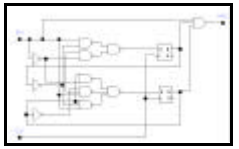
F7: Konstruktion av sekventiella kretsar

- **Innehåll:**
  - Procedur för konstruktion av synkron tillståndsmaskin
  - Exempel på konstruktion av synkron tillståndsmaskin
  - Tilldelning av koder för tillstånd (*state assignment*)
  - Syntes av tillståndsmaskiner med D-vippor

Analys jämfört med konstruktion och syntes

- **Analys**

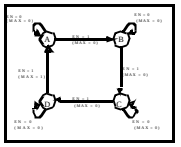
kretsdiagram



⇒

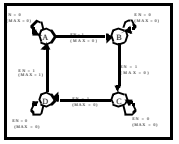
logiska grindar
logiska funktioner F,G
transition tabell
state/output tabell

beskrivning av beteendet med tillståndsgraf



- **konstruktion och syntes**

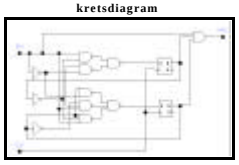
beskrivning av beteendet



⇒

beteende i naturligt språk
tillståndsgraf
tillståndstabell
transition-tabell
logiska uttryck
kretsdiagram

kretsdiagram



Procedur för syntes av tillståndsmaskin

- **Starta med en specifikation i ett naturligt språk (t.ex svenska)**
  - konstruera en *state/output*-tabell som motsvarar specifikationen
  - minimera antal tillstånd i *state/output*-tabellen
  - tilldela varje tillstånd en kod (*state assignment*)
  - konstruera en transition/output-tabell (samma som *state/output*-tabell fast med binära koder för tillstånden)
  - Välj typ av vippa (D- eller JK-vippa)
  - konstruera en *excitation*-tabell som visar vilka värden in till vipporna som krävs för att nå önskat tillstånd
  - ta fram logiska uttryck för *next-state* logiken (F)
  - ta fram logiska uttryck för *output* logiken (G)
- **Avsluta med att rita ett kretsdiagram**

Exempel på konstruktion av synkron tillståndsmaskin

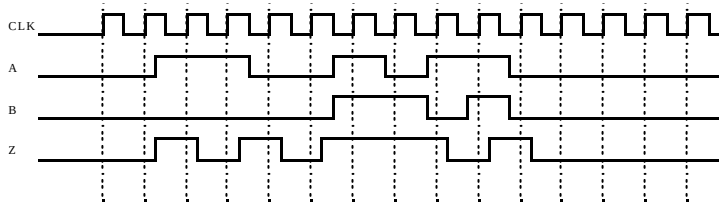
- **Specifikation**

Den synkrona tillståndsmaskinen har två ingångar A och B, samt en utgång Z. Utgången Z ska erhålla värdet 1 då:

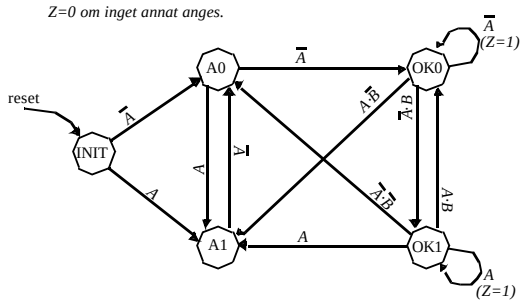
- A har haft samma värde under två på varandra följande klockcykler
- B har varit 1 från det att ovanstående villkor har varit sant. (om Z har satts till 1 p.g.a första villkoret, så håller B=1 kvar utgången Z till 1)

I alla övriga fall ska Z vara 0.

Tolkning av specifikationen i form av ett timingdiagram

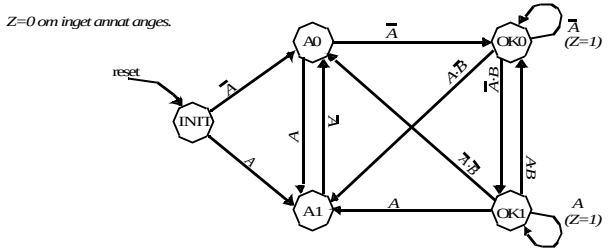


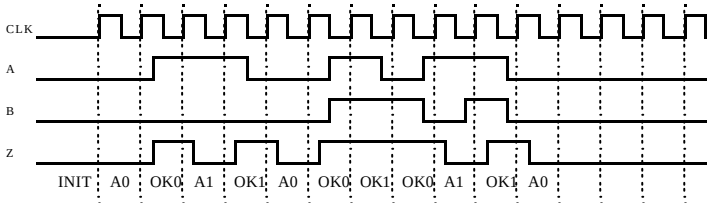
- **Tolkning av specifikationen i form av ett tillståndsdiagram**

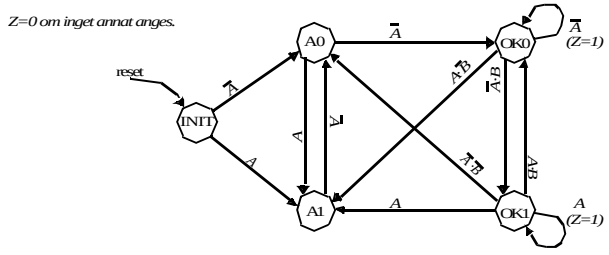


- **Konstruktion av tillståndstabell**

	A B				
	00	**01**	**11**	**10**	**Z**
INIT	A0	A0	A1	A1	0
A0	OK0	OK0	A1	A1	0
A1	A0	A0	OK1	OK1	0
OK0	OK0	OK0	OK1	A1	1
OK1	A0	OK0	OK1	OK1	1







- **Tilldelning av koder till tillstånd (state-assignment)**

Genom att välja koder kan man påverka storleken på *next-state* logiken. Målsättningen är att få så enkel logik som möjligt.

Namn på tillstånd	**Enkel binär**	**"Uppdelad"**	***One-hot***	**Nästan *one-hot***
INIT	000	000	00001	0000
A0	001	100	00010	0001
A1	010	101	00100	0010
OK0	011	110	01000	0100
OK1	100	111	10000	1000

Vi har 5 tillstånd med 3 bitar i tillståndsminnet som maximalt kan ha 2^3 antal tillstånd.

Detta innebär att vi har oanvända tillstånd. Önskas minimal logik sätts dessa till *don't-care* i Karnaugh-diagrammet vid optimering av de logiska uttrycket för *next-state* logiken.

Syntes av tillståndsmaskin med D-vippor

- **Procedur**
  - **konstruera en *excitation*-tabell som visar vilka värden in till vipporna som krävs för att nå önskat tillstånd**
  - **ta fram logiska uttryck för *next-state* logiken (F)**
  - **ta fram logiska uttryck för *output* logiken (G)**

Då tillstånden har tilldelats koder så utgår man från *transition/output* tabellen:

	A B 00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	Q1+ Q2+ Q3+				

- **Ta fram excitation tabell**

Med D-vippor så blir övergången från transition-tabell till excitation tabell enkel.
Den karakterisktisk ekvationen är: Q+ = D

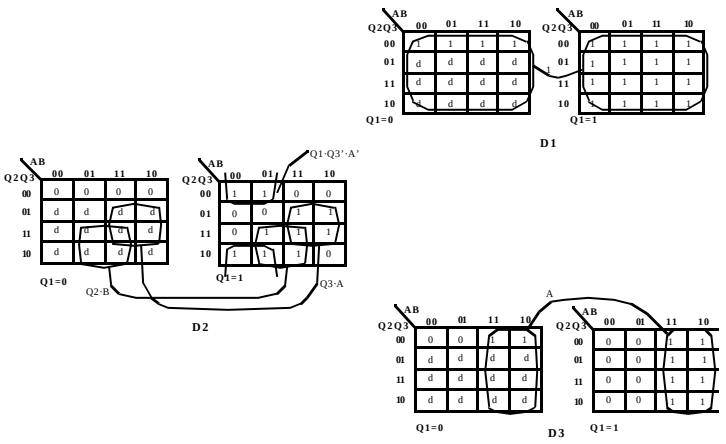
Q1 Q2 Q3	A B 00	01	11	10	Z
000	100	100	101	101	0
100	110	110	101	101	0
101	100	100	111	111	0
110	110	110	111	101	1
111	100	110	111	111	1
	D1 D2 D3				

De logiska uttrycken för D1, D2 och D3 fås fram via Karnaugh-diagram.

D1 = f(Q1, Q2, Q3, A, B)
D2 = f(Q1, Q2, Q3, A, B)
D3 = f(Q1, Q2, Q3, A, B)

5 variabler!!

- **Karnaugh-diagram för *next-state* logik**



- **Logiska uttryck för *next-state* logik**

D1 = 1
D2 = Q1·Q3'·A' + Q3·A + Q2·B
D3 = A

- **Logiskt uttryck för utgångslogiken**

Z = Q2