

## Oblig nr. 2

# Digital kodelås

### INF-1400

Kode: 331

Kodelåsen fungerer for tallene 1-7, det er mulig å holde inne knappen så lenge han ønsker uten at dette medfører et problem. Kodelåsen er også designet slik at brukeren kan ha to like tall etter hverandre i koden.

Tilstandstabell:

	Nåværende tilstand		In-put			Neste tilstand		Out-put
	$Q_0(t)$	$Q_1(t)$	$X_0$	$X_1$	$X_2$	$Q_0(t+1)$	$Q_1(t+1)$	$y$
Tilstand 00	0	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0
	0	0	0	1	0	0	0	0
	0	0	0	1	1	0	1	0
	0	0	1	0	0	0	0	0
	0	0	1	0	1	0	0	0
	0	0	1	1	0	0	0	0
	0	0	1	1	1	0	0	0
Tilstand 01	0	1	0	0	0	0	0	0
	0	1	0	0	1	0	0	0
	0	1	0	1	0	0	0	0
	0	1	0	1	1	1	1	0
	0	1	1	0	0	0	0	0
	0	1	1	0	1	0	0	0
	0	1	1	1	0	0	0	0
	0	1	1	1	1	0	0	0
Tilstand 11	1	0	0	0	0	X	X	X
	1	0	0	0	1	X	X	X
	1	0	0	1	0	X	X	X
	1	0	0	1	1	X	X	X
	1	0	1	0	0	X	X	X
	1	0	1	0	1	X	X	X
	1	0	1	1	0	X	X	X
	1	0	1	1	1	X	X	X
Tilstand 10	1	1	0	0	0	0	0	0
	1	1	0	0	1	0	0	1
	1	1	0	1	0	0	0	0
	1	1	0	1	1	0	0	0
	1	1	1	0	0	0	0	0
	1	1	1	0	1	0	0	0
	1	1	1	1	0	0	0	0
	1	1	1	1	1	0	0	0

### Karnaugh-diagram:

$X_0X_1X_2$	000	001	011	010	110	111	101	100
$Q_aQ_b$								
00	0	0	0	0	0	0	0	0
01	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	x	x	x	x	x	x	x	x

$$D_b = Q_a' Q_b X_0' X_1 X_2$$

$X_0X_1X_2$	000	001	011	010	110	111	101	100
$Q_aQ_b$								
00	0	0	1	0	0	0	0	0
01	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	x	x	x	x	x	x	x	x

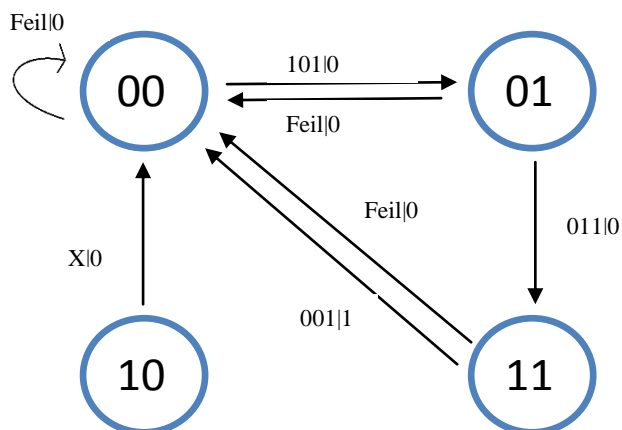
$$D_b = Q_b' X_0' X_1 X_2$$

$X_0X_1X_2$	000	001	011	010	110	111	101	100
$Q_aQ_b$								
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0
10	x	x	x	x	x	x	x	x

$$D_b = Q_a X_0' X_1' X_2$$

Den kombinatoriske logikken ligger i beholderen «Kombinatorisk logikk» i logisim.

### Tilstandsdiagram:



Istedenfor å måtte liste opp alle kombinasjonene som ikke er riktig så har jeg valgt å skrive «Feil». Dette representerer da alle taste - trykk som ikke er den riktige i den tilstanden låsen befinner seg i. X i tilstand (10) er don't care, dvs. at uansett hvilken knapp som blir trykket til den neste tilstanden være (00).

## Flipp-flopper:

JK-flippflopp, tilstandstabell:

J	K	Q(t+1)
0	0	Q(t)
0	1	0 (reset)
1	0	1 (set)
1	1	Q'(t)

Problemet med å bruke D-flipp-flopper som minneelement i tilstandsmaskinen er at brukeren kun har et klokkesignal på å trykke inn neste tall i koden. Endret derfor tilstandsmaskinen til å inneholde JK-flipp-flopper istedenfor. Da utgangsverdien her er låst så lenge inngangssignalet er 0 på begge inngangene fører dette til at brukeren har bedre tid på å trykke inn neste tall i koden

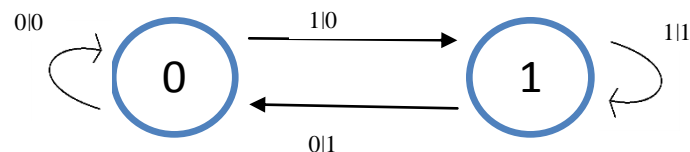
## Venter:

Venteren er en enkel tilstandsmaskin som venter med å sende ut signalet den får inn frem til alle signalene er 0 igjen. Med andre ord signalet blir ikke sendt ut før etter at brukeren har sluppet knappen.

Tilstandstabell:

Q(t)	In	Q(t+1)	Ut
0	0	0	0
0	1	1	0
1	0	0	1
1	1	1	0

Tilstandsdiagram:



Karnaugh-diagram:

$\frac{In}{Q(t)}$	1	1
0	0	1
1	0	1

$Q(t+1) = In$

$\frac{In}{Q(t)}$	1	1
0	0	0
1	1	0

$Ut = Q(t) + In'$

Denne tilstandsmaskinen kan kun ta inn et signal av gangen og det må derfor være en slik tilstandsmaskin til hvert inngangssignal. Beholderen «venter» inneholder 5 av disse tilstandsmaskinene siden hvert av inngangssignaler trenger sin egen venter for ellers blir det blitt sendt ut et signal som ikke er (000) rest tilstanden, men heller den inverterte til det signalet vi egentlig ønsker. Dette ville medført at koden alltid ville vært feil.

Ved hjelp av noen enkle justeringer slik som «venter» og en KJ-flipp-flopp fremfor en D-flipp-flopp har kodelåsen blitt mye mer brukervennlig ved at det ikke ligger et tidspress på brukeren når man trykker inn en knapp og ved at koden kan inneholde to like tall etter hverandre.