

### Self-Evaluation Exercise on Pipelining (with solution)

Q Reorder the following code to avoid hazards and also reduce execution time, assuming

- A. We use software techniques, and have 2 delay slots.
- B. We use interlocks, and predict not taken.
- C. We use forwarding, and predict not taken.

**add r1, r2, r3**

**sub r4, r1, r1**

**mul r8, r9, r10**

**cmp r8, r9**

**beq .foo**

Hint: for software technique, you can add NOP instruction and for interlock/forwarding techniques, you can insert a bubble if required (a bubble is also represented as a NOP instruction, but it need not go through all the pipeline stages. See this: <https://stackoverflow.com/questions/33071948/what-is-the-difference-using-nop-and-stalls-in-mips>). Further, in all the three: A, B, and C, you can reorder the code.

**A.**

**mul r8, r9, r10**

**add r1, r2, r3**

**nop**

**nop**

**cmp r8, r9**

**beq .foo**

**nop**

**sub r4, r1, r1**

**B.**

```
add r1, r2, r3
```

```
mul r8, r9, r10
```

```
cmp r8, r9
```

```
beq .foo
```

```
sub r4, r1, r1
```

```
C.
```

```
mul r8, r9, r10
```

```
cmp r8, r9
```

```
beq .foo
```

```
add r1, r2, r3
```

```
sub r4, r1, r1
```