EE 457 Final Review
Redekopp

1. **Virtual Memory**: Given a virtual memory system with 24-bit virtual addresses, 1 KB pages, and 20-bit physical addresses.

    a. How many address bits will be used for the page offset field? _____

    b. How many bits will each Physical Page Frame Number require? _____

    c. Given an 8-way set associative TLB with 256 entries, perform the virtual address breakdown (i.e. how many address bits will be used for the set field and for the tag field of the address mapping. Show any work.

    Reference Layout of TLB Address Mapping:

Tag	Set	Page offset

    Number of set bits: _____

    Number of tag bits: _____

    d. Given a single level page table, how much memory would be required to hold the table assuming each entry in the table requires **4 bytes** (this includes the page frame, valid, dirty and other bits). [Hint: Size = # of entries * bytes per entry]

    e. Given a two level page table where the 1st level has 32 entries and the 2nd level contains the rest of the needed entries, find a sequence of 3 memory accesses that would require accessing 2nd level page tables 0, 1, and 31 to be allocated.

2. **Cache.** Examine the following sequence of memory accesses.

1. Read 0x0a0 2. Write 0x0b4 3. Read 0x124 4. Write 0x170 5. Read 0x33c 6. Read 0x128 7. Write 0x4ac 8. Read 0x33c 9. Read 0x4b0		Assumptions • 12-bit byte addresses • Word accesses only • Cache Size and Block size = **8 blocks** of **8 words** each • No-Load-Through • Write-back • 2 way Set-Associative Cache

a. Perform the address breakdown for the given cache configuration

b. Now list the block operations the 2-way set associateive cache will perform for each access. Possible block operations are: *Fetch Block XX-YY, Evict Block XX-YY w/o writeback, Evict Block XX-YY w/ Writeback, Final Writeback of Block XX-YY, where XX-YY is the block address range.* Hits do not require any block operation. Hint 1: Each access may required 0-2 block ops.
Hint 2: It will help to keep track of which blocks are in the cache.

1. Read 0x0a0	6. Read 0x128
2. Write 0x0b4	7. Write 0x4ac
3. Read 0x124	8. Read 0x33c
4. Write 0x170	9. Read 0x4b0
5. Read 0x33c	Final Writebacks:

3.) (12 pts.) Given the code below, perform explicit register renaming to solve all WAW, WAR hazards present in the original code. When performing register renaming, use register numbers \$10, \$11, \$12… in that order so that everyone's answer will hopefully be more uniform.

`lw  $5,0($2)`
`add  $6,$4,$5`
`sub  $7,$7,$6`
`lw  $4,0($3)`
`sub  $6,$4,$2`
`add  $3,$7,$2`
	`lw  ___,0(___)`
	`add  ___,___,___`
	`sub  ___,___,___`
	`lw  ___,0(___)`
	`sub  ___,___,___`
	`add  ___,___,___`

4.) (5 pts.) Given the code below, (same as in Question 0, assume the first 'lw' instruction stalls due to a cache miss. Assuming an out-of-order, dynamically scheduled processor (that performs automatic register renaming), which instructions would be allowed to execute (i.e. are independent) and which instructions would need to stall due to the 'lw'.

Code	**Circle the correct answer**
`lw  $5,0($2)`	CACHE MISS
`add  $6,$4,$5`	Stall / Execute
`sub  $7,$7,$6`	Stall / Execute
`lw  $4,0($3)`	Stall / Execute
`sub  $6,$4,$2`	Stall / Execute
`add  $3,$7,$2`	Stall / Execute

5.) (22 pts.) Given the code below, (same as in Question 0, *assuming all functional units are currently stalled* (none of the instructions below can execute/complete) show the state of the register status table after each instruction issues. Then show what source operands will be in each reservation station (operand value or RS name of producer) for each instruction. Use reservation station names (A1, A2, S1, S2, and L1, L2) in both the reservation stations and the register status table ('-' in the table means blank) . Assume the following initial values for each register: R[2] = 0x02, R[3] = 0x03, R[4] = 0x04, R[5] = 0x05, R[6] = 0x06, R[7] = 0x07. Your entries in the status table should either be ('-' = blank or a NAME like L1,L2,A1,...). Your entries in the reservation station should be the 2 operands that the instruction will use (either the VALUE of the register if no one else is writing it, or the NAME of the reservation station.

Code	Register Status Table (After Exec. Of Each Instruction)					
	$2	$3	$4	$5	$6	$7
lw $5,6($2)		-	-		-	-
add $6,$4,$5						
sub $7,$7,$6						
lw $4,5($3)						
sub $6,$4,$2						
add $3,$7,$6