Datenverschlüsselung bei 1 Gbit/s

Autor(en): **Eberle, Hans**

Objekttyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses**

Band (Jahr): **85 (1994)**

Heft 9

PDF erstellt am: **08.08.2024**

Persistenter Link: https://doi.org/10.5169/seals-902558

Zur Verschlüsselung von Daten, beispielsweise in schnellen Rechnernetzwerken, wird leistungsfähige Chiffrierhardware benötigt. Der in diesem Beitrag beschriebene Chip chiffriert Daten gemäss dem Data Encryption Standard mit einer Verschlüsselungsrate von 1 Gbit/s und ist damit der schnellste bekannte Chiffrierchip. Zur Realisierung wurde ein Gallium-Arsenid-Gate-Array mit einer Kapazität von 50000 Transistoren verwendet.

Datenverschlüsselung bei 1 Gbit/s

■ Hans Eberle

Die Verschlüsselung von Daten in Rechnersystemen bietet Schutz vor unberechtigtem oder missbräuchlichem Zugriff. In diesem Artikel wird ein Chiffrierchip vorgestellt, welcher für die sichere Datenübertragung in Rechnernetzen entwickelt wurde. Der beschriebene Chip realisiert den Data Encryption Standard (DES), der in kommerziellen Anwendungen weit verbreitet ist und eine effiziente Hardwareimplementation ermöglicht.

Für die sichere Netzwerkübertragung wird Chiffrierhardware benötigt, welche schnell genug ist, um mit der Übertragungsrate im Netz mithalten zu können. Der beschriebene Chiffrierchip wurde für das AN2-Netzwerk [1] entwickelt, welches Übertragungsraten von bis zu 1 GBit/s erlaubt. Eine Chiffrierrate von 1 GBit/s kann mittels einer schnellen VLSI-Technologie erreicht werden. Kandidaten von möglichen Logikfamilien und Halbleitermaterialien sind emittergekoppelte Logik (ECL) für Silizium (Si) und direktgekoppelte FET-Logik (DCFL)[1] für Gallium-Arsenid (GaAs). Während sich Si-ECL seit längerer Zeit etabliert hat, muss sich die Alternative GaAs-DCFL erst noch durchsetzen. GaAs ist ein attraktives Halbleitermaterial; wegen der höheren Elektronenmobilität des GaAs-Halbleiters schalten GaAs-Transistoren rund zweimal so schnell wie Si-Transistoren. Des weiteren erreichen die Elektronen in GaAs ihre maximale Geschwindigkeit bei einer niedrigeren Spannung als in Silizium, weshalb für den Betrieb eine kleinere Versorgungsspannung und damit ein geringerer Leistungsverbrauch resultiert. Der Nachteil von GaAs ist der wenig ausgereifte Herstellungsprozess. Obschon GaAs seit mehr als 20 Jahren als mögliche Alternative zu Si bekannt ist, konnten die Herstellungsschwierigkeiten erst in neuerer Zeit überwunden werden. Diese Verbesserungen ermöglichen den Einsatz von GaAs für VLSI-Schaltungen

Wer ist der Schnellste?

Im Bulletin SEV/VSE 9/92 haben wir einen Artikel über den von der ETH Zürich und von der Ascom Tech AG entwickelten Blockchiffrieralgorithmus Idea sowie dessen Implementierung in CMOS-Technik publiziert [8]. Der Autor des vorliegenden Beitrags stellt jetzt einen in GaAs-Technologie implementierten DES-Chip vor, welchen er während seines Aufenthalts am Systems Research Center der Digital Equipment Corporation in Palo Alto, Kalifornien, entwickelt hat. Dieser soll dem Idea-Chip bezüglich Geschwindigkeit um rund einen Faktor 5 überlegen sein. Trotzdem stellt sich für den Autor die Frage, ob die Leistungs- und Geschwindigkeitsvorteile der GaAs-Technologie genügen, um die Vorteile der hohen Packungsdichte und des gut eingeführten Prozess-Handlings wettzumachen, welche Si-CMOS auszeichnen.

[1] Direktgekoppelte FET-Logik ist ähnlich im Schaltungsaufbau wie Enhancement-Depletion-nMOS-Logik. Dieser Logiktyp ist einfach im Schaltungsaufbau und bietet eine hohe Integrationsdichte.

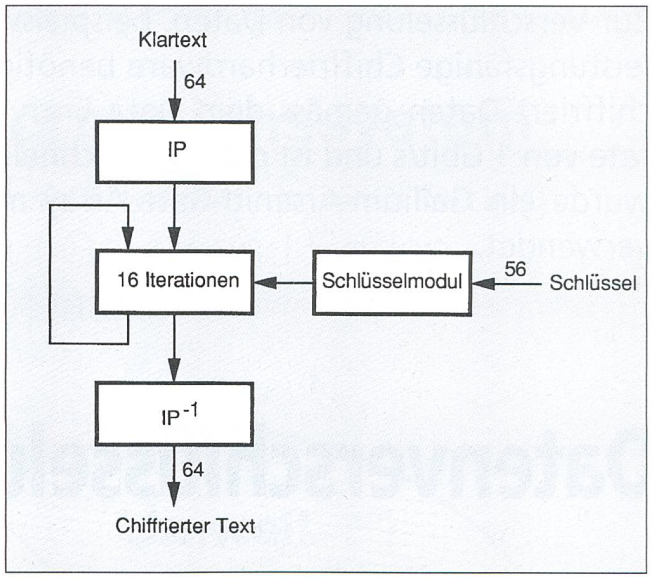
Adresse des Autors:
Prof. Dr. *Hans Eberle*, ETH-Zentrum, Institut für Computersysteme, 8092 Zürich.

und waren auch die Motivation für die vorliegende Arbeit.

DES-Algorithmus

Der DES-Algorithmus wurde vom amerikanischen Standardisierungsbüro, dem National Bureau of Standards (NBS) im Jahre 1977 entwickelt [2; 3]. Der Algorithmus verschlüsselt 64 Bit breite Datenblöcke unter Verwendung eines 56 Bit breiten Schlüssels[2]. Der Algorithmus benutzt zwei verschiedene Operationstypen: Permutation und Substitution. Wie in Bild 1 gezeigt ist, wird ein Datenblock zuerst einer Initialpermutation (IP), dann 16 Iterationen einer komplexen, vom Schlüssel abhängigen Berechnung und schliesslich der inversen Initialpermutation IP^{-1} zugeführt[3].

Bild 1 Überblick über den DES-Algorithmus



Das Bild 2a zeigt eine erweiterte Version der 16 Iterationsschritte, die für das Chiffrieren benötigt werden. Zuerst wird das von der Initialpermutation erzeugte 64-Bit-Resultat in zwei 32 Bit breite Hälften L_0 und R_0 aufgeteilt. Die Resultate L_n und R_n einer Iteration sind wie folgt definiert:

$$L_n = R_{n-1} \quad (1a)$$
$$R_n = L_{n-1} \text{ XOR } f(R_{n-1}, K_n), \quad (1b)$$

wobei n im Bereich von 1 bis 16 liegt. Nach Beenden der 16. Iteration werden die beiden 32-Bit-Worte L_{16} und R_{16} wieder zu einem 64-Bit-Block zusammengefügt und anschliessend IP^{-1} zugeführt.

Das Bild 2b zeigt eine detaillierte Version des Schlüsselmoduls, welche ebenfalls dem Chiffrieren dient. Die 56 Schlüssel-Bits werden zuerst in PC1 permutiert und anschliessend in zwei 28-Bit-Worte C_0 und D_0 aufgeteilt. Die Resultate C_n und D_n werden erhalten, indem C_{n-1} und D_{n-1} um eine oder zwei Bitpositionen rotiert werden, wobei n im Bereich von 1 bis 16 liegt. Die Anzahl zu rotierender Positionen ist ein fester Bestandteil des Algorithmus. Nach 16 Iterationen wurde jede Schlüsselhälfte um insgesamt 28 Positionen rotiert, das heisst C_{16} entspricht C_0 und D_{16} entspricht D_0. Die 16 Schlüsselworte K_n werden erhalten, indem jeweils 48 Bits von C_n und D_n gewählt werden und in PC2 permutiert werden.

Verschlüsselungs- und Entschlüsselungsalgorithmus benutzen denselben Datenpfad. Sie unterscheiden sich lediglich in der Wahl der Schlüssel-Bits, welche von der Funktion f benutzt werden. Zwar werden die gleichen 16 Teilschlüssel benutzt, jedoch in unterschiedlicher Reihenfolge. So wird beispielsweise der Schlüsselwert K_1 beim Chiffrieren für die erste Iteration und beim Dechiffrieren für die 16. Iteration verwendet; der Schlüsselwert K_2 wird beim Chiffrieren für die zweite Iteration und beim Dechiffrieren für die 15. Iteration verwendet usw. Die Reihenfolge der Teilschlüssel wird umgekehrt, einfach indem die Richtung der Rotationsoperation umgekehrt wird, welche auf C_{0-15} und D_{0-15} ausgeführt wird; konkret bedeutet dies, dass C_{0-15} und D_{0-15} während der Verschlüsselung nach links, während der Entschlüsselung nach rechts rotiert werden.

Bild 3 beschreibt die Berechnung, welche von der Funktion f ausgeführt wird. Zuerst werden die 32 Bits der rechten Hälfte R permutiert und in E auf 48 Bits erweitert. Die Erweiterung E geschieht durch Wiederholen gewisser Bits. Das 48-Bit-Resultat wird dann mittels der XOR-Operation mit dem vom Schlüsselmodul erhaltenen 48-Bit-Wert K verknüpft. Die resultierenden 48 Bits werden nun aufgeteilt in 6-Bit-Blöcke und den acht Substitutionsboxen S_{1-8} zugeführt. Jede dieser Substi-

Wie funktioniert ein Chiffrieralgorithmus?

Mathematisch gesehen entspricht ein Chiffrieralgorithmus wie der DES-Algorithmus einer extrem grossen Anzahl von Transformationen. Jede Transformation gibt an, wie eine Folge von Symbolen, beispielsweise eine Mitteilung, von einer lesbaren, verständlichen Form in eine verrauschte, unverständliche Form abgebildet wird. Welche Transformation angewendet werden soll, wird durch den Chiffrierschlüssel bestimmt.

Der Algorithmus muss für jede Transformation eine inverse Transformation besitzen, damit die verrauschte Mitteilung wieder zurück in ihre lesbare Form übersetzt werden kann. Dieser Vorgang wird Dechiffrieren genannt und ist nur möglich, falls der beim Chiffrieren verwendete Schlüssel bekannt ist.

Für das sichere Übertragen einer Mitteilung zwischen zwei mit Chiffrierhardware oder -software ausgestatteten Rechnern genügt es, dass der Schlüssel geheimgehalten wird. Die Details des Algorithmus können wie im Falle des DES-Algorithmus öffentlich bekannt sein.

Die Transformationen, welche von einem Chiffrieralgorithmus angewendet werden, benutzen im wesentlichen zwei Operationstypen: Permutation und Substitution. Permutationen verändern die Position von Symbolen, nicht aber deren Wert, während Substitutionen den Wert der Symbole verändern, nicht aber deren Position. Es hat sich gezeigt, dass durch Zusammenfügen von alternierenden Schritten von Permutationen und Substitutionen besonders sichere Chiffrieralgorithmen erhalten werden.

Die im DES-Algorithmus vorkommenden Permutationen werden Initial Permutation (IP), Inverse Initial Permutation (IP^{-1}), E-Bit Selection Table (E), Permutation Function (P), Permutated Choice 1 (PC1) und Permutated Choice 2 (PC2) genannt. Die Substitutionen werden Substitution Boxes 1–8 (S1–S8) genannt.

[2] Inklusive Paritätsbits ist der Schlüssel 64 Bit breit.
[3] Die Namen von Funktionen und Registern wurde in Anlehnung an die DES-Spezifikationen in [2] gewählt.

tutionsboxen (S-Boxen) realisiert eine unterschiedliche nichtlineare Funktion. Schliesslich werden die acht 4-Bit-Resultate der S-Boxen zu einem 32-Bit-Wort zusammengefügt und der Permutation P zugeführt.

Sollen Datenströme chiffriert werden, welche länger als 64 Bit sind, so besteht die naheliegende Methode darin, den Strom in 64-Bit-Blöcke zu unterteilen, welche einzeln chiffriert werden. Diese Methode nennt sich Electronic Code Book (ECB) [4]. Da bei gegebenem Schlüssel und Klartextblock der resultierende verschlüsselte Block stets derselbe ist, kann der Originalblock mittels Frequenzanalyse herausgefunden werden. Es werden deshalb alternative Verfahren verwendet, welche auf dem Prinzip der Diffusion beruhen und zum Ziel haben, die im Klartext enthaltene Redundanz im chiffrierten Text zu reduzieren. Das am häufigsten angewendete Diffusionsverfahren nennt sich Cipher Block Chaining (CBC) [4], welches in Bild 4 dargestellt ist. Der Klartext p wird dabei in 64-Bit-Blöcke p_{1-n} aufgeteilt. Der Chiffrierblock c_i wird wie folgt berechnet:

$$c_i = DES_k(p_i \text{ XOR } c_{i-1}) \qquad (2)$$

Der resultierende Chiffriertext ist $c = c_1 c_2 - c_n$. Kennt man den Schlüssel k und den Chiffrierblock c_0, welcher auch Initialisierungsvektor genannt wird, so kann der Chiffriertext entschlüsselt werden, indem man den Klartext p_i wie folgt berechnet:

$$p_i = DES_k^{-1}(c_i) \text{XOR } c_{i-1} \qquad (3)$$

GaAs-Technologie

GaAs eignet sich für schnelle VLSI-Schaltungen aus folgenden Gründen [5]:

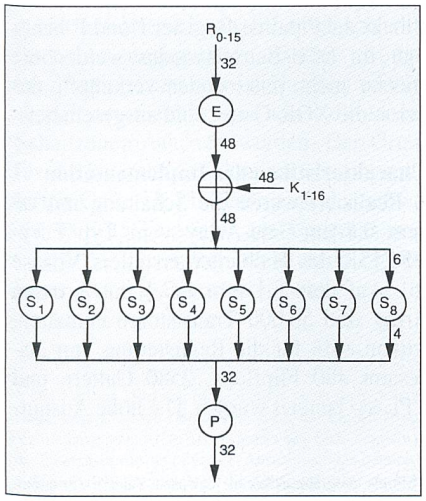


Bild 3 Erweiterte Version der Funktion f

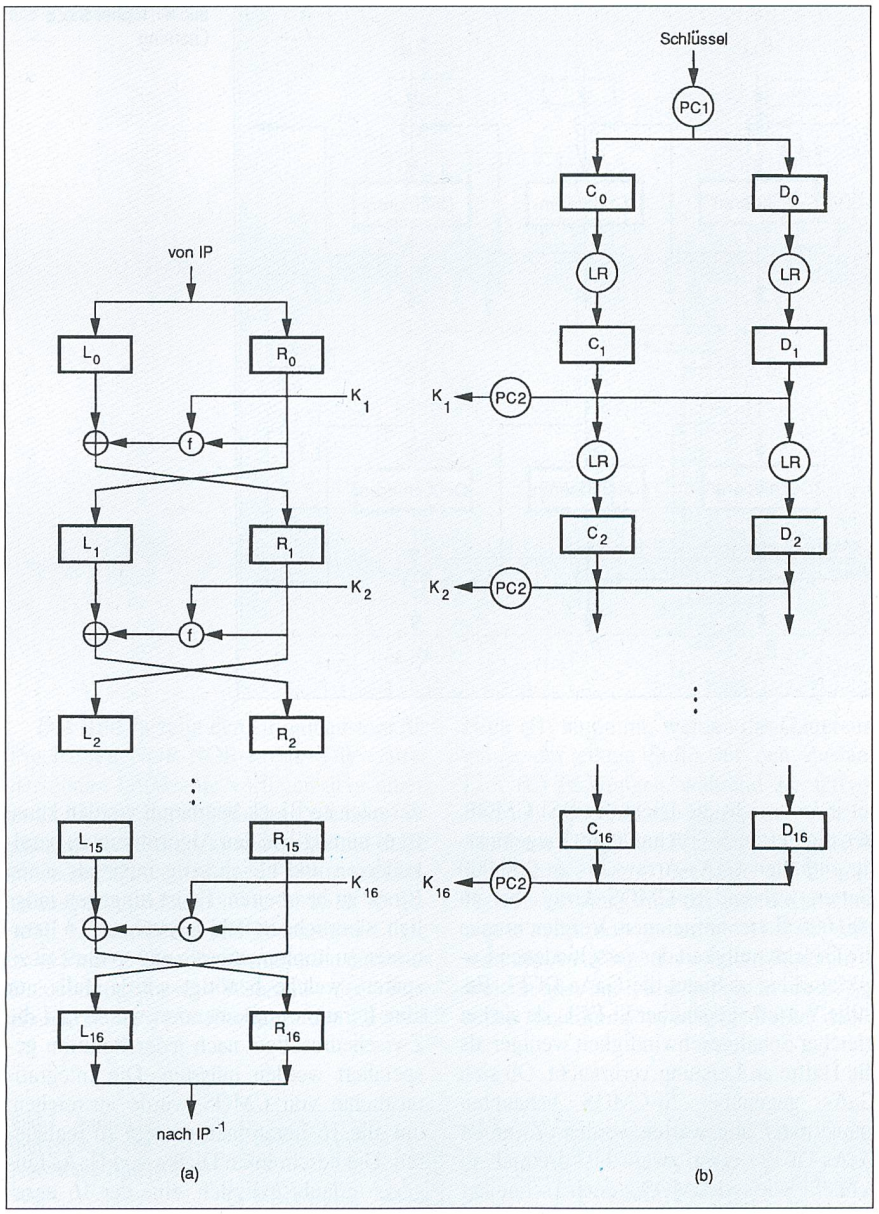


Bild 2 Erweiterte Version der 16 Iterationen (a) und des Schlüsselmoduls (b) für das Verschlüsseln von Daten

– Elektronen bewegen sich in GaAs bis zu zehn Mal schneller als in Si. Für GaAs-DCFL-Schaltungen resultiert damit ein Geschwindigkeitsvorteil von einem Faktor zwei.
– GaAs-Schaltungen sind immuner gegen Strahlung als Si-Schaltungen. Der Grund liegt darin, dass GaAs-Schaltungen keine dielektrischen Lagen besitzen, wie beispielsweise die bei Si-Schaltungen verwendeten Oxidschichten, welche durch Strahlung zerstört werden können.
– Da GaAs einen hohen Widerstandswert besitzt, können GaAs-Transistoren auf der Oberfläche des halbisolierenden GaAs-Substrates ganz einfach dadurch isoliert werden, indem man sie um einen Mikrometer voneinander entfernt plaziert.

Ein Nachteil von GaAs ist die schlechte Mobilität der Löcher im Gegensatz zur hohen Beweglichkeit der Elektronen, was p-Kanal-Transistoren und damit komplementäre Schaltungen unattraktiv macht. Eine weitere Schwierigkeit besteht in der Herstellung von Oxidschichten und damit von MOS-Transistoren. Der Herstellungsprozess für GaAs-Gate-Arrays ist verhältnismässig einfach. Insgesamt werden elf Masken benötigt, was höchstens der Hälfte der Anzahl Masken entspricht, welche für die Herstellung von Si-Chips gebraucht werden.

Vergleicht man GaAs mit Si bezüglich Integrationsdichte, so offeriert GaAs-DCFL eine höhere Integrationsdichte als Si-ECL, die schnellste Si-Technologie, bie-

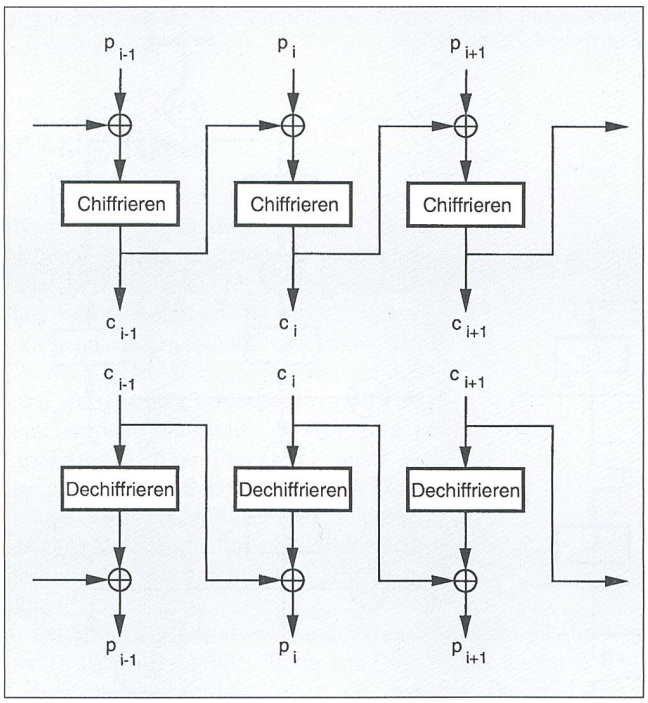


Bild 4 Cipher Block Chaining

tet jedoch nicht die Dichte von Si-CMOS, der dichtesten Si-Technologie. Gegenwärtig enthalten GaAs-Arrays bis zu 200 000 Gatter, während Si-CMOS-Arrays bis zu 800 000 Gatter aufnehmen. Vergleicht man die Geschwindigkeit der verschiedenen Logikfamilien, so bietet die GaAs-DCFL-Familie Vorteile gegenüber Si-ECL, da sie bei gleicher Schaltgeschwindigkeit weniger als die Hälfte an Leistung verbraucht. Ob sich GaAs gegenüber Si-CMOS behaupten kann, muss abgewartet werden. Zwar ist GaAs-DCFL etwa zwei- bis dreimal so schnell wie Si-CMOS, doch schneidet GaAs bezüglich Stromverbrauch erst bei Taktraten über 100 MHz vorteilhaft ab.

Implementierung des DES-Chips

Organisation

Die Geschwindigkeit eines in Hardware realisierten Algorithmus kann auf zwei Arten verbessert werden. Entweder wählt man eine dichte, jedoch verhältnismässig langsame Technologie wie Si-CMOS und erreicht eine hohe Leistung, indem man den Algorithmus parallelisiert und die Anzahl Logikstufen reduziert, oder man wählt eine schnelle, jedoch wenig dichte Technologie wie beispielsweise Si-ECL oder GaAs-DCFL. Der DES-Algorithmus ist nur beschränkt parallelisierbar. Wie in Bild 4 gezeigt wurde, wird in der CBC-Betriebsart ein Block nach dem Chiffrieren mit dem darauffolgenden Klartext-Block verknüpft. Da das Resultat verfügbar sein muss, bevor der nächste Block bearbeitet werden kann, ist es unmöglich, den Algorithmus zu parallelisieren und gleichzeitig mehr als einen Block zu bearbeiten. Es ist hingegen möglich, sämtliche in Bild 1 gezeigten 16 Iterationen zu implementieren und so die Zeit zu sparen, welche benötigt würde, falls nur eine Iteration implementiert würde und die Zwischenresultate nach jeder Iteration gespeichert werden müssten. Die Integrationsdichte von CMOS würde ausreichen, um alle 16 Iterationen einzeln zu realisieren. Die beschränkte Dichte von GaAs hingegen erlaubt lediglich, eine der 16 Iterationen zu implementieren, welche dann für alle 16 Iterationen wiederverwendet wird. Um die Datenrate von 1 GBit/s zu erreichen, muss jeder Block in 64 ns verarbeitet werden, was 4 ns pro Iteration oder einer Taktrate von 250 MHz entspricht.

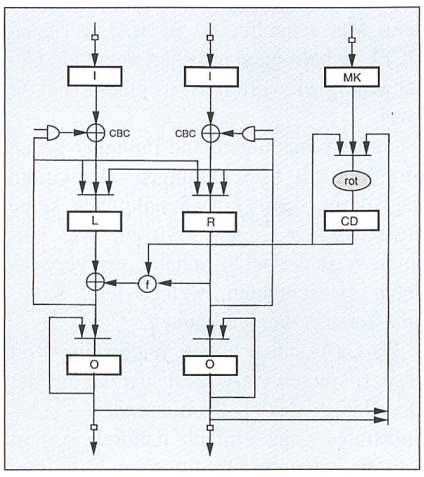


Bild 5 Chiffrieren

Blockdiagramme mit den Funktionsblöcken und Datenpfaden für Chiffrierung und Dechiffrierung sind in den Bildern 5 und 6 dargestellt. Der DES-Chip realisiert eine dreistufige Pipeline: Ein Datenblock wird zuerst ins Eingaberegister I geschrieben, wird danach in die Register L und R transferiert, wo der Block die 16 Iterationen der Funktion f durchläuft, und wird schliesslich ins Ausgaberegister O übertragen.

Das Schlüsselmodul beinhaltet das Register MK für die Speicherung des Chiffrier- bzw. Dechiffrierschlüssels und das Register CD, welches für jede der 16 Iterationen einen unterschiedlichen Schlüssel erzeugt. In die Register MK und CD kann von der externen Logik nur geschrieben werden; sie können nicht gelesen werden. Diese Eigenschaft ist wichtig, da die Sicherheit eines Verschlüsselungssystems, welches einen geheimen Schlüssel verwendet, auf die Sicherheit des Schlüssels angewiesen ist. Die Kenntnis des Schlüssels würde es erlauben, Übermittlungen zu entschlüsseln oder gefälschte Mitteilungen ins Netzwerk einzuspeisen.

Die beschriebene Realisierung des DES-Algorithmus unterstützt nebst dem ECB-Betriebsmodus auch den CBC-Modus. Letzterer wird realisiert, indem der Klartext-Block während des Chiffrierens mittels der XOR-Operation zuerst mit dem vorhergehenden, chiffrierten Block verknüpft wird, bevor mit dem eigentlichen Chiffrieren in den Registern L und R begonnen wird. Beim Dechiffrieren muss der dechiffrierte Block zuerst mit dem vorhergehenden chiffrierten Block XOR-verknüpft werden, bevor das endgültige Resultat in das Ausgaberegister O geschrieben werden kann. Beim Dechiffrieren werden deshalb zusätzlich für die Zwischenspeicherung des vorhergehenden chiffrierten Blocks die Pipeline-Register I' und I'' benötigt. Im ECB-Betriebsmodus werden die Blöcke nicht miteinander verknüpft, das heisst die XOR-Gatter sind ausgeschaltet.

Charakteristiken der Implementation

Realisiert wurde die Schaltung mit einem 0,8-µm-Gate-Array vom Typ Fury-VSC15K des Halbleiterherstellers Vitesse [6]. Auf dem 8,1 mm x 7,1 mm grossen Array sind 50 000 Transistoren enthalten, wovon 84% für die Realisierung von insgesamt 480 Flipflops, 2580 Gattern und 8 PLAs[4] benutzt wurden. Die hohe Ausnut-

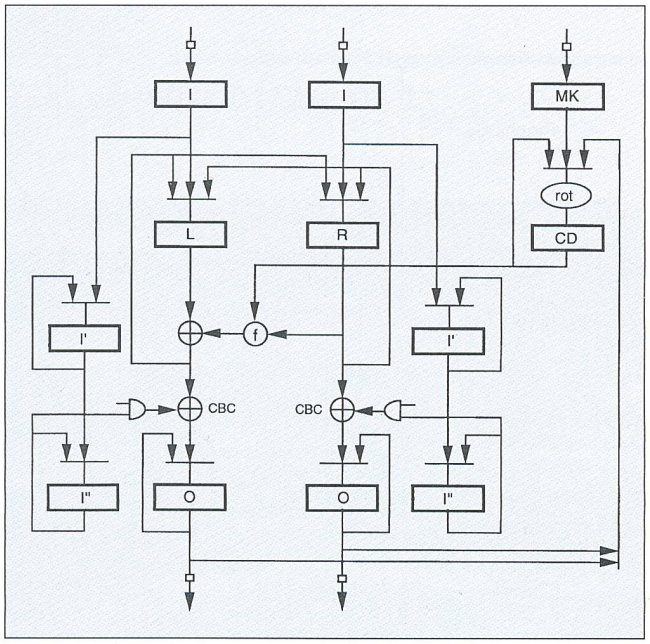
[4] Mittels programmierbaren logischen Feldern (Programmable Logic Array, PLA) können beliebige logische Funktionen realisiert werden.

zung konnte nur durch manuelles Plazieren der Komponenten erreicht werden.

Während der 4 ns langen Taktperiode werden bis zu 10 Logikstufen durchlaufen, was einer Gatterdurchlaufzeit von durchschnittlich 400 ps entspricht.

Die Schnittstelle des DES-Chips ist vollständig asynchron, was den Vorteil bietet, dass keine Synchronisierung mit dem vom DES-Chip benötigten 250-MHz-Takt notwendig ist. Der DES-Chip verfügt über drei Ports: je ein Port für die Dateneingabe und Datenausgabe, beide mit einer Breite von wahlweise 8, 16 oder 32 Bits und ein 7 Bit breiter Port für das Laden des Schlüssels. Mit Ausnahme des 250-MHz-Taktsignals, welches ECL-kompatibel ist, verwenden alle Signale TTL-Levels. Zur Speisung werden –2 V für die GaAs-Logik und +5 V für die TTL-kompatiblen Ausgangstreiber benötigt. Der maximale Stromverbrauch beträgt 8 W.

Bild 6 Dechiffrieren



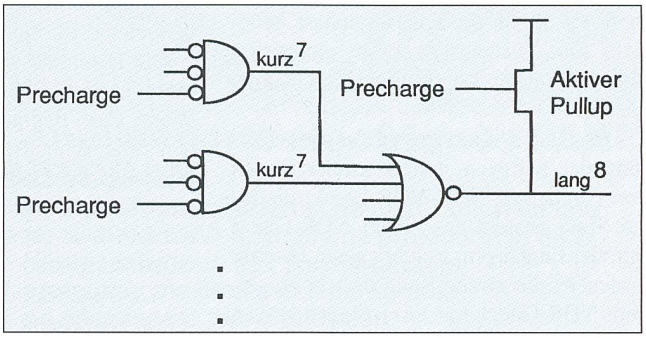
Precharged S-Box

Das Herz des DES-Algorithmus sind die acht in Bild 3 gezeigten S-Boxen. Jede S-Box berechnet eine unterschiedliche Boolesche Funktion und besitzt sechs Eingänge und vier Ausgänge. Die Entwicklung von S-Boxen, welche schnell sein sollen und gleichzeitig wenig Chipfläche beanspruchen dürfen, war der schwierigste und interessanteste Teil der Entwicklungsarbeit. Die naheliegende Struktur für die Realisierung der S-Boxen ist ein PLA. Um den zeitlichen und räumlichen Anforderungen zu genügen, wurden kundenspezifische Schaltungen entwickelt, welche Precharged Logic verwenden.

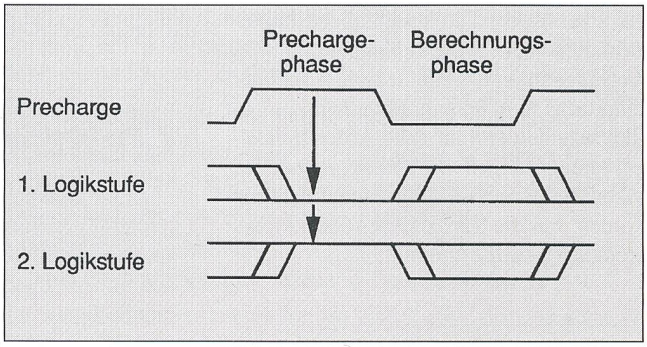
Precharged Logic ist eine bekannte Methode für den Entwurf von Si-nMOS-Schaltungen. Precharged Logic bietet die Integrationsdichte von ungepufferten Gattern und die Geschwindigkeit von gepufferten Gattern. Beim gewählten Gate-Array sind ungepufferte Gatter rund viermal so dicht wie gepufferte Gatter, während gepufferte Gatter bis zu zehnmal schneller sind als ungepufferte Gatter. Mittels Precharging kann nun der Nachteil von ungepufferten Gattern, das heisst die langen Schaltzeiten reduziert werden. Der Grund für die längeren Schaltzeiten von ungepufferten GaAs-DCFL-Gattern ist der schwache Pull-up-Transistor am Gatterausgang, welcher insbesondere bei grossen Lasten[5] lange Anstiegszeiten verursacht.

Das Bild 7a zeigt den Grundbaustein für Precharged NOR-NOR-Logik. Die Gatter der ersten Logikstufe verfügen über einen zusätzlichen Eingang für das Precharge-Signal, während die Gatter der zweiten Stufe einen zusätzlichen aktiven Pull-up am Gatterausgang besitzen[6]. Wie in Bild 7b gezeigt ist, durchläuft die Precharged Logic zwei Phasen: eine Prechargephase und eine Berechnungsphase. Während der Prechargephase, wenn Precharge den Zustand High (H) annimmt, werden die Gatterausgänge der ersten Stufe auf den Zustand Low (L) gezwungen, während die aktiven Pull-ups die Gatterausgänge der zweiten Stufe auf H ziehen. Während der Berechnungsphase, wenn Precharge auf L geht, bleiben die Ausgänge der ersten Stufe entweder auf L oder gehen auf H über, während die Ausgänge der zweiten Stufe auf H bleiben oder auf L übergehen. In einer typischen Schaltung werden mehrere dieser

Bild 7 Aufbau (a) und Logik (b) von Precharged NOR-NOR-Logik



(a)



(b)

[5] Die Last am Ausgang eines Gatters ist grösser, je mehr Gatter angeschlossen sind und je länger der angeschlossene Verbindungsdraht ist. In der besprochenen Technologie wird die Lastgrösse von der Länge der Anschlussdrähte dominiert.

[6] Aktive Pull-ups erlauben, die Gatterausgänge auch bei grösseren Lasten während der Prechargephase von L auf H zu bringen.

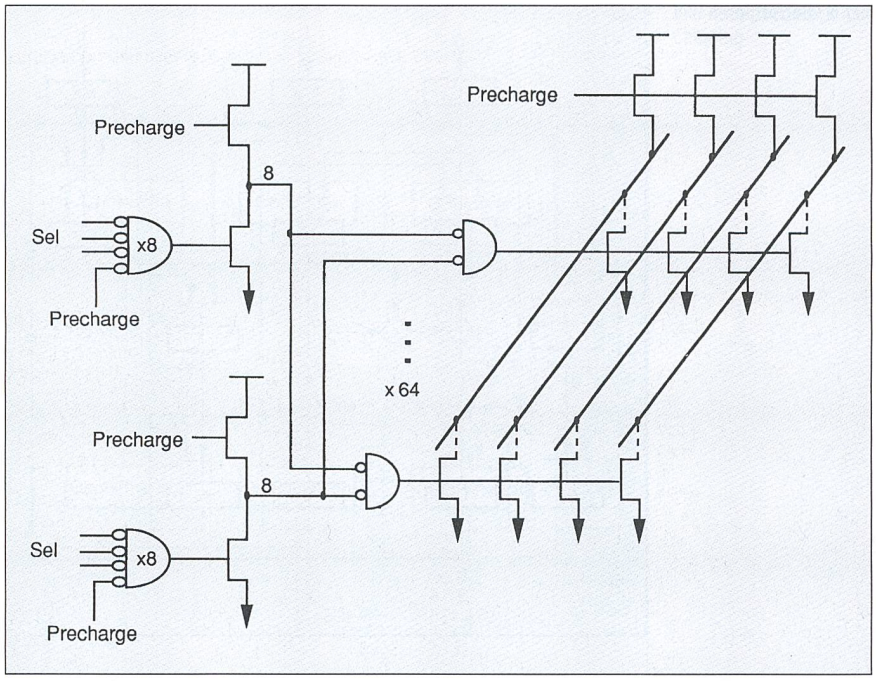


Bild 8 Precharged S-Box

Grundbausteine seriell miteinander verkettet, wobei lange Verbindungsdrähte nur an den Ausgängen der zweiten Logikstufe erlaubt sind[7]. Schaut man sich nun eine solche Kette an, so sieht man, dass die langsamen, steigenden Flanken der Gatterausgänge der zweiten Logikstufe parallel während der Prechargephase auftreten. Während der Berechnungsphase werden an den Ausgängen der zweiten Stufe nur fallende Flanken propagiert, welche wenig Zeit benötigen. Der Nachteil dieser Entwurfsmethode ist die für die Prechargephase benötigte Zeit, welche je besser amortisiert wird, je mehr Logikstufen durchlaufen werden.

Das Bild 8 illustriert die Implementierung der S-Boxen mittels zweier Stufen von Precharged NOR-NOR-Logik: die erste Stufe besteht aus einem NOR-Gatter mit vier Eingängen, welches einen Inverter ansteuert; die zweite Stufe besteht aus einem NOR-Gatter mit zwei Eingängen, an dem bis zu vier Pull-down-Transistoren angeschlossen sind. Der Reihendecoder ist zweistufig realisiert, um Logik und damit Platz zu sparen. Die beschriebene Implementation der S-Boxen mit Hilfe von Precharged Logic nimmt lediglich 10% der Chipfläche in Anspruch. Hätte man die Logikbausteine der vom Gate-Array-Hersteller bereitgestellten Bibliothek verwendet, wären 5,5mal soviel Transistoren benötigt worden, was die Kapazität des Gate-Arrays überstiegen hätte.

[7] Die Drähte an den Ausgängen der ersten Logikstufe müssen kurz gehalten werden, um die Zeit für den Übergang von L nach H während der Berechnungsphase kurz zu halten.

Schlussbetrachtung

Mit einer Chiffrierrate von 1 GBit/s ist der beschriebene Chip um mehr als einen Faktor 5 schneller als sämtliche bekannten Chiffrierchips. Dazu gehören der DES-Chip VM007 von VLSI Technology [7], welcher eine Chiffrierrate von 192 MBit/s aufweist, aber auch Realisierungen anderer Algorithmen wie der IDEA-Chip Vinci, welcher den IDEA-Chiffrieralgorithmus bei einer Chiffrierrate von 178 MBit/s implementiert [8; 9].

Die Datenrate von 1 GBit/s basiert auf einer Taktrate von 250 MHz, welche für Worst Case-Bedingungen berechnet wurden. Die schnellsten Chips, welche mit Messungen gefunden wurden, können mit einer Taktfrequenz von 350 MHz betrieben werden, was einer Chiffrierrate von 1,4 GBit/s entspricht.

Wie dieser Bericht zeigt, ist es möglich, selbst mit einem Semicustom-Chip eine hohe Leistung zu erreichen. Die hohe Effizienz des beschriebenen DES-Chips wurde erzielt, indem der Kern des Algorithmus, das heisst die S-Boxen mit kundenspezifischen PLA-Strukturen implementiert wurden. Damit konnte sowohl eine hohe Dichte als auch eine hohe Geschwindigkeit erreicht werden.

GaAs-DCFL bietet eine Alternative zu den etablierten Si-Technologien wie CMOS und ECL beim Bau von schnellen Schaltungen. Der DES-Chip ist ein Beweis dafür, dass GaAs mittlerweile eine Reife erlangt hat, welche dieses Halbleitermaterial zum Konkurrenten für schnelle VLSI-Schaltungen macht. Ob sich GaAs als VLSI-Alternative für schnelle Anwendungen gegenüber Si behaupten kann, ist jedoch fraglich. Der geringere Leistungsverbrauch von GaAs-DCFL gegenüber Si-ECL und die höhere Geschwindigkeit von GaAs-DCFL gegenüber Si-CMOS fallen vermutlich zu wenig ins Gewicht, um der GaAs-Technologie als VLSI-Prozess zum Durchbruch zu verhelfen. Insbesondere die hohe Dichte von Si-CMOS wird bei den meisten Anwendungen den Geschwindigkeitsvorteil von GaAs-DCFL mehr als wettmachen.

Literatur

[1] *M. Goguen:* AN2 – A Self-configuring Local ATM Network. Proceedings of the National Communications Forum '92. Chicago, Illinois.

[2] *National Bureau of Standards:* Data Encryption Standard. Federal Information Processing Standards Publication FIPS PUB 46-1, Januar 1988 (ersetzt FIPS PUB 46, Januar 1977).

[3] *National Bureau of Standards:* Guidelines for Implementing and Using the NBS Data Encryption Standard. Federal Information Processing Standards Publication FIPS PUB 74, April 1981.

[4] *National Bureau of Standards:* DES Modes of Operation. Federal Information Processing Standards Publication FIPS PUB 81, Dezember 1980.

[5] *D. Hodges, H. Jackson:* Analysis and Design of Digital Integrated Circuits. McGraw-Hill, 1988.

[6] *Vitesse Semiconductor Corporation:* FURY Series Gate Array Design Manual. Version 3.0, Juni 1990.

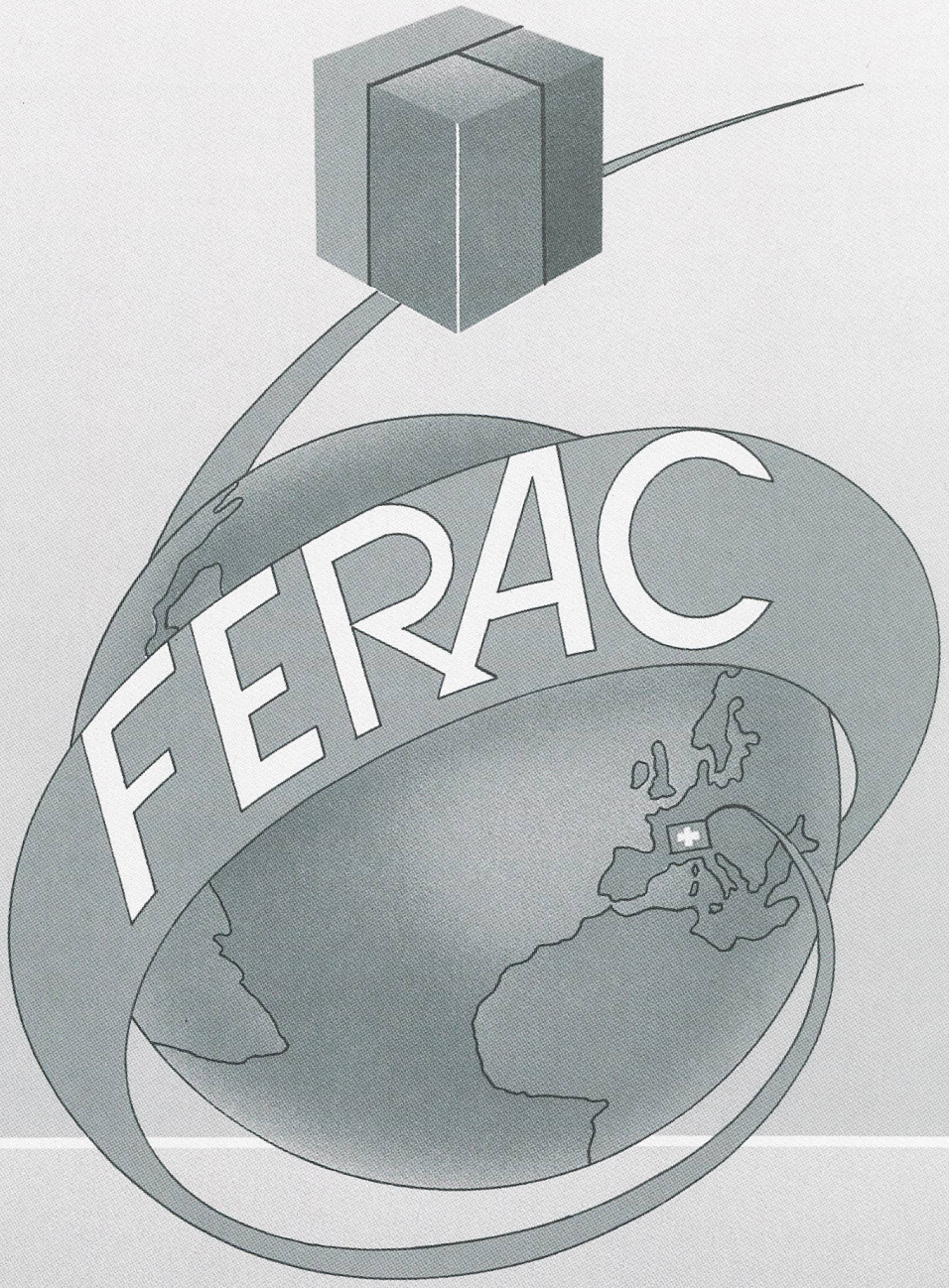
[7] *VLSI Technology:* VM007 Data Encryption Processor. Datenblatt, Oktober 1991.

[8] *A. Curiger:* Eine Alternative zum amerikanischen DES-Chiffrier-Code – Neuer VLSI-Chip für neuen Blockchiffrieralgorithmus. Bulletin SEV/VSE 83(1992)9, S. 41.

[9] *A. Curiger, H. Bonnenberg, R. Zimmermann, N. Felber, H. Kaeslin, W. Fichtner:* VINCI: VLSI Implementation of the New Secret Key Block Cipher IDEA. Proceedings of the IEEE, 1993 Custom Integrated Circuits Conference (CICC '93), S. 15.5.1 bis 15.5.4, San Diego, CA, Mai 1993.

Chiffrement des données à 1 GBit/s

Le chiffrement des données au travers d'un canal rapide comme par exemple dans un réseau d'ordinateurs, nécessite la mise en œuvre d'un puissant hardware de codage. La puce décrite dans cet article chiffre les données selon le Data Encryption Standard (DES) avec une vitesse de 1 GBit/s, ce qui en fait la puce cryptographique la plus rapide connue actuellement. Pour sa réalisation un Gate-Array à l'arséniure de gallium comportant 50 000 transistors a été utilisé.



FERAC SA CH-1527 Villeneuve/Lucens
tél. 037/64 13 34 - fax 037/64 24 43

Import et export de matériel électrique, travaux spéciaux et de montage en électromécanique.

Import und Export von Elektromaterial, Spezialarbeiten, sowie Montagearbeiten für Elektromechanik.

Importing and exporting of electrical equipment, special and electromechanical construction projects.

Importazione ed esportazione di materiale elettrico, lavori speciali e di montaggio nel campo dell'elettromeccanica.

FERAC SA, Pully

Adresse postale: CH-1527 Villeneuve
Tél. 037/64 13 34 - Fax 037/64 24 43