**Due Date: December 8, 2017 Points: 48**

This assignment will have to be printed out, completed by hand and returned to the faculty member. Note the assignment can be taken to the ECE main office 3081 and given to the staff there.

1) Number Conversion: Convert the following numbers to the requested base. Show process for conversion! (10 points)

1. 101001012 → \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Four digit Hexadecimal
2. 101010 → \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(16 bit signed number)
3. Continued
   1. 55AA16 → \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(16 bit binary)
   2. 222210 → \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Four digit Hexadecimal
   3. 101116 → \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_16 bit binary
4. Binary Addition: Show the results of the following binary additions, being sure to ***show all carries.***

Assume a zero carry in. (9 points)

* 1. 0 0 1 0 1 0 1 0 0

+ 0 0 1 0 1 0 1 0

\_\_|

Carry Out ^

* 1. 1 0 1 0 0 0 1 0 0

+ 1 0 1 0 1 0 0 0

\_\_|

Carry Out ^

* 1. 1 0 1 0 0 0 1 0 0

+ 1 0 1 0 1 0 0 0

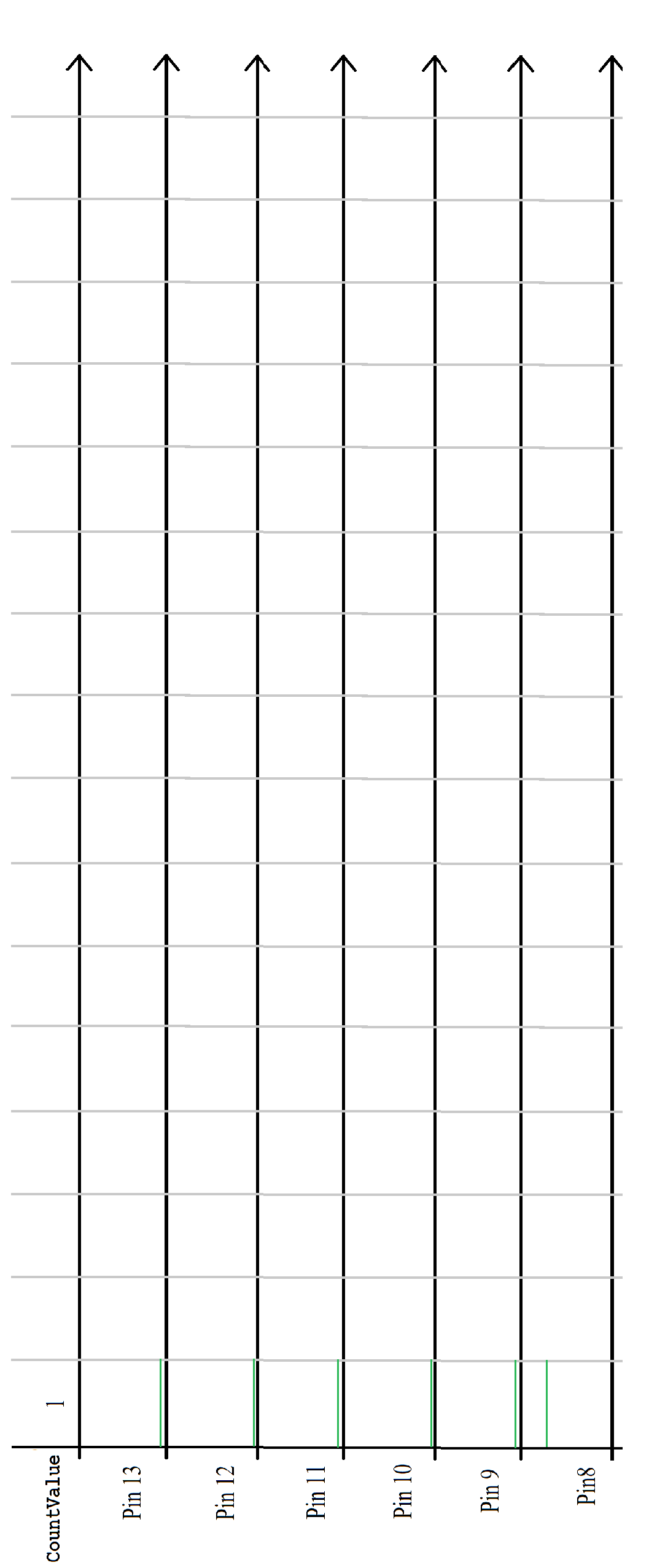
\_\_|

Carry Out ^

* 1. Overflow: Interpreting the additions in question 2 as 8-bit signed numbers, do any show a signed overflow and why? (3 points)

1. Masking: Write a line of c code that performs the following operations. **You can not use the bitSet or bitClear functions**. (8)
   1. Force bit 4 in the variable A, high.
   2. Force bit 9 in the variable B, low.
   3. Toggle bit 6 in the variable A.
   4. Toggle bit 6 in the variable A.

1. Conditionals: For what values of A do the following logical equations evaluate as true. (8)
   1. if( A > 5 && A < 0 )
      1. = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. if( A )
      1. = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   3. if( A < 10 && A > -10 )
      1. = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   4. if( A & 0x0042 )
      1. = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Code Reading/Timing: Fill in the timing diagram, as it would appear if the code in Appendix A were running. (10)



Appendix A: Output Function Generator.

#include <MsTimer2.h> int CountValue = 1; void CountValueISR() { // Compute next output value.

// if bit 5 set is equal to bit 4

if (bitRead(CountValue, 5) == bitRead(CountValue, 4))

CountValue = CountValue << 1; // shift to the left else

CountValue = (CountValue << 1) // shift to the left + 1; // and set bit 0.

CountValue &= 0x3f; // Send output to portb, or pins 13-8.

PORTB = CountValue;

} // End of ISR

// Runs at start up to initialize system. void setup() { DDRB |= 0x3f; // Set pins 13-8 as outputs.

// set up Timer2 for 50 millisecond interval.

MsTimer2::set(50, CountValueISR);

MsTimer2::start();

} // End of setup()

// loop called continuously. void loop() { // Nothing in loop.

} // End of loop.