



Lunds Universitet
LTH
Ingenjörshögskolan
Ida, IEA
Helsingborg

Laboration nr 5 i digitala system, ht-13

Introduktion till syntesverktyget Altera Max+PlusII

**Beskrivning i VHDL och realisering
av några enkla kombinatoriskt nät.**

.....
namn

.....
godkänd (datum/sign)

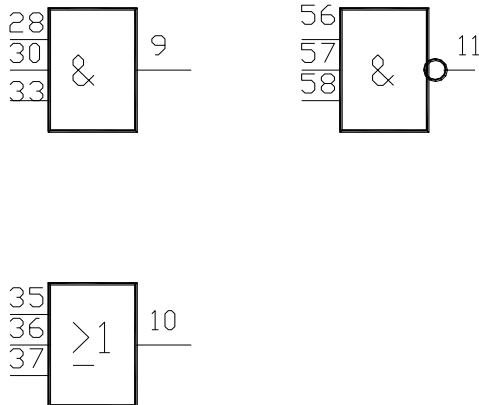
Del1:

Laborationens syfte

- ge en introduktion till hantering av Altera Max + PLUS II
- ge träning i att beskriva en kombinatorisk krets i VHDL

Förberedelseuppgifter

Studera i läroboken, grunderna i VHDL. (t.ex. sid. 41-52, 398-400,402-404)

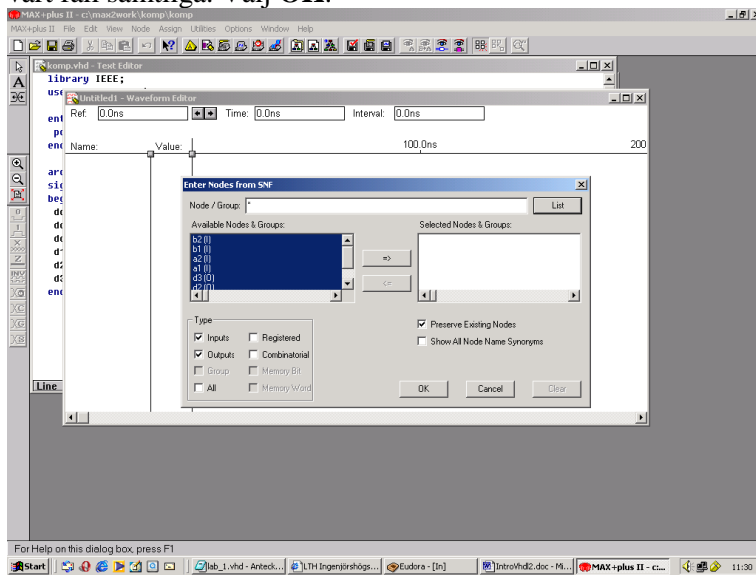


Laborationen har två uppgifter.

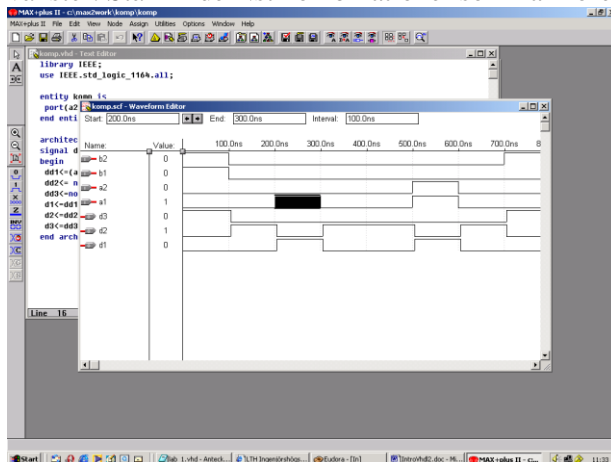
Laborationsuppgift 1:

- a) 3-ingångars **och**-grind. Adresser på in och utgångar ser du ovan. Simulera först grinden, programmera därefter kretsen. Projektets namn: **och3**.
 - b) 3 -ingångars **OR**-grind. Pojektets namn: **OR3**.
 - c) 3-ingångars **NAND** –grind. Projektets namn: **NAND3**
OBS! Du kan programmera alla tre grindarna i en fil om du vill.
1. Beskriv kretsen i VHDL. Kretsnamn: t.ex. **och3**. Studera gärna läroboken på sid 41-52. (Obs! använd paranteser, för att styra prioritetsordningen mellan operatorerna **or** och **and**.) Om du vill kan du beskriva alla 3 grindarna i samma fil.
 2. Gör i utforskaren en mapp (t.ex H:\VHDL) och en undermapp (t.ex. H:\VHDL\och3).
 3. Starta **MAX+PLUSII**.
 4. Välj **MAX+PLUSII-Text editor**.
 5. Välj **File-save as komp.vhd** i din undermapp. Skriv in din VHDL - fil . Då du sparar filen med extension *.vhd, vet editorn att det är en VHDL-fil och reserverade ord i VHDL blir färgade.
 6. Spara med **Save**.
 7. Välj **File-Project –set Project to Current File**.

- Välj **List**. Då är samtliga signaler markerade. Med \Rightarrow över du de signaler som du vill använda. I vårt fall samtliga. Välj **OK**.



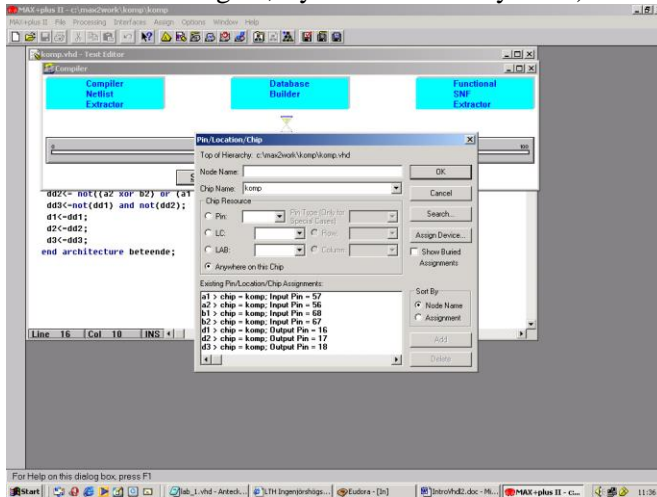
- Välj **Options-snap to Grid** och **Show Grid**, välj även **Grid Size**, t ex 10 ns
- Välj **File- End-Time** t.ex 200ns. Nås då Waveform editorn är aktiverad.
- Välj **View-Time –range- show time Range in Window From 0 ns to 200ns**.
- I tidsintervallet skall du nu ställa in de signalkombinationer, som du har fått i tabellen ovan. Du markerar en ett delintervall med musen och sedan väljer du värdet med knapparna längst till vänster. Ställ in de Nst kombinationer som kan förekomma i de N första intervallen.



- Välj **File-save As- komp.scf**.
- Stäng **Waveform Editor** och välj sedan **Compiler-Processing-Functional SNF Extractor**. Stäng **Compiler** fönstret.
- Välj **Max+PLUS – Simulator**. Välj start tid och sluttid (t. ex 200ns). Starta simuleringen och öppna sedan vågformsfönstret. Kontrollera att utsignalerna blev de förväntade. Om inte, så får du ändra i källkoden (komp.vhd) och göra en ny kompilering och simulering.

Kompilering av projektet till Altera kapsel

21. Välj **MAX+PLUS II – Compiler**. Välj **Processing-Timing SNF Extractor**. Välj **Assign – Device – EPM7128SLC84-15**. (kapseln på kortet).
22. Välj **Assign-Pin/Location /chip**. Skriv in adresser på in- och utsignaler. **Se aktuell figur!** Låt t.ex $a_2=28$, $a_1=30$, $b_2=56$, $b_1=57$, $u_3=9$, $u_2=10$, $u_1=11$. (Tips; använd **search**-knappen, sedan **list**, markera aktuell signal, tryck **add** sedan tryck **ok**)



När du är klar välj **OK** och **kompilera** därefter!.

Programmering av kapseln

OBS! De switchar, som inte är programmerade som insignaler, måste stå i sitt övre läge (vilket innebär att motsvarande ben på IC-kapseln är oanslutna).

23. Anslut flatkabeln (printerportskabeln) mellan dator och lab.kort
24. Anslut matningsspänningen till lab.kortet.
25. Välj **MAX+PLUS II-Programmer**. Kontrollera mha. JTAG-Multi Device JTAG Chain setup att rätt fil programmeras till kapseln. I vårt fall t.ex komp.pof.
26. Starta programmeringen av kapseln med **Program**.
27. Ställ DIP-switcharna i de olika kombinationer du har i din funktionstabell. Kontrollera att resultatet stämmer.
28. Visa handledaren pkt 28.

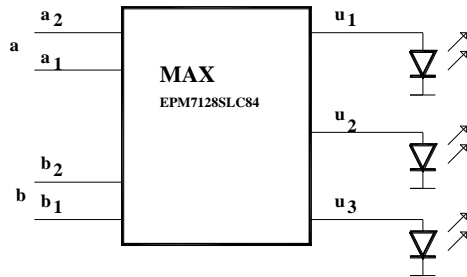
Laborationsuppgift 2, En komparator, jämförare, med vilken två binära tal $a = (a_2, a_1)$ och $b = (b_2, b_1)$ skall jämföras med avseende på relationerna likhet(=), mindre än (<) och större än (>), skall realiseras i en en PLD (Altera MAX EPM7128SLC84). Komparatorn skall ha tre

utgångar u_1, u_2, u_3 , som anslutes till lysdioder.

Om $a < b$, så skall $u_1 = 1$ och lysdioden D_1 tändas. (Övriga släckta)

Om $a = b$, så skall $u_2 = 1$ och lysdioden D_2 tändas. (Övriga släckta)

Om $a > b$, så skall $u_3 = 1$ och lysdioden D_3 tändas. (Övriga släckta)



1.

a_2	a_1	b_2	b_1	u_3	u_2	u_1
0	0	0	0			
0	0	0	1			
0	0	1	0			
0	0	1	1			
0	1	0	0			
0	1	0	1			
0	1	1	0			
0	1	1	1			
1	0	0	0			
1	0	0	1			
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1			
1	1	1	0			
1	1	1	1			

Fyll i funktionstabellen ovan.

Tag fram de förenklade logiska villkoren för u_3, u_2 , resp u_1 . Använd gärna karnaugh-diagram.

Del 2:

Laborationen skall ge

- träning i beskrivning av kombinationskretsar i VHDL, simulering och realisering i PLD
- kunskap om struktur och beteende hos fundamentala kombinationskretsar såsom *heladderare* och *adderare*
- beskrivning av ett *sekvensnät* i VHDL samt realisering i PLD
- träning i hantering av den grafiska editorn

Förberedelseuppgifter

Gör förslag till källkod till uppgift 1 och till uppgift 3 nedan.

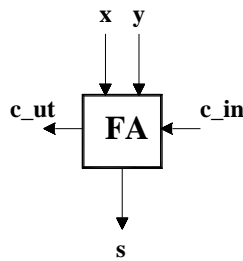
Laborationsuppgifter

Beskriv i VHDL, simulera och realisera i PLD EPM7128SLC84-7:

1. Heladderare(FA)

Se läroboken figur 4.45 och 4.46 samt exempel 9.5

Komponent och filnamn: FA



Laborationens syfte

Välj lämpliga adresser till portarna (t.ex c_in= ben 28).

Simulera och programmera kretsen .

2. Adderare för två 4-bitars tal, uppbyggd med fyra heladderare(FA)

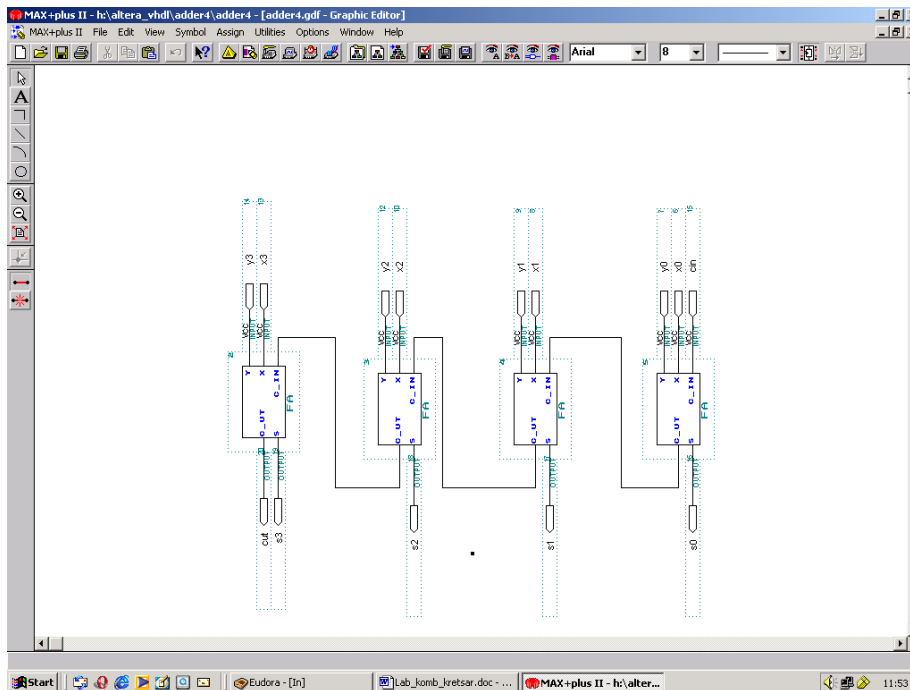
Använd den grafiska editorn.

Komponent och filnamn: Adder4.

Skapa en ny mapp (t.ex. H:\VHDL\Adder4)

1. Välj MAX+plusII- Graphic Editor.
2. Välj File –Save –As och spara som t.ex Adder4.gdf i mappen Adder4
3. Välj File-Project-Set to Current File.
4. **Utplacering av symbolen FA.**
Kopiera FA.vhd till Adder4. Kompilera filen.
Då du har kompilerat filen FA.vhd , så skapas en symbol med namnet FA.sym.
Dubbelklicka i ritarean. Då öppnas ett fönster, **Enter Symbol**. Leta upp filen FA.sym i mappen H:\VHDL\Adder4 Placera ut symbolen FA. Upprepa förfaringssättet för de övriga tre FA. Förbind symbolerna med 'ledningar'. Använd horisontal och vertikal linje ('knappen' ganska långt upp till vänster(snabbkommando F3). Om du vill redigera din symbol, använder du symboleditorn (här har signalerna kastats om för att förenkla ledningsdragningen.).
5. **Utplacering av in-och utgångar.**
Dubbelklicka i ritarean. Samma fönster som förut öppnas. Skriv **input** eller **output** som

symbolnamn. Välj OK. Då placeras en input- eller en outputsymbol på ritarean. Markera din in- eller outputsymbol (att den är markerad, kännetecknas av att symbolen får en röd 'ram'). Högerklicka och välj: Edit pin-name. (fungerar endast med horisontell symbol!). Skriv lämpligt namn.



6. När du har ditt grafiska gränssnitt klart.: Välj **File-Project-Save&Check**. Rätta eventuella fel. När detta är klart: **kompilera**.
7. Gör funktionsimulering av din 'krets'. Kontrollera att de simulerade värdena stämmer de förväntade.
8. Kompilera projektet till kapseln EPM7128SLC84-6. Välj lämpliga **stiftnummer** för in- och utsignalen.
9. Programmera kapseln och provkör. Naturligtvis ska in- och utsignaler stämma med det förväntade resultatet.
10. Visa upp resultatet för handledaren.