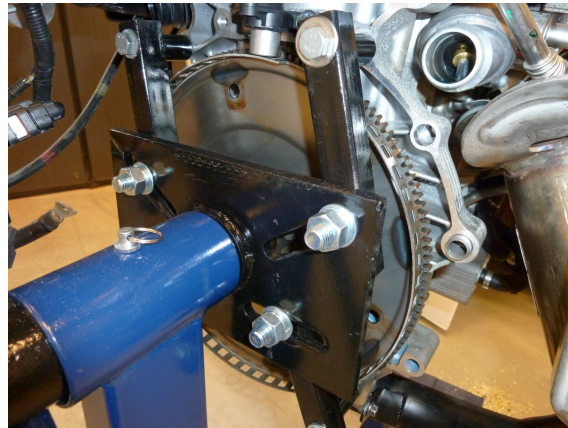
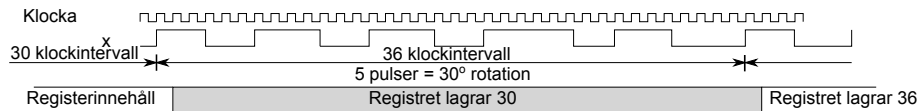


# Lektionsuppgifter om speciella sekvenskretsar



Figur 1: Svänghjul med perforeringar som syns tydligast i underkant.



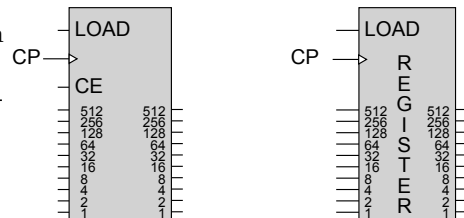
Figur 2: Exempel på hur kretsen ska fungera.

**Uppgift 1.** Motorns varvtal (rotationshastighet) är bland de viktigaste mätsignalerna för styrning och övervakning av förbränningsmotorer. Figur 1 visar ett svänghjul på en motor. Motorns varvtal mäts genom att en Hallsensor känner av svänghjulets perforeringar som syns tydligast på hjulets underkant. Sensorn skickar ut 1 under tiden då den passerar förbi utstansade hål och 0 annars. Det är 60 hål som är jämnt fördelade över varvet, vilket betyder att hjulet roterar  $360/60 = 6^\circ$  från det att ett hål detekteras till dess att nästa hål detekteras.

Klockfrekvensen är 100kHz och motorns hastighet mäts genom att räkna antalet klockintervall som det tar för hjulet att rotera  $30^\circ$ . Hallgivarens utsignal  $x$  är synkroniserad och kan se ut som i figur 2. Under det markerade tidsintervallet, i detta fall 36 klockintervall långt, så har svänghjulet roterat exakt  $30^\circ$  eftersom det har inkommit 5 pulser på  $x$ . Pulserna i  $x$  har varierande längd, beroende på motorns momentana rotationshastighet.

Er uppgift är att konstruera en synkron sekvenskrets som tar  $x$  som insignal och lagrar antalet klockintervall det tar att rotera  $30^\circ$  i ett register. Uppdatering av registret ska ske efter varje gång svänghjulet roterat ytterligare  $30^\circ$ . Det korrekta registerinnehållet för exemplet i figur 2 ses längst ner i figuren. Notera att när hjulet har roterat  $30^\circ$  ska tidtagningen för nästa  $30^\circ$ -rotation starta omedelbart. Till ert förfogande har ni:

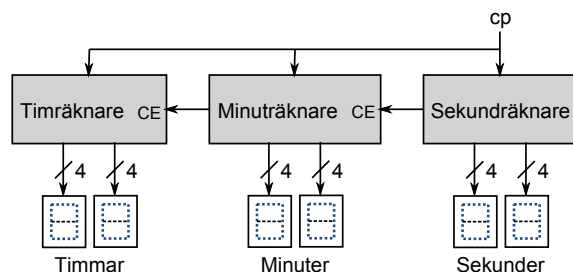
- 10-bitars räknaren till höger med count enable och load
- 10-bitars registret till höger med parallell in parallell ut
- en valfri 4-bitars räknare
- valfria grindar och vippor samt inverterare.



Ni får anta att antalet klockintervall som ska räknas och lagras är mellan 10 och 700, vilket motsvarar motorns arbetsområde. Ange hur valda komponenter ska initieras vid start av systemet.

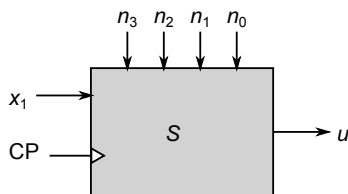
**Uppgift 2.** En digital klocka kan konstrueras enligt blockdiagrammet i figur 4. Er uppgift är att konstruera sekundräknaren. Insignalen till sekundräknaren är klocksignalen  $cp$  med frekvensen 1 Hz. Utsignalerna är dels de två fyrabitars BCD-kodade tal som styr de två 7-segmentsdisplayerna som ska visa sekunder och dels en rippel carry signal som ska vara kopplad till enable ingången på minuträknaren. Enableingången antas vara aktivt hög.

Till ert förfogande har ni två 4-bitarsräknare och valfria grindar och inverterare. Observera att ni inte får använda dekadräknare i uppgiften.



Figur 3: Blockdiagram av en digital klocka.

**Uppgift 3.** På ingången  $x$  inkommer synkroniserade pulser av varierande längd till ett synkront sekvensnät,  $S$ . Nätet har dessutom fyra ingångar på vilka ett binärkodat tal  $n = (n_3, n_2, n_1, n_0)$ , ( $1 \leq n \leq 15$ ), kan ställas in. Konstruera  $S$  så, att det för var  $n$ :te ingångspuls genererar en utgångspuls av längden  $n$  på utgången  $u$ . För övrigt ska  $u = 0$ . Utgångspulsen behöver inte starta i samma klockintervall som den  $n$ :te ingångspulsen. Däremot ska ingångspulser kunna räknas också under den tid  $u = 1$ .



Figur 4: Sekvenskrets med in- och utgångar.

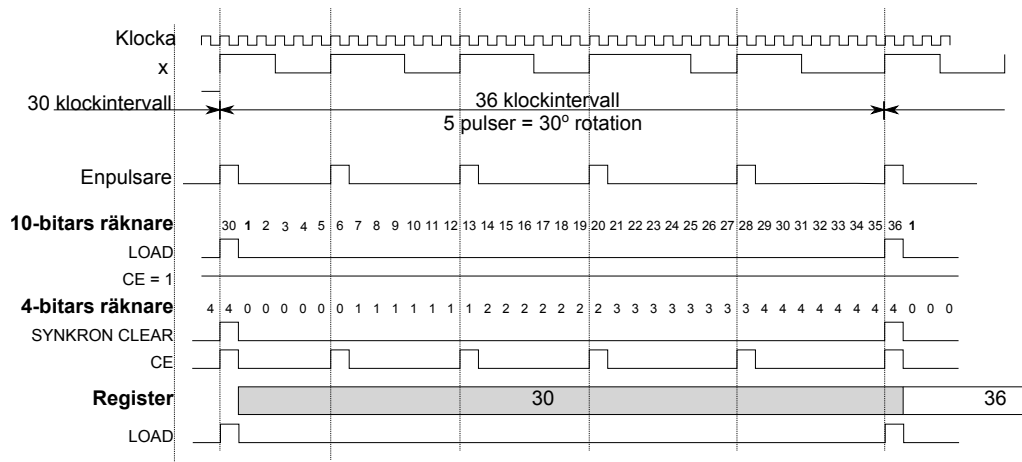
Exempel för  $n = 3$ :

```
x 0 1 1 1 0 0 1 0 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0
u 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0
```

Tillåtna komponenter är 4-bits binärräknare av valfria typer samt valfria grindar och vippor.

# Facit

## Uppgift 1. Tidsdiagram för funktionen



LOAD och CLEAR antas som vanligt vara prioriterade före count enable CE. Låt utsignalen från enpulsaren kallas för  $y$  och tillståndet i 4-bitars räknaren  $q = (q_3, q_2, q_1, q_0)$  där  $q_0$  är minst signifikant bit. Enpulsaren beskrivs av

$$y(t) = x(t)x'(t - 1)$$

10-bitars räknaren:

$$\begin{aligned} \text{LOAD} &= yq_2 \\ \text{CE} &= 1 \end{aligned}$$

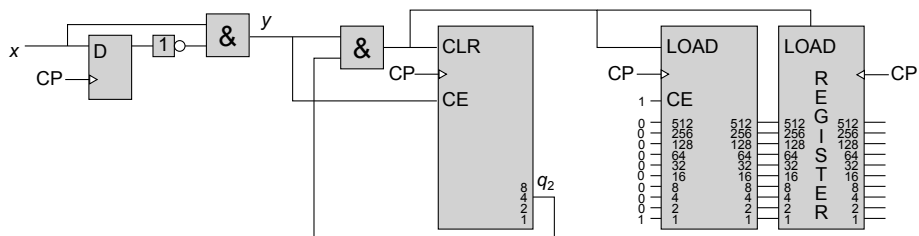
4-bitars räknaren:

$$\begin{aligned} \text{CLEAR} &= yq_2 \\ \text{CE} &= y \end{aligned}$$

Registret:

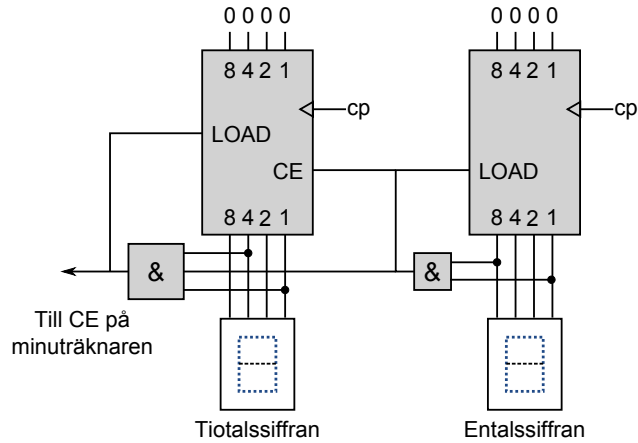
$$\text{LOAD} = yq_2$$

Exempel på hur kretsen kan konstrueras.



Vid start ska D-vippan nollställas, 4-bitars räknaren ställas så att  $q_2 = 1$  och 10-bitars räknaren kan vara i godtyckligt tillstånd.

**Uppgift 2.** Nedan visas ett exempel på hur kretsen kan konstrueras.



Vid start ska 4-bitarsräknarna vara nollställda. Vid entalssiffran avkodas talet 9 och vid tiotalssiffran talet 59. När entalssiffran är 9 laddas entalsräknaren vid nästa klockintervall till 0 samtidigt som tiotalräknaren räknar upp med 1. När räknarna står på 59 är CE på ingången till minuträkaren 1 och tiotalssiffran nollställs vid nästa klockpuls.

**Uppgift 3.** Använd två reversibla räknare för kretsen den första för att räkna antalet inkommande pulser den andra för att räkna längden på utgående pulser. Låt count enable på räknarna betecknas  $CE_1$  respektive  $CE_2$ . Låt Load vara gemensam loadsignal till båda räknarna och  $q_1$  respektive  $q_2$  räknarnas tillstånd. Följande värden på insignaler, tillstånd och utsignal ger det specificerade in-/ut-signalbeteendet:

x	0	1	1	1	0	0	1	0	1	0	0	1	1	0	1	1	1	0	0	1	1	1	1	1	1	0	
CE1	0	1	0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	
q1	3	3	2	2	2	2	2	1	1	0	3	3	2	2	2	1	1	1	1	0	3	3	3	3	3	3	
Load	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		
q2	0	0	0	0	0	0	0	0	0	0	3	2	1	0	0	0	0	0	0	0	0	0	3	2	1	0	0
u=CE2	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0

Detta ger att

- $CE_1$  är utsignalen från en enpulsare matad med  $x$ .
- Load = 1 omm  $q_1 = 0$ .
- Load = 1 sätter  $q_1 = n$  och  $q_2 = n$ .
- $u = CE_2 = 1$  omm  $q_2 \neq 0$ .

vilket kan realiseras med kretsen

