DESIGN TECHNIQUES FOR FAULT TOLERANCE

Design Techniques

- We can classify the design techniques as two classes:
  - Systems using fault masking
  - Systems not using fault masking
- With fault masking:
  - No need to locate fault or contain it!
  - Errors are not generated by faults
- Without fault masking:
  - Detect faults -> locate faults -> recover from fault

Concept of Redundancy

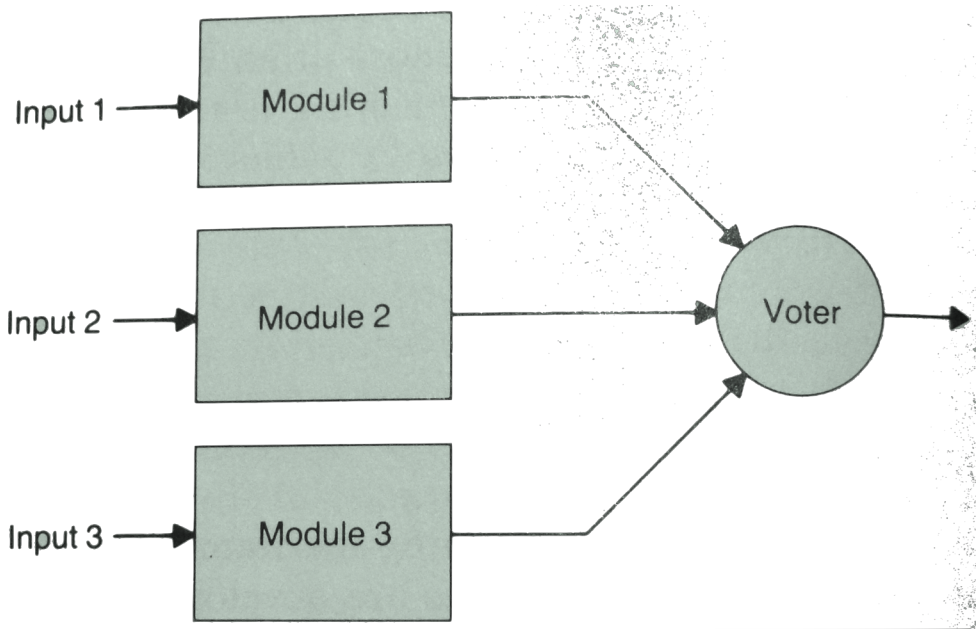
- Redundancy is simply addition of information, resources, or time beyond normal system ops.
  - Hardware redundancy
  - Software redundancy
  - Information redundancy
  - Time redundancy

Hardware Redundancy

- Replication is the most common way of achieving redundancy
- Three basic forms of hardware redundancy:
  - Passive techniques – mask the faults
  - Active techniques – called dynamic method as well – fault tolerance via detection
  - Hybrid techniques – combines attractive features of both passive and active redundancies

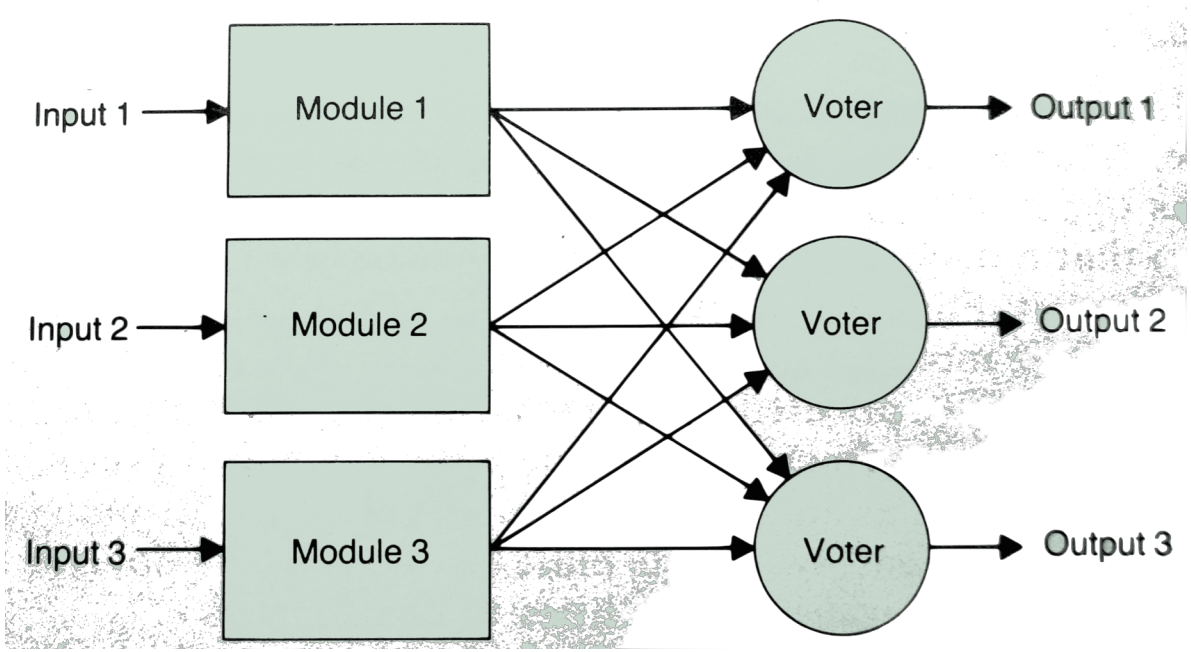
Passive Hardware Redundancy

- Passive hardware redundancy relies on voting mechanisms to mask faults
  - No need to detect which module is at fault
- Triple modular redundancy:



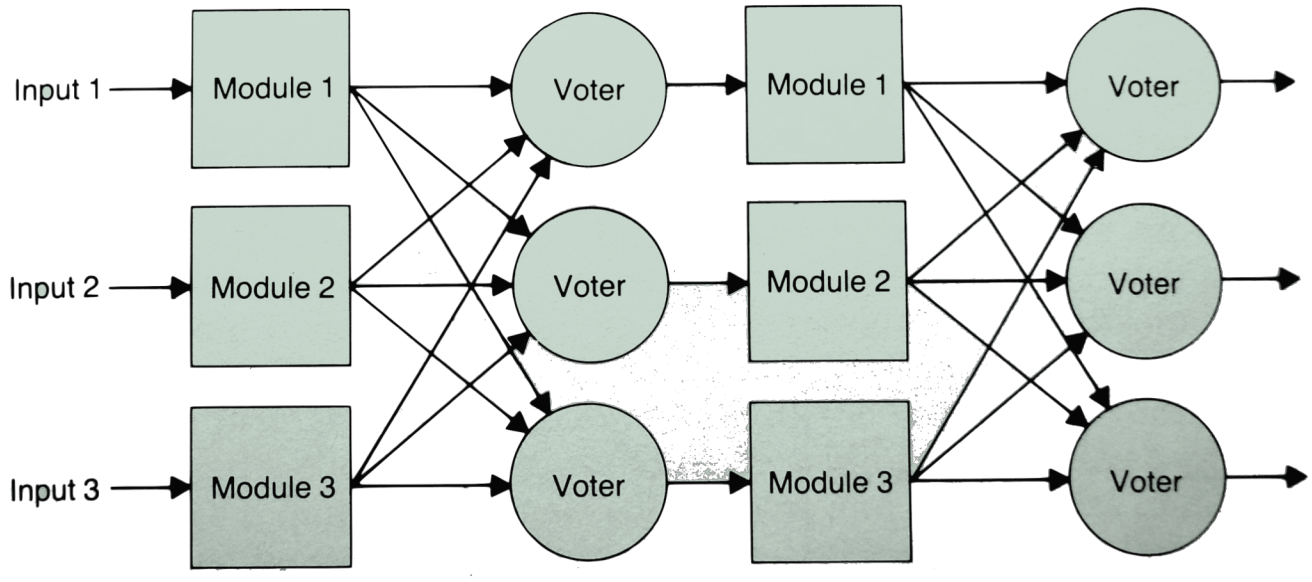
Triple Modular Redundancy .. Alternatives

- Problem with TMR is the voter: single point of failure
- Solution:: Replicate voters
- Each output is correct if no more than one module or input is faulty



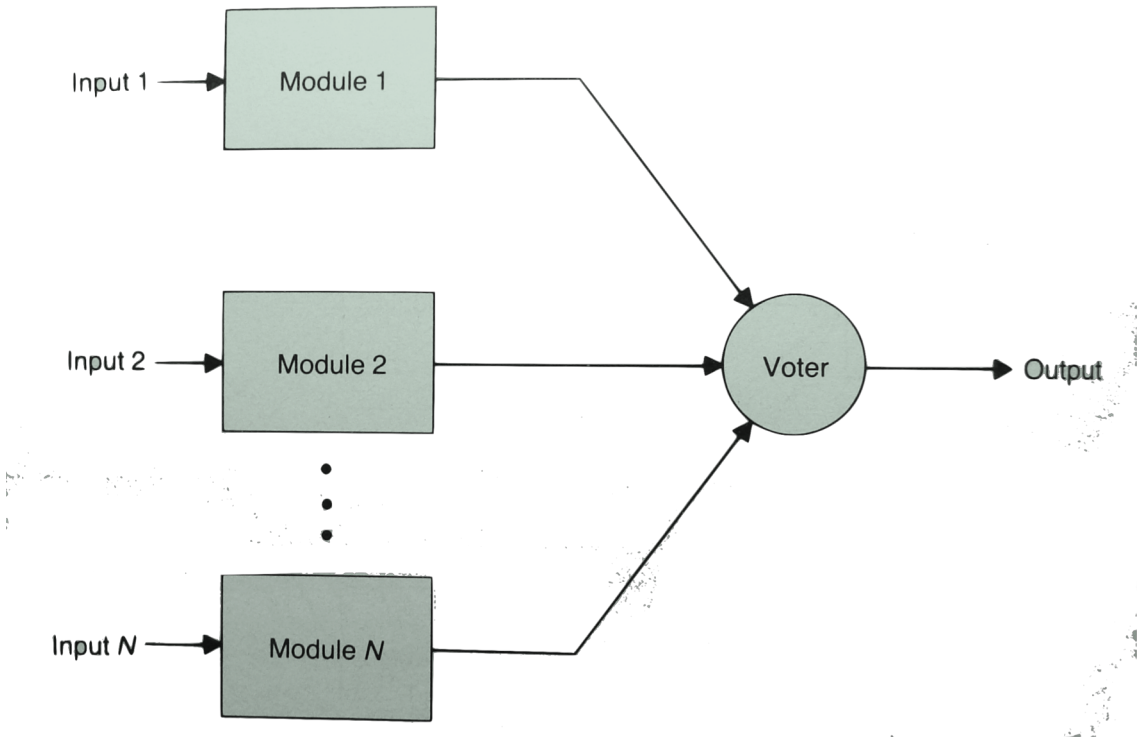
Triple Modular Redundancy.. Alternatives

- Multi-stage TMR systems: voting occurs between each stage to correct errors
- If a voter fails – next stage can correct it – only one module can fail



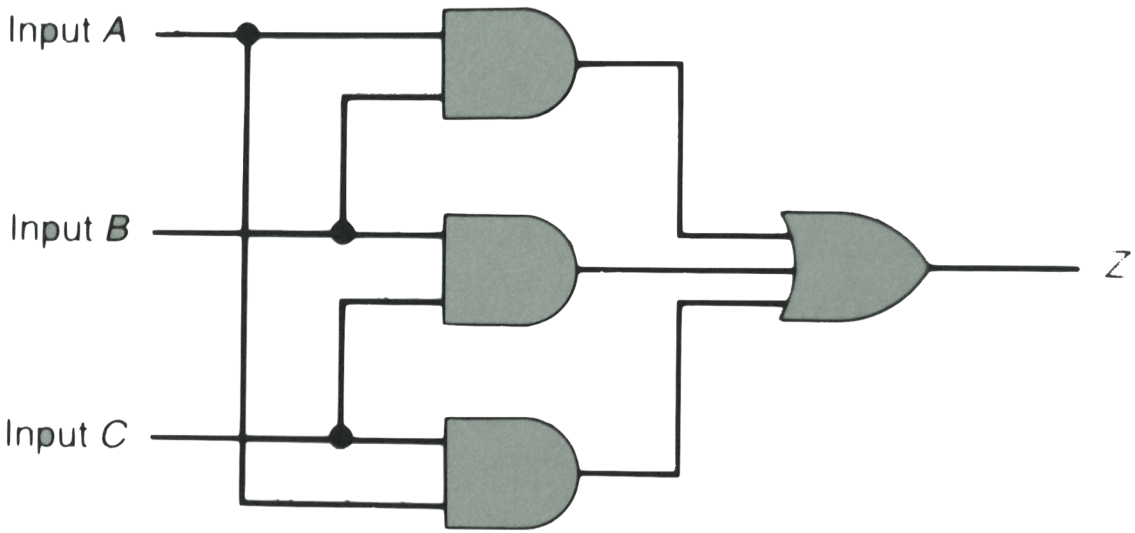
N-Modular Redundancy

- Generalizes TMR to N: N should be odd number so majority voting can work
- Trade-off: Fault tolerance versus hardware cost



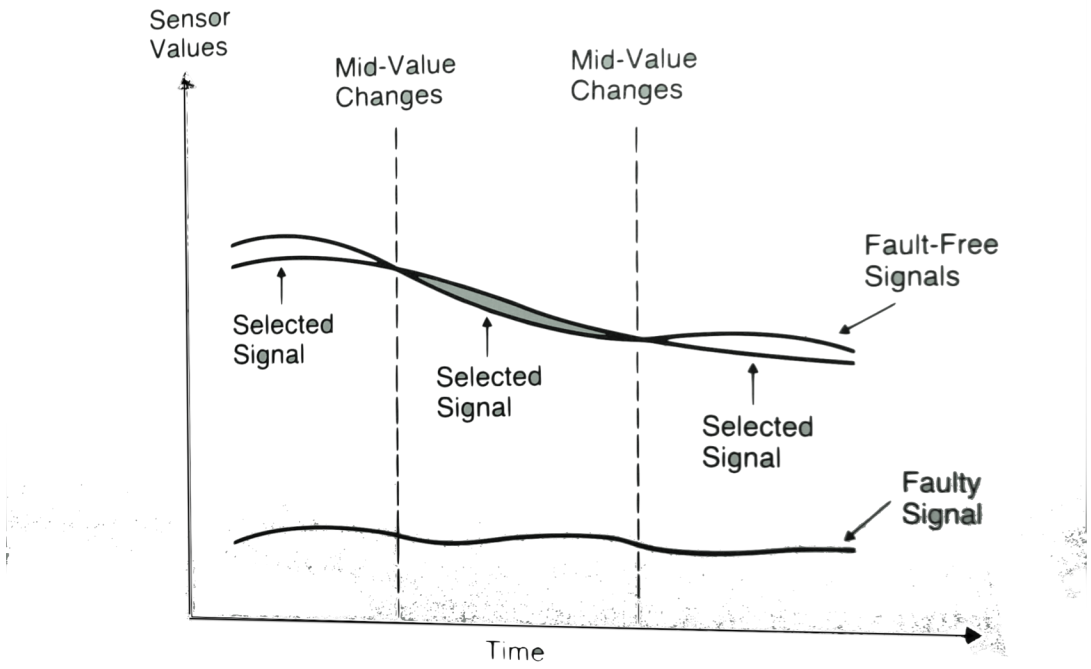
Voting Systems

- Voting closer to "sensors" provides fault containment
- Trade-off: Fault containment versus cost/complexity of system
- A simple majority voter: output is 0 if two inputs are 0 and 1 if two inputs are 1



Voting Systems…

- Signals may not agree even in a fault free environment
- Select "mid signal" value is a good heuristic with odd number of sensors



Active Hardware Redundancy

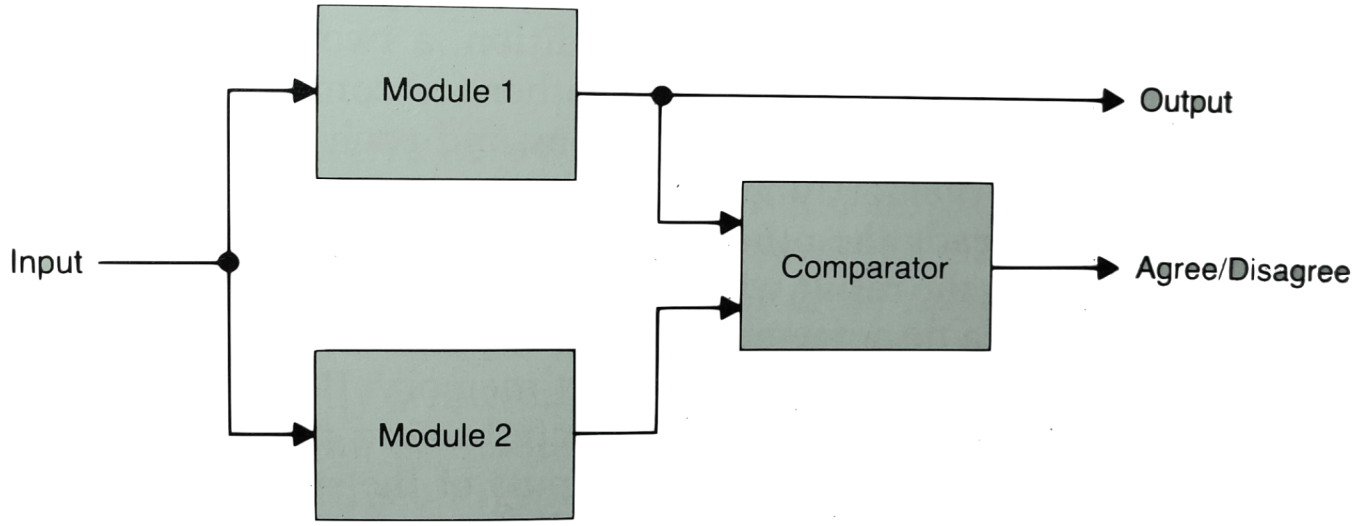
- Attempt to achieve fault tolerance by:
  - Fault detection
  - Fault location
  - And Fault recovery
- Because faults are detected (in many cases) using errors
- We can say fault tolerance is achieved by:
  - Error detection
  - Error location
  - Error recovery

Active Hardware Redundancy

- Active hardware redundancy does not use fault masking
- That is, does not prevent errors due to faults
- Suitable for applications where momentary service lapses are tolerable
- E.g., satellite systems for locationing and tele-communications

Duplication with Comparison

- Simple duplication has two components computing in parallel:
- Disagreement between the two components -> error message generated
- Duplication can detect errors NOT tolerate them

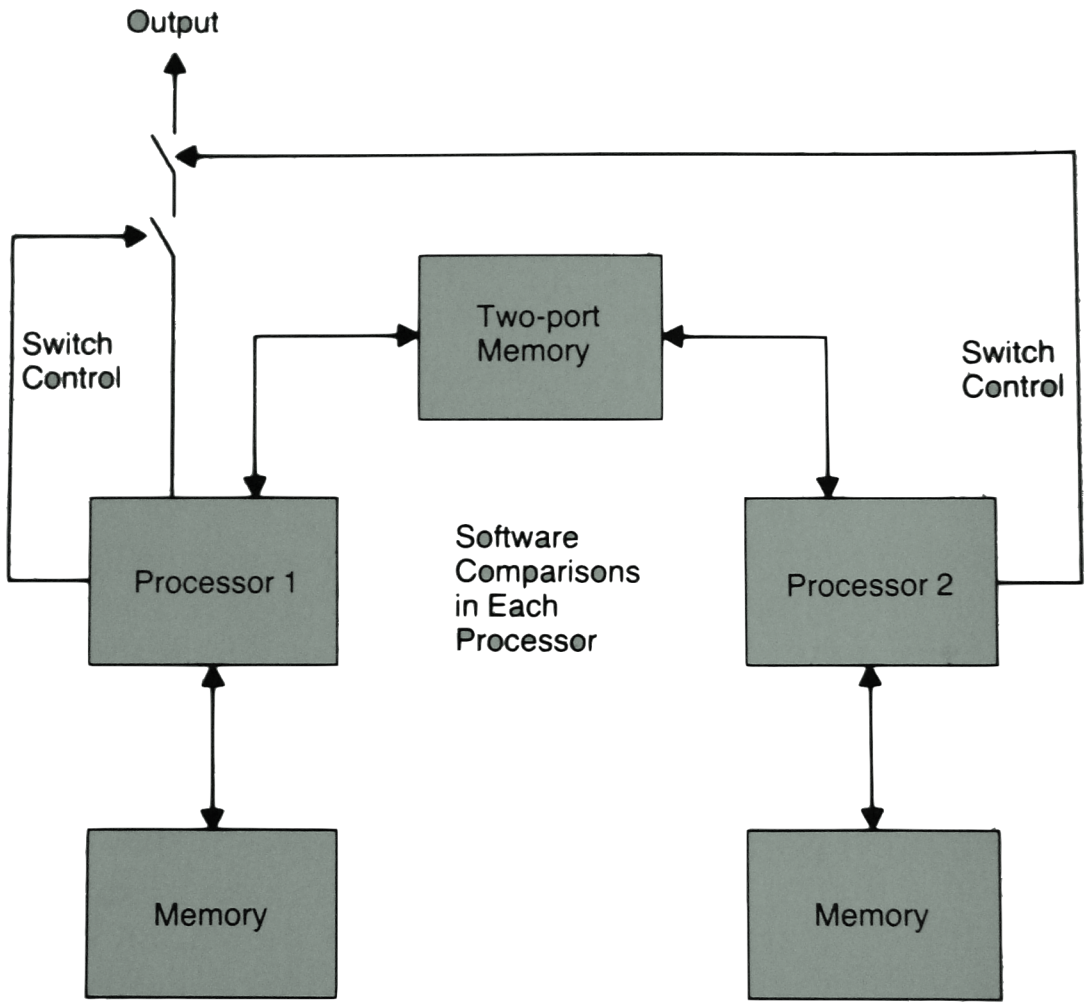


Problems with Duplication with Comp.

- Errors in the input or input lines will not be detected – both modules giving the same output
- Comparator may not perform an exact comparison depending on the application area
- Faulty comparator could give no error signal even when there is an error condition

- Solution: Implement the comparison in software and execute in two redundant processes. The executions need to agree for correct operation. That is either fault or no fault.

Duplicate with Comparison using Software



Standby Sparing

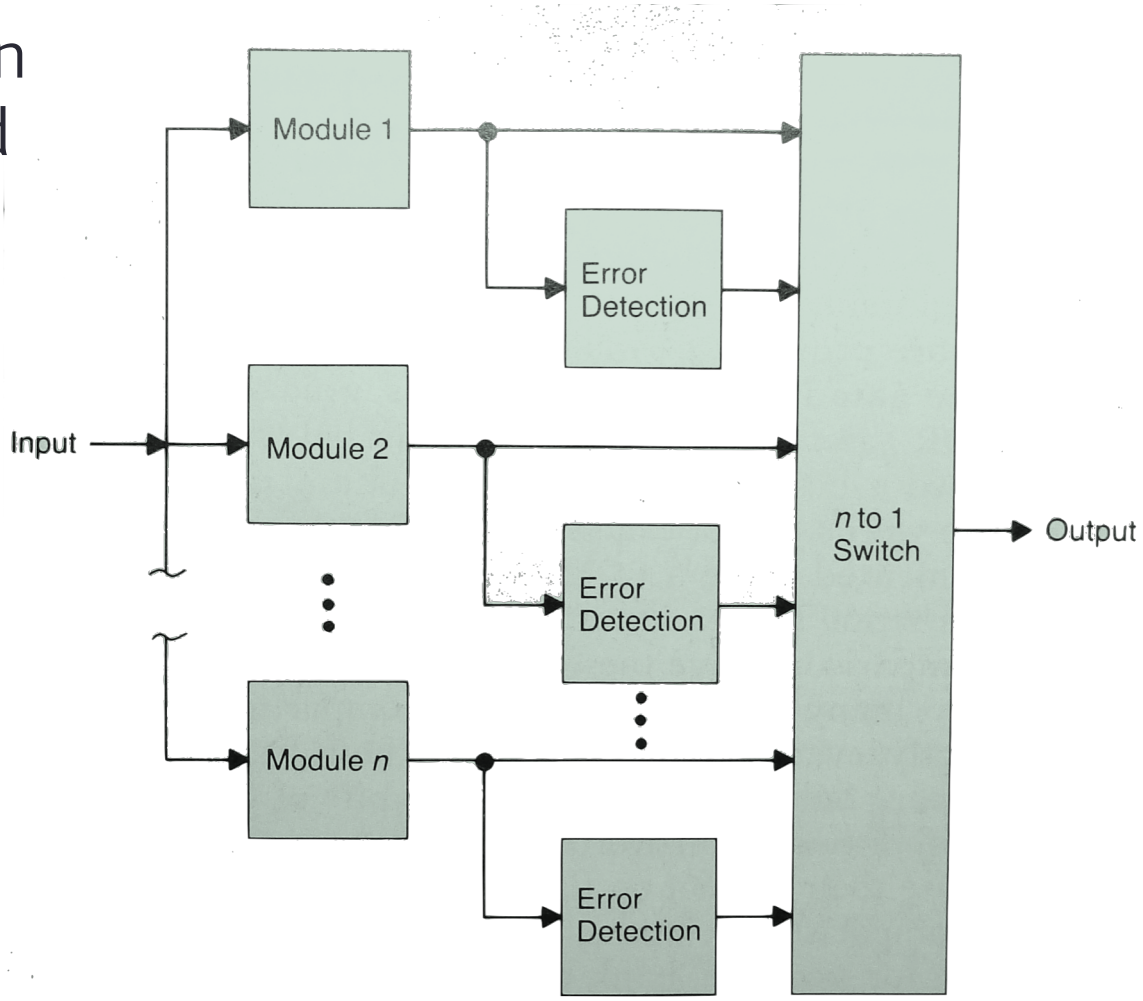
- One module is operational
- One or more modules are spares or standbys
- Different fault/error detection techniques are used to determine when a module becomes faulty
- Two types of standbys:
  - Hot standby: spare module operates synchronously with the active module
  - Cold standby: spares are unpowered until needed to replace a faulty module

Standby Sparing

- Hot standby sparing:
  - Spares operate synchronously with the online module
  - Can take over at any given time with minimal disruption – there will be a short disruption
- Cold standby sparing:
  - Spares are powered down, leading to minimal power use
  - There is longer service disruption at failure
  - Also standby modules may not be subject hazardous operating conditions like power surges
- With P modules in the system (e.g., P parallel processor), we could use k << P modules to provide standby sparing – standby are shared.

Standby Sparing…

- Fault/error detection techniques are used to determine the faulty module
- Reconfiguration => switch output from non faulty module



Pair-and-a-Spare

- Standby sparing + duplication & comparison
- Error or mismatch from comparison used to trigger a reconfiguration
- Another variation is always operate modules in pairs. When two modules do not agree, discard both

Pair-and-a-Spare

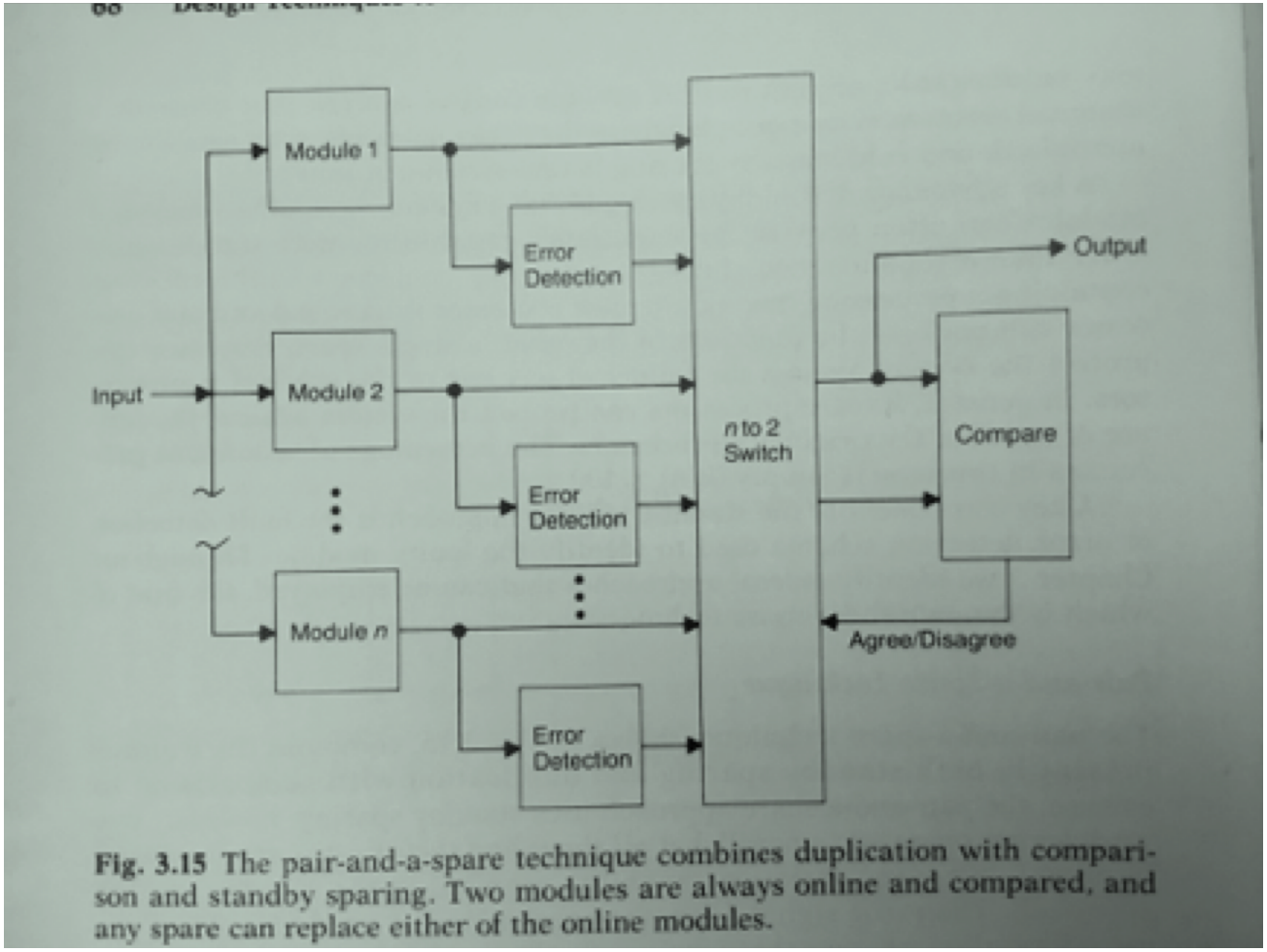


Fig. 3.15 The pair-and-a-spare technique combines duplication with comparison and standby sparing. Two modules are always online and compared, and any spare can replace either of the online modules.

Watchdog Timers

- Watchdog timer is an active form of hardware redundancy
- Watchdog timer should be reset periodically under no fault condition
- It is a good fault detection mechanism

- Assumption: The system is fault free if it has the capability to repetitively perform a resetting function.

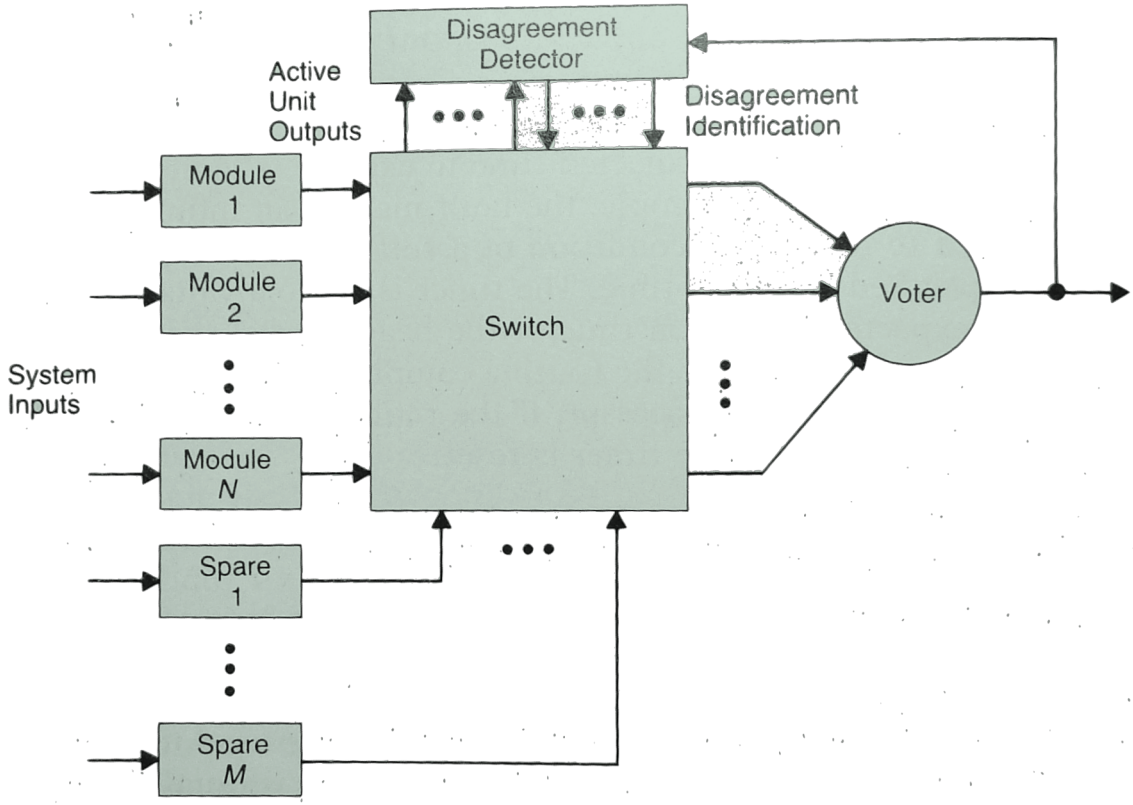
Hybrid Hardware Redundancy

- Passive -> Fault masking
- Active -> Fault detection, location, recovery, and reconfiguration

- Passive + Active -> Hybrid
- Hybrid is usually expensive
- Used in applications requiring high integrity computations

N-Modular Redundancy with Spares

- Spares provided to replace failed modules
- System can operate in basic NMR mode until first failure detection – then switching can take place



Self Purging Redundancy

- All modules are actively connected
- Erroneous module is taken out of the voting process