

# Elektronik-Praktikum

## Dokumentation des Abschlussprojektes

### *eBlackjack*

### Sommersemester 2009

Johannes Wild (Physik B.Sc.) & Florian Rappl (Physik B.Sc.)

20. Februar 2009



Abbildung 1: Ein Blackjack.

## 1 Grundlegendes

Ziel des Abschlussprojektes ist es, zu zeigen, dass man in diesem zweiwöchigen Kurs einiges über den Aufbau elektrischer Schaltungen gelernt hat. Mit der Wahl unseres Abschlussprojektes „**Elektronisches (Quasi) Blackjack Spiel**“ wollten wir nicht nur ein möglichst Breites Spektrum des im Kurs erlernbaren als Verstanden demonstrieren, sondern auch auf den hohen Spaßfaktor im Umgang mit Elektronik hinweisen. Die Idee hinter dem Projekt entstand während einer Lernaufgabe, welche darin bestand einen *elektronischen Würfel* zu bauen. Im Gegensatz zum realen Blackjack Spiel besitzt unser elektronisches Pendant ein paar kleine Modifikationen welche im folgenden aufgeführt sind:

## 2 Eigenschaften

Um die Modifikationen unseres Abschlussprojektes sehen zu können, müssen wir uns zunächst an den Spielablauf des realen Blackjack Spiels erinnern:

- Black Jack wird mit sechs Paketen französischer Spielkarten zu 52 Blatt, also 312 Karten, an einem etwa halbkreisförmigen Tisch gespielt. An der geraden Seite sitzt der Croupier (Dealer, Bankhalter), ihm gegenüber befinden sich die Plätze für bis zu sieben Spieler (Pointeure).
- Ziel des Spiels ist es, mit zwei oder mehr Karten näher an 21 Punkte heranzukommen als der Croupier, ohne dabei den Wert von 21 Punkten zu überschreiten.

- Asse zählen nach Belieben ein oder elf Punkte.
- Zweier bis Zehner zählen entsprechend ihren Augen zwei bis zehn Punkte.
- Bildkarten (Buben, Damen, Könige) zählen zehn Punkte.
- Sind die Einsätze getätigt, beginnt der Croupier die Karten zu teilen. Jeder Spieler und der Croupier erhalten zuerst eine offene Karte, danach erhält jeder Spieler - nicht aber der Croupier - eine zweite offene Karte.
- Sind alle Spieler bedient, so zieht der Croupier seine zweite Karte. Hat er 17 oder mehr Punkte, so muss er stehen bleiben, hat er 16 oder weniger Punkte, so muss er eine weitere Karte ziehen („Dealer must stand on 17 and must draw to 16“ ).
- Dabei gilt folgende Regel: Der Croupier muss ein Ass stets mit elf Punkten zählen, es sei denn, er würde auf diese Weise den Wert 21 überschreiten; nur dann zählt er das Ass mit einem Punkt. Hat der Croupier z. B. ein Ass und eine Sechs, so muss er das Ass mit elf und die Hand mit siebzehn Punkten bewerten und darf keine weitere Karte ziehen („Dealer stands on soft 17“ ).
- Wenn der Croupier 21 Punkte überschreitet, so haben alle noch im Spiel verbleibenden Teilnehmer automatisch gewonnen. Sonst gewinnen nur jene Spieler, deren Kartenwert näher an 21 Punkte heranreicht als der des Croupiers.
- Die meisten der weiteren Regeln beziehen sich auf das Kasino Spiel (Setzverhalten) und sind daher aus Prinzip nicht im Projekt enthalten.

Aufgrund der Komplexität des Blackjack Spiels mussten diese Regeln ein wenig vereinfacht werden. Folgender Regelsatz kam bei unserer Umsetzung zum Einsatz:

- Man zieht aus einem *unendlichen* „Kartensatz“ Karten mit Werten von 1 – 9.
- Diese Zufallszahlen sind nahezu nicht vorhersagbar (Siehe Realisierungsmerkmale - linearer Zähler mit sehr hoher Zählrate) und können daher als perfekt gemischtes unendlicher Kartensatz angesehen werden.
- Es gibt keinen Croupier - die Spieler spielen nur gegeneinander / sich selbst.
- Jeder Spieler kann aufhören wann er will - solange er unter 21 ist.

- Bei 21 gewinnt ein Spieler sofort - über 21 hat er sofort verloren.
- Es gibt keine Karte mit definierbaren (gutartigen) Wert (Analog zum Ass - Werte je nach Einsatz  $\in \{1, 11\}$ ).

Geplant wurde das Spiel direkt für zwei Spieler (mit zwei **Controllern**) - allerdings musste aus Zeitgründen auf den Aufbau des zweiten Moduls verzichtet werden. Das Spiel ansich funktioniert jedoch auch mit nur einem Controller - es können dadurch beliebig viele Spieler teilnehmen, allerdings nur nacheinander - und die einzelnen Ergebnisse müssen gemerkt / aufgeschrieben werden um einen Sieger zu bestimmen.

Der Spielablauf sieht nun folgendermaßen aus:

1. Strom anschalten.
2. Spiel starten Knopf drücken.
3. Zufallszahl generieren (*Karte ziehen*) über den Ziehen Knopf.
4. Schritt 3 solange ausführen bis man gewonnen hat (= 21), verloren hat (> 21) oder bis man denkt „genug“ zu haben.
5. Spiel beenden über den Spiel starten Knopf.
6. Schritte 3 bis 5 für jeden teilnehmenden Spieler / in jeder Partie die man Spielen möchte durchführen.
7. Strom ausschalten.

### 3 Realisierungsmerkmale

Man kann die aufgebaute Schaltung in folgende Module unterteilen:

1. **Frequenzgenerator** (Realisiert über Bauteil 74LS124) mit Kondensator  $2.2\mu F$  welcher an den Zähler 74LS193 an den Count-Up Eingang angeschlossen wurde (Count-Down verbunden mit  $+5V \equiv 1$ ).
2. **Schalterelemente** (alle Prellfrei - realisiert über RS-Flipflops) - meistens über den 74LS173A positiv flankengetriggert.
3. **Anzeigeelemente** welche über einen Treiber (74LS245) und eine **BCD-BIN EPROM** Einheit als zweistellige Dezimalzahl angesteuert werden.

4. Logikeinheit welche die Regeln des Spiels verwaltet und bei ausgewählten Ereignissen die Spielsteuerung übernimmt (Spiel beendet bei einem größeren Wert als 21, Spiel gewonnen bei exakt 21, sonst nichts).
5. Ein Additionselement welches den aktuellen Spielstand speichert und beim *ziehen* einer Karte aktualisiert.

Im folgenden werden die aufgeführten Module näher beschrieben.

### 3.1 Frequenzgenerator

Vom Prinzip her wurde der Frequenzgenerator übernommen, welcher als Aufgabe am Tag 8 des Praktikums gebaut werden sollte (Aufgabe 11.2 im Skript). Als Grundlegenden Baustein für diese Schaltung entschieden wir uns für den 74LS124.

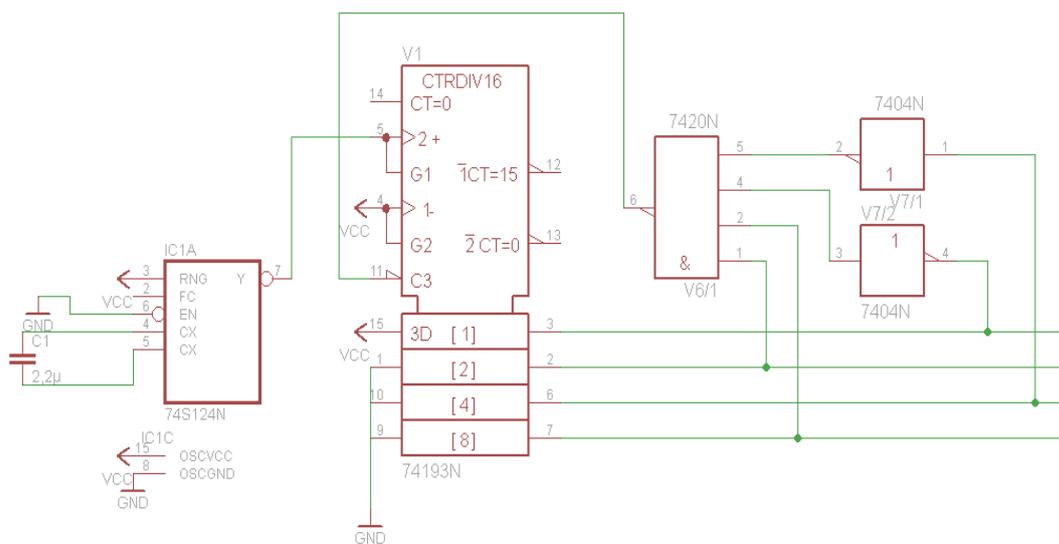


Abbildung 2: Der Aufbau des Frequenzgenerator Moduls.

Direkt am Frequenzerzeugenden IC 74LS124 wurde der Zähler 74LS193 angeschlossen, welcher über eine Logikeinheit einen automatischen Clearvorgang (Load auf 1) durchführt. Die Logik dazu wurde so entworfen, dass bei Ausgabe von 1010 bzw.  $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$  der Load Eingang auf 0 geschaltet wurde - dieser Eingang liegt sonst immer auf 1. Realisiert werden konnte dies über einen **NAND** Schalter, welcher das 4. und 2. Bit direkt annimmt (die beiden 1er) - und Bit 1/3 invertiert (aus 0en werden 1er). Somit gibt

diese 4er **AND** Schaltung nur 1 zurück wenn die oben genannte Folge 1010 erfüllt ist. Da es sich jedoch hier um eine **NAND** Schaltung handelt, wird das AND-Ergebnis noch invertiert - und schon haben wir den gewünschten Effekt erhalten. Die 4 Ausgänge können nun mit den weiteren Modulen verbunden werden.

### 3.2 Schalterelemente

Neben den eigentlichen Schaltern - hier wurden LED Kippschalter verwendet - war ein Hauptbestandteil dieses Moduls die Umsetzung von Prellfreien Schaltern, welche in manchen Fällen nur kurz (meistens bei aufsteigender Flanke) schalten sollten. Die LEDs sollten außerdem nur dann leuchten, wenn ein gedrückter Schalter eine Funktion ausführen kann (z.B. kann keine Karte mehr gezogen werden wenn das Spiel verloren wurde - erst nach Neustart des Spiels ist dies wieder möglich).

### 3.3 Anzeigeelemente

Die Anzeigeelemente spielen eine sehr wichtige Rolle, da sie die direkte Schnittstelle vom Benutzer des Spiels zum Spiel selbst darstellt. Als Primäre Anzeige wurde die im Schaltbrett Vierfachsiebensegmentanzeige verwendet. Die vorderen 2 Stellen wurden für die Anzeige des Gesamtstandes (Spielstand) verwendet - die hinteren 2 Stellen (nur eine notwendig - aber aufgrund der Symmetrie so ausgeführt) für den zuletzt gezogen Zufallswert. Des weiteren wurden noch LEDs zur Anzeige eines direkten Sieges (bei einem Gesamtstand von genau 21) bzw. einer direkten Niederlage (bei einem Gesamtstand größer als 21) verwendet. Die Aktivität der Knöpfe (Kann gedrückt werden / Kann nicht gedrückt werden) kann über deren LEDs eingesehen werden.

Bei Werten  $\in \{0, 1, 2, \dots, 9\}$  zeigt die Anzeige auch diesselben an, weshalb die gezogenen Zufallszahlen ( $\in \{1, \dots, 9\}$ ) direkt an die Siebensegmentanzeige (rechte Stelle) gesendet werden kann. Beim Gesamtstand hat man Werte  $\in \{0, \dots, 29\}$  - welche bei von der Siebensegmentanzeige im Hexadezimalsystem angezeigt werden würde (z.B.  $29_{10} \mapsto 13_{16}$ ). Dies gilt es zu verhindern, damit der Spielstand direkt und intuitiv erfasst werden kann. Prinzipiell ist es möglich dies über eine Verschaltung von diversen Logik Elementen zu erreichen - aus Platz und v.a. Zeitgründen wurde ein vorprogrammiertes **EPROM**, welches genau für diese Aufgabe geschaffen wurde, verwendet. Eingespeist werden unsere notwendigen 5 Bit, sowie alle weiteren Bit auf Masse ( $\equiv 0$ ). Als Ausgabe kann man genau die 4 jeder Siebensegment Anzeige, also 2 Siebensegment Anzeige à 4 Bit - daher die 8 Bit Ausgangsbreite, verwenden.

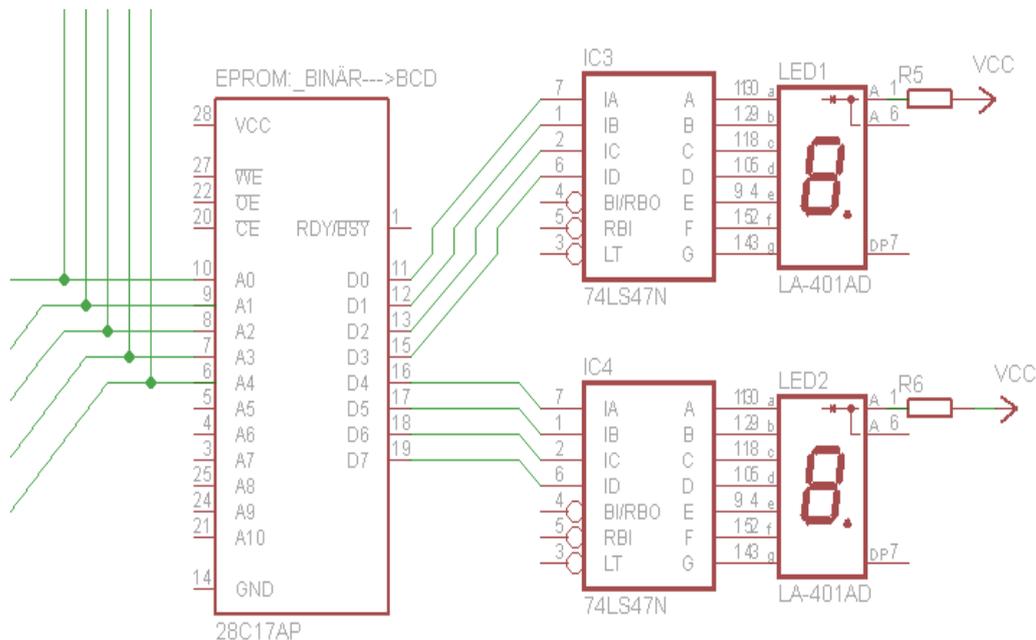


Abbildung 3: Anschluss des BCD-Binary EP-ROMs an die Siebensegmentanzeige.

### 3.4 Logikeinheit

Ohne die Logikeinheit könnte man den Namen „Blackjack“ ganz streichen, da der Addieren permanent hochzählen würde und somit der Spieler selbst entscheiden müsste, ob er gewonnen oder verloren hat. Somit bildet die Logikeinheit das Gehirn unserer Schaltung.

An die Logikeinheit werden die 5 Bit des Gesamtstandes angeschlossen. Dabei werden die ersten 4 Bit (gezählt vom **LSB** aus) an einen Komparator, 74LS85 angeschlossen und das 5. Bit ( $A_4$ , **MSB**) an zwei AND Schaltung. Die erste AND Schaltung ( $O_1$ ) wird mit dem  $A > B$  Ausgang des Komparators verbunden, die zweite Schaltung ( $O_2$ ) mit dem  $A = B$  Ausgang des Komparators. Neben den 4 Bit ( $A : A_0..A_3$ ) werden weitere 4 Bit direkt an den Komparator angelegt ( $B$ ) - welche auf den  $0101_2 \equiv 5_{10}$  eingestellt werden. Nun ergibt sich folgendes Verhalten der Logikschaltung (an Logikausgängen der beiden AND Schaltungen):

$$\begin{cases} A_4 & A_3 & A_2 & A_1 & A_0 & O_1 & O_2 \\ 0 & x & x & x & x & 0 & 0 \\ 1 & 0 & 0 & x & x & 0 & 0 \\ 1 & 0 & 1 & 0 & y & 0 & y \\ 1 & x & 1 & 1 & x & 1 & 0 \end{cases}$$

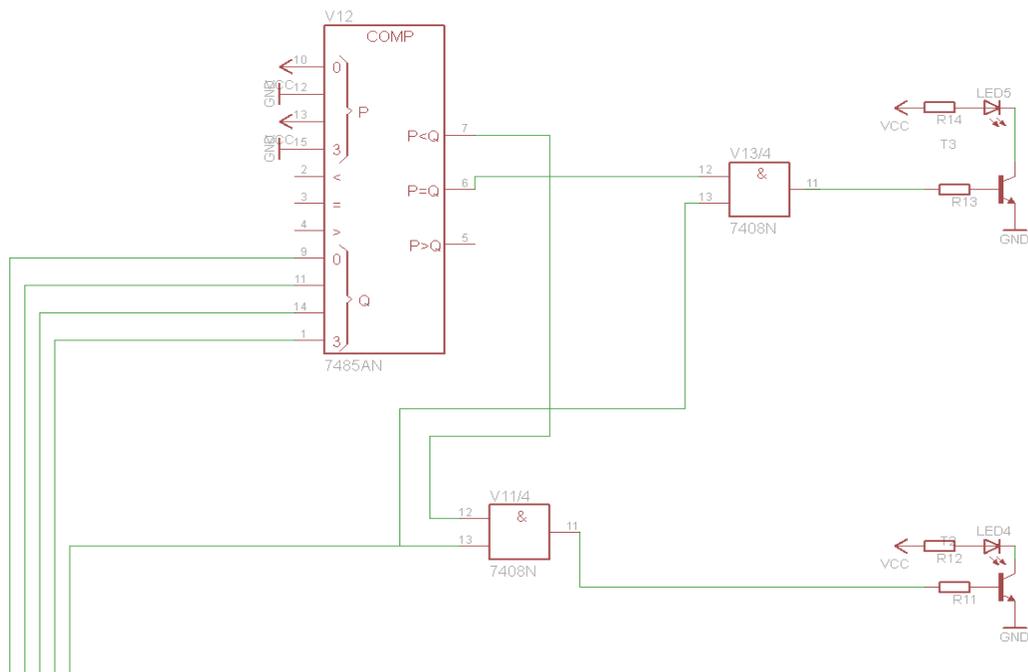


Abbildung 4: Struktur der Logikeinheit.

Dies ist der gewünscht Ablauf des Spiels wie Eingangs beschrieben. Die beiden Ausgänge können nun mit LEDs und weiterer Logik verbunden werden - um Anzeige und Bedienbarkeit des Spiels zu steuern.

### 3.5 Additionselement

Wenn die Logikeinheit das Gehirn unserer Schaltung ist, so ist das Additionselement das Herz dieser. Ohne die Addition der generierten Zufallszahlen würde das Spiel nicht vorranschreiten und wir hätten einen permanenten statischen Zustand. Die Realisierung dieses Moduls erwies sich wegen der implizierten Timing Unterschiede als komplex. Somit bestand die Hauptschwierigkeit beim Aufbau der Additionseinheit im Triggern bzw. dem Steuern des zeitlichen Verlaufs der Schaltung nach dem Drücken des *ziehen* Knopfes. Diese wurde von uns gelöst durch den Einsatz diverser Flankengetriggert (bei positiver Flanke) *Flip-Flops*. Hier kam der 74LS173 zum Einsatz, welcher 4 D-Eingänge (Daten) und (für den Reset-Knopf sehr wichtig) einen Clear Eingang besitzt.

Das Additionselement addiert permanent, das Ergebnis jedoch wird nur kurz nach drücken des Knopfes angenommen (jedoch nachdem die Zufallszahl generiert und weitergeleitet wurde). Die erste Stelle der Siebensegmentanzeige (von rechts) ist direkt mit dem Ausgang der Durchschaltenden Zufallszahl

(74LS173 welcher am Frequenzmodul dran hängt) geschaltet - genau wie die Weiterleitung zum Additionsbaustein (74LS283). Ein besonderes Problem stellte der Carry-Out dar. Als Carry-In Signal wird immer Masse ( $\equiv 0$ ) verwendet. Wenn einmal ein Carry-Out Signal ausgegeben wird, wird dies gespeichert (bleibt dauerhaft größer als  $16 \equiv 2^4$ ) und bleibt bis zum Reset erhalten. Direkt am zweiten 74LS173, welcher nach dem Addierer geschaltet ist und zeitlich getimed (durch Monoflop / 74LS123) das Ergebnis annimmt ist der Ausgang über einen Treiber verstärkt weitergeschaltet zur Anzeige / Logikeinheit.

## 4 Schaltplan

Abbildung 7 zeigt den konzipierten und umgesetzten Schaltplan. Im Prinzip ist der Schaltplan die Realisierung und Verknüpfung der im vorherigen Abschnitt beschriebenen Module. Natürlich kann das Schaltbild beliebig erweitert werden (mögliche Ideen wären Lautsprecher zur akustischen Ausgabe der Spielergebnisse (gewonnen / verloren), oder etwa weitere LEDs). Die am Anfang der Dokumentation erwähnte Erweiterung auf 2 oder mehr Spieler wäre in gewisser Weise analog zu diesem Aufbau möglich, jedoch müssten verschiedene Flipflops geschaltet werden um den aktuellen Spieler zu bestimmen bzw. deren Spielstände zu speichern. Auch die Logik müsste erweitert werden. Diese Aufgabe wäre wohl für einen weiteren Elektronik Kurs oder zur Heimbastelei geeignet.

## 5 Zusammenfassung

Das Projekt konnte im zeitlichen Rahmen von rund 1.5 Arbeitstagen abgeschlossen werden. Die Umsetzung machte uns großen Spaß - auch wenn uns die Aufgabe in der Theorie einfacher vorkam als sie Realität umzusetzen war. Wir hoffen, dass das Projekt trotz der Eingangs genannten Einschränkungen in zufrieden stellendem Maße realisiert und ein wenig von dem Spaß, den wir an der praktischen Umsetzung unserer Idee hatten, vermittelt wurde. Es scheint uns möglich dieses Projekt nochmals mittels modernen „Microcontrollern“ in einem der nächsten Kurse (**Computer- und Microcontroller-Technik** bzw. **Elektronik für Fortgeschrittene**) upgraden zu können um wirklich ein nahezu ideales Blackjack Spiel (mit möglichen Computer-Croupier etc.) bieten zu können.

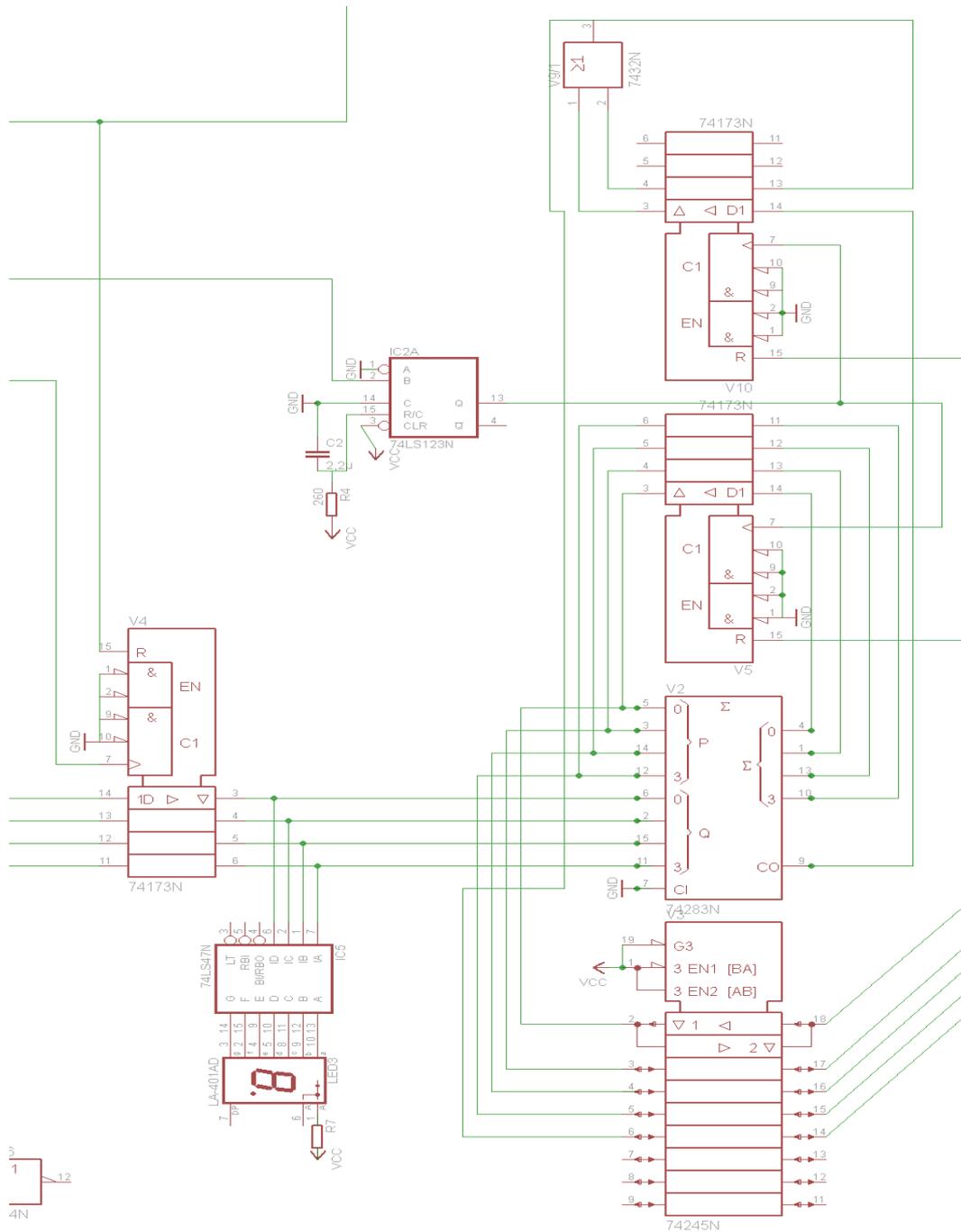


Abbildung 5: Planung der Additionseinheit.

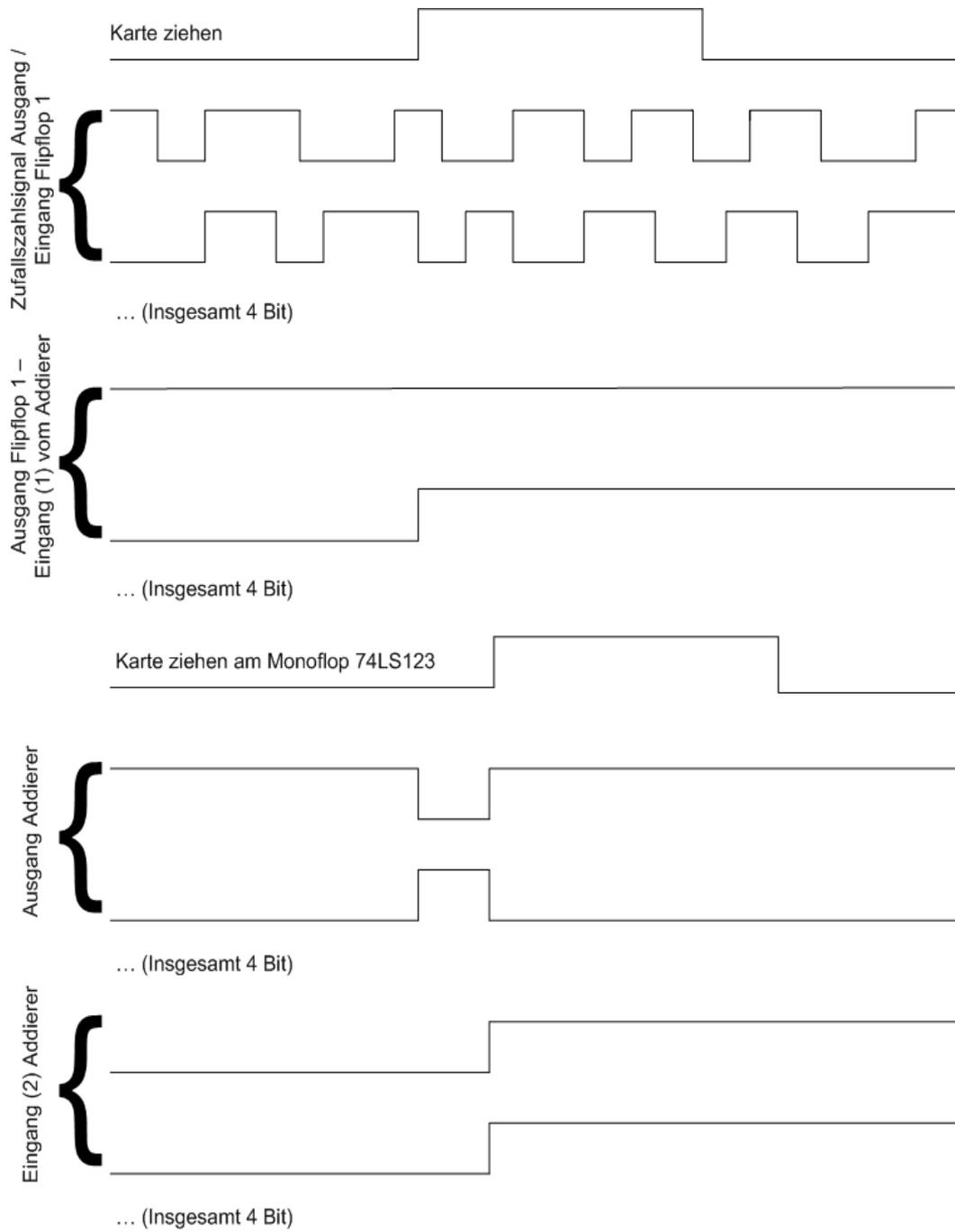


Abbildung 6: Ablaufdiagramm für die Addition (Ziehen einer Karte).

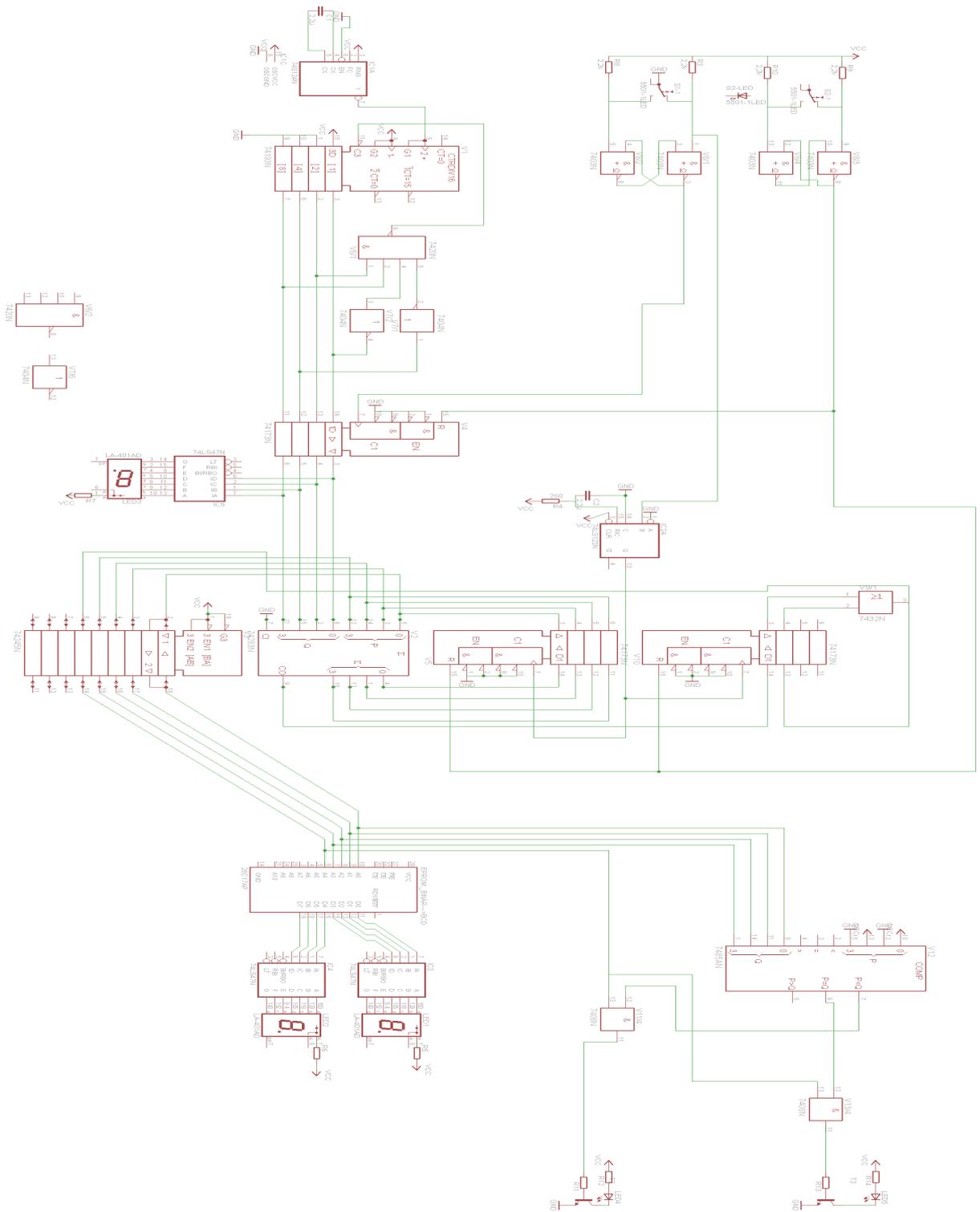


Abbildung 7: Der verwendete Schaltplan - erstellt mit Eagle 5.0.4 Light.