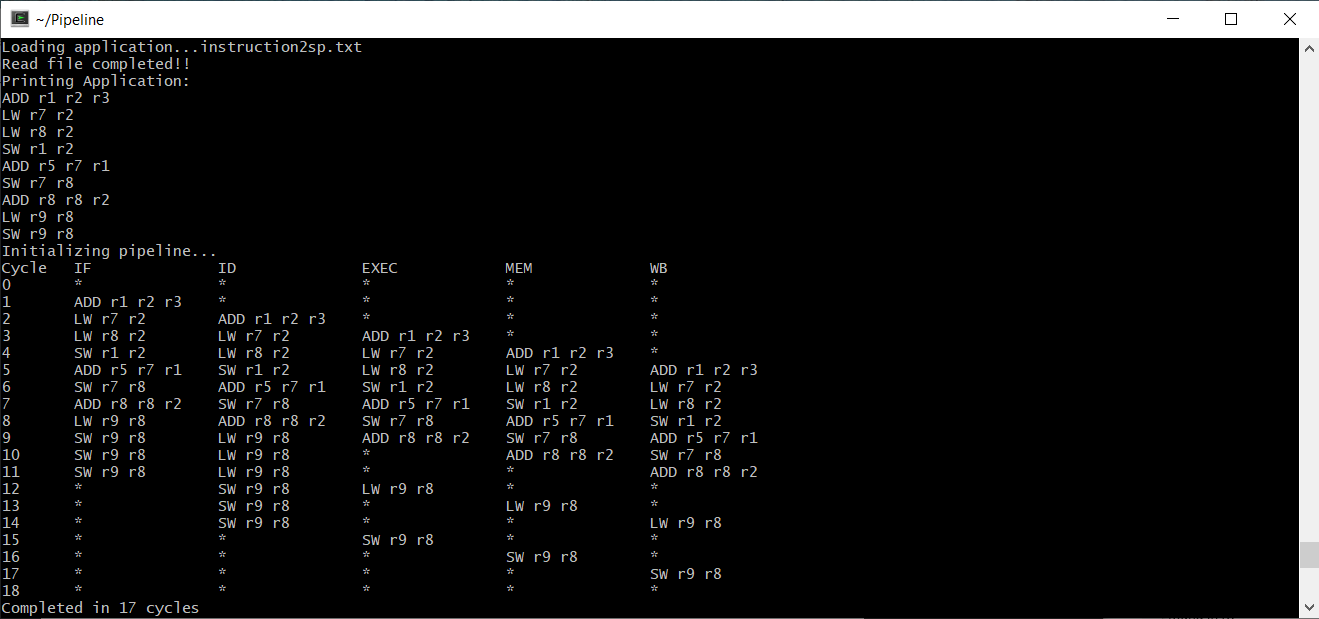
Tanish Arora

Part 4

1. In instruction.txt, there are two types of hazards in this code. The first is Data Hazards which occurs on register r1 on instruction 1 (ADD) for instruction 3. The answer stored in r1 is needed to be used in instruction 3 (MULT). The second occurrence appears on instruction 4 (DIV) for instruction 5 (LW) to store the new value at address r4 to r5. The third situation occurs on instruction 6 (SW) where r5’s value is found from instruction 5 (LW) operation. The second possible problem is the Control Hazard on the BNEZ which either computes the next instruction found at address r8 or the next instruction in line (which is nothing).

In instruction2.txt, there are only Data Hazards in this file. The first appears on instruction 2 (SW) which takes the answer found in r1 of instruction 1 (ADD) and puts it into the address of r2. The seconds is on instruction 4 (ADD) which requires the value of r7 to be loaded from instruction 3, the LW operation. The next hazard occurs on instruction 6 which requires the address value found in r8 of instruction 5 to be used as the storage place for SW operation. Instruction 8 also requires the value of r8 to be stored in the LW operation of instruction 6. The next hazard occurs on instruction 8 which requires the address value found in r8 of instruction 4 to be used as the storage place for LW operation. Finally, the last instruction (SW) needs the previous value found in r9 of instruction 8 to be placed into the address of r8.

1. For instruction.txt, the number of cycles used is 18 cycles with no forwarding and 14 cycles with forwarding. For instruction2.txt, the number of cycles used is 23 cycles with no forwarding and 17 cycles with forwarding. The CPI for instruction.txt (7 instructions) with no forwarding it is 18 / 7 = 2.514 cycles/instruction and with forwarding it is 14 / 7 = 2.000 cycles/instruction. The CPI for instruction2.txt (9 instructions) with no forwarding it is 23 / 9 = 2.556 cycles/instruction and with forwarding it is 17 / 9 = 1.889 cycles/instruction. The speedup for instruction1.txt is 2.514 / 2.000 = 1.26 and the speedup for instruction2.txt is 2.556 / 1.889 = 1.35.
2. This program in instruction2.txt can be improved from 23 cycles to 17 cycles with no forwarding.



The speedup of this new file (instruction2sp.txt) is 23 / 17 = 1.35.

1. No, doubling the frequency does not improve the CPI because frequency only shows how fast instructions get processed. Frequency does not affect how many cycles it takes to do the instruction. CPI can improve by decreasing the total number of cycles each instruction takes. For instance, if a Branch instruction only were to take 3 cycles instead of 5 or an Arithmetic instruction only were to take 4 cycles instead of 5, then the CPI would be improved.
2. Suppose that the speedup (S) is infinite, then by Amdahl’s Law, the maximum possible speedup is the following:
3. By Amdahl’s Law, the maximum possible speedup using parallelization is the following: