

FPGA Architecture for Gigahertz-Sampling Wideband IF-to-Baseband Conversion

Jeffrey O. Coleman James J. Alter Dan Scholnik
 jeffc@alum.mit.edu alter@radar.nrl.navy.mil scholnik@nrl.navy.mil

Naval Research Laboratory
 Washington, DC

ABSTRACT

We tutorially outline downconversion to (complex) baseband of a 750 MHz IF of 400 MHz bandwidth using a 1 GHz sample rate and polyphase processing with look-up-table multipliers in an FPGA clocked at 125 MHz.

1. INTRODUCTION

In this tutorial design-example paper, we outline design approaches for an FPGA-based DSP system for IQ downconversion of a 400 MHz wide IF signal centered at 750 MHz and sampled at 1 GHz. Downconversion through sampling eliminates explicit quadrature carriers. An FPGA clock rate of only 125 MHz requires polyphase (parallel) processing. Such downconversion systems are important in radar, satellite communication, terrestrial microwave communication, and base stations for wireless communication.

2. SIGNAL-PROCESSING DESIGN

2.1. System Design

Figure 1 illustrates the required signal-processing steps with spectral sketches. Signals have “=” on the left, and sample rates are marked with triangular tics below the axis. Operations on signals are marked as spectral products (filters) or spectral convolutions and have input and output sample rates marked with tics above and below the axis respectively. (The derivation strategy and this notation are presented in [1]. Another IQ-demodulator design is detailed in [2].)

This work was supported by the Office of Naval Research (ONR) Base Program at NRL.

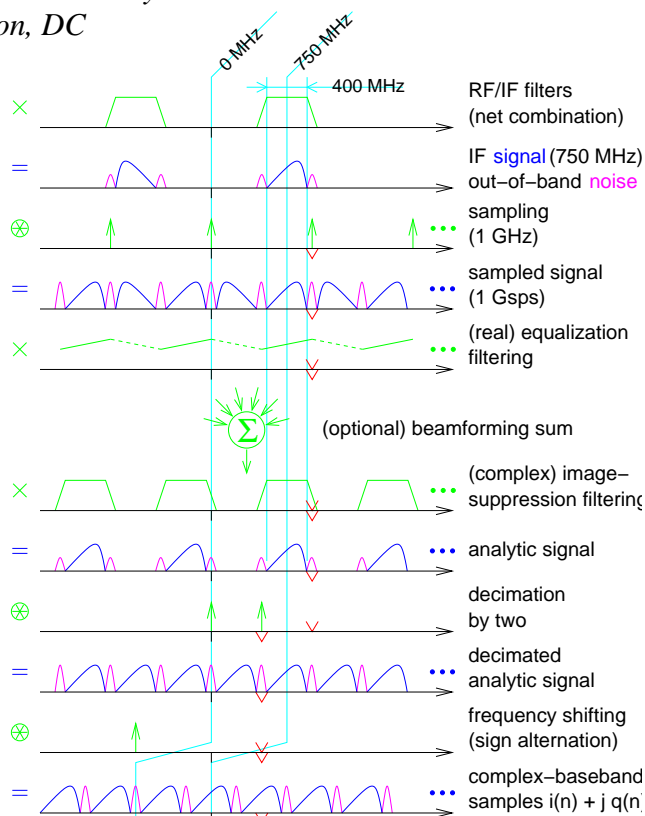


Figure 1: Signal-processing steps required.

Preprocessing comprises RF/IF filtering, sampling, and equalization of the RF/IF filters. Sampling aliases 100 MHz transition bands together and out of band. A “beamforming sum” follows in certain multi-channel systems only [3, 4] (else ignore). Then actual IQ downconversion begins.

Downconversion is simple. A digital image-suppression filter removes signal components originally at negative frequencies, then decimation halves the sampling rate and thus the computation rate of the preceding filter. The original 750 MHz

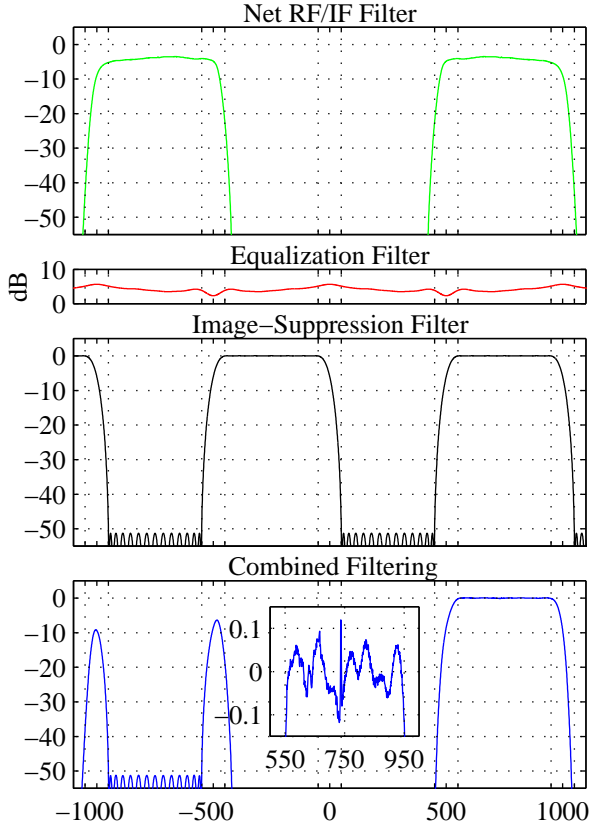


Figure 2: Filter magnitude responses.

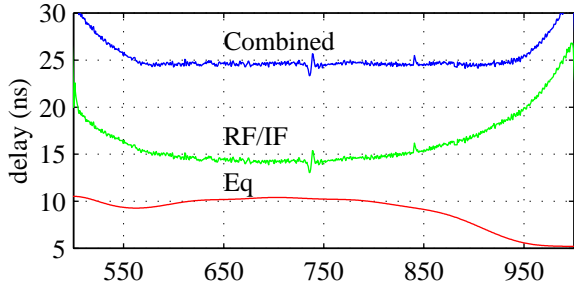


Figure 3: Filter group-delay responses.

signal lobe is shifted down in frequency to become complex baseband output signal $i(n) + jq(n)$. The complex exponential required is just $(-1)^n$, so the multiplication is just sign alternation.

2.2. Filter-Coefficient Design

Figures 2 and 3 show the responses of the actual filters, all referred to the sampler input.

The linear-phase, halfband image-suppression filter's coefficients were designed by starting with $-j$ times a length-14 equiripple Hilbert filter, zero interpolating by two, replacing the zero center coefficient by unity, and halving the result. This gave an impulse response of length 27, a unit-length real part, and an odd imaginary part with seven nonzero coefficients on either side of center. The image-suppression filter's passband was implied by its stopband, and the latter was just the Hilbert filter's 400 MHz passband at 250 MHz.

Combined passbands of the image-suppression filter, the RF/IF filters (measured), and the length-14 nonlinear-phase real FIR equalization filter approximate, by optimization of the latter, a pure delay [5] with -37 dB of minimized rms error.

3. IMPLEMENTATION

3.1. Polyphase FIR-Filter Structures

Table 1 represents polyphase (block) processing clocked at one eighth the sample rate. The x 's represent the sample stream $x(n), x(n-1), \dots$ input so far to an FIR filter, and the y rows represent filter outputs $y(n), y(n-1), \dots$. Numbers in the table are coefficient indices for required terms, so the second y row, for example, means $y(n-1) = c_9x(n-1) + c_8x(n-2) + c_7x(n-3) + \dots + c_8x(n-17) + c_9x(n-18)$. The even coefficient symmetry of this length-19 real filter gives it linear phase. Conjugate symmetry would do so for a complex filter: $y(n-1) = c_9^*x(n-1) + \dots + c_9x(n-18)$.

Either classic form in Fig. 4 can compute an output block from current and past input blocks using the Table 1 coefficient alignment. Computation is at eight-sample intervals, so each eight-sample delay z^{-8} is just a single clock tick, a one-register delay. Writing $k = 8m + r$, with r a residue modulo 8, permits output term $cx(n-k)$ to be realized by delaying $x(n-r)$, element r of the input block, by m clock ticks (register delays). Scaling by c follows the register delays in the direct form but precedes them in the transposed.

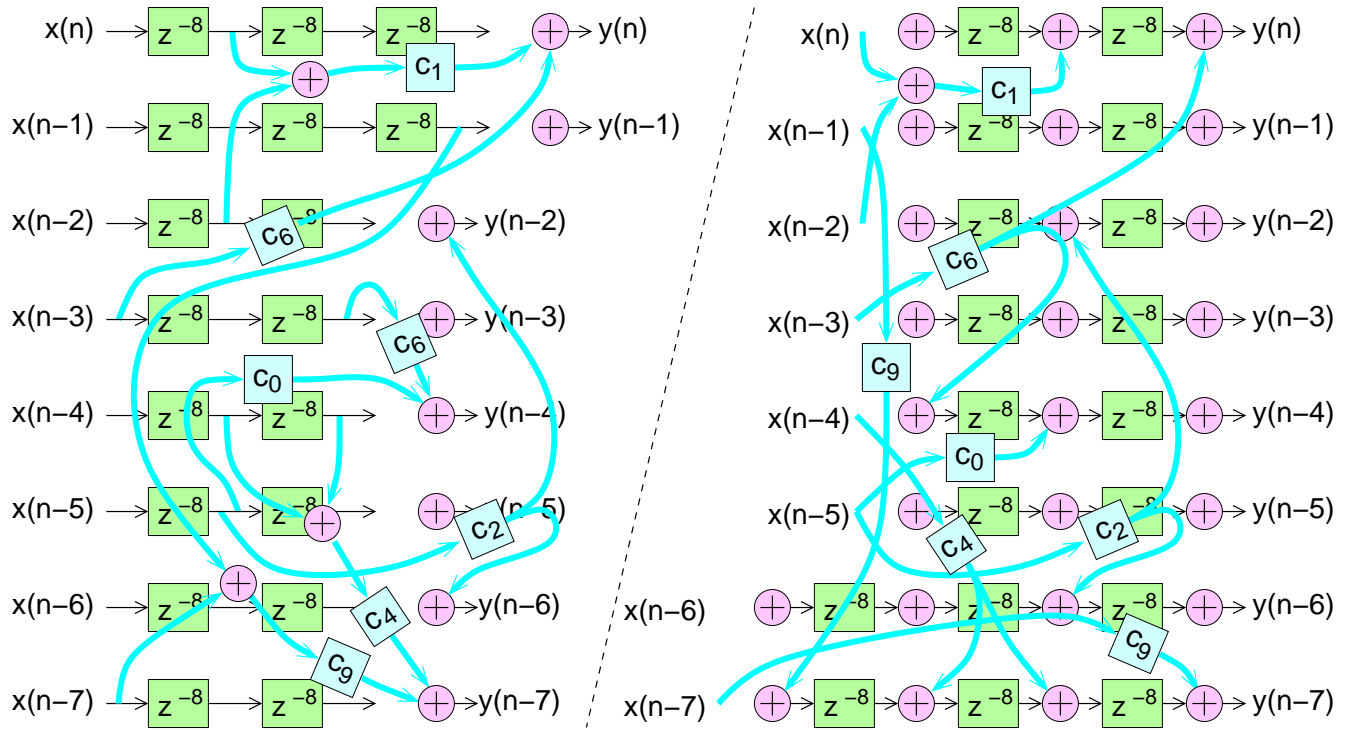


Figure 4: Direct-form (left) and transposed-form (right) polyphase structures with example output terms.

3.2. Savings from Filter Symmetry

Showing all the terms of Table 1 would make Fig. 4 unreadable. The terms shown (indicated in blue), the same for each form, show the several available structures for shared coefficient scaling in a linear-phase filter to save computation. (The negation in certain symmetries is not shown.) When inputs to a coefficient-sharing structure in the direct form are from the same “column” (register delay) of delayed inputs, as for c_1 , the structure is the same in the transposed form. The c_0 term is a degenerate case. Likewise, when a scaled input in the transposed form drives outputs through sums in the same column, as with c_2 , the structure is the same in the direct form. Inputs from the same input-data row in the direct form, as for c_4 , become multiple outputs in the transposed form (hinting at duality). When coefficient-scaling inputs in the direct form share neither row nor column, as with c_9 , there is no shared structure in the transposed form. When the coefficient-scaling output in the transposed form drives sums sharing

current	xxxxxxxx	xxxxxxxx	xxxxxxxx	xx
y	98765432	10123456	789	
y	9876543	21012345	6789	
y	987654	32101234	56789	
y	98765	43210123	456789	
y	9876	54321012	3456789	
y	987	65432101	23456789	
y	98	76543210	12345678	9
y	9	87654321	01234567	89

Table 1: Computing a linear-phase FIR filter in terms of eight polyphase components.

neither row nor column, as with c_6 , there is no shared structure in the direct form. The choice of a set of shared structures to realize a filter is seldom unique.

3.3. Lookup-Table Scaling by Coefficients

Lookup tables (LUT) are easily built as in Fig. 5 from the latched four-input logic blocks that are paired as “slices” in Xilinx’s Virtex FPGAs [6]. Other FPGAs are similar. So assume that LUTs,

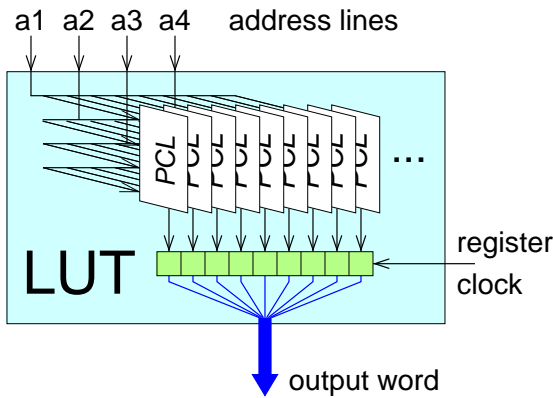


Figure 5: Programmable Combinatorial Logic and a register form a LookUp Table of any width.

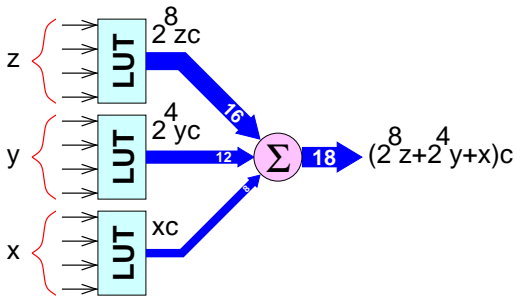


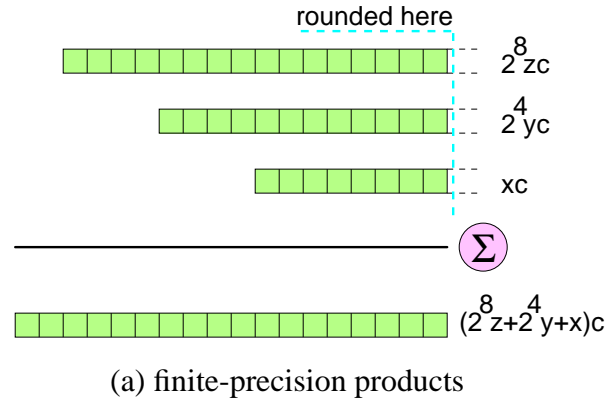
Figure 6: Scaling by a coefficient c .

adders, and registers are available in all widths.

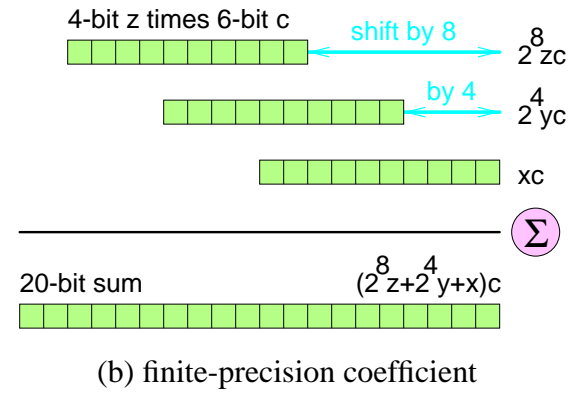
Figure 6 shows LUT scaling of data. Storing the required product, the bottom LUT would suffice for four-bit input. For wider inputs, four-bit pieces are scaled and results summed. Infinite-precision coefficients require different LUTs and rounded products, as in Fig. 7-(a), while finite-precision coefficients use identical LUTs and shifted output words as in Fig. 7-(b). Product output xc can be narrower if it is rounded along with the final sum (case not shown).

3.4. Looking Up Sums of Products

Without linear-phase coefficient sharing in Fig. 4, each transposed-form sum adds scaled versions of some or all of the eight inputs. In Fig. 8 such a sum requires just one copy per input-word bit of each of two distinct LUTs. Summing these two LUT outputs and then shifting and adding those totals according to bit position is not shown.



(a) finite-precision products



(b) finite-precision coefficient

Figure 7: Two approaches to precision.

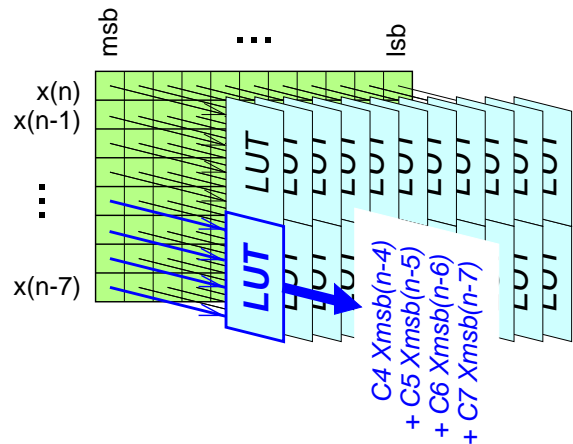


Figure 8: Linearly combining eight data words of arbitrary length with no unused LUT inputs.

An LUT's inputs need not come from just one data word as in Fig. 6 nor from just one bit position as in Fig. 8. In Fig. 9, six 10-bit words are linearly combined with 15 LUTs that do neither. The one-to-one mapping of data-input bits to LUT

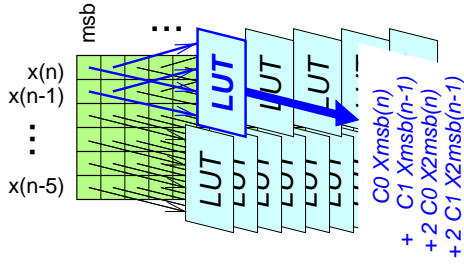


Figure 9: A given LUT can be driven with any four bits involved in the linear combination.

inputs is in general arbitrary, but minimizing the range of bit positions driving a single LUT, as in Fig. 9, minimizes required LUT output width.

Sign considerations were omitted from this paper for simplicity, but they are straightforward. Scaling a complex input simply requires LUTs driven by both real and imaginary parts of the input data. In general twice as many input bits to the LUT system are needed. Complex outputs require LUTs of twice the width, with half the width dedicated to the real part of the output and half dedicated to the imaginary part.

3.5. Pipeline Registers

For maximum FPGA clock speed, as required in this design, pipeline registers are required after all additions. Each programmable block of combinatorial logic is paired in the FPGA itself with a latch (in the Virtex series, for example) as in Fig. 5, so this is quite natural. But the registers of the direct form in Fig. 4 are not driven by adders (except possibly at the $x(n-k)$ inputs) and so may require wasting the logic paired with the latches. Alternatively, many of the pipeline registers associated with the necessary additions double as the z^{-8} delays in the transposed form, while the latter delays are distinct in the direct form. For the same reason, the overall latency of the transposed form may be lower. Such comparisons are complicated by the very different computational sharing and LUT configuration options in the two forms.

4. SUMMARY

While the full range of design considerations for a system such as this is beyond the scope of a short paper, we outlined here an approach, methods, and techniques with which a design of a wideband IQ downconverter or similar high-speed, multi-rate DSP system can be implemented in an FPGA using lookup-table approaches and a polyphase architecture.

REFERENCES

- [1] J. O. Coleman, "Multi-rate DSP before discrete-time signals and systems," in *Proc. First IEEE Workshop on Signal Processing Education (SPE 2000)*, Hunt TX, Oct. 2000.
- [2] Dan P. Scholnik and J. O. Coleman, "Integrated I-Q demodulation, matched filtering, and symbol-rate sampling using minimum-rate IF sampling," in *Proc. 1997 Symp. on Wireless Personal Communication*, Blacksburg, VA, June 1997.
- [3] J. J. Alter, M. G. Parent, J. O. Coleman, D. P. Scholnik, and F. J. Caherty, "A wideband digital beamformer using true time delay," in *Proc. Military Sensing Symp. (MSS99)*, North Charleston SC, Nov. 1999.
- [4] J. J. Alter, J. O. Coleman, M. G. Parent, J. P. McConnell, and D. P. Scholnik, "An FPGA-based wideband digital beamformer using true time delay," in *Proc. Workshop on Beamforming Technology and Applications*, Huntsville AL, July 2000.
- [5] D. P. Scholnik and J. O. Coleman, "Computationally efficient multirate passband equalization for bandpass digital/analog conversion," in *Proc. 1999 Midwest Symp. on Circuits and Systems (MWSCAS '99)*, New Mexico State University, Aug. 1999.
- [6] Xilinx, Inc., "FPGA data book 2000," <http://support.xilinx.com/partinfo/databook.htm>.