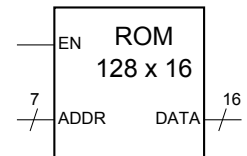


Aufgabe 1: ROM-Erweiterung

Ein 256×64 ROM soll aus mehreren 128×16 ROMs (vgl. Abbildung rechts: Enable-Leitung EN, Adressleitungen ADDR, Datenleitungen DATA) aufgebaut werden.

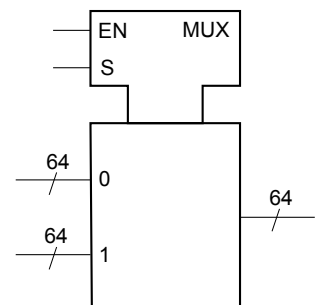


- a) Wie viele ROM-Bausteine benötigen Sie für den Aufbau des 256×64 ROMs?

- b) Wie viele Adress- bzw. Datenleitungen benötigt der 256×64 ROM?

- c) Wie viele Bytes Daten können im 256×64 ROM insgesamt gespeichert werden?

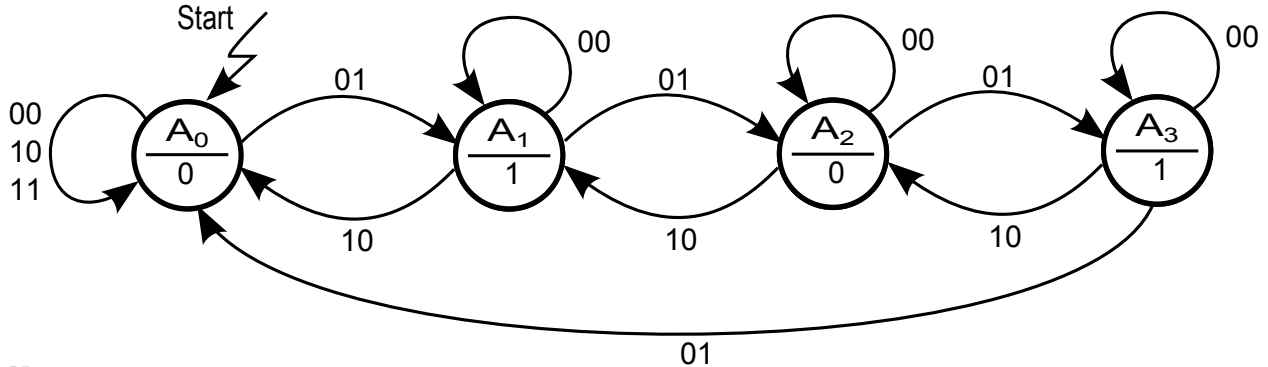
- d) Skizzieren Sie den Aufbau des 256×64 ROM, bestehend aus der entsprechenden Anzahl von 128×16 ROM-Bauteilen und dem unten vorgedruckten (2 zu 1) Multiplexer. Weitere Bauteile (Decoder, NOT-Gatter, etc.) stehen Ihnen nicht zur Verfügung. Achten Sie auf eine korrekte Beschriftung der Leitungen!



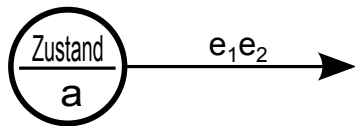
Allgemeiner Hinweis: Zwecks Vereinfachung entfällt bei den nachfolgenden Aufgabenstellungen mit Automaten die Kennzeichnung von Endzuständen – es können jeweils alle Zustände als gültige Endzustände betrachtet werden. Startzustände sind aber immer zu kennzeichnen!

Aufgabe 2: Zustandsgraph – Wahr oder falsch?

Gegeben ist der folgende Automat:



Notation:



Welche Aussagen treffen zu? Begründen Sie Ihre Antwort!

(1) Beim dargestellten Automaten handelt es sich um einen Moore-Automaten.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(2) Der dargestellte Automat ist vollständig und deterministisch.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(3) Der dargestellte Automat ist endlich.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(4) Der dargestellte Automat hat unendlich viele Zustände.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(5) Der dargestellte Automat verfügt über zwei Eingänge und vier Ausgänge.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(6) Sobald der Automat gestartet wurde gibt er '0' aus.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(7) Wird im Zustand A_0 '00' eingelesen, wird '10' ausgegeben.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(8) Liegt nach dem Start an beiden Eingängen einen Takt lang '0' an, befindet sich der Automat jedenfalls im Zustand A_1 .	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(9) Wenn drei Takte lang '10' am Eingang anliegt, befindet sich der Automat danach immer im Zustand A_0 .	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(10) Liegt nach dem Start am Eingang e_2 konstant '0' an, verharrt der Automat jedenfalls im Zustand A_0 .	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(11) Die Zustände des dargestellten Automaten können bei 1 aus n-Codierung mit einem Flip-Flop realisiert werden.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch
(12) Der dargestellte Automat kann minimiert und bei dichter Codierung mit einem Latch realisiert werden.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch

Aufgabe 3: Entwurf eines Zustandsgraphen

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



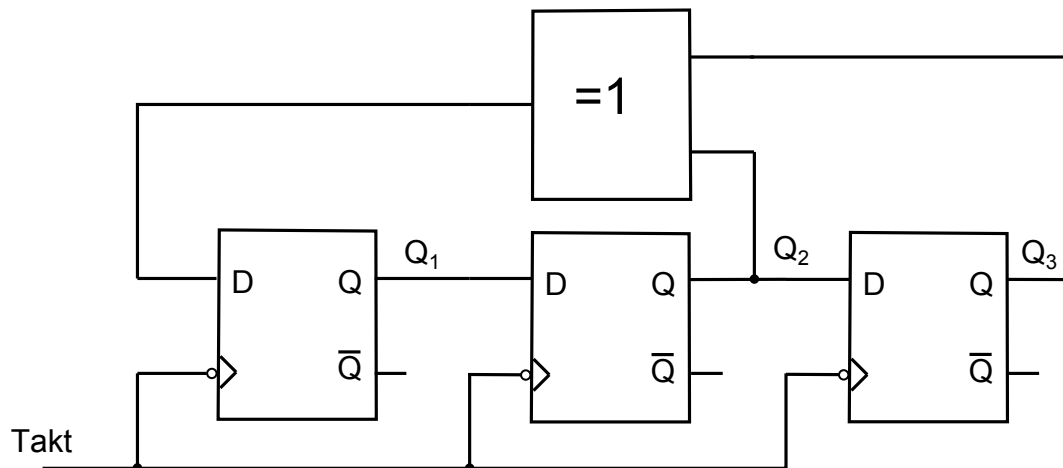
Das Schaltwerk soll am Ausgang A logisch '1' ausgeben, wenn die Teilfolge '0101' am Eingang E eingelesen wurde, anderenfalls soll '0' ausgegeben werden. Aufeinanderfolgende '0101'-Folgen können sich überlappen.

Beispiel:

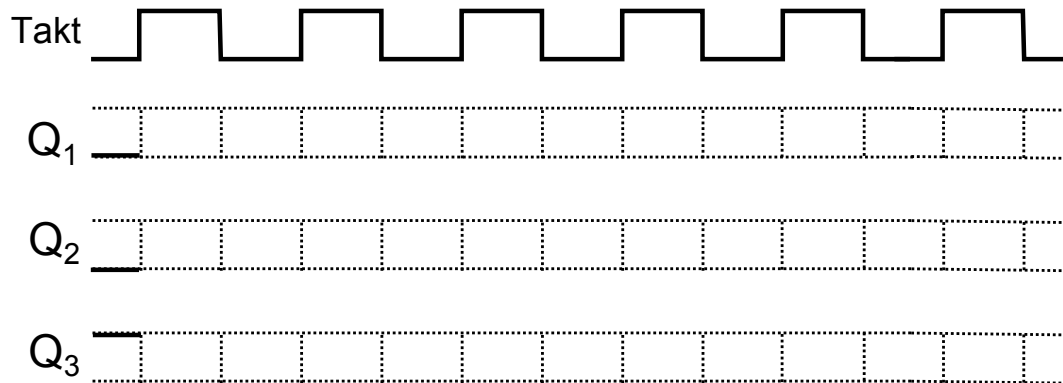
E	=	11101010110101110...
A	=	00000010100001000...

Aufgabe 4: 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 5: Maximale Taktfrequenz

Berechnen Sie die maximale Taktfrequenz für das Schaltwerk aus Aufgabe 4! Es gelten folgende Werte:

Durchlaufzeit XOR-Gatter: 15 ns

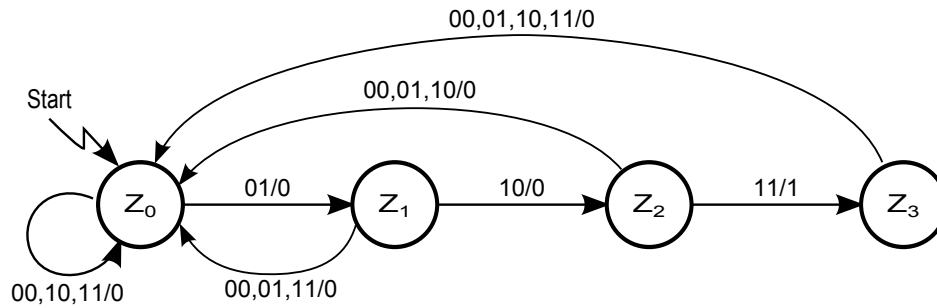
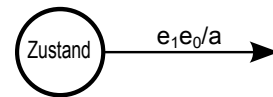
Durchlaufzeit D-Flip-Flop: 50 ns

Vorbereitungszeit D-Flip-Flop: 5 ns

Haltezeit D-Flip-Flop: 3 ns

Maximale Taktfrequenz D-Flip-Flop: 10 MHz

Der Zustandsgraph eines Schaltwerks ist gegeben, es gilt folgende Notation:



- a) Analysieren Sie den Zustandsgraphen. Um welche Art von Schaltwerk handelt es sich? Welche Funktion realisiert dieses Schaltwerk, wenn die Eingänge als zweistellige Binärzahl $(e_1 e_0)_2$ aufgefasst werden?

- b) Tragen Sie in der unten vorgedruckten Tabelle zunächst die Zustandsübergänge und die Ausgangsfunktion a des Schaltwerks ein! Q_i bezeichnet dabei den aktuellen Zustand und Q'_i den Nachfolgezustand des i -ten Zustandsspeichers. Es gilt die rechts vorgegebene dichte Zustandskodierung.

Q_1	Q_0	Zustand
0	0	Z_0
0	1	Z_1
1	0	Z_2
1	1	Z_3

- c) Wie Sie später herausfinden, sind die Zustandsspeicher des Schaltwerks mit JK-Flip-Flops realisiert, wobei **nur** die Beschaltungen J=0, K=1 (Reset) bzw. J=1, K=0 (Set) verwendet wurden. Tragen Sie die resultierenden Werte für die Vorbereitungseingänge J_1 , K_1 , J_0 und K_0 in die Tabelle ein!

e_1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
e_0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Q_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Q_0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
Zustand																
Q'_1																
Q'_0																
a																
J_1																
K_1																
J_0																
K_0																

Aufgabe 7: Lichtsteuerung mit Dimm-Funktion

Realisieren Sie eine Lichtsteuerung. Die Lichtsteuerung besteht aus zwei Tastern e_0 und e_1 . Die Lichtintensität kann vier mögliche Zustände annehmen: *dunkel*(00), *gedimmt-dunkel*(01), *gedimmt-hell*(10) und *hell*(11). Durch Betätigen von Taster 1 ($e_0 = 1$) wird die Helligkeit um eine Stufe erhöht, sofern nicht bereits die maximale Helligkeit erreicht ist. Betätigen von Taster 2 ($e_1 = 1$) verringert die Helligkeit, sofern das Licht nicht bereits ausgeschaltet ist. Ist die Lichtintensität im Zustand *hell*, so bewirkt ein gleichzeitiges Betätigen von beiden Tastern das sofortige Abschalten des Lichtes. In jedem anderen Zustand bewirkt das Eingangssignal 11 den sofortigen Wechsel auf volle Lichtintensität.

Weiters besitzt die Steuerung drei Ausgänge: a_0 , a_1 , a_2 , die wie folgt vom Zustand abhängen (Notation: $Zustand \mapsto a_2 a_1 a_0$):

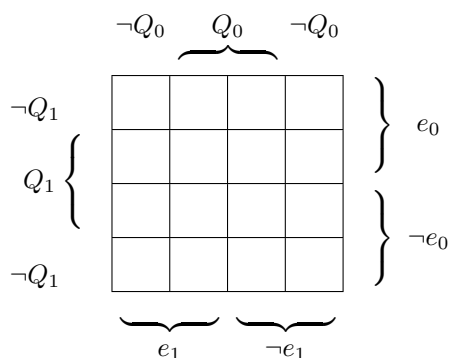
$dunkel \mapsto 000$
 $gedimmt-dunkel \mapsto 001$
 $gedimmt-hell \mapsto 011$
 $hell \mapsto 111$

- a) Befüllen Sie die nachfolgende Tabelle der Zustandsübergänge! D_i bezeichnet den Folgezustand von Q_i . (D_i ist der Vorbereitungseingang jenes D-Flip-Flops, das den Zustandsspeicher Q_i realisiert.)

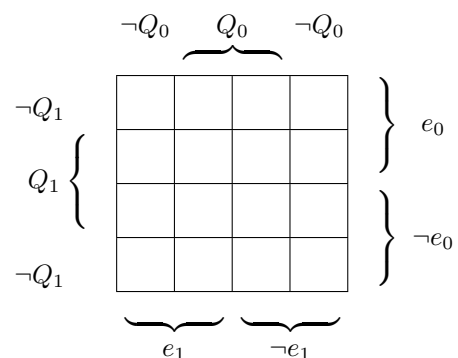
e_1	e_0	Q_1	Q_0	D_1	D_0
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

- b) Vereinfachen Sie die Schaltfunktionen für D_1 und D_0 mit den nachfolgenden KV-Diagrammen, sodass sie anschließend in dem bei Teilaufgabe c) vorgedruckten PLD realisiert werden können!

D_1 :



D_0 :



Aufgabe 8: Schaltwerksentwicklung – Tintenstrahldrucker / Teil 1

Entwerfen Sie einen Mealy-Automaten für die getaktete Steuerung des nachfolgend beschriebenen Tintenstrahldruckers:

Zustandsübergänge erfolgen ausschließlich zum Takt. Die Steuerung hat die drei binären Eingangssignale A , P und T . Eingang A signalisiert, ob aktuell ein noch offener Druckauftrag vorhanden ist ($A = 1$) oder nicht ($A = 0$). Die anderen beiden Eingänge sind jeweils mit einem Störungssensor verbunden. Eingang P gibt an, ob noch Papier im Papierfach ist ($P = 0$) oder nicht ($P = 1$) und Eingang T gibt an, ob die Tintenpatrone leer ist ($T = 1$) oder nicht ($T = 0$).

Weiters hat die Steuerung die drei Ausgänge E , D und F , wobei E für den Papiereinzug verantwortlich ist, D den Druckvorgang steuert und F mit einer LED verbunden ist, die einen Störfall anzeigt.

Der Drucker selbst kennt vier verschiedene Zustände:

- **Warten:** Der Drucker wartet auf Aufträge und ist bereit, mit einem Druckauftrag zu beginnen. Sobald ein Auftrag ankommt, wechselt der Drucker in den Zustand **Papiereinzug**, sofern noch Papier im Papierfach ist ($P = 0$). Anderenfalls wechselt der Drucker in den Zustand **Fehler** und schaltet die LED an ($F = 1$).
- **Papiereinzug:** Der Drucker holt sich Papier aus dem Papierfach. Dies nimmt genau eine Taktperiode in Anspruch. Danach wird in den Zustand **Drucken** gewechselt, sofern die Tintenpatrone nicht leer ist. Ist die Patrone leer, so wird in den Zustand **Fehler** gewechselt und die LED beginnt zu leuchten. Sobald der Drucker in den Zustand **Papiereinzug** wechselt, muss $E = 1$ gesetzt werden. In jedem anderen Fall und Zustand muss $E = 0$ gelten.
- **Drucken:** Der Auftrag wird gedruckt, was wiederum genau einen Takt in Anspruch nimmt. Ist danach noch immer Eingang $A = 1$, d.h. ein weiterer Druckauftrag ist vorhanden, wird wieder in den Zustand **Papiereinzug** zurück gewechselt, sofern das Papierfach nicht leer ist. Bei vorhandenem weiteren Auftrag und leerem Papierfach wird wieder in den Fehlerzustand gewechselt und die LED wird aktiviert. Ist kein weiterer Auftrag vorhanden, wird in den Zustand **Warten** zurückgekehrt. Beim Betreten des Zustands **Drucken** muss $D = 1$ gesetzt werden. Beim Verlassen des Zustands **Drucken** wird D wieder auf 0 gesetzt.
- **Fehler:** Das System verharrt so lange im Zustand **Fehler**, bis alle Fehler beseitigt sind oder bis kein Druckauftrag mehr vorhanden ist. Sind alle Fehler behoben, wird in den Zustand **Papiereinzug** gewechselt, falls ein Druckauftrag vorhanden ist. Ansonsten wird in den Zustand **Warten** gewechselt. Beim Verlassen des Fehlerzustandes wird die LED jedenfalls deaktiviert.

Ein Druckauftrag kann jederzeit abgebrochen werden. Befindet sich der Drucker also aktuell nicht im Fehlerzustand und der Eingang A wird 0, so wechselt er in den Wartezustand.

Das System geht nur dann in den Fehlerzustand, wenn eine Ressource, die aktuell benötigt wird, nicht verfügbar ist. Also wenn in den Zustand **Papiereinzug** gewechselt werden soll, aber kein Papier vorhanden ist, oder wenn in den Zustand **Drucken** gewechselt werden soll, aber die Tintenpatrone leer ist.

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

