

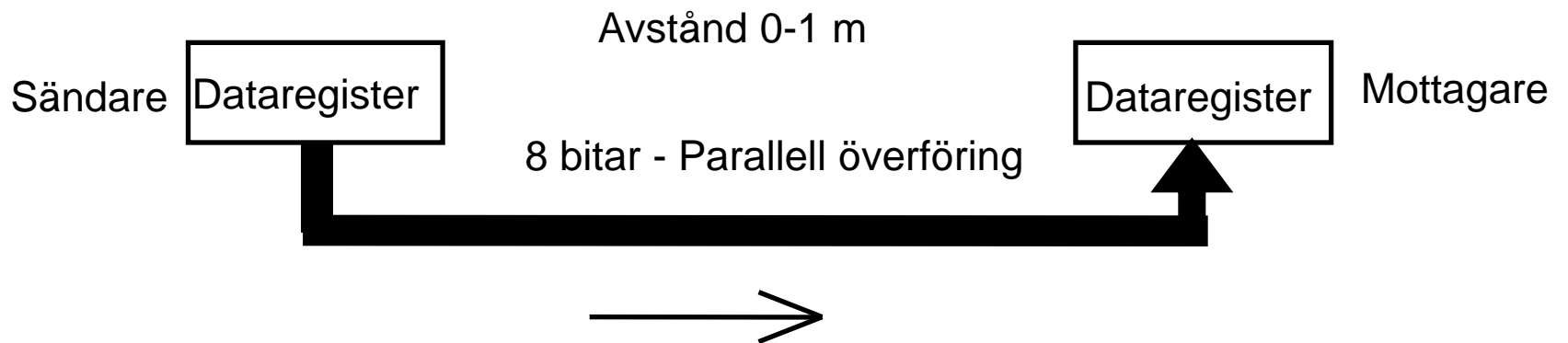
Dataöverföring på Parallell- och serieform

Parallell dataöverföring

- I digitala system finns ofta behovet att flytta ett eller flera dataord från ett ställe (sändare) till ett annat ställe (mottagare).
- Vi har redan stött på detta i samband med FLEX-processorn där man i en operation flyttar 8-bitars dataord mellan två register eller mellan ett register och arbetsminnet.
- I FLEX-processorn använde vi databussen för överföringen och en klocksignal som verkställde laddningen av dataordet hos mottagaren.

Parallell dataöverföring (forts.)

- Att överföra ett helt dataord i en operation på detta sätt kallas parallell (samtidig) överföring.
- Parallell överföring används av praktiska och ekonomiska skäl där avståndet mellan sändare och mottagare är litet .

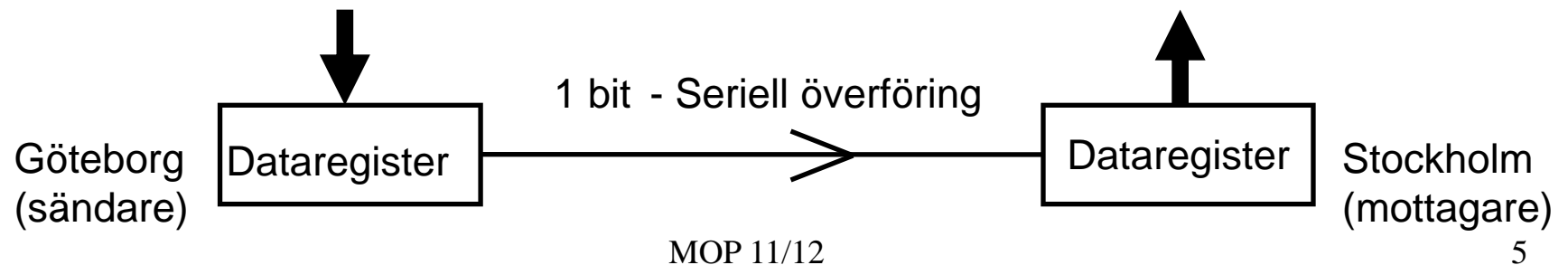


Parallell dataöverföring (forts.)

- Ett praktiskt problem med parallell överföring är att de olika bitarna i ett dataord anländer till mottagaren vid något olika tidpunkter på grund av olika fördröjning i bitledningarna.
- Detta fenomen blir mera märkbart vid ökat avstånd mellan sändare och mottagare eller vid ökad datahastighet (dataord/sekund) och kallas **skevning (skewing)**.
- Ett ekonomiskt problem med parallell överföring är att varje bitledning i bussen inklusive sändar- och mottagarkretsar kostar pengar. Bredare buss ger alltså högre kostnad.

Seriell dataöverföring

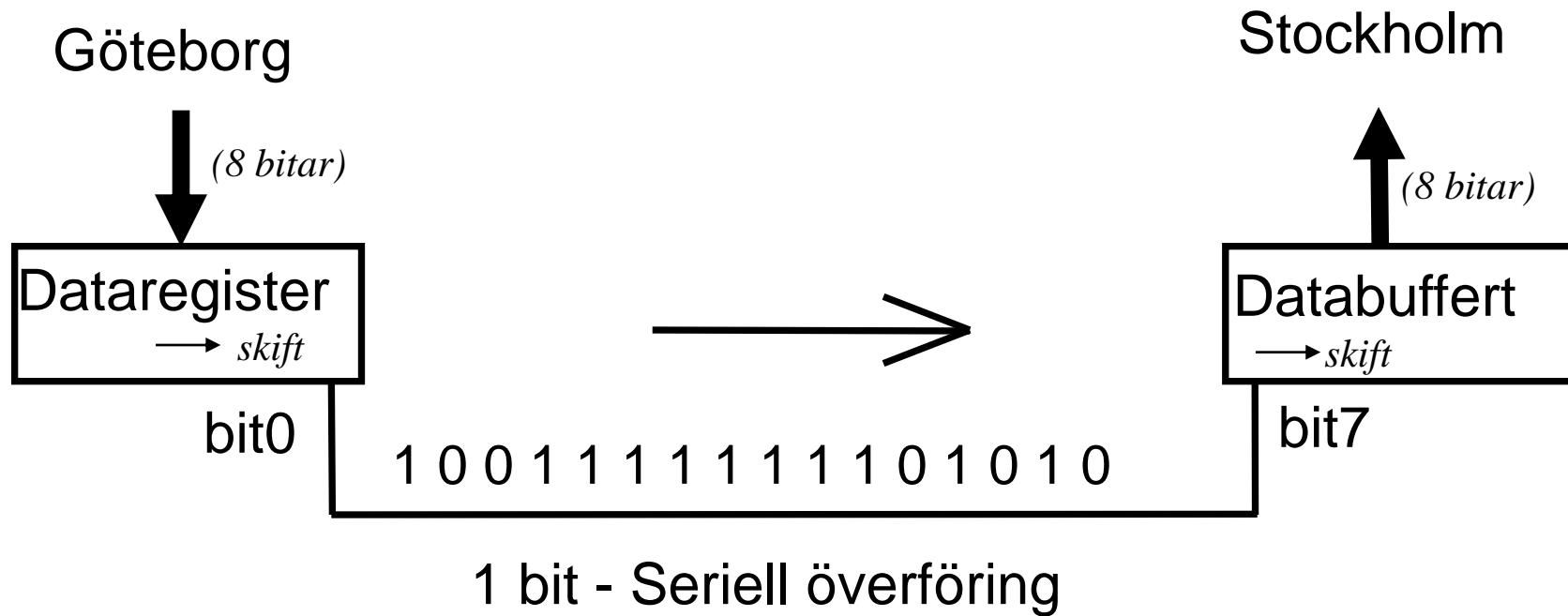
- Istället för att överföra ett helt dataord åt gången kan man skicka dataordets bitar en och en, efter varandra. Detta kallas **seriell överföring**.
- Seriell överföring används normalt då sändare och mottagare befinner sig på medellångt (1 m) till långt (km) avstånd från varandra.
- Av lätt insedda skäl blir detta långsammare men samtidigt billigare.



Seriell dataöverföring (forts.)

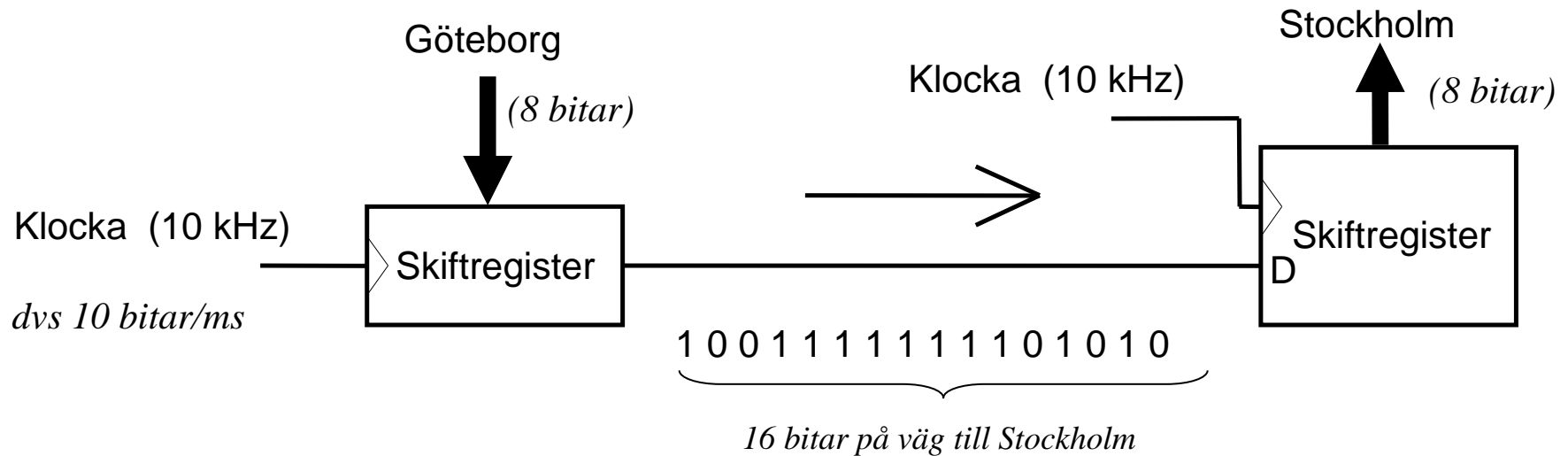
Hur lång tid tar det minst för en bit att "resa" från Göteborg till Stockholm?

$$t = \frac{s}{c} = \frac{480 \cdot 10^3}{3 \cdot 10^8} = 1,6 \text{ ms}$$



Seriell dataöverföring (forts.)

*Hårdvarulösning för
sändare och mottagare:*



*Sändaren i Göteborg hinner
skicka 16 bitar innan den först
sända biten når Stockholm.*

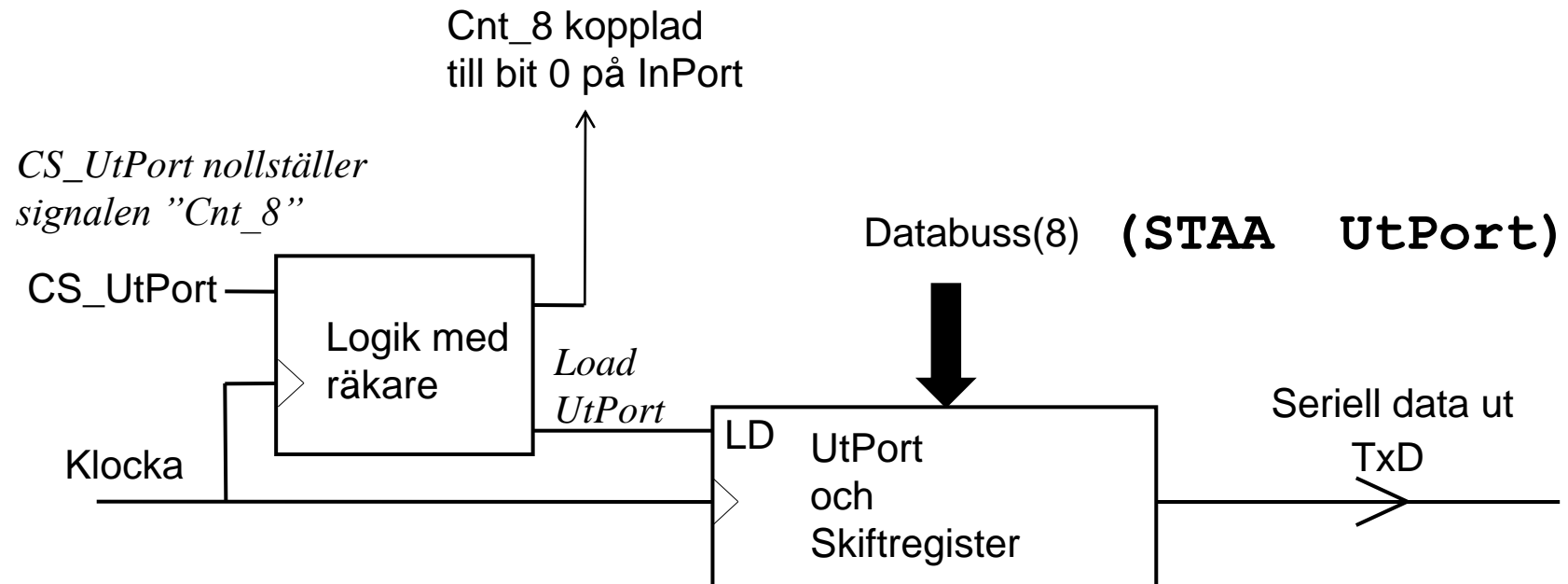
Seriell dataöverföring (forts.)

Alternativ mjukvarulösning för sändaren:

Start	<code>ldaa InPort</code>	<code>Läs Indata</code>
	<code>ldab #8</code>	<code>Antal skift</code>
Next	<code>staa UtPort</code>	<code>Sänd bit0</code>
	<code>lsra</code>	<code>Skifta fram nästa</code>
	<code>jsr Delay01ms</code>	<code>Vänta 0,1 ms</code>
	<code>dec b</code>	<code>Sista biten?</code>
	<code>bne Next</code>	<code>..hopp om Nej</code>
	<code>bra Start</code>	
Delay01ms	<code>nop</code>	<code>Dummyrutin för Delay</code>
	<code>rts</code>	

Seriell dataöverföring (forts.)

Enkel hårdvarulösning för sändare:



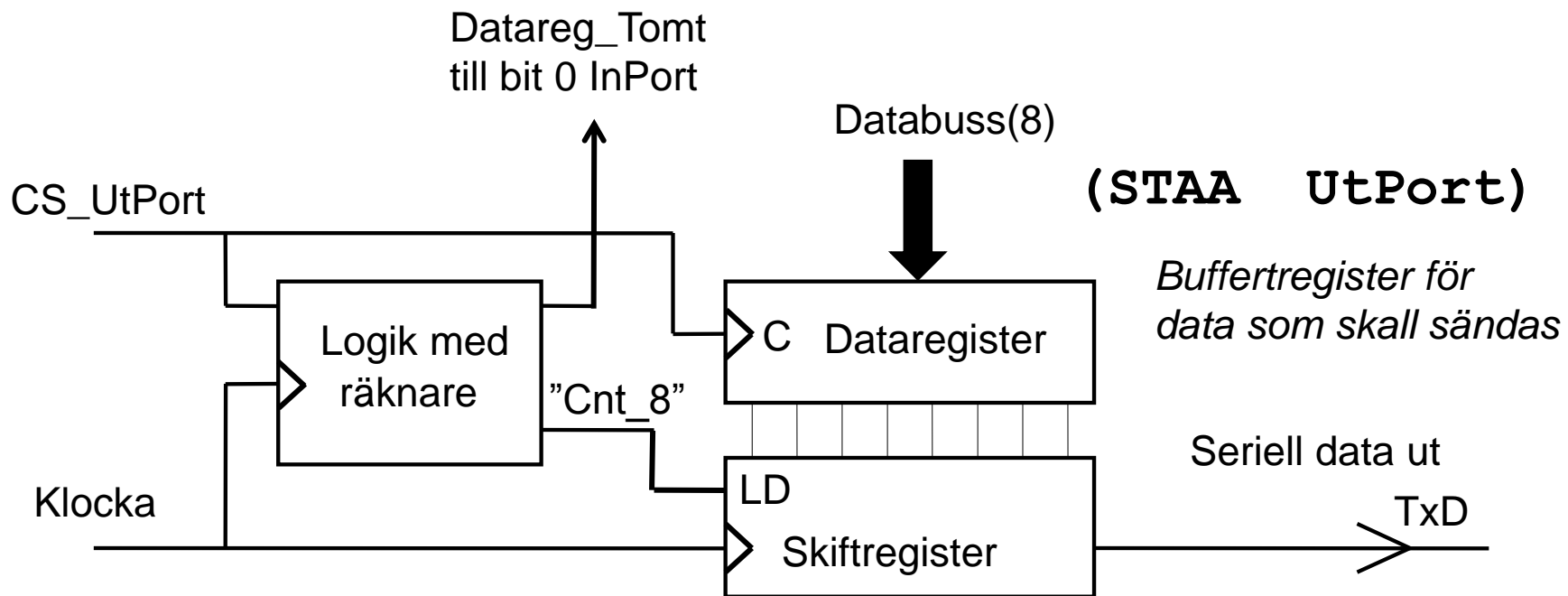
Seriell dataöverföring (forts.)

Program för sändning med enkel hårdvarulösning:

Mask	<code>equ</code>	<code>%00000001</code>	Maskbit för skiftat ut 8 bitar
	<code>ldx</code>	<code>#Data</code>	Pekare till dataarea
NextCh	<code>ldaa</code>	<code>1,x+</code>	Läs dataord
	<code>cmpa</code>	<code>#D_end</code>	Sista dataordet?
	<code>beq</code>	<code>Slut</code>	..hoppa om JA
Wait	<code>ldab</code>	<code>InPort</code>	Läs Status
	<code>bitb</code>	<code>#Mask</code>	Skiftregister tomt?
	<code>beq</code>	<code>Wait</code>	Nej, vänta
	<code>staa</code>	<code>UtPort</code>	Ja, nytt dataord till Skiftreg
	<code>bra</code>	<code>NextCh</code>	Fortsätt med nästa
Slut	<code>nop</code>		
	<code>bra</code>	<code>Slut</code>	

Seriell dataöverföring (forts.)

Förbättrad hårdvarulösning med databuffert för sändare:



Seriell dataöverföring (forts.)

Program för sändning med databuffert:

```
Empty    equ        %00000001    Maskbit för Datareg tomt
          ldx        #Data        Pekare till dataarea
NextCh   ldaa       1,x+         Läs dataord
          cmpa       #D_end       Sista dataordet?
          beq        Slut         ..hoppa om JA

Wait     ldab       InPort       Läs Status
          bitb       #Empty       Dataregister tomt?
          beq        Wait        Nej, vänta

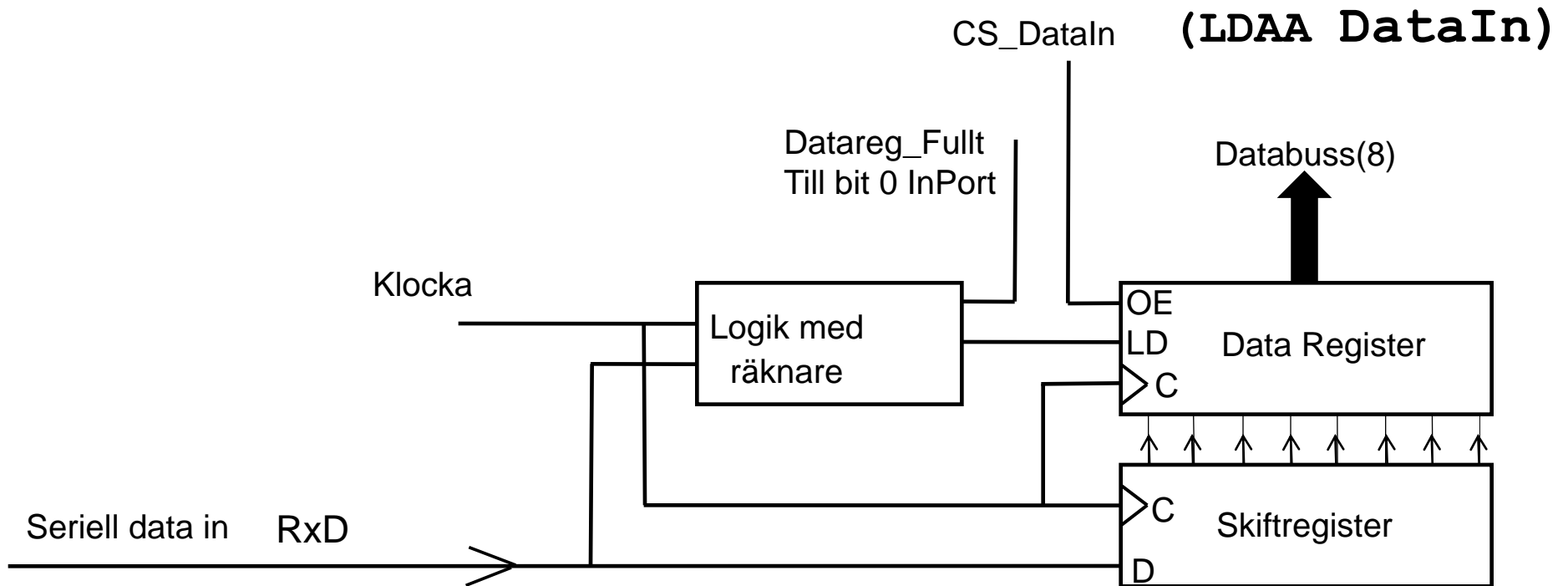
          staa       UtPort       Ja, nytt datord till Datareg

          bra        NextCh       Fortsätt med nästa

Slut     nop
          bra        Slut
```

Seriell dataöverföring (forts.)

Buffrad mottagning:



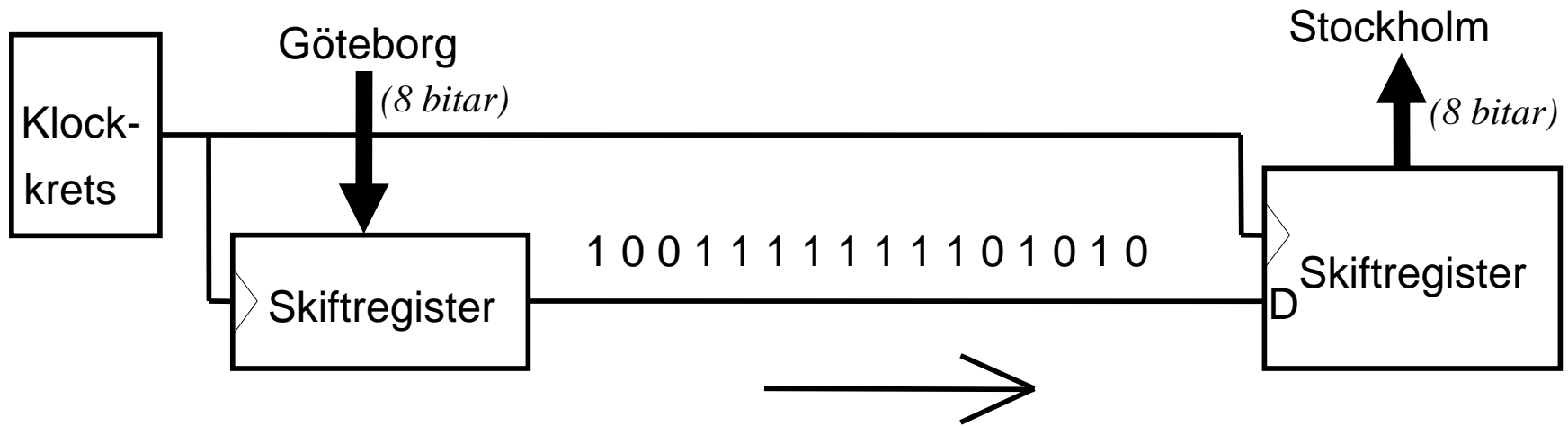
Seriell dataöverföring (forts.)

Program för mottagning med databuffert:

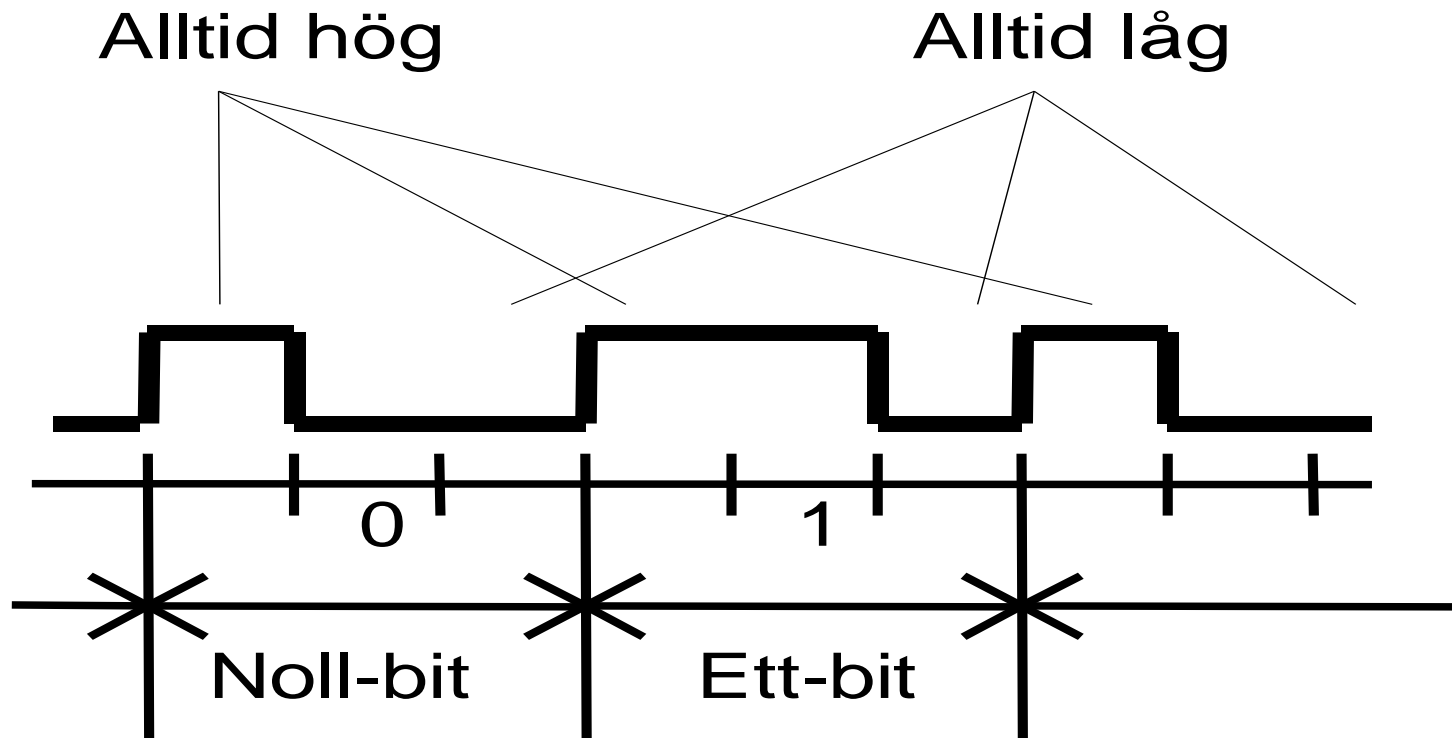
Full	equ	%00000001	Maskbit för Datareg fullt
	ldx	#Data	Pekare till dataarea
Wait	ldab	InPort	Läs Status
	bitb	#Full	Dataregister fullt?
	beq	Wait	Nej, vänta
	ldaa	DataIn	Ja, nytt datord mottaget
	staa	1,x+	Spara dataord i dataarea
	cmpa	#D_end	Sista dataordet?
	bne	Wait	Nej, fortsätt med nästa
Slut	nop		
	bra	Slut	

Seriell dataöverföring (forts.)

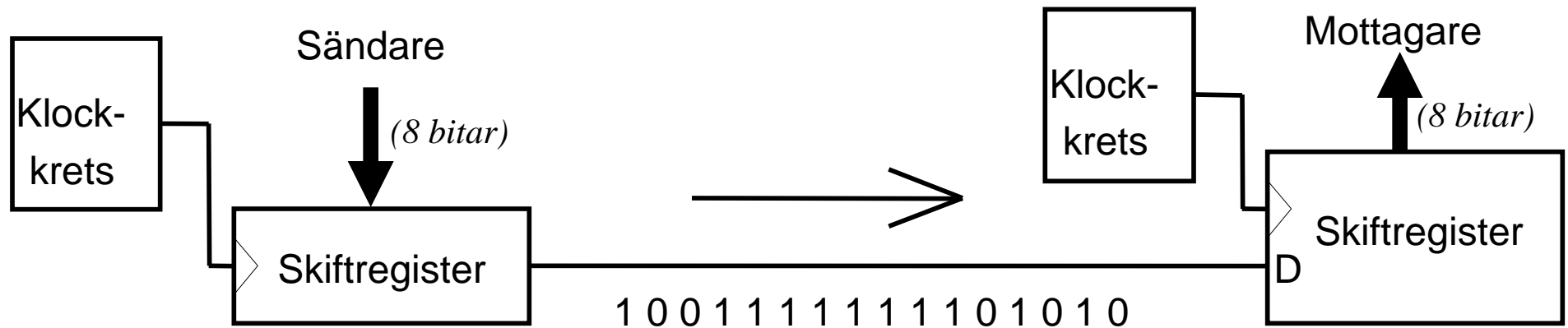
Fungerar förslaget nedan?



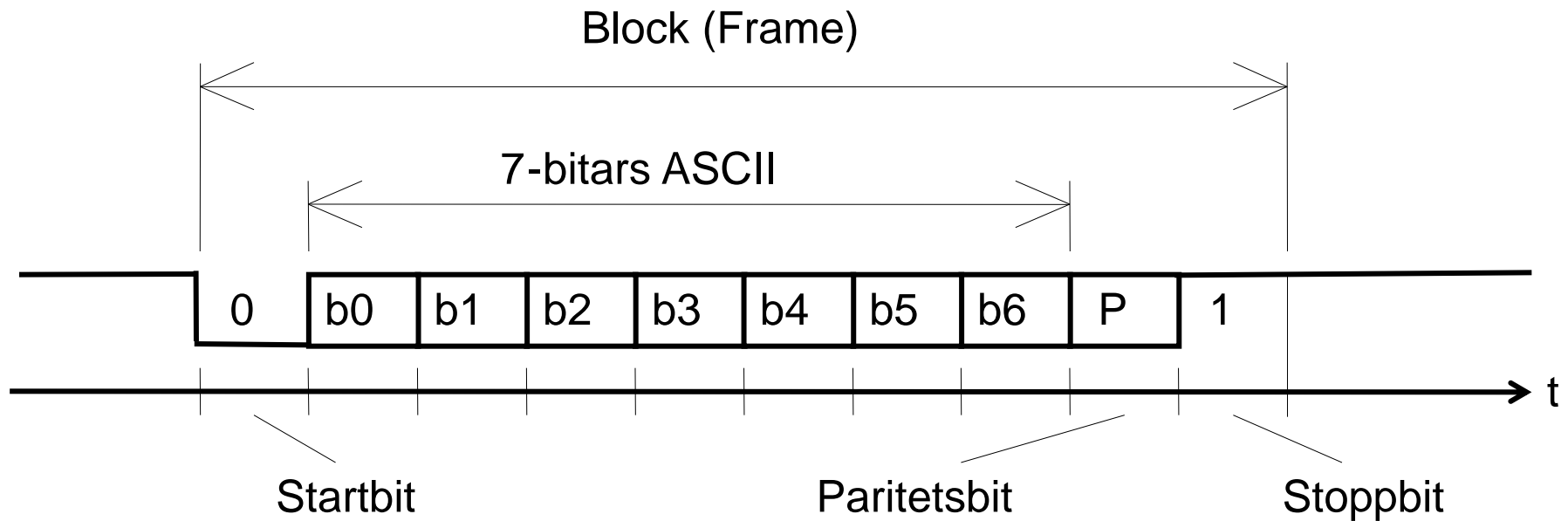
Synkron seriell dataöverföring



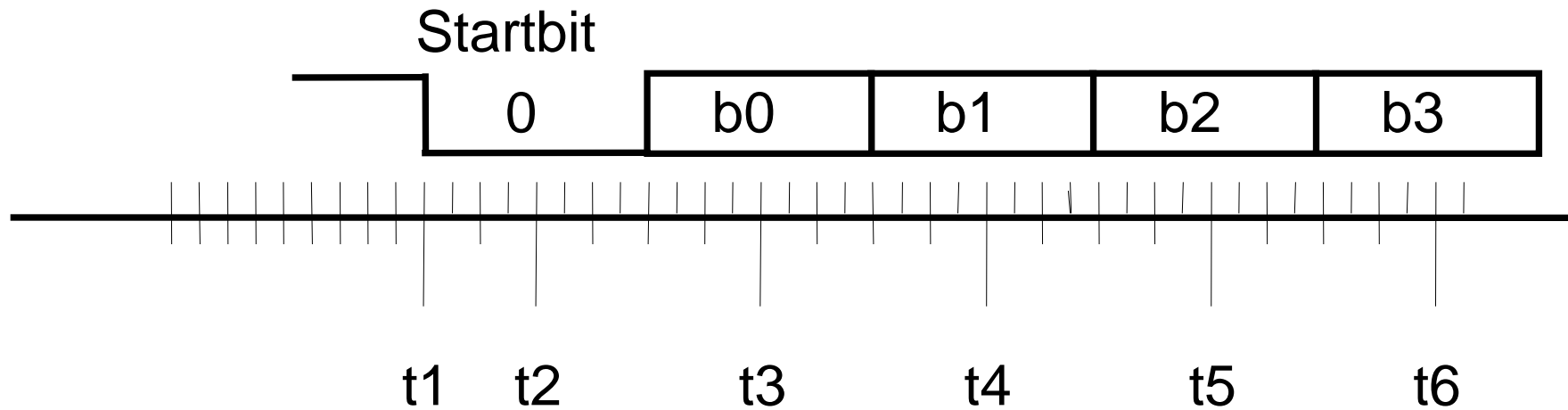
Asynkron seriell dataöverföring



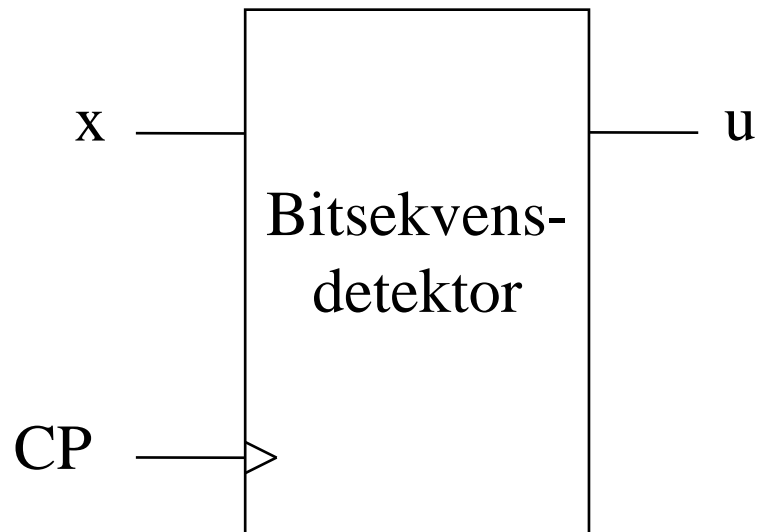
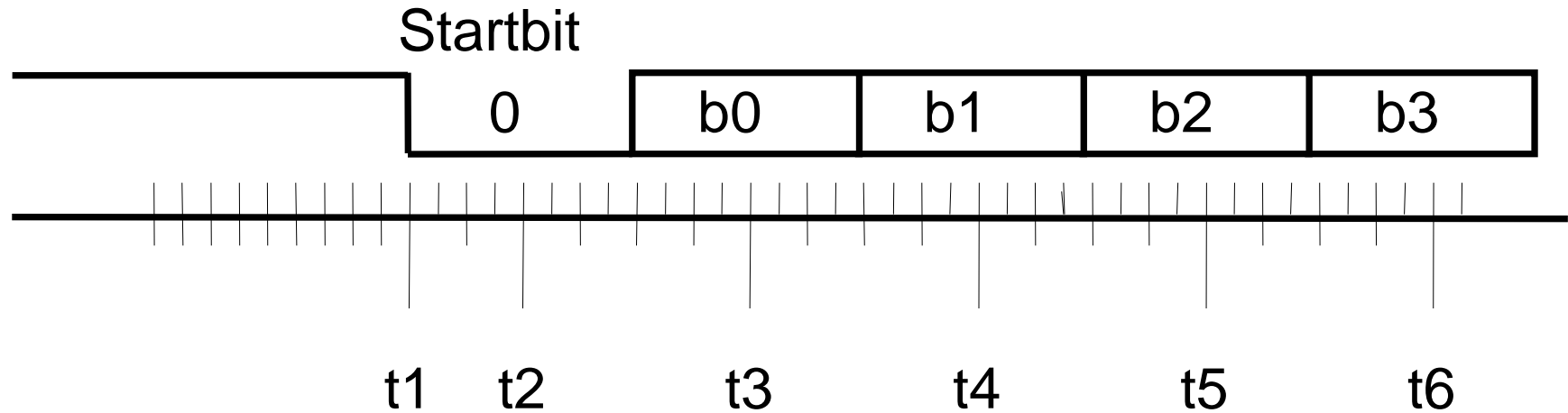
Asynkron seriell dataöverföring



Seriekommunikation – Asynkron överföring



Seriekommunikation – Asynkron överföring



Seriekommunikation

Hur många databitar kan "vara i luften" samtidigt mellan Göteborg och Stockholm?

Vi förutsätter en klockfrekvens på 1 MHz och 1 databit per klockperiod.

$$\text{Antal bitar} = \frac{s}{c \cdot \frac{1}{f}} = 1600 \text{ bitar}$$

Seriekommunikation- Överföringskapacitet

Överför 100 Gbyte från Göteborg till Stockholm på serieform med bithastigheten 10^6 bitar per sekund:

$$t = \frac{\text{ant. bitar}}{f} = \frac{100 \cdot 10^9 \cdot 8}{1 \cdot 10^6} = 8 \cdot 10^5 \text{ s} \approx 9,3 \text{ dygn}$$

Seriekommunikation- Datanät- Media

	<i>Tvinnad partråd</i>	<i>Koaxialkabel</i>	<i>Fiberoptik</i>
<i>Kapacitet</i>	<i>Låg</i>	<i>Medium</i>	<i>Mycket hög</i>
<i>Flexibilitet</i>	<i>Mycket hög</i>	<i>Medium</i>	<i>Låg</i>
<i>Kontaktering</i>	<i>Billig</i>	<i>Medium</i>	<i>Mycket dyr</i>
<i>Störökänslighet</i>	<i>Låg</i>	<i>Bra</i>	<i>Mycket hög</i>

Seriekommunikation- typiskt datablock



└─ Startfält