Dependable Systems

Hardware Dependability - Redundancy

Dr. Peter Tröger

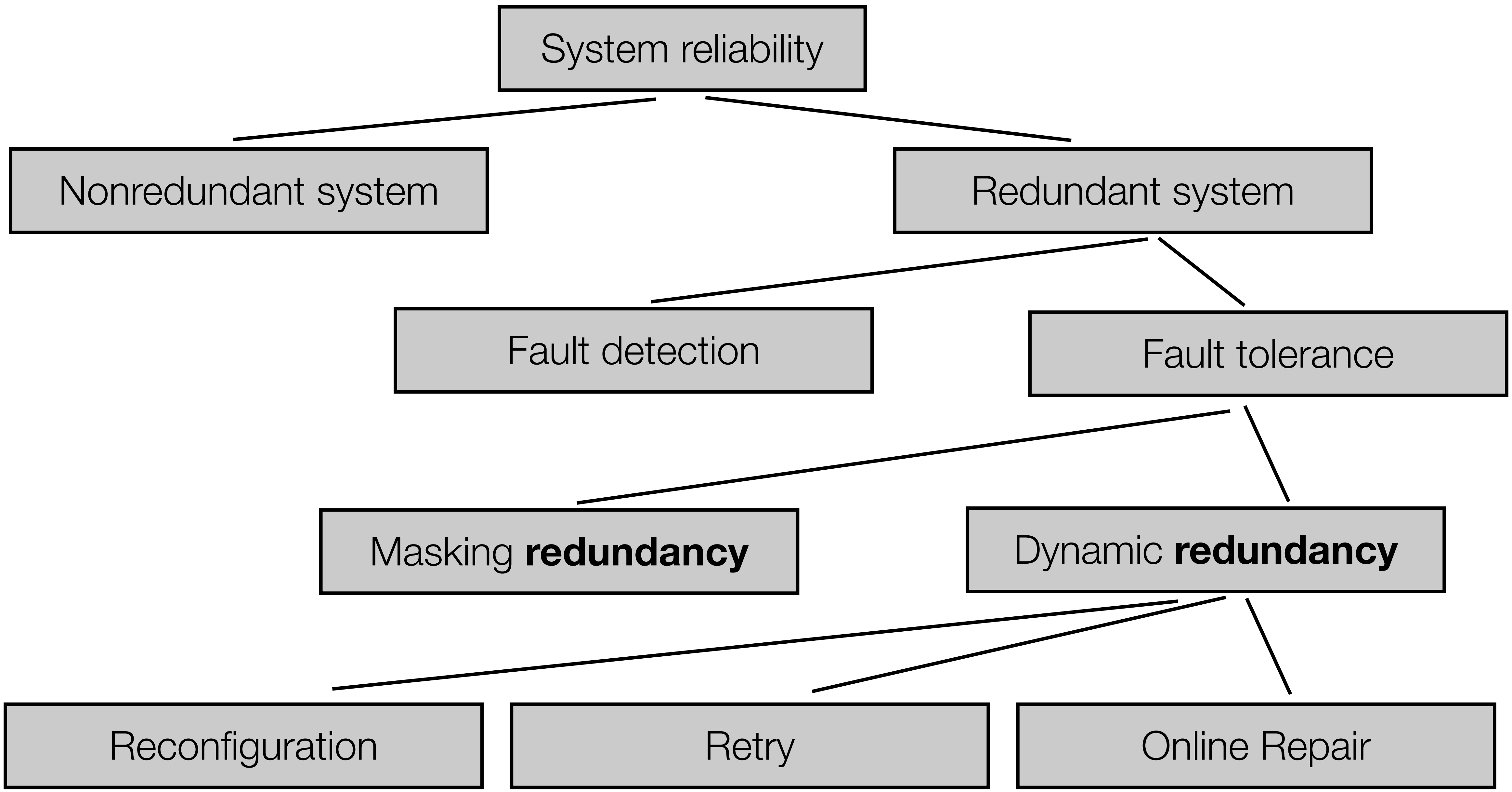
Sources:

Siewiorek, Daniel P.; Swarz, Robert S.:
Reliable Computer Systems. third. Wellesley, MA : A. K. Peters, Ltd., 1998. , 156881092X
Roland Trauner, IBM Mainframe Summit, Hasso Plattner Institute, 2012
IBM zEnterprise System Technical Guide, IBM RedBooks
Some images (C) Elena Dubrova, ESDLab, Kungl Tekniska Högskolan

Redundancy (Reiteration)

- Redundancy for **error detection** and **forward error recovery**
- Redundancy types: **spatial**, **temporal**, **informational** (presentation, version)
  - Redundant not mean identical functionality, just perform the same work
- **Static redundancy** implements error mitigation
  - Fault does not show up, since it is transparently removed
  - Examples: Voting, error-correcting codes, N-modular redundancy
- **Dynamic redundancy** implements error processing
  - After fault detection, the system is reconfigured to avoid a failure
  - Examples: Back-up sparing, duplex and share, pair and spare
- **Hybrid approaches**

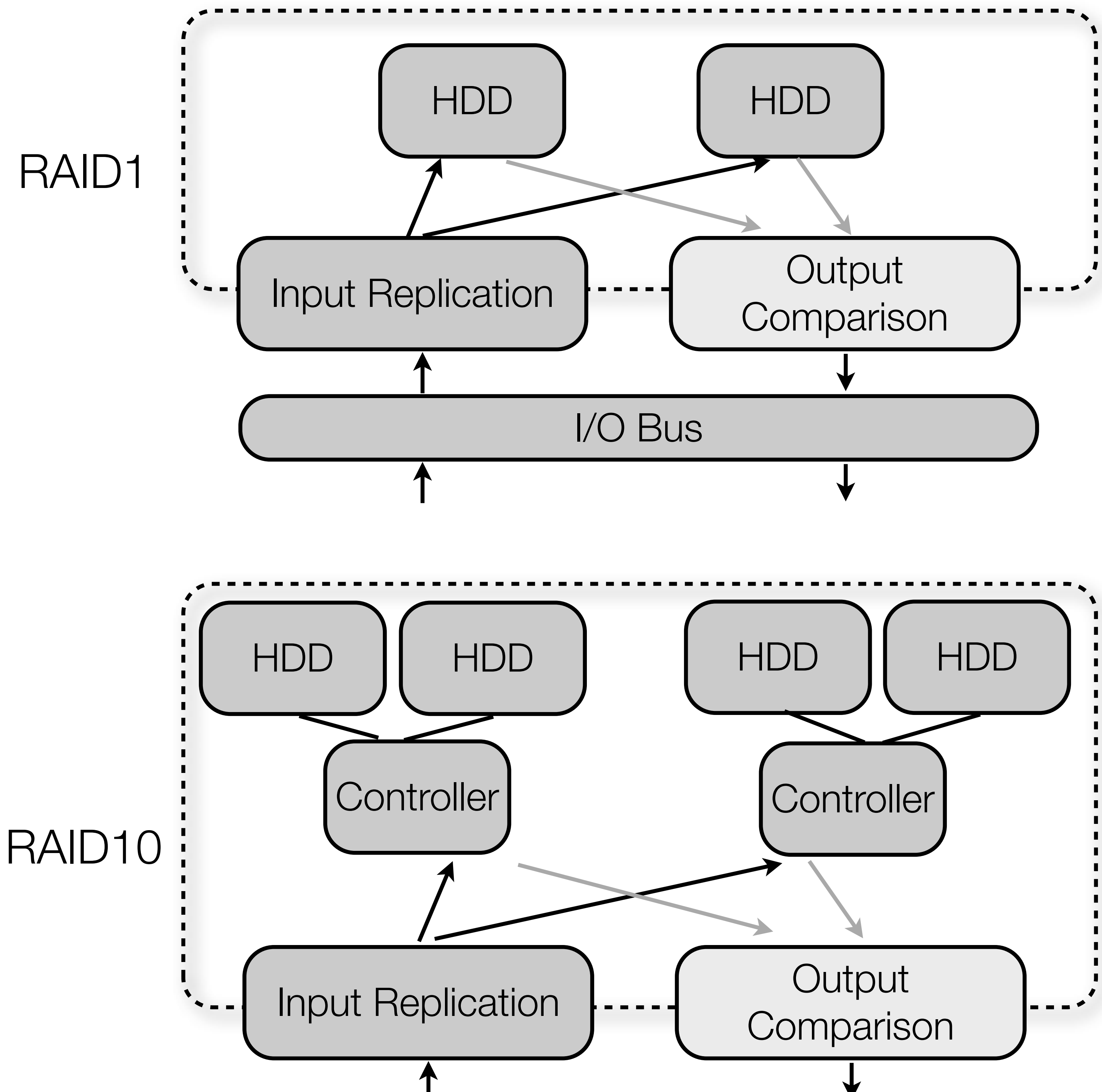
System-Failure Response Strategies [Sieworek / Swarz]



Redundancy

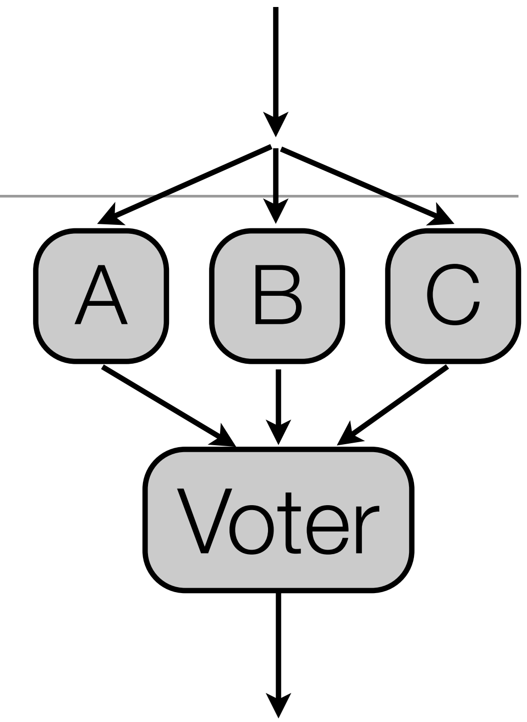
- Redundancy is never for free !
  - Hardware: Additional components, area, power, shielding, ...
  - Software: Development costs, maintenance costs, ...
  - Information: Extra hardware for decoding and encoding
  - Time: Faster processing (CPU) necessary to achieve same performance
- Tradeoff: Costs vs. benefit of redundancy; additional design and testing effort
- **Sphere of replication [Mukherjee]**
  - Identifies logical domain protected by the fault detection scheme
  - Questions: For which **components** are faults detected ?
    Which **outputs** must be **compared** ? Which **inputs** must be **replicated** ?

Sphere of Replication



- Components outside the sphere must be protected by other means
- Level of output comparison decides upon fault coverage
- Larger sphere tends to decrease the required bandwidth on input and output
    - More state changing happens just inside the sphere
- Vendor might be restricted on choice of sphere size

Masking / Static Redundancy: Voting



- **Exact voting:** Only one correct result possible
  - **Majority vote** for uneven module numbers
  - **Generalized median voting** - Select result that is the median, by iteratively removing extremes
  - **Formalized plurality voting** - Divide results in partitions, choose random member from the largest partition
- **Inexact voting:** Comparison at high level might lead to multiple correct results
  - **Non-adaptive voting** - Use allowable result discrepancy, put boundary on discrepancy minimum or maximum (e.g. 1,4 = 1,3)
  - **Adaptive voting** - Rank results based on past experience with module results
    - Compute the correct value based on „trust“ in modules from experience
    - Example: Weighted sum $R=W_1*R_1 + W_2*R_2 + W_3*R_3$ with $W_1+W_2+W_3=1$

Static Redundancy: N-Modular Redundancy

- Fault is transparently removed on detection
- **Triple-modular redundancy (TMR)**
  - 2/3 of the modules must deliver correct results
- **Generalization with N-modular redundancy (NMR)**
  - *m+1/N* of the modules must deliver correct result, with *N=2m+1*
- Standard case without any redundancy is called **simplex**

$$\begin{aligned} R_{TMR} &= R_V \cdot R_{2-of-3} \\ &= R_V(R_M^3 + 3R_M^2(1-R_M)) \end{aligned}$$

N-Modular Redundancy (with perfect voter)

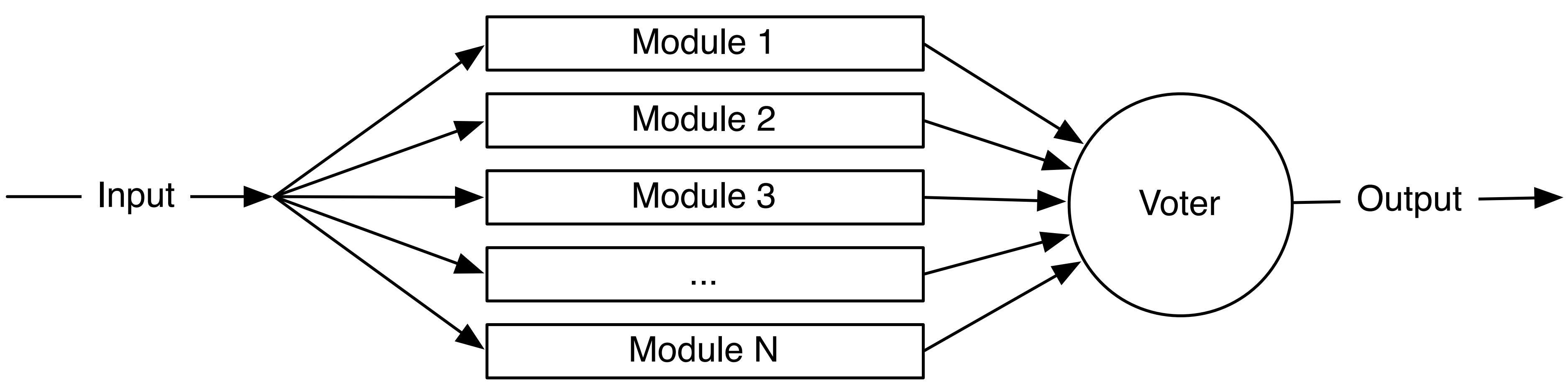
$$R_{NMR} = \sum_{i=0}^{m} \binom{N}{i} (1-R)^i R^{N-i}$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

$$R_{2-of-3} = \binom{3}{0} (1-R)^0 R^3 + \binom{3}{1} (1-R) R^2$$

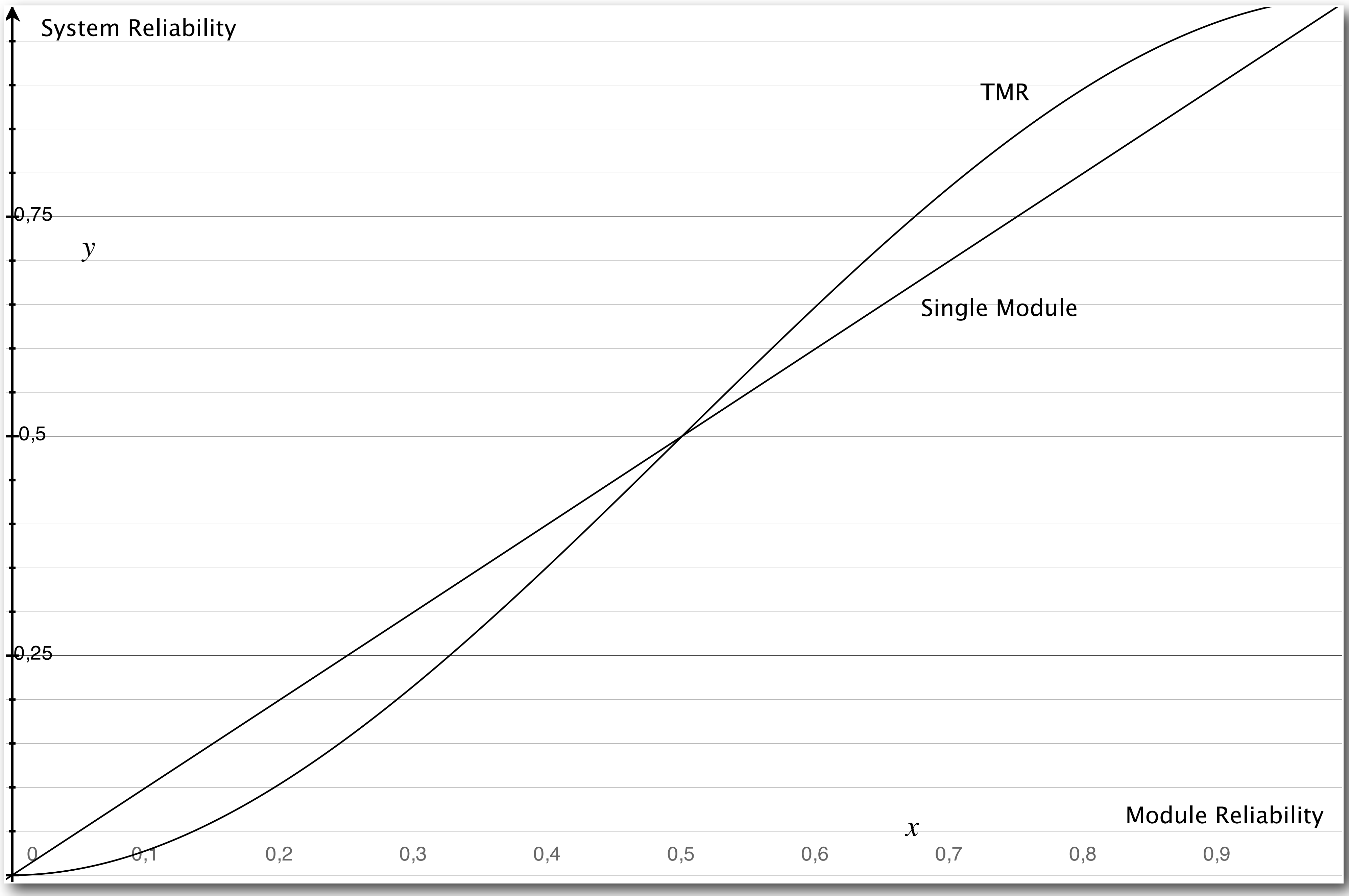
$$R_{2-of-3} = R^3 + 3(1-R)R^2$$

$$R_{3-of-5} = ...$$



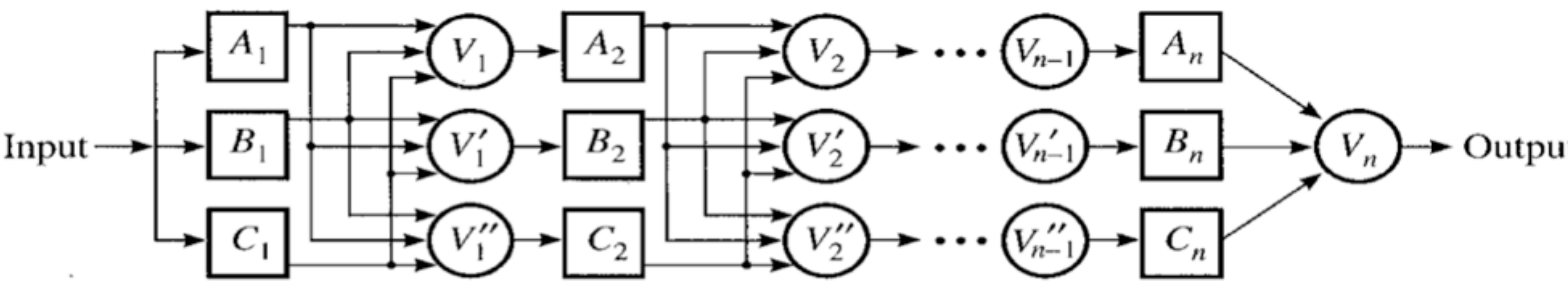
TMR Reliability

- TMR is appropriate if $R_{TMR} > R_M$ (for given t)
- TMR with perfect voter only improves system reliability when $R_M > 0.5$
- Voter needs to have $R_V > 0.9$ to reach $R_{TMR} > R_M$



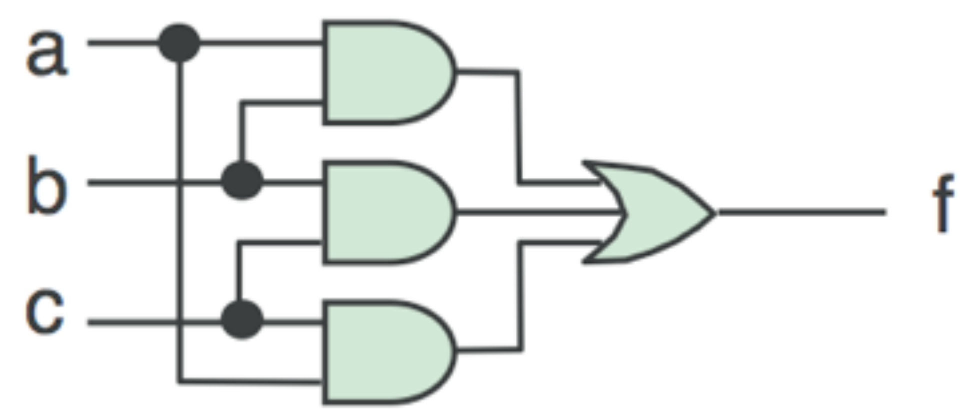
Imperfect Voters

- Redundant voters
  - Module errors do not propagate
  - Voter errors propagate only by one stage
- Assumption of multi-step process, final voter still needed



Hardware Voting

- Smallest hardware solution is the 1-bit majority voter
  - *f=ab + ac + bc*
  - Delivers the bit that has the majority
  - Requires 2 gate delays and 4 gates
- Hardware voting can become expensive
  - 128 gates and 256 flip-flops for 32-bit voter
- Input must be synchronized
  - Central clock source may be single point of failure
  - Can be solved by special event latching



A	B	Y = A ∧ B
0	0	0
0	1	0
1	0	0
1	1	1

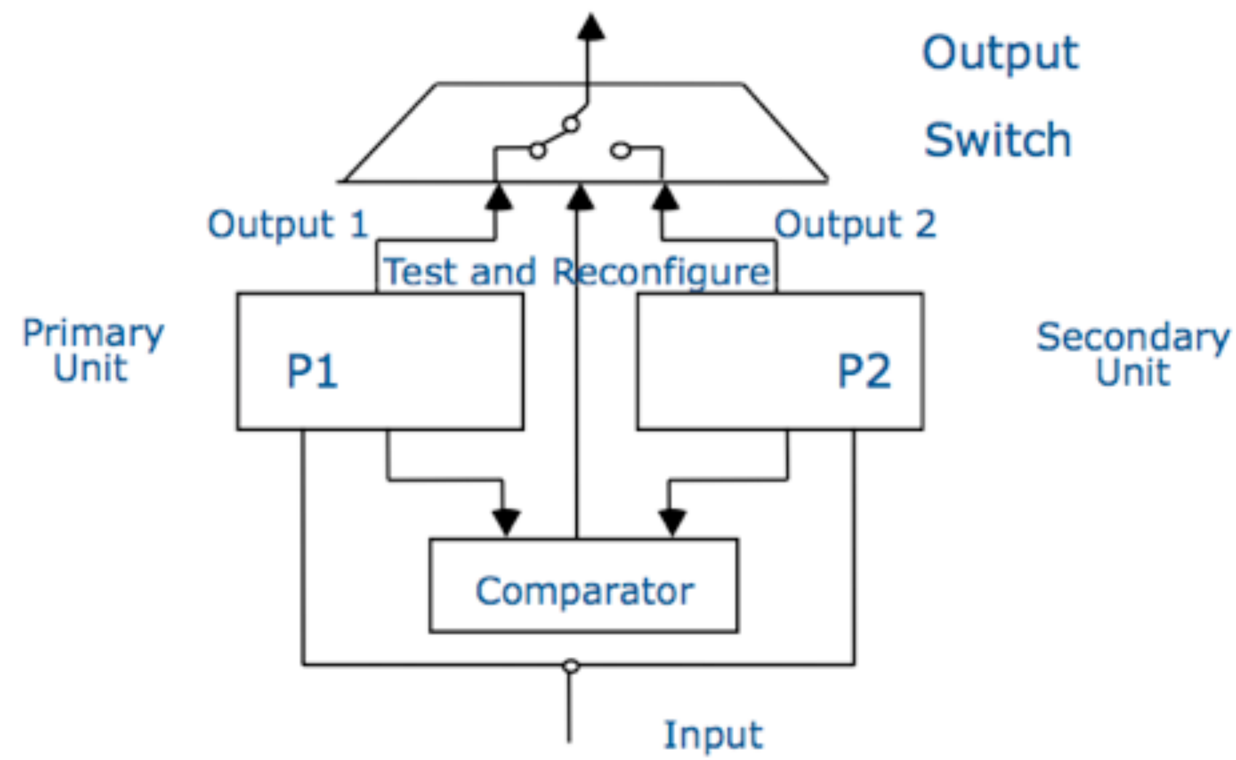
A	B	Y = A ∨ B
0	0	0
0	1	1
1	0	1
1	1	1

Dynamic Redundancy

- Reconfiguration of the system in response to an error state
  - Prevents error propagation
  - Triggered by internal fault detection in the unit, or external error detection based on the outputs
- Dynamic redundancy **combines error confinement with fault detection**
  - Still questions of coverage and diagnosability
- On transient errors, good modules may be deactivated
  - Typically solved by combination of dynamic redundancy with retry approach
- Typical approaches: Duplex, sparing, degradation, compensation

Duplex Systems

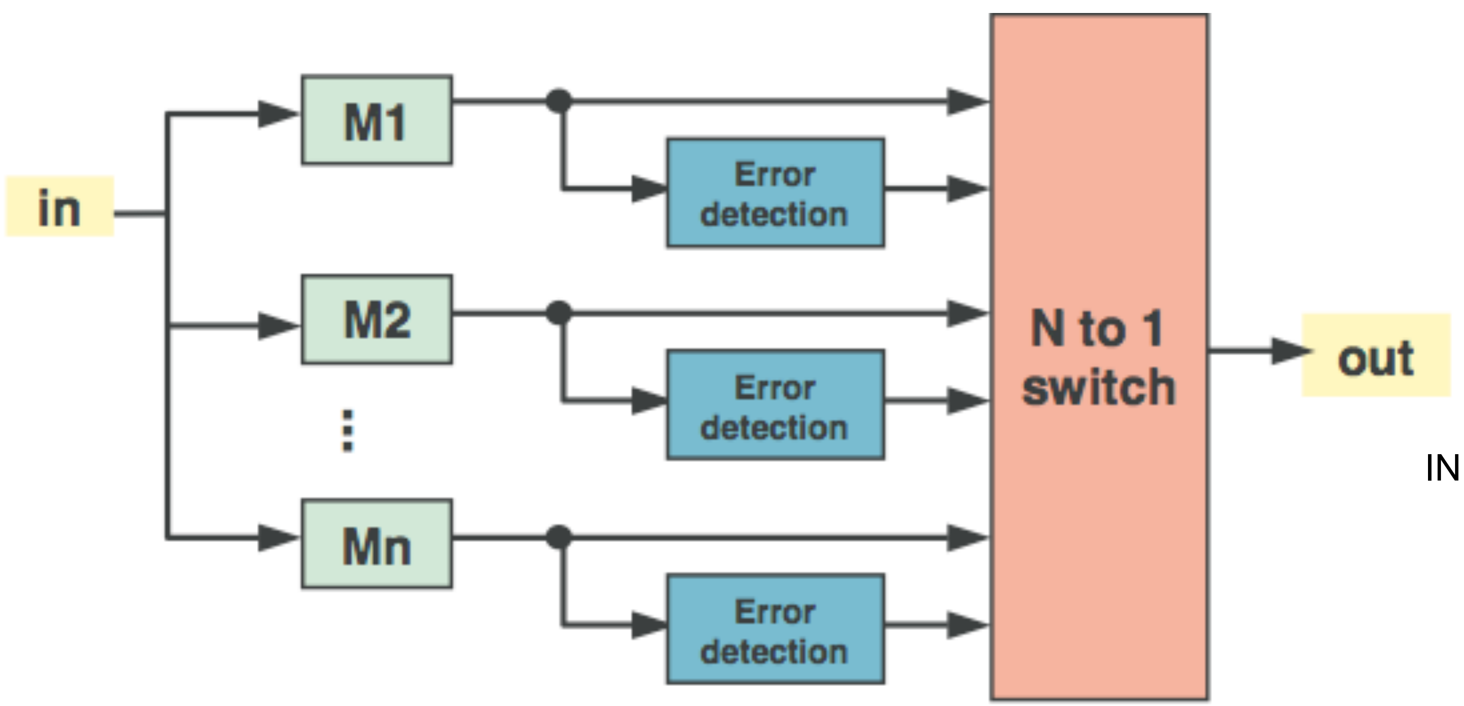
- Reconfigurable duplication : Have relevant modules redundant, switch on failure
- Identification on mismatch („test“)
    - Self-diagnostics procedure
    - Self-checking logic
    - Watchdog timer, e.g. for having components resetting each other (e.g. split brain)
    - Outside arbiter for signatures or black box tests

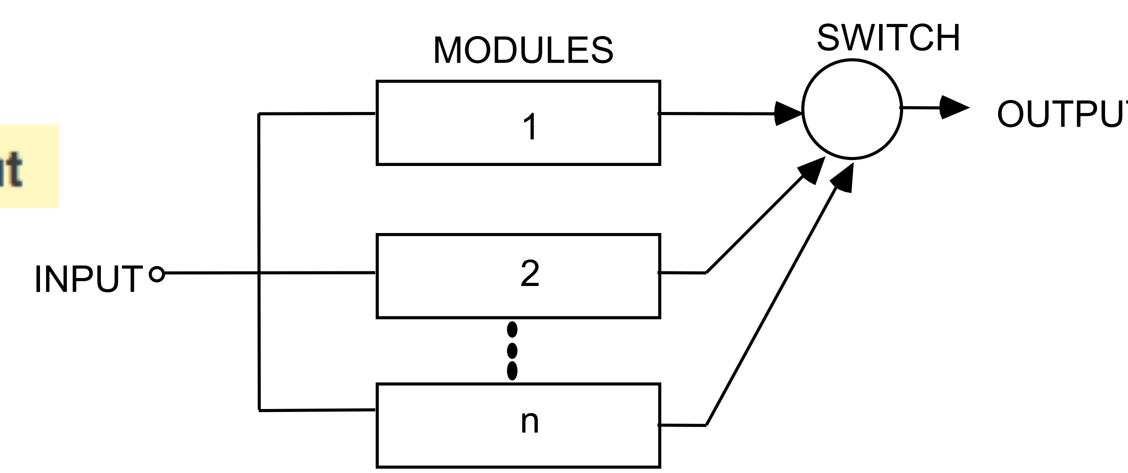


- Test interval depends on application scenario - each clock period / bus cycle / ...
- Also called **dual-modular redundancy**
- Reliability computation as with parallel / serial component diagram

Back-Up Sparing

- Combination of working module and a set of spare modules (‚replacement parts‘)
- **Hot spares**: Receive input with main modules, have results immediately
- **Warm spares**: Are running, but receive input only after switching
- **Cold spares**: Need to be started before switching

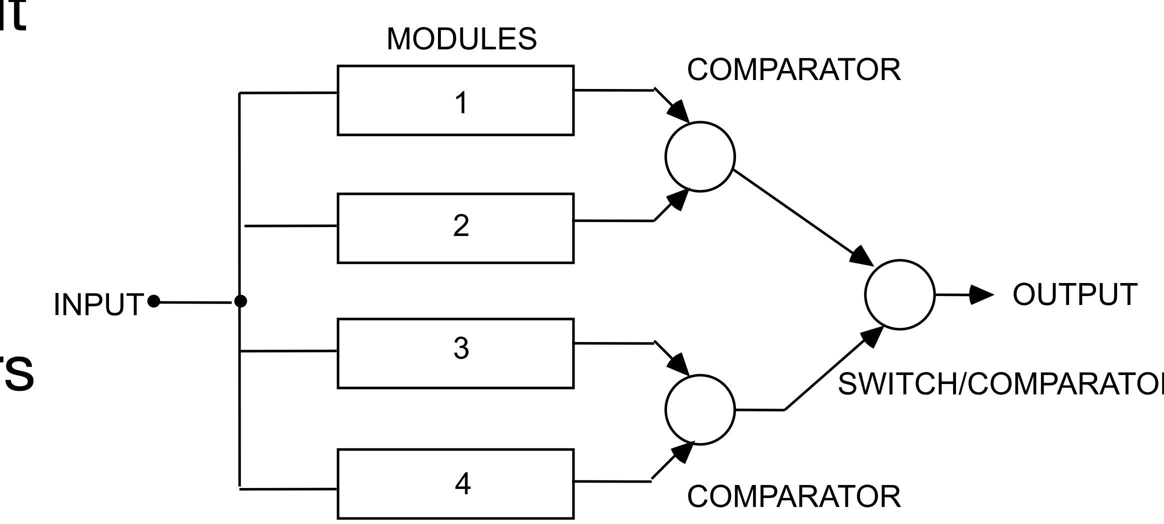


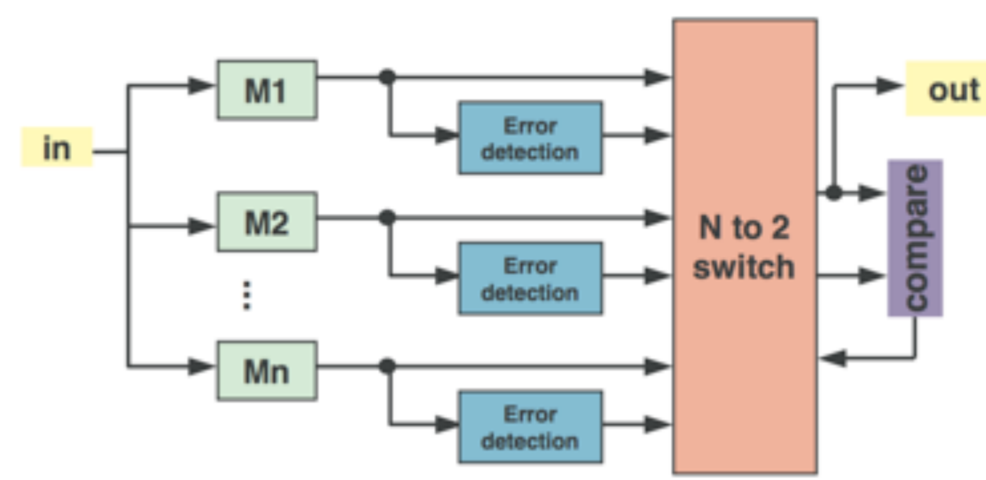


HOT, WARM AND COLD SPARES

Pair and Spare

- Special cases for combination of duplex (with comparator) and sparing (with switch)
- **Pair and spare** - Multiple duplex pairs, connected as standby sparing setup
    - Two replicated modules operate as duplex pair (lockstep execution), connected by comparator as voting circuit
    - Same setting again as spare unit, spare units connected by switch
    - On module output mismatch, comparators signal switch to perform failover
    - Commercially used, e.g. Stratus XA/R Series 300





Graceful Degradation

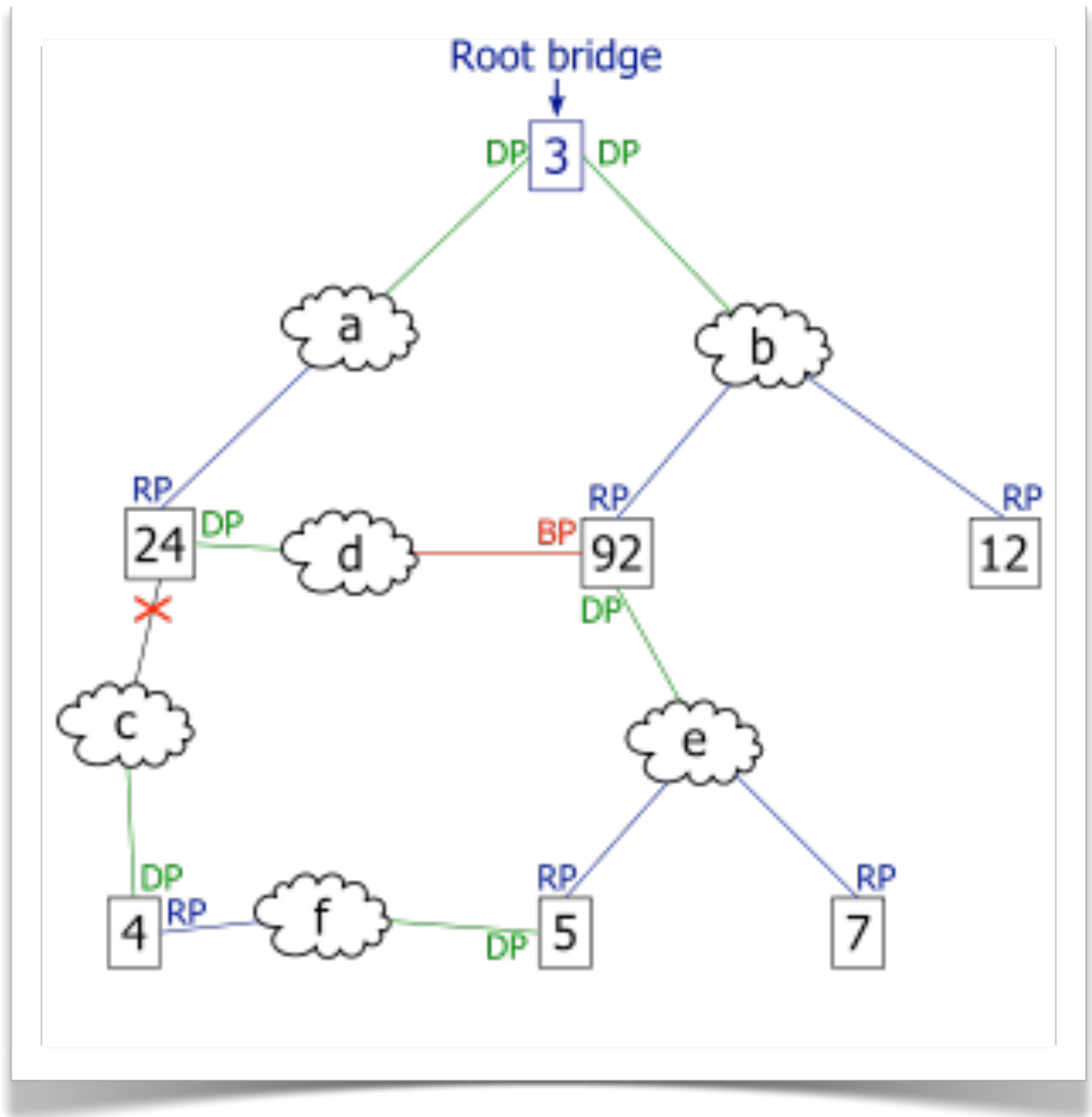
- Performance design of the system allows continued operation with spares
    - Many commercial systems supports this, but lack automated error processing
    - Example: Operating system support for CPU off-lining, but no MCA handling
- **Designed-In Resources:**
    - Replaceable or bypass-able components (f.e. caches, disks, processors)
    - Support for operation with degraded performance
- **Added-On Resources:**
    - Redundant units used for excess capacity during normal operation
    - Still non-degraded performance on failure
- **Interconnect reconfiguration**: Use alternative paths in the network
    - Hardware solutions in telco industry, today replaced by software solutions

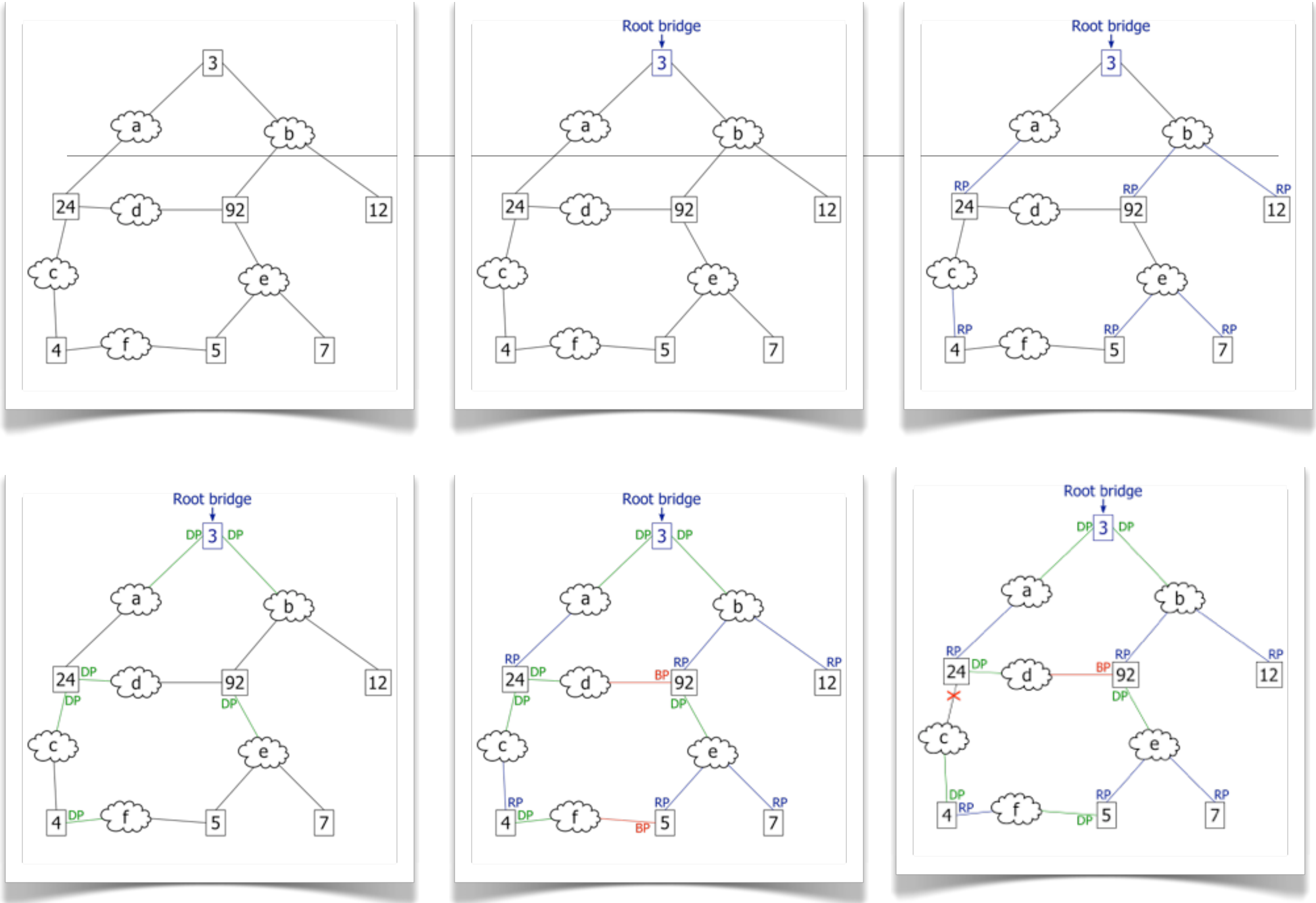
Example: Spanning Tree Protocol

- Modern implementation of **interconnect reconfiguration** for **dynamic redundancy**
- Bridges for connecting different Ethernet sub-networks
- By default no coordination, better products use the **spanning tree** protocol
  - Explicit removal of redundant paths (loops),
    while still supporting all point-to-point communication
  - Each bridge has its own MAC address, protocol based on broadcast
  - Create a tree of bridges, starting from a chosen root bridge
  - All paths start from the root bridge
  - Ports participating in redundant paths have to be switched off
  - Cost model for paths to make a choice (root distance, speed)

Example: Spanning Tree Protocol

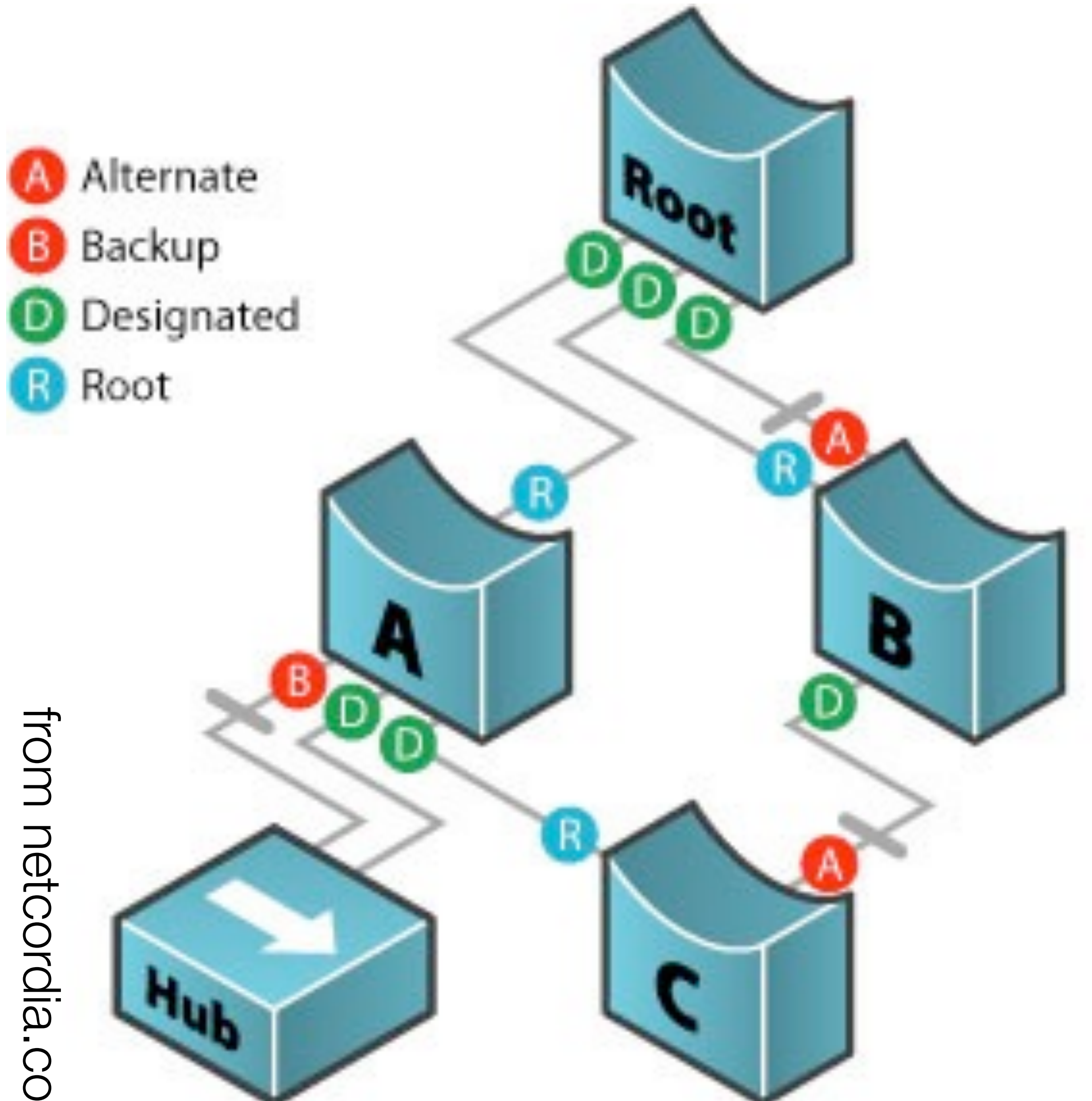
- Determine root bridge
  - Send your ID (MAC address) to a multicast group, smallest ID wins
- Each non-root bridge determines the ‚cheapest' path to the root bridge
  - This port becomes the *root port (RP)*
- For multiple bridges in a segment, the ‚cheapest' representative is elected - *designated port (DP)*
- All ports that are not DP or RP are deactivated - *blocked port (BP)*





(from Wikipedia)

Rapid Spanning Tree Protocol (RSTP)

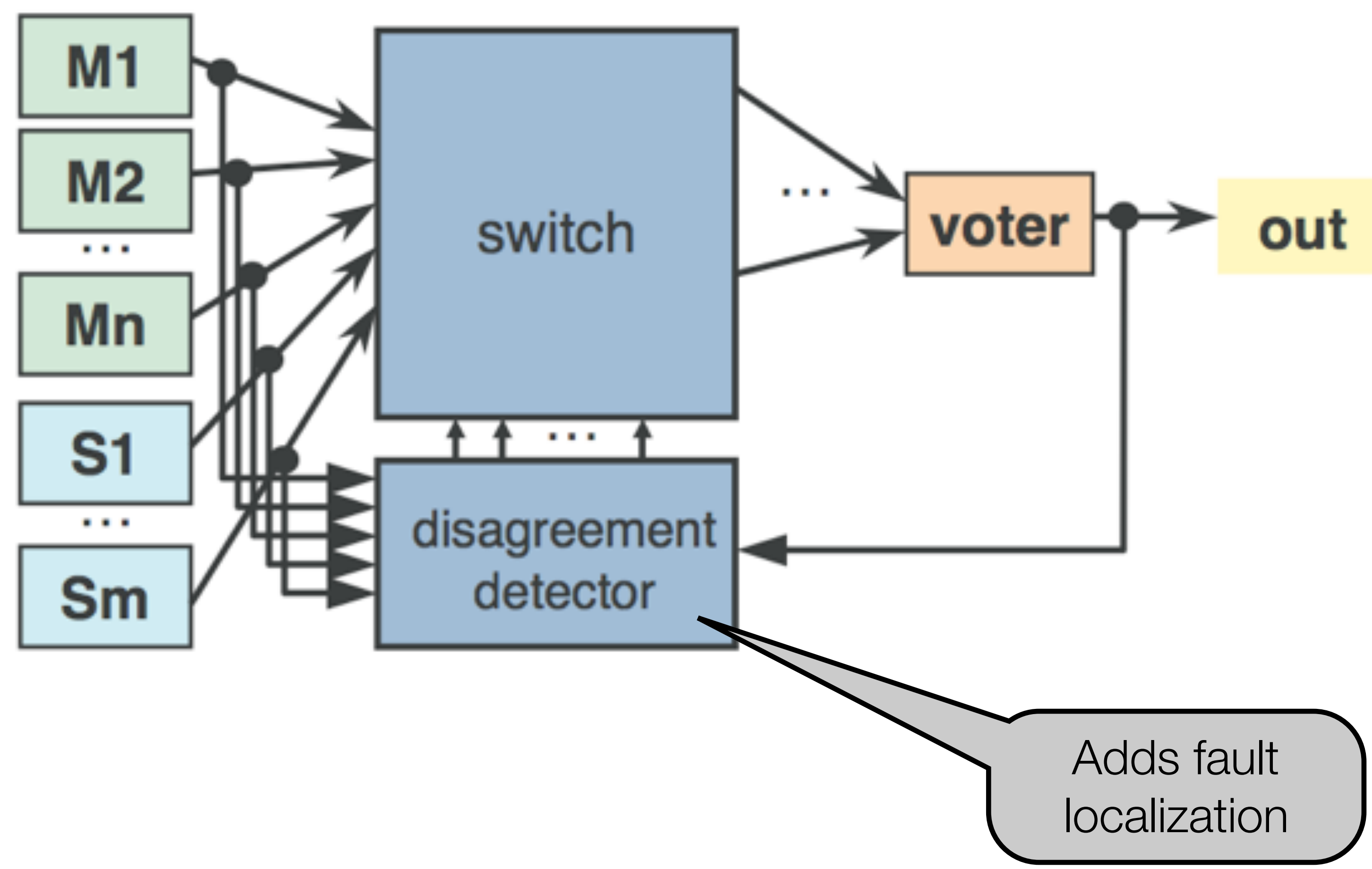


from netcordia.com

- *Alternate Port:* Blocked port to a network that can be currently reached by another bridge in a better way
  - Reduces time for re-arrangement of the tree on errors
- *Backup Port:* Blocked port to a network that can be reached by another port in a cheaper way
  - May be used in parallel, or only as update
- *Bridge Protocol Data Unit (BPDU)* is sent every other second
  - *„Hello-Time“*

Hybrid Approaches

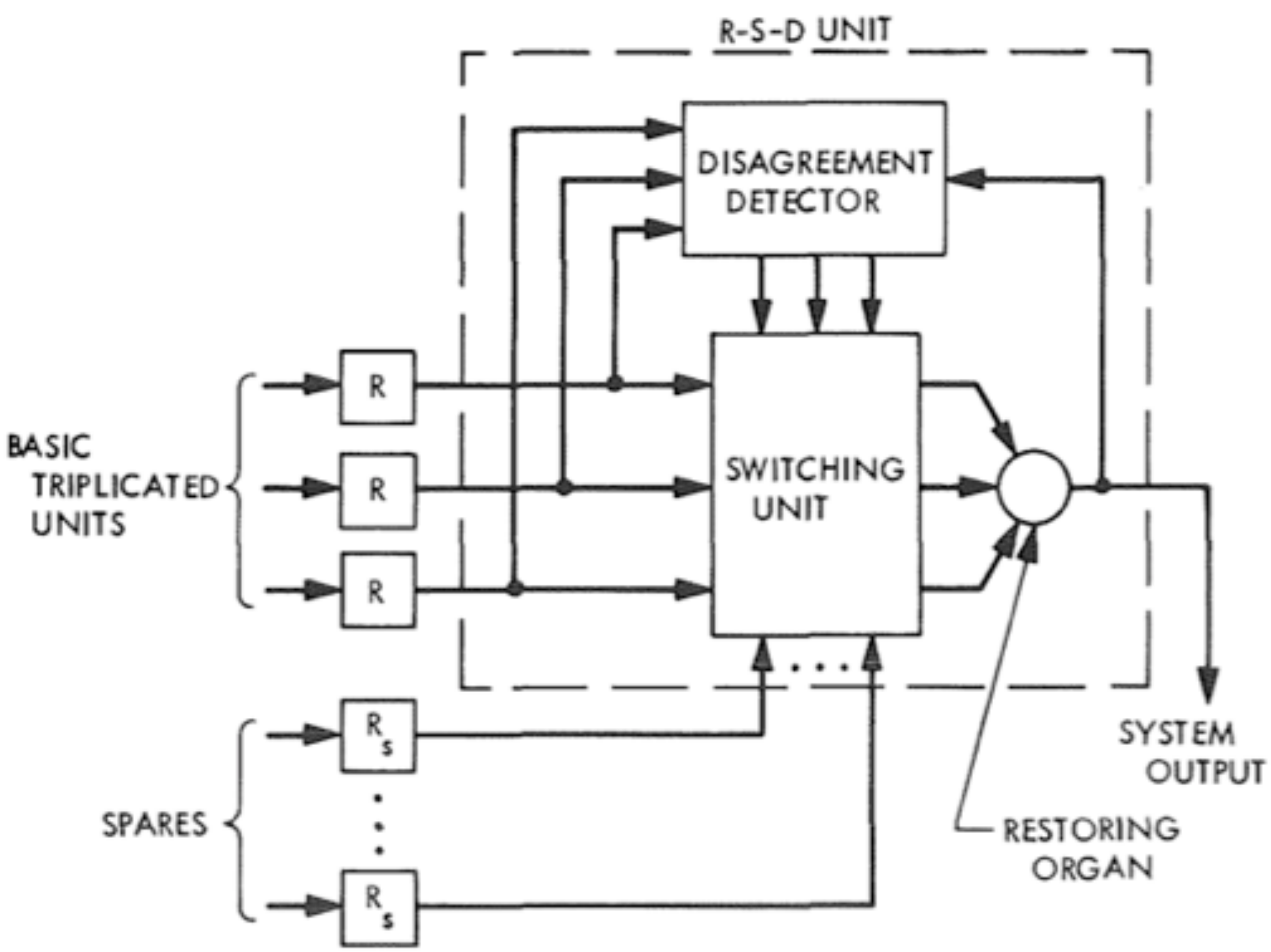
- **N-modular redundancy with spares**
  - Also called **hybrid redundancy**
  - System has basic NMR configuration
  - Disagreement detector replaces modules with spares if their output is not matching the voting result



  - Reliability as long as the spare pool is not exhausted
  - Improves fault masking capability of NMR
    - Can **tolerate two faults with one spare**, while classic NMR would need 5 modules with majority voting to tolerate two faults

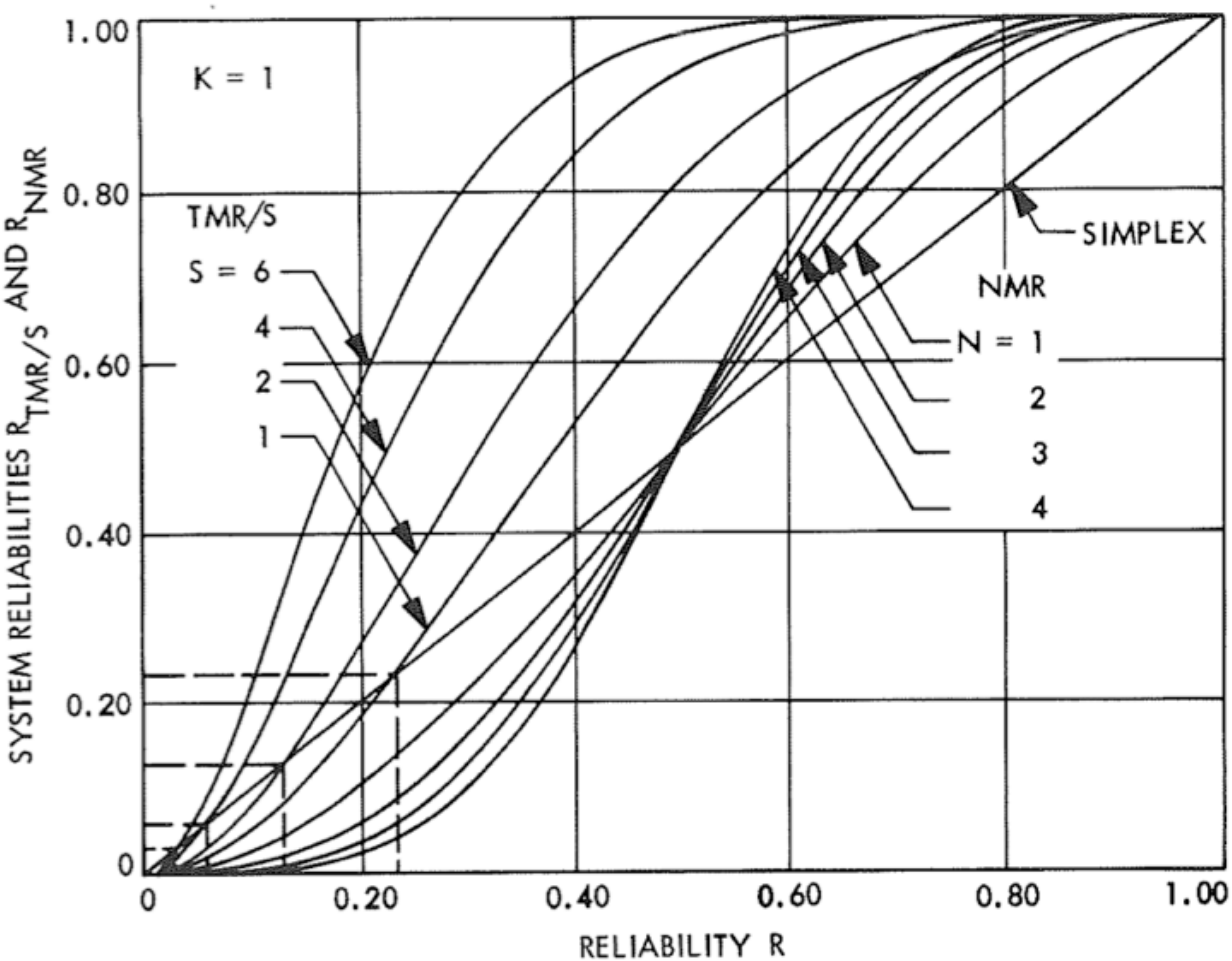
TMR with Spares

- Basic reliability computation based on the assumption of similar module failure rates in spares and non-spares
  - At least any two of all S+3 modules must survive



NASA Technical Report 32-1467, 1969

Comparison TMR vs. TMR/S vs. NMR

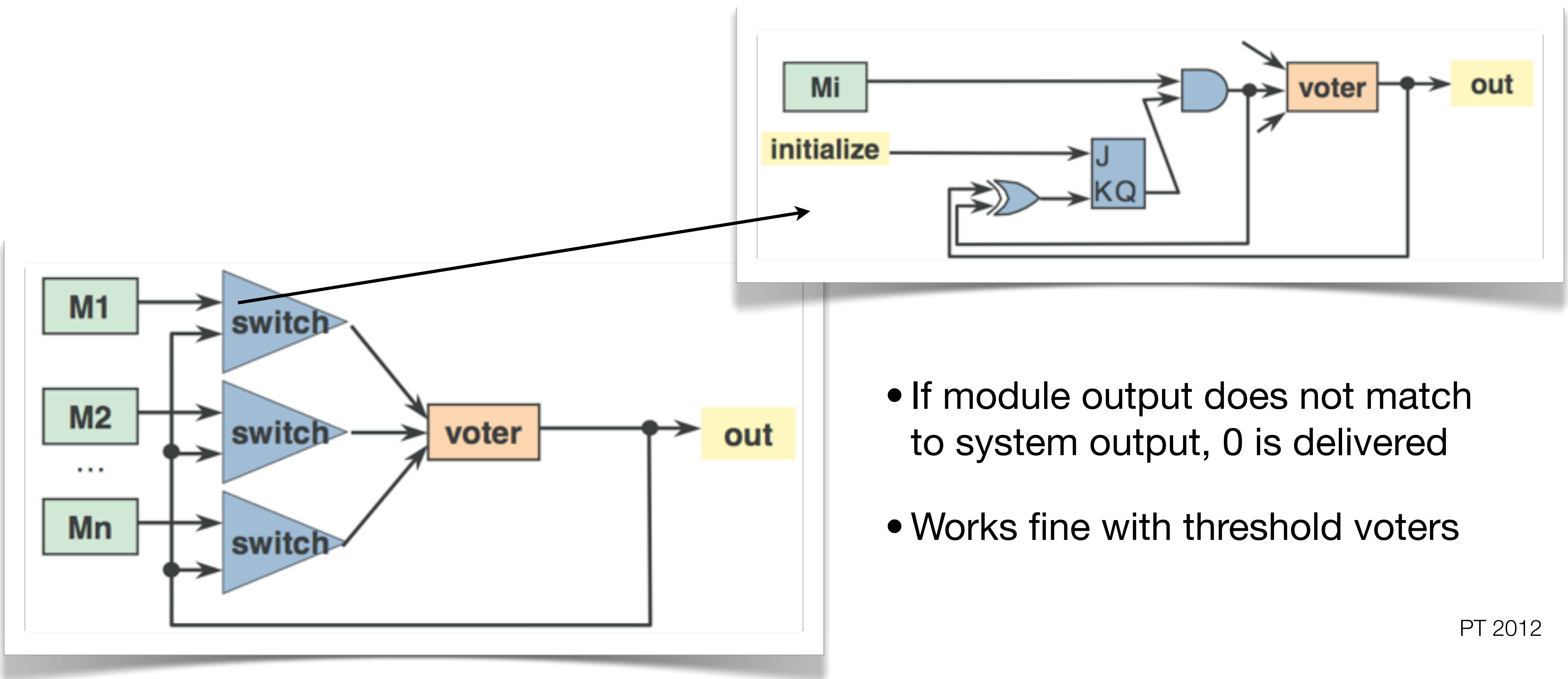


$\#Units = 2N + 1$

NASA Technical Report 32-1467, 1969

Hybrid Approaches

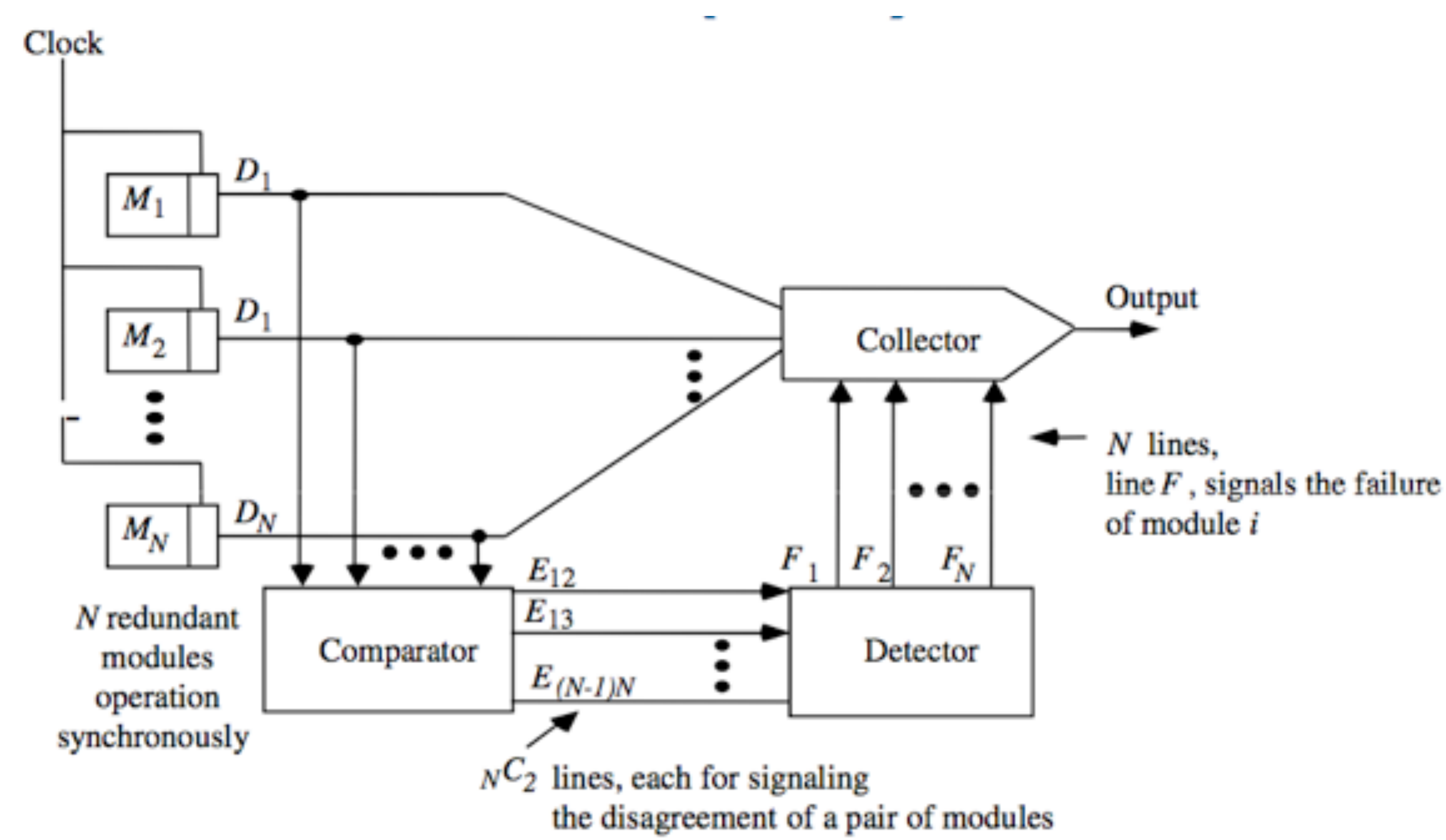
- **Self-purging redundancy**
    - Active redundant modules, each can remove itself from the system if faulty
    - Basic idea: Test for agreement with the voting result, otherwise 0



- If module output does not match to system output, 0 is delivered
- Works fine with threshold voters

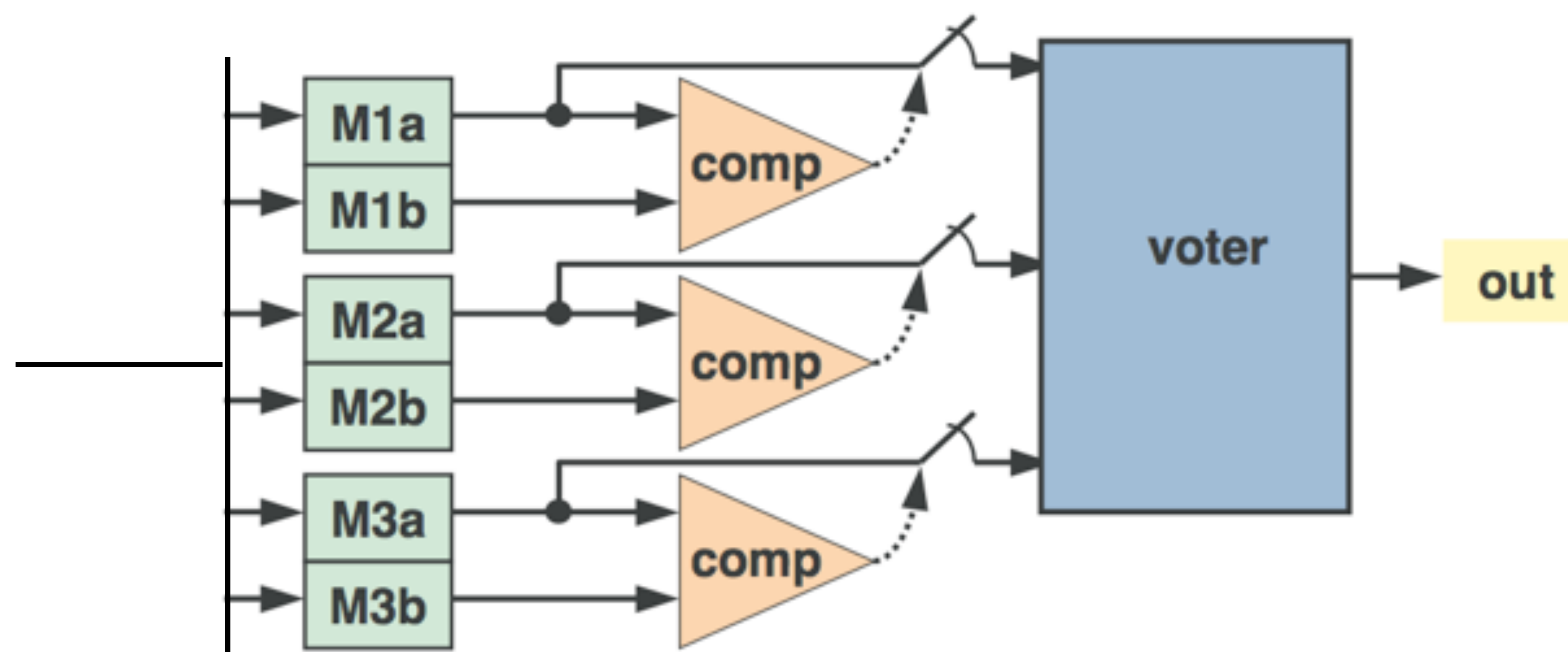
Hybrid Approaches

- **Sift-out modular redundancy** (N-2), no voter required
  - Pair-wise comparison of module outputs by **comparator**
    - N inputs and N-over-2 outputs
    - **Detector** uses these signals to identify the faulty module, includes also memory cells for failed modules
    - **Collector** sifts out the faulty input, based on information from detector



Hybrid Approaches

- Triple Duplex Architecture
  - TMR with duplex modules, used in the Shinkansen (Japanese train)
  - Fault masking with comparator, no more contribution to voting from faulty one
  - Allows tolerating another fault in the further operation, since comparator localizes again the faulty module
  - Adds again fault location capability to redundancy scheme
  - Supports also hot plugging of deactivated components



The Real World of Hardware Redundancy - Replacement Frequencies [Schroeder 2007]

760 node cluster, 2300 disks

HPC1	
Component	%
Hard drive	**30.6**
Memory	28.5
Misc/Unk	14.4
CPU	12.4
PCI motherboard	4.9
Controller	2.9
QSW	1.7
Power supply	1.6
MLB	1.0
SCSI BP	0.3

ISP, multiple sites, 26700 disks

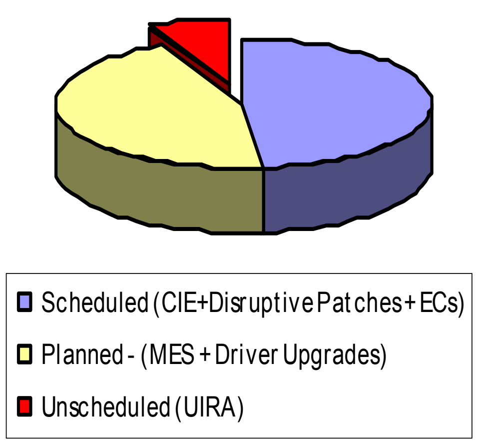
COM1	
Component	%
Power supply	34.8
Memory	20.1
Hard drive	**18.1**
Case	11.4
Fan	8.0
CPU	2.0
SCSI Board	0.6
NIC Card	1.2
LV Power Board	0.6
CPU heatsink	0.6

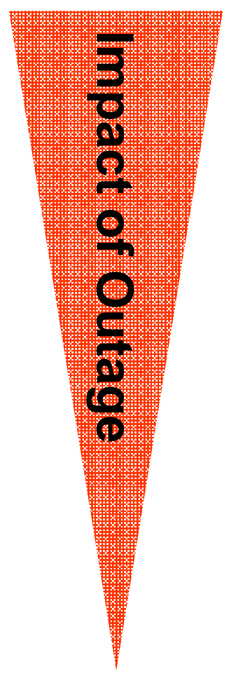
ISP, multiple sites, 9200 machines, 39000 disks

COM2	
Component	%
Hard drive	**49.1**
Motherboard	23.4
Power supply	10.1
RAID card	4.1
Memory	3.4
SCSI cable	2.2
Fan	2.2
CPU	2.2
CD-ROM	0.6
Raid Controller	0.6

IBM System z

Sources of Outages
Pre z9
-Hrs/Year/Syst-





	Prior Servers	z9 EC	Z10 EC	z196
Unscheduled Outages	✓	✓	✓	✓
Scheduled Outages	✓	✓	✓	✓
Planned Outages		✓	✓	✓
Preplanning requirements			✓	✓
Power & Thermal Management				

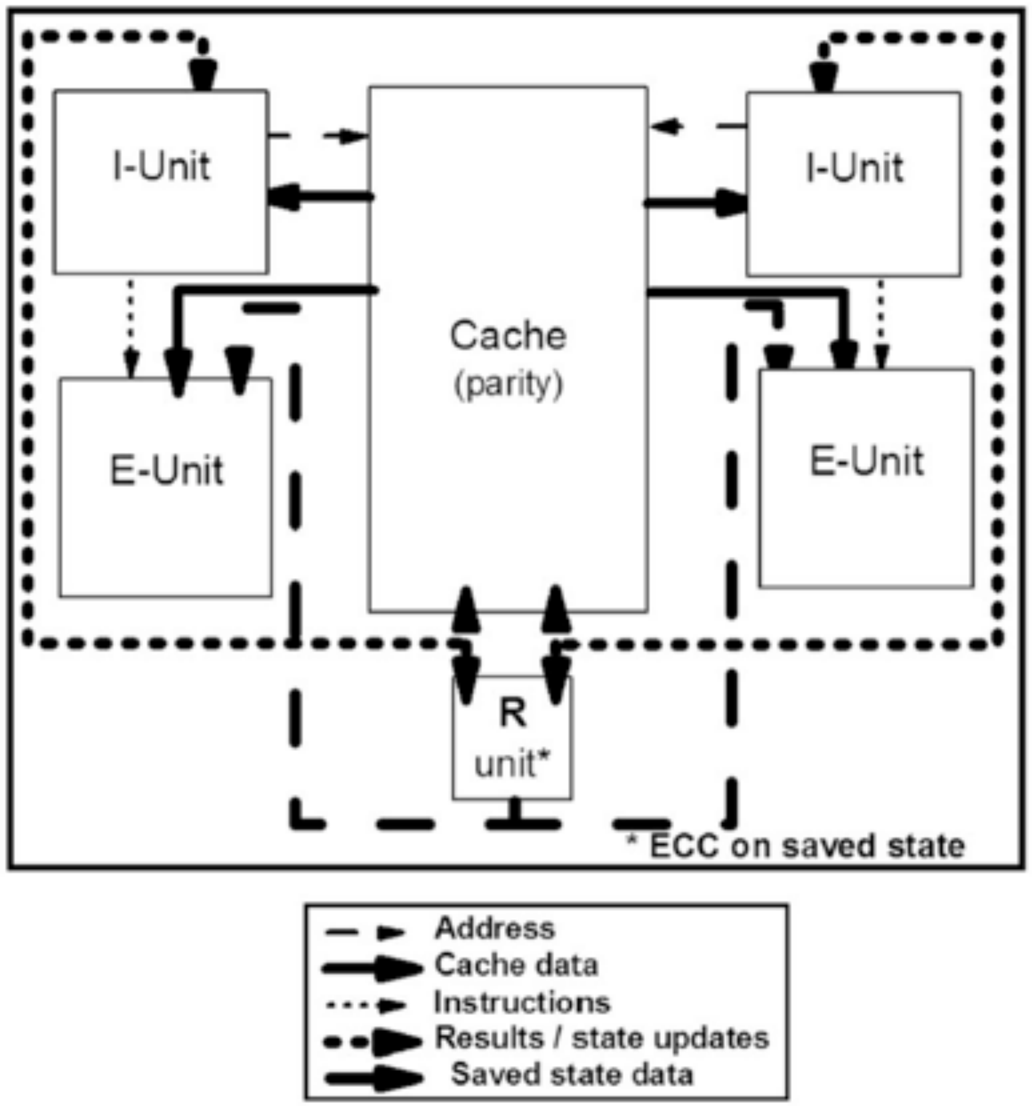
Increased Focus over time

Temperature = Silicon Reliability Worst Enemy
Wearout = Mechanical Components Reliability Worst Enemy.

IBM System z

- Machine-Check-Handling mechanism in z/Series
  - Equipment malfunction detection
  - Permit automatic recovery
  - Error states are reported by machine-check interruption
- Data error detection through information redundancy
- Recovery from machine-detected error states
  - Error checking and correction - use **circuitry redundancy**
  - CPU retry - **checkpoint** written at instruction-based synchronization points
  - Channel-subsystem recovery - **restart** of I/O components
  - Unit deletion - automated **degradation** of malfunctioning units

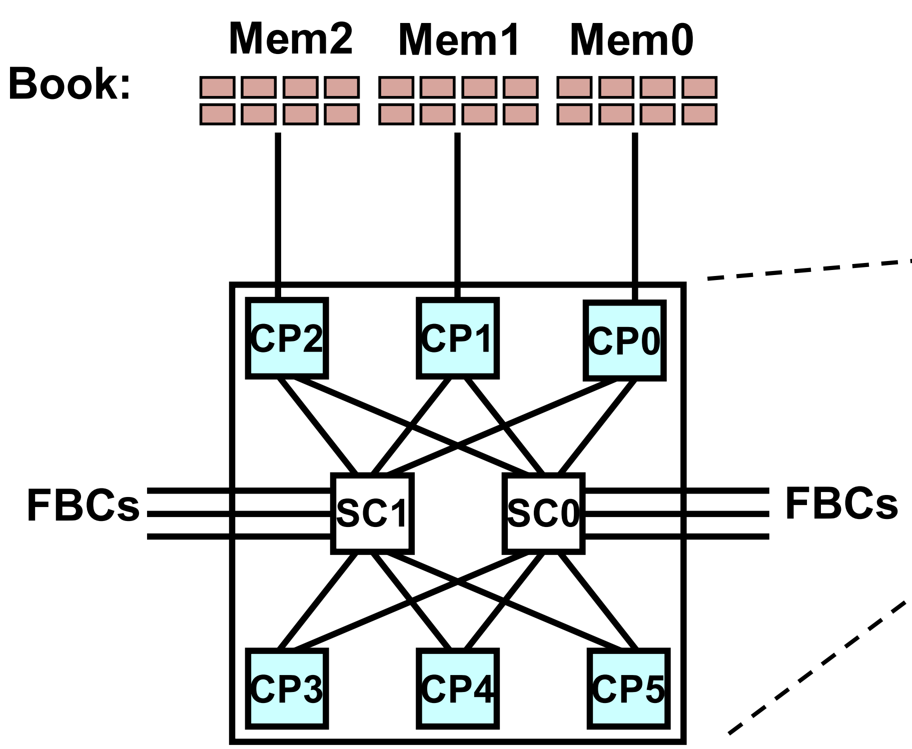
zEnterprise: Processor



- Instruction fetch and execution units are replicated
- Error check at the end of the pipeline
- R-unit keeps CPU registers and processor checkpoint
- E-units have shadow copy of registers for speed improvement
- All register / cache writes are compared, instruction retry in case
  - On fault, overwrite with R unit state
- Since z6, reverted to non-lockstepping and more fault sensors

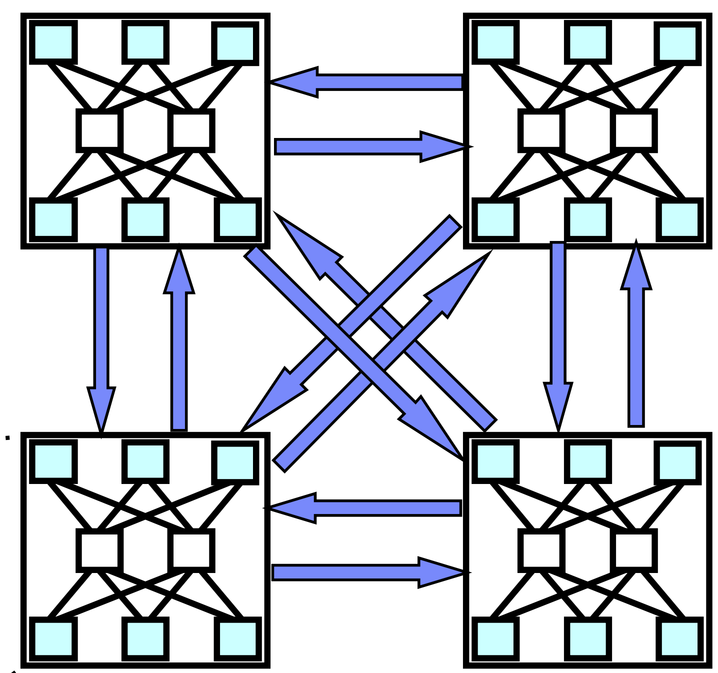
IBM System z - Processor Books

z196 Cache / Book Topology



Multi-Chip Module (MCM)

Fully connected 4 Book system:

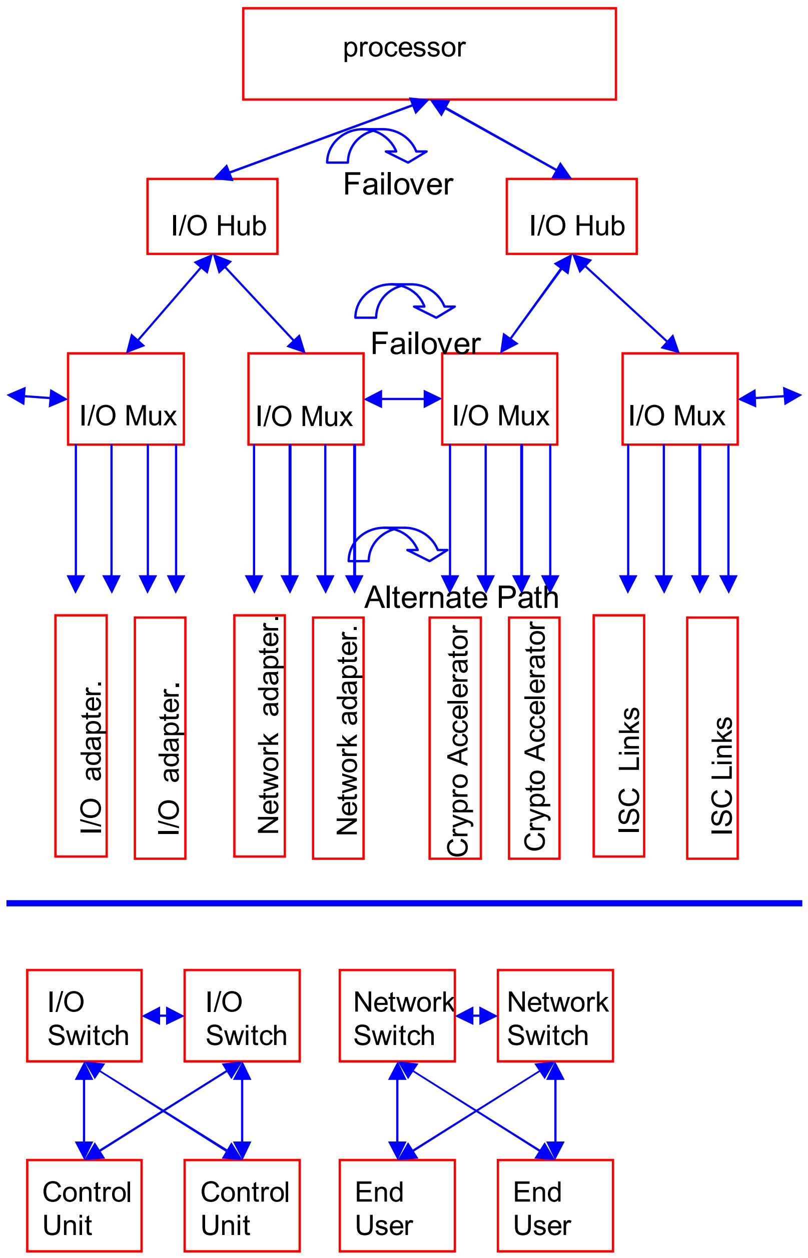


- **96 total cores**
- **Total system cache**
  - 768 MB shared L4 (eDRAM)
  - 576 MB L3 (eDRAM)
  - 144 MB L2 private (SRAM)
  - 19.5 MB L1 private (SRAM)

- 4 PU's (cores) per CP + co-processors
- SC: Storage Control
  - 96-192 MB L4 cache per SC, accessible from other MCMs
- FBC: Fabric book connectivity
- Support for dynamic book addition and repair
- 2+1 redundancy for book power supply

IBM System z

z196 RAS Design of
……..Fully Redundant I/O Subsystem – of existing IO cage and drawers

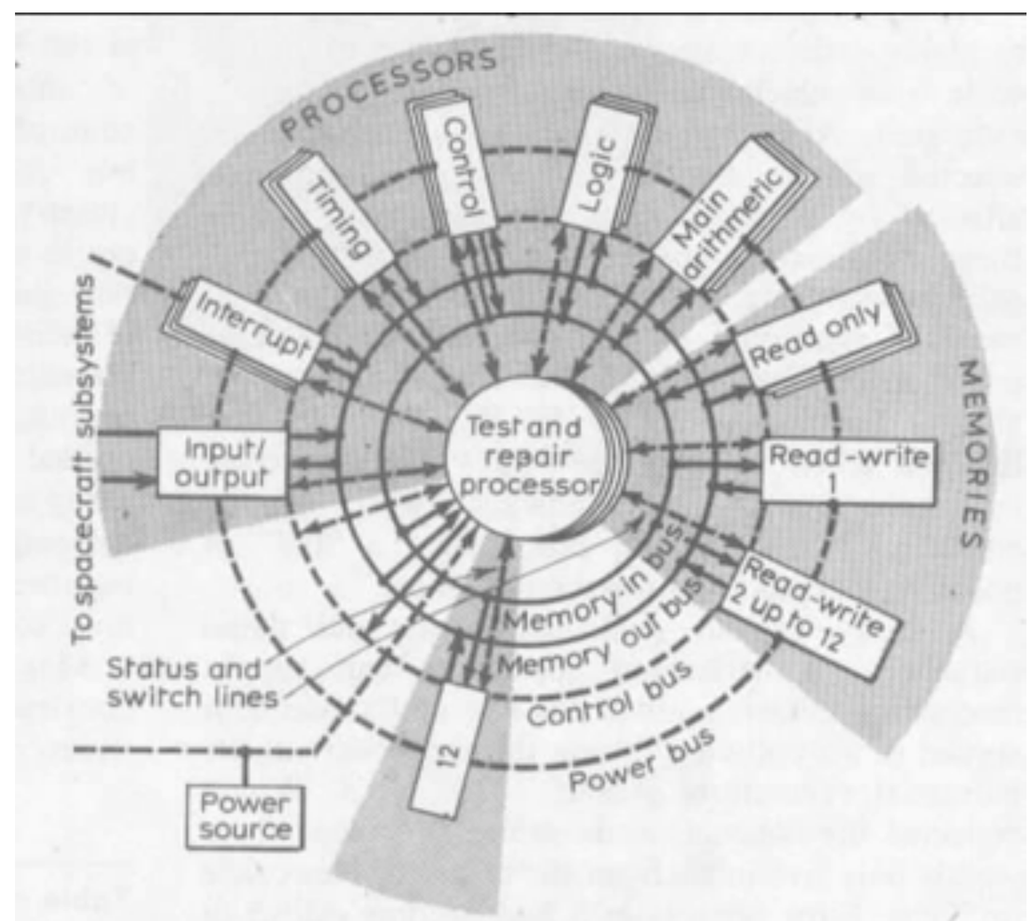


Fully Redundant I/O Design

- SAP / CP sparing
- SAP Reassignment
- I/O Reset & Failover
- I/O Mux Reset / Failover
- Redundant I/O Adapter
- Redundant I/O interconnect
- Redundant Network Adapters
- Redundant ISC links
- Redundant Crypto processors
- I/O Switched Fabric
- Network Switched/Router Fabric
- High Availability Plugging Rules
- I/O and coupling fanout rebalancing on CBA
- Channel Initiated Retry
- High Data Integrity Infrastructure
- I/O Alternate Path
- Network Alternate Path
- Virtualization Technology

Memory Redundancy

- Redundancy of memory data for masking
- Replication / coding at different levels
- Examples
  - STAR (Self-testing and self-repairing computer, for early spacecrafts), 1971
  - COMTRAC (Computer-aided traffic control system for Shinkansen train system)
  - Stratus (Commercial fault-tolerant system) http://www.stratus.com/uptime/
  - 3B20 by AT & T (Commercial fault-tolerant system)
  - Most modern memory controllers in servers



Memory Redundancy

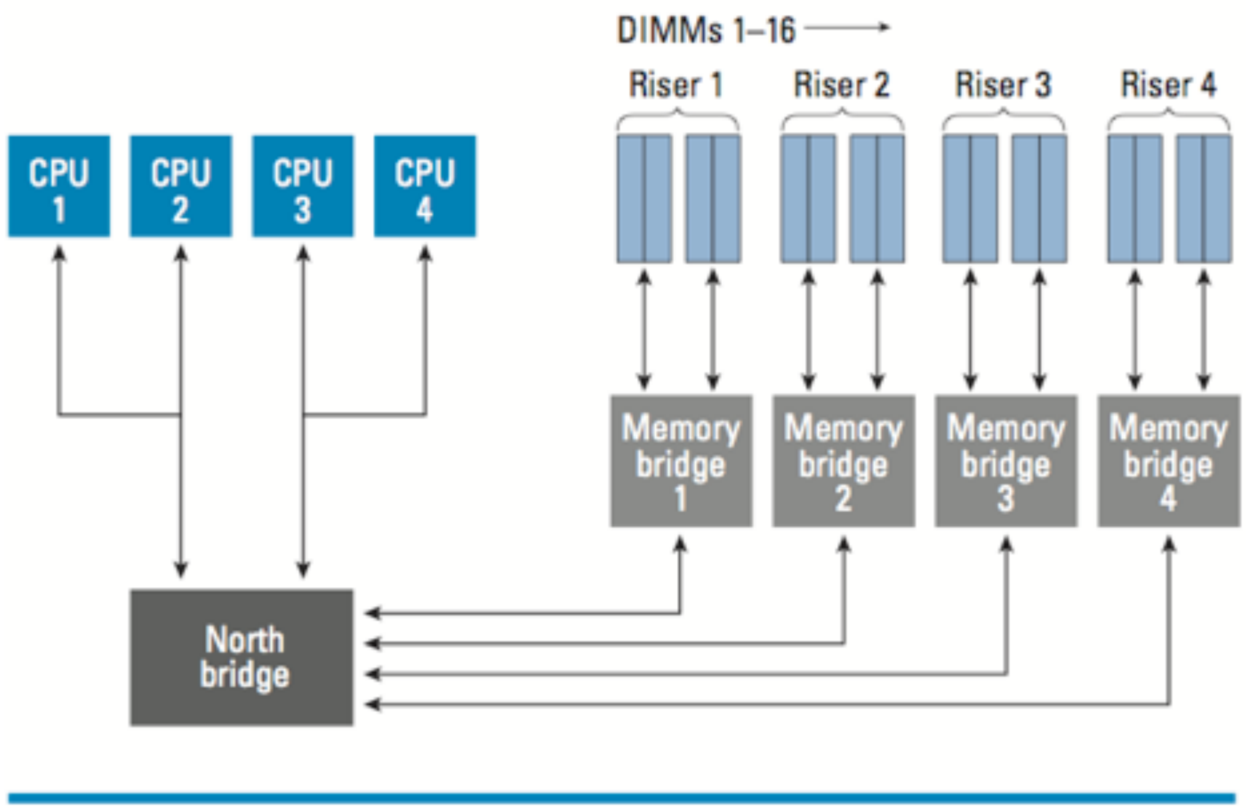
- Standard technology in DRAMs
  - Bit-per-byte **parity**, check on read access
  - Implemented by additional parity memory chip
  - **ECC** with Hamming codes - 7 check bits for 32 bit data words, 8 bit for 64 bit
    - Leads to 72 bit data bus between DIMM and chipset
    - Computed by memory controller on write, checked on read
    - Study by IBM: ECC memory achieves R=0.91 over three years
- Hewlett Packard **Advanced ECC** (1996)
  - Can detect and correct single bit and double bit errors

Memory Redundancy

- **IBM ChipKill**
  - Originally developed for NASA Pathfinder project, now in X-Series
  - Corrects up to 4 bit errors, detects up to 8 bit errors
  - Implemented in chipset and firmware, works with standard ECC modules
  - Based on striping approach with parity checks (similar to RAID)
  - 72 bit data word is split in 18 bit chunks, distributed on 4 DIMM modules
  - 18 DRAM chips per module, one bit per chip
- **HP Hot Plug RAID Memory**
  - Five memory banks, cache line is striped, fifth bank for parity information
  - Corrects single bit, double bit, 4-bit, 8-bit errors; hot plugging support

Memory Redundancy

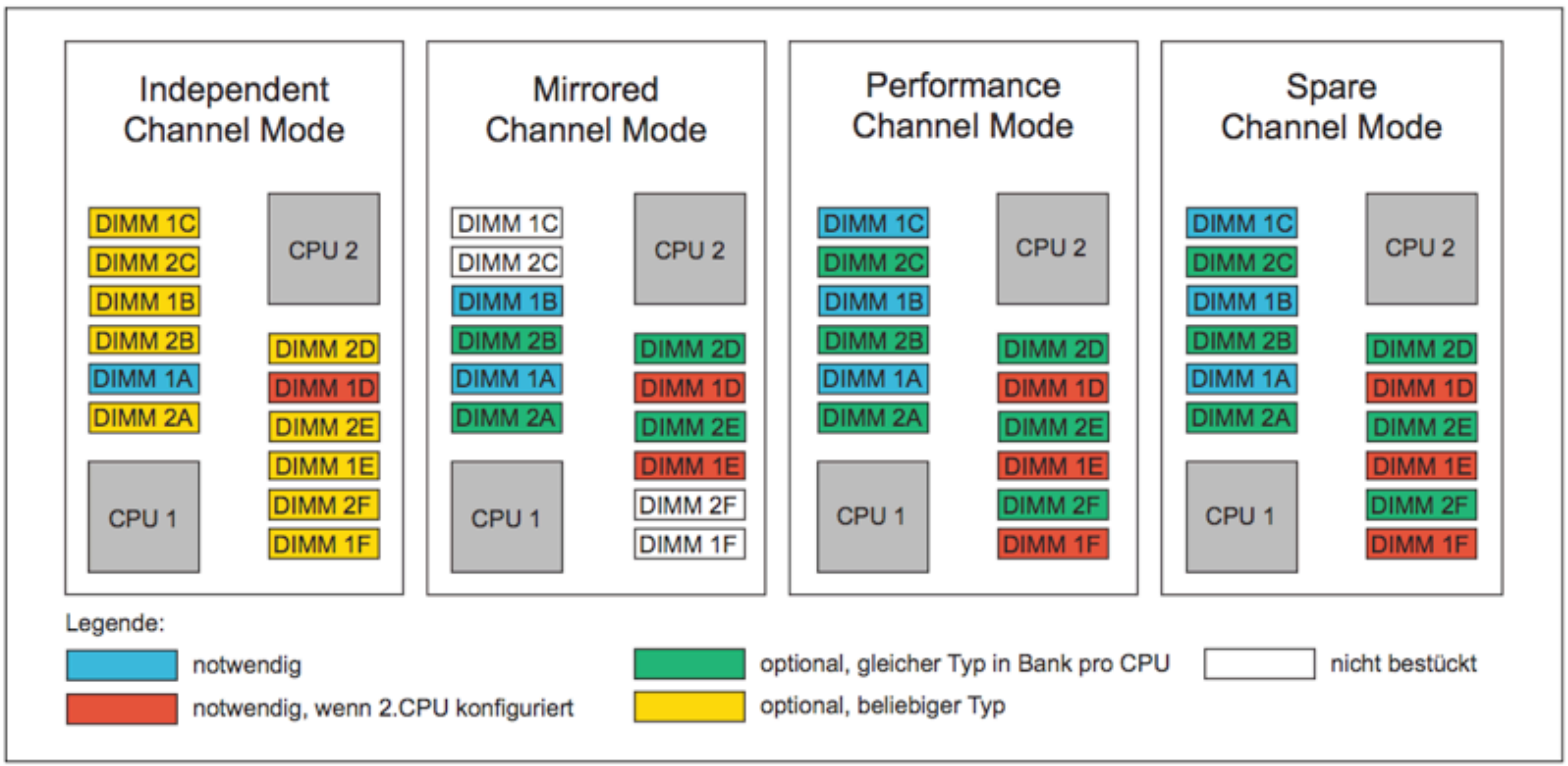
- Dell PowerEdge Servers, 2005 (taken from www.dell.com)



BIOS options	Sparing	Mirroring	RAID	Hot addition	Hot replacement
Spare-bank memory	Support depends on memory card	Not supported	Not supported	Not supported	Not supported
Memory mirroring	Not supported	Supported if riser 1 and riser 2 have equal memory and/or riser 3 and riser 4 have equal memory (only memory mirroring is enabled)	Not supported	Not supported	Supported
Memory RAID	Not supported	Not supported	Supported if all four risers have equal memory (only memory RAID is enabled)	Not supported	Supported
Redundancy Disabled	Not supported	Not supported	Not supported	Hot addition in previously empty slots is supported	Not supported

Memory Redundancy

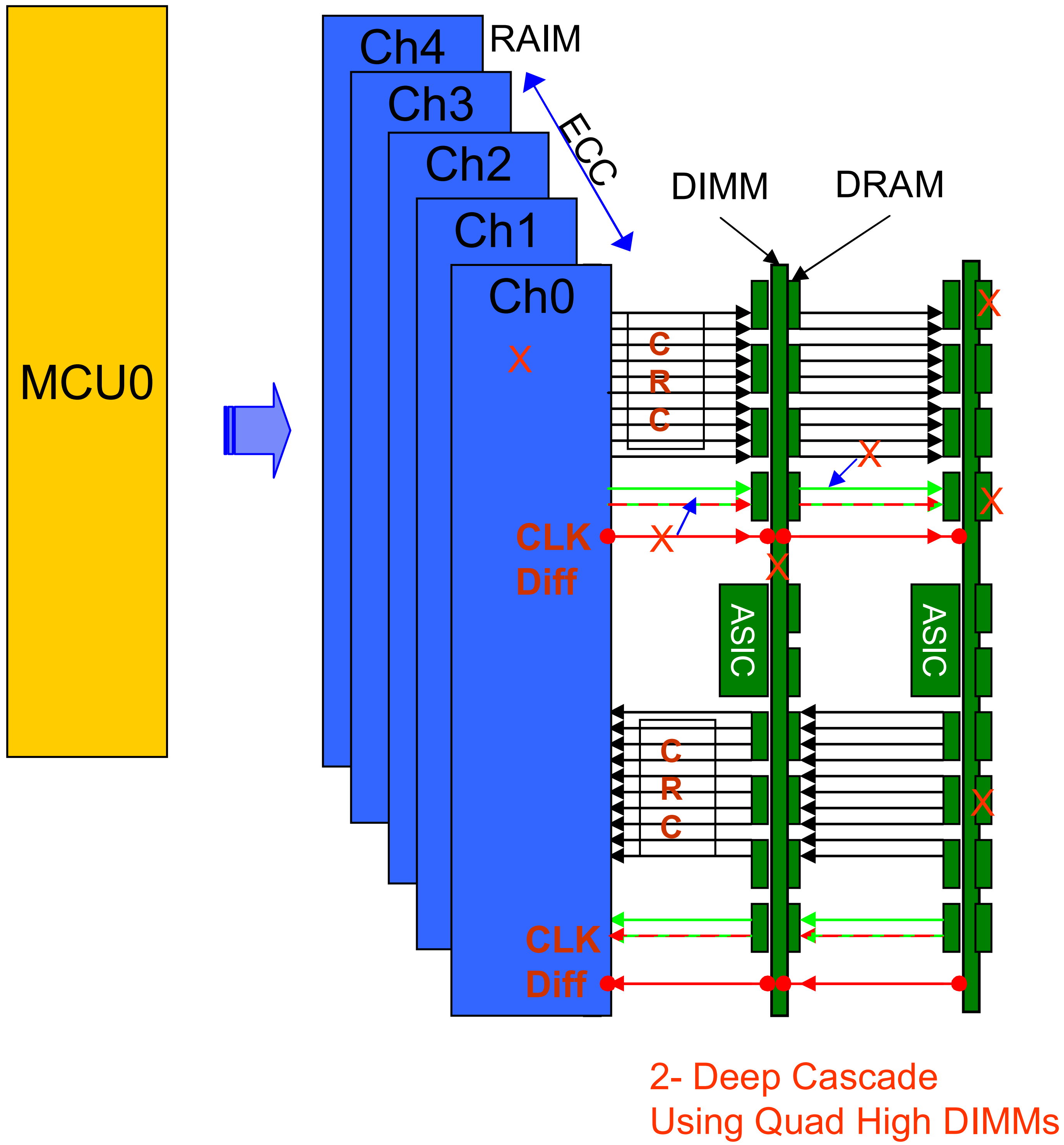
- Fujitsu System Board D2786 for RX200 S5 (2010)
- Independent Channel Mode: Standard operational module, always use first slot
- Mirrored Channel Mode: Identical modules on slot A/B (CPU1) and D/E (CPU2)



IBM System z - Memory RAID

- System z10 EC memory design
  - Four Memory Controllers (MCUs) organized in two pairs, each MCU with four redundant channels
  - 16 to 48 DIMMs per book, plugged in groups of 8
  - 8 DIMMs (4 or 8 GB) per feature, 32 or 64 GB physical memory per feature
  - 64 to 384 GB physical memory per book = 64 to 384 GB for use (HSA and customer)
- z196 memory design:
  - Three MCUs, each with five channels. The fifth channel in each z196 MCU is required to implement *Redundant Array of Independent Memory (RAIM)*
    - Detected and corrected: Bit, lane, DRAM, DIMM, socket, and complete memory channel failures, including many types of multiple failures

IBM System z196 - Memory RAID



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

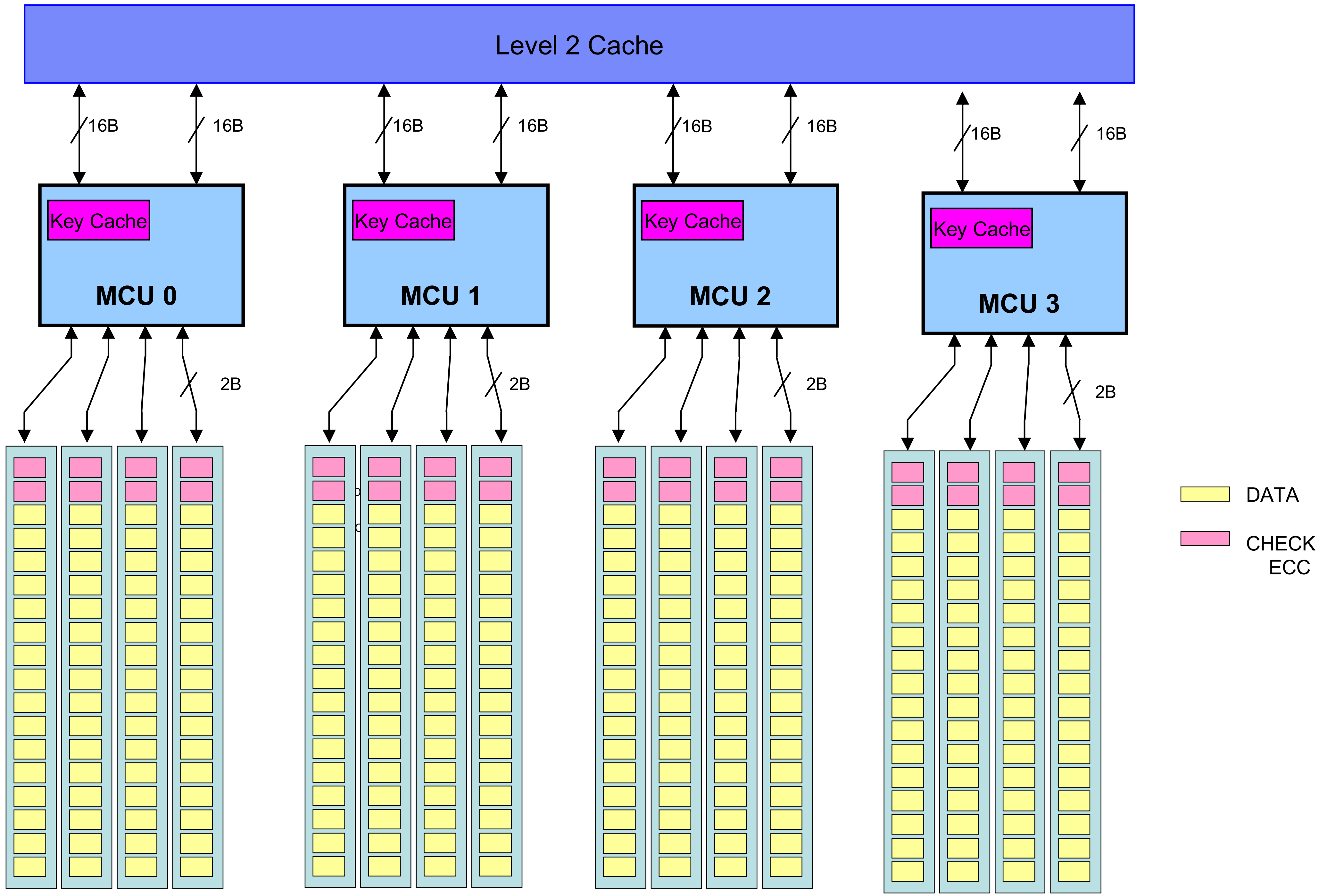
Lane Failure
- CRC with Retry
- Data – lane sparing
- CLK – RAIM with lane sparing

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

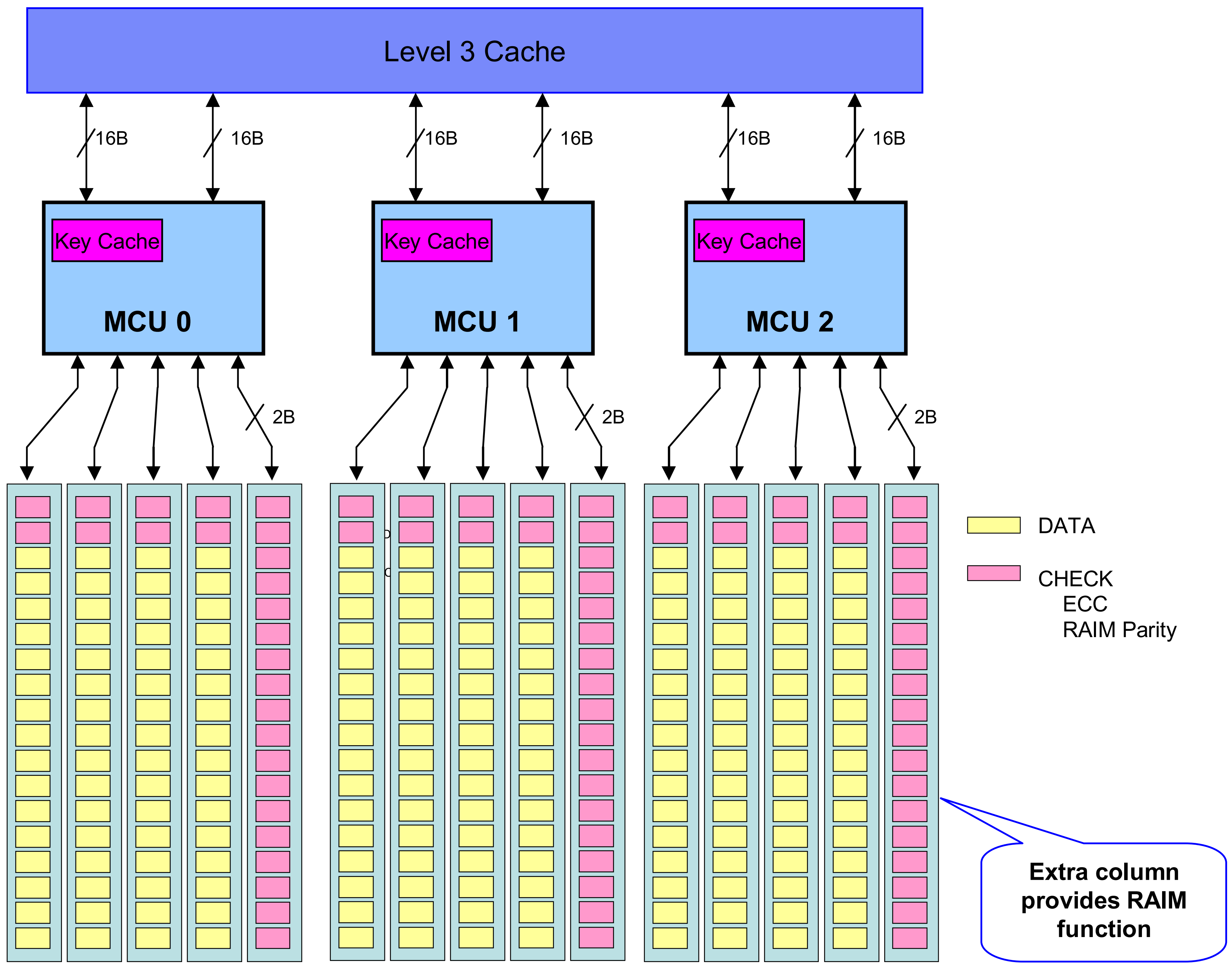
DIMM Controller ASIC Failure
- RAIM Recovery

Channel Failure
- RAIM Recovery

IBM z10 EC Memory Structure



IBM System z196 - RAIM



Disk Redundancy

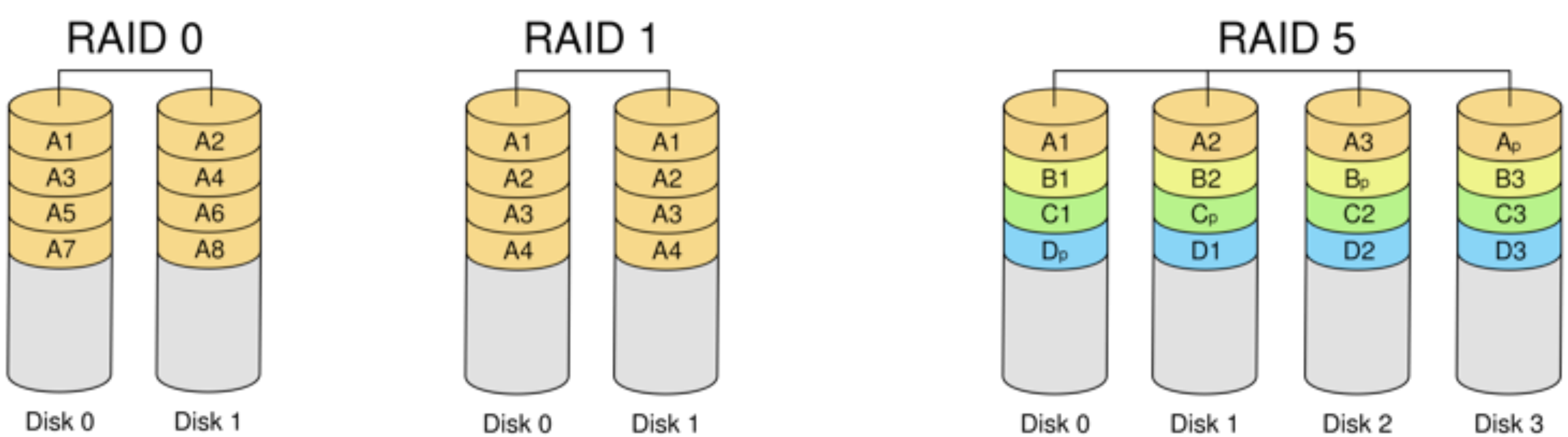
- Typical measure is the *annual failure rate (AFR)* - average number of failures / year

$$AFR = \frac{1}{MTBF_{years}} = \frac{8760}{MTBF_{hours}}$$

  - Can be interpreted as failure probability during a year
  - MTBF = Mean time **before** failure, here
- Disk MTTF: On average, one failure takes place in the given disk hours
- Example: Seagate Barracuda ST3500320AS: MTTF=750000h=85.6 years
  - With thousand disks, on average every 750h (a month) some disk fails
  - Measured by the manufacturer under heavy load and physical stress
  - AFR=0.012

RAID

- **Redundant Array of Independent Disks (RAID)** [Patterson et al. 1988]
  - Improve I/O performance and / or reliability by building *raid groups*
  - Replication for information reconstruction on disk failure (*degrading*)
  - Requires computational effort (dedicated controller vs. software)
  - Assumes failure independence



RAID Reliability Comparison

- Treat disk failing as Bernoulli experiment - independent events, identical probability
- Probability for k events of probability p in n runs

$$B_{n,p}(k) = p^k (1-p)^{n-k} \binom{n}{k}$$

  - Probability for a failure of a RAID 1 mirror - all disks unavailable:

$$p_{allfail} = \binom{n}{n} {p_{fail}}^n (1 - p_{fail})^0 = {p_{fail}}^n$$

  - Probability for a failure of a RAID 0 strip set - any faults disk leads to failure:

$$\begin{aligned} p_{anyfail} &= 1 - p_{allwork} \\ &= 1 - \binom{n}{n} (1 - p_{fail})^n {p_{fail}}^0 \\ &= 1 - (1 - p_{fail})^n \end{aligned}$$

RAID MTTF Calculation [Patterson]

- Works for RAID levels were second outage during repair is fatal
- Core idea is that groups of data disks are protected by additional check disks
  - D - Total number of data disks
  - G - Number of data disks in a group (e.g. G=1 in RAID1)
  - C - Number of redundant check disks (parity / mirror) in a group (e.g. C=1 in RAID1)
  - n_G = D / G = number of groups, G+C : Number of disks in a group

$$MTTF_{Group} = \frac{MTTF_{Disk}}{G+C} \cdot \frac{1}{p_{SecondFailureDuringRepair}}$$

RAID MTTF Calculation [Patterson]

- Assuming exponential distribution, the probability for a second disk failure during the repair time can be determined by:

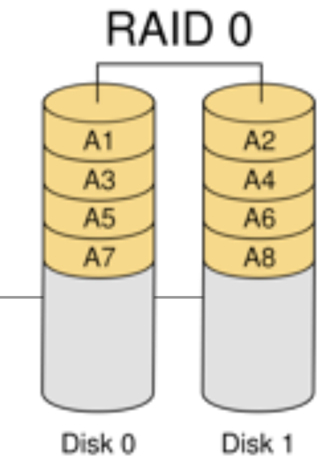
$$p_{SecondFailure} = \frac{MTTR}{\frac{MTTF_{Disk}}{G+C-1}}$$

- So:

$$MTTF_{Group} = \frac{MTTF_{Disk}}{G+C} \cdot \frac{1}{p_{SecondFailureDuringRepair}}$$

$$\begin{aligned} MTTF_{Raid} &= \frac{MTTF_{Group}}{n_G} \\ &= \frac{{MTTF_{Disk}}^2}{(G+C) * n_G * (G+C-1) * MTTR} \end{aligned}$$

(C) Wikipedia



RAID 0

- **Raid 0** - Block-level striping
  - I/O performance improvement with many channels and drives
    - One controller per drive
  - Optimal stripe size depends on I/O request size, random vs. sequential I/O, concurrent vs. single-threaded I/O
    - Fine-grained striping: Good load balancing, catastrophic data loss
    - Coarse-grained striping: Good recovery for small files, worser performance
    - One option: Strip size = Single-threaded I/O size / number of disks
  - Parallel read supported, but positioning overhead for small concurrent accesses
  - No fault tolerance

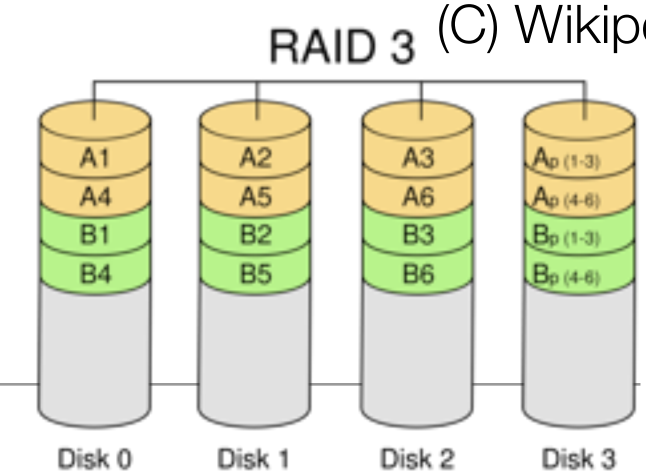
$$MTTF_{Raid0} = \frac{MTTF_{Disk}}{N}$$

(C) Wikipedia

RAID 1

- **Raid 1** - Mirroring and duplexing
  - Duplicated I/O requests
  - Decreasing write performance, up to double read rate of single disk
    - RAID controller might allow concurrent read and write per mirrored pair
  - Highest overhead of all solutions, smallest disk determines resulting size
  - Reliability is given by probability that one disk fails and the second fails while the first is repaired
  - With D=1, G=1, C=1 and the generic formula, we get

$$MTTF_{Raid1} = \frac{MTTF_{Disk}}{2} \cdot \frac{MTTF_{Disk}}{MTTR_{Disk}}$$



(C) Wikipedia

Raid 2/3

- **Raid 2** - Byte-level striping with Hamming code -based check disk
  - No commercial implementation due to ECC storage overhead
  - Online verification and correction during read
- **Raid 3** - Byte-level striping with dedicated XOR parity disk
  - All data disks used equally, one XOR parity disk as bottleneck (C=1)
  - Bad for concurrent small accesses, good sequential performance (streaming)
  - Separate code is needed to identify a faulty disk
  - Disk failure has only small impact on throughput
  - RAID failure if more than one disk fails:

$$MTTF_{Raid3} = \frac{MTTF_{Disk}}{D+C} \cdot \frac{\frac{MTTF_{Disk}}{D+C-1}}{MTTR_{Disk}}$$

Parity With XOR

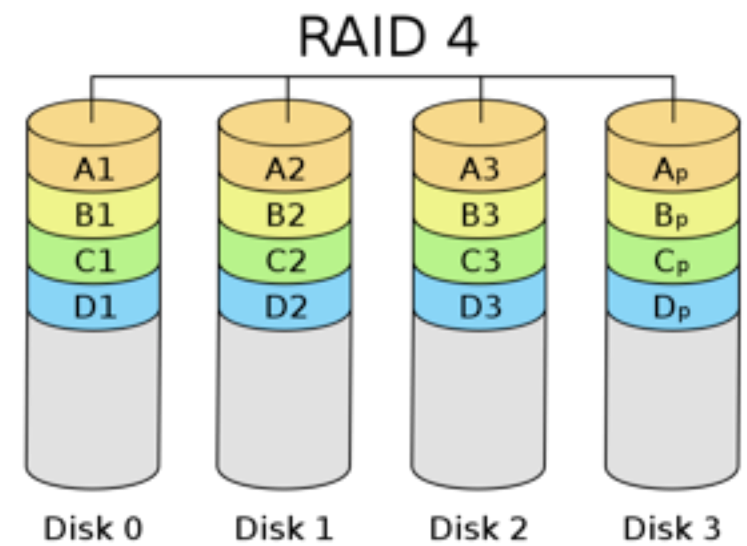
- Self-inverse operation
  - 101 XOR 011 = 110, 110 XOR 011 = 101

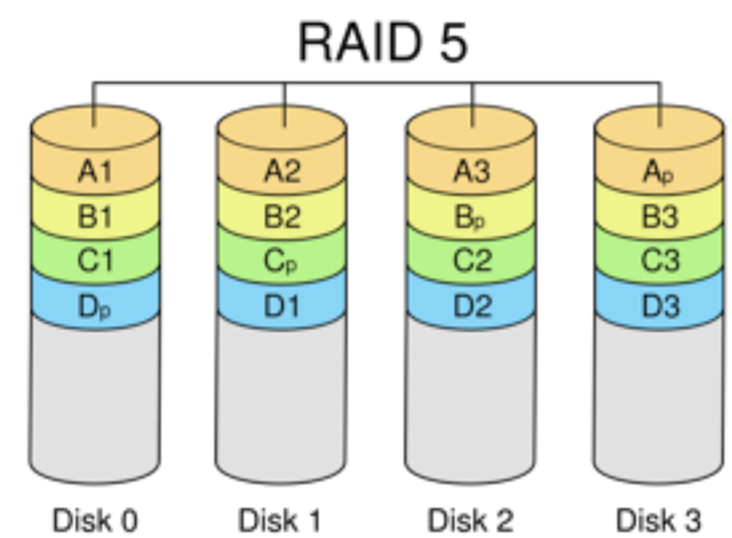
Disk	Byte							
1	1	1	0	0	1	0	0	1
2	0	1	1	0	1	1	1	0
3	0	0	0	1	0	0	1	1
4	1	1	1	0	1	0	1	1
Parity	0	1	0	1	1	1	1	1

Disk	Byte							
1	1	1	0	0	1	0	0	1
Parity	0	1	0	1	1	1	1	1
3	0	0	0	1	0	0	1	1
4	1	1	1	0	1	0	1	1
Hot Spare	0	1	1	0	1	1	1	0

RAID 4 / 5

- Raid 4 - Block-level striping with dedicated parity disk
  - RAID 3 vs. RAID 4: Allows concurrent block access
- Raid 5 - Block-level striping with distributed parity
  - Balanced load as with Raid 0, better reliability
  - Bad performance for small block writing
  - Most complex controller design, difficult rebuild
  - When block in a stripe is changed, old block and parity must be read to compute new parity
    - For every changed data bit, flip parity bit



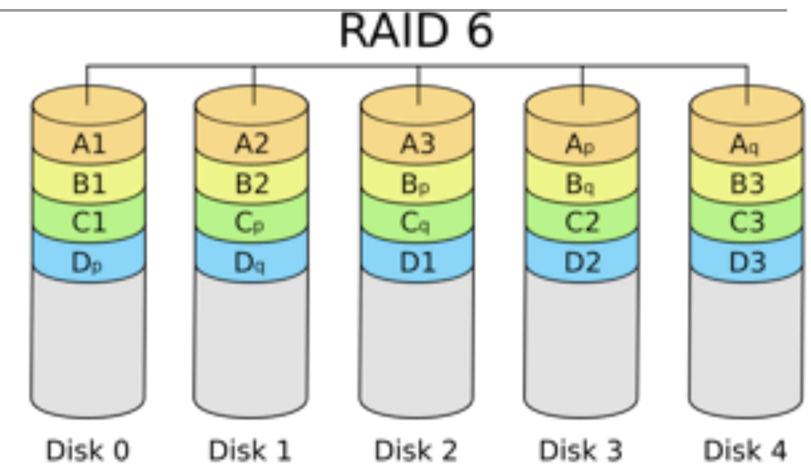


(C) Wikipedia

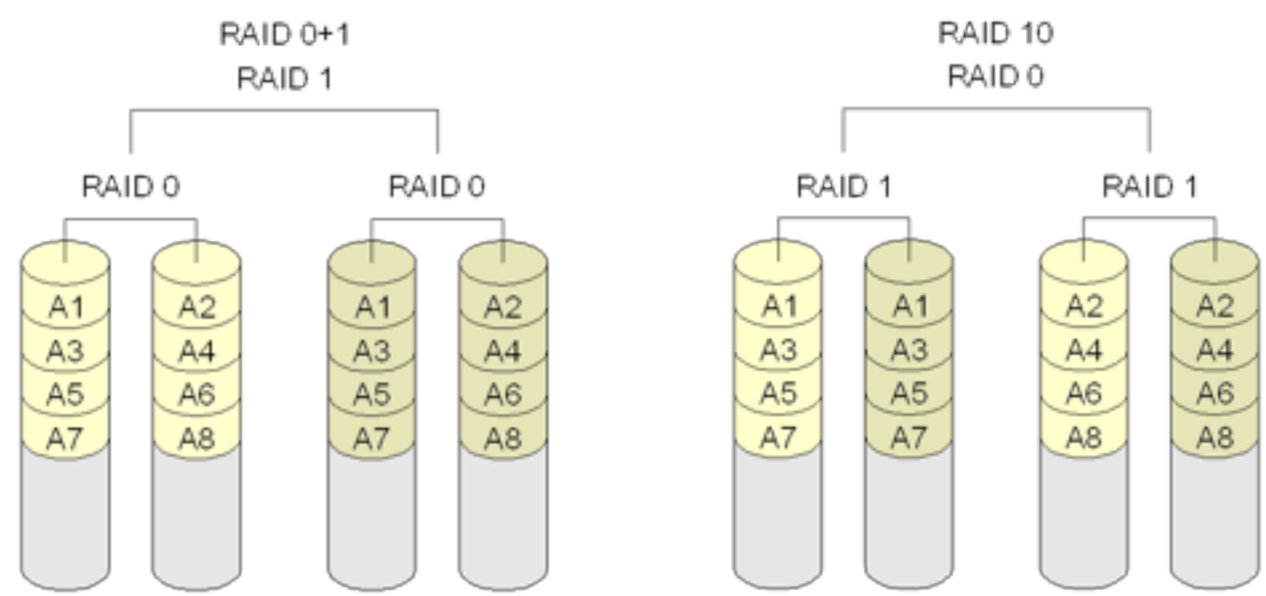
$$MTTF_{Raid5} = \frac{MTTF_{Disk}}{N} \cdot \frac{\frac{MTTF_{Disk}}{N-1}}{MTTR_{Disk}}$$

RAID 6 / 01 / 10

- Raid 6 - Block-level striping with two parity schemes
    - Extension of RAID5, can sustain multiple drive failures at the same time
    - High controller overhead to compute parities, poor write performance

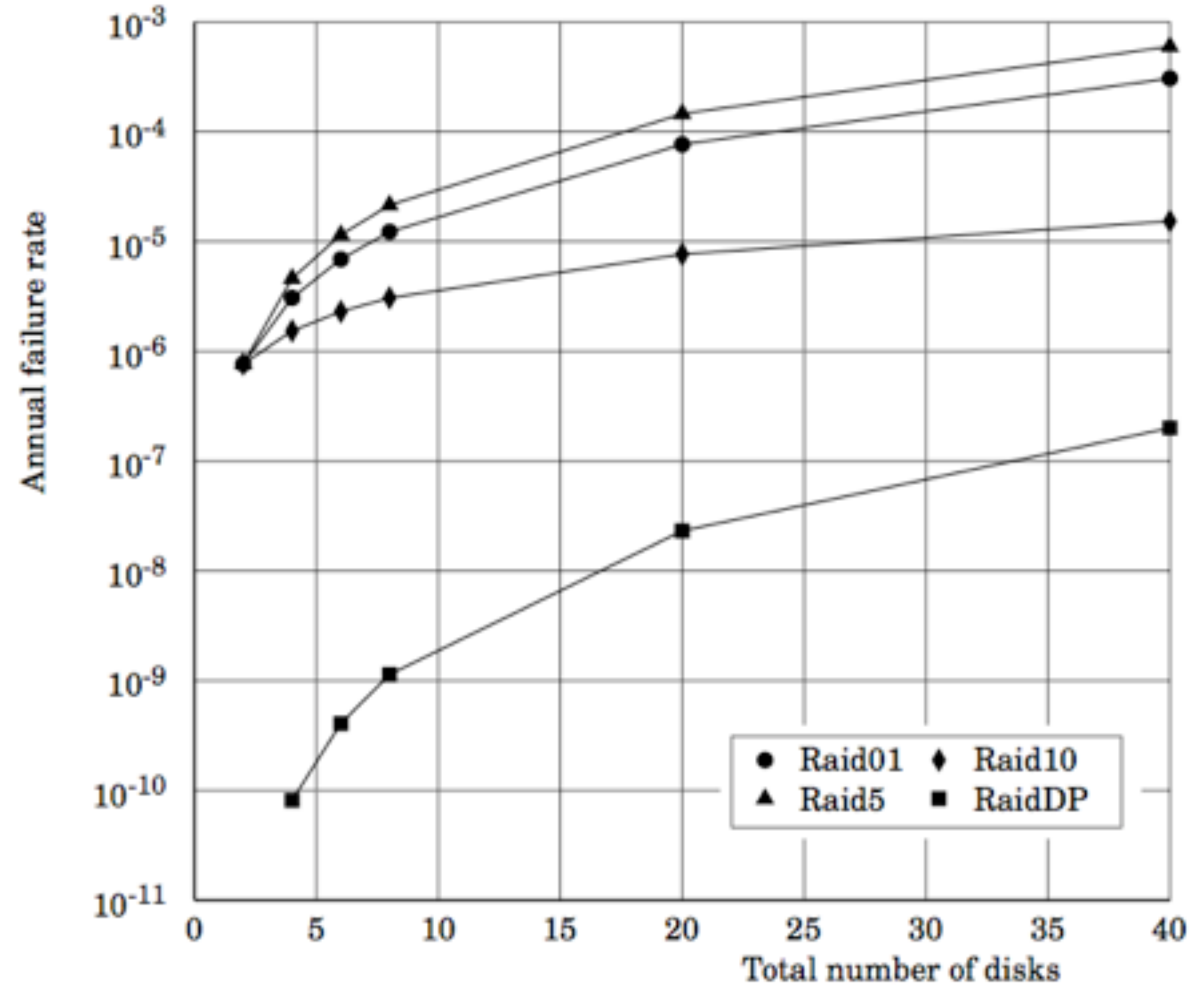


- Raid 01 - Every mirror is a Raid 0 stripe (min. 4 disks)
- Raid 10 - Every stripe is a Raid 1 mirror (min. 4 disks)
- RAID DP - RAID 4 with second parity disk
    - Additional parity includes first parity + all but one of the data blocks (diagonal)
    - Can deal with two disk outages



RAID Analysis (Schmidt)

- Take the same number of disks in different constellations
  - AFR_{Disk} = 0.029, MTTR=8h
- RAID5 has bad reliability, but offers most effective capacity
- In comparison to RAID5, RAID10 can deal with two disk errors
- Also needs to consider different resynchronization times
  - RAID10: Only one disk needs to be copied to the spare
  - RAID5 / RAID-DP: All disks must be read to compute parity
- Use RAID01 only in 2+2 combination



RAID Analysis (TecChannel.de)

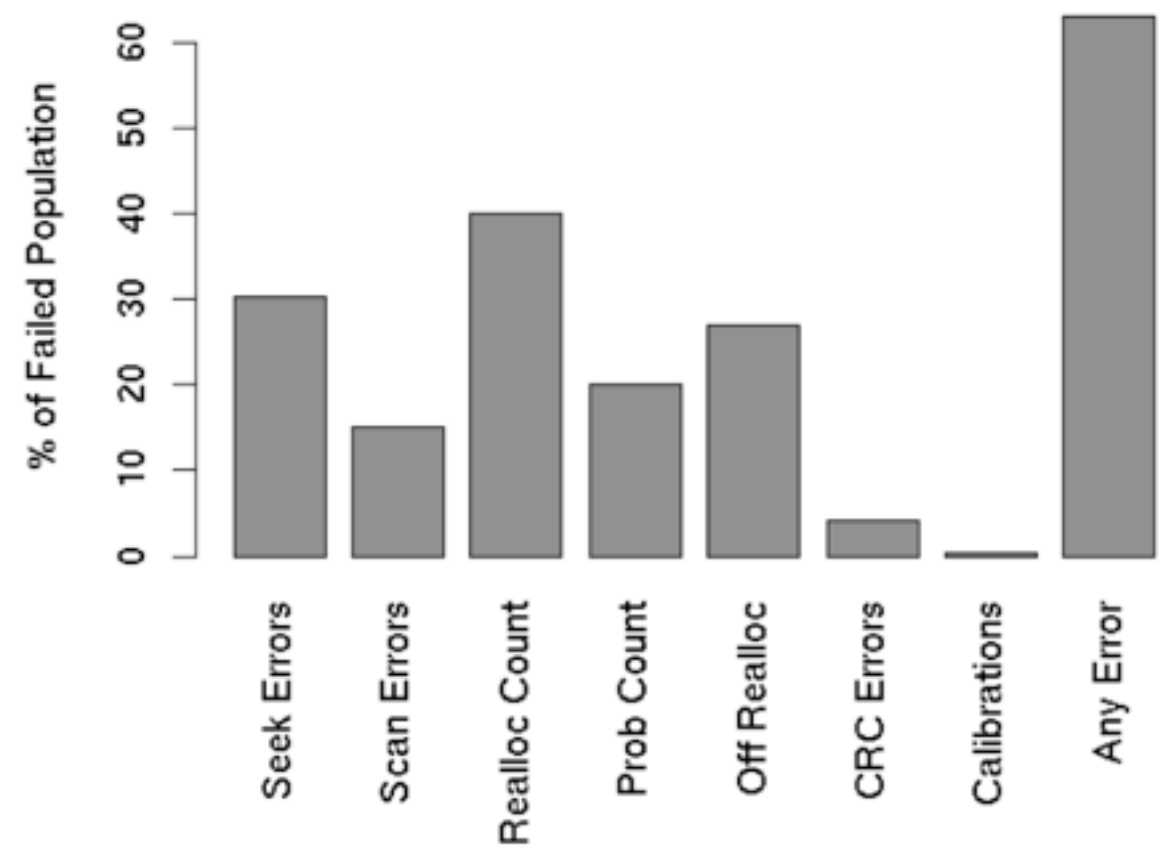
	RAID 0	RAID 1	RAID 10	RAID 3	RAID 4	RAID 5	RAID 6
Number of drives	n > 1	n = 2	n > 3	n > 2	n > 2	n > 2	n > 3
Capacity overhead (%)	0	50	50	100 / n	100 / n	100 / n	200 / n
Parallel reads	n	2	n / 2	n - 1	n - 1	n -1	n - 2
Parallel writes	n	1	1	1	1	n / 2	n / 3
Maximum read throughput	n	2	n / 2	n - 1	n - 1	n - 1	n - 2
Maximum write throughput	n	1	1	1	1	n / 2	n / 3

Software RAID

- Software layer above block-based device driver(s)
- Windows Desktop / Server, Mac OS X, Linux, ...
- Multiple problems
  - Computational overhead for RAID levels beside 0 and 1
  - Boot process
  - Legacy partition formats
- Driver-based RAID
  - Standard disk controller with special firmware
  - Controller covers boot stage, device driver takes over in protected mode

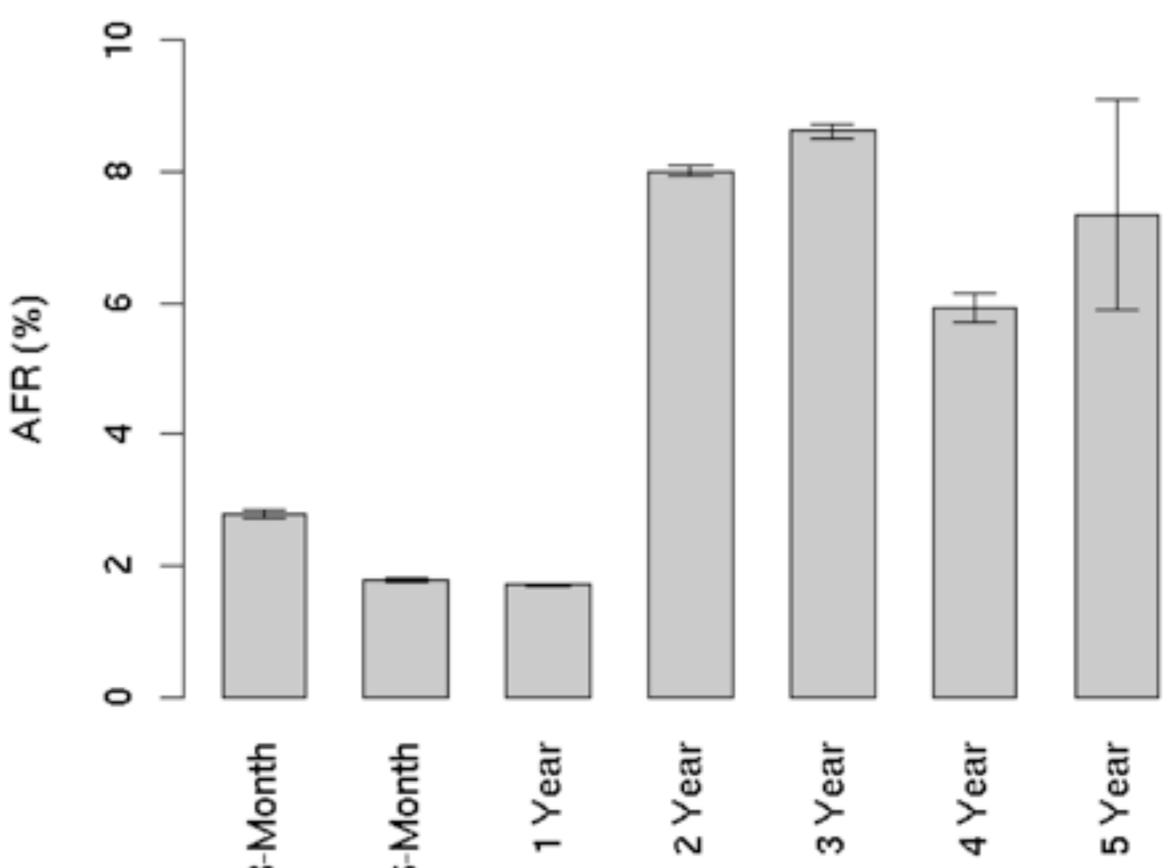
Disk Redundancy: Google

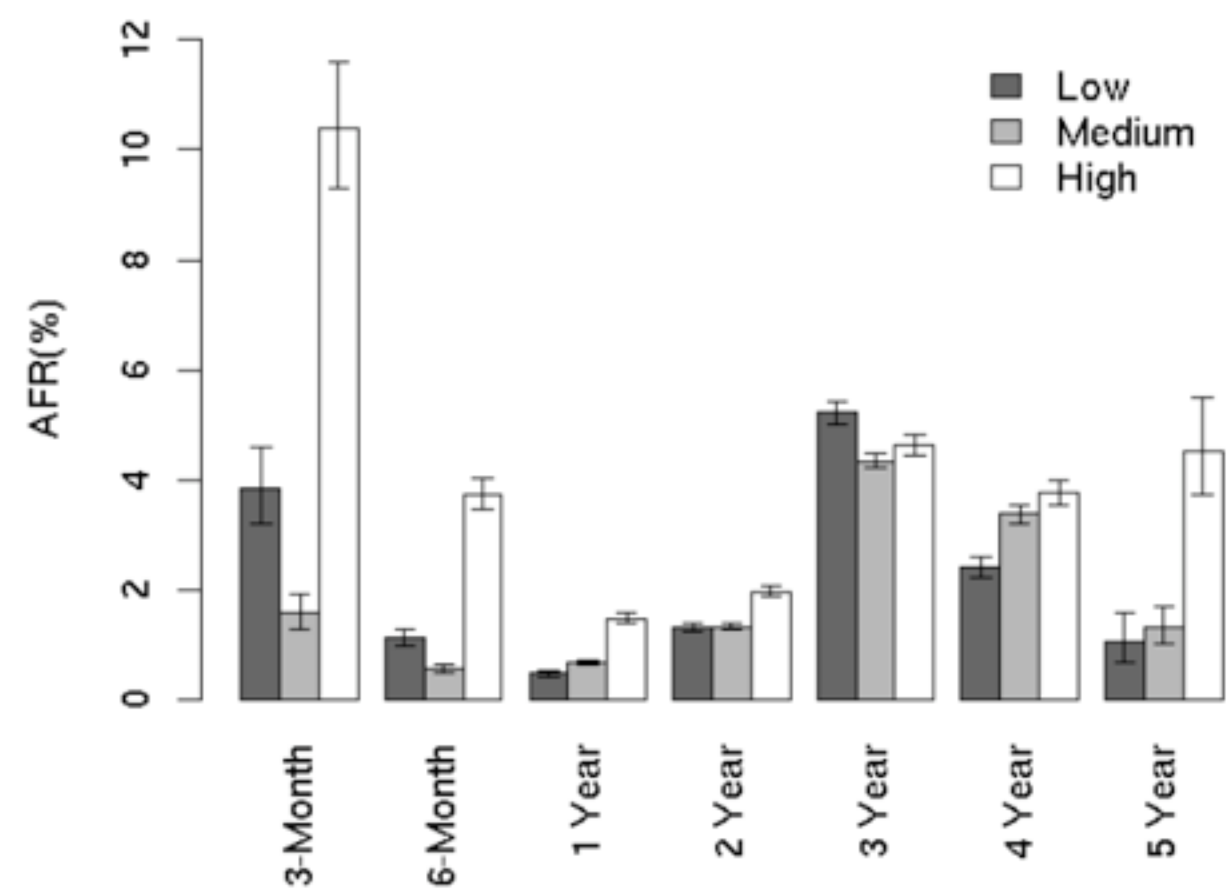
- Failure Trends in a Large Disk Drive Population [Pinheiro2007]
  - > 100.000 disks, SATA / PATA consumer hard disk drives, 5400 to 7200 rpm
  - 9 months of data gathering in Google data centers
  - Statistical analysis of SMART data
- Failure event: *„A drive is considered to have failed if it was replaced as part of a repairs procedure.“*
- Prediction models based on SMART only work in 56% of the cases



Disk Redundancy: Google

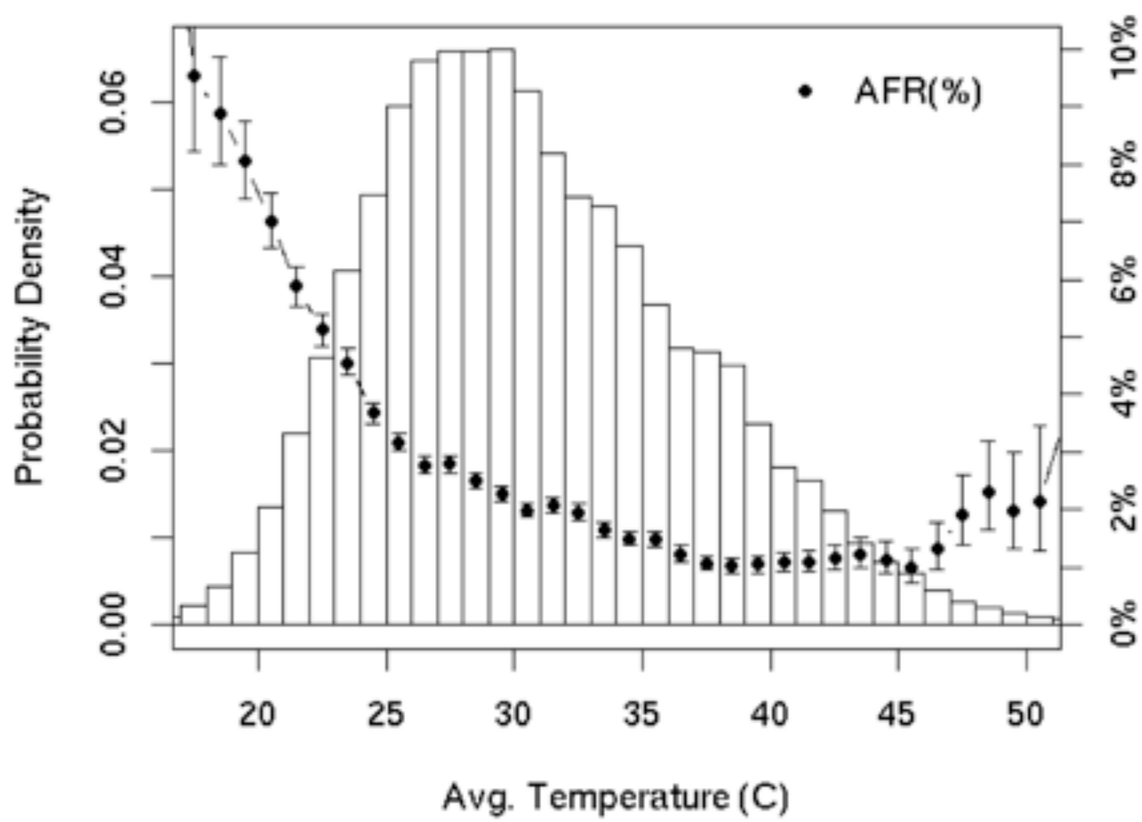
- Failure rates are correlated with drive model, manufacturer and drive age
- Indication for infant mortality
- Impact from utilization (25th percentile, 50-75th percentile, 75th percentile)
  - Reversing effect in third year - „Survival of the fittest“ theory

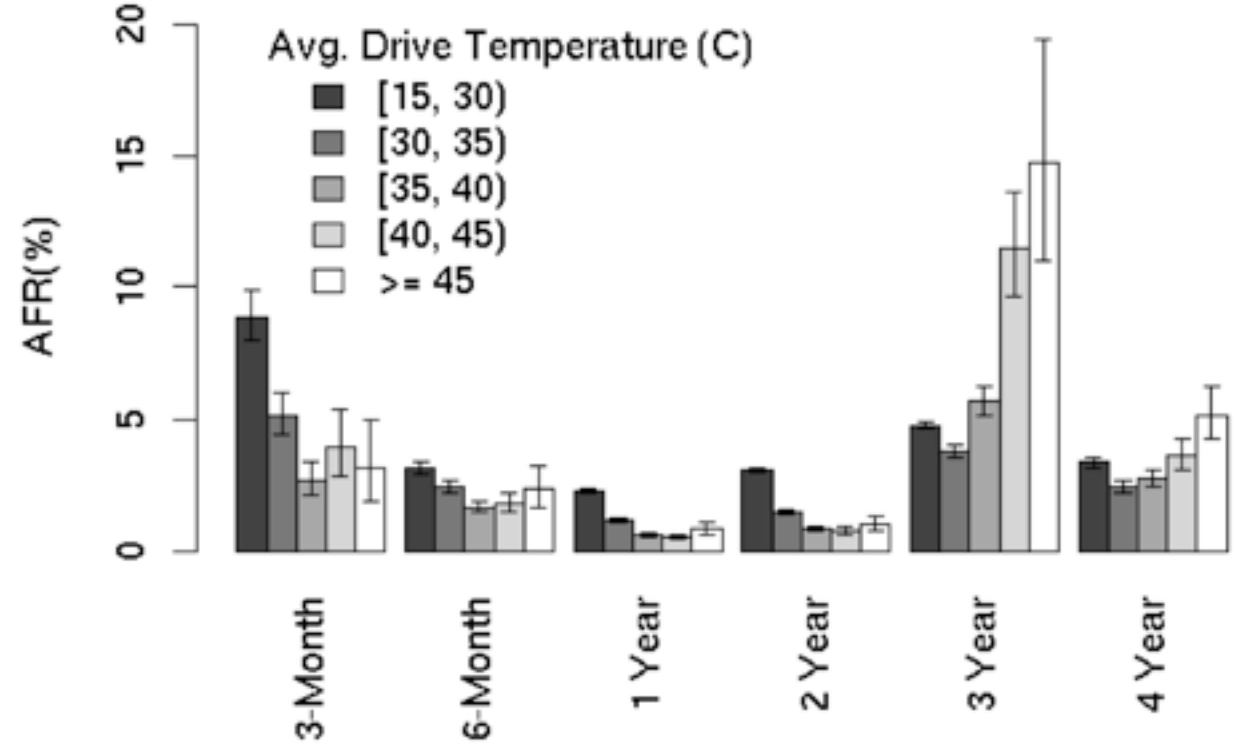




Disk Redundancy: Google

- Temperature effects only at high end of temperature range, with old drivers

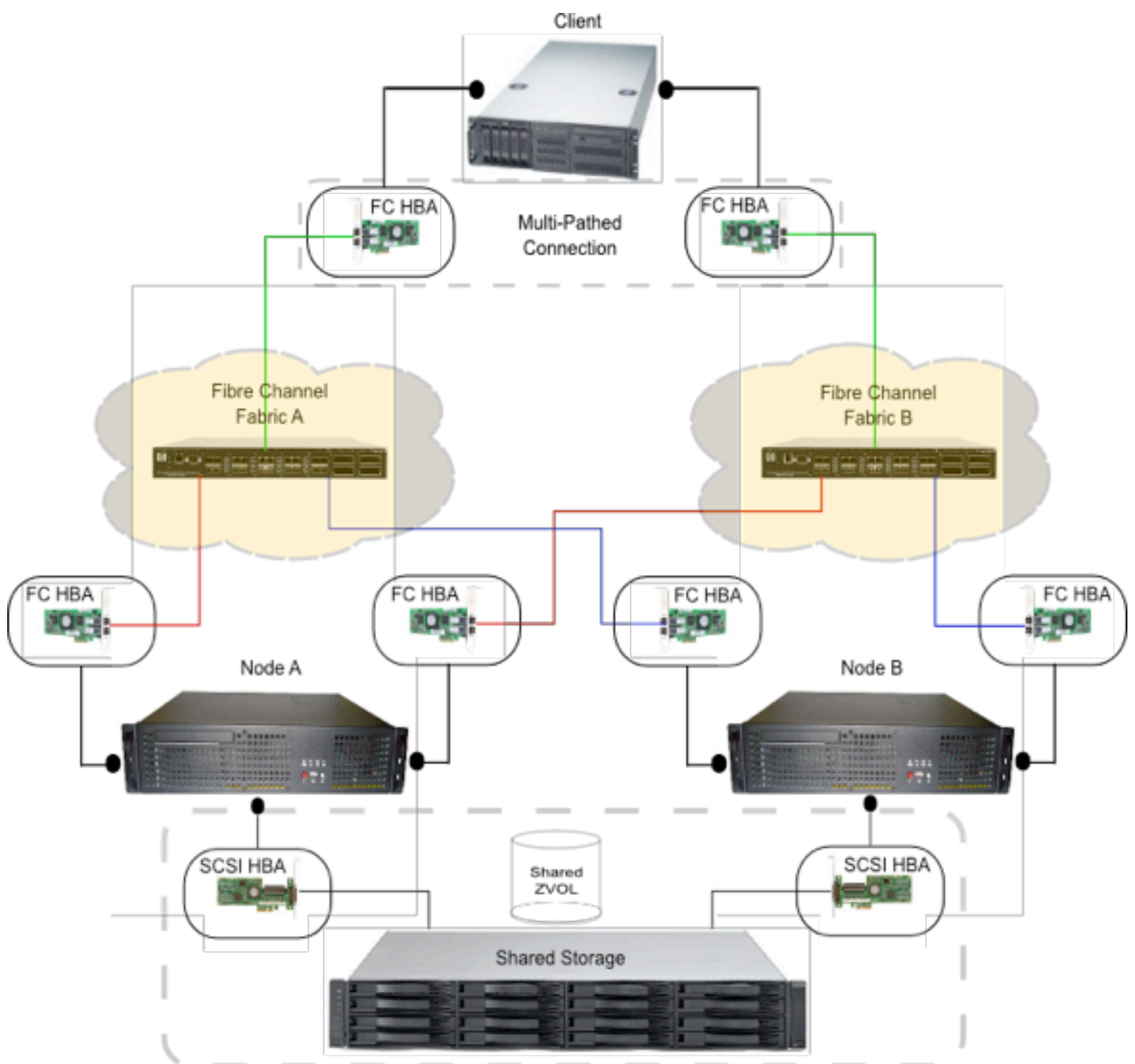




Connection Redundancy - Fibre Channel

- Fibre Channel
  - Developed for HPC, meanwhile standard in SAN technology
  - Can run on copper and fiber-optic channels, primarily SCSI transport
  - Host bus adapter (HBA), switch, disk - all connected by *ports*
- **Multi-pathing** with switched fabric (FC-SW)
  - Combination of switches as *fabric* supports failover and shortest route approach
  - Multi-pathing - redundant HBAs connected to multiple switches
  - Also possible to connect redundant HBAs to different (linked) fabrics
- **Bonding** (client) / **trunking** (switch): Bundle multiple connections to one logical
  - Implementations support failover between the bonding lanes

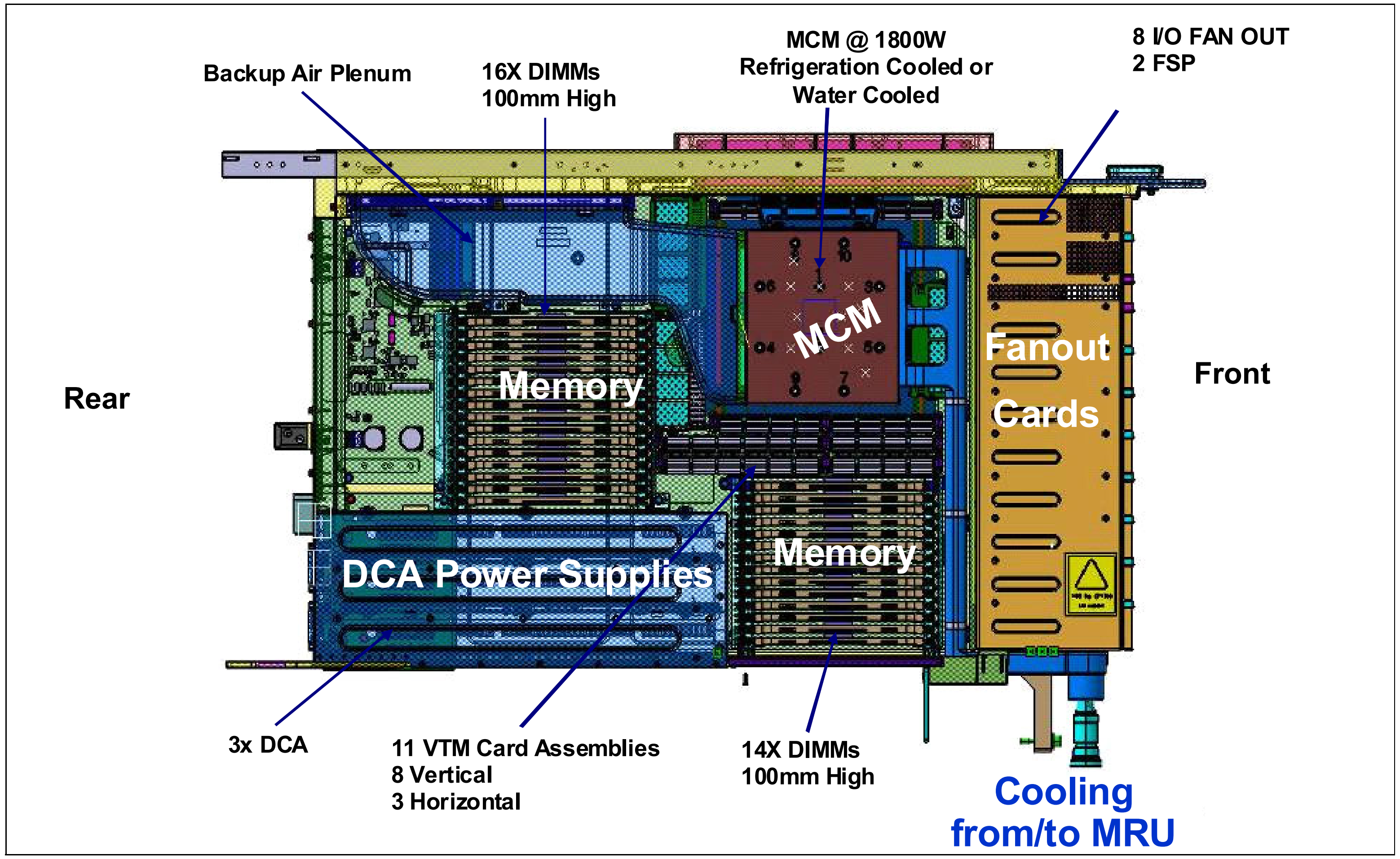
Connection Redundancy - Fibre Channel



www.high-availability.com

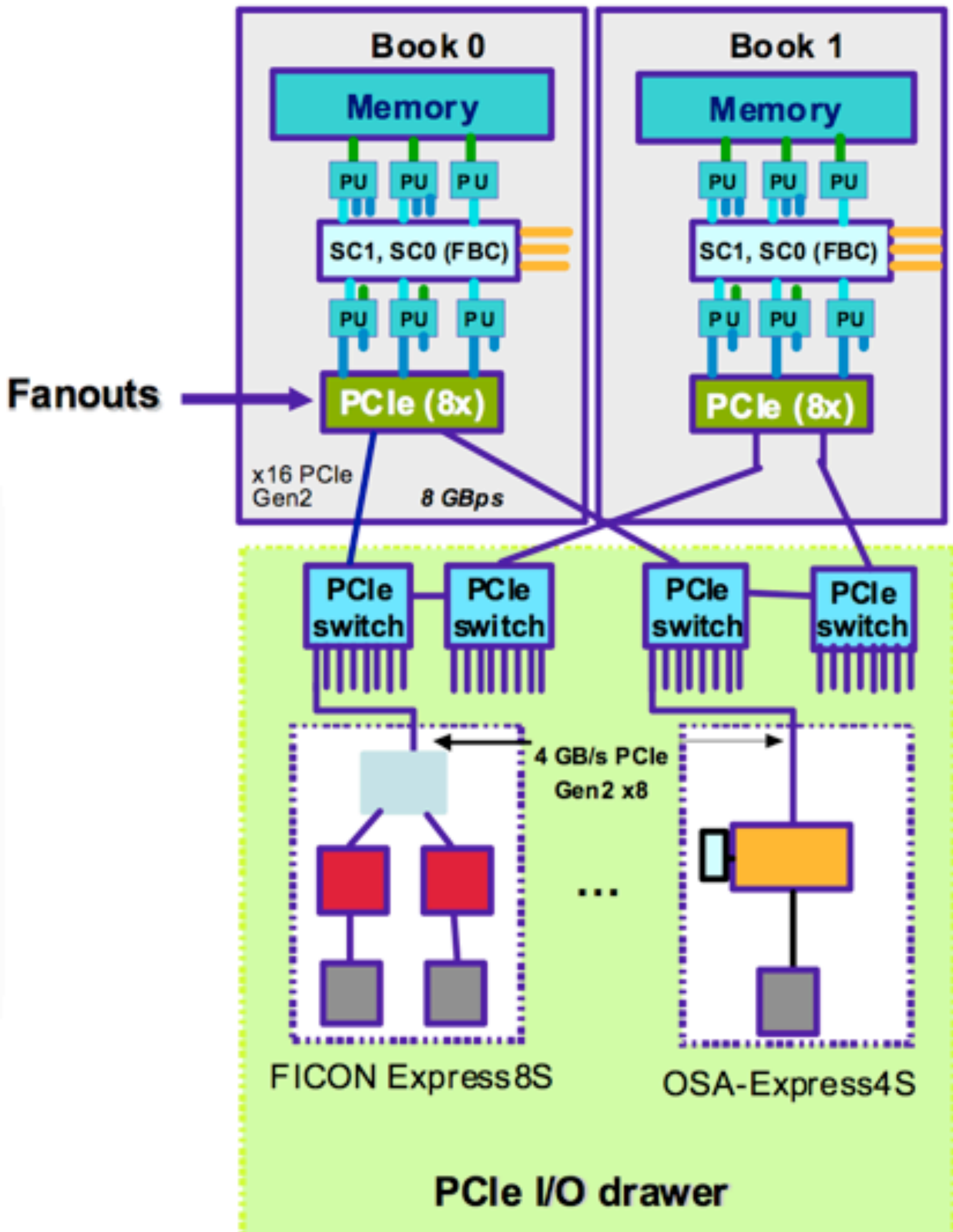
IBM System z - Redundant I/O

- Each processor book has up to 8 dual port fanouts
  - Direct data transfer between memory and PCI/e (8 GBps) or Infiniband (6 GBps)
  - Optical and copper connectivity supported
  - Fanout cards are hot-pluggable, without loosing the I/O connectivity

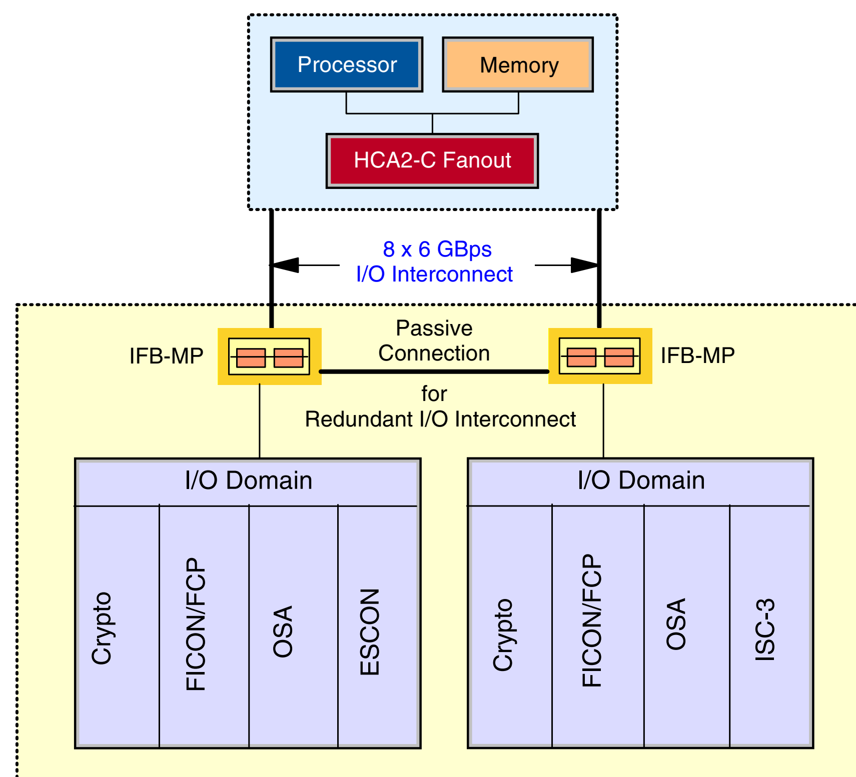


- Air-moving devices (AMD) have N+1 redundancy for fanouts, memory and power

IBM System z - Redundant I/O



- PCI/e I/O drawer supports up to 32 I/O cards from fanouts in 4 domains
  - One PCI/e switch card per domain
  - Two cards provide backup path for each other (f.e. with cable failure)
  - 16 cards max. per switch



I/O Cage

IBM System z - Redundant I/O