

4.2 Register

Register sind Aneinanderreihungen von Latches oder Flipflops mit gemeinsamem Takt. Durch Zusatzbeschaltung wird aus der einfachen Aneinanderreihung ein Register mit speziellen Funktionen. Einige Funktionen, die besonders häufig benötigt werden, haben den Charakter von Grundschaltungen. Das betrifft vor allem die Schieberegister und Zähler.

4.2.1 Standardschaltkreise

Die Schaltkreishersteller haben im Laufe der Zeit viele Registertypen entwickelt. Einige sind zu echten Industriestandards geworden (daran erkennbar, dass sie in nahezu allen Baureihen angeboten werden). Abb. 4.70 gibt – anhand einer kleinen Auswahl – einen Überblick über typische, vielseitig einsetzbare Register.

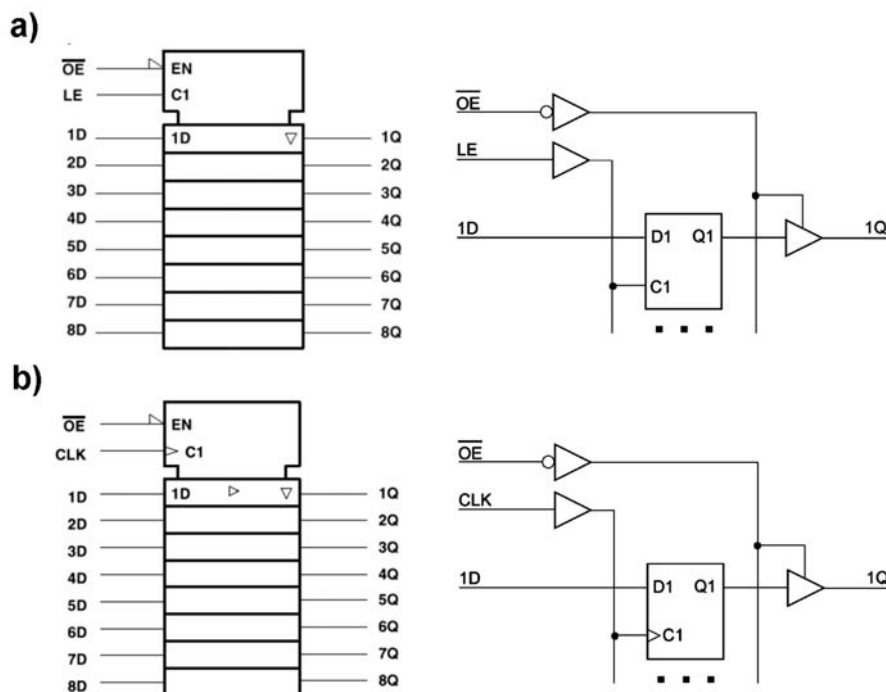


Abb. 4.70 Typische Registerschaltkreise – eine kleine Auswahl. a) Latchregister; b) Flipflopregister. Links Schaltsymbole nach DIN 40 900, rechts Innenschaltung (nach [2.10]).

Die meisten Registerschaltkreise haben Tri-State-Ausgänge. Braucht man zweiwertige Ausgänge, ist der Erlaubniseingang (OE) mit Masse zu verbinden. In vielen Baureihen werden Latch- und D-Flipfloptypen in ansonsten gleicher Auslegung (Gehäuse, Anschlussbelegung, Technologie usw.) angeboten. Gängige Breiten (Anzahl der Bitpositionen): 8, 10, 16, 18, 20, 22, 32, 36. Solche Registerschaltkreise sind vor allem als Puffer, als Haltereister, als Buskoppelstufen und zur Pegelwandlung vorgesehen.

Besonderheiten

Der typische Registerschaltkreis ist heutzutage das einfache Latch- oder Flipflopregister mit nicht invertierten Tri-State-Ausgängen (wie in Abb. 4.70 gezeigt). Viele der in den Datenbüchern angegebenen Typen mit Sonderfunktionen werden für Neuentwicklungen nicht mehr empfohlen oder gar nicht mehr gefertigt. Abb. 4.71 veranschaulicht einige Sondertypen.

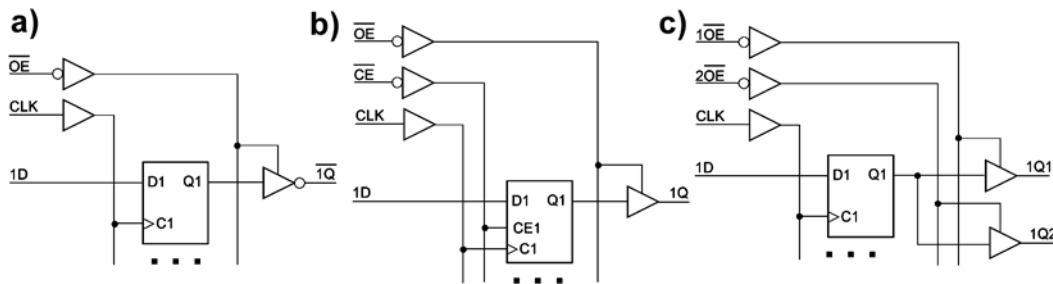


Abb. 4.71 Registerschaltkreise mit Besonderheiten (nach [2.51]). a) invertierte Ausgänge; b) Taktsteuereingang; c) zwei Tri-State-Ausgänge.

Universelle Buskoppelschaltkreise

Der Grundgedanke: ein Schaltkreis, vielfältige Einsatzmöglichkeiten (Treiber ohne Speicherfunktion, Latchregister, Flipflopregister). Abb. 4.72 zeigt einen Schaltkreis, der für beide Übertragungsrichtungen ausgelegt ist.

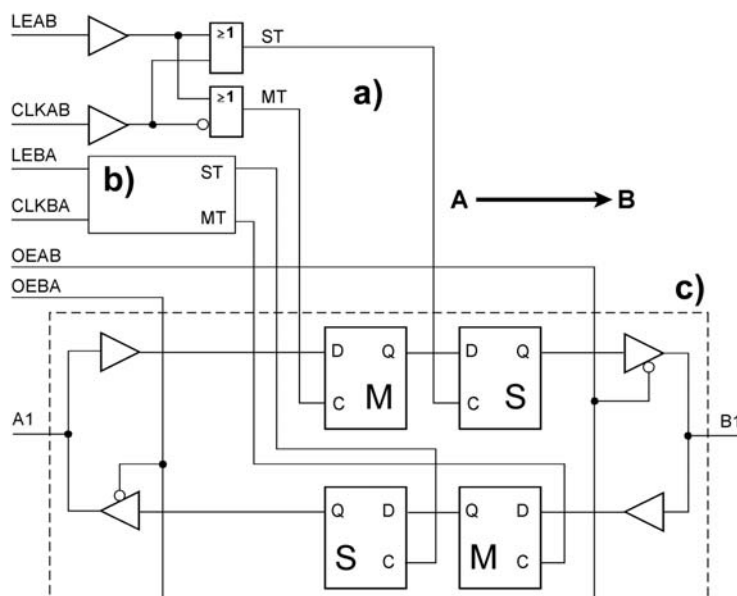


Abb. 4.72 Ein universeller Buskoppelschaltkreis (Universal Bus Transceiver; nach [2.10]). M - Master; S - Slave; MT - Master-Takt; ST - Slave-Takt. a) zentrale Steuerung für Richtung A-B; b) zentrale Steuerung für Richtung B-A (wie a) aufgebaut); c) eine Datenbitposition (von beispielsweise 18).

Die Universalität kommt durch trickreiches Ausnutzen der Eigenschaften von D-Latches zustande. Das soll anhand der Übertragungsrichtung von A nach B erläutert werden. Das Steuersignal LEAB entscheidet über die Funktionsweise:

- LEAB = 0: ST und MT hängen allein vom Takteingang CLKAB ab. Das Master-Latch wird aktiviert bei CLKAB = 0, das Slave-Latch bei CLKAB = 1.
- LEAB = 1: Über die ODER-Gatter werden sowohl ST als auch MT aktiv. Beide Latches wirken somit als Durchreiche (Ausgang folgt Eingang).
- Betrieb als Latchregister: durch Schalten von LEAB bei Festbelegung von CLKAB:
 - Wenn CLKAB = 0 ist, wirkt S als Latch, und M ist Durchreiche. ST schaltet mit LEAB, MT wird von CLKAB fest auf 1 gehalten.
 - Wenn CLKAB = 1 ist, wirkt M als Latch, und S ist Durchreiche. MT schaltet mit LEAB, ST wird von CLKAB fest auf 1 gehalten.

Initialisierung

Die üblichen Registerschaltkreise haben keine Rücksetzfunktion. Müssen die Ausgänge während des Hochlaufens der Versorgungsspannung definierte Belegungen annehmen, so kommen u. a. folgende Lösungen in Betracht:

- Beim Rücksetzen werden die Tri-State-Ausgänge deaktiviert (indem das Rücksetzsignal auf die OE-Eingänge einwirkt). Die Anfangsbelegung wird durch Pull-up- oder Pull-down-Widerstände oder durch anderweitiges Aufschalten eines Festwertes dargestellt.
- Die Registerfunktionen werden in programmierbaren Schaltkreisen implementiert. Außen werden keine Register, sondern nur kombinatorische Koppelstufen eingesetzt (oder Latchregister, deren Takteingang ständig aktiv gehalten wird).
- Das Problem wird umgangen, indem an anderen Stellen gewährleistet wird, dass die Anfangsbelegung keinen Schaden hervorruft, z. B. über Betriebsbereitschaftssignale, die nach der funktionellen (z. B. programmseitigen) Initialisierung aktiviert werden.

4.2.2 Registerentwurf

Register in programmierbaren und anwendungsspezifischen Schaltkreisen sind aus dem jeweiligen Funktionselementesortiment auszuwählen (Abb. 4.73) oder von Grund auf zu entwerfen. Zumeist genügt es, passende Latches oder Flipflops mit den gemeinsamen Signalen (Takt, Rücksetzen usw.) zu verbinden.

Übernahmesteuerung

Die Vorzugslösung ist der vollsynchroner Betrieb mit durchlaufendem Takt und Erlaubnissteuerung der Datenübernahme. Das Speicherelement des steuerbaren Datenregisters ist das DE-Flipflop. Zur Taktversorgung sollten die Taktressourcen des Schaltkreises ausgenutzt werden. Anwendungsspezifische Taktsignale sollten soweit möglich vermieden werden. Dass vollsynchroner Lösungen gelegentlich mehr Aufwand in der Kombinatorik erfordern, spielt praktisch keine Rolle, da die benötigten Funktionszuordner in den Zellen ohnehin vorhanden sind.

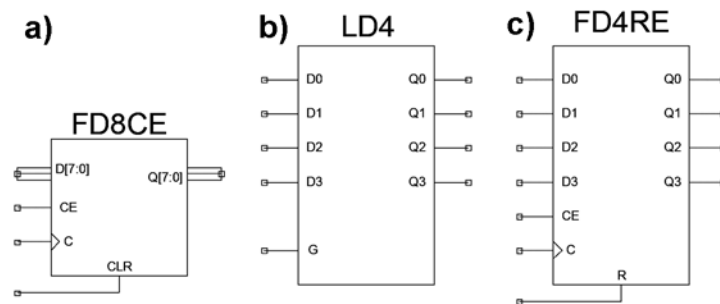


Abb. 4.73 Register als Funktionselemente. Eine kleine Auswahl (nach [S04]). a) Flipflopregister (acht Bits) mit Erlaubniseingang und asynchronem Löschen; b) Latchregister (vier Bits); c) Flipflopregister mit Erlaubniseingang und synchronem Rücksetzen (vier Bits).

Initialisierung

Das Problem erledigt sich, indem man entsprechende Funktionselemente auswählt oder das Register aus Flipflops mit geeigneten Setz- und Rücksetzfunktionen aufbaut.

Die Initialisierung eines beliebigen Bitmusters

Um anfänglich einen bestimmten Registerinhalt aus Einsen und Nullen einzustellen, kann man das Register aus Latches oder Flipflops mit Setz- und Rücksetzeingängen aufbauen (Abb. 4.74a).

Invertierte Initialisierung

Der Registerinhalt soll beispielsweise auf Eins (FF...FH) initialisiert werden, es sind aber nur Register verfügbar, die sich löschen lassen (00...0H). Der Ausweg: das Register an den Datenein- und Ausgängen mit Negatoren beschalten. Ein beliebiges Bitmuster lässt sich initialisieren, indem man nur die Bitpositionen, die auf Eins gesetzt werden sollen, mit Negatoren beschaltet (Abb. 4.74b).

4.2.3 Vollsynchrone Mehrfunktionsregister

Vollsynchrone Register können mehr als nur Daten speichern. Abb. 4.75 zeigt den grundsätzlichen Aufbau von Registern, die verschiedene Funktionen ausführen (Daten parallel übernehmen, in verschiedene Richtungen schieben, zählen usw.) Im allgemeinen Fall sind mehrere Funktionen F1, F2, F3 usw. vorgesehen, z. B. F1 = paralleles Laden, F2 = Linksschieben, F3 = Vorwärtzählen usw. Für n Funktionen ist jedem Flipflop ein Multiplexer oder Datenselektor mit n + 1 Eingängen vorgeschaltet. Der zusätzliche Eingang dient der Selbsthaltung. An die verbleibenden n Eingänge sind die jeweiligen funktionellen Verbindungen und Netzwerke angeschlossen (Dateneingänge, Ausgänge benachbarter Flipflops (zum Schieben), Zählnetzwerke usw.). Ist keine Funktion auszuführen, so ist die Selbsthaltung zu aktivieren. Vorrangregeln sind mit entsprechenden Zusatzbeschaltungen zu

implementieren¹⁾. Manchmal ist es von Vorteil, die kombinatorischen Verknüpfungen mit Schaltnetzen zu erledigen, die aus 2-zu-1-Multiplexern aufgebaut sind. Dann ergibt sich die Vorrangregelung aus der Reihenfolge der Multiplexer. Je näher der Multiplexer dem Flipflopeingang ist, desto höher ist die Priorität des Auswahlsignals.

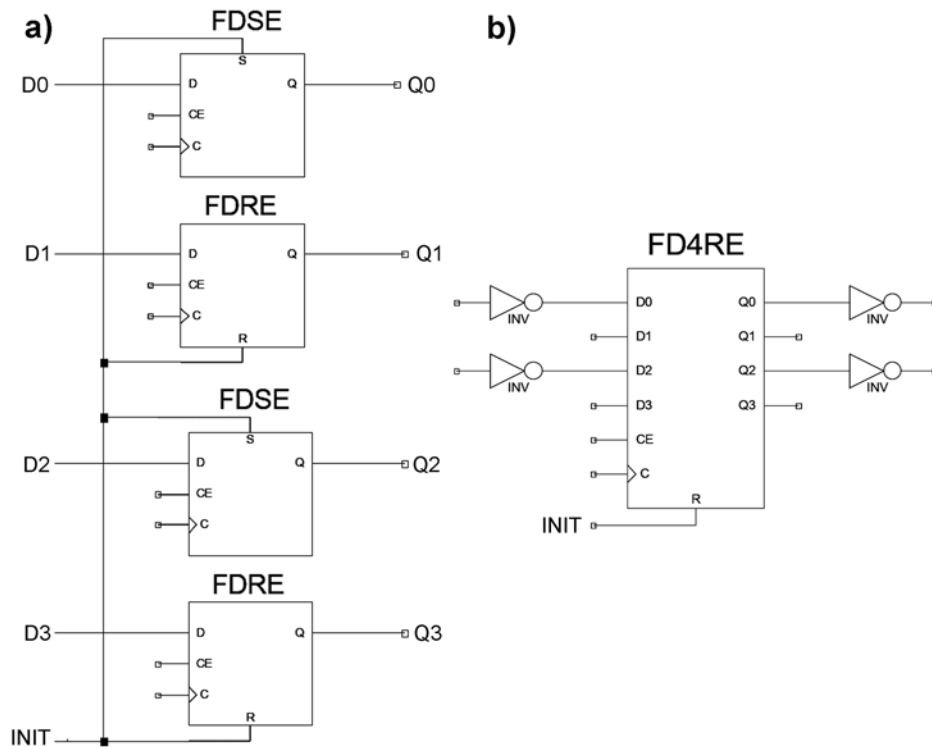


Abb. 4.74 Sonderfälle der Initialisierung. Im Beispiel sind die Bitpositionen Q3...Q0 auf 0101B = 5H zu initialisieren. a) das Register wird aus Flipflops mit Setz- und Rücksetzeingängen aufgebaut. b) dieses Register lässt sich nur löschen. Abhilfe: Die auf Eins zu initialisierenden Bitpositionen werden ein- und ausgangsseitig mit Negatoren beschaltet.

Werden DE-Flipflops eingesetzt, so entfällt die Rückführung (da sie schon im Innern der Flipflops vorgesehen ist). Stattdessen ist der Erlaubniseingang CE dann zu aktivieren, wenn eine Funktion auszuführen ist (z. B. über eine disjunktive Verknüpfung der Funktionsauswahlsignale; Abb. 4.76).

1: Das sind zumeist konjunktive Verknüpfungen mit den invertierten Signalen, die jeweils mit Vorrang wirken sollen (Inhibition).

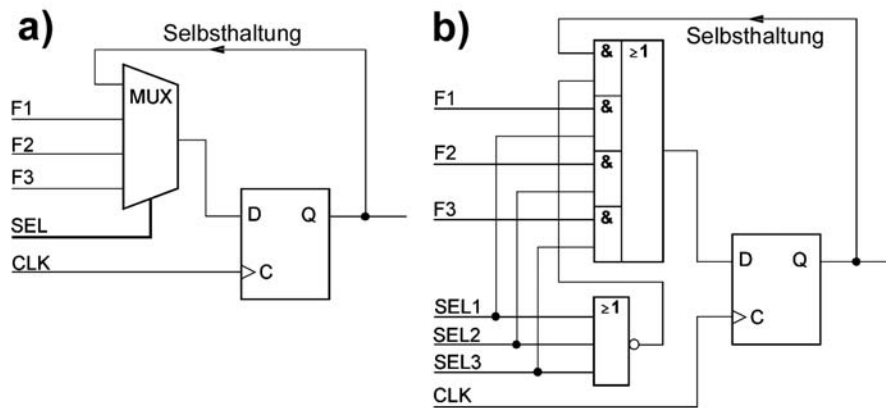


Abb. 4.75 Vollsynchrone Mehrfunktionsregister. a) mit Multiplexer (binär codierte Funktionsauswahl); b) mit Datenselektor (Funktionsauswahl 1 aus n). Die Funktionssignale F1, F2 usw. können Eingänge sein, aber auch Ausgänge anderer Flipflops oder Ausgänge von Verknüpfungsschaltungen, beispielsweise von Zählnetzwerken. Nähere Erläuterung im Text.

- Funktionsauswahl über Multiplexer. Sind die SELECT-Eingänge mit Nullen belegt, so geschieht nichts, und die Datenbelegung bleibt erhalten (Selbsthaltung). Ansonsten wählt jede SELECT-Belegung eine bestimmte Funktion aus.
- Funktionsauswahl über Datenselektor. Die Auswahleingänge SEL1, SEL2 usw. werden im 1-aus-n-Code angesteuert. Sind alle Auswahleingänge mit Nullen belegt, so wird über das NOR-Gatter die Rückführung aktiviert (Selbsthaltung).

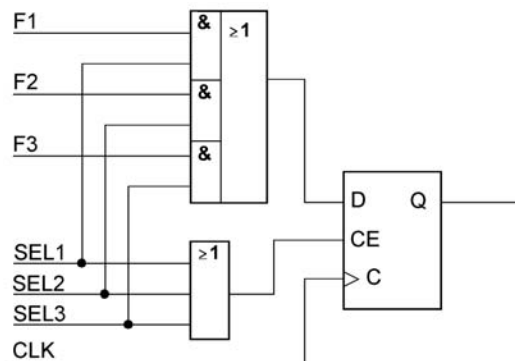


Abb. 4.76 Vollsynchrones Mehrfunktionsregister mit DE-Flipflops. Das Erlaubnissignal CE ist dann zu aktivieren, wenn eine Funktion auszuführen ist.

Die Abb. 4.77 bis 4.79 veranschaulichen das Prinzip des vollsynchronen Mehrfunktionsregisters anhand von zwei Beispielen. Beide Register haben acht Bitpositionen.

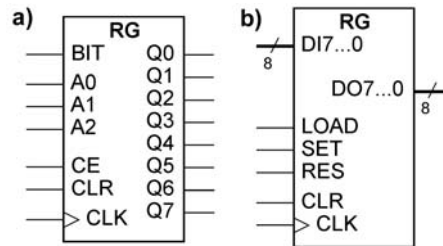


Abb. 4.77 Zwei Beispiele von Mehrfunktionsregistern. a) Einzelbitzugriff; b) selektives Setzen und Löschen.

Beispiel 1: Einzelbitzugriff (Abb. 4.77a und 4.78)

Über die Adresseingänge A2..A0 kann jede der acht Bitpositionen Q7...Q0 ausgewählt werden. Bei aktivem Erlaubniseingang CE wird die Belegung des Dateneingangs BIT in das ausgewählte Flipflop übernommen. Der Löscheingang CLR bewirkt ein synchrones Löschen des Registers. Diese LösCHFunktion wirkt dominierend. Die Selbsthaltebedingung ist wirksam, wenn CLR und CE beide inaktiv sind oder wenn CE aktiv, die betreffende Bitposition aber nicht ausgewählt ist.

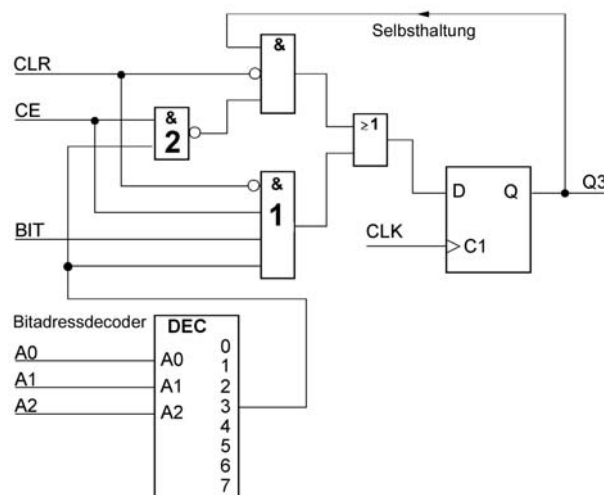


Abb. 4.78 Register mit Einzelbitzugriff. Jede Bitposition ist an den entsprechenden Ausgang des gemeinsamen Bitadressdecoders angeschlossen. Das Beispiel zeigt die Bitposition 3. 1 - Eingangsbelegung übernehmen; 2 - Selbsthaltung trennen, wenn Bitposition ausgewählt.

Beispiel 2: selektives Setzen und Löschen (Abb. 4.77b und 4.79)

Es gibt vier Funktionen: Gesamtlöschen (CLEAR), Laden (LOAD), selektives Setzen (SET) und selektives Löschen (RES). Beim selektiven Setzen und Löschen werden die Dateneingänge ausgewertet. Ist ein Dateneingang mit Eins belegt, so wird die betreffende Bitposition gesetzt oder gelöscht. Ansonsten bleibt sie unverändert. In der Schaltung von Abb. 4.79 wurde – der Überschaubarkeit wegen – auf eine Vorrangregelung verzichtet; es darf somit jeweils nur eines der Steuersignale aktiv sein. Ist LOAD aktiv, wird die Belegung

des jeweiligen Dateneingangs übernommen. Die Selbsthaltebedingung ist wirksam, wenn CLEAR und LOAD sowie die selektiven Funktionen SET und RES inaktiv sind. Eine selektive Funktion ist dann aktiv (also in der betreffenden Bitposition wirksam), wenn der zugehörige Dateneingang mit Eins belegt ist. Dann wird die Selbsthaltung getrennt. Ist RES aktiv, so liegt am D-Eingang des Flipflops eine Null an, ist SET aktiv, eine Eins.

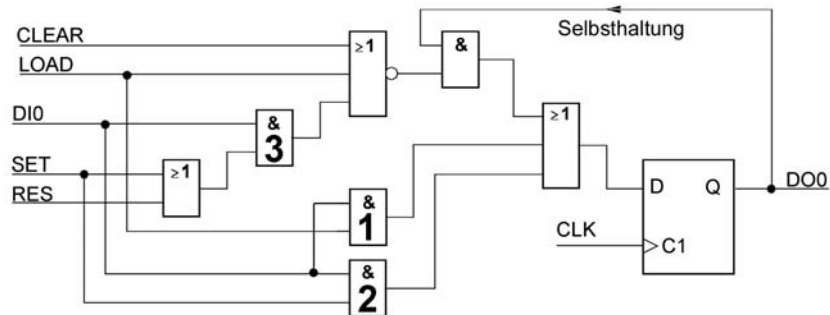


Abb. 4.79 Register mit selektivem Setzen und Löschen. Das Beispiel zeigt die Bitposition 0. 1 - Laden; 2 - Setzen; 3 - Selbsthaltung trennen, wenn SET oder RES aktiv.

4.3 Adressierbare Speicher

4.3.1 Grundlagen

Latches und Flipflops können mit kombinatorischen Schaltungen direkt verbunden werden. Alle Latches und Flipflops können gleichzeitig schalten; ihre Ausgangsbelegungen können gleichzeitig ausgewertet werden. Die Speicherzellen der adressierbaren Speicher sind hingegen nur über besondere Schnittstellen erreichbar. Speicher haben im Grunde nur drei Funktionen: Schreiben, Lesen und Daten halten. Der Datenerhalt ist die eigentliche Aufgabe der Speicherzelle, die beispielsweise als Latch (SRAM), als Kondensator (DRAM) oder als Feldeffekttransistor mit isoliertem Gate (EPROM, Flash-ROM) ausgebildet ist. Um Inhalte einzutragen (Schreiben) oder abzuholen (Lesen) ist jeweils eine Speicherzelle auszuwählen (Speicheradressierung). In elementaren Speicheranordnungen kann zu einer Zeit nur jeweils ein Zugriff (Schreiben oder Lesen) auf eine ausgewählte Speicherzelle ausgeführt werden. Abb. 4.80 veranschaulicht die einfachste Speicherschnittstelle.

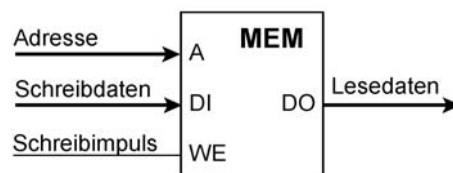


Abb. 4.80 Die einfachste Speicherschnittstelle.

Speicheradressierung

Alle Speicherzellen werden von Null an durchnummeriert. Die Adresse einer Speicherzelle ist deren laufende Nummer, die als vorzeichenlose Binärzahl codiert ist. Wird eine Adresse an die Speicherschnittstelle angelegt, so wird die betreffende Speicherzelle ausgewählt.

Speicherzugriffe:

- Schreiben: Adresse und Schreibdaten werden angelegt. Dann wird ein Schreibimpuls abgegeben.
- Lesen: Nach Anlegen der Adresse und Ablauf der Zugriffszeit liegt der Inhalt der adressierten Speicherzelle an den Lesedatenausgängen an.

Speichertypen

Die grundsätzlichen Speichertypen (ROM, SRAM, DRAM usw.; Abb. 4.81) unterscheiden sich vor allem im Aufbau der Speicherzellen. Die Speicherzellen sind in Speichermatrizen zusammengefasst. Der Speicher als Funktionselement oder Schaltkreis ist ein Verbund aus der Speichermatrix (oder aus mehreren Speichermatrizen) und dem jeweiligen Interface (Abb. 4.82). Es gibt asynchrone und synchrone Speicherinterfaces. Die Hersteller sind in der Lage, jede Speichertechnologie mit jedem Interface zu fertigen. Im Folgenden sollen nur Speicher betrachtet werden, die man ohne weiteren Zusatzaufwand in Digitalschaltungen einbauen oder auf programmierbaren Logikschaltkreisen anordnen kann. Das beschränkt die Auswahl auf Speicher mit elementaren Schnittstellen ähnlich Abb. 4.80 – also auf direkt ansteuerbare Speichermatrizen – und auf Technologien, die zur Fertigung programmierbarer Logikschaltkreise eingesetzt werden können.

RAM und ROM:

- RAM = Random Access Memory. Das ist der übliche Fachbegriff für Speicher, die für Lese- und Schreibzugriffe ausgelegt sind.
- ROM = Read-Only Memory. Diese Speicher haben einen festen Speicherinhalt (Festwertspeicher). Sie sind nur für Lesezugriffe ausgelegt. Der Speicherinhalt wird durch ein Programmierverfahren oder bei der Schaltkreisfertigung eingetragen (Maskenprogrammierung).

ROM-Schaltkreise mit Einfachschnittstellen ähnlich Abb. 4.80 kann man als RAMs ansehen, die keine Daten- und Schreibsteuereingänge haben. ROMs auf programmierbaren Logikschaltkreisen sind zumeist RAMs, die beim Einschalten mit ihrem Inhalt gefüllt werden (Konfigurieren), und deren Schreibfunktionen im Normalbetrieb abgeschaltet sind. Deshalb können sich die folgenden Erläuterungen auf statische RAMs (SRAMs) beschränken.

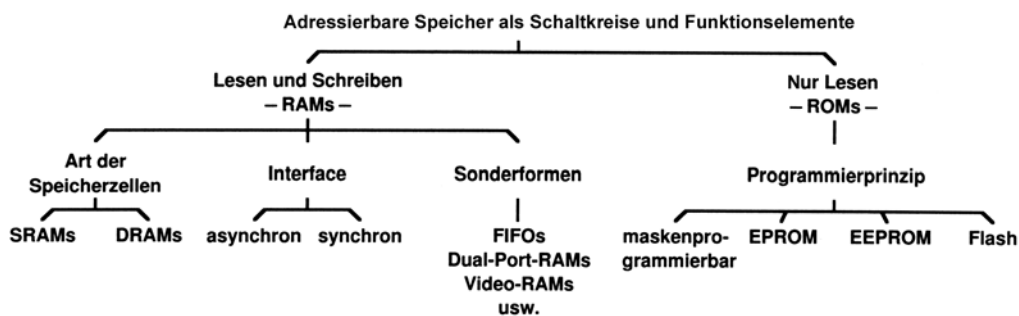


Abb. 4.81 Speichertypen.

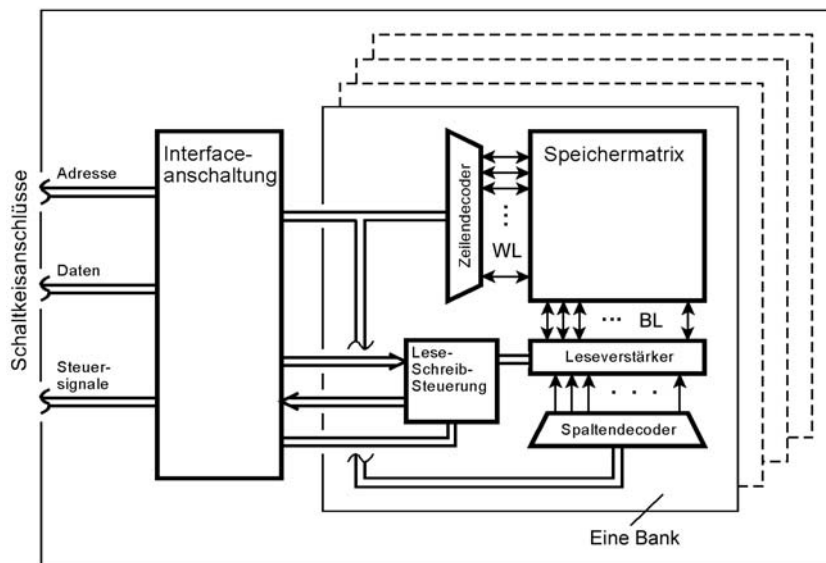


Abb. 4.82 Die grundsätzliche Struktur eines Speicherschaltkreises. WL - Wortleitungen; BL - Bitleitungen.

Speicherorganisation

Die elementaren Kennwerte der Speicherorganisation sind die Speicherkapazität und die Zugriffsbreite, also die Anzahl der Bits, die gleichzeitig geschrieben oder gelesen werden können. Diese Bits bilden das sog. Speicherwort. Die Angabe der Speicherkapazität betrifft entweder die Anzahl der Speicherzellen insgesamt oder die Anzahl der Speicherworte. Die Organisation wird oft in der Form Anzahl der Speicherworte • Zugriffsbreite angegeben. Beispiele: (1) 1M • 1-Organisation, (2) 256k • 4-Organisation. Beide Speicher haben eine Speicherkapazität von 1 MBits. Die Vorsätze der Speicherkapazitätsangaben stehen für Zweierpotenzen:

- $1k = 2^{10} = 1024$ (~ 1000).
- $1M = 2^{20} = 1\,048\,576$ (~ 1 Million).
- $1G = 2^{30} = 1\,073\,741\,824$ (~ 1 Milliarde).

Für Einfachschnittstellen ähnlich Abb. 4.80 gilt:

- Zugriffsbreite = Anzahl der Datenein- oder Ausgänge.
- Anzahl der Speicherworte = $2^{\text{Anzahl der Adresseingänge}}$.
- Anzahl der Adresseingänge = \log_2 Anzahl der Speicherworte.
- Speicherkapazität in Bits = Anzahl der Speicherworte • Zugriffsbreite.

Speichermatrizen

Große Speicherkapazitäten kann man nur verwirklichen, indem man den Schaltungsaufwand auf der Ebene der Halbleitertechnologie minimiert. Das betrifft sowohl die Speicherzellen als auch die Adressdecodierung. So bestehen die Speicherzellen der statischen RAMs aus Latches, die als optimierte Transistorschaltungen aufgebaut sind. DRAM-Zellen bestehen lediglich aus einem Speicherkondensator und einem Transistor, der zum Durchschalten des Signalwegs dient. Um den Aufwand für die Adressdecodierung zu verringern, ordnet man die Zellen in Matrixform an (Speichermatrix; Memory Array), so dass anstelle einer langen Adresse zwei Adressabschnitte zu decodieren sind, die Zeilen- und die Spaltenadresse.

In der Speichermatrix selbst werden keine einzelnen Zellen, sondern ganze Zeilen ausgewählt (vgl. Abb. 4.82). Der Zeilendecoder erregt je Zeile eine Zeilenauswahl- oder Wortleitung (Word Line). Eine aktive Wortleitung bewirkt, dass alle Speicherzellen der betreffenden Zeile parallel über Bitleitungen (Bit Lines) auf eine Leseverstärkeranordnung geschaltet werden. Eine Speicheranordnung mit 1024 Zeilen und 1024 Spalten hat also 1024 Wortleitungen und 1024 über Bitleitungen angeschlossene Leseverstärker. Der Spaltendecoder wählt den Leseverstärker der jeweils adressierten Spalte aus. Die Bitleitungen werden sowohl zum Lesen als auch zum Schreiben ausgenutzt²⁾.

Multiplexer als Speicher

Kleine Speicher können mit Multiplexern aufgebaut werden³⁾. Die Beschaltung der Dateneingänge mit Festwerten ergibt einen ROM, das Anschließen an Flipflops einen RAM.

Dual-Port-RAMs

Dual-Port-RAMs (DPRAMs) sind statische RAMs mit zwei unabhängigen Adress- und Zugriffswegen (Abb. 4.83). Von beiden Seiten aus gesehen stellt sich der DPRAM so dar wie ein SRAM mit einer Schnittstelle ähnlich Abb. 4.80. Der gesamte Speicher ist von Grund auf für den Zweifachzugriff ausgelegt: die Adressdecoder, Leseverstärker usw. (vgl. Abb. 4.81) sind doppelt vorhanden, und in der Speichermatrix sind zwei Zugriffswege an die Speicherzellen geführt.

2: Der Verbund aus Spaltendecoder und Leseverstärkern ist (über entsprechende Umschaltvorkehrungen) für Signalflüsse in beiden Richtungen nutzbar.

3: Die Funktionszuordner (Lookup Tables; LUTs) in den Zellen mancher FPGAs sind im Grunde Multiplexer, die an Flipflops angeschlossen sind. Solche Zuordner kann man auch als Multiplexer, Schieberegister oder kleine adressierbare Speicher konfigurieren.

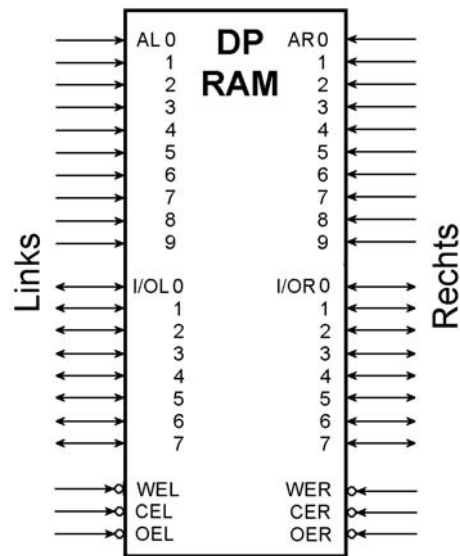


Abb. 4.83 Dual-Port-RAM. Die Schnittstellen beider Seiten entsprechen denen der typischen asynchronen SRAM-Schaltkreise (s. Abb. 4.87).

FIFOs

Ein FIFO ist ein Dual-Port-RAM mit eingebauter Adressierung, der auf der einen Seite nur für Schreibzugriffe und auf der anderen nur für Lesezugriffe ausgelegt ist (Abb. 4.84). Schreibzugriffe füllen den Speicher, Lesezugriffe leeren ihn. Dabei kann nur der jeweils gefüllte Bereich gelesen werden, wobei die Daten in der gleichen Reihenfolge gelesen werden, in der sie geschrieben wurden (die zuerst geschriebenen Daten werden auch zuerst gelesen; FIFO = First In – First Out). Die wichtigste Anwendung ist der Zwischenpuffer, der zwischen Einrichtungen angeordnet ist, die zwar grundsätzlich (im zeitlichen Mittel) mit der gleichen Datenrate zurechtkommen, wo aber der zeitliche Abstand zwischen den einzelnen Übertragungszyklen von Schwankungen betroffen ist.

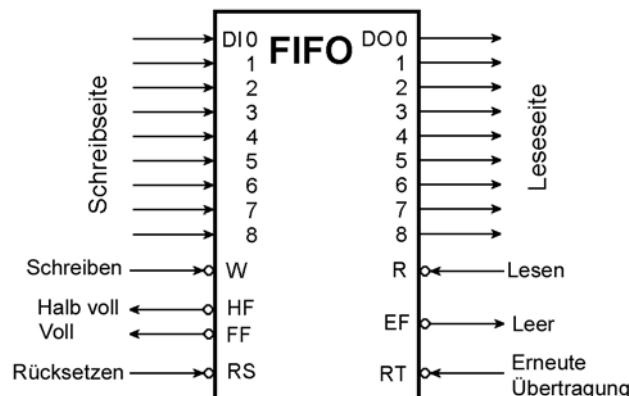


Abb. 4.84 FIFO mit asynchronen Schnittstellen (vereinfacht).

4.3.2 Registersätze

Ein Registersatz (Register File) ist eine adressierbare Anordnung gleichartiger Register. Registersätze können als reine Logikschaltungen aufgebaut werden (Abb. 4.85 und 4.86). Solche Anordnungen weisen sehr kurze Zugriffszeiten auf. Sie lassen sich problemlos in das Taktraster der umgebenden Digitalschaltungen einfügen. Schreib- und Leseadressen können unabhängig voneinander sein. Auch ist es offensichtlich nicht schwierig, mehrere Lesezugriffswege (Multiplexer zum Auswählen der Registerausgänge) vorzusehen⁴⁾. Aus Aufwandsgründen kommen solche Anordnungen aber nur für geringe Speicherkapazitäten (Richtwert: 4...64 Register) in Betracht.

*:

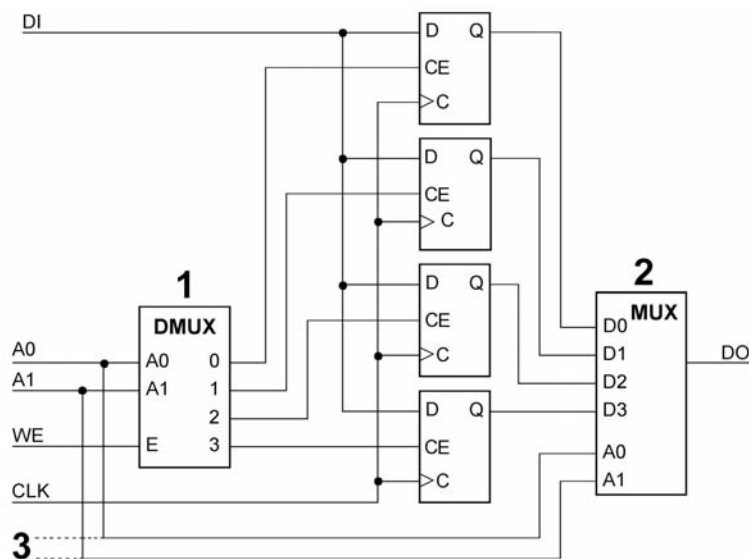


Abb. 4.85 Registersatz aus Flipflopregistern. Im Beispiel werden DE-Flipflops eingesetzt. Es ist nur eine Bitposition dargestellt. Die Schnittstelle entspricht Abb. 4.80, hat aber einen zusätzlichen Takteingang (synchrone Schnittstelle). Die Flipflopregister können in rückgekoppelte Schaltungsanordnungen einbezogen werden. 1 - Schreibadressdecodierung; 2 - Leseadressdecodierung und Lesedatenauswahl; 3 - unabhängige Leseadresse (wahlweise).

4: Mehrere Schreibzugriffswege erfordern Auswahl-schaltungen für die Daten-, Adress- und Schreibsteuersignale sowie entsprechende Vorkehrungen zur Vermittlung der Vorrangsteuerung.

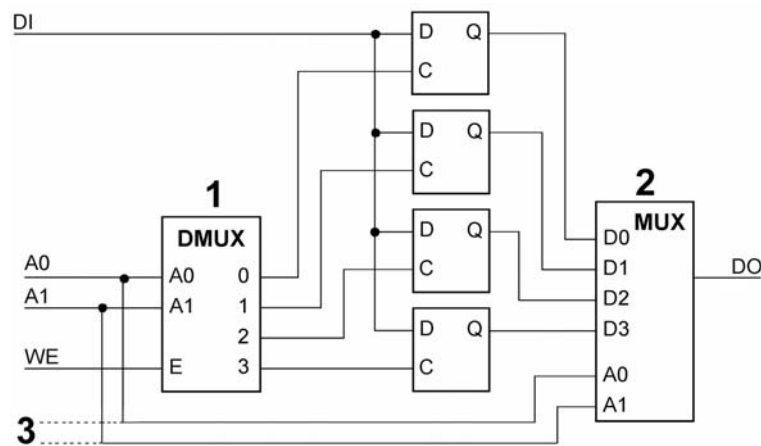


Abb. 4.86 Registersatz aus Latchregistern. Der Schreibsteuereingang WE wirkt als Takteingang (asynchrone Schnittstelle gemäß Abb. 4.80). Es ist nur eine Bitposition dargestellt. 1 - Schreibadressdecodierung; 2 - Leseadressdecodierung und Lesedatenauswahl; 3 - unabhängige Leseadresse (wahlweise).

4.3.3 Speicherschaltkreise

Für die hier in Rede stehenden Einsatzfälle (unmittelbare Einbeziehung in Logikschaltungen) kommen vor allem Schaltkreise mit asynchronen und synchronen SRAM-Schnittstellen in Betracht. Mit solchen Schnittstellen werden u. a. auch RAMs mit Datenerhalt (SRAM + Batterie, SRAM + EEPROM, ferroelektrische RAMs usw.) sowie DRAM- und Flash-ROM-Schaltkreise angeboten.. Der grundsätzliche JEDEC-Standard für Speicherschnittstellen: JESD-21C.

Asynchrone SRAM-Schnittstellen

Für die Auslegung der SRAM-Schnittstellen war vor allem maßgebend, die Schaltkreise ohne Zusatzaufwand an herkömmliche Mikroprozessor-Bussysteme anschließen zu können. Deshalb haben die Schaltkreise bidirektionale Datenanschlüsse, und die Zugriffsbreiten entsprechen den Datenwegbreiten der üblichen Bussysteme. Typische Organisationsformen asynchroner SRAMs: 8k • 8; 32k • 8; 128k • 8; 128k • 16; 256k • 16; 512k • 16; 1M • 16.

Die Einfachschnittstelle von Abb. 4.80 wurde um ein Lesesteuersignal OE und ein Schaltkreisauswahlsignal CE (oder CS) erweitert (Abb. 4.87 und 4.88). Tabelle 4.7 gibt an, wie die Steuersignale wirken. Die Abb. 4.89 und 4.90 veranschaulichen typische Zugriffsabläufe.

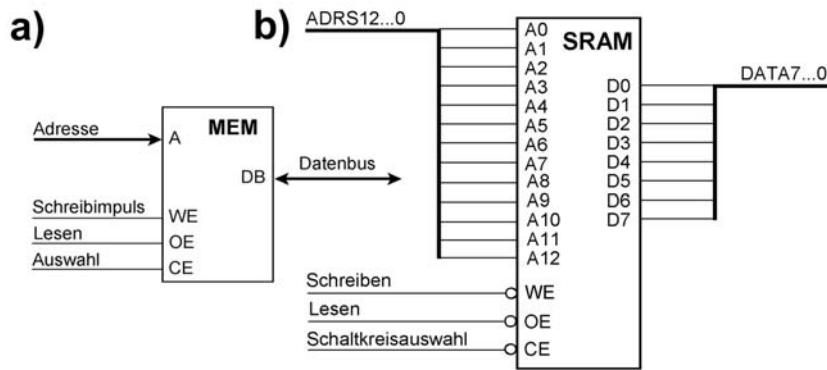


Abb. 4.87 Die asynchrone SRAM-Schnittstelle – ein Industriestandard. a) Prinzip; b) Praxisbeispiel (SRAM 8k • 8).

Funktion	\overline{CE}	\overline{OE}	\overline{WE}	Daten (I/O)	Stromaufnahme (I_{CC})
Ruhezustand (Daten halten)	High	X	X	Hochohmig	Standby-Strom
Lesen	Low	Low	High	Gelesene Bits (Low oder High)	Betriebsstrom
Datenbus freigeschaltet	Low	High	High	Hochohmig (kann von außen mit Schreibdaten belegt werden)	Betriebsstrom
Schreiben	Low	X	Low	Von außen mit Schreibdaten belegt	Betriebsstrom

X = beliebige Belegung.

Tabelle 4.7 Zur Wirkungsweise der Steuersignale. Die Adresssignale müssen während des gesamten Speicherzyklus stabil anliegen.

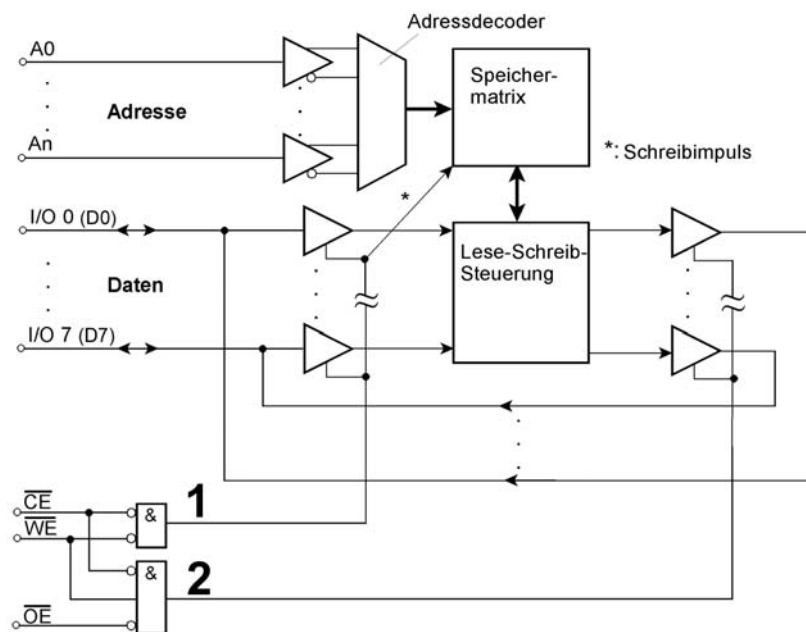


Abb. 4.88 Ein typischer SRAM-Speicherschaltkreis. 1 - Schreibzugriff, 2 - Lesezugriff.

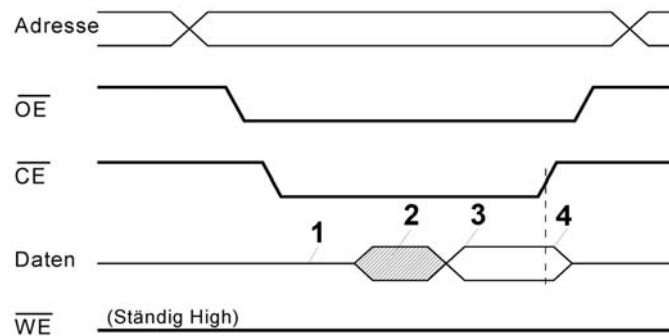


Abb. 4.89 Lesezugriff (Ablaufbeispiel). 1 - die Datenausgänge sind zunächst hochohmig; 2 - die Datenausgänge werden aktiv; 3 - der Inhalt der adressierten Speicherzelle erscheint auf den Datenleitungen; mit dem Inaktivwerden von CE oder OE werden die Datenausgänge wieder hochohmig.

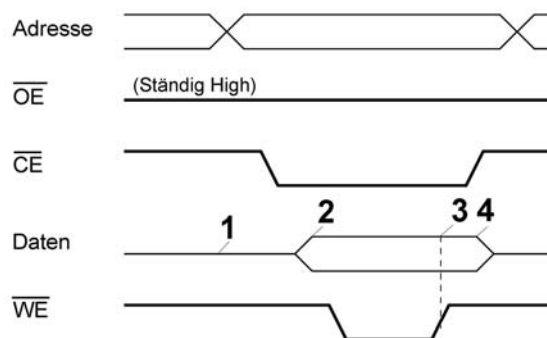


Abb. 4.90 Schreibzugriff (Ablaufbeispiel). 1 - die Datenausgänge müssen hochohmig sein, um die Schreibdaten aufschalten zu können; 2 - Aufschalten der Schreibdaten; 3 - mit der Rückflanke des Schreibimpulses werden die Schreibdaten in die adressierte Speicherzelle übernommen; 4 - die Schreibdaten werden abgeschaltet.

Zwei Schaltkreisauswahlsignale

Zwei Schaltkreisauswahlsignale (Chip Selects), die konjunktiv verknüpft sind, ermöglichen gelegentlich eine einfachere Außenbeschaltung (Vereinfachung der Adressdecodierung). Abb. 4.91 veranschaulicht die typische Auslegung. Das eine Signal ist aktiv Low ($\overline{\text{CS1}}$), das andere aktiv High (CS2). Der einfachste Nutzungsfall: besteht das Speichersubsystem aus zwei derartigen Schaltkreisen, so wird das niedrigstwertige Adresssignal an $\overline{\text{CS1}}$ des ersten und an CS2 des zweiten Schaltkreises angeschlossen. Die verbleibenden Signale werden mit Festwerten beschaltet (CS2 = High, $\overline{\text{CS1}}$ = Low).

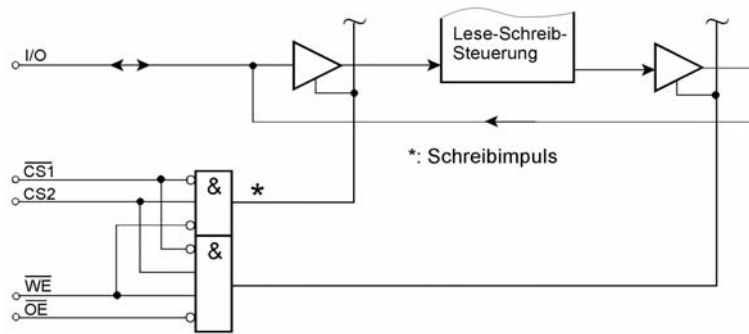


Abb. 4.91 SRAM-Steuerung mit zwei Schaltkreisauswahlsignalen (vgl. beispielsweise [4.16]).

Breite SRAMs: die •16-Organisation

SRAMs in •16-Organisation sind so ausgelegt, dass sich die beiden Bytes, aus denen das 16-Bit-Wort besteht, einzeln lesen und schreiben lassen (byteselektive Zugriffe). Hierfür sind zusätzlich zu den Steuersignalen CS, WE und OE zwei Byteerlaubnisgänge UB und LB vorgesehen (Upper Byte, Lower Byte). Lese- und Schreibzugriffe zur jeweiligen Byteposition werden nur dann ausgeführt, wenn das betreffende Byteerlaubnisignal aktiv ist (Abb. 4.92).

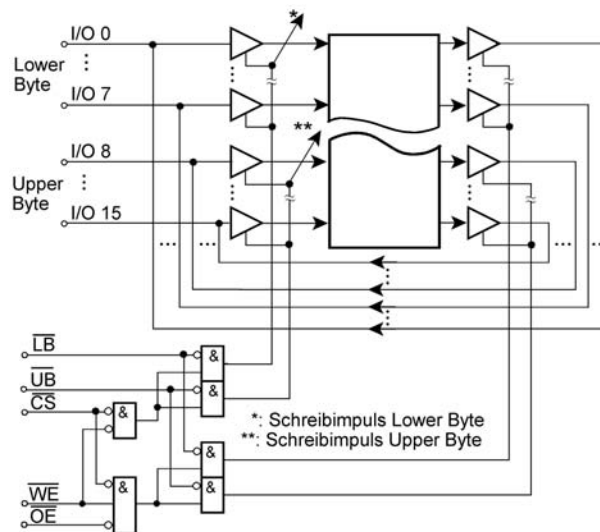


Abb. 4.92 Datenweg und Steuerung eines SRAM in •16-Organisation (vgl. beispielsweise [4.17]).

Pseudostatische (mobile) SRAMs

Die weitaus meisten SRAM-Schaltkreise sind für mobile Telekommunikationsgeräte bestimmt. Moderne Mobiltelefone u. dergl. kommen aber mit vergleichsweise bescheidenen Speicherkapazitäten (z. B. 128 kBytes) längst nicht mehr aus. Deshalb werden Schaltkreise gefertigt, die typische SRAM-Schnittstellen aufweisen, aber Speicherkapazitäten von mehreren MBytes haben (Beispiele: 2M • 16 (32 MBits), 4M • 16 (64 MBits), 8M • 16 (128

MBits), $16\text{M} \cdot 16$ (256 MBits))⁵⁾. Diese Schaltkreise enthalten DRAM-Speicherzellen, können aber wie SRAMs betrieben werden (die für DRAMs typischen Refresh-Abläufe werden intern erledigt).

Synchrone SRAM-Schnittstellen

Synchrone Speicherschaltkreise enthalten Register für Adressen, Daten und Steuersignale (Abb. 4.93). Hierdurch werden alle Übertragungsvorgänge auf einen gemeinsamen Takt bezogen. Die mit der Low-High-Flanke des Taktes übernommenen Steuersignalbelegungen bestimmen, was im aktuellen Taktzyklus getan wird (Tabelle 4.8). Die Steuersignale haben zwar noch die von den asynchronen Speichern her bekannten Namen, dienen aber praktisch dazu, ein Kommando in den Speicherschaltkreis zu übertragen. Synchrone SRAMs wurden ursprünglich vor allem für externe Caches von Personalcomputern entwickelt. Aus diesem Einsatzfall haben sich Funktionsmerkmale ergeben, die bis heute zum Industriestandard gehören. Neuere Typen werden zumeist als Zwischenspeicher in Telekommunikations- und Netzwerkeinrichtungen eingesetzt. Der folgende Überblick beschränkt sich auf herkömmliche Typen und elementare Einsatzfälle (wahlfreie Zugriffe im direkten Zusammenwirken mit Logikschaltungen). Typische Organisationsformen synchroner SRAMs: $128\text{k} \cdot 32$, $128\text{k} \cdot 36$, $256\text{k} \cdot 18$.

Flow-Thru-Organisation

Nur die Eingänge der Speicherschnittstelle sind auf Register geführt. Bei Lesezugriffen verhält sich ein solcher Speicher ähnlich wie ein asynchroner SRAM: die Adresse wird mit der jeweiligen Taktflanke übernommen, und nach Ablauf der Zugriffszeit stehen die gelesenen Daten an den I/O-Anschlüssen zur Verfügung. Flow-Thru-Speicher liefern die gelesenen Daten im selben Zyklus, in dem der Zugriff gestartet wurde. Der wesentliche Kennwert ist die Zugriffszeit t_{CDV} (Data Output Valid after Clock Rise = von der Low-High-Flanke des Taktes bis zum Erscheinen der Daten an den Ausgängen).

Pipelining-Organisation

Im Unterschied zur Flow-Thru-Organisation ist auch in den Lesedatenweg ein Register eingefügt (Datenausgangsregister; vgl. Abb. 4.93). Die gelesenen Daten erscheinen somit erst nach der jeweils nächsten Low-High-Flanke des Taktes an den I/O-Anschlüssen. Da die gelesenen Daten einen Taktzyklus später ausgegeben werden, steht für den eigentlichen Zugriff (auf die Speichermatrix) nahezu die gesamte Taktperiode zur Verfügung. Der wesentliche Kennwert ist die maximale Taktfrequenz oder die minimale Periodendauer t_{CYC} (Clock Cycle Time).

5: Solche Schaltkreise werden auch als Mobile Specified RAM o. dergl. bezeichnet. Aus dem Datenmaterial sind auch die Spitzfindigkeiten ersichtlich (Page Mode, Standby-Betrieb usw.).

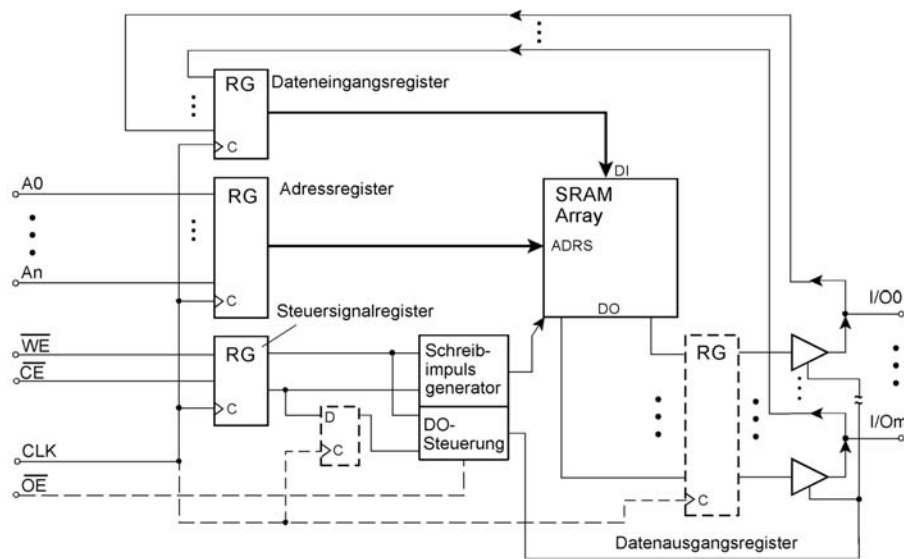


Abb. 4.93 Synchroner SRAM-Schaltkreise. Flow-Thru-Organisation ohne, Pipelining-Organisation mit Datenausgangsregister und CE-Verzögerung.

Zugriff	Signalbelegung vor Taktflanke	Wirkung
Lesen (Flow Thru)	<ul style="list-style-type: none"> An...A0 = Zugriffsadresse, WE inaktiv (High), CE aktiv (Low). 	Adresse und Steuersignale werden in die jeweiligen Register übernommen. Nach Ablauf der spezifizierten Zugriffszeit können die gelesenen Daten an den I/O-Anschlüssen abgenommen werden.
Lesen (Pipelining)	<ul style="list-style-type: none"> An...A0 = Zugriffsadresse, WE inaktiv (High), CE aktiv (Low). 	Adresse und Steuersignale werden in die jeweiligen Register übernommen. Mit der Low-High-Flanke des folgenden Taktimpulses werden die gelesenen Daten in das Datenausgangsregister übernommen und erscheinen somit an den I/O-Anschlüssen.
Schreiben	<ul style="list-style-type: none"> An...A0 = Zugriffsadresse, WE aktiv (Low), CE aktiv (Low), I/Om...I/O0 = zu schreibende Daten. 	Adresse, Daten und Steuersignale werden in die jeweiligen Register übernommen. Die Daten werden in das SRAM-Array geschrieben. Hierzu wird intern ein Schreibimpuls erzeugt. Das eigentliche Schreiben ist vom Takt unabhängig. Somit ist es nicht notwendig, beim der Festlegung des Taktrasters auf Schreibimpulsbreiten usw. Rücksicht zu nehmen.
Daten halten	<ul style="list-style-type: none"> CE inaktiv (High). 	Keine. Datenanschlüsse hochohmig.

Tabelle 4.8 Zur Wirkungsweise der synchronen SRAM-Schaltkreise.

Buskonflikte beim Umschalten zwischen Lesen und Schreiben (Bus Turnaround)

Die bidirektionalen Datenleitungen müssen zwischen beiden Zugriffsweisen umgeschaltet werden. Abb. 4.94 veranschaulicht die schnellstmögliche Folge, in der Lese- und

Schreibzugriffe einander abwechseln können.

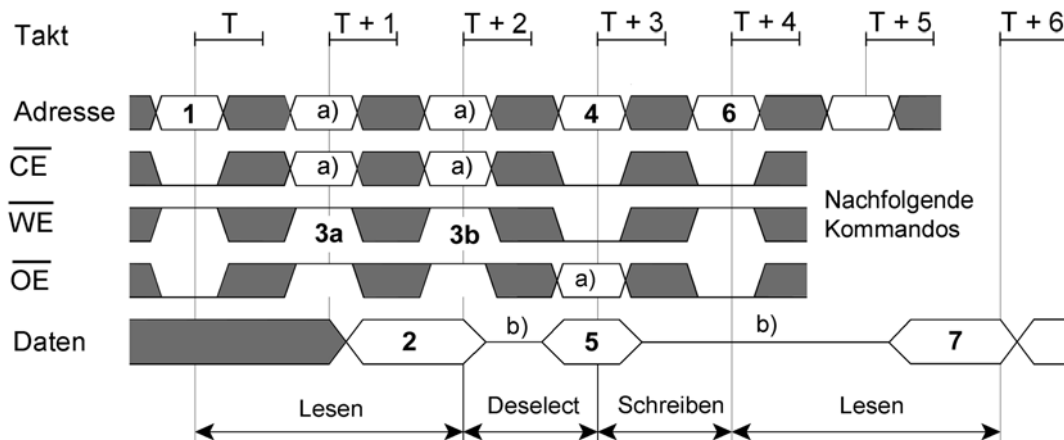


Abb. 4.94 Beispiele abwechselnder Lese- und Schreibzugriffe (Pipelining-Organisation). 1 - Lesebefehl; 2 - Leseergebnisse; 3a - Deselect-Befehl (wirkt ab $T+2$); 3b) weiteres Deselect-Befehl, da der Datenbus erst zu $T+3$ befreit werden kann (Wartezustand); 4 - Schreibbefehl; 5 - Schreibeergebnisse; 6 - zweites Lesebefehl; 7 - Leseergebnisse. Gelingt es nicht, die Schreibergebnisse zu $T+3$ aufzuschalten, sind weitere Wartezustände 3b einzufügen. a) beliebige, aber stabile Belegung; b) Datenbus hochohmig. Grau hinterlegte Bereiche: beliebige Belegung.

Erst lesen, dann schreiben

Nach dem Lesen müssen die Datenleitungen zunächst freigeschaltet werden. Hierzu ist für wenigstens einen Taktzyklus OE auf High zu schalten, wobei WE inaktiv ist (Deselect-Befehl)*.

Erst schreiben, dann lesen

Auf einen Schreibzyklus darf sofort ein Lesezugriff folgen. Wichtig ist, dass der Datenbus hochohmig wird, bevor die Leseergebnisse erscheinen.

*: Zu Spitzfindigkeiten und Sonderlösungen (Single Cycle Deselect SCD, Double Cycle Deselect DCD, frühes und spätes Schreiben (Early/Late Write), Zero Bus Turnaround ZBT, No Bus Latency NoBL) muss auf die Datenblätter und Anwendungsschriften der Hersteller verwiesen werden (vgl. beispielsweise [4.23] bis [4.29]). Das gilt sinngemäß für Typen mit doppelter oder vierfacher Datenrate (DDR / QDR), für synchrone Dual-Port-RAMs, FIFOs usw.

4.3.4 Funktionselemente

Größere programmierbare Schaltkreise sind mit Speicherblöcken (Block RAM und Flash ROM) ausgerüstet. Zudem können die RAM-Funktionszuordner der FPGA-Zellen als Speicher mit wahlfreiem Zugriff verwendet werden. Beide Arten von Speichereinrichtungen lassen sich als RAM oder als ROM einsetzen und nach dem Zuschalten der

Versorgungsspannung beim anfänglichen Konfigurieren mitladen. Im Gegensatz zu den Speicherschaltkreisen haben die Speicherelemente typischerweise getrennte Datenein- und Ausgänge. Die Einzelheiten sind hersteller- und baureihenspezifisch. Deshalb muss sich die folgende Darstellung auf einen allgemeinen Überblick beschränken.

Dass man Speicherkapazität bedarfsweise mitten in der Logik einsetzen kann – und dass die Speicheranordnungen mit dem Taktraster der Logik mithalten – eröffnet eine Vielzahl von Möglichkeiten und Alternativen der Systemauslegung und des Schaltungsentwurfs. Hier sollen lediglich einige Anregungen genannt werden:

- Implementierung von Sonderausführungen, wie beispielsweise FIFOs, RAMs mit mehreren Zugriffswegen, Ringpuffern usw. Eine solche Lösung (Ansteuerung + Speicher gemeinsam in einem FPGA) ist der herkömmlichen Kombination von Speicher- und Logikschaltkreisen deutlich überlegen – es entfallen u. a. die Probleme der Gehäusebauformen und der Signalpegel sowie der Beschaffbarkeit spezieller Speicherschaltkreise (Speciality RAMs).
- Funktionszuordner zur Implementierung komplexer Schaltfunktionen und Zustandsautomaten (die folgenden Kapitel enthalten mehrere Beispiele).
- Tabellenspeicher, die es ermöglichen, zeitaufwendige Rechenabläufe durch das Nachsehen in fertig berechneten Wertetabellen zu ersetzen.
- Steuerspeicher zur Implementierung von Zustandsautomaten und Mikroprogrammsteuerungen (Kapitel 6****).
- Universalprozessoren mit mehreren getrennten Speichern, die den einzelnen Funktionen direkt zugeordnet sind (Programmspeicher, Datenspeicher, Stack usw. – wobei auch an eigene Speicher für jede Task zu denken wäre (Hardware-Multitasking)).
- Spezialprozessoren mit Speicheranordnungen, die dem jeweiligen Problem angepasst sind (Datenstrukturmaschinen).
- Virtuelle Funktionen (Abschnitt 4.4) als Verallgemeinerung des als Hardware-Multitasking bekannten Prinzips.

Solche Entwurfsprinzipien sind vor allem von Interesse, um die Kosten und die Stromaufnahme gering zu halten (möglichst kleine Siliziumfläche, möglichst wenige Schaltvorgänge). Durchgehende (gleichsam aus einem Guss bestehende) Hardware-Software-Verbundlösungen ergeben zudem einen beachtlichen Grad an Unabhängigkeit. Werden die anwendungsseitigen Funktionen in entsprechenden Entwurfssprachen beschrieben (Verhaltensbeschreibung), so ist es unproblematisch, sie an die jeweilige Aufgabenstellung genau anzupassen und immer wieder auf neue Schaltkreisbaureihen zu übertragen.

Geringe Speicherkapazitäten

Die Speicherkapazitäten der Speicherblöcke beginnen bei 16 Bits oder Speicherworten. Geringe Speicherkapazitäten (Richtwert: 4...64 Speicherworte) kann man alternativ mit Registern darstellen, Festwertspeicher auch mit Multiplexern, deren Dateneingänge entsprechend belegt werden. Solche Anordnungen können gelegentlich mit höheren

Taktfrequenzen betrieben werden. Auch lassen sich manche Sonderaufgaben vergleichsweise einfach lösen (mehrere Zugriffswege, FIFOs, Ringpuffer usw.). Ob eine derartige Lösung weniger Aufwand kostet als die Nutzung der bereitgestellten Speicherblöcke, ist oftmals nur anhand von Probeentwürfen herauszufinden⁶⁾.

Verteilter RAM (Distributed RAM)

Die Funktionszuordner der Zellen werden als Speicher eingesetzt, wobei jede Zelle eine oder zwei Bitpositionen enthält. Typische Organisationsformen: $16 \cdot 1$; $16 \cdot 2$; $32 \cdot 1$; $32 \cdot 2$; $64 \cdot 1$; $128 \cdot 1$; $256 \cdot 1$; $64 \cdot 4$. Aus solchen Speichern können Speicheranordnungen mit beliebiger Zugriffsbreite aufgebaut werden. Die Grenzen sind aus der jeweiligen Schaltungsdokumentation ersichtlich (wichtig ist allem, auf wie viele Bits parallel zugegriffen werden kann, ohne globale Verbindungsressourcen zu beanspruchen). Abb. 4.95 veranschaulicht ein Beispiel. Die Funktionselemente haben eine einfache Schnittstelle ähnlich Abb. 4.80. Schreibzugriffe laufen synchron ab, Lesezugriffe asynchron (das Taktsignal ist nur für die Schreibzugriffe von Bedeutung). Synchroner Lesezugriff ergibt sich, wenn man den Datenausgang mit dem Flipflop der betreffenden FPGA-Zelle verbindet. Beim Schreiben erscheinen die Schreibdaten auf den Datenausgängen. Die Ausführung gemäß Abb. 4.95b hat einen zweiten unabhängigen asynchronen Lesezugriffsweg. Die RAMs können beim Konfigurieren mit einem bestimmten Inhalt geladen werden (ansonsten wird der Inhalt Null konfiguriert). Wird der WE-Eingang fest auf Low gehalten, wirkt das Funktionselement als asynchroner ROM.

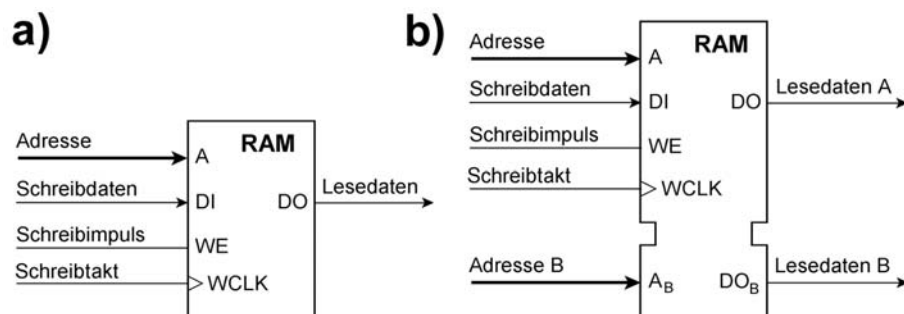


Abb. 4.95 Verteilter RAM (Funktionselemente nach [2.7]). a) ein Zugriffsweg (Single-Port); b) zwei Zugriffswege (Dual-Port).

Eingebaute Speicherblöcke (Block RAM)

Block RAMs sind "echte" Speichermatrizen mit programmierbarer Konfiguration. Beispiel von Organisationsformen: $16k \cdot 1$; $8k \cdot 2$; $4k \cdot 4$; $2k \cdot 8$; $2k \cdot 9$; $1k \cdot 16$; $1k \cdot 18$; $256 \cdot 18$; $512 \cdot 32$; $512 \cdot 36$; $256 \cdot 72$; $4k \cdot 144$ (eine Zusammenstellung aus [2.3] bis [2.7]). Abb. 4.96 veranschaulicht ein Beispiel. Die hier gezeigten Funktionselemente beruhen auf einer Speichermatrix von 18 432 Bits und zwei unabhängigen synchronen Zugriffswegen (Dual-

6: Es kommt u. a. auf das Entwicklungssystem an – und auch darauf, wie man die Entwurfsabsicht in der jeweiligen Beschreibungssprache ausgedrückt hat.

Port Memory). Speicher mit einem Zugriffsweg (Single-Port; Abb. 4.96a) ergeben sich durch entsprechendes Konfigurieren. Aus derartigen Elementen können – über besondere Verbindungsressourcen – Speicheranordnungen mit mehr und/oder breiteren Speicherworten aufgebaut werden. Die Block RAMs anderer Schaltkreisfamilien sind ähnlich ausgelegt. Die Unterschiede betreffen vor allem die Speicherkapazität, die Konfigurationsvarianten und die Sonderfunktionen (Initialisierung, Rücksetzen, byteselektive Zugriffe, zusätzliche Steuersignale, wählbare Polarität usw.).

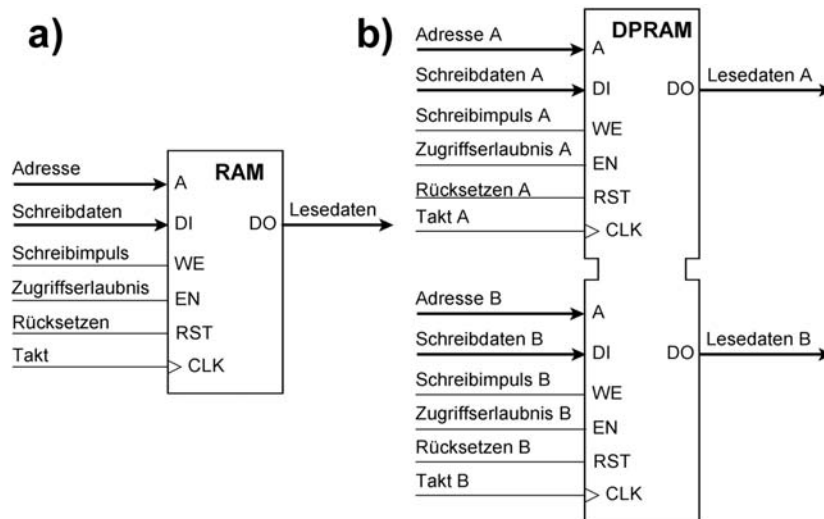


Abb. 4.96 Block RAM (Funktionselemente nach [2.7]). a) ein Zugriffsweg (Single-Port); b) zwei Zugriffswege (Dual-Port). Zu typischen Signalverläufen s. die Abb. 4.97 bis 4.99.

Schreiben und Lesen

Da es getrennte Datensignale gibt, kann gleichzeitig geschrieben und gelesen werden. Hierfür lassen sich mehrere Vorrangregeln konfigurieren. Diese werden im Folgenden anhand eines Zugriffsbeispiels erläutert (Abb. 4.97 bis 4.99). Zunächst findet ein Lesezugriff mit einer Adresse a_1 statt. Dann folgen zwei Schreibzugriffe auf eine Adresse a_2 , wobei die Datenbelegungen d_1 und d_2 geschrieben werden. Das Beispiel endet mit einem Lesezugriff auf Adresse a_3 .

Zuerst schreiben (Write First, Write Through, Transparent Mode)

Am Datenausgang erscheint der neu eingeschriebene Inhalt (Abb. 4.97). Der Speicher verhält sich ähnlich wie ein (älterer) SRAM mit getrennten Datenein- und Ausgängen (Rückwärtskompatibilität).

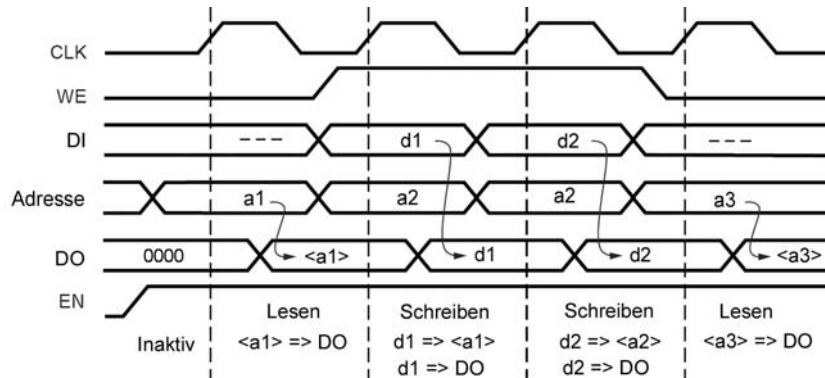


Abb. 4.97 Zuerst schreiben (nach [2.7]). Die Belegung des Dateneingangs erscheint auch am Datenausgang (transparentes Schreiben).

Zuerst lesen (Read-before-Write)

Beim Schreiben erscheint der bisherige Inhalt der Speicherzelle am Datenausgang (Abb. 4.98). So kann mit dem jeweils nächsten Takt die bisherige Belegung in nachgeschaltete Speichermittel übernommen werden. Die adressierte Speicherzelle verhält sich somit praktisch wie ein Flipflop in einem synchronen Schaltwerk⁷⁾. Diese Betriebsweise eignet sich besonders für Schieberegister, FIFOs, Ringpuffer (Abb. 4.100) usw.

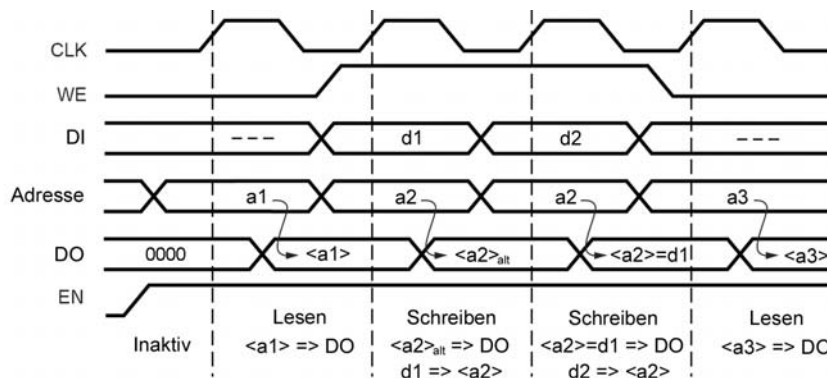


Abb. 4.98 Zuerst lesen (nach [2.7]). Beim Schreiben erscheint der bisherige Inhalt der Speicherzelle am Datenausgang und kann mit dem folgenden Takt in nachgeschaltete Speichermittel übernommen werden.

Nichts ändern (No Change)

Die Belegung des Datenausgangs ändert sich nur bei Lesezugriffen (Abb. 4.99). Ein typisches Einsatzbeispiel ist die Nutzung als Funktionstabelle, deren Inhalt während des Betriebs geändert wird. In dieser Betriebsweise wird das Auslesen nicht von gleichzeitig ablaufenden Änderungen des Speicherinhalts beeinflusst.

7: Deshalb wird diese Betriebsweise in modernen Schaltkreisen bevorzugt. Hinweis: Es muss erst die bisherige Belegung übernommen werden, ehe die neue am Ausgang erscheinen darf (Datenverzögerung).

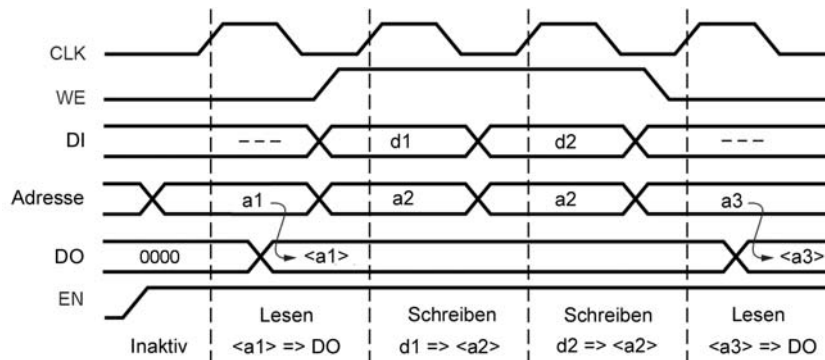


Abb. 4.99 Beim Schreiben nichts ändern (nach [2.7]). Die Belegung des Datenausgangs ändert sich nur bei Lesezugriffen.

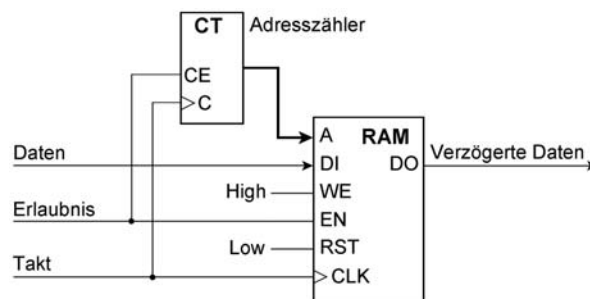


Abb. 4.100 Ringpuffer (Anwendungsbeispiel nach [2.7]). Datenbelegungen sollen um eine bestimmte Anzahl von Taktzyklen verzögert werden. Hierzu wird ein gemäß Abb. 4.98 konfigurierter Block-RAM mit einem Adresszähler verbunden. In jedem Taktzyklus wird der alte Speicherinhalt gelesen und der neue eingeschrieben. Hat der Adresszähler eine Zählweite von n , so wird jeweils nach n Taktzyklen wieder die gleiche Adresse erreicht. Dabei wird die zuvor (also vor n Taktzyklen) gespeicherte Datenbelegung ausgelesen. Somit ergibt sich eine Verzögerung von n Taktzyklen zwischen Dateneingang DI und Datenausgang DO.

Datenbits und Prüfbits

Speicherorganisationen • 9, • 18 und • 36 sind vor allem dazu vorgesehen, 8-Bit-Bytes mit einem zusätzlichen Prüfbits (Paritätsbit) zu ergänzen. Die nutzbare Zugriffsbreite beträgt ein Byte, zwei Bytes oder vier Bytes. Diese zusätzlichen (neunten) Bitpositionen können jedoch auch zur Datenspeicherung verwendet werden.

4.4 Virtuelle Funktionen

Virtuell = scheinbar. Der Begriff wird hier in Entsprechung zum virtuellen Speicher und zur virtuellen Maschine eingeführt⁸⁾. Der Grundgedanke: den Schaltungsaufwand für mehrere gleichartige Funktionen verringern, indem die kombinatorischen Verknüpfungen nur einmal vorgesehen und die Register durch Speicher ersetzt werden (Serialisierung; Zeitmultiplexprinzip). In der Rechnerarchitektur ist dieses Prinzip seit Jahrzehnten⁹⁾ als Hardware-Multitasking, Vertical Multithreading, Hyper-Threading o. ä. bekannt. Dabei handelt es sich darum, mehrere gleichartige Prozessorfunktionen zur Verfügung zu stellen. Die Abb. 4.101 und 4.102 veranschaulichen das Prinzip. Das Beispiel betrifft vier Prozessorfunktionen. Es liegt nahe, einfach vier komplette Prozessorschaltungen (kombinatorische Verknüpfungen + Register) vorzusehen (Abb. 4.101 oben). Der Aufwand ist aber beachtlich. Abb. 4.101 zeigt aber auch eine Alternative: die Register und Flipflops werden durch RAM-Blöcke ersetzt; die kombinatorischen Verknüpfungen sind nur einmal vorhanden. In jedem RAM-Block ist jeder Prozessorfunktion eine Speicherzelle zugeordnet. Alle RAM-Blöcke werden von einem Umschaltzähler adressiert. Im ersten Taktzyklus werden die Speicherzellen der ersten Prozessorfunktion ausgewählt, im zweiten Takt die Speicherzellen der zweiten Prozessorfunktion usw. Die Schaltung verhält sich so wie eine Anordnung von vier Prozessoren, von denen jeder mit einem Viertel der Taktfrequenz arbeitet.

Abb. 4.102 zeigt das Taktschema. Der Takt, mit dem die Schaltung tatsächlich arbeitet, heißt hier der Grundtakt. Jeder Grundzyklus entspricht einem Maschinentakt einer Prozessorfunktion. Alle vier Taktperioden hat jede Prozessorfunktion einen Maschinentakt ausgeführt; die einzelne Prozessorfunktion arbeitet praktisch (von außen gesehen) mit einem Takt, dessen Frequenz ein Viertel der Frequenz des Maschinentaktes beträgt. Ein vollständiger Umlauf des Umschaltzählers ergibt einen Hauptzyklus aus vier Grundzyklen.

Das Prinzip kann auf beliebige RTL-Strukturen angewendet werden. Eine RTL-Struktur kann mehrere gleichartige Funktionen ausführen, wenn man die Latches und Flipflops durch RAM-Anordnungen ersetzt (Abb. 4.103). Das kann immer dann von Vorteil sein, wenn – vereinfacht gesagt – die einzelne Funktion Zeit hat, wenn also im Grunde gar keine extrem hohen Taktfrequenzen erforderlich sind. Dann kann man Logikzellen gleichsam gegen Zeit eintauschen, wobei vielfältige Kompromiss- und Mischlösungen möglich sind. Die Aufwandsersparnis ergibt sich daraus, weil RAM-Blöcke weniger Schaltkreisfläche benötigen als mehrfach angeordnete kombinatorische Netzwerke.

8: In diesem Sinne heißt “virtuell” im Grunde “so tun, als ob...” (... eine nahezu unbegrenzte Arbeitsspeicherkapazität oder eine Vielzahl unabhängiger Prozessoren usw. zur Verfügung stünde). Das beruht stets auf der Nutzung kostengünstiger Speichermittel (Festplatte statt RAM, RAM statt Register) und entsprechenden Umlagerungsverfahren.

9: Das Prinzip wurde bereits 1964 in der Rechanlage CDC 6600 verwirklicht, um mit einer Anordnung ähnlich Abb. 4.101 zehn Ein-Ausgabe-Prozessoren bereitzustellen. Der Grundzyklus (Minor Cycle) war 100 ns lang. Somit hatte jeder E-A-Prozessor einen Maschinentakt (Major Cycle) von 1 μ s.

Abb. 4.104 veranschaulicht ein Ausführungsbeispiel. Von insgesamt 64 Messstellen laufen Impulse ein, die zu zählen sind. Der Zählwert ist mit einem Sollwert zu vergleichen. Die herkömmliche Lösung würde 64 Zähler und 64 Vergleicher erfordern. Man kommt aber mit einer einzigen Zähl- und Vergleichsschaltung aus, wenn man anstelle der Register und Flipflops RAM-Blöcke mit 64 Zellen einsetzt. Konfiguriert man diese als Dual-Port-RAMs, ist auch das Problem der Schnittstelle zu weiterverarbeitenden Einrichtungen gelöst, z. B. zu einem auf dem Schaltkreis untergebrachten Mikrocontroller, der Vergleichswerte einstellen sowie Zählwerte und Vergleichsergebnisse abfragen kann. Im Speicher sind unterzubringen: Zählwert (z. B. 16 Bits), Vergleichswert (z. B. 16 Bits), Vergleichsaussage (1 Bit). Hierzu könnten beispielsweise ein Block-RAM $256 \cdot 16$ für den Vergleichswert (der nicht zurückzuschreiben ist) und ein Block-RAM $256 \cdot 18$ für Zählwert und Vergleichsaussage eingesetzt werden. Es sind 64 Grundtakte erforderlich, um alle Eingangssignale einmal abzufragen (Hauptzyklus).

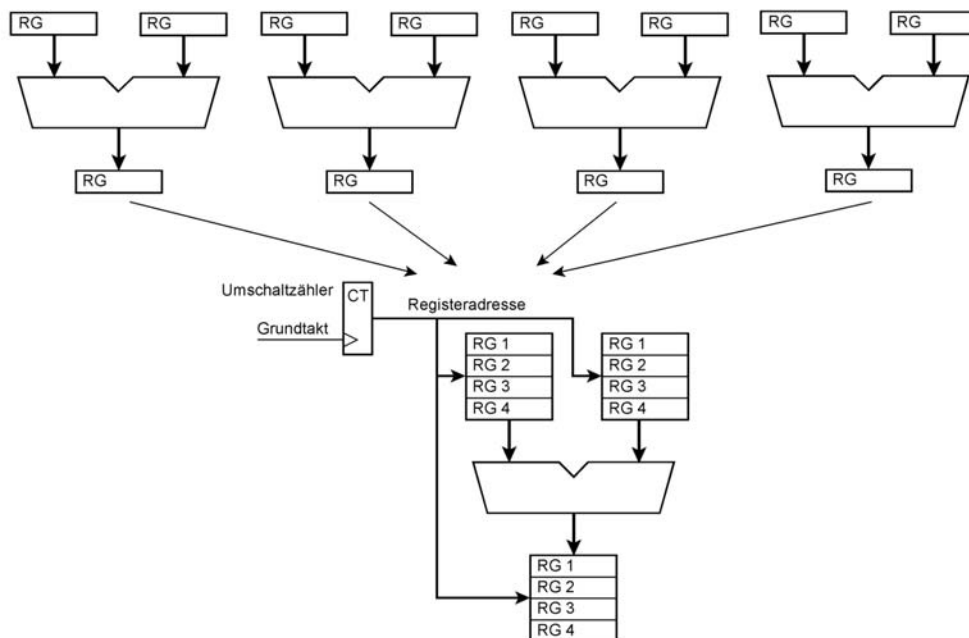


Abb. 4.101 Zum Prinzip der virtuellen Funktionen. Oben: vier gleichartige Schaltungsanordnungen (z. B. Prozessoren). Darunter: diese Anordnung kann die vier Funktionen taktweise zeitversetzt ausführen. Die Register und Flipflops wurden durch RAM-Blöcke ersetzt, die kombinatorischen Verknüpfungen sind nur einmal vorhanden.

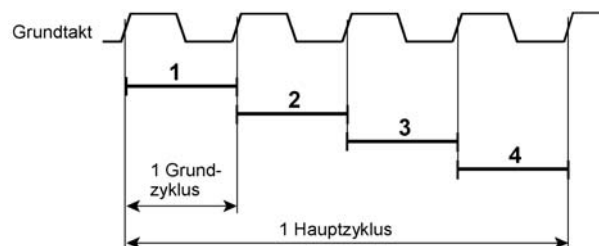


Abb. 4.102 Das grundsätzliche Taktschema der Anordnung von Abb. 4.101.

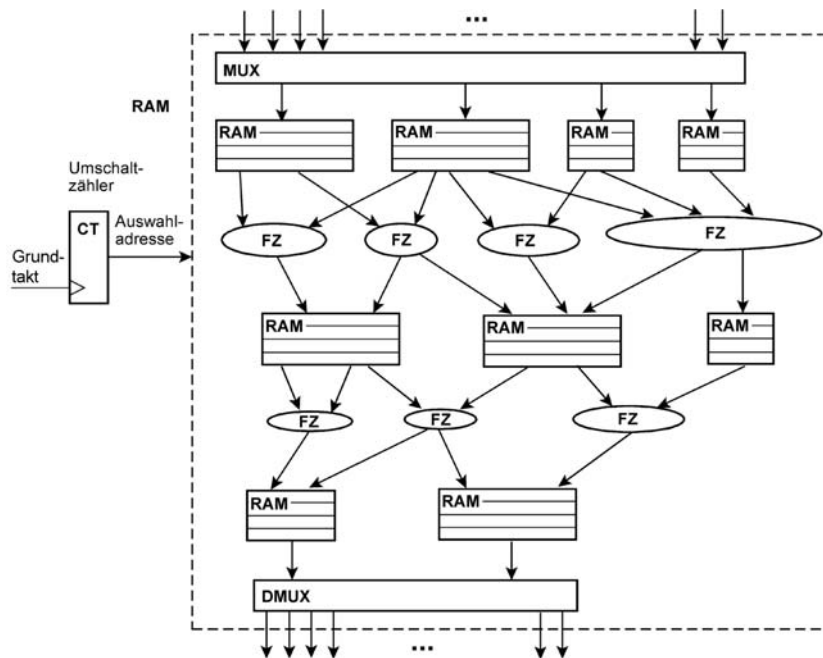


Abb. 4.103 Eine RTL-Struktur für mehrere gleichartige Funktionen. Die Eingangssignale der einzelnen Funktionen müssen beispielsweise über Multiplexer ausgewählt werden. Sinngemäß werden die Ausgangssignale der RTL-Anordnung auf die einzelnen Funktionen verteilt (Demultiplexer).

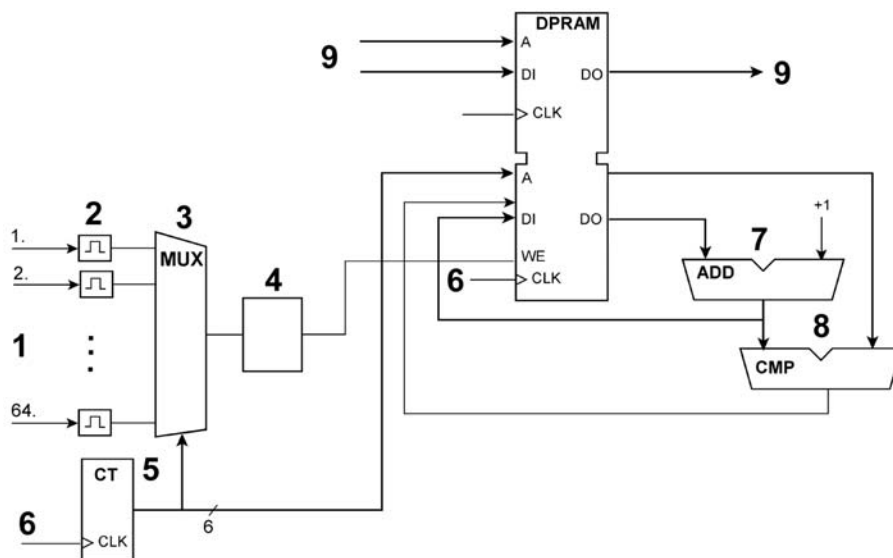


Abb. 4.104 Zähl- und Vergleichseinrichtung für 64 Eingänge (Prinzipschaltung unter Vernachlässigung von Einzelheiten). 1 - 64 Eingangssignale; 2 - 64 • Impulsformung und Synchronisation; 3 - Multiplexer; 4 - Schreibsteuerung (Schreiberlaubnis wird gegeben, wenn eine Zählung auszuführen ist); 5 - Umschaltzähler (modulo 64); 6 - Grundtakt; 7 - Zählnetzwerk (Addieren einer Eins); 8 - Vergleich; 9 - Schnittstelle zu anderen Einrichtungen.