LABORATION 4

KALKYLATOR

Laborationens syfte

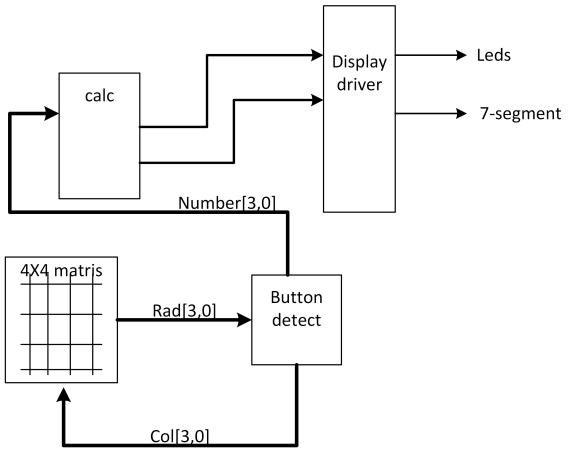
I denna laboration ska en enkel kalkylator konstrueras med hjälp av VHDL och utvecklingsverktyget Vivado från Xilinx. Hårdvaran realiseras på det redan bekanta Nexys4 FPGA-kortet. (se laboration 1) En del av konstruktionen ska lösas med sekvensnät av Moore-typ andra delen med olika register. Uppgiften består av att konstruera en 4x4 matris tangentbordsavkodare samt en enkel kalkylator.

OBS! Det finns ett antal hemuppgifter som skall vara avklarade och uppvisade för att få lov att genomföra laborationen!

.....
Grupp
.....
Namn
.....
Godkänd

Introduktion

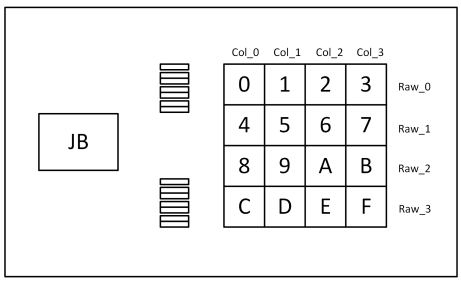
Hela kalkylatorn kommer att bestå av ett antal olika delar enligt figur 4.1



Figur 4.1: Kalkylator

4x4 matris

Matrisen består av fyra rader och fyra kolumner med vardera fyra knappar i varje rad/kolumn vilket medför att vi får ett tangentbord som består av 16 knappar. Då en knapp trycks ned blir det en elektrisk kontakt mellan en kolumn och en rad. Detta ska vi utnyttja för att avgöra om en knapp är nedtryckt och i så fall vilken. Denna enhet finns på ett externt kort som kopplas till Nexys4 FPGA-kortet via kontakten **JB** och ser ut enligt figur 4.2



Figur 4.2: Tangentbord

Via JB-kontakten är de fyra kolumnerna (insignaler till matrisen) och de fyra raderna (utsignaler från matrisen) kopplade. Kortet är även konfigurerar på sådant sätt att om ingen knapp är nedtryckt finns det nollor på Raw_0 – Raw_3.

Kopplingen mellan kortet och Nexys4 är enligt nedanstående tabell:

col	**JB**	**raw**	**JB**
col_0	JB1	raw_0	JB7
col_1	JB2	raw_1	JB8
col_2	JB3	raw_2	JB9
col_3	JB4	raw_3	JB10

Denna information är inskriven i begränsningsfilen (.xdc) tillsammans med signalerna till displayerna och lysdioderna.

Button detect

Denna enhet är den delen som känner av om en knapp är nedtryckt och i så fall vilken. Enheten har utsignalerna Col_0 – Col_3 och number$[3,0]$ samt insignaleran Raw_0 – Raw_3. Tanken är att generera olika och smarta signalkombinationer på Col och läsa av Raw för att på ett lämpligt sätt kunna avkoda vilken knapp som är nedtryckt. Om alla raw är noll, finns det ingen nedtryckt knapp! Resultatet kodas på lämpligt sätt och skickas ut på de fyra bitarna number$[3,0]$ som är anslutna till beräkningsenheten calc.

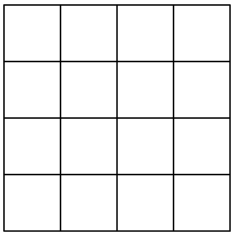
Förslagsvis behövs det knapparna 0 ...9, $+$,$-$, return samt clear, för att få en fungerande miniräknare.

Hemuppgift 4.0.1

Ge ett (bra) förslag på hur signalerna på Col_0 – Col_3 bör se ut för att på ett enkelt sätt kunna detektera vilken knapp som är nedtryckt.

Hemuppgift 4.0.2

Fyll i figur 4.3 hur ni vill organisera ert tangentbord med knapparna 0 – 9 , enter, clear, plus och minus.



Figur 4.3: Tangentbord

Calc

Calc är beräkningsenheten som kan utföra de matematiska operationerna. Detta blir en mycket enkel räknare och kan endast utföra plus och minus med två tal. varje tal består av fyra bitar med mest signifikanta biten som teckenbit (signed).

Hemuppgift 4.0.3

Vilket är talområdet för räknemaskinen?

Räknaren kommer att räkna med omvänd polsk notation (Eng. textitReverse Polish Notation RPN) vilket är en metod som gör det möjligt att skriva matematiska uttryck utan att använda parenteser. Operanderna lagras i register vilket i detta enkla fallet behövs två som sedan operatorn arbetar med.

Exempel 4.1
Beräkningen 5+2 utföres enligt 5 Enter 2 + vilket ger svaret 7.

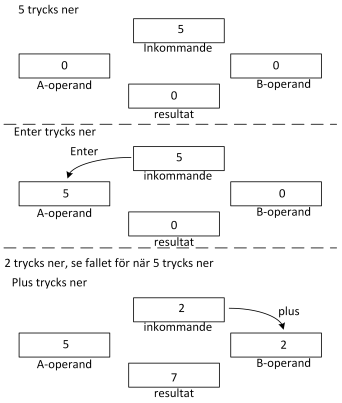
Vad gör koden i calc?

Signalen number innehåller 5 (när knapp 5 trycks ned) vilket lagras i inkommande register.

*När number innehåller **enter** flyttas innehållet i inkommande register till operand-a.*

Signalen number innehåller 2 (när knapp 2 trycks ned) vilket lagras i inkommande register.

När signalen innehåller + eller - flyttas innehållet i inkommande register till operand-b och operationen plus eller minus utföres mellan a och b-operanderna, resultatet hamnar i resultat registret.

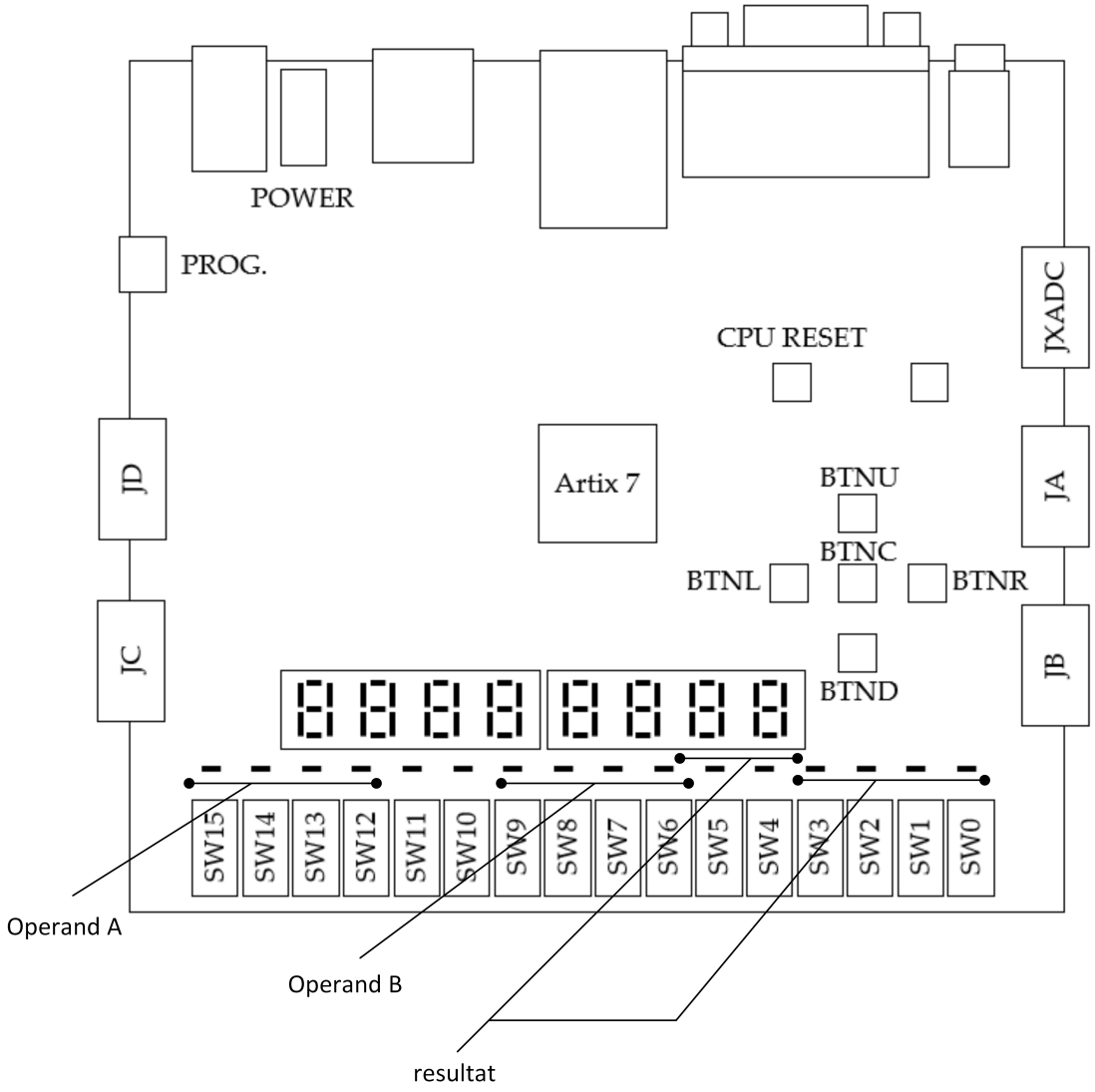


Figur 4.4: Calc-register

Knappen clear nollställer alla register.

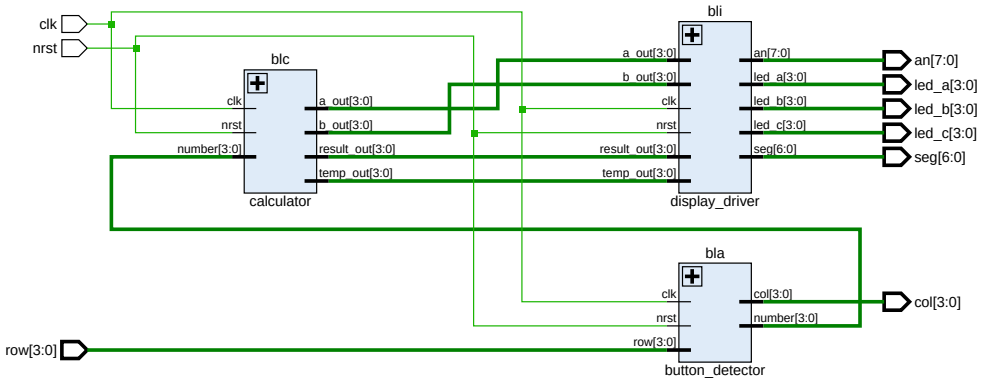
display driver

Enheten som presenterar diverse data på Nexy's-kortet 7-segments diplayer och leds. Modulen är färdigskriven i VHDL. Insignalerna kommer från Calc olika register och presenteras på Nexys-kortet enligt figur 4.5



Figur 4.5: Display

I denna laboration ska vi nu konstruera en mycket enkel kalkylator. Plus och minus mellan två fyrabitars tal är vad den ska klara av. Blocket button_detect ska kompletteras i VHDL medan blocket calc.vhd och display_drive är färdigskrivet. I uppgift 4.2 ska beteendet och koden i calc analyseras. På S-disken finns alla filer ni behöver i VHDL. Allt binds samman av filen kalkylator.vhd. Det finns även en begränsningsfil, kalkylator.xdc, som hanterar kopplingen mellan Nexys4 kortet kuntakt JB och det externa tangenbordskortet. Allt detta bör ni känna igen från föregående laboration. Strukturen och upplägget är på liknande sätt som i laboration 3. Skapa ett nytt projekt och kopiera in alla filerna ni behöver från `S:\Courses\eit\EITA15\Laborationer\Lab4`? Schemat av er konstruktion kommer att se ut enligt figur 4.6



Figur 4.6: RTLschema

När ni editerar VHDL-filen var noga med att inga signalnamn ändras!

Uppgift 4.1. konstruera button_detect i VHDL

Filen button_detect.vhd består av ett antal olika delar. Det finns sekvensmaskinen av typ Moore, en tabell som genererar ett fyrabitars tal beroende på signalerna från Col och Raw. Dessa två delar behöver kompletteras i VHDL för att det ska fungera att detektera om en knapp är nertryckt och vilken knapp det är.

Labbuppgift 4.1.1

Skriv ner ett förslag på hur Moore-maskinen bör se ut. Maskinen behöver ett antal tillstånd (hur många?) för att generera lämpliga signaler på kolumnerna. Dessa genererade signaler tillsammans med avlästa signaler på raderna genererar ett unikt bitmönster (*bit_pattern*)som används för att avkoda nedtryckta knappens binära värde (*number*). Detta förslag ska du skriva in i filen button_detect.vhd på plats labbuppgift 4.1.1

Labbuppgift 4.1.2

Nästa del som behöver kompletteras är tabellen som omvandlar bitmönstret (8 bitar) från tangentbordet till ett fyra bitars binärt tal på number, som i sin tur skickas vidare till beräkningsenheten. Hur tycker du att denna tabell bör se ut? Denna tabell kommer att bestämma layouten på tangentbordet. Skriv in tabellen på plats labbuppgift 4.1.2 i samma fil som tidigare. För att verifiera funktionen är det lämpligt att köra en simulering av button_detect. Kopiera in filen button_detect_tb.vhd i projektet och simulera projektet på liknande sätt som i tidigare laborationer.
Kalla på labbhandledaren och förklara resultatet.

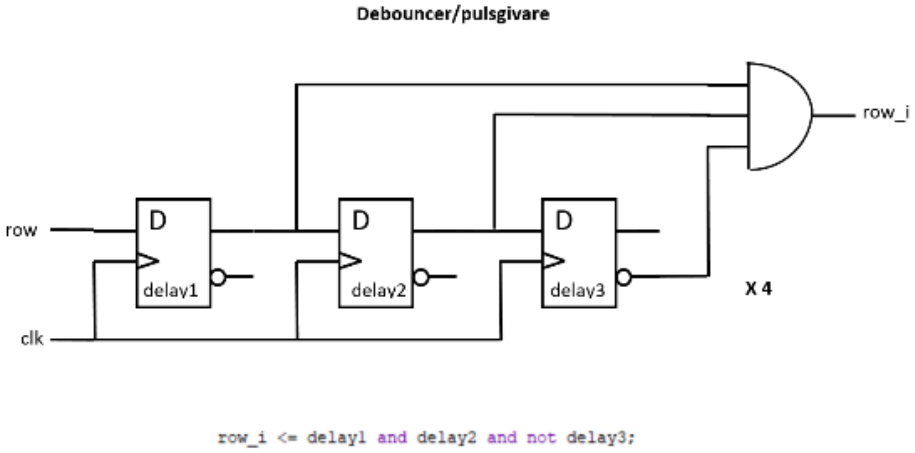
Andra funktioner som är färdigskrivna i button_detect.vhd är en synkronoseringsenhet/flankdetektor vars funktion framgår av figur 4.8.

Hemuppgift 4.1.1

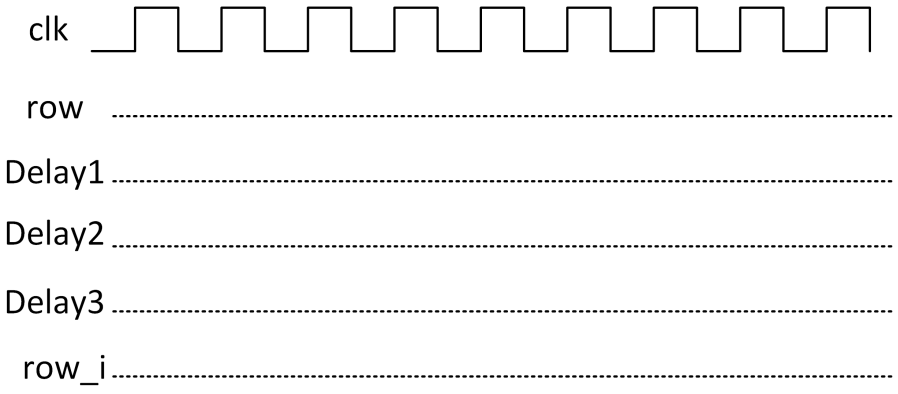
Rita timingdiagram för flankdetektorns row_i.

Det finns även en funktion som har till uppgift att dela ner frekvensen från 100MHz till 1KHz. Detta behöva för att minska kontaktstudsarna som uppstår då man trycker ner eller släpper upp en knapp.

Slut på uppgift 4.1



Figur 4.7: synkronisering/flankdetektor



Figur 4.8: timingdiagram

Uppgift 4.2. Undersök kalkylatorns funktion

Nu när vi kan detektera nedtryckt knapp, avgöra vilken knapp och skapa värde på signalbitarna number, är nästa uppgift att analysera beräkningsenheten Calc.
Filen calc.vhd innehåller koden till en fungerande beräkningsenhet. Blocket består i princip av fyra olika register.

Labbuppgift 4.2.1

Titta i filen calc.vhd och skriv ner namnen på de olika registren. Exempel 4.2 visar ett sätt att skapa ett register i VHDL.

Exempel 4.2
*4 bitars register med **en** och asynkron **reset***

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity d_reg is
Port ( clk : in STD_LOGIC;
nrst : in STD_LOGIC;
en : in STD_LOGIC;
d_in : in std_logic_vector (3 downto 0);
d_out: out std_logic_vector (3 downto 0);
end d-reg;

architecture Behavioral of reg is
process(clk, nrst)       -- Asynkron reset
begin
if(nrst = '1') then

d_out <= (others => '0');

elsif CLK='1' and CLK'event then

if (en = '1') then
d_out <= d_in;
end if;

end if;
end process;

end Behavioral;
```

Labbuppgift 4.2.2

Operationerna plus och minus utföres mellan operand A och operand B. Resultatet sparas i ett register som sedan skickas till display_drive, se figur 4.5.
Vad är det för typdeklaration på detta register? (analysera calc.vhd)
Vad medför detta för räknarens talområdet?

Labbuppgift 4.2.3

Knapparna 0 till 9 kan ni koda hur ni vill , medan operationerna + , - , clear och enter har vardera en förutbestämd kod. Analysera koden i calc.vhd och avgör vilken kod respektive funktion har.
CLEAR =
ENTER =
+ =
- =
Utför några olika typer av beräkningar och förklara för labbhandledaren.

Slut på uppgift 4.2