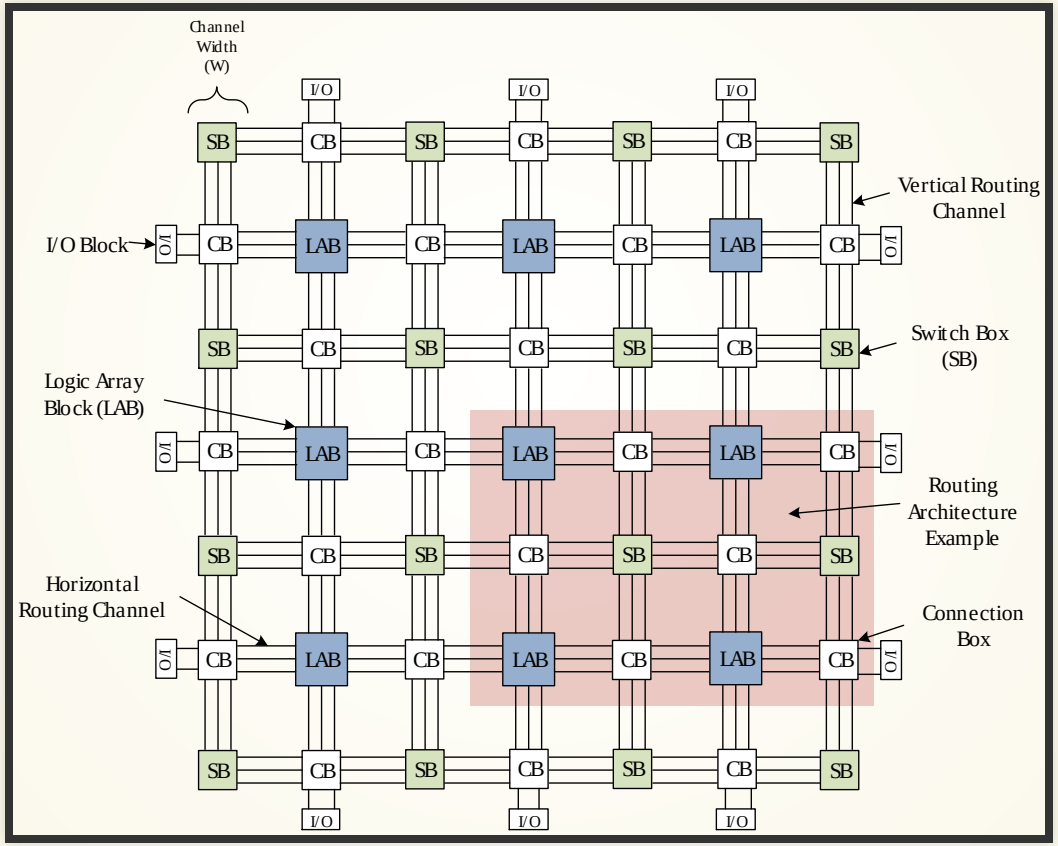
ACCELERATOR DESIGN WITH OPENCL

(ATHENS WEEK 19-24 MARCH, 2018)

FPGA ARCHITECTURE: OVERVIEW



FPGA ARCHITECTURE: BASIC LOGIC ELEMENT

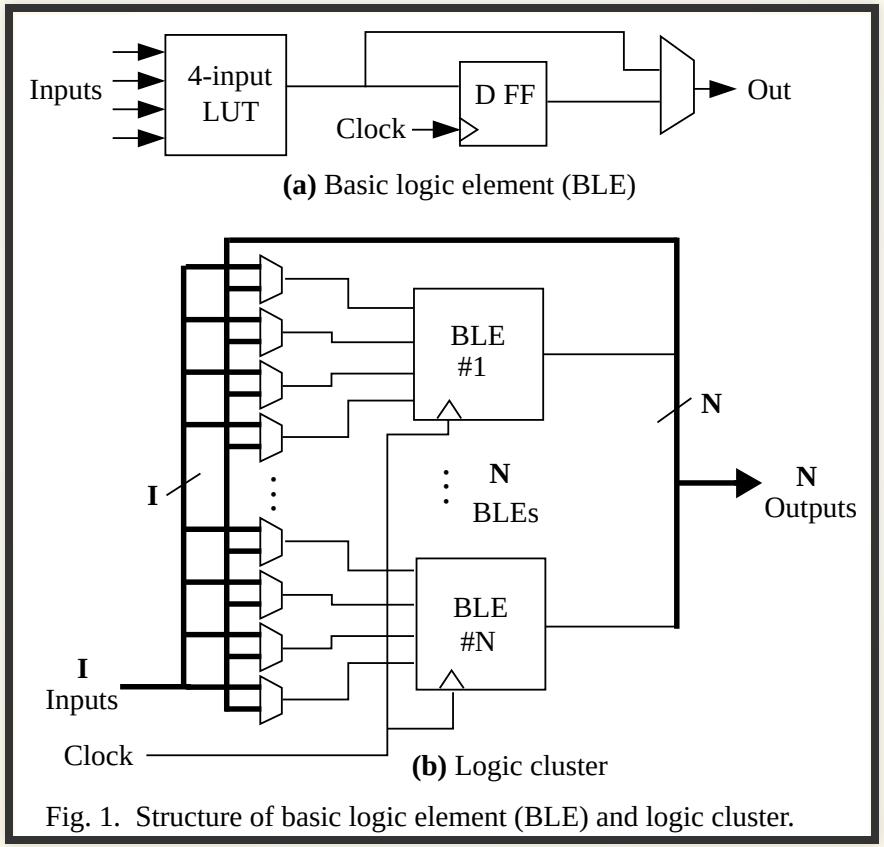


Fig. 1. Structure of basic logic element (BLE) and logic cluster.

FPGA ARCHITECTURE: CONFIGURABLE LOGIC BLOCK (CLB/LAB)

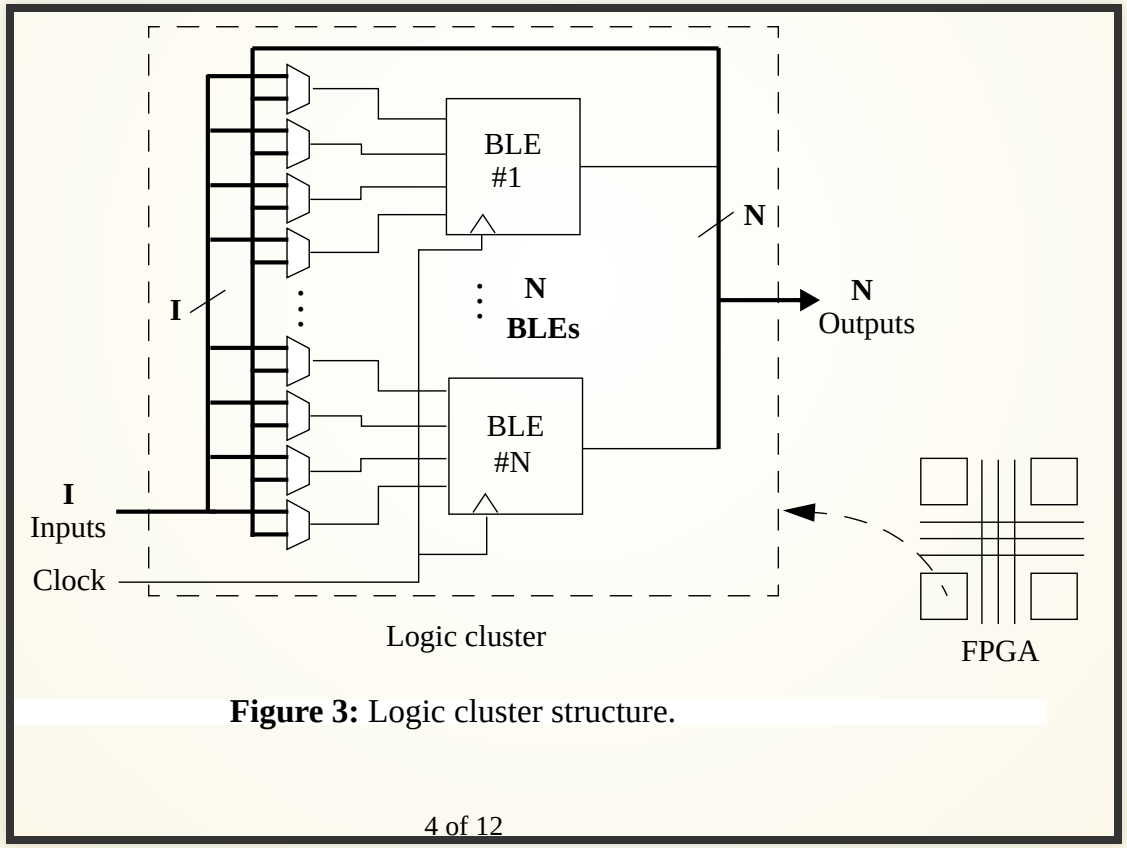
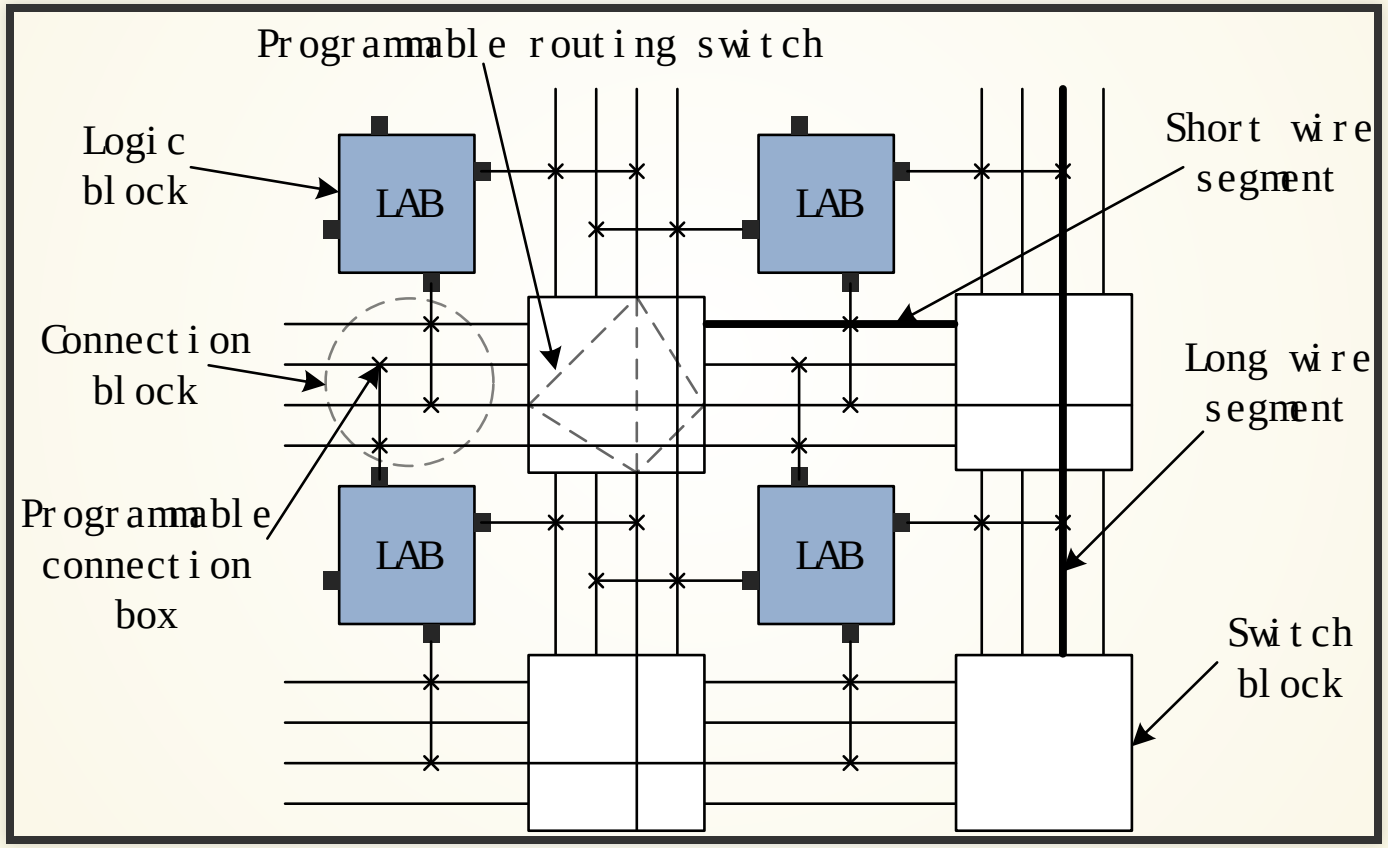
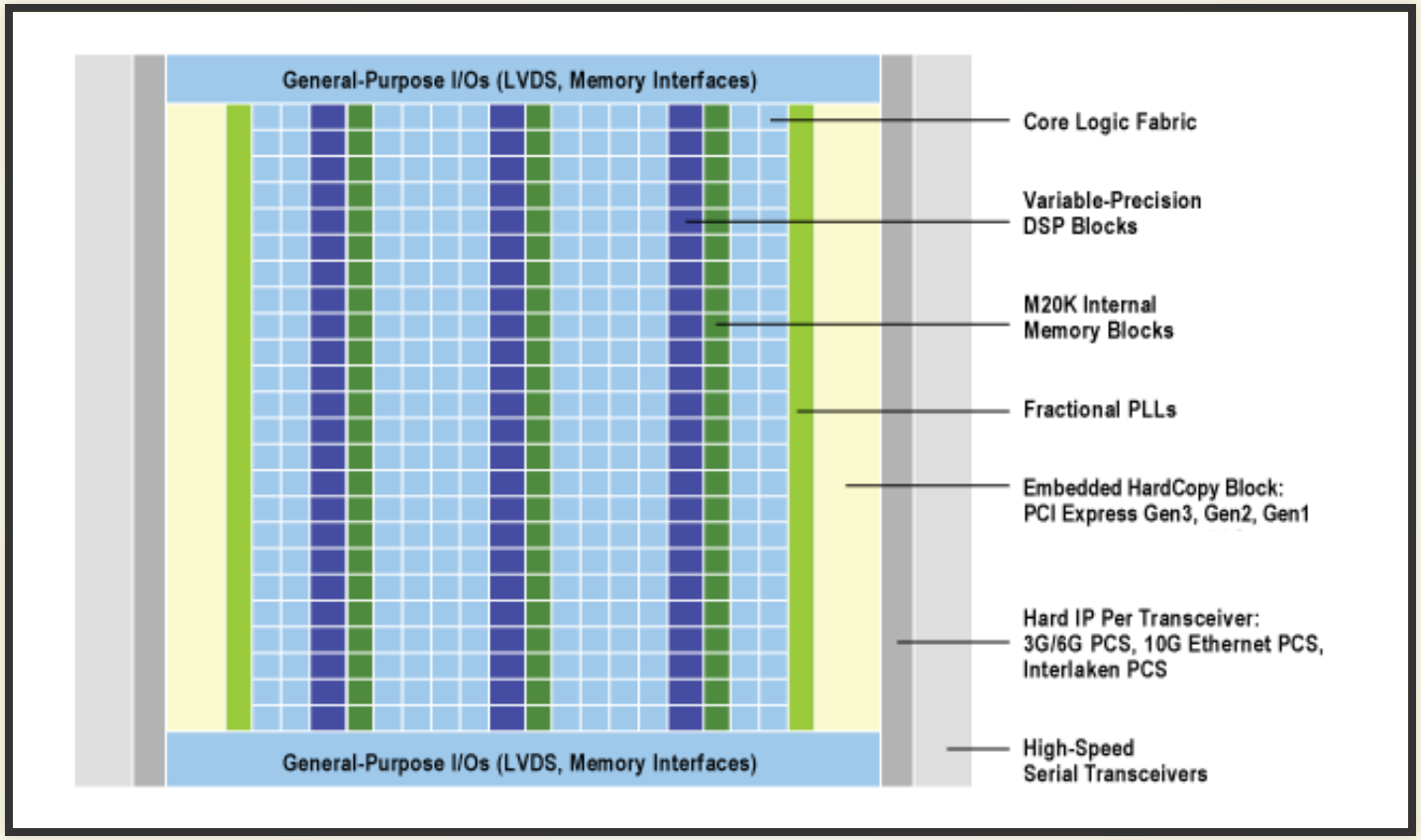


Figure 3: Logic cluster structure.

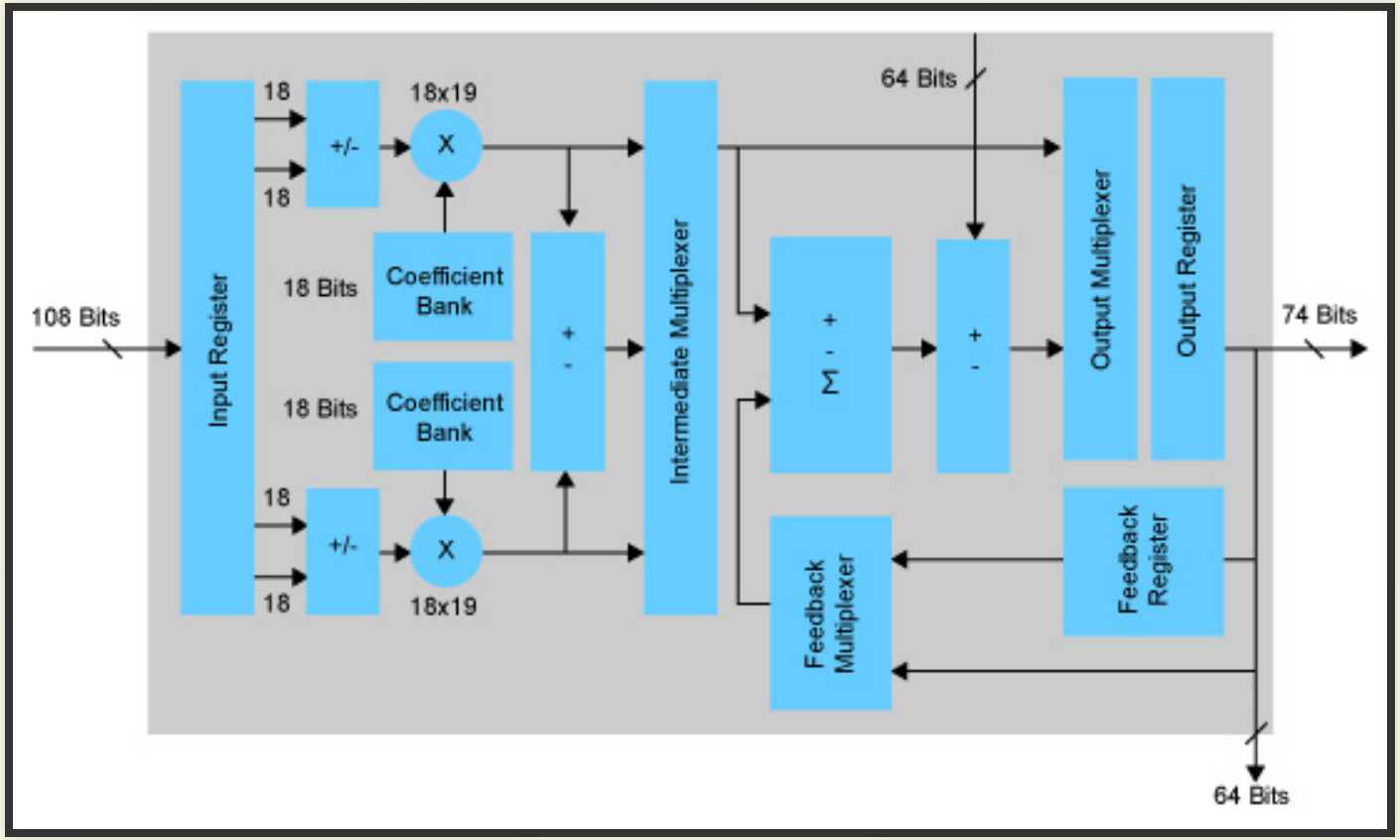
FPGA ARCHITECTURE: ROUTING



FPGA ARCHITECTURE: HARD MACROS



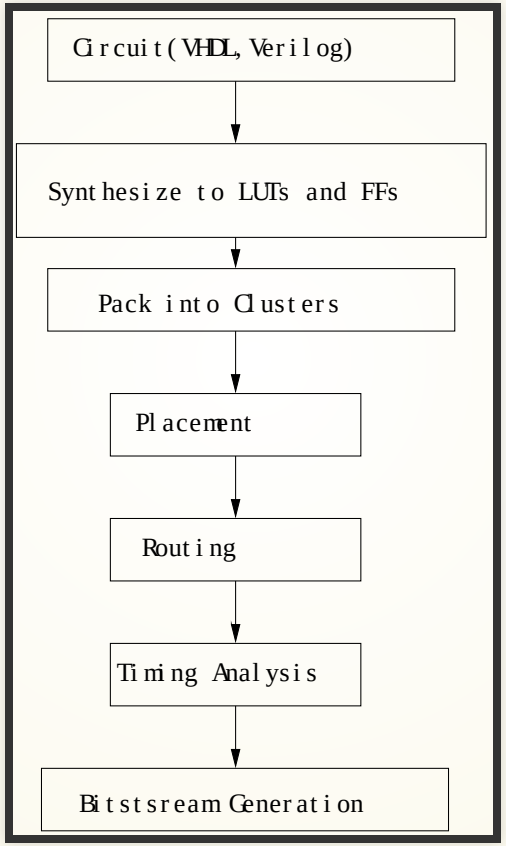
FPGA ARCHITECTURE: DSP BLOCKS



FPGA : DESIGN INPUT

- Hardware Description Language
  - VHDL
  - Verilog
- Schematic

FPGA : CAD FLOW



FPGA: APPLICATIONS

- Prototyping, ASIC Emulation
  - FPGAs can emulate all hardware
- Low/Medium volume hardware
  - Telecom Equipment
- Low Latency Real-Time Systems
  - Drones, Financial Feeds
- Military
- Accelerators

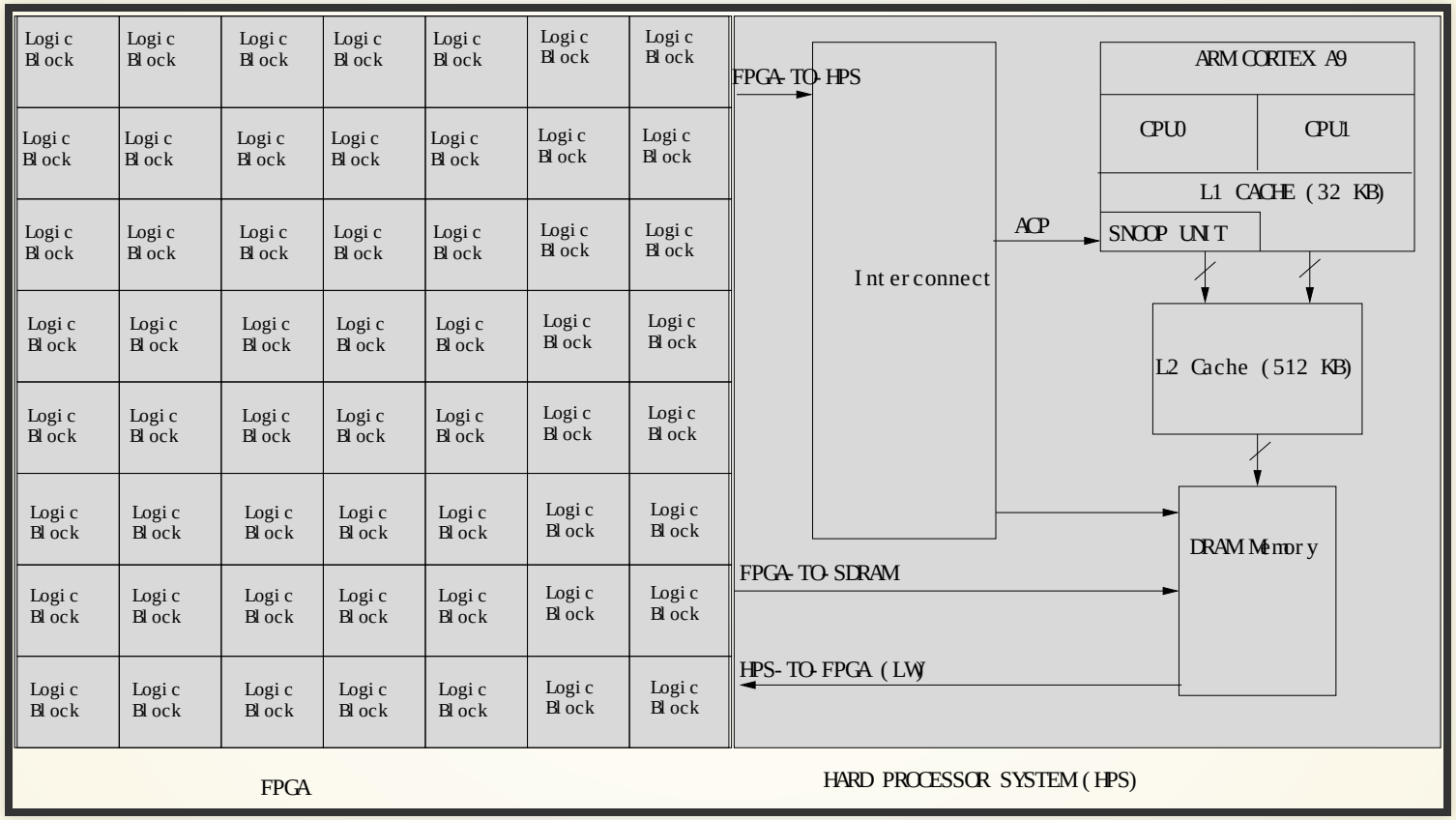
FPGAS AS ACCELERATORS

- Traditional
  - PCI-Xpress add-on cards for desktop.
- Embedded
  - SoCFPGAs
- Cloud
  - Amazon cloud F1 Instances |
  - OVH

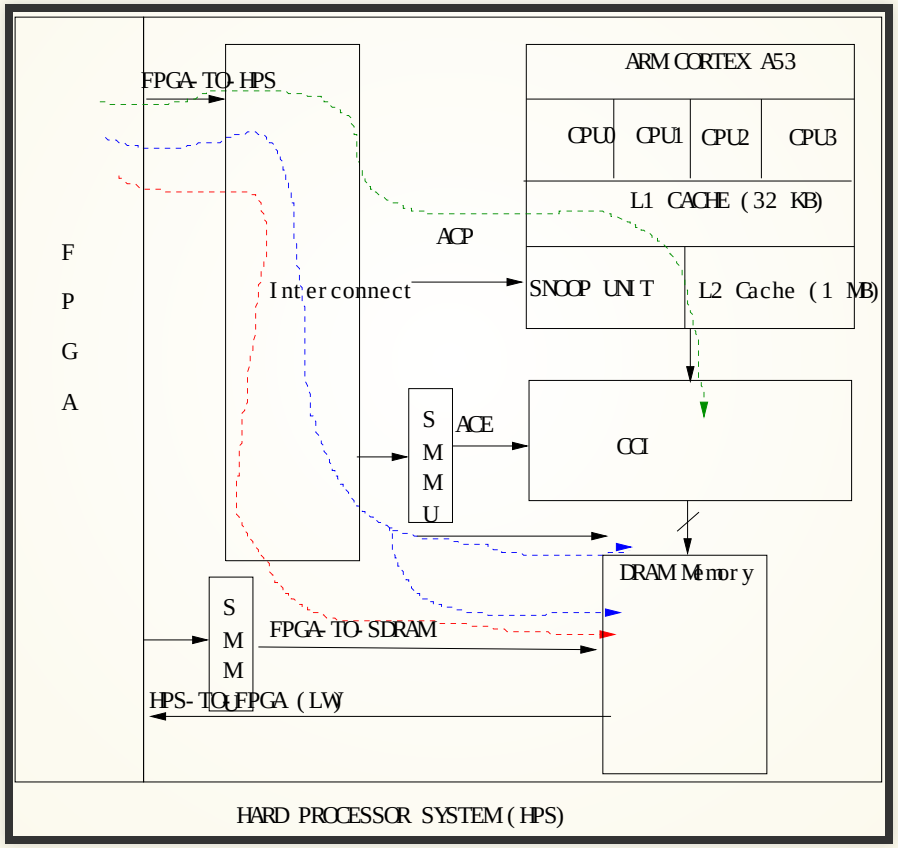
SOCFPGAS

- FPGAs on the same die as Chip Multi-Processors
  - Altera/Intel Cyclone V, Arria V, Stratix 10
  - Xilix Zynq, Zynq Ultrascale
- OpenCL can be use to program both PCI-X cards and SOcFPGAs.

SOCFPGA ARCHITECTURE: ALTERA CYCLONE V



SOCFPGA ARCHITECTURE: XILINX ULTRASCALE



FPGA ARCHITECTURE: EXAMPLE

- PCI-X based card Stratix 10
  - 5,510,000 BLEs (equivalent)
  - 7,470,420 Registers
  - 3960 DSP Slices
  - 170 MB Memory Blocks
  - 7.9 TMACs/s
  - 3.2 TFlops/s

FPGA ARCHITECTURE: EXAMPLE

- SoCFPGA Ultrascale
  - 600,000 BLEs (equivalent)
  - 32.1 MB memory
  - 2520 DSP Slice
  - Quad Core ARM Cortex-A53
  - Dual Core ARM Cortex R5
  - ARM Mali 400 MP2

OPENCL FOR FPGAS

- each loopnest is transformed to a hardware pipeline.
- Extensions: FIFO stream.

OPENCL FOR FPGAS: STREAMS

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,10)	(0,11)	(0,12)	(0,13)	(0,14)	(0,15)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,10)	(1,11)	(1,12)	(1,13)	(1,14)	(1,15)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)	(2,10)	(2,11)	(2,12)	(2,13)	(2,14)	(2,15)

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,10)	(0,11)	(0,12)	(0,13)	(0,14)	(0,15)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,10)	(1,11)	(1,12)	(1,13)	(1,14)	(1,15)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)	(2,10)	(2,11)	(2,12)	(2,13)	(2,14)	(2,15)

(0,0)	(0,1)	(0,2)
(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,10)	(0,11)	(0,12)	(0,13)	(0,14)	(0,15)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,10)	(1,11)	(1,12)	(1,13)	(1,14)	(1,15)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)	(2,10)	(2,11)	(2,12)	(2,13)	(2,14)	(2,15)

(0,0)	(0,1)	(0,2)
(1,0)	(1,1)	(1,2)
(2,0)	(2,1)	(2,2)

(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,9)	(0,11)	(0,12)	(0,13)	(0,14)	(0,15)	(1,0)
(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)	(1,11)	(1,12)	(1,13)	(1,14)	(1,15)	(2,0)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)	(2,9)	(2,11)	(2,12)	(2,13)	(2,14)	(2,15)	(3,0)

(0,1)	(0,2)	(0,3)
(0,1)	(0,2)	(0,3)
(2,1)	(2,2)	(2,3)

OPENCL FOR FPGAS: LAB WORK

- setup
  - git pull
  - source init.sh
  - module load altera/17.0
  - module load altera/s5_ref

OPENCL FOR FPGAS: LAB WORK

- Go to FPGA/vector_add
  - compile for emulation
    - make cl_compile
    - make all
    - make sim
  - build
    - make build
- Check reports