Dependability

- Qualitative term for the ability of the system to perform properly
- encapsulates reliability, availability, safety, maintainability, performability, testability
Reliability - Unreliability

- $R(t)$ is the probability that the system performs as specified without interruption over the entire interval $[0,t]$.
- $R(t)$ is conditioned on the system being operational at time $t=0$.
- Unreliability $F(t)$ is the probability that the system fails at any time in the interval $[0,t]$.
- $F(t) = 1 - R(t)$
Reliability - Unreliability

- time $t$ can be very long, e.g. years in case of space applications
- Notation $0.9_i = .99999999$  i  9s
- This notation is often used for reliability

E.g.

$Q(t) = 10^{-x}$

$R(t) = 0.9^x$

$= (1 - 10^{-x})$
Safety $S(t)$

- $S(t)$ is the probability that the system does not fail in the interval $[0,t]$ in such a manner as to cause unacceptable damage or other catastrophic effects.

- Safety is a measure of the fail-safe capability of the system
  - system can be unreliable, yet safe
  - bias towards safe failure
  - e.g. duplex system (detector)
  - e.g. babbling driver (not safe)
**Availability \( A(t) \)**

- \( A(t) \) is the probability that the system is up and running correctly at time \( t \)

- This is different from reliability.
  - Reliability considers the interval \([0,t]\)
  - Availability takes an instance of time

- examples: transaction processing systems, e.g. reservation systems
Performability

- $P(L,t)$ is the probability that the system performance will be at or above some level $L$ at time $t$
- Measure of the likelihood that some subset of the function is performed correctly
- This differs from reliability, which dictates that all functions are performed correctly
Graceful Degradation

- The ability of system to automatically decrease its level of performance to compensate for hardware failure and software errors.
Maintainability

- $M(t)$ is the probability that a failed system will be restored within a specified period of time $t$.
- Restoration process
  - locating problem, e.g. via diagnostics
  - physically repairing system
  - bringing system back to its operational condition
**Fault - Error - Failure**

- **Fault** = physical defect or flow occurring in some component (hardware or software)

- **Error** = incorrect behavior caused by a fault
  - manifestation of fault

- **Failure** = inability of the system to perform its specified service
Fault - Error - Failure

Fault $\rightarrow$ Error $\rightarrow$ Failure

- physical universe
- informational universe
- external universe
- bit “stuck-at” $\rightarrow$ incorrect data to ALU $\rightarrow$ system crash, incorrect bank balance

Note: presents of fault does not ensure that error will occur, e.g. memory stuck-at-0
Characteristics of faults

- **Cause**
  - specification errors
    - very dangerous
    - generic fault
  - implementation
    - very hard to formally verify
  - random component faults
    - random, not manufacturing defects
  - external disturbance
    - noise, EMP, radiation
    - much like random component
Characteristics of faults

- **Origin**
  - software or hardware
  - don’t care, except:
    - hardware can be analog
    - indeterminate voltage level
Characteristics of faults

◆ Duration
  - permanent fault
    » once component fails, it never works correctly again
    » easiest to diagnose
  - transient fault
    » 1 time only
    » 10 times as likely as permanent fault
  - intermittent fault
    » re-occurring
    » may appear to be transient (if long period)
    » hard and expensive to detect
Avoidance - Masking - Tolerance

from Johnson 1989, Fig 2.12

Specification Mistakes

Implementation Mistakes

External Disturbances

Component Defects

Software Faults

Hardware Faults

Fault Avoidance

Fault Masking

Fault Tolerance

Errors

System Failure