# Electronics/Computers

## MOSFET Switching Circuit Protects Shape Memory Alloy Actuators

*John H. Glenn Research Center, Cleveland, Ohio*

A small-footprint, full surface-mount-component printed circuit board employs MOSFET (metal-oxide-semiconductor field-effect transistor) power switches to switch high currents from any input power supply from 3 to 30 V.

High-force shape memory alloy (SMA) actuators generally require high current (up to 9 A at 28 V) to actuate. SMA wires (the driving element of the actuators) can be quickly overheated if power is not removed at the end of stroke, which can damage the wires.

The new analog driver prevents overheating of the SMA wires in an actuator by momentarily removing power when the end limit switch is closed, thereby allowing complex control schemes to be adopted without concern for overheating. Either an integral pushbutton or microprocessor-controlled gate or control line inputs switch current to the actuator until the end switch line goes from logic high to logic low state. Power is then momentarily removed (switched off by the MOSFET). The

analog driver is suited to use with nearly any SMA actuator.

*This work was done by Mark A. Gummin of Miga Motor Company for Glenn Research Center. Further information is contained in a TSP (see page 1).*

*Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Innovative Partnerships Office, Attn: Steven Fedor, Mail Stop 4–8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-18581-1.*

## Optimized FPGA Implementation of Multi-Rate FIR Filters Through Thread Decomposition

### This technique is used in design automation and in digital circuit design.

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

Multi-rate finite impulse response (MRFIR) filters are among the essential signal-processing components in spaceborne instruments where finite impulse response filters are often used to minimize nonlinear group delay and finite-precision effects. Cascaded (multistage) designs of MRFIR filters are further used for large rate change ratio in order to lower the required throughput, while simultaneously achieving comparable or better performance than single-stage designs. Traditional representation and implementation of MRFIR employ polyphase decomposition of the original filter structure, whose main purpose is to compute only the needed output at the lowest possible sampling rate.

In this innovation, an alternative representation and implementation technique called TD-MRFIR (Thread Decomposition MRFIR) is presented. The basic idea is to decompose MRFIR into output computational threads, in contrast to a structural decomposition of the original filter as done in the polyphase decomposition. A naïve implementation

of a decimation filter consisting of a full FIR followed by a downsampling stage is very inefficient, as most of the computations performed by the FIR state are discarded through downsampling. In fact, only $1/M$ of the total computations are useful ($M$ being the decimation factor). Polyphase decomposition provides an alternative view of decimation filters, where the downsampling occurs before the FIR stage, and the outputs are viewed as the sum of $M$ sub-filters with length of $N/M$ taps. Although this approach leads to more efficient filter designs, in general the implementation is not straightforward if the numbers of multipliers need to be minimized.

In TD-MRFIR, each thread represents an instance of the finite convolution required to produce a single output of the MRFIR. The filter is thus viewed as a finite collection of concurrent threads. Each of the threads completes when a convolution result (filter output value) is computed, and activated when the first input of the convolution becomes available. Thus, the new threads get spawned at exactly the rate

of $N/M$, where $N$ is the total number of taps, and $M$ is the decimation factor. Existing threads retire at the same rate of $N/M$. The implementation of an MRFIR is thus transformed into a problem to statically schedule the minimum number of multipliers such that all threads can be completed on time. Solving the static scheduling problem is rather straightforward if one examines the Thread Decomposition Diagram, which is a table-like diagram that has rows representing computation threads and columns representing time. The control logic of the MRFIR can be implemented using simple counters. Instead of decomposing MRFIRs into sub-filters as suggested by polyphase decomposition, the thread decomposition diagrams transform the problem into a familiar one of static scheduling, which can be easily solved as the input rate is constant.

*This work was done by Kayla N. Kobayashi, Yutao He, and Jason X. Zheng of Caltech for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov. NPO-47409*