Techniques for data processing in real-time using FPGAs

Dr Grzegorz Korcyl

Department of Information Technologies

Jagiellonian University, Cracow
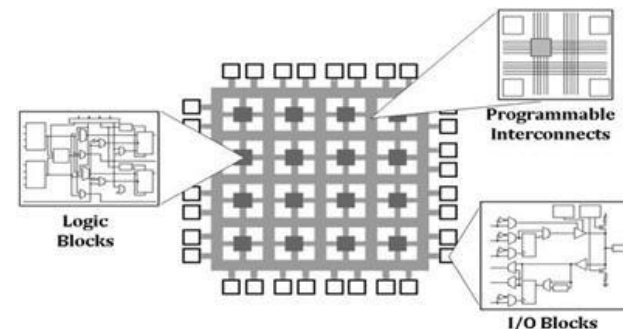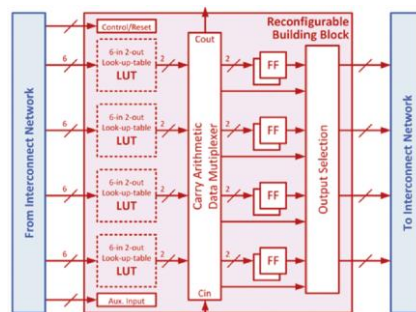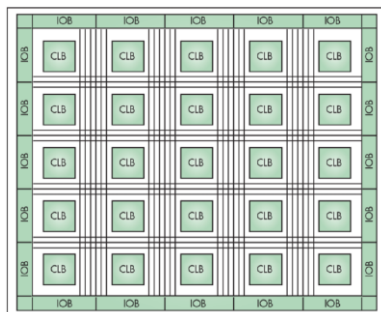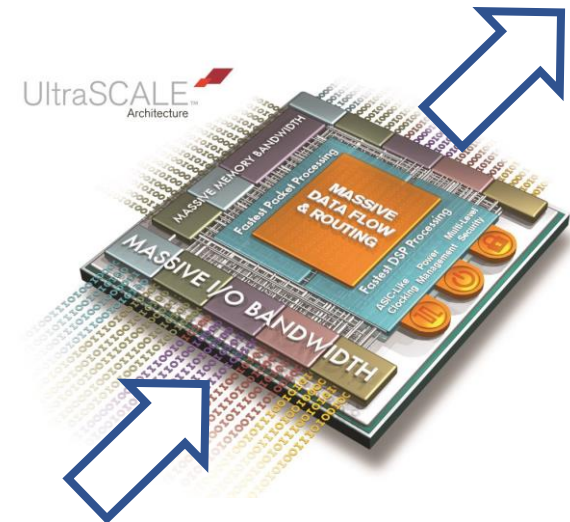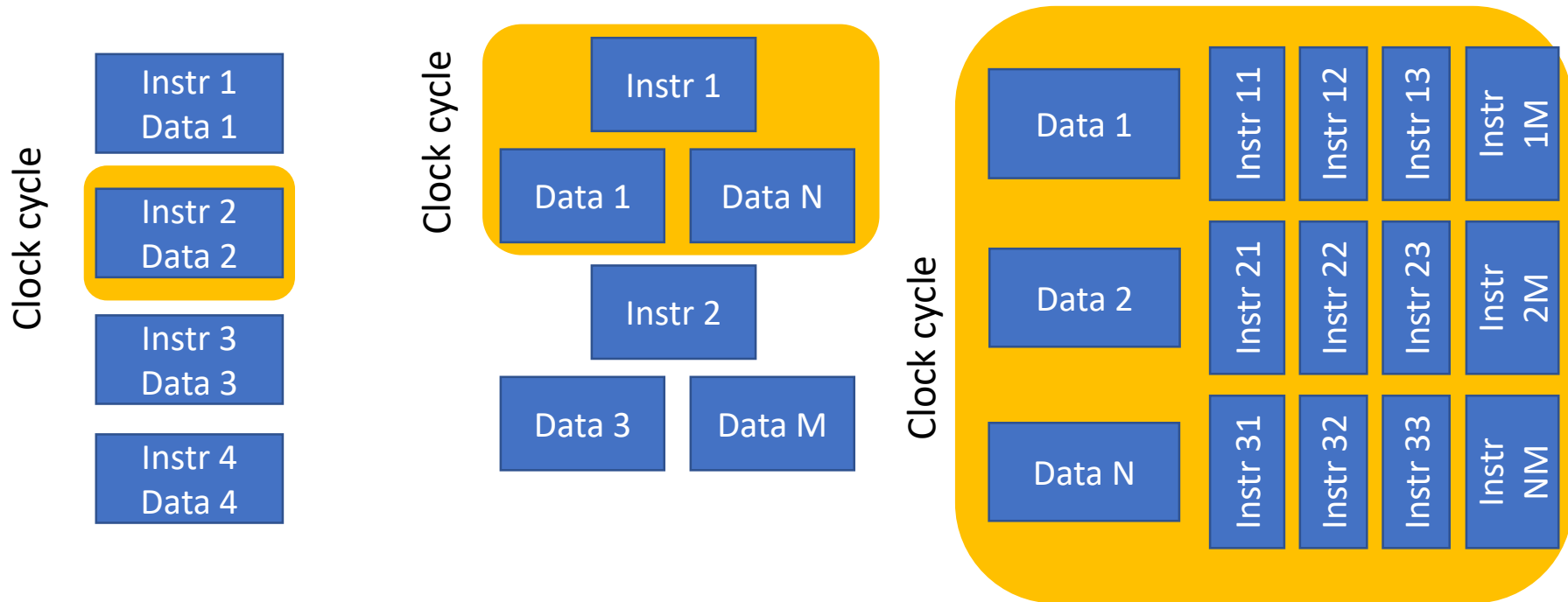
What are FPGAs

- ## Field Programmable Gate Arrays
  - Reconfigurable devices for processing digital data streams
  - Adaptable computing resources
  - No predefined architecture
  - Massive parallelism
  - Streamlined processing

CPU   vs   GPU   vs   FPGA

Clock cycle (CPU)

| Instr 1 / Data 1 |
| Instr 2 / Data 2 |
| Instr 3 / Data 3 |
| Instr 4 / Data 4 |

Clock cycle (GPU)

Instr 1 — Data 1 / Data N

Instr 2 — Data 3 / Data M

Clock cycle (FPGA)

Data 1	Instr 11	Instr 12	Instr 13	Instr 1M
Data 2	Instr 21	Instr 22	Instr 23	Instr 2M
Data N	Instr 31	Instr 32	Instr 33	Instr NM

- **CPU**
  - Single Instruction Single Data per core
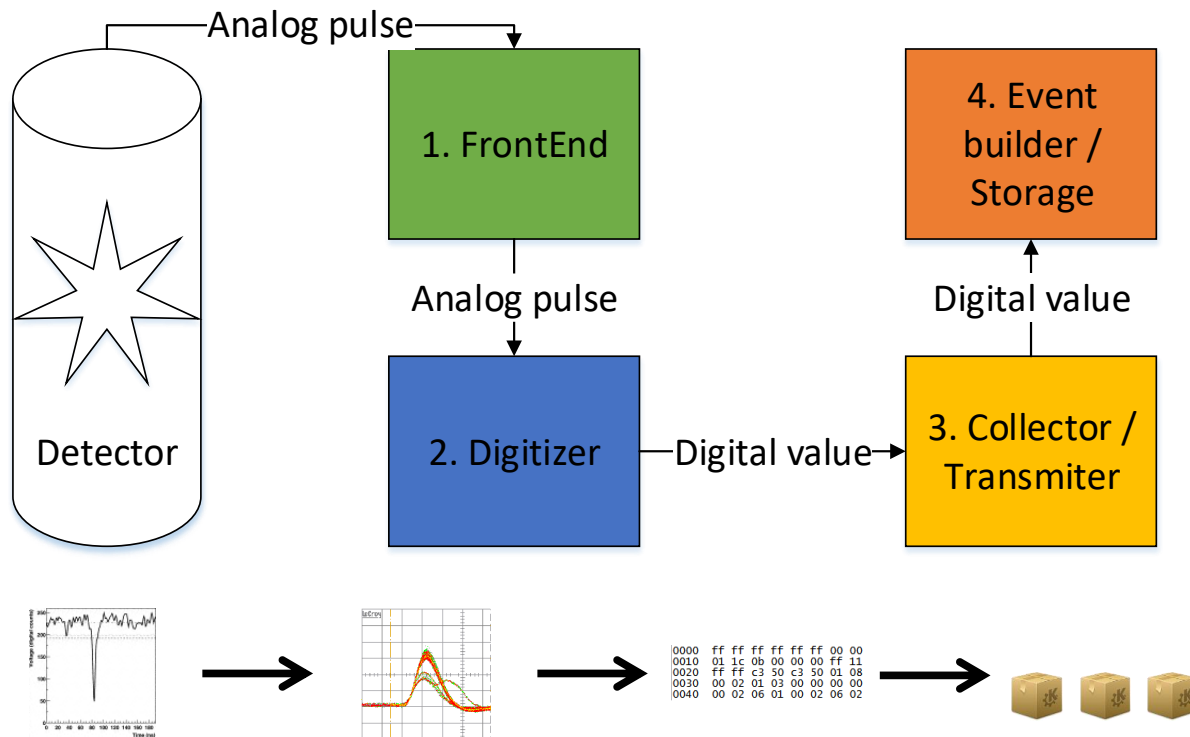 - Multiple cores
  - High clock freq.
  - Operating system

- **GPU**
  - Single Instruction Multiple Data
  - High clock freq.
 - Memory access
  - Accelerates CPU

- **FPGA**
  - Massive parallelism
  - Streamlined processing
  - Low clock freq.
  - Instant memory access
  - Standalone platforms

Processing pipeline

JPET Readout System

- Entirely based on FPGAs
  - Front End boards
    - Analogue signal discrimination
  - Digitizers / Data collectors
    - TRBv3 boards
    - TDC in FPGA
  - Data processing and visualization
    - Controller board
    - Event by event processing
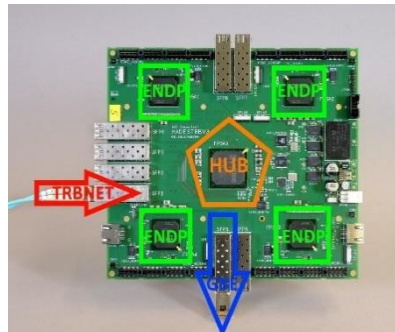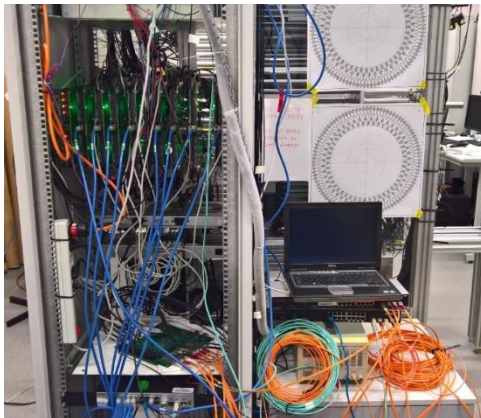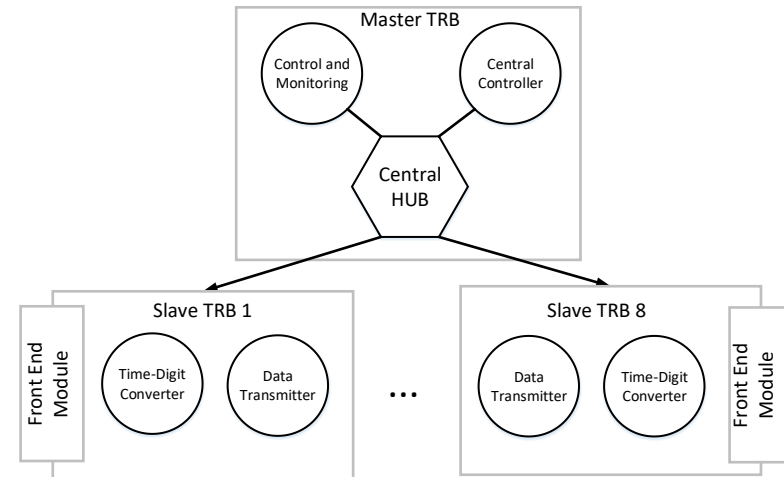
Traxler, M.; Korcyl, G.; Bayer, E.; Maier, L.; Michel, J.; Palka, M.
„A compact system for high precision time measurements (<14 ps RMS)
and integrated acquisition for a large number of channels",
JINST 10.1088/1748-0221/6/12/C12004
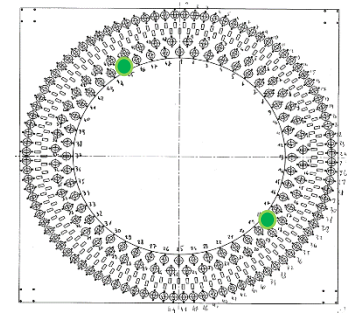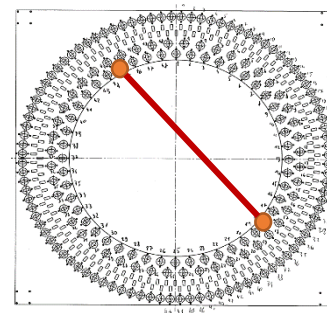
JPET processing pipeline

- Data processing steps
  - Data units reception and assembly
  - Extraction of timing data
  - Application of detector geometry
  - Application of calibration parameters
  - Search for time coincidences
  - Filtration
  - Construction of histogram and visualization

```
0000  ff ff ff ff ff ff 00 00
0010  01 1c 0b 00 00 00 ff 11
0020  ff ff c3 50 c3 50 01 08
0030  00 02 01 03 00 00 00 00
0040  00 02 06 01 00 02 06 02
```

Hit1: ch 1, 115 ns, TOT 5 ns
Hit2: ch 2, 116 ns, TOT 7 ns
...
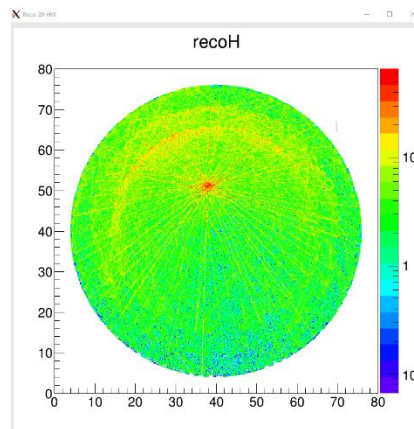
JPET processing pipeline

- CPU   vs   FPGA
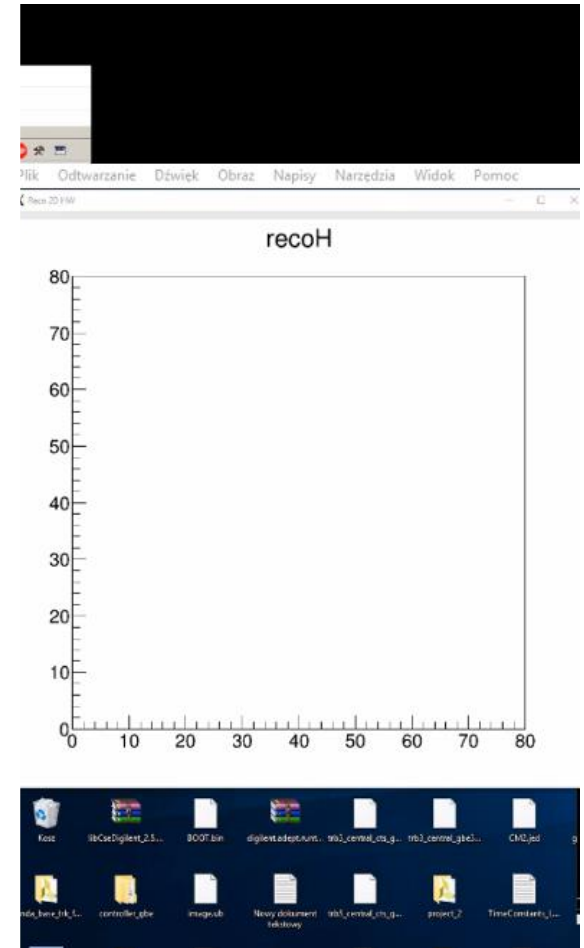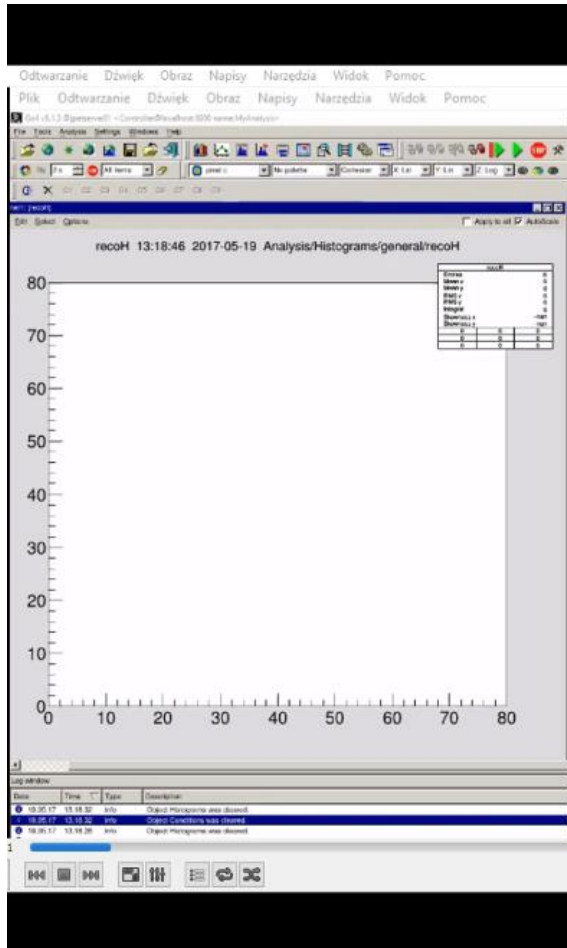
Modular JPET readout

	JPET	Modular JPET	
Scintillators	192	312	**1.6x**
Analog channels	1536	4992	**2.9x**
Digitizers	32	48	**6x**
Logic [k cells]	350	5400	**15.4x**
Memory [Mb]	19	272	**14.3x**
DSP	900	3972	**4.4x**
ARM cores	2	4 + 2x RT + 1x GPU	**>2x**
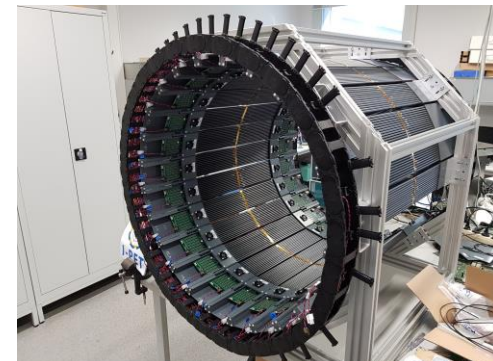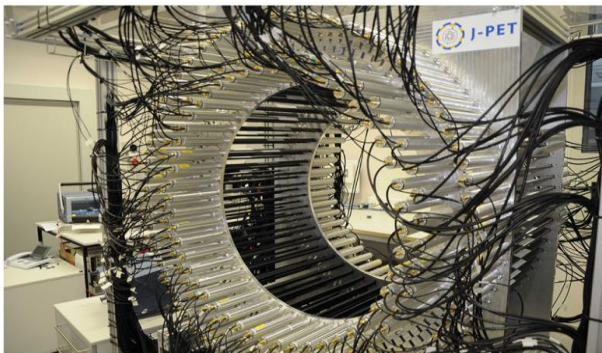
Novel development techniques

- Reduce HDL logic development to minimum
  - Time consuming, requiring experience, error-prone process

- Block designs
  - Library of ready to use, configurable components (IPCores)
- High Level Synthesis
  - Component development in C/C++/OpenCL
  - Compilation into HDL IPCore

- Algorithmic/data processing components in HLS
- Hardware interfacing in HDL

- Build entire systems without HDL
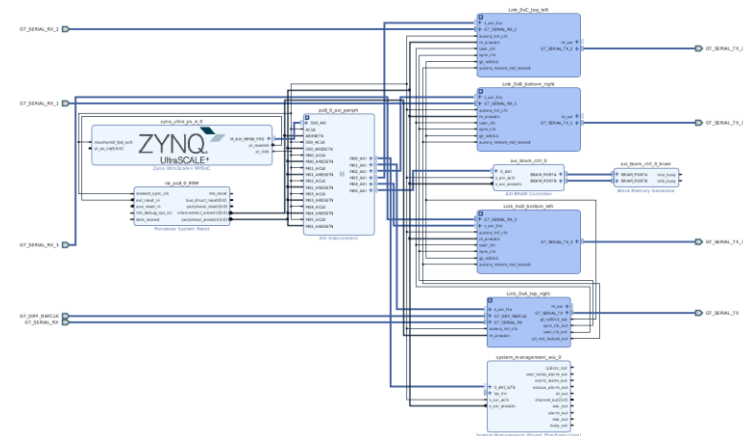
Development in HLS

- Single function – single component
  - Function arguments become component interface
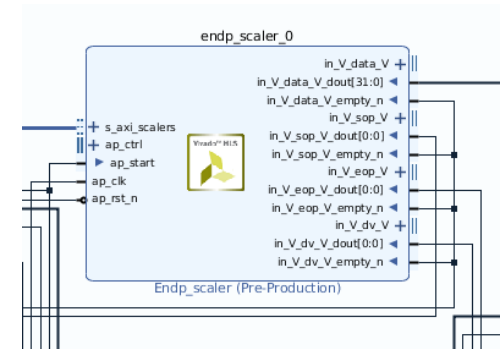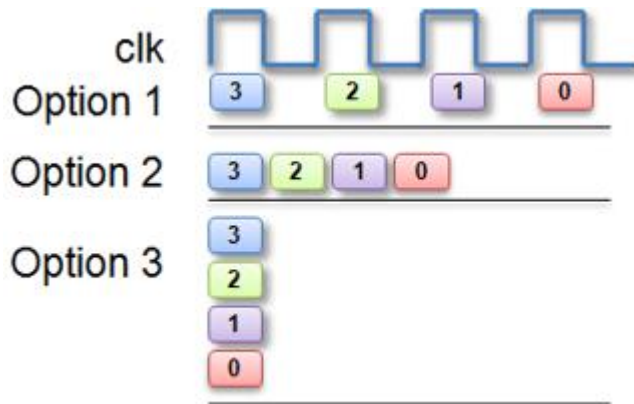 - Function body translated into logic
  - Results analysis with a set of reports
    - Timings, resources
  - Compilation process controlled with a set of #pragmas

endp_scaler_0

s_axi_scalers
ap_ctrl
ap_start
ap_clk
ap_rst_n

in_V_data_V
in_V_data_V_dout[31:0]
in_V_data_V_empty_n
in_V_sop_V
in_V_sop_V_dout[0:0]
in_V_sop_V_empty_n
in_V_eop_V
in_V_eop_V_dout[0:0]
in_V_eop_V_empty_n
in_V_dv_V
in_V_dv_V_dout[0:0]
in_V_dv_V_empty_n

Endp_scaler (Pre-Production)

clk
Option 1   3   2   1   0
Option 2   3 2 1 0
Option 3   3
            2
            1
            0

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	3.33	2.820	0.90

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
2097	2097	2097	2097	none

Detail

Instance

Loop

Loop Name	Latency		Iteration Latency	Initiation Interval		Trip Count	Pipelined
	min	max		achieved	target		
- Loop 1	1568	1568	290	1	1	1280	yes
- data_out_transfer	514	514	4	1	1	512	yes

Utilization Estimates

Summary

| Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	237	-
FIFO	-	-	-	-	-
Instance	150	2596	505116	290902	-
Memory	0	-	512	0	8
Multiplexer	-	-	-	352	-
Register	0	-	24049	320	-
Total	150	2596	529677	291811	8
Available	5376	12288	3456000	1728000	1280
Utilization (%)	2	21	15	16	~0

HLS Example

- Entire flow for Neural Network implementation on FPGA with a single HDL module

- Advantages: fixed latency, level of parallelism, data types

- Used for L1 trigger at ATLAS CERN

Python   C++   Bitstream   hls4ml
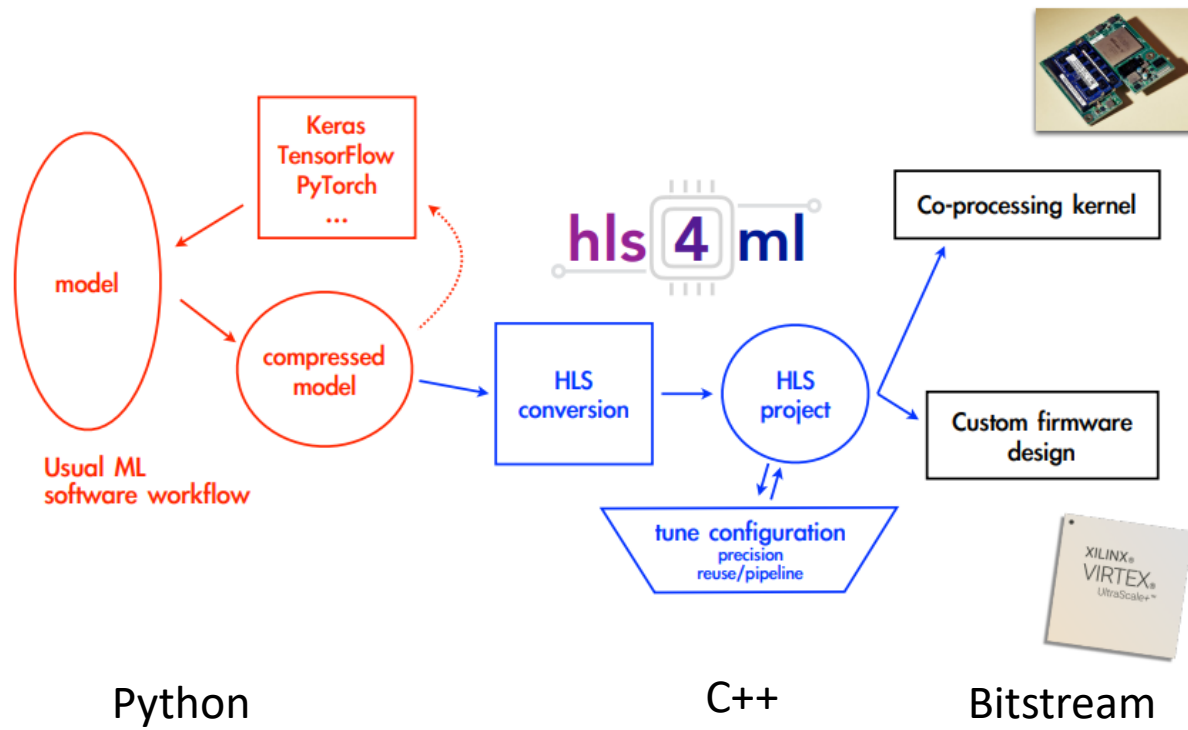
Software Defined environment

- Large selection of hardware platforms on market
    - Standalone boards – System-on-Chip devices – SDSoC
    - Accelerator boards – PCIe enabled - SDAccel

- Development environment
    - Entire project in C/C++
    - Host <–> Kernel architecture
    - Main function – starting point on the host
    - Kernel – hardware accelerated function
    - Encapsulates all other tools and compilers
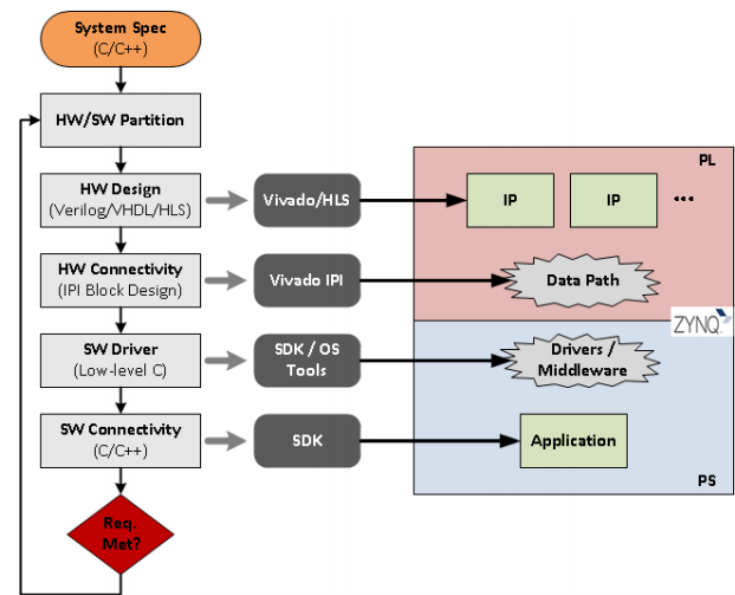
SDx Example

- Conjugate Gradient as HPC benchmark
  - Host prepares data and streams to the accelerated kernel
  - 1464 floating point numerical operations per single iteration
  - Kernel implemented with II=1, latency 150 at 300 MHz
  - 2x faster than Intel Xeon Phi 64-core, 1.7 GHz

  - Not a single HDL line written
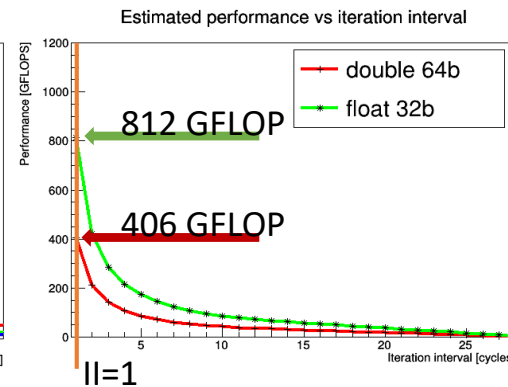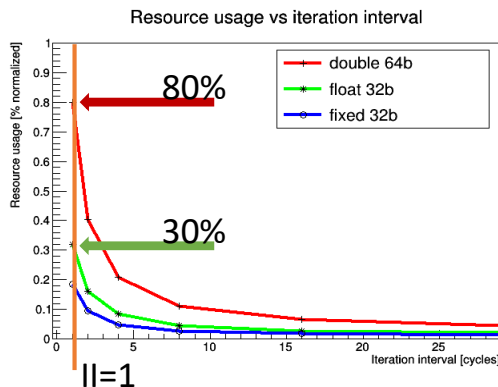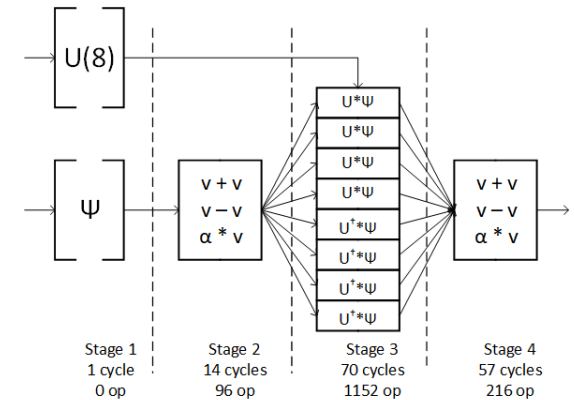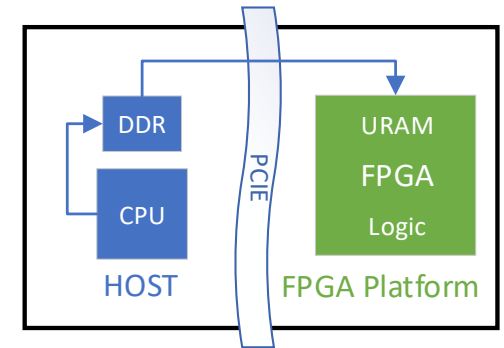
Summary

- FPGAs are no longer reserved for experienced engineers

- Ready to use platforms and new development tools accelerate project timeline
  - High Level Synthesis

- FPGA resources capable to cope with complex problems
  - Numerical algorithm
  - Image processing
  - Artificial Intelligence

- All that in real-time and fixed latency