Please Note

Each chapter ends with case studies and accompanying exercises. Authored by experts in industry and academia, the case studies explore key chapter concepts and verify understanding through increasingly challenging exercises. Instructors should find the case studies sufficiently detailed and robust to allow them to create their own additional exercises.

Brackets for each exercise (<chapter.section>) indicate the text sections of primary relevance to completing the exercise. We hope this helps readers to avoid exercises for which they haven't read the corresponding section, in addition to providing the source for review. Exercises are rated, to give the reader a sense of the amount of time required to complete an exercise:

[10] Less than 5 min (to read and understand)
[15] 5–15 min for a full answer
[20] 15–20 min for a full answer
[25] 1 h for a full written answer
[30] Short programming project: less than 1 full day of programming
[40] Significant programming project: 2 weeks of elapsed time
[Discussion] Topic for discussion with others

Case Study 1: Chip Fabrication Cost

Concepts illustrated by this case study

- Fabrication Cost
- Fabrication Yield
- Defect Tolerance Through Redundancy

Many factors are involved in the price of a computer chip. Intel is spending $7 billion to complete its Fab 42 fabrication facility for 7 nm technology. In this case study, we explore a hypothetical company in the same situation and how different design decisions involving fabrication technology, area, and redundancy affect the cost of chips.

1.1 Practice Problem

[10/10]<1.6> Figure 1.26 gives hypothetical relevant chip statistics that influence the cost of several current chips. In the next few exercises, you will be exploring the effect of different possible design decisions for the Intel chips.

Chip	Die Size (mm^2)	Estimated defect rate (per cm^2)	*N*	Manufacturing size (nm)	Transistors (billion)	Cores
BlueDragon	180	0.03	12	10	7.5	4
RedDragon	120	0.04	14	7	7.5	4
Phoenix[8]	200	0.04	14	7	12	8

Figure 1.26 Manufacturing cost factors for several hypothetical current and future processors.

Problem

a) [10] <1.6> What is the yield for the Phoenix chip?

b) [10] <1.6> Why does Phoenix have a higher defect rate than BlueDragon?

Solution

a) $Yield = 1/(1 + (0.04 \times 2))^{14} = 0.34$

b) It is fabricated in a larger technology, which is an older plant. As plants age, their process gets tuned, and the defect rate decreases.

1.2 Practice Problem

Problem

[20/20/20/20] <1.6> They will sell a range of chips from that factory, and they need to decide how much capacity to dedicate to each chip. Imagine that they will sell two chips. Phoenix is a completely new architecture designed with 7 nm technology in mind, whereas RedDragon is the same architecture as their 10 nm BlueDragon. Imagine that RedDragon will make a profit of $15 per defect-free chip. Phoenix will make a profit of $30 per defect-free chip. Each wafer has a 450 mm diameter.

a) [20] <1.6> How much profit do you make on each wafer of Phoenix chips?

b) [20] <1.6> How much profit do you make on each wafer of RedDragon chips?

c) [20] <1.6> If your demand is 50,000 RedDragon chips per month and 25,000 Phoenix chips per month, and your facility can fabricate 70 wafers a month, how many wafers should you make of each chip?

Solution

a) Phoenix:

$$Dies\ per\ wafer = \frac{\pi \times (\frac{45}{2})^2}{2} - \frac{\pi \times 45}{sqrt\ (2 \times 2)} = 795 - 70.7 = 724.5 = 724$$

$$Yield = 1/(1 + (0.04 \times 2))^{14} = 0.340$$

$$Profit = 724 \times 0.34 \times 30 = \$7384.80$$

b) Red Dragon:

$$Dies\ per\ wafer = \frac{\pi \times (\frac{45}{2})^2}{2} - \frac{\pi \times 45}{sqrt\ (2 \times 1.2)} = 1325 - 91.25 = 1234$$

$$Yield = 1/(1 + (0.04 \times 1.2))^{14} = 0.519$$

$$Profit = 1234 \times 0.519 \times 15 = \$9601.71$$

c) Phoenix chips: 25,000/724¼34.5 wafers needed
Red Dragon chips: 50,000/1234¼40.5 wafers needed
Therefore, the most lucrative split is 40 Red Dragon wafers, 30 Phoenix wafers.

1.3 Practice Problem

Problem

[20/20] <1.6> Your colleague at AMD suggests that, since the yield is so poor, you might make chips more cheaply if you released multiple versions of the same chip, just with different numbers of cores. For example, you could sell Phoenix8, $Phoenix^4$, $Phoenix^2$, and $Phoenix^1$, which contain 8, 4, 2, and 1 cores on each chip, respectively. If all eight cores are defect-free, then it is sold as $Phoenix^8$. Chips with four to seven defect-free cores are sold as $Phoenix^4$, and those with two or three defect-free cores are sold as $Phoenix^2$. For simplification, calculate the yield for a single core as the yield for a chip that is 1/8the area of the original Phoenix chip. Then view that yield as an independent probability of a single core being defect free. Calculate the yield for each configuration as the probability of at the corresponding number of cores being defect free.

a) [20] <1.6> What is the yield for a single core being defect free as well as the yield for $Phoenix^4$, $Phoenix^2$ and $Phoenix^1$?
b) [5] <1.6> Using your results from part a, determine which chips you think it would be worthwhile to package and sell, and why.
c) [10] <1.6> If it previously cost $20 dollars per chip to produce $Phoenix^8$,what will be the cost of the new Phoenix chips, assuming that there are no additional costs associated with rescuing them from the trash?

Solution

a) $Defect - Free\ single\ core = Yield = \frac{1}{1+0.04\times0.25}^{14} = 0.87$
Equation for the probability that N are defect free on a chip:

$$\#combinations \times (0.87)^N \times (1-0.87)^{8-N}$$

# defect-free	# combinations	Probability
8	1	0.32821167
7	8	0.39234499
6	28	0.20519192
5	56	0.06132172
4	70	0.01145377
3	56	0.00136919
2	28	0.0001023
1	8	4.3673E-06
0	1	8.1573E-08

Yield for $Phoenix^4$: (0.39 + 0.21 + 0.06 + 0.01) = 0.57

Yield for $Phoenix^2$: (0.001 + 0.0001) = 0.0011

Yield for Phoenix^1: 0.000004

b) It would be worthwhile to sell Phoenix$_4$. However, the other two have such a low probability of occurring that it is not worth selling them.

c) $20 = Wafer size / odd dpw_0:28
Step 1: Determine how many Phoenix4 chips are produced for every Phoenix8 chip.
There are 57/33 Phoenix4 chips for every Phoenix8 chip = 1.73
$30 + 1:73 X $25 = $73:25

Exercises

1.7 Practice Problem

[10/15/15/10/10]<1.4,1.5> One challenge for architects is that the design created today will require several years of implementation, verification, and testing before appearing on the market. This means that the architect must project what the technology will be like several years in advance. Sometimes, this is difficult to do.

Problem

a) [10] <1.4> According to the trend in device scaling historically observed by Moore's Law, the number of transistors on a chip in 2025 should be how many times the number in 2015?
b) [15] <1.5> The increase in performance once mirrored this trend. Had performance continued to climb at the same rate as in the 1990s, approximately what performance would chips have over the VAX-11/780 in 2025?
c) [15] <1.5> At the current rate of increase of the mid-2000s, what is a more updated projection of performance in 2025?
d) [10] <1.4> What has limited the rate of growth of the clock rate, and what are architects doing with the extra transistors now to increase performance?
e) [10] <1.4> The rate of growth for DRAM capacity has also slowed down. For 20 years, DRAM capacity improved by 60% each year. If 8 Gbit DRAM was first available in 2015, and 16 Gbit is not available until 2019, what is the current DRAM growth rate?

Solution

a) Somewhere between 1.410 and 1.5510, or 28.9 - 80*x*
b) 6043 in 2003, 52% growth rate per year for 12 years is 60,500,000 (rounded)
c) 24,129 in 2010, 22% growth rate per year for 15 years is 1,920,000 (rounded)
d) Multiple cores on a chip rather than faster single-core performance
e) $2 = x^4$, $x = 1.032$, 3.2% growth

1.8 Practice Problem

[10/10] <1.5> You are designing a system for a real-time application in which specific deadlines must be met. Finishing the computation faster gains nothing. You find that your system can execute the necessary code, in the worst case, twice as fast as necessary.

Problem

a) [10] <1.5> How much energy do you save if you execute at the current speed and turn off the system when the computation is complete?
b) [10]<1.5> How much energy do you save if you set the voltage and frequency to be half as much?

Solution

a) 50%
b) Energy: $Energy_{new}/Energy_{old}$ = $(Voltage \times 1/2)^2/Voltage^2 = 0.25$

1.9 Practice Problem

[10/10/20/20] <1.5> Server farms such as Google and Yahoo! provide enough compute capacity for the highest request rate of the day. Imagine that most of the time these servers operate at only 60% capacity. Assume further that the power does not scale linearly with the load; that is, when the servers are operating at 60% capacity, they consume 90% of maximum power. The servers could be turned off, but they would take too long to restart in response to more load. A new system has been proposed that allows for a quick restart but requires 20% of the maximum power while in this "barely alive" state.

Problem

a) [10]<1.5> How much power savings would be achieved by turning off 60% of the servers?
b) [10] <1.5> How much power savings would be achieved by placing 60% of the servers in the "barely alive" state?
c) [20] <1.5> How much power savings would be achieved by reducing the voltage by 20% and frequency by 40%?
d) [20] <1.5> How much power savings would be achieved by placing 30% of the servers in the "barely alive" state and 30% off?

Solution

a) 60%
b) 0.4+0.6 X 0.2 = 0.58, which reduces the energy to 58% of the original energy
c) newPower/oldPower = ½Capacitance X $(Voltage \times 0.8)^{z}$ X (Frequency X 0.6)/½ Capacitance X Voltage X Frequency = 0.8^{z} X 0.6 = 0.256 of the original power.
d) 0.4+0.3 X 2 = 0.46, which reduces the energy to 46% of the original energy

1.10 Practice Problem

[10/10/20] <1.7> Availability is the most important consideration for designing servers, followed closely by scalability and throughput.

Problem

a) [10]<1.7> We have a single processor with a failure in time (FIT) of 100. What is the mean time to failure (MTTF) for this system?
b) [10]<1.7> If it takes one day to get the system running again, what is the availability of the system?

Solution

a) $10^{Z}/100 = 10^{7}$
b) $10^{7}/10^{7}+24 = 1$

1.11 Practice Problem

[20/20/20]<1.1, 1.2, 1.7> In a server farm such as that used by Amazon or eBay, a single failure does not cause the entire system to crash. Instead, it will reduce the number of requests that can be satisfied at any one time.

Problem

a) [20]<1.7> If a company has 10,000 computers, each with an MTTF of 35 days, and it experiences catastrophic failure only if 1/3 of the computers fail, what is the MTTF for the system?
b) [20] <1.1, 1.7> If it costs an extra $1000, per computer, to double the MTTF, would this be a good business decision? Show your work.
c) [20]<1.2> Figure 1.3 shows, on average, the cost of down times, assuming that the cost is equal at all times of the year. For retailers, however, the Christmas season is the most profitable (and therefore the most costly time to lose sales). If a catalog sales center has twice as much traffic in the fourth quarter as every other quarter, what is the average cost of downtime per hour during the fourth quarter and the rest of the year?

Solution

a) 35/10,000 X 3333 = 11.67 days

b) There are several correct answers. One would be that, with the current system, one computer fails approximately every 5 min. 5 min is unlikely to be enough time to isolate the computer, swap it out, and get the computer back on line again. 10 min, however, is much more likely. In any case, it would greatly extend the amount of time before 1/3 of the computers have failed at once. Because the cost of downtime is so huge, being able to extend this is very valuable.

c) \$90,000 = $(x + x + x + 2x)/4$
\$360,000 = $5x$
\$72,000 = x
4th quarter = \$144,000/h