

回路記述言語Verilog-HDL

佐藤証 西9-613 akashi.satoh@uec.ac.jp

佐藤田安安 古時 日2000 ◆ Satoh	▲ 受信 (12,512) CSTUECTACTOR SAKURA	×)
電気通信大学 情報理	工学研究科 佐藤研究室	Â
SATOH SAT	University of Communications OH Lab	
佐藤研究室 > 講義 > 実験第 一	· J2課是且	
ホーム	▶ 実験第一 J2課題	
メンバー	□ 実験課題J1、J2の履修について	
設備	□ 論理回路とVerilog-HDL	
研究	■ 12言単語解2言はスライド	
講義	論理回路学	
文献	計算機アーキテクチャ基 どフルコード 礎論	Ц
リンク	コンピュータグラフィックス	
サイトマップ	実験第一	
	実験第二	
satoh.cs.uec.ac.jp/ja/lecture/Engineerin	gExperiment1/index.html	-

まずこの本でお勉強

- 加算器のVerilog-HDL
 記述を勉強
- その後ModelSimでシ
 ミュレーション





FPGAのデザインフロー





モジュール eq





シミュレーションモデル eqSim

module	eqSim;	/* -	致検出回	路 */
wire	S;	/* の	シミュレ	·ーター */
reg	х, у;			
eq	g1(s, x, y);			
initia				
begir	า			
\$mc	onitor(″%b %	b %b″,	X, Y, S	3, \$ stime);
\$d 1	isplay(″xy	S	•	time");
	x=0; y=0;			
#50) y=1;			
#50) x=1; y=0;			
#50) y=1;			
#50) Šfinish;			
end	• • • • • • • • • • •			
endmodule	9			



ModelSimの使い方

佐藤証 西9-613 akashi.satoh@uec.ac.jp

ModelSimの起動とプロジェクト作成

● コンソールで"vsim"と入力

File→New→Proeject... で"Create Project"ウィンドウが開く
Project Name: eq
Project Location: C:/J2

ModelSim ALTERA STARTER E	DITION 10.1d - Custom Altera Version		
<u>File</u> dit <u>V</u> iew <u>C</u> ompile <u>S</u> ir	nulate A <u>d</u> d L <u>i</u> brary T <u>o</u> ols Layo <u>u</u> t Boo <u>k</u> m	narks <u>W</u> indow <u>H</u> elp	
🖹 • 🚅 🔛 🤹 🍈 I 🐰 🖿 🕯	8 💭 🗋 💿 - 🗛 🖺 🗖 🛛 🍲 🎬 🛺 🏹	🔹 🎲 🛊 🗽 🔔 Layout NoDesign 🔍	
Columnicayout AllColumns		\$	
Library		: + @ ×	
Vame Type	Path	<u> </u>	
👖 work (empty) Library	C:/J2/work		
⊡ 220model Library	\$MODEL_TECH//altera/vhdl/220model		
	\$MODEL_TECH//altera/verilog/220m	M Create Project	
E altera Library	\$MODEL_TECH//altera/vhdl/altera	Project Name	
ditera_Insim Library	\$MODEL_TECH//altera/vhdl/altera_I	ed	
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	\$MODEL_TECH//altera/verilog/altera		
Library	\$MODEL_IECH//altera/vhdi/altera_mf	Project Location	
E ditera_mt_ver Library	\$MODEL_IECH//altera/verilog/altera	C:/J2 Browse	
	\$MODEL_TECH//altera/vehiog/altera		
E altoyh lih Library	MODEL_TECH/ /altera/vhdi/altgxb	Default Library Name	
	\$MODEL_TECH//altera/verilog/altgyb	work	
	\$MODEL_TECH//altera/vhdl/arriagx		
→ arriagx hssi Library	\$MODEL TECH//altera/vhdl/arriagx	Copy Settings From	
□ arriagx_hssi_ver Library	\$MODEL_TECH//altera/verilog/arriag	0/modelsim_ase/modelsim.ini Browse	
Tull arriagy ver Library	MODEL TECH/ /altera/verilon/arriagy	Copy Library Mappings C Reference Library Mappings	
+ Loading project eg		OK Cancel	
<pre># reading C:\altera\13.0\mo</pre>	delsim ase\win32aloem//modelsim.ini		
ModelSim>			
1		_	
	<no design="" loaded=""></no>	<no context=""></no>	

Verilog-HDLファイルの追加

- "Add items to the Project"ウィンドウ: Add Existing File
 またはメインウィンドウから Project→Add to Project→Existing File…
- "Add file to Project"ウィンドウ: Brows
- "Select files to add to project"ウィンドウ:
 c:/J2/Verilogeq.vを選択して開く.-HDL/
- "Add file to Project"ウィンドウ: OK
- "Add items to the Project"ウィンドウ: Close

Add items to the F	Project 💌
Click on the icon to a	add items of that type:
Create New File	Add Existing Pile
Create Simulation	Create New Folder
	Close

Add file to Project	
File Name	Browse
Add file as type	Folder Top Level
Reference from current location	C Copy to project directory

Verilog-HDLファイルの追加

● C:/J2/Verilog-HDLの下のeq.vを選択して"開く"

M	Select files to add t	o project		×
🔄 🏵 🗸 🕇 🌗 🕨 PC	Windows (C:) → J2 →	× ¢	J2の検索	Q
整理 ▼ 新しいフォルダー			:== ▼	
☆ お気に入り	□ 名前 ^	更新日時	種類 サイ	ズ
	퉬 work	2014/06/18 1:16	ファイル フォルダー	
la OneDrive	📓 add4.v	2014/05/27 15:05	V ファイル	1 KB
	🛃 dffn.v	2014/05/18 16:33	V ファイル	2 KB
👰 PC	eq.v	2014/05/19 2:10	V ファイル	1 KB
P akashi (ches-va	📔 eq2.v	2014/05/28 0:09	V ファイル	1 KB
🐌 ダウンロード	📔 fa.v	2014/05/27 14:38	V ファイル	1 KB
 ▶ デスクトップ ▶ ドキュメント ▶ ピクチャ ▶ ビデオ ▶ ミュージック ▲ Windows (C:) 	🖹 sel.v	2014/05/27 4:13	V ファイル	1 KB
77./11	夕(N):			vhd * v v
7777	طريع)، ا _ل ہم			vnu,*.v マ キャンセル :

コンパイル

- メインウィンドウ: Compile→Compile Selected
- Verilog-HDLファイルのStatusが?からレに変わり、

"Compile of eq.v was successful."と出る.

ModelSim ALTE	RA STARTER EDITION	10.1d - Custom Altera Version
File Edit View	Compile Simulate	Add Project Tools Layout Bookmarks Window Help
🖹 • 🚔 🔛 参	<u>C</u> ompile C <u>o</u> mpile Options	○ · AA இ: 兩 ② @ ② ③ ▲ ↓ * > * * > * * > *
Layout NoDesign	<u>S</u> ystemC Link	- inLayout AllColumns 💽 📙 🖏 🖓 🖼 - 🥠
	Compile <u>A</u> ll	
Project - C:/J2/eq	Compil <u>e</u> Selected	: + & ×
▼ Name	Compile Order	1odified
eq.v	Compile Report	05/18/14 10:16:25 PM
	Compile Summary	
		-
👖 Library 🗶 🎬 Pro	oject ×	< >
A Transcript		
<pre># Loading proje # Compile of eq</pre>	ct eq .v was successful.	•
ModelSim>		▼
		Project : eq <no design="" loaded=""> eq.v</no>



 エラーがあるとTranscriptウィンドウに赤く表示されるので、 そこをダブルクリックするとエラーの詳細が表示される
 さらに*** Error: "の赤い行をクリックするとソースコード ウィンドウが開いてエラーの行がオレンジで表示される

M2/Verilog-HDL/eq.v Unsuccessful Compile	
vlog -work work C:/J2/Verilog-HDL/eq.v Model Technology ModelSim ALTERA vlog 10.1d Compiler 2012.11 Nov 2	Version
Compiling module eq ** Error: C:/J2/Verilog-HDL/eq.v(5): (vlog-2730) Undefined variable	Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp
^{* _a} 5行目の'_a'に文法エラー Compiling module eqSim	
Close	
Project - C:/J2/eq	C:/J2/Verilog-HDL/eq.v - Default * Z
eq.v X Verilog 0 05/18/14 11:36:38 PM	1 module eq(s, a, b); /* êvoñH
コンパイル失敗	2 input a, b; 3 output s;
	4 wire na, nb, s1, s2; 5 assign #5 na = a, nb = ~b;
↓ ↓ Library × Project ×	6 assign #10 s1 = a & b, s2 = ▼ ◆ ~aが aになっている ◆
A Transcript	::::::::::::::::::::::::::::::::::::::
# Compile of eq.v failed with 1 errors. ModelSim> 1つのコンパイルエラー	
Ln: 5 Col: 0 Proje	ct : eq <no design="" loaded=""> <no context=""></no></no>



Simulate→Start Simulation
 Start Simulationウィンドウでwork/eqSimを選択

▼ Name	Type	Path	•
	Library	work	
- The part of the	Module	C:/J2/Verilog-HDL/eq.v	
eqSim	Module	C:/J2/Verilog-HDL/eq.v	
+ 220model	Library	\$MODEL_TECH//altera/vhdl/220model	
+ 220model_ver	Library	\$MODEL_TECH//altera/verilog/220m	
🕂 📕 altera	Library	\$MODEL_TECH//altera/vhdl/altera	
	Library	\$MODEL_TECH//altera/vhdl/altera_l	
	Library	\$MODEL_TECH//altera/verilog/altera	
	Library	\$MODEL_TECH//altera/vhdl/altera_mf	
an dan sa		······································	
•			
Design Unit(s)		Resolution	
work.egSim		default	_
Optimization			
· ·			



- シミュレーションウィンドウが開く
- Add→add to wave→all items in regionで波形ウィンド ウが開く
- "All items in region and below"と"All items in design" は下位レベルやソースの全ての信号線が表示される

ModelSim ALTERA STARTER EDITION 10.1d			
File Edit View Compile Simulate Add Structure Tools Layout Book	kmarks Window Help		
Image: Strange of the strange of t			
Io Log All Items in design Io Log All Items in design To Dataflow Io Watch Vinstance Design unit	X X X X X Y X Y		
Image: Constraint of the second state of the second sta	<pre>s StX Net Internal x x RegisInternal y x RegisInternal x regisInterna</pre>		
Library × 🕮 Project × 😰 sim ×	→ #ASSIGN#6 Assign Ready 5 → 11 wire s; /* it →		
A Transcript			
ModelSim> cls # D # Compile of eq.v failed with 1 errors. # Compile of eq.v was successful. ModelSim> vsim -qui work.eqSim			
Project : eq Now: 0 ps De	Delta: 0 sim:/eqSim		





● メインウィンドウで, Simulate→Run→All ● Finsh Vsimウィンドウで「いいえ(N)」を選択

- 「はい(Y)」を選択するとModelSimが終了してしまうので注意

ModelSim ALTERA STARTER EDITION 10.1d	Finish Vsim
File Edit View Compile Simulate Add Source Tools Layout Bookmark	
Image: Simulate Image: Simulate Design Optimization Layout Simulate Simulation Run Image: Simulation in the simulatine sin the simul	Are you sure you want to finish?
eqSim eqSir Break Run -Next Image: Provide the state of the stat	(はい(Y) いいえ(N)
Transcript	
Ln: 5 Col: 19 Project : eq Now: 0 ps Delta	

シミュレーション結果表示

- Waveウィンドウに結果が表示される
- ●画面を虫眼鏡で拡大して表示

📰 Wave	x
<u>File E</u> dit <u>V</u> iew <u>A</u> dd F <u>o</u> rmat <u>T</u> ools Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp	
Wave - Default	≷ ≝ X
🖹 - 🛎 🖬 🖏 🖨 ¾ 🖻 🛍 ଛାଇ । ◎ - 🚧 🔚 🦠 🕮 🕼 🛣 💁 🕇 🖛 🛶 ☷ 🗖 100 ៰₀ 븆 트↓ 트≱ 💥 🤹 🎬 🦉	1
▋ <mark>▏ ╡ ネネ ╡┆ ╬ Ѧ </mark> ╬ <i>│ ┩</i> ╶ <i>┩ ┩ ┩ ┩ ┩ ┩ ┨</i> ╡ ┠ ┣ <u></u> ╬ ┇ ┣ │ <u></u>	
3+ - +€ - 3- Search:	
Msgs	
/eqSim/s -No Data-	
I Wave /eqSim/x -No Data-	
Re Wave Now 200 ps ps 50 ps 100 ps 150 ps 200 ps 250 ps 300 ps 350 ps	
* Now: 200 ps Delta: 0	
] → + + € +] → Search: 🚽 魚 礁 🕸 🔍 🔍 🔍 🔍 🔍 🔍 🛄 📗 📗 📗 👘	
✓ /eqSim/s -No Data-	
√/eqSim/x -No Data-	
/eqSim/y	
Ale S Now 200 ps 200 ps 200 ps 4000 ps 6000 ps 8000 ps 10000 ps 12000	
680 ps 680 ps	
0 ps to 12100 ps Now: 200 ps Delta: 0	



シミュレーションモジュールに\$monitorコマンドを記述しているのでTranscriptウィンドウにテキストでも表示される

ModelSim ALTERA STARTER EDITION 10.1d					
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>C</u> ompile <u>S</u> imulate A <u>d</u> d T <u>r</u> anscript T <u>o</u> ols Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp					
🖹 • 🚘 🎧 🗇 X 🖻 🎕 🗠 🔍 📀 • 🗛 🖺 🔽 🦠 🕮 🚜 🌋					
🙀 sim - Default :::::: 🛨 🖻 🗵 📦 :cts :::::: 🛨 🖻 🗵 📄 C:/J2/Verilog-HDL/eq.v (/eqSim) - Default:::::: 🛨 🖻 🗵					
🔹 Instance Design unit Design unit type Visibility Total co 📥 💌 Name 🚔 Ln#					
eqSim eqSim Module +acc=< 17 \$display(" x y s					
$\begin{array}{cccccccccccccccccccccccccccccccccccc$					
* WillAL#14 Eq3iii Process fact-< • • • • • • • • • • • • • • • • • •					
Library × Mroject × Sim ×					
A Transcript 🖂 🚽 🗹 🗙					
VSIM 5> run -all					
# x y s time					
# 0 0 X 0					
# 0 0 1 25 # 0 1 1 50					
# 010 75					
# 1 0 0 100					
# ** Note: \$finish : C:/J2/Verilog-HDL/eg.v(22)					
# Time: 200 ps Iteration: 0 Instance: /eqSim					
Project : eq Now: 200 ps Delta: 0 sim:/eqSim/#INITIAL#14					



メインウィンドウで Simulate→Restart
 Restartウィンドウが出るのでOKをクリック
 「シミュレーションの実行」に戻って再シミュレーション
 全てのコマンドはTranscriptウィンドウで実行されているので、ここに手入力した方が早い

ModelSim ALTERA STARTER EDITION 10.1d	Keep:					
<u>File E</u> dit <u>V</u> iew <u>C</u> ompile <u>S</u> imulate A <u>d</u> d T <u>r</u> anscript T <u>o</u> ols Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp	✓ List Format					
■ • ☞ ■ ◎ ● X ʰ @ ① ① ◎ • A 指 丙 ◇ @ 2 X	Wave Format					
	Breakpoints					
ColumnLayout Default 🖤 🚳 - 🚳 - 🥵 - 🥵 🕺 🔆 💥 🛅 🐘 🐒 🚺 🚳 🗸	✓ Logged Signals					
	Virtual Definitions					
	Assertions					
	Cover Directives					
Image: Specific sector with the sector with the sector	ATV Format					
Pristance Design unit (Design unit type Visibility Total ct (Visibility Total ct (Visibility)) Pristance Unit (Visibility) Pristance Unit (Visib						
Image: Second	OK Cancel					
Image: String of the string	_					
Library × 🕮 Project × 🛺 sim × 4 × 4 × and a visit of the second						
R Transcript	+ @ ×					
# Break in Module eqSim at C:/ 32/Verilog-HD /eq. 120 22						
$\forall SIM 11 > run -all \leftrightarrow SI = 0$						
Project : eq Now: 0 ps Delta: 0 sim:/eqSim						

内部信号の表示

● Simウィンドウでモジュールを指定し,内部信号を表示

● \$monitor関数では "g1.s1"のようにモジュール内の信号や さらに"."でつないでモジュール内のモジュールを指定

ModelSim ALTERA STA	ARTER EDIT	TION 10.1d				
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>C</u> ompile <u>S</u> imulate A <u>d</u> d T <u>r</u> anscript T <u>o</u> ols Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp						
📗 • 📽 🔛 🛸 🖨 🥇	¥. 🐚 🛍 j	2210	- M 🖺 M] 🗇 🕮 🚑 🕱	_	100 ps 🔶 🚉 💱 🍩 🖄 🥨
Layout Simulate		ColumnLay	out AllColumn:	3		
🔊 sim - Default		- + 🗗 🗙	沟 Objects 🚃		C:/J2/	Verilog-HDL/eq2.v (/eqSim2/g1) - Default:::::::: 🛨 🛃 🗙
▼ Instance [Design unit	Design ur	▼ Name	Value	Ln#	Now ≑ →
eqSim2 e	eqSim2	Module	🥠 а	St1	5	assign s1 = ~(a & b), s2 = a b;
🚽 🚽 g1 e	eq2	Module	🥠 Ь	St1	6	assign s = ~(s1 & s2);
—🥝 #ASSIGN#5 ε	eq2	Process	🔶 s	St1	7	endmodule
— 🥥 #ASSIGN#5 €	eq2	Process	🧇 s1	St0	8	
ASSIGN#6 e	eq2	Process	🔶 s2	St1	9	module eqSim2;
📄 📮 🚅 g2 e	eq2	Module			10	wire 31, 32, 3;
−● #ASSIGN#5 €	eq2	Process			12	$reg x_1, y_1, x_2, y_2;$ eq2 $g_1(g_1, y_1, y_1), g_2(g_2, y_2, y_2);$
— — #ASSIGN#5ε	eq2	Process			13	assign s = s1 & s2:
L→ #ASSIGN#6 €	eq2	Process			14	initial
#ASSIGN#13 €	eqSim2	Process			15	begin
III └─ऄ #INITIAL#14 €	eqSim2	Process			16	\$monitor(" %b %b %b %b %d",
#vsim_capacity#		Capacity			17	g1.s1, g1.s2, g2.s1, g2.s2, s,\$stime);
					18	\$display("g1.s1 g1.s2 g2.s1 g2.s2 s time");
					19	x1=0; y1=0; x2=0; y2=0;
					20	#50 x1=1;
					21	#50 y1=1; y2=1;
					22	#50 X2=1; #50 \$finich.
] 🗐 📷 🖂		1	_	20	#50 \$IINI8N;
	And sime a		•			
🔒 Transcript =====						:
# g1.s1 g1.s2 g2.s1 g	12.s2 s		time			
# 1 0 1	0 1		0			
# 1 1 1	0 0		50			
# 0 1 1	1 0		100			
# 0 1 0	· C· / 72/	Warilog-P	150 TDI (eq2 11(22)			
# Time: 200 ps It.	eration:	0 Instar	nce: /eaSim2			
			,			_
			Proje	ect : eq2 Now: 200	ps Delta:	0 Module
						- //

別回路のシミュレーション

Simulate→End Simulation Projectウィンドウで右クリックしてポップアップメニューを出し、Add to Project→Existing File... eq2.vを追加

ModelSim ALTERA STARTER EDITION 10.1d					
File Edit View Compile Simulate Add Project Tools Layout Bookmarks Window Help					
┃ 🖻 - 😹 🖬 🗇 🍪 🐰 🐚 🎕 😂 😂 ⊘ - 🗛 🖺 🗖 🛛 🦠 🖽 🚑 🖻	[] * * * * *	Layout NoDesign			
ColumnLayout AllColumns 🗾 🖏 - 🚱 - 🥵 -	🥰 🛛 🗶 🕺 🕅 📓 🤇	🐐 🛛 🖸 🖸 🚺 👬			
L	<u>↓ ↓ </u>				
I Project - C:/J2/eq I IIIII	🗕 🗕 🛨 🛃 🗋 C:	/J2/Verilog-HDL/eq.v (/eqSim) - Defau	ilt: 🛨 🗗 🗙		
Name Status Type Orde Modified	Ln#	#	<u> </u>		
eq.v 🖌 Verilog 0 05/19/14 02: 10:09 AM	1	module eq(s, a, b); /* êvoñH */		
	2	input a, b;			
	3	output s;			
	4	wire na, nb, sl	, s2;		
	Edit	assign #5 na - ~	a, IID - aD;		
	Execute	assign $#10 \text{ s} = s$	1 s2:		
	Compile	endmodule			
	Add to Project	New File			
	Remove from Project	Existing File	/* êvoñH */ 🔻		
Library × E Project ×	Close Project	Optimization Configuration	▶		
A Transcript	Update	Simulation Configuration			
VSIM 10> restart -f	Properties	Folder	_		
VSIM 11> quit -sim	Project Settings				
ModelSim> run -all					
Project : eq <no design="" loaded=""> <no context=""></no></no>					

別回路のシミュレーション









別回路のシミュレーション

 Window内の追加したファイルを選択して, Compile→Compile Selected(選択したファイルが対象)または Compile All(すべてのファイルが対象)でコンパイル

ModelSim ALTERA STARTER EDITION 10.1d					
File Edit View	Compile Simulate	Add Source Tools Layout Bookmarks Window Help			
🔄 🖬 🕶 🚘 🚭 参	<u>C</u> ompile Compile Options	💿 - 🚧 🚰 两 🛛 🕸 🕮 🌠 🛐 🍷 🏞 🋊 🗼 🏦 📙 Layout NoDesign			
ColumnLayout A11	SystemC Link	💌 🛛 🖧 • 🖧 • 🥵 📫 • 🤻 👌 X4 👀 🖻 🖄 🗍 🔲 🖉 🔟 🖬 🖉			
	Compile <u>A</u> ll	}→ Search:			
Project - C:/J2/eq	Compil <u>e</u> Selected	🚞 🔜 📩 🖬 🗶 🔄 C:/J2/Verilog-HDL/eq2.v - Default 🚞	+ d ×		
▼ Name	Compile Order	10dified Ln#			
eq.v	Compile Report)5/19/14 02:10:09 AM 1 module eq2(s, a, b	() ;		
eq2.v	Compile Summary	15/19/14 02:43:14 AM 2 1nput a, b;			
L		4 wire s1, s2;			
		5 assign #10 s1 = ~(a & b), s2 = ~(a		
		6 assign #10 s = ~(7 endmodule	s1 & s2);		
I Library X R	piect ¥				
A Transcript			<u> </u>		
ModelSim> cls			_		
# □					
ModelSim>					
			•		
	Ln: 1	Col: 0 Project : eq <no design="" loaded=""> <no context=""></no></no>			

別回路のシミュレーション

Libraryタグのworkの下に、eq2とeqSim2ができている eqSimと同様にeqSim2をシミュレーション

ModelSim ALTERA	STARTER E	DITION 10.1d			
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>C</u> ompile <u>S</u> imulate A <u>d</u> d L <u>i</u> brary T <u>o</u> ols Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp					
🖹 - 🚔 🗑 🗇 🐇 🖹 🎕 🖄 💭 📿 ⊘ - 🗛 🖺 🗖 🤣 🎬 🛺 🎇 🕺 🋊 🆘 🏌 🍰 🏔 🧎 Layout NoDesign 💽					
ColumnLayout AllCo	ColumnLayout AllColumns 🗾 🖉 - 🖏 - 🥵 - 🥵 🛛 🛪 🖉 🖄 🖓 👔 🖉 👘				
🖺 Library 🚃				< 🗋 C:/.]2/Verilog-HDL/eq2.v (/eqSim2) - Default 👬 📶 🔟
* Name	Type	Path		Ln#	▲
□ work	Library	work	-	1	<pre>module eq2(s, a, b);</pre>
-M eq	Module	C:/J2/Verilog-HDL/eq.v		2	input a, b;
-M eq2	Module	C:/J2/Verilog-HDL/eq2.v		3	output s;
eqSim	Module	C:/J2/Verilog-HDL/eq.v		4	wire s1, s2;
eqSim2	Module	C:/J2/Verilog-HDL/eq2.v		5	assign s1 = ~(a & b), s2 = a b;
+ 220model	Library	\$MODEL_TECH//altera/vhdl/220model		6	assign s = ~(s1 & s2);
+- 220model_ver	Library	\$MODEL_TECH//altera/verilog/220m		7	endmodule
+	Library	\$MODEL_TECH//altera/vhdl/altera		8	The second secon
	Library	\$MODEL_TECH//altera/vhdl/altera_l		-	
Library 🗙 🛗 Projec	t ×		4	» [eq.	v × _ eq2.v ×
A Transcript					
VSIM 26> guit -sim					A
# Compile of eq2.v was successful.					
ModelSim>					-
					_
		"eq2.v" was m	nodified after it was compiled Lr	: 21 Col:	0 Project : eq <no design="" loaded=""></no>

プロジェクトの変更

- File→Close Projectでプロジェクトを閉じる
- File→New Projectで新しいプロジェクトを作る
 - 前のプロジェクトを閉じていない時は閉じるかどうか聞かれる
- Project Name: eq2で新しいプロジェクトを開く
 - 他のプロジェクトとモジュールを共有するときや,名前が同じ異なるモジュールがなければDefault Library Name: workでよい

	M Create Project
	Project Name
Class Project	eq2
	Project Location
	Browse
This operation will close the current project. Continue?	Default Library Name
	Work
(はい(Y) いいえ(N)	Copy Settings From
	0/modelsim_ase/modelsim.ini Browse
	Copy Library Mappings Reference Library Mappings
	OK Cancel



 プロジェクトeqに戻るときはFile→Resent Projectから選ぶ か、File→Openでファイル選択のポップアップウィンドウの 右下を"Project Files"に切り替えてmpf (Modelsim Project File)ファイルを表示して選択



問題1

- モジュールeqのゲートの遅延時間を0 にしてシミュレーションを行い、遅延ありのシミュレーションと比較せよ。
- それぞれのシミュレーション結果について、各変数の時間に対する変化を図(または波形)に表し、違いを明確にせよ。
- また、eqの内部端子の変化状況が分かるように表示してみよ。(シミュレーション波形では、"All items in region and below"か"All items in design"を設定)



- 変数a, b, c, d を受け取り、a = b とc = d がともに成り 立つとき出力s を1 に、それ以外のときs を0 にする回路 のモジュールを、モジュールeq を2 個使って作れ。
- このとき、シミュレーション手続きからeqの内部端子は どのように参照すればよいか。適当なシミュレーション を行い、実行例とともに示せ。
 - "Verilog-HDL 階層参照"でネット検索

問題3

設計した組み合わせ回路が正しく動作することを検証するためには、シミュレーターをどのように作ったらよいかを考えよ。そして、レポートにて説明せよ。





佐藤証 西9-613 akashi.satoh@uec.ac.jp

順序回路

D-FF(Delay:遅延)はクロックckの立下り(negedge)に同期してデータDを取り込み(遅延#10で)Qに出力する
 ckは#50ごとに反転する(1周期#100)

```
module clk( ck );
output ck;
reg ck;
initial ck = 1;
always #50 ck = ~ck;
endmodule
```

module dffn(Q, D, ck); input D, ck; output Q; reg Q1; initial Q1 = 0; assign #10 Q = Q1; always @(negedge ck) Q1 = D; endmodule



1ビットの記憶回路

● 制御信号 | でD-FFへの書き込みを制御

```
module r1( q, l, d, ck );
input l, d, ck;
output q;
wire nl, s1, s2, d1;
dffn f( q, d1, ck );
assign #5 nl = ~l;
assign #10 s1 = nl & q, s2 = l & d;
assign #10 d1 = s1 | s2;
endmodule
```



|=0のときqを保持



1=1のときdを書込み



逐次制御回路

現在の状態s1s0,入力a,出力b,次の状態s1's0'から状態遷移
 図と真理値表を作る





$$s1' = \overline{s1} \cdot s0 + s1 \cdot \overline{s0}$$

$$s0' = a \cdot \overline{s1} + s1 \cdot \overline{s0}$$

$$b = s1 + s0$$

逐次制御回路

$$s1' = \overline{s1} \cdot s0 + s1 \cdot \overline{s0}$$

$$s0' = a \cdot \overline{s1} + s1 \cdot \overline{s0}$$

$$b = s1 + s0$$

module m1(b, a, ck); input a, ck; output b; wire ns1, ns0, s1, s0, d1, d0, c1, c2, c3; dffn f1(s1, d1, ck), f2(s0, d0, ck); assign #5 ns1 = \sim s1, ns0 = \sim s0; assign #10 c1 = s1 & ns0, c2 = ns1 & s0, c3 = a & ns0; assign #10 d1 = c1 | c2, d0 = c1 | c3, b = s1 | s0; endmodule



状態遷移図

- ●順序回路を状態遷移図で表記して動作を解析,あるいは状態遷移図から順序回路を設計するときにはFFの値が「状態」を表す
- 入力と出力によってその状態間がどう遷移するかを調べる
 - CS実験第一J1課題の回路の動作を調べる
 - 入力はE, 出力はQ₁Q₀
 - 状態はQ₁Q₀(D-FFから直接なので出力と一致)
 - 2bitなので最大で4状態だが,全ての組合せを取らない場合もある



状態遷移図

● 組み合わせ回路の部分をE=0とE=1の場合に分けて簡単にする。



E=1のとき








• 真理値表から E, Q_1Q_0 と D_1D_0 の関係を求める



E=0 のとき 前の状態を保持 $Q_1 - D_1$ $Q_0 - D_0$





●現在の状態Q1Q0と次の状態D1D0から状態遷移図を作る

- クロックTが入る毎に状態が変化する
- 状態はQ1Q0={00,01,10,11}の4つ
- 入力はE={0,1}
- E=0ならば状態は変わらない
- この回路では状態Q₁Q₀がそのまま出 力になっている。







問題4

- ●状態遷移表m2に従って動く回路m2の状態遷移図を書き、回路を設計せよ。
- さらに、モジュールを書き、シミュレーションで動作を 確かめよ。ただし、モジュール内ではregを使わず、 dffnを用いて状態を保持せよ。シミュレーションでは、 クロックと入力とのタイミングについて注意せよ。
- そのシミュレーション結果の何をもって動作確認としたのかも説明せよ。
 - 回路m2は、初期状態0 にあるとき入力a から1 を受けとり、さらに、3 回1 を受けると出力b から1 を出力し、初期状態0 に戻る。







佐藤証 西9-613 akashi.satoh@uec.ac.jp



● 多ビットのデータは, x[3:0]のように配列で表す

- Parameter 宣言を使うことでビット数を容易に変更できる
 - ただし、プログラムのように動的に配列を確保するわけではないので、配列の大きさをmoduleの引数にすることはできない



module eorn(z, x, y);

$$x[n-1:0]$$
, $x[n-1:0]$, $x[n-1:0]$, $y[n-1:0]$, $y[n-1:0]$, $y[n-1:0]$, $x[n-1:0]$, $x[n-1:$



{ }で信号を囲むことで多ビットのデータが表現できる

 x[3:0] {x[3:2], x[1:0]} {x[3], x[2], x[1], x[0]} 等は同じ
 左シフト: {x[2:0], 0} 左巡回シフト: {x[2:0], x[3}
 右シフト: {0, x[3:1]} 右算術シフト: {x[3], x[3:1]}

 下のコードはwを自己参照しているように見えるが,実際の回路は1ビットずつずらしながらNOTしている.

module	notn(y,	x);
paramet	er	n = 4;
input		Χ;
output		у;
wire	[n-1:0]	w;
assign	#5	{ y, w[n-1: 0] }
	= '	~{ w[n-1: 0], x };
endmodule		



問題5(加算回路の設計)





問題5(加算回路の設計)

- 全加算器をサブモジュールとして使うと、ビットの長さnをパラ メーターにした記述はできない。ビットのベクトルを使って、n桁の加算器をひとつのモジュールで実現せよ。
- このモジュールはadd4のように、最上位の桁上げcuと、最下位に加えるciを持つこと。
- さらに、シミュレーションにより評価せよ。



条件付きassign文

assign 変数= 条件? 条件が1(真) のときの値: 条件が0(偽)のときの値;

● 条件はC言語と同じ >,<, >=,<=, ==, !=, <=などが, またその論理積(&&)や論理和(||)などが利用できる



問題6(nビットレジスター)

- 以下に示す仕様のクロックに同期して動作するn ビット レジスターのモジュールrを設計し、シミュレーショ ンにより評価せよ。
 - データ入力: d,
 - 制御信号入力: load,
 - データ出力: *q*.
 - 動作:

繰り返し動作の記述

for (制御変数の初期設定; 継続条件; 制御変数の変更) 文;

```
module sel(z, x, c);
  parameter n = 8;
  input [n-1: 0] †x;
  input c;
  output [n-1: 0] z;
  assign #10 z = c? x: 0;
endmodule
               / n=2に指定
module selSim; /
 wire [1:0] p;
  reg [2:0] v;
  sel \#2 / g(p, v[1: 0], v[2]);
  initial
   begin
      $monitor("%b %b", v, p, $stime);
      for (v=0; v<7; v=v+1) #100;
      #200 $finish;
   end
endmodule
```

```
module sel2( z, x, c );
input [1: 0] x;
input c;
output [1: 0] z;
assign #10 z[1] = x[1] & c,
z[0] = x[0] & c;
endmodule
```

問題7(加減算回路の設計)

- ●以下に示す仕様のn桁の加減算回路のモジュールを設計 し、シミュレーションにより評価せよ。ただし、負の数 は2の補数表現で表されているものとせよ。
 - 数の入力: *n*桁の整数*x*, y;
 - 制御信号入力: k;
 - 出力: 桁上げcu とn 桁の和s;
 - 動作: k = 0のときs = x + y、

$$k = 1$$
のとき $s = x - y$ 。

●何れの場合にも、cu はaddn からの桁上げを与える。

負数を2の補数で表したとき、x - y = x + y + 1 である。した がって、kの値によってyとyの何れかを選ぶ選択回路の出 カとxを加算すればよく、1の加算には、桁上げ入力を使う。



 Xilinx PlanAheadとNEXSYS 3を用いて、実際に加算器 を動作させよ.

実験第一J2課題

FPGAボード NEXSYS 3

佐藤証 西9-613 akashi.satoh@uec.ac.jp

NEXSYS 3の構成



NEXYS 3のブロック図



Xilinx開発ツール



UCF(User Constraints File)

● FPGA実装の制約を与えるファイル
 ● 実験ではIOピンの接続を設定

adder4.ucf

# 4-bit adder				
#I FD	IOSTANDARD):最小	標準	最大
NET " cy " LOC = M11 IOSTANDARD = LVCMOS33;	LVTTL	3.0	3.3	3.465
NET " $s < 3$ >" LOC = V15 IOSTANDARD = LVCMOS33;	LVCMOS33	3.0	3.3	3.465
NET " $s < 2$ >" LOC = U15 IOSTANDARD = LVCMOS33;	IVCMOS25	2.3	2.5	2.7
NET $s < 1 > LUC = V16 IUSTANDARD = LVCMUS33;$	LV CMOS25	1.65	1.0	1.05
NET $s < 0 > LOC = UT6 TOSTANDARD = LVCMOS33;$	LVCMOS18	1.65	1.8	1.95
#SWITCH	LVCMOS15	1.4	1.5	1.6
NET " $x < 3$ >" LOC = T5 IOSTANDARD = LVCMOS33;	LVCMOS12	1.1	1.2	1.3
NET $X < 2 >$ LUC = V8 IUSTANDARD = LVGMUS33, NFT $x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > x < 1 > $	PCI33 3	3.0	3.3	3.465
NET " $x < 0$ >" LOC = N8 IOSTANDARD = LVCMOS33;	PCI66_3	3.0	3.3	3.465
NET $y < 3 > 2$ LOC = M8 IOSTANDARD = LVCMOS33;	— דומידי ד 10	17	10	1.0
NET $y < 2 > 10 \text{ LOC} = V9 IOSTANDARD = LVCMOS33;$	HSIL_I_I8	1./	1.8	1.9
NET " $y < 1>$ " LOC = T9 IOSTANDARD = LVCMOS33;	$HSTL_{III}18$	1.7	1.8	1.9
NET " $y < 0$ >" LOC = V10 IOSTANDARD = LVCMOS33;	SSTL18 I	1.7	1.8	1.9
#PUSH BUTTON	— т с ттор	22	2.5	2.7
NET "c" $LOC = B8$ IOSTANDARD = LVCMOS33;	551L2_I	2.3	2.5	2.7

ピンアサイン

Nexsys3の回路図 J2/NEXYS3_sch.pdf





デフォルトはLVCMOS33 LVCMOS33とLVTTLは同じ0~3.3V動作だが、LVTTL の入力は0.8-2.0V間が不安定なため普通は使わない.



PlanAheadを起動

コンソールで"Planahead"と入力 Create New Projectをクリック



プロジェクトフォルダを指定

- Project name: add4
- Project location: J2/work
- Create project Subdirectoryにチェック

O	New Project	×		
	Create a New PlanAhead Project This wizard will guide you through the creation of a new project To create a PlanAhead project you will need to provide a name and a location for your p Next, you will specify the type of flow you'll be working with. Finally, you will specify you sources and choose a default part.	project files. µr project		
		O	New Project	×
		Project Name Enter a name	e for your project and specify a directory where the project data files will be stored	R
		Project name:	add4	0
		Project location:	C:/J2	0-
		Create proje	ot subdirectory	
		Project will be cr	eated at: C:/CS-Jikken/work/seg7dec	
PlanAhead	To continue, click Next.			
	< Back Einish			
			< Back Next > Einish	Cancel

プロジェクトタイプの指定

● RTLプロジェクトにチェック

New Project	×
Project Type Specify the type of project to create.	
RTL Project You will be able to add sources, generate IP, run RTL analysis, synthesis, implementation, design planning and analysis.	
 Post-synthesis Project You will be able to add sources, view device resources, run design analysis, planning and implementation. Do not specify sources at this time 	
 I/O Planning Project Do not specify design sources. You will be able to view part/package resources. 	
 Import ISE Place & Route results You will be able to do post-implementation analysis of your design. 	
 Imported Project Create a PlanAhead project from a Synplify, XST or ISE Project File. 	
< <u>B</u> ack Next >	



● Add filesで次のファイルを追加

- J2/Verilog-HDL/add4.v (fa.vなどを適宜作成、追加)
- Copy sources into projectTarget language: Verilog

] Nev	v Pr	oject				×
Add \$	Sour lecify	ces / HDL and	d netlist file	es, or directories containing	g HDL and netlist files, to add to your project. Create a new source	
ŤI	e on	disk and	add it to yr	our project. You can also a	dd and create sources later.	
	Id	Name	Library	HDL Source for	Location	
ve	1	add4.v	work	Synthesis & Simulation	← C¥J2	
(Ve)	2	ta.v	WORK	Synthesis & Simulation		
						*
						F
			<	Add Files	Add Directories <u>C</u> reate File	
	Scan	and add	RTL <u>i</u> nclud	le files into project		
V	Сору	<u>s</u> ources	into projec	t		
\checkmark	Add	so <u>u</u> rces 1	from subdir	ectories		
Ta	arget	language	: Verilog	-]	すべてのサン
	-				L	
						Connect
					<u><u>Next</u></u>	Cancer

その他のIPを追加

●何も指定しない

New Project	×
Add Existing IP (optional) Specify existing configurable IP, DSP composite, and Embedded composite files to add to your project.	Solution
Id IP Name IP File	
Copy sources into project	
< <u>Back</u> <u>Next</u> <u>F</u> inish Ca	



● Add filesで次のファイルを追加

- J2/add4.ucf
- Copy constraints file into project

Constraint File	Location				
add4.ucf	C:¥J2				
		Add Files	D <u>C</u> reate File		
	ints files into project				

デバイスの指定

Package: csg324

• Speed grade: -3

• Temp grade: C

Family: Spartan-6Sub-Family: Spaltan-6 LX

New Project x **Default Part** ि Choose a default Xilinx part or board for your project. This can be changed later. Specify Filter 📎 Parts Package csg324 Product category All Ŧ 📕 Boards Family Spartan-6 Ŧ Speed grade -3 w. Sub-Family Spartan-6 LX Ŧ Temp grade C Ŧ Reset All Filters Search: Q-I/O Pin Available LUT Block Min Ope Part FlipFlops DSPs MCBs. Count IO Bs Elements RAMs Temper xc6slx9csg324-3 2 324 200 5720 11440 32 16 0 xc6slx16csg324-3 xc6slx25csg324-3 324 226 15032 30064 52 38 2 0 58 2 🔷 xc6slx45csg324-3 324 218 27288 54576 116 0 ٠ 111 <u>N</u>ext > < <u>B</u>ack Finish Cancel



●内容を確認してFinish

C	New Project	×
	New Project Summary	
	A new RTL project named 'seg7dec' will be created.	
	(i) 1 source file will be added.	
	⚠️ No Configurable IP files will be added. Use Add Sources to add them later.	
	1 constraints file will be added.	
	The default part and product family for the new project: Default Part: xc6slx16csg324-3 Product: Spartan-6 Family: Spartan-6 LX Package: csg324 Speed Grade: -3	
Plan Ahea	To create the project, click Finish	
	< <u>B</u> ack <u>N</u> ext > <u>Finish</u> Cancel	





Sourcesウィンドウでadd4を選んでGenerate Bitstream を実行しYesをクリック



66





ビットファイルのFPGAへの書き込み

Program and DebugのLaunch iMPACTをクリック



ビットファイルのFPGAへの書き込み

●書き込みツールが起動.プロジェクトファイルは作成不要.

Boundary S ISE iMPACT (P.20131013) - [Boundary S	Scan]	
🛞 <u>F</u> ile <u>E</u> dit <u>V</u> iew Operations <u>O</u> utpu	t Debug <u>W</u> indow <u>H</u> elp	_ B ×
🗋 🌶 🖶 📫 🛍 🗰 🗰 🗮	₩ K?	
iMPACT Flows ↔ □ ● × Image: Boundary Scan Image: SystemACE Image: Create PROM File (PROM File For Image: Create PROM File (PROM File For) Image: Create PROM File (PROM File For) <td>Right click device to select operations</td> <td></td>	Right click device to select operations	
Console	Save Project File	
Setting Target Device to Mode		・
Console C	ファイル名(<u>N</u>): ファイルの種類(<u>T</u>): iMPACT Project File (*.ipf)	•
		保存(S) キャンセル

ビットファイルのFPGAへの書き込み

● Operations → Program

File Edit View Operations Output Debug Window Help IMPACT Flows Get Device ID Get Device Signature/Usercode One Step SVF Access FUSE Read Device DNA Six16 4.bit
IMPACT Flows Get Device ID Get Device Signature/Usercode One Step SVF One Step SVF One Step XSVF Read Device DNA Access eFUSE Registers MPACT Processes
IMPACT Flows Get Device ID Impact Flows Get Device Signature/Usercode Impact Flows Get Device Signature/Usercode Impact Flows Get Device Signature/Usercode Impact Flows One Step SVF Impact Flows One Step SVF Impact Flows One Step SVF Impact Flows Impact Flows Impact Flows
Image: Second and Secon
Image: SystemACE One Step SVF Image: Create PROM F One Step XSVF Image: One Step XSVF Image: Create PROM F Image: One Step XSVF
Image: Cleate PROM One Step XSVF Read Device DNA Access eFUSE Registers iMPACT Processes + □ ⊕ × Available Operations are:
Read Device DNA slx16 Access eFUSE Registers 4.bit iMPACT Processes ↔ □ ₽ × Available Operations are: ▲
Access eFUSE Registers sta16 iMPACT Processes ↔ □
iMPACT Processes ↔ □ ♬ × Available Operations are: ▲
Available Operations are:
Program
Get Device ID
Read Device Status
One Step SVF Boundary Scan
Console 😵 Device Programming Properties - Device 1 Programming Properties
Setting Target Device
Boundary-Scan
Device 1 (FPGA xc6slx16) Property Name Value
Console C Errors 1
Programs the selected devices

ビットファイルのFPGAへの書き込み





●SW7-0の切り替えに応じてLEDが点灯する

