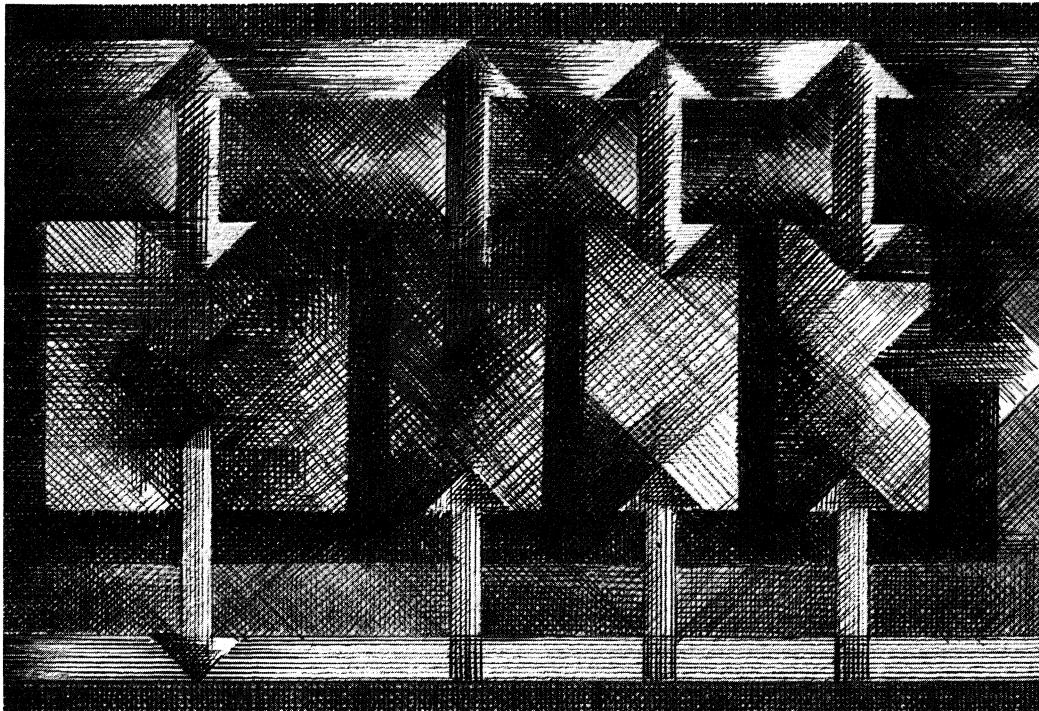




Valvo-Mikroprozessoren

Entwicklungssystem TWIN



VALVO-MIKROPROZESSOREN



TWIN

MIKROCOMPUTER-
ENTWICKLUNGSSYSTEM



Valvo-Mikroprozessoren

TWIN

**Mikrocomputer-Entwicklungssystem
— eine Einführung —**

bearbeitet von J. Wüstehube

Ausgabe März 1977

**Herausgeber: VALVO Unternehmensbereich Bauelemente der Philips GmbH,
2000 Hamburg 1, Burchardstraße 19**

**Nachdruck, auch auszugsweise, ist nur gestattet, wenn im Einzelfall die
Genehmigung des Herausgebers vorliegt.**

Der Herausgeber übernimmt keine Gewähr, daß die in dieser Schrift angegebenen Schaltungen, Geräte, Maschinen, Anlagen, Bauelemente, Baugruppen oder Verfahren frei von Schutzrechten sind. Die hier gegebenen Ratschläge sind unverbindliche und keine Haftung begründende Empfehlungen.

Druck: Hanseatische Druckanstalt GmbH, 2000 Hamburg 70

Einbandentwurf: Bruno Büsche

Vorwort

Mit der Einführung des Mikroprozessors hat eine völlig neue Entwicklungsphase in der Elektronik eingesetzt. Sie ist vor allem gekennzeichnet durch eine fundamentale Änderung des Arbeitsprinzips elektronischer Geräte. Während bisher jede digitale Funktion durch eine festverdrahtete Schaltung aus SSI- und MSI-Bausteinen realisiert wurde, führt ein mit Mikroprozessoren aufgebautes System, häufig Mikrocomputer genannt, diese Funktionen aufgrund von Befehlen aus, die in Form eines Programms festgelegt sind. Diese für viele Gerätehersteller neuartige Technik erfordert bei der Projektierung, Entwicklung, Produktion, Prüfung und Instandhaltung elektronischer Geräte natürlich auch neue Methoden und entsprechend hoch entwickelte technische Hilfsmittel zusätzlich zum Digitalvoltmeter, Pulsgenerator und Oszilloskop. Nur so kann man die komplexen Aufgaben bei der Entwicklung von Mikroprozessorsystemen zügig und rationell bewältigen. Die in dieser Beziehung wirksamste und eleganteste Methode besteht in der Benutzung eines laboreigenen, sogenannten Mikrocomputer-Entwicklungssystems.

Ein solches System muß vor allem folgende Grundanforderungen erfüllen:

- Universalität, d. h. keine Beschränkung auf einen bestimmten Prozessortyp oder eine Prozessorfamilie, damit Investitionen (Personal-Ausbildung, Software, Geräteausstattung) optimal genutzt werden können und keine einschneidenden Kompromisse in bezug auf technische Eigenschaften oder Kosten nötig werden.*
- Einfache Bedienbarkeit.*
- Test- und Fehlersuchmöglichkeiten für Soft- und Hardware sowie für das hieraus integrierte Anwendersystem.*
- Einfache Anschlußmöglichkeit an das zu prüfende Anwendersystem, ohne daß dessen Funktion wesentlich beeinflußt wird.*
- Massenspeicher mit einem auf diese Anwendungen zugeschnittenen Betriebssystem für eine effiziente Entwicklung der Anwender-Software.*

Diese Gesichtspunkte waren maßgebend für die Konzeption des Mikrocomputer-Entwicklungssystems TWIN. Das System läßt mehrere Ausbaustufen in der Hardware und der entsprechenden Soft-

ware zu. Es unterstützt den Gerätehersteller bei der Entwicklung seiner Mikroprozessorsystem-Hardware und -Software, angefangen von der ersten Programmiederschrift bis zur Abnahmeprüfung des fertigen Prototyps. Es kann darüber hinaus bei entsprechender Programmierung auch zur Mikroprozessor-Eingangskontrolle und als Prüfgerät in der Fertigung eingesetzt werden.

Die vorliegende Broschüre gibt eine Einführung in das TWIN-System und soll die vielfältigen Möglichkeiten erläutern, die TWIN aufgrund seines modularen Aufbaus und eines ausgewogenen Hardware-Software-Zusammenspiels bietet. Um einen ersten Überblick zu gewinnen, braucht man zunächst nur die Abschnitte 1 und 4 zu lesen. Die Abschnitte 2 und 3 vermitteln einen genaueren Eindruck von den Hardware- und Software-Eigenschaften des TWIN-Systems und dessen Arbeitsweise, während die vielseitigen Einsatzmöglichkeiten des TWIN-Systems aus der Kommandoliste im Anhang erkennbar werden. Ausführliche Antwort auf alle Detailfragen geben die Arbeitsanleitungen, die zusammen mit dem TWIN-System geliefert werden.

V A L V O
Unternehmensbereich Bauelemente
der Philips GmbH

INHALT

Geräte des TWIN-Systems	8
1. Überblick	9
1.1. Aufgaben des TWIN-Systems	9
1.2. Aufbau des TWIN-Systems	11
2. TWIN-Hardware	13
2.1. Verkehr des Rechners mit TWIN- und Anwender-Peripherie	13
2.2. Der TWIN-Rechner	15
2.2.1. Master-Slave-Prinzip	15
2.2.2. Getrennte Speicher	17
2.2.3. Master-Slave-Wechselwirkungen	18
2.2.4. Ein-/Ausgabe- und Unterbrechungs-Möglichkeiten	18
2.2.5. Steuer-/Debug-Platine	19
2.2.6. PROM-Programmier-Platinen	20
2.2.7. TWICE-Hardware	21
2.2.8. Kurze Gerätebeschreibung	21
2.3. Zweifach-Diskettenspeicher	24
2.4. Periphere Geräte	26
3. TWIN-Software	27
3.1. Betriebssystem SDOS	27
3.1.1. Aufgaben von SDOS	27
3.1.2. Dateien, Ein-/Ausgabegeräte, Kanäle	27
3.1.3. Kommando-Dateien	28
3.2. Editor	29
3.3. Assemblierer	29
3.4. Debug-Programm	30
3.5. PROM-Programmierer	31
4. Das Arbeiten mit TWIN	31
Anhang	42
Wichtige Benutzer-Kommandos beim TWIN-System	42
A.1. SDOS-Kommandos	42
A.2. Editor-Kommandos	45

Geräte und Baugruppen des TWIN-Systems und zugehörige Software

Basis-TWIN

Rechner
Zweifach-Diskettenspeicher
TWICE-Kabel
Disketten-Betriebssystem SDOS
Editor
2650-Assemblierer
Debug-Programm
PROM-Programmier-Software

Super-TWIN

Rechner mit
PROM-Programmier-Platine für 82 S 115
PROM-Programmier-Platine für 1702 A
Standard-Ein-/Ausgabe-Platine
Zweifach-Diskettenspeicher
TWICE-Kabel
Datensichtgerät mit Tastatur
Zeilendrucker
Disketten-Betriebssystem SDOS
Editor
2650-Assemblierer
Debug-Programm
PROM-Programmier-Software

Ferner stehen zur Verfügung

Platinen mit je 4 k Bytes statischem Schreib-/Lesespeicher
PROM-Programmier-Platinen für 82 S 115 und 1702 A
Frontplatte mit erweitertem Bedienfeld
Erweiterungs-Platinen
Disketten (formatiert)
Diagnostik-Software

1. Überblick

1.1. Aufgaben des TWIN-Systems

Das System TWIN (Test Ware INstrument) dient der Entwicklung von Mikroprozessorsystemen. Es unterstützt den Benutzer bei der Bewältigung von Entwurfs-, Simulations- und Testproblemen. Es kann sowohl in der Entwicklung als auch in der Fertigung eingesetzt werden. TWIN zeichnet sich dadurch aus, daß man es nicht nur zur Unterstützung der Software-Entwicklung benutzen kann, sondern daß sich auch die zu fertigende Hardware des geplanten Mikroprozessorsystems schrittweise in die Testphasen der Entwicklung mit einbeziehen läßt. Folgende Aufgaben lassen sich mit Hilfe von TWIN zügig und rationell bewältigen:

- Schreiben und Speichern von Anwenderprogrammen,
- Assemblieren dieser Programme,
- Testen von Programmen und Beseitigen von Fehlern in Programmen,
- Testen des Zusammenspiels von Programm und Prototyp in mehreren Stufen zur Fehlererkennung in Software und Hardware,
- Programmieren von Festwertspeichern (hardwaremäßige Fixierung des Programms).

Das TWIN-System kann also in jeder Phase der Entwicklung eines Mikroprozessorsystems eingesetzt werden. Daraüber hinaus läßt es sich zur Funktionsprüfung bei der Produktion und zur Fehlersuche verwenden, um Schwierigkeiten in der Fertigung schnell und damit kostensparend zu beheben. TWIN kann bei entsprechender Programmierung auch zur Funktionsprüfung von Mikroprozessoren selbst, zum Beispiel in der Eingangskontrolle, ausgebaut werden.

Einzelheiten zu den genannten Einsatzbereichen sowie die Möglichkeiten, die das TWIN-System insbesondere auch bei allen Entwicklungsschritten im Zusammenhang mit der Hardware bietet, sollen später ausführlicher erläutert werden. An dieser Stelle sei jedoch schon auf die besondere Architektur des TWIN-Rechners hingewiesen. Sie besteht darin, daß zwei Prozessoren mit den zugehörigen Speichern vorhanden sind, die als Master-Prozessor und Slave-Prozessor bezeichnet werden. Der Name TWIN (Zwilling) soll auf diese beiden Prozessoren hinweisen. Diese bei Mikrocomputer-Entwicklungssystemen erstmalig angewandte Rechner-Architektur bietet u. a. zwei wesentliche Vorteile, die hier – zunächst ohne nähere Erläuterung – genannt sein sollen.

Erstens ist dadurch sichergestellt, daß das TWIN-Betriebssystem nicht durch ungetestete Anwenderprogramme zerstört werden kann; man sagt, das Betriebssystem ist gegen Zusammenbruch geschützt oder „crash-proof“. Zum zweiten ermöglicht die Zwillings-Prozessor-Architektur jederzeit ein Umrüsten auf andere Mikroprozessortypen, indem man weitere Slave-Prozessoren hinzufügt und entsprechende Programme verwendet. TWIN läßt sich stets dem neuesten Stand der Mikroprozessortechnik anpassen, das System ist zukunftsicher („obsolete proof“). Das TWIN-System wird heute für den 8-bit-Mikroprozessor 2650 geliefert. Erweiterungen für andere Mikroprozessoren werden folgen.

In Bild 1 sind noch einmal die Möglichkeiten zusammengestellt, die TWIN im Vergleich zu anderen Entwicklungshilfen bietet, und Bild 2 zeigt die vielfältigen Vorteile, die sich daraus für den Gerätehersteller ergeben.

	Software-entwicklung	System-Simulation	PROM-Programmierung	Emulation im Prototyp	crash-*) proof	unabhängig *) vom Prozessortyp
TWIN	x	x	x	x	x	x
Platten-Systeme	x	x	x	zusätzlich		
Lochstreifen-Systeme	x	x	zusätzlich	zusätzlich		
Cross - Software	x	x				

VE440066

*) aufgrund der Zwillings-Prozessor-Architektur

Bild 1. Vergleich von Entwicklungshilfen für Mikroprozessorsysteme

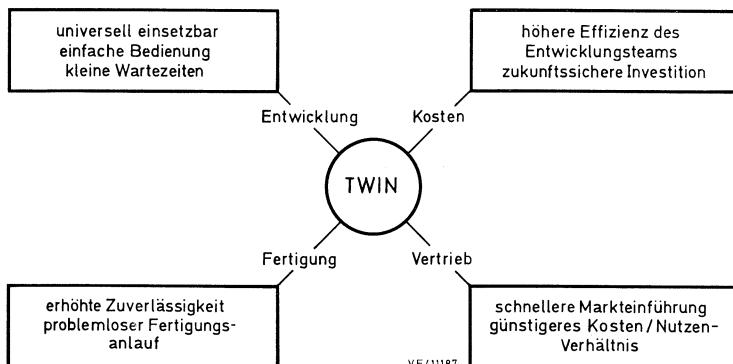


Bild 2. TWIN-Vorteile für den Gerätehersteller

1.2. Aufbau des TWIN-Systems

Zum TWIN-System gehören Hardware und Software. In Bild 3 ist die TWIN-Hardware mit ihren wichtigsten Geräten zu sehen:

- TWIN-Rechner mit PROM-Programmier-Einrichtung,
- Zweifach-Diskettenspeicher,
- TWICE-Kabel (Test Ware In Circuit Emulator),
- Datensichtgerät mit Tastatur (Konsole),
- Zeilendrucker.

Das TWIN-System ist modular aufgebaut. Man kann also bei der Anschaffung mit einer Grundausstattung, bestehend aus drei Baueinheiten mit der zugehörigen Software, beginnen und das System dann nach Bedarf weiter ausbauen.

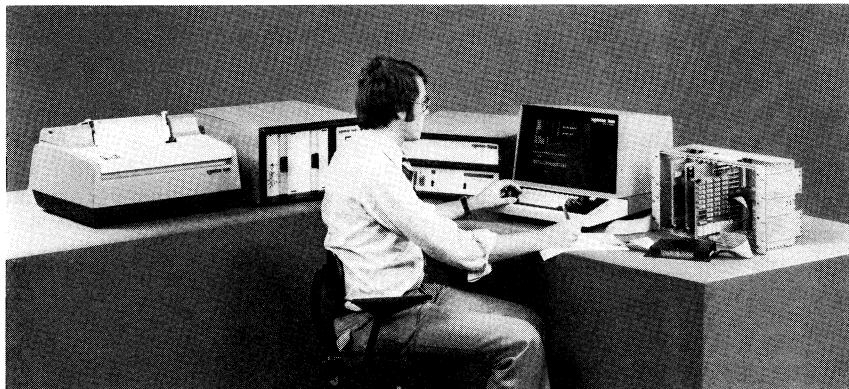


Bild 3. Die wichtigsten Geräte des TWIN-Systems (von links nach rechts):
Zeilendrucker
Zweifach-Diskettenspeicher (Dual-Floppy-Disk-Speicher)
Rechner
Datensichtgerät mit Tastatur
TWICE-Kabel, angeschlossen an eine Prototypen-Platine

Der TWIN-Rechner, der Zweifach-Diskettenspeicher und das TWICE-Kabel bilden die erwähnte Grundausstattung, zu der noch ein Ein-/Ausgabegerät, im einfachsten Fall eine Teletypemaschine hinzugefügt werden muß. Darüber hinaus können noch weitere Geräte wie z. B. ein Lochstreifenleser angeschlossen werden. In Bild 4 ist schematisch das Zusammenspiel der wichtigsten TWIN-Geräte dargestellt.

Der TWIN-Rechner, auf den in Abschn. 2.2 ausführlich eingegangen wird, ist die zentrale Funktionseinheit des Systems. Als Besonderheiten von TWIN sollen der Diskettenspeicher, das TWICE-Kabel und die PROM-Programmier-Einrichtung schon an dieser Stelle kurz erläutert werden.

Mit dem **Zweifach-Diskettenspeicher** wird die im Vorwort gestellte Forderung nach einem geeigneten Massenspeicher erfüllt. Der TWIN-Diskettenspeicher ist robust, lässt sich leicht bedienen und macht das TWIN-System zu einem sehr flexiblen Entwicklungs- und Testinstrument; denn der Benutzer kann, ohne sich um Einzelheiten kümmern zu müssen, schnell von einem Anwenderproblem auf ein anderes übergehen und die dazu erforderlichen TWIN-Hilfsprogramme von der Diskette aufrufen; er kann Dateien öffnen, ausdrucken lassen, kopieren und wieder löschen. Mehrere Entwickler können abwechselnd am selben TWIN-System mit ihren eigenen Disketten arbeiten. Schließlich eignen sich die sehr preiswerten Disketten gut zur Archivierung.

Das **TWICE-Kabel** ermöglicht die im Vorwort geforderte problemlose Verbindung zwischen TWIN-System und Anwender-Prototyp. Es erlaubt, die vom TWIN-System zur Verfügung gestellten Hilfsmittel (Peripheriegeräte, Speicher, Software) bei der Integration von Anwender-Hardware und Anwender-Software sowie beim Austesten der Prototypen wirkungsvoll einzusetzen.

Mit der **PROM-Programmier-Einrichtung** kann das Anwenderprogramm direkt vom TWIN-Rechner in einen programmierbaren Festwertspeicher (PROM) geschrieben werden. Man braucht also kein separates PROM-Programmier-Gerät und vermeidet die mit dem Umweg über Datenzwischenträger (z. B. Lochstreifen) verbundenen Schwierigkeiten.

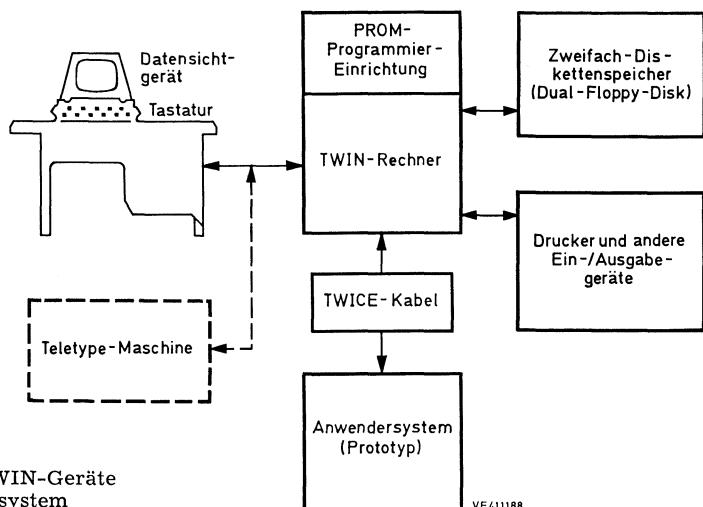


Bild 4.
Schema der TWIN-Geräte
mit Anwendersystem

Das Softwarepaket des TWIN-Systems enthält im wesentlichen folgende Programme:

- Betriebssystem SDOS (Signetics Disk Operating System),
- Editor,
- 2650-Assemblierer,
- Debug-Programm,
- PROM-Programmierer.

Alle diese Programme stehen auf Diskette zur Verfügung und werden bei jedem TWIN-System mitgeliefert. Weitere Programme sind in Vorbereitung. Die Bedeutung und die wesentlichen Eigenschaften der TWIN-Programme werden in Abschnitt 3 erläutert.

2. TWIN-Hardware

In Bild 5 ist der Aufbau des gesamten TWIN-Systems etwas ausführlicher dargestellt als in Bild 4, um die Aufgabenverteilung deutlich zu machen. Im folgenden Abschnitt sollen zunächst die Schnittstellen zwischen dem Rechner und der Peripherie behandelt werden. Im Abschnitt 2.2 werden dann der TWIN-Rechner selbst und seine einzelnen Funktionsblöcke besprochen.

2.1. Verkehr des Rechners mit TWIN- und Anwender-Peripherie

Bezüglich des Verkehrs zwischen Rechner und Peripherie kann man vier funktionelle Bereiche unterscheiden. Wir erläutern zunächst die mit dem Master-Prozessor zusammenhängenden Bereiche 1 und 2.

Im Bereich 1 spielt sich der Verkehr zwischen dem Master-Prozessor und den peripheren Hilfsgeräten, bestehend aus Diskettenspeicher, Zeilendrucker, Datensichtgerät und Tastatur, ab. Anstelle von Datensichtgerät und Tastatur lässt sich auch eine Teletypemaschine anschließen. Durch Vermittlung des Masters kann aber auch der Slave-Bereich mit diesen peripheren Geräten in Verbindung treten (Abschn. 2.2.3). Ferner lassen sich im Bereich 1 durch Hinzufügen einer Standard-Ein-/Ausgabe-Platine weitere periphere Geräte, wie Lochstreifenleser und -stanzer sowie ein Modem zur Datenfernübertragung anschließen (s. Abschn. 2.2.4).

Im Bereich 2, der ebenfalls dem Master zugeordnet ist, wird die PROM-Programmierungseinrichtung betrieben. Hier kann das Anwenderprogramm in einen programmierbaren Festwertspeicher eingeschrieben werden, den man hierzu in eine der PROM-Fassungen einsetzt. Zur Zeit können MOS-EPROMs vom Typ 1702 A und Bipolar-PROMs vom Typ 82 S 115 programmiert werden (s. Abschn. 2.2.6).

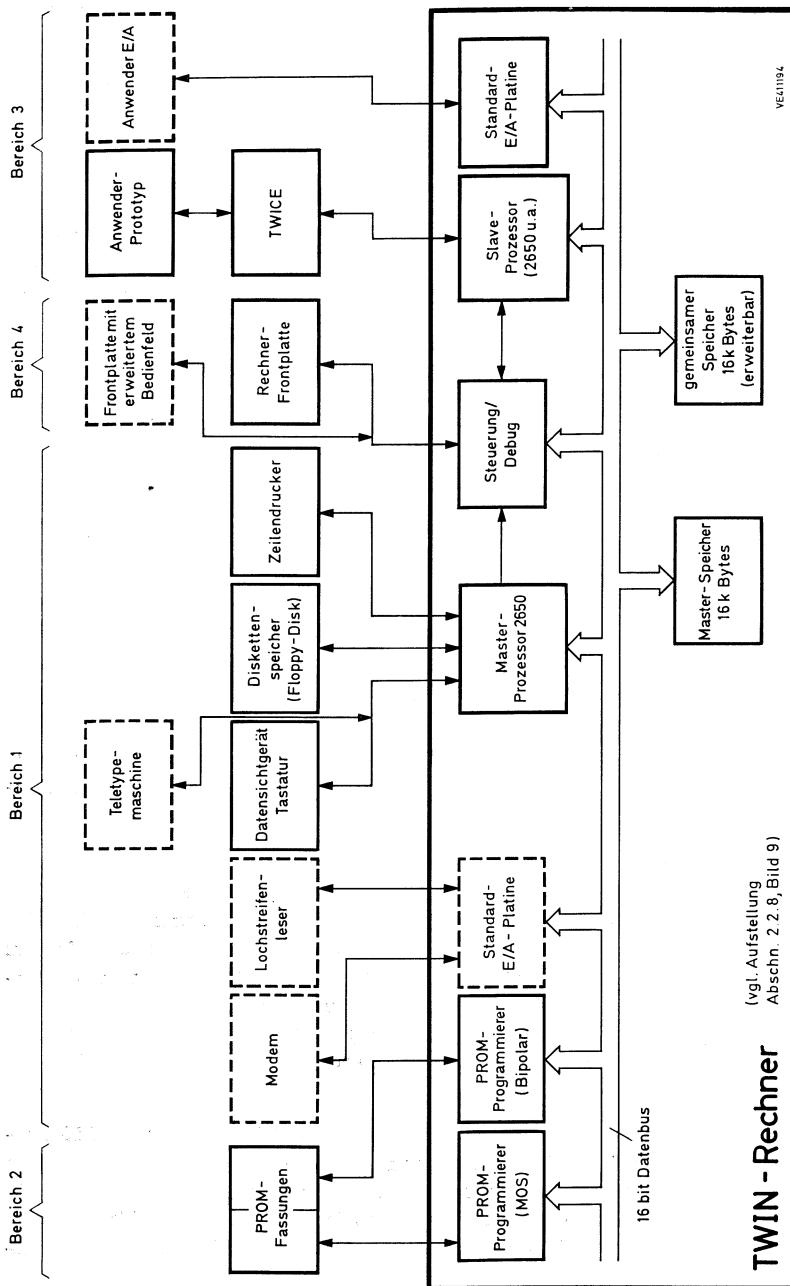


Bild 5. Blockbild des TWIN-Systems mit Anwender-Prototyp; Erweiterungsmöglichkeiten sind gestrichelt dargestellt

Im Bereich 3, der dem Slave-Prozessor zugeordnet ist, verkehren der TWIN-Rechner und das zu entwickelnde Anwendersystem miteinander. Mit Hilfe des TWICE-Kabels kann der Slave-Prozessor den Prozessor im Anwender-System ersetzen (vgl. Bilder 17 und 18). Das TWICE-Kabel enthält Pufferstufen zur elektrischen Anpassung unter Berücksichtigung der Kabellänge. Ferner ist im Bereich 3 über eine zusätzliche Standard-Ein-/Ausgabe-Platine ein direkter Anschluß (ohne TWICE) an die Anwender-Peripherieschaltungen möglich (vgl. Bild 16).

Im Bereich 4 schließlich werden die auf der Frontplatte des Rechners verfügbaren Bedien- und Anzeigefunktionen mit Hilfe des Steuer-/Debug-Moduls realisiert. Die normale Frontplatte kann durch eine Frontplatte mit zusätzlichen Bedienungsschaltern und Anzeigeelementen ersetzt werden, die in Verbindung mit Diagnostik-Programmen auch zur Prüfung des TWIN-Systems benutzt wird.

2.2. Der TWIN-Rechner

Die in Bild 5 dargestellten Funktionsblöcke des Rechners sind als steckbare Platinen ausgeführt. Ihre Aufgaben sollen in den folgenden Abschnitten näher erläutert werden. Alle Platinen sind an eine gemeinsame Busstruktur angeschlossen. Der Hauptbus enthält einen Datenbus, einen Adreßbus und einen Steuerbus. Der Datenbus und Adreßbus sind jeweils für 16 bit ausgelegt; die Treiberschaltungen wurden so dimensioniert, daß sie auch für zukünftige Mikroprozessoren mit kürzeren Zykluszeiten geeignet sind.

Die peripheren Geräte und die Debug-Logik werden nach dem Unterbrechungsverfahren (interrupt) und nicht durch zyklisches Abfragen (polling) bedient. Das heißt, jede Unterbrechungsanforderung kann nahezu unmittelbar bearbeitet werden. Entsprechend schnell ist der Datendurchsatz beim TWIN-System.

2.2.1. Master-Slave-Prinzip

Wie erwähnt, enthält der TWIN-Rechner zwei Prozessoren, den Master-Prozessor und den Slave-Prozessor.

Der **Master**, ein Mikroprozessor 2650, arbeitet als Hauptprozessor. Die entsprechende Software wird durch das TWIN-Betriebssystem gebildet. Der Master bedient Unterbrechungsanforderungen nach ihrer Priorität, unabhängig davon, ob sie aus dem Master- oder dem Slave-Bereich kommen. Der Master ist für alle Funktionen des TWIN-Systems zuständig, die nicht mit dem Anwendersystem zusammenhängen, hierzu gehören im wesentlichen:

- das Verwaltungsprogramm für den Diskettenspeicher (Abspeichern und Laden von Programmen und Daten; Verwaltung der Dateien);

- die Ein-/Ausgabe-Funktionen für periphere TWIN-Geräte in den Bereichen 1 und 2 (Bild 5; siehe auch Abschn. 2.2.4);
- die Debug-Funktionen (der Master-Prozessor führt das Debug-Programm aus und steuert dabei den Slave-Prozessor mit Hilfe des Steuer-/Debug-Moduls (Bild 5)).

Auch bei Programmen, die an sich vom Slave-Prozessor ausgeführt werden, wie z. B. dem Editor, wird der Verkehr mit den peripheren TWIN-Geräten vom Master-Prozessor gesteuert. Das geschieht dadurch, daß der Slave-Prozessor, sobald ein solcher Datenverkehr verlangt wird, eine entsprechende Anforderung (supervisor call) an den Master-Prozessor richtet (Abschn. 2.2.3).

Der **Slave**-Prozessor führt vor allem anwenderbezogene Aufgaben aus. Er steht dem TWIN-Benutzer als Prozessor für das Anwendersystem zur Verfügung. Außerdem führt er einige Hilfsprogramme aus. Dabei ist in jedem Fall sichergestellt, daß unerlaubte Eingriffe in das TWIN-Betriebssystem vermieden werden. In der gegenwärtigen TWIN-Ausführung ist der Slave ein Mikroprozessor 2650. Es können jedoch in Verbindung mit entsprechender Software zukünftig auch andere Mikroprozessortypen eingesetzt werden. Hierbei sind aufgrund der erwähnten Busstruktur (16-bit-Adreßbus und 16-bit-Datenbus) 16-bit-Mikroprozessoren mit einem Adressierungsbereich von 64 k einbezogen. Im wesentlichen fallen folgende Funktionen in den Aufgabenbereich des Slave-Prozessors:

- die Ausführung des Editor- und Assemblerprogramms;
- die Abarbeitung von Anwenderprogrammen, wobei der Slave-Prozessor als ein überwachbarer Prozessor des Anwendersystems eingesetzt wird;
- das Anfordern von Verbindungen zu TWIN-Peripheriegeräten über den Steuer-/Debug-Modul (Bild 5), wenn Daten zwischen dem Slave-Bereich und der TWIN-Peripherie ausgetauscht werden sollen;
- die Einbeziehung der Anwenderperipherie in Programm-Testläufe ohne TWICE-Kabel (Abschn. 2.2.4).

Damit sind die Hauptaufgaben von Master- und Slave-Prozessor beschrieben. Folgende Eigenschaften des TWIN-Systems beruhen wesentlich auf der Master-/Slave-Architektur:

- Es kann der gesamte Adressierungsraum des gemeinsamen Speichers (64 k) und der E/A-Adressierungsraum des Slave-Prozessors ausgenutzt werden. (Mit dem 2650 als Slave-Prozessor sind maximal 4×8 k Adressen des gemeinsamen Speichers erreichbar.)
- Die peripheren TWIN-Geräte und der Master-Speicher sind gegen ungetestete Anwenderprogramme vollkommen geschützt. Sie sind ausschließlich durch einwandfreie System-Routinen beeinflußbar, die vom Master-Prozessor ausgeführt werden. Das heißt, ein Anwenderprogramm kann, wenn es im Slave-Bereich zu Prüzfzwecken abläuft, nicht versehentlich auf die Diskette schreiben und dort das Betriebssystem oder andere Dateien zerstören.

- Neue Prozessortypen können im Slave-Bereich in Verbindung mit entsprechender Software hinzugefügt werden. Die Betriebssystem-Software von TWIN kann weitgehend unverändert bleiben, weil weiterhin derselbe Master-Prozessor verwendet wird.

2.2.2. Getrennte Speicher

Der TWIN-Rechner enthält – in Anlehnung an die Zwillings-Prozessor-Architektur – auch zwei Speicher, den Master-Speicher und den gemeinsamen Speicher (Bild 5). Auf letzteren können sowohl Master als auch Slave-Prozessor zugreifen, während der Master-Speicher gegen jeglichen Zugriff von Seiten des Slave-Prozessors und der von ihm bearbeiteten Anwenderprogramme vollkommen geschützt ist. Durch diese Trennung der Speicher können die in den Master-Speicher geladenen Programme, wie z. B. das Betriebssystem, nicht versehentlich durch Anwenderprogramme überschrieben werden.

Der **Master-Speicher** ist für die internen Überwachungsprogramme und das Debug-Programm vorgesehen. Außerdem wird dieser Speicher von der Diskette mit den Dienst-Programmen geladen, die den Dialog zwischen Benutzer und TWIN-System ermöglichen. Die Speicherkapazität beträgt in der Grundausstattung 16 k Bytes. Außerdem gehören zum Master-Speicher 256 Bytes PROM-Kapazität, die auf 2 k Bytes erweitert werden können.

Der **gemeinsame Speicher** ist primär für Anwenderprogramme vorgesehen, aber auch Editor- und Assemblerprogramm sowie ein Teil des Debug-Trace-Programms werden in diesen Speicher geladen, wenn der Slave-Prozessor sie ausführen soll. Die Kapazität des gemeinsamen Speichers kann bis auf 64 k Bytes erweitert werden. Diese Kapazität wurde mit Rücksicht auf zukünftige Slave-Prozessortypen festgelegt. Der zur Zeit verwendete Slave-Prozessor 2650 kann maximal 4×8 k Adressen aufrufen.

Mittels eines speziellen Kommandos kann man eine der vier Bänke des gemeinsamen Speichers von je 16 k Bytes dem Master-Prozessor zuordnen. Die ausgewählte Bank wird dann vom Master-Prozessor mit den Adressen 16 k bis 32 k angesprochen, wie es in Bild 6 dargestellt ist. Auf diese Bank hat natürlich auch der Slave-Prozessor Zugriff. Somit ist es also möglich, Daten zwischen Master- und Slave-Prozessor auszutauschen. Zum Beispiel kann im Debug-Modus mit dem Befehl TRACE der Inhalt der Slave-Prozessor-Register in den gemeinsamen Speicher übertragen, dort vom Master-Prozessor geholt und auf dem Datensichtgerät angezeigt werden.

Ebenso kann der Benutzer mit Hilfe der Tastatur den Inhalt des gemeinsamen Speichers ändern und auf diese Weise zum Beispiel in das Anwenderprogramm eingreifen, indem er Inhalte von Speicherplätzen oder Registern im Verlauf eines Debug-Prozesses ändert (PATCH, Anhang A.1.4).

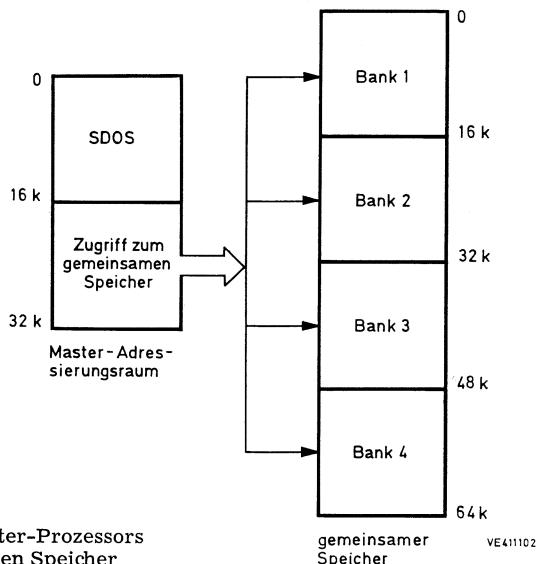


Bild 6.
Zugriff des Master-Prozessors
zum gemeinsamen Speicher

VE411102

2.2.3. Master-Slave-Wechselwirkungen

Von den beiden Prozessoren – Master und Slave – ist jeweils nur einer aktiv, d. h., nur einer bearbeitet Befehle. Welcher das ist, wird vom Betriebssystem SDOS bestimmt. Ein HALT-Befehl an den Master-Prozessor gibt den Slave-Prozessor-Betrieb frei. Wenn dagegen der Master-Prozessor arbeitet, wird der Slave-Prozessor stillgesetzt. Der Master-Prozessor kann den Slave-Bereich mit einer Reihe von Steuersignalen beeinflussen. Hierzu gehören unter anderem:

- Laden des gemeinsamen Speichers aufgrund entsprechender Benutzer-Kommandos, bevor der Slave-Prozessor zu arbeiten beginnt;
- Erteilen eines Rücksetz- oder Sprungbefehls an den Slave-Prozessor;
- Setzen von zwei Haltepunkt-Registern (mit Lese-/Schreib-Kontrolle) für das im Slave-Bereich laufende Programm (Anwenderprogramm).

Wenn das vom Slave-Prozessor bearbeitete Programm einen Datenaustausch mit den peripheren TWIN-Geräten verlangt, gibt der Slave-Prozessor eine entsprechende Anforderung an den Master ab, der dann die Ein-/Ausgabe-Operationen ausführt, während der Slave-Prozessor solange anhält. Nach Abschluß der Ein-/Ausgabe-Operationen gibt der Master den Slave wieder frei, so daß er in der Ausführung seines Programms fortfahren kann.

Auch beim Debug-Betrieb treten Master- und Slave-Prozessor abwechselnd in Aktion. Der Master lädt die Adressen der vom Benutzer angegebenen Haltepunkte in die hierfür vorgesehenen Register. Der Slave arbeitet dann das Anwenderprogramm ab, bis er zu einer Haltepunkt-Adresse kommt. Der Programmablauf wird jetzt unterbrochen, und die Steuerung geht an den Master-Prozessor zurück. Mit Hilfe der Debug-Software kann nun – gesteuert vom Master-Prozessor – das Anwenderprogramm überprüft werden.

2.2.4. Ein-/Ausgabe- und Unterbrechungs-Möglichkeiten des Master- und Slave-Prozessors und der Standard-E/A-Platine

Master- und Slave-Bereich haben separate Ein-/Ausgabe- und Unterbrechungs-Möglichkeiten. Für den **Master-Prozessor** gilt:

- Adressierungsbereich für 256 Ein-/Ausgabeeinheiten,
- 32 Prioritätsstufen für gerichtete Programm-Unterbrechungen,
- Anschluß für Diskettenspeicher und Zeilendrucker
- Anschluß (Schnittstelle 20 mA oder EIA/RS-232-C) für TWIN-Datensichtgerät mit Tastatur oder Teletypemaschine.

Mit der Standard-E/A-Platine, die sowohl auf der Master- als auch auf der Slave-Seite hinzugefügt werden kann (Bild 5), bestehen folgende zusätzliche Ein-/Ausgabe-Möglichkeiten:

- 20 mA, 110 Baud TTY-Schnittstelle,
- EIA/RS-232 C-Schnittstelle, steckbar für 110/150/300/600/1200 Baud,
- vier Eingangstore zu je 8 bit
- vier Ausgangstore mit Halteregister zu je 8 bit } TTL-kompatibel,
- acht Flipflops zur Programm-Unterbrechung.

Somit können zum Beispiel schnelle Lochstreifen- oder Lochkartenleser sowie Modems angeschlossen werden.

Für den **Slave-Prozessor** gilt:

- Adressierungsbereich für 256 Ein-/Ausgabeeinheiten,
- acht Prioritätsstufen für gerichtete Programm-Unterbrechungen.

Der Einsatz der Standard-E/A-Platine auf der Slave-Seite ermöglicht eine direkte Kommunikation zwischen dem Slave-Prozessor und der Anwender-Peripherie wie z. B. einer Tastatur oder einem Drucker. So können bereits im Entwicklungsstadium eines Anwenderprogramms – noch bevor ein Prototyp erstellt ist und das TWICE-Kabel zum Einsatz kommen kann – Probeläufe in einer Hardware-Umgebung durchgeführt werden, die der endgültigen ähnelt (vgl. Bild 16).

2.2.5. Steuer-/Debug-Platine

Der in Bild 5 mit Steuerung/Debug bezeichnete Block hat zwei Aufgaben. Er stellt die bereits erwähnte Verbindung zur Frontplatte des TWIN-Rechners her und bildet das Bindeglied zwischen Master- und Slave-Prozessor.

Im einzelnen werden folgende Funktionen ausgeführt oder gesteuert:

- Initialisierung des Systems beim Einschalten,
- Wechselspiel zwischen Master- und Slave-Aktivität,
- Zugriff des Master-Prozessors zum gemeinsamen Speicher (Bild 6),
- Kommunikation des Slave-Prozessors mit den peripheren TWIN-Geräten,
- Überwachung der Haltepunkte bei Probeläufen des Anwenderprogramms,
- Rücksetzen des Slave-Prozessors und Neubeginn des Programms bei vor-gegebener Adresse,
- direkter Speicherzugriff.

Die Debug-Logik enthält die beiden bereits erwähnten, voneinander unabhängigen Vergleichsregister für die Adressen der Haltepunkte sowie einen Pseudoprogrammzähler, der den Stand des Slave-Befehlszählers festhält. Die Verfolgung des Programmablaufs (dynamic trace) mit Hilfe der Debug-Logik kann entweder für das ganze Programm oder für einen ausgewählten Programmabschnitt zwischen zwei vorzugebenden Befehlsadressen Befehl für Befehl oder aber jeweils nur bei den Sprungbefehlen vorgenommen werden, wodurch die Ablaufverfolgung schneller vor sich gehen kann.

2.2.6. PROM-Programmier-Platinen

Die beiden PROM-Programmier-Platinen (Bild 5) sind ausgelegt für das Bipolar-PROM 82 S 115 und das löschrbbare MOS-PROM (EPROM) 1702 A. Sie können durch Platinen für andere PROM-Typen ersetzt werden.

Wenn das Anwenderprogramm assembledt worden ist, kann es unmittelbar aus dem gemeinsamen Speicher in einen programmierbaren Festwertspeicher überschrieben werden. Hierfür braucht man kein zusätzliches Gerät; der Master-Prozessor übernimmt diese Aufgabe.

Die PROM-Programmier-Platinen erfüllen unter anderem folgende Aufgaben:

- Regelung der erforderlichen Versorgungsspannungen auf der Platine,
- Strombegrenzung und Überspannungsschutz, um die PROMs nicht zu überlasten,
- Schreiben, Lesen und Vergleichen der zu programmierenden bzw. programmierten Daten,
- Steuerung der Adressen und der Anzahl der Programmzyklen,
- Kontrolle der Adressen und der Anzahl der Programmzyklen.

Durch die zuletzt erwähnte Kontrolle ist sichergestellt, daß aufgrund von Bauelemente-Toleranzen der PROMs kein unvollständig programmiertes Bit übrigbleibt. Die Spannungsversorgung der Programmierungseinrichtung kann mit einem Schalter auf der Frontplatte abgeschaltet werden, damit die PROM-Schaltungen beim Hineinstecken oder Herausziehen nicht zerstört werden.

2.2.7. TWICE-Hardware

Die TWICE-Hardware (Test Ware In Circuit Emulator) besteht aus Treiber- und Empfängerschaltungen zur elektrischen Anpassung unter Berücksichtigung der Kabellänge und den Anschlußkabeln mit Steckern. Auf der einen Seite wird das TWICE-Kabel mit seinem 40poligen, zum Mikroprozessor 2650 angeschlußkompatiblen Stecker in eine entsprechende Fassung des Anwender-Prototyps gesteckt (Bild 3). Die andere Seite schließt man an die Slave-Prozessor-Platine an. Die Funktion des TWICE-Kabels wird anhand der Bilder 17 und 18 erläutert.

2.2.8. Kurze Gerätebeschreibung

Der in Bild 7 gezeigte Rechner hat folgende Bedienungs- und Kontrollelemente:

- Schaltschloß zum Ein- und Ausschalten;
- Fenster mit Leuchtanzeigen

PWR	Rechner ist eingeschaltet
MSTR	Master-Prozessor ist aktiv
SLV	Slave-Prozessor ist aktiv
RUN	Programm wird abgearbeitet;
- Taste DIAG INT für Diagnostikzwecke;
- Taste RESET unterbricht jeden Programmlauf, initialisiert Hardware und bewirkt erneutes Laden des Betriebssystems SDOS;
- Schalter PROM POWER schaltet Stromversorgung für PROM-Programmier-Einrichtung ein; dies wird durch die darüberliegende Lampe angezeigt;
- IC-Fassungen mit Spannhebeln, z. Z. für MOS-PROM 1702 A und Bipolar-PROM 82 S 115. Fassung PROM 3 für zukünftigen Gebrauch.

Auf der Rückfront des Rechnergehäuses (Bild 8) befindet sich u. a. der RS-232-Anschluß für das TWIN-Datensichtgerät oder eine Teletypemaschine. Durch die beiden großen Öffnungen werden das Vielfach-Flachkabel für den Anschluß des Diskettenspeichers an die Master-Prozessor-Platine und des TWICE-Kabels an die Slave-Prozessor-Platine geführt.



Bild 7. Frontansicht des TWIN-Rechners

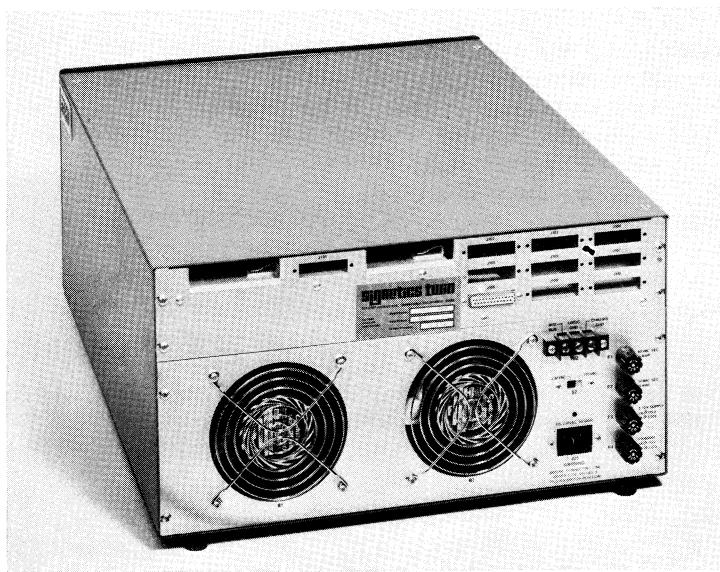


Bild 8. Rückwand des TWIN-Rechners

Die in Bild 9 erkennbaren Flachkabel verbinden die Programmier-Platinen und die Steuer-/Debug-Platine mit der Frontplatte und die Master-Platine mit der RS-232-Fassung in der Rückwand. In der folgenden Aufstellung sind die zur Super-TWIN-Ausstattung gehörenden Platinen aufgeführt.

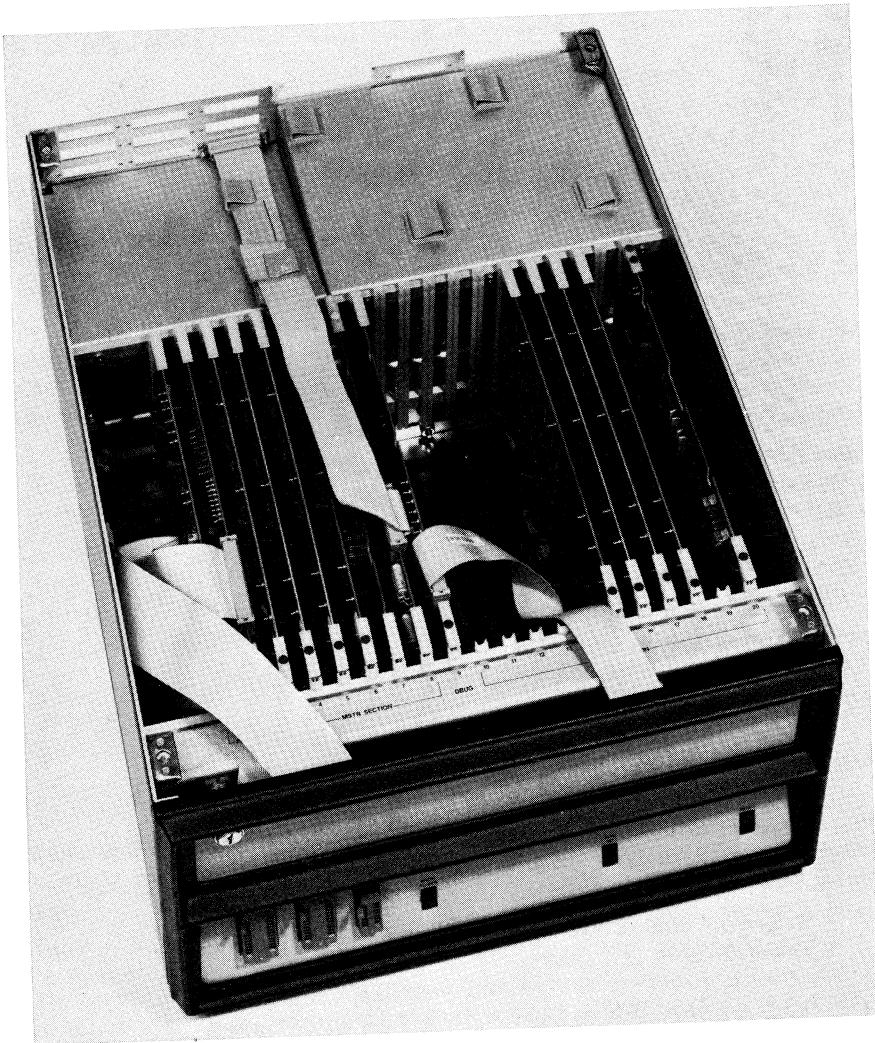


Bild 9. Blick in den TWIN-Rechner (Kabel für Diskettenspeicher und TWICE nicht angeschlossen)

Platinen-Anordnung im TWIN-Rechner (Super-TWIN-Ausstattung), vgl. Bild 9

Position	Bezeichnung
J 1	PROM-Programmierer, MOS 1702 A
J 2	PROM-Programmierer, Bipolar 82 S 115
J 3	Standard-Ein-/Ausgabe
J 4	Master-Speicher, 4 k Bytes RAM/2 k Bytes PROM
J 5 . . . J 7	Master-Speicher je 4 k Bytes RAM
J 8	Master-Prozessor 2650
J 9	Steuerung/Debug
J 10	frei
J 11 . . . J 14	gemeinsamer Speicher, je 4 k Bytes RAM
J 15 . . . J 19	frei
J 20	Slave-Prozessor 2650

2.3. Zweifach-Diskettenspeicher

Als Massenspeicher des TWIN-Systems wird ein Diskettenspeicher verwendet. Das in Bild 10 gezeigte Gerät enthält zwei Disketten-Laufwerke, die Steuer-Einheit und die Stromversorgung. Folgende Informationen werden auf der Diskette gespeichert:

System-Programme

- diskettenorientiertes Betriebssystem SDOS
- Editor
- Assemblierer
- Debug-Programm
- PROM-Programmierer

Anwenderprogramme und -daten

- Quellenprogramme
- Objektprogramme
- Texte

Jedes Programm wird als Datei gespeichert (Abschn. 3.1.2).

Eine Diskette besteht aus magnetisch beschichtetem biegsamem Kunststoff (daher: floppy disk). Die Aufteilung der Speicherbereiche auf der Diskette geht aus Bild 11 und den folgenden Angaben hervor:

- 77 Spuren auf einer Diskette,
- 4 Blöcke je Spur,
- 8 Sektoren je Block, d. h. 32 Sektoren auf dem Umfang,
- 128 Bytes je Sektor, d. h. ca. 315 k Bytes \approx 2,5 M bit je Diskette,
- 360 Umdrehungen/Minute,
- 250 k bit/s Daten-Transfersgeschwindigkeit,
- 10 ms Positionierungszeit von Spur zu Spur.

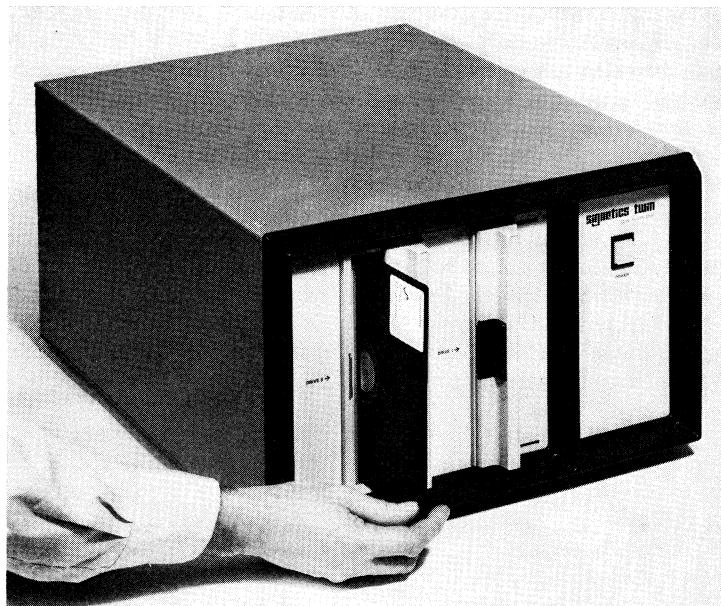


Bild 10. Einschieben einer Diskette in den TWIN-Zweifach-Diskettenspeicher. Die Diskette bleibt dabei in ihrer Plastikschutzhülle.

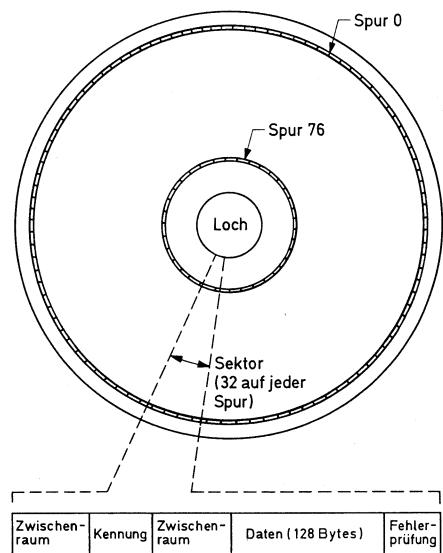


Bild 11.
Anordnung der Speicher-
bereiche auf der Diskette

Der Sektor mit einer Kapazität von 128 Bytes = 1024 bit ist die Daten-Grundeinheit der Diskette. Spur 0 ist für das Dateien-Verzeichnis der betreffenden Diskette reserviert. Dieses Verzeichnis enthält unter anderem die Namen der Dateien, die verwendeten Sektoren, die Anfangs- und Endadressen der Dateien sowie Informationen, die eine Benutzung unbrauchbarer Disketten-Sektoren verhindert. Auf den Spuren 1 bis 15 sind die oben erwähnten Systemprogramme untergebracht.

Bevor man das erstmal Daten auf einer Diskette speichert, müssen bestimmte Informationen eingeschrieben werden, die im wesentlichen die Einteilung in Sektoren betreffen. Das nennt man formatieren. Die zum TWIN-System gelieferten Disketten sind bereits formatiert und enthalten die Systemprogramme. Andere Disketten können vor der Benutzung mit Hilfe von TWIN formatiert werden (Anhang A.1.1).

Die Steuer-Einheit im Diskettenspeicher enthält unter anderem einen Pufferspeicher für 128 Bytes, um den asynchronen Datenverkehr mit der Diskette zu ermöglichen. Ferner werden Funktionen wie Blockbildung von Daten, Anlauf des Systems beim Einschalten und Wiederholungsversuche bei Schreib- oder Lesefehlern realisiert. Die Steuer-Einheit läßt sich erweitern, so daß sie bis zu 8 Disketten bedienen kann. (Erweiterungsanschluß auf der Rückseite des Diskettenspeichers.)

2.4. Periphere Geräte

Das **Datensichtgerät mit Tastatur** (Bild 3) ist das wichtigste Ein-/Ausgabe-Gerät des TWIN-Systems. Die Verbindung zum TWIN-Rechner erfolgt über eine Schnittstelle RS-232-C. Die ASCII-Tastatur entspricht im wesentlichen einer Standard-Schreibmaschinen-Tastatur mit einigen zusätzlichen Funktionstasten. Datensichtgerät und Tastatur kann man getrennt aufstellen.

Anstelle von Datensichtgerät und Tastatur kann man als Ein-/Ausgabe-Gerät eine **Teletypemaschine** (20 mA Stromschleife oder RS-232-C) anschließen, womit auch Lochstreifen gelesen oder gestanzt werden können.

Außerdem kann man, wie in Abschn. 2.2.4 beschrieben, mit Hilfe der Standard-E/A-Platine im Master-Bereich weitere Anschlußmöglichkeiten für Geräte mit einer Schnittstelle **RS-232-C** sowie für **TTL-kompatible** 8-bit-Parallel-Ein- und -Ausgänge schaffen. Falls diese peripheren Schaltungen oder Geräte dem TWIN-Betriebssystem zugeordnet werden sollen, ist für ihre Steuerung zusätzliche Software erforderlich. Schließlich kann man einen **Zeilendrucker** Centronics 306 C, passend zum TWIN-System, anschließen (vgl. Bild 3). Er druckt 120 Zeichen pro Sekunde bei einer Zeile von 80 Zeichen und 160 Zeichen pro Sekunde bei einer Zeile von 120 Zeichen.

3. TWIN-Software

Die Software zum TWIN-System besteht aus dem Betriebssystem SDOS, dem Editor, dem Assemblierer, dem Debug-Programm und dem PROM-Programmierer. Auf diese Programme wird in den folgenden Abschnitten kurz eingegangen. Eine vollständige Beschreibung für die Benutzung dieser Programme sowie eine Aufzählung aller Kommandos und Meldungen mit Definitionen und Erläuterungen findet sich zusammen mit zahlreichen Beispielen in den ausführlichen Unterlagen, die mit dem TWIN-System geliefert werden.

3.1. Betriebssystem SDOS

3.1.1. Aufgaben von SDOS

Das Betriebssystem SDOS (Signetics Disk Operating System) ist für die internen Funktionen des TWIN-Systems zuständig. Der TWIN-Benutzer braucht sich also nicht im Detail mit dem internen Funktionsablauf zu beschäftigen. Mit Hilfe zahlreicher SDOS-Kommandos (Anhang A.1) kann man die Arbeitsweise aller TWIN-Funktionseinheiten steuern, wobei das Programm SDOS u. a. folgende Aufgaben ausführt:

- Verwalten von Dateien,
- Festlegen und Zuordnen von Ein- und Ausgabe-Kanälen,
- Ausführen von Dienst-Programmen,
- Laden von Programmen,
- Manipulieren und Ändern von Anwenderprogrammen,
- Ausführen und Überwachen von Anwenderprogramm-Abläufen,
- Anzeigen wichtiger System-Zustände.

Das in ASSEMBLER 2650 geschriebene Programm SDOS wird vom Master-Prozessor bearbeitet. 256 Bytes von SDOS sind in einem hierfür eigens vorgesehenen PROM abgelegt, weitere 16 k Bytes werden in den gegen unerlaubte Zugriffe geschützten Master-Speicher geladen.

3.1.2. Dateien, Ein-/Ausgabegeräte, Kanäle

Wesentlich für die Struktur des Betriebssystems SDOS ist das Arbeiten mit Dateien. Eine Datei ist eine Zusammenstellung von Daten für einen bestimmten Zweck. Beim TWIN-System sind Dateien auf Disketten gespeichert. Das Aufrufen einer Datei erfolgt durch Nennung ihres Namens, und der Zugriff wird mit Hilfe des vom Programm SDOS angelegten Dateien-Verzeichnisses vorgenommen, das auf der betreffenden Diskette in Spur 0 gespeichert ist und Auskunft darüber gibt, wo die gewünschte Datei abgelegt ist.

Alle diese mit der Dateienverwaltung zusammenhängenden Aufgaben werden vom Betriebssystem SDOS selbständig erledigt. Als Beispiel sei das Aufrufen einer Datei innerhalb eines Dienst-Kommandos (Anhang A.1.1) angeführt. Das Kommando

```
PRINT PROG 1; S/0 CONO 50 100
```

hat folgende Bedeutung:

Gib die Zeilen 50 bis 100 der Datei mit dem Namen PROG 1; S, die im Diskettensystem 0 abgelegt ist, auf dem Datensichtgerät (CONsole Output) aus.

Mit Diskettensystem ist hier eines der beiden erwähnten Disketten-Laufwerke gemeint.

Die fünf Standard-Peripherie-Geräte (Tastatur und Datensichtgerät bzw. Teletypemaschine, Zeilendrucker, schneller Lochstreifenleser und Teletype-Lochstreifenleser) haben feste Namen, wie z. B. CONO, unter denen sie vom Benutzer aufgerufen werden können. Zum Beispiel veranlaßt das SDOS-Kommando

```
COPY TTYR LPT 1
```

ein Übertragen der Information vom Teletype-Lochstreifenleser (Tele-TYpe Reader) auf den Zeilendrucker 1 (Line PrinTer 1).

Beim Betriebssystem SDOS kann eine Datei wie ein peripheres Gerät behandelt werden, indem der Benutzer anstatt eines Gerätenamens einen Dateinamen und ggf. noch den Namen des Diskettenlaufwerks angibt. Um z. B. den Inhalt der Datei mit dem Namen DATA 1, die sich auf der Diskette im Laufwerk 1 befindet, auf den Zeilendrucker 1 (Line PrinTer 1) auszudrucken, muß der Benutzer folgendes Kommando geben:

```
COPY DATA 1/1 LPT 1
```

Das Betriebssystem SDOS bietet die Möglichkeit, sogenannte Kanäle zu verwenden, die beim Abarbeiten von Slave-Programmen während der Debug-Phase zur Anwendung kommen. Der Benutzer kann mit Hilfe des SDOS-Kommandos ASSIGN acht logische Kanäle des Slave-Prozessors mit den Ein-/Ausgabegeräten des TWIN-Systems oder auch mit Dateien verbinden. Auf diese Weise sind Ein-/Ausgabe-Operationen zwischen dem Slave-Prozessor und den TWIN-Dateien bzw. peripheren TWIN-Geräten möglich.

3.1.3. Kommando-Dateien

Das Betriebssystem SDOS bietet die Möglichkeit, sogenannte Kommando-Dateien einzurichten. Darunter versteht man eine Folge mehrerer SDOS-Kommandos, die dadurch zur Ausführung kommen, daß man nur ein einziges selbstgewähltes Kommando – den Namen der Kommando-Datei –

eingibt. Hiervon wird man Gebrauch machen, wenn bestimmte Kommando-folgen häufig ausgeführt werden sollen. Dabei können auch Parameter einge-führt werden.

3.2. Editor

Die Hauptaufgabe des TWIN-Text-Editors besteht darin, das Aufnehmen neuer Quellenprogramme oder das Ändern vorhandener Quellenprogramme zu ermöglichen. Auch bereits auf der Diskette abgespeicherte Programme können mit Hilfe des Editors geändert werden. Die vom Editor angefor-derten Ein-/Ausgabe-Operationen erledigt das Betriebssystem SDOS.

Der Editor beansprucht 7000 Bytes im gemeinsamen Speicher (Bild 5) und wird vom Slave-Prozessor ausgeführt. Der Rest des gemeinsamen Speichers steht für den einzugebenden bzw. zu korrigierenden Text zur Verfügung. Bei einem gemeinsamen Speicher von 16 k Bytes sind das etwa 150 Zeilen zu je 60 Zeichen.

Dieser Bereich im gemeinsamen Speicher wird Pufferspeicher genannt. Die Daten werden vom Editor in diesen Puffer eingeschrieben oder aus ihm gelesen. Der Puffer ist in Zeilen eingeteilt. Der Editor kann auf jede dieser Zeilen zugreifen. Er kann Zeilen prüfen, ändern, einfügen und ersetzen. Mit Hilfe des sogenannten Zeilenzeigers markiert der Editor (softwaremäßig) diejenige Zeile, die vom TWIN-Benutzer jeweils bearbeitet wird.

Der Editor hat zwei grundlegende Arbeitsweisen, den **EDIT-Modus** und den **INPUT-Modus**. Während im EDIT-Modus sämtliche Editor-Funktionen ausgeführt werden können, ist der INPUT-Modus ausschließlich für das Eingeben von Quellentexten vorgesehen. Der Übergang von einem Modus zum anderen erfolgt durch sehr einfache Kommandos.

Einen Eindruck vom Arbeiten des Editors vermitteln auch die Bilder 15a, b und d.

3.3. Assemblierer

Das mit Hilfe des Editors eingegebene und auf der Diskette gespeicherte Quellenprogramm muß in ein ladefähiges Maschinenprogramm (Objekt-programm) übersetzt werden. Hierzu dient der TWIN-Assemblierer. Dieses Programm wird ebenso wie der Editor in den gemeinsamen Speicher ge-laden und vom Slave-Prozessor ausgeführt. Den noch verfügbaren Teil des gemeinsamen Speichers benutzt der Assemblierer für Ein-/Ausgabe-Puffer und – da es sich um einen Zweischritt-Assemblierer handelt – für das Auf-stellen von Symboltabellen. Das Betriebssystem SDOS bedient alle Ein-/Ausgabe-Anforderungen des Assemblierers.

Der Assemblierer ist wie alle Entwicklungshilfen des TWIN-Systems zunächst für den Mikroprozessor 2650 vorgesehen, d. h., er übersetzt Quellenprogramme in der Sprache ASSEMBLER 2650 in Objektprogramme, die vom Slave-Prozessor, einem Mikroprozessor 2650, ausgeführt werden können.

Der TWIN-Assemblierer 2650 erlaubt u. a.:

- bedingtes Assemblieren,
- symbolische, relative und indizierte Adressierung,
- Vorwärtsverweise,
- formatfreie Quellentext-Eingabe,
- Verwendung selbstdefinierender Konstanten,
- Verarbeitung komplexer Ausdrücke,
- Datenverarbeitung in verschiedenen Zahlensystemen sowie in Form von ASCII-Zeichen.

Vor dem Assemblieren muß das Quellenprogramm als Datei auf einer Diskette gespeichert worden sein. Um den Assemblierer aufzurufen und das Assemblieren zu veranlassen, braucht der Benutzer dann lediglich ein SDOS-Kommando wie z. B.

ASM TEST; S/0 TEST; L/0 TEST; H/0

mit folgender Bedeutung an das TWIN-System zu geben:

Assembliere das Programm aus der Datei mit dem Namen TEST; S (Source), die sich auf der Diskette im Laufwerk 0 befindet, und lege – ebenfalls auf dieser Diskette – eine Datei TEST; L (List) für die sogenannte Liste des assemblierten Programms sowie eine Datei TEST; H für das hexadezimal notierte Maschinenprogramm an.

Aufgrund eines solchen Kommandos führt der Assemblierer im wesentlichen folgende Aufgaben durch:

- Er übersetzt das Programm aus der angegebenen Source-Datei in ein Objektprogramm in hexadezimaler Form, das in die vom Benutzer festgelegte Objekt-Datei geschrieben wird.
- Er erzeugt eine sogenannte Liste (Bild 15 f). Sie enthält das Quellenprogramm einschließlich der Pseudobefehle, die beim Assemblieren gebildeten Befehlsadressen, das erhaltene Objektprogramm (hexadezimal) und alle Fehlermeldungen. Diese Liste wird ebenfalls in einer Datei abgespeichert.
- Er meldet, wie viele Fehler vorliegen.

3.4. Debug-Programm

Das Debug-Programm wird als Subsystem des Betriebssystems SDOS im Master-Bereich des TWIN-Systems abgearbeitet. Alle Ein-/Ausgabe-Funktionen werden vom Betriebssystem SDOS ausgeführt.

Das Debug-Programm unterstützt den TWIN-Benutzer bei der Fehlersuche und Fehlerbeseitigung im Anwendersystem in folgender Weise:

- Es bringt Register- und Speicherinhalte des Slave-Prozessors und des gemeinsamen Speichers zur Anzeige und erlaubt eine Änderung dieser Werte.
- Es steuert den Ablauf von Anwenderprogrammen und erlaubt eine Kontrolle dieser Programme an bestimmten Stellen durch Setzen von Haltepunkten.
- Es ermöglicht ein schrittweises Verfolgen des Programmablaufs (TRACE) und die Anzeige wichtiger Prozessorzustände.

Um den Zustand des Anwendersystems testen oder ggf. ändern zu können, kann man mit Hilfe des Debug-Programms Haltepunkte setzen. Darunter versteht man bestimmte, vom Benutzer vorgegebene Befehlsadressen, bei denen der Programmablauf unterbrochen wird.

3.5. PROM-Programmierer

Innerhalb des Programms SDOS gibt es einige Kommandos, mit denen man die in die Programmier-Fassungen eingesetzten programmierbaren Festwertspeicher programmieren, lesen oder ihren Inhalt mit dem des gemeinsamen Speichers im TWIN-Rechner vergleichen kann. Die hierfür erforderlichen Programmteile werden PROM-Programmierer genannt. Für die verschiedenen Typen der auf dem Markt befindlichen programmierbaren Festwertspeicher ist unterschiedliche Software und Hardware erforderlich; sie steht gegenwärtig für zwei PROM-Typen zur Verfügung (Abschn. 2.2.6).

Das TWIN-System kann man auch ganz allgemein zum Programmieren von PROMs benutzen. Hierfür muß man den gemeinsamen Speicher im TWIN-Rechner mit entsprechenden Daten in hexadezimaler Form von einem Lochstreifen oder über die Tastatur laden. Geeignete Kommandos dafür stehen innerhalb des Programms SDOS zur Verfügung.

4. Das Arbeiten mit TWIN

Im folgenden soll der Entwicklungsgang eines Mikroprozessorsystems mit TWIN-Unterstützung beschrieben werden. Ein Mikroprozessorsystem hat bekanntlich eine Struktur, wie sie Bild 12 zeigt. Die prinzipielle Wirkungsweise eines solchen Systems sei zum besseren Verständnis des nachfolgenden hier kurz beschrieben.

Die Befehle, die der Mikroprozessor ausführen soll, z.B. um einen Fahrradstuhl zu steuern, sind als sogenanntes Anwenderprogramm im Programmierspeicher aufbewahrt, der meistens als programmierbarer Festwertspeicher

(PROM) oder als maskenprogrammierter Festwertspeicher (ROM) ausgeführt ist. Der Prozessor ruft diese Befehle nacheinander durch Angeben der jeweiligen Adresse ab und bearbeitet sie. Die Daten, die der Prozessor nach Vorschrift des Anwenderprogramms zu verarbeiten hat, holt er sich durch Angabe von Adressen aus dem Datenspeicher, einem Schreib-/Lesespeicher (RAM), oder aus den ebenfalls adressierbaren Ein-/Ausgabeeinheiten und gibt sie nach der Verarbeitung wieder an diese ab. Hierbei stellen die Ein-/Ausgabeeinheiten die Verbindung zwischen Mikroprozessorsystem und Außenwelt her. Sie machen es eigentlich erst möglich, daß der Mikroprozessor eine Aufgabe wie die Fahrstuhlsteuerung ausführen kann. Über die Ein-/Ausgabeeinheiten erhält der Mikroprozessor Meldungen bzw. Befehle wie z. B. „rufe den Fahrstuhl“ oder „gewünschtes Stockwerk erreicht“, und er gibt Signale ab wie z. B. „Tür schließen“ oder „Motor hält“. Die Geschwindigkeit des Programmablaufs wird vom Takt bestimmt.

In Bild 13 ist der Entwicklungsgang eines solchen Systems dargestellt. Wie bei jeder technischen Entwicklung beginnt man damit, daß zunächst die Aufgabenstellung eindeutig formuliert wird. Hierzu gehört u. a. die Festlegung von Schnittstellen und Randbedingungen wie auch gegebenenfalls

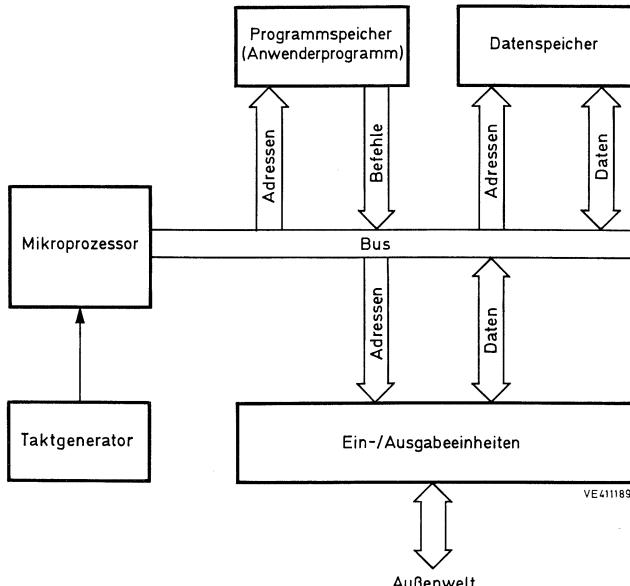


Bild 12. Vereinfachte Struktur eines Mikroprozessorsystems. Die für den Funktionsablauf erforderlichen Steuerleitungen sind nicht mit dargestellt.

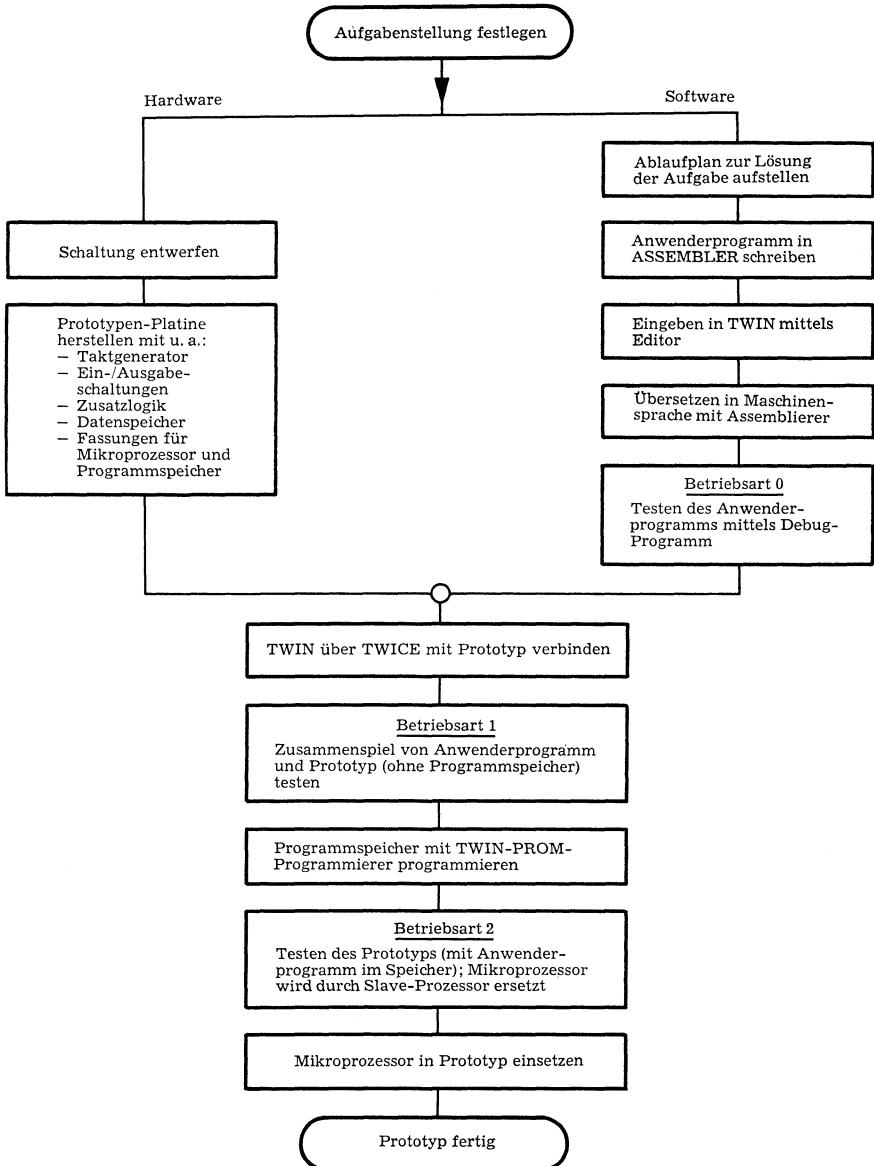


Bild 13. Entwicklung eines Mikroprozessorsystems mit Hilfe von TWIN, vereinfacht dargestellt

eine Entscheidung darüber, welche Teilprobleme besser mit Hardware oder mit Software gelöst werden können. Anschließend können Hardware- und Softwareentwicklung bis zu einem bestimmten Stadium parallel, wenn auch nicht unabhängig voneinander, vorgenommen werden.

Bei der **Hardwareentwicklung** wird eine Schaltung entworfen; sie enthält im wesentlichen den Mikroprozessor, die zugehörigen Speicher, Anpassungsschaltungen an die peripheren Schaltungen oder Geräte, den Taktgenerator und eventuell erforderliche Zusatzlogik. Das Layout für die gedruckte Schaltung wird erstellt, wobei für Mikroprozessor und Programmspeicher zunächst Steckfassungen vorzusehen sind. Eine Musterplatine wird angefertigt und geprüft.

Für die **Softwareentwicklung** (Bild 13, rechts) teilt man die Lösung des gestellten Problems in einzelne Schritte auf; letztlich so, daß jeder Schritt mit einem oder mehreren Befehlen aus dem Befehlsvorrat des Mikroprozessors realisierbar ist. Auf diese Weise entsteht ein **Ablaufplan** für die Lösung des Problems, der abhängig von Erfahrung und Intuition des Entwicklers unterschiedlich aussehen kann. Bild 14 zeigt einen Ausschnitt aus einem Ablaufplan für ein Fahrstuhl-Steuerprogramm.

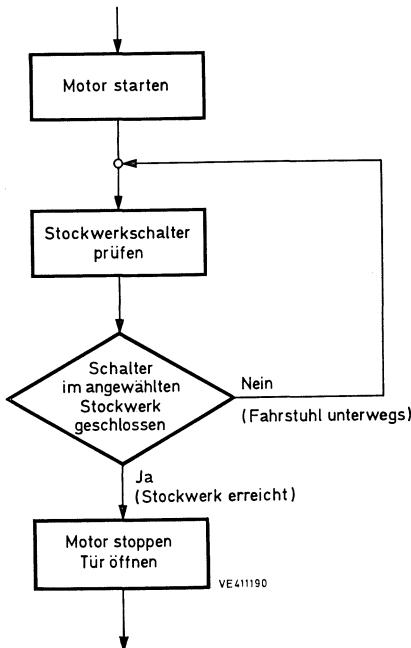


Bild 14. Ausschnitt aus einem Ablaufplan für ein Anwenderproblem (vereinfachte Fahrstuhlsteuerung)

Dieser Ablaufplan muß nun – unter Verwendung von Befehlen aus dem Befehlsvorrat des Mikroprozessors – in ein **Anwenderprogramm** umgesetzt werden, das in Assemblersprache geschrieben wird. Jetzt tritt das TWIN-System erstmalig in Aktion. Der Benutzer gibt das Programm über die Tastatur in das TWIN-System ein und kontrolliert die **Eingabe** am Datensichtgerät. Hierfür ist in Bild 15a ein einfaches Beispiel in Assemblersprache 2650 angegeben. Die Eingabe erfolgt mit Hilfe des Editors. Dieses TWIN-Programm bietet dem Benutzer zusätzlich die Möglichkeit, seinen Text zu korrigieren oder zu ändern (Bild 15b) und ihn schließlich auf einer Diskette abzuspeichern.

```
>
> SDOS VER 2.0

K > EDIT TEST; S/0

** EDIT VER 2.0 **
** NEW FILE **

K *TAB!
K *INPUT
INPUT:
P R0!EQU!0
P UN!EQU!3
P INDEV!EQU!0
P OUTDEV!EQU!1
P LOOP!REDE, R0!INDEV
P !ERTE, R0!OUTDEV
P !BCTR, UN
P !END!LOOP
```

```
K *TYPE B-E
R0    EQU    0
UN    EQU    3
INDEV  EQU    0
OUTDEV EQU    1
LOOP   REDE, R0 INDEV
        ERTE, R0 OUTDEV
        BCTR, UN
        END     LOOP

K *BEGIN
K *DOWNS
        ERTE, R0 OUTDEV
K *SUBSTITUTE /E/W/
        WRTE, R0 OUTDEV
K *FILE
        *SLJ* EOJ
```

Bild 15a.

Beispiel für das Eingeben eines Anwenderprogramms in Assemblersprache und die gleichzeitige Wiedergabe auf dem Datensichtgerät.
Die durch K gekennzeichneten, vom Benutzer eingegebenen Kommandos haben folgende Bedeutung:

rufe den Editor auf und eröffne eine Datei mit dem Namen TEST; S (Source) auf der Diskette 0,
das Zeichen ! wird als Tabulatorsymbol definiert (um den formatfrei eingegebenen Text in eine übersichtliche Darstellung (Bild 15b) umzusetzen),
rufe den INPUT-Modus auf.

Das vom Benutzer eingegebene Programm (P) enthält fünf Pseudobefehle (EQU, END) und drei 2650-Befehle.

Die weder durch K noch durch P gekennzeichneten Zeilen sind Antworten des TWIN-Systems auf die Kommandos K.

Bild 15b.

Bildschirmausgabe des Programms von Bild 15a und Textkorrektur.

Die durch K gekennzeichneten Kommandos haben folgende Bedeutung:

gib den eingegebenen Text von Beginn bis Ende aus,
gehe an den Anfang des eingegebenen Textes,
gehe fünf Zeilen abwärts und gib diese Zeile aus,
substituiere E durch W, d.h., korrigiere ERTE in WRTE,
speichere dieses Quellenprogramm in der oben angegebenen Datei.

Im nächsten Schritt muß das Anwenderprogramm aus der Assemblersprache in die Maschinensprache übersetzt werden. Hierfür steht im TWIN-System der 2650-Assemblierer zur Verfügung. Der Benutzer ruft dieses Pro-

gramm auf (Bild 15c) und läßt anschließend das **Assemblieren** ausführen. Fehler, die gegen die Regeln der Assemblersprache verstößen, werden vom Assemblierer erkannt, und die fehlerhaften Zeilen werden ausgegeben (Bild 15c). In einem weiteren Arbeitsgang kann der Benutzer diese Fehler mit Hilfe geeigneter Kommandos des Editorprogramms korrigieren (Bild 15d). Danach muß erneut assembliert werden (Bild 15e). Das nunmehr in **Maschinensprache** vorliegende Anwenderprogramm, das man sich ausgeben lassen kann (Bild 15f), wird ebenfalls auf der Diskette gespeichert, um für Test-Programmläufe bzw. später für die Programmierung von Festwertspeichern zur Verfügung zu stehen.

```
K > ASM TEST;S TEST;L TEST;H
ASSEM VER 2.0
0007 0002      S          BCTR, UN
TOTAL ASSEMBLY ERRORS = 0001
*SLJ* EOJ
```

Bild 15c. Aufruf des Assemblers und Assemblieren.

Das Kommando (K) bedeutet:

assembliere das Programm aus der Datei TEST; S. Eröffne eine Datei mit dem Namen TEST; L für die Liste des assemblierten Programms (Bild 15f) und eine Datei mit dem Namen Test; H (für das hexadezimal notierte Maschinenprogramm);

die ausgegebene Zeile 7 enthält als ersten Fehler einen syntaktischen Fehler, gekennzeichnet durch S,
Gesamtanzahl der Fehler = 1.

```
K > EDIT TEST;S
*** EDIT VER 2.0 ***
K *GET 99
** EOF ***
K *BEGIN
K *FIND $BCTR#
          BCTR, UN
K *REPLACE           BCTR, UN LOOP
          BCTR, UN LOOP
K *FILE
** EOF ***
*SLJ* EOJ
```

Bild 15d.

Korrektur der beim Assemblieren aufgetauchten Fehler.

Die Kommandos (K) bedeuten:

rufe den Editor und die Datei TEST; S auf,
lade 99 Zeilen dieser Datei in den gemeinsamen Speicher,
gehe an den Anfang der Datei,
suche die erste Zeile mit der Folge BCTR,
ersetze diese Zeile durch den folgenden Text,
speichere diesen Text in der angegebenen Datei.

```
K > ASM TEST;S TEST;L TEST;H
ASSEM VER 2.0
TOTAL ASSEMBLY ERRORS = 0000
*SLJ* EOJ
```

Bild 15e.

Assemblieren des korrigierten Quelltextes (vgl. Bild 15c)

```
K > PRINT TEST;L CONO
TWIN ASSEMBLER VER 2.0

LINE ADDR OBJECT E SOURCE

0001 0000      R0    EOU    0
0002 0003      UN    EOU    3
0003 0000      INDEV EOU    0
0004 0001      OUTDEV EOU    1
0005 0000 5400   LOOP   REDE,R0 INDEV
0006 0002 D401   WRTA,R0 OUTDEV
0007 0004 1B7A   BCTR,UN LOOP
0008 0000      END    LOOP
TOTAL ASSEMBLY ERRORS = 0000
*PRN+ EOJ
```

Bild 15f. Ausdruck nach der Assemblierung (sogenannte Liste).

Das Kommando (K) bedeutet:

schreibe den Inhalt der Datei TEST; L auf der Konsole (CONsOLE) bzw. dem Datensichtgerät.

Rechts unter SOURCE das Quellenprogramm (vgl. Bild 15b); Maschinenprogramm unter OBJECT in Hexadezimalziffern; in Spalte E wird die Art eines eventuellen syntaktischen Fehlers durch einen Buchstaben gekennzeichnet (vgl. Bild 15c).

```
K > RHEX TEST;H
*RHXX* EOJ
```

```
K > DEBUG
```

```
K > TRACE ALL
```

```
K > BKPT 4
```

```
K > GO 0
LOC INST MNEMON XR U OPAD IADD IV EADD R0 R1 R2 R3 R4 R5 R6 PU PL
0000 5400 RDE ,00    00          00 01 77 00 11 0F 13 00 00
0002 D401 WTE ,00    01          00 01 77 00 11 0F 13 00 00
0004 1B7A BTR ,03    - 0000      =0000 00 01 77 00 11 0F 13 00 00
0004 BREAK
```

```
K > ABORT DEBUG
```

Bild 15g. Testen in Betriebsart 0 mit Debug-Programm.

Die Kommandos (K) bedeuten:

lade den Inhalt der Datei TEST; H in den gemeinsamen Speicher,
rufe das DEBUG-Programm auf,
verfolge die Ausführung aller Anwender-Befehle,
unterbrich (Break Point) bei Adresse (LOCATION) 4,
starte das Programm bei Adresse 0,
beende das DEBUG-Programm.

Die Spalten R0 bis PL stellen den jeweiligen Inhalt der Register des Slave-Mikroprozessors dar, der den Prozessor des Prototyps ersetzt.

Das Anwenderprogramm kann jetzt – zunächst noch ohne Hardware – in der sogenannten **Betriebsart 0** getestet werden. Unter Benutzung des Debug-Programms lässt man das Anwenderprogramm vom Slave-Prozessor des TWIN-Systems abarbeiten, wobei verschiedene Möglichkeiten bestehen, die Funktion der endgültigen Ein-/Ausgabeeinheiten nachzubilden. Eine dieser Möglichkeiten ist in Bild 16 dargestellt. Mit Hilfe des Debug-Programms lässt sich der Zustand des Slave-Prozessors, d. h. der Inhalt aller seiner Register, verfolgen und der Probelauf an gewünschter Stelle im Programm unterbrechen (Bild 15g). Auf diese Weise kann der Benutzer seine Programmierfehler ausfindig machen und korrigieren. Die Betriebsart 0 wird also angewandt, um die Software möglichst weit auszutesten, bevor man sie zusammen mit der Hardware (Prototyp) in Betrieb nimmt.

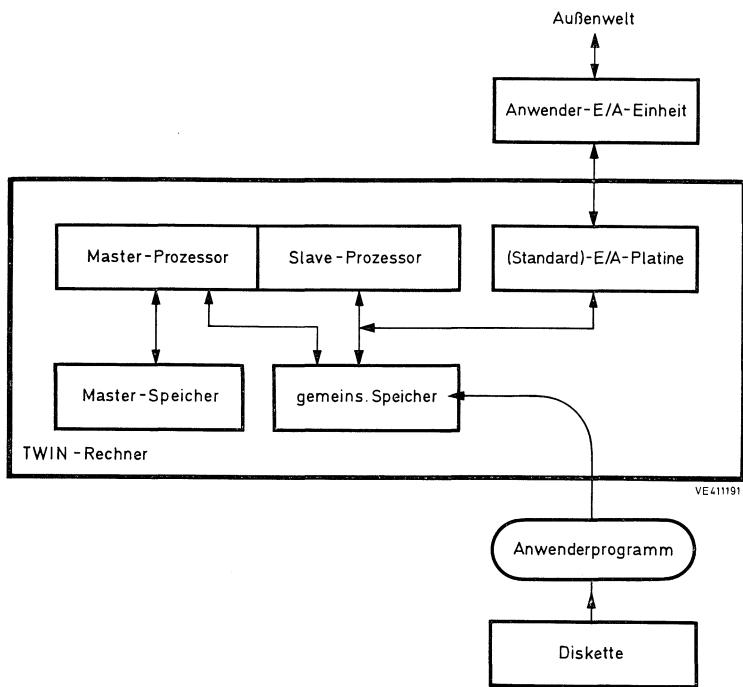


Bild 16. TWIN-System in Betriebsart 0.

Die Funktion der Anwender-Ein-/Ausgabeeinheiten kann hier mit Hilfe einer Ein-/Ausgabe-Platine (z. B. TWIN-Standard-E/A-Platine) mit einbezogen werden.

Im nächsten Entwicklungsschritt bezieht man, wie Bild 13 zeigt, den Prototyp mit in das Testsystem ein, um die Integration von Software und Hardware vorzunehmen. Die Integration erfolgt in zwei Stufen – Betriebsart 1 und Betriebsart 2.

Zur Einbeziehung der Hardware in die Testphasen dient das **TWICE-Kabel**. Der TWICE-Stecker ist anschlußkompatibel zu den 40 Anschlüssen des Mikroprozessors 2650 und wird in die 2650-Fassung auf der Prototypen-Platine des zu entwickelnden Anwendersystems gesteckt (Bild 3). TWIN-System und Anwendersystem sind somit über das TWICE-Kabel verbunden. Der Slave-Prozessor ersetzt den Prozessor des Anwendersystems und wird vom Taktgenerator des Anwendersystems gesteuert.

Bei der **Betriebsart 1**, die in Bild 17 dargestellt ist, befindet sich das Anwendungsprogramm im gemeinsamen Speicher von TWIN, und man testet mit Hilfe von TWICE und dem Taktgenerator des Prototyps das Zusammenspiel mit den Ein-/Ausgabeeinheiten. Dabei können – wie in der Betriebsart 0 – mit geeigneten Kommandos des Debug-Programms das Anwendungsprogramm und der Zustand des Slave-Prozessors verfolgt werden. Hier werden Fehler, die entweder im Programm infolge der Hinzunahme der Hardware erkennbar werden oder aber in den Ein-/Ausgabeschaltungen selbst liegen, beseitigt. Die Betriebsart 1 wird mit am häufigsten benutzt. Sie bietet die Möglichkeit, Anwendungsprogramm und Anwenderalternative optimal aufeinander abzustimmen. Nach Abschluß dieser Test-Stufe hat man den Teil der Entwicklungsprobleme, die

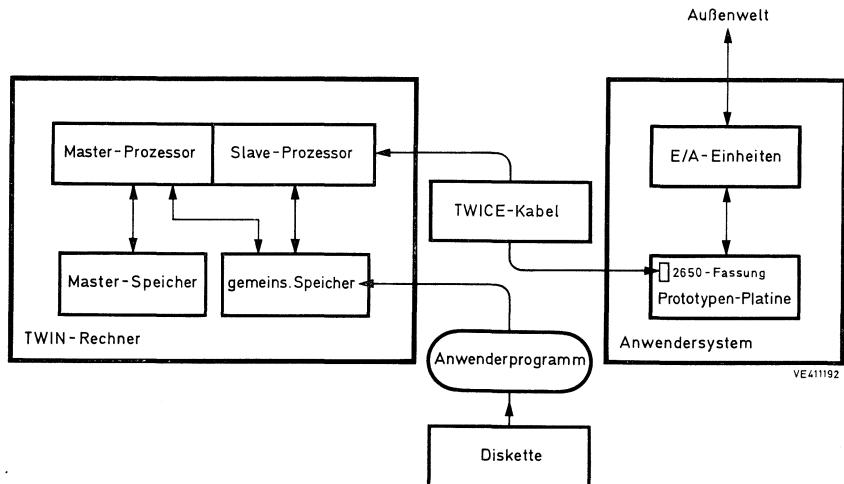


Bild 17. TWIN-System in Betriebsart 1

mit der Ein-/Ausgabe und der Software zusammenhängen, weitgehend überwunden.

Die Entwicklung des Anwenderprogramms ist damit abgeschlossen, so daß man es in den Programmspeicher des Prototyps, in der Regel ein PROM oder ROM, laden kann (vgl. Bild 13). Mit TWIN lassen sich sowohl Bipolar- als auch MOS-PROMs direkt mit dem fertigen Anwenderprogramm programmieren oder aber auch Daten in einem Standard-Format ausgeben, nach denen der Halbleiterhersteller maskenprogrammierte ROMs herstellen kann. Der Programm speicher mit dem Anwenderprogramm kann anschließend in die Prototypen Platine eingesetzt werden.

Nun folgt der letzte der in Bild 13 aufgeführten Entwicklungsschritte. Er wird in der **Betriebsart 2**, erläutert durch Bild 18, vorgenommen. In dieser Betriebsart testet man das komplette Anwendersystem, bestehend aus Takt-Generator, Ein-/Ausgabeeinheiten und Speicher (Programm und Daten). Lediglich der Mikroprozessor 2650 wird wieder über das TWICE-Kabel vom Slave-Prozessor vertreten, der jedoch, wie erwähnt, den Takt des Anwendersystems benutzt.

Ähnlich wie bei Betriebsart 0 und 1 können der Programmablauf und der Zustand des Mikroprozessors mit Hilfe des Debug-Programms verfolgt werden (Bild 15g). Die Betriebsart 2 benutzt man hauptsächlich zur Prüfung des Speichers und des endgültigen Anwendersystems, aber auch zur wirkungsvollen Fehlersuche, wenn Schwierigkeiten in der Produktion rasch beseitigt werden sollen.

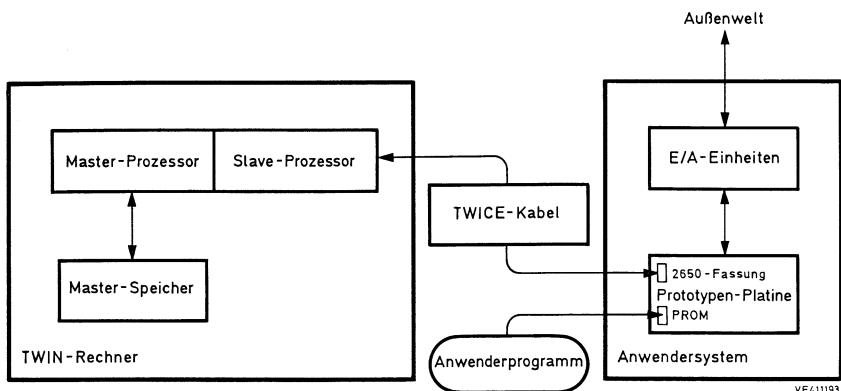


Bild 18. TWIN-System in Betriebsart 2

Damit ist der Einsatz von TWIN bei der Entwicklung eines Mikroprozessorsystems im wesentlichen beschrieben. Einzelheiten müssen den Arbeitsanleitungen zum TWIN-System entnommen werden. Folgende Hauptpunkte seien zum Schluß dieses Abschnitts noch einmal herausgestellt:

- TWIN ermöglicht eine optimale Lösung des Problems der Software/Hardware-Integration bei der Entwicklung von Mikroprozessorsystemen.
- TWIN hilft sowohl dem Software- als auch dem Hardware-Entwickler, die Entwicklung zu rationalisieren und damit Zeit und Kosten zu sparen.
- Programmieranfänger können schon nach kurzer Einarbeitungszeit alle Vorteile von TWIN bei der Entwicklung größerer Programme nutzen.
- Zusätzlich zu den genannten Punkten legen erfahrene Programmierer bei der Entwicklung längerer Programme vor allem Wert auf folgende Besonderheiten von TWIN:
 - Mit Hilfe der Standard-Ein-/Ausgabe-Platine (Aufstellung Abschn. 2.2.8) können Anwenderprogramme in der Betriebsart 0 in begrenztem Maße bereits auch ohne die endgültigen Ein-/Ausgabeschaltungen getestet werden.
 - Das Debug-Programm bietet die Möglichkeit, Ein-/Ausgabebefehle des Anwenderprogramms auch ohne angeschlossene Anwender-Hardware zu prüfen, indem man die betreffenden Daten über die peripheren TWIN-Geräte aus- oder eingibt (z. B. Datensichtgerät und Tastatur).

Anhang

Wichtige Benutzer-Kommandos beim TWIN-System

Zur Vereinfachung beim Eingeben sind für die meisten Kommandos Kurzformen zugelassen, z. B. V statt VERIFY, D statt DUP, PR statt PRINT usw.

A.1. SDOS-Kommandos

A.1.1. Dienst-Kommandos

Für die Verwaltung von Disketten und Dateien sowie für den Datenverkehr innerhalb des TWIN-Systems gibt es folgende Kommandos

FORMAT	formatiert eine Diskette Vorbereiten (Initialisieren) einer Diskette für die Benutzung im TWIN-System (Abschn. 2.3)
VERIFY	verifiziert eine Diskette Ermitteln und Katalogisieren unbrauchbarer Blöcke auf der Diskette
RENAME	ändert den Namen einer Datei oder einer Diskette
DUP	dupliziert eine Diskette kopiert den Inhalt einer Diskette auf eine zweite Diskette in einem anderen Laufwerk
LDIR	(List DIRectory) gibt das Dateien-Verzeichnis (Abschn. 2.3) der Diskette aus
DELETE	löscht eine Datei auf der Diskette
COPY	überträgt Daten von einem Eingabegerät oder einer Datei auf ein Ausgabegerät oder eine andere Datei
PRINT	gibt eine vorzugebende Zeilenzahl einer Diskettendatei auf einem bestimmten Ausgabegerät aus

A.1.2. System-Kommandos

Mit den folgenden Kommandos an das Betriebssystem SDOS kann der TWIN-Benutzer die durch SDOS gegebenen Möglichkeiten variieren bzw. erweitern

SYSTEM	kennzeichnet dasjenige Disketten-Laufwerk, auf dem sich das zu benutzende SDOS-Programm befindet
DEVICE	informiert das Betriebssystem SDOS über die Verfügbarkeit peripherer Geräte
SEARCH	veranlaßt das automatische Suchen von Dateien auf allen Disketten

A.1.3. Steuer-Kommandos

Der TWIN-Benutzer kann System- oder Anwenderprogramme durch Drücken folgender Tasten steuern

(ESC)	(ESCAPE) unterbricht die Ausführung eines Programms
(ESC)(ESC)	(zweimaliges Drücken der Taste (ESC)) unterbricht die Ausführung des Anwenderprogramms und des überwachenden Systemprogramms
(Leertaste)	stoppt oder startet die Ausgabe auf dem Datensichtgerät

Ferner kann der Benutzer die Ausführung von Programmen sowie die Steuerung von Slave-Kanälen durch folgende Kommandos beeinflussen

SUSPEND	unterbricht die Ausführung eines Programms (vorwiegend bei Verwendung von Kommando-Dateien)
CONT	(CONTINUE) setzt die Ausführung eines (durch SUSPEND oder (ESC)(ESC)) unterbrochenen Programms fort
ABORT	beendet ein aktives Programm
ASSIGN	ordnet einem Slave-Kanal ein bestimmtes Gerät oder eine Datei zu
CLOSE	hebt die durch ein vorheriges ASSIGN-Kommando hergestellten Zuordnungen wieder auf

A.1.4. Debug-Kommandos

Das Debug-Programm ist zur Fehlersuche und Fehlerbeseitigung vorgesehen. Seine wichtigsten Kommandos sind hier aufgezählt. Darüber hinaus stehen während des Debug-Betriebs auch einige der unter A.1.1 und A.1.3 genannten SDOS-Kommandos zur Verfügung

DEBUG	das Debug-Programm wird von der Diskette in den Master-Speicher geladen
LOAD	lädt ein Anwenderprogramm (durch MODULE erzeugt, A.1.5.) in den gemeinsamen Speicher
GO	startet das Anwenderprogramm
XEQ	(eXEcute) lädt und startet das Anwenderprogramm (LOAD und GO)
BKPT	(BreaK PoinT) legt einen Programm-Haltepunkt fest
CLBP	(CLear Break Point) löscht einen vorher festgelegten Haltepunkt
TRACE	veranlaßt ein schrittweises Verfolgen des Programmablaufs
DUMP	gibt Inhalt des gemeinsamen Speichers aus

PATCH	ändert den Inhalt des gemeinsamen Speichers von einer vorzugebenden Adresse an
EXAM	(EXAMine) gibt den Inhalt eines Speicherplatzes im gemeinsamen Speicher für eine vorzugebende Adresse an
DSTAT	(Display STATus) gibt die Adresse des zuletzt ausgeführten Befehls, den Inhalt der Slave-Prozessor-Register und die Haltepunkt-Adressen aus
SET	ändert den Inhalt von Registern im Slave-Prozessor
RESET	setzt den Slave-Prozessor in Anfangsstellung

A.1.5. Dienst-Kommandos für Objektprogramme

Für Speicherung, Übertragung und Modifikation von Objektprogrammen stehen dem Benutzer folgende Kommandos zur Verfügung

RHEX	(Read HEXadecimal) lädt ein hexadezimal ¹⁾ geschriebenes Objektprogramm in den gemeinsamen Speicher
WHEX	(Write HEXadecimal) überträgt ein hexadezimal ¹⁾ geschriebenes Objektprogramm vom gemeinsamen Speicher zu einem Ausgabegerät oder einer Datei
WSMS	gibt einen 512-Byte-Block aus dem gemeinsamen Speicher im SMS-Format aus (Signetics Memory Standard)
CSMS	liest eine Datei im SMS-Format und vergleicht sie mit dem Inhalt des gemeinsamen Speichers
MODULE	erzeugt aus einem Objektprogramm, das im gemeinsamen Speicher hexadezimal ¹⁾ abgelegt ist, ein binär, d. h. in Maschinensprache geschriebenes Programm, einen sogenannten Lade-Modul, und speichert ihn in einer Datei ab

A.1.6. Kommandos für Editor, Assemblierer und PROM-Programmierer

EDIT	ruft das Editorprogramm auf
ASM	ruft das Assemblerprogramm auf und bringt es zur Ausführung
RROM	(Read) liest den Inhalt eines PROMs und speichert ihn im gemeinsamen Speicher ab
CPROM	(Compare) liest den Inhalt eines PROMs und vergleicht ihn mit dem Inhalt des gemeinsamen Speichers
WPROM	(Write) programmiert ein PROM mit dem Inhalt des gemeinsamen Speichers

¹⁾ codiert im ASCII

A.2. Editor-Kommandos

Für das Niederschreiben, Abspeichern, Ändern und Ausgeben von Quellenprogrammen hält der EDITOR folgende Kommandos bereit

A.2.1. Eingeben, Ändern, Löschen

INPUT	ruft den INPUT-Modus auf, der für das Eingeben von Quellenprogrammen vorgesehen ist
INSERT	fügt eine vorgebbare Textzeile ein
SUBSTITUTE	ersetzt eine Textstelle innerhalb einer Zeile
REPLACE	ersetzt eine ganze Zeile
KILL	löscht eine vorgebbare Anzahl von Textzeilen

A.2.2. Ein-/Ausgabe

GET	liest eine vorzugebende Anzahl von Zeilen von einer Datei in den Pufferspeicher
PUT	schreibt eine vorzugebende Anzahl von Zeilen aus dem Pufferspeicher in eine Datei
COPY	überträgt eine vorzugebende Anzahl von Zeilen von einer Datei auf eine andere Datei
LIST	druckt eine vorzugebende Anzahl von Zeilen auf dem Zeilendrucker
TYPE	gibt eine vorzugebende Anzahl von Zeilen auf dem Datensichtgerät aus

A.2.3. Zeilenzeiger

BEGIN	setzt den Zeilenzeiger auf die erste Zeile des Pufferspeichers
END	setzt den Zeilenzeiger auf das Ende des Textes im Pufferspeicher
FIND	führt den Zeilenzeiger zu der Zeile, bei der eine vorzugebende Zeichenfolge erstmalig vorkommt
DOWN	führt den Zeilenzeiger um eine vorzugebende Zeilenanzahl abwärts
UP	führt den Zeilenzeiger um eine vorzugebende Zeilenanzahl aufwärts
N	gibt die Zeilenzeiger-Position (Zeilennummer) auf dem Datensichtgerät aus

A.2.4. Editor-Hilfen

AGAIN	wiederholt die Ausführung des zuvor gegebenen Kommandos
BRIEF	unterdrückt die Ausgabe der vom Zeilenzeiger gekennzeichneten Zeile auf dem Datensichtgerät, ohne die Ausführung des Zeilenzeiger-Kommandos zu beeinträchtigen
FILE	überträgt den Text vom Pufferspeicher in eine Datei (Diskette) und beendet den EDIT-Modus (SDOS übernimmt die Steuerung)
QUIT	beendet den EDIT-Modus, ohne Text auf die Diskette zu übertragen
TAB	definiert ein vorzugebendes Zeichen als Tabulator-Symbol
TABS	setzt den Tabulator auf vorzugebende Spalten
MACRO	erzeugt ein Makro-Kommando, bestehend aus einer Folge mehrerer Kommandos
MACRO m	ruft den Makrobefehl mit der Nummer m auf
m <>	veranlaßt eine mehrfach wiederholte Ausführung der in <> angegebenen Kommandofolge
?	zeigt den Ein-/Ausgabezustand des Editors auf dem Datensichtgerät an
SDOS	setzt das Editor-Programm außer Funktion und gibt die Steuerung an das Betriebssystem SDOS zurück

