Economical Implementation of a Filter Engine in an FPGA

There are numerous potential uses in general signal processing.

NASA's Jet Propulsion Laboratory, Pasadena, California

A logic design has been conceived for a field-programmable gate array (FPGA) that would implement a complex system of multiple digital state-space filters. The main innovative aspect of this design lies in providing for reuse of parts of the FPGA hardware to perform different parts of the filter computations at different times, in such a manner as to enable the timely performance of all required computations in the face of limitations on available FPGA hardware resources.

A digital state-space filter of the type to which the design applies (see figure) is represented by the equations

$$\mathbf{x}(k + 1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) \qquad (1)$$

and

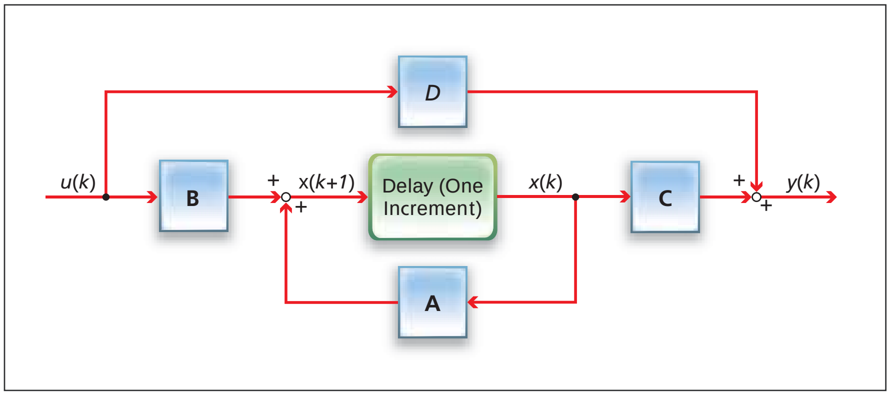
$$y(k) = \mathbf{C}x(k) + Du(k) \qquad (2)$$

where:

k is an integer representing an increment of time,
N is an integer representing the number of dimensions in the state space,
$u(k)$ is a scalar input at time k,
$y(k)$ is a scalar output at time k,
$\mathbf{x}(k)$ is an N×1 state vector at time k,
$\mathbf{A}$ is an N×N matrix of filter coefficients,
$\mathbf{B}$ is an N×1 vector of filter coefficients,
$\mathbf{C}$ is a 1×N vector of filter coefficients, and
D is a scalar filter coefficient.

The implementation of the digital state-space filter involves matrix·vector multiplications, which, in the absence of the present innovation, would ordinarily necessitate some multiplexing of vector elements and/or routing of data flows along multiple paths.

The design concept calls for implementing vector registers as shift registers to simplify operand access to multipliers and accumulators, obviating both multiplexing and routing of data along multiple paths. Each vector register would be reused for different parts of a calculation. Outputs would always be drawn from the same register, and inputs would always be loaded into the same register.



This **Block Diagram** represents a digital state-space filter, which could be one of multiple such filters in a complex system. The same FPGA circuitry could be used to implement different filters at different times.

A simple state machine would control each filter. The output of a given filter would be passed to the next filter, accompanied by a "valid" signal, which would start the state machine of the next filter. Multiple filter modules would share a multiplication/accumulation arithmetic unit. The filter computations would be timed by use of a clock having a frequency high enough, relative to the input and output data rate, to provide enough cycles for matrix and vector arithmetic operations.

This design concept could prove beneficial in numerous applications in which digital filters are used and/or vectors are multiplied by coefficient matrices. Examples of such applications include general signal processing, filtering of signals in control systems, processing of geophysical measurements, and medical imaging. For these and other applications, it could be advantageous to combine compact FPGA digital filter implementations with other application-specific logic implementations on single integrated-circuit chips. An FPGA could readily be tailored to implement a variety of filters because the filter coefficients would be loaded into memory at startup.

This work was done by James E. Kowalski of Caltech for NASA's Jet Propulsion Laboratory. Further information is contained in a TSP (see page 1). *NPO-44662.*