Q2

Connector Numer	Name of Connector	function
0	PS/2 port	legacy port used for mouse and keyboards
1	USB type A (Female)	Universal Serial Bus (USB) port for connecting other devices to the computer, e.g. for data/power transfer
2	hdmi	a digital display port to connect to a monitor to output video (and audio)
3	VGA	analog display port to displaying video only (old)
4	DVI	digital and analog display port for video
5	SPDIF optical audio port	port for transfering digital audio between computer to another device without needing to convert to analog signal first
6	Thunderbolt port	bi-directional port that can be used for audio, data (10 Gigabit/s), power, and video transfer
7	SATA port	port (typically internal) for connecting storage devices to a computer
8	RAM slots	slots for RAM sticks to be inserted. RAM is used for low latency data storage for compute tasks
9	ethernet port	port used to connect ethernet cable for connecting the computer to a network.
10	24pin power slot	internal power supply connector used to power components on the motherboard
11	PCI-E expansion slots	slots for connecting additional components, typically peripheral cards like a graphics card or networking card

Q3

$M = 3000000$

- Type A: 20% of instructions, 25ns
- Type B: 45% of instructions, 40ns
- Type C: 35% of instructions, 20ns

A)

Since

$$\begin{aligned} I_6 &= \prod_{k=1}^{6} i_k \\ &= 4 \cdot 2 \cdot 3 \cdot 4 \cdot 3 \\ &= 288 \end{aligned}$$

so one instruction at level 6 is equivalent to 288 at level 1, and the program will have $I_6 M = 864000000$ level 1 instructions.

B)

Average instruction time at level 1 is then the weighted sum of all instruction types and their time.

$$\begin{aligned} t_1 &= 0.2(25) + 0.45(40) + 0.35(20) \\ &= 30 \end{aligned}$$

C)

Recall that $t_n = I_n t_1$, so

$$\begin{aligned} I_3 &= 4 * 3 = 12 \\ t_3 &= 12 \cdot 30 \\ &= 360 \text{ ns} \end{aligned}$$

D)

same idea as c)

$$
\begin{aligned}
I_6 &= 288 \\
t_6 &= 288 * 30 \\
&= 8640 \text{ ns}
\end{aligned}
$$

E)

since $T_{prog} = Mt_N$ with $N = 6$ in this case

$$
\begin{aligned}
T_{prog} &= 3000000 * 8640 \text{ ns} \\
&= 2.592 \cdot 10^{10} \text{ ns} \\
&= 25.92 \text{ sec}
\end{aligned}
$$

F)

$$
\begin{aligned}
I_{6,new} &= 4 \cdot 2 \cdot 3 \cdot 4 \cdot 2 \\
&= 192
\end{aligned}
$$

Since everything else is the same, ratio $I_{6,new}/I_{6,old} = 192/288$ which is $2/3$. So new program can complete the same task in ~66% of the time

Q4

Assume M instructions at level n, since each instruction at level n is translated into S instructions at $n-1$ level, we have MS^5 instructions at level 1. In general, we have MS^{n-1} level 1 instructions .

If the translation was optimal, we would have MW^5 and MW^{n-1} instructions instead (level 6 and in general). Threfore the ratio is

$$
\left(\frac{S}{W}\right)^5
$$

and in general

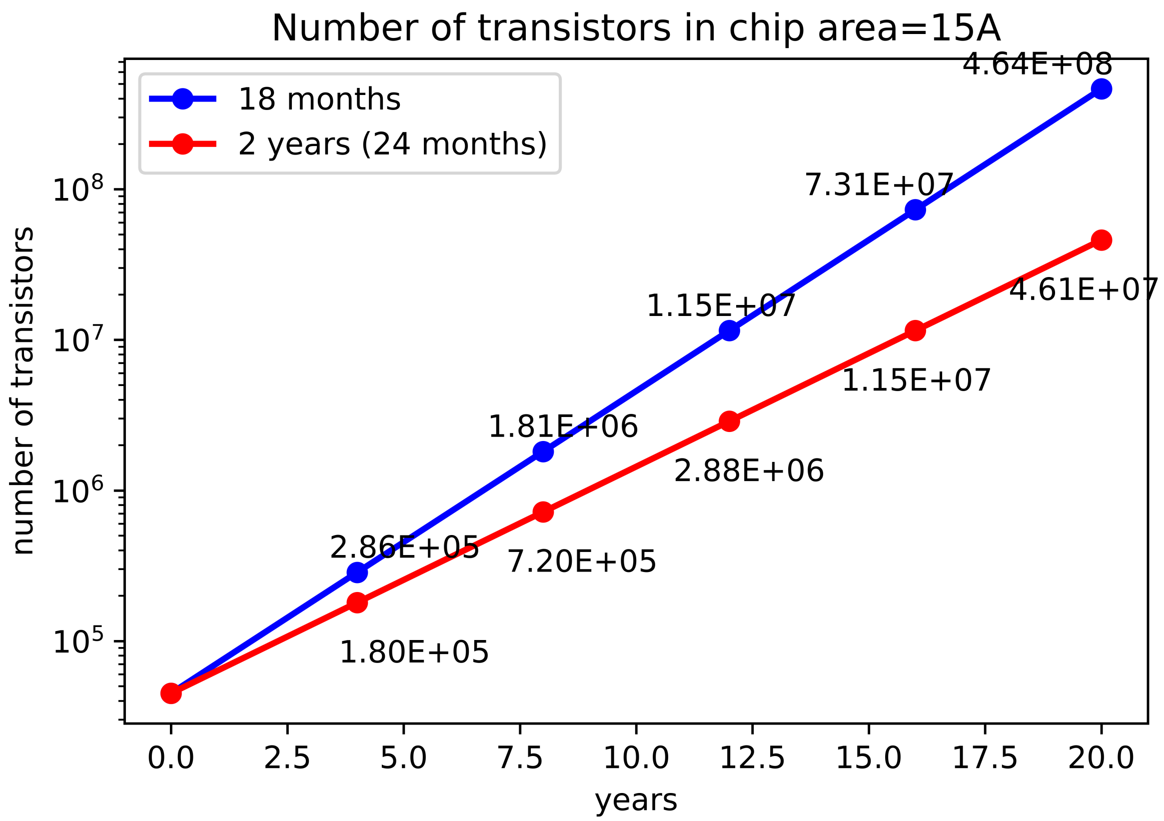
$$
\left(\frac{S}{W}\right)^{n-1}
$$

for n levels.

Q5

Chip/Device	Component Class	Approximate Price
AMD Ryzen Threadripper PRO 5995WX	CPU	$6499
Intel Xeon W-3175X Skylake X 28	CPU	$2000
Intel 10PK OPTANE 800P SERIES	Storage	$150
NVIDIA GEFORCE RTX 3090 Ti	GPU/PICE card	$1990
Corsair CX Series CX450M 450 Watt	Power Supply	$50
ASUS WS C422 SAGE/10G	(server) motherboard	$700
AMD Radeon Pro WX 9100	GPU	$2000
Intel Core i7 i7-7700K	CPU	$350
WD 40TB My Cloud PR4100 Pro Series	NAS drive	$2000
Intel Core i9-7980XE X-Series	CPU	$1200
Cisco 32GB DDR4-2666-MHZ RDIMM PC4-21300 DUAL	RAM	$400
NVIDIA QUADRO RTX 8000	GPU	$6000
Seagate BarraCuda ST3000DM008 3TB	Hard drive	$80
Kingston 16GB DDR4, 2133MHz DIMM - KVR21R15D4/16	RAM	$50
AMD Phenom II X2 560	CPU	$20
NVIDIA Tesla K80 24GB	GPU	$100 (used)
Creative Sound Blaster Z Series ZXR	Sound Card/PCIE card	$80

Q6




```
import pylab
import matplotlib.pyplot as plt

a = [pow(10, i) for i in range(10)]
fig = plt.figure()
ax = fig.add_subplot(111)
I = 15*3000

x = np.linspace(0,20,6)
y = I*2**(x/1.5)
y2 = I*2**(x/2)

line, = ax.plot(x,y,color='blue', lw=2,label="18
months",marker="o")
line, = ax.plot(x,y2,color='red', lw=2,label="2 years (24
months)",marker="o")

for i in range(1,6):
        ax.annotate( "{:.2E}".format(y[i]), (x[i], y[i]),
(0.85*x[i], 1.25*y[i]))
        ax.annotate( "{:.2E}".format(y2[i]), (x[i], y2[i]),
(0.9*x[i], 0.4*y2[i]))
```

```python
ax.legend()
ax.set_yscale('log')
ax.set_xlabel('years')
ax.set_ylabel('number of transistors')
ax.set_title('Number of transistors in chip area=15A')
plt.savefig("moores_law.png",facecolor="white",dpi=1000)
pylab.show()
```

Q6B

Let length at year 0 L_0 and length at year 24 L_{24}.

doubling every 2 years

$n_{doubling} = \frac{24}{2}$

$$L_{24} = \frac{L_0}{\sqrt{2^{n_{doubling}}}}$$
$$L_{24} = \frac{L_0}{\sqrt{2^{12}}}$$
$$ratio = \frac{L_{24}}{L_0} = \frac{1}{\sqrt{2^{12}}}$$
$$= 0.015625$$

doubling every 18 months

$n_{doubling} = \frac{24}{1.5} = 16$

$$L_{24} = \frac{L_0}{\sqrt{2^{n_{doubling}}}}$$
$$L_{24} = \frac{L_0}{\sqrt{2^{16}}}$$
$$ratio = \frac{L_{24}}{L_0} = \frac{1}{\sqrt{2^{16}}}$$
$$= 0.00390625$$