

**The University of Alabama in Huntsville**  
**Electrical & Computer Engineering Department**  
**CPE 431 01**  
**Final Exam Solution**  
**Fall 2013**

1. (1 point) A **\_track\_** is one of thousands of concentric circles that make up the surface of a magnetic disk.
2. (1 point) A **\_handshaking protocol\_** is a series of steps used to coordinate asynchronous bus transfers in which the sender and receiver proceed to the next step only when both parties agree that the current step has been completed.
3. (1 point) **\_Polling\_** is the process of periodically checking the status of an I/O device to determine the need to service the device.
4. (1 point) **\_Striping\_** is the allocation of logically sequential blocks to separate disks to allow higher performance than a single disk can deliver.
5. (1 point) **\_Hot swapping\_** is replacing a hardware component while the system is running.
6. (5 points) What is the value of the following 32 bits if they represent a single precision floating point number?

0xC850 B300

**1100 1000 0101 0000 1011 0011 0000 0000**

**S = 1, number is negative**

**Exponent = 1001 0000 = 144**

**Fraction = 1010 0001 0110 0110 0000 000**

**Bias = 127**

$$\begin{aligned}
 \text{Number} &= (-1)^S \times (1 + \text{Fraction}) \times 2^{\text{Exponent} - \text{Bias}} \\
 &= (-1)^1 \times (2^{23} + 2^{22} + 2^{20} + 2^{15} + 2^{13} + 2^{12} + 2^9 + 2^8) / 2^{23} \times 2^{144-127} \\
 &= - (2^{23} + 2^{22} + 2^{20} + 2^{15} + 2^{13} + 2^{12} + 2^9 + 2^8) / 2^{23} \times 2^{17} \\
 &= - (2^{23} + 2^{22} + 2^{20} + 2^{15} + 2^{13} + 2^{12} + 2^9 + 2^8) / 2^6 \\
 &= - 213,708
 \end{aligned}$$

7. (10 points) Here is a series of address references given as word addresses: 1, 4, 8, 5, 20, 17, 19, 56, 209, 11, 4, 43, 5, 36, 8, 16. Assuming a two-way set associative cache with two word blocks, a total size of 16 words that is initially empty, and an LRU replacement policy (a) label each reference in the list as a hit or a miss and (b) show the entire history of the cache

$$16words \times \frac{1block}{2words} \times \frac{1set}{2blocks} = 4sets$$

Index BO Hit/Miss

```

1    000_0000_0000 0 00 1 m
4    000_0000_0000 0 10 0 m
8    000_0000_0000 1 00 0 m
5    000_0000_0000 0 10 1 h
20   000_0000_0001 0 10 0 m
17   000_0000_0001 0 00 1 m
19   000_0000_0001 0 01 1 m
56   000_0000_0011 1 00 0 m
209  000_0000_1101 0 00 1 m
11   000_0000_0000 1 01 1 m
4    000_0000_0000 0 10 0 h
43   000_0000_0010 1 01 1 m
5    000_0000_0000 0 10 1 h
36   000_0000_0010 0 10 0 m
8    000_0000_0000 1 00 0 m
16   000_0000_0001 0 00 0 m

```

0	MEM[0:1], MEM[16:17], MEM[208:209], MEM[16:17]	MEM[8:9], MEM[56:57], MEM[8:9]
1	MEM[18:19], MEM[42:43]	MEM[10:11]
2	MEM[4:5]	MEM[20:21], MEM[36:37]
3		

8. (8 points) Mean Time Between Failures (MTBF), Mean Time To Replacement (MTTR), and Mean Time To Failure (MTTF) are useful metrics for evaluating the reliability and availability of a storage resource. Explore these concepts by answering the questions about a device with the following metrics.

MTTF	MTTR
5 Years	20 days

- (a) Calculate the MTBF for the devices.  
(b) Calculate the availability for this device.

(a)  $MTBF = MTTF + MTTR = 5 \times 365 + 20 = 1845$  days  
(b)  $Availability = MTTF / MTBF = (5 \times 365) / 1845 = 98.9\%$

9. (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 3, x is 5 and y and z are 1. w, x, y, and z are type `int`.

Core 1:  $y = 5 / (z + w);$

Core 2:  $x = (x + y) / (w + 1);$

Core 3:  $z = w * x - y;$

What are all the possible resulting values of w, x, y, and z? You will need to examine all possible interleavings of instructions.

	w	s	y	z
1, 2, 3	3	1	1	2
1, 3, 2	3	1	1	14
2, 1, 3	3	1	1	2
2, 3, 1	3	1	1	2
3, 1, 2	3	1	0	14
3, 2, 1	3	1	0	14

10. (8 points) A cache designer wants to increase the size of a 4 KB virtually indexed, physically tagged cache. Given a page size of 64 KB, is it possible to make a 16 KB direct-mapped cache, assuming 4 words per block? If not, how would the designer increase the size of the cache? If so, is 16 KB the largest direct-mapped cache size possible?

$$16KB \times \frac{1word}{4bytes} \times \frac{1block}{4words} \times \frac{1set}{1block} = 1Ksets$$

There is a byte offset of 2 bits, a block offset of 2 bits and an index of 10 bits. For this to be possible, all of these need to fit inside the page offset. For a page size of 64 KB, the page offset is 16, so yes, it is possible.  $10 + 2 + 2 \leq 16$ . Since the number of bits is less than the max (14), we can make a bigger direct mapped virtually indexed, physically tagged cache with 4 words per block. We can make a direct mapped cache 4 times bigger, or 64 KB. We could also make a bigger cache by having multiple blocks per set.

11. (5 points) Pseudoinstructions are not part of the MIPS instruction set but often appear in MIPS programs. For the pseudoinstruction in the following table, produce a minimal sequence of actual MIPS instructions to accomplish the same thing. You may need to use \$at for some of the sequences. In the table, big refers to a specific number that requires 32 bits to represent and small to a number that can fit in 16 bits.

Pseudoinstruction	What it accomplishes
beq \$t2, big, L	if (\$t2 = big) go to L

```
lui  $at, big[31..16]
ori  $at, $at, big[15..0]
beq  $t2, $at, L
```

12. (4 points) For the following MIPS instruction, show the value of the opcode (OP), source register (RS), and target register (RT) fields. For the I-type instructions, show the value of the immediate field, and for the R-type instructions, show the value of the destination register (RD) field.

```
srl  $t0, $s0, 6
```

**OP = 0, RS = 0, RT = 16, RD = 8**

13. (5 points) Consider the following loop. Assume that perfect branch prediction is used (no stalls due to control hazards), that there are no delay slots, and that the pipeline has full forwarding support. Also assume that many iterations of this loop are executed before the loop exits.

```
Loop: add  $s1, $s2, $s1
      lw   $s2, 0($s1)
      lw   $s2, 16($s2)
      slt  $s1, $s2, $s4
      beq  $s1, $s7, Loop
```

At the start of the cycle in which we fetch the first instruction of the third iteration of the loop, what is stored in the IF/ID register?

Cycle	IF	ID	EX	MEM	WB
1	add				
2	lw	add			
3	lw	lw	add		
4	slt	lw	lw	add	
5	slt	lw	bubble	lw	add
6	beq	slt	lw	bubble	lw
7	beq	slt	bubble	lw	bubble
8	add	beq	slt	bubble	lw

Cycle 8 is the cycle of interest and beq \$s1, \$s7, Loop is the instruction in the IF /ID register.

14. (10 points) Consider two different implementations, P1, and P2, of the same instruction set. There are five classes of instructions (A, B, C, D, and E) in the instruction set. P1 has a clock rate of 4 GHz, and P2 has a clock rate of 6 GHz. The average number of cycles for each instruction class for P1 and P2 are listed in the following table.

	CPI Class A	CPI Class B	CPI Class C	CPI Class D	CPI Class D
P1	1	2	3	4	5
P2	3	3	3	5	5

If the number of instructions executed in a certain program is divided equally among the five classes of instructions except for class A, which occurs twice as often as each of the others, how much faster is P2 than P1?

$$ET = \frac{IC * CPI}{CR}, \quad CPI = \sum_i w_i CPI_i$$

$IC_{P1} = IC_{P2},$

$$w_A = 2/6, w_B = 1/6, w_C = 1/6, w_D = 1/6, w_E = 1/6$$

$$CPI_{P1} = (2*1 + 1*2 + 1*3 + 1*4 + 1*5)/6 = 16/6 \quad CPI_{P2} = (2*3 + 1*3 + 1*3 + 1*5 + 1*5)/6 = 22/6$$

$$\frac{P_{P2}}{P_{P1}} = \frac{ET_{P1}}{ET_{P2}} = \frac{\frac{IC_{P1} * CPI_{P1}}{CR_{P1}}}{\frac{IC_{P2} * CPI_{P2}}{CR_{P2}}} = \frac{\frac{CPI_{P1}}{CPI_{P2}}}{\frac{CR_{P1}}{CR_{P2}}} = \frac{CPI_{P1} * CR_{P2}}{CPI_{P2} * CR_{P1}} = \frac{16/6 * 6 \times 10^9}{22/6 * 4 \times 10^9} = 1.09$$

15. (20 points) a) (5 points) Consider the following loop executing on a MIPS pipeline with MEM/WB forwarding only. Calculate the number of cycles it takes to execute this loop, neglecting pipeline fill cycles. b) (12 points) Unroll the loop so that 2 iterations of the loop are executed at once and schedule the unrolled code for a single issue MIPS pipeline. c) (3 points) Calculate the speedup from the original loop to the unrolled loop.

```

    for (i=0; i < n; i++)
        A[i] = B[i] + C[i];

    addi    $s0, $t1, 200
Loop: add   $t5, $t2, $t1
    add    $t6, $t3, $t1
    add    $t7, $t4, $t1
    lw     $t5, 0($t5)
    lw     $t6, 0($t6)
    add    $t5, $t5, $t6
    sw     $t5, 0($t7)
    addi   $t1, $t1, 4
    bne    $t1, $s0, Loop

```

(a)

```

    addi    $s0, $t1, 200
Loop: add   $t5, $t2, $t1
    add    $t6, $t3, $t1
    add    $t7, $t4, $t1
    lw     $t5, 0($t5)
    lw     $t6, 0($t6)
    stall
    add    $t5, $t5, $t6
    stall
    sw     $t5, 0($t7)
    addi   $t1, $t1, 4
    stall
    bne    $t1, $s0, Loop
    stall

```

Number of iterations is  $200/4 = 50$

Number of cycles per iteration is 9 instructions + 3 data stall cycles + 1 control stall cycle

Total number of cycles =  $49 (13) + (13) = 649$

### Unrolling

```

Loop:  add   $t5, $t2, $t1
      add   $t6, $t3, $t1
      add   $t7, $t4, $t1
      lw    $t8, 0($t5)
      lw    $t9, 0($t6)
      add   $t8, $t8, $t9
      sw    $t8, 0($t7)
      lw    $t5, 4($t5)
      lw    $t6, 4($t6)
      add   $t5, $t5, $t6
      sw    $t5, 4($t7)
      addi  $t1, $t1, 8
      bne   $t1, $s0, Loop

```

**(b) Scheduling**

```
Loop:  addi  $t1, $t1, 8
        add  $t5, $t2, $t1
        add  $t6, $t3, $t1
        add  $t7, $t4, $t1
        lw   $t0, -8($t5)
        lw   $t1, -8($t6)
        lw   $t2, -4($t5)
        lw   $t3, -4($t6)
        add  $t0, $t0, $t1
        add  $t2, $t2, $t3
        sw   $t0, -8($t7)
        sw   $t2, -4($t7)
        bne  $t1, $s0, Loop
```

**(c)**

Number of iterations = 25

Number of cycles per iteration = 13 instructions + 1 control stall

Total number of cycles = 24(14) + 13 = 349

Speedup –  $E_{old}/E_{new} = 649 \text{ cycles}/349 \text{ cycles} = 1.86$