Chris Bero

HW 2 2.2, 2.4, 2.5, 2.7, 2.13.1, 2.13.2 CPE 431

2.2 f = (g+h) + i;

2.4 f → $s0, g → $s1, h → $s2, i → $s3, j → $s4
A[0] → $s6, B[0] → $s7

```
lw   $s0, 0($t0)      f = A[f]
addi $t2, $t0, 4      $t2 = &A[f+1]
lw   $t0, 0($t2)      $t0 = A[f+1]
add  $t0, $t0, $s0    $t0 = A[f+1] + A[f]
sw   $t0, 0($t1)      B[g] ← $t0
```

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

2.5.

```
sll  $t0, $s0, 2
add  $t0, $s6, $t0
sll  $t1, $s1, 2
add  $t1, $s7, $t1
lw   $s0, 0($t0)
lw   $t0, 4($t0)
add  $t0, $t0, $s0
sw   $t0, 0($t1)
```

q Ask about typo on pg. 69

2.7 0xABCDEF12

Big Endian:

Addr	Val
0	AB
4	CD
8	EF
12	12

Little Endian

Addr	Val
0	12
4	EF
8	CD
12	AB

2.13.1 $s1 > 2,147,483,519
$s1 ≤ 2,147,483,647

2.13.2 $s1 ≤ -2,147,483,520
$s1 > -2,147,483,648

2.13.3 $s1 ≤ -2,147,483,520
$s1 > -2,147,483,648

2.15 I-type

op	rs	rt	addr
43	10	9	32
101011	1010	1001	100000

0xAEA60

2.24 JumpAddr = PC+4[31:28], addr, 2'b0
(addr: 27..2, 2^25)

Jump: No, need 32 bits for address field.
Beq: No, need 32 bits for address.

2.26 .1) $s2 = 20

.2)
```
for (int i=10; i>0; i--) {
    B = B+2;
}
```

.3) (5·N)+2

2.40 No, need 32 bits of address, only have 27.

Christopher
Bero

HW2 3.3, 3.6, 3.8 CPE 431

3.3 $0x5ED4 \rightarrow (0101\ 1110\ 1101\ 0100)_2$

- Hex is a valuable tool for discussing binary numbers because it is a compact notation and can be quickly translated by expanding each 4-bit character

3.6 $185_{10} \rightarrow 1011\ 1001_2$ $122_{10} \rightarrow 0111\ 1010_2$

$$\begin{array}{r} 1011\ 1001 \\ -\ 0111\ 1010 \\ \hline 0011\ 1111 \rightarrow 63_{10} \end{array}$$

Neither overflow nor underflow.

3.8 $185_{10} \rightarrow 1011\ 1001_2$

- This problem says to calculate 185-122 in 8-bit signed binary, but you cannot even begin because 185 is too large for this representation. Without even running the operation there is overflow.