

Praktikum Digitaltechnik SS 2008

Versuchsbeschreibungen

Stand: 18. 5. 08

Gesamtablauf:

1. Termin

- Einführung
- Rechnergestütztes Entwerfen über Schaltplan (Demonstration)
- Praktische Arbeit: herkömmlicher Schaltungsaufbau durch Stöpseln

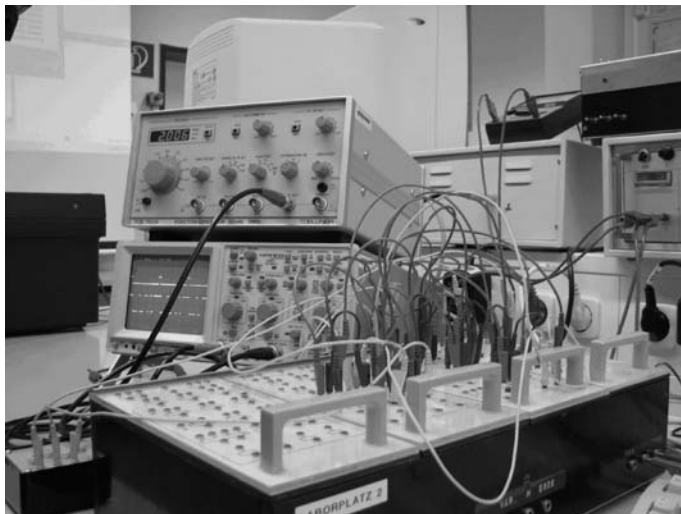
2. Termin: Schaltungsentwicklung auf Grundlage von CPLD-Schaltkreisen (1)

Die Gruppen A und B führen Versuch 3 durch, die anderen Versuch 2 (Stationsausbildung¹).

3. Termin: Schaltungsentwicklung auf Grundlage von CPLD-Schaltkreisen (2)

Die Gruppen A und B führen Versuch 2 durch, die anderen Versuch 3 (Stationsausbildung).

Hinweis: Vor den vielen Aufgaben bitte nicht erschrecken. Was fertig wird, wird fertig ...



1 Eine Maßnahme, um mit dem vorhandenen Bestand an Übungsgeräten auszukommen – wer gedient hat, weiß Bescheid ...

Versuch 1

Versuchsziele:

- Kennenlernen elementarer sequentieller Schaltungen (State Machines, Zähler, Schieberegister),
- Erprobung von Digitalschaltungen im statischen und dynamischen Betrieb,
- Einführung in die Nutzung von Oszilloskop und Logikanalysator.

Versuchsplattform:

Herkömmliche Stecktafeln mit TTL-Schaltkreisen (Abb. 1, 2), Taktgenerator (Signalgenerator), Oszilloskop.

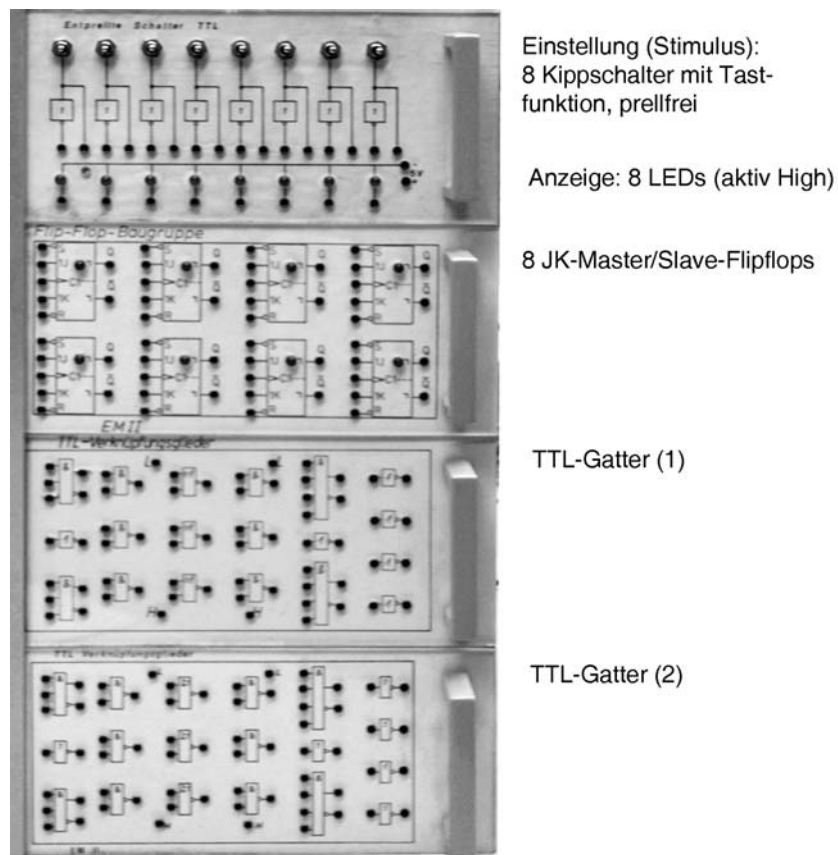


Abb. 1 Stecktafel mit TTL-Schaltkreisen

Hinweis:

Im Versuch darf davon Gebrauch gemacht werden, daß offene TTL-Eingänge so wirken, als wären sie mit einem High-Pegel belegt. Diese Arbeitserleichterung nicht auf ernsthafte Projekte übertragen!

Aufgabe 1: Bauen Sie eine Schaltung zum Nachweisen von Impulsen.

Diese Anordnung (Abb. 3) soll erkennen, ob ein einzelner Impuls, zwei Impulse und mehr, fortlaufende Impulse oder gar kein Impuls ankommt. Funktionsnachweis: mit Schaltern und Taktgenerator. Die Schaltung ist Voraussetzung für die folgende Aufgabe 2.

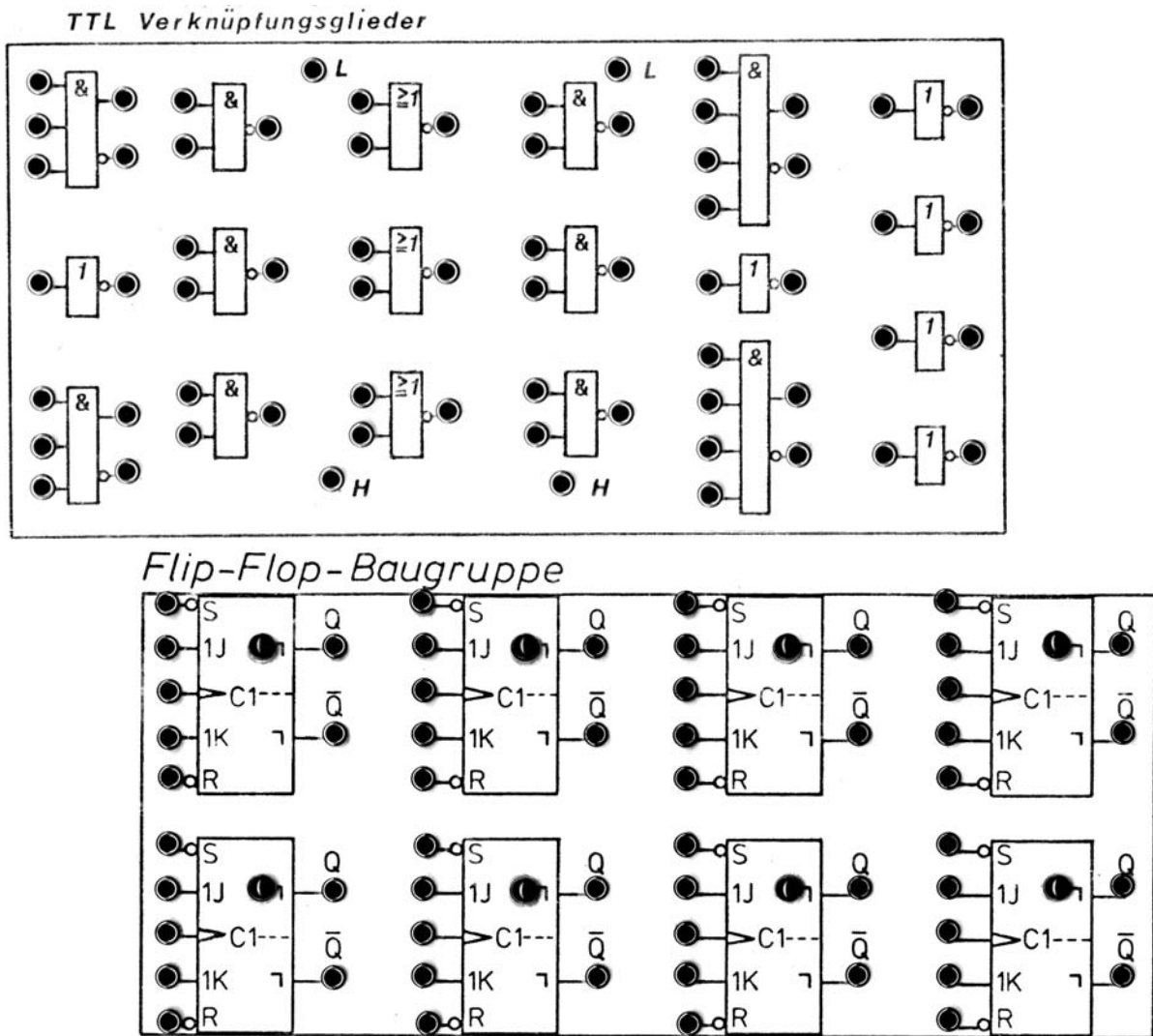


Abb. 2 Stecktafeln. Oben: Gatter. Es stehen zwei derartige Tafeln zur Verfügung. Darunter: JK-Flipflops (eine Tafel). Die Flipflops haben Leuchtdioden zur Zustandsanzeige

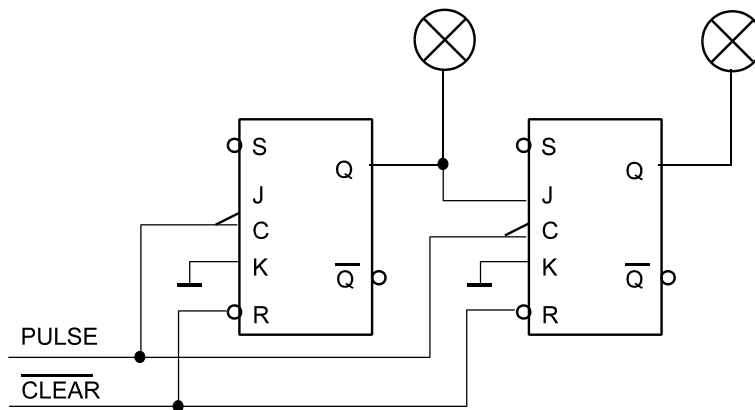


Abb. 3 Impulsnachweis

Aufgabe 2: Bauen Sie einen Single-Shot-Generator mit JK-Flipflops (RS-Schaltweise)

Diese State Machine (Abb. 4 bis 7) soll auf eine Tastenbetätigung (KEY) hin einen einzigen Impuls (PULSE) abgeben. Das Problem: Die Tastenbetätigung dauert viel länger als eine einzelne Taktperiode. Deshalb ist ein Wartezustand (WAIT) einzuführen, in dem auf das Loslassen der Taste gewartet wird.

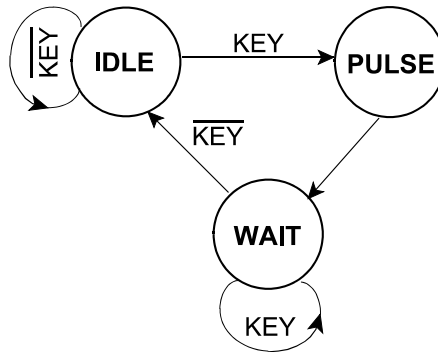


Abb. 4 Single-Shot-Generator (1)

Der IDLE-Zustand wird eingeleitet, wenn im WAIT-Zustand die Taste losgelassen wird.

Er wird verlassen, wenn die Taste betätigt wird.

$$\text{IDLE}_J = \text{WAIT} \cdot \overline{\text{KEY}}$$

$$\text{IDLE}_K = \text{KEY}$$

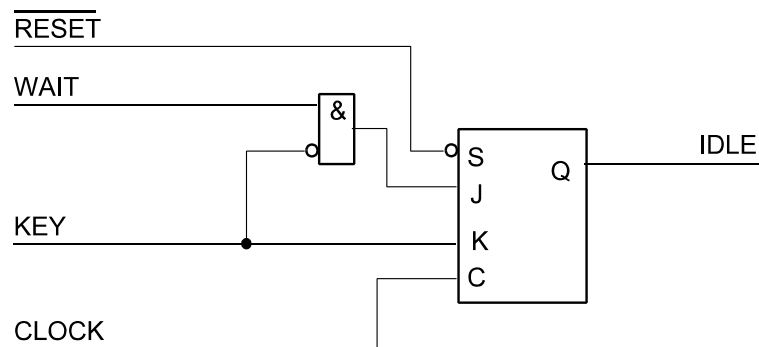


Abb. 5 Single-Shot-Generator (2). IDLE-Zustand

Der PULSE-Zustand wird eingeleitet, wenn im IDLE-Zustand die Taste betätigt wird.

Er wird verlassen, wenn er aktiv ist (= mit dem nächsten Takt).

$$\text{PULSE}_J = \text{IDLE} \cdot \text{KEY}$$

$$\text{PULSE}_K = \text{PULSE}$$

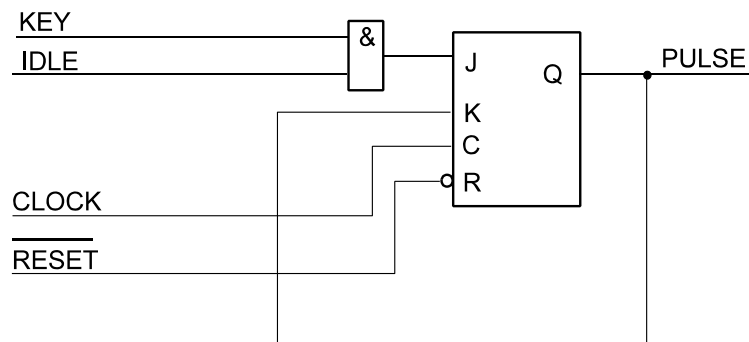


Abb. 6 Single-Shot-Generator (3). PULSE-Zustand

Der WAIT-Zustand wird eingeleitet, wenn der PULSE-Zustand aktiv ist.

Er wird verlassen, wenn die Taste nicht mehr betätigt (= losgelassen) ist.

$$\text{WAIT}_J = \text{PULSE}$$

$$\text{WAIT}_K = \text{WAIT} \cdot \overline{\text{KEY}}$$

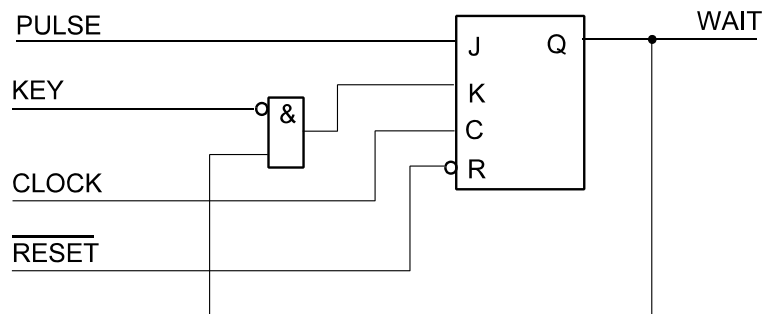


Abb. 7 Single-Shot-Generator (4). WAIT-Zustand

Takt (CLOCK) vom Taktgenerator, Impulsauslösung und anfängliches Rücksetzen (KEY, RESET) vom Schalter. Funktionsnachweis mit Schaltung von Aufgabe 1. Bei jeder Betätigung von KEY darf nur ein einziger Impuls (PULSE) ausgelöst werden.

Bauen Sie nun die Schaltung von Aufgabe 1 ab und verwenden Sie vier der freien Flipflops, um einen einfachen Asynchronezähler zusammenzustecken. Er dient als Stimulusgenerator für die dynamische Prüfung: Taktgenerator an CLOCK des Single-Shot-Generators und an das erste Zähler-Flipflop; Ausgang des letzten Zähler-Flipflops an KEY (anstelle des Schalters). Schaltweise am Oszilloskop beobachten.

Aufgabe 3: Bauen Sie mit drei der JK-Flipflops einen Zähler, der im Graycode modulo 6 zählt (Tabelle 1; Abb. 8).

Die Aufgabe wird zweckmäßigerweise so gelöst, daß die JK-Flipflops als T-Flipflops genutzt werden. Bei Einschalten ist die Anordnung in Stellung 001 zu bringen. Hierzu sind die asynchronen Setz- und Rücksetzeingänge der Flipflops passend zu beschalten (A setzen, C und D rücksetzen).

Stellung	C	B	A
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1

Tabelle 1 Zählen im Graycode

Funktionsnachweis:

- statisch (Takt von Schalter)
- dynamisch (Taktgenerator, Oszilloskop). Z. B. mit Stellung 1 synchronisieren.

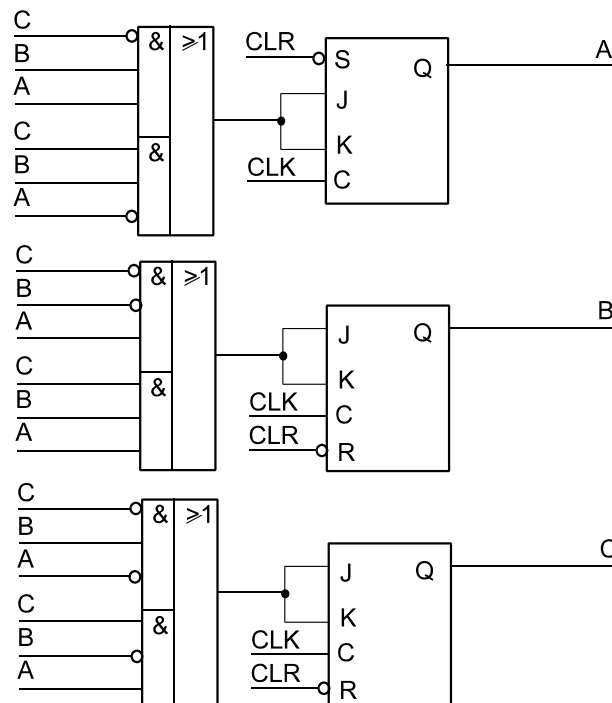


Abb. 8 Zählschaltung für Graycode (modulo 6)

Aufgabe 4: Bauen Sie ein rückgekoppeltes Schieberegister mit maximaler Zykluslänge

Alles bisher Aufgebaute wieder abbauen. Zunächst mit vier Flipflops eine Schiebekette zusammenstecken. Die Rückführung wird durch Antivalenzverknüpfung der Flipflopausgänge C und D gebildet (Abb. 9).

Schaltung zunächst statisch erproben. Was passiert, wenn wir nach dem Löschen Takte auslösen? Nichts – wie zu erwarten war (wenn an den Eingängen einer Antivalenzverknüpfung keine Eins auftritt, kann auch keine herauskommen, so daß im Schieberegister lediglich Nullen umlaufen). Abhilfe: wenigstens ein Flipflop setzen (CLEAR-Leitung vom R-Eingang an den S-Eingang verlegen). Erprobung: statisch und dynamisch (vgl. Aufgabe 3). Achten Sie auf die Periode des Umlaufs (nach wievielen Takten sich das anfänglich eingestellte Muster wiederholt) und auf die Zählweise – es geht bunt durcheinander (Pseudo-Zufallsfolge). Probieren Sie (durch Umstecken des CLEAR-Signals) verschiedene Anfangswerte (Seed Values) durch. Notieren Sie die einzelnen Belegungen, um die Folge der Bitmuster zu erkennen. Es werden immer alle Muster (außer 0H) auftreten. Beim zyklischen Umlauf wiederholen sich die Muster immer wieder in gleicher Folge. Der Anfangswert bestimmt lediglich, an welcher Stelle die Bitmustererzeugung nach dem Rücksetzen beginnt. Wie sieht die Bitmusterfolge aus, wenn anstelle des Flipflops D das Flipflop A in die Antivalenzverknüpfung einbezogen wird?

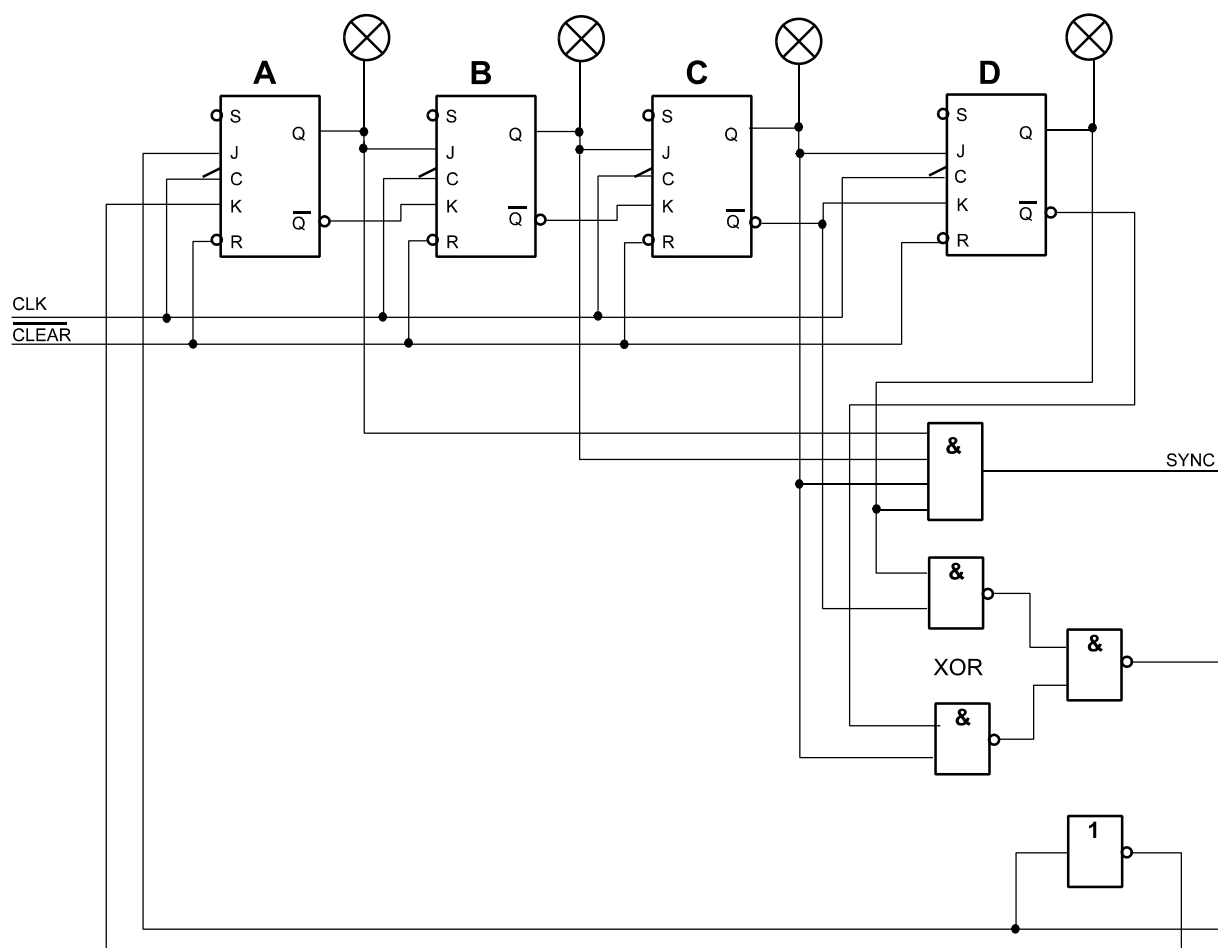


Abb. 9 Ein Schieberegister mit maximaler Zykluslänge. Damit es anschwingt, ist die CLEAR-Leitung bei wenigstens einem Flipflop auf den S-Eingang umzulegen. Da die Belegung 0H (alles Nullen) nicht vorkommt, wird das SYNC-Signal bei Belegung FH (alles Einsen) abgegeben

Versuch 2

Versuchsziele:

- Einstieg in das Entwerfen von Digitalschaltungen auf Grundlage programmierbarer Schaltkreise,
- Kennenlernen des Entwicklungsganges: Schaltplangentwurf am PC – Schaltkreisprogrammierung – Erprobung.

Versuchsplattform:

Entwicklungsumgebung Xilinx ISE 8 / 9, provisorische CPLD-Stecktafel (Stöpselplattform; Abb. 10), Taktgenerator (Signalgenerator), Oszilloskop.

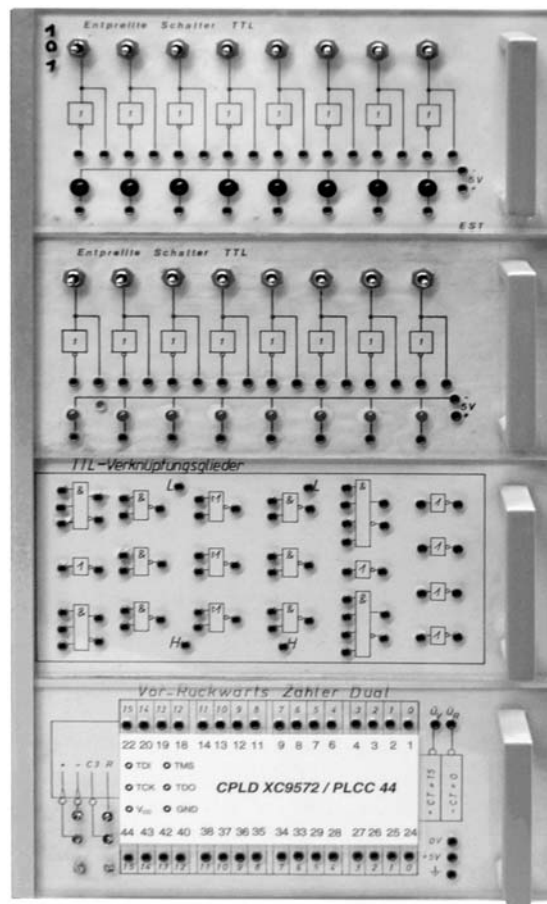


Abb. 10 CPLD-Stecktafel (Provisorium)

Aufgabe 1: Lauflicht auf Grundlage eines selbsteinschwingenden Ringzählers

Eintwerfen Sie einen Ringzähler mit acht Flipflops, der über eine entsprechende Verknüpfung so einschwingt, daß er eine zyklisch umlaufende Eins ausgibt. Funktionsnachweis: mit LEDs bei Ansteuerung mit niedriger Taktfrequenz, so daß sich der Eindruck eines Lauflichts ergibt.

Aufgabe 2: Erweitern Sie die Lauflichtschaltung so, daß das Licht hin- und herläuft. Lassen Sie sich was einfallen ...

Aufgabe 3: Impulsmustergenerator

Entwerfen Sie einen Impulsmustergenerator. Es geht darum, zyklisch eine Impulsfolge zu erzeugen, die über Schalter eingestellt werden kann. Die Stöpselplattform hat 16 Schalter, also können wir uns eine

Zykluslänge von 16 leisten. Die Schalterstellungen werden reihum abgefragt. Bei betätigtem Schalter wird zum jeweiligen Zeitpunkt ein High-Signal ausgegeben, bei unbetätigtem Schalter ein Low-Signal. Abb. 11 zeigt eine Schaltung, die sich mit den vergleichsweise bescheidenen Mitteln aufbauen läßt: einen Binärzähler mit nachgeordnetem Datenselektor (Multiplexer).

Erprobung/Demonstration:

- Oszilloskop,
- Pulsweitenmodulation (PWM),
- Tester für serielle Schnittstelle (Windows-PC und Hyperterminal).

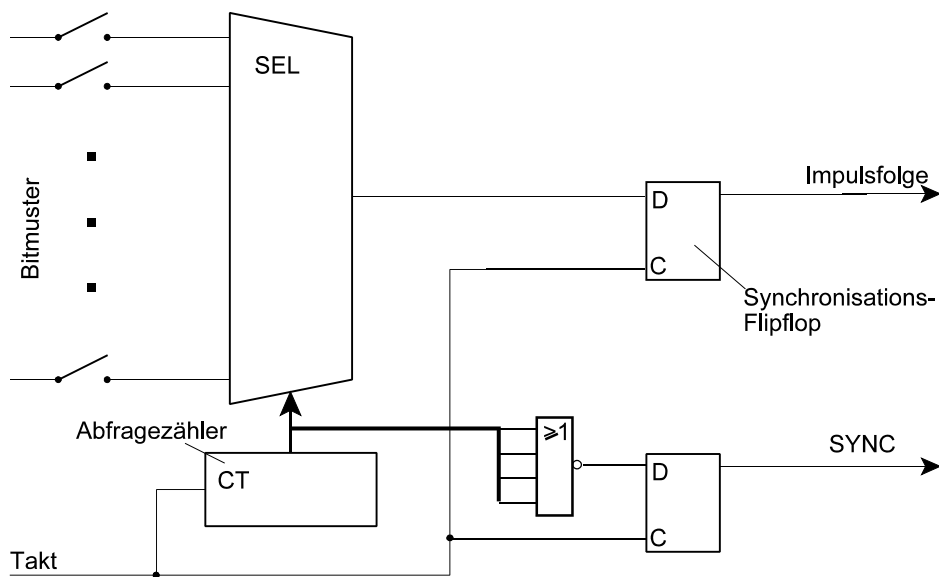


Abb. 11 Impulsmustergenerator

Aufgabe 5: Schnittstellentester

Ergänzen Sie den Impulsmustergenerator so, daß er – bei Auslösung durch eine weitere Taste (KEY) – eine einzige Impulsfolge abgibt (Single Shot). Hierzu brauchen wir eine State Machine ähnlich der von Versuch 1 (Abb. 12).

Erprobung: als Schnittstellentester mit Hyper-Terminal (Windows).

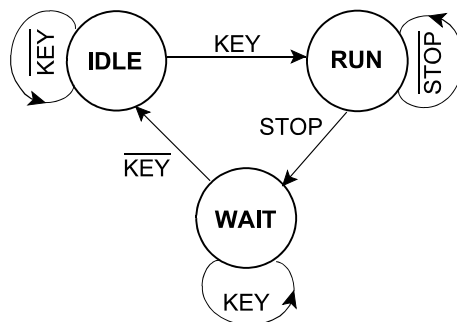


Abb. 12 State Machine zur Steuerung des Impulsmustergenerators. Impulsausgabe im Zustand RUN. STOP wird wirksam, nachdem der letzte Impuls ausgegeben wurde.

Versuch 3

Versuchsziele:

- Entwerfen von Digitalschaltungen auf Grundlage programmierbarer Schaltkreise,
- Anpassung des Schaltkreises an eine vorgegebene externe Beschaltung (Constraints).

Versuchsplattform:

Entwicklungsumgebung Xilinx ISE 8 / 9, CPLD-Übungstafeln (Abb. 13).

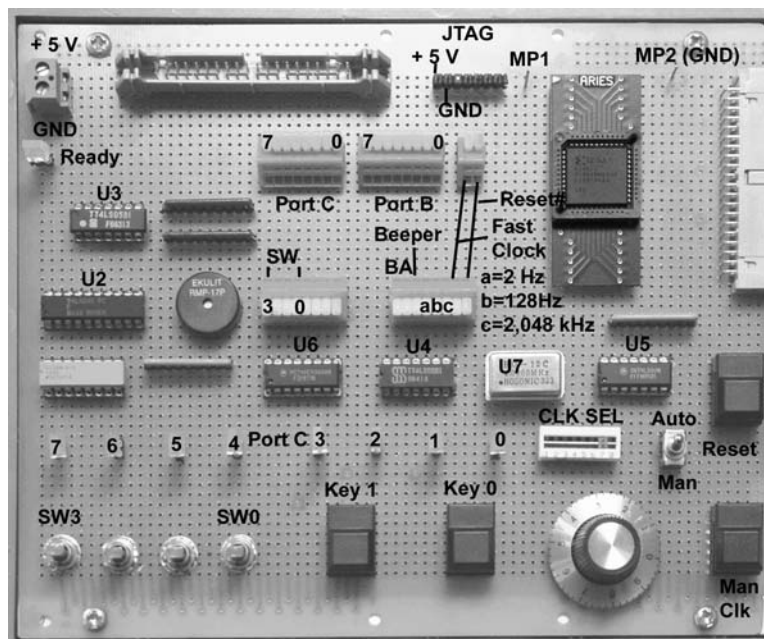
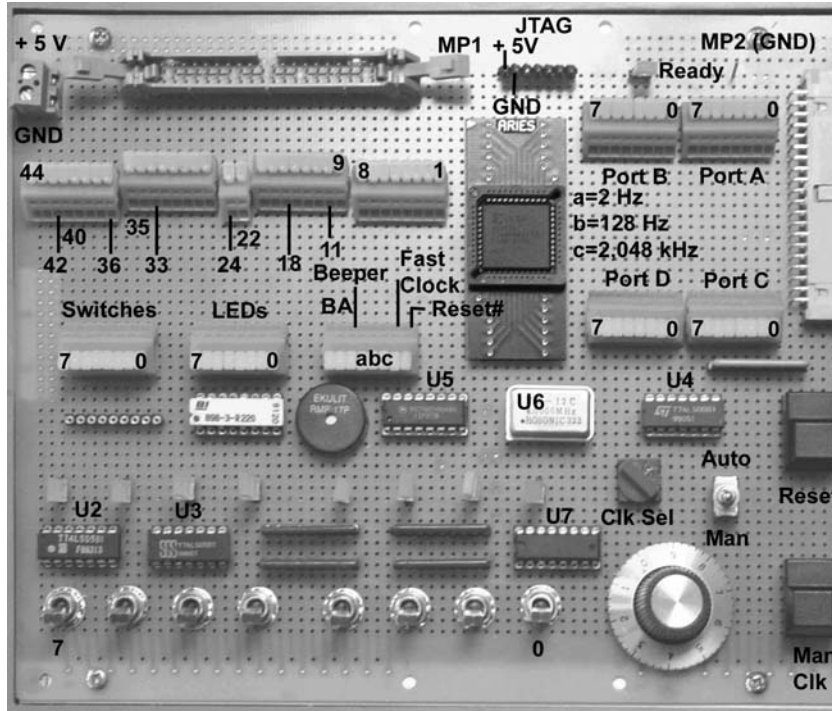


Abb. 13 CPLD-Übungstafeln. Oben Typ 06a, darunter Typ 06b.

Aufgabe 1: Entwerfen Sie einen Siebensegmentdecoder für die BCD-Zahlen 0...9

Zur Erprobung wird ein BCD-Zähler (als fertiges Funktionselement aufzurufen) vorgeschaltet. Hierzu können die in den Digitaltechnik-Übungsstunden erarbeiteten Ergebnisse ausgenutzt werden. Es ist jedoch auch möglich, intuitiv zu entwerfen (Abb. 16) und die Optimierung dem System zu überlassen.

Aufgabe 2: Zweistelliger BCD-Zähler mit Siebensegmentanzeige

Erweitern Sie die Anordnung von Aufgabe 1 auf zwei Dezimalstellen. Sie können dies tun, indem Sie auf Grundlage der Schaltung von Aufgabe 1 einen Funktionsblock erstellen. Womöglich kommen Sie aber schneller zum Erfolg, indem Sie nach dem Prinzip "Kopieren und Einfügen" die Schaltung einfach verdoppeln.

Aufgabe 3: Stoppuhr

Grundlage: die Schaltung von Aufgabe 2. Die Zähl- und Anzeigeschaltung ist um zwei Tasten T1, T2 und die zugehörige Steuerlogik zu erweitern.

Zeitraster: 100 ms (mit näherungsweise passendem Taktsignal der Übungstafel).

Zeitählung: von 0,0 bis 9,9 s. Sind die 9,9 s erreicht, soll die Uhr stehenbleiben.

Taste T1:

1. Betätigung: Zeit läuft (zählen),
2. Betätigung: anhalten (Stop),
3. Betätigung: weiter zählen,
4. Betätigung: anhalten (Stop) usw.

Taste T2:

Alles löschen. Grundzustand.

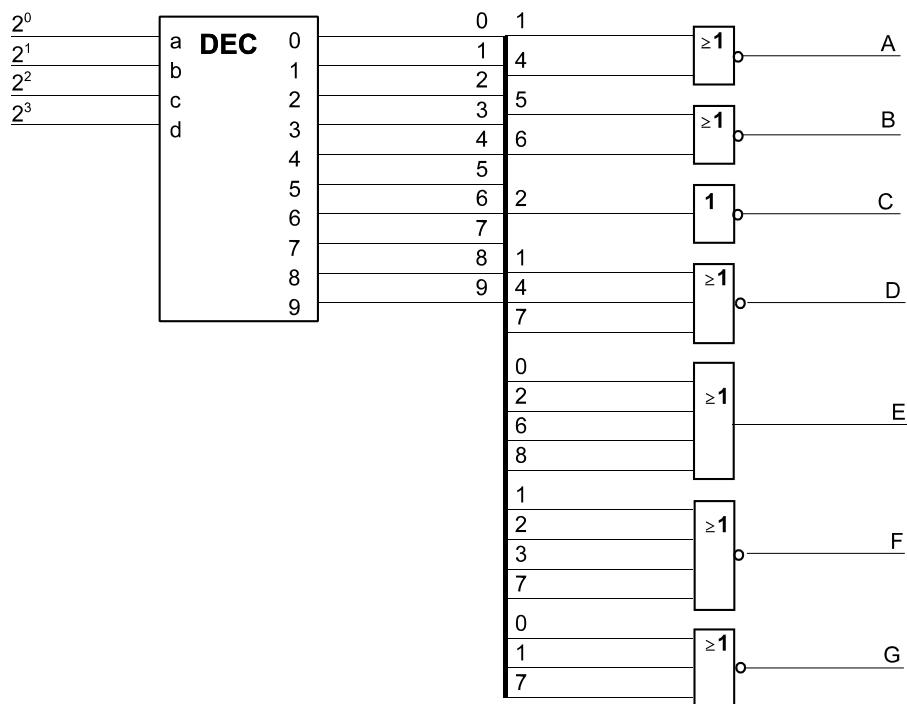


Abb. 14 Ein intuitiv entworfener Siebensegmentdecoder (Prinzipialschaltung). Hinweise: 1. dem BCD-Decoder wird ein Dezimalzähler vorgeschaltet. 2. Die Siebensegmentanzeigen werden aktiv Low angesteuert (alle Ausgangssignale sind zu negieren)

Aufgabe 4: Ziehen von Lottozahlen

Grundlage: die Schaltung von Aufgabe 2. Der Zähler soll zyklisch von 1 bis 49 zählen. Zur Bedienung ist eine Taste vorzusehen.

Taste losgelassen: Zähle angehalten.

Taste niedergehalten: Zähler zählt (Takt von Übungstafel).

Entprellen ist nicht erforderlich. (Denksportaufgabe: Das könnte in *dieser* Anwendung sogar einen Vorteil bedeuten. Weshalb?)

Aufgabe 5: Vorwärts-Rückwärts-Zähler und Incrementalgeber

Entwerfen Sie zunächst einen dezimalen Vorwärts-Rückwärts-Zähler. Er soll zwei Steuereingänge haben: FWD und BWD. Es genügt eine Dezimalstelle.

Lösungsmöglichkeiten:

1. systematischer Zählerentwurf von Grund auf,
2. zwei Vorwärtszähler und Subtraktion,
3. binären Vorwärts-Rückwärts-Zähler von Xilinx nehmen (z. B. cb4x1) und auf Dezimalzählung umbauen,
4. ein dezimaler Addierer für die wahlweise Addition der Festwerte +1 und -1.

Zur Erprobung FWD und BWD über Tasten ansteuern.

Ergänzen Sie dann diese Schaltung um die Auswertung des Incrementalgebers (Abb. 17).

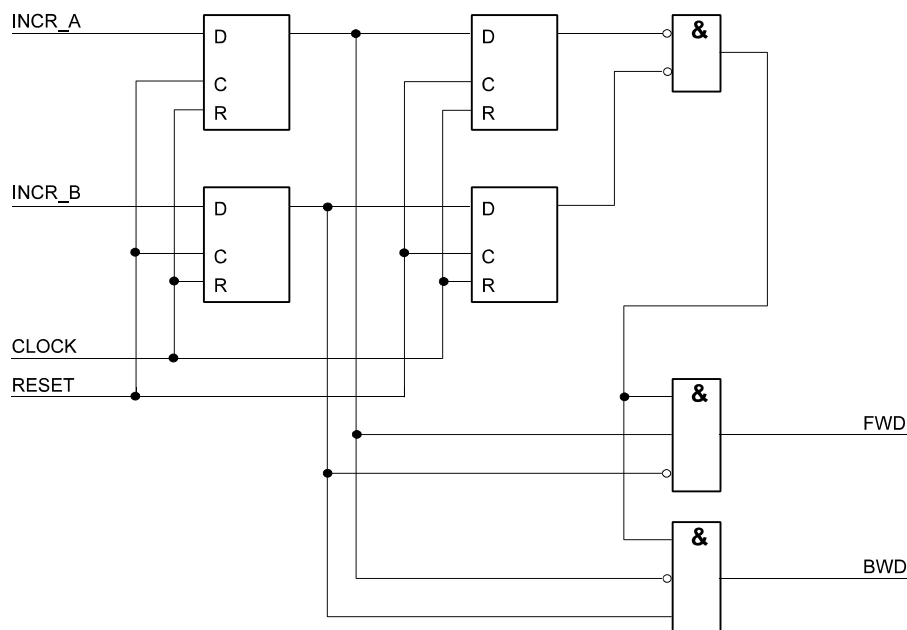


Abb. 15 Auswertung Incrementalgeber (Prinzipschaltung)