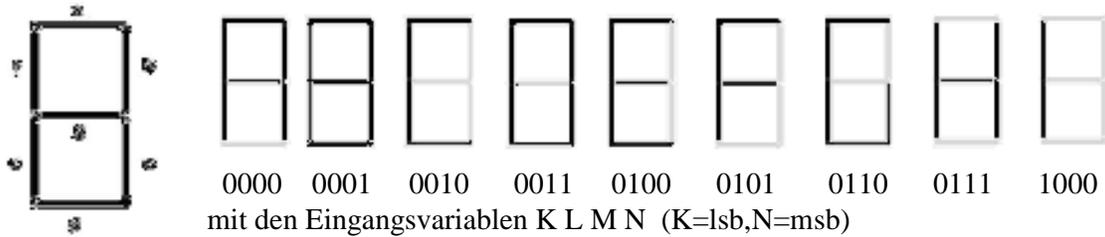


Einführung in die Technische Informatik 2. Semester 2.Test

1.Punkt: Segmentanzeige

Die Angabe besteht aus einer Segmentanzeige und deren Zustandskonfigurationen und es soll eine Schaltung erstellt werden die über eine Kodierung alle Segmente korrekt ausrichtet

Bei einem Flughafen gibt es die Terminals A-I. Entwerfen sie für jedes der 7 Segmente der Anzeigentafel eine Schaltung die über 4bit angesteuert wird. Segmentmuster:



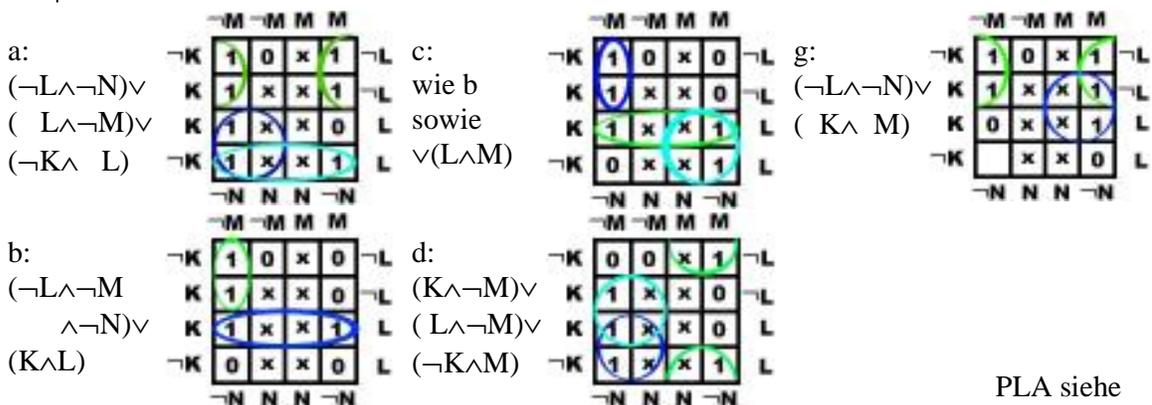
Vorgehensweise:

- 1.Schritt: Tabelle mit den 4 Variablen aufstellen und 7 Segmenten erstellen indem aktive Segmente als 1 und inaktive als 0 gewertet werden. Alle restlichen unverwendeten Kodierungskombinationen mit don't cares auffüllen
- 2.Schritt: KV Diagramme zeichnen und alle log. 1 und don't care aus der Tabelle eintragen
- 3.Schritt: Don't cares sinnvoll interpretieren um möglichst 8er, 4er, 2er Blöcke aus kleineren logisch-1er-Blöcken zu bekommen.

Beispiel mit Angaben von oben:

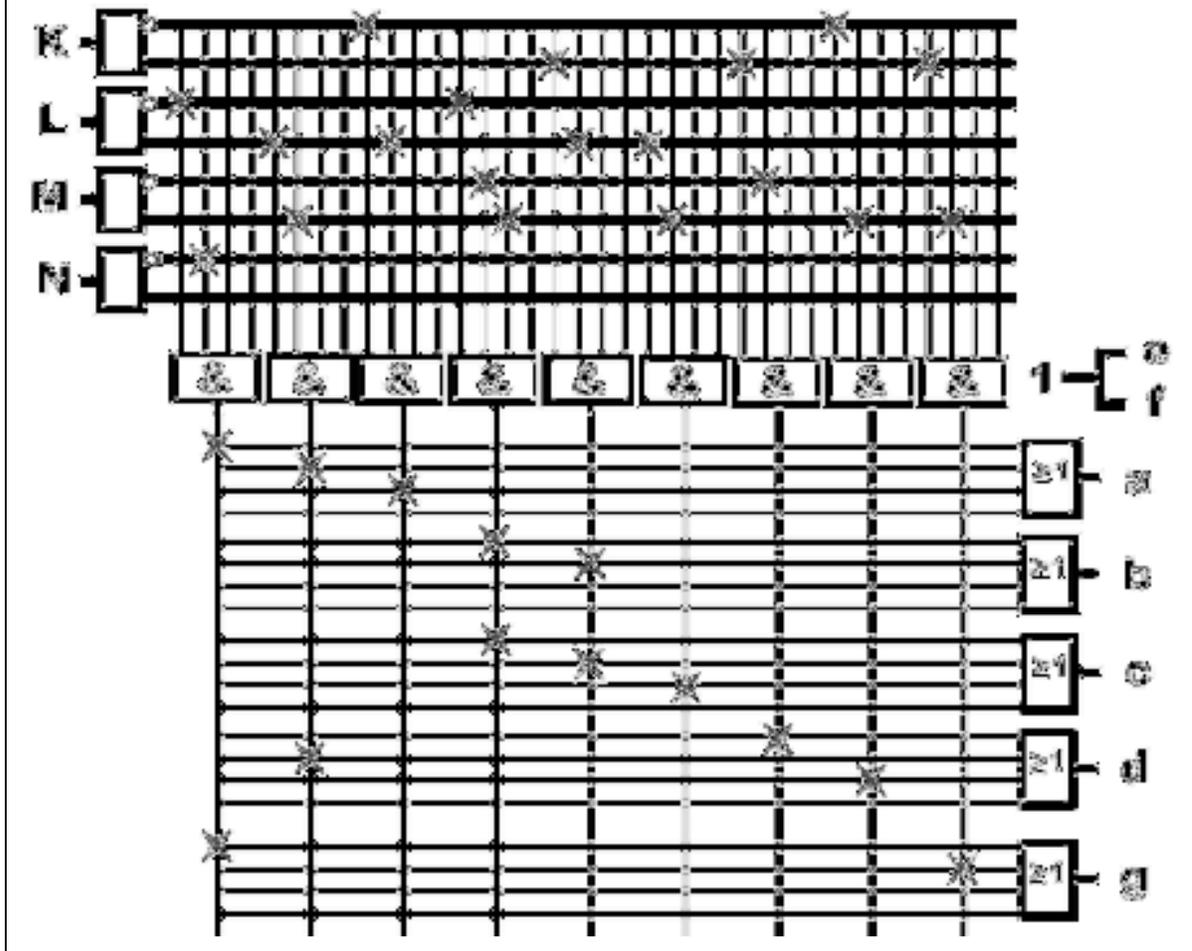
	A	B	C	D	E	F	G	H	I	-	-	-	-	-	-
K:	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
L:	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
M:	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
N:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
a:	1	1	1	1	1	1	1	0	0	x	x	x	x	x	x
b:	1	1	0	1	0	0	0	1	0	x	x	x	x	x	x
c:	1	1	0	1	0	0	1	1	0	x	x	x	x	x	x
d:	0	1	1	1	1	0	1	0	0	x	x	x	x	x	x
e:	1	1	1	1	1	1	1	1	1	x	x	x	x	x	x
f:	1	1	1	1	1	1	1	1	1	x	x	x	x	x	x
g:	1	1	0	0	1	1	0	1	0	x	x	x	x	x	x

Man sieht das e und f immer aktiv sind. Die verbleibenden Segmente sind ab I immer gleich denn bei N ist immer ein 0er und sonst x. Die „Mitte“ der KV's sieht also immer gleich aus. Für die restlichen Segmente die KV-Diagramme mit den Lösungstermen:



PLA siehe nächste Seite

Fortsetzung Bsp1: PLA Schaltung:



2.Punkt: Schaltwerk

Die Angabe ist eine Aufgabenstellung die in Form eines Schaltwerks realisiert werden soll.

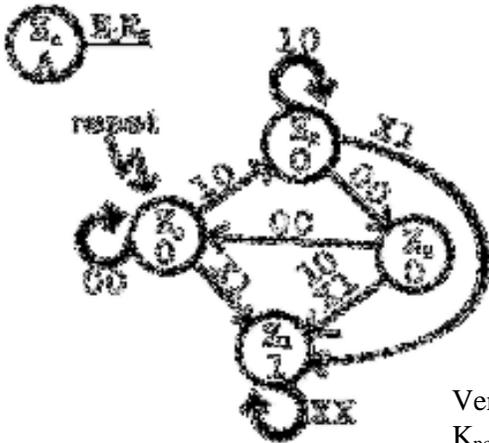
Eingänge E_1, E_2 , Ausgang A; Falls E_1 101 hintereinander liefert soll der Ausgang auf 1 gesetzt werden. Falls allerdings E_2 zu irgendeinem Zeitpunkt 1 liefert soll sofort 1 am Ausgang anliegen. Realisieren sie die Funktion als Moore-Schaltwerk

Vorgehensweise:

- 1.Schritt: Aufgabenstellung analysieren und eine Skizze der Zustände mit Übergängen anlegen
- 2.Schritt: Zustandskodierung (n Zustände $\Rightarrow \lceil \lg(n) \rceil$ Variable, bzw. n Variable je nach Angabe)
- 3.Schritt: Übergangsfunktionstabelle anlegen
- 4.Schritt: Vereinfachen und in eine Schaltung umwandeln (PLA falls nötig)

Beispiel mit Angaben von voriger Seite:

Skizze:



Übergangsfunktion:

	Z ₀	Z ₁	Z ₂	Z ₃
K _{alt}	0000	1111	0000	1111
L _{alt}	0000	1111	1111	0000
E1	0101	0101	0101	0101
E2	0011	0011	0011	0011
K _{neu}	0011	1111	1011	0111
L _{neu}	0111	1111	0111	0111
A	0011	1111	0011	0111

K_{neu}

	$\neg E_1$	E ₁	
$\neg K$	0	0	$\neg L$
K	0	1	$\neg L$
K	1	1	L
$\neg K$	1	0	L
	E ₂	$\neg E_2$	

L_{neu}

	$\neg E_1$	E ₁	
$\neg K$	0	1	$\neg L$
K	0	1	$\neg L$
K	1	1	L
$\neg K$	0	1	L
	E ₂	$\neg E_2$	

Kodierung:

Für dichte Kodierung

	K	L
Z ₀	0	0
Z ₁	1	1

Z₂ 0 1 (Anm: die Kodierung kann beliebig erfolgen, aber für Z₁=11 errechnet sich
 Z₃ 1 0 A sofort aus (K UND L) und für K_{neu}, L_{neu} ergeben sich schöne KV's)

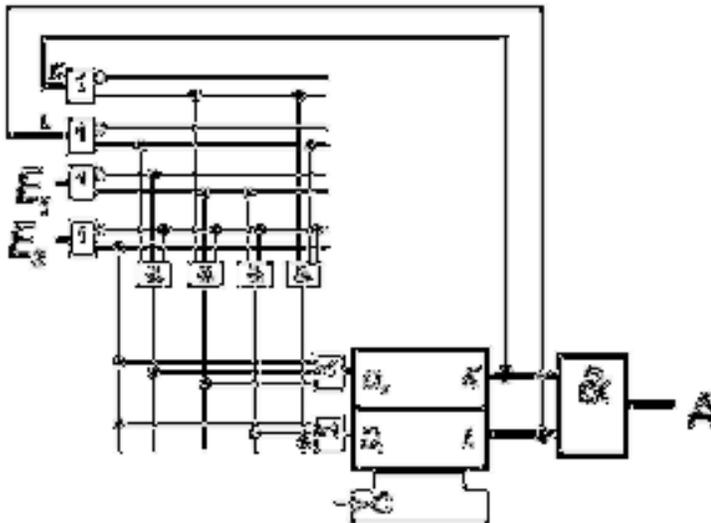
Vereinfacht über KV Diagramme:

$$K_{neu} = E_2 \vee (L \wedge \neg E_1 \wedge \neg E_2) \vee (K \wedge E_1 \wedge \neg E_2)$$

$$L_{neu} = E_2 \vee (E_1 \wedge \neg E_2) \vee (K \wedge L \wedge \neg E_2)$$

(da bei E₂ immer 1 steht wurde diese Spalte im KV nicht extra aufgeschrieben)

Schaltungsskizze:



Der 45 minütige Test beinhaltet zu den Praxisfragen auch Theoriefragen über den Stoff der Vorlesung bis einschließlich Kapitel 8.2 des Skriptums(Teil1&Teil2). Einige Fragen zum üben:

1. Über welche Funktionen verfügt eine ALU und was ist eine ALSU?
2. Aus welchen Teilen ist ein Micro16 aufgebaut und welche Funktionen haben diese?
3. Welche Adressierungsarten gibt es und wodurch unterscheiden sie sich?
4. Was bedeutet Interleaved Memory?
5. Was ist der Unterschied zwischen einem Moore und einem Meely Schaltwerk?
6. Was ist ein Prozess und ist ein Thread das selbe wie ein Prozess?
7. Um was geht es bei *Dijkstras Dining Philosophers Problem*?
8. Welche Möglichkeiten des Prozess-Schedulings gibt es?
9. Was bedeutet *Paging* in der Adresszuordnung?
10. Was sind Semaphoren?

Antworten:

1. parallele Addition, bitweise \wedge Verknüpfung, Komplementbildung und Durchschalten zweier Datenwörter. Ausgegeben werden Resultat, Übertrag, ein Vorzeichenflag und ein Nullflag. Verbindet man zusätzlich die ALU mit einem Shifter so bekommt man eine ALSU (Arithmetic Logic Shift Unit)
2. die ALSU (Berechnungen), ein Register (Zwischenspeicher), dem MAR und MBR (Speicheranbindung) sowie der Control Unit (Programmcode aufschlüsseln und durchführen)
3. Implied Model, Register Mode, Immediate Mode, Direct-Adressing Mode, Register-Indirect Mode, Program-Counter-Relativ-Adressing-Mode, Indirect-Adressing Mode
4. gliedert Daten auf mehrere Speicherbereiche auf, sodass die Daten beim aufrufen schneller Verarbeitet werden können.
5. Bei einem Meelywerk ist die Ausgangsfunktion zusätzlich auch direkt von den Eingangswerten abhängig und nicht nur indirekt wie das Moorewerk bei dem alleinig die Zustände entscheiden. Graphisch gesehen stehen bei Meely die Ausgänge bei jeder Kanten, bei Moore bei jedem Knoten.
6. Prozess= Programm in Exekution mit Registern und Daten, Thread = logisches Objekt
7. Veranschaulichung eines Deadlocks, wenn alle gleichzeitig etwas wollen und sich Ressourcen teilen müssen so kann alles zusammenbrechen.
8. First Come First Serve, Round Robin Scheduling, Static Priority Scheduling, Dynamic Priority Scheduling
9. ein virtuelles Speicherkonzept
10. ein Datentyp, der aus einem Counter und einer Prozess-Queue besteht. Falls mehrere Prozesse eine Ressource verlangen, so wird der Counter immer um 1 herabgesetzt und falls er unter 0 geht, der Prozess in die Queue eingetragen. Bei einem Abholen von Prozessen wird der Counter um 1 erhöht und das vorderste Element der Queue zugriff verschafft.