

# Tentamen

## TSEA22 Digitalteknik

Datum	2022-03-23										
Lokal	T1, U4, U1, T2										
Tid	08-12										
Kurskod	TSEA22										
Kursnamn	Digitalteknik										
Provkod	TEN1										
Antal uppgifter	6										
Kursansvarig	Anders Nilsson										
Lärare som besöker skrivsalen	Anders Nilsson										
Telefon under skrivtiden	013-28 2635										
Besöker skrivsalen	Ca 09 och 11										
Kursadministratör	Maria Hamnér, 013-28 5715										
Tillåtna hjälpmedel	Inga										
Preliminära betygsgränser	<table><thead><tr><th>Poäng</th><th>Betyg</th></tr></thead><tbody><tr><td>41-50</td><td>5</td></tr><tr><td>31-40</td><td>4</td></tr><tr><td>21-30</td><td>3</td></tr><tr><td>0-20</td><td>U</td></tr></tbody></table>	Poäng	Betyg	41-50	5	31-40	4	21-30	3	0-20	U
Poäng	Betyg										
41-50	5										
31-40	4										
21-30	3										
0-20	U										
Visning	Kl 10.00-11.00 13/4 på Anders Nilssons kontor (3B:512) på ISY/DA										

### Viktig information

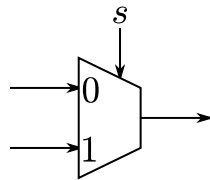
- Tänk igenom din lösning NOGGRANT och använd de lösningsprinciper som kursen förevisar. Okonventionella eller tvetydiga lösningar ger poängavdrag.
- Om inget annat sägs innebär "konstruera" att nätet ska ritas upp, samt att hela lösningsgången måste redovisas.
- AND-, OR-, NAND-, NOR-grindar får ha godtyckligt antal ingångar. XOR- och XNOR-grindar har alltid 2 ingångar, inverterare alltid en ingång.
- "Minimal" tillståndskodning gäller alltid med avseende på vald kodning. Dvs om inget annat sägs behöver inte flera tillståndskodningar provas för att hitta en minimal tillståndskodning. Starttillstånd kan väljas fritt, men ange alltid vad som är starttillstånd.
- Om inget annat sägs kan räknare, D-vippor och dylikt förutsättas vara nollställda vid uppstart.
- Skriv läsbart! Oläsbar text kan inte bedömas och ger därmed inga poäng.

**Lycka till!**

**Uppgift 1. Blandade uppgifter (5p)**

a) Omvandla det decimala talet 323 till binärt. (1p)

b) En 2-1-multiplexer är given enligt figur 1. Använd denna till att dels skapa en tvåingångars AND-grind  $u = ab$ , dels en tvåingångars OR-grind  $u = a + b$ . Det är alltså två olika lösningar som eftersöks. (2p)



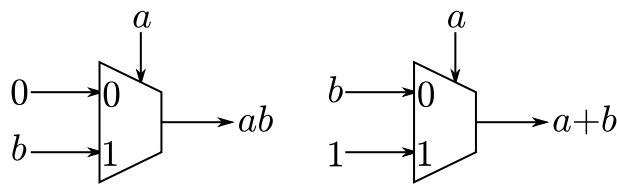
Figur 1: 2-1-multiplexer

c) En T-vippa byter värde på utsignalen om insignalen  $T = 1$ . Skapa en T-vippa med en D-vippa och valfria grindar. (2p)

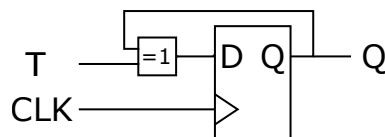
**Lösning.**

a)  $323_{10} = 101000011_2$

b)

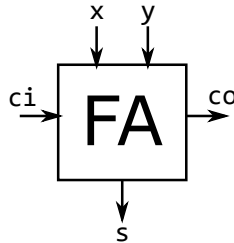


c)



## Uppgift 2. Fulladderaren (5p)

a) En fulladderare (FA), figur 2, adderar bitarna  $x$  och  $y$  tillsammans med ingående carry  $ci$ . Som resultat blir utgående carry  $co=1$  om summan  $s$  är större än 1, annars blir  $co=0$ . Ta fram de minimala uttrycken på fulladderaren FA för de utgående signalerna  $co$  och  $s$  som ett resultat av de inkommande signalerna  $ci$ ,  $x$  och  $y$ . Du behöver inte rita något kopplingsschema. (3p)



Figur 2: Fulladderare

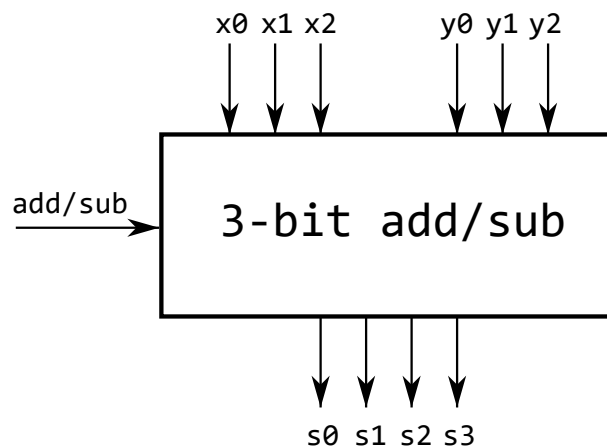
b) Använd fulladderaren tillsammans med valfria grindar och rita kopplingsschemat för en 3-bitars adderare/subtraherare, se figur 3. Subtraktionen  $X - Y = X + Y' + 1$ , alltså  $Y$  kan subtraheras från  $X$  genom att ta fram tvåkomplementet till  $Y$ , dvs invertera talet  $Y$  och addera 1. Signalen  $add/sub$  väljer om addition ( $add/sub=0$ ) eller subtraktion ( $add/sub=1$ ) ska genomföras. Alltså:

$$\text{add/sub}=0 \text{ ska ge resultatet } S = X + Y$$

$$\text{add/sub}=1 \text{ ska ge resultatet } S = X - Y$$

Du behöver bara rita upp kopplingsschemat.

Tips: Enklast är att använda bara fulladderare och XOR-grindar. (2p)



Figur 3: 3-bitars adderare/subtraherare

### Lösning.

a)

$c_i$	$x$	$y$	$c_o$	$s$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$c_o = c_i y + c_i x + xy$$

$$s = c_i x' y' + c_i' x' y + c_i x y + c_i' x y' = c_i \oplus (x \oplus y)$$

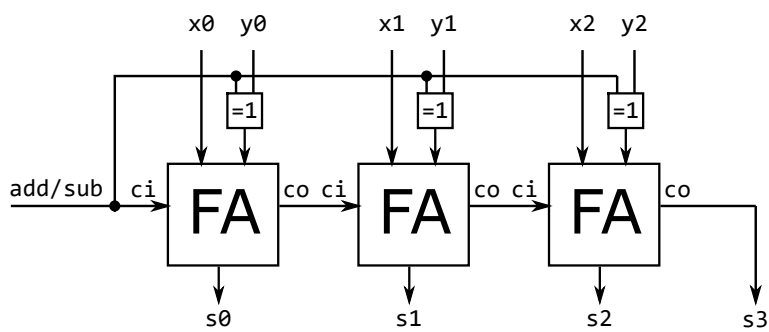
		$xy$			
		00	01	11	10
$c_i$	0	0	0	1	0
	1	0	1	1	1

$c_o = c_i y + c_i x + xy$

		$xy$			
		00	01	11	10
$c_i$	0	0	1	0	1
	1	1	0	1	0

$s = c_i x' y' + c_i' x' y + c_i x y + c_i' x y' = c_i \oplus (x \oplus y)$

b)



### Uppgift 3. Kombinatorik (10p)

Konstruera en kombinatorisk krets som utför beräkningen

$$Y = (y_2, y_1, y_0) = (X^2 \bmod 5) + 1,$$

där  $X$  är ett fyrbitars binärt tal,  $X = (x_3, x_2, x_1, x_0)$ ,  $0 \leq X \leq 9$ . mod 5 är modulo 5, dvs resten vid heltalsdivision med 5. Som ett exempel är  $17 \bmod 5 = 2$  eftersom  $17 - 3 \times 5 = 2$ .

Använd enbart NAND-grindar och inverterare vid realiseringen.

#### Lösning.

$x_3$	$x_2$	$x_1$	$x_0$	$X$	$Y$	$y_2$	$y_1$	$y_0$
0	0	0	0	0	1	0	0	1
0	0	0	1	1	2	0	1	0
0	0	1	0	2	5	1	0	1
0	0	1	1	3	5	1	0	1
0	1	0	0	4	2	0	1	0
0	1	0	1	5	1	0	0	1
0	1	1	0	6	2	0	1	0
0	1	1	1	7	5	1	0	1
1	0	0	0	8	5	1	0	1
1	0	0	1	9	2	0	1	0
1	0	1	0	10	-	-	-	-

Notera att  $y_1 = y'_0$ .

		$x_1x_0$			
		00	01	11	10
$x_3x_2$	00	1	0	1	1
	01	0	1	1	0
	11	-	-	-	-
	10	1	0	-	-

$$y_0 = x_1x_0 + x_2x_0 + x'_2x'_1$$

Om inte  $y_1 = y'_0$  noteras fås  $y_1$  från

		$x_1x_0$			
		00	01	11	10
$x_3x_2$	00	0	0	1	1
	01	0	0	1	0
	11	-	-	-	-
	10	1	0	-	-

$$y_2 = x_1x_0 + x_3x'_0 + x'_2x_1$$

		$x_1x_0$			
		00	01	11	10
$x_3x_2$	00	0	1	0	0
	01	1	0	0	1
	11	-	-	-	-
	10	0	1	-	-

### Uppgift 4. Kodlås (10p)

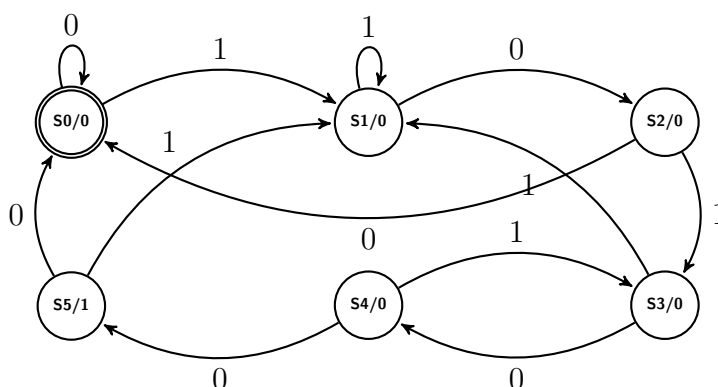
Ett enkelt kodlås som tar en binär sekvens som kod ska konstrueras. Koden matas in bit för bit på ingång  $x$  i takt med systemklockan. När rätt kod matats in ska utsignalen  $u$  bli 1 (låset öppet). Låset ska därefter låsa sig (dvs  $u=0$ ) så snart en ny bit matas in. Koden för att öppna kodlåset ska vara 1,0,1,0,0. Kodlåset ska klara gissningar med överlappande sekvenser. En exempelsekvens ses nedan:

$x$  : 0 1 1 0 1 0 0 1 0 1 1 0 1 0 0 0 1 0 1 0 1 0 0 0 0  
 $u$  : 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0

Du behöver inte tänka på hur bitarna matas in på ingången utan kan anta att det kommer en ny bit varje klockcykel.

Du får använda valfria grindar för att konstruera kodlåset. Tänk på att visa hela lösningsgången.

**Lösning.** Tillståndsdigram (moore) med bågmarkeringar  $x$ .



#### Kodning alternativ 1: Binär

Starttillståndet är  $S_0$ ,  $Q = 000$ .

	$q_2 q_1 q_0$	$q_2^+ q_1^+ q_0^+$		$u$
		$x = 0$	$x = 1$	
$S_0$	000	000	001	0
$S_1$	001	010	001	0
$S_2$	010	000	011	0
$S_3$	011	100	001	0
$S_4$	100	101	011	0
$S_5$	101	000	001	1
-	110	—	—	-
-	111	—	—	-

$$q_2^+ = q_2 q_0' x' + q_1 q_0 x'$$

$$q_1^+ = q_2 q_0' x + q_1 q_0' x + q_2' q_1' q_0 x'$$

$$q_0^+ = q_2 q_0' + x$$

$$u = q_2 q_0$$

		$q_0x$			
		00	01	11	10
$q_2q_1$	00	0	0	0	0
	01	0	0	0	1
	11	-	-	-	-
	10	1	0	0	0

$$q_2^+ = q_2q_0'x' + q_1q_0x'$$

		$q_0x$			
		00	01	11	10
$q_2q_1$	00	0	0	0	1
	01	0	1	0	0
	11	-	-	-	-
	10	0	1	0	0

$$q_1^+ = q_2q_0'x + q_1q_0'x + q_2'q_1'q_0x'$$

		$q_0x$			
		00	01	11	10
$q_2q_1$	00	0	1	1	0
	01	0	1	1	0
	11	-	-	-	-
	10	1	1	1	0

$$q_0^+ = q_2q_0' + x$$

		$q_0x$			
		00	01	11	10
$q_2q_1$	00	0	0	0	0
	01	0	0	0	0
	11	-	-	-	-
	10	0	0	1	1

$$u = q_2q_0$$

### Kodning alternativ 2: Gray

Starttillståndet är  $S_0$ ,  $Q = 000$ .

	$q_2q_1q_0$	$q_2^+q_1^+q_0^+$		$u$
		$x = 0$	$x = 1$	
$S_0$	000	000	001	0
$S_1$	001	011	001	0
$S_3$	010	110	001	0
$S_2$	011	000	010	0
-	100	---	---	-
-	101	---	---	-
$S_4$	110	111	010	0
$S_5$	111	000	001	1

$$q_2^+ = q_1q_0'x'$$

$$q_1^+ = q_2q_0' + q_1q_0'x' + q_1'q_0x' + q_2'q_1q_0x$$

$$q_0^+ = q_1'x + q_1'q_0 + q_2q_0'x' + q_2'q_0'x + q_2q_0x$$

$$u = q_2q_0$$



		$q_0x$			
		00	01	11	10
$q_2q_1$	00	0	0	0	0
	01	1	0	0	0
	11	1	0	0	0
	10	-	-	-	-

$q_2^+ = q_1q_0x'$

		$q_0x$			
		00	01	11	10
$q_2q_1$	00	0	0	0	1
	01	1	0	1	0
	11	1	1	0	0
	10	-	-	-	-

$q_1^+ = q_2q_0' + q_1q_0'x' + q_1'q_0x' + q_2'q_1q_0x$

		$q_0x$			
		00	01	11	10
$q_2q_1$	00	0	1	1	1
	01	0	1	0	0
	11	1	0	1	0
	10	-	-	-	-

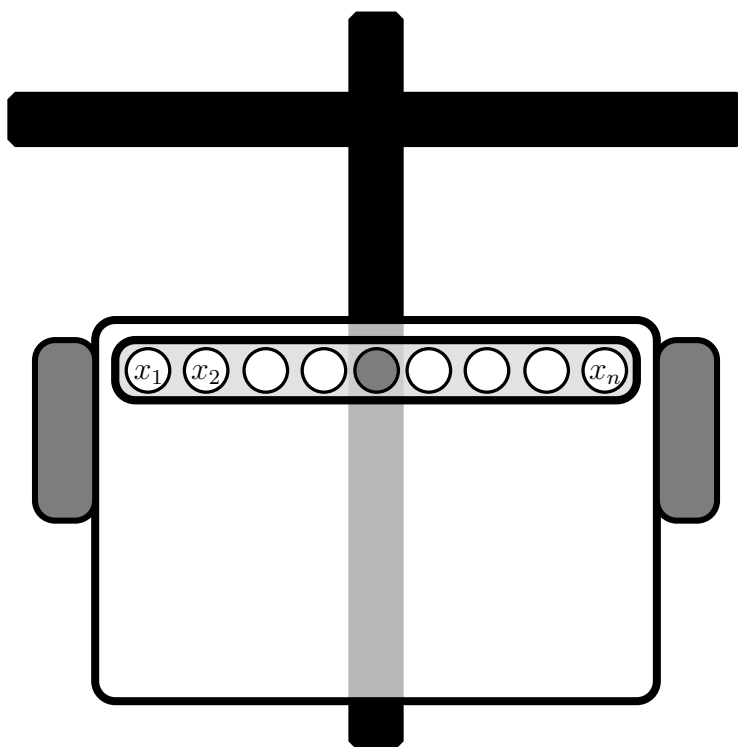
$q_0^+ = q_1'x + q_1'q_0 + q_2q_0'x' + q_2'q_0'x + q_2q_0x$

		$q_0x$			
		00	01	11	10
$q_2q_1$	00	0	0	0	0
	01	0	0	0	0
	11	0	0	1	1
	10	-	-	-	-

$u = q_2q_0$

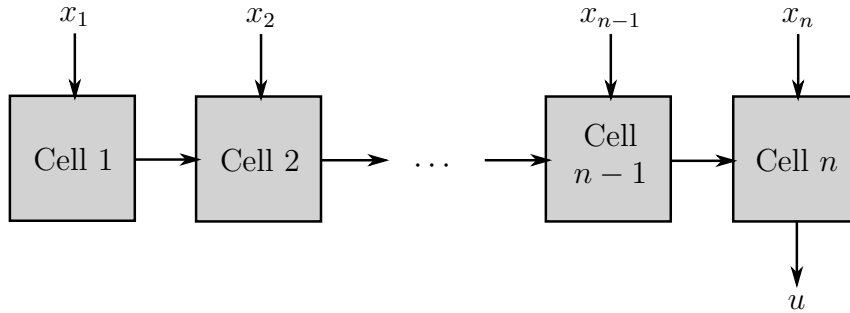
### Uppgift 5. Linjeföljande robot. (10p)

En linjeföljande robot ska följa svarta linjer på ett golv enligt figur 4. För att göra det har roboten en rad med  $n$  sensorer,  $x_1, x_2, \dots, x_n$ , riktade mot golvet. En sensor ger  $x_i = 1$  när den befinner sig ovanför en linje och 0 annars. När roboten kör längs en linje detekterar en eller två intilliggande sensorer linjen. I övre delen av figuren syns en korsning. När roboten är i korsningen kommer fler än 2 sensorer detektera linjer och då behövs andra styrstrategier.



Figur 4: Robot med sensorrad för linjeföljning.

Konstruera en iterativ kombinatorisk krets som detekterar normalfallet att roboten körs längs en linje. Kretsen ska ha strukturen visad i figur 5 med utsignal  $u$  som är 1 om och endast om exakt 1 eller 2 intilliggande insignaler är 1. I samtliga andra fall ska utsignalen vara 0.



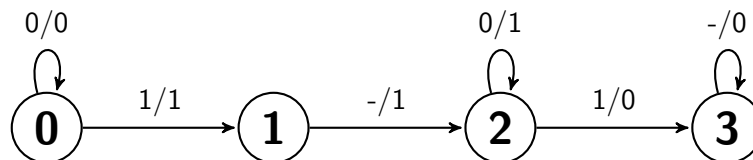
Figur 5: Kretsens struktur, insignaler och utsignal.

För att förtydliga specifikationen ges några exempel för  $n = 8$ , dvs  $x = (x_1, x_2, \dots, x_8)$ , tillsammans med en tolkning av situationen:

- $x = (00000000) \Rightarrow u = 0$  ingen linje under roboten
- $x = (00000010) \Rightarrow u = 1$  normalfall, linje under en sensor
- $x = (01100000) \Rightarrow u = 1$  normalfall, linje under två sensorer
- $x = (01110000) \Rightarrow u = 0$  roboten står något snett i korsning
- $x = (00010001) \Rightarrow u = 0$  roboten roterar i korsning, ser två linjer
- $x = (10010001) \Rightarrow u = 0$  svårtolkad signal, något är fel

Ni har tillgång till AND-, OR-grindar och inverterare och kan anta att  $n \geq 5$ . För full poäng krävs tillståndsdigram med minimalt antal tillstånd, tillståndstabell, minimerade uttryck för alla celler och krettschema med minimerade celler.

**Lösning.** Tillståndsdigram med bågmarkeringar  $x/u$ .



**Kodning alternativ 1: Binär**

Starttillståndet är  $q = 00$ .

$q_1q_0$	$q_1^+q_0^+/u$	
	$x = 0$	$x = 1$
00	00/0	01/1
01	10/1	10/1
11	11/0	11/0
10	10/1	11/0

Cell 1:  $(q_1, q_0) = (0, 0)$

$$\begin{aligned} q_1^+ &= 0 \\ q_0^+ &= x_1 \end{aligned}$$

Cell 2:  $(q_1, q_0) \in \{(0, 0), (0, 1)\}$

$$\begin{aligned} q_1^+ &= q_0 \\ q_0^+ &= q_0'x_2 \end{aligned}$$

Cell 3:  $(q_1, q_0) \in \{(0, 0), (0, 1), (1, 0)\}$

$$\begin{aligned} q_1^+ &= q_1 + q_0 \\ q_0^+ &= q_0'x_3 \end{aligned}$$

Cell  $k \in \{4, \dots, n-1\}$ :

$$\begin{aligned} q_1^+ &= q_1 + q_0 \\ q_0^+ &= q_1q_0 + q_0'x_k \end{aligned}$$

Cell  $n$ :

$$u = q_1'x_n + q_1'q_0 + q_1q_0'x_n'$$

Detta ger  $n$  inverterare,  $2n - 6$  OR-grindar och  $2n - 3$  AND-grindar.

### Kodning alternativ 2: Gray

Starttillståndet är  $q = 00$ .

$q_1q_0$	$q_1^+q_0^+/u$	
	$x = 0$	$x = 1$
00	00/0	01/1
01	11/1	11/1
11	11/1	10/0
10	10/0	10/0

Cell 1:  $(q_1, q_0) = (0, 0)$

$$\begin{aligned} q_1^+ &= 0 \\ q_0^+ &= x_1 \end{aligned}$$

Cell 2:  $(q_1, q_0) \in \{(0, 0), (0, 1)\}$

$$\begin{aligned} q_1^+ &= q_0 \\ q_0^+ &= x_2 + q_0 \end{aligned}$$

Cell 3:  $(q_1, q_0) \in \{(0, 0), (0, 1), (1, 1)\}$

$$\begin{aligned} q_1^+ &= q_0 \\ q_0^+ &= q_0x_3' + q_1'x_3 \end{aligned}$$

Cell  $k \in \{4, \dots, n - 1\}$ :

$$\begin{aligned}q_1^+ &= q_1 + q_0 \\q_0^+ &= q_0 x'_k + q_1 x_k\end{aligned}$$

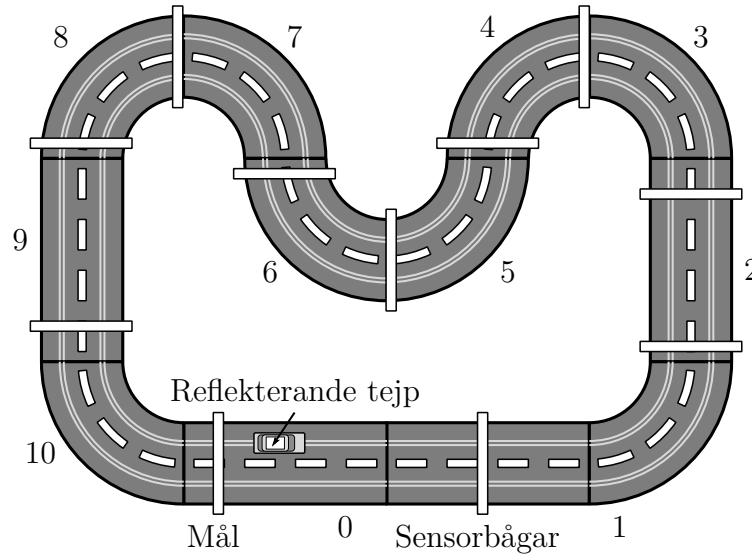
Cell  $n$ :

$$u = q_0 x'_n + q_1 x_n$$

Detta ger  $2n - 4$  inverterare,  $2n - 5$  OR-grindar och  $2n - 4$  AND-grindar.

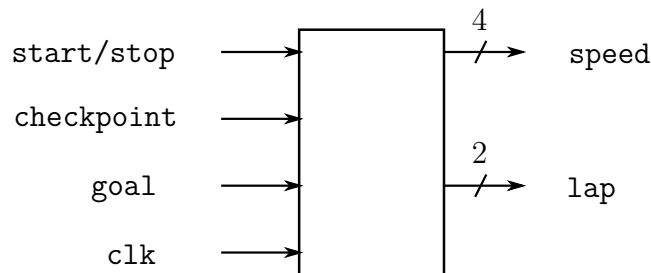
### Uppgift 6. Bilbaneskoj (10p)

Figur 6 visar en bilbana med en bil som ska styras automatiskt med en synkron sekvenskrets som ni ska konstruera. Banan har sensorbågar som delar varvet i segment i figuren numrerade 0-10. Det finns två typer av sensorbågar en speciell vid start/mål och övriga som här kallas för checkpoints.

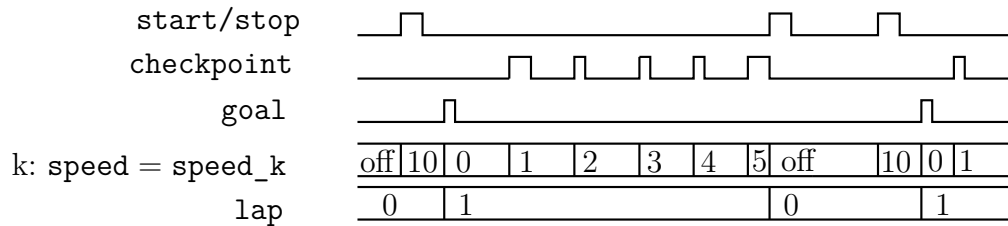


Figur 6: Bilbana med bil och sensorbågar.

Kretsens in- och ut-signaler syns i figur 7. Kretsens insignaler är **start/stop** som kommer från en knapp som slår på eller av körningen, **checkpoint** som blir 1 när bilen passerar en checkpoint (målet räknas inte till checkpoints), **goal** som blir 1 om bilen passerar målet och **clk** som är klocksignal. Kretsens utsignaler är **speed** ett binärkodat 4-bitstal som representerar bilens aktuella gaspådrag och **lap** ett binärkodat 2-bitstal som anger vilket varv som körs.



Figur 7: In- och ut-signaler på den krets som ska konstrueras.



Figur 8: Tidsdiagram som visar exempel på kretsens korrekta beteende.

För att köra fort krävs olika gaspådrag i olika segment. Antag att gaspådraget i segment  $k \in \{0, 1, \dots, 10\}$  ska vara `speed_k`. För att starta bilen ska den först ställas i segment 10 före mål och sedan startas med start/stopp-knappen. Bilen ska stanna om man trycker på start/stopp-knappen när den kör eller om bilen har åkt 3 varv runt banan. Varje gång bilen stannat förutsätts det att bilen manuellt placeras i segment 10 innan en ny körning startas.

Signalerna från knapp och sensorer är osynkroniserade. Ett exempel på hur systemet ska fungera syns i figur 8. Det går hundratals klockpulser under de pulser som visas i figuren. På raden för `speed` betyder `off` att `speed = 0` och där det står till exempel 10 innebär det att `speed = speed_10` där `speed_10` är ett förbestämt gaspådrag för segment 10. När bilen inte kör är `lap = 0`. När den passerar målsensorn första gången räknar den upp till 1 eftersom bilen då har påbörjat det första varvet. Om bilen har kört 3 varv och stängts av automatiskt ska det räcka med ett knapptryck på start/stopp-knappen för att återstarta bilen (underförstått att bilen också manuellt placeras i segment 10).

Konstruera en synkron sekvenskrets som styr bilen enligt ovan. Till ert förfogande har ni valfria grindar, inverterare, D-vippor och ett eller flera PROM med plats för 16 ord på vardera 4:a bitar. Dessutom har ni valfria räknare med upp till 4-bitar. Tillåtna ingångar på räknarna är `clk`, `count enable`, `up/down`, `load`, `data in` och tillåtna utgångar `ripple carry out` och `räknetillståndet`. Istället för att använda ingångarna `load` och `data in` så kan flera `load`-ingångar användas med specificerade datavärden, som till exempel `load 0`, `load 2`, men då måste prioriteten anges, dvs vilken `load`-funktion som slår igenom om flera är aktiva samtidigt. Klockfrekvensen är hög i förhållande till bilens rörelse så det spelar ingen roll om utsignalerna fördröjs några klockpulser. Kretsens initialtillstånd ska anges men den asynkrona kretsen för att initiering behöver inte redovisas. Onödigt komplicerade konstruktioner ger poängavdrag, asynkrona lösningar ger kraftigt reducerad poäng.

**Lösning.** Figur 9 visar ett exempel på en lösning. Alla insignaler synkroniseras och enpulsas. En 2-bitarsräknare räknar varv. Om körningen stoppas nollställs varvräk-naren och om ett nytt varv påbörjas räknar den upp. När det 3:e varvet är slut och körningen klar skickas detta ut med RCO. En 4-bitarsräknare håller reda på om bilen kör och i så fall vilket segment den är i. Här indikerar 0 att bilen inte kör och  $k \neq 0$  att bilen är i segment  $k - 1$ . Räknelägena 12-15 används inte. PROMet används för att generera rätt gaspådrag till respektive segment.

Bilen stannar om bilen körs och start/stopp-knappen aktiveras eller om 3 varv körts klart. Bilen ska starta om bilen inte körs och start/stopp-knappen aktiveras. Ett nytt varv påbörjas när målsensorn reagerar och bilen körs. Segmentet räknas upp om checkpoint går hög och bilen körs. När bilen har kört 3 varv kommer `new_lap` och `3-laps` båda vara 1. För att bilen ska stanna måste LOAD 0 ha prioritet före LOAD 1.

Om räknaren har ingångarna LOAD och data in  $(d_3, d_2, d_1, d_0)$  samt tillståndet  $Q = (q_3, q_2, q_1, q_0)$  så kan de dom uttryckas

$$\text{LOAD} = \text{stop} + \text{3-laps} + \text{new\_lap} + \text{start}$$

$$d_3 = \text{start}$$

$$d_2 = 0$$

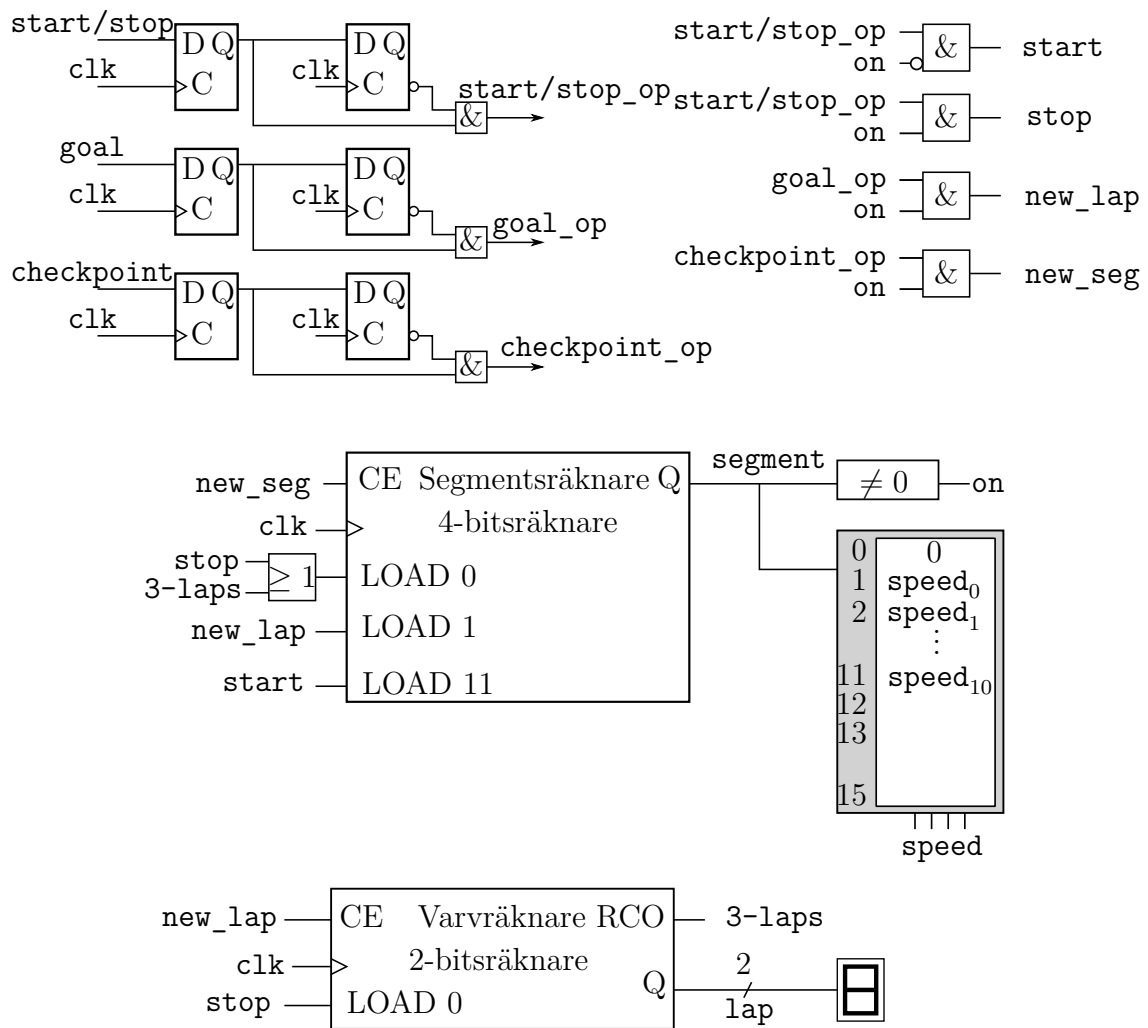
$$d_1 = \text{start}$$

$$d_0 = \text{start} + \text{new\_lap} \cdot \text{3-laps}'$$

$$\text{on} = q_3 + q_2 + q_1 + q_0$$

Den sista termen i  $d_0$  gör att 0 laddas när 3 varv har körts, dvs då `new_lap = 3-laps = 1`. Alla register ska ha initialtillstånd 0 utom synkroniseringsvipporna som inte bör resettas alls.





Figur 9: Kretsens struktur, insignaler och utsignal.