Lecture 5 Pipelining Exercises

Q1:
The standard MIPS pipeline has 5 stages. Suppose we split each of those stages in half, resulting in a total of 10 stages. Based on what you know about pipelines, which of these is the most likely outcome of increasing the number of stages:

A. Instruction count would stay the same, CPI would increase, cycle time would increase
B. Instruction count would decrease, CPI would decrease, cycle time would decrease
C. Instruction count would stay the same, CPI would increase, cycle time would decrease
D. Instruction count would increase, CPI would decrease, cycle time would increase
E. Instruction count, CPI, and cycle time would remain the same

Answer: C

As an example, consider a program with four instructions,

A 5-stage pipeline without any stalling:
Instr 1: _____ _____ _____ _____ _____
Instr 2:   _____ _____ _____ _____ _____
Instr 3:   _____ _____ _____ _____ _____
Instr 4:   _____ _____ _____ _____ _____
Total number of cycles = 8, So CPI = 8 / 4 = 2

A 10-stage pipeline without any stalling:
Instr 1: ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
Instr 2:   ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
Instr 3:   ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
Instr 4:   ___ ___ ___ ___ ___ ___ ___ ___ ___ ___
Total number of cycles = 13, so CPI = 13 /4 = 3.25

Therefore, CPI has increased.

Q2:
Consider the following two ISAs. In both ISAs, Operand 1 is register, Oprand 2 and Oprand 3 can be register or immediate.

(1) The first is a **fixed length** ISA that has the following instruction encoding:

Opcode | Oprand 1 (Destination) | Oprand 2 (Source 1) | Oprand 3 (Source 2)

An opcode is 1 byte. Each operand is 1 byte.

(2) The second is a **variable length** ISA that the following instruction encoding:

Opcode | Oprand 1 (Destination) | Oprand 2 (Source 1) | Oprand 3 (**Optional**, Source 2)

The opcode is 1 byte. Each operand (register/immediate) is 1 byte. Note that Oprand 3 is optional. If an instruction does not need the third oprand, it is not used. In other words, the third oprand is not a part of every instruction.

Consider the following assembly code sequence (please see comment about what each line of code does.)

```
ADD r3, r1, r2 // r3 = r1+r2
SLL r3, 0x2 // r3 = r3 << 2
MOV r5, 0xa // r5 = 0x0a
STW r3, (r5) //MEMORY[r5] = r3
```

What would be the code size for the given assembly language sequence, for each of the two ISAs?

A. Fixed length ISA 18 bytes, variable length ISA 13 bytes
B. Fixed length ISA 16 bytes, variable length ISA 12 bytes
C. Fixed length ISA 16 bytes, variable length ISA 13 bytes
D. Fixed length ISA 18 bytes, variable length ISA 12 bytes

Solution:
Fixed Length ISA -- 4×4 = 16 bytes
Variable Length ISA -- 1 × 4 (ADD instruction) +3 × 3 (other instructions) = 13 bytes
Answer is C