

DIGITALTECHNIK

Zusammenfassung zur Vorlesung von
Prof. Dr. G. Tröster

Lukas Cavigelli, Januar 2010
lukasc@ee.ethz.ch

MOS-TRANSISTOREN / MOS-FET

MOS = metal oxide semiconductor
FET = field effect transistor, Transistor = *Trans*-Resistor

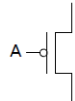
NMOS (NEGATIVE MOS)

- Leitet bei A high
- Nur gegen Masse/Ground verwenden
- „Pull-Down-Pfad“



PMOS (POSITIVE MOS)

- Leitet bei A low
- Nur gegen Speisung/VCC verwenden
- „Pull-Up-Pfad“

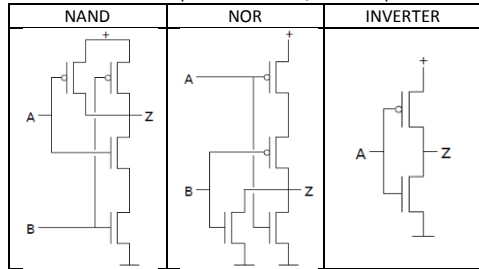


CMOS (COMPLEMENTARY-MOS)

Es leitet der Pull-Up- oder der Pull-Down-Pfad, nie keiner/beide
CMOS vs. NMOS-Technik:

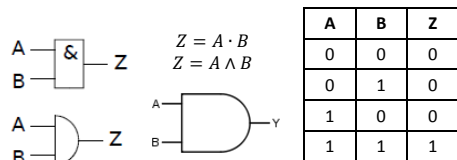
- Bei CMOS fließt nur beim Schalten Strom
- Das PMOS ersetzt den Widerstand
- CMOS: Verlustleistung linear zur Schaltfrequenz

Anderer Pfad zeichnen: parallel → seriell, seriell → parallel

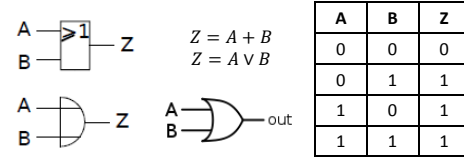


GATTER

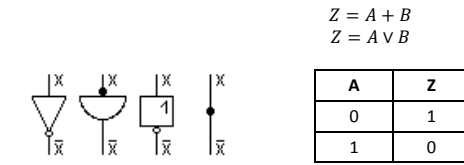
AND



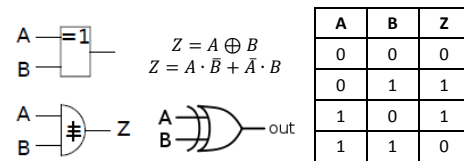
OR



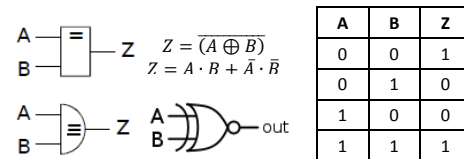
NOT / INVERTER



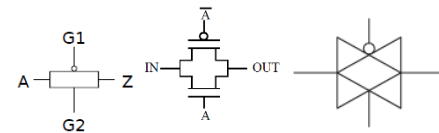
XOR (EXOR)



XNOR



TRANSMISSION GATE

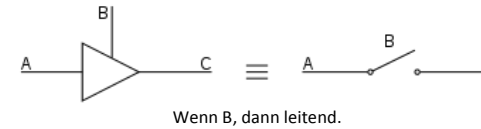


Verwendung: Anbindung Device → Bus, NICHT Interrupt

G1 (PMOS)	G2 (NMOS)	Z
0	0	0
0	1	A
1	0	Z „floating“
1	1	1

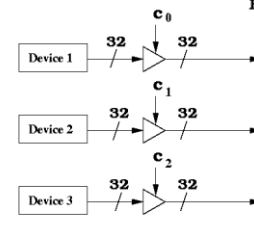
		Boolesche Logik	Algebra	Schaltb
OR	Disjunktion	$x \vee y$	$x + y$	
NOR	Neg. Disjunktion	$\neg(x \vee y)$	$\overline{x + y}$	
AND	Konjunktion	$x \wedge y$	$x \cdot y$	
NAND	Neg. Konjunktion	$\neg(x \wedge y)$	$\overline{x \cdot y}$	
NOT	Negation	$\neg x$	\overline{x}	
XOR	Exklusiv-Oder	$x \neq y$	$x \oplus y$	
XNOR	Äquivalenz	$x = y$	$\overline{x \oplus y}$	
	Implikation	$x \rightarrow y, \neg x \vee y$	$x \leq y$	

TRI-STATE-BUFFER (ÄHNL. TRANSM.-GATE)

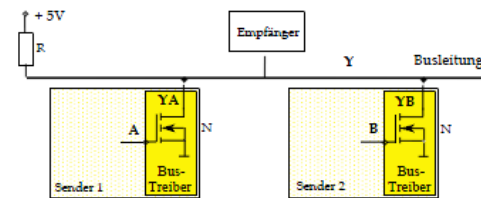


A	B „output enable“	C
0	0	C = X, „floating“
1	0	0
0	1	0
1	1	1

Verwendung: Anbindung Device → Bus, NICHT Interrupt



WIRED-AND



Y high, solange kein Device durchschaltet

Wired-OR: Mit Widerstand gegen GND, Devices gegen VCC

LOGIK

DE MORGAN

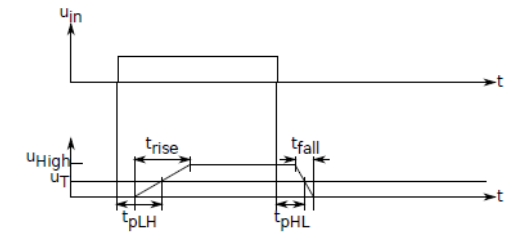
$$Z = \overline{A \cdot B} = \overline{A} + \overline{B} \quad \text{bzw.} \quad Z = A \cdot B = \overline{\overline{A} + \overline{B}}$$

$$Z = \overline{A + B} = \overline{A} \cdot \overline{B} \quad \text{bzw.} \quad Z = A + B = \overline{\overline{A} \cdot \overline{B}}$$

ALLGEMEIN

Doppelte Negation: $\overline{\overline{A}} = A$
 Kommutativität: $A \cdot B = B \cdot A$ $A + B = B + A$
 Assoziativität: $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
 $A + (B + C) = (A + B) + C$
 Distributivität: $A \cdot (B + C) = A \cdot B + A \cdot C$
 $A + (B \cdot C) = (A + B) \cdot (A + C)$
 Idempotenz: $A \cdot A = A$ $A + A = A$
 Komplemente: $A \cdot \overline{A} = 0$ $A + \overline{A} = 1$
 Neutrale: $A \cdot 1 = A$ $A + 0 = A$
 Dominanz: $A \cdot 0 = 0$ $A + 1 = 1$
 Absorption: $A \cdot (A + B) = A$ $A + (A \cdot B) = A$

TIMING



t_{rise} = Anstiegszeit der steigenden Flanke.

t_{fall} = Fallzeit der fallenden Flanke.

t_{pLH} = Zeit, bis das Signal der steigenden Flanke am Ausgang als "high" interpretiert wird. Als Schwelle dient die Referenzspannung u_T .

t_{pHL} = Zeit, bis das Signal der fallenden Flanke am Ausgang als "low" interpretiert wird. Als Schwelle dient die Referenzspannung u_T .

$$u_T = u_{High}/2$$

NORMALFORMEN

MINTERME (=VOLLKONJUNKTION)

UND-Verknüpfungen, welche alle Schaltvariablen genau einmal in negierter oder unnegierter Form enthalten.

Für 2 Var. A und B: z.B. $\overline{A} \cdot B$

MAXTERME (=VOLLDISJUNKTION)

ODER-Verknüpfungen, welche alle Schaltvariablen genau einmal in negierter oder unnegierter Form enthalten.

Für 2 Var. A und B: z.B. $A + B$

KANONISCH DISJUNKTIVE NORMALFORM DNF

Verknüpfung *aller* Minterme einer Schaltfunktion.

Für 3 Var. A, B, C: z.B. $Z = A \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot C$

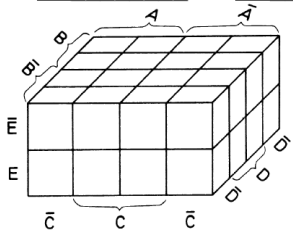
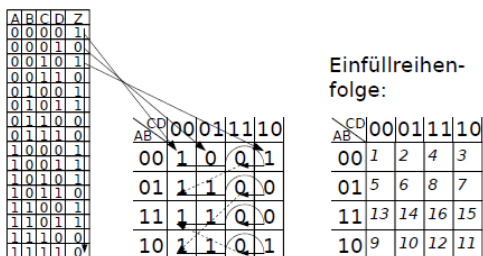
KANONISCH KONJUNKTIVE NORMALFORM KNF

Verknüpfung *aller* Maxterme einer Schaltfunktion.

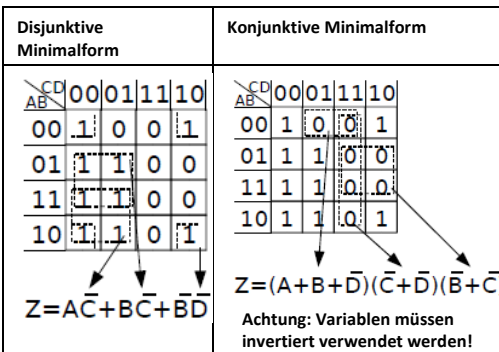
Für 3 Var. A, B, C: z.B. $Z = (A + B + C) \cdot (\overline{A} + B + \overline{C}) \cdot (\overline{A} + \overline{B} + C)$

KARNAUGH (KV)-DIAGRAMME

1. EINFÜLLEN AUS WAHRHEITSTABELLE



2. KV-DIAGRAMM -> LOGIKFUNKTION



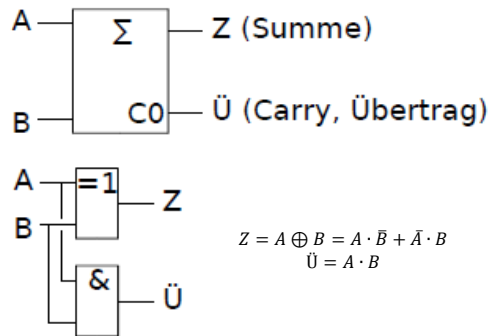
Päckchen können auch über Ränder und Ecken gemacht werden. Sie sind aber immer rechteckig und 2^n gross.

3. HAZARDS, SPIKES, GLITCHES

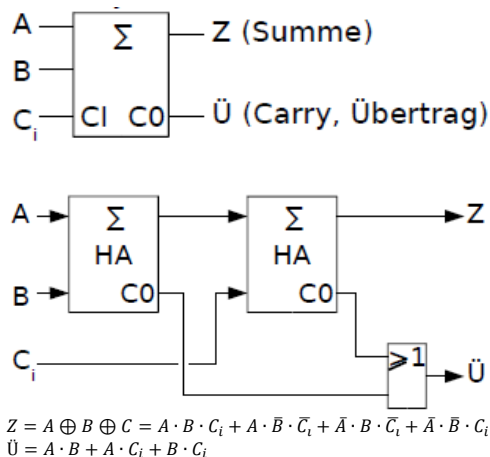
Können auftreten, wenn zwei Zustandsvariablen gleichzeitig ändern und die Gatter nicht genau gleichzeitig schalten.
Vermeidung: zusätzliche Blöcke in KV-Diagramm einfügen.
 Gibt es keine Übergänge mehr, die nicht abgedeckt sind, so ist die Schaltung hazardfrei (nicht immer möglich). Gray-Code

ADDIERER

HALBADDIERER

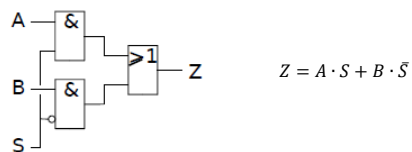


VOLLADDIERER

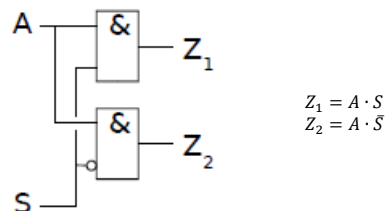


(DE-)MULTIPLEXER (MUX)

MULTIPLEXER

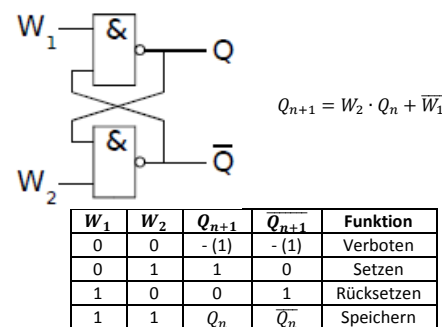


DEMUTIPLEXER

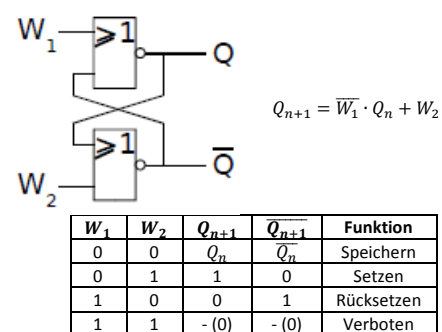


LATCHES

NAND-LATCH (SR-LATCH)



NOR-LATCH (SR-LATCH)



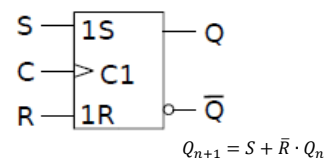
TAKTGESTEuerte LATCHES UND D-LATCH

Taktgesteuerte Latches, im Gegensatz zu FFs, nicht taktflanken gesteuert, sondern taktgesteuert. D.h. Latches sind transparent (=sofort schaltend) während Takt = Clk = high.
 D-Latch: Aufbau wie D-FF

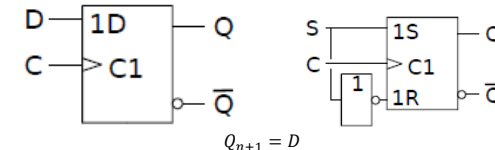
FLIPFLOPS (FF) / BISTABILER MULTIVIBRATOR

Unterschied zu Latches: FFs können Zustand nur bei steigender oder fallender Taktflanke ändern.
 Taktflankensteuerung: mit „Knödel“: bei fallender Flanke

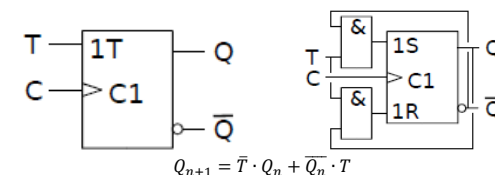
SET-RESET-FLIPFLOP (SR-FF)



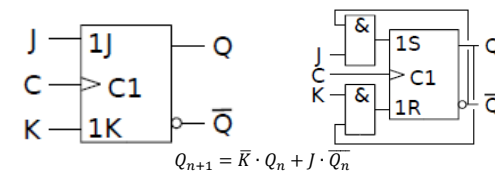
DELAY-FLIPFLOP (D-FF)



TOGGLE-FLIPFLOP (T-FF)



JUMP-KILL-FLIPFLOP (JK-FF)

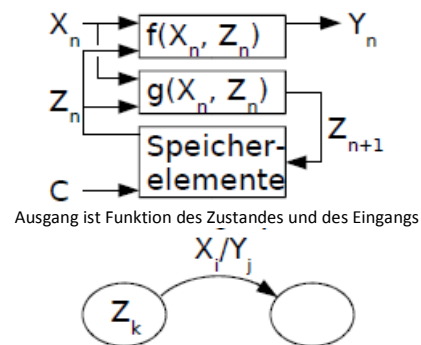


MASTER-SLAVE-FLIPFLOP

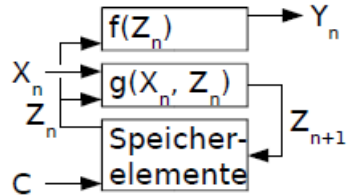
ENDL. AUTOMATEN / FINITE STATE MACHINE

Anzahl benötigte FFs: #FFs $\geq \log_2(\text{\#Zustände})$

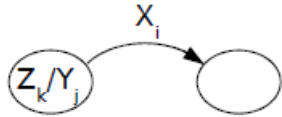
MEALY



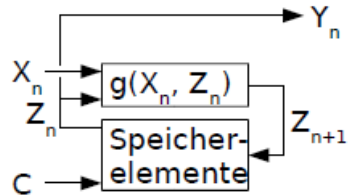
MOORE



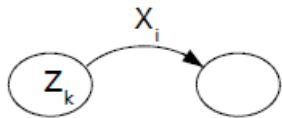
Der Ausgang ist nur vom Speicherzustand abhängig.



MEDWEDJEW



Der Ausgang entspricht den Speicherzuständen.



AUTOMATENSYNTHESE

1. Zustandsdiagramm erstellen
2. Zustandstabelle erstellen
3. Binäre Zustandskodierung
4. Binäre Zustandstabelle
5. Karnaugh

BINÄRE DARSTELLUNG

ZWEIERKOMPLEMENT

2er Komplemente negieren:
Alle Werte links von und ohne der ersten 1 von rechts flippen.

MOST & LEAST SIGNIFICANT BIT (MSB)

Selbsterklärend, aber wichtige Begriffe

ADDITION

-> Wie normales schriftliches Addieren

SUBTRAKTION

Zahl negieren und zur anderen addieren

MULTIPLIKATION

$$1100 \cdot 1101 = 10011100$$

$$1100 +$$

$$000000 +$$

$$1100000 +$$

$$11000000$$

$$= 10011100$$

DIVISION

1000010 / 11 = 0010110, Rest 0:

$$1 \text{ mod } 11 = 0, \text{ Rest } \underline{1}$$

$$\underline{10} \text{ mod } 11 = 0, \text{ Rest } \underline{10}$$

$$\underline{100} \text{ mod } 11 = 1, \text{ Rest } \underline{001}$$

$$\underline{0010} \text{ mod } 11 = 0, \text{ Rest } \underline{0010}$$

$$\underline{00100} \text{ mod } 11 = 1, \text{ Rest } \underline{00001}$$

$$\underline{000011} \text{ mod } 11 = 1, \text{ Rest } \underline{000000}$$

$$\underline{0000000} \text{ mod } 11 = 0, \text{ Rest } 0000000$$

ANSATZ

Ansatz: $a \cdot 2^0 + b \cdot 2^1 + c \cdot 2^2 + d \cdot 2^3 + \dots$

KONVERTIEREN ZWISCHEN ZAHLENSYSTEMEN

	Stellenwert						
	32	16	8	4	2	1	
Dualzahl	0	0	0	1	0	1	5
	1	0	0	0	1	1	35
	0	0	1	0	1	0	10
							Dezimalzahl

41(dez) zu (bin):

$$41 / 2 = 20, \text{ Rest } \underline{1} \rightarrow \text{LSB}$$

$$20 / 2 = 10, \text{ Rest } \underline{0}$$

$$10 / 2 = 5, \text{ Rest } \underline{0}$$

$$5 / 2 = 2, \text{ Rest } \underline{1}$$

$$2 / 2 = 1, \text{ Rest } \underline{0}$$

$$1 / 2 = 0, \text{ Rest } \underline{1} \rightarrow \text{MSB}$$

$$= \underline{101001}$$

KODIERUNGEN

Eigenschaften:

- Gray: keine Hazards beim linearen Zählen (nur 1 Bit ändert)
- BCD: Binary coded digit -> klar Ziffer ↔ Binär-Zuordnung
- 9er-Komplement durch invertieren: Exc3, Aiken
- Einschnittige Codes (hazard-frei): O'Brian, Gray, Gray erw.
- Einschnittige, zyklische Codes: O'Brian, Gray erw.
- Kein Wert bei Speisungsausfall (0000): Excess-3, O'Brian
- a-b-c-d: die Werte der jeweiligen Binärstelle

Binär	Normal 0	Komplement	Gray (erw.)	BCD 8-4-2-1	Excess-3	Aiken 2-4-2-1	4-2-2-1	O'Brian
0000	0		0	0		0	0	
0001	1		1	1		1	1	
0010	2		3	2		2	2	0
0011	3		2	3	0	3	3	
0100	4		7	4	1	4		4
0101	5		6	5	2			3
0110	6		4	6	3		4	1
0111	7		5	7	4		5	2
1000	8		(15)	8	5			
1001	9		(14)	9	6			
1010	10		(12)		7			9
1011	11		(13)		8	5		
1100	12		8		9	6	6	5
1101	13		9			7	7	6
1110	14		(11)		8	8	8	8
1111	15		(10)			9	9	7

TECHNISCHE DATEN

LOGIKPEGEL

Technologie	Eingang		Ausgang	
	Low V_{IL}	High V_{IH}	Low V_{OL}	High V_{OH}
TTL 5V	≤ 0.8	≥ 2.0	≤ 0.4	≥ 2.4
CMOS 5V	≤ 1.5	≥ 3.5	≤ 0.5	≥ 4.44
LVTTTL 3.3V	≤ 0.8	≥ 2.0	≤ 0.4	≥ 2.4
CMOS 2.5V	≤ 0.7	≥ 1.7	≤ 0.2	≥ 2.3
CMOS 1.8V	≤ 0.7	≥ 1.17	≤ 0.45	≥ 1.2
RS-232 *	-15 bis -3	+3 bis +15	-15 bis -5	+5 bis +15

* negative Logik, d.h. low=1, high=0

Gemäss Skript (5V CMOS): 0V - 0.8V low, 4.5V - 5.5V high

CMOS-ZEITEN

Symbol	Bedeutung	Messung
t_{PHL}	Propagation delay high->low	bei 50% des Pegelhubs
t_{PLH}	Propagation delay low->high	bei 50% des Pegelhubs

	low->high	Pegelhubs
$t_R = t_{TLH}$	Anstiegs- (Rise-) Zeit Transition low->high	zwischen 10% und 90% des Pegelhubs
$t_F = t_{THL}$	Abfall- (Fall-) Zeit Transition high->low	zwischen 90% und 10% des Pegelhubs

WICHTIGE ZAHLENSYSTEME

Dual/Binär	Oktal	Dezimal	Hexadezimal
0000	00	00	0
0001	01	01	1
0010	02	02	2
0011	03	03	3
0100	04	04	4
0101	05	05	5
0110	06	06	6
0111	07	07	7
1000	10	08	8
1001	11	09	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

FEHLERQUELLEN

- Alle Tabellen beachtet (auch Aufgabenblatt)
- Richtiger Automat gewählt?
- Hazards
- PMOS/NMOS richtig?
- MSB <- -> LSB richtig?

HÄUFIGE FRAGEN

- Schaltung auch nur NOR
- Wieso nicht NMOS gegen VCC schalten, bzw. PMOS -> GND

TUDO

korrektheit bei Überlauf
Was, wenn: tristatebuffer auf EN, A low, Z high
Automaten/Karnaugh mit gegebenem FF-Typ
Automaten-Beispiele: Zähler, Freq-Teiler
Multiplizierer: shift & add
interrupt-leitung, wired-or
Booth-Algo
Karnaugh für spezielle FFs