The University of Alabama in Huntsville
Electrical & Computer Engineering Department
CPE 431 01
Final Exam Solution
Fall 2009

1. (1 point) GPUS were developed for the _gaming_ industry.
2. (1 point) _Amdahl's_ law deals with the amount of speedup possible.
3. (1 point) Block is to cache as _page_ is to virtual memory.
4. (1 point) MIPS uses the _callee_ save convention.
5. (1 point) _True_(True or False) The computing industry has committed itself to multicores.

6. (10 points) The following table shows the instruction type breakdown of a given application executed.

FP instructions	INT instructions	L/S instructions	Branch Instructions	CPI (FP)	CPI (INT)	CPI (L/S)	CPI (Branch)
560×10^6	2000×10^6	1280×10^6	256×10^6	1	1	4	2

Assume that the processor has a 2 GHz clock rate. By how much must we improve the CPI of FP instructions if we want the program to run two times faster?

CT = 0.5 ns

$ET = (IC_{FP}*CPI_{FP} + IC_{INT}*CPI_{INT} + IC_{L/S}* CPI_{L/S} + IC_{Branch}*CPI_{Branch})*CT$

$= (500 \times 10^6*1 + 2000 \times 10^6*1 + 1280 \times 10^6*4 + 256 \times 10^6*2)*0.5 \times 10^{-9}$

$= (500 + 2000 + 5120 + 512) \times 10^6 * 0.5 \times 10^{-9}$

$= 8132*0.5 \times 10^{-3}$

$= 4.066$ s

$$2 = \frac{P_{new}}{P_{old}} = \frac{ET_{old}}{ET_{new}} = \frac{4.066s}{(IC_{FP} * CPI_{FP} + IC_{INT} * CPI_{INT} + IC_{L/S} * CPI_{L/S} + IC_{Branch} * CPI_{Branch}) * CT}$$

$(IC_{FP}*CPI_{FP} + IC_{INT}*CPI_{INT} + IC_{L/S}* CPI_{L/S} + IC_{Branch}*CPI_{Branch})*CT = 4.066/2$

$(500 \times 10^6*x + 2000 \times 10^6*1 + 1280 \times 10^6*4 + 256 \times 10^6*2)*0.5 \times 10^{-9} = 2.033$

$500 \times 10^6*x + 7632 \times 10^6 = 4.066 \times 10^9$

$500 \times 10^6*x = -3566 \times 10^6$

$x = -7.13$

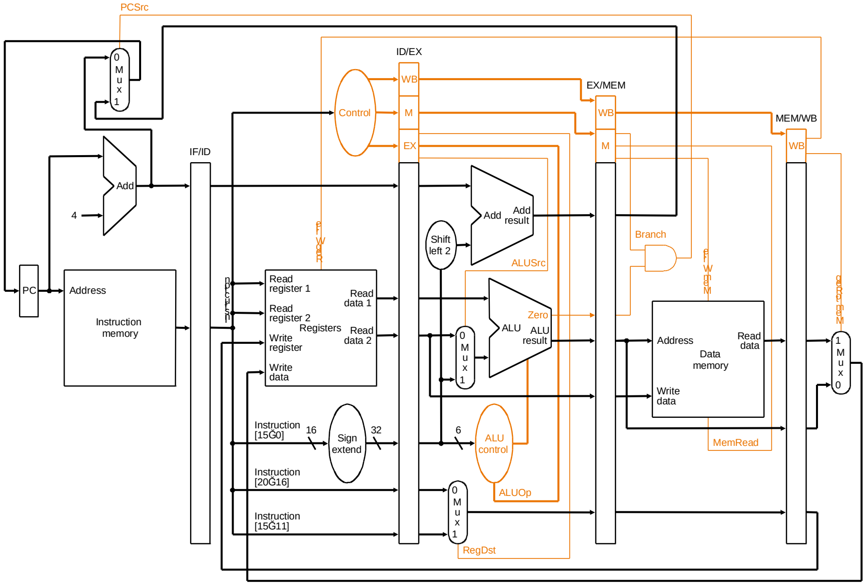
Therefore, it is not possible to increase the performance by a factor of 2 by improving only FP instructions.

7. (10 points) For the following C statement what is the corresponding MIPS assembly code? Assume that the variables `f`, `g`, `h` , `i` and `j` are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Further assume that the base addresses of the arrays A and B are in registers `$s6` and `$s7`, respectively and that the declaration of the arrays is `char A[20], B[20];`

f = A[B[g] + 1];

```
add   $t0, $s7, $s1
lw    $t0, 0($t0)
addi  $t0, $t0, 1
add   $t0, $t0, $s6
lw    $s0, 0($t0)
```

8. (10 points) Consider executing the following code on a pipelined datapath like the one shown which has no forwarding but does support jump. A jump instruction completes in the ID stage.



```
sort:     addi $sp, $sp, -20
          sw   $ra, 16($sp)
          sw   $s3, 12($sp)
          sw   $s2, 8($sp)
          sw   $s1, 4($sp)
          sw   $s0, 0($sp)
          add  $s2, $a0, $zero
          add  $s3, $a1, $zero
          add  $s0, $zero, $zero
for1tst:  slt  $t0, $s0, $s3
          beq  $t0, $zero, exit1
          addi $s1, $s0, -1
for2tst:  slt  $t0, $s1, $zero
          bne  $t0, $zero, exit2
          add  $t1, $s1, $s1
          add  $t1, $t1, $t1
          add  $t2, $s2, $t1
          lw   $t3, 0($t2)
          lw     $t4, 4($t2)
          slt    $t0, $t4, $t3
          beq    $t0, $zero, exit2
          add    $a0, $s2, $zero
          add    $a1, $s1, $zero
          jal    swap
          addi   $s1, $s1, -1
          j      for2tst
exit2:    addi   $s0, $s0, 1
          j      for1tst
exit1:    lw     $s0, 0($sp)
          lw     $s1, 4($sp)
          lw     $s2, 8($sp)
          lw     $s3, 12($sp)
          lw     $ra, 16($sp)
          addi   $sp, $sp, 20
          jr     $ra
```

If the `addi $s1` instruction one instruction before the `for2tst` label begins executing in cycle 1 and the `bne $t0, $zero, exit2` is taken, what are the instructions in each stage of the pipeline in the 14th cycle? If there is a bubble in any stage of the pipeline, also indicate which instruction was there before it became a bubble.

IF: __slt $t0______________

ID: __bubble (lw $s0)_____

EX: __j for1tst____________

MEM: __addi $s0_____________

WB __bubble (lw $t3)_____

Cycle	IF	ID	EX	MEM	WB
1	addi$s1				
2	slt $t0	addi$s1			
3	bne $t0	slt $t0	addi$s1		
4	bne $t0	slt $t0	bubble	addi$s1	
5	bne $t0	slt $t0	bubble	bubble	addi$s1
6	add $t1	bne $t0	slt $t0	bubble	bubble
7	add $t1	bne $t0	bubble	slt $t0	bubble
8	add $t1	bne $t0	bubble	bubble	slt $t0
9	add $t2	add $t1	bne $t0	bubble	bubble
10	lw $t3	add $t2	add $t1	bne $t0	bubble
11	addi $s0	bubble	bubble	bubble	bne $t0
12	j for1tst	addi $s0	bubble	bubble	bubble
13	lw $s0	j for1tst	addi $s0	bubble	bubble
14	slt $t0	bubble	j for1tst	addi $s0	bubble

9. (10 points) Stall cycles due to mispredicted branches increase the CPI. Assume the breakdown of dynamic instructions into various categories and that the branch predictor accuracies are as follows:

R-type	beq	jmp	lw	sw
50 %	15 %	10 %	15 %	10%

Always-taken	Always not-taken	2-bit
50 %	15 %	10 %

Further assume that branch outcomes are determined in the EX stage, that there are no data hazards, and that no delay slots are used. What is the extra CPI due to mispredicted branches with the always-taken predictor?

If a branch is decided in the EX stage, two instructions have been fetched before the decision is made and may be flushed when the prediction is incorrect. The misprediction penalty is two cycles.

Extra CPI = %beq*%wrong*penalty of being wrong = 0.15*0.5*2 = 0.15

10. (5 points) What is the minimum number of bits needed to represent -287 in 2’s complement? What is that representation in hexadecimal?

8 bits: $-2^7 - 2^7 - 1$ =-128 - +127
9 bits: $-2^8 - 2^8$ = -256 - +255
10 bits: $-2^9 - 2^9$ = -512 - +511

10 bits are required

$-287 = -512 + 128 + 64 + 32 + 1 = 10\_1110\_0001_2 = 2E1_{16}$

11. (10 points) Here is a series of address references given as word addresses: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6. Assuming a two-way set associative cache with one word blocks and a total size of 8 words that is initially empty, (a) label each reference in the list as a hit or a miss and (b) show the entire history of the cache

$8words\times\frac{1block}{1word}\times\frac{1set}{2blocks}=4sets$, index = 2 bits, block offset = 0 bits, byte offset = 0 bits

```
          Tag       Index
1       0000 0000 00 01       miss
4       0000 0000 01 00       miss
8       0000 0000 10 00       miss
5       0000 0000 01 01       miss
20      0000 0001 01 00       miss
17      0000 0001 00 01       miss
19      0000 0001 00 11       miss
56      0000 0011 10 00       miss
9       0000 0000 10 01       miss
11      0000 0000 10 11       miss
4       0000 0000 01 00       miss
43      0000 0010 10 11       miss
5       0000 0000 01 01       miss
6       0000 0000 01 10       miss
```

Index	Tag	Data	Tag	Data
0	0x00..01, 0x00..11, 0x00..01	M[4], M[20], M[4]	0x00..02, 0x00..32	M[8], M[56]
1	0x00..00, 0x00..01, 0x00..01	M[1], M[17], M[5]	0x00..01, 0x00..02	M[5], M[9]
2	0x00..01	M[6]		
3	0x00..10	M[19], M[43]	0x00..02	M[11]

12. (5 points) What MIPS instruction does the following collection of bits represent?
0x00020A82

000000 00000 00010 00001 01010 000010
Opcode = 0, R-type
rs = 0, `zero`
rt = 2, `$v0`
rd = 1, `$at`
shamt = 10
funct = 2, shift right logical

```
srl $at, $v0, 10
```

13. (10 points) Assume you are configuring a Sun Fire x4150 server and assume that this configuration contains four processors. Determine whether configurations of 4, 8, and 16 disks present an I/O bottleneck.

Program Instructions Per I/O Operation	OS Instructions Per I/O Operation	Workload (KB reads)	Processor Speed (Instructions/Second)
500,000	100,000	32	1 Billion

The seek time for the disks is 2.9 ms. Consider sequential reads and writes. The disks rotate at 15,000 RPM and have a sustained transfer rate of 112 MB/s. Assume that the bandwidth of all other elements is sufficient to sustain the I/O rate of the processors and the disks.

Maximum I/O rate of 1 processor $= \frac{1\times 10^9\, instructions/s}{(100+500)\times 10^3\, instructions/I/O} = 1{,}667\frac{I/O}{s}$

4 processors can do 6,667,
Time per I/O at disk = transfer time = 32 KB/112MB/s = 0.28 ms

I/O rate of 1 disk = $\frac{1}{0.28ms/I/O} = 3500\frac{I/O}{s}$

	4 Disks	8 Disks	16 Disks
	14,000 I/O/s	28,000 I/O/s	56,000 I/O/s
Bottleneck?	No	No	No

The bandwidth of all the disk configurations is greater than all of the other part of the system listed above, so the disks are never the bottleneck.

14. (10 points) Consider the following portions of two different programs running at the same time on three processors in a symmetric multicore processor (SMP). Assume that before this code is run, w is 0, x is 3 and y and z are 1. w, x, y, and z are type `int`.

Core 1: `y = 5/z;`
Core 2: `w = x + y + 1;`
Core 3: `z = w*x + y;`

What are all the possible resulting values of w, x, y, and z? Show all possible interleavings of instructions and the resulting values of w, x, y, and z.

	w	x	y	z
1,2,3	5	3	5	20
1,3,2	9	3	5	5
2,1,3	5	3	5	20
2,3,1	5	3	0	16
3,1,2	7	3	5	1
3,2,1	5	3	5	1

15. (5 points) Represent 0.0 in single precision floating-point format.

0x0000 0000

16. (10 points) There are several parameters that have an impact on the overall size of a page table. Listed below are several key page table parameters.

Virtual address size	Page size	Page table entry size
32 bits	4 KB	4 bytes

Calculate the total page table size for a system running five applications that utilize half of the memory available.

Page offset = 12 bits, Number of pages = 2^{20},

Page table size = number of applications * number of pages * page table entry size * utilization

= 5 * 2^{20} * 4 bytes * 0.5 = 5 * 2 MB = 10 MB