1: summarizing performance numbers

(a) Let us compute the execution time of all the systems. We use arithmetic mean for computation as it is good for time and latencies.

$averagetime = \frac{1}{n}\sum_{i=1}^{n} x_i$

BASE - $\frac{3+2.5+1+12}{4} = 4.625s$
NEW1 - $\frac{7+3+5+1}{4} = 4s$
NEW2 - $\frac{2+1+3+8}{4} = 3.5s$
NEW3 - $\frac{1+3+2+13}{4} = 4.75s$
We can see that system 'NEW2' has the least average execution time.

(b) Let us compute the average energy consumption of all the systems. We use arithmetic mean for computation as it is good for time and latencies.

$averagetime = \frac{1}{n}\sum_{i=1}^{n} x_i$

BASE - $\frac{20+40+50+15}{4} = 31.25J$
NEW1 - $\frac{10+30+15+30}{4} = 21.25J$
NEW2 - $\frac{30+60+20+20}{4} = 32.5J$
NEW3 - $\frac{70+35+30+10}{4} = 36.25J$
We can see that system 'NEW1' has the least energy consumption.

(c) we know that

$energy = Power \times time$
$Power = \frac{Energy}{time}$

Table 1: Power across machines

Individual Power				
BASE	$\frac{20}{3} = 6.67$	$\frac{40}{2.5} = 26$	$\frac{50}{1} = 50$	$\frac{15}{12} = 1.25$
NEW1	$\frac{10}{7} = 1.42$	$\frac{30}{3} = 10$	$\frac{15}{5} = 3$	$\frac{30}{1} = 30$
NEW2	$\frac{30}{2} = 15$	$\frac{60}{1} = 60$	$\frac{20}{3} = 6.67$	$\frac{20}{8} = 2.5$
NEW3	$\frac{70}{1} = 70$	$\frac{35}{3} = 11.67$	$\frac{30}{2} = 15$	$\frac{10}{13} = 0.76$

calculating the power across machines for all applications in table

Average power of the systems can be calculated using arithmetic mean

BASE - $\frac{6.67+26+50+1.25}{4} = 18.48W$
NEW1 - $\frac{1.42+10+3+30}{4} = 11.1W$
NEW2 - $\frac{15+60+6.67+2.5}{4} = 21.05W$
NEW3 - $\frac{70+11.67+15+0.76}{4} = 24.35W$
From the results, it is clear that system 'NEW1' consumes less power.

2: Optimizing CPU time

(a)

Old processor-
We know that average Cycles per instruction(CPI) is given as
$CPI_{avg} = \sum frequency cycles$
$CPI_{avg} = 0.1 * 2 + 0.05 * 1 + 0.05 * 2 + 0.3 * 1 + 0.5 * 4 = 2.65$

average PCI is given as
$PCI_{avg} = \frac{1}{CPI_{avg}} = \frac{1}{2.65} = 0.377$ for old processor

New processor-
Assumptions - For old processor, Let us assume that the total instructions are 100. Let us calculate the total instructions/ frequencies for the new processor.
Out of 100, number of MULT instruction = 50. 60% of them are MULT followed by ADD, thus 0.6*50 = 30

calculating the instruction frequencies in new processor-
number of MULT followed by ADD - FMAD = 30
number of MULT remaining = 50 - 30 = 20
number of ADD remaining = 30 - 30 = 0 (since 30 of these instructions are executed in FMAD)
To normalize the instruction freqeuncy, we need to find the total instructions in new processor.

total instructions = 20 + 30 + 0 + 10 + 5 + 5 = 70

number of Load remaining = $\frac{10}{70} * 100 = 14.28\%$ number of Store remaining = $\frac{5}{70} * 100 = 7.14\%$
number of Branch remaining = $\frac{5}{70} * 100 = 7.14\%$ number of ADD remaining = $\frac{0}{70} * 100 = 0\%$
number of MULT remaining = $\frac{20}{70} * 100 = 28.59\%$ number of FMAD remaining = $\frac{30}{70} * 100 = 42.85\%$

Now using the same formula for
$CPI_{avg} = 2 * 0.142 + 1 * 0.071 + 2 * 0.071 + 1 * 0 + 4 * 0.285 + 4 * 0.428 = 3.35$

$PCI_{avg} = \frac{1}{CPI_{avg}} = 0.29$ for new processor

(b) Old Processor -
CPI = 2.65
InstructionCount (IC) = 100
CPU time = CPI*IC*CT = 100*2.65*CT = 265*CT

New Processor -
CPI = 3.35
InstructionCount (IC) = 70
CPU time = CPI*IC*CT = 70*3.35*CT = 235*CT

efficiency = $\frac{OldCPUtime}{NewCPUTime} = \frac{265*CT}{235*CT}$
Assuming the same cycle time(CT), we have efficiency = 1.13, since the value is more than 1, the new processor is faster(speedup) than the older processor.

3: Amdahl's Law

From Amdahl's law of diminishing returns, we know that
$speedup_{overall} = \frac{ExecutionTime_{old}}{ExecutionTime_{new}} = \frac{1}{(1-Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$

Using the above equation in following sections for analyzing the energy optimizations

(a)reducing wireless interface energy by 10%
$Fraction_{enhanced} = 50\% = 0.5$
$Speedup_{enhanced} = \frac{x}{(1-\%reduced)*x} = \frac{1}{0.9} = 1.11$
substituting the values in Amdahl's equation:
$Speedup_{overall} = \frac{1}{(1-0.5)+\frac{0.5}{1.11}} = 1.053$

(b)reducing CPU energy by 60%
$Fraction_{enhanced} = 10\% = 0.1$
$Speedup_{enhanced} = \frac{x}{(1-\%reduced)*x} = \frac{1}{0.4} = 2.5$
substituting the values in Amdahl's equation:
$Speedup_{overall} = \frac{1}{(1-0.1)+\frac{0.1}{2.5}} = 1.064$

(c)reducing display energy by 50%
$Fraction_{enhanced} = 20\% = 0.2$
$Speedup_{enhanced} = \frac{x}{(1-\%reduced)*x} = \frac{1}{0.5} = 2$
substituting the values in Amdahl's equation:
$Speedup_{overall} = \frac{1}{(1-0.2)+\frac{0.2}{2}} = 1.11$
From the above experiments, it is clear that reducing display energy by 50% gives the best energy savings.

4: Power and energy

We know that
$power = voltage \times Current(P = VI)$
$Energy = Power \times Time(E = PT)$
$Energy = (Power_{static} + Power_{dynamic}) \times Time(E = PT)$
$power_{static} = voltage \times Current_{static}$
$power_{dynamic} = activity \times capacitance \times voltage^2 \times frequency$
given:
frequency (f) = 2 GHz
Application time (t) = 15 s
dynamic power ($P_{dynamic}$)= 70 W
static power (P_{static})= 30 W

(a) Using the third equation -

Energy = (70+30)*15 = 1500J

(b) frequency scaled down by 30%. Change in frequency affects the execution time and dynamic power, but the static power remains the same.
new frequency - $f_{new} = (1 - 0.3) * 2GHz = 1.4GHz$
new time - $t_{new} = \frac{f}{f_{new}} * time = \frac{2}{1.4} * 15 = 21.43s$
dynamic power is directly proportinal to f, thus:
new dynamic power - $P_{dynamic} = \frac{70*0.7f}{f} = 49W$
energy = (30+49)*21.43 = 1693J

(c) both voltage and frequency scaled down by 30%. Hence time, static power and dynamic power change.

new frequency - $f_{new} = (1 - 0.3) * 2GHz = 1.4GHz$
new time - $t_{new} = \frac{f}{f_{new}} * time = \frac{2}{1.4} * 15 = 21.43s$
new static power - $P_{static} = 30 * 0.7 = 21W$(from 4th equation)
new dynamic power - $P_{dynamic} = 70 * 0.7 * 0.7^2 = 24.01W$(from 5th equation)
energy = (21+24)*21.43 = 964.35J

5: Instruction Set Architecture

1. LOAD R5, 6000(R0)
   R5 = Mem[6000+R0]
   R5 = Mem[6000+1000]
   R5 = Mem[7000]
   R5 = 1

2. ADD R4,(R4)
   R4 = R4 + Mem[R4]
   R4 = 6000 + Mem[6000]
   R4 = 6000 + 12
   R4 = 6012

3. SUB R2,R1
   R2 = R2 - R1
   R2 = 99 - 25
   R2 = 74

4. LOAD R6, @(R0)
   R6 = Mem[Mem[R0]]
   R6 = Mem[Mem[1000]]
   R6 = Mem[3000]
   R6 = 33

5. ADD R6, R4
   R6 = R6 + R4
   R6 = 33 + 6012
   R6 = 6045

6. SUB R5, R6
   R5 = R5 - R6
   R5 = 1 - 6045
   R5 = -6044

7. ADD R2, R5
   R2 = R2 + R5
   R2 = 74 + (-6044)
   R2 = -5970

8. ADD R2, (R3+R0)
   R2 = R2 + Mem[R3 + R0]
   R2 = -5970 + Mem[4000 + 1000]

R2 = -5970 + Mem[5000]
R2 = -5970 + 71
R2 = -5899