Chapter 7

Exercises 1, 6, 10,11, 14, 15, 20, 22, 24, 29 .

Exercise 1 :

1. a) Calculate the overall speedup of a system that spends 65% of

its time on I/O with a disk upgrade that provides for 50% greater throughput.

$$\text{overall speed up} = \frac{1}{0.35+\frac{0.65}{1.5}} = 1.276$$

therefore, the speedup is **28%**

Exercise 6 :

Using armdahl's formula,

$$speedup = \frac{1}{(1-f)+(\frac{f}{k})}$$

from prompt, speedup = **112%** $\rightarrow$ **1.12** and the current processor accounts for 25% or 0.25.

$$1.12 = \frac{1}{(1-0.25)+(\frac{0.25}{k})}$$

In [9]:
```
from latex2sympy2 import latex2sympy, latex2latex
from sympy import *

tex = r"1.12 = \frac{1}{(1-0.25)+(\frac{0.25}{k})} "
val = latex2sympy(tex)
val
```

Out[9]:
```
[Eq(k, 7/4)]
```

therefore, having solved for k = 1.75, the processor needs to be 75% faster

Exercise 10:

1. **a) 90% of a program is made to run 10 times as fast (900% faster).**

$$speedup = \frac{1}{(1-0.9)+(\frac{0.9}{10})} = 5.26$$

2. **b) 80% of a program is made to run 20% faster.**

$$speedup = \frac{1}{(1-0.8)+(\frac{0.8}{1.20})} = 1.16$$

Exercise 11.

Name the four types of I/O architectures. Where are each of these typically used, and why are they used there?

Programmed IO - Used for specialized applications in embedded systems. The device is constantly polled for events. Interrupt driven IO - Interrupt driven io is used in small embedded systems such as keyboards. One key press would be an interrupt Direct Memory Access - Used for performant applications such as graphics cards and high speed sensors etc. DMA is used to relieve the processor of repetetive sequences of commands Channel IO - used for large systems with multiple users. Mainframes for example use this type of IO because they perform complex tasks and require multiplexing and logic to be applied to the data.

Exercise 14

**14. Of programmed I/O, interrupt-driven I/O, DMA, or channel I/O, which is most suitable for processing the I/O of a:*

1. a) Mouse - **Programmed IO**
2. b) Game controller **Interrupt**
3. c) CD **DMA**
4. d) Thumb drive or memory stick **channel**

Exercise15

15. Why are I/O buses provided with clock signals?

I/O buses are provided with clock signals to synchronize the timing of data transfers between the processor and I/O devices. The clock signal provides a regular signal that controls the timing of when data is transferred on the bus.

Exercise 20

50 * 4 = 200ns

50 * 10 = 500ns

50 * 1 = 50ns

sum = 750ns

Without address lines

50 * 3 = 200ns

50 * 10 = 500ns

50 * 1 = 50ns

sum = 700ns

Exercise 22:

While the memory addressing in the disk is random, the process of getting data from the disk is not and does a seek to capture the data.Therefore it's a misnomer because the **access of the device is not random**.

Exercise 24

Verify the average latency rate cited in the disk specification of Figure 7.15. Why is the calculation divided by 2?

$$\text{Latency} = \frac{\frac{60s}{7200rpm} * \frac{1000ms}{1s}}{2} = 4.1667$$

the latency listed in Figure 7.15 is 4.17, the math checks out.

The calculation is divided by 2 because on average the required sector will be half of an entire rotation (180 degrees) from the current position of the head

Exercise 29

Suppose a disk drive has the following characteristics:

1. Five surfaces
2. 1,024 tracks per surface
3. 256 sectors per track
4. 512 bytes/sector
5. Track-to-track seek time of 8ms
6. Rotational speed of 7,500rpm

1. a) What is the capacity of the drive?

$$capacity = 1024tracks * \frac{256sectors}{track} * \frac{512bytes}{sector} = 134217728bytes = 1.073GB$$

2. b) What is the access time?

access time = seek time + average rotational latency

access time = $\frac{60}{7500} + \frac{0.008}{2}$ = 12ms

3. c) Is this disk faster than the one described in Exercise 28? Explain.

using the specs provided in exercise 28

access time = $\frac{60}{5000} + \frac{0.005}{2}$ = 14.5ms

12ms<14.5ms therefore the disk in this problem is faster than the one in exercise 28

In [15]:
```
60/7500 + (60/7500)/2
```

Out[15]:
```
0.012
```

In [ ]:
```

```

In [ ]:
```

```