

Rechnerstrukturen

3. Elementare Bausteine

Inhalt

- ▼ Latches und Register
- ▼ Decoder
- ▼ Multiplexer
- ▼ Speicher
- ▼ Arithmetische Einheiten
- ▼ Endliche Automaten

Elementare Bausteine

- ▼ Häufig verwendete Grundfunktionen
 - Umwandeln (Decoder)
 - Verteilen (Multiplexer) und Zusammenfassen (Demultiplexer)
 - Speicherung
 - Steuerungen (Endliche Automaten, State Machine)
 - Arithmetisch-logische Funktionen
 - Zählen (Zähler, Counter)
 - Addieren, Subtrahieren, ...
 - Vergleichen (Komparator)
 - Shiften, Rotieren
- ▼ Funktionsweise, Realisierungsvarianten, Zeit- und Platzeffizienz
- ▼ Spezifikation
 - Ein- und Ausgänge, Steuerungsanschlüsse
 - Zeitverhalten, elektrische Eigenschaften



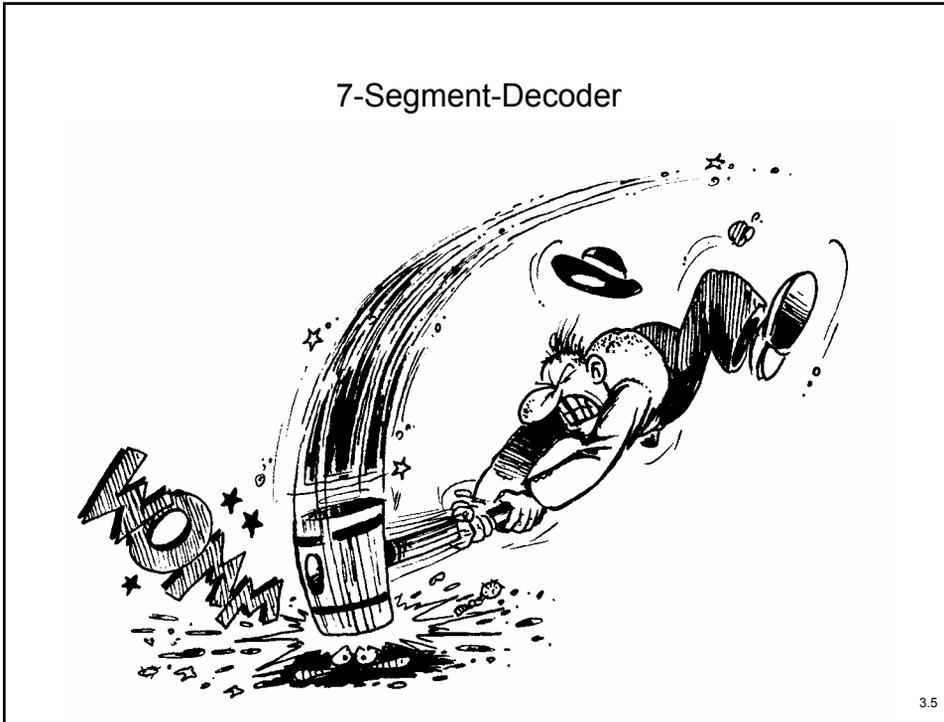
3.3

Decoder

- ▼ Umwandlung
- ▼ Inverse Funktion häufig auch interessant
 - Decoder
 - Encoder
- ▼ Beispiele
 - 7-Segment
 - 1 aus n
 - Priorität
 - Parität
- ▼ Klassische Schaltnetze
 - Minimierung



3.4



Realisierung: 7-Segment-Decoder

▼ Zusätzliche Steuereingänge

- LT = Lamp Test
- BI = Blank Input
- RBI = Ripple Blank Input
- RBO = Ripple Blank Output

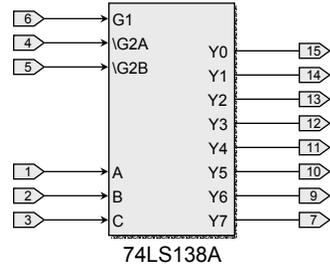
74LS47

\LT	\RBI	D	C	B	A	\BI/\RBO	a	b	c	d	e	f	g
H	H	L	L	L	L	H	L	L	L	L	L	L	H
H	X	L	L	L	H	H
.
H	X	H	H	H	H	H
X	X	X	X	X	X	L	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
L	X	X	X	X	X	H	L	L	L	L	L	L	L

3.6

“1 aus n”-Dekoder

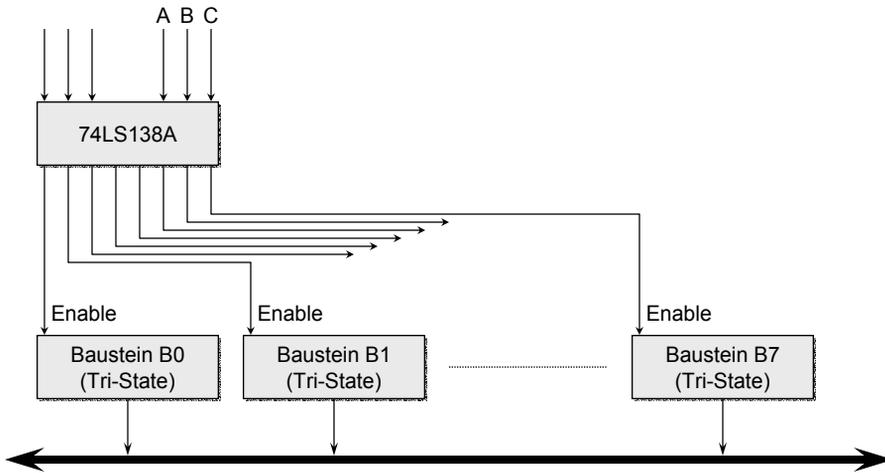
- ▼ Maximal 1 aus n Ausgängen aktiv
 - 1 aus 4
 - 1 aus 8
 - 1 aus 16
- ▼ Kaskadierbarkeit



G1	\G2A	\G2B	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	X	H	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H	H
H	L	L	L	H	L	H	H	H	H	L	H	H	H
H	L	L	H	H	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

3.7

Primäre Anwendung eines “1 aus n”-Decoders



3.8

Realisierung: "1 aus n"-Decoder

3.9

Kaskadierung

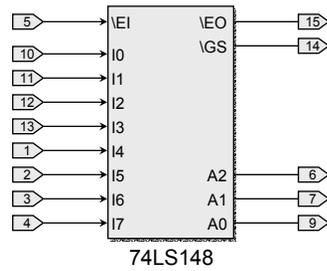
A
B
C
D

Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8
Y9
Y10
Y11
Y12
Y13
Y14
Y15

3.11

“1 aus n”-Encoder (Prioritätsdecoder)

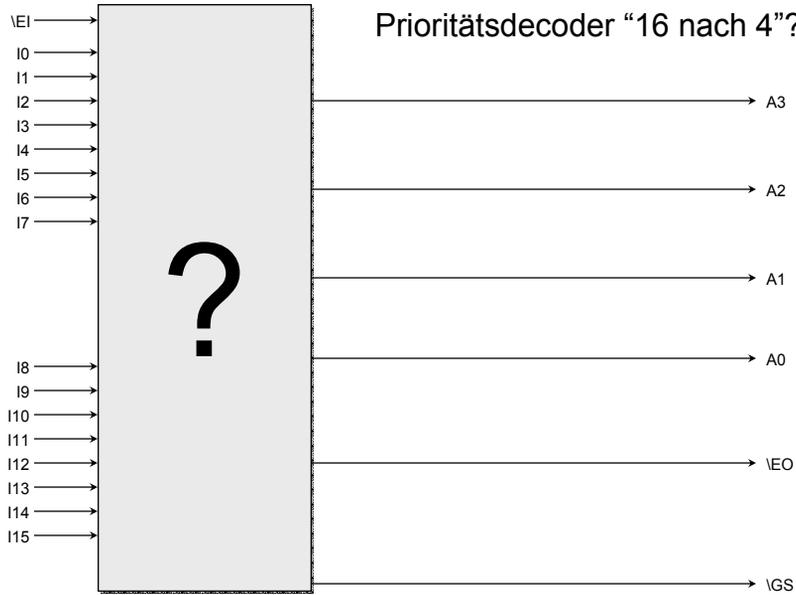
- ▼ Umkehrung des “1 aus n”-Decoders
 - 8 nach 3
 - 16 nach 4



\EI	I0	I1	I2	I3	I4	I5	I6	I7	A2	A1	A0	\GS	\EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	L	H	H	L	H
L	X	X	L	H	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

3.14

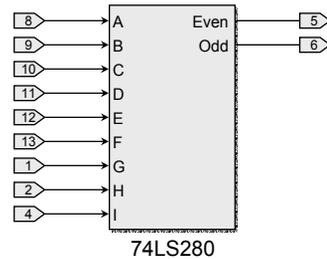
Prioritätsdecoder “16 nach 4”?



3.15

Paritätsdecoder

- ▼ Einfaches Prüfsummenverfahren
 - Gerade Parität: n Bits enthalten gerade Anzahl 1
 - Ungerade Parität: n Bits enthalten ungerade Anzahl 1
- ▼ Wahrheitstabelle etwas umständlich?
- ▼ Baustein einsetzbar
 - Erzeugung des Paritätsbit
 - Überprüfung des Paritätsbit



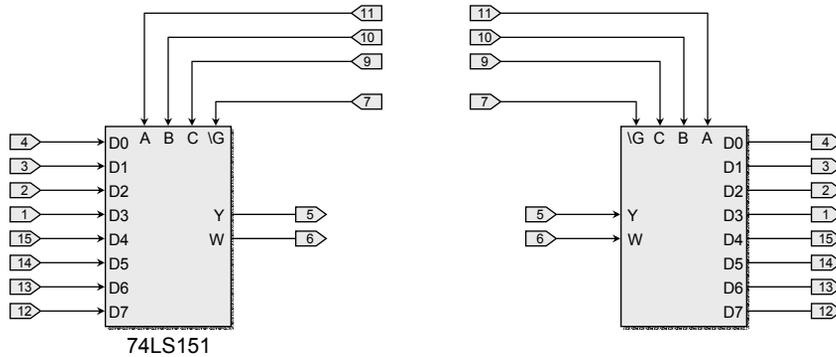
3.17

Realisierung: Paritätsdecoder

3.18

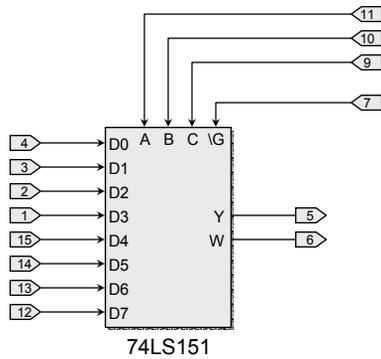
Multiplexer und Demultiplexer

- Erweiterung des einfachen Wechselschalters auf n Eingänge
- Demultiplexer: Umkehrung des Multiplexers



3.20

Funktionsweise Multiplexer



C	B	A	\overline{G}	Y	W
X	X	X	H	L	H
L	L	L	L	D0	\backslash D0
L	L	H	L	D1	\backslash D1
L	H	L	L	D2	\backslash D2
L	H	H	L	D3	\backslash D3
H	L	L	L	D4	\backslash D4
H	L	H	L	D5	\backslash D5
H	H	L	L	D6	\backslash D6
H	H	H	L	D7	\backslash D7

3.21

Realisierung: Multiplexer?

3.22

Kaskadierung

3.24

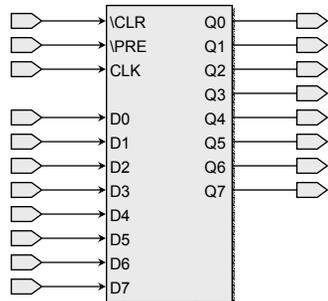
Register

- ▼ Speicherung von Bitvektoren
- ▼ Zusammenfassen mehrerer Flip-Flops
 - Manuell über externe Verdrahtung
 - Spezielle Registerbausteine
- ▼ Breites Spektrum
 - Verschiedene Flip-Flops
 - Preset
 - Clear
 - Tri-State-Ausgänge



3.26

Beispiel: 8 Bit-Register

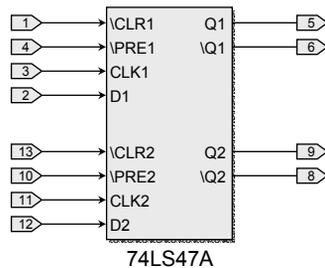
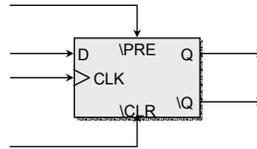


\CLR	\PRE	CLK	Dx	Qx
L	L	X	X	X
L	H	X	X	L
H	L	X	X	H
H	H	↑	L	L
H	H	↑	H	H

3.27

Realisierung: Mit 74LS47A

- ▼ 74LS47A: 2 D-Flip-Flop
 - Preset
 - Clear
- ▼ Wieviele Chips sind notwendig?
- ▼ Wieviele Ein/Ausgänge?



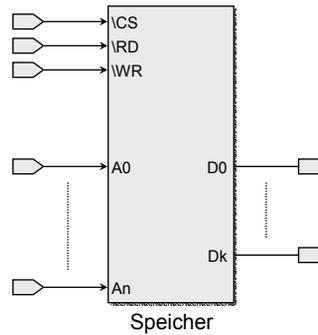
3.28

Realisierung: 8 Bit-Register?

3.29

Einfache Speicher

- ▼ Adressierung einzelner Speicherzellen
- ▼ Speicherbaustein
 - Kapazität (Anzahl Zellen, 2^n)
 - Größe des Datenvektors pro Adresse (k Bit)
- ▼ Tri-State-Ausgänge
- ▼ Häufig kombinierte Leitung $\overline{R/W}$
 - High = Lesen
 - Low = Schreiben
- ▼ Zwei Grundtypen
 - RAM: Lesen und Schreiben
 - ROM: Nur Lesen



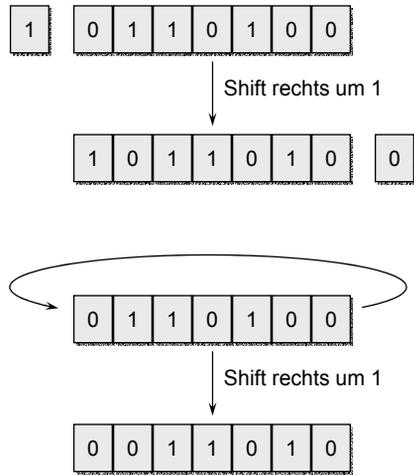
3.31

Realisierung: 16 mal 4 Bit RAM?

3.32

Shiften und Rotieren

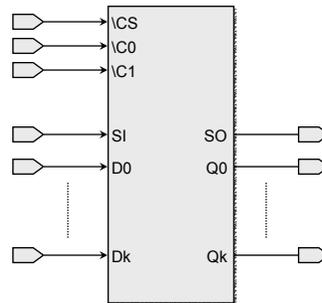
- ▼ Shift
 - Bitvektor um k Bit nach links oder rechts schieben
 - k Bit "fallen raus"
 - k Bit kommen hinzu
- ▼ Rotieren
 - Bitvektor um k Bit nach links oder rechts zyklisch schieben
- ▼ Betrachtung von Shift reicht?



3.37

Beispiel: Shift-Register

- ▼ 4 Zustände
 - Wert speichern
 - Wert laden
 - Shift links um 1
 - Shift rechts um 1
- ▼ Tri-State-Ausgang



\CS	\C0	\C1	Dx	Qx	SO	
L	X	X	Ω	Ω	Ω	
H	L	L	X	Qx	X	Hold
H	L	H	Dx	Dx	X	Load
H	H	L	X	Qx+1	Q0	Shift right, Qk=SI
H	H	H	X	Qx-1	Qk	Shift left, Q0=SI

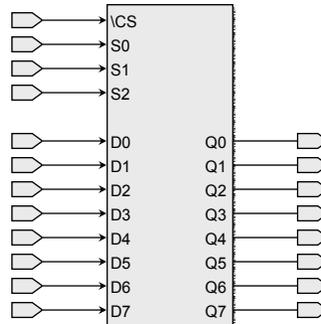
3.38

Realisierung: 2 Bit-Shift-Register?

3.39

Barrel-Shifter

- ▼ Jede beliebige Shift-Operation eines k Bit-Vektors in einem Schritt
- ▼ Beispiel k=8
- ▼ Realisierung: Multiplexer
 - Für jeden Ausgang: 2^k nach 1



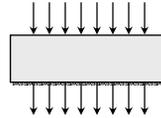
\CS	S0	S1	S2	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0
L	L	L	L	D7	D6	D5	D4	D3	D2	D1	D0
L	L	L	H	D6	D5	D4	D3	D2	D1	D0	D7
L	L	H	L	D5	D4	D3	D2	D1	D0	D7	D6
L	L	H	H	D4	D3	D2	D1	D0	D7	D6	D5
L	H	L	L	D3	D2	D1	D0	D7	D6	D5	D4
L	H	L	H	D2	D1	D0	D7	D6	D5	D4	D3
L	H	H	L	D1	D0	D7	D6	D5	D4	D3	D2
L	H	H	H	D0	D7	D6	D5	D4	D3	D2	D1

3.41

Serielle/Parallele Shift-Register

- ▼ Bisher Parallel/Parallel

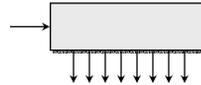
- Eingabe: Bitvektor
- Ausgabe: Bitvektor



- ▼ Umwandlung Seriell-Parallel

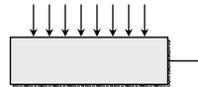
- ▼ Seriell nach Parallel

- Eingabe: 1 Bit
- Ausgabe: Bitvektor



- ▼ Parallel nach Seriell

- Eingabe: Bitvektor
- Ausgabe: 1 Bit



3.42

8 Bit-Seriell In/Parallel Out-Shift-Register?

- ▼ Welche Ein- und Ausgänge sind notwendig?
- ▼ Realisierung?

3.43

Logische und arithmetische Operationen

- ▼ Bitvektor
 - Logische Operationen bitweise
 - Gleichheit
- ▼ Bitvektor wird als Dualzahl aufgefaßt
 - Ganze Zahl ohne Nachkomma, Festkommazahl
 - Addition, Subtraktion, Multiplikation, Division
 - Gleichheit
 - Größer, Kleiner
- ▼ Bitvektor ist BCD-kodierte Dezimalzahl
 - Addition, Subtraktion, Multiplikation, Division
 - Gleichheit
 - Größer, Kleiner
- ▼ Bitvektor ist IEEE 754-kodierte Gleitkommazahl

3.47

Exkurs: Binäre Zahlensysteme

- ▼ Hier nur ganze Zahlen oder Festkommazahlen
- ▼ Zahlen zur Basis 2

$$01101.111_2$$

$$= 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$= 8 + 4 + 1 + 0.5 + 0.25 + 0.125 = 13.875_{10}$$
- ▼ Feste Anzahl Stellen
 - 8, 16, 32, 64 Bit gängig
 - Führende Nullen
- ▼ Darstellung negativer Zahlen?

3.48

2er-Komplement

- ▼ Zahl negativ \Leftrightarrow MSB = 1

$$-N = 2^n - N$$

- ▼ Einfache Umwandlung

- 1) 1er-Komplement

- 0 \rightarrow 1
- 1 \rightarrow 0

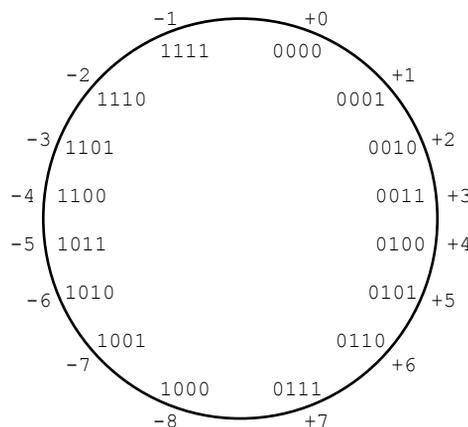
- 2) 1 addieren

-00110101 (-53)

```

11001010
+00000001
-----
11001011

```



3.49

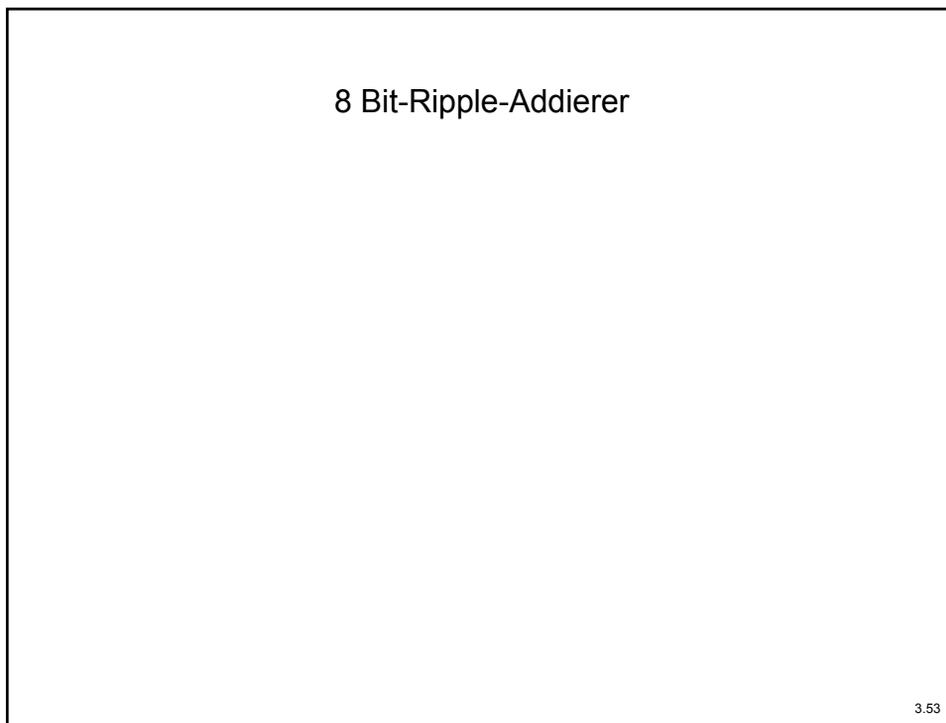
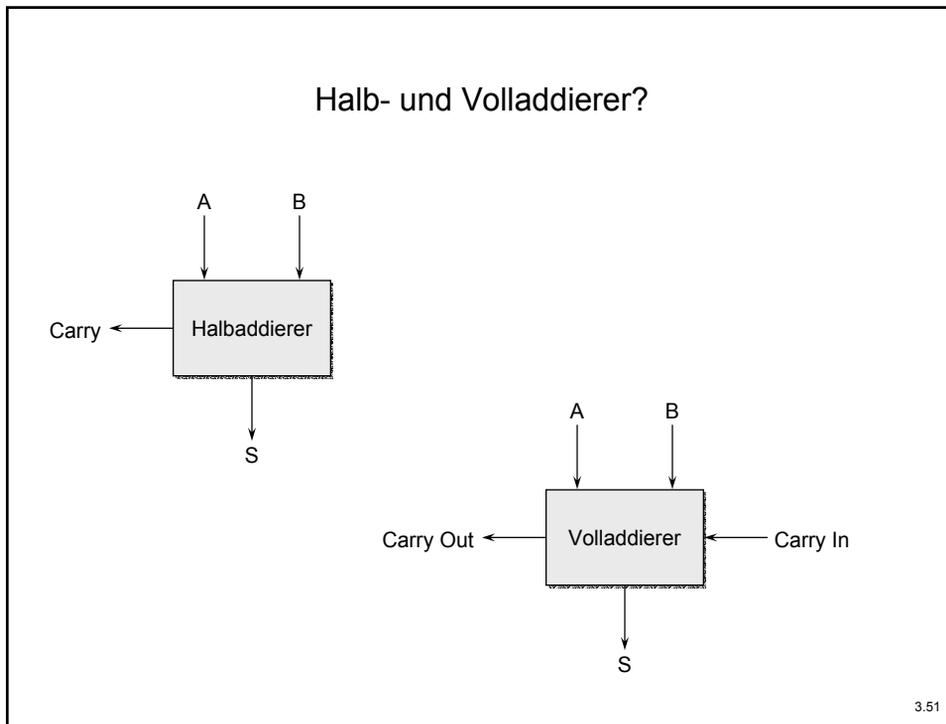
Addition

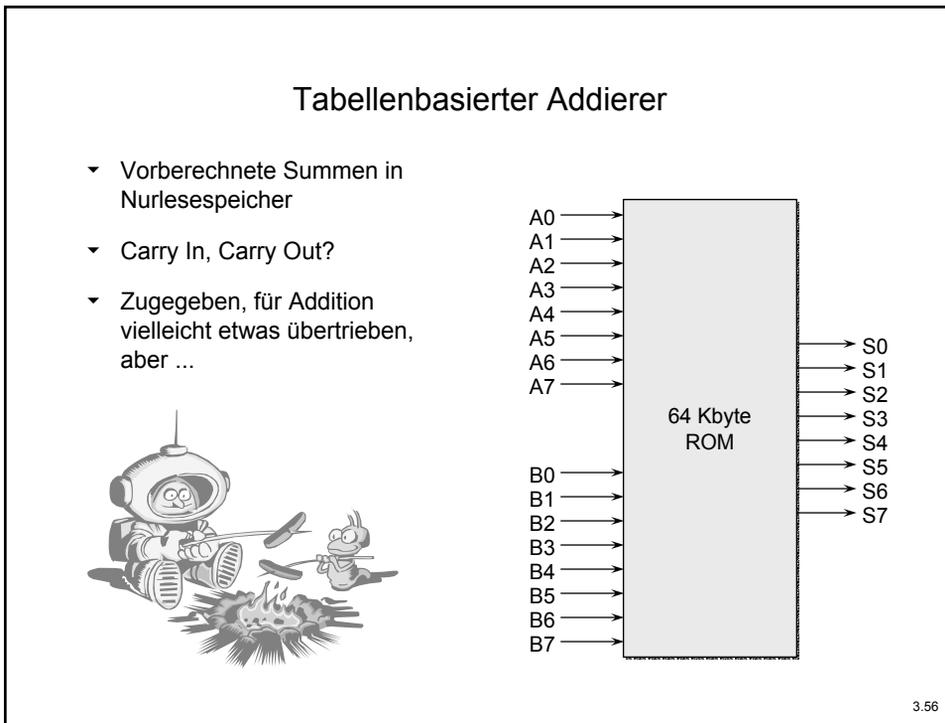
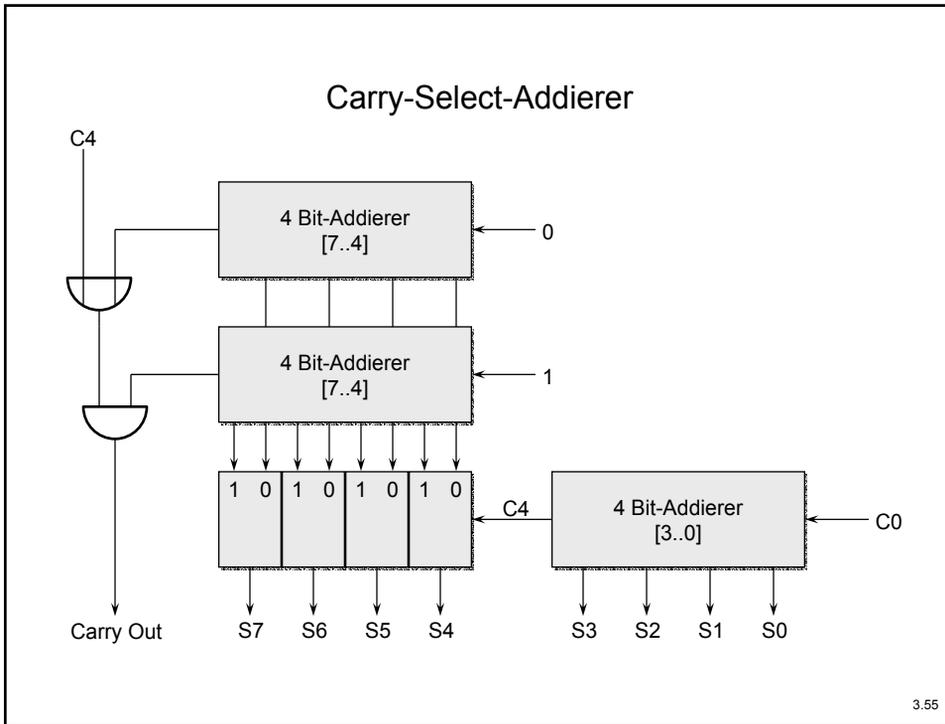
```

  00101111
+ 01111101
-----
 10101100

```

3.50





Carry-Lookahead-Addierer

- ▼ Ripple-Addierer

$$C_{i+1} = A_i B_i + C_i A_i + C_i B_i$$

$$C_0 = 0$$

- Laufzeit $O(n)$

- ▼ Carry-Lookahead

- Direkte Berechnung eines Carry-Bits

$$C_{i+1} = f(A_0, \dots, A_i, B_0, \dots, B_i)$$

- Zweistufiges Schaltnetz
- Laufzeit $O(1)$



3.57

Carry-Generate und Carry-Propagate

- ▼ Wann entsteht ein Carry?

$$G_i = A_i \cdot B_i$$

- ▼ Wann wird ein Carry weitergegeben?

$$P_i = A_i \oplus B_i$$

- ▼ Addition durch G und P beschreiben

$$S_i = A_i \oplus B_i \oplus C_i = P_i \oplus C_i$$

$$C_{i+1} = A_i B_i + A_i C_i + B_i C_i$$

$$= A_i B_i + C_i (A_i + B_i)$$

$$= A_i B_i + C_i (A_i \oplus B_i)$$

$$= G_i + C_i P_i$$

3.58

4 Bit Carry-Lookahead-Addierer

3.59

Subtraktion

- ▼ 2er-Komplement addieren
- ▼ Erweiterung n Bit-Addierer zu n Bit-Addierer/Subtrahierer?

3.61

Multiplikation

- ▼ Vorzeichenlos
- ▼ Einfach erweiterbar auf vorzeichenbehaftete Multiplikation?
- ▼ Maximale Größe des Resultats?
 - n Bit Multiplikant
 - n Bit Multiplikator
- ▼ Schaltnetz
- ▼ Akkumulation der partiellen Produkte

$$\begin{array}{r}
 1001 \quad (9) \\
 * 1011 \quad (11) \\
 \hline
 1001 \\
 1001 \\
 0000 \\
 1001 \\
 \hline
 1100011
 \end{array}$$

3.63

Akkumulation der partiellen Produkte?

	A_3	A_2	A_1	A_0		
	B_3	B_2	B_1	B_0		
S_6	S_5	S_4	S_3	S_2	S_1	S_0

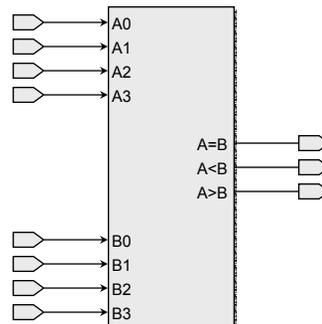
3.64

Realisierung mit Halb- und Volladdierern?

3.66

Komparatoren

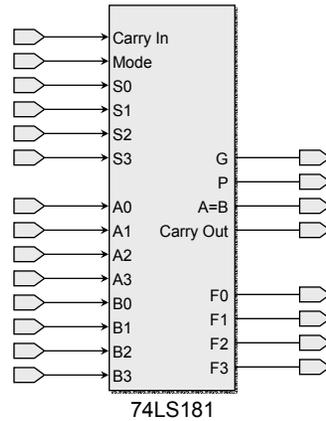
- ▼ Test auf
 - Gleichheit
 - Kleiner
 - Größer
- ▼ Realisierung
4 Bit-Komparator?



3.68

Arithmetische-logische Einheit (ALU)

- ▼ Zusammenfassung vieler arithmetischer und logischer Operation
- ▼ Beispiel 74LS181
 - Mode = 1: Logik
 - 16 Funktionen
 - Mode = 0: Arithmetik
 - 32 Funktionen
 - z.T. außergewöhnlich
 $F = AB \text{ plus } (A + \text{not } B) \text{ plus } 1$
 - G, P?
 - Auch als Komparator einsetzbar



3.70

2 Bit-ALU

3.71

BCD-Arithmetik

- ▼ BCD = Binary Coded Decimal

```

0000 0
0001 1
...
1001 9

```

- ▼ Beispiel Addition
 - Realisierungsvarianten?
 - Explizite Modellierung
 - Binäre Addition und Korrektur?
- ▼ Umwandlung Binär ↔ BCD

$$\begin{array}{r}
 0111 \quad (7) \\
 + 0110 \quad (6) \\
 \hline
 1101_2 \quad (13)
 \end{array}$$

$$\begin{array}{r}
 0111 \quad (7) \\
 + 0110 \quad (6) \\
 \hline
 1\ 0011_{\text{BCD}} \quad (1,3)
 \end{array}$$

3.73

Gleitkomma-Arithmetik

- ▼ Getrennte Behandlung von Mantisse und Exponent
- ▼ Mantisse normalisieren
 - Welche Operation wird hier eingesetzt?



3.74