Space Shuttle

◆ **Based on:**

− Redundancy Management Technique for Space Shuttle Computers, by Sklaroff, J., R., IBM Journal on Research and Development, Vol. 20, No. 1, pp. 20-28, January 1976.

◆ **General Purpose Computer (GPC)**

− First operational use of *off-the-shelf* computers.

− First planned operational use of multiple simplex computers for fault-tolerant purposes.

− Concept fight-proven in Tactical Aircraft Guidance research and development program (TAGS).

  » TAGS: TMR helicopter fight control system

− Non-redundant (internal) processors in a fight-critical application.

Space Shuttle

◆ **Previous space program**

– internally redundant computers

  » considerable extra circuitry to provide fault-tolerance

  » fault detection using logical comparisons at selected points in the data flow within the computer

– prime-backup configuration

– Saturn: redundancy at a modular level within the computer

– Skylab: dual computer in active/standby mode (in-orbit), relying on self-test techniques. Redundant hardware to switch to standby computer (2.75s switch-over time).

Space Shuttle

◆ **Periods of Risk:**

- Boast Phase

- Reentry Phase

- Landing Phase

◆ **System Architecture:**

- 5 Computers consisting of CPU/IOP pair

- 28 Redundant interconnections busses

- Critical operation:

  » 4-NMR for flight critical operations (voting)

  » 5 th for non-critical operations

Space Shuttle

- ◆ General Purpose Computer (GPC)
  - » Each GPC performs about 325000 ops/s during critical phases
  - » About 440 synchronization and cross-checks per second
  - » Physically distributed in 3 avionics bays
  - » Air cooled by 1 of 2 fans/bay (600W/GPC)
  - » CPU IBM AP-101B (AP-101S upgrade) 32bit processor
  - » IOP
    - ■ Transformer-coupled to busses, which in turn are transformer-coupled to multiplexer/demultiplexer units (MDM). MDM contain A/D and D/A converters, interface with analog subsystems, e.g. flight control sensors and effectors.
    - ■ 24 bi-directional bus channels
    - ■ 1-MHz serial data
  - » Each individual channel
    - ■ Command Mode (can read/write to bus)
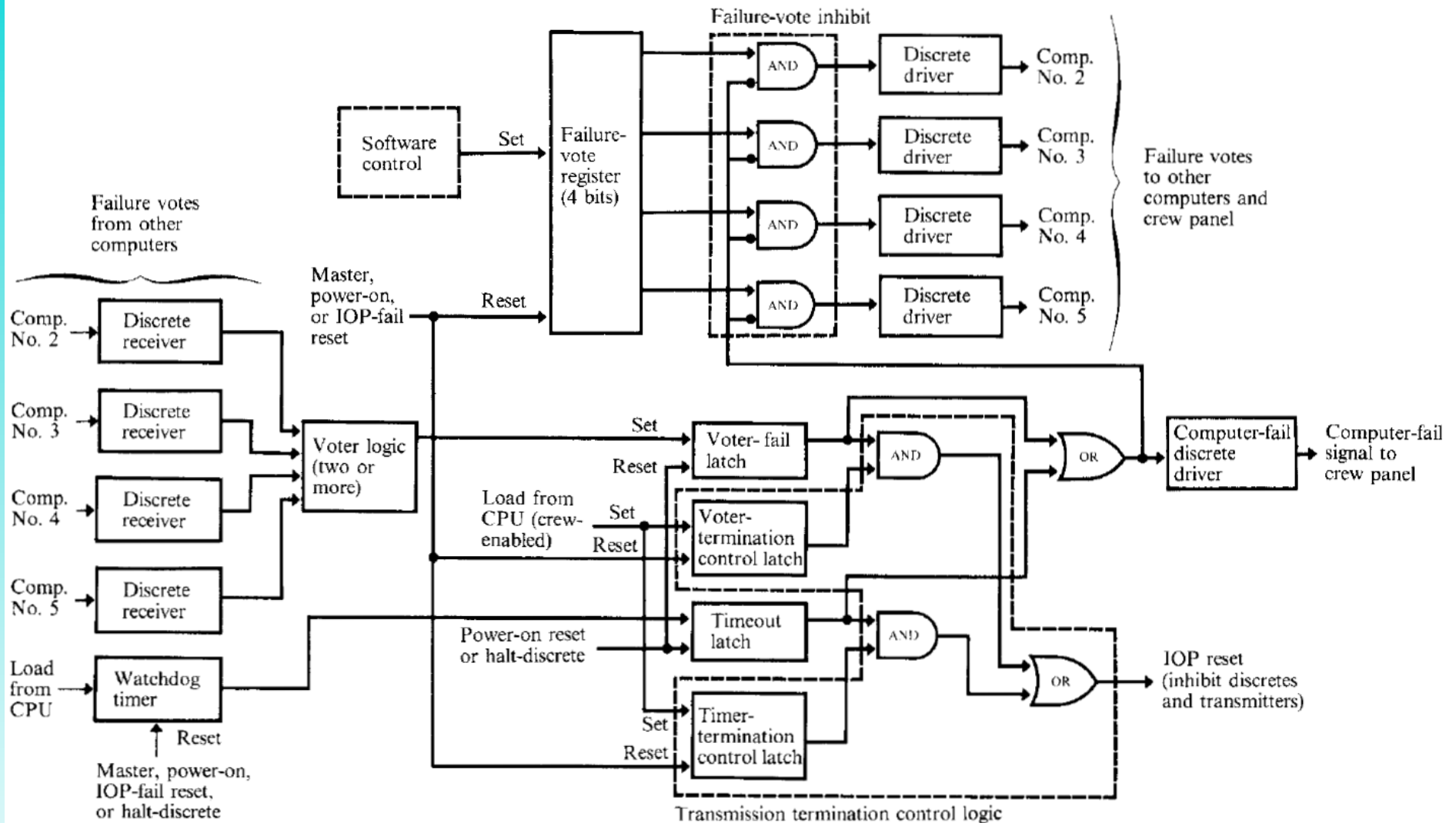   - ■ Listen Mode (can only read from bus)

Space Shuttle

Failure-vote inhibit

Failure votes
to other
computers and
crew panel

Failure votes
from other
computers

Comp. No. 2 → Discrete receiver

Comp. No. 3 → Discrete receiver

Comp. No. 4 → Discrete receiver

Comp. No. 5 → Discrete receiver

Load from CPU → Watchdog timer

Reset
Master, power-on, IOP-fail reset, or halt-discrete

Software control — Set → Failure-vote register (4 bits)

Master, power-on, or IOP-fail reset — Reset

AND → Discrete driver → Comp. No. 2

AND → Discrete driver → Comp. No. 3

AND → Discrete driver → Comp. No. 4

AND → Discrete driver → Comp. No. 5

Voter logic (two or more)

Set
Reset — Voter-fail latch

Load from CPU (crew-enabled) — Set
Reset — Voter-termination control latch

Power-on reset or halt-discrete — Timeout latch

Set
Reset — Timer-termination control latch

AND

AND

OR → Computer-fail discrete driver → Computer-fail signal to crew panel

OR → IOP reset (inhibit discretes and transmitters)

Transmission termination control logic

Figure 4 Dedicated redundancy management logic, shown for computer 1.

Space Shuttle

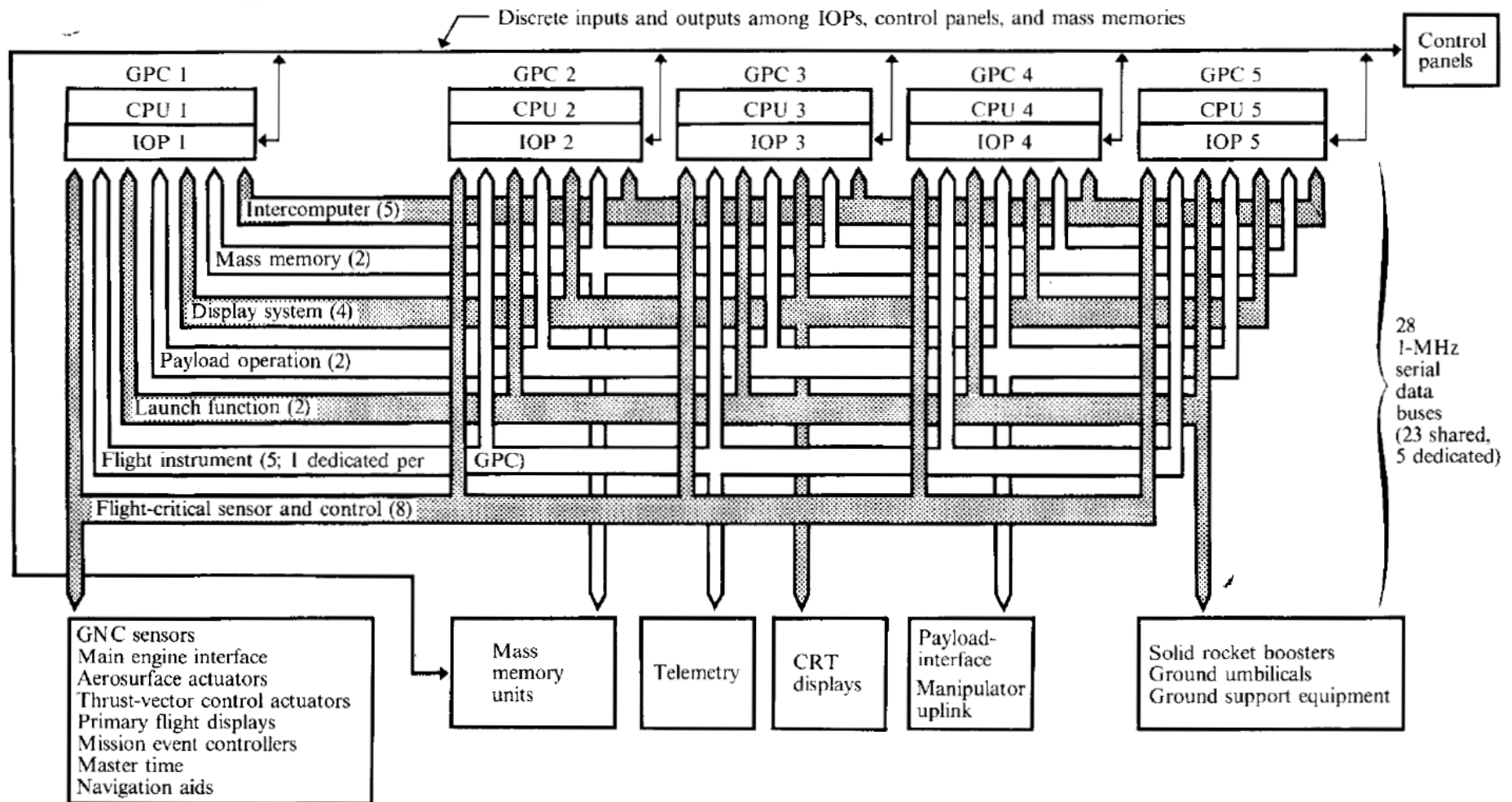Discrete inputs and outputs among IOPs, control panels, and mass memories

Figure 1   Space Shuttle avionics system block diagram.

Space Shuttle

◆ Busses

» Total 28 busses in 7 groups, grouped by function.

» Subsystems have varying levels of redundancy at the unit level depending on their criticality.

■ e.g. 3 inertial measurement units, 2 radar altimeters, 4 air data transducer assemblies.

» To prevent loss of more than one redundant units *no* two redundant units interface to the same bus.

» Each Unit attached to the bus is addressable by a *command word*.

» Some systems are internally redundant, e.g. hand controllers, actuators for main engine, main engine interface unit, mission event controllers.

» These subsystems receive redundant commands on separate input channels.

» Use internal algorithm to generate one output. Algorithms detect incorrect commands.

Space Shuttle

◆ Actuator Voting Example: Aerosurface actuator

  » 4 independent servo channels driving a 4-element force-summed actuator.

  » Failure of any 3 of the 4 channels can be tolerated.

  » Hydraulic fault detection is provided by sensing pressure differential.

  » If more than 2 channels are fault free

  ■ faulty modules are automatically bypassed.

  » If only 2 channels operating

  ■ pressure difference is detected

  ■ resulting *standoff* is resolved by manual reset.

  » Actuator Voting

  ■ allows computer to transmit incorrect commands to critical subsystems for an indefinite number of cycles without causing adverse effects.

  ■ significantly relaxes fault detection time constraints

Space Shuttle

◆ System Operation

   – 4MR configuration => Redundant Set.

      » Inter-Computer Channel (ICC)

         ■ 5 busses

         ■ 1 bus operates in command mode (one IOP permanently assigned command mode)

         ■ 4 remaining are in listening mode

      » Flight-Critical Sensor and Control-Data-Bus

         ■ 2 subgroups of 4 busses

         ■ one IOP in redundant set assigned command mode in each subgroup

         ■ 3 remaining are in listening mode

Space Shuttle

◆ **Data Collection**

- Inputs
    - » each redundant subsystem connected to different bus.
    - » "Command" computer requests data from all subsystems
    - » Returned Data available to all 4 computers (listen mode)
- Data Outputs
    - » each IOP connected to different bus
    - » each computer sends data to each of the voter-effector channels
    - » each computer listens to the command sent out by other computers

Space Shuttle

◆ Communication

   – Inter-Computer Communication

      » uses Inter-computer group

      » common cross checking

      » error messages (accusations)

   – Synchronization

      » needed to avoid command divergence

      » task level synchronization

      » timing-skew, data-skew: synchronization via

         ■ inter-computer discrete signals

         ■ synchronization software

Space Shuttle

◆ **Redundancy Management: Requirements**

– Faulty computer identified with 96 % coverage before assignment to the redundant set (extensive preliminary diagnostics)

  » before lift-off

  » before critical in-orbit phase

– Of 2 GPCs in the redundant set

  » first 2 sequential faults must be automatically identified to the crew

  » best practical diagnosis of third fault (1 of 2)

– System capable of automatically inhibiting all transmissions from faulty processor

  » disabled at power-up

  » enabled by crew

– No computer failure should cause another GPC to

  » identify itself as failed

  » generate an incorrect output

– Restoration of GPC excluded due to transient fault by crew in non-critical phase (Figure 3)
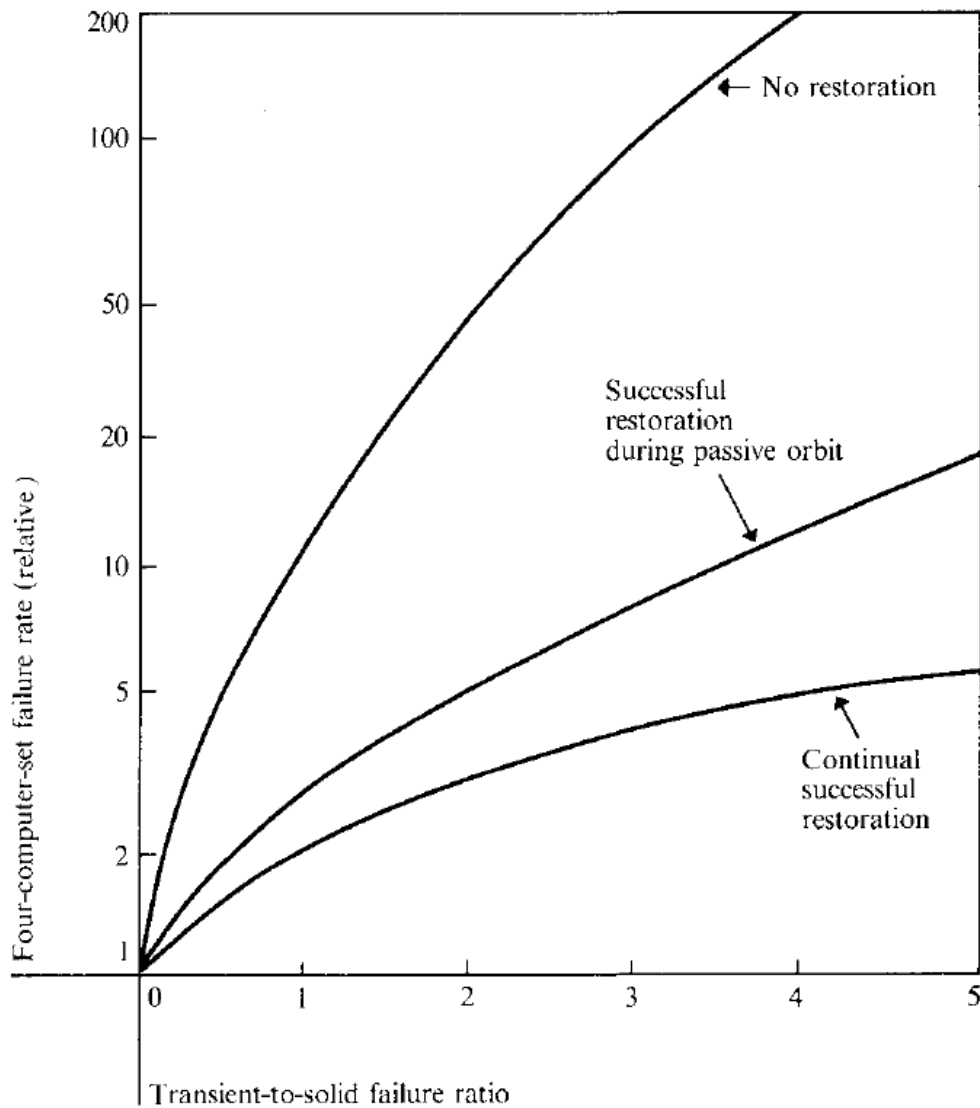
Figure 3 Effect of GPC transient failures on the computer-set failure rate as a function of restoration policy, in relation to the zero-transient failure rate. The operational-hour equivalents are 6.6, ascent; 128, passive orbit; 43, active orbit; and 4.4, reentry/landing.

Space Shuttle

◆ **Design Considerations for Redundancy Management**

  – Primary detection is Cross-Checking, using command and listen modes, process synchronization and grouping of busses

    » Minimize the need to depend on the computer to detect and identify its own failure => use special hardware

    » Use software to judge the "health" of other computers.

    » Sum-checking of critical output => check-sum signature and compare signature word.

      ■ Motivated by insufficient processor power to compare all output.

    » Transmit and compare word over Inter-computer buses. Then use built-in-test-equipment (BITE) in each IOP to perform comparisons.

    » Use test programs to generate faults which test BITE logic. (Guard the guardians)!