

CMPT295: Assignment 3

Daniel Wallman (301383524)

January 31, 2019

1. a.

i.
$$\begin{array}{r} 01010110 \\ + 0110011 \\ \hline 11001001 \end{array}$$
 No overflow since carry out of MSB is 0.

ii.
$$\begin{array}{r} 11111011 \\ + 01000111 \\ \hline 01000010 \end{array}$$
 Overflow occurs since carry out of MSB is 1.

iii.
$$\begin{array}{r} 00101000 \\ + 11001110 \\ \hline 11110110 \end{array}$$
 No overflow.

1.(b)

i.
$$\begin{array}{r} 10100111 \\ + 0010011 \\ \hline 11001010 \end{array}$$
 No overflow occurred since carry-in and carry-out of MSB are identical.

ii.
$$\begin{array}{r} 01000101 \\ + 0011011 \\ \hline 10000000 \end{array}$$
 Overflow occurred since carry-in was 1 and carry-out 0.

iii.
$$\begin{array}{r} 10011111 \\ + 1101111 \\ \hline 01011110 \end{array}$$
 Overflow occurred since carry-in was 0 and carry-out 1.

1. (c) The first one de-references and adds in one line. The second one de-reference and then adds. Another difference is that the first one will add the value of the whole 64-bit while the second will only add the value of the last 32-bit.

2. (b) Case 1

x:	⁰ 1 ⁰	001 100 ¹ 1
y:	0	110 0101
res:	1	111 1110

Case 2

x:	¹ 1 ¹	100 100 ¹ 1
y:	0	101 0101
res:	0	001 1110

3.


```

1 # %edi = signal[]
2 # %esi = h[]
3 # %edx = length of arrs
4 # %eax = sum
5 # %ecx = loop counter
6 # %r8d = h counter
7 # %r9d = temp sum
8 # %r10d = negation of ecx
9
10      .globl conv
11 conv:
12      movl $0, %eax
13      movl $0, %ecx
14      leal (%esi, %edx, 8), %r8d      # r8d = adress of h[n-1]
15
16 loop:  cml %ecx, %edx
17      je end
18
19      movl %ecx, %r10d                # negate ecx for backwards loop
20      neg %r10d
21
22      movl $0, %r9d                  # r9d = 0
23      cmp %rsi, (%r8d, %r10d, 8)      # if r9d < h then r9d = h[n-1-m]
24      cmovg (%r8d, %r10d, 8), %r9d
25
26      imull (%edi, %ecx, 8), %r9d      # r9d *= signal[m]
27      addl %r9d, %eax                # eax = r9d
28
29      incl %ecx
30      jmp loop
31 end:
32      ret

```