Lecture 3: Memory Buffers and Scheduling

- Topics: buffers (FB-DIMM, RDIMM, LRDIMM, BoB, BOOM), memory blades, scheduling policies

HMC Details

- 32 banks per die x 8 dies = 256 banks per package
- 2 banks x 8 dies form 1 vertical slice (shared data bus)
- High internal data bandwidth (TSVs) ➔ entire cache line from a single array (2 banks) that is 256 bytes wide
- Future generations: eight links that can connect to the processor or other HMCs – each link (40 GBps) has 16 up and 16 down lanes (each lane has 2 differential wires)
- 1866 TSVs at 60 um pitch and 2 Gb/s (50 nm 1Gb DRAMs)
- 3.7 pJ/bit for DRAM layers and 6.78 pJ/bit for logic layer (existing DDR3 modules are 65 pJ/bit)

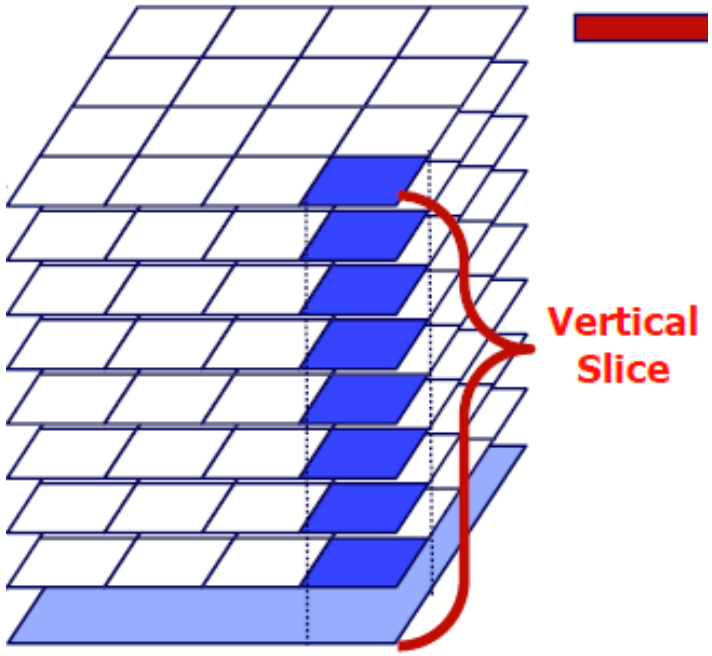
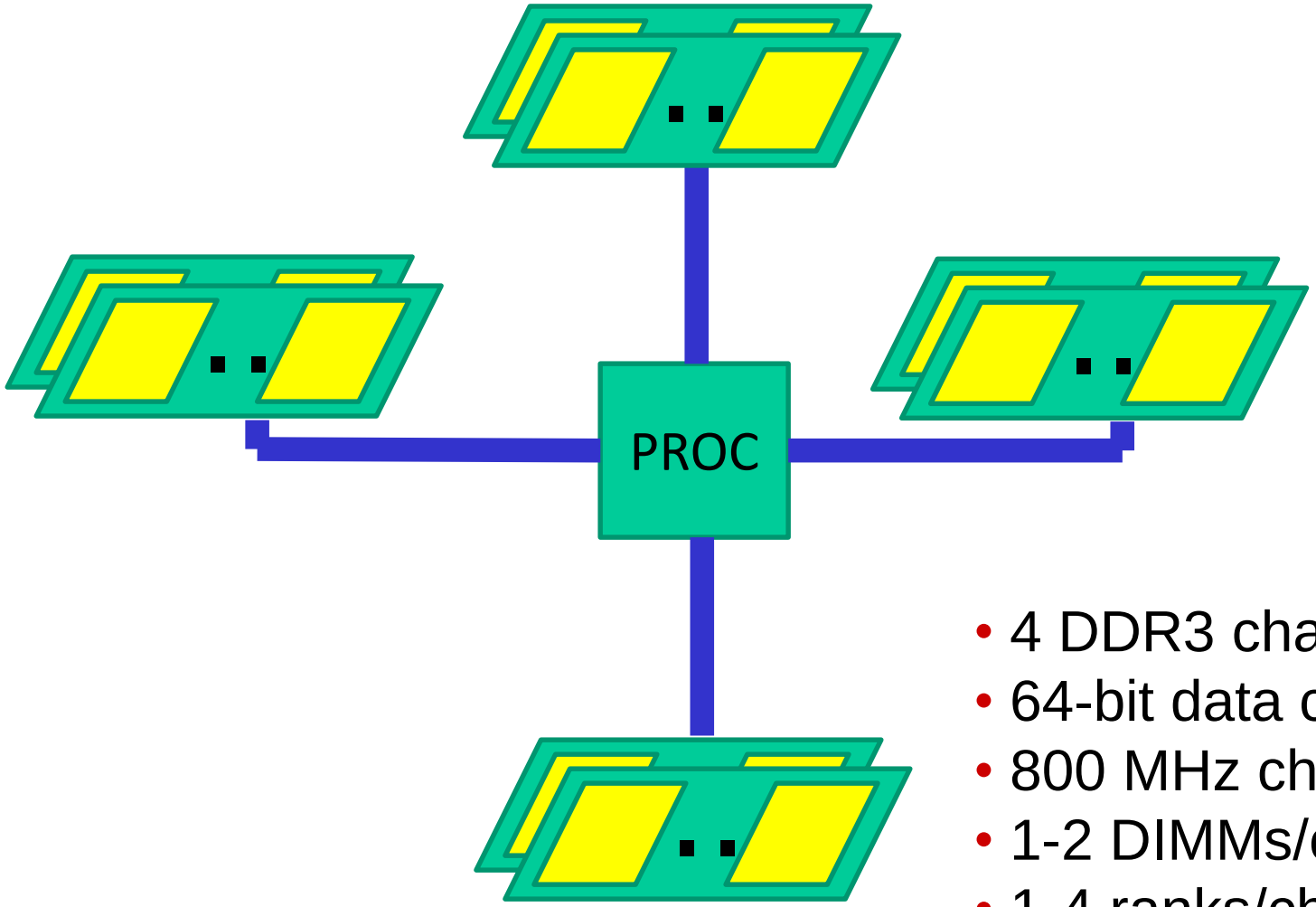


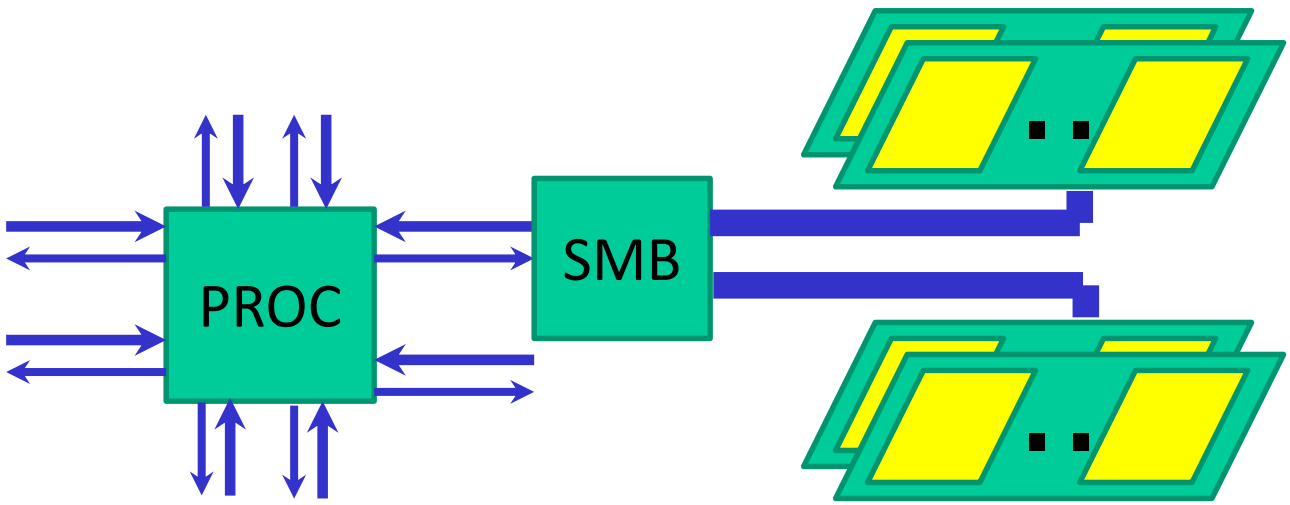
Figure from: T. Pawlowski, HotChips 2011

Modern Memory System



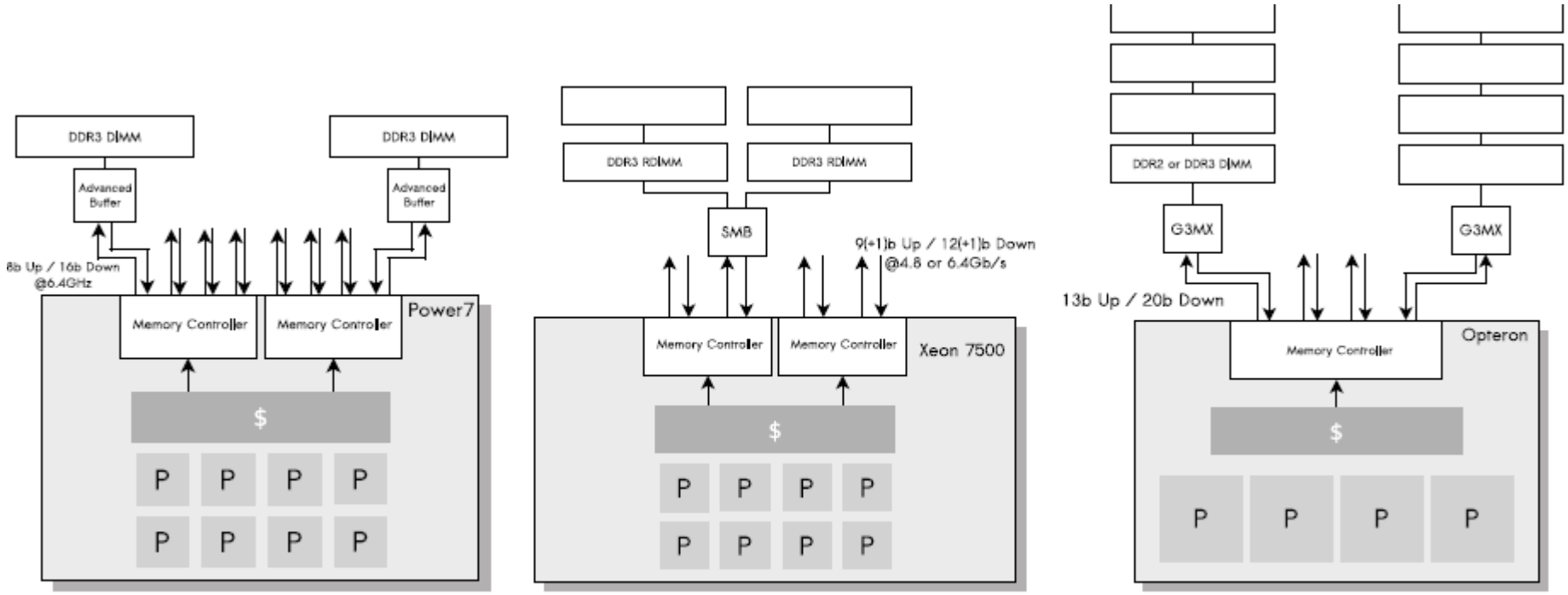
- 4 DDR3 channels
- 64-bit data channels
- 800 MHz channels
- 1-2 DIMMs/channel
- 1-4 ranks/channel

Cutting-Edge Systems



- The link into the processor is narrow and high frequency
- The Scalable Memory Buffer chip is a “router” that connects to multiple DDR3 channels (wide and slow)
- Boosts processor pin bandwidth and memory capacity
- More expensive, high power

Buffer-on-Board Examples



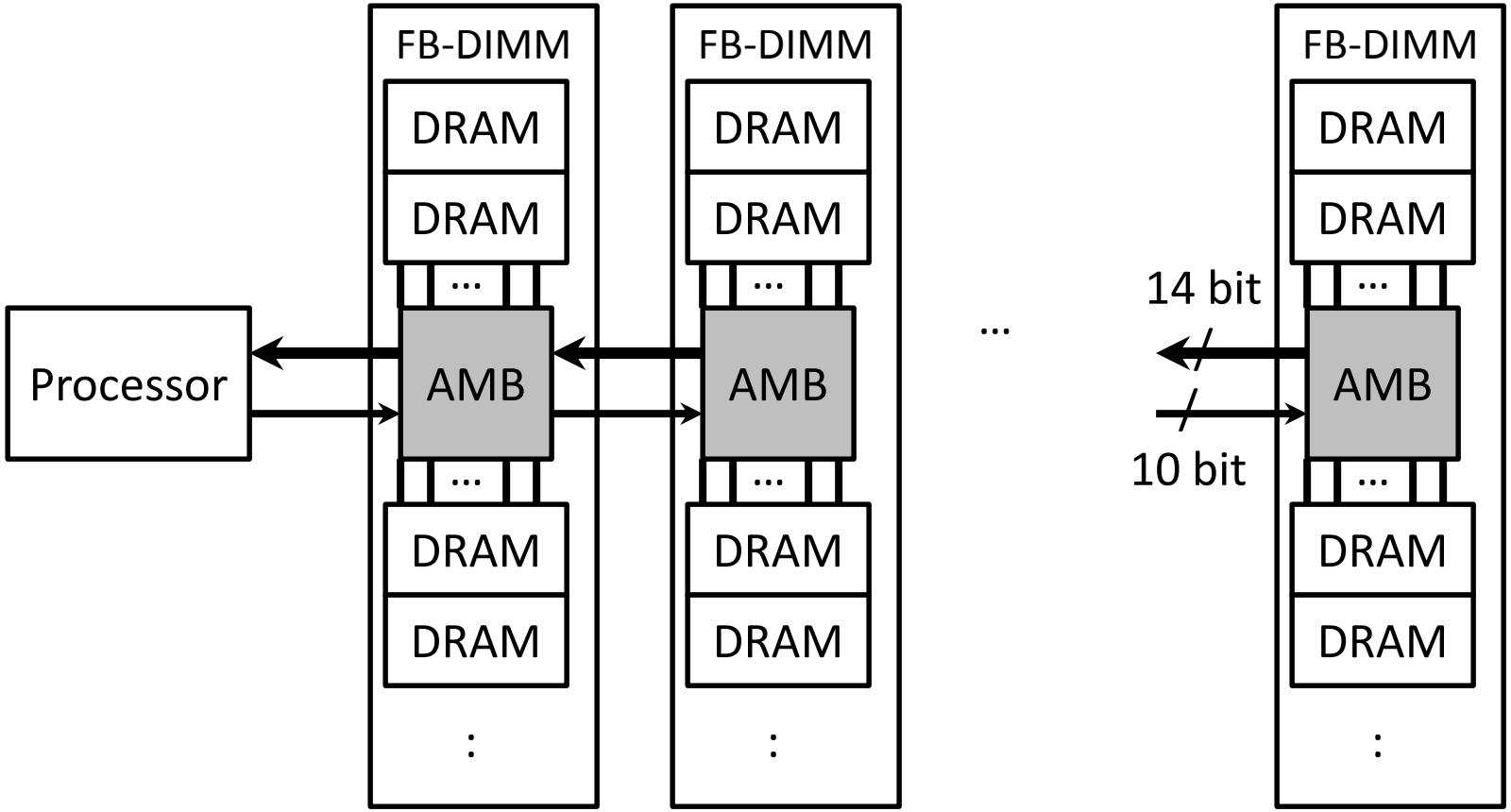
(a) IBM Power 795 memory system

(b) Intel SMI/SMB memory system

(c) AMD G3MX memory system

From: Cooper-Balis et al., ISCA 2012

FB-DIMM



FB-DIMM architecture; up to 8 FB-DIMMs can be daisy-chained through AMBs.

100 W for a fully-populated FB-DIMM channel under high load.

LR-DIMM

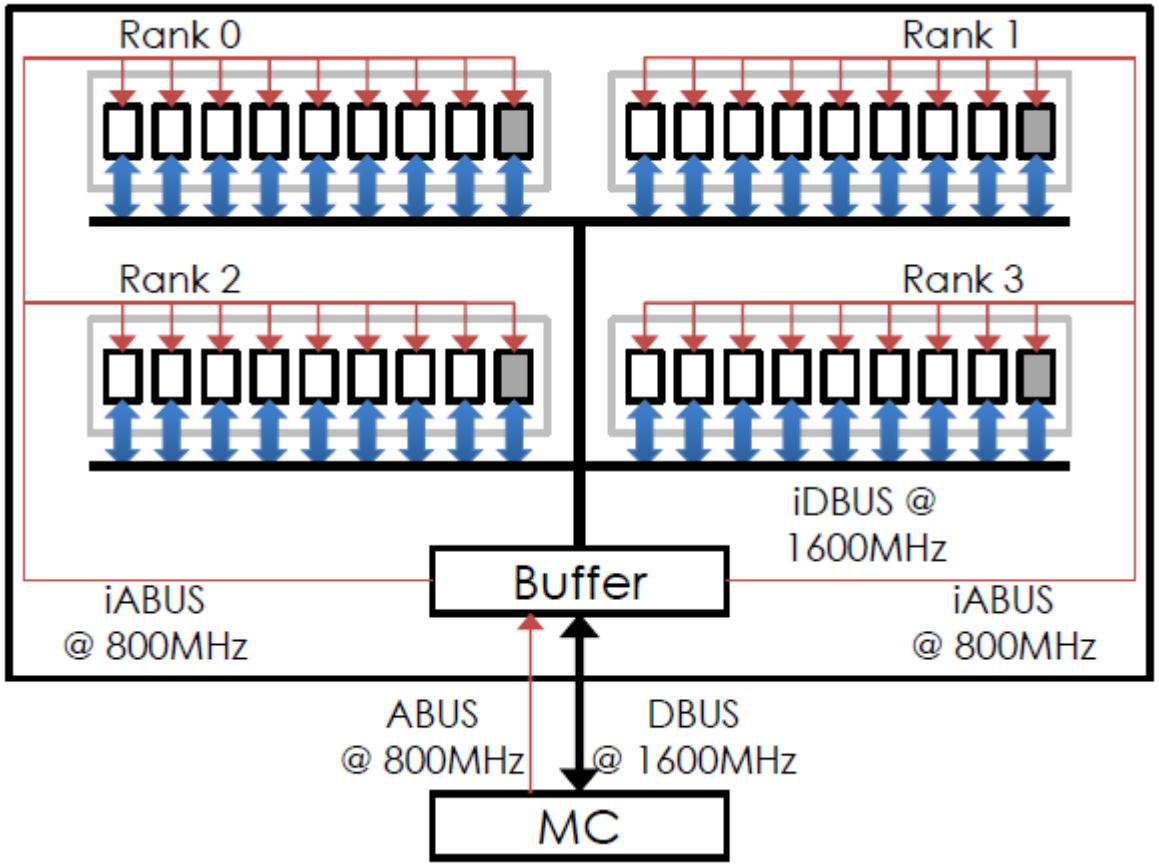
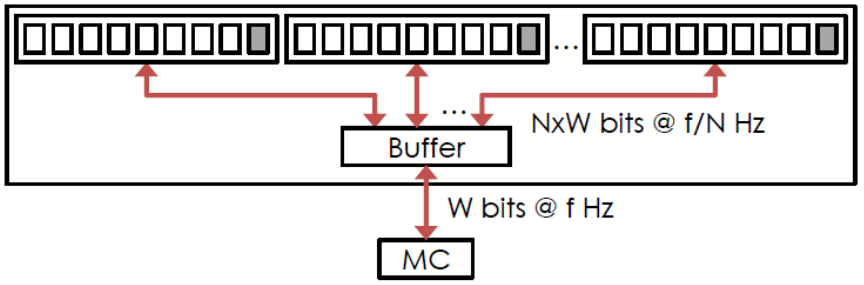
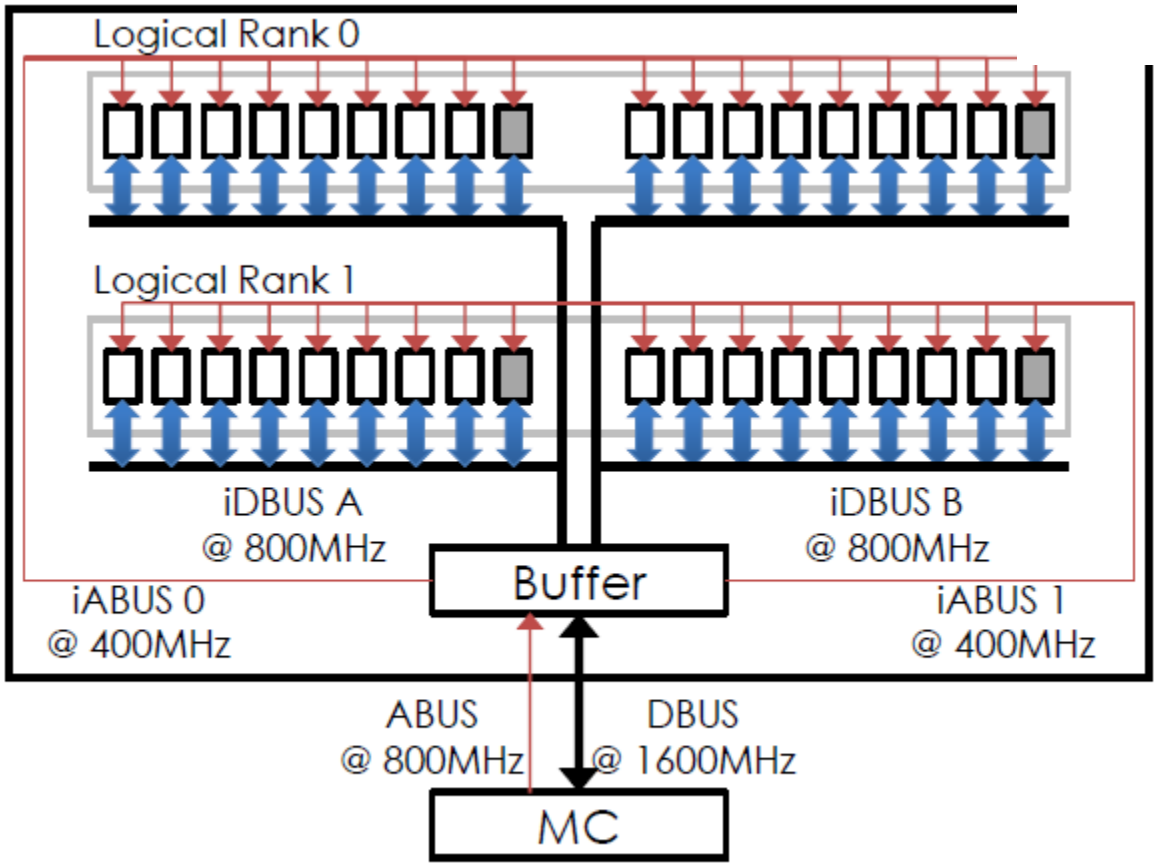


Figure from: Yoon et al., ISCA 2012

BOOM

Yoon et al., ISCA 2012

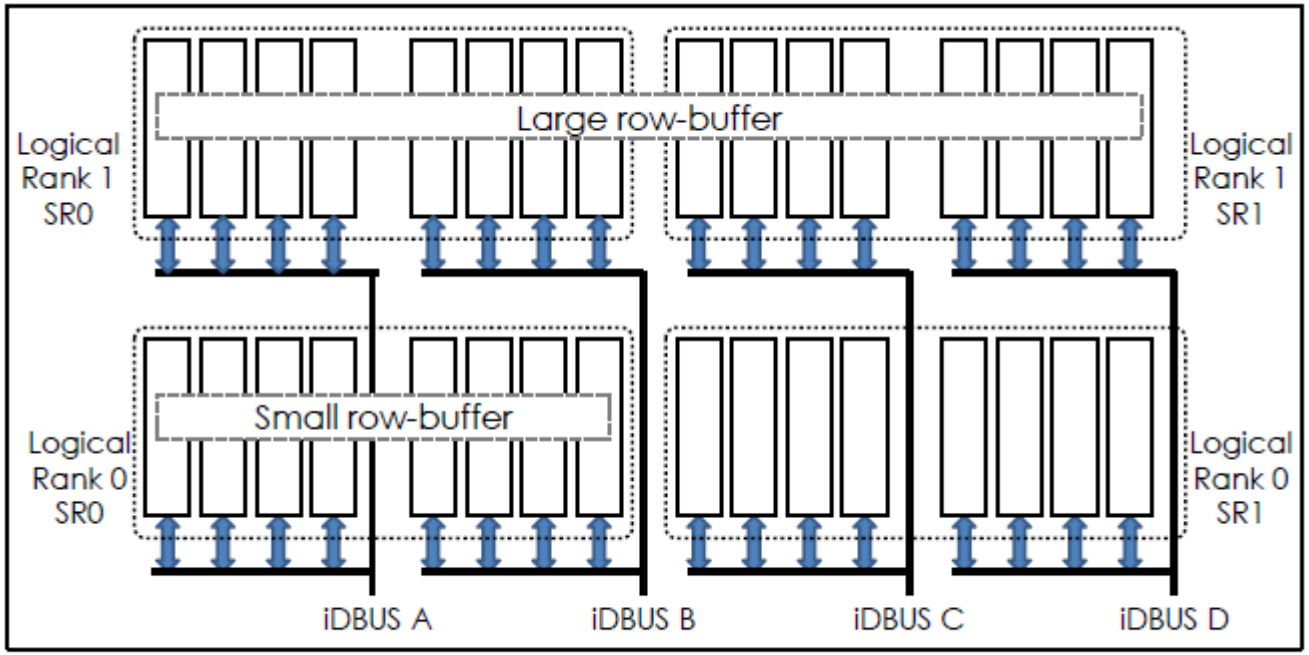




(b) BOOM with a 2× internal bus

- Using LPDDR chips at low frequency
- Wide internal datapath
- Longer burst length on DBUS
- No cache line buffering
- Higher activate power
- Same bandwidth, but lower power and higher latency

BOOM with Sub-Ranking



(a) Sub-ranked 4× BOOM using ×16 DRAM

Disaggregated Memory

Lim et al., ISCA 2009

- To support high capacity memory, build a separate memory blade that is shared by all compute blades in a rack
- For example, if average utilization is 2 GB, each compute blade is only provisioned with 2 GB of memory; but the compute blade can also access (say) 2 TB of data in the memory blade
- The hierarchy is exclusive and data is managed at page granularity
- Remote memory access is via PCIe (120 ns latency and 1 GB/s in each direction)

Scheduling Policies Basics

- Must honor several timing constraints for each bank/rank
- Commands: PRE, ACT, COL-RD, COL-WR, REF, Power-Up/Dn
- Must handle reads and writes on the same DDR3 bus
- Must issue refreshes on time
- Must maximize row buffer hit rates and parallelism
- Must maximize throughput and fairness

Address Mapping Policies

- Consecutive cache lines can be placed in the same row to boost row buffer hit rates
- Consecutive cache lines can be placed in different ranks to boost parallelism
- Example address mapping policies:
  row:rank:bank:channel:column:blkoffset

  row:column:rank:bank:channel:blkoffset

Reads and Writes

- A single bus is used for reads and writes
- The bus direction must be reversed when switching between reads and writes; this takes time and leads to bus idling
- Hence, writes are performed in bursts; a write buffer stores pending writes until a high water mark is reached
- Writes are drained until a low water mark is reached

Refresh

- Example, tREFI (gap between refresh commands) = 7.8us, tRFC (time to complete refresh command) = 350 ns
- JEDEC: must issue 8 refresh commands in a 8*tREFI time window
- Allows for some flexibility in when each refresh command is issued
- Elastic refresh: issue a refresh command when there is lull in activity; any unissued refreshes are handled at the end of the 8*tREFI window

Maximizing Row Buffer Hit Rates

- FCFS: Issue the first read or write in the queue that is ready for issue (not necessarily the oldest in program order)
- First Ready - FCFS: First issue row buffer hits if you can

STFM

Mutlu and Moscibroda, MICRO'07

- When multiple threads run together, threads with row buffer hits are prioritized by FR-FCFS
- Each thread has a slowdown: $S = T_{alone} / T_{shared}$, where T is the number of cycles the ROB is stalled waiting for memory
- Unfairness is estimated as S_{max} / S_{min}
- If unfairness is higher than a threshold, thread priorities override other priorities (Stall Time Fair Memory scheduling)
- Estimation of T_{alone} requires some book-keeping: does an access delay critical requests from other threads?

PAR-BS

Mutlu and Moscibroda, ISCA'08

- A batch of requests (per bank) is formed: each thread can only contribute R requests to this batch; batch requests have priority over non-batch requests
- Within a batch, priority is first given to row buffer hits, then to threads with a higher "rank", then to older requests
- Rank is computed based on the thread's memory intensity; low-intensity threads are given higher priority; this policy improves batch completion time and overall throughput
- By using rank, requests from a thread are serviced in parallel; hence, parallelism-aware batch scheduling

TCM

Kim et al., MICRO 2010

- Organize threads into latency-sensitive and bw-sensitive clusters based on memory intensity; former gets higher priority
- Within bw-sensitive cluster, priority is based on rank
- Rank is determined based on “niceness” of a thread and the rank is periodically shuffled with insertion shuffling or random shuffling (the former is used if there is a big gap in niceness)
- Threads with low row buffer hit rates and high bank level parallelism are considered “nice” to others

Minimalist Open-Page

Kaseridis et al., MICRO 2011

- Place 4 consecutive cache lines in one bank, then the next 4 in a different bank and so on – provides the best balance between row buffer locality and bank-level parallelism
- Don’t have to worry as much about fairness
- Scheduling first takes priority into account, where priority is determined by wait-time, prefetch distance, and MLP in thread
- A row is precharged after 50 ns, or immediately following a prefetch-dictated large burst

Other Scheduling Ideas

- Using reinforcement learning Ipek et al., ISCA 2008
- Co-ordinating across multiple MCs Kim et al., HPCA 2010
- Co-ordinating requests from GPU and CPU
  Ausavarungnirun et al., ISCA 2012
- Several schedulers in the Memory Scheduling Championship at ISCA 2012
- Predicting the number of row buffer hits Awasthi et al., PACT 2011

Title

- Bullet