

The Evolution of the Fred Machine

Gordon Brebner
Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ
Britain

1. Introduction

This paper reviews the history of the Fred Machine¹ (henceforth abbreviated to FMac), a flexible computer designed in the Department of Computer Science at the University of Edinburgh. A main intention is to illustrate the progress of an advanced computer system design under the influence of the constraints and demands of an academic computing environment. It must be stated from the outset that the author has not been a major contributor to the work involved, but writes from the viewpoint of a Department member who has had a fairly close acquaintance with the relevant people and computer facilities. Accordingly, the paper is not rich in technical detail, which may be found elsewhere.

The FMac project has been active since 1981, and is now in a position in which further major development is unlikely to take place. In common with many pieces of academic work involving computer systems, it has occupied an uneasy niche between the areas of research and development. At the outset, the aim was to produce a flexible architecture to facilitate experimentation with sophisticated computer components. However, the subsequent evolution of the FMac has been such that much work has been done with the dual aims of providing a better computing service to the Department, and of enhancing practical educational facilities. Because of this, no research funding has ever been sought for the project (although other research has made use of its products).

The FMac project involved an extension of the traditional areas of expertise in the Department, namely software and theory, into hardware and firmware. It is reassuring that the products of the project have, in general, achieved goals such as modularity and flexibility which are associated with the former "higher level" activities. In this respect, the project has achieved an objective worthy of a research project, namely the design of a computer system using structured principles from the lowest level upwards. The disappointing shortcoming of the project has been a marked lack of provision of sophisticated system software to complement the hardware facilities. Innovation in this area has largely been delegated to the authors of application programs (an observation which might, of course, be construed as evidence that the system is, indeed, modular and flexible).

¹ The title "Fred Machine" is not a result of the tendency of computer scientists to choose unimaginative names, nor is it an acronym; rather, it is chosen in honour of the principal designer, Fred King.

The remainder of the paper is organised so that the reader can follow the evolution of the FMac over the last seven years. Section 2 briefly outlines the state of relevant work which had been pursued in the Department in the period immediately prior to the FMac project. Section 3 describes the central feature of the FMac, the system bus, which was designed to allow the interconnection of a wide variety of hardware components. Section 4 discusses the range of hardware and software which has been provided as part of the project work; in the main, this has been directed towards the development of a powerful single-user workstation. Section 5 discusses how the resulting system has been used for research and teaching, and mentions some of the additional hardware and software which has emerged from such activities. Finally, Section 6 draws some conclusions from the way in which the project has progressed, and comments on the future prospects of the FMac.

2. Background

In the late 1970s, a significant amount of work was carried out in the Department of Computer Science at Edinburgh on the development of various computer systems. A number of projects had come to fruition around 1980, and so it was not surprising that a collection of people focussed upon the FMac project in order to apply their skills further. Thus, past work had a significant impact on the course of the project. Four main strands are worthy of note, and they are described in the next four subsections.

2.1 The "Micro Kits"

The micro kits had been produced mainly as hardware teaching aids, and provided a breadboard upon which an assembly of chips could be placed, interconnected, and used experimentally. In particular, various microprocessor chips (Motorola M6502 and M6809, and Zilog Z80) could be incorporated; programs for these were stored in EPROMs, which could be written and erased as required. The micro kit can be seen as the hardware progenitor of the FMac; while it allowed experimentation with the contents of one circuit board, the FMac was designed to allow easy experimentation with a collection of circuit boards, via a flexible system bus under the overall control of a "boot processor" board. Therefore, the main intended use of the FMac was to assist design work at a higher level of hardware functionality. (It is interesting to note that the micro kits are now mainly used for testing VLSI chips designed in the Department - technology now allows rich functionality at the chip level.)

2.2 ELAN

The development of the Ethernet Local Area Network (LAN) at Xerox PARC in the mid-1970s had aroused immediate interest in the Department. This led to two alternative experimental designs for a contention bus network to be used within the Department. By 1980, one design had been preferred, and preliminary work was being done on cabling the whole Department, and producing interfaces to allow various computers to use it. In general, ELAN (the Edinburgh LAN) had similar properties to Ethernet, the main operational difference being that it ran at 2 MBits/sec, rather than 10 MBits/sec. Given this communications infrastructure, it was an important goal to interface the FMac to it, with a view to subsequent resource sharing.

2.3 ISYS, LEGOS and the Filestore

Within the Department, minicomputer provision was mainly concentrated upon Interdata series 70 minicomputers. The ISYS operating system, a local product, enabled the use of clusters of disc-less Interdata 74s as limited workstations, backed up by more powerful Interdata 70s with disc storage. With the advent of further disc-less minicomputers of various kinds, the Filestore, a file server based on an Interdata 70, had been produced. It communicated (via point-to-point links) with minicomputers using a protocol which allowed the reading and writing of files, either as a whole or blockwise, together with elementary directory operations. To use it fully, the LEGOS operating system had been written; based on ISYS, it enabled a disc-less Interdata 70 to be used as a workstation, with the Filestore as a storage facility. Importantly (as the name may suggest to a reader familiar with children's toys), it was designed to allow users to tailor the system to their own needs by using appropriate software building blocks. This philosophy was important in determining both hardware and software goals of the FMac project.

2.4 EMAS and IMP

EMAS (the Edinburgh Multi-Access System) was a time-sharing operating system developed for large mainframe computers (initially the ICL 4-75 - an IBM 360 simulacrum - and subsequently the ICL 2900 series) at the University of Edinburgh during the 1970s. While many features of EMAS were not applicable to smaller-scale systems based on the FMac, its guiding principle of well-structured system software organisation was applicable, in particular the policy of writing all software in a high-level language. The language used to implement EMAS was IMP (IMPlimentation language), an Edinburgh-designed derivative of Atlas Autocode. Since languages such as C and Pascal had not yet become fashionable at that time, IMP had become the usual language for all programming in the Department. Although doubts about the wisdom of concentrating on a parochial programming language were beginning to surface at the time the FMac project started, it was not surprising that the main initial software goal was the establishment of a reliable IMP compiler and run-time system. This was a prerequisite for other software development.

These four projects effectively defined the framework in which the FMac project developed. In subsequent sections, the course of this development will be traced.

3. The Fred Bus

The central feature of the FMac project is the common system bus, the Fred Bus. Indeed, it is the only item that is a mandatory part of any computer system based on the FMac. It was designed to allow experimentation with sophisticated components built on circuit boards, as mentioned in Section 2.1. To achieve this aim, it was intended to have sufficient capacity and flexibility to handle not only current components, but also those which were likely to be available in the future. Further, the specification was deliberately kept simple, in order to allow easy and fast introduction of new components.

The main characteristics of the Fred Bus are a 32-bit data bus and a 32-bit address bus. At the time the FMac project started, the normal microcomputer data bus width was 8-bit or 16-bit, and so the Fred Bus was anticipating future microprocessor developments; standards for 32-bit buses, such as the IEEE Futurebus project, were in their infancy. The first microprocessor envisaged as a component on the Fred Bus was the Motorola MC68000, which needed a 16-bit data bus and a 24-bit address bus, and so could be comfortably accommodated. The address bus supports the addressing of bytes, half-words, or words, within 32-bit words.

Components attached to the Fred Bus can behave as either "masters" or "slaves". A master (e.g. a processor) may initiate a data transfer by specifying the transfer direction and an address within the bus address space, and then asserting a transfer request signal. A slave (e.g. a memory module) responds to this signal with a transfer acknowledge signal. This asynchronous handshake protocol accommodates different response speeds from slaves. The upper limit of the bus itself is 3 million transfers per second. In order to ensure safe allocation of bus usage to masters, additional bus request and bus acknowledge signals are used to support acquisition of the bus, and arbitration between potential masters. In addition, there is provision for atomic multi-transfer operations (e.g. test and set on a memory location).

There is no specific mechanism to support interrupts. The intention is that processor components be designed so that they can also function as slaves on the bus, and have addresses associated with special registers which, when accessed, cause interrupt requests to be generated. This type of mechanism for controlling processors can be extended to include such things as halting or resetting, if a component designer wishes. The Fred Bus also supports further facilities, including a 100 Hz clock signal, slave error indications, bus transfer timeout indications, power failure indications and bus-wide resetting. An FMac normally contains a Fred Bus, a power supply and bus slots for up to nineteen double-height Eurocards, contained within a tinted-perspex fronted cabinet which lends it the appearance of a microwave oven.

The reader who is familiar with the operation of the MC68000 microprocessor may have noted that the general operation of the Fred Bus is well suited to the production of a processor component containing this chip. Of course, this is no surprise for, as was alluded to above, the MC68000 was seen as an exciting new microprocessor at the outset of the FMac project, and so had an inevitable impact on the bus design. In order to have confidence in the Fred Bus, it was important to demonstrate that it could cope with different sophisticated components. Section 6 contains evidence that it has had enough flexibility to allow widespread experimentation; what has been true (and is inevitable given the general nature of the bus design) is that sometimes components cannot be operated at peak efficiency. For example, an early component consisting of a Zilog Z8001 processor with a Z8010 memory management unit was slowed by the need to have bus addresses completely specified when a bus transfer request was issued. However, as the aim of the exercise was to get speedy experience of a new microprocessor, the slow-down was not crucial. This phenomenon is reminiscent of portable compilers, which allow software to run on several computers, but sacrifice the optimisations made by a hand-crafted compiler. In summary, it can be said that the Fred Bus has been shown to be adequate over the years.

4. Harnessing the Fred Machine

During the academic year 1981-82, three prototype FMacS were built. As a first step, a boot processor component was designed by Fred King, in order to allow control to be exercised over the system. Then, as a useful experimental exercise, this component was combined with further hardware and software to implement a workstation attached to ELAN, the local area network mentioned in Section 2.2, by a group consisting of Hamish Dewar, Fred King and Rainer Thonnes.

The boot processor component was based on a 8 MHz MC68000 microprocessor, with the address space split into local and system-wide areas. The local address space contained 16K of either RAM or ROM, together with an RS232 port and a programmable timer. The system-wide address space was provided by using simple mapping registers to convert 24-bit microprocessor addresses into 32-bit Fred Bus addresses. The local on-board bus was made available on a separate edge connector to allow off-board expansion of the local resources, independently of the Fred Bus; this was to simplify the incorporation of less sophisticated system components.

While the boot processor board was capable of stand-alone operation in conjunction with a terminal, the next two obvious hardware additions were a memory component to increase the storage capacity, and a networking component to enable the bulk transfer of data to and from the outside world. The memory component interfaced 0.5 MByte of RAM, with parity checking and an access time of 225 ns, to the Fred Bus; its address range within the system-wide address space was configurable by on-board switches. The network component interfaced ELAN to the boot processor local bus, and was of a rather more complex nature, in that it incorporated a Zilog Z80 processor, whose operation was controlled via special registers made available on the local bus. The Z80 was responsible for extracting data packets transmitted along the ELAN channel to the FMac, and holding them on an on-board queue until the boot processor copied them into main memory. Similarly, when the boot processor copied packets from main memory to the network component, they were held in an on-board queue until it was able to transmit them along the ELAN channel. Nowadays, given a standard type of LAN, such functions could be obtained from a single-chip controller but, at that time, LANs were still very much in their infancy.

In parallel with this hardware development, work was done to produce basic system software. As noted in Section 2.4, the priorities were an IMP compiler and an IMP run-time system. The compiler was written in IMP from scratch, and it was compiled on LEGOS workstations (mentioned in Section 2.3) until it was reliable enough to compile itself into MC68000 machine code. Given that IMP can express reads and writes to arbitrary addresses, no special language extensions were necessary to enable full exercising of the FMac. The IMP run-time system was written in MC68000 assembly language and interacted directly with the bare MC68000 processor, acting as a primitive operating system. It contained features to support IMP input/output and event signalling facilities, to handle a video terminal and the clock, to access ELAN, and to provide a simple software "front panel".

Once this work had been completed, a usable workstation had been created and, unanticipated by anyone then, it was going to exist for a long time.

Given the availability of the Filestore (mentioned in Section 2.3), which acted as both a file and print server on ELAN by then, and support for the IMP language, it was not long before the FMac workstation had an equivalent functional capability to the existing LEGOS workstations. The additional attractive feature was the availability of a faster processor, and eight times as much memory. This fact encouraged various people to move applications to the FMac workstation; one particular example was ESDL, a circuit board design suite, and so an FMac could soon be used to design extensions to itself. Unfortunately, this quick harnessing of the capabilities of the FMac was the harbinger of a certain amount of system stagnation.

In most academic environments, there is a conflict of interests between the research and development of computer systems, and the demands of the user community for a sophisticated computing service. Satisfying the latter need can, of course, supply essential feedback to the former activity, but inevitably requires that systems have firm specifications and stable behaviour. In the case of the FMac, the experimental workstation effectively became public property as soon as it was functional, and before there had been any detailed thought about what might be the best way to offer a public service. In the summer of 1982, the following sentence appeared in publicity accompanying a demonstration of the workstation at a University Open Day:

"The Edinburgh Advanced Personal Machine (APM) has been developed in the Department of Computer Science to fulfil its requirements for general purpose computing over the next five or so years."

This marked the transition of the FMac from experimental research and development to official computing resource. The name APM (chosen in haste for publicity purposes) has become an accepted mode of address for the FMac although, strictly speaking, it only refers to a particular application of the FMac as a workstation. It is encouraging to note that the APM has, indeed, served its purpose for the last five years, and continues to do so. Unfortunately, the rise of the APM caused the loss of the pioneering momentum which should have led to the FMac continuing to justify the "advanced" tag as the years progressed. This was exacerbated by the departure of personnel from the project, and lively discussions of the advantages and disadvantages of having more managerial control over what was a group of self-motivated and competent individuals.

It is important to stress that complete inertia did not set in with the emergence of the APM. A substantial amount of further work, particularly on the hardware side, has been done since. In the rest of this section, a brief account of the subsequent development of the FMac project is given. The ordering is meant to be logical, not chronological.

As far as the hardware described above is concerned, developments have reflected advances in technology. On the boot processor component, the MC68000 processor chip has been replaced by a small "daughter board" carrying an MC68010 processor chip and associated memory management chips; this gives a virtual memory capability to the component. The next likely Motorola candidate for a processor component is the MC68030, which would fully exercise the 32-bit data and address buses, and has an on-chip memory management unit. On the memory component, 64K chips have been replaced by 256K chips, leading to a four-fold increase in on-board storage capacity to 2MBytes. The network component remains the same, although extensions have been

made to the controlling software in the Z80. However, the advent of an IEEE 802.3 standard 10 MBits/sec contention bus LAN has led to an experimental interface being built for the FMac. As well as updating the original three components, new components have been designed to interface various other devices to the Fred Bus, so that FMacS can have more powerful capabilities when used as workstations, and also so that FMacS can be used as the basis of specialised servers attached to ELAN.

The most important new facility for workstations has been a colour graphics component. An initial version consisted of a single-board frame store which generated a display of 1024 x 1024 4-bit pixels at a rate of 25 MHz, and was accessible as a write-only slave on the Fred Bus. The current version has a second board added, which extends the frame store to have 8-bit pixels, and offers a range of 32768 colours. This component has been substantially responsible for ensuring the popularity of FMac-based workstations with the user community.

In the context of server implementation, two main components have been produced. The first is a Winchester disc controller, which has been used in producing new file servers, as well as in supporting stand-alone FMac systems. The original Interdata 70 Filestore no longer exists, and has been replaced by several FMac-based file servers equipped with either 160 Mbyte or 300 MByte Fujitsu disc drives. These servers support a large number of disc-less FMac-based workstations via ELAN. The second component is a laser printer driver which can transfer arrays of pixels from FMac system memory to a laser printer. Several FMac-based printer servers using Canon laser printers are now available on ELAN; these print plain text files, output files from the Scribe and TeX text formatters, and input files for the local Layout text formatter.

Three multi-function components have been produced. The DMAK (Disc, Mouse, Arbiter and Keyboard) component contains a collection of useful facilities. It has a floppy disc interface, which has made the production of a portable stand-alone FMac system possible; a mouse interface, which allows a three-button mouse to be used in association with the colour graphics component; a bus master arbiter module, for use in multiple master systems; and a serial interface which, amongst other things, has been employed to connect terminal-type keyboards to FMacS. The QSART (Quadruple Synchronous/Asynchronous Receiver/Transmitter) component interfaces up to four synchronous or asynchronous links to the Fred Bus. Connected to RS232 links, it has been used in both forward mode, in which four terminals are multiplexed on to an FMac, and reverse mode, in which an FMac can access four terminal ports on other computer systems. Finally the RT (Real Time) component contains a serial interface, two 8-bit parallel interfaces, a programmable timer and a Motorola M6809 microprocessor, which can be downloaded and controlled via the Fred Bus. It has been used to interface various real-time equipment to FMacS.

Obviously, the hardware is of only limited use without software to facilitate its operation. A prevailing philosophy of the FMac project was that system software should be a thin filling sandwiched between the hardware and user software. While this approach fitted with the original spirit of the FMac project, it was less desirable for the APM, where fairly rich support for user software was required. Therefore, it can be said that FMac system software has not advanced as much as the hardware.

The core software for the FMac remains the original IMP run-time/operating system. It supports programs written in IMP, and in Pascal (compiled by a compiler derived from the IMP compiler). It also supplies libraries for floating point arithmetic, for handling virtual video terminals, for drawing colour graphics primitives, for using ELAN, and for accessing files on the filestore. User object code libraries may also be created and, irrespective of original language, can be dynamically linked with main programs at run-time. As well as the compilers, many workstation facilities are built upon this basic core, including several screen editors and text formatters, chip and circuit board design suites, file maintenance and printing commands and the ability to communicate with other computer systems on ELAN. The standard user interface is via an interpreter which reads commands from a terminal keyboard, and reports to the terminal screen.

In most respects, this collection of system software supplies what academic computer scientists need and, to an extent, it is consistent with the "building block" nature of the hardware in that it provides a solid, if monolithic, basis for users to add new software modules. Two major criticisms have been aimed at the system, first that it is non-standard and so cannot run software written elsewhere, and second (particularly since the rise of Unix²) that it does not support multiple processes. A simultaneous solution to both of these problems, namely implementing a standard multi-process operating system on the FMac, is mentioned in Section 5. The absence of processes has caused the evolution of an event-driven style of program organisation, akin to that recommended for the single-process Apple Macintosh. This is adequate in many circumstances, but user programs cannot be expected to handle all system events and, as a result, one conspicuous failing of the FMac systems has been the absence of background communications activities, such as the receipt of electronic mail. In the interests of simplicity, interrupt handlers in the run-time system have not been expanded into complex event handlers in order to deal with this problem.

In fact, a lightweight multi-process kernel was produced for the FMac as long ago as 1983 but, for no good reason, it remained in cold storage until 1986. This kernel was a software analogue of the Fred Bus, being restricted to first-level hardware event handling and process coordination over multiple processors, with a flexible range of process modules used to deal with such things as store management, naming and device/user interfacing. A variant of the kernel is now in use in the FMac-based file servers and, after some modification to deal with such things as virtual memory, a further variant is scheduled for release into user service shortly. The mistake of bringing new software to the attention of users immediately has not been repeated this time, and it is to be hoped that the design will have had time to mature. Unfortunately, however, the delay in introducing it may mean that users will be reluctant to harness it fully, given that the lifetime of the FMac is limited.

This section has described only those components of FMac systems which are seen as "officially supplied". In line with the open-ended nature of the system, much hardware and software has also been developed as a result of user applications of the FMac, and some examples are outlined in the next section.

² Unix is the trademark of a bunch of capitalist bastards.

5. Using the Fred Machine

The FMac, particularly in its APM guise, has found widespread application in the research and teaching activities of the Department of Computer Science at Edinburgh. Projects performed by final year undergraduates and M.Sc. postgraduates have been a particularly fruitful source of innovation (even when they have not yielded completely functional products). A large amount of usage of the FMac has been for the development of application software; inasmuch as this could have been performed on most workstations or mainframes, it is not detailed here. Therefore, there is further bias towards hardware work in this section, although this has been a minority activity as far as users are concerned.

As a result of user efforts, much useful software exists. Several window and menu based user interfaces, using colour graphics and a mouse, have been produced, as well as a more advanced command line interpreter. A compiler for C is available, and it has been used to port various applications, such as the TeX text formatter and an interpreter for ML (a functional programming language), to the FMac. The multi-process MUSS operating system, developed at Manchester University, has been ported to an FMac system with local disc, and this system can run Unix on top of itself. Independently, a direct Unix implementation was started but, with the current proliferation of other Unix workstations in the Department, was recently abandoned.

Research has been done on a microprogrammable processor component for the FMac. It is centred upon a 16-bit wide horizontally microprogrammed data path using AMD AM29203 bit-slices. Its major application has been in the production of an intelligent graphics component, which can compute images from structured descriptions residing in main memory, and display them. Research work in computer graphics, image processing, computer-aided design and VLSI design has made extensive use of the standard colour graphics capability of the FMac. Research work on distributed computing systems, in particular on network file and name servers, has had the collection of FMacS, connected via ELAN, available as an experimental testbed; a spin-off from this work has been the latest type of FMac-based file server.

The FMac has been used in conjunction with teaching at all levels. Courses using it in an interesting way have included Microprogramming, where the microprogrammable component mentioned above has been used to emulate various instruction sets; Operating Systems, where simple multi-process operating systems have been written for the FMac; Digital Communications, where several FMacS have been linked together using standard protocols; Advanced Graphics, where various algorithms have used the colour graphics component to produce sophisticated three-dimensional images; and Real Time Systems, where an assortment of real-time equipment, including robot arms and turntables, has been interfaced to the FMac.

Student projects have resulted in an array of components for the FMac, which can merely be listed here. Processor components have been based upon the Zilog Z8000, Intel 8086 and iAPX286, and National Semiconductor 16032 and 32032. Unfortunately, lack of software, such as compilers, has meant that these components have only been exercised at a low level. Other components have included a speech coder-decoder interface, a piano keyboard interface, a digitising camera interface, a

Rugby time signal receiver, a satellite picture receiver (in conjunction with the Meteorology Department), an X.25 level 2 communications interface, a floppy disc controller and hardware floating point support; further additions to this collection are expected in the future. Student projects have also contributed a Basic interpreter, an Occam compiler and run-time environment, and the Visicalc spreadsheet package.

6. Conclusions and the future

The previous section indicates that the FMac has been a success with computer users in the Department of Computer Science at Edinburgh. Doubts about the success of the FMac project might arise from two directions. From one point of view, it might be felt that the original potential of the FMac to encourage work on structured hardware and software design has largely been wasted, due to the premature concentration on the APM; moreover, the commitment to supporting a computing service implied constraints, such as rigidity of specification and an increased degree of managerial interference. From another point of view, it might be regretted that a large amount of effort was spent on providing a parochial computing system, and that further effort is necessary to make imported software available on it. These two points are both valid, and have been reflected in the fact that most FMac work is now undertaken by computing support staff, and that various proprietary workstations are available in addition to the APM. What cannot be denied is that the FMac was ahead of its time, and users in the Department have had a disc-less workstation with a colour graphical display, backed up by a powerful processor, a large memory and a link to various networked servers, available for a number of years.

Two of the three principal members of the FMac project group have now left the Department, one to found a company selling laser printer controllers partially based on FMac technology. Therefore, significant further development of the FMac is unlikely. At the time of writing, around sixty FMacs are in service, about thirty of these being workstations with colour graphics. During term-time, there is often a shortage of public systems for students to use. However, it is unlikely that any more FMacs will be built, since the Department is moving towards proprietary facilities. A large number of Sun workstations, connected by a standard 10 MBit/sec contention bus, are now available, as well as a significant number of Apple Macintoshes, connected by Appletalk. The FMacs will remain in useful service within the Department for several more years, presumably until they are incapacitated by old age.

Acknowledgements

Without the work of Fred King, there would have been no Fred Machine. Further, without the benefit of countless hours spent in conversation with him over the years, the author could not have written this paper. The other major contributors to the project have been Hamish Dewar and Rainer Thonnes. Many other people have helped the project, in particular Igor Hansen, Jimmy Johnstone and George Ross. Any errors or omissions, whether historical or technical, in this paper are the fault of the author.