



(19)  
**Bundesrepublik Deutschland**  
**Deutsches Patent- und Markenamt**

(10) **DE 10 2006 037 162 A1 2008.03.27**

(12)

## Offenlegungsschrift

(21) Aktenzeichen: **10 2006 037 162.3**

(22) Anmeldetag: **01.08.2006**

(43) Offenlegungstag: **27.03.2008**

(51) Int Cl.<sup>8</sup>: **H01L 21/822 (2006.01)**

**H01L 21/768 (2006.01)**

**G06F 17/50 (2006.01)**

(71) Anmelder:

**Qimonda AG, 81739 München, DE**

(72) Erfinder:

**Ludwig, Burkhard, Dipl.-Ing., 81379 München, DE;**

**Müller, Uwe, Dr. rer. nat., 80804 München, DE**

(74) Vertreter:

**Maikowski & Ninnemann, Pat.-Anw., 10707 Berlin**

(56) Für die Beurteilung der Patentfähigkeit in Betracht  
gezogene Druckschriften:

**US2003/01 59 120 A1**

**US 52 49 134 A**

**Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen**

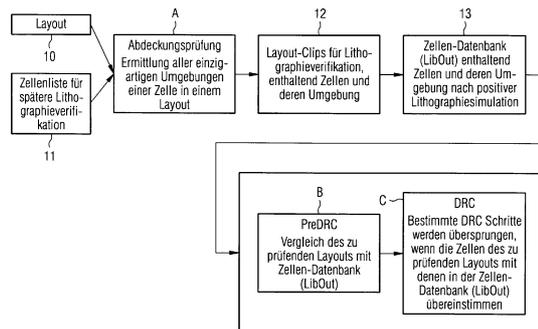
Prüfungsantrag gemäß § 44 PatG ist gestellt.

(54) Bezeichnung: **Verfahren und Vorrichtung und deren Verwendung zur Prüfung des Layouts einer elektronischen Schaltung**

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren und eine Vorrichtung zur Prüfung des Layouts einer elektronischen Schaltung eines Halbleiterbauelementes, insbesondere eines DRAM-Chips, wobei

a) in einer Abdeckungsprüfung (A) mindestens aus einer Zellenliste (11) und mindestens einem Layout (10) automatisch eine Einteilung der Zellen im mindestens einen Layout (10) in Umgebungen der Zellen ermittelt wird, wobei die Umgebungen derart definiert sind, dass Zellen des mindestens einen Layouts (10), deren Schnittstellen zu anderen Zellen identisch sind, einer Umgebung zugeordnet werden, b) die Informationen über die Umgebungen in einer Zellen-Datenbank (LibOut) (13) gespeichert werden und anschließend

c) eine Überprüfung eines zu prüfenden Layouts dadurch erfolgt, dass in einem PreDRC-Schritt (B) ein Vergleich zwischen dem zu prüfenden Layout und Informationen der Zellen-Datenbank (13) erfolgt, wobei automatisch Übereinstimmungen und Unterschiede ermittelt werden.



## Beschreibung

**[0001]** Die Erfindung betrifft ein Verfahren zur Prüfung des Layouts einer elektronischen Schaltung nach dem Oberbegriff des Anspruchs 1 und eine Vorrichtung zur Prüfung einer elektronischen Schaltung nach Anspruch 13 und einer Verwendung nach Anspruch 14.

**[0002]** Am Anfang des Herstellungsprozesses hochintegrierter elektronischer Schaltungen für Halbleiterbauelemente (z.B. DRAM) steht der Layoutentwurf, durch den die Funktionalität der Schaltung bestimmt wird. Durch das Layout wird die physische Umsetzung der Schaltung, z.B. auf einem Silizium-Wafer, festgelegt.

**[0003]** Bekannte Verfahren zur Herstellung der Schaltung auf dem Wafer basieren auf Lithographie, d.h. dass unter anderem das Schaltungslayout erst auf eine Maske und anschließend durch Belichtung der Maske auf dem Wafer abgebildet wird.

**[0004]** Die Herstellung eines Chips erfolgt in vielen Schritten. Jeder der Herstellungsschritte weist eine intrinsische Ungenauigkeit auf, die idealerweise bereits beim Layoutentwurf berücksichtigt werden sollte. Dazu werden Regeln bereitgestellt (so genannte Design- oder Layoutregeln), die beim Layoutentwurf beachtet werden müssen, um eine tatsächliche spätere Herstellbarkeit des Layouts sicherzustellen. Beispielsweise können solche Layoutregeln den Mindestabstand zweier Transistoren oder die Breite von Metallbahnen bestimmen.

**[0005]** Nach Fertigstellung des Layoutentwurfes wird das Layout nach diesen Layoutregeln (DRC design rule check) geprüft und gegebenenfalls korrigiert. Dieser Schritt wird Verifikation genannt und wird durch Computer durchgeführt. Da die Bauelemente einer integrierten Schaltung im Layout durch Polygone dargestellt sind, beziehen sich die Layoutregeln auf die geometrischen Eigenschaften der Polygone sowie deren geometrische Verhältnisse untereinander. Beispielsweise ist eine aus einem p-leitenden und einem n-leitenden Gebiet bestehende Diode im Layout durch aneinander stoßende Rechtecke dargestellt.

**[0006]** Eine Layoutregel kann sich im einfachsten Fall auf ein einziges Maß beziehen. Komplexe Layoutregeln hingegen stellen eine große Anzahl von Maßen zueinander in Beziehung.

**[0007]** Komplexe Layoutregeln sind meist nur unter Verwendung zahlreicher Befehle zu formulieren, wodurch die Rechenzeit in einem unerwünschten Maß zunimmt. Manche Layoutregeln sind gar so formuliert, dass der Befehlssatz der Verifikationssoftware nicht ausreicht und so genannte Dummyfehler (ver-

meintliche Fehler) in Kauf genommen werden müssen. Dummyfehler bedeuten, dass die geometrischen Regeln nicht mehr in allen Bereichen eines Chiplayouts gelten.

**[0008]** Dabei besteht das Problem, dass bei vielen komplexen Prozessen die Anzahl der Dummyfehler so stark ansteigt, dass die „wahren“ Fehler nicht mehr erkannt werden.

**[0009]** Ein weiteres Problem ist, dass in der Praxis die Verifikation eines Layouts in manchen Bereichen mit einer Lithographiesimulation erfolgen muss, da eine Verifikation mit konventionellen DRC nicht mehr ausreicht. Somit werden zwei nicht miteinander integrierte Verfahren für die Verifikation benötigt, was den Verifikationsaufwand in die Höhe treibt. Ferner ist die Korrektheit eines Layouts nicht mehr eindeutig nachweisbar, da unterschiedliche Verifikationen herangezogen werden.

**[0010]** Diese Probleme führen dazu, dass manuelle Inspektionen oder andere „Ausnahmebehandlungen“ vorgenommen werden müssen. Dazu dienen handgezeichnete Waiverebenen oder die Katalogisierung von bekannten Fehlermustern.

**[0011]** Die vorliegende Erfindung hat zur Aufgabe, diese Probleme zumindest abzumildern. Diese Aufgabe wird durch die in den unabhängigen Ansprüchen angegebene Erfindung gelöst. Vorteilhafte Ausgestaltungen der Erfindung sind den Unteransprüchen zu entnehmen.

**[0012]** Die Erfindung wird nachfolgend unter Bezugnahme auf die Figuren der Zeichnungen an mehreren Ausführungsbeispielen näher erläutert. Es zeigen:

**[0013]** [Fig. 1](#) eine Übersicht über den Ablauf einer Ausführungsform des erfindungsgemäßen Verfahrens;

**[0014]** [Fig. 2](#) eine Übersicht über den ersten Programmblock einer Ausführungsform;

**[0015]** [Fig. 3](#) ein Beispiel für ein Layout mit Wiederholungen von Strukturen;

**[0016]** [Fig. 4](#) das Beispiel der [Fig. 4](#) ohne Wiederholungen;

**[0017]** [Fig. 5A](#), [B](#) Darstellungen einer hierarchischen Struktur mit vier Instanzen einer Zelle;

**[0018]** [Fig. 6A](#) bis [Fig. 6E](#) Schritte zur Generierung einer Zellklasse;

**[0019]** [Fig. 7A](#) bis [Fig. 7D](#) Schritte zur Generierung einer Umgebung;

[0020] [Fig. 8A](#) bis [Fig. 8D](#) Beispiele für drei Zellklassen;

[0021] [Fig. 9A](#), B Beispiele für Zellcluster;

[0022] [Fig. 10](#) Beispiel für die Klassen-Repräsentanten;

[0023] [Fig. 11](#) Beispiel für Referenzzelle;

[0024] [Fig. 12](#) Übersicht über den zweiten Programmblock, Pre-DRC;

[0025] [Fig. 13A](#), B Beispiel für Cell-Mismatches (Unterschiede);

[0026] [Fig. 14A](#) bis [Fig. 14D](#) Beispiel für einen Umgebungs-Mismatch;

[0027] [Fig. 15](#) Venn-Diagramm zur Illustration der Matches und Mismatches (Unterschiede);

[0028] [Fig. 16A](#) bis [Fig. 16C](#) Beispiel für eine Abdeckung;

[0029] [Fig. 17A](#) bis [Fig. 17C](#) Beispiel für einen DRC-Schritt nach einem PreDRC-Schritt.

[0030] Im Folgenden wird ohne Einschränkung der Allgemeinheit beispielhaft ein Array als Spezialfall eines Layouts zur Beschreibung verwendet. Unter einem Array wird hier ein sehr regelmäßiges Layout verstanden, wie z.B. ein Speicherfeld eines DRAM-Speicherchips. In [Fig. 3](#) ist ein solches Array dargestellt. Das Verfahren und auch die Vorrichtung sind für Layouts allgemein anwendbar.

[0031] Die hier beschriebenen Ausführungsformen des erfindungsgemäßen Verfahrens gehen von drei Verifikationsschritten A, B, C aus, die im Folgenden detaillierter beschrieben werden.

A) Eine Abdeckungsprüfung prüft, wie weit ein Layout durch Simulationen abgedeckt werden muss. Dabei wird für eine Fehlersuche gespeichert, was simulierbar ist oder simuliert ist. Dazu wird die simulierte Layout Geometrie in einer Zelldatenbank **13** (LibOut) gespeichert.

B) Die Zelldatenbank **13** (LibOut) wird u.a. in einem nachfolgenden Schritt, dem Pre Design-Rule Check (PreDRC) verwendet.

C) Anschließend findet vorteilhafterweise ein konventioneller und/oder auch modifizierter Design Rule Check (DRC) statt.

[0032] Alle drei Schritte (A, B, C) werden im Folgenden detailliert beschrieben. Wesentlich ist vor allem die Kombination der ersten beiden Verifikationsschritte A, B. Bevor die Schritte A, B, C beschrieben werden, wird anhand von [Fig. 1](#) das Zusammenwirken dieser Schritte dargestellt.

[0033] In [Fig. 1](#) ist dargestellt, dass der wesentliche Input für den Schritt A (die Abdeckungsprüfung) das Einlesen eines oder auch mehrerer Layouts **10** und der Zellenliste **11** ist.

[0034] Ziel der Abdeckungsprüfung A ist es, Umgebungen zu finden, in denen Zellen immer wieder vorkommen, so dass die nachfolgenden Schritte vereinfacht werden können.

[0035] Als Output werden Layout-Clips **12** generiert.

[0036] Wichtig für das Folgende ist die Generierung einer Datenbank u.a. mit den Umgebungen der Zellen, der Zellen-Datenbank (LibOut) **13**. Ferner enthält die Zellen-Datenbank (LibOut) **13** Klassen, die bei der Abdeckungsprüfung A ermittelt wurden.

[0037] Diese Zellen-Datenbank (LibOut) **13** wird dann zusammen mit einem zu prüfenden Layout **10'** im PreDRC eingelesen (siehe [Fig. 12](#)). Dabei findet ein Vergleich des zu prüfenden Layouts und der Zellen-Datenbank (LibOut) statt. Das zu prüfende Layout **10'** kann zum Beispiel eine Weiterentwicklung eines bereits früher untersuchten Layouts **10** sein, wobei Teile des früheren Layouts **10** in das zu untersuchende Layout **10'** übernommen wurden.

[0038] Schließlich folgt als vorteilhafter Schritt ein an sich bekannter DRC als Schritt C. Dabei können bestimmte DRC Regeln übersprungen werden, wenn die Zellen im zu untersuchenden Layout **10'** mit denen der Zellen-Datenbank (LibOut) **13** übereinstimmen.

[0039] Dabei wird jeweils vorausgesetzt, dass die Ausführung der Schritte automatisch erfolgt, d.h. durch ein Computerprogramm oder einen Mikroprozessor.

A) Abdeckungsprüfung (siehe [Fig. 2](#))

[0040] Die Abdeckungsprüfung A geht von folgendem Input aus:

- Einer Zellenliste **11**. Die Zellenliste **11** enthält bereits eine Liste von Zellen, die simuliert werden sollen. Zellen, die nur mittels geometrischer Regeln geprüft werden sollen, sind in dieser Zellenliste nicht enthalten.

- Einer Darstellung eines Layouts **10**, z.B. in Form einer DFII Datenbank.

- Dem Namen einer Top-Zelle **15** im Layout **10** und die Bestimmung mindestens eines Check-Lagers **16** (d.h. der Bezeichnung des gerade geprüften Lagers).

[0041] Mit diesem Input werden die folgenden drei Schritte ausgeführt:

1. Bildung einer Abdeckungsliste **21** aus der Zellenliste **11**.

**[0042]** Die Abdeckungsliste **21** wird aus der Zellenliste **11** generiert, aus der die Hüllflächen (bounding boxes) der Zellen bestimmbar sind. Die abgedeckte Fläche im zu simulierenden Lager wird dann aus den Hüllflächen errechnet. Das erhaltene Lager stellt dann diese Teile des gesamten Layouts dar, die durch die Simulation abgedeckt werden und die nicht mittels eines üblichen DRC geprüft werden.

**[0043]** In **Fig. 3** ist eine Übersichtsdarstellung eines Arrays **100** gegeben, bei dem der größte Teil **101** des Arrays für eine Simulation vorgesehen ist. Da das Array sehr regulär aufgebaut ist, d.h. sehr viele Wiederholungen vorkommen, müssen nur bestimmte Zellen des Arrays simuliert werden, beispielsweise Word-Line Driver oder Sense-Amplifier.

**[0044]** In **Fig. 4** ist das Array gemäß **Fig. 3** dargestellt, wobei die Zellinstanzen **101** weggelassen wurden, die nicht mehr simuliert werden müssen, da Wiederholungen vorliegen. Die im Folgenden beschriebenen Schritte dienen u.a. dazu, eine Datenbasis zu erstellen, die diese Wiederholungen aufzeigt und späteren Verfahrensschritten B, C zur Verfügung stellt. Gerade hier zeigen sich die vorteilhaften Wirkungen der vorliegenden Erfindung.

2. Klassifizierung der Zellen (Verfahrensschritt **22** in **Fig. 2**).

**[0045]** Wenn eine Abdeckung durch eine Anzahl von Zellen definiert ist, wird jede Instanz jeder ausgewählten Zelle in ihrer speziellen Umgebung analysiert. Instanzen innerhalb der gleichen Umgebung werden automatisch jeweils einer Zellen-Umgebungsklasse zu geordnet, was im Folgenden einzeln beschrieben wird.

**[0046]** Für jede Zelle in der Zellenliste wird eine Umgebung für jede flache Instanz innerhalb einer Top-Zelle generiert. In **Fig. 5A** ist eine hierarchische Struktur einer beispielhaften Anordnung von Zellen L, C, R, B, F eines Layouts dargestellt. Die Zelle F ist jeweils in Zelle B instantiiert. Die Zellen B sind Parent-Zellen für die Zellen F. Die Zellen C sind Parent-Zellen für die Zellen B. Die TOP Zelle ist die Parent-Zelle für die Zellen L, C und R.

**[0047]** In **Fig. 5B** ist die hierarchische Zellenstruktur in einer anderen Darstellung wiedergegeben, die Verdrahtungen der Zellen deutlich macht. In beiden Fällen gibt es vier Instanzen der Zelle F innerhalb der TOP Hierarchie.

**[0048]** Für die weitere Betrachtung werden die Begriffe des Check-Layers und des Referenz-Lagers eingeführt. Das Check-Lager ist das Lager, für das

eine Lithographie Simulation durchgeführt werden soll.

**[0049]** Referenz-Lager sind Lager, die für die Beurteilung des Simulationsergebnisses notwendig sind. Das Referenz-Lager ist z.B. eine Kontaktlochebene, auf der geprüft wird, ob ein simuliertes Layout tatsächlich auch funktioniert.

**[0050]** In Zusammenhang mit den **Fig. 6A** bis **Fig. 6E** und **Fig. 7A** bis **Fig. 7D** werden die einzelnen Schritte zum Aufbau der Zellenumgebung dargestellt.

**[0051]** **Fig. 6A** stellt die gleiche Struktur wie die **Fig. 5B** dar, wobei das Check-Lager, d.h. die Verbindung zwischen den einzelnen Zellen, dargestellt sind.

**[0052]** In einem ersten Schritt wird für jede Zelle in den Check-Layers und Referenz-Lagers ein Zellenrand (bounding box) bestimmt (in **Fig. 6A** bis **Fig. 6E** beispielhaft fett dargestellte Umrandung für die linke Instanz der Zelle F). Der Zellenrand schließt insbesondere alle Shapes (z.B. einen Kontakt oder eine Leitung) ein, die sich in der Zelle befinden.

**[0053]** Anschließend wird um jeden Zellenrand eine Halo gebildet, indem der Zellenrand um einen vorbestimmten Offset H vergrößert wird. In **Fig. 6A**, **Fig. 6C**, **Fig. 6D** und **Fig. 6E** ist dies durch einen gestrichelten Kasten angedeutet. Der Offset H ist in **Fig. 6A** dargestellt. Das Offset kann z.B. 1 bis 2  $\mu\text{m}$  betragen.

**[0054]** **Fig. 6B** zeigt eine auseinander gezogene Darstellung der einzelnen Zellen mit deren Verbindungen. **Fig. 6C** zeigt die Halo um die erste Instanz der Zelle F zusammen mit inneren Verdrahtungen.

**[0055]** Als Nächstes werden geometrisch benachbarten Zellen automatisch untersucht, indem alle Teile von Shapes, die innerhalb der Halo liegen, der Zelle zugerechnet werden, alle Teile von Shapes, die außerhalb der Halo liegen, werden abgeschnitten. **Fig. 6D** zeigt den Zustand vor dem Abschneiden, **Fig. 6E** den Zustand danach. Die Zelle wird nun nur noch ohne die inneren Verdrahtungen betrachtet (vergleiche z.B. **Fig. 6C** mit **Fig. 6E**), indem diese Verdrahtungen aus der Darstellung entfernt werden.

**[0056]** Dieses Vorgehen wird in Zusammenhang mit den **Fig. 7A**, **Fig. 7B**, **Fig. 7C** und **Fig. 7D** beschrieben. In **Fig. 7A**, **78**, **Fig. 7C**, **Fig. 7D** ist zusätzlich zu dem Check-Lager noch das Referenz-Lager dargestellt.

**[0057]** In **Fig. 7A** ist die gleiche Zelle wie in **Fig. 6C** dargestellt, d.h. einschließlich der inneren Verdrahtungen. In **Fig. 7B** ist die Zelle ohne die inneren Ver-

drahtungen, aber mit den Shapes dargestellt, die sich aus den Überlappungen mit benachbarten Zellen ergeben. In [Fig. 7C](#) sind nur die inneren Verdrahtungen der Zelle dargestellt. In einem ersten Schritt wird ein Clipping am gestrichelten Rechteck der [Fig. 7A](#) und [Fig. 7B](#) durchgeführt, so dass in [Fig. 7D](#) keine Shapes außerhalb des gestrichelten Randes vorkommen. Zieht man die Strukturen der [Fig. 7C](#) in jeder Ebene für sich von dem geclippten Zwischenergebnis ab (logisches UND NICHT), so bleibt eine Zelle ohne innere Verdrahtungen übrig (Es bleibt lediglich ein Kontakt übrig, der durch ein Quadrat dargestellt ist), wobei innerhalb der Halo die Verbindungen noch bestehen bleiben. Die [Fig. 7D](#) entspricht der [Fig. 6E](#).

**[0058]** Diese Schritte werden für alle Referenz-Lagers wiederholt.

**[0059]** Diese reduzierte Repräsentation der Zelle wird als Umgebung bezeichnet.

**[0060]** Diese Umgebungen werden normalisiert, z.B. werden alle Shapes der Umgebung in das Koordinatensystem der Zelle transformiert. Dies geschieht, um geometrische Verhältnisse für eine Klasseneinteilung vergleichbar zu machen. Dazu gehört die Wahl eines geeigneten Koordinatensystems.

**[0061]** Es zeigt sich, dass in einem Array sehr viele Zellen die gleiche Umgebung aufweisen, d.h. die Zelle gemäß [Fig. 6E](#) stellt z.B. einen allgemeinen Typus dar, der auch an anderen Stellen des Arrays vorkommt.

**[0062]** Ausgehend von diesen Umgebungen für jede Instanz werden nun Klassen gebildet. Auf Grund der vorgenommenen Normalisierung sind die Zellen untereinander vergleichbar. Ein Vergleich ermöglicht, dass die flachen Zell-Instanzen in Zellen-Umgebungs-Klassen eingeteilt werden können. Alle Zell-Instanzen einer Klasse haben die gleiche Umgebung (siehe z.B. [Fig. 6E](#)). Umgekehrt haben Zell-Instanzen unterschiedlicher Klassen unterschiedliche Umgebungen.

**[0063]** Die Einteilung der Instanzen in Klassen ist in [Fig. 8A](#), [Fig. 8B](#), [Fig. 8C](#) und [Fig. 8D](#) dargestellt. In [Fig. 8A](#) ist das Layout analog der [Fig. 5A](#) oder [Fig. 6A](#) dargestellt. Hier ist interessant, dass vier Instanzen der Zelle F in drei Klassen eingruppiert werden können. Die beiden mittleren Zellen F ([Fig. 8C](#)) gehören derselben Klasse an.

**[0064]** Für jede Klasse wird ein Klassenrepräsentant **102** im ausgeflachten Layout (siehe [Fig. 10](#) doppelt schraffierte Bereiche) gewählt. Damit kann eine Klasse von Zellen gezielt an einem Beispiel simuliert werden, was Zeitersparnis bringt.

### 3. Bestimmung von Ausschnitten, insbesondere Clustern, im Array

**[0065]** Zur Vereinfachung der Simulation ist es vorteilhaft, mehrere Zellen zusammenzufassen. Wie zuvor beschrieben, werden dazu mehrere Klassenrepräsentanten ausgewählt. Einander räumlich benachbarte Klassenrepräsentanten **102** werden zu Clustern **103** (einfach schraffierte Felder in [Fig. 10](#)) zusammengefasst, so dass diese Cluster als Ganzes einer Simulation unterzogen werden können.

**[0066]** Wenn zwei Klassenrepräsentanten **102** sich überlappen oder einen geringeren Abstand haben, als ein Offset einer Halo, werden sie zu einem Cluster zusammengefasst. Anders formuliert, zwei Instanzen von unterschiedlichen Clustern **103** werden immer mindestens durch einen Halo Offset voneinander getrennt.

**[0067]** In [Fig. 9A](#) werden die drei Zellrepräsentanten **102** (siehe [Fig. 8B](#), [Fig. 8C](#), [Fig. 8D](#)) zu einem Cluster zusammengefasst.

**[0068]** In [Fig. 9B](#) wird ein anderes Beispiel für einen Cluster angegeben, der durch einen bestimmten Abstand der Halo charakterisiert wird.

**[0069]** In [Fig. 10](#) ist die gleiche Ansicht wie in [Fig. 3](#) und [Fig. 4](#) dargestellt, wobei die Cluster **103** (schraffierte Felder) und die Klassenrepräsentanten **102** (doppelt schraffierte Felder) dargestellt sind.

**[0070]** Jede Klassen-Zelle enthält die Geometrie der entsprechenden Klasse, die Zellenbegrenzung (ein Rechteck) und eine Instanz einer Referenzzelle der klassifizierten Zelle (automatische Auswahl). Diese Instanz wird an den Ursprung der klassifizierten Zelle gesetzt ([Fig. 11](#)).

**[0071]** Am Ende der drei Schritte wird das Ergebnis automatisch in der Klassen-Datenbank (LibOut) **13** gespeichert. Die Klassen-Datenbank (LibOut) **13** enthält neben Informationen über die ermittelten Zellenklassen auch Simulationsdaten. Die für die Ausführungsformen der Erfindung wesentlichen Zellenklassen beschreiben alle Umgebungs-klassen, die nach dem oben beschriebenen Verfahren ermittelt wurden.

**[0072]** Es ist für die hier beschriebene Ausführungsform der Erfindung wesentlich, dass ein zu prüfendes Layout **10'** unter Verwendung der Klassen-Datenbank **13** (LibOut) (das Daten auch über andere Layouts enthält) einer Überprüfung unterzogen wird, die dem eigentlichen DRC vorgelagert ist (siehe [Fig. 1](#)); dem Pre-DRC (siehe [Fig. 12](#)).

**[0073]** Der Pre-DRC (Verfahrensschritt B in [Fig. 1](#)) umfasst vorteilhafterweise folgende Schritte (siehe [Fig. 12](#)):

a) Auslesen **31** der Klassen-Datenbank (LibOut) **13**. Die Klassen-Datenbank (LibOut) **13** wird als Referenz-Layout **17** für die Überprüfung verwendet. Ausgelesen werden ferner der Name der Top-Zelle **15** und die Zellenliste **11**, die in einem früheren Lauf mit einem der vorgehenden Schritte (**Fig. 2** bis **Fig. 10**) ermittelt worden waren. Auch erfolgt ein Einlesen des zu untersuchenden Layouts **10'** eines Chips.

b) Automatisches Vergleichen **32** jeder Referenzzelle in der Klassen-Datenbank (LibOut) **13** mit der entsprechenden Zelle auf dem zu untersuchenden Layout **10'** des Chip. Die Referenzzelle dient dazu, eine Vorlage für den Vergleich zu haben, wobei eine beim Vergleich ermittelte Abweichung als Fehler aufgefasst wird. Ein Beispiel ist, dass als Referenz-Layout **10** (in der LibOut enthalten) ein Speicherbaustein für 512 MB verwendet wird, wobei das zu untersuchende Layout **10'** ein 1 GB Layout aufweist. Auch wenn die beiden Layouts **10**, **10'** sehr unterschiedlich sind, findet sich doch eine große Übereinstimmung einzelner Zellenklassen, so dass die Simulation des 1 GB Layouts vereinfacht wird. Durch die Verwendung des PreDRC kann der Aufwand erheblich verringert werden.

c) Automatische Ermittlung **33** neuer Klassen für alle Zellen in der Zellenliste **11** mit dem Chip als Topzelle. Für die Ermittlung der Klassen werden die gleichen Methoden verwendet, die oben u.a. im Zusammenhang mit **Fig. 8** beschrieben wurden. Dabei geht der PreDRC von den bekannten Zellenklassen aus.

d) Automatische Ermittlung von Unterschieden **35** zwischen der Klassen-Datenbank (LibOut) **13** (d.h. des Referenz-Layouts), der Zellenliste **11** und dem zu untersuchenden Chip-Layout **10'**. Die Unterschiede (wie z.B. im obigen Beispiel erläutert) werden anhand der **Fig. 13**, **14** und **Fig. 15** näher beschrieben.

e) Automatische Bestimmung einer Abdeckungsfläche **34** (siehe z.B. **Fig. 16**).

**[0074]** Der Schritt e) kann auch vor Schritt d) ausgeführt werden. Alternativ können beide Schritte auch parallel ausgeführt werden.

**[0075]** Im Folgenden wird auf drei Unterschiede (mismatches) eingegangen, die durch den PreDRC ermittelt werden.

**[0076]** In **Fig. 13** ist ein Zellen-Layout Unterschied dargestellt. Ein Zellen-Layout Unterschied liegt vor, wenn sich die Geometrie des Zellen-Layouts auf dem Chip von der Geometrie der Referenzzelle (**Fig. 11**) in der Zellen-Datenbank (LibOut) **13** unterscheidet. Wenn ein solcher Unterschied auftritt, bedeutet dies, dass die Zellen-Datenbank nicht das gleiche Array/Layout beschreibt, das untersucht wird; es muss etwas korrigiert werden.

**[0077]** Der Unterschied zwischen den Layouts **10**, **10'** werden z.B. mittels eines EXOR Schrittes bestimmt.

**[0078]** In **Fig. 13A** ist dargestellt, dass in der Zellen-Datenbank (LibOut) eine Referenzkopie einer Zelle F enthalten ist. In **Fig. 13B** sind zwei Zellen dargestellt, die sich in dem zu untersuchenden Layout **10'** finden. Beide unterscheiden sich in ihrer Geometrie von der Referenzkopie, was zu einem Unterschied führt, der durch das PreDRC ermittelt wird.

**[0079]** Eine andere Form eines zu ermittelnden Unterschiedes wird in **Fig. 14** dargestellt. Dabei geht es um einen Unterschied in den Umgebungen. Dies tritt auf, wenn bei der Untersuchung des Layouts **10'** eine Klasse ermittelt wird, die nicht mit einer Klasse in der Zellen-Datenbank (LibOut) **13** übereinstimmt.

**[0080]** In **Fig. 14A** ist das Layout **10** dargestellt, das zu der Klassenbildung in der Zellen-Datenbank (LibOut) geführt hat (**Fig. 14B**). Es handelt sich um das Beispiel, das bereits in Zusammenhang mit **Fig. 8** dargestellt worden war.

**[0081]** Bei der Analyse des zu untersuchenden Layouts **10'** des Chips (**Fig. 14C**) stellt sich heraus, dass eine Umgebung nicht mit einer Umgebung in der Zellen-Datenbank (LibOut) **13** übereinstimmt. Dies ist eine Instanz einer neuen Klasse. Wie oben erläutert, wird ein Unterschied hier als Fehler gewertet, so dass eine gesonderte Prüfung notwendig ist. Dies ist ein wesentliches Ergebnis der Ausführungsform des erfindungsgemäßen Verfahrens.

**[0082]** Ein weiterer Unterschied, der durch den PreDRC ermittelt wird, sind Unterschiede in den Zellenbezeichnungen, wobei die Unterschiede zwischen dem zu prüfen Layout und der Zellenliste und zwischen der Zellen-Datenbank (LibOut) und der Zellenliste bestehen können.

**[0083]** Die erste Menge **41** enthält alle Zellennamen der Zellenliste, die zweite Menge **42** enthält alle Zellennamen der Zellen-Datenbank (LibOut), die dritte Menge **43** enthält alle Zellennamen des zu untersuchenden Layouts.

**[0084]** Eine vierte Menge **44** enthält alle Zellen, die zwar in der Zellenliste genannt sind, aber nicht im zu untersuchenden Layout **10'** vorkommen.

**[0085]** Eine fünfte Menge **45** enthält alle Zellen, die zwar in der LibOut enthalten sind, nicht aber in der Zellenliste.

**[0086]** Eine sechste Menge **46** enthält alle Zellen, die zwar in dem zu untersuchenden Layout **10'** und der Zellenliste enthalten sind, nicht aber in der LibOut.

[0087] Die vierte, fünfte und sechste Menge **44**, **45**, **46** werden automatisch als Unterschiede ermittelt und daher wie Fehler behandelt, die eine Intervention eines Benutzers erfordern.

[0088] Die Zellen, die zwar in der LibOut **13** vorkommen, nicht aber in der Zellenliste, werden in dieser Ausführungsform nicht behandelt.

[0089] In [Fig. 16A](#) bis [Fig. 16C](#) wird dargestellt, wie sich die Ermittlung einer neuen Klasse (d.h. eines Unterschiedes zwischen dem vorab gespeicherten Layout **10** in der Zellen-Datenbank LibOut **13** und dem zu untersuchenden Layout **10'**) auf die Bildung einer Abdeckung auswirkt.

[0090] In [Fig. 16A](#) ist grundsätzlich die gleiche Situation mit vier Instanzen der Zelle F dargestellt, wie in [Fig. 5](#). Allerdings ist die Umgebung der zweiten Zelle F von rechts gegenüber der Vorlage modifiziert, da ein rechteckiges Shape **200** (eingekreist) vorkommt. Dies führt zu einer neuen Klasse, die in der Zellen-Datenbank LibOut **13** nicht enthalten ist.

[0091] In [Fig. 16B](#) ist die Abdeckung **201** (schraffiert) dargestellt, dies sich aus der Abdeckungsprüfung ([Fig. 1](#), Schritt A) ergeben hat). Der Unterschied hat die Folge, dass die Abdeckung **202** für das geänderte Layout in [Fig. 16C](#) anders ist, d.h. die geänderte Zelle fällt nach dem PreDRC ([Fig. 1](#), Schritt B) aus der Abdeckung **202** heraus, da sie nicht mit dem Bekannten übereinstimmt. Wie oben dargestellt, wird im PreDRC grundsätzlich der gleiche Klassenbildungsmechanismus ausgeführt, so dass die geänderte Situation zu einer neuen Klasse führt.

[0092] Im Anschluss an den PreDRC-Schritt (B) erfolgt ein an sich bekannter Design-Rule Check (C), wobei Modifikationen vorgenommen werden können (siehe [Fig. 1](#), 17).

[0093] Anhand von [Fig. 17A](#) bis [Fig. 17C](#) wird deutlich gemacht, welche Auswirkungen die effiziente Gestaltung einer Abdeckung bei einem darauf folgenden DRC-Schritt hat.

[0094] In [Fig. 17A](#) ist schematisch ein Ausschnitt eines zu prüfenden Layouts **10'** dargestellt. Innerhalb des regelmäßigen zu untersuchenden Patterns des Layouts **10'** liegen Bereiche, die als Fehler **400** identifiziert wurden. Zwei Teile des zu untersuchenden Layouts **10'** werden durch Abdeckungen **201** erfasst, die nach den oben beschriebenen Verfahren ermittelt wurden.

[0095] In [Fig. 17B](#) ist dargestellt, dass ein normaler DRC, d.h. ohne Abdeckungen, alle Fehler, d.h. auch sehr viele Dummy-Errors, berücksichtigen müsste.

[0096] Verwendet man hingegen die Abdeckungen

**201**, so wird anhand von [Fig. 17C](#) deutlich, dass Bereiche, deren Dummy-Fehler nicht mehr untersucht werden müssen, da sie von den Abdeckungen **201** erfasst sind, deutlich kleiner sind. Die hier dargestellte Ausführungsform geht davon aus, dass alle Dummy-Fehler, die sich einer Abdeckung zuordnen lassen, herausfallen, so dass sogar Dummy-Fehler, die zwischen den Abdeckungen **201** liegen, herausfallen.

[0097] Die Erfindung beschränkt sich in ihrer Ausführung nicht auf die vorstehend angegebenen bevorzugten Ausführungsbeispiele. Vielmehr ist eine Anzahl von Varianten denkbar, die von dem erfindungsgemäßen Verfahren und der erfindungsgemäßen Vorrichtung auch bei grundsätzlich anders gearbeteten Ausführungen Gebrauch machen.

### Patentansprüche

1. Verfahren zur Prüfung des Layouts einer elektronischen Schaltung eines Halbleiterbauelementes, insbesondere eines DRAM-Chips, wobei
  - a) in einer Abdeckungsprüfung (A) mindestens aus einer Zellenliste (**11**) und mindestens einem Layout (**10**) automatisch eine Einteilung der Zellen im mindestens einen Layout (**10**) in Umgebungen der Zellen ermittelt wird, wobei die Umgebungen derart definiert sind, dass Zellen des mindestens einen Layouts (**10**), deren Schnittstellen zu anderen Zellen identisch sind, einer Umgebung zugeordnet werden,
  - b) die Informationen über die Umgebungen in einer Zellen-Datenbank (LibOut) (**13**) gespeichert werden und anschließend
  - c) eine Überprüfung eines zu prüfenden Layouts dadurch erfolgt, dass in einem PreDRC-Schritt (B) ein Vergleich zwischen dem zu prüfenden Layout (**10'**) und Informationen der Zellen-Datenbank (**13**) erfolgt, wobei automatisch Übereinstimmungen und Unterschiede ermittelt werden.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass nach der Überprüfung des zu prüfenden Layouts ein DRC-Schritt (C) durchgeführt wird.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass in der Abdeckungsprüfung (A) automatisch eine Zuordnung von Zellen gleicher Umgebung in Klassen erfolgt.
4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass die Informationen betreffend die Klassen in der Zellen-Datenbank (**13**) gespeichert werden.
5. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Umgebung einer Zelle dadurch bestimmt wird, dass der Zellenrand um einen bestimmten Betrag (H) erweitert wird, um eine Halo um den Zellenrand zu bilden.

6. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass für jede Zellenklasse ein Zellenrepräsentant (**102**) in einem zu untersuchenden Layout (**10'**) bestimmt wird.

7. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass räumlich benachbarte Zellenrepräsentanten (**102**) zu einem Cluster (**103**) zusammengefasst werden.

8. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass im Rahmen des PreDRC (B) automatisch mindestens ein Unterschied bezüglich einer in der Zellen-Datenbank (**13**) gespeicherten Umgebung und mindestens einer Umgebung im zu untersuchenden Layout (**10'**) ermittelt wird.

9. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass im Rahmen des PreDRC (B) automatisch mindestens ein Unterschied bezüglich eines in der Zellen-Datenbank (**13**) gespeicherten Zellen-Layouts und mindestens eines Zellen-Layouts im zu untersuchenden Layout (**10'**) ermittelt wird.

10. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass im Rahmen des PreDRC (B) automatisch mindestens ein Unterschied bezüglich eines in der Zellen-Datenbank (**13**) gespeicherten Zellen-Namens und mindestens eines Zellen-Namens im zu untersuchenden Layout (**10'**) ermittelt wird.

11. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Zellen-Datenbank (**13**) als selbstlernende Datenbank ausgebildet ist, bei der insbesondere Informationen über Umgebungen von Zellen gelernt werden.

12. Verfahren nach mindestens einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass in einem DRC-Schritt (C) Unterschiede zwischen einem Referenz-Layout (**10'**) und/oder der Zellen-Datenbank LibOut (**13**) einer Abdeckung (**102**) zugeordnet werden.

13. Vorrichtung zur Prüfung des Layouts einer elektronischen Schaltung eines Halbleiterbauelementes, insbesondere eines DRAM-Chips, mit  
a) einem Mittel für eine Abdeckungsprüfung (A), bei der mindestens aus einer Zellenliste (**11**) und aus mindestens einem Layout (**10**) automatisch eine Einteilung der Zellen im mindestens einen Layout (**10**) in Umgebungen der Zellen ermittelbar ist, wobei die Umgebungen derart definiert sind, dass Zellen des mindestens einen Layouts (**10**), deren Schnittstellen

zu anderen Zellen identisch sind, einer Umgebung zuordbar sind, und

b) einer Zellen-Datenbank (LibOut)(**13**), die Informationen über die Umgebungen der Zellen speichert und

c) einem Vergleichsmittel zur Überprüfung eines zu untersuchenden Layouts (**10'**) in einem PreDRC-Schritt (B), wobei ein Vergleich zwischen dem zu untersuchenden Layout (**10'**) und Informationen der Zellen-Datenbank (**13**) erfolgt, wobei vom Vergleichsmittel automatisch Übereinstimmungen und Unterschiede ermittelt werden.

14. Verwendung des Verfahrens gemäß Anspruch 1 oder der Vorrichtung gemäß Anspruch 13 zur Reduzierung des Rechenaufwandes beim Entwurf des Layouts einer elektronischen Schaltung, insbesondere eines DRAM-Chips.

Es folgen 17 Blatt Zeichnungen

Anhängende Zeichnungen

FIG 1

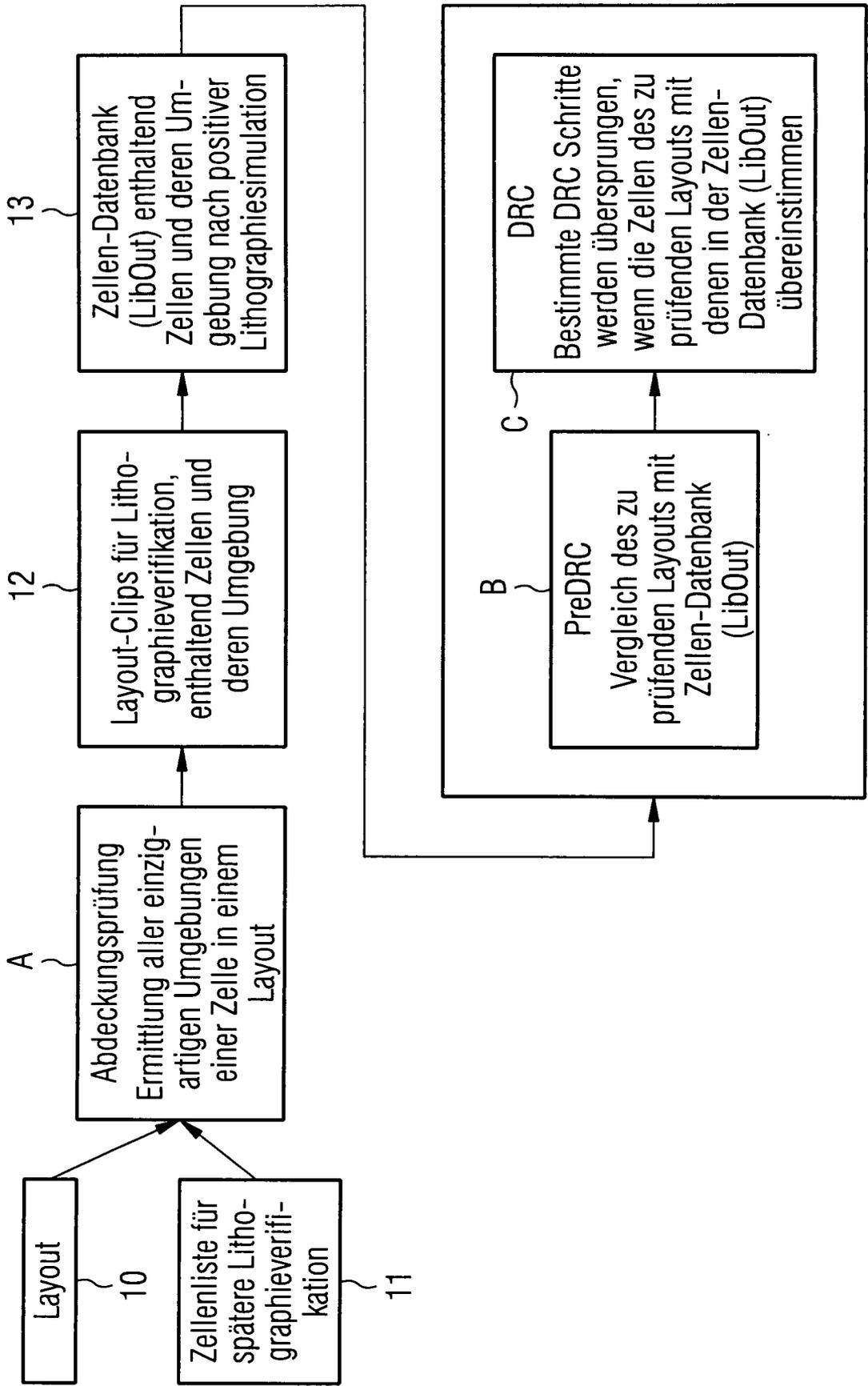
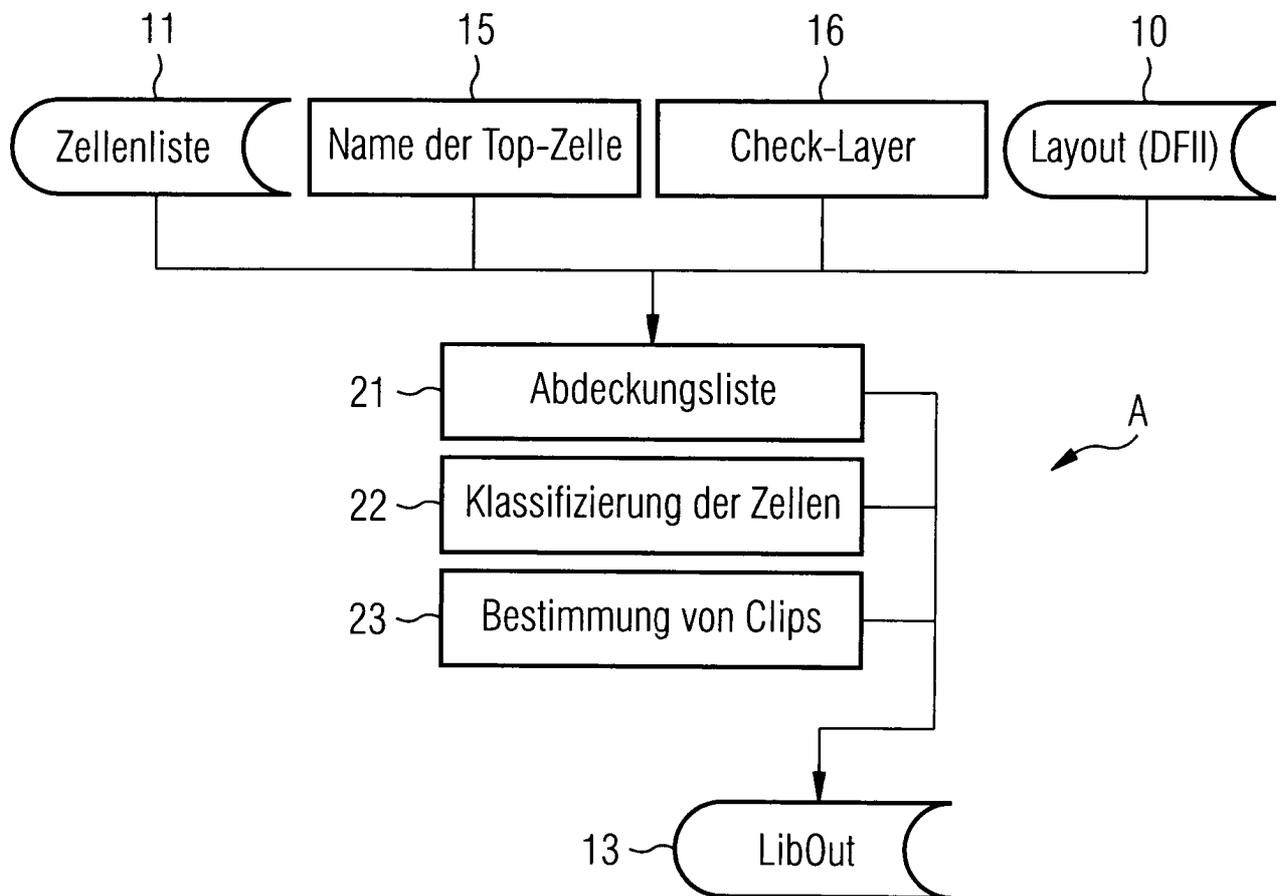


FIG 2



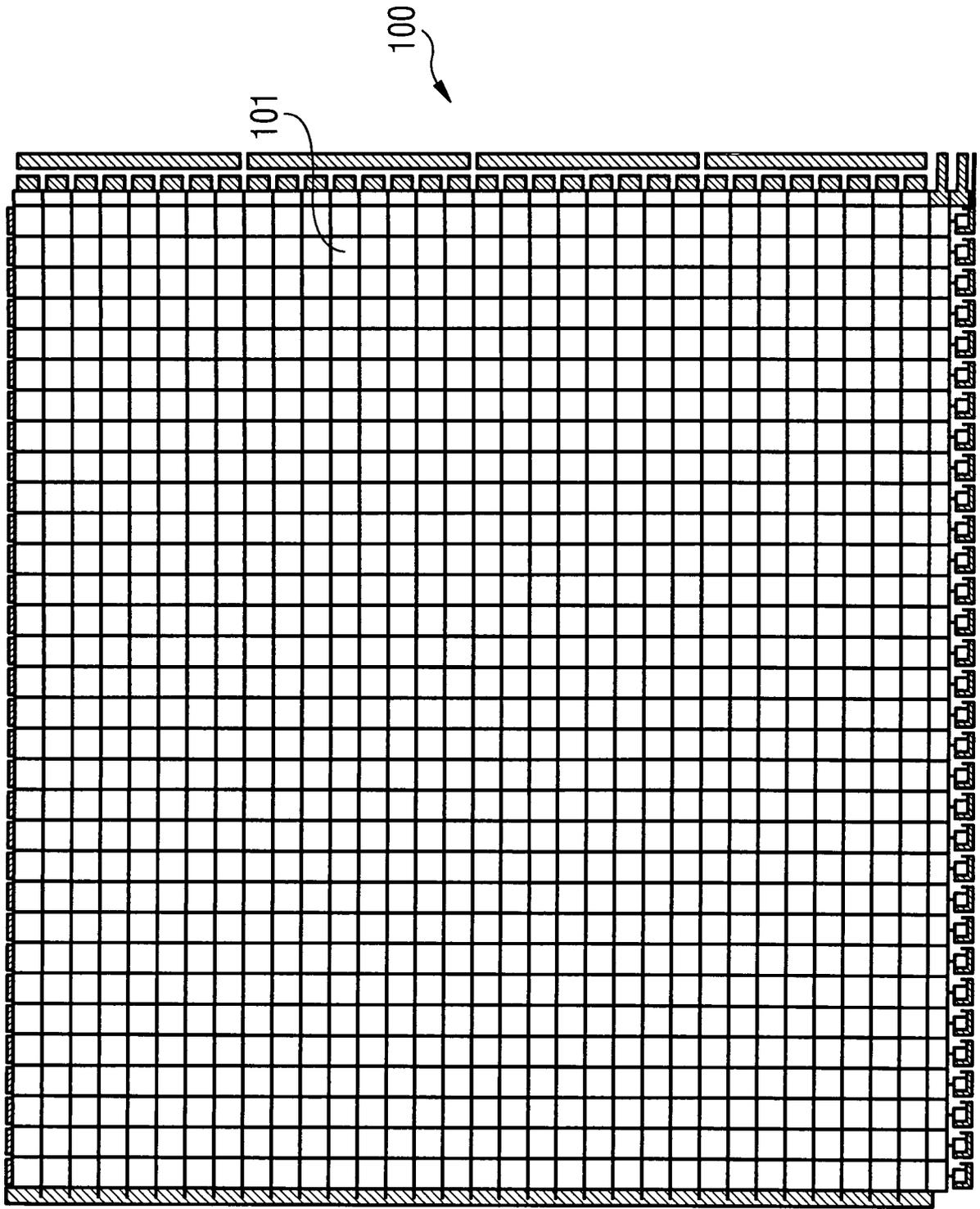


FIG 3

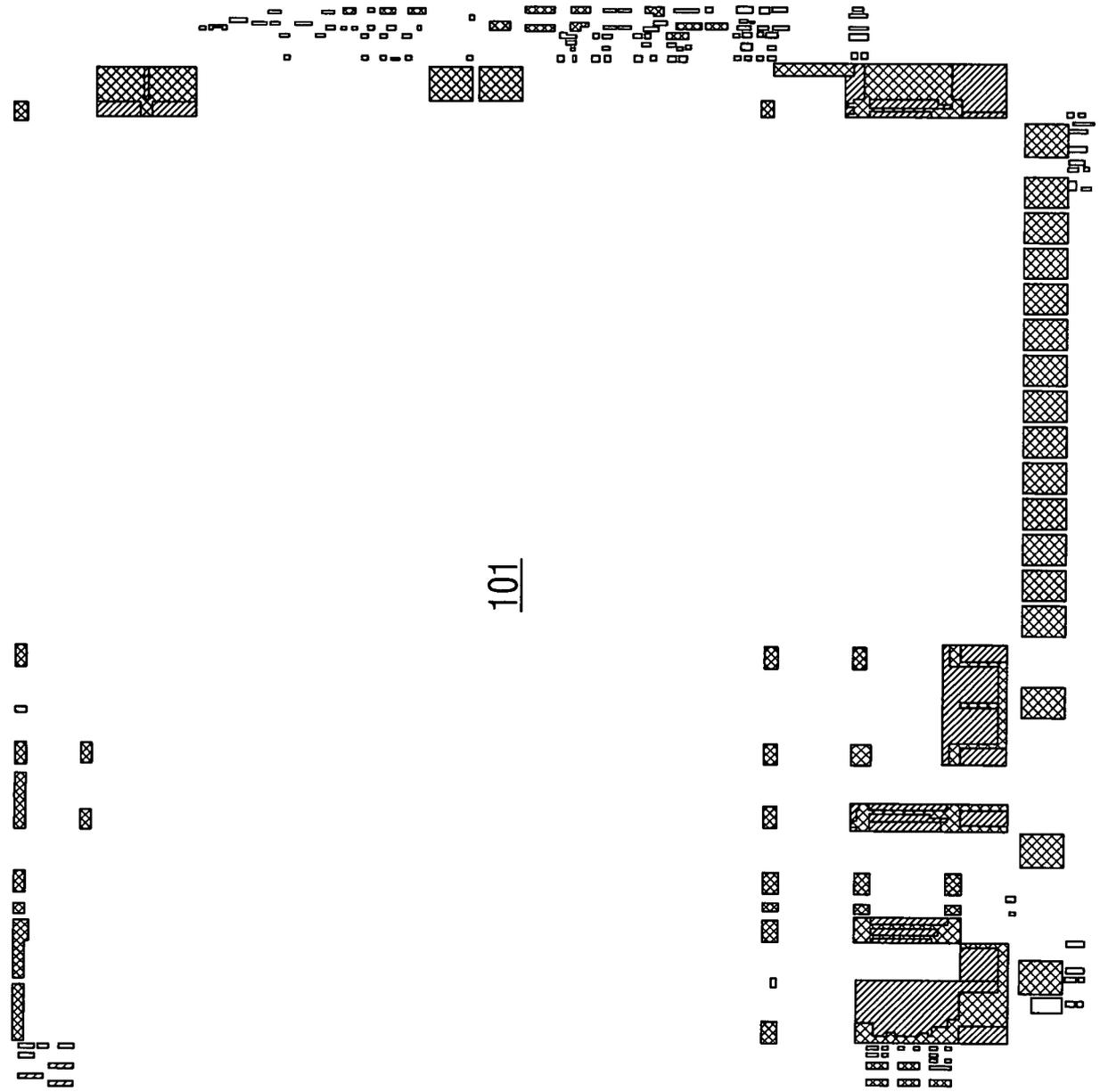
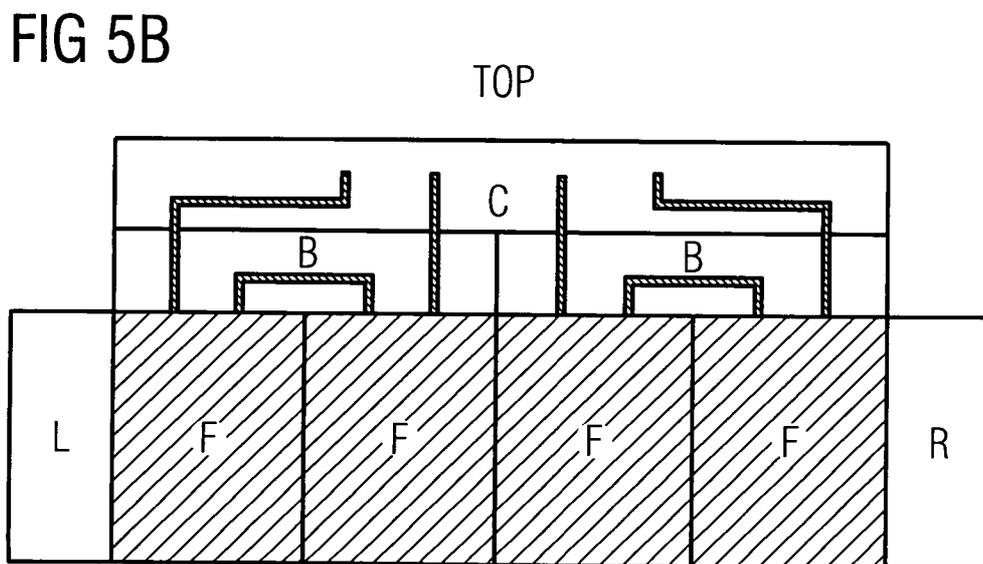
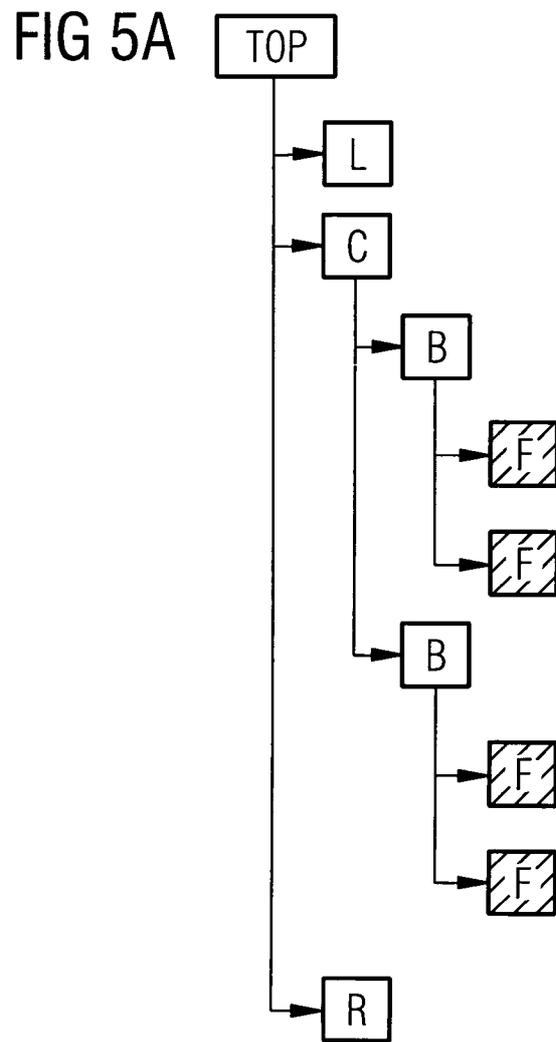


FIG 4



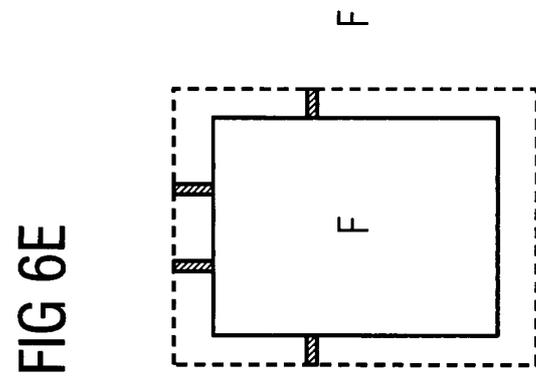


FIG 6E

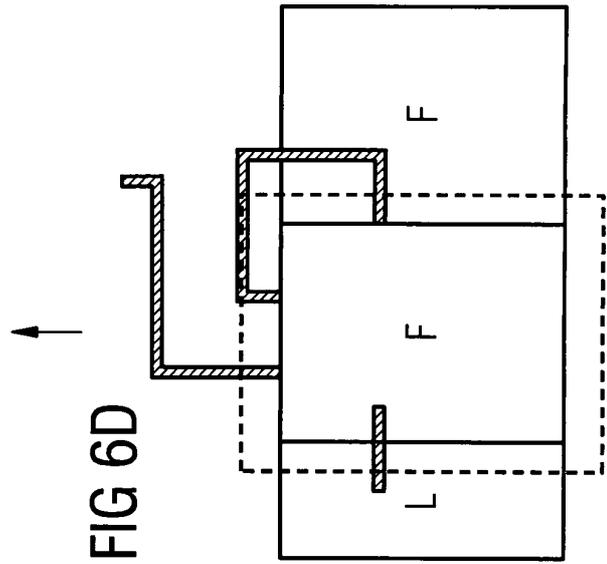


FIG 6D

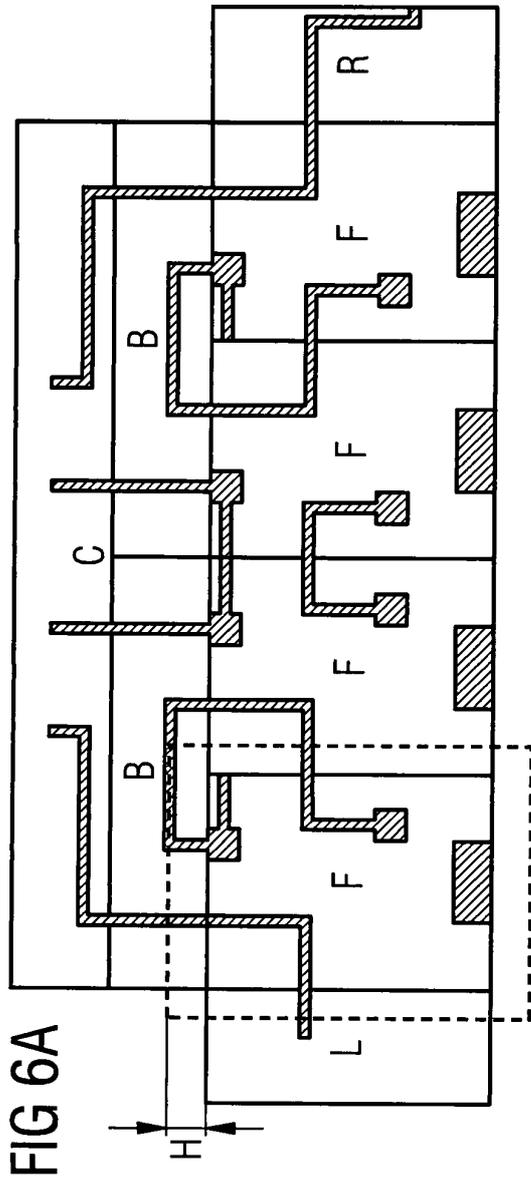


FIG 6A

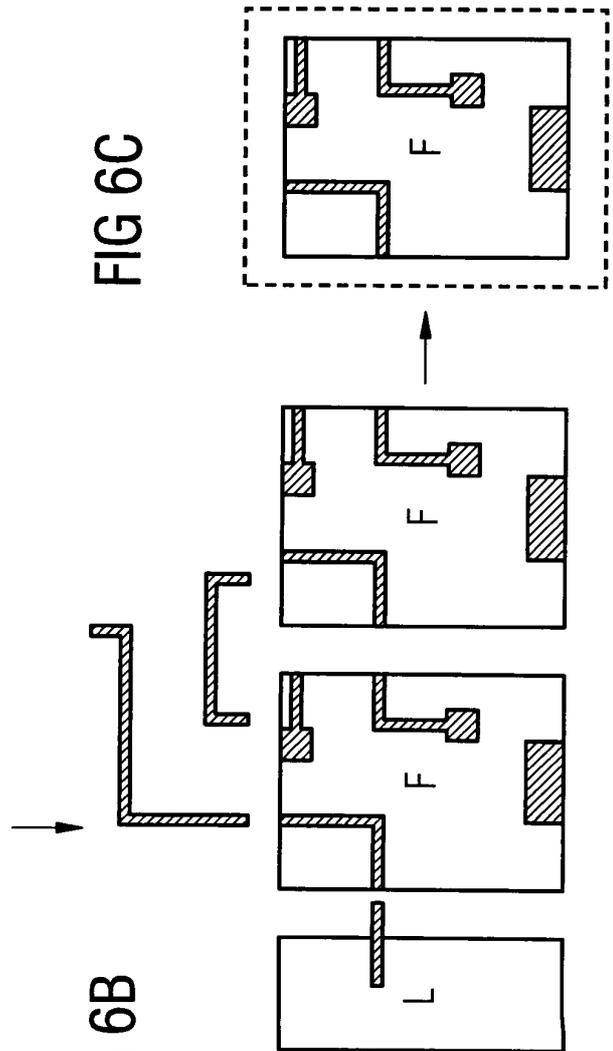


FIG 6B

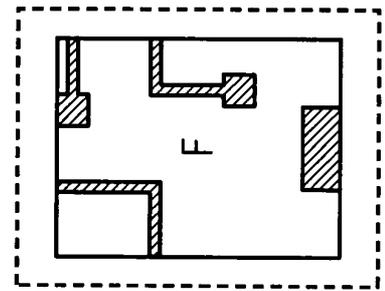


FIG 6C

FIG 7A

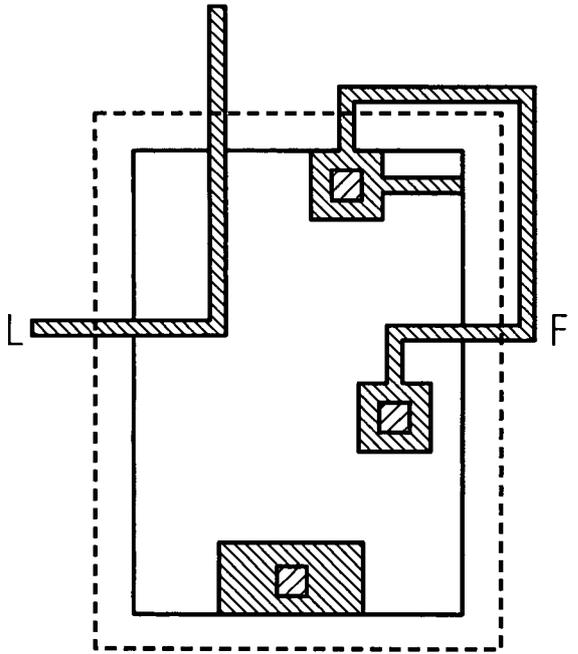


FIG 7B

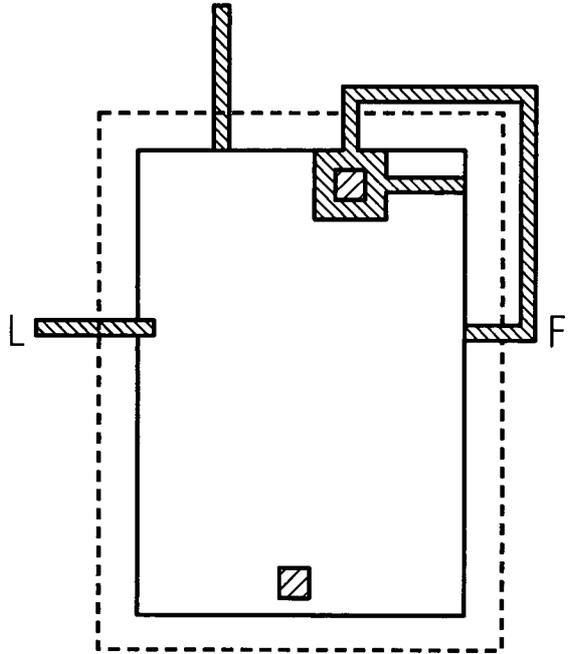


FIG 7C

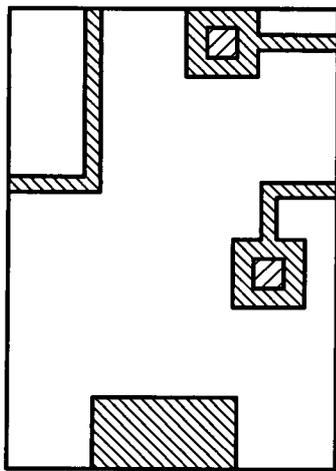
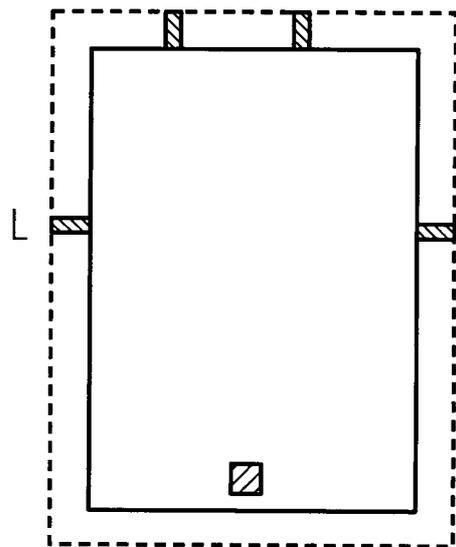


FIG 7D



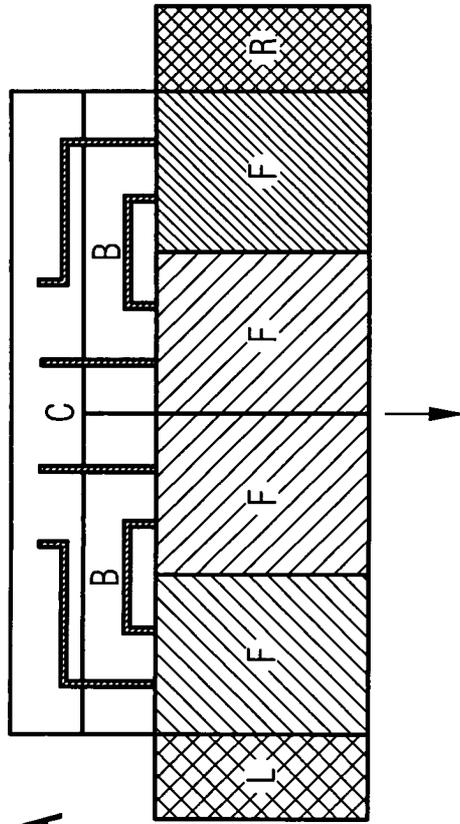


FIG 8A

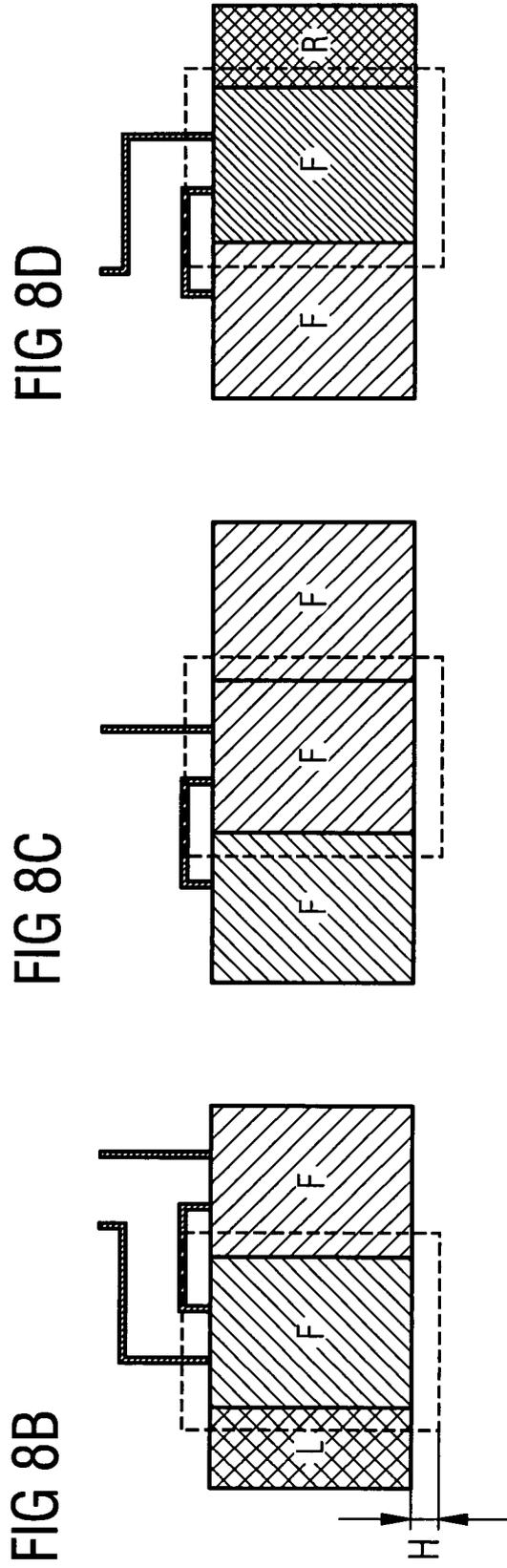


FIG 8B

FIG 8C

FIG 8D

FIG 9A

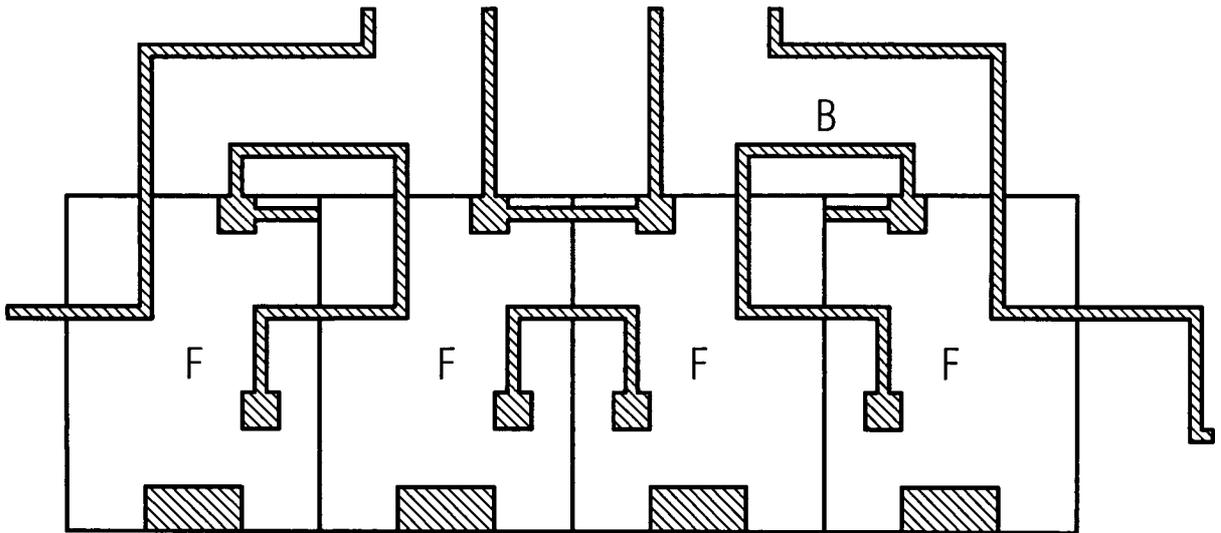
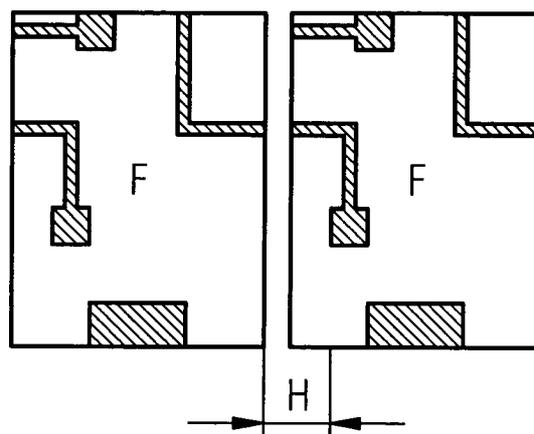


FIG 9B



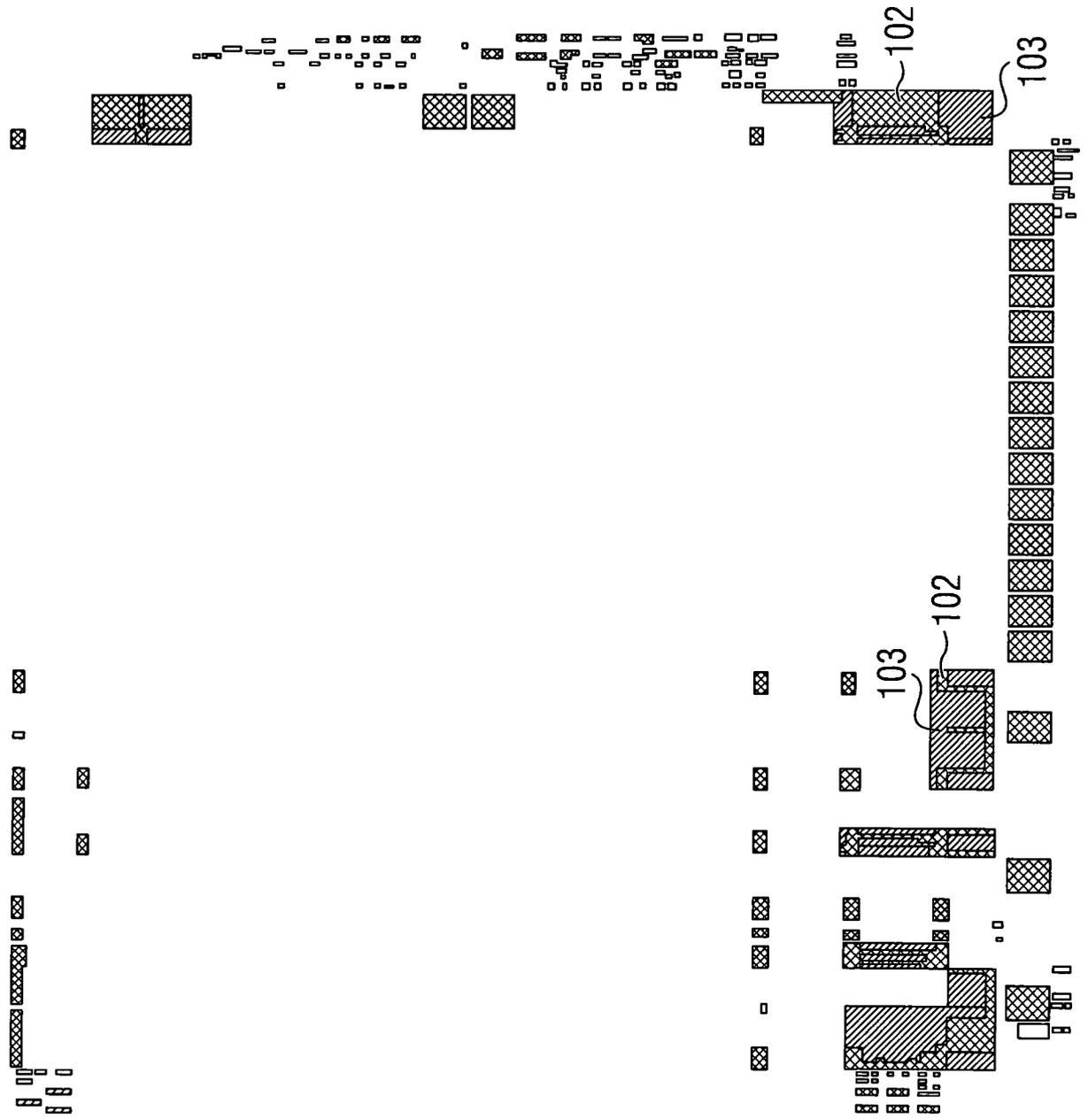
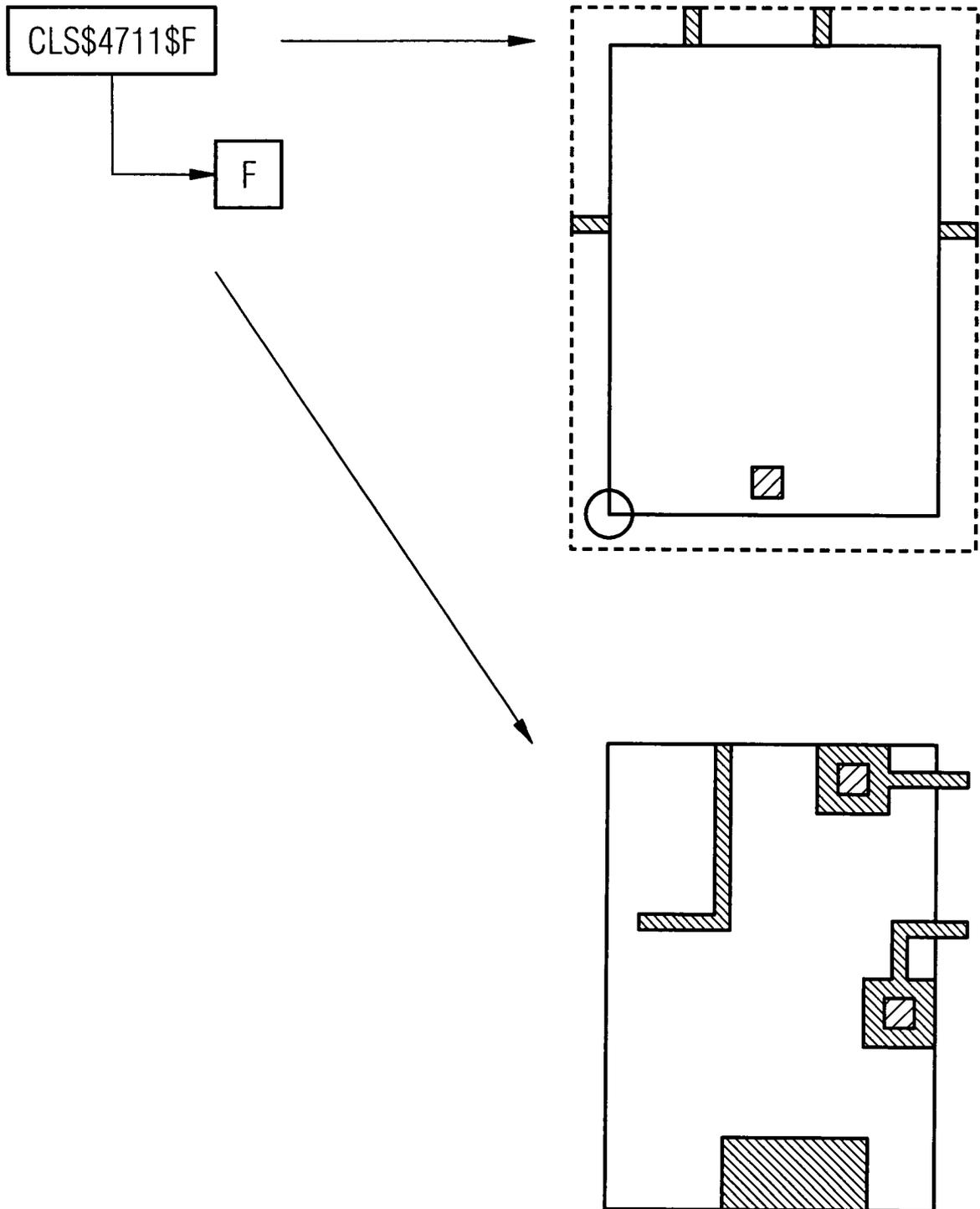


FIG 10

FIG 11



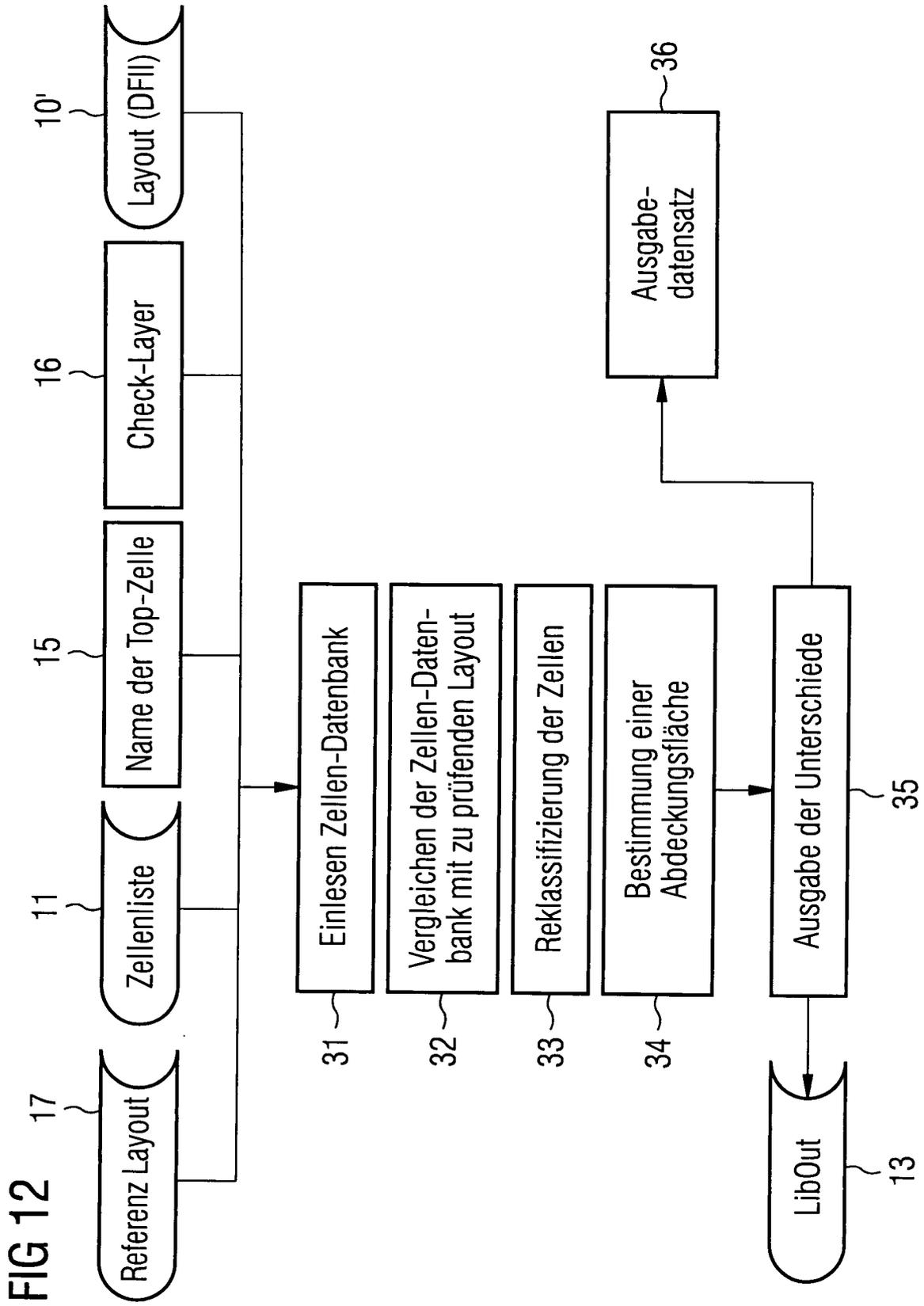


FIG 13A

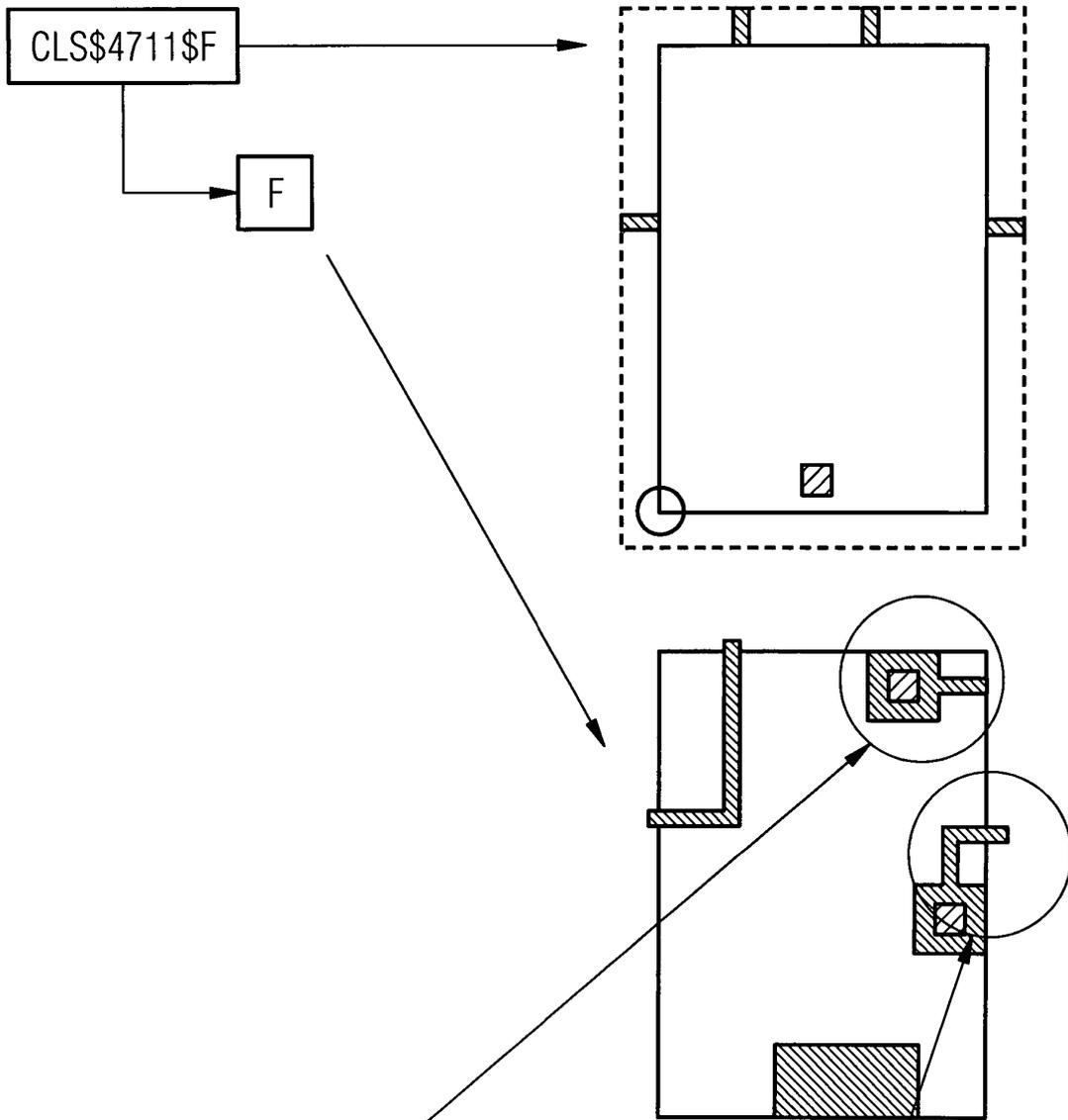


FIG 13B

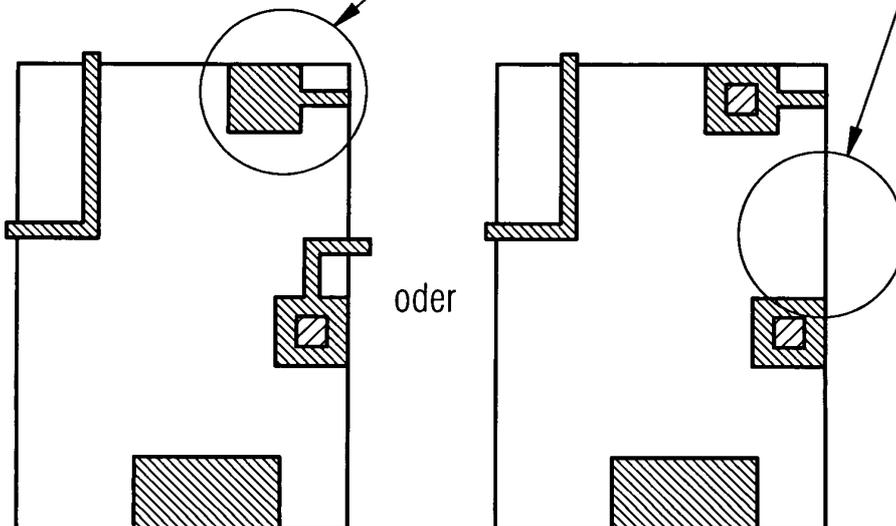


FIG 14A

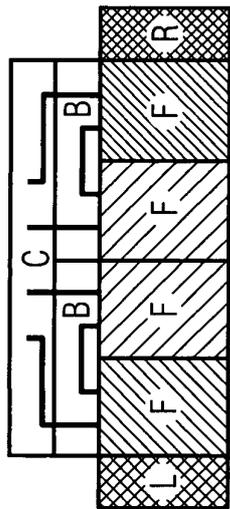


FIG 14C

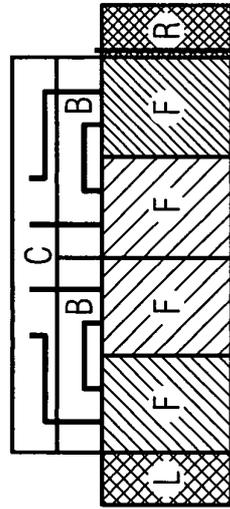


FIG 14D

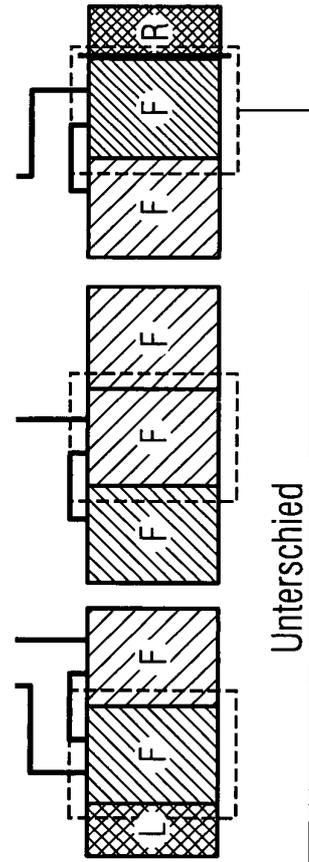


FIG 14B

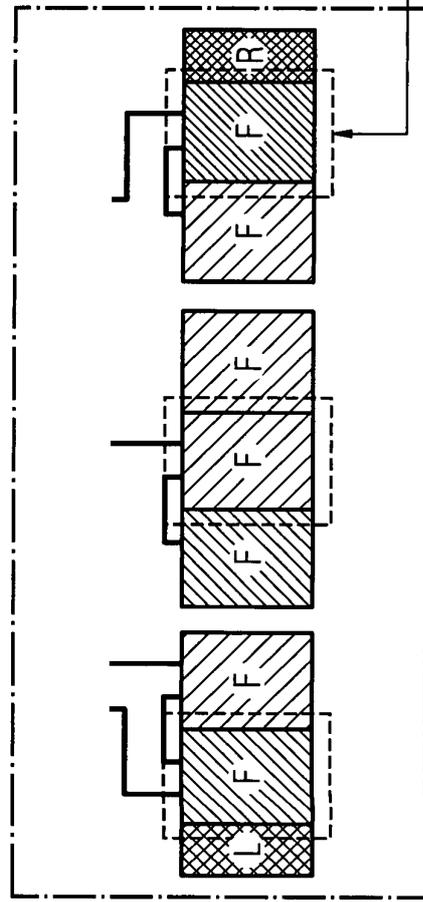
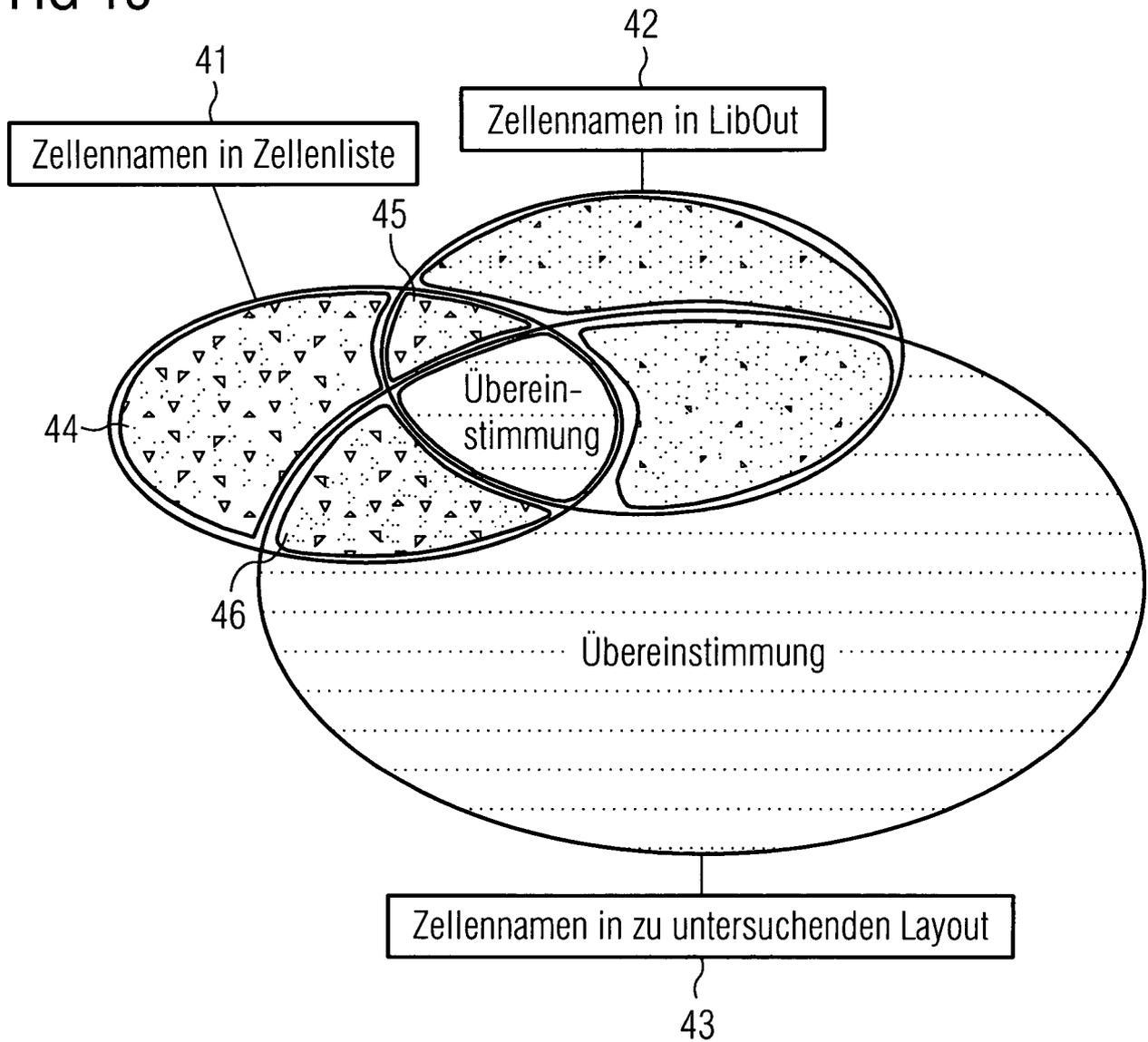


FIG 15



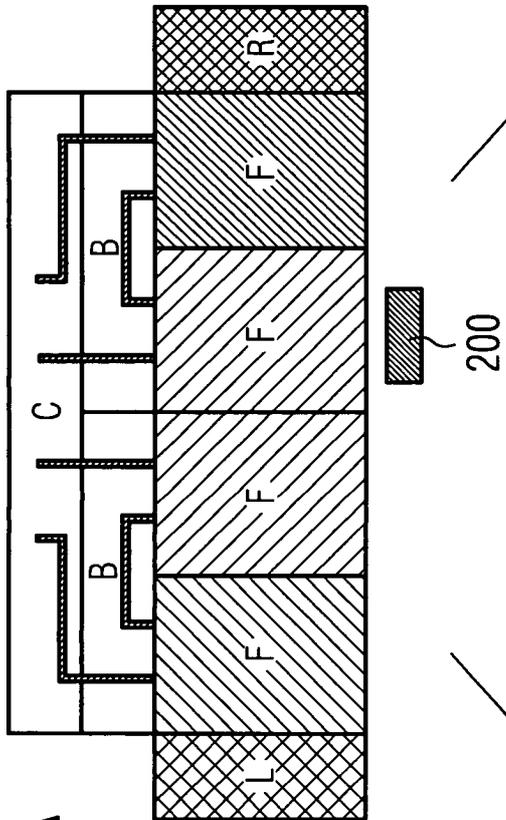


FIG 16A

FIG 16B

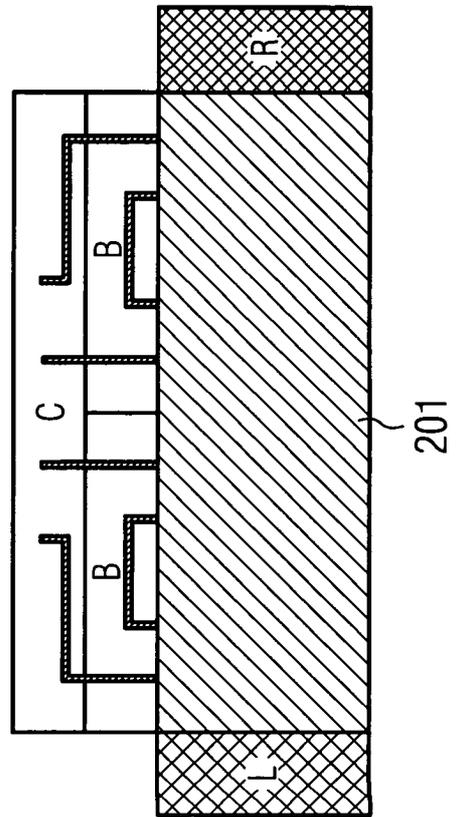


FIG 16C

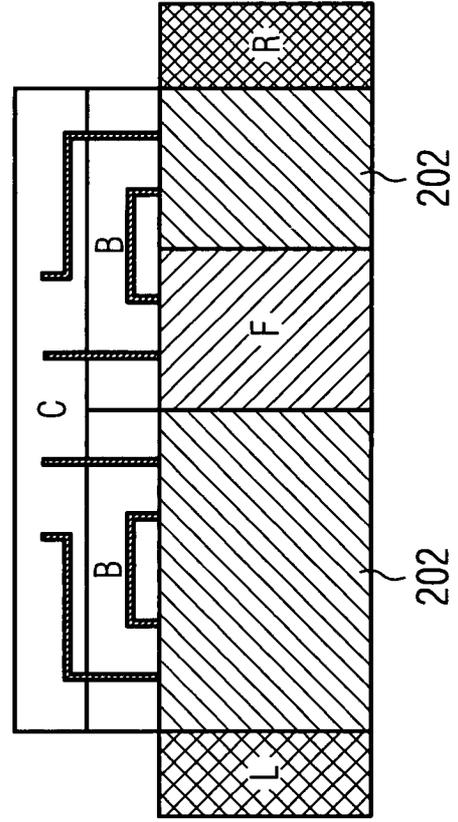


FIG 17A

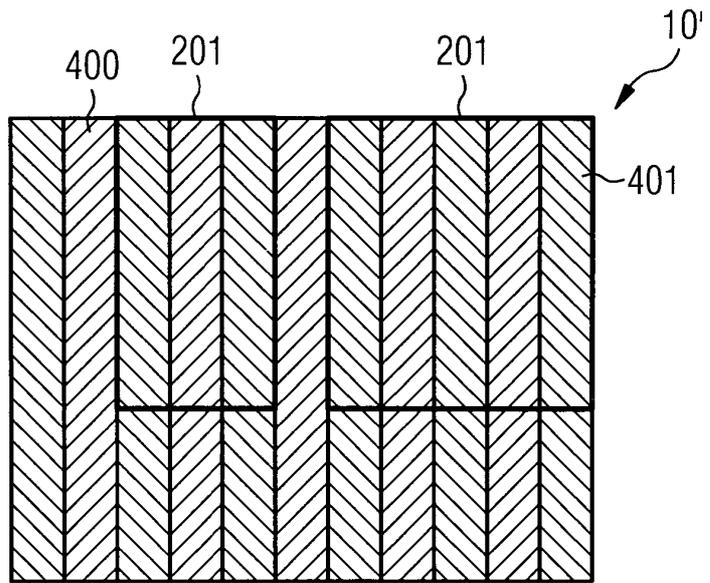


FIG 17B

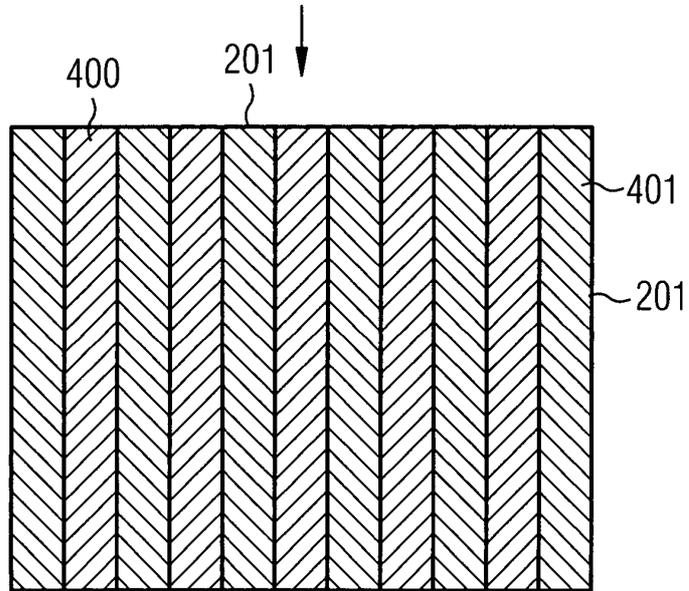


FIG 17C

