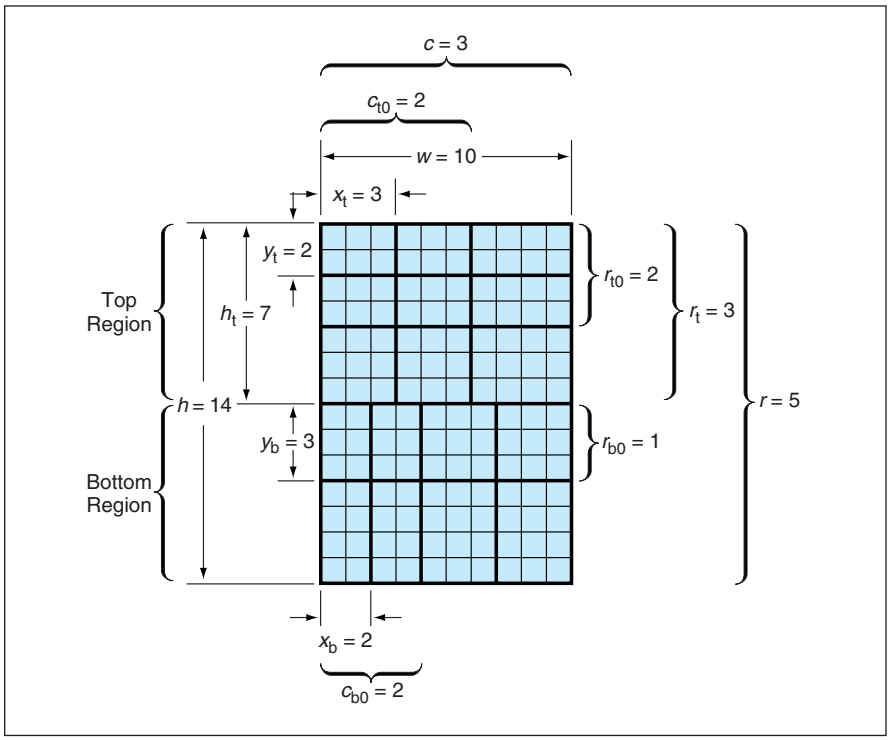



A **Rectangular Image** of 10 by 14 pixels is partitioned into 17 segments by the algorithm described in the text.

pixel units. The limit on $s$ for a valid result is given by $s \le wh$.

The output of the algorithm is characterized by, and can be described completely in terms of, several parameters that are illustrated by the example shown in the figure. The segments are arranged in $r$ rows. A top region of the rectangle contains segments arranged in $c$ columns. A possible bottom region contains segments arranged in $c + 1$ columns. The top region has height $h_t$ and contains $r_t$ rows of segments. The first $r_{t0}$ rows of segments in the top region have height $y_t$; the remaining rows (if any) in the top region have height $y_t + 1$. The first $c_{t0}$ columns in the top region have width $x_t$; any remaining columns in the top region have width $x_t + 1$. Similarly, the first $r_{b0}$ rows in the bottom region have height $y_b$, and the remaining rows in the bottom region have height $y_b + 1$, while the first $c_{b0}$ columns in the bottom region have width $x_b$ and the remaining columns in the bottom region have width $x_b + 1$.

The steps of the algorithm are the following: First, $r$ is computed. If $h > (s-1)w$ then $r = s$. Otherwise, $r$ is the unique positive integer that satisfies

$$(r-1)rw < hs \le (r+1)rw.$$

Some of the other parameters are computed as follows:

$$c = \left\lfloor \frac{s}{r} \right\rfloor$$
$$r_t = (c+1)r - s$$
$$h_t = \max\left( r_t, \left\lfloor \frac{hcr_t}{s} + \frac{1}{2} \right\rfloor \right)$$
$$x_t = \left\lfloor \frac{w}{c} \right\rfloor$$
$$c_{t0} = (x_t + 1)c - w$$
$$y_t = \left\lfloor \frac{h_t}{r_t} \right\rfloor$$
$$r_{t0} = (y_t + 1)r_t - h_t.$$

If $r_t < r$, so that there is a bottom region, then the remaining parameters are computed as follows:

$$x_b = \left\lfloor \frac{w}{c+1} \right\rfloor$$
$$c_{b0} = (x_b + 1)(c+1) - w$$
$$y_b = \left\lfloor \frac{h - h_t}{r - r_t} \right\rfloor$$
$$r_{b0} = (y_b + 1)(r - r_t + 1) - (h - h_t).$$

It has been verified by straightforward algebraic analysis of these equations that the algorithm has the properties mentioned in the first paragraph.

*This work was done by Matthew Klimesh and Aaron Kiely of Caltech for* **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

*This software is available for commercial licensing. Please contact Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-30479.*

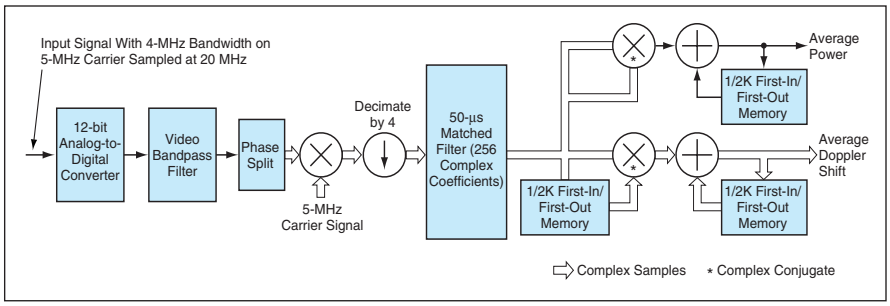
# Digital Radar-Signal Processors Implemented in FPGAs

## Processing can be performed onboard at relatively low power.

*NASA's Jet Propulsion Laboratory, Pasadena, California*

High-performance digital electronic circuits for onboard processing of return signals in an airborne precipitation-measuring radar system have been implemented in commercially available field-programmable gate arrays (FPGAs). Previously, it was standard practice to downlink the radar-return data to a ground station for postprocessing — a costly practice that prevents the nearly-real-time use of the data for automated targeting. In principle, the onboard processing could be performed by a system of about 20 personal-computer-type microprocessors; relative to such a system, the present FPGA-based processor is much smaller and consumes much less power. Alternatively, the onboard processing could be performed by an application-specific integrated circuit (ASIC), but in comparison with an ASIC implementation, the present FPGA implementation offers the advantages of (1) greater flexibility for research applications like the present one and (2) lower cost in the small production volumes typical of research applications.

The generation and processing of signals in the airborne precipitation-




The **Signal-Processing Functions** of one of the four channels require $5 \times 10^9$ multiplications per second.

measuring radar system in question involves the following especially notable steps:

- The system utilizes a total of four channels — two carrier frequencies and two polarizations at each frequency.
- The system uses pulse compression: that is, the transmitted pulse is spread out in time and the received echo of the pulse is processed with a matched filter to despread it.
- The return signal is band-limited and digitally demodulated to a complex baseband signal that, for each pulse, comprises a large number of samples.
- Each complex pair of samples (denoted a range gate in radar terminology) is associated with a numerical index that corresponds to a specific time offset from the beginning of the radar pulse, so that each such pair represents the energy reflected from a specific range. This energy and the average echo power are computed.
- The phase of each range bin is compared to the previous echo by complex conjugate multiplication to obtain the mean Doppler shift (and hence the mean and variance of the velocity of precipitation) of the echo at that range.

The processing for each of the four channels (see figure) requires $>5 \times 10^9$ multiplications per second — well beyond the capabilities of traditional microprocessors. The design effort involved the application of some algorithmic tricks, careful planning of the allocation of the areas on the FPGA to the various processing functions, and exploitation of the high circuit density and performance of the commercially available FPGAs chosen for this application. The design has made it possible to perform all the processing required by the radar system on two FPGAs — each one handling the data for two of the four channels. The algorithmic tricks and other design techniques used here could be applied to FPGA implementations of other signal-processing systems in other applications in radar, general imaging, and communications.

*This work was done by Andrew Berkun and Ray Andraka of Caltech for* **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*
*NPO-30517*