Check for updates

EVALUATION AND EXPERIENCE WITH THE IBM SERIES/1 MINICOMPUTER

by

Jeffrey W. Graham
James G. Stringer, Jr.
Graham Computer Enterprises, Inc.
3 Office Park Cir., Suite 106
Birmingham, Al. 35223

ABSTRACT

The IBM Series/1 is IBM's first serious entry in the minicomputer industry. This paper presents an evaluation by a software consulting firm with experience in both minicomputers and mainframes. The information presented is based on extensive studies of the industry publications, IBM Series/1 manuals, and actual programming projects.

The hardware is evaluated from a programming viewpoint rather than the elusive number comparison approach. Furthermore, we evaluate the IBM software and lack of software. We describe our programming projects on the Series/1 and make predictions for the future of the Series/1 product line.

1. INTRODUCTION

The IBM Series/1 may or may not have a significant impact upon the minicomputer marketplace. It is clear, however, that the industry is looking at the Series/1 with tremendous curiosity. One only has to count the number of articles concerning the Series/1 which have recently appeared in computer trade journals to appreciate the fact that the Series/1 has not gone unnoticed. As one of the first purchasers of a Series/1, we are presenting herein our ideas, evaluations and expectations with respect to the Series/1.

2. OVERVIEW OF SERIES/1

The IBM Series/1 was officially announced by the General Systems Division of IBM in November, 1976. The true impact on the computer industry will not be known for quite some time. However, there has been a wide range of predictions and evaluations from all corners of the business world. The established minicomputer manufacturers disclaimed any competition with their products due to the limited software. Initial reports in industry trade journals and some financial analysts discounted the Series/1 as "technologically unexciting" (1) and showed tables indicating the Series/1 to be slower, smaller and more expensive than the competition (2). The OEM market initially turned away because there are no quantity discounts.

What is the real significance of the Series/1? What is its computing potential? What is its market potential? These questions cannot be answered by glancing at product announcement brochures or by shallow comparative statistics. This paper will present the opinion of our company and the actual experiences we have had programming the Series/1. Hopefully this information will present a more accurate view of the Series/1 for long-range evaluation.

The general architecture of the Series/1 is a 16-bit minicomputer with four hardware levels. Each level has eight general purpose registers. There are 256 possible I/O device addresses. Most devices with the Series/1 operate on a cycle Steal DMA with highly flexible channel programming capability. The instruction repertoire is impressive with hardware stacks, indirect addressing, autoincrementing of registers, variable length instruction operands and a complex but powerful addressing scheme.

The software announced with the Series/1 is extremely limited as reported by trade journals. The basic assembler is very close to the IBM 370 series assembler with full macro capability. The machine comes with limited stand-alone utilities but requires a rather large configuration to execute them. The Basic Program Preparation Facility (BPPF) provides only a text editor. The BPPF requires a large configuration and a license fee for the first 24 months for each machine. There are also I/O and task management routines called Control Program Support which provides some operating system type support. Again, a license fee on a per machine basis is required. Although CPS was announced with the Series/1, the availability of CPS at the time of this writing, is late Summer 1977. The documentation for the hardware and software is extensive and useful.

Our company was trying to select a minicomputer for basic software development for resale when the Series/1 was announced. Based on our background with IBM equipment, the marketing record of IBM, extensive analysis of the hardware, the excellent maintenance support from IBM, and our experience with other minicomputers, we decided to order a Series/1. We were fully aware that the first users of a new system will face frustrating problems unique to the burn-in phase of new hardware and software. However, we believe the new marketing strategy of IBM will be successful and the demand for quality software and experienced Series/1 personnel will offset the problems. Our system was installed late January, 1977 with virtually no problems.

3. HARDWARE

The Series/1 hardware represents a distinct difference from other small IBM systems. The interface capabilities provided marks perhaps the biggest change. It is actually easy to add a non-IBM device to a Series/1 system. As many as 256 device attachments are allowed. An attachment is the controlling interface which physically resides in the Series/1 chassis. Currently offered attachments provide for such functions as

* standard device control
* digital I/O (isolated or non-isolated)
* variety of communications
* non-IBM device control

The standard peripherals offered include a line printer (120 CPS, bi-directional),

CRT, diskette and disk (9.3 megabytes). Some obvious peripherals not available but which might prove useful in some applications are a high-speed printer, large disk storage and tape drive. The manufacturing quality of the equipment appears excellent especially when compared to other minicomputers. Our contention is that a cursory inspection does not do justice to the Series/1. For example, we have listed below some features of the Series/1 which might not be found as standard equipment on some of the minicomputers to which it was compared:

* eight general registers for each of the four processing levels
* immediate interrupt handler dispatching, i.e., no device address decoding is required
* hardware multiply/divide
* double word processing
* byte addressing and arithmetic
* bit set/reset
* memory-to-register and memory-to-memory operations
* immediate operand instructions
* full shift capability
* stack instructions with automatic limit checks
* priviledged instructions
* memory parity
* memory protection

An example of the power of the instruction set is one of the subroutine linkage combinations. Upon entry into a subroutine, one instruction stacks all or a specified number of registers, allocates a specified size work area in the stack, and sets a register with the address of the work area. When the subroutine is ready to return, one instruction unstacks the work area, restores the registers and branches through the return register.

Optional equipment offered on the model 4955 processor includes a floating point arithmetic unit and an address translation feature. The address translation feature provides the capability of defining an address field of twenty-four bits.

We are not trying to be salesmen for IBM but we are trying to point out that the trade journals have perhaps underestimated the Series/1.

4. SOFTWARE

The software announced for the Series/1 consists of three packages: Stand alone Utilities, Basic Program Preparation Facility (BPPF), and Control Program Support (CPS). The utilities are free of charge and provide the basic functions. The BPPF requires a 24 month license fee and provides a limited text editor, full macro assembler, and a linkage editor. The CPS also requires a 24-month license fee and provides I/O modules, interrupt handlers, and task management. However, CPS is strictly a set of routines to be linkedited with the user's program. CPS is a long way from being an operating system such as those available from other minicomputer manufacturers. Furthermore, the current release date for CPS is late summer.

The real question about these tools from IBM is how effective are they in producing production software? They are certainly limited in many respects:

* They must run in stand alone mode and an IPL is required to go from one function to another.

* They require a Teletype compatible device for an operator station (not available from IBM).

* All disk and diskette files must be specified absolute -- i.e. as device address, cylinder, head and sector.

* There is virtually no flexibility in the configuration during development.

* The text editor is very limited in its repertoire of commands and is cumbersome to use.

* We have found several bugs in the BPPF programs.

However, the software story is not all bad and there are some real advantages:

* The macro assembler is excellent in design. The macro capabilities are identical to those provided with OS/360.

* The assembler provides a certain amount of optimization in that it will select the more economical form of a variable length instruction.

* The text editor and assembler will accept input from key to diskette offline devices.

* The documentation on operation and the internal logic is extensive and very well written.

* There were no problems in getting the software running.

Our evaluation of the software from our investigations is that those items which were designed to be used in future operating systems (e.g. the assembler and linkage editor) are well designed and competitive with any similar program in the market; however, the support for running these programs (e.g. the disk I/O routine) is a temporary solution and shows the limited effort in design and implementation. These temporary solutions are apparently intended to be dropped from the product line after GCE SOFTWARE operating systems are released in the future. Of course, this is merely our interpretation of the situation. "What can a user accomplish reasonable with these limited tools?" Firstly, if one is developing a small resident function for process control, real-time, or communications control, these tools are adequate. Secondly, these tools can be used to produce other tools.

Our company is primarily taking the approach of developing software tools for the Series/1. The first development was a set of structured programming macros to relieve some of the pain of writing assembler language programs. These macros provide the control structures of IF-THEN-ELSE, REPEAT-UNTIL, WHILE-DO, SEARCH and CASE. When used in conjunction with proper indexing and

a PRINT NOGEN statement, the assembler listing becomes much more readable and maintainable as shown in Figure 1. The macros were intentionally designed to be as close as possible to the control structures of PASCAL. The purpose behind that decision is to develop a PASCAL compiler for the Series/1 later this year.

Our second project on the Series/1 was a limited BASIC interpreter. The design of this interpreter is similar to those found in the computer hobbyst publications. Our original intent was to simply have a training exercise and provide a demo package for the Series/1. However, there have been so many inquires for a BASIC interpreter that we have begun development of a commercial multi-user BASIC.

As a result of our initial development efforts, we are painfully aware of the need for a multi-terminal, more easily used text editor and are developing one. The machine time is too valuable to remain idle while a single person is making changes to text. Of course, due to no operating system, text editing cannot be shared with assemblies or test runs. We fully expect IBM to provide a solution to that problem within the near future.

Once the software tools are developed such that programmer production levels are reasonable, one must begin to develop production applications. This leads to another software problem area, Control Program Support. The major problem with CPS at this stage is the limited information available and the availability date. However, from the information we have gathered, CPS offers the following features:

1.) READ/WRITE level I/O support for operator station (teletype), disk storage, diskette storage and printer. It does support multiple units of the same type.

2.) Task Management Functions: Task Dispatcher, Basic Overlay Support, Timer Support and Supervisor Call.

3.) Bootstrap and loader routines.

4.) Error logging to disk or diskette.

The major disadvantages of CPS are the following:

1.) No data management or named file support.

2.) Modules must be linkedited into every load module and remain resident.

3.) Uses large amounts of core for the functions provided.

4.) No device independence in I/O routines.

5.) No support for display, DI-PI/DO, or communications. (Some additional RPQ's are becoming available for these devices.)

For the commercial software house, there is the disadvantage of CPS being a licensed product on a per/machine basis.

Future software development by IBM for the Series/1 will depend a great deal on what their marketing strategy is. Since their current marketing thrust is toward the large, "sophisticated" user, the new tools developed will probably be aids for a systems programmer. Producing a COBOL or RPG compiler would provide systems houses with tools to produce systems which compete directly with the System/32 and System/3; therefore we don't think those type tools will come out of IBM in the near future. However, we do think these general purpose support tools will be produced by companies like ours as well as the larger users. Already under development around the country are system software packages which will compete with CPS. The point is, we think the quality of the hardware is good enough to cause the fairly rapid evolution of software either within or outside IBM.

5. FUTURE FOR THE SERIES/1

In our opinion, the Series/1 architecture will adequately meet the needs of commercial process-control, communications controllers, distributed processing, small business systems, and the smaller OEM firms. The peripheral manufacturer should soon fill the gap on available peripherals for stand alone business applications where larger volumes of data must be processed and maintained.

The future for the minicomputer industry is one of increasing applications, increasing processing power and decreasing hardware costs. The real future for the Series/1 is the ability of IBM to capture a significant portion of that market. IBM has certain advantages when pursuing the market. Namely, IBM has a significant customer base which will purchase a significant number of systems. IBM has superior marketing support and organizations. Furthermore, IBM has produced quality hardware.

However, the weak link seems to be the software. Of course, that weakness will be diminished significantly by firms such as ours and by software released from IBM. Our opinion is that IBM software will have several drawbacks:

1.) IBM software will not be available soon.

2.) IBM will probably generate operating systems which provide too many services and consume excessive overhead.

3.) The small end of Series/1 configurations will probably not run the operating systems.

The software market from non-IBM suppliers will be heavily competitive and those firms which produce quality software backed by reasonable prices and marketing will be highly successful.

In the long run, the Series/1 currently available will probably be the beginning of a long line of products and services directly competitive with the established minicomputer manufacturers. In our opinion, the potential to capture a large share of that market is definitely there and IBM is one of the best at making potential become reality.

```
          PRINT NOGEN
GCECNV    CSECT
 STM    R6,(R1),WRKLNTH          SAVE REGS AND ALLOCATE WORK AREA
   USING PARMS,R0                SET UP BASE REGS
   USING WRKAREA,R6
   MVWS  SOURCE,R2               GET ADR OF INPUT STRING
   MVWS  TARGET,R3               ..AND OUTPUT STRING
   DROP  R0
   MVW   TYPECNT,R4              GET TYPE AND COUNT
   SRLD  12,R4                   SEPARATE TYPE AND
   IF    (SRL,4,R5),P,THEN       ..IF COUNT IS POSITIVE THEN
     CASE  R4,TYPATH             PROCESS CONVERSION
     CASEOF TYPETA,TYPATE        CASE: E TO A OR A TO E
       IF(CBI,TYPETA,R4,E,THEN      IF EBCIDIC TO ASCII
         MVA     EBCDI,WRKWD1
         MVA     ASCII,WRKWD2
       ELSE
         MVA ASCII,WRKWD1
         MVA EBCDI,WRKWD2
       ENDIF
       MVWI  TBLCNT,WRKWD3       PUT LENGTH IN CWKAREA
       RBTWI CHEX,WRKFLG         AND RESET FLAG FOR CNV TO HEX
       REPEAT                    BEGIN CONVERSION LOOP
         MVW   WRKWD1,RL         GET CONVERT- FROM TBL
         MVW   WRKWD2,R0                    - TO TBL
         MVWITBLLCNT,R7                     - LENGTH
         STRTSRCH REPEAT                     LOOP UNTIL END OF TABLE
           EXITIF (CB,(R1)+,(R2)),E,THEN     IF CHAR FOUND & INCR TBL
*                                            ..POINTER
           ORELSE
             ABI   1,R0                   GET NEXT CHAR
           ENDLOOP UNTIL,(JCT,R7)         LOOP TO END OF TABLE
         ENDSRCH
         ABI   1,R2              INCREMENT SOURCE PTR
         MVB   (R1),(R3)+        PUT CHAR IN TARGET & INCREMENT
       UNTIL (JCT,R5)            LOOP UNTIL SOURCEEXHAUSTED
*
           .
           .
           .
```

Figure 1.

REFERENCES

1. Angeline Pantages, Datamation, Dec. 1976, pp 146-147

2. George W. Potts, Research Note, Dec. 1976

3. Look Ahead, Datamation, Jan. 1977, pp 16

4. Computerworld, Jan. 31, 1977, pp 49-54

5. Computerworld, Nov. 22, 1976, pp 1

6. Computerworld, Dec. 5, 1976, pp 47

7. Computerworld, Feb. 21, 1977, pp 50

8. Minicomputer News, Dec. 2, 1976, pp 1

9. Business Week, Nov. 29, 1976