Hardware Implementation of a Digital Anti-aliasing Filter

David Conell
Gulton Data Systems
6600 Gulton Court N.E.
Albuquerque NM 87109
505-345-9031

ABSTRACT

This paper presents a practical application of a realtime, in-flight programmable digital filter. This filter consists of one module in a PCM system consisting of a central unit and one or more remotes. The paper will also discuss how filtering is achieved given that the PCM format is also user programmable.

BACKGROUND

Historically in data acquisition systems, filtering has been part of the signal conditioning circuitry. When this circuitry is brought into the PCM package the need for filtering usually does not disappear. Since the size and power requirements of the analog filter components are large with respect to other PCM components, it is often deleted. This is particularly true for low-level signals which typically require an amplifier per channel.
A digital filtering approach, where the filtering hardware is time shared by all inputs and is programmable by the user, is particularly appropriate since any analog filter function can be duplicated or improved on in digital form.

FEATURES

Time sharing allows one filter arithmetic system to support all analog channels of a data acquisition system, whether they be located in a central or remote unit. By integrating it into the PCM system a considerable economy results. A single port for loading the PCM format exists which controls both the typical format parameters (such as sample rate, gain, offset, etc) as well as the filtering functions to be performed by the format. The option of not filtering a given measurement is also programmable. Multiple filter types are programmed by the selection of filter coefficients. Once determined these filters may be time shared among any set of

inputs or a set of filters may be applied to a given input. Although low-pass filtering is the usual filter effected, any type of filter can be designed by selection of filter coefficients. The filter employed is a digital finite impulse response filter. The filter is characterized by the coefficients which have been derived by using the fourier series method and applying iterative windowing techniques to achieve equal ripple in the stop-band.

IMPLEMENTATION CONCEPTS

Inherent in the synthesis of low-pass digital filters is the concept of taking more input samples than are output in the PCM format. The implication of this for PCM control systems is to abandon the single instruction/single sample per PCM output word architecture. In its place a control architecture has been devised which allows for multiple instructions/multiple samples per PCM output word to support the various input/output (I/O) ratios of the filter.

Finite impulse response filters are employed where the filter characteristics are determined by the multiplicative coefficients. Greater stop-band width is possible by increasing the filter length (the I/O ratio). The filter as designed supports filter lengths from 24 to 224 (I/O ratios from 3 to 28) and provides for 256 filtered measurements.

OPERATION CONCEPTS

All sampling and filtering operations are initiated by the control system. Attendant to each sample request is a filter or no filter flag. When there is a filter flag a subsequent instruction results in the appropriate filter type and channel selection. The detailed arithmetic operations are controlled by a set of micro-code PROMs resident on the filter board. Once the filter is addressed by the control system, a high speed clock sequences through the PROM addresses whose outputs provide the necessary controls for the filter. These include the multiplier-accumulator, coefficient PROMS, and data storage RAMs.

DIGITAL FILTER STRUCTURE

The filter’s major components consist of two 8kx8 RAMs to store the results of partially calculated difference equations, a multiplier-accumulator (MAC) to perform the required arithmetic, two 2kx8 PROMs to hold the coefficients, an ASIC device for communication on the

internal address and data buses, and three 512x8 PROMs to serve as micro-controllers (refer to figure 1).

The main bus in the filter exists between the RAMs and the MAC. This bus is 16 bits wide and is bidirectional and is referred to as the RAM bus.

Input Data

The analog data from the digitized sampled is taken from the internal 8 bit data bus and held in the input latch. The digitized data ranges from all zeroes to all ones. Since the MAC operates on twos complement numbers, the data is treated as positive fractions. The data occupies the immediate eight bit positions in lesser significance to the binary point. The digital value greater than the binary point is fixed at a logic zero, thus making the data a positive fraction. The remaining least significant bits (LSBs) are fixed at logic zeroes to fill out one of the 16 bit input fields to the MAC.

Coefficient Data

All filter coefficients are stored in local PROM devices. These numbers are both positive and negative fractions. The PROMs are organized as a 2kxl6 device and provide the other 16 bit input field to the MAC.

Accumulator/RAM Data

The RAM data contains the product of the coefficient and the sampled analog data. This is read from the accumulator output by the RAMs and is also a bipolar fraction. The accumulator output is truncated after the 2**-12 bit position and stored in RAM. The remaining 3 bits of the RAM are used as an index for addressing the RAM itself. Since the MAC is a 16xl6 device the accumulator is designed with a 36 bit field; 16 bits in each the most and least significant product and 4 bits in an extended product. The accumulator is input from the RAM using 13 bits from a previous calculation and is placed in the 13 most significant bits of the most significant product field. All other bits of the accumulator are input as zero.

Bus Interface Device

This is an ASIC device used by all cards in the data acquisition system. This interface provides the communication path from the system controller to the filter

via the internal address bus. The system controller provides the filter with the appropriate pointers to RAM and PROM address spaces of the filter for the operation. Data is transmitted to and received from the filter via the internal data bus.

FILTER OPERATION

The architecture of the filter provides 8 RAM registers for each channel in the digital filter. Each of these registers represents a partial accumulation of the difference equation used by that channel. However, each register (each difference equation) represents a partial accumulation that is unique; that is each register has started it’s computation of the difference equation at different times. Each time the filter is addressed it multiplies the input sample by a set of 8 coefficients and adds the result to the RAM register. This set of 8 coefficients is a subset of all the coefficients necessary to effect the difference equation. As the filter is addressed the subset selected sequences through the set of coefficients defining the filter. The 8 RAM register architecture requires that there be coefficient subsets, each containing 8 coefficients. Thus the length of the filter (N) is a multiple of N=n*8, where n is the I/O ratio.

PROM/RAM Rotation

Figure 2 displays the RAM registers for one input sample. The registers are identified as Rl through R8 and the contents are displayed horizontally. Time progresses left to right in the figure such that at any given time the RAM contents are the accumulation of all products that appear to the left. The order of coefficients is from A16 to Al for an N=16 filter. Once an input is multiplied by the Al coefficient the difference equation is complete for that register and it is available for output to the PCM on the subsequent access. In this figure it can be seen that the 16 coefficients are divided into two groups (even and odd). As time progresses and each subset of coefficients is accessed, the order of the members of the subset are rotated. This technique is known as PROM rotation.

While PROM rotation is conceptually straightforward, it is difficult to effect without maintaining an index register for each set of 8 coefficients. This implementation problem is solved by using RAM rotation and using 3 bits of the RAM contents to serve as the index for the set of 8 RAM registers which exist for each filter. Figure 3 shows such a

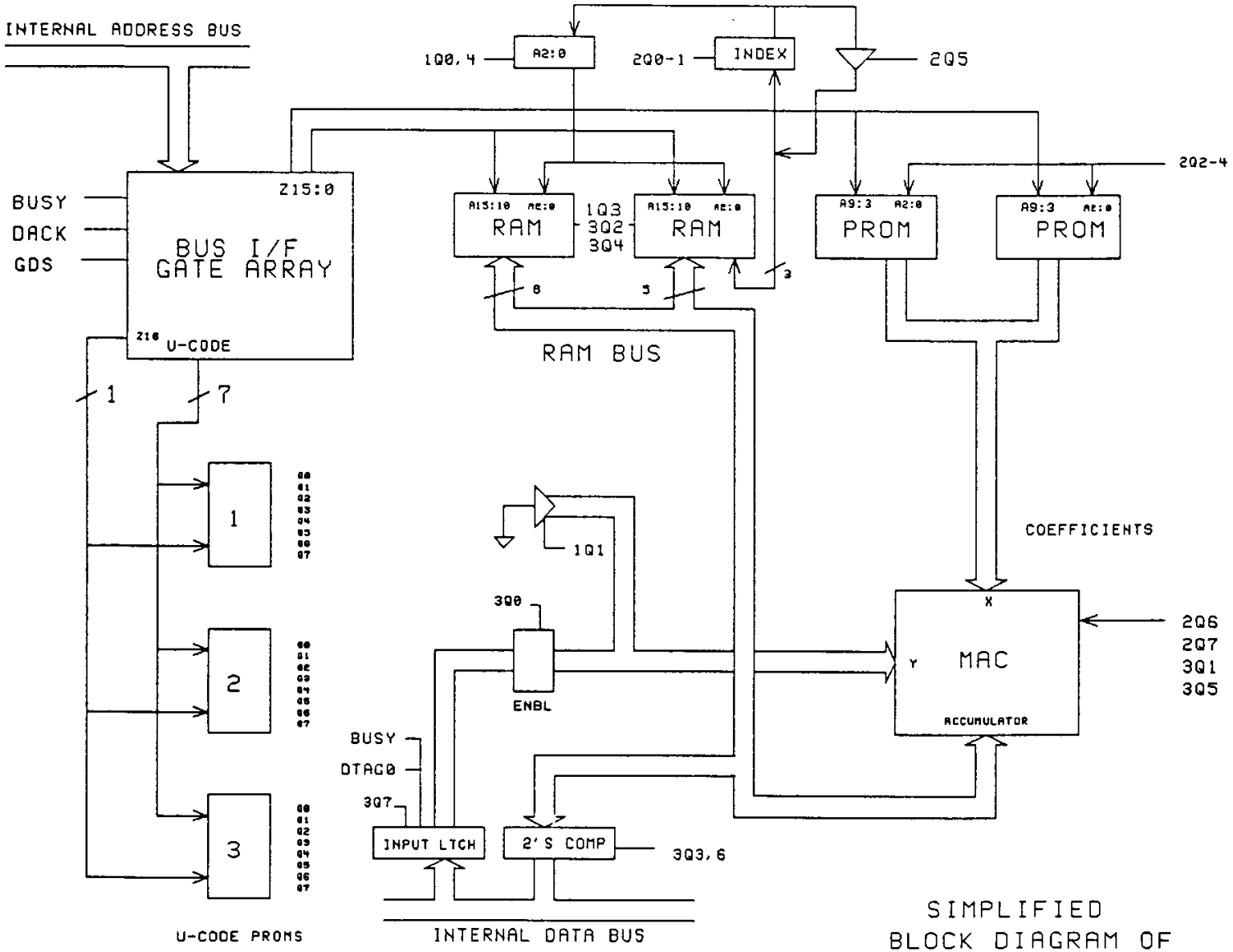
RAM rotation technique. In this figure each of the 2 subsets of coefficients are stationary and the RAM registers are rotating. When R2 is configured to be at the top of the register stack and sample X3 is present, the register R3 has completed it’s difference equation calculation and it is available for output to the PCM. For each filter channel the index of rotation is maintained in that register. When the channel is output the register is cleared and the index is incremented by one. This index is maintained in all 8 register locations. When the control system provides pointers to the filter indicating which channel and subset of coefficients are to be used, these pointers will direct the hardware to Rl of the register stack. At that time the index is determined for this register stack and the multiply-accumulate sequence is begun with the register stack pointer appropriately shifted.

MAC Cycles

The MAC device provides the arithmetic necessary to effect the filter. The contents of the RAM register are written to the accumulator, then the input sample and the coefficient are written to the X and Y inputs of the multiplier. The product formed by the multiplier is accumulated with the contents of this register and the new data is written back to RAM. The RAM and PROM pointers are incremented by one and the multiply-accumulate process is repeated 7 more times. When an output is available the microcode directs data from the RAM bus to the twos complement latch.

Filter Output

Since the filter input is a positive fraction the output of the filter is a positive fraction. The twos complement register is latched from the accumulator output whenever a difference equation calculation is complete. This register occupies the bit positions as the input sampled data. The implication of this is that the filter output is mag(H(w)) rather than H(w). Once data is restored to a positive fraction by this latch, it is transmitted to the PCM via the internal data bus.



SIMPLIFIED
BLOCK DIAGRAM OF
DIGITAL FILTER
FIGURE 1

PROM ROTATION

* = OUTPUT

N = 16

2/1 SAMPLING RATIO

SAMPLES	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	. . .
RAM	* OUTPUT		* OUTPUT		* OUTPUT		* OUTPUT		* OUTPUT		* OUTPUT		
R1	a_{16} *	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_{9}	a_{8}	a_{7}	a_{6}	a_{5}	
R2	a_{2}	a_{1}	a_{16} *	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_{9}	a_{8}	a_{7}	
R3	a_{4}	a_{3}	a_{2}	a_{1}	a_{16} *	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_{9}	
R4	a_{6}	a_{5}	a_{4}	a_{3}	a_{2}	a_{1}	a_{16} *	a_{15}	a_{14}	a_{13}	a_{12}	a_{11}	. . .
R5	a_{8}	a_{7}	a_{6}	a_{5}	a_{4}	a_{3}	a_{2}	a_{1}	a_{16} *	a_{15}	a_{14}	a_{13}	
R6	a_{10}	a_{9}	a_{8}	a_{7}	a_{6}	a_{5}	a_{4}	a_{3}	a_{2}	a_{1}	a_{16} *	a_{15}	
R7	a_{12}	a_{11}	a_{10}	a_{9}	a_{8}	a_{7}	a_{6}	a_{5}	a_{4}	a_{3}	a_{2}	a_{1}	
R8	a_{14}	a_{13}	a_{12}	a_{11}	a_{10}	a_{9}	a_{8}	a_{7}	a_{6}	a_{5}	a_{4}	a_{3}	

THE OUTPUT EQUATION IS: $$y(n) = \sum_{k=1}^{16} a_k \, x(n-k+1)$$

FIGURE 2

X*Y+RAM

RAM ROTATION

N=16

2/1 SAMPLING RATIO

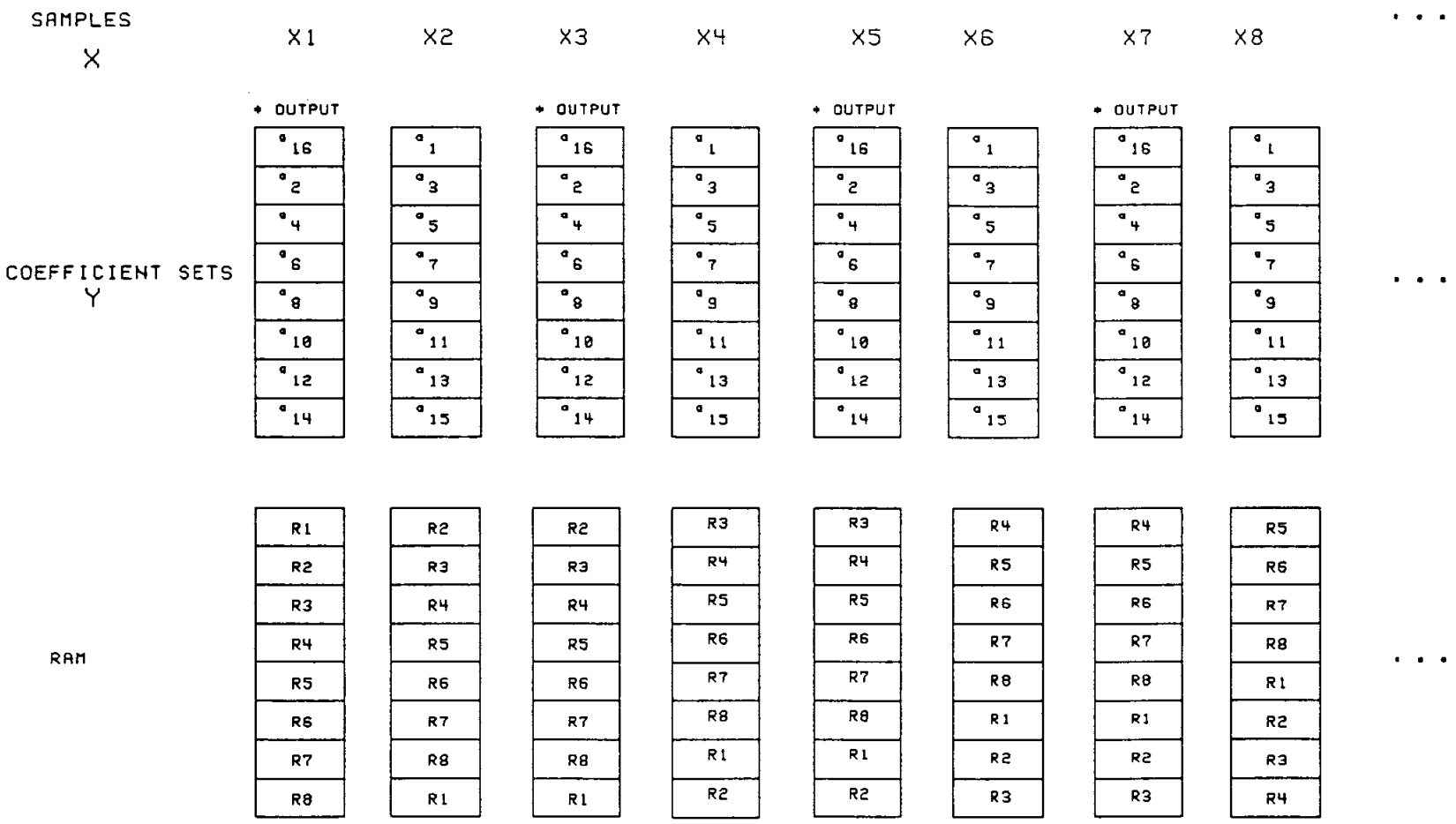


FIGURE 3