Answers to End of Chapter Reviews and Exercises

for Assembly Language for x86 Processors, 7th Edition

by Kip R. Irvine

Chapters 1 and chapter 2 sample

Revision date: 1/18/2014

Chapter 1

1.7.1 Short Answer Questions

1. Most significant bit (the highest numbered bit).
2. (a) 53 (b) 150 (c) 204
3. (a) 110001010 (b) 110010110 (c) 100100001
4. 00000110
5. (a) 8 (b) 32 (c) 64 (d) 128
6. (a) 12 (b) 16 (c) 16
7. (a) 35DA (b) CEA3 (c) FEDB
8. (a) 0000 0001 0010 0110 1111 1001 1101 0100
   (b) 0110 1010 1100 1101 1111 1010 1001 0101
   (c) 1111 0110 1001 1011 1101 1100 0010 1010
9. (a) 58 (b) 447 (c) 16534
10. (a) 98 (b) 1203 (c) 671
11. (a) FFE8 (b) FEB5
12. (a) FFEB (b) FFD3
13. (a) 27641 (b) –16093
14. (a) 19666 (b) –32208
15. (a) −75 (b) +42 (c) −16
16. (a) −128 (b) −52 (c) −73
17. (a) 11111011 (b) 11010110 (c) 11110000
18. (a) 10111000 (b) 10011110 (c) 11100110
19. (a) AB2 (b) 1106
20. (a) B82 (b) 1316
21. 42h and 66d
22. 47h and 71d
23. 2^{29}– 1, or 6.80564733841876926926749214863 53 X 10^{38}

24. $2^{86} - 1$, or 77371252455336267181195263

25. Truth table:

A	B	$A \vee B$	$\neg(A \vee B)$
F	F	F	T
F	T	T	F
T	F	T	F
T	T	T	F

26. Truth table: (last column is the same as #25)

A	B	$\neg A$	$\neg B$	$\neg A \wedge \neg B$
F	F	T	T	T
F	T	T	F	F
T	F	F	T	F
T	T	F	F	F

27. It requires 2^4 (16) rows.

28. 2 bits, producing the following values: 00, 01, 10, 11

1.7.2Algorithm Workbench

1. Code example (C++)

```
int toInt32(string s) {
    int num = 0;
    for(int i = 0; s[i] >= '0' && s[i] <= '1'; i++) {
        num = num * 2 + s[i]-'0';
    }
    return num;
}
```

2. Code example (C++)

```
int hexStrToInt32(string s) {
    int num = 0;
     for(int i = 0; ; i++) {
      if( s[i] >= '0' && s[i] <= '9' )
        num = num * 16 + s[i]-'0';
      else if( s[i] >= 'A' && s[i] <= 'F' )
        num = num * 16 + (s[i]-'A'+10);
      else
        break;
    }
     return num;
}
```

3. Code example (C++)

```
string intToBinStr( int n )  {
    vector<int> stack;

    do {
      int quotient = n / 2;
```

```
      int remainder = n % 2;
      stack.push_back(remainder);
      n = quotient;
    } while( n > 0 );

    string s;
    while( stack.size() > 0 ) {
      s += (stack.back() + '0');
      stack.pop_back();
    }
    return s;
}
```

4. Code example (C++)

```
string intToHexStr( int n )  {
    vector<int> stack;

    do {
      int quotient = n / 16;
      int remainder = n % 16;
      stack.push_back(remainder);
      n = quotient;
    } while( n > 0 );

    string s;
    while( stack.size() > 0 ) {
      int d = stack.back();
      if( d >= 0 && d <= 9 )
        s += (stack.back() + '0');
      else // probably a hex digit
        s += (stack.back() - 10 + 'A');
      stack.pop_back();
    }
    return s;
}
```

5. Code example (C++)

```
string addDigitStrings( string s1, string s2, int base ) {
    string sumStr;
    int carry = 0;

    for(int i = s1.size() - 1; i >= 0; i--) {
      int dval = (s1[i] - '0') + (s2[i] - '0') + carry;
      carry = 0;
      if( dval > (base - 1) ) {
        carry = 1;
        dval = dval % base;
      }
      sumStr.insert(sumStr.begin(), (dval + '0'));
    }
    if( carry == 1 )
      sumStr.insert( sumStr.begin(), 1 + '0');
    return sumStr;
}
```

Chapter 2

2.8.1Short Answer Questions

1. EBP

2. Choose 4 from: Carry, Zero, Sign, Direction, Aux Carry, Overflow, Parity.
3. Carry flag
4. Overflow flag
5. True
6. Sign flag
7. Floating-point unit
8. 80 bits
9. True
10. False
11. True
12. False
13. True
14. False
14. False
15. False
16. True
17. False
18. True
19. False
20. False
21. True
22. True
23. False
24. False
25. Hardware, BIOS, and OS
26. It gives them more precise control of hardware, and execution is faster.