

Datorteknik DALIA Lab 1

IR-länk — v0.7

Inledning



I alla inbyggda system är kommunikation med omvärlden viktig. Det kan handla om något så enkelt som att läsa av en yttre digital signal eller att skicka ut en signal för att tända en lysdiod.

Ofta vill man också överföra hela bytes eller ASCII-tecken. Nuförtiden görs det ofta seriellt men viss hårdvara kan fortfarande kräva parallell överföring.

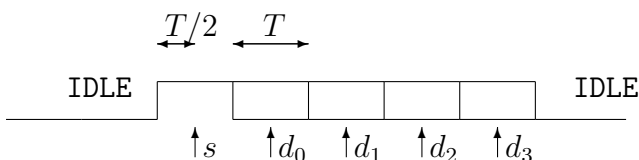
I den här laborationen kommer du behöva skriva ett program som hanterar alla metoderna ovan. Det handlar om att tillverka en avkodare för en IR-signal som kommer från ett tangentbord.

Din del är att med hjälp av processorn läsa av den mottagna signalen och omvandla den till det ursprungliga värdet för visning på en sju-segments display med inbyggd BCD till sju-segmentsavkodare.

Till din ledning finns insignalens datastruktur i JSP samt motsvarande strukturdiagram för kod.

Den seriella signalen

Den tidsseriella signal som IR-mottagaren lämnar är uppbyggd enligt:



När ingen insignal är för handen är kanalen i vila (eng. *idle*) och mottagaren ger en låg utsignal.

Den seriella signalen består därefter av en startbit, s , av längden T , som alltid är logiskt hög och fyra databitar, d_0, d_1, d_2, d_3 , även dessa av längden T , vars värden beror på det överförda datat. Före och efter transaktionen antar kanalen tillståndet IDLE för en tid längre än T .

Säkrast avläsning av signalens värde görs i mitten av varje bit indikerat av pilarna.

Laborationsuppgift

Skriv ett program som avkodar den mottagna signalen.

Programmet måste vara förberett, simulerat och "färdigt" innan laborationen.

Du **måste** använda fördröjningsrutinen `DELAY` för T och $T/2$. Rutinen skickar ut en skvallersignal till oscilloskopet som är hög när rutinen körs. `DELAY` möjliggör tidsanpassning av programmet till insignalen.

Programmet skall konstrueras i samklang med inkommande data.

I tillståndet IDLE skall en stigande flank letas upp, därefter kontrolleras att den fortfarande är logiskt hög vid tiden $T/2$, i vilket fall de resterande bitarna kan läsas av med en fördröjning T , annars återgår programmet till IDLE.

Använd strukturdiagrammet över signalen på nästa sida och kodstrukturdiagrammet på sista sidan som ledning.

Hårdvara

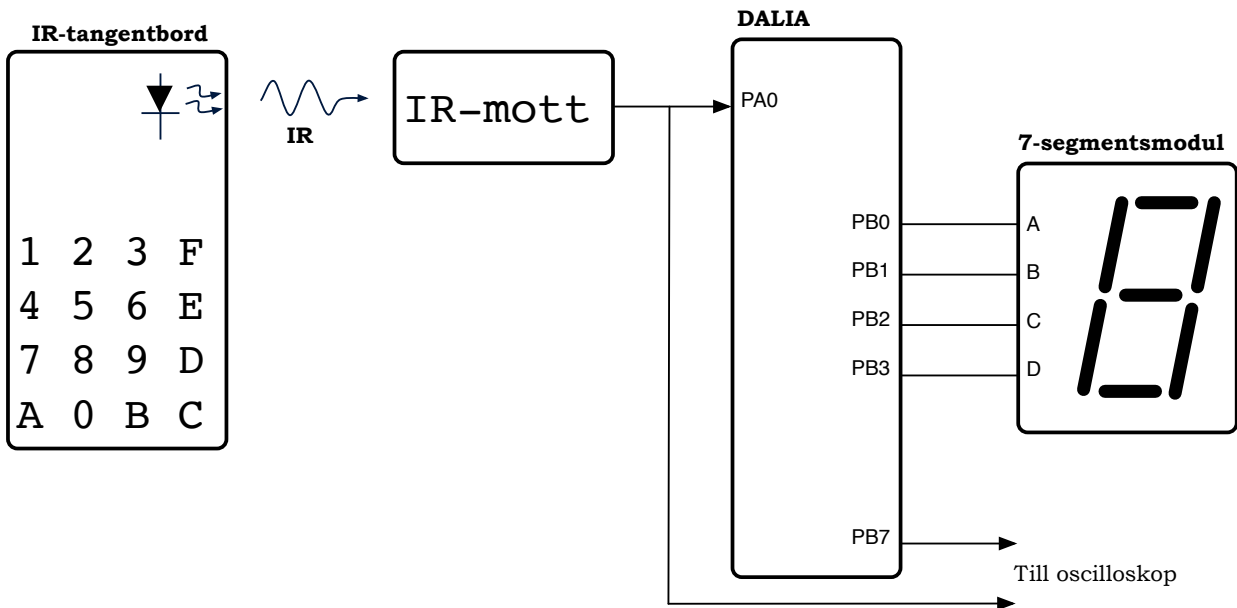
Innan laborationen måste du läsa på om:

- Tangentbordet med IR-utgång
- IR-mottagaren
- Oscilloskopet
- Hur den seriella signalen är uppbyggd

De olika modulerna ansluts på laborationen enligt figuren nedan. Sändarsidan är redan uppkopplad och består av ett hexadecimalt tangentbord som

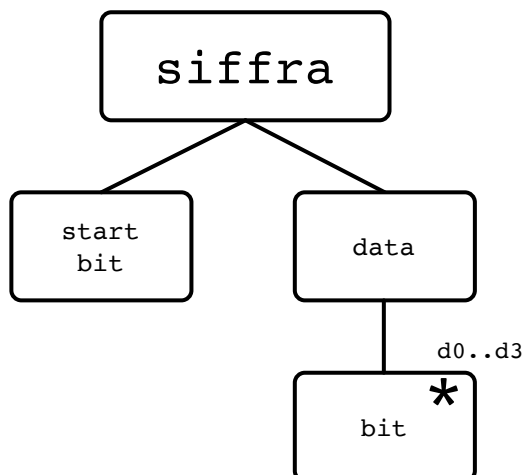
skickar en fyra-bitars seriell signal till en infraröd sändare.

Insignalen från IR-mottagaren påförs PA0 samt oscilloskopet, så du vet vad programmet har för indata. Utsignalerna PB3–PB0 är de fyra parallella databitarna från programmet och PB7 är en signal till oscilloskopet för att ge en uppfattning om vad programmet håller på med. Glöm inte att konfigurera PB7 som utsignal.



Strukturdiagram för indata

Strukturdiagrammet nedan beskriver insignalen enligt JSP. Diagrammet skall användas som grund för programmets struktur. På nästa sida återfinns ett förslag på utvecklad programstruktur som du kan använda för din assemblerkod.

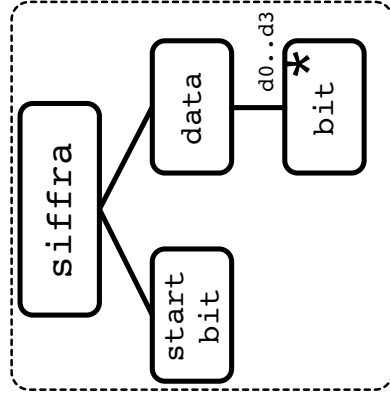


Fördröjningsrutinen DELAY

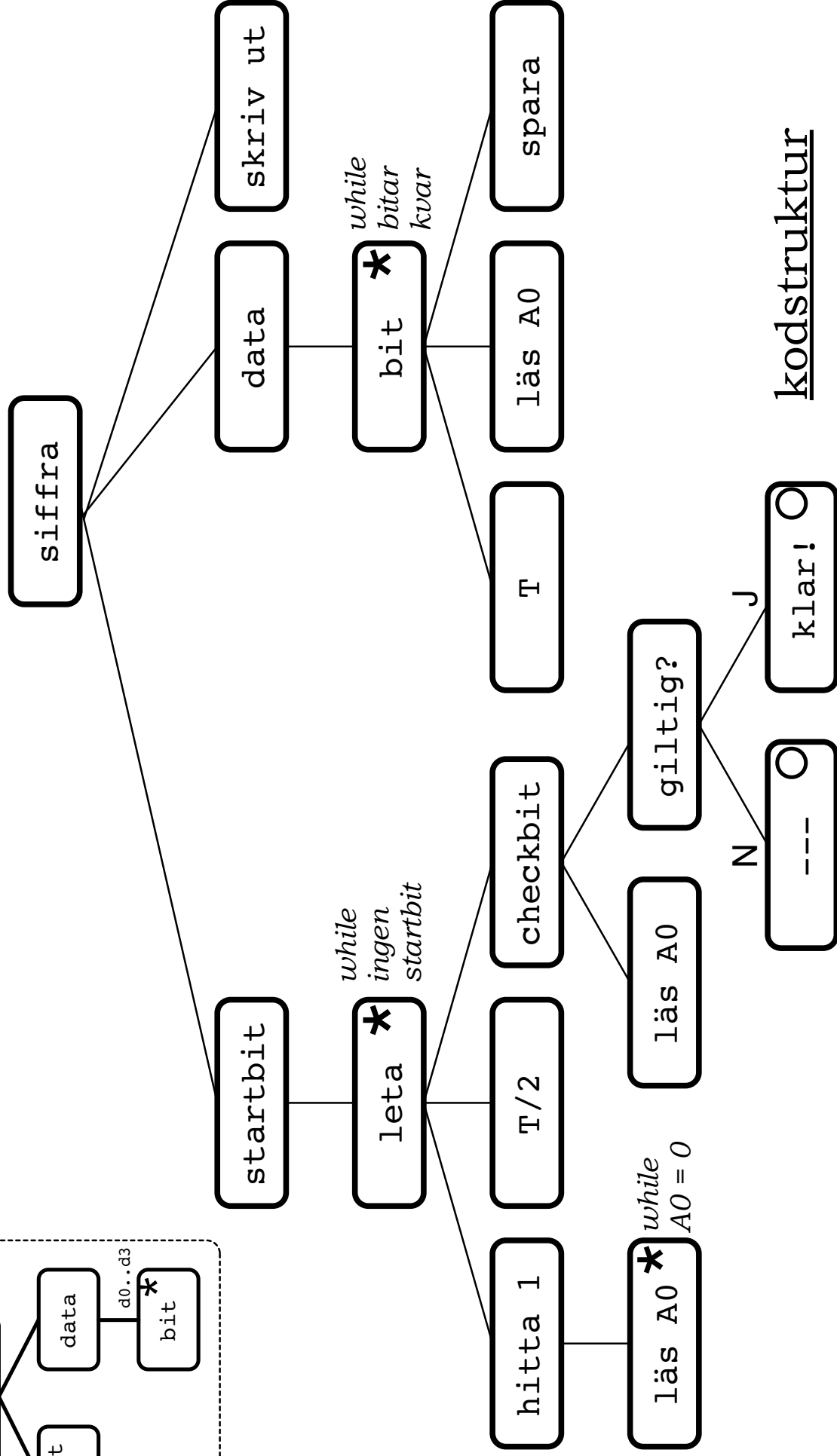
```
; Rutinen DELAY är en vänteloop som sam-
; tidigt avger en skvallersignal på PB7.
; PB7 är hög när rutinen körs.
;
; Med angivet värde i r16 väntar rutinen
; ungefär en millisekund @ 1 MHz.
;
; PORTB måste konfigureras separat.
```

DELAY:

```
    sbi        PORTB,7
    ldi        r16,10    ; Decimal bas
delayYttreLoop:
    ldi        r17,$1F
delayInreLoop:
    dec        r17
    brne       delayInreLoop
    dec        r16
    brne       delayYttreLoop
    cbi        PORTB,7
    ret
```



datastruktur



kodstruktur

Bedömning för examination

En godkänd lösning måste uppfylla kriterierna nedan.

1. Generellt gäller att uppgiften skall lösas. Den presenterade koden skall dessutom vara läslig och utformad enligt vår kodstil, dvs

- att koden är indenterad.
- att subrutiner
 - har bra namn
 - har nödvändiga lokala variabler,
 - avslutas på **ett** ställe, längst ner i rutinen.

2. I denna laboration speciellt:

- Då datastrukturen är en iteration för varje bit måste denna del av programmet loopas. En rent sekvensiell lösning godtas inte.
- Lösning utan **DELAY**:-subrutin tillåts. Används subrutin för **DELAY**: skall det dock vara samma rutin som används överallt med olika argument (i **r16** till exempel).
- Tidsfördröjningen **T** skall anges globalt och **T/2** beräknas där det behövs.
- En slutlig kodstorlek på cirka 50–60 rader är rimlig.

-o-O-o-