Computer Organization & Assembly Language

Final Exam – 2012/1/13

Dept. of Engineering Science, National Cheng Kung University

1. Answer *True* or *False* to the following statements. (30%)

T (A) A left shift instruction can replace an integer multiply by a power of 2 if no overflow occurs.

T (B) In computer arithmetic, floating-point addition may not be associative; that is, x+(y+z) != (x+y)+z.

T (C) Since computer arithmetic is with limited precision, computation results obtained from computers are usually erroneous and suspicious.

T (D) Underflow in the floating-point representation means that a negative exponent becomes too large to fit in the exponent field.

F (E) The register is considered as a type of combinational elements.

F (F) The CPI value of an ideal pipeline processor is the same as that of a single-cycle processor.

F (G) Designing the control part of a processor is straightforward since it can be easily built on the basis of existing datapath.

F (H) In a pipelined system, forwarding will eliminate the need of any stalls.

T (I) Increasing the depth of pipeline (i.e., number of stages) without balancing the cost of each stage may not improve the overall performance.

T (J) Pipeline bubbles cause loss of performance since hardware utilization is relatively low.

8

1

16進

E0

FFF17

- 001F 15

FE158

1.

FEF8

2. Perform the following operations of computer arithmetic. (24%)

(A) Convert 0xCA83 to binary.

(B) Convert 0xF2A7 to decimal, interpreting it as a signed 16-bit integer.

(C) Subtract 0x001F from 0xFF17, interpreting each as 2's complement 16-bit integer. Express the answer both in hex and decimal.

(D) Assuming single precision IEEE 754 format, what decimal number is represent by this word: 10111110100100000000000000000000 ?

(E) Show the IEEE 754 representation of -3.75_{ten} in single precision.

(F) Show the IEEE 754 representation of $1/6_{ten}$ in single precision.

3. Consider the following sequence of actual outcomes for a single static branch where T means the branch is taken and N means the branch is not taken. Also, assume that this is the only branch in the program.

T N T N T N T T T N N N T T T N T N N N

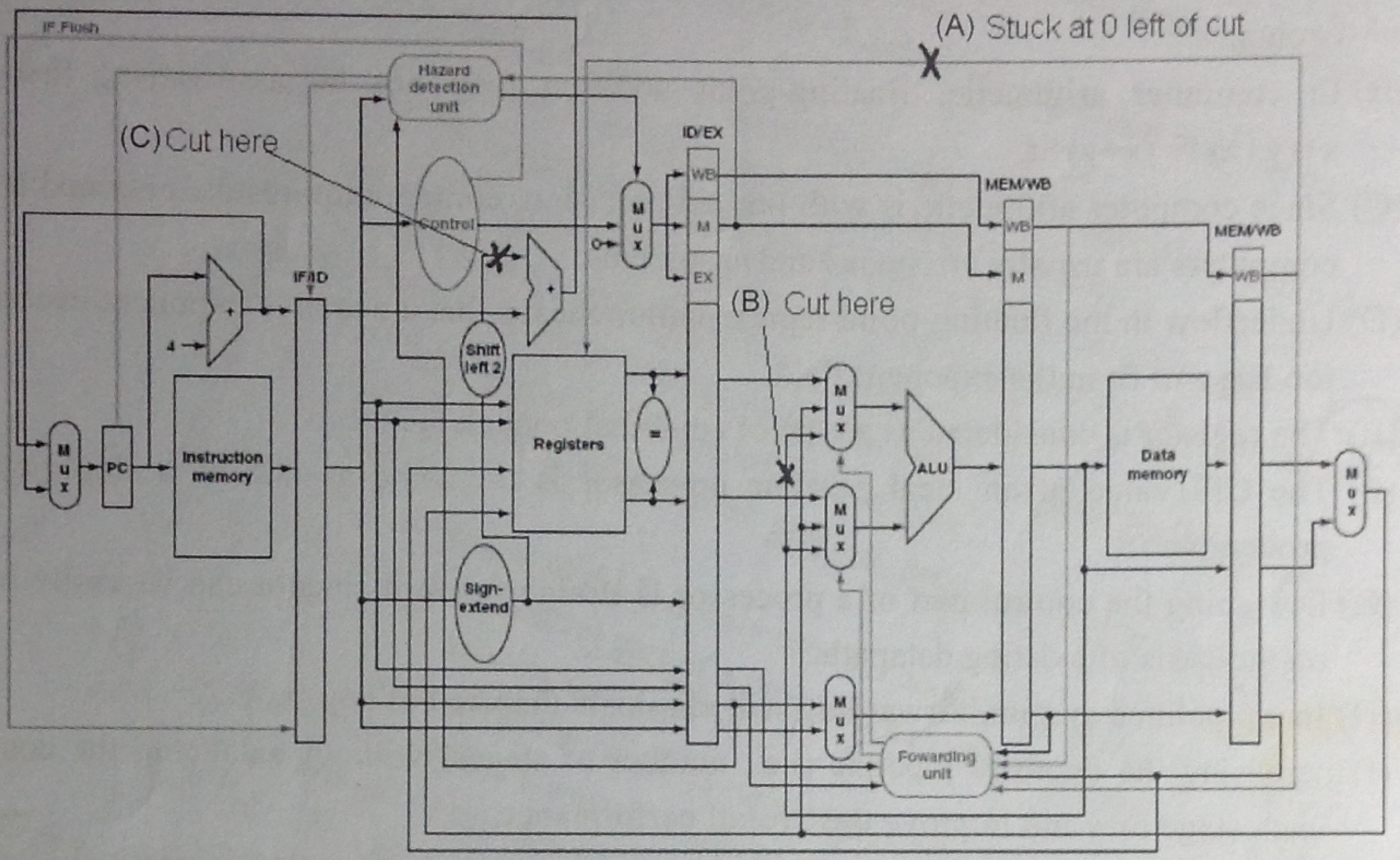
(A) Assume that we use a static predictor which always predicts N. How many branches are mis-predicted? (4%)

(B) Assume that we try to predict this sequence with the 1-bit predictor (which is initialized to the N state). What is the predicted sequence? How many branches are mis-predicted? (4%)

(C) How about using the 2-bit predictor (which is initialized to the T state)? (4%)

(D) Please design another outcome sequence of length 10 so that the 1-bit predictor in (B) works better than the 2-bit predictor in (C). Also, show the corresponding predicted sequences and precision values in percentage. (4%)

4. For the MIPS datapath shown below, several lines are marked with "X". For each one, describe in words the negative consequence of cutting this line relative to the working, unmodified processor. Also, provide a snippet of code that will fail. (18%)



5. Consider a MIPS machine with a 5-stage pipeline with a cycle time of 10ns. Assume that you are executing a program where a fraction *f* of all instructions immediately follow a load upon which they are dependent. Also, there is no other data hazard or control hazard.

(A) With forwarding enabled, what is the total execution time for *N* instructions (in terms of *f*)? (4%)

(B) Consider a scenario where the MEM stage, along with its pipeline registers, needs 12ns. There are now two options: *add another MEM stage so that there are MEM1 and MEM2 stages* or *increase the cycle time to 12ns so that the MEM stage fits within the new cycle time and the number of pipeline stages remain unaffected.* For a program mix with the above characteristics, when is the first option better than the second. Your answer should be based on the value of *f*. (You may ignore the first few cycles to fill the pipeline.) (4%)

(C) Embedded processors have two different memory regions - a faster scratchpad memory and a slower normal memory. Assume that in the 6-stage machine (with MEM1 and MEM2 stages), there is a region of memory that is faster and for which the correct value is obtained at the end of the MEM1 stage itself while the rest of the memory needs both MEM1 and MEM2 stages. For the sake of simplicity, assume that there are two load instructions *lw.fast* and *lw.slow* that indicate which memory region is accessed. If 40% of the fraction *f* mentioned above get their value from the fast memory, how does the answer to the previous question change? (4%)