# List of Scenarios Outline (1)

Honeywell

- Leading the list are examples of Byzantine failures because of the very widespread "that can't happen" disbelief in these failures and the fact that there are known solutions
- Introduction to the Byzantine Generals Problem
  - Definitions of Fault, Failure, and Error
  - Byzantine failure definitions and background
- Examples of actual occurrences
  - Space shuttle mission STS-124
  - Space shuttle data bus standing wave
  - Mid-value select
  - Command / Monitor wrap-back
  - Time-Triggered Protocol (TTP/C) heavy ion fault injection
  - Multi-Microprocessor Flight Control System (M²FCS)
  - Potential grounding of an entire aircraft fleet
  - A pushbutton input to the command and monitor lanes of an airplane brake system caused the system to fail (see section 1.6.7 of www.fss.aero/accident-reports/dvdfiles/ES/1998-05-21-ES.pdf)

➢ ***All these problems could have been found by formal analysis***

# Fault, Failure, and Error Definitions

Honeywell

- The following three definitions are those agreed to by the "fault tolerance fraternity" (IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance).
  - *Error*
    The part of a device's **state** which differs from that state which would exist in an unimpaired device. The term state usually refers to information stored in a device. However, the definition of this term sometimes is stretched to include the structure of a device as well.*
  - *Fault*
    The **phenomenological** cause of a failure. When the stretched definition of state is used, a fault is the cause of an error in the device's structure. Failures and errors are effects of a fault.
  - *Failure*
    The deviation of a device's **service** from the expected service proscribed by some agreed to **specification**. The deviation, and therefore the failure, can be classified by type, persistence, and degree of severity. A measure of the degree to which the failure may adversely affect the device's service is called the criticality of the failure.
- *Fault propagation*
  The inducement of other faults, failures, and errors in additional devices. Actually, this is a misnomer. Except for cracks, faults don't propagate. This really is "failure propagation". However, the term is so deeply ingrained into the industry, it can't be corrected.

* J. C. Laprie "Dependable Computing and Fault-Tolerance: Concepts and Terminology" Proceedings 15th Fault Tolerant Computing Symposium, Ann Arbor MI, June 1985; and, IFIP WG 10.4 Summer 1984 meeting, Kissimimee, Fla.; and, LAAS Report No. 84.035, June 1984.
Updated in "Basic Concepts and Taxonomy of Dependable and Secure Computing" IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, Jan-Mar 2004

# The Byzantine Generals Problem

Honeywell

- A type of failure –
  described in some literature as a story about Byzantine-era generals trying to co-ordinate an attack, with possible traitors among the generals and/or their messengers. The point of this story is mutual agreement – agreement wins, disagreement looses. (There are thousands of papers on this subject.)
  (see L. Lamport, R. Shostak, M. Pease. *The Byzantine Generals Problem*, ACM Transactions on Programming Languages and Systems, Vol. 4, No. 3, July 1982, pages 382-401; http://research.microsoft.com/en-us/um/people/lamport/pubs/byz.pdf)
- Typical map to real world: Generals = processors, Messengers = data network communication
- Note: "Byzantine" is not synonymous with "bizarre". "Byzantine" has a precise meaning which deals with failure behavior that works against reaching agreement.
- In our Safecomp 2003 paper, we created concise practical definitions for designers:
  - Byzantine **fault**
    any fault that presents different symptoms to different observers
  - Byzantine **failure**
    the loss of a system agreement service due to a Byzantine fault

  (see K. Driscoll, B. Hall, H. Sivencrona, P. Zumsteg. Byzantine Fault Tolerance, from Theory to Reality, LNCS Computer Safety, Reliability, and Security, Volume 2788/2003, pp. 235-248, 2003; http://www.cs.indiana.edu/classes/p545-sjoh/read/Driscoll-Hall-Sivencrona-Xumsteg-03.pdf; or [better version] The Real Byzantine Generals, 23rd Digital Avionics System Conference (DASC), 2004)
- Any operational data link between redundant devices must exist for some type of agreement. Even asynchronous systems without voting need "equalization" to prevent divergence.
- It is nearly impossible to create a highly-dependable system without Byzantine Fault tolerance.

# First Picture of a Byzantine Fault?

Honeywell

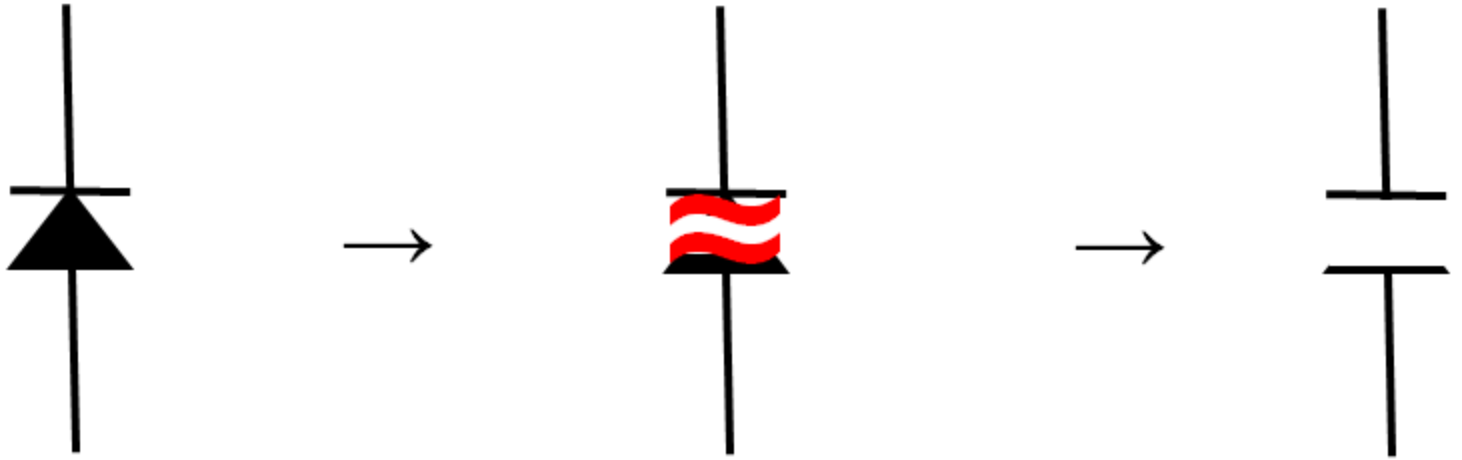
At 12:12 GMT 13 May 2008, a NASA Space Shuttle was loading hypergolic fuel for mission STS-124 when a 3-1 split of its four control computers occurred. Three seconds later, the split became 2-1-1. During troubleshooting, the remaining two computers disagreed (1-1-1-1 split). ***Complete system disagreement.*** But, none of the computers or their intercommunications were faulty! The ***single fault**** was in a box (MDM FA2) that sends messages to the 4 computers via a multi-drop data bus that is similar to the MIL STD 1553 data bus. This fault was a simple crack (fissure) through a diode in the data link interface.

**Figure 1.** Two views (90 degrees apart) of a fissure that appears to go through the silicon - Red arrows.

* the Byzantine Assassin

# Transmogrification*

Honeywell



A diode … with a perpendicular crack … is really a capacitor

How many Failure Modes and Effects Analysis (FMEA) procedures ask what would happen if one electrical part (a diode) changed into another (a capacitor)? At the electrical part level, this appears to be magic. And, yet, the simplest of failures (a crack) caused this transmogrification. The literature includes other examples of capacitors becoming resistors, transistors becoming silicon controlled rectifiers (SCRs), amplifiers becoming oscillators, …

* Transmogrification definition: the act of changing into a different form or appearance (especially a fantastic or grotesque one), often as if by magic

# Digital Circuitry Behavior

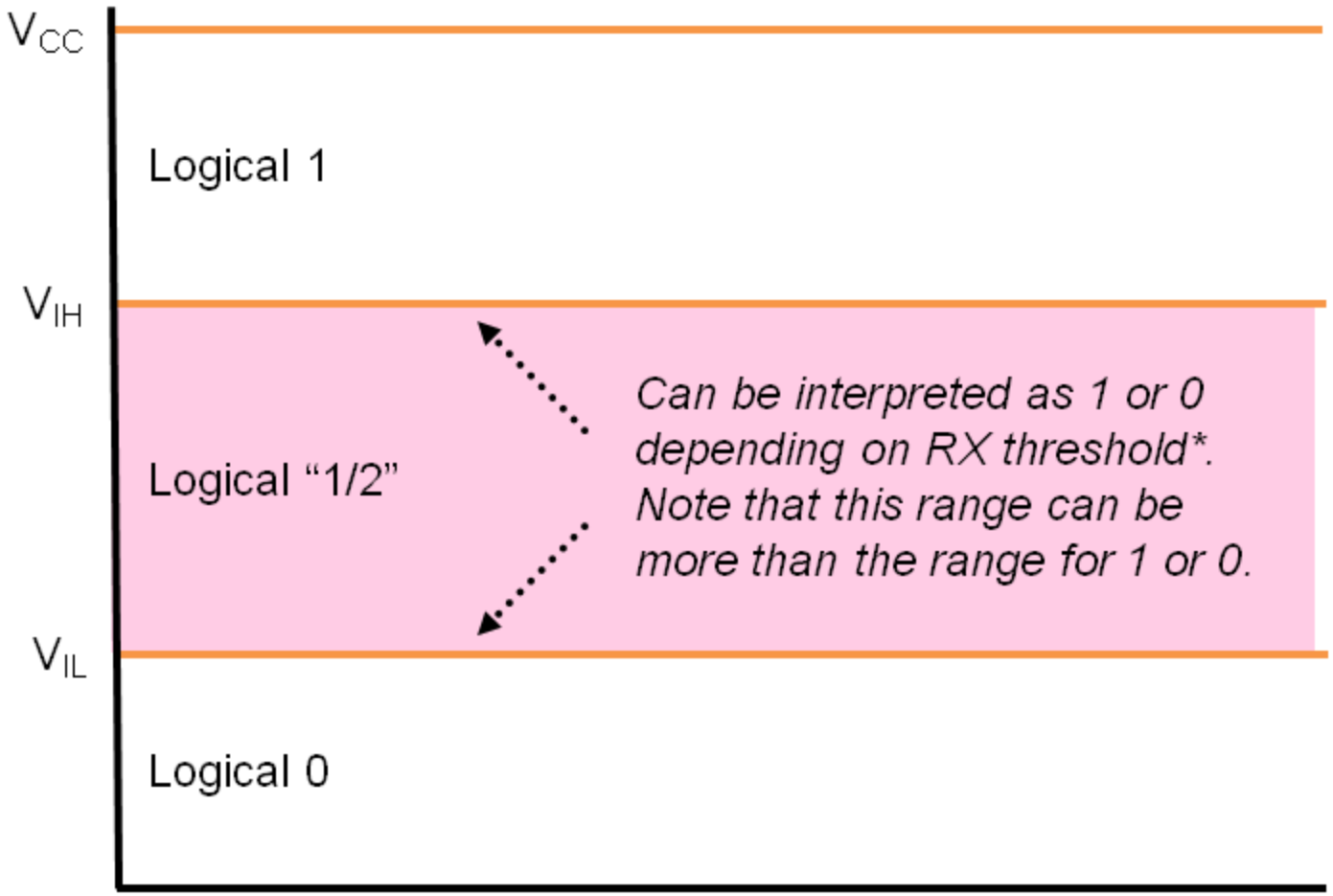
Honeywell

"There is no such thing as digital circuitry. There is only analog circuitry driven to extremes." – [I stole this quote from ???]

This leads to the possibility of a faulty binary logic signal being "1/2" (an indeterminate state between 0 and 1).

# Definition of “1/2”

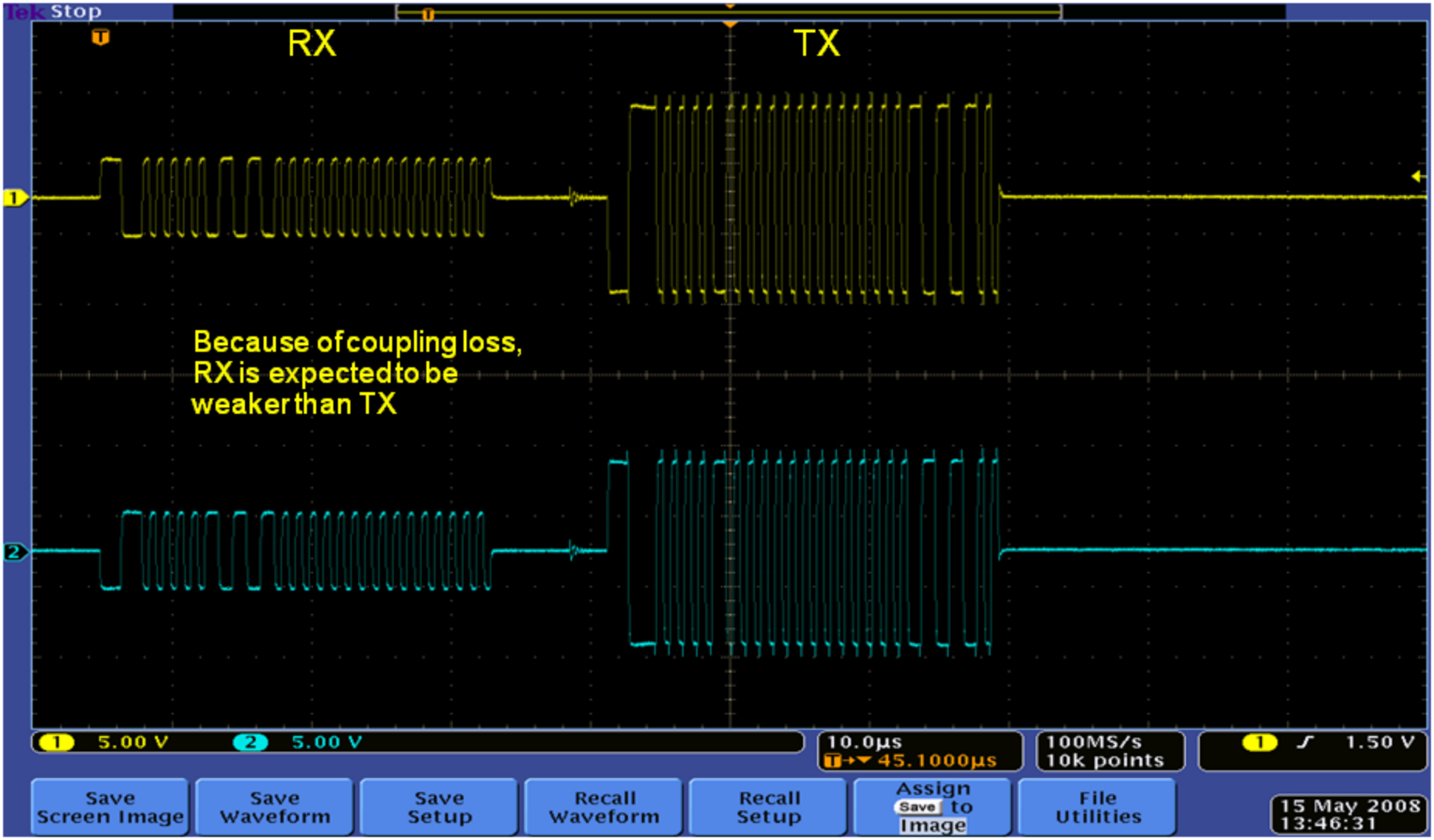
Honeywell




* RX threshold depends on manufacturing tolerance, power supply voltage, ground shift, temperature, fields from nearby signals, accumulated stress, …

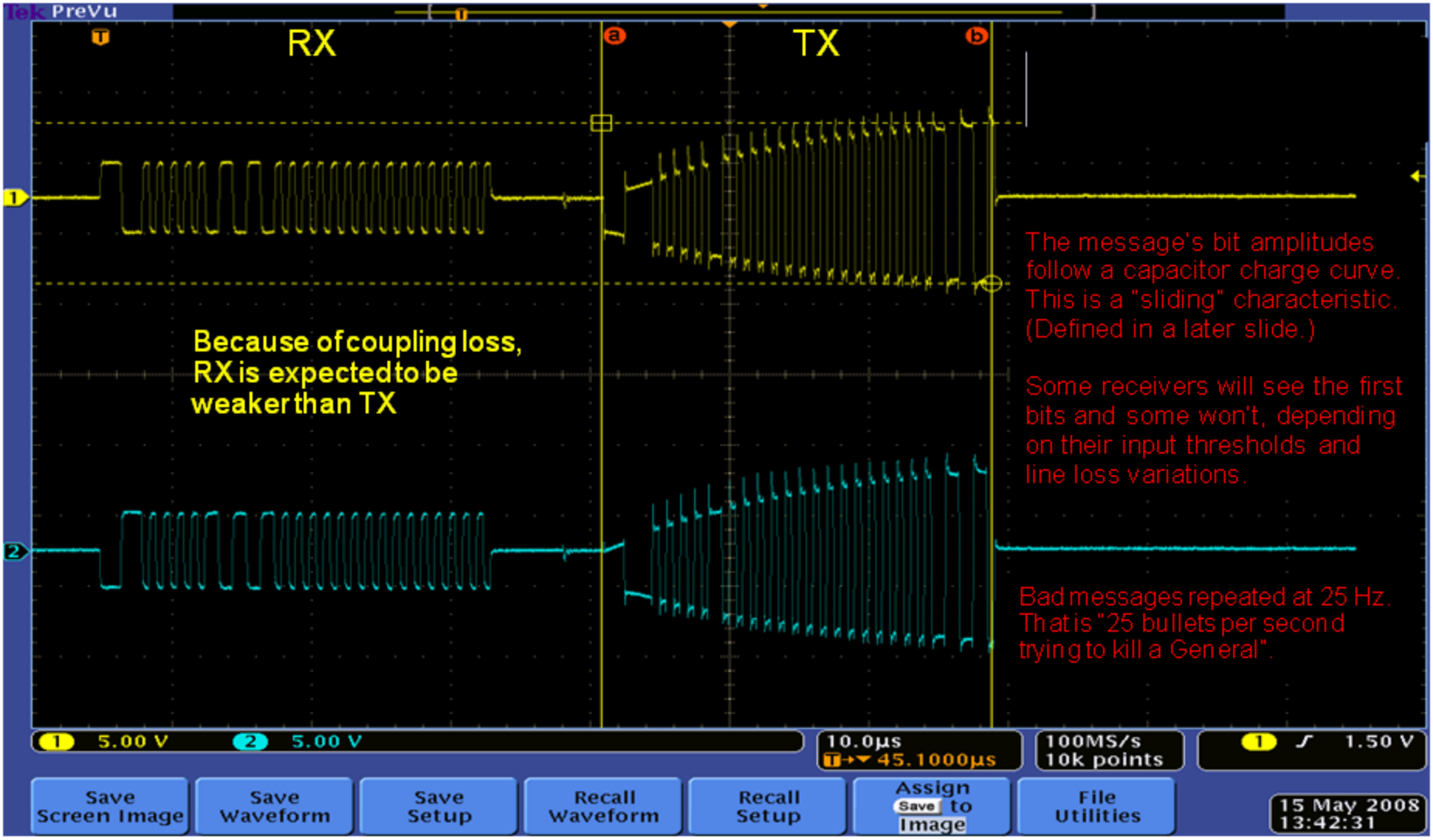
# Normal Messages (differential traces)

Honeywell



# Faulty TX Message on the Right

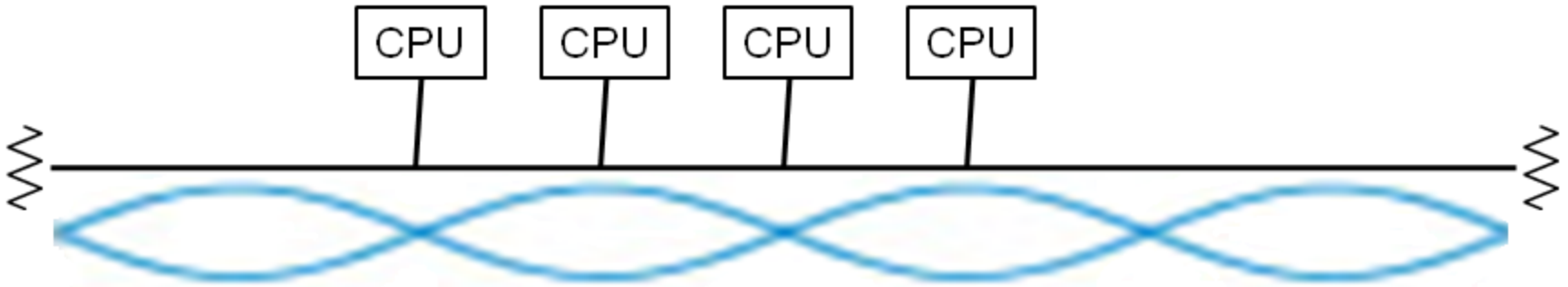
Honeywell



# Shuttle Databus: Terminated with Extreme Prejudice

Honeywell

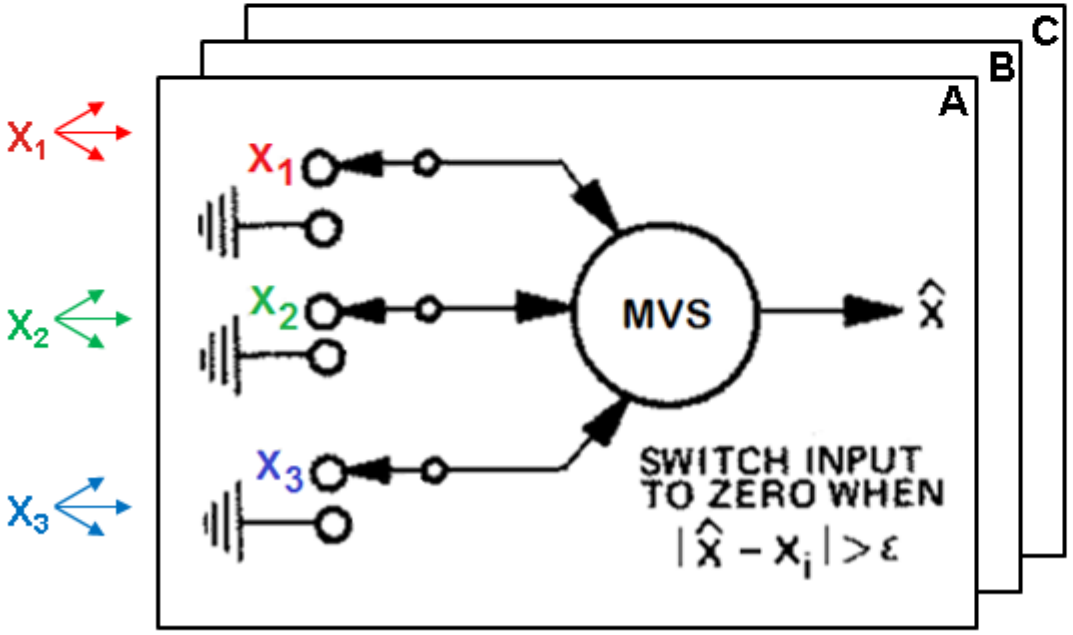
- This was not the first time a single Byzantine fault on a Shuttle databus has caused the loss of the quad computer system. There have been several of them, with different causes.
- One example:
  - More than 25 years ago, a technician used the wrong resistor value for bus termination.
  - Reflections from the impedance mismatch caused a standing wave on the bus.
  - As Murphy would have it, two of the four central computers were attached to the bus at antinodes of this standing wave and two were attached at nodes.
  - This caused a 2-2 split of the computers. Luckily, this happened in the lab.
- The lesson learned here should not be that diodes or terminating resistors are dangerous, but that Byzantine faults must be tolerated.
  - If you think these were hardware problems, you have succumbed to a specialization silo boundary
  - These were software failures in "non-universal error" handling (which "voted out" good processors).
  - If this software had been written correctly (to tolerate Byzantine faults), these system failures would not have occurred! The processors would have stayed in agreement.
- George Santayana: "Those who cannot remember the past are condemned to repeat it."
- Steve Thompson: "These should be lessons *learned*, not just lessons *observed*."
- Kevin Driscoll: "Those who *don't understand* the past are condemned to repeat it."



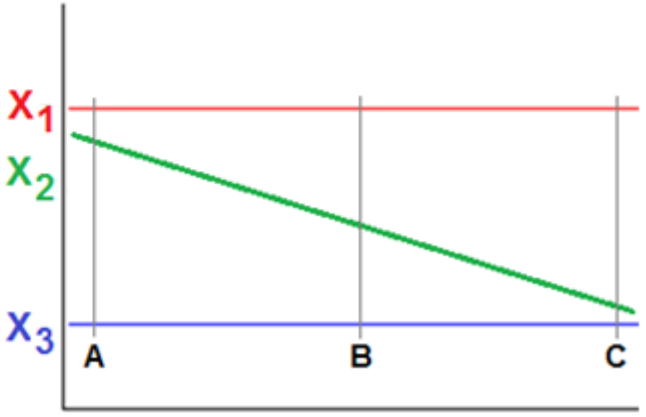
# Mid-Value Select (MVS)

Honeywell

- **Shows that Byzantine problems are not unique to synchronous or exact-match redundant systems; dispels myths that async and inexact systems are immune**
- **An example of "if it ain't broke" fix it until it *is* broke**
- **Counterproductive attempt to tolerate two faults (a very common design)**
  - Any redundant input, $X_i$ , whose value is too much different (ε) than the previous MVS value, is assumed to be faulty and forced to be zero (ground), the middle of the valid range
  - The points **A**, **B**, and **C** on the graph show three sampling points of three asynchronous redundant mid-value selectors, named **A**, **B**, and **C** respectively
  - If $X_1 = X_3 + (\varepsilon, 2\varepsilon]$, it can happen that a persistent state is created in which: the **A** MVS sees $X_3$ as faulty; the **C** MVS sees $X_1$ as faulty, and the **B** MVS sees no fault



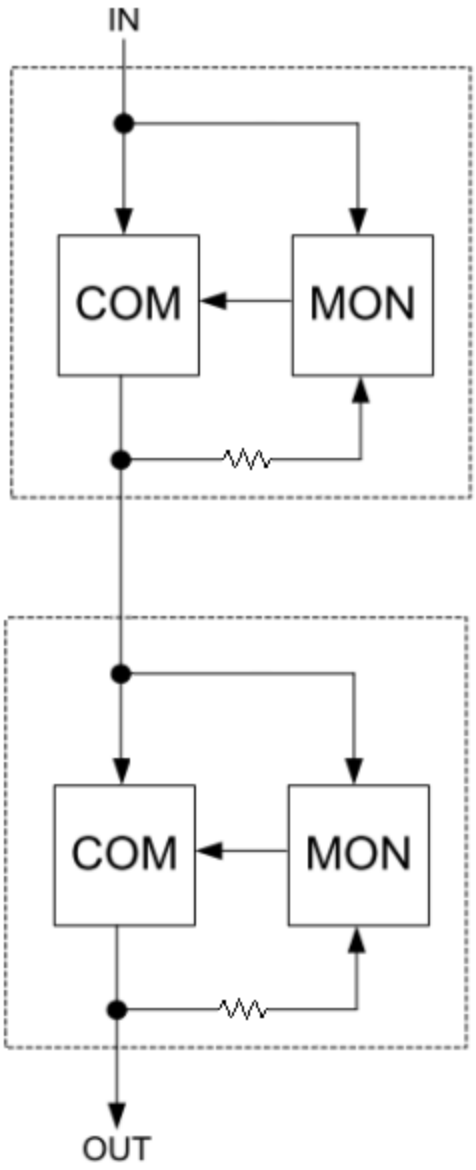

**A**, **B**, and **C** could see a different $\hat{x}$ after "tolerating" a fault, depending where zero is on this graph.



# Command (COM) / Monitor (MON)

Honeywell

- Couldn't afford the SWaP to do a self-checking pair bus, per the original design
- Used COM / MON instead
- The signal fed back to the MON was made weaker than the weakest possible signal seen by any receiving node at the other end of the communication link. (Rationale: The MON would see a degraded signal before a far end receiving node would.)
- A design flaw in the circuitry caused the receiving node to lock up on a signal that the transmitting node's MON did not catch, even though the receiver and the MON input circuitry were identical.
- Lesson learned: ***It may be impossible to create a COM / MON (or any wrap-back fault detection mechanism) which can observe all failures that might escape!*** (The reason for the extreme emphasis here is because these kinds of mechanisms are ubiquitous in safety-critical and security-critical systems. Most don't have the intentional weakening of the wrap-back signal, which makes them more susceptible to escapes.)



# Time-triggered Protocol (TTP/C) Heavy Ion Fault Injection

Honeywell

As part of the Fault Injection Techniques (FIT) project at the Carinthia Tech Institute, a first generation time-triggered communication controller (TTP-C1) was radiated with heavy ions, which caused several system failures due to Byzantine faults. The dominant Byzantine failure mode was due to marginal transmission timing caused by bit-flips in registers controlling the transmission time-base. This caused a node's timing to be permanently (until reset) slightly-out-of-specification (a major subcategory of Byzantine faults).

Sivencrona, H., P. Johannessen, M. Persson, J. Torin, 2003, Heavy-ion Fault Injection in the Time- triggered Communication Protocol, Proc. 1st Latin American Symposium on Dependable Computing LNCS 2847, pp. 69-80.

# Multi-Microprocessor Flight Control System ($M^2FCS$)

Honeywell

- During testing of the $M^2FCS$ system prototype, Byzantine failures were observed with a mean repetition period of 50 seconds with a variance of 10 seconds. Because this was in a 20 Hz control loop, the probability of any loop experiencing a Byzantine fault was 1/1000.
- The root cause of these failures was isolated to the marginal behavior of the physical layer that had been pushed to the limits in the initial prototype set-up.
  - A similar situation could occur to a fielded system due to degradation of oscillators, bus drivers, or cabling
- Many of the Byzantine failures resulted in 4-way splits. Some of the nodes saw a good message, some saw the same message as arriving late, some nodes saw a checksum error in that message, and some nodes saw both types of errors.

# Potential Grounding of an Entire Aircraft Fleet

Honeywell

- Highly redundant system (theoretically, enough redundancy to tolerate at least two Byzantine faults)
- But, system was not designed to tolerate Byzantine faults
  - Therefore, no amount of redundancy was sufficient
- Several instances were reported where multiple channels of the system lost sync at the same moment
- Solution found just a couple days before the entire fleet would have been grounded

# Some Lessons from Byzantium

Honeywell

- A Byzantine Assassin can …
  - … be an Outsider (not a General nor one of the Generals' Messengers)
  - … be created by the simplest of faults (e.g. crack) in the simplest of parts (e.g. diode)
  - … convince "good guys" to kill (or ostracize) themselves
  - … cause as many corpses (or cliques) as there are entities to attack
    - Manual reversion to the shuttle's dissimilar backup probably would not have helped
      - ➔ *Without Byzantine Fault tolerance, no amount of redundancy is enough*
- *"Murder mechanisms" (e.g. vote-out reconfiguration, hybrid NMR) are inherently dangerous*
  - A Byzantine Assassin can subvert the mechanism into being a mass murder accomplice
  - Suicide is safer; that is one reason to use atomic self-checking pairs (e.g. Boeing 777 AIMS)
- Need to consider "sliding failures"
  - A part's behavior gradually changes
    - From in-specification to (slightly) out-of-specification, or vice versa
  - Higher probably of hitting a Byzantine region of behavior than one would expect
- Faults can convert one type of part into another
  - Diode → capacitor, capacitor → resister, transistor → SCR, input → output, amplifier → oscillator, analog circuit → digital circuit, digital circuit → analog circuit
  - Related phenomenon: a fault can create a part from "nothing" ("partogenesis" ☺)
    - Typically due to the fault causing a large increase in some parasitic properties
    - Today's higher speed circuits are more susceptible to parasitic property changes
    - Can also be caused by emergent properties
- *It may be impossible to create a COM / MON or **any** type of wrap-back fault detection mechanism which can observe all failures that might escape*
- FMEA teams should include:
  - Curmudgeons, skeptics, and "pathological thinkers" (to counterbalance designers, who tend to be optimists)
  - Members of related or neighboring disciplines
  - Physicists (find other Feynmans)

# List of Scenarios (2)

Honeywell

- **"Out of Band" Fault Propagation**

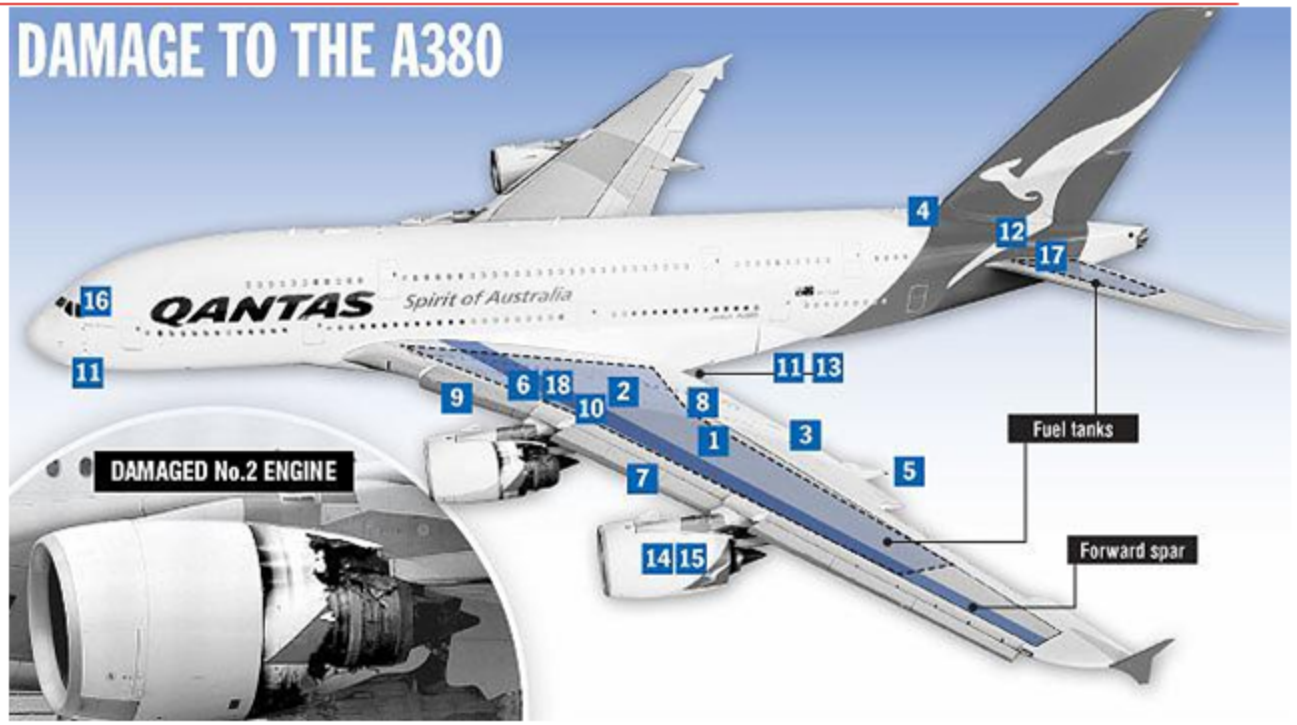
  **(An "in band" fault propagation is a failure or error that is transferred through a mechanism, which was intended to connect components; everything else is "out of band")**

  - **Shrapnel**
    - Qantas A-380 (plus Sioux City DC-10 and others)
    - Exploding capacitor
    - Under heavy vibration, a toroidal transformer came loose and, as it flew around in the box, it pulverized all the other components in the box
  - **Photonic EMI, blinding**
    - Fiber optic
    - Multi-chip module (MCM) die died?
  - **Thermal**
    - Multicore [phenomenon is known, no known failures]
      - Even in a single core, thermal "fault propagation" invalidates partitioning
- **Power supply and ground return [out-of-band subset?]**
  - **Director Evaluation Flight Test**
  - **Backup power over-voltage**
  - **Distributed Processor System Lockup**
  - **"Sync Bandit"**
  - **Other Ground(less) Problems**

# Self-Inflicted Shrapnel, 1st Example

Honeywell

1 Massive fuel leak in left mid fuel tank -- there are 11 tanks, including tail's horizontal stabilizer
2 Massive fuel leak in the left inner fuel tank
3 A hole on the flap fairing big enough to climb through
4 Aft fuel system failed, preventing many fuel transfer functions
5 Problem jettisoning fuel [180K lbs]
6 Massive hole in the top of wing
7 ~~Partial~~ failure of leading edge slats
8 Partial failure of speed brakes and ground spoilers [and ailerons]
9 Shrapnel damage to the flaps
10 Loss of all hydraulic fluid in one of the jet's two systems
11 Manual extension required for landing gear




12 Loss of one generator and associated systems [electrical busses 1 and 2 failed]
13 Loss of brake anti-skid system
14 No.1 engine could not be shut down in the usual way after landing because of major damage to systems
15 No.1 engine could not be shut down using the fire switch, which meant fire extinguishers wouldn't work
16 There were 54 different warning messages
17 Fuel was trapped in the trim tank (in the tail) creating a balance problem for landing
18 Left wing forward spar penetrated by debris

Only one engine (of 4) was working normally and had thrust reversing. Four blown tires. Leaked fuel on 1600°F wheels.

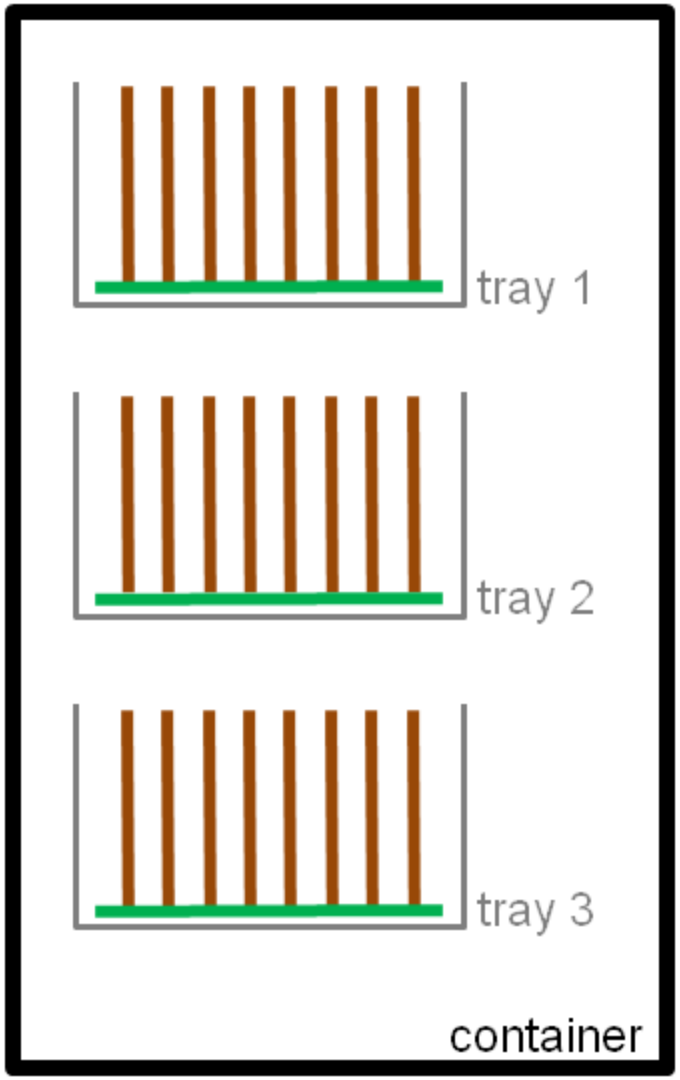
Richard Woodward (a Qantas A380 pilot and deputy president of the Australian and International Pilots Association) said that the "number of failures is unprecedented, [...] There is probably a one in 100 million chance to have all that go wrong." But, it was very precedented. There have been over a half-dozen previous similar incidents (the Sioux City DC-10 crash is well known). "Those who cannot remember the past are condemned to repeat it."

# Self-Inflicted Shrapnel, 2nd Example

Honeywell

- Three trays of a redundant military system were enclosed in a single truly "bullet-proof" enclosure. Each **aluminum tray** had a **horizontal motherboard** and several **vertical printed-circuit cards**.

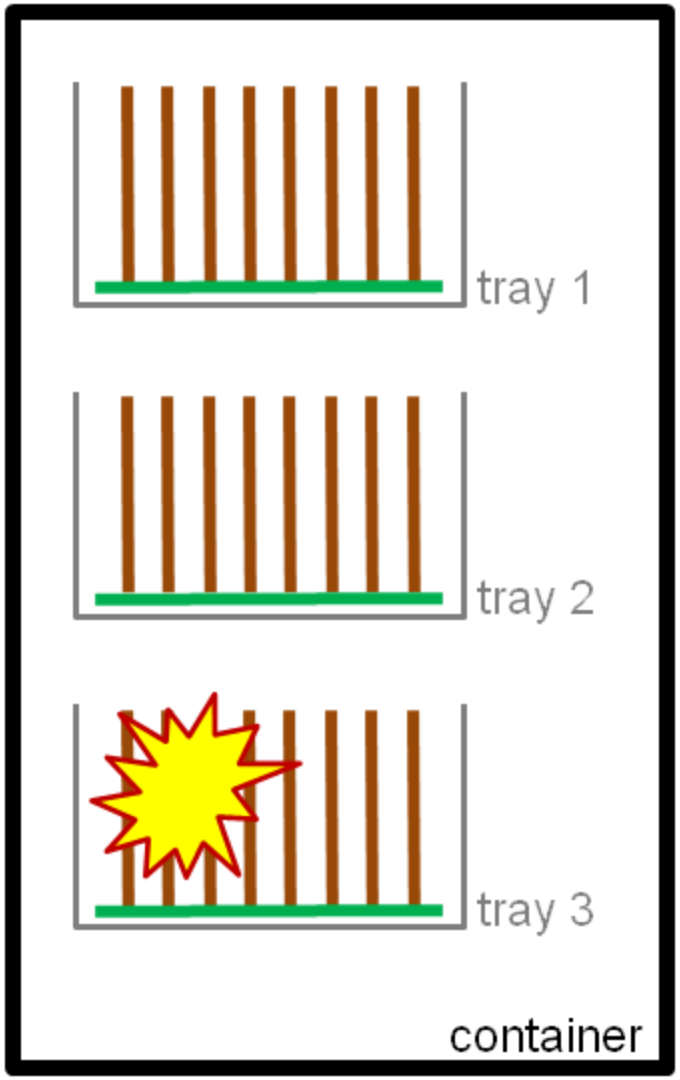
…



# Self-Inflicted Shrapnel, 2nd Example

Honeywell

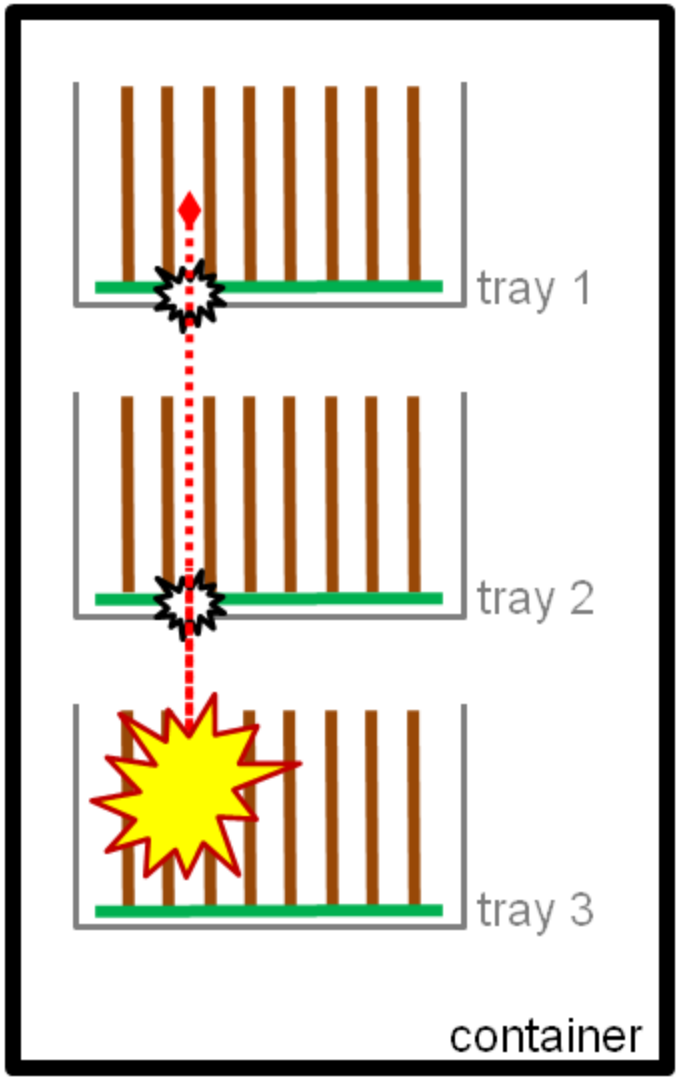
- Three trays of a redundant military system were enclosed in a single truly "bullet-proof" enclosure. Each **aluminum tray** had a **horizontal motherboard** and several **vertical printed-circuit cards**.
- A capacitor in the lowest tray exploded, …



# Self-Inflicted Shrapnel, 2nd Example

Honeywell

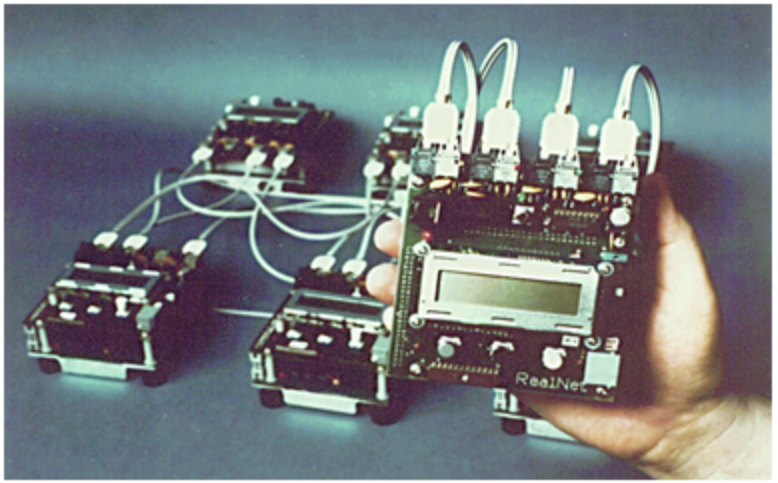
- Three trays of a redundant military system were enclosed in a single truly "bullet-proof" enclosure. Each **aluminum tray** had a **horizontal motherboard** and several **vertical printed-circuit cards**.
- A capacitor in the lowest tray exploded, sending "shrapnel" (pieces of the thick plastic that had encased the capacitor) ***upward*** through the other two trays, causing the redundant trays to fail. The "shrapnel" was sharp, shaped like little arrowheads.
- How many failure modes and effects analysis checklists would include self-inflicted shrapnel that could penetrate multiple aluminum trays and motherboards?
- The violence of this explosion was surprising. It shook the walls of the room in which it occurred.
- Having molten pieces from an electrical failure propagate the failure ***downward*** is generally known (e.g. United B777 P200 panel fire).



# Fiber Optic EMI

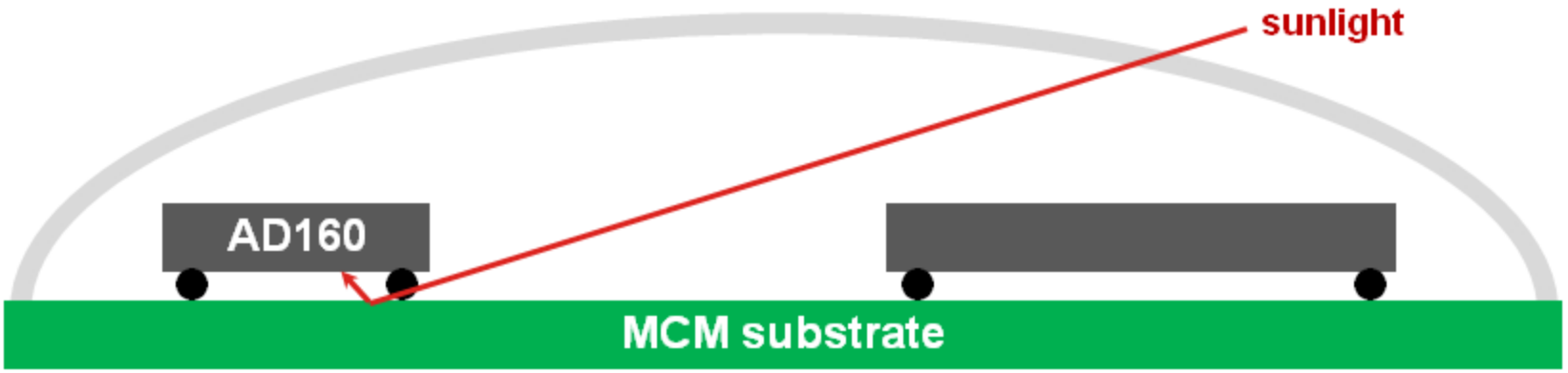
Honeywell

- Common "wisdom" believes that fiber optic (FO) systems are immune to EMI
- But, fiber optic receivers must be more sensitive than wire media receivers
  - Power in fiber optic media is much less than in wire media
  - Fiber optic receivers must be well shielded
- A fiber optic data network was used in an experimental hybrid city bus
  - Controlled all aspects of the vehicle (brakes, steering, driver's control and instruments, engines, motors, battery management, air conditioning, signage/signaling, doors, ...)
  - Direct sunlight getting into an open fiber optic cable blinded the FO receiver
  - The system survived because the network's mesh topology used point-to-point links
  - This sunlight infiltration would have been fatal to bus or passive (multi-)star topologies
  - **Another caveat**: FO was used because the 60 MA/s motor inrush currents (di/dt), could cause significant voltage spikes due to V = L di/dt, with just tiny inductances. Parasitic capacitive coupling makes transformers not fully effective in blocking these spikes. This is a concern for all "more electric" vehicles (aerospace, ground, and sea) that have high di/dt and can develop inductances in cables, connectors, frame grounds, etc. as they age.

# Multi-chip Module (MCM) Die Died?

Honeywell




- A device that used an MCM worked indoors, but not in direct sunlight
- The problem was that …
  - The dome covering the MCM was slightly translucent
  - Sunlight filtering through the dome was able to get underneath an AD160 integrated circuit die (which was mounted active side down) because of the gap between the die and the MCM substrate due to solder-ball type mounting
  - The sunlight hitting the active surface of the die caused photo currents in the die's active devices, which made it shut down as long as the device was in the presence of sufficiently bright light (even though the AD160 was a power supply IC that shouldn't have been particularly susceptible to photo currents)
- Most semiconductor devices have some susceptibility to photocurrents

# Director Evaluation Flight Test

Honeywell

- Failure only occurred when given a “perfect storm” unique set of conditions
  - A climbing right-turn
  - Pulling greater than 3 Gs
  - With some side slip
  - Above 10,000 feet
- All the ICs in an avionics box failed
- No indication of why the failure occurred
- Cause: Only under this set of conditions, a physically unsupported wire carrying a narrow 18 kV pulse moved close enough to the box’s main 12 V power supply and the air rarified enough that an arc-over occurred
  - The rise time of the pulse was so fast that protection devices didn’t have time to work
  - The narrow pulse and low current didn’t leave any burn marks

# Backup Power Over-voltage

Honeywell

- In an F-16 variant's flight control system
  - Each flight control processor used diodes to combine sources of power
  - Each flight control processor protected itself by disconnecting it power supply on any overvoltage
  - The overvoltage detection and the disconnect were after the diode combining of the power inputs
  - A Ram Air Turbine (RAT) backup source of power had a valve stick open, which produced an over-voltage that went to all the flight control processors and through the diode power combiner on each processor
- No flight control processor remained running

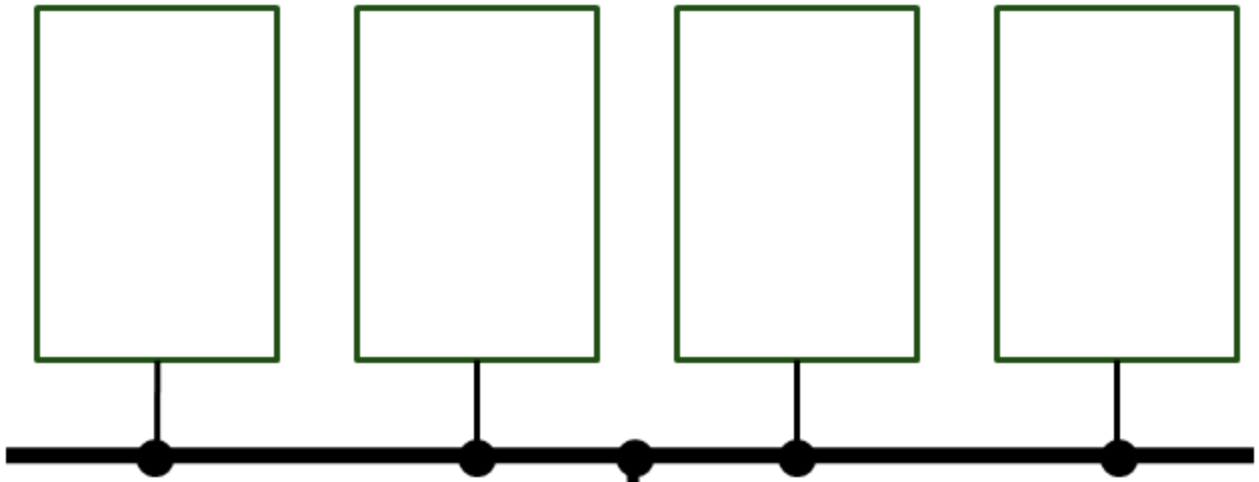
# Distributed Processor System Lockup

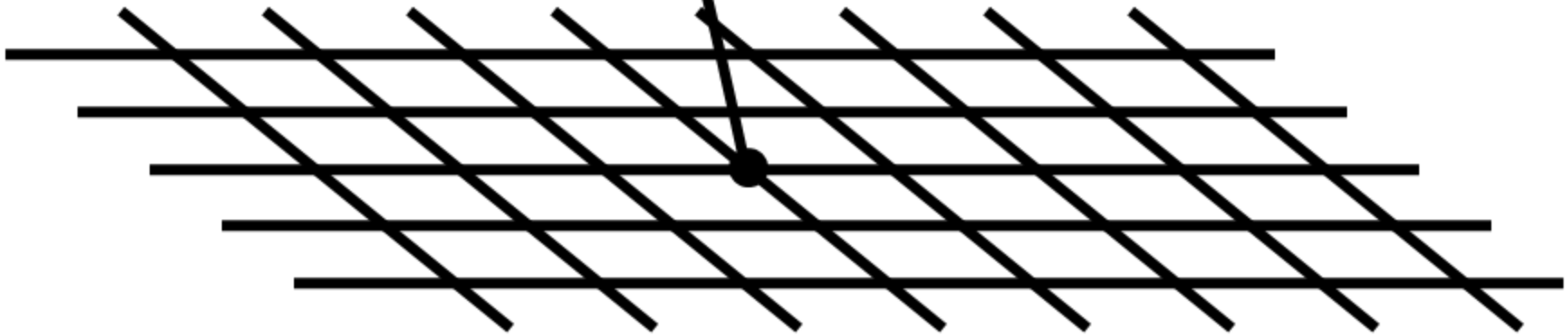
Honeywell

- In one processor, the power supply that converted 28v to 12v failed to an overvoltage output
- The overvoltage caused data bus drivers to turn on constantly, which jammed the data bus to which the driver was connected
- Each processor drove all the system's data busses, with a separate driver for each bus (it was assumed that driver failures were independent within the node)
- All of the bad node's drivers were similarly affected
- All the system's buses were jammed
- This system was designed to be a flight control system with a less than a $10^{-9}$ probability of failure; but, this failure mode showed that each processor's power supply was a system single-point-of-failure

# “Sync Bandit”

Honeywell

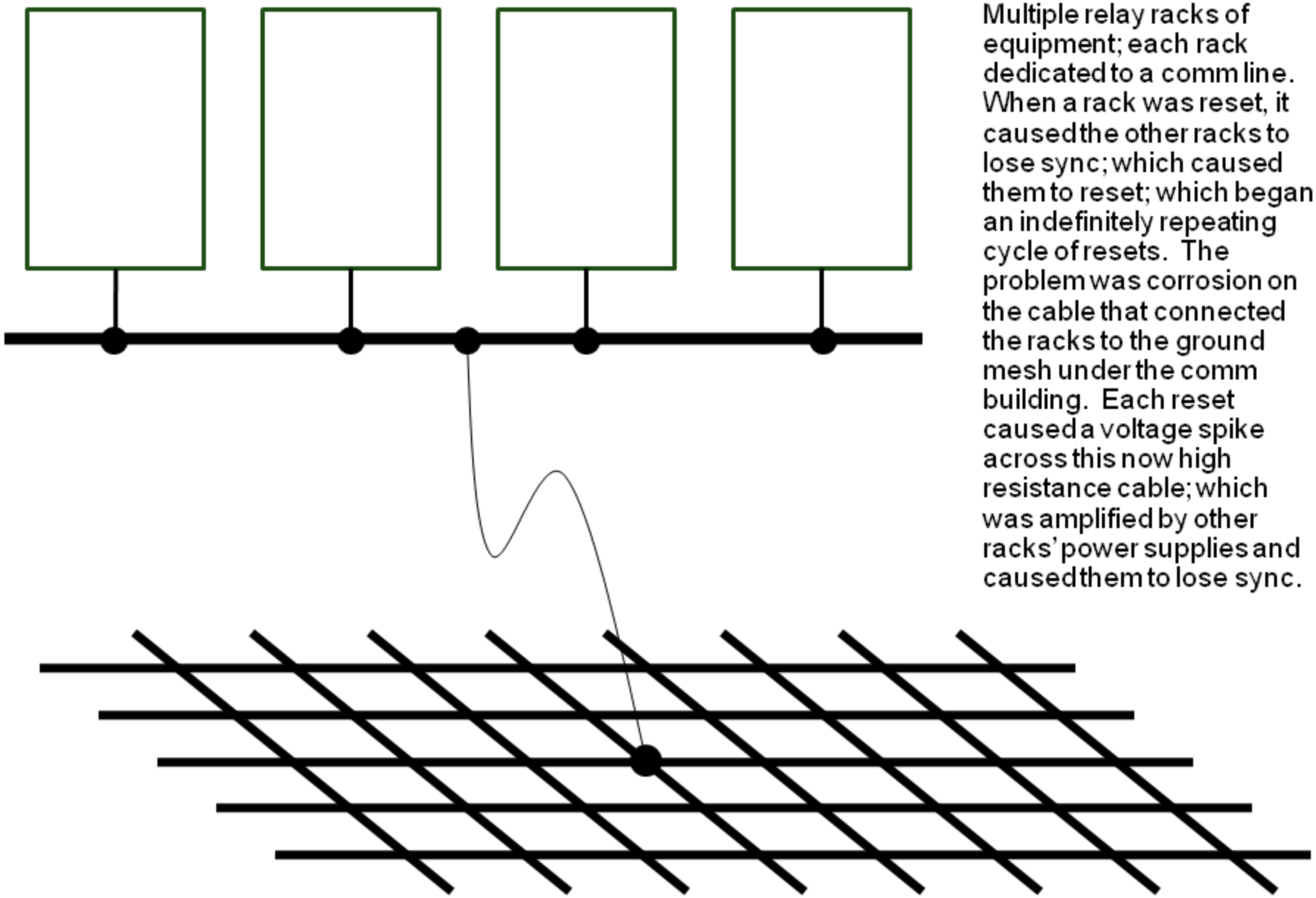




Multiple relay racks of equipment; each rack dedicated to a comm line. When a rack was reset, it caused the other racks to lose sync; which caused them to reset; which began an indefinitely repeating cycle of resets. The problem was ...

# “Sync Bandit”

Honeywell



Multiple relay racks of equipment; each rack dedicated to a comm line. When a rack was reset, it caused the other racks to lose sync; which caused them to reset; which began an indefinitely repeating cycle of resets. The problem was corrosion on the cable that connected the racks to the ground mesh under the comm building. Each reset caused a voltage spike across this now high resistance cable; which was amplified by other racks’ power supplies and caused them to lose sync.

# Other Ground(less) Problems

Honeywell

- A 60v (near lethal) difference in voltage between "grounded" cases of two boxes in an avionics lab gave a worker a sever shock
  - Two power sources: 115v 400Hz AC and 28v DC
  - The generators were location at the other end of the building
  - The two sources' "ground" return lines were independent
  - High currents through the ground lines caused the 60v difference
- In a high security equipment cabinet, problems were traced to the single-piece metal frame that was used as the equipment's ground
  - Parts of the metal frame were of out-of-phase with respect to other parts of the frame
  - How can a ground be out of phase with itself? Answer: eddy currents
- **The existence or absence of a ground near a high speed signal can effect that signal**
  - See http://ftp.sunet.se/jargon/html/magic-story.html

# List of Scenarios (3)

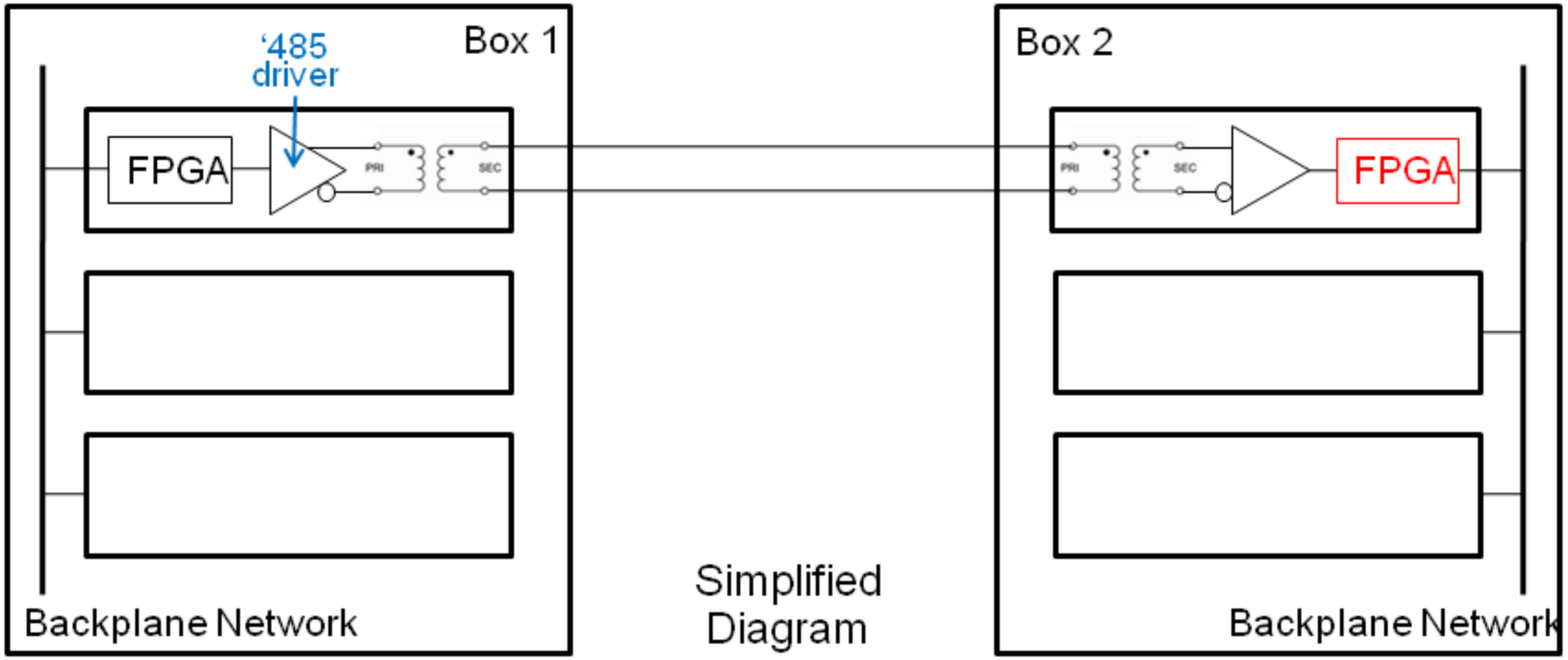
Honeywell

- Conversion of "stuck at" failures to oscillatory failures
  - RS-485
  - MIL STD 1553
- Transmogrification
  - Conversion
    - Space shuttle mission STS-124 [listed under Byzantine]
    - Input → Output
    - A frequency divider → multiplier
  - Partogenesis
    - Submicron IC
    - Avionics cooling
- "Solid-state reliability" isn't
- Mechanical (In)tolerance
  - Ignoring clearance specification
  - Quadrax geometry
- Flash EPROM
  - Incorrect busy test
  - Evaporating software
- Component Documentation Errors ($M^2FCS$ examples)
- Read the errata sheets

# An Oscillatory "Stuck At" Failure

Honeywell

- Our customer naïvely assumed only stuck-at faults (a common assumption) and needed to prevent fault propagation from box to box → the "solution" was to use transformers
- But, due to previous experience and our healthy respect for Murphy, we added an RX fault-tolerance FPGA to block arbitrary noise on the inter-box links
- During lab testing, an unconfigured TX FPGA created a stuck-at input to an RS-485 driver
  1. Constant drive into the transformer eventually looked like a short to the driver
  2. This triggered short-circuit protection in driver, which turned off the driver's output
  3. After 5 ms, short-circuit protection resets, go to step 1
- This converted a stuck-at into a 174 KHz signal that could have killed the second box

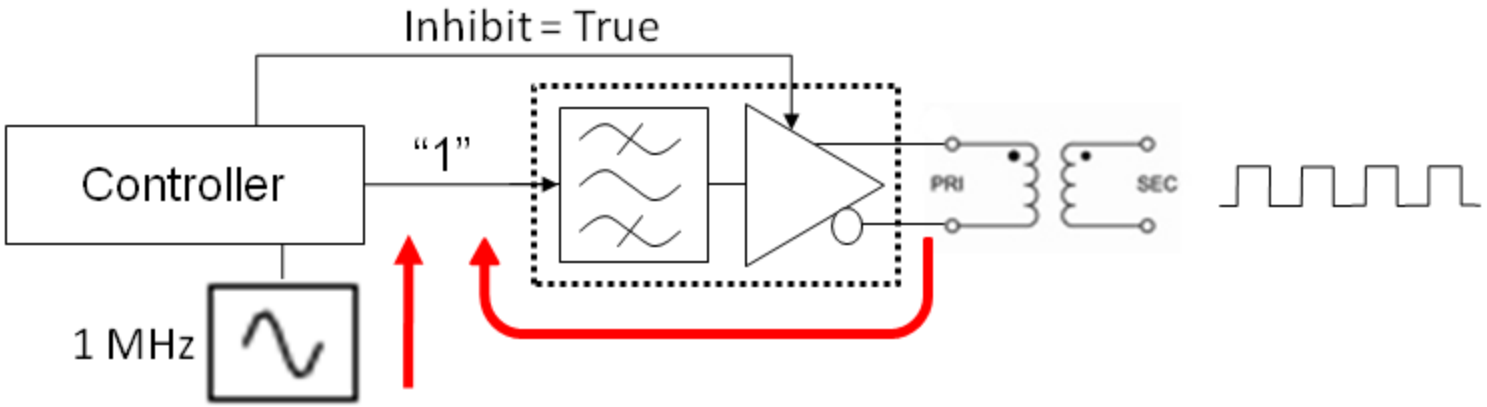



Simplified Diagram

# Another Oscillatory "Stuck At" Failure

Honeywell

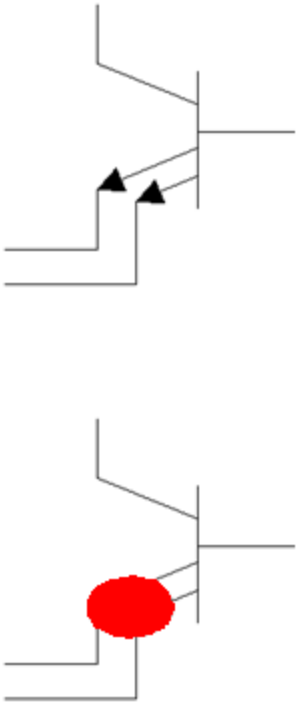
- A previous incident had a similar symptom, but a different mechanism
- A simple (no internal state) bipolar transistor based MIL STD 1553 bus driver continued to produce a perfectly good Manchester signal (low jitter, period accurate to 50 ppm), even when its output was inhibited and its input was a constant "1"
- The failing driver went into thermal runaway (a common bipolar failure mode)
  - Thermal runaway: increased heat → increased gain → increased current → increased heat
  - One obvious result is extremely high gain
- Because of the high gain, the fraction of the output signal that leaked back to the input was amplified until the driver went into oscillation
  - The conversion of an amplifier into an oscillator is a very common form of transmogrification
- To minimize EMI, the driver included a bandpass filter
  - This kept the oscillation within 1553's allowed bandwidth (0.5 to 1 MHz)
- The oscillation was further stabilized by noise from a local 1 MHz oscillator
- The result was a very stable 1 MHz square wave on the data bus
- Transmogrification sequence: Amplifier → oscillator → phase-locked-loop



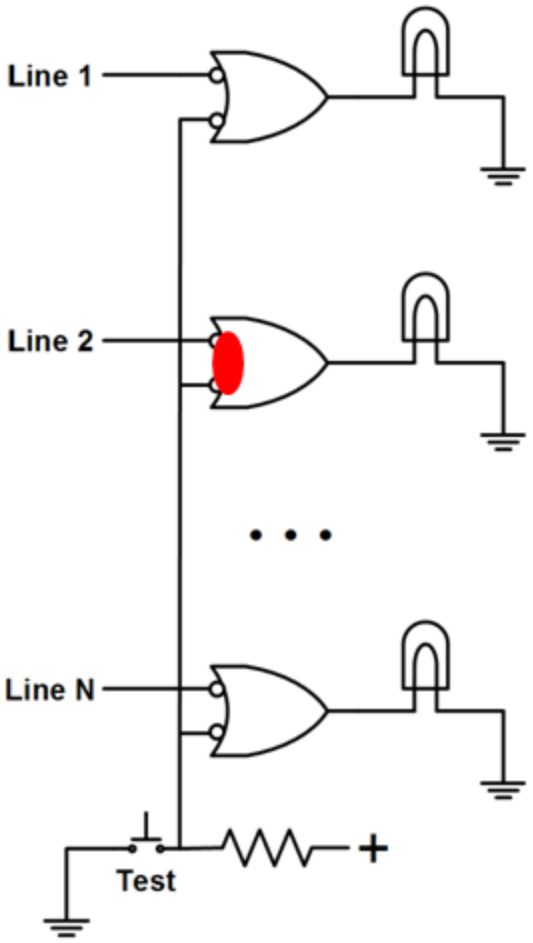
# Input ➔ Output, in a Security-Critical Voice Switchboard

Honeywell

Each input of a TTL OR-gate is connected to one of multiple emitters in a single transistor shared among all inputs. A failure shorted the emitters together without other effects to the remainder of the IC. Thus, a a signal coming in one input could go out the other input. It is easy to re-create this failure mode (e.g. connect the two wires of an idle phone line to the two inputs).



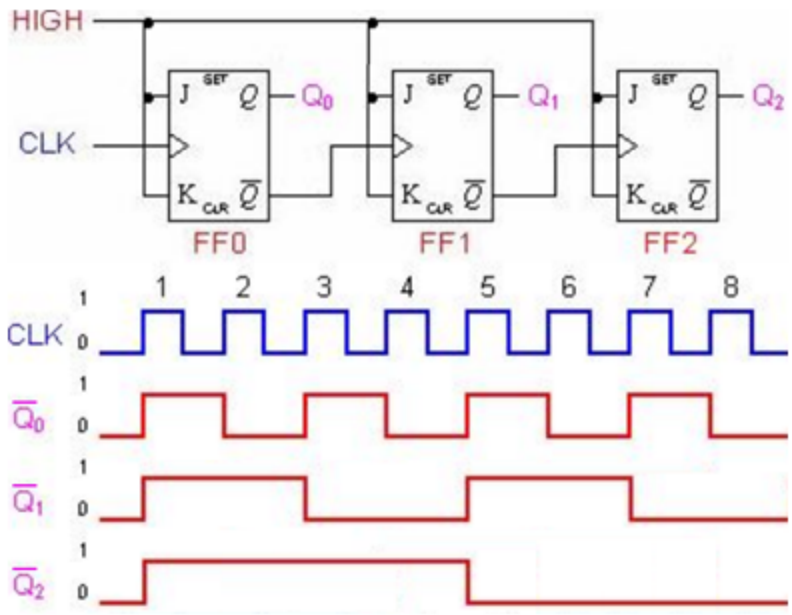
An active-low signal coming in on Line 2 coupled via the short to the other input of the OR-gate, causing the Test line to go low; which caused all the lights in a secure voice switchboard to turn on. This made it impossible to tell real incoming calls; which were critical at that time.

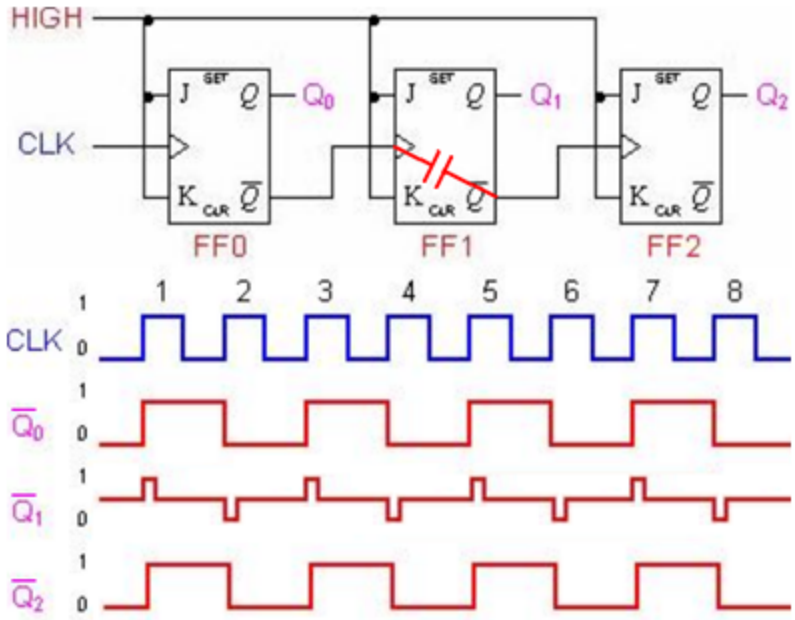


# Frequency Divider Becomes Multiplier

Honeywell

- A faulty divide-by-8 counter's output frequency was four times too high
- One of the counter's flip-flops multiplied its input frequency by 2 instead of dividing it by 2 (good circuit and waveforms shown at left below)
  - The fault caused a capacitive coupling between the bad flip-flop (FF1) clock input ($\overline{Q}_0$) and its output ($\overline{Q}_1$). See right-side figure below.
  - The square wave input to the capacitor caused a "doublet" waveform (which is typical for capacitive or inductive coupling) on its output that consisted of two spikes, one for the rising and one for the falling edge of the input
  - The rising edge of each spike caused FF2 to change state



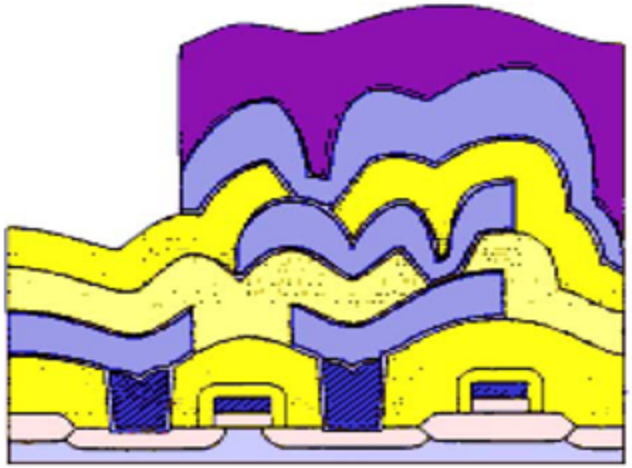
# A Partogenesis Example, Submicron IC

Honeywell

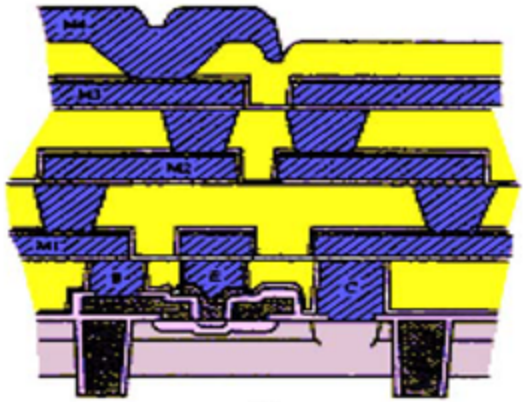
- Integrated circuit (IC) manufacturers often use "planarization" to reduce die surface roughness during processing, sometimes by filling in low spots with metal. Of course, any adjacent signal or power traces must be insulated from this metal.
- If a fault occurs in the insulation, a large block of metal can become attached to a trace → creating a capacitor.
- Typically, the only effect of this capacitor is to slow down the edges of the signal on the trace to which it is attached.
  - Might not be caught by testers that run at less than full speed (as is often the case for today's high-speed integrated circuits)
- The signal's added delay can make its operation marginal, failing only under certain conditions – temperature, voltage, fields from nearby signal traces, etc.
  - Can create Byzantine faults (more Byzantine faults are caused by timing than by amplitude)
- Of course, the fault can occur after the part has been in service for some time.

IC cross-sections:
(a) Without planarization
(b) With planarization

Figure taken from "Chemical mechanical planarization for microelectronics applications" Parshuram B. Zantye, et al.



(a)




(b)

# Avionics Cooling

Honeywell

- An avionics cooling system consisted of a simple fan and a plenum which fed air into three flight control boxes. Each box had a metal-mesh filter on its input.
- The system created an air conditioner.
  - The fan acted as a compressor.
  - The plenum acted as the condenser.
  - The filters acted as the evaporators.
- Caused the ~100% humidity air to condense on the backside of the filters, with water running down the backside of the filters.
- During a long ground hold, the water continued to condense out until the bottoms of the boxes (area under the mother boards) filled with water
- When the aircraft rotated on take off, the water sloshed back over the rear of the mother boards, causing massive shorts in all three flight control boxes.
- A fatal accident was narrowly avoided.

# "Solid-State Reliability" Isn't

Honeywell

- Current calculations for the reliability of electronics use a constant failure rate; that is, $R(t) = 1-e^{-\lambda t}$
  - This is based on the assumption that electronic devices do not wear out
  - But, this is no longer true; electronic devices do wear out
  - For wear-out, need to use a Weibull distribution: $R(t) = 1-e^{-(\lambda t)^{\beta}}$ with $\beta > 1$
- "Commercial semiconductor road maps show component reliability timescales are being reduced to 5–7 years,"
  - http://www.aerospace.org/2013/08/05/design-hardening-of-commercial-technologies
- Trilogy Systems found they couldn't make on-chip redundancy (e.g. TMR) relievable, due to "wide-area defects"; which was a major reason causing it to be the best funded (to that time) Silicon Valley startup to go bankrupt. Ignoring history, some still suggest that this type of on-chip redundancy be used as a solution for the decreasing IC reliability trend.
- The first 8 million Intel Cougar Point chips sold had a problem which would cause the chip to wear-out prematurely (about 3 years)
  - http://www.anandtech.com/show/4143/the-source-of-intels-cougar-point-sata-bug
- A330 entertainment system caught fire. Bad IC packaging (before 2000) allowed contaminants to penetrate the ICs' plastic cases and create an environment conducive to premature IC failure in a power IC.

# Ignoring Clearance Specification

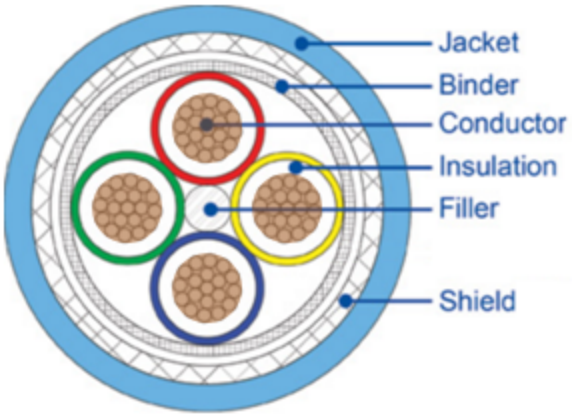
Honeywell

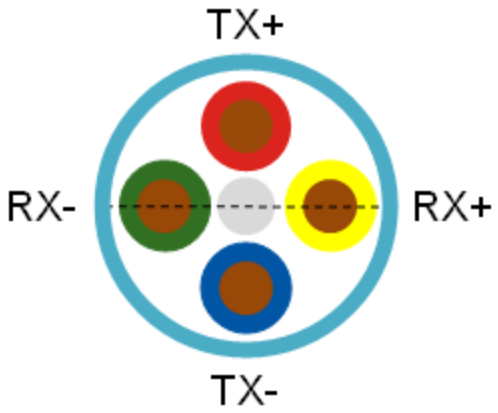
- Causes
  - Willful deviation from requirements and specifications
  - Cross-discipline ignorance
- Scenario
  - Clearance requirements from electronic components to metal cover ignored
  - Designer thought stated clearance was excessive
  - Designer didn't take into account
    - Cover becomes concave at altitude due to pressure differentials
    - Localized heating aggravates the concavity
  - Caused arching only after being at altitude (pressure deformation) for a significant period of time (additional temperature deformation)

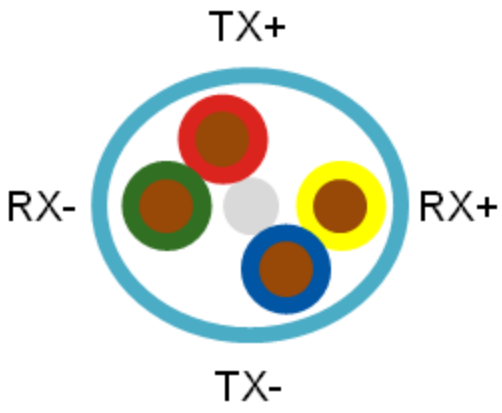
# Quadrax Geometry

Honeywell

- A 25m long quadrax cable (see cross-section in left figure below) carrying a 1 Gbps signal had a very high number of bit errors when half the cable was wound in a 0.5m diameter coil.
- There were no errors when the cable was uncoiled.
- The 0.5m diameter was much greater than the cable's minimum bend radius of 0.017m.
- The quadrax design requires that the TX and RX pairs be arranged so that the conductors of one pair lie on the plane equidistant from the other pair's conductors, e.g. the RX conductors in the middle figure below are exactly on the plane equidistant between the TX conductors (dashed line). This geometry causes the differential signal in one pair to cancel in the other pair (+ and – signals sum to zero on this plane), preventing TX-to-RX cross-talk.
- The outermost wire of a coil (e.g., the red wire in the third figure below, if the coil axis is below the figure and horizontal side-to-side) travels a greater distance path and the inner wire (e.g. blue) less distance than the other two wires. Because all the wires are the same length, the inner and outer wires get squeezed towards the middle. The greater the number of loops in the coil, the greater the accumulated path length differences and the greater the squeeze (even if the loop diameter stays the same). This distorted geometry couples cross-talk red-to-green and blue-to-yellow, causing interference and a greatly elevated bit error rate.
- Note that ARINC 664 Part 7 AFDX uses a type of quadrax cable.



# Intermittent Flash Memory Errors

Honeywell

- Customer was having problem with intermittent flash memory errors
- What questions should be asked? (appears to be a hardware problem)
- It was a software problem in the "flash busy test" procedure
  - Flash part pinout matched standard RAM, had no pin for a busy signal
  - Busy indicated by toggling a bit in the byte returned for consecutive reads
  - Procedure failed to detect this condition and exited too early
  - Subsequent flash reads were corrupted due to the continuation of the toggling
- The "flash busy test" procedure (should exit only when flash is not busy)

  A ← contents of some location x (in flash)

  B ← contents of same location x

  If A ≠ B, repeat procedure
- What else is needed to make this work?
  - Mark x as volatile, so that an optimizing compiler doesn't delete all this code
  - Disable cache
  - Flush cache (in case location x is already in the cache)
  - Between 1st and 2nd flash reads, read something not in flash
    - In order for the flash to see two separate reads vs. one read with a lot of wait states

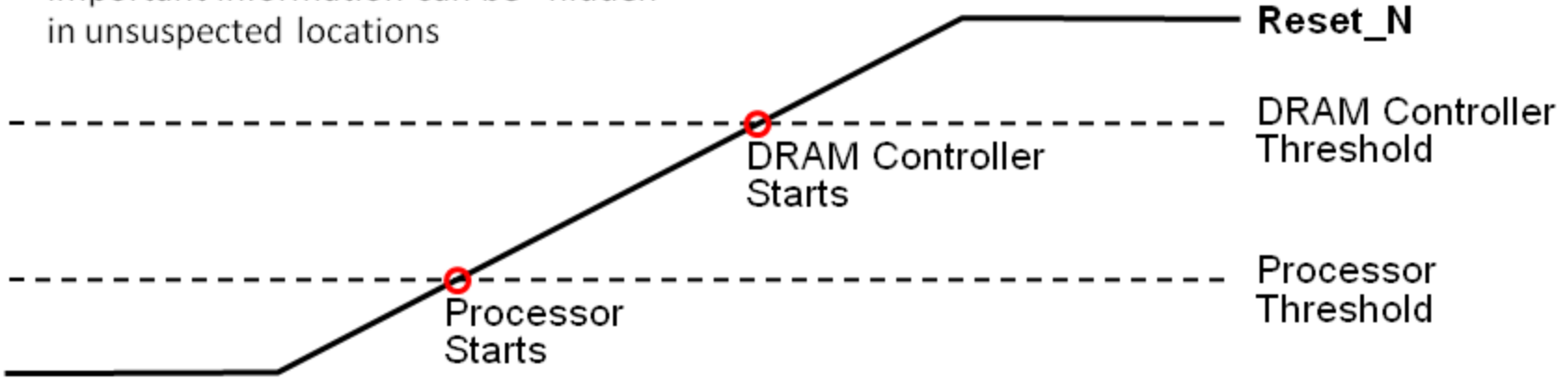
# “Evaporating” Software

Honeywell

- When reset at certain temperatures in an environmental test chamber, a military-quality processor had 1 (and only 1) of its tasks’ software “evaporate”. (numerous task rates, up to 1600 Hertz)
- The problem could not be duplicated outside of the environmental chamber.
  - Even when air was blown on the components with temperatures duplicating the test chamber
- Its DRAM controller had several programmable refresh rates, including 0, 50, 60, and 70 Hertz.
  - The datasheet said that the power-on default was 50 Hz, which was an acceptable rate.
  - Because 50 Hz was acceptable, the software programmer did nothing to set the rate.
  - The programmer did not read a later page of the datasheet, which said:
    Then, on the rising edge of Reset_N, refresh rate is set by the upper two address bus bits.
- The processor exited reset and started changing the address lines before the DRAM controller started (see figure). For certain values of the upper address bits, the refresh was turned off.
- Rise time of Reset_N varied by load capacitance and temperature; creating a Heisenbug.
  - Load capacitance greatly increased when the package was assembled; rise time increased ~100x
- Most software ran at 40 Hz or more, which was good enough for it to self refresh.
  - The victim subroutine ran at a rate too slow to self refresh

➔ ***Programmers should read the entirety of all documentation***

  - Important information can be “hidden” in unsuspected locations



# Component Documentation Errors

Honeywell

- M²FCS examples
  - **Intel UART**
    - Didn't document the 26 clock delay in reading the RS-232 Data Terminal Ready (DTR) pin after start up
  - **Intel Timer**
    - Documentation lied about having three capture registers (there was only one, with three different names)
    - Caused synchronization failure between redundant processors
  - **The single 8086 AAM instruction was either 1, 2, or 4 bytes long, depending on which chapter of the manual you read.**
  - **Reponses from Intel included: "We don't know why the part doesn't work as documented, the designer(s) of that part have left the company."**
- For more than 30 years, our design lab has seen that no IC greater than 16 pins (except memory) has worked according to its documentation

# Read The Errata Sheets

Honeywell

- A microprocessor with a built-in direct memory access (DMA) controller wrote data to the wrong locations
  - If the sequence of input data was a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q...
  - **Desired Result**

| Address | Value |
|---|---|
| 0 | a |
| 1 | b |
| 2 | c |
| 3 | d |
| 4 | e |
| 5 | f |
| 6 | g |
| 7 | h |
| 8 | i |
| 9 | j |
| 10 | k |
| 11 | l |
| 12 | m |
| 13 | n |
| 14 | o |
| 15 | p |
| 16 | q |
| … | … |

**Actual Result**

| Address | Value |
|---|---|
| 0 | a |
| 1 | b |
| 2 | c |
| 3 | d |
| 4 | -- |
| 5 | -- |
| 6 | -- |
| 7 | e |
| 8 | -- |
| 9 | -- |
| 10 | -- |
| 11 | -- |
| 12 | -- |
| 13 | -- |
| 14 | -- |
| 15 | f |
| 16 | -- |
| … | … |

  - The carry chain in the DMA's address increment adder was too slow
- Problem was described in the errata sheet …
  - … that the designers weren't given

# List of Scenarios (4)

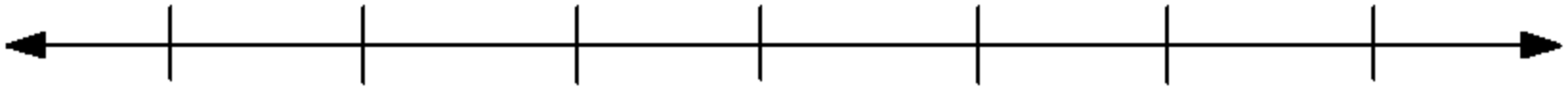
Honeywell

- **Dangerous Interrupts**
  - Floating point in interrupt handling routines
  - The "missing link" in task ready queue
  - Stack lockup
- **Mismatched assumptions**
  - Wrong units or bad units conversions
    - Many well known cases (just list a couple of links here)
    - "Gimli glider" (http://en.wikipedia.org/wiki/Gimli_Glider)
    - Mars Climate Orbiter (http://en.wikipedia.org/wiki/Mars_Climate_Orbiter)
  - Inexact counters
    - Patriot missile (http://mate.uprh.edu/~pnm/notas4061/patriot.htm)
    - Five second delay, not exactly
  - Compiler, library, linker, locator, loader
    - Recompile produced different results
    - "Completely" tested software
    - Linker magic string
      - Linker assumed no op-code could have a string of zero bytes more than a certain length
      - New processor broke that assumption; caused linker to think op-code was unsatisfied external
- **Mechanical generic failures**
  - New alloys in plumbing and resonance in hydraulic caused both hydraulic systems to fail on an aircraft
- **Willful deviation from requirements and specifications**
  - Mechanical subcontractor used wrong lubricant
  - Ignoring clearance specification, listed under Mechanical (In)tolerance

# Most Intermittents are Caused by Asynchronism

Honeywell

- Synchronism versus Asynchronism is not dichotomous.



micro cycle clock | instruction | serial bit | serial message | minor frame | major frame | external closure

- Synchronism occurs in multiple dimensions, e.g.
  - In pipeline from input to output
  - Across redundancies
- Typical asynchronism problems are hardware / software coupled
  - Interrupts (should be avoided like the plague)
  - Designs should use only two interrupts
    - Clock tick (for task scheduling)
    - Fatal (e.g. imminent power failure, there is no return from Fatal)

# Floating Point in Interrupt Handling Routines

Honeywell

- A fly-by-wire system used floating point operations in its interrupt handling routines.
- Programmer made sure that critical regions were observed and all floating point registers used in the routines were saved and restored.
- But, some floating point operations were not atomic! If the interrupt occurred in the middle of one of these operations, some hidden state would be lost.
- Very difficult problem to find because the symptom was an intermittent small loss of accuracy in a set of variables, but these integrated into a sizeable problem over time.

# The "Missing Link" in Task Ready Queue

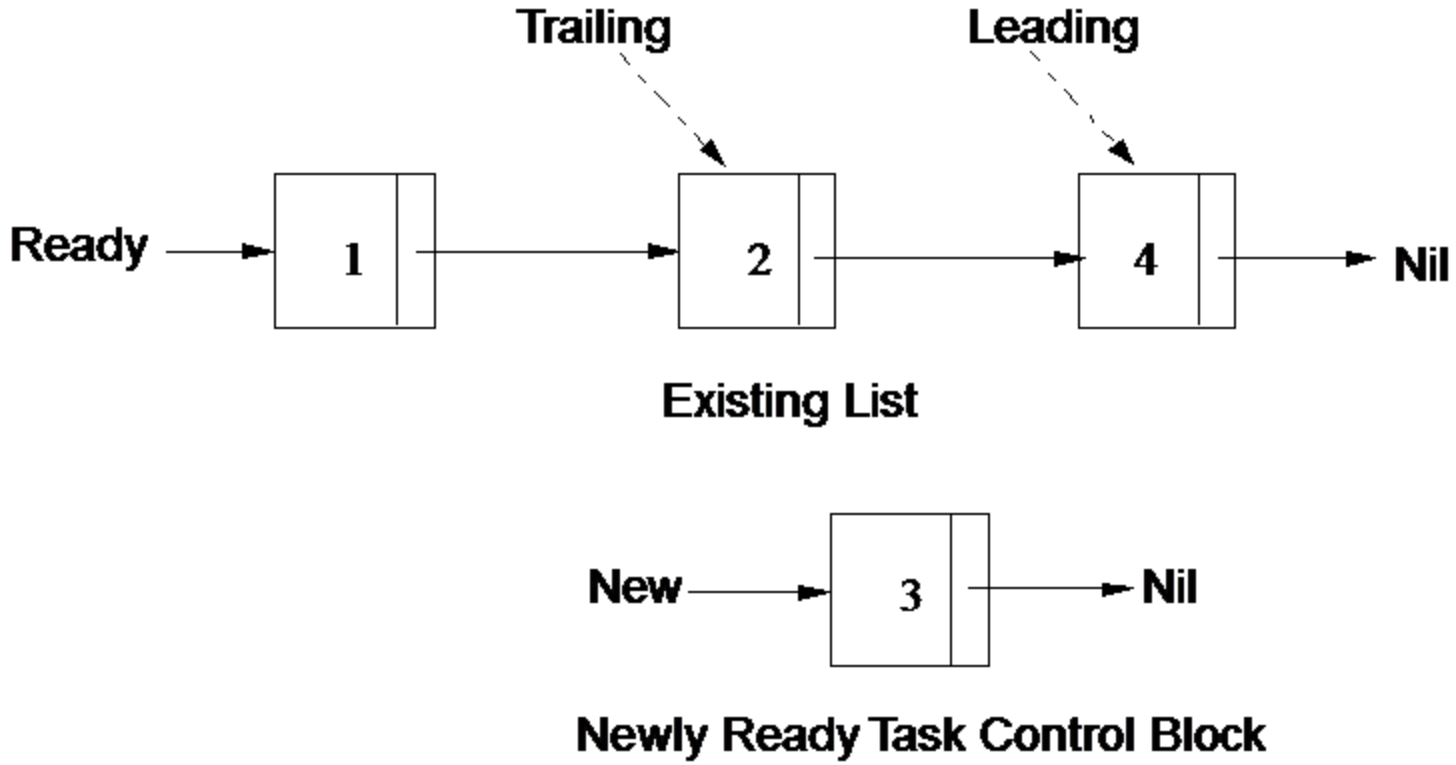
Honeywell

- **Symptoms:**
  - Intermittent problem, two to eight hour mean time between failure (MTBF)
  - Military-quality processor would "lock up"
  - Not even the front panel reset button worked
  - Had to disconnect the power supply
- **How do you troubleshoot this?**
  - Appears to be a hardware problem
  - Multiple tasks running at 20, 40, and 80 Hertz
  - A couple of asynchronous tasks (bad idea)
- **Due to bad operating system design**
  - An interrupt service routine updated the operating system's ready task queue without using critical region locking
  - It also used an unwise order in updating the linked list's links

# Condition of Ready List Before Links Are Updated

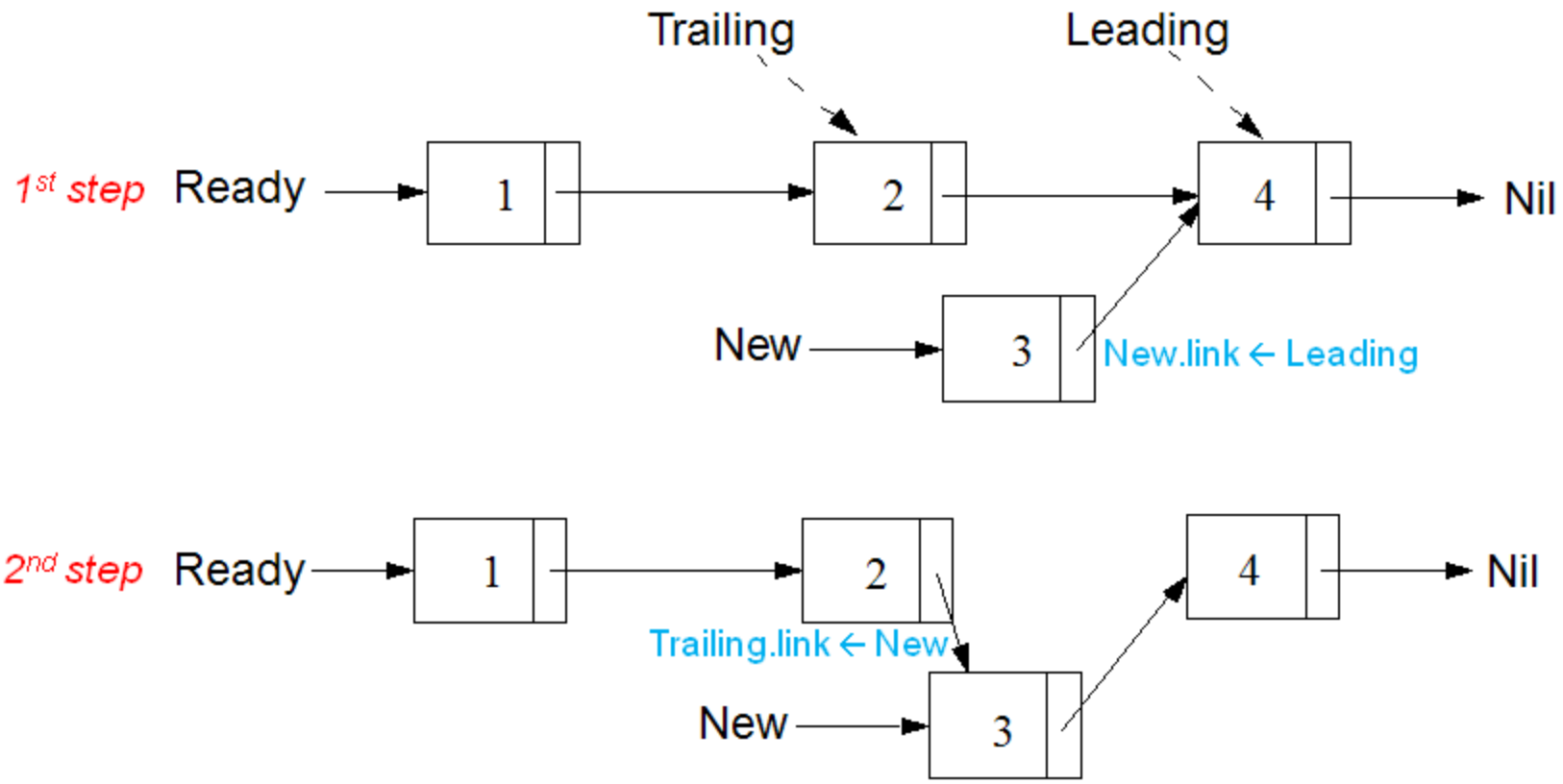
Honeywell

*Used typical linked list search procedure where two pointers (Leading and Trailing) advance through the list until the insertion point is found between them. In this example, we want to insert block 3 just after block 2.*



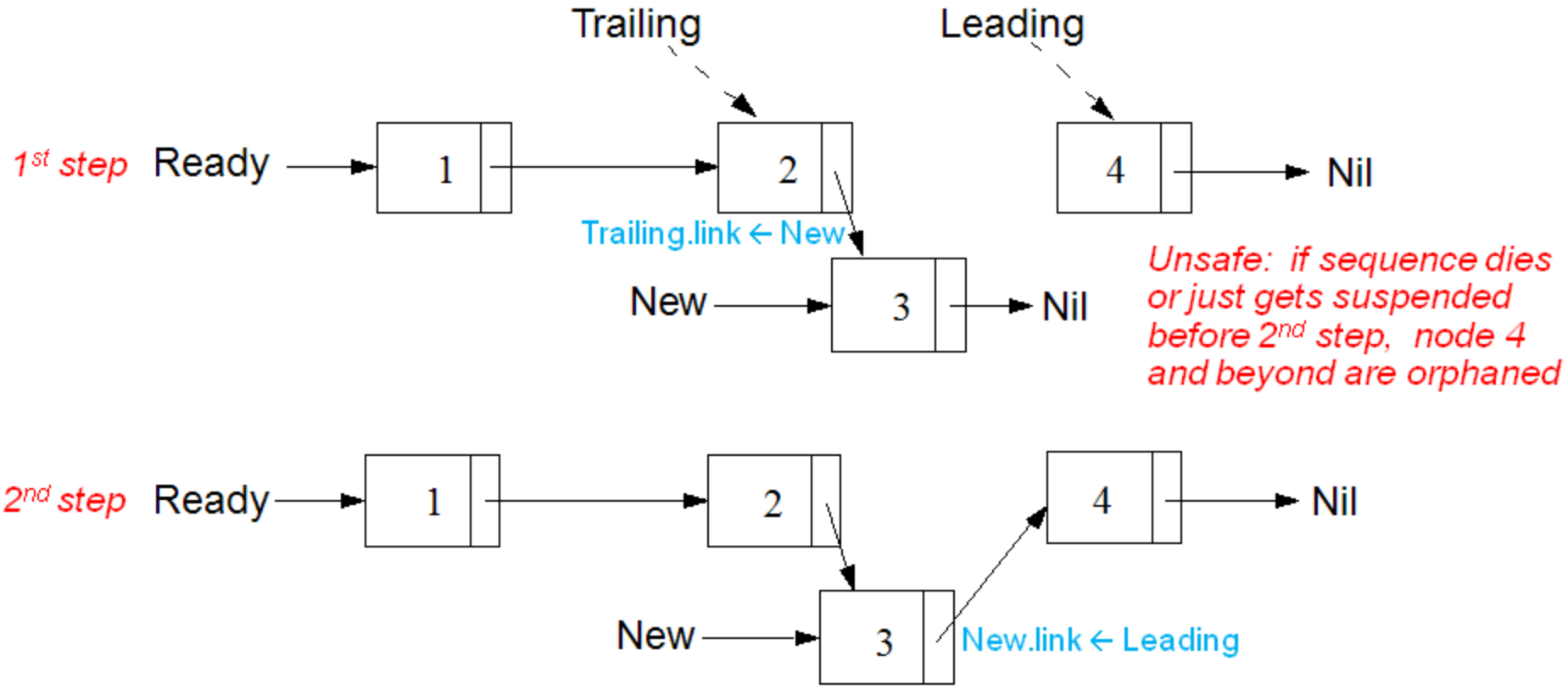
# Safe Way of Updating Links

Honeywell



# Unsafe Update that Succeeds (Lucky)

Honeywell

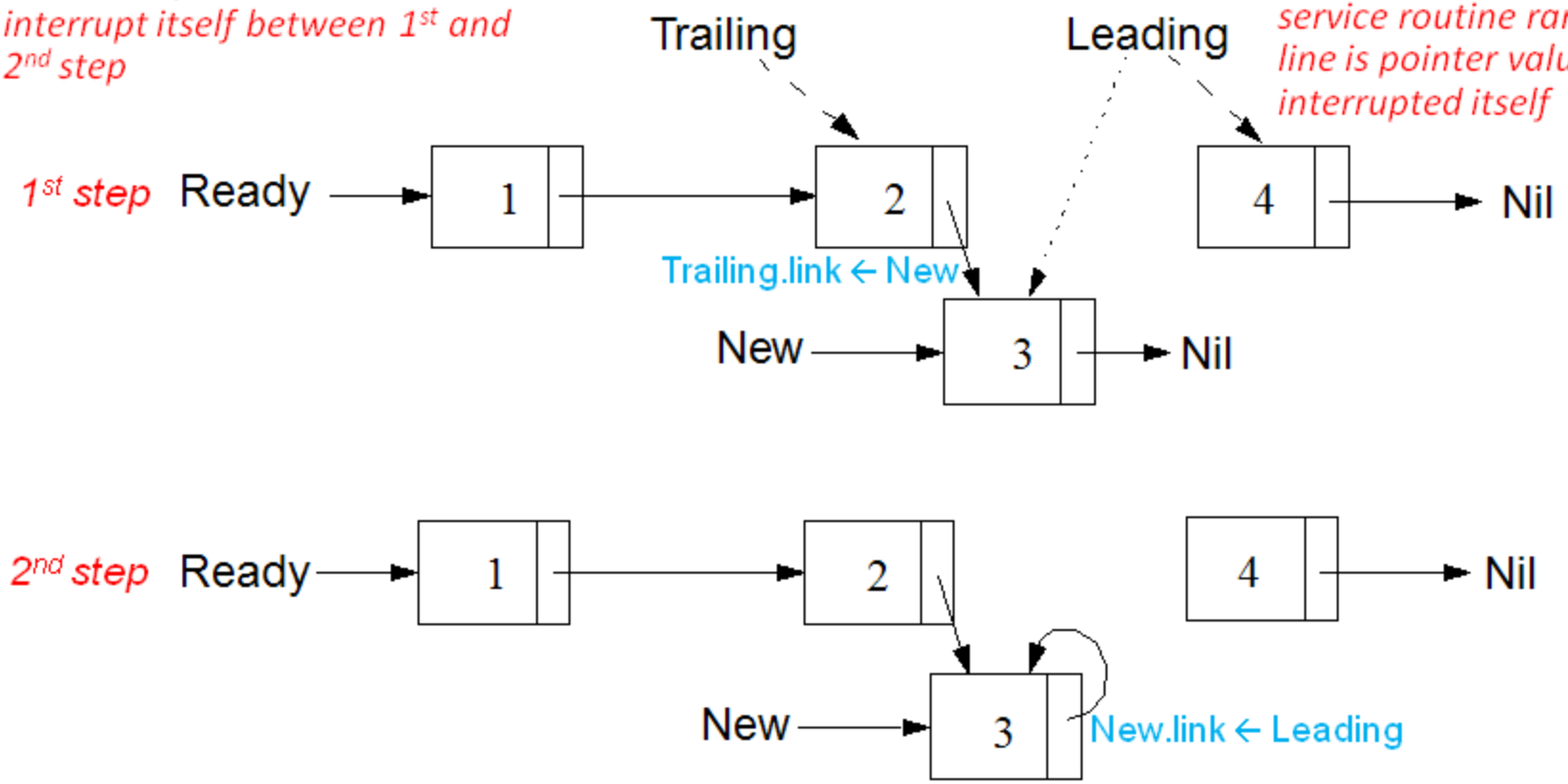


# Unsafe and (Very) Unlucky

Honeywell

*Interrupt signal "bounce" caused the interrupt service routine to interrupt itself between 1st and 2nd step*

*The dashed line is pointer value for 1st time interrupt service routine ran, dotted line is pointer value after it interrupted itself*




- Linked list search was a single instruction (microcode sequence)
- Having new task control block point to itself caused a microcode infinite loop
- Front panel was "virtual" (polled by microcode); but not polled in the linked list search loop
- This is a "Halt and Catch Fire" data structure
  - Similar to "Halt and Catch Fire" CPU opcodes (see, http://en.wikipedia.org/wiki/Halt_and_Catch_Fire)
- ***Can robust partitioning be claimed if a processor can have "Halt and Catch Fire" data structures and/or op-codes?***

# Stack Lockup

Honeywell

- To add a space for N words (two bytes each) on the stack in a kernel mode routine, an Ada compiler generated assembly code that consisted of the following two instructions
  - Subtract N from the kernel stack pointer
  - Subtract N from the kernel stack pointer
- This was more efficient than multiplying N by 2 and then doing a single subtraction
- These two instructions were supposed to be atomic
  - Interrupts were masked, upon entry into kernel mode
  - But, there was one special case interrupt that couldn't be masked
- Occasionally, the unmasked interrupt was raised between these two instructions
- When the interrupt servicing tried to push data unto the stack, the processor locked up due to the stack pointer being an odd value

-- story courtesy of Devesh Bhatt

# Five Second Delay, Not Exactly

Honeywell

- A function in a safety-critical avionics application had an iteration rate of $\chi$ Hz and needed to delay for 5 seconds
  - A simple way of doing this would be to hold the delay for 5 * $\chi$ iterations
  - Instead, the software added 1/$\chi$ to a time accumulation variable during each iteration and checked to see if the time accumulation variable's value exceeded 5 seconds
  - The 1/$\chi$ value couldn't be represented exactly in IEEE 754 floating point format
    - The time accumulation variable's value couldn't be checked for exact equality to the 5 seconds delay value
    - The inexact comparison between the time accumulation variable's value and 5 seconds added timing jitter into the software execution

# Not Quite Identical Recompile

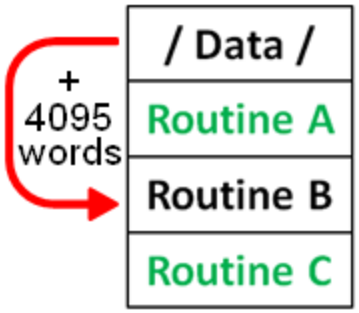
Honeywell

- **A regression test of a software module failed**
  - When all of the changes were removed, its behavior still didn't match the original
  - Nothing was changed in the compilation environment
- **Problem was in the compiler**
  - Used heuristics for optimization
  - Heuristics seeded by a random number for each compile?!
  - Floating point doesn't obey many mathematics laws
    - E.g. Associative, Commutative, Distributive
    - Order of execution differences can produce different result values
- **Compiler subsequently was fixed**

# "Completely" Tested Software

Honeywell

- Two Fortran routines (A and C) were exhaustively tested (every possible combination of internal state and input values)
- Routine B was not exhaustively tested
- The system had no timing problems
- Routine B started misbehaving after a trivial change to A (and only A)
  - Which of the routines was causing the problem?
  - Of course, it was C ("can't happen": exhaustively tested, not changed)
- Routine C contained calls to a library that did bit manipulation
  - The library included routines to set, clear, and test bits within a word
  - The two arguments to each of these library routines were the same: (1) a memory address and (2) the bit number within that memory location
- The programmer knew that this machine's bits were numbered 0 to 15, left to right
- What the programmer didn't know was that the compiler writer viewed the bits as an array and Fortran arrays always begin with 1. So, the compiler subtracted 1 from the bit number before creating the machine instructions. The compiler didn't check for bit number being 0.
- On this 16-bit machine, attempts to access bit 0 actually accessed bit $2^{16}-1 = 65,535$
  - After reaching bit 15, the microcode kept counting bits, rolling over into subsequent words
  - Thus, bit 0 was mapped to a location $\lfloor 65,535/16 \rfloor = 4095$ words beyond the intended word
- In previous compilations of A, these mis-mapped bits put the target bits in unused bits of B
- The trivial change to A changed its size and the relative position of B with respect to the /Data/ Common Block; which then mapped one of these bits to a used instruction bit
- This processor's Access Protection Module implemented the memory protection normally done in a Memory Management Unit, but the address granularity was too large to differentiate between the /Data/ Common Block, Routine A, and the beginning of Routine B.



# List of Scenarios (5)

Honeywell

- **Organic Growth**
  - **ILS Signal "In the Weeds"**
  - **Growth of organisms within fiber optic (FO) connectors**
    - Needs only the FO supplied light and some hydrocarbon contaminants in order to grow
    - Can lead to gradual degradation of the FO signal that causes undetected errors
      - For errors in gigabit FO Ethernet, see these Laura James publications:
        - » Passive and Active Measurement Workshop (PAM) 2004
          *Structured Errors in Optical Gigabit Ethernet*
          L B James, A W Moore, M Glick
        - » London Communications Symposium (LCS) 2004
          *Beyond Gigabit Ethernet: Physical Layer Issues in Future Optical Networks*
          L B James, A W Moore, R Plumb, M Glick, A Wonfor, I H White, D McAuley and R V Penty
        - » Optical Fiber Communications Conference (OFC) 2005
          *Packet error rate and bit error rate non deterministic relationship in optical network applications*
          L B James, A W Moore, A Wonfor, R Plumb, I H White and R V Penty
        - » IEEE Communications Magazine, August 2005
          *Chasing errors through the network stack: A testbed for investigating errors in real traffic on optical networks*
          A W Moore, L B James, M Glick, A Wonfor, I H White, D McAuley and R V Penty
        - » IEEE Journal of Lightwave Technology To appear October 2005
          *Optical Network Packet Error-Rate due to Physical Layer Coding*
          A W Moore, L B James, M Glick, A Wonfor, R Plumb and I H White
        - » PhD thesis, Department of Engineering, University of Cambridge
          *Error Behaviour in Optical Networks*
          Laura Bryony James
        - » Passive and Active Measurement Workshop (PAM}
          *Physical Layer Impact upon Packet Errors*
          Laura B. James , Andrew W. Moore , Madeleine Glick , James Bulpin

# ILS Signal "In the Weeds"

Honeywell

- Two planes nearly crash at Israel's main international airport, Ben-Gurion Airport in Tel Aviv
- The airport's authority said in a statement: "On June 3, 2009, Ben-Gurion Airport was operating under poor visibility due to heavy fog in the area of the airport. Two airplanes, belonging to El Al and Israir, attempted landings on Runway 26, which is equipped with ILS. As they were approaching the ground, a traffic controller at the control tower alerted the pilots to the possibility of error in their angle of approach. They understood the error, but couldn't land at the airport due to the poor weather. The slight malfunction in the ILS was fixed that very day. Two calibration flights confirmed that the equipment is now fully operational."
- Thick, tall vegetation in the area of the ILS antennas obstructed its signals. Initial attempts to reset the instrument failed. It returned to normal operation only after the ground around it had been cleared.

# List of Scenarios, to aggregate

Honeywell

- **If a problem is well known and has many examples, it is better to aggregate them into a single description**
- **Premier among these is the blockage of pitot tubes**
  - **Blocked due to icing**
    - Air France flight 447
    - Austral flight 2553
    - Northwest flight 6231
    - U.S. Air Force A-12 article 123 (serial number 60-6926)
    - …
  - **Didn't remove tape after washing**
    - Aero Peru 603
  - **Mud-dauber wasps building nests in the tubes**
    - F-16
      - Newly installed Flight Control Maintenance Diagnostic System warned that all three pitot tubes had failed at takeoff, but the pilot didn't believe it; returned safely
    - Probable cause of Birgenair flight 301 crash
    - Wasps in Brisbane caused five A330s and three B777s to abort take-offs, in 2006
  - **Boeing's list**
    - **Mud Daubers**
    - **Icing**
    - **Volcanic Ash**
    - **Hail**
    - **Radome failure**
    - **Birds**
    - **Pitot covers**
    - **Taped Static Ports**
    - **Maintenance errors (pneumatic plumbing)**
    - www.ata-divisions.org/S_TD/pdf/other/IntroducingtheB-787.pdf

# List of Scenarios, not included

Honeywell

- No lessons to be learned that aren't already known
- (Unavoidable) human error
  - Flight crew
  - Maintenance (A-12 mis-wire, DC-10 oil plugs)
  - ATC (Cali, Warsaw, Uberlingen)
  - Other organizational issues
- "Stuff happens" (not economical to avoid by design)
  - YF-12A fuel leak fire
- "Y" fighter PIOs
  - All had this problem, in spite of design advances and experience
  - Some are listed in http://en.wikipedia.org/wiki/Pilot-induced_oscillation