

Mikko Lipasti  
Spring 2002

**ECE/CS 552 : Introduction to Computer Architecture  
FINAL EXAM  
May 12th, 2002**

NAME: \_\_\_\_\_ SOLUTION \_\_\_\_\_

This exam is to be done individually.

Total 6 Questions, 100 points

Show all your work to receive partial credit for incorrect solutions

**1. (15 Points) Integer Multiplication and Division**

- a) (2 points) What is the simple Booth encoding of the two's complement number  $10011101_2$  ? Fill in the table below to encode the number with digits (1,0,-1) in the same manner that the simple Booth multiplier would.
- b) (3 Points) Fill in the table to encode the number using Modified Booth's algorithm. I.e encode with the digits (-2,-1,0,+1,+2)

1	0	0	1	1	1	0	1	2's complement
-1	0	+1	0	0	-1	+1	-1	1-bit Booth's
-2		+2		-1		+1		2-bit Booth's

**Show your work here:**

- c) **(10 points)** Using 4-bit unsigned division, 15 divided by 3 gives a quotient of 5 and a remainder of 0. i.e.  $1111 / 0011 = 0101$  For the restoring and non restoring division algorithms, how many separate additions and subtractions are required?

**Restoring (5 Points)**

Number of additions: 2

Number of subtractions: 4

**Show your work here:**

**Non-Restoring (5 Points)**

Number of additions: 2

Number of subtractions: 2

**Show your work here:**

**2. (20 Points) Floating Point Representation and Arithmetic**

- a) (5 points) Consider FPS-10, a 10-bit low-cost floating-point representation that has a sign bit, a 4-bit biased or excess exponent with a bias of 7, and 5 significand bits. Values are normalized to have an implicit leading '1' to the left of the floating point (just as in IEEE 754). With this standard, represent  $(-0.2)_{10}$ .

Sign	Exponent	Significand
1	0100	10011

**Show your work here:**

$S = 1$

$0.2 = 0.00110011001100\dots$  normalized is  $1.10011$  so exp is  $-3$ , with bias is  $4$  or  $0100$

- b) (5 Points) Fill in the following table to specify the range of values that can be represented with FPS-10 when only normalized numbers are allowed, and also when denormalized numbers are allowed. Specify min/max values in both binary and decimal scientific notation.

Minimum of Range		Maximum of Range		Case
Binary	Decimal	Binary	Decimal	
				Normalized Numbers Only
				Denormalized numbers also

**Show your work here:**

**Smallest normalized:  $\pm 1 \times 2^{-6}$  Largest:  $\pm 1 \times 2^7$  (assume  $e=1111$  is for inf/NaN)**

**Denorm smallest:  $\pm 1 \times 2^{-11}$**

**(10 points)** Given A and B below, compute  $R = A * B$  using the FPS-10 FP standard. Show the sign bit and exponent of the result, and fill in the table below assuming a 1 bit per cycle multicycle multiplier for computing the significand (you may use a Booth multiplier if you wish, but show your work).

$A = 1.11010_2 \times 2^4$  (FP binary representation: 0101111010)

$B = 1.00101_2 \times 2^2$  (FP binary representation: 0100100101)

Sign bit of product: 0 (positive)

Exponent of product: 6

Significand computation:

Multiplicand: 111010

Multiplier: 100101

Step	Explanation	Product Register
1	+111010	<u>0111010</u>
2	+0	<u>00111010</u>
3	001110+111010	<u>100100010</u>
4	+0	<u>0100100010</u>
5	+0	<u>00100100010</u>
6	001001+111010	<u>100001100010</u>
7		
8		
9		
10		

Record the properly normalized and rounded result in binary format below:

Sign	Exponent	Significand
0	7+7=14 or 1110	00010

**Show any additional work here:**

**Product is  $10.0001100010 \times 2^6$ , normalize to  $1.00001100010 \times 2^7$**

**Truncated would be 1.00001**

**Rounded would be 1.00010 since  $r=1, g=0, s=1$**

Mikko Lipasti  
Spring 2002

3. **(20 Points)** Consider a processor with 32-bit virtual addresses, 4KB pages and 36-bit physical addresses. Assume memory is byte-addressable (i.e. the 32-bit VA specifies a byte in memory).
- L1 instruction cache: 64 Kbytes, 128 byte blocks, 4-way set associative, indexed and tagged with virtual address.
  - L1 data cache: 32 Kbytes, 64 byte blocks, 2-way set associative, indexed and tagged with physical address, write-back.
  - 4-way set associative TLB with 128 entries in all. Assume the TLB keeps a dirty bit, a reference bit, and 3 permission bits (read, write, execute) for each entry.
- a) **(10 points)** Specify the number of offset, index, and tag bits for each of these structures in the table below. Also, compute the total size in number of bit cells for each of the tag and data arrays.

Structure	Offset bits	Index bits	Tag bits	Size of tag array	Size of data array
I-cache	7	7	18	$2^9 \times 19$	$2^{16} \times 8$
D-cache	6	8	22	$2^9 \times 24$	$2^{15} \times 8$
TLB	12	5	15	$2^7 \times 21$	$2^7 \times 24$

**Show your work here:**

- b) (5 Points) Explain why accesses to the data cache would take longer than accesses to the instruction cache. Suggest a lower-latency data cache design with the same capacity and describe how the organization of the cache would have to change to achieve the lower latency.

**Have to access TLB in series with Dcache.**

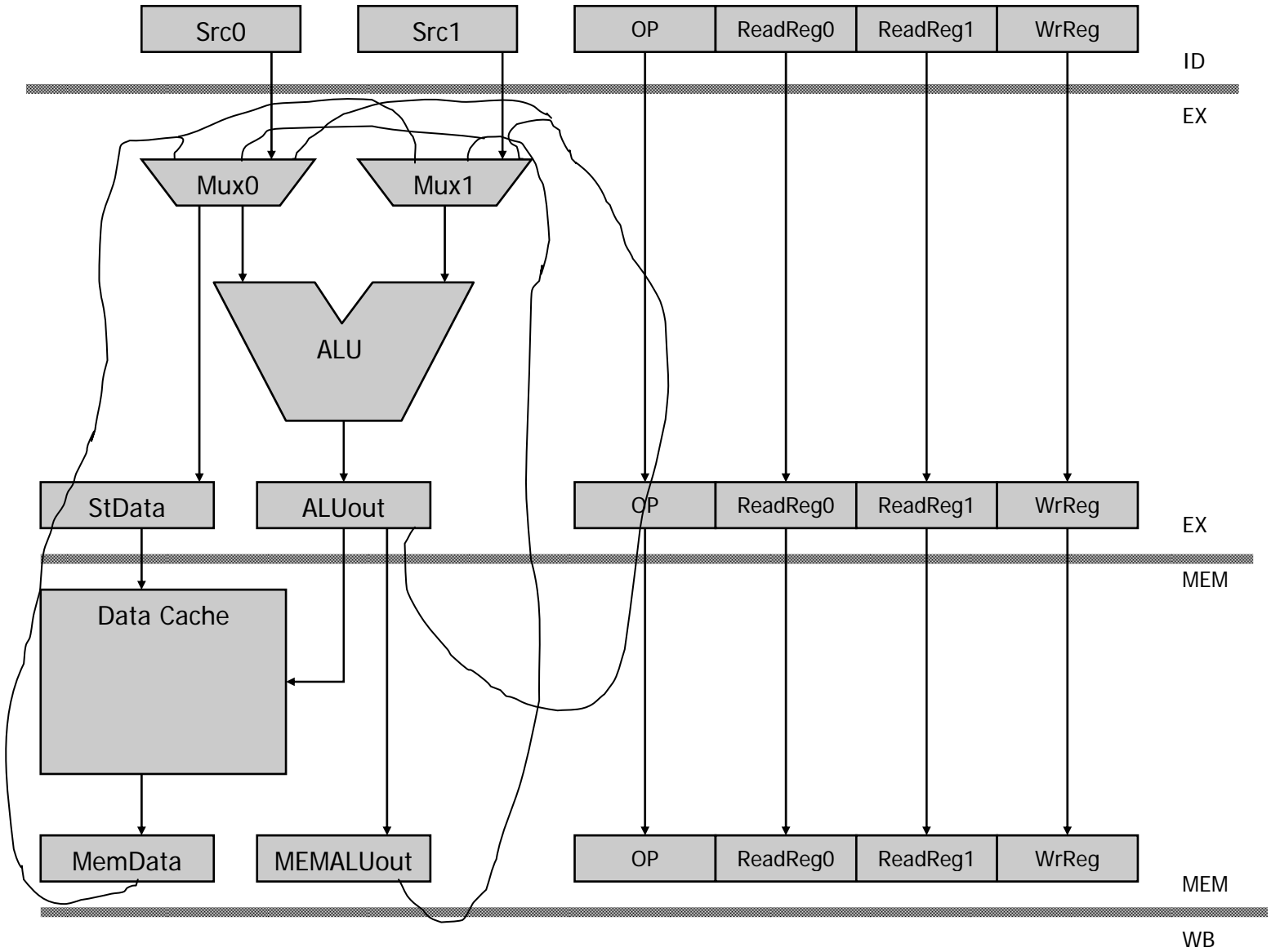
**Could use 32K 8-way set-associative design. Here offset+index bits can all come from the page offset, so they need not be translated. This way, the cache lookup and TLB lookup can proceed in parallel.**

- c) (5 points) Assume the architecture requires writes that modify the instruction text (i.e. self-modifying code) to be reflected immediately if the modified instructions are fetched and executed. Explain why it may be difficult to support this requirement with this instruction cache organization.

**The data cache is physically addressed, while the instruction cache is virtually addressed. One way to keep the I\$ coherent is to snoop all writes against it, but these have to be reverse-translated from physical back to virtual before the I\$ snoop can occur.**

**4. (5 Points) Bypass Network Design**

Given the following ID, EX, MEM, and WB pipeline configuration, draw all necessary Mux0 and Mux1 bypass paths to resolve RAW data hazards. Assume that load instructions are always separated by at least one independent instruction (possibly a NOP) from any instruction that reads the loaded register (hence you never stall due to a RAW hazard).



5. (20 points) Given the forwarding paths in problem 4, draw a detailed design for Mux0 and Mux1 that clearly identifies which bypass paths are selected under which control conditions. Identify each input to each mux by the name of the pipeline latch that it is bypassing from. Specify precisely the Boolean equations that are used to control Mux0 and Mux1. Possible inputs to the Boolean equations are:

- ID.OP, EX.OP, MEM.OP = { 'load', 'store', 'alu', 'other' }
- ID.ReadReg0, ID.ReadReg1 = [0..31,32] where 32 means a register is not read by this instruction
- EX.ReadReg0, etc. as in ID stage
- MEM.ReadReg0, etc. as in ID stage
- ID.WriteReg, EX.WriteReg, MEM.WriteReg = [0..31,33] where 33 means a register is not written by this instruction

Draw Mux0 and Mux1 with labeled inputs; you do not need to show the controls using gates. Simply write out the control equations using symbolic OP comparisons, etc. (e.g. Ctrl1 = (ID.op == 'load') & (ID.WriteReg==MEM.ReadReg0)).

**11: source is MemData: Memop=='load' & (ID. ReadReg0==Mem.WrReg)**

**01: source is MemALU: source is MemALU: Mem.op=='alu' & ""**

**10: source is ALUout: Ex.op=='alu' & (ID.readReg0 == EX.WrReg)**

**Mux1 is identical except use readReg1**

**Logic design is not shown here.**



Mikko Lipasti  
Spring 2002

(20 points) Match the following terms or concepts to the best definition.

Term or Concept	Best Match	ID	Definition
VLIW	t	a	Dynamic multiple arbitration
SIMD	m	b	Caused by inadequate space in a cache
CC-NUMA	j	c	Resolves WAR and WAW hazards in an out-of-order processor
Cold misses	k	d	Broadcasts all cache misses on a bus shared with all other processors
SECDED ECC	w	e	Resolved when a dirty copy of a cache block is found in the cache of another processor
Directory protocol	s	f	Size of this structure is proportion to amount of physical memory instead of size of virtual address space
Address translation	l	g	Requires handshaking to communicate
Reorder or commit buffer	r	h	Used by the operating system to store address translations in a sparse tree structure
Branch predictor	p	i	Only stalls execution of data-dependent instructions vs. all instructions
Seek and rotational	q	j	Machine organization where memory latency varies with the location of memory
Capacity misses	b	k	Caused by a program's first reference to a memory location
DMA	u	l	Enables controlled sharing and protection of physical memory pages
Hashed page table	f	m	Applies the same operation to many data operands at the same time
Snoopy coherence	d	n	Special-purpose cache for virtual address translations
Out-of-order execution core	i	o	Occur when multiple addresses map to the same set
Asynchronous bus	g	p	Resolves control dependences speculatively
Conflict misses	o	q	Delays caused by mechanical components in a storage device
Amdahl's Law	v	r	Enables precise exceptions in an out-of-order processor
Register renaming	c	s	Can maintain coherent caches in a machine with hundreds of CPUs
Coherence misses	e	t	Executes parallel instructions in lock step
		u	Used by input/output device to transfer data
		v	Illustrates the performance bottleneck caused by serial phases of computation
		w	Makes it possible to build reliable memory even when individual storage cells can fail.

Mikko Lipasti  
Spring 2002