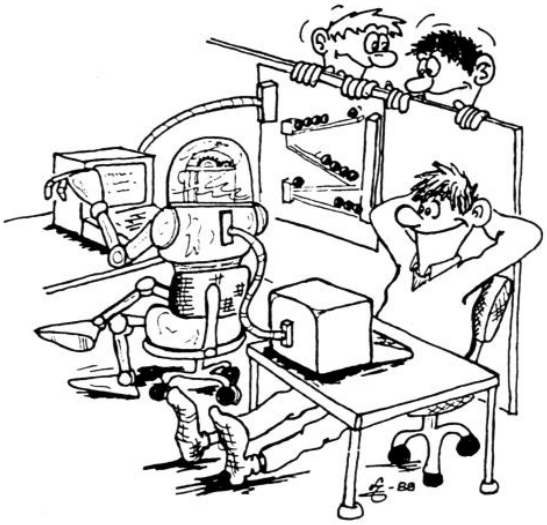
Industriautomation

Tentamen SSY 065, måndag 16/12, 14:00-18:00, V

Lärare: Petter Falkman, (772) 3723

Tider för lärarens närvaro: 15:00, 16:30



Fullständig lösning ska lämnas på samtliga uppgifter. I förekommande fall av tvetydigt formulerade tentamensuppgifter ska den föreslagna lösningen och eventuella antaganden motiveras. Examinator förbehåller sig rätten att godkänna rimligheten i antaganden och motiveringar.

Totalt omfattar tentamen 25 poäng. För betygen tre, fyra och fem krävs 10, 15 resp 20 poäng.

Lösningar anslås första vardagen efter tentatillfället på kursens hemsida i Studieportalen.

Granskning av rättningen får ske 10/1 kl. 12:30 – 13:30 på institutionen.

OBS. Inga hjälpmedel är tillåtna.

Uppgift 1. Robotsimulering

a) Förklara skillnaden mellan "Beredning" och "OLP" simulering. (2p)

Beredning; man väljer lämpliga verktyg och fördelar uppgifter mellan olika robotar, balanserar, ingen programmering av robot, ofta har man bara verktyget synligt.

OLP; off-line programmering, generering av banor och sekvenser för robotrörelser.

b) Vilka hastigheter och laster skall en robot ha vid ISO tester av prestanda? (1p)

Full last och full hastighet

Uppgift 2. Produktionssimulering

a) Kursdelen om simulering av produktionsflöden har belyst ett flertal fördelar med att använda sig av simulering. Det finns dock en del nackdelar/risker som personer med kopplingar till simuleringsprojekt bör känna till. Nämn 4 av dessa nackdelar/risker. (1.5p)

- Simulering tar tid! – Värdera kostnad mot nytta.
- Även om indatan håller dålig kvalitet presenterar simuleringen resultat som uppfattas som väldigt trovärdiga och detaljerade. Detta kan leda till att beslut fattas trots att modellen var en dålig avbild av verkligheten.
- Kräver stora resurser i pengar och personal
- Kan vara svårt att få acceptans för arbetssättet inom den egna organisationen.
- Simulering ger ingen slutgiltig lösning. Fler analyser krävs ofta.

b) Beskriv innebörden av att ***verifiera*** respektive ***validera*** en simuleringsmodell. Beskriv även vilken av de två aktiviteterna som bör utföras först och motivera varför. (1.5p)

Att verifiera en simuleringsmodell innebär att man testar så att modellen uppför sig som man tänkt sig. Att alla logiska kopplingar fungerar. Helt enkelt att man har programmerat den riktigt. Detta gör man genom att jämföra datamodellen med den konceptuella modell (flödesschema) som man skissat fram tidigare i projektet (0,5p).

Att validera en simuleringsmodell innebär att man jämför simuleringsmodellen mot det verkliga produktionssystemet. Alltså ingår då även all indata till modellen utöver att man har kodat riktigt (0,5p).

Verifiering bör utföras före validering för att säkerställa att modellen fungerar korrekt. Det är ingen idé att jämföra en felaktig modell med verkligheten eftersom man då inte kan veta om det är indatan eller modellens uppbyggnad som orsakar en eventuell skillnad (0,5p)

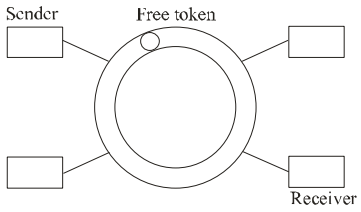
Uppgift 3. Kommunikation

a) Beskriv *Token Passing* metoden för access av det fysiska mediet. (2p)

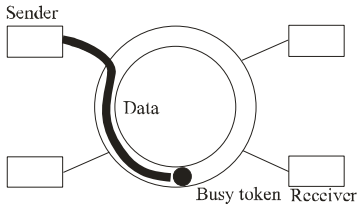
In the *Token Ring* (IEEE 802.5) access method a *token* is sent from node to node, and the node owning the token may transmit data. After it has transmitted the data, the token is passed on. The line is, thus, always free and there are no collisions.

Token Ring has the benefit that it is deterministic in the sense that if we know the number of nodes and we know the time it takes to pass a token, then we can calculate an upper bound on

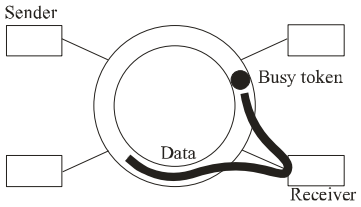
the time it takes for a node to transmit its data. However, as the number of nodes increases, so does this upper bound. Furthermore, a node can actually have to wait for exactly that time before being able to transmit its data.



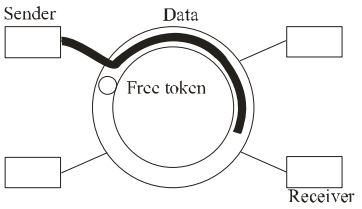
token.



and sends out the data.



received"..



token.

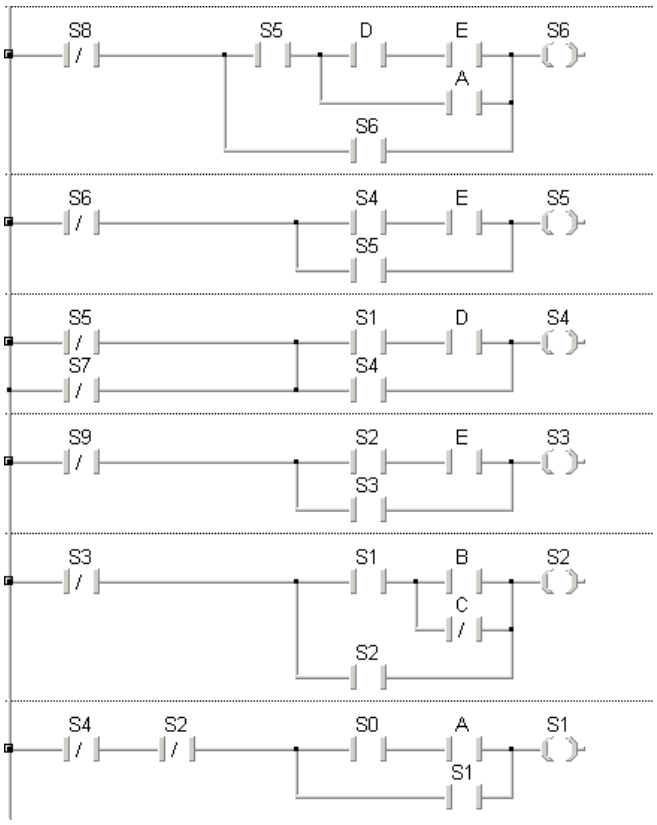
b) Förklara problemet med *Hidden Station* som kan uppstå vid trådlös kommunikation. Vilken typ av medium-access är lämplig för att undvika problemet? (2p)

The CSMA/CD access method does not work very well in wireless networks. This is due to the problem of doing collision detection. An illustrative example of this is the *hidden station problem*. Assume that two sending nodes S1 and S2 are on opposite sides of a third receiving node, R, such that the two senders cannot hear each other across the wireless network. Then, when S1 and S2 sense the channel they will hear an idle channel even as the other is transmitting. This may result in both sending at the same time with the transmissions colliding at R. Because of this, the CSMA/CA, *carrier sense multiple access with collision avoidance*, was developed. Collision avoidance involves talking to R and asking for permission to send. All nodes within range of R then hear R permitting one of the senders to send and become aware that one node is just now actually transmitting to R

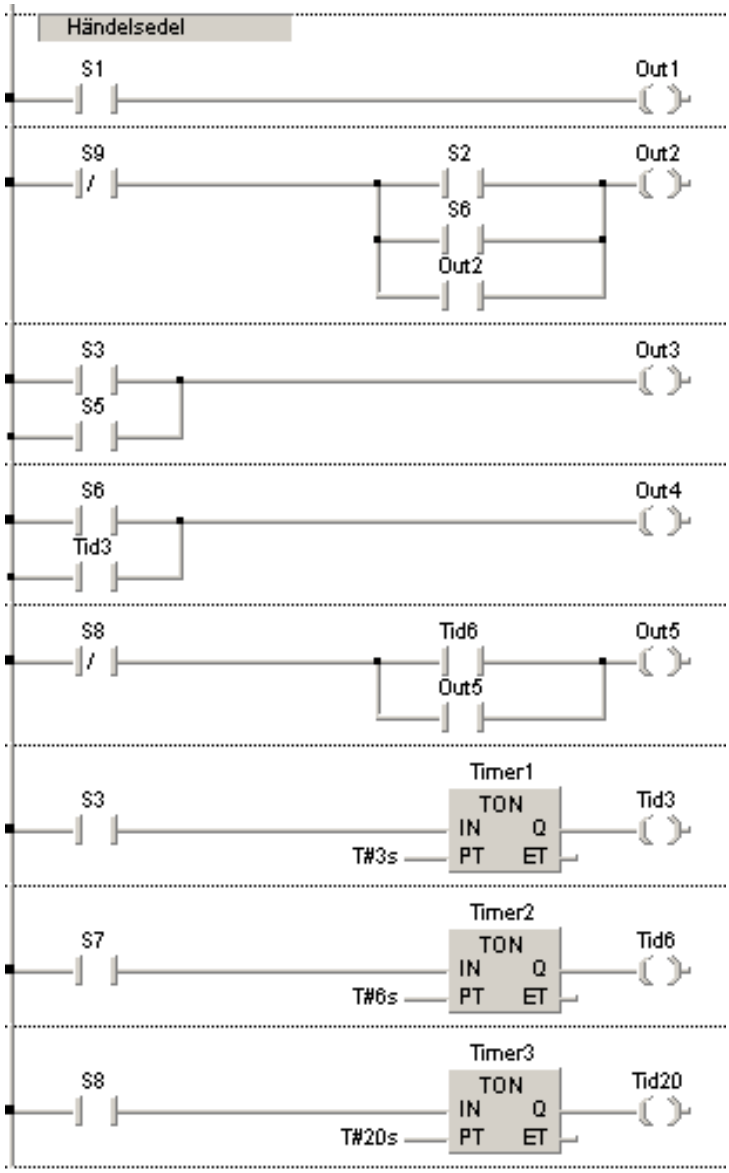
The token ring protocol, on the other hand, is simple to implement in wireless networks. As long as a node can communicate with its nearest neighbors, it can pass on the token, and no "hidden station problem" can occur, as in the case of the Ethernet.

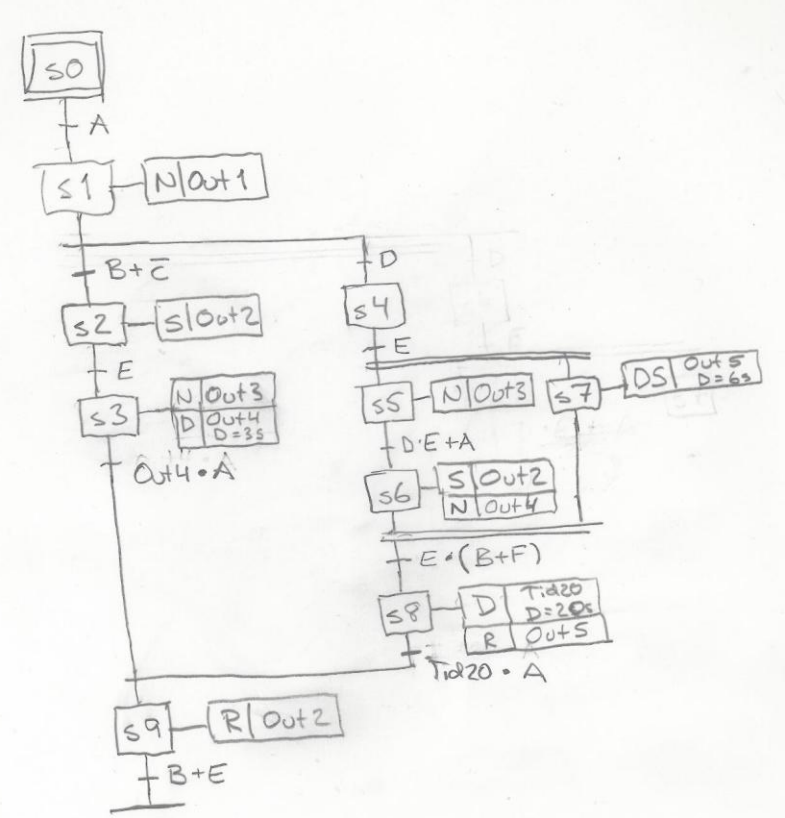
Uppgift 4.

a) Rita funktionsdiagram (SFC) som motsvarar nedanstående Ladder-program. (7p)



Forts nästa sida!


Händelsedel
S1
Out1
S9
/
S2
S6
Out2
Out2
S3
S5
Out3
S6
Tid3
Out4
S8
/
Tid6
Out5
Out5
Timer1
S3
TON
IN
Q
T#3s
PT
ET
Tid3
Timer2
S7
TON
IN
Q
T#6s
PT
ET
Tid6
Timer3
S8
TON
IN
Q
T#20s
PT
ET
Tid20



b) CAN-bussen använder accessmetoden CSMA/CA där CA står för ”Collision avoidance”. Metoden för CA hos CAN-bussen kallas arbitrering bygger på en uppbyggnad så att logisk ”0” är dominant och logisk ”1” är recessiv. Beskriv hur arbitreringen går till med utgångspunkt från en dataram som skall sändas. (2p)

Data Frame forts1. - ID-fältet

ID – identifyer – 11 bitar (standard CAN ver 2.0A) där högsta biten sänds först. (Extended CAN ver 2.0B har 29 bitars identifyer.)

Vi tittar på ett exempel där 4 noder, A-D, börjar sända samtidigt:

Nod som sänder	ID	Första bit	Andra bit	Tredje bit	Fjärde bit	Övriga bitar
A	0010101…	0	0	1	tystnar	
B	0001011…	0	0	0	1	fortsätter ensam
C	0101000…	0	1	tystnar		
D	0110111…	0	1	tystnar		
Signal på buss		0	0	0	1	

Då den egna sända biten inte stämmer med den som avlyssnas på bussen så tystnar noden. Till slut finns endast den nod med högst prioritet kvar och den slutför då sin sändning. Detta förlopp kallas *arbitrering*.

Vid hög belastning på bussen kan meddelanden med låg prioritet få vänta länge. Detta har lösts så att efter viss tid utan att få igenom ett meddelande så höjs automatiskt prioriteten genom att göra vissa bitar i ID-fältet dominanta.

ID-koden (de bitar om inte används för prioritetshöjning) innehåller också identiteten på de data som sänds. Identiteten är förknippad med prioriteten t.ex är bilbältessträckare ”till” viktigare än höger blinkers ”till”.

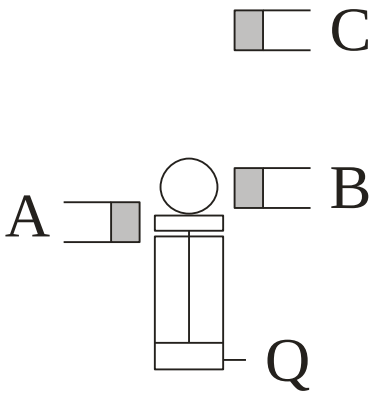
Hos mottagande noder finns sedan ett filter som läser identifieraren och avgör om om hela meddelandet skall läsas och MCU skall meddelas (genereras interupt) eller om meddelandet är ointressant och ignoreras.

59

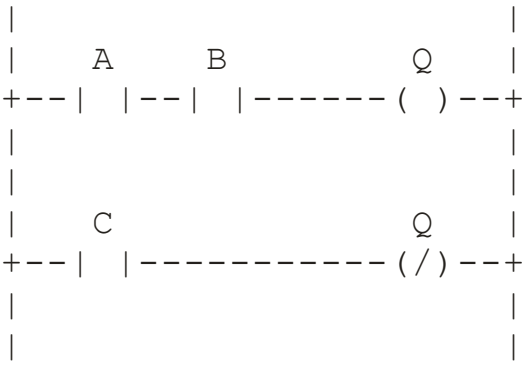
Uppgift 5. PLC programmering

Emil och Emilia skulle programmera en PLC att styra ett enkelt labsystem, se bilden nedan till vänster. De skrev LD-koden till höger nedan och förväntade sig att när kolven var nere

och kulan var på plats så skulle kolven gå upp till sitt övre läge, vid C, och därefter skulle kolven gå ner igen.

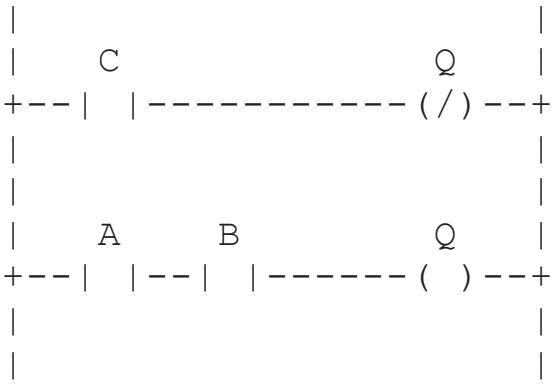


Det enkla labsystemet. A, B och C är givarsignaler, A talar om att kolven är nere, B talar om att kulan är på plats, C talar om att kolven är uppe. Q är utsignal som, då den är hög, kör upp kolven.



Första försöket till LD-program som skulle styra kolven till sitt övre läge vid C när kolven är nere och kula är på plats.

Tyvärr fick Emil och Emilia ett helt annat beteende än de hade tänkt sig. Efter att ha funderat en stund föreslog Emilia nedanstående LD-program, men inte heller det fungerade som tänkt.



Andra försöket till LD-program som skulle styra kolven till sitt övre läge vid C när kolven är nere och kula är på plats.

a) Beskriv det beteende första LD-programmet torde ha uppvisat. (2p)

b) Beskriv det beteende det andra LD-programmet torde ha uppvisat. (2p)

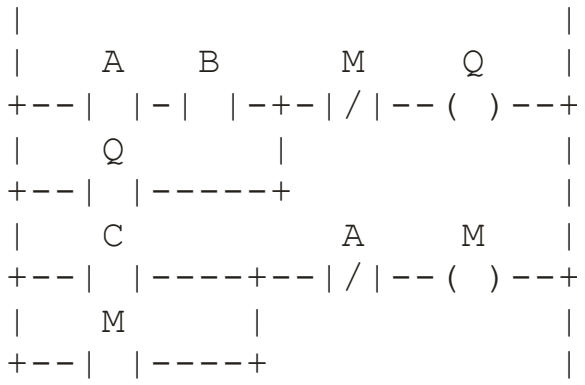
c) Ge ett LD-program, som uppvisar det tänkta beteendet, alltså: upp när kolven är nere och en kula kommer, och ner när kolven väl nått sitt övre läge. Notera att kulan åker med både upp och ner, och det räcker att detta bara sker en enda gång. (2p)

In both cases Q is output from both rungs. The PLC scan-cycle behavior will show the last "calculated" value of Q to the outside world. Thus, the first attempt will start its up-going movement immediately, whether a ball has arrived or not. Since the C signal is false, its negation will be true and this value will be set to Q. When the cylinder arrives at its top-most position, C becomes true and its negation false. Hence, the cylinder will start descending, until C once again becomes false. Then the cylinder will rise again. This oscillation around the C sensor will then continue indefinitely.

In the second attempt, it is instead the conjunction of A and B that governs the value of Q. In this case, when a ball arrives the cylinder will start its ascent, until A or B (or both),

depending on which becomes false first, will result in a false output on Q. Then the cylinder will descend, until QW becomes true again. As before, this oscillation will continue indefinitely, but now at the bottom of the cylinders work stroke.

To function correctly, the program must remember that the system has been all the way up at C, and the program must hold Q true until this condition has been met. Two memories solve this.



In the LD-program above, when both A and B are true, Q becomes true (assuming M is false, a reasonable initial value) and the cylinder starts its ascent. At some point A (or B or both) become false, but the self-hold circuit keeps the cylinder moving upward. Once, C is reached, M is set and breaks the self-hold on Q. This makes the cylinder go down, and once it's down again, A breaks the self-hold on M and it all repeats. To have the up-down-sequence performed only once, remove the break on A of the M-circuit.