

Zeitschrift:	Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses
Herausgeber:	Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen
Band:	78 (1987)
Heft:	11
Artikel:	Entwurf eines 32-Bit-Mikrocontrollers
Autor:	Eicken, T. von
DOI:	https://doi.org/10.5169/seals-903870

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 26.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Entwurf eines 32-Bit-Mikrocontrollers

T. von Eicken

An den Instituten für Integrierte Systeme und für Informatik der ETH Zürich wurde kürzlich ein 32-Bit-Mikrocontroller als integrierte Schaltung entworfen. Der vorliegende Artikel fasst die wesentlichen Punkte dieses Designprojektes zusammen. Die beim Entwurf gewonnenen Erfahrungen zeigen, dass ein VLSI-Projekt nur durch konsequente Anwendung hierarchischer Designmethoden gelingen kann und entscheidend vom Vorhandensein und der Qualität von CAD-Programmen abhängig ist.

Récemment un microcontrôleur 32-bit a été réalisé en circuit intégré aux Instituts de Systèmes Intégrés et d'Informatique à l'EPF Zurich. Cet article décrit les points essentiels de cette réalisation. Les expériences faites montrent qu'un projet VLSI ne peut aboutir qu'en appliquant des méthodes hiérarchiques de conception et qu'il dépend en grande mesure de la présence et qualité de programmes CAO.

Diese Arbeit wurde vom Autor im Rahmen einer grösseren Studienarbeit am Institut für Integrierte Systeme der ETH Zürich verfasst.

Adresse des Autors

T. von Eicken, cand. Inf.-Ing. ETH, Institut für Integrierte Systeme, ETH-Zentrum, 8092 Zürich.

1. Einführung

Bereits vor einigen Jahren wurde von Prof. Wirth am Institut für Informatik eine Vorstudie über einen einfachen Prozessor durchgeführt. Die Architektur dieses *FlintStone* getauften Prozessors wurde auf Grafikoperationen, die in modernen Arbeitsplatzrechnern benötigt werden, zugeschnitten. Geplant ist der Einsatz als Koprozessor im Arbeitsplatzrechner Ceres [1], der vor kurzem am Institut für Informatik entwickelt wurde.

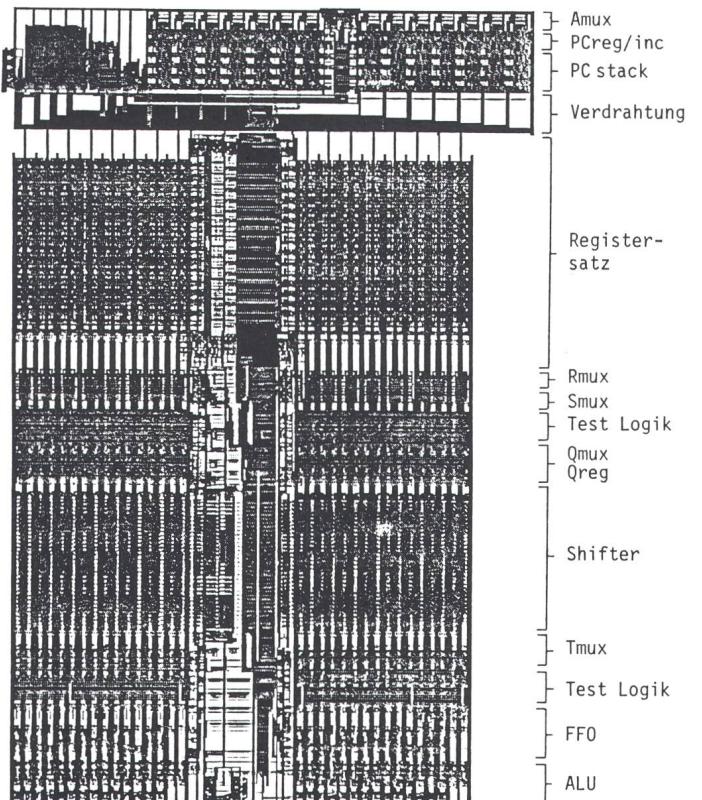
Der Entwurf einer komplexen integrierten Schaltung ist, wie in [2] bereits beschrieben, ein äusserst kostspieliger und zeitaufwendiger Vorgang. Aufgrund ihrer Einfachheit eignet sich die

FlintStone-Architektur sehr gut für eine Realisierung als integrierte Schaltung.

Der vorliegende Artikel beschreibt die wesentlichen Punkte der Realisierung dieses Prozessors von der Architektur zum Layout. Das Projekt hatte drei Ziele:

1. Aufbau von Know-how für den Entwurf komplexer integrierter Schaltungen,
2. kritische Evaluation der Qualität der vorhandenen Hard- und Software,
3. Generierung eines Core-Designs für weitere, zukünftige Designprojekte.

Figur 1
Vollständiger Flintstone-Layout

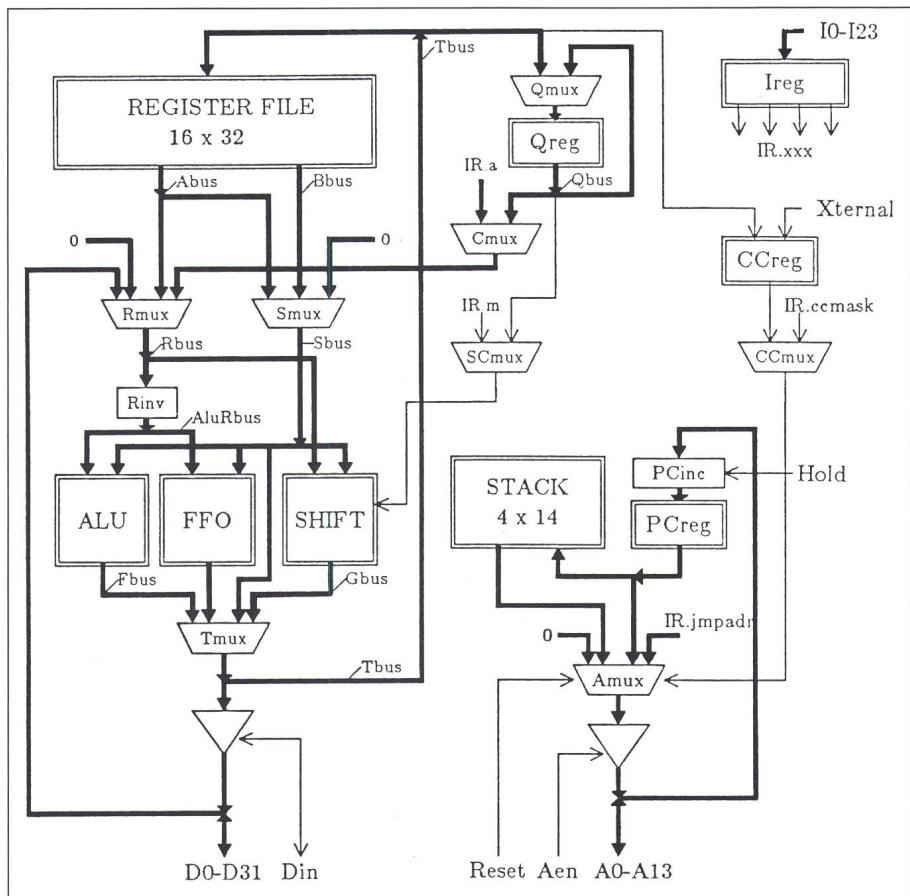


2. Architektur

Leitend bei der Definition der FlintStone-Architektur (Fig. 1) war die Idee, die Hauptfunktionen eines 32-Bit-Mikroprozessors auf einen IC zu integrieren, ohne allerdings den Ablauf der Operationen fest zu definieren¹. Um dies zu erreichen, wurde eine mikroprogrammierbare Architektur gewählt. Der Datenfluss durch die einzelnen Teile des IC wird dabei durch ein externes Mikroprogramm (ROM) gesteuert, welches für verschiedene Anwendungen programmiert werden kann. Um den FlintStone in einem minimalen System einzusetzen, benötigt man nebst Mikroprogrammspeicher nur zwei Register (Daten- und Adressregister) für Speicherzugriff sowie einen Taktgenerator (Fig. 2).

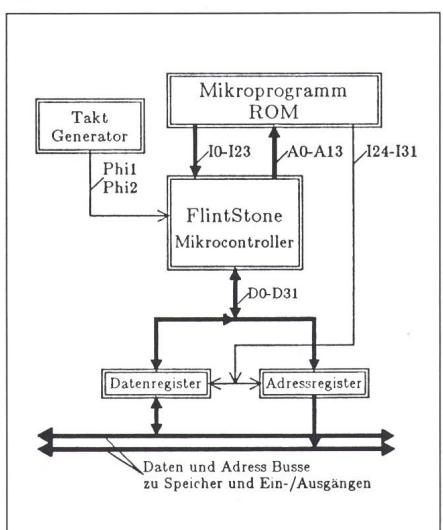
Die FlintStone-Architektur besitzt Ähnlichkeit mit dem weitverbreiteten 4-Bit-Slice-Baustein AMD-2901 [3]. Die grössten Unterschiede betreffen die Wortbreite (32 statt 4 Bit), die auf höhere Programmiersprachen und Grafikanwendungen optimierten Funktionen sowie die Integration der Mikroprogrammsteuerung auf denselben IC.

¹ Man kann dies etwa mit einem Mikroprozessor vergleichen, bei dem der Benutzer selber den Instruktionssatz definiert.



Figur 3 Blockdiagramm der FlintStone-Architektur

Der Datenpfad (im linken Teil) bearbeitet 32-Bit-Daten. Er besteht aus Registersatz (*Register File*), Rechenwerk (*ALU*), Schieberegister (*SHIFT*) und mehreren Multiplexern (*Mux*). Der Kontrollpfad (im rechten Teil) steuert den Mikroprogrammablauf mittels eines Programmzählers (*PCreg*), einer Condition-Code-Einheit (*CCreg*) und eines Multiplexers zum Ausführen bedingter Verzweigungen (*Amux*). Ein kleiner *STACK* erlaubt das Aufrufen von Unterprogrammen.



Figur 2 Möglicher Systemeinsatz des Flintstone-Mikrocontrollers

Das Mikroprogramm-Rom steuert den Ablauf der Operationen. Über zwei Spezialregister kann auf den Hauptspeicher zugegriffen werden; zuerst wird die Adresse in das Adressregister geschrieben, dann kann die Speicherzelle über das Datenregister ein- oder ausgelesen werden.

Das Blockdiagramm in Figur 3 erläutert die FlintStone-Architektur; Datenpfad und Kontrollpfad bilden die Grundelemente. Im Datenpfad werden die 32-Bit-Daten verarbeitet. Er besteht hauptsächlich aus Registersatz (*Register File*), Rechenwerk (*ALU*), Schieberegister (*SHIFT*) und mehreren Multiplexern (*Mux*). Der Kontrollpfad steuert den Ablauf des Mikroprogramms. Dies geschieht anhand eines Programmzählers (*PCreg*), einer Condition-Code-Einheit (*CCreg*) und eines Multiplexers zum Ausführen bedingter Verzweigungen (*Amux*). Vier Stackregister (*STACK*) erlauben zudem das Aufrufen von Unterprogrammen.

Um eine Idee der Funktionsweise des Prozessors zu vermitteln, soll hier die folgende Sequenz von zwei Mikroinstruktionen kurz durchgegangen werden:

$a := a + b$; if $a = 0$ then goto l ;

wobei a und b die Inhalte der Register 0 und 1 des Registersatzes und l eine beliebige Adresse des Mikrogramms darstellen.

Beim Programmstart muss die Adresse der ersten Instruktion vom PC-Register (*PCreg*), über den Adressmultiplexer (*Amux*), an die Adresspins (*A0-A13*) gelangen (Fig. 3). Der externe Instruktionsspeicher legt dann diese Instruktion an die Instruktionsspins (*I0-I23*), von wo sie im Instruktionssregister (*Ireg*), zwischengespeichert und dekodiert wird. Die so gewonnenen Signale steuern den gesamten Prozessor. Die erste Instruktion wird nun folgendes auslösen:

- Auslesen der Register 1 und 0 auf den *Abus* bzw. *Bbus* (es können gleichzeitig zwei Register ausgelesen werden),
 - Durchschalten des *Abus* und des *Bbus* durch die beiden Operanden-Multiplexer (*Rmux* und *Smux*) wodurch beide Werte am Eingang des

- Rechenwerkes (*ALU*) liegen,
- Wählen der Addieroperation in der *ALU*,
 - Durchschalten des *Fbus* durch den Resultat-Multiplexer (*Tmux*), womit das Resultat über den *Tbus* wieder an den Registersatz gelangt und in das Register 0 zurückgespeichert wird,
 - Berechnung der Condition-Codes und Zwischenspeicherung im *CC-Register* (*CCreg*),
 - Inkrementierung des Programmzählers und Ausgabe an den Adresspins damit die zweite Instruktion eingelesen werden kann.

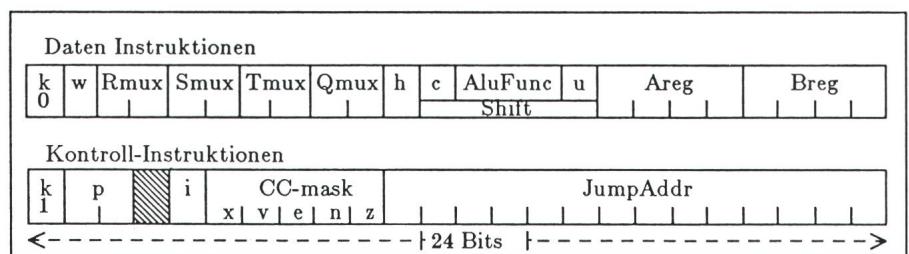
Die zweite Instruktion, die nun in das *Ireg* geladen wird, betrifft nur den Kontrollpfad:

- Die Sprungadresse *l* ist in der Instruktion enthalten und erscheint an einem der *Amux*-Eingänge.
- Der wiederum inkrementierte Programmzähler wird ebenfalls an den *Amux* angelegt.
- Aufgrund der zwischengespeicherten Condition Codes wird entweder der inkrementierte Programmzähler oder die Sprungadresse über den *Amux* an die Adresspins gelegt. So mit kann die nächste Instruktion eingelesen werden.

Dieses kleine Beispiel beansprucht nur wenige der vorhandenen Möglichkeiten, es zeigt jedoch recht anschaulich wie der FlintStone-IC funktioniert. Insbesondere weist es auf die zwei verschiedenen Instruktionstypen hin: die Dateninstruktionen zur Ausführung von Rechenoperationen, bei denen der Programmzähler nur inkrementiert wird, und die Kontrollinstruktionen, bei denen anhand der vorhergehenden Dateninstruktion eine Entscheidung gefällt wird.

Die Verbindung zur Aussenwelt geschieht über eine beachtliche Zahl von insgesamt 96 Pins: An den 14 Adresspins erscheint am Anfang eines Taktzyklusses die Adresse für den Mikrogrammmspeicher. Die abgerufene Instruktion wird hierauf über 24 Instruktionsspins geladen. Über die 32 Datenpins können Daten fliessen und weitere 26 Pins werden für Taktgeber, Reset, Interrupt, Testen und Speisung benötigt.

Die Figur 4 zeigt die Enkodierung der beiden Instruktionstypen. Jedes Feld steuert eine der Funktionseinheiten des IC, wobei das erste Feld über den Instruktionstyp entscheidet. Um Operationen auf den 32-Bit-Daten



Figur 4 Kodierung der Daten- und Kontrollinstruktionen

Jedes Feld steuert direkt einen Funktionsblock. Das erste Instruktionstyp (Daten, Kontrolle) an.

auszuführen, stehen drei Funktionsblöcke zur Verfügung (Fig. 3):

- Das Rechenwerk (*ALU*) kann addieren, subtrahieren und logische Operationen (AND, OR sowie XOR) ausführen. Multiplikation und Division können mit Hilfe des *Q-Registers* (*Qreg*) in 32 Taktzyklen berechnet werden.
- Im Schieberegister (*SHIFT*) können 64 Bits um bis zu 31 Positionen nach links geschoben werden; die 32 ersten Bits erscheinen dann am Ausgang.
- Die Find First One Funktion (*FFO*) entspricht einem Prioritäts-Encoder. Sie berechnet die Position der obersten (wahlweise auch untersten) «1» (oder «0») in einem 32-Bit-Wort.

Die zwei letzten Funktionsblöcke sind besonders bei Grafikoperationen hilfreich.

3. Implementation

3.1 Technologie

Die CMOS-Implementation des IC wird derzeit bei der Firma Faselec AG durchgeführt. Die Wahl von CMOS lässt sich durch die guten Eigenschaften dieser Technologie (Geschwindigkeit, Leistungs- und Flächenverbrauch) begründen. Der FlintStone-Prozessor wurde in statischer CMOS-Technik implementiert. Verglichen mit dynamischen CMOS-Techniken erlaubt diese einen einfacheren Schaltungsentwurf und einfacheres Testen, allerdings auf Kosten von Fläche und Geschwindigkeit. Der Vorteil beim Testen liegt darin, dass die Schaltung beliebig langsam laufen kann. Der Taktgeber kann jederzeit angehalten werden, so dass mittels der Testlogik die IC-internen Zustände in aller Ruhe (daher auch mit einfachen Testgeräten) angeschaut werden können.

3.2 Planung und Plazierung der Funktionsblöcke

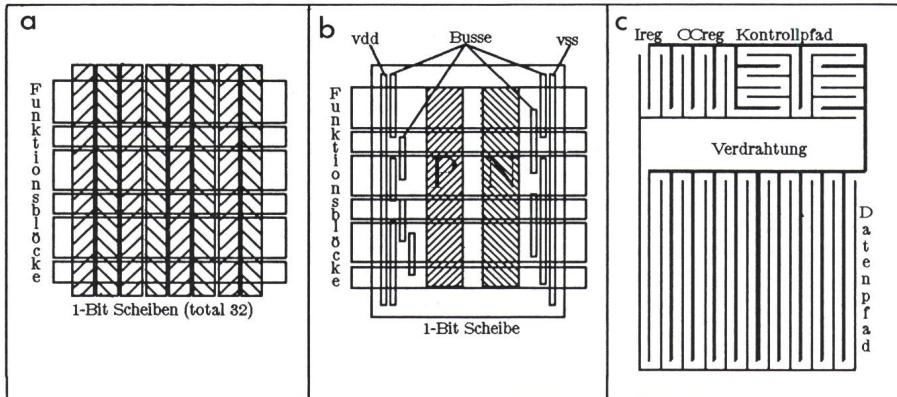
Ein VLSI-Projekt kann nur dann erfolgreich sein, wenn der Entwurf nach dem Top-Down-Prinzip durchgeführt wird. Zu Beginn der Planung werden die grossen Funktionsblöcke (Datenpfad, Kontrollpfad) auf dem IC plaziert. Diese Planung muss so früh wie möglich durchgeführt werden, da nur wenige (meist zwei, bei einigen Prozesstechnologien drei) Verdrahtungsbenen zur Verfügung stehen und die Anzahl benötigter Verbindungen meist sehr hoch ist. Bei unüberlegter (oder fehlender) Planung kann es leicht passieren, dass Verbindungen fast die Hälfte der gesamten IC-Fläche beanspruchen.

Gleich zu Beginn der Planungsphase bei FlintStone wurde der grösste Funktionsblock, der gesamte Datenpfad, bearbeitet. Im ersten Schritt wurden die Funktionsblöcke in einer Linie untereinander angeordnet (linearisiert), um den ganzen Pfad in ein Rechteck «verpacken» zu können. Dieses Rechteck (Fig. 5a) besteht aus vertikalen 1-Bit-Scheiben (bei FlintStone insgesamt 32) und übereinander angeordneten Funktionsblöcken.

Bei dieser Bit-Slice-Methode verlaufen alle Busse innerhalb der 1-Bit-Blöcke direkt neben den Transistoren. Somit entfallen die sonst nötigen breiten Verbindungskanäle (Fig. 5b). Allerdings müssen die Funktionsblöcke gleiche Breite haben, damit sie nahtlos aneinanderpassen (Abutment).

Der Kontrollpfad wurde nach der gleichen Methode ausgelegt. Die Breite der Adressen (14 Bits) ergab eine Organisation in 14 Bit-Slices. Dazu kommen das Instruktionsregister und die Condition-Code-Logik.

In der Planungsphase wird die Hauptausrichtung für Transistoren



Figur 5 Planung und Plazierung der Funktionsblöcke

- a Aufbau des Datenpfades aus vertikalen 1-Bit-Scheiben und horizontal liegenden Funktionsblöcken.
- b Die Datenbusse verlaufen innerhalb der Bit-Scheiben direkt neben den Transistoren. So werden breite Verbindungskanäle vermieden.
- c Sowohl V_{ss} (Masse) als auch V_{dd} (Speisespannung) müssen vollständig in Metall verlaufen, was eine verzahnte Struktur zur Folge hat.

— Masse (V_{ss})

— Speisung (V_{dd})

und Verbindungsdrähte der verschiedenen Ebenen in einem Farbplan² festgehalten. Diese wurde für FlintStone so gewählt, dass die Datenbusse vertikal in Metall verlaufen und die Steuersignale horizontal in Polysilizium. Um die Laufzeiten in diesen, quer über den Datenpfad laufenden, Polysilizium-Steuerleitungen zu reduzieren, wurde die Steuerlogik in die Mitte des Datenpfades plaziert, was eine Verbesserung um einen Faktor 4 brachte. Mit der Farbplanung wird auch die Stromversorgung geplant. Da sowohl V_{ss} (Masse) als auch V_{dd} (Speisespannung) vollständig in Metall verlaufen müssen, dürfen sich bei Netze nirgends kreuzen. Dies wird durch schöne, in Figur 5c veranschaulichte Verzahnung der beiden Netze erreicht.

Die Figur 6 veranschaulicht das Resultat der Planungsphase; die einzelnen Blöcke können auch im fertigen Layout (Fig. 1) erkannt werden.

3.3 Schaltungsentwurf

Beim FlintStone-Projekt wurde der Schaltungsentwurf mit dem VIVID-Layout-System [4] vorgenommen. Dieses erlaubt ein sogenanntes symboli-

sches Layout [5], welches das Zeichnen der Schaltung erheblich vereinfacht und beschleunigt; Änderungen können in kurzer Zeit durchgeführt werden. Das Arbeiten mit Symbolen statt mit Rechtecken und das Entfallen von «Design Rules» befreit den Designer etwas von der enormen Komplexität des Full-Custom-Designs [2]. Außerdem ist das symbolische Layout weitgehend technologieunabhängig.

Ein weiterer sehr wichtiger Punkt beim Schaltungsentwurf ist der hierarchische Aufbau der Schaltung. Hierbei bestehen nur die Zellen auf dem untersten Hierarchieniveau aus den Grundelementen Transistoren und Drähten. Alle höheren Zellen werden aus tieferliegenden zusammengesetzt. Ein solch modularer Schaltungsaufbau erlaubt einen besseren Überblick, und zudem können öfters Module wiederverwendet werden. Die Figur 7 zeigt als Beispiel die Hierarchie der hier auf vier Bit reduzierten FlintStone ALU mit Carry-Lookahead (CLA).

All dies hat natürlich seinen Preis. Der schwache Punkt bei symbolischem Layout liegt eindeutig in der Qualität des vom Compactor³ produzierten Layout.

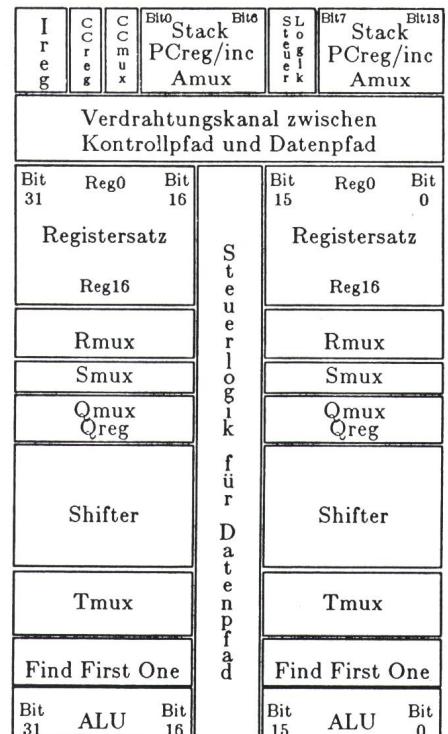
4. Simulation und Test

Simulation und Test gehören zu den wichtigsten Punkten beim Entwurf komplexer integrierter Schaltungen. Während die Aufgabe der Simulation ist, sicherzustellen, dass der Prototyp auf Anhieb funktioniert, ist die Aufgabe des Testens, festzustellen, ob wirklich alles richtig funktioniert, und wenn nicht, zu bestimmen, was genau defekt ist. Dass beides bei komplexen Schaltungen von über 20 000 Transistoren nicht einfach ist, liegt auf der Hand. Besonders das Testen gestaltet sich oft trickreich, kann doch der Zustand der Schaltung nur über die vorhandenen Pins betrachtet werden.

4.1 Test Logik

Um den FlintStone nach der Herstellung testen zu können, wurden die zwei Grundprinzipien des «Design for Testability» angewandt:

1. Kontrollierbarkeit: die Eingänge jedes Schaltungselementes müssen unabhängig vom Funktionieren der übrigen Schaltung steuerbar sein. Dies verhindert, dass ein Fehler im ersten Element einer Kette das Testen der folgenden Elemente unmöglich macht oder verfälscht.

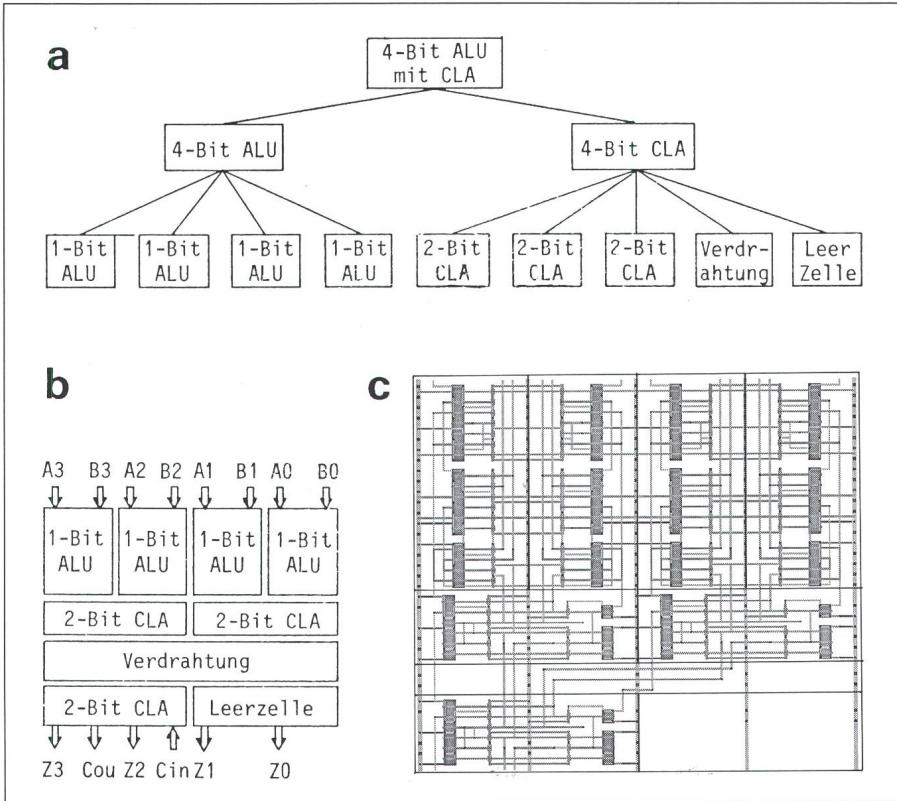


Figur 6 Lage der einzelnen Funktionsblöcke

Der Kontrollpfad liegt oberhalb des Datenpfades und ist mit diesem über einen Verdrahtungs-kanal verbunden.

² Um ein möglichst kompaktes Layout zu erreichen, wird für jede Verdrahtungsebene eine Hauptrichtung festgelegt. Der Name «Farbplan» (Color Plan) stammt daher, dass jede dieser Ebenen in einer anderen Farbe dargestellt wird.

³ Der Compactor übersetzt die symbolische Information in Maskengeometrie.



Figur 7 Hierarchisches Layout am Beispiel einer 4-Bit-ALU mit Carry-Lookahead

a Zellhierarchie: Nur die Zellen auf dem untersten Hierarchieniveau bestehen aus Transistoren und Drähten. Alle höheren Zellen werden aus tieferliegenden zusammengesetzt.

b Plazierung der Zellen

c Layout der 4-Bit-ALU mit Carry-Lookahead. Die Umrahmungen deuten die Zellgrenzen an.

2. **Sichtbarkeit** der Zustand jedes Schaltungselementes muss unabhängig vom Funktionieren der übrigen Schaltung sichtbar sein. Dies verhindert, dass ein Fehler im letzten Element einer Kette das Testen der vorhergehenden Elementen unmöglich macht oder verfälscht.

Im FlintStone-Datenpfad können mittels spezieller Testlogik die Zustände von sieben Bussen direkt an den Datenpins beobachtet werden (Sichtbarkeit); sowie drei spezielle Testregister geladen und an einen Bus zugeschaltet werden (Kontrollierbarkeit). So lassen sich beispielsweise **ALU**, **FFO** und **SHIFT** völlig unabhängig voneinander testen. Die Figur 8 zeigt (in grauer Schraffur) die im FlintStone eingebaute Testlogik. Die Implementation der Testlogik «kostet» beim FlintStone-IC acht zusätzliche Pins sowie etwa 8% mehr Flächenaufwand.

4.2 Simulationsmethoden

Die Komplexität der Schaltung und die damit verbundenen, sehr langen Simulationslaufzeiten machen her-

kömmliche Simulationsmethoden bei grossen integrierten Schaltungen unbrauchbar. Um ans Ziel zu gelangen, müssen neue Methoden eingesetzt werden wie hierarchische Simulation, Simulation auf verschiedenen Abstraktionsebenen und automatische Verifikation. Beim FlintStone-Projekt wurde die bereits im symbolischen Layout eingeführte Hierarchie in die Simulation miteinbezogen. Nur so konnte die Schaltung in Submodule aufgeteilt werden, deren Funktion und Zusammenspiel schnell simuliert bzw. verifiziert werden konnte.

Die Simulation wurde auf drei verschiedenen Abstraktionsebenen (funktional, Switch-Level, analog) durchgeführt, was verschiedene Kompromisse zwischen Genauigkeit und Simulationszeit erlaubte. Dazu wurden folgende Programme eingesetzt:

1. Ein **funktionaler Simulator**, der die Funktion und das Zusammenspiel grösserer Blöcke bzw. des gesamten IC simuliert. Die Funktionsweise wird in einer speziellen HDL-Programmiersprache (Hardware Description Language) beschrieben. Funktionale Si-

mulation ist sehr schnell (etwa 1 Sekunde für einen Taktzyklus bei der Simulation des gesamten FlintStone-IC), recht einfach und übersichtlich. Sie hat aber den Nachteil, dass kein direkter Zusammenhang zur wirklichen Schaltung besteht.

2. Ein **Switch-Level Simulator**, der Transistoren vereinfacht als Schalter modelliert. Damit konnte die Funktionsweise von Schaltungen meist richtig simuliert werden, wenn man von Feedback-Problemen und Laufzeiten absieht. Dieser Simulator ist zwar immer noch recht schnell, aber doch schon um eine Größenordnung langsamer als der funktionale Simulator (etwa 10 Sekunden für einen Taktzyklus). Er liefert aber die schnellste Methode, die «wirkliche» Schaltung zu simulieren.

3. Ein **analoger Simulator**, der Transistoren mehr oder weniger aufwendig als analoge Bauelemente modelliert. Die Simulation wird hierbei beliebig langsam (z.B. Stunden für einen 100-ns-Taktzyklus).

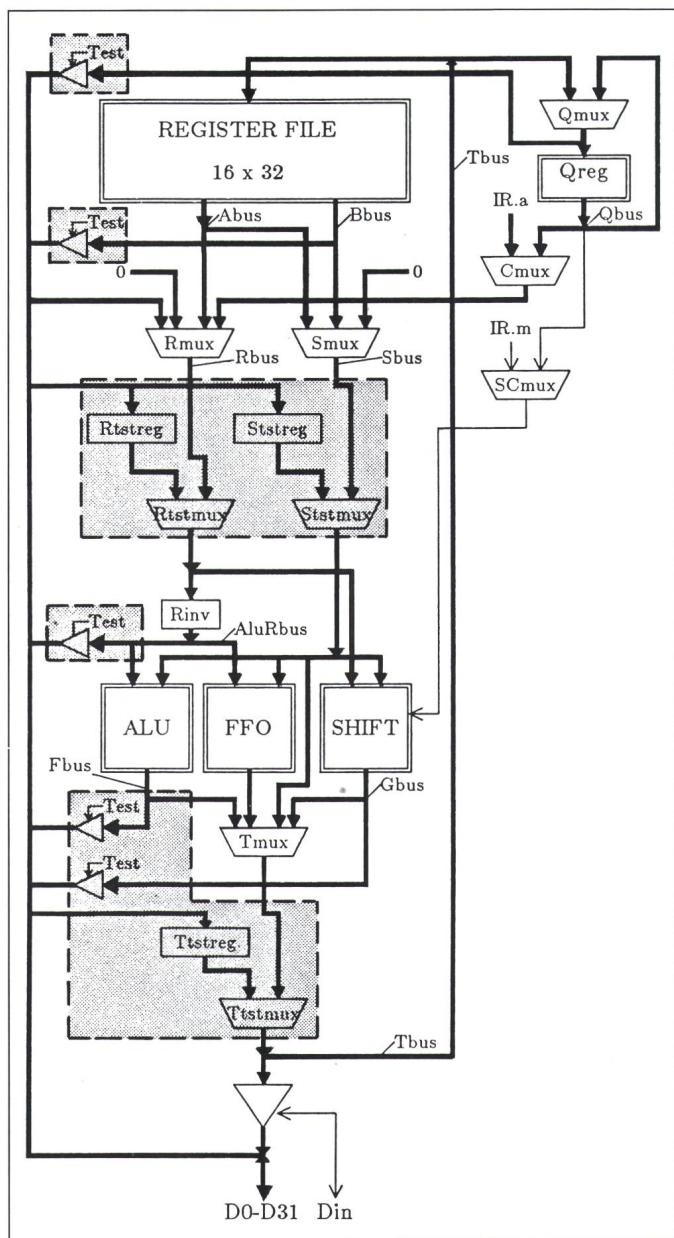
Besonders zum Tragen kamen die verschiedenen Simulationsmöglichkeiten, weil damit die einzelnen Blöcke unterschiedlich simuliert werden konnten: kritische Bestandteile analog, andere Switch-Level und bereits getestete oder noch nicht existierende funktional.

Da die grösseren Blöcke im FlintStone-IC sehr viele Anschlüsse haben (allein der **Rmux** hat schon etwa 150), ist eine automatische Kontrolle der Simulationsresultate unabdingbar. Für jeden grösseren Funktionsblock wurde eine HDL-Beschreibung und ein Testprogramm erstellt. Die Korrektheit eines Blocks konnte durch Testsequenzen überprüft werden, die das Testprogramm erzeugte und die gleichzeitig dem Switch-Level-Simulator und dem Funktional-Simulator eingegeben wurden. Ein für das FlintStone-Projekt entwickeltes Verifikationsprogramm verglich die Resultate beider Simulationen und meldete auf Fehler hinweisende Differenzen.

Durch die Vollautomatisierung der Simulation konnte jede Änderung der Schaltung (und war es auch nur das Verschieben eines Transistors) «per Knopfdruck» neu verifiziert werden.

5. Resultate

Mit dem Entwurf des FlintStone-Prozessors wurde im April 1986 begonnen. Der Layout ist (bei Vollen-



Figur 8
Blockdiagramm des
Datenpfades mit
Testlogik in
grauer Schraffur

hungen noch verbessert werden sollen. Besonders die in Tabelle I zusammengefassten Laufzeiten und Datenmengen können bei grösseren Entwicklungen nicht mehr toleriert werden.

Chip Grösse	21 826 Transistoren
	6,2mm × 4,2mm
Symbolischer Layout	159 Zellen
	1,560 Mbytes
Masken	70 Mbytes
	2,7 Mio. Rechtecke
Compaction Datenpfad	12 h CPU Zeit
	54 Mbytes virtueller Speicher (Sun 3/160)
Design-Rule Check	14 h CPU Zeit (VAX 785)

Tabelle I 9 Laufzeiten und Datenmengen beim Entwurf

Trotz diesen Schwierigkeiten konnte das Projekt erfolgreich durchgeführt werden. Die dabei gewonnenen Erfahrungen im Entwurf komplexer integrierter Schaltungen und in der Benutzung moderner CAD-Werkzeuge, die im übrigen durch Vergleich mit anderen, grösseren Projekten [6; 7] bestätigt werden, sind für die weiteren Arbeiten sehr wertvoll.

Literatur

- [1] H. Eberle: Hardware Description of the Workstation Ceres. ETH Zürich, Institut für Informatik, Bericht Nr. 70, Januar 1987.
- [2] W. Fichtner: Design-Methoden für digitale Schaltungen. Bull. SEV/VSE 78(1987)11, S. 595...600.
- [3] Bipolar microprocessor logic interface, Am 2900 family data book. Sunnyvale/California, Advanced Micro Devices Inc., 1985.
- [4] J.B. Rosenberg: CAD tools for mask generation. In: W. Fichtner and M. Morf: VLSI tools and applications. Boston, Kluwer Academic Publishers, 1987; p. 147...212.
- [5] N. Weste and K. Eshragian: Principles of CMOS VLSI design, a systems perspective. Reading/Massachusetts, Addison-Wesley, 1985.
- [6] S. Przybylski: The Design Verification and Testing of MIPS. Conference on Advanced Research in VLSI, M.I.T, Januar 1984, S. 100...109.
- [7] D.R. Ditzel and A.D. Brenbaum: Experience with CAD Tools for a 32-Bit VLSI Microprocessor. In: W. Fichtner and M. Morf: VLSI CAD Tools and Applications, Boston, Kluwer Academic Publishers, 1987; S. 327...350.

dung dieses Manuskriptes im Mai 1987) bis auf die Einbettung in den Pad-Frame abgeschlossen. Alle Funktionsblöcke wurden einzeln simuliert und verifiziert. Die noch ausstehenden Arbeiten umfassen Simulation und Verifikation des gesamten ICs.

Die Verwendung von symbolischem Layout hat wesentlich dazu beigetragen, dass dieses Projekt mit einem Mi-

nimum an Personal realisiert werden konnte. Allerdings soll hier nicht verschwiegen werden, dass die Qualität des vom Compactor generierten Maskenlayouts nicht immer zufriedenstellend war. Im allgemeinen kann festgestellt werden, dass die verwendeten (und durchaus dem Stand der Technik entsprechenden) CAD-Werkzeuge durch konzentrierte Forschungsbemü-

Câbles à haute tension
Hochspannungskabel

10-220 KV

EPR

10 ans
Jahre



COSSONAY



10 ans de câbles EPR en Suisse

Historique

Lorsqu'en 1976, après quatre ans de développement et d'essais, les premiers câbles 20 kV isolés à l'EPR furent mis en service dans les réseaux, peu de personnes, si ce n'est les intéressés directs, auraient pensé que ce type de câble connaîtrait une carrière aussi extraordinaire au cours des années à venir.

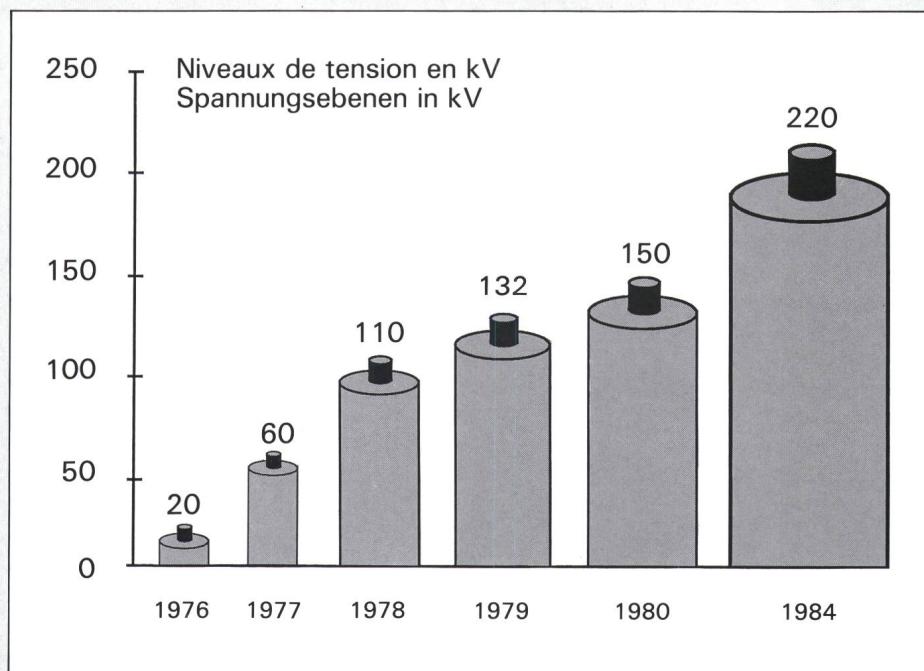
En 1977 déjà, les premiers câbles de 45 et 60 kV firent leur apparition sur le marché. Grâce à un large programme d'essais envisagé dès le début, les étapes visant des niveaux de tensions toujours plus élevés ont été franchies avec une rapidité étonnante. Le graphique suivant démontre clairement cette évolution :

10 Jahre EPR-Kabel in der Schweiz

Historische Entwicklung

Als im Jahre 1976 die ersten 20kV-EPR-Kabel nach mehr als vierjähriger Entwicklungs- und Erprobungszeit zum normalen Netzeinsatz kamen, glaubten wohl ausser den, an deren Geburt direkt Beteiligten nur wenige daran, dass diesem Kabeltyp in den folgenden Jahren eine ganz aussergewöhnliche Karriere beschieden sei.

Schon 1977 kamen bereits die ersten 45 und 60kV-EPR-Kabel auf den Markt und dank einem von Anfang an breit ausgelegten Erprobungsprogramm wurden auch die Sprünge zu immer höheren Spannungsebenen überraschend schnell gemeistert. Aus folgender Darstellung sind diese Etappen deutlich ersichtlich:



Les divers essais de longue durée réalisés dès le début avec des exigences très élevées ont joué un rôle déterminant dans ce développement particulièrement rapide. Les expériences nécessaires au niveau des accessoires ont été acquises simultanément.

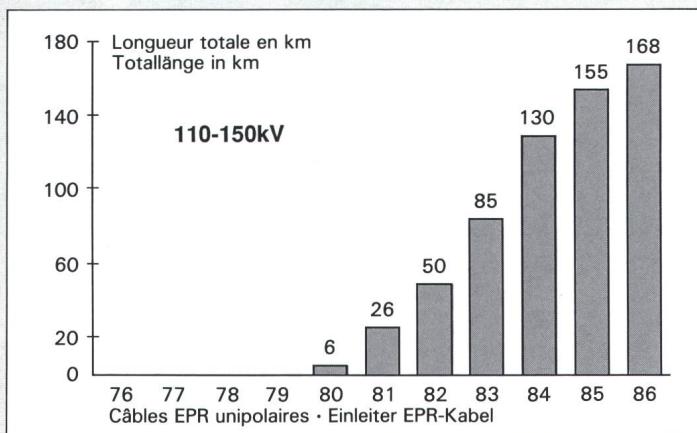
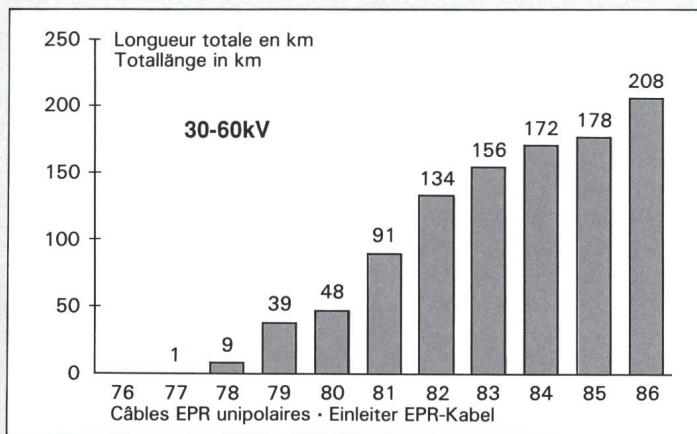
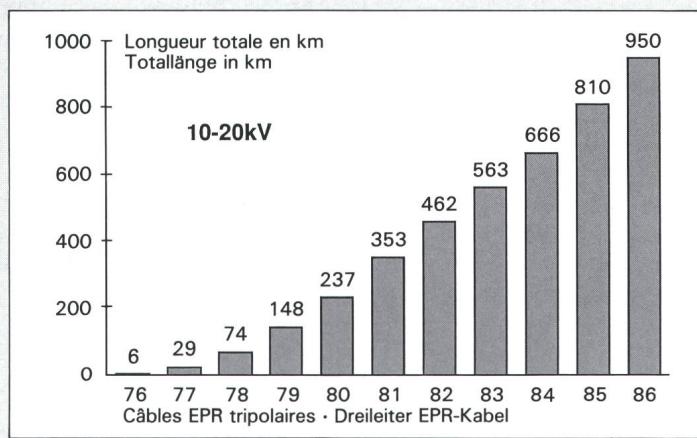
Für diese ungewöhnlich rasche Entwicklung spielten die von Anfang an, mit immer höheren Beanspruchungen durchgeföhrten Langzeitversuche eine entscheidende Rolle; dabei konnten auch gleichzeitig die notwendigen Erfahrungen mit den Zubehören gesammelt werden.

Evolution de la mise en service de câbles EPR-Cossonay au cours des 10 dernières années.

Les graphiques suivants indiquent les longueurs cumulées de **câbles EPR Cossonay** installées depuis 1976 :

Übersicht über die nach 10 Jahren im Betrieb stehenden Cossonay-EPR-Kabel.

Die folgenden Darstellungen geben Aufschluss über die kumulierten Längen der in der Schweiz seit 1976 in Betrieb stehenden **Cossonay-EPR-Kabel**:



Depuis 1984, un **câble EPR Cossonay de 220 kV** est en service à la sous-station de Mettlen à l'entière satisfaction des intéressés.

Seit 1984 versieht zudem ein **220kV-Cossonay-EPR-Kabel** den Betrieb in der Unterstation Mettlen zur vollen Zufriedenheit aller Beteiligten.

RÉSULTATS D'EXPLOITATION

Du point de vue de l'exploitant, ces résultats constituent le critère de jugement essentiel d'un câble ou d'un type d'isolation. Aucune étude théorique sur les propriétés de matériaux ou la construction, aussi bien fondée soit-elle, n'a pour l'exploitant un poids aussi fort qu'un essai réalisé dans des conditions réelles.

L'expérience effectuée avec ce type de câble, pour tous les niveaux de tension mentionnés, peut se résumer très succinctement:

aucune défaillance connue pouvant être imputée au câble!

(abstraction faite des seuls défauts de montage et de causes extérieures rencontrés à ce jour, soit:

3 défauts de montage (1 extrémité, 2 jonctions)
3 dommages occasionnés en cours d'installation
1 coup de foudre sur une tête de câble).

Aujourd'hui, plus de **1,5 million de kg de matière isolante EPR** ont été utilisés pour la fabrication de câbles Cossonay.

Parallèlement à la facilité de montage fort appréciée du câble EPR, sa grande fiabilité constitue une des raisons principales de son développement spectaculaire.

BETRIEBSERGEBNISSE

Vom Betreiber her gesehen, ist dies sicher der wichtigste Punkt für die Beurteilung eines Kabel- oder Isolationstypes. Keine noch so gut fundierte theoretische Abhandlung über die Eigenschaften von Materialien oder Konstruktionen hat für den Betrieb einer Kabelanlage eine ähnliche Aussagekraft wie die Betriebserfahrungen.

Die Betriebserfahrungen mit diesem Kabeltyp können für alle vorerwähnten Spannungsebenen sehr kurz zusammengefasst werden; es braucht dazu keine Fehlerstatistik.

Keine Ausfälle, auf Grund von Kabelfehlern bekannt!

(ausgenommen sind selbstverständlich die heute bekannten Fremdeinflüsse und Montagefehler, d. h.:

3 Montagefehler (1 Endverschluss, 2 Muffen)
3 Baubeschädigungen
1 direkter Blitzschlag in einen Endverschluss).

Damit stehen heute über **1,5 Millionen kg EPR-Isolationsmaterial**, auf den Cossonay-Kabel verwendet, im Einsatz.

Nebst der hochgeschätzten Montage-freundlichkeit, dürfte diese aussergewöhnlich hohe Betriebssicherheit der Hauptgrund sein, dass dieser Kabeltyp, in solch kurzer Zeit, eine so weite Verbreitung gefunden hat.



SOCIETE ANONYME DES

CABLERIES & TREFILERIES DE COSSONAY

CH-1305 COSSONAY-GARE TEL. 021/87 17 21 TELEX 24 199 FAX 021/87 30 35