IBM zSeries Mainframes

~ Development
IBM Corporation
Charles F. Webb
Evolution of z/Architecture

- 1960s S/360: 24-bit address
- 1970s S/370: virtual address/DAT
- 1980s 370/XA: 31-bit address, new I/O
- 1990s ESA/390: multiple address spaces
- 1998 added IEEE 754 Standard
- 2000 z/Architecture: 64-bit address
Uni Processor Performance

<table>
<thead>
<tr>
<th>Year</th>
<th>Relative Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>75</td>
<td>2</td>
</tr>
<tr>
<td>80</td>
<td>3</td>
</tr>
<tr>
<td>85</td>
<td>5</td>
</tr>
<tr>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td>95</td>
<td>20</td>
</tr>
<tr>
<td>96</td>
<td>30</td>
</tr>
<tr>
<td>97</td>
<td>50</td>
</tr>
<tr>
<td>98</td>
<td>100</td>
</tr>
<tr>
<td>99</td>
<td>200</td>
</tr>
</tbody>
</table>

CMOS-94 = 9672-R1
CMOS-95 = 9672-G4
CMOS-96 = 9672-G3
CMOS-97 = 9672-G5
CMOS-98 = 9672-G6
CMOS-99 = 9672-G6

Performance x 2 every 4 to 5 years
Performance x 2 every 12-24 months
Processor Organization

- Tags
- 256K I-Cache
- BTB
- Inst Fetch
- Inst Buffer
- Decode
- Inst Queue
- AGen
- ST Buf
- AG
- Inst Queue
- BTB
- Fetch
- Decode
- Inst Queue
- AGen
- ST Buf
- DAT Engine
- Co-Processor
- Compare
- Checkpoint State
- GRs
- Operand Buffer
- FXU Execution
- FPRs
- FPU Execution

Data
Address
Inst/Cntl
CBus
CP Pipeline

IF, Dec, Agen, C1, C2, E1, WR

Decode / Op Fetch

Execute/Commit

I Fetch

Ifetch request

ADDR

Cache access

Data

I Buff

I REG

Decode

B, X, D

Op Req

Addr

Address

Cache

Data

Op Buff

Operands

Instr

Exec

Result

Stage

Check

Stage

Chkpt
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP Chip</td>
<td>17.9mm x 9.9mm – 47 million transistors</td>
</tr>
<tr>
<td>D Cache</td>
<td>256 KB</td>
</tr>
<tr>
<td>I Cache</td>
<td>256 KB</td>
</tr>
<tr>
<td>BTB</td>
<td>2k x 4</td>
</tr>
<tr>
<td>I Unit A</td>
<td></td>
</tr>
<tr>
<td>FXU A</td>
<td></td>
</tr>
<tr>
<td>FPU A</td>
<td></td>
</tr>
<tr>
<td>I Unit B</td>
<td></td>
</tr>
<tr>
<td>FXU B</td>
<td></td>
</tr>
<tr>
<td>FPU B</td>
<td></td>
</tr>
<tr>
<td>R Unit</td>
<td></td>
</tr>
<tr>
<td>Compression Translator</td>
<td></td>
</tr>
</tbody>
</table>
Binodal L2 System

16 byte bidi

Between SCs
2 uni 16 byte

4 busses
To MEM
16 byte bidi

4 busses
To MEM
Millicode

- Licensed Internal Code layer for complex functions
  - System/control ops, interruptions, service operations, etc.

- Variant of z/Architecture ISA
  - Unique GRs and ARs
  - Includes all hardwired z/Architecture ops
  - Modest set of millicode-only ops
  - Access to all processor state via R-Unit

- Millicode mode entered under hardware control
  - Mode-changing branch with minimal context switch

- Uses same instruction pipeline as normal code
  - Minimal unique hardware

- Enables architectural and design flexibility
  - New ops and features, workarounds
  - Full CISC support with manageable complexity
R-Unit

- Focal point for hardware fault checking
  - Mirrored unit comparators and other checkers
- Buffers entire processor architected state
  - GRs, FPRs, ARs, CRs, PSW
  - Millicode CRs, SysReg, Timing facility, etc.
- Maintains CP checkpoint for recovery
  - Processor state protected via ECC or equivalent
  - Granularity: every HW instruction (regular or millicode)
- Provides R/W access to processor state
  - Millicode special ops plus a few hardwired z/Arch ops
  - State mapped into 256 x 64-bit register space
R-Unit

CBus-A
CBus-B

IU/EU-A
IU/EU-B

Error Checkers
Recovery Control

ECC Gen Check
Buffer

ECC Gen Check
Buffer

Checkpoint State-Registers
Timing Facility
System Registers

Inst Addr
PSW

Write Address
Read Address

EU-A
EU-B
IU-A
IU-B

EU-A
EU-B

System Operations
Async Interrupts

Processor State
Fault Checking

- Combination of checking schemes used
  - Mirrored units: complex logic and dataflow
  - Parity check: byte-coherent dataflow, BTB, etc.
  - Functional / state checks: cache controls, co-processor
  - ECC / duplicate parity: checkpoint state in R-Unit

- All processor state updates sent to R-Unit
  - Checked on hardware-instruction granularity

- Results committed to checkpoint only if clean
  - All mirrored compares equal
  - No faults detected anywhere in processor

- Target: near-100% detection of hardware faults
  - Both hard/permanent and soft/transient varieties
CP Chip - Checking Strategy by Unit
zSeries RAS Priorities

1. Ensure data integrity
   • Requires ~100% error detection
   • zSeries is industry leader

2. Keep applications on the air
   • Whenever #1 is not compromised
   • Requires fine-grained recovery
   • zSeries is industry leader

3. Repair on-line
   • Primarily 2nd level packaging constraint
   ➢ Crash and Re-boot is not good enough!
Fault Recovery

I-Unit (unchecked) -> Cache (parity) -> I-Unit (mirror)
E-Unit (unchecked) -> R-Unit (ECC on saved state) -> E-Unit (mirror)

Check all state updates
Preserve known good state
If error
  Stop state updates
  Refresh from saved state
  Restart CPU
If error persists
  Extract saved state (SE)
  Load into spare CPU
  Start spare CPU

Address
Cache data
Instructions
Results / state updates
Saved state data
Dynamic CPU Sparing

Operating CPU
Check all state updates
Preserve known good state
If error
Stop state updates
Refresh from saved state
Restart CPU
If error persists
Signal service processor

Service Processor
Extract saved state from CPU
Process CPU state
Adjust CPU numbers
Check for special conditions
Store CPU state in memory
Signal Spare CPU

Spare CPU
Wait in idle loop until needed
Load CPU state from buffer
Special CPU instruction
Replace R-Unit contents
Refresh CPU state
Restart CPU with new state
Other RAS Features

- **CP Arrays (caches / tags / TLBs / BTB)**
  - Data stored through to L2 to get ECC protection
  - Line and set deletion for persistent array faults

- **L2 and Memory**
  - ECC on arrays and busses
  - Retry on failing commands
  - DRAM chip sparing

- **I/O Subsystem**
  - Multiple paths to devices
  - Multiple identical hubs / channels
  - Retry on failing commands

- **Power / Cooling / Service**
  - N+1 redundancy
Conclusion

- Custom CISC Microprocessor
- Durable Design Point
- Industry-Leading RAS
- More to Come
Want to know more?


Questions? cfw@us.ibm.com
Thanks for Listening!