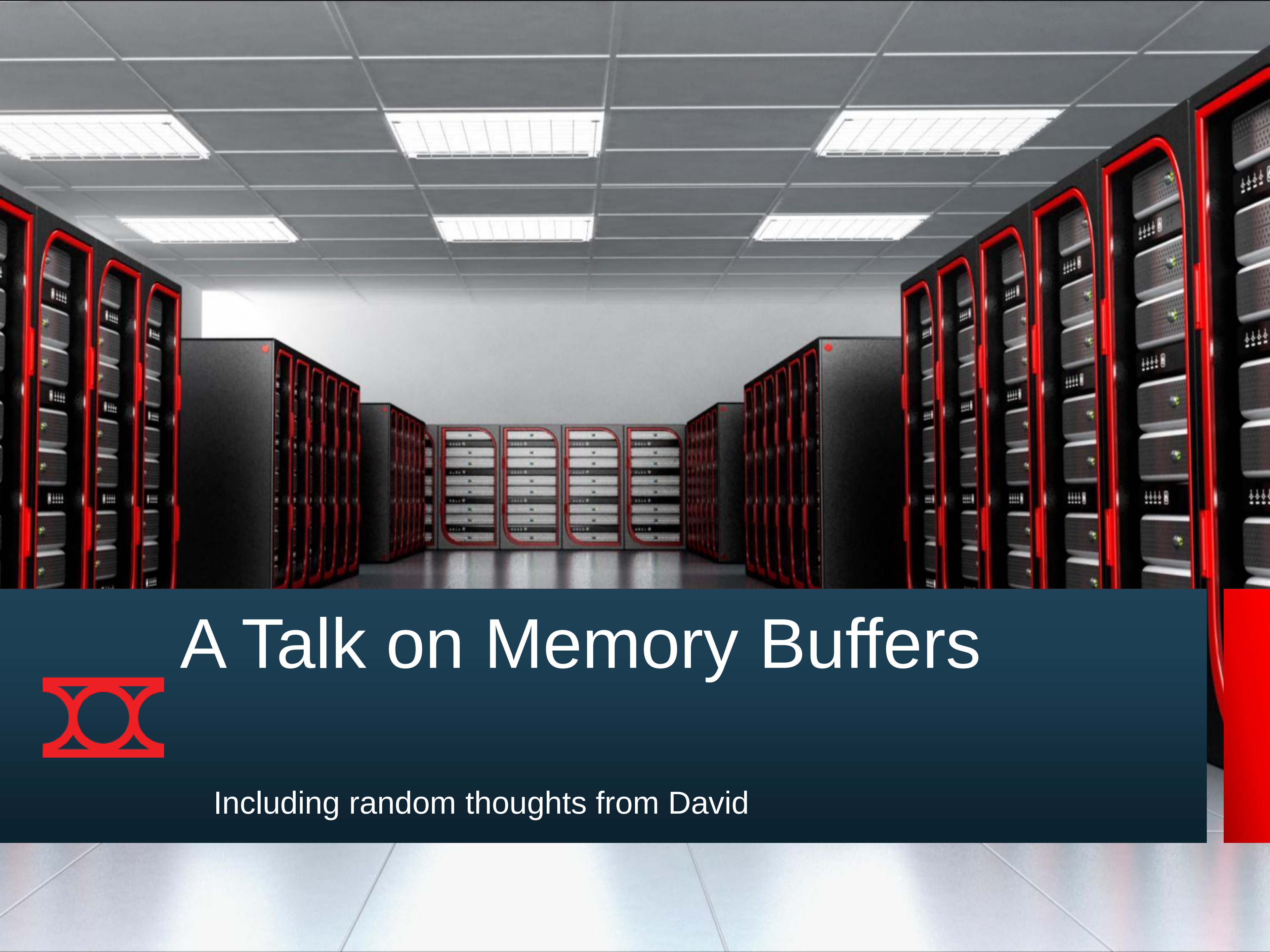


# A Talk on Memory Buffers

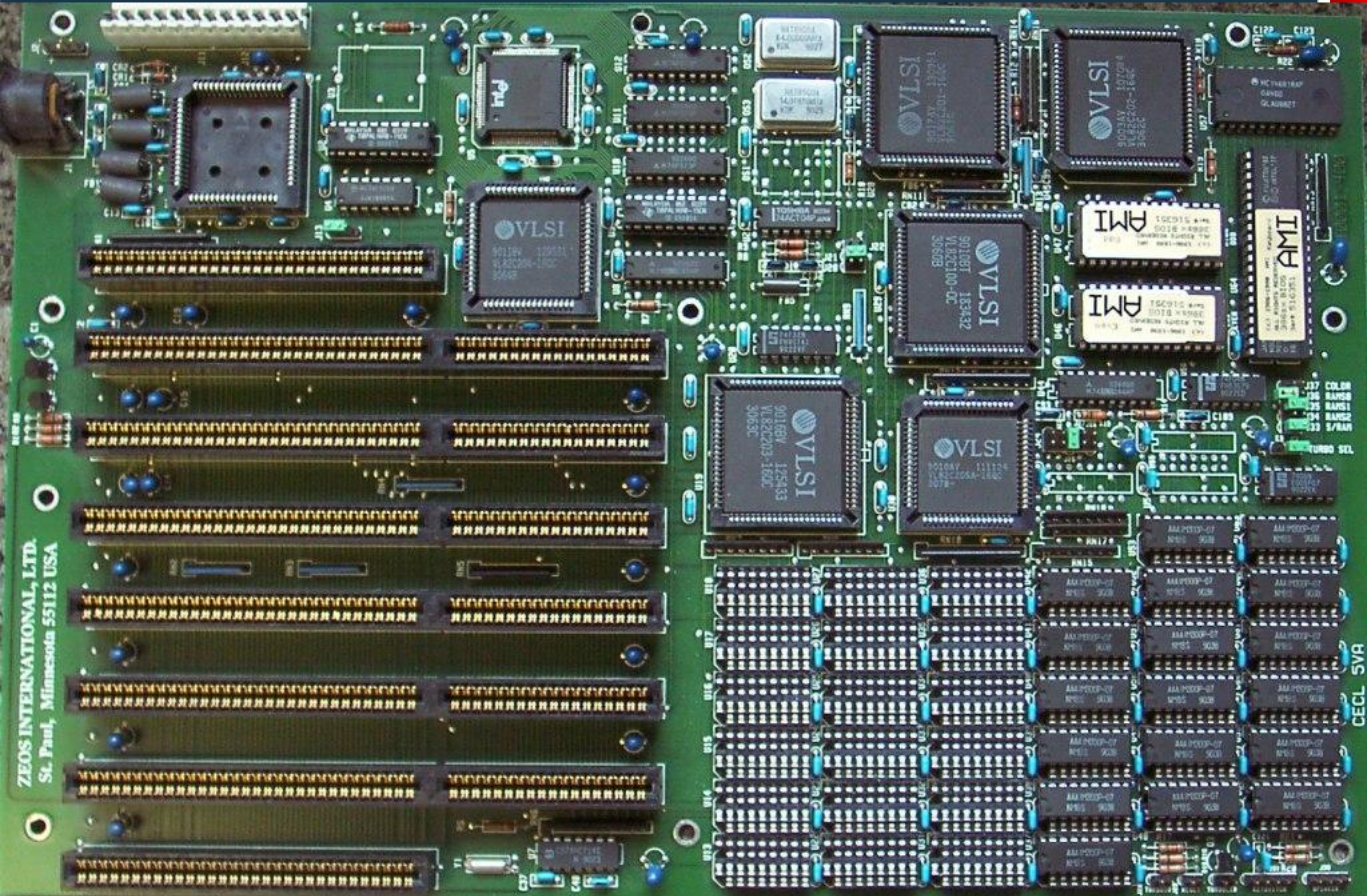
Including random thoughts from David

## Before We get Started – Random Thoughts

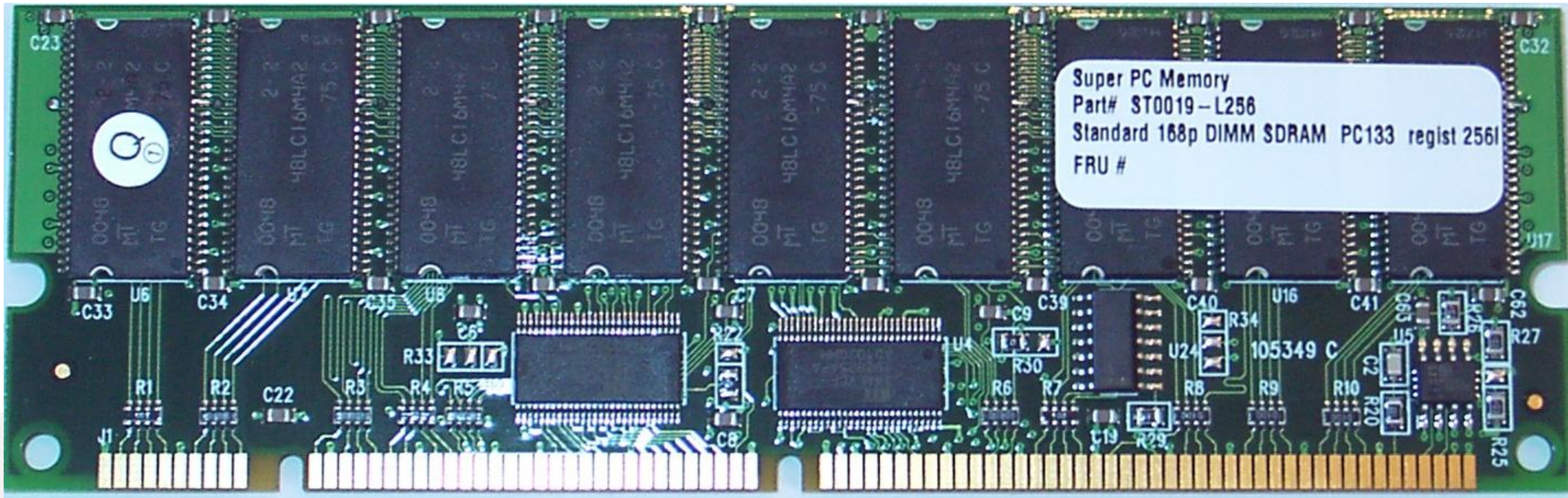
- For Memory Systems: More == Faster
  - DRAM as a caching layer to reduce SSD/Disk access
  - (Workload-specific) speedup may be attained from increasing capacity
  - More (DRAM) capacity at lower bandwidth and/or slightly longer latency may still be faster
- Difference Between Academia and Industry
  - Academia: Write papers about using 2X resources to get 10% speed up
  - Industry: Figure out how to reduces costs by 50% and keep 90% of performance
- Cheaper is better than better
  - (Alternative statement) Barely good enough and cheaper is better than much better and a bit more expensive

## Quick Historical Review

# Memory Expansion – 20+ years ago



# Then, Memory Modules

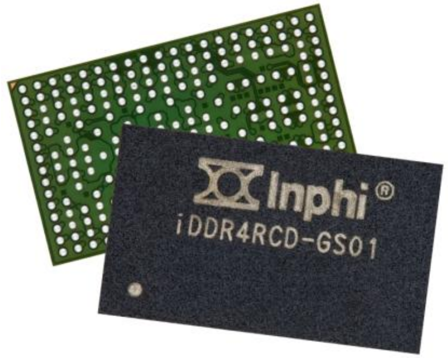


## Inphi ExacTik® Memory Interface Products

DDR2 Memory Interface

DDR3 Memory Interface

DDR4 Memory Interface

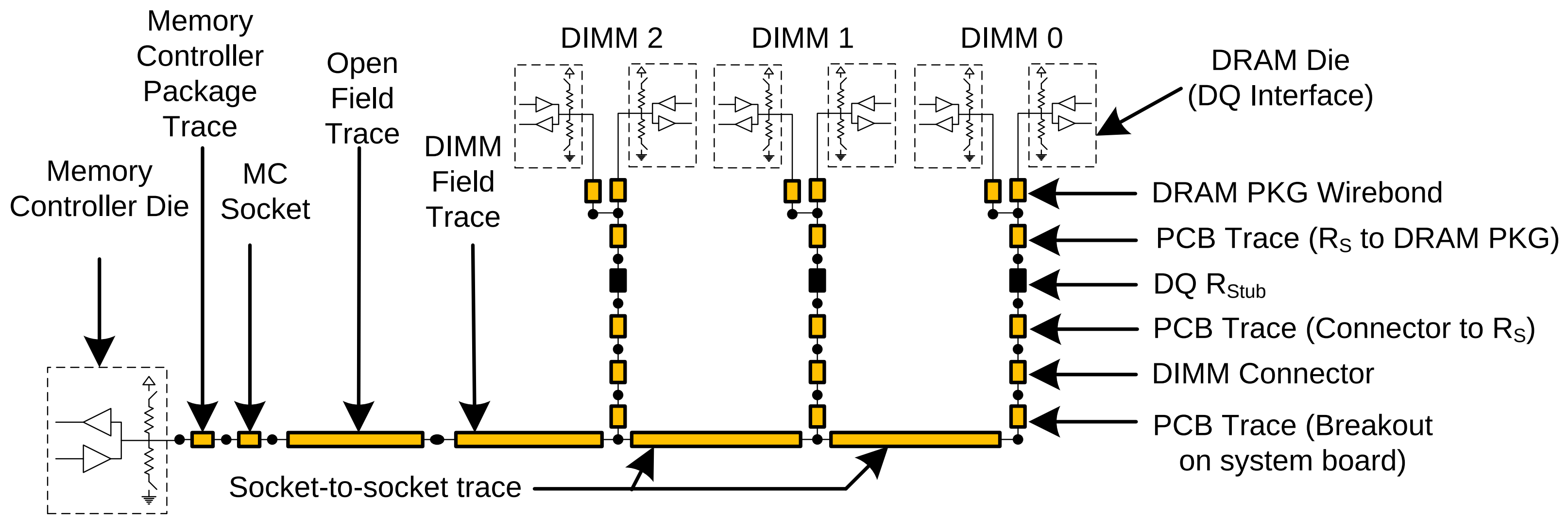


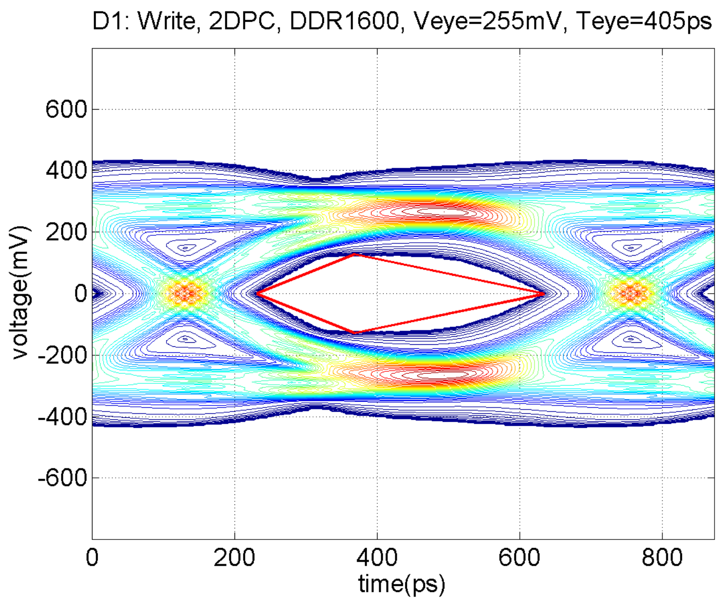
Isolation Memory Buffer

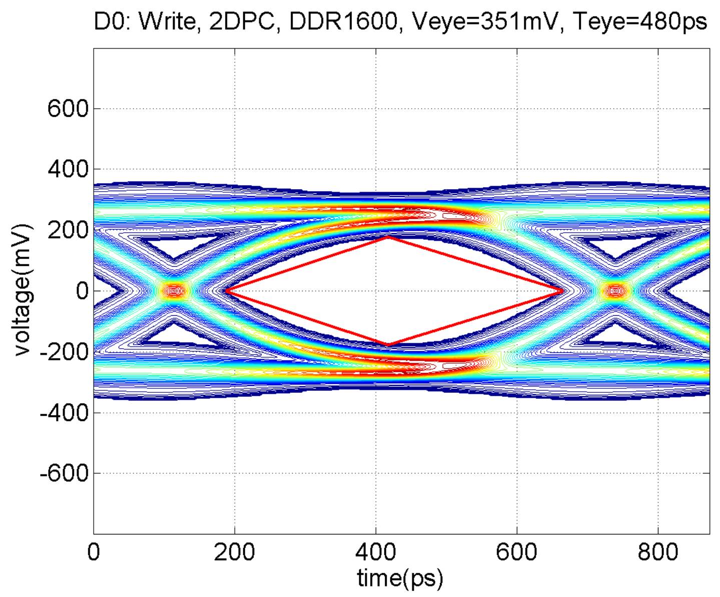
NVDIMM iSC
Storage Controller

2H LRDIMM

# Not So Simple Anymore





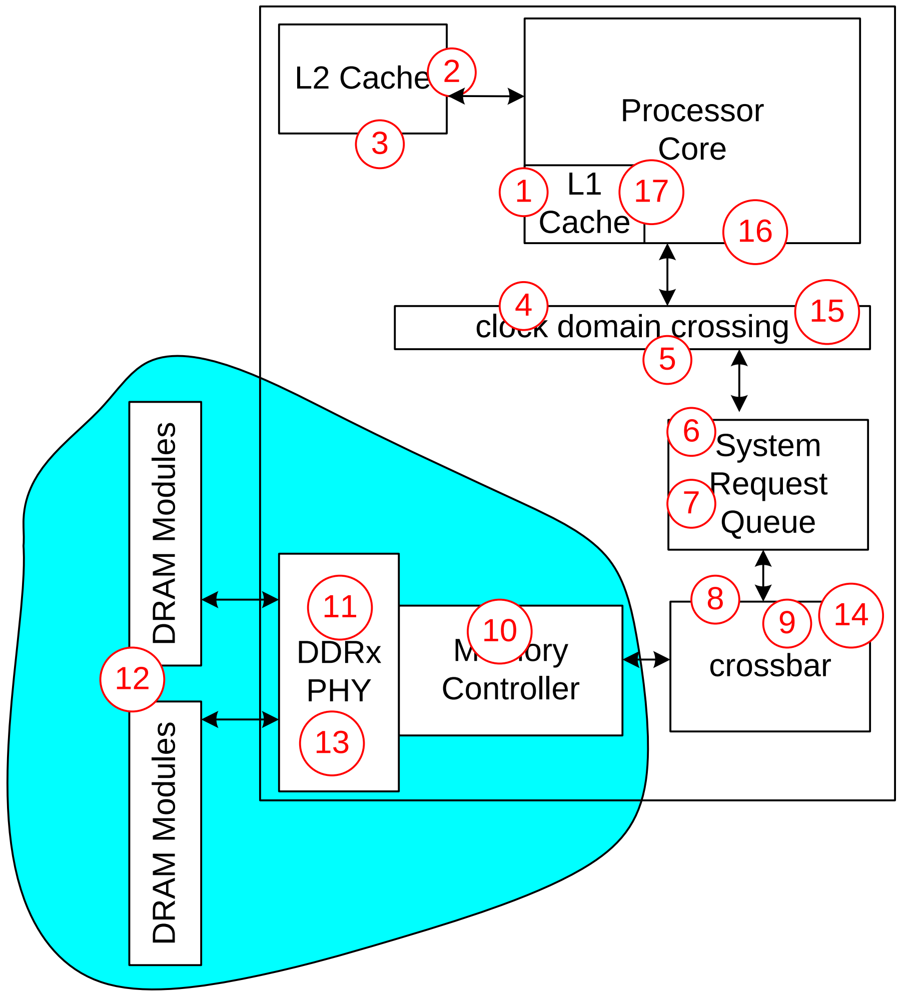


## Why Memory Buffering?
## Benefits of Buffering

## Benefits

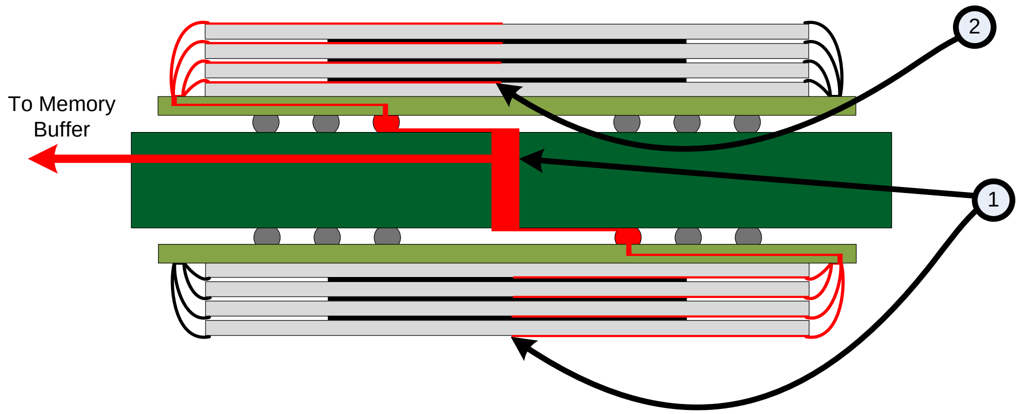
- More Capacity
  - One controller to connect to many DRAM devices
- More Bandwidth
  - Multiple loads (DRAM devices) slow down bus.
  - Buffer-and-re-drive cleans up system signal integrity, enabling higher operating data rate
- Better RAS
  - Buffers can check correctness of commands/data
- Additional Logic Interface to do “interesting things”, e.g.
  - (Flash-backed, DRAM Access) NVDIMM
  - NVRAM-only NVDIMM

## Memory Buffering could Reduce Operating Latency



- LRDIMM adds ~3 ns over RDIMM at same frequency
- But, parts of memory controller operates at same frequency as memory
- At higher frequency, latency through memory controller is lower
- LRDIMM total memory access latency may be lower than RDIMM, depending on frequency of comparison
- e.g. 3 DPC LRDIMM @ 1333 MT/s has lower idle latency than 3 DPC RDIMM @ 800 MT/s
  - 85 ns vs 92 ns

# Isolating DRAM DQ could Also Reduce Power



1. From via to DRAM pad, typically ~10 mm distance. Non-target ODT may not be needed at low datarates
2. Even when non-target ODT is used, only bottom-most die needed to provide termination

- Isolated channels means that
  - Termination values may be changed/reduced or even disabled
  - Power management schemes and termination schemes may be separately optimized

## But . . . Memory Buffers should behave and cost like a wire

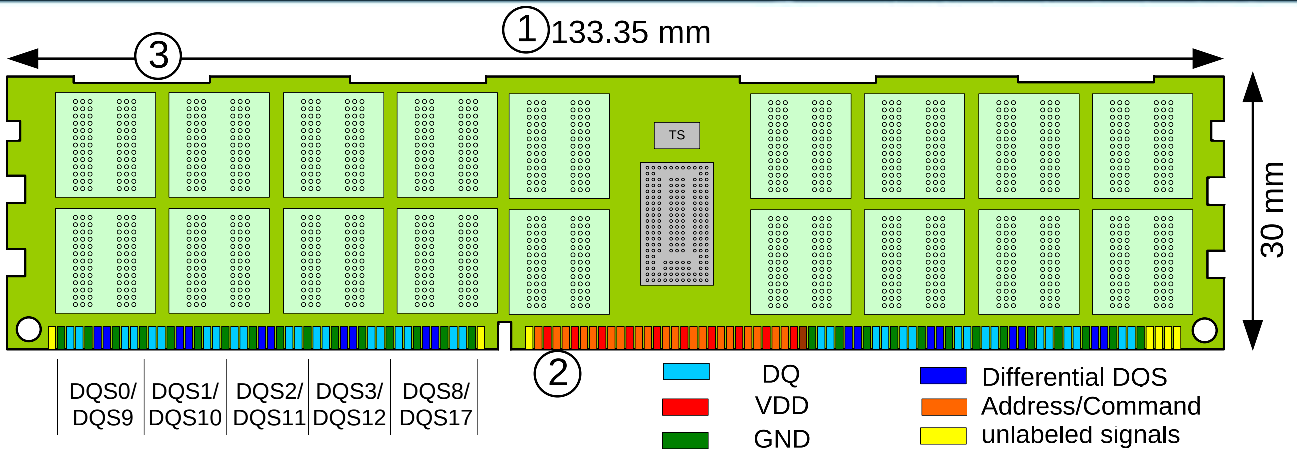
- Zero cost
  - Should be tiny and simple to design
- Zero latency
  - No junk (logic) in data path – logic slows things down
- Zero footprint
  - DIMM’s and System Board should carry “useful stuff”, not chips that just buffer and re-drive signals
- Zero power
  - Ideally . . .

## How Do You Build (Architect) a Memory Buffer?

## So What Do You Do When you want a Memory Buffer?

- Define Scope
  - Improve pin-capacity-bandwidth of CPU interface?
  - Interface conversion for compatibility
  - On-DIMM or On-system-board application
- How many pins do you need for this chip or chipset?
  - More pins -> Larger package -> higher cost & larger footprint
  - Will it fit?
- Speed target and power budget

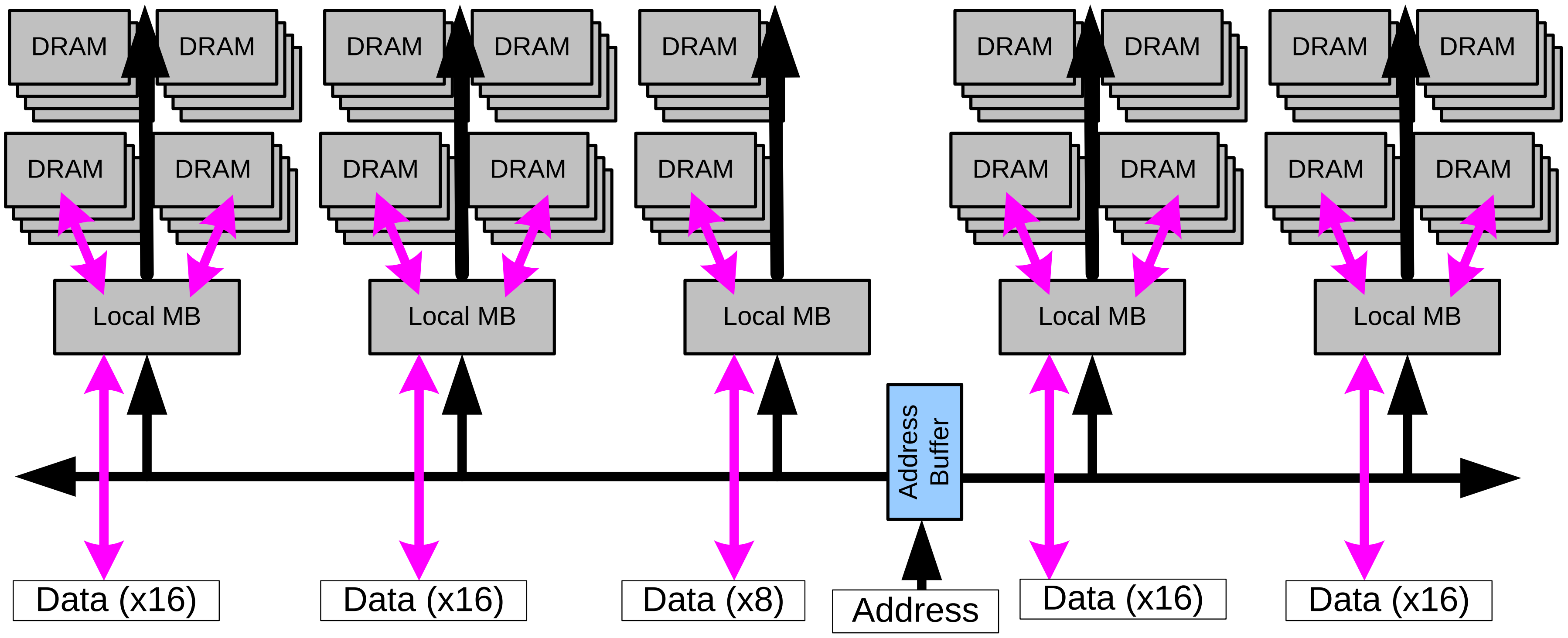
## Anatomy of a DDR3 (LP) DIMM



1. 133.35 mm x ~30 mm
   - Component area ~125 mm x ~25 mm
   - 40 “DRAM sites”, ~11.5 mm x ~11.5 mm each
2. 240 pins, 1 mm pitch
   - DQ GND reference, Addr/Cmd Vdd reference
   - 2:1 signal-to-ground ratio
   - Same interface for UDIMM, RDIMM, LRDIMM
3. Notches for heat spreader attachment

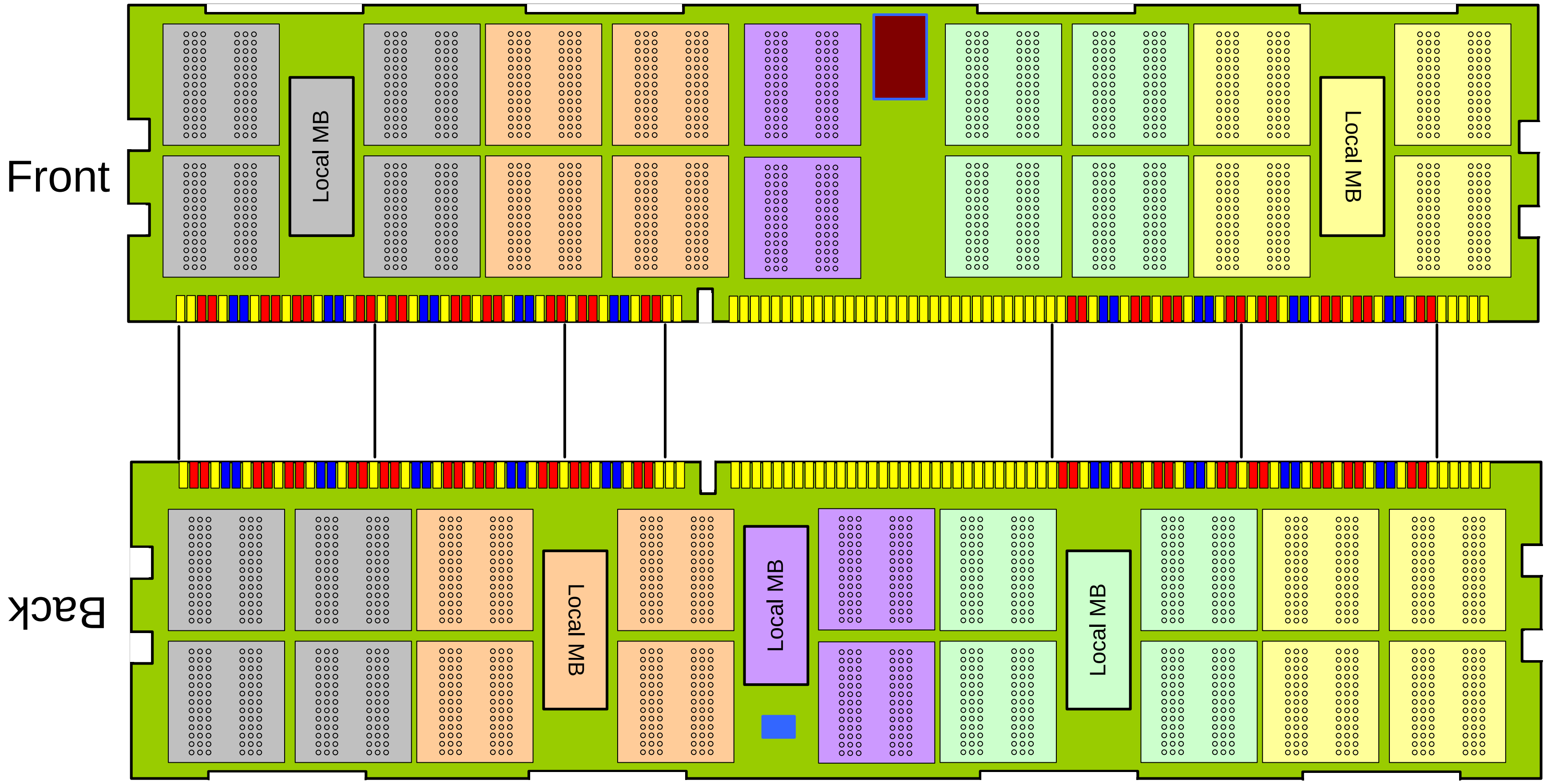
- Find some free space to put some buffers!

# This is a “Nice Architecture”



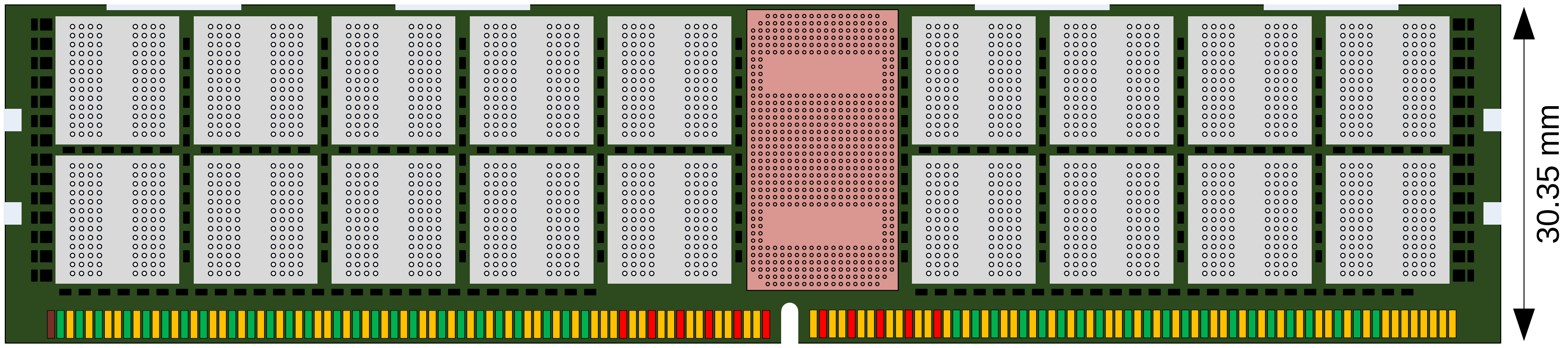
- Register re-drives address and control to local MB
- Local MB re-drives address to DRAM on local (x16) slice
- Local MB re-drives data between DRAM and host on local slice
- Simple chips, everything localized, no training needed

# But . . .



- Local Buffers take a lot of precious DIMM real estate
  - Constrains DRAM package size

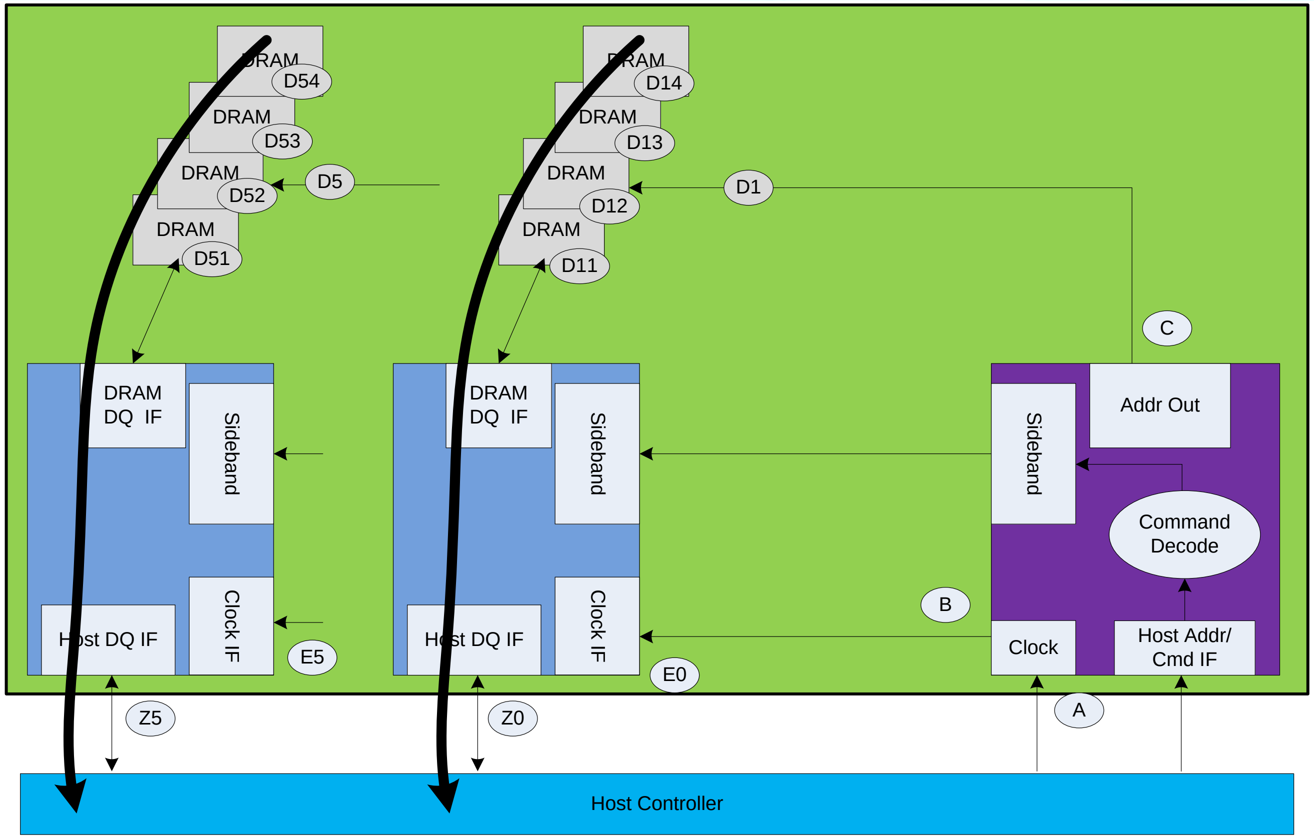
# Inphi iMB – Single Chip Address and Data Buffer



- Winning JEDEC Architecture for DDR3 SDRAM Memory Buffering
- Single Chip
  - Low Cost
  - Low DIMM Surface Area Impact
  - 36 Max Size (11 mm x 11.5 mm) DRAM Devices
  - Long DQ Stub Lengths
    - Not scalable to very high data rates, but . . .
    - High enough – 2 DPC @ 1866/1.5V
- More complicated chip design than previous architecture, lots of training for timing and find phase adjustments

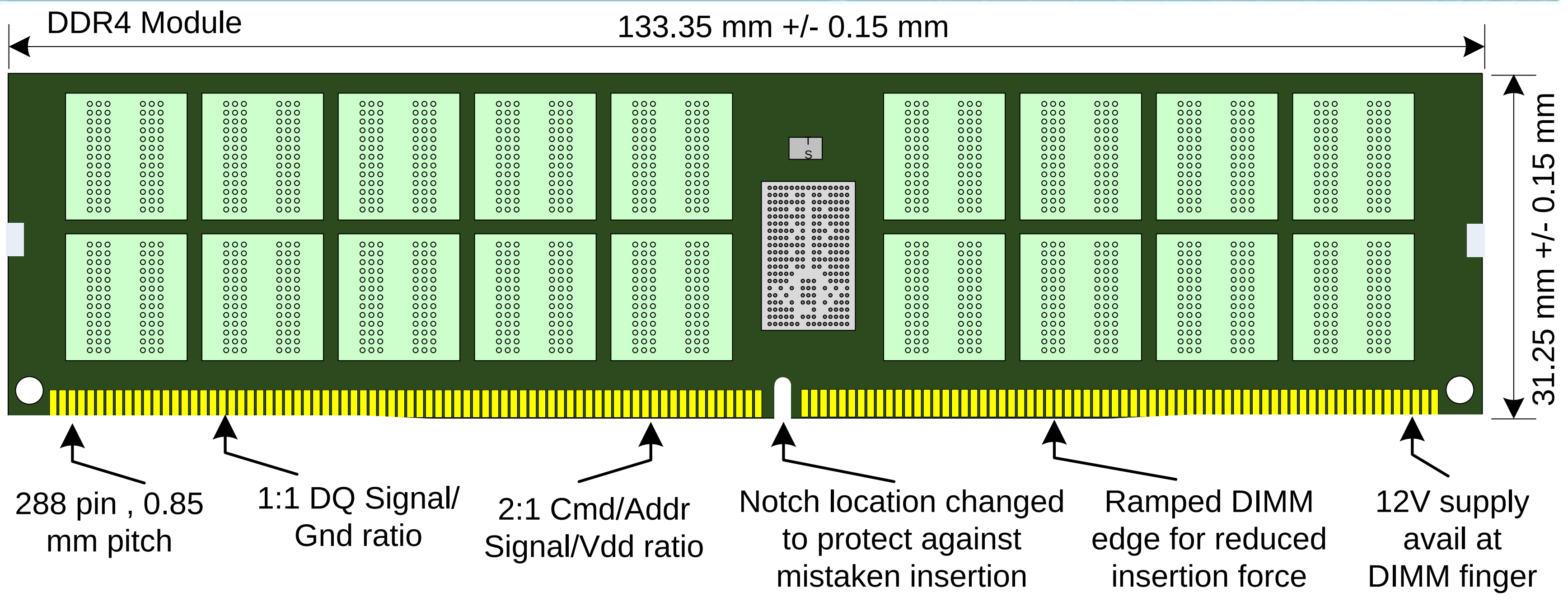
## Example: DDR4 LRDIMM

# DDR4 LRDIMM Chipset Architecture



- RCD (Registering Clock Driver) is the Address and Control Buffer, generates command sequences to Data Buffers (DB)
- Data Buffers must be trained to resolve 3-body synchronization problem (RCD, host MC, DRAM)

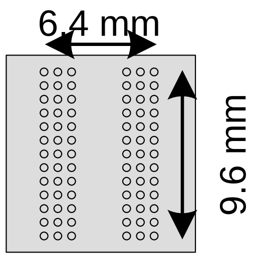
# DDR4 LP DIMM

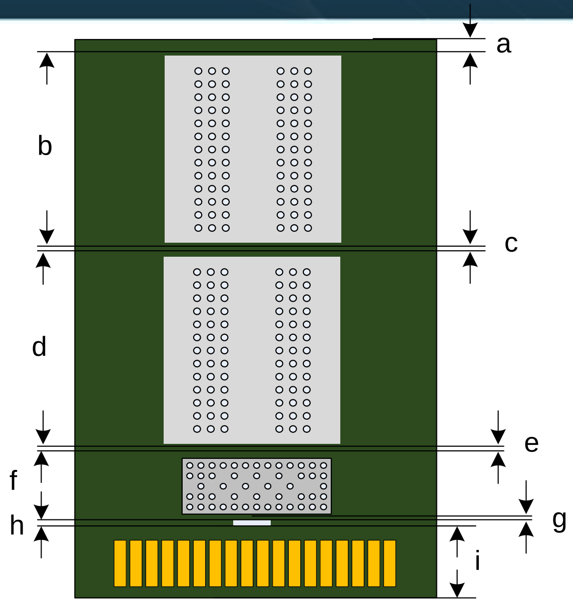


DDR4 Module
133.35 mm +/- 0.15 mm
31.25 mm +/- 0.15 mm
S
288 pin , 0.85 mm pitch
1:1 DQ Signal/ Gnd ratio
2:1 Cmd/Addr Signal/Vdd ratio
Notch location changed to protect against mistaken insertion
Ramped DIMM edge for reduced insertion force
12V supply avail at DIMM finger

- 133.35 mm x 31.25 mm
  - Added 0.9 mm DIMM height for DB, attained by using low seating plane DIMM connector
- 288 pins, 0.85 mm pitch
  - DQ still GND referenced, Addr/Cmd still Vdd referenced
  - 1:1 DQ signal-to-ground ratio

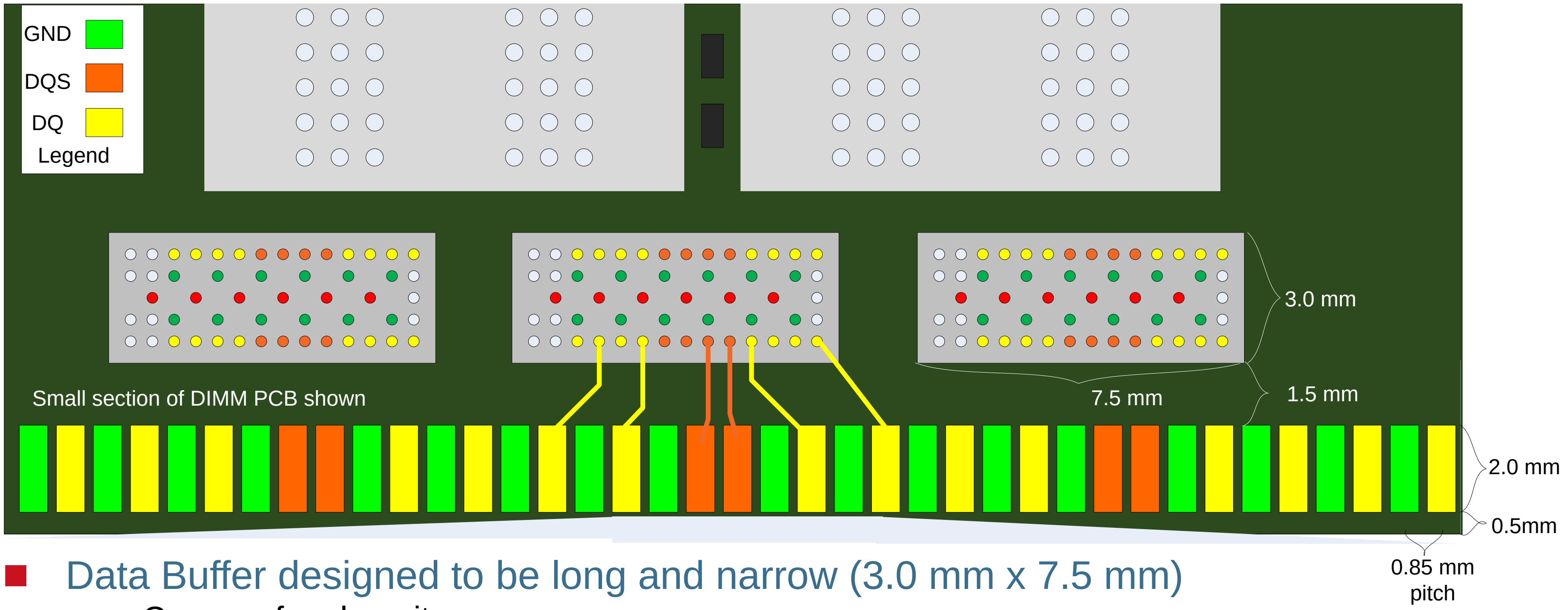
# DRAM Dimension for DDR4 LRDIMM





- Ideally, DRAM Manufactures (SEC, Micron, Hynix) would like to place 36 Max Dimension DRAM Devices on Module
  - 11.0 mm x 11.5 mm or 9 mm x 13 mm
- DRAM devices cannot shrink (much) below 11.5 mm in y dimension due to ball footprint constraint
- Data Buffer (DB) competes with DRAM for area
  - DB needs to be as small as possible

# Data Buffer Size and Placement

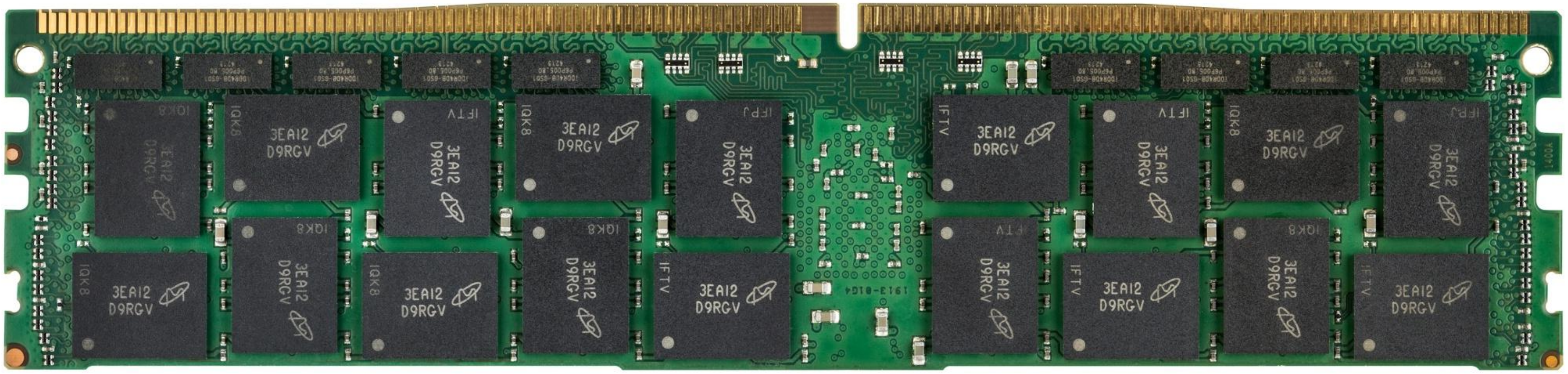


- Data Buffer designed to be long and narrow (3.0 mm x 7.5 mm)
  - Concern for planarity
- Data Buffer placed as close to DIMM finger as possible
  - Short DQ Stubs
- DIMM x4 DQ ports alternate front/back of DIMM
  - E.g. DQ[3:0] on front, DQ[7:4] on back
- The two x4 DQ ports on Data Buffers are interleaved
  - Facilitate routing to DIMM finger
- Note: DB-to-DIMM-finger routing shown without series stub resistor

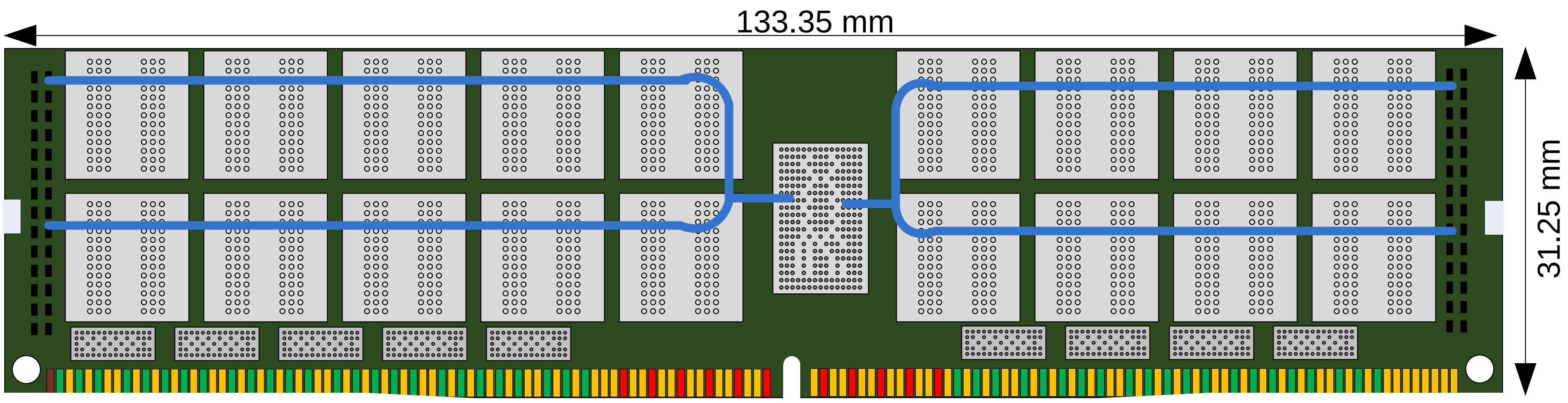
# Inphi-Enabled DDR4 LRDIMM

FRONT

REVERSE

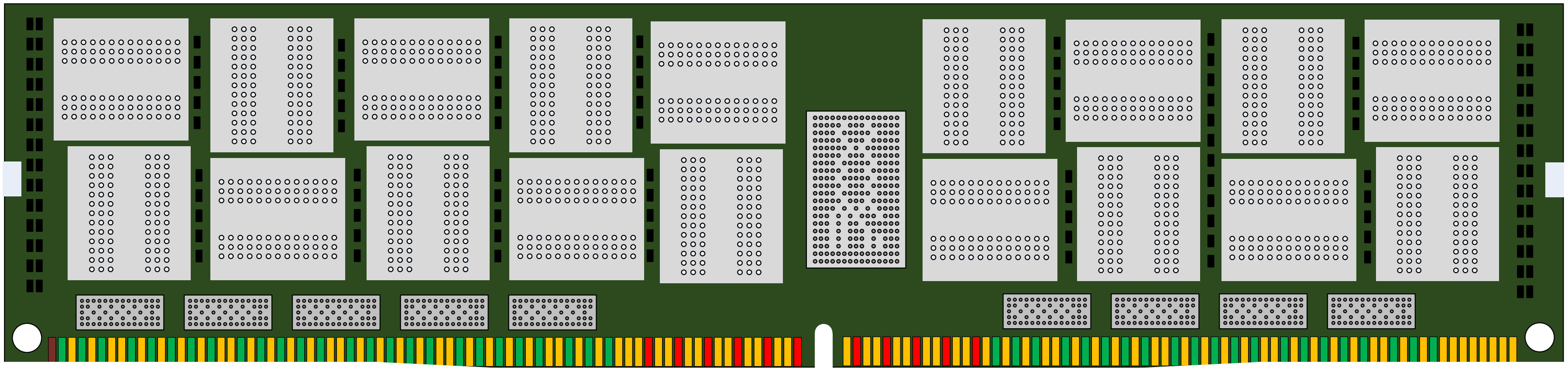


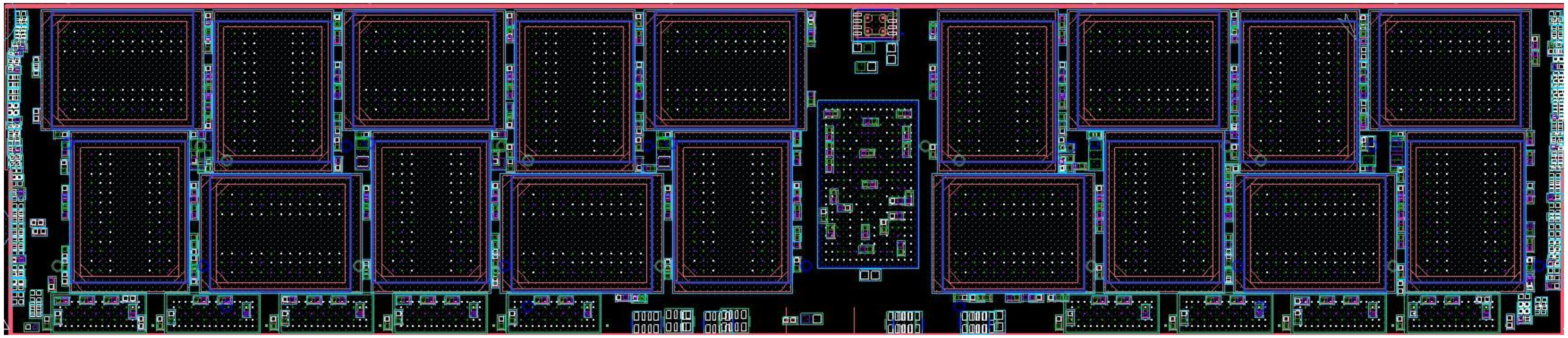
## Ideal LRDIMM Component Placement



- DRAM Devices in “vertical” orientation
- Far easier to route Address, Command, Control and Clock Signals
  - Relatively easier task of path length matching

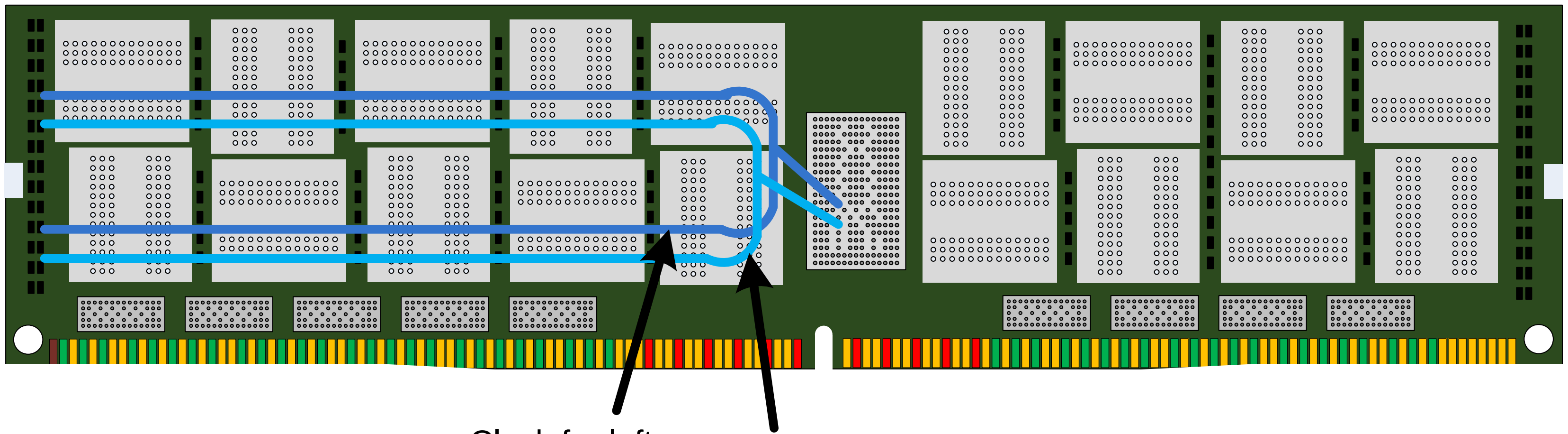
# DDR4 LRDIMM R/C E Component Placement





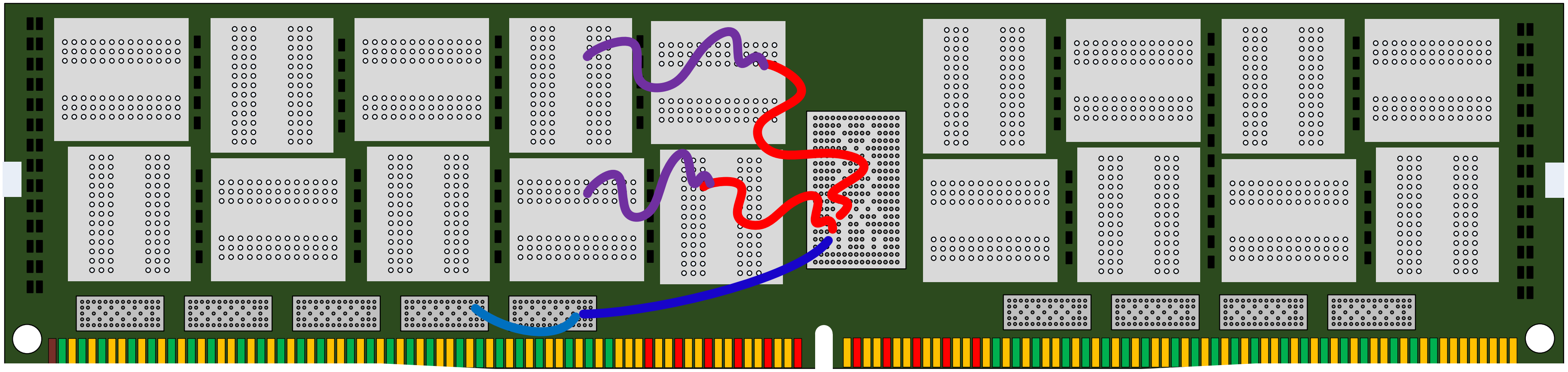
- “Windmill” or “Flower” supports DRAM devices with larger aspect ratios

# LRDIMM R/C E Clock Topology



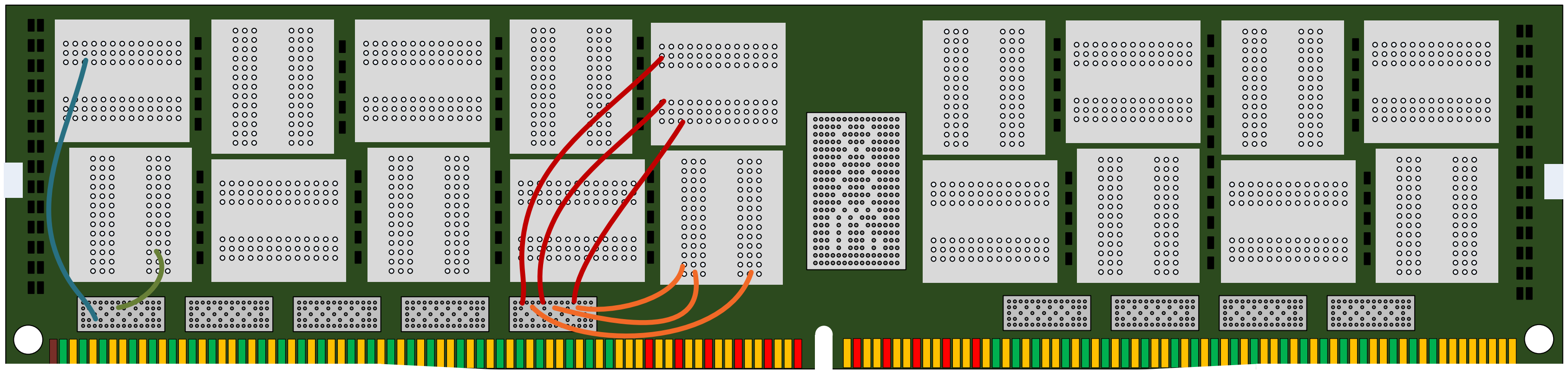
- Four y-clock pairs support
  - Left front : Rank 0 (and Rank 2)
  - Left reverse (back) : Rank 1 (and Rank 3)
  - Right front : Rank 0 (and Rank 2)
  - Right reverse : Rank 1 (and Rank 3)

# DDR4 LRDIMM R/C E Clock Topology



| | |
|---|---|
| Y clock RCD to first DRAM top row | - 35 mm |
| Y clock RCD to first DRAM bottom row | - 35 mm |
| Y clock first DRAM to second DRAM top row | - 28 mm |
| Y clock first DRAM to second DRAM bottom row | – 28 mm |
| DB Clock RCD to First DB | - 31.3 mm |
| DB Clock First DB to Second DB | - 10.6 mm |

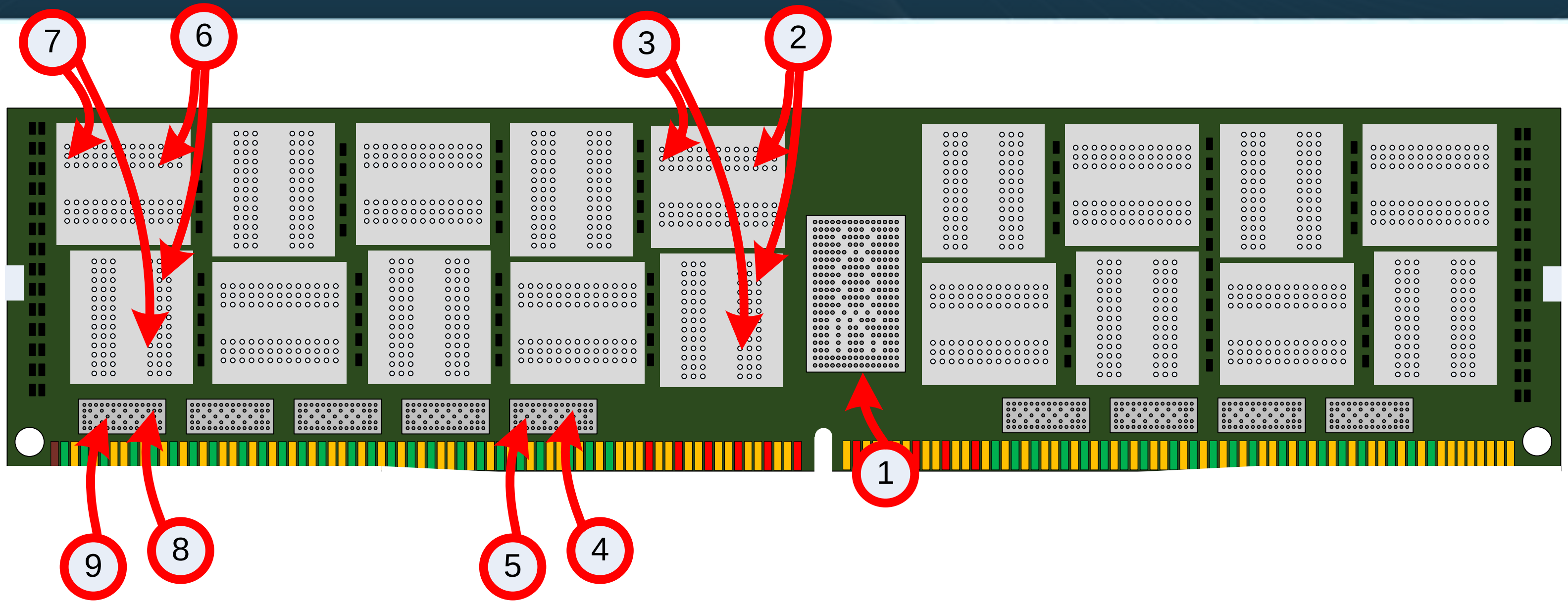
# DDR4 LRDIMM R/C E DRAM-to-DB DQ Routing



First DRAM top row to First DB – 32 mm
First DRAM bottom row to first DB - 27 mm
Last DRAM top row to First DB – 33 mm
Last DRAM bottom row to first DB - 11 mm

- For each DB
  - One nibble is on top row of a given slice
  - Second nibble is on bottom of the same slice
- Routing length differential between nibbles depends on slice (DB)

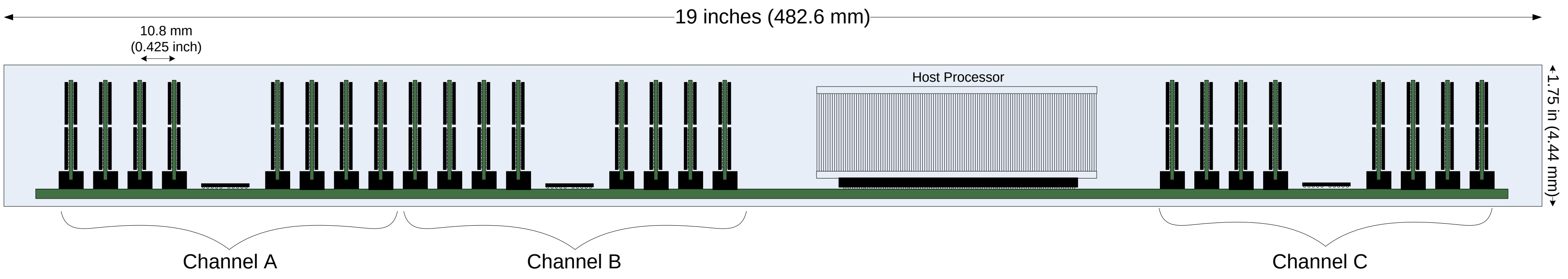
# Relative Timing Illustrations for DDR4 LRDIMM R/C E



1. Clock for read command arrives at RCD
2. tPDM + y_clock_to_DRAM_flight_time later, clock for read command arrives at DRAM devices
3. AL + PL + CL time after step #2, DRAM devices launch data, subject to tDQSCK variances
4. 27 mm (@ 7ps/mm) ~= 189 ps after step #3, first DB receives data from first DRAM on bottom row
5. 32 mm (@ 7ps/mm) ~= 224 ps after step #3, first DB receives data from first DRAM on top row
6. 100 mm(@ 10 ps/mm) ~= 1 ns after step #2, clock arrives at last DRAM devices
7. AL + PL + CL time after step #6, DRAM devices launch data, subject to tDQSCK variances
8. 11 mm (@ 7ps/mm) ~= 77 ps after step #7, first DB receives data from first DRAM on bottom row
9. 33 mm (@ 7ps/mm) ~= 231 ps after step #7, first DB receives data from first DRAM on top row

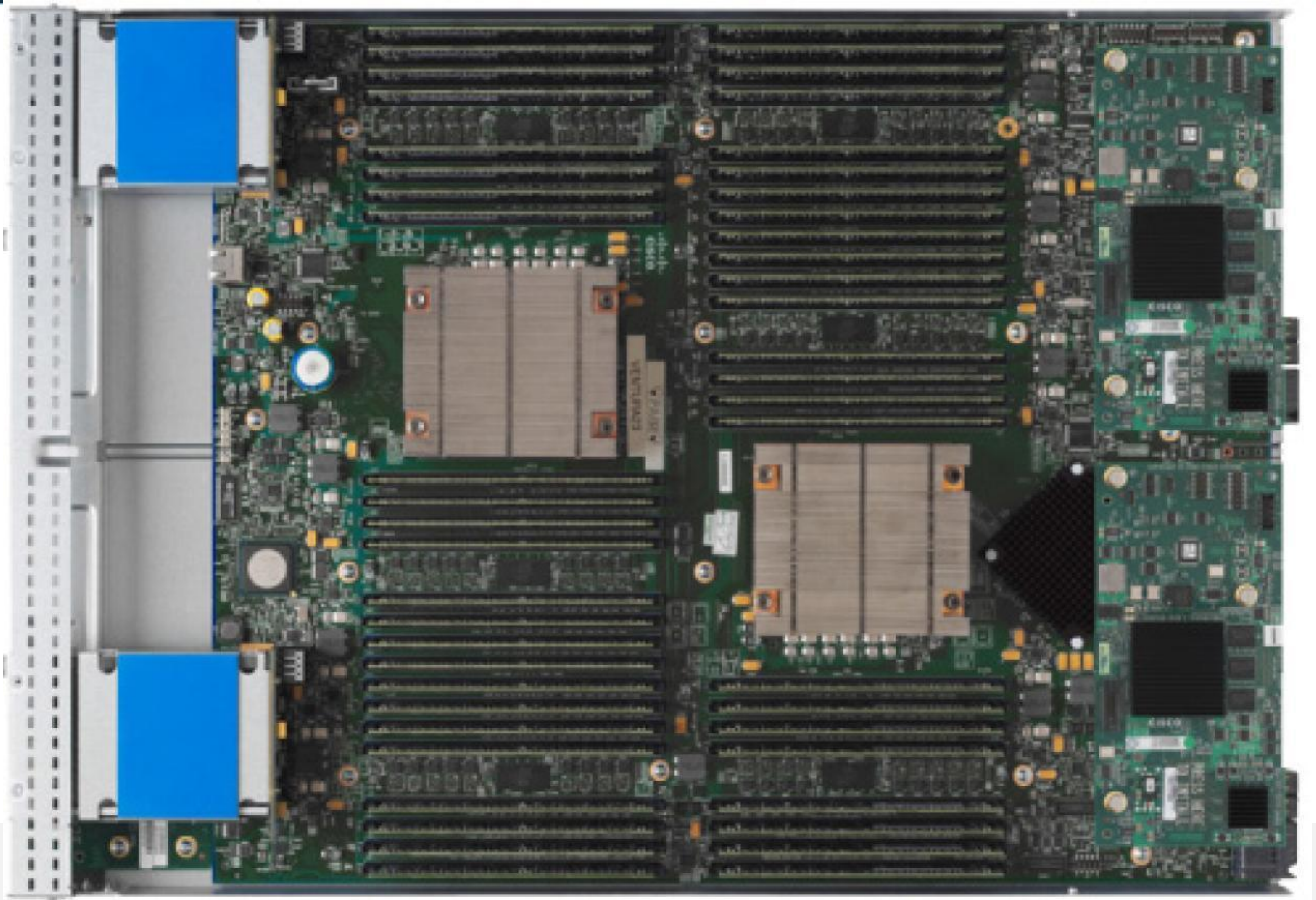
## Buffer on Board

# Cisco UCS 4:1 Switched BoB on NHM Platform

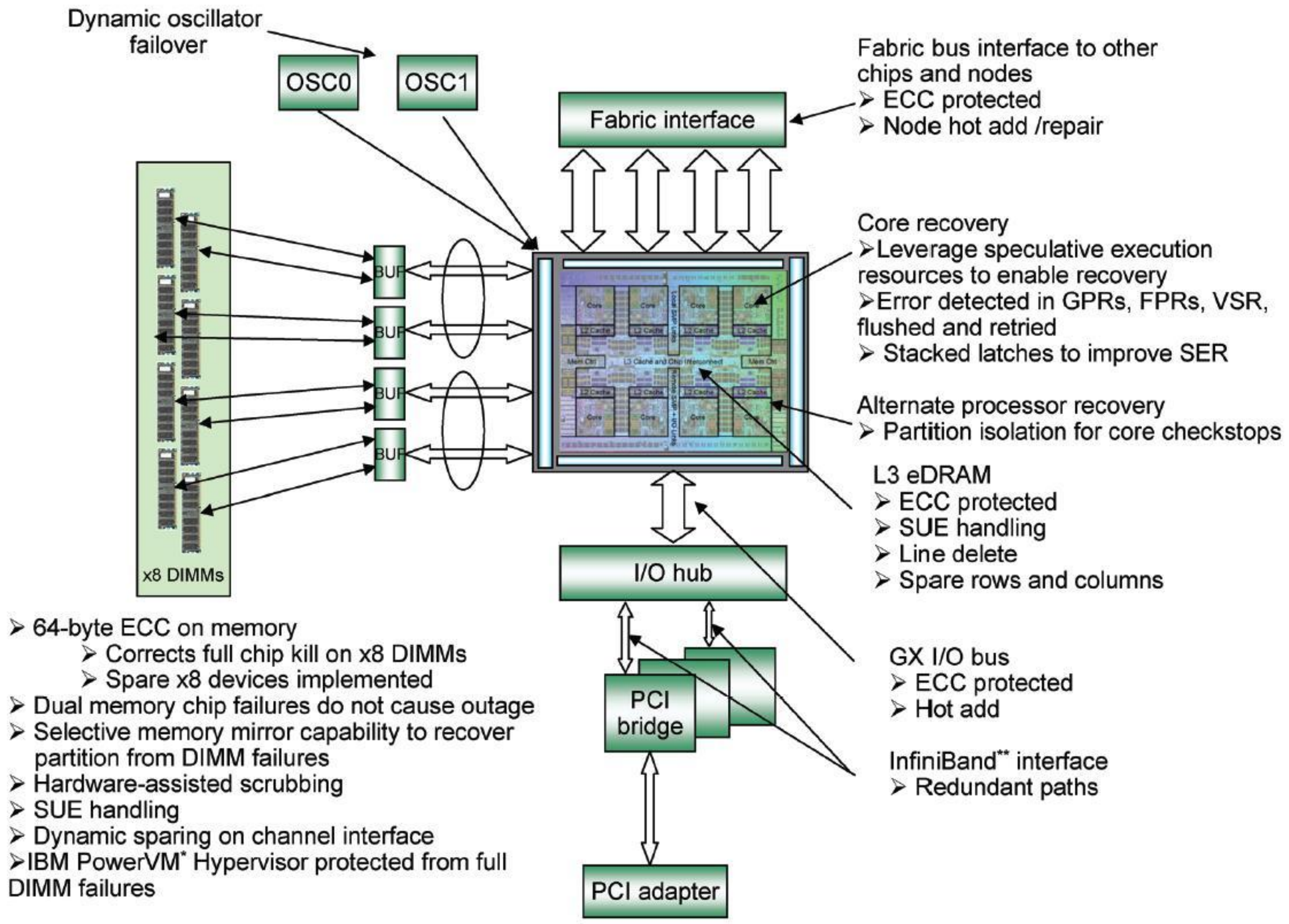


- Data buffers are 4:1 Switches
- Address and Data Buffers on same side as CPU/DIMMs
- Expands channel capacity from 2 DPC to 8 DPC
- Total spacing is 42 DIMM positions across
  - CPU occupies 12 DIMM positions
  - Address and Data buffers occupy 2 DIMM positions per channel (total of 6)
  - 6 + 12 + 24 = 42

## Cisco UCS (Nehalem-based System Memory Expansion)



# IBM Power 7 High Speed Serdes to DDRx



Fabric bus interface to other chips and nodes
- ECC protected
- Node hot add /repair

Core recovery
- Leverage speculative execution resources to enable recovery
- Error detected in GPRs, FPRs, VSR, flushed and retried
- Stacked latches to improve SER

Alternate processor recovery
- Partition isolation for core checkstops

L3 eDRAM
- ECC protected
- SUE handling
- Line delete
- Spare rows and columns

GX I/O bus
- ECC protected
- Hot add

InfiniBand** interface
- Redundant paths

- 64-byte ECC on memory
    - Corrects full chip kill on x8 DIMMs
    - Spare x8 devices implemented
- Dual memory chip failures do not cause outage
- Selective memory mirror capability to recover partition from DIMM failures
- Hardware-assisted scrubbing
- SUE handling
- Dynamic sparing on channel interface
- IBM PowerVM* Hypervisor protected from full DIMM failures

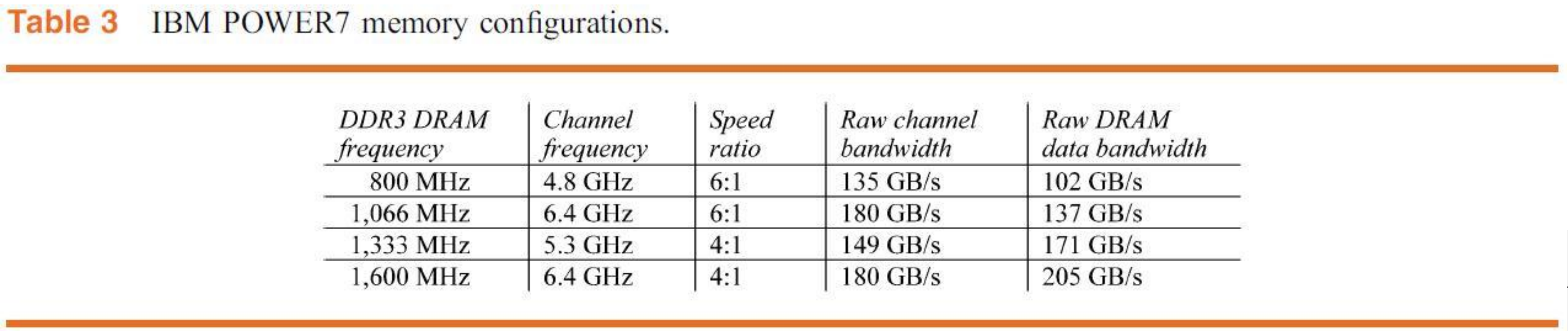
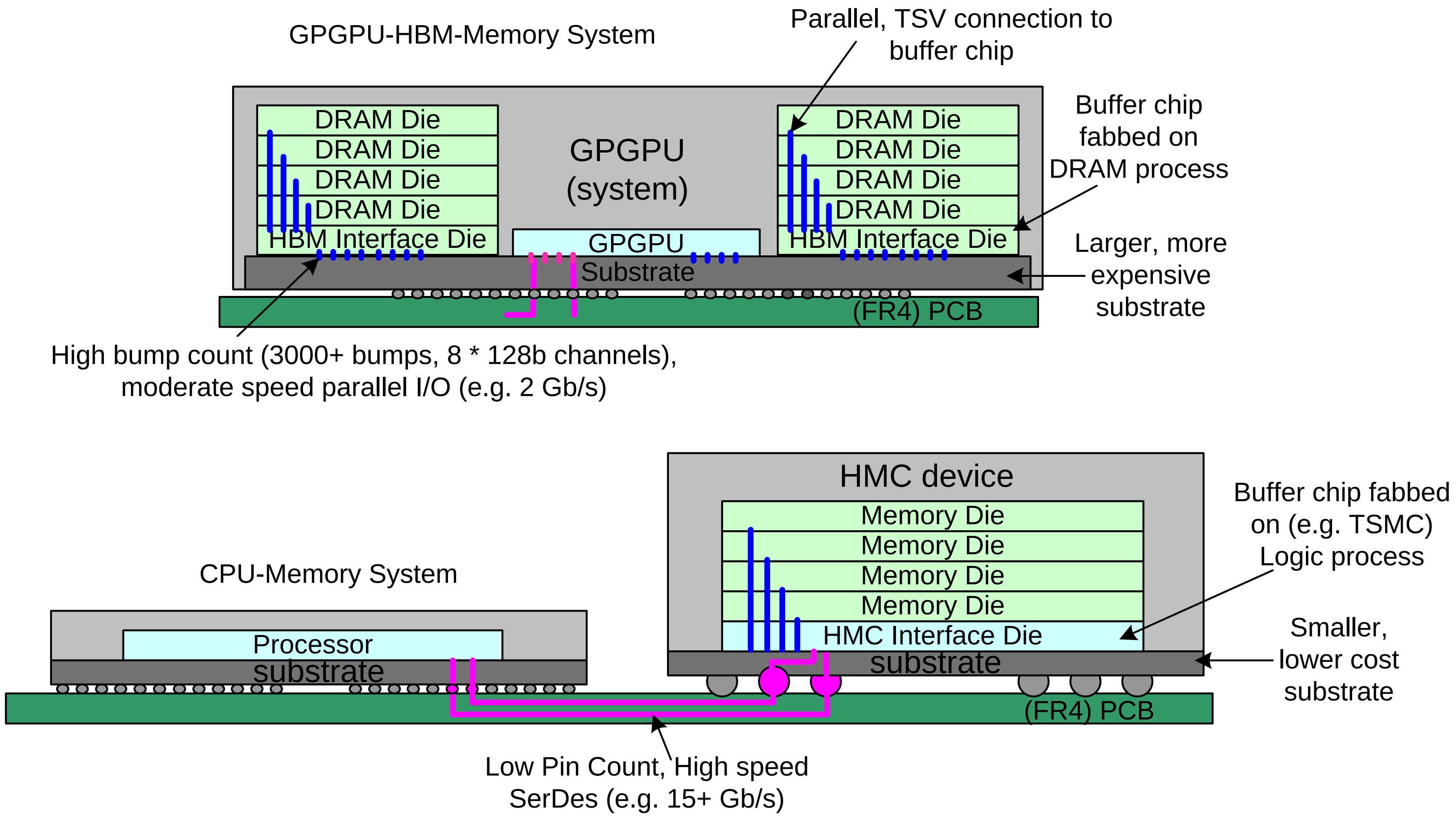


**Table 3** IBM POWER7 memory configurations.

| *DDR3 DRAM frequency* | *Channel frequency* | *Speed ratio* | *Raw channel bandwidth* | *Raw DRAM data bandwidth* |
|---|---|---|---|---|
| 800 MHz | 4.8 GHz | 6:1 | 135 GB/s | 102 GB/s |
| 1,066 MHz | 6.4 GHz | 6:1 | 180 GB/s | 137 GB/s |
| 1,333 MHz | 5.3 GHz | 4:1 | 149 GB/s | 171 GB/s |
| 1,600 MHz | 6.4 GHz | 4:1 | 180 GB/s | 205 GB/s |

## Buffering Within Memory Stack

# HMC vs. HBM


GPGPU-HBM-Memory System
Parallel, TSV connection to buffer chip
DRAM Die
DRAM Die
DRAM Die
DRAM Die
HBM Interface Die
GPGPU (system)
GPGPU
Substrate
(FR4) PCB
Buffer chip fabbed on DRAM process
Larger, more expensive substrate
High bump count (3000+ bumps, 8 * 128b channels), moderate speed parallel I/O (e.g. 2 Gb/s)
CPU-Memory System
Processor
substrate
HMC device
Memory Die
Memory Die
Memory Die
Memory Die
HMC Interface Die
substrate
(FR4) PCB
Buffer chip fabbed on (e.g. TSMC) Logic process
Smaller, lower cost substrate
Low Pin Count, High speed SerDes (e.g. 15+ Gb/s)

## Summary

## Summary

- Address and Data Buffers commonly used to improve pin-bandwidth/capacity of workstation and server platforms
  - Multitudes of high-speed-serdes-to-DDRx solutions have been implemented – AMB (FBDIMM), BoB
  - Multitudes of DDRx-to-DDRx also implemented
- Memory Buffers CAN do a lot more, but “doing more” typically means “higher cost”.
  - Standard buffers are typically cost-optimized solutions
- New Buffering concepts are being explored/implemented to do “interesting” things
  - DRAM + NAND backup as NVDIMM for power-failure protection
  - NAND-only Flash DIMM enables Flash devices to sit on DDR memory bus
  - Use of new memory technology (MRAM, PCM, ReRAM) on DDRx memory bus