

Aufgabe 1: Decoder mittels Tabellenspeicher

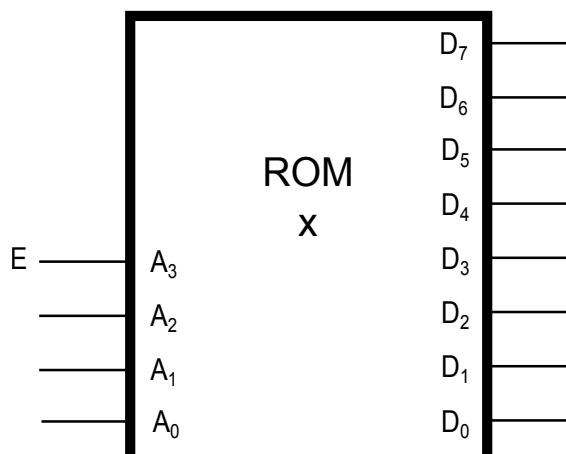
Entwerfen Sie mittels Tabellenspeicher einen 3 zu 8 Decoder mit Enable-Leitung E , den drei Eingängen e_2 bis e_0 und den acht Ausgängen a_7 bis a_0

Die Realisierung soll durch einen entsprechenden ROM-Baustein erfolgen.

- a) Vervollständigen Sie die nachfolgende Speichertabelle eines ROM-Bausteins.

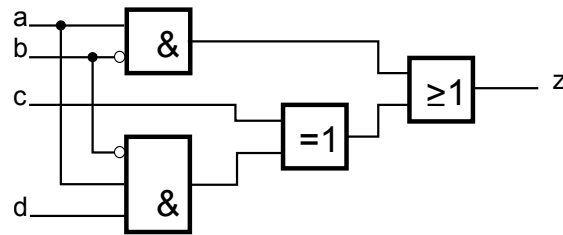
A_3 E	A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0								
0	0	0	1								
0	0	1	0								
0	0	1	1								
0	1	0	0								
0	1	0	1								
0	1	1	0								
0	1	1	1								
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1								
1	1	0	0								
1	1	0	1								
1	1	1	0								
1	1	1	1								

- b) Vervollständigen Sie das zugehörige Schaltbild des ROM-Bausteins. Achten Sie auf eine korrekte Zuordnung von Adress- und Datenleitungen und beschriften Sie diese entsprechend.



Aufgabe 2: Realisierung mit NAND-Gattern

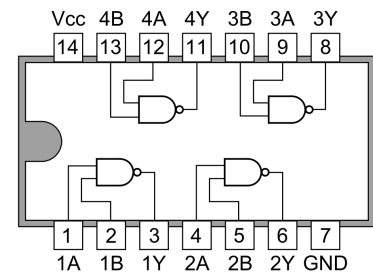
Gegeben ist die folgende Schaltung:



- Transformieren Sie die vorgegebene Schaltung so, dass nur mehr Grundgatter (AND, OR und NOT) mit maximal 3 Eingängen verwendet werden und zeichnen Sie das entsprechende Schaltnetz.
- Für den Aufbau des Schaltnetzes soll der abgebildete IC-Typ 7400 verwendet werden. Die Schaltung muss daher mit NAND-Gattern realisiert werden.

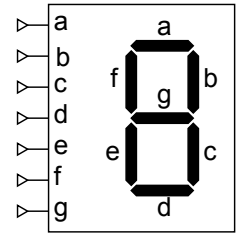
Formen Sie die transformierte Schaltung aus Teilaufgabe a) *graphisch* um, sodass ausschließlich NAND-Gatter verwendet werden. Zeichnen Sie das entsprechende Schaltnetz mit europäischen Schaltsymbolen. Geben Sie außerdem an, wie viele ICs des abgebildeten Typs für Ihre Schaltung benötigt werden. (Link zum Datenblatt für interessierte Studierende: <http://www.ti.com/lit/ds/symlink/sn7400.pdf>)

7400 Quad 2-input NAND Gates



Aufgabe 3: Entwurf & Realisierung einer Boole'schen Funktion

Entwerfen Sie einen Teil eines Binär-zu-BCD Decoder mit den Eingängen e_3, e_2, e_1 und e_0 , der die dargestellte Siebensegmentanzeige ansteuert. Die Eingänge stellen dabei eine Binärzahl (e_3 ist das MSB) dar und auf der Siebensegmentanzeige sollen die Ziffern von 0 bis 9 dargestellt werden können. Ist die Binärzahl größer als $(9)_{10}$, so soll kein Segment leuchten. Ein Segment der Anzeige leuchtet, wenn am zugehörigen Eingang (a-g) logisch 1 anliegt.

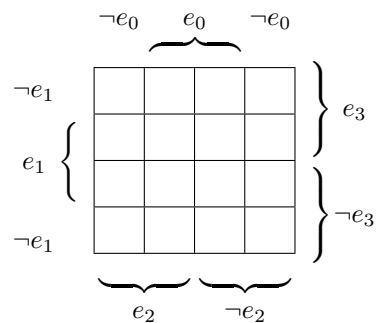


- a) Stellen Sie die Ziffern von 0 bis 9 graphisch dar und achten Sie dabei auf die Unterscheidbarkeit der einzelnen Symbole.

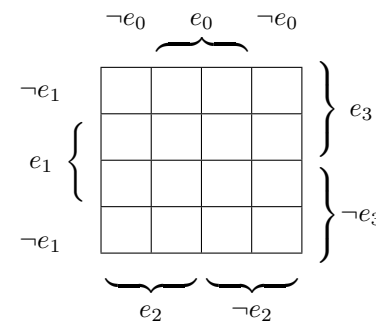
- b) Realisieren Sie eine **minimale** Schaltung für die Segmente "a" und "b" im unten dargestellten PLA.

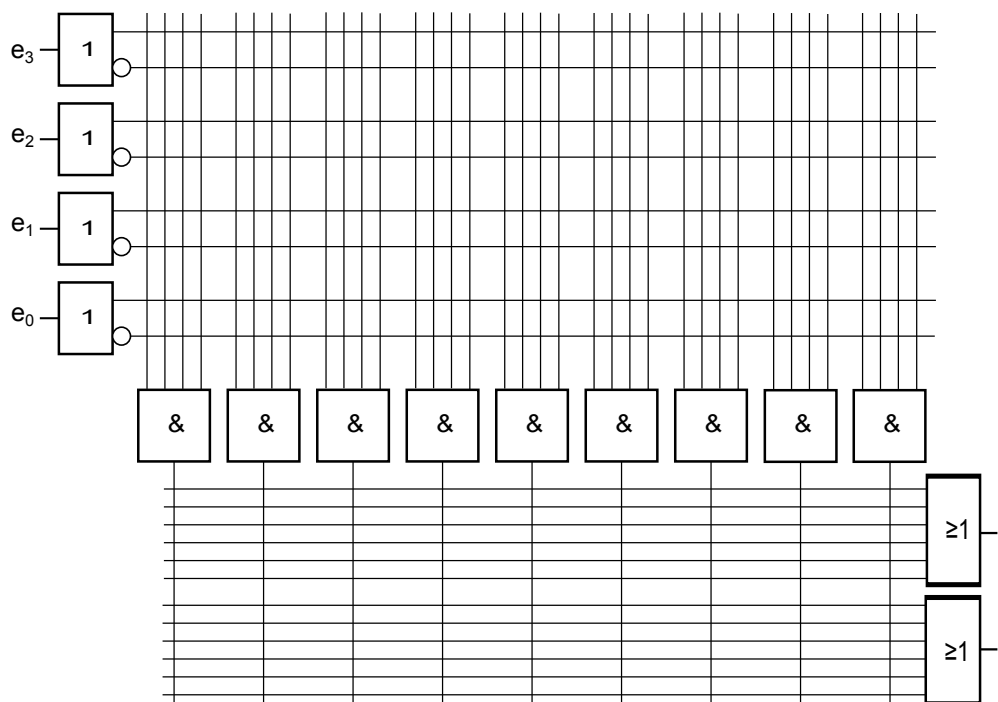
e_3	e_2	e_1	e_0	a	b
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

Ausgang "a":



Ausgang "b":



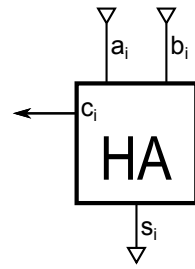


Aufgabe 4: 4-Bit Addierer

Entwerfen Sie einen Addierer, der eine fünfstellige Summe $S = (s_4 s_3 s_2 s_1 s_0)_2$ zweier vierstelliger Binärzahlen $A = (a_3 a_2 a_1 a_0)_2$ und $B = (b_3 b_2 b_1 b_0)_2$ bildet. s_0 , a_0 und b_0 sind jeweils das *least significant bit (lsb)*.

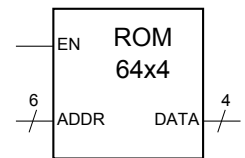
Für die Realisierung können Sie beliebig viele 1-Bit-Halbaddierer (siehe Abbildung rechts) verwenden, andere Bauteile – wie beispielsweise AND-, OR-, NOT-Gatter – stehen Ihnen nicht zur Verfügung.

Zeichnen Sie die resultierende Schaltung. Vergessen Sie nicht, Ein- und Ausgänge entsprechend zu beschriften.

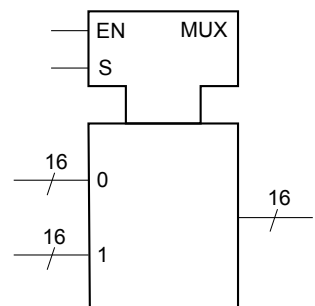


Aufgabe 5: ROM-Erweiterung

Ein 128x16 ROM soll aus mehreren 64x4 ROMs (vgl. Abbildung rechts: Enable-Leitung EN, Adressleitungen ADDR, Datenleitungen DATA) aufgebaut werden.



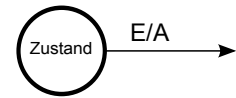
- Wie viele ROM-Bausteine benötigen Sie für den Aufbau des 128x16 ROMs?
- Wie viele Adress- bzw. Datenleitungen benötigt der 128x16 ROM?
- Wie viele Bytes Daten können im 128x16 ROM insgesamt gespeichert werden?
- Skizzieren Sie den Aufbau des 128x16 ROM, bestehend aus der entsprechenden Anzahl von 64x4 ROM-Bauteilen und dem unten vorgedruckten (2 zu 1) Multiplexer. Weitere Bauteile (Decoder, NOT-Gatter, etc.) stehen Ihnen nicht zur Verfügung. Achten Sie auf eine korrekte Beschriftung der Leitungen.



Aufgabe 6: Mustererkennung

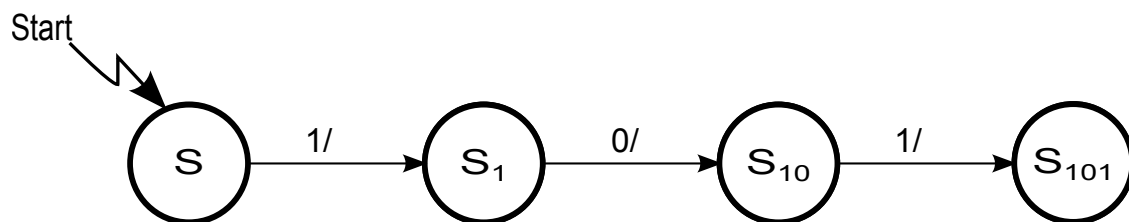
Gegeben ist das unten abgebildete Grundgerüst eines Mealy-Automaten mit einem Eingang E und einem Ausgang A , der eine beliebig lange Bitfolge einliest.

Vervollständigen Sie den Automaten so, dass am Ausgang immer dann logisch '1' ausgegeben wird, wenn der Automat die Teilfolge '1011' erkennt (Bit-Reihenfolge links nach rechts). In allen anderen Fällen soll '0' ausgegeben werden. Verwenden Sie dabei die rechts angeführte Notation.



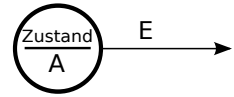
Hinweis: Beachten Sie, dass sich aufeinanderfolgende '1011'-Folgen überlappen können.

Beispiel: $E = 1110110111011110\dots$
 $A = 0000010010001000\dots$



Aufgabe 7: Entwurf eines Zustandsgraphen

Zeichnen Sie den Zustandsgraphen eines Moore-Schaltwerks, das der nachfolgenden Beschreibung entspricht. Verwenden Sie die rechts angeführte Notation.



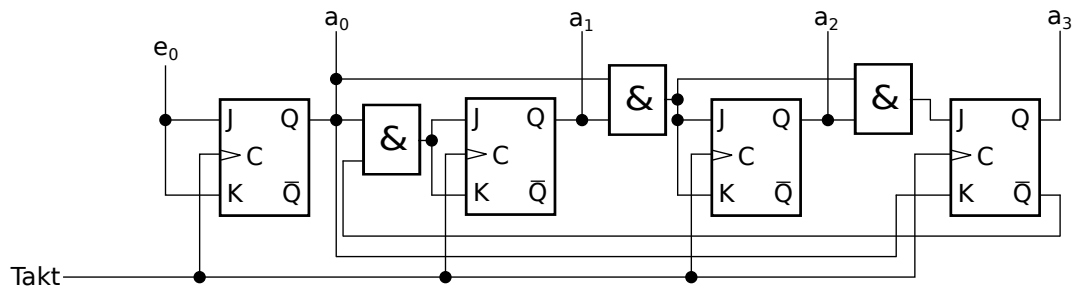
Das Schaltwerk soll am Ausgang A logisch '1' ausgeben, wenn die Teilfolge '1011' am Eingang E eingelesen wurde, anderenfalls soll '0' ausgegeben werden (Bit-Reihenfolge links nach rechts). Aufeinanderfolgende '1011'-Folgen können sich überlappen.

Beispiel:

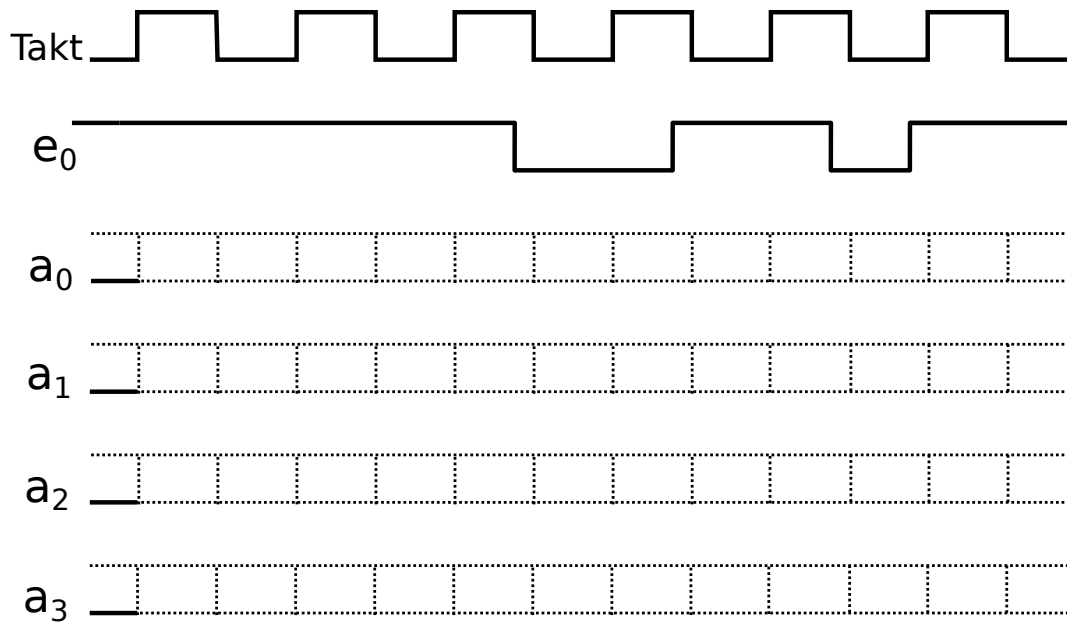
E	=	1110110111011110...
A	=	0000001001000100...

Aufgabe 8: Timing-Diagramm

Es ist folgende Schaltung gegeben:

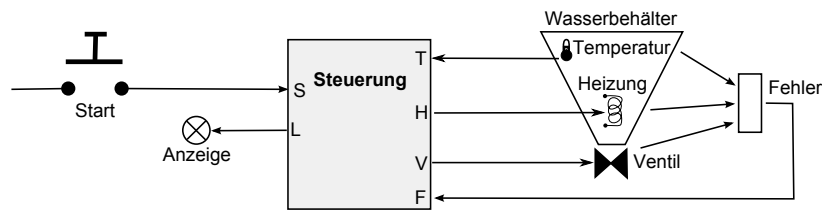


Überlegen Sie sich die Funktionsweise der Schaltung, sodass Sie diese in der Übung erklären können und vervollständigen Sie das nachfolgende Timing-Diagramm. Achten Sie dabei auf die Flankentriggerung.



Aufgabe 9: Schaltwerksentwicklung – Kaffeemaschine / Teil 1

Entwerfen Sie einen Mealy-Automaten für die getaktete Steuerung einer Kaffeemaschine. Der schematische Aufbau dieser Kaffeemaschine ist der folgende:



Die Zustandsübergänge erfolgen ausschließlich zum Takt. Nach dem Einschalten befindet sich die Steuerung der Kaffeemaschine solange im Zustand "Standby", bis die Start-Taste gedrückt wird. Die Steuerung wechselt dann für mindestens einen Takt in den Zustand "Heizen" und sendet ein entsprechendes Signal an die Heizung um das Wasser zu erhitzen. Ein Temperatursensor misst die Temperatur des Wassers und sendet ein Signal, sobald das Wasser die notwendige Mindesttemperatur überschritten hat. Sobald die Mindesttemperatur erreicht oder überschritten wurde, wechselt die Maschine in den Zustand "Zubereiten" und die Steuerung gibt über ein Ventil das erhitzte Wasser frei. Die Zubereitung des Kaffees dauert genau einen Takt lang. Danach wird die Heizung abgeschaltet, das Ventil geschlossen und die Steuerung kehrt in den "Standby"-Modus zurück.

Die Kaffeemaschine verfügt über Sensoren zur Erkennung von Fehlern (z.B. wenn der Wassertank leer ist, ein Defekt im Heizsystem erkannt wird oder der Filter verstopft ist). Falls ein Fehler auftritt, wechselt der Automat mit dem nächsten Takt in den Zustand "Fehler" und verharrt dort, bis der Fehler beseitigt wurde. Solange sich der Automat im Zustand "Fehler" befindet, sollen die Heizung abgeschaltet und das Ventil geschlossen sein. Wurde der Fehler behoben, kehrt die Kaffeemaschine wieder in den Zustand "Standby" zurück.

Die Kaffeemaschine besitzt eine Anzeige-LED, deren Farbe entweder auf rot oder grün geschaltet werden kann. Solange kein Fehler vorliegt, soll die LED grün leuchten. Tritt ein Fehler auf, soll sie rot leuchten.

Ein Betätigen der Start-Taste in den Zuständen "Heizen", "Zubereiten" oder "Fehler" hat keine Auswirkungen und wird ignoriert.

Die Steuerung besitzt 3 Eingänge S , T und F sowie 3 Ausgänge L , H und V mit folgender Bedeutung:

- Das Signal S liefert den Zustand der Start-Taste. Logisch '0' bedeutet, dass diese gedrückt wurde.
- Der Eingang T liefert logisch '1', wenn die Wassertemperatur die notwendige Mindesttemperatur erreicht hat oder darüber liegt.
- Am Eingang F liegt bei Vorliegen eines Fehlers logisch '0' an, anderenfalls logisch '1'.
- Liegt am Ausgang H logisch '1' an, ist die Heizung eingeschaltet. Bei logisch '0' ausgeschaltet.
- Über das Signal V wird durch logisch '1' das Ventil geöffnet, durch logisch '0' geschlossen.
- Das Signal L steuert die Anzeige-LED. Bei logisch '1' leuchtet die LED rot, bei logisch '0' grün.

Entwerfen und zeichnen Sie einen entsprechenden Mealy-Automaten. Verwenden Sie dabei die nebenstehende Notation:

