Hardware Design: Fault Tolerant Architectures

Prof. Chris Johnson,
School of Computing Science, University of Glasgow.
johnson@dcs.gla.ac.uk
http://www.dcs.gla.ac.uk/~johnson

University of Glasgow

Introduction: Hardware Design

- Fault Tolerant Architectures.
- Basics of hardware management.
- Fault models.
- Hardware redundancy.
- Space Shuttle GPC Case Study.

University of Glasgow

Parts Management Plan

- MIL-HDBK-965
  - help on hardware acquisition.
- General dependability requirements.
- Not just about safety.
- But often not considered enough...

University of Glasgow

The Basics: Hardware Management

- MIL-HDBK-965: Acquisition Practices for Parts Management
  - Preferred Parts List
  - Vendor and Device Selection
  - Critical Devices, Technologies & Vendors
  - Device Specifications
  - Screening
  - Part Obsolescence
  - FRACAS:
  - Failure Reporting, Analysis & Corrective Action

University of Glasgow

Types of Faults

- Design faults:
  - erroneous requirements;
  - erroneous software;
  - erroneous hardware.
- These are systemic failures;
  - not due to chance but design.
- Don’t forget management/regulators!

University of Glasgow

Types of Faults

- Intermittent faults:
  - fault occurs and recurs over time;
  - faulty connections can recur.
- Transient faults:
  - fault occurs but may not recur;
  - electromagnetic interference.
- Permanent faults:
  - fault persists;
  - physical damage to processor.
-

University of Glasgow

Fault Models

- Single stuck-at models.
- Hardware seen as `black-box'.
- Fault modelled as:
  - input or output error;
  - stuck at either 1 or 0.
- Models permanent faults.

University of Glasgow

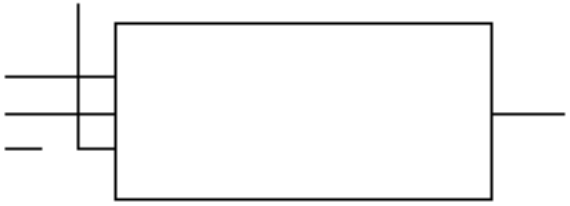
Fault Models - Single Stuck-At...

Fault-free

Single stuck-at fault modes

Input connected to 1 or 0

Input connected to 1 or 0

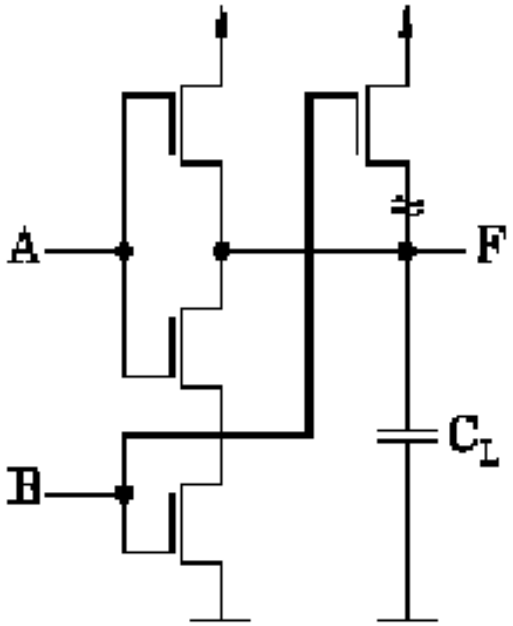


Input connected to 1 or 0

Output connected to 0 or 1

University of Glasgow

Fault Models



A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

CMOS - NAND Gate - fault detection on:
(A1, B1, A1, B0), (A1, B1, A0, B1) and (A1, B1, A0, B0).

University of Glasgow

Fault Models

- Bridging Model:
  - input not `stuck-at' 1 or 0;
  - but shorting of inputs to circuit;
  - input then is wired-or/wired-and.
- Stuck-open model:
  - both CMOS output transistors off;
  - results is neither high nor low...
- Transition and function models.

University of Glasgow

Software Faults (Aside...)

- Much more could be said...
  - see Leveson or Storey.
- Huge variability:
  - specification errors;
  - coding errors;
  - translation errors;
  - run-time errors...

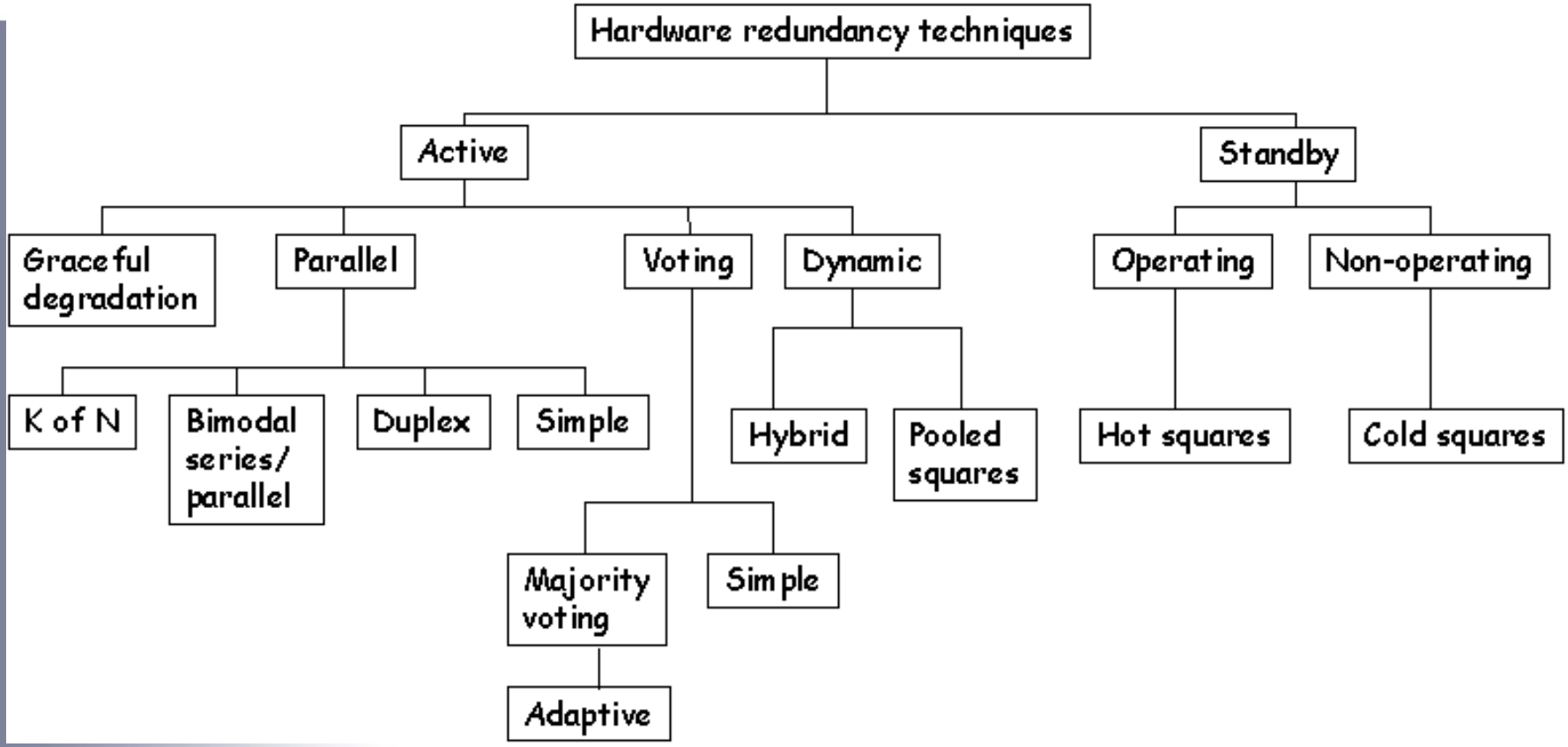
University of Glasgow

Hardware Redundancy

- Adds:
  - cost; weight; power consumption;
  - complexity (most significant).
- These can outweigh safety benefits.
- Other techniques available:
  - improved maintenance;
  - better quality materials;
- Sometimes no choice (Satellites).

University of Glasgow

Redundancy Techniques



University of Glasgow

Active Redundancy

- When component fails...
- Redundant components do not have:
  - to detect component failure;
  - to switch to redundant resource.
- Redundant units always operate.
- Automatically pick up load on failure.

University of Glasgow

Standby Redundancy

- Must detect failure.
- Must decide to replace component.
- Standby units can be operating.
- Stand-by units may be brought-up.

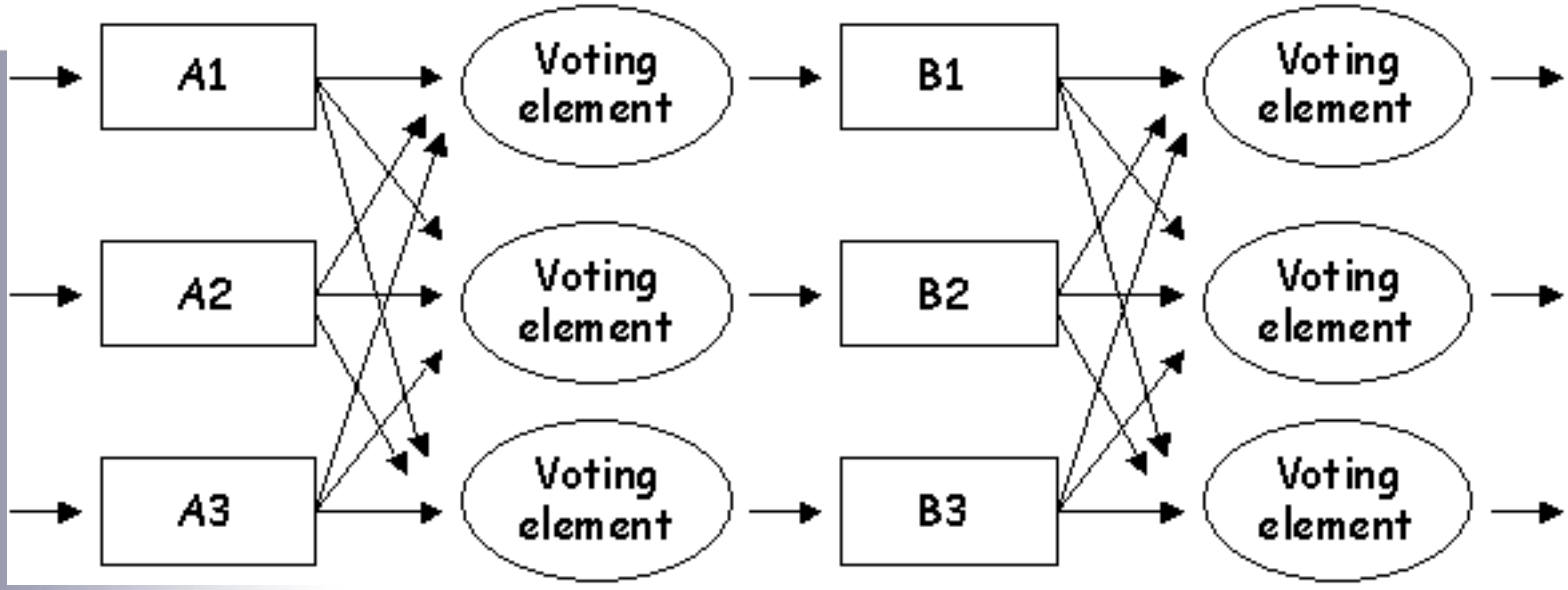
University of Glasgow

Triple Modular Redundancy (TMR)

- Possibly most widespread.
- In simple voting arrangement,
  - voting element -> common failure;
  - so triplicate it as well.
- Multi-stage TMR architectures.
- More cost, more complexity...

University of Glasgow

Multilevel Triple Modular Redundancy (TMR)



- No protection if 2 fail per level.
- No protection from common failure
  - eg if hard/software is duplicated.

University of Glasgow

Fault Detection

- Functionality checks:
  - routines to check hardware works.
- Signal Comparisons:
  - compare signal in same units.
- Information Redundancy:
  - parity checking, M out of N codes...
- Watchdog timers:
  - reset if system times out.
- Bus monitoring:
  - check processor is `alive'.
- Power monitoring:
  - time to respond if power lost.


WINDOW JETTISON HANDLE PIN
REMOVE BEFORE JETTISON


GPC STATUS
FAILED GPC
1
2
3
4
5
ON
OFF
01

Space Shuttle General Purpose Computer: GPC

GPCs running together in the same GN&C (Guidance, Navigation and Control) OPS (Operational Sequence) are part of a redundant set performing identical tasks from the same inputs and producing identical outputs.

Therefore, any data bus assigned to a commanding GN&C GPC is heard by all members of the redundant set (except the instrumentation buses because each GPC has only one dedicated bus connected to it).

These transmissions include all CRT inputs and mass memory transactions, as well as flight-critical data. Thus, if one or more GPCs in the redundant set fail, the remaining computers can continue operating in GN&C. Each GPC performs about 325,000 operations per second during critical phases. "

http://spaceflight.nasa.gov/shuttle/reference/shutref/orbiter/avionics/dps/software.html

University of Glasgow

Space Shuttle General Purpose Computer: GPC

``GPC status information among the primary avionics computers. If a GPC operating in a redundant set fails to meet two redundant multiplexer interface adapter receiver during two successive reads of response data and does not receive any data while the other members of the redundant set do not receive the data, they in turn will vote the GPC out of the set. A failed GPC is halted as soon as possible."

``GPC failure votes are annunciated in a number of ways. The GPC status matrix on panel O1 is a 5-by-5 matrix of lights. For example, if GPC 2 sends out a failure vote against GPC 3, the second white light in the third column is illuminated. The yellow diagonal lights from upper left to lower right are self-failure votes. Whenever a GPC receives two or more failure votes from other GPCs, it illuminates its own yellow light and resets any failure votes that it made against other GPCs (any white lights in its row are extinguished). Any time a yellow matrix light is illuminated, the GPC red caution and warning light on panel F7 is illuminated, in addition to master alarm illumination, and a GPC fault message is displayed on the CRT. "

Space Shuttle General Purpose Computer: GPC

“Each GPC power on , off switch is a guarded switch. Positioning a switch to on provides the computer with triply redundant normally, even if two main or essential buses are lost. "

“(There are) 5 identical general-purpose computers aboard the orbiter control space shuttle vehicle systems. Each GPC is composed of two separate units, a central processor unit and an input/output processor. All five GPCs are IBM AP-101 computers. Each CPU and IOP contains a memory area for storing software and data. These memory areas are collectively referred to as the GPC's main memory. The IOP of each computer has 24 independent processors, each of which controls 24 data buses used to transmit serial digital data between the GPCs and vehicle systems, and secondary channels between the telemetry system a nd units that collect instrumentation data. The 24 data buses are connected to each IOP by multiplexer interface adapt ers that receive, convert and validate the serial data in response to discrete signals calling for available data to be transmitted or received from vehicle hardware."

Space Shuttle General Purpose Computer: GPC

``A GPC on orbit can also be "freeze-dried;" that is, it can be loaded with the software for a particular memory configuration and then moded to standby. It can then be moded to halt and powered off. Since the GPCs have non-volatile memory, the software is retained. Before an OPS transition to the loaded memory configuration, the freeze-dried GPC can be moded back to run and the appropriate OPS requested.

A failed GPC can be hardware-initiated, stand-alone-memory-dumped by switching the powered computer to terminate and halt and then selecting the number of the failed GPC on the GPC memory dump rotary switch on panel M042F in the crew"

http://spaceflight.nasa.gov/shuttle/reference/shutref/orbiter/avionics/dps/software.html

Space Shuttle General Purpose Computer: GPC

``A simplex GPC is one in run and not a member of the redundant set, such as the BFS (Backup Flight System) GPC. Systems management and payload major functions are always in a simplex GPC."

``Even though the four primary avionics software system GPCs control all GN&C functions during the critical phases of the mission, there is always a possibility that a generic failure could cause loss of vehicle control. Thus, the fifth GPC is loaded with different software created by a different company than the PASS developer. This different software is the backup flight system.

To take over control of the vehicle, the BFS monitors the PASS GPCs to keep track of the current state of the vehicle. If required, the BFS can take over control of the vehicle upon the press of a button. The BFS also performs the systems management functions during ascent and entry because the PASS GPCs are operating in GN&C. BFS software is always loaded into GPC 5 before flight, but any of the five GPCs could be made the BFS GPC if necessary."

University of Glasgow

Conclusions: Hardware Design

- Fault Tolerant Architectures.
- Basics of hardware management.
- Fault models.
- Hardware redundancy.
- Space Shuttle GPC Case Study.

University of Glasgow

Any Questions…