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

(10) **DE 698 30 968 T2** 2006.06.14

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 0 921 639 B1**
(21) Deutsches Aktenzeichen: **698 30 968.5**
(96) Europäisches Aktenzeichen: **98 108 914.7**
(96) Europäischer Anmeldetag: **15.05.1998**
(97) Erstveröffentlichung durch das EPA: **09.06.1999**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **27.07.2005**
(47) Veröffentlichungstag im Patentblatt: **14.06.2006**

(51) Int Cl.[8]: ***H03K 19/096*** (2006.01)
H03K 19/017 (2006.01)

(30) Unionspriorität:
955729 **22.10.1997** **US**

(73) Patentinhaber:
Hewlett-Packard Development Co., L.P., Houston, Tex., US

(74) Vertreter:
Schoppe, Zimmermann, Stöckeler & Zinkler, 82049 Pullach

(84) Benannte Vertragsstaaten:
DE, FR, GB

(72) Erfinder:
Naffziger, Samuel D., Fort Collins, US

(54) Bezeichnung: **Dynamische logische Schaltung mit beschränktem Synchronisierungsbedarf und gespeichertem Ausgangszustand**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

Gebiet der Erfindung

[0001] Diese Erfindung bezieht sich allgemein auf eine digitale Logikschaltungsanordnung und insbesondere auf dynamische Logikgatter.

Hintergrund der Erfindung

[0002] Für dynamische Logikgatter wird während einer Vorladetaktphase eine Kapazität geladen und die Kapazität wird während einer Auswertungstaktphase bedingt entladen, abhängig von dem logischen Kombinationszustand der Logikschaltungsanordnung. Fig. 1 stellt ein vereinfachtes herkömmliches MOS-Single-Rail-Dominodynamiklogikgatter dar. Der Begriff „Single-Rail (Einzelschiene)" bedeutet, dass logisch WAHR oder FALSCH bestimmt wird dadurch, ob ein einzelner Auswertungsknoten (**110**) in einem hohen oder einem niedrigen Zustand ist. Im Gegensatz dazu haben Dual-Rail-Gatter (Doppelschienengatter) zwei Auswertungsknoten und verallgemeinerte „Mausefallen"-Gatter können eine beliebige Anzahl von Auswertungsknoten haben. Der Begriff „Domino" bedeutet, dass das Gatter einen invertierenden Puffer (**114**) an dem Ausgang hat. Wenn in Fig. 1 der Takt **100** in einem niedrigen Zustand ist (Vorladephase), lädt der Transistor **102** die Auswertungsknotenkapazität **104** auf die Versorgungsspannung V_{DD}. Wenn der Takt **100** in einem hohen Zustand ist (Auswertungsphase) entlädt der Transistor **106** die Auswertungsknotenkapazität **104** bedingt, abhängig von dem logischen Kombinationszustand der Logikschaltungsanordnung **108**.

[0003] Fig. 2A und Fig. 2B stellen die Zeitgebung für die dynamische Logikschaltung von Fig. 1 dar. Fig. 2A stellt den Taktsignalverlauf dar und Fig. 2B stellt die Spannung auf dem Knoten **110** dar. Wenn in Fig. 2A der Takt in einen niedrigen Zustand geht (Vorladezustand), wird die Spannung auf dem Knoten **110** (Fig. 2B) in einen hohen Zustand getrieben. Aus Effizienzgründen ist der Vorladetransistor **102** (Fig. 1) typischerweise klein und erfordert manchmal den größten Teil der Vorladezeit, um die Knotenkapazität **104** zu laden. Wenn der Takt in einen logisch hohen Zustand geht (Auswertungszustand), ist der Knoten **110** in einem logisch hohen Zustand (vorgeladen), und manchmal während dem Auswertungszustand kann die Knotenkapazität **104** durch die Logikschaltungsanordnung entladen werden (der Knoten **110** zu einem logisch niedrigen Zustand gezogen werden). Es ist anzumerken, dass die Auswertung „monoton" ist. Das heißt, falls der Knoten **100** während der Auswertungsphase in einen logisch niedrigen Zustand geht, kann er während der Auswertungsphase nicht erneut in einen logisch hohen Zustand gehen, da der Knoten **110**, sobald er entladen ist, für die Dauer der Auswertungsphase entladen bleibt.

[0004] Es ist insbesondere anzumerken, dass der Takt **100** unvermeidlicherweise Jitter aufweist, was in Fig. 2A dargestellt ist, wo der Takt **100** eine nominale Abfallzeit **200** aufweist, eine ungünstigste späte Abfallzeit **202** und eine ungünstigste frühe Abfallzeit **204**. Für Gatter, wie sie in Fig. 1 dargestellt sind, muss die Auswertung vor der ungünstigsten frühen Abfallzeit **204** für den Takt **100** abgeschlossen sein. Wie es nachfolgend näher erörtert wird, legen Serienverzögerungs- und Latch-Speicher-Einrichtungszeiten zusätzliche Beschränkungen auf die Auswertungszeit, was es erfordert, dass die Auswertung im Wesentlichen früher abgeschlossen ist als die ungünstigste frühe Abfallzeit **204** für den Takt **100**.

[0005] In Fig. 1 ist der Ausgang **112** typischerweise während der gesamten Vorladephase ungültig. Wenn dynamische Gatter als Eingänge in eine statische Logik oder eine dynamische Logik, die eine andere Taktphase aufweist, verbunden sind, müssen die Ausgänge der dynamischen Gatter typischerweise zwischengespeichert sein, um während der Vorladephase gültige Ausgangssignale zu liefern. Mit ordnungsgemäßer Zeitsteuerung und getrennten verzögerten Taktsignalen zusätzlich zu regelmäßigen Taktsignalen kann es sein, dass einige Latch-Speicher nicht benötigt werden, aber allgemein werden Latch-Speicher benötigt, um Zustände während der Vorladung beizubehalten. Außerdem werden Latch-Speicher für einen Test benötigt (nachfolgend näher erörtert).

[0006] Fig. 3A stellt das Logikgatter von Fig. 1 mit einem Latch-Speicher **300** auf dem Ausgang dar. Eine allgemeine Konfiguration für den Ausgangs-Latch-Speicher **300** ist in Fig. 3B dargestellt. In Fig. 3B sind die Transistoren, die die Inverter **306** und **308** bilden, relativ klein, so dass der Ausgang des Inverters **306** durch den Inverter **114** übersteuert werden kann (Fig. 3A). In Fig. 3B, wenn der Schalter **304** leitet, gibt es für einen Niedrig-zu-Hoch-Übergang (d. h., wenn die Logik **108** den Entladungsknoten **110** auswertet) eine Verzögerung aufgrund des finiten Widerstands des Schalters **304**, gefolgt von einer zusätzlichen Verzögerung, während der Inverter **306** „kämpft", um zu verhindern, dass der Eingang **112** ansteigt. Daher fügt die Schaltung in Fig. 3B eine Verzögerung hinzu, bevor sich das Ergebnis einer Auswertung zu einer nachgeschalteten Logik ausbreitet. Eine Verbesserung ist in Fig. 3C gezeigt. Mit kreuzgekoppelten NOR-Gattern **310** und **312**, wie es dargestellt ist, geht der Takt **100** wesentlich früher in einen logisch hohen Zustand für den Auswertungszustand, als ein Übergang auf dem Eingang **112**, der sich von der Auswertung ergibt. Dann muss der Eingang **112** nur den Zustand eines einzelnen Transistors ändern, ohne „Antriebskämpfe". Es gibt jedoch dennoch eine gewisse Verzögerung. Wenn Daten

nach der Auswertung zwischengespeichert werden, müssen die Daten für eine Zeitperiode vor dem Zwischenspeicher-Taktübergang stabil sein, was als Einrichtungszeit bezeichnet wird. Beispielsweise müssen die Daten für die Schaltung in Fig. 3B ausreichend früh stabil sein, um eine Verzögerung durch den Schalter **304** zu erlauben, bevor der Takt **100** fällt. Der Latc-Speicher von Fig. 3C hat keine Einrichtungszeiterfordernis.

[0007] Mit erneuter Bezugnahme auf die Zeitdiagramme von Fig. 2A und Fig. 2B muss die Auswertung ausreichend früher abgeschlossen sein als die ungünstigste frühe Abfallzeit **204** des Takts **100**, um die ungünstigste Zwischenspeicher-Einrichtungszeit und jede Reihenverzögerung zwischen dem Auswertungsknoten und dem Latch-Speicher zu berücksichtigen.

[0008] Eine zusätzliche Anforderung für Logikgatter ist Testen. Es ist üblich, eine Schaltungsanordnung aufzunehmen, um den Zustand der internen Schaltungsanordnung zu erfassen und die erfassten Zustände zu einem Teststift zu bringen, wo die Zustände seriell gelesen werden. Außerdem kann die Testschaltungsanordnung verwendet werden, um den Zustand einer internen Schaltungsanordnung zu steuern. Ein üblicherweise verwendeter Standard für eine solche Testschaltungsanordnung ist der IEEE-Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture, erhältlich von The Institute of Electrical and Electronic Engineers, Inc., 345 East 47th Street, New York, NY 10017-2394. Der IEEE Standard 1149.1 definiert eine Standardserienschnittstelle, durch die Testbefehle und Testdaten kommuniziert werden. Die Technik umfasst die Aufnahme einer Schieberegisterstufe benachbart zu jedem Knoten von Interesse, so dass Signale an Knoten von Interessen gesteuert und beobachtet werden können. Die speziellen Schieberegister und Testbefehle können bei getrenntem Komponententesten verwendet werden und beim Testen einzelner Komponenten, die in größere Systeme zusammengebaut sind. Im Allgemeinen gibt es einen Bedarf, den Mehraufwand der Schaltungsanordnung zu minimieren, der erforderlich ist, um innere logische Knoten durch Grenzabtast-Testen zu beobachten und zu steuern. Vorzugsweise werden statt zusätzlichen Schieberegisterstufen, die dem Testen zugewiesen sind, Ausgangs-Latch-Speicher, wie sie in Fig. 3A (**300**) dargestellt sind, ebenfalls für Grenzabtast-Testen verwendet. Das Testen erfordert jedoch typischerweise komplexere Latch-Speicher als diejenigen, die in Fig. 3B und Fig. 3C dargestellt sind. Insbesondere muss der beobachtete Ausgang zwischengespeichert werden bevor der Ausgang durch das Testsystem geändert (gesteuert) wird, was typischerweise eine Master-/Slave-Latch-Speicher-Anordnung erfordert. Falls außerdem in Fig. 1 der Takt angehalten wird, kann der Knoten **110** nicht gesteuert werden, falls der Vorladetransistor **102** ein ist oder falls die Logik auswertet, den Knoten **100** zu entladen und der Auswertungstransistor **106** ein ist. Daher ist ein bidirektionaler Isolationsschalter (wie der Schalter **304** von Fig. 3B) zwischen dem Knoten **110** und dem Knoten, der gesteuert wird, erforderlich. Ein Trennschalter fügt Verzögerung hinzu.

[0009] Die US-A-5,557,620 beschreibt ein System und ein Verfahren für Ruhestromtesten einer dynamischen Logikschaltungsanordnung. Knoten, die zu Masse kurzgeschlossen sind, werden während einem dynamischen Vorladezustand erfasst. Knoten, die zu einem Leistungsversorgungspotential kurzgeschlossen sind, werden durch Treiben aller Knoten von Interesse auf Masse während einer dynamischen Auswertungsphase erfasst. Knoten von Interesse werden direkt zu Masse getrieben durch einen zusätzlichen Transistor pro Knoten oder indirekt durch logische Ausbreitung von vorgeschalteten Knoten. Als Folge werden nur zwei Strommessungen für alle kurzgeschlossenen Knotenfehler benötigt, selbst für pipelineartige Systeme mit mehreren Takten. Es gibt keinen Bedarf an Eingabetestsignalsequenzen und keinen Bedarf an Signalausbreitung zu Ausgängen für die Erfassung. Spezifische Ausführungsbeispiele sind vorgesehen für Single-Rail-Logik, pipelineartige Single-Rail-Systeme, Dual-Rail-Logik und pipelineartige Dual-Rail-Systeme. Für pipelineartige Single-Rail-Systeme ermöglichen optionale Transistoren zwischen den Stufen eine Beibehaltung von logischen Zuständen während dem Testen. Für eine Dual-Rail-Logik werden Speicherknoten und statische Knoten zu einem logischen Zustand gezwungen, der während einem normalen Betrieb nicht möglich ist. Für eine pipelineartige Dual-Rail-Logik behält das Testen abwechselnder Stufen inhärent den logischen Zustand des Systems während dem Testen bei.

[0010] Die US-A-5,550,487 und die U5-A-5,440,243 beschreiben ein statisch betriebenes dynamisches CMOS-Logikgatter, das ein FET-Logiknetzwerk zum Durchführen einer vordefinierten Logikfunktion bezüglich ihrer Logikeingänge, einen Ausgangsknoten, einen Vorladetransistor und bei einigen Beispielen einen Auswertungstransistor umfasst. Während dem Betrieb wird der Vorladetransistor zuerst eingeschaltet durch ein Taktsignal während einer Vorladephase, um einen Ausgangsknoten des dynamischen Logikgatters auf einen ersten Spannungszustand vorzuladen. Während der Vorladephase wird der Auswertungstransistor durch das Taktsignal abgeschaltet. Der Vorladephase folgt typischerweise eine Auswertungsphase und während der Auswertungsphase wird der Auswertungstransistor durch das Steuersignal eingeschaltet, um es dem Logiknetzwerk zu ermöglichen, die vordefinierte Logikfunktion bezüglich ihrer Eingänge durchzuführen, und das Logiknetzwerk lädt oder entlädt selektiv den Ausgangsknoten

auf einen zweiten Spannungszustand über den Auswertungstransistor gemäß der vordefinierten Logikfunktion, die den Logikeingängen zu dem Logikgatter gegeben wird. Eine Treiberschaltung ist vorgesehen zum Anlegen einer Vorspannung an das Gatter des Vorladetransistors, wenn der Vorladetransistor den Ausgangsknoten nicht vorlädt (z. B. Auswertungsphase). Die Vorspannung hat einen Spannungspegel, der sich von dem ersten Spannungszustand unterscheidet, um weniger als die Größe der Schwellenspannung des Vorladetransistors, der mit dem Vorladetransistor in einer Unterschwellenwertleitungsregion arbeitet, um sicherzustellen, dass der Ausgangsknoten des Logikgatters bei dem ersten Spannungszustand ist, wenn das Logiknetzwerk den Ausgangsknoten nicht durch den Auswertungstransistor auf den zweiten Spannungszustand entlädt, als Folge der vorbestimmten Logikfunktion. Auf diese Weise kann die dynamische Logikgatterschaltung statisch arbeiten, mit im Wesentlichen minimiertem Leistungsverbrauch.

[0011] Es ist die Aufgabe der vorliegenden Erfindung, ein verbessertes dynamisches Logikgatter und ein Verfahren zum Steuern desselben zu schaffen, mit: (1) entspannten Zeitsteuerungsanforderungen für den Abschluss der vorgeschalteten Auswertung, (2) Ausgabe-Halten während der Vorladephase und (3) vereinfachte Beobachtbarkeit und Steuerbarkeit.

[0012] Diese Aufgabe wird durch ein Verfahren gemäß Anspruch 1 und durch ein dynamisches Logikgatter gemäß Anspruch 4 gelöst.

Zusammenfassung der Erfindung

[0013] Ein dynamisches Logikgatter ist offenbart, das die folgenden Zeitsteuerungs-Charakteristika aufweist:

(a) das Gatter lädt während der ersten Hälfte der Auswertungsphase vor;

(b) das Gatter wertet während dem verbleibenden Abschnitt der Auswertungsphase aus und kann teilweise in der Vorladephase weiterhin auswerten;

(c) die Auswertungsknotenspannung wird während der Vorladephase statisch in einem Tri-State-Zustand gehalten.

[0014] Als Folge sind die Zeitsteuerungsanforderungen auf der vorgeschalteten Logik im Wesentlichen entspannt, weil die vorgeschaltete Logik die Auswertung nicht vor der abfallenden Flanke des Takts abschließen muss, und da das Gatter selbst als ein Latch-Speicher arbeitet, gibt es keine explizite Zwischenspeicherverzögerung. Mit ausgedehnter Auswertung und Halten ist das resultierende Gatter besonders sinnvoll als Schnittstelle zwischen dynamischer Logik und statischer Logik oder als Schnittstelle zwischen dynamischer Logik mit Auswertung während einer Taktphase und dynamischer Logik mit Auswertung während der entgegengesetzten Taktphase. Da der logische Zustand während der Vorladephase statisch gehalten wird, liefert das Gatter die Funktionalität eines Latch-Speichers ohne die Verzögerung eines Latch-Speichers. Da der Gatterausgang während der Vorladephase effektiv tristatemäßig betrieben wird, kann der Zustand schließlich für Teststeuerung ohne weiteres übersteuert werden.

[0015] Ein Dual-Rail-Ausführungsbeispiel ist ebenfalls offenbart, das zusätzlich zu den obigen Vorteilen effektiv Erstes-Eintreffen-Zwischenspeichern liefert und dadurch einen unzulässigen Ausgabezustand verhindert.

Kurze Beschreibung der Zeichnungen

[0016] Fig. 1 (Stand der Technik) ist ein Blockdiagrammschema eines Single-Rail-Dynamiklogikgatters.

[0017] Fig. 2A (Stand der Technik) sind Zeitdiagramme für Signa- und Fig. 2B le, die sich auf die Schaltung von Fig. 1 beziehen.

[0018] Fig. 3A (Stand der Technik) ist ein Blockdiagrammschema des Logikgatters von Fig. 1 mit einem Ausgangs-Latch-Speicher.

[0019] Fig. 3B (Stand der Technik) ist ein Blockdiagrammschema, das zusätzliche Einzelheiten für ein Beispiel des Ausgangs-Latch-Speicher von Fig. 3A darstellt.

[0020] Fig. 3C (Stand der Technik) ist ein Blockdiagrammschema, das zusätzliche Einzelheiten für ein alternatives Beispiel des Ausgangs-Latch-Speichers von Fig. 3A darstellt.

[0021] Fig. 4 ist ein vereinfachtes Blockdiagrammschema für ein beispielhaftes Ausführungsbeispiel eines Single-Rail-Dynamiklogikgatters gemäß der Erfindung.

[0022] Fig. 5A sind Zeitdiagramme von Signalverläufen, die sich bis Fig. 5D auf die Schaltung von Fig. 4 beziehen.

[0023] Fig. 6 ist ein Blockdiagrammschema eines Dual-Rail-Beispiels.

Detaillierte Beschreibung des bevorzugten Ausführungsbeispiels der Erfindung

[0024] Fig. 4 stellt ein beispielhaftes Ausführungsbeispiel eines Single-Rail-Dynamiklogikgatters gemäß der Erfindung dar. Von Fig. 1 und Fig. 2A wird daran erinnert, dass die Schaltung von Fig. 1 vorlädt, wenn der Takt **100** in einem niedrigen Zustand ist und

auswertet, wenn der Takt **100** in einem hohen Zustand ist. Im Gegensatz dazu lädt die Schaltung in Fig. 4 für eine sehr kurze Zeit am Anfang der Phase vor, wenn der Takt **100** in einem hohen Zustand ist, wertet unmittelbar nach dem Vorladen aus und die Auswertung wird über die Zeit hinaus fortgesetzt, wenn der Takt **100** von einem hohen zu einem niedrigen Zustand geht. Außerdem wird der Auswertungsknoten effektiv tristatemäßig betrieben, nach der Auswertung während des gesamten verbleibenden Teils der Phase, wenn der Takt **100** in einem niedrigen Zustand ist. Während der tristatemäßig betriebenen Periode wird der Auswertungsknoten „schwach" gehalten (durch physikalisch kleine Transistoren), um Driften zu verhindern (insbesondere, wenn der Takt zum Testen angehalten wird), kann aber für Steuerung während dem Testen ohne weiteres übersteuert werden.

[0025] Für das Gatter in Fig. 4 werden zwei getrennte Taktsignale von dem Haupttakt **100** abgeleitet. Ein Rücksetztakt RCK **400** steuert einen Vorladetransistor **402**. Ein Auswertungstakt ECK **404** steuert einen Auswertungstransistor **406**. Wenn der Takt **100** in einen logisch hohen Zustand geht, pulst RCK **400** in einem niedrigen Zustand für eine Zeit, die durch eine Verzögerungsschaltung **408** bestimmt wird, und kehrt dann zu einem hohen Zustand zurück. RCK **400** pulst nur lang genug in einem niedrigen Zustand, um sicherzustellen, dass der Vorladetransistor **402** den Auswertungsknoten NOH **410** vollständig lädt. Der Transistor **402** ist größer als typische Vorladetransistoren und lädt den Auswertungsknoten **410** in weniger als der Hälfte der hohen Phase des Takts **100**. Zur gleichen Zeit, wenn der Vorladetransistor **402** abschaltet, geht ECK **404** in einen hohen Zustand und der Auswertungstransistor **406** ist eingeschaltet. Nachdem der Takt **100** in einen niedrigen Zustand geht (Verzögerung **408**), geht NCKD **409** in einen hohen Zustand und der Auswertungstransistor **406** schaltet aus. Als Folge sind während des verbleibenden Teils der niedrigen Phase des Takts **100** beide Transistoren **402** und **406** aus und betreiben den Auswertungsknoten **410** effektiv tristatemäßig.

[0026] Kurz bevor der Takt **100** von einem niedrigen zu einem hohen Zustand übergeht, ist der Transistor **412** aus, der Transistor **414** ist ein, der Transistor **416** ist ein (RCK ist in einem hohen Zustand) und der Transistor **418** ist aus. Wenn der Takt **100** in einen hohen Zustand geht, wird der Transistor **416** ausgeschaltet und der Transistor **412** wird eingeschaltet, was RCK in einen niedrigen Zustand zieht, was den Vorladetransistor **402** einschaltet. Nach einer Verzögerung **408** geht NCKD **409** in einen niedrigen Zustand, was den Transistor **414** aus- und den Transistor **418** einschaltet, was RCK in einen hohen Zustand zieht, was den Vorladetransistor **402** ausschaltet. Wenn der Takt **100** von einem hohen zu einem niedrigen Zustand geht, wird der Transistor **412** ausgeschaltet und der Transistor **416** wird eingeschaltet, so dass RCK in einem hohen Zustand bleibt. Die Verzögerung **408** kann eine Reihe von Invertierern (gerade Anzahl) sein, oder kann ein einfaches RC-Filter sein.

[0027] Die Transistoren **420** und **422** bilden einen Inverter. Die Transistoren **424** und **426** halten den Zustand des Auswertungsknoten **410** schwach. Die Transistoren **424** und **426** sind physikalisch klein und liefern nur gerade ausreichend Strom, um Drift des Auswertungsknotens **410** zu verhindern, falls der Takt zum Testen angehalten wird, der aber während dem Testen ohne weiteres übersteuert wird. Der Transistor **426** ist optional.

[0028] Fig. 4 stellt auch dar, dass das tristatemäßige Betreiben des Auswertungsknotens **410** mit einer schwachen Haltefähigkeit das Steuern während dem Test stark vereinfacht. Ein Mastertakt kann angehalten werden und der Takt **100** kann unabhängig gesteuert werden oder für einen Test gezwungen werden. Falls der Takt **100** in einem niedrigen Zustand angehalten wird, sind dann alle Gatter wie in Fig. 4 in einer Haltephase. Für alle Gatter, wie das Gatter von Fig. 4, kann der Haltezustand des Knotens **410** über den Schalter **428** zu einem Test-Latch-Speicher gelesen werden, wie es durch ein erstes Testsignal SHIFT1 (**430**) gesteuert wird. Der Bedarf an einem physikalischen Test-Latch-Speicher ist eliminiert, weil der Zustand des Knotens **410** gehalten wird. Dann kann ein neuer Zustand auf den Knoten NOH **410** gezwungen werden, durch den Schalter **432**, wie er durch ein zweites Testsignal SHIFT2 (**434**) gesteuert wird. Falls der Knoten NOH **410** nicht tristatemäßig betrieben wurde, würden Trennschalter (mit ihrer inhärenten Verzögerung) benötigt werden, um es zu ermöglichen, dass NOH **410** übersteuert wird. Der Takt **100** kann dann zyklisch betrieben werden, so dass alle nachgeschaltete Logik auswertet, um die Auswertung mit bekannten Eingangssignalen zu testen. Es ist anzumerken, dass bei einigen Systemen SHIFT2 einfach das Inverse von SHIFT1 sein kann.

[0029] Die Zeitgebung für die Schaltung von Fig. 4 ist in Fig. 5A – Fig. 5D dargestellt. Fig. 5A stellt den Takt **100** wie in Fig. 2A dar, aber mit anderen Definitionen der Phasen. Die hohe Phase wird nun als „Vorladen und Auswerten" bezeichnet, und die niedrige Phase wird als „Halten" bezeichnet. Fig. 5B stellt den Rücksetztakt RCK **400** dar. Es ist anzumerken, dass RCK in einen niedrigen Zustand geht, wenn der Takt **100** in einen hohen Zustand geht, und nach einer kurzen Verzögerung erneut in einen hohen Zustand geht. Fig. 5C stellt das Signal NOH **410** dar, das die Auswertungsknotenspannung ist. Es ist anzumerken, dass NOH auf einen niedrigen Zustand getrieben werden kann als Folge einer Logikauswertung, oder in einem hohen Zustand bleiben kann als Folge einer Logikauswertung. Fig. 5D stellt den Auswertungstakt ECK **404** dar. Es ist anzumerken, dass RCK **400** und

ECK **404** gleichzeitig in einen hohen Zustand gehen, verzögert um eine Verzögerung **408**, nachdem der Takt **100** in einen hohen Zustand geht. Solange ECK in einem hohen Zustand ist, ist die Logikschaltung in einem Auswertungsmodus. ECK geht in einen niedrigen Zustand, nachdem der Takt **100** in einen niedrigen Zustand geht, verzögert um die Verzögerung **408**.

[0030] Es ist insbesondere in Fig. 5D anzumerken, dass das Gatter von Fig. 4 den Bedarf eliminiert, dass die Auswertung vor der ungünstigsten frühen Abfallzeit des Takts **100** abgeschlossen ist. In der Tat „stiehlt sich die Auswertung zyklusmäßig" teilweise in die Phase, wo der Takt **100** in einem niedrigen Zustand ist, und setzt sich für eine kurze Zeit fort, nachdem der Takt **100** fällt, wobei die vorgeschaltete Logik in die Vorladephase eintritt. Daher sind die Zeitanforderungen für vorgeschaltete Logik im Wesentlichen entspannt und Empfindlichkeit gegen den Taktjitter ist reduziert. Es ist schließlich anzumerken, dass das Halten des Zustands von NOH während der Vorladephase die Funktion eines Latch-Speichers ohne die Verzögerung oder Einrichtungszeit eines Latch-Speichers liefert. Als eine Folge der späten Auswertung und des Haltens ist das Gatter von Fig. 4 besonders sinnvoll als die Schnittstelle zwischen dynamischer Logik und statischer Logik. Alternativ ist das Gatter von Fig. 4 besonders sinnvoll als Schnittstelle zwischen dynamischer Logik mit Auswertung während einer Taktphase und dynamischer Logik mit Auswertung während der entgegengesetzten Taktphase.

[0031] Fig. 6 stellt ein vereinfachtes Dual-Rail-Gatter gemäß der Erfindung dar. Um die Darstellung zu vereinfachen, sind einige Teile, die gleich sind wie in Fig. 4, ausgelassen. Ein Dual-Rail-Gatter hat zwei Eingänge für jedes Logikeingangssignal (das Logikeingangssignal und das Komplement des logischen Eingangssignals) und hat zwei Ausgänge (einen normalen Ausgang und das Komplement des normalen Ausgangs). Der Rücksetztakt **600** hat einen schwachen (physikalisch kleinen) Pull-Up-Transistor **626**, der gerade ausreicht, um den RCK davon abzuhalten, zu einem niedrigen Zustand zu driften, wenn alle Transistoren, die RCK treiben, aus sind. Wenn der Takt **100** in einen hohen Zustand geht, geht RCK **600** in einen niedrigen Zustand und beide Vorladetransistoren **602** und **604** schalten sich ein. Nach der Verzögerung (Fig. 4, **408**) geht NCKD in einen niedrigen Zustand und schaltet den Transistor **414** aus. Selbst wenn der Transistor **414** ausschaltet, bleiben RCK und die Gatter der Vorladetransistoren **602** und **604** in einem niedrigen Zustand, es sei denn, sie werden aktiv in einen hohen Zustand gezogen. Der kleine Pull-Up-Transistor **626** kann RCK und die Gatter der Transistoren **602** und **604** langsam in einen hohen Zustand ziehen, aber bevor dies eintritt, werden beide Auswertungsknoten **606** und **608** geladen. Wenn beide Auswertungsknoten **606** und **608** geladen werden, schaltet das logische UND-Gatter **610** den Transistor **624** ein, wodurch RCK **600** in einen hohen Zustand gezogen wird und beide Vorladetransistoren ausgeschaltet werden. Gleichzeitig bewirkt das logische UND-Gatter **610** über das Signal NEVAL **612**, dass beide Auswertungstransistoren **616** und **618** eingeschaltet werden (es ist anzumerken, dass NCKD **409** in einen niedrigen Zustand geht, bevor NEVAL in einen niedrigen Zustand geht, wie es oben erörtert ist). Eine Gruppe von Transistoren **620** ist als ein logisches UND-Gatter konfiguriert, was erfordert, dass die Signale A, B, und C alle in einem hohen Zustand sind, um zu bewirken, dass der Knoten NOH **606** während der Auswertung entladen wird. Eine zweite Gruppe von Transistoren **622** ist als ein logisches ODER-Gatter konfiguriert, was den Knoten NOL **608** entlädt, falls das Inverse von einem der Signale A, B und C in einem hohen Zustand ist. Falls die Signale A, B und C in einem hohen Zustand sind, wird der Knoten NOH **606** entladen und der Knoten NOL **608** bleibt in einem hohen Zustand. Für alle anderen logischen Zustände für A, B und C wird der Knoten NOL entladen und der Knoten NOH bleibt in einem hohen Zustand. Wenn für das in Fig. 6 dargestellte Dual-Rail-Gatter entweder der Knoten NOH oder NOL entlädt, bewirkt das logische UND-Gatter **610** über das Signal NEVAL **612**, dass beide Auswertungstransistoren **616** und **618** ausgeschaltet werden. Dann werden beide Knoten NOH und NOL effektiv tristatemäßig betrieben und schwach gehalten. Mit komplementären Eingangssignalen ist garantiert, dass ein Auswertungsknoten entlädt, so dass nach der Auswertung beide Auswertungsknoten während beiden Phasen des Takts tristatemäßig betrieben werden. Daher gibt es keinen Bedarf, den Takt **100** für den Test in einen niedrigen Zustand zu zwingen.

[0032] Das Dual-Rail-Gatter von Fig. 6 hat alle Vorteile des Single-Rail-Gatters von Fig. 4: es eliminiert den Bedarf, dass die Auswertung vor der ungünstigsten frühen Abfallzeit des Takts **100** abgeschlossen ist, die Auswertung kann „einen Zyklus abzweigen" in die Phase, wo der Takt **100** in einem niedrigen Zustand ist, und damit fortfahren, bis gültige Eingänge bewirken, dass das Gatter auswertet; es liefert die Funktion eines Dual-Rail-Latch-Speichers ohne die Verzögerung eines Latch-Speichers; und beide tristatemäßig betriebene Ausgänge können ohne weiteres für Teststeuerung übersteuert werden. Da außerdem die Entladung von entweder NOH oder NOL bewirkt, dass beide Knoten unmittelbar in einen Haltemodus gehen, liefert das Gatter effektiv Erstes-Eintreffen-Zwischenspeichern und verhindert dadurch aktiv einen illegalen Ausgabezustand. Das heißt, das Gatter verhindert, dass sowohl NOH als auch NOL beide unbeabsichtigt entladen werden aufgrund einer Wettlaufbedingung zwischen den Eingängen oder einem Störimpuls auf einem der Eingänge. Noch ein weiterer Vorteil ist, dass sich das Gatter auf die Auswertung hin selbst tristatemäßig betreibt, so dass der

Auswertungstakt für Testabtasten nicht auf einen niedrigen Zustand gezwungen werden muss.

Patentansprüche

1. Ein Verfahren zum Steuern eines dynamischen Logikgatters, wobei das Verfahren folgende Schritte umfasst:
Empfangen eines Taktsignals (**100**) mit einem ersten Übergang, der einen Anfang einer ersten Phase definiert, und einem zweiten Übergang, der einen Anfang einer zweiten Phase definiert;
Vorladen, das an dem ersten Taktübergang beginnt und vor dem zweiten Taktübergang endet;
Aktivieren einer Auswertung, nachdem das Vorladen beendet ist und vor dem zweiten Übergang; und
Deaktivieren der Auswertung während zumindest einem Teil der zweiten Phase.

2. Das Verfahren gemäß Anspruch 1, das ferner folgenden Schritt umfasst:
Halten einer Spannung an einem Auswertungsknoten während der zweiten Phase.

3. Das Verfahren gemäß Anspruch 1, das ferner folgenden Schritt umfasst:
Deaktivieren des Vorladens während zumindest einem Teil der zweiten Phase.

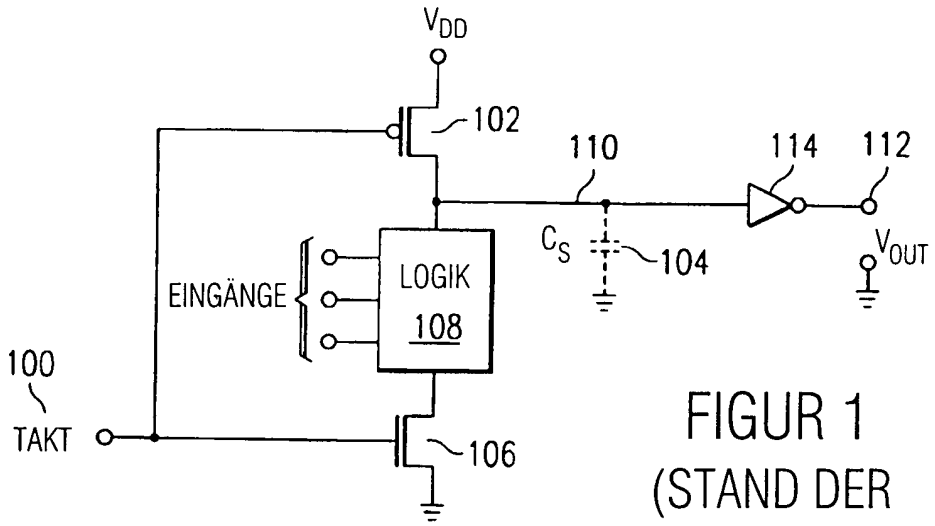
4. Ein dynamisches Logikgatter, wobei das dynamische Logikgatter ein Taktsignal (**100**) empfängt, das eine erste Phase und eine zweite Phase aufweist, wobei das dynamische Logikgatter folgende Merkmale umfasst:
einen Auswertungsknoten (**410**);
einen Vorladeschalter (**402**), der zwischen ein erstes Leistungsversorgungspotential und den Auswertungsknoten geschaltet ist, so dass sich eine Spannung an dem Auswertungsknoten dem ersten Leistungsversorgungspotential nähert, wenn der Vorladeschalter geschlossen ist;
eine Verzögerungsschaltung (**408**) mit einem Eingang und einem Ausgang, wobei der Eingang das Taktsignal empfängt;
eine Vorladesteuerschaltung (**412**, **414**, **416**, **418**), die das Taktsignal und das Ausgangssignal der Verzögerungsschaltung empfängt und mit dem Vorladeschalter verbunden ist;
wobei der Vorladeschalter durch die Vorladesteuerschaltung gesteuert wird, um sich an einem Anfang der ersten Phase zu schließen und nach einer festen Verzögerung und vor einem Anfang der zweiten Phase zu öffnen.

5. Das dynamische Logikgatter gemäß Anspruch 4, das ferner folgende Merkmale umfasst:
einen Auswertungsschalter (**406**);
eine Logikschaltungsanordnung (**108**), wobei die Logikschaltungsanordnung zwischen den Auswertungsknoten und den Auswertungsschalter geschaltet ist, und der Auswertungsschalter zwischen die Logikschaltungsanordnung und ein zweites Leistungsversorgungspotential geschaltet ist;
eine Auswertungssteuerschaltungsanordnung, die das Ausgangssignal der Verzögerungsschaltung empfängt und mit dem Auswertungsschalter verbunden ist; und
wobei der Auswertungsschalter durch die Auswertungssteuerschaltungsanordnung gesteuert wird, um sich nach der festen Verzögerung nach dem Anfang der ersten Phase zu schließen.

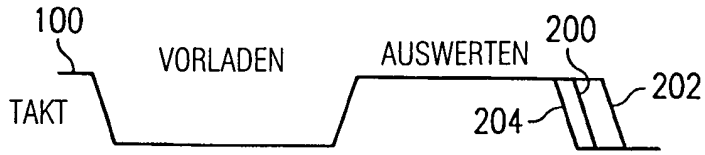
6. Das dynamische Logikgatter gemäß Anspruch 4, das ferner folgendes Merkmal umfasst:
der Vorladeschalter und der Auswertungsschalter werden gesteuert, um während der zweiten Phase geöffnet zu bleiben.

Es folgen 3 Blatt Zeichnungen

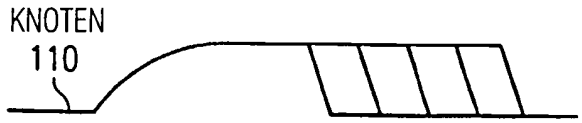
Anhängende Zeichnungen



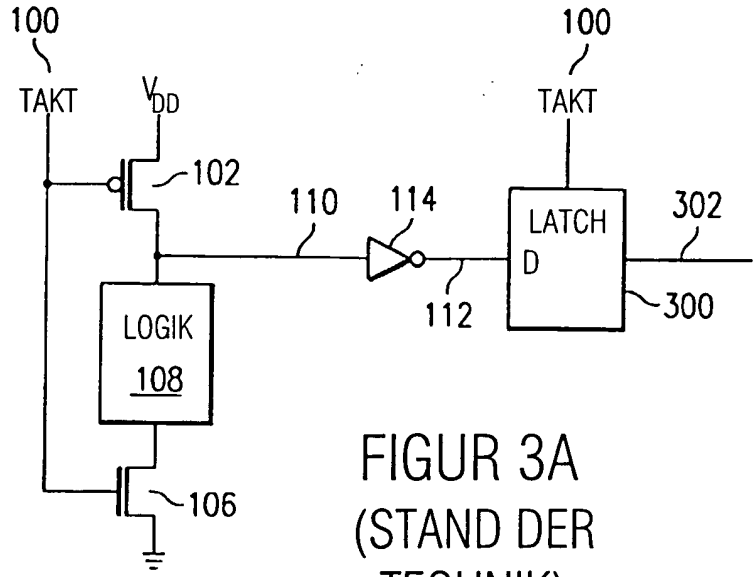
FIGUR 1
(STAND DER TECHNIK)



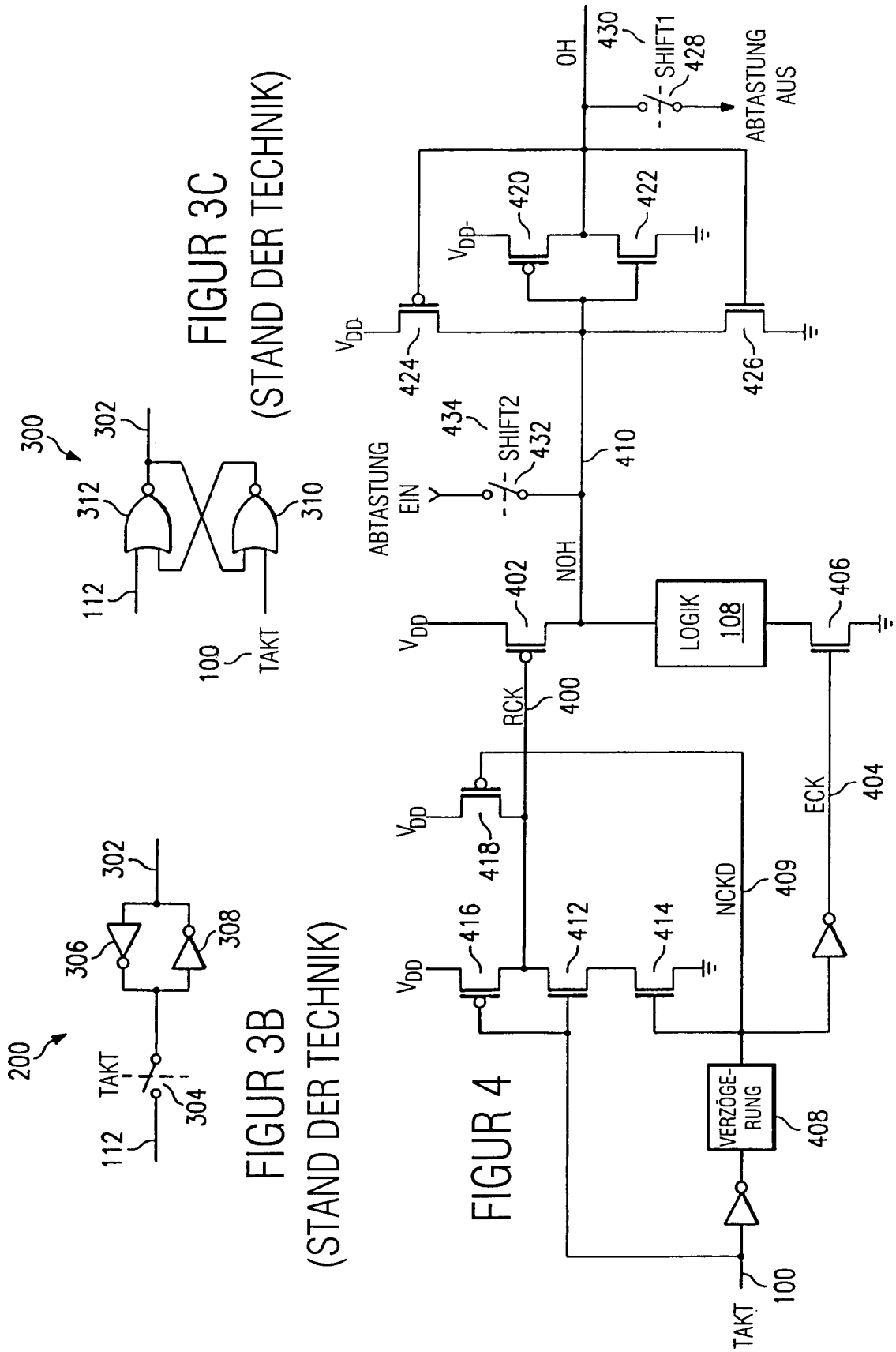
FIGUR 2A
(STAND DER TECHNIK)



FIGUR 2B
(STAND DER TECHNIK)



FIGUR 3A
(STAND DER TECHNIK)



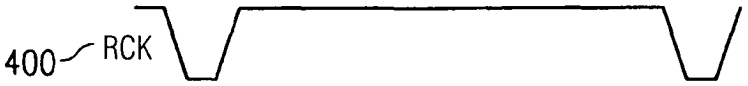
FIGUR 3B
(STAND DER TECHNIK)

FIGUR 3C
(STAND DER TECHNIK)

FIGUR 4



FIGUR 5A



FIGUR 5B



FIGUR 5C



FIGUR 5D

FIGUR 6

