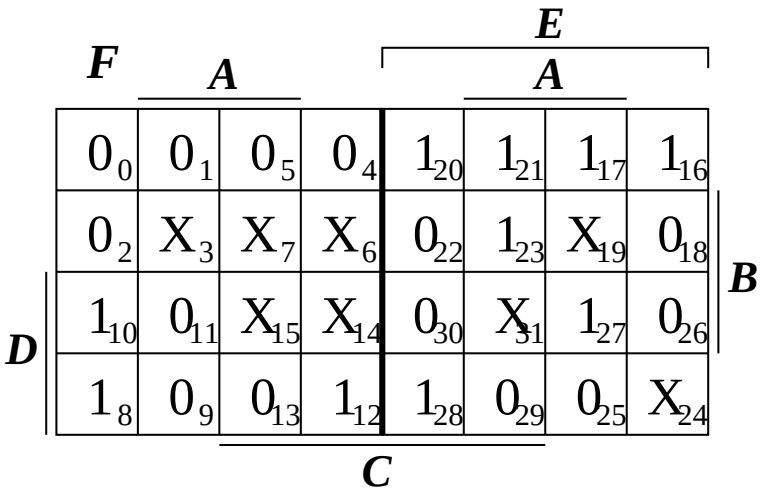
Aufgabe 1

Primterme und Kernimplikanten

Gegeben ist die durch das folgende KV-Diagramm partiell definierte Funktion F.



a) Bestimmen Sie die Primterme der Funktion F. Tragen Sie dazu die Einser-Blöcke in das KV-Diagramm ein und notieren Sie die entsprechenden Terme !

b) Bestimmen Sie die Kernimplikanten der Funktion F. Geben Sie für jeden Kernimplikanten an, aufgrund welcher Minterme Sie diesen Kernimplikanten als solchen identifiziert haben !

Aufgabe 2

Entwurf eines Schaltnetzes

Für eine bestimmte Anwendung soll eine spezielle Multiplexerschaltung (MUX) mit zwei Signaleingängen (E_0, E_1), zwei Steuereingängen (S_0, S_1) und vier Signalausgängen (A_0, A_1, A_2, A_3) entworfen werden. Die MUX-Schaltung soll folgendes Verhalten besitzen:

$$A_0 = \begin{cases} 0 & \text{falls} \quad S_1 = 0 \wedge S_0 = 0 \\ E_0 & \text{falls} \quad S_1 = 0 \wedge S_0 = 1 \\ E_1 & \text{falls} \quad S_1 = 1 \wedge S_0 = 0 \\ 1 & \text{falls} \quad S_1 = 1 \wedge S_0 = 1 \end{cases} \qquad A_1 = \begin{cases} 0 & \text{falls} \quad S_1 = 0 \wedge S_0 = 0 \\ E_1 & \text{falls} \quad S_1 = 0 \wedge S_0 = 1 \\ E_0 & \text{falls} \quad S_1 = 1 \wedge S_0 = 0 \\ 1 & \text{falls} \quad S_1 = 1 \wedge S_0 = 1 \end{cases}$$

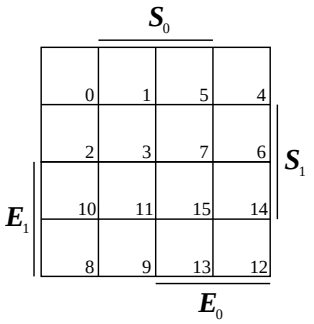
$$A_2 = \begin{cases} \text{X} & \text{falls} \quad S_1 = 0 \wedge S_0 = 0 \\ E_0 & \text{falls} \quad S_1 = 0 \wedge S_0 = 1 \\ E_1 & \text{falls} \quad S_1 = 1 \wedge S_0 = 0 \\ \text{X} & \text{falls} \quad S_1 = 1 \wedge S_0 = 1 \end{cases} \qquad A_3 = \begin{cases} \text{X} & \text{falls} \quad S_1 = 0 \wedge S_0 = 0 \\ E_1 & \text{falls} \quad S_1 = 0 \wedge S_0 = 1 \\ E_0 & \text{falls} \quad S_1 = 1 \wedge S_0 = 0 \\ \text{X} & \text{falls} \quad S_1 = 1 \wedge S_0 = 1 \end{cases}$$

In den obigen Gleichungen kennzeichnet 'X' einen beliebigen Ausgangszustand.

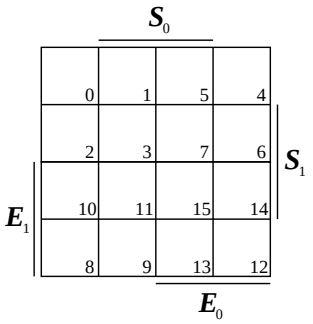
a) Tragen Sie die Zustände der Ausgänge A_0, A_1, A_2 und A_3 in die nachfolgende Tabelle ein. Kennzeichnen Sie dabei den Zustand 'beliebig' durch das Symbol 'X'.

	Signal-eingänge		Steuer-eingänge		Signal-ausgänge			
#	E_1	E_0	S_1	S_0	A_3	A_2	A_1	A_0
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				
10	1	0	1	0				
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

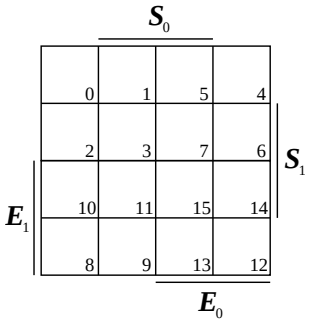
b) Übertragen Sie nun die Zustände in die nachfolgend abgebildeten KV-Diagramme und verwenden sie anschließend die KV-Diagramme zur Minimierung der Ausgangsfunktionen. Nutzen Sie dabei die Ausgangszustände beliebig ('X') möglichst geschickt aus, um die Funktionen zu minimieren.
Tragen Sie anschließend die 4 optimierten Ausgangsgleichungen A_0, A_1, A_2 und A_3 in die vorgesehenen Felder ein.



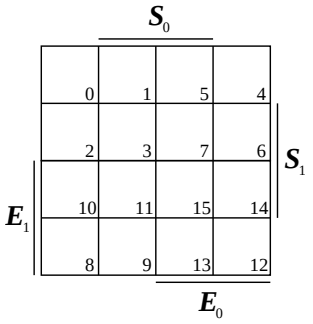
$A_0 =$ _____



$A_1 =$ _____



$A_2 =$ _____



$A_3 =$ _____

c) Zeichnen Sie nun für die 4 Ausgänge A_0, A_1, A_2 und A_3 die Ansteuerschaltung gemäß den Ausgangsgleichungen. Dabei stehen Ihnen UND- und ODER-Gatter mit bis zu 4 Eingängen zur Verfügung. Die Signaleingänge (E_0, E_1) und die Steuereingängen (S_0, S_1) liegen sowohl negiert als auch nicht negiert vor.

d) Die Ausgangssignale können anstatt mit UND/ODER-Gattern auch mit Hilfe von 2-auf-1-Multiplexern erzeugt werden. Zeigen Sie dies exemplarisch für das Ausgangssignal A_0, indem Sie eine entsprechende Schaltung, die nur aus 2-auf-1-Multiplexern besteht, entwerfen. Beachten Sie, dass Sie an die Eingänge der Multiplexer auch das konstante Eingangssignal '0' bzw. '1' anlegen dürfen!
Die zur Verfügung stehenden Multiplexerbausteine besitzen ein Ausgangssignal A_M, einen Steuereingang S_M und die beiden Signaleingänge E_{M1} und E_{M2}. Für das Ausgangssignal gilt:

$$A_M = \begin{cases} E_{M1} & \text{falls} \quad S_M = 0 \\ E_{M2} & \text{falls} \quad S_M = 1 \end{cases}$$

Aufgabe 3

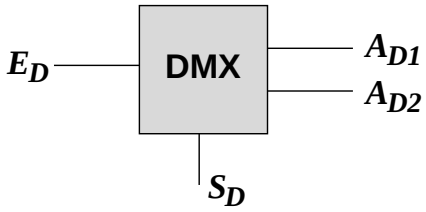
Entwurf einer Demultiplexerschaltung

Es sollen drei verschiedene Schaltungen mit 1-auf-2-Demultiplexern aufgebaut werden. Die zur Verfügung stehenden Demultiplexerbausteine besitzen einen Signaleingang E_D, einen Steuereingang S_D und zwei Signalausgänge A_{D1} und A_{D2}. Für die beiden Ausgangssignale gilt:

$$A_{D1} = \begin{cases} E_D & \text{falls} \quad S_D = 0 \\ 0 & \text{falls} \quad S_D = 1 \end{cases}$$

$$A_{D2} = \begin{cases} 0 & \text{falls} \quad S_D = 0 \\ E_D & \text{falls} \quad S_D = 1 \end{cases}$$

Die nachfolgende Abbildung veranschaulicht die Anschlußbelegung.



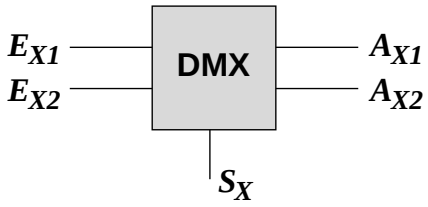
a) Entwerfen Sie eine Schaltung, die zwei Eingangsignale E_{X1} und E_{X2} an Ihre Ausgänge A_{X1} und A_{X2} weiterleitet. Dabei soll in Abhängigkeit von einem Steuereingang S_X wahlweise entweder

   - $A_{X1} = E_{X1}$ und $A_{X2} = E_{X2}$ oder
   - $A_{X1} = E_{X2}$ und $A_{X2} = E_{X1}$ gelten.

   Die mathematische Formulierung der gewünschten Funktion lautet also:

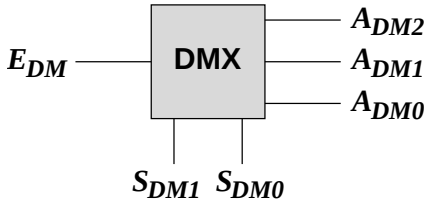
$$A_{X1} = \begin{cases} E_{X1} & \text{falls} \quad S_X = 0 \\ E_{X2} & \text{falls} \quad S_X = 1 \end{cases} \qquad A_{X2} = \begin{cases} E_{X2} & \text{falls} \quad S_X = 0 \\ E_{X1} & \text{falls} \quad S_X = 1 \end{cases}$$

   Die nachfolgende Abbildung veranschaulicht die Anschlußbelegung.



   Der Schaltungsentwurf soll nur ODER-Gatter und 1-auf-2-Demultiplexer enthalten. Skizzieren Sie die Schaltung.

b) Bauen Sie mit Hilfe des gegebenen 1-auf-2-Demultiplexers einen 1-auf-3-Demultiplexer auf. Der 1-auf-3-Demultiplexer soll einen Eingang E_{DM}, zwei Steuereingänge S_{DM0} und S_{DM1}, sowie drei Ausgänge A_{DM0}, A_{DM1} und A_{DM2} besitzen. Die nachfolgende Abbildung veranschaulicht die Anschlußbelegung.

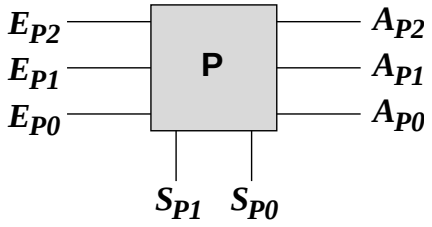


Die nachfolgend aufgeführte Tabelle zeigt die gewünschten Ausgangssignale in Abhängigkeit der Steuersignale S_{DM0} und S_{DM1}.

S_{DM1}	S_{DM0}	A_{DM2}	A_{DM1}	A_{DM0}
0	0	0	0	0
0	1	0	0	E_{DM}
1	0	0	E_{DM}	0
1	1	E_{DM}	0	0

In Ihrem Schaltungsentwurf dürfen 1-auf-2-Demultiplexer der gezeigten Art, sowie Standard-Logikgatter (AND, NAND, OR, NOR, XOR) enthalten sein. Skizzieren Sie die Schaltung.

c) Verwenden Sie den von Ihnen in Teilaufgabe b) entwickelten 1-auf-3-Demultiplexer, um eine Schaltung zu realisieren, die drei Eingangsignale E_{P0}, E_{P1} und E_{P2} in verschiedenen Kombinationen an Ihre Ausgänge A_{P0}, A_{P1} und A_{P2} weiterleitet.



Die nachfolgend aufgeführte Tabelle zeigt die gewünschten Ausgangssignale in Abhängigkeit der Steuersignale S_{P0} und S_{P1}.

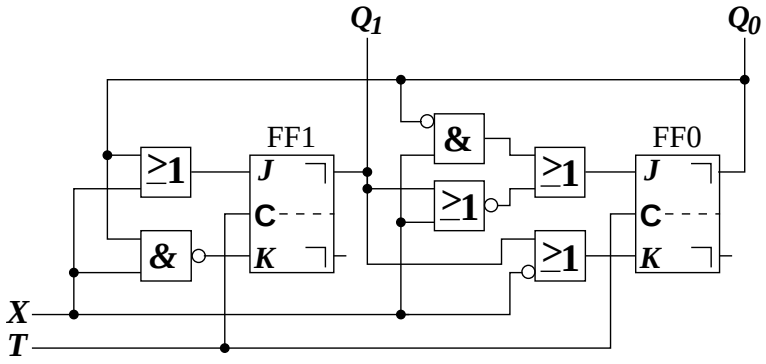
S_{P1}	S_{P0}	A_{P2}	A_{P1}	A_{P0}
0	0	0	0	0
0	1	E_{P2}	E_{P1}	E_{P0}
1	0	E_{P1}	E_{P0}	E_{P2}
1	1	E_{P0}	E_{P2}	E_{P1}

Der Schaltungsentwurf soll nur ODER-Gatter und die in Aufgabenteil b) beschriebenen 1-auf-3-Demultiplexer enthalten. Skizzieren Sie die Schaltung.

Aufgabe 4

Schaltwerksanalyse

Gegeben ist das unten abgebildete synchrone Schaltwerk mit JK-MS-Flipflops, welches in der Folge analysiert werden soll. Ausgangsgröße sei die dual kodierte Zustandsnummer $(Q_1\,Q_0)|_2$.



a) Um welchen Automatentyp handelt es sich bei dem gezeigten Schaltwerk? Kreuzen Sie die entsprechende richtige Antwort an und begründen Sie kurz ihre Antwort:

- ☐ Es handelt sich weder um einen Moore-Automaten, noch um einen Mealy-Automaten.
- ☐ Das Schaltwerk ist der Definition nach ein Moore-Automat, allerdings kein Mealy-Automat.
- ☐ Das Schaltwerk ist der Definition nach ein Mealy-Automat, allerdings kein Moore-Automat.
- ☐ Das Schaltwerk ist der Definition nach sowohl ein Moore-Automat, als auch ein Mealy-Automat.

b) Geben Sie Funktionsgleichungen für alle Flipflop-Eingänge in Abhängigkeit von Q_1, Q_0 und X an. Dabei bezeichnen J_0, K_0 die Eingänge von FF0, J_1 und K_1 entsprechend die von FF1.

c) Vervollständigen Sie die folgende Zustandstabelle des Schaltwerks.

X	Q_1	Q_0	J_1	K_1	J_0	K_0	Q_1^+	Q_0^+
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

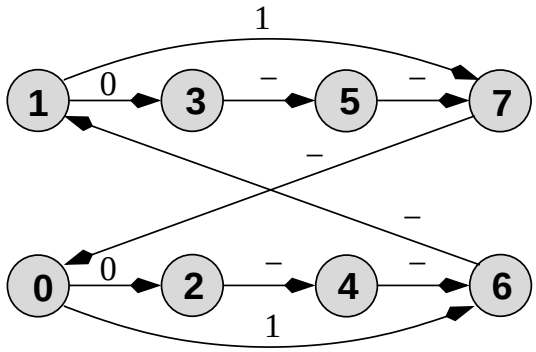
d) Zeichnen Sie den Zustandsgraphen des Automaten. Denken Sie dabei an die duale Zustandskodierung. Kennzeichnen Sie die Zustandsübergänge (Pfeile) durch den zugehörigen Wert von X.

e) Angenommen, das Schaltwerk befindet sich zum Zeitpunkt t_0 im Zustand 0 ($Q_1 = 0$, $Q_0 = 0$). In welchem Zustand befindet sich das Schaltwerk nach vier Taktzyklen T, also zum Zeitpunkt $t_1 = t_0 + 4 \cdot T$, wenn $X = 1$ ist?

Aufgabe 5

Schaltwerkssynthese

Gegeben ist der unten abgebildete Zustandsgraph. Gesucht ist ein Schaltwerk mit JK-MS-FFs, welches diesen Zustandsgraphen realisiert.



Die binäre Variable an den Kanten des Zustandsgraphen stelle eine Eingangsvariable X dar, über die das Schaltwerk von außen in seinem Verhalten gesteuert werden kann. Ein "-" bedeutet *don't-care*, dabei besteht keine Verhaltensalternative.

a) Wieviele FFs werden mindestens benötigt? Begründen Sie Ihre Antwort kurz.

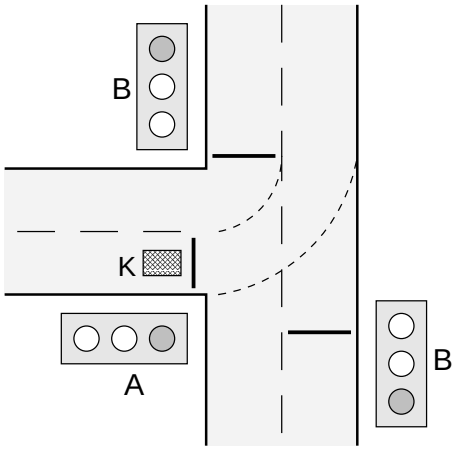
b) Ermitteln Sie die Zustandstabelle des Schaltwerks. Benutzen Sie dabei so viele *don't-care*-Einträge wie möglich. Verwenden Sie eine duale Zustandscodierung, d.h. es gelte $z = (Q_n \ldots Q_0)_2$.

c) Geben Sie möglichst kurze Ansteuergleichungen für die Eingänge der Flipflops an. Verwenden Sie KV-Diagramme zur Vereinfachung.

Aufgabe 6

Entwurf einer Steuerung

Zur bedarfsgerechten Steuerung der Grünphase einer Ampel A an der Einmündung einer Neben- in eine Hauptstrasse befindet sich in der Fahrbahn der Nebenstraße vor der Ampel A ein Kontakt K.



Das Überfahren des Kontaktes K erzeuge ein Signal $K = 1$. Als dessen Folge sollen die Ampeln A und B, bei Stellung "rot" von A und "grün" von B, nacheinander in die durch die nachfolgende Tabelle festgelegte Phasen geschaltet werden:

		Ampel **A**	Ampel **B**	Dauer der Phase
Ausgangszustand	0	rot	grün	mindestens $4T$
	1	rot	gelb	T
	2	rot/gelb	rot	T
	3	grün	rot	$4T$
	4	gelb	rot	T
	5	rot	rot/gelb	T
Ausgangszustand	0	rot	grün	mindestens $4T$

Die Grünphase der Ampel A soll wie in der Tabelle festgelegt stets vier Zeitelemente ($4T$) lang andauern, die Grünphase der Ampel B soll mindestens vier Zeitelemente ($4T$) lang andauern, selbst wenn an der Nebenstrasse ein Fahrzeug wartet und den Kontakt K vor Ablauf der vier Zeitelemente ausgelöst hat.

a) Wieviele und welche Eingangs- und Ausgangsvariablen werden zur Realisierung des entsprechenden Schaltwerks benötigt, wenn Sie davon ausgehen, dass für jedes Lichtsignal eine eigene Variable verwendet wird?

b) Entwerfen Sie ein Zustandsdiagramm, das die Funktionalität der gesuchten Ampelsteuerung beschreibt. Sie können dabei von folgenden Annahmen ausgehen:

   Sie können der Einfachheit halber davon ausgehen, dass Sie dieses Automatenmodell durch ein synchrones Schaltwerk realisieren, bei dem die Taktperiodendauer T ist. Außerdem können Sie davon ausgehen, dass das Signal K solange gleich 1 ist, solange sich dort ein Fahrzeug befindet.

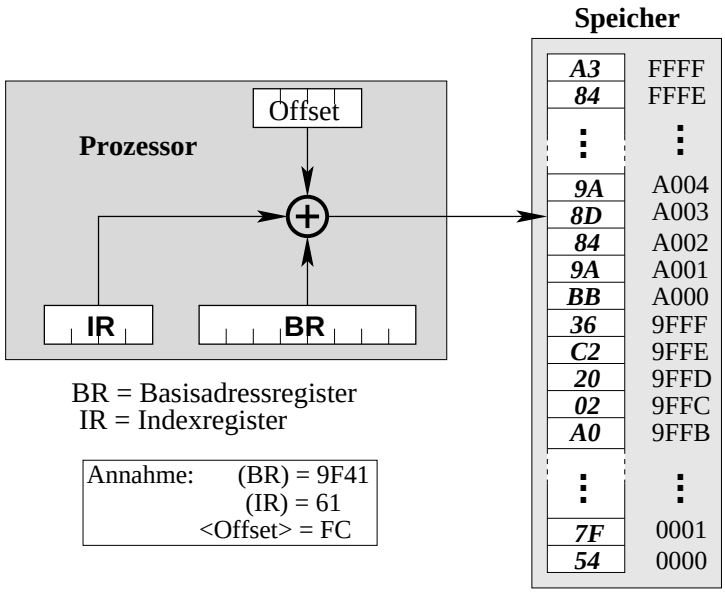
Lösungshinweis:

Das Zustandsdiagramm besteht aus einem einzelnen Zyklus von insgesamt $6+2\cdot 3=12$ Zuständen mit den durch die Tabelle vorgegebenen Ampelphasen.
Achten Sie bei der Darstellung des Zustandsdiagramms auf eine vollständige Kantenbeschriftung.

Aufgabe 7

Prozessor

Ein 8-bit-Mikroprozessor besitze ein 16-bit-**Basisadressregister BR** und ein 8-bit-**Indexregister IR**. Bei einer indizierten Adressierung werde die *effektive Adresse EA* eines Operanden durch die Addition des Inhalts von IR und eines im Befehl angegebenen Offsets zum Inhalt von BR gewonnen. Dabei werden die Inhalte von BR und IR als vorzeichenlose ganze Zahlen, der Offset als vorzeichenbehaftete ganze Zahl im Zweierkomplement aufgefasst. Das folgende Bild verdeutlicht die Bildung der EA und zeigt einen Ausschnitt der Speicherbelegung zum aktuellen Zeitpunkt.



Das Indexregister IR verfüge über die Möglichkeit der automatischen Modifikation (autoinkrement/-autodekrement) um die Werte $n = 1, 2, 3, 4$ sowie die Möglichkeit der Skalierung mit den Faktoren $m = 1, 2, 4$.

Der Inhalt des Basisregisters BR sei wie im obigen Bild gezeigt 9F41 und der Offset FC (Hinweis: $\mathrm{FC}|_{16} = -4|_{10}$), der Inhalt des Indexregisters IR sei 61.

Geben Sie an, welche effektive Adresse EA bei **indizierter Adressierung** mit Offset und Skalierung mit $n = 1$ und $m = 2$ angesprochen wird und wie diese Adresse dabei gebildet wird.

Führen Sie diese Betrachtungen für die drei Fälle

(1) keine automatische Modifikation,

(2) automatische Predekrementierung und

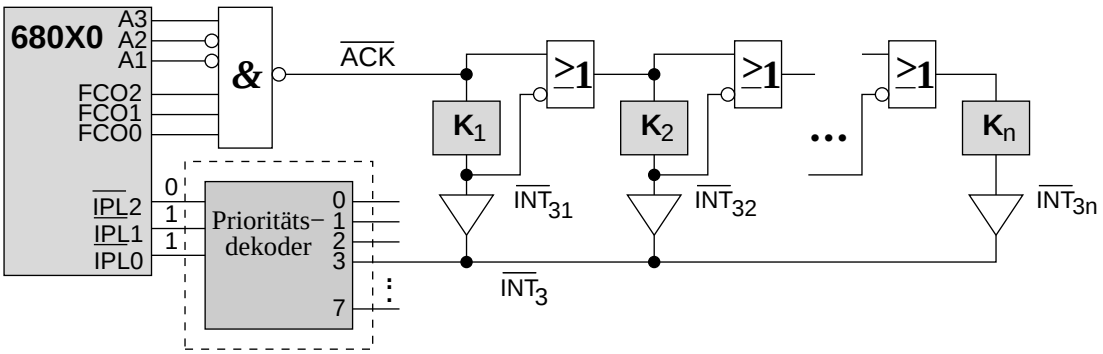
(3) automatische Postdekrementierung

durch. Geben Sie ferner an, welches Datum aus dem Speicher gelesen wird und welchen Wert das Indexregister IR nach der Befehlsausführung hat.

Aufgabe 8

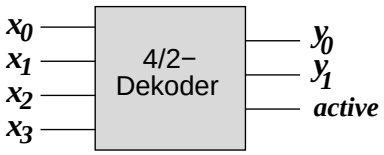
Entwurf eines Prioritätsdekoders

Im Kurs wurde das Daisy-Chain-Verfahren zur Verwaltung mehrerer Unterbrechungsquellen an einem Prozessor MC68000 erläutert. Der prinzipielle Aufbau einer solchen Logik ist im nachfolgenden Bild noch einmal dargestellt.



In dieser Aufgabe soll eine Schaltung für den im Bild gezeigten Prioritätsdekoder entworfen werden. Ein solcher Dekoder realisiert eine Funktion, welche die acht Eingangssignale $\overline{\mathrm{INT}}_0, \ldots, \overline{\mathrm{INT}}_7$ auf die drei Ausgangssignale $\overline{\mathrm{IPL}}0, \overline{\mathrm{IPL}}1, \overline{\mathrm{IPL}}2$ abbildet. Da die Minimierung einer Funktion von acht Eingangsvariablen mit Hilfe eines KV-Diagramms schlecht möglich ist, soll der gesamte Dekoder aus zwei gleichartigen Schaltungen zusammengesetzt werden, die jeweils vier Eingangssignale auf zwei Ausgänge durchschalten.

a) Erstellen Sie eine Wahrheitstabelle für einen Prioritätsdekoder gemäß folgendem Blocksymbol mit vier Eingängen $x_0, \ldots, x_3$, wobei der Index 0 für die niedrigste Priorität steht, und zwei Ausgängen y_0, y_1 für die Kennung des jeweils priorisierten aktiven Einganges. Beachten Sie dabei, dass die Eingangs- und Ausgangssignale in negativer Logik vorliegen sollen, d.h. alle Eingangs- und Ausgangswerte sind im aktiven Zustand gleich 0.



Zusätzlich soll dieser vereinfachte Dekoder hier einen Ausgang *active* besitzen, der in positiver Logik signalisiert, dass tatsächlich eine Unterbrechungsanforderung vorliegt. (6P)

b) Entwerfen Sie eine minimale Realisierung des unter b) beschriebenen Dekoders mit Hilfe eines KV-Diagramms und geben Sie das Schaltbild des minimierten Dekoders an. (6P)

c) Entwerfen Sie die Gesamtschaltung des Dekoders mit 8 Eingängen und 3 Ausgängen bestehend aus zwei minimierten Dekoderbausteinen nach d) und einem zusätzlichen Multiplexer (MUX). Verwenden Sie für die Darstellung der Dekoder sowie für den MUX ausschließlich Blocksymbole. Beachten Sie wiederum, dass Eingangs- und Ausgangssignale in negativer Logik arbeiten. (10P)

Hinweis: Die Zusammenführung der Ausgänge der einzelnen Dekoder kann mit Hilfe eines 4-auf-2-MUX erfolgen. Dabei wird einer der beiden *active*-Ausgänge als Ansteuerung benötigt.