# ES 215:
# Computer Organisation and Architecture

## Assignment 2

100 points
Due Date: 22.02.2022

Jayesh Salunkhe
18110072

Link to code on Github: https://github.com/jayeshsalunkhe/ES215-Assignments

## 1.

Code can be found at abovementioned link.: Assignment 2 folder.

## 2.

Find combined throughput:

Throuhput is total work done per unit time. Here by work we measure number of instructions executed in per unit of time( sec) or millions instruction executed per unit time.

This can be expressed in two ways IPS/ MIPS as units.

Let us choose MIPS as unit

Throught for processor A is

$$\frac{IC}{Exec.Time \times 10e6} = \frac{CR}{CPI \times 10e6} = \frac{10e9}{6 \times 10e6} = 166.6 \text{ MIPS}$$

Throught for processor B is

$$\frac{IC}{Exec.Time \times 10e6} = \frac{CR}{CPI \times 10e6} = \frac{10e9}{5 \times 10e6} = 200 \text{ MIPS}$$

Combined throughput will be addition of the throughput of both processor A & B as it is given dual core. Hence

Combined Throughput = 166.6 MIPS + 200 MIPS = 366.6 MIPS

That is 366.6 Millions Instruction Per Seconds.

Similarly IPS as unit wil give

Throught for processor A is

$$\frac{IC}{Exec.Time} = \frac{CR}{CPI} = \frac{10e9}{6} = 1.667 \times 10^8 \text{ IPS}$$

Throught for processor B is

$$\frac{IC}{Exec.Time} = \frac{CR}{CPI} = \frac{10e9}{5} = 2 \times 10^8 \text{ IPS}$$

Combined throughput will be addition of the throughput of both processor A & B as it is given dual core. Hence

Combined Throughput = $1.667 \times 10^8$ IPS + $2 \times 10^8$ IPS = $3.667 \times 10^8$ IPS

That is 366.6 Millions Instruction Per Seconds.

3.

| Processor $X$ | Processor $Y$ |
| --- | --- |
| $CR = 2\ \text{GHz}$ | $CR = 4\ \text{GHz}$ |
| $IC = 10 \times 10^9$ | $\text{TC} = 7 \times 10^9$ |
| $CPI = 3$ | $CPI = 5$ |

---

$$ExT_{\text{Time}} = \frac{IC \times CPI}{CR}$$

$$ExT_{\text{x}} = \frac{10 \times 109 \times 3}{2 \times 10^9} \qquad ExT_y = \frac{7 \times 10^9 \times 5}{4 \times 10^9}$$

$$ExT_{\text{x}} = 15\text{secs} \qquad ExT_y = \frac{35}{4}\text{secs}$$

---

$$\text{Speedup } \left(\frac{y}{x}\right) = \frac{\text{ExT }(x)}{\text{ExT(y)}} = \frac{15}{35/4} = \frac{12}{7}$$

$$\text{Speedup } = \frac{12}{7} = 1.716$$

4.

Given

| $A$ | $B$ |
| --- | --- |
| $CR = 1\ \text{GHz}$ | $CR = 2 =\ \text{GHz}$ |
| $IC = 9 \times 10^9$ | $ExT_B = \frac{1}{4} ExT_A$ |
| $CPI = 1.5$ | $CPI =?$ |

$$\text{Find } ExT_A = \frac{IC_A \times CPI_A}{CR_A}$$

$$ExT_b = \frac{IC_A \times CPI_A}{4CR_A} = \frac{IC_B \times CPI_B}{CR_B}$$

Assuming same architecture for both processors, therefore IC(instruction count) will be same for the both processors. $IC_A = IC_B = 1 \text{ x } 10^9$

$$ExT_A = \frac{9 \times 10^9 \times 1.5}{1 \times 10^9}$$

$$ExT_B = \frac{1}{4} \times \frac{9 \times 10^9 \times 1.5}{1 \times 10^9} = \frac{IC_B \times CPI_B}{2 \times 10^9}$$

$$CPI_B = \frac{2 \times 10^9}{IC_B} \times \frac{1}{4} \times \frac{9 \times 10^9}{1 \times 10^9} \times 1.5$$

$$IC_B = 9 \times 10^9 \quad (Assumed)$$

$$CPI_B = \frac{1.5}{2} = \frac{3}{4} = 0.75$$

## 5.

Total power= 80 W
Total Power = Static power + Dynamic Power

$$P_{static} = IV$$
$$P_{dynamic} = 1/2\ CV^2f$$

where as the C is capacitive load of the transistor, V is voltage, and f the frequency

Since distribution of static and dynamic power is not mentioned we assume 30 W is of static power and 50 W of dynamic power. This is because dynamic power dominates bcs of switching etc.

a) $P_{dynamic} = 1/2\ CV^2f$

$$\frac{P\ dynamic\_new}{P\ dynamic\_old} = 0.5 \times C \times (5)\text{^}2 \times 5 \times 2 \times 10\text{^}9 \ / \ 0.5 \times C \times (5)\text{^}2 \times 2 \times 2 \times 10\text{^}9 = 2.5$$

New Dynamic power is **2.5** times of the old ones
$P_{dynamic}$_new = 125 W ( assumed $P_{dynamic}$_*old = 50 W)*

b) *Because frequency ~ to V...*

$$P_{dynamic} \sim to\ f^3\ or\ V^3$$

For $P_{static} = IV$
New_$P_{static}$ = old_$P_{static}$ x new_V / old_V
New_$P_{static}$ = old_$P_{static}$ x 2/ 5
New_$P_{static}$ = 30 x 2/5 = 12 W

New_$P_{dynamic}$ = old_$P_{dynamic}$ * (new_V / old_V)^3
New_$P_{dynamic}$ = old_$P_{dynamic}$ * (2 / 5)^3
New_$P_{dynamic}$ = 50 x ( ⅖)**3 = 3.2 W
Overall power consumption = 15.2 W
So **78.94 %** of total power is static power which was earlier 37.5 %.

New_$P_{dynamic}$ = old_$P_{dynamic}$ * (2 / 5)^3
New_$P_{static}$ = old_$P_{static}$ x 2/ 5

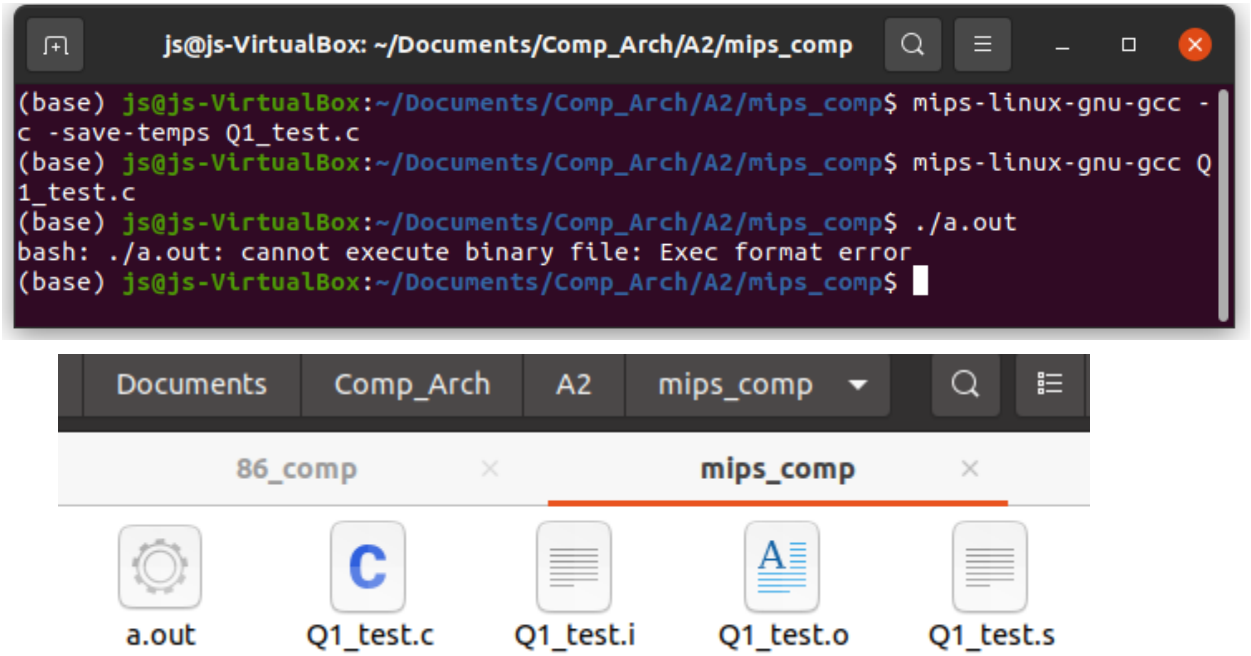
New_$P_{Total}$ = old_$P_{dynamic}$ * (2 / 5)^3 + old_$P_{static}$ x 2/ 5

**Fraction of New_$P_{static}$** = $\frac{old\_Pstatic\ *\ 2/5}{old\_Pdynamic\ *\ (2\ /\ 5)\text{^}3 + old\_Pstatic\ *\ 2/5} = \frac{old\_Pstatic}{old\_Pdynamic\ *\ (2\ /\ 5)\text{^}2 + old\_Pstatic}$

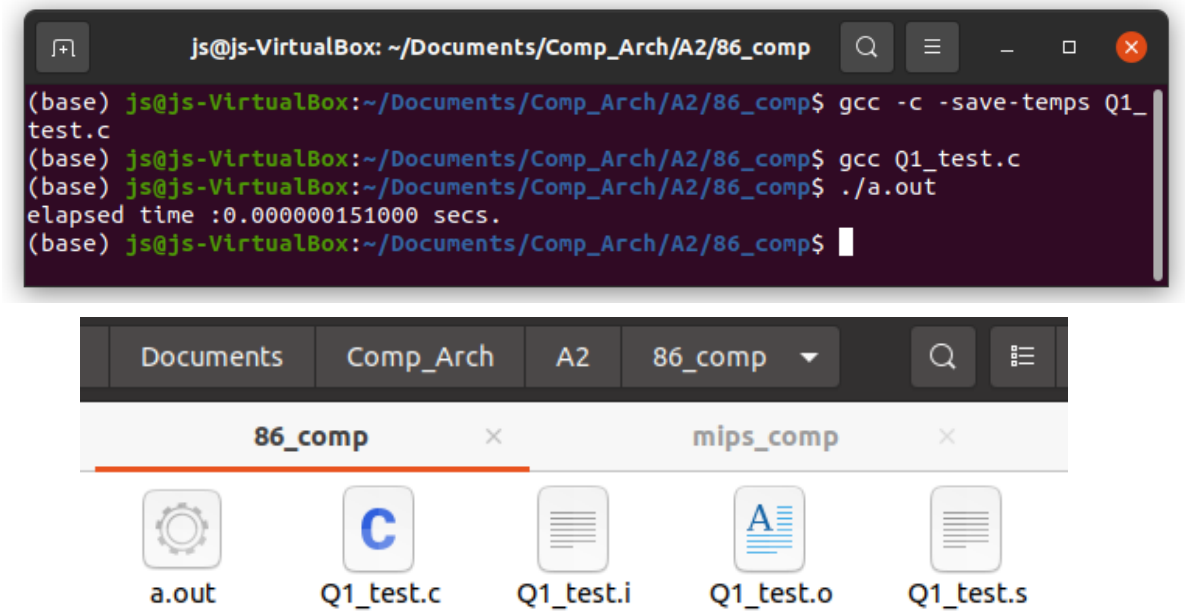
Grace Question:

## Observations :

- MIPS Compilation :




- x86 Compilation :



- Both the programs can be compiled on the system
- **Size comparison:**

| File type | architecture | File size ( in bytes) | Comments | |
|---|---|---|---|---|
| Preprocessed ( **.i)** | x86_64 | 25,292 | MIPS file size marginally larger than x86_84 | These files do not directly affect the program. But helps in debugging. |
| | MIPS | 26,688 | | |
| Compiled code ( **.s**) | x86_64 | 2,203 | MIPS file size marginally larger than x86_84<br><br>We can infer that IC(Instruction count ) may be larger in case of MIPS | These files contains assembly code which is executed on the given ISA. |
| | MIPS | 2,879 | | |
| Assembled code( .**o**) | x86_64 | 2,240 | x86_84 file size marginally larger than MIPS | Contains the machine code |
| | MIPS | 2,060 | | |
| Binary code (**.out**) | x86_64 | 16,832 | x86_84 file size larger than MIPS by the factor of 2. | Compiled executable file. |
| | MIPS | 7,996 | | |

- The output of the MIPS code can not be read because binaries of only x86 can be read on this system(being x86 system). Hence the output of the x86 program can be seen.
- As ubuntu system does not have have ABI( application binary interface) for MIPS architecture it can not implement those instructions mentioned in .s file.

## Reference:


1. `Linux Toolchain – MIPS
2. How can I execute MIPS assembly programs on an x86 linux? - Stack Overflow

3. How to install gcc-mips-linux-gnu on Ubuntu 20.04 (Focal Fossa)?
4. An Introduction to GCC - Preprocessing source files
5. https://www.linuxtopia.org/online_books/an_introduction_to_gcc/index.html