**Lecture #20 Worksheet, Answer Master**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Classification/section: \_\_\_\_\_\_\_\_\_\_\_\_\_**

**Fill in blanks to answer questions below. Then email this sheet to your TA.**

1. **Although the MIPS multicycle implementation provides a substantial 30-40% speed improvement, it has the disadvantage that the five segments of the processor must be completed to finish an instruction. When one segment is active, what are the other four segments doing?**
2. **Patterson and Hennessy provide the “laundry example” to illustrate how parts of the processor might be simultaneously active. Explain this example.**
3. **Slides 4-6 illustrate the basic idea of the pipeline. Explain in your own words.**
4. **Why does the speed advantage of the pipeline approach s (s the number of stages in the pipeline)?**
5. **State the five stages of the pipeline, which correspond to the five stages of the multicycle implementation.**
6. **Slide 9 shows the overlapping of the instruction cycles. How many instructions are being simultaneously processed when the pipeline is full?**
7. **Going back to the single cycle implementation, we see that once again, there is a need to save partial results as instructions go down the pipeline. How does this compare to the multicycle implementation?**
8. **Slide 11 shows the rudimentary pipeline (no control lines) with storage. What are the acronyms for the four interfaces?**
9. **Slides 12-16 show the progress of an instruction down the pipeline. Study these carefully. Why is the program counter contents carried along?**
10. **Slide 18 shows the pipeline with all control lines added (the jump circuitry, which is fairly simple, is not included for clarity). Questions:** 
    1. **Why are the identities of the registers carried along on the register address bus?**

* 1. **Control signals are also taken down the pipeline. Why?**
  2. **What does the lower MUX in the ALU (EX) cycle do?**

1. **Slides 19-31 demonstrate the transition of five instructions down the pipeline. Study these carefully to understand how instructions are processed. In the multicycle implementation, only those sections of the instructions that are needed are included. Is this true in the pipeline?**
2. **Complete the exercise on slide 32, using the diagram on slide 33. The answers are shown on slide 34.**
3. **Hazards are a problem due to pipeline design that can result in the wrong data being accessed. The two types of hazards are discussed in slides 35-38. Although differently named, control hazards (in branches) and data hazards (as in register instructions) both result from exactly the same pipeline problem. Explain this problem in your own words.**
4. **How does forwarding solve the hazard problem?**
5. **Study slide 40.How does the forwarding unit decide when forwarding is required?**
6. **As slides 42-44 explain, sometimes forwarding will not work by itself. When does this occur and how does the pipeline process address this problem?**
7. **Study slides 45-48. Branching-related problems are due to the fact that in the unaltered pipeline, the branch decision is made in the ALU or EX cycle. With no way of identifying the branch results early, two incorrect instructions may have been loaded into the pipeline. Name three ways that this problem can be addressed.**
8. **What two improvements can be made on these basic methods?**

1. **Answer the questions on slide 49, then check your answers on the last two slides (50-51).**
2. **Study the summary slide on branching. Remember: performance is bought with $$$ and complexity. The MIPS pipeline is a good example of both.**