

Kernel: Python 3 (Ubuntu Linux)

Fakultät für Physik

Physikalisches Praktikum P1 für Studierende der Physik

Versuch P1-63, 64, 65 (Stand: Oktober 2023)

[Raum F1-08](#)

Schaltlogik

Name: Aichert Vorname: Julius E-Mail: uhoeb@student.kit.edu

Name: Achtner Vorname: Martin E-Mail: urrvl@student.kit.edu

Name: Möll Vorname: Jonas E-Mail: uboyb@student.kit.edu

Gruppennummer: Mo01

Betreuer: Joshua Gabriel

Versuch durchgeführt am: 06.11.2023

Beanstandungen:

Testiert am: _____ Testat: _____

Durchführung

Dieser Praktikumsversuch ist so gegliedert, dass die einzelnen Aufgabenteile eine logische Folge bilden. Auch ohne Vorkenntnisse erhalten Sie so wertvolle Einblicke in die Grundlagen der Digitaltechnik. Es genügt, wenn Sie etwa zwei Drittel der vorgeschlagenen Aufgaben bearbeiten. Treffen Sie Ihre Auswahl je nach Vorkenntnissen und Interesse. Die mit (*) gekennzeichneten **Aufgaben 3.1 und 3.2, 4, 5.1 und 6.1 sollten jedoch auf jeden Fall bearbeitet werden.**

Aufgabe 1: Gatter aus diskreten Bauelementen

Bei dieser Aufgabe lernen Sie einfachste Grundsaltungen der Schaltlogik kennen.

Aufgabe 1.1: Das AND-Gatter

Bauen Sie ein Dioden-AND-Gatter auf und prüfen Sie seine Funktion.

Lösung:

Ein AND-Gatter lässt sich mit Dioden folgendermaßen realisieren:

 AND-Gatter.png

Abb. 1.1.1: AND-Gatter

Falls A und B auf dem Potential 5V sind, kann an R1 keine Spannung abfallen, folglich ist C auch auf 5V. Ist A, oder und B auf 0V, so kann die Spannung aufgrund der Dioden D1 bzw D2 an R1 abfallen, und C ist auf 0V. Damit ergibt sich folgende Wahrheitstabelle:

 AND-Wahrheitstabelle.png

Abb. 1.1.2: AND Wahrheitstabelle

Aufgabe 1.2: NOT- und NAND-Gatter

Bauen Sie zusätzlich zum AND-Gatter ein Transistor-NOT-Gatter auf und bilden Sie durch Hintereinanderschalten ein NAND-Gatter. Prüfen Sie seine Funktion.

Lösung:

Ein NOT-Gatter lässt sich folgendermaßen realisieren:



Abb. 1.2.1: NOT-Gatter

Ist A auf einem Potential von 5V, so schaltet der Transistor durch, und Strom kann von den angeschlossenen 5V im oberen Bildbereich über den Widerstand R3, an dem die Spannung abfällt, zu GND fließen. Da B zwischen Widerstand und Transistor liegt, ist das Potential von B 0V. Falls umgekehrt A auf 0V sein sollte, so schaltet der Transistor nicht durch und es kann kein Strom fließen, daher fällt auch keine Spannung an R3 ab. B liegt folglich auf einem Potential von 5V.



Abb. 1.2.2: NOT Wahrheitstabelle

Ein NAND-Gatter ist die Kombination eines AND- und eines NOT-Gatters. Hierbei wird der Output des AND-Gatters als Input des NOT-Gatters verwendet. Es lässt sich folgendermaßen realisieren:



Abb. 1.2.3: NAND-Gatter

Die Wahrheitstabelle ergibt sich, indem die Wahrheitstabelle des AND-Gatters negiert wird:



Abb. 1.2.4: NAND Wahrheitstabelle

Aufgabe 1.3: OR-Gatter

Bauen Sie ein Dioden-OR-Gatter auf und prüfen Sie seine Funktion.

Lösung:

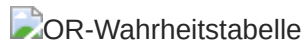
Ein OR-Gatter lässt sich folgendermaßen realisieren:



Abb. 1.3.1: OR-Gatter

Ist A und / oder B High (also auf einem Potential von 5V / 1), so kann ein Strom durch D1 bzw. D2 zu GND fließen. Dabei fällt die Spannung an R1 ab. Da C zwischen D1 / D2 und R1 angebracht ist, ist C dann ebenfalls High. Falls A und B Low (also auf einem Potential von 0V / 0) sind, fließt

nirgendwo ein Strom und an R1 fällt keine Spannung ab, also ist C ebenfalls Low. Die Wahrheitstabelle lautet:



OR-Wahrheitstabelle

Abb. 1.3.2: OR Wahrheitstabelle

Aufgabe 2: Weitere einfache logische Funktionen (Gatter)

Die folgenden Gatter sollen Sie als *Integrated Circuits* (ICs) mit Hilfe einer Experimentiertafel (von Fischer TB05) realisieren. Vergessen Sie nicht, die ICs an die Betriebsspannung ($+5\text{ V}$ und \perp) anzuschließen. Bei allen verwendeten IC-Typen wirken freie Eingänge so, als seien sie an das Potential 'Logisch 1' angeschlossen (sprich 'auf 1 gelegt'). Die bei den Teilaufgaben in eckigen Klammern angegebenen Zahlen bezeichnen die vorgeschlagenen IC-Typ-Nummern.

Aufgabe 2.1: Inverter (NOT-Gatter) aus NAND- oder NOR-Gatter [7400, 7402]

Realisieren Sie einen digitalen Inverter (NOT-Gatter) aus einem NAND- oder einem NOR-Gatter. Betrachten Sie hierzu die Wahrheitstabellen der Gatter. Es gibt für beide Gatter jeweils zwei verschiedene Möglichkeiten, einen Inverter zu realisieren. Das Invertieren einer Dualziffer (der wechselseitige Austausch von 0 und 1) wird auch als 'Negieren' bezeichnet. Dies sollte nicht mit der negativen Zahl (vgl. **Aufgabe 3.3**) verwechselt werden. Das Invertieren aller Ziffern einer Dualzahl wird auch als 'Komplementieren' bezeichnet.

Lösung:

Um ein NOT-Gatter aus einem NAND-Gatter zu realisieren, gibt es zwei verschiedene Möglichkeiten: \

1. Die Eingabewerte A und B des NAND-Gatters werden gleichgeschaltet, also $A = B$. Daraus folgt, dass der AND-Teil des NAND-Gatters immer den Input ergibt, und dieser im NOT-Teil lediglich negiert wird.
2. Ein Eingabewert, z.B. A, wird auf High gesetzt. Falls B dann Low ist, wird C zu High. Falls B High ist, wird C zu Low.

Um ein NOT-Gatter aus einem NOR-Gatter zu realisieren, gibt es ebenfalls zwei Möglichkeiten: \

1. Die Eingabewerte A und B des NOR-Gatters werden gleichgeschaltet, also $A = B$. Daraus folgt, dass der OR-Teil des NOR-Gatters immer den Input ergibt, und dieser im NOT-Teil lediglich negiert wird.
2. Ein Eingabewert, z.B. A wird auf Low gesetzt. Der OR-Teil des NOR-Gatters liefert dann immer den Wert von B, welcher im NOT-Teil dann negiert wird.

Aufgabe 2.2: XOR [7400, 7408, 7432]

Lesen Sie aus der Wahrheitstabelle der XOR-Funktion (siehe Vorbereitungshilfe S.10) deren disjunktive Normalform ab. Realisieren Sie diese (ohne vorherige Umformung) mit Hilfe von

Gattern und überprüfen Sie die Funktion der Schaltung. Sie lernen hiermit ein Verfahren kennen, mit dessen Hilfe Sie ein zunächst nur durch eine Wahrheitstabelle gegebenes Problem durch eine Schaltlogik-Funktion (in der Booleschen Algebra) beschreiben und schließlich als logische Schaltung realisieren können.

Lösung:

Die Wahrheitstabelle der XOR-Funktion lautet:



Abb. 2.2.1: XOR Wahrheitstabelle

Es wird also dann, und nur dann, Wahr zurückgegeben, falls die Eingaben A und B verschieden sind. Die disjunktive Normalenform lautet: $(\bar{A} \wedge B) \vee (A \wedge \bar{B})$, wobei \bar{Q} für die boolesche Negation von Q steht. Zur Realisierung dieser Schaltung braucht man also insgesamt vier Eingaben, wobei zwei davon die Negation der anderen Beiden sind. Ein Schaltplan sieht folgendermaßen aus:



Abb. 2.2.2: XOR-Gatter

Die linken Bauteile (7400N) negieren jeweils A und B. Das Ergebnis wird dann mit dem jeweils anderen Wert in einem AND-Gatter kombiniert. Das ist der $(\bar{A} \wedge B)$ bzw. $(A \wedge \bar{B})$ Teil der disjunktiven Normalenform. Der \vee Schritt wird mit Bauteil 7432N realisiert.

Aufgabe 2.3: XOR mit NAND-Gattern [7400]

Versuchen Sie die Umformung der in **Aufgabe 2.2** aufgestellten XOR-Funktion in die Form

$$f = \overline{\overline{a} \overline{b} \overline{b} a}$$

Realisieren Sie das XOR in dieser Form und überprüfen Sie seine Funktion.

Lösung:

Die Umformung funktioniert folgendermaßen:



Abb. 2.3.1: Umformung XOR

Von (1) nach (2) wurde lediglich zweimal die Identität hinzugefügt, dann vereinfacht. In (4) und (6) wurde ausgenutzt, dass $\overline{\overline{x}} = x$.

Die zugehörige Schaltung ist:



Abb. 2.3.2: XOR-Gatter 2

Der Teil $(\overline{A \wedge B}) = \neg(A \wedge B)$ lässt sich mit einem NAND-Gatter realisieren. Alle andere Teile folgen genau dem gleichen Schema, weshalb sich die gesamte Schaltung mit NAND-Gattern aufbauen lässt.

Aufgabe 3: Addierer

Aufgabe 3.1: Halbaddierer [7408, 7486] (*)

Der Halbaddierer soll zwei einstellige Dualzahlen addieren. Überlegen Sie sich die zugehörige Wahrheitstabelle (einschließlich Summe und Übertrag). Realisieren Sie den Halbaddierer mit je einem passenden Gatter für Summe und Übertrag und prüfen Sie seine Funktion.

Lösung:

Die Wahrheitstabelle ist wie folgt:

 Addierer-Wahrheitstabelle

Abb. 3.1.1: Halbaddierer Wahrheitstabelle

Wenn A und B beide null sind, ist deren Summe auch null. Falls nur eine der beiden nicht null ist, ist das Ergebnis eins. Falls beide eins sind, ist das Ergebnis 2, in binärer Schreibweise also 10. D.h., dass der Übertrag eins ist, während das Ergebnis wieder null ist. Es ist zu erkennen, dass die Summe der XOR-Wert von A und B ist, während der Übertrag der AND-Wert von A und B ist. Deshalb sieht das Schaltbild folgendermaßen aus:

 Halbaddierer-Schaltplan

Abb. 3.1.2: Halbaddierer Schaltplan

Aufgabe 3.2: Volladdierer [7408, 7486, 7432] (*)

Überlegen Sie sich eine 1-Bit-Volladdierer-Schaltung, die aus zwei Halbaddierern und einem OR-Gatter für deren Übertragsausgänge besteht. Bauen Sie die Schaltung auf und prüfen Sie ihre Funktion.

Lösung:

Der Volladdierer ist bereits wesentlich komplexer. Hier soll ein dritter Wert, nämlich der Übertrag einer vorherigen Addition berücksichtigt werden. Dieser ist mit den eigentlichen Eingängen A und B gleichberechtigt. In der Wahrheitstabelle wird der Übertrag aus der vorhergehenden Rechnung mit C bezeichnet. Die Wahrheitstabelle lautet dann:

 Volladdierer-Wahrheitstabelle

WahrheitAbb. 3.2.1: stabelle Volladdierer

Die dazugehörige Schaltung ist:



Abb. 3.2.2: Volladdierer Schaltplan

Zuerst wird der Wert von A und B mit einem Halbaddierer addiert. Die Summe von A und B wird dann mit dem Übertrag in einem Halbaddierer addiert. Die Summe des Volladdierers ist die Summe des letzten Halbaddierers. Der Übertrag des Volladdierers ist der Output eines OR-Gatters mit den Eingängen des Übertrags der zweiten Halbaddition und des Übertrags der ersten Halbaddition. Es ist festzustellen, dass bei dem OR-Gatter niemals beide Eingänge eins sein können, da sonst A oder B gleichzeitig eins oder null sein müsste.

Wenn man die Summe zweier Binärzahlen berechnen will, so muss jede Stelle mit einem Volladdierer addiert werden. Bei der letzten Stelle (und damit bei den ersten zwei Bit, die in den Volladdierer Eingegeben werden), ist der Übertrag 0.

Aufgabe 4: Speicherelemente (*)

Mit dieser Aufgabe stellen wir Ihnen eine Reihe von Flip-Flop-Typen vor. Flip-Flops (FF) sind bistabile Kippstufen, die als digitale Speicher dienen. Sie sind auch die Bausteine von Schieberegistern und Zählern.

Aufgabe 4.1: RS-Flip-Flop (RS-FF) [7400]

Verbinden Sie zwei NAND-Gatter zu einem FF. Ermitteln Sie seine Funktionstabelle. Eine Funktionstabelle beschreibt die Abhängigkeit der Ausgangszustände (hier an Q und \overline{Q}) von den Eingangszuständen (hier an R (Reset) und S (Set)).

Lösung:

Die Funktionstabelle lautet:



Abb. 4.1.1: RS-FF Funktionstabelle

Ist R eins, so wird Q auf null gesetzt. Ist dagegen S eins, so wird Q auf eins gesetzt. Es darf nicht passieren, dass S und R gleichzeitig eins sind, denn sonst wäre $Q = \overline{Q}$, was ein offensichtlicher Widerspruch ist. Der Schaltplan ist:



Abb. 4.1.2: RS-FF Schaltung

Die Negation der Eingänge S und R ist zu beachten.

Aufgabe 4.2: Getaktetes RS-Flip-Flop (RST-FF) [7400]

Bauen Sie ein RST-FF (siehe Vorbereitungshilfe S.19) auf. Ermitteln Sie seine Funktionstabelle. Finden Sie eine Möglichkeit, den „verbotenen Zustand“ zu eliminieren.

Lösung:

Ein Getateter RS-FF (RST-FF) ist ein normaler RS-FF mit einem zusätzlichen Eingang, den extern vorgegebenem Takt. Der Takt wird in einem konstanten Intervall auf High und danach wieder auf Low gesetzt. Es ist nur möglich, den Speicherzustand Q zu ändern, wenn der Takt gerade auf High ist. Um dies zu realisieren, wird S und R jeweils mit einem NAND-Gatter mit T verknüpft. Die Schaltung ist:



Abb. 4.2.1: RST-FF Schaltplan

Die Funktionstabelle lautet also:



Abb. 4.2.2: RST-FF Funktionstabelle

Falls der Takt gerade auf High ist, ist der RST-FF ein ganz normaler FF.

Eine Möglichkeit, den Verbotenen Zustand zu eliminieren, besteht darin, den Takt auf null zu setzen, falls S und R beide auf High sind. Dazu werden S und R mit einem NAND Gatter vor dem RST-FF verknüpft. Der Ausgang aus dem NAND-Gatter wird dann mit einem AND-Gatter mit dem Takt verknüpft. Der Ausgang aus dem AND-Gatter wird dann an den Takt angeschlossen. Sind jetzt S und R High, so ist $\text{NAND}(S, R)$ Low, weshalb auch der Takt auf Null gesetzt wird, also keine Änderungen an dem RST-FF vorgenommen werden. Ist mindestens S oder R Low, so ist $\text{NAND}(S, R)$ High, und der Takt ändert sich nicht.

Aufgabe 4.3: JK-Master-Slave-Flip-Flop (JK-MS-FF) [7400, 7410]

Bauen Sie ein JK-MS-FF (siehe Vorbereitungshilfe S.23) auf. Ermitteln Sie seine Funktionstabelle, in der sowohl die Master-, als auch die Slave-Ausgänge enthalten sein sollen, und die zwischen dem 0-1-Wechsel und dem 1-0-Wechsel des Taktsignals unterscheidet. Beschreiben Sie die Unterschiede und Vorteile dieses FF-Typen gegenüber den zuvor untersuchten FF-Typen.

Lösung:

Der JK-MS-FF wird anders als der RST-FF nicht mit dem Zustand des Taktes gesteuert, sondern mit der "Änderung" des Zustandes des Takts, der sog. Taktflanke. Diese ist Positiv, falls von Low zu High gewechselt wird, und negativ, falls von High zu Low gewechselt wird.

Die Bezeichnung "Master-Slave" kommt daher, dass dieser Flip-Flop eine Hintereinanderschaltung von zwei RST-FF ist. Der erste Flip-Flop heißt Master, der dahintergeschaltete Slave. Sie sind so aneinandergeschaltet, dass der Ausgang des Masters, Q_{master} der Eingang des Slaves (bei S bzw. R) ist. Der Takt wird beim zwischen dem Slave und dem Master negiert. Dies führt zur oben beschriebenen Taktflankensteuerung. Der Schalplan sieht folgendermaßen aus:



Abb. 4.3.1: JK-MS-FF-Schaltplan

Wenn der Takt High ist, speichert der Master FF die Eingänge J und K normal als Ausgang des Masters. Der Slave ist während dessen inaktiv, da dessen Takt auf Low ist. Beim Taktübergang übernimmt der Slave den Ausgang des Masters in der jeweiligen S und R Eingang. Nun ist der Takt des Masters Low, dieser ist folglich nicht aktiv. Die Funktionstabelle lautet:



Abb. 4.3.2: JK-MS-FF Funktionstabelle

Aufgabe 5 Schieben, Multiplizieren, Rotieren

Aufgabe 5.1: 4-Bit-Schieberegister [7400, 7476] (*)

Bauen Sie ein 4-Bit-Schieberegister (siehe Vorbereitungshilfe S.25) auf. Löschen Sie das Register über den C-Eingang. Laden Sie dann das Register durch geeignete Stellungen des Eingangsschalters bei den folgenden vier Taktzyklen (0-1-0) mit einer gewünschten 4-Bit-Dualzahl. Beobachten Sie nach jeder Taktflanke die Ausgänge Q_A , Q_B , Q_C , Q_D . Machen Sie sich klar, dass Sie seriell (zeitlich nacheinander auf einer Leitung) ankommende Information jetzt parallel, gleichzeitig auf verschiedenen Leitungen, vorliegen haben.

Anmerkung: Da mechanische Schalter beim Ein- und Ausschalten prellen, müssen Sie mit Hilfe eines Flip-Flops ein prellfreies Taktsignal erzeugen (siehe Vorbereitungshilfe S.21).

Lösung:

Der Schalplan für ein 4-Bit-Schieberegister sieht folgendermaßen aus:



Abb. 5.1.1: 4-Bit-Schieberegister Schaltplan

Ein 4-Bit-Schieberegister besteht aus vier aneinandergeschalteten JK-MS-FF. Diese sind so miteinander verbunden, dass der Ausgang des ersten JK-MS-FF (Q und \overline{Q}) an die Eingänge des nächsten (J und \overline{K}) angebunden sind. Die Ausgänge Q_i nach dem i-ten JK-MS-FF können auch direkt ausgelesen werden. Des Weiteren sind alle JK-MS-FF an denselben Takt angeschlossen. Sie können auch mit der C Leitung gleichzeitig resetet werden. J und \overline{K} des ersten FF werden mit der Datenleitung, bzw. mit der negierten Datenleitung angesteuert. (Die Negation deshalb, um den verbotenen Zustand zu umgehen.) Bei der ersten Taktflanke werden die Eingaben vom Master des ersten JK-MS-FF gelesen, bei der zweiten werden diese vom Slave des ersten JK-MS-FF gelesen. Bei der dritten werden sie an den Master des zweiten JK-MS-FF anliegen. Die Eingegebenen Information werden also immer weiter, zum nächsten JK-MS-FF geschoben, deshalb der Name "Schieberegister".

Aufgabe 6: Zähler

Elektronische Zähler sind heute vielbenutzte Messinstrumente. Zählt man Ereignisse während einer bestimmten Zeit, so spricht man bei statistischen Ereignissen von Zählratenmessung, bei periodischen von Frequenzmessung. Speist man den Zähler mit einer periodischen Impulsfolge bekannter Frequenz, erhält man eine Uhr. Gibt eine solche Uhr beim Erreichen einer vorgewählten Zeit ein Schaltsignal ab, dann bezeichnet man sie Timer (oder auch auch Wecker).

Aufgabe 6.1: 4-Bit-Asynchrone Zähler [7476] (*)

Schalten Sie gemäß Vorbereitungshilfe S.27 vier JK-MS-FF hintereinander, löschen Sie den Inhalt und beobachten Sie nach jedem Taktzyklus am Zählereingang T die an Q_A , Q_B , Q_C , Q_D angezeigte Dualzahl.

Lösung:

Der Schaltplan für ein 4-Bit-Asynchrone Zähler sieht folgendermaßen aus:



Abb. 6.1.1: 4-Bit-Asynchrone Zähler Schaltplan

Wie das 4-Bit-Schieberegister besteht der 4-Bit-Asynchrone Zähler ebenfalls aus 4 JK-MS-FF. Diese sind jedoch etwas anders miteinander verbunden: der erste Q Ausgang des $n - 1$ JK-MS-FF ist jeweils an den \overline{T} Eingang des n JK-MS-FF angebracht. Alle JK-MS-FF können mit dem C Input resetet werden. Dadurch gibt sich für die Ausgänge Q_i nach dem i -ten JK-MS-FF:

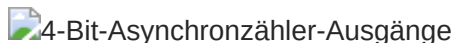


Abb. 6.1.2: 4-Bit-Asynchrone Zähler Ausgänge

Der 4-Bit-Asynchrone Zähler zählt also die Taktflanken des vorgegebenen Takts. Die Ausgänge, hier Q_A bis Q_D sind demnach die Binärschreibweise der Anzahl an vergangenen Takten. Diese Schaltung lässt sich jedoch nicht beliebig erweitern, um beliebig große Zahlen zu zählen, da jeder JK-MS-FF eine gewisse Zeit zum durschalten benötigt, welche sich auf die Zeit des Taktes aufaddiert. Dadurch wird der Asynchrone Zähler langsam und ungenau.

Anmerkung: Alle Bilder sind, falls nicht anders angegeben, aus der Vorbereitungsdatei

<https://gitlab.kit.edu/kit/etp-lehre/p1-praktikum/students/-/blob/main/Schaltlogik/doc/Schaltlogik-Hilfe.pdf> entnommen worden.