CMPE 120 Homework #2

2.1 Consider the operation of a machine with the data path of Fig. 2.2. Suppose that loading the ALU input registers takes 5 nsec, running the ALU takes 10 nsec, and storing the result back in the register scratchpad takes 5 nsec.
What is the maximum number of MIPS this machine is capable of in the absence of pipelining?
<Data path in Fig. 2.2 consists of a register bank and ALU.>

Total data path cycle = 5 nsec + 10 nsec + 5 nsec = 20 nsec
Cycles/second = 1/20 nsec = 50 millions
Best performance = 50 MIPS (Million Instructions Per Second)

∴ Maximum number of MIPS this machine is capable of is 50MIPS in the absence of pipelining

2.2 -

2.3 On computer 1, all instructions take 10 nsec to execute. On computer 2, they all take 5 nsec to execute. Can you say for certain that computer 2 is faster? Discuss.

No. We cannot conclude which computer performs better because computer 1 may divide the 10 ns path into 5 pipeline stages.

For computer 1, MIPS = $\frac{1 \times 10^{9}}{10}$ = 1×10^{8} = 100,000,000 nsec = 100 MIPS but let's assume

computer 1 consists of a pipeline of five stages, it can have MIPS up to 500 MIPS.

For computer 2, MIPS = $\frac{1 \times 10^9}{5}$ = 2×10^8 = 200,000,000 nsec = 200 MIPS

and let's assume computer 2 does not consist of a pipeline, so it can only have a maximum of 200 MIPS.

2.4 Imagine you are designing a single-chip computer for an embedded system. The chip is going to have all its memory on chip and running at the same speed as the CPU with no access penalty. Examine each of the principles discussed in Sec. 2.1.4 and tell whether they are so important (assuming that high performance is still desired).

<The principles discussed in Sec. 2.1.4 (slide 29 chapter 3 Organization) include
--all instructions are directly executed in hardware
--maximize the rate at which instructions are issued
--instructions should be easy to decode
--only loads and stores should reference memory --
Provide plenty of registers>

The first three principles, which are that all instructions are directly executed in hardware, maximize the rate at which instructions are issued and instructions should be easy to decode are **important**. Having memory on chip and running at the same speed as the CPU with no access penalty does not affect the first three principles.

The fourth principle which only loads and stores should reference memory is **not important** in this scenario because the single-chip computer to be designed is going to have all its memory on chip, so memory accessing behaves like register-to-register operation which is extremely fast, thus memory-accessing instructions does not have to be limited to LOAD/STORE instructions.

The fifth principle which is providing plenty of registers is also **not important** in this scenario because all the memory is on the chip. Register is to store data temporarily and having plenty of registers reduces the need to access memory which takes a long time. In

this scenario, since all the memory is on the chip, there are no access speed concerns thus this principle is no longer important.