

Hochsprachenentwicklung auf dem PC in Verbindung mit Echtzeitdebugging

D. Gröber

Instrumatic Electronic GmbH

Lochhammerschlag 5a

8032 Gräfelfing

Wie läuft Softwareentwicklung heute ab?

An die Entwurfs- und Definitionsphase schließt sich der Zeitpunkt der Quellcodeerstellung (Source) an. Erfolgt die Programmierung in einer Hochsprache, so sind deren Vielzahl nicht abzugrenzen. Es gibt Anwendungen, die es aus technischen Gründen erfordern, direkt in Assembler zu programmieren - also möglichst maschinen-nah. In solchen Fällen werden zur Übersetzung der in mnemotechnischen Kürzeln geschriebenen Sourcetexte in die Maschinensprache, Crossassembler eingesetzt.

Bei der Auswahl von Crossassemblern sollten Prozessorherstellere-kompatibilität, die Verfügbarkeit eines leistungsfähigen Linkers/Locators und die Lauffähigkeit des Crossassemblers auf einem weitverbreiteten Betriebssystem wichtige Entscheidungsmerkmale sein.

Ferner sind zu prüfen: Bedienungskomfort, Dokumentation und Geschwindigkeit. Durch den Einsatz von Crossassembler der Instrum-atic-Softwareworkstation stehen dem Anwender Möglichkeiten zur Verfügung, die er erst bei wesentlich grösseren Rechnern wieder vorfindet. Trotz allem wurde schwerpunktmässig aber auf eine einfache Bedienung geachtet.

Hervorragende Merkmale der Crossassembler sind zum Beispiel:

- relocatibel
- recursive Macros
- Symbole die bis zu 132 Charakter signifikant sind,
- verknüpfte INCLUDE - Files,
- ausführliche Fehlermeldungen auf dem Bildschirm
- die Anzahl der relocatiblen Segmente ist unbegrenzt
- 32 Bit Arithmetik,
- Conditional-Assembly,
- lokale Symbole auch im Objektformat und
- eine einheitliche Bedienung

und des Linkers:

- universell für alle Crossassembler und CPU's,
- bietet volle Unterstützung der 8086-Segmentierung
- Standard-Output-Formate:
wie INTEL, MOTOROLA, TEXAS, TEKTRONIX und RCA
- er durchsucht automatisch die Libraries
- Kommandofile-Option (Batch-Abarbeitung)
- fehlende Parameter werden erfragt
- erzeugen von einer Map- und einer Crossreferenz
- Bereichskontrolle für relocatible Module
- eine Help-Utility.

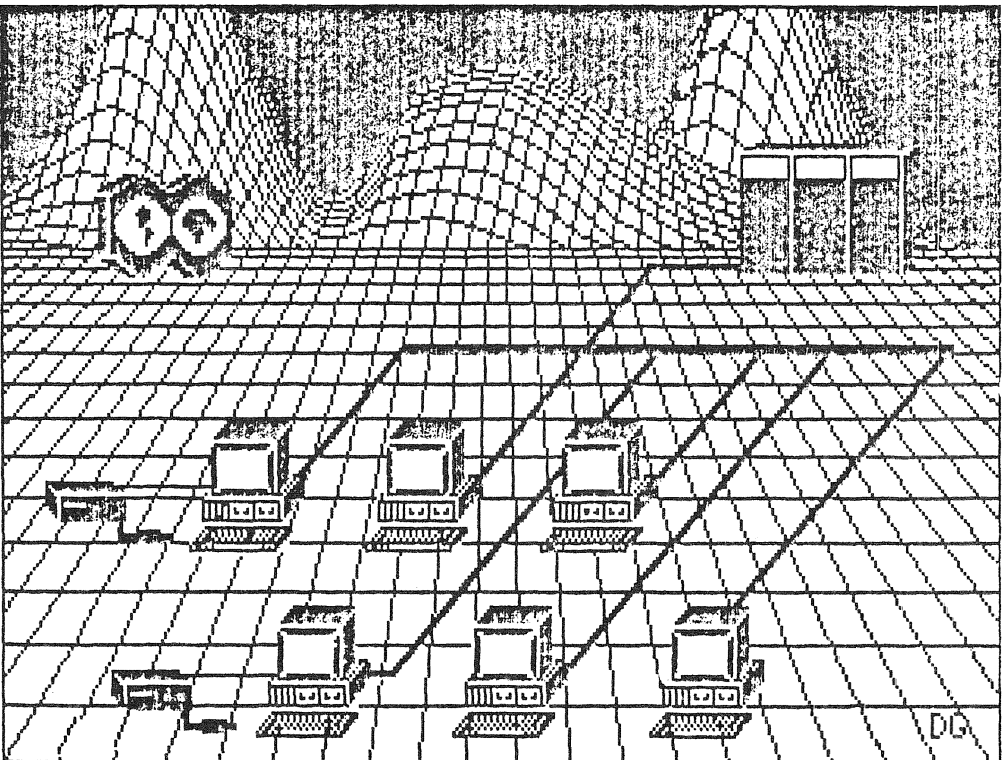
Der ebenfalls verfügbare Librarian erleichtert Ihnen wesentlich die Pflege Ihrer Programm-Bibliothek. Das Zusammenspiel mit den In-Circuit-Emulatoren dieser Software-Workstation ermöglicht die umfassende Symboltabelle: durch die Generierung von Symbolen,

Modulnamen, Segmenten usw. und sorgt so für ein effizientes Debugging.

Während diese Crossassembler einen großen Teil der Anforderungen an eine Softwareworkstation abdecken, wollen wir uns doch einem spezielleren Gebiet zuwenden. Schauen wir uns die Entwicklung von Software unter Hochsprachen an. Zum Beispiel an den Intel-Hochsprachen, da diese Prozessoren eine sehr große Verbreitung gefunden haben.

Auf der Softwareworkstation sind neben den zuvor erwähnten Crossassemblern auch **alle** Intel-Hochsprachen lauffähig! So laufen mit dem Paket **ACCESS-II** auf der INSTRUMATIC-Softwareworkstation 8 BIT-INTEL-Hochsprachen. Zusätzlich können mit diesem Programmpaket Files zwischen der INSTRUMATIC-Workstation und einer vorhandenen Intel-Entwicklungsstation in beide Richtungen transferiert werden.

Es ist somit möglich ein oder zwei Emulationsplätze aufzubauen und drei oder vier reine Programmierplätze vorzusehen. Jetzt kommen wir auf den Vorteil des Netzwerkes. Sie können nun über das Netzwerk schnell von der Emulationsstation aus auf die zentral angelegte Datenbank zugreifen, sich Files herunterladen und "unten" emulieren.



Die INSTRUMATIC-Softwareworkstation ist außerdem in der Lage für alle Intel **16-Bit** Prozessoren Software zu generieren. Wir werden sehen, wie das auf sehr komfortable Art und Weise möglich sein kann. Ähnlich wie auf der 8 Bit Seite, ist es selbstverständlich auf der 16 Bit Seite mit dem Paket **ACCESS-86** ebenso möglich, außer dem ASM-86 auch Hochsprachen wie PLM-86, Pascal-86, Fortran-86, C-86, Alter, Aedit, den Linker, Locator, OH-86 usw. laufenzulassen.

Transfer von BINAER- und ASCII-Files in beide Richtungen



Darüber hinaus kann bereits erstellte Software von Systemen der Serie-II, -III oder -IV, auf die INSTRUMATIC-Software-Workstation transferiert und dort selbständig weiterverarbeitet werden. Dies ist einleuchtend, denn wenn wir den Unterschied zwischen der Serie II und III anschauen, dann besteht er darin, daß die Serie II mit dem 8085 Prozessor durch Hinzustecken einer 8086-Prozessorkarte zu einer Serie III aufgerüstet wird. Diese Prozessorkarte besteht im wesentlichen aus einer 16 Bit-CPU, Memory, dem Businterface und Software. Diese Konfiguration entspricht im wesentlichen auch der Serie IV.

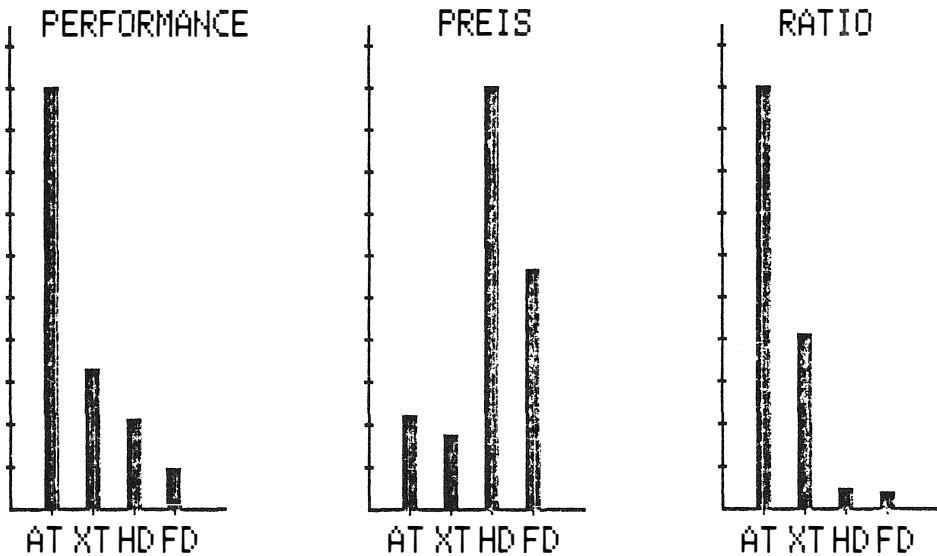
Sehen wir uns die INSTRUMATIC-Softwareworkstation etwas näher an:

Sie besteht aus dem zum Industriestandard gewordenen IBM-PC-DOS Computer mit einer 8088, 8086 oder 80286 CPU. Die große Stückzahl mit der dieses System gefertigt wird, erlaubt ein sehr gutes Preis-/Leistungs-Verhältnis. Preise, wie sie noch vor kurzem für spezielle Entwicklungssysteme mit geringerer Leistungsfähigkeit bezahlt wurden, gehören damit der Vergangenheit an. Bei vergleichbarer CPU-Leistung, erhöhtem Komfort und einschließlich einem standardisierten Betriebssystem liegen wir heute bei ca. 1.000 DM / MByte. Dies ist ein Verhältnis von bis zu 1:40 !



INSTRUMATIC

SOFTWARE - WORKSTATION



Mit dem Wachsen der Microprozessoren von 4 über 8 Bit in die 16 Bit und jetzt neuerdings in die 32 Bit Größe hinein, wachsen selbstverständlich auch die Softwareprogramme. Von damals 20K Programmen sind sie heute zu 100K und noch mehr angewachsen. Projekte, die damals von einem Ingenieur alleine realisiert werden konnten, werden heute von Teams oder ganzen Abteilungen übernommen. Das ist einer der vielen Gründe, der für eine Vernetzung der einzelnen Softwareworkstationen spricht. Das zeigt aber auch, daß Software-Engineering-Methoden heute anders aussehen als damals. Wobei "Damals" nur fünf oder höchstens 10 Jahre zurückliegt. So kam Anfang der 70iger-Jahre NASSI und SHNEIDERMAN zu einem wesentlichen Hilfsmittel, dem Struktogramm.

Struktogramme sind Hilfsmittel, die das engültige Programm in übersichtliche, deutlich zu trennende Bilder aufteilen. Aber bislang waren diese Struktogramme nur Bleistift und Papier orientiert. Das heisst, wenn etwas zu ändern war, dann mußten als Editor Schere und Radiergummi herhalten. Das wiederum hatte aber zur Folge, daß die Spezifikationen und dazugehörigen Dokumentationen einfach in der Schublade liegenblieben und man froh war, wenn man das Programm richtig zum Laufen brachte, da das nächste Projekt bereits wieder anstand. Diese Probleme des hohen Aufwandes bei der manuellen bleistiftmäßigen Erstellung von Struktogrammen ist mit dem Einsatz des Struktogrammeditors nicht mehr vorhanden.

Auf der INSTRUMATIC-Softwareworkstation wird mit wenigen Tastendrücken das Struktogramm aufgebaut und verändert. "Top-Down geführt" wird nun die Software im Struktogramm einschließlich der Kommentare entwickelt. Dann werden die Kommentare durch Statements, die an die gewünschte Zielsprache angelehnt sind, ergänzt. Die Generierung des Sourcetextes oder des Quelltextes erfolgt nun automatisch aus dem Struktogramm heraus. Der Sourcetexter wandelt die Struktur und die eingetragenen Statements in Quelltexte der dazugehörigen Hochsprache um.

Jeder, der schon Programme geschrieben hat, weiß, daß die ersten Compilerläufe meistens wegen Syntaxfehlern einfach abgebrochen werden. Um dem ständigen Wechselspiel zwischen Editieren, Compilieren, Editieren, usw. nun ein Ende zu machen, sind für die 8086-Hochsprachen Precompiler erhältlich. Diese zeigen Syntax- und Semantikfehler durch die direkte Cursorpositionierung an, erläutern mit einem Kurztext den Fehler und erlauben eine direkte Verbesserung im Struktogramm.

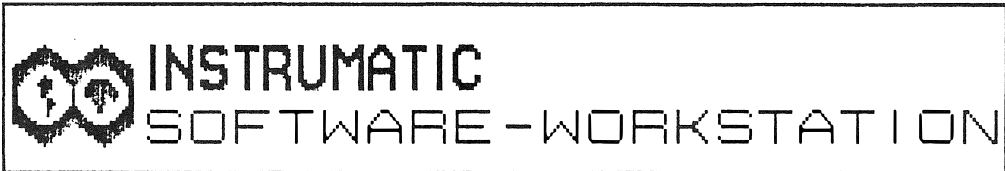
Liegen bereits umfangreiche Softwarepakete vor, die nachdokumentiert, gewartet oder geändert werden müssen, so läuft hierfür auf der INSTRUMATIC-Softwareworkstation auch ein Strukturtransformator. Dieser Strukturtransformator erzeugt aus existierenden Quelltexten Struktogrammdateien. Ist nur ein Dokumentieren erforderlich so wird dies ganz einfach durch einen Ausdruck geschehen. Im anderen Fall ist es jedoch auch möglich, nachträglich die gewünschten Änderungen im Struktogramm durchzuführen.

Oft ist es jedoch nötig, die geschriebene Software mit einem Debugger auszutesten. Das eben betrachtete Beispiel bezog sich auf die Syntax und Semantik. Das nächste Beispiel, das wir betrachten werden, ist die Programmierlogik. Hierzu steht auf der INSTRUMATIC-Softwareworkstation ein einfach zu bedienender Softwaredebugger für die 8086-Familie zur Verfügung.

Genescope, so nennt sich dieser Programmteil, setzt einen neuen Standard in Sachen Debuggingtool. Bisher bekannte Debuggingprogramme arbeiten im "Teletype"-Mode, d.h. nach einer Eingabe vom Benutzer führen sie sie aus und listen das Ergebnis wie ein Drucker auf dem Bildschirm. Im Gegensatz dazu arbeitet Genescope mit einem vollen Displaymodus. Über die Funktionstasten können mehrere Displays gesteuert werden: So ist z.B.

- ein Tracedisplay vorhanden,
- ein Display zum Breakpoints setzen,
- ein anderes Display erlaubt das Ansehen eines Files (z.B. MAP-Files) und selbstverständlich auch
- noch ein Helpdisplay.

Es können im Genescope Singlestepping, Breakpointsetzen, Execution starten, Disassemblieren des Codes, Ansehen und Modifizieren von Registern und Variablen und vieles mehr ausgeführt werden. Für die reine Softwareentwicklung mögen diese Werkzeuge genügen. Ist es jedoch notwendig die Software an Ort und Stelle auch in die Hardware zu integrieren, so kann die INSTRUMATIC-Softwareworkstation mit einem In-Circuit-Emulator für folgende Prozessoren erweitert werden:



Z80	8080 8085	80515
	NSC-800	
Z80001 Z80002	80080 80086 80088 80089 80090 80091 80092 80093 80094 80095 80096 80097 80098 80099	8031/51
68000 68008 68010	68000 68002 68004 68006 68008 68010 68012 68014 68016 68018 68020 68022 68024 68026 68028 68030 68032 68034 68036 68038 68040 68042 68044 68046 68048 68050 68052 68054 68056 68058 68060 68062 68064 68066 68068 68070 68072 68074 68076 68078 68080 68082 68084 68086 68088 68090 68092 68094 68096 68098 68099	68000 68002 68004 68006 68008 68010 68012 68014 68016 68018 68020 68022 68024 68026 68028 68030 68032 68034 68036 68038 68040 68042 68044 68046 68048 68050 68052 68054 68056 68058 68060 68062 68064 68066 68068 68070 68072 68074 68076 68078 68080 68082 68084 68086 68088 68090 68092 68094 68096 68098 68099
8032/52	68009 6809E	8048- FAMILIE

Was wird von einem Emulator erwartet? Der Emulator darf keine Speicherplätze im System und auch keine I/O-Ports belegen. Er sollte sich völlig transparent verhalten, d.h. er sollte den Prozessor ersetzen ohne eine Auswirkung auf das System zu haben. Der Emulator darf keine zusätzlichen Wait-Zyklen in das System einfügen. Wenn er keine Wait-States erzeugen darf, muß er sehr schnell sein, denn das Gerät muß z.B. eine Adresse als den Break-point erkennen, die Emulation anhalten und dabei alle aktuellen Registerinhalte abspeichern.

Der Emulator muß das Austesten eines Programmes einmal in Echtzeit und das andere mal in Einzelschritten erlauben. Der Emulator muß es darüberhinaus dem Anwender ermöglichen, in seinem Zielsystem ohne jede Einschränkung den Inhalt sämtlicher Speicherplätze, I/O-Ports und Register zu überwachen oder zu verändern.

Die **Emulatoren**, die in die INSTRUMATIC-Softwareworkstation integriert werden, können jedoch wesentlich mehr. So hat der 8086 Emulator z.B. einen 72 Bit breiten und 2K tiefen Tracespeicher. Diese 72 Bit setzen sich aus 24 Bit Adressen, 16 Bit Daten, 16 Bit Status- und Kontrolleitungen und 16 Bit externen Logikeingängen zusammen. Das Overlay-Memory läßt sich von 0 - 512KB ausbauen und in 2K-Schritten über den gesamten Adressbereich mappen.

Dieser Satelittenemulator ist ein Einschubsystem. Aus diesem Grund ist es möglich ihn für verschiedene Prozessoren, wie 8086-, 68000- und Z8000-Familien auszurüsten. Bei der **8086-Familie** werden der 8086 und 8088, beide in Min- und Max-Mode, der 80186 und der 80188 mit **einem** Controllerboard unterstützt. Beim Wechsel von der 8086 in die 80186-Familie ist lediglich ein Austausch des Pods nötig. Auf der 68000 Seite wird der **68000**, **68008**, **68010** und auf der Z8000 Seite der **Z8001** und der **Z8002** unterstützt. Für die Anwender der 68000-Familie wird auch die Emulation mit einem PIN-GRID-GEHÄUSE unterstützt.

Das Besondere an diesen Emulatoren ist das **Triggersystem**. Es gliedert sich in **4 Gruppen**, die frei definiert werden können, auf. Jede einzelne Gruppe hat wiederum mehrere Felder. So existieren in jeder Gruppe

- zwei 24 Bit breite Felder für die Adressen, die
- auch als Bereiche definiert werden können,
- zwei separate 16 Bit Felder für die Daten, wobei
- zwischen Adressen und Daten eine AND, NOT, OR -
- Verknüpfung möglich ist;
- zwei 16 Bit Felder für Status und Controlleitungen,
- ein 16 Bit breiter PASS-Counter.

In jeder Gruppe kann nun die Erkennung eines Ereignisses eine oder mehrere aus sieben verschiedenen Aktionen auslösen:

1. Trace ein- und ausschalten
2. Emulation anhalten (Break)
3. Break and Jump (Sprung in eine Unterroutine!)
4. Zähler inkrementieren
5. Zähler zurücksetzen
6. Oszillographen-Trigger-Output freigeben
7. nächste Eventgruppe freigeben.

So ist es z.B. mit dem Emulator möglich, die Abarbeitung eines Programmes anzuhalten, wenn versucht wird, in einen READ-Bereich hineinzuschreiben. Die folgenden vier Seiten zeigen nur einen kleinen Einblick in das sehr komfortable Triggersystem. Die Daten können selbstverständlich Binär, Oktal, Hexadezimal, Dezimal oder in ASCII formatiert dargestellt und auch eingegeben werden. Mit der Disassemblierungs-Firmware werden die Instruktionen in die Prozessor-Mnemonic übersetzt und zusammen mit den Symbolen auf dem Bildschirm dargestellt. Sowohl das Dissassembly als auch die Tracefunktion stellt dem Anwender die symbolischen Referenzen zur Verfügung.



INSTRUMATIC

SOFTWARE-WORKSTATION

IN JEDEM BREAKSYSTEM:

26 KOMBINATIONEN:

WHEN AC1 THEN....

WHEN AC2 AND DC1 THEN....

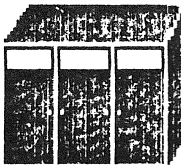
WHEN (AC1 OR AC2) AND (S1 OR S2) THEN....

SEPARATE ZUWEISUNG DANACH:

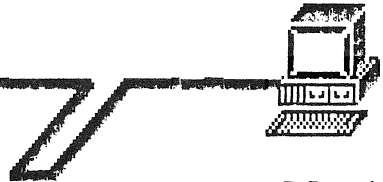
(AUCH WÄHREND DEM DEBUGGEN)

AC1=0FFF, AC2=%01110110 + #19 * &237, ..

Die Satelittenemulatoren haben noch einen darüber hinausgehenden **Vorteil:** Durch ihre zwei seriellen Schnittstellen können sie außer an der INSTRUMATIC-Softwareworkstation auch direkt an jedem Hostrechner oder Entwicklungssystem betrieben werden. Wird so z.B. an einer VAX mit Crosscompilern Software entwickelt, so kann in die serielle Leitung zwischen Host und Terminal (VT100) der Emulator eingeschleift werden.



PDP 11/xx
VAX 11/7xx
VME 10
INTEL



PC od.
UT100



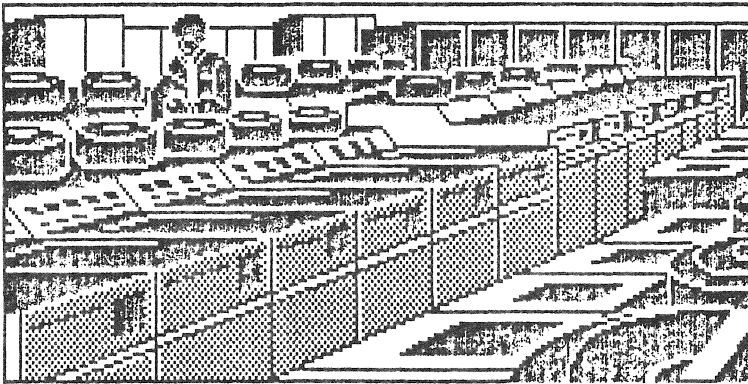
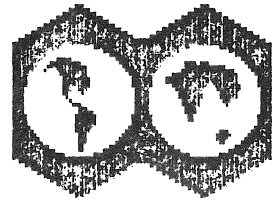
68000	8088	
68008	8086	
68010	80188	Z8001
	80186	Z8002

Im Einschaltmoment ist der Emulator transparent. Vom Terminal aus kann auf den Hostrechner zugegriffen werden, um die zu emulierenden Softwarepakete in den Emulator hinunterzuladen. Ist dies geschehen, wird er in den stationären Mode mit einer Escapesequenz umgeschaltet, und nun erfolgt die Kommunikation zwischen Terminal und Emulator, der Host wird jetzt nicht mehr belastet.

Dadurch, daß der Emulator über einen **Inlineassembler** verfügt, können nun kleine Veränderungen direkt unten vorgenommen und die Auswirkung sofort getestet werden. Eine dritte Möglichkeit erlaubt es, lediglich die Bedienungskommandos zwischen dem Emulator und dem Terminal auszutauschen, während der gesamte Datenverkehr zwischen dem Targetsystem und dem Hostrechner über den Emulator abläuft.

Für die INSTRUMATIC-Softwareworkstation wurde ein spezielles **Treiberprogramm** entwickelt, das die Bedienung des Emulators mit symbolischem Echtzeitdebugging vereint: ES-TERM. Mit ihm können mehrere Einzelkommandos zu Macros zusammengefaßt werden. Es können sich die Macros auch gegenseitig aufrufen (Nesting), auf Diskette abgespeichert werden sowie bedingte Weiterführung und Sprünge innerhalb eines Macros ausgeführt werden. Darüberhinaus besteht die Möglichkeit, sich den Bildschirm frei nach eigener Wahl aufzubauen. Auch unter dieser Software werden Symbole, Modulnamen und Zeilennummern im Disassembly und im Trace dargestellt. Variablen können als Byte, Wort, Longword, Segmentpointer, 32 Bit- und 64 Bit Floatingpoint dargestellt werden.

INTEGRATION DER SOFTWARE-WORKSTATION IN HOSTSYSTEME.



Firmen, die umfangreiche Softwareentwicklung vornehmen, setzen oft größere Hostrechner ein. Die INSTRUMATIC-Softwareworkstation läßt sich in diese Netzwerke ohne Probleme integrieren. So bietet die Firma INSTRUMATIC nicht nur die Integration einer solchen Softwareworkstation an, sondern auch die Programmpakete für die Kommunikation mit Hostsystemen, sowie Softwareentwicklungswerkzeuge für die Hostrechner selbst.

Zusammenfassend kann man noch sagen, daß die INSTRUMATIC-Softwareworkstation nicht nur vom Preis eine "tragbare" Lösung ist, kann sie doch auch auf tragbaren Rechner wie den IBM-PPC und dem Compaq-Portable mit einer speziellen Harddisk, installiert werden.

