Homework #2: DSP Building Blocks

Professor Deepa Kundur
University of Toronto

Questions

Please print this out and answer the following questions in the space provided below. Please add additional sheets if necessary or use the backs of sheets. For full points, please provide explanations and reasoning in your solutions.

1. *Multiplier*. In class we discussed the 4x4 Braun multiplier structure, which is the basis of most of today's commercial multiplier implementations. Draw the structure of a 3x3 Braun multiplier for unsigned fixed-point integers using the same notation and structure as used for the 4x4 Braun in class. Please use the back of this page or an additional page (and draw in landscape mode if needed).

2. *Shifter*. One purpose of a shifter is to scale numbers prior to addition in order to avoid overflow at the cost of precision. In this question, we will walk you through the process of why a shifter is needed and the trade-off of using one. Please answer all parts.

    a. It is required to find the sum of 64 numbers each represented by 16 bits in unsigned fixed-point format. How many bits should the accumulator have so that the sum can be computed without overflow error or loss of accuracy?

    b. If it is decided for consistency and cost to have an accumulator with only 16 bits but shift the numbers before the addition to prevent overflow, by how many bits and in what direction (left or right) should each number be shifted to accommodate this practical issue?

c. What scale factor must one multiply to the accumulator register to obtain the *actual* sum of the 64 numbers? How can this be implemented with a shifter? *Hint*: shifting a binary number to the right by one place effectively divides it by two (throwing out any remainder) and shifting it to the left by one place multiplies it by two.

d. How many least significant bits (in the addition computation) are lost by implementing shifting to address overflow issues? What is the *maximum* possible error that can arise in computing the sum using the approach of part b?

3. *MAC*. Consider a MAC unit whose inputs are 16-bit numbers. If 256 products are to be summed up in this MAC, how many guard bits should be provided for the accumulator to prevent overflow?

4. *Addressing Modes*. Consider an input signal consisting of 16 samples x_0, x_1, x_2, …, x_{15}. What is the bit-reversed order of the input sequence?

5. *Parallelism and Pipelining*. Consider a 4-tap FIR filter:

$$y(n) = \sum_{k=0}^{3} h(k)x(n-k)$$

a. How would you implement this using a single MAC unit? Let T be the time taken to compute one product term and add it to the accumulator. You may use multiplexers if needed in your implementation.

b. How would you implement this using *two* MAC units in a parallel implementation? Let T be the time taken to compute one product term and add it to the accumulator. You may use multiplexers if needed in your implementation.

c. If you wanted to speed up the implementation of part 5(a) by a factor of *four*, what type of implementation could you use? What additional hardware would you need?