CPU Project in Western Digital: From Embedded Cores for Flash Controllers to Vision of Datacenter Processors with Open Interfaces

Zvonimir Z. Bandic, Sr. Director

Robert Golla, Sr. Fellow

Dejan Vucinic, Director

Western Digital Corporation

December 4th, 2018

Forward-Looking Statements

Safe Harbor | Disclaimers

This presentation contains certain forward-looking statements that involve risks and uncertainties, including, but not limited to, statements regarding: the RISC-V Foundation and its initiatives; our contributions to and investments in the RISC-V ecosystem; the transition of our devices, platforms and systems to RISC-V architectures; shipments of RISC-V processor cores; our business strategy, growth opportunities and technology development efforts; market trends and data growth and its drivers. Forward-looking statements should not be read as a guarantee of future performance or results, and will not necessarily be accurate indications of the times at, or by, which such performance or results will be achieved, if at all. Forward-looking statements are subject to risks and uncertainties that could cause actual performance or results to differ materially from those expressed in or suggested by the forward-looking statements.

Additional key risks and uncertainties include the impact of continued uncertainty and volatility in global economic conditions; actions by competitors; business conditions; growth in our markets; and pricing trends and fluctuations in average selling prices. More information about the other risks and uncertainties that could affect our business are listed in our filings with the Securities and Exchange Commission (the "SEC") and available on the SEC's website at www.sec.gov, including our most recently filed periodic report, to which your attention is directed. We do not undertake any obligation to publicly update or revise any forward-looking statement, whether as a result of new information, future developments or otherwise, except as otherwise required by law.

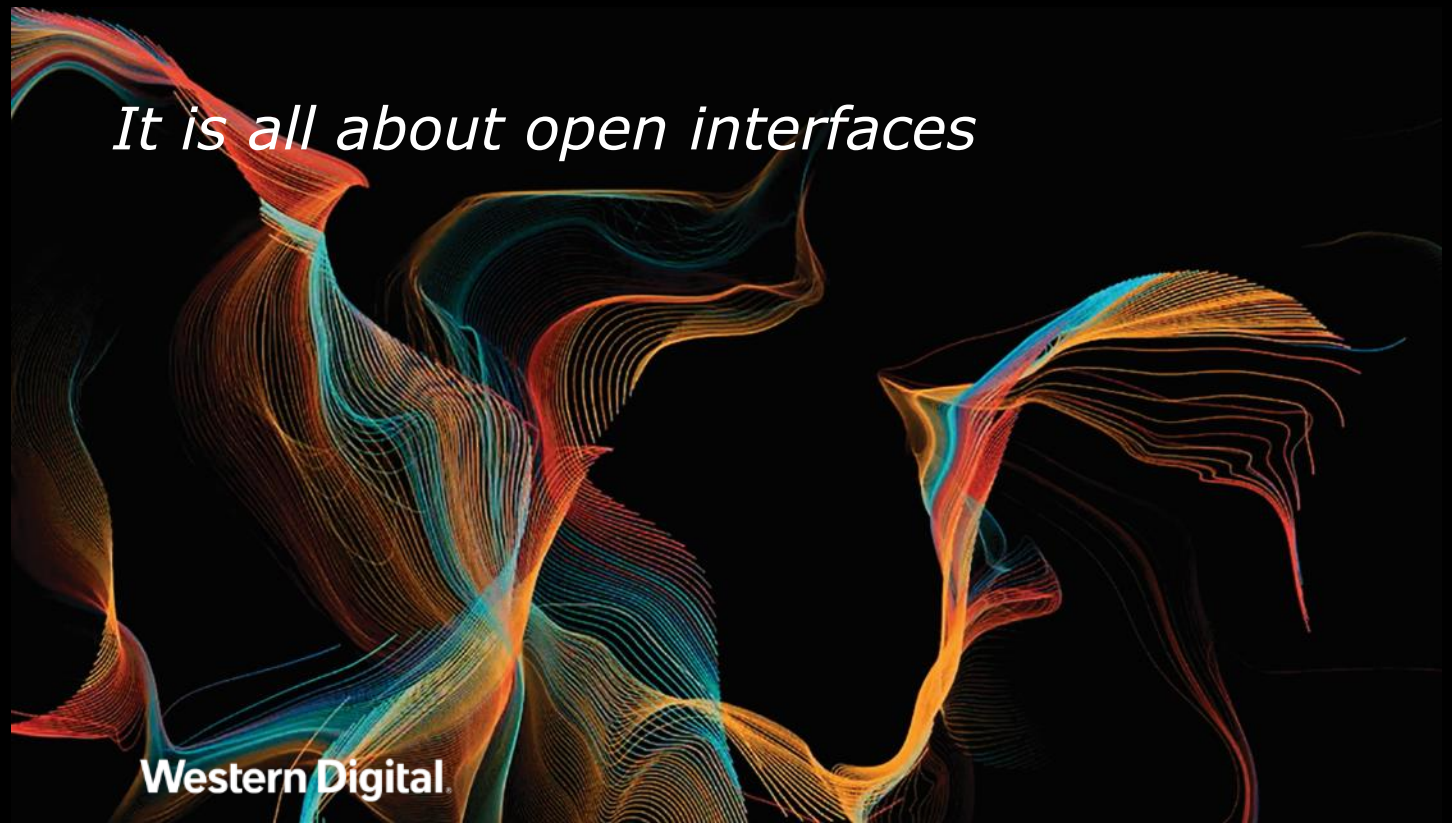
Agenda

- Enterprise datacenter RISC-V CPU vision – all about open interfaces:
  - RISC-V multi-core
  - NVDIMM-P memory interfaces
  - Accelerator interfaces (PCIe, OpenCAPI)
  - OmniXtend™ – memory protocol interface enabling memory centric architectures
- Planned open source contributions
- Western Digital first core SweRV™:
  - Microarchitecture introduction
  - Performance benchmarks
- OmniXtend™ protocol:

#LetDataThrive

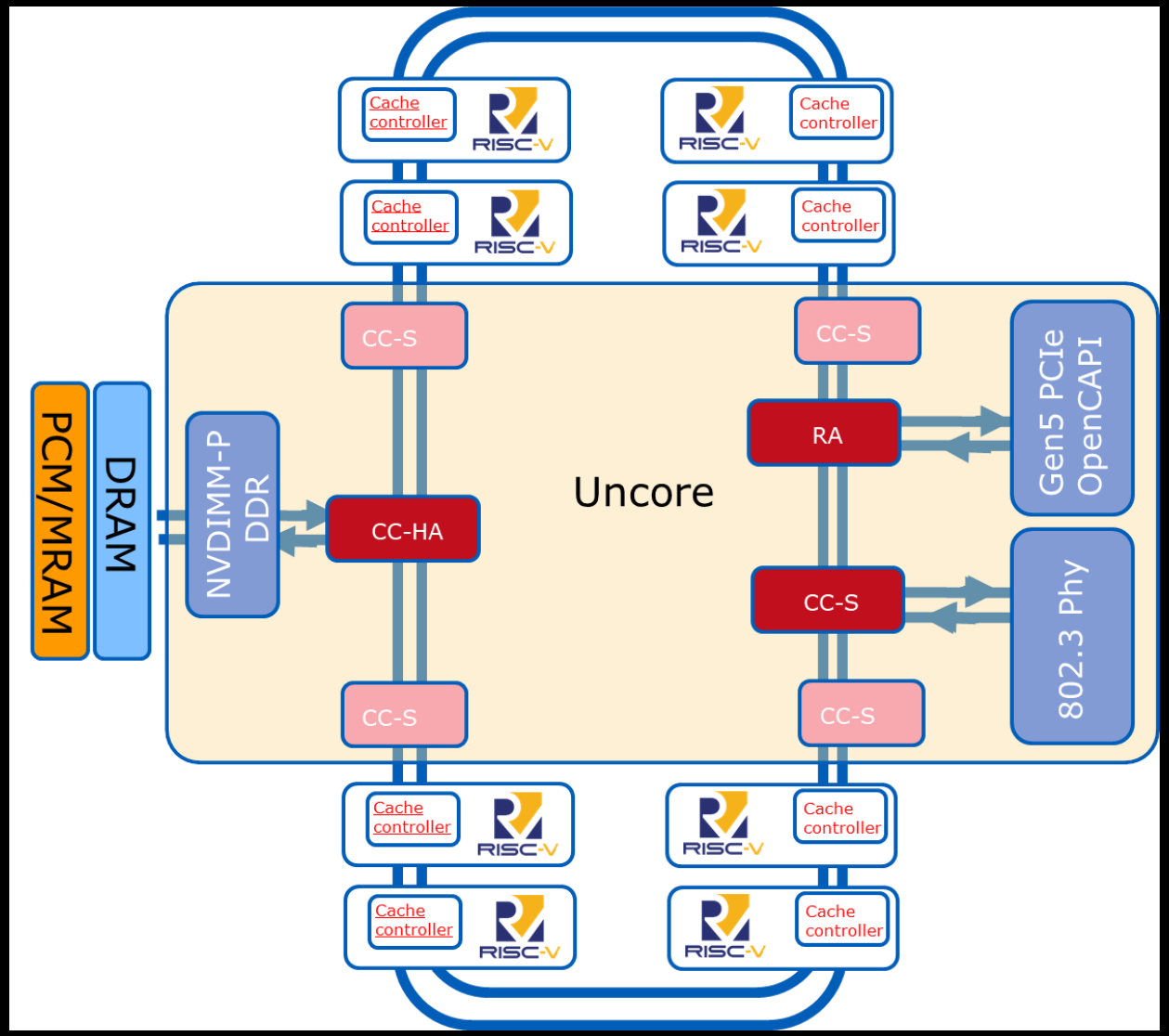
Vision of RISC-V open architecture datacenter CPU

It is all about open interfaces



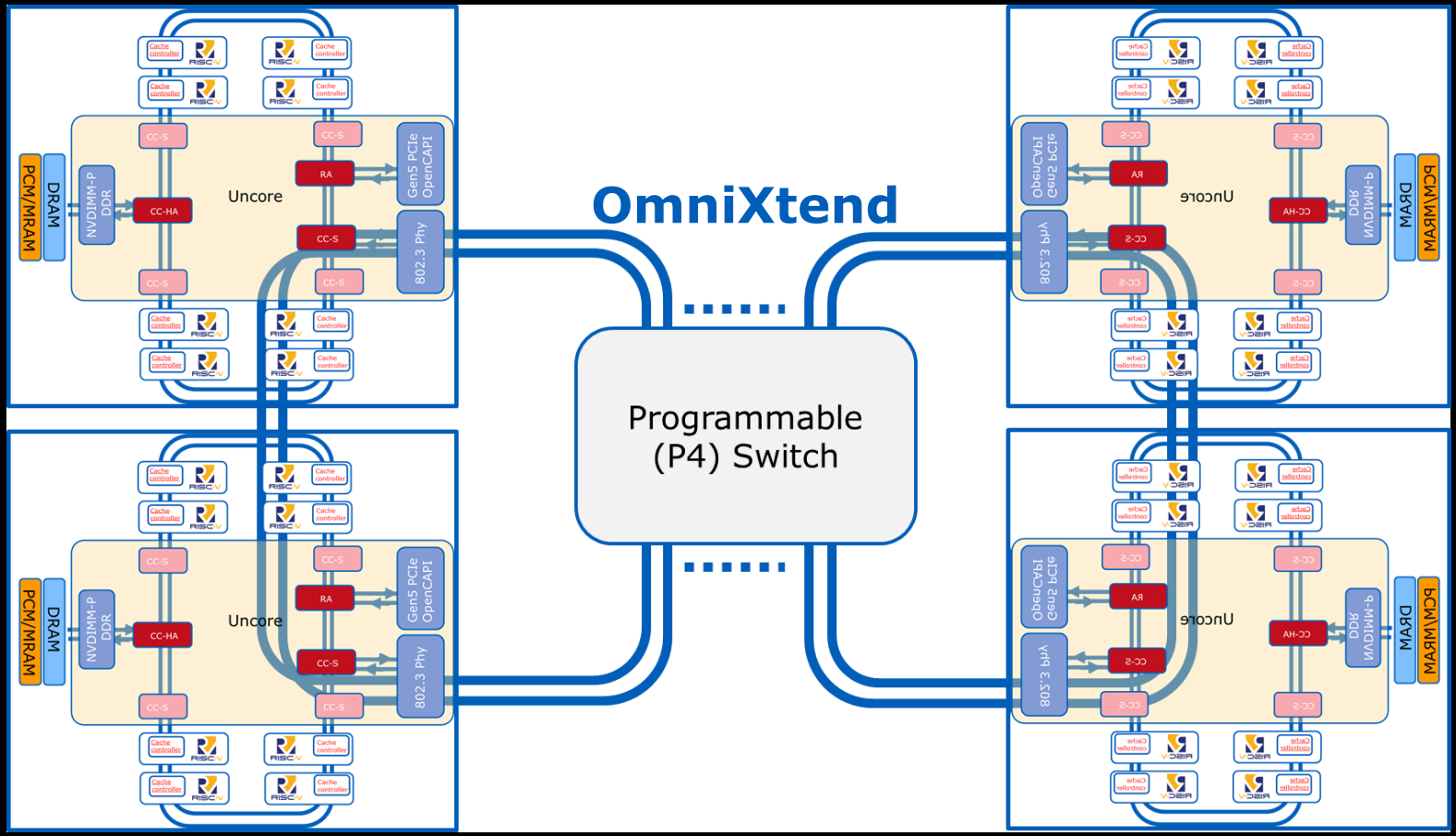
Western Digital.

Vision for future datacenter CPU architecture



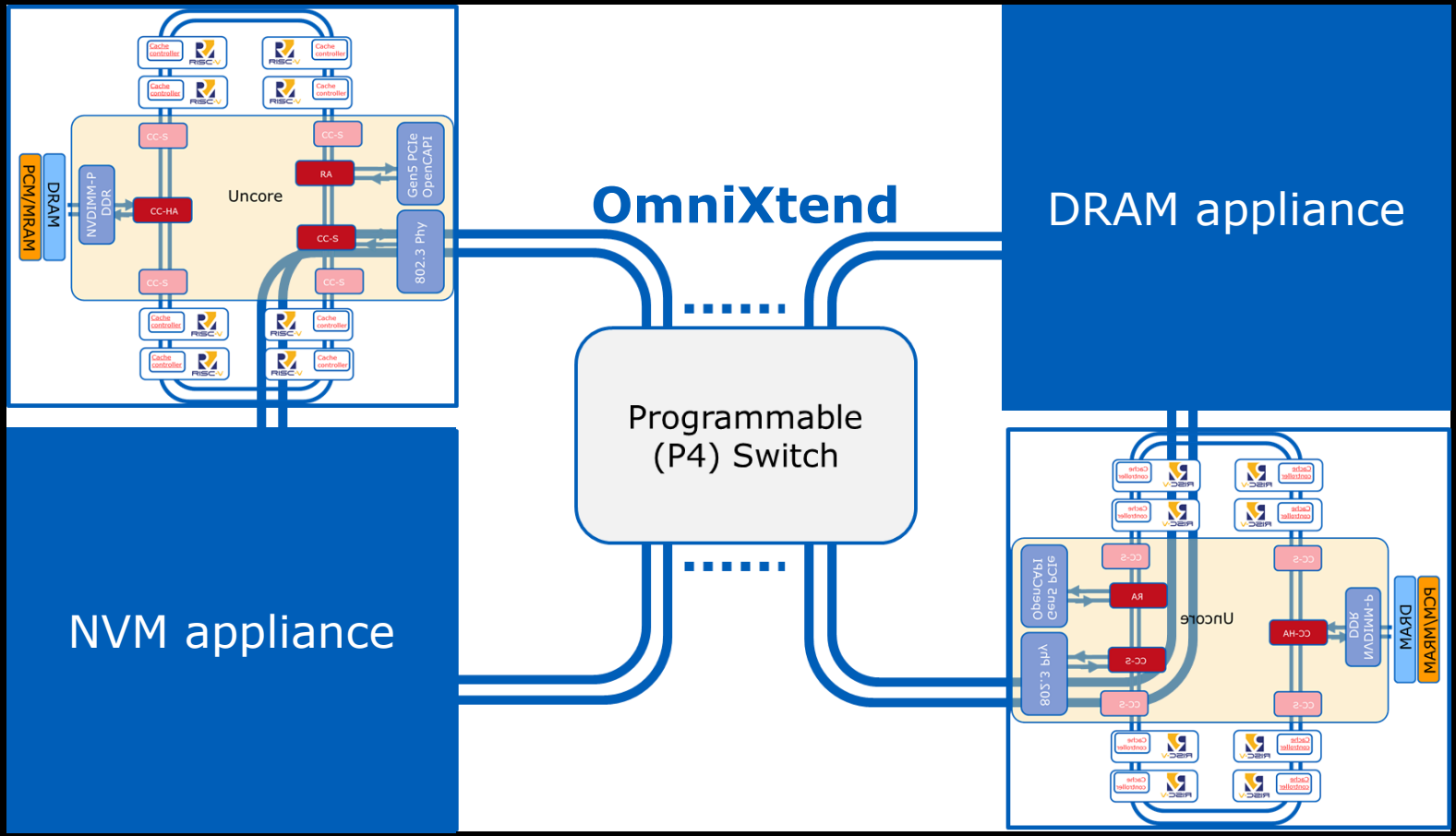
- Multi-threaded, multi-core CPU:
  - 1) Medium performance, OOO RISC-V Core for general purpose OS and software applications
  - 2) Standardized and open JEDEC interface architecture (NVDIMM-P) for high density emerging non-volatile memories
  - 3) Support for high bandwidth and low latency accelerator interfaces:
    - Supporting machine learning and inference engine accelerators
  - 4) Support for standardized memory protocol fabric – e.g.OmniXtend - Tilelink over 802.3:
    - Allowing coherent scale-out for memory-centric architectures

Memory-centric architecture with OmniXtend



- Allows large numbers of RISC-V compute nodes to connect to universally shared memory (NUMA) – standardized and open coherence protocols
- Enables memory appliance, aggregation/disaggregation

Memory-centric architecture with OmniXtend



- Allows large numbers of RISC-V compute nodes to connect to universally shared memory (NUMA) – standardized and open coherence protocols
- Enables memory appliance, aggregation/disaggregation

Planned open source contributions

Subject to internal approvals

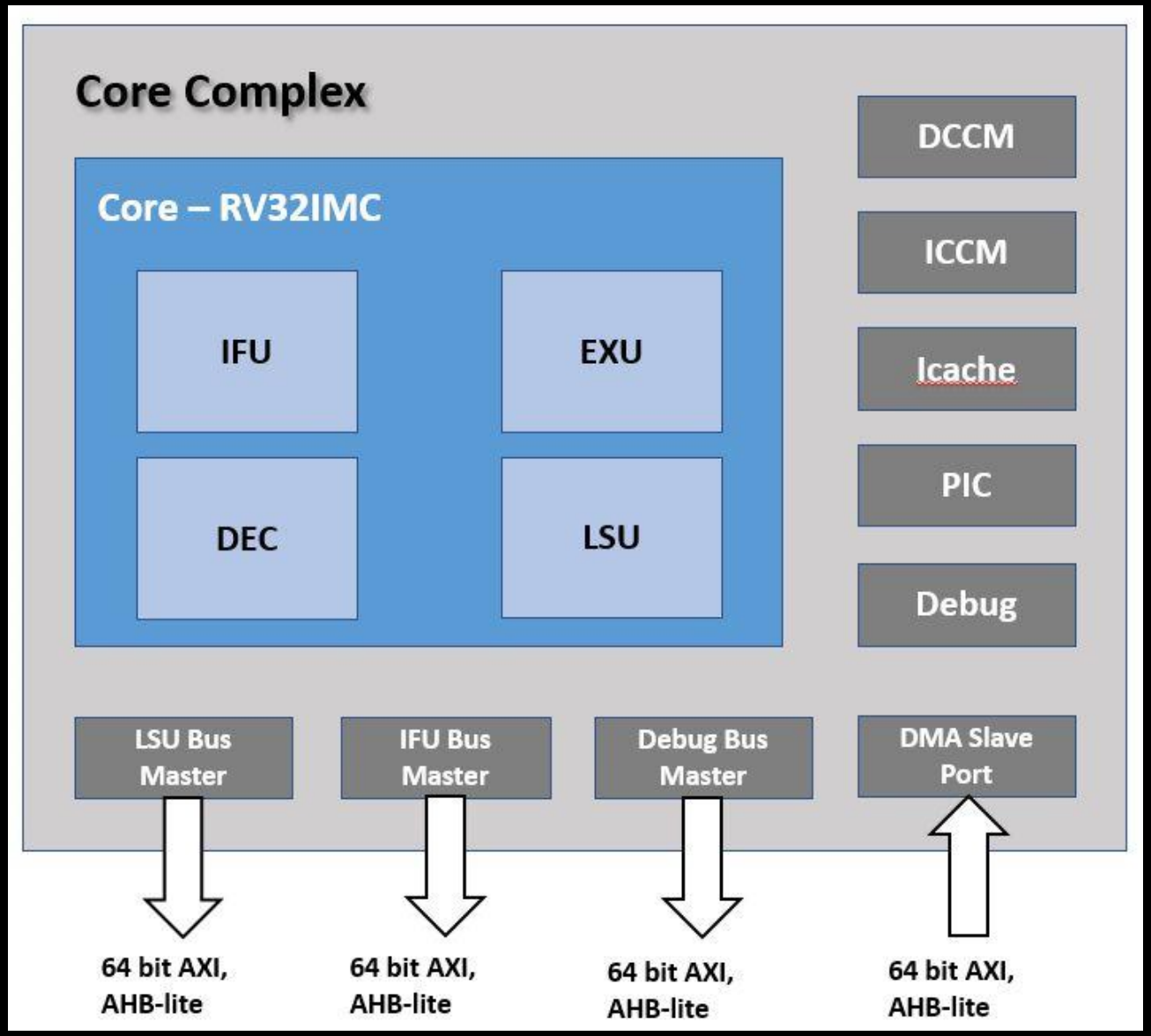
- Production grade instruction set simulator SweRV ISS™ (December 4th 2018):
  - https://github.com/westerndigitalcorporation/swerv-ISS
- RTL of the 2-way superscalar Western Digital SweRV Core™ (January 24th 2019):
  - https://github.com/westerndigitalcorporation/swerv
- OmniXtend reference implementations:
  - https://github.com/westerndigitalcorporation/omnixtend
  - Specification (December 4th 2018)
  - Switch P4 implementation (t.b.d)
  - Board designs (t.b.d.)
- RISC-V Firmware development toolchain

#LetDataThrive

SweRV Core™: Western Digital's First RISC-V Core

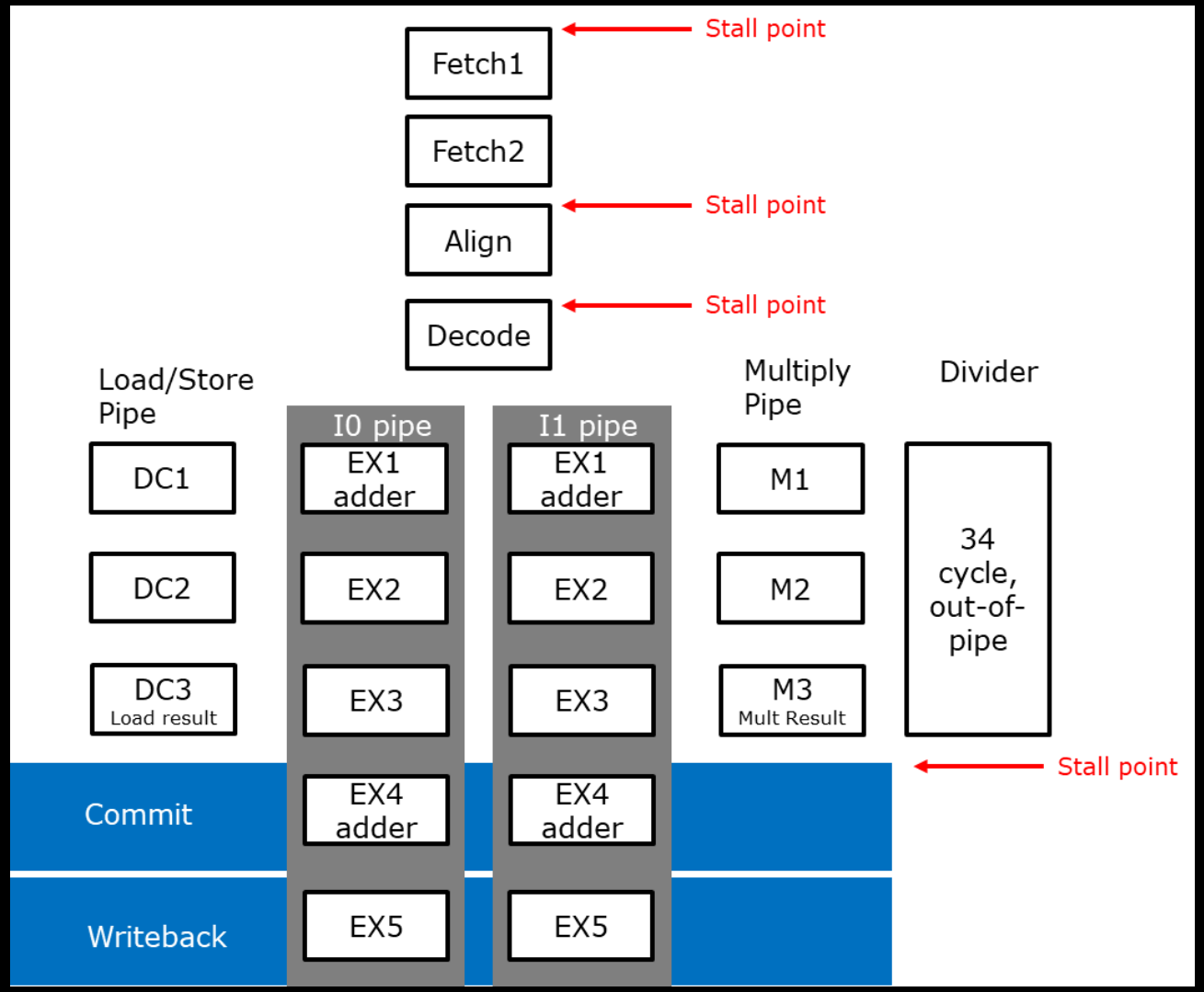
Robert Golla

SweRV Core™ Complex



- RISCV 32IMC Core
  - First internally developed RISCV core
- RISCV debug support
- Programmable Interrupt Controller
  - Support for up to 255 external interrupts
- AHB-lite, AXI bus support
- Frequency target
  - 1 GHz at SSG process corner
- Technology
  - TSMC 28 nm

SweRV Core Microarchitecture



- 9 stage pipeline
- 4 stall points
  - Fetch1
    - Cache misses, line fills
  - Align
    - Form instructions from 3 fetch buffers
  - Decode
    - Decode up to 2 instructions from 4 instruction buffers
  - Commit
    - Commit up to 2 instructions / cycle
- EX pipes
  - ALU ops statically assigned to I0, I1 pipes
  - ALU's are symmetric
- Load/store pipe
  - Load-to-use of 2
- Multiply pipe
  - 3 cycle latency
- Divide pipe
  - 34 cycles, out-of-pipe

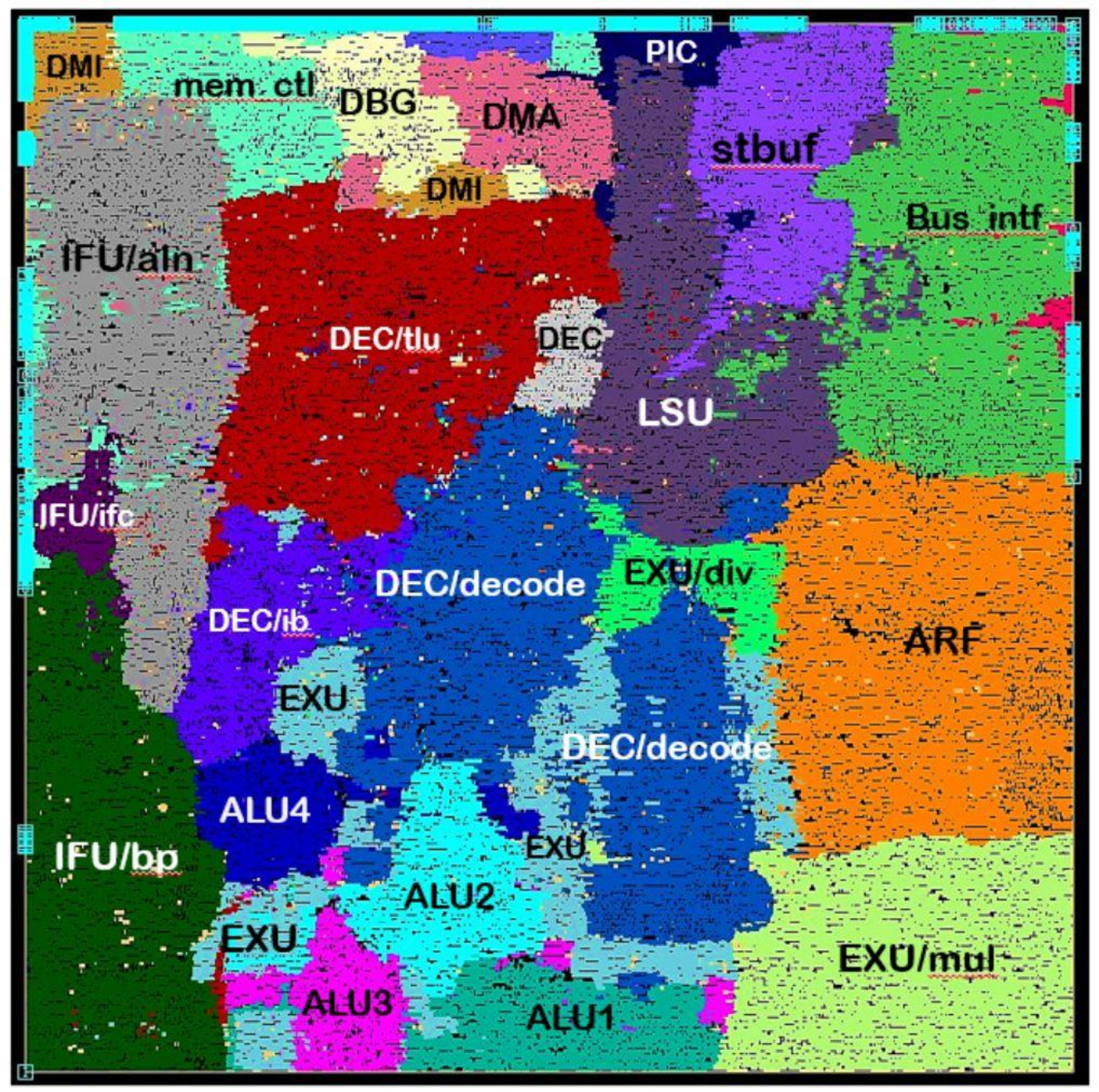
SweRV Core Branch Prediction / Branch Handling

- Branch direction is predicted using GSHARE algorithm
  - XOR of global branch history and PC
    - Used to lookup branch direction in branch history table (BHT)
  - PC hash
    - Used to lookup branch target in branch target table (BTB)
- Branches that hit in the BTB result in 1 cycle branch penalty
- Branches that mispredict in primary alu's result in 4 cycle branch penalty
- Branches that mispredict in secondary alu's result in 7 cycle branch penalty

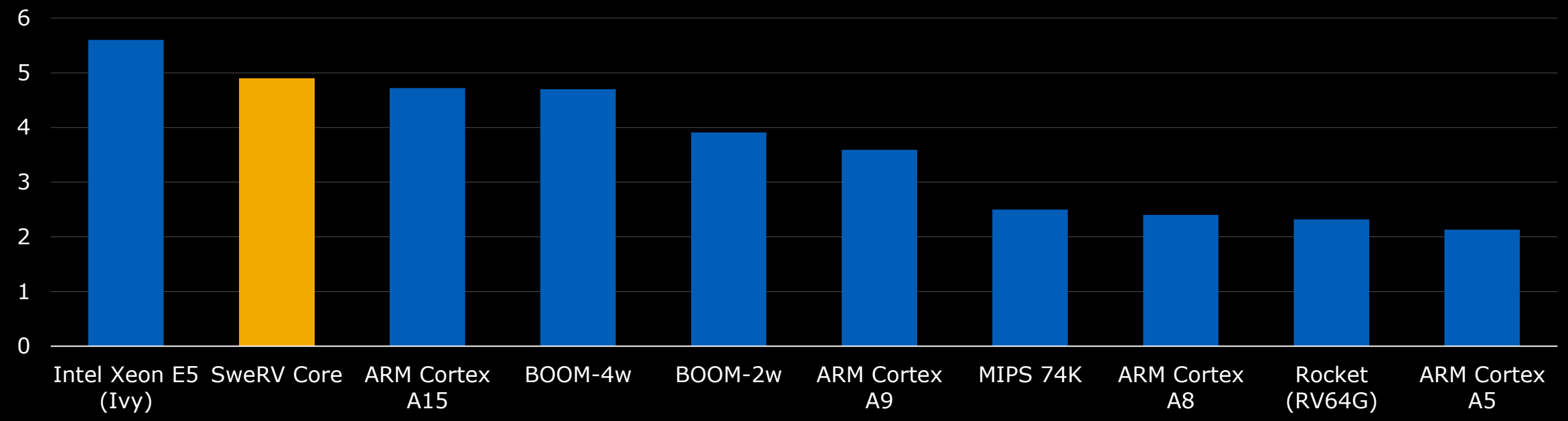
FETCH1		B		T1			T2			T3
FETCH2			B							
ALIGN				B						
DECODE					B					
E1/DC1						B				
E2/DC2							B			
E3/DC3								B		
E4/COMMIT									B	
E5/WRITEB										B

SweRV Core Physical Design

- TSMC 28 nm
  - 125 C, SVT, 150 ps clock skew
- SSG corner w/out memories
  - 1 GHZ
    - .132 mm^2
  - 800 MHZ
    - .100 mm^2
  - 500 MHZ
    - .093 mm^2
- TT corner w/out memories
  - 1 GHZ
    - .092 mm^2
  - 800 MHZ
    - .091 mm^2
  - 500 MHZ
    - .088 mm^2



SweRV Core Performance



- 4.9 CoreMark/MHz
  - Additional performance gains are possible with compiler optimizations
  - Multi-threaded/multi-core results are always renormalized to a single execution context
- 2.9 Dhrystone MIPs/MHz
  - Using optimized strcpy function

CoreMark data from C.Celio, D.Patterson, K.Asanovic,https://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-167.pdf

Driving Momentum

Western Digital ships in excess of
1 Billion cores per year
…and we expect to **double that.**

Western Digital.

#LetDataThrive

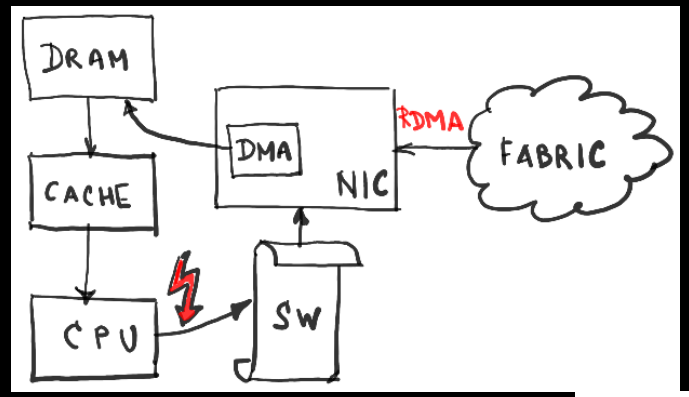
OmniXtend™: direct to caches over commodity fabric

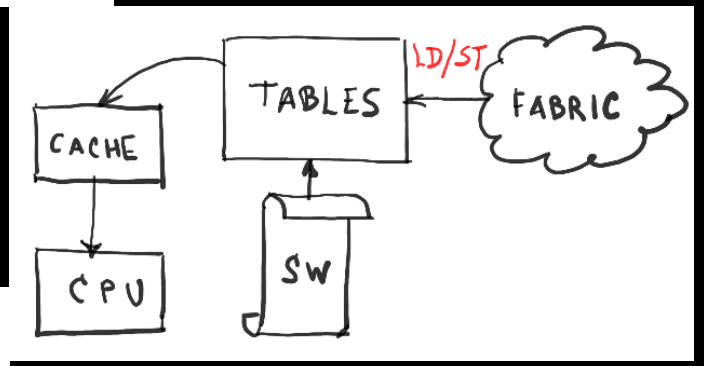
Dejan Vucinic

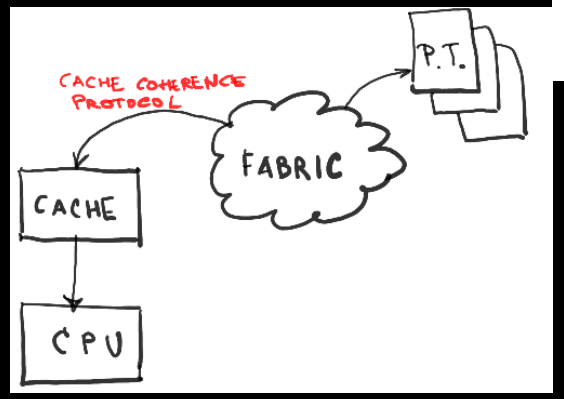
Western Digital.

Emergence of memory fabric

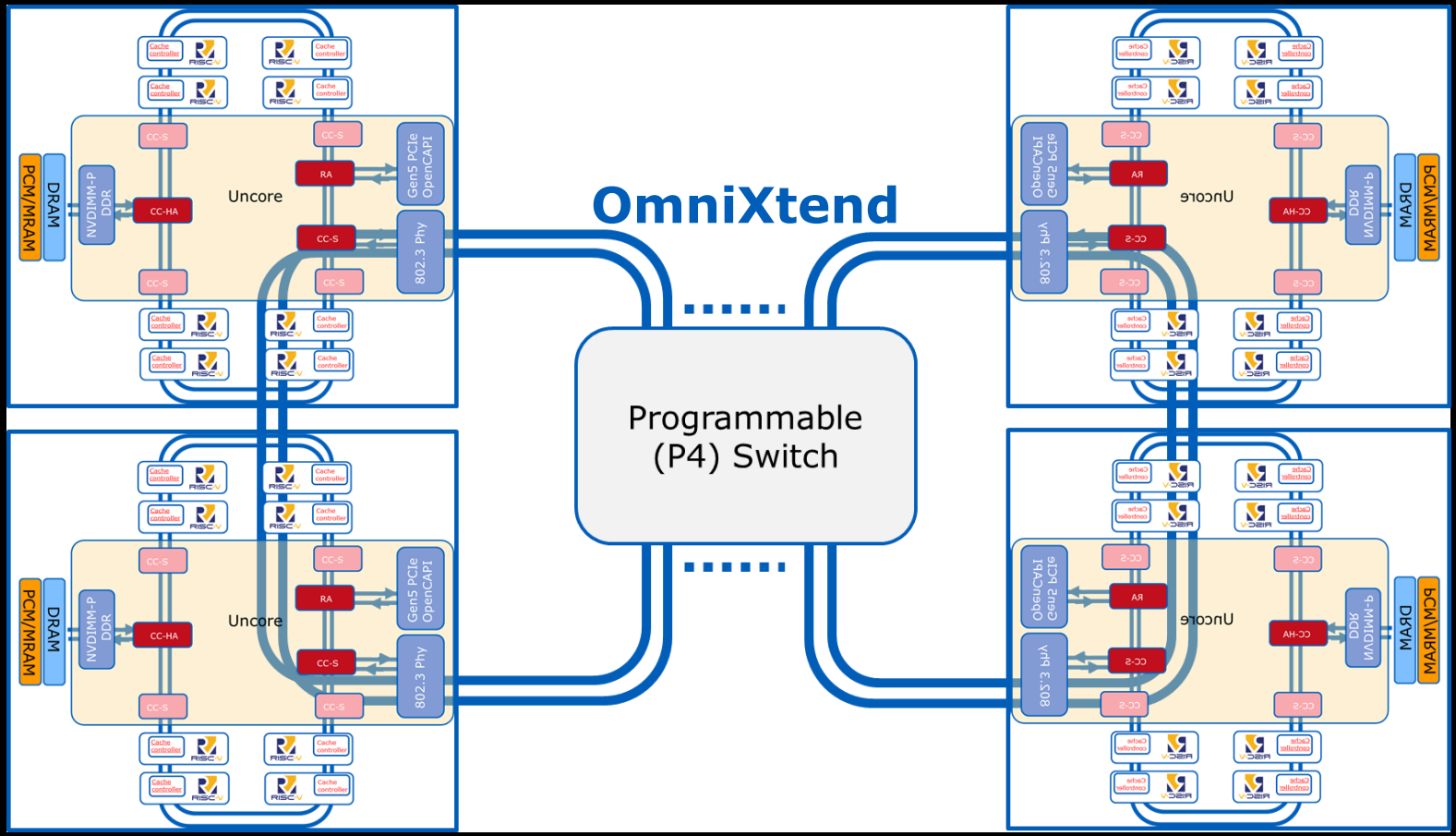
- Memory fabric may mean different things to different people:
  - Page fault trap leading to RDMA request (incurs context switch and SW overhead)
  - Global address translation management in SW, leading to LD/ST across global memory fabric
  - Coherence protocol scaled out, global page management and no context switching





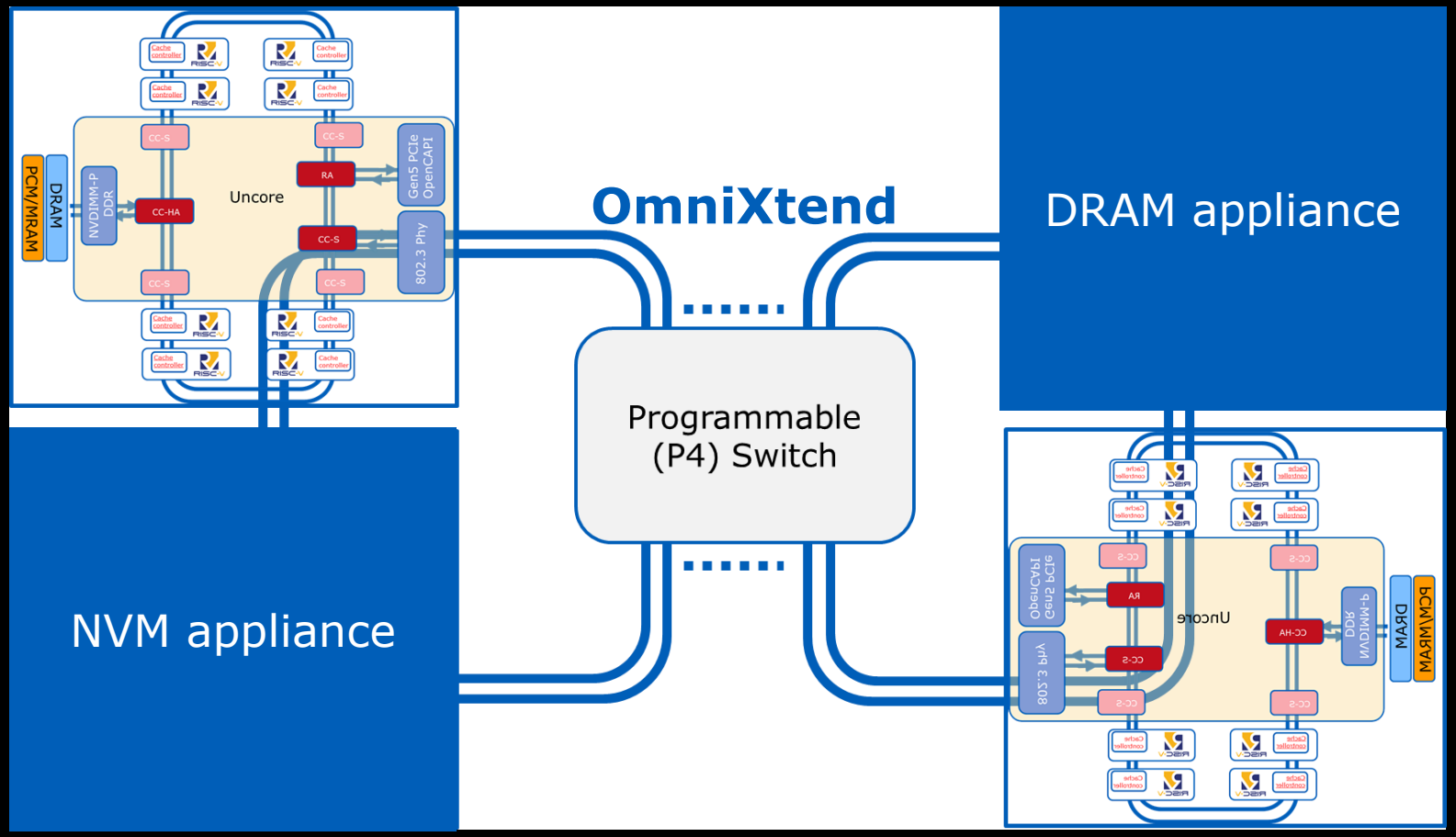


Memory-centric architecture with OmniXtend



- Allows large numbers of RISC-V compute nodes to connect to universally shared memory (NUMA) – standardized and open coherence protocols
- Enables memory appliance, aggregation/disaggregation

Memory-centric architecture with OmniXtend



- Allows large numbers of RISC-V compute nodes to connect to universally shared memory (NUMA) – standardized and open coherence protocols
- Enables memory appliance, aggregation/disaggregation

OmniXtend memory-centric fabric architecture

- Replaces Ethernet L2 with serialized TileLink messages
  - Keeps standard 802.3 L1 frame, interoperates with Barefoot Tofino and future OTS Ethernet switches
  - Custom frames are parsed and processed in P4 language
  - Enables stateful message processing inside the switching fabric
  - Supports innovation required for RAS
  - FPGA or ASIC switch; not limited to 802.3
- Protocol translation and modification inside fabric:
  - Requires no new silicon
- 100 Gb/s is available today
  - Clear roadmap to 200 and 400 with 56Gb PAM4 and x8

2.2.1. TileLink Header Format

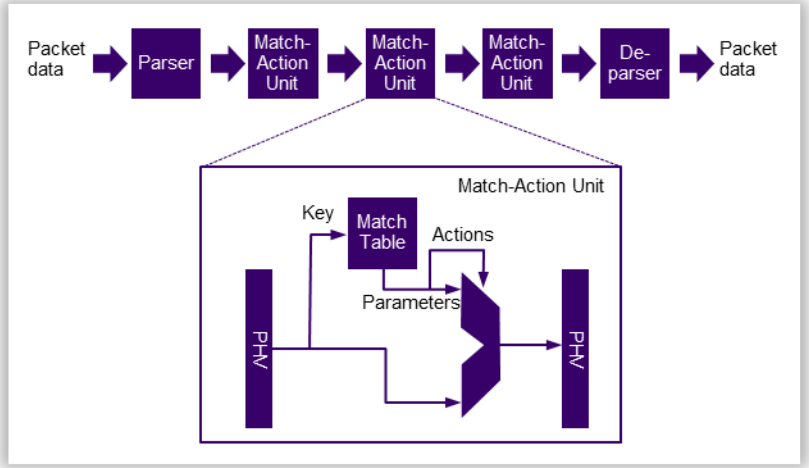
Position	Width	Name	Description
31:16	16	Source	TileLink transaction ID
15:13	3	Domain	Ordering domain ID
12: 9	4	Size	Logarithm of number of bytes in transaction
8: 6	3	Param	Parameter
5: 3	3	Opcode	Operation code, identifies type of message
2: 0	3	Format	Channel number 0-5 (A-F)

802.3 Ethernet packet and frame structure

Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46–1500 octets	4 octets	12 octets
Layer 2 Ethernet frame			← 64–1522 octets →						
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 oct. →

P4 example: OmniXtend programmable switch

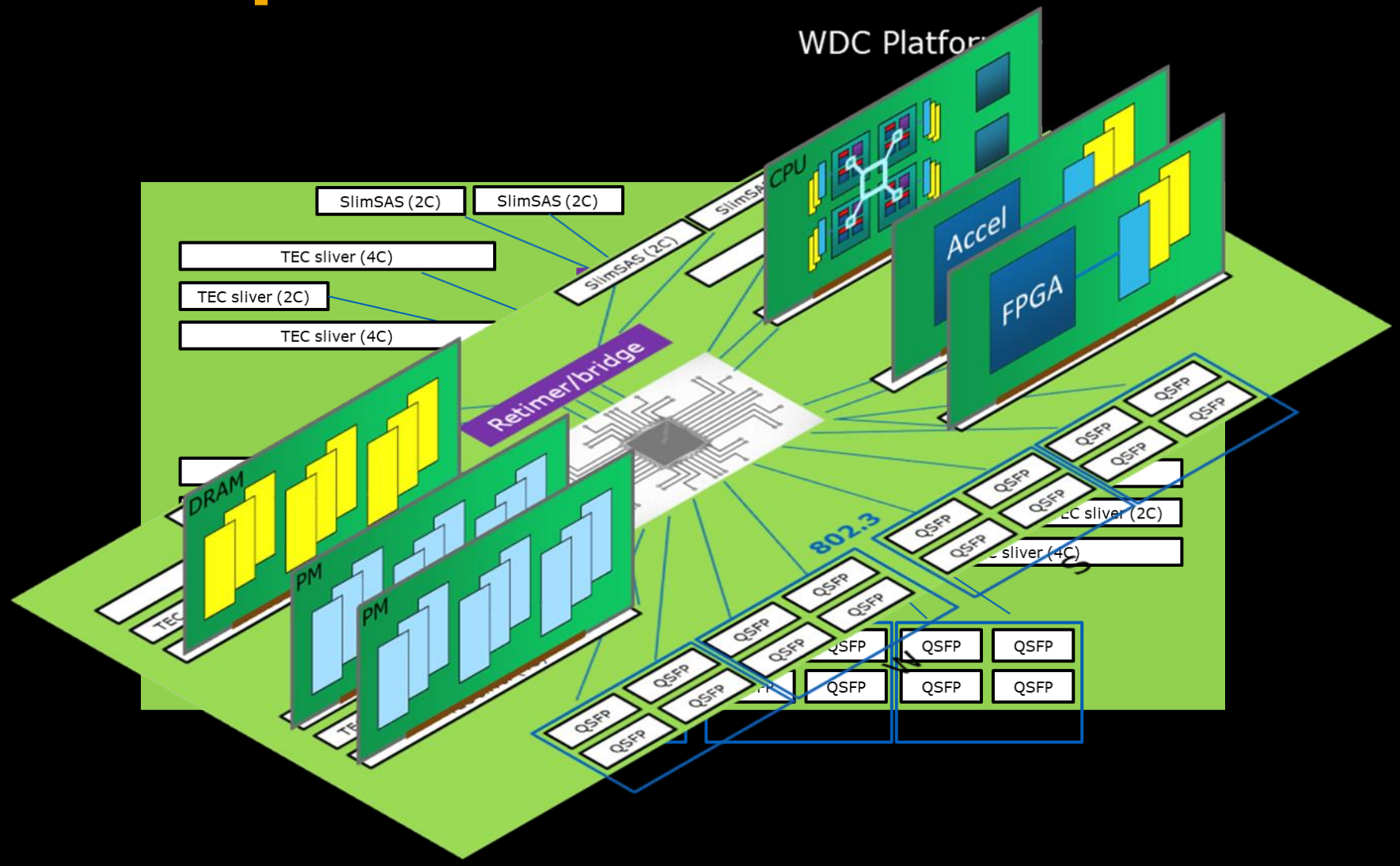
- Barefoot Tofino ASIC (or FPGA with e.g. Xilinx SDNet):
  - 64-port 100 GigE switch, 6.4 Tbit/s aggregate throughput, < 400 ns latency
  - Supports P4 HDL, successor to OpenFlow enabling protocol innovation
  - Describe TileLink message format in P4
  - Match-Action Pipeline (a.k.a. “flow tables”) enables line-rate performance
  - Modifications to coherence domains, protocols require no new silicon




```
/* -*- P4_16 -*- */

#include <core.p4>
#include <v1model.p4>

/*
 * Define the headers the program will recognize
 */

/*
 * This is a custom protocol header for the Gen-Z packets.
 * We'll use ethertype 0x1234 for is (see parser)
 */

const bit<16> P4GENZ_P2P_CORE_ETYPE = 0x1234;
const bit<1>  P4GENZ_P2P_END2END_ETYPE = 0x0;
const bit<1>  P4GENZ_P2P_LINKLOCAL_ETYPE = 0x1;

/*
 * Defined the headers of Gen-Z packets
 *
 */

header p4genz_p2p_corereq_t {
    bit<5>    p4genz_pcrc;
    bit<1>    p4genz_oc_0;
    bit<1>    p4genz_o;
```

```
parser MyParser(
    packet_in              packet,
    out   my_headers_t   hdr,
    inout my_metadata_t   meta,
    inout standard_metadata_t standard_metadata)
{
    state start {
        packet.extract(hdr.ethernet);
        transition select(hdr.ethernet.etherType) {
            P4GENZ_P2P_CORE_ETYPE : check_p4genz;
            default               : accept;
        }
    }


    //check L of Gen-Z packet
    state check_p4genz {
        transition
select(packet.lookahead<p4genz_p2p_corereq_t>().p4genz_l) {
            P4GENZ_P2P_END2END_ETYPE  : parse_p4genz;
            default                   : accept;
        }
    }

    state parse_p4genz {
        packet.extract(hdr.p4genz_p2p_corereq);
        transition accept;
    }
}
```

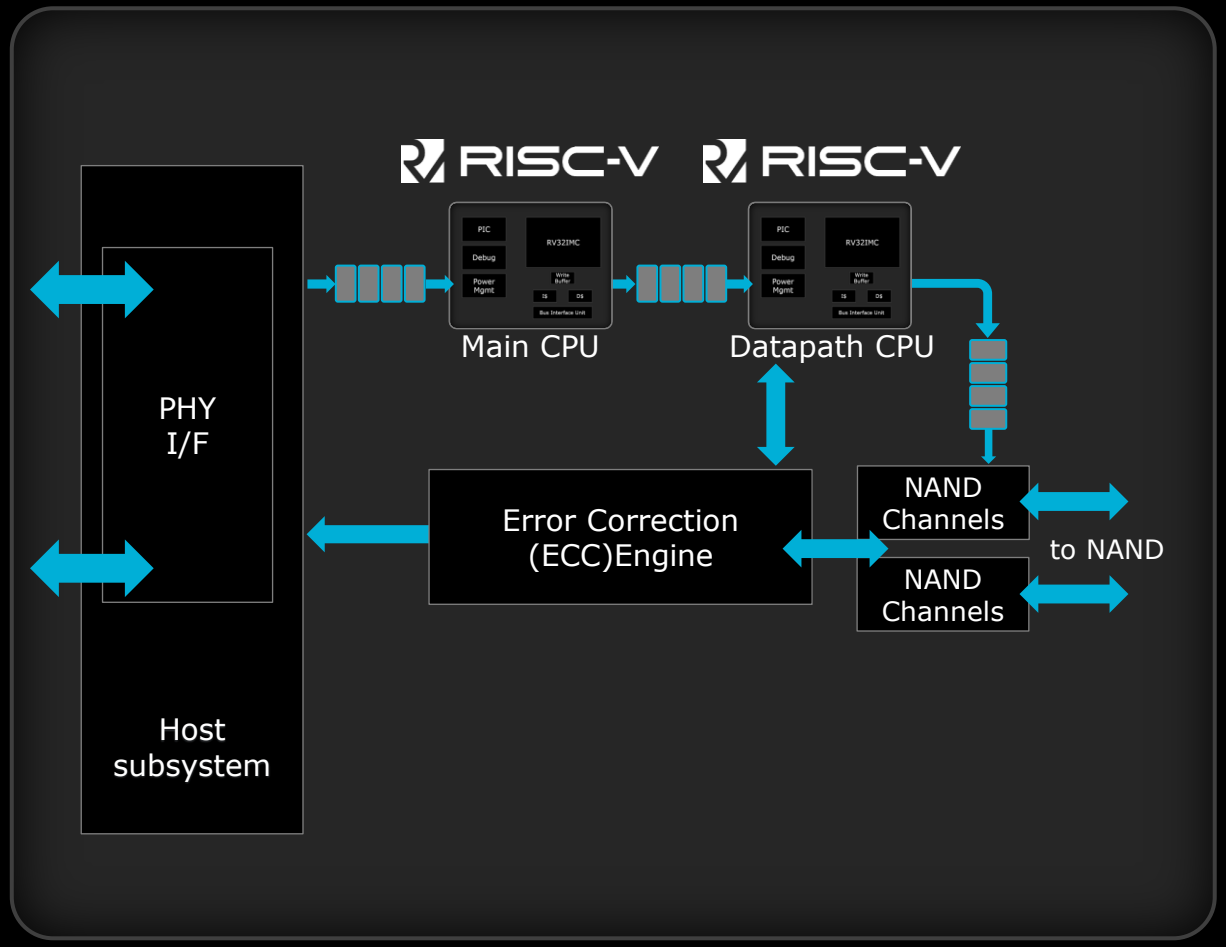
Memory fabric protocol OmniXtend innovation platform



#LetDataThrive

BACKUP

NAND Controller SoC applications



- Multi-purpose SoC for consumer SSD applications
- First RISC-V based SoC for NAND controller applications
- Advantages:
  - Full advantage of open source software ecosystem for RISC-V
  - Instruction optimization for NAND media handling
  - Freedom of power and performance optimization for end application