

# Obliq nr 2

## Digital kodelås

### INF-1400



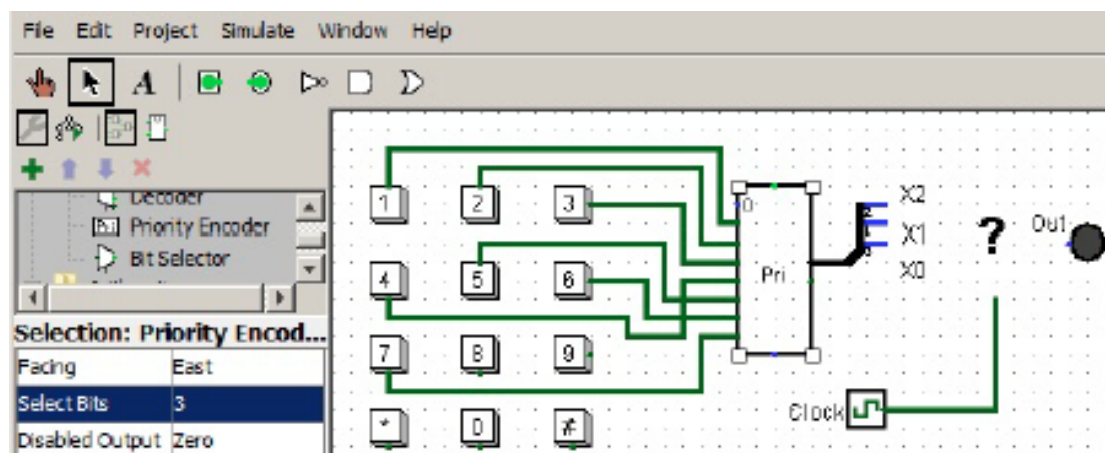
#### Faglig mål

Å forstå hvordan man kan bruke tilstandsmaskin (-konseptet) til å løse sekvensielle oppgaver. Å få en ide om hvordan logikken i en digital kodelås kan lages.

#### Oppgave

I denne oppgaven skal dere designe logikken i en digital kodelås med lignende funksjonalitet som det vi finner på vanlige digitale dørlåser. Et slikt digitalt system kan lages på flere måter, men i denne oppgaven bruker vi vanlig sekvensiell logikk basert på tilstandsmaskin.

Skal man lage en digital dørlås med full funksjonalitet og god brukervennlighet basert på sekvensiell logikk, får man et digitalt system som er litt for stort til at det er hensiktsmessig å designe det for hånd. Det man da bruker er vanligvis VHDL eller andre høynivåhjelpemidler. Hittil i kurset har dere ikke lært å bruke slike hjelpemidler, men i denne oppgaven kan dere bruke den innebygde kretsgeneratoren i LogiSim til å generere logikken i tilstandsmaskinen basert på en tilstandstabell som dere spesifiserer for hånd. For å ikke få en altfor stor tilstandstabell burde dere gjøre en god del forenklinger av funksjonaliteten, samt reduserer brukervennligheten noe.



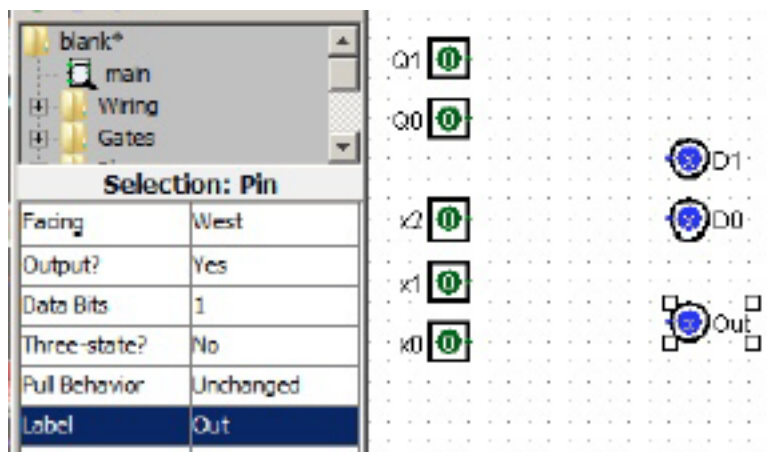
#### Spesifikasjon for systemet:

- Koden du velger skal bestå av 3 siffer
- Koden skal ikke inneholde tallet 0 (dette for å gjøre oppgaven enklere da enkoderen kan settes opp til å gi ut 0,0,0 for tilstanden 0,0,0,0,0,0,0 inn, dvs. ingen tast trykket ned)
- Koden skal ikke inneholde tallene 8 og 9. Dette for å redusere lengden på tilstandstabellen
- To nabosiffer i koden skal ikke inneholde samme tall. Dette for å slippe å håndtere situasjonen der brukeren holder nede samme tast over flere klokkeperioder.

- Klokkesignalet settes til 8Hz. Ved denne lave frekvensen kan vi kreve at brukeren holder hver tast inne i minst 1/8 sekund.
- Utgangssignalet fra tilstandsmaskinen består av ett bit. Dette skal være høyt over en klokkeperiode kun når 3 riktige tall er tastet inn i riktig rekkefølge, ellers skal det være lavt. Dette signalet går til en lysdiode samt releet som styrer selve dørlåsen (ikke en del av oppgaven i år).
- Inngangssignalet til tilstandsmaskinen består av 3 bit - X0, X1 og X2 (utgangen av enkoderen)
- Klokkegating\* er ikke lov.

### Anbefalt fremgangsmåte

- Definer inngangene, utgangene samt tilstandene. Det anbefales å ikke bruke mer enn 4 tilstander for å redusere størrelsen på tabellen
- Velg tilstandskoder, og tegn tilstandsdiagram
- For å designe den kombinatoriske delen av tilstandsmaskinen gjør du følgende: Åpne opp et nytt design i



- LogiSim, legg inn innganger og utganger til den kombinatoriske logikken. Gi inngangene og utgangene fornuftige navn ved bruk av "Label" slik som illustrert i figuren over.
- Åpne opp menyen "Project" - "Analyze Circuit". Sjekk at inngangsnavnene og utgangsnavnene du har definert ligger under "Inputs" / "Outputs" i vinduet "Combinatorial Analysis".
- Under "Table" finner du tilstandstabellen din. Her klikker du inn de utgangsverdiene du vil ha. Trykk "Build Circuit" og den kombinatoriske delen av tilstandsmaskinen din blir automatisk generert.
- Legg inn tastene, encoderen med splitter, lysdioden og klokkegeneratoren, samt D FlipFloppene i

Q1	Q0	x2	x1	x0	D1	D0	Out
0	0	0	0	0	x	x	x
0	0	0	0	1	x	1	0
0	0	0	1	0	x	x	x
0	0	0	1	1	x	x	x
0	0	1	0	0	x	x	x
0	0	1	0	1	x	x	x
0	0	1	1	0	x	x	x
0	0	1	1	1	x	x	x
0	1	0	0	0	x	x	x
0	1	0	0	1	x	x	x
0	1	0	1	0	x	x	x
0	1	0	1	1	x	x	x

tilstandsmaskinen.

- Velg ”Simulate” – ”Tick Frequency” samt ”Simulate” – ”Tick Enabled”. Trykk inn koden du har valgt, og sjekk at systemet virker.

\* *Når man designer et slikt system kan det være fristende å legge inn logikk i selve klokkesignalet (klokkegating). Dette er en meget uheldig designpraksis da all logikk medfører portforsinkelse som i dette tilfelle øker sjansen for at visse deler av systemet blir asynkront med resten. I tillegg er en av de store utfordringene ved design av komplekse systemer nettopp å beholde oversikten. Klokkegating vil redusere oversikten betraktelig. Hvis man har behov for å ”skru av” FlipFlopper i visse perioder, bruker man heller ”Enable” inngangen på FlipFloppene til dette.*

Husk man kan kopiere og lime inn deler av logikk mellom forskjellige designvinduer slik at man kan lagre hele systemet i et vindu mens man genererer den kombinatoriske logikken temporært i et annet vindu. Dette kan synes litt tungvint, men LogiSim er ikke verdens mest profesjonelle designprogram – fordelene er at LogiSim er gratis, og meget lett å lære og å komme i gang med.

### **Levering av oblig**

Når kodelåsen virker som den skal, lagres designet til fil (.circ). Denne filen pluss en rapport som inneholder tilstandsdiagram, tilstandstabellen, karnaughdiagram og fremgangsmetode, leveres inn gjennom DEVILRY. Når dere leverer, så skal ALLE filer leveres i SAMME innlevering på Devily. Gruppelærer kommer bare til å rette siste innlevering. Obligfristen denne gangen kommer til å være hard, som vil si at det ikke er mulighet til å levere etter fristen. Rapporten skal være en PDF-fil, og den skal hete "oblig2\_<brukernavnet ditt>.pdf". Kall kretsen "kodelaas\_<brukernavnet ditt>.circ". Ikke bruk æ, ø, eller å i fil-navnene.

### **En ”profesjonell” løsning (frivillig)**

For de av dere som har lyst å lage en fullt ut funksjonell og brukervennlig kodelås er også dette lov. En slik lås bør kunne ha alle tall 0-9 tilgjengelig i koden (4bit encoder), samt tåle to like etterfølgende tall i koden. Brukeren bør også kunne holde inn hver knapp så kort eller lang tid han/hun vil (krever hurtigere klokkefrekvens). En slik fullverdig løsning vil nok medføre flere tilstander, lengre tilstandstabell, bruk av ”Enable” inngangen på FlipFloppene samt ”Group Select” (Valid) utgangen på encodere.

Lykke til ☺