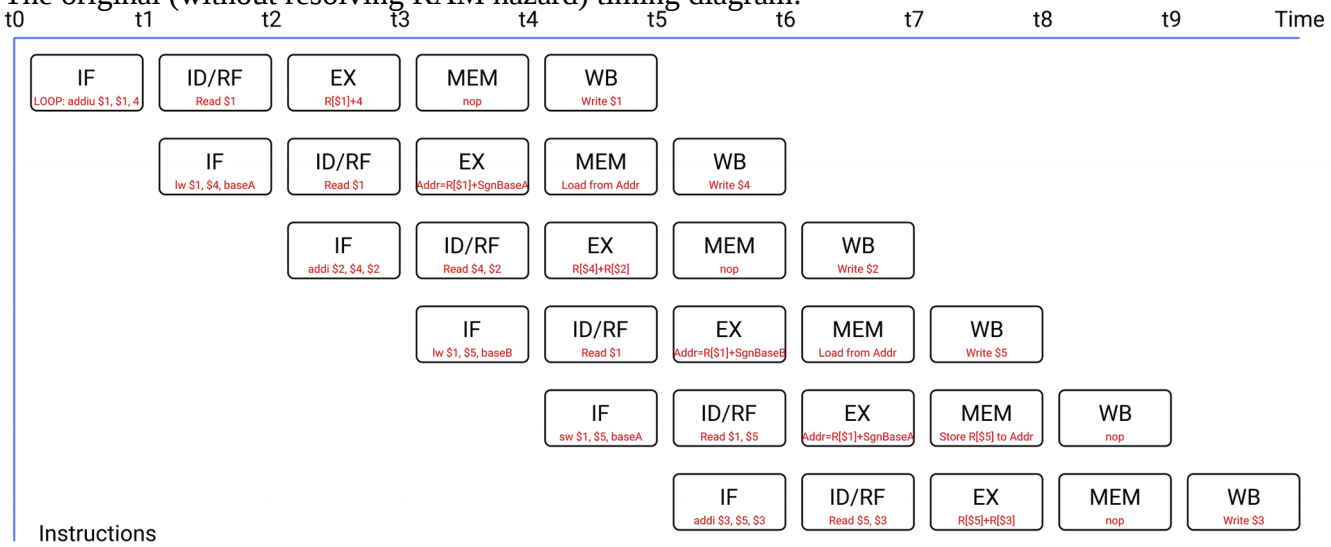
3.1

The original (without resolving RAM hazard) timing diagram:

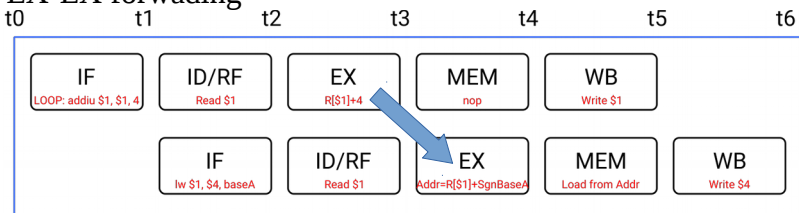


Possible RAM hazards:

1. The Write $1 in the Instruction 1 (LOOP: …) and the Read $1 in the Instruction 2.
2. The Write $4 in the Instruction 2 and the Read $4, $2 in the Instruction 3.
3. The Write $5 in the Instruction 4 and the Read $1, $5 in the Instruction 5.
4. The Write $5 in the Instruction 4 and the Read $5 in the Instruction 6.

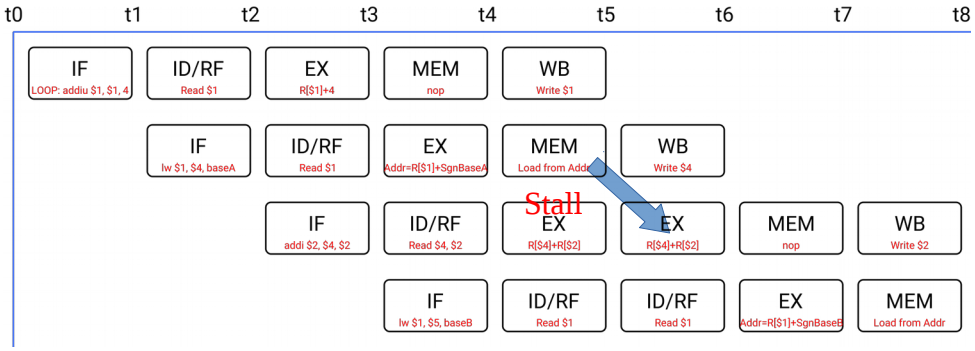
Solution for 1:

EX-EX forwading

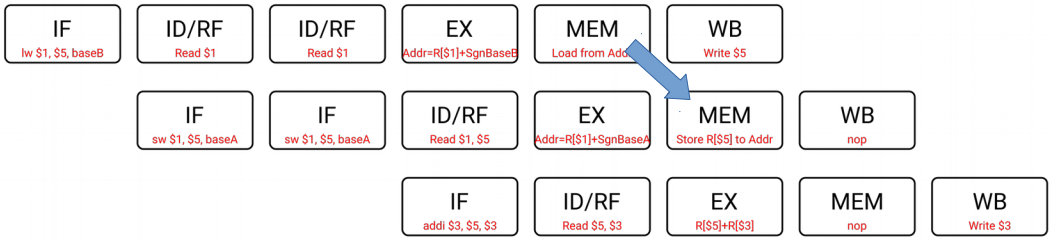


Solution for 2:

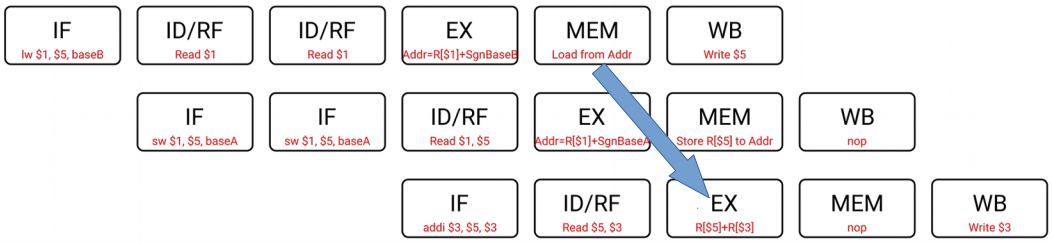
Stall and MEM-EX forwarding



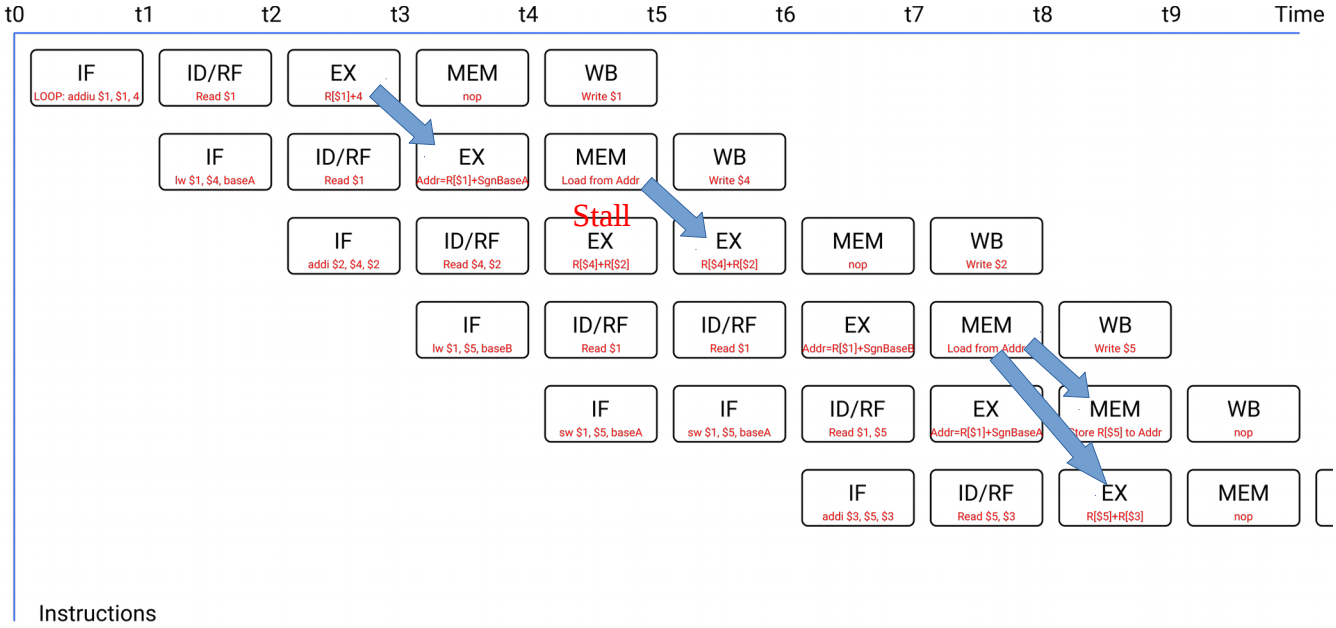
Solution for 3:
MEM-MEM forwarding



Solution for 4:
MEM-EX forwarding

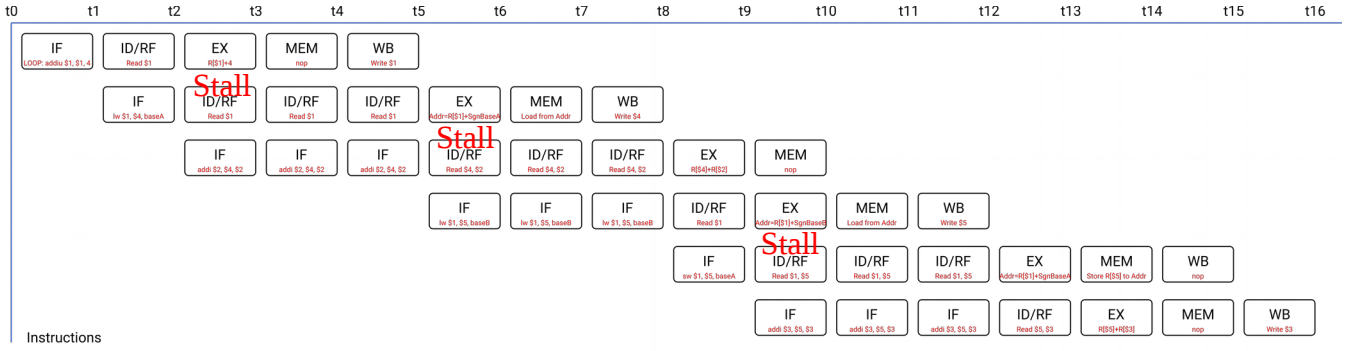


3.2



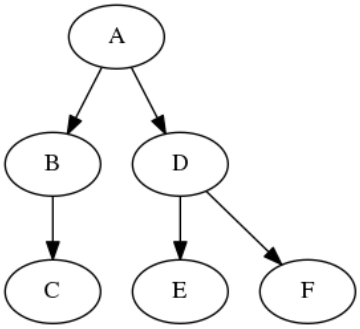
3.3
The total clocking cycles are:
10*10+1=101

3.4



3.5

Name the first six instruction (exclude the bne instruction) A, B, …, F, respectively and since only stalls allowed, store to memory/load from memory will not cause RAW hazard, so we can take each instruction as an abstract only read/write from register is take into consideration.



Also because only stalls allowed, it's impossible that an instruction will and it's previous previous instruction form RAW hazard, only consecutive instructions can form. Thus we use a simple program to fully list all possible arrangement. It’s OK in this only five instructions need to be arrange (A is known to be the first) but more efficient algorithm is need for practical use. The final result is F C is put last and the first three can be B D E, D B E and D E B:

A B D E F C
A D B E F C
A D E B F C