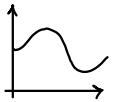
SIGNALE

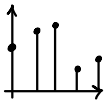
ANALOG

Wertekontinuierlich & Zeitkontinuierlich



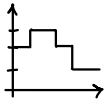
ZEITDISKRET

Exakte Werte nur zu bestimmten Zeiten



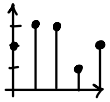
WERTEDISKRET

Zu jeder Zeit nur bestimmte Werte



DIGITAL

Nur bestimmte Werte zu bestimmten Zeiten

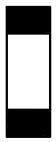


BITS

Ein Bit kann 2 Zustände annehmen
n Bits können 2^n Zustände beschreiben

LOGISCHE ZUSTÄNDE

(x) nicht definiert



-> HIGH (0,7V - 0,9V) Speisespannung

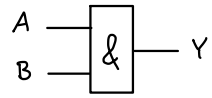
-> LOW (0V - 0,15V) Masse

LOGISCHE VERKNÜPFUNGEN

UND

Wenn beide Eingänge 1, dann Ausgang auch

$Y = A \wedge B = A \cdot B$

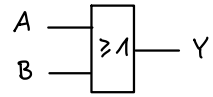


A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

ODER

Wenn einer der Eingänge 1, dann Ausgang auch

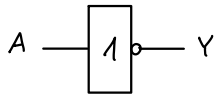
$Y = A \vee B = A + B$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

INVERTER

Der Eingang wird invertiert

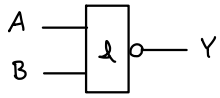


A	Y
0	1
1	0

NAND

UND-Funktion wird invertiert

$Y = \overline{A \wedge B} = \overline{A \cdot B}$

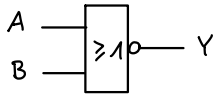


A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR

ODER-Funktion wird invertiert

$Y = \overline{A \vee B} = \overline{A + B}$

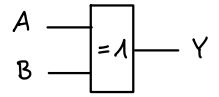


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

XOR

Ausgang 1, wenn genau einer der Eingänge 1

$Y = A \oplus B$

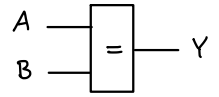


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

XOR-Funktion wird invertiert

$Y = \overline{A \oplus B}$



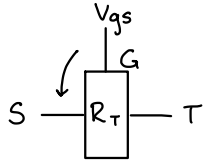
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

MOS-TRANSISTOREN

TRANSISTOR

Steuerbarer Widerstand

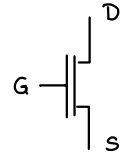
Elektronen/Löcher fließen von S nach D



$|V_{gs}| < |V_{th}|$: kein Strom

$|V_{gs}| > |V_{th}|$: Strom fließt

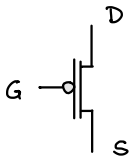
NMOS-TRANSISTOREN



- Strom fließt in entgegengesetzte Richtung wie Ladungsträger

→ Bei Pull-Down Pfad

PMOS



- Strom in gleiche Richtung wie Ladungsträger

→ Bei Pull-Up Pfad

ZUSTÄNDE BESTIMMEN

Zu (leitet), wenn G ≠ S → NMOS: G = 1
→ PMOS: G = 0

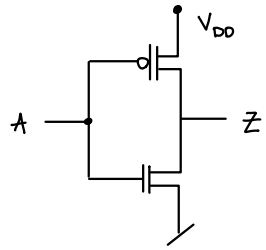
offen (leitet nicht), wenn G = S → NMOS: G = 0
→ PMOS: G = 1

NN (Unbestimmt), wenn vorheriger gesperrt

CMOS-GATTER

NICHT

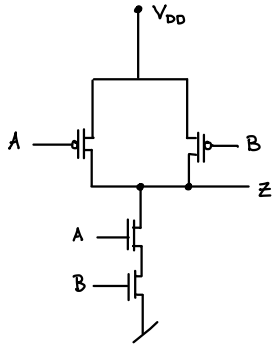
Genau 1 NMOS & 1 CMOS



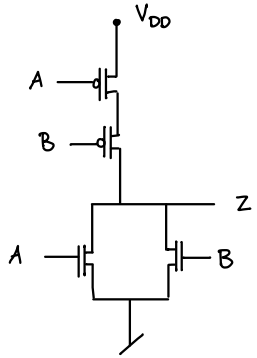
NAND

PMOS parallel

NMOS in Serie



NOR



PMOS in Serie

NMOS parallel

PULL-UP ⟷ PULL-DOWN

I Gleichung bestimmen (bei P-d invert.)

II Gleichung umkehren (+ ⟷ · / A ⟷ $\bar{A}$)

III Neuen Pfad zeichnen ( = UND /  = ODER)

ZEITVERZÖGERUNG

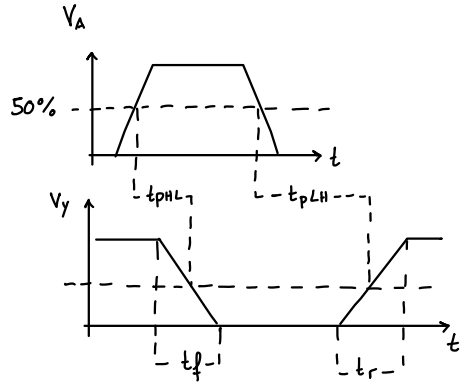
$t_d = (t_{pHL} + t_{pLH}) / 2$

t_{pHL} / t_{pLH}: Gemessen bei 50%

→ wie lange dauert es, bis das Signal vom Eingang beim Ausgang ist

t_r / t_f: Gemessen bei 10% & 90%

→ wie lange dauert der Aufstieg/Abstieg?



BOOL'SCHE ALGEBRA

KOMMUTATIVITÄT: $A \wedge B = B \wedge A \;/\; A \vee B = B \vee A$

ASSOZIATIVITÄT $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
$(A \vee B) \vee C = A \vee (B \vee C)$

DISTRIBUTIVITÄT: $(A \wedge B) \vee (A \wedge C) = A \wedge (B \vee C)$
$(A \vee B) \wedge (A \vee C) = A \vee (B \wedge C)$

NICHT-THEOREM: $\bar{\bar{A}} = A$

NULL-THEOREM: $A \wedge 0 = 0 \;/\; A \vee 0 = A$

EINS-THEOREM: $A \wedge 1 = A \;/\; A \vee 1 = 1$

IDEMPOTENZ: $A \vee A = A \;/\; A \wedge A = A$

KOMPLEMENT: $A \wedge \bar{A} = 0 \;/\; A \vee \bar{A} = 1$

ADSORPTION: $A \vee (\bar{A} \wedge B) = A \vee B$
$A \wedge (\bar{A} \vee B) = A \wedge B$

ABSORPTION: $A \vee (A \wedge B) = A$
$A \wedge (A \vee B) = 1$

NACHBARSCHAFTSGESETZE: $(A \wedge B) \vee (\bar{A} \wedge B) = B$
$(A \vee B) \wedge (\bar{A} \vee B) = B$

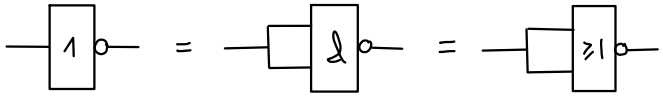
BINDUNGSREGELN

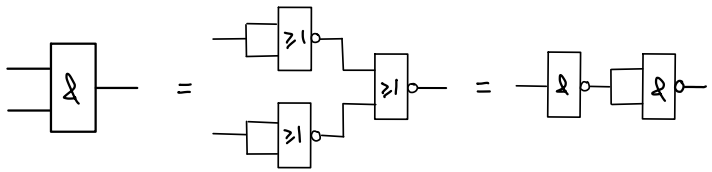
I Klammern
II Nicht
III {UND, ODER, NAND, NOR} vor X

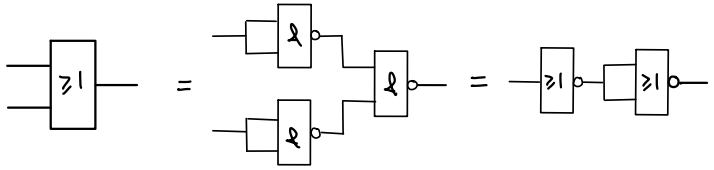
DE MORGANSCHE GESETZE

$\overline{A \wedge B \wedge C \wedge \ldots} = \bar{A} \vee \bar{B} \vee \bar{C} \vee \ldots$

$\overline{A \vee B \vee C \vee \ldots} = \bar{A} \wedge \bar{B} \wedge \bar{C} \wedge \ldots$







NORMALFORMEN

MINTERM

UND-Verknüpfung mit allen 0 invertiert
wird gebildet wenn Ausg. = 1

MAXTERM

ODER-Verknüpfung mit allen 1 invertiert
wird gebildet, wenn Ausg. = 0

DISJUNKTIVE NORMALFORM

ODER-Verknüpfung aller Minterme

KONJUNKTIVE NORMALFORM

UND-Verknüpfung aller Maxterme

KANONISCHE FORM

In jedem Min-/Maxterm kommt jede Variable ein Mal vor

KARNAUGH DIAGRAMME

VEREINFACHUNG DNF

1er - Päckchen

Für jedes Päckchen Minterm mit gleich bleibenden Variablen

VEREINFACHUNG KNF

0er - Päckchen

Für jedes Päckchen Maxterm mit gleich bleibenden Variablen

Aber: $V \leftrightarrow \wedge$, $\bar{A} \leftrightarrow A$ } austauschen

DON'T CARE ZUSTÄNDE

Nicht gebrauchte Kombinationen.

Im Karnaugh-Diagramm = X
↳ können als 0 & 1 benutzt werden

HAZARDS

Wenn Päckchen sich berühren aber nicht schneiden

Lösung: Extra Päckchen (& Gleichung) um den Berührpunkt

ZAHLENSYSTEME

DEZIMAL Basis: 10
Notation: $X_{(10)}$

HEXADEZIMAL Basis: 16
Notation: $X_{(16)}$ / 0x...

OKTAL Basis: 8
Notation: $X_{(8)}$ / 0o...

BINÄR Basis: 2
Notation: $X_{(2)}$ / 0b

UMWANDLUNG

DEZ >1 → ANDERE:
i) Division durch Basis → Rest von LSB aus eintragen
ii) Quotient wieder Teilen
iii) Ende wenn Quotient = 0

DEZ <1 → ANDERE:
i) Mit Basis multiplizieren → Abgerundetes Produkt von vorne eintragen
ii) Nachkommastellen mult.
iii) Ende wenn Produkt = ganz

HEX ↔ BINAR: 4er-Päckchen (HEX) = 1 Stelle (BIN)

OCT ↔ BINÄR: 3er-Päckchen (OCT) = 1 Stelle (BIN)

DUALZAHLEN

Um negative Binärzahlen darzustellen

POSITIV: Normale Binärzahl mit 0 als MSB

NEGATIV: Positives Äquivalent invertiert und +1 an LSB

ADDITION: Bitweise von rechts nach links
Max +1 Bit vorne hinzufügen

SUBTRAKTION: Zu subtrahierende Zahl negieren und dann addieren

MULTIPLIKATION: $b_3 b_2 b_1 b_0 \times a_3 a_2 a_1 a_0$

$$+ \begin{cases} b_0(a_3 a_2 a_1 a_0) \\ b_1(a_3\ a_2\ a_1\ a_0)\,0 \\ b_2(a_3\ a_2\ a_1\ a_0)\,0\,0 \\ b_3(a_3\ a_2\ a_1\ a_0)\,0\,0\,0 \end{cases}$$

CODES

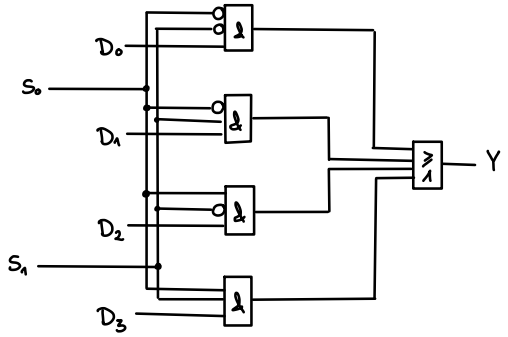
BCD: Für jede Ziffer in DEZ. eine eigene Tetrade

GRAY: Es verändert sich immer nur eine Stelle

RECHENSCHALTUNGEN

MULTIPLEXER

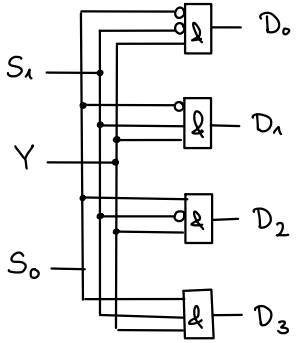
Schaltet mehrere Eingänge auf einen Ausgang



n Steuereingänge n → 2^n Datenpfade

DEMULTIPLEXER

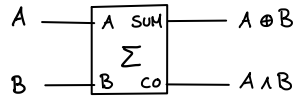
Schaltet einen Eingang auf mehrere Ausgänge



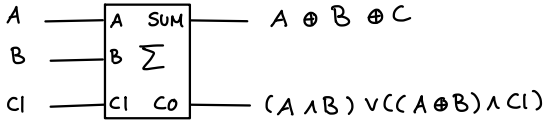
n Steuereingänge → 2^n Datenpfade

ADDIERER

HALBADDIERER: Addiert 2 Bits



VOLLADDIERER: Addiert 3 Bits (= 2 HA in Serie)



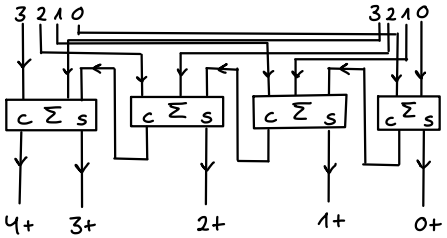
MEHRBIT-ADDIERER

Um Dualzahlen mit mehreren Bits zu addieren

NORMALFORM: Für jedes Bit eigene Gleichung

⊕ Schnell, da alle Bits auf einmal
⊖ Großen Schaltungsaufwand

RIPPLE-CARRY:

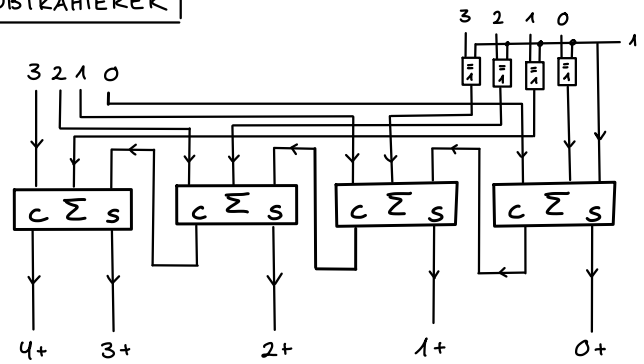


⊕ Geringer Schaltaufwand
⊖ Langsam → Daten "rieseln" durch

CARRY-LOOKADEAD: Kombiniert Vorteile von Normalform & Ripple Carry

→ Ripple-Carry aber Überträge separat

SUBTRAHIERER

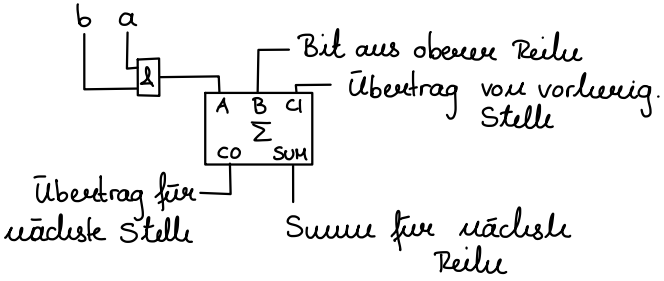


wie Ripple-Carry

Invertieren durch XOR am Eingang
+1 durch Volladdierer am Eingang

MULTIPLIKATOR

über Basiszellen

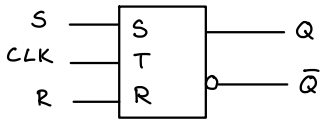


LATCHES

BASICS

Taktzustandgesteuert

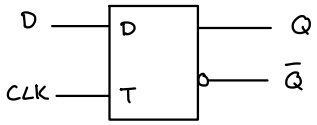
SR-LATCH



$$Q_{n+1} = \begin{cases} 1, & S=1, R=0 \\ 0, & R=1, S=0 \\ Q_n, & S=R=0 \end{cases}$$

Taktzustandgesteuert → verändert sich nur, wenn $T=1$

D-LATCH



$$Q_{n+1} = \begin{cases} D, & \text{wenn } T=1 \\ Q_n, & \text{wenn } T=0 \end{cases}$$

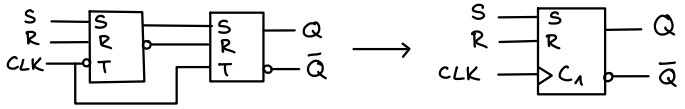
VORTEIL ZU SR: Kein unerlässiger Zustand $S=R=1$ möglich

FLIPFLOPS

BASICS

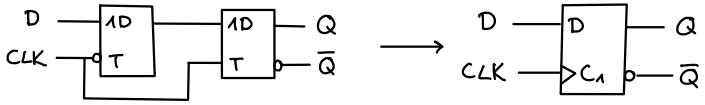
Taktflankengesteuert

SR-FLIPFLOP



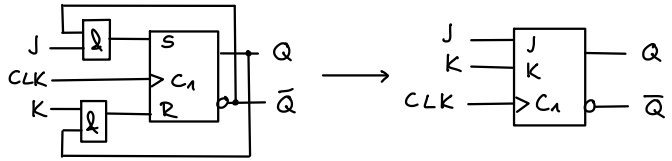
$$Q_{n+1} = (S \vee (\bar{R} \wedge Q_n))_n \quad , \quad (R \overset{!}{=} S \overset{!}{=} 0 \;\&\; CLK \overset{!}{=} 0 \to 1)$$

D-FLIPFLOP



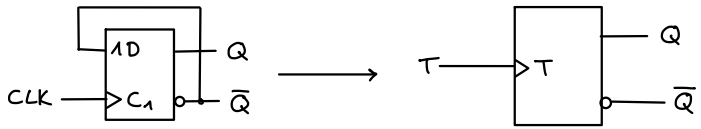
$$Q_{n+1} = D_n \quad , \quad (CLK \overset{!}{=} 0 \to 1)$$

JK-FLIPFLOP



$$Q_{n+1} = (J \wedge \bar{Q}_{1n}) \vee (\bar{K} \wedge Q_{1n}) \quad , \quad (CLK \overset{!}{=} 0 \to 1)$$

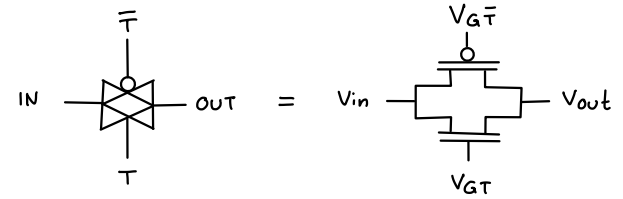
T-FLIPFLOP



$$Q_{n+1} = \bar{Q}_n \quad , \quad (CLK \overset{!}{=} 0 \to 1)$$

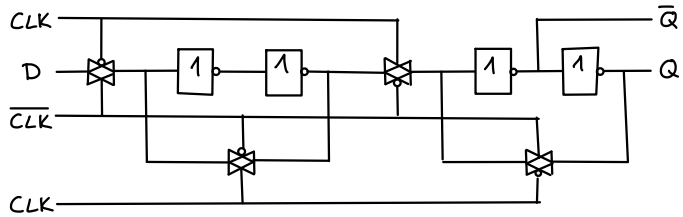
auch möglich mit zusätzl. Taktsteuerung

TRANSMISSION GATES



→ Eingang am Ausgang sichtbar, wenn $T=1$

D-FLIPFLOP IN CMOS



DYNAMIK

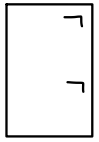
t_{pd}: Verzögerung zw. aktiver Taktflanke & Wirkung

t_s: Wie lange Signal vor Taktflanke konstant

t_h: Wie lange Signal nach Taktflanke konstant

MAX TAKTFREQUENZ: $1/(t_{pd,ff1} + t_{pd,s} + t_{su,ff2})$

MASTER-SLAVE FLIPFLOP



- liest ein bei $CLK = 0 \to 1$
- Gibt aus bei $CLK = 1 \to 0$

AUTOMATEN

CHARAKTERISIERUNG

EINGABEALPHABET: $X = \{x_1, x_2, \dots\}$

AUSGABEALPHABET: $Y = \{y_1, y_2, \dots\}$

INTERNE ZUSTÄNDE: $Z = \{z_1, z_2, \dots\}$

ANFANGSZUSTAND: $z_0 \in Z$

ÜBERFÜHRUNGSFUNKTION: $f_{c1}(x_n, z_n) \rightarrow z_{n+1}$

AUSGABEFUNKTION: $f_{c2}(x_n, z_n) = y_n$

ENDLICHE AUTOMATEN

Mögliche X, Y, Z

sind endlich & vorprogrammiert

SYNCHRONE AUTOMATEN

Alle Flipflops haben das gleiche Clock

SPEICHERELEMENTE

D-Flipflops

→ Weniger störungsanfällig als Latches

→ Günstiger als JK- oder MS-Flipflop

AUTOMATENTYPEN

Ineinander Umwandelbar

MEALY: Y hängt von X & Z ab.

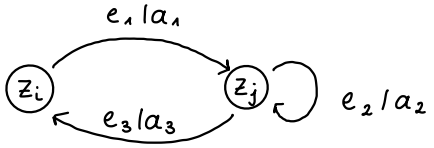
MOORE: Y hängt nur von Z ab

→ Sonderfall vom Mealy-Automat

MEDWEDJEW: Moore-Automat mit $Y = Z$

ZUSTANDSDIAGRAMME

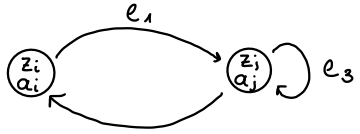
MEALY:



Knoten = interne Zustände

Kanten = Zustandsübergänge mit Eing. & Ausg.

MOORE:



Knoten: Interne Zustände mit Ausgängen

Kanten: Zustandsübergänge mit Eingängen

ZUSTANDSFOLGETABELLE

→ 2^{x+z} Zeilen

X_n	Z_n	Z_{n+1}	Y_n
⋮	⋮	⋮	⋮

→ X+2Z+Y Spalten

AUTOMATENSYNTHESE

i) Zustände bestimmen

n Flipflops → 2^n Zustände

ii) Eingänge / Ausgänge / Zustände kodieren

n Bits → 2^n E/A/Z

iii) Zustandsdiagramm zeichnen

Zustandsfolgetabelle aufstellen

KV-Diagramm & Gleichungen bestimmen

iv) Schaltplan zeichnen

DYNAMISCHES VERHALTEN

MEALY: Änderung Eingang → sofort Änderung am Ausgang

MOORE: Änderung Eingang → erst bei nächster Taktflanke Änderung am Ausgang

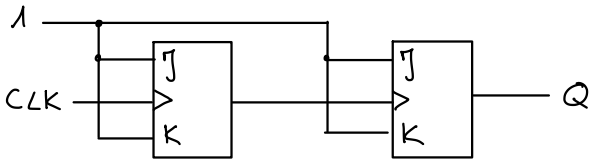
SYNCHRONISIERUNG EINGÄNGE

Fehlervermeidung

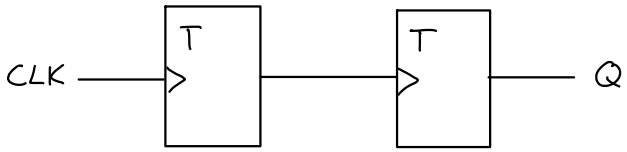
D-FF mit Automaten CLK an Eingänge

FREQUENZTEILER

OHNE ZÄHLER



=



$Q = \frac{CLK}{2^n}$ → Q = Ausgangsfrequenz

→ CLK = Eingangsfrequenz

→ n = Anzahl Flipflops

MIT ZÄHLER

$Q = \frac{CLK}{2(n-k+1)}$

→ Q = Ausgangsfrequenz

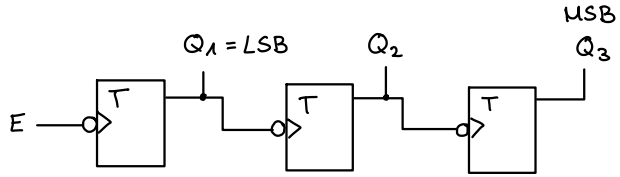
→ CLK = Eingangsfrequenz

→ k = Zahl, bei der angefangen wird

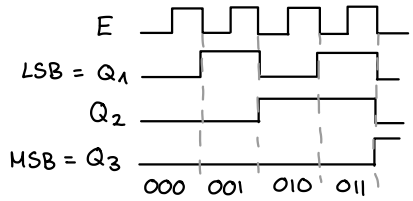
→ n = Zahl, bis zu der gezählt wird

ZÄHLER

ASYNCHRONZÄHLER



Zählt bis 2^n (n = # Flipflops)



ZEITVERZÖGERUNG

Summieren sich

$f_{max} = 1/\Sigma t_{pd,FF_i}$ (damit keine Fehler)

SYNCHRONZÄHLER

≙ Medwedjew Automat

Alle Flipflops mit dem gleichen CLK verbunden

z_{n+1} durch kombinatorische Schaltung

SYNTHESE

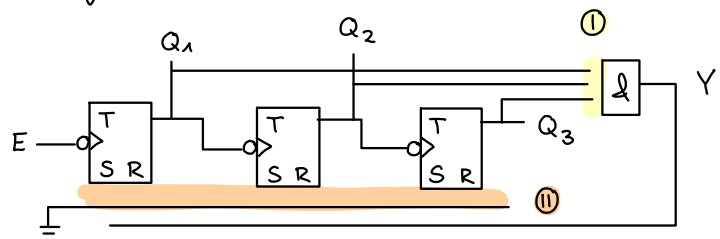
normal wie bei Automaten

JK-FLIPFLOP: Päckchen für Q & $\bar{Q}$

$(Q_{n+1} = (J_i \wedge \bar{Q}_i) \vee (\bar{K}_i \wedge Q_i))$

MODULO-N-ZÄHLER

Anfangs- & Endzustand einstellbar



① Rücksetzzustand einstellen mit INV

(= letzte gewünschte Zahl + 1)

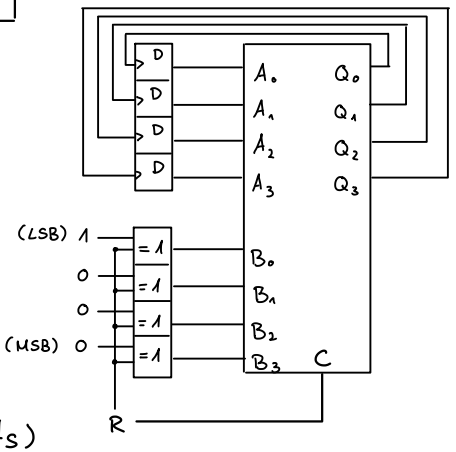
→ wenn erreicht Y = 1

⑪ Anfangszustand einstellen mit S & R

→ 1 gewünscht: S = Y = 1

→ 0 gewünscht: R = Y = 1

RÜCKWÄRTS-ZÄHLER



R = 0: + (Vorwärts)

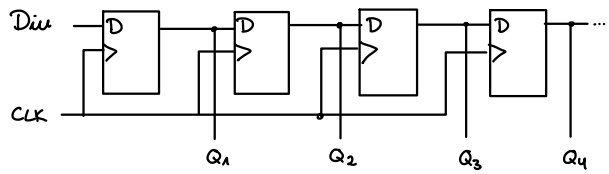
R = 1: − (Rückwärts)

SPEICHER

SCHIEBEREGISTER

Daten werden seriell eingelesen & sind dann parallel abrufbar

#Flipflops = # gespeicherte Bits



↯ Teuer und Platzintensiv

SPEICHERMEDIEN

MAGNETBAND: Speicherkapazität > 100TByte
Zugriffszeit ~ 3min

PLATTENLAUFWERK: Speicherkapazität ~ TB
Zugriffszeit ~ ms

FLASH: Speicherkapazität ~ 2TB
Zugriffszeit ~ µs Bereich

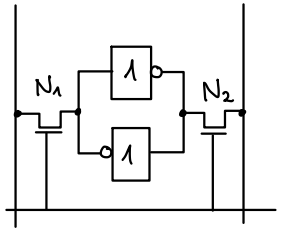
DRAM: Speicherkapazität = 8GB
Zugriffszeit = 5ns

FLÜCHTIGE SPEICHER

Speichert nur, wenn eine Spannung anliegt.

SRAM

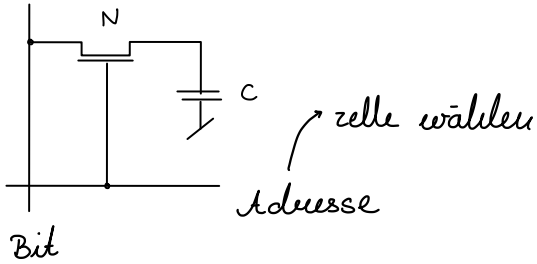
Static random access memory



Bit	$\overline{Bit}$	
=	=	
1	1	→ lesen
0	1	→ 0 schreiben
1	0	→ 1 schreiben

DRAM

Dynamic random access memory



Bit	
=	
gesetzt	→ schreiben
nicht gesetzt	→ lesen

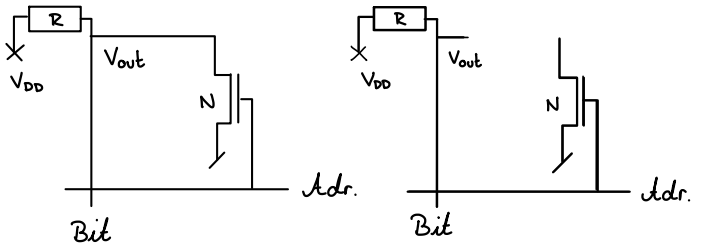
→ Daten werden periodisch aufgefrischt
↯ längere Zugriffszeit als SRAM

NICHT FLÜCHTIGE SPEICHER

Speichert auch ohne angelegter Spannung

ROM

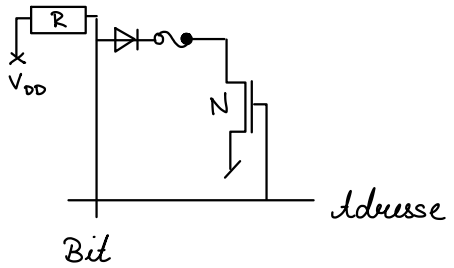
Read only memory



→ Wird zur Herstellungszeit programmiert

PROM

Programmierbares ROM



EPROM

Erasable PROM