READING AND RESEARCH ASSIGNMENT

A) [15] [Max 1 page] Read and comment on the paper labelled with R2a. Read pages 1-35 of the paper labeled with R2b and comment on it, particularly ranking in order of importance to ILP the features such as alias analysis, branch prediction, register renaming, and simultaneous speculative execution across different paths; and comment on the weaknesses of this study.

B) [10] In the table below, common parallel I/O buses are shown. Please search on the web or use other resources to fill the required fields with the most up-to-date data.

	SATA	**SCSI**	**PCI-X**	**A choice of yours**
Data Width				
Clock Rate				
Bandwidth				

EXERCISES

1) [10] In a server farm such as that used by Amazon or Google, a single failure does not cause the entire system to crash. Instead, it will reduce the number of requests that can be satisfied at any one time. If a company has 10,000 computers, each with a MTTF of 35 days, and it experiences catastrophic failure only if 1/4 of the computers fail, what is the MTTF for the system?

2) [10] Your company is trying to choose between purchasing the Opteron or Itanium 2. You have analyzed your company's applications, and 50% of the time it will be running applications similar to facerec, 20% of the time applications similar to applu, and 30% of the time applications similar to sixtrack. (See Figure 1.17 in the textbook). If you were choosing just based on overall SPEC performance, which would you choose and why? What is the weighted average of execution time ratios for this mix of applications for the Opteron and Itanium 2?

3) [10] Suppose we have a deeply pipelined processor, for which we implement a branch-target buffer for the conditional branches only. Assume that the misprediction penalty is always four cycles and the buffer miss penalty is always three cycles. Assume a 90% hit rate, 80% accuracy, and 10% branch frequency. How much faster is the processor with the branchtarget buffer versus a processor that has a fixed two-cycle branch penalty? Assume a base clock cycle per instruction (CPI) without branch stalls of 1.

4) [10] Please do the exercise 3.11 from the textbook. Show your work.

5) [15] In a machine M1 clocked at 1GHz it was observed that 10% of the computation time of integer benchmarks is spent in the subroutine Multiply (A, B, C) which multiplies integer A and B and returns the result in C. Furthermore, each invocation of Multiply takes 800 cycles to execute. It is proposed to introduce a new instruction MULT to improve the performance of the machine on integer benchmarks. Please answer the following questions, if you have enough data. If there is not enough data simply answer "not enough data".

   a) How many times is the Multiply routine executed in the set of programs?

   b) An implementation of the MULT instruction is proposed for a new machine M2. MULT executes the multiplication in 40 cycles (which is an improvement over the 800 cycles needed in M1.) Besides the Multiplies, all other instructions, which were not part of the multiply routine in M1, have the same CPI in M1 and M2. Because of the added complexity however, the clock rate of M2 is 900MHz. How much faster (or slower) is M2 over M1?

   c) A faster hardware implementation of the MULT instruction is designed and simulated for a proposed machine M3, also clocked at 900MHz. A speedup of 10% over M1 is observed. Is this possible or is there a bug in the simulator? If it is possible, how many cycles does the MULT instruction take in this new machine? If it is not possible, why is this so?

CASE STUDY

6) [20] In this exercise, we assume that the following MIPS code is executed on a pipelined processor with a 5-stage pipeline (IF, ID, EX, MEM, WB), **full forwarding**, and a **predict-taken branch predictor**.

   Hints:

   First find data dependencies.

   Then examine if these dependencies can cause data hazard.

   Branch instruction can cause control hazards. Instructions will be flushed if the prediction is wrong.

```
            LW R2, 0(R1)
LABEL1:     BEQ R2, R0, LABEL2    #Not taken once, then always taken
            LW R3, 0(R2)
            BEQ R3, R0, LABEL1    #Always taken
            ADD R1, R3, R1
LABEL2:     SW R1, 0(R2)
            ADD R4, R5, R6
```

Draw the pipeline execution diagram for this code, assuming that branches execute in the EX stage.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LW R2, 0(R1)	IF	ID	EX	MEM	WB										
BEQ R2, R0, LABEL2		IF													
