September 1969

SYSTEM ORGANISATION

B. R. Gaines

Lecture notes for IEE Vacation School on
"Engineering Aspects of Microelectronics"

Man-Machine Systems Laboratory,
Department of Electrical Engineering
Science,
University of Essex.

1. Introduction

Device technology has advanced rapidly during the last ten years, and made it possible to include in equipment data-processing sub-systems of increasing power and decreasing cost. However, this advance has not been uniform throughout the whole range of data-processing devices, neither has it proceeded uniformly in time. The trauma of transition from valve to transistor was over more rapidly than expected, but each successive change has had almost as much impact as that transition, and change follows change at a quickening pace. The transition from the early germanium transistors to silicon planar radically affected the speed of digital equipment, the accuracy of analogue equipment, and the temperature range of both. The coming of digital integrated circuits opened up a world of all-digital systems, and made the small digital computer a potential 'component'. However, integrated circuit operational amplifiers, have followed close behind, and their possibilities have been enhanced by thin-film networks and FET switches. Over the horizon looms LSI, and the very low-cost, micro-programmed digital machine, but we can be equally sure that over that same horizon are many developments of which we have taken no account, yet whose impact may be greater than that of LSI.

Despite all these developments and uncertainties, systems have been specified, and systems have been built and sold, and will continue to be so. But the difficulty remains, in a highly competitive market, of not only being that one jump ahead, but also of not being the greater jump behind. "All-digital" has customer appeal, but so has low cost, and when analogue circuitry works (which it often does and has done for a long time) it may be an order of magnitude cheaper. Equally, the small computer, as a black box, is here at ever-decreasing cost, and circuitry is being replaced by software. On the other hand the modules which make up the computer have their own identity as large-scale "components" and may be put together in a variety of non computer-like ways.

Although there is no complete answer to the problems of rapid technological change, there is clearly an initial requirement for a systematic appraisal of available, potential and possible data-processing techniques, in which a clear separation is maintained between device and function. There is little that is new in the data-processing techniques themselves, and an almost exhaustive variety of computing functions have been described in the literature of the past twenty years, but it is only at present that devices are becoming available to make these functions sufficiently low-cost, accurate and reliable, to be commercially viable. In these notes, some of the factors affecting choice of data-processing techniques are outlined and four data-processing techniques which have proven attractive in commercial equipment during the recent past are briefly described.

2. Data-Representation

In evaluating the effectiveness of a sub-system of devices for a particular data-processing function, it is important to distinguish between the effect of the sub-system on the physical signals (generally voltage/current time sequences) and the effect of the sub-system on the quantities represented by these signals. The same sub-system may perform any one of a number of computational functions according to the coding of quantities into its inputs and outputs; for example, a single AND gate may be used as a multiplier, an adder or a gating unit, dependent on the data representation. Equally important, a sub-system may be a cheap and effective computational element in its own right, but the coding units to convert quantities into an appropriate form for its inputs and outputs may themselves be expensive and prone to errors - "all-digital" systems often lose their advantages at the analogue/digital interface.

The ways of representing data are obviously infinite, but some important characteristics may be abstracted: data may be represented by signals which are regarded as continuous in level or discrete in level; by signals which are regarded as continuous in time or discrete in time; by signals which are on a single-line or on multiple lines; and sequential characteristics of the signal may or may not be part of the coding. All the vast variety of data-processing elements may be characterised in these terms and a general, but practical, theory developed for them (de Bakker and Verbeek 1968). However, the importance of the characterisation is that it forms a basis for rationalizing techniques, and exhaustively exploring all solutions to a given data-processing problem. It is generally too restrictive to maintain a single representation in a given sub-system, and the most elegant systems designs (and most cost-effective) are those in which the designer switches from representation to representation, just happening to have the appropriate ones at input and output and the optimum representation for the computation in between.

3. From Component to Sub-System

The systems designer requires components which make no demands on him in their own right. In particular, interconnections should be purely functional and related to the operation of the system, and not the "internal" connections of a system component which has been arbitrarily split into a number of units for the convenience of the device manufacturer; for example, the external "compensation" elements of an operational amplifier.

The importance of the various stages of medium-scale integration and future large-scale integration is not only in economy of manufacture and greater packing density, but also in that these characteristics enable complete sub-systems to be fabricated as single components. The systems designer does not then have to bother with unnecessary characteristics of his components - for example, it is now possible to specify "adders" and "registers" rather than combinations of gates and flip-flops (which were themselves an advance on combinations of transistors, resistors and capacitors).

By dealing in well-defined functional sub-systems with known characteristics and given rules of operation, the systems designer can concentrate on those aspects of the system organisation which are functionally relevant and leave the "nuts and bolts" to the device manufacturer. Within any framework of systems organisation, for example analogue and digital computers, there are sub-systems which form unitary modules, for which there is no point in further sub-division. It is the definition of these "unit sub-systems" which is vital to the device manufacturer in providing mass-produced units satisfying a wide market. A given data-processing technique is ultimately defined in terms of its unit sub-systems, and the effective application of the technique and the availability of suitable unit sub-systems go hand in glove. In the later sections an indication is given of possible unit sub-systems for the techniques described.

4. Hardware and Software

In the computing world a rigid distinction is made between hardware and software, but this distinction is breaking down in modern systems engineering. A computer-aided design program which takes a functional specification of a module and turns it into a specification of components, interconnections and layout, is very similar to a FORTRAN compiler taking a functional specification of a numerical problem and turning it into a sequence of machine orders. This is best seen in languages such as APT for numerical machine tool control where instructions in a FORTRAN-like language are turned into a specification of the movements of tools and workpieces for machining. This similarity is now accepted as fundamental to software theory (di Forino 1969) and the connotations of computing for computer operation, inherent in the term "compiler", have been eradicated by terming all such programs, "translators". Many CAD programs are translators from the designer's terminology to a hardware manufacturing specification.

The view of hardware design as a software operation is interesting because it throws new light on some characteristics of design, such as the arguments on mass-production or customization of i.c.s. One of the vital features of a software system is the "debugging aids" which check-out the actual software. It is easy to design modules which are very difficult to diagnose for faults, and in a customized system where the modules are unique fault-diagnosis can be a massive overhead on system fabrication and maintenance. Diagnosis of faults, and perhaps redundancy to decrease their occurrence are not ruled out by customization, but the complexity of the design requirement suggests that the design overhead will be very high unless the modules are mass-produced.

The similarity between generation of software and customization of hardware becomes very close in the micro-programmed mini-computer (Section 8), where the micro-instructions are set in an LSI array of gates either by masking interconnection patterns or by burning out metallization. The combination of mass-production and customization and of hardware and software in these systems may well be the model for many features of future systems design and its relationship to device manufacture.

The following sections give examples of data-processing techniques which take advantage of modern device availability, and exemplify some of the concepts previously outlined.

5. Analogue Arithmetic with Digital Multiplexing and Sequencing

The classical analogue computer is essentially a continuous differential equation solver using summing integrators as parallel processing elements. Many discrete computations can be expressed simply as the end result of the solution of a differential equation, however, and since the speed of analogue computation is comparatively fast, it is possible to multiplex some of the analogue computing elements and perform the computation sequentially. The configuration obtained is similar to that of a digital computer in which the arithmetic unit has been replaced by a set of switched analogue integrators, whose interconnections are changed by the stored program.

A detailed description of such a data-processor and some of its possibilities has been given by Schmid (1965). The configuration is similar to that of a modern 'hybrid' computer, although the rationale and type of application are very different.

To indicate the possibilities, two integrators may be used to:- add, subtract, multiply, divide, square, square-root, convert polar co-ordinates to rectangular, and vice versa. For example, if two integrators, with outputs y and z, are connected in cascade to realise the equations:

$$\dot{y} = -x$$

$$\dot{z} = -y$$

where x is the constant input to the first integrator, then the solution of the equations as a function of time is:

$$y(t) = -xt + y(o)$$

$$z(t) = \frac{xt^2}{2} - y(o)t + z(o)$$

Hence if the initial conditions are set to

$$y(o) = 0$$

$$z(o) = a$$

$$x = -\tfrac{1}{2}$$

and the integrators are allowed to run until:

$$z(t) = 0$$

when $\quad y(t) = b$

Then we have: $\quad a = t^2/4$

$$b = t/2$$

so that $\quad b = a^{1/2}$

Thus the two-integrator sub-system is a square-root extractor.

Overall long-term accuracies of $\pm 0.1\%$ are now feasible with such multiplexed analogue data-processors, and the technique is particularly attractive when the original data is in analogue form. Because the solution may often be represented as a time-interval, or placed in time-interval form by timing an integrator holding a voltage running to zero, simple digital counters may be used to provide digital outputs in binary or BCD form.

Another important possibility using this technique is the synthesis of "analogue" waveforms using two-state digital signals, since these signals are readily used to modulate true analogue signals through a simple switch. A hybrid representation of one set of data as pulse modulation on two-state lines, whose Fourier transform has low-frequency components of the required form, and another as true analogue data is most attractive.

A case history is given by the requirement for the detection of the deviation from a reference phase angle of a single-cycle of an approximately 10Hz simple-harmonic waveform, heavily contaminated with 3rd and higher harmonics. The new all solid-state equipment had to replace a previous electromechanical system whose accuracy was not better than $\pm 1^{\circ}$. The output was required in BCD for data transmission; an offset was to be applied to the output so that it was a deviation from normal.

The obvious technique to use is phase-sensitive detection of the in-phase and quadrature components of the input by multiplying it with appropriate reference waveforms and integrating over a full cycle. Because of the odd harmonics in the input it appeared that for chopping multipliers with square-wave references to be used, the 3rd and higher harmonics would have to be filtered from the input using very stable filters in the 20Hz region, introducing known phase change at the fundamental. Once the in-phase and quadrature components had been obtained a $\tan^{-1}$ transformation would have to be applied at the output converted to BCD form; complex equipment seemed necessary.

However, by using a pulse-pattern, rather than a square-wave reference, it was possible to eliminate the effects of 3rd and 5th harmonics in the input and require only a simple analogue filter in the 60Hz region. By using the

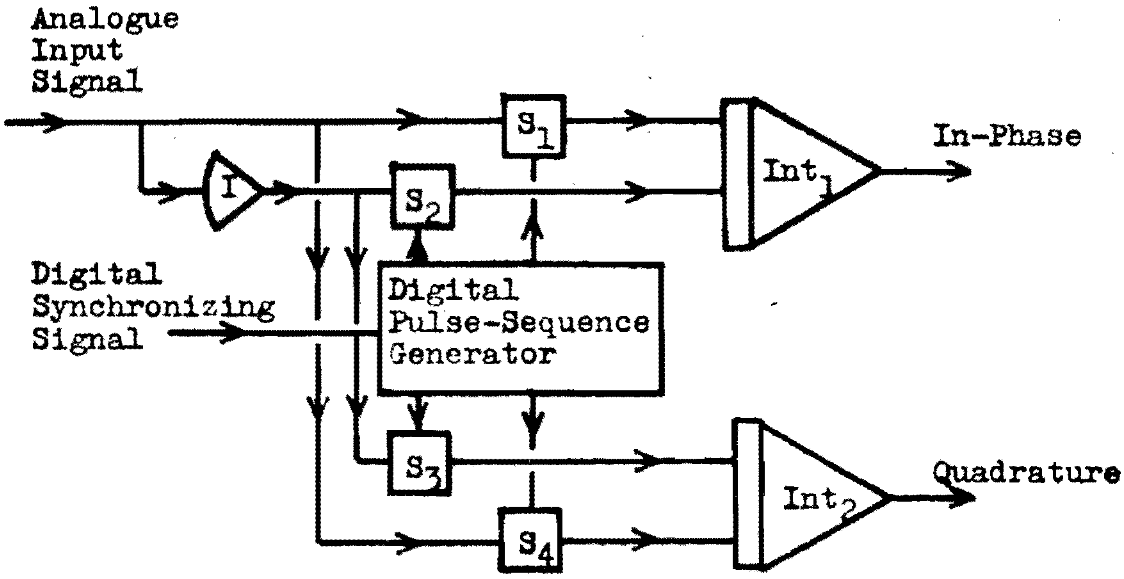

Figure 1 Phase-Sensitive Detector with Harmonic Rejection

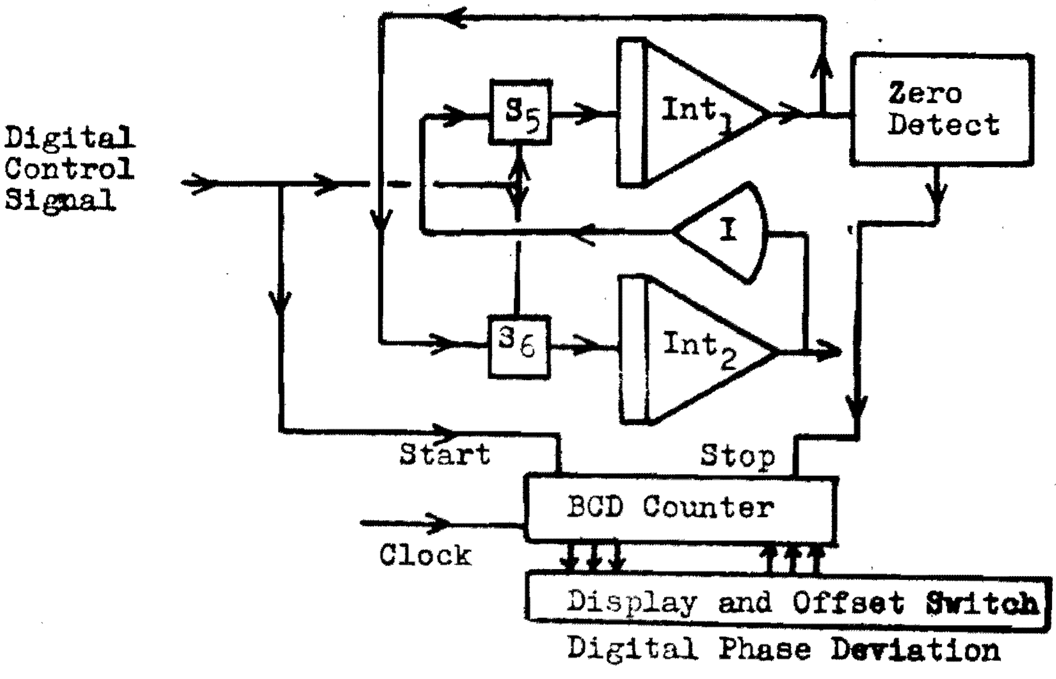

Figure 2 Rectangular/Polar Convertor with Digital Output

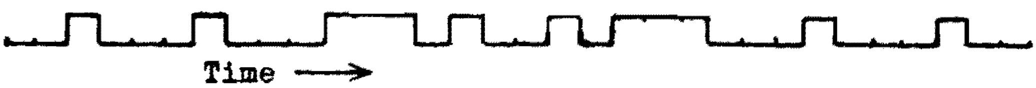

Figure 3 Half Cycle of Waveform with Low 3rd/5th Harmonics

same two integrators that formed the in-phase and quadrature components, in the mode previously outlined, to rotate then so that the quadrature component was zero, the $\tan^{-1}$ calculation was eliminated. Since the time interval for this to occur was proportional to the phase-angle, a simple BCD counter counting at the same time provided a BCD output; the deviation was obtained by setting in the expected angle from switches as the initial value in the computer.

Figure 1 shows the set-up for signal extraction using FET switches, S_1 through S_4,as chopping multipliers, multiplying the reference signal (represented by a pulse pattern from the Digital Pulse Sequence Generator) into the normal and inverted input. At the end of one cycle of the reference, the summing integrator, Int_1, contains a quantity proportional to the in-phase component of the input signal, and Int_2 contains the quadrature component.

When the reference cycle is complete, the BCD counter is triggered to start counting and the integrators are connected by FET switches in the mode shown in Figure 2, so that they undergo simple harmonic oscillation, corresponding to rotation of the vector of components. When the in-phase component reaches zero the BCD counter is stopped and then contains the required phase angle. The integrators are then reset ready to perform the signal extraction again. The timing is such that every 3rd cycle of input from a continuous input can be analysed in this way.

A half-cycle of the switching waveform applied to S_1 through S_4 is shown in Figure 3. This waveform is applied to S_1 for one half-cycle and to S_2 for the other. Similar components in quadrature were applied to S_3 and S_4. This waveform may be shown to have 0.024% 3rd harmonic and .28% fifth harmonic. It has 44% seventh harmonic, but simple analogue filters ensure that there is negligible 7th harmonic in the input signal.

This phase-angle measuring sub-system has enabled great economies to be made relative the previous electromechanical system. Long-term accuracies of better than $\pm 1^{\circ}$ are being obtained using single ic operational amplifiers and FET switches. The fascinating features of the units is the use of various data representations in such a way that the data is always in the right form at the right time without artificial transformations.

The Unit Analogue Sub-System

The disadvantage of most present operational amplifiers are that they require many external components, far bulkier than the ic package containing the amplifier. An ideal unit sub-system would be a fully compensated operational amplifier with built-in current and voltage offset correction, temperature compensation and frequency compensation, combined with a set of similarly compensated FET switches at its input, driven by TTL level digital signals.

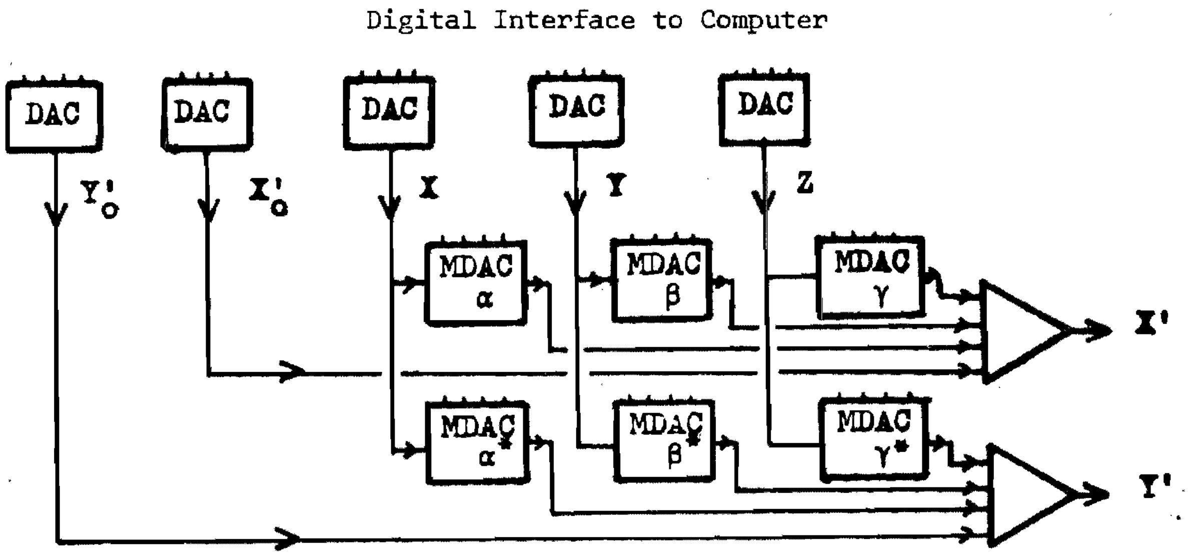

Figure 4 Rectangular Co-ordinate Transformation

6. Digital/Analogue Convertors in Arithmetic Units

Analogue data-processing may also be brought into the CPU by replacing digital arithmetic, not with analogue integrators, but with multiplying digital-analogue convertors (D/A convertors with analogue signals on their 'reference' lines). This is particularly attractive for display generation, and analogue signal processing, and its application in the Adage "Ambilog" computers and displays has been described by Hagan (1966). Recent falls in cost of high-precision high-speed D/A convertors (from £500 to £50) has made their use in simpler equipment economically viable.

Figure 4 shows five digital/analogue convertors (DACs) used in conjunction with six multiplying digital/analogue convertors (MDACs) and two summing amplifiers to perform the co-ordinate transformation:

$$X' = X'_0 + dX + \ Y + \ Z$$

$$Y' = Y'_0 + d{*}X + \ {*}Y + \ {*}Z$$

This system has been used in conjunction with an analogue vector generator in a commerically-available computer-driven graphic display which can generate 1000 long vectors or 5000 short vecotrs at a 30Hz repetition rate and provides a true perspective visual display with facilities for arbitrary translation and rotation.

MDAC Unit Sub-System

A double-buffered multiplying digital-analogue convertor with hiway drive facilities enabling any unit to be addressed and up-dated and then the up-dated information to be transferred to the outputs of all units simultaneously forms a unit sub-system.

7. Incremental Computing Techniques

A further variation on the system described in Section 5 is to make it 'all-digital' by replacing the analogue integrators with digital counters (that is, DDA integrators). This is attractive when higher accuracy is required, or the original data is in the digital form. The use of counting techniques in data-processing has been described by Gaines (1968), and an outstanding example of their application, in the Solartron Digital Transfer-Function Analyser, has been given by Elsden and Ley (1969).

The fascinating characteristic of DDA techniques is that they date back to the same time as the general-purpose digital computer (Gaines and Shemer 1969), and are continually being re-invented in some form or the other. Like the analogue computer, the DDA is most effective in providing continuous dynamic solutions of differential equations at speeds far higher than is possible with GP digital machines. In the single-solution multiplexed mode, DDA speeds drop below that of both analogue and GP digital machines. This is clear if the DDA is regarded as using counting techniques rather than parallel digital arithmetic. Nevertheless counters are often under-used as data-processing devices, and the DDA literature is a useful source of novel ideas for the use of counters.

Incremental Unit Sub-Systems

Five basic units are described in Gaines (1968a, 1968b): these are the counters themselves, asynchronous units with synchronous control lines; the "modulators", adder/register combinations to convert parallel valves in the counters to sequential DDA pulse-streams; "input-control" units, digital routing logic to vary the interconnections of the counters and modulators; the "sequencing units", digital equivalents of rotary switches which drive the routing logic to set up and change different configurations; the "differentiators", condition-detectors which drive the sequencing units to different states as the computation proceeds. An alternative set of basic elements capable of performing parallel-digital as well as counting computations has been described by Shemer (1968).

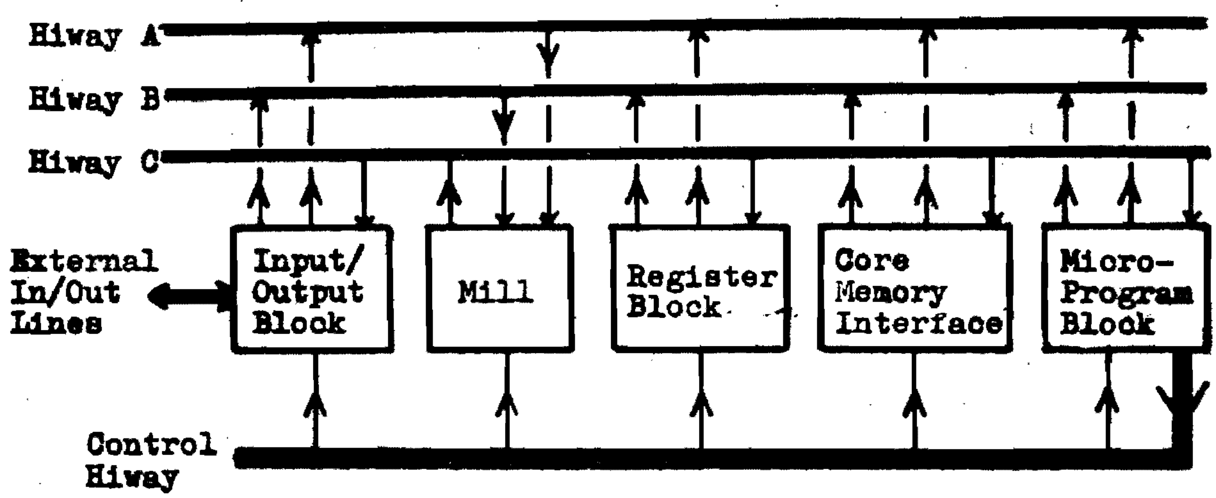

Figure 5 Microprogrammed Minicomputer

8. Micro-programmed small machines

The techniques described so far have involved modifying the arithmetic units of a GP-machine-like structure to generate special operations required in equipment. There is the alternative possibility of maintaining parallel-digital processing, but generating special operations through extensive micro-programming, that is, having a single machine order give rise to a long sequence of stored operations which themselves may be varied or "programmed". This is attractive now that single package arithmetic units are promised by LSI, and micro-circuit read-only stores with 1,000's bits/package are already with us. Micro-programming has been exploited in emulators on large machines, but its full potential in small machines is only just being utilized; customized micro-programming has become available in a number of recent commercial mini-computers.

The importance of the micro-programmed mini-computer is not only in that it enables complex operations to be made available low cost, but that it breaks the black-box unitary "computer" into a set of modules which have identities in their own right, and which may be combined into other sub-systems with less resemblance to conventional "computers". These modules may be manufactured as basic components and combined in a variety of ways to form not just one of a range of computers, but also virtually any other piece of digital equipment. Micro-programming enables us to break away from the use of computers to simulate other systems, to the fabrication of "computers" which actually become other systems.

Figure 5 is the block diagram of a typical micro-programmed mini-computer. There are three data hiways, A, B and C, and one control hiway. Each block, except the "mill" is able to "write" on hiways A or B, and read from hiway C. The mill reads from hiways A and B, performs an operation on the data, and drives its output into hiway C. All "writing" on to the hiways A and B is through bus-gates, (open collector transistors), so that the modules are independent of one another and may be plugged in anywhere along the hiway. The operation of the various blocks and the control of what devices are writing on the hiway is performed by the "control hiway" which is itself driven from a read-only store in the micro-program block. The logic throughout the computer operates synchronously with a clock.

Digital Computer Unit Sub-Systems

Each of the blocks shown in Figure 5, for an 8-bit hiway machine, takes some 40-70 74N TTL packages, and is well suited to single-chip construction.

The "register block" is a set of flip-flop registers, one of whose inputs may be set to read from hiway C and two of whose outputs may be set to write on hiways A and B respectively, under the control of register address lines in the control hiway.

The core memory interface controls the read-write operations of a core store, and enables addresses and data to be transferred to the core memory to and from hiways A, B and C.

The "mill" is a combinational logic unit enabling arithmetic, logic and shift operations to be performed on hiways A and B together and the result placed on hiway C.

The input-output block provides for data-transfer to and from peripheral devices attached to the computer.

The micro-program block contains the micro-program counter pointing to a word in a read-only store which is output on to the control hiway. The counter itself may be made to operate under micro-program control, enabling normal programming techniques to be used in the micro-program.

9. Conclusions

The interesting characteristic of all the data-processing techniques outlined in Sections 5 through 8 is the great variety of computations possible with so few unit sub-systems modules. It is also interesting to note how readily these sub-systems separate out, and how they act as a virtual "definition" of the data-processing technique.

All of the techniques have been "known" for at least twenty years, but all were severely restricted in their application until the advent of integrated circuits. More important than any one "technique" however is the way in which well-defined modules may be used together to form effective overall systems.

10. References

de Bakker, W., Verbeek, L. (1966) Study of Analog, Digital and Hybrid Computers Using Automata Theory, ICC Bulletin 5 215-244.

di Forino, A.C. (1969) Programming Languages, in Advances in Information Systems Science Vol. 1 Ed. Tou, Plenum Press.

Elsden, C.S., Ley, A.J. A Digital Transfer Function Analyser Based on Pulse Rate Techniques, Automatica 5 (1).

Gaines, B.R. (1968a) Varieties of Computer - Their Applications and Interrelationships, Proc. IFAC Symposium on Pulse Techniques, Prague.

Gaines, B.R. (1968b) A Modular Programmed DDA for Real-Time Computation, Proc. IFIP Congress, Edinburgh.

Gaines, B.R., Shemer, R.A. (1969) From Control Mathematics to Control Hardware, Automatica 5 (1).

Hagan, T.G., Treiber, R. (1966) Hybrid Analog/Digital Techniques for Signal Processing Applications, AFIPS 28 SJCC.

Schmid, H. (1965) Sequential Analog-Digital Computer (SADC) AFIPS 27 (1) FJCC.

Shemer, R.A. (1968) A Hybrid-Mode Modular Computing System, Proc. IFAC Symposium on Pulse Techniques, Prague.