Safety-Critical Dependable Computing: A Retrospective & A Personal Journey

Presented at the 50th Meeting
IFIP WG10.4
Annapolis, MD, USA
30 June 2006
By
Jaynarayan H. Lala

APOLLO
Guidance, Navigation, & Control Computer


UPLINK ACTY
TEMP
AUTO
GIMBAL LOCK
HOLD
PROG
FREE
RESTART
NO ATT
TRACKER
STBY
ALT
KEY REL
OPR ERR
COMP ACTY
PROG
VERB
NOUN
+
7
8
9
CLR
ENTR
−
4
5
6
STBY
RSET
1
2
3
KEY REL
VERB
NOUN
Word Length: 15 bits plus parity.
Fixed Memory Registers: 36,864 Words.
Erasable Memory Registers: 2,048 Words.
Number of Normal Instructions: 34.
Number of Involuntary Instructions (Increment, Interrupt, etc.): 10.
Number of Interface Circuits: 227.
Memory Cycle Time: 11.7 microseconds.
Addition Time: 23.4 microseconds.
Multiplication Time: 46.8 microseconds.
Number of Logic Gates: 5,600 (2,800 packages).
Volume: 0.97 cubic feet.
Weight: 70 pounds.
Power Consumption: 55 watts.
In August 1961 NASA contracted the MIT Instrumentation Laboratory (later called the Charles Stark Draper Laboratory) to develop the Apollo guidance, navigation and control system. Eldon Hall (shown above) was selected to lead the development team, and astronaut David Scott

Apollo Computer Performance

- Word Length: 15 bits plus parity
- Fixed Memory Registers: 36,864 Words
- Erasable Memory Registers: 2,048 Words
- Throughput: ~40,000 Instructions/sec
- Number of Logic Gates: 5,600 (2,800 packages)
- Volume: 0.97 cubic feet, Weight: 70 pounds.
- Power Consumption: 55 watts

- Dependability: Never failed in over 100,000 hrs of cumulative operation and testing

Selected Draper Fault Tolerant Computers: Architectural Attributes

	1960's	1970's			1980's		1990's & Beyond
	APOLLO GNC	HUDAP	F-8 DFBW	FTMP	FTP	AIPS	FTPP
Open Architecture	No	No	No	No	No	Partially	Yes
Computational Redundancy	Simplex	Duplex	Triplex	Parallel Hybrid	Simplex, Duplex, Triplex, Quad	Mixed Redundancy	Mixed Redundancy
Program Rollback	Yes	Yes	No	No	No	Yes	Yes
Exact Agreement	N/A	No	Yes	Yes	Yes	Yes	Yes
Design Diversity	N/A	No	No	No	Yes	Yes	Yes
Byzantine Resil.	N/A	No	No	Yes	Yes	Yes	Yes
Synch. Granularity	N/A	Frame	Frame	Instruction	Microframe	Microframe	Functional
Primary Means for FT	Quality Control	Comparison	Voting Error Mask	Voter + Spares	Congruent Computation	Congruent Computation	Congruent Computation
H/W Fault Coverage	Low	Medium	High	Extremely High	Extremely High	Low to Extremely High	Low to Extremely High
Logical Org.	Centralized	Centralized	Centralized	Hybrid	Centralized	Distributed	Distributed
Physical Org.	Centralized	Centralized	Centralized	Centralized	Distributed	Distributed	Distributed

APOLLO GNC: Apollo Guidance & Navigation Computer
HUDAP: Hydrofoil Universal Digital Auto-Pilot
F-8 DFBW: F-8 Digital Fly-By-Wire

FTMP: Fault Tolerant Multi-Processor
FTP: Fault Tolerant Processor
AIPS: Advanced Information Processing System
FTPP: Fault Tolerant Parallel Processor

Dear Dr. Kuhlman:

This is in response to the inquiry from your office regarding quantification of probability terms used in connection with acceptable levels of reliability for airborne systems in civil aircraft.

Section 25.1309 of the Federal Aviation Regulations requires that airplane systems be designed so that the occurrence of any failure condition (combinations of failures in addition to single failure considerations) which would prevent the continued safe flight and landing of the airplane is extremely improbable. The Federal Aviation Administration has accepted substantiating data for compliance with that requirement which shows by analysis that the predicted probability of occurrence of each such failure condition is 10^{-9} per hour of flight.

To date, this criteria has been applied in the certification/evaluation process to Concorde systems, fully-powered hydraulic primary flight controls on wide-body subsonic transports and automatic landing systems for low weather minimum operation. It is currently being applied to the first complete airplane to which this requirement applies.

We refer to the Rockwell International Model 750, to be designed and built by the Bethany Aircraft Division, Bethany, Oklahoma. On February 20, 1974, their staff was informed that the 10^{-9} per hour of flight was the numerical value associated with the term "extremely improbable."

We further believe that failure of all channels on the same flight in a "fly-by-wire" flight control system should be extremely improbable; that is, be shown to have a probability of occurrence equivalent to that which has been shown for similar failure of all fully-powered hydraulic flight con-

The Charles Stark Draper Laboratory, Inc.

68 Albany Street, Cambridge, Massachusetts 02139 Telephone (617) 258- 1451

Mail Station #35

MEMO

TO: Distribution
FROM: Albert Hopkins
DATE: 23 July 1974
SUBJ: Report of Visit to NASA/Langley on Advanced Fault-Tolerant Multiprocessor

The problems we have to face to get a flyable prototype include the following.

1. Develop the appropriate system architecture to meet the computing requirement with a system failure rate of 10^{-10} crashes (in the computer sense) per hour.

2. Identify and nurture a source for the LSI we need with adequate environmental limits, testability, and reliability, but reasonable cost.

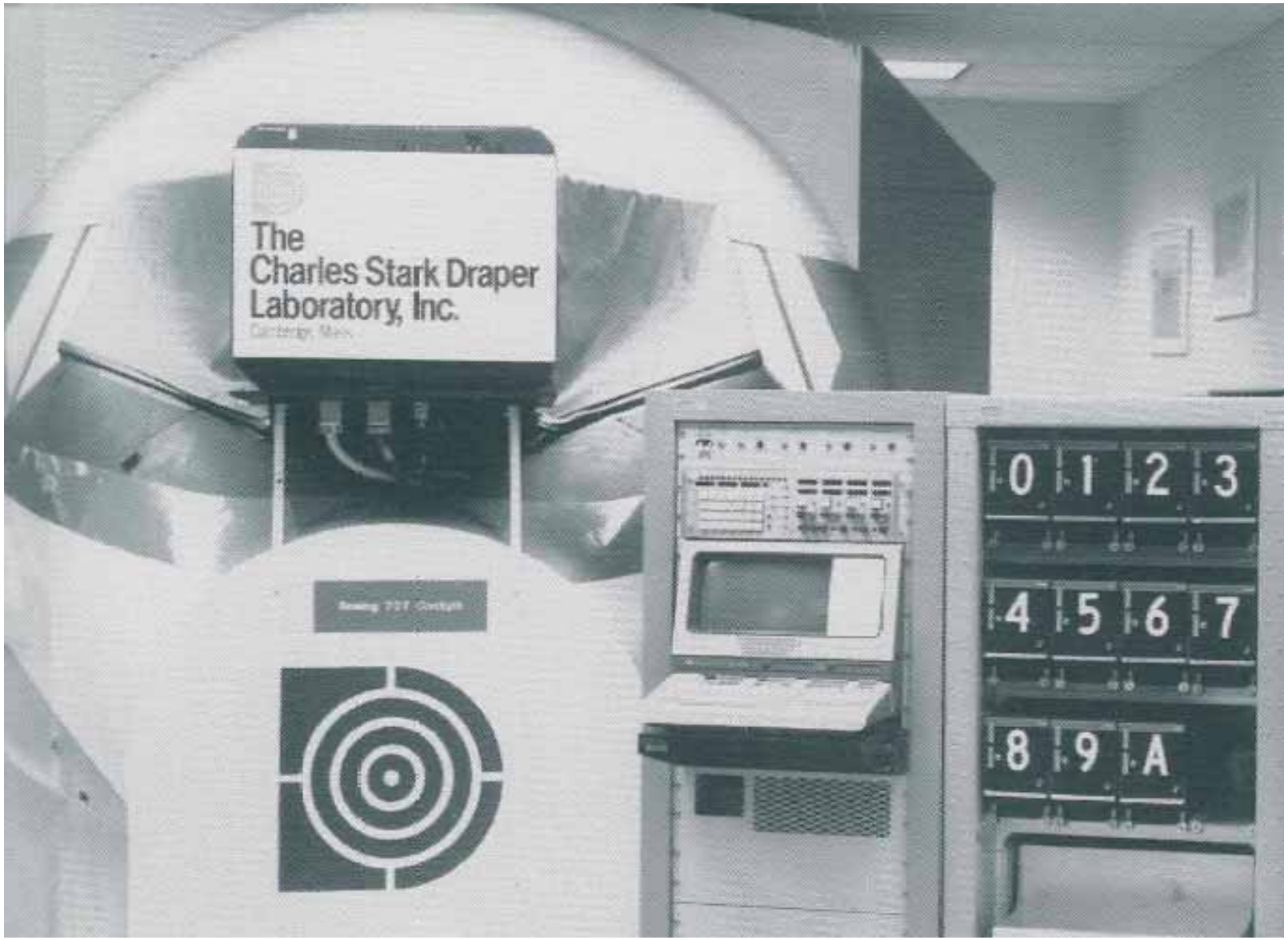
3. Generate reliable software at moderate cost.

4. Make maintenance simple and cheap.

5. Packaging, which may be awkward for a distributed system, particularly if we have processors in the wings and tail. This includes environmental control problems.

Thus our computer architecture is the tip of an iceberg, as usual. Nevertheless we have the resources to do the job if funds are available. We would have to have a flying prototype somewhere around 1981. We haven't discussed

Fault Tolerant Multi-Processor (FTMP)



CSDL-P-2701

EVOLUTION OF FAULT-TOLERANT COMPUTING AT THE CHARLES STARK DRAPER LABORATORY, 1955–86

by

Albert L. Hopkins, Jr.
Jaynarayan H. Lala
T. Basil Smith, III

June 1986

Presented At

International Federation for Information Processing (IFIP)
Working Group 10.4 – Reliable Computing and Fault Tolerance
Symposium on "The Evolution of Fault-Tolerant Computing"
Baden, Austria
June 30, 1986

Advanced Information Processing Systems (AIPS)

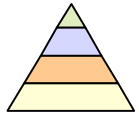
Self Regenerative Systems (SRS): The Fourth Generation

Prevent Intrusions
(Access Controls, Cryptography, Trusted Computing Base)


Trusted Computing Base


Access Control & Physical Security

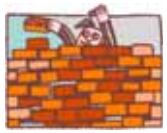

Cryptography


Multiple Security Levels

1st Generation: Protection

But intrusions will occur

Detect Intrusions, Limit Damage
(Firewalls, Intrusion Detection Systems, Virtual Private Networks, PKI)


Firewalls


Boundary Controllers


Intrusion Detection Systems


VPNs


PKI

2nd Generation: Detection

But some attacks will succeed

Tolerate Attacks
(Redundancy, Diversity, Deception, Wrappers, Proof-Carrying Code, Proactive Secret Sharing)


Intrusion Tolerance


Big Board View of Attacks Real-Time Situation Awareness & Response


Graceful Degradation


Hardened Operating System

3rd Generation: Tolerance

So the system must reconstitute

Restore System
(Diagnosis, Learning, Reconfiguration, S/W Rejuvenation, Natural Immunity, Reflection)


Self-Aware


Diagnosis


Learning


Reconfiguration


Reflection

4th Generation: Regeneration

PROGNOSTICATIONS

"Making predictions is hard

……..especially, about the future."

- Anon.

Defending against Attacks

- In 2031, we will still be fortifying our defenses.
- Threat is not going to disappear
  - Intent,
  - Capability, and
  - Exploitable Vulnerabilities, will all be there.
- Unless there is a fundamentally new architecture construct, secure by design, systems will always be exploitable.
- Unfortunately, the situation is more analogous to a arms race between attackers and defenders.
- Can Cognitive Systems technologies help?

Accidental Faults

- For safety-critical systems (or at least mission-critical systems), we may revert to a radical (old) idea:
  - The Apollo Approach
- Is it possible to make systems nearly perfect and defect free yet affordable?
- Can the Apollo GN&C approach be scaled to the demanding functionality of today's and future applications?

Simplicity + Moore's Law= Dependability?

- Can the results of Moore's Law and a smart approach to limiting "bells & whistles" be combined to create single-string dependable computer systems?
  - Effect of Moore's Law: This laptop is about 10^5 more powerful than the Apollo computer in raw power.
    - But its functionality and usefulness has not increased proportionally.
  - Simplicity: If unnecessary bells and whistles were stripped from all hardware, firmware, OS, middleware (get rid of middleware?), and applications, could this laptop should provide 10^5 more functionality than the AGC?
- Has there been sufficient progress in specification & production processes, verification, and validation technologies to assure acceptable residual defects?