The University of Alabama in Huntsville
ECE Department
CPE 431 01
Test 1 Solution
Fall 2017

You *must* show your work to receive full credit: You may use additional sheets of paper for your work, please put your name on each additional sheet.
Remember: ET = IC*CPI*CT

1. (1 point) When a planned instruction cannot execute in the proper cycle because the hardware does not support the combination of instructions that are set to execute, it is due to a _**structural hazard**_.
2. (1 point) An _**address**_ is a value used to delineate the location of a specific data element within a memory array.
3. (1 point) _**Hexadecimal**_ numbers are numbers expressed in base 16.
4. (1 point) The _**opcode**_ is the field that denotes the operation and format of an instruction.
5. (1 point) A _**return address**_ is a link to the calling site that allows a procedure to return to the proper address; in MIPS it is stored in register $ra.
6. (10 points) Consider a computer running a program that requires 250 s, with 70s spent executing FP instructions, 85 s executing L/S instructions, and 40 s spent executing branch instructions. By how much is the total time reduced if the time for INT operations is reduced by 20%.

   ET_{total_old} = 250s
   ET_{INT_old} = 250s – 70s – 85s – 40s = 55s
   $ET_{INT\_new} = ET_{INT\_old} * 0.8$ = 55s *0.8 = 44s
   $ET_{INT\_old} – ET_{INT\_new}$ = 55s – 44s = 11s
   ET_{total_old} = 250s – 11s = 239 s
   $P_{New}/P_{old} = ET_{old}/ET_{new}$ = 250s/239s = 1.05
   The total time is reduced by 11s.

7. (15 points) Translate the following C code to MIPS. Assume that the variables `f`, `g`, `h`, `i` and `j` are given and are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Assume that the base address of the arrays A and B are in registers `$s6` and `$s7`, respectively. Assume that the elements of the arrays A and B are 4-byte words:

```
B[i+j] = A[8]
add   $t0, $s3, $s4          # $t0 ← i + j
sll   $t0, $t0, 2            # $t0 ← 4*(i + j)
add   $t0, $s7, $t0          # $t0 ← &B[i + j]
lw    $t1, 32($s6)           # $t1 ← A[8]
sw    $t1, 0($t0)            # B[i + j] ← A[8]
```

8. (15 points) For a benchmark assume an execution time of 1040 s, CPI of 2.59, and clock rate of 3.5 GHz. If the execution time is reduced by an additional 10% without affecting the CPI and with a clock rate of 4 GHz, determine the number of instructions.

$ET_{new} = 0.9 * ET_{old} = 0.9 * 1040s = 936s$
$CPI_{new} = CPI_{old} = 2.59$ cycles/instruction
$CR_{new} = 4$ E9 cycles/s
$ET = IC * CPI * CT$
$ET = (IC * CPI)/CR$
$IC = (ET * CR)/CPI$
$IC_{new} = (ET_{new} * CR_{new})/CPI_{new} = (936s * 4$ E9 cycles/s$)/2.59$ cycles/instruction
$= 1.44$ E12 instructions

9. (5 points) Unconditional branches are not part of the MIPS assembly language. Show the MIPS I instruction(s) that can be used to have the same effect as an unconditional branch.

```
beq     $zero, $zero, target
```

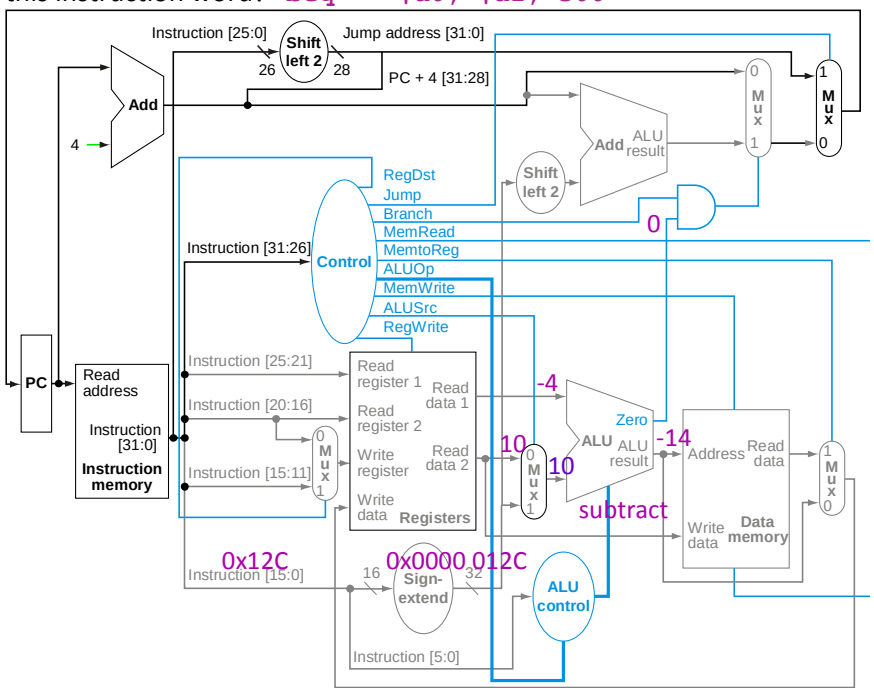
10. (10 points) A MIPS processor fetches the following instruction word:

0001 0000 1000 0101 0000 0001 0010 1100

Assume that data memory is all zeros and that the processor's registers have the following values at the beginning of the cycle in which the above instruction word is fetched.

r0	r1	r2	r3	r4	r5	r6	r8	r12	r31
0	-1	2	-3	-4	10	6	8	2	-16

What instruction is this and what is the output of the sign-extend, Zero, and the ALU result for this instruction word? `beq $a0, $a1, 300`



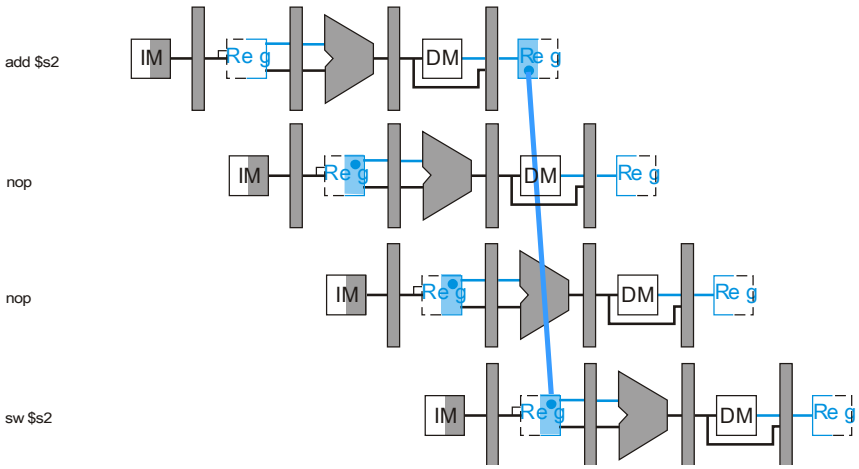
Instruction: _ `beq $a0, $a1, 300`_
Sign Extend: _0x0000 012C_
Zero: _0_
ALU result: -14, 0xFFFF FFF2_

11. (15 points) In this exercise, we examine how data dependences affect execution in the basic 5-stage pipeline. Refer to the following sequence of instructions:

```
1    add    $s2, $s2, $s1
2    sw     $s2, 0($t1)
3    lw     $s1, -4($s1)
4    add    $s2, $s2, $s1
```

Assume there is no forwarding in this pipelined processor. Indicate hazards and add nop instructions to eliminate them.

Hazards are (1)-(2)- `$s2` and (3) – (4) `$s1`, With no forwarding, the ID stage of the consuming instruction must be aligned with the WB stage of the producing instruction, requiring two nop instructions.




```
add    $s2, $s2, $s1
nop
nop
sw     $s2, 0($t1)
lw     $s1, -4($s1)
nop
nop
add    $s2, $s2, $s1
```

12. (10 points) In this exercise, we examine how pipelining affects the clock cycle time of the processor. Problems in this exercise assume that individual stages of the datapath have the following latencies:

IF	ID	EX	MEM	WB
300 ps	250 ps	180 ps	350 ps	200 ps

Also, assume that instructions executed by the processor are broken down as follows:

alu	beq	lw	sw
45%	20%	20%	15%

What is the clock cycle time in a pipelined and non-pipelined processor?

$CT_{non\text{-}pipelined}$ = ∑(all stage latencies) = 300ps + 250 ps + 180 ps + 350 ps + 200 ps = 1280ps

$CT_{pipelined}$ = Max(all stage latencies) = Max(300ps, 250ps, 180ps, 350ps, 200ps) = 350 ps

13. (15 points) Write down the hexadecimal representation of the decimal number -1468.265625 in both the IEEE 754 single and double precision formats.

```
          0   1               x2    0.265625
/2        1   0               x2    0.531250      0
/2        2   1               x2    1.06250       1
/2        5   1               x2    0.1250        0
/2       11   0               x2    0.250         0
/2       22   1               x2    0.5           0
/2       45   1               x2    1.0           1
/2       91   1
/2      183   1
/2      367   0
/2      734   0
/2     1468
```

$101\ 1011\ 1100.010001 = 1.0110\ 1111\ 0001\ 0001 \times 2^{10}$

Single Precision

S = 1, EXP + BIAS = 10 + 127 = 137 = 1000 1001
Fraction = 0110 1111 0001 0001
1100 0100 1011 0111 1000 1000 1000 0000 = 0xC4B7 8880

Double Precision

S = 1, EXP + BIAS = 10 + 1023 = 1033 = 100 0000 1001
Fraction = 0110 1111 0001 0001
1100 0000 1001 0110 1111 0001 0001 0000
0000 0000 0000 0000 0000 0000 0000 0000 = 0xC096 F110 0000 0000