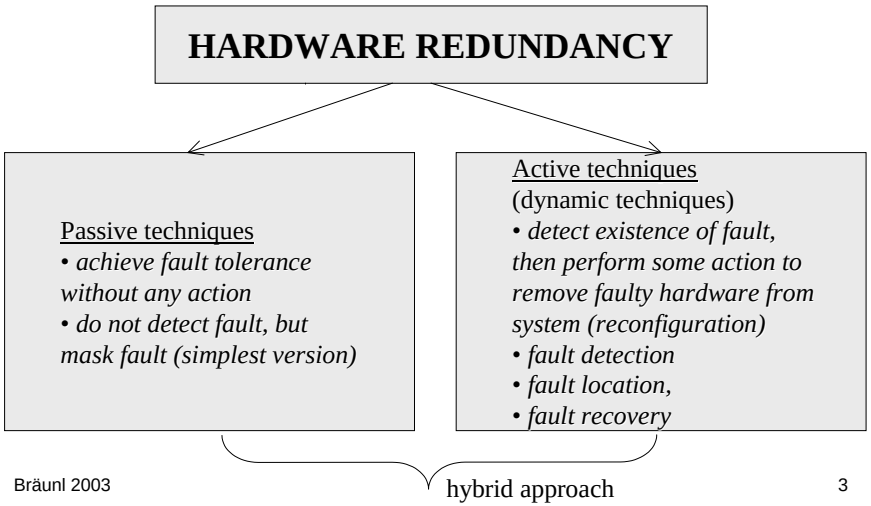
2. Redundancy

2.1. Hardware redundancy
2.2. Information redundancy
2.3. Time redundancy
2.4. Software redundancy

2.1 Hardware Redundancy

- **Idea:** overcome hardware faults by using additional hardware
  - hardware becomes cheap, so approach is feasible

Hardware Redundancy



Hardware Redundancy

Hybrid approach

- is combination of passive + active techniques
- e.g. use fault masking to prevent erroneous results *(prevent temporary errors)* and provide spares to replace faulty hardware *(high reliability)*
- usually expensive

2.1.1 Passive Hardware Redundancy

- Use additional hardware plus voting mechanism to mask occurrence of faults
- E.g. majority voting
- No reconfiguration - inherently fault tolerant

Passive Hardware Redundancy

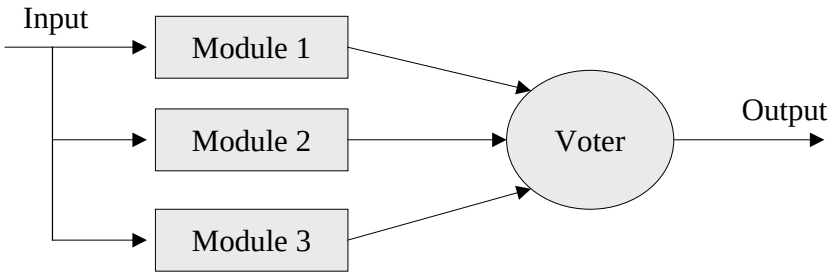
TMR Triple Modular Redundancy

- Use 3 times the hardware + majority vote
- if one system becomes faulty, the two other correct ones mask the fault

Passive Hardware Redundancy

- Assume module can fail
- Make this system fault-tolerant

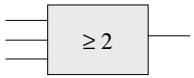
TMR



TMR

Idea:

- All modules receive same input in synch (!) and perform same computation
- Voter does bit-wise comparison



- If one module fails, output is still correct [using only 2 modules does not work]

TMR

Observations:

1. Big problem if voter fails (complete system fails)
   counter argument: voter is very simple circuit and unlikely to fail

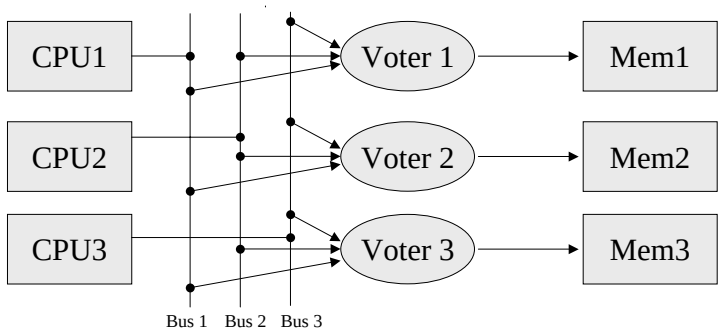
   Nevertheless:
   - Reliability of TMR ≤ reliability of voter
   - "**Single-point-of-failure**" (failure of single system component leads to system failure)

TMR

Observations:

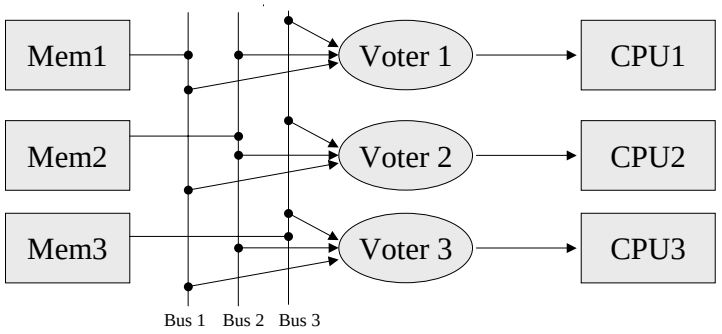
2. Model is not directly applicable to "**standard computer system**"
   - Works only on low-level (bit-level)
   - Bit-wise synchronization is required
   - TMR has no internal state

Extended TMR



Schema for writing to memory

Extended TMR



Schema for reading from memory

Extended TMR

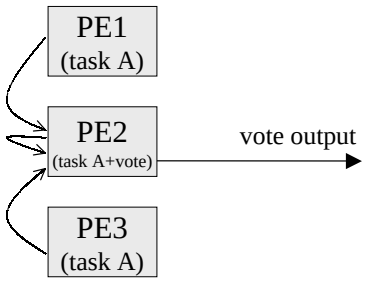
- Schema required for **writing** to memory and **reading** from memory
- Takes care of
  - processor fault
  - voter fault
  - memory fault
  - bus fault
- System has **no single-point-of-failure**
- This approach is implemented in **Tandem Integrity** system

Extended TMR

- Further possibilities for TMR
- Implement voter in *hardware* or *software* ?
- Voter in software:
  - more flexible +
  - longer execution time –
  - less extra hardware +

Extended TMR

- Example: voter in software



Extended TMR

Generalization of TMR

- More than 3 modules, e.g. **5MR**
- In general: **NMR**
  (always odd number)

Embedded Systems and TMR

Before: General purpose computer systems with TMR

Now: Embedded System

- Even more options for voting
- E.g. sensor input
  - Triplicate sensors and vote on sensor data
  - E.g. car: three independent crash sensors for airbag

Embedded Systems and TMR

In general: voting can be done on digital or analog data

- Application: temperature measurement
- Method: take 3 measurements, compute **median** value
- Example
  - Sensor 1: 99°C
  - Sensor 2: 100 °C
  - Sensor 3: 45,217 °C

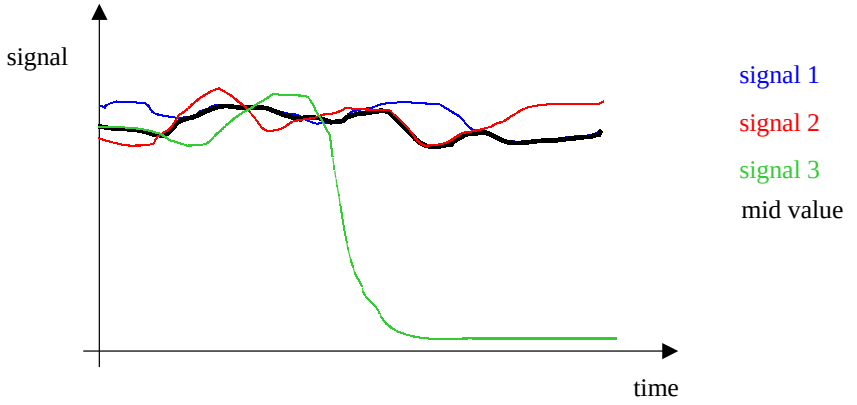
Embedded Systems and TMR

- Example
  - Sensor 1: 99°C
  - Sensor 2: 100 °C
  - Sensor 3: 45,217 °C ← **Error**
- Median (99, 100, 45217) = 100 = Sensor2
- Average(99, 100, 45217) = 15139

Embedded Systems and TMR

- This median method is also called "Mid-Value-Select"
- In similar way used at Olympic figure skating events!
  From 10 votes, lose highest and lowest vote,
  then average the remaining 8 votes

Mid-Value-Select

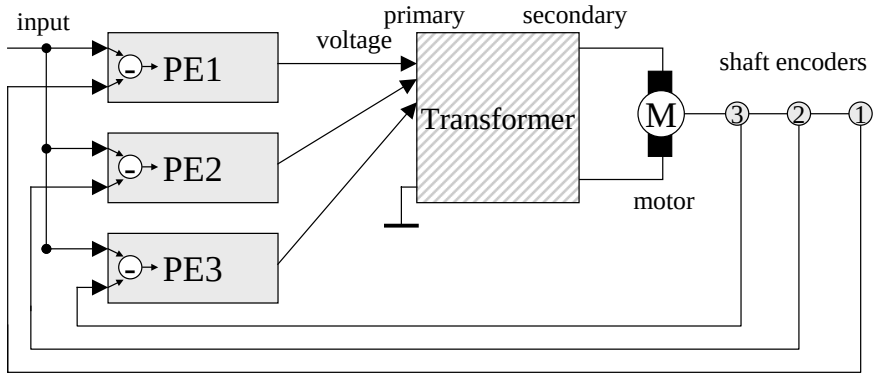


signal 3 becomes corrupted

Flux-Summing

- "Mid-Value-Select" works well for multiple sensors (inputs)
- What about actuators?
  - many systems have multiple sensors, but only single actuator
  - single control signal must be generated
- Possible solution: Flux-Summing Technique
  - e.g. to control motor at a specified speed

Flux-Summing



feedback 1..3

Flux-Summing

Principle

- transformer adds output voltages (separate windings)
- if one PE fails, voltage at transformer primary will change, e.g. **drop**
- motor will then slow down
- feedback loop then indicates too low speed
- remaining 2 PEs will increase output to correct motor speed (e.g. PID controllers)
- **motor back at correct speed!**

Flux-Summing

Principle

- Flux-Summing does **not** use voting like TMR
- Still: Flux-Summing has similar effect of fault masking as TMR

2.1.2 Active Hardware Redundancy

1. Fault detection
2. Fault Location
3. Fault recovery

No fault masking

- Fault may occur
- Erroneous result must be acceptable in application
- After detection, system is reconfigured and returns to original status

Active Hardware Redundancy

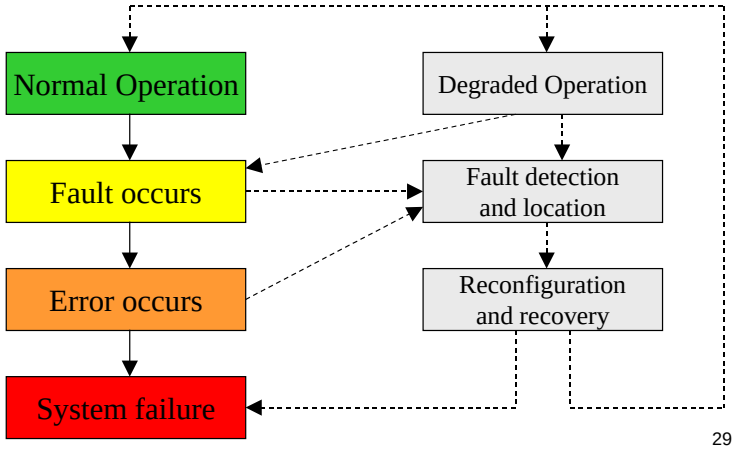
Example:

Satellites

- Infrequent temporary failures are acceptable, as long as reconfiguration is possible
- Cheaper solution than passive redundancy

Active Hardware Redundancy

System Status Diagram

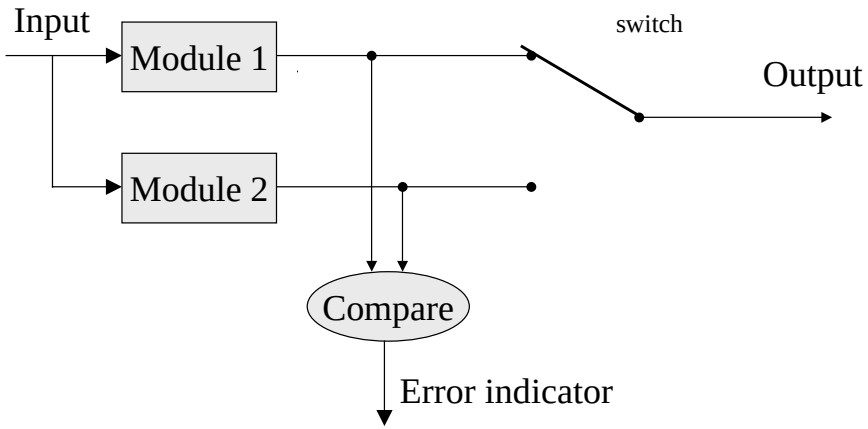


Duplication with Comparison

- Run 2 hardware modules in parallel with same computation
- Compare the results
- If not equal:
  - Fault detected
  - Action may be taken from outside to switch
    (It cannot be determined automatically which unit is faulty.)

- This approach is used in the **Stratus System**
  (similar to “2MR”)

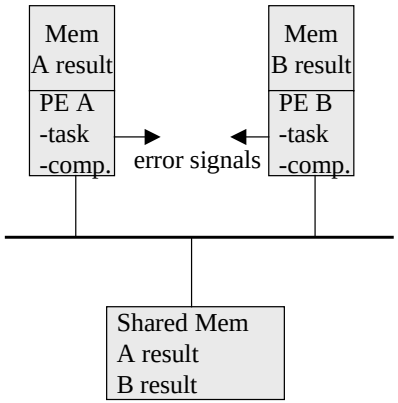
Duplication with Comparison



Duplication with Comparison

Software implementation
in parallel system

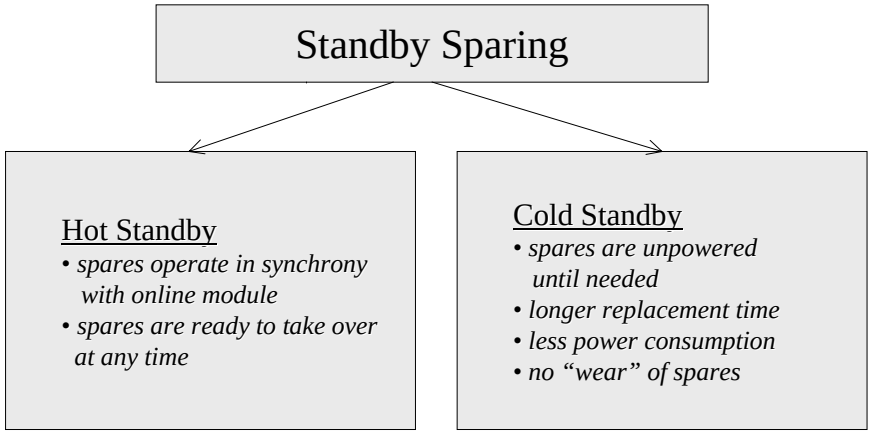
- Compute task parallel in PE A and PE B
- Copy local result to shared memory
- Read other PE’s result from shared memory and compare
- Error signal if not equal



Standby Sparing

- Also called: Standby replacement
- One module is operational
- One or more modules are standbys (spares)
- Use fault detection and localization
- When fault is detected, remove faulty module from operation and replace by spare

Standby Sparing



Standby Sparing

- Hot Standby
  - Sample application: Production plant (e.g. chemical)
  - Dominating factor: exchange speed
- Cold Standby
  - Sample application: Space system
  - Dominating factor: minimal power consumption

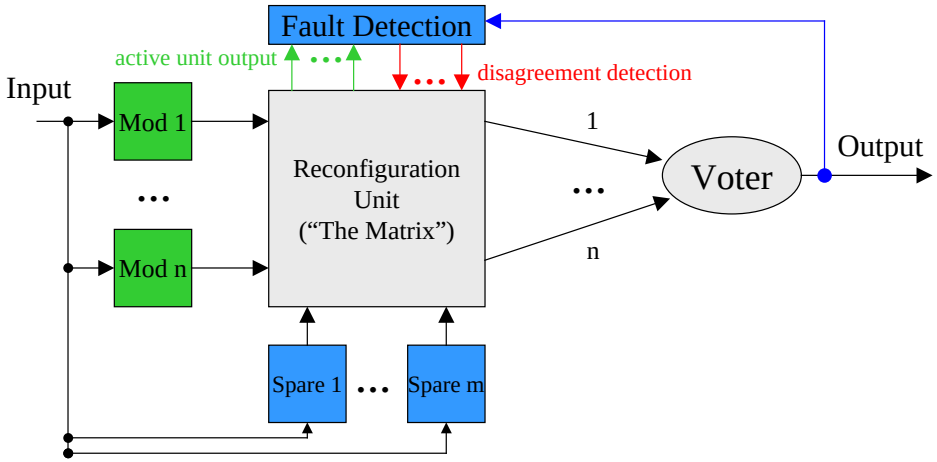
Pair-And-A-Spare

- Run 2 modules in parallel (as before)
- When error is signaled, replace faulty module with spare (spares are not used until needed)
- This technique is used in the **Stratus System**

NMR with Spares

- A common approach of hybrid hardware redundancy
- NMR guarantees fault masking
- Spares are used to restore fault-free state

NMR with Spares



Triple-Duplex

- A variation of NMR with spares
- Use TMR with each module as a duplex
- Duplication with comparison allows:
  - Fault detection
  - Fault localization
  - Replacement of faulty module with spare

Triple-Duplex