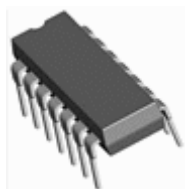


# Tutorial Lab1



## Kretsar och Kopplingsdäck



Vid laborationerna använder vi digitala kretsar ur den så kallade 74-serien i kapslar av DIL-typ (Dual In Line) för hål-montering på mönsterkort. Se bilden. Detta är idag att betrakta som rena reservdelskretsar, och utvecklingen har också gått mot helt andra kapseltyper för yt-montering.

Dessa gamla kretsar är dock oslagbara för experiment med enkla kopplingar på kopplingsdäck. Kopplingsdäck används numera också till mycket komplexa kretsar - som då är monterade på Breakout Board (breakout board har samma pin-avstånd som DIL-kretsar). Att arbeta med kretsar på kopplingsdäck är därmed det naturliga arbetsättet vid projektarbeten och examensarbeten i skolan och för hobbyverksamhet.



[Fritzing](#) Är namnet på en bra programvara för att dokumentera kopplingar med kopplingsdäck, och generera kretsschema, och eventuellt tillverka kretskort. (Detta program kommer att användas i kursen IE1206 men behövs ej för denna kurs). Den som är intresserad kan ladda hem det redan nu.

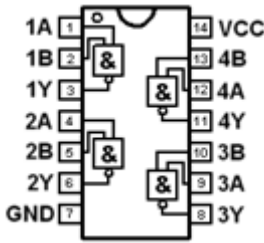
## Datablad till laborationernas kretsar

De digitala kretsarna har kapslar med 14 eller 16 ben. För orientering har kapslarna ett uttag på ena kortsidan, där kan också finnas en "prick" för att markera pinne nummer 1. Två av benen ska anslutas till matningsspänning och jord de övriga pinnarna används som logiska ingångar och utgångar, ibland kan någon pinne sakna anslutning till något inuti kretsen det kallas då för NC (No Connection). Pinnarnas funktion framgår av databladerna, och i viss mån också av symboler och beteckningar i figurerna.

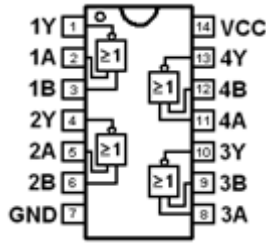
### Några varningar.


Två utgångar får inte kopplas ihop med varandra. Det blir en så kallad kortslutning och utgångsvärdet blir osäkert 1/0 (?). Oanslutna ingångar fångar upp elektriska störningar och måste därför också betraktas som osäkra 1/0 (?) de bör därför alltid anslutas till lämplig fast nivå 1/0 beroende på funktionen.

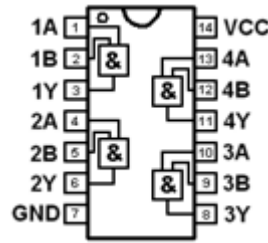
 7400.pdf



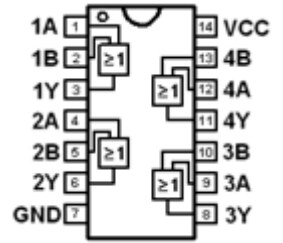
 7402.pdf



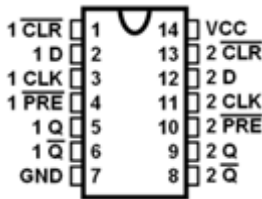
 7408.pdf



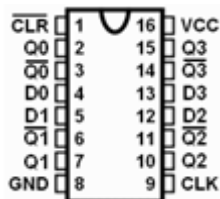
 7432.pdf



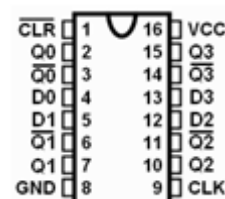
 7474.pdf



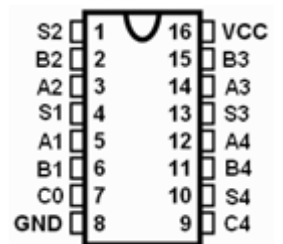
 7486.pdf



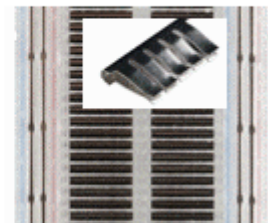
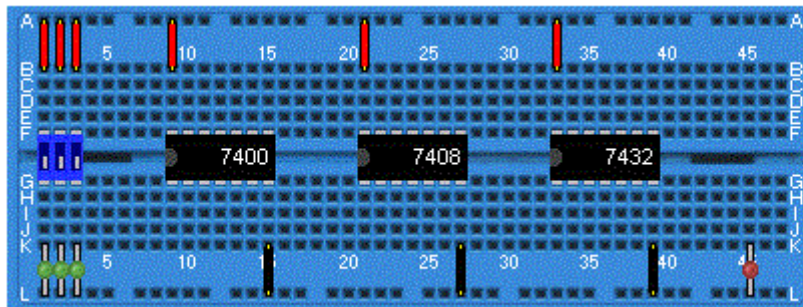
 74175.pdf



 74283.pdf



## Kopplingsdäck



Kopplingsdäck vid laborationen. Matningsspänning och jord är redan anslutna till kretsarna på labutrustningarna. Till höger en bild på kopplingsdäckets undersida med kontaktfjädrarna synliga och en bild på en kontaktfjäder för fem hål.

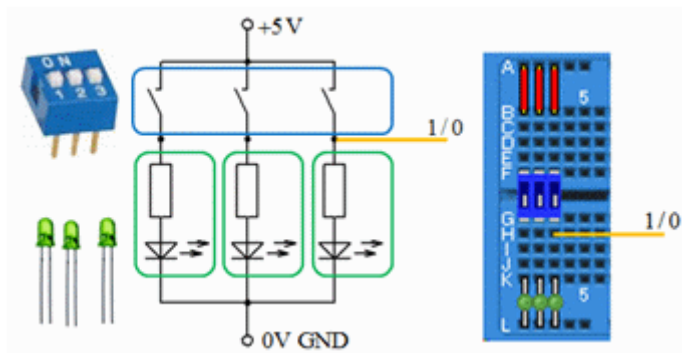
Kopplingsdäck vid laborationen. Kretsarna är instuckna i hål på kopplingsdäckets översida av plast. På undersidan finns kontaktfjädrar av metall som ansluter elektriskt till pinnarna och för vidare anslutningen till ytterligare fyra hål. I hålen kan man sticka ned kopplingsledningar så att anslutningen kan föras vidare till andra pinnar.

Kontaktfjädrarna är i grupper om 5 hål vertikalt. Detta motsvarar en kopplingspunkt (en nod) i ett kopplingsschema. I mitten av kopplingsdäcket finns ett isolerande "dike" så att pinnarna på kretsarnas båda långsidor isoleras från varandra. Kopplingsdäckets långsidor har extra långa kontaktfjädrar, så att alla hålen är här anslutna till varandra. Långsidornas hål används för att ansluta matningsspänning och jord till övriga kretsar. Övre långsidans hål är alla anslutna till +5V och nedre långsidans till Jord (0V).

Många kopplingsdäck har dubbla rader med hål på långsidorna - varje långsida kan då ha både matningsspänning och jord, man kan då enklare undvika att behöva dra ledningar över kretsarna, detta underlättar senare utfall om någon krets går sönder och måste bytas.

Långa kopplingsdäck har långsidans kontaktfjädrar uppdelade i två sektioner och två rader. Sektionerna kan då användas för flera olika matningsspänningar till kretsarna - någon standard för detta finns egentligen inte - för ett sådant fall man får helt enkelt noga ta reda på vad som gäller!

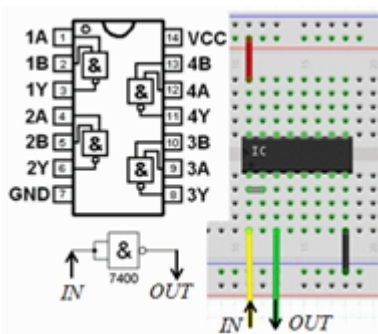
## Övriga komponenter



För att generera logiska insignaler använder vi DIP-switchar och gröna lysdioder. Observera att vi använder speciella lysdioder med inbyggda strömbegränsande resistorer (dom kostar någon krona mer, men förenklar kretsen)! När strömbrytaren är öppen blir logiksignalen 0 på den gula ledningen, eftersom lysdioden är släckt och håller ledningen på 0V. När strömbrytaren är sluten går det ström till den gröna lysdioden, spänningen är +5V och den lyser, och logiksignalen på den gula ledningen är då 1. Man manövrerar DIP-switchen med en skruvmejselspets.



För att se logikkretsarnas utgångsnivåer använder vi röda lysdioder.



## Från schema till kopplingsdäck.

I schemat har vi en NAND-grind kopplad som inverterare. I verkligheten, på kopplingsdäcket väljer man ut en av de fyra NAND-grindarna i chippet. En bygel (grå kopplingstråd) kopplar ihop ingångarna. Gul ledning IN leder till ingångarna och grön ledning OUT till grindens utgång. Matningsspänning (röd ledning) och jord (svart ledning) är anslutna till chippet.

# Simulering av logikfunktioner

- Kan man kontrollera logikfunktioner utan att behöva koppla upp kretsarna?

Det ursprungliga sättet att kontrollera att elektronikkonstruktioner fungerar som avsett var att koppla upp och därefter kontrollmäta på kretsarna. I takt med att elektroniken blivit mer komplicerad och integrerad har detta blivit svårt - samtidigt som datorprogrammen har utvecklats så att det numera är naturligt att i hög utsträckning lita till simuleringar.

1973 utvecklades simulatorprogrammet **SPICE** (Simulation Program with Integrated Circuit Emphasis) vid *University of California, Berkeley* för att användas vid konstruktion av integrerade elektronikkretsar. Det är detta program som används än i dag av elektronikkonstruktörer - och i kursen kommer vi att använda den modernare versionen **LTSpice**!

Spice är en analog simulator, den använder således kontinuerliga värden på spänningar och strömmar. Detta är nödvändigt för den som ska konstruera integrerade kretsar, men för att kontrollera logiska funktioner skulle man kunna nöja sig med en enklare simulator som bara räknar med 1 och 0. Beräkningskraften hos dagens datorer är dock så hög att vi slipper välja!



Besök: <http://www.linear.com/> för att hämta och installera programmet **LTSpice** på din egen dator. Du behöver inte registrera dig om Du inte vill. Installationen är sedan "rättfram" enligt instruktionerna på hemsidan. Programmet finns för de flesta operativsystem, men denna Tutorial gäller utseendet under Windows.

## Kursens simuleringsfiler

Programmet finns installerat på skolans datorer i labsal **Ka-305** och i datorsalen **Ka-309**. I skolan packar Du upp kursens simuleringsfiler i din servermapp under `H:\IE1204`. (hemma spara "var som helst")

Du kan starta LTSpice genom att klicka på valfri  \*.asc - fil.

(Du kan även starta  **LTSpice** från startmenyn).



Om du klickar på simulerings ikonen i programmet så genomförs en simulering med de av oss förvalda inställningarna. Därefter kan Du enkelt gå vidare genom att ändra värden eller inställningar för att själv utforska kursens kretsar!



`IE1204.zip` alla kursens simuleringsfiler.

# Ett första simuleringsprogram med LTSpice

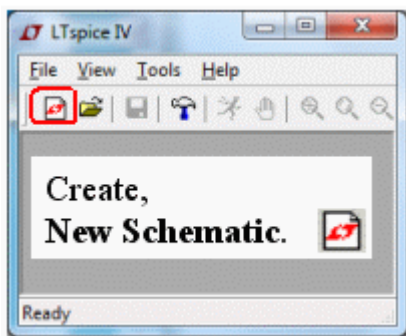
Vi använder lysdioder med inbyggda motstånd - vad är det och varför? Vi tar hjälp av simulatoren **LTSpice** för att förstå detta! Samtidigt får vi pröva på hur man ritat kretsschemor och simulerar med programmet.

## Gör ett elektronikschema



Gör ett elektronikschema med en lysdiod!

 Starta LTSpice. **File - New Schematic**



I skolans datorsalar är det viktigt att Du sedan sparar filerna på ditt serverbibliotek `H:\`

Man hämtar komponenter under **Edit** - menyn. Eller om dom redan finns framme, från **Snabb menyn**.



 Resistor

 Capacitor

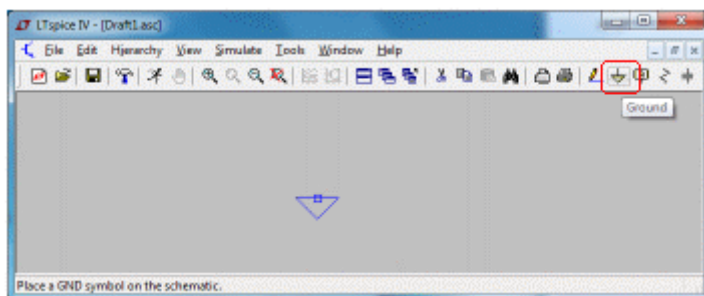
 Inductor

 Component

 Wire

 Label net

## Viktigt! Jordsymbolen



Välj komponent, placera, klicka, tryck på `<esc>` när Du inte längre vill fortsätta att placera ut samma komponenttyp.

För att vrida på komponenterna **Ctrl-R**.



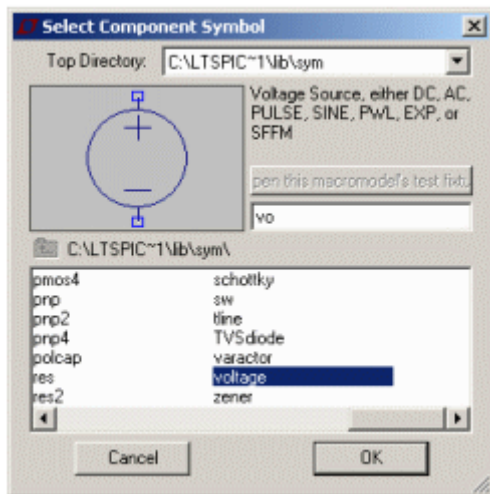
Alla ritningar måste ha en jordsymbol **GND**. Placera den först så att Du inte glömmet den. Detta är helt nödvändigt - inga simuleringar fungerar annars!

## Spänningskälla

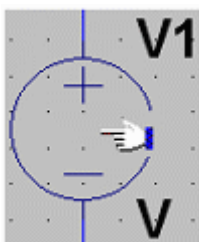
Därefter behöver vi en spänningskälla (ett batteri). Den finns i komponentbiblioteken.



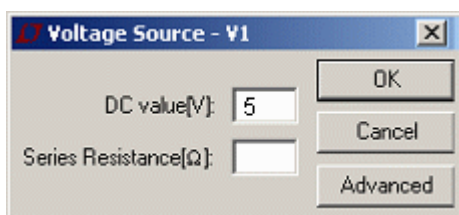
Component, öppnar komponent fönstret. I textrutan kan man söka alla komponenter, komponentnamnet är `voltage` men det kan räcka med att skriva de första bokstäverna `vo..` för att den ska hittas.




Placera spänningskällan med musen, klicka och tryck sedan på `<esc>`. Spänningskällan har beteckningen/namnet **V1** och värdet/modellen **V**.

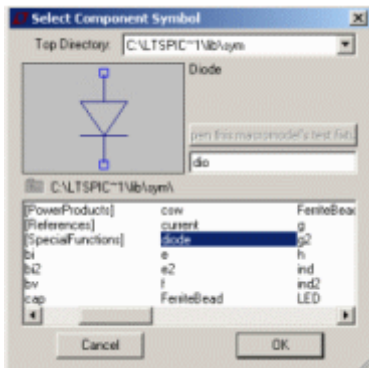


För att förändra en komponent, nu spänningskällan, för man muspekaren över symbolen för **V1** så att den antar formen av en hand - högerklicka och fyll i parametern `DC value[V]` till 5. Nu har vi ett 5V batteri.

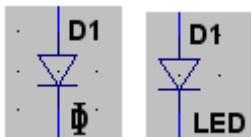


## Diod

 En diod finns i komponentbiblioteken. I textrutan söker man på komponentnamnet diode (dio .. kan räkka). Placera sedan ut dioden på schemat.



I schemat har dioden beteckningen/namnet **D1** och värdet/modellen **D**.




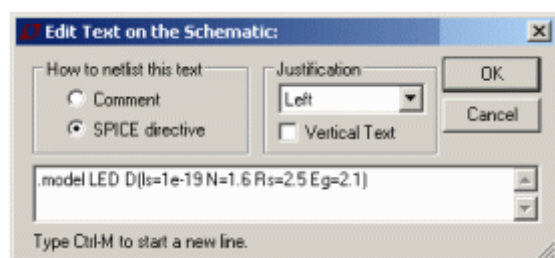
Vi ska inte ha någon vanlig diod - vi behöver en lysdiod (LED Light Emitting Diode), och någon sådan modell finns inte medlevererat till programmet? (Vanliga dioder är tillverkade av Kisel, lysdioderna av GalliumArsenid - skillnaderna i data är stora, inte bara det att lysdioden avger synligt ljus!).

Nu så brukar de flesta leverantörer av elektronikprodukter publicera SPICE-modeller över sina komponenter, så efter lite sökande så har vi hittat följande modell för lysdioden (vi bryr oss här inte om vad parametrarna betyder):

```
.model LED D(Is=1e-19 N=1.6 Rs=2.5 Eg=2.1)
```

För först muspekaren mot **D1** modell-namnet **D** så att den antar formen av bokstaven **I**, högerklicka och byt ut modellnamnet **D** mot något annat namn - tex. **LED**. Därefter måste vi beskriva modellen LED för simulatören. Det gör man enklast genom att föra in modellen som ett **SPICE direktiv** direkt någonstans på schemat.


 Välj SPICE directive på Edit menyn. Kopiera och klistra in modell-texten från ovan. (OBSERVERA att den inledande punkten ingår i direktivet!) Klicka på OK, och placera sedan ut texten på ledig plats på schemat.




```
.model LED D(Is=1e-19 N=1.6 Rs=2.5 Eg=2.1)
```

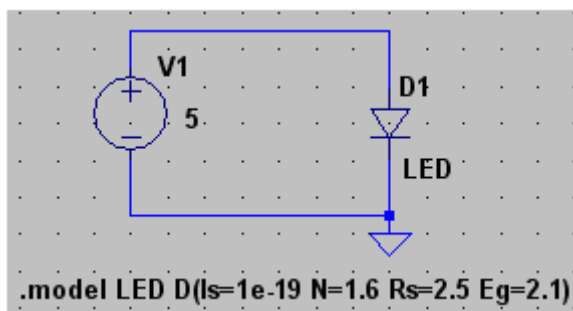
(När det gäller längre modell-beskrivningar blir det otympligt med modellerna direkt i schemat, då brukar man placera dem i textfiler med filnamnstillägget `*.lib.`)

## Wires

 Klicka på Wire. Dra ledningar genom att klicka vid start, vid varje böj och vid slutpunkten. Avsluta med `<Esc>`.

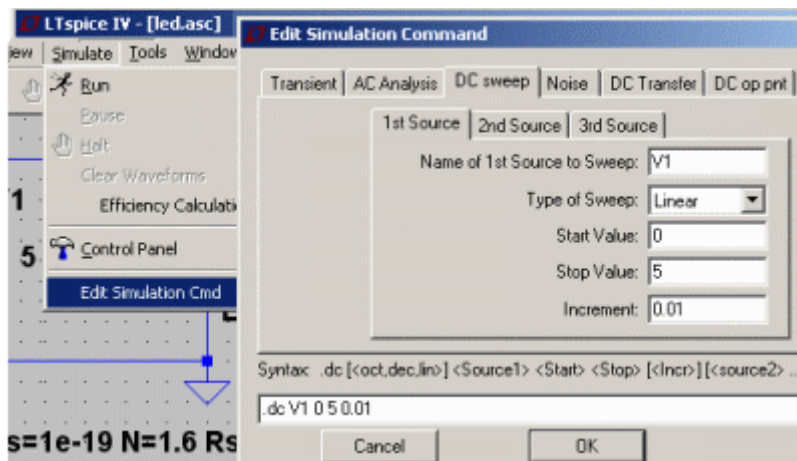
 Move. Kanske behöver man flytta på komponenter, eller texter?

 Cut. Kanske måste man ta bort felaktigheter för att börja om?



## Simulering

Spice har många olika simuleringstyper. **Transient** (tidsvep), **AC Analyses** (frekvens svep), **DC sweep**, **Noise**, **DC Transfer**, **DC op pnt**. För laborationsförberedelserna behöver Du använda **Transient** och **DC op pnt**.



Under **Simulate** menyn, **Edit Simulation Cmd**, kan man ställa in vilken simulering som ska göras.

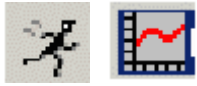
Under fliken **DC sweep** anger vi att källan **V1** ska förändras linjärt från 0V upp till 5V i steg om 0,01V (increment). Vi kommer då att kunna se hur strömmen genom dioden ändras vid olika spänningar - vi kommer däremot *inte* att få se om lysdioden lyser (en så rolig simulator är det inte).



Klicka på OK, och placera simuleringskommandot på schemat. (En van användare hade kunnat skriva dit simuleringskommandot direkt utan att gå via menyvalet **Edit Simulation Cmd**).


```
.dc V1 0.5 0.01
```


## Simulate, Waveform viewer

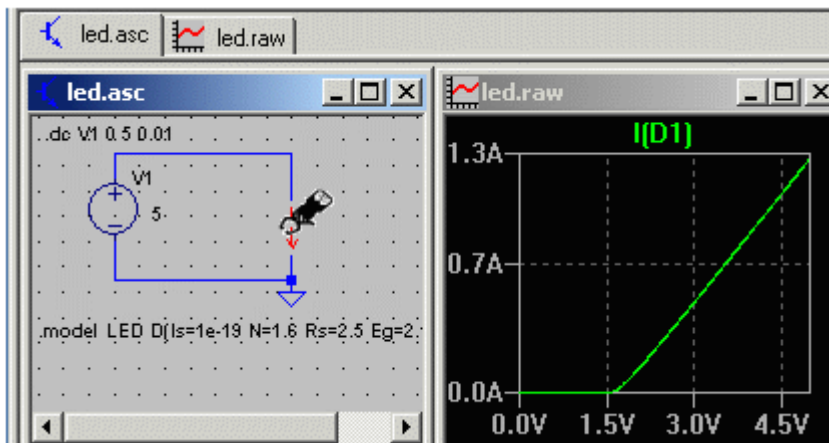



 Run. Simulera genom att klicka på simuleringsikonen.

 Waveform Viewer visas.

 För muspekaren mot någon **ledning** så antar den formen av en **spännings probe** - vänsterklicka för att välja att visa valfri spänning med waveform viewer.

 För muspekaren mot någon **komponent** så antar den formen av en **ström probe** - vänsterklicka för att visa valfri ström med waveform viewer.





 För muspekaren mot Diodsymbolen och vänster-klicka när den förändras till en ström probe. Nu kommer diodströmmen att visas i waveform viewer.

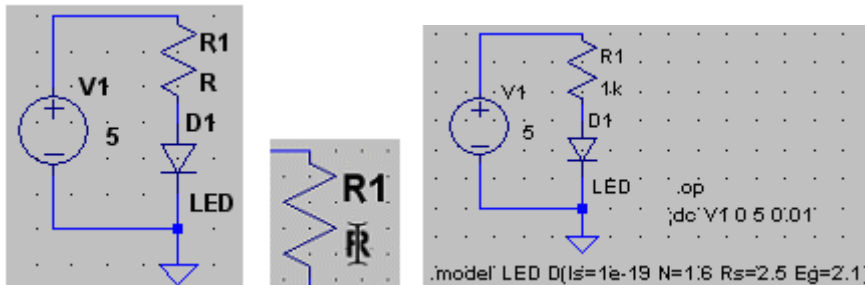
Vi ser att sambandet mellan spänning och ström för dioden är olinjärt (med ett "knä" vid 1,5V). När dioden ansluts till en 5V spänningskälla blir strömmen 1.3A, 1300 mA. Normal ström till en liten lysdiod är 11,5 mA så strömmen blir 100 gånger för hög när lysdioden ansluts direkt till 5V! Lysdioden tål inte detta (100 ggr överström) - det är inget som simulatören visar - utan det är något som en utbildad ingenjör ändå ska förstå. (Visst hade det varit roligt med ljudeffekter och animeringar, men kom ihåg att simulatören ursprungligen härstammar från 1970-talet!).

Lämna simuleringen genom att stänga Waveform fönstret.

- Lysdioder måste ha seriemotstånd för att begränsa strömmen till önskat värde. Vi använder lysdioder med inbyggt motstånd - vilket värde har detta motstånd?

 Klipp bort någon ledning för att bereda plats för en resistor.

 placera ut resistorn.



För muspekaren över **R** vid **R1** så att den antar I-form - högerklicka och fyll i fönstret **Enter new Value for R1** till tex. 1000 (1000 ohm, kan även skrivas 1k).

**Simulate** menyn, **Edit Simulation Cmd**, klicka på fliken **DC op pnt**. Klicka på OK och placera texten i schemat.

Det tidigare simuleringskommandot har fått ett semikolon framför sig (=kommenterats bort). Nu gäller `.op`. Detta är den enklaste simuleringstypen, vi kommer endast att få en textlista över spänningar och strömmar i kretsen.

```


--- Operating Point ---

V(n002): 1.58402    voltage
V(n001): 5         voltage
I(D1):  0.00341598 device_current

```

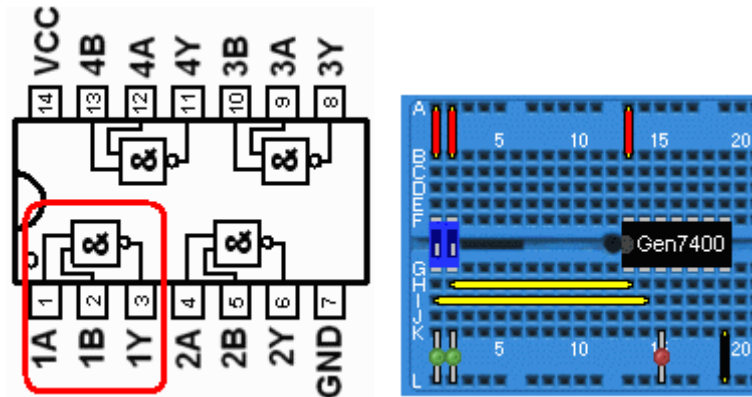
Strömmen genom lysdioden blir tydligen 3,4 mA - vi önskar 11,5 mA (enligt uppgift i datablad) - så vi kan minska värdet på **R1**. Gör det och simulera dig fram till ett rimligt värde. Ange detta värde för serieresistor i förberedelserna till Lab1. Vi kan utgå ifrån att det är ett liknande resistansvärde som finns inbyggt i våra lysdioder.

Det är numera så vanligt med lysdioder så att en del websidor har infört script för att räkna ut serieresistorn. Du kan prova [Online series resistor calculator](#) (använd värdet på forward voltage  $V_f = 2.1$  V).

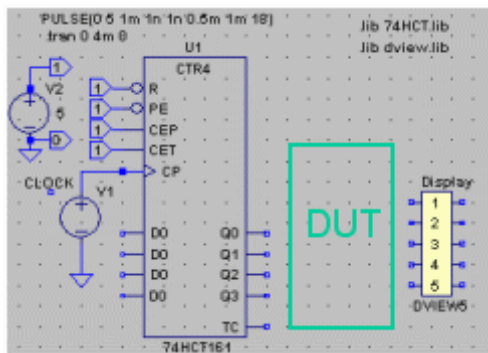
- Om Du har haft stora problem med att få till schemat, så finns det färdigritat som [ledx.asc](#) bland kursens simuleringsfiler. Dubbelklicka på den filen och simulera sedan med  Run.

# Simulering av kombinatoriska kretsar - logikgrundarnas sanningstabeller ( *inför lab Kombinatoriska kretsar* )

## NAND-grind 7400



Testuppkopplingen för att undersöka en NAND-grind vid laborationen *Kombinatoriska kretsar*.






För att simulera sanningstabeller för kombinatoriska nät har vi gjort några mall-filer. Man placerar sitt kombinatoriska nät inom det inringade gröna området i figuren, DUT (**D**evice **U**nder **T**est). Det som behövs är binärkodade in-signaler, och sådana kan man få från en "räknare" som räknar i binärkod (Vi går igenom räknare vid föreläsning 8, men använder en sådan med beteckningen 74161 redan nu!). Q3Q2Q1Q0 räknar från 0000 till 1111. För "scanna" igenom sanningstabellen så ansluter man dem till det kombinatoriska nätets ingångar.

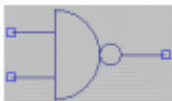
Digitala utgångar är antingen 1 eller 0, visar man dem med **Waveform viewer** så hamnar alla signaler precis över varandra på ett överskådligt sätt. I mall-filerna har vi därför blocket **DVIEW5** som förskjuter fem digitala signaler till en tydlig bild. Det finns också en **DVIEW10** för upp till 10 signaler.

Mall-filen innehåller två Spice-direktiv med namnen på två filer med modeller som vi behöver. Alla 74-seriens logikkretsars modeller finns i filen `74hct.lib` och **Digital Signal Viewer** modellerna finns i filen `dview.lib`.

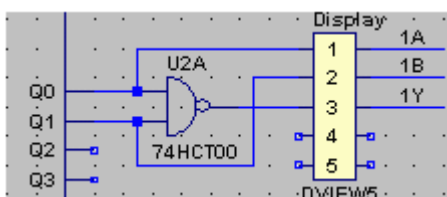
```
.lib 74HCT.lib
.lib dview.lib
```

Öppna nu mall-filen  `truth_template2.asc` och spara den direkt under namnet  `7400_test.asc` ( i skolan i mappen `H:\IE1204\` ).

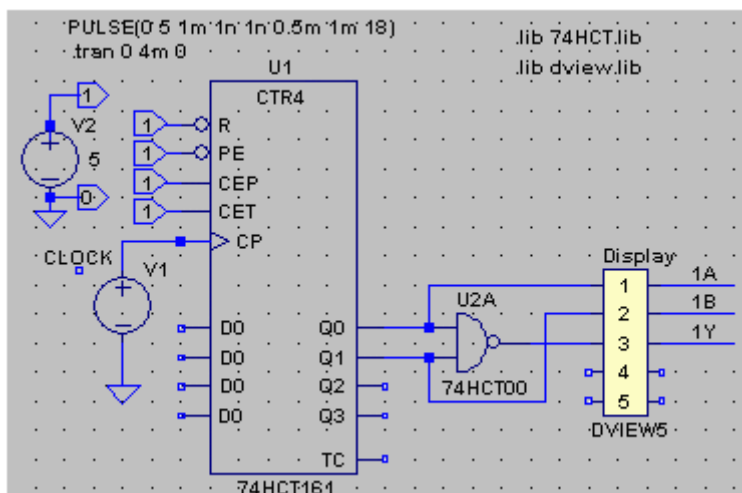
 Component. **Top Directory** Bläddra fram till mappen med kursens simuleringsfiler. Välj där `74HCT00`.



 Dra ledningar.   Snygga upp!



 Label Net. Namnge ledningarna **1A 1B 1Y** så blir det lättare att skilja på signalerna. Placera namnet vid ledningen (= i kontakt med ledningen).



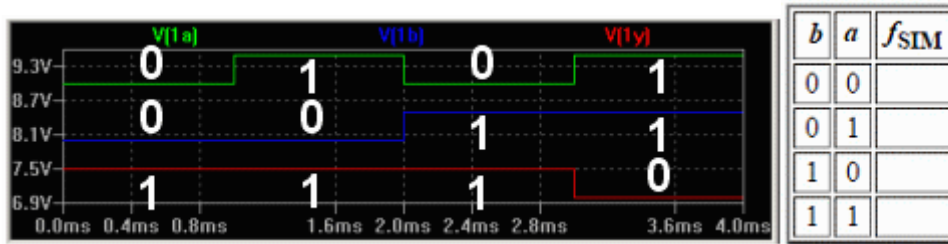
Ritningens simuleringsprofil är denna gång **Transient**, ett tidsvep. Spänningskällan **V2** är konstant 5V och ger oss 0 eller 1 till kretsarna. Spänningskällan **V1** är nu en pulsgenerator som avger 18 pulser i följd. Detta är fler än vad som behövs för att scanna igenom sanningstabellen. Matningsspänningen till logikkretsarna är underförstådd av simulatorm.

```
PULSE(0 5 1m 1n 1n 0.5m 1m 18)
.tran 0 4m 0
```

 Run. Simulera.  Waveform Viewer visas.



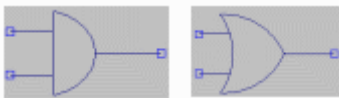
För muspekaren mot ledningen **1A** och klicka när den förändras till en spännings probe. Upprepa för **1B** och **1Y**.



För nu in de simulerade värdena i labbhäftets tabell - vid laborationen får Du sedan jämföra dem med dina uppmätta värden!

- Om Du har haft stora problem med att få till schemat, så finns det färdigritat som [7400\\_testx.asc](#) bland kursens simuleringsfiler. Dubbelklicka på den filen och simulera sedan med Run.

## Simulera nu AND-grind och OR-grind

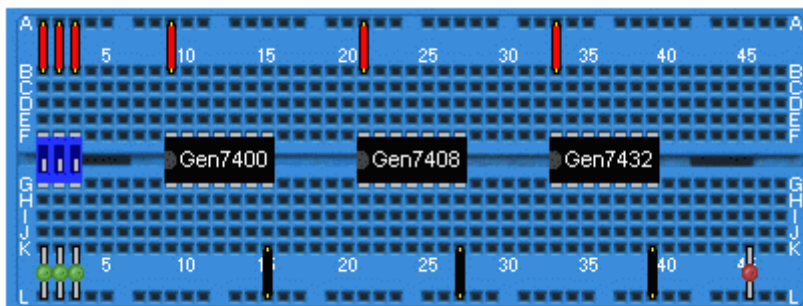


Spara om ritningen som 7408\_test.asc respektive 7432\_test.asc. Redigera om genom att byta till kretsen 74HCT08 respektive 74HCT32.

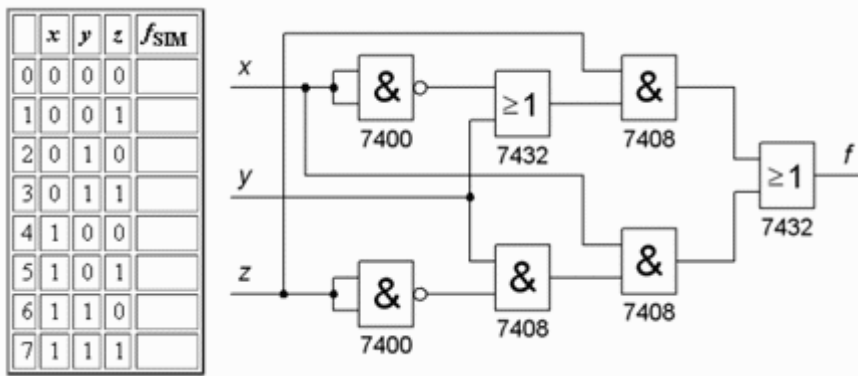
Simulera och fyll sedan i sanningstabellerna i labbhäftet. Vid laborationen får Du jämföra dem med dina uppmätta värden!

[7408\\_testx.asc](#)   [7432\\_testx.asc](#)

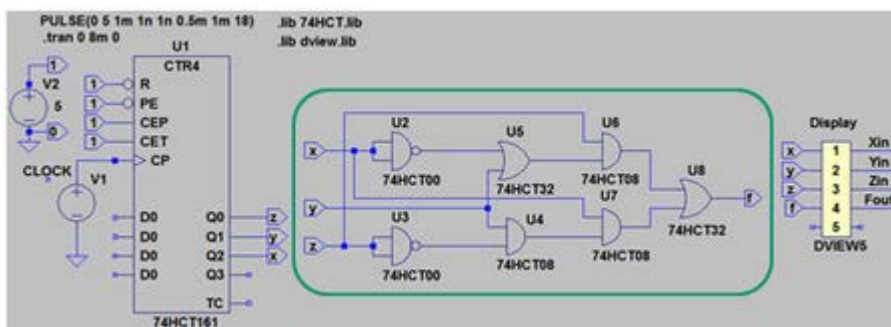
## Simulera kombinatorisk krets ( inför lab *Kombinatoriska kretsar* )



Testuppkoppling för att vid laborationen bygga och undersöka en kombinatorisk krets.



Öppna nu mall-filen [truth\\_template3.asc](#) och spara den direkt under namnet [circuit\\_test.asc](#) ( i skolan i server mappen H:\IE1204\ )



Rita den kombinatoriska kretsen, den inringade delen av figuren, och anslut den till de övriga delarna, räknaren och displayen, så att kretsen kan simuleras.

För att slippa dra ledningar "kors och tvärs" över schemat kan man använda Label Net. Förutom att en punkt i schemat kan ges ett namn, kan man även ange den som en output eller en input.



En output och en input med **samma namn** innebär att de är anslutna till varandra (tex. Z i figuren ovan). (Varning! två output får inte ha samma namn, däremot kan flera input ha samma namn).

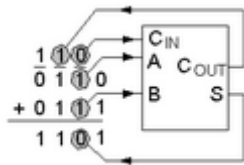
Tips! Med Duplicate kan man återanvända (kopiera, klistra in) sådant som redan finns på ritningen, då behöver man inte leta reda på det i menyerna.

Run. Simulera. Waveform Viewer visas. För därefter in värdena för Fout till kolumnen  $f_{SIM}$  i sanningstabellen i laborationshäftets förberedelseuppgift. Passa också på att studera tidsdiagrammen - kan du direkt komma på vilken slags krets det är?

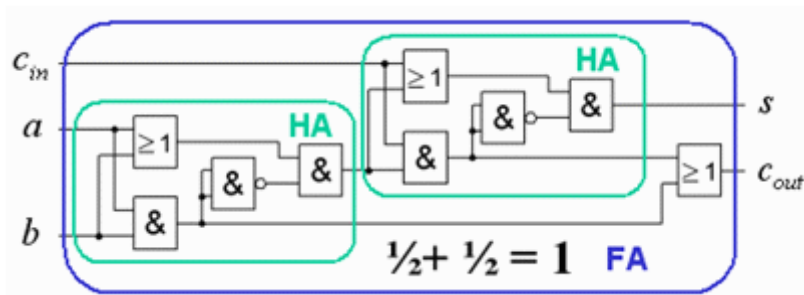
[circuit\\_testx.asc](#)

# Simulera Heladderaren

( inför lab Kombinatoriska kretsar )



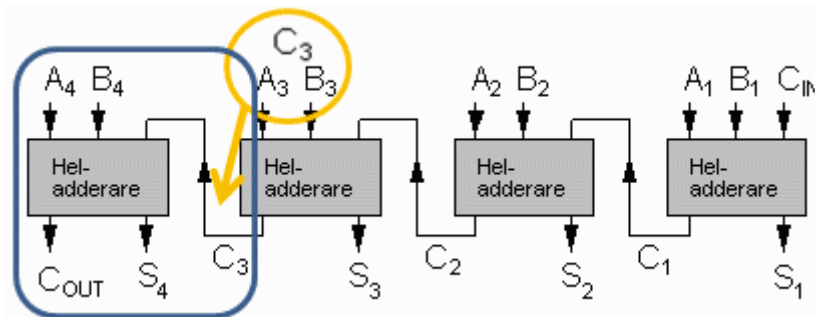
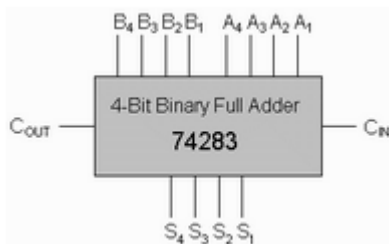
Som förberedelse har Du resonerat Dig fram till vilken sanningstabell som måste gälla för en heladderare. Nedan är en krets som påstås vara en "heladderare".



Öppna nu den färdiga filen [fulladd.asc](#) simulera och kontrollera att resultatet överensstämmer med din sanningstabell. Du kommer att bygga upp kretsen och testa den under laborationen.

## Jämföra kretsar


Ett annat sätt att kontrollera kretsen, är att jämföra den med en färdig halvadderare.



Vid laborationen har vi kretsen 74283 som är en 4-bitsadderare. Den består av fyra heladderare med en Carry-kedja i serie mellan dem. Man kan *inte direkt* komma åt alla anslutningar till en av heladderarna tex. den fjärde med C<sub>out</sub> S<sub>4</sub> A<sub>4</sub> B<sub>4</sub> men då är C<sub>3</sub> bara åtkomlig internt i kretsen.


Man får använda ett knep. Om  $A_3$  och  $B_3$  kopplas ihop så att dom alltid har samma värde kommer dom att generera  $C_3$  eftersom carry ut från en heladderare har samma värde som en *majoritet av* bitarna in.

Öppna nu den färdiga filen [fulladd\\_compare.asc](#) I filen jämförs de två heladderarkretsarna för alla möjliga insignaler. Utsignalerna tas till två xor-grindar för jämförelsen.

 simulera och se om de båda kretsarna är exakt lika. Förmodligen ser Du några avvikelser som Du säkert vill diskutera med labbassistenterna vid laborationen?

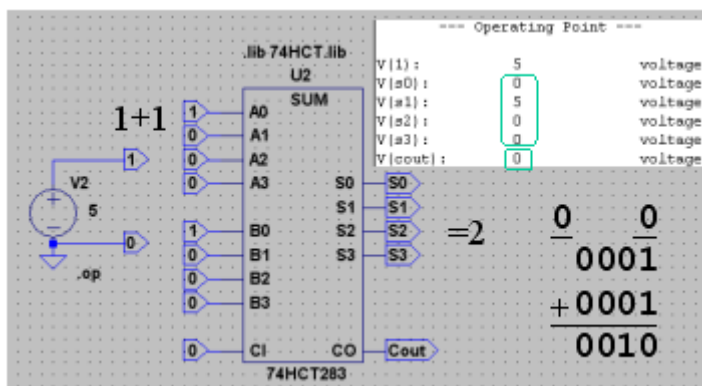
## Heladderarens symmetriegenskaper

Vad skulle hända om man kastar om ordningen mellan insignalerna till en av heladderarna? I filen [fulladd\\_compare\\_crossed.asc](#) har vi gjort så med ledningarna till kretsen 74383.


 simulera och se om det blir någon skillnad - förklara resultatet.

## Addition av 4-bitstal ( *inför lab Kombinatoriska kretsar* )


En fyrabits adderare har en sanningstabell med 256 (28) rader! Ska vi simulera kretsen 74283 så får vi nog nöja oss med några testvärden.

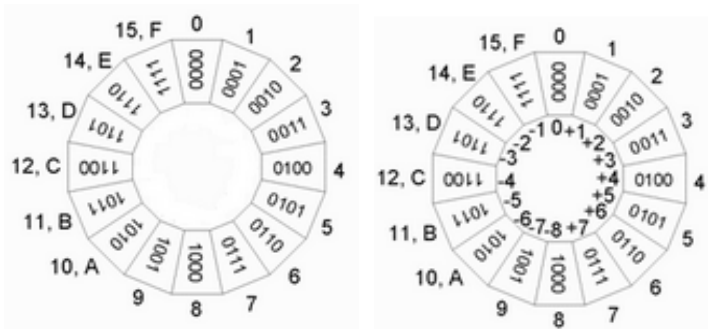


Öppna nu den färdiga filen [4bit\\_add.asc](#) I filen simuleras  $1+1=2$  (och det är ju sant).

Simuleringen ger en textfil med uppgift om spänningar, och 5V betyder där "1". Man ändrar värden genom att föra muspekaren mot input symbolen  tills muspekaren ändras till I-form, då högerklickar man och byter texten till önskat värde.

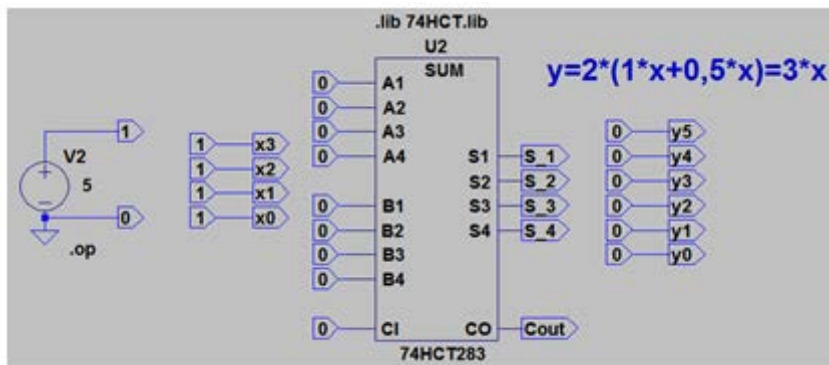


 Simulera med olika siffror så att Du blir säker på addition (och subtraktion) av binära tal, med och utan tecken.



Calculator för addition av hexadecimala tal

## Multiplikation med konstant (3×) med en adderare ( inför lab Kombinatoriska kretsar )



När Du funderat färdigt på hur man kan multiplicera ett 4-bitstal  $x$  med konstanten 3 med hjälp av en 4-bitsadderare kan Du prova din idé med filen [mul3\\_template.asc](#). Du behöver inte dra några ledningar det räcker med att knyta ihop output med input med namnen.

***Lycka till med laborationsförberedelserna!***