VHDL-Synthese

Werkzeuge : SYNOPSYS Design-Vision Design-Kits : AMS Hit-Kit designSetup : syn ams



Diese Anleitung beschreibt die Synthese mit den SYNOPSYS Werkzeugen: Wegen der vielfältigen Möglichkeiten in den Syntheseprozess einzugreifen, können hier nur die einfachsten Einstellungen vorgestellt werden. Genauere Informationen finden sich in der SYNOPSYS Online-Dokumentation unter: "Design Compiler"

Design-Flow

Voraussetzung für die Synthese ist eine korrekt simulierte (hierarchische) VHDL-Beschreibung. Die Kodierung von *synthesegerechtem VHDL* ist in der "VHDL Kurzbeschreibung" und in dem SYNOPSYS "HDL Compiler for VHDL Reference Manual" beschrieben.

Der Syntheseprozess lässt sich in folgende Schritte unterteilen:

- 1. Einlesen der VHDL-Quelldatei(en)
- 2. Behandlung der Hierarchie
- 3. Randbedingungen für den Syntheseprozess festlegen
- 4. Synthese der Schaltung und Analyse der Ergebnisse
- 5. Ausgabedateien für Simulation und Back-End Programme schreiben

Arbeitsschritte

VHDL Einlesen

1. (meist schon vor der Simulation ldv) Initialisierung der Shell:

maeder on t	ams85: /hom	e/tams_1/maeder/demo - Befe	hisfenster - Konsole	\odot \bigcirc	×
Sitzung Bea	rbeiten Ansich	t Lesezeichen Einstellungen Hilf	e		
[maeder@tam	s85]~⁄demo> <mark>so</mark>	urce ~maeder/design.Setup			
design.Se ⁻	ես թ	Andreas Maeder	2006.01		
SYNOPSYS	[syn]	Synthesis, Verification	v2005.09 , v2005.12		
	LSIMJ	Simulation tools		• • •	
	[]au]	Smartmodel / hspice			
	LIAYJ		02003.03, 02003.03	•••	
CADENCE	[ic]	IC-Design Framework	ic5.1.41		
	[]dv]	Simulation, Synth., Verif.	ius5.5		
	[soc]	SoC Encounter: Synthese+P&R	soc4.1		
	[pr]	Placement & Routing	se5.4		
	[pcb]	PCB-Design	spb15.2		
Design-Ki	ts	SYNOPSYS / CADENCE			
	LamsJ	+AMS HITKIT	03.70		
FPGAs	[alt]	QuartusII, NIOS	υ5.0, υ3.2		
	[xil]	Alliance	v8.1.01i		
input:	[info] [none] ams syn ldv	-information about the tools -reset all paths to original	values		
AMS	Hit-Kit		-0ersion		
Summersus	Sunthesis	->sumbeln	u2005.09-SP2		
	Tetramax	, og mo 1p	u2005.09-SP2		
	Formalitu		v2005.12		
	PrimePower		v2005.12		
	PrimeTime		v2005.12		
	Leda		v4.2.0		
SystemC			v1.0.2		:::
Cadence	logic design	/verific. IUS ->ldvhelp	v5.5		
	synth	->rchelp	v4.2USR4		
	Conformal	->cnthelp	A2.102R1		
[maeder@tam	s85]~⁄demo>				
🔏 💻 Befeh	Ilsfenster				

2. Start des Systems:

maeder on tams85: /home/tams_1/maeder/demo - Befehlsfe	nster - Konsole 🛛 🕞 🚳					
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe						
[maeder@tams85]~/demo> <mark>ams_synopsys</mark>						
DEFINE YOUR TARGET PROCESS: <input/>	erster Start					
Imput Finis-Frocess I um Finet						
 target process [c35]: Note: Creating .AMSProcDat Note: Creating .synopsys_dc.setup Note: Creating hdl.var Note: Creating cds.lib Note: Creating pearl.cmd Note: Creating netlist.sdf.cmd Note: Creating work Note: target process is c35 at 3.3V [c35_3.3V] Synopsys program is design_vision-xg 	bestätigen					
DC Professional (TM) DC Expert (TM) DC Ultra (TM) FloorPlan Manager (TM) HDL Compiler (TM) UHDL Compiler (TM) Library Compiler (TM) DesignWare Developer (TM) DFT Compiler (TM) BSD Compiler Power Compiler (TM)						
Version X-2005.09-SP2 for suse32 Jan 03, 2006 Copyright (c) 1988-2006 by Synopsys, Inc. ALL RIGHTS RESERVED						
This software and the associated documentation are confidential and proprietary to Synopsys, Inc. Your use or disclosure of this software is subject to the terms and conditions of a written license agreement between you, or your company, and Synopsys, Inc.						
The above trademark notice does not imply that you are lig all of the listed products. You are licensed to use only for which you have lawfully obtained a valid license key. Initializing	censed to use those products					
Please use: set_fix_multiple_port_nets -all [get_designs] design_vision-xg-t> design_vision-xg-t> ■						
A Befehlsfenster	íí x					

Der Befehl startet ein Skript, dass beim ersten Aufruf nach dem AMS-Prozess fragt und die passenden Initialisierungsdateien für die Synthese und nachfolgende Schritte erzeugt. Für die $0,35 \,\mu m$ Bibliotheken ist die Voreinstellung c35 zu bestätigen.

 Design Vision - TopLevel.1 - [Console.1]

 File Edit View Select List Hierarchy Design Schematic Attributes Timing Test Window Help
 🖆 🖩 🔮 📓 🔍 🔍 🐂 -- IX **₽**_Hier.1 Cells (Hierarchical) Logical Hiera Cell Name Ref Name Cell Path Dor -all [get designs] 'lease use: set_fix_multiple_port_nets
lesign_vision-xg-t> gui_start Log History Errors/Warnings design_vision-xg-t> Selected:

Anschließend starten die Synthesewerkzeuge, wobei in der Shell die Kommandozeilenversion dc_shell läuft, die dann die grafische Benutzeroberfläche startet:

Tipp: Das Kommandozeileninterface (mit TCL-Syntax) der GUI oder der Synthese-Shell wird meist im Batch-Betrieb, beispielsweise bei großen Entwürfen, eingesetzt. In der Datei command.log sind die eingegebenen Befehle mitprotokolliert. Für die Erstellung einer Batch-Datei kann man sich diese Datei ansehen und entsprechend modifizieren.

3. Einlesen der Quelldateien

Bei der Aufbereitung der Daten für die Synthese werden zwei Schritte unterschieden: Bei der *Analyse* wird der VHDL-Code auf die Synthetisierbarkeit untersucht und es werden systeminterne Eingabedateien – in Form von Templates – erzeugt. Die anschließende *Elaboration* generiert dann daraus die Datenstrukturen der Synthese.

Beide Schritte können auch gemeinsam ausgeführt werden. Ob eine Trennung von Analyse und Elaboration notwendig ist, hängt von der Art der VHDL-Beschreibung und der bisherigen Vorgehensweise ab. Sie zwingend notwendig wenn:

- zu einer Entity mehrere Architekturen existieren und man explizit eine auswählen möchte Ansonsten gilt die zeitliche Reihenfolge bei der Codeanalyse.
- eine Entity generic-Parameter besitzt an die Werte übergeben werden sollen.

Read — Analyse und Elaboration in einem Schritt

In dem einfachen Fall können die Entities der Hierarchie direkt eingelesen werden:



Die Abarbeitungsreihenfolge der einzelnen Dateien ist beliebig, es muss nur gewährleistet sein, dass vor der Synthese alle Quelldateien in der SYNOPSYS Datenbasis vorhanden sind:

nanden sind:	
Read Designs	\odot \odot \otimes
Look in: 🔄/home/tams_1/maeder/demo/	£ 💣 🏢 🇰
work	
l opw.vhd	
opwTst.vhd	
File <u>n</u> ame: "alu.vhd" "opw.vhd"	<u>O</u> pen
File type: Database Files (*.ddc *.ddc.gz *.db *.db.gz *.gdb *.sdb *.pdb *.edif *.eqn *	Cancel
Eormat: Auto	SYNOPSYS'

Analyse und Elaboration getrennt Besitzen VHDL-Entities generic-Parameter an die Werte zu übergeben sind oder stehen mehrere Architekturen zur Auswahl, dann dann müssen Analyse und Elaboration getrennt ausgeführt werden. Die Analyse ist für alle Dateien der Hierarchie (in beliebiger Reihenfolge) durchzuführen:



Nach Abschluss der Analysephase wird die Hierarchie, ausgehend von dem gewählten Entwurf, durch Elaboration abgearbeitet. Architekturen werden über ihren Bezeichner, nach folgendem Schema (*entity*) ((*architecture*)), unterschieden. Das Beispiel zeigt auch, wie generic-Parameter spezifiziert werden:



4. Kontrolle der Daten

Ist der VHDL-Code syntaktisch korrekt und *synthetisierbar*, dann werden Objekte für die Synthese erzeugt, andernfalls enthält die Log-Datei entsprechende Fehlermeldungen.

In dieser Datei wird unter anderem ausgegeben, welche speichernden Elemente (Flipflops/Latches) und Tristate-Treiber durch welche Zeilen des VHDL-Codes impliziert werden. Die Ausgabe sieht dabei folgendermaßen aus:

Eile Edit View Select List Hierarchy Desi	. 1] aiu .gn S <u>c</u> hematic <u>A</u> ttributes <u>T</u> iming Test <u>}</u>	<u>M</u> indow Hel <u>p</u>	
🖆 🖬 🍯 📔 🔍 🍳 🌂 📃] € € € > € ▼ ▲ D ≣	🐵 🎇 👬 🏥 👔 👔 📔 🔳	
Hier.1			
Cells (Hierarchical)			
Toferred THREE-STATE control device	s in process		
in routine opw line 45 in file - (targews from 1	(mandar (dana (ann mhd)		
	/madeder/demo/opw.vnd .		
Inree-state Device Name	1ype	======	
ioBus_tri3[15] ioBus_tri3[14]	Three-state Buffer Three-state Buffer	N N	
ioBus_tri3[13]	Three-state Buffer		
ioBus tri3[11]	Infee-state Buffer	N N	
ioBus_tri3[10]	Three-state Buffer	IN I	
10Bus_tri3[9]	Three-state Buffer Three-state Buffer	N N	
ioBus_tri3[7]	Three-state Buffer	N	
ioBus_tri3[6]	Three-state Buffer		
ioBus_tri3[4]	Three-state Buffer	N	
ioBus_tri3[3]	Three-state Buffer		
10Bus_tr13[2] 1 ioBus_tr13[1]	Inree-state Buffer Three-state Buffer	IN I IN I	
ioBus_tri3[0]	Three-state Buffer	IN I	
		=====	
Inferred memory devices in process	'regP'		
in routine opw line 69 in f //tamsNFS/home/tams 1/maed	ile er/demo/onw.whd:		
oReg_reg Flip-flop reg0 reg Flip-flop	16 Y N Y N N 16 Y N N N N	N N N N	
reg1_reg Flip-flop	16 Y N N N N	N N	
reg2_reg Flip-flop		N N N N	
reg3_reg Flip-flop	16 Y N N N N	N N N	
reg5_reg Flip-flop	16 Y N N N N	N N	
rego_reg Flip-Flop reg7 reg Flip-flop	16 1 N N N N 16 Y N N N N	N N N N	
Current design is now '/tamsNFS/hom	e/tams_1/maeder/demo/alu.db:alu'		
Loaded 2 designs. Current design is 'alu'.			-
Log History Errors/Warnings			
design_vision-xq-t>			
Ready			Selected:

Dabei ist darauf zu achten, dass diese Elemente auch wirklich vom Designer so vorgesehen waren und nicht die Folge einer *ungeschickten* VHDL-Beschreibung sind. Solche möglichen Fehlerquellen können sein:

- Signale oder Variable vom Typ integer haben keine Wertebereichseinschränkung bei der Deklaration erhalten. Entsprechend dem Datentyp werden 32-bit breite Register erzeugt.
- Obwohl nur das Verhalten eines Schaltnetzes beschrieben werden soll, wurden Latches für Signale eingefügt. In diesem Fall werden Signalzuweisungen im VHDL-Code von Bedingungen abhängig gemacht. Dann müssen entweder in allen möglichen Verzweigungen Signalzuweisungen vorkommen oder eine *Default-Zuweisung* muss im sequentiellen Prozess *vor* der Verzweigung stehen.

Die VHDL-Beschreibung ist dann abzuändern damit nicht unnötige Hardware generiert wird — im Falle von Latches wird meist auch die Funktion der Schaltung fehlerhaft!

Behandlung der Hierarchie

- 5. (optional) Kontrolle der Hierarchie
 - Nach Auswahl eines (Teil-) Entwurfs wird dessen Hierarchie in dem Hierarchiebrowser des Synthesewerkzeugs dargestellt:



Neben der grafischen Darstellung der Hierarchie erlaubt dieses Werkzeug auch die weitere Handhabung der Instanzen, die Festlegung von Attributen für die Synthese etc. Die rechten Maustaste öffnet dazu ein kontextsensitives Menü:



Achtung: Die folgenden Schritte sind nur notwendig, wenn innerhalb der Hierarchie Entities mehrfach instanziiert werden, andernfalls kann direkt mit der Eingabe der Syntheserandbedingungen ab Punkt 7 (Seite 12) begonnen werden!

Normalerweise wird die Hierarchie während der Synthese automatisch traversiert und alle instanziierten Entities bearbeitet. Für Teile der Hierarchie die mehrfach instanziiert werden, gibt es die zwei Vorgehensweisen: unterschiedliche und gleichartige Behandlung der einzelnen Instanzen.

6. (Instanzen mehrmals vorhanden) Unterschiedliche Behandlung / ein Syntheselauf Diese Strategie ist anzuwenden, wenn Instanzen in *unterschiedlicher Weise* mehrfach benutzt werden, beispielsweise durch andere Generics, nicht benutzte Ausgänge oder konstante Eingangsbelegungen. Beispiele: konfigurierbare FIFOs mit unterschiedlicher Wortlänge und -Breite (Generics), die Instanzen eines Multiplizierer erhalten aus dem übergeordneten Design eines digitalen Filters jeweils einen konstanten Faktor (konstante Input-Ports).

Ausgehend von der top-level Entity wird die Hierarchie durchlaufen und mehrfach vorhandene Elemente werden unterscheidbar gemacht, um sie bei der später folgenden Synthese individuell bearbeiten zu können:



6. (Instanzen mehrmals vorhanden) Gleichartige Behandlung / mehrere Syntheseschritte Werden mehrfach referenzierte Entities in der Hierarchie in immer *gleicher Weise* benutzt, so sollten sie (aus Effizienzgründen) nur einmal synthetisiert werden. Beispiele: identische Recheneinheiten eines systolischen Arrays, Multipliziererinstanzen eines programmierbaren Filter (Faktoren frei wählbar).

Tipp: Bei sehr großen (Teil-) Entwürfen ist es auch sinnvoll die Synthese in kleinere "Portionen" zu unterteilen, um die Programmlaufzeit und den Speicherbedarf geringer zu halten. In diesen Fällen wird auch die folgende Strategie der getrennten Synthese einzelner Teile eingesetzt.

Dazu werden zuerst diejenigen Teile der Synthesehierarchie, die bei einem top-down Vorgehen nicht berücksichtigt werden sollen, mit dem Attribut Don't touch gekennzeichnet. Dies kann sowohl für Instanzen – hier immer "Cell" genannt – als auch für Entities erfolgen. Die hier skizzierte Vorgehensweise behandelt einzelne Instanzen:



Anschließend wird eine top-down Synthese, wie ab Seite 12, beschrieben duchgeführt:

- 7. Top-level Entwurf Auswählen
- 8. Taktfrequenzen festlegen
- 9.–11. Synthesevorgaben machen
 - 12. Operationsbedingungen einstellen
 - 13. Synthese der Gatternetzliste

Die Randbedingungen des top-level Designs (Taktrate, Timing, Flächenvorgaben...) werden danach auf noch nicht synthetisierte Teilentwürfe propagiert:



Dann müssen diese Teilentwürfe, wie ab Punkt 13, Seite 20 beschrieben, synthetisiert werden. Nachdem *alle* Teile verarbeitet wurden, können die Ergebnisse ausgewertet (Seite 21) und die Daten ausgegeben werden (Seite 27).

Synthesebedingungen festlegen

Bei der Realisierung einer Schaltung durch eine Gattenetzliste, gibt es nicht nur eine, sondern beliebig viele Lösungen. Dieser *Suchraum* wird während des Syntheseprozesses nach einer "möglichst guten" Realisierung (bezogen auf eine Bewertungsfunktion) hin untersucht. Die unterschiedlichen Realisierungen unterscheiden sich hinsichtlich ihres Flächenbedarfs und den Verzögerungszeiten (Geschwindigkeit). Im Allgemeinen sind kleine Lösungen langsam (viele gemeinsame logische Teilausdrücke \Rightarrow große sequentielle Tiefe), während sehr schnelle Realisierungen sehr groß werden.

Wegen der Möglichkeiten den Syntheseprozess zu beeinflussen, sei hier nochmals auf die SYNOPSYS Dokumentation verwiesen. Im folgenden werden drei "einfache" Möglichkeiten vorgestellt, die auch miteinander beliebig kombiniert werden können. **Tipp:** für "optimale" Syntheseergebnisse ist es besser mit realistischen Werten für Fläche bzw. Geschwindigkeit zu synthetisieren und diese Randbedingungen über mehrere Syntheseläufe zu verschärfen.

7. Auswahl des top-level Entwurfs im Menü der GUI, bzw. im Hierarchiebrowser. Alle weiteren Befehle beziehen sich darauf:



Sollen später Taktfrequenzen und Zeitbedingungen festgelegt werden, dann sollte man ein Symbol und ein Schematic erzeugen, um dort durch Selektion mit der Maus Signale und Ports auszuwählen.

Sy	mbol erzeu	gen							
De	sign Vision - TopLeve	el.1 - [Symb	ol.1 opw] (pw					\odot \odot \otimes
<u>E</u> ile	e <u>E</u> dit <u>V</u> iew <u>S</u> elect List	Hierarchy [esign S <u>c</u> hematic	Attributes Tin	ning Test <u>W</u>	indow Help			
	▼ 🖬 😅 🛛 🛄 🔍 🦄 P∈ Hier 1	4 ()					× <u> </u>	_	
 ⊕	Logical Hierarchy	Cells (Hierar	chical)			reate Symbol View			
a,	Ė- <mark>i©</mark> ==> opw	Cell Name	Ref Name	Cell Path					
O	12/2001	(G) alul	alu	alul					
					Symbol.1	opw			
						⊠rst			
						Brenio			
						210910			
						🔁 outlo			
						[0:8]ago			
						🗅 clk ioBus[15:0]🖸			
						🔁 bSel[2:0]			
						L≥ aluio			
						🔁 aSel[2:0]			
						51-1-1			
						La albu	V		
		•		Þ			×		
	ရာ 🔁 🔁 Symbol.1 opw	J							
×	design vision-xq-t> :	cne uesign :cmenus::cre	aru . (HDE-19 sate schematic	.,					
	Loading db file '/loc Information: Generati	al/tams1.1/: ng schemati	synopsys/2005.0 : for design: o	9/libraries/: pw	syn/generic	. sdb '			
	Information: Generati	ng schemati	for design: o	nw					
	design vision-xg-t) o	ui zoom -wir	dow (mui met c	r" urrent windo	w -viewl -f	11			-
	Log History Errors/Warn	ings	laow [gar_gee_e	arrene_windo	* *icwj i				
-	design_vision-xg-t>								
Cre	ate Symbol View of current	tly selected ce	lls					Design: opw	

Schematic erzeugen

Das Schematic der Schaltung enthält *vor* der Synthese nur künstliche Elemente einer internen Bibliothek. Erst *nach* dem Syntheseprozess ist die Netzliste des Schematic aus den Zellen der Gatterbibliothek aufgebaut, wie in dem Beispiel:



8. Taktfrequenz(en) festlegen

In der Regel enthalten die Schaltungen Taktleitungen, deren Taktschema (Frequenz, Phasenlage zueinander) man unbedingt angeben sollte. Sind externe Taktfrequenzen der Schaltung nicht explizit vorgegeben, dann ist eine geschätzte/gewünschte Arbeits-frequenz anzugeben.

Randbedingungen für das Taktschema stellen die einfachste und sicherste Möglichkeit dar, um Zeitbedingungen in der Synthese zu definieren. Die Taktperiode wird in *ns* angegeben. Bei der Optimierung wird der Pfad zwischen, bzw. vor, Registern berücksichtigt, so dass die explizite Vorgabe von Zeitpfaden (s.u.) überflüssig wird.

D	esign Vision - TopLeve	el.1 - [Symb	ol.1 opw] o	pw						
Ei	le <u>E</u> dit <u>V</u> iew <u>S</u> elect L <u>i</u> st	<u>H</u> ierarchy <u>D</u>	esign S <u>c</u> hematic	<u>Attributes</u> Tir	ming Test <u>W</u> i	ndow Hel <u>p</u>				
	≩∎⊜ [6] (4. 4.	Q	<u> </u>			<mark>: 11 88 88 18 1</mark> 1	opw 🛛	<u> </u>		
4	₽ <mark>e</mark> Hier.1			<u>O</u> perating	Environment	•				
•	Logical Hierarchy	Cells (Hierard	chical)	Optimizati	ion <u>C</u> onstraints	•				
	Ė-₲==> opw	Cell Name	Ref Name	Optimizati	ion <u>D</u> irectives	•				
	(-1 <u>G</u>) alul	© alul	alu	alul						
ľ										
					Symbol.1	opw				
								<u> </u>		
						≥rst				
						regio				
						Jourio				
						opc[3:0]				
						🛛 clk ioBus[15:0]	¢			
						🔁 bSel[2:0]				
						🔁 alulo				
						aSel[2:0]				
						aLou				
				•	1					
	D E Sumhal 1 anu				<u>.</u>					
	E Symbol Opw									
×	design vision-xg-t> :	cmenus::cre	ate schematic	,						_
	Loading db file '/loc	al/tams1.1/s	ynopsys/2005.09	/libraries/	syn/generic.	sdb '				
	Information: Generati	ng schematic	for design: op	W						
	Information: Generati	ng schematic	for design: op	W						
	design vision-xa-t> a	ui zoom -win	dow (qui get cu	rrent windo	w -view] -fu	111				Ţ
	Log History Errors/Warn	inas l								
	acts alack and act alach at	ributos							Deut all	
l Cr	eale clock and set clock att	nuutes							POIT: CIK	

Dazu sind in dem Symbol des top-level Entwurfs die Taktleitungen zu selektieren:

Daran anschließend werden die Takte definiert:

Specity Сюск				<u> </u>
<u>C</u> lock name: <mark>Clk</mark>	:50MHz			
Port name:				
🗖 <u>R</u> emove cloci	k			
Clock creation				
Perio <u>d</u> : 20				
Edge	Value		Add <u>e</u> dge pair	
Rising Falling		10 20	Rem <u>o</u> ve edge pa	air
			Invert <u>w</u> ave form	n
0.00		10.00		20.
🗖 Don't touch n	etwork	Г	Fix <u>h</u> old	
	ОК	Car	ncel <u>A</u> pply	

9. (optional) Flächenvorgaben — werden in der Regel nicht weiter benötigt, da als Voreinstellung möglichst kleine, kompakte Netzlisten synthetisiert werden:

De	sign Visio	n - TopLe	vel.1 - [Hier	r.1] opw			
Eile	<u>E</u> dit <u>V</u> iew	<u>S</u> elect L	<u>i</u> st <u>H</u> ierarchy	<u>D</u> esign S <u>c</u> hematic	<u>Attributes</u> Timing Test Wi	ndow Help	
🖻	: 🛛 🍮 🛛 (₽∥@,@	× •		Specify Clock	🖁 👯 🔡 🏜 🚠 😭 📗 opw 🔽	
	R Hier.1				Operating Environment	•	
e e	Logical Hie	rarchy	Cells (Hier	archical)	Optimization Constraints	Pauline Constraints	
	Ē- Ē}==> 0	pw	Cell Name	Ref Name	Optimization <u>D</u> irectives	Design Constraints	
	i;[G) al	ul	:© alul	alu	alul		
						Derive Constraints	
					🚦 Symbol.1	opw	
						🔁 rst	
						Dereglo	

Design Constraints	
Current design: opw	
- Optimization constrain	nts
	Constraint value: Unit:
Max ar <u>e</u> a :	
Max <u>d</u> ynamic power:	
Max <u>l</u> eakage power:	
Max t <u>o</u> tal power:	
Design rules	
Max <u>f</u> anout:	
Max <u>t</u> ransition:	
ОК	Cancel <u>A</u> pply

☐ Reset path OK

Cancel

<u>Apply</u>

10. (optional) Timingvorgaben — werden in der Regel nicht benötigt, da das Zeitverhalten über die Taktung (s.o.) definiert ist. Das Timing kann zwischen beliebigen Stellen der Netzliste explizit angegeben werden; dabei sind sowohl minimale als auch maximale Laufzeiten möglich. Die Anfangs- oder Endpunkte von Pfaden sollten vorher im Schematic, bzw. Symbol selektiert werden:



11. (optional) Syntheseattribute — sind sinnvoll voreingestellt und sollten nur in Ausnahmefällen (siehe SYNOPSYS Dokumentation) geändert werden:

	Attributes Timing Test Window, Help	
] ⋵ : : : : : : : : : : : : : : : : : : :	Specify Clock 🗧 🧯 🚮 ជីវិទី 🚛 👘 🗍 opw 💌	
Terrer Logical Hierarchy Cells (Hierarchical)	Operating Environment Optimization Constraints	
Cell Name Ref Name	Optimization Encouver 1	
🕐 🕞 alul alu '	alul Input Port	
	Output Port	
	<u>Pin</u>	
	Symbol.1 op Timing Boths	
	Zinnig ratio ⊨	
	Deregio	
	Doutlo	
	□ opc[3:0]	
Design Attributes		
Current design: opw		
	on't touc <u>h</u>	
Roundary ontimization	sable wired logic(ECL)	
Convential elements	sable <u>wired logic(ECE)</u>	
l Po <u>r</u> tis pad lest <u>s</u>	scan style: Multiplexed Flip Flop	
Elip-Flop	atch Design Pad	
Elatten logic		
- I ložicil ložici		
Indiana affend		
Flatten ellon:		
Fratten enon:	im 🔿 High	
Flatten enor:	um O High	
Flatten eiron: C Low C Mediut Flatten minimize: C Single output C Multigl	um O High gle output O Non <u>e</u>	
Flatten eiron: C Low C Mediu Flatten minimize: C Single output Flatten phase: C Don't continue	um O High gle output O Non <u>e</u>	
Flatten eiror: C Low C Mediu Flatten minimize: C Single output C Multipl Flatten phase: C Don't apply C Apply	um O High Die output O Non <u>e</u>	
Flatten eiron: C Low C Mediu Flatten minimize: C Single output C Multigl Flatten phase: C Don't apply C Apply	um O High gle output O Non <u>e</u> / strategy	
Flatten einn:	um C High gle output C Non <u>e</u> / strategy	
	Im C High Je output C Nong / strategy ng C Apply boolean optimization	
	um O High gle output O Non <u>e</u> • strategy ing O Apply boolean optimization	
	Im C High gle output C Nong / strategy ing Apply boolean optimization OK Cancel Apply	
Flatten enon: Low Low Flatten minimize: Single output Multigi Flatten phase: O Don't apply O Apply Apply Apply timing driven structuring	Im C High de output C Nong r strategy ng C Apply boolean optimigation OK Cancel Apply	

12. (optional) Operationsbedingungen einstellen

Für die später folgenden Schritte (Synthese und Timinganalyse) können die Zeitmodelle der Gatter festgelegt werden:

De	sign Vision - TopLeve	el.1 - [Hier	.1] opw					
<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>S</u> elect L <u>i</u> st	<u>H</u> ierarchy	<u>D</u> esign S <u>c</u> hematic	Attributes Timing Test Windo	w Hel <u>p</u>			
🖻	e e 🖉 🖻 🖉	× 🗌		Specify Clock	1. 🕄 🕂 🖁 🕂 🐉	epw	•	
	₽ _E Hier.1			Operating Environment	land Datas			
e e	Logical Hierarchy	Cells (Hiera	archical)	Optimization <u>C</u> onstraints	Input Delay			
	Ė- <mark>©</mark> ==> opw	Cell Name	Ref Name	Optimization Directives	Output Delay			
	⊡_[G) alul	:G) alul	alu	alul	Drive Strength			
					Load			
					<u>C</u> haracterize			
					Ogerating Condit	ions		
					<u>W</u> ire Load			
				C Sumbol 1 o	<u>⊤</u> iming Range			
				Symbol.1 0	h.			
					Pirst			
					Preglo			
					Doutlo			
					T			
					opc[3:0]			
		1						
		1			Clk ioBus[15:	:0]63		
		1						
	I I			1				

Neben den Ober- und Untergrenzen der Verzögerungszeiten, wird bei Standardzellen meist auch noch nach Temperaturbereichen (Standard, INDustrial und MILitary) unterschieden:



Synthese der Gatternetzliste

13. Hardwaresynthese und Abbildung auf die Zellbibliothek

Ausgehend von dem aktuellen (Teil-) Entwurf, durchläuft die Synthese die gesamte Hierarchie — die Ausnahme bilden *Don't touch*-Attribute, siehe "Behandlung der Hierarchie", ab Seite 9.

Für die Synthese stehen zwei verschiedene Programmmodi zur Verfügung:



... und hier die "neue" Engine:



Design Vision - TopLevel.1 - [Console.1] opw								
Eile Edit View Select List Hierarchy Design Schematic Attributes Timing Test Window Help								
☆ 🖬 Š 🖆 4, 4, 3, 🔽 🔄 4, 4, 4 → 4 → 5 → 5 → 5 → 5 → 5 → 5 → 5 → 5 →								
Cells (Hierarchy Cells (Hierarchical)								
Cell Name Ref Name Cell P								
Beginning Mapping Optimizations (Medium effort)								
ELAPSED WORST NEG TOTAL NEG DESIGN								
0.00.13 120220.0 0.00 0.0 0.0								
Beginning Delay Optimization Phase								
ELAPSED WORST NEG TOTAL NEG DESIGN								
Beginning Area-Recovery Phase (cleanup)								
ELAPSED WORST NEG TOTAL NEG DESIGN								
TIME AREA SLACK SLACK RULE COST ENDPOINT								
Aptimization Complete								
Log History Errors/Warnings								
Ready	Selected:							

Die einzelnen Schritte des Syntheseprozesses werden in der Log-Datei mitprotokolliert:

14. Bewertung der Syntheseergebnisse

Entspricht das Ergebnis nicht den Anforderungen, so müssen die Randbedingungen der Synthese (ab Punkt 7, Seite 12) entsprechend angepasst und ein neuer Syntheselauf (Punkt 13, Seite 20) gestartet werden.

Hier werden nur einige der umfangreichen Analysemöglichkeiten des Synthesewerkzeugs vorgestellt. Die bei der Timinganalyse ausgegebene Information bezieht sich immer auf das unter Punkt 12, Seite 19 festgelegte Zeitmodell. Durch Auswahl anderer Operationsbedingungen können "worst-case" und "best-case" Timing der synthetisierten Struktur ermittelt werden.

Kontrolle des Schematic und der Hierarchie

Wie schon zuvor auf Seite 14 gezeigt, kann ein Schematic der synthetisierten Netzliste erzeugt, angesehen und hierarchisch durchlaufen werden.

Output options

🔽 Appe<u>n</u>d to file

ОK

Ausgabe von Statistiken — Flächenbedarf

SYNOPSYS erlaubt die Ausgabe sehr detaillierter Statistiken, wobei besonders die Fläche interessant ist:

Design Vision - Top	Level.1 - [Sche	ematic.1 opw] opw					
Eile Edit View Select	t List <u>H</u> ierarchy	Design Schematic Attributes Tin	ning Test <u>W</u> indow	Help Racillate re a latellite			
	<u>अ</u> स् ।	Compile Design	🗕 D 📗 🕬 🏹	• 화태[]] 몇0 년 년 대도 550 [] ob/	<i></i>		
Logical Hierarchy	Cells (Hierarch	Compile <u>U</u> itra	ochemater U				
€ ⊡-D==> opw	Cell Name	Report Design					
🗳 🗄 Đalul	t⊖alul	Report Design Report Design Hierarchv					
		Report Design <u>R</u> esources					
		Report Constraints					
		Report Reference				I	
		Report <u>P</u> orts				#4	
		Report <u>C</u> ells				- H- 1	
		Report <u>N</u> ets Report <u>Cleaks</u>				#1	
		Report Clocks Report Area				1	
		Report Compile Options					
		Report Power				44	
		Reset Current Design		22,200,211,221,4		++-	
	· ·					- H -1	
				14 - Feiter			
							×
	<u> </u>	•	4				
De Opw							
alu_DW01_addsub_	0	flatten	false				
		structure structure boolean	true false				
		structure_timing	true				
Information: Gen	erating schemat	ic for design: opw					
design_vision-xg	-t> gui_zoom -w	indow [gui_get_current_windo	/ -view] -full				-
Log History Errors	/Warnings						
design_vision-xg-t>							
	2					Selected:	
Report Area	1		3	\diamond \bigcirc \diamond			
-Report for							
Current de sie							
Current desig	in: Jobw						
Current insta	nce:						
Current mata							
Depart option							
-Report option	5						
🔽 🔽 No line sp	olit						
1. the Ture ob							

<u>A</u>pply

Cancel

Design Vision - TopLevel.1	- [Report.1 - Area] opw						
📮 <u>F</u> ile <u>E</u> dit <u>V</u> iew <u>S</u> elect L <u>i</u> st	<u>H</u> ierarchy <u>D</u> esign S <u>o</u>	<u>c</u> hematic <u>A</u> ttribute	s <u>T</u> iming Test <u>W</u> i	ndow Hel <u>p</u>				_ & ×
⊯ ⊒ ⊜ ¶ Q Q ₹	🔽 🛛 🖉	t et 🌒 🔍 🛛 🔻	r 🔺 🗗 🔳 🕾	🎇 H 🛛 🏗 🖬 🕯	🛃 💕 🛛 opw	-		
(*****							
Report : area								
• Version: X-2005.09-SP2								
Date : Wed Feb 15 17:3	11:33 2006							
Library(s) Used:								
c35_CORELIB (File:	/local/tams1.1/ams/	v3.70/synopsys/	/c35_3.3V/c35_C0B	ELIB.db)				
Number of ports:	32							
Number of nets: Number of cells:	412 413							
Number of references:	12							
Combinational area	53526 042969	Fla	äche der Logikgat	ter				
Noncombinational area:	46883.238281	Flä	äche der Flipflops	/ Latches				
Net Interconnect area:	18729.000000	ge	eschätzte Verdrah	tungsfläche				
Total cell area: Total area:	100409.398438 119138.398438							
***** Fod Of Report ***	**							
Pe opw 🚆 Area								
×								
Combinational area:	53526.042969							
Noncombinational area:	46883.238281							
Met interconnect area.	10729.000000							
Total call area.	100400 200420							
Total area:	119138.398438							•
Log History Errors/Warning	3							
design_vision-xg-t>						 		
Ready						Selecte	ed:	

Die Ausgabe, hier in der Log-Datei, enthält folgende Angaben (μm^2):

Timing der Schaltung

Das gesamte Timing der Schaltung wird ausgegeben, wenn kein Netz explizit ausgewählt wurde, ansonsten wird das Zeitverhalten dieses Netzes ausgegeben:

Design Vision - TopLevel.1 - [Schematic.1 File Edit View Select List Hierarchy Design Si	opw] opw chematic <u>At</u> tributes Timing Test <u>W</u> indow Help		
] ☆ ∎⊜ [ช]] Q , Q, Q [⊻] Q	🚭 🚭 🔍 📗 🔳 Path Inspector	at 📪 jopw	_
Logical Hierarchy Cells (Hierarchical)	L_L⊑ I Timing <u>A</u> nalysis Driver tto Path Slack		
Cell Name Ref Name ⊕ Dalul Dalul alu	e Cell Path Slack of Selected Logic		
	Endpoint Slack		
	Let Capacitance Capacitance of Selected Nets		
	Path Profile View		
	C <u>h</u> eck Timing Report <u>Timing</u>		
	Report Timing Reguirements Report Clock Skew		
	Report Clock Tree		
	Report Path <u>G</u> roup Report <u>W</u> ire Load		
Report Timing Paths		000 8	9
Paths			1
F <u>r</u> om: pin 💌		Selection[1]	
Thr <u>o</u> ugh: pin 💌		Selection[2]	
		Selection[3]	
Report options			1
Worst paths per endpoint: 1	Ma <u>×</u> imum path del	ay:	
Max paths per group: 1	Minimum pat <u>h</u> dela	ay:	
Path type: full ▼	True path reporting		
Delay type: max	Report timing loops		
	🗖 Justify paths with input vector		
Sort by:	Find true path		
I group 💌	Path delay threshold: 0		
Significant digits: 2 🚔			
🔽 No line split	<u> </u>	able asynchronous arcs	
Show nets in <u>c</u> ombinational	path 🗖 Sho	ow net trans <u>i</u> tion time	
🗖 Show input pins in combinati	onal path 🔽 <u>S</u> ho	ow net capacitance	
Show dont_touch, size_only	attributes for nets and cells		
Output options			
✓ To report viewer			
To <u>file</u> : Report.txt		Browse	
Annend to file			
The property of the			
	OK	ancel <u>A</u> pply	

Die Ausgabe des hier dargestellten Timing-Reports bezieht sich auf den kritischen (längsten) Pfad. Das Timing wurde, was meistens der Fall sein dürfte, durch den Chiptakt spezifiziert:

Des	ign Vision - TopLevel.1 - [Report.1 - Timing] o	pw					
芦 Ei	ile <u>E</u> dit <u>V</u> iew <u>S</u> elect L <u>i</u> st <u>H</u> ierarchy <u>D</u> esign S <u>c</u> hematic	<u>A</u> ttributes <u>T</u> iming	Test <u>W</u> indow H	elp			_ 8 ×
1 🚘		Q 🗸 🛧 🗗	I (200 Mits 1 (1 ಜಿನಿ ಪನಿ ವಕಿ 🕴	s l opw	-	
		* LL			- <u> </u>		
h							
(^
e	clock clk5UMHz (rise edge)	10.00	10.00				
	reg3 reg[0]/C (DFF1)	0.00	10.00				
	reg3 reg[0]/0 (DFE1)	0.68	10.68 r				
	U476/Q (A0I221)	0.15	10.84 f				
	U473/Q (NAND41)	0.82	11.66 r				
	aluI/a[0] (alu)	0.00	11.66 r				
	alul/U23//U (INV3)	0.24	11.90 f				
	alu1/0321/0 (Umi2111) alu1/r21/B[0] (alu DW01 addsub 0)	0.26	12.10 I 12.16 r				
	alu1/r21/U21/0 (XNR20)	0.77	12.10 f				
	aluI/r21/U1 0/C0 (ADD32)	0.50	13.43 f				
	aluI/r21/U1_1/CO (ADD32)	0.36	13.79 f				
	aluI/r21/U1_2/C0 (ADD32)	0.36	14.16 f				
	aluI/r21/U1_3/CO (ADD32)	U. 36 0. 20	14.52 f				
	alu1/r21/01_4/00 (ADD32) alu1/r21/01_5/00 (ADD32)	0.36	14.88 E 15.95 f				
	alu1/r21/U1 6/C0 (ADD32)	0.36	15.61 f				
	aluI/r21/U1 7/C0 (ADD32)	0.36	15.97 f				
	aluI/r21/U1_8/CO (ADD32)	0.36	16.34 f				
	aluI/r21/U1_9/C0 (ADD32)	0.36	16.70 f				
	aluI/r21/U1_10/C0 (ADD32)	U. 36 0. 20	17.06 f				
	alu1/121/01_11/00 (ADD32)	0.36	17.45 I 17.79 f				
	aluI/r21/U1 13/C0 (ADD32)	0.36	18.16 f				
	aluI/r21/U1_14/C0 (ADD32)	0.33	18.48 f				
	aluI/r21/U1_15/Q (XOR31)	0.30	18.78 r				
	<pre>aluI/r21/SUM[15] (alu_DW01_addsub_0)</pre>	0.00	18.78 r				
	aluI/U239/Q (INV3)	U.1U 0.01	18.88 f				
	alui/0323/U (IMUX2I) alui/U319/0 (IMUX40)	0.21	19.08 r 10.30 f				
	alut/U317/0 (TMUX30)	0.21	19.30 I 19.75 r				
	aluI/v[15] (alu)	0.00	19.75 r				
	oReg_reg[15]/D (DFC3)	0.00	19.75 r				
	data arrival time		19.75				
	-lash -ll-Comma (wise seles)	20.00	20.00				
	clock cikbuMHZ (rise edge)	30.00	30.00				
	oBeg reg[15]/C (DFC3)	0.00	30.00 r				
	library setup time	-0.01	29.99				
	data required time		29.99				
	data required time		29.99				
	dada allivat cime						
	slack (MET)		10.24		Timing OK!!!		
	***** End Of Report *****						-
	D						
Read	iy					Selected:	

Design Vision - TopLevel.1 - [Schematic.1 opw] -- opw Eile Edit View Select List Hierarchy Design Schematic Attributes Iming Test Window Help ६ अट्रै 🕾 🔍 🔍 🗶 🔺 🗩 🔳 D> 🧏 🕴 🛣 🛣 🛣 👘 💭 opw ____X • Schematic.1 opw) 😅 🖬 é • <u>R</u>efresh - 🗆 × **P_E Hie** <u>Z</u>oom 4 Cell Pat Logic & Selection Tool Esc ⊕**0**0€ B-D € Zoom In Tool B € Zoom Out Tool ame = alul 💮 P<u>a</u>n Tool Pollow Selection Preferences... Ctrl+K Restore System Preferences Save Preferences Load Preferences AutoSave Preferences Select Highlighted Max Path • Min Path Toolbars ✓ <u>S</u>tatus Bar Selected by Color • × Set Current Color <u>C</u>lear Þ Clear <u>A</u>ll Ctrl+M <u>F</u>irst Color <u>N</u>ext Color Ctrl+N ✓ Auto Cycle Colors Design Vision - TopLevel.1 - [Schematic.1 opw] -- opw ▷ Eile Edit View Select List Hierarchy Design Schematic Attributes Timing Test Window Help _ 8 × 🖆 🖬 🎒 🕋 🔍 🍳 🔍 💽] & & & Q] 🔻 🔺 🕨 🔳 🗇 🦉 👬 📅 🚠 🚔 🛛 opw • • € Q O Flyline: From: reg3_reg[0]/Q Arrival: 0.681 To: U476/B Arrival: 0.681 Of Path: reg3_reg[0]/C - oReg_reg[15]/D Slack: 10.240 6 Ro-Represented Net: reg3[0] Capacitance: 0.036 999995661 Ħ B 3-Be <u>ŝ</u> la p 35-10-• • **F** Dopw P_E Hier.1 Ready Selected:

Dieser kritische Pfad, wie auch die Pfade ausgewählter Netze, können im Schematic visualisiert werden:

15. Sichern der internen Datenbasis: Design Vision - TopLevel.1 - [Hier.1] -- opw Elle Edit View Select List Hierarchy Design Schematic Attributes Timing Test Window Help 🔄]] 後、低、雪、雪、目、一〇 🚺 🗈 👯 🔡 🏜 🛃 😭 🚺 opw • 避 <u>R</u>ead... _ 🗆 🗙 🗠 S _ 🗆 × Remo<u>v</u>e All Designs (Hierarchical) Cell Pat Analyze... Name Ref Name Elaborate.. Setup... alu alul Link Design... <u>I</u>mport 🚽 <u>S</u>ave Ctrl+S Save Info Execute Script... Lice<u>n</u>ses... 🛃 Erint Schematic Close GUI E⊻it

Soll der Entwurf später noch einmal bearbeitet werden, so kann man die hier gesicherte Datei laden (anstatt Schritt 3, Seite 4). Sie beinhaltet neben allen Elementen der Hierarchie auch die individuellen Einstellungen für Syntheserandbedingungen, Attribute etc.

Save Design As	
Look in: 🔄 /home/tams_1/maeder/demo/	à 🔿 🔢 🏛
in work	
alu.vhd	
opw.vhd	
File <u>n</u> ame: opw.ddc	<u>S</u> ave
File type: Database Files (*.ddc *.ddc.gz *.db *.db.gz *.gdb *.sdb *.pdb *.edif *.eqn *.m	Cancel
Eormat: Auto	Synopsys:
Save all designs in hierarchy	

Sichern und Ausgabedateien erzeugen



16. Verilog Netzliste schreiben

Für die Simulation der synthetisierten Netzliste, aber auch für die folgende Platzierung und Verdrahtung durch ein Standardzell-Backend (CADENCE SoC Encounter), wird eine Datei in der Hardwarebeschreibungssprache Verilog erzeugt:

Save Design As	\odot \odot
Look in: 🔄 /home/tams_1/maeder/demo/	È 💣 🔳 🖩
in the second se	
alu.vhd	
pw.ddc	
opw.vna	
File <u>n</u> ame: opwSyn.v	<u>S</u> ave
File type: Database Files (*.ddc *.ddc.gz *.db *.db.gz *.gdb *.sdb *.pdb *.edif *.eqn *.m	Cancel
Eormat: Auto	SYNOPSYS'
Save all designs in hierarchy	
	14

Aus Effizienzgründen wird eine gemischte Simulation von VHDL-Testumgebung und Verilog-Netzliste bevorzugt. Die dazu notwendigen Schritte sind in der extra Beschreibung "VHDL- und mixed-mode Netzlistensimulation" erläutert.

17. ... fertig, Programm beenden:

