Kortlaboration DIK

Digitalteknik, kombinatorik.

I denna laboration bekantar vi oss med datorprogrammet LabVIEW. Programmet har blivit något av en ”industristandard” för att automatisera mätning och styrning, och för att genomföra simuleringar. Här utnyttjar vi programmet som en ”digitalteknisk bygglåda”.

I laborationens första del bygger vi upp de 16 möjliga Boolska funktionerna av två variabler. Här stöter vi på alla de grundläggande grindtyperna och får möjlighet att tillämpa Booles algebra på enkla exempel.

Se till att ni turas om med att använda LabVIEW, så att alla medlemmar i labgruppen får erfarenhet av programmet.

Huvuddelen av laborationen går ut på att realisera ett praktiskt problem, en uppgift beskrivs som skall lösas med hjälp av ett antal logiska grindar.

En fristående uppgift är att jämföra användningen av Binärkod och Graykod i praktiken. Den uppgiften löser Du i mån av tid. (Antalet labutrustningar är begränsat).

1 De 16 funktionerna av två variabler

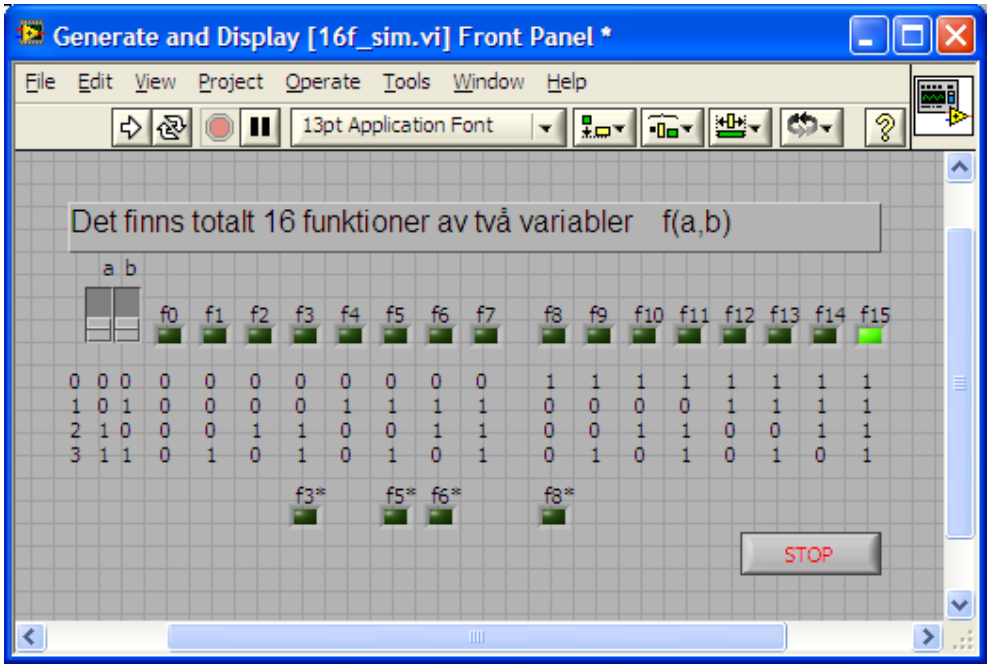
LabVIEW-filer till laborationen finner Du på kurshemsidan

Börja med att ladda hem filen `DIK.zip`. Dubbelklicka därefter på filens ikon för att packa upp filerna och placera dem i en egen mapp.

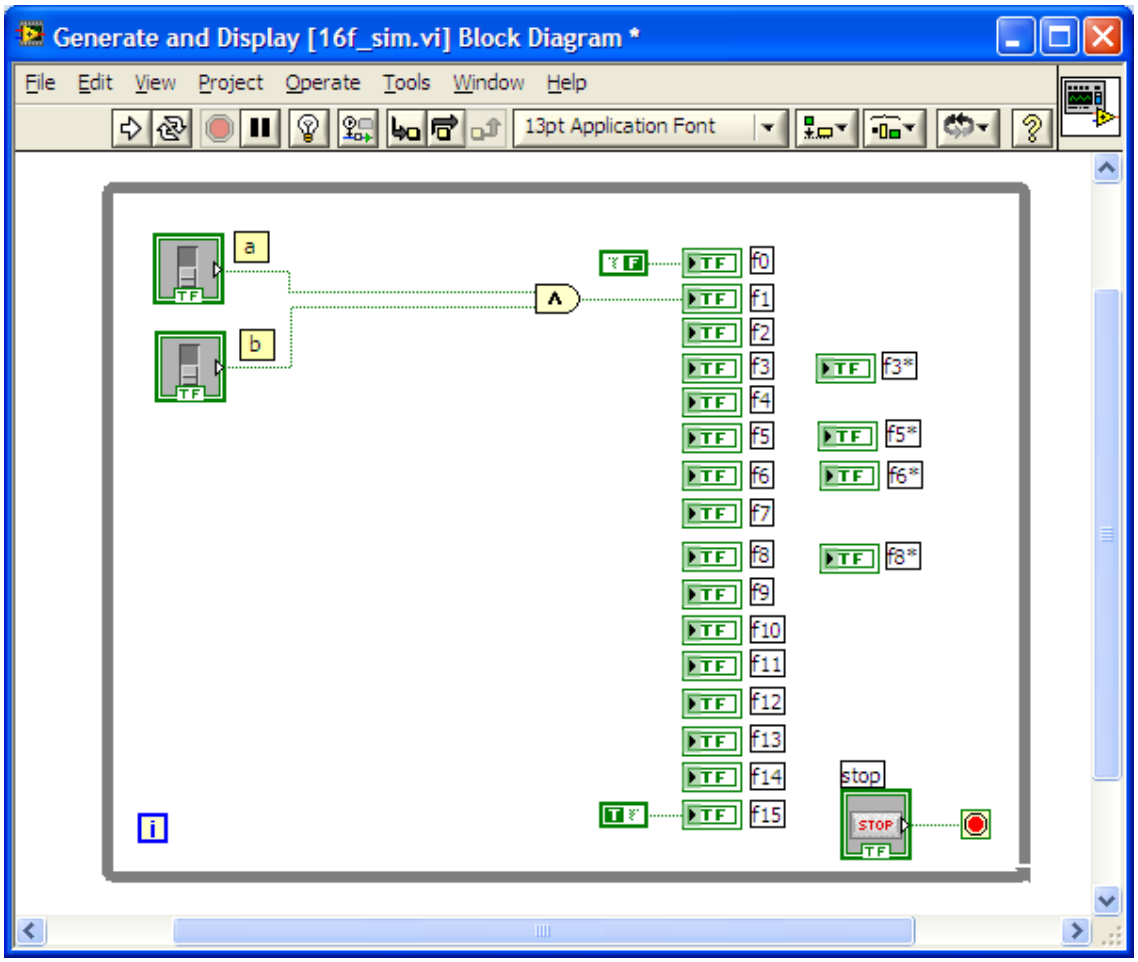
Första filen som ska användas är `16f_sim.vi`. Dubbelklicka på filens ikon för att starta programmet LabVIEW.

LabVIEWs filer kallas för VI (**V**irtual **I**nstruments). Vid körning visas de i två fönster, **Front Panel** och **Block Diagram**.

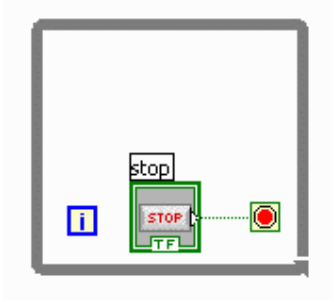
Man startar sitt *virtuella instrument* med startpilen på frontpanelens menyrad, och stannar instrumentet med stop-symbolen, eller med stop-knappen på frontpanelen.



Blockdiagrammet innehåller ett blockschema som specificerar programmet som styr det virtuella instrumentet.



I allmänhet är blocken "inramade" inuti en while-slinga. När man startat programmet med start-pilen genomlöps alla block i tur och ordning, gång på gång, tills man trycker på stop-knappen.



Utan while-slinga kör programmet bara en gång när man trycker på start-pilen. Bredvid start-pilen finns därför en pil för upprepad körning, med den kan man få liknande effekt som med while-slingan.

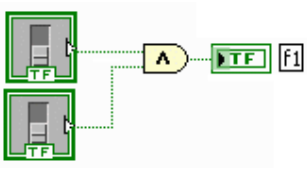
1 .1 Programmet `16f_sim.vi`

Programmet `16f_sim.vi` ska demonstrera de 16 möjliga Boolska funktionerna av två variabler, och hur dessa kan sättas ihop med grindar på en mängd olika sätt. I Digitaltekniken efterstävar man naturligtvis att använda så få grindar som möjligt.

Här begränsar vi oss till två variabler och då kan man ofta se de möjliga förenklingarna direkt ur sanningstabellen. Med fler variabler blir den Boolska algebran ett nödvändigt verktyg för att förenkla funktionsuttryck.

I programmet finns tre funktioner inlagda i blockdiagrammet.
Funktionen f_0 är enligt sanningstabellen alltid noll, oavsett vad man gör med knapparna a och b. Denna (oanvändbara) funktion får man om man ansluter en konstant "0" (False) till indikator f0. På ett liknande sätt får man den (lika oanvändbara) funktionen f_{15} som alltid är ett. Se figur.

Den tredje funktionen som är inlagd i blockdiagrammet är f_1. Enligt sanningstabellen blir funktionen "1" när både a och b är "1". Detta är ju AND-funktionen, och ett block med den funktionen finns tillgängligt i LabVIEW.



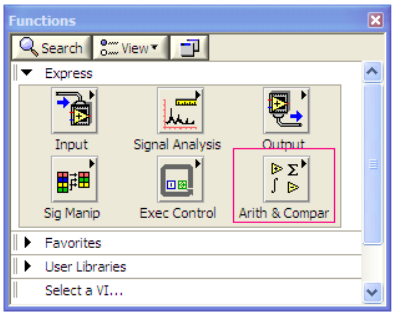
- Provkör det virtuella instrumentet och kontrollera att funktionerna f_0, f_1, och f_{15} blir korrekta enligt sanningstabellerna!

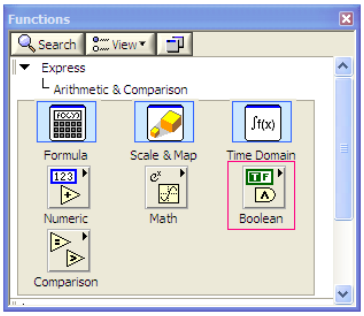
Stoppa därefter programmet.

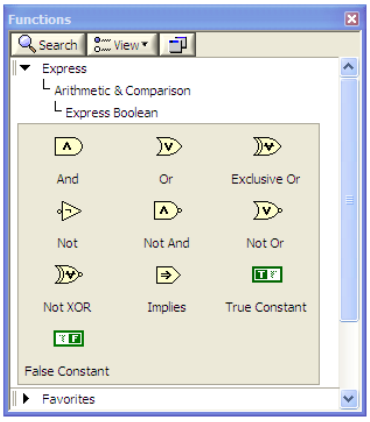
Bygga egna logikfunktioner med LabVIEW

Du ska nu öva på hur man lägger in logiska funktioner i LabVIEWs blockdiagram. För att alla ska få övning med LabVIEW, så bör ni turas om med att bygga funktionerna. Labassistenten kan ge tips och råd.

OBSERVERA! LabVIEW använder sina *egna* grindsymboler. När blockschemat är det aktiva fönstret, når man grindsymbolerna från funktionspalettens ruta **Arith & Compar**, rutan **Boolean**. (Om fönstret med funktionspaletten inte är synligt väljer man blockschemats menyalternativ **View**, menyval **Functions Palette**.)







- Placera komponenter. Välj komponenten genom att placera muspekaren över den och klicka. Muspekarens utseende ändras till en öppen hand. Placera komponenten i blockdiagrammet och klicka för att avsluta.
- Flytta komponent. Klicka på komponenten och dra den med nedtryckt musknapp till den nya placeringen. Släpp musknappen och klicka utanför komponenten.
- Drag tråd mellan noder. När muspekaren är nära en nod ändras utseendet från ett kors till en trådrulle. Klicka, dra och klicka. Använd Esc-tangenten för att "stänga av" tråddragningen.
- Ta bort ledningstråd. Klicka på ledningen, tryck på Delete.
- Tryck på  så får du hjälp med att hitta de oanslutna ingångarna.
- Använd endast AND, OR och NOT om ej annat anges.

Funktionen f_2

Skriv upp funktionsuttrycket och rita kopplingsschema med IEC-symboler (se läroboken). Lägg därefter in schemat i LabVIEWs i blockdiagram (med LabVIEW's egna symboler). Provkör och verifiera att det stämmer med sanningstabellen!

f_2 =

Funktionen f_3

Skriv upp funktionen, rita schema med grindsymboler och för in schemat i LabVIEW.

f_3 =

Om Du tittar lite närmare i sanningstabellen, så ser Du säkert direkt att funktionen kan förenklas? Använd Booles algebra för att förenkla funktionen (även om Du kan se slutresultatet direkt). Skriv upp den förenklade funktionen, rita schema och för in i LabVIEW.

f_3* =

Provkör och verifiera att båda uttrycken stämmer med sanningstabellen!

Funktionen f_4

f_4 =

Provkör och verifiera att det stämmer med sanningstabellen!

Funktionen f_5

Skriv upp funktionen, rita schema med grindsymboler och för in schemat i LabVIEW.

f_5 =

Även här kan man direkt ur sanningstabellen se att funktionen kan förenklas. Gör detta med användande av Booles algebra. För in det förenklade schemat i LabVIEW. Provkör och verifiera att båda uttrycken stämmer med sanningstabellen!

f_5* =

Funktionen f_6

Skriv upp funktionen, rita schema med grindsymboler och för in schemat i LabVIEW.

f_6 =

Funktionen går inte att förenkla. Det är en ofta använd funktion, så den har ett eget namn och finns med bland LabVIEWs symboler.

Vad kallas funktionen:

f_6* =

För in schemat i LabVIEW genom att använda den färdiga symbolen. Provkör och verifiera att *båda* uttrycken stämmer med sanningstabellen!

Funktionen f_7

f_7 =

Du känner kanske igen funktionen:

Provkör och verifiera att det stämmer med sanningstabellen!

Funktionen f_8

Funktionen f_8 är som synes funktionen f_7:s motsats. Om man inverterar utsignalen från den tidigare byggda funktionen f_7 får man f_8. För in funktionen i LabVIEW på detta sätt.

Detta är en välkänd funktion. LabVIEW har en symbol för denna. Vad kallas funktionen:

För även in funktionen med dess symbol i LabVIEW som f_8* . Provkör och verifiera att båda sätten stämmer med sanningstabellen!

(funktionerna f_0 ... f_7 och f_8 ... f_{15} är varandras inverser, så på detta sätt kan man göra med resten av funktionerna.)

Funktionen f_9

Funktionen f_9 är välkänd och har ett namn och en egen symbol. För in den i LabVIEW med symbolen, och verifiera att det stämmer med sanningstabellen.

Vad kallas funktionen:

Funktionen f_{14}

Funktionen f_{14} är välkänd och har ett namn och en egen symbol. För in den i LabVIEW med symbolen, och verifiera att det stämmer med sanningstabellen.

Vad kallas funktionen:

2 Ett styrsystem till en enkel hiss

Hissen i fråga går mellan två våningar. På varje våningsplan finns en knapp där man kallar på hissen. I hisskorgen finns två knappar, upp respektive ner. För att styrsystemet skall veta vilken våning hissen befinner sig på finns en sensor på varje våningsplan som indikerar om hissen är där eller ej.

Insignaler:

1. Knapp på plan 1 (Kp1): Ger 1 om intryckt, annars 0.
2. Knapp på plan 2 (Kp2): Ger 1 om intryckt, annars 0.
3. Knapp ”Upp” i hissen. Ger 1 om intryckt, annars 0.
4. Knapp ”Ner” i hissen. Ger 1 om intryckt, annars 0.
5. Sensor på plan 1 (Sp1): Ger 1 om hissen står på plan 1. Om hissen är mellan våningarna, eller på plan 2 ger 0.
6. Sensor på plan 2 (Sp2): Ger 1 om hissen står på plan 2. Om hissen är mellan våningarna, eller på plan 1 ger 0.

Utsignaler:

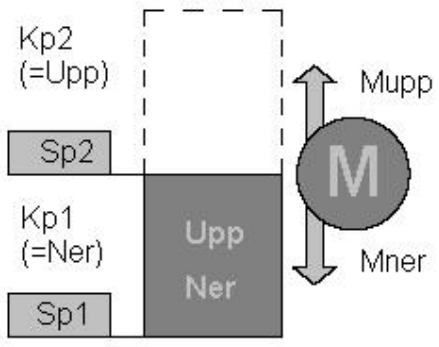
1. Motor upp (Mupp): Om 1 körs motorn uppåt. Om 0 still.
2. Motor ner (Mner): Om 1 körs motorn nedåt. Om 0 still.

Funktion:

Om någon trycker på knappen på våningsplanen skall hissen köras dit. Man kan tänka sig att knappen måste hållas intryckt tills hissen är på plats.

Om någon trycker på en knapp i hissen skall hissen köras i motsvarande riktning. Även här kan knappen tänkas hållas intryckt tills hissen stannar.

Övriga fall såsom om flera knappar trycks in samtidigt skall behandlas på valfritt ”lämpligt sätt”.



Förberedelseuppgift

Tips: Minska ner insignalerna till fyra st genom att koppla ihop några. Att rita Karnaughdiagram med sex variabler är klurigt värre.

Lös uppgiften ”på pappret”. Använd Karnaughdiagram, sanningstabeller och rita upp kopplingsschemat.

Labuppgift

- Provkör filen `hiss_sim.vi` den visar en simulering över hur det hela är tänkt. För att det ska fungera måste även filen `hiss_drive.vi,` som är ett ”sub vi” finnas i samma mapp.

(Simuleringsfilens grindnät är förenklat, detta är *inte* den korrekta lösningen på hissproblemet)

- Det finns ett programskal i filen `hiss_skal_sim.vi`. I blockdiagrammet ska Du införa din lösning på hissproblemet. Simulera olika driftfall med hjälp av programmets strömställare.

Visa nu ditt program för labassistenten innan Du går vidare.

När Du är nöjd med resultatet, så startar även filen `hiss_IO.vi`, den filen innehåller anslutningar till vårt labkort.

- Prova labkortet. PC:ns labkort är anslutet till en kopplingsbox. Drag en ledning mellan utgång P0.0 (Mupp) och ingång P0.2 (Sp2). När man provkör programmet ska man nu kunna tända och släcka lysdioden P0.2 med den simulerade strömställaren P0.0.

Ta bort de två simulerade strömställarna. De ska ersättas med ditt grindnät.

Markera dina grindar i blockschemat `hiss_skal_sim.vi` och kopiera dem på vanligt Windowsvis och klistra in dem i blockschemat `hiss_IO.vi`. De flesta ledningstrådarna följer med, men en del kan behöva kompletteras (eller städas bort). Labassistenten kan bistå med tips!

Ett servicekort med strömbrytare och lysdioder finns på labplatsen. Använd strömbrytarna för att simulera knappar och sensorer. Använd lysdioderna för att simulera motor upp respektive motor ner. Du måste ansluta 5V matningsspänning och jord till servicekortet från spänningsaggregatet. Du måste nu även ansluta kopplingsboxens jord DGND till servicekortets jord. Följ nedanstående kopplingstabell.

Spänningsaggregat	+5V	Servicekort	+U
Spänningsaggregat	GND	Servicekort	
Kopplingsbox	DGND	Servicekort	
Kopplingsbox	P0.0	Servicekort	M-upp
Kopplingsbox	P0.1	Servicekort	M-ner
Kopplingsbox	P0.2	Servicekort	Sp2
Kopplingsbox	P0.3	Servicekort	Kp2
Kopplingsbox	P0.4	Servicekort	Upp
Kopplingsbox	P0.5	Servicekort	Ner
Kopplingsbox	P0.6	Servicekort	Kp1
Kopplingsbox	P0.7	Servicekort	Sp1

- Kör programmet och verifiera att hiss-signalerna blir de önskade. Visa labassistenten ditt program.
- Vi har en hissmodell i LEGO-teknik. Koppla in den och prova funktionen. Hissen behöver 5V matningsspänning och jord. Hissen och kopplingsboxens jord ska sammanbindas. Ställ dig i kö för att testa legohissen.
- Efter redovisning skall alla filer ni laddat ner och sedan ändrat i raderas.

Dukningslista Klab Dik

Digitalteknik, kombinatorik

Antal	Utrustning	
1	Multispänningsaggregat	Står framme
1	Kopplingsbox (ansluten till mätkort)	Står framme
1	Servicekort (strömbrytare, lysdioder mm)	
8	Röd laboratoriesladd	
4	Svart laboratoriesladd	

090310/HJ/MP