Next Generation Defect Inspection Computing Challenges

MADEin4 project

Lior Yehieli – Distinguished Member of the Technical Staff, Arik Evdaev, Gal Popovich

Semicon Europa 2021

November 19, 2021

APPLIED MATERIALS®
make possible

Applied Materials in Israel

- ~1,700 Employees
  - » ~79% RD&E (including Application Engineers, Product Support, other)
- Facilities: 613,000 sq. feet
  - » Clean Rooms & Labs: 49,000 sq. feet

APPLIED MATERIALS®
make possible

Haifa

Tel Aviv

Jerusalem

Migdal Haemek

Rehovot

Research Development & Manufacturing Center

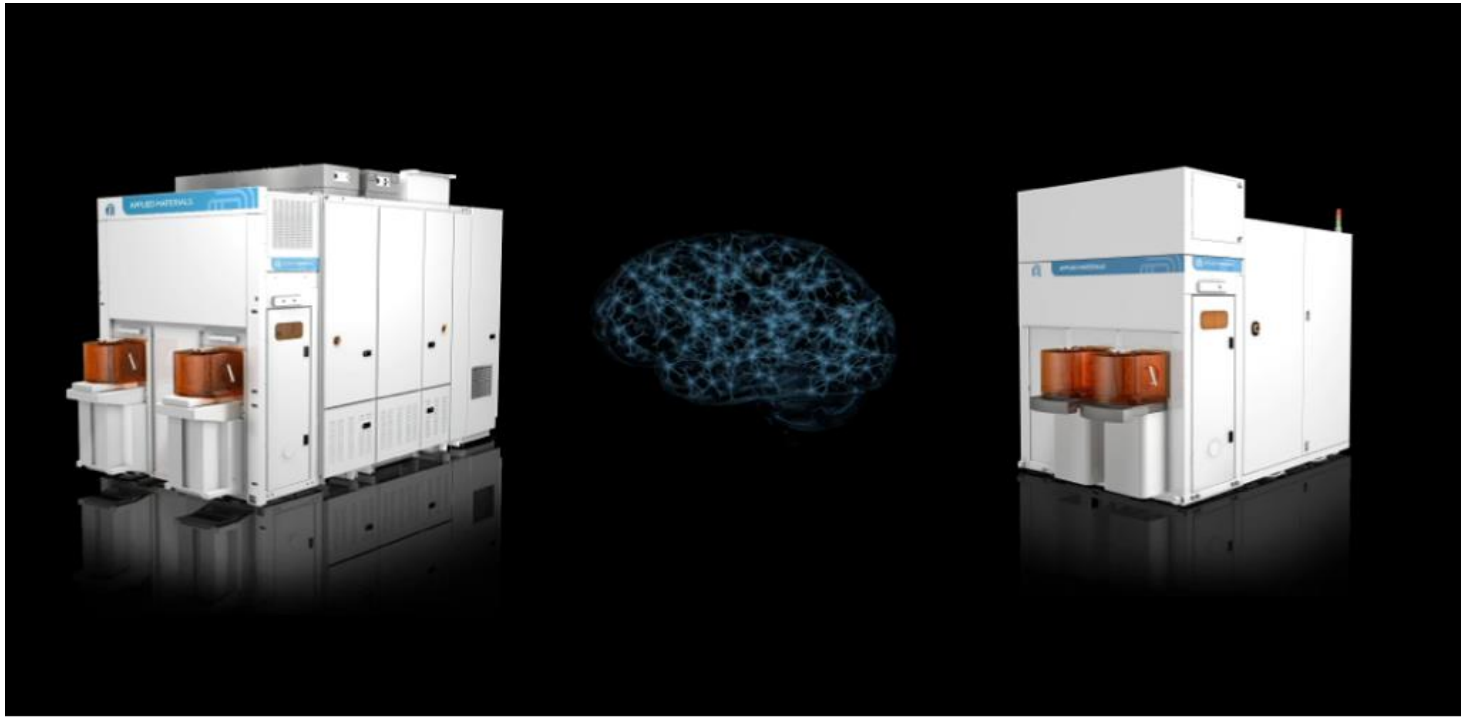
Kiryat Gat

AGENDA

- Optical Wafer Inspection Introduction
- Optical Wafer Inspection Challenges in MADEin4
- Optical Wafer Inspection Computing Development: Methodology & Solutions
- Challenge 1: Defects Images Grabbing
- Challenge 2: Accessing Address Space of the Image's Grabber
- Challenge 3: Image Processing
- Success Story & Summary

Optical Wafer Inspection (OPWI) Introduction

https://www.appliedmaterials.com/products/enlight



Optical Wafer Inspection (OPWI) Challenges in MADEin4

- Main challenges:
  - » Defects images grabbing; high bandwidth channel data acquisition
  - » Accessing address space of the image's grabber; rapid accessing of large DDR memory
  - » Image processing; throughout, utilization & processing capabilities

- High-end incoming data rate ~Tens GB/s
- Acquisition path directly from detector to computing system
- Extreme real-time processing demands
- Hybrid image processing algorithms on Hybrid HW architecture
  - » Conventional computer vision algorithms
  - » Advanced AI / Deep Learning algorithms
- Advanced eco-system
  - » Huge storage for data recording & re-injection
  - » Support massive injection from external Computer Aided Design (CAD) data base
  - » Support automatically advanced & complex Deep Learning based algorithms
- Highly efficient cost/performance computing (cost limitation)
- Computing cabinets footprint limitations
- Fab noise restriction & limitations
- Power limitation, high reliability (24/7)
- No external cloud connectivity due to FAB IP limitations
- Scalability for more advanced algorithms

OPWI Development: Methodology & Solutions

Optimized HPC Server

Methodology

- Since OPWI must keep pace with transistor design shrink, Off-the-Shelf (OTS) components must be considered for appropriate Time To Market (TTM) with the following optimizations:
  - SW architecture
  - HW architecture and configuration
  - Load balancing
  - Processing dispatching

Network connectivity

OTS accelerator/GPU

Source:www.nvidia.com

Cabinet

Source: Prodirve

Source: Prodirve

Data Acquisition path

know your architecture

OPWI Development: Methodology & Solutions

Solutions:

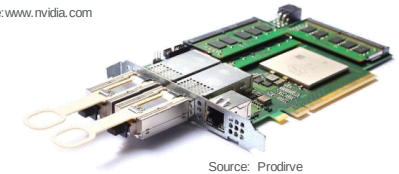
- Integrated OTS components for creating advanced optimized & customized solution
- Select smartly the most appropriate:
  - » Server architecture for your use case application
  - » Accelerator/GPU for your use case
- Design your Acquisition path with generic components as much a possible directly from detectors
- Plan compact computer cabinet – minimal footprint

Optimized HPC Server

Network connectivity

OTS accelerator/GPU

Source: www.nvidia.com



Source: Prodirve

Data Acquisition path

Cabinet

Source: Prodirve

know your architecture

Optical Defect Inspection Tools Computing Architecture

Prodrive Joint concept

Prodrive Joint dev

Optic interface

Large memory DDR4

Customized Dense servers

Acquisition High resolution/BW

- Frame Grabber
- CXP interface
- FPGA processing (Xilinx Zynq Ultrascale+ ZU11EG)

Processing CPU multiple cores + up 4GPU

- Rack Server
- Intel Cascade Lake / Ice Lake
- GPU optimized (up to 4x Tesla V100/A100)

Storage

- NVMe SSD
- SATA

Detectors

Metrology / Inspection Tool

DSP Based Solution

Existing Image Processing System

Advanced New Image Processing System

Unique SW-Infra structure management
Cabinet customized infra-structure

Fast Ethernet interface

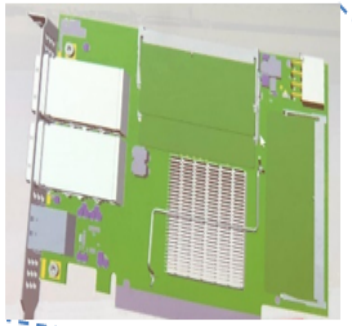
Special Cabinet+Grabber interface

Prodrive Joint Dev

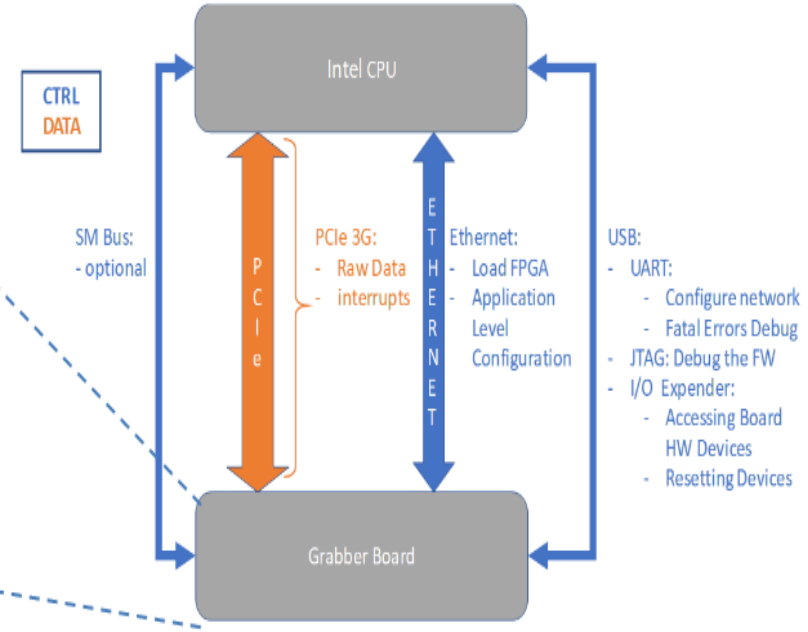
Challenge 1: Defects Images Grabbing

High bandwidth channel data acquisition

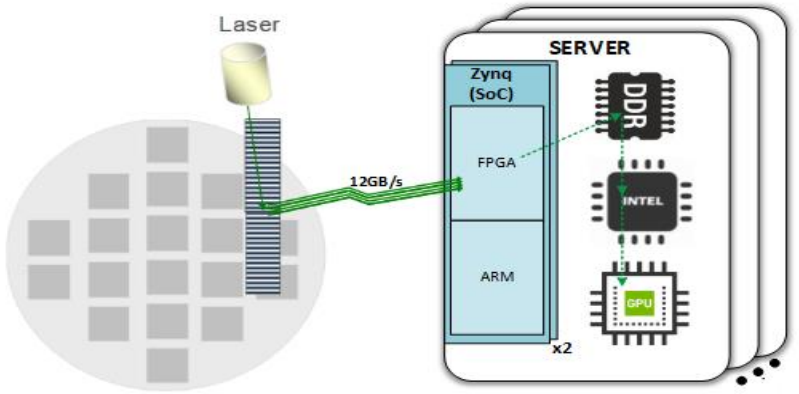
- Problem statement: grabbing the data to > 300GBytes of DDR4 memory
- Solution:
  - » Advanced devices integration:
    - Processing sub system: ARM based system with common interfaces
    - Programmable logic: FPGA for design your custom interface
  - » OTS components block design
  - » On-tool Advanced Debugging infra-structure
  - » Advanced Monitor & diagnostic Using ARM processor



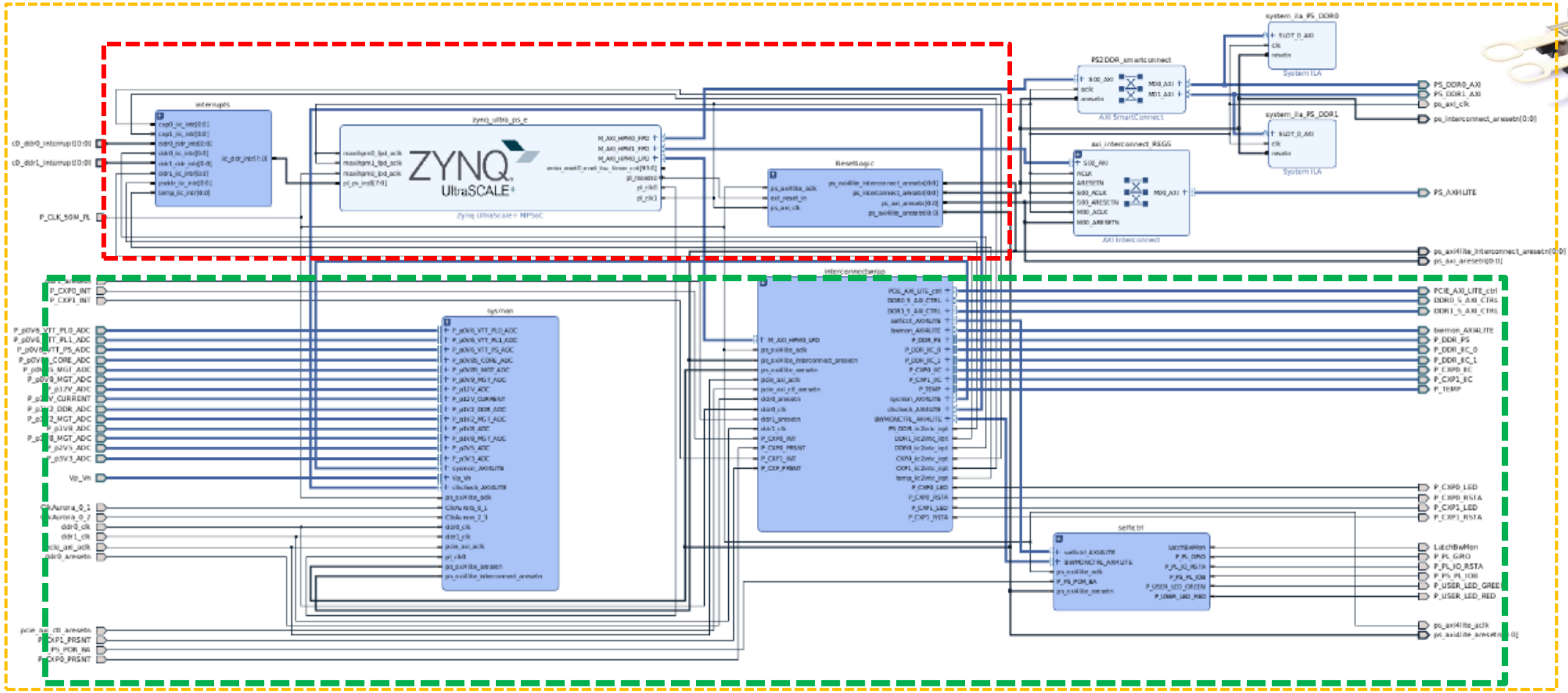
Interfaces between the board & CPU

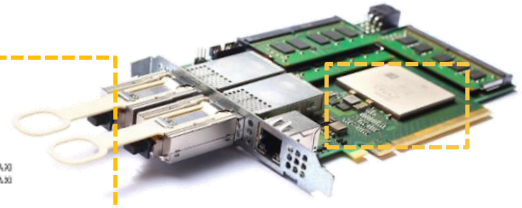


Data Flow from detector to server



Callanage1: High speed connectivity – FPGA Interfaces & Design





Grabber

Processing sub system

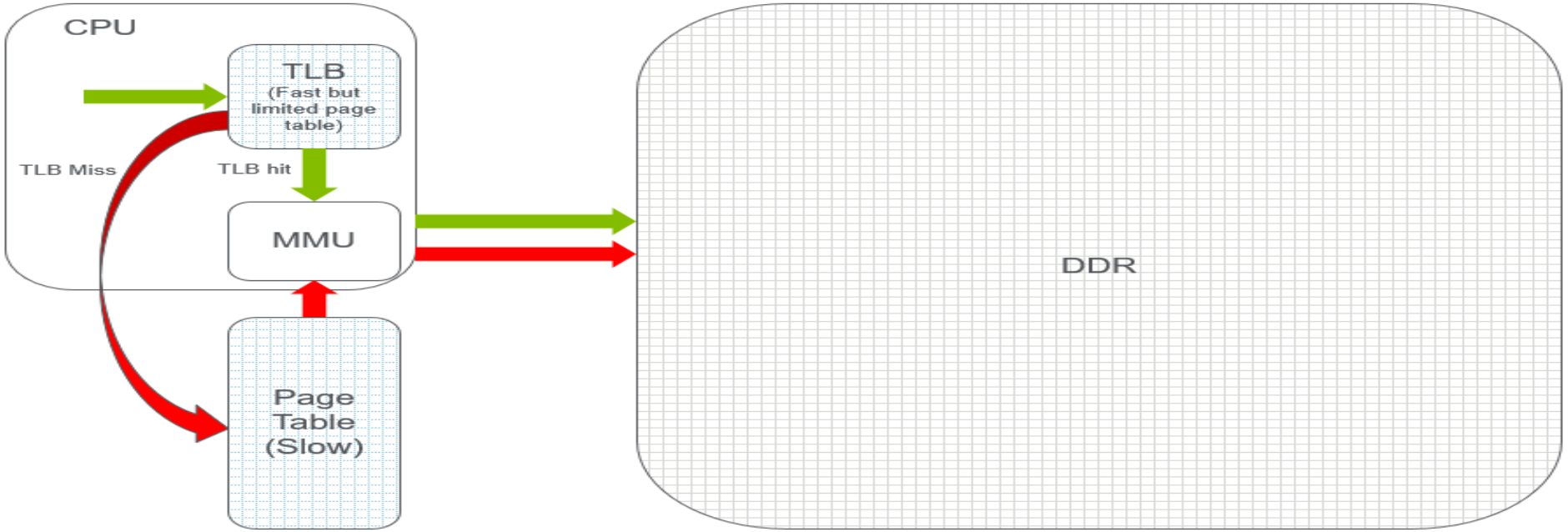
Programable logic

Generic FPGA interfaces without need to write special code – VHDL free design for fast development, mature design & enhanced performance, already optimized modules such as DMA & AXI-Connect

Challenge 2: Accessing Address Space of the Image Grabber

Rapid accessing of large DDR memory

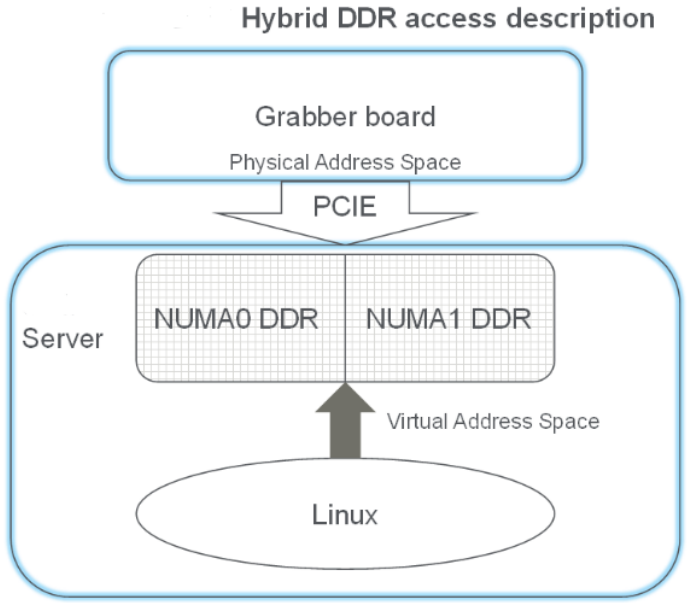
- Problem statement: there is no available solution for accessing large DDR memory address space for wafer optical inspection scenarios
  - » Algorithms require multiple random memory operations in parallel up to 200K image patches per second
  - » Since the application uses huge memory, the translation lookaside buffer (TLB) cannot hold the entire address space which results in low memory access performance

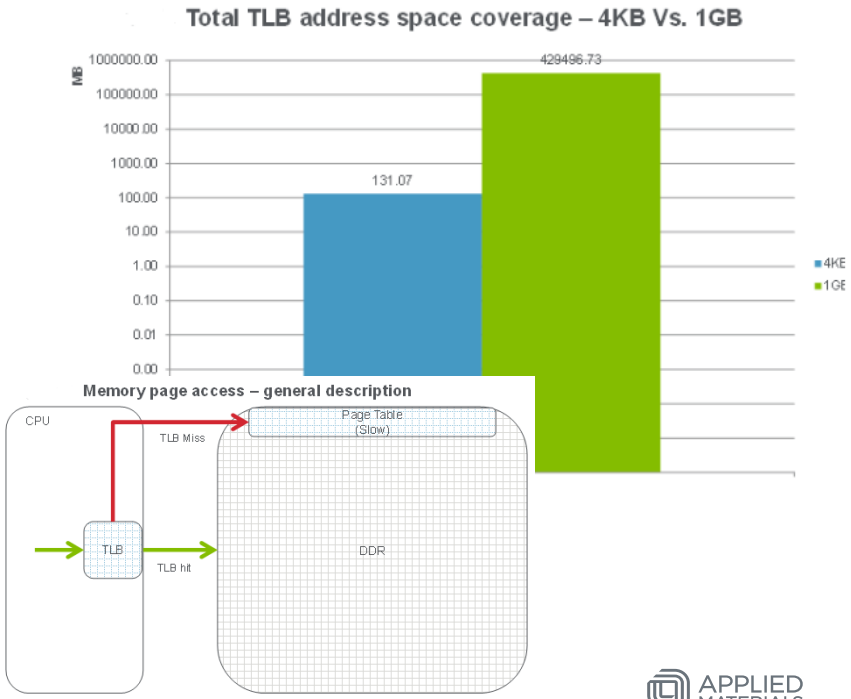


Challenge 2: Utilizing Large DDR Memory

Gain rapid accessing x10 vs. Naïve implementation

- Solutions & Optimizations:
  - » Linux makes it easy to use memory allocation Pages of 1GB: this reduces significantly the number of pages used by the OS, so most of them can reside in the TLB
  - » Split the memory between the CPU sockets: use non-uniform memory accessing (NUMA) architecture that enables faster parallel memory access
  - » Allocate 300GB of memory at boot time: ensure Huge Pages availability before any SW is running
  - » Image split into 1GB fragments: guarantee more than 1GB of contiguous physical memory
  - » Linux open-source SW library (hugetlbfs): enables user friendly access to the pre-allocated memory
  - » FPGA interface: develop a bilateral translation module between Physical and Virtual Address spaces
  - » TLB access: use Memory Warm Up – pushing actively the huge memory pages into the TLB before receiving wafer image

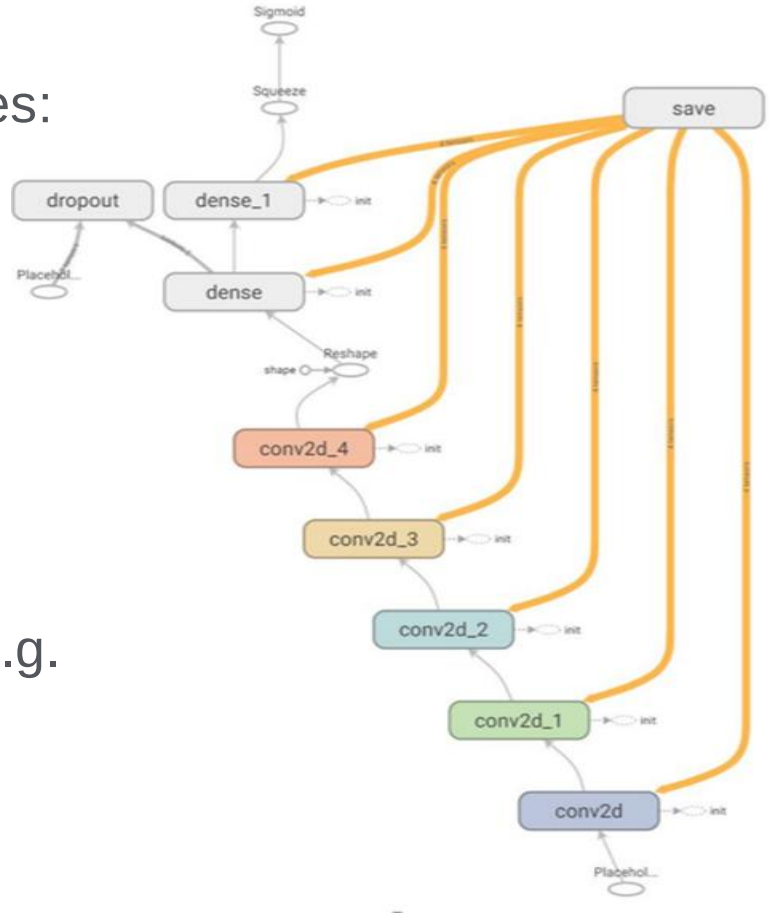




Challenge 3: Image Processing

Throughout, utilization & processing capabilities

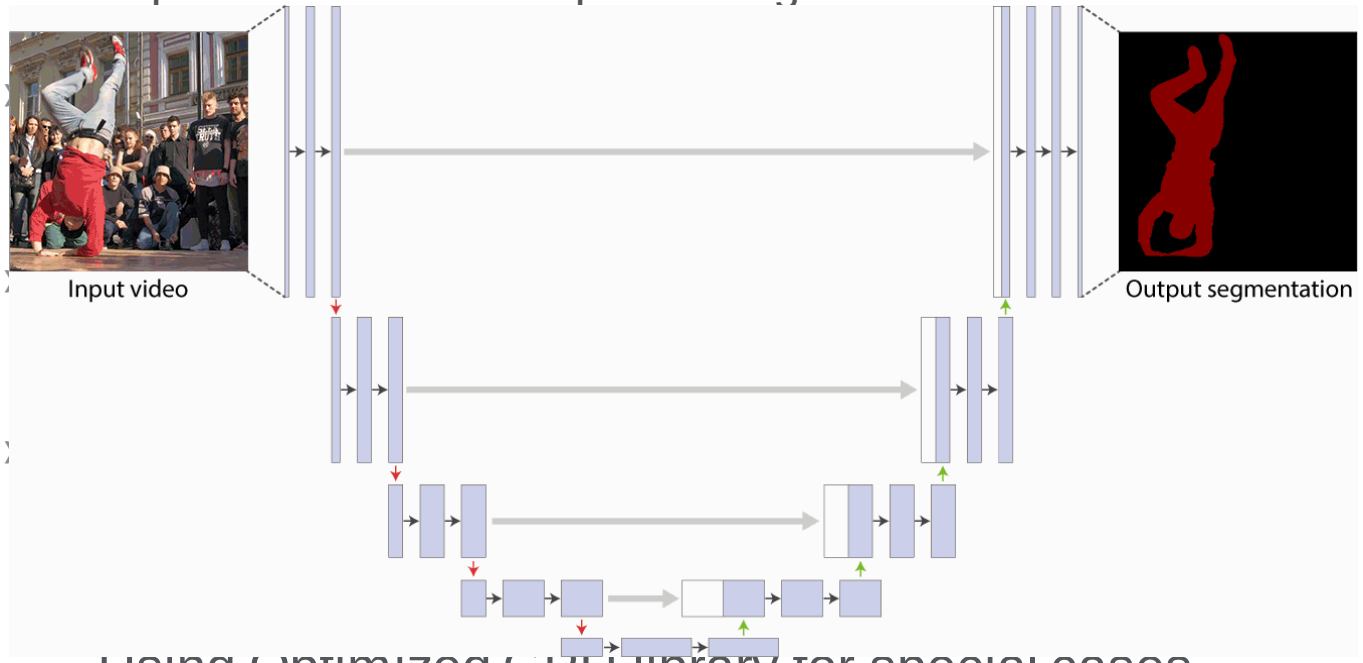
- Problem statement: upon data arrival (GB/s) defect detection requires:
  - » Arranging the data (e.g. clip the images around defects)
  - » Processing (e.g. align the clipped images)
  - » Heavy pixel crunching to distinguish signal from noise and detect defects
    - Computer vision
    - Deep Learning neural network model
  - » Get complementary data and combine it as part of the algorithm (e.g. CAD data)



Hybrid approach: CPU & GPU in Massive data processing

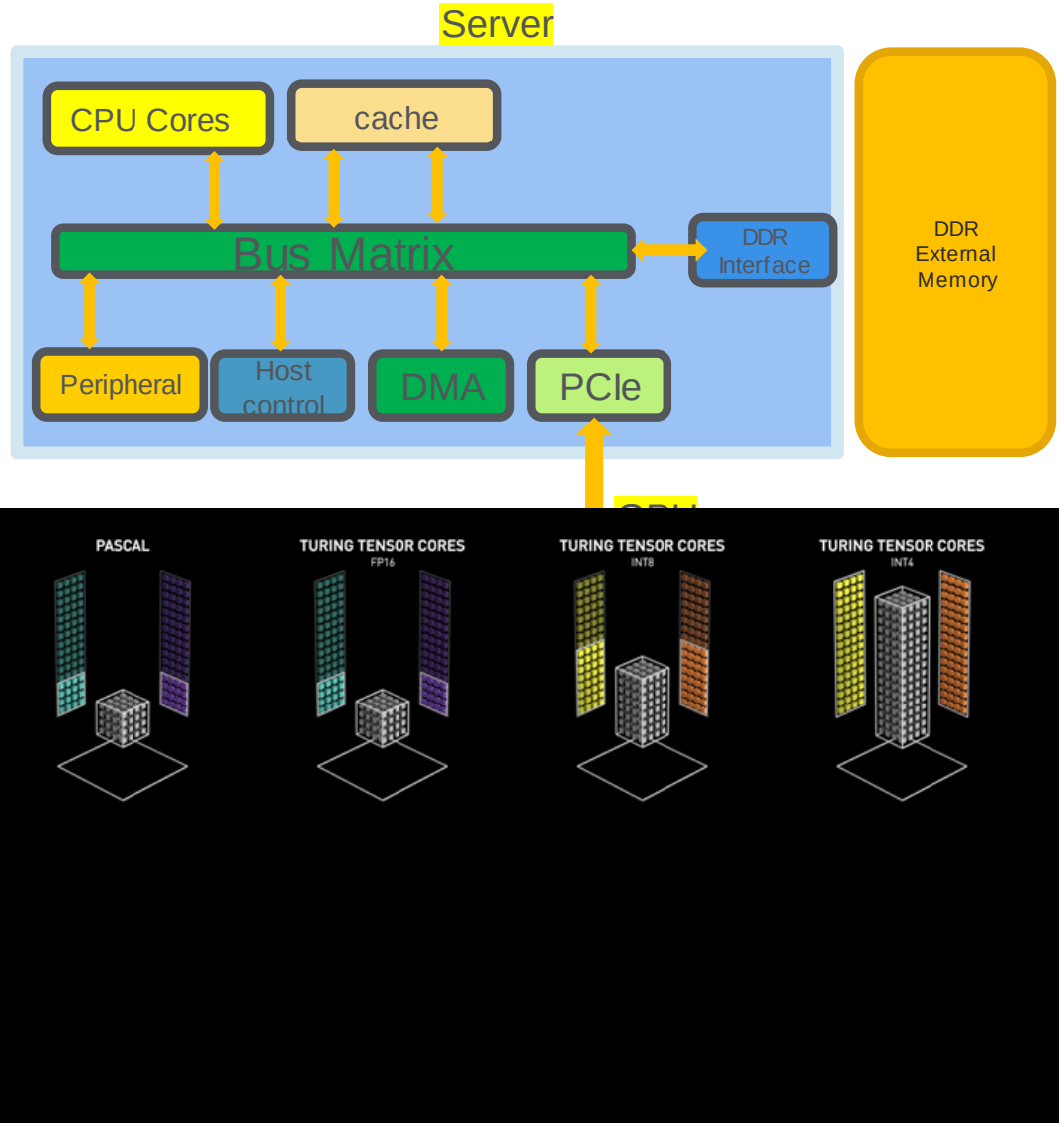
Challenge 3 Optimization Techniques for Processing Acceleration

- Solutions Guidelines:
  - » Evaluate overall processing capabilities:
    - GPU, CPU cores, what is the theoretical vs. practical performance
      - GPU FP16 maximal theoretical 320Tops, in our use case 20Tops
    - Accuracy: single precision 32bits, 16 bits, int8 ?
  - » Computer vision vs. Deep Learning: Neural network is x10



    - Using Optimized GPU library for special cases (e.g.XLA library)

https://www.inag.ufscar.br/projects/conv-neural-networks-1



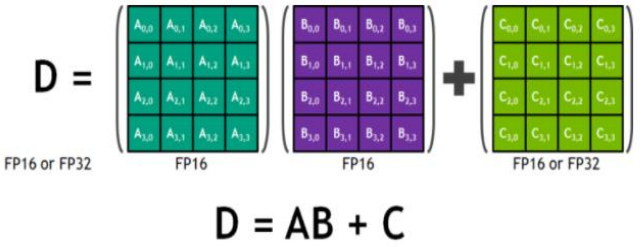
Source:www.nvidia.com

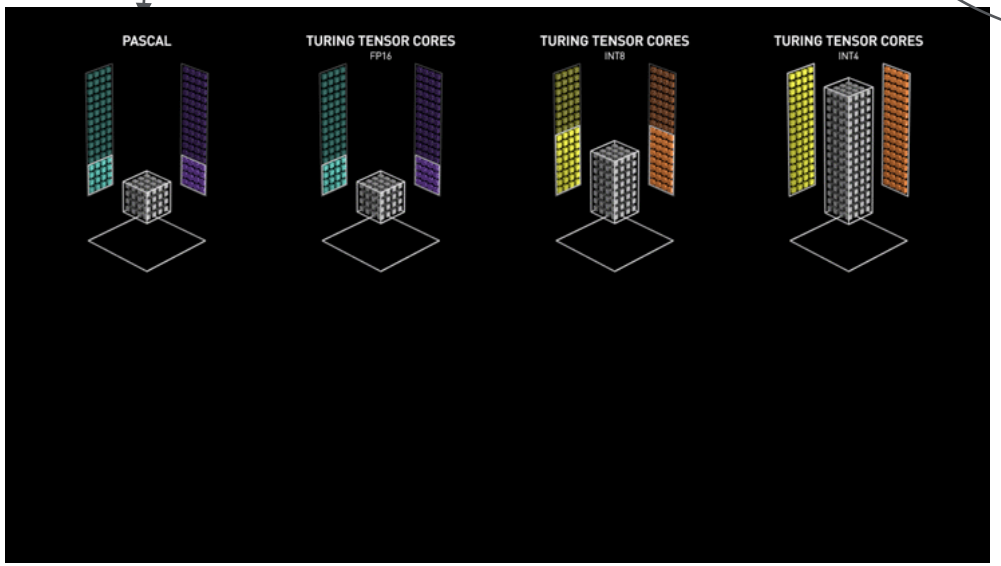
Innovated Optimization techniques for Processing Acceleration

Overlap memory with logic and algorithm adaptation

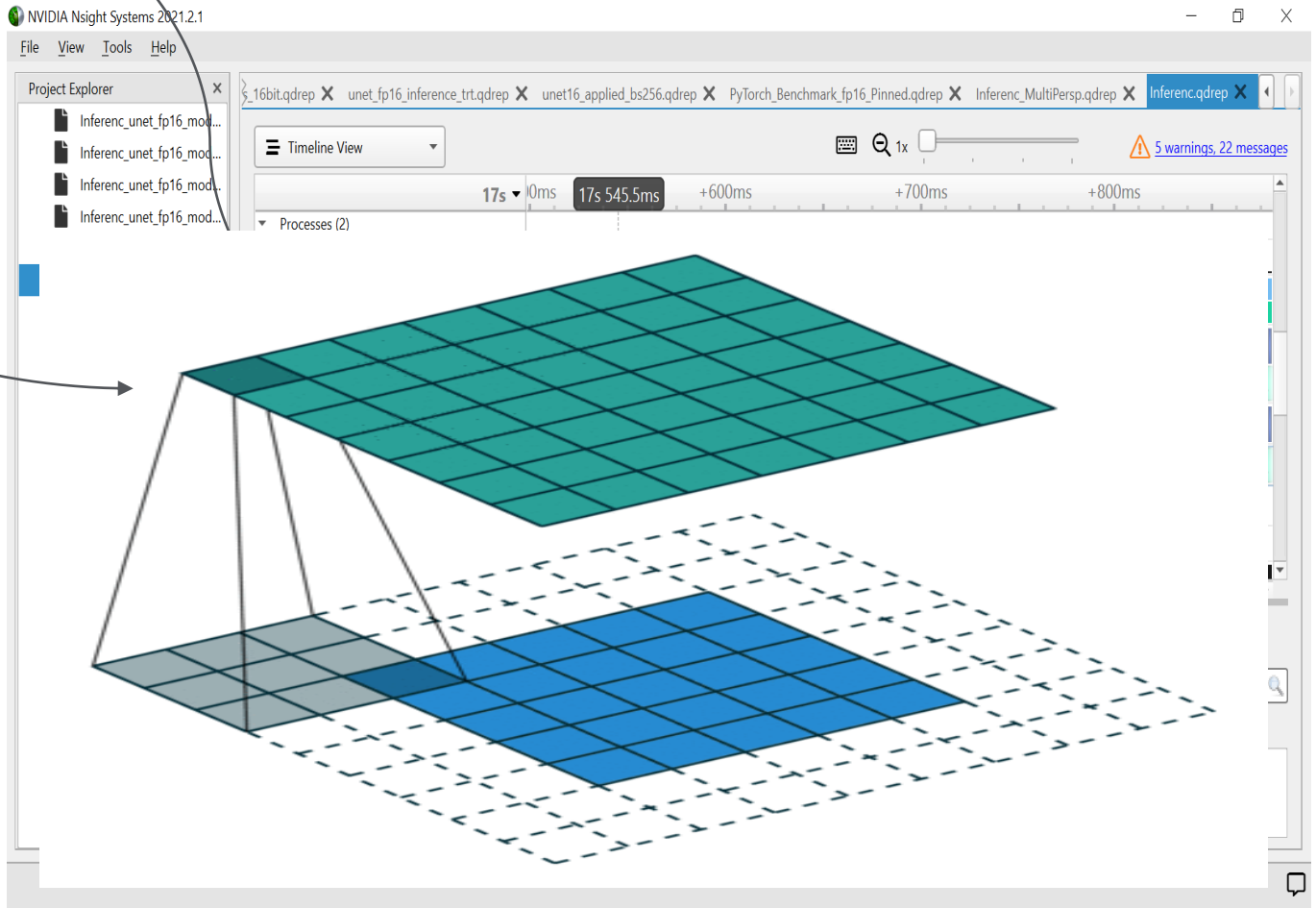
Solution 1:

- Overlap memory with processing on GPU improve the x1.3 (save IO)
  - » Apply NVIDIA stream's infra-structure
  - » Optimize the convolution layers to better adapt the general Matrix multiplication (GEMM) which is not convolved (not suitable directly for convolution operation)





Source:www.nvidia.com

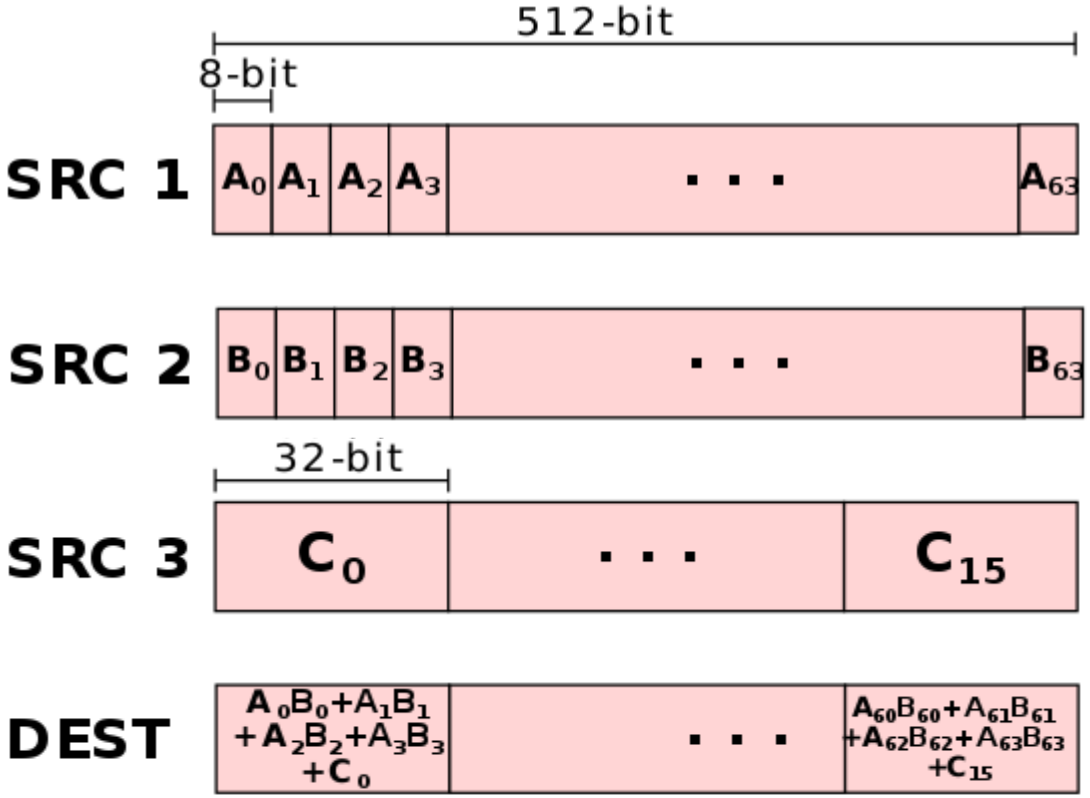


Innovated Optimization Techniques for Processing Acceleration

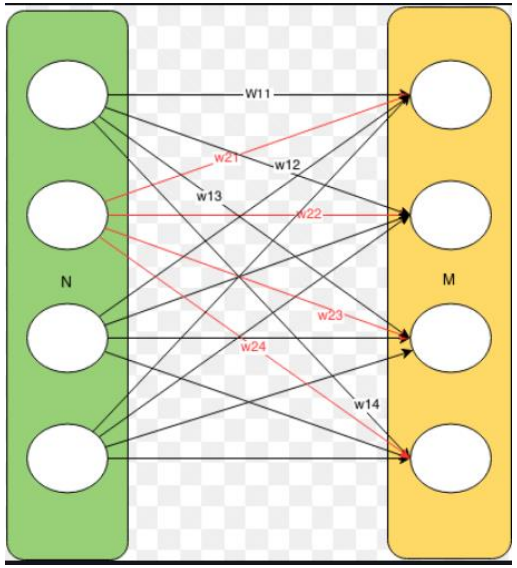
Optimized CPU cores using advanced VNNI-AVX512 instructions

Solution 2:

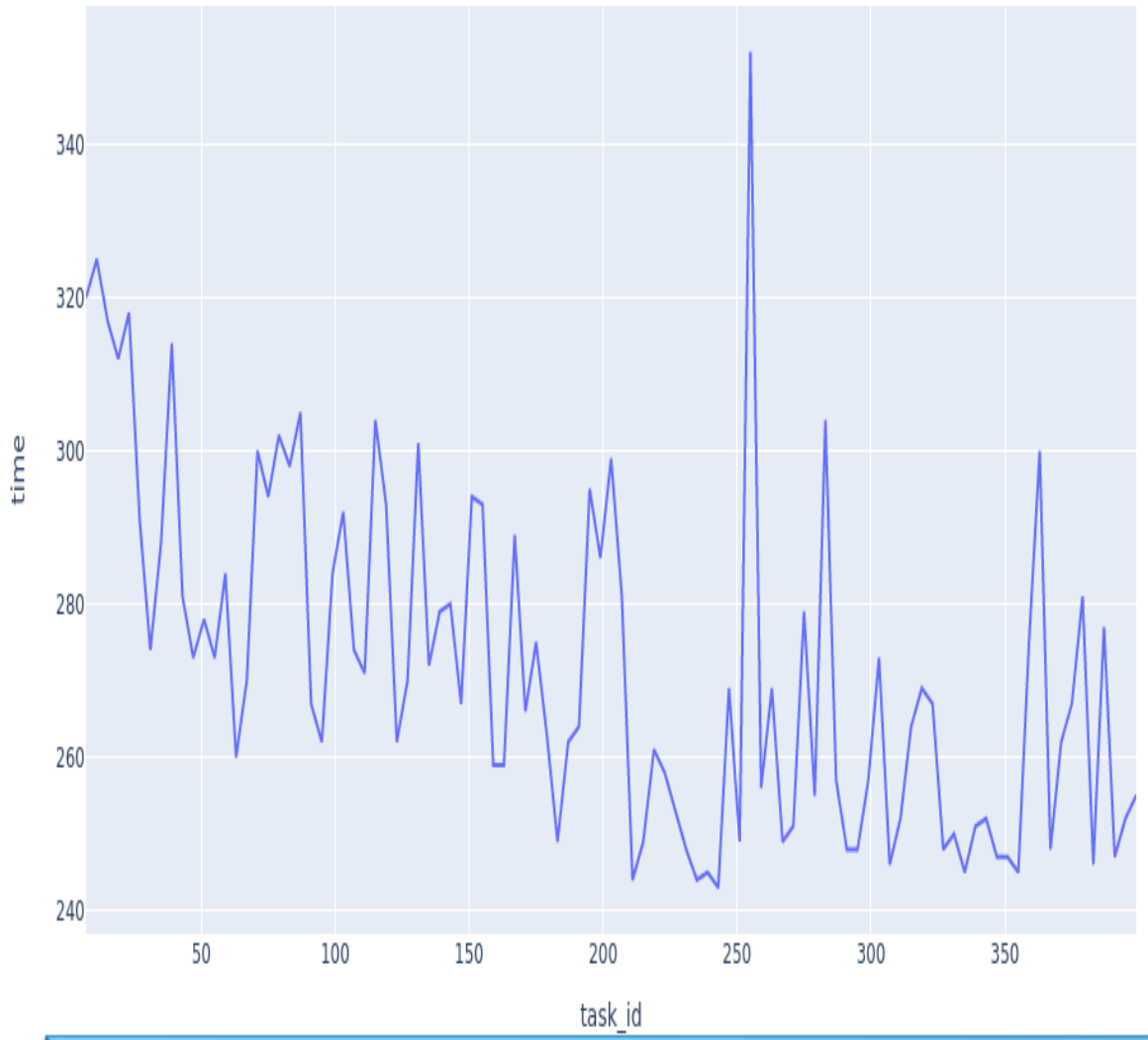
- VNNI-AVX-512 Vector Neural Network Instructions (VNNI) - x86
- Implement efficiently computer vision on server cores



$$A_0B_0+A_1B_1+A_2B_2+A_3B_3+C_0$$



Success story: Tensor Flow Open-Source Contribution



- Testing and benchmarking of combined processing mode over CPU + GPU
- Average processing min: 243ms, max: 352ms, mean: 272ms

- Finding BUG in Google Tensor Flow memory management

Gal Popovich 4:36 PM
https://github.com/tensorflow/tensorflow/issues/35524
open bug for tensorflow (google)
memory leak when loading model used as a reference for other bugs developers opened.
most likely helped to fix the bug in the new tensorflow version 2.2

Suspected memory leak - when loading multiple models wit...
Please make sure that this is a bug. As per our GitHub Policy, we only address code/doc bugs, performance issues, feature requests and...
github.com

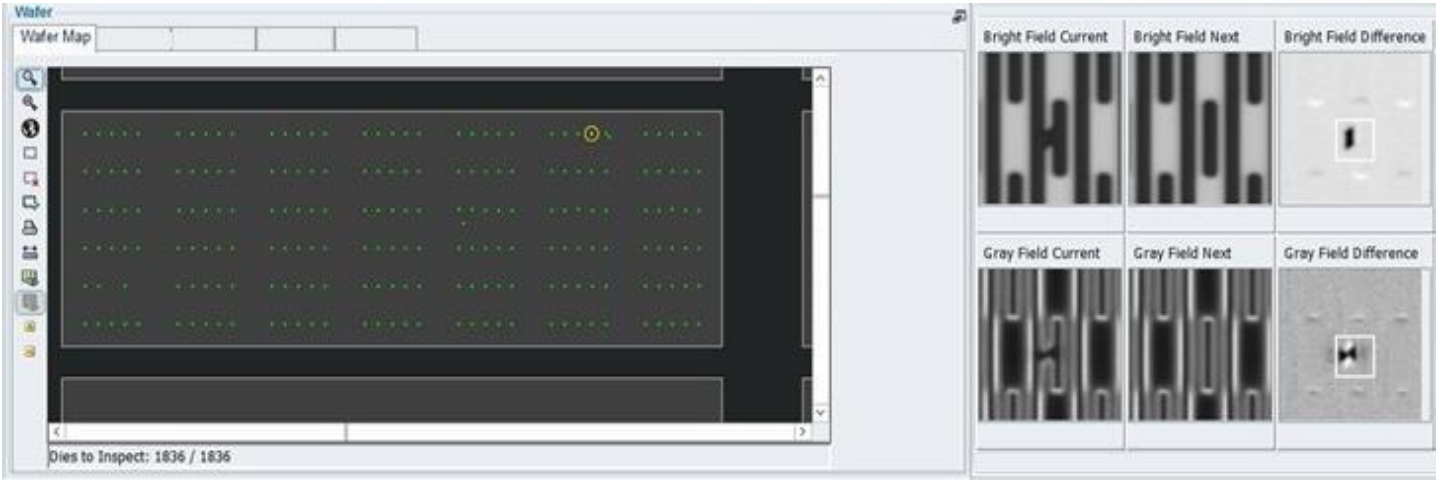
https://github.com/tensorflow/tensorflow/issues/35524

Solving a bug found by AMIL enabled efficient CPU+GPU hybrid processing combination data set

Summary

- Adapting OTS components for the next generation ODWI yielded:
  - » Highly cost-effective solution
  - » Perform advanced & complex algorithms
  - » Grabbing very high data rates for serving real-time requirements
- 'Know your architecture' approach improved the inspection computing performance by a significant factor of X10 by:
  - » Algorithm to hardware components optimizations (e.g. GEMM)
  - » Memory configuration optimizations (large DDR page & TLB)
- Madein4 development can be exploited for computing intensive tools

Source: Prodirve



Acknowledgment

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826589. The JU receives support from the European Union's Horizon 2020 research and innovation programs and Netherlands, Belgium, Germany, France, Italy, Austria, Hungary, Romania, Sweden and Israel.

The authors would like to thank Prodrive for a great collaboration as part of MADEin4 project

Thank you for your attention