Solution

Problem 1: (14 points)

[1]	0101	[2]	0000 0101	[3]	0011
[4]	0001 1100	[5]	0	[6]	0000 1001
[7]	0000 0100				

Problem 2: (10 points)

[1]	0x00000610	[2]	0x10223400
[3]	0x00000614	[4]	0x11ff00a8
[5]	0x0000061c	[6]	0x87654010
[7]	0x00000618	[8]	0xcdab0114
[9]	%eax	[10]	0x0000061c

Problem 3: (16 points)

1

[1]	40	[2]	80
[3]	8	[4]	16
[5]	0x8049790	[6]	0x600a20
[7]	0x804978d	[8]	0x600a19
[9]	0x8049794	[10]	0x600a28
[11]	0x80497cc	[12]	0x600a98

2 X86: 9 bytes wasted; X86-64: 25 bytes wasted

3 X86: 8 bytes can be saved; X86-64: 24 bytes can be saved

Problem 4: (10 points)

1

[1]	15	[2]	1 11111 000000
[3]	1 00000 111111	[4]	0 00001 000000

2. 1 01101 100000

3. 0 10010 111010

Problem 5: (23 points)

1
[1]	4	[2]	'f'
[3]	NONE	[4]	'h'
[5]	.L8	[6]	movb
[7]	$0x1	[8]	.L9
[9]	%ebx	[10]	$0x61
[11]	*.L7(, %ecx, 4)	[12]	subl
[13]	addl	[14]	orl

2 'H'

Problem 6: (27 points)

1
[1]	%edx	[2]	0xc(%ebp)
[3]	%eax	[4]	10 00 00 00
[5]	8048404	[6]	80484a1

2 size = 8

3
[1]	0xffffd4cc	[2]	0xffffd4f8
[3]	0xffffd4c4	[4]	0xffffd4c8
[5]	0xffffd4d0	[6]	0xffffd4f8

4 %edx = 0xa (%eax) = 0x9

next instruction: 804842c or mov %esi,0x4(%esp)

5
[1]	0x3	[2]	0xd
[3]	0xffffd4c8	[4]	0x804842a