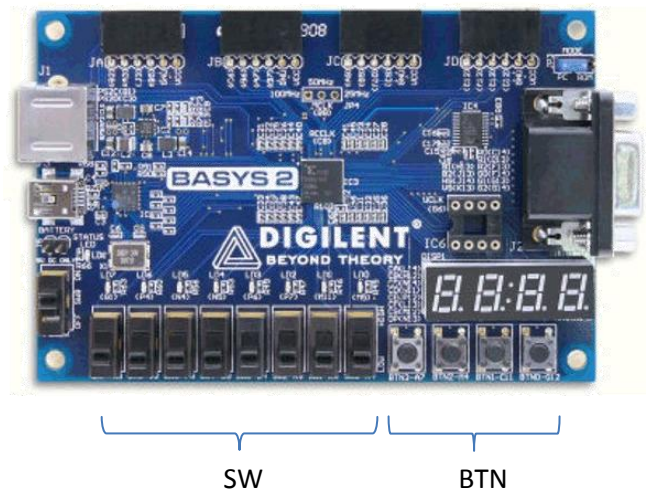
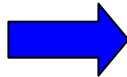


Lommeregner

I dette projekt skal der implementeres en programmerbar lommeregner på et Basys 2 FPGA-kort.



SW

BTN

Funktions beskrivelse:

Lommeregneren skal implementeres med følgende aritmetiske operationer: addition "+", subtraktion "-", multiplikation "*" og division "/" .

Lommeregneren skal kunne programmeres til at udføre følgende beregning

$$\text{RESULT} = (\text{InputA Op1 InputB}) \text{ Op2 Input C}$$

Input A, B og C er tre 8 bit værdier der indtastes via trykknapper(BTN) og switche (SW) på FPGA boardet. Op1 og Op2 er aritmetiske operationer som ligeledes indtastes (programmeres) via trykknapper og switche.

SW	00000001	00000010	00000011	00000100
Op	+	-	*	/

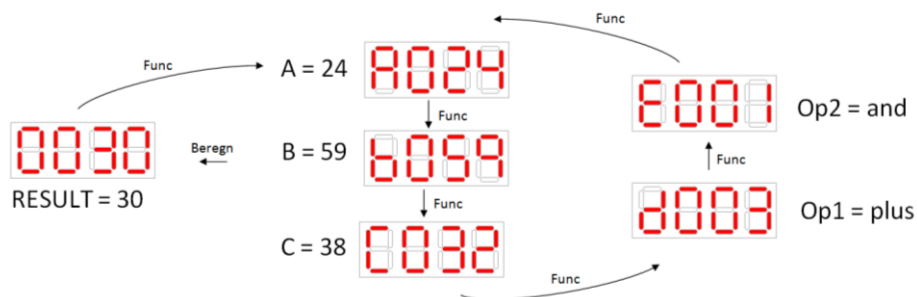
Sw er de 8 skydeknapper (swiche) på boardet.

De fire trykknapper (BTN0-3) bruges til at styre hvad der indtastes. Funktionen af de enkelte knapper fremgår af nedenstående tabel:

BTN3	BTN2	BTN1	BTN0
Reset	Beregn	Func	Enter

Enter (BTN0) knappen bruges til at opdatere et af de 5 registre (InputA-B, Op1-2), Func (BTN1) bruges til at skifte mellem de fem registre, Beregn (BTN2) sætter beregningen i gang., Reset (BTN3) er et asynkront master reset for alle registre på boardet.

Syvsegment displayet bruges først og fremmest til at vise resultatet (RESULT) af beregningen men også til at vise indholdet af de fem registre (InputA-B, Op1-2). Værdien af registrene skal vises i hexadecimal form.



Figuren herover viser et eksempel på hvordan indtastning og beregningsproceduren skal foregå.

Man kan med fordel benytte lysdioderne til at indikere hvilken tilstand man befinder sig i.

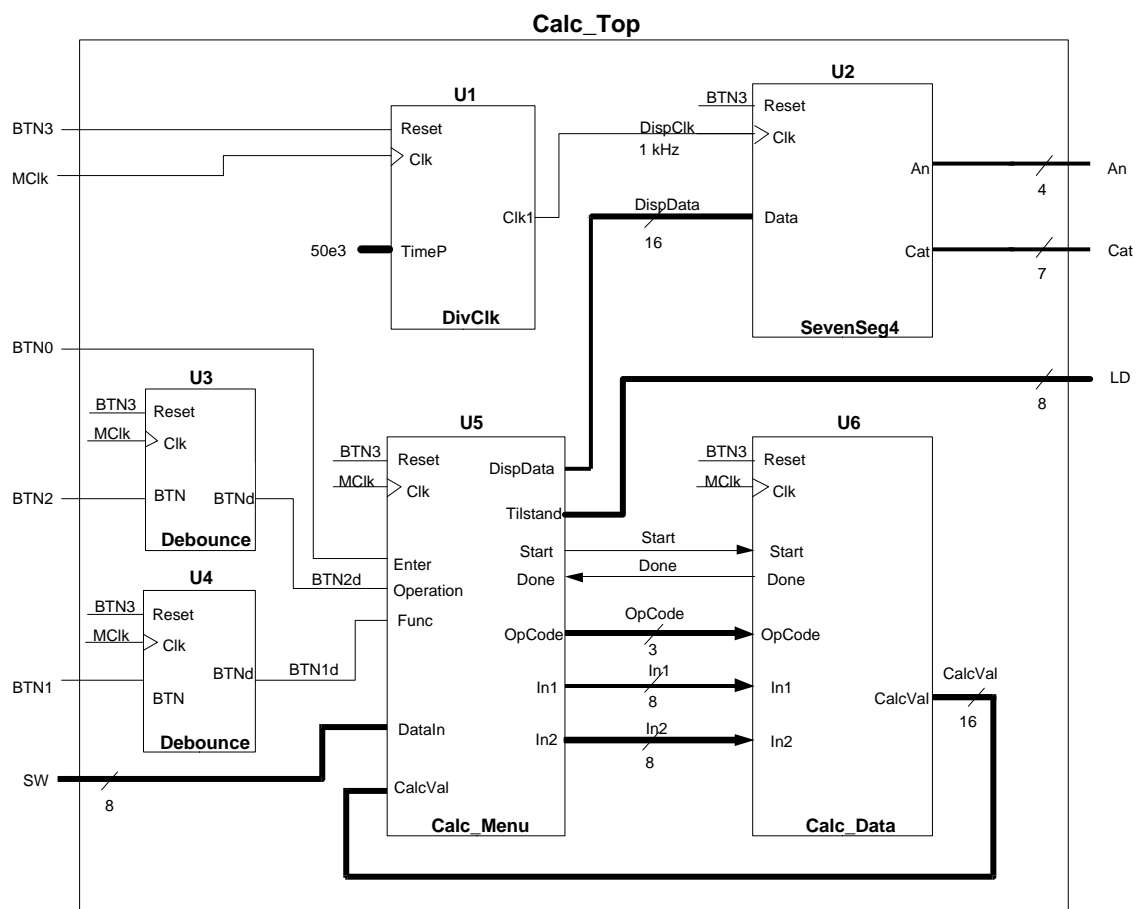
Løsningen af opgaven kan følge nedenstående systemdesign og interfacedefinition. Hvis der laves ændringer i forhold til nedenstående skal der redegøres for det.

Division skal udføres ved hjælp af sekventiel binær multiplikation. Generel information omkring digital division kan findes på campusnet.

Systemdesign og interfacedefinition

Systemet (lommeregneren) skal opbygges v.h.a. en top komponent med 6 underkomponenter.

Nedenstående figur viser funktionsdiagrammet af top komponenten og hvordan de 6 underkomponenter er forbundet. Bemærk at modulet "Debounce" er instantieret 2 gange



DivClk er et modul til at neddele FPGA-kortets 50 MHz klok. Dette kan bruges til at generere et 1 kHz kloksignal til registrene i **SevenSeg4** modulet for at displayet opdateres optimalt.

Indgange	Reset	Master reset
	Clk	Master Clock
	TimeP :integer	Bestemmer hvor mange gange klokken skal neddeles
Udgange	Clk1	Det neddelte kloksignal. Bruges her til Display clock

Dette modul er givet på forhånd og kan findes på Campusnet.

SevenSeg4 er et modul til at styre syvsegmentdisplayet så det kan vise et 16 bit tal (hexadecimalt) på de fire cifre.

Indgange	Reset	Master Reset
	Clk	Display clock (ca. 1 kHz)
	Data	Et 16 bit tal der skal vises på displayet

Udgange	Anode,	Anode signal til dioderne på 7 segment displayet
	Cathode	Cathode signal til dioderne på 7 segment displayet

Information om 7-segment displayet kan findes brugervejledningen af FPGA boardet og i VHDL Primer.

Dette modul er givet på forhånd og kan findes på Campusnet. Hvis man ønsker at vise andre tegn end de hexadecimale cifre skal der laves en ændring af dette modul (vent med det til det andet virker).

Calc_Menu er kontrolleren (tilstandsmaskine + tilhørende registre og logik) for systemet og holder bl.a. styr på brugerfladen (Input, Operation, Resultat). Det sørger for at udskrive input, operation og resultatet via **SevenSeg4** modulet og beder de aritmetiske operationer i Calc_Data om at blive udført v.h.a. signalerne input1, input2 og et enable signal.

Indgange	Reset	Master Reset. Styret af BTN3
	Clk	Master Clock (50 MHz)
	Beregn	Starter den programmeret beregning.
	Func	Skifter mellem de forskellige menu punkter. Styret af et afprellet BTN1 signal
	Enter	Fastfryser en indtastning af DataIn. Styret af BTN0
	DataIn	Bruges til at indtaste input værdien. Styret af de 8 SW knapper.
	Done	En puls (høj i en master clk periode) som signalere at Calc_Data er færdig med begerning.
	CalVal	Resultatet af den aritmetriske operation beregnet af modulet Calc_Data
Udgange	DispData	Et 16 bit tal som vises v.h.a. SevenSeg4. Tallet skal angive input, operationen eller resultatet afhængig af menutilstanden
	In1	Værdien af input1 til aritmetrisk operation.
	In2	Værdien af input2 til aritmetrisk operation
	OpCode:	En kode der angiver hvilken operation som Calc_Data skal udføre.
	Start	En puls (høj i en master clk periode) som signalere at Calc_Data skal starte beregning.
	Tilstand	En kode der viser hvilken tilstand man er i. Dette går ud til de 8 lysdioder på FPGA boardet.

Debounce er et modul der sørger for at afprelle trykknappen Func og Operation. Hvis dette ikke gøres risikerer man at der skiftes ukontrolleret mellem menu tilstandene INPUT1-INPUT2-OPERATION-RESULT (eller addition-subtraktion-multiplikation-division).

Indgange	Reset,	
	Clk,	
	BTN	Signal fra trykknop som skal afprelles
Udgange	BTNd	Afprellede trykknop signal. Signalet er høj i en klokperiode efter BTN knappen er sluppet. Ved kontaktskift ventes en passende periode før kontakten checkes igen.

Information om afprellning af trykknapper kan findes i VHDL Primer eller her:

[http://www.eewiki.net/display/LOGIC/Debounce+Logic+Circuit+\(with+VHDL+example\)](http://www.eewiki.net/display/LOGIC/Debounce+Logic+Circuit+(with+VHDL+example))

Dette modul er givet på forhånd og kan findes på Campusnet.

Calc_Data er Datapath for systemet. Dette modul udføre de aritmetriske operationer. Et signal (OpCode) bestemmer hvilken operation der udføres. Addition, Substraktion og Multiplikation kan udføres direkte ved brug af + og -. Multiplikation kan udføres ved brug af * men vil i det tilfælde bruge en indbygget 16bit multiplikations enhed i FPGAen. Division tager mere end en clock cycle at udføre og der er derfor indført et "Start" og et "Done" signal. På campusnet er der en note der beskriver hvordan division kan udføres.

Indgange	Reset	Master Reset. Styret af BTN3
	Clk	Master Clock (50 MHz)
	In1	Værdien af input1 til aritmetrisk operation.
	In2	Værdien af input1 til aritmetrisk operation.
	OpCode:	En kode der angiver hvilken operation som Calc_Data skal udføre.
	Start	En puls (høj i en master clk periode) som signalere at Calc_Data skal starte beregning.
Udgange	Done	En puls (høj i en master clk periode) som signalere at Calc_Data er færdig med begerning.
	CalVal	Resultatet af den aritmetriske operation beregnet af modulet Calc_Data

Ved løsning af opgaven skal følgende designregler følges:

- Der må kun bruges standard register samt logik-blokke (RTL-Design)
- Sekventielle funktioner skal fortrinsvis udføres af tilstandsmaskiner

- Alle registre bør køre på samme klok. Hvis der er behov for flere clock-domæner skal der redegøres for nødvendigheden samt evt. konsekvenser
- Asynkrone input signaler (på nær reset) bør synkroniseres og evt. debounces (afprell)
- Registrenes asynkrone reset må ikke benyttes aktivt i det synkrone netværk.
- Variable foretrækkes frem for signaler hvor det er muligt.
- Implementeringen af addition/substraktion/multiplikation kan implementeres vha de simple operatorer "+", "-", "*"
- Implementering af division skal være sekventiel.

På CampusNet kan i finde moduleerne DivClk, SevenSeg4 og Debounce som kan bruges ved løsning af opgaven.

Der skal således laves 2 VHDL sub-moduler (Calc_Menu og Calc_Data) samt et top-modul (Calc_Top) som binder de 6 sub-moduler sammen. En projektgruppe (bestående af to 2-mands hold) skal lave projektet sammen. I skal i gruppen aftale hvem der tager sig af hvilke moduler.

I skal sørge for med jævne mellemrum at mødes i hele gruppen for at aftale hvordan moduleerne "taler" sammen. Når de to sub-moduler er færdige kan hele projektgruppen lave top-modulet (Calc_Top) sammen.

Der skal for hver projektgruppe (4-mandshold) afleveres tre udgaver af rapporten,

.....O (Draft udgave)
.....O (- udgave)

Man kan få tilbagemelding på d

Lommeregneren (Draft udgave)

Rapporten uploades på O hvis man ønsker en tilbagemelding på diagrammerne. Afleveringsdatoen fremgår af lektionsplanen på CampusNet.

Draftudgaven skal indeholde

- Løsningsforslag til alle originale (dem i selv skal lave) underkomponenter i form af funktionsdiagrammer, timingdiagrammer og/eller tilstandsdiagrammer. Diagrammerne kan være billeder af håndskrevne tegninger (bare de er læsbare).

Draftrapporten bør således ikke indeholde VHDL kode eller beskrivelse deraf. Den skal udformes så jeres medstuderende kan forstå jeres design og skal så vidt muligt opbygges som en rapport med indledning samt beskrivelse af jeres diagrammer. Tekst skrevet her kan bruges i den endelige rapport.

Tilbagemeldning

Tilbagemeldning af draft rapporten vil foregå mundtligt i projektgrupperne. Lektionen efter aflevering vil således gå med at alle grupper fremlægger deres design for underviser eller hjælpelærer og får feedback udfra de fremlagte diagrammer på hvordan opgaven bedst løses.

Lommeregneren (Endelig..udgave)

Rapporten uploades på CampusNet (under "opgaver") hvis man ønsker en tilbagemelding af rapporten. Afleveringsdatoen fremgår af lektionsplanen på CampusNet.

Rapporten skal indeholde beskrivelser af alle komponenter som projektgruppen har lavet og der skal gives en status for implementeringen.

Beskrivelse af en komponent bør indeholde følgende

- Tekstbeskrivelse af funktion.
- Beskrivelse af registre og logiske blokke
- Funktionsdiagram (forbindelser mellem processer og underkomponenter)
- Timingdiagrammer og/eller tilstandsdiagrammer
- Simuleringstest med kommentarer
- Kommenteret VHDL kode (som bilag)

Rapporten skal opbygges som en teknisk rapport med indledning, krav, design, test og konklusion. Den skal skrives til censor som ikke nødvendigvis kender opgaven/hardware men har tilstrækkelig teknisk indsigt til at forstå jeres løsning. Den skal indeholde en detaljeret beskrivelse af arbejdsdelingen mellem de to hold i projektgruppen og det skal fremgå hvilket hold der er ansvarlig for hvilke dele af rapporten. Original kildekode til projektet skal være vedlagt som bilag til rapporten (i samme pdf fil). Koden formateres så den er let læselig.

Tidsplan

	Projektgruppe	Hold A	Hold B
Lektion 6	Læs og forstå opgaveformuleringen Fordel VHDL modulerne mellem de to hold og lav tidsplan (denne) Diskuter hvordan modulerne skal tale sammen	Hold A tager sig af x: ????? Arbejd med Design_x	Hold B tager sig af z: ????? Arbejd med Design_z
Tidsplan	Aflever Tidsplan (denne)		
Lektion 7	Skriv Draft rapport	Design_x færdig	Design_z færdig
Draft rapport	Aflever Rapport		
Lektion 8	Diskuter hvordan modulerne skal tale sammen	Bearbejd Design_x VHDL kodning af Design_x	Bearbejd Design_z VHDL kodning af Design_z
Lektion 9		Simmulering af Design_x	Simmulering af Design_z
Lektion 10	VHDL kodning af Calc_Top	Skriv afsnit om Design_x	Skriv afsnit om Design_z
Lektion 11	Simmulering af Calc_Top Implementer Calc_Top Færdiggør rapporten	Skriv Indledning og konklusion	Skriv om det overordnede desin (Calc_Top)
Final rapport	Aflever Rapport		

Med "Design" menes udformning af Funktionsdiagram samt Timingdiagram og/eller Tilstandsdiagram