Versatile digital arithmetic unit with rams

A common method of implementing fast arithmetic circuits is to realise them as look-up tables using semiconductor read-only memories, but they are still expensive for the general user to purchase and programme. In addition, their contents cannot be altered to suit different operating parameters. Even with field-programmable and erasable roms, it still takes time to prepare the data paper tapes and to erase previously stored contents with an ultra-violet source.

A simple and efficient alternative makes use of random access read/write memories instead. As shown in Fig. 1, the ram, when operating in real time, is addressed by the signal from the environment and outputs the relevant word to it.

The ram is volatile, so that it has to have its contents written every time the system is switched on. But because the contents are computable, the ram is easily programmed using an operand simulator (os) and slow arithmetic unit (sau). The os generates all possible combinations of input values, and the sac, which is easily designed with conventional serial arithmetic techniques, computes the required arithmetic function.

As an example, a binary digital filter is shown in Fig. 2. It has four six-bit weighting coefficients, A_0, A_1, A_2 and A_3, and its output Y_n is given by:

$$Y_n = \sum_{k=0}^{3} x_{n-k} A_k \qquad (1)$$

A table look-up realisation of the portion of the filter which is enclosed by the broken lines in Fig. 2 would require a 16-word by eight-bit ram addressed by the binary vector (x_n, x_{n-1}, x_{n-2}, x_{n-3}).

The complete circuit is detailed in Fig. 3. The os is a simple four-bit counter type SN7493, while the sac is a four-word serial one-bit adder which is made up of a pair of full adders type SN74183 and an SN7482 two-bit adder. The filter output Y_n is computed as follows. Expressing the weighted coefficients in binary:

$$A_k = \sum_{j=0}^{5} a_{k,j} 2^j \qquad (2)$$

where $k = 0, 1, 2, 3$ and $a_{k,j} = 0$ or 1. By putting (2) in (1) and re-ordering the double summation:

$$Y_n = \sum_{j=0}^{5} \sum_{k=0}^{3} x_{n-k}\, a_{k,j} 2^j \qquad (3)$$

Thus coefficient bits of the same significance are added in one bit time, each bit $a_{k,j}$ being weighted by the relevant simulated data bit x_{n-k}, for every combination of the os output.

In operation, the weights A_0 to A_3 are entered serially, via the SN74157 two-to-one multiplexer, into the SN7491 eight-bit registers. This is done by connecting the programme clock line to S4, which is used as a manual clock. Each weight is padded by two zeroes following its most significant bit. S1 is set to one, the counters are reset to zero, and the ram address is now switched to the os output via S2. The programme clock line is reconnected to the clock, S4 set to one, and the clock is initiated.

The three-bit and four-bit counters and the associated *nand* logic are designed so that, after every eight clock pulses, the *write enable* of the ram is strobed to zero, writing in the relevant filter output which has meanwhile been computed. The next clock pulse brings the *write enable* back to one and clocks the os to a new four-bit address, with two *nand* gates between the counters preventing data being written into the wrong address. After another eight pulses the process is repeated.

The system has been designed to stop automatically after the os output has reached (1,1,1,1) and the necessary arithmetic corresponding to this address has been duly completed. S4 is now set to zero, disenabling the clock and the counters reset to zero, thus holding the *write enable* to one. After switching the ram address back to the environment input, the memory is now ready for real-time application.

Digital arithmetic units built with this technique are fast in operation, with a 30 ns data rate typical for the example given, simple and inexpensive, since rams are general purpose msi/lsi devices, and extremely versatile, since operand parameters may be altered quickly.

The slow arithmetic unit and the ram may be used in their more traditional roles when the system is not operating in the fast mode.

M. A. Bin Nun, Department of Electronics and Electrical Engineering, University of Technology, Loughborough, Leics.

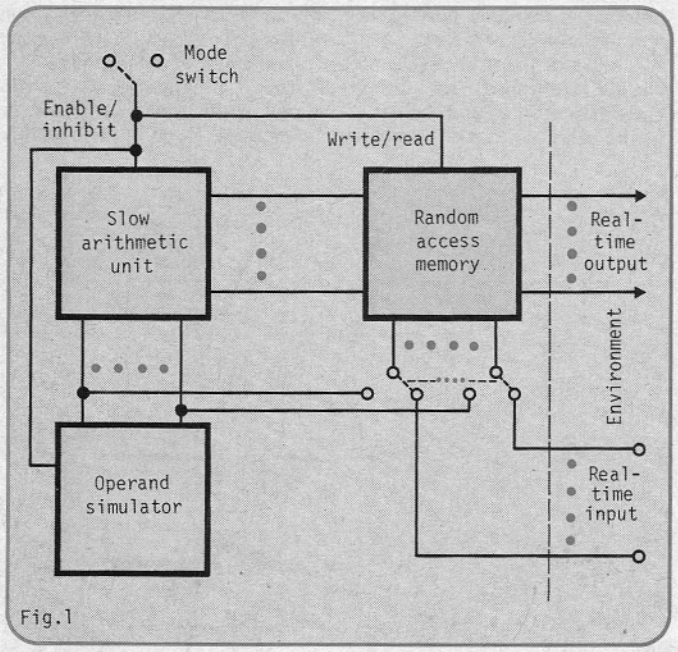

Fig.1

Fig. 1: Digital arithmetic unit.
Fig. 2: Binary digital filter with six-bit coefficients.
Fig. 3: Implementation of binary filter using a ram.

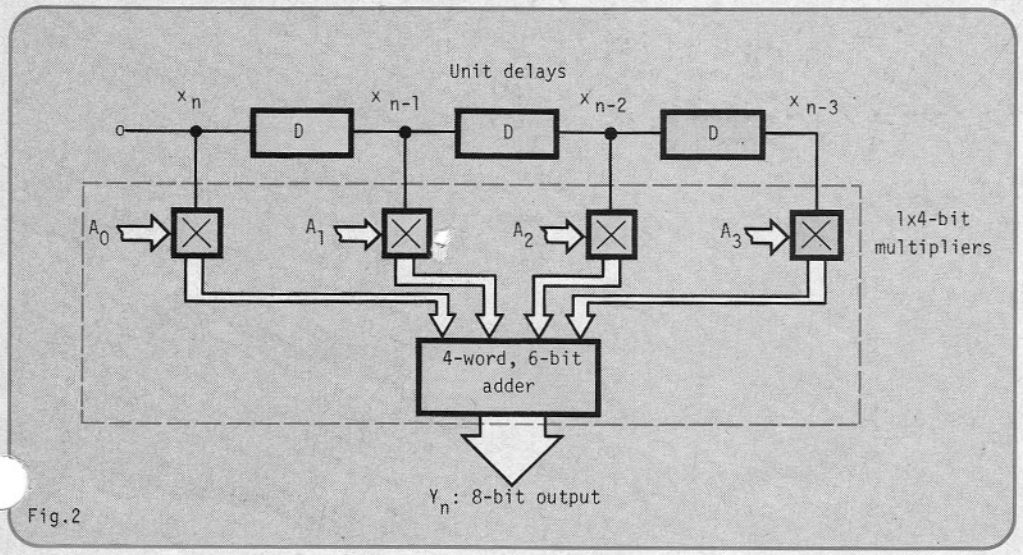

Fig.2