Q1
Suppose a new CPU has 70% of the capacitive load of the previous generation, 15% voltage reduction, and 20% slower clock. How much power reduction the new CPU brings compared to the old generation?

Power = Capacitive load * Voltage^2 * Frequency

Power reduced: 70% * 85% * 85% * 80% = 40.46%

1 - 40.46% = 59.54%

Q2
The following measurements have been made using a simulator for a design. What is the design's CPI?

Instruction class	CPI	Frequency
ALU	1	40%
Load	4	30%
Branches	2	20%
Stores	3	10%

1*40% + 4*30% + 2*20% + 3*10% = 2.3

Q3
Using the measurements in Question 2, how much faster would the machine be if a better data cache could reduce the loads and stores by one cycle each?
1*40% + 3*30% + 2*20% + 2*10% = 1.9

Q4
Write the MIPS assembly code that creates the 32-bit constant
0001 0000 0000 0010 0100 1001 0010 0100, and stores it in register $r1.

```
Lui $r1 0b0001000000000010
Ori $r1 0b0100100100100100
```

See:
http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html

Q7
For the following C statement, please give the corresponding MIPS assembly code (assume that the variables f, g, h and i are given and could be considered 32-bit integer as declared in a C program.) Use a minimal number of MIPS assembly instructions.

F = g + (h - 5);

```
addi f, h, −5 (note, no subi)
add f, f, g
```

Q8
In this question, we examine how resource hazards, control hazards, and Instruction Set Architecture (ISA) design can affect pipelined execution. Problems in this exercise refer to the following fragment of MIPS code:

sw r16, 12(r6)
lw r16, 8(r6)
beq r5, r4, Label //Assume r5!=r4
add r5, r1, r4
slt r5, r15, r4
Assume that individual pipeline stages have the following latencies:

IF	ID	EX	MEM	WB
150ps	100ps	120ps	140ps	80ps

Assume that all branches are perfectly predicted (effectively eliminating all control hazards) and that no delay slots are used. If we only have one memory (for both instruction and data), there is a structural hazard every time we need to fetch an instruction in the same cycle in which another instruction accesses data. To guarantee "forward progress", this hazard must always be resolved in favor of the instruction that accesses data. What is the total execution time of this instruction sequence in the 5-stage pipeline that only has one memory?

SW	F	D	E	M	W							
LW		F	D	E	M	W						
BEQ			F	D	E	M	W					
ADD							F	D	E	M	W	
SLT								F	D	E	M	W

sw r16, 12(r6)
lw r16, 8(r6)
beq r5, r4, Label
add r5, r1, r4
slt r5, r15, r4
Total time: 12*150ps = 1800ps

Q9

Using the latencies and the assembly code in the last question. In this question, let us continue to assume that all branches are perfectly predicted (this eliminates all control hazards) and that no delay slots are used. If we change load/store instructions to use a register (without an offset) as the address, these instructions no longer need to use the ALU. As a result, MEM and EX stages can be *overlapped* and the pipeline has only 4 stages. Change this code to accommodate this changed ISA. Assuming this change does not affect clock cycle time, what speedup is achieved in this instruction sequence?

SW	F	D	M	W						
LW		F	D	M	W					
BEQ					F	D	E	W		
ADD						F	D	E	W	
SLT							F	D	E	W

Total time: 10*150ps = 1500ps

Speedup 1800/1500 = 1.2

Q10

For the following repeating pattern (e.g., in a loop) of branch outcomes:

N, N, N, T, N, T, T, T, T, T, N, T, N, T, N, T

a. What is the accuracy of always-taken and always-not-taken predictors for this sequence of branch outcomes?

always-taken: 9/16

always-not-taken: 7/16

b. What is the accuracy of a one-bit predictor assuming the predictor starts off in the predict-taken state. What is

the accuracy of this predictor if this pattern repeats forever?
Accuracy: 6/16
Forever: 6/16

c. What is the accuracy of a two-bit dynamic branch predictor assuming that the predictor starts off in the weakly predict-not-taken state? What is the accuracy of this predictor if this pattern is repeated forever?

N, N, N, T, N, T, T, T, T, T, N, T, N, T, N, T
n, N, N, N, n, N, n, t, T, T, T, t, T, t, T, t, T
Accuracy: 10/16

N, N, N, T, N, T, T, T, T, T, N, T, N, T, N, T
n, N, N, N, n, N, n, t, T, T, T, t, T, t, T, t,
N, N, N, T, N, T, T, T, T, T, N, T, N, T, N, T
T, t, n, N, n, N, n, t, T, T, T, t, T, t, T, t,
…

Accuracy: 50%