D60

7.4A HARDWARE SCHEMES FOR FAST FOURIER TRANSFORM

G. R. Stitt and S. A. Bowhill

Aeronomy Laboratory
Department of Electrical and Computer Engineering
University of Illinois, Urbana, Illinois 61801

Because of the interest in real-time FFT processing of a MST radar data, a study has been made of cost-effective approaches to hardware FFT generation. Results are summarized in Table 1. In this table, the first six entries represent previously devised hardware FFT configurations which have been described in the open literature, including the estimated number of chips used and the time required to perform a 1024-point FFT. The remaining entries in the table correspond to original designs, which presuppose the availability of a microcomputer -- an Apple II+ has been assumed -- and a modestly complicated hardware peripheral. These original designs, all of which implement a radix-4 FFT with twiddle factors, have been assigned model numbers to make them easier to refer to.

The Model 10 performs a complete FFT at one time, and is implemented using standard TTL chips. An Apple microcomputer is responsible for transferring data to a large set of input registers, and retrieving the transformed results from another set of output registers. The Model 20 is a much simpler design which calculates the value of a single butterfly output mode, and must therefore be used repeatedly by the microcomputer during the computation of a complete FFT. In order to improve throughput, the Model 30 essentially uses four Model 20s in parallel, so that all four output mode values of a single radix-4 butterfly are calculated simultaneously. Like the Model 30, the Model 40 processes an entire radix-4 butterfly at one time, but takes superior advantage of the inherent symmetry of the FFT algorithm by utilizing multiplexers and control ROMs to reduce the overall chip count.

All of the devices outlined above are inefficient from the standpoint that they require the Apple microcomputer to spend most of its time transferring data back and forth between external registers. Models 50 through 70 attempt to alleviate this problem by utilizing an external microcontroller, in these cases a Signetics 8X305. The Model 50 uses memory access (DMA) transfer of the data to be transformed to very fast external RAM, after which the microcontroller directs the flow of data between the external RAM and a Model 40. When the FFT is completed, the 8X305 DMAs the transformed results back into Apple RAM. The Model 60 eliminates the DMA steps, and 8X305 being used simply to transfer data back and forth between Apple RAM and the I/O registers of a Model 40. Lastly, the Model 70 is very similar to the Model 50, but uses a Motorola MC6844 Direct Memory Access Controller chip to perform the DMAs.

The specifications for the various processors shown in Table 1 are plotted on the graph of Figure 1. This graph indicates that those implementations which attain a relatively fast transformation time also tend to possess a relatively high chip count, regardless of the FFT algorithm chosen. Figure 1 also seems to indicate that once a particular FFT algorithm has been chosen, it may be implemented in a nearly endless variety of ways, each striking a different compromise between the characteristics of transform size and computation speed, and system size and complexity.

In Figure 1, a line has been included which possesses a slope of -1/2, and which passes through the point (1 chip, 3000 s); it can be seen that the data points fall approximately along this line. If n is the chip count of an FFT processor, and t_c is the time required by the processor to perform a 1,024-

Table 1. Specifications of various FFT processors.

TYPE	NUMBER OF CHIPS	TIME (1,024-POINT)
KOBYLINSKI et al. (1979)	≈15	15.0 s
LIU and PELED, (19759	≈950	41 μs
Mini-MAP; CSP Inc. (1982)	≈800	4.2 ms
CORINTHIOS et al. (1975)	1,800	852 μs
GROGINSKY and WORKS (1970)	≈10,000	9.11 ms
FLADUNG and MERGLER (1978)	120	28.2 ms
Compiled BASIC	5	174 s
Model 10	61,952	21.8 μs
Model 20	68	1.51 s
Model 30	225	0.441 s
Model 40	109	0.535 s
Model 50	150	48.8 ms
Model 60	153	91.7 ms
Model 70	160	41.0 ms

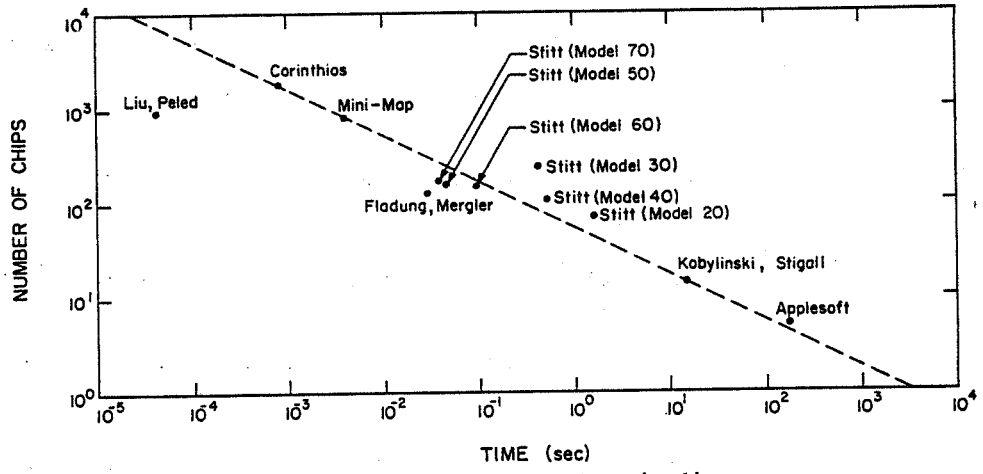

Figure 1. Figure of merit diagram.

point transform, then an equation for the line may be written in terms of these variables as follows:

$$\frac{\log n - \log 1}{\log t_c - \log 3000} = -1/2$$

$$\log n = 1/2 \log \left(\frac{-t_c}{3000}\right)$$

$$\log n^2 = \log \left(\frac{3000}{t_c}\right)$$

$$n^2 t_c = 3000$$

$$100 = \frac{300{,}000}{n^2 t_c}$$

In view of this equation, it is possible to define the following figure of merit (FM) for a FFT processor:

$$FM = \frac{300,000}{n^2 t_c}$$

With this definition, an FFT processor with specifications falling on the line will possess an FM of approximately 100. Specifications lying above this line correspond to FMs less than 100, and specifications below the line to FMs greater than 100. Clearly, a FFT processor with a high figure of merit is preferable to a processor with a low one.

The FMs corresponding to the various FFT processors in Table 1 are shown in Table 2.

The figure of merit represents one way of defining the efficiency of a particular FFT processor. By this definition, it can be seen from Table 2 that the Model 10 through Model 40 processors are fairly inefficient; as has been mentioned before, this is primarily due to the large difference which exists between the computation time of the external processor and the time required by the Apple to transfer data back and forth between itself and the processor. Models 50, 60, and 70 FFT processors introduce various forms of 8X305 based external control in an attempt to alleviate this problem; the relatively high FM values for these processors indicate that the scheme has been largely successful. At least in terms of efficiency, there appears to be little to choose between the Model 50 and 70 processors.

At present, the Model 50 processor uses fixed point arithmetic, requiring the data to be scaled before it may be transformed. Easier data handling and a wider range of data values would be possible if the Model 50 were redesigned using floating point hardware. In a similar vein, the Model 50 currently possesses no method for detecting and handling an overflow error. At the very least, some method should be provided for alerting the Apple to some incorrect transformation results. Finally the spectacular results achieved using the Liu-Peled algorithm (e.g., FLADUNG and MERGLER, 1978) suggest that a system utilizing this algorithm should be investigated.

Table 2. Figure of merit values of various FFT processors.

TYPE	FIGURE OF MERIT (FM)
KOBYLINSKI et al. (1979)	88.9
LIU and PELED, (1975)	8,110
Mini-MAP; CSP Inc. (1982)	112
CORINTHIOS et al. (1975)	109
GROGINSKY and WORKS (1970)	0.329
FLADUNG and MERGLER (1978)	739
Compiled BASIC	69.0
Model 10	3.59
Model 20	43.0
Model 30	13.4
Model 40	47.2
Model 50	273
Model 60	140
Model 70	286

ACKNOWLEDGEMENTS

Research described was supported in part by the National Aeronautics and Space Administration grant NSG 7506 and in part by the National Science Foundation grant ATM 81-20371.

REFERENCES

Corinthios, M. J. and K. C. Smith (1975), A Parallel Radix-4 Fast Fourier Transform Computer, IEEE Trans. Computers, C-24, 80.

CSP, Inc. (1982), Array Processor is small but fast, Electron. Design, 30, (13), 206.

Fladung, R. L. and H. W. Mergler (1978), High-Performance Fast Fourier Transformer, IEEE Trans. Indust. Control Instrum., IECI-25, 322.

Groginsky, H. L. and G. A. Works (1970), A pipeline fast Fourier transform, IEEE Trans. Computers, C-19, 1015.

Kobylinski, R. A., P. D. Stigall and R. E. Ziemer (1979), A microcomputer-based data acquisition system with hardware capabilities to calculate a fast Fourier transform, IEEE Trans. Acoust., Speech Signal Process., ASSP-27, 202.

Liu, B. and A. Peled (1975), A new hardware realization of high-speed fast Fourier transformers, IEEE Trans. Acoust., Speech Signal Process., ASSP-23, 543.