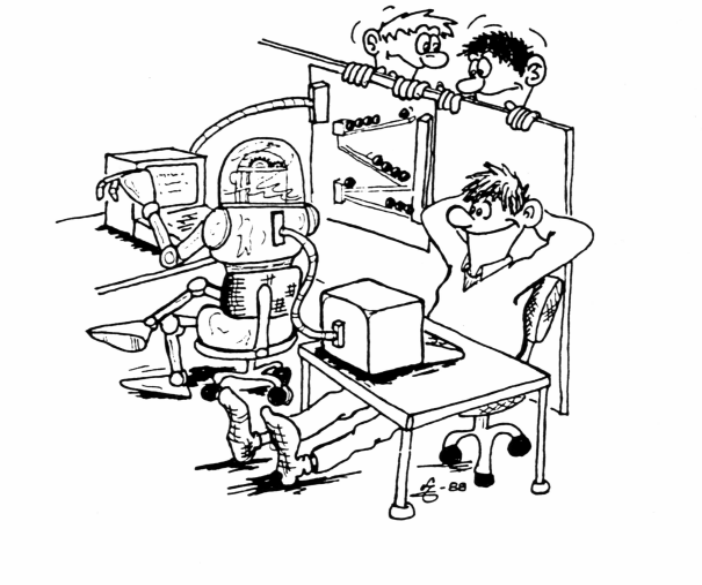
Industriautomation

Tentamen SSY 065, fredag 28/3, 14:00-18:00, V

Lärare: Martin Fabian, (772) 3716

Tider för lärarens närvaro: 15:00, 17:00



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.

Tentamensresultaten anslås senast 18/4 på institutionens anslagstavla, kl. 12:30.

Granskning av rättningen får ske 18/4 kl. 12:30 – 13:30 på institutionen.

OBS. Inga hjälpmedel är tillåtna.

Uppgift 1. Robotsimulering

a) Nämn tre grundläggande krav på ett avancerat ”off-line” system för robotprogrammering. (3p)

Kunskap om applikationen
3-D rums modell
Kinematisk & Dynamisk modell
Grafisk eller Text programmetod
Verifiering/Simulering
Interface till Robot: download/upload

b) Beskriv (rita gärna) axelkonfiguration och arbetsvolym för en:
Cylindrisk robotarm
Articulated/universell robotarm
SCARA (3p)

Rotation kring z, prismatiskt z prismatiskt i xy planet, cylinderskal
rotation kring z, 2 till rotations leder som är parallella, sfärisk volym med litet hål i.
Prismatiskt i z, två rotations axlar båda kring z arbetsvolym som en puck.

c) Vad betyder förkortningen TCP inom robotteknik. (1p)

Tool Centre Point

Uppgift 2. Säkerhet

Uppge minst fyra motiv för att utföra riskanalys. (2p)

- ger ett systematiskt sätt att arbeta, risken att missa något väsentligt minskar
- man frestas inte lika lätt att ta till snabba och ofta dåliga kompromisslösningar
- risker samt åtgärder blir ordentligt dokumenterade (även krav i AFS 1994:48 och AFS 2001:1)
- kan innebära att man gör grundläggande förändringar som gynnar både produktion och säkerhet
- kan innebära att man måste utreda och åtgärda driftstörningar vilket både produktion och säkerhet tjänar på
- sist men inte minst, olyckor förebyggs!

Uppgift 3. Produktionssimulering

a) I kursdelen om simulering av produktionsflöden har vi fokuserat på användningsområdet produktionssystem i tillverkande industri. Nämn 5 andra applikationsområden där denna typ av simulering (diskret händelsestyrd simulering) regelbundet används. (5p)

- Sjukvård
- Trafik (vägkorsningar, kollektivtrafik, sjöfart etc…)
- Militär
- Serviceföretag (affärer, call-centers, etc…)
- Flygplatser

- Logistik (Supply-chain)
- Simulering av miljöpåverkan för produktionssystem

Det finns även fler applikationsområden som någon kan komma på. Gör egen bedömning när ni rättar eller fråga mig i tveksamma fall.

b) Beskriv skillnaden mellan en ***variabel*** och ett ***attribut*** när man programmerar i simuleringsmjukvaran Automod. (2p)

En variabel är global, vilket medför att alla laster i modellen kan ändra och avläsa dess värde.

Ett attribut är kopplat till en viss load (likt en ryggsäck med information som den enskilda lasten bär med sig). Värdet på ett attribut riskerar inte att ändras av andra laster.

Uppgift 4. Kommunikation

a) Förklara relationen mellan de olika protokollen TCP, UDP och IP. (2p)

TCP/IP is (currently) the protocol that makes the Internet tick. Though there exists one protocol acronymed TCP, and another acronymed IP, TCP/IP actually names a protocol stack. The stack includes the *transmission control protocol*, TCP, and the *internet protocol*, IP, as well as a number of other well-known protocols, such as the *hypertext transfer protocol*, HTTP, and the *Ethernet* access protocol. A number of other well-known protocols are typically (see Figure 1) also considered to belong to the TCP/IP protocol suite. These are the *simple mail transfer protocol*, SMTP, the *real-time protocol*, RTP, the *domain name* system, DNS, and the *user datagram protocol*, UDP. These latter all are application and transport layer protocols that run over IP, see Figure 1.

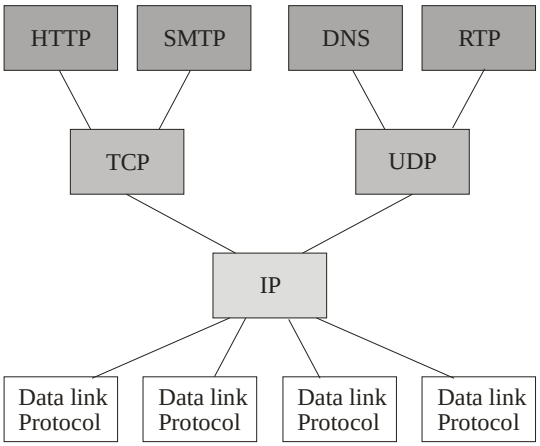


Figure 1. **TCP/IP protocol graph.**

IP is concerned with getting packets from one computer to another, as opposed to TCP and UDP that concern themselves with getting the packets from the sender to the receiver. The packet may have to pass through many computers on the way from the sender to the receiver, and IP figures out which computer to send the packet to next. This is known as a *hop*, and TCP/IP transmission may include many, many hops between the sender and the receiver. Figuring out which hop to make next (to which computer) is known as *routing*, and this is the main concern of IP. Thus, IP is also *connectionless*.

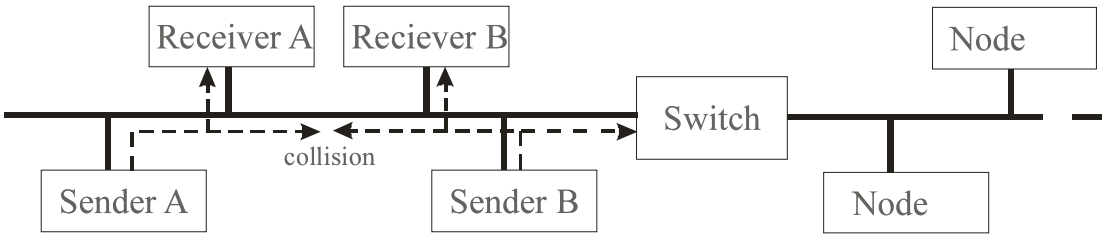
IP is an unreliable protocol. This means that lost or distorted packets are not resent; there are no guarantees that a sent packet actually reaches the intended receiver. Using IP, the TCP protocol implements a *reliable* transmission, guaranteeing that the sent data actually reches the receiver. This TCP does by using acknowledgments and having the possibility to ask for retransmission of lost packets. At the receiver end, TCP puts the packets together in the correct order so that the actual data is recovered.

In contrast to TCP, UDP does not care about reliability. UDP ads just a minimal overhead on the IP packet, and does not sequence the data or retransmit. UDP does also not chop the data up into manageable pieces, like TCP does, but rather expects the application to hand over the data in appropriate chunks. The aim of UDP is to provide sender to receiver streaming of data, such as real-time information (video, music, sensor-signals). In these cases a lost packet or two is not crucial, since a new one will soon arrive and the earlier data will be outdated anyway.

b) Beskriv CSMA/CD metoden för access av det fysiska mediet. (2p)

The *Ethernet* (IEEE 802.3) protocol uses the *carrier sense multiple access with collision detection*, CSMA/CD, access method. When a station wants to transmit it first performs the *carrier* sense part, that is, it listens to the channel. If the cable is busy, the station waits until the channel goes silent; otherwise it transmits immediately. If two or more stations simultaneously (*multiple access*) begin transmitting on an idle channel, they will collide. All colliding stations then terminate their transmission, wait for a random time, and repeat the whole process again.

Methods for collision detection are media dependent, but on an electrical bus such as Ethernet, collisions can be detected by comparing transmitted data with received data. If they differ, another transmitter is overlaying the first transmitter's signal (a collision), and transmission terminates immediately. A jam signal is sent which will cause all transmitters to back off by different random intervals, reducing the probability of a collision when the first retry is attempted.

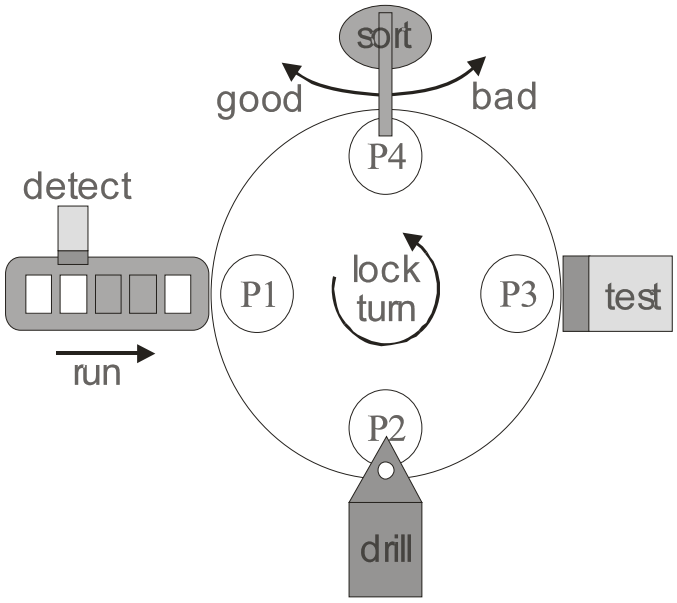


Nodes and collisions on an Ethernet.

CSMA/CD is non-deterministic in the sense that the nodes wait for random amounts of time, so that there is really know way of knowing any upper bound on the communication delay. In practice, though, this approach works but it is sensible to high loads. Though the theoretical problems with communications delay in CSMA/CD has been considered a problem in manufacturing systems, Ethernet is now gaining ground even here. One reason is the introduction of *switched* Ethernets, where special computers called *switches* partitions the system into smaller self-contained sub-systems. Then the communications delays can be kept very small as long as the messages are sent within the specific sub-system.

Uppgift 5. PLC programmering

På ett rundbord sker borrning, testning och sortering av arbetsdetaljer, se nedan. Detaljerna kommer in via transportbandet till vänster vid P1. På transportbandet finns fem platser, och bandet flyttar sig ett ”steg” för varje run-kommando den får. Tillgången på detaljer är högst sporadisk, så i varje ögonblick är noll till fem platser på bandet upptagna. Givaren som ”ser” detaljerna är placerad på transportbandet en bit ifrån bordet, se bilden. Inga givare finns för att kolla om arbetsdetaljer finns i position vid respektive bearbetningsplats. Vid P2 sker borrning, som inte får ske utan detalj på plats. Vid P3 sker avsyning och denna får heller inte utföras utan detalj på plats. Resultatet av avsyningen är ”good” eller ”bad” och beroende på resultatet ska detaljen flyttas till vänster resp. höger vid P4. Inte heller här får någon operation ske utan att det verkligen finns en detalj att operera på.



Notera att detaljer kommer in i systemet sporadiskt, och inga givare finns vid respektive station som håller ordning på om någon detalj verkligen är på plats. Borrning, testning och sortering får inte ske utan detalj på plats. Alla operationerna måste ske med låst bord, men naturligtvis får inte bordet vridas om det är låst.

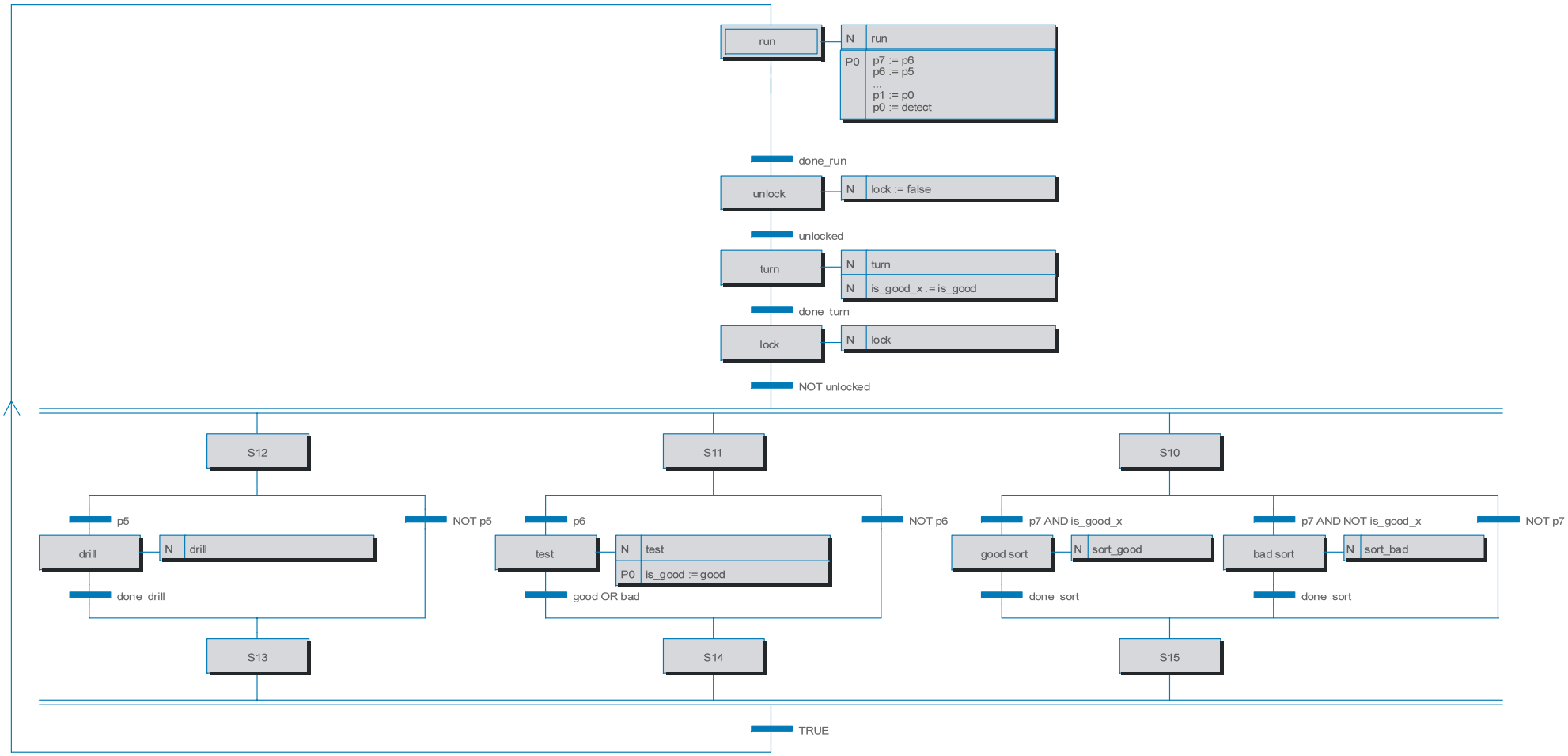
Inputs (high state)		Outputs (high state)	
detect	Work piece detected	run	Run conveyor one step
done run	Conveyor moved one step	lock	Lock table
unlocked	Table unlocked	turn	Turn table one quarter turn
done turn	Quarter turn finished	drill	Operate drill
done drill	Drilling finished	test	Operate test device
good	Positive result from test device	sort good	Sort out good piece to the left
bad	Negative result from test device	sort bad	Sort out bad piece to the right
done sort	Sort operation finished		

Ge ett eller flera funktionsdiagram (SFC, enligt IEC 61131-3) som tillsammans styr stationen. Systemet ska fungera så effektivt som möjligt, vilket innebär att operationerna i största möjliga mån ska utföras parallellt. (5p)

As always, the spec is incomplete and open for interpretation. For instance, there was no mention of the initial state. We have to assume something, and the "most reasonable" assumption seems to be that the system starts up "empty", no pieces on the table, and no pieces on the conveyor.

Furthermore, there seems to be two ways to run the conveyor. Either we run the conveyor so that we always end up with a piece in P1, and only then do we turn. Or, we run the conveyor only one step before turning the table. The first one seems to save on some work, while introducing a necessary extra loop. The other one needs us to remember more positions. Either way is fine.

We need some way to keep track of whether there is a piece at a certain location or not. This is the key issue. There are eight possible locations, including the one in front of the camera. We can simply use eight variables for this, calling them, say, p0, p1, etc, with p0 as the leftmost position (in front of the camera). After a conveyor step we update the variables; p7 is set to the value of p6, etc, and p0 is set to the value of detect. This is a typical implementation of a *shift register*.


run
N
run
P0
p7 := p6
p6 := p5
...
p1 := p0
p0 := detect
done_run
unlock
N
lock := false
unlocked
turn
N
turn
N
is_good_x := is_good
done_turn
lock
N
lock
NOT unlocked
S12
S11
S10
p5
NOT p5
drill
N
drill
done_drill
p6
NOT p6
test
N
test
P0
is_good := good
good OR bad
p7 AND is_good_x
p7 AND NOT is_good_x
NOT p7
good sort
N
sort_good
bad sort
N
sort_bad
done_sort
done_sort
S13
S14
S15
TRUE