

module 1

1.2. (A lot of these feel unclear to me.)

- a) Assembly lines in automobile manufacturing → Performance via Pipelining
- b) Suspension bridge cables → Performance via Parallelism. (They look parallel?)
- c) Aircraft that incorporate wind information → Performance via Prediction
- d) Express elevators in buildings → Make the common case fast
- e) Library reserve desk → Hierarchy of memories
 (Gives impression that the library's archives are just as fast as books in the main level)
- f) Increasing gate area on transistor → Dependability via Redundancy
- g) Adding electromagnetic aircraft catapults → Design for Moore's Law.
- h) Building self driving cars where control systems rely on existing sensor systems already installed into the base vehicle → Use abstraction to simplify design
 (we don't need to know how the underlying systems work right?)

1.4) a) since 8 bits = 1 byte, 24 bits per pixel because 8 bits = 1 color and there are 3 colors.

$$1280 \times 1024 \text{ pixels} = 1,310,720 \text{ pixels} \times 3 = [3,932,160 \text{ pixels}]$$

↑
24 bits per pixel
8 bits in a byte

$$b) 3,932,160 \text{ bytes} \times \frac{8 \text{ bit}}{\text{bytes}} \times \frac{1 \text{ second}}{100 \cdot 10^6 \text{ bits}} = [0.31 \text{ seconds}]$$

3 bytes per pixel

1.5) P1 has a 3 GHz clock rate and a CPI of 1.5.

P2 has a 2.5 GHz clock rate and a CPI of 1.0.

P3 has a 4.0 GHz clock rate and a CPI of 2.2.

Process	Clock Rate	CPI	perf
P1	$3 \times 10^9 \text{ per second}$	1.5	2×10^9
P2	2.5×10^9	1.0	2.5×10^9
P3	4×10^9	2.2	1.8×10^9

Process	Clock Rate	seconds	CPI	Cycles	Instructions
P1	$3 \times 10^9 \text{ per sec}$	10	1.5	30×10^9 cycles	$3 \times 10^9 \times 10 \div 1.5 = 2 \times 10^{10}$
P2	2.5×10^9	10	1.0	25×10^9 cycles	$2.5 \times 10^9 \times 10 \div 2.5 = 2.5 \times 10^{10}$
P3	4×10^9	10	2.2	40×10^9 cycles	$4 \times 10^9 \times 10 \div 2.2 = 1.82 \times 10^{10}$

control

option

command

control

option

c) execution time = $(\text{num of instructions} \times \text{CPI}) / (\text{clock rate})$

↑
cycles per instruction

= total clock cycles / cycles per second
= number of seconds

execution time $\times 0.7 = (\text{num of instructions} \times \text{CPI} \times 1.2) / \text{new clock rate}$

so new clock rate = old clock rate $\times 1.2 / 0.7 = 1.71 \times \text{old clock rate}$

$$\text{P1: } 3 \text{ GHz} \times 1.71 = 5.13 \text{ GHz}$$

$$\text{P2: } 2.5 \text{ GHz} \times 1.71 = 4.27 \text{ GHz}$$

$$\text{P3: } 4 \text{ GHz} \times 1.71 = 6.84 \text{ GHz}$$

1.7) nanosecond = 10^{-9}

a) CPU time = instruction count \times CPI \times clock cycle time

$$\text{CPI} = \frac{\text{CPU time}}{(\text{Instruction count} \times \text{clock cycle time})}$$

$$\text{Compiler A: } \text{CPI} = 1.1 \quad | \quad (1.0 \times 10^9 \times 1.1 \times 10^{-9}) = 1.1$$

$$\text{Compiler B: } \text{CPI} = 1.2 \quad | \quad (1.02 \times 10^9 \times 1.1 \times 10^{-9}) = 1.25$$

b) execution time A = Instruction Count A \times CPI A \times Clock cycle time A

$$\text{execution time B} = 1.2 \times 10^9 \times 1.25$$

rearrange and we know execution time A = execution time B
 $\frac{\text{clock cycle time A}}{\text{clock cycle time B}} = \frac{1.2 \times 10^9 \times 1.25}{1.0 \times 10^9 \times 1.1} \approx 1.36$

notice A is 36% slower than clock of B

c) compiler C = $6 \times 10^8 \times 1.1 = \text{execution time} = \text{CPU time}$ notice we flip the equation, A is now better than B

$$\frac{\text{performance C}}{\text{performance A}} = \frac{\text{exec time A}}{\text{exec time C}} = \frac{1 \times 10^9 \times 1.1}{6 \times 10^8 \times 1.1} \approx 1.67$$

C is 1.67 times faster than compiler A code.

C is 2.27 times faster than compiler B's code.

$$1.10.1.) \text{ yield} = \frac{1}{(1 + (\text{defect rate} \cdot \frac{\text{die area}}{2}))^2}$$

$$\text{Die Area 1} = \frac{\text{wafer area}}{\text{die count}} = \frac{\pi (7.5 \text{ cm})^2}{84} \approx 2.10 \text{ cm}^2$$

$$\text{Die Area 2} = \frac{\pi (10 \text{ cm})^2}{100} = 7 \text{ cm}^2$$

$$\text{Yield}_2 = \frac{1}{(1 + 0.031(0.5)(\pi))^2} \approx 0.91$$

$$\text{So Yield 1} = \frac{1}{1 + (\text{defect rate}, \frac{\text{die area}}{2})^2}$$

$$= \frac{1}{1 + 0.020(0.5)(2.10)^2} \approx 0.96$$

1.10.2

$$\text{cost per die}_1 = \frac{\text{cost per wafer}}{\text{Dies per wafer} \times \text{yield}_1} = \frac{12}{(84)(\cancel{0.96})} = 6.149$$

$$\text{cost per die}_2 = \frac{15}{100(0.91)} = 0.165$$

1.14.1 a) execution time $\Rightarrow \frac{\text{clock cycles}}{\text{clock rate}}$ (Typo in book, please note this in future instructions)

$$(50 \times 10^6 \times 1) + (110 \times 10^6 \times 1) + (80 \times 10^6 \times 4) + (16 \times 10^6 \times 2) / 26 \text{ Hz} = 256 \text{ ms}$$

$\approx 1.256 \text{ seconds}$

b) so if we want the execution time to be 128 ms then... execution time

$$\frac{\text{Ex-time (old)}}{\text{Ex-time (new)}} = \frac{2}{1} = \frac{(1 \times 20 + 110 \times 1 + 280 \times 4 + 200 \times 2) \times 10^6}{(\text{CPI}_{fp} \times 1 + 110 \times 1 + 280 \times 4 + 200 \times 2) \times 10^6}$$

CPI_{fp} has to be negative, impossible

b) $128 \text{ ms} = \frac{(50 \times 10^6 \times 1) + (110 \times 10^6) + (80 \times 10^6 \times ?) + (16 \times 10^6 \times 2)}{2,000,000,000}$

$$X = 0.8$$

have to make it $5 \times$ ~~fewer~~ cycles per instruction

c) Execution time = $\frac{\text{clock cycles}}{\text{clock rate}}$

$$\frac{(50 \times 10^6 \times 0.6) + (110 \times 10^6 \times 0.6) + (80 \times 10^6 \times 2.8) + (16 \times 10^6 \times 1.4)}{26 \text{ Hz}} = 171.2 \text{ ms}$$

Speed up of $\frac{256}{171.2} = 1.495$