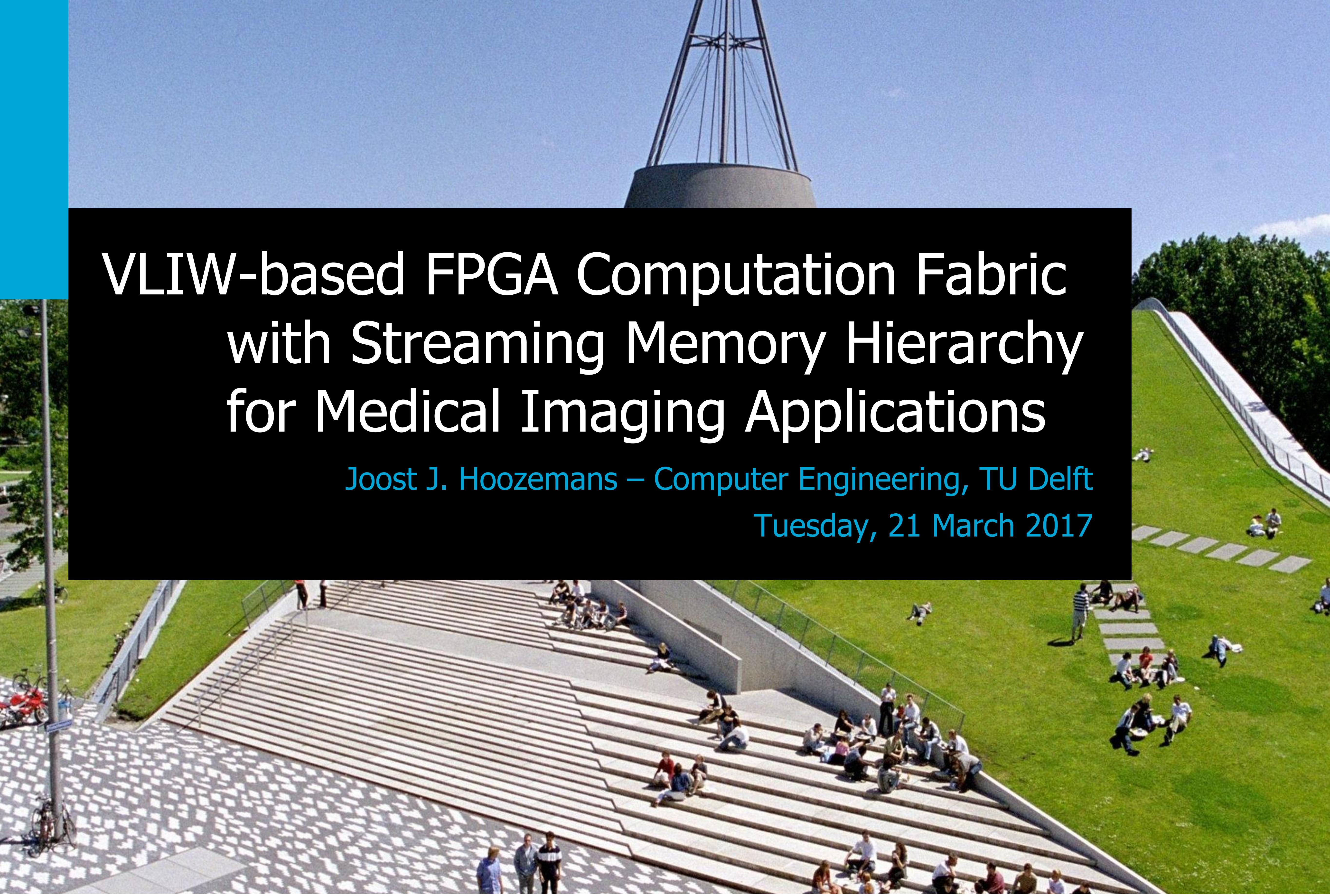


VLIW-based FPGA Computation Fabric with Streaming Memory Hierarchy for Medical Imaging Applications

Joost J. Hoozemans – Computer Engineering, TU Delft

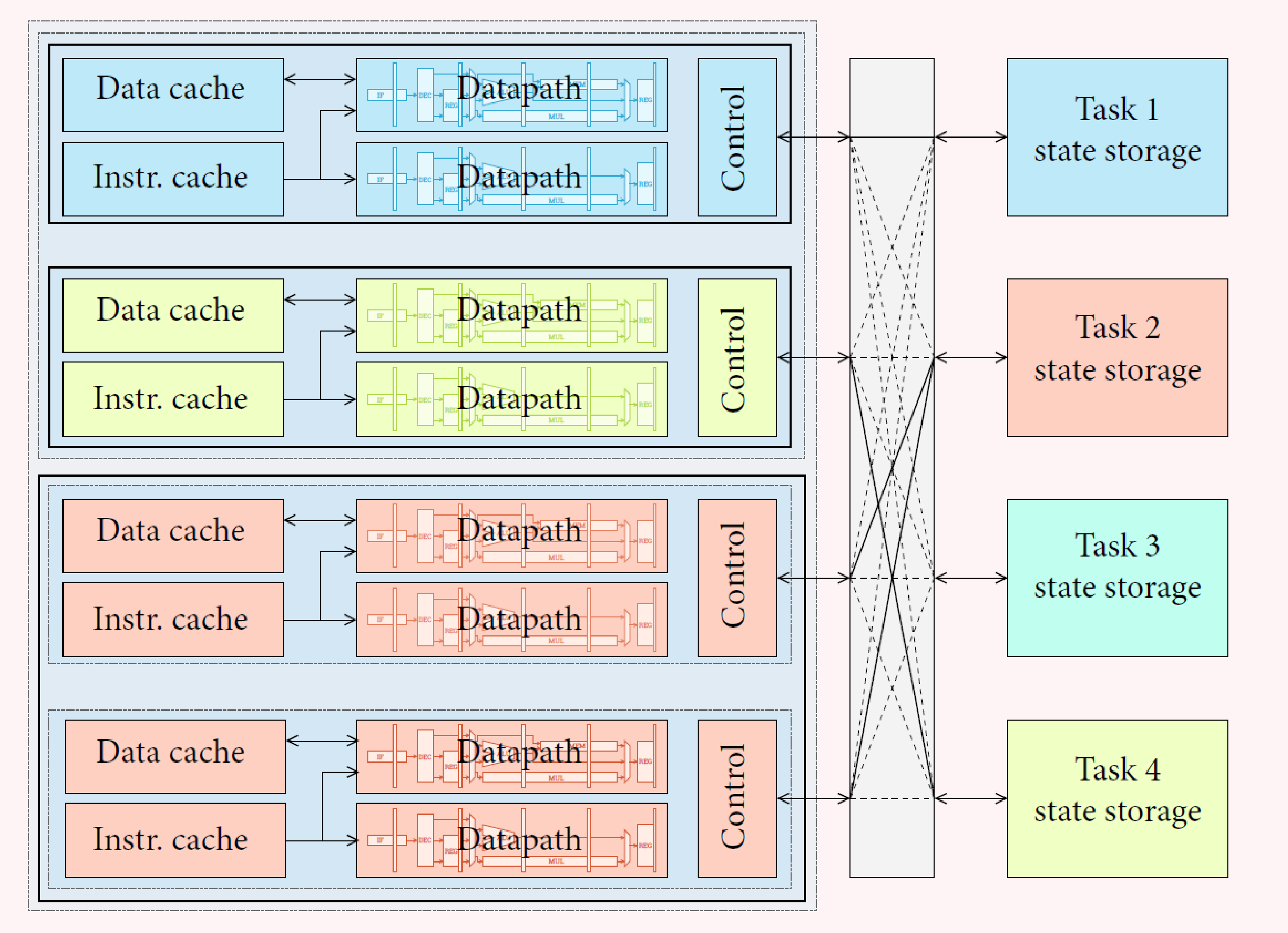
Tuesday, 21 March 2017

TU Delft

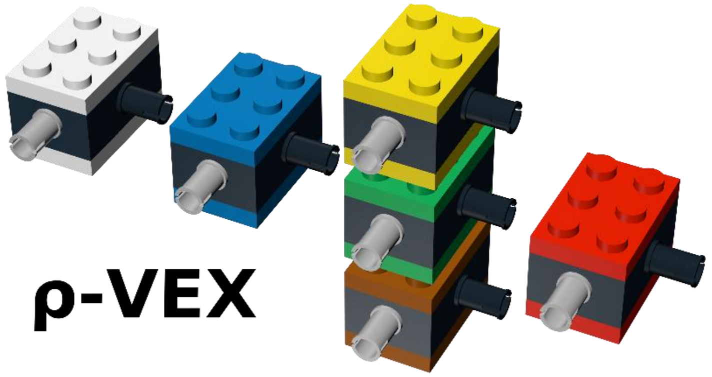
CE Lab Computer Engineering Laboratory

ρ-VEX

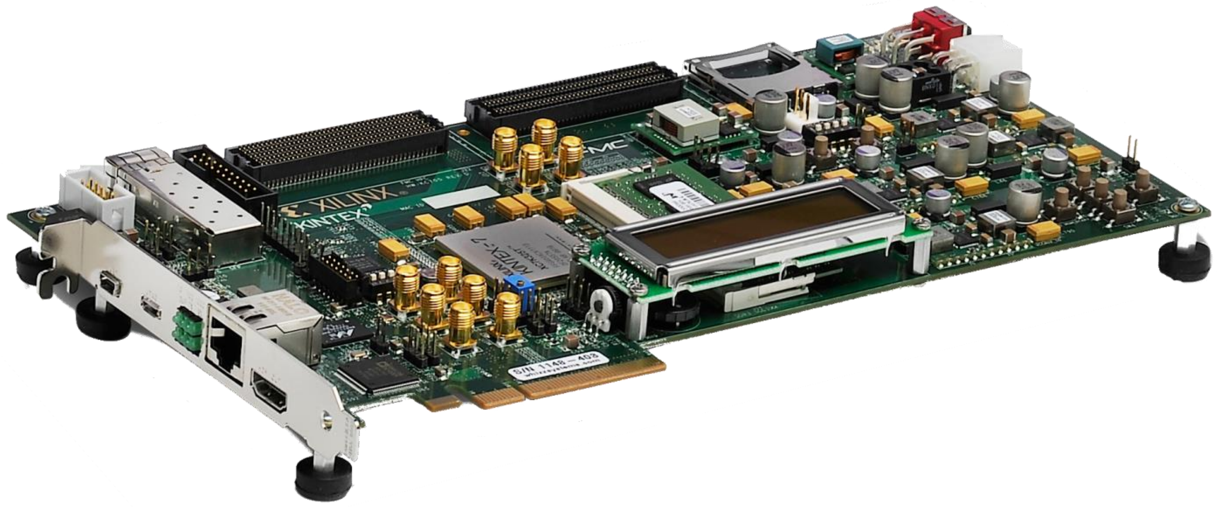
ρ-VEX Overview



ρ-VEX Overview



- Dynamically reconfigurable VLIW processor
- Adapt processor to workload
- Assign resources (datapaths) to threads
- Developed at TU Delft
- Prototyped on FPGA (softcore)



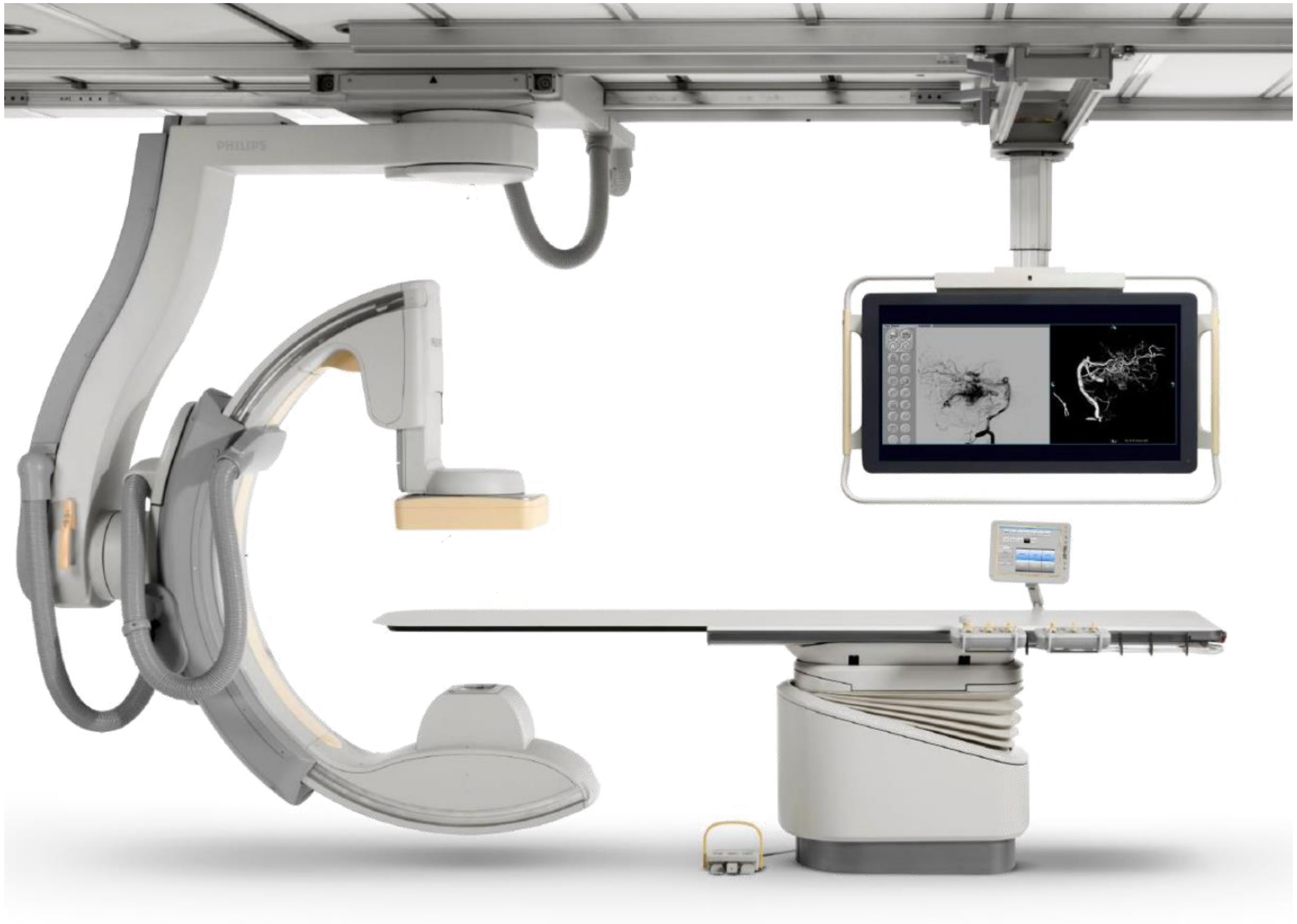
EU Project ALMARVI consortium

Almarvi

TU Delft

PHILIPS
Healthcare

Medical Imaging

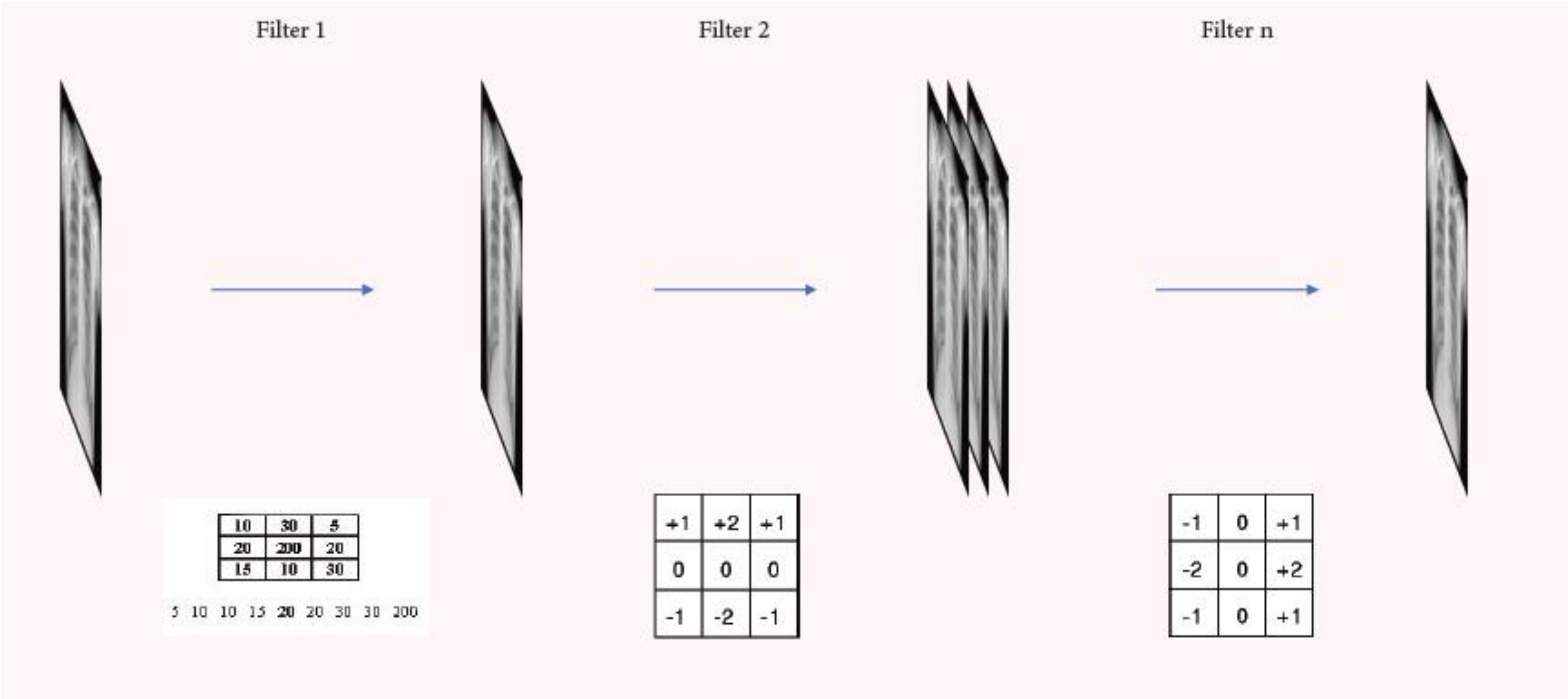


PHILIPS
Healthcare

Medical Imaging

- Real-time feedback (continuous radation)
- Reduce radiation doses
- Keep/improve image quality
- Limit latency
- Maintainability ~15 years

Medical Imaging



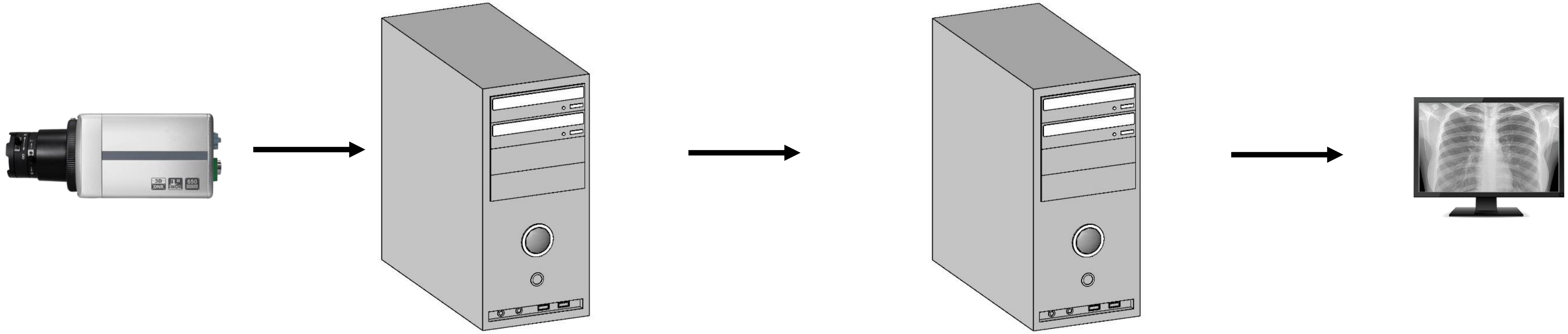
10	30	5
20	200	20
15	10	30

5 10 10 15 20 20 30 30 200

+1	+2	+1
0	0	0
-1	-2	-1

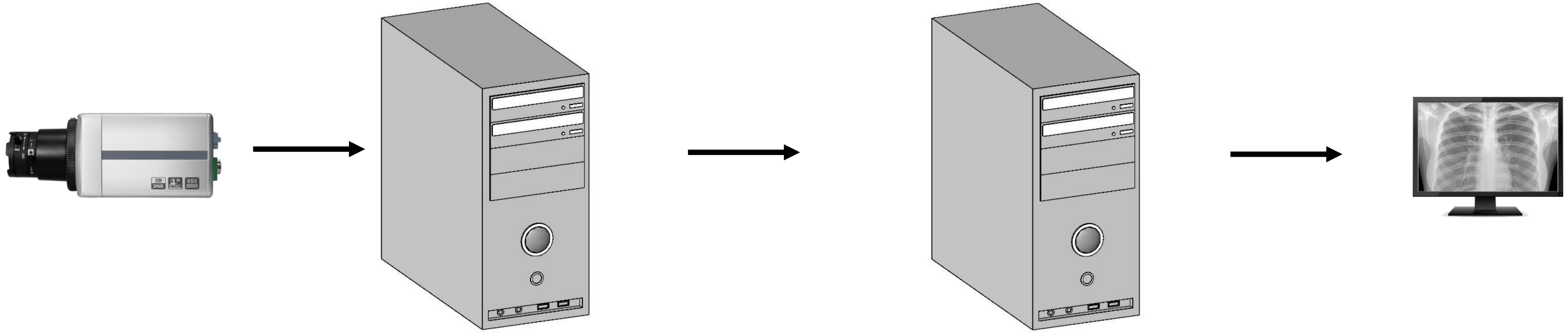
-1	0	+1
-2	0	+2
-1	0	+1

Medical Imaging



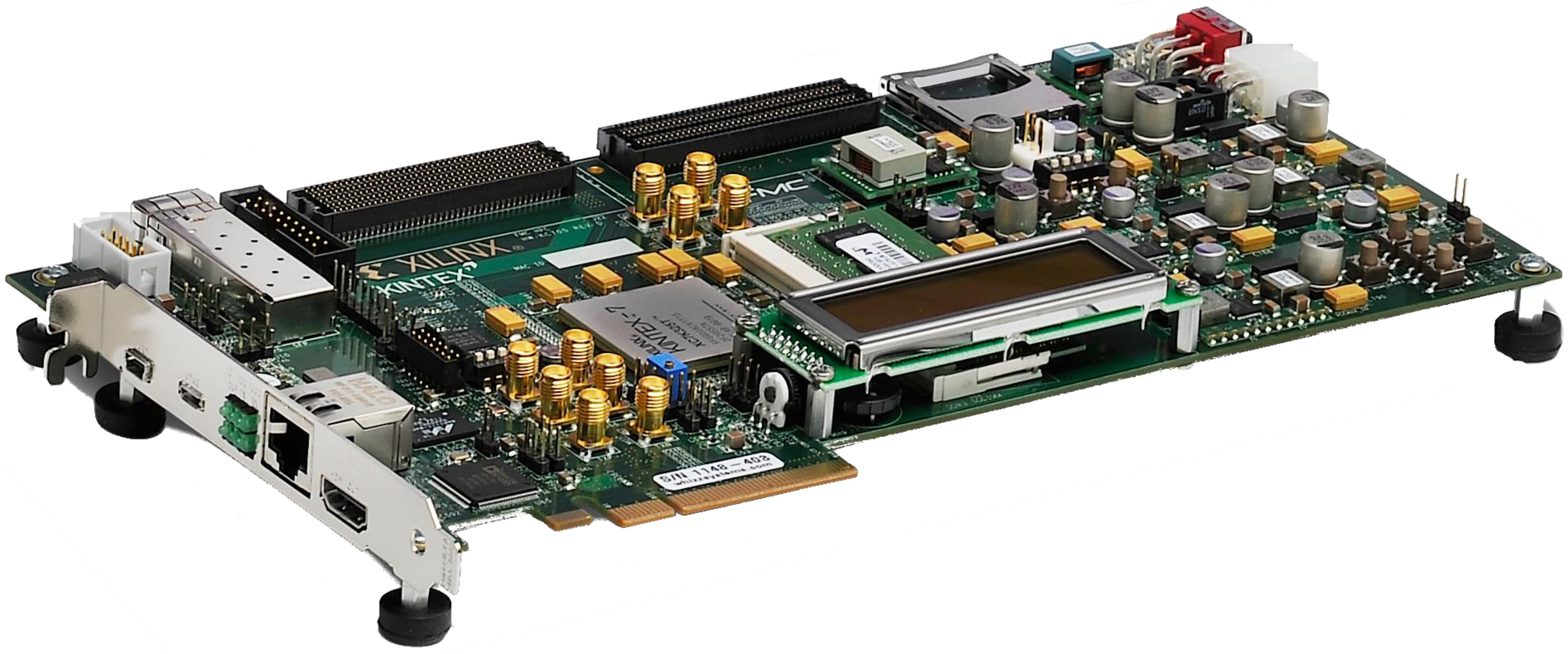
Medical Imaging

- Network latency between machines
- Hardware availability too short
- Multiple machines -> difficult to debug



Acceleration

- Possibility: FPGA
- Large amounts of resources (parallellism!)
- Long availability



FPGA

FPGA design:

- VHDL (hand-written)
- High-Level Synthesis (HLS) – generated from C code

- Synthesis time-consuming (hours)

FPGA

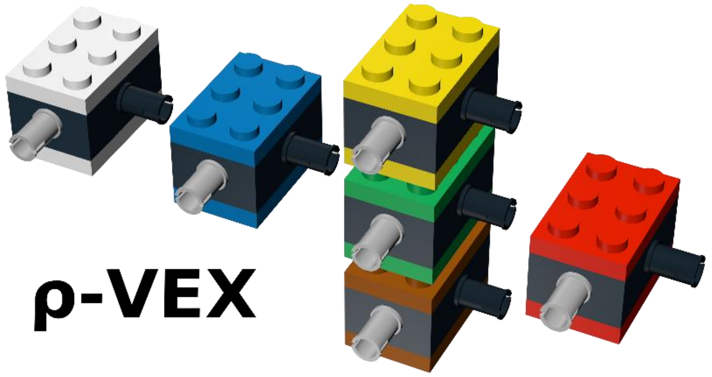
FPGA design:

- VHDL (hand-written)
- High-Level Synthesis (HLS) – generated from C code

- Synthesis time-consuming (hours)

- Alternative: Softcore
  - Processor running on FPGA
  - Synthesize once
  - Compiling, running, debugging as normal

FPGA



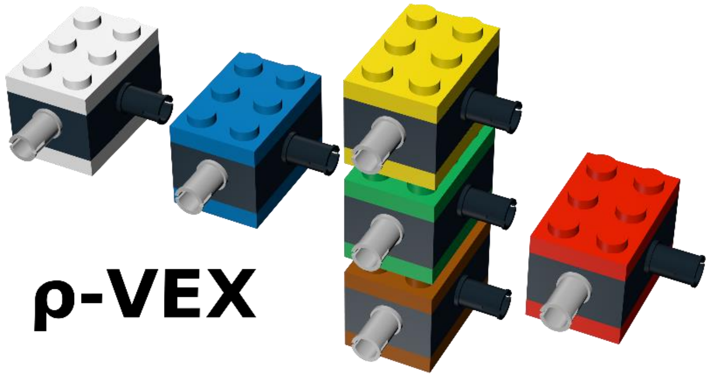
FPGA design:

- VHDL (hand-written)
- High-Level Synthesis (HLS) – generated from C code

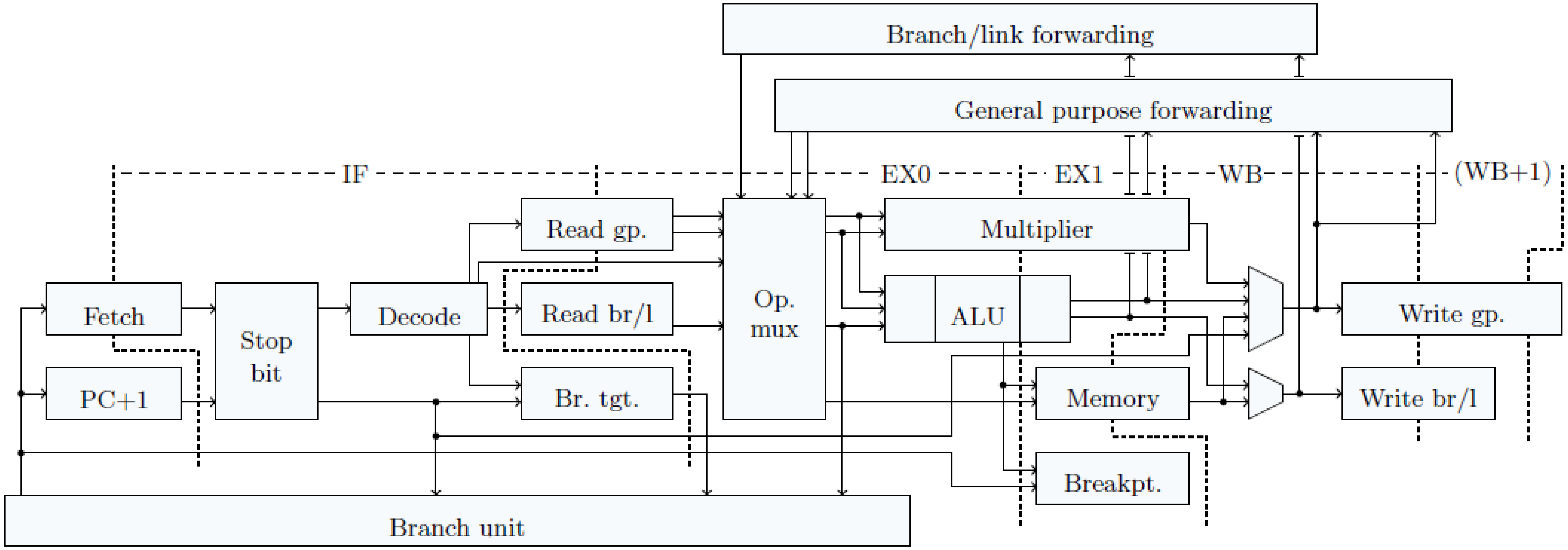
- Synthesis time-consuming (hours)

- Alternative: Softcore
  - Processor running on FPGA
  - Synthesize once
  - Compiling, running, debugging as normal

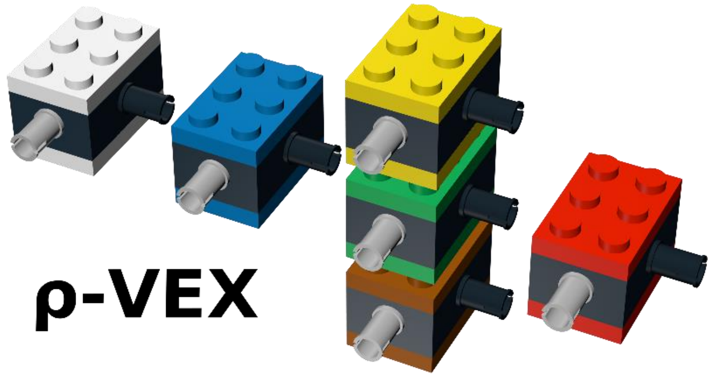
Softcore processor: ρ-VEX



- Written in VHDL
- Highly generic (*also* design-time)



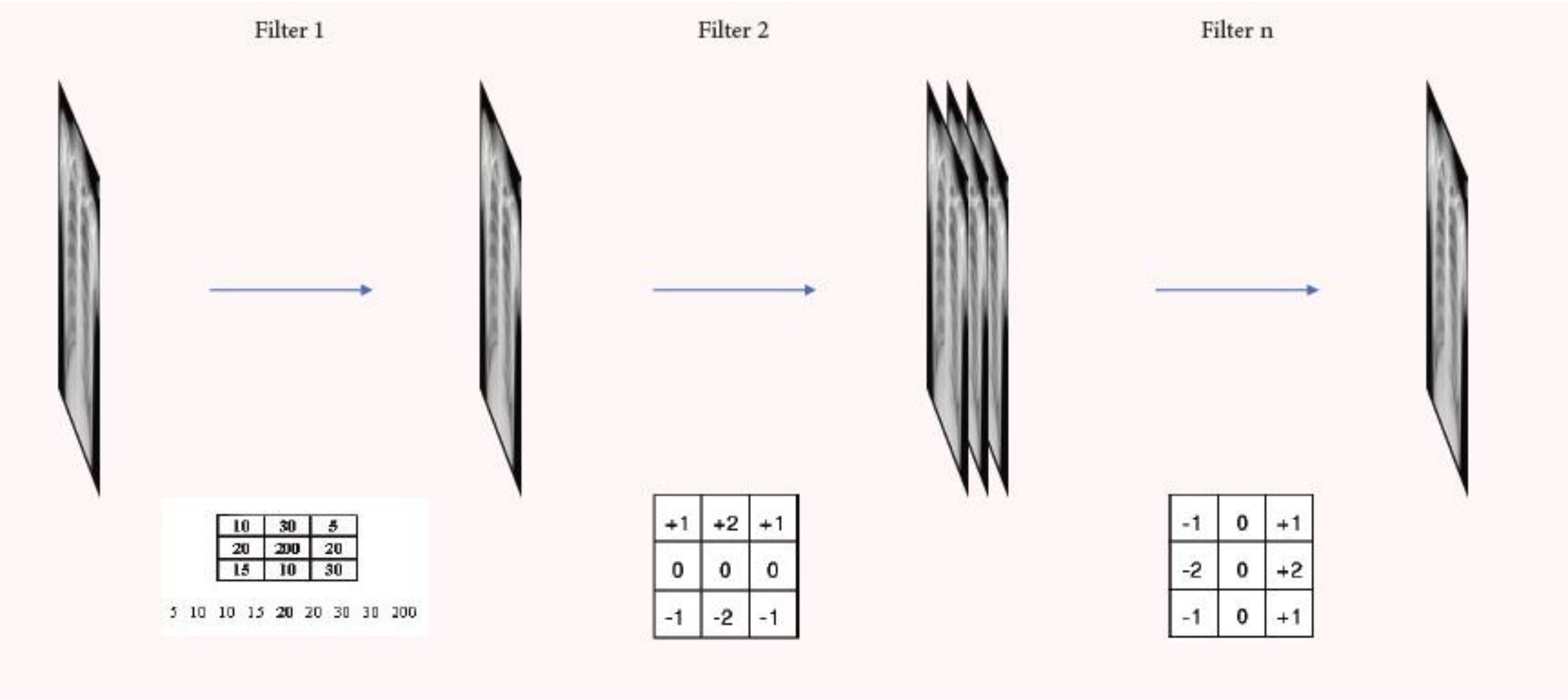
Softcore processor: ρ-VEX



- Workload characteristics:
- Image processing
- Lots of parallellism
  - ILP
  - DLP
- Streaming
- Aims: small area footprint, high clock frequency
  - 2-issue VLIW
  - Disable forwarding (decreases area, increases clock frequency)
  - Loop unrolling limits performance penalty

Memory Hierarchy

- Image processing pipeline
- Stages stream data from one filter to the next



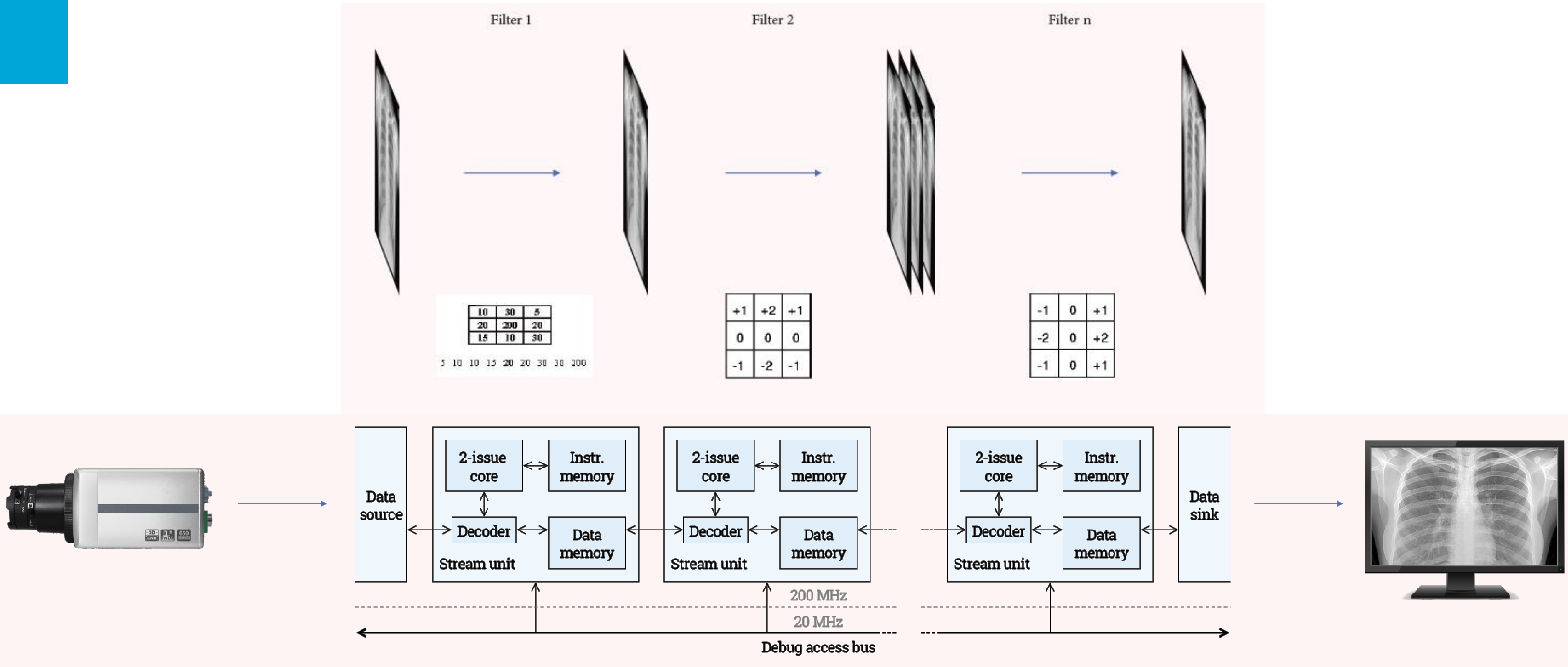
10	30	5
20	200	20
15	10	30

5 10 10 15 20 20 30 30 200

+1	+2	+1
0	0	0
-1	-2	-1

-1	0	+1
-2	0	+2
-1	0	+1

Memory Hierarchy



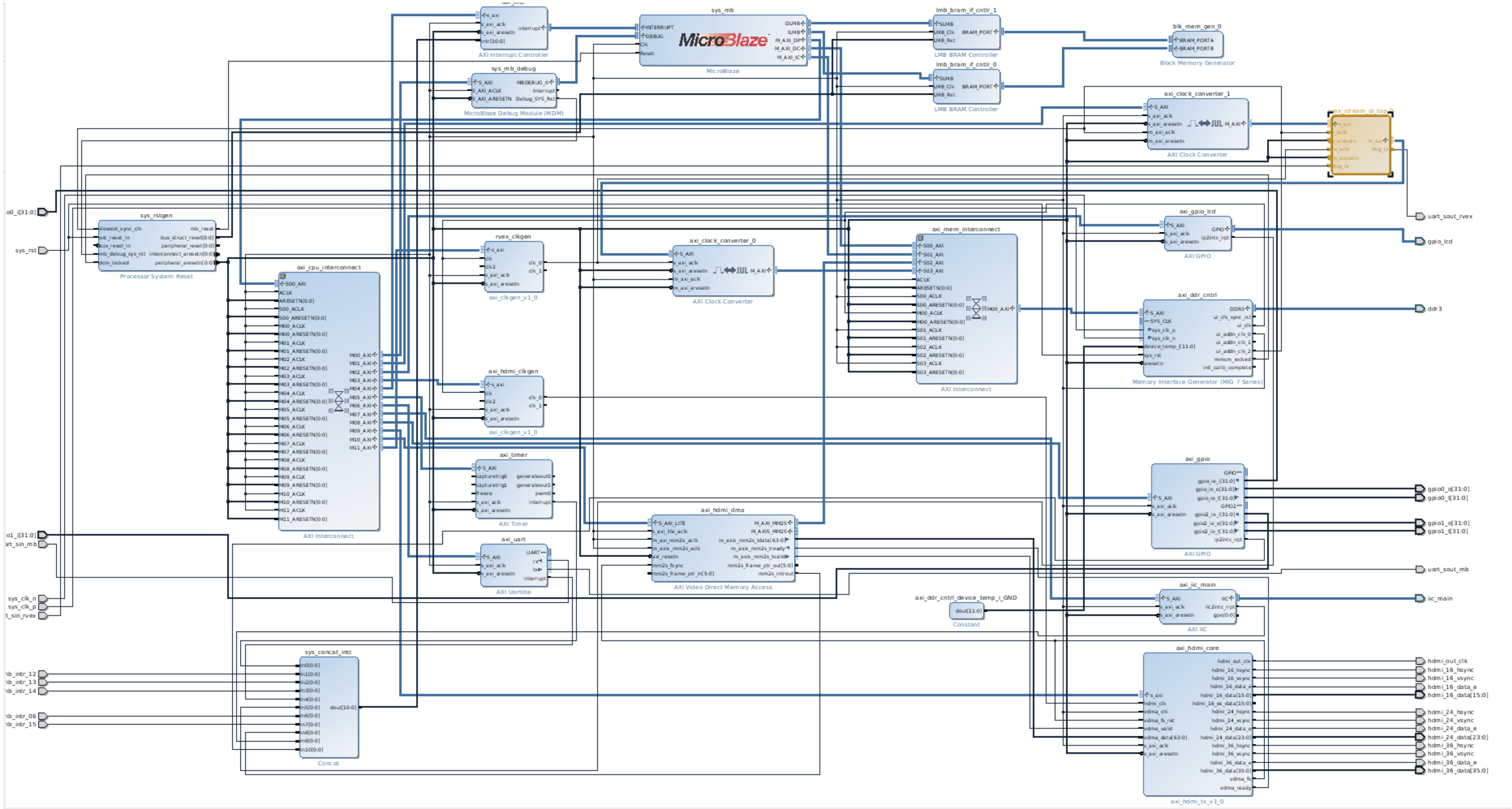
10	30	5
20	200	20
15	10	30

5 10 10 15 20 20 30 30 200

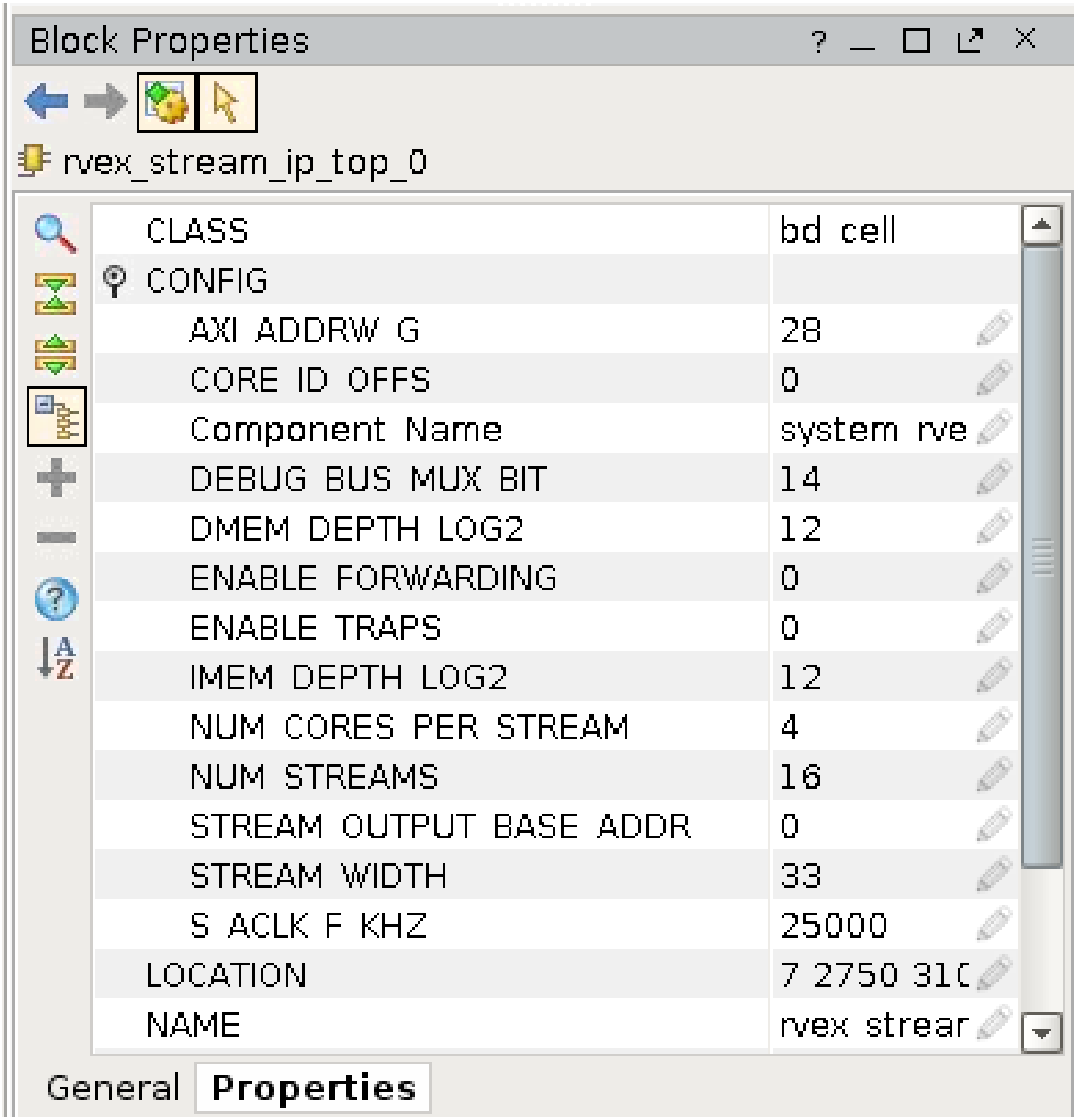
+1	+2	+1
0	0	0
-1	-2	-1

-1	0	+1
-2	0	+2
-1	0	+1

Implementation



Implementation



Block Properties

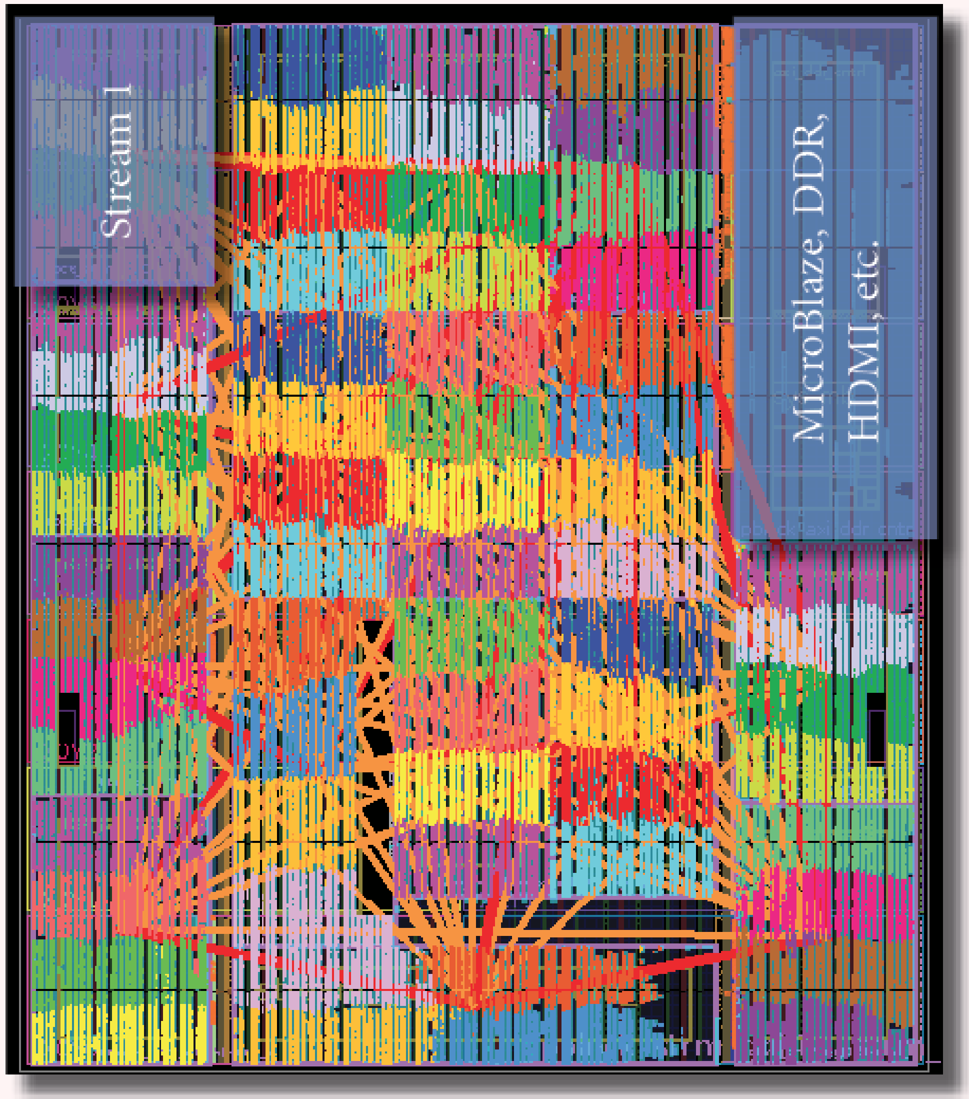
rvex_stream_ip_top_0

CLASS	bd cell
CONFIG	
AXI ADDRW G	28
CORE ID OFFS	0
Component Name	system rve
DEBUG BUS MUX BIT	14
DMEM DEPTH LOG2	12
ENABLE FORWARDING	0
ENABLE TRAPS	0
IMEM DEPTH LOG2	12
NUM CORES PER STREAM	4
NUM STREAMS	16
STREAM OUTPUT BASE ADDR	0
STREAM WIDTH	33
S ACLK F KHZ	25000
LOCATION	7 2750 31(
NAME	rvex strear

General | **Properties**

Synthesized Design

Details

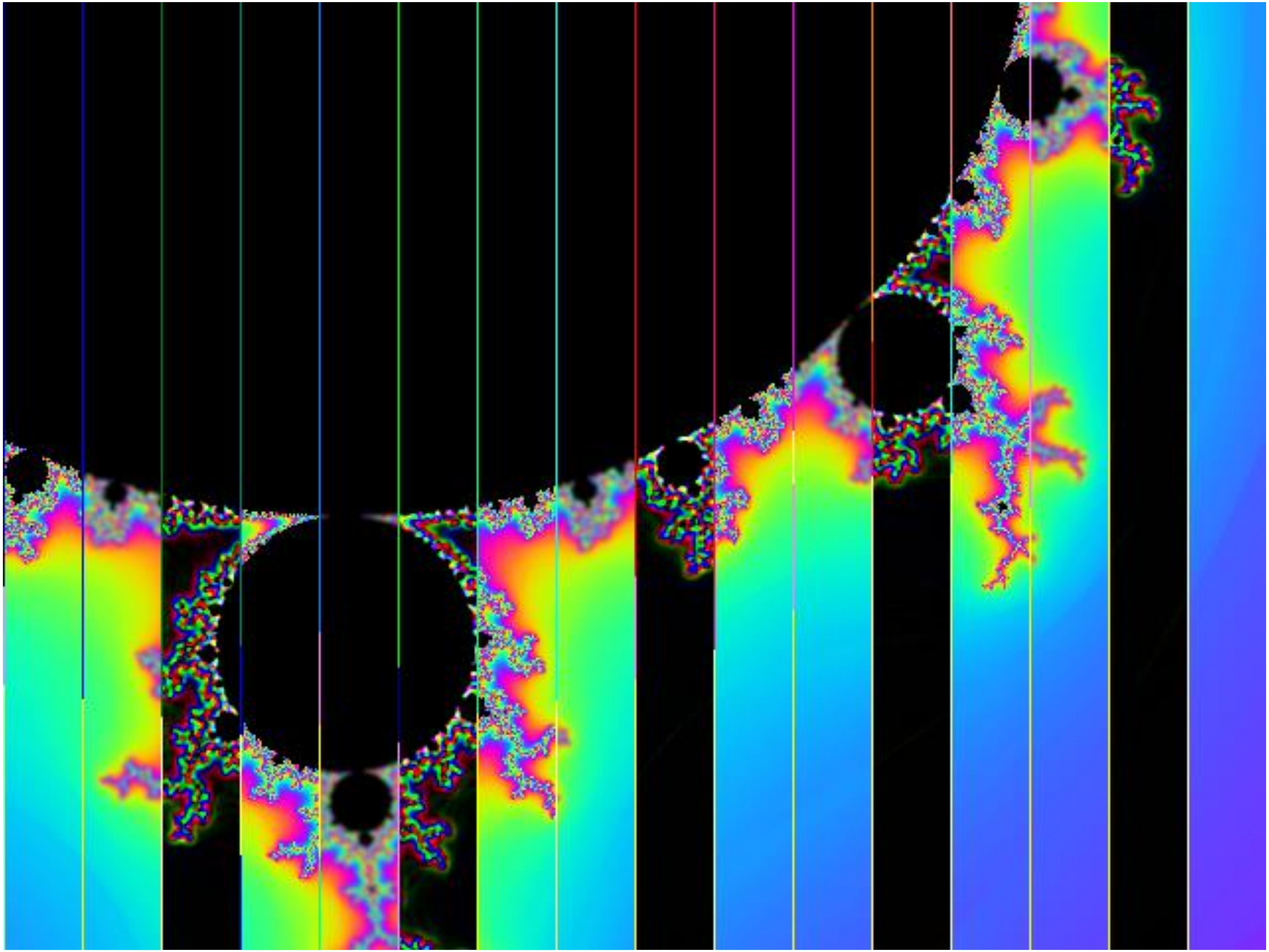


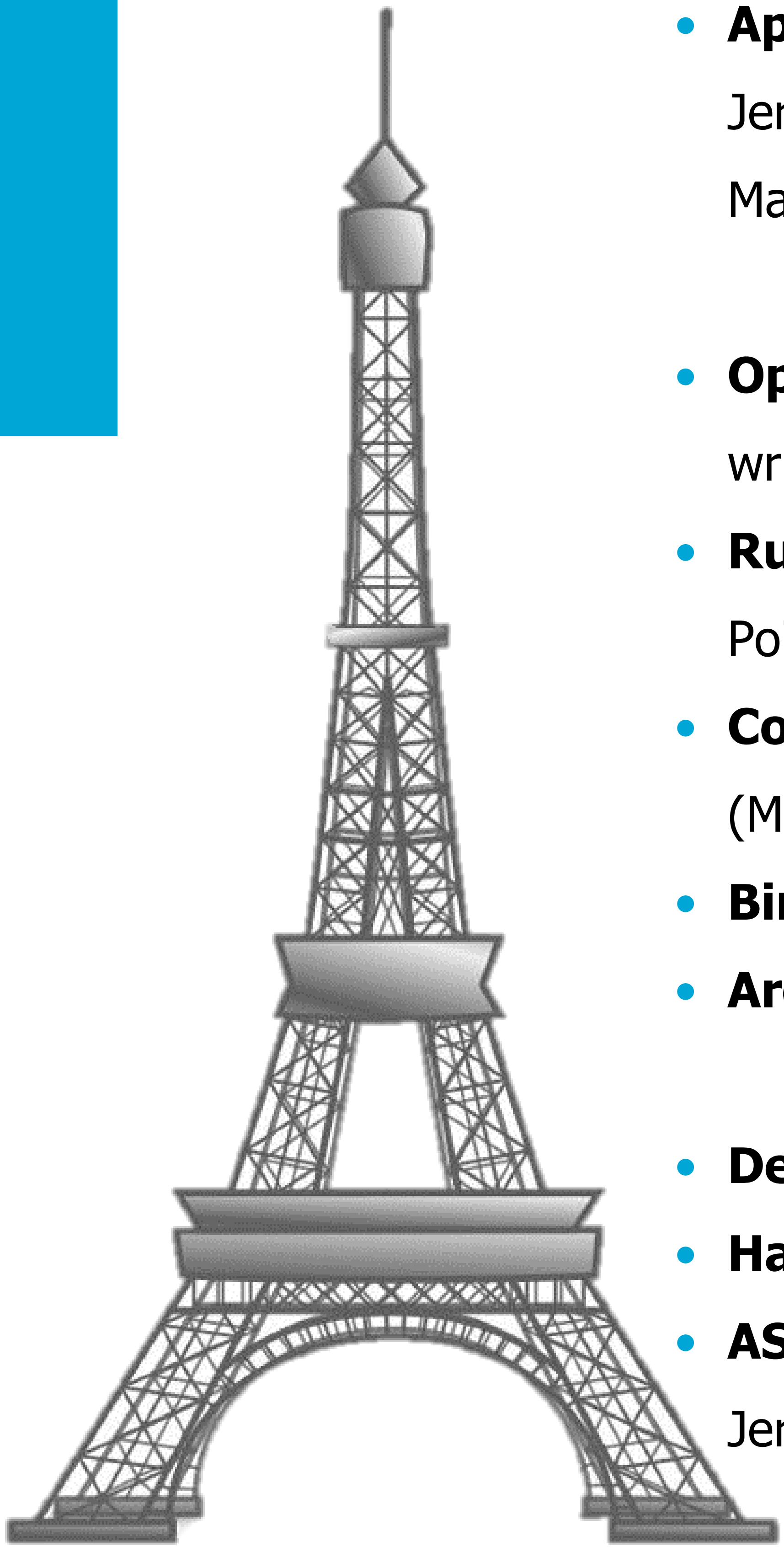
Platform

- **64 cores** (16 streams, 4 cores per stream)
- 200 MHz
- DMA
- MicroBlaze
- DDR
- HDMI

Each colored block is a 2-issue ρ-VEX core.

All 16 streams operate on a slice of the current frame in parallel.

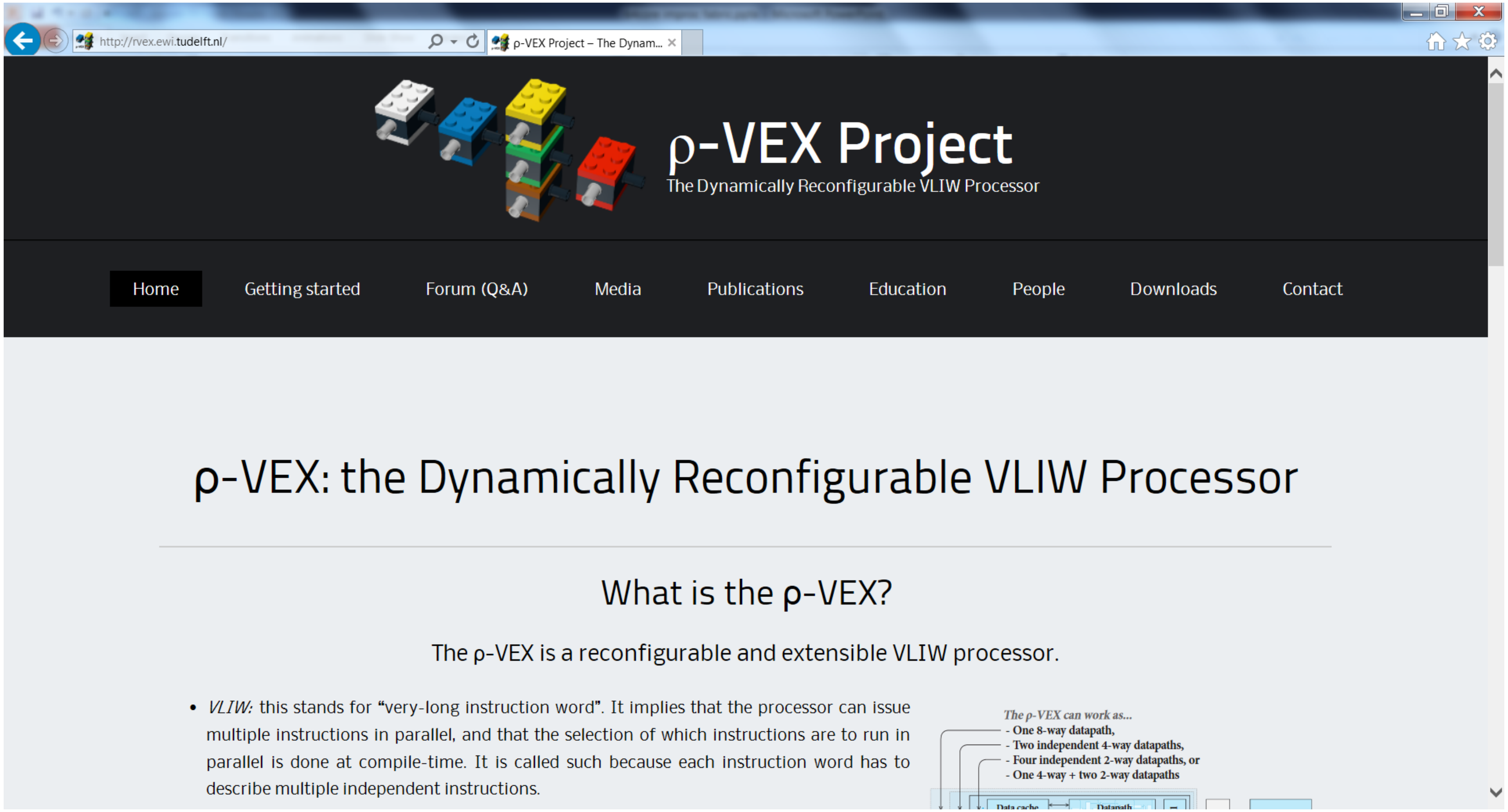




- **Applications** - Image processing pipeline (Rolf, Joost), Doom (Koray, Jeroen), Demos (Muneeb, Joost, Jeroen), Benchmarks SPEC, MiBench, Malardalen, Powerstone (Anthony, Joost)

- **Operating System support** - Linux (Mainly Joost, some low-level code written/fixed/updated by Anthony & Jeroen), FreeRTOS (Jeroen, Muneeb)
- **Runtime libraries** - Newlib (Joost, Anthony), uCLibc (Tom, Joost), Floating Point & Division, math (Joost)
- **Compilers** - HP VEX, GCC (IBM, Anthony, Joost), Cosy (Hugo), LLVM (Maurice, Hugo), Open64 (Joost)
- **Binutils** - Assembler, linker, etc. (Anthony), VEXparse (Anthony, Jeroen)
- **Architectural Simulator** (Joost)

- **Debug** hardware, tools and interface (Jeroen)
- **Hardware** design - VHDL (Jeroen)
- **ASIC** manufacturing effort - core (Lennart), interface (Shizao) supported by Jeroen


http://rvex.ewi.tudelft.nl/
ρ-VEX Project – The Dynam...
ρ-VEX Project
The Dynamically Reconfigurable VLIW Processor
Home
Getting started
Forum (Q&A)
Media
Publications
Education
People
Downloads
Contact

ρ-VEX: the Dynamically Reconfigurable VLIW Processor

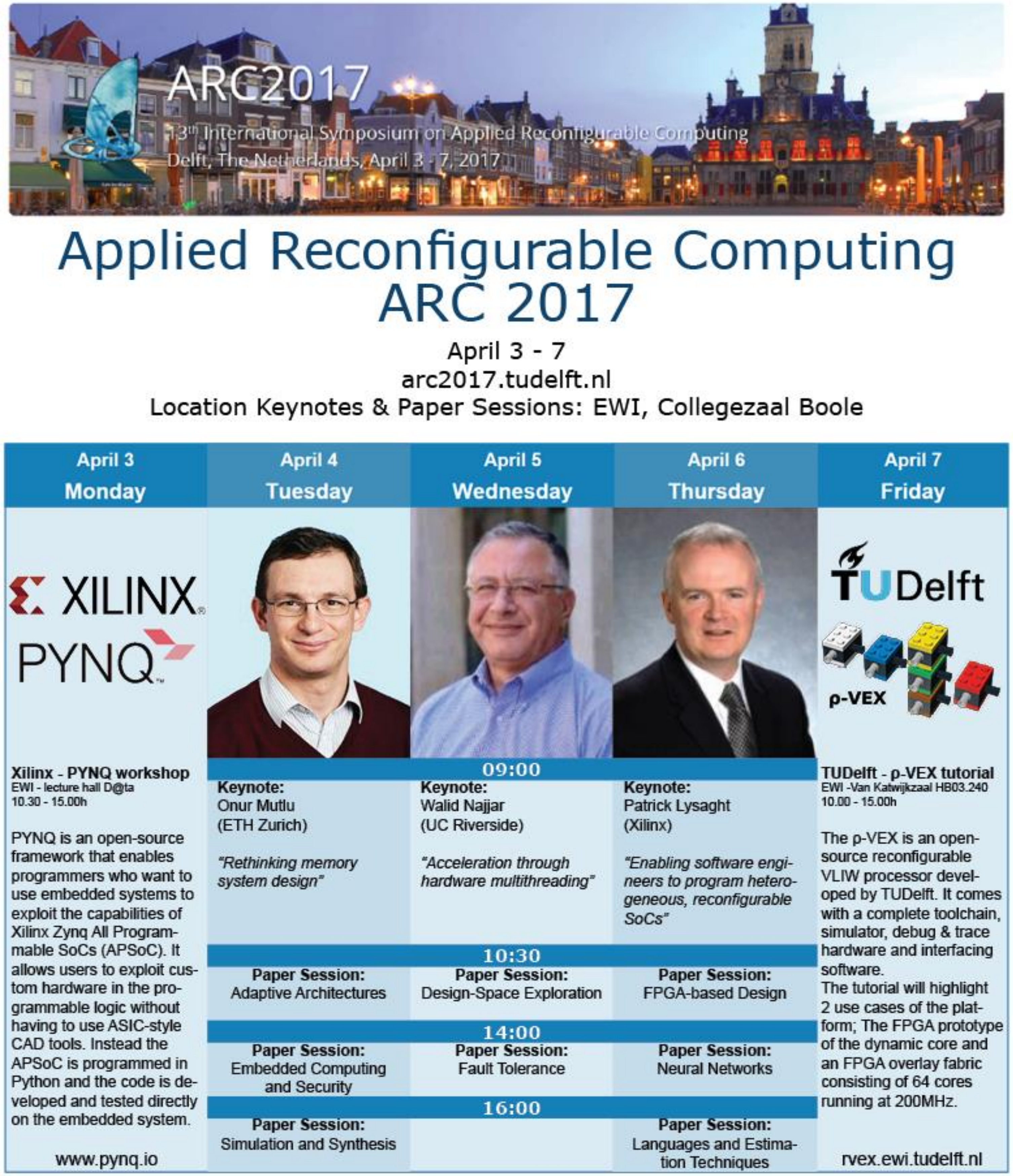
What is the ρ-VEX?

The ρ-VEX is a reconfigurable and extensible VLIW processor.

- *VLIW:* this stands for "very-long instruction word". It implies that the processor can issue multiple instructions in parallel, and that the selection of which instructions are to run in parallel is done at compile-time. It is called such because each instruction word has to describe multiple independent instructions.

The ρ-VEX can work as...
- One 8-way datapath,
- Two independent 4-way datapaths,
- Four independent 2-way datapaths, or
- One 4-way + two 2-way datapaths
Data cache
Datapath

http://rvex.ewi.tudelft.nl


ARC2017
13th International Symposium on Applied Reconfigurable Computing
Delft, The Netherlands, April 3 - 7, 2017
XILINX
PYNQ
TUDelft
ρ-VEX

Applied Reconfigurable Computing
ARC 2017

April 3 - 7
arc2017.tudelft.nl
Location Keynotes & Paper Sessions: EWI, Collegezaal Boole

	April 3 Monday	April 4 Tuesday	April 5 Wednesday	April 6 Thursday	April 7 Friday
09:00	**Xilinx - PYNQ workshop** EWI - lecture hall D@ta 10.30 - 15.00h PYNQ is an open-source framework that enables programmers who want to use embedded systems to exploit the capabilities of Xilinx Zynq All Programmable SoCs (APSoC). It allows users to exploit custom hardware in the programmable logic without having to use ASIC-style CAD tools. Instead the APSoC is programmed in Python and the code is developed and tested directly on the embedded system. www.pynq.io	**Keynote:** Onur Mutlu (ETH Zurich) *"Rethinking memory system design"*	**Keynote:** Walid Najjar (UC Riverside) *"Acceleration through hardware multithreading"*	**Keynote:** Patrick Lysaght (Xilinx) *"Enabling software engineers to program heterogeneous, reconfigurable SoCs"*	**TUDelft - ρ-VEX tutorial** EWI -Van Katwijkzaal HB03.240 10.00 - 15.00h The ρ-VEX is an open-source reconfigurable VLIW processor developed by TUDelft. It comes with a complete toolchain, simulator, debug & trace hardware and interfacing software. The tutorial will highlight 2 use cases of the platform; The FPGA prototype of the dynamic core and an FPGA overlay fabric consisting of 64 cores running at 200MHz. rvex.ewi.tudelft.nl
10:30		**Paper Session:** Adaptive Architectures	**Paper Session:** Design-Space Exploration	**Paper Session:** FPGA-based Design	
14:00		**Paper Session:** Embedded Computing and Security	**Paper Session:** Fault Tolerance	**Paper Session:** Neural Networks	
16:00		**Paper Session:** Simulation and Synthesis		**Paper Session:** Languages and Estimation Techniques	

Springer

XILINX

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

TU Delft Delft University of Technology

TU Delft

ρ-VEX