

**** Informationselektronik 1 ****

Praktikumsaufgaben WS 1994/95

Praktikum 1

1. Nehmen Sie die Wahrheitstabellen für folgende Gatter mit zwei Eingängen auf: UND, ODER, NAND, NOR, XOR.
2. Bauen Sie ein Äquivalenzgatter (XNOR) mit zwei Eingängen auf und ermitteln Sie die Wahrheitstabelle.
3. Bauen Sie ein 5-fach-NAND aus 2-fach NANDs.
4. Bauen Sie ein 5-fach-NAND mit minimaler Schaltungstiefe (aus beliebigen Elementen).
5. Bauen Sie ein 3-fach-NOR (aus beliebigen Elementen).
6. Ermitteln Sie die Wahrheitstabellen der Schaltungen gemäß Abbildung 1. Formulieren Sie Ihre Erkenntnisse daraus.
7. Bauen Sie ein 2-fach-XOR aus NANDs.
8. Bauen Sie ein 3-fach-XOR aus NANDs.
9. Bauen Sie ein 8-fach NAND aus zwei 4-fach NANDs und weiteren beliebigen Elementen. Angestrebt wird minimale Schaltungstiefe.

Für alle aufgebauten Schaltungen ist die Funktionsfähigkeit nachzuweisen, z. B. durch Aufnehmen der Wahrheitstabelle. Überlegen Sie, wie Sie bei vielen Eingängen (z. B. 8) die Testprozedur verkürzen können (Stichwort: geschickte Wahl von Testbeispielen).

Hinweis: Ein Schalter-Modul zum Einspeisen der Eingangsbelegungen wird gesondert bereitgestellt.

Praktikum 2

1. Bauen Sie mit Gattern der Experimentierplattform einen 1-aus-10-Decoder auf. Als Eingangssignale werden binär codierte Zahlenwerte geliefert. Die Ausgänge sollen aktiv HI sein. Für diese Aufgabe sind drei Schaltungsvarianten zu entwerfen und aufzubauen:

a) Die vier binären Eingänge können alle Werte von 0 bis FH annehmen. Werte > 9 sollen bewirken, daß alle Ausgänge inaktiv werden.

b) Es können nur Werte zwischen 0 und 9 auftreten. Versuchen Sie, die Don't Cares AH...FH zur Schaltungsoptimierung auszunutzen.

c) Es können alle Eingangsbelegungen von 0 bis FH auftreten. Die Schaltung soll sich wie im Fall a) verhalten, zusätzlich soll bei Anliegen von AH...FH ein Fehlersignal abgegeben werden. Es ist Ihnen freigestellt, entweder die Schaltung gemäß a) oder jene gemäß b) zu modifizieren.

2. Bauen Sie einen Codierer für eine 7-Segment-Anzeige auf (diese wird zusätzlich bereitgestellt). Die Eingänge sollen im 1-aus-10-Code aktiv LO (invertiert) anliegen. Die 7-Segment-Anzeige wird ebenfalls aktiv LO angesteuert. Als Signalquelle können wahlweise die Tasten oder einer der eingebauten 1-aus-10-Decoder verwendet werden.

Praktikum 3

Bauen Sie mit den Schaltmitteln der Experimentierplattform eine universelle 4-bit-Logikeinheit auf und erproben Sie diese mit vorbereiteten Testbeispielen. Die Eingänge:

- zwei 4-bit-Binärvektoren A, B
- ein Befehlscode i

Die Ausgänge:

- ein 4-bit-Binärvektor C (Resultat),
- ein Bedingungssignal "Resultat Null" (Zero Flag),
- ein Signal "Keine Operation" (NOP),
- ein Signal "Inkorrekte Operation" (Operation Exception).

Die Logikeinheit (Abbildung 2) soll folgende Operationen ausführen können:

- Keine Operation (NOP),
- bitweise Konjunktion $C = A \& B$,
- bitweise Disjunktion $C = A \vee B$,
- bitweise Antivalenz $C = A + B$,
- Transport von A ($C = A$),
- Transport von B ($C = B$).

Der Binärvektor A soll auch in invertierter Form verarbeitet werden können (für den Binärvektor B ist das nicht vorgeschrieben, kann aber auch implementiert werden). Darüber hinaus soll der Binärvektor A wahlweise mit 4 Bits oder mit den zwei niedrigstwertigen Bits und Nullerweiterung (Zero Extend) zugeführt werden können (die Nullerweiterung ist der Invertierung vorzuschalten).

Geben Sie einen möglichst kompakten Befehlscode an (nicht

mehr als 5 Bits). Sehen Sie dabei einen NOP-Code vor (entweder alles Nullen oder alles Einsen). Alle unzulässigen Codes sollen abgefangen werden und das Signal Operation Exception erregen.

Liefern Sie bitte:

- die Befehlsliste,
- einen Schaltplan (skizzenhaft, es reicht eine Bitposition)
- die zur Erprobung gewählten Testbeispiele mit den praktisch ermittelten Ergebnissen.

Praktikum 4

Bauen Sie mit den Schaltmitteln der Experimentierplattform eine 8-bit-Verschiebeeinheit auf und erproben Sie diese mit vorbereiteten Testbeispielen. Die Einheit soll folgende Operationen ausführen können:

1. Weitergabe des Binärvektors ohne Verschiebung
2. Linksverschieben um ein Bit
3. Rechtsverschieben um ein Bit
4. Linksrotieren um ein Bit
5. Rechtsrotieren um ein Bit

Das jeweils hinausgeschobene bzw. (beim Rotieren) umlaufende Bit soll zusätzlich als ausgangsseitiges Flagbit (FLAG OUT) erscheinen.

Beim Verschieben soll die frei werdende Position wahlweise mit Null oder mit einem eingangsseitigen Flagbit (FLAG IN) aufgefüllt werden.

Eingänge:

- ein 8-bit-Binärvektor
- ein Flagbit (FLAG IN)
- Steuerleitungen:
 - SHIFT
 - ROTATE (einfache Weitergabe, wenn $\text{SHIFT} \vee \text{ROTATE} = 0$)
 - LEFT (LO)/RIGHT (HI)
 - FILLER CTL (LO: mit Null füllen/ HI: mit FLAG IN füllen)

Ausgänge:

- ein 8-bit-Binärvektor
- ein Flagbit (FLAG OUT)

Hinweise:

1. Die Schaltung (Abbildung 3) soll rein kombinatorisch aufgebaut werden (kein Schieberegister).
2. Die Gatter können nach Zweckmäßigkeit (bequemer Verschaltung) gewählt werden; eine besondere Optimierung ist nicht gefordert (Sie dürfen z. B. ein 4-fach-NAND als 2-fach-NAND verwenden, wenn es gerade paßt).

Praktikum 5

Bauen Sie mit den Schaltmitteln der Experimentierplattform eine 4-bit-Arithmetikeinheit für Binärzahlen in 2er-Komplement-Darstellung auf und erproben Sie diese mit vorbereiteten Testbeispielen. Die Einheit (Abbildung 4) soll folgende Operationen ausführen können:

1. Addition/Subtraktion a) ohne und b) mit einlaufendem Übertrag
2. Bildung des Zweierkomplements (nur einer der Eingänge)

Ausgangsseitig sollen folgende Flagbits vorgesehen sein:

1. Ausgangsübertrag (CARRY)
2. Verlassen des Wertebereichs (OVERFLOW)

Eingänge:

- 4-bit-Binärzahl A
- 4-bit-Binärzahl B
- Eingangsübertrag (CARRY-IN)
- Steuerleitungen:
 - Addieren/Komplement
 - Eingangsübertrag berücksichtigen (CARRY INJECT)

Ausgänge:

- 4-bit-Resultat
- Ausgangsübertrag (CARRY)
- Bereichsüberschreitung (OVERFLOW)

Hinweis: Die interne Übertragsbehandlung ist beliebig

(Ripple Carry ist zulässig). Die Gatter können nach Zweckmäßigkeit (bequemer Verschaltung) gewählt werden; eine besondere Optimierung ist nicht gefordert (Sie dürfen z. B. ein 4-fach-NAND als 2-fach-NAND verwenden, wenn es gerade paßt).

Praktikum 6

Bauen Sie mit der Experimentierplattform ein 12-bit-Addierwerk für Binärzahlen in 2er-Komplement-Darstellung auf (Abbildung 5) und erproben Sie diese mit vorbereiteten Testbeispielen. Dafür werden 4-bit-Addierer zusätzlich bereitgestellt. Für die Überträge zwischen den Addierern und für den Ausgangsübertrag soll das Carry-Look-Ahead-Prinzip verwirklicht werden. Es soll möglich sein, den A-Operanden wahlweise als 12-bit-Wort oder als 8-bit-Wort mit Vorzeichenerweiterung (Sign Extend) zuzuführen. Das Addierwerk soll nur addieren können; Komplementbildung und Berücksichtigung eines Eingangsübertrages sind nicht gefordert.

Eingänge:

- 12-bit-Wort A
- 12-bit-Wort B
- Steuersignal LOAD WITH SIGN EXTEND

Ausgänge:

- 12-bit-Resultat
- Ausgangsübertrag (CARRY)

Hinweise:

1. Um die insgesamt 24 Eingänge bereitzustellen, wird eines der 12-bit-Worte aus einem Zwischenregister entnommen, das gesondert aufzubauen ist (Schaltungsempfehlung wird vorgegeben).

2. Die Gatter können nach Zweckmäßigkeit (bequemer Verschaltung) gewählt werden; eine besondere Optimierung ist nicht gefordert (Sie dürfen z. B. ein 4-fach-NAND als 2-fach-NAND verwenden, wenn es gerade paßt).

3. Tip: Bauen Sie das Addierwerk erst mit Ripple Carry auf und erproben Sie es. Ersetzen Sie dann den Ripple Carry nach und nach durch Carry Look-Ahead und überzeugen Sie sich, ob noch dieselben Ergebnisse herauskommen.

Praktikum 7

Bauen Sie mit der Experimentierplattform eine Arithmetikeinheit für 4-bit-BCD-Zahlen auf (Abbildung 6) und erproben Sie diese mit vorbereiteten Testbeispielen. 4-bit-Addierer werden zusätzlich bereitgestellt. Die Einheit soll folgende Operationen ausführen können:

1. Addition
2. Subtraktion

In beiden Operationen soll ein einlaufender Übertrag berücksichtigt werden.

Eingänge:

- Tetrade A
- Tetrade B
- Eingangsübertrag
- Steuerleitung ADD/SUBTRACT

Ausgänge:

- Resultattetrade (BCD-Darstellung)
- Ausgangsübertrag

Hinweise:

1. Es ist nicht notwendig, einlaufende Pseudotetraden zu erkennen.
2. Die Gatter können nach Zweckmäßigkeit (bequemer Verschaltung) gewählt werden.

Praktikum 8

Bauen Sie mit der Experimentierplattform einen Multiplizierer für zwei natürliche (vorzeichenlose) 4-bit-Binärzahlen auf und erproben Sie ihn mit vorbereiteten Testbeispielen. 4-bit-Addierer werden zusätzlich bereitgestellt.

Eingänge:

- Binärzahl A (Multiplikand)
- Binärzahl B (Multiplikator)

Ausgänge:

- 8-bit-Resultat

Hinweis: Die Gatter können nach Zweckmäßigkeit (bequemer Verschaltung) gewählt werden; eine besondere Optimierung ist nicht gefordert (Sie dürfen z. B. ein 4-fach-NAND als 2-fach-NAND verwenden, wenn es gerade paßt).

Praktikum 9

1. Bauen Sie ein schnelles D-Latch (Multiplexer-Prinzip) aus NAND-Gattern auf. Versuchen Sie, ob die Schaltung noch funktioniert, wenn das Übernahmesignal (GATE) mit anderer Polarität zugeführt wird, d. h. GATE ist aktiv HI und liegt direkt am Gatter mit dem D-Eingang, aber über einen Negator am Rückführungs-Gatter. Schalten Sie so viele Negatoren in Reihe (ungerade Anzahl!), bis die Schaltung garantiert nicht funktioniert. Das Ergebnis kann von Plattform zu Plattform abweichen.
2. Bauen Sie ein JK-Master-Slave-Flipflop aus NANDs auf. Überzeugen Sie sich davon, daß bei $J, K=0$ und aktivem Takt kurzzeitige Störungen auf J bzw. K nach der fallenden Taktflanke in den Slave übernommen werden und so am Ausgang erscheinen.
3. Betreiben Sie die Schaltung als T-Flipflop.
4. Ergänzen Sie die Schaltung (das JK-Flipflop) um DATA LOCKOUT und überprüfen Sie, ob das Fehlverhalten damit beseitigt worden ist. (Die Taktverzögerung müssen Sie experimentell bestimmen. Wieviele Negatoren in Reihe brauchen Sie dazu? Minimale Verzögerung wird angestrebt.)
5. Bauen Sie ein flankengesteuertes D-Flipflop aus NANDs auf.
6. Betreiben Sie die Schaltung als T-Flipflop.

7. Ergänzen Sie die Schaltung so, daß sie als JK-Flipflop arbeitet.

8. Bauen Sie aus einem der vorhandenen JK-Flipflops ein D-Flipflop.

9. Bauen Sie aus zwei vorhandenen JK-Flipflops eine Impulsfangschaltung. Diese Schaltung wird in den folgenden Praktika als Impulsnachweiser verwendet werden.

(Alle Aufgaben verstehen sich einschließlich Erprobung, z. B. durch Abprüfen der Automatentabelle. Wir verwenden statische Taktierung - Tasten - und Anzeige-LEDs.

Praktikum 10

1. Bauen Sie einen Single-Shot-Generator auf, der mit durchlaufendem Takt betrieben wird und auf Tastendruck einen einzigen Taktimpuls ausgibt.

2. Bauen Sie einen Start/Stop-Generator auf. Eingänge: durchlaufender Takt, Umschaltsignal RUN/STOP, Starttaste. Der Generator soll Zustände RUNNING und STOPPED haben. Auf ein STOP-Signal soll er aus RUNNING in STOPPED übergehen und keine Taktimpulse mehr abgeben. Im Zustand STOPPED soll bei Betätigung der Starttaste ein einziger Taktimpuls abgegeben werden, sofern STOP wirksam ist. Ist hingegen RUN wirksam, so soll der Zustand RUNNING wieder erreicht werden (Abgabe fortlaufender Taktimpulse). Die ausgegebenen Impulse müssen dem durchlaufenden Takt entsprechen. Sie dürfen keine Störungen, Verkürzungen, Verlängerungen usw. aufweisen, dürfen aber zum Original-Takt 180 Grad phasenverschoben (invertiert) sein. Bei aktivem RUN und Zustand RUNNING muß das Betätigen der Starttaste wirkungslos bleiben.

Erprobung: Mit LED-Anzeigen und Impulsfalle. Es wird der Taktgenerator der Experimentierplattform verwendet. Zusätzlich stehen Logikanalyzer zur Verfügung (aber nicht für jeden Experimentierplatz).

Praktikum 11

Bauen Sie mit den vorhandenen Flipflops ein 4-bit-Vorwärts-Rückwärts-Schieberegister auf (Abbildung 7). Alle Funktionen sollen voll synchron gesteuert werden. Die Funktionen sind:

- nichts tun
- paralleles Laden
- Vorwärtsschieben
- Rückwärtsschieben

Erprobung: statisch. Überprüfen Sie die Kaskadierbarkeit, indem Sie zwei Aufbauten miteinander koppeln.

Praktikum 12

Bauen Sie einen 4-bit-Vorwärts-Rückwärts-Binärzähler auf (Abbildung 8). Funktionen (alle synchron):

- nichts tun
- parallel laden
- vorwärts zählen
- rückwärts zählen.

Der Zähler soll für beide Zählrichtungen mit seinesgleichen kaskadierbar sein, wobei alle Zähler mit demselben Takt betrieben werden sollen (vollsynchroner Arbeitsweise).

Erproben Sie die Kaskadierbarkeit, indem Sie zwei Aufbauten miteinander koppeln.

Praktikum 13

Entwerfen Sie einen Schaltkreis, der wahlweise als 2:1 teilender Takttreiber, als Schieberegister mit parallelen Ausgängen, als Ringzähler oder als Johnson-Zähler betreibbar ist (Abbildung 9). Der Schaltkreis soll haben:

- einen asynchronen Rücksetzeingang,
- einen Takteingang,
- einen synchron wirkenden Stop-Eingang (ist dieser aktiv, wird bei laufendem Takt das Teilen bzw. Schieben angehalten),
- 2 Betriebsarteneingänge MODE 1,0:

MODE 1	MODE 0	Funktion
0	0	Taktteiler
0	1	Schieberegister
1	0	Ringzähler
1	1	Johnson-Zähler

Grundlage: JK-FFs mit asynchronen Setz- und Rücksetzeingängen.

In der Betriebsart "Taktteiler" soll von allen Ausgängen ein synchrones 2:1 geteiltes Signal geliefert werden.

Bauen Sie die Schaltung für vier Bitpositionen auf.

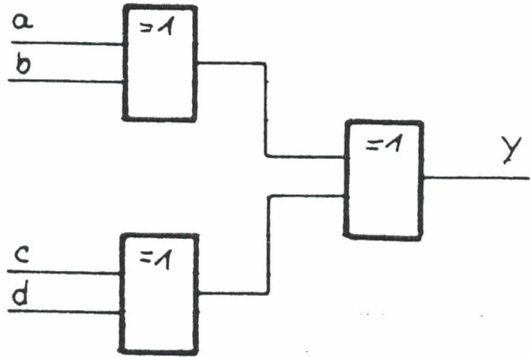
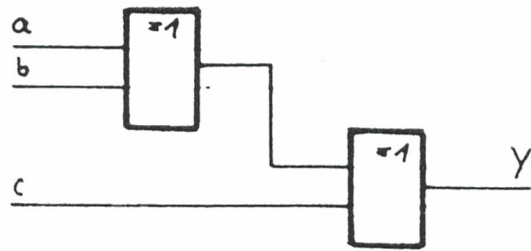


Abbildung 1

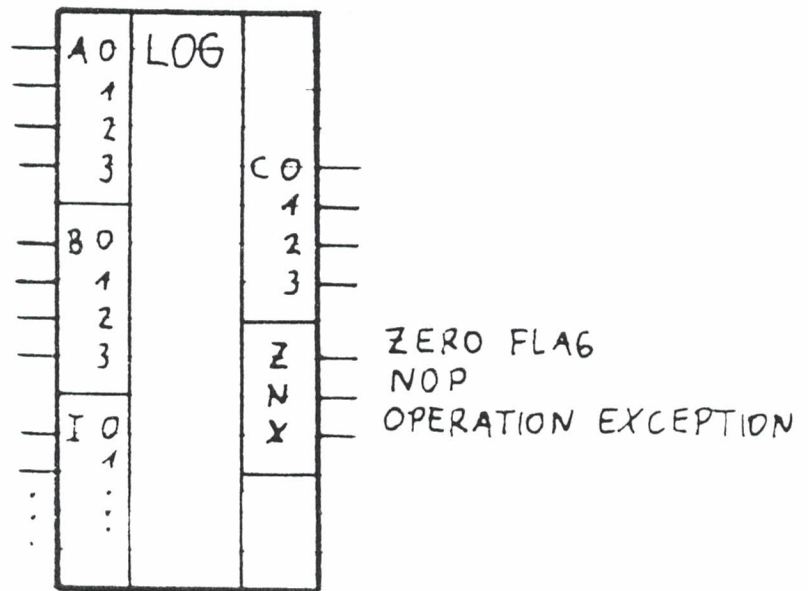


Abbildung 2

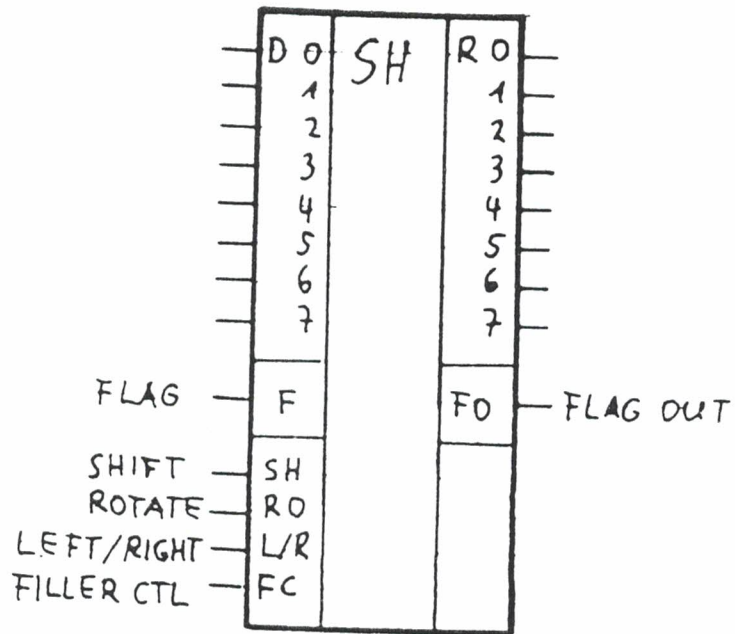


Abbildung 3

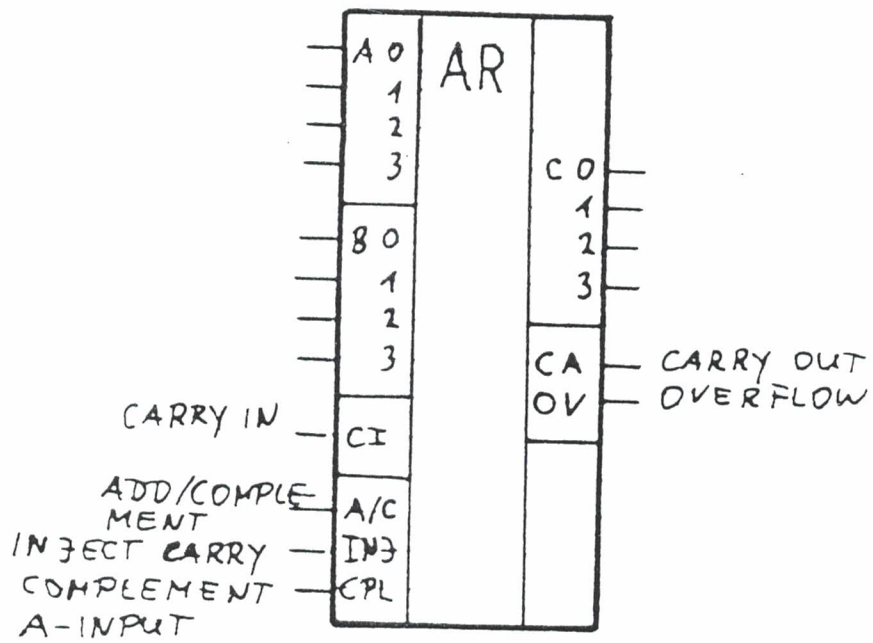


Abbildung 4

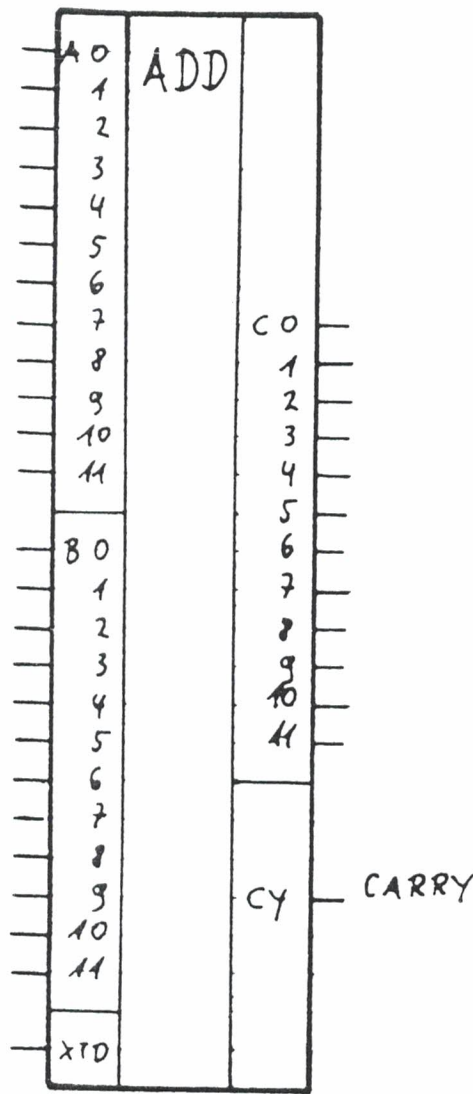


Abbildung 5

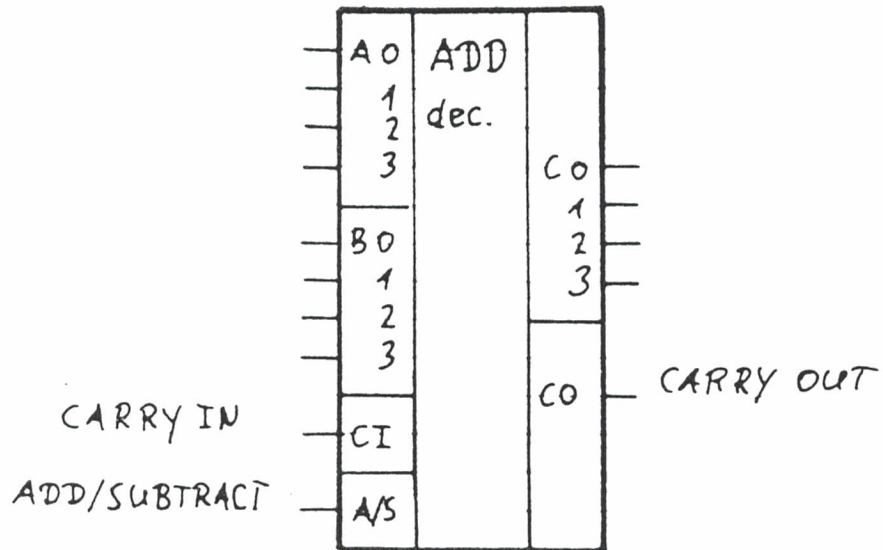


Abbildung 6

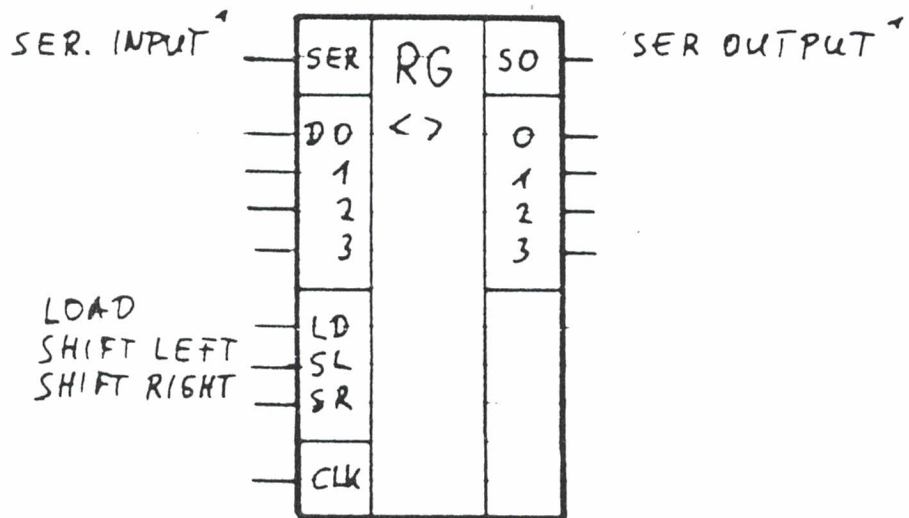


Abbildung 7

Vorrang: LOAD über SHIFT

LEFT über RIGHT

! umsteuerbar zwischen "links" und "rechts"

Experimentierplattform (1)

30 Negatoren



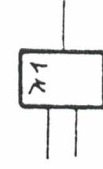
30 2-f. NAND



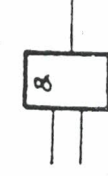
5 2-f. NAND/Schmitt-trigger



10 2-f. ODER



10 2-f. UND



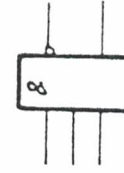
10 2-f. NOR



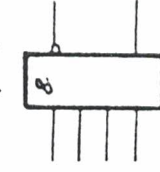
5 2-f. XOR



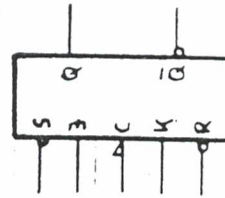
20 3-f. UND/NAND



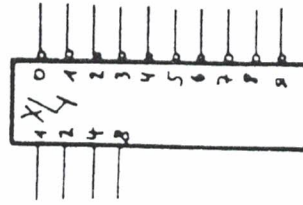
15 4-f. UND/NAND



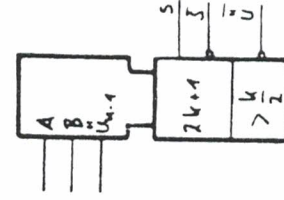
20 J-K-Flipflops



2 BCD-Decoder

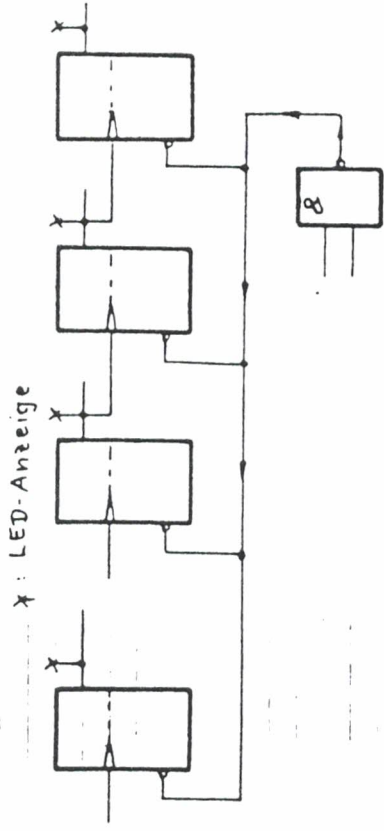


1 Adder



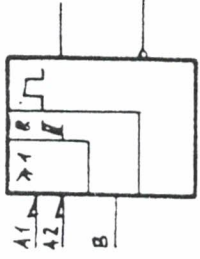
Experimentierplattform (2)

2 Zählerstufen (4 Flipflops)



x: LED-Anzeige

1 Monoflop 74121

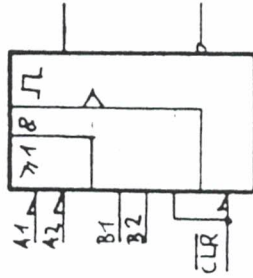


RC-Glied:

Poti 25 kΩ,
Kondens 1n, 10n
100n, 1μ, 10μ

- 12 Taster (prellfrei; wahre u. invertierte Ausgänge)
- 20 binäre LED-Anzeigen (5 HEX-Stellen)

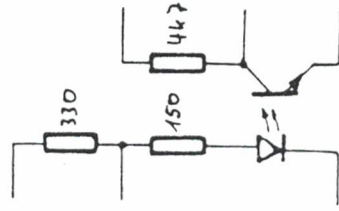
1 Monoflop 74122 (retriggerbar)



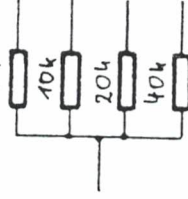
RC-Glied:

Poti 25 kΩ
Kondens. 100p, 1n,
10n, 100n, 1μ, 10μ

3 Optokoppler



1 Widerstandsnetzwerk



- 2 IC-Fassungen 16-pol.
- 3 8-pol. Fassungen für DP Amps; Zus. Stromversorgung ± 15 V
- Quarzgenerator (10 MHz, 1 MHz, 100 kHz, 10 kHz, 1 kHz, 100 Hz, 10 Hz, 1 Hz)
- durchstimmbarer Generator (5 MHz... 1 Hz)
- diverse Anschlüsse für feste HI- und LO-Pegel

Technik der DV-Anlagen

Probe-Prüfungsaufgaben

Stand: 4. 1. 93

Hinweis: Der Umfang geht etwas darüber hinaus, was als eigentliche Prüfungsarbeit vorgesehen ist. Die Prüfungsarbeit wird zudem einige recht elementare Aufgaben enthalten. Sie dürfen beliebige Unterlagen benutzen.

1. Es ist eine Schaltung anzugeben, die wahlweise eine Nullerweiterung bzw. eine Vorzeichenerweiterung von Bytes und 16-bit-Halbworten auf 32-bit-Worte ausführt. Die Steuersignale:

- OPERATION SELECT (0: ZERO EXTEND; 1: SIGN EXTEND)
- DATA TYPE 1,0 (0,0: BYTE; 0,1: HALFWORD; 1,0 und 1,1: WORD).

Es reicht aus, ein überblicksmäßiges Blockschaltbild sowie für jene Anordnungen, die sich mehrfach wiederholen, je eine Bitposition anzugeben.

2. Entwerfen Sie einen Empfänger für Morsezeichen. Der Punkttakt ist vorgegeben, und die ankommenden Morsezeichen sind zu diesem synchron. Die Schaltung soll folgende Ausgangssignale liefern:

- Punkt
- Strich
- Zeichenzwischenraum
- Wortzwischenraum
- kein Signal

Darzustellen sind: (1) Schaltbild, (2) Impulsdiagramm(e), erforderlichenfalls stichwortartige Funktionsbeschreibung und Zustandsgraph(en).

Morsezeichen:

Grundlage ist die Punkt-Länge.

Strich-Länge: 3 Punkt-Längen

Abstand zwischen zwei Zeichen eines Buchstabens: 1 Punkt-Länge

Abstand zwischen zwei Buchstaben: 3 Punkt-Längen

Abstand zwischen zwei Worten: 5 Punkt-Längen