Industriautomation

Tentamen SSY 066, torsdag 20/12, 14:00-16:00, M

Lärare: Petter Falkman, (772) 3723

Tider för lärarens närvaro: 14:45

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 15 poäng. För betygen tre, fyra och fem krävs 7, 10 resp 13 poäng.

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

Granskning av rättningen får ske 21 Jan kl. 12:30 – 13:30 och 23 Jan kl. 12.30 -13.30 på institutionen.

OBS. Inga hjälpmedel är tillåtna.

Uppgift 1. Datorkommunikation

Att styra accessen av det delade kommunikatiosnmediet är en viktig sak i samband med datorkommunikation. Olika kommunikationsprotokoll gör detta olika på de lägsta nivåerna i OSI-stacken

a) Beskriv *token passing* metoden, och ge dess för- och nackdelar. (2p)

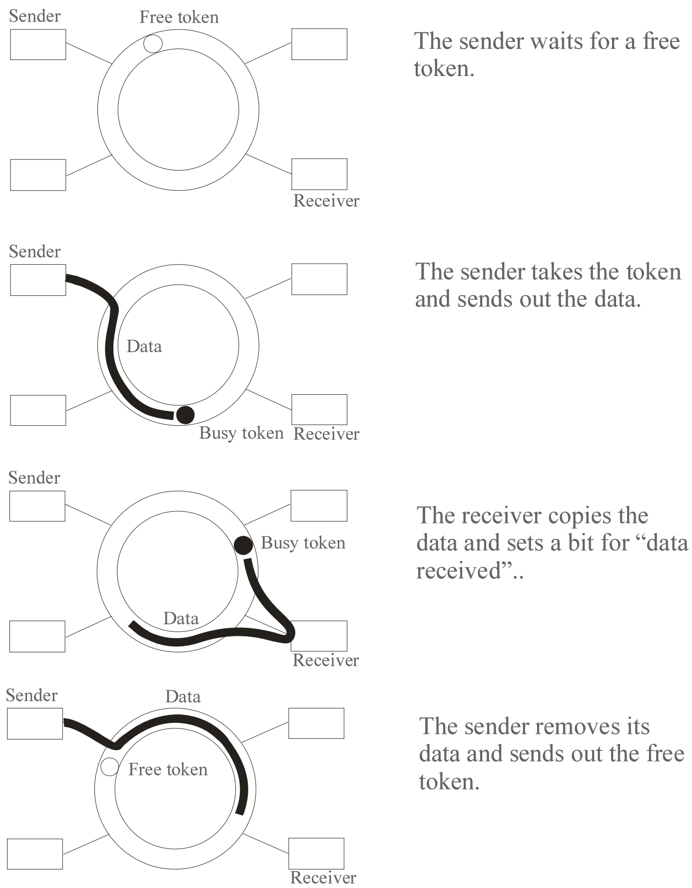
b) Beskriv CSMA/CD (*carrier sense, multiple access, collision detection*) metoden, och ge dess för- och nackdelar. (2p)

c) Beskriv hur detta sköts av CAN-protokollet, och ge för- och nackdelar med den metoden. (2p)

d) Förklara varför CSMA/CD inte fungerar bra i trädlösa sammanhang. Föreslå också en metod som är lämplig för trådlös kommunikation. (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.

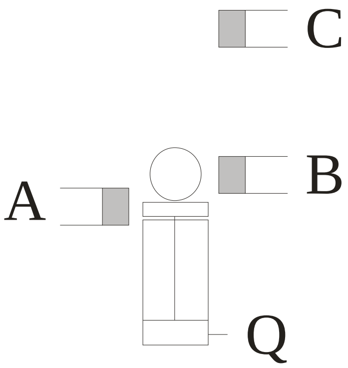


The CSMA/CD access method does not work very well in wireless networks. This is due to the problem of doing collision detection. An illustrative example of this is the *hidden station problem*. Assume that two sending nodes S1 and S2 are on opposite sides of a third receiving node, R, such that the two 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 2. 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 (signaleras av givare A) och en kula var på plats (signaleras av givare B) så skulle kolven gå upp till sitt övre läge (styrs av utgång Q) och därefter skulle kolven gå ner igen. Givaren C känner av kulan och ger alltså signal när kolven är uppe med en kula.




```
|                    |
|   A    B       Q   |
+--| |--| |------( )--+
|                    |
|                    |
|   C            Q   |
+--| |-----------(/)--+
|                    |
|                    |
```

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.

```
|                    |
|   C            Q   |
+--| |-----------(/)--+
|                    |
|                    |
|   A    B       Q   |
+--| |--| |------( )--+
|                    |
|                    |
```

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

b) Beskriv beteendet som 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 med kulan. Notera att kulan åker med både upp och ner, och det räcker att detta sker en enda gång. (3p)

In both cases Q is output from both rungs. The PLC scan-cycle behavior will output 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. Since the cylinder starts its upward movement immediately as the PLC is turned on, it seems reasonable to assume that it will move upwards without a ball, and thus C will never become true. This means that the system will be stuck in the state were the cylinder has reached its top position; C will never become true, hence Q will never become false.

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 Q becomes true again. This oscillation at the bottom of the cylinders work stroke will then continue indefinitely.

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.

```
|                      |
|   A   B     M    Q   |
+--| |-| |-+-|/|--( )--+
|   Q      |           |
+--| |-----+           |
|   C         A    M   |
+--| |----+--|/|--( )--+
|   M     |            |
+--| |----+            |
```

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.
