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

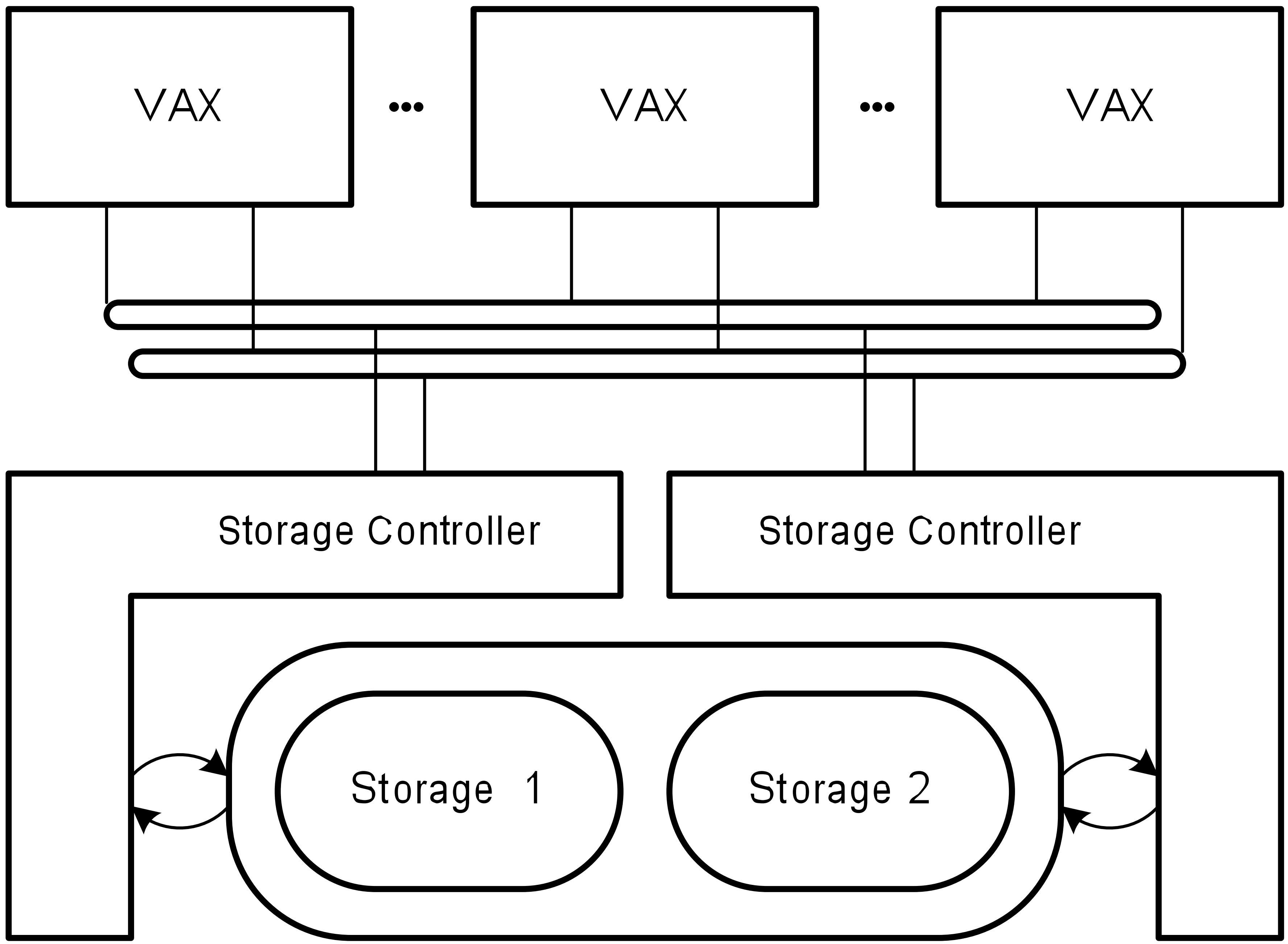
Redundancy (Reiteration)

- 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 fault masking
  - Fault does not show up, since it is transparently removed
  - Examples: Voting, correcting codes, N-modular redundancy (NMR), (4-2) concept, special logic, TMR with duplex
- **Dynamic redundancy**
  - After fault detection, the system is reconfigured to avoid a failure
  - Examples: Back-up sparing, duplex and share, pair and spare
- **Hybrid approaches**

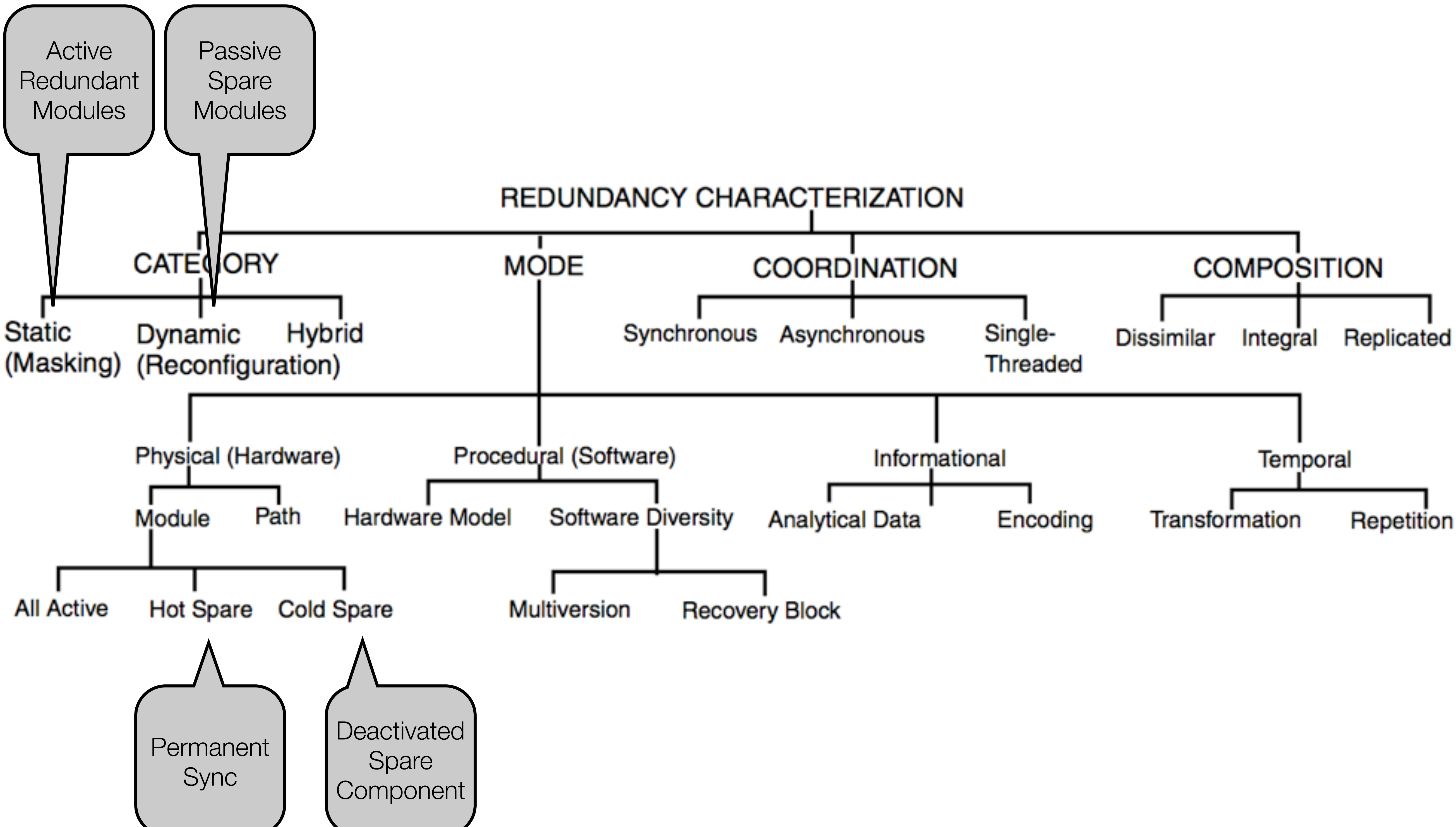
Redundancy

- Redundancy is never for free !
  - Hardware: Additional components, area, power, shielding, ...
  - Software: Development costs, maintenance costs, ...
  - Information: Extra hardware for decoding / encoding
  - Time: Faster processing (CPU) to achieve same application performance
- Always demands tradeoff against achievable dependability

Example: VAX Spatial Hardware Redundancy



Redundancy Classification (Hitt / Mulcare)

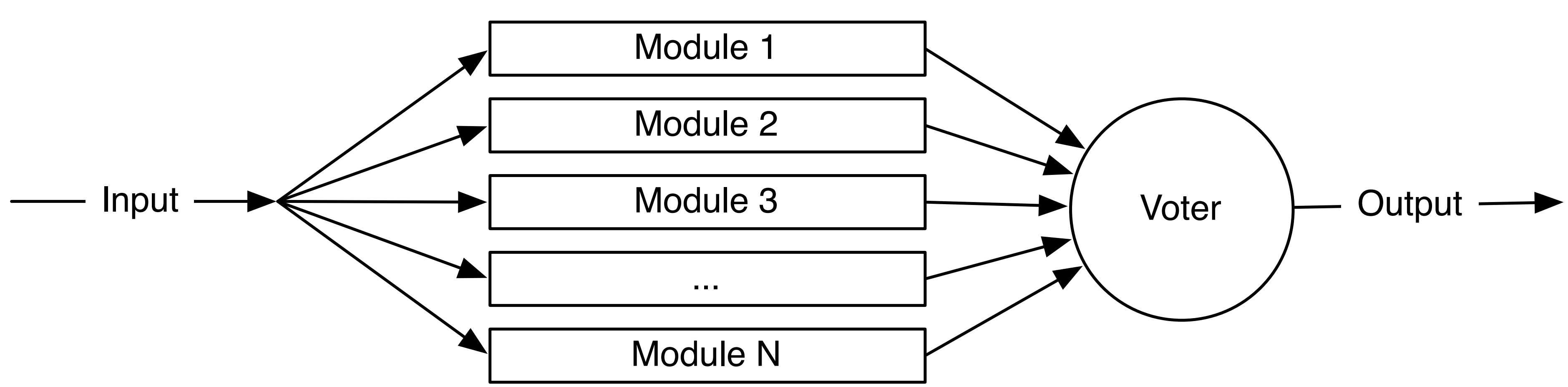


Static Redundancy: N-Modular Redundancy

- Voter gives correct result if the voter is correct and the module majority are correct
  - Compare results itself or checksums of it
- Tripe-modular redundancy (TMR): 2 out of 3 modules deliver correct results
- Generalization with N-modular redundancy: N=2m+1
- Standard case without any redundancy is called *simplex*

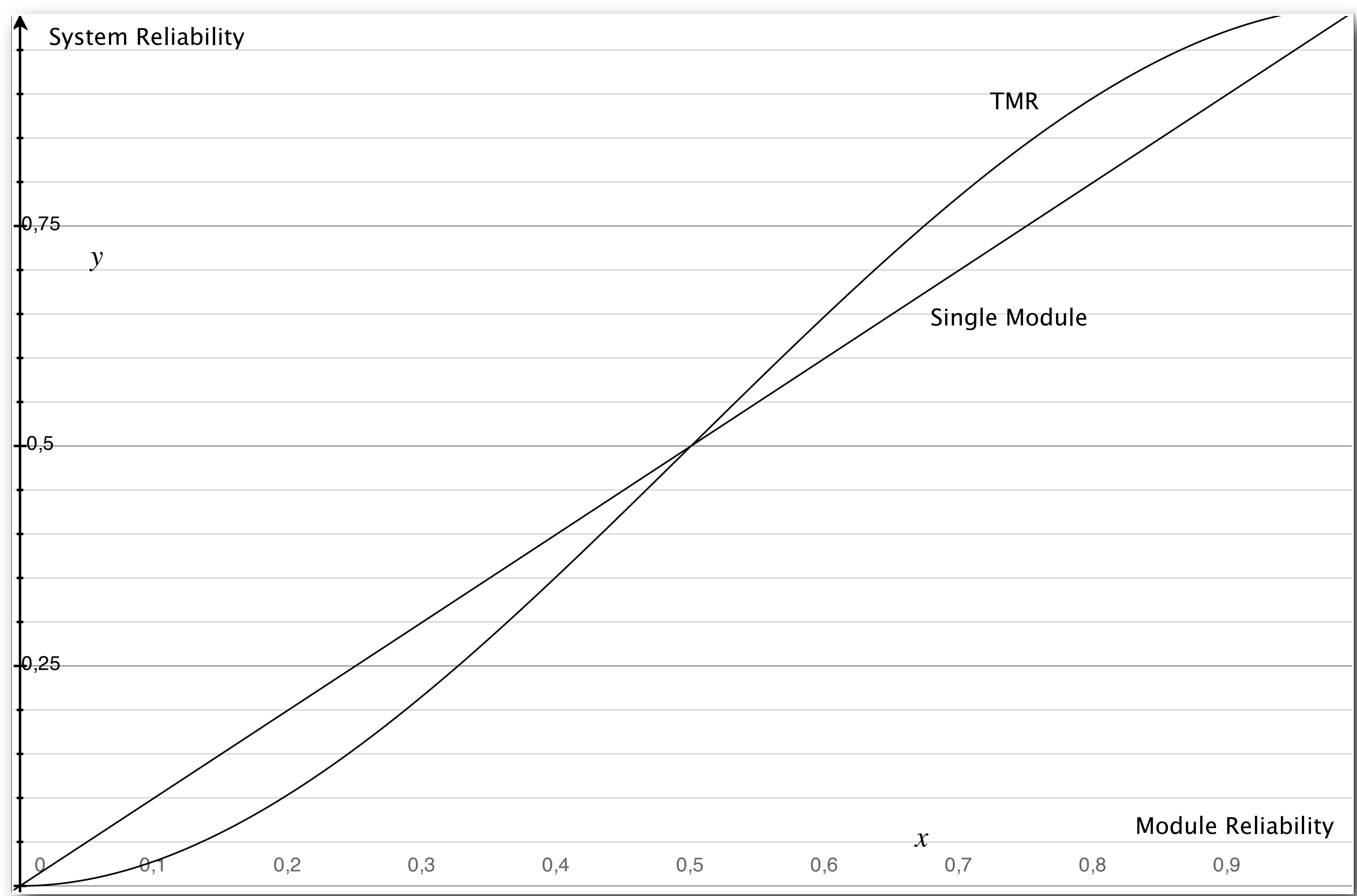
$$\begin{aligned} R_{TMR} &= R_V \cdot R_{2-of-3} \\ &= R_V(R_M^3 + 3R_M^2(1-R_M)) \end{aligned}$$

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



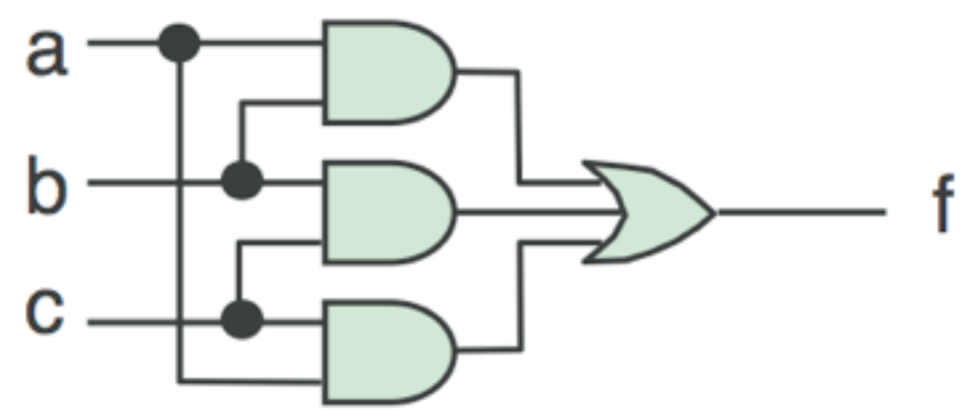
TMR Reliability

- TMR is appropriate if $R_{TMR} > R_M$
  - Example with perfect voter - TMR only improves system when $R_M > 0.5$
  - Voter must have $R_V > 0.9$ for $R_{TMR} > R_M$



Hardware Voting

- Base for hardware solution is the 1-bit majority voter
  - $f=ab + ac + bc$
  - Delivers 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-bite voter



A	B	Y = A ∧ B
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

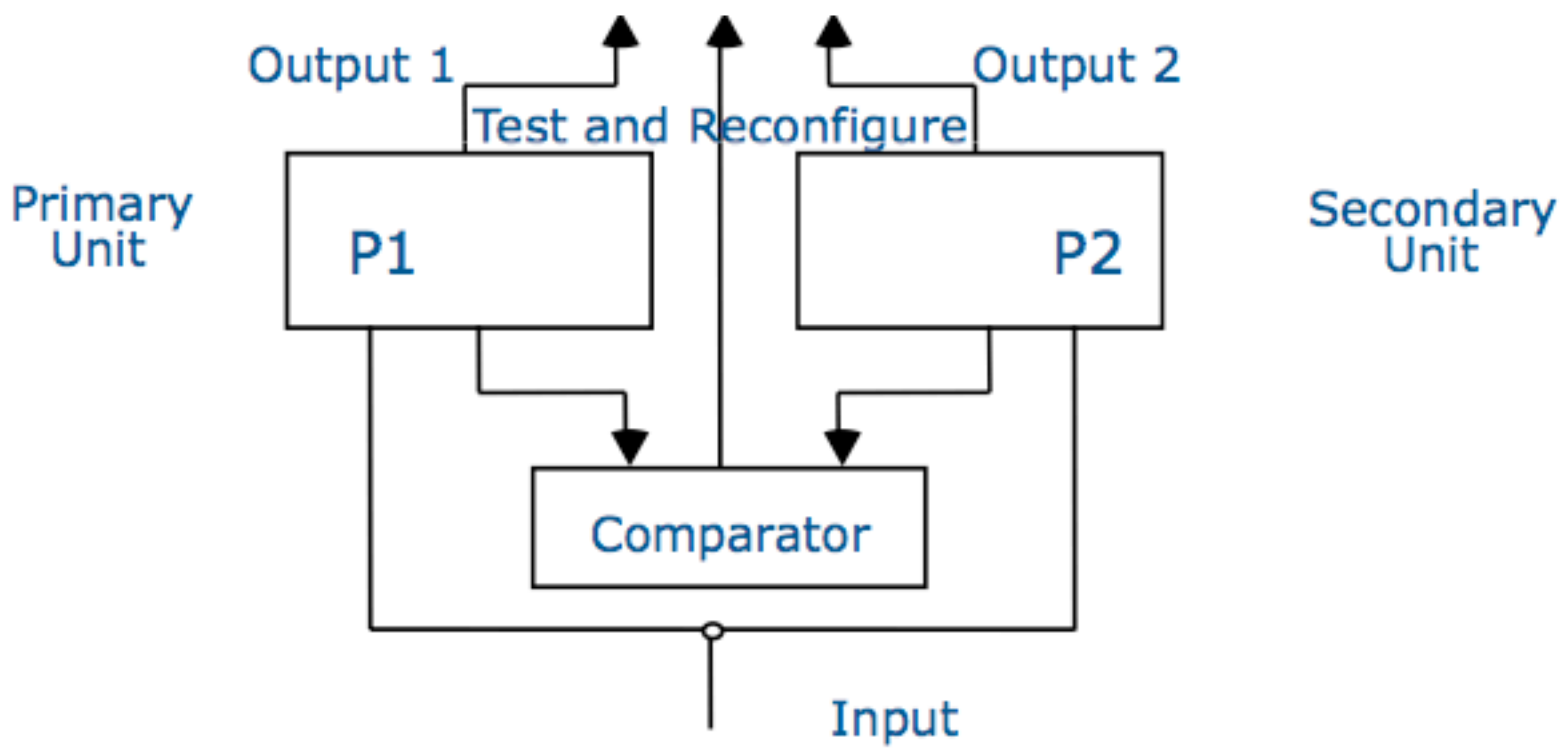
A	B	Y = A ∨ B
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**1**

Voting Strategy (Reiteration)

- **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 $W_1+W_2+W_3=1$

Dynamic Redundancy: Duplex Systems

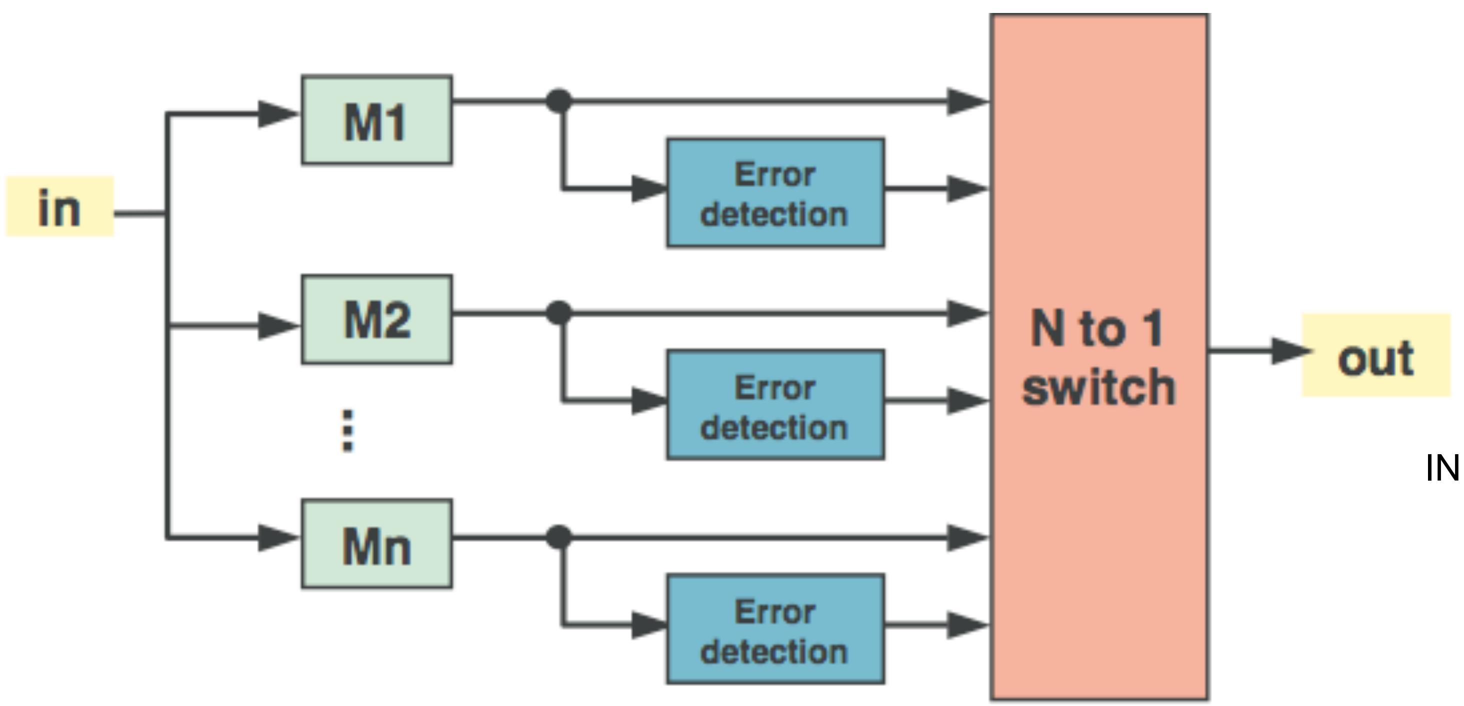
- Have relevant modules redundant, switch on detected failure
- Identification on mismatch („test“)
  - Self-diagnostics procedure
  - Self-checking logic
  - Watchdog timer, e.g. components resetting each other
  - Outside arbiter for signatures or black box tests

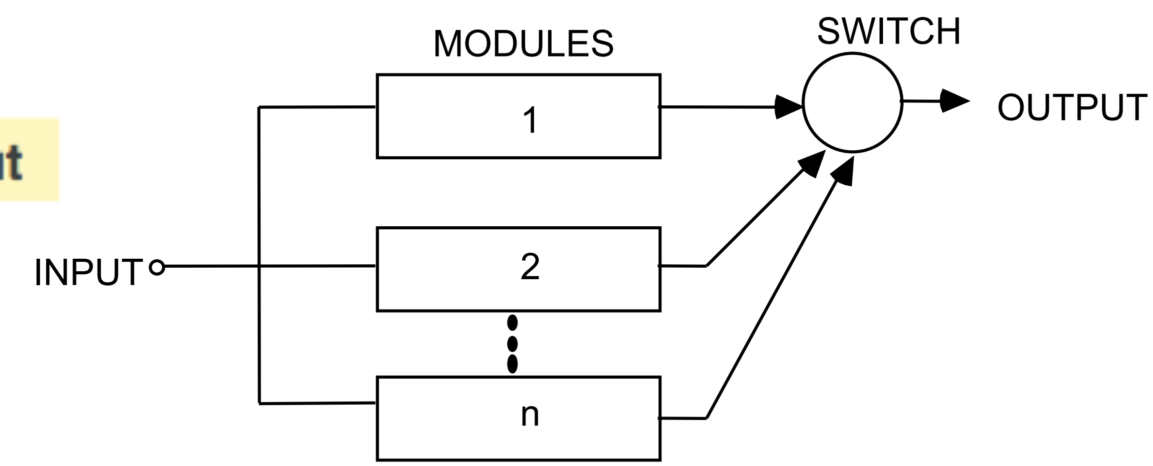


- Test interval depends on application scenario - each clock period / bus cycle / ...
- Also called **dual-modular redundancy**

Dynamic Redundancy: Back-Up Sparing

- Working module, set of spare modules that can be switched in as replacement for a faulty module
- **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

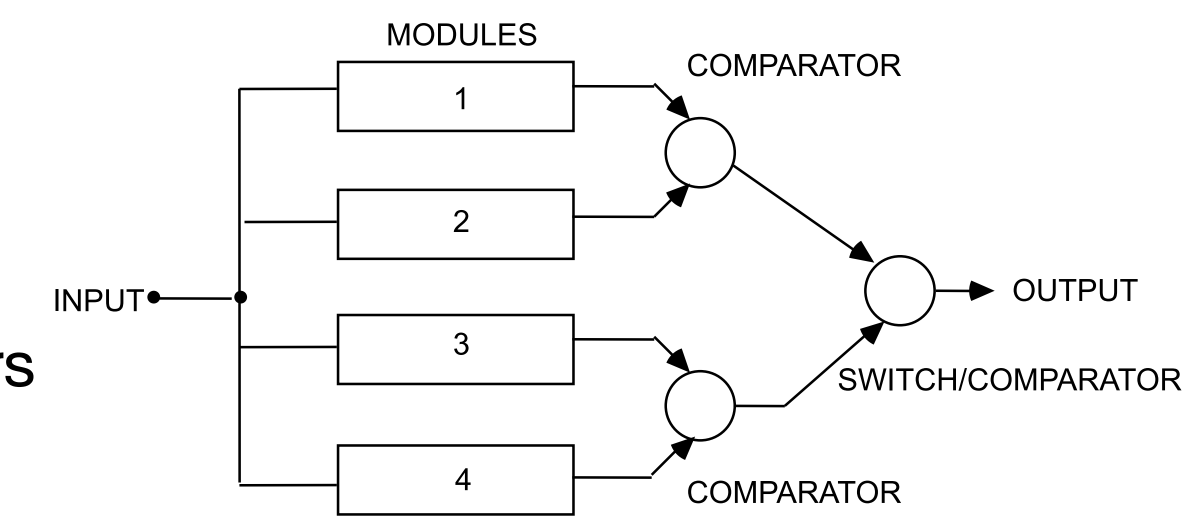




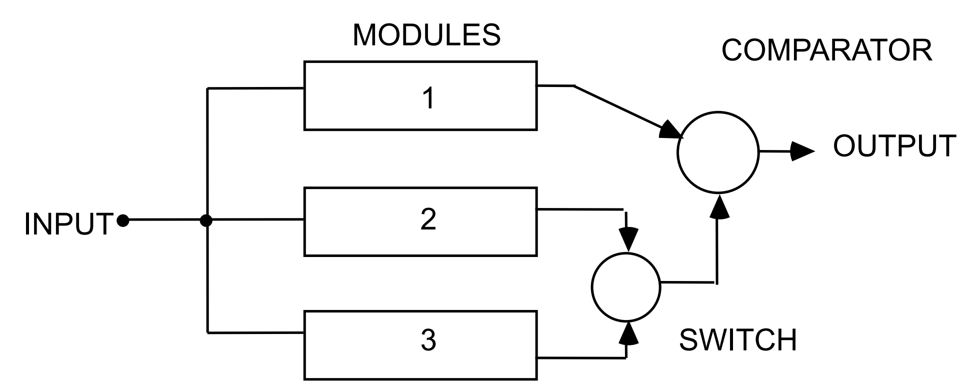
HOT, WARM AND COLD SPARES

Pair and Spare / Duplex and Spare

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

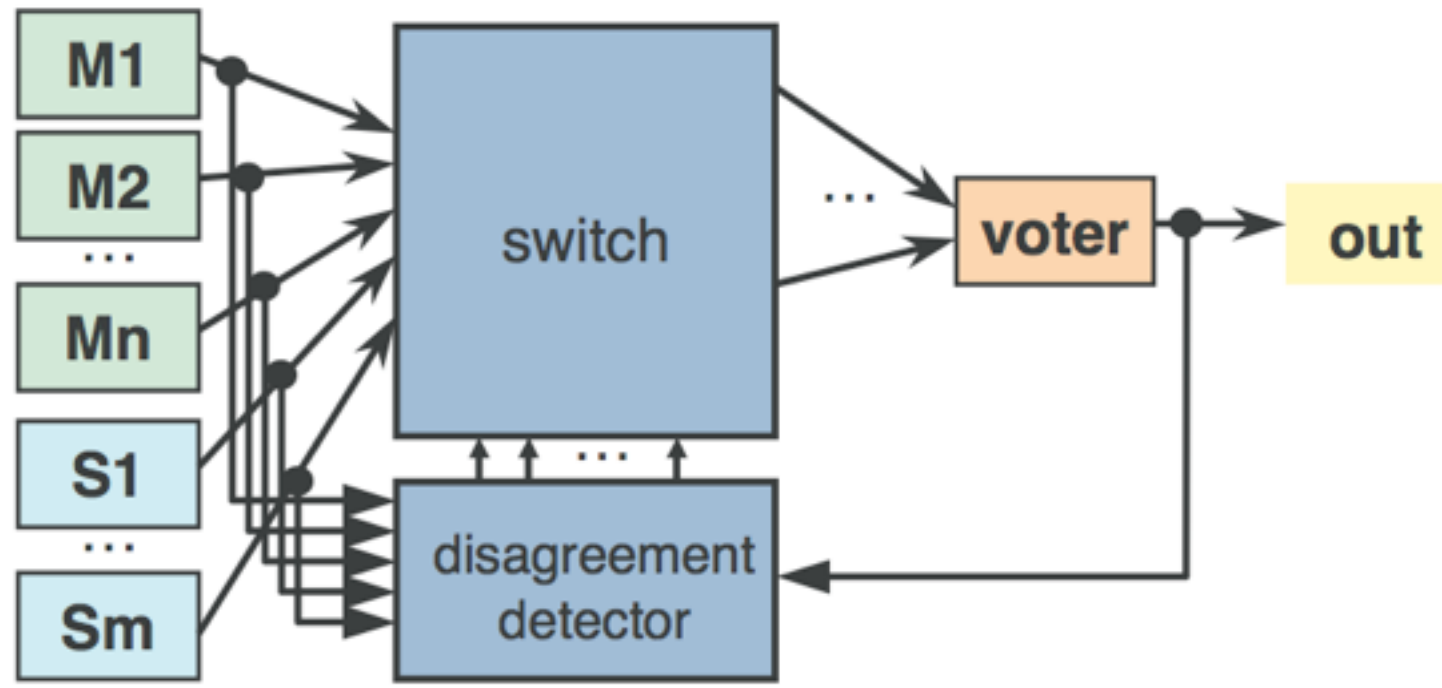


- **Duplex and spare**
  - Extend spare by another duplex unit



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

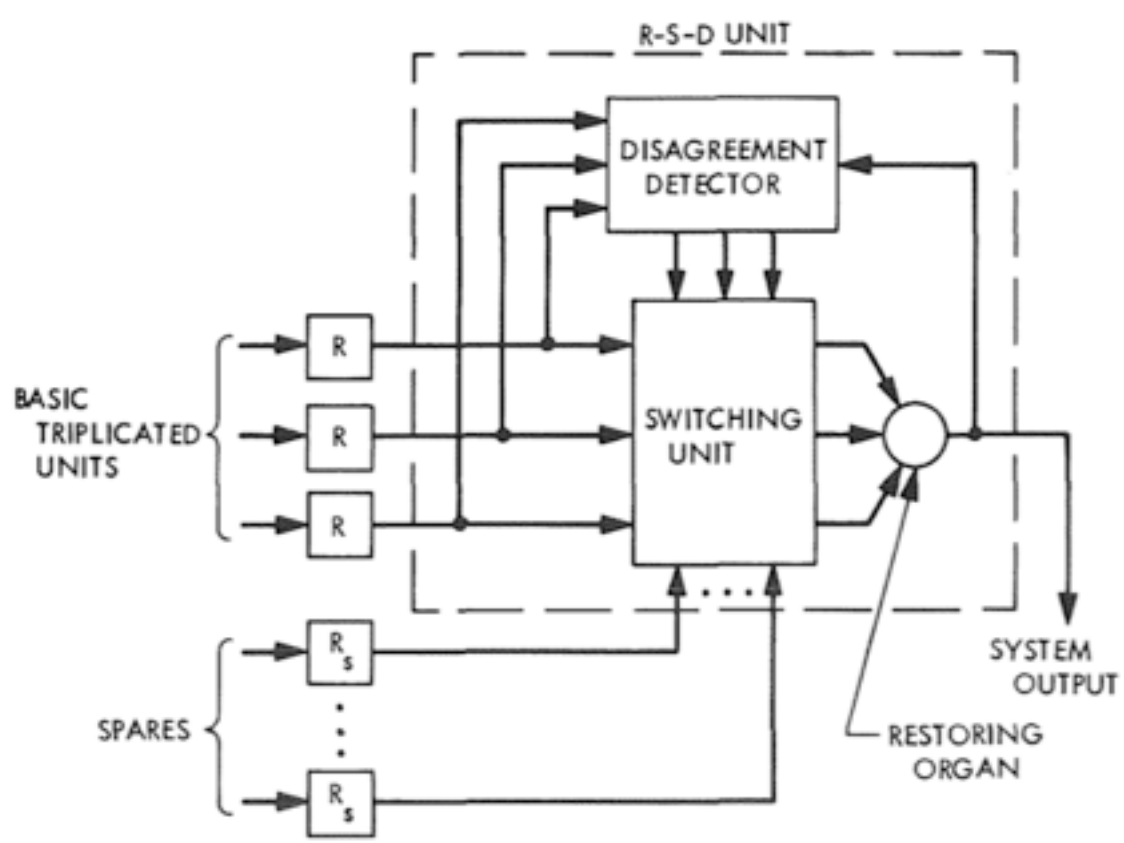


- Reliability as long as the spare pool is not exhausted
- Improves fault masking capability of TMR
  - Can tolerate 2 faults with one spare, while classic NMR would need 5 modules (with the typical majority voting)

TMR with Spares

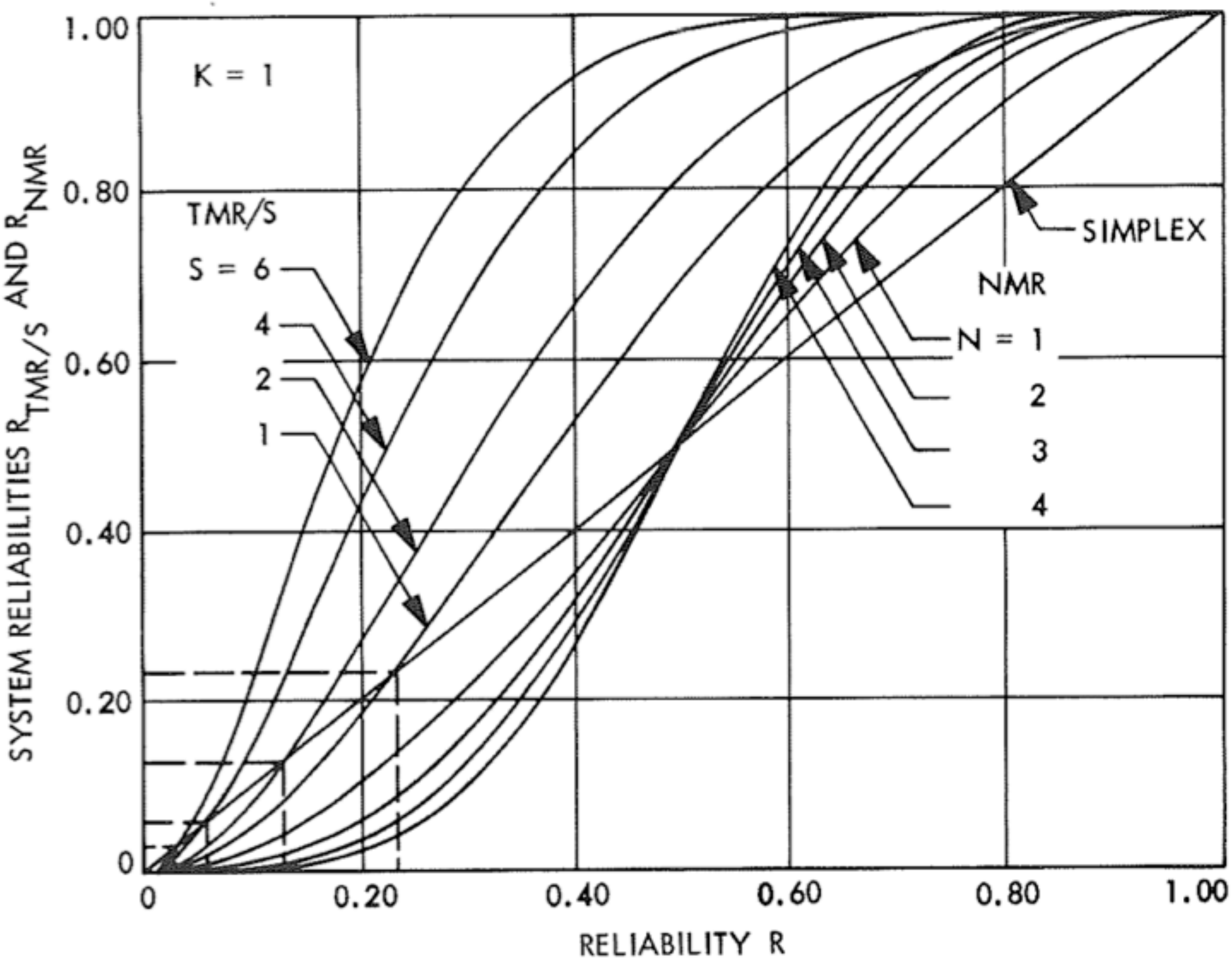
- Basic reliability computation based on similar module failure rate in spares and non-spares
  - At least any two of all modules must survive

$$R_{TMR/S} = 1 - (1 - R)^{S+2}[1 + R \times (S + 2)]$$



NASA Technical Report 32-1467, 1969

Comparison TMR vs. TMR/S vs. NMR

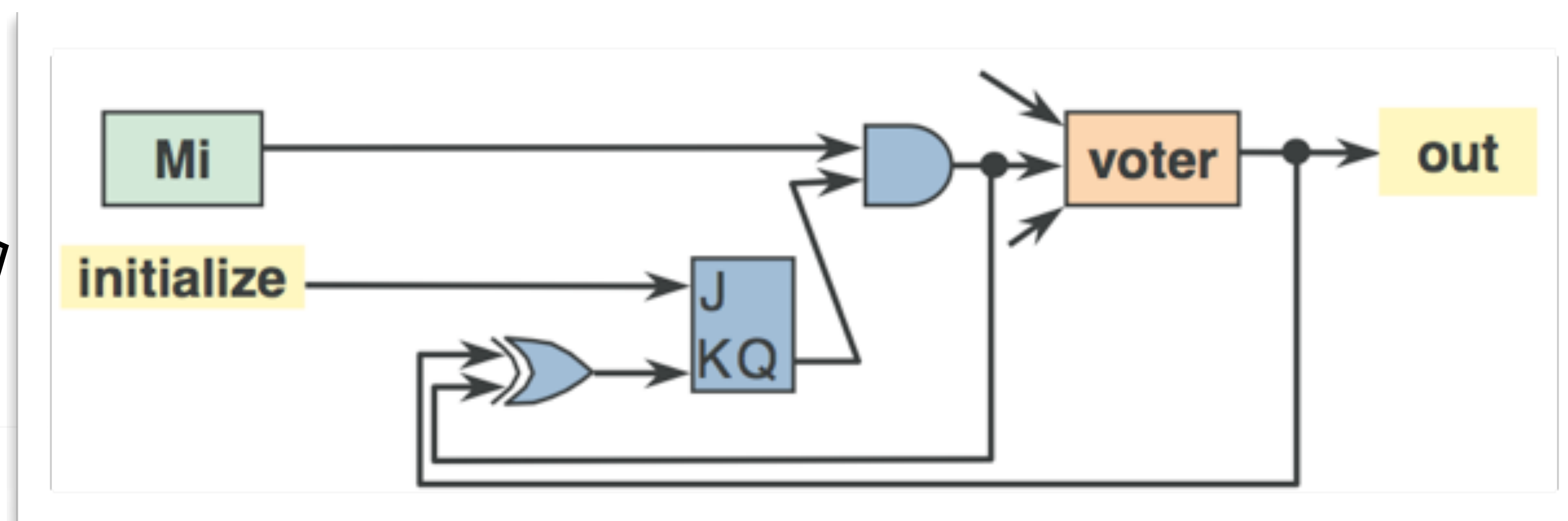


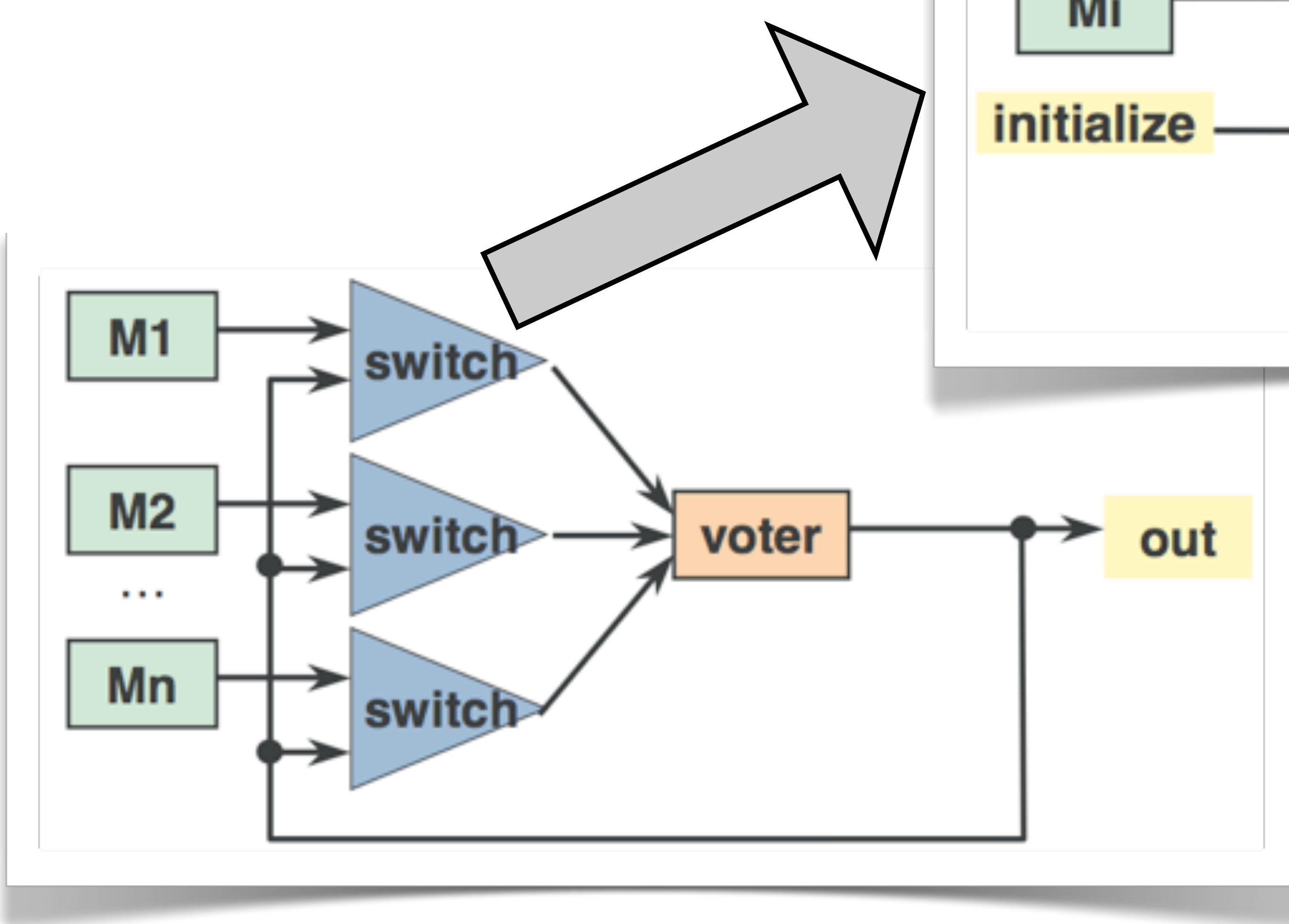
$\#Units = 2N + 1$

NASA Technical Report 32-1467, 1969

Hybrid Approaches

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

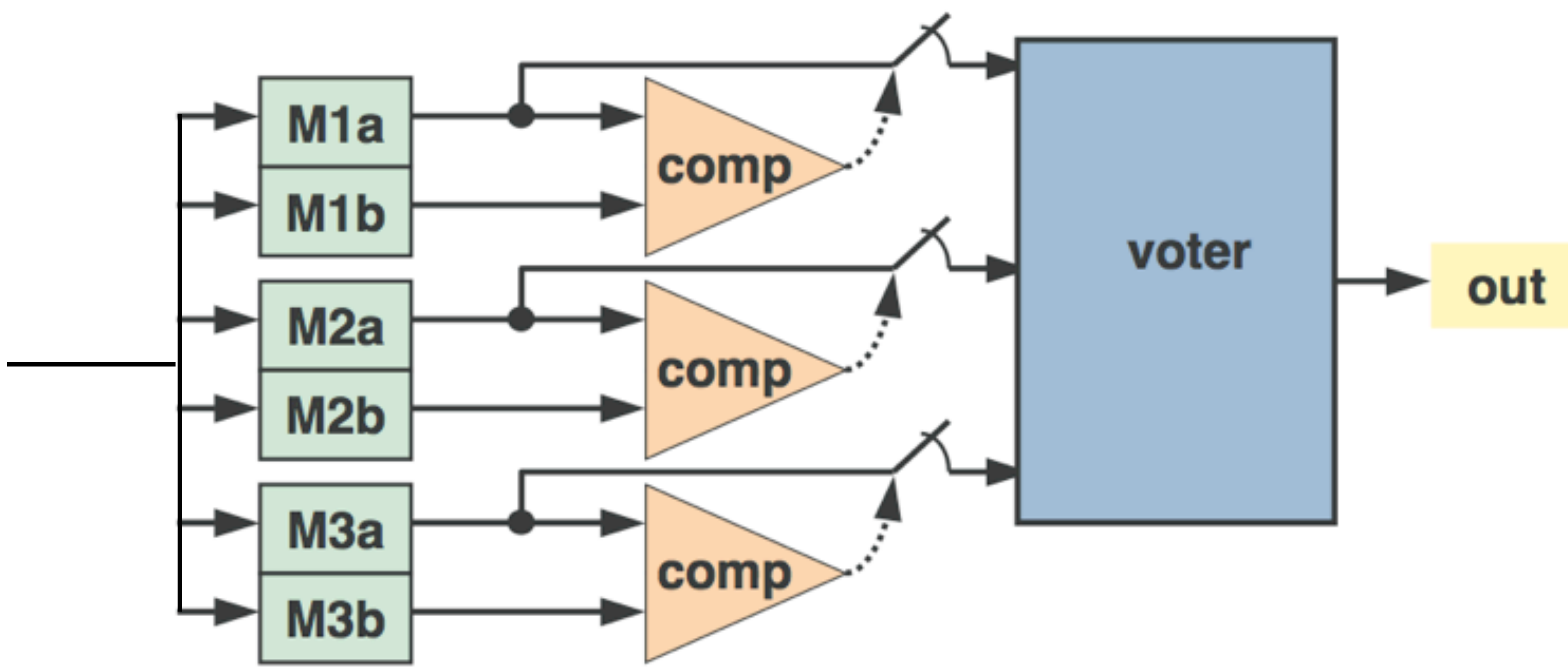




- Threshold gates are analog circuit elements
- Potentially more reliable than hybrid

Hybrid Approaches

- Triple Duplex Architecture
  - TMR with duplex modules, used in the Shinkansen (Japanese train)
  - Removal of faulty module based on comparator, allows tolerating another fault (on the same comparator) in the further operation



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

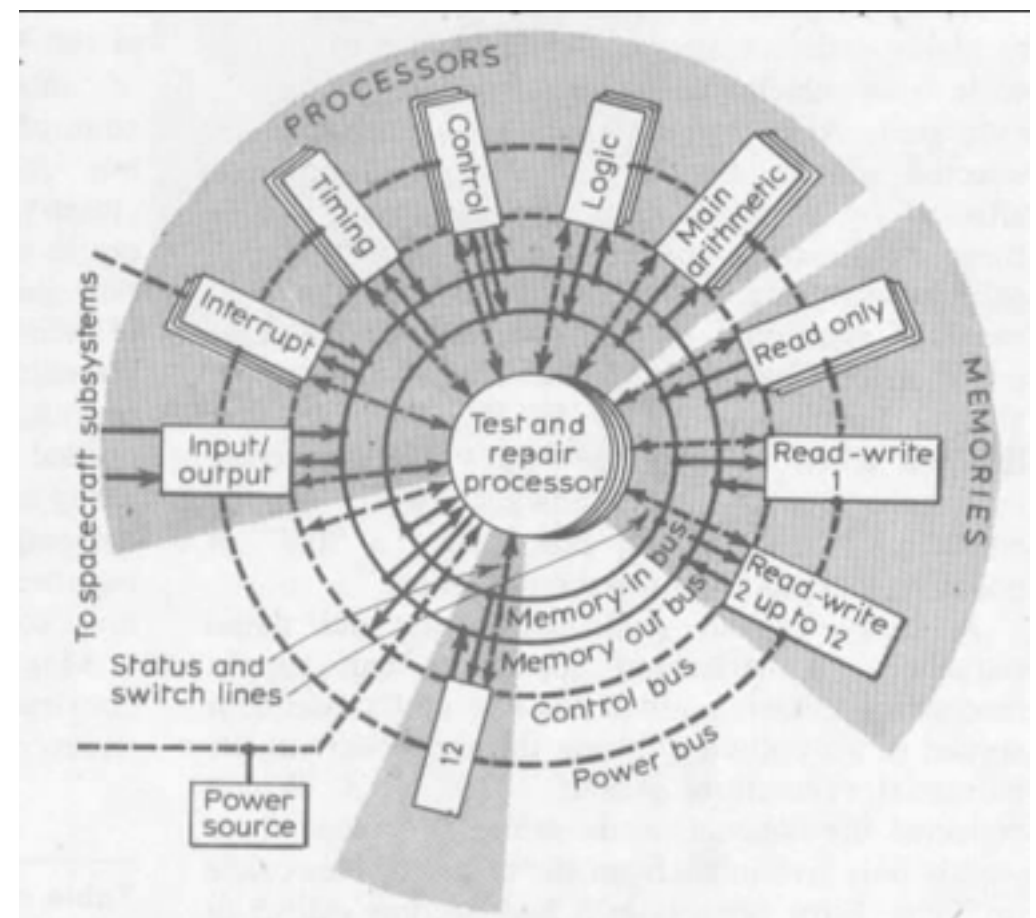
COM1	
Component	%
Power supply	34.8
Memory	20.1
Hard drive	**18.1**
Case	11.4
Fan	8.0
CPU	2.0
SCSI Board	0.6
NIC Card	1.2
LV Power Board	0.6
CPU heatsink	0.6

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

Memory Redundancy

- Redundancy of memory for data for fault 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

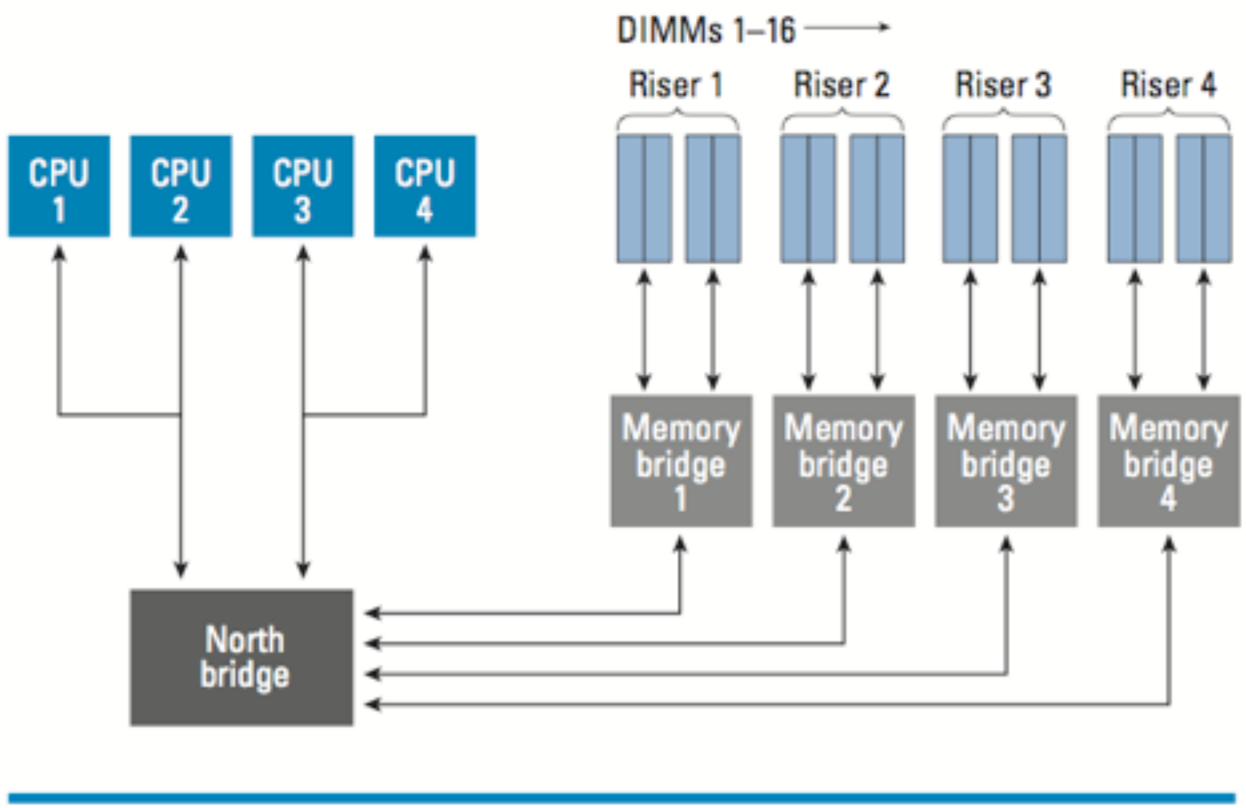
- Standard technology in DRAMs
  - Bit-per-byte **parity**, check on read access - implemented by additional parity memory chip
  - **ECC** with Hamming codes - 7 check bits for 32 bit data words, 8 bit for 64 bit
    - Leads to 72 bit data bus between DIMM and chipset
    - Computed by memory controller 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
  - 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 splitted in 18 bit chunks an distributed on 4 DIMM modules
    - Sum of 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

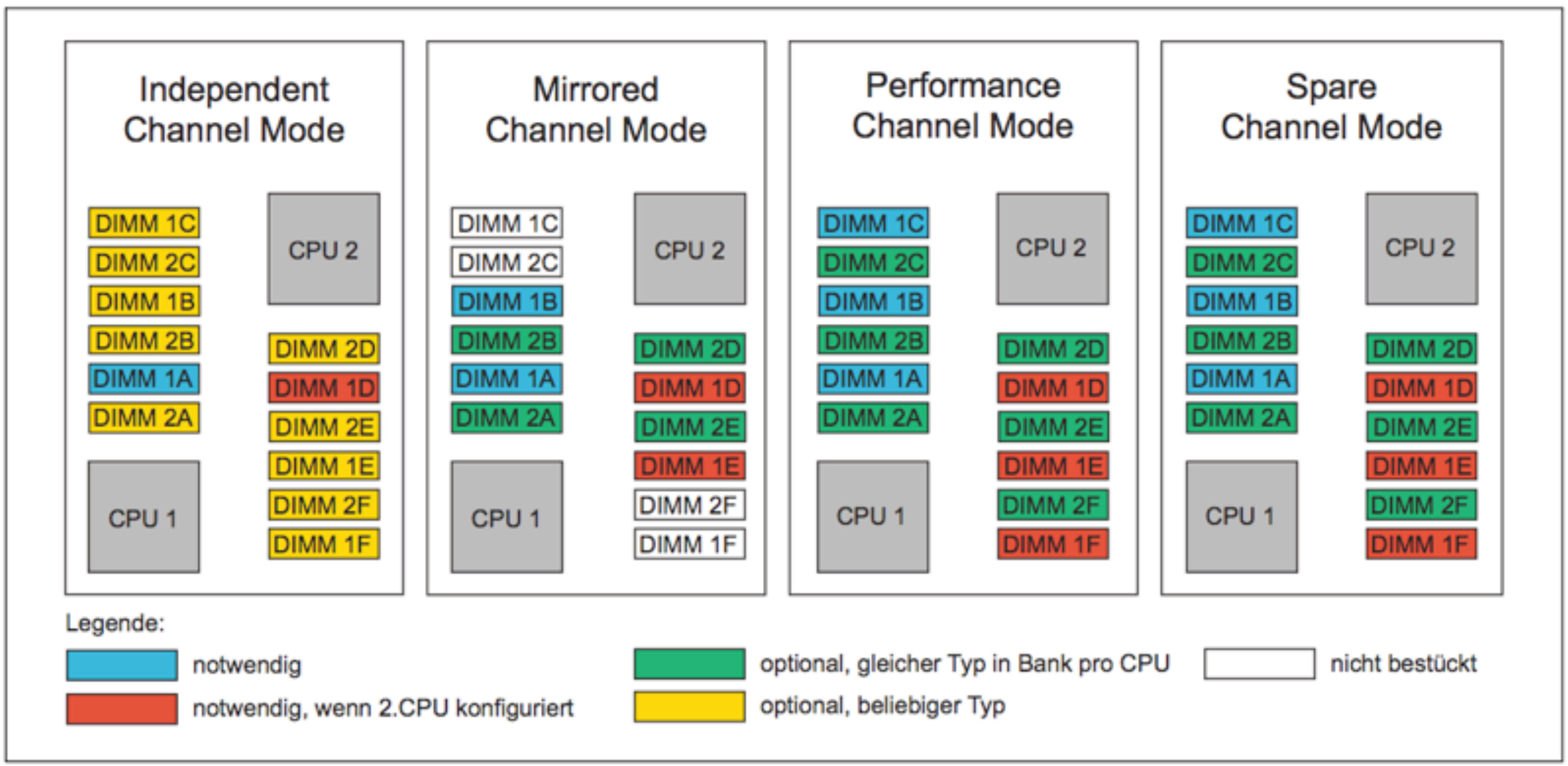
- Dell PowerEdge Servers, 2005 (taken from www.dell.com)



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)



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, if AFR < 1
- Disk MTTF: On average, one failure takes place in the given disk hours
- Example: Seagate Barracuda ST3500320AS: MTTF=750000h=85.6 years
  - With thousands disk, on average every 750h a disk fails
  - Measured by the manufacturer under heavy load and physical stress
  - MTTF equals roughly MTBF with these numbers, so 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:

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

RAID MTTF Calculation [Patterson]

- From last slide: $MTTF_{DiskArray} = \frac{MTTF_{Disk}}{n}$
- D - Total number of data disks
- G - Number of data disks in a group (e.g. G=1 in RAID1)
- C - Number of check disks (e.g. parity) in a group (e.g. D=1 in RAID1)
- n_G = D / G = number of groups

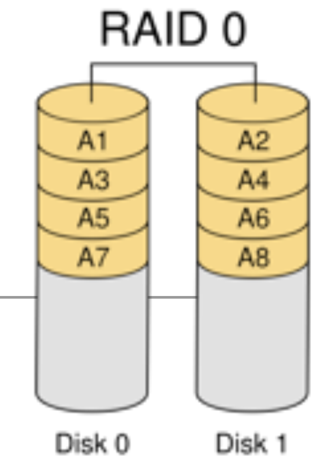
$$MTTF_{Group} = \frac{MTTF_{Disk}}{G+C} \cdot \frac{1}{p_{SecondFailureDuringRepair}}$$

- Average number of second failures during repair comes again from disk MTTF

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

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

(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

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

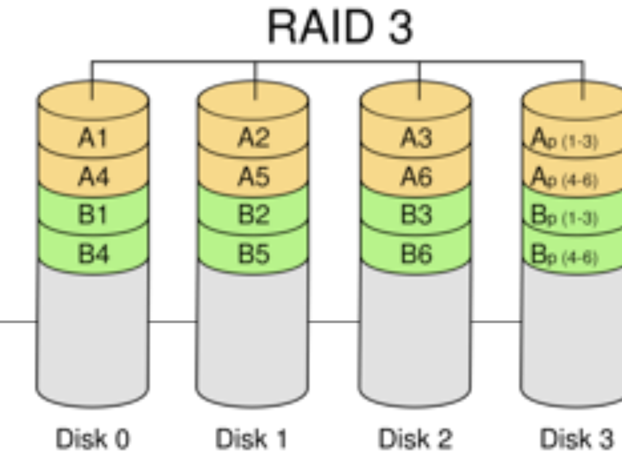
(C) Wikipedia

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 ECC Hamming code disk
  - No commercial implementation, since high ECC disk capacity needs
  - Online verification and correction during read
- **Raid 3** - Byte-level striping with dedicated parity disk
  - All data disks used equally, one parity disk as bottleneck (C=1)
  - Bad for concurrent small accesses, good sequential performance
  - 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

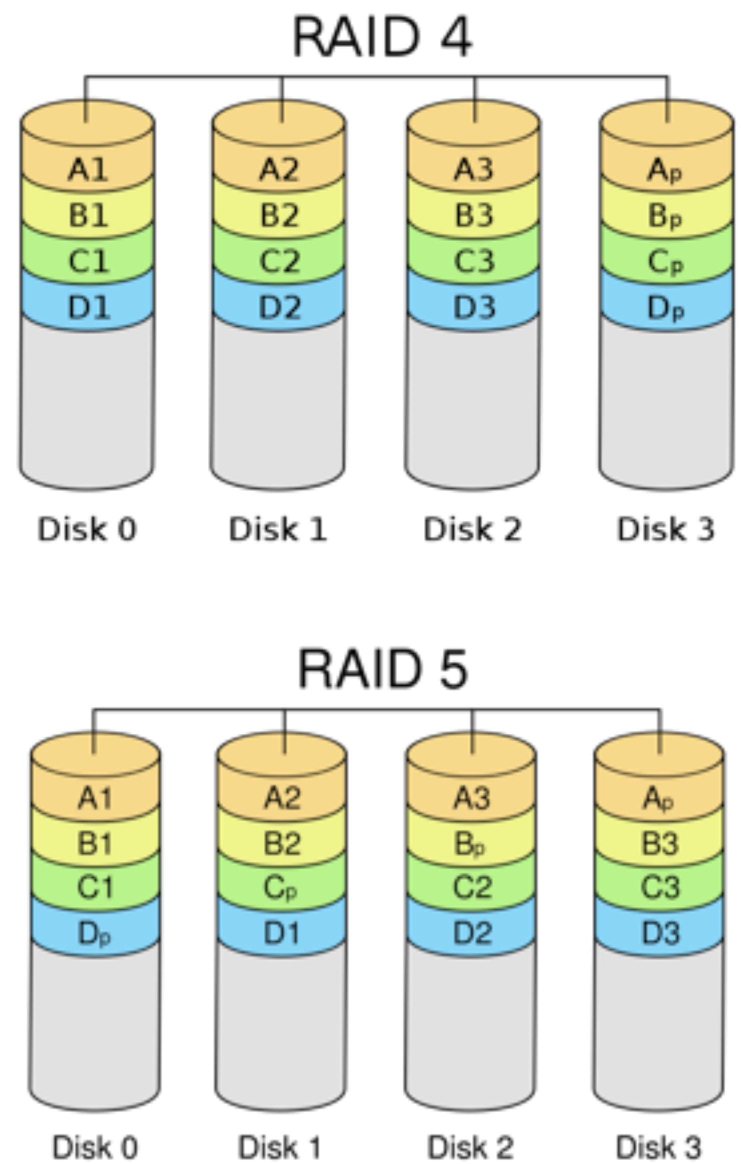
Disk	Byte							
1	1	1	0	0	1	0	0	1
2	0	1	1	0	1	1	1	0
3	0	0	0	1	0	0	1	1
4	1	1	1	0	1	0	1	1
Parity	0	1	0	1	1	1	1	1

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, but 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 are read to compute new parity
    - For every changed data bit, flip parity bit

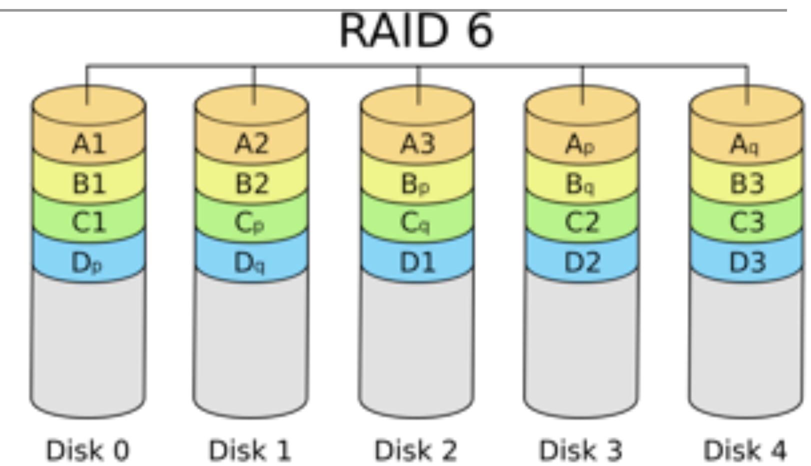
$$MTTF_{Raid5} = \frac{MTTF_{Disk}}{N} \cdot \frac{\frac{MTTF_{Disk}}{N-1}}{MTTR_{Disk}}$$

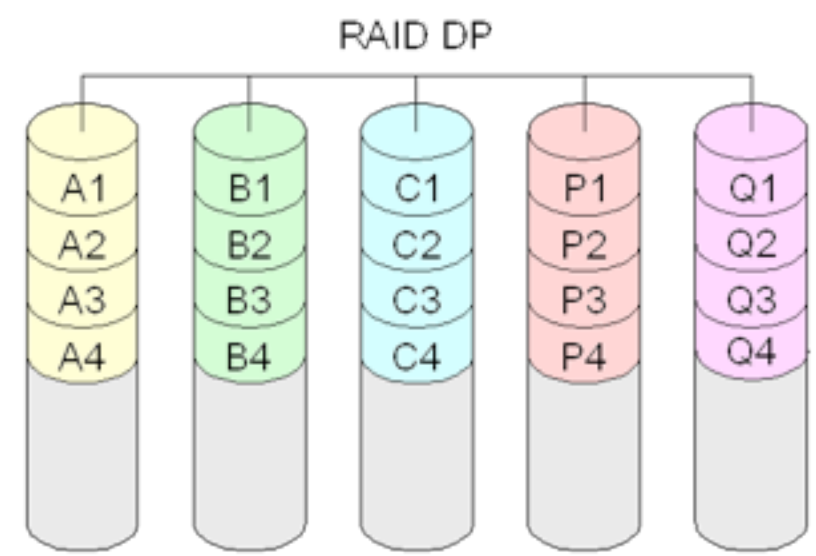


(C) Wikipedia

RAID 6 / DP

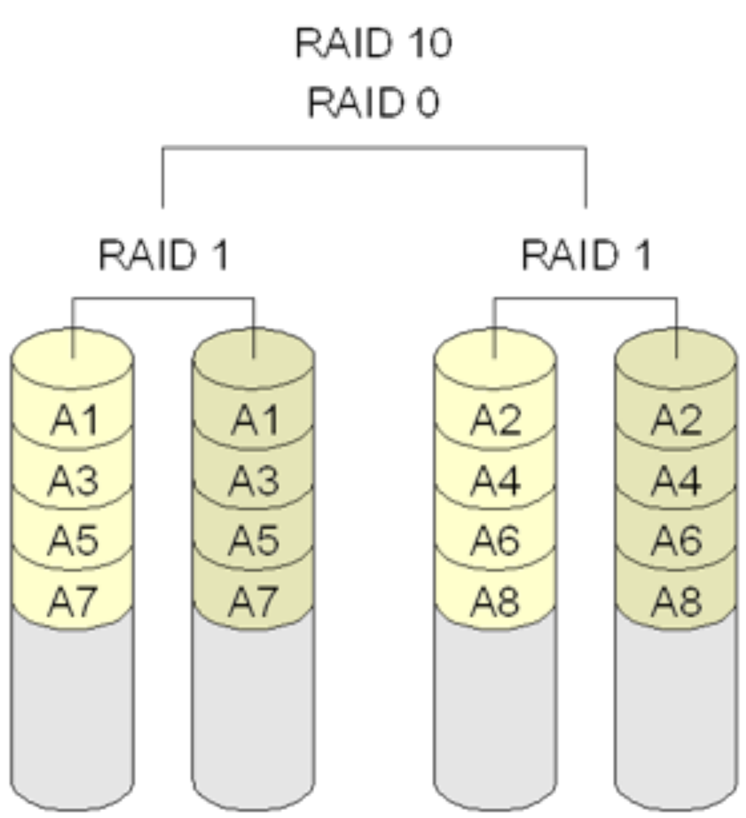
- Raid 6 - Block-level striping with two parity schemes
  - Extension of RAID5, can sustain multiple drive failures at the same time
  - High controller overhead to compute parities, poor write performance
- RaidDP - RAID 4 with additional diagonal parity
  - Easier recovery, can compensate two disk failures

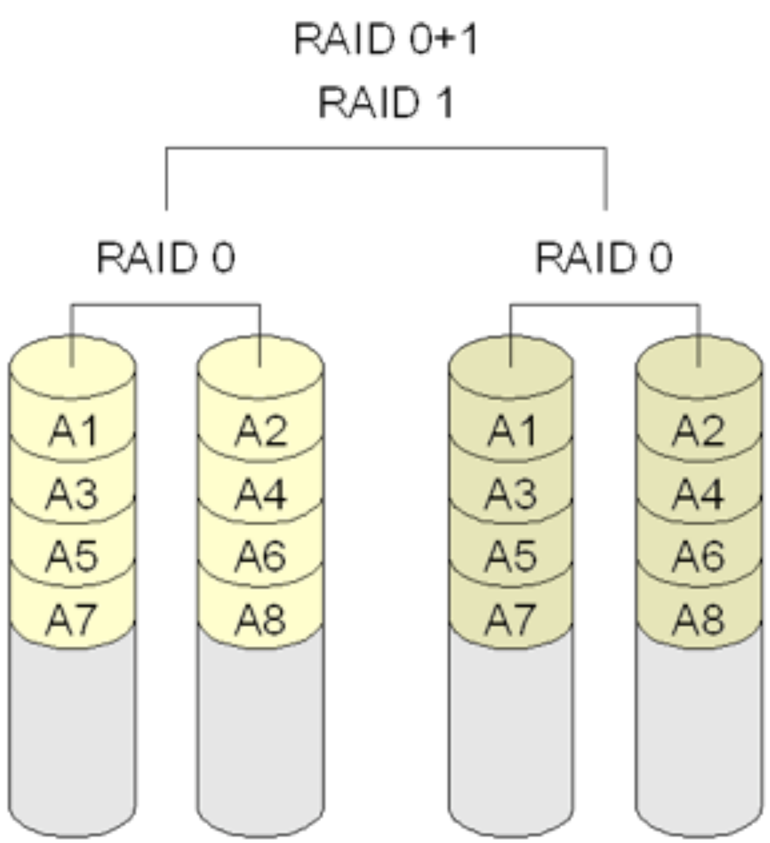




RAID

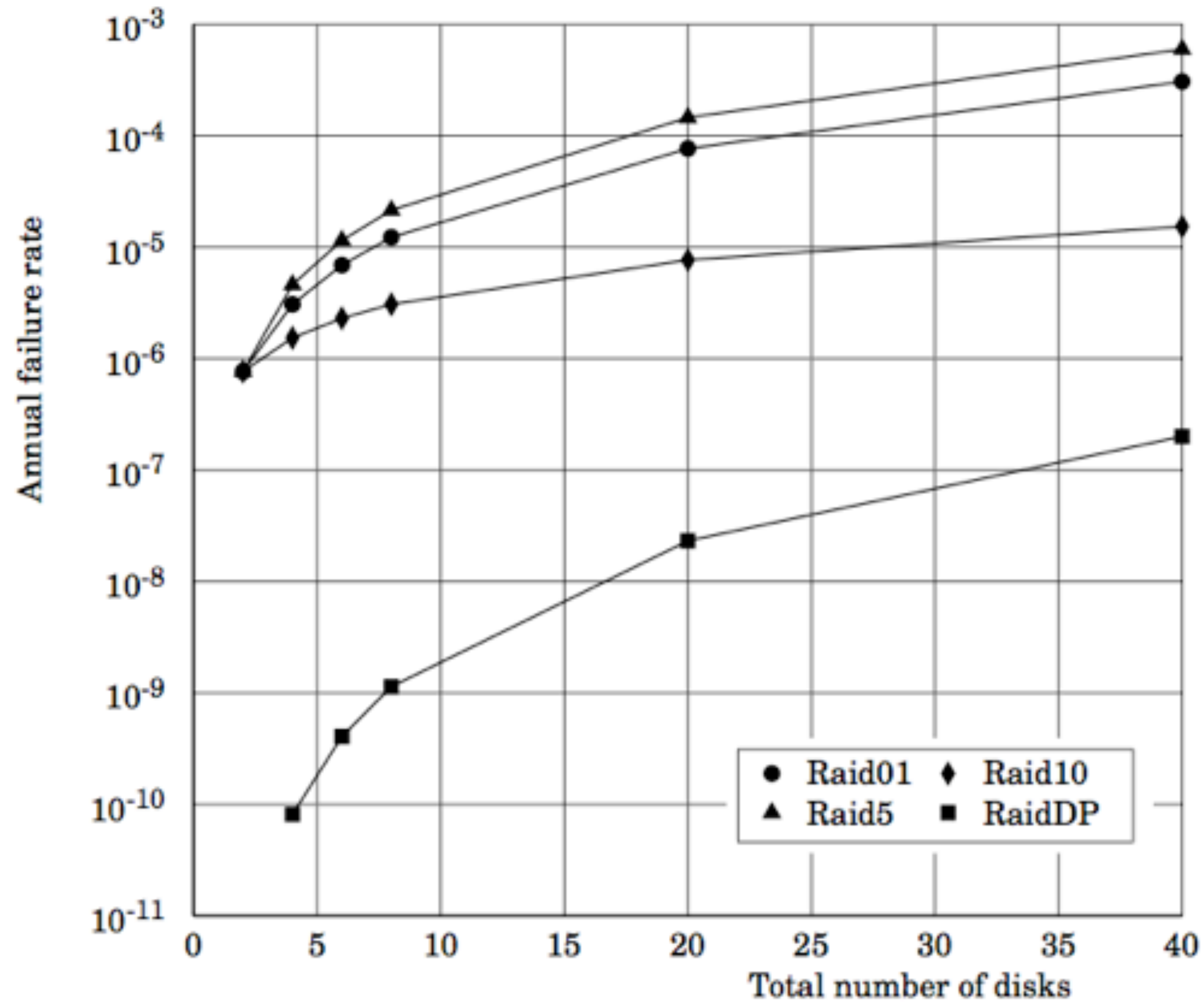
- Raid 01 - Every mirror is a Raid 0 stripe (min. 4 disks)
- Raid 10 - Every stripe is a Raid 1 mirror (min. 4 disks)





RAID Analysis (Schmidt)

- Take the same number of disks in different constellations
  - Ignores resulting capacity, $AFR_{Disk} = 0.029$, 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 resynchronisation times



  - RAID10: Only one disk needs to be copied to the spare
  - RAID5 / RAIDDP: 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 for statistical analysis of SMART data
  - Failure rates are correlated with drive model, manufacturer and vintage
  - Temperature effect only for high end and older drives
  - Prediction models based on SMART only work in 56% of the cases