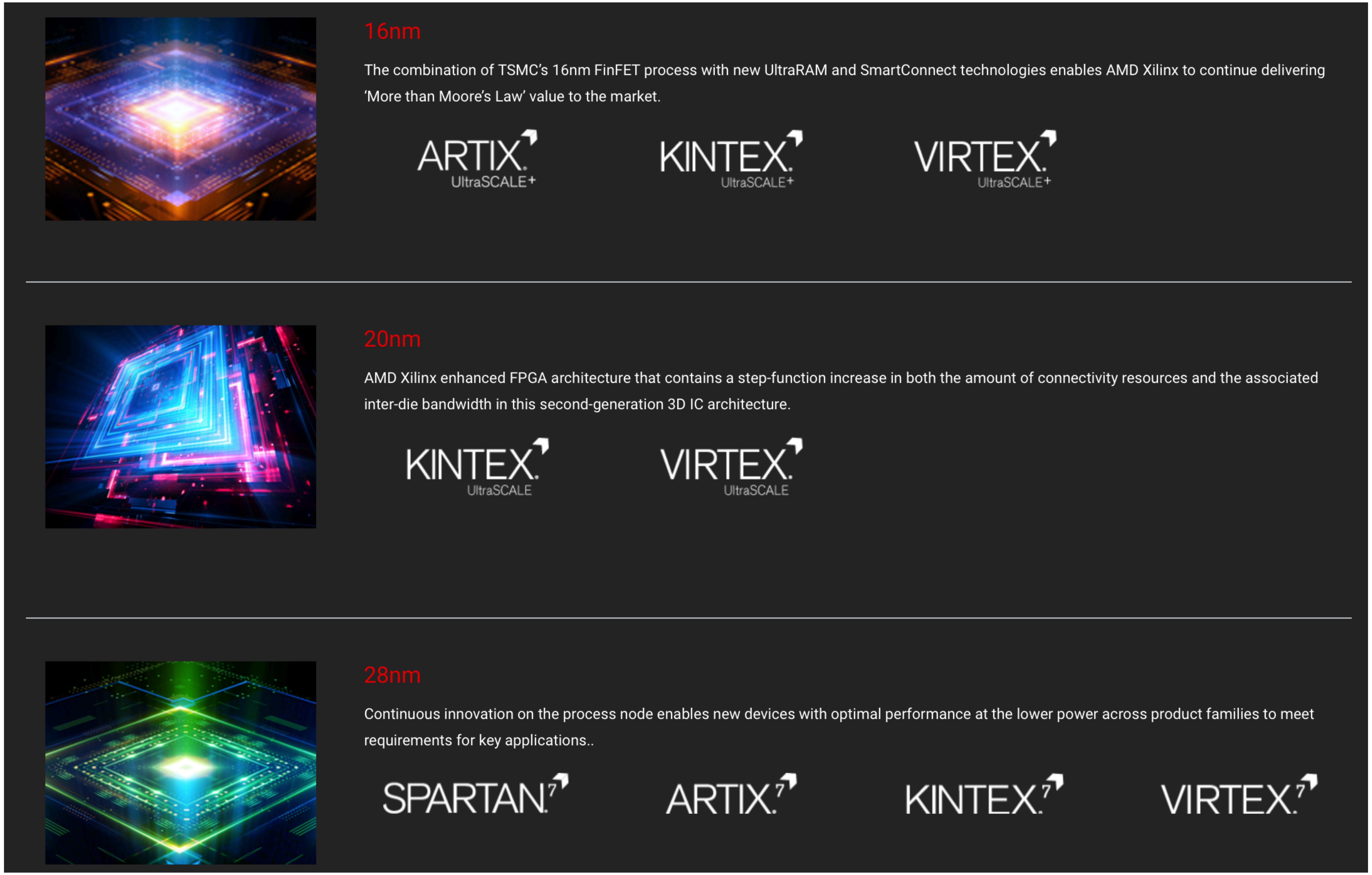


XILINX Ultrascale DSP slice

Input/Output Interface

- Configure I/O: map design I/O on physical chip pins (depends on PCB design, chip package)
- Several electrical standard are compatible (LVCMOS, LVDS, …)
- High speed transceivers (up to tenth of Gbit/s) to interface:
  - Fast busses (PCIe, …)
  - Optical driver/receiver
  - High bandwidth memory (DDRx, HBMx)

Xilinx Families



16nm

The combination of TSMC’s 16nm FinFET process with new UltraRAM and SmartConnect technologies enables AMD Xilinx to continue delivering ‘More than Moore’s Law’ value to the market.

ARTIX UltraSCALE+ KINTEX UltraSCALE+ VIRTEX UltraSCALE+

20nm

AMD Xilinx enhanced FPGA architecture that contains a step-function increase in both the amount of connectivity resources and the associated inter-die bandwidth in this second-generation 3D IC architecture.

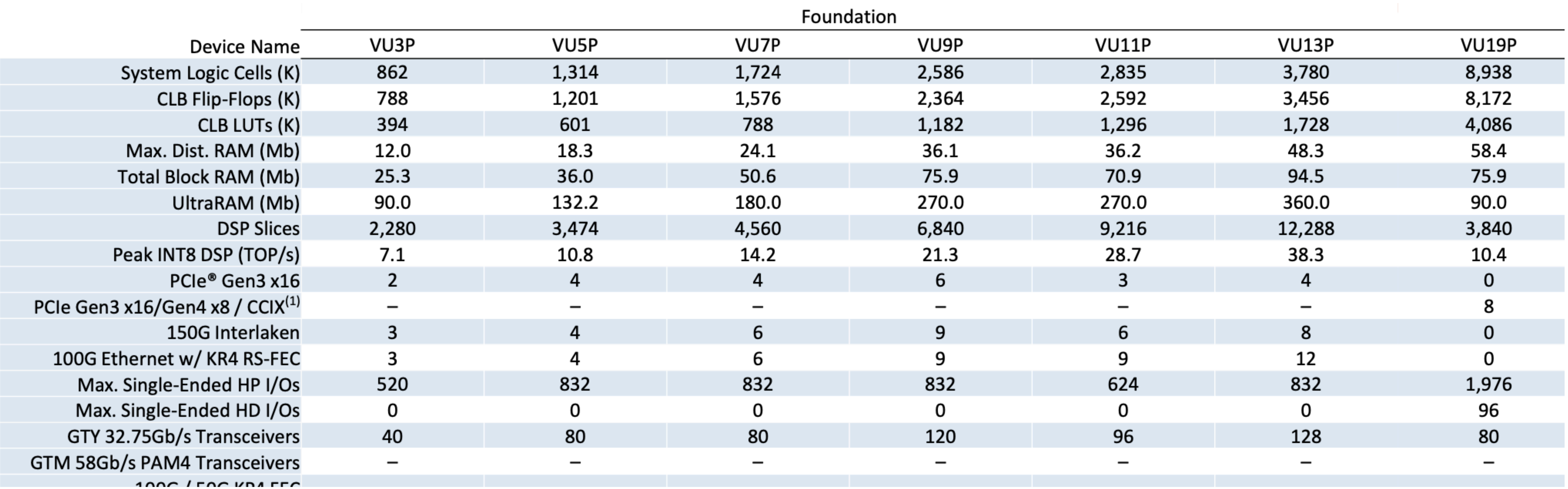
KINTEX UltraSCALE VIRTEX UltraSCALE

28nm

Continuous innovation on the process node enables new devices with optimal performance at the lower power across product families to meet requirements for key applications..

SPARTAN 7 ARTIX 7 KINTEX 7 VIRTEX 7

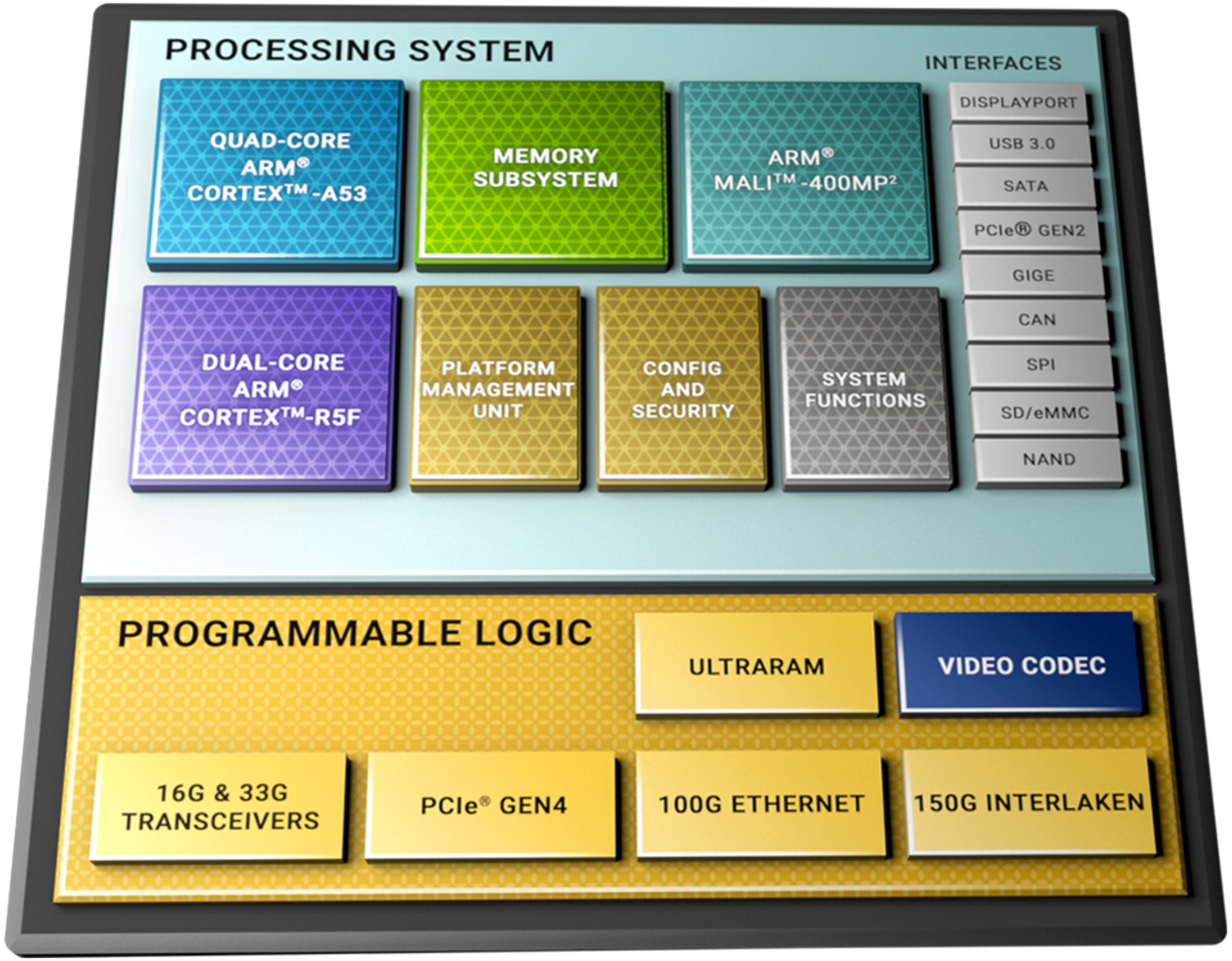
Xilinx Virtex Ultrascale+



	Foundation						
Device Name	VU3P	VU5P	VU7P	VU9P	VU11P	VU13P	VU19P
System Logic Cells (K)	862	1,314	1,724	2,586	2,835	3,780	8,938
CLB Flip-Flops (K)	788	1,201	1,576	2,364	2,592	3,456	8,172
CLB LUTs (K)	394	601	788	1,182	1,296	1,728	4,086
Max. Dist. RAM (Mb)	12.0	18.3	24.1	36.1	36.2	48.3	58.4
Total Block RAM (Mb)	25.3	36.0	50.6	75.9	70.9	94.5	75.9
UltraRAM (Mb)	90.0	132.2	180.0	270.0	270.0	360.0	90.0
DSP Slices	2,280	3,474	4,560	6,840	9,216	12,288	3,840
Peak INT8 DSP (TOP/s)	7.1	10.8	14.2	21.3	28.7	38.3	10.4
PCIe® Gen3 x16	2	4	4	6	3	4	0
PCIe Gen3 x16/Gen4 x8 / CCIX[1]	–	–	–	–	–	–	8
150G Interlaken	3	4	6	9	6	8	0
100G Ethernet w/ KR4 RS-FEC	3	4	6	9	9	12	0
Max. Single-Ended HP I/Os	520	832	832	832	624	832	1,976
Max. Single-Ended HD I/Os	0	0	0	0	0	0	96
GTY 32.75Gb/s Transceivers	40	80	80	120	96	128	80
GTM 58Gb/s PAM4 Transceivers	–	–	–	–	–	–	–
100G / 50G KP4 FEC							

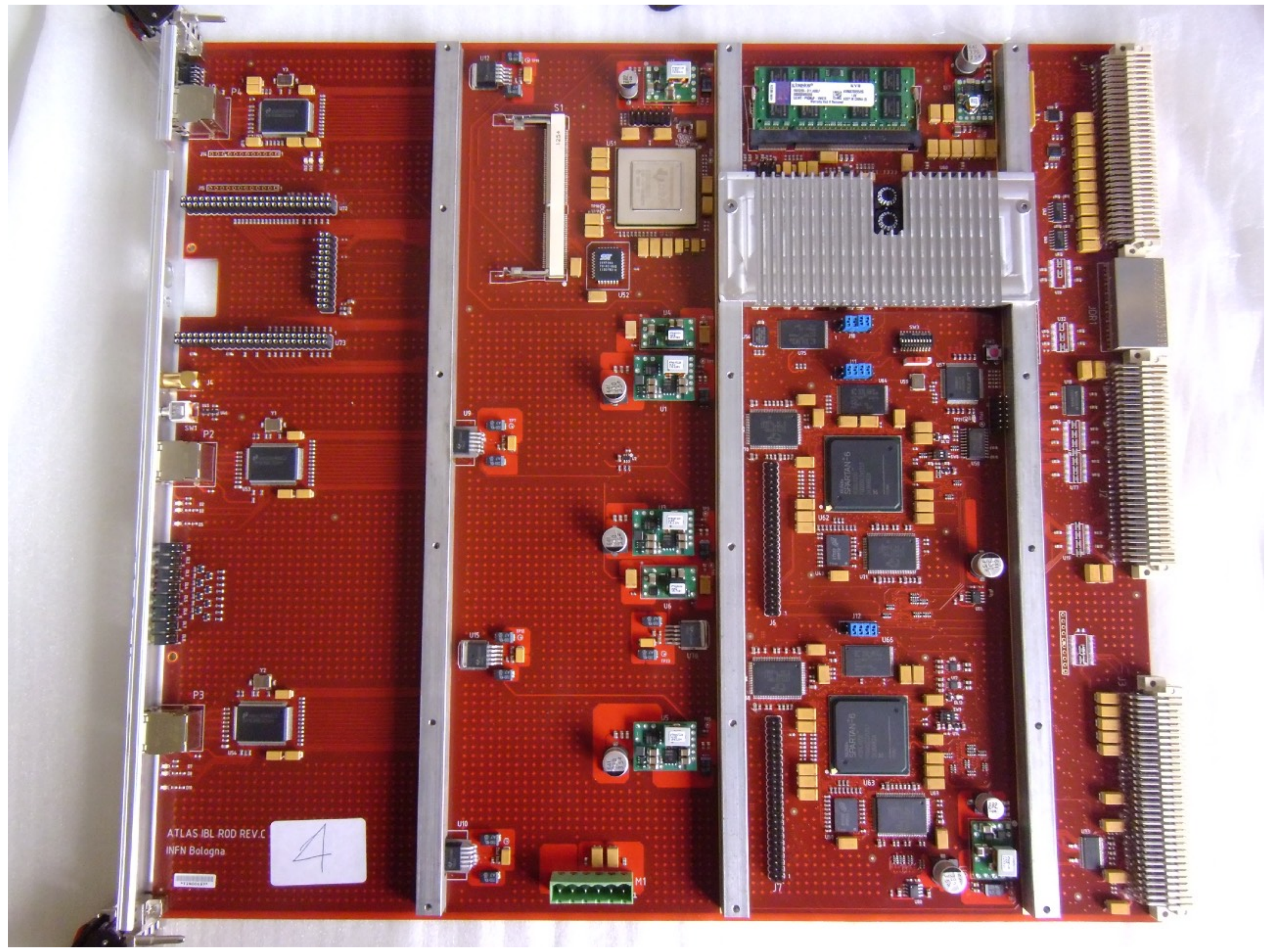
System on Chip

- PS: complete configurable and autonomous processing system
  - Running custom sw app (even Linux!)
- PL (called Fabric)
  - Custom peripheral IF
  - Accelerators
  - …
- Connected via AXI busses

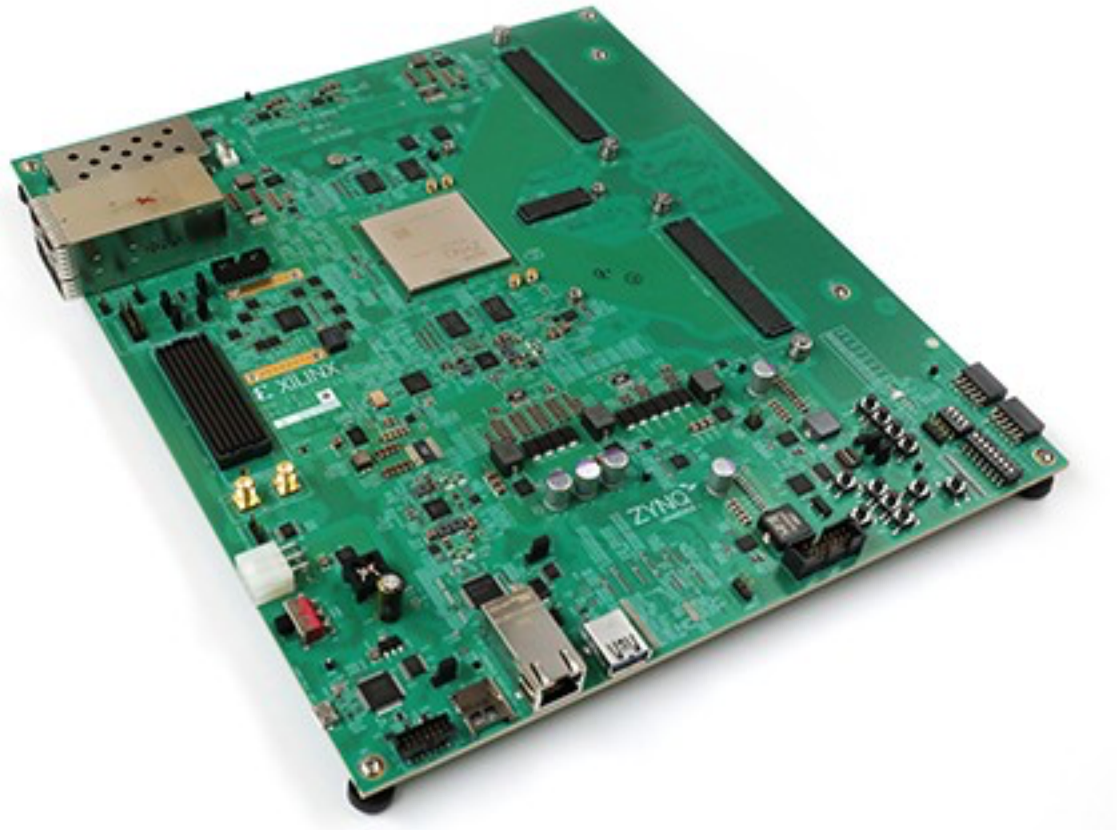


Put them in action

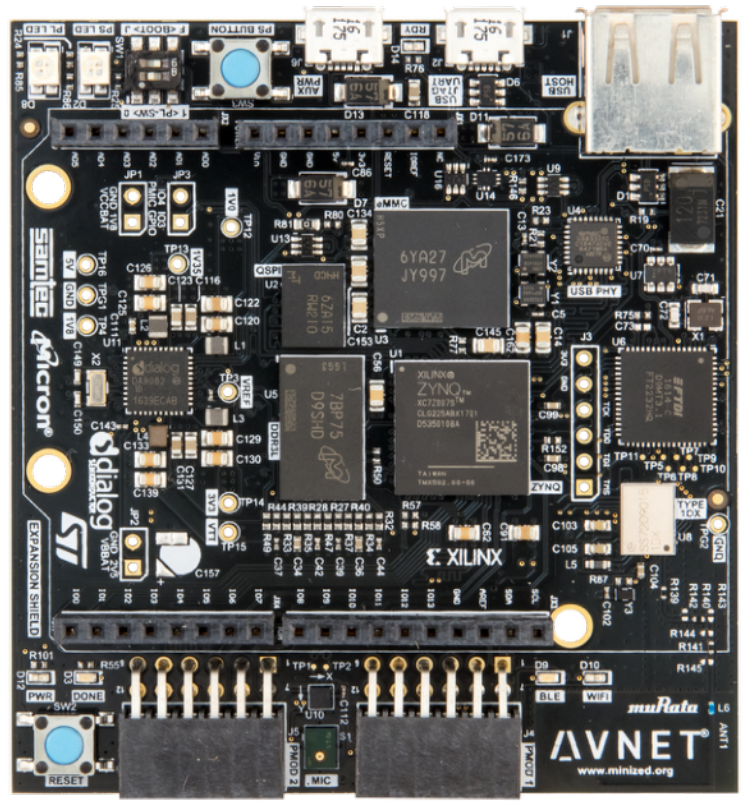
Ad-hoc board, Evaluation & Kits, System on Modules



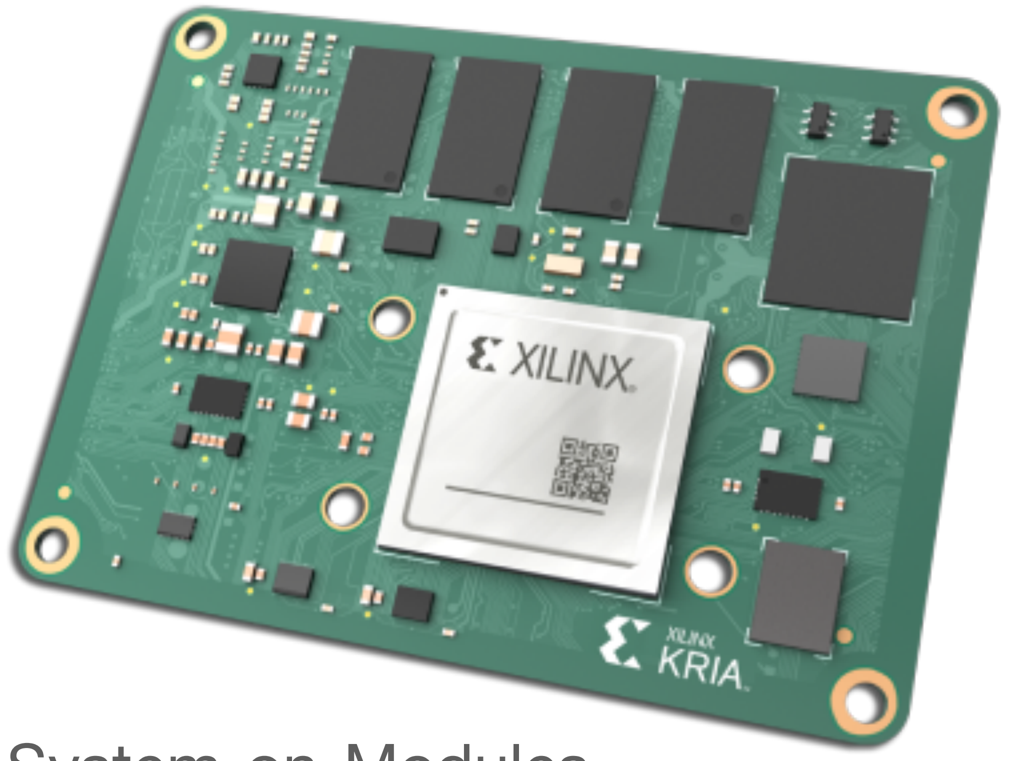
ATLAS IBL Read Out Driver
Form factor: VME 9U
Xilinx Fpga: 1 Virtex-5 - 2 Spartan 6



Zynq UltraScale+ RFSoC ZCU216 Evaluation Kit



Avnet MInized - Zynq 7000

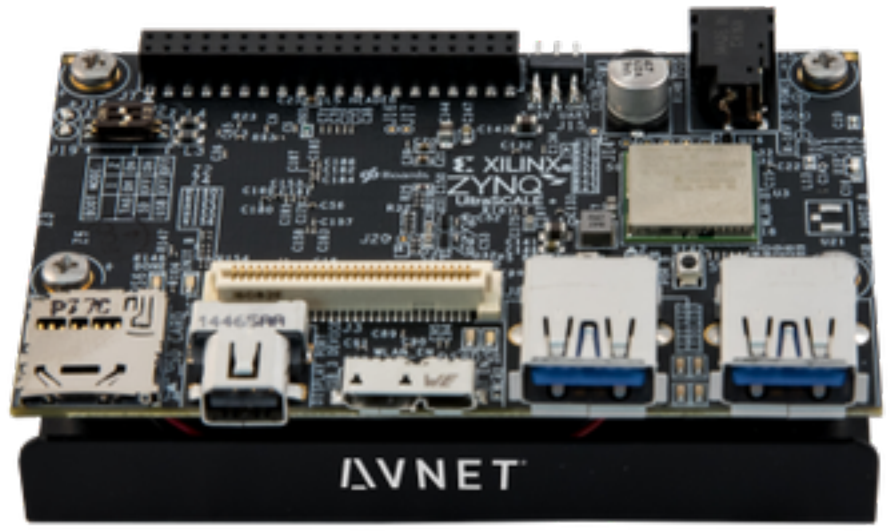


Xilinx Kria System-on-Modules

Classification by location

Edge, on premise, on cloud

- Edge Computing : move processing close to data “producers”, typically sensors, IoT, …
- On premise: dedicated resources on proprietary servers (es: FPGA accelerators)
- On cloud: dedicated resources accessible through cloud computing (e.g. Amazon AWS)


Ultra96 Avnet - Zynq

Edge Devices

Computing power
Storage
Connectivity

Cost
Latency
Data locality (privacy issues)

On premise/cloud

Persistent vs. Volatile programming

Technology dependent behaviour

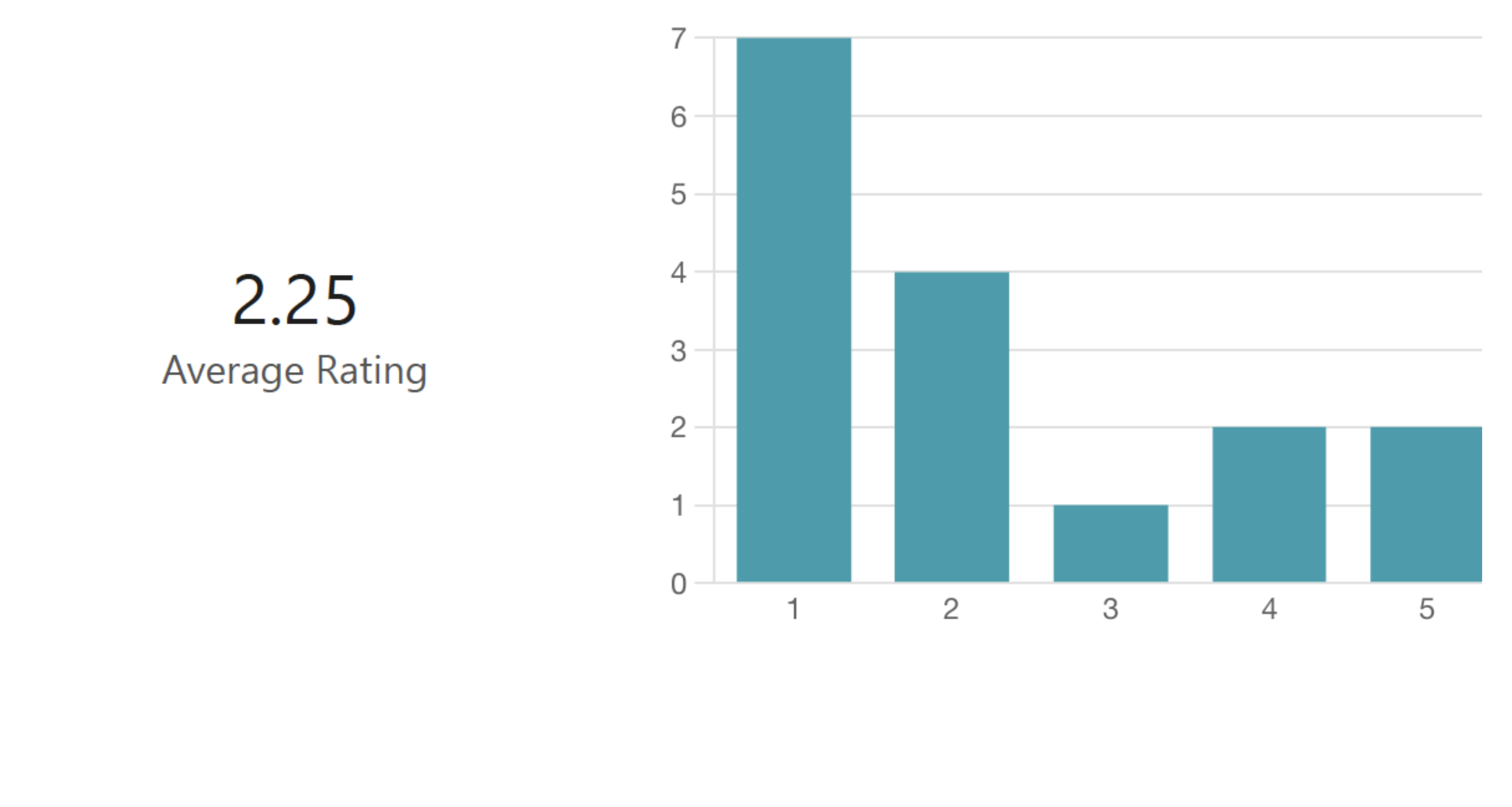
- AMD/Xilinx and Intel are SRAM-based: not programmed at the power on
  - Often there’s a companion non-volatile memory to speed-up programming the FPGA at the switch on

- Other technologies can have persistent programming - best suited for specific applications (e.g. rad-hard)
  - Microchip (formerly Actel): flash-based
  - Fuse/Antifuse (eg. Quicklogic)

Outline

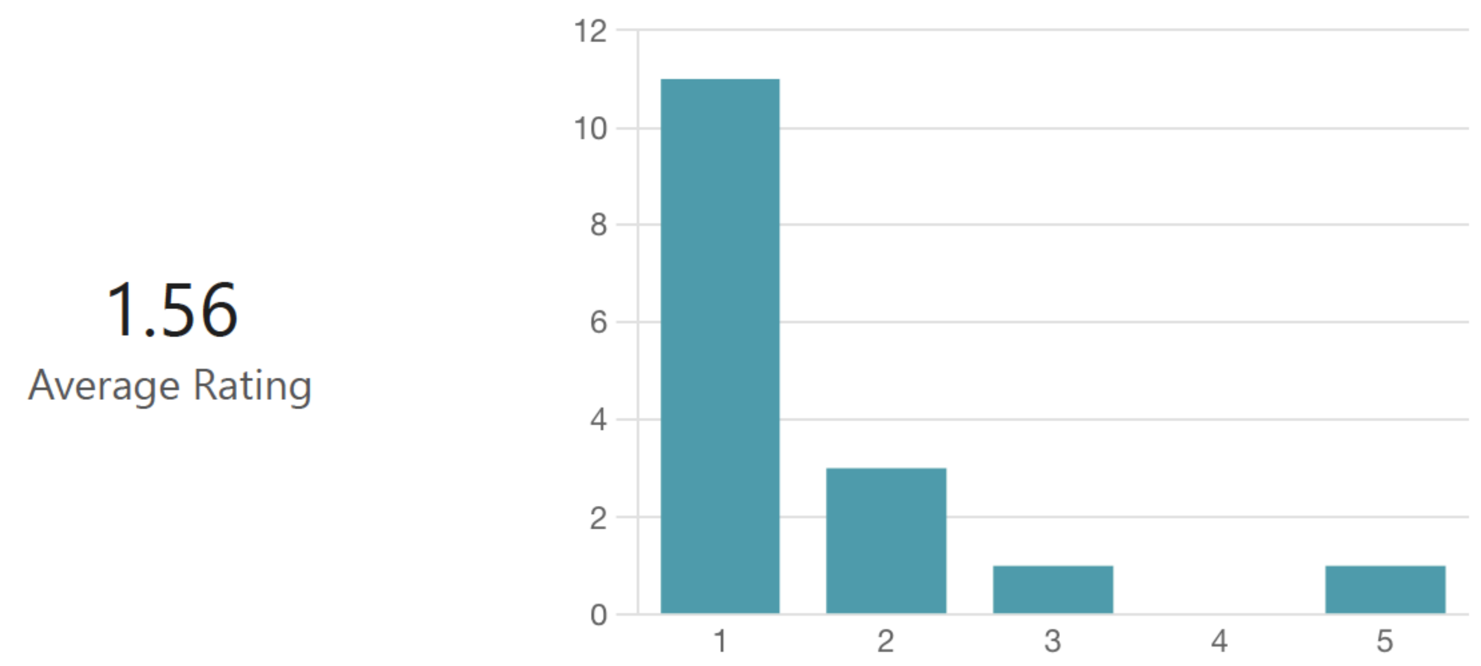
- Digital circuits: combinational, sequential and synchronous logic
- FPGA hardware - Xilinx devices as an example
- How to design

4. How much are you proficient with VHDL/Verilog (1:not at all - 5:expert)



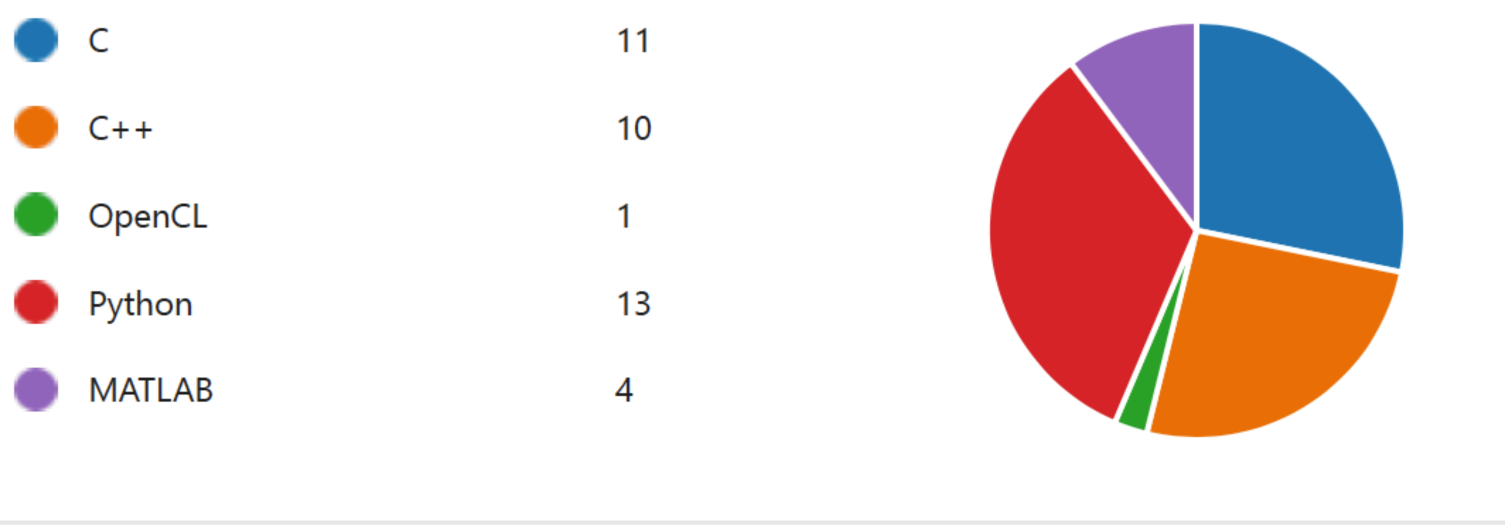
2.25
Average Rating

5. How much are you proficient with High Level Synthesys (1:not at all - 5:expert)



1.56
Average Rating

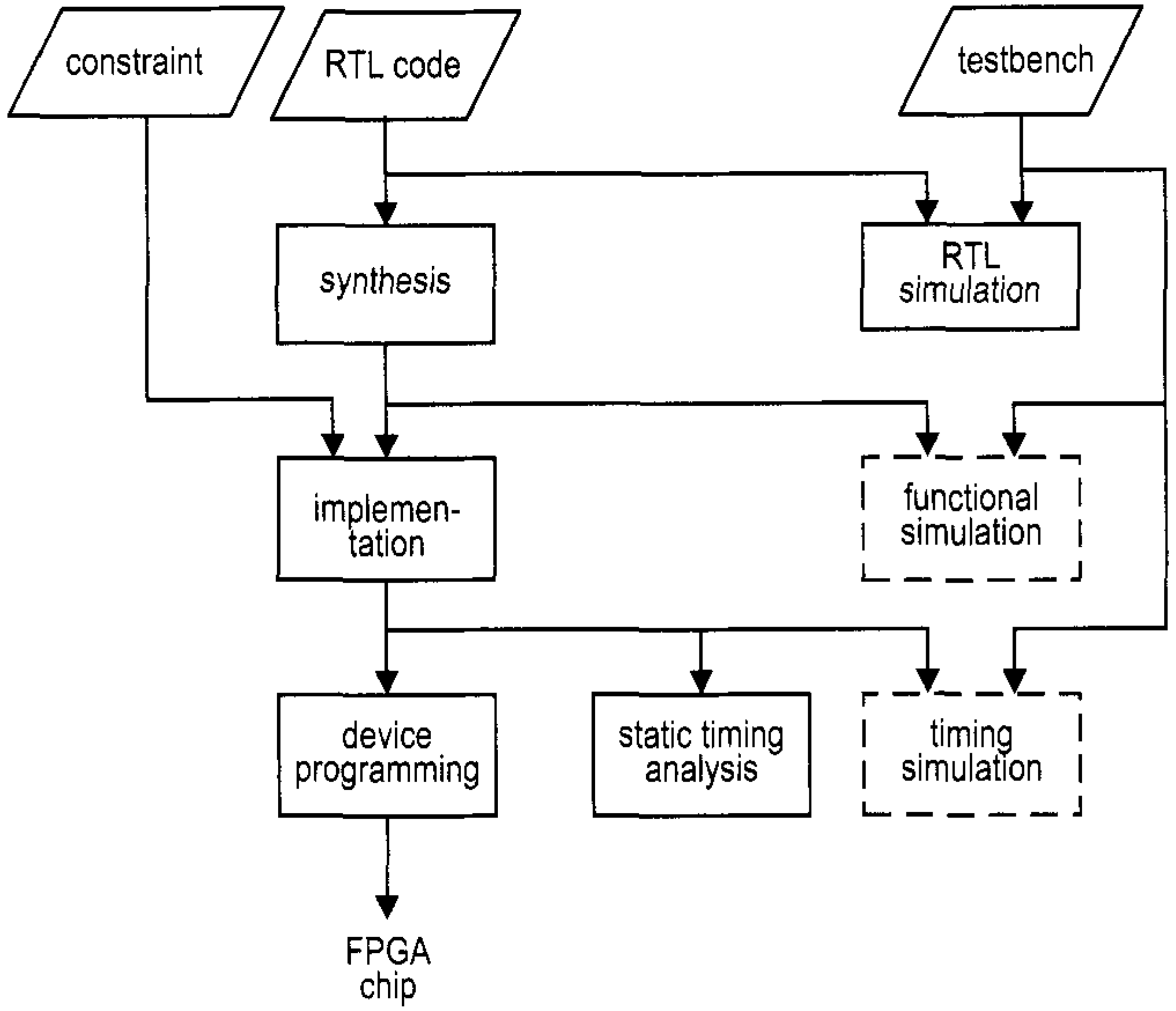
7. Which of the following programming languages you are familiar with?



●	C	11
●	C++	10
●	OpenCL	1
●	Python	13
●	MATLAB	4

Traditional design strategy

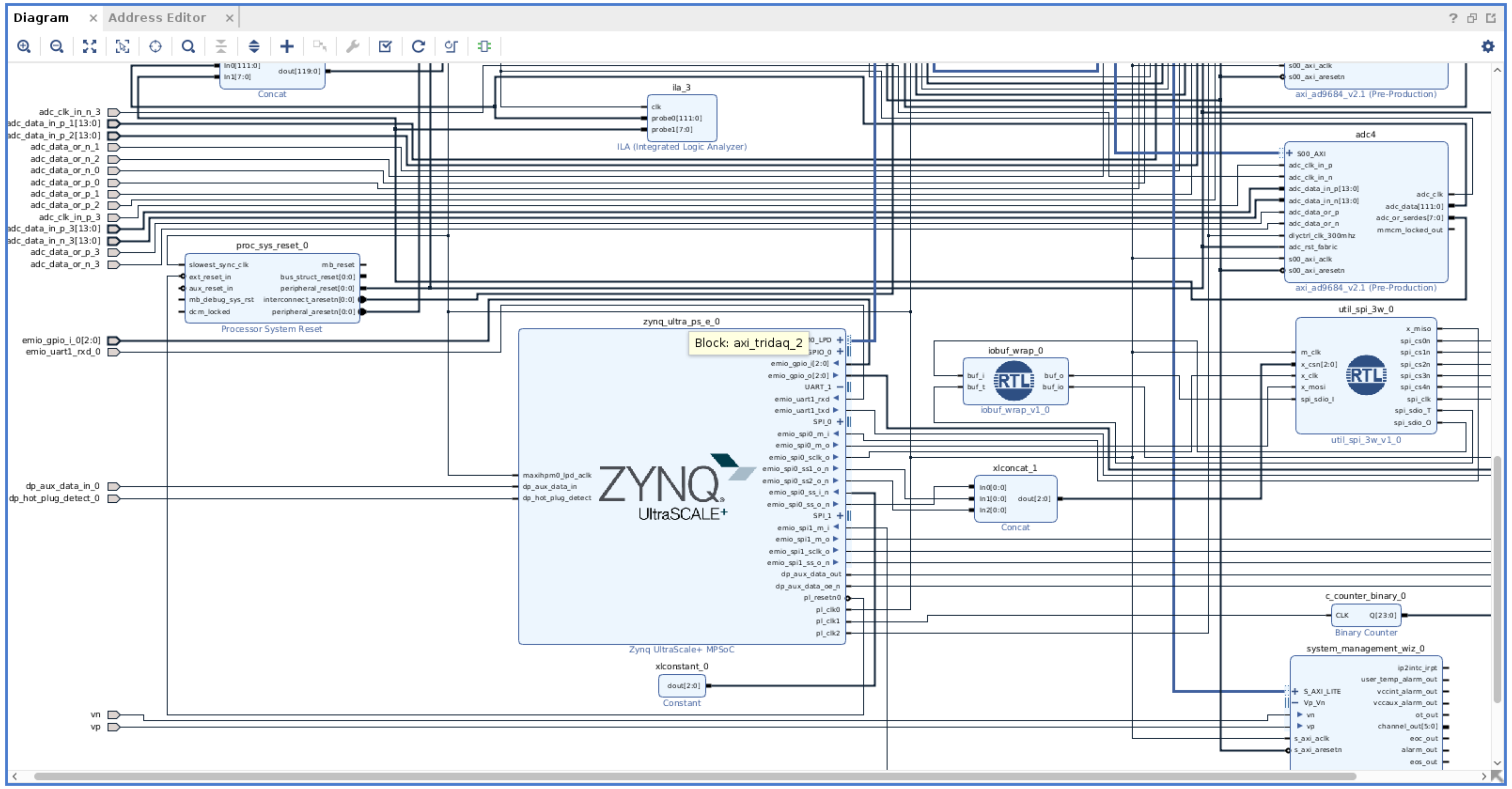
- Describe algorithms with an Hardware Description Language (VHDL, Verilog)
- Behavioural simulation
- Synthesise (convert description in basic logic elements - gates, memories, DSP, …)
- Place & Route driven by constraints
  - Place: map logic elements to the chosen FPGA
  - Route: enable/disable connections to route signal between logic elements
- Create programming file
- Load programming file to FPGA



IP cores

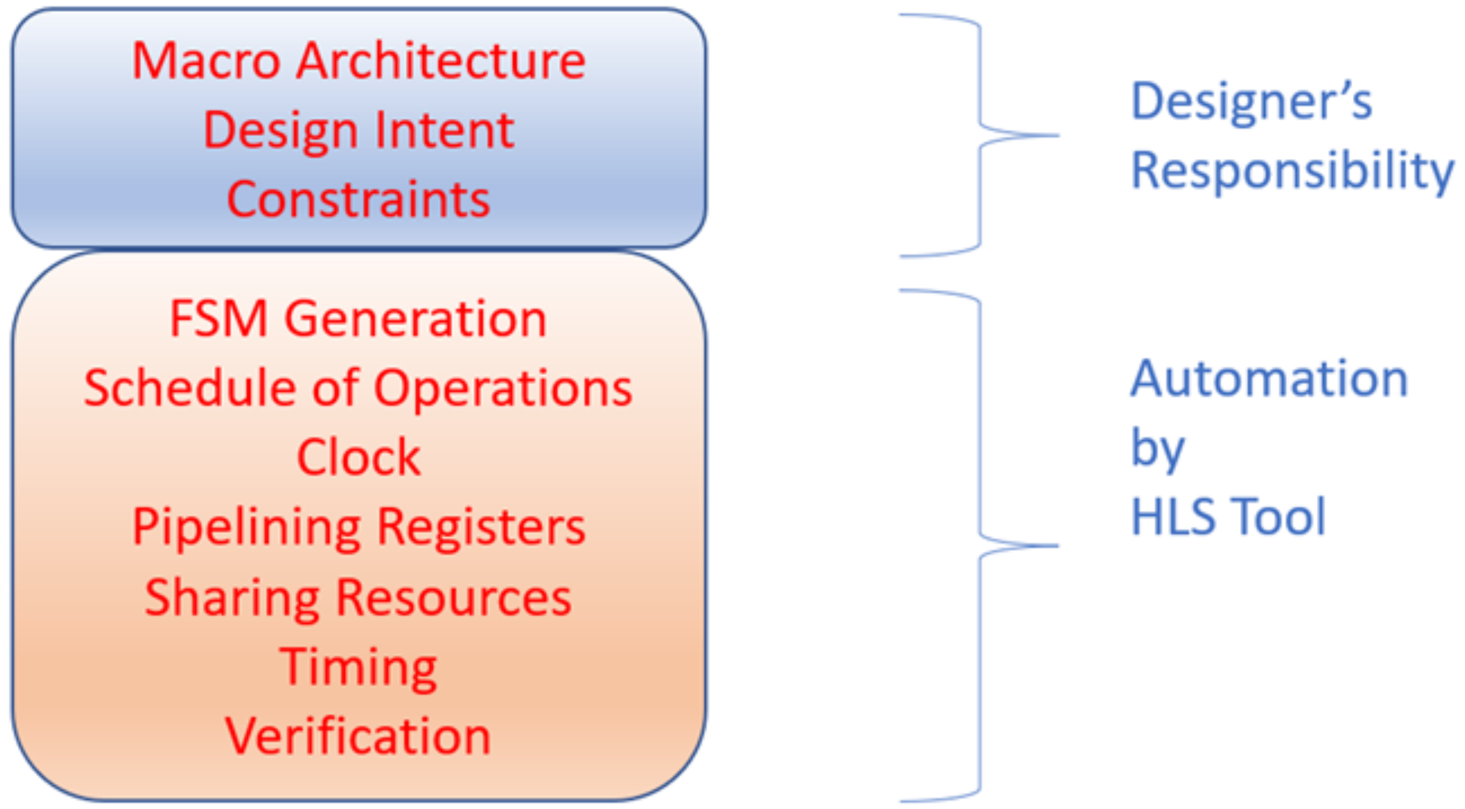
Intellectual Property

- Reusable code
- Can be used as a “black box"
- Can be encrypted
- Xilinx provides a GUI (IP integrator) as a schematic editor for IP-based designs



High level synthesis

In a glance

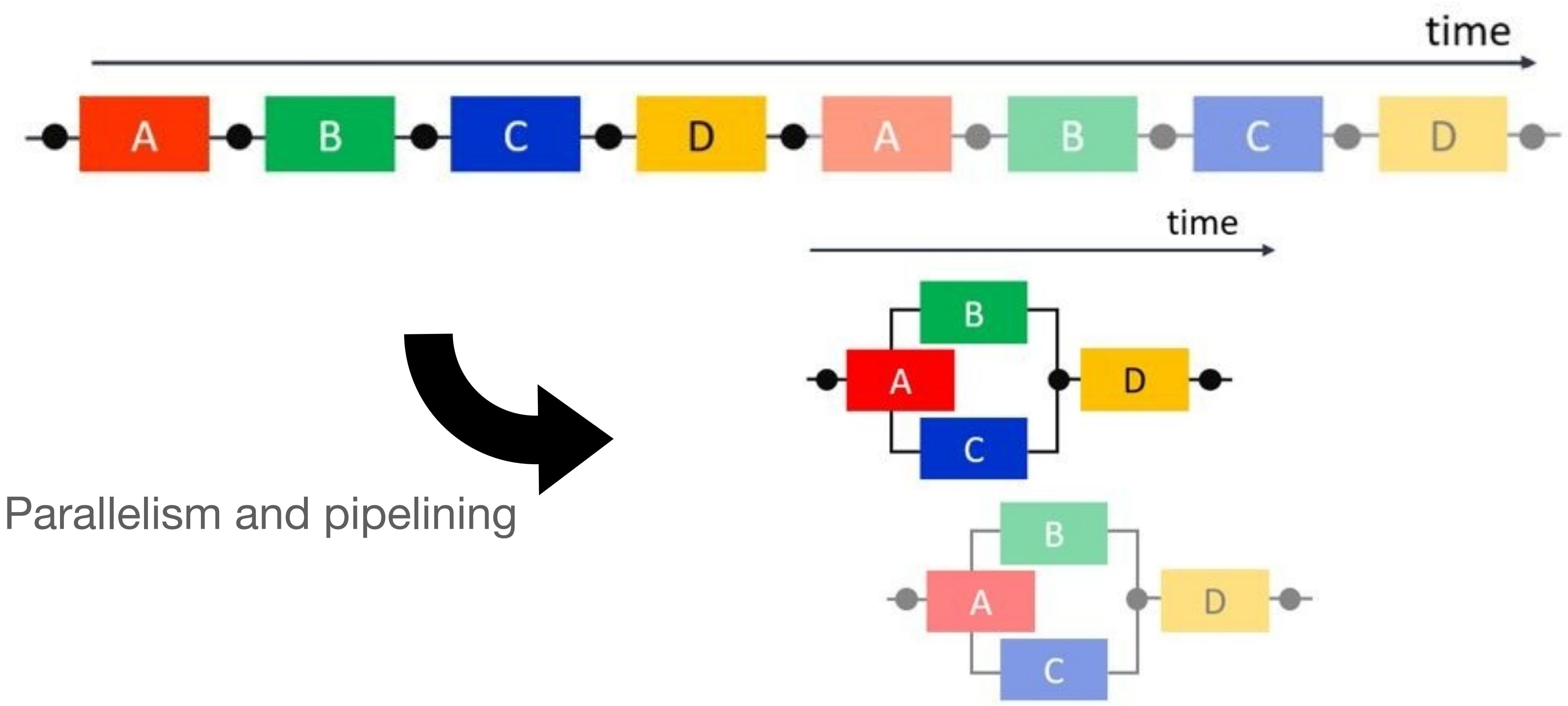


Macro Architecture
Design Intent
Constraints

Designer's Responsibility

FSM Generation
Schedule of Operations
Clock
Pipelining Registers
Sharing Resources
Timing
Verification

Automation by HLS Tool



Parallelism and pipelining

A subset of C/C++

with synthesis directives

```
vadd: for(int i = 0; i < len; i++) {
#pragma HLS PIPELINE
c[i] = a[i] + b[i];
}
```

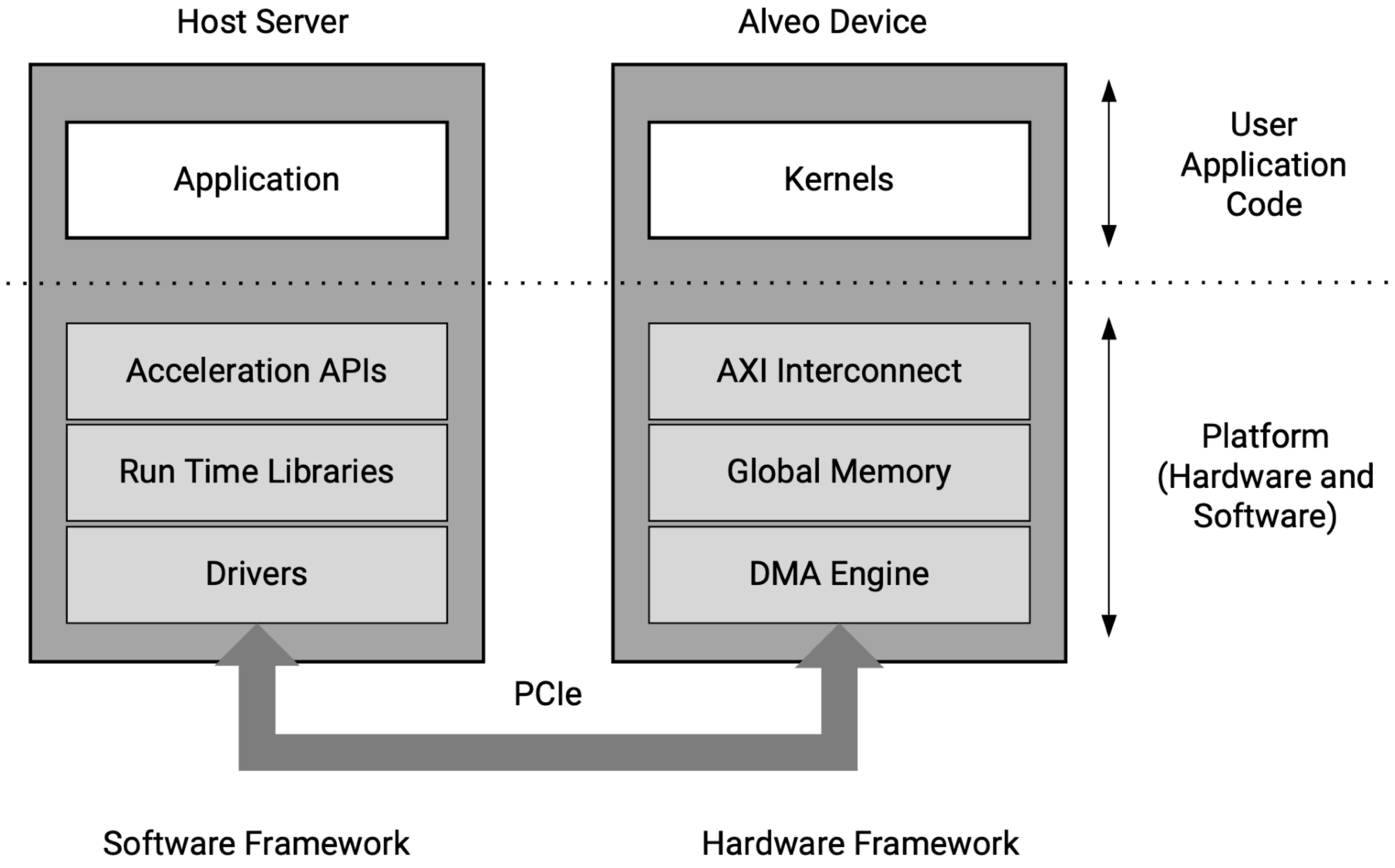
Focus on:

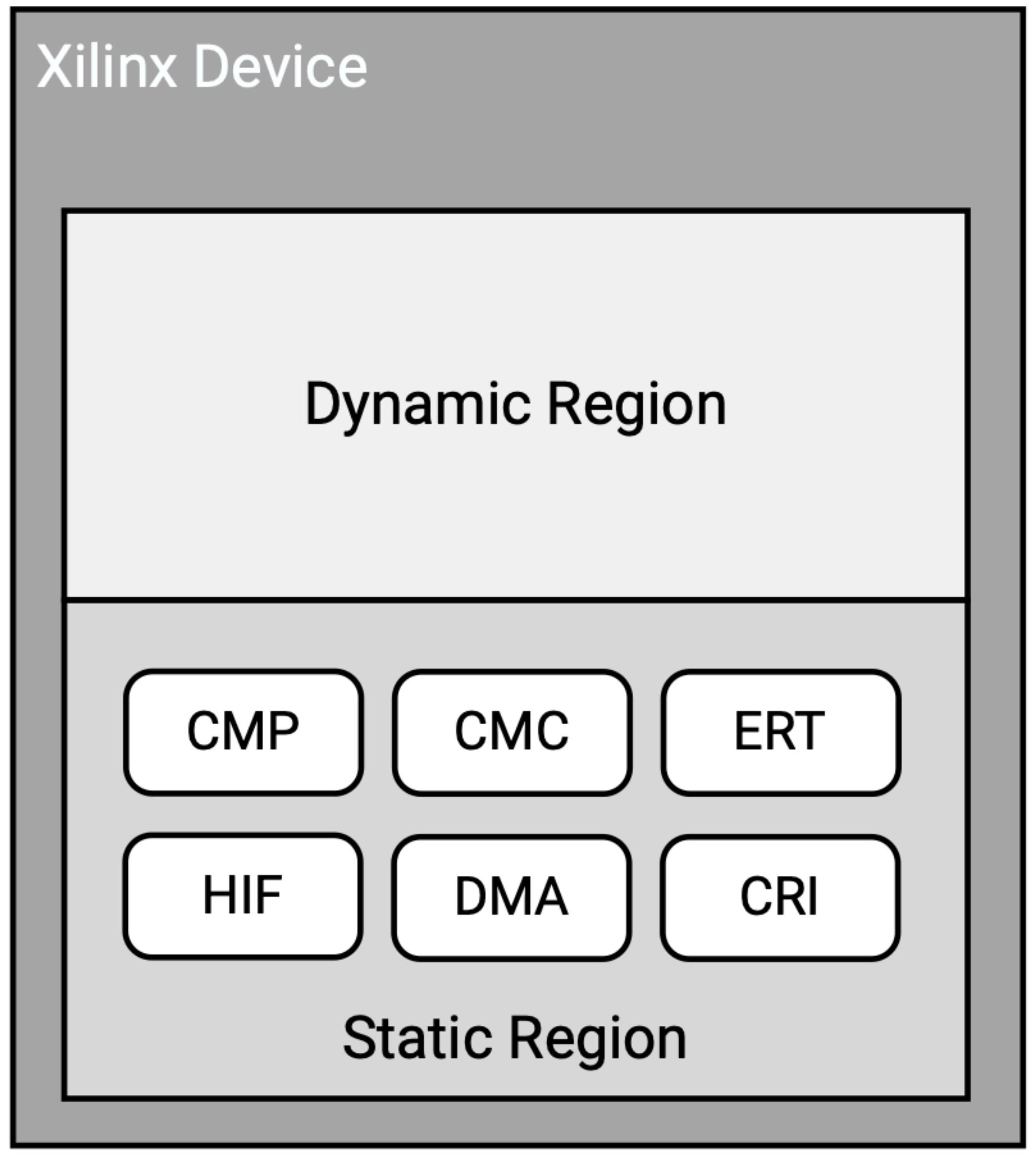
- Loops (pipelining, unrolling)
- Arrays (memories)
- Arbitrary precision data types (Integer and fixed-points)

Accelerators and platforms

Based on dynamic programming

- "For AMD Xilinx accelerator cards on premises or in the cloud, the Vitis target platform automatically configures the PCIe interfaces that connect and manage communication between your FPGA accelerators and x86 application code—you don't need to implement any connection details!"





Overlays

From PYNQ_Workshop for v2.6 - AMD/Xilinx University Program
https://github.com/Xilinx/PYNQ_Workshop

- **Overlays are generic FPGA designs that target multiple users with new design abstractions and tools**

- **Overlay characteristics**
  - Post-bitstream programmable via software APIs
  - Typically optimized for given application domains
  - Encourages the use of open source tools & fast compilation
  - Enables productivity by re-using pre-optimized designs
  - Makes benefits of FPGAs accessible to new users