Colin Fitzsimons

a.

daddi r1, r0, 10
daddi r1, r1, 7
Causes a read after write stall

dmul r3, r1, r2
dadd r3, r1, r2
Causes a Write after write stall

No instructions can be written that would cause a WAR stall in MIPS but it 'would' look as follows:
dadd r1, r2, r3
sub r2, r4, r1
Where there is an artificial name depends between on the r2.

b.
Pros
+ Less complex
+ Requires a much smaller CPU
+ Is far cheaper to build

Cons
+ Cannot perform complex equations
+ Much slower, as data will be accessed from memory more often

c.
Reasons why we don't have that instruction:

1. It involves a read from memory and write to memory operation.
2. The pipeline isn't set up to handle this type of instruction
3. Not all data would be available by the first time we hit the EX phase (it would require us going back and calculating each individual part of the instruction)
4. Defeats the purpose of RISC

d.
The war stall here is caused by the f2 on the second line which has an artificial dependency on the f2 in the first instruction.
We can fix this by changing things as follows:
div.d f1, f2, f3
add.d f8, f4, f5
mul.d f6, f4, f8