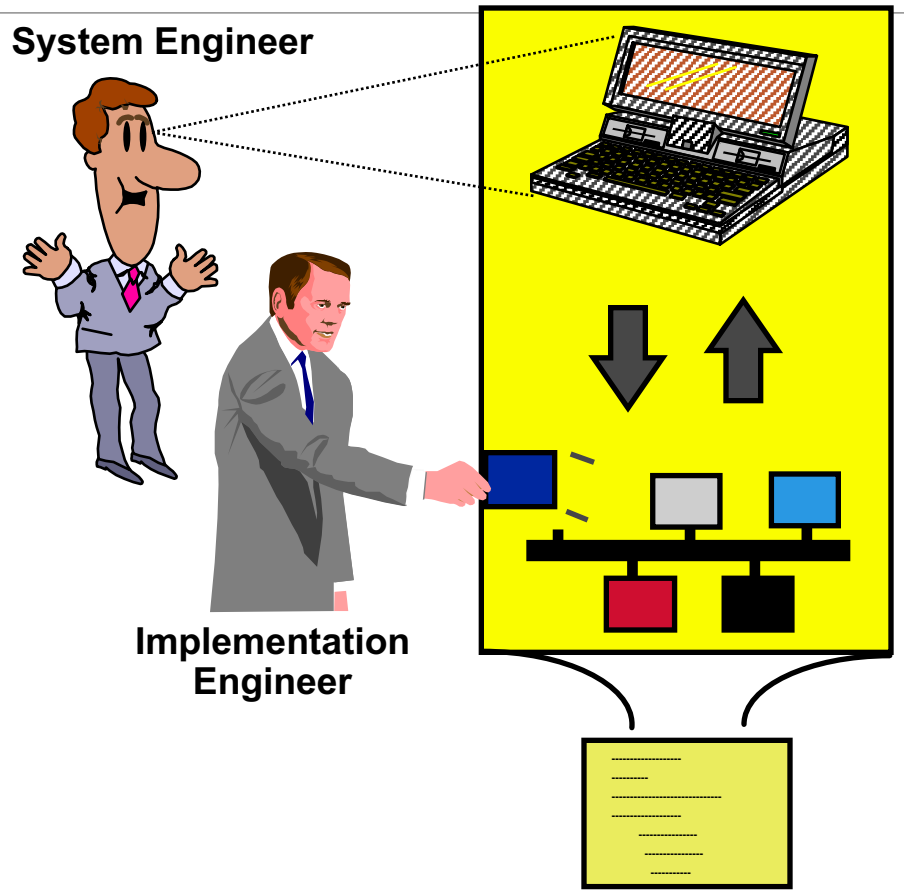
MIRABILIS
design

Using System-Level Architecture Exploration for Chiplet SoC

Deepak Shankar
Chief Technologist
Mirabilis Design Inc.

What is System Level Design



System Design Focuses on:

Designing the *Right Product*

- System decisions are optimized, repeatable and linked to implementation
- Robust system design can survive implementation compromises and get to market faster
- **Mirabilis Design** is focused here while EDA companies are planning to move here

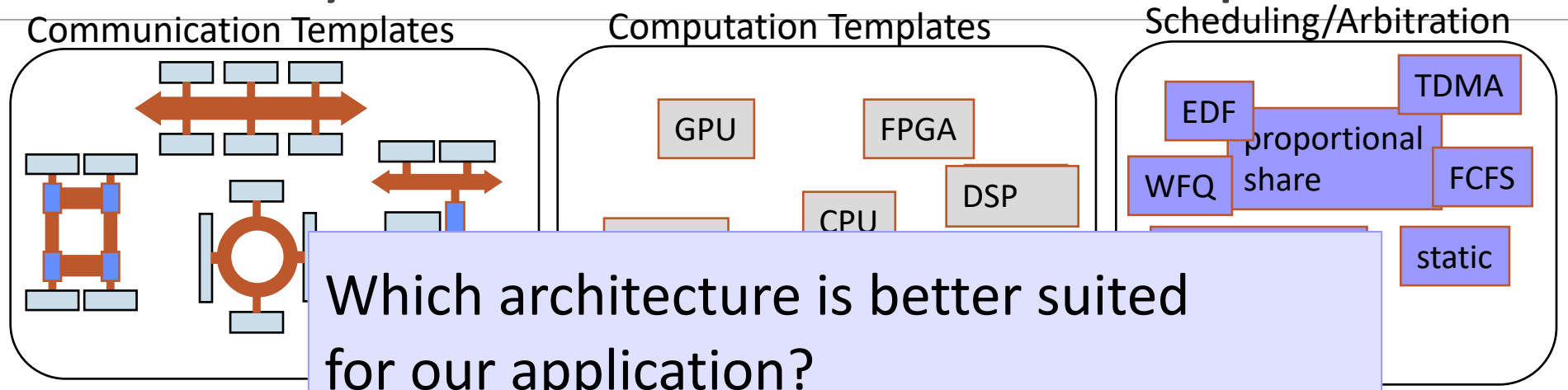
Implementation Focuses on:

Implementing the *Product Right*

- Perfect implementation cannot rescue product from bad design assumptions
- Historical EDA Companies focus

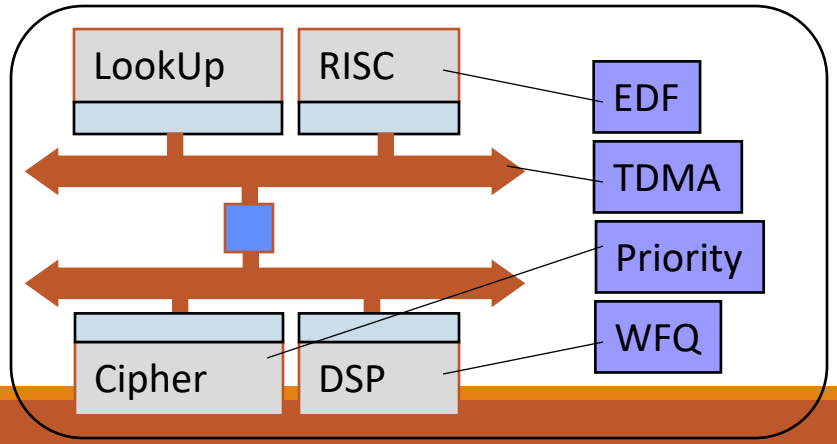
Escalating complexity means increasing need for system design

What is System Architecture Exploration?

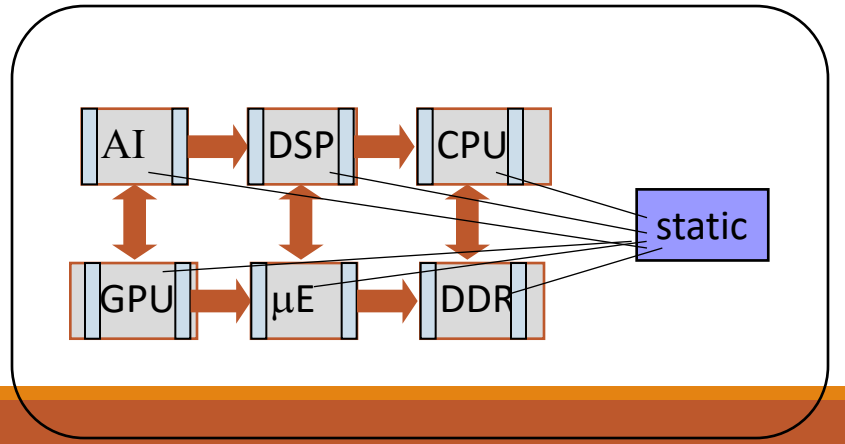


Which architecture is better suited for our application?

Architecture # 1

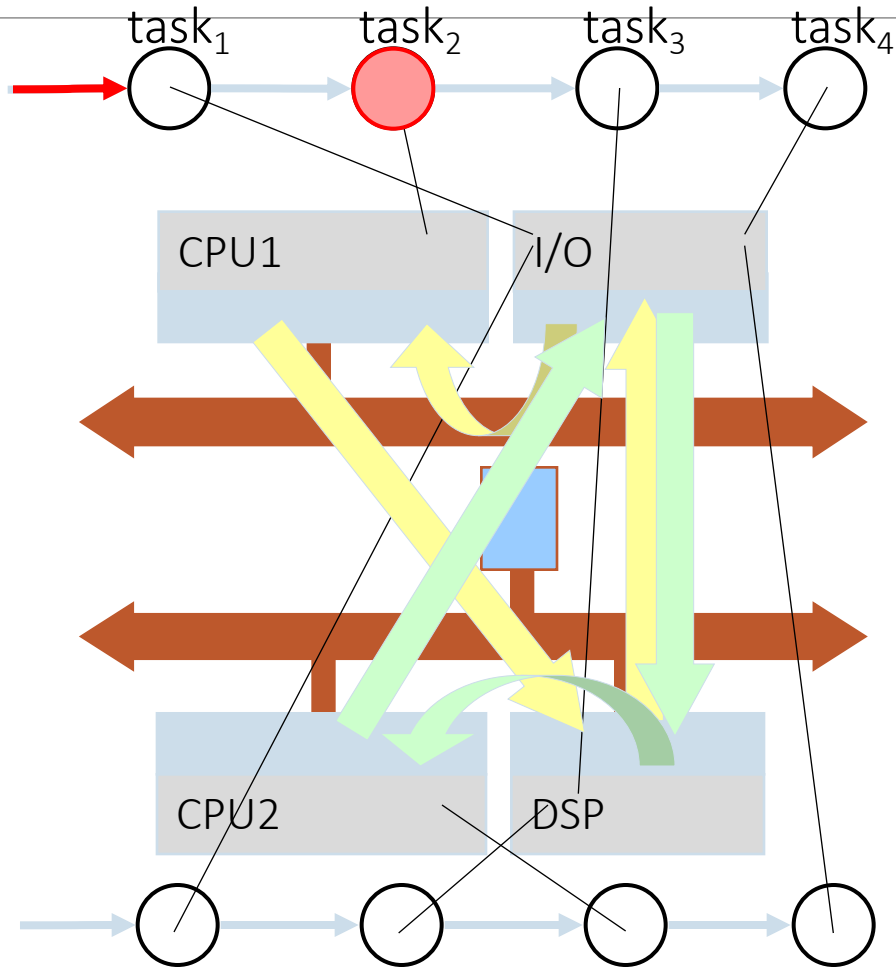


Architecture # 2



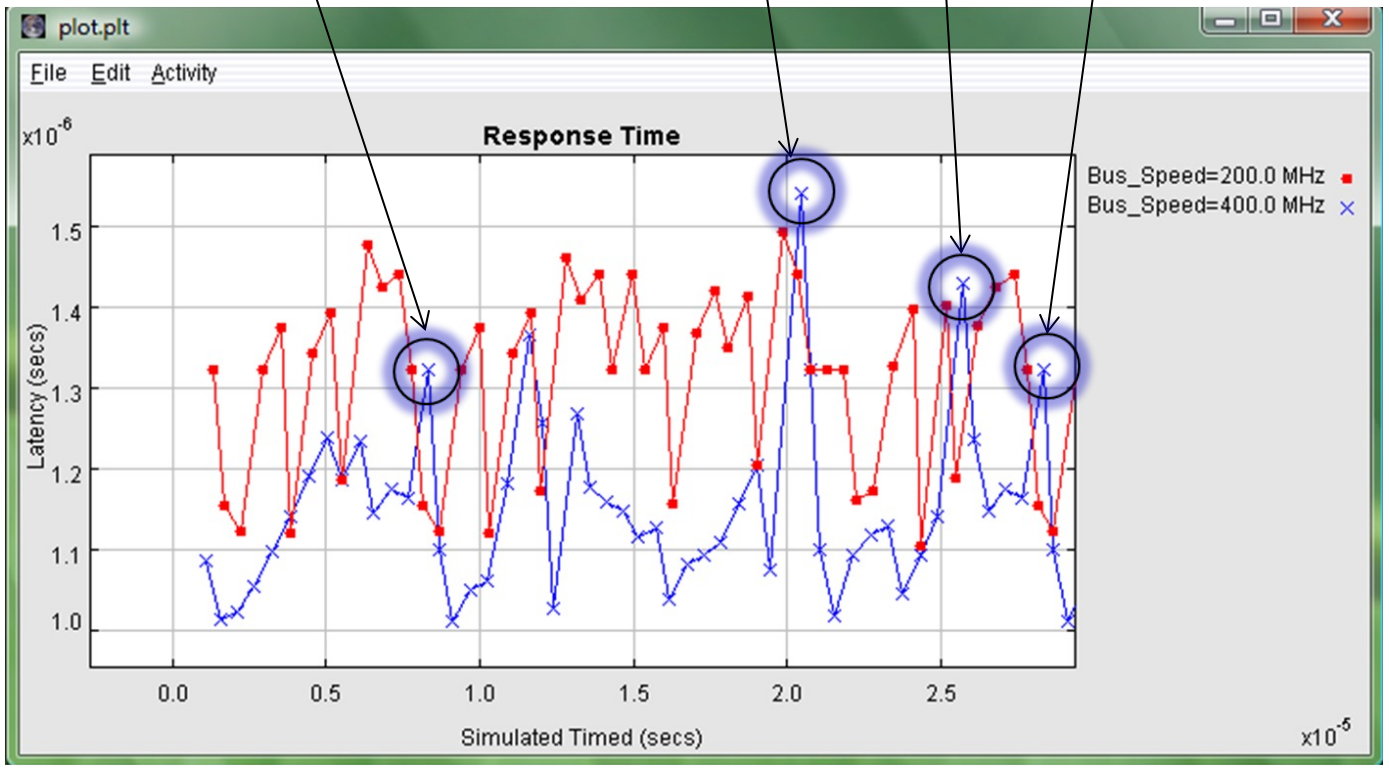
Motivation for Analyze and Validate with VisualSim

- Complex behavior
  - input stream
  - data dependent behavior
- Contention
  - limited resources
  - scheduling/arbitration
- Interference of multiple applications
  - limited resources
  - scheduling/arbitration
  - anomalies



Justification for System-level Model

System with faster Bus is slower in places



Unpredictable System Response

Design Challenges in Implementing UCIe

- Large memory transaction can block a high priority control access
  - For time critical application, these situations are not desirable
  - Example : Automotive communication system
- Inter-chiplets and intra-chiplet integration must support the new applications
  - Resource sizing of the interconnects is required to maximize bandwidth usage
  - Example applications : Data Center and AI Accelerators
- Migrating from monolithic die to Chiplet requires new memory allocation strategies
  - Limited memory needs to be partitioned across dies for both coherent and non-coherent operations
  - Example: Apple M1 Ultra uses Chiplets to double the performance

System Modeling in Chiplet Architecture Trade-offs

UCIe and inter-chiplet communication use detailed traffic and task graph to study the bandwidth consumption

Overhead associate with streaming and packet protocols will have an impact on the interconnect

System settings such as packet size, virtual channels, scheduling and frequency/size of requests determine the system configuration and the power consumed

Integration of requirements with the system studies enables optimization of the specification

Defining the power management units

Use-Case Example to Understand the Technical Challenges

- A chiplet has two CXL stacks sharing the physical link.
- Arbiter across the Die-to-Die adapter must send Flits alternatively between the 2 protocols.
  - If one of the Protocol layers doesn't have data to transmit, then instead of payload, "NOP" frames are inserted. If one of the Protocol stacks is idle for most of the time, then bandwidth could essentially be wasted on the "NOP" frames.
- Increasing the number of modules for either the standard or advanced package provides more bandwidth.
  - But is that extra bandwidth needed for the application?
- What happens if multiple chiplets in your design require the data stored at the same address location which is in another chiplet?
  - Consider the impact of cache coherency
- Can peak throughput be guaranteed for your application in a shared resource environment?
  - CPU is on one die and controls the execution, while the AI Engine is on a different die, while memory controller is a different die

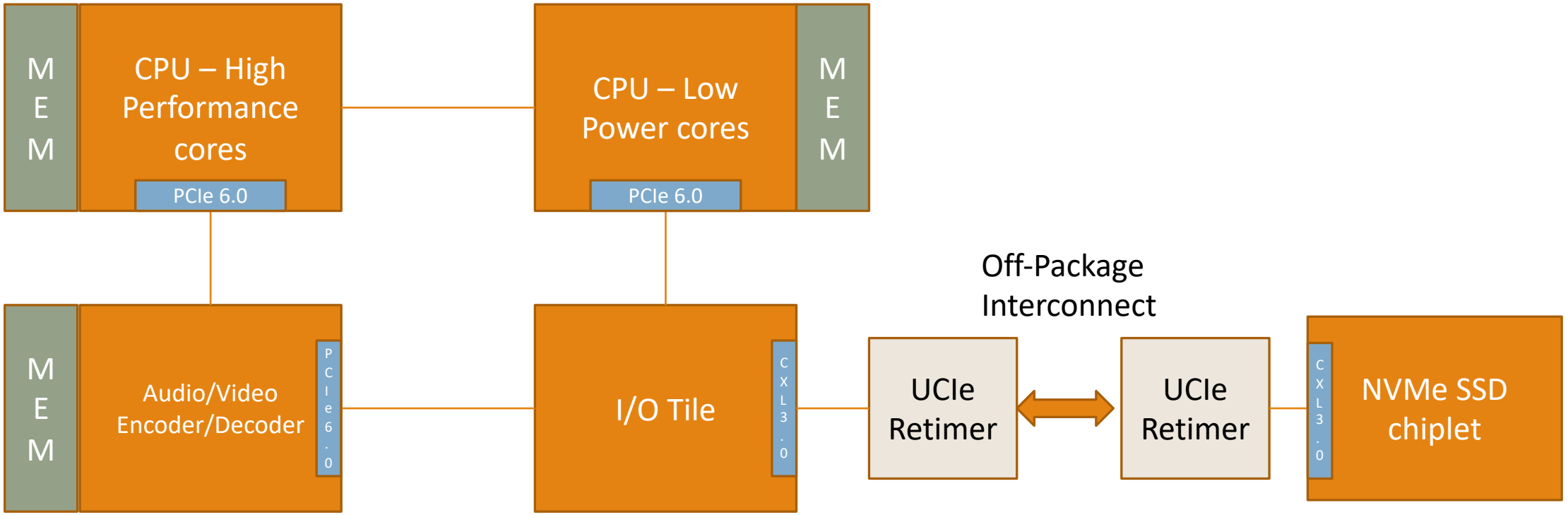
MIRABILIS
design

How does System Modeling of UCIe based multi-die SoC work?

Multi-Media Application – UCIe Template provided by Intel

Do we need a multi module setup?

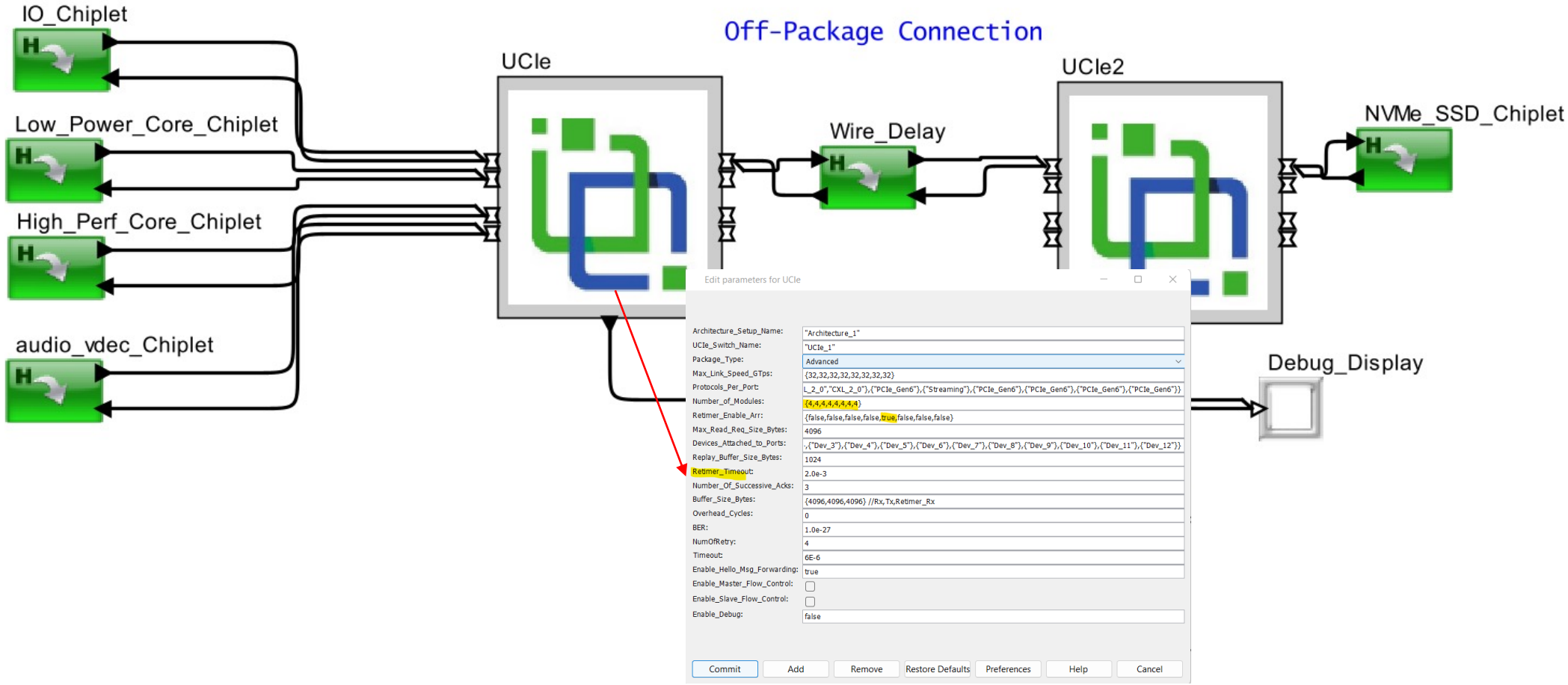
How much should the transfer rate between UCIe links be set to? 4 GTs or 8 GTs … or 32 GTs?



How much should the retimer timeout be set to?

Start with a System Block Diagram

VisualSim Model



Create a VisualSim model using existing building blocks

Stats

Advanced package, 4 module, 32 GT/s config

VisualSim Architect - .UCIe_Demo_Retimer_3.Stats

```
UCIe_Switch_UCIe_1_Port_1_Tx_MBps        = 362.0240000000018,
UCIe_Switch_UCIe_1_Port_1_to_Port_5_Max_Latency  = 2.0953814100499E-9,
UCIe_Switch_UCIe_1_Port_1_to_Port_5_Mean_Latency         = 1.225212114076
UCIe_Switch_UCIe_1_Port_1_to_Port_5_Min_Latency  = 1.2207031297243E-9,
UCIe_Switch_UCIe_1_Port_2_Drop_Count     = 0,
UCIe_Switch_UCIe_1_Port_2_Rx_MBps        = 22.6500000000001,
UCIe_Switch_UCIe_1_Port_2_Total_MBps     = 120.5280000000006,
UCIe_Switch_UCIe_1_Port_2_Tx_MBps        = 97.8780000000005,
UCIe_Switch_UCIe_1_Port_2_to_Port_5_Max_Latency  = 2.0734162000172E-9,
UCIe_Switch_UCIe_1_Port_2_to_Port_5_Mean_Latency         = 1.228556690107
UCIe_Switch_UCIe_1_Port_2_to_Port_5_Min_Latency  = 1.2207031297243E-9,
UCIe_Switch_UCIe_1_Port_3_Drop_Count     = 0,
UCIe_Switch_UCIe_1_Port_3_Rx_MBps        = 262.1720000000013,
UCIe_Switch_UCIe_1_Port_3_Total_MBps     = 1245.0320000000063,
UCIe_Switch_UCIe_1_Port_3_Tx_MBps        = 982.8600000000049,
UCIe_Switch_UCIe_1_Port_3_to_Port_5_Max_Latency  = 1.7893030700763E-9,
UCIe_Switch_UCIe_1_Port_3_to_Port_5_Mean_Latency         = 1.220923243569
UCIe_Switch_UCIe_1_Port_3_to_Port_5_Min_Latency  = 1.2147207601766E-9,
UCIe_Switch_UCIe_1_Port_4_Drop_Count     = 0,
UCIe_Switch_UCIe_1_Port_4_Rx_MBps        = 4.468,
UCIe_Switch_UCIe_1_Port_4_Total_MBps     = 27.2880000000001,
UCIe_Switch_UCIe_1_Port_4_Tx_MBps        = 22.8200000000001,
UCIe_Switch_UCIe_1_Port_4_to_Port_5_Max_Latency  = 4.5410156299282E-9,
UCIe_Switch_UCIe_1_Port_4_to_Port_5_Mean_Latency         = 2.785317100585
UCIe_Switch_UCIe_1_Port_4_to_Port_5_Min_Latency  = 1.2207031299411E-9,
UCIe_Switch_UCIe_1_Port_5_Drop_Count     = 0,
UCIe_Switch_UCIe_1_Port_5_Rx_MBps        = 1465.6380000000074,
UCIe_Switch_UCIe_1_Port_5_Total_MBps     = 1846.8000000000093,
UCIe_Switch_UCIe_1_Port_5_Tx_MBps        = 381.1620000000019,
UCIe_Switch_UCIe_1_Port_5_to_Port_1_Max_Latency  = 1.220703130158E-9,
UCIe_Switch_UCIe_1_Port_5_to_Port_1_Mean_Latency         = 1.220703129998
UCIe_Switch_UCIe_1_Port_5_to_Port_1_Min_Latency  = 1.2207031297243E-9,
```

~300x latency difference can be observed. However, for non-time critical applications, Standard UCIe package option looks attractive

Standard package, Single module, 4 GT/s config

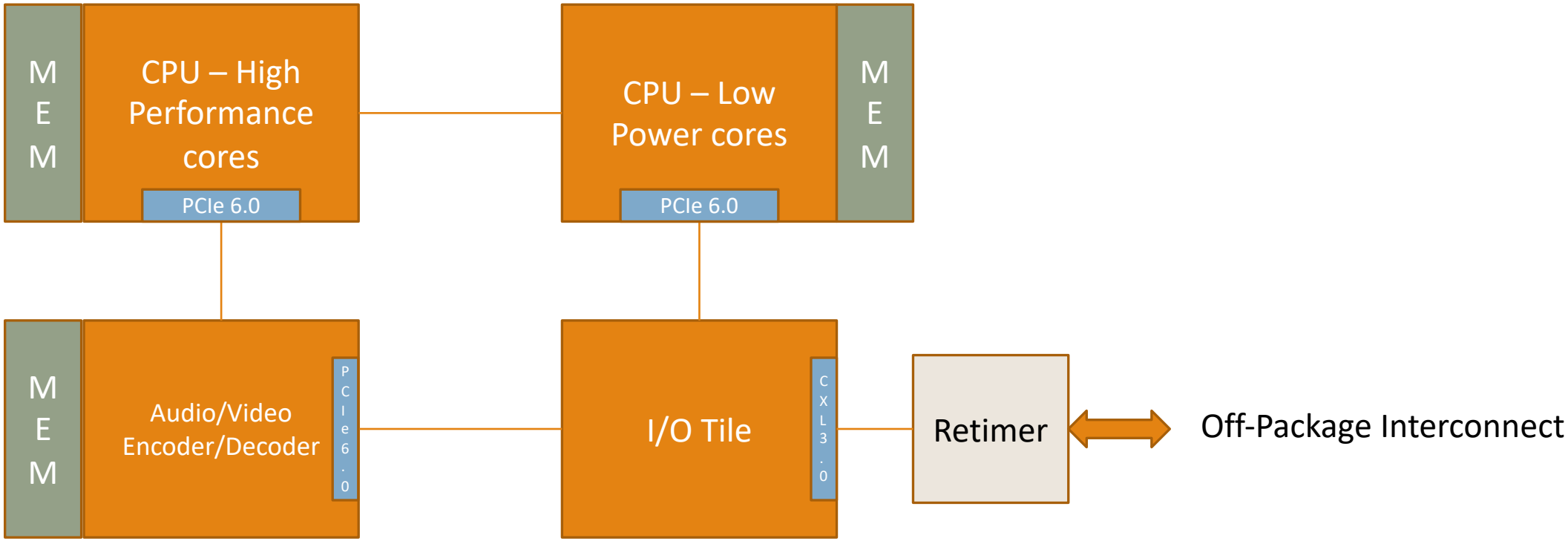
VisualSim Architect - .UCIe_Demo_Retimer_4.Stats

```
UCIe_Switch_UCIe_1_Port_1_Tx_MBps        = 354.5260000000018,
UCIe_Switch_UCIe_1_Port_1_to_Port_5_Max_Latency  = 4.9868349838987E-7,
UCIe_Switch_UCIe_1_Port_1_to_Port_5_Mean_Latency         = 1.777048657484E-7,
UCIe_Switch_UCIe_1_Port_1_to_Port_5_Min_Latency  = 1.4874999999977E-7,
UCIe_Switch_UCIe_1_Port_2_Drop_Count     = 0,
UCIe_Switch_UCIe_1_Port_2_Rx_MBps        = 23.3580000000001,
UCIe_Switch_UCIe_1_Port_2_Total_MBps     = 120.5280000000006,
UCIe_Switch_UCIe_1_Port_2_Tx_MBps        = 97.1700000000005,
UCIe_Switch_UCIe_1_Port_2_to_Port_5_Max_Latency  = 3.6062531230999E-7,
UCIe_Switch_UCIe_1_Port_2_to_Port_5_Mean_Latency         = 1.8548214682754E-7,
UCIe_Switch_UCIe_1_Port_2_to_Port_5_Min_Latency  = 1.3556449996E-7,
UCIe_Switch_UCIe_1_Port_3_Drop_Count     = 0,
UCIe_Switch_UCIe_1_Port_3_Rx_MBps        = 247.9180000000012,
UCIe_Switch_UCIe_1_Port_3_Total_MBps     = 1206.848000000006,
UCIe_Switch_UCIe_1_Port_3_Tx_MBps        = 958.9300000000048,
UCIe_Switch_UCIe_1_Port_3_to_Port_5_Max_Latency  = 4.378956127201E-7,
UCIe_Switch_UCIe_1_Port_3_to_Port_5_Mean_Latency         = 1.8992489593206E-7,
UCIe_Switch_UCIe_1_Port_3_to_Port_5_Min_Latency  = 1.2907678362001E-7,
UCIe_Switch_UCIe_1_Port_4_Drop_Count     = 0,
UCIe_Switch_UCIe_1_Port_4_Rx_MBps        = 60.6680000000003,
UCIe_Switch_UCIe_1_Port_4_Total_MBps     = 81.1600000000004,
UCIe_Switch_UCIe_1_Port_4_Tx_MBps        = 20.4920000000001,
UCIe_Switch_UCIe_1_Port_4_to_Port_5_Max_Latency  = 1.71480219027E-6,
UCIe_Switch_UCIe_1_Port_4_to_Port_5_Mean_Latency         = 9.175198666788E-7,
UCIe_Switch_UCIe_1_Port_4_to_Port_5_Min_Latency  = 1.4874999999998E-7,
UCIe_Switch_UCIe_1_Port_5_Drop_Count     = 0,
UCIe_Switch_UCIe_1_Port_5_Rx_MBps        = 1431.046000000007,
UCIe_Switch_UCIe_1_Port_5_Total_MBps     = 1847.7080000000092,
UCIe_Switch_UCIe_1_Port_5_Tx_MBps        = 416.6620000000021,
UCIe_Switch_UCIe_1_Port_5_to_Port_1_Max_Latency  = 3.8781000000004E-7,
UCIe_Switch_UCIe_1_Port_5_to_Port_1_Mean_Latency         = 1.697507352307E-7,
UCIe_Switch_UCIe_1_Port_5_to_Port_1_Min_Latency  = 1.4874999999977E-7,
```

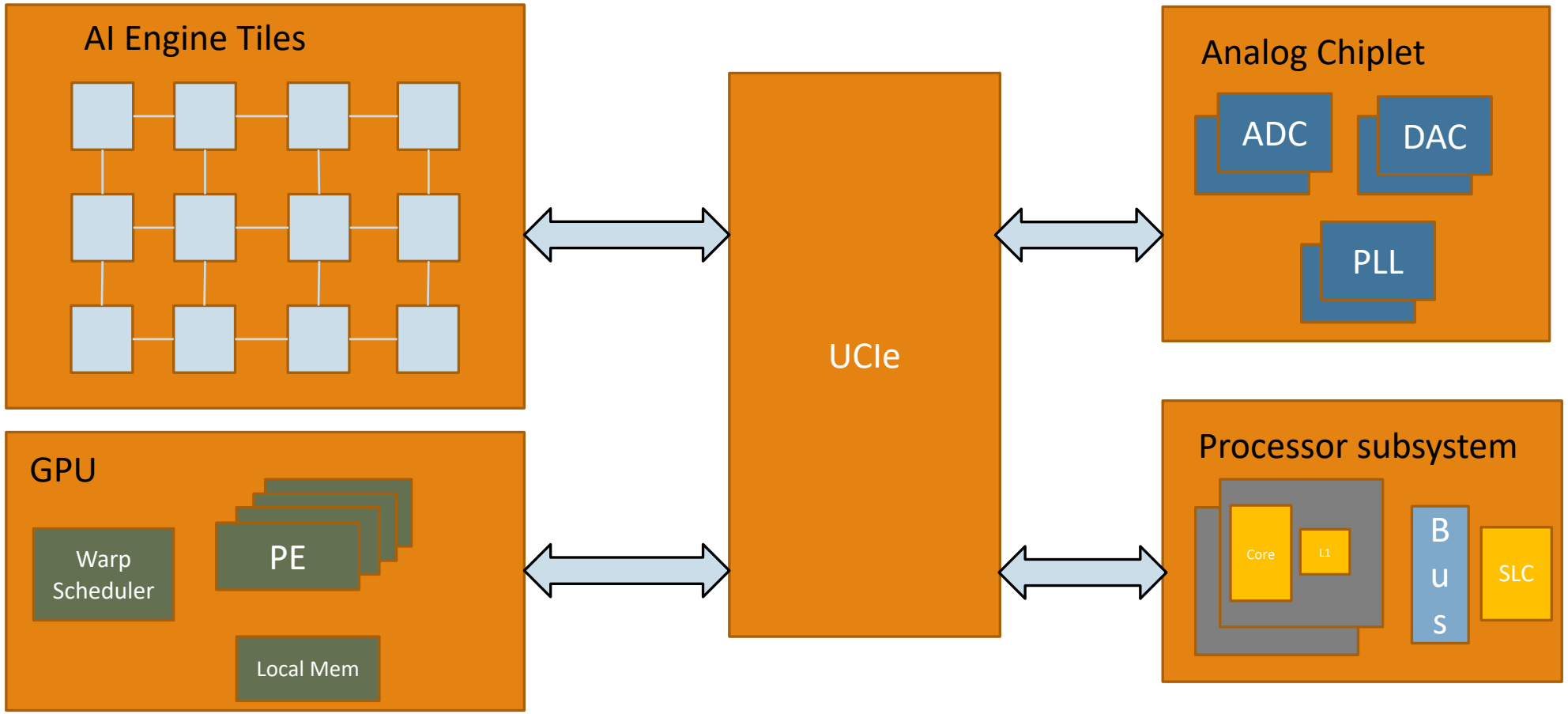
Study the statistics to decide on the best configuration

Application Examples of UCIe based multi-die SoC

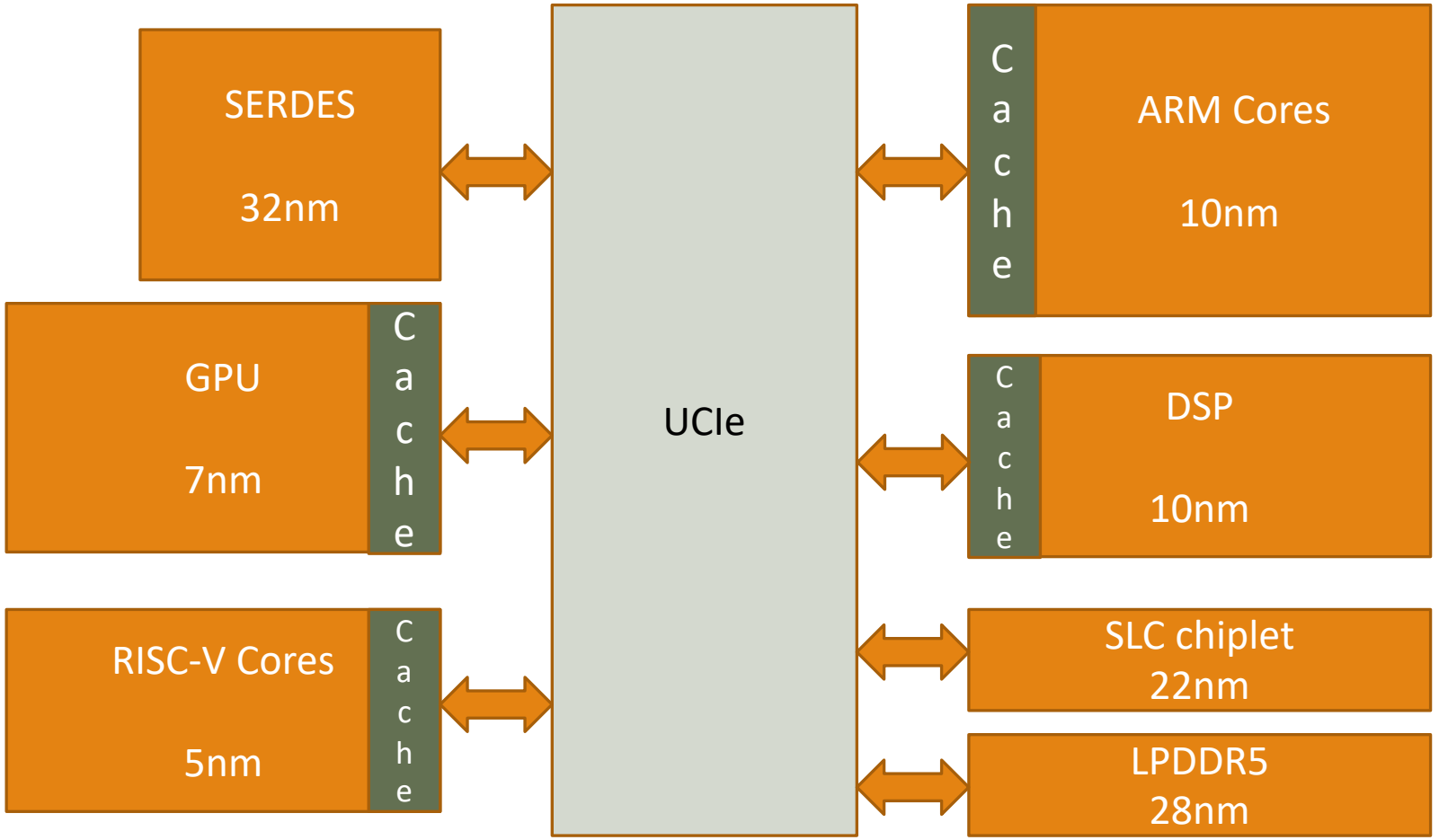
Example 1 – Multi-Media applications



Example 2 : Automotive Autonomous Driving



Example 3 : Cache Coherency using UCIe

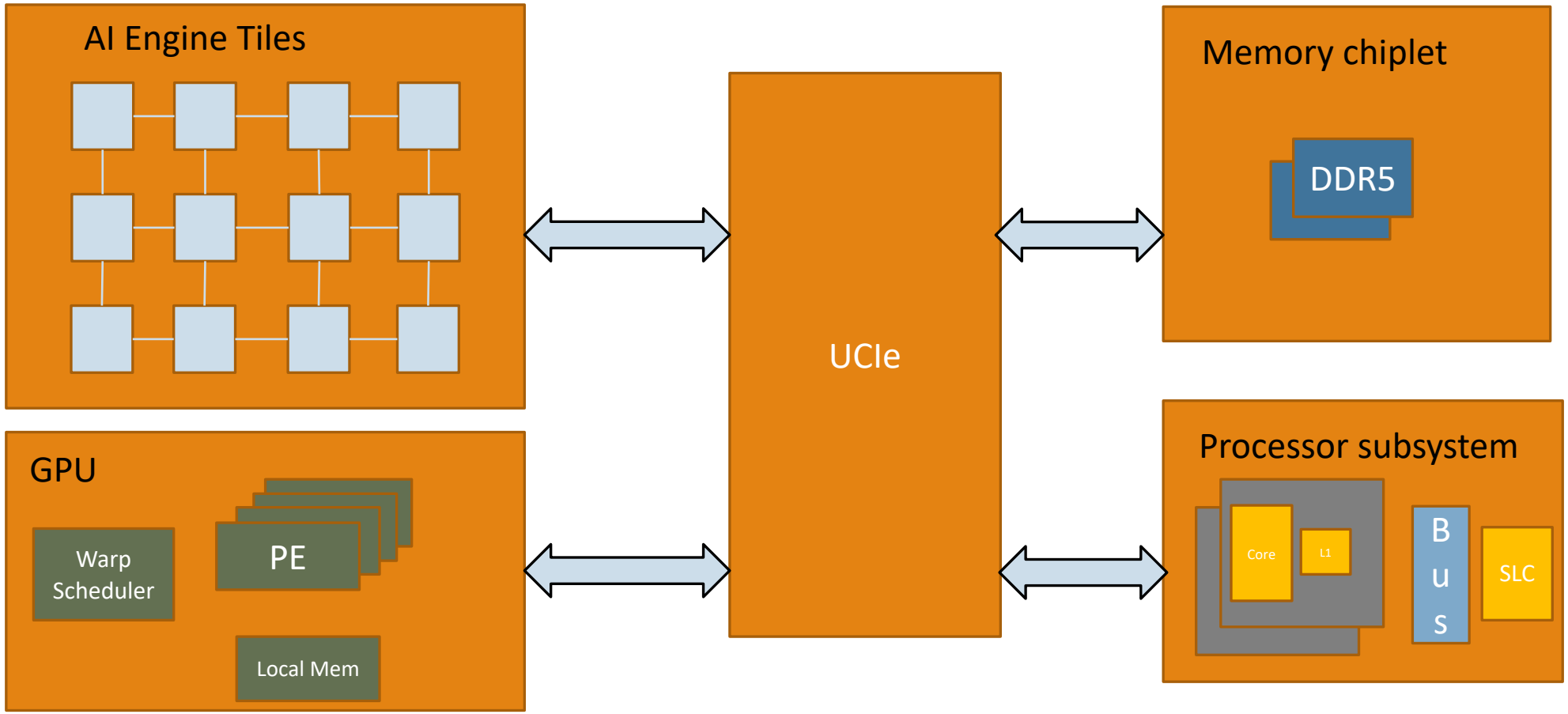


Analyzing UCIe based multi-die SoC using VisualSim System Model

Autonomous driving

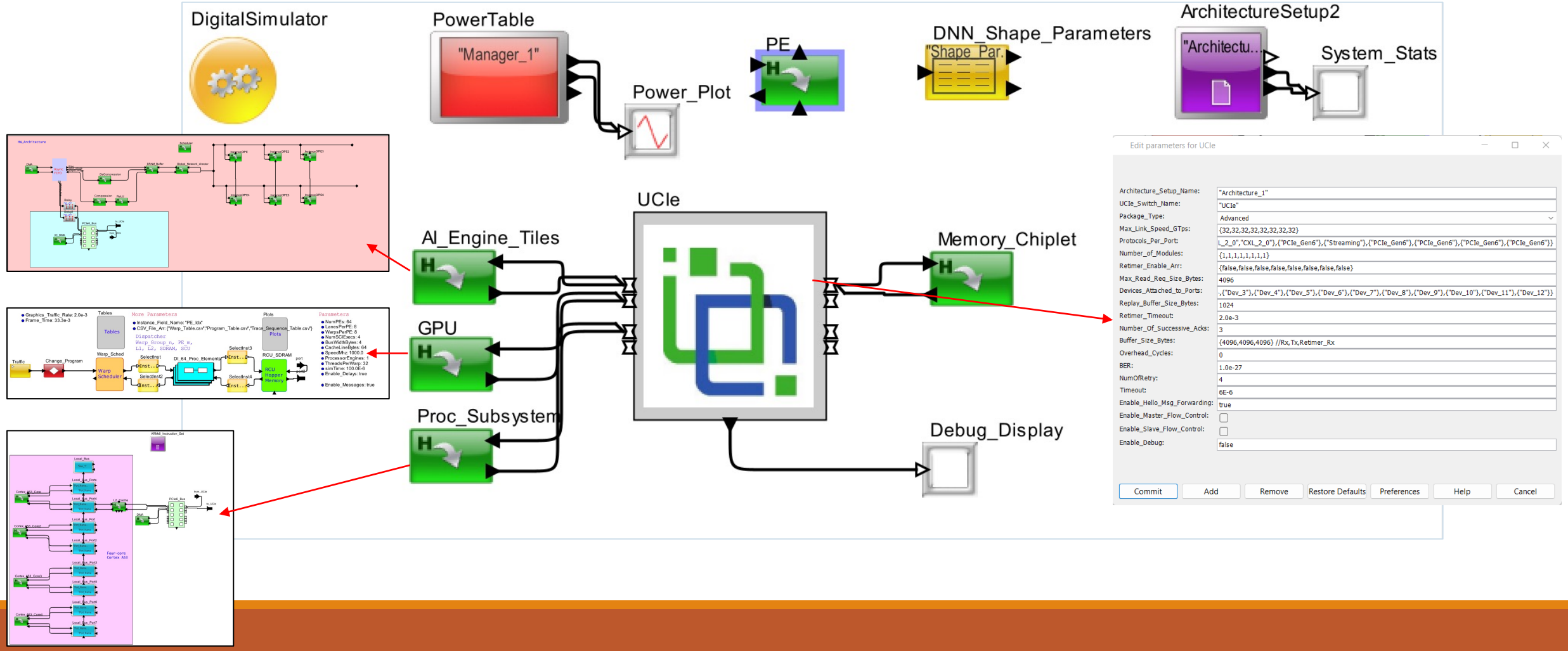
- Optimal mesh size (mxn) ?
- Best sample size (16 bytes vs 32 bytes etc) ?

Do we need PCIe gen6 or still use gen5 for meeting application requirements?

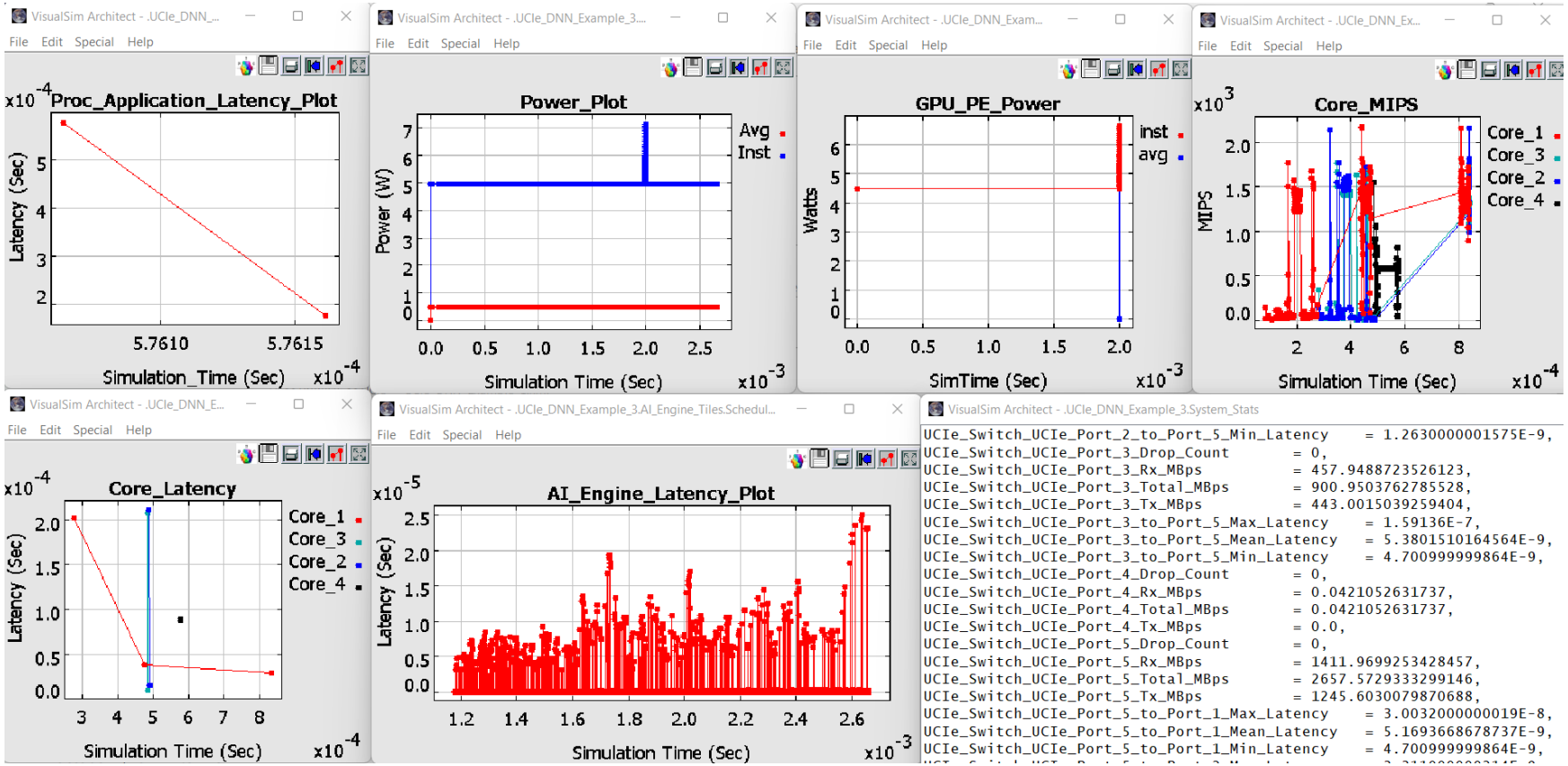


Use a single protocol stack or multi protocol stack?

VisualSim System Model using UCIe in ADAS SoC



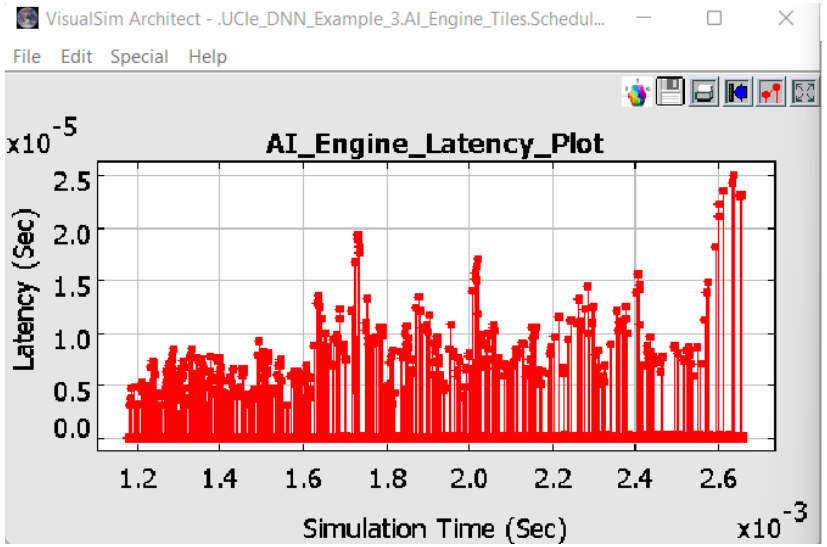
Statistics for Multi-Die SoC



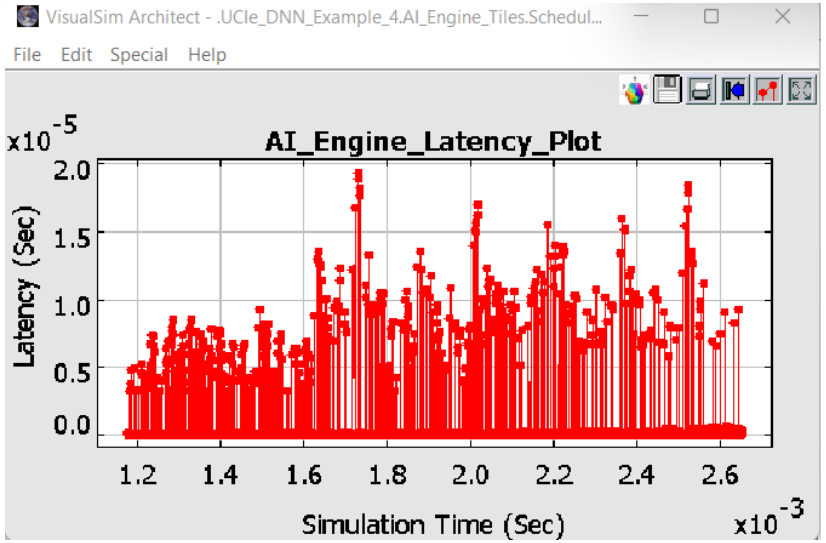
- Note the AI Engine latency spikes
- For multi protocol, half bandwidth for each protocol.
- Older gen protocols are mixed with PCIe 6,
- Lower FLIT size increases latency.

Comparing Different Configurations using UCIe Interface

Die Adapters use PCIe 6.0 and Streaming Protocols (AXI)



All Die Adapters use PCIe 6.0



Lower latency when using PCIe 6.0

MIRABILIS
design

Mirabilis Design
VisualSim Architect

About Mirabilis Design

Software Company with Key Development Team in India

Integrates Model-based Systems Engineering with the electronics development flow

Development and Support Centers

India, USA, South Korea, Japan and China

VisualSim Architect - Modeling and Simulation Software

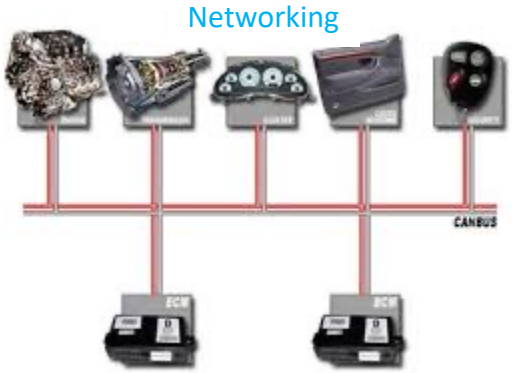
Graphical modeling, multi-domain simulator, system-level IP, analysis tools and open API

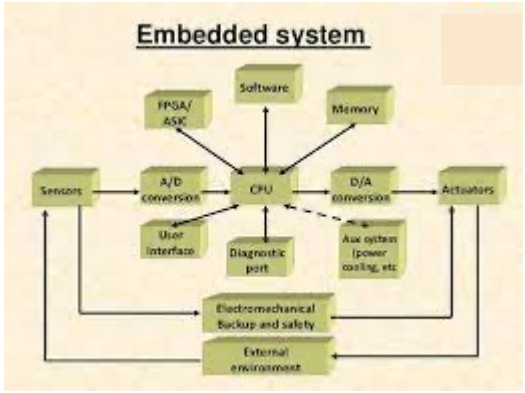
Market Segments

Semiconductors, Automotive and, Aerospace and Defense

Design Enablement

Architecture trade-offs, system validation, early functional testing and communication





What we provide……

Architecture Exploration
- IP blocks
- Semiconductors
- Networks
- Systems
- Software

To Trade-off
- Performance (Latency, Throughput)
- Power (Peak, Instant, Cumulative, Heat, Temp and Battery lifecycle)
- Functionality (algorithm, arbitration, scheduling, flow control)

To Make Decisions
- Product feasibility
- System sizing
- Mapping Task graphs to heterogeneous resources
- Hardware-software partitioning
- Generating documentation and models for sharing
- Test benches, design optimization and validation

Using
- System-level modelling IP and Generators
- Graphical, Hierarchical with Polymorphic types
- Multi-domain simulator
- AI-based multi-core diagnostic systems integrated with tracking
- Open API to integrate software code and third-party simulators

MIRABILIS design

MIRABILIS design

Evaluating UCIe based multi-die SoC to meet timing and power