Challenges and opportunities in Physical Design for Versal ACAP products

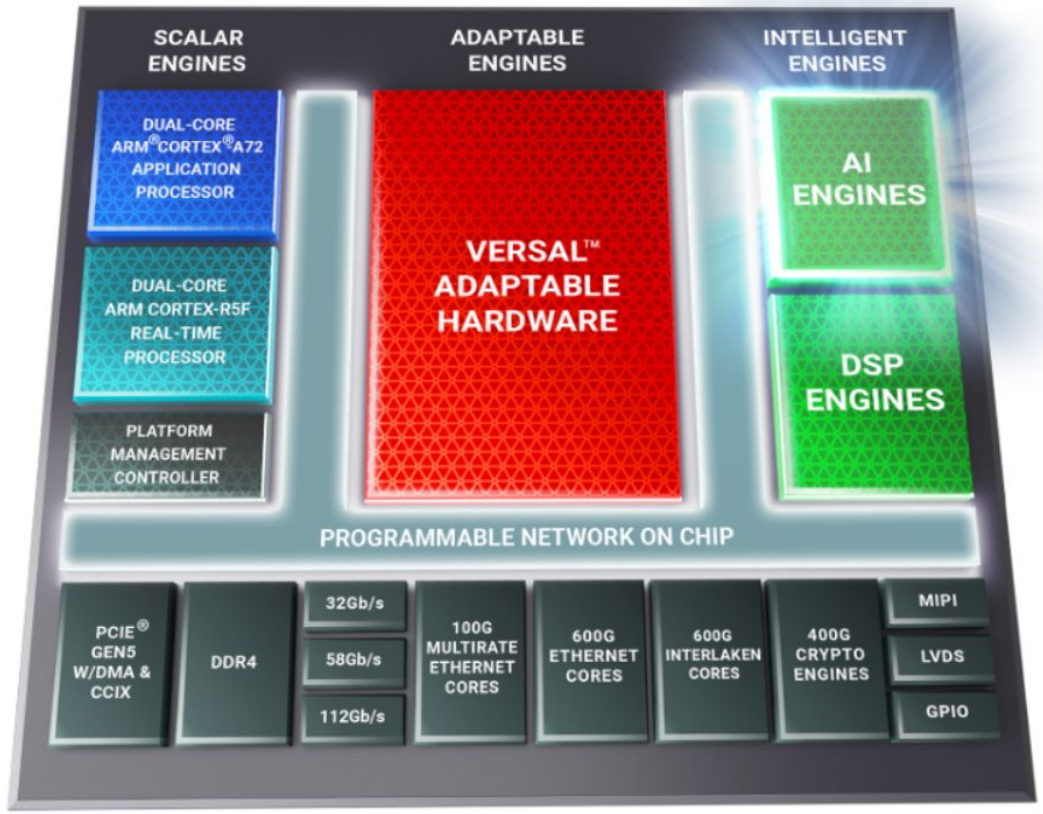
Amit Gupta – CVP, AMD Inc

What does an FPGA look like?

Versal Premium

Alveo V70

What does it look like under the lid?



Where does it get used?

AUTOMOTIVE
- Driver Assistance
- Surround Cameras & Radar
- Vehicle Infotainment

INDUSTRIAL & VISION
- Industrial Robotics
- SmartGrid, Renewables & EV Charging
- SmartCity, SmartRetail & SmartFactory

AEROSPACE & DEFENSE
- Commercial & Military Avionics
- Space & Military Satellite Communication
- Missiles & Munitions

HEALTHCARE & SCIENCES
- Medical Imaging & Ultrasound
- Surgical Robotics
- Clinical Medical (Patient Monitor)

DATA CENTER
- Live Video Transcoding at Cloud Scale
- Ultra Low Latency FinTech
- Board Management Controllers

TEST & EMULATION
- Memory & PCIe Testers
- Emulation Platforms
- Prototyping Platforms

WIRED COMMUNICATIONS
- Telecomm (Packet Networks, Cable Access)
- Network Security (Link Security & App Security)
- Telco Acceleration

PROAV & BROADCAST
- Video Walls, Projectors & Signage
- Pro Audio
- AV Routers/Switchers

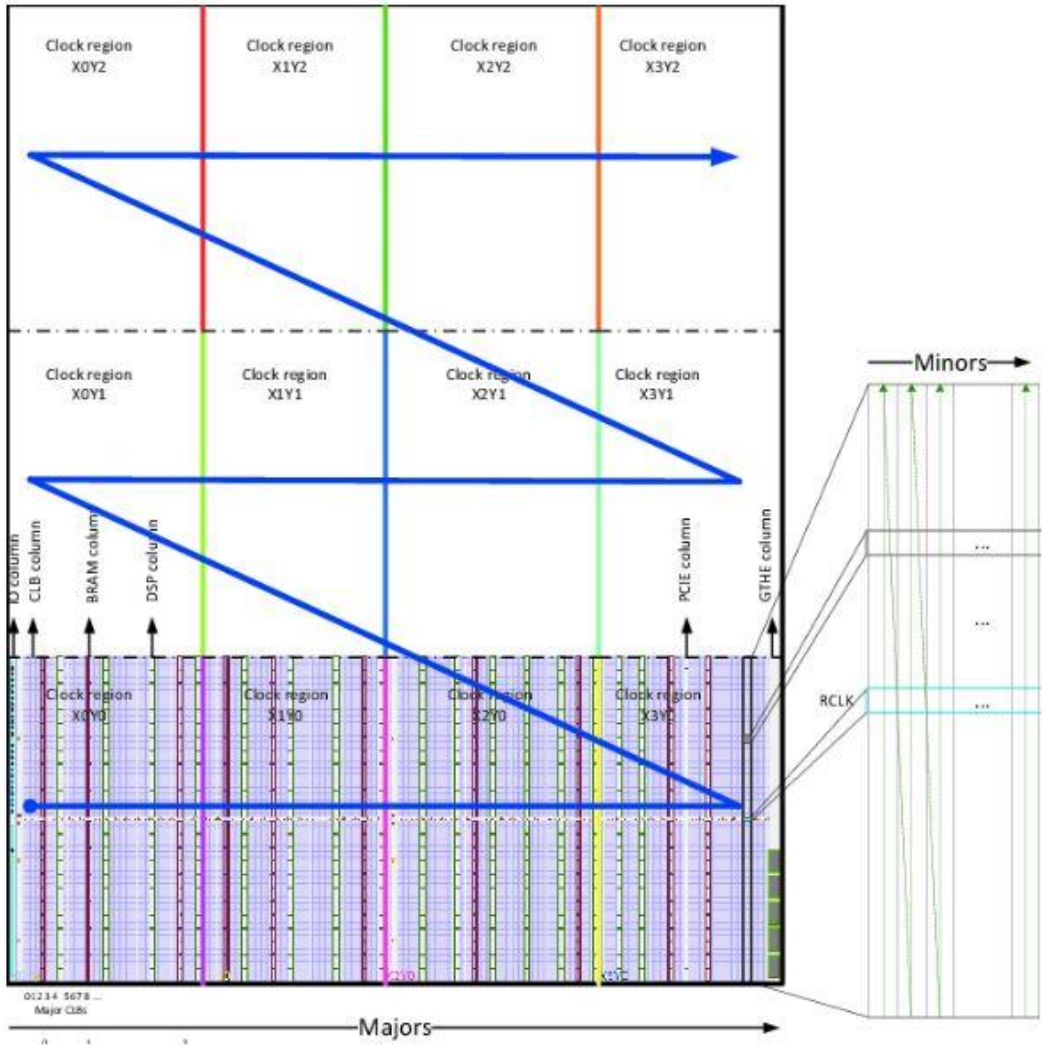
WIRELESS COMMUNICATIONS
- Remote Radio Units
- Beamforming & mMIMO
- xHaul & Connectivity

CONSUMER
- Televisions
- Drones
- Printers (3D, Multi-function, Portable)

FPGA CAD – Bird's-eye view

Placement in FPGA

- **Layout Site Grid**
  - Columns of Configurable logic blocks, memory blocks, and DSP
  - A CLB contains LUTs and Flops
- **Cell Library**
  - LUT (Lookup Table)
  - Flop
  - Carry (Lookahead)
  - DSP
  - BRAM
  - URAM



Overview of Kintex UltraScale XCKU025's device layout. This device has in total 3 repetitive resource rows and 12 clock regions. Figure adopted from Vivado Design Tool.

Interconnect/Routing in FPGA

- **WS: W**ire **S**egment (short/long)
- **PIP: P**rogrammable **I**nterconnect **P**oints
- **SM: S**witch **M**atrix
- A routed path: **WS** are connected if the **PIP** between them are configured as "**on**".

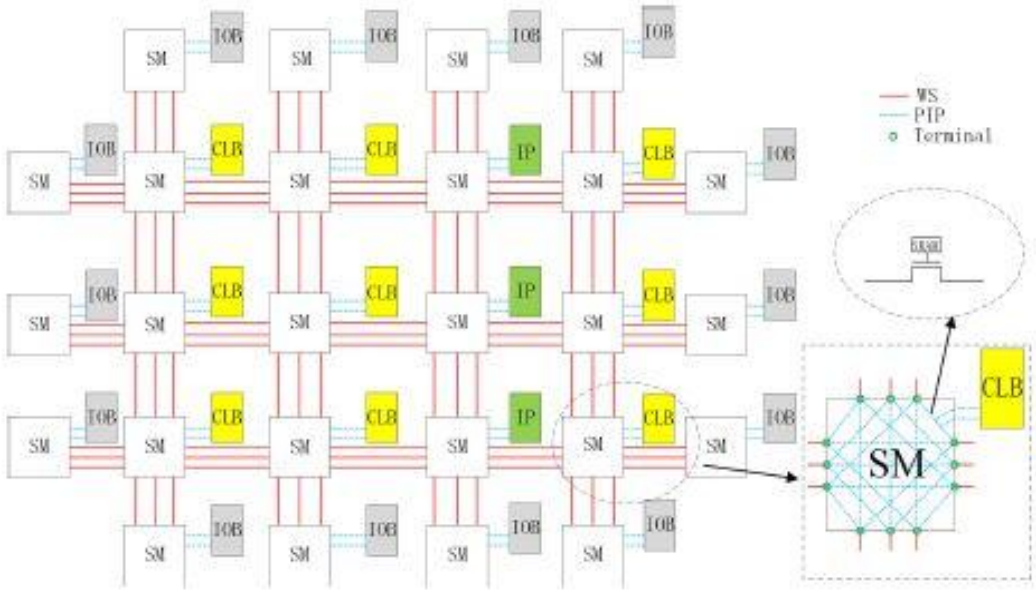
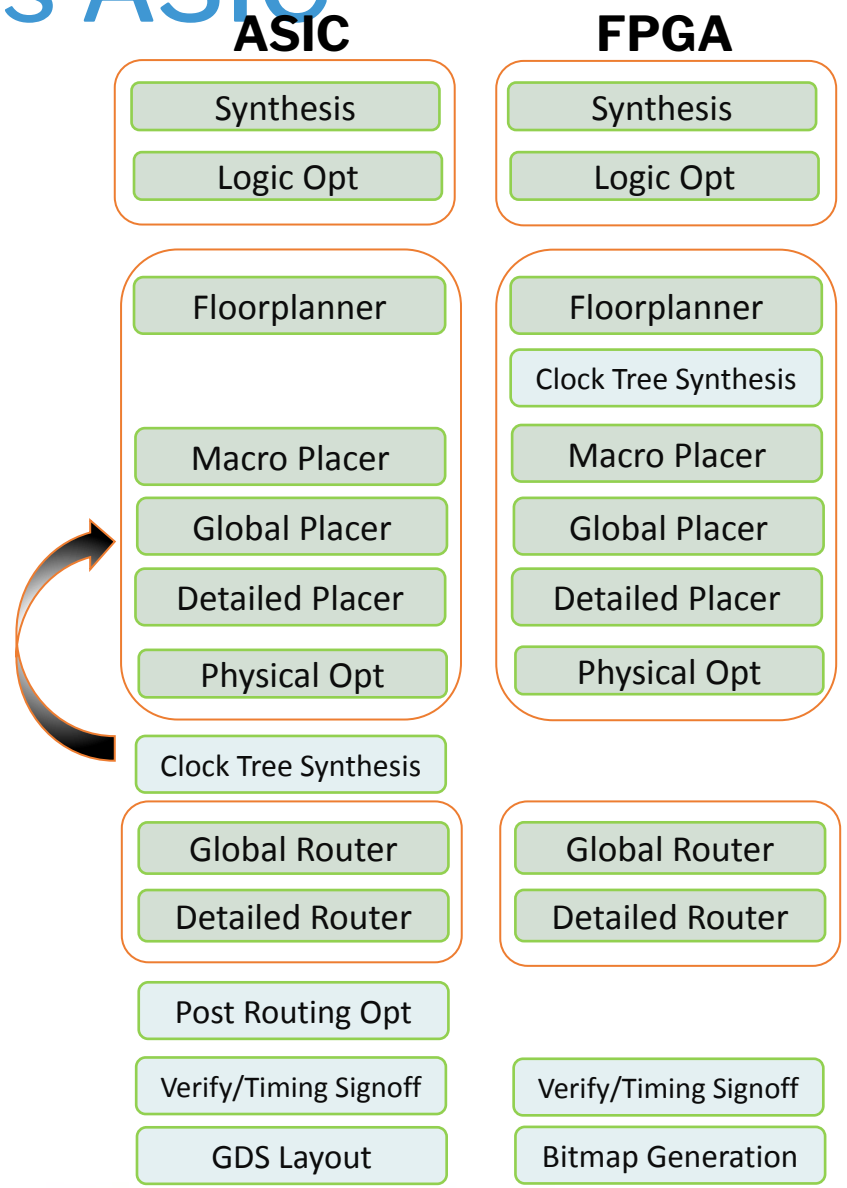


Fig. 1. SRAM-based FPGA architecture.

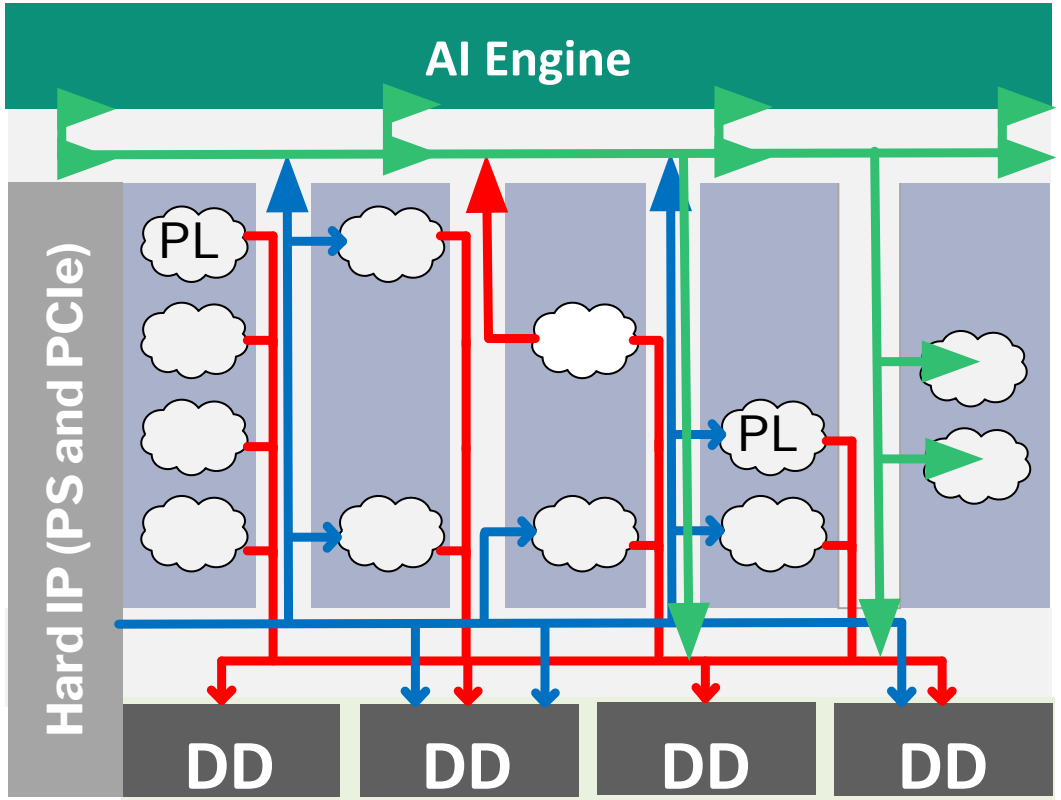
Physical Design Flow in FPGA vs ASIC

- **Synthesis and Logic Optimization**
  - Different target mapping library: LUT, Flop, Carry, BRAM, URAM, DSP.
  - Memory blocks resourcing is inferred, rather than explicitly specified
  - Differences after technology mapping, especially in opt
- **Placement**
  - **Floorplanning**
    - Single die vs. multi die
    - Top level partitioning
  - **IO/Clocking**
    - Programmable clock tree, no slew fixing, skew optimization
  - **Global and Detailed placement**
    - Discrete, collimated site layout grid, clock region constraints
  - **Physical Optimization**
    - Discrete optimization for WNS Improvement
- **Routing**
  - Multiple 3D layers vs. 2D segmented (short/medium/long) wires
  - No DRC, rather PIP state transition rules
- **Timing**
  - Buffered delay model vs. RC networks
  - No gate sizing or buffer insertion in opt
- **Design Flows**
  - Custom flows per application domain: Emulation flow (capacity), prototyping (capacity/performance), HPC flows (performance),



Connecting heterogenous components with NoC

- Data Flow
  - Host (PCIe) reads/writes to DDR or PL
  - PS reads/writes DDR, and controls PL & AIE
  - PL accesses DDR and AIE
  - AIE accesses DDR and PL
- Traffic Source & Destination:
  - Masters: PS, PCIe, PL, AIE Array
  - Slaves: 4x DDR, PL, AIE Array
- NoC provides high-speed interconnect
  - No PL resources consumed for Routing, Switching, and QoS
  - Compiler to optimally route all communication

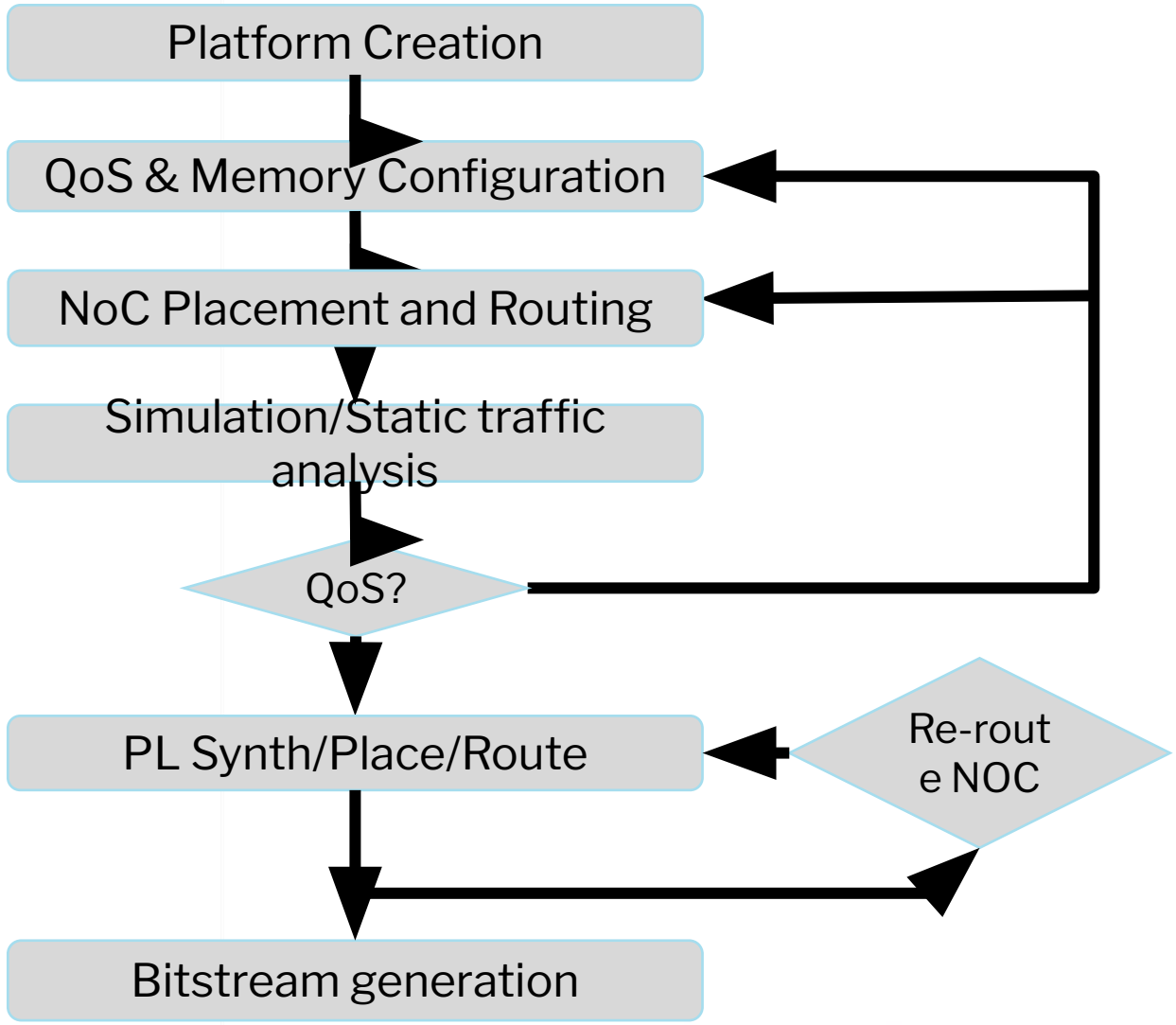


VLSI CAD Problems in NoC

- Application mapping (map tasks to core)
- Floorplanning/placement (within the network)
- Routing (of messages)
- Buffer sizing
- Timing closure (link bandwidth, latency)
- Simulation (traffic/delay/power-modeling)
- Other NoC design problem (virtual channel, arbitration, flow control)

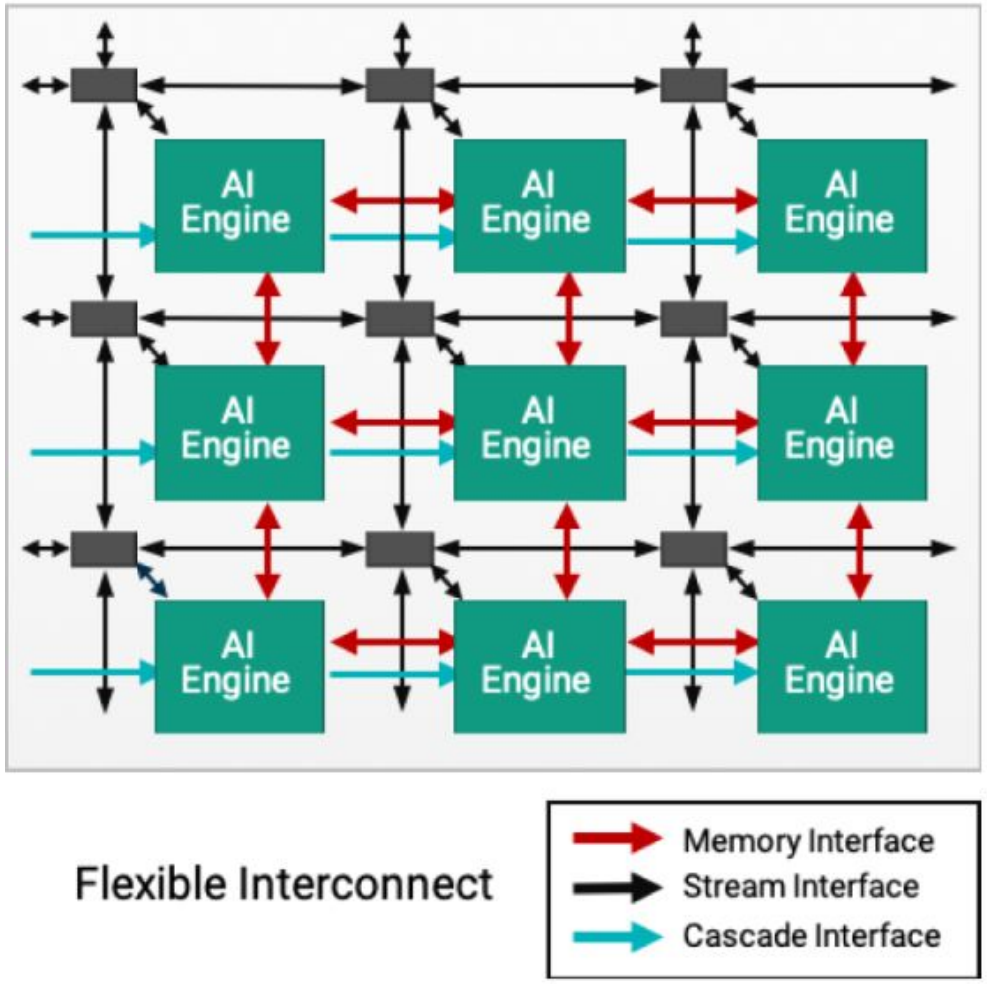
- Credit – Hongkong University of Science and Technology

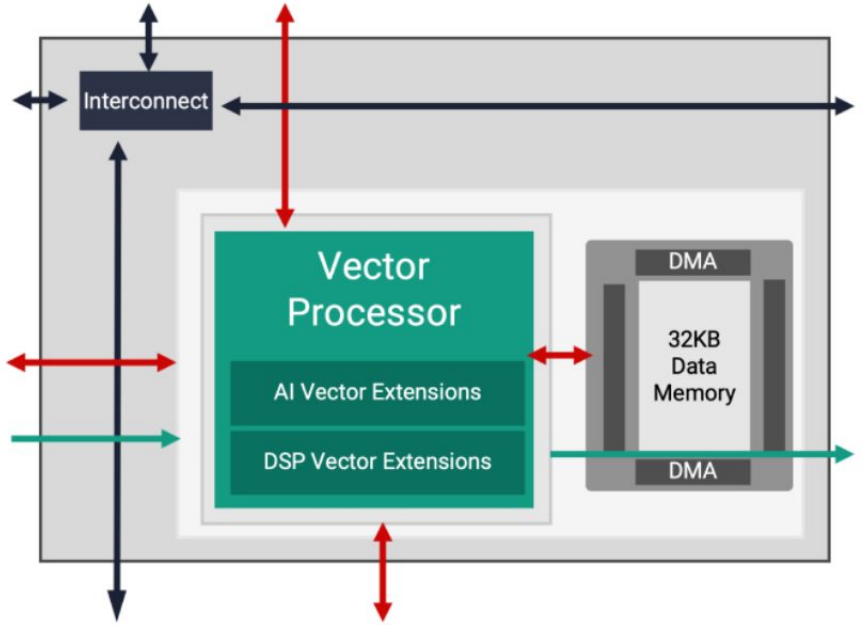
NoC design flow



- Additional constraints
  - Deadlock avoidance
  - Stall avoidance
  - Priority arbitration
  - Virtual Channel allocation
  - Latency optimization
  - Throughput optimization

AI Engine





AIE Mapper and Router – Objective

Problem:

Map:
- **Kernels to AIE cores**
- **Buffers to Memory Banks**

Route:
- **AIE Kernel to Kernel/PL Connections**
- **DMA to DMA/PL Connections**

Satisfy:
- **User constraints**

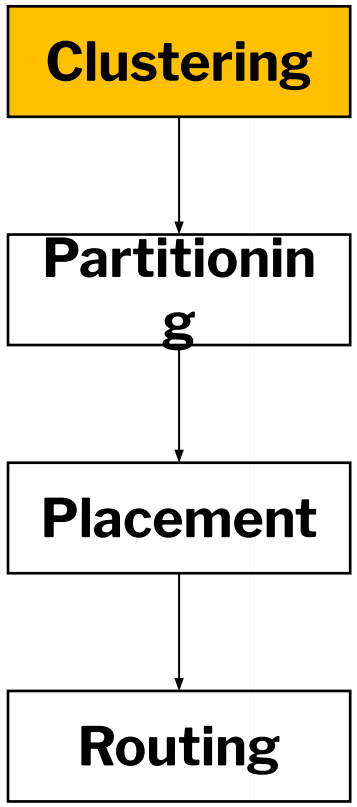
Ideal Solution:

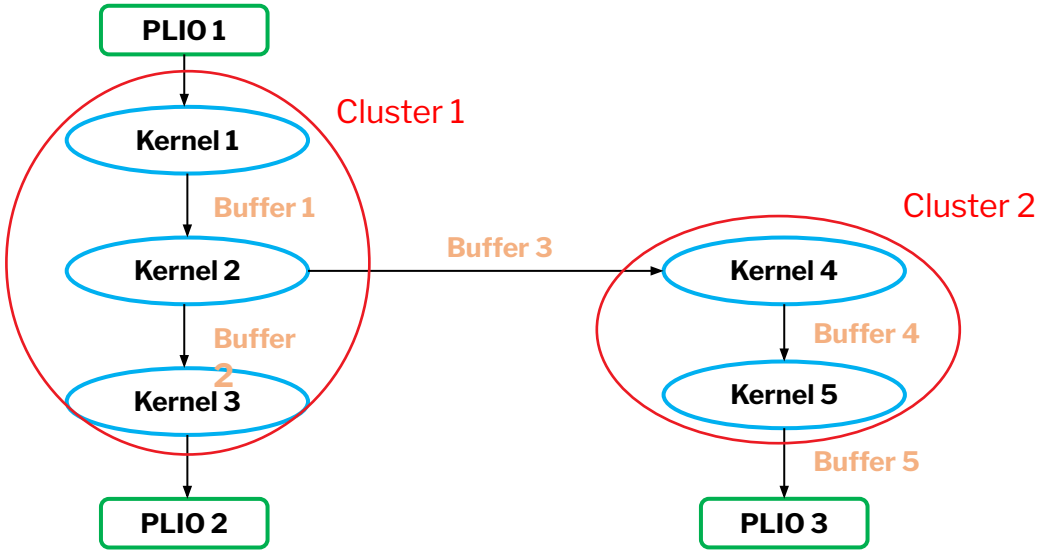
- **Min DMA □ Lower Latency, Higher Throughput**
- **Min memory conflicts □ Higher Throughput**
- **Min WL □ Lower Latency, Better Routability**
- **Compile Time in minutes!**

Challenges:

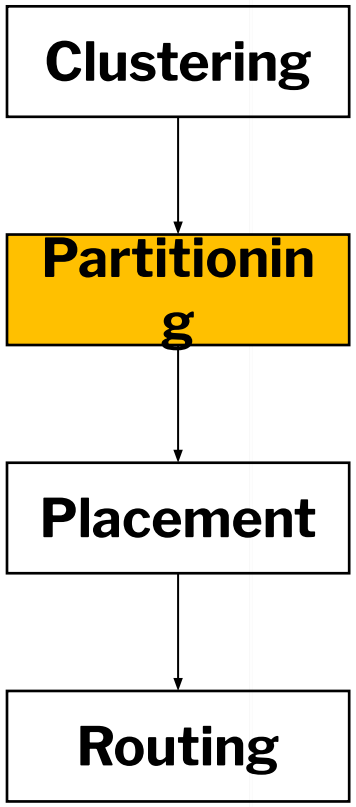
- Very high core utilization
  - Most designs have > 90% utilization
- Large number of memory buffers
  - Impact on throughput due to DMA & conflicts
- Design and Architecture constraints
  - Ping-pong buffers
  - Data-split buffers
  - Broadcasts, Splits & Merges

AIE Mapper and Router– Implementation





AIE Mapper and Router– Implementation



Aim

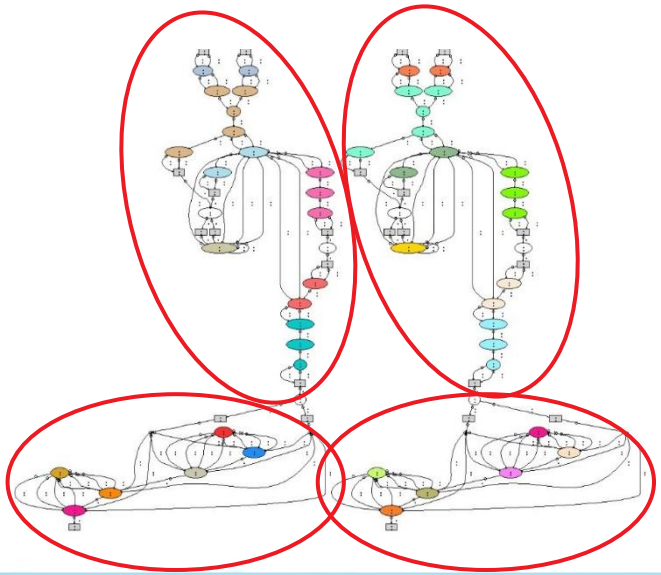
- Reduce SDF into multiple sub-graphs
- Reduce device size

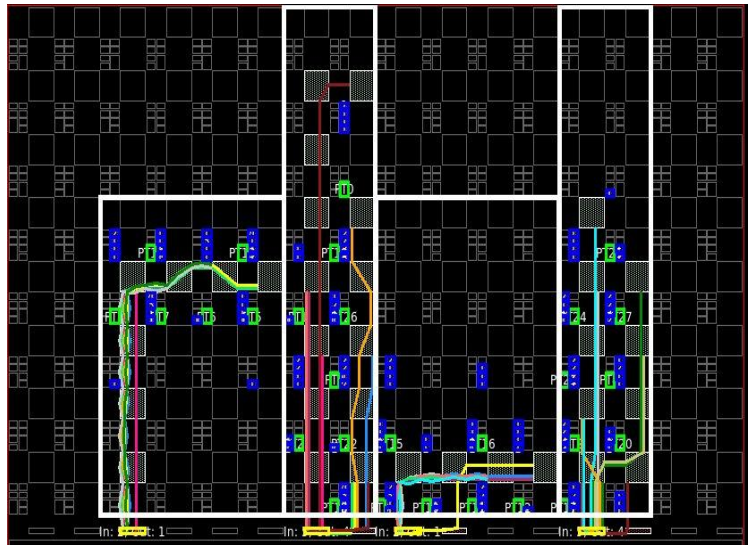
Approach

- Recursive bi-partitioning of SDF graph

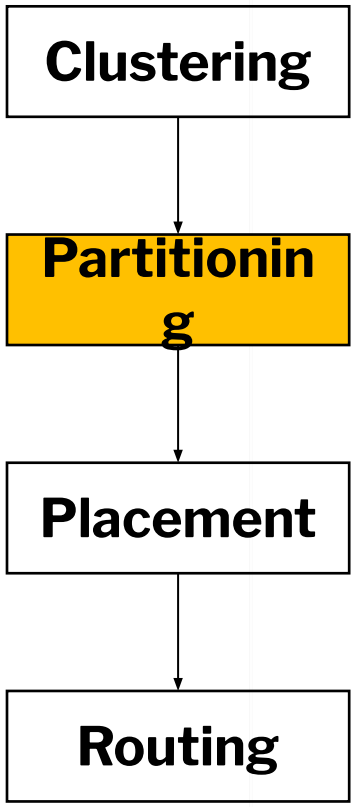
Result

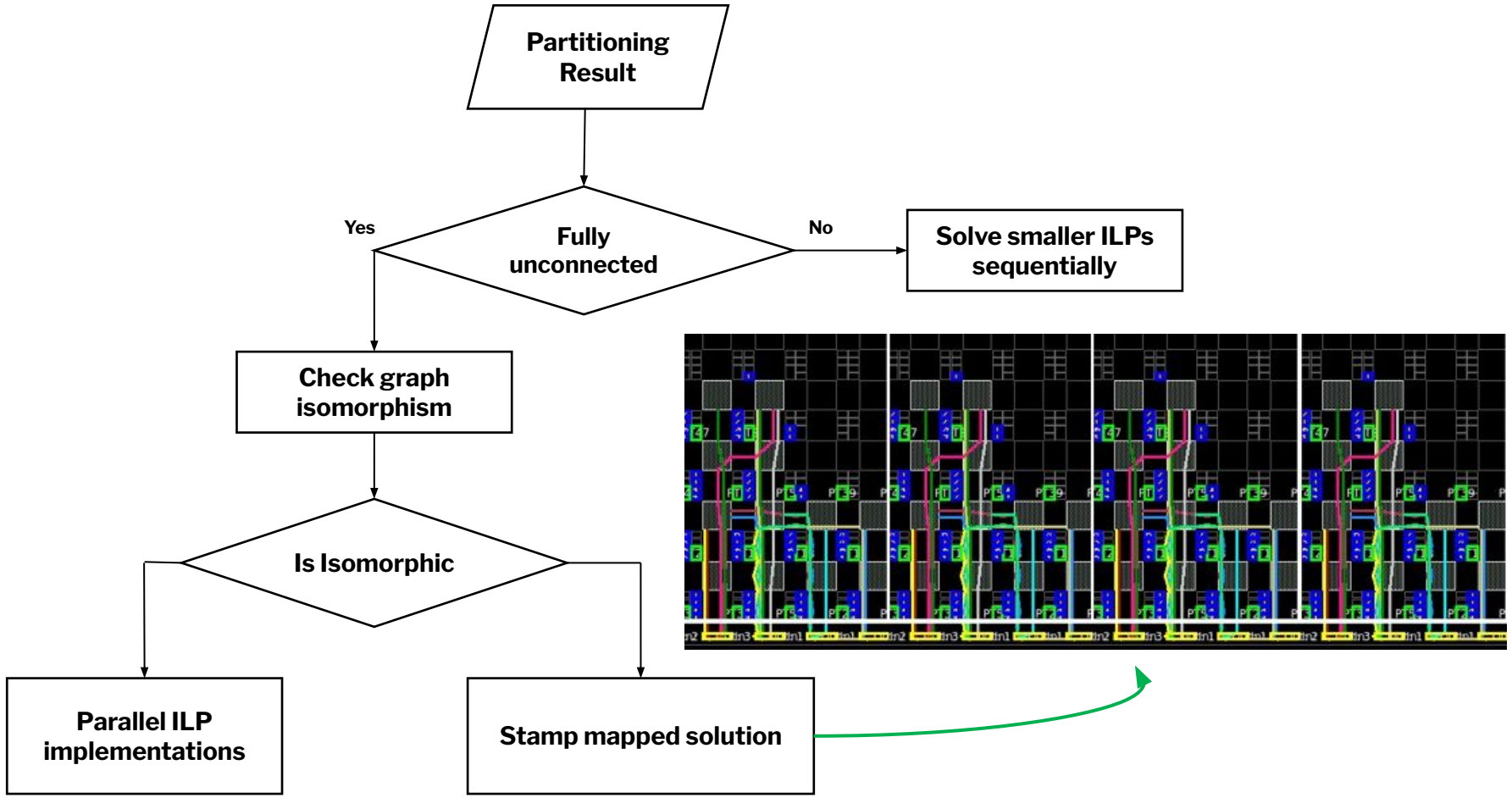
- Sub-graphs with designated device regions



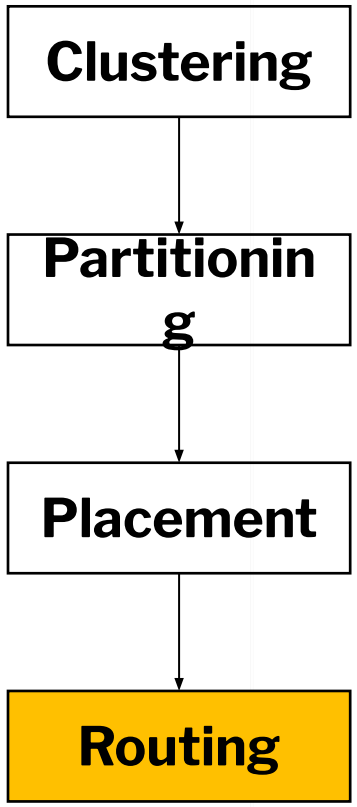


AIE Mapper and Router– Implementation

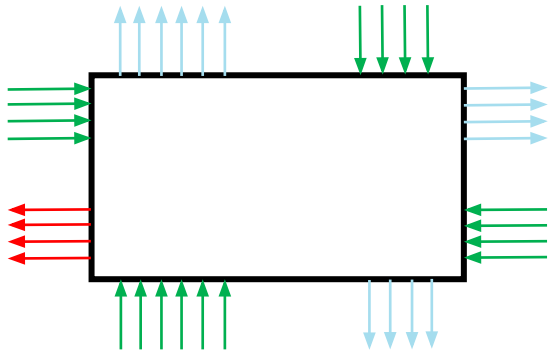


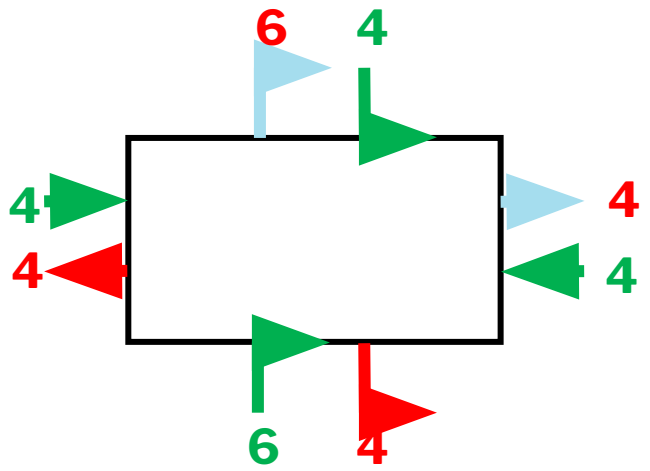


AIE Mapper and Router– Implementation

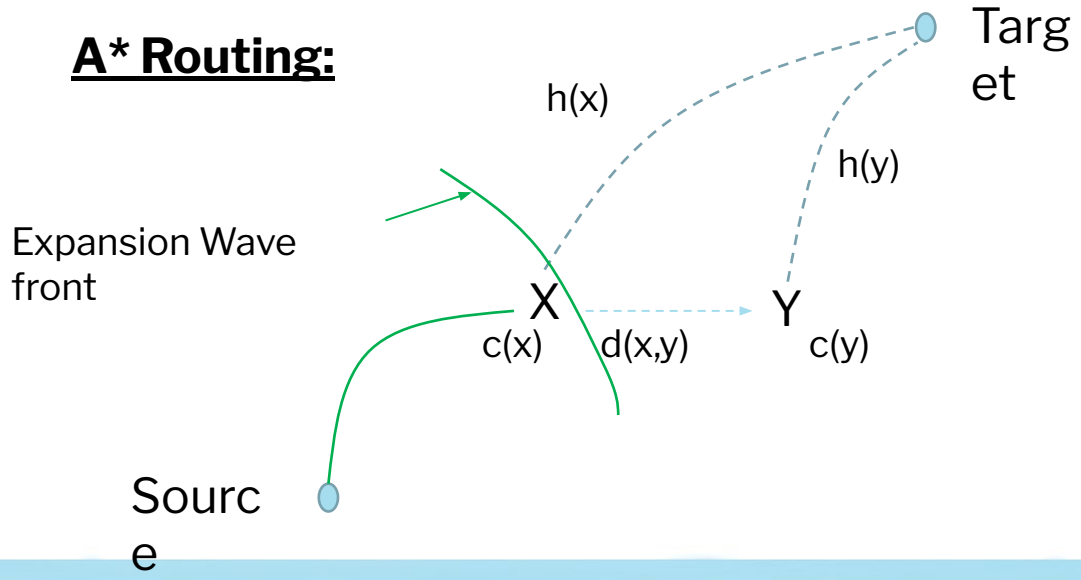


Coarsening Routing Graph:



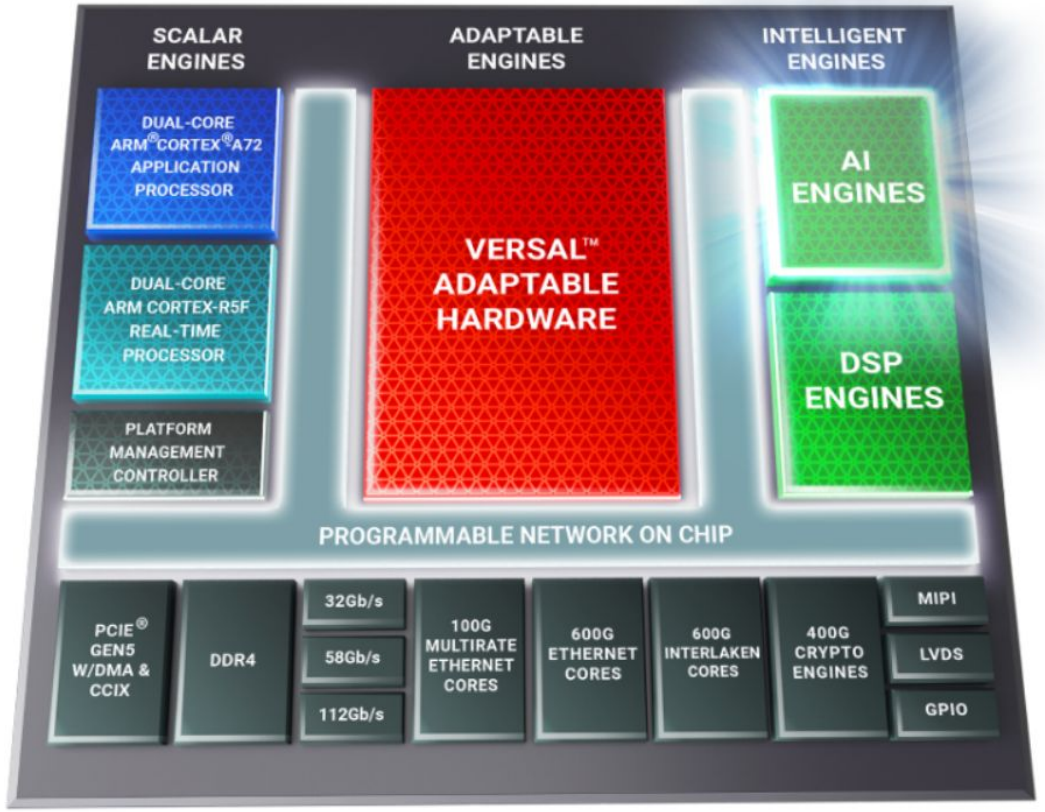


A* Routing:



Bringing it all together

- Engineering fast, reliable and consistent solution for each computing sub-system and data movement
- Managing dependencies across the subsystems
- Providing incremental solution with stability



COPYRIGHT AND DISCLAIMER

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.


AMD