

## ECE512 Final Examination Spring 2005

May 10, 2005

Name: \_\_\_\_\_ UIUC ID: \_\_\_\_\_

1. The exam is to last until 4:30pm CDT.
2. You are allowed to use any notes, books, papers, web sites, or other reference material as you desire. No interactions with others allowed.
3. This examination is organized into ten multiple-choice questions and one open-ended question. The multiple-choice questions are based on lectures, class project presentations, as well as class reading material. Each question is concerned with one topic we covered in the course. These multiple-choice questions are randomly selected from the topics we covered this semester.
4. Each multiple-choice question consists of four statements. You are asked to select the statements that are consistent with the lectures and readings. Each answer may be any of the 16 possible combinations of the four choices.
5. Submit your answers in e-mail to w-hwu@uiuc.edu. Use plain text form! For each question, list the statements you agree with. For each statement, you can give a one-sentence explanation of why you agree or disagree with it.
6. I will continue to accept completed exams until 4:35pm CDT.

**Question 1** (10 pts) This question tests your understanding of instruction fetch mechanisms.

1. Assume that each fetch size is always 32 bytes in a VLIW machine. Further assume that the starting position is uniformly distributed at all 8-byte boundaries in the memory space. If we have the instruction cache line size to be 32 bytes with two banks, we can satisfy exactly half of the fetches in one access to the instruction cache.
2. Trace cache reduces the need for wide instruction cache lines and instruction cache banking.
3. In [Huynh, AMD-03], the AMD Athlon uses a trace cache to speedup the fetch and decode of X86 instructions.
4. In [Hinton, Intel-01], the Pentium 4 trace cache design allows an instruction to be present in multiple traces as long as these traces have different head instructions.

**Question 2** (10 pts) This question tests your understanding of the Out-of-order execution core design.

1. The original Tomasulo algorithm in [Tomasulo, IBM R&D-67] uses a reorder buffer to implement precise interrupt.
2. In the P6 design, each store instruction is handled as a single entry in the reservation station.
3. Assume that the release of a reservation station entry consists of three pipeline stages: (1) distribute execution result, (2) identify the entry for release, (3) read entry out of reservation station. Further assume that a load instruction requires **three** execution cycle before it distributes results to its dependent entries in reservation stations. The hit/miss status is determined at the end of the first load execution cycle. The register bypass to an operation is done at the beginning of the cycle when it is shipped to a function unit. Do you agree with the statement that "In order to execute a dependent instruction of a load immediately after the load completes execution, the entry for the dependent instruction must be speculatively identified as the entry for release during the first cycle of the load execution."
4. In a value-capture reservation station design where operand values are stored in the reservation station entries, there is no need to bypass a value to a released reservation station entry before it enters the execution function units in order to achieve back to back execution of dependent operations.

**Question 3** (10 pts) This question tests your understanding of dynamic and static speculative execution techniques.

1. Assume that a processor repairs a mispredicted branch when the branch retires, as in the P6-like processor described in class. When the recovery begins, there may still be useful instruction fetched from the correct path remaining in the reservation station.
2. In general speculation, a data TLB miss caused by a speculative instruction must be handled during its speculative execution. This can increase the TLB misses handled in the program.
3. In sentinel scheduling, an exception detected by implicit checks is recovered by depositing and propagating the pc of the original excepting instruction in the register file.
4. In [August, et al, ISCA-98], the use of R-tags and pR allows the processor to avoid executing speculative instructions that are not direct or indirect consumers of the result that triggered the recovery.

**Question 4** (10 pts) This question tests your understanding of the VLIW and EPIC architectural concepts.

1. In an EQ latency model, one can continue to use the old contents of the destination register of an operation between the time it is issued and the time exposed latency has expired.
2. In [Colwell, ASPLOS-87] the Multiflow machine, the memory system hardware does not enforce any dependence between memory operations executed in the same clock cycle. The compiler must ensure that the memory accesses in the same clock cycle do not hit the same memory bank.
3. In the Itanium ISA, all instructions to be issued in the same cycle must be contained in the same instruction bundle.
4. In the Itanium ISA, the compiler must schedule the instructions to explicitly accommodate their latencies. There is no interlocking in the architecture specification.

**Question 5** (10 pts) This question tests your understanding of trace and superblock scheduling.

1. With trace scheduling as described in [Fisher, Transactions on Computers, July 1981], one can always optimize away the negative effect of compensation code using a pass of classical code optimization after scheduling.
2. In superblock scheduling [Hwu, et. al., JSC-93], the bookkeeping code is inserted before scheduling. Thus, a pass of optimization can be performed before code scheduling to reduce the negative effects of bookkeeping code.
3. In a superblock loop [Hwu, et al., JSC-93], an invariant variable on the most frequently executed path can be moved outside the loop even when a function call with unknown side effects exists on one of the infrequent paths.
4. A list scheduling algorithm completes the processing of each processor cycle before it begins to process any subsequent cycles.

**Question 6** (10 pts) This question tests your understanding of modulo scheduling.

1. In [Rau, et al, MICRO-92], a remainder loop can be generated either before or after a modulo scheduled counter do (for) loop to reduce the number of epilogues for a loop with modulo variable expansion.
2. In the Itanium ISA, rotating registers have the same design as the Cydra-5 rotating registers [Rau, et. al., IEEE-89].
3. If a VLIW processor has unlimited execution resources, modulo scheduling without loop unrolling cannot achieve an execution rate of more than one iteration per clock cycle.
4. In a modulo scheduler that does list scheduling, all instructions are scheduled only once even though they may be replicated several times at the end of the scheduling process. Thus, a modulo scheduler is intrinsically more efficient in terms of scheduling time than an acyclic scheduler that requires unrolling to overlap loop iterations.

**Question 7** (10 pts) This question tests your understanding of predicated execution and data speculation.

1. In the Itanium ISA, one uses the OR-type predicate defines to generate a predicate that guards the execution of a block executed under two or more conditions.
2. In [Schlansker and Park, HP Labs 1991] and the lecture, the if-conversion algorithm assigns the same predicate to all blocks that have the same control dependence set, which is calculated based on dominators.
3. When converting predicated code from fully predicated architecture to a code sequence in a cmov architecture, one cannot allow any store instructions in the original code sequence.
4. In the Itanium ISA, a data speculative load cannot work properly if it accesses a memory-mapped I/O location. The compiler must make sure that the load does not access a memory-mapped location.

**Question 8** (10 pts) This question tests your understanding of DRAM techniques.

1. EDO DRAM is inherently a SDRAM technology that allows the clock to go faster.
2. DDR SDRAM accomplishes high data rate by doubling the clock frequency.
3. Packet DRAM allows multiple streams to be simultaneously accessed, a feature that allows better memory bus utilization.
4. RDRAM is slower than SDRAM and therefore not used by DRAM manufacturers.

**Question 9** (10 pts) This question tests your understanding of DSP and graphics microarchitectures .

1. Saturating arithmetic operations do not have associative property. A compiler cannot reorder saturating arithmetic operations on the same variable without explicit user permission.
2. Bit reversal addressing allows the processor to access a sequence of memory locations in the reverse order of the explicitly generated address sequence.
3. Hardware loops work with branch predictor in a DSP processor to achieve highly efficient loop execution.
4. Data bitwidth of DSP processors must be at least 32 bits in order to preserve signal integrity for speech applications.

**Question 10** (10 pts) This question tests your understanding of class project presentations and future trends.

1. Dynamo is a hardware scheme for run-time optimization that requires a branch behavior table to identify hotspots of execution.
2. It is more important to reuse runahead results execution results in an in-order pipeline than in an out-of-order pipeline.
3. Reducing voltage and clock frequency is an effective way to reduce dynamic switching power consumption.
4. The increasing variability of transistors in future fabrication process makes it more likely for deeper pipelining microarchitectures to achieve high clock frequency.

Open Question. Identify up to three key contributions you made to the ECE411 course this semester.