EECE499 – Computers & Nuclear Energy
Department of Electrical and Computer Engineering
Howard University
Dr. Charles Kim
Fall 2013

Website: www.mwftr.com/CNE.html

How computer systems fail

André Duarte Palhares

Topics

- 3.3 COMPUTER HARDWARE FAILURE MODES AND EFFECTS
- 3.4 SOFTWARE FAULTS AND FAILURES
- 3.5 DESIGN FAULTS AND FAILURES

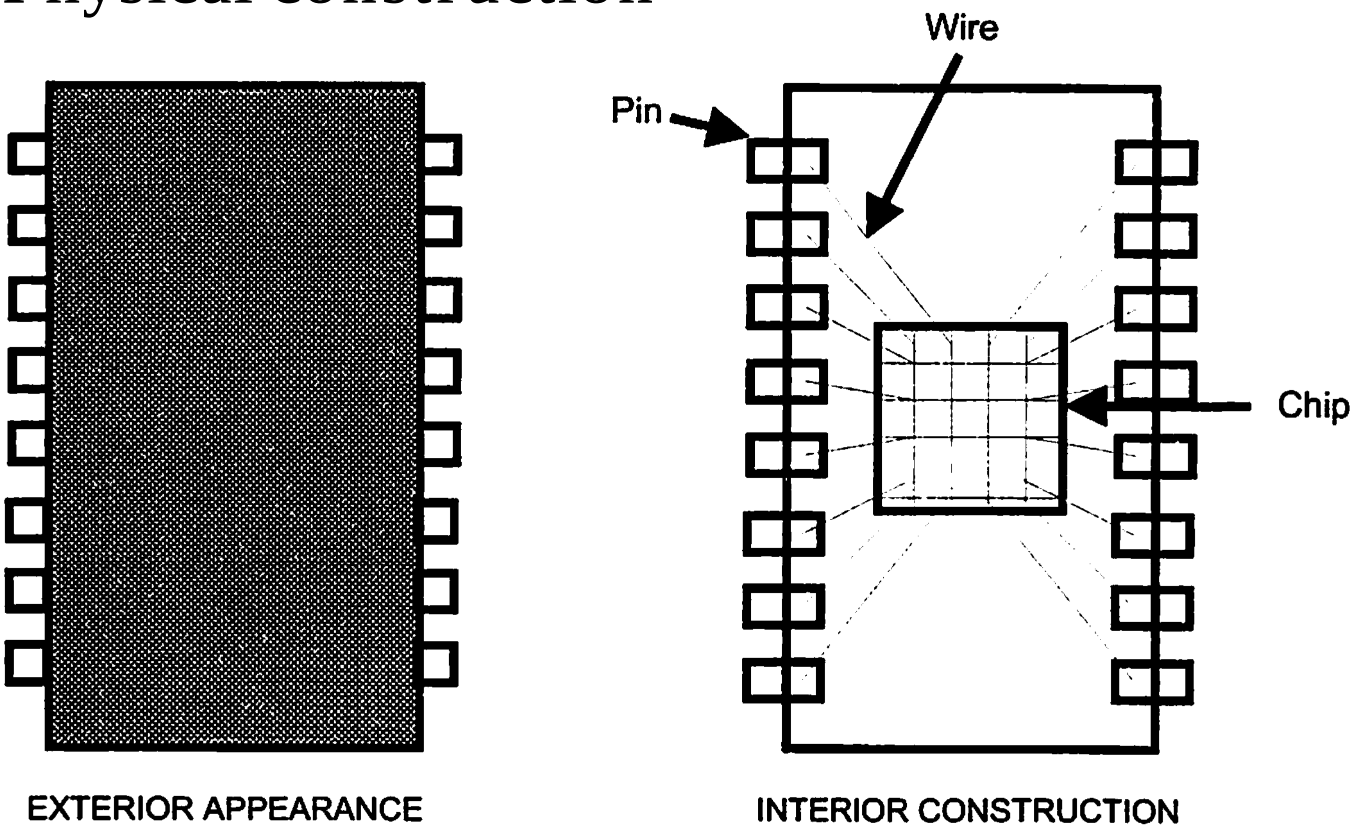
3.3 - Computer hardware failure modes and effects

- To adequately understand the function of a system under failed conditions, it is necessary to look “inside” its components to see how they can fail and how these failures alter their behavior.

- Discussion begins with the computer’s most basic ingredient—the digital integrated circuit.

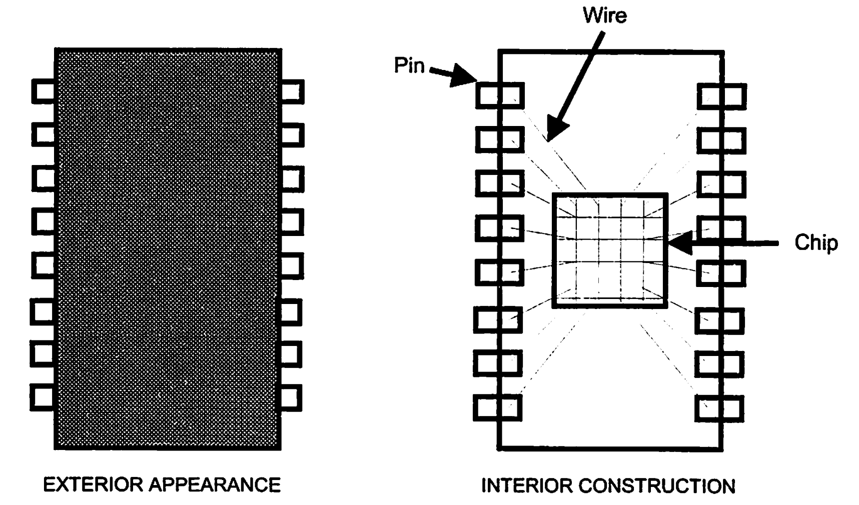
1. The Digital Integrated Circuit

- Physical construction



The Digital Integrated Circuit

- Two types of pins:
  - Signal pins (most of them)
    - Allow the exchange of data along the circuit.
  - Support pins (two or more)
    - Are used to supply power to the IC.



The Digital Integrated Circuit – Failure behavior

- Given correct inputs, provides incorrect outputs, or at the wrong time/with wrong timing.

But HOW does this happen?

The Digital Integrated Circuit – Failure modes

- Basically, 3 ways

1.) Input data is altered between the pins and the chip;
2) Output data is altered between chip and the pins;

3) The chip fails to perform its intended input/output function.

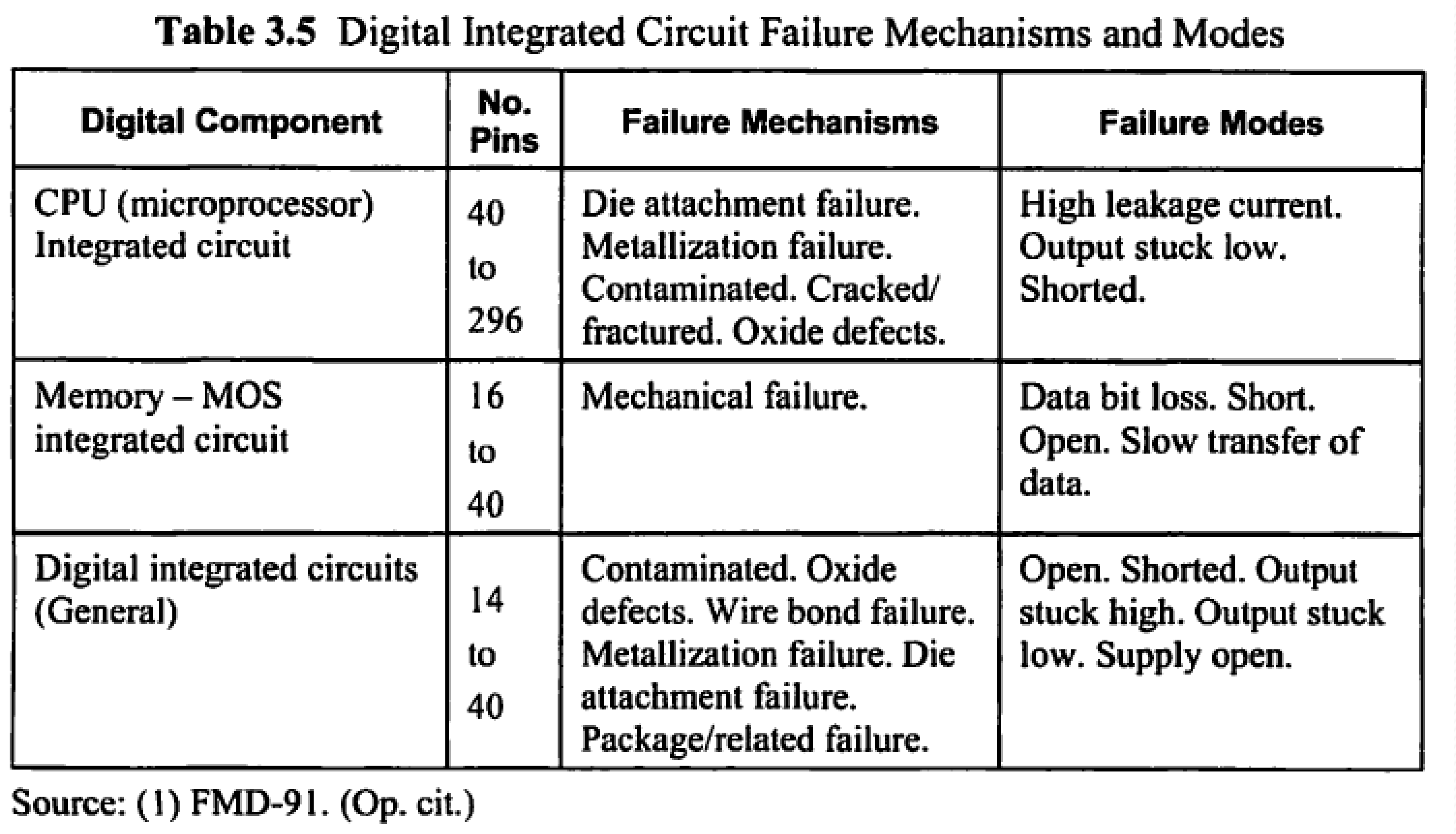


Table 3.5 Digital Integrated Circuit Failure Mechanisms and Modes

Digital Component	No. Pins	Failure Mechanisms	Failure Modes
CPU (microprocessor) Integrated circuit	40 to 296	Die attachment failure. Metallization failure. Contaminated. Cracked/ fractured. Oxide defects.	High leakage current. Output stuck low. Shorted.
Memory – MOS integrated circuit	16 to 40	Mechanical failure.	Data bit loss. Short. Open. Slow transfer of data.
Digital integrated circuits (General)	14 to 40	Contaminated. Oxide defects. Wire bond failure. Metallization failure. Die attachment failure. Package/related failure.	Open. Shorted. Output stuck high. Output stuck low. Supply open.

Source: (1) FMD-91. (Op. cit.)

- Given a set of binary inputs, the integrated circuit can generate virtually any binary output.

What else can fail?

What else can fail
2. Electronic components

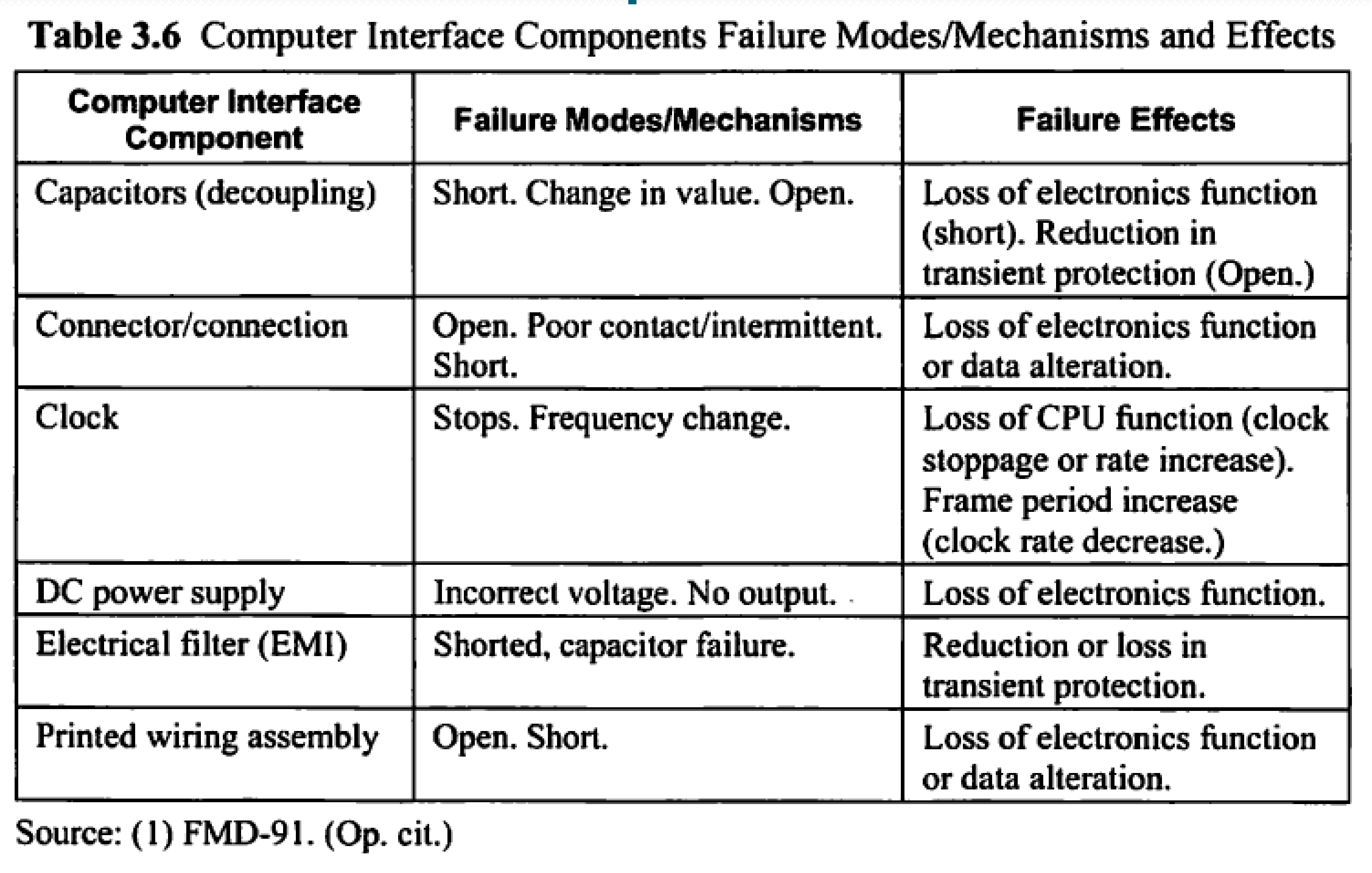


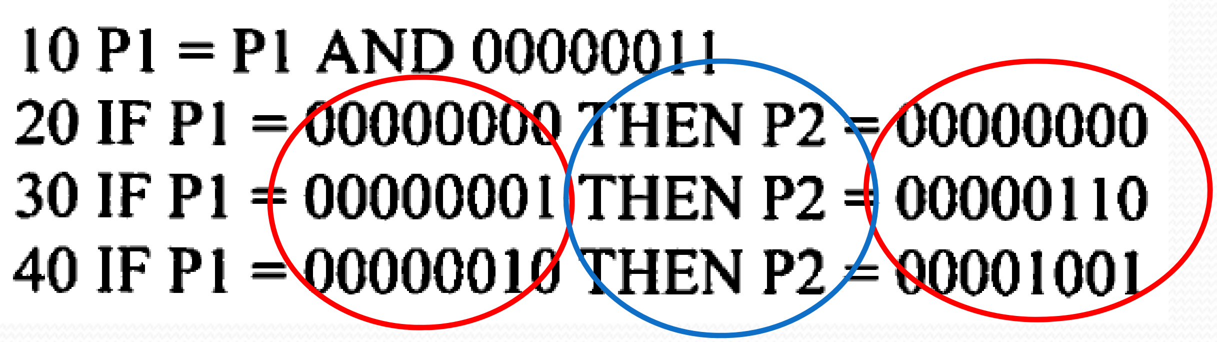
Table 3.6 Computer Interface Components Failure Modes/Mechanisms and Effects

Computer Interface Component	Failure Modes/Mechanisms	Failure Effects
Capacitors (decoupling)	Short. Change in value. Open.	Loss of electronics function (short). Reduction in transient protection (Open.)
Connector/connection	Open. Poor contact/intermittent. Short.	Loss of electronics function or data alteration.
Clock	Stops. Frequency change.	Loss of CPU function (clock stoppage or rate increase). Frame period increase (clock rate decrease.)
DC power supply	Incorrect voltage. No output.	Loss of electronics function.
Electrical filter (EMI)	Shorted, capacitor failure.	Reduction or loss in transient protection.
Printed wiring assembly	Open. Short.	Loss of electronics function or data alteration.

Source: (1) FMD-91. (Op. cit.)

What else can fail
3. Memory and CPU

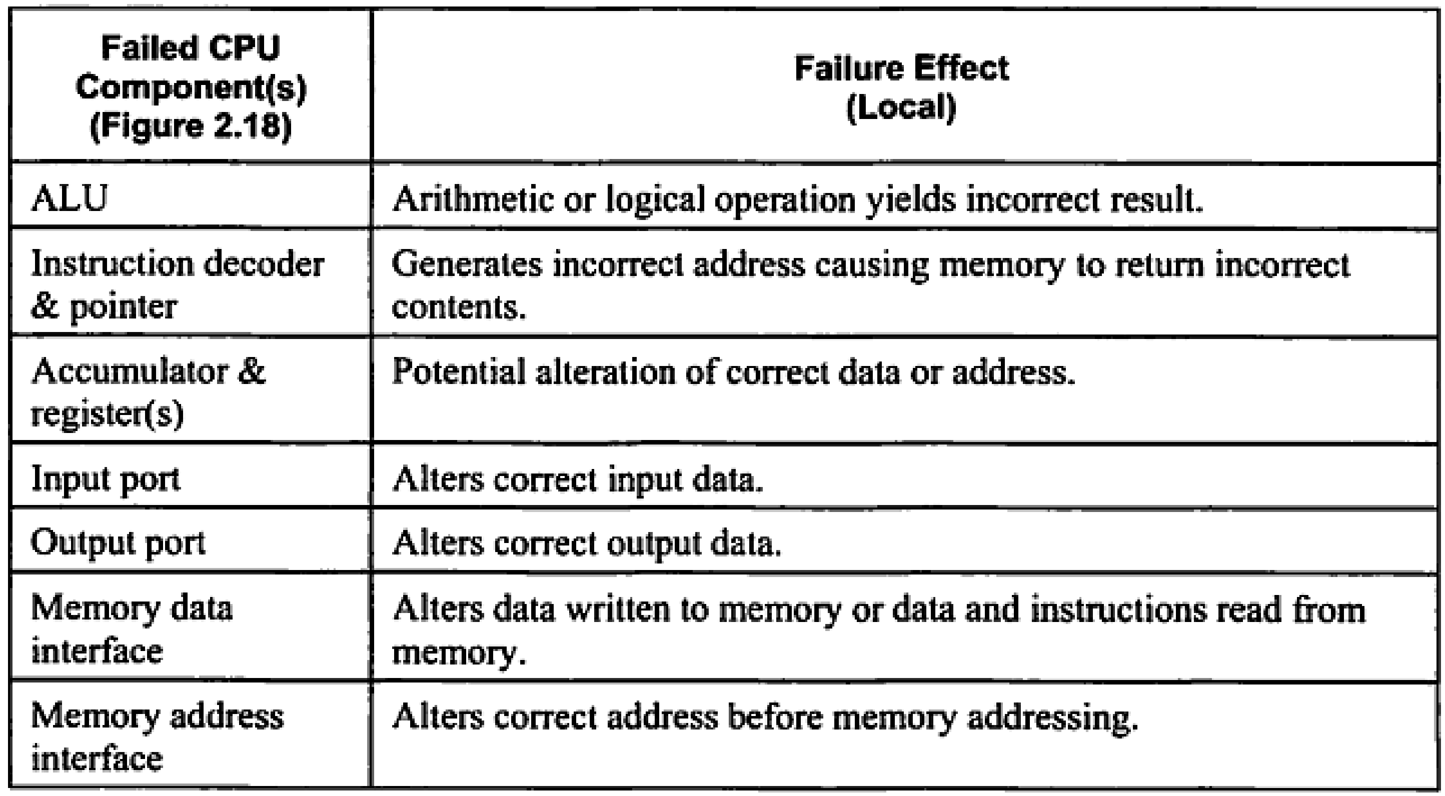
- What can go wrong
- The memory, given an input address, will fail to correctly store or return stored data and/or instructions from/to that address.



```
10 P1 = P1 AND 00000011
20 IF P1 = 00000000 THEN P2 = 00000000
30 IF P1 = 00000001 THEN P2 = 00000110
40 IF P1 = 00000010 THEN P2 = 00001001
```

What else can fail
3. Memory and CPU

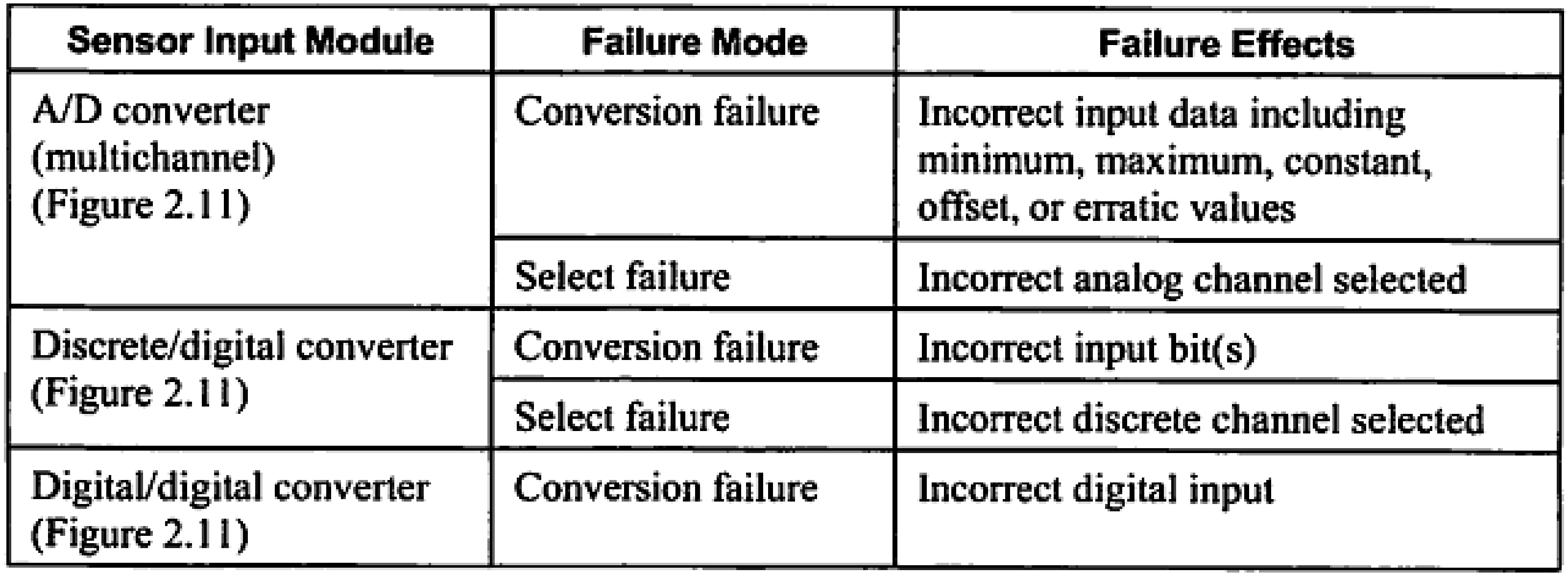
Failures internal to the CPU are not a direct threat—it is the propagation of these failures to the outside that produces safety concern



Failed CPU Component(s) (Figure 2.18)	Failure Effect (Local)
ALU	Arithmetic or logical operation yields incorrect result.
Instruction decoder & pointer	Generates incorrect address causing memory to return incorrect contents.
Accumulator & register(s)	Potential alteration of correct data or address.
Input port	Alters correct input data.
Output port	Alters correct output data.
Memory data interface	Alters data written to memory or data and instructions read from memory.
Memory address interface	Alters correct address before memory addressing.

What else can fail
4. Input and Output modules

The multi-input A/D converter employs a combination of analog and digital circuits. Thus, its failure modes will include those found in both types of circuits.



Sensor Input Module	Failure Mode	Failure Effects
A/D converter (multichannel) (Figure 2.11)	Conversion failure	Incorrect input data including minimum, maximum, constant, offset, or erratic values
	Select failure	Incorrect analog channel selected
Discrete/digital converter (Figure 2.11)	Conversion failure	Incorrect input bit(s)
	Select failure	Incorrect discrete channel selected
Digital/digital converter (Figure 2.11)	Conversion failure	Incorrect digital input

What else can fail

5. Input and Output devices

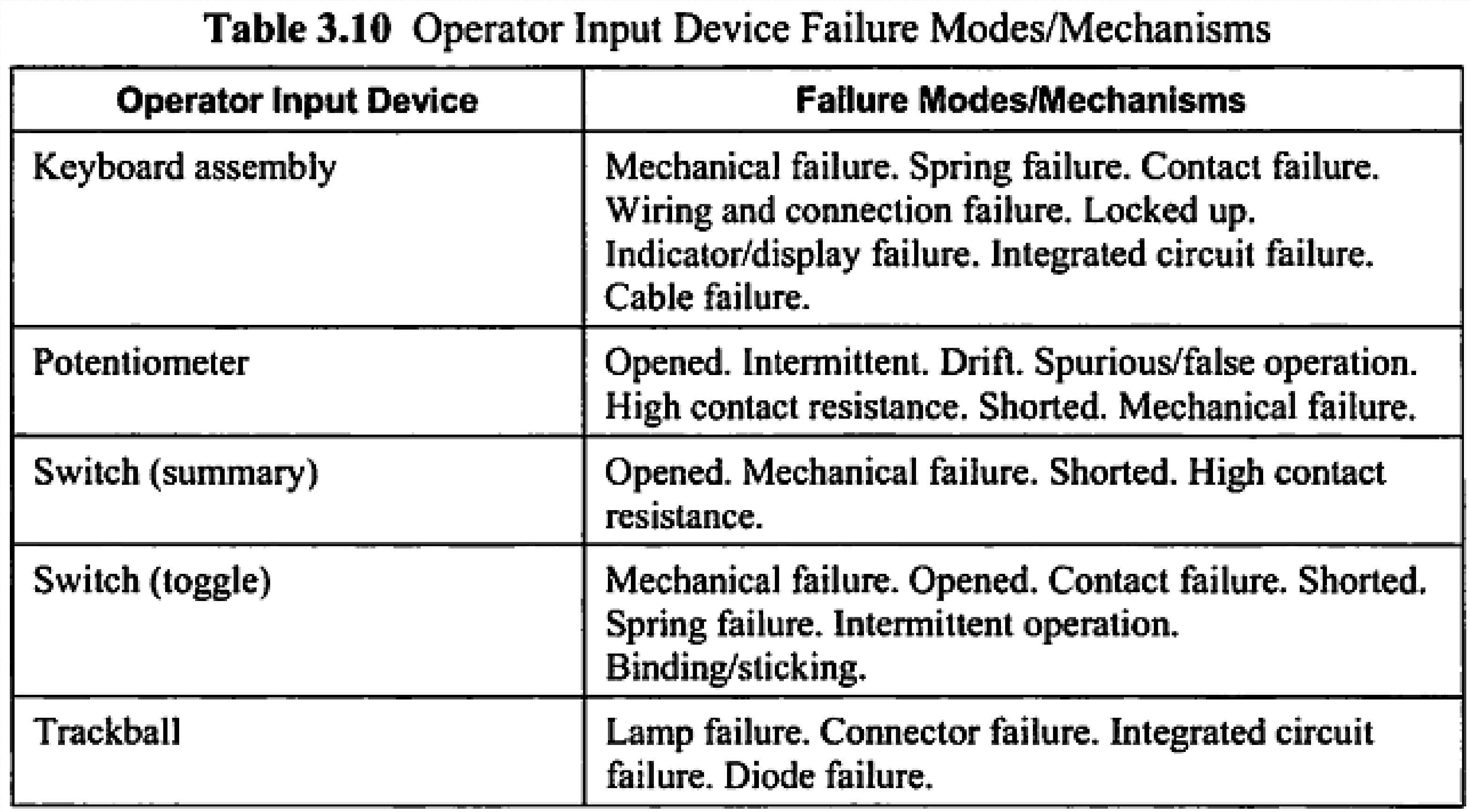


Table 3.10 Operator Input Device Failure Modes/Mechanisms

Operator Input Device	Failure Modes/Mechanisms
Keyboard assembly	Mechanical failure. Spring failure. Contact failure. Wiring and connection failure. Locked up. Indicator/display failure. Integrated circuit failure. Cable failure.
Potentiometer	Opened. Intermittent. Drift. Spurious/false operation. High contact resistance. Shorted. Mechanical failure.
Switch (summary)	Opened. Mechanical failure. Shorted. High contact resistance.
Switch (toggle)	Mechanical failure. Opened. Contact failure. Shorted. Spring failure. Intermittent operation. Binding/sticking.
Trackball	Lamp failure. Connector failure. Integrated circuit failure. Diode failure.

What else can fail
5. Input and Output devices

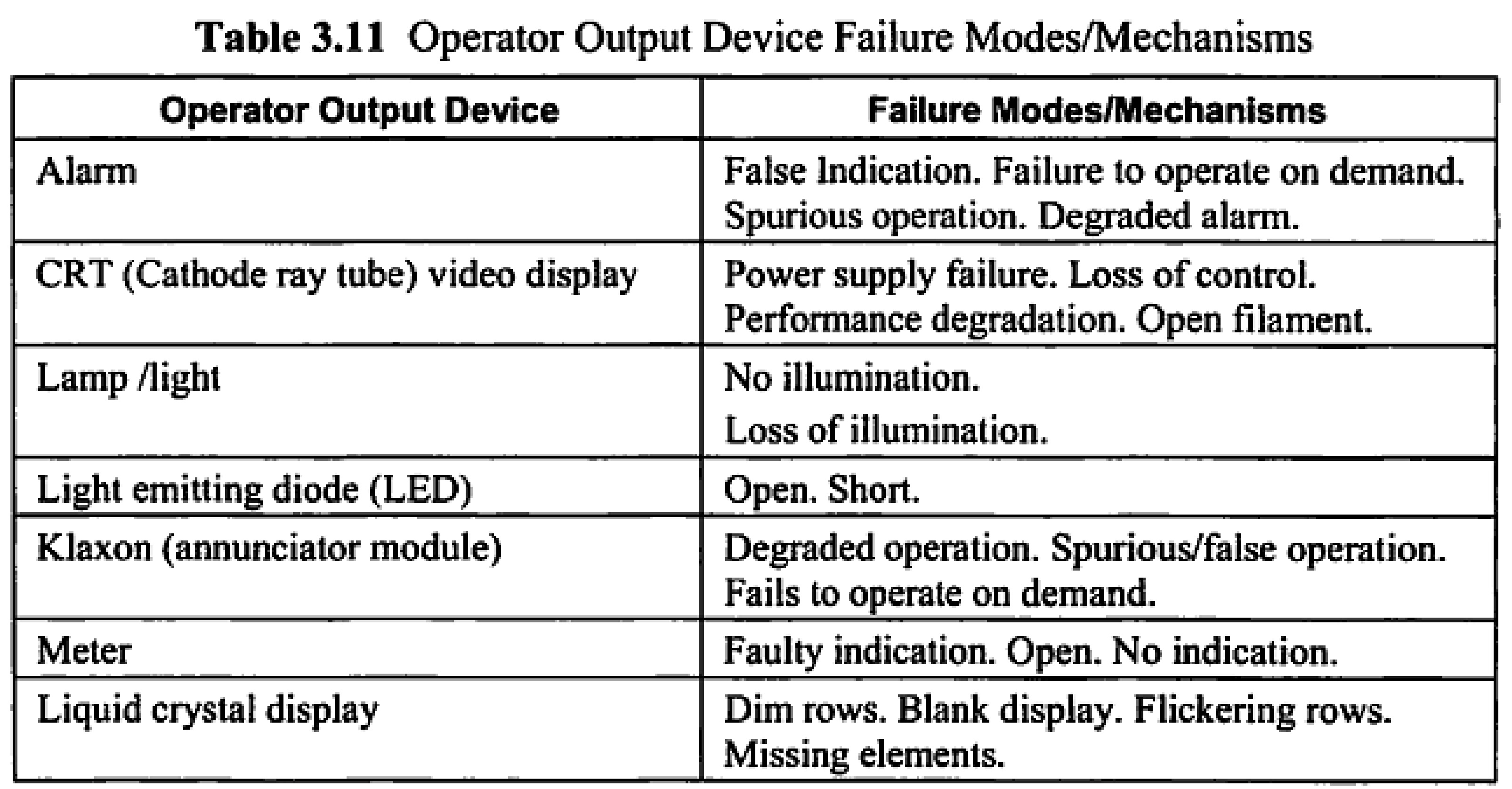


Table 3.11 Operator Output Device Failure Modes/Mechanisms

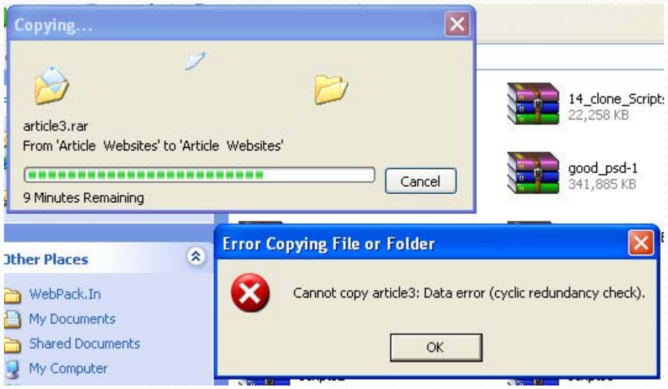
Operator Output Device	Failure Modes/Mechanisms
Alarm	False Indication. Failure to operate on demand. Spurious operation. Degraded alarm.
CRT (Cathode ray tube) video display	Power supply failure. Loss of control. Performance degradation. Open filament.
Lamp /light	No illumination. Loss of illumination.
Light emitting diode (LED)	Open. Short.
Klaxon (annunciator module)	Degraded operation. Spurious/false operation. Fails to operate on demand.
Meter	Faulty indication. Open. No indication.
Liquid crystal display	Dim rows. Blank display. Flickering rows. Missing elements.

What else can fail

6. Communication Modules

Employ both analog and digital electronics and are coupled to communication lines or busses using a variety of devices such as optoisolators, electrical transformers, and optical transceivers (fiber optics). Failure modes for these kinds of devices have already been discussed. Accordingly, a communication module could experience failure by:

- Failing to transmit and/or receive data
- Transmitting incorrect data
- Distorting received data



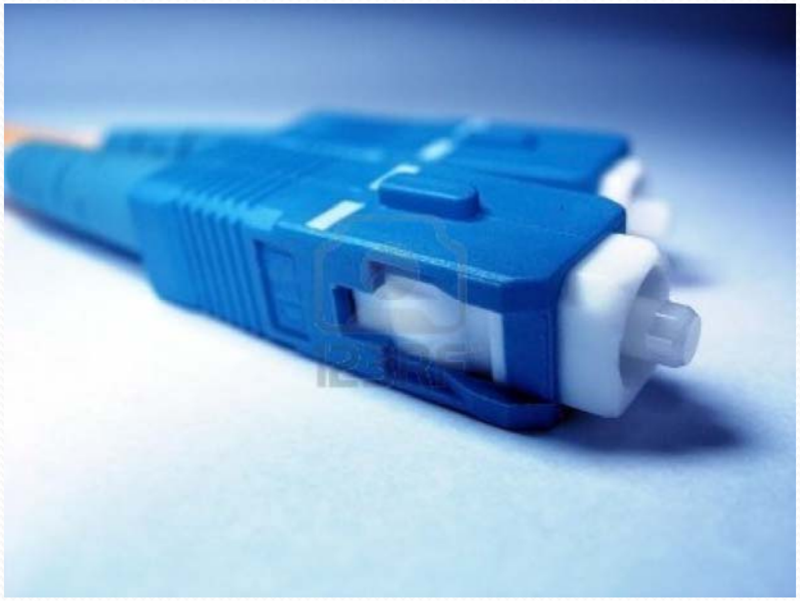
Copying...

article3.rar
From 'Article Websites' to 'Article Websites'

9 Minutes Remaining

Error Copying File or Folder

Cannot copy article3: Data error (cyclic redundancy check).



What else can fail
7. Peripheral units

- Includes disk drives, tape drives, printers, event recorders, etc.
- Usually not considered safety-critical items, but faults in these units could produce failures in the real-time operational system.

3.4 Software faults and failures

Computer software failure modes and effects

- Programs can and do fail and, like hardware failures, can produce incorrect operator and effector outputs that can lead to mishaps.

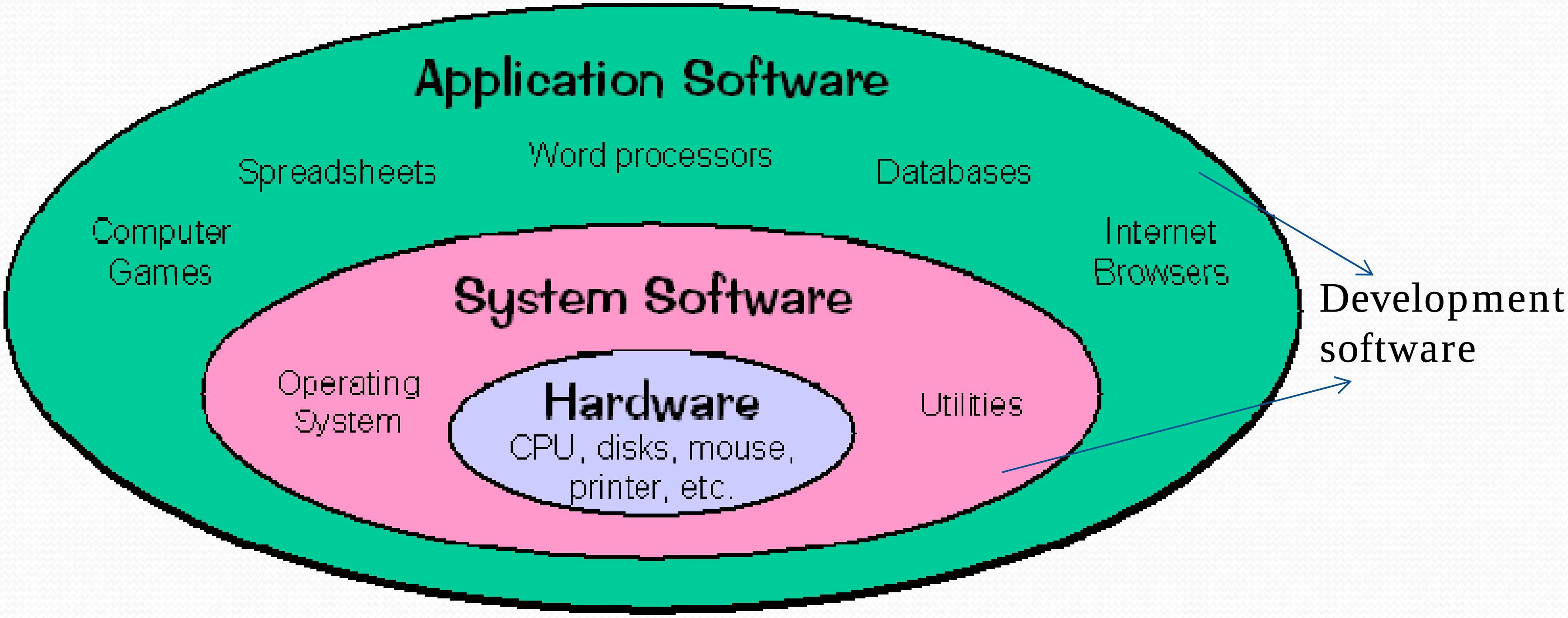
Fault-free software

- For example, the home microwave oven, VCR, and modern automotive systems rarely exhibit software “bugs.” If one looks at software in these applications, it is found that it
- Usually employs discrete variables only
- Involves a finite discrete input/output function
- Has no real-time constraints
- Can be exhaustively tested

Not so complex appliances!

Software faults, failures and effects

- 3 types of software



Development software

Software faults, failures and effects

Application Software Faults

- There are three basic categories of software faults:
  - Misinterpreted requirements
  - Incorrect software design or implementation
  - Clerical error

How the customer explained it

Software faults, failures and effects
Misinterpreted requirements

- Misinterpreted requirements example:
- Imagine a written requirement that states:

- *"With the PURGE switch "on," the nitrogen valves should be open. With the RUN switch "on," the oxygen valve and the hydrogen valves should be open."*

Software faults, failures and effects
Misinterpreted requirements

- Now, suppose that statement is misunderstood to mean: “With the RUN switch “on,” the nitrogen valves should be open. With the PURGE switch “on”, the oxygen valve and the hydrogen valves should be open.”

Pop-up quiz: What will happen if the PURGE switch is pressed in a situation like this?

Software faults, failures and effects
Misinterpreted requirements

- If this misinterpretation is implemented, the PURGE switch, intended to make the system safe, becomes a trigger for an explosion.

The documents were correctly written, but the programmer interpreted it erroneously. This looks obvious in a small example, but software requirements documentation may have more than 100 pages of text.

"We don't need to plan – we're Agile!"

Software faults, failures and effects

Incorrect Software Design or Implementation

Requirements correctly understood, but software has coding errors, where the programmer may be…

- Mistaking the correct logic operator (AND, OR, NOT, …)
- Using the wrong variable name
- Employing the wrong function
- Mistaking a loop index range
- Failing to initialize variables
- Calling the wrong subroutine
- Falling into an infinite loop, etc

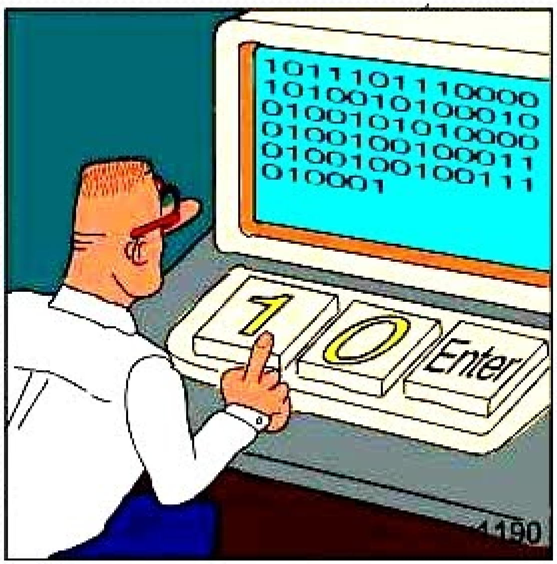
Software faults, failures and effects Clerical errors

- Typographic errors...

E.g.: Writing 0.9 instead of -0.9

3.5 Design faults and failures

- How do they originate?
  - Personnel Error
  - Limited Engineering Knowledge
  - Added Complexity – for safety-critical systems



REAL Programmers code in BINARY.

- Potential consequences