Concurrent High Performance Processor design: From Logic to PD in Parallel

Leon Stok, VP EDA, IBM Systems Group

EETIMES

ASICs won't die, IBM EDA manager says

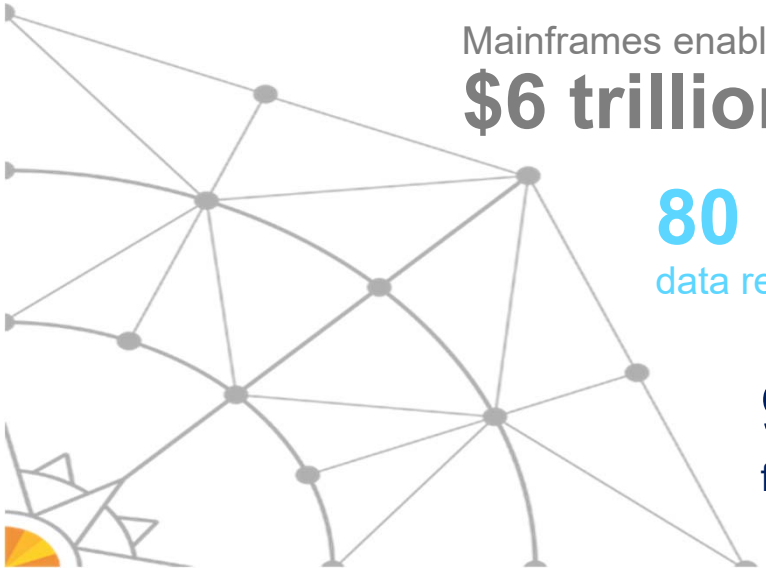
By Richard Goering, EE Times
April 9, 2003 (10:13 a.m. EST)
URL: http://www.eetimes.com/story/OEG20030409S0025

MONTEREY, Calif. — Rumors of the death of ASICs due to soaring costs are "much exaggerated," according to Leon Stok, senior manager of design automation for IBM's Watson Research Center.

Speaking at the International Symposium on Physical Design (ISPD) here Monday (April 7), Stok said mask costs won't be prohibitive and that new EDA tools will help control design costs.

The mainframe is everywhere, making the world work better

Mainframes process
30 billion business transactions per day



Mainframes enable
$6 trillion in card payments annually

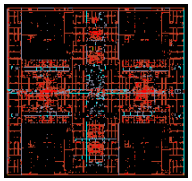
80 percent of the world's corporate data resides or originates on mainframes

91 percent of CIOs said new customer-facing apps are accessing the mainframe

IBM Z – Processor Roadmap

65 nm

z10
2/2008



Workload Consolidation and Integration Engine for CPU Intensive Workloads

Decimal FP

Infiniband

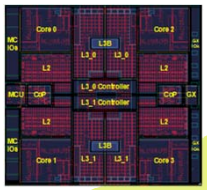
64-CP Image

Large Pages

Shared Memory

45 nm

z196
9/2010



Top Tier Single Thread Performance,System Capacity

Accelerator Integration

Out of Order Execution

Water Cooling

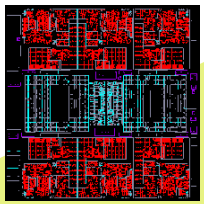
PCIe I/O Fabric

RAIM

Enhanced Energy Management

32 nm

zEC12
9/2012



Leadership Single Thread, Enhanced Throughput

Improved out-of-order

Transactional Memory

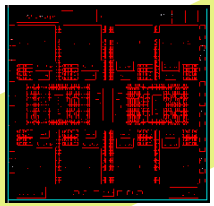
Dynamic Optimization

2 GB page support

Step Function in System Capacity

22 nm

z13
3/2015



Leadership System Capacity and Performance

Modularity & Scalability

Dynamic SMT

Supports two instruction threads

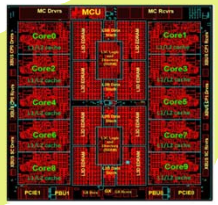
SIMD

PCIe attached accelerators

Business Analytics Optimized

14 nm

z14
9/2017



Pervasive encryption

Low latency I/O for acceleration of transaction processing for DB2 on z/OS

Pause-less garbage collection for enterprise scale JAVA applications

New SIMD instructions

Optimized pipeline and enhanced SMT

Virtual Flash Memory

z14 processor design summary

Micro-Architecture

- 10 cores per CP-chip
- 5.2GHz

- Cache Improvements:
  - 128KB I$ + 128KB D$
  - 2x larger L2 D$ (4MB)
  - 2x larger L3 Cache
  - symbol ECC

- New translation & TLB design
  - Logical-tagged L1 directory
  - Pipelined 2nd level TLB
  - Multiple translation engines

- Better Branch Prediction
  - 33% Larger BTB1 & BTB2
  - New Perceptron & Simple Call/Return Predictor

- Pipeline Optimizations
  - Improved instruction delivery
  - Faster branch wakeup
  - Improved store hazard avoidance
  - 2x double-precision FPU bandwidth
  - Optimized 2nd generation SMT2

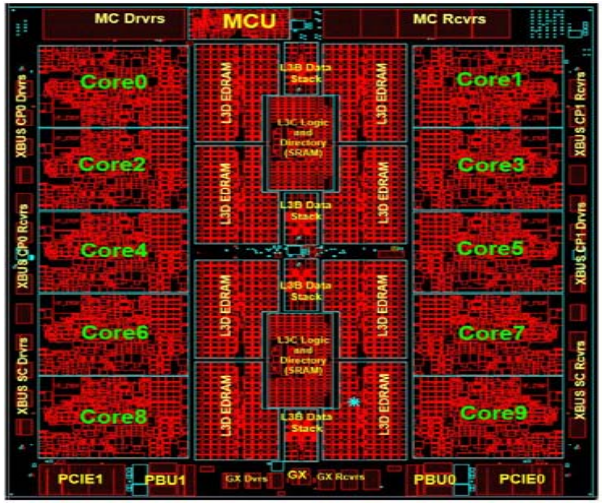
Architecture

- PauseLess Garbage Collection
- Vector Single & Quad precision
- Long-multiply support (RSA, ECC)
- Register-to-register BCD arithmetic

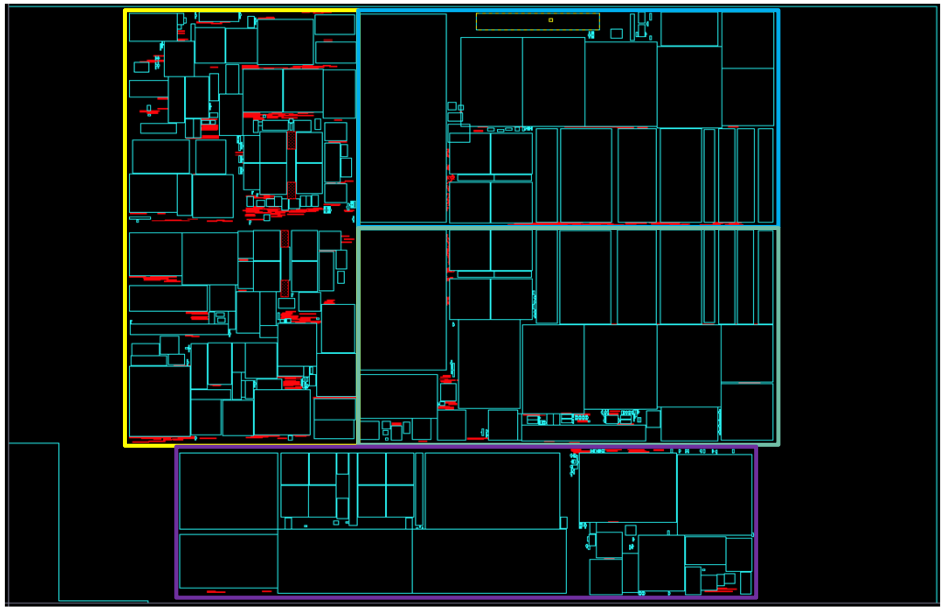
Accelerators

- Redesigned in-core crypto-accelerator
  - Improved performance
  - New functions (GCM, TRNG, SHA3)

- Optimized in-core compression accelerator
  - Improved start/stop latency
  - Huffman encoding for better compression ratio
  - Order-preserving compression



Core shrinkage in 14nm



- 33% area reduction
- Timing within ~-5ps range (FOM's ~-2500)
- ~40% less logic gate width, ~20% less total gate width
- At least as good LVT width, some versions show improvement to significant improvement

Why was this so difficult ? – Logic designers from Venus, PD designers from Mars

Logical Organization Preference

Verification Focus

Logic Ownership

Functional Adjacency

Physical Organization Preference

Implementation Focus

Physical Optimization

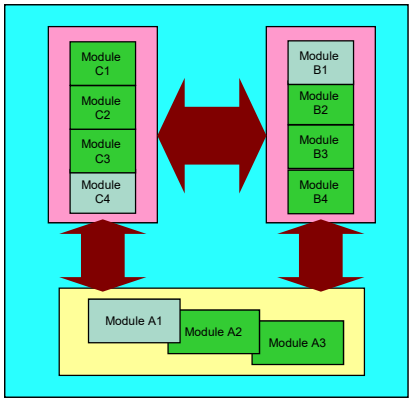
Geographic Adjacency

~~**Combined Single Hierarchy**~~

~~Iterative PD Annotation~~

~~High Coordination Effort~~

~~Less Efficient~~

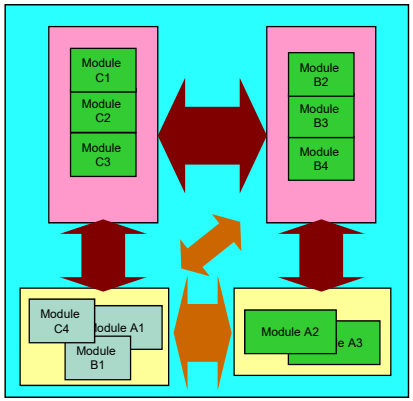


Design Quality

Performance

Power

Area

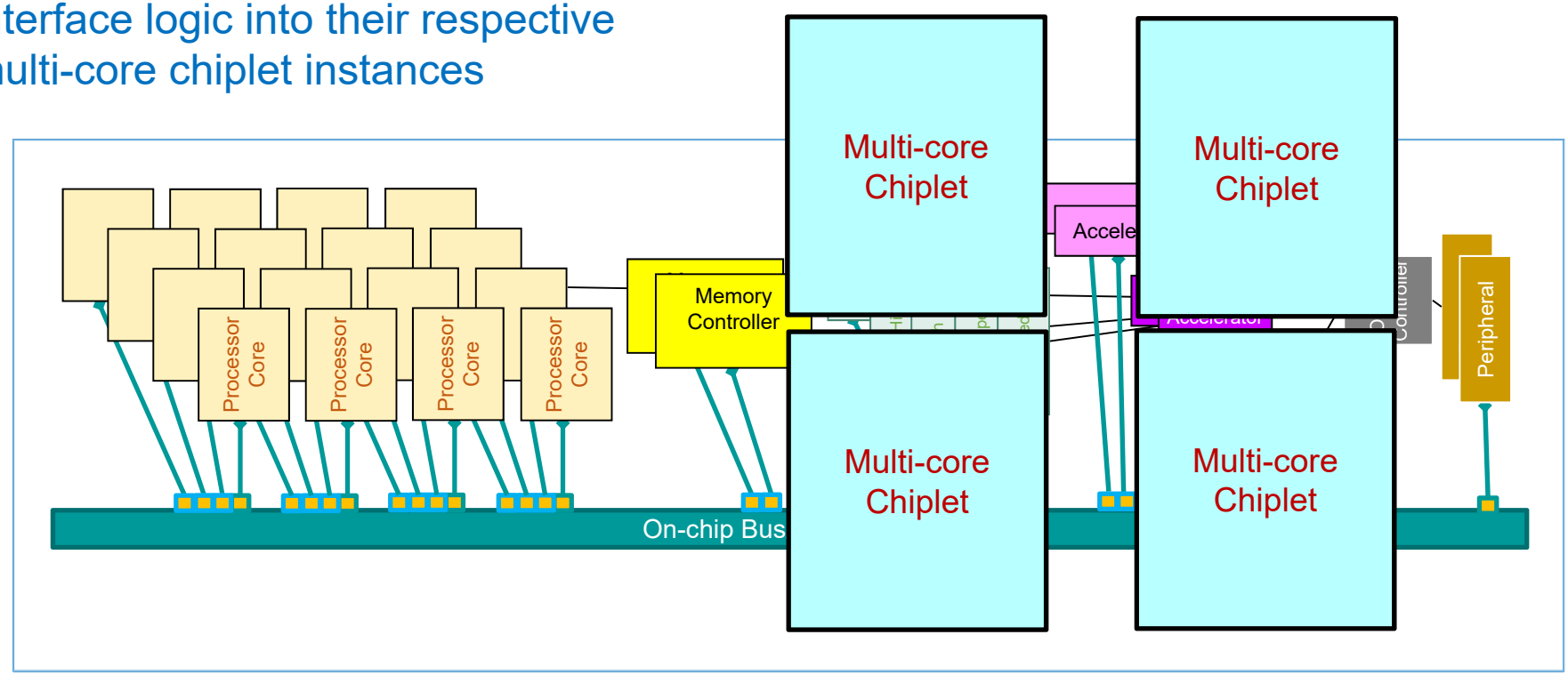


Logic Designers View

An obvious benefit is to create a multi-core chiplet

Move processor cores and bus interface logic into their respective multi-core chiplet instances

Create a multi-core chiplet entity and instantiate it multiple times

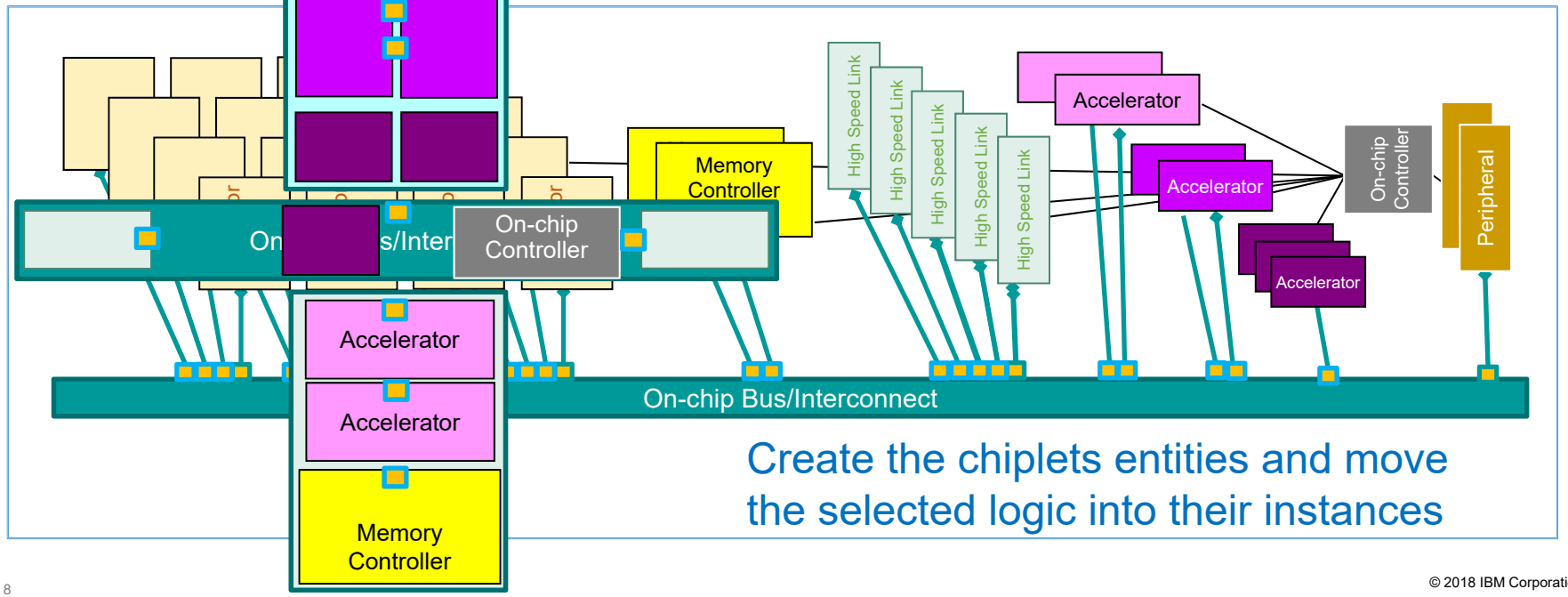


[Alvan Ng, Automated Physical Hierarchy Generation: Tools and Methodology, DVCon2018]

Create Integration Chiplets For Manageability

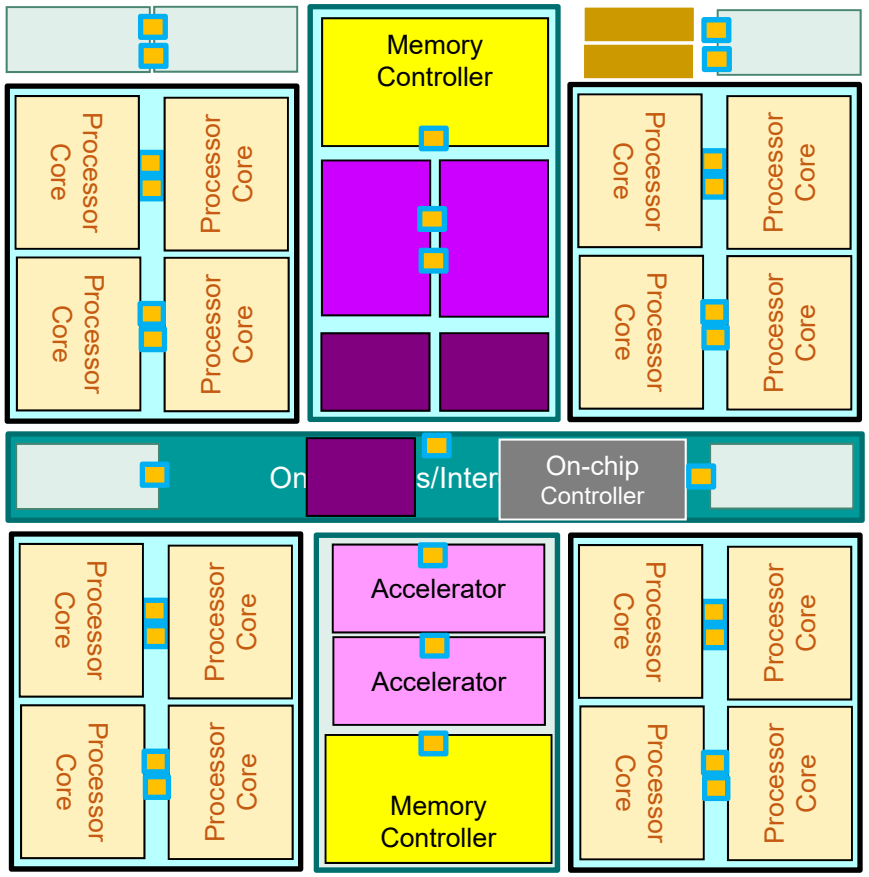
The physical blocks are reshaped to fit into the physical chiplets

A North and South chiplets and a Bus chiplet are good choices



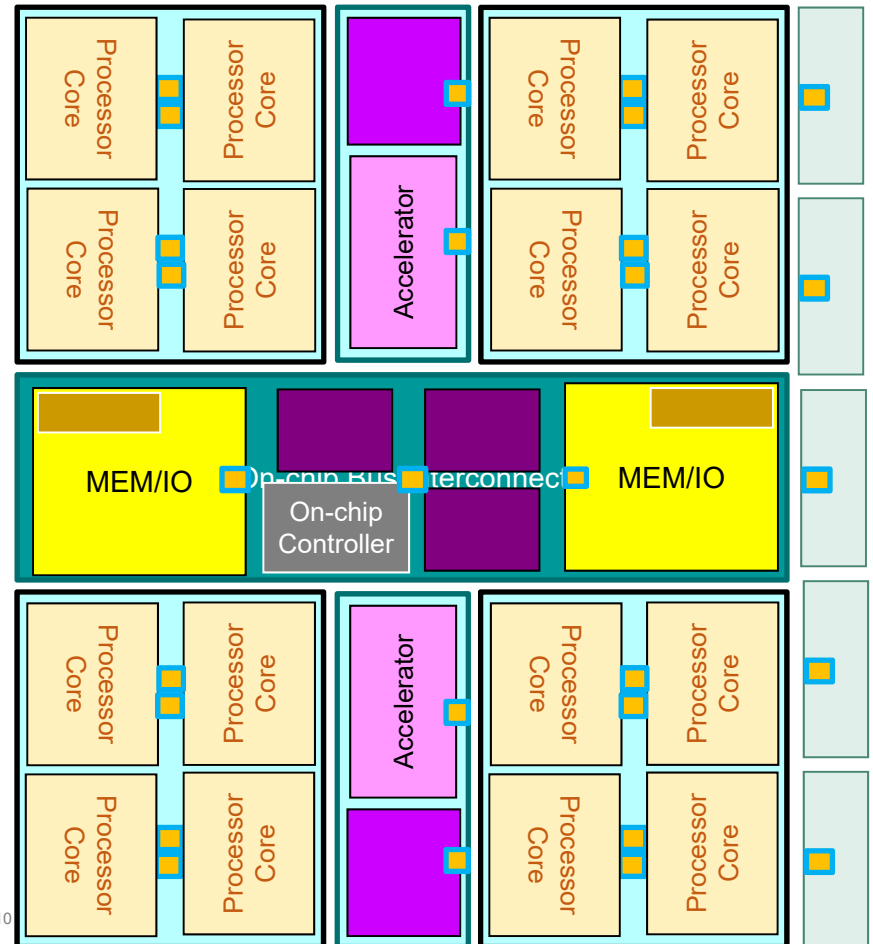
Create the chiplets entities and move the selected logic into their instances

Multi-core Processor Chip Physical Floorplan



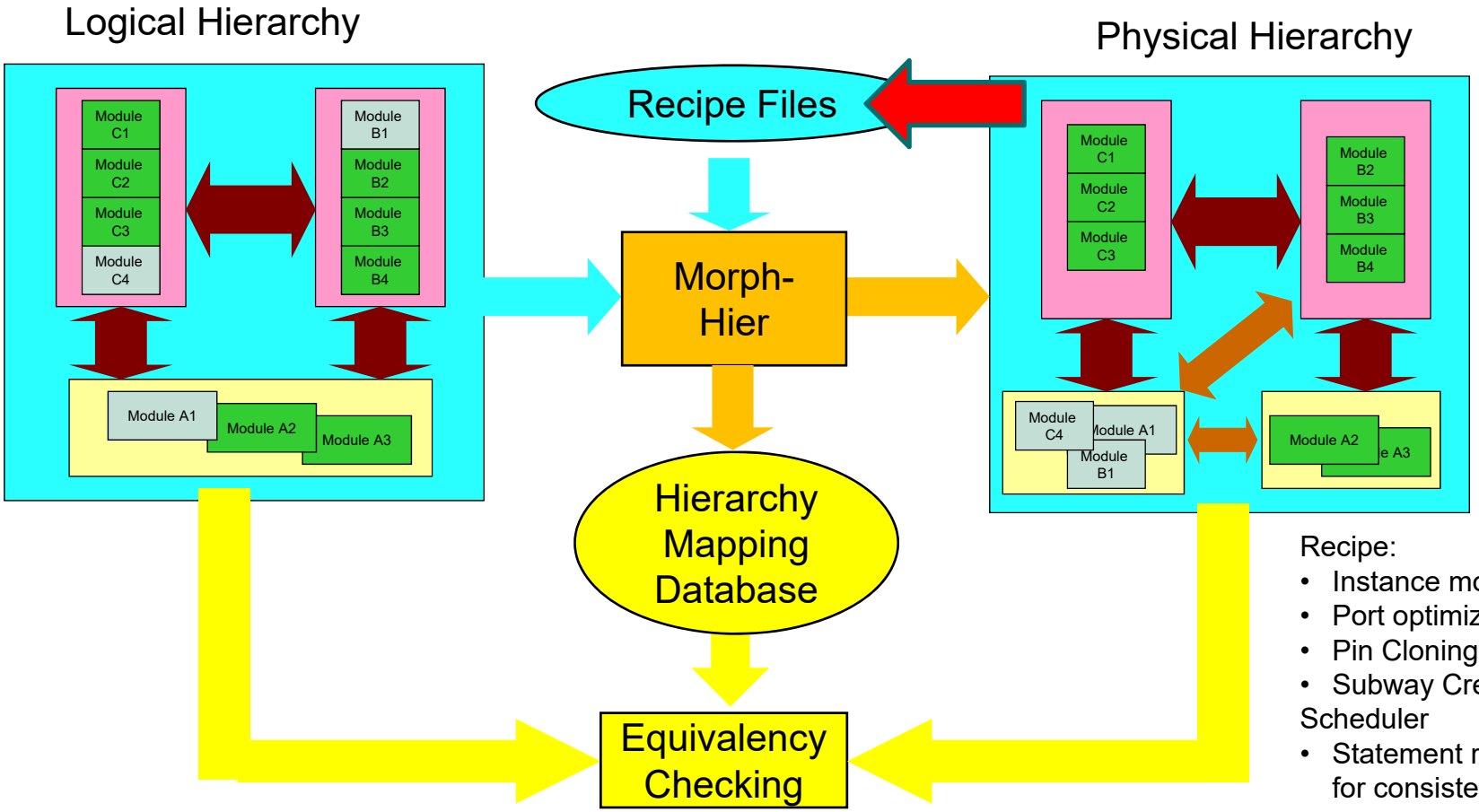
- Quad-core chiplet instantiated 4 times
- Center stripe bus chiplet with 2x high speed link, 1 small accelerator, and the on-chip controller
- Top chiplet contains 1 memory controller, 2 small accelerators, and 2 medium accelerators
- Bottom chiplet contains 1 memory controller and 2 large accelerators
- Stack the rest of circuitries in the open spaces at the top

Alternative Chip Physical Floorplan



- **Quad-core chiplet instantiated 4 times**
- **Center stripe bus chiplet contains 2x Memory-Peripheral combined unit, 3x small accelerator, and the on-chip controller**
- **One accelerator chiplet instantiated twice which contains a large and a medium accelerator**
- Stack the High-Speed Links on the right

Morph: RTL to RTL morphing



Recipe:

- Instance move
- Port optimization
- Pin Cloning
- Subway Creation

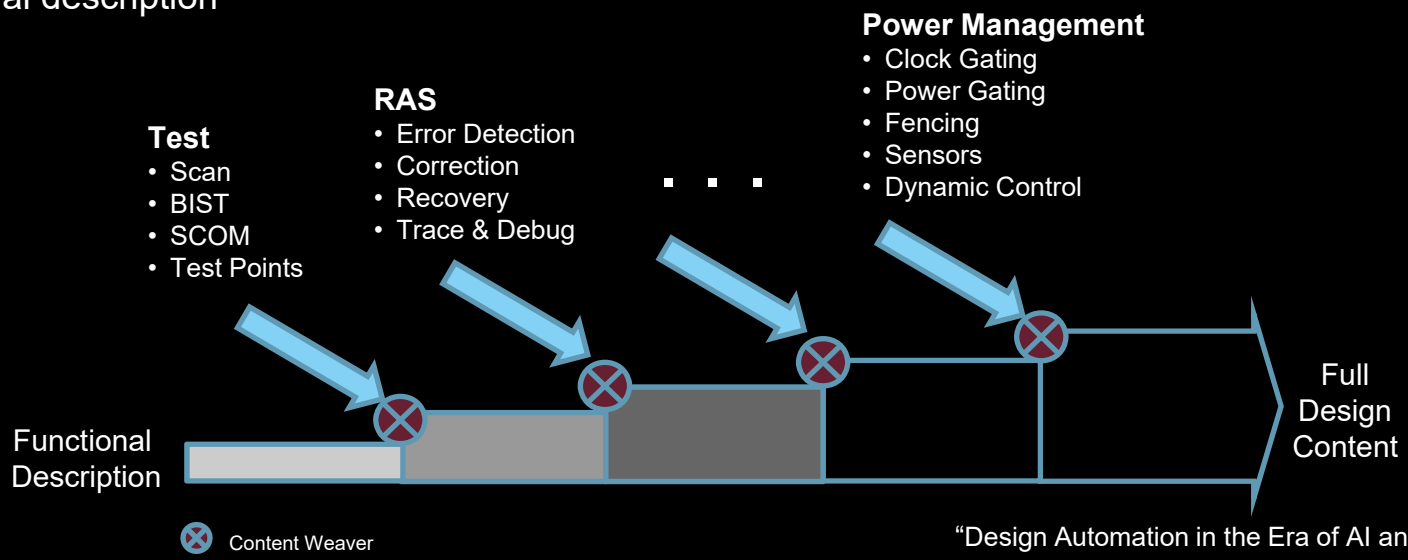
Scheduler

- Statement reordering for consistency

IoT Design Automation Tools – Aspect Oriented Design

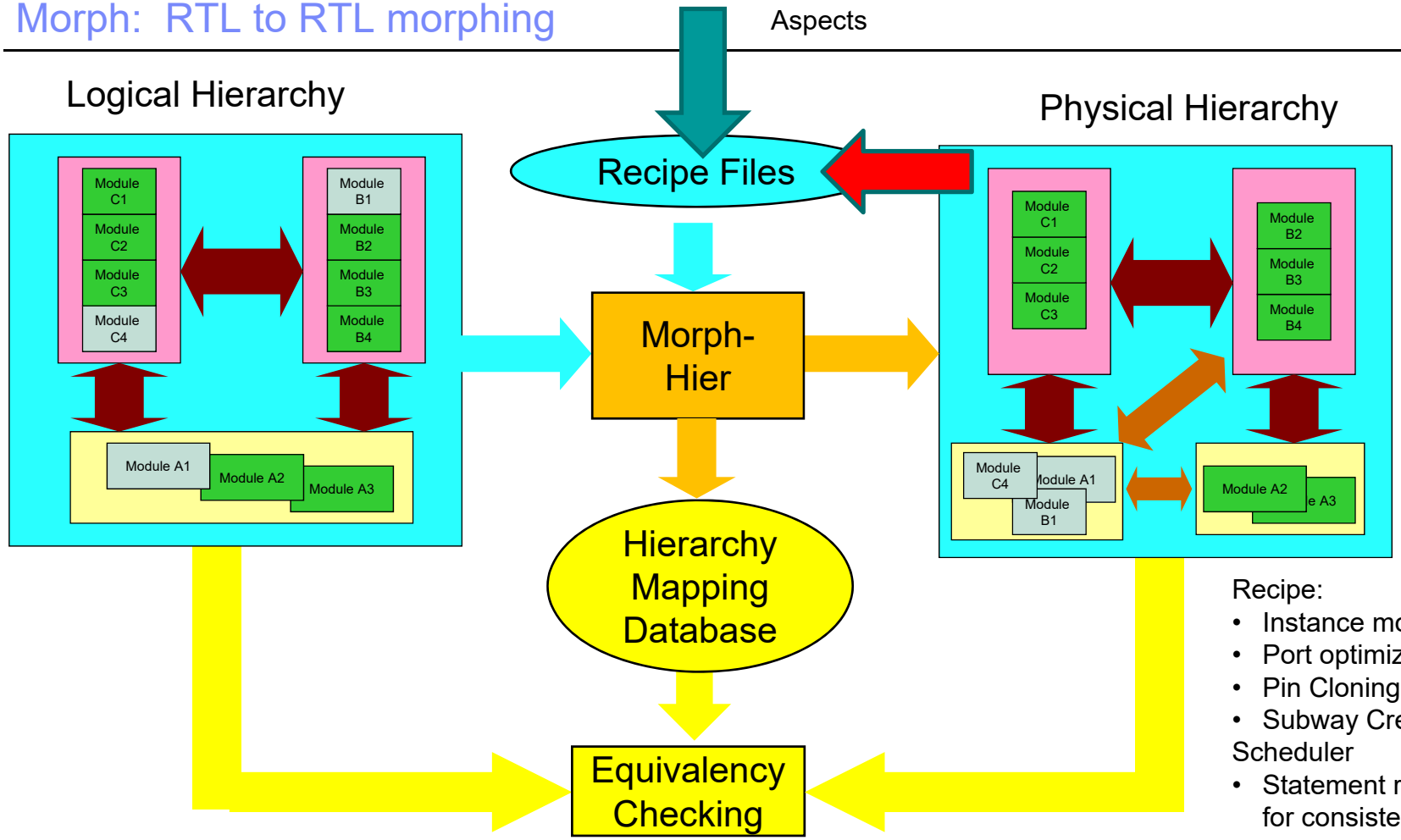
Significant design content exists to support *non-mainline* functionality. This impacts the ability to readily reuse design IP and hinders productivity by forcing designers to include such concerns while implementing core functionality

Need a design system that fully separates the insertion of non-mainline aspects from the core functional description



"Design Automation in the Era of AI and IoT", Arvind Krishna, IEEE/ACM DATE Conference, March 28, 2017

Morph: RTL to RTL morphing



Recipe:

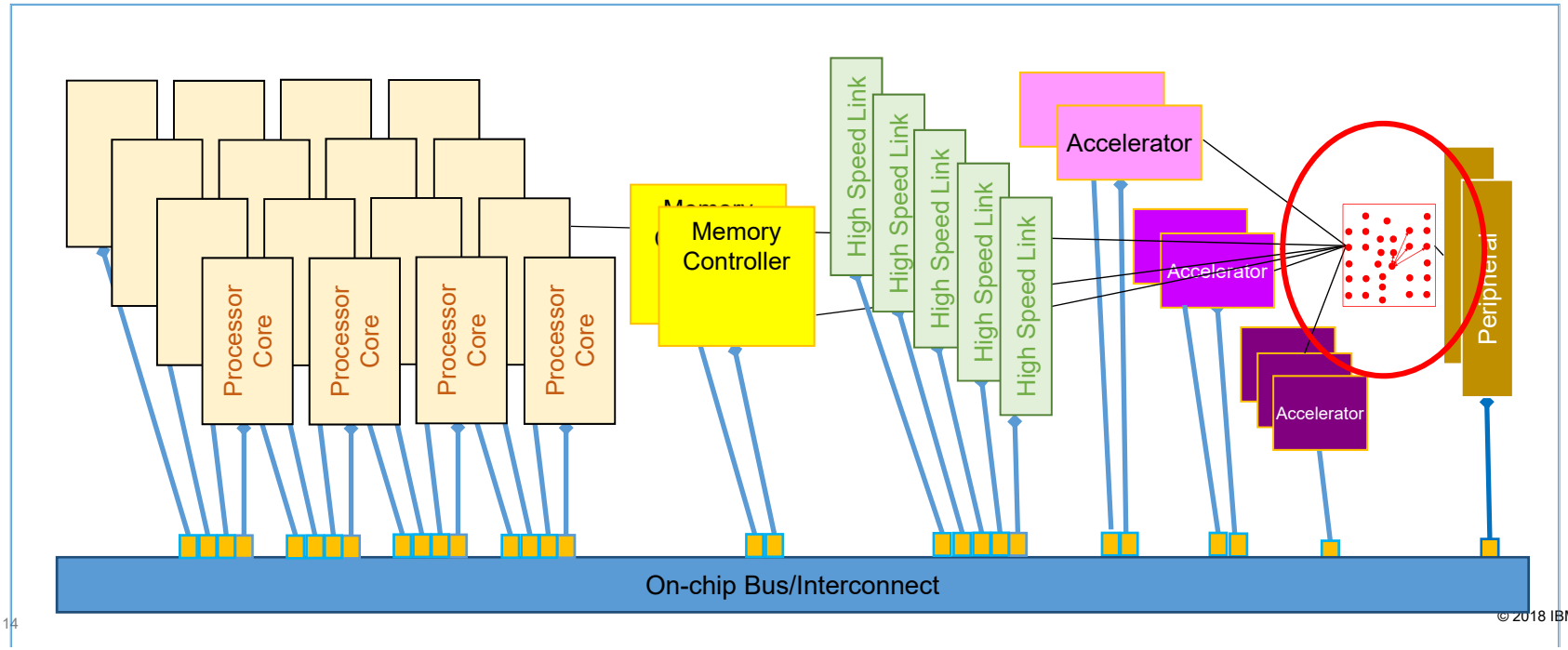
- Instance move
- Port optimization
- Pin Cloning
- Subway Creation

Scheduler

- Statement reordering for consistency

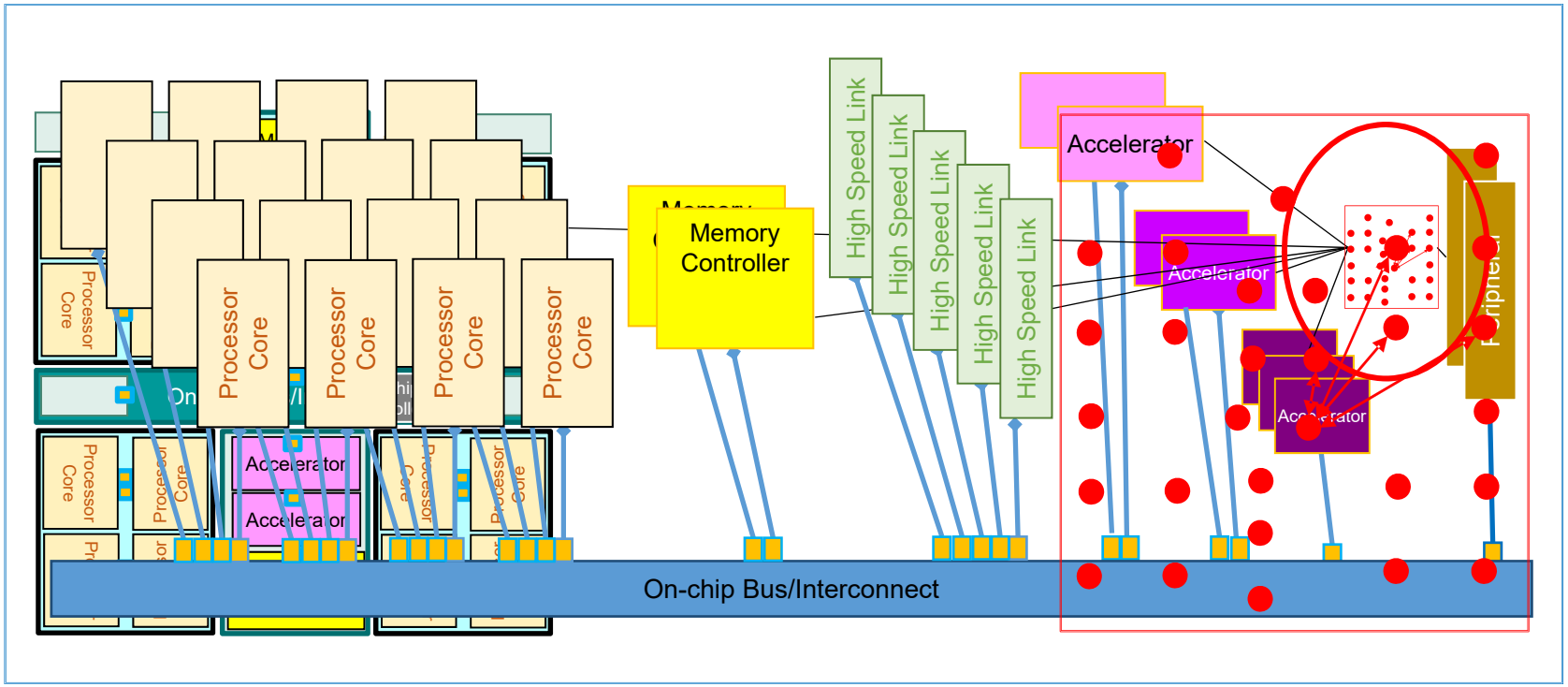
Pervasive Logic Centralized VHDL Organization

- On-chip Controller Logic
- Test Logic
- Miscellaneous Circuitries

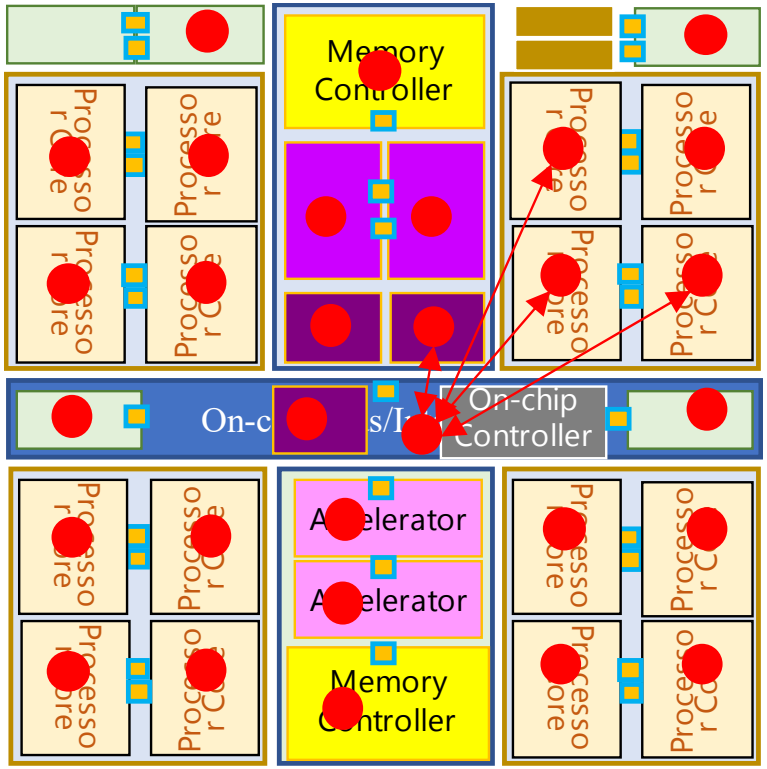


Distribute Pervasive Logic Using Morph-Hier

The Pervasive logic can be distributed physical partitions using Morph-Hier



Centralized Pervasive Logic Distributed To Physical Units



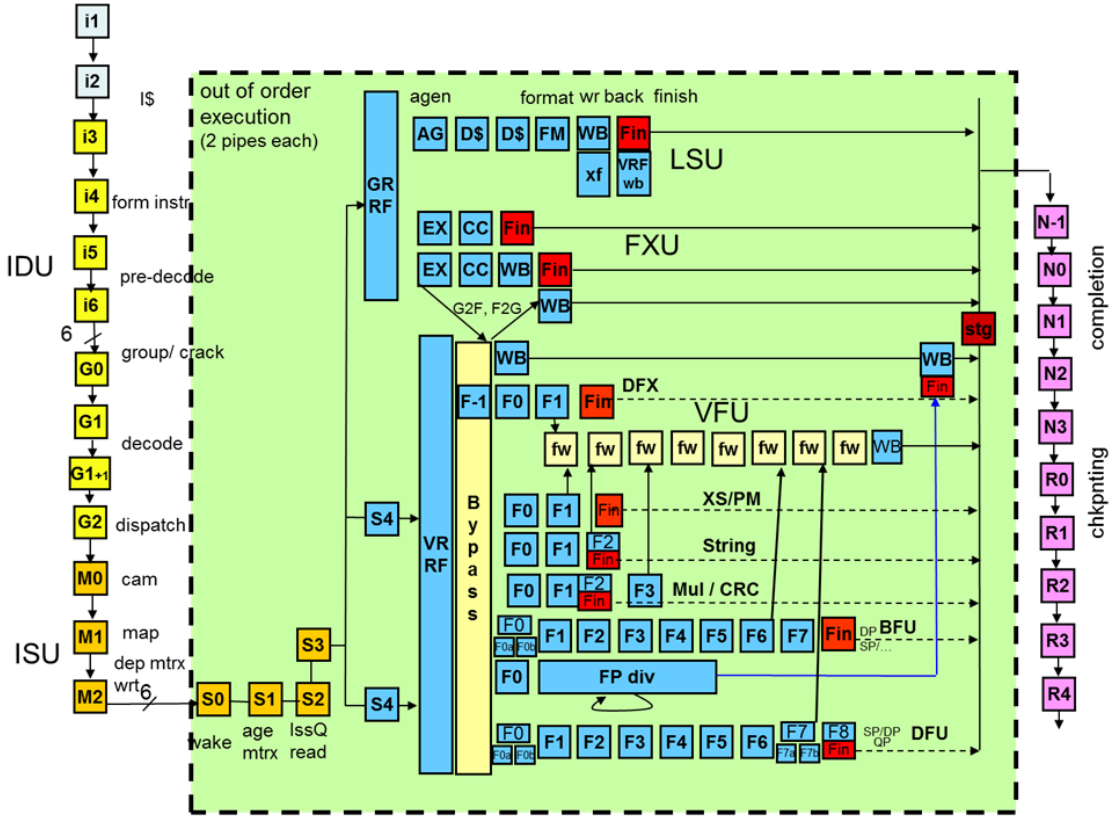
Benefits:

- Parallel logic design
  - Concurrent with functional units
- Verification Speedup
  - Self contained unit
- Design quality
  - Lower bug rate

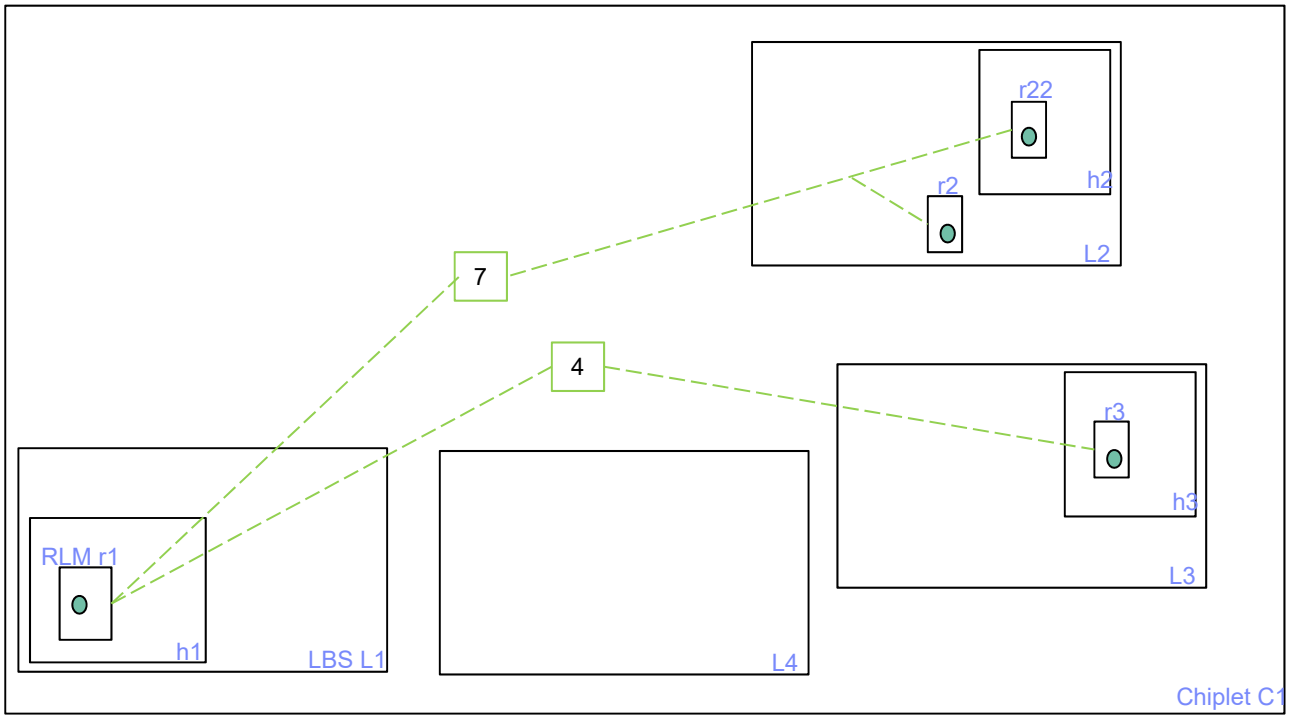
z14 Pipeline

Deep high frequency pipeline

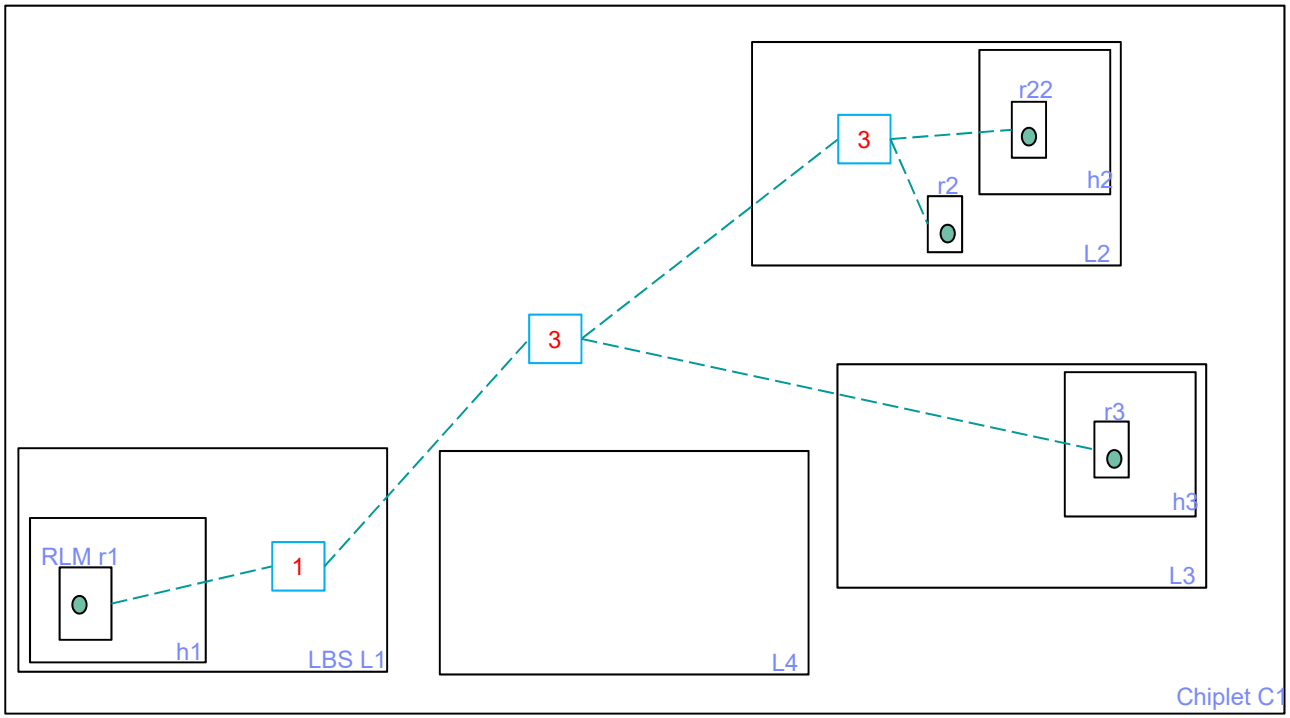
- Async branch prediction ahead of ifetch
- 32B/cycle ifetch
- 6 instruction / cycle parse & decode
- CISC instruction cracking
- Unified OOO issue queue
- 2 LSU, 4-cycle load-use
- 4 FXU, 2 SIMD/FP/BCD
- In-order completion & checkpoint



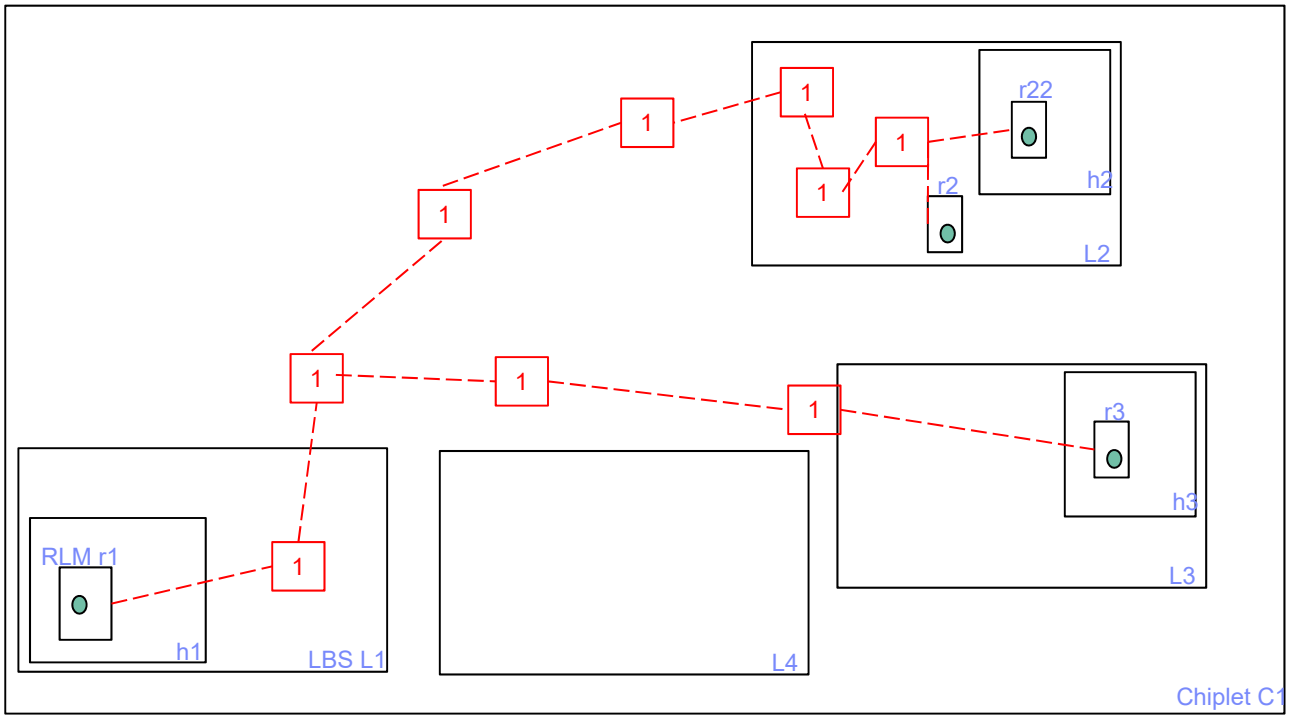
Physical constraints on the pipeline



PD micro-architect allotment



Sequential Buffering



Conclusions

- Most **innovation** in micro-processors is nowadays coming from
  - Architecture, micro-architecture and accelerators
  - Physical design optimization at micro-architectural level
  - **In place of**
    - Moore's law technology progress and
    - 'Fixed block' level PPA optimization.
- This is leading significantly more 'new' Logic being designed and modified, **concurrently** with the Physical Design
- Concurrent design of Logic and PD leads to
  - interesting new problems to be explored with significantly larger potential pay-off due the micro-architectural / PD co-optimization design space.