

CS356 Unit 15

Review

Final Jeopardy

Binary Brainteasers	Instruction Inquiry	Random Riddles	Memory Madness	Processor Predicaments	Programming Pickles
<u>100</u>	<u>100</u>	<u>100</u>	<u>100</u>	<u>100</u>	<u>100</u>
<u>200</u>	<u>200</u>	<u>200</u>	<u>200</u>	<u>200</u>	<u>200</u>
<u>300</u>	<u>300</u>	<u>300</u>	<u>300</u>	<u>300</u>	<u>300</u>
<u>400</u>	<u>400</u>	<u>400</u>	<u>400</u>	<u>400</u>	<u>400</u>
<u>500</u>	<u>500</u>	<u>500</u>	<u>500</u>	<u>500</u>	<u>500</u>

Binary Brainteaser 100

- Given the binary string “10001101”, what would its decimal equivalent be assuming a 2’s complement representation?

ANSWER: $-128+8+4+1 = -115$

Binary Brainteaser 200

- Assuming the 12-bit IEEE shortened FP format, what is the decimal equivalent of the following number?

1	10010	100010
---	-------	--------

ANSWER: $-1.100010 \cdot 2^3 = -1100.010 = -12.25$ (excess 15)

Binary Brainteaser 300

- Under what conditions does overflow occur in signed arithmetic (addition/subtraction)?

ANSWER: when $p+p=n$ or $n+n=p$

Binary Brainteaser 400

- The following C expression is equivalent to what arithmetic expression?

$$(x \ll 3) + (x \ll 1) + \sim y + 1$$

ANSWER: $8x + 2x - y = 10x - y$

Binary Brainteaser 500

- Given the following normalized FP number, what would the result be after using the round-to-nearest method?

$$+1.011011 \mathbf{100} * 2^5$$

ANSWER: Round to 0 in the LSB, so round up to $+1.011100 * 2^5$

Instruction Inquiry 100

- Initial conditions:
 - `%ebx = 0xf0000001`
 - `%rdi = 0x10010040`
 - `M[0x10010044] = 0xabcdef98`
 - `M[0x10010040] = 0x12345678`
 - `M[0x1001003c] = 0x11122233`
- What is the result of the following instruction?
 - `movb 5(%rdi), %bl`

ANSWER: `0xf00000ef`

Instruction Inquiry 200

- Initial conditions:
 - `%rbx = 0xffff ffff ffff ffff`
 - `%rdi = 0x10010040`
 - `%eax = 0x12345678`
 - `M[0x10010044] = 0xabcdef34`
 - `M[0x10010040] = 0x12345678`
 - `M[0x1001003c] = 0x11122288`
- What is the result of the following instruction?
 - `movsbw (%rdi,%rbx,4),%ax`

ANSWER: `0x1234ff88`

Instruction Inquiry 300

- Initial conditions:
 - `%ebx = 0xf000000f`
- What is the result of the following instruction?
 - `xorl %ebx,%ebx`

ANSWER: `0x00000000`

Instruction Inquiry 400

- Initial conditions:
 - `%eax = 0x 8001 0000`
- What is the result of the following instruction?
 - `sarl $1,%eax`

ANSWER: `0x c000 8000`

Instruction Inquiry 500

- Initial conditions:
 - `%rbx = 0x00000001`
 - `%rdi = 0x1001003c`
 - `M[0x10010044] = 0xabcdef98`
 - `M[0x10010040] = 0x12345678`
 - `M[0x1001003c] = 0x11122233`
- What is the result of the following instruction?
 - `leal 6(%rdi,%rbx,2), %eax`

ANSWER: `0x10010044`

Random Riddles 100

- True/False: The symbol table in an object file has entries for local variables, non-static global variables, and non-static functions?

ANSWER: False (local variables are not tracked...the other 2 are)

Random Riddles 200

- What advantage(s) do shared (dynamically linked) libraries have compared to statically linked libraries?

ANSWER:

- Does not waste memory with multiple copies of the code
- Allows for updated library code to be used without recompilation

Random Riddles 300

- Name at least three possible placement algorithms that may be used by a memory allocator?

ANSWER:

- Best fit
- First Fit
- Next Fit
- optional: Buddy System

Random Riddles 400

- What is placed in the .bss section and why is the .bss section used in an object file or executable?

ANSWER:

- Uninitialized global variables or 0-initialized globals
- Saves space in the executable/object file

Optional: Random Riddles 500

- When seeking to improve the performance of a program, focus should be given to the _____ case which can be found through the help of a software tool called a _____.

ANSWER

- common
- profiler

Memory Madness 100

- True/False: SDRAM will read/write one word at a time to/from the processor

ANSWER: False...Read/write bursts of words

Memory Madness 200

- In a 4-way set associative cache with 512 total blocks, how many bits will be used to index the set (i.e., the set field of the address breakdown)?

ANSWER: $512/4 = 128$ sets \Rightarrow 7-bits

Memory Madness 300

- A 1-way set associative cache could equivalently be called what?

ANSWER: 1-way means only 1 option for each set which is equivalent to a direct mapped cache

Memory Madness 400

- The page table is located in the **(TLB / memory)** and has entries for **(all pages residing in physical memory / all pages)?**

Answer:

- memory
- all pages

Memory Madness 500

- Assume a 24-bit virtual addresses, 1 kB pages and a fully-associative TLB with 128 entries. Assume page table and TLB entries are 2-bytes. How large would the page table be?

ANSWER:

1 kB pages => 10-bits for page offset leaving 14-bits for virtual page number.

This implies $2^{14}=16k$ pages and thus entries in page table.

At 2-bytes each, this would require 32 kB of memory.

Processor Predicaments 100

- A superscalar processor means that the maximum IPC (instructions per clock cycle) is greater than _____?

ANSWER: > 1 instruction per clock cycle

Processor Predicaments 200

- A control hazard occurs when we execute what kind of instruction(s)?

ANSWER: jumps, calls

Processor Predicaments 300

- Of the three kinds of data hazards (RAW, WAR, WAW) which is the only true dependency?

ANSWER: RAW

Processor Predicaments 400

- WAR and WAW hazards prevent us from **(reordering instructions / predicting a branch)** and can be solved through _____?

ANSWER:

- reordering instructions
- register renaming

Processor Predicaments 500

- *Statically* schedule superscalars rely on _____ to schedule the code to avoid hazards, while *dynamically* scheduled superscalars rely on _____ to schedule the code.

ANSWERS: Compiler, HW

Programming Pickles 100

- A programming technique to expose more parallelism in a loop body to the compiler is known as: _____

ANSWER: Loop unrolling

Programming Pickles 200

- Calling a subroutine will result in the return address being stored **(in the PC / on the stack)?**

ANSWER: on the stack

Programming Pickles 300

- The stack frame of a subroutine includes space for three sections of data, what are they?

ANSWER:

- Local variables
- Saved registers
- Arguments for subroutines

Optional: Programming Pickles 400

- The compiler optimization of reproducing the function code at each location where it is called is known as _____

ANSWER: Inlining

Programming Pickles 500

- A special value placed on the stack between local variables and return address is known as a _____

ANSWER: stack canary

Cache Operation Example

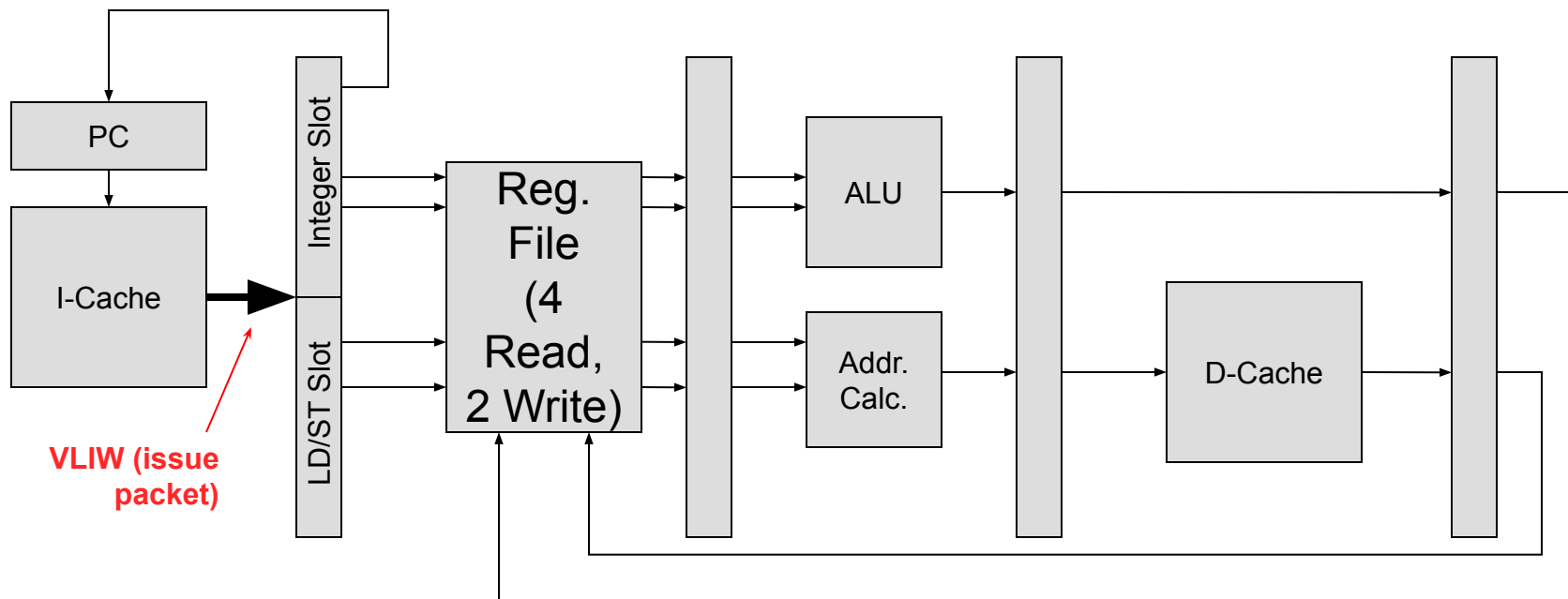
- Address Trace
 - R: 0x3c0
 - W: 0x048
 - R: 0x3d4
 - W: 0xb50
- Operations
 - Hit
 - Fetch block XX
 - Evict block XX (w/ or w/o WB)
 - Final WB of block XX)
- Perform address breakdown and apply address trace
- 2-Way Set-Assoc, N=8, B=32 bytes

Address	Tag	Set	Byte Offset
0x3c0	0011 1	10	00000
0x048	0000 0	10	01000
0x3d4	0011 1	10	10100
0xb50	1011 0	10	10000

Processor Access	Cache Operation
R: 0x3c0	Fetch Block 3c0-3df
W: 0x048	Fetch Block 040-05f
R: 0x3d4	Hit
W: 0xb50	Evict 040-05f w/ WB, Fetch b40-b5f
Done!	Final WB of b40-b5f

2-way VLIW Scheduling

- No forwarding w/in an issue packet (between instructions in a packet)
- Full forwarding paths for instructions already in the pipeline even across slots/pipes (i.e. from 'add' in MEM stage to 'lw' in EX stage)
- Latency of LW is still 1 stall cycle for dependent instructions
- Assume early branch detection (in DECODE stage)



Sample Scheduling

- Schedule the following loop body on our 2-way static issue machine
 - You can modify code and re-arrange but not unroll loops or rename registers

```

for(i=MAX-1; i != 0; i--,A++,B++)
    *A = *A + *B;
```

```

%rdi = pointer to A
%rsi = pointer to B
%edx = i = # of iterations
```

```

L1: ld    (%rdi),%eax
     ld    (%rsi),%ebx
     addl %ebx,%eax
     st    %eax,(%rdi)
     addl $4,%rdi
     addl $4,%rsi
     addl $-1,%edx
     jne  $0,%edx,L1
```

Int./Branch Slot	LD/ST Slot
addl \$-1,%edx	ld (%rdi),%eax
addl \$4,%rdi	ld (%rsi),%ebx
addl \$4,%rsi	
addl %ebx,%eax	
jne \$0,%edx,L1	st %eax,-4(%rdi)

Sample Scheduling

- Now unroll the loop two ways and use register renaming and schedule the code (feel free to modify aspects of the code as needed to ensure better scheduling).

```
%rdi = pointer to A
%rsi = pointer to B
%edx = i = # of iterations
```

```
L1: ld    (%rdi),%eax
     ld    (%rsi),%ebx
     addl %ebx,%eax
     st    %eax,(%rdi)
     ld    4(%rdi),%r8d
     ld    4(%rsi),%r9d
     addl %r9d,%r8d
     st    %r8d,4(%rdi)
     addl $8,%rdi
     addl $8,%rsi
     addl $-2,%edx
     jne  $0,%edx,L1
```

Int./Branch Slot	LD/ST Slot
addl \$-2,%edx	ld (%rdi),%eax
addl \$8,%rdi	ld (%rsi),%ebx
addl \$8,%rsi	ld -4(%rdi),%r8d
addl %ebx,%eax	ld -4(%rsi),%r9d
	st %eax,-8(%rdi)
addl %r9d,%r8d	
jne \$0,%edx,L1	st %r8d,-4(%rdi)