Eurecom
Signal Processing Technologies

Final Examination

Date: June 26, 2015

Duration: 2 hours

Your responses should not be long, but rather precise. Answer all questions.

1 Short Questions

1. When is fixed-point processing too problematic for signal processing algorithms, so that floating-point would be a better option?

2. Why did we perform rescaling in intermediate stages of the FFT implementation in the first lab session? Be sure to explain what can be gained by optimizing the way scaling is distributed.

3. In the second lab exercise on SIMD programming, what is the efficiency of the vectorized FFT butterfly (with twiddle factors) in terms of real multiplications per instruction and real additions per instruction? You can assume that one instruction corresponds to a basic SIMD intrinsic.

4. Suppose you have an arbitrary input signal to an FIR filter using a Q1.15 representation. Suppose the coefficients of the FIR filter are also represented using Q1.15 and satisfy $\sum_i |h_i| < 1$, where h_i is the i^{th} filter coefficient. Assume that additions are Q1.15 with saturation and the results of multiplications are rescaled to Q1.15.

   Will overflow or saturation ever occur? Explain your answer.

2 Long Question

In the second lab session we first optimized a 16-point FFT on the 128-bit SSE4 SIMD engine. The input and output was assumed to be Q1.15 real and imaginary samples taken in the normal order. There were three basic operations that had to be performed,

1. a radix-4 butterfly without twiddle factors
2. a 4x4 transpose operation
3. a radix-4 butterly with multiplication of the input by twiddle factors

First, explain how each of these operations was parallelized for 128-bit SIMD and argue that the 16-point FFT maps well to this size of SIMD arithmetic.

Second, suppose that now that we have a 256-bit integer SIMD engine (like the AVX2 in generation 4/5 Intel processors).

1. How many complex samples could be handled by a single instruction?
2. Assuming similar intrinsics as in the 128-bit case, what difficulties do you see in performing the same parallelization technique?
3. Do you think it would be appropriate to consider larger basic DFTs instead of 16-point (say 64-point). Or, in other words, can you find a way to parallelize better for 64-point instead. Take only one of the above operations to make your case.