Murphy Was an Optimist

Kevin R. Driscoll

Kevin.Driscoll@Honeywell.com

Murphy's Law

- Murphy's Law says:
  "If anything can go wrong, it will go wrong."
- Revised for Critical Embedded Systems:
  "… and, if anything *can't* go wrong, it will go wrong anyway."

- NASA's C. Michael Holloway (after some studies of accidents):
  "To a first approximation, we can say that accidents are almost always the result of incorrect estimates of the likelihood of one or more things."
  - We fatally underestimate the probabilities of what can happen.
  - Designers will say a failure mode is "not credible", when it actually can happen with probabilities far greater than requirements allow.

Why?

- William Shakespeare's Hamlet (Act I, Scene 5):
  "There are more things in heaven and earth, Horatio, than are dreamt of in your philosophy."

Requirements Are Beyond Our Experience

- A typical designer's total hands-on system experience time is almost non-existent compared to typical system requirements and fleet exposures.
  - Safety-critical systems usually require the probability of failure to be less than $1/10^7$ to $1/10^9$ for a one hour exposure (= 10^7 to 10^9 hours MTBF, sort of)
  - A typical designer (20 years into a 40 year career)
    - Has less than 5,000 hours of real hands-on system experience
      … and, almost none of this is in the system's real environment
      (When was the last time you saw a designer riding in an electronics bay of an aircraft?)
    - Sees a negligible number of failed units returned from the field
      - 90% are transient and not returned: retest OK (RTOK), cannot duplicate (CND)
      - Most units are repaired (or scrapped) without being returned to the manufacturer
      - Manufacturers' failure analysis teams are not the designers
- So, when a designer says that the failure can't happen, this means that it hasn't been seen in less than 5,000 hours of observation
- But … 5,000 is minuscule compared to 10,000,000 or 1,000,000,000
  - And, when compared to total fleet exposure times, this is less than
    - A few days of flight time for any popular aircraft type (e.g. B737, A320)
    - One day of drive time for any popular automobile type

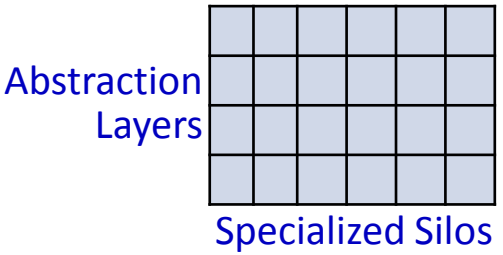
➔ ***We cannot rely on our experience-based intuition to determine whether a failure can happen within required probability limits***

Why not use the literature to gain virtual experience?

- Designers generally aren't given enough time to read ("academic") papers
- Papers about real occurrences of rare faults are difficult to find
  - Designers aren't given enough time to write papers
  - Repair people (who have more field experience than designers)
    - Never(?) write papers about failures
    - Don't have insight into causes, just symptoms
  - Organizations/people don't want to admit failures
    - Counter-arguments: moral, marketing (pro's for publishing, con's for not), legal?
  - The feeling that "once-in-a-lifetime" faults are not worth reporting
    - Most potential authors have this feeling ***(the biggest reason for not publishing)***
    - Some referees/reviewers have the same feeling (even less experience than designers)
    - Remember that an 40-year "lifetime" of experience is less than 10,000 hours
  - Some referees have "If I don't know about it, it doesn't exist" arrogance
  - Hard to find popular venue and category for papers about single failures
    - Not enough material (1 to 2 pages) for a paper, even for an "Experience" paper
    - Not really a "Quick Abstract"
    - Best fit is a "Note" (but, where?)
    - Risks Forum (Digest): web page, email list, comp.risks newsgroup
      - » Not a peer-reviewed publication (some posts are published in journal columns)
      - » Most contributions are hearsay and don't have enough detail to be useful

➔ ***Treat each paper about real faults as a priceless source of meditation***
– Try to think about other failures with similar causes or symptoms

Failures on the Borders of our Domain



- To manage increasing complexity, we:
  - Abstract
    - ❖ Partition the problem into layers of less/more detail
    - ❖ Failures can happen in layers that designers don't see
      - ➢ An effect crossing a sufficient number of layers is indistinguishable from magic* (as we shall see in later slides)
  - Specialize
    - ❖ Partition the problem into slices (silos) of different technology disciplines
    - ❖ The number of disciplines continues to proliferate
      - » Hardware
        - Power Supplies
        - Digital
        - Analog
          - Audio
          - Control system sensors
          - RF
      - » Software
        - Platform / Infrastructure
        - Applications (there's no end to specialized in applications)
    - ❖ Failures can hide in the "cracks" between disciplines
      - ➢ More disciplines → more cracks

* Apologies to Arthur C. Clark and his 3rd law of prediction

Portrait of a Byzantine Assassin

The Strange Case of STS–124

The Byzantine Generals Problem

- 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?

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.

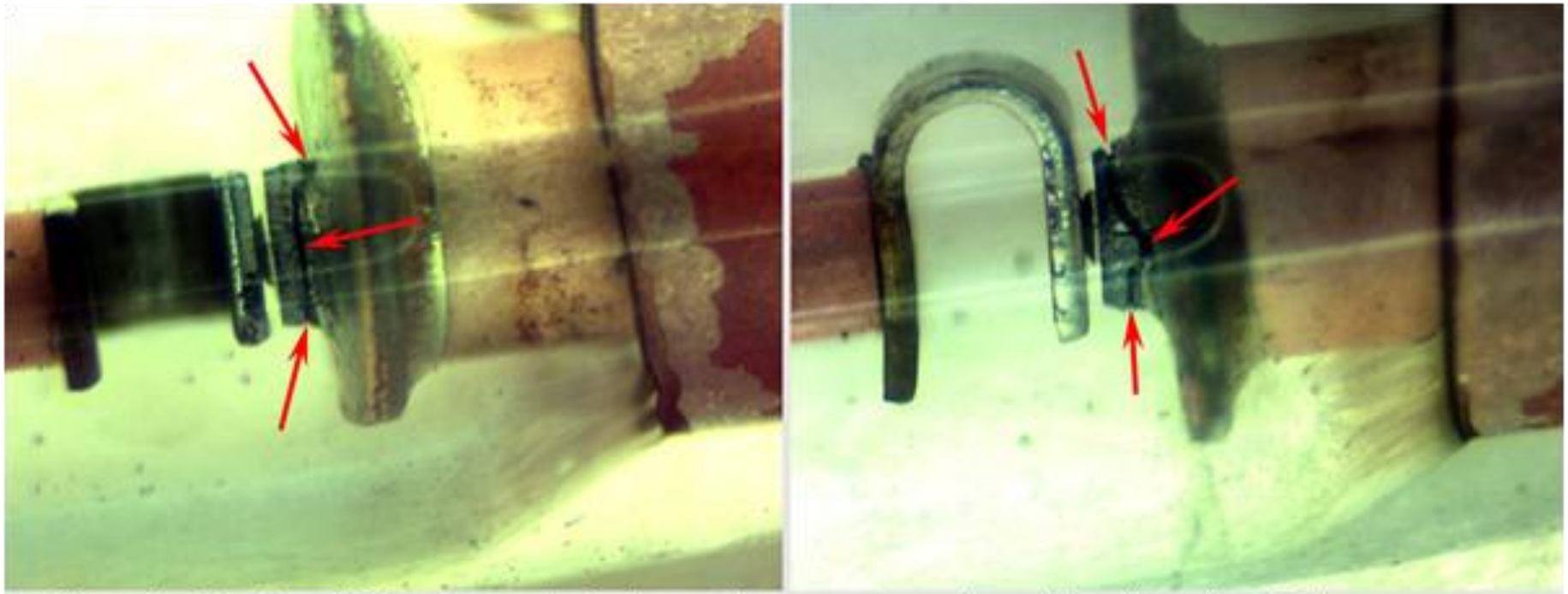
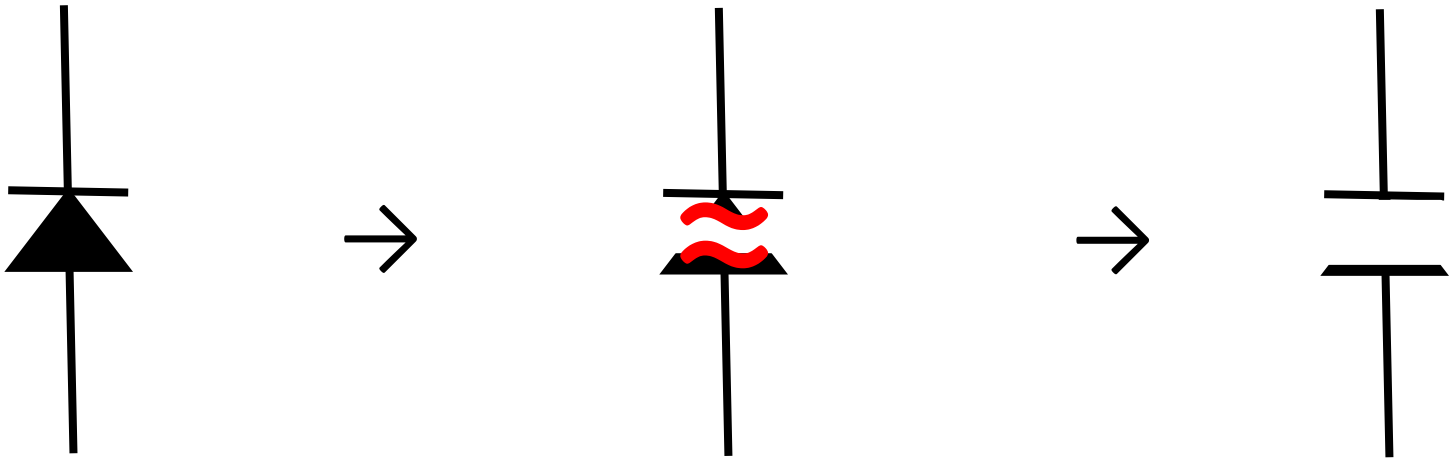


Figure 1. Two views (90 degrees apart) of a fissure that appears to go through the silicon - Red arrows.

* the Byzantine Assassin

Transmogrification*



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, …

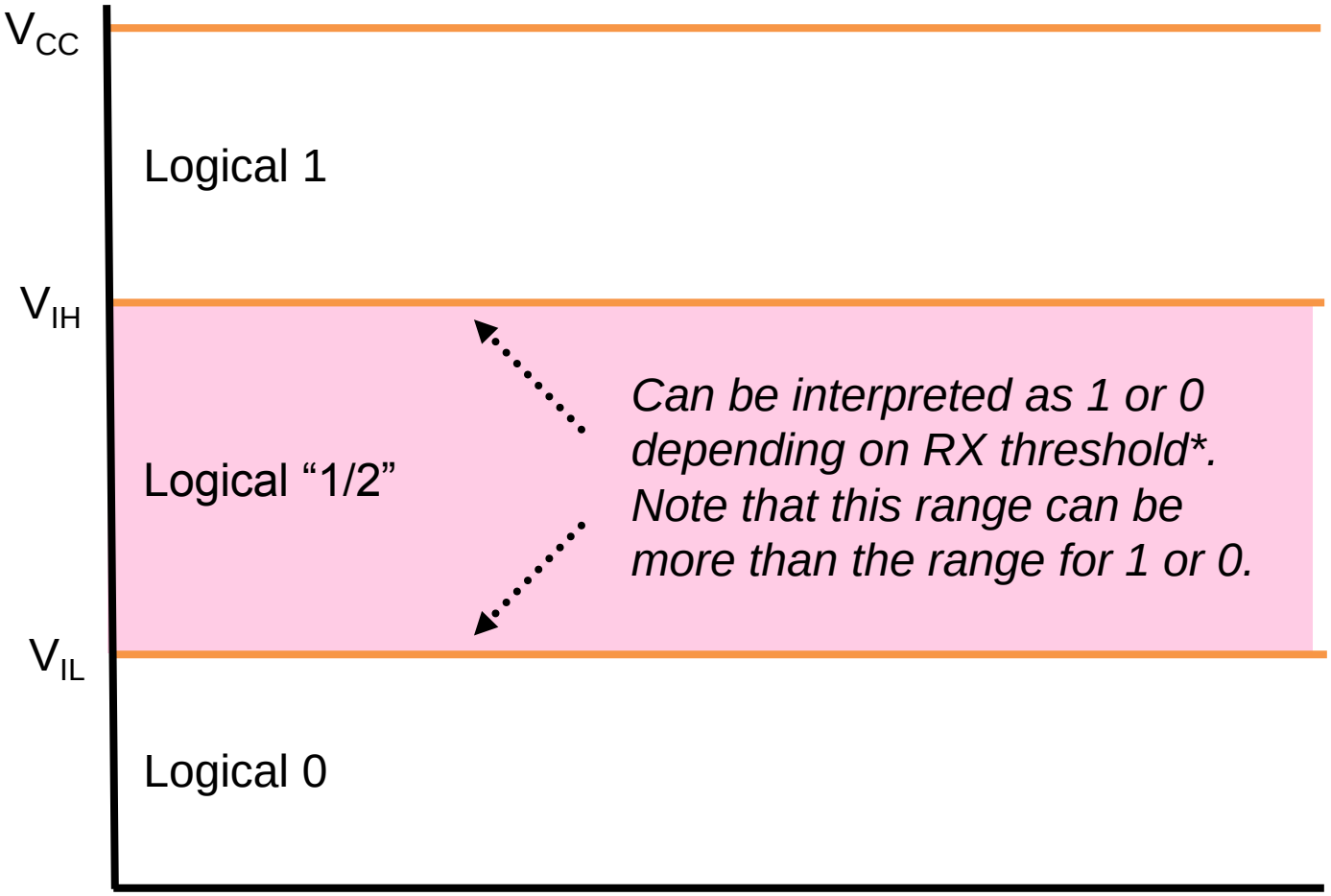
* Transmogrification definition: the act of changing into a different form or appearance (especially a fantastic or grotesque one), often as if by magic

Digital Circuitry Behavior

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

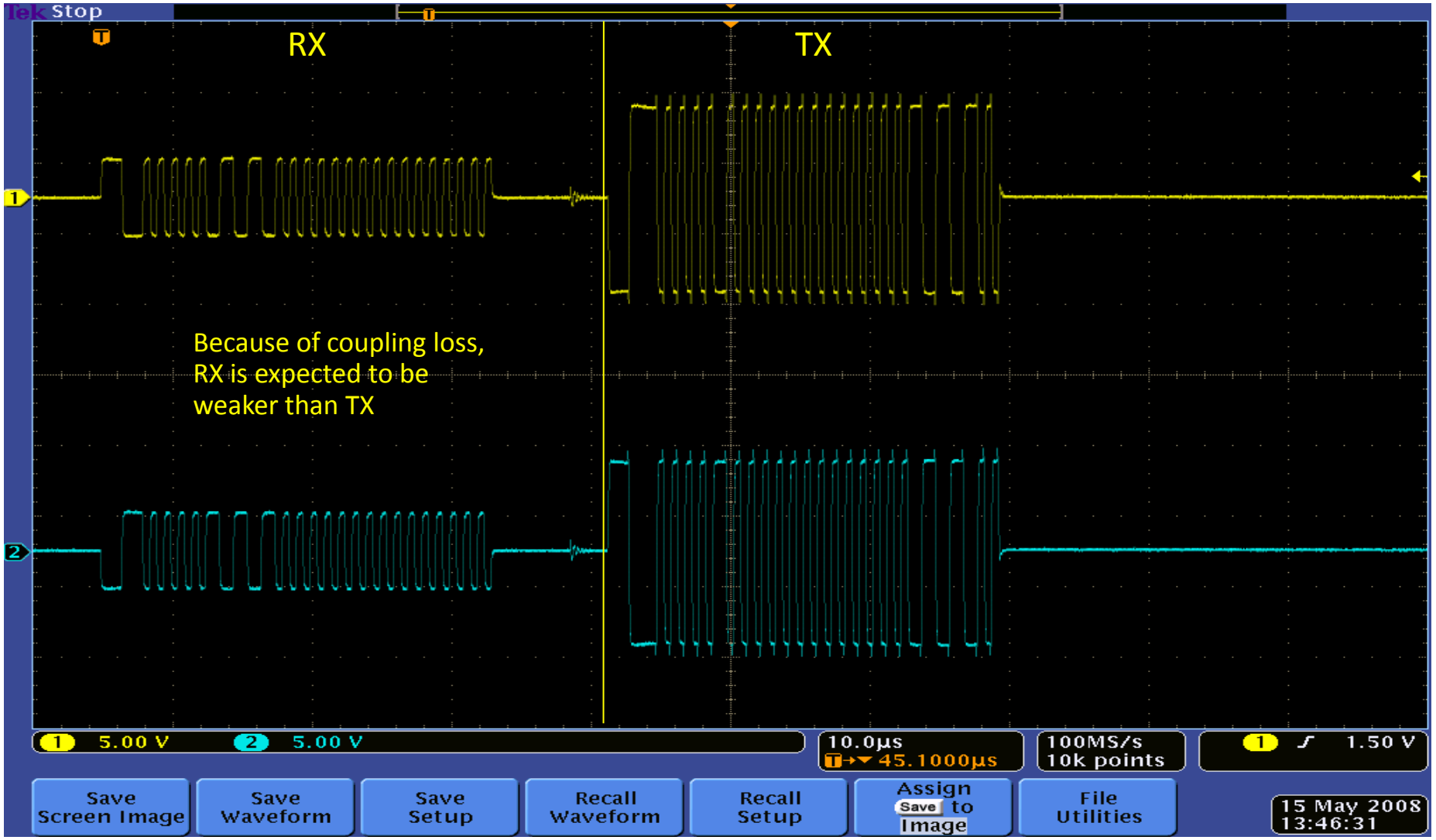
This leads to the possibility of a faulty binary logic signal being "1/2" (an indeterminate state between 0 and 1).

Definition of “1/2”

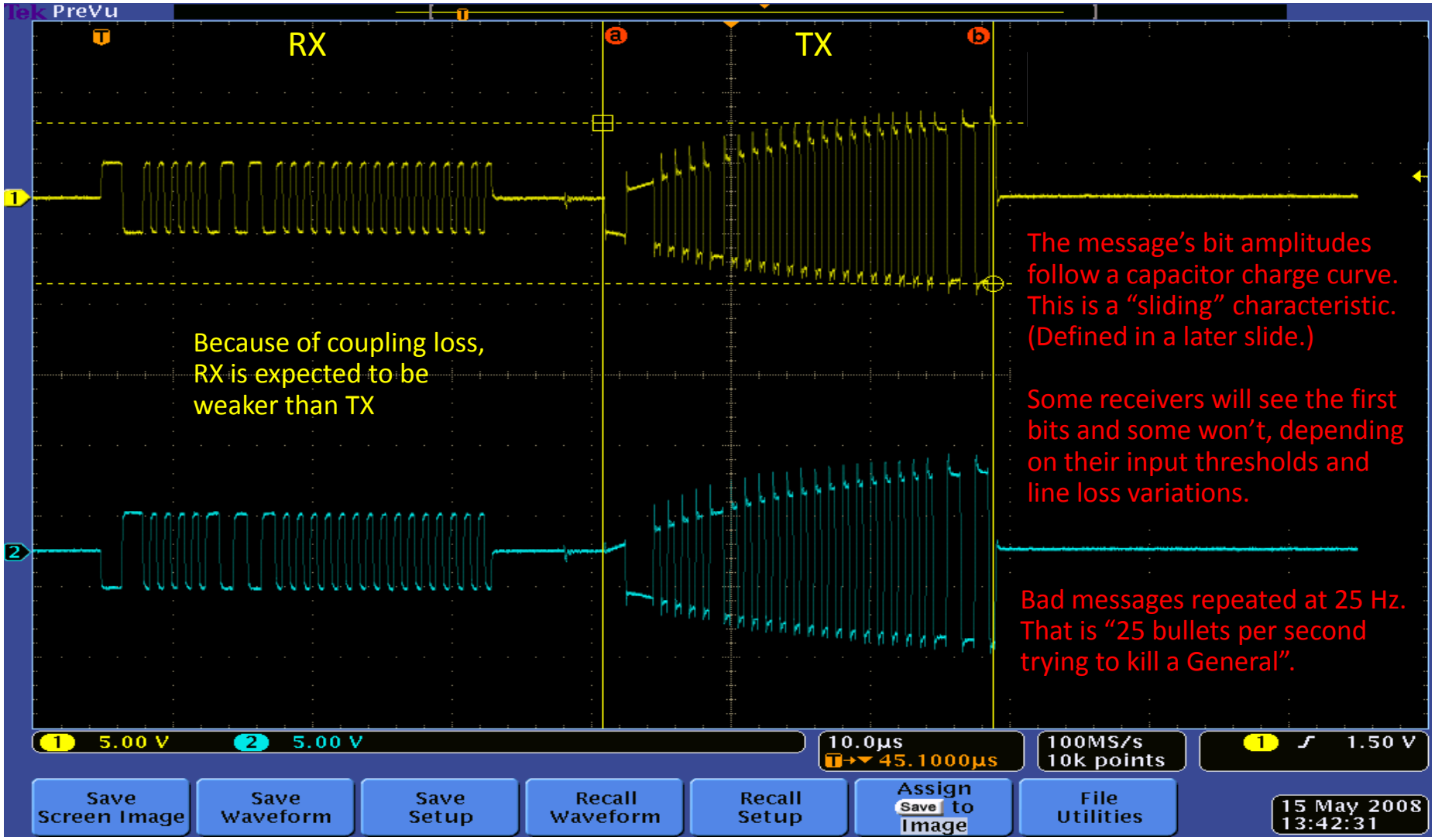


* RX threshold depends on manufacturing tolerance, power supply voltage, ground shift, temperature, fields from nearby signals, accumulated stress, …

Normal Messages (differential traces)

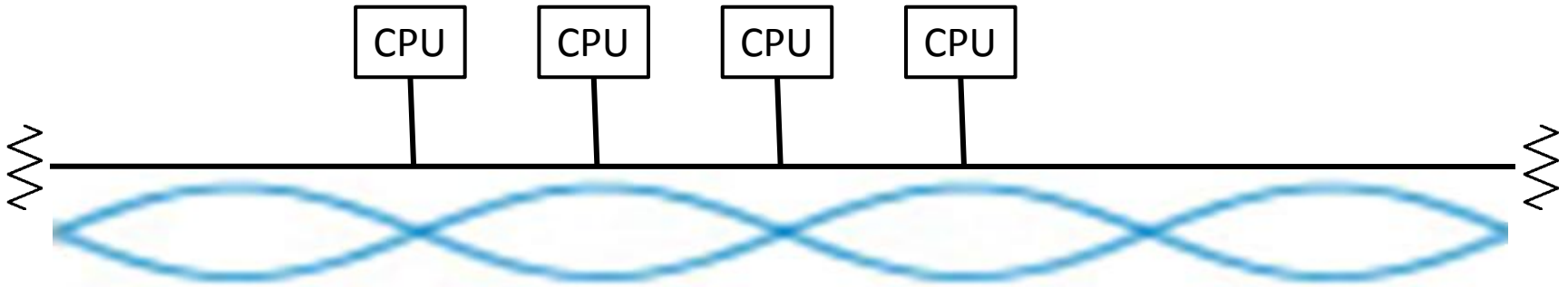


Faulty TX Message on the Right



Shuttle Byzantine History

- 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.
- 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."



Summary of STS−124 Observations

- 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 an accomplice for mass murder
  - 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
- FMEA teams should include:
  - Curmudgeons, skeptics, & "pathological thinkers" (to counterbalance designers, who are optimists)
  - Members of related/neighboring disciplines
  - Physicists (find other Feynmans)

Arthur C. Clark's 1st law of prediction: "When a distinguished but elderly scientist states that something is possible, he is almost certainly right. When he states that something is impossible, he is very probably wrong."

Short Stories

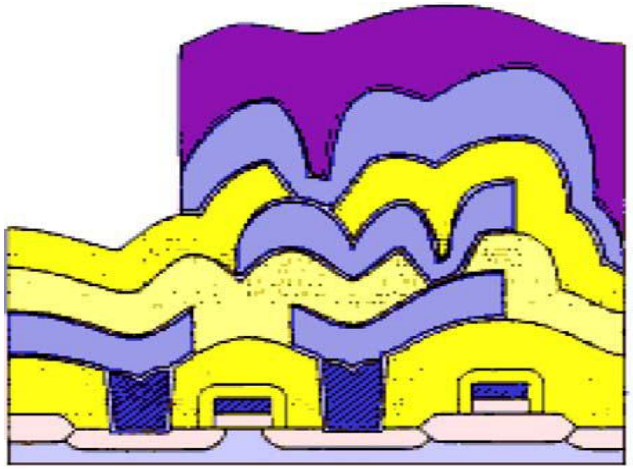
Simplified and with some details changed to protect the guilty. ☺

A Partogenesis Example

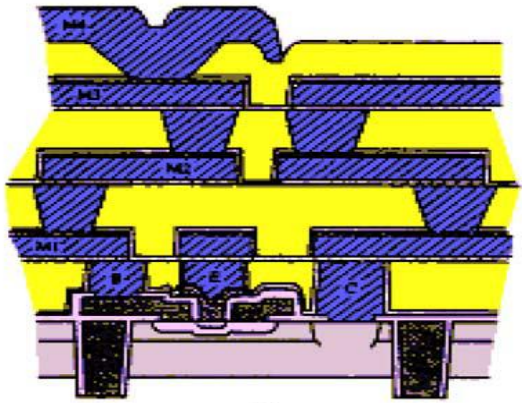
- Integrated circuit (IC) manufacturers often use "planarization" to reduce die surface roughness during processing, sometimes by filling in low spots with metal. Of course, any adjacent signal or power traces must be insulated from this metal.
- If a fault occurs in the insulation, a large block of 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.
- 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.



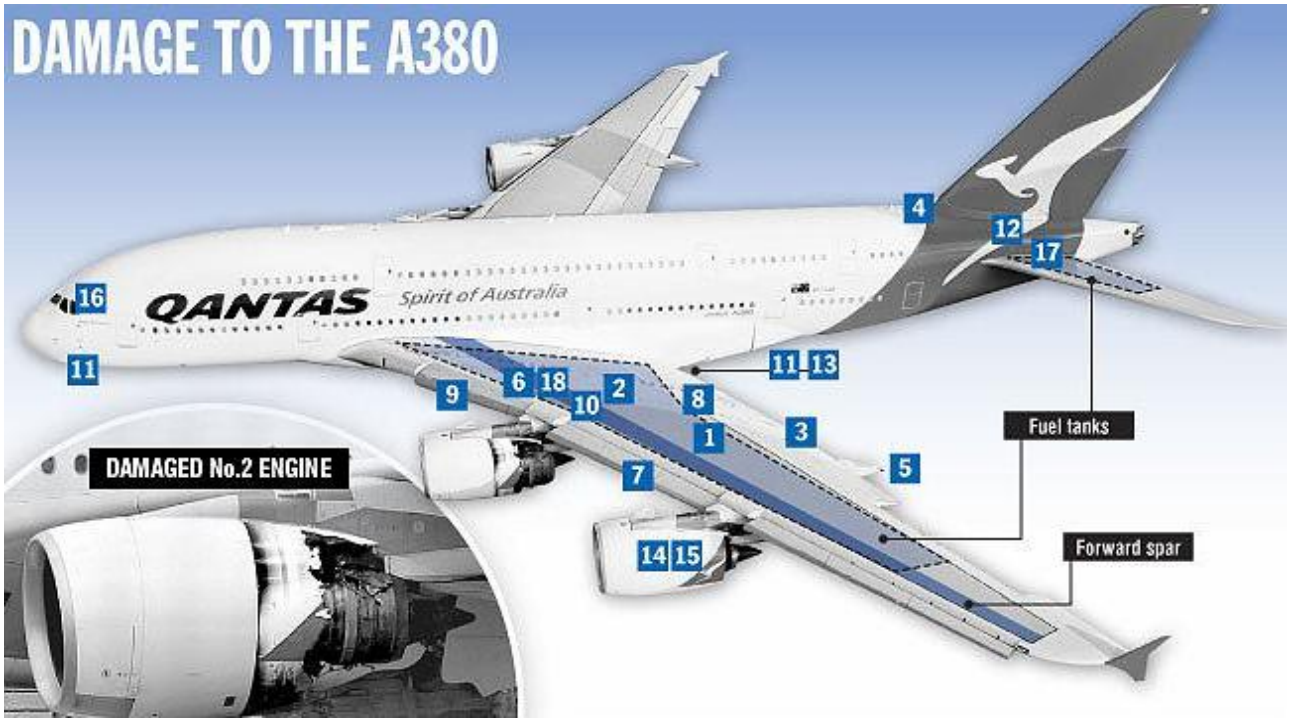
(a)



(b)

Self-Inflicted Shrapnel, I

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



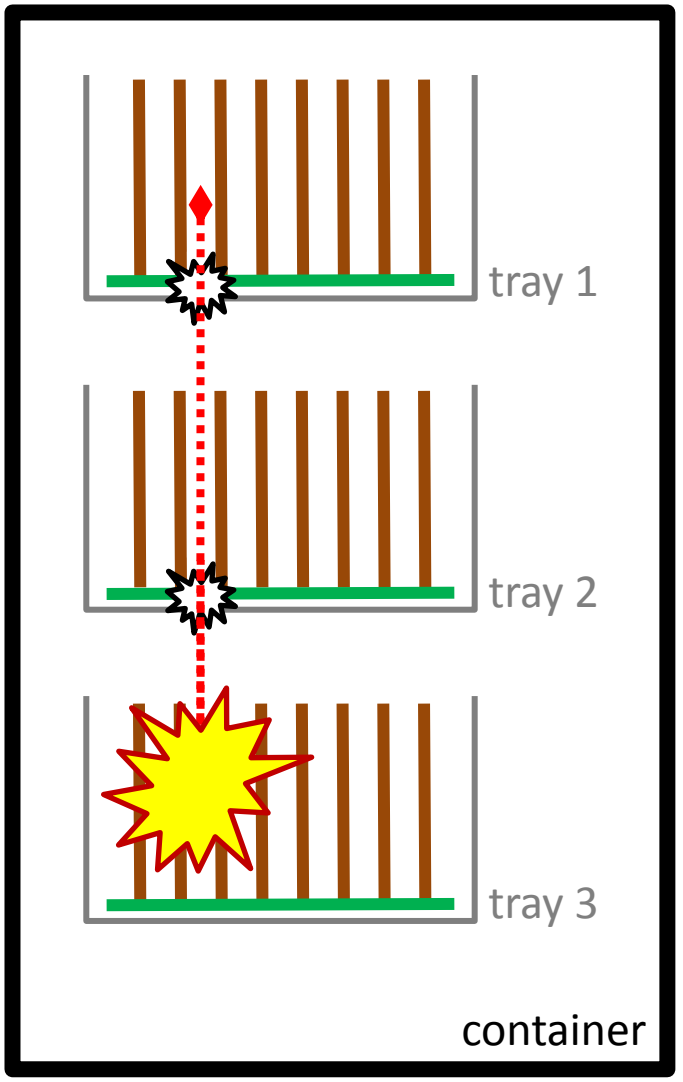
12 Loss of one generator and associated systems [electrical busses 1 and 2 failed]
13 Loss of brake anti-skid system
14 No.1 engine could not be shut down in the usual way after landing because of major damage to systems
15 No.1 engine could not be shut down using the fire switch, which meant fire extinguishers wouldn't work
16 Major fuel imbalance (because of fuel leaks on left side) could not be fixed with cross-feeding
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. Leaking fuel on 900°C 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, there have been over a half-dozen previous similar incidents (Sioux City DC-10 crash is well known). "Those who cannot remember the past are condemned to repeat it."

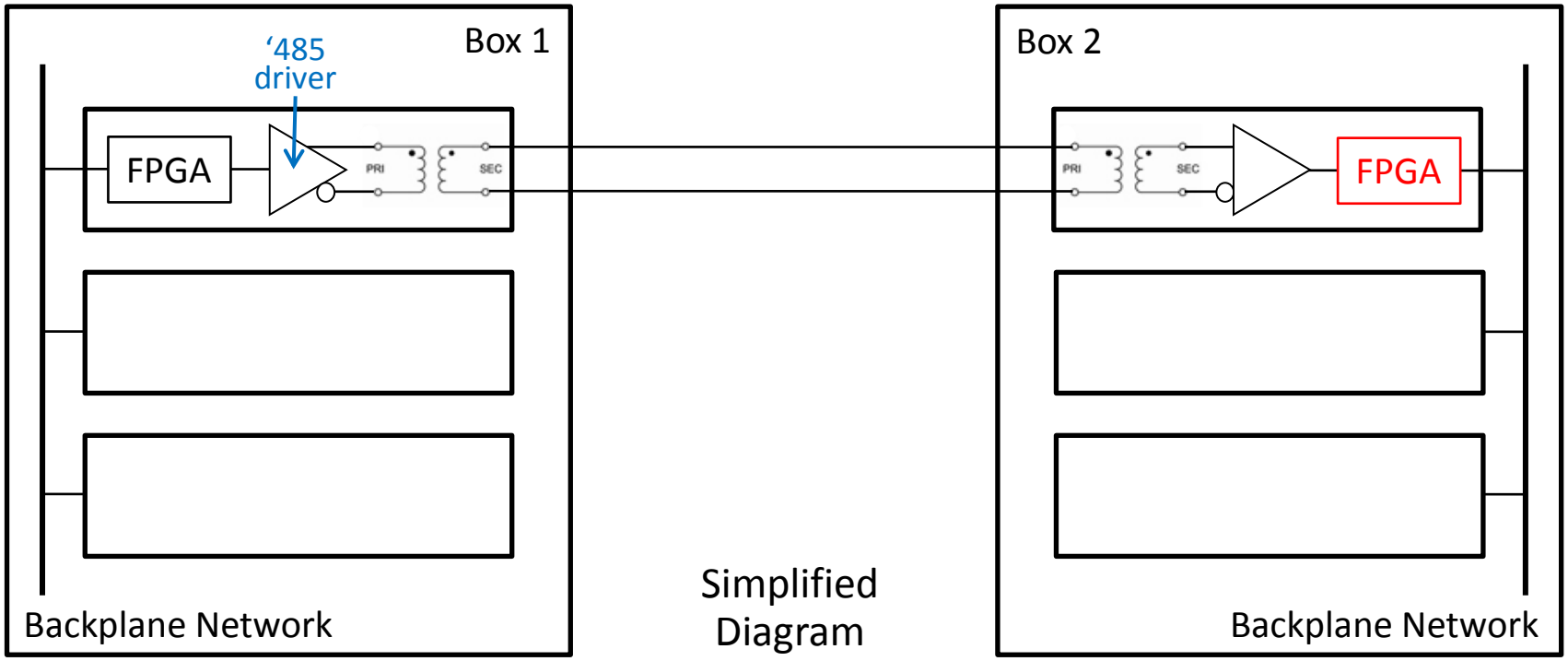
Self-Inflicted Shrapnel, II

- Three trays of a redundant military system were enclosed in a single “bullet-proof” enclosure. Each **aluminum tray** had a **horizontal motherboard** and several **vertical printed-circuit cards**.
- A capacitor in one of the trays exploded, sending “shrapnel” (pieces of the thick plastic that had encased the capacitor) 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.



Not So "Stuck-At"

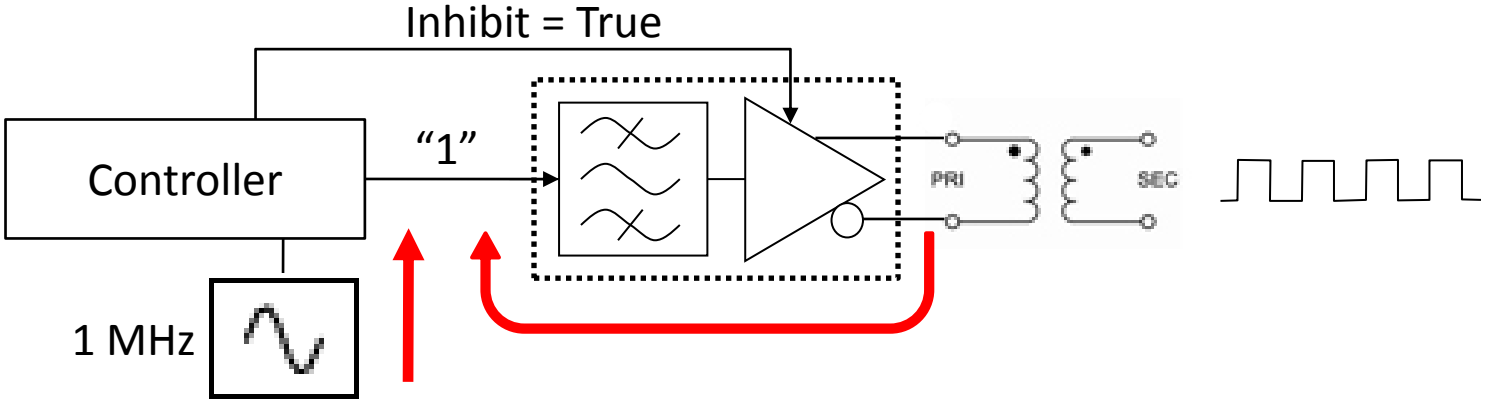
- Our customer naïvely assumed only stuck-at faults (a common assumption) and needed to prevent fault propagation from box to box → the "solution" was to use transformers
- But, due to previous experience and our healthy respect for Murphy, we added an RX fault-tolerance FPGA to block arbitrary noise 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

Not So "Stuck-At", Déjà Vu

- 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

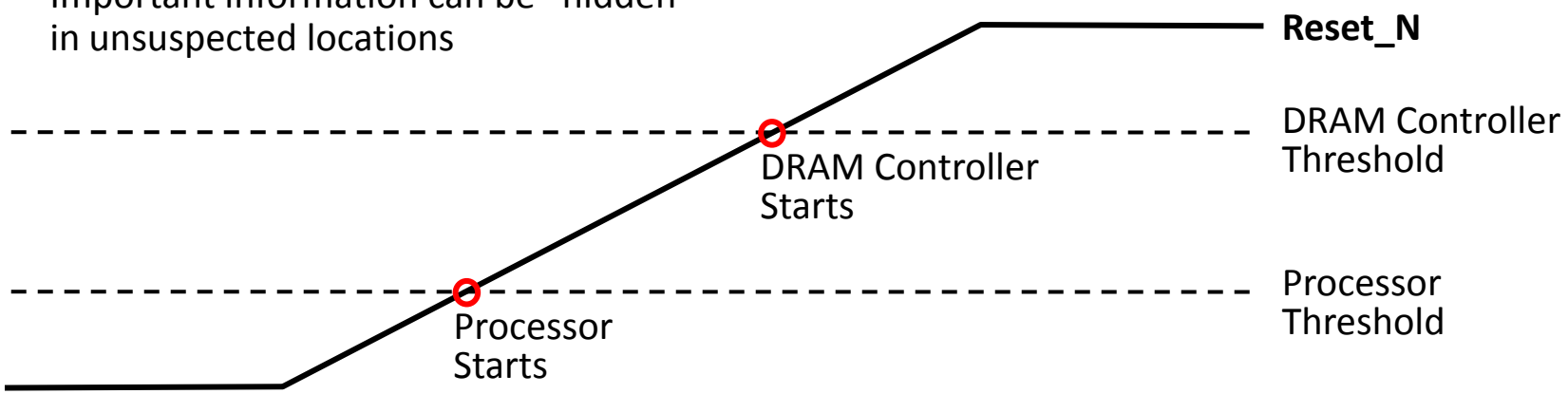


“Evaporating” Software

- When reset at certain temperatures in an environmental test chamber, a military-quality embedded processor had one (and only one) of its subroutines “evaporate”.
- 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 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

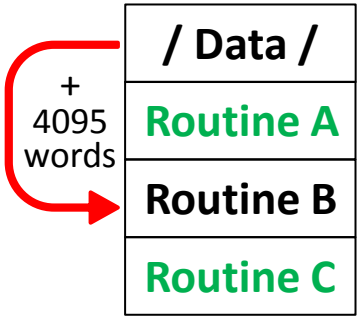
➔ ***Programmers should read the entirety of all documentation***

  - Important information can be “hidden” in unsuspected locations



“Completely” Tested Software

- **Two Fortran routines (A and C) were exhaustively tested (every possible combination of internal state and input), which meets the DO-178 definition of “simple”**
- Routine B was not completely 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.



**If anything "can't" go wrong,
it will go wrong anyway.**

**→ Any statement of "that can't happen"
should be given a great deal of skepticism.**

Honest self-assessment:
An hour ago, how many of the presented failure *symptoms* would you have said, "that can't happen"?
Of course, after explanation, it is obvious that the root-cause faults can happen.

If you have any stories about rare failure modes, please e-mail them to me.