HW2: Dependable Systems and Networks

1. (10 points) Some systems are designed for reliability whereas others are designed for availability. Explain the difference between reliability and availability, and give an example of an application requiring high availability and one requiring high reliability.

Ans:

Reliability: the conditional probability that a system performs correctly throughout an interval of time [0, t], given that it was performing correctly at time 0.
Availability: the probability that a system is operating correctly at the instant of time t.
The difference between reliability and availability is that reliability depends on an interval of time, whereas availability is taken at an instant of time.

Application requiring high availability: Banking systems
Application requiring high reliability: Spacecraft

2. (10 points) For the following systems (A and B), identify which attribute (reliability, availability etc.) is considered least important. Justify your answers.
A. An aircraft system has three computers voting on the results of every operation performed by the auto-pilot. If the auto-pilot fails, a warning alarm goes off in the cockpit to alert the pilot, who can then take over the manual controls of the aircraft and guide it to safety. However, the pilot does not interfere as long as the autopilot does not raise the alarm.
B. An online trading website allows its customers to place bids on various items, and to track their bidding online. While it is acceptable for a user to not be able to place bids if the traffic is too high, it is not acceptable for a user who has placed bids to not track their bid's status and modify the bid. Also, as far as possible, the website should not display an incorrect value of the item's current bid, as this can cause users to over/under-bid for it.

In each of the following descriptions (C and D), identify the fault, error and failure.
C. A program contains a rare race condition that is only triggered when the OS schedules threads in a certain order. Once triggered however, the race condition corrupts a value in the program, which in turn is used to make a branching decision. If the branching decision is incorrect, the program will go into an infinite loop and hang, thus failing to produce any output.
D. A radar system uses an array of processors to track its target in real-time. A soft error in a processor can lead to the processor computing an incorrect value for the target's location. However, the system can compensate for this effect by redundantly allocating the tasks to processors and comparing the results. But this compensation

entails a performance overhead, which in some cases, can cause the system to miss the tasks' deadlines and lose the target.

Ans:

A. Reliability is considered least important because the pilot can take over the manual controls of the aircraft if the autopilot fails. On the other hand, high availability is required because the aircraft system is expected to operate without interruptions. It is likely to cause the destruction of the aircraft if the autopilot does not raise the alarm.

B. Availability is considered least important. Short interruptions can be tolerated because it is acceptable for a user not to be able to place bids if the traffic is too high. On the other hand, high reliability is required. The website is expected to display the correct value of the item's current bid as far as possible (without interruptions).

C.
Fault: the incorrect branching decision caused by the race condition
Error: the program goes into an infinite loop and hang
Failure: the program fails to produce any output

D.
Fault: a soft error in a processor
Error: the processor computes an incorrect value for the target's location and triggers the compensation process
Failure: the system misses the tasks' deadlines and loses the target caused by performance overhead.

3. (10 points) A telephone system has less than 3 min per year downtime. What is its steady-state availability?

Ans:

Steady-state availability = proportion of total operating time system is operational.
Since the downtime is less than 3 mins/year,

The steady-state availability $A_\infty > \frac{60\times24\times365-3}{60\times24\times365} = 0.999994292237.....$

4. (10 points) A copy machines manufacturer estimates that the reliability of the machines he produces is 73% during the first 3 years of operation.

(a) How many copy machines will need a repair during the first year of operation?
(b) What is the MTTF of the copy machines?
(c) The manufactures guarantees MTTR = 2 days. What is the MTBF of the copy machines?

(d) Suppose that two copy machines work in parallel and the failures are independent. What is the probability of failure during the first year of operation?

Ans:

(a) $R(t) = e^{-\lambda t} \Rightarrow 0.73 = e^{-3\lambda} \Rightarrow \lambda = \frac{\ln(0.73)}{-3} = 0.1049$
About 10.46% copy machines will need a repair during the first year of operation.

(b) MTTF $= \frac{1}{\lambda} = \frac{-3}{\ln(0.73)} = 9.53256$ (years)

(c) MTBF = MTTF + MTTR = 9.5325 + 2/365 = 9.53804 (years)

(d) $\lambda^2 = 0.1049^2 = 0.011$

5. (10 points) Devise an original example (different from the lecture examples) to illustrate the difference between faults, errors, and failures. As you illustrate these concepts, relate them to the three-universe model.

Ans:

In electric power systems, overloading of equipment, insulation failure due to lighting surges, or mechanical damage by the public are faults (in the physical universe). These faults cause short circuits in currents, which is an error (in the informational universe) and can lead to fire and explosion in equipment such as transformers and circuit breakers. These are failures (in the external universe)

6. (10 points) Why redundancy techniques used in hardware system cannot be used for software fault tolerance. If you are employed as a software quality engineer, what techniques you will prefer?

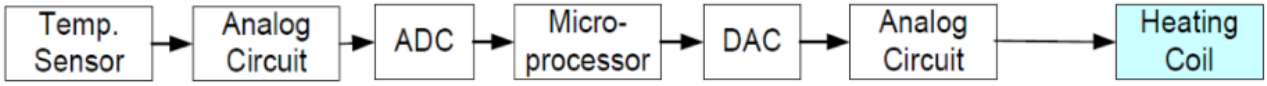
Ans:

First, modules in software tend to have highly correlated failures, so we cannot simplify the reliability evaluation by assuming that failures of components are independent events as we do in the case of hardware. Second, different environments might result in different reliability values in software, so reliability might not be a measure of the dependability of software. Furthermore, the hardware fault tolerance techniques do not offer sufficient protection against design faults, which are dominant in software. Overall, software is considerably more complex than hardware.
If I am a software quality engineer, I will prefer N-Version Programming (NVP) or Recovery Blocks (RB).

7. (20 points) The company that you work for is designing an industrial controller that maintains the temperature of a fluid during a chemical reaction. The non-redundant controller (figure below) contains: (1) temperature sensor; (2) analog circuitry to

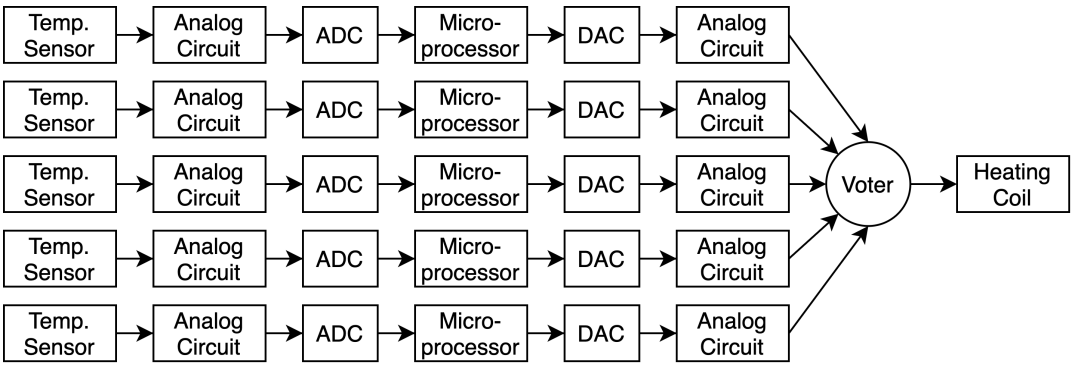
process the temperature sensor's output signal; (3) analog-to-digital converter (ADC); (4) microprocessor (including hardware and software); (5) digital-to-analog converter (DAC); (6) analog circuitry to process the output of the DAC; and (7) heating coil to control the temperature. You have been asked to develop at least two approaches for making the controller tolerant of any two faulty components. The term "component" means one of the blocks of functionality listed above, excluding the heating coil.

(a) Show block diagrams of your two approaches and compare them qualitatively. Note that your designs should be able to handle faults of any two components, including any two same components (e.g., 2 ADCs) and any two different components (e.g., 1 ADC and 1 temperature sensor).
(b)Which approach would you recommend for implementation and why?

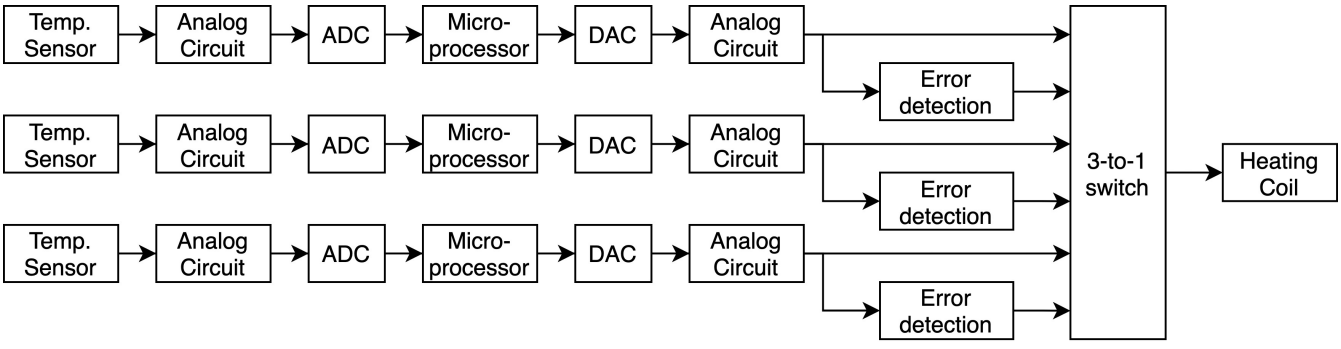


Ans:

(a)
Approach 1: passive hardware redundancy — 5MR



Approach 2: active hardware redundancy — standby sparing



Approach 1 are designed to achieve fault tolerance without requiring any action on the part of the system, while Approach 2 achieve fault tolerance by detecting the existence of faults and performing some action to remove the faulty hardware from the system.

(b) The selection should be based on more information such as the extra redundancy required, the complexity of the circuitry, the fault confinement capability, and the limitations on the system weight, cost, size, and power consumption, and the available resources.

8.(10 points) Moon Systems, a manufacturer of scientific workstations, produces its Model 13 System at sites S1, S2, S3; 20% at S1, 35% at S2, and the remaining 45% at S3. The probability that a Model 13 System will be found defective upon receipt by a customer is 0.01if it is shipped from site S1, 0.06 if from S2, and 0.03 if from S3.

(a)What is the probability that a Model 13 System selected at random at a customer location will be found defective?
(b) Suppose a Model 13 System selected at random is found to be defective at a customer location. What is the probability that it was manufactured at site S3?

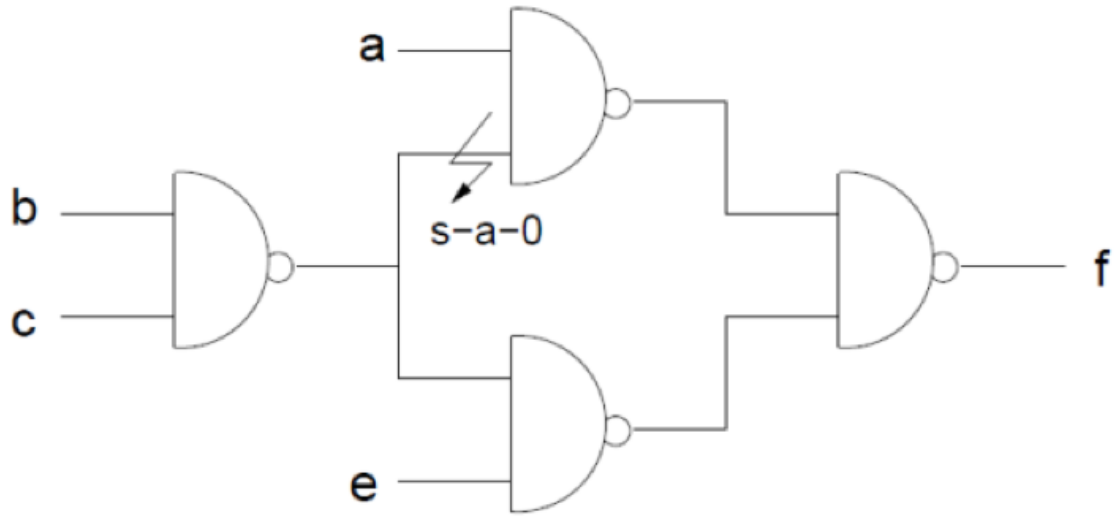
Ans:

(a) Let D be the event that a Model 13 is found to be defective at a customer site.

$$P(D) = P(D \mid S1)P(S1) + P(D \mid S2)P(S2) + P(D \mid S3)P(S3)$$
$$= (0.01)(0.2) + (0.06)(0.35) + (0.03)(0.45) = 0.0365$$

(b) $P(S3 \mid D) = \frac{P(D|S3)P(S3)}{P(D)} = \frac{(0.03)(0.45)}{0.0365} = 0.3699$

9. (10 points) Find all tests for the stuck-at-0 fault on the marked line.



Ans:

a	b	c	e	f	f(s-a-0)
1	1	1	1	0	
1	1	1	0	0	
1	1	0	1	1	
1	1	0	0	1	0
1	0	1	1	1	
1	0	1	0	1	0
1	0	0	1	1	
1	0	0	0	1	0
0	1	1	1	0	
0	1	1	0	0	
0	1	0	1	1	
0	1	0	0	0	
0	0	1	1	1	
0	0	1	0	0	
0	0	0	1	1	
0	0	0	0	0	

(a, b, c, e) = {(1, 1, 0, 0), (1, 0, 1, 0), (1, 0, 0, 0)}