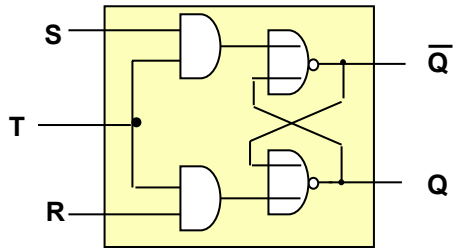
5. Schaltwerke und Speicherelemente

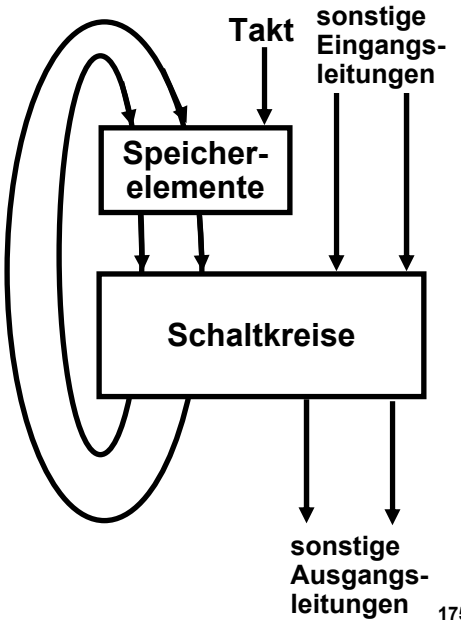


Schaltwerke
Takt, Speicherelemente, Flip-Flops
Verwendung von Flip-Flops
Speicherzellen, Register
Kodierer, Speicher

Schaltwerke vs. Schaltkreise

Schaltkreise bestehen aus
- Eingangsleitungen,
- Schaltnetz und
- Ausgangsleitungen.

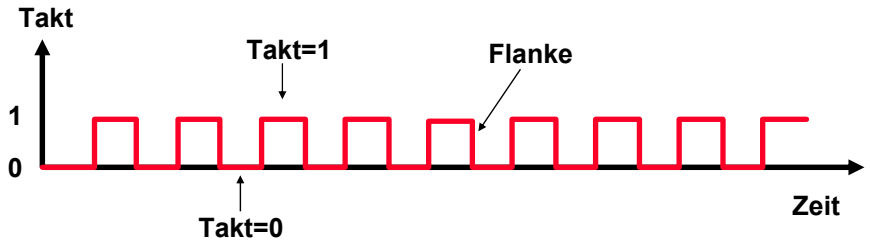
Schaltwerke bestehen aus
- Schaltkreisen,
- Speicherelementen und
- Taktleitungen.



Takt

Schaltwerke befinden sich zu gegebener Zeit in einem Zustand Q. Ein solcher Zustand ist beispielsweise durch die Werte der Ausgangsleitungen zu diesem Zeitpunkt charakterisiert.

Zustände ändern sich nur zu bestimmten diskreten Zeitpunkten, bei Eintreffen eines Taktes, das heißt, wenn eine 1 auf der Taktleitung anliegt.



Speicherelemente

**Binäre Speicherelemente sind kleinste logische Bausteine zur Aufbewahrung von Information. Es wird hier nur einer der Zustände "0" oder "1" gespeichert.
Sie sind als Schaltwerke realisiert.**

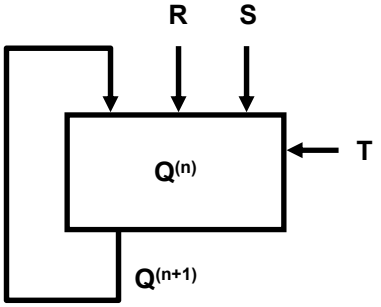
Dabei ist ein Zustand $Q^{(n)}$ durch den gespeicherten Wert und durch die Werte auf den Eingangsleitungen im n-ten Taktzyklus bestimmt.

Der gespeicherte Wert steht über eine Ausgangsleitung zur Verfügung, meist gibt es zusätzlich eine Leitung an der $Q^{(n)\prime}$ anliegt.

RS-Flip-Flop

Ein RS-Flip-Flop besitzt zwei besondere Eingänge zum
- **Setzen (auf "1"; Set) und**
- **Rücksetzen (auf "0"; Reset)**

des Inhalts.



Sei T der Takt mit:
T=0: Zustand ändert sich nicht,
T=1: Zustand kann sich ändern.
Dann wird das RS-Flip-Flop durch folgende Tabelle beschrieben:

S	R	$Q^{(n+1)}$
0	0	$Q^{(n)}$
0	1	0
1	0	1
1	1	nicht def.

RS-Flip-Flop (Forts.)

Unter Einbeziehung des aktuellen Zustandes $Q^{(n)}$ ergibt sich folgende Tabelle für die Schaltfunktion $Q^{(n+1)}$, wenn ein Takt anliegt (T=1). Sonst (T=0) ändert sich nichts.

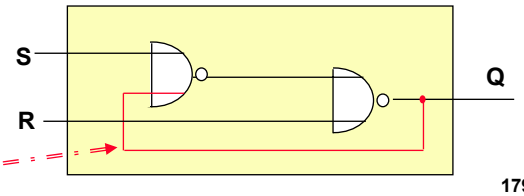
S	R	$Q^{(n)}$	$Q^{(n+1)}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	undef.
1	1	1	undef.

➡ **Schaltfunktion für $Q^{(n+1)}$:**

$$Q^{(n+1)} = R'(S + Q^{(n)})$$
$$= [\, R + (S + Q^{(n)})' \,]'$$

(NOR-Darstellung)

➡ **Schaltung für $Q^{(n+1)}$:**

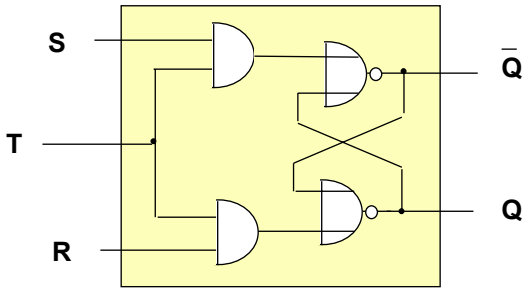


Getaktetes RS-NOR-Flip-Flop

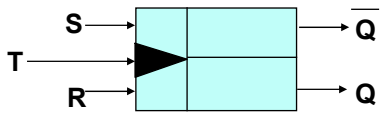
Schaltfunktion:

$$Q^{(n+1)}$$
$$= T' Q^{(n)} + T R'(S + Q^{(n)})$$
$$= (T' + TR')Q^{(n)} + T R' S + T T' S$$
$$= (TR)' Q^{(n)} + (TR)' TS$$
$$= (TR)' (TS + Q^{(n)})$$
$$= [\, TR + (TS + Q^{(n)})' \,]'$$

Schaltung:



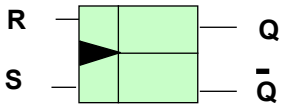
Schaltsymbol:



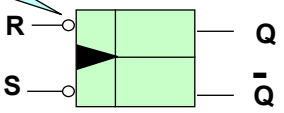
RS-NAND-Flip-Flops

Ersetzt man alle NOR-Glieder eines RS-NOR-Flip-Flops durch NAND-Glieder, so erhält man den RS-NAND-Flip-Flop. Dessen Verhalten ist dual zu dem des RS-NOR-Flip-Flops.

Die Kreise an den Eingängen deuten an, dass dieser Flip-Flop aktiv ist, wenn r = 0 oder s = 0 gilt.



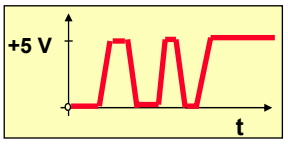
RS-NOR-Flip-Flop



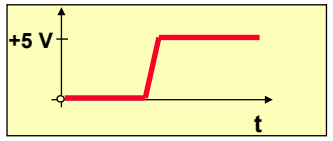
RS-NAND-Flip-Flop

Verwendung von Flip-Flops

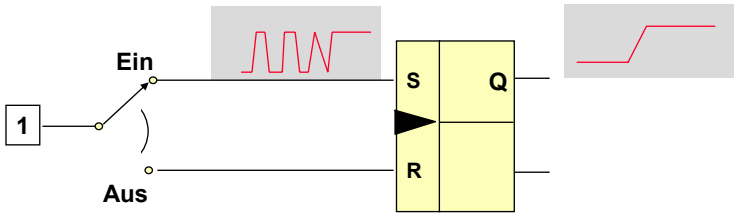
Eine elementare Verwendung eines Flip-Flops ergibt sich für das Entprellen eines Schalters, z.B. einer Taste der Tastatur.



Prellender Schalter



Idealer Schalter

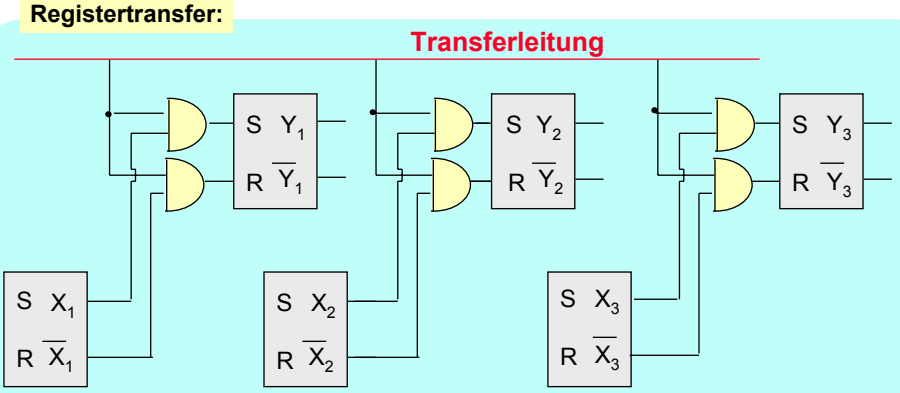


Entprellung mit Flip-Flop

Datentransfer

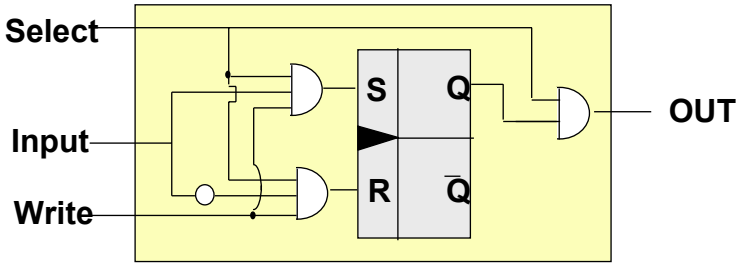
Ein Register ist eine Speicherzelle für ein Wort. Es besteht im wesentlichen aus mehreren 1-Bit Speicherzellen. Zur Übertragung des Inhaltes von Register $X=X_1X_2 \ldots X_n$ zu Register $Y=Y_1Y_2 \ldots Y_n$ kann etwa folgende Schaltung verwendet werden:

Registertransfer:



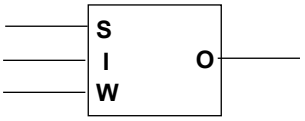
Eine Speicherzelle

Eine Speicherzelle besteht im wesentlichen aus einem RS-FlipFlop. Für das Schreiben, Lesen und Ansteuern der Speicherzelle werden weitere Zuleitungen eingesetzt.



Nur bei Select = 1 steht der gespeicherte Wert bei OUT zur Verfügung.
Nur bei Select = Write = 1 kann der Inputwert gespeichert werden!

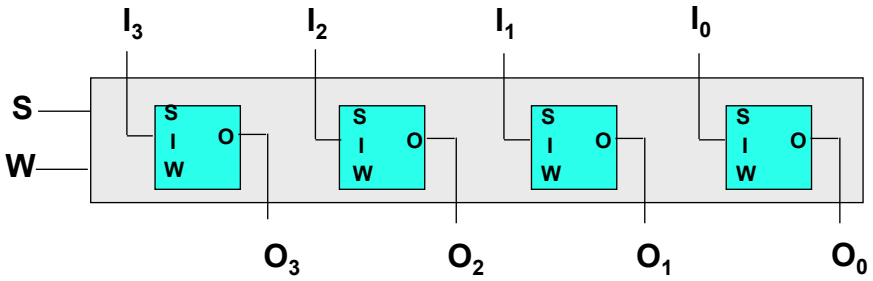
Vereinfachtes Schaltbild obiger Speicherzelle:



Register

Ein **Register** ist eine Gruppe von Speicherzellen. Die Anzahl der Speicherzellen in einem Register ist meist gleich der Wortgröße.

Da man nie einzelne Zellen eines Registers anspricht, kann man die Set- und Write-Ausgänge miteinander verbinden.



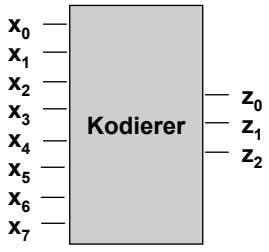
Kodierer / Dekodierer

Ein **Kodierer** hat 2^n Eingänge und n Ausgänge.

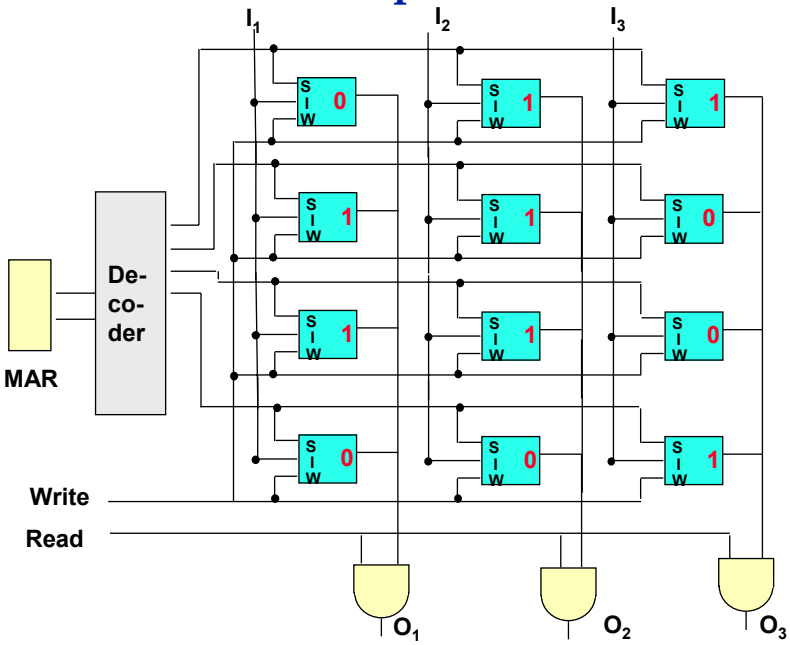
Für den Fall, dass an dem k-ten Eingang eine 1 und an allen anderen Eingängen eine 0 liegt, soll an den Ausgängen die Zahl k binär dargestellt werden. Für alle anderen Inputs ist das Ergebnis unspezifiziert.

Vertauscht man Eingang und Ausgang, erhält man einen **Dekodierer**.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	z_2	z_1	z_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



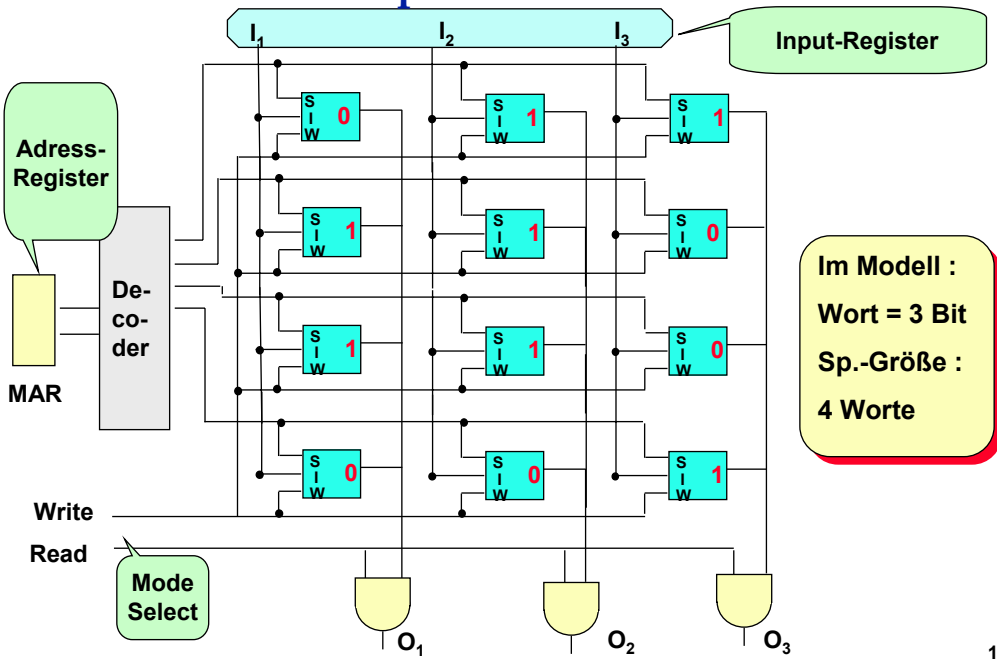
Adressierbarer Speicher



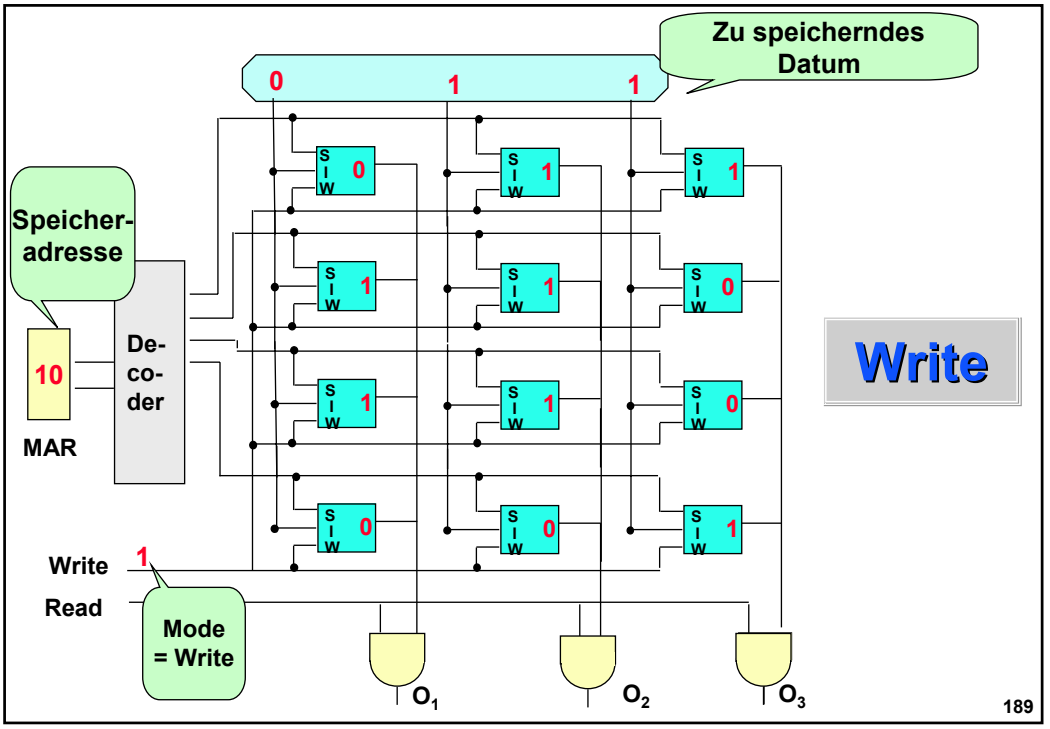
Im Modell :
Wort = 3 Bit
Sp.-Größe :
4 Worte

MAR = **m**emory
address **r**egister

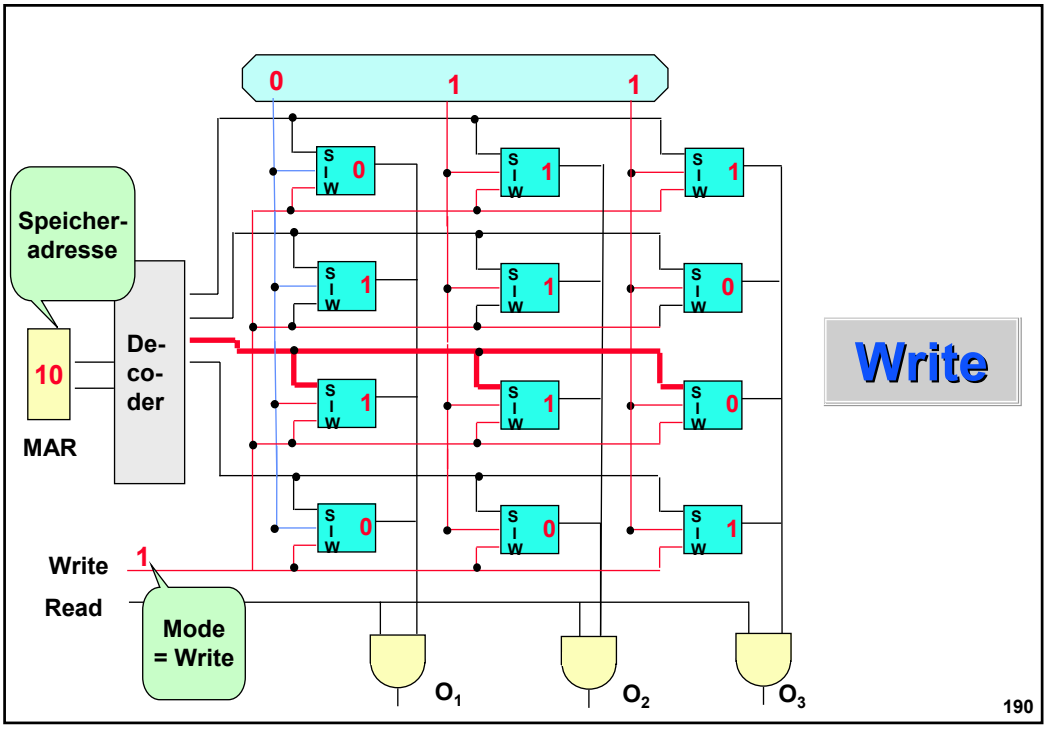
Adressierbarer Speicher



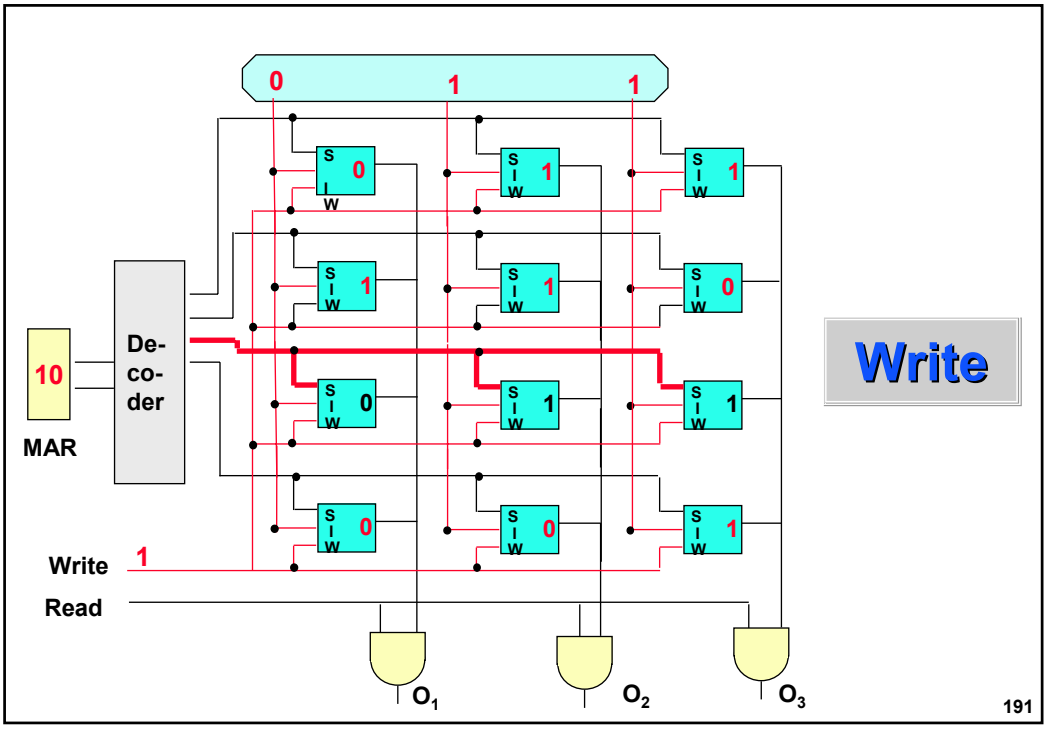
Im Modell :
Wort = 3 Bit
Sp.-Größe :
4 Worte

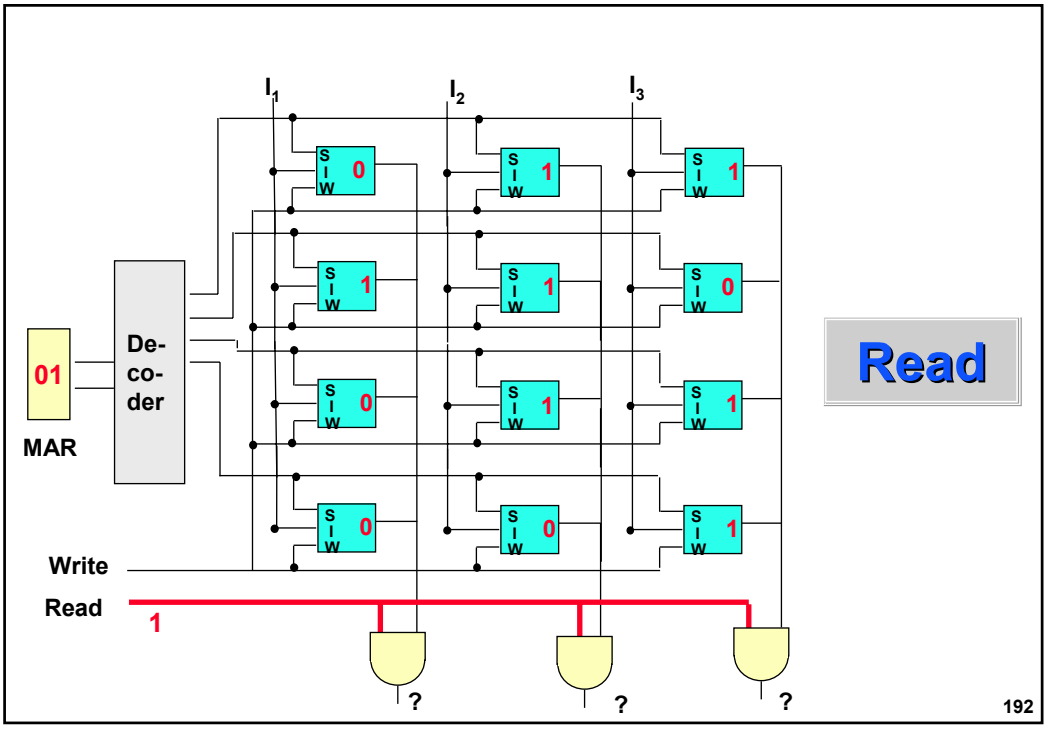

Zu speicherndes Datum
0
1
1
Speicher-adresse
10
MAR
De-co-der
Write
Read
1
Mode = Write
O1
O2
O3
Write

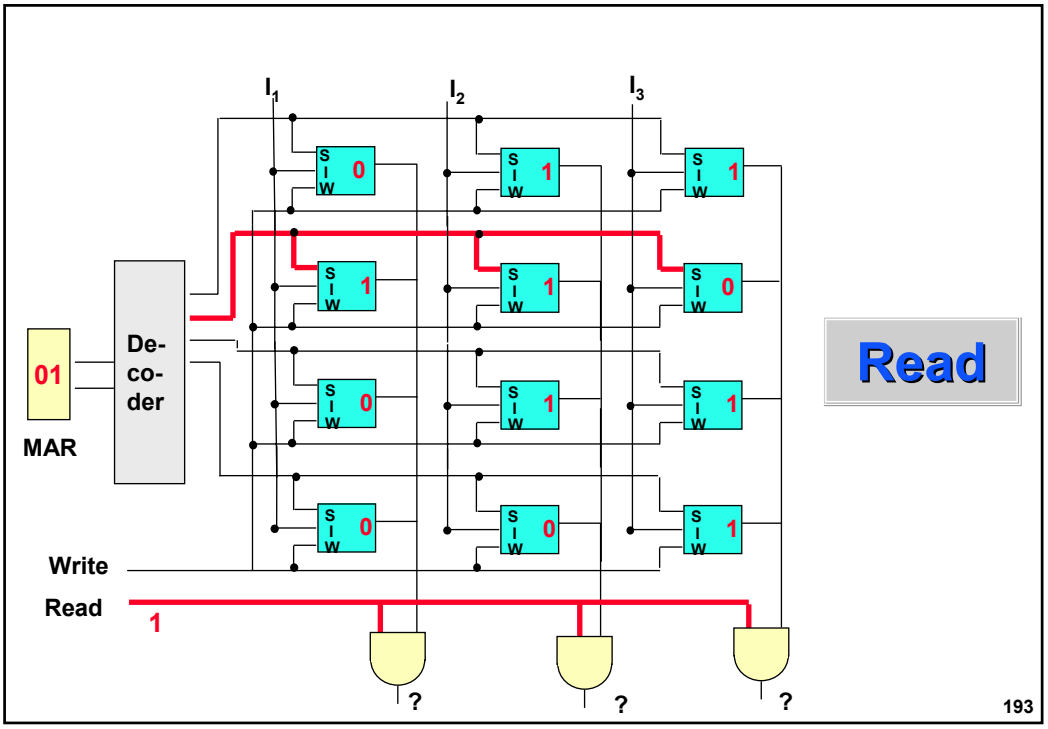
189

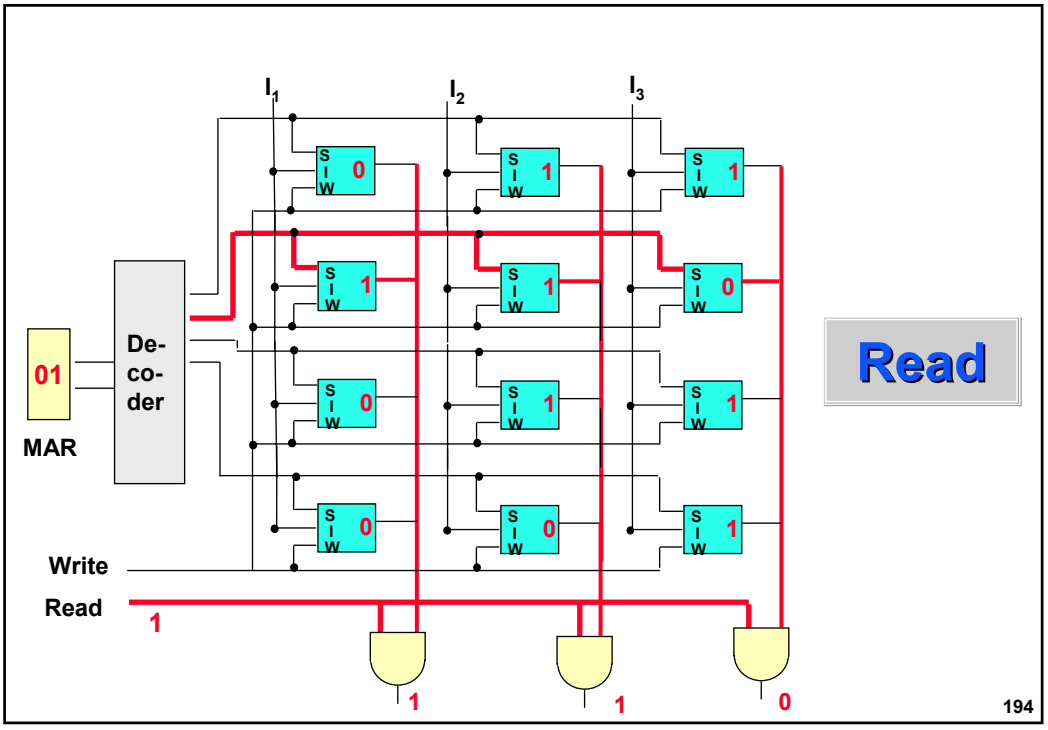

0
1
1
Speicher-adresse
10
MAR
De-co-der
Write
Read
1
Mode = Write
O1
O2
O3
Write

190


0
1
1
10
De-
co-
der
MAR
Write
1
Read
O1
O2
O3
Write

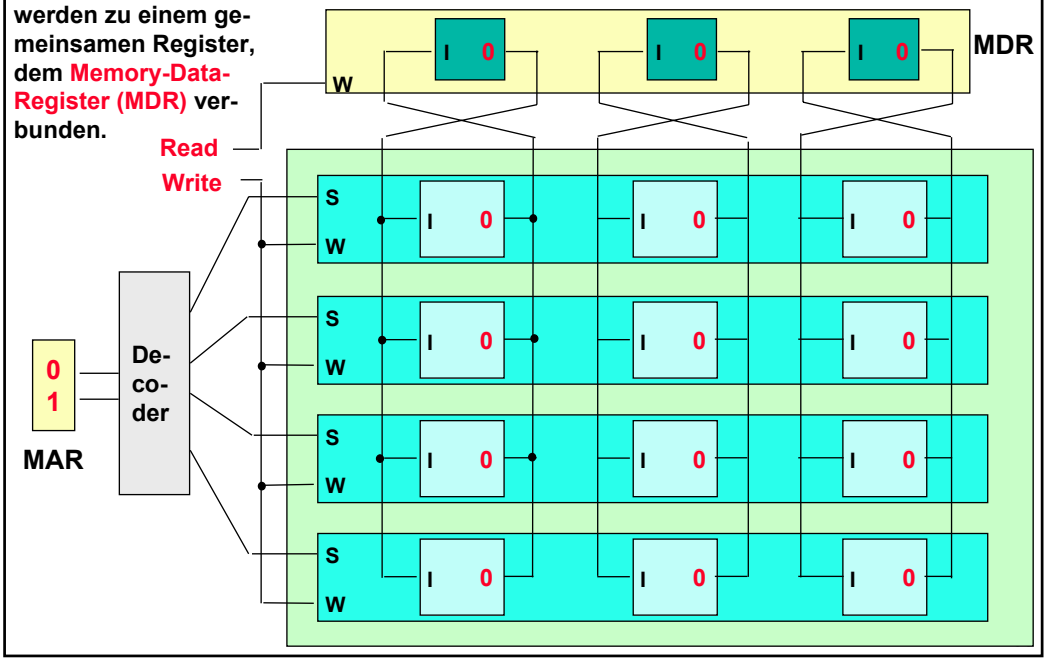

I1
I2
I3
01
De-
co-
der
MAR
Write
Read
1
?
?
?
Read


I1
I2
I3
SIW 0
SIW 1
SIW 1
SIW 1
SIW 0
SIW 0
SIW 1
SIW 1
SIW 0
SIW 0
SIW 1
01
De-
co-
der
MAR
Write
Read
1
?
?
?
Read
193


I1
I2
I3
SIW 0
SIW 1
SIW 1
SIW 1
SIW 0
SIW 0
SIW 1
SIW 1
SIW 0
SIW 0
SIW 1
01
De-
co-
der
MAR
Write
Read
1
1
1
0
Read
194

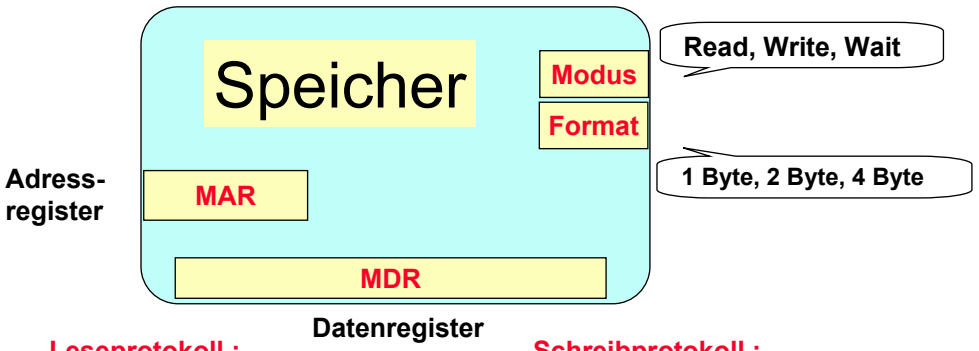
Speicher mit MDR

Eingabe- und Ausgaberegister des Speichers werden zu einem gemeinsamen Register, dem Memory-Data-Register (MDR) verbunden.



Externe Sicht des Speichers

Nach außen ist der Speicher durch die Register MAR, MDR, Mode und Format ansprechbar.



Leseprotokoll :

Stelle Adresse in MAR ein
Setze gewünschtes Format
Setze Modus auf "Read"
Empfange Datum in MDR

Schreibprotokoll :

Bringe Datum nach MDR
Stelle Adresse in MAR ein
Setze gewünschtes Format
Setze Modus auf Write