

## Typische Entwicklungswerkzeuge

Die folgenden Darstellungen sind vor allem gedacht

- als Anregung zum Auswählen und Beschaffen,
- als Anregung, die jeweiligen Prinzipien gleichsam zu Fuß auszunutzen (Grundgedanken aufgreifen und z. B. mit C oder Assembler auf eigene Faust implementieren).

### 1. Programmentwicklung über Flußdiagramme (Flowcharts)

Einschlägige Systeme gibt es in verschiedenen Preis- und Leistungsklassen. Das Flußdiagramm ist ein bewährtes Ausdrucksmittel zum Erfassen und Veranschaulichen von Abläufen. Es ist aber kaum geeignet, sämtliche Einzelheiten exakt darzustellen (Arbeitsaufwand, Umfang, Überschaubarkeit). Deshalb Arbeitsteilung:

- die grundsätzlichen Abläufe werden als Flußdiagramm dargestellt,
- die Funktionen der einzelnen Blöcke werden auf herkömmliche Weise ausprogrammiert,
- das Entwicklungssystem kümmert sich darum, daß alles zusammenpaßt. Es erzeugt aus der Ablaufdarstellung und den einzelnen Programmstücken den vollständigen Quellcode.

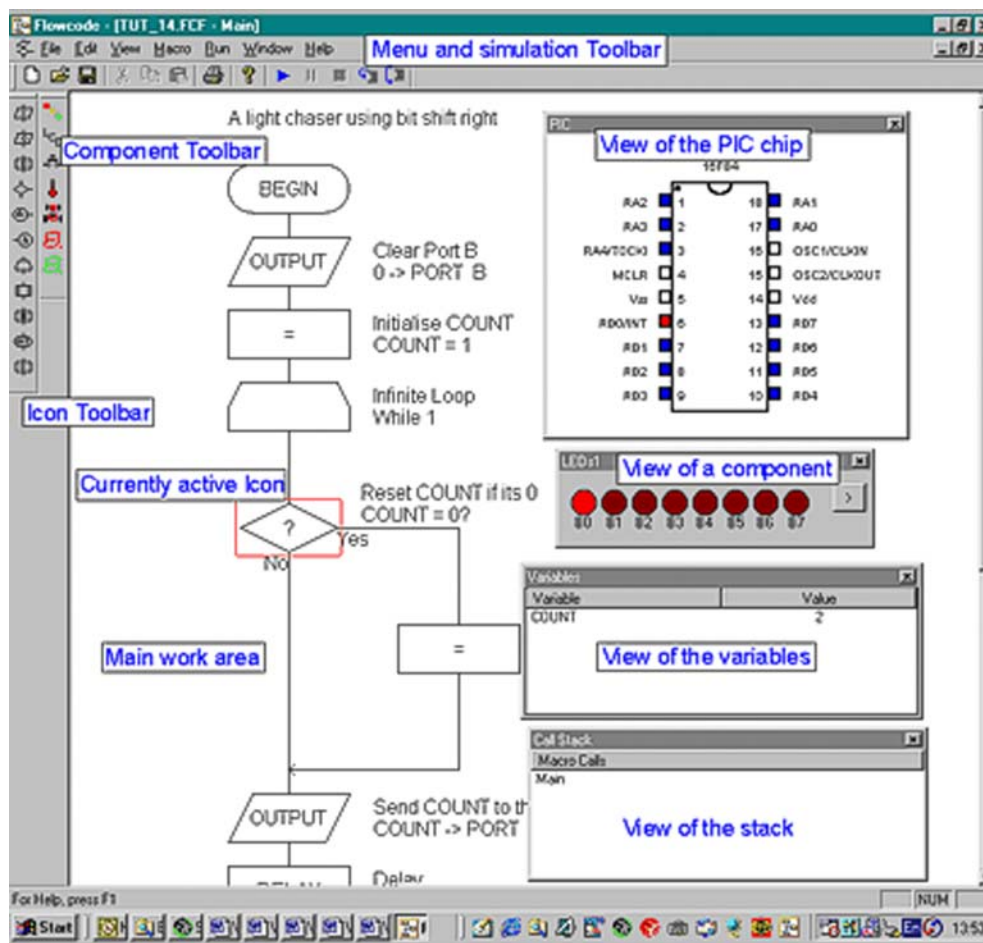


Abb. 1 Flowcode (Matrix Multimedia). Dieses System ist vor allem zu Lehrzwecken vorgesehen

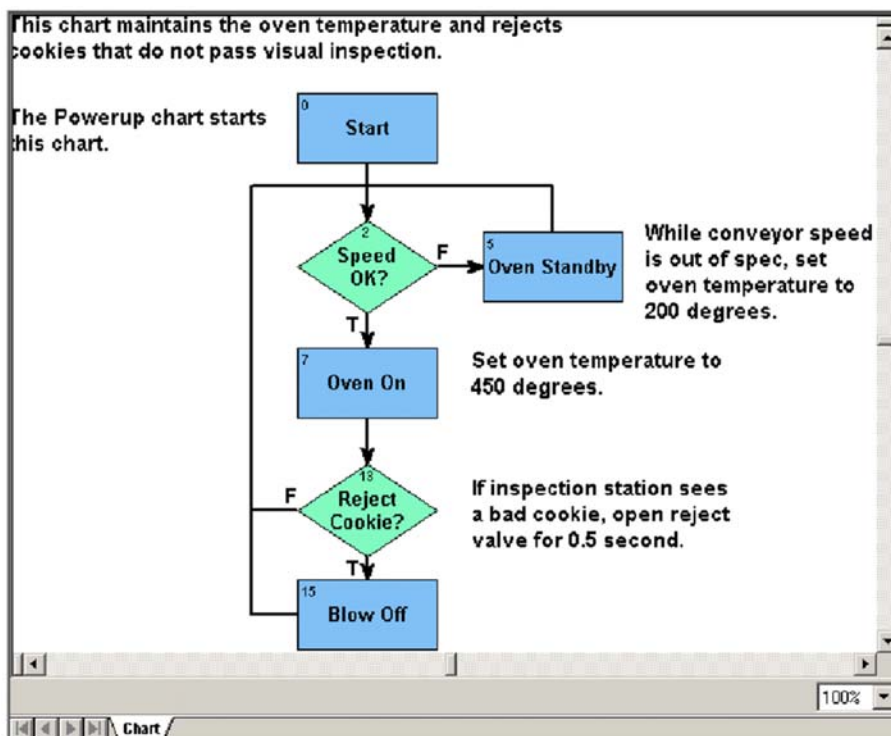


Abb. 2 ioControl (Opto 22). Ein professionelles Entwicklungssystem

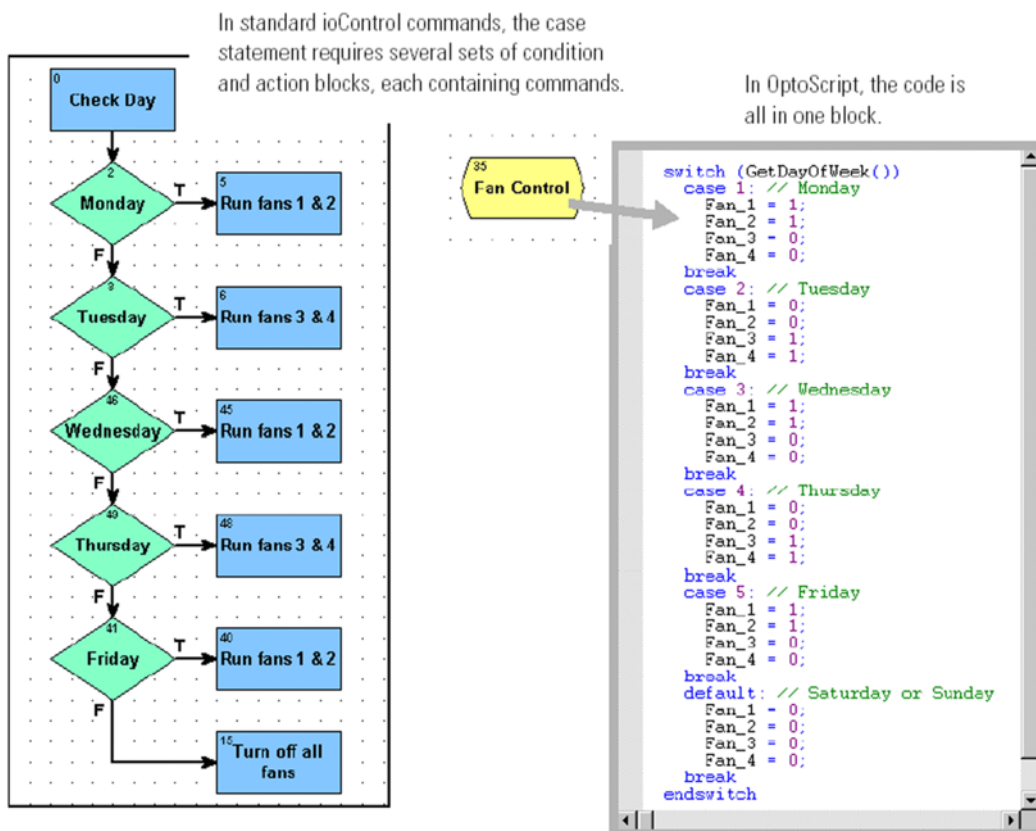
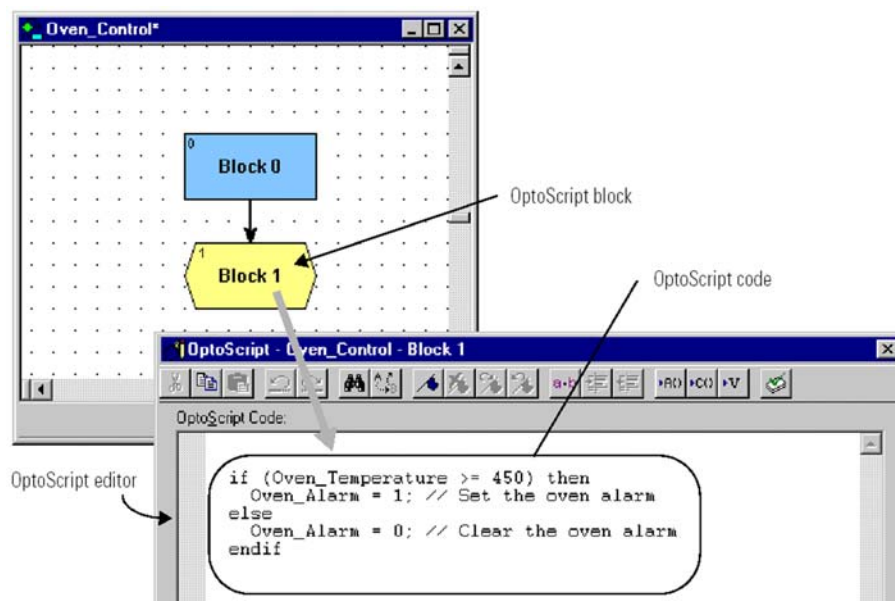


Abb. 3 ioCONTROL unterstützt den Wechsel zwischen Flußdiagramm und Programmiersprache (Opto 22)



**Abb. 4** Was leistet ein Funktionsblock? Oben Diagrammdarstellung, darunter die ausprogrammierte Funktion (Opto 22)

## 2. Programmentwicklung mit Zustandsdiagrammen (State Charts)

Für viele Steuerungsaufgaben liegt es nahe, die Begriffsbildungen und Modellvorstellungen der Automatentheorie auszunutzen. Das einfachste Modell – der Zustandsautomat (State Machine) – wird aber unhandlich, wenn wirklich alle Einzelheiten exakt zu beschreiben sind (das gilt bereits für Vorhaben, die ausgesprochen harmlos aussehen). Die Auswege:

- mehrere Zustandsautomaten im Verbund,
- die Zustände werden um weitere Funktionen angereichert (die z. B. beim Eintreten in den Zustand oder beim Verlassen des Zustandes wirksam werden).

Ein entsprechender Industriestandard: UML = Unified Modeling Language. Diese Sprache wird von verschiedenen sog. Codegeneratoren unterstützt.

Der typische Entwicklungsgang:

1. Erfassung der Entwurfsabsicht im Form von Zustandsdiagrammen (State Charts). Zustände und Zustandsübergänge werden graphisch dargestellt, Übergangsbedingungen in Form von Booleschen Gleichungen erfaßt.
2. Das System erzeugt daraus einen C-Quellcode.
3. Der Quellcode wird für den jeweiligen Mikrocontroller übersetzt.

Muß man sich nicht unbedingt an UML halten, kann das Verfahren abgewandelt werden:

- vereinfachte Zustandsmodelle,
- Nutzung weiterer Ansätze aus der Theorie (z. B. Parallelautomat nach Zakrevskij),
- andere (= effektivere) Formen der Codierung (vor allem in Hinsicht auf die Booleschen Gleichungen, die die Zustandsübergänge beschreiben).

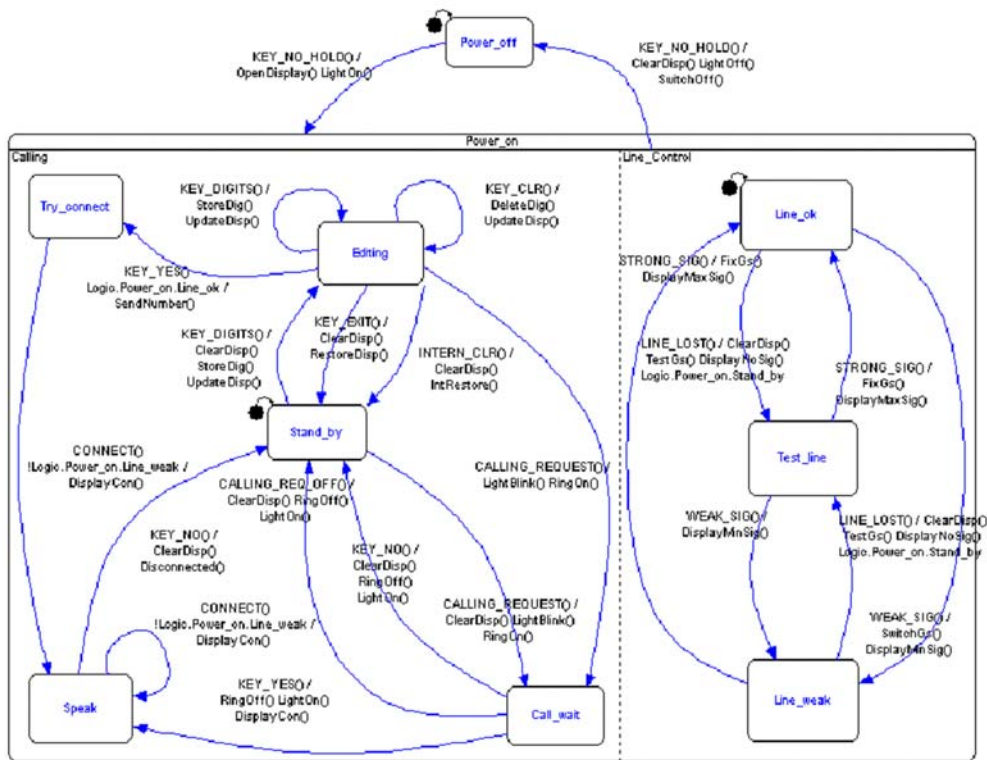


Abb. 5 UML State Chart in der Entwicklungsumgebung (Code Generator) IAR Visual State (IAR)

### 3. Programmentwicklung mit Schaltplänen

Die Entwurfsabsicht wird nicht als Programm erfasst, sondern als (fiktive) Hardware (Blockschaltbild, Schaltplan). Das Entwicklungssystem setzt das Schaltbild in einen Quellcode um. Im Grunde handelt es sich um eine (funktionelle) Simulation der fiktiven Hardware.

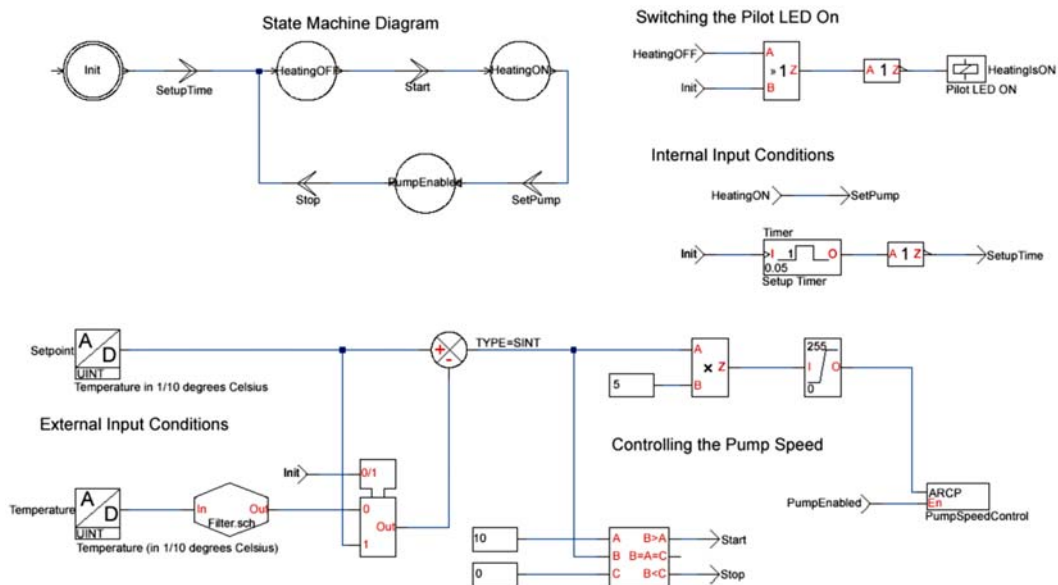
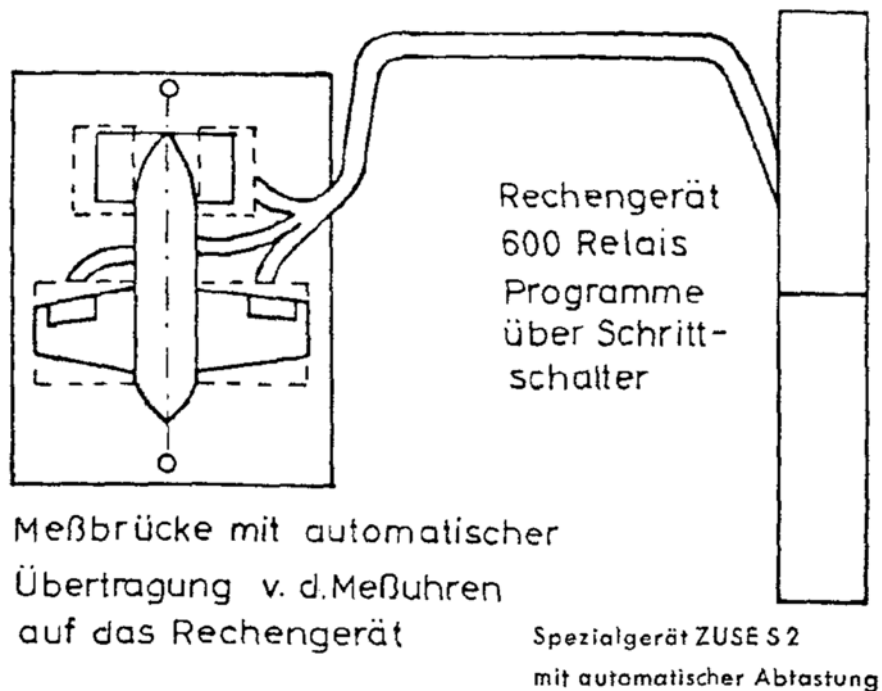


Abb. 6 Anwendungsentwicklung mit Schaltplänen. Beispiel: Realizer (Actum Solutions)

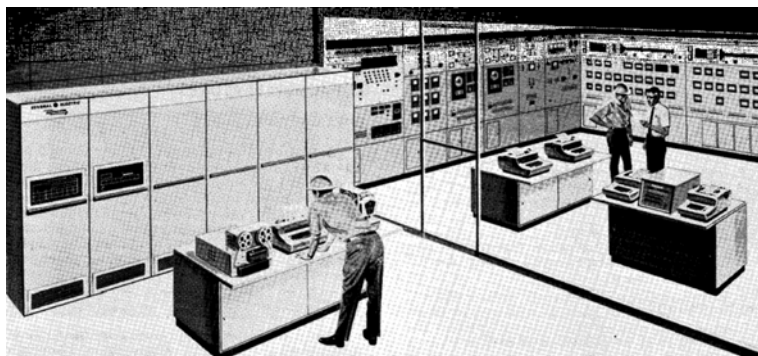
## Zur Geschichte des Prozeßrechners

### 1. Die ersten Prozeßrechner der Welt: Zuse S1 / S2.

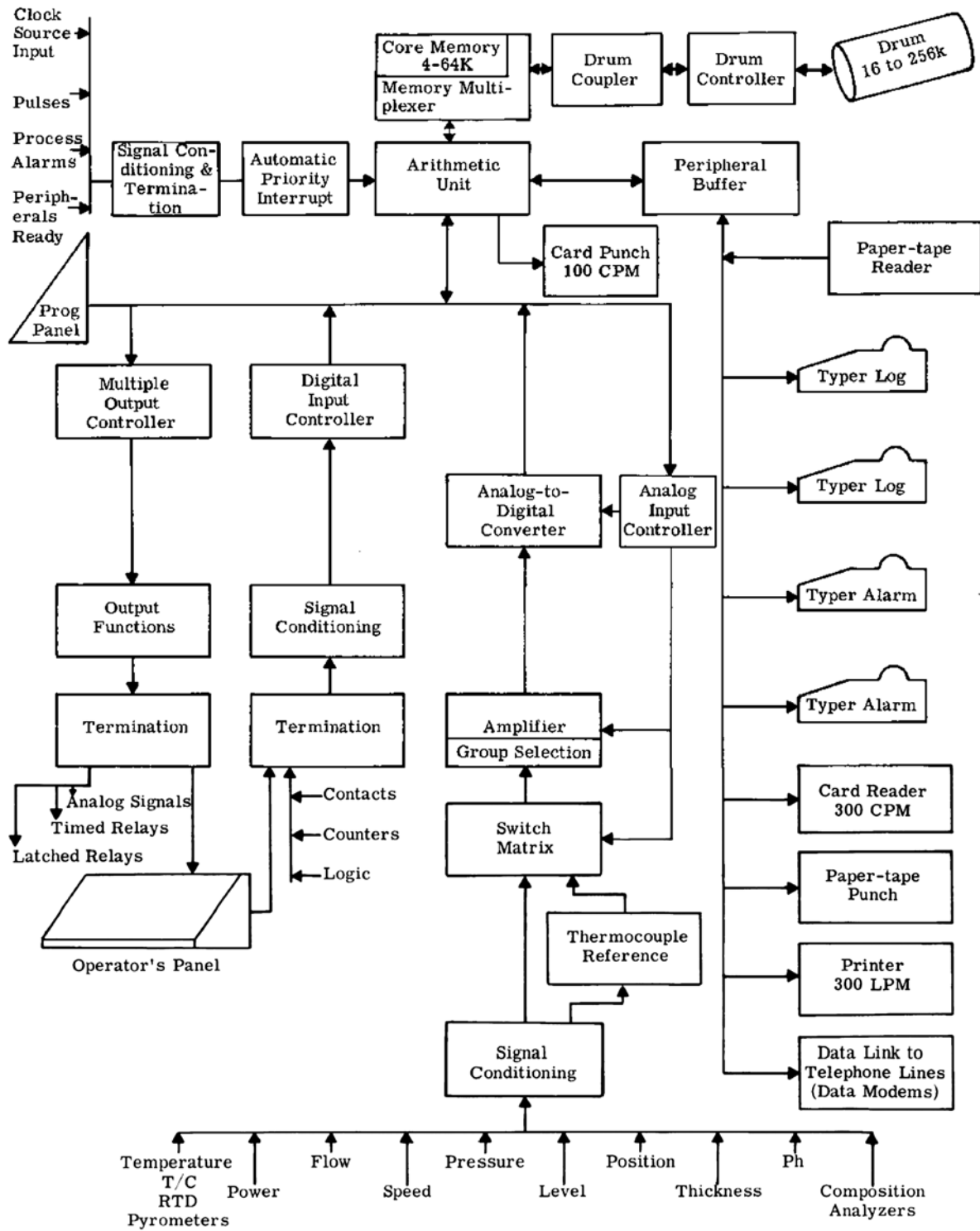
- Anwendung: Vermessung der Bauungenauigkeiten von Flugkörpern und Errechnen von Korrekturdaten (z. B. Einstellwinkel).
- Datenerfassung: über mehr als 100 Meßuhren.
- Eingabe: S1: von Hand (off line), S2: automatisch (on line).
- Keine direkte Rückwirkung auf den Fertigungsprozeß (open loop).
- Zeitraster: gemäß Takt des Fertigungsbandes (Richtwert: 1 Flugkörper alle 10 Minuten).



Solche Aufgaben der Prozeßführung (Werte erfassen – rechnen – Einstellungen vornehmen) entsprechen direkt dem Prinzip der EDV (Eingabe – Verarbeitung – Ausgabe). Parallele Abläufe gibt es nicht; alles wird nacheinander erledigt. Deshalb kann auch mit vergleichsweise bescheidener Hardware Beachtliches geleistet werden.



Zwei klassische Prozeßrechner (General Electric, Univac)



Ein klassischer Prozeßrechner im Blockschaltbild (General Electric)