

# IBM zSeries Mainframes

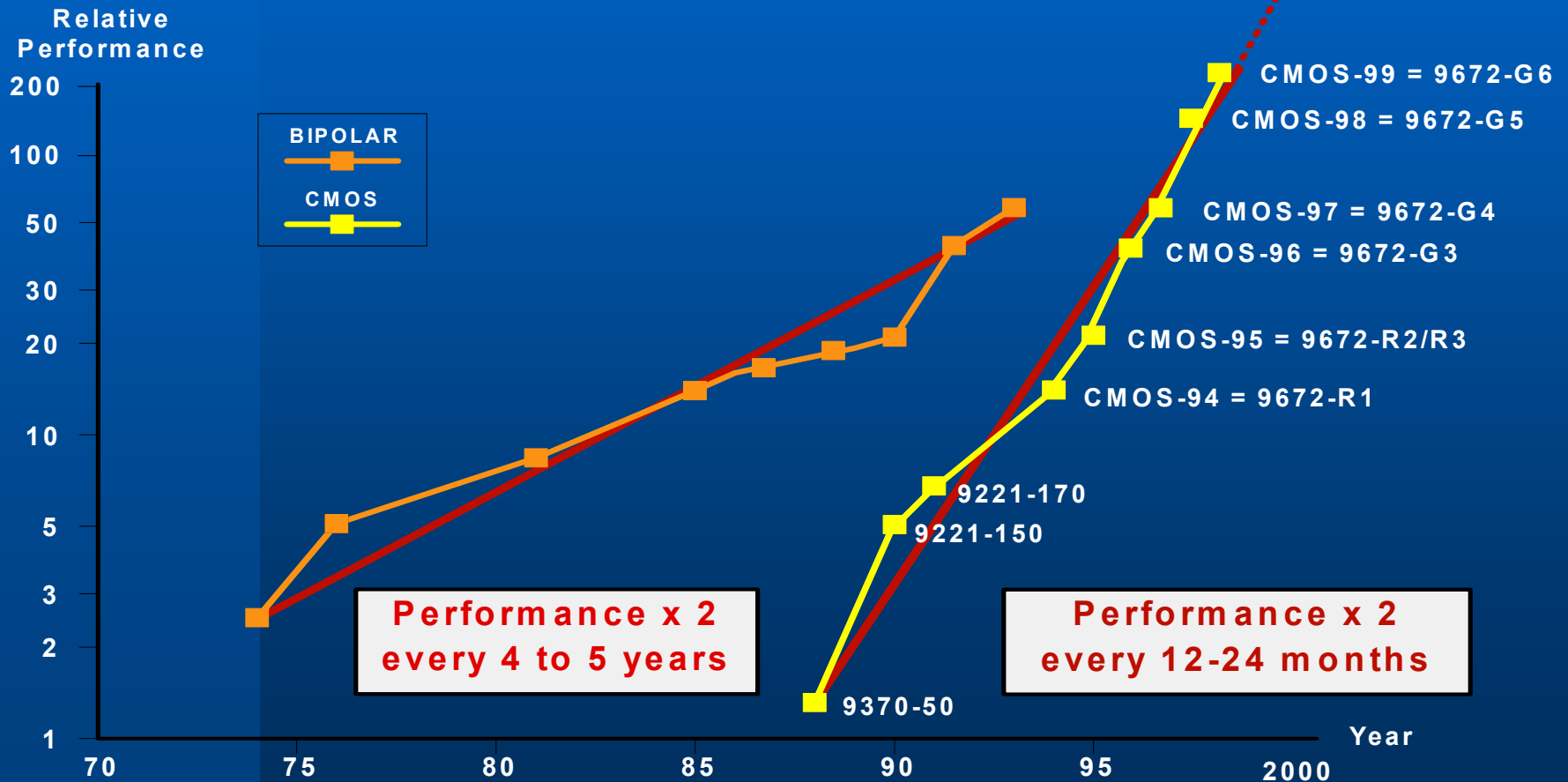


~ Development  
IBM Corporation  
Charles F. Webb

# 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





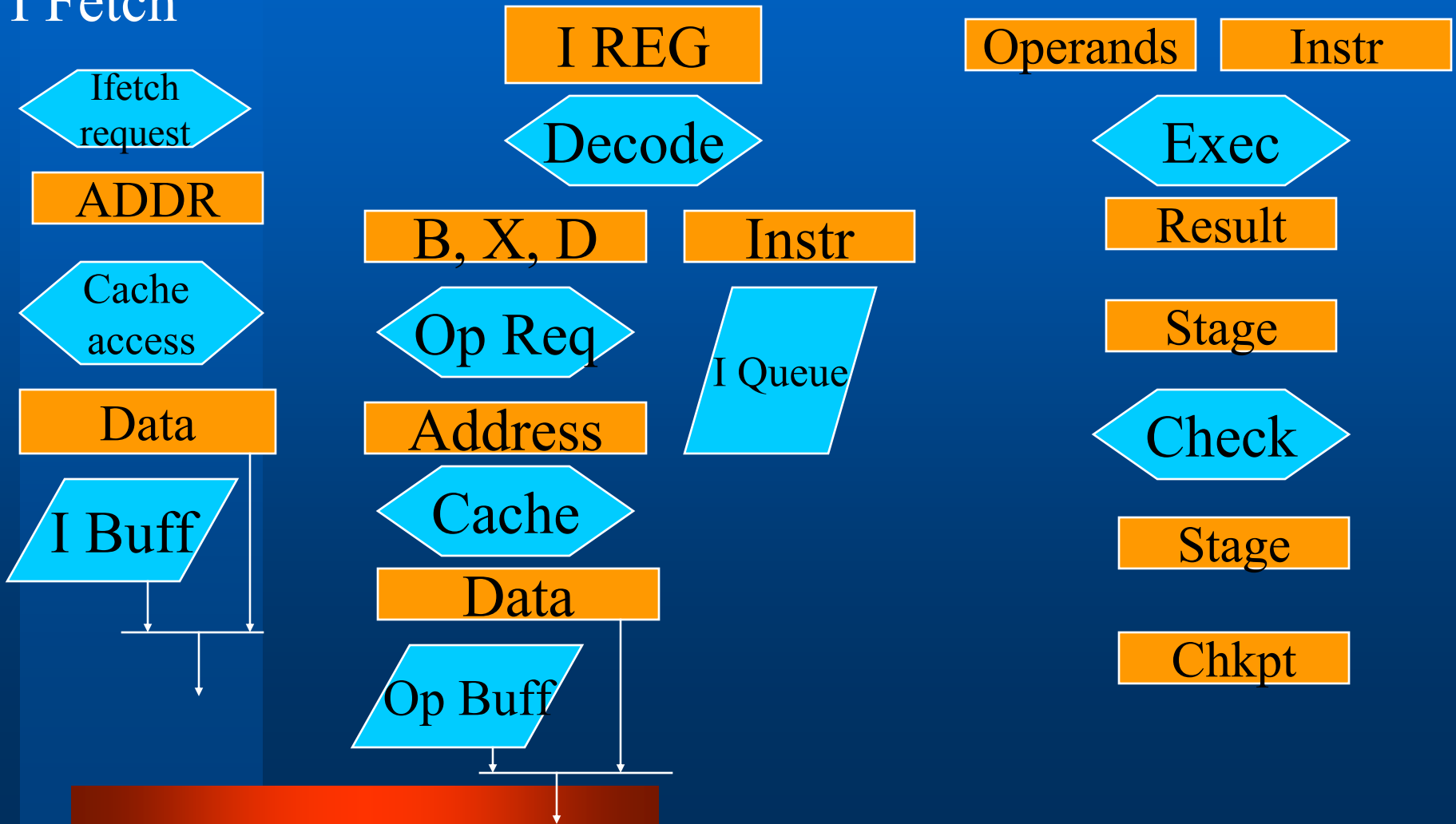
# CP Pipeline

IF, Dec, A gen, C1, C2, E1, WR

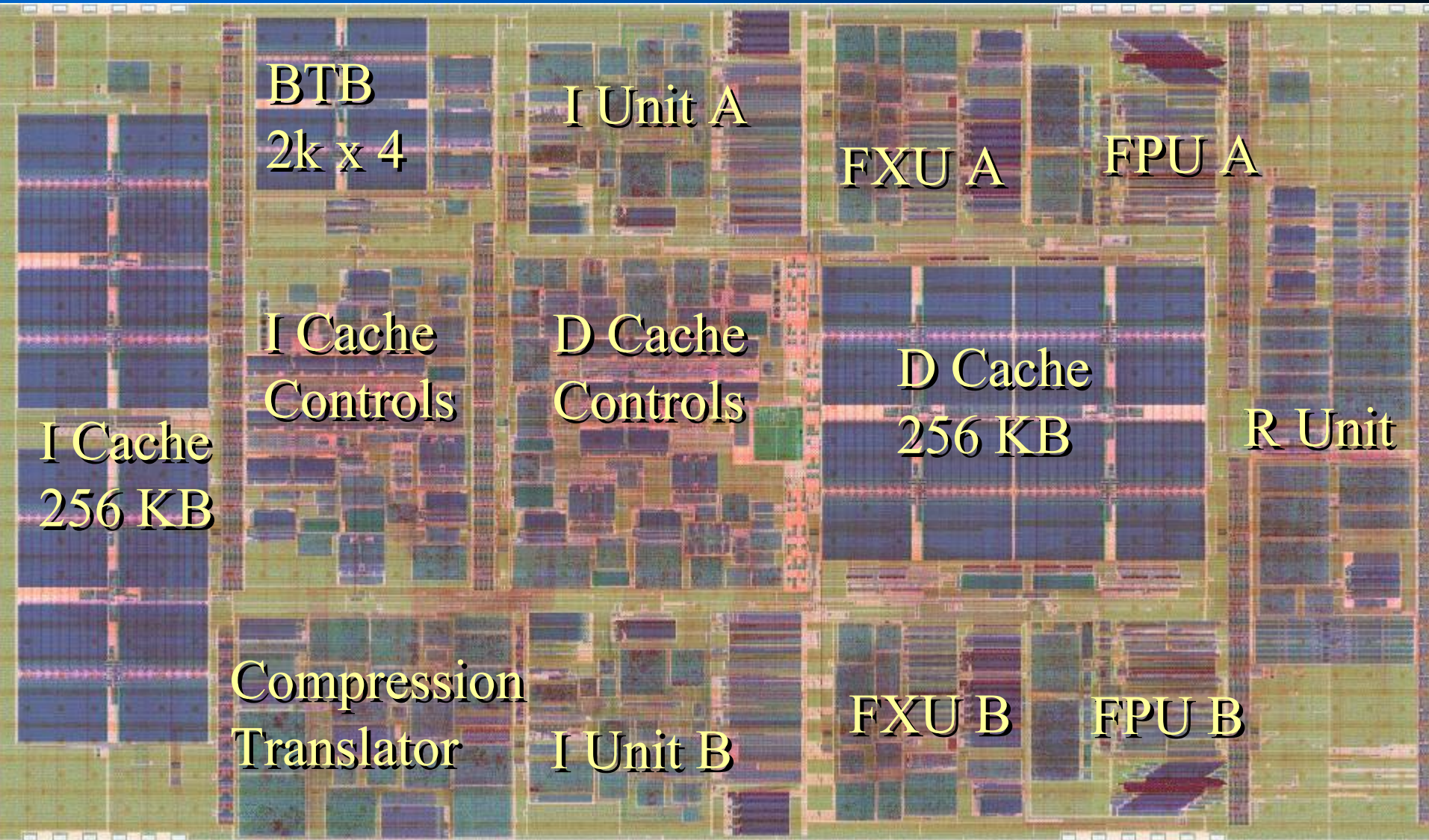
Decode / Op Fetch

Execute/Commit

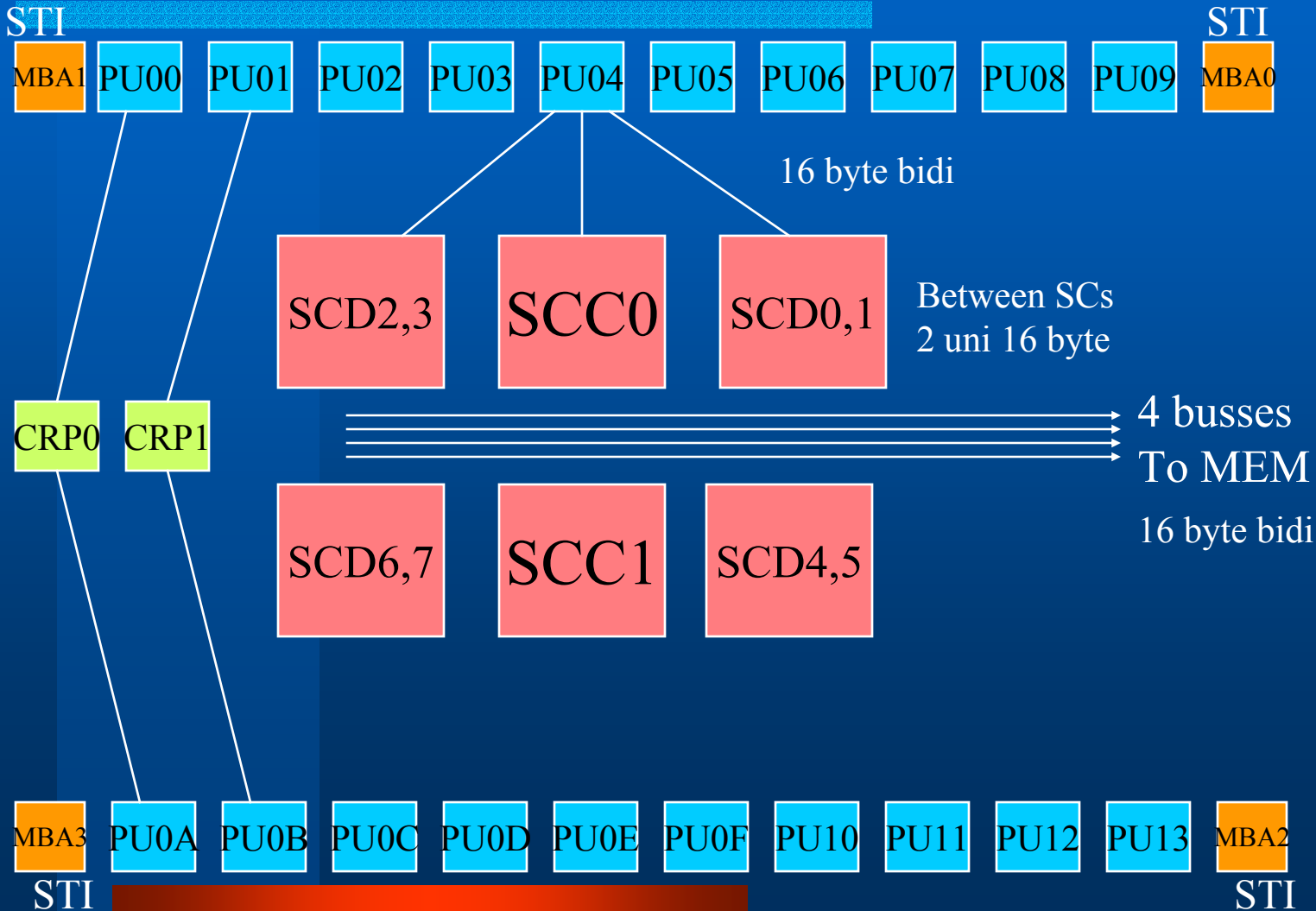
I Fetch



# CP Chip - 17.9mm x 9.9mm – 47 million transistors



# Binodal L2 System



# Millicode

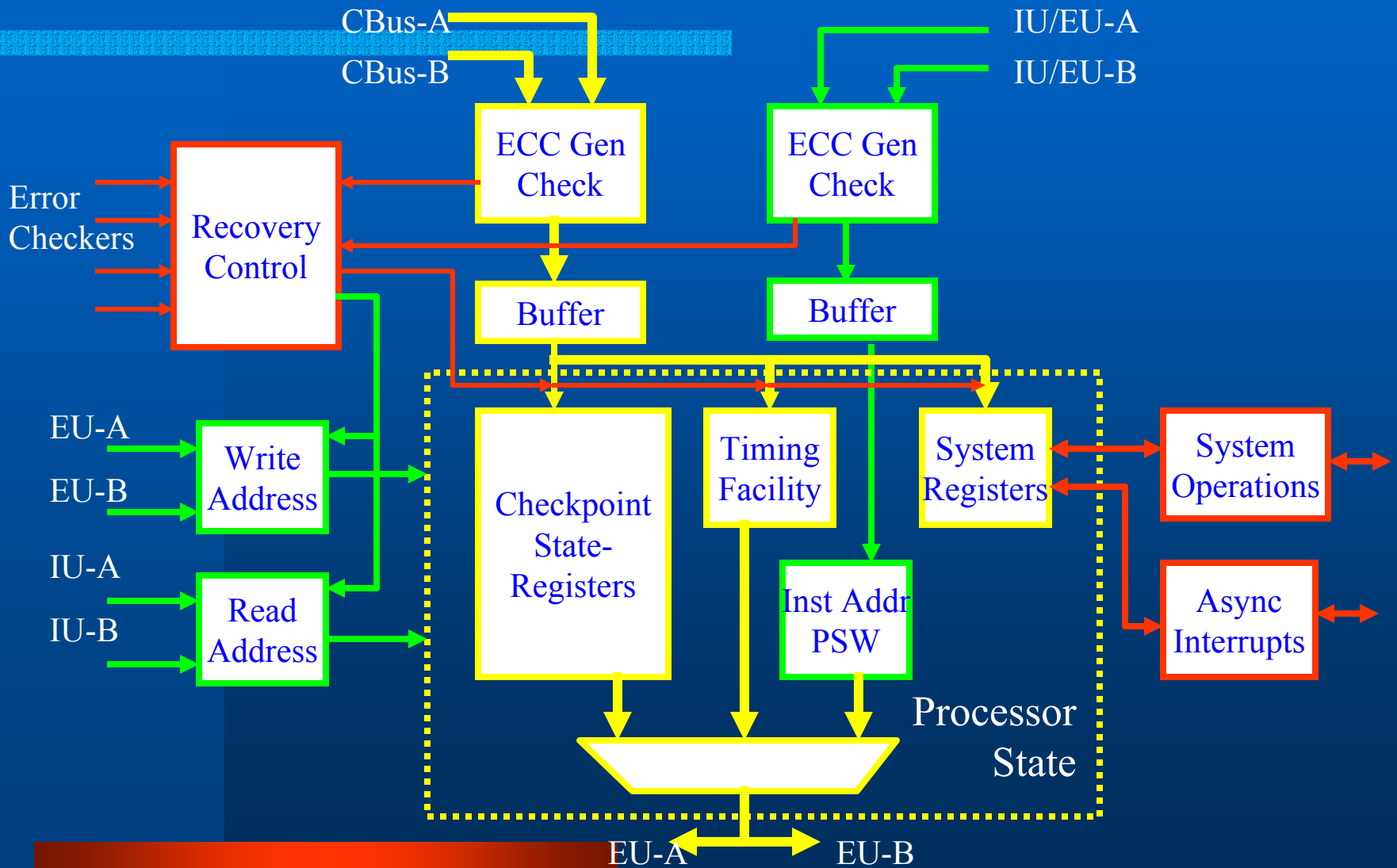
- **Licensed Internal Code layer for complex functions**
  - System/control ops, interruptions, service operations, etc.
- **Variant of z/Architecture ISA**
  - Unique GRs and ARs
  - Includes all hardwired z/Architecture ops
  - Modest set of millicode-only ops
  - Access to all processor state via 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, SysRegs, 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

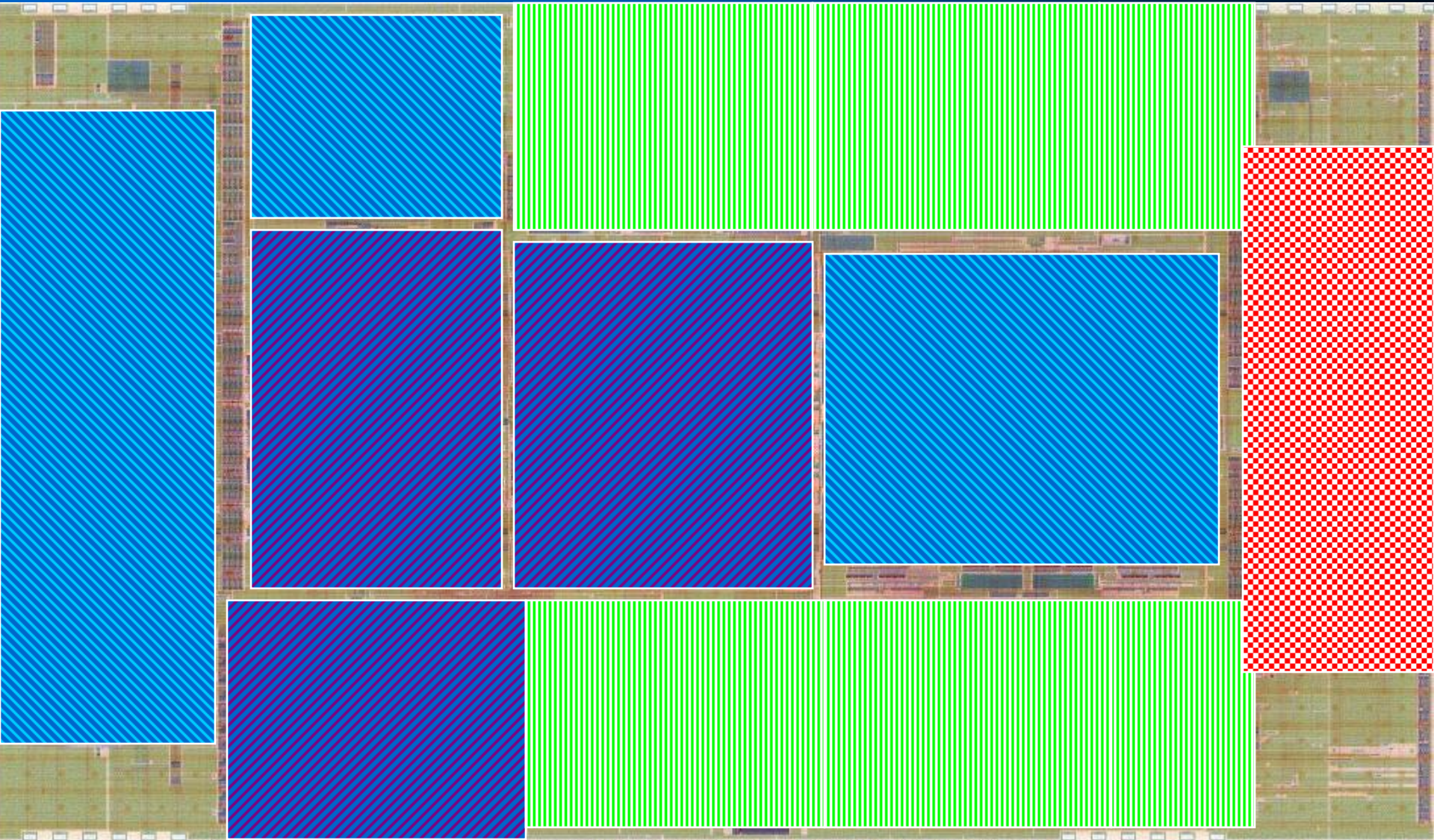
# R-Unit



# 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

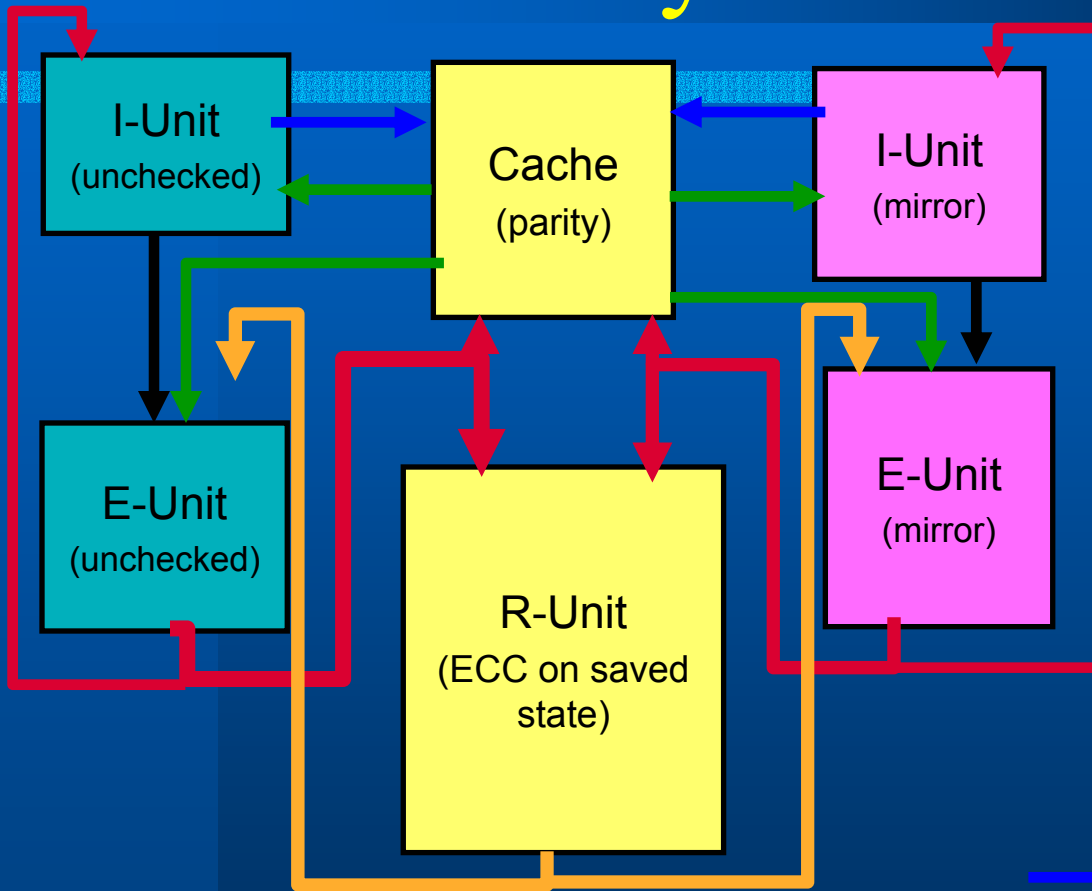
- Whenever #1 is not compromised
- Requires fine-grained recovery
- zSeries is industry leader

## 3. Repair on-line

- Primarily 2nd level packaging constraint

➤ **Crash and Re-boot is not good enough!**

# Fault Recovery



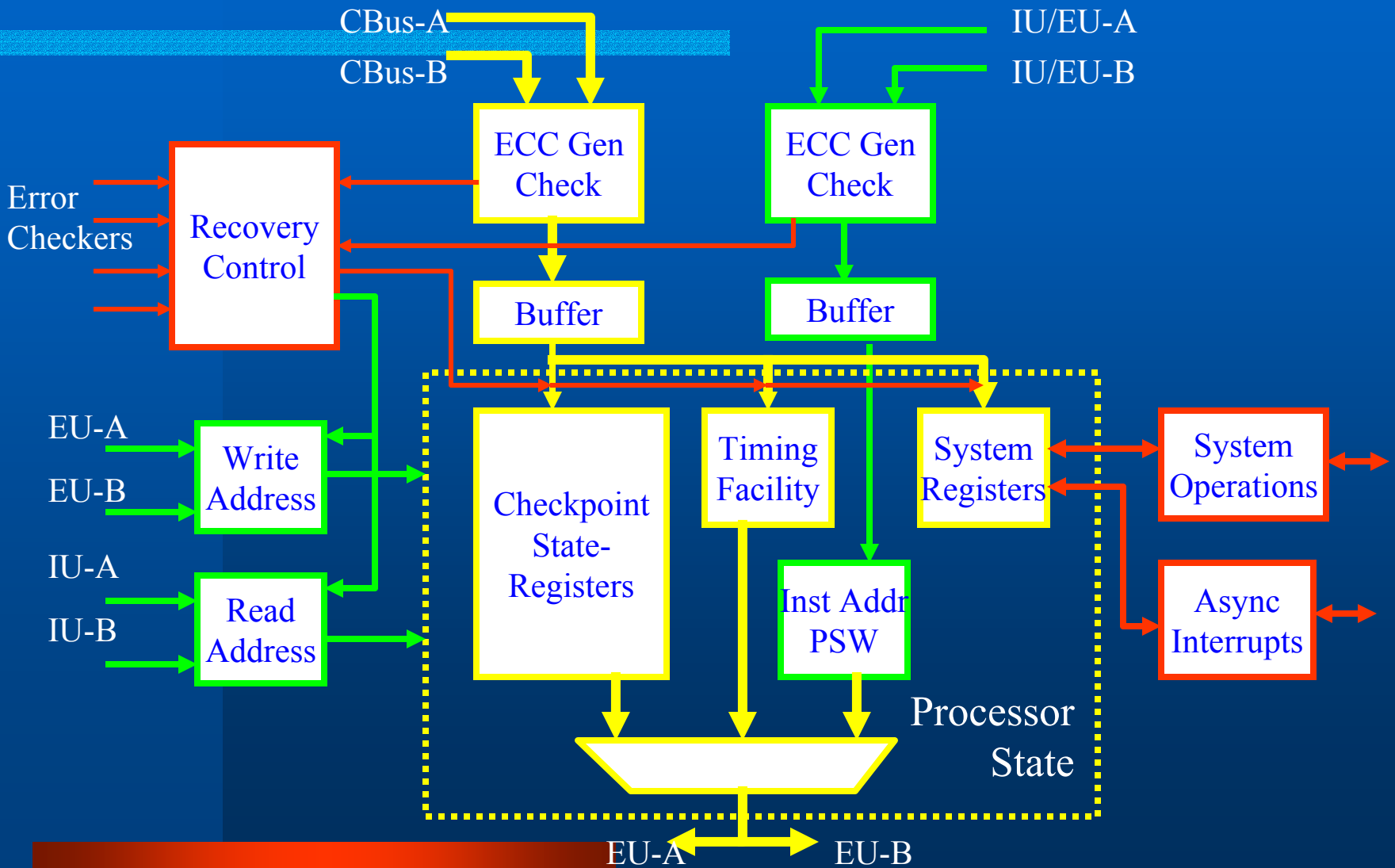
Check all state updates  
Preserve known good state  
If error

Stop state updates  
Refresh from saved state  
Restart CPU

If error persists  
Extract saved state (SE)  
Load into spare CPU  
Start spare CPU

**Address** (Blue arrow)  
**Cache data** (Green arrow)  
**Instructions** (Black arrow)  
**Results / state updates** (Red arrow)  
**Saved state data** (Orange arrow)

# R-Unit



# Dynamic CPU Sparing

## Operating CPU

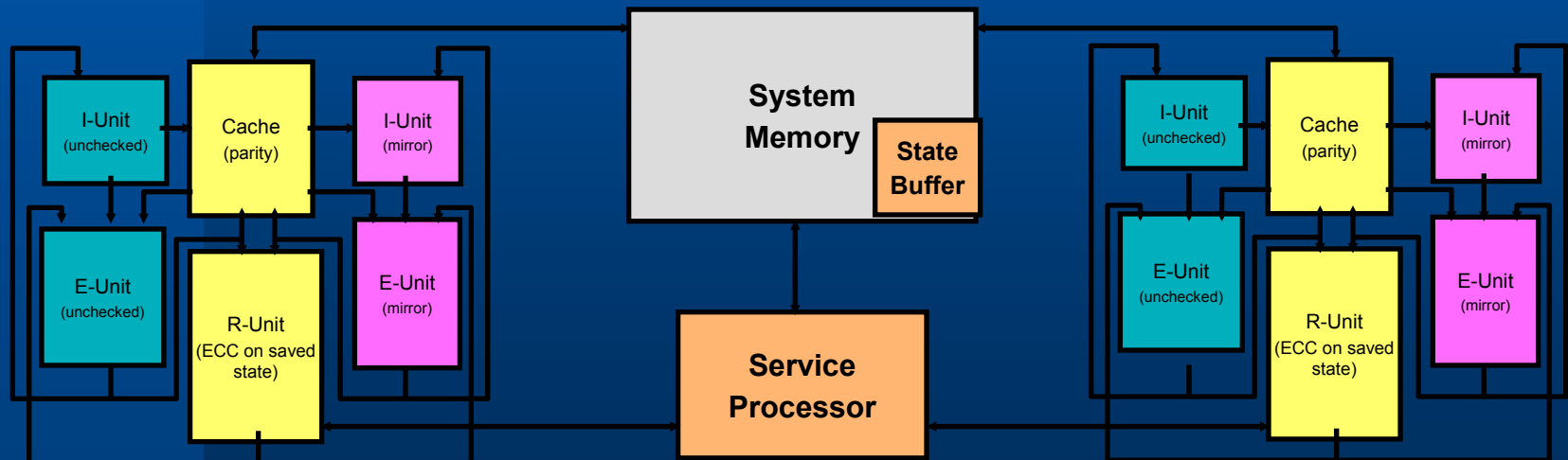
Check all state updates  
Preserve known good state  
If error  
    Stop state updates  
    Refresh from saved state  
    Restart CPU  
If error persists  
    Signal service processor

## 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?

- C.F.Webb & J.S.Liptay, “A High-frequency Custom CMOS S/390 Microprocessor”, IBM Journal of R&D, July/September 1997
- T.J.Slegel et al., “IBM’s S/390 G5 Microprocessor”, IEEE Micro, March/April 1999
- C.F.Webb, “ S/390 Microprocessor Design”, IBM Journal of R&D, November 2000
- E.M.Schwarz et al., “The Microarchitecture of the IBM eServer z900”, IBM Journal of R&D, July/September 2002
- K.E.Plambeck et al., “Development and Attributes of z/Architecture”, IBM Journal of R&D, July/September 2002
- Questions? [cfw@us.ibm.com](mailto:cfw@us.ibm.com)

# Thanks for Listening!

