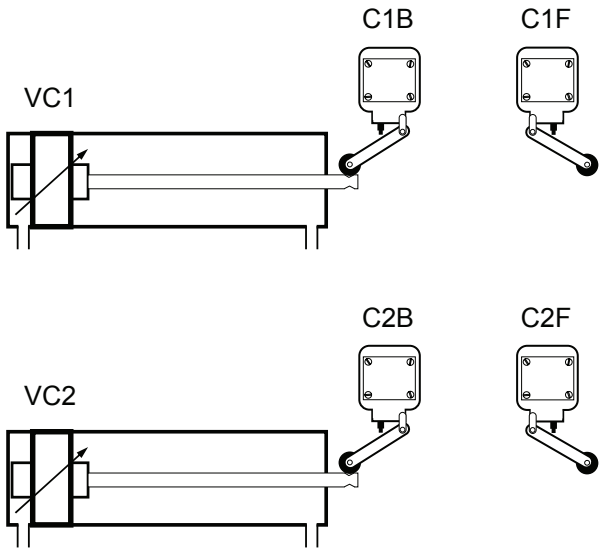
Följddiagram för händelsestyrda rörelser

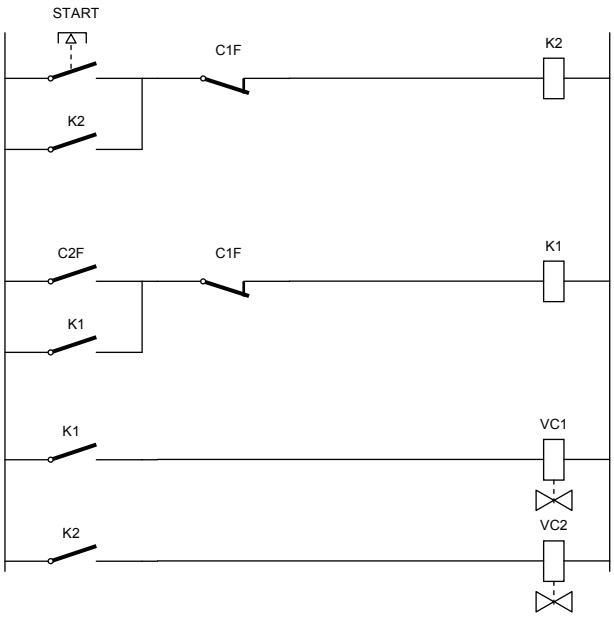
2 STYROBJEKT – UNIKA FASER

Två arbetscylindrar ska röra sig i följande ordning. När man ger startkommando ska kolvstången i cylinder VC2 gå ut. När den har nått sitt yttre ändläge ska den stanna kvar här och kolvstången i cylinder VC1 ska gå ut. När den har nått sitt yttre ändläge ska båda kolvstängerna gå tillbaka till sitt inre ändläge. Ett programvarv (arbetscykel) har utförts.



Fri lösning med reläteknik

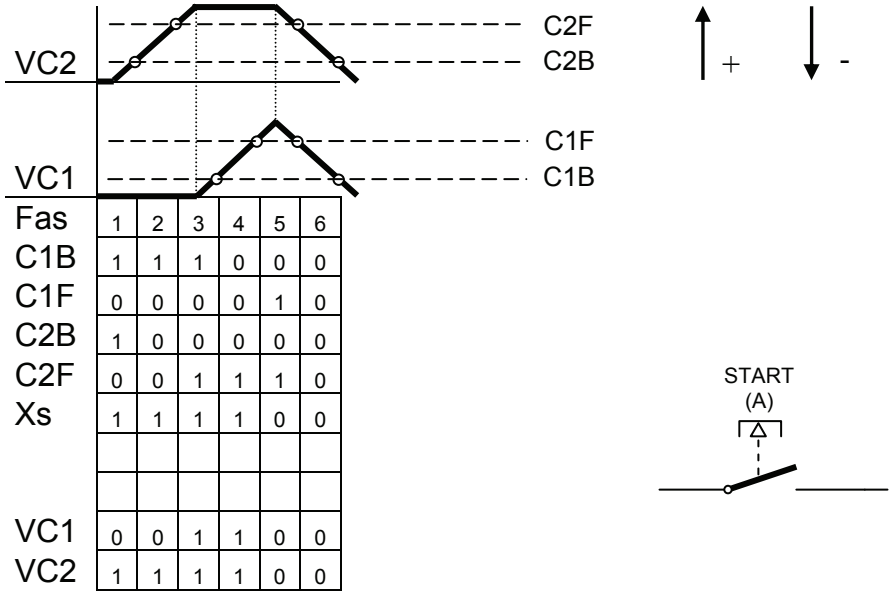
Om man arbetar en stund med reläteknik kan man snart få följande funktion.



Lösning med följddiagram

Uppgiften kan också lösas med ett s k följddiagram. Fördelen med detta är att man har god överblick över alla händelser och kan lätt hantera en stor mängd objekt. Dessutom kan man tillämpa de logiska uttrycken för resp styrobjekt mot olika system som t ex reläteknik, pneumatiska, PLC, dator, mikroprocessor, m fl.

Nu ska vi lösa styrningen med hjälp av ett följddiagram. Man börjar med att rita in de önskade händelserna/rörelserna (VC1 och VC2). Observera när rörelserna börjar resp slutar. Därefter markerar man signalerna (C1B, C1F, C2B och C2F) från de olika rörelserna. Sedan noterar man tillstånden för samtliga signaler. När detta är klart fyller man i de önskade utsignalerna på VC1 och VC2 (längst ned i listan). Nu är signaltabellen klar att bearbetas.



Fas	1	2	3	4	5	6
C1B	1	1	1	0	0	0
C1F	0	0	0	0	1	0
C2B	1	0	0	0	0	0
C2F	0	0	1	1	1	0
Xs	1	1	1	1	0	0
VC1	0	0	1	1	0	0
VC2	1	1	1	1	0	0

Arbetsgång för ett givet händelseförlopp:

1. Rita rörelserna för varje cylinder i ett diagram.
2. Numrera faserna för varje tillstånd.
3. Skriv in samtliga in-variabler för cylindrarna (C1B…C2F).
4. Fyll i ettor (1) och nollor (0) för samtliga in-variabler.
5. Skriv in samtliga ut-variabler (VC1 och VC2) för cylindrarna. Gör detta längst ned.
6. Fyll i ettor (1) och nollor (0) för samtliga ut-variabler.
7. Notera om det finns lika faser.
8. Vid lika faser, arrangera hjälpminne så de blir olika (1 minne gör 2 faser olika, 2 minne gör 3 faser olika, osv).
9. Lägg in ett startminne (Xs). Skapa kod för set respektive reset.
10. Lös de logiska uttrycken för ut-signalerna (cylindrarna).

Xs Startminne

$$SET = A \bullet C1B \bullet \overline{C1F} \bullet C2B \bullet \overline{C2F}$$

$$RST = C1F$$

VC1 Cylinder 1

$$VC1 = \left(C1B \bullet \overline{C1F} \bullet \overline{C2B} \bullet C2F \bullet Xs\right) + \left(\overline{C1B} \bullet \overline{C1F} \bullet \overline{C2B} \bullet C2F \bullet Xs\right)$$

Man läser av varje fas (kolumn) man önskar en etta.
Tillämpa sedan Booles algebra och lagar för att förkorta uttrycket.
Vi får följande uttryck vid förkortning

$$VC1 = \overline{C1F} \bullet \overline{C2B} \bullet C2F \bullet Xs$$

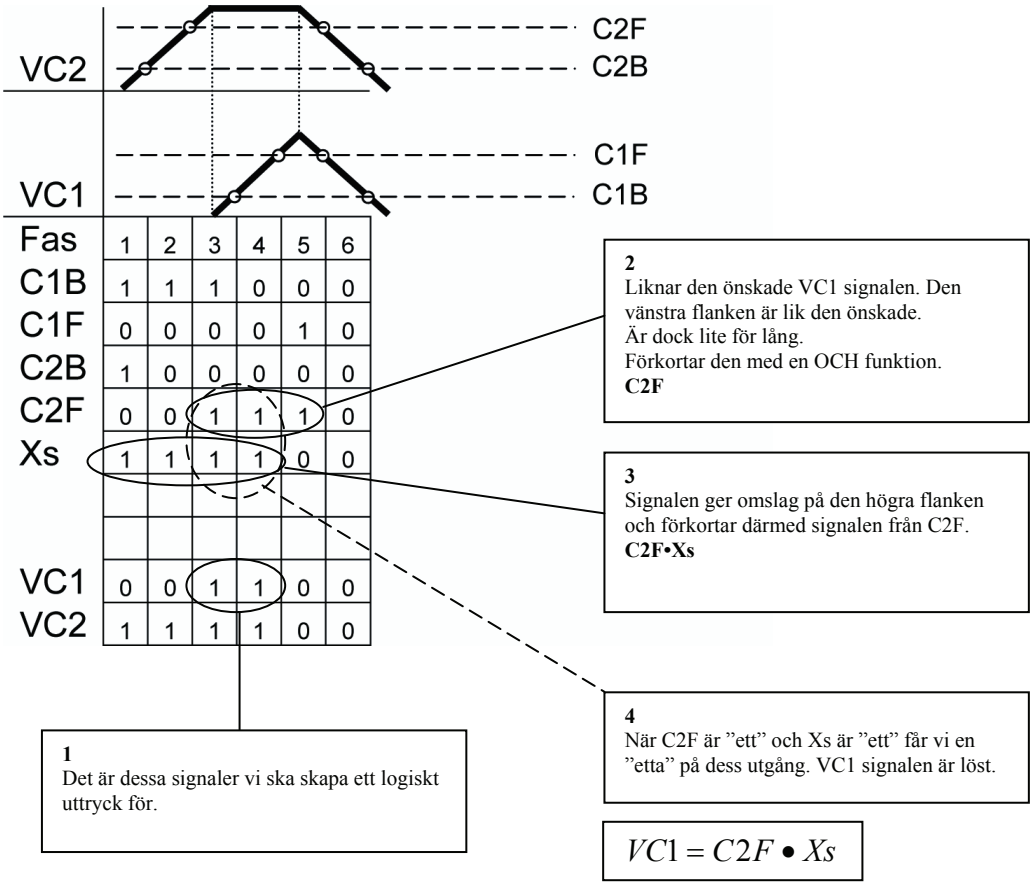
Detta uttryck är dock inte det mest förenklade. Booles algebra och lagar räcker inte till för det.
Det mest förkortade uttrycket är följande
$VC1 = C2F \bullet Xs$

Detta uttryck får man genom följande resonemang.
Leta efter en signal (1 eller 0) som liknar den önskade så långt som möjligt.
Den ska helst börja eller sluta där den önskade signalen börjar/slutar.
Är signalen för lång, vilket den oftast är, förkortar man den med en OCH funktion.
För denna förkortning letar man på den andra sidan av den önskade signalen som ger ett omslag på rätt ställe.

Blir signalen för kort kan man förlänga den med en ELLER funktion.

Nedan visas hur vi tänkt i ovanstående exempel, men först de 2 regler vi talat om.

OCH (AND) **Förkortar**
ELLER (OR) **Förlänger**



Det uttryck vi fick fram tidigare $VC1 = \overline{C1F} \bullet \overline{C2B} \bullet C2F \bullet Xs$ är alltså inte det mest förkortade. Variablerna $\overline{C1F} \bullet \overline{C2B}$ påverkar alltså inte uttrycket. För att kunna lösa ut de mest förkortade logiska uttrycken krävs det träning och rutin.

VC2 Cylinder 2
$VC2 = Xs$

Sammanfattning

Xs Startminne

$SET = A \bullet C1B \bullet \overline{C1F} \bullet C2B \bullet \overline{C2F}$

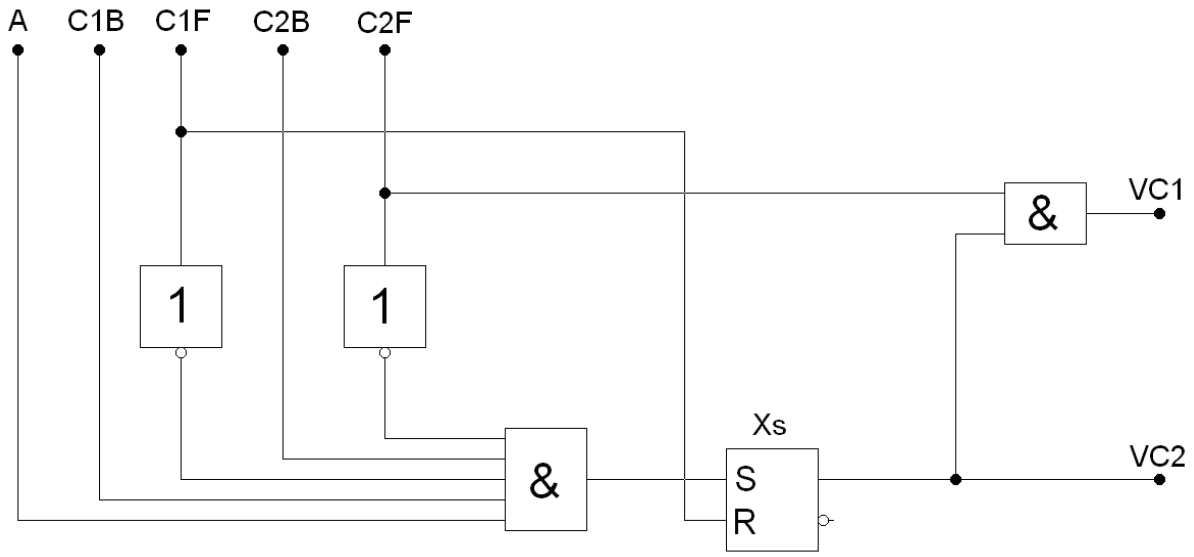
$RST = C1F$

VC1 Cylinder 1

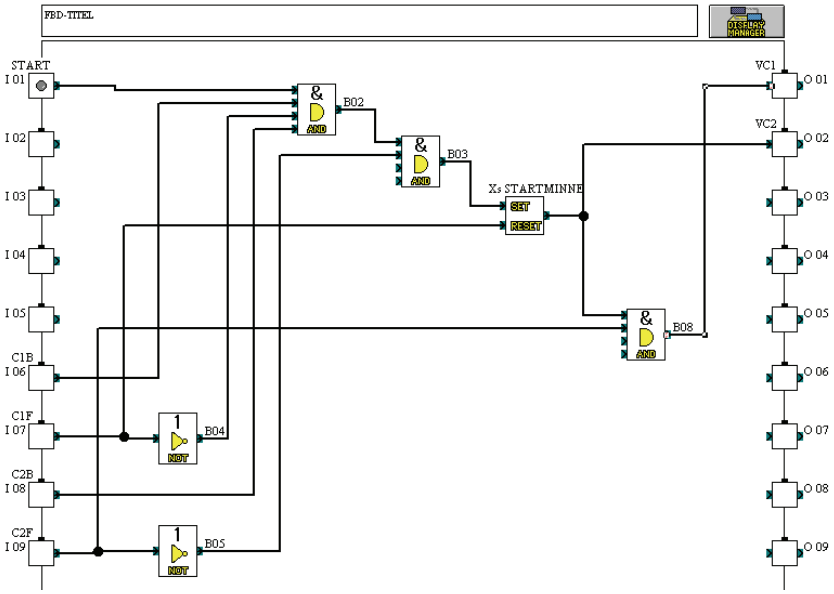
$VC1 = C2F \bullet Xs$

VC2 Cylinder 2

$VC2 = Xs$

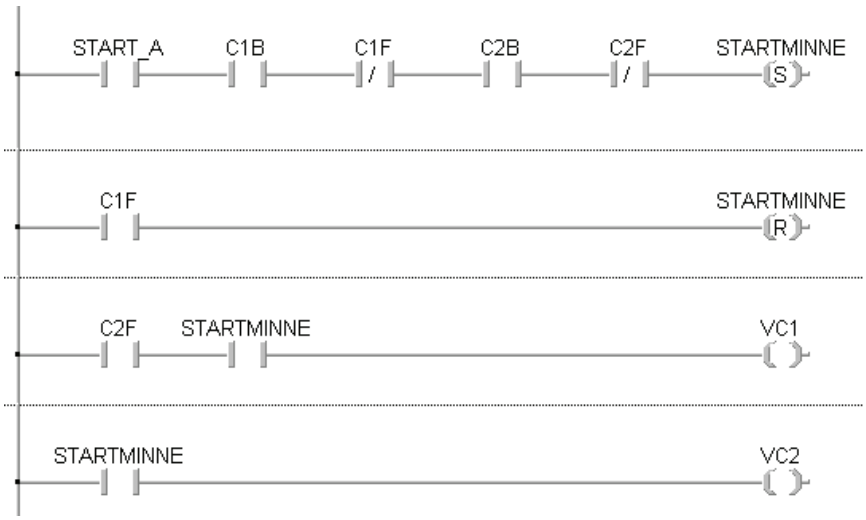


Logiskt kopplingsschema



Lösning i ALPHA

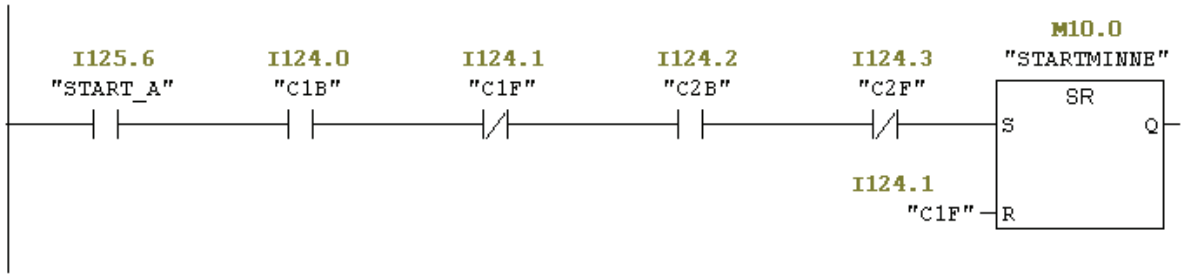
```
LD    X0
AND   X2
ANI   X3
AND   X4
ANI   X5
SET   M1

LD    X3
RST   M1

LD    X5
AND   M1
OUT   Y0

LD    M1
OUT   Y1

END
```



Lösning i IL Instruktionslista
GX IEC Developer
Mitsubishi

Lösning i Ladder
GX IEC Developer
Mitsubishi

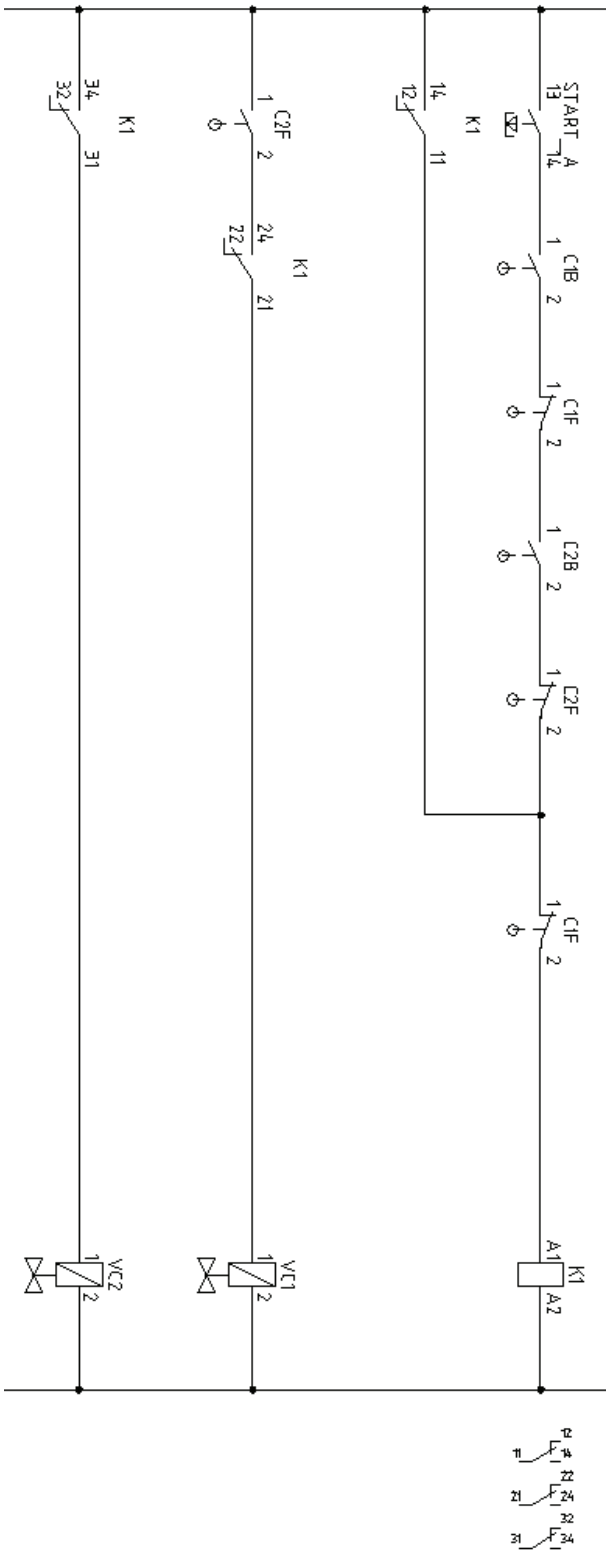
```
FC1 : Title:
Network 1: Title:
```




```
Network 2 : Title:
```


```
Network 3 : Title:
```

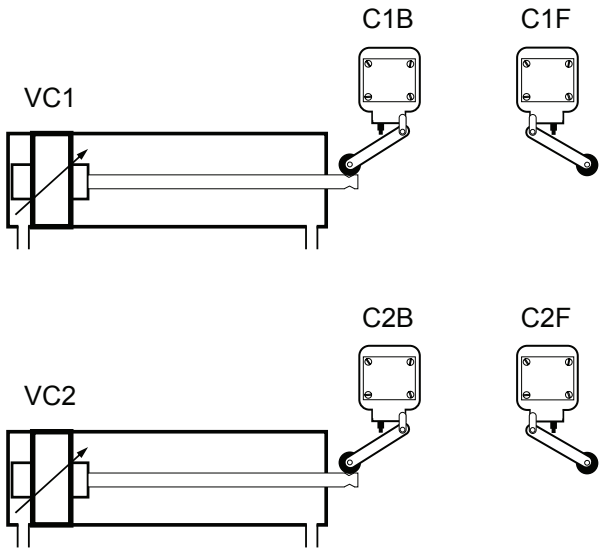
Lösning i Ladder
Step 7
Siemens



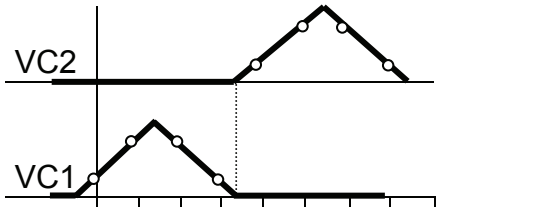
Kopplingsschema för trådbunden logik.
Elprocad

2 STYROBJEKT – LIKA FASER

Nu ska vi ta ett exempel där det finns lika faser. Två arbetscylindrar ska röra sig i följande ordning. När man ger startkommando ska kolvstången i cylinder VC1 gå ut och sedan in igen. När den har nått sitt inre ändläge ska VC2 gå ut och in igen. Ett programvarv (arbetscykel) har utförts.



Vi löser uppgiften med hjälp av ett följddiagram.



	Δ	□		□	Δ	♥		♥
Fas	1	2	3	4	5	6	7	8
C1B	1	0	0	0	1	1	1	1
C1F	0	0	1	0	0	0	0	0
C2B	1	1	1	1	1	0	0	0
C2F	0	0	0	0	0	0	1	0
X1	0	0	1	1	1	1	0	0
Xs	1	1	0	0	0	0	0	0
VC1	1	1	0	0	0	0	0	0
VC2	0	0	0	0	1	1	0	0

Som vi kan se är följande faser lika: 1 och 5, 2 och 4 samt 6 och 8. Detta kontrolleras innan vi fyllt i 1:or och 0:or från X1 och neråt i diagrammet. Dessa måste göras olika så att alla faser blir unika.

Lika faser

För att göra faser olika får vi använda en extern variabel som vi ”stoppar” in i systemet. Vi använder ett minne för denna extra variabel och kallar den för hjälpminne 1 (X1). Denna ska se till att åstadkomma en förändring. Eftersom vi har parvis lika faser räcker det med en extra variabel. Detta eftersom en binär variabel (även kallad Boolsk) kan anta två värden, 0 och 1. Om ser till att vår extra variabel ger en etta eller nolla i fas 1 och motsatta statusen på fas 5 kommer de att bli olika. Så gör vi även för de andra faserna som är lika.

Om vi låter signalen C1F ett-ställa våran extra variabel och C2F nollställa den så kommer man att få denna extra variabel (som vi kallar för X1) att ge ettor i faserna 3-6. Övriga blir noll. Nu är alla faser olika (unika). Vi fyller i dessa 1:or och 0:or för X1 i sanningstabellen.

Not
Om man har 3 faser som är lika fordras 2 extra variabler, eftersom vi arbetar med binära signaler.

X1 hjälpminne

$$SET = C1F$$
$$RST = C2F$$

Nu är den extra variablen löst och vi kan gå vidare till vårat startminne.
Det är viktigt att man bearbetar signalerna i denna ordning som vi redovisar här. Man kan t ex inte börja med startminnet och sedan hjälpminnet.

Xs Startminne

$$SET = A \bullet C1B \bullet \overline{C1F} \bullet C2B \bullet \overline{C2F} \bullet \overline{X1}$$
$$RST = C1F$$

VC1 Cylinder 1

$$VC = Xs$$

Nu har vi bara VC2 kvar. Eftersom vi var tvungna att föra in en extra variabel, X1 måste den användas nu då vi inte tidigare (i Xs eller VC1) har använt den. Annars skulle den ju inte fylla någon funktion. Så vi kan utan vidare notera X1 i uttrycket för VC2. Det vi får kontrollera är om den ska vara 0 eller 1. Detta ser vi i sanningstabellen. Där kan vi se att den måste vara 1. Sedan får vi som tidigare, kontrollera om vi behöver förkorta eller förlänga signalen.

VC2 Cylinder 2

$$VC2 = X1 \bullet C1B$$