Linköpings universitet | Institutionen för systemteknik Laborationer i digitalteknik | Datorteknik 2016

Simulering med ModelSim

– En kort introduktion

TSEA22 Digitalteknik D



Linköpings universitet SE-581 83 Linköping 013-28 10 00, <u>www.liu.se</u>

Simulering i ModelSim

Följande dokument beskriver steg för steg hur en VHDL-modell simuleras i Modelsim. Sist i dokumentet finns några övningsexempel.

1. Kopiera katalogen "TSEA22" från "U:\da " till "H:\"

2. Starta Modelsim

Start \rightarrow All Programs \rightarrow ModelsimSE.. \rightarrow Modelsim \triangleright Close

3. Skapa ett projekt

File \rightarrow *New* \rightarrow *Project*

- Project name: "XOR_sim"
- Project Location: "H:/TSEA22/VHDL_lektion/XOR"
- ≻ ок
- Add Existing File: "H:/TSEA22/VHDL_lektion /XOR/XOR_gate.vhd"
- ≻ ОК
- Close

I fliken **Library** finns en lista på alla tillgängliga bibliotek och dess tillhörande objekt. Konstruktion finns under "work". I fliken **Project** listas alla VHDL-filer som finns i det nuvarande projektet.

4. Kompilering

 $Compile \rightarrow Compile \ all$

5. Simulering

För att simulera kretsen måste insignaler specificeras. Detta kan antingen göras med GUI:t, via transcriptfönstret eller genom att skriva in de kommandon som ska köras i ett makro, en .do-fil, som sedan anropas från transcriptfönstret. Börja med att följa stegen i guiden för GUI:t. När ni kör kommandona från GUI:t kommer motsvarande textkommandon att dyka upp i transcriptfönstret. Skapa sedan en fil sim.do där ni klipper in de kommandon ni vill köra. Run Length och run-knapp

GUI

- Library: work/xor_gate
- \blacktriangleright Höger klick \rightarrow Simulate
- Objects: markera alla signaler
- \blacktriangleright Höger klick \rightarrow Add Wave
- Wave: högerklicka på "a", välj "Clock": offset="25ns", Duty="50", Period="100ns". OK
- Wave: högerklicka på "b", välj "Force": Value="0", OK "Force" Value"1", "Delay for"="200ns", OK
- Ändra "Run Length" till 400 ns
- ➢ Klicka på "run"

ModelSim SE-64 10.1b			- 0 -×
Eile Edit View Compile Simulate Add Objects	Tools Layout Bookmarks Window Help		
🖹 • 😹 🖬 🏷 🍜 🗼 🛸 🎕 🎞 🗋 🔷 • 🗛	🗄 🛛 нер 👘 🦓 🖄	🕒 🖫 + 🛊 🐲 🗱 🚺 100 ns 🛊 🖳 🔛 🕸 🍩	i 🛍 🔟 🚸
🕇 🕀 🏌 📩 🏡 🏦 🛛 Layout Simulate	ColumnLayout Default	🗾 🤹 · 😼 - 🐷 - 🗸 📔 💿 🗤	i all 🌮
	∃ →ξ - ∰• Search:		LT J J
🖽 Project - H:/Undervisning/TSEA22/Kursbiblotek/Lektioner/Modelsimiektion/TSEA22/HDL_Jektion/XOR/XOR 😑 🖬 🛪 💊 Objects 🔤 👘 🖊 🕇 🗶			
▼Name Status △ Type	Orde Modified	Value Kind Mode	
XOR_gate.vhd VHOL	o oso4/15 ot:33:36 PM Signale	o Sgral h → b 0 Sgral h → b 0 Sgral h v U Sgral Out	
		Processes (Active)	:स व ×
		VName Type (filtered) State Or	rder Parent Path
Library × 😹 sim × 🖽 Project ×		4) ·	•
A Transcript			+ a ×
Loading project XOR worksmy via - woptargs=tack work xor_gate # vsim -woptargs=tack work xor_gate # loading std.testaintd # loading teck testaintd # loading teck testaintd # loading teck xor_gate(tl) #1 housing teck xor_gate(tl) #1 hist pack www.interior.cs.hody.pw.wf add wave -position interpoint \ aim:/xor_gate(b \ imir.xor_gate(y)			
	Project : XOR Now: 0 ns Delta: 0	sim:/xor_gate	

Do-fil

Prova nu att göra om simuleringen genom att skriva kommandona i en .do-fil. I XOR-katalogen finns en tom sim.do-fil. Öppna den:

- ▶ File → Open
- Välj Do Files (*.do)
- Välj sim.do

Skriv in följande rader kod i sim.do. Dessa svarar mot GUI-inmatningarna gjorda innan.

```
vsim xor_gate
add wave sim:/xor_gate/*
force -freeze sim:/xor_gate/a 1 25, 0 75 -r 100
force -freeze sim:/xor_gate/b 0 0, 1 200
run 400
```

Spara sim.do. Kör filen genom att i transcript-fönstret skriva > do sim.do

Om man är nöjd med simulerade variabler men vill t ex modifiera insignalerna så behöver inte simuleringen startas om helt. Då räcker det med att använda restart-kommandot enligt följande kod:

```
# vsim xor_gate
# add wave sim:/xor_gate/*
restart --force
force -freeze sim:/xor_gate/a 1 25, 0 75 -r 100
force -freeze sim:/xor_gate/b 0 0, 1 200
run 400
```

Här har rader i koden kommenterats bort med #. Prova att ändra insignalerna genom att modifiera dofilen och simulera xor-grinden med de nya insignalerna.

6. Visa resultat

Resultatet av en simulering visas i Figur 1. Med vänster musknapp flyttas markören, en gul vertikal linje. Värdena vid markerad tidpunkt visas till höger om signalnamnen. Figur 2 visar vågfönstrets zoomfunktionalitet.



Figur 1. Resultatet av simuleringen.



Figur 2. Wave-fönstrets zoom-funktionalitet. Klicka på musens scrollhjul och rör musen i riktning längs pilarna för att få motsvarande funktion.

Mäta tid

För att mäta tid mellan två tidpunkter behöver man lägga till en markör. Detta görs på följande sätt.

➤ Lägga till markör: Add → Cursor

Dividers

I variabellistan kan det vara trevligt att dela in variabler i t ex in- och ut-gångar. Det finns möjlighet att lägga till så kallade dividers, en slags rubrikrader i signallistan, på följande sätt.

- ➢ Högerklicka på signalen "a" → Add →New Divider Divider Name="Ingångar"
- → Högerklicka på signalen "y" → Add → New Divider

Divider Name="Utgångar"

Byta färg för logiska nivåer

I Figur 1är hög signalnivå indikerad med guldfärg. Detta erhålls genom att göra följande inställning: → ModelSim fönstret: Tools→Edit Preferences.. → Wave Windows → LOGIC_1 → Palette → Gold



7. Addera en signal i VHDL filen

Antag att ni vill skapa en inverterad utsignal till XOR-grinden, dvs z = not y. Börja med att beskriva den nya utsignalen i vhdl-filen XOR_gate.vhd. Detta kan göras i modelsim genom följande steg

- Gå till Modelsims huvudfönster och fliken Project
- ➢ Höger klicka på XOR_gate.vhd → Edit
- Lägg till signalen "z = not y"

Filen kompileras genom att klicka på
 Project → Compile→Compile Selected

Vid kompilering kan det hända att det dyker upp röda felmeddelanden i transcript-fönstret. Dessa går att dubbelklicka på så ges mer information om vad som gått fel. Om koden kompilerat felfritt står det med grön text i transcript-fönstret att "Compile of XOR_gate.vhd was successful". Då är det dags att simulera

kretsen

do sim.do.

Det går att inkludera kompileringssteget i do-filen genom att lägga till raden vcom XOR_gate.vhd först. Exempel på kod:

vcom XOR_gate.vhd restart -force force -freeze sim:/xor_gate/a 1 25, 0 75 -r 100 force -freeze sim:/xor_gate/b 0 0, 1 200 run 400

8. Kombinera signaler i vågfönstret.

Antag att räknetillståndet på en räknare simulerats fram med 4 tillståndsvariabler. I detta fall vore det trevligare om det i vågfönstret istället för de fyra binära signalerna visade motsvarande hexadecimala siffra. Detta är möjligt. Låt oss göra stegen som krävs för att kombinera utsignalerna för XOR-grinden.

- > Markera de signaler i vågfönstret som skall slås ihop till en vektor i detta fall signal y och z.
- ▶ Höger klick på signal \rightarrow Combine Signals ...
 - Fyll i vektornamn, Result Name="u", OK

Nu ska den nya vektorsignalen dyka upp i vågfönstret och värdena indikeras med binära tvåbitsord. För att visa värdet på vektorn som positiva heltal gör:

Wave fönstret: Högerklicka på signalnamnet \rightarrow Radix \rightarrow Unsigned

9. Övriga funktionalitet som kan var bra att känna till

Byta färg på en signal Wave fönstret: *Höger klick på signal* \rightarrow *Properties* \rightarrow *Wave Color* **Spara vågfönster från en simulering** Wave fönstret: *File* \rightarrow *Export* \rightarrow *Image..* **Uppgift 1.** Konstruera med hjälp av VHDL en BCD-räknare som har samma funktion som 74LS160 och simulera konstruktionen i Modelsim. I den kopierade katalogen finns en underkatalog med namn Counter. I den katalogen finns Counter.vhd och sim.do som kan användas för att skapa räknaren och simuleringen. Använd funktioner som finns i VHDL, tex "a <= a + 1" (minimering mha Karnaughdiagram behöver INTE göras). Simulera det beteende som finns uppritat i databladet för räknaren och som exemplifieras nedan.

Insignaler: CLR, LOAD, A, B, C, D, CLK, EN_P, EN_T Utsignaler: Q3, Q2, Q1, Q0, RCO



Uppgift 2. Simulera VHDL-filen till kodlåset från lab 2 (uppgift 10). Nedan visas ett förslag på hur simuleringsresultatet kan se ut. I simuleringen har en asynkron clear-signal lagts till som nollställer kretsen vid 25 ns. Värdena på q beror förstås på vald tillståndskodning. Värdena på *x* ändras asynkront vid *n**100 + 10 ns där *n* är ett heltal. Den synkroniserade insignalen är x sync. Utsignalen betecknas *u*.



Uppgift 3. Gör förberedelseuppgiften till laboration 4. Se sida 2 och 3 i labkompendiet.