INTEL® MICROCOMPUTER DEVELOPMENT SOFTWARE

We Speak Your Language.

The microprocessor is the essence of many of today's most advanced, exciting products. It has fostered today's most exciting, explosive industry. For the designer and for the consumer.

Intel has figured prominently in this incredible growth of designs, products, companies, and people. We think our success results from our directed effort to provide you, the designer, with all the tools you need to turn your microprocessor ideas and our microprocessor components into today's most exciting products.

All the Tools You Need.
Intel's Total Solution.

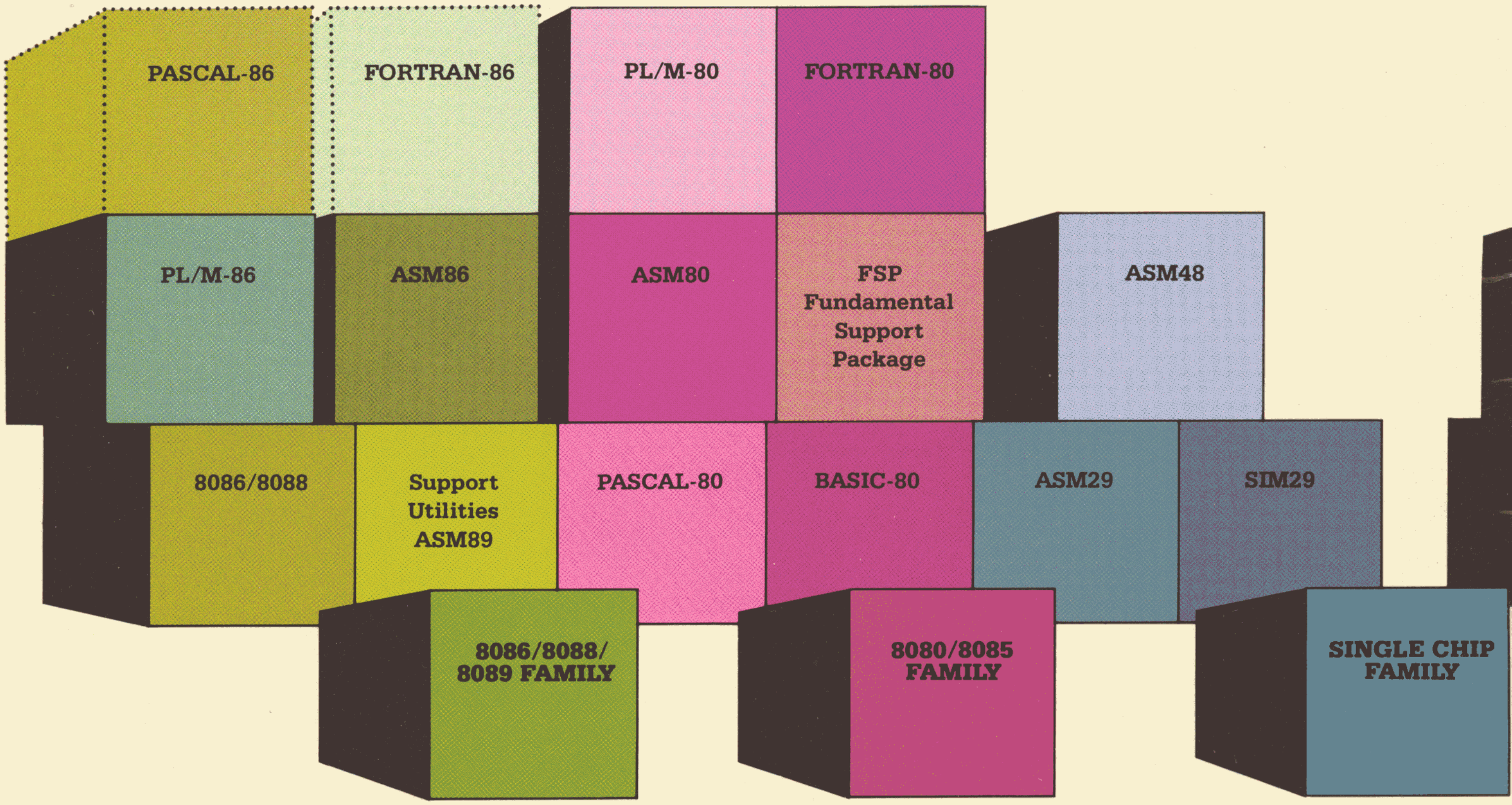
At every step in the design process, from conception to prototype to production, you can rely on Intel for the development tools you need for success. The Intellec Development System, ICE Emulation, trained Intel Field Applications Engineers, the Insite users' library. And software.

We know the moment you talk about developing your new product ideas with 16-bit or 8-bit microprocessors, you are talking about software. And for software, talk to Intel. In the Intel tradition, we've gathered a comprehensive and very flexible software product line. When you talk software, we speak your language.

Rising Programmer Costs.
Descending Component Costs.
The Intel Software Solution.

In today's topsy-turvy world of descending component costs and spiralling software development and maintenance costs, using the right language to develop your application becomes increasingly critical. The programmer efficiency gained by using high-level

**A complete array of software tools ready for your instructions.
Part of the total Intel solution for product development.**



languages, modular program development, and top-down design has become a major factor in reducing software development costs.

We have the software solution. Software designed for Intel processors. High-level languages. Our PL/M is the first high level language designed specifically for the microprocessor. We offer the first ANSI 77 FORTRAN compiler available for a microprocessor — our FORTRAN-80.

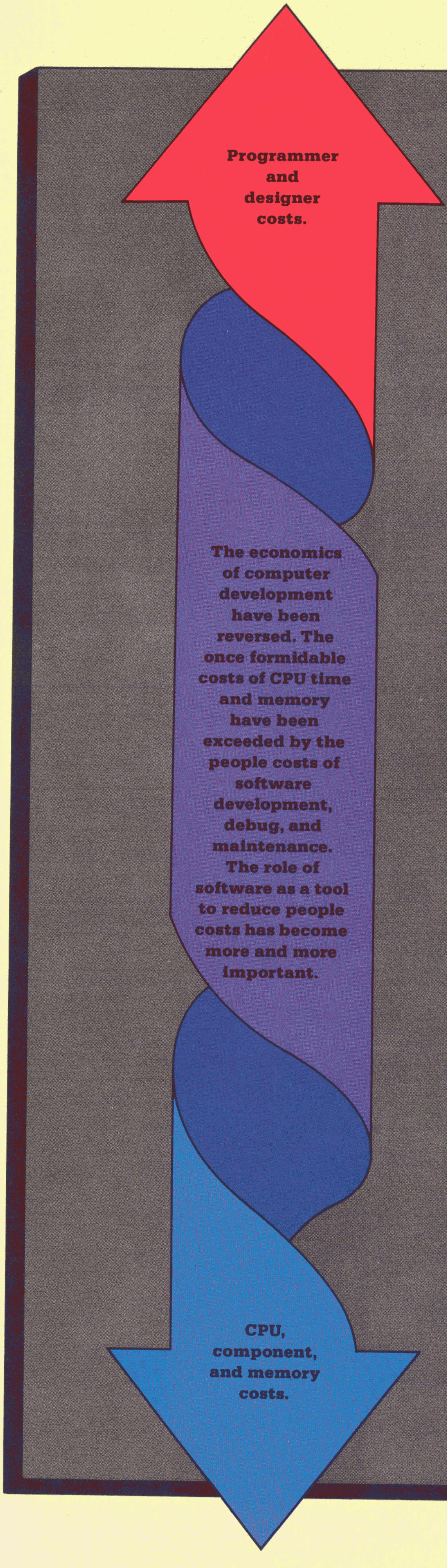
We have a BASIC interpreter for your engineering problem solving, a COBOL interpreter for your management problem solving, and the impressive PASCAL. All available now.

Of course, we have macro assemblers for every Intel microprocessor, plus a range of utilities, including linkage and relocation facilities, library management, mathematics and scientific functions, and the INSITE users' library.

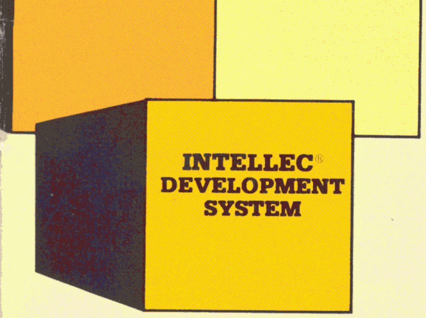
Armed with such a range of software, you can really put the Intellec Microcomputer Development system to work. The ISIS-II disk operating system will run up to four double density floppy disks, plus two single density floppy disks, and now supports Intel's new large-capacity hard disk. And CREDIT, our friendly screen-oriented text editor helps you get much more done with much less time at the keyboard.

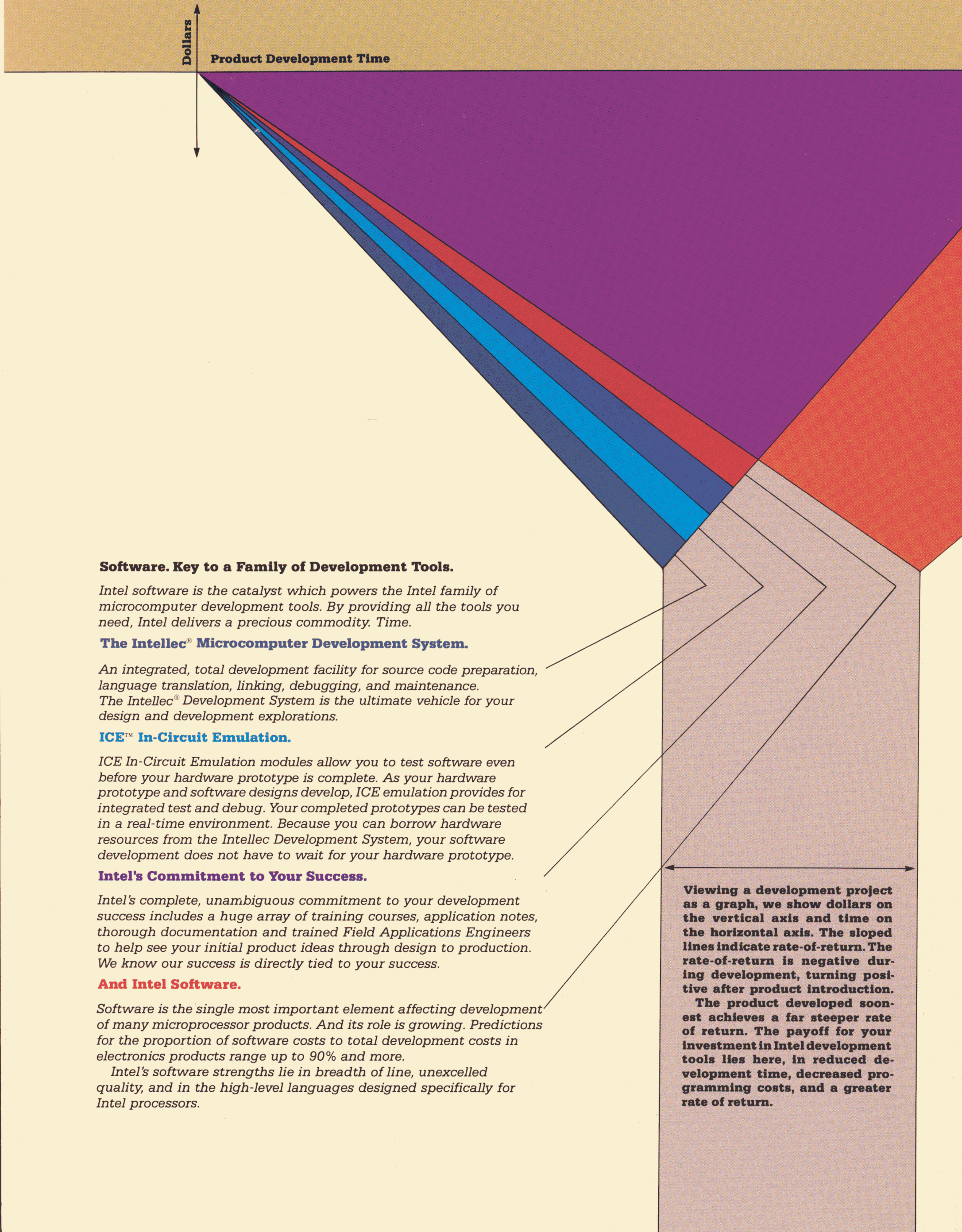
You know us for our outstanding components. We want you to know more about our outstanding development systems software too.

With our 16-bit, 8-bit, single chip and Intellec software families, we're ready for your instructions. Let us show you how our software can speed your development efforts.


Programmer and designer costs.
CPU, component, and memory costs.

The economics of computer development have been reversed. The once formidable costs of CPU time and memory have been exceeded by the people costs of software development, debug, and maintenance. The role of software as a tool to reduce people costs has become more and more important.





Software. Key to a Family of Development Tools.

Intel software is the catalyst which powers the Intel family of microcomputer development tools. By providing all the tools you need, Intel delivers a precious commodity. Time.

The Intellec® Microcomputer Development System.

An integrated, total development facility for source code preparation, language translation, linking, debugging, and maintenance. The Intellec® Development System is the ultimate vehicle for your design and development explorations.

ICE™ In-Circuit Emulation.

ICE In-Circuit Emulation modules allow you to test software even before your hardware prototype is complete. As your hardware prototype and software designs develop, ICE emulation provides for integrated test and debug. Your completed prototypes can be tested in a real-time environment. Because you can borrow hardware resources from the Intellec Development System, your software development does not have to wait for your hardware prototype.

Intel's Commitment to Your Success.

Intel's complete, unambiguous commitment to your development success includes a huge array of training courses, application notes, thorough documentation and trained Field Applications Engineers to help see your initial product ideas through design to production. We know our success is directly tied to your success.

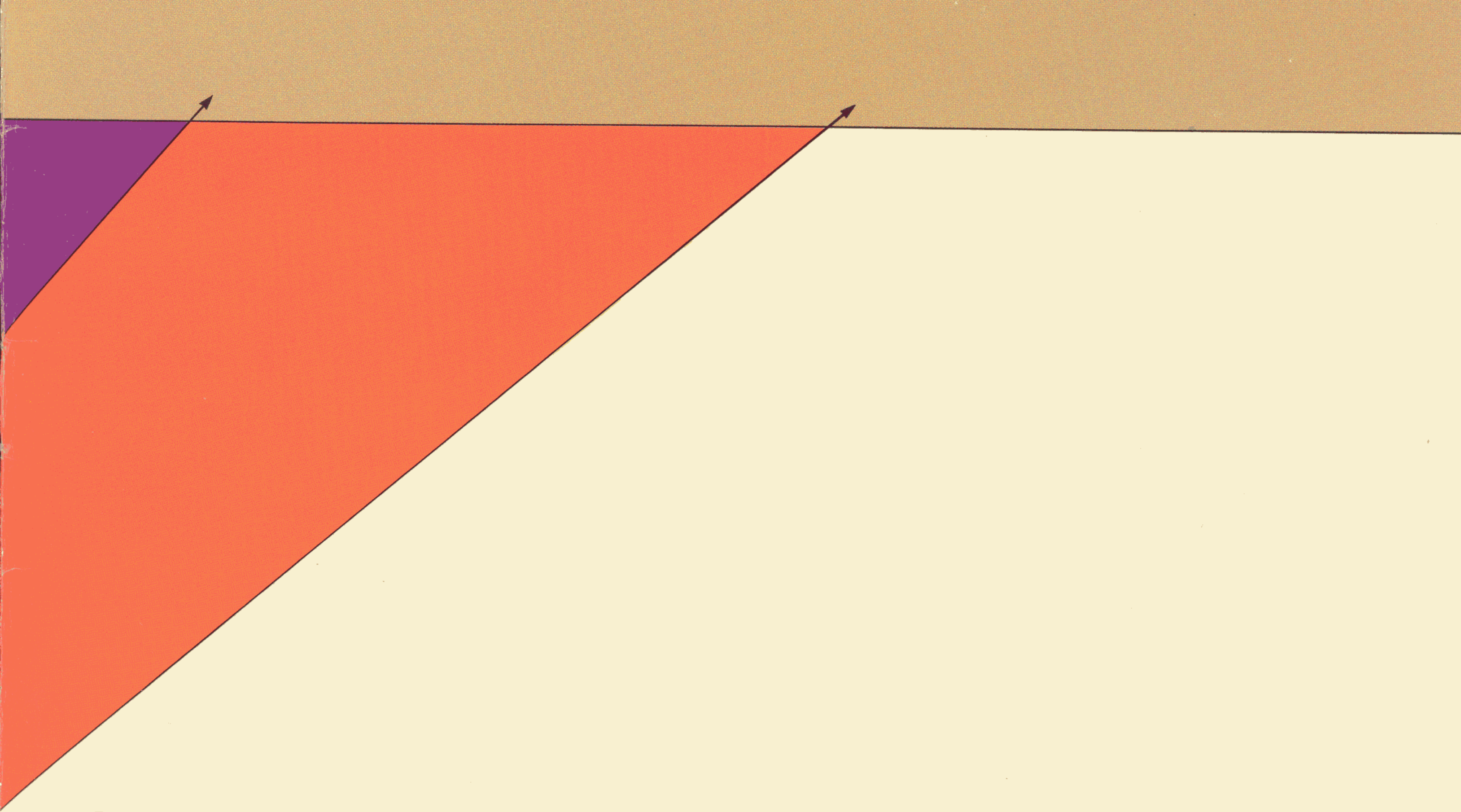
And Intel Software.

Software is the single most important element affecting development of many microprocessor products. And its role is growing. Predictions for the proportion of software costs to total development costs in electronics products range up to 90% and more.

Intel's software strengths lie in breadth of line, unexcelled quality, and in the high-level languages designed specifically for Intel processors.

Viewing a development project as a graph, we show dollars on the vertical axis and time on the horizontal axis. The sloped lines indicate rate-of-return. The rate-of-return is negative during development, turning positive after product introduction.

The product developed soonest achieves a far steeper rate of return. The payoff for your investment in Intel development tools lies here, in reduced development time, decreased programming costs, and a greater rate of return.



Success and Development Speed

The time between product concept and product delivery can be the most important period in the life of your product. In the lightning-fast world of microprocessor-based products, a difference of a few months in product availability can spell the difference between success and failure, profit and loss.

Hundreds of Intel success stories attest to our claim that Intel development tools can dramatically shorten development time for your microprocessor-based products. Because with Intel, all the design tools are on hand to initiate development on Intel microprocessors and carry your project rapidly to completion.

The Power of High-Level Languages.

High-level languages make programmers more productive. And the savings from the use of high-level languages only begin with faster coding.

Program maintenance, debug, and enhancements are easier when working in the self-documenting, natural language code of Intel's high-level languages. The languages are easier to learn, and easier to use.

The built-in applications-oriented functions and capabilities of our high-level languages enable the programmer to code targeted applications quickly.

And standardization of our compilers delivers the highly desirable advantages of program portability, and maximizes use of the programmer's experience. PL/M-86 is fully upward compatible from our PL/M-80, for example. FORTRAN-80 is an extended version of ANSI 77 standard FORTRAN. And our iCIS-COBOL meets and exceeds ANSI Level I standards.

The Power of Structured Programming.

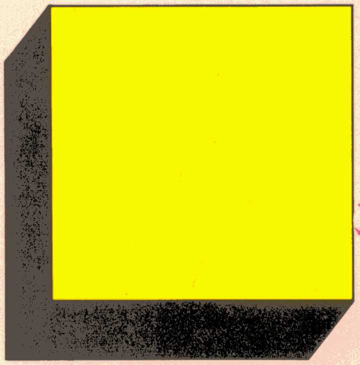
Without doubt, the movement toward structured programming techniques is the single factor which has contributed most to advancement of the programmer's art. PL/M is a block structured language, designed to support modular, top-down structured programming.

Power Beyond High-Level Languages.

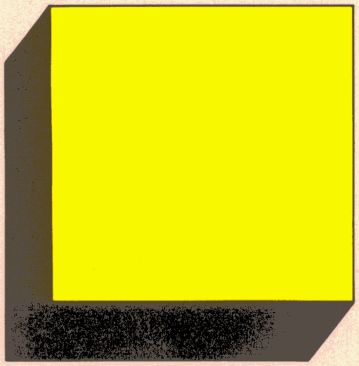
The power of Intel's software extends beyond high-level languages. Our easy-to-use, symbolic macro assemblers are available for each and every Intel microprocessor. And we provide a host of utility programs for the 8-bit and 16-bit software families.

The product line? A wide selection of software tools designed specifically to fully exploit the features of Intel's microprocessors, while delivering the keys to rapid development speed.

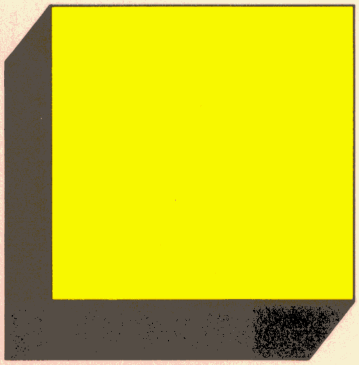
The result? You work with reliable, high-quality software products. You enjoy lower development and maintenance costs. And you produce a superior product at a substantially lower cost.



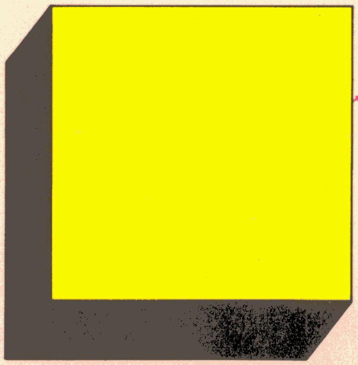
A module coded in PL/M for overall control and timing.



A module coded in FORTRAN for numeric processing.



Where speed and space are critical, the designer selects an assembler.



FSP (Fundamental Support Package) mathematical routines or routines from the INSITE™ users' library can save development time.

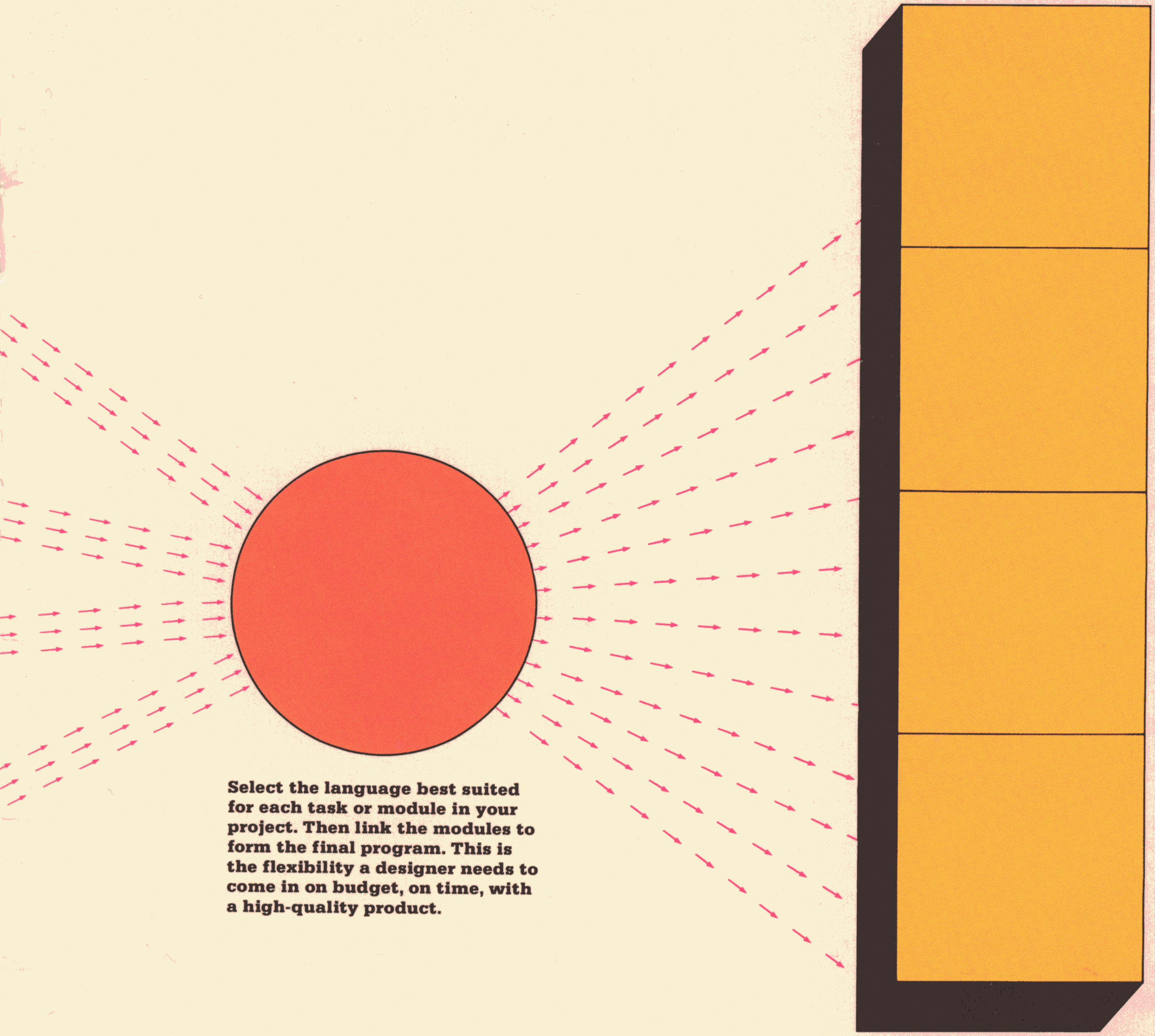
Modular Flexibility

By producing relocatable and linkable object code, our languages provide tremendous development strengths.

First, programs can be divided into logical, manageable modules. One module for I/O, for example. A separate module for numeric calculations. A separate module for overall control and timing.

Each module can then be developed in the language best suited to the application. PL/M's strengths in high-level hardware control ease the task of overall program control and timing, while FORTRAN's number-crunching power simplifies the programming of numeric calculations found in process control. Assembler modules provide the punch for time-critical or space-critical segments.

By designing your programs using well-defined module



Select the language best suited for each task or module in your project. Then link the modules to form the final program. This is the flexibility a designer needs to come in on budget, on time, with a high-quality product.

Only the affected modules need be reassembled or recompiled when changes are required.

interfaces, different programmers can code separate modules at the same time.

With modular programs, changes need only affect the module in question, and can be made more quickly, since only the affected module need be recompiled. Maintenance and enhancements are thus easier and less expensive.

When the modules are debugged, they are easily linked to form the final program. You don't have to worry about addresses, subroutine calls, or variable references between modules, because they are resolved automatically.

An Unexpected Benefit

When you apply Intel development tools to your software development projects, you will appreciate yet another benefit: more room in your budget. More room to explore, prototype, and test new features and capabilities. Maybe even something you just didn't have time or resources for before. When you apply the right tool to the job, you save time. You save effort. We have the tools.

We Speak Your Language, When You're Talking 16-Bit Product Development.

Intel's 16-bit development software is designed to fully exploit the advanced architecture of our 16-bit microprocessor family and single board computers.

The designer can choose from our PL/M-86 high-level compiler, or our ASM86 16-bit Macro assembler. Both produce compatible linkable and relocatable object code; both are modular. Each is a powerful tool for product development.

The languages are supported by the 8086/8088 family of support tools — utilities for linking, locating, library management, and conversion of 8080/8085 assembly language source to 8086/8088 assembly language source.

The Intel 8086/8088 software family includes the advanced 8089 Assembler Support Package for the 8089 processor.

Exciting New Products on the Way.

We're not stopping here. You can expect PASCAL and FORTRAN compilers for the 16-bit family in the near future.

PASCAL-86 will implement the newly published ANSI/IEEE/ISO standard language. FORTRAN-86 will be an upward compatible extension of our ANSI-1977 FORTRAN-80.

Of course, all our 16-bit development software (including upcoming PASCAL-86 and FORTRAN-86) executes on the Intellec Microcomputer Development System with the ISIS-II operating system, minimizing requirements for additional hardware or training for Intel Microcomputer Development System users.

If your object is product development using Intel's 8086/8088 processor family, you will want to read on.

Advanced PL/M-86

PL/M-86 is an advanced, structured, high-level programming language which allows users to achieve maximum benefits of Intel's 16-bit microcomputer, the 8086, and the 8-bit 8088. PL/M-86 is ideally suited to systems implementation.

PL/M-86 is the second generation of Intel's PL/M. And PL/M is the first high-level language designed specifically for the special requirements of microcomputers. In fact, the architecture of Intel's 16-bit family was conceived by the designers of PL/M. This co-ordination of hardware and software design is, as you would expect, effective and powerful.

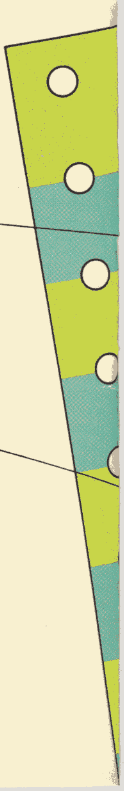
PL/M-86 includes features which allow the user high-level access to the microcomputer hardware, giving the designer close contact and control over the processor. At the same time, the language offers the natural advantages of any high-level language: the programmer writes code in statements which are more natural than assembly language; coding, modifications, and maintenance are easier and faster.

PL/M-86 remains upwardly compatible with PL/M-80, so you can take advantage of time and money invested in PL/M-80 training and programming. Most PL/M-80 source will directly compile under PL/M-86 to operate in a 16-bit environment.

PL/M-86 is a block structured language supporting structured, modular programming. This translates to greater control and management of your 8086/8088 software projects.

Compilation produces linkable, relocatable object code for greater co-ordination between programmers, and more flexibility in design.

PL/M-86 source is largely "self documenting." Long variable names, natural language statements, and structured design makes the source code easier to understand and use. This reduces programming, maintenance, and enhancement time and cost, increasing reliability. Debugging is faster; enhancements or later modifications are much easier.



Direct port I/O — PL/M-86 Inputs and Outputs data to the 8086 I/O ports directly.

PL/M-86 is a block structured language. "DO" blocks and procedures isolate tasks and restrict the scope of variables and labels. The flow of control constructs such as IF, THEN, ELSE allows top-down, structured programming.

Major features of the Intel PL/M-86 compiler include support for five data types (byte, word, integer, real, and pointer). The language supports 16-bit signed integer and 32-bit floating point arithmetic. The language also provides the advanced data structuring facilities of arrays and structures, allowing them in any combination. Built-in functions for string handling are provided (MOVE, COMPARE, TRANSLATE, FIND, SKIP, and SET).

PL/M-86 provides high-level hardware access via constructs for direct port I/O, interrupt handling, and direct memory access. Access is also provided to the hardware stack and the CPU flags. PL/M-86 supports the ICE-86 Emulator, and symbolic debugging.

A compile time option allows you to choose from three levels of code optimization—from a quick compile pass with no optimization to a full-optimization pass. Your final code is extremely memory efficient, yet in the early development stages the quicker, less efficient compiles save development time.

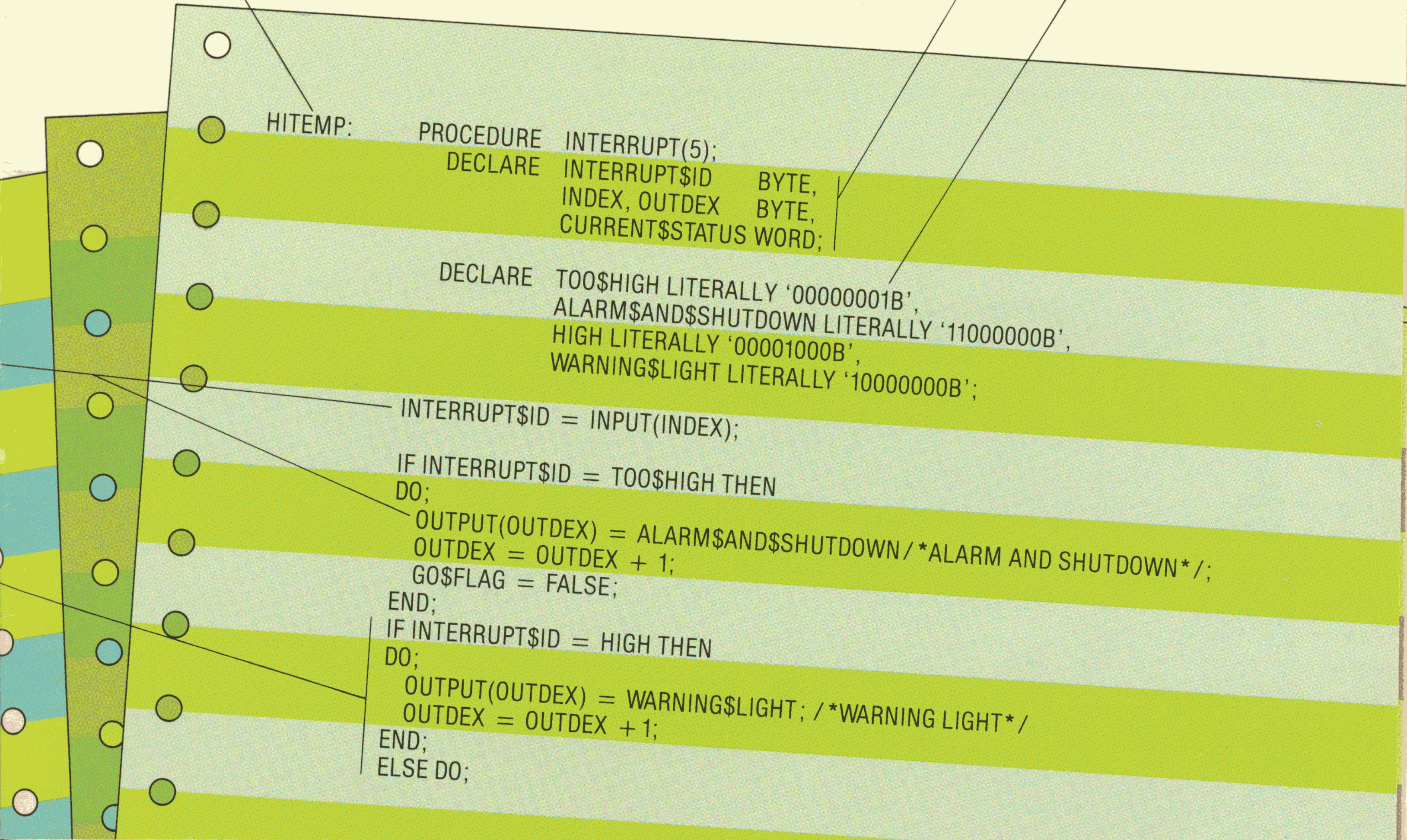
PL/M-86 allows you to write re-entrant procedures. And a built-in "lockset" function facilitates communication and coordination between multiple processors.

Modular programming with structured code and self-documenting source: a combination that results in high reliability and a real savings in development time and development cost. Compared with assemblers, PL/M-86 means significantly fewer statements. Its simplicity, power, and compactness ensure that PL/M-86 will join PL/M-80 as an industry standard.

PL/M-86 provides high level control of hardware resources. This example shows an INTERRUPT procedure for an 8086 system when a peripheral device initiates an "Interrupt 5."

PL/M-86 provides five data types. Here the BYTE (8-bit unsigned) and WORD (16-bit unsigned) are used.

The LITERALLY construct is one of many reasons PL/M-86 is easier to read and maintain.



```
HITEMP:     PROCEDURE  INTERRUPT(5);
              DECLARE  INTERRUPT$ID       BYTE,
                       INDEX, OUTDEX      BYTE,
                       CURRENT$STATUS WORD;

              DECLARE  TOO$HIGH LITERALLY '00000001B',
                       ALARM$AND$SHUTDOWN LITERALLY '11000000B',
                       HIGH LITERALLY '00001000B',
                       WARNING$LIGHT LITERALLY '10000000B';

            INTERRUPT$ID = INPUT(INDEX);

            IF INTERRUPT$ID = TOO$HIGH THEN
            DO;
              OUTPUT(OUTDEX) = ALARM$AND$SHUTDOWN/*ALARM AND SHUTDOWN*/;
              OUTDEX = OUTDEX + 1;
              GO$FLAG = FALSE;
            END;
            IF INTERRUPT$ID = HIGH THEN
            DO;
              OUTPUT(OUTDEX) = WARNING$LIGHT; /*WARNING LIGHT*/
              OUTDEX = OUTDEX +1;
            END;
            ELSE DO;
```

Support Utilities for 16-bit Development.

LINK86 combines separate assembled or compiled object code modules into the final program module. This allows one to subdivide a programming task into functional modules and subdivide the modules among different programmers. This method of modular development reduces programming, maintenance, and debug costs.

LOC86 relocation utility allows coding of program blocks without knowing the final location of object code modules in memory. Programs can be developed and the object code relocated to suit the specific application.

LIB86 LIBRARY MANAGER™ allows the user to create specially formatted files to contain libraries of object modules. The library modules can be reviewed, listed, and linked to other modules using LINK86. This promotes the use of standardized routines by all development team members, eliminating duplication of effort.

OH86 (Object Code to Hexadecimal Symbolic Object Code Converter) translates system generated absolute machine language into printable hexadecimal format, for use with the Intel Universal PROM Programmer.

CONV86 translates 8080/8085 assembly source to 8086/8088 assembly source. This provides you with an easier, cost-effective upgrade path from the Intel 8-bit microprocessor family to the Intel 16-bit family.

Designed for the Designer.

ASM86 produces relocatable, linkable object modules compatible with PL/M-86. This allows the designer to employ the extensive capabilities of this assembler in any application where time or space are critical, then link completed assembler modules with other ASM86 or PL/M-86 modules. And the benefits of modular programming are yours.

ASM86 8086/8088 Macro-assembler.

ASM86 is Intel's powerful "high-level" Macro Assembler for the 8086/8088 16-bit processor family.

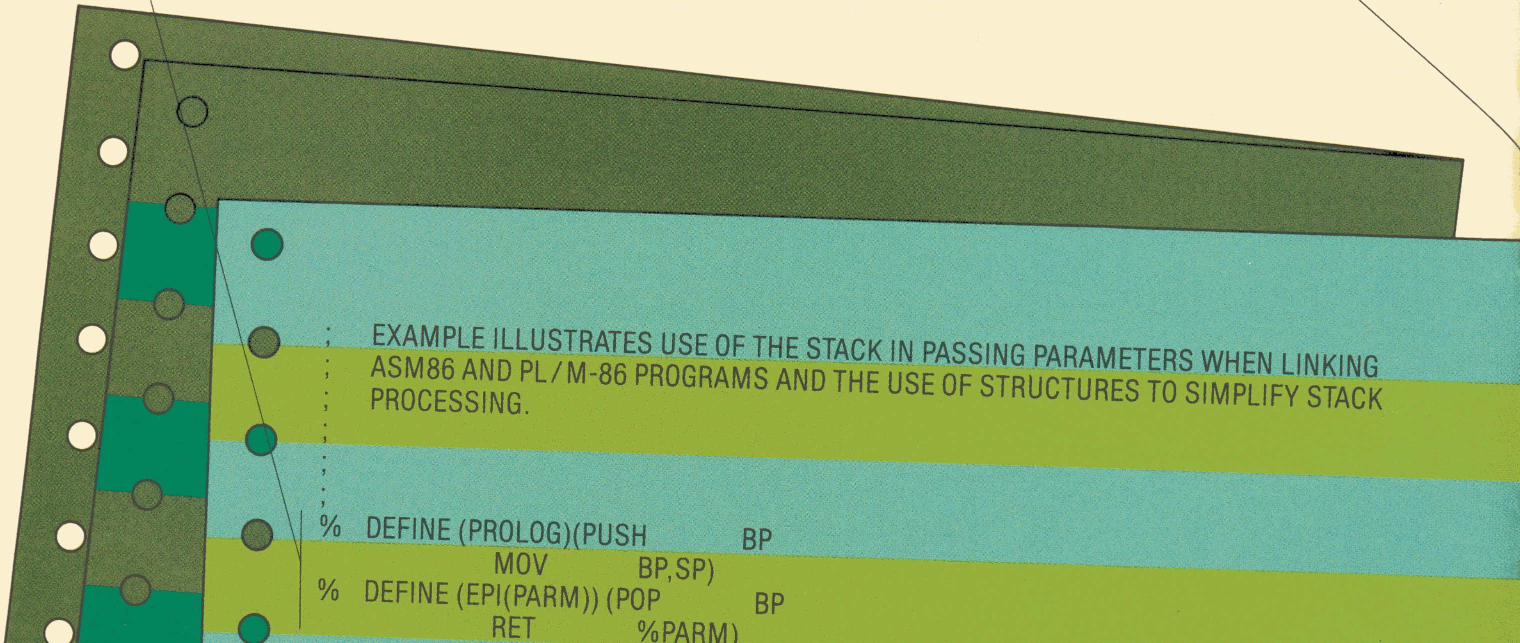
The language supports many high-level features not ordinarily found in an assembly language, including facilities for structures, arrays, and records. These features allow the user to achieve efficiency and machine control of the 8086 or 8088 in a relatively sophisticated fashion.

ASM86 is available when maximum code efficiency or hardware control is needed. The language is highly mnemonic and compact, with its 100 mnemonics representing over 3600 distinct machine instructions. The assembler is able to generate the proper machine instruction by evaluating the operand of the instruction. The user need only remember the generic mnemonic for the operation, thus easing the programming task. ASM86 will generate the shortest machine instruction possible given no forward referencing.

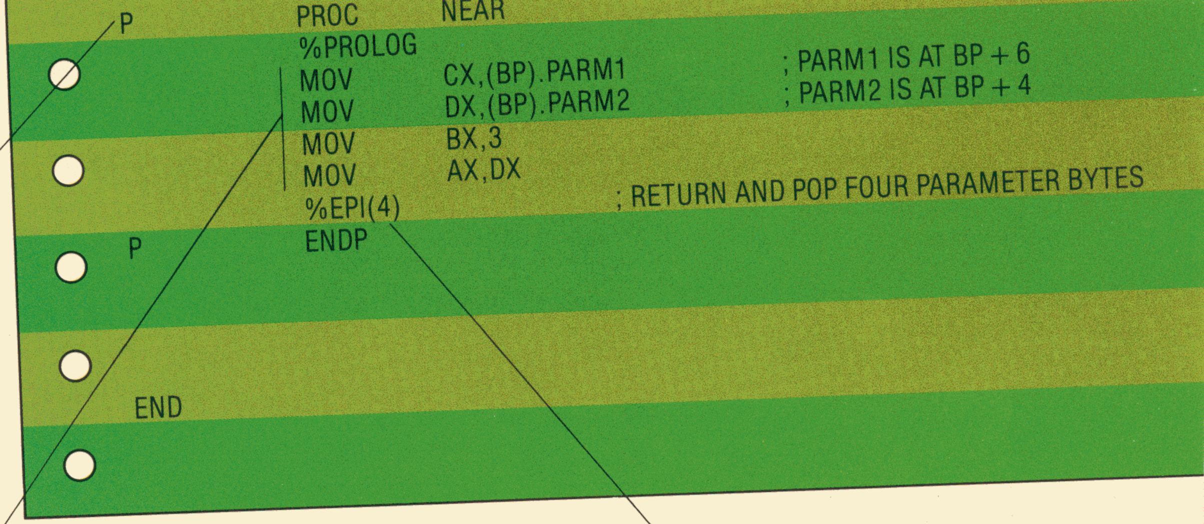
The 8086/8088 Assembly Language is strongly typed. This means that the assembler checks variable and label types, ensuring their consistent use throughout the program. This means that many programming errors will be detected when the program is assembled, long before the errors are found on hardware.

ASM86 provides a powerful and flexible MACRO facility. Here two macros are defined for later use.

ASM86 features a powerful STRUCTURE facility. A structure is defined to be a template, and can be used to allocate storage. Each field of a structure can be named.



```
;    EXAMPLE ILLUSTRATES USE OF THE STACK IN PASSING PARAMETERS WHEN LINKING
;    ASM86 AND PL/M-86 PROGRAMS AND THE USE OF STRUCTURES TO SIMPLIFY STACK
;    PROCESSING.
;
;
;
%  DEFINE (PROLOG)(PUSH          BP
               MOV          BP,SP)
%  DEFINE (EPI(PARM)) (POP       BP
               RET          %PARM)
```



```
P      PROC     NEAR
       %PROLOG
       MOV      CX,(BP).PARM1          ; PARM1 IS AT BP + 6
       MOV      DX,(BP).PARM2          ; PARM2 IS AT BP + 4
       MOV      BX,3
       MOV      AX,DX
       %EPI(4)                  ; RETURN AND POP FOUR PARAMETER BYTES
P      ENDP

END
```

The use of the PROCEDURE encourages modular programming.

Generic mnemonics simplify the programming task. One instruction mnemonic can be translated into many different machine instructions, depending on the operand used. Here, one move instruction is used to represent a move immediate data to a register or move register to register.

The previously defined macros are called.

A macro facility enables one to code frequently used sequences only once, using "dummy parameters," and to call this code sequence with a single instruction at assembly. The real values are then substituted for the "dummy parameters," and the code lines are assembled, all automatically.

ASM86 provides a more powerful and flexible text macro facility than normally found in an assembler. For example, while most macro assemblers allow one macro call per line in the operation field only, ASM86 macros may appear anywhere.

Powerful macro functions include conditional assembly, string-processing functions, repetition, and functions to support assembly-time input/output to the console.

Three listing modes are provided, including a GEN mode which provides a complete trace of macro expansions at all levels of nesting.

Naturally, ASM86 fully supports the extended addressing of the 8086/8088 microprocessors.

ASM89. A powerful assembler for the 8089 Input/Output Processor.

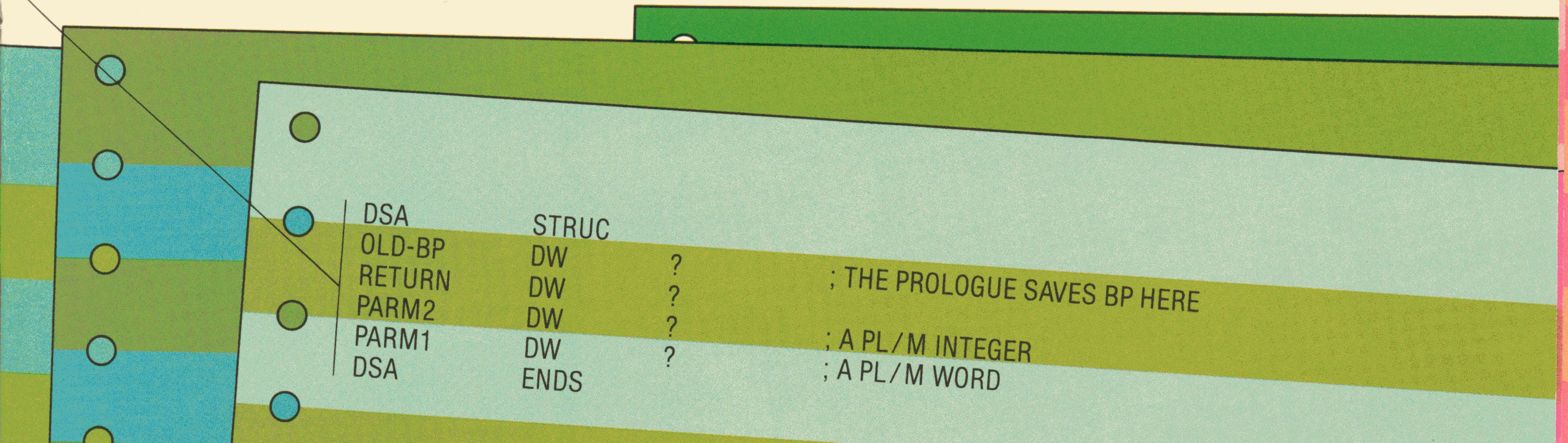
Intel's 8089 I/O processor, teamed with the 8086 or 8088 microprocessor, provides a total system solution for microprocessor applications.

We have a support package ready for your 8089 designs. The ASM89 assembler for the 8089 translates symbolic assembly source code into 8089 object modules. Linkage and relocation utilities are included.

The 8089 assembler provides maximum efficiency and control of the 8089 I/O processor. This assembler provides object module compatibility with the 8086 and 8088 microprocessors. This object module compatibility, along with the 8086/8088/8089 relocation and linkage utilities, easily supports the design of an 8089 into an 8086 or 8088 system.

ASM89 fully supports the based addressing modes of the 8089. A structure facility in the assembler provides easy access to based data. The structure facility allows the user to define a template which enables accessing of based data symbolically.

The 8089 assembler and utilities package operates on the Intellec Microcomputer Development System.



```
DSA       STRUC
OLD-BP    DW       ?        ; THE PROLOGUE SAVES BP HERE
RETURN    DW       ?
PARM2     DW       ?        ; A PL/M INTEGER
PARM1     DW       ?        ; A PL/M WORD
DSA       ENDS
```

We Speak Your Language When You're Talking 8-Bit Product Development.

When you're talking 8-bit product development, Intel's microprocessor leadership translates into comprehensive software experience and flexibility. Our development languages for the 8080/8085 family — PL/M-80, FORTRAN-80, and the ASM80 assembler — provide three linkable levels of software development capability. A further development is FSP, our Fundamental Support Package — nine mathematics libraries which may be linked into your final program to save development time.

And for projects employing the iSBC Single Board Computers, we've got PASCAL-80 and BASIC-80 to operate with the RMX (Realtime Multi-Tasking Executive) run-time packages.

Of course, all these languages execute on the Intellec Microcomputer Development System, and support ICE Emulation, with symbolic debugging.

To converse with the Intel 8-bit processor family, you need the right languages. They are available now, from Intel.

PL/M-80

PL/M-80 is Intel's dramatically successful high-level compiler for the 8080/8085 family. Offered since 1973, PL/M-80 was the first high-level language designed specifically for the special requirements of microprocessors, and has figured prominently in thousands of application successes.

And with good reason. PL/M-80 is easy to learn and use. Like other Intel 8-bit languages, it produces relocatable and linkable object code for modular programming.

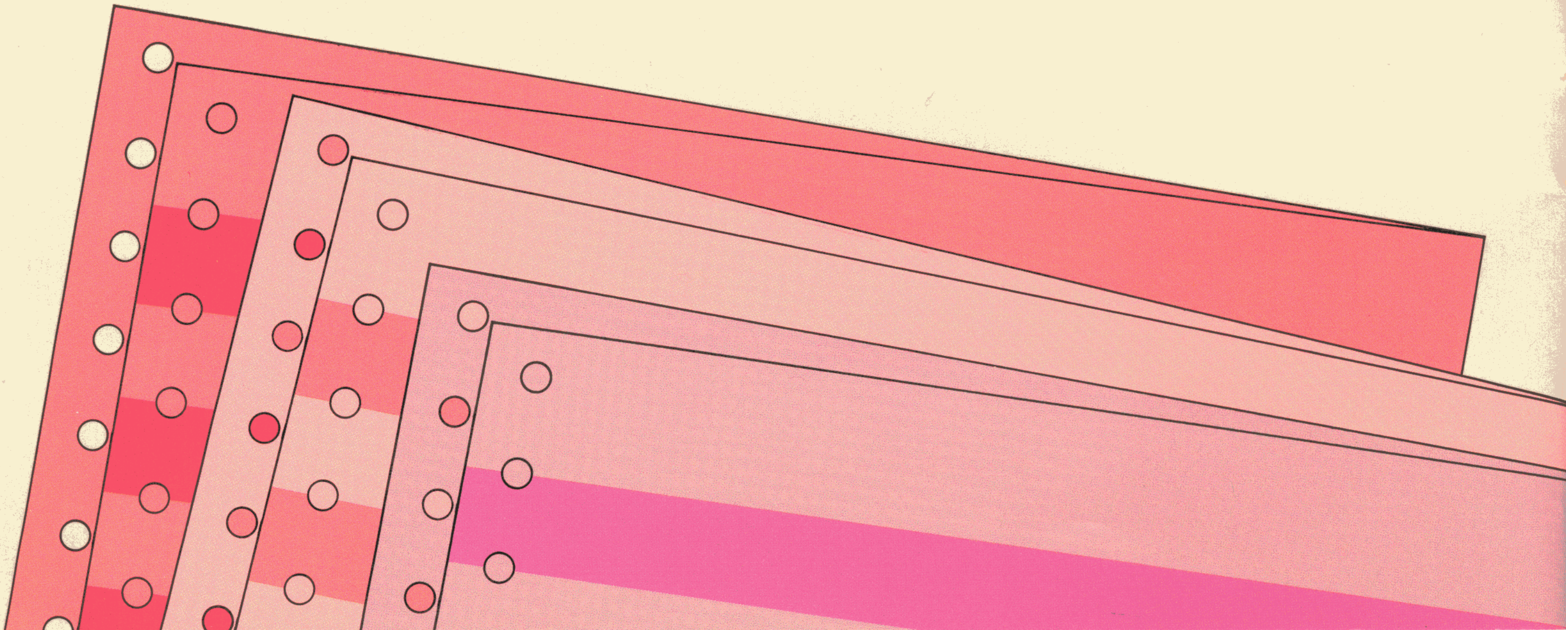
PL/M-80's strength lies in its powerful hardware and I/O accessing features. Like its 16-bit counterpart, PL/M-80 is ideally suited to systems' implementations (such as operating systems) and microcomputer language design.

PL/M-80's natural language syntax
+ *Code efficiency*
+ *Modular, block-structured language*
+ *Direct processor access and control*
+ *Processor capabilities reflected in language structure*
= *Optimal High Level Language for Microprocessor Software Development.*

PL/M-80 uses logical, natural language statements, and is virtually "self documenting." Well suited to structured programming techniques, the language produces highly reliable, maintainable code.

The sophisticated "code optimization" of PL/M-80 makes the final code extremely memory efficient. And extensive compiler options enable you to print clean source and object listings.

Naturally, PL/M-80 operates on the Intellec Microcomputer Development System. The PL/M-80 compiler fully supports ICE-80 and ICE-85 In-Circuit Emulation, including full use of symbolic debugging.



FORTRAN-80

FORTRAN is a familiar, widely accepted programming language. Intel brings FORTRAN to the 8080/8085 family in the form of FORTRAN-80, an ANSI 77 FORTRAN compiler.

FORTRAN-80 produces the relocatable and linkable object code that helps reduce programming time and speed 8080/8085 product development.

Since Intel's FORTRAN-80 meets the ANSI 77 subset standards, programs written in standard FORTRAN will compile under FORTRAN-80. Intel has added several extensions to the ANSI 77 standard, making the language even more useful and flexible for microcomputer designers.

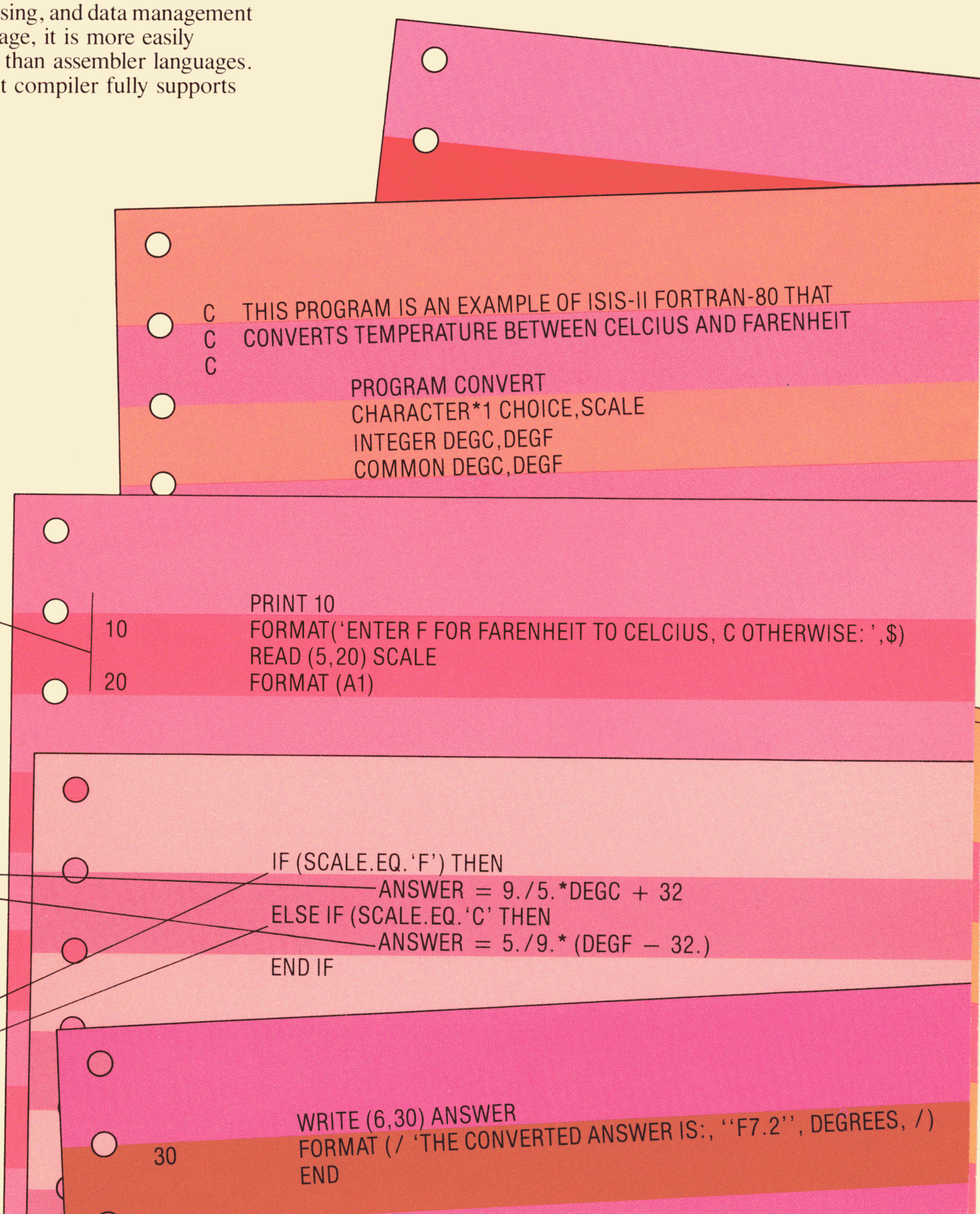
FORTRAN-80 is complementary to PL/M, adding strengths in arithmetic and mathematical functions, formatted I/O, numeric processing, and data management support. As a high level language, it is more easily learned, coded, and enhanced than assembler languages.

This Intellec system-resident compiler fully supports all ICE Emulation options.

ASM80.

Intel's ASM80 rounds out the 8-bit development package with a flexible assembler, offering macro capabilities with conditional assembly, plus relocatable and linkable object code.

Time and space-critical modules should be written in ASM80, which complements PL/M-80 and FORTRAN-80. The macro capabilities of ASM80 allow the programmer to code frequently used sequences once. The programmer then references the macro sequence using only the name given the sequence. At assembly time, the named macro is inserted in the code with appropriate arguments. This means reduced errors, easier maintenance, faster de-bug.



```
C   THIS PROGRAM IS AN EXAMPLE OF ISIS-II FORTRAN-80 THAT
C   CONVERTS TEMPERATURE BETWEEN CELCIUS AND FARENHEIT
C
          PROGRAM CONVERT
          CHARACTER*1 CHOICE,SCALE
          INTEGER DEGC,DEGF
          COMMON DEGC,DEGF

          PRINT 10
 10       FORMAT('ENTER F FOR FARENHEIT TO CELCIUS, C OTHERWISE: ',$)
          READ (5,20) SCALE
 20       FORMAT (A1)

          IF (SCALE.EQ.'F') THEN
                ANSWER = 9./5.*DEGC + 32
          ELSE IF (SCALE.EQ.'C' THEN
                ANSWER = 5./9.* (DEGF - 32.)
          END IF

          WRITE (6,30) ANSWER
 30       FORMAT (/ 'THE CONVERTED ANSWER IS:, ''F7.2'', DEGREES, /)
          END
```

FORTRAN-80's formatted I/O, for file access, report generation, and well-defined access formats.

A strength of FORTRAN is its numeric processing capability, making it ideal for engineering applications.

Structured IF, THEN, ELSE statements enhance code readability, reliability.

FSP Fundamental Support Package.

The FSP Fundamental Support Package of mathematics and technical libraries provides a series of unique capabilities. These routines enable the programmer to focus on the application problem, while Intel supplies proven numeric algorithms.

The Fundamental Support Library includes the following:

FSP Machine for fast, low-level string handling, binary and decimal integer arithmetic.

Binary Integer Arithmetic for arithmetic operations on signed and unsigned binary integers.

Floating-Point Arithmetic for arithmetic operations on binary floating-point numbers of various precisions.

Decimal Arithmetic for arithmetic operations and scaling operations on decimal integers.

String Handling for string manipulation, including search, comparison, scan, translation, table formatting.

Numeric Conversions and Numeric I/O to convert numeric data from one representation (i.e., binary) to another (i.e., decimal).

Transcendental Functions for trigonometric, exponential, and other transcendental functions.

Statistics for computation of mean, variance, standard deviation, correlation and other statistical operations.

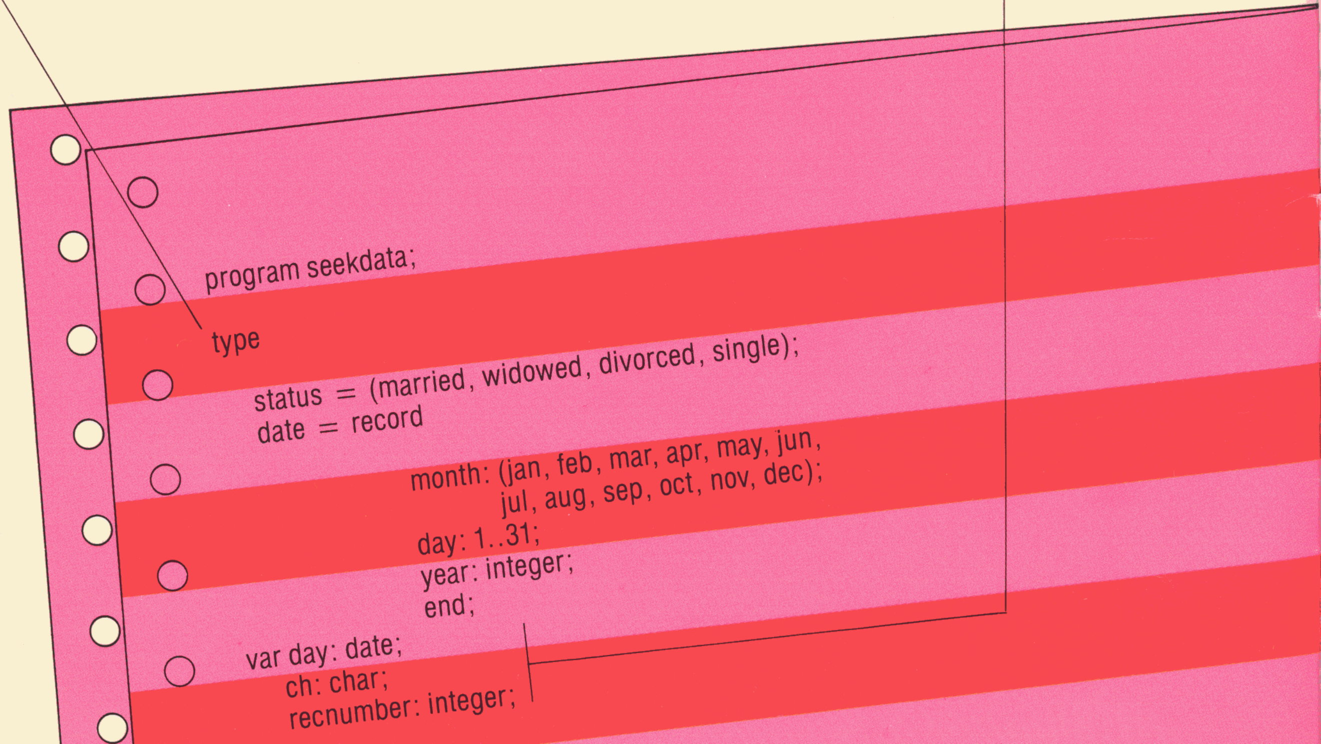
The P.I.D. process control routines produce output signals in response to input signals, using a formula with proportional, integral, and/or derivative terms, for real time control applications.

PASCAL-80 and BASIC-80 for iSBC™ Intel Single Board Computer Applications.

For applications employing the iSBC Single Board Computer family, you have two very attractive additional language choices, in addition to PL/M-80, FORTRAN-80, ASM-80, and the FSP Fundamental Support Package.

PASCAL provides for user-defined data types.

PASCAL variables can be defined as one of 10 system-defined data types. Here the user-defined type 'date' is used along with the system-defined 'integer' and 'character' types.



```
program seekdata;
type
   status = (married, widowed, divorced, single);
   date = record
            month: (jan, feb, mar, apr, may, jun,
                    jul, aug, sep, oct, nov, dec);
            day: 1..31;
            year: integer;
            end;
   var day: date;
       ch: char;
       recnumber: integer;
```

PASCAL-80. The Popular New Language.

PASCAL-80 is easy to use, and highly portable. It is especially strong because it is a block-structured language, with solid data-typing capabilities. A generalist's tool, PASCAL-80 is ideally suited for all non-hardware oriented applications.

Intel's PASCAL-80 contains all the features of standard PASCAL, enabling you to recompile existing PASCAL programs. Of course, as a high-level language, PASCAL-80 is much faster to code, test, and debug than assembler.

PASCAL-80 is both a compiler and an interpreter. Your PASCAL-80 source is compiled into an intermediate, or pseudo code, which is executed at run-time by the interpreter, or "run-time" package.

This run-time package is available to interface with the RMX Real-Time Multi-Tasking Executive operating system for the iSBC family. PASCAL-80 will also run on the Intellec system, giving you the advantage of development and testing in a familiar, known environment.

The pseudo code is optimized for speed and memory space. This allows the user to create more powerful and larger programs to run on the Intellec system or the target iSBC computer than would be possible with a straight compiler.

PASCAL has won a large and enthusiastic following. When you employ this valuable Intel implementation, you will see why.

BASIC-80

Easy to learn and implement, BASIC-80 is also a very comprehensive and useful tool. This interpreter meets the ANS 1978 standard, and adds many powerful extensions of its own.

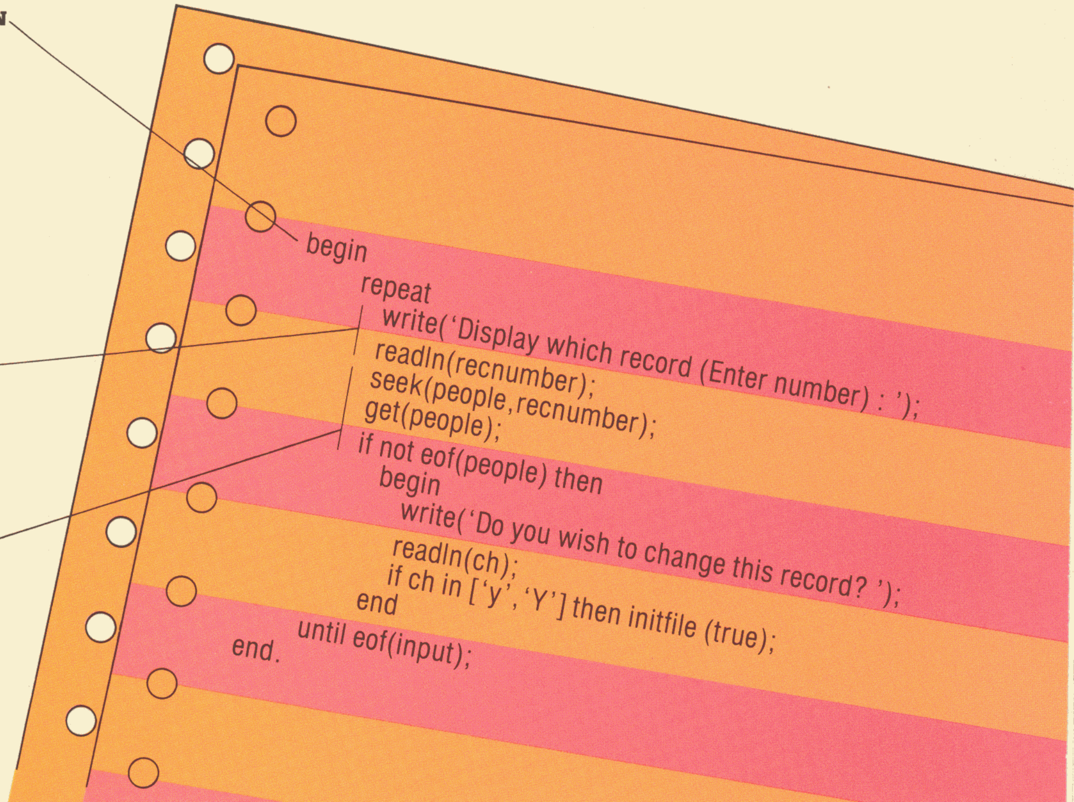
The BASIC-80 programs you create will be fully compatible with the iSBC (Intel Single Board Computer) run-time system, even under the RMX-80 real-time system, using the RMX Run Time Package.

BASIC-80 will also operate on the Intellec with ISIS-II, to provide a very useful extension of the capabilities of your development system.

Your urgent engineering or math problems can be programmed quickly for immediate results. Developed as an instruction language, BASIC is a natural, easy language for the beginning programmer.

BASIC-80 also allows an instant tool for prototyping program logic for testing and debugging. And it can be programmed to handle a multitude of business or reporting tasks, from simple files and lists to inventory files and other accounting chores. Because BASIC-80 is ANSI standard, existing BASIC programs may be easily ported over to your Intellec Microcomputer Development System.

BASIC-80 adds substantial features to the ANS 78 standard BASIC, including PEEK and POKE, plus IF, THEN, ELSE statements, single and double precision floating-point standard, and calls to user-supplied subroutines. These subroutines may have been written in PL/M-80, FORTRAN-80, or ASM80. Running on ISIS-II, or under the RMX operating system, the interactive interpreter provides full sequential and random disk file I/O.



A highly block-structured language, PASCAL-80's begin... end blocks and procedures isolate tasks and restrict the scope of variables and labels. The flow of control constructs such as REPEAT...UNTIL and IF...THEN allow top-down structured programming techniques.

Built-in routines for console input/output simplify programming.

PASCAL-80 also provides built-in routines for access to disk files.

```
begin
  repeat
    write(‘Display which record (Enter number) : ’);
    readln(recnumber);
    seek(people,recnumber);
    get(people);
    if not eof(people) then
      begin
        write(‘Do you wish to change this record? ’);
        readln(ch);
        if ch in [‘y’,‘Y’] then initfile (true);
      end
  until eof(input);
end.
```

ASM48 Assembler for the MCS®-48 Single-Chip Microcomputer Family.

Intel's Single-Chip family has found acceptance in a myriad of applications, including instrumentation, telecommunications, peripheral control, games, and appliance controls.

To put the 8048 Single-Chip family to work, Intel delivers a powerful macro assembler, ASM48.

Like other Intel assemblers, ASM48 is easy to use. The language supports symbolic references, code macros, and conditional assembly features. ASM 48 succeeds in the Intel tradition of enhancing programmer productivity.

Macro capabilities allow the programmer to code commonly used code sequences only once. Should corrections to the program be required, only the affected macro is recoded. Error correction is faster, since the correction need be made only once, rather than several times. Programming, debug, and maintenance are faster, and less expensive.

Conditional assembly allows the user to specify that only flagged portions of source code will actually be assembled into the "production" code. This allows the user to install special "debug" sequences, or to code for a variety of applications and specify at assembly time which sections should be assembled.

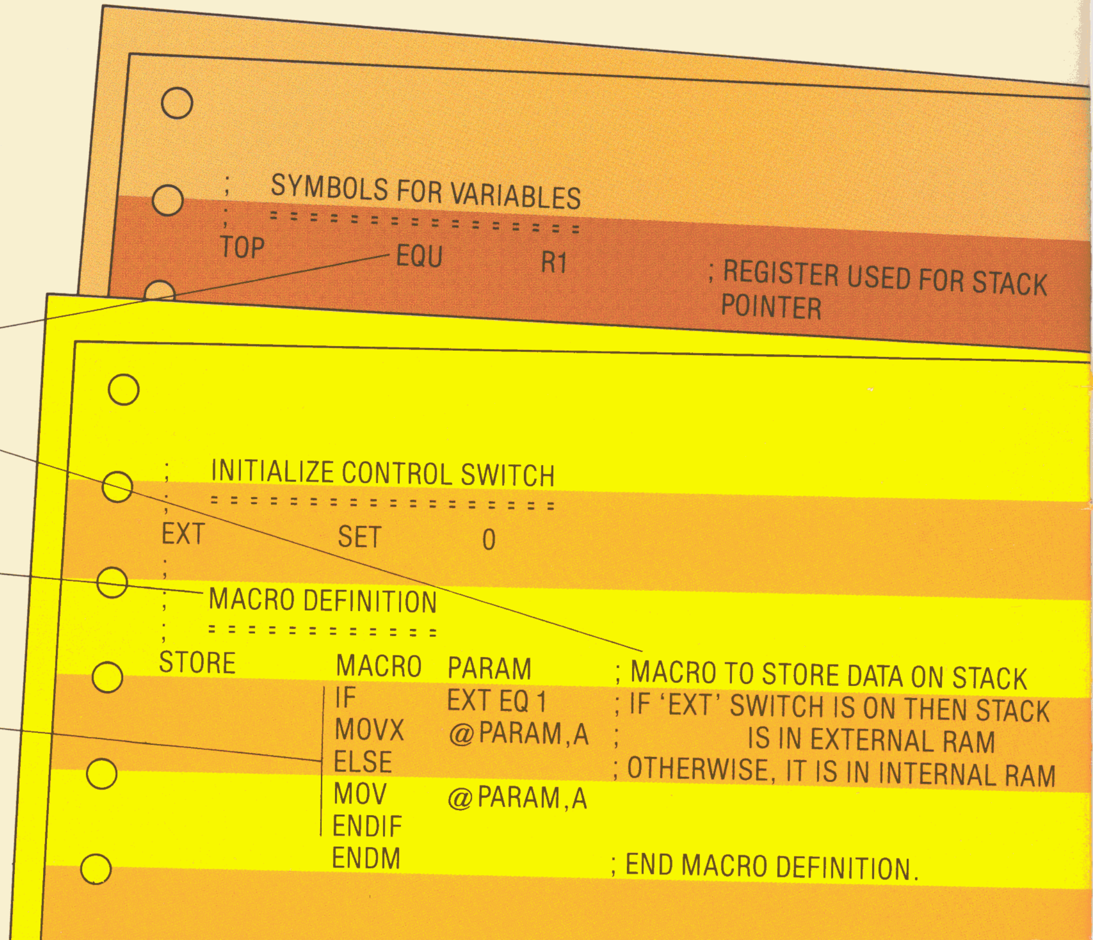
Combined with the MCS-48 single-chip family, the ASM48 assembler provides an effective tool for all sorts of applications. Naturally, it runs on the Intellec Microcomputer Development System, and is available now, for your applications.

ASM29 and SIM29 for the 2920 Analog Signal Processor.

Our new analog signal processor, the 2920, now receives Intellec Development System support. ASM29 is an assembler for symbolically programming analog applications. SIM29 is a simulator which monitors the 2920's execution.

And watch for more offerings as Intel delivers this new analog technology to the industry.

A macro which pushes data into a stack is defined so that if switch 'EXT' (external) is on, then the stack is built in external RAM. If the switch is off, then the stack is built in internal RAM.



```
;     SYMBOLS FOR VARIABLES
;     = = = = = = = = = = = = = = = = =
TOP                EQU          R1              ; REGISTER USED FOR STACK
                                                  POINTER
```

```
;     INITIALIZE CONTROL SWITCH
;     = = = = = = = = = = = = = = = = = = = =
EXT                SET          0
;
;     MACRO DEFINITION
;     = = = = = = = = = = = = =
STORE              MACRO    PARAM           ; MACRO TO STORE DATA ON STACK
                 | IF       EXT EQ 1        ; IF 'EXT' SWITCH IS ON THEN STACK
                 | MOVX     @PARAM,A        ;           IS IN EXTERNAL RAM
                 | ELSE                     ; OTHERWISE, IT IS IN INTERNAL RAM
                 | MOV      @PARAM,A
                 | ENDIF
                   ENDM                     ; END MACRO DEFINITION.
```

The EQUATE facility allows the user to define symbols.

Comments for program documentation.

ASM48's MACRO facility allows easy definition of macros with parameters. The code in this "STORE" macro will be inserted wherever "STORE" is referenced in the program.

An example of CONDITIONAL ASSEMBLY. Only one of the two MOVE instructions (MOVX or MOV) will be assembled, depending on the value of the "EXT" switch.

We Speak Your Language With Software Strength for the Intellec® System.

Intel's full complement of software for the Intellec Microcomputer Development System helps make the Intellec system more than the best development tool. Your Intellec system becomes a great all-around lab computer and business tool as well.

With Intel software packages running on the ISIS-II operating system, your Intellec system can be prototyping software with an interactive interpreter, or quickly solving complex engineering problems. Or your Intellec system can handle in-house management problem solving and reporting.

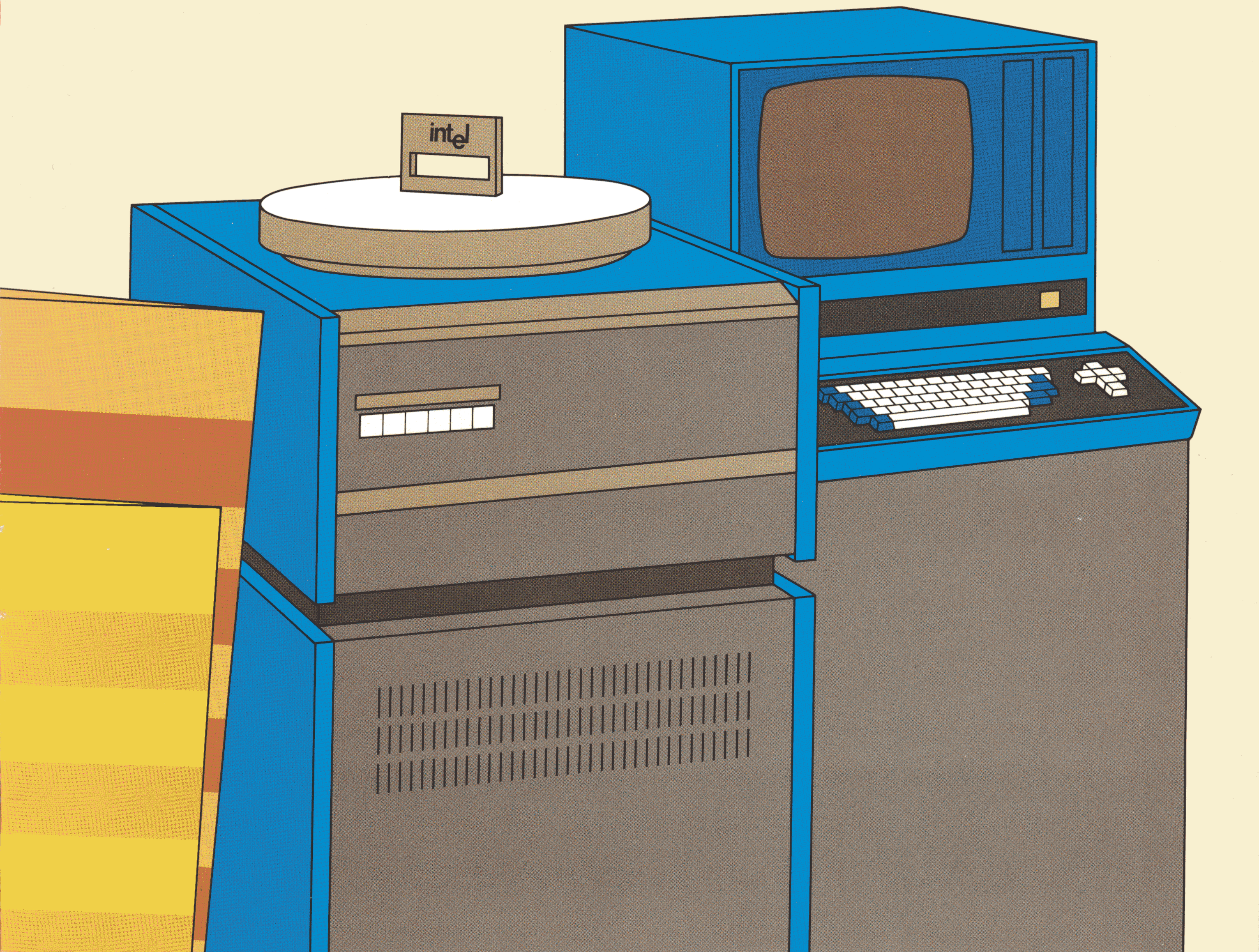
You can run all of the Intel 8-bit languages on the Intellec system — PL/M-80, FORTRAN-80, ASM80, the FSP Fundamental Support Package; plus the BASIC-80 interpreter and PASCAL-80 — two high-level languages highly useful in a lab environment.

The Intellec system's range of capabilities is extended even further with the addition of iCIS-COBOL, an ANSI Level I compiler-interpreter, and CREDIT, our powerful, screen oriented text editor.

All run on the friendly ISIS-II disk operating system, which now supports the new large storage hard disk subsystem for the Intellec.

iCIS-COBOL. Intel® Compact Interactive Standard COBOL.

iCIS-COBOL is ideal for business applications, because it employs business terminology stated in verbs and phrases. You may want to put iCIS-COBOL to work on such projects as inventories, mailing lists, materials



planning projects, or even payroll. There exists a vast number of business applications in standard COBOL which could be easily ported over to the Intellec Microcomputer Development System to run under this ANSI Level I standard COBOL.

iCIS-COBOL features both a compiler and an interpreter. The compiler translates COBOL source programs into an intermediate, or pseudo code. This pseudo code is optimized for speed and memory space. (A compile-time option flags non-ANSI standard features, to help you port other COBOL programs to the Intellec.)

The pseudo code is then executed by the run-time package. Completed code may be permanently linked to the run-time interpreter to form a free-standing, ISIS II loadable program. And you can call subroutines written in PL/M-80, FORTRAN-80, or ASM80.

Whether you decide to code your own applications, using the flexible FORMS and CRT formatting tools, or to port existing COBOL programs to your Intellec system, iCIS-COBOL is a business-like choice for your business applications.

ISIS-II Disk Operating System.

ISIS-II is a full-function, powerful disk-operating system for the Intellec Microcomputer Development System. ISIS-II supports up to five 8″ floppy disk drives in different Intellec system configurations, storing up to 2.5 megabytes.

And ISIS-II runs the new hard disk subsystem for the Intellec, which features 7.3 megabytes of disk storage, half on a fixed platter and half on a removable platter.

A flexible, friendly operating system, ISIS-II includes a Library Manager, Link and Locate capabilities. These allow the user to combine object code modules from separately compiled FORTRAN-80 and PL/M-80 programs and assembled ASM80 programs into a final run-time package.

Powerful file manipulation commands enable the user to copy, rename, erase, or combine disk files easily.

CREDIT.

CREDIT is Intel’s screen-oriented text editor for the Intellec system. CREDIT allows you to perform file entry and modifications on your Intellec Microcomputer Development System as if it were a word processing machine.

The package features two operating modes. The SCREEN mode makes full use of the CRT display, with movable cursor, typeover, character or line deletion and insertion. The text or code you key in appears on the CRT exactly as it is stored in memory. You can easily make corrections by positioning the cursor over the desired characters.

The second mode, the COMMAND mode, provides the string search and replace capabilities, formatting, and other editing commands. You can freely switch back and forth between modes.

CREDIT will increase your throughput at the keyboard dramatically. It is easy to key in source. Corrections are simple. And it is fun to use.

Software to Make the Intellec® System Work Harder.

The extra value of these Intellec system-based software products actually extends the power of your development computer, to almost any project or problem you might wish to solve.

We Speak Your Language. When You're Ready to Talk.

When you're ready to talk product development, let's talk about Intel extras. By extras we mean support. Training courses to keep you current with new hardware and software developments. The INSITE users' library of programs. And the support of Intel's trained Field Applications Engineers.

Intel Training Courses.

Throughout the year, Intel offers training courses in major cities and at Intel Training Centers around the world. These practical workshops in microcomputer technology include lectures, demonstrations, and laboratory work. They provide "hands-on" experience in implementing Intel microcomputers for system development. And "hands-on" experience on the Intellec Microcomputer Development System, too.

A continually updated comprehensive schedule of courses covers even the most current data at introductory, intermediate, and advanced levels. Classes last from two to five days, and are tailored to varied audiences — from design engineers and programmers to managers, project leaders, managers, administrative and even non-technical personnel.

Workshops are available at your facility.

INSITE™ Users' Program Library.

Through INSITE, Intel's Software Index and Technology exchange, Intel makes available a broad collection of programs, procedures and macros written for the 8080/8085, 8086/8088, and 8048 processor families. These general-purpose programs can substantially cut programming and debugging time for INSITE users.

INSITE offers hundreds of programs, games, and utilities, and is continually updated. The library is created by Intel customers, who exchange their own programs or routines (subject to acceptance) for selections from the INSITE library. Programs from the library are also available at a modest charge.

FAE Field Applications Engineer Suppport.

Intel provides Field Applications assistance at most field locations throughout the world. These applications engineers provide technical expertise to assist you in your development efforts with Intel components, Intel software, and your Intellec Microcomputer Development System.

intel®

U.S. AND CANADIAN SALES OFFICES

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080
TWX: 910-338-0026
TELEX: 34-6372

ALABAMA

Intel Corp.
3322 S. Parkway, Ste. 71
Holiday Office Center
Huntsville 35802
Tel: (205) 883-2430

Pen-Tech Associates, Inc.
Holiday Office Center
3322 Memorial Pkwy., S.W.
Huntsville 35801
Tel: (205) 881-9298

ARIZONA

Intel Corp.
10210 N. 25th Avenue, Suite 11
Phoenix 85021
Tel: (602) 997-9695

BFA
4426 North Saddle Bag Trail
Scottsdale 85251
Tel: (602) 994-5400

CALIFORNIA

Intel Corp.
7670 Opportunity Rd.
Suite 135
San Diego 92111
Tel: (714) 268-3563

Intel Corp.*
1651 East 4th Street
Suite 105
Santa Ana 92701
Tel: (714) 835-9642
TWX: 910-595-1114

Intel Corp.*
15335 Morrison
Suite 345
Sherman Oaks 91403
(213) 986-9510
TWX: 910-495-2045

Intel Corp.*
3375 Scott Blvd.
Santa Clara 95051
Tel: (408) 987-8086
TWX: 910-339-9279
TWX: 910-338-0255

Earle Associates, Inc.
4617 Ruffner Street
Suite 202
San Diego 92111
Tel: (714) 278-5441

Mac-I
2576 Shattuck Ave.
Suite 4B
Berkeley 94704
Tel: (415) 843-7625

Mac-I
P.O. Box 1420
Cupertino 95014
Tel: (408) 257-9880

Mac-I
11725 Espen Circle
P.O. Box 8763
Fountain Valley 92708
Tel: (714) 839-3341

Mac-I
110 Sutter Street
Suite 715
San Francisco 94104
Tel: (415) 982-3673

Mac-I
20121 Ventura Blvd., Suite 240E
Woodland Hills 91364
Tel: (213) 347-5900

COLORADO

Intel Corp.*
650 S. Cherry Street
Suite 720
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289

Westek Data Products, Inc.
25921 Fern Gulch
P.O. Box 1355
Evergreen 80439
Tel: (303) 674-5255

Westek Data Products, Inc.
1322 Arapahoe
Boulder 80302
Tel: (303) 449-2620

Westek Data Products, Inc.
1228 W. Hinsdale Dr.
Littleton 80120
Tel: (303) 797-0482

CONNECTICUT

Intel Corp.
Peacock Alley
1 Padanaram Road, Suite 146
Danbury 06810
Tel: (203) 792-8366
TWX: 710-456-1199

FLORIDA

Intel Corp.
1001 N.W. 62nd Street, Suite 406
Ft. Lauderdale 33309
Tel: (305) 771-0600
TWX: 510-956-9407

FLORIDA (cont.)

Intel Corp.
5151 Adanson Street, Suite 203
Orlando 32804
Tel: (305) 628-2393
TWX: 810-853-9219

Pen-Tech Associates, Inc.
201 S.E. 15th Terrace, Suite K
Deerfield Beach 33441
Tel: (305) 421-4989

Pen-Tech Associates, Inc.
111 So. Maitland Ave., Suite 202
P.O. Box 1475
Maitland 32751
Tel: (305) 645-3444

GEORGIA

Pen Tech Associates, Inc.
Cherokee Center, Suite 21
627 Cherokee Street
Marietta 30060
Tel: (404) 424-1931

ILLINOIS

Intel Corp.*
2250 Golf Road, Suite 815
Rolling Meadows 60008
Tel: (312) 981-7200
TWX: 910-651-5881

First Rep Company
9400-9420 W. Foster Avenue
Chicago 60656
Tel: (312) 992-0830
TWX: 910-227-4927

Technical Representatives
1502 North Lee Street
Bloomington 61701
Tel: (309) 829-8080

INDIANA

Intel Corp.
2212 Maplecrest Rd.
Ft. Wayne 46815
Tel: (219) 493-2571

IOWA

Technical Representatives, Inc.
St. Andrews Building
1930 St. Andrews Drive N.E.
Cedar Rapids 52405
Tel: (319) 393-5510

KANSAS

Intel Corp.
9393 W. 110th St., Ste. 265
Overland Park 66210
Tel: (913) 642-8080

Technical Representatives, Inc.
8245 Nieman Road, Suite #100
Lenexa 66214
Tel: (913) 888-0212, 3, & 4
TWX: 910-749-6412

KENTUCKY

Lowry & Associates Inc.
P.O. Box 1827
Lexington 40593
Tel: (606) 273-3771

MARYLAND

Intel Corp.*
7257 Parkway Drive
Hanover 21076
Tel: (301) 796-7500
TWX: 710-862-1944

Glen White Associates
57 W. Timonium Road, Suite 307
Timonium 21093
Tel: (301) 252-6360

Mesa Inc.
11900 Parklawn Drive
Rockville 20852
Tel: Wash. (301) 881-8430
Balto. (301) 792-0021

MASSACHUSETTS

Intel Corp.*
27 Industrial Ave.
Chelmsford 01824
Tel: (617) 667-8126
TWX: 710-343-6333

EMC Corp.
381 Elliot Street
Newton 02164
Tel: (617) 244-4740

MICHIGAN

Intel Corp.*
26500 Northwestern Hwy.
Suite 401
Southfield 48075
Tel: (313) 353-0920
TWX: 910-420-1212
TELEX: 2 31143

Lowry & Associates, Inc.
135 W. North Street
Suite 4
Brighton 48116
Tel: (313) 227-7067

Lowry & Associates, Inc.
3902 Costa NE
Grand Rapids 49505
Tel: (616) 363-9839

MINNESOTA

Intel Corp.
7401 Metro Blvd.
Suite 355
Edina 55435
Tel: (612) 835-6722
TWX: 910-576-2867

Dytek North
1821 University Ave.
Room 163N
St. Paul 55104
Tel: (612) 645-5816

MISSOURI

Technical Representatives, Inc.
320 Brookes Drive, Suite 104
Hazelwood 63042
Tel: (314) 731-5200
TWX: 910-762-0618

NEW JERSEY

Intel Corp.*
Raritan Plaza
2nd Floor
Raritan Center
Edison 08817
Tel: (201) 225-3000
TWX: 710-480-6238

NEW MEXICO

BFA Corporation
P.O. Box 1237
Las Cruces 88001
Tel: (505) 523-0601
TWX: 910-983-0543

BFA Corporation
3705 Westerfield, N.E.
Albuquerque 87111
Tel: (505) 292-1212
TWX: 910-989-1157

NEW YORK

Intel Corp.*
350 Vanderbilt Motor Pkwy.
Suite 402
Hauppauge 11787
Tel: (516) 231-3300
TWX: 510-227-6236

Intel Corp.
80 Washington St.
Poughkeepsie 12601
Tel: (914) 473-2303
TWX: 510-248-0060

Intel Corp.*
2255 Lyell Avenue
Lower Floor East Suite
Rochester 14606
Tel: (716) 254-6120
TWX: 510-253-7391

Measurement Technology, Inc.
159 Northern Boulevard
Great Neck 11021
Tel: (516) 482-3500

T-Squared
4054 Newcourt Avenue
Syracuse 13206
Tel: (315) 463-8592
TWX: 710-541-0554

T-Squared
2 E. Main
Victor 14564
Tel: (716) 924-9101
TWX: 510-254-8542

NORTH CAROLINA

Intel Corp.
154 Huffman Mill Rd.

Tel: (919) 584-3631

Pen-Tech Associates, Inc.
1202 Eastchester Dr.
Highpoint 27260
Tel: (919) 883-9125

Glen White Associates
4009 Barrett Dr.
Raleigh 27609
Tel: (919) 787-7016

OHIO

Intel Corp.*
6500 Poe Avenue
Dayton 45415
Tel: (513) 890-5350
TWX: 810-450-2528

Intel Corp.*
Chagrin-Brainard Bldg. #210
28001 Chagrin Blvd.
Cleveland 44122
Tel: (216) 464-2736

Lowry & Associates Inc.
1440 Snow Road
Suite 216
Cleveland 44134
Tel: (216) 398-0506

Lowry & Associates, Inc.
1524 Marsetta Drive
Dayton 45432
Tel: (513) 429-9040

Lowry & Associates, Inc.
2735 Cleveland Ave.
Columbus 43224
Tel: (614) 436-2051

OREGON

Intel Corp.
10700 S.W. Beaverton
Hillsdale Highway
Suite 324
Beaverton 97005
Tel: (503) 641-8086

PENNSYLVANIA

Intel Corp.*
275 Commerce Dr.
200 Office Center
Suite 300
Fort Washington 19034
Tel: (215) 542-9444
TWX: 510-661-2077

Lowry & Associates, Inc.
Seven Parkway Center
Suite 455
Pittsburgh 15520
Tel: (412) 922-5110

Q.E.D. Electronics
300 N. York Road
Hatboro 19040
Tel: (215) 674-9600

TEXAS

Intel Corp.*
2925 L.B.J. Freeway
Suite 175
Dallas 75234
Tel: (214) 241-9521
TWX: 910-860-5487

Intel Corp.*
6420 Richmond Ave.
Suite 280
Houston 77057
Tel: (713) 784-3400

Industrial Digital Systems Corp.
5925 Sovereign
Suite 101
Houston 77036
Tel: (713) 988-9421

Intel Corp.
313 E. Anderson Lane
Suite 314
Austin 78752
Tel: (512) 454-3628

VIRGINIA

Glen White Associates
Route 2, Box 193
Charlottesville 22901
Tel: (804) 295-7686

Glen White Associates
P.O. Box 10186
Lynchburg 24506
Tel: (804) 384-6920

Glen White Associates
Rt. #1, Box 322
Colonial Beach 22442
Tel: (804) 224-7764

WASHINGTON

Intel Corp.
Suite 114 Bldg. 3
1603 116th Ave. N.E.
Bellevue 98005
Tel: (206) 453-8086

WISCONSIN

Intel Corp.
150 S. Sunnyslope Rd.
Brookfield 53005
Tel: (414) 784-9060

First Rep Company
9401 W. Beloit Rd.
Suite 304
Milwaukee 53227
Tel: (414) 546-2033

CANADA

Intel Semiconductor Corp.*
Suite 233, Bell Mews
39 Highway 7, Bells Corners
Ottawa, Ontario K2H 8R2
Tel: (613) 829-9714
TELEX: 053-4115

Intel Semiconductor Corp.
50 Galaxy Blvd.
Unit 12
Rexdale, Ontario
M9W 4Y5
Tel:(416) 675-2105
TELEX: 06983574

Multilek, Inc.*
15 Grenfell Crescent
Ottawa, Ontario K2G 0G3
Tel: (613) 226-2365
TELEX: 053-4585

Multilek, Inc.
Toronto
Tel: (416) 245-4622

Multilek, Inc.
Montreal
Tel: (514) 481-1350

*Field application location

PRINTED IN U.S.A.
H212/1279/40K/MS/GR