HW1:

Appendix A: A.3, A.9(a,b)

Chapter 1: 1.1, 1.4, 1.7, 1.9, 1.10, and 1.16

These problems help you understand the prerequisites and some fundamental concepts.

Note: Question A3: for branches, you can assume 60% of branches are taken.

Question A9: The sub-questions a and b are considered separately.

Points:

A3: 6 pts, A9: 10 pts, for other problems, 4 pts for each sub-question

Reference solutions:

A3.

For gobmk and mcf, the average instruction frequencies are shown in Figure S.3. The effective CPI for programs in Figure A.29 is computed by combining instruction category frequencies with their corresponding average CPI measurements. Note that the total instruction frequency of gobmk adds up to 99%, so an additional category is considered to account for the remaining 1% of instructions, each requiring 3.0 clock cycles.

Instruction category	gobmk	mcf	Average of gobmk and mcf	Clock cycles
ALU operations	50%	29%	39.5%	1.0
Loads	21%	35%	28%	3.5
Stores	12%	11%	11.5%	2.8
Branches	14%	24%	19%	T: 4.0, NT: 2.0
Jumps	2%	1%	1.5%	2.4
Other	1%	0%	0.5%	3.0

Figure S.3 RISC-V dynamic instruction mix average for gobmk and mcf programs.

Effective CPI = sum(Instruction category frequency × Clock cycles for category)

=39.5%*1.0+28%*3.5+11.5%*2.8+19%*[0.6*4+0.4*2]+1.5%*2.4+0.5%*3

=2.356

A.9

This exercise focuses on the challenges related to instruction set encoding. The length of an instruction is 14 bits. There are 64 general-purpose registers, thus the size of the address fields is 6 bits for each register operand. Since there are instructions with two-addresses, then the 2 most significant bits are reserved

for opcodes. The notation addr[13:0] is used to represent the 14 bits of an instruction.

a)

a. First, we need to support 3 two-address instructions. These can be encoded as follows:

	addr[13:12]	addr[11:6]	addr[5:0]
3 two-address instructions	‘00’, ‘01’, ‘10’	‘000000’ to ‘111111’	‘000000’ to ‘111111’
other encodings	‘11’	‘000000’ to	‘000000’ to

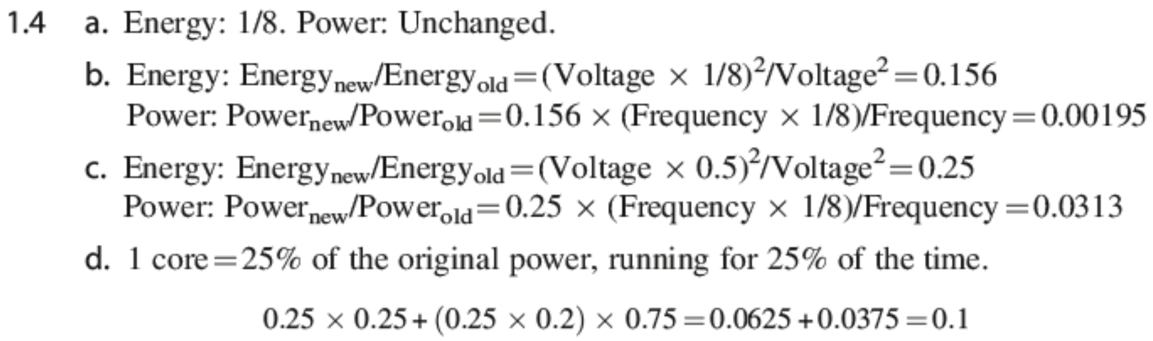
Hence, the one-address and zero-address instructions must be encoded using addr [13:12] =‘11’. For the one-address instructions, the opcode is extended by using 63 of the possible 64 combinations from the bits in addr[11:6], that is, from ‘000000’ to ‘111110’. Consequently, the opcode of zero-address instructions is extended with the remaining combination of ‘111111’ from addr[11:6]. There are 45 zero-address instructions, so using ‘000000’ to ‘101100’ from addr[5:0] suffices for the encoding.

	addr[13:12]	addr[11:6]	addr[5:0]
3 two-address instructions	‘00’, ‘01’, ‘10’	‘000000’ to ‘111111’	‘000000’ to ‘111111’
63 one-address instructions	‘11’	‘000000’ to ‘111110’	‘000000’ to ‘111111’
45 zero-address instructions	‘11’	‘111111’	‘000000’ to ‘101100’
19 unused encodings	‘11’	‘111111’	‘101101’ to ‘111111’

b. The 3 two-address instructions can be encoded similar to part (a). The one- address and zero-address instructions must be encoded using addr[13:12] =‘11’. In order to encode 65 one-address instructions, at least 7 bits are required, so addr[11:5] can be used for this encoding. Then addr[4:0] is left for extending the opcode of zero-address instructions. The maximum number of zero-address instructions that can be encoded for this processor is 2^5=32, thus encoding 35 zero-address instructions is not possible.

1.1 a. Yield= $1/(1 + (0.04 \times 2))^{14}$ = 0.34

b. It is fabricated in a larger technology, which is an older plant. As plants age, their process gets tuned, and the defect rate decreases.



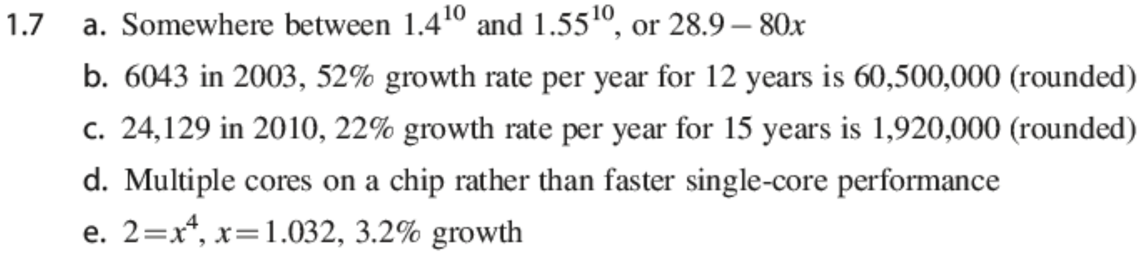
1.4 a. Energy: 1/8. Power: Unchanged.

b. Energy: $\text{Energy}_{\text{new}}/\text{Energy}_{\text{old}} = (\text{Voltage} \times 1/8)^2/\text{Voltage}^2 = 0.156$
Power: $\text{Power}_{\text{new}}/\text{Power}_{\text{old}} = 0.156 \times (\text{Frequency} \times 1/8)/\text{Frequency} = 0.00195$

c. Energy: $\text{Energy}_{\text{new}}/\text{Energy}_{\text{old}} = (\text{Voltage} \times 0.5)^2/\text{Voltage}^2 = 0.25$
Power: $\text{Power}_{\text{new}}/\text{Power}_{\text{old}} = 0.25 \times (\text{Frequency} \times 1/8)/\text{Frequency} = 0.0313$

d. 1 core = 25% of the original power, running for 25% of the time.

$$0.25 \times 0.25 + (0.25 \times 0.2) \times 0.75 = 0.0625 + 0.0375 = 0.1$$



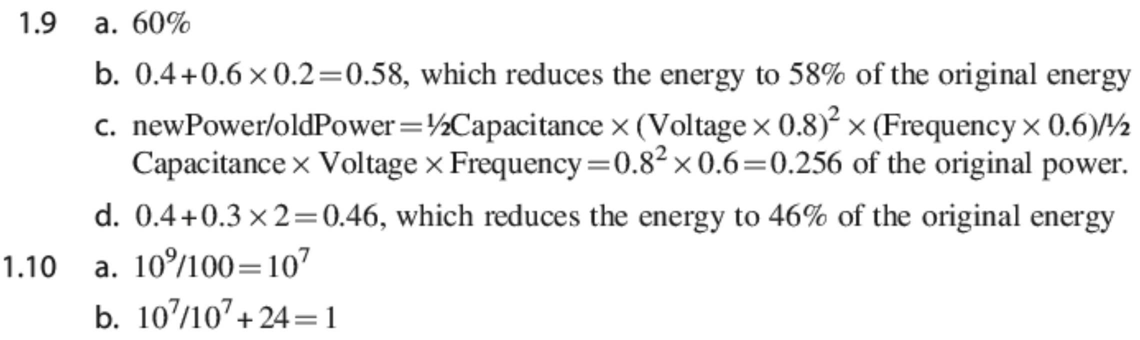
1.7 a. Somewhere between 1.4^{10} and 1.55^{10}, or $28.9 - 80x$

b. 6043 in 2003, 52% growth rate per year for 12 years is 60,500,000 (rounded)

c. 24,129 in 2010, 22% growth rate per year for 15 years is 1,920,000 (rounded)

d. Multiple cores on a chip rather than faster single-core performance

e. $2 = x^4$, $x = 1.032$, 3.2% growth



1.9 a. 60%

b. $0.4 + 0.6 \times 0.2 = 0.58$, which reduces the energy to 58% of the original energy

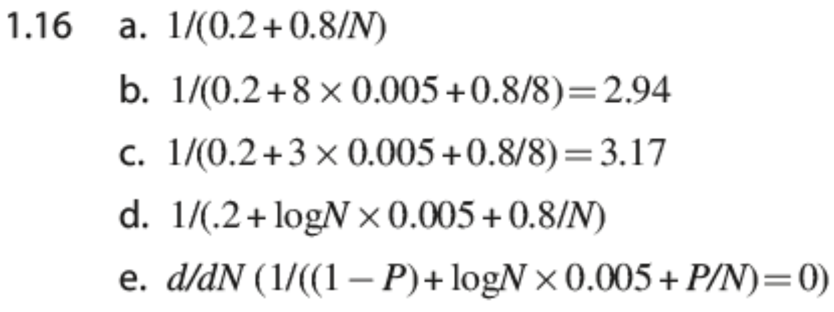
c. newPower/oldPower = ½Capacitance × (Voltage × 0.8)^2 × (Frequency × 0.6)/½ Capacitance × Voltage × Frequency = $0.8^2 \times 0.6 = 0.256$ of the original power.

d. $0.4 + 0.3 \times 2 = 0.46$, which reduces the energy to 46% of the original energy

1.10 a. $10^9/100 = 10^7$

b. $10^7/10^7 + 24 = 1$

C. $1/(100*1000/10^9)$=10,000 hours.



1.16 a. $1/(0.2 + 0.8/N)$

b. $1/(0.2 + 8 \times 0.005 + 0.8/8) = 2.94$

c. $1/(0.2 + 3 \times 0.005 + 0.8/8) = 3.17$

d. $1/(.2 + \log N \times 0.005 + 0.8/N)$

e. $d/dN\ (1/((1 - P) + \log N \times 0.005 + P/N) = 0)$