

Basispraktikum Technische Informatik Sommersemester 2001

David Frese und Christos Freris

Inhalt

1	Geräteeinführung	5
1.1	Versuche mit dem Oszilloskop	5
1.1.1	Frequenzbestimmung	5
1.2	Ein Draht am Oszilloskop	5
1.2.1	Frequenzbestimmung	7
1.3	Tiefpaß	7
1.3.1	Vorbemerkung	7
1.3.2	Grenzfrequenz der Schaltung	7
1.3.3	Verhältnis von Eingangs- und Ausgangsspannung bei Sinusspannung	8
1.3.4	Verhältnis von Eingangs- und Ausgangsspannung bei Rechteckspannung	9
1.4	Kennlinie einer Diode	11
1.4.1	Versuchsaufbau	11
1.4.2	Durchlaßspannung	11
2	Transistorschaltungen	13
2.1	Messung der Kennlinie des n-Kanal MOS-FET BS107A	13
2.1.1	Schaltung	13
2.1.2	Kennlinie	14
2.2	Stromrückgekoppelte Emitterschaltung als Verstärker	14
2.2.1	Verstärkung	15
2.2.2	Verzerrungen	16
3	Digitale Elektronik	17
3.1	Inverter und NOR-Gatter	17
3.1.1	Kennlinie eines Inverters	17
3.1.2	Dynamisches RAM	17
3.1.3	NOR-Gatter	17
3.2	Aufbau eines XOR-Gatters	20
3.2.1	. . . mit Hilfe von NAND-Gattern	20
3.2.2	. . . mit Hilfe von AND-, OR- und NOT-Gattern	20
3.3	Hintereinanderschaltung drei Invertoren	22
3.3.1	Aufbau der Schaltung	22
3.3.2	Ausgangssignal und dessen Frequenz	22
3.4	Hazards	22

3.4.1	Aufbau der Schaltung	22
3.4.2	Eingangs- und Ausgangssignal von 5 V	22
3.4.3	Erklärung des Hazards	24
3.4.4	Eliminierung des Hazards	24
4	Digitaler Schaltungsentwurf	25
4.1	Entprellter Taster	25
4.2	Asynchroner 4-Bit BCD-Zähler	25
4.2.1	Ohne Entpreller	26
4.3	7-Segment-Anzeige	26
5	Mikrocontroller	33
5.1	DAvE	33
5.2	Programmcode	33
5.3	Kompilierung	34
5.4	Ausführung	34
5.4.1	Erklärung	34
6	Minirechner	37
6.1	Aufgabe 1 und 2	37
6.2	Division durch 7	37
6.3	4 Bit * 4 Bit Multiplikation	39
7	Mikroprozessor	41
7.1	Einbindung	41
7.2	Bitschiebeoperationen	41
7.2.1	Ergebnisse	42
7.3	Logische und arithmetische Operationen	42
7.3.1	Ergebnisse	43
7.4	Rechenoperationen	43
7.4.1	Ergebnisse	44
7.5	(4-Bit * 4-Bit)-Multiplikation	44
7.5.1	Ergebnisse	45

Versuch 1: Geräteeinführung

Protokollführer Christos Freris

Verwendete Geräte Oszilloskop HM303-6 Nr. 13, Funktionsgenerator FG 607 Nr. 8, Netzteil Nr. 10, Versuchsplatine Nr. 10, Universalmeßgerät Nr. 11.

1.1 Versuche mit dem Oszilloskop

Wie in der Versuchsanleitung beschrieben wird, haben wir zunächst das Oszilloskop in die Grundeinstellung gebracht.

Einstellungen am Funktionsgenerator Sinussignal auf die Frequenz 1 kHz eingestellt.

Einstellungen am Oszilloskop Für ein gut ablesbares Bild haben sich am Oszilloskop folgende Einstellungen ergeben:

TIME/DIV Einstellung: 0.1 ms/DIV. VOLTS/DIV Einstellung für Ch I: 10 V/DIV.

Nun haben wir am Oszilloskop mit dem Drehknopf X-POS. die Kurve in horizontaler Richtung so verschoben, bis sich eine Nullstelle der Kurve im Koordinatenursprung befand. Am Oszilloskop war ein Bild nach Abbildung 1.1 sichtbar.

1.1.1 Frequenzbestimmung

Vom Oszilloskop ließ sich die Dauer einer halben Schwingung ablesen. Die Einheit dieses Wertes ist zunächst DIV. Mit Hilfe des oben angegebenen Umrechnungsfaktors läßt sich der Wert in Sekunden berechnen. Die Dauer einer Schwingung ist

$$T = 2 \cdot 4.7DIV \cdot 0.1ms/DIV = 0.94ms$$

und damit die Frequenz:

$$f = T^{-1} = \frac{1}{0.94ms} = 1.0638kHz$$

1.2 Ein Draht am Oszilloskop

Durch Variation der Eingangsempfindlichkeit und der Zeitbasis ergibt sich folgende Einstellung des Oszilloskops, bei der ein Signal sichtbar wird: Drehschalter TIME/DIV: 5 ms/DIV, Drehschalter VOLTS/DIV für Ch I: 50 mV/DIV. Eine Skizze des Signals ist in Abbildung 1.2 dargestellt.

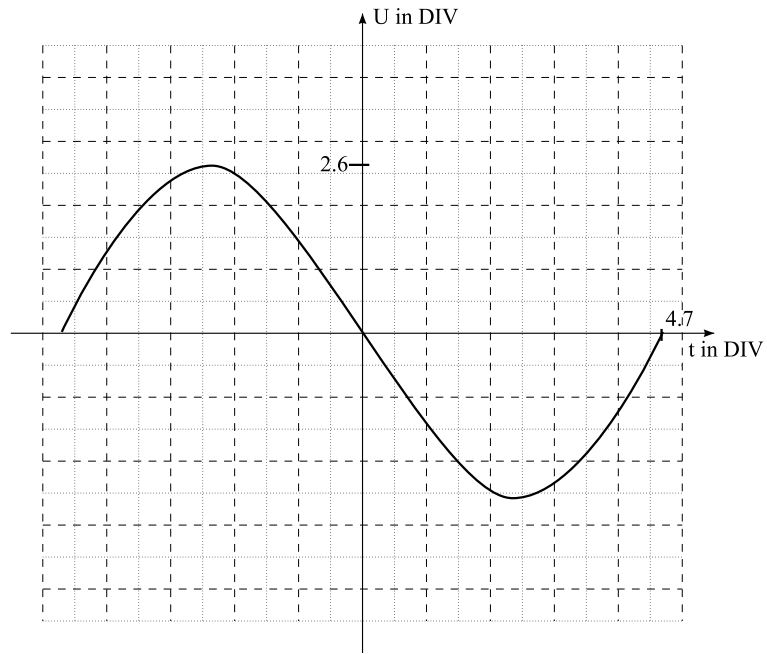


Abbildung 1.1: Sinusschwingung

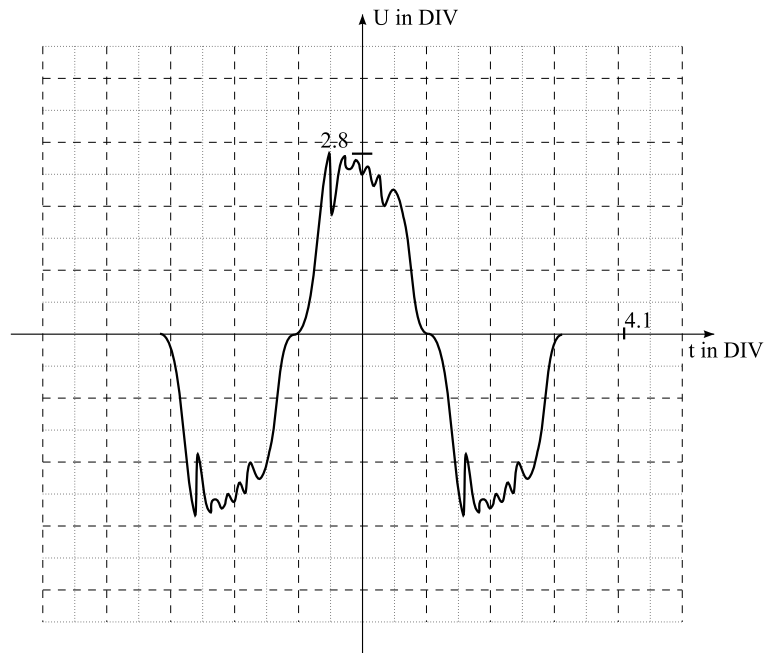


Abbildung 1.2: Oszilloskop bei Anschluß eines Drahtes

1.2.1 Frequenzbestimmung

Die Dauer einer Schwingung ist

$$T = 4.1DIV \cdot 5ms/DIV = 20.5ms$$

und daher die Frequenz

$$f = T^{-1} = 1/20.5ms = 48.78kHz$$

1.3 Tiefpaß

1.3.1 Vorbemerkung

Bei der angegebenen Schaltung handelt es sich um einen Tiefpaß, bei welchem folgende Gleichungen gelten:

$$\frac{U_{Bmax}}{U_{Amax}} = \frac{1}{\sqrt{1 + \omega^2 + R^2 + C^2}} \quad (1.1)$$

$$\tan \phi = -\omega \cdot RC \quad (1.2)$$

Diese Schaltung läßt tiefe Frequenzen praktisch ungedämpft hindurch, wohingegen hohe Frequenzen unterdrückt werden.

Versuchsaufbau Zuerst haben wir die Tastköpfe abgeglichen und dann den Versuch aufgebaut, wie es in der Praktikumsanleitung in Abbildung 1.6 dargestellt und im Text beschrieben ist. Um beide Signale miteinander vergleichen zu können, haben wir das Oszilloskop auf Zweikanalbetrieb umgestellt.

Am Punkt A (Channel I) messen wir den zeitlichen Verlauf der durch den Frequenzgenerator erzeugten Spannung, am Punkt B (Channel II) hingegen die Spannung, die über dem Kondensator abfällt. Die Farbcodierung des Widerstandes (braun-grün-rot-gold) weist diesen als 1,5 kW aus, was durch die Messung mit dem zuvor geeichten Universalmeßgerät (Einstellung: Rx1K) bestätigt wird.

1.3.2 Grenzfrequenz der Schaltung

Unter der Grenzfrequenz versteht man, die Frequenz, bei der das Verhältnis der maximal erreichten Eingangsspannung von Channel I zur maximalen Eingangsspannung von Channel II gleich $\sqrt{2}$ ist:

$$\frac{U_{Bmax}}{U_{Amax}} = \sqrt{2}$$

Aus Gleichung 1.1 erkennt man leicht, daß hierfür

$$\omega = \frac{1}{RC}$$

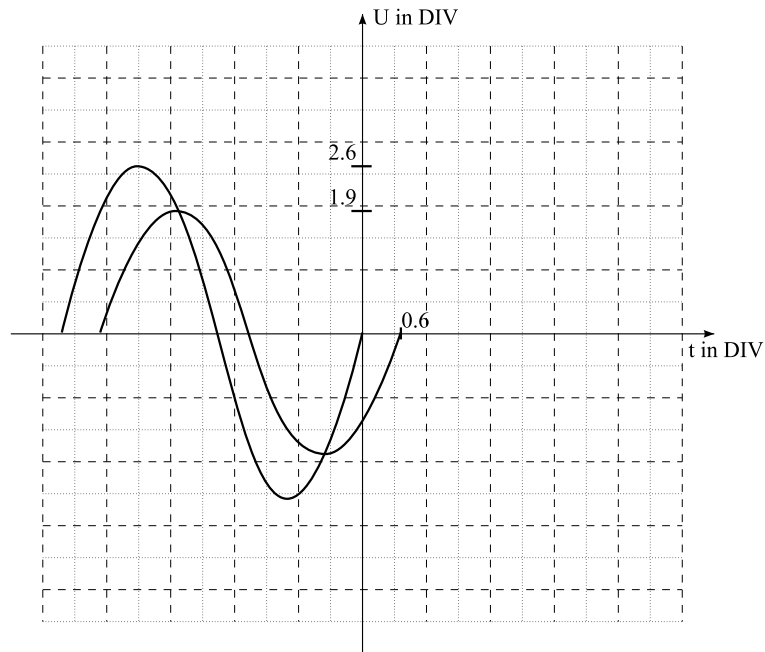


Abbildung 1.3: Eingangs- und Ausgangsspannung bei 1kHz Sinusspannung

gelten muß. Somit ergibt sich die Grenzfrequenz wie folgt:

$$\begin{aligned}\omega &= \frac{1}{1.5k\Omega \cdot 100nF} = 6666Hz \\ f &= \frac{\omega}{2 \cdot \pi} = \frac{6666Hz}{6.2823} \\ &= 1061Hz = 1.061kHz\end{aligned}$$

1.3.3 Verhältnis von Eingangs- und Ausgangsspannung bei Sinusspannung

Der Funktionsgenerator ist auf Sinus-Spannung eingestellt. Dreht man den Regler des Funktionsgenerators über den gesamten Frequenzbereich von kleineren zu größeren Frequenzen, beobachtet man, daß sich bei der Channel I Kurve der maximal erreichte Wert nicht ändert, während der Maximalwert der Channel II Kurve kontinuierlich kleiner wird.

Für die Skizze des Oszilloskopbildes (siehe Abbildung 1.3) bei einer Frequenz von 1kHz wählten wir folgende Einstellungen am Oszilloskop: Drehschalter TIME/DIV: 0.2 ms/DIV, Drehschalter VOLTS/DIV für Ch I: 0.2 V/DIV, Drehschalter VOLTS/DIV für Ch II: 0.2 V/DIV.

Spannungsverhältnis U_B/U_A bei 1 kHz:

$$U_A = 2.6DIV \cdot 0.2V/DIV = 0.52V$$

$$U_B = 1.9DIV \cdot 0.2V/DIV = 0.38V$$

$$\frac{U_B}{U_A} = 0.7307$$

bei 10 kHz:

$$U_A = 2.7DIV \cdot 0.2V/DIV = 0.54V$$

$$U_B = 0.4DIV \cdot 0.2V/DIV = 0.08V$$

$$\frac{U_B}{U_A} = 0.148$$

Phasenverschiebung Unter Phasenverschiebung versteht man die zeitliche Verschiebung zwischen Wechselgrößen. Da die Frequenzen der beiden Schwingungen in unserem Fall identisch sind, bleibt die Phasenverschiebung konstant, andernfalls ist sie eine zeitabhängige Funktion. Kennt man die Zeitdifferenz Δt (= Abstand der Schnitstellen der Kurven mit dem time-Achse) und die Schwingungsdauer T , welche für beide Schwingungen gelten muß, ergibt sich die Phasenverschiebung ϕ durch

$$\phi = \frac{2\pi}{T} \cdot \Delta t$$

beziehungsweise

$$\phi = \frac{360^\circ}{T} \cdot \Delta t$$

bei 1 kHz:

$$\Delta t = 0.6DIV \cdot 0.2ms/DIV = 0.12ms$$

$$T = 0.94ms$$

$$\phi = 360^\circ \cdot \frac{0.12ms}{0.94ms} = 45.96^\circ$$

bei 10 kHz benötigen wir noch eine Veränderung am Oszilloskop: Drehschalter TIME/DIV: $50\mu s/DIV$, Drehschalter VOLTS/DIV für Ch I: $0.2 V/DIV$, Drehschalter VOLTS/DIV für Ch II: $0.2 V/DIV$.

$$\Delta t = 0.6DIV \cdot 50\mu s/DIV = 30\mu s$$

$$T = 1.9DIV \cdot 50\mu s/DIV = 95\mu s$$

$$\phi = 360^\circ \cdot \frac{30\mu s}{95\mu s} = 113.68^\circ$$

1.3.4 Verhältnis von Eingangs- und Ausgangsspannung bei Rechteckspannung

Da eine Rechteckspannung aus der Überlagerung unzähliger sinusförmiger Wechselspannungen mit verschiedenen Frequenzen entsteht (Fourier-Entwicklung), sind theoretische Berechnungen äußerst aufwendig.

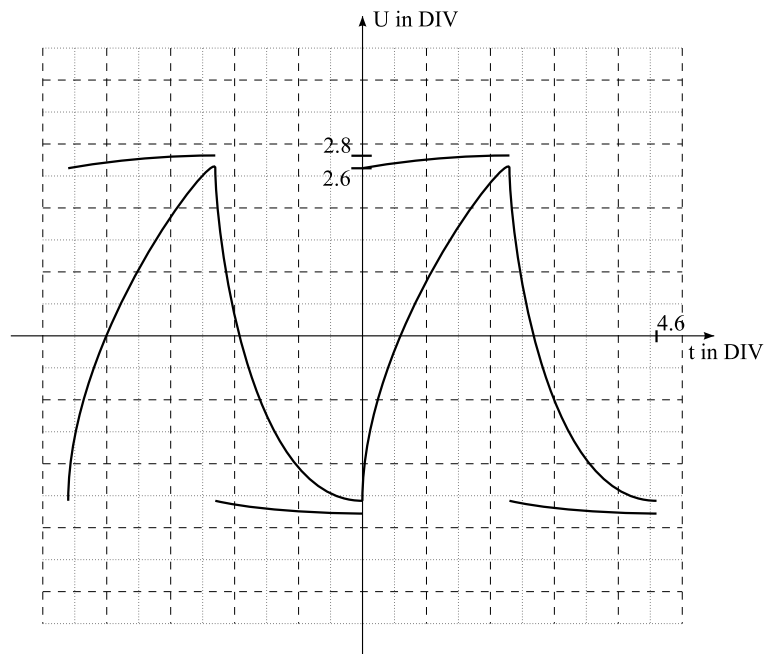


Abbildung 1.4: Eingangs- und Ausgangsspannung bei 1kHz Rechteckspannung

Der maximale Ausschlag der Channel II Kurve ist umso größer, je kleiner die angelegte Frequenz ist und steigt für sehr kleine Frequenzen auf den Wert, den die Channel I Kurve markiert (Sättigungsspannung). Desweiteren ist zu beobachten, daß die Steigung der Channel II Kurve zu Beginn einer Schwingung sehr groß ist und bis $T/2$ kontinuierlich abnimmt.

Für die Skizze des Oszilloskopbildes (siehe Abbildung 1.4) bei einer Frequenz von 1kHz wählten wir folgende Einstellungen am Oszilloskop: Drehschalter TIME/DIV: 0.2 ms/DIV, Drehschalter VOLTS/DIV für Ch I: 0.2 V/DIV, Drehschalter VOLTS/DIV für Ch II: 0.2 V/DIV.

Spannungsverhältnis U_B/U_A bei 1 kHz:

$$U_A = 2.8 \text{ DIV} \cdot 0.2 \text{ V/DIV} = 0.56 \text{ V}$$

$$U_B = 2.6 \text{ DIV} \cdot 0.2 \text{ V/DIV} = 0.52 \text{ V}$$

$$\frac{U_B}{U_A} = 0.928$$

bei 10 kHz:

$$U_A = 2.8 \text{ DIV} \cdot 0.6 \text{ V/DIV} = 0.56 \text{ V}$$

$$U_B = 0.6 \text{ DIV} \cdot 0.2 \text{ V/DIV} = 0.12 \text{ V}$$

$$\frac{U_B}{U_A} = 0.214$$

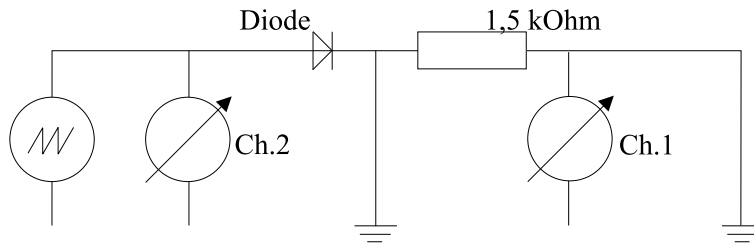


Abbildung 1.5: Schaltung zur Messung der Kennlinien

Phasenverschiebung Bei der Rechteckspannung ist keine Phasenverschiebung meßbar, da sie hier nicht definiert ist.

1.4 Kennlinie einer Diode

1.4.1 Versuchsaufbau

Zum Aufbau der Schaltung siehe Abbildung 1.5. Der Funktionsgenerator erzeugt eine Dreiecksspannung. Diese wird im Oszilloskop auf die x-Achse abgetragen. Sie liefert die ansteigende Spannung, von welcher abhängt, ob die Diode leitet oder nicht.

1.4.2 Durchlaßspannung

Die Durchlaßspannung ist die Spannung, ab welcher die Diode leitet. Man bestimmt sie aus der Diodenkennlinie, indem man die linear ansteigende Gerade nach unten verlängert und den Wert am U-Achsen-Schnittpunkt abliest.

Einstellungen des Oszilloskops Drehschalter VOLTS/DIV für Ch I: 1 V/DIV, Drehschalter VOLTS/DIV für Ch II: 0.1 V/DIV.

Abgelesene Durchlaßspannung	Die Skizzen der Kennlinien sind in Abbildung 1.6 dargestellt.	
	Diode	Durchlaßspannung U_D
dargestellt.	Universaldiode 1N4148	$0.5DIV \cdot 1V/DIV = 0.5V$
	Grüne Leuchtdiode	$1.4DIV \cdot 1V/DIV = 1.4V$
	Rote Leuchtdiode	$1.9DIV \cdot 1V/DIV = 1.9V$

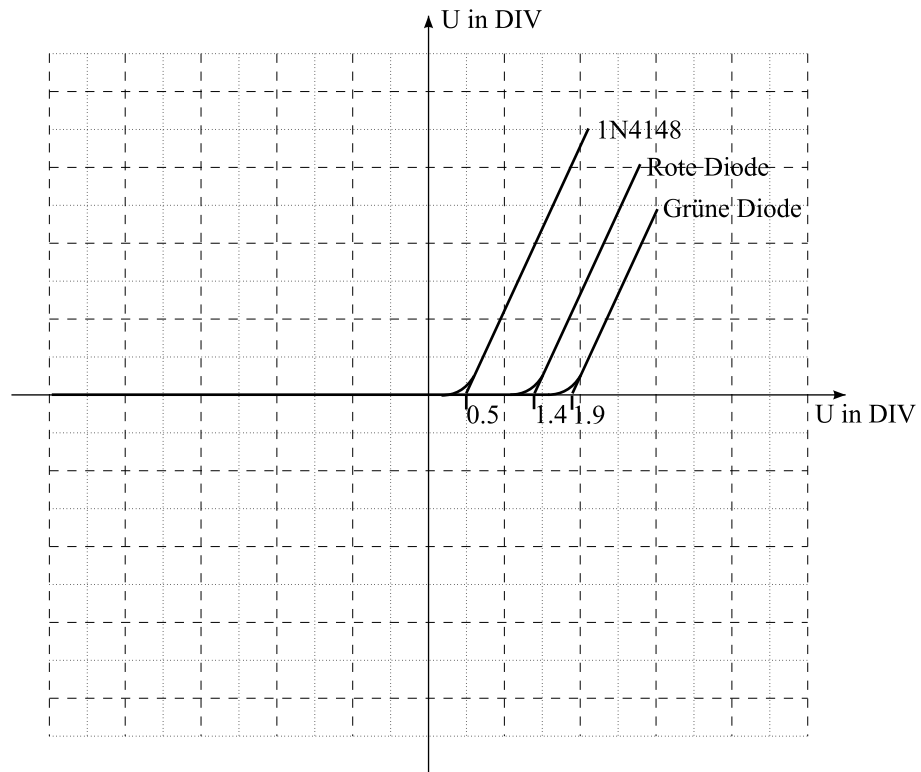


Abbildung 1.6: Kennlinien der 3 Dioden

Versuch 2: Transistorschaltungen

Gruppe David Frese, Christos Freris.

Protokollführer David Frese

Verwendete Geräte Oszilloskop HM604 Nr. 4, Funktionsgenerator FG 607 Nr. 6, Netzteil Nr. 7, Versuchsplatine Nr. 8, Universalmeßgerät Nr. 5.

2.1 Messung der Kennlinie des n-Kanal MOS-FET BS107A

2.1.1 Schaltung

Zur Messung des Drain-Stromes I_D in Abhängigkeit von der Gate-Source-Spannung U_{GS} benutzen wir eine Schaltung nach Abbildung 2.1.

Channel 2 wird dabei auf die X-Achse abgetragen. Auf *Channel 1* messen wir zwar die Spannung die über dem Widerstand abfällt, aber diese ist direkt proportional zu I_{DS} :

$$I_{DS} = \frac{U_R}{R}$$

Als U_{DS} wählen wir einfach die Platinenspannung $5V$.

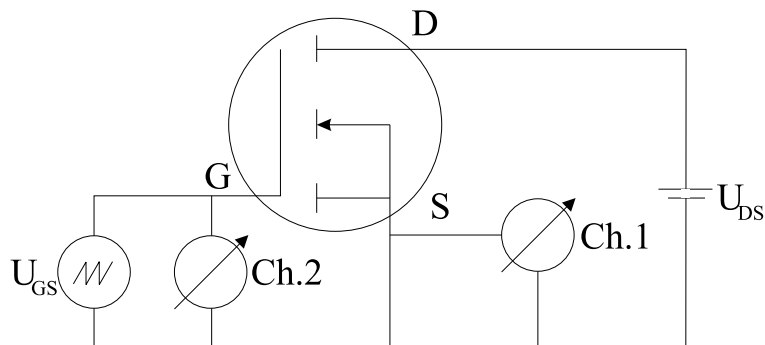


Abbildung 2.1: Schaltung zur Aufnahme der Kennlinie

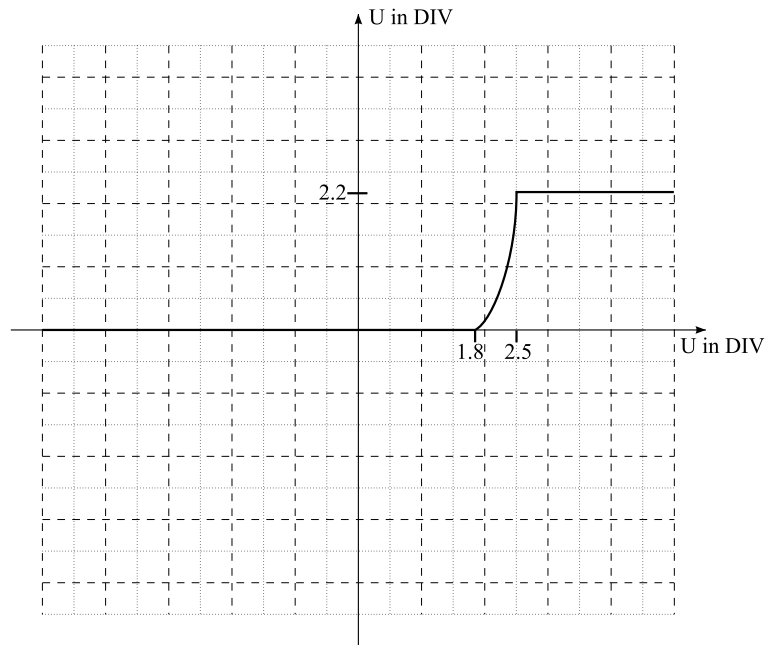


Abbildung 2.2: Verlauf der Kennlinie des MOS-FET BS107A

2.1.2 Kennlinie

Mit den Einstellungen 0.2 VOLTS/DIV für Channel 1 und 0.1 VOLTS/DIV für Channel 2 konnte man die Kennlinie des Transistors gut erkennen. Die Kennlinie ist in Abbildung 2.2 dargestellt. Sie verläuft bis 0.18 Volt auf 0 Volt. Zwischen 0.18 Volt und 0.25 Volt steigt sie dann schnell bis auf 0.44 Volt an und verläuft dann wieder flach auf gleichem Niveau weiter.

2.2 Stromrückgekoppelte Emitterschaltung als Verstärker

Aus der Aufgabenstellung wissen wir folgende Werte

$$\begin{aligned}
 U_V &= 5V \\
 I_C &= 500 \cdot I_B \\
 U_{BE} &= 0.7V
 \end{aligned}
 \tag{2.1}$$

Für ausgeschaltete Eingangsspannung $U_E = 0$ berechnen wir erstmal die Widerstände. Dabei gilt nach den Kirchhoffschen Gesetzen:

$$\begin{aligned}
 I_V &= I_1 + I_{R_C} = I_1 + I_C \\
 I_1 &= I_B + I_2 \\
 U_V &= U_{R_C} + U_{C_E} + U_{R_E} = R_C \cdot I_C + U_{C_E} + R_E \cdot I_E
 \end{aligned}$$

$$U_R = U_{BE} + UR_E = U_{BE} + I_E \cdot R_E$$

$$U_V = U_1 + U_2$$

Um die Widerstände berechnen zu können, muß man einige vereinfachende Annahmen machen. Aus dem Buch "Grundlagen der Elektronik" von Albrecht Rost haben wir den Hinweis, daß I_1 am Besten das 5 bis 10-fache von I_B betragen sollte. Desweiteren nehmen wir die Spannung die über der Kollektor-Emitterstrecke abfällt als 0 V an. Damit lassen sich Formeln für die Ströme in den Widerständen aufstellen:

$$I_1 = 7.5 \cdot I_B$$

$$I_V = 7.5 \cdot I_B + 500 \cdot I_B$$

$$I_B = \frac{I_V}{507.5} = \frac{40mA}{507.5} = 0.079mA$$

$$I_1 = 0.6mA$$

$$I_2 = 6.5 \cdot I_B = 0.5mA$$

$$I_C = I_V - I_1 = 40mA - 0.6mA = 39.4mA$$

$$I_E = I_B + I_C = 39.5mA$$

Und für die Widerstände gilt dann

$$R_E = \frac{0.15 \cdot U_V}{I_E} = 19\Omega$$

$$R_2 = \frac{U_{BE} + I_E \cdot R_E}{I_2} = 2.9k\Omega$$

$$R_1 = \frac{U_V - I_2 R_2}{I_1} = 5.9k\Omega$$

$$R_C = \frac{U_V - R_E I_E}{I_C} = \frac{0.85 \cdot U_V}{39.4mA} = 108\Omega$$

Da unsere Herleitung nicht auffindbare Fehler enthält, gab uns einer der Betreuer die korrekten Ergebnisse, mit Hilfe derer wir die Schaltung aufbauen konnten:

$$R_1 = 5.6k\Omega$$

$$R_2 = 15k\Omega$$

$$R_E = 6\Omega$$

$$R_C = 56\Omega$$

$$C_b = 4.7\mu F$$

$$C_L = 22\mu F$$

2.2.1 Verstärkung

Die Verstärkung lässt sich mit dem Quotienten

$$V = \frac{\text{Hochpunkt} - \text{Tiefpunkt von } U_A}{\text{Hochpunkt} - \text{Tiefpunkt von } U_E}$$

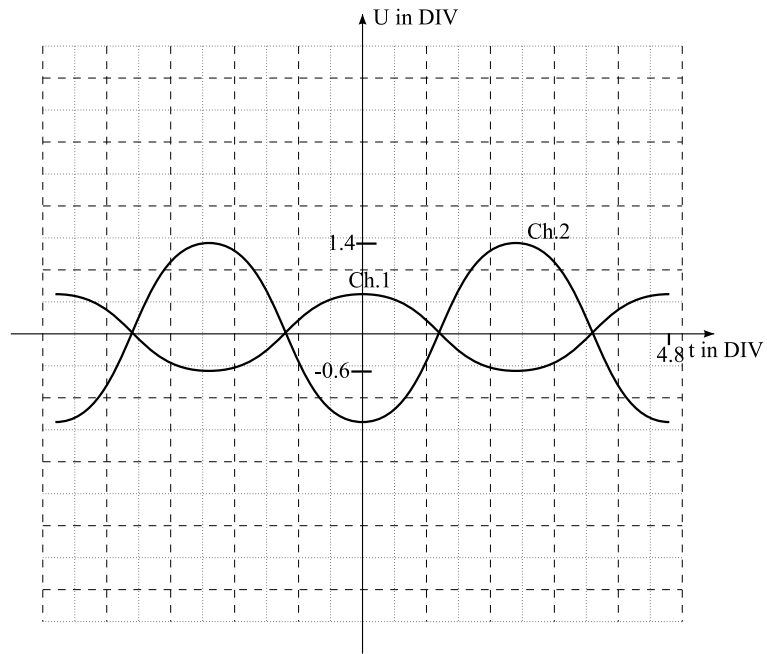


Abbildung 2.3: Skizze des Oszilloskops mit Eingangs- und Ausgangsspannung

bestimmen. Die Werte kann man vom Oszilloskop ablesen (siehe Abbildung 2.3) und es ergibt sich:

$$V = \frac{1.4V - (-1.4V)}{0.6V - (-0.6V)} = 2.33$$

2.2.2 Verzerrungen

Eine Quelle für Verzerrungen ist die Übersteuerung. Das heißt daß die Amplitude des Eingangssignals so hoch ist, daß die Strommenge die für die Verstärkung dieses Signals nötig ist, von der Platine nicht mehr geliefert werden kann.

Versuch 3: Digitale Elektronik

Gruppe David Frese, Christos Freris.

Protokollführer Christos Freris

Verwendete Geräte Oszilloskop HM604 Nr. 4, Funktionsgenerator FG 607 Nr. 6, Netzteil Nr. 7, Versuchsplatine Nr. 8, Universalmeßgerät Nr. 5, Digitales Experimentierboard Nr. 6.

3.1 Inverter und NOR-Gatter

3.1.1 Kennlinie eines Inverters

Wir schlossen das Oszilloskop gemäß Abbildung 3.1 auf und legten *Channel 2* auf die X-Achse des Oszilloskops (Schalter X-Y). An U_V legten wir die Platinenspannung 5V an. Mit den Einstellungen

Channel 1	0.2 VOLTS/DIV
Channel 2	0.2 VOLTS/DIV
Time	0.2 ms/DIV

ergab sich am Oszilloskop ein Bild nach Abbildung 3.2.

3.1.2 Dynamisches RAM

Die Schaltung für ein dynamischen RAM von 1 Bit ist in Abbildung 3.3 noch einmal verkürzt dargestellt. Beim Verbinden des Eingangs mit Masse wird die '0' invertiert und man beobachtet einen '1' Pegel am Ausgang. Dieser bleibt bestehen, wenn man den Draht entfernt. Das selbe, nur umgekehrt beobachtet man, wenn man den Eingang an 5 V anschließt und den Draht wieder entfernt. Somit entsteht ein dynamisches RAM von 1 Bit, das nach 2^{29} Wiederholungen zu einem mächtigen Speicher wird!

3.1.3 NOR-Gatter

In Abbildung 3.4 ist das Schaltbild des NOR-Gatters dargestellt. Zum Überprüfen der Korrektheit legten wir den Ausgang des Gatters auf eine der Leuchtdioden und probierten alle möglichen Eingangssignale aus.

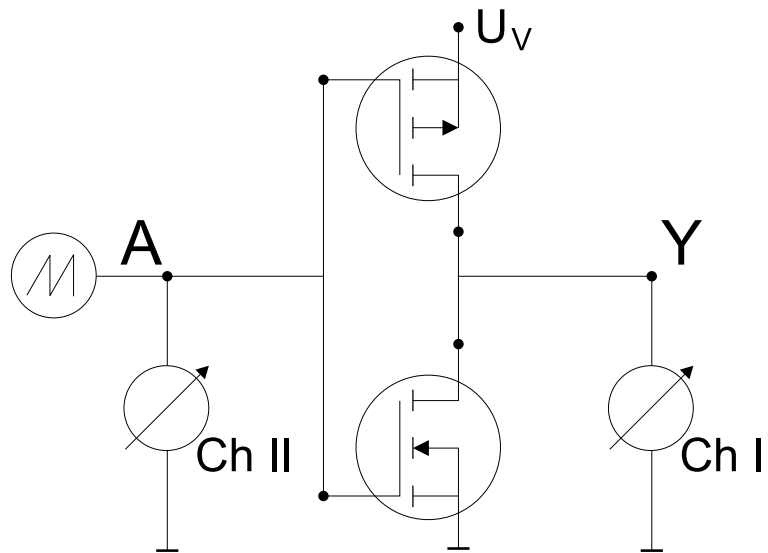


Abbildung 3.1: Schaltung zur Aufnahme der Kennlinie eines Inverters

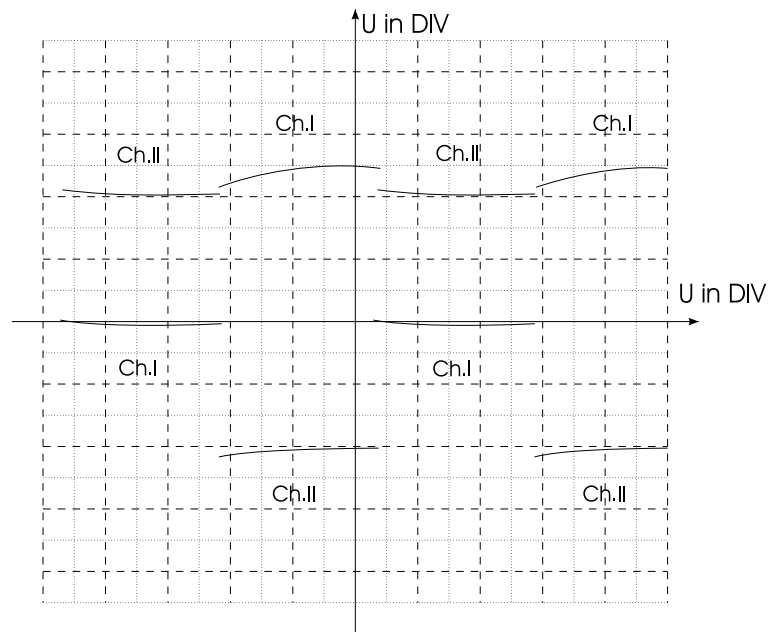


Abbildung 3.2: Kennlinie des Inverters

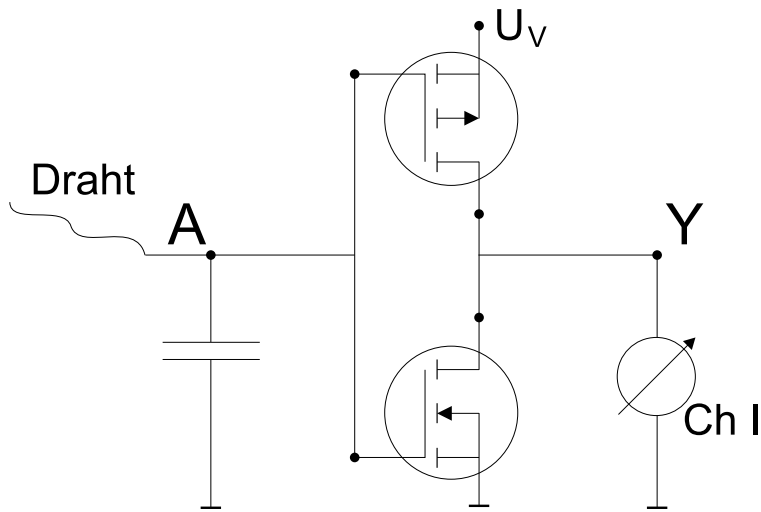


Abbildung 3.3: Dynamisches 1-Bit RAM

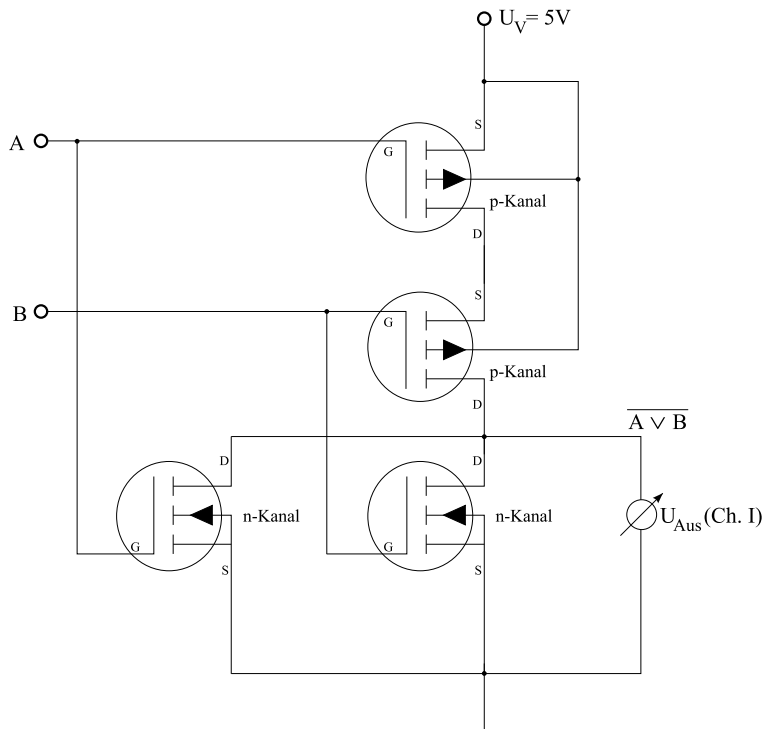


Abbildung 3.4: CMOS NOR-Gatter

Dadurch ergab sich folgende Funktionstabelle:

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

3.2 Aufbau eines XOR-Gatters ...

3.2.1 ... mit Hilfe von NAND-Gattern

Die Definition von XOR nach der Booleschen Algebra ist:

$$A \text{ xor } B := (A \wedge \overline{B}) \vee (\overline{A} \wedge B)$$

Mit Hilfe der De Morgan'schen Gesetze, kann dieser logischer Ausdruck wie folgt umgeformt werden:

$$\begin{aligned} A \text{ xor } B &= (A \wedge \overline{B}) \vee (\overline{A} \wedge B) \\ &= \overline{\overline{(A \wedge \overline{B}) \vee (\overline{A} \wedge B)}} \\ &= \overline{(A \wedge \overline{B}) \wedge (\overline{A} \wedge B)} \\ &= \overline{(A \wedge (\overline{B} \wedge \overline{B})) \wedge ((\overline{A} \wedge \overline{A}) \wedge B)} \end{aligned}$$

Anhand dieses Ausdrucks, können wir mit 5 NAND-Gattern ein XOR-Gatter aufbauen, und zwar so wie in Abbildung 3.5 dargestellt. Die Schaltung wurde von dem Betreuer begutachtet und akzeptiert.

3.2.2 ... mit Hilfe von AND-, OR- und NOT-Gattern

In der Definition von XOR liegt auch die Antwort dieser Aufgabe:

$$\begin{aligned} A \text{ xor } B &:= (A \wedge \overline{B}) \vee (\overline{A} \wedge B) \\ &= \underbrace{(A \vee \overline{A})}_{=1} \wedge (A \vee B) \wedge (\overline{B} \vee \overline{A}) \wedge \underbrace{(\overline{B} \vee B)}_{=1} \\ &= (A \vee B) \wedge \overline{(B \wedge A)} \end{aligned}$$

Anhand dieses Ausdrucks, können wir mit 1 OR-Gatter, 3 AND-Gattern und 1 NOT-Gatter ein XOR-Gatter ohne Hazards aufbauen, und zwar so wie in Abbildung 3.6 dargestellt. Die Schaltung wurde von dem Betreuer begutachtet und akzeptiert.

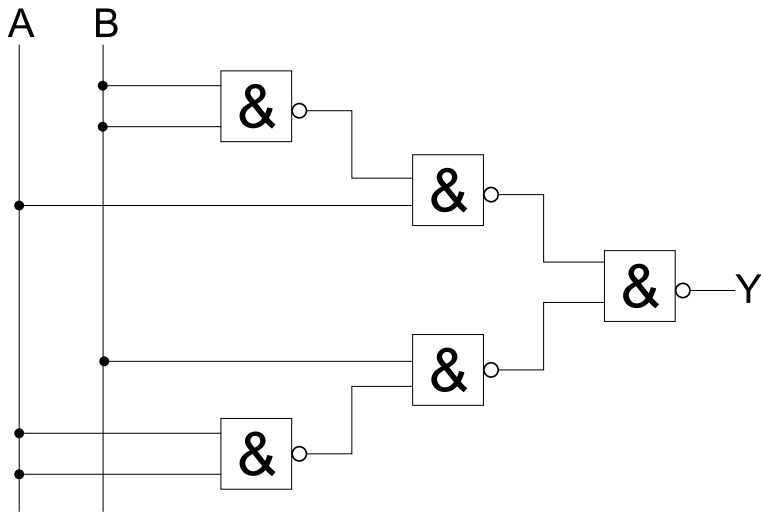


Abbildung 3.5: XOR-Gatter mit NAND

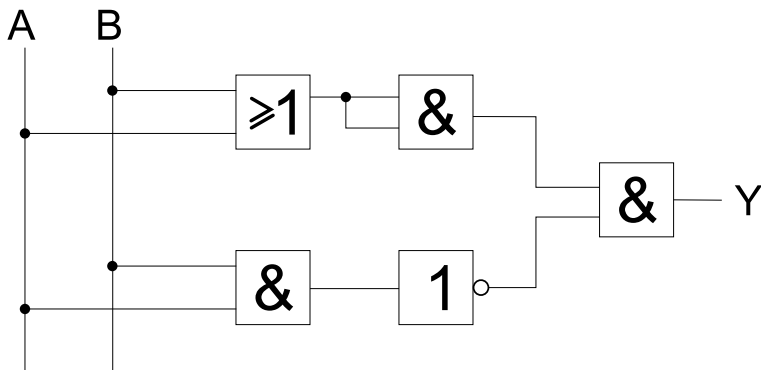


Abbildung 3.6: XOR-Gatter mit OR, AND, NOT

3.3 Hintereinanderschaltung drei Invertoren

3.3.1 Aufbau der Schaltung

Nachdem wir die Schaltung, gemäß der Abbildung des Scripts, auf dem digitalen Experimentierboard aufgebaut haben, wurde sie von dem Betreuer begutachtet und akzeptiert.

3.3.2 Ausgangssignal und dessen Frequenz

Mit folgenden Einstellungen

Channel 1	0.1 VOLTS/DIV
Channel 2	0.2 VOLTS/DIV
Time	0.05 μ S/DIV

ergab sich am Oszilloskop ein Bild nach Abbildung 3.7. Die Dauer einer Schwingung ist

$$T = 2 \cdot 0.75 \text{DIV} \cdot 0.05 \mu\text{s}/\text{DIV} = 0.0375 \mu\text{s}$$

und damit die Frequenz:

$$f = T^{-1} = \frac{1}{0.0375 \mu\text{s}} = 26.6666 \text{MHz}$$

Erklärung des Phänomens Durch die Reihenschaltung dreier Inverter wird das Eingangssignal einmal invertiert am Ausgang ausgegeben. Außerdem gibt es eine bestimmte Zeitverzögerung zwischen einer Eingangssignal- und der zugehörigen Ausgangssignaländerung. Koppelt man nun den Ausgang auf den Eingang zurück, so geschieht Folgendes: Zunächst liegt am Eingang L(ow) an. Nach der oben beschriebenen kurzen Zeitverzögerung liegt am Ausgang dann H(igh) an, was auf den Eingang zurückgekoppelt wird. Dieses Signal ergibt wiederum nach kurzer Zeitverzögerung am Ausgang den Wert L(ow). Somit oszilliert die Schaltung.

3.4 Hazards

3.4.1 Aufbau der Schaltung

3.4.2 Eingangs- und Ausgangssignal von 5 V

Mit folgenden Einstellungen

Channel 1	0.2 VOLTS/DIV
Channel 2	0.2 VOLTS/DIV
Time	0.1 μ s/DIV

ergab sich am Oszilloskop ein Bild nach Abbildung 3.8.

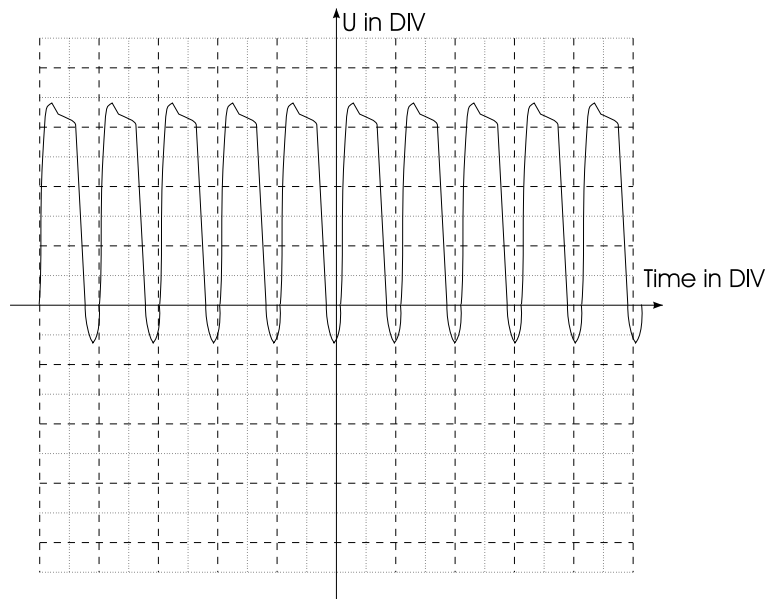


Abbildung 3.7: Schwingung durch 3 rückgekoppelte Inverter

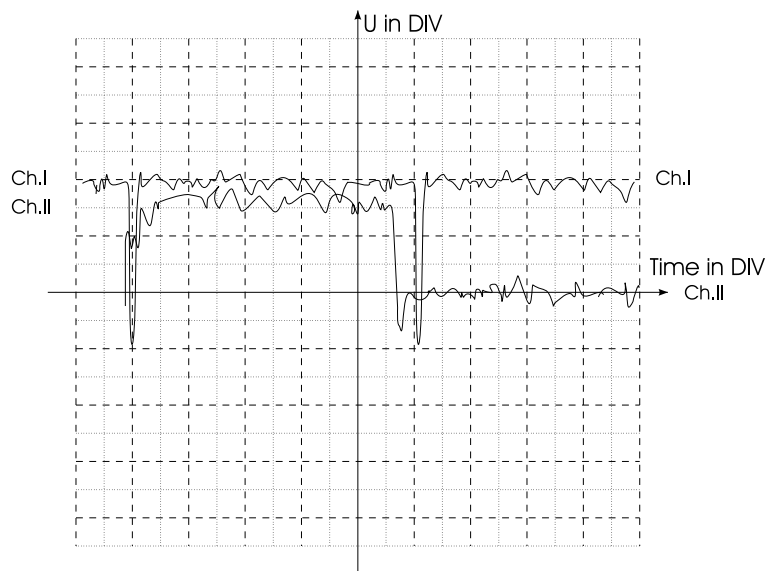


Abbildung 3.8: Hazardeffekt

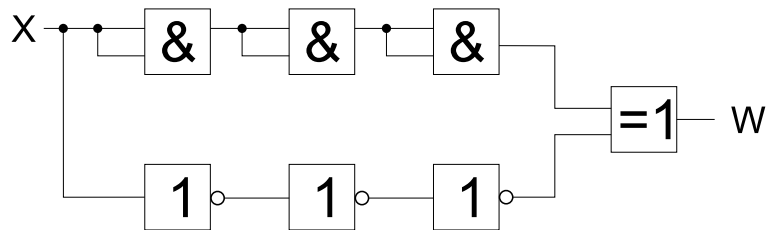


Abbildung 3.9: Schaltung zur Eliminierung des Hazards

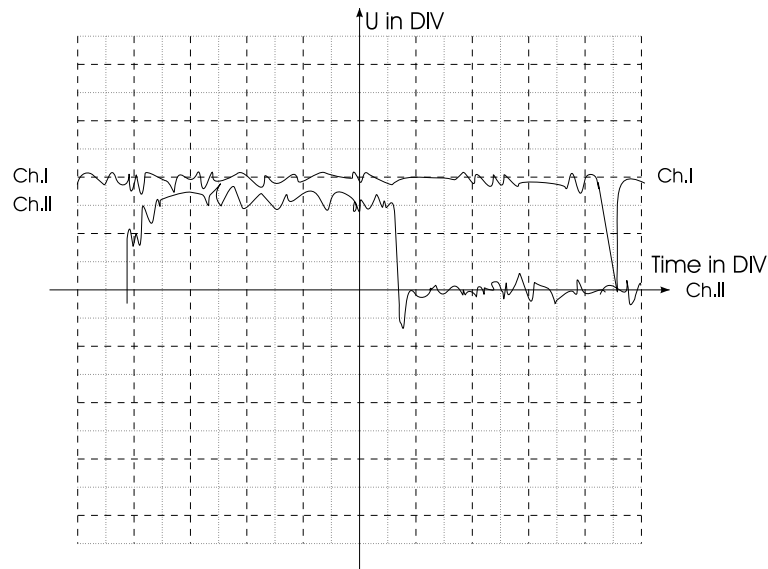


Abbildung 3.10: Eliminierung des Hazardeffekts

3.4.3 Erklärung des Hazards

Die Laufzeit des Signals durch die 3 Inverter ist größer als der direkte Weg zum Eingang des XOR-Gatters. Durch den schnellen Wechsel des Eingangssignals X kommt am Ausgang W ein gestörtes Signal an.

3.4.4 Eliminierung des Hazards

Eine Standard-Methode um Hazards zu eliminieren ist es, in die zu schnelle Leitung, einfach AND-Gatter mit einzubauen. Dabei legt man das selbe Signal auf beide Eingänge, und verändert damit das Signal nicht ($x \wedge x = x$). Eine entsprechende Schaltung ist in Abbildung 3.9 dargestellt, dessen Protokollierung zeigt die Abbildung 3.10.

Versuch 4: Digitaler Schaltungsentwurf

Gruppe David Frese, Christos Freris.

Protokollführer David Frese

Verwendete Geräte Netzteil Nr. 7, Digitales Experimentierboard Nr. 1.

4.1 Entprellter Taster

Zur Entprellerschaltung benutzten wir einen Schalter mit zwei Ausgängen. Aus zwei NAND-Gatter stellen wir, nach Abbildung 4.1 aus dem Praktikumsskript, ein RS-Flipflop zusammen. Dazu verwenden wir die Gatter im Chip 00.

Kabelverbindungen Zur eindeutigen Identifizierung der Anschlüsse buchstabieren wir die Chips auf dem Experimentierboard mit griechischen Buchstaben durch. Die einzelnen Anschlüsse an die Gatter sind ja bereits durchnummeriert.

P/3 ↔ P/4
P/2 ↔ P/6
On-Eingang ↔ P/1
Off-Eingang ↔ P/5
P/6 ↔ On-Ausgang

Kontrolle Wir haben die Funktionsfähigkeit überprüft.

Verwendete Gatter 2 × NAND2.

4.2 Asynchroner 4-Bit BCD-Zähler

Als Ausgangspunkt für den Zähler betrachtet man ersteinmal einen Zähler für alle 16 Zahlen. Dazu muß man einfach nur den Q -Ausgang eines Flipflops jeweils auf den CLK -Eingang des Nächsten schalten. Nun wollen wir aber beim 10. Taktimpuls am Eingang, ein anderes Verhalten der Flipflops bewirken.

Beim 10. Taktimpuls würde an den Ausgängen die Zahl 1010 anliegen. Da es aber 0000

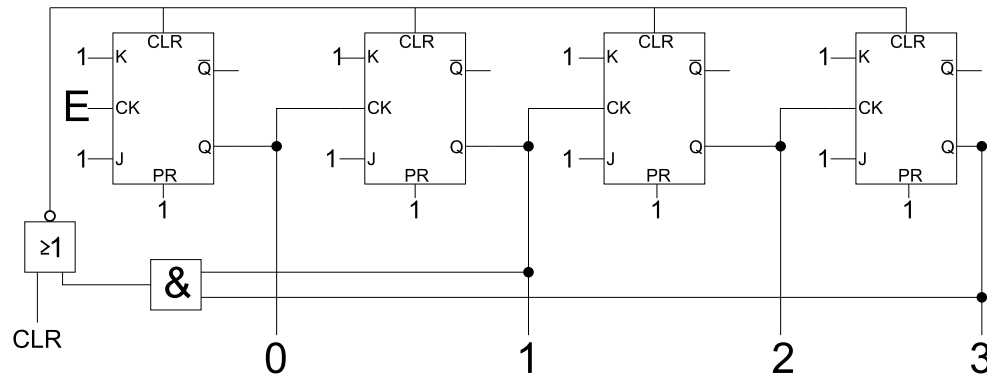


Abbildung 4.1: Schaltung des 4-Bit BCD-Zählers

sein sollte, macht man einfach einen Clear wenn Bit-3 und Bit-1 beide 1 sind. Das liefert uns das gewünschte Ergebnis, weil bei den andern Werten niemals Bit-3 und Bit-1 gleichzeitig 1 sind.

Verwendete Gatter 4 × JK-Flipflop, 1 × AND2, 1 × NOR2.

4.2.1 Ohne Entpreller

Ohne Entprellerschaltung ergeben sich am Eingang des Zählers ungewollte Spannungswankungen. Diese führen dazu, daß der Zähler pro Tastendruck mehrere Zahlen gleichzeitig (in sehr kurzer Zeit) hochzählt.

4.3 7-Segment-Anzeige

Die Vektorfunktion die für die 7-Segmentanzeige benötigt wird

$$(A, B, C, D) \mapsto (a, b, c, d, e, f, g)$$

ist in Tabelle 4.1 dargestellt. Daraus kann man die KV-Diagramme in Abbildung 4.2 bis Abbildung 4.5 aufstellen. Aus den KV-Diagrammen ergeben sich dann die Einzel-funktionen

$$\begin{aligned} a &= (\bar{A} \bar{B} C) + (A \bar{B} \bar{C} \bar{D}) \\ b &= (\bar{A} B C) + (A \bar{B} C) \\ c &= \bar{A} B \bar{C} \\ d &= (\bar{A} \bar{B} C) + (A B C) + (A \bar{B} \bar{C} \bar{D}) \\ e &= A + (\bar{A} \bar{B} C) \\ f &= (A \bar{C} \bar{D}) + (\bar{A} B \bar{C}) + (A B) \\ g &= (\bar{B} \bar{C} \bar{D}) + (A B C) \end{aligned}$$

	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

Tabelle 4.1: Vektorfunktion für die 7-Segmentanzeige

BA \ DC	00	01	11	10
00		1		
01	1			
11	-	-	-	-
10			-	-

BA \ DC	00	01	11	10
00				
01		1		1
11	-	-	-	-
10			-	-

Abbildung 4.2: KV-Digramm für a und b

BA \ DC	00	01	11	10
00				1
01				
11	-	-	-	-
10			-	-

BA \ DC	00	01	11	10
00		1		
01	1		1	
11	-	-	-	-
10			-	-

Abbildung 4.3: KV-Digramm für c und d

BA \ DC	00	01	11	10
00		1	1	
01	1	1	1	
11	-	-	-	-
10		1	-	-

BA \ DC	00	01	11	10
00		1	1	1
01			1	
11	-	-	-	-
10			-	-

Abbildung 4.4: KV-Digramm für e und f

BA \ DC	00	01	11	10
00	1	1		
01			1	
11	-	-	-	-
10			-	-

Abbildung 4.5: KV-Digramm für g

Und aus den Funktionen wiederum die Schaltung. Diese ist in Abbildung 4.6 abgebildet.

Kabelverbindungen Die invertierten Eingänge greifen wir dabei über die entsprechenden \overline{Q} -Ausgängen der Flipflops ab, anstatt vier Inverter zu benutzen - das spart einige Kabel ein.

\overline{A}	\leftrightarrow	$\Lambda/13$
\overline{B}	\leftrightarrow	$\Lambda/1$
C	\leftrightarrow	$\Lambda/2$
A	\leftrightarrow	$\Lambda/9$
B	\leftrightarrow	$\Lambda/10$
C	\leftrightarrow	$\Lambda/11$
A	\leftrightarrow	$\Lambda/3$
\overline{C}	\leftrightarrow	$\Lambda/4$
\overline{D}	\leftrightarrow	$\Lambda/5$
$\Lambda/6$	\leftrightarrow	$\Pi/1$
\overline{B}	\leftrightarrow	$\Pi/2$
\overline{A}	\leftrightarrow	$\Pi/4$
B	\leftrightarrow	$\Pi/5$
$\Pi/6$	\leftrightarrow	$\Pi/9$
C	\leftrightarrow	$\Pi/10$
A	\leftrightarrow	$\Pi/12$
\overline{B}	\leftrightarrow	$\Pi/13$
$\Pi/11$	\leftrightarrow	$\Gamma/1$
C	\leftrightarrow	$\Gamma/2$
$\Pi/6$	\leftrightarrow	$\Gamma/4$
\overline{C}	\leftrightarrow	$\Gamma/5$
\overline{C}	\leftrightarrow	$M/9$
\overline{D}	\leftrightarrow	$M/10$
\overline{B}	\leftrightarrow	$M/11$
$M/8$	\leftrightarrow	$N/13$
A	\leftrightarrow	$\Gamma/9$
B	\leftrightarrow	$\Gamma/10$

$\Lambda/12$	\leftrightarrow	$\Sigma/1$
$\Pi/3$	\leftrightarrow	$\Sigma/2$

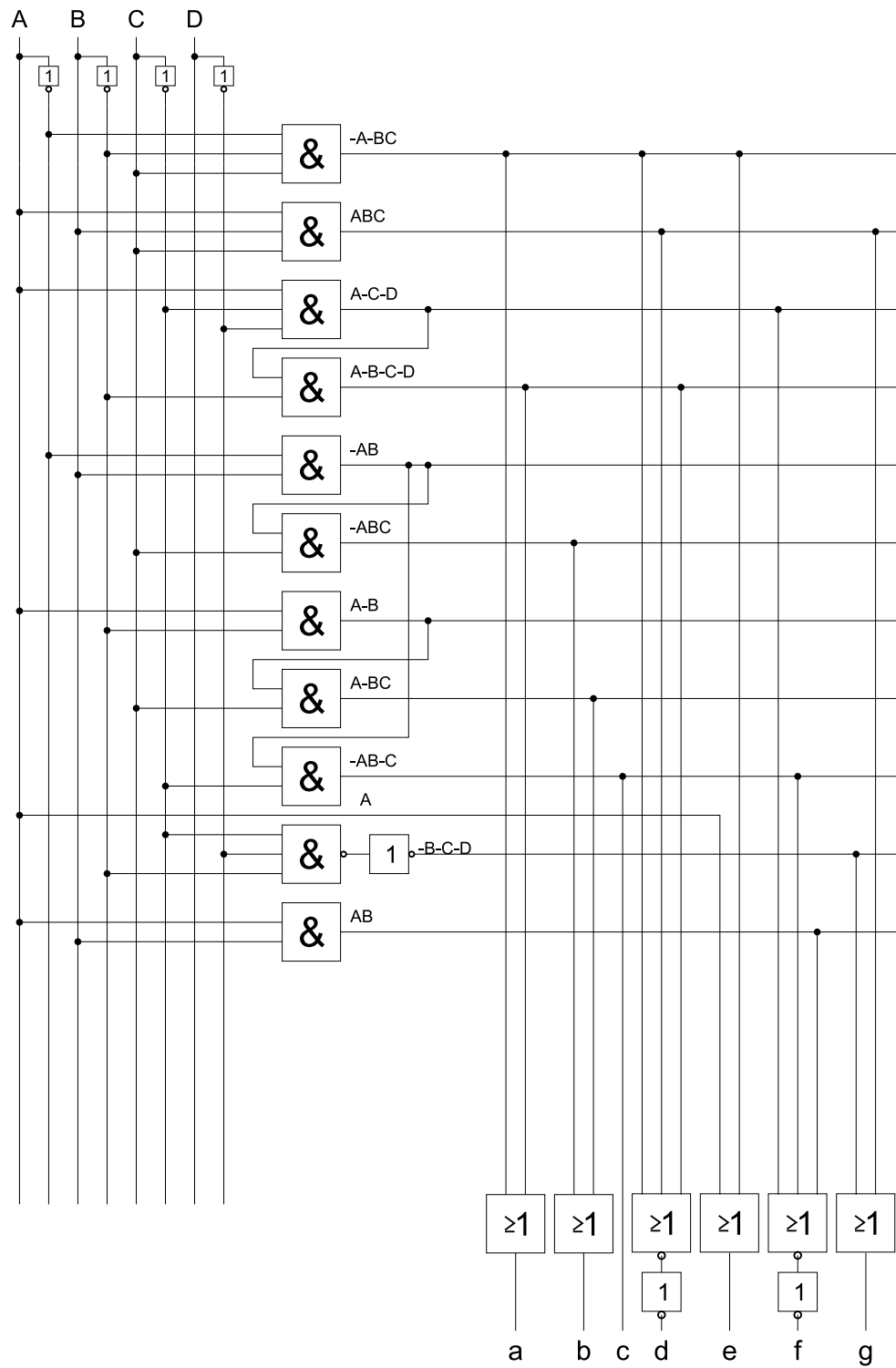


Abbildung 4.6: Schaltung des 4 zu 7 BCD-Decoders

$\Sigma/3$	\leftrightarrow	Ausgang a
$\Pi/8$	\leftrightarrow	$\Sigma/4$
$\Gamma/3$	\leftrightarrow	$\Sigma/5$
$\Sigma/6$	\leftrightarrow	Ausgang b
$\Gamma/6$	\leftrightarrow	Ausgang c
$\Lambda/12$	\leftrightarrow	$\Theta/13$
$\Lambda/8$	\leftrightarrow	$\Theta/1$
$\Pi/3$	\leftrightarrow	$\Theta/2$
$\Theta/12$	\leftrightarrow	N/11
N/10	\leftrightarrow	Ausgang d
A	\leftrightarrow	$\Sigma/9$
$\Lambda/12$	\leftrightarrow	$\Sigma/10$
$\Sigma/8$	\leftrightarrow	Ausgang e
$\Lambda/6$	\leftrightarrow	$\Theta/3$
$\Gamma/6$	\leftrightarrow	$\Theta/4$
$\Gamma/8$	\leftrightarrow	$\Theta/5$
$\Theta/6$	\leftrightarrow	N/9
N/8	\leftrightarrow	Ausgang f
N/12	\leftrightarrow	$\Sigma/12$
$\Lambda/8$	\leftrightarrow	$\Sigma/13$
$\Sigma/11$	\leftrightarrow	Ausgang g

Verwendete Gatter 3 \times Inverter, 3 \times AND3, 7 \times AND2, 1 \times NAND3, 4 \times OR2, 2 \times NOR3.

Kontrolle Wir haben die Funktionsfähigkeit der Anzeige überprüft. Leider funktionierte sie nicht komplett richtig. Bei den Ziffern 5,6,7 und 9 wurden die Segmente a und f nicht korrekt angezeigt. Nachdem wir die errechneten Funktionen überprüft hatten, wußten wir, daß irgendeins der ca. 60 Kabel falsch gesteckt sein musste (oder mehrere). Wir konnten dies aber nicht mehr finden.

Versuch 5: Mikrocontroller

Gruppe David Frese, Christos Freris.

Protokollführer Christos Freris

Verwendete Geräte Arbeitsplatz prakt1

5.1 DAvE

Wir haben alle Einstellungen die im Praktikumsskript angegeben waren vorgenommen und den C-Code erzeugt.

5.2 Programmcode

Wir haben folgenden Programmteil in den Abschnitt (Main,1) eingefügt:

```
uword i, ival, oval;
```

und in den Teil (Main,2) folgendes eingefügt:

```
while (1) {
    for (i = 1; i <= 30; i++) {
        ival = i*34;           // dadurch val aus [34..1020]
        IO_vWritePort(P2, ival); // Wert senden
        Delay(150);          // Senden dauert ca. 50ms
        ADC_vStartConv();    // D/A/D-Wandlung starten
        // Auf das Ergebnis warten
        while(ADC_bConvReady() != 1);

        oval = ADC_uwReadConv(); // Ergebnis lesen
        // Meldung ausgeben
        printf("input: %d\toutput: %d\tdiff: %d\n", ival,
              oval, oval-ival);
    }
    getchar(); // aus Tastendruck warten
}
```

5.3 Kompilierung

Die Kompilierung verlief erfolgreich.

5.4 Ausführung

Der Betreuer ließ unser Programm auf dem C167 laufen und es wurde folgendes auf dem Terminal ausgegeben (uns wurde die Ausgabe nicht vollständig ausgehändigt):

```
input: 34      output: 30      diff: -4
input: 68      output: 64      diff: -4
input: 102     output: 99      diff: -3
input: 136     output: 134     diff: -2
input: 170     output: 167     diff: -3
input: 204     output: 201     diff: -3
input: 238     output: 235     diff: -3
input: 272     output: 269     diff: -3
input: 306     output: 303     diff: -3
input: 340     output: 336     diff: -4
input: 374     output: 370     diff: -4
input: 408     output: 404     diff: -4
input: 442     output: 438     diff: -4
input: 476     output: 472     diff: -4
input: 510     output: 507     diff: -3
input: 544     output: 540     diff: -4
input: 578     output: 572     diff: -6
input: 612     output: 607     diff: -5
input: 646     output: 640     diff: -6
input: 680     output: 674     diff: -6
input: 714     output: 708     diff: -6
input: 748     output: 741     diff: -7
input: 782     output: 775     diff: -7
input: 816     output: 809     diff: -7
input: 850     output: 843     diff: -7
input: 884     output: 876     diff: -8
```

5.4.1 Erklärung

In der for-Schleife wird i von 1 bis 30 hochgezählt. Bei jedem Schleifendurchgang wird der Wert $ival = 34 * i$ auf den E/A-Port P2 geschrieben. Da das Signal kurze Zeit braucht um am D/A-Wandler anzukommen, machen wir einen Aufruf der `delay`-Funktion mit Parameter 150, was einer Pause von $75ms$ entspricht. Dieser Wert wird dann mit Hilfe des an P2 angeschlossenen D/A-Wandlers in ein analoges Signal umgewandelt, welches dann auf den Eingang des A/D-Wandlers übertragen wird. Als nächstes starten wir

den Einlesevorgang des ADC's (`ADC_vStartConv()`). Die `while`-Schleife wird nun so oft durchlaufen, bis der ADC bestätigt, daß er das Signal gelesen hat (`ADC_bConvReady()`). Danach wird der Ausgabewert des ADC's in *oval* gespeichert (`ADC_uwReadConv()`). Diese Werte, sowie deren Differenz, werden nun auf dem Terminal ausgegeben.

Abweichungen Die aufgetretenen Abweichungen sind auf die Digital-Analog und Analog-Digital-Wandlung zurückzuführen. Diese Wandlungen sind anscheinend nicht perfekt realisierbar.

Versuch 6: Minirechner

Gruppe David Frese in Zusammenarbeit mit der Gruppe Thomas Gärtner / Norbert Freudemann, Christos Freris machte diesen Versuch mit der Gruppe Claus Kunst / Luca Ivkiç zusammen.

Protokollführer David Frese

Verwendete Geräte Experimentierboard: Nr. 2, Netzteil: Nr. 1

6.1 Aufgabe 1 und 2

Die Programme sind in Abbildungen 6.1 dargestellt. Sie wurden jeweils von einem Betreuer kontrolliert.

6.2 Division durch 7

Die möglichen Eingabewerte für $A \text{ div } 7$ sind nur die 0, 7 und 14. Es ergibt sich also eine Funktionstabelle:

Eingabe	Ausgabe
0000	0000
0111	0001
1110	0010

Diese Werte kann man mit folgender Funktion berechnen:

$$A \mapsto 2r(1r(A) + 1)$$

Wie man leicht anhand einer Wertetabelle sieht:

	A	1r(A)	1r(A)+1	2r(1r(A)+1)	A div 7
0	0000	0000	0001	0000	0
7	0111	0011	0100	0001	1
14	1110	0111	1000	0010	2

Ein entsprechendes Programm ist in Abbildung 6.1 dargestellt.

Befehl (Mnemonic)					Adresse					Bit IC	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
SRG B	SRG A	MUX	ALU	SRG F	8	7	6	5	4	3	2	1	0	Pin	SRG B	SRG A			MUX	SRG F	ALU								
															S1	S0	SIL	SIR	S1	S0	SEL	S1	S0	-Cr	M	S3	S2	S1	S0
Aufgabe 1(a)																													
-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	0	
-	-	-	-	-	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1
-	-	-	-	-	0	0	0	0	0	0	0	1	0	1	1	1	0	1	0	1	0	0	1	1	0	1	1	1	1
-	-	-	-	-	0	0	0	0	0	0	1	1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	0	
Programm 1(b)																													
-	LD A	-	-	-	0	0	0	0	0	0	0	0	0	-	-	-	-	1	1	-	-	-	-	-	-	-	-	-	
-	SHR 0	-	-	-	0	0	0	0	0	0	0	1	-	-	0	-	1	0	-	-	-	-	-	-	-	-	-	-	
-	SHR 1	-	-	-	0	0	0	0	0	0	1	0	-	-	1	-	1	0	-	-	-	-	-	-	-	-	-	-	
-	SHL 0	-	-	-	0	0	0	0	0	0	1	1	-	-	0	0	1	-	-	-	-	-	-	-	-	-	-	-	
-	SHL 1	-	-	-	0	0	0	0	0	1	0	0	-	-	1	0	1	-	-	-	-	-	-	-	-	-	-	-	
Programm 1(c)																													
LD B	LD A	A	A XOR B	LD	0	0	0	0	0	0	0	0	0	1	1	-	-	1	1	0	1	1	1	1	0	1	1	0	
NOP	NOP	A	A + B	LD	0	0	0	0	0	0	0	0	1	0	0	-	0	0	0	1	1	1	0	1	0	0	0	1	
NOP	NOP	A	A - B	LD	0	0	0	0	0	0	1	0	0	0	0	-	0	0	0	1	1	0	0	0	1	1	0	0	
NOP	NOP	A	(A v ~B)+(A ^ B)+1	LD	0	0	0	0	0	0	0	1	1	0	0	-	0	0	0	1	1	0	0	1	0	1	0	1	
NOP	NOP	A	A	LD	0	0	0	0	0	1	0	0	0	0	0	-	0	0	0	1	1	1	0	0	0	0	0	0	
NOP	NOP	A	A v B	LD	0	0	0	0	0	1	0	1	0	0	0	-	0	0	0	1	1	1	0	0	0	0	0	1	
Programm 2																													
(15 - 3) / 4																													
LD 3	LD 15	A	A - B	LD	0	0	0	0	0	0	0	0	0	1	1	-	-	1	1	0	1	1	0	0	0	1	1	0	
-	-	-	-	SHR	0	0	0	0	0	0	0	1	-	-	-	-	-	-	-	1	0	-	-	-	-	-	-	-	
-	-	-	-	SHR	0	0	0	0	0	0	1	0	-	-	-	-	-	-	-	1	0	-	-	-	-	-	-	-	
(6 * 2 - 9) * 2																													
-	LD 6	-	-	-	0	0	0	0	0	0	1	1	-	-	-	-	1	1	-	-	-	-	-	-	-	-	-	-	
LD 9	SHL 0	A	A - B	LD	0	0	0	0	0	1	0	0	0	1	1	-	0	0	1	0	1	1	0	0	0	1	1	0	
-	-	-	-	SHL	0	0	0	0	0	1	0	1	-	-	-	-	-	-	-	0	1	-	-	-	-	-	-	-	
((4 * 2 - 7) * 2 + 6) / 4																													
-	LD 4	-	-	-	0	0	0	0	0	1	1	0	-	-	-	-	1	1	-	-	-	-	-	-	-	-	-	-	
LD 7	SHL 0	A	A - B	LD	0	0	0	0	0	1	1	1	1	1	1	-	0	1	0	1	1	0	0	0	1	1	0	0	
-	-	-	-	SHL	0	0	0	0	1	0	0	0	-	-	-	-	-	-	-	0	1	-	-	-	-	-	-	-	
LD 6	-	F	A + B	LD	0	0	0	0	1	0	0	1	1	1	1	-	1	1	1	1	1	0	1	0	0	0	1	1	
-	-	-	-	SHR	0	0	0	0	1	0	1	0	-	-	-	-	-	-	-	1	0	-	-	-	-	-	-	-	
-	-	-	-	SHR	0	0	0	0	1	0	1	1	-	-	-	-	-	-	-	1	0	-	-	-	-	-	-	-	
(5 - (1 + 5) / 2) * 4																													
LD 5	LD 1	A	A + B	LD	0	0	0	0	1	1	0	0	0	1	1	-	-	1	1	0	1	1	1	0	1	0	0	1	
NOP	-	-	-	SHR	0	0	0	0	1	1	0	1	-	-	0	-	-	-	-	1	0	-	-	-	-	-	-	-	
NOP	-	F	~A	LD	0	0	0	0	1	1	1	0	0	0	0	-	-	-	-	1	1	1	1	1	0	0	0	0	
NOP	-	F	A + B + 1	LD	0	0	0	0	1	1	1	1	1	1	1	-	-	-	-	1	1	1	0	0	1	0	0	1	
-	-	-	-	SHL	0	0	0	1	0	0	0	0	-	-	-	-	-	-	-	0	1	-	-	-	-	-	-	-	
-	-	-	-	SHL	0	0	0	1	0	0	0	1	-	-	-	-	-	-	-	0	1	-	-	-	-	-	-	-	
Programm 3																													
-	LD 0/7/14	-	-	-	0	0	0	0	0	0	0	0	0	-	-	-	-	1	1	-	-	-	-	-	-	-	-	-	
-	SHR 0	A	A + 1	LD	0	0	0	0	0	0	0	0	1	-	-	0	-	1	0	0	1	1	0	0	0	0	0	0	
-	-	-	-	SHR	0	0	0	0	0	0	1	0	-	-	-	-	-	-	-	1	0	-	-	-	-	-	-	-	
-	-	-	-	SHR	0	0	0	0	0	0	1	1	-	-	-	-	-	-	-	1	0	-	-	-	-	-	-	-	
Programm 4																													
LD B	LD A	-	NULL	LD	0	0	0	0	0	0	0	0	0	1	1	-	-	1	1	-	1	1	0	0	0	0	1	1	
NOP	NOP	-	-	NOP	0	0	0	0	0	0	0	0	1	0	0	-	-	0	0	-	0	0	-	-	-	-	-	-	
NOP	SHL 0	-	-	SHL	0	0	0	0	0	0	1	0	0	0	0	-	0	0	1	-	0	1	-	-	-	-	-	-	
NOP	NOP	-	-	NOP	0	0	0	0	0	0	1	1	0	0	0	-	0	0	-	0	0	-	-	-	-	-	-	-	
NOP	SHL 0	-	-	SHL	0	0	0	0	0	1	0	0	0	0	0	-	0	0	1	-	0	1	-	-	-	-	-	-	
NOP	NOP	-	-	NOP	0	0	0	0	0	1	0	1	0	0	0	-	0	0	-	0	0	-	-	-	-	-	-	-	
NOP	SHL 0	-	-	SHL	0	0	0	0	0	1	1	0	0	0	0	-	0	0	1	-	0	1	-	-	-	-	-	-	
NOP	NOP	F	A + B	LD	1	0	0	0	0	1	0	1	0	0	0	-	0	0	1	1	1	1	0	1	0	0	1	1	
NOP	SHL 0	-	-	SHL	1	0	0	0	0	1	0	0	0	0	0	-	0	0	1	-	0	1	-	-	-	-	-	-	
NOP	NOP	F	A + B	LD	1	0	0	0	0	1	0	1	0	0	0	-	0	0	1	1	1	1	0	1	0	0	1	1	
NOP	SHL 0	-	-	SHL	1	0	0	0	0	1	1	0	0	0	0	-	0	0	1	-	0	1	-	-	-	-	-	-	
NOP	NOP	F	A + B	LD	1	0	0	0	0	1	1	0	0	0	0	-	0	0	1	1	1	1	0	1	0	0	1	1	
NOP	SHL 0	-	-	SHL	1	0	0	0	0	1	1	0	0	0	0	-	0	0	1	-	0	1	-	-	-	-	-	-	
NOP	NOP	F	A + B	LD	1	0	0	0	0	1	1	0	0	0	0	-	0	0	1	1	1	1	0	1	0	0	1	1	

Abbildung 6.1: Funktionstabelle

6.3 4 Bit * 4 Bit Multiplikation

Diese Multiplikation lässt sich informell beschreiben durch:

- Initialisiere F mit 0
- Wenn das höchstwertige Bit von A 1 ist, dann addiere B zu F
- Schiebe sowohl A als auch F um eins nach links.
- Wiederhole die letzten beiden Schritte noch 3 mal.

Um verschiedene Befehle in Abhängigkeit vom obersten bit von A ausführen zu können, muß man den Schalter *A8 manuell/D3* auf D3 schalten. Ein entsprechendes Programm ist in Abbildung 6.1 dargestellt.

Versuch 7: Mikroprozessor

Gruppe David Frese, Christos Freris.

Protokollführer Christos Freris

7.1 Einbindung

Wir haben in die erzeugte Datei `main.c` die Aufgabenteile eingefügt. Die einzelnen Aufgabenteile sind unten im einzelnen aufgeführt bzw. erklärt. Drumherum mußte noch folgender Rahmen eingefügt werden:

```
ubyte AB = 0;
ubyte A = 0, B = 0;
int i;

/* Fragmente */

while (1); // Endlosschleife
```

7.2 Bitschiebeoperationen

```
//----- Aufgabe 1 -----
printf("Bitschiebeoperationen\n");
printf("Bitte AB auf 10110010 setzen und Taste dr\"){u}cken.");
while(!getchar()); // auf Tastendruck warten
AB = (ubyte)IO_uwReadPort(P8); // Wert einlesen

//--- Arithmetisch nach Rechts schieben
if (AB & 0x80) // Ist das Vorzeichen gesetzt?
    AB = (AB >> 2) | 0xC0; // Schieben und beide obersten Bits
    // wieder setzen
else
    AB = (AB >> 2); // sonst nur schieben
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
while(!getchar()); // auf Tastendruck warten
```

```

//--- Linksschieben um 2 Bits
AB = AB << 2;
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
while(!getchar()); // auf Tastendruck warten

//--- Logisch Rechtsschieben um 2 Bits
AB = AB >> 2;
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
while(!getchar()); // auf Tastendruck warten

//--- Rotieren um 5 Bits nach links
// 5 mal um 1 rotieren, d.h. das oberste Bit unten reinsetzen
for (i=0; i < 5; i++) {
    if (AB & 0x80)
        AB = (AB << 1) | 1; // das oberste Bit war 1
    else
        AB = AB << 1;
}
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
while(!getchar()); // auf Tastendruck warten

```

7.2.1 Ergebnisse

Rechnung	erwartet	abgelesen
2r arith.	11101100	11101100
2l	10110000	10110000
2r	00101100	00101100
rotl5	10000101	10000101

7.3 Logische und arithmetische Operationen

```

//----- Aufgabe 2 -----
printf("Logische und arithmetische Operationen\n");
printf("Bitte A auf 5 und B auf 3 setzen und Taste dr\"{u}cken.");
while(!getchar()); // auf Tastendruck warten
AB = (ubyte)IO_uwReadPort(P8); // Wert einlesen
A = (AB >> 4); B = (AB & 0x0F);

IO_vWritePort(P2, ~(A+B)); // A+B auf LEDs ausgeben
while(!getchar()); // auf Tastendruck warten
IO_vWritePort(P2, ~(A-B)); // A-B auf LEDs ausgeben
while(!getchar()); // auf Tastendruck warten
IO_vWritePort(P2, ~(A&B)); // A&B auf LEDs ausgeben
while(!getchar()); // auf Tastendruck warten

```

```

IO_vWritePort(P2, ~(A|B)); // A|B auf LEDs ausgeben
while(!getchar());        // auf Tastendruck warten
IO_vWritePort(P2, ~(A^B)); // A^B auf LEDs ausgeben
while(!getchar());        // auf Tastendruck warten
IO_vWritePort(P2, ~(~A)); // ~A auf LEDs ausgeben
while(!getchar());        // auf Tastendruck warten

```

7.3.1 Ergebnisse

Rechnung	erwartet	abgelesen
5+3	1000	1000
5-3	0010	0010
5&3	0001	0001
5 3	0111	0111
5^3	0110	0110
~5	1010	1010

7.4 Rechenoperationen

```

//----- Aufgabe 3 -----
printf("Rechenoperationen\n");

//--- (15-3)/4
AB = (15-3) >> 2;
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
printf("Ergebnis von (15-3)/4 ablesen!");
while(!getchar());      // auf Tastendruck warten

//--- (6*2 - 9)*2
AB = ((6 << 1) - 9) << 1;
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
printf("Ergebnis von (6*2 - 9)*2 ablesen!");
while(!getchar());      // auf Tastendruck warten

//--- ((4*2 - 7)*2 + 6)/4 oder (4*2 - 7)*2 + 6/4 ??????????????
AB = (((4 << 1) - 7) << 1) + 6 >> 2;
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
printf("Ergebnis von (4*2 - 7)*2 + (6/4) ablesen!");
while(!getchar());      // auf Tastendruck warten

//--- (5-(1+5)/2)*4
AB = (5-((1+5) >> 1)) << 2;
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
printf("Ergebnis von (5-(1+5)/2)*4 ablesen!");

```

```
while(!getchar()); // auf Tastendruck warten
```

7.4.1 Ergebnisse

Rechnung	erwartet	abgelesen
$(15 - 3)/4$	0011	0011
$(6 * 2 - 9) * 2$	0110	0110
$(4 * 2 - 7) * 2 + (6/4)$	0011	0011
$(5 - (1 + 5)/2) * 4$	1000	1000

7.5 (4-Bit * 4-Bit)-Multiplikation

```
//----- Aufgabe 4 -----
printf("(4-Bit * 4-Bit)-Multiplikation\n");
printf("Bitte A auf 2 und B auf 5 setzen und Taste dr\"{u}cken.");
while(!getchar()); // auf Tastendruck warten
AB = (ubyte)IO_uwReadPort(P8); // Wert einlesen
A = (AB >> 4); B = (AB & 0x0F);
AB = 0; // Initialisieren auf 0
for (i=0; i<4; i++) {
    if (A & 0x8) // Ist oberstes Bit von A gesetzt?
        AB = AB + B; // Dann B zum Ergebnis addieren
    AB = AB << 1; // Ergebnis und
    A = A << 1; // A um eins nach links schieben
}
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
printf("Ergebnis von 2*5 ablesen!")
while(!getchar()); // auf Tastendruck warten

printf("Bitte A auf 3 und B auf 3 setzen und Taste druecken.");
while(!getchar()); // auf Tastendruck warten
AB = (ubyte)IO_uwReadPort(P8); // Wert einlesen
A = (AB >> 4); B = (AB & 0x0F);
AB = 0; // Initialisieren auf 0
for (i=0; i<4; i++) {
    if (A & 0x8) // Ist oberstes Bit von A gesetzt?
        AB = AB + B; // Dann B zum Ergebnis addieren
    AB = AB << 1; // Ergebnis und
    A = A << 1; // A um eins nach links schieben
}
IO_vWritePort(P2, ~AB); // Wert auf LEDs ausgeben
printf("Ergebnis von 3*3 ablesen!")
while(!getchar()); // auf Tastendruck warten
```

7.5.1 Ergebnisse

Rechnung	erwartet	abgelesen
2*5	1010	1010
3*3	1001	1001