No. 7
4July79

University of Queensland

PRENTICE COMPUTER CENTRE

MINI/MICRO NEWSLETTER

Contents

Authorisation: Director of the Centre

1.Ø WORDS FROM THE GURUS

"As fuel costs continue to climb we may decrease the use of the automobile as an auxiliary information gathering tool and begin to move knowledge toward the person via personal terminals."

Keith Uncapher
ICCC Kyoto 1977

2.Ø DIRECTOR'S COMMENTS

Elsewhere in this issue (Section 5 PDP11 User Group News) reference is made to the possible need to charge users some fee for the distribution and maintenance of PDP11 software. This was initially established as a free service when it was thought that the number of systems involved would allow the work to be performed with a marginal amount of effort. As you may be aware there has been a substantial growth in mini-computers and we are now supporting around 4Ø systems involving the efforts of a full time programmer.

Following a submission from the Computer Centre the Finance Committee has agreed to subsidise the cost of this work for 1979. The policy regarding funding of this activity for subsequent years has been referred by the Finance Committee to the Vice-Chancellor's Computing Resources Review Committee. In the University News No. 13Ø of June 2Øth this Committee has inserted submissions.

Director
extension 2189

3.Ø SOFTWARE NEWS

3.1 FORTRAN IV Version 2.1 for RT-11 V3B

This software has recently been received from DEC and arrangements may be made re distribution by phoning extension 3938. Prompt service may be approximated by supplying a disk for the compiler and relevant FORTRAN libraries, and a note describing the system on which it will be run and the operating system most often used e.g. SJ or XM.

Bryan Claire
extension 3938

3.2 FORTRAN Libraries under RT-11

Since FORTRAN IV Version 2 and later compilers support Virtual arrays stored in memory which is not normally accessible to an RT-11 program (above 32K) it is important to realise the significance of the 4 different libraries that can be supplied with the compiler.

Firstly, as with Version 1 compilers and libraries, it is possible to have code added to the user's program to check for any array operations going out of bounds. This obviously adds some overhead to the program, so once a program is working properly, it is possible to link the program to the library that does not have bounds checking. This reduces the overheads when the user goes into full production.

With the introduction of virtual arrays and the XM (eXtended Memory) monitor, we find that SJ/FB and XM monitors handle the extended memory in different ways; in fact SJ and FB monitors do not want to know about the extra memory - the program has to do it all itself. So for each of the libraries described above, we have to introduce a library for SJ and FB monitors and a library for XM monitors, giving a total of 4 possible libraries. It does not matter which monitor is running when the program is linked to a particular library; it is important to link the program to the library suited to the monitor under which the program will be running if Virtual arrays are going to be used.

Bryan Claire
extension 3938

3.3 Problems with DSCSYS.SYS

A problem has been observed using DSCSYS.SYS (the stand-alone Disk Save and Compress utility for FILES-11 volumes) when used on RLØ1 disks. It appears to be possible to split a contiguous file across a bad or replaced block, without showing any differences during the verify pass. The task still has the contiguous bit set in the directory, but if it is a task file, it will not load.

We have received another version of DSCSYS.SYS from DEC that appears to have cured this problem. We also have a task that will scan the disk file structure for contiguous files that are not really contiguous.

Bryan Claire
extension 3938

3.4 DEC10 Cross Assembler for Micro-processors

A very recent addition to the DEC10 DECUS library is the CROSS program which is able to translate source code for 6502, 6800, 8080, 8085, Z80, CDP1802 and 8008 micro-processors. The syntax is very similar to MACRO-11 but using source instruction format appropriate to the particular micro-processor.

Features of CROSS not found in other cross assemblers include full MACRO-11 compatible MACRO and conditional assembly, extensive listing control, optional cross-reference listing, and optional instruction timing information.

Further information on CROSS can be found in the files MXI:CROSS.MEM and MXI:CROSS.HLP.

Output from CROSS is in PDP11 absolute format rather than the more usual MOTOROLA or INTEL ascii hex format. Work is in progress for a conversion program.

Following is a segment of MOTOROLA 6800 code in MOTOROLA cross assembler (M68XAS) format followed by equivalent code acceptable to CROSS. The assembly using M68XAS required 2.3 sec CPU time and cost $0.55, the assembly using CROSS required 0.34 sec CPU time and cost $0.14.

Program in M68XAS format

```
         NAM FORMAT A DISC TRACK
         ORG    $DD00
BEGA     EQU    $FF80
ENDA     EQU    $FF82
         LDX    $$D9E7  ISR
         STX    $FFF8
         CLI
START    LDX    #INDEX+7 END OF ADDRESS RECORD
         STX    ENDA
         LDX    #INDEX+1      START OF ADDRESS RECORD
         LDA B  #$F4      WRITE TRACK CODE
         STA B  $FD40     GIVE COMMAND
LOOP     JSR    WRIT      WRITE ADDRESS REC
         LDX    #DATAS+130      END OF DATA REC
         STX    ENDA
         LDX    #DATAST START OF DATA REC
         JSR    WRIT      WRITE DATA REC
         LDX    #INDEX+7      END OF NORMAL ADDRESS RECORDS
         STX    ENDA
         LDX    #INDEX+1      START OF NORMAL ADDRESS REC
         LDA A  INDEX+4 GET SECTOR NUMBER
         INC A
         STA A  INDEX+4 NEW SECTOR NUMBER
         CMP A  #27            FINISHED?
         BNE    LOOP
```

```
EXIT    RTS
*
*
WRIT    LDA B   #$02        DRQ?
X1      BIT B   $FD40
        BEQ     X1
        LDAA X  ;GET CHAR
        STA A   $FD43            INTO DATA REG
        INX                      NEXT CHAR
        CPX     ENDA             DONE ALL CHARS?
        BLE     X1               NO
        RTS
INDEX   FCB     $FC,$FE,00,00,01,00,$F7
DATAST  FCB     $F8
        FCB     00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
        FCB     00,00,00,00,00
        FCB     00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
        FCB     00,00,00,00,00
        FCB     00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
        FCB     00,00,00,00,00
        FCB     00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
        FCB     00,00,00,00,00
        FCB     00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
        FCB     00,00,00,00,00
        FCB     00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
        FCB     00,00,00,00,00
        FCB     $F7
        END
```

Program in CROSS format

```
          .TITLE FORMAT A DISC TRACK
          .      =       $DD00
          BEGA   =       $FF80
          ENDA   =       $FF82
          LDX    #$D9E7 ;ISR
          STX    $FFF8
          CLI
START:    LDX    #INDEX+7      ;END OF ADDRESS RECORD
          STX    ENDA
          LDX    #INDEX    ;START OF ADDRESS RECORD
          LDAB   #$F4      ;WRITE TRACK CODE
          STAB   $FD40      ;GIVE COMMAND
LOOP:     JSR    WRIT       ;WRITE ADDRESS REC
          LDX    #DATAST+130      ;END OF DATA REC
          STX    ENDA
          LDX    #DATAST   ;START OF DATA REC
          JSR    WRIT       ;WRITE DATA REC
          LDX    #INDEX+7       ;END OF NORMAL ADDRESS RECORDS
          STX    ENDA
          LDX    #INDEX+1      ;START OF NORMAL ADDRESS REC
          LDAA   INDEX+4   ;GET SECTOR NUMBER
          INCA
          STAA   INDEX+4   ;NEW SECTOR NUMBER
          CMPA   #27      ;FINISHED?
          BNE    LOOP
EXIT:     RTS
;
;
WRIT:     LDAB   #$02      ;DRQ?
X1:       BITB   $FD40
          BEQ    X1
          LDAA   0,X      ;GET CHAR
          STAA   $FD43      ;INTO DATA REG
          INX          ;NEXT CHAR
          CPX    ENDA      ;DONE ALL CHARS?
          BLE    X1      ;NO
          RTS
INDEX:    .BYTE  $FC,$FE,00,00,01,00,$F7
DATAST:   .BYTE  $F8
          .REPT  6
          .BYTE  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
          .ENDR
          .BYTE  $F7
          .END
```

Arthur Hartwig
extension 3021

3.5 RT11 Documentation Update

The following documentation update notices have been issued by DEC:

(a) RT-11 System Release Notes Update No. 1
(b) RT-11 Advanced Programmer's Guide Update No. 1
(c) RT-11 Advanced Programmer's Guide Update No. 2
(d) RT-11 Advanced Programmer's Guide Update No. 3

Due to the price of these ($4.50 each) it is intended to keep the costs for University departments to a minimum by making available photocopies of items (a) and (b) above, at five cents per page (95 cents and 40 cents respectively).

Orders may be placed through the Prentice Computer Centre by filling in and returning the order form on the back of this newsletter.

The information included in these updates is as follows:

(a) Information about the RX02 diskette device.
(b) Information concerning engineering changes made to the RL01 disk hardware.
(c) Information concerning arguments for the .MTIN and .MTOUT multi-terminal programmed requests, and differences between V2 and V3 device handlers in processing end-of-file conditions.
(d) Additional I/O information; specifically

1. I/O data structures (monitor device tables, queue element formats, etc.)
2. The flow of events in I/O processing.
3. A detailed study of the RK and PC device handlers.
4. System device handler requirements.
5. RT-11 file storage and file formats (REL, OBJ, LDA, SAV).
6. Device directories.
7. Magtape structure and label formats.
8. Cassette structure.

3.6 Fortran Documentation

DEC have released documentation update #1 for the PDP11 Fortran Language Reference Manual to make this manual reflect the Version 2 features of Fortran IV V2. Orders may be placed through the Computer Centre by using the order form on the back of this newsletter.

Bryan Claire
extension 3938

3.7 DDT11 Corrections and Enhancements (Decus #11-313)

DDT has been patched to correct its decoding of the SOB instruction. The errors caused the branch address to be wrong on type-out and an incorrect offset to be calculated on instruction type-in.

DDT has been enhanced by the addition of a facility to use the linker or task builder to construct the symbol table. If DDT is assembled with the symbol EXTSTB defined, then another module SYMTAB.MAC is used to construct the symbol table. To use this facility the SYMTAB macro definition in SYMTAB.MAC is edited to include references to all global symbols to be initially in the symbol table. How to do this is explained in SYMTAB.MAC itself. The symbol SYMTSZ is used to specify how many symbols the table is to be built for. The symbol table will also be built so that at least 8 more symbols may be added to it after linking.

To use DDT11 under RT-11 the recommended procedure is to define $$RT11 and XTCOM. Then if a control-C is typed on the console while DDT11 is looking for input it will exit to the RT11 monitor.

Copies of the new version may be obtained by arrangement with Bryan Claire (extension 3938).

4.0 LETTERS TO THE EDITOR

Contributions to this section of the newsletter should be addressed to

The Editor, Mini-Micro Newsletter
Prentice Computer Centre
University of Queensland
St. Lucia 4067

(letter from Geoff Smith, Social and Preventative Medicine)

(1) Attached is a list of programs used by this department mainly in the context of analysing questionaire type data. You may like to include them in the mini/micro newsletter. However an alternative method of exchanging programs among the PDP-11 installations could be the following:

(a) Each installation could be encouraged to maintain a brief computer index of programs available (similar to the attached list perhaps).

(b) At regular intervals, say 6 months, each installation could forward about thirty copies to

the Computer Centre for collating into folders. Each installation could then get back a folder indexing all available programs on the University's minis.

A system such as this, or something similar, could perhaps be better than having programs scattered throughout many issues of the mini/micro newsletter (or perhaps such a system could be used as well as newsletters).

(2) Something we are likely to need this year is one or more random number generators, with known properties, for the PDP-11. If you know of anyone who may have some experience with R.N. generators I would be grateful if you could let me know. (Any PDP-10 R.N. generators which could be adapted for PDP-11's?).

Editor's Reply:

Thanks for your suggestion. The list which you submitted has not been published due to its size. Your suggestion (b) will be adopted in that representatives of each installation who forward to me by the end of August a number of copies of a program index will each receive a copy of the other indices sent to me. This may help to avoid some "re-invention of the wheel".

I can't help you with sources for random number generation but Knuth in "The Art of Computer Programming - Volume 2 (Seminumerical Algorithms)" may provide some help if you wish to write your own generators. Perhaps some readers could help.

Arthur Hartwig
extension 3021

5.0 PDP11 USER GROUP NEWS

At the annual general meeting of the PDP11 User Group held on March 21 the following people were elected to the executive.

G. Smith	Social and Preventive Medicine
R. Owen	Pharmacy
G. Dowideit	Surveying
J. Fairbairn	Griffith University
K. Kratzmann	Computer Science

After a talk by Mr. A. Coulter, Director, Prentice Computer Centre, on the topic "Prentice Computer Centre Software Services and Maintenance for Mini-computers" there was a lively

discussion out of which the following motions were passed:

1. "The PDP11 User Group requests the Prentice Computer Centre to continue to provide comprehensive software support, acquisition and distribution for PDP11 computer systems within the University of Queensland and Griffith University".
   Moved J. Fairbairn, seconded G. Dowideit and passed unanimously.

2. "The PDP11 User Group requests the Director of the Prentice Computer Centre to supply heads of departments owning PDP11s with a list of the proposed charges for software maintenance, acquisition and distribution."
   Moved H. Mugglestone (Education), seconded J. Fairbairn.

3. "The PDP11 User Group requests its executive to investigate possible sources of funds to pay for the Computer Centre software support services".
   Moved D. Woolard (Architecture), seconded G. Dowideit and passed unanimously.

6.0 SEMINARS

Seminars which may be of interest on PDP11 users will be held by the Computer Centre in room 2.14 of the Hawken Building at 4.00 pm. Topics and dates are

July 4	High level languages for systems programming on PDP11s - C.
July 11	High-level languages for systems programming on PDP11s - BLISS-11.
July 17	High level languages for systems programming on PDP11s - panel discussion.
Aug 1	Microprocessr Development Facility

Users with suggestions for other topics may forward their suggestions to Sarah Barry (extension 3941).

7.0 M6800 MICROPROCESSOR SYSTEM FOR CONTROLLING AN X-Y PLOTTER

by Peter Craig (Dept. Electrical Engineering)

A system has been developed in which a microprocessor is used for the control of the pen movements of an X-Y plotter. This plotter controller system is in turn subject to the control

of a main computer system, which is also microprocessor based. The intention is that the main system will send data to be plotted to the plotter controller which then proceeds to carry out this task and signals the main system when it has completed this task. The system is described in detail in Ref. (1).

The principal design objective in interfacing the analogue X-Y plotter to the main computer system was to minimize the processing overhead imposed on this system. While this could have been partially accomplished by the use of fairly complex hardwired logic such as that described by Andrews (2) it was decided that a better approach would be to implement the pen control in firmwire. This approach leads to simplicity and hence minimum cost of the actual hardware interface to the plotter.

The minimum hardware approach means there is a significant amount of software required. This software is implemented on a dedicated plotter controller micro-computer. Thus most of the processing load associated directly with the control of the plotter is not imposed on the main computer system. Furthermore the availability of intelligence in the plotter controller allows the removal of ancillary tasks such as data scaling from the main computer system, thus further minimizing the processing load imposed on it.

The main computer system is an MC6800 based microcomputer system with a Shugart minifloppy disk drive and an ASR-33 Teletype. One of the uses of this system is real time data logging for experimental purposes. Comparing the speed of the M6800 microprocessor with the frequencies of some of the data it is trying to monitor, and also considering the long interrupt overhead time (approximately 25 uS), it was found to be desirable to remove as much as possible of the processing load associated with the plotter from this system. With a separate plotter processor in the computer system, the main system may be used to log data or perform calculations concurrently with the plotting of output, thus maximizing the throughput of the system.

The task of controlling the plotter is well suited to the capabilities of a microprocessor. While the capital outlay required for the microprocessor hardware is greater than that for a hardwired plotter controller, the microprocessor system offers much more powerful and versatile facilities. The system may be modified or extended by simply changing the software. Also, the maintenance cost would be lower for the standard microprocessor hardware (which uses LSI packages) than for a piece of very specialized hardware logic, using SSI and MSI technology.

Tasks performed by the Plotter Controller

The basic task of the plotter controller is to control the X and Y movements and up/down position of the pen of the plotter, in order to plot information such as graphs and diagrams. Since

the plotter is a mechanical device, it has a fairly slow response. This may be considered to impose velocity and acceleration limits on the movements of the pen, if the accuracy of pen positioning is to be maintained. The plotter controller must provide a series of points (ie. X co-ordinate and Y co-ordinate) to the plotter to guide the movement of a pen along a desired line segment. The techniques used to generate this series of intermediate points are referred to as "vector generation algorithms". These algorithms usually generate constant velocity vectors, making no allowance for the pen acceleration limit. The algorithms may be patched up so as not to exceed the acceleration limits at the beginning and end of a line segment. However, the author believes a simpler solution is to filter the outputs of a constant velocity vector generation system to limit the pen acceleration and deceleration. The vector generation algorithm used is known as a "symmetrical digital differential analyzer" (SDAA). This method was chosen for its ease of implementation, considering the limited arithmetic facilities available on the MC6800 microprocessor.

The other functions of the plotter controller are communication with the main computer system and interpretation of the plotting commands received from the main system. The communication with the main system is asynchronous and hence uses a handshaking protocol.

Implementation

The plotter controller hardware consists of a dedicated microprocessor system based on the Motorola MC6800 and its family of support devices. The control program for this system is stored in an EPROM (2708). The program communicates with the main system via a Peripheral Interface Adapter or PIA (MC6820), receiving from it blocks of commands and data which are stored in a buffer in RAM (MC6810's). The program then generates the required vectors and sends the points to be plotted to the plotter interface hardware as parallel digital data via two PIAs, one for each of the X and Y channels.

The plotter controller program has four main sections:

1. The PIA and working register initialisation routine which zeros the pen position.
2. The routine which communicates with the main system.
3. The command interpreter routine.
4. The plotting routines which perform the vector generation and the associated service subroutines.

Since the vector generation routine is controlling the pen movement in real time, there must be some provision for timing in order to control the pen velocity. This is accomplished by the use of time delay loops. The timing arrangements also indicate that only about one third of the available processing time is

actually used. Thus there is plenty of processing time available for the incorporation of more sophisticated facilities.

Interfacing Details

The plotter controller interfaces to the main system through an MC6820 PIA. The plotting data is transferred asynchronously from the main system through this PIA.

This interface to the plotter requires two PIAs, one for each of the X and Y channels. The outputs of these PIAs need to be latched in order to provide 10 bits of parallel data simultaneously for the digital to analogue converters. This is necessary because the data bus of the MC6800 is only eight bits wide.

The outputs of the DACs are filtered to limit the second time derivative of the DAC voltages to a finite value and hence limit the pen acceleration. The filter used is a Bessel or constant time delay filter. These signals are then fed to the X and Y inputs of the analogue X-Y plotter. Provision is also made for control of the pen up/down position by the plotter controller.

Summary

This article describes the hardware and software implementation of a dedicated peripheral controller using a microprocessor.

While the use of display processors is common in video graphics displays, very little work has been done on intelligent controllers for hard copy plotters. This is probably because of the lack of economic justification in dedicating a minicomputer to what, for it, is a fairly trivial task. The ready availability of microprocessor technology, however, now provides a comparatively inexpensive means of implementing an intelligent plotter controller.

References

(1) Craig, P.C., "A Microprocessor Controller Plotter", B.E. Thesis, U.Q. 1978.

(2) Andrews, C.J., "Development of a Microprocessor Controlled Digital Plotter", M.Eng.Sc. Thesis, U.Q. 1976

* * * * *

No. 7
4Jul79

The Supervisor,
Mini-Micro Services Group,
Prentice Computer Centre,
University of Queensland,
St. Lucia. Qld. 4067

Please supply the following items and charge all costs as indicated:

Quantity

(AD-52868-TI) RT-11 System Release Notes Update #1
.....($4.50)
(copy) RT-11 System Release Notes Update #1
.....($0.95)
(AD-5280B-TI) RT-11 Advanced Programmer's Guide Update #1
.....($4.50)
(copy) RT-11 Advanced Programmer's Guide Update #1
.....($0.40)
(AD-5280B-T2) RT-11 Advanced Programmer's Guide Update #2
.....($4.50)
(AD-5280B-T3) RT-11 Advanced Programmer's Guide Update #3
.....($4.50)
(AA-1855C-TI) PDP11 Fortran Language Referencce Manual Update 1
.....($4.50)

(Note: Copies available to University Departments only)

Charge to University Account No. .....

or Prentice Computer Centre Charge Code No. .....

Purchaser's Name: .....

Position: .....

Department: .....

Address: .....

.....

.....