

COMP273 - Assignment 1

Richard Ni

260674646

1.

a) Like a C string, the sequence of characters is always terminated by a '\0', which is represented by 8 bits of 0s. For a fraction, I would do the same as I would allocate 40 bits of memory. The first 16 bits for the numerator integer, the second 8 bits to be all 1s to represent a fraction line, then the third 16 bits for the denominator integer. When addressing, this fraction will use up 5 bytes at the least.

b) For example, 5/7 would be: 00000000 00000101 11111111 00000000 00000111

c) Because each fraction uses up 40 bits of space, 10 fractions will use up 400 bits, or 50 bytes.

d) Because 8 bits of 1s already represents a character, we would need to compromise that character such that when it is between two integer values, it will be used as a fraction symbol instead.

2.

A) (i) Decimal: $= (D \cdot 16^0) + (C \cdot 16^1) + (4 \cdot 16^2) + (5 \cdot 16^3) + (1 \cdot 16^4) = 13 + 192 + 1024 + 20480 + 65536 = 87245$

Binary: $= (0001) + (0100) + (0011) + (1100) + (1101) = 00010100001111001101$

(ii) Decimal: $= (2^0) + (2^1) + (2^2) + (2^3) + (2^4) + (2^5) + (2^7) + (2^8) + (2^{11}) = 2495$

Hex: $= (1001) + (1011) + (1111) = 9 + B + F = 9BF$

(iii) Binary: $= (2^{10}) + (2^0) = 10000000001$

Hex: $= (4 \cdot 16^2) + (1 \cdot 16^0) = 401$

B)

Address	Value
0010 0010 0101	1001000
0010 0010 0110	1100101
0010 0010 0111	1101100
0010 0010 1000	1101100
0010 0010 1001	1101111
0010 0010 1010	100000

0010 0010 1011	1010111
0010 0010 1100	1101111
0010 0010 1101	1110010
0010 0010 1111	1101100
0010 0011 0000	1100100
0010 0100 0000	0

3.

A)

```
int byte;
int address = 0x100;
for (byte = 0; byte < 100; byte += 2) {
    LOAD R1, address;
    LOAD R2, address + 1;
    SAVE R1, 0x0F;
    SAVE R2, 0x0F;
    address += 2;
}
```

B)

The IR gives the LOAD command to the sequencer for every byte of the 100 bytes. Then the address of the byte is stored in AR, and data in DR is set to the value stored in the address in AR and the data is copied to R1. IP does the same with the next byte, and puts it in R2. The IP uses SAVE R1 to 0F(16) which pushes the address in AR, and then goes through the bus to address 0F(16) in RAM. Then IR does the same with R2. Since 2 bytes are being sent per loop, the byte must increment by 2 each time.