Very Short Answer:

(1) (2) Which is more effective, dynamic or static branch prediction?

(2) (2) Issuing multiple instructions per cycle puts tremendous pressure on what two parts of the machine?

(3) (2) Predicting the direction of a branch is not enough. What else is necessary?

(4) (4) Flynn said there were 4 catagories of parallelism - what are they?

(5) (2) What is the name of the programming language used for GPU’s?

(6) (3) How many entries are there in a (5,3) Gshare branch predictor? How many bits?

(7) (3) List 3 examples of existing dynamic Branch Prediction strategies in order of (average) increasing effectiveness.

(8) (3) The book lists several things that limit the amount of achievable ILP. List 3 of them.

Short Answers:

(10) (5) What is the primary difference between Scoreboarding and Tomasulo’s algorithm? What hardware feature makes Tomasulo’s work?

(11) (6) Understanding the hardware can influence how you write programs. Give at least 2 examples of how you might write software differently for a heavily pipelined machine verses a non-pipelined one.

(12) (6)What is a predicated instruction? What are the advantages to using predicated instructions? When would you *not* want to use one?

(13) (6) What is the definition of a basic block? Why is there a desire to create larger ones?

(14) (6) There are at least two types of control flow changes that standard dynamic branch predictors have trouble with. There is a technique that works well for one of these types ... name the two types of branches, and the technique used to successfully deal with one of them.

(15) (6) Supporting precise interrupts in machines that allow out of order completion is a challenge. Briefly explain why, and give three different techniques that can be used to provide precise interrupts.

(16) (6) What does SMT stand for? What is SMT trying to accomplish? What is the difference between coarse MT, fine MT, and SMT?

(17) (6) Forwarding is often used to overcome RAW data hazards. Is there a technique that can be used even when forwarding cannot be? If so, what is it and how does it work? If not, why not?

(18) (6) Compare and contrast Superscalar and VLIW. Describe each, and list two advantages and disadvantages of each approach.

(19) (6) Briefly outline how a Vector machine works, and what type of parallelism it is exploiting.

(20) (6) The memory system presents challenges to ILP designers as well. What is it about the memory system that makes it hard for compilers to optimize code, and also for execution units to achieve maximal performance? (This is not a question about technology, it's a higher-level question).

(21) (14) You are given the following code sequence:

```
ADDF      F1,F2,F3
SUBF      F1,F1,F5
MULTF     F5,F6,F7
DIVF      F1,F5,F7
```

Assume there are 8 logical and 16 physical registers. On the left below is the register mapping upon entering the code sequence. Your job is to fill in the mappings after the execution of the DIVF instruction, including what is on the free list. (Assume that during the execution of this code, no registers are released - in other words, the free list will be shorter at the end than at the beginning.)

BEFORE	
Logical	Physical
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	0

AFTER	
Logical	Physical
0	
1	
2	
3	
4	
5	
6	
7	

Free Pool: 1,3,5,7,9,11,13,15

Free Pool:

Now, rewrite the code sequence below using the actual physical register names instead of the logical ones.

```
ADDF   P__,P__,P__
SUBF   P__,P__,P__
MULTF  P__,P__,P__
DIVF   P__,P__,P__
```