CS24 Elementary Computer Organization
Chapter 1 Exercises

1.11 Assume a 15cm diameter wafer has a cost of 12, contains 84 dies, and has 0.020 defects/cm^2. Assume a 20cm diameter wafer has a cost of 15, contains 100 dies, and has 0.032 defects/cm^2.

1.11.1 [10] <1.5> Find the yield for both wafers.

From the above question, we have been given the following quantities:

Wafer 1:

Diameter	15cm
Cost	12
Dies	84
Defects	0.020 defects/cm^2

Wafer 2:

Diameter	20cm
Cost	15
Dies	100
Defects	0.032 defects/cm^2

From chapter 1.5 of the textbook, we know that the following formula will calculate the percentage of good dies from the total number of dies on the wafer:

$$Yield = \frac{1}{(1+(DefectsPerArea \times DieArea))^N}$$

In this case, N represents the number of critical processing steps, which we will quantify with the number 2.

There is one quantity from this formula that we're missing: The $DieArea$. First, lets calculate the $DieArea$ by using the following formula provided by chapter 1.5 of the textbook:

$$DiesPerWafer = \frac{WaferArea}{DieArea}$$

If we multiply $DieArea$ to the left side of the equals sign and divide

$WaferArea$ by $DiesPerWafer$, we have a formula to calculate the $DieArea$.

$$DieArea = \frac{WaferArea}{DiesPerWafer}$$

Since we have not yet calculated the $WaferArea$, we need to calculate it for each wafer using the following formula for the area of a circle:

$$Area = \pi r^2$$

Wafer 1:

$$Area = \pi(7.5cm)^2 = \boxed{176.7cm^2}$$

Wafer 2:

$$Area = \pi(10cm)^2 = \boxed{314.16cm^2}$$

With the areas of both wafers, we can now calculate the $DieArea$:

Wafer 1:

$$DieArea = \frac{176.7cm^2}{84Dies} = \boxed{2.10cm^2}$$

Wafer 2:

$$DieArea = \frac{31.16cm^2}{100Dies} = \boxed{3.14cm^2}$$

Finally, we can calculate the yield for each wafer using our calculated quantities:

Wafer 1:

$$Yield = \frac{1}{(1+(0.020DefectsPerArea \times 2.10cm^2))^2} = 0.92 \times 100 = \boxed{92\%}$$

Wafer 2:

$$Yield = \frac{1}{(1+(0.032DefectsPerArea \times 3.14cm^2))^2} = 0.83 \times 100 = \boxed{83\%}$$

1.11.2 [5] <1.5> Find the cost per die for both wafers.

To find the cost per die for both wafers, we need to utilize the following formula from chapter 1.5 of the textbook:

$$CostPerDie = \frac{CostPerWafer}{DiesPerWafer \times Yield}$$

Since we already have the $CostPerWafer$, $DiesPerWafer$, and $Yield$, we can plug those values directly into the above formula for both wafers:

Wafer 1:

$$CostPerDie = \frac{12}{84 \times 0.92} = \boxed{0.16 CostPerDie}$$

Wafer 2:

$$CostPerDie = \frac{15}{100 \times 0.83} = \boxed{0.18 CostPerDie}$$

1.11.3 [5] <1.5> If the number of dies per wafer is increased by 10% and the defects per area unit increases by 15%, find the die area and yield.

Since the values for the $NumberOfDies$ and $DefectsPerArea$ are changing, we will need to modify our current values for our wafers:

Wafer 1:

$$NewNumberOfDies = 84 Dies + 84(0.1) =$$

$$\boxed{92.4 Dies}$$

$$NewDefectsPerArea = 0.020 Defects/cm^2 + 0.020(0.15)$$

$$= \boxed{0.023 Defects/cm^2}$$

Wafer 2:

$$NewNumberOfDies = 100 Dies + 100(0.1) =$$

$$\boxed{110 Dies}$$

$$NewDefectsPerArea = 0.032 Defects/cm^2 + 0.032(0.15)$$

$$= \boxed{0.036 Defects/cm^2}$$

With these new calculations for $NumberOfDies$ and $DefectsPerArea$, we can calculate the $DieArea$ for both wafers:

$$DieArea = \frac{WaferArea}{DiesPerWafer}$$

Wafer 1:

$$DieArea = \frac{176.7cm^2}{92.4Dies} = \boxed{1.91cm^2}$$

Wafer 2:

$$DieArea = \frac{314.16cm^2}{110Dies} = \boxed{2.86cm^2}$$

Finally, we will calculate the yield for both wafers with these modified values:

Wafer 1:

$$Yield = \frac{1}{(1+(0.023DefectsPerArea\times 1.91cm^2))^2} = 0.91 \times 100 = \boxed{91\%}$$

Wafer 2:

$$Yield = \frac{1}{(1+(0.036DefectsPerArea\times 2.86cm^2))^2} = 0.82 \times 100 = \boxed{82\%}$$

1.15 Assume a program requires the execution of 50×10^6 FP instructions, 110×10^6 INT instructions, 80×10^6 L/S instructions, and 16×10^6 branch instructions. The CPI for each type of instruction is 1, 1, 4, and 2, respectively. Assume that the processor has a 2GHz clock rate.

1.15.1 [10] <1.11> By how much must we improve the CPI of FP instructions if we want the program to run two times faster?

From the above question, we know the following information:

FP:	50×10^6 Instructions	1 CPI
INT:	110×10^6 Instructions	1 CPI
L/S:	80×10^6 Instructions	4 CPI
Branch:	16×10^6 Instructions	2 CPI

To find the total amount of clock cycles in the program, we need to use the following formula:

$$ClockCycles = \sum Instructions \times CPI$$

Using the data given, we can calculate the total amount of $ClockCycles$ for the programs execution as follows:

$$ClockCycles = (FP \times CPI_{FP}) + (INT \times CPI_{INT}) + (L/S \times CPI_{L/S}) + (Branch \times CPI_{Branch})$$

$$= \boxed{5.12 \times 10^8 ClockCycles}$$

If the program runs twice as fast, we expect the program to take only half of the clock cycles calculated. Therefore,

$$NewClockCycles = \boxed{2.56 \times 10^8 ClockCycles}$$

To observe how much the CPI_{FP} instructions must be improved to achieve this amount of clock cycles, we must utilize the formula above and solve for CPI_{FP}

$$NewClockCycles = (FP \times CPI_{FP}) + (INT \times CPI_{INT}) + (L/S \times CPI_{L/S}) + (Branch \times CPI_{Branch})$$

Isolating CPI_{FP} in the above equation gives us the following:

$$CPI_{FP} = \frac{NewClockCycles - ((INT \times CPI_{INT}) + (L/S \times CPI_{L/S}) + (Branch \times CPI_{Branch}))}{FP}$$

$$= \boxed{-4.2 CyclesPerInstruction}$$

Since we calculated a $CyclesPerInstruction$ value that is negative, improving CPI_{FP} to double the programs efficiency is impossible.

1.15.2 [10] <1.11> By how much must we improve the CPI of L/S instructions if we want the program to run two times faster?

To answer this question, we must apply the same procedure as the question above, except this time, we will be isolating the $CPI_{L/S}$ variable:

$$CPI_{L/S} = \frac{NewClockCycles-((INT\times CPI_{INT})+(FP\times CPI_{FP})+(Branch\times CPI_{Branch}))}{L/S}$$
$$= \boxed{0.8 CyclesPerInstruction}$$

In order to double the speed of the program, the $CPI_{L/S}$ would need to complete an instruction every 0.8 cycles.

1.15.3 [10] <1.11> By how much is the execution time of the program improved if the CPI of INT and FP instructions is reduced by 40% and the CPI of L/S and Branch is reduced by 30%?

Reducing CPI_{INT} and CPI_{FP} by 40%, and $CPI_{L/S}$ and CPI_{Branch} by 30% will yield the following quantities:

FP:	50×10^6 Instructions	0.6 CPI
INT:	110×10^6 Instructions	0.6 CPI
L/S:	80×10^6 Instructions	2.8 CPI
Branch:	16×10^6 Instructions	1.4 CPI

To calculate the $ExecutionTime$ of the program, we can utilize the following formula:

$$ExecutionTime = \frac{ClockCycles}{ClockRate}$$

To get $ClockCycles$, we need to use the following formula once more with our newly calculated CPI values:

$$ClockCycles =$$
$$(FP\times CPI_{FP})+(INT\times CPI_{INT})+(L/S\times CPI_{L/S})+(Branch\times CPI_{Branch})$$

$$= \boxed{3.424 \times 10^8 ClockCycles}$$

We know that the $ClockRate$ of our computer is 2GHz, which is equal to $2 \times 10^9 CyclesPerSecond$. We can use this quantity as well as $ClockCycles$ to calculate the total execution time for the whole program:

$$ExecutionTime = \frac{3.424 \times 10^8 ClockCycles}{2 \times 10^9 CyclesPerSecond}$$

$$= \boxed{0.1712 Sec}$$

If we utilize the $ClockCycles$ value calculated with the original CPI values, we can determine the programs original execution time with the new one:

$$ExecutionTime = \frac{5.12 \times 10^8 ClockCycles}{2 \times 10^9 CyclesPerSecond}$$

$$= \boxed{0.256 Sec}$$

Overall, the original time of $0.256 Sec$ was reduced by 33% in order to achieve the faster execution time of $= 0.1712 Sec$.