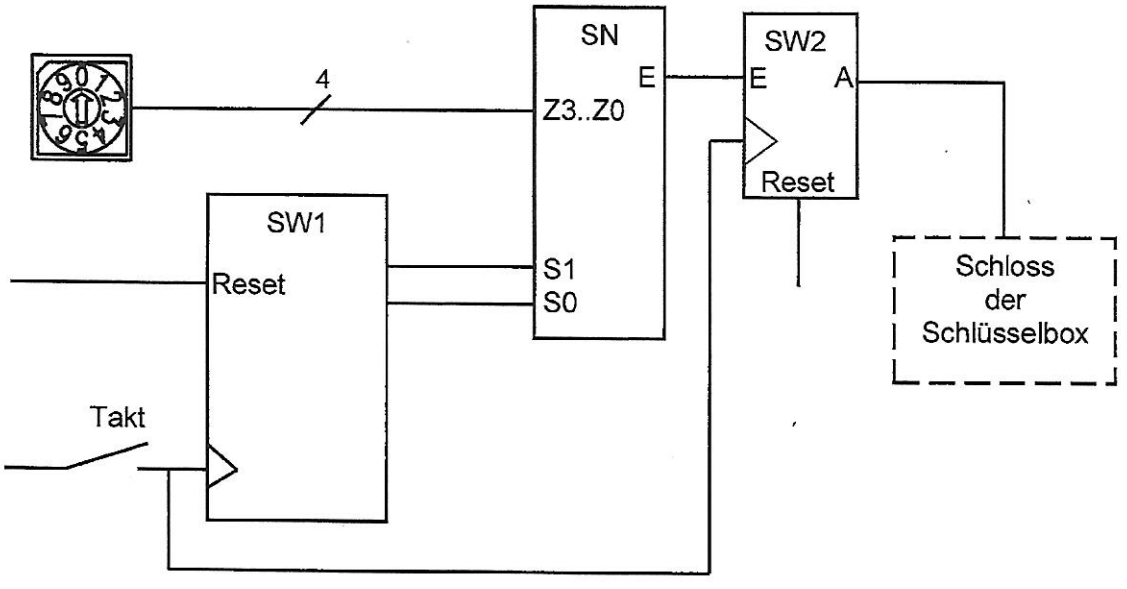
1 Hardware

Ein Vermieter von Ferienhäusern will zur Übergabe des Hausschlüssels eine, mit einem elektronischem Code zu öffnende, Schlüsselbox einsetzen.
Die Schlüsselbox hat ein Schloss mit einem Eingang. Ist der Eingang auf 1 öffnet es sich.
Über einen BCD-Codedrehschalter wird der Code (3-stellig hier: 9 – 0 – 3) Ziffer für Ziffer eingegeben.
Nachdem eine Ziffer eingestellt ist, wird diese mit einer Taste (Takt des Zählers) bestätigt.

Schaltbild:



Die Schaltung besteht aus dem Schaltwerk SW1, welches zählt, wie viele Ziffern eingegeben wurden.

Das Schaltnetz SN überprüft ob die BCD-Zahl korrekt ist. Ist die aktuelle Ziffer korrekt, ist der Ausgang E=1, sonst 0.

Der Ausgang A des Schaltwerks SW2 bleibt auf 0 bis die korrekter Ziffernfolge eingegeben wurde.

Eine neue Codefolge kann erst nach einem RESET eingegeben werden.

Ausgabe des BCD-Codedrehschalters:

Ziffer	Z3	Z2	Z1	Z0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

1.1 Schaltnetz SN

Das Schaltnetz SN liefert am Ausgang E eine 1, wenn die aktuelle Ziffer Z3..Z0 für die aktuelle Position S1, S0 korrekt ist.
Der korrekte Code 9-0-3 ist fest in das Schaltnetz SN eingebaut.
Somit liefert das Schaltnetz bei Eingabe der Position S0=0, S1=0 und Z3...Z0=1001_2 eine 1.

1.1.1. Geben Sie (in Tabellenform) die 3 Eingangsbelegungen für das Schaltnetz SN an, in welchen eine 1 ausgegeben wird. 3

1.1.2. Geben Sie eine Formel in DNF für den Ausgang E an. 3

1.2 Schaltwerk SW1

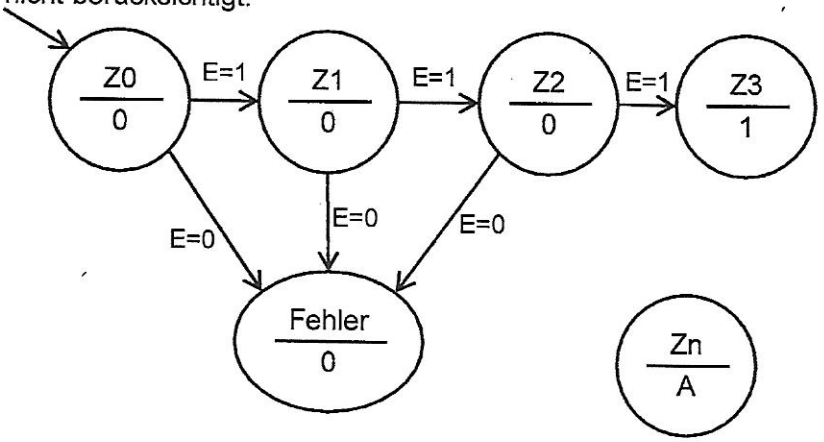
Das SW1 zählt von 0 bis 3. Nach einem Durchlauf soll der Zähler bei dem Wert 3 verweilen und nicht auf 0 zurückgehen.

1.2.1. Wie viele Speicher (Flip-Flops) benötigt der Zähler SW1 mindestens? Begründen Sie ihre Antwort. 2

1.2.2. Erstellen Sie ein Zustandsdiagramm für SW1. 3

1.3. Schaltwerk SW2

Das Schaltwerk SW2 hat neben dem Taktsignal den Eingang E. Im folgenden Zustandsdiagramm ist als Codierung das Ausgabesignal A angegeben.
Ein asynchroner Reset kann das Schaltwerk in Zustand Z0 zurückversetzen – dies wird hier nicht berücksichtigt.



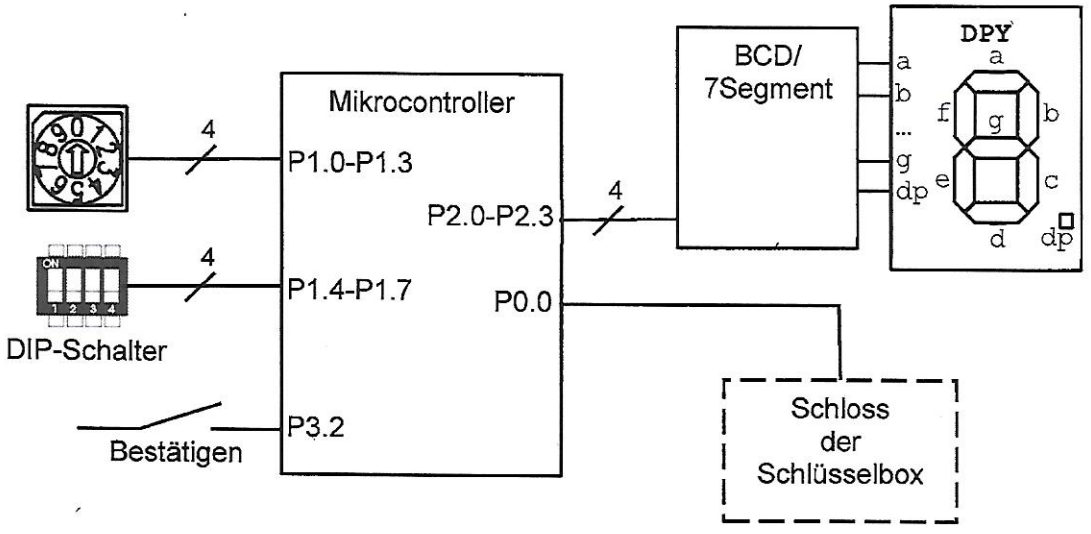
Geben Sie eine Zustandsfolgetabelle für das Schaltwerk an, welche abhängig von aktuellem Zustand und Eingabesignal E den Folgezustand angibt. Sie können für die Zustände die Zustandsnamen (Z0-Z3 und Fehler) oder die binär codierten Zustände verwenden. 4

1.4. Mikrocontroller

Das Codeschloss wird nun mit einem Mikrocontroller realisiert.
Es soll hierbei mit vier DIP-Schaltern (P1.4-P1.7) zwischen mehreren fest hinterlegten dreistelligen Geheimzahlen ausgewählt werden können.
Die Zifferneingabe erfolgt weiterhin über einen BCD-Codedrehschalter (P1.0-P1.3) und eine Bestätigen-Taste (P3.2). Die aktuell eingegebene Ziffer wird an einer 7-Segment-Anzeige ausgegeben.

Der Wert von P1 beinhaltet somit in den höherwertigen 4 Bit die Nummer der gewählten Geheimzahl und in den niederwertigen 4 Bit die aktuell eingegebene Ziffer.

Schaltbild:

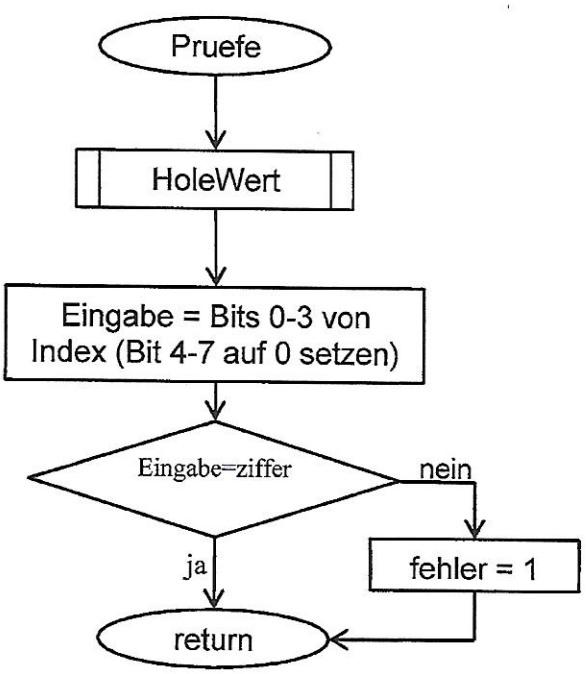


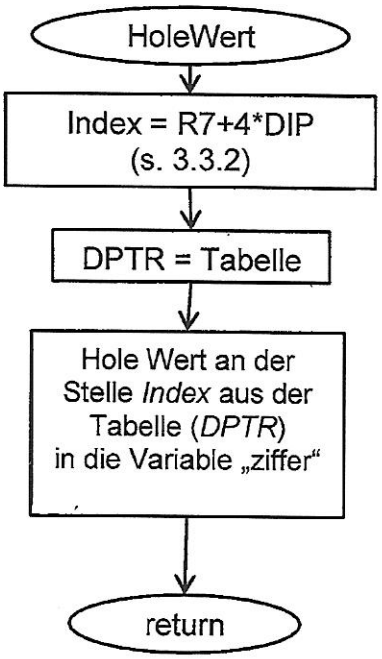
1.4.1. Geben Sie an, wie viele unterschiedliche Geheimzahlen über die 4 Dip-Schalter eingestellt werden können. 3
Beurteilen Sie welcher Speicher (RAM oder ROM) zur Speicherung der Geheimzahlen am besten geeignet ist.

1.4.2. Das Hauptprogramm initialisiert die Zählvariable „count" für die Zifferneingabe mit 0 und die Variable „fehler" mit 0. 4
Danach gibt es die eingestellte Ziffer (P1.0-P1.3) an der 7-Segment-Anzeige (P2.0-P2.3) aus, bis der Taster „Bestätigen" ein LOW-Signal liefert.
Wird „Bestätigen" gedrückt wird das Unterprogramm ***Pruefe*** aufgerufen und die Zählvariable „count" hochgezählt.
Ist „count" kleiner 3 springt man zurück zur Anzeige der aktuellen Ziffer.
Ist die Zählvariable „count" auf 3 (alle 3 Ziffern wurden eingegeben), so wird das Unterprogramm ***Oeffne*** aufgerufen, falls „fehler" noch auf 0 ist.
Danach verweilt das Programm in einer Endlosschleife.
Zeichnen Sie den PAP für das Hauptprogramm. Die genannten Unterprogramme dürfen verwendet werden, sind aber nicht auszuformülieren.

1.4.3. Folgende PAPs zeigen den Ablauf der Unterprogramme ***Pruefe*** und ***HoleWert***.

Pruefe holt den korrekten Code-Wert mittels des Unterprogramms ***HoleWert*** aus einer Tabelle. Der Wert wird in der Variable „ziffer" gespeichert (diese liegt im RAM).
Danach maskiert ***Pruefe*** die Eingabe an Port 1 so aus, dass der Input der DIP-Schalter gelöscht wird um die Eingabe direkt mit dem Tabellenwert in R0 vergleichen zu können.
Schlägt der Vergleich fehl wird eine Variable „fehler" auf 1 gesetzt.





1.4.3.1. Geben sie den Code des Unterprogramms ***Pruefe*** in Assembler an. Wenn sie statt der Namen Adressen verwenden geben sie eine Zuordnung von den Namen im PAP zu den gewählten Adressen an. 4

1.4.3.2. Schreiben Sie den Assembler-Code für das Unterprogramms ***HoleWert***. Die Tabelle liegt an der Marke „tabelle". 4
In der Tabelle liegen immer 4 Byte für jeden Code hintereinander.
Die 4 Byte enthalten Ziffer 0, Ziffer 1, Ziffer 2 und abschließend den Wert FFh für das Codeende.

R7 enthält die aktuelle Ziffernposition. Zu R7 muss der mit 4 multiplizierte DIP-Schalter-Wert addiert werden, um die richtige Tabellenposition zu ermitteln.
Bsp.: Sei R7 = 2 und der DIP-Schalter Wert = 1 dann ist die Tabellenposition 2+4*1=6

in der Beispieltabelle unten steht an Position 6 die Ziffer 7.

Einen Tabelle mit den Code Nr. 0: 9-0-3, Code Nr. 1: 2-4-7, ... hat folgenden Aufbau:

	Code 0				Code 1				...
tabelle: db	**09h,**	**00h,**	**03h,**	**FFh,**	**02h,**	**04h,**	**07h,**	**FFh,**	**...**
Ziffern-Nr.	0	1	2	Codeende	0	1	2	Codeende	
Position:	0	1	2	3	4	5	6	7	...

30