IBM System zEnterprise 196: Memory Subsystem Overview

A special thanks to Pat Meaney for his assistance and for his graphics, many of which appear herein.
Why System z Today?

**The most secure, the most reliable commercial machine on Earth!**

- System z is the most available platform you can buy (99.999%)
- The security on System z is unsurpassed by any other platform
- Stringent Service Level Agreements

**The backbone of data processing in large enterprise environments**

A large portion of today's operational data resides on the mainframe – *and it's growing*....

**Workload management for mixed workloads**

Manage multiple workload types with priority and work types scheduling

**Lower costs through reduced complexity**

Simplified management

Reduced environmental costs

Greater flexibility to meet changing needs
But what if something goes wrong?

- IBM System z environments host mission-critical applications: going down is not an option!
- A typical big-box retailer can lose $100,000s of business for every minute of downtime
- A typical credit card transaction processor can lose $1,000,000s of business for every minute of downtime
- Typical distributed systems measure intervals between service interruptions in days
- Typical mainframe environments measure intervals between service interruptions in years – and that's still not good enough
- On the mainframe, reliability isn't just a feature: it's a design philosophy
... the answer is RAS: reliability, availability, and serviceability

- Every conceivable error scenario and fault must be considered: how to detect, how to heal, how to recover?
- Massive design work is undertaken to handle every niggling little potential fault; downtime is not an option!

Impact of Outage

- Unscheduled Outages
- Scheduled Outages
- Planned Outages
- Preplanning requirements

Power & Thermal Management

Temperature = Silicon Reliability Worst Enemy
Wearout = Mechanical Components Reliability Worst Enemy.
Tremendous capacity possible with tightly-coupled design:
- Six CP chips per node; six cores per CP chip
- Three memory controllers per node
- Memory controller in z196 an all-new design
- Five memory channels per MCU
- Two cascades of memory cards supported
- Differential I/O between memory cards and MCU allows for tremendous bandwidth

- 96 total cores
- Total system cache
  - 768 MB shared L4 (eDRAM)
  - 576 MB L3 (eDRAM)
  - 144 MB L2 private (SRAM)
  - 19.5 MB L1 private (SRAM)
Memory controller processes data fetch/store requests and key ops from local L3 controllers.

To maximize bandwidth, two independent L3 ports are used, sliced on address.

Each memory controller communicates with five memory 'channels,' each of which contains the DRAMs and SuperNova AMB controller; two cascades of memory cards can exist in each channel.

Key cache capacity and performance improvements have been made from previous IBM System z mainframes.

Vast improvements in recovery from errors over previous systems thanks to our new RAIM system.

And we're running pretty brisk, too!

**Speeds and Feeds:**

- 1.4GHz in MCS, 1.6GHz in MCA
- 42GB/sec between MCU and L3
- 24GB/sec store throughput to memory
- 38GB/sec fetch throughput from memory
# z196 Memory Cards – Detailed View

<table>
<thead>
<tr>
<th>DIMM Size</th>
<th>DIMM Tech</th>
<th>DRAM Width</th>
<th>DRAM Size</th>
<th># of Chips</th>
<th># of Ranks</th>
<th># of banks</th>
<th>Data Rate (Mhz)</th>
<th>Max/MCU</th>
<th>Max/Node</th>
<th>Max/System</th>
</tr>
</thead>
<tbody>
<tr>
<td>4GB</td>
<td>DDR3-800</td>
<td>x8</td>
<td>1Gb</td>
<td>36</td>
<td>4</td>
<td>8</td>
<td>800</td>
<td>32GB</td>
<td>96GB</td>
<td>384GB</td>
</tr>
<tr>
<td>16GB</td>
<td>DDR3-800</td>
<td>x8</td>
<td>2Gb</td>
<td>80 planar</td>
<td>8</td>
<td>8</td>
<td>800</td>
<td>128GB</td>
<td>384GB</td>
<td>1.5TB</td>
</tr>
<tr>
<td>32GB</td>
<td>DDR3-800</td>
<td>x8</td>
<td>2Gb</td>
<td>80 dual die</td>
<td>16</td>
<td>8</td>
<td>800</td>
<td>256GB</td>
<td>768GB</td>
<td>3TB</td>
</tr>
</tbody>
</table>

**Front**

SuperNOVA

96 mm
z196 Memory Controller – Physical View

- 1 Independent MC per CP
- Unused MCs gated to save power
- 5 Channels per MC
  - 5 Lock Step Channels (RAIM)
  - First four channels consist of data + ECC; fifth channel ECC only
- Up to 2 cascades
- Channels – 4.8Gb/s
- Differential Clock and Data
- Packet based CRC
  - Down Stream
    - 1 Clock
    - 13 Data / Command
    - 2 Spares (1 Spare for Clk)
    - 6:1 = 12 beats, CRC18
  - Up Stream
    - 1 Clock
    - 20 Data / Command
    - 2 Spares (1 Spare for Clk)
    - 6:1 = 8 beats, CRC16
Layers of Memory Recovery

ECC
- Powerful 90B/64B Reed Solomon code

DRAM Failure
- Marking technology; no half sparing needed
- 2 DRAM can be marked
- Call for replacement on third DRAM

Lane Failure
- CRC with Retry
- Data – lane sparing
- CLK – RAIM with lane sparing

DIMM Failure (discrete components, VTT Reg.)
- CRC with Retry
- Data – lane sparing
- CLK – RAIM with lane sparing

DIMM Controller ASIC Failure
- RAIM Recovery

Channel Failure
- RAIM Recovery
Preventing Unscheduled Outages

- Advanced Memory RAIM (Redundant Array of Independent Memory) design
- Enhanced Reed-Solomon code (ECC) – 90B/64B
- Protection against Channel/DIMM failures
- Chip marking for fast DRAM replacements
- Mirrored Key cache
- Improved chip packaging
- Continued focus on Firmware

Handling Preventable Outages

- Double memory data bus lane sparing (reducing repair actions)
- Single memory clock bus sparing
- Power distribution using N+2 Voltage Transformation Modules (VTM)
- Redundant (N+2) Humidity Sensors
- Redundant (N+2) Altitude Sensors
z196 MCU Recovery: ECC/RAIM Protection

<table>
<thead>
<tr>
<th>Marks/New Errors</th>
<th>No Marks</th>
<th>Single Chip Marked</th>
<th>Two Chips Marked</th>
<th>DIMM Marked or 3+ Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>GOOD</td>
<td>GOOD</td>
<td>GOOD</td>
<td>GOOD</td>
</tr>
<tr>
<td>One Chip</td>
<td>CE</td>
<td>CE</td>
<td>CE</td>
<td>Service Request</td>
</tr>
<tr>
<td>Two Chips, Same Channel</td>
<td>CE</td>
<td>CE</td>
<td>CE</td>
<td>Call home for part replacement</td>
</tr>
<tr>
<td>Full Channel Error (CRC or other)</td>
<td>CE</td>
<td>CE</td>
<td>CE</td>
<td>Call home for part replacement</td>
</tr>
</tbody>
</table>

- No need for spare chips! If we 'know' in advance of a location where errors are extant, we 'mark' it. This can be applied on a DRAM or a channel basis. Marking ahead of time allows us to find new errors on top of the known ones.
- Marks are much more flexible than spare DRAMs, and can keep the card cost down as well, to say nothing of the reduced design cycle given simpler verification.

- If we ever reach a threshold where the system can no longer correct dynamically, we 'call home' to let IBM know to replace the defective part; the system continues to operate without performance penalty in the interim.
- Firmware is always looking at the health of DRAMs in the background.
z196 MCU Recovery: ECC/RAIM Protection (cont.)

Level 3 Cache

Key Cache

MCU 0

Key Cache

MCU 1

Key Cache

MCU 2

DATA

CHECK

ECC

RAIM Parity

Extra column provides RAIM function
Store pipe in MCU calculates ECC across entirety of 64B ECC group
Data is then packaged up on a per-channel basis
RAIM is generated by using a simple set of XOR4s
Any pre-existing error on the data is stamped with an SPUE (special UE)
SPUEs allow us, on subsequent fetches, to know that the error did not originate in memory and/or that the error is not new
They also enable us to avoid going into recovery in the core, and demanding a refetch of the data
The SPUE codepoint is such that, if new errors occur on top, we can detect that properly
Chip marks in are applied according to variable criteria set by designers and/or field feedback as a system ages.

Channels with CRC errors are also 'marked' internally, using the RAIM mechanism to apply full channel correction.

Marking literally recreates the data *ex nihilo* within the decoder; it doesn't use the bad data during this process.

- **RAIM ECC Decoder Logic**
- **ECC Status**
- **Corrected Data 64 Bytes**
- **CRC Check**
- **Chan0 Data**
- **Chan1 Data**
- **Chan2 Data**
- **Chan3 Data**
- **Chan4 Data**

**Marks:**
- up to 2 Chip
- up to 1 DIMM (orig or CRC)
z196 MCU Recovery: Bus CRC Errors

Downstream Data/Commands:
- CRC18 is generated on 12-beat packets
- 156/138 packet size
- CRC generated across all three blocks and sent in last portion of frame
- Errors detected by memory card, forwards poison CRC code back to MCU to initiate CRC recovery

Upstream Data:
- CRC16 is generated on 8-beat packets
- 160/144 packet size
- 18 bytes x 5 channels = 90 Bytes
- Must wait for all data before seeing CRC error
- If error seen, if only in one channel, RAIM can be used for full channel correct
- Otherwise CRC recovery initiated
z196 MCU Recovery: Bus Error Recovery (Tier 1)

1. Normal (powdown)
   - MC Sends DS Poison CRC
   - Wait 550 cycles (Establish STR)

2. Error Detection
   - MC Sends Error Ack
   - Tier 1 Retry

3. MC Retry Resend Stores
   - MC Waits for prev ops to complete

4. Poison CRC up to MC
   - MC Sends DS Poison CRC
   - MC Sends Error Error Ack

5. Tier 1 Retry

- Stores Pending
- New Stores
- New Fetches
- Fetches Pending
- IFC
z196 MCU Recovery: Bus Error Recovery (Tier 2)

- Escalation to higher tiers of recovery depends on rate of errors seen.
- The MCU watches and counts the number of errors in a given time window and also if previous attempts eliminated the errors or not.
- If still firing, we escalate to tier 2, where we effectively re-init the channel taking the error.
- Typically this will also involve a clock failover.

Diagram:
- Normal Operation
- Error Detection
- MC Retry
  - Resend Stores
  - Poison CRC
  - up to MC
- MC Halts new ops
- MC Waits for prev ops to complete
- MC Sends DS Poison CRC
- MC Sends Error Ack
- Tier 1 Retry
  - Wait 550 cycles (Establish STR)
- Tier 2 Fast Init
  - TS2-TS7
  - N Chan Trouble?
  - Y Chan Trouble?
  - Degrade bad channel
z196 MCU Recovery: Bus Error Recovery (Tier 3)

- **Tier 1**: Retry
  - **Tier 2 Link**
    - **FastInit**
    - **MC Sends Error Ack**
  - **MC Waits for prev ops to complete**
  - **MC Sends DS Poison CRC**
  - **MC Halts new ops**
  - **Poison CRC up to MC**

- **Tier 3 GOAL**: Use TS0 to clean up clocks and reset SN hardware
  - **RAIM-Degraded SlowInit**
    - **TS2-TS7 5 channels**
      - **Wait (hang check)**
      - **Y Chan Trouble?**
    - **Recently in Tier 2**
      - **TS2-TS7**
      - **Wait 550 cycles (Establish STR)**
  - **TS0 bad channel**
  - **N Chan Trouble?**
    - **STR-> Powerdown**
      - **MC Retry Resend Stores**
      - **Normal Operation**
      - **Error Detection**
    - **MC Scrub pass complete**
  - **RAIM Degraded**

- **Subsequent Channel error**
  - **Call Home**
    - **Channel Degrade**
      - **Start MCU Scrub Pass**
        - **Successful**
      - **Fail**
        - **Subsequent Channel error**
          - **TS2-TS7 5 channels**
            - **Wait (hang check)**
            - **N Chan Trouble?**
              - **STR-> Powerdown**
                - **MC Retry Resend Stores**
                - **Normal Operation**
                - **Error Detection**
              - **MC Scrub pass complete**
            - **RAIM Degraded**
z196 MCU Recovery: Bus Error Recovery Summary

**Store Errors:**
- Before retiring the queue, wait a fixed number of cycles for receipt of poison CRC code; if none received, operation successful
- If poison CRC received in given window, begin CRC recovery
- Retry the store that received the error and any fetches that would have been in flight during CRC recovery

**Fetch Errors:**
- If bad CRC calculated on fetch data frame, and error is limited to one and only one channel, and no other channel mark extant, we can internally act as though that channel is 'marked,' using RAIM to reconstruct the data for the channel with bad CRC
- For retry mechanism we rely on the core: if it receives bad data (UE), it will refetch the line itself; this saves hardware in the MCU, but adds latency for retrying fetches that return with uncorrectable errors
- Keys are 'mirrored' across a 64B basis. For every 128B key fetch, you get two copies of the same 64B set of keys; if a key fetch has a UE, we look at the 'mirrored' copy; otherwise it's treated like a normal demand fetch from the recovery standpoint
# z196 MCU RAS Summary

<table>
<thead>
<tr>
<th>RAS feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAIM ECC</td>
<td>Five-channel ECC that can detect and correct new DRAM failures as well as most varieties of single-channel failures in the memory subsystem.</td>
</tr>
<tr>
<td>DRAM chip marking</td>
<td>Up to two DRAM chip marks can be applied per rank in order to ignore errors from known defective DRAM chips. Unlike DRAM chip sparing, these marks can be applied without having to replicate any chip data.</td>
</tr>
<tr>
<td>Channel marking</td>
<td>Channel marking is the ability to designate one of five RAIM channels as defective. The channel mark provides 100% correction of the data in the ignored channel. There are four levels of channel marking: dynamic, Tier 3, temporary, and permanent.</td>
</tr>
<tr>
<td>CRC bus detection</td>
<td>Upstream and downstream channels are checked using CRC.</td>
</tr>
<tr>
<td>Tier 1 reset</td>
<td>Tier 1 recovery quiesces the channels, resets memory channel resources, and then resends stores that may have been dropped.</td>
</tr>
<tr>
<td>Tier 2 data calibration Lane sparing</td>
<td>Tier 2 recovery recalibrates memory data buses and spares out bad data lanes.</td>
</tr>
<tr>
<td>Tier 3 clock calibration Lane sparing</td>
<td>Tier 3 recovery recalibrates memory clocks and spares out bad clock lanes. Firmware performs fast scrub to clean up stale data.</td>
</tr>
<tr>
<td>Scrubbing</td>
<td>Scrubbing is the process of periodically reading, correcting, and writing back memory to correct soft errors. Scrubbing provides chip error counts that are used to apply DRAM chip and channel marks.</td>
</tr>
<tr>
<td>Service request</td>
<td>A service request is an event that requests a part replacement. Some examples of memory-related service requests include the following:</td>
</tr>
<tr>
<td></td>
<td>- Permanent, full-channel RAIM degrade.</td>
</tr>
<tr>
<td></td>
<td>- Overflow of the DRAM mark capabilities within a rank.</td>
</tr>
<tr>
<td></td>
<td>- Overflow of bus spare lanes within a channel or cascade.</td>
</tr>
</tbody>
</table>
How Does Something Like This Get Designed?

- Product Engineering
- Research Group
- RAS Council
- Logic Design and uArch
- Firmware
- Verification
- HW Bringup and Test
Backup Pictures
z196 Air cooled – Under the covers (Model M66 or M80) Front view

- Internal Batteries (optional)
- Power Supplies
- 2 x Support Elements
- I/O cage
- PCIe I/O drawers
- Processor Books, Memory, MBA and HCA cards
- Ethernet cables for internal System LAN connecting Flexible Service Processor (FSP) cage controller cards
- InfiniBand I/O Interconnects
- 2 x Cooling Units (MRUs)
- Optional FICON & ESCON FQC (not shown)
IBM System zEnterprise 196 Memory Subsystem Overview

z196 Water cooled – Under the covers (M66 or M80) front view

- Internal Batteries (optional)
- Power Supplies
- Support Elements
- I/O cage
- I/O drawers
- Ethernet cables for internal System LAN connecting Flexible Service Processor (FSP) cage controller cards
- Processor Books, Memory, MBA and HCA cards
- InfiniBand I/O Interconnects
- 2 x Water Cooling Units
z196 Multi-Chip Module (MCM) Packaging

- 96mm x 96mm MCM
  - 103 Glass Ceramic layers
  - 8 chip sites
  - 7356 LGA connections
  - 20 and 24 way MCMs
  - Maximum power used by MCM is 1800W

- CMOS 12s chip Technology
  - PU, SC, S chips, 45 nm
  - 6 PU chips/MCM – Each up to 4 cores
    - One memory control (MC) per PU chip
    - 23.498 mm x 21.797 mm
    - 1.4 billion transistors/PU chip
    - L1 cache/PU core
      - 64 KB I-cache
      - 128 KB D-cache
    - L2 cache/PU core
      - 1.5 MB
    - L3 cache shared by 4 PUs per chip
      - 24 MB
    - 5.2 GHz
  - 2 Storage Control (SC) chip
    - 24.427 mm x 19.604 mm
    - 1.5 billion transistors/SC chip
    - L4 Cache 96 MB per SC chip (192 MB/Book)
    - L4 access to/from other MCMs
  - 4 SEEPROM (S) chips
    - 2 x active and 2 x redundant
    - Product data for MCM, chips and other engineering information
  - Clock Functions – distributed across PU and SC chips
    - Master Time-of-Day (TOD) function is on the SC
z196 Quad Core PU Chip Detail

- **Up to Four active cores per chip**
  - 5.2 GHz
  - L1 cache/ core
    - 64 KB I-cache
    - 128 KB D-cache
  - 1.5 MB private L2 cache/ core

- **Two Co-processors (COP)**
  - Crypto & compression accelerators
  - Includes 16KB cache
  - Shared by two cores

- **24MB eDRAM L3 Cache**
  - Shared by all four cores

- **Interface to SC chip / L4 cache**
  - 41.6 GB/sec to each of 2 SCs

- **I/O Bus Controller (GX)**
  - Interface to Host Channel Adapter (HCA)

- **Memory Controller (MC)**
  - Interface to controller on memory DIMMs
  - Supports RAIM design

- **Chip Area – 512.3mm²**
  - 23.5mm x 21.8mm
  - 8093 Power C4’s
  - 1134 signal C4’s

- **12S0 45nm SOI Technology**
  - 13 layers of metal
  - 3.5 km wire

- **1.4 Billion Transistors**
z196 SC Chip Detail

12S0 45nm SOI Technology
13 layers of metal

Chip Area –
478.8mm^2
24.4mm x 19.6mm
7100 Power C4’s
1819 signal C4’s

1.5 Billion Transistors
1 Billion cells for eDRAM

eDRAM Shared L4 Cache
96 MB per SC chip
192 MB per Book

6 CP chip interfaces
3 Fabric interfaces
2 clock domains
5 unique chip voltage supplies
z196 Book Layout

- Backup Air Plenum
- 16X DIMMs 100mm High
- MCM @ 1800W Refrigeration Cooled or Water Cooled
- 8 I/O FAN OUT 2 FSP
- 2 FSP
- 3x DCA
- 11 VTM Card Assemblies 8 Vertical 3 Horizontal
- 14X DIMMs 100mm High
- DCA Power Supplies
- Memory
- Cooling from/to MRU
- Fanout Cards

Rear

Front