

School of Engineering

Diploma in Electronics Computer & Communications Engineering (EGDF01)

EXPERIMENT NO	:	Lab 5 (Duration : 2 hours)
EXPERIMENT TITLE	:	RTL-to-Gates Level Synthesis of <mark>Counte</mark> r using Cadence RTL Compiler
OBJECTIVE	:	 Code a Counter in Verilog and simulate using Cadence Incisive Simulator Synthesize the RTL codes to gates using Cadence RTL Compiler Perform Post-synthesis simulation of design

Exercise 1 : To simulate a 4-Bit Counter in RTL codes.

- (i) Use editor, gedit, to view the codes
- (ii) Use Cadence Incisive Simulator to simulate the design
- Open a new terminal and type: *pwd* This linux command will show the present working directory.
 cd term2 Change directory to "term2"
 source cshrc Run a script file that will setup a proper environment so that various software can be launched.
- 2. Type *nclaunch*& to start the GUI as shown in Figure 1. Select **counter.v** and **right-click→Edit**.

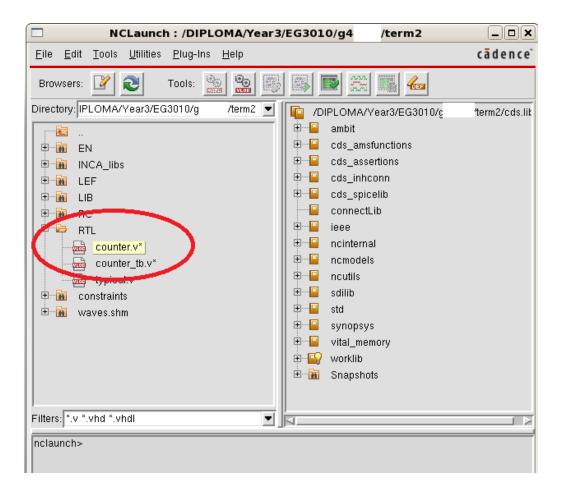


Figure 1 NCLaunch Window

3. Figure 2 shows the verilog code of the counter. It has 4 bits output named as **count**. Read the code to gain some understanding about the counter's behavior. As you can see, **count** will only change either at positive-edge of **clock** or negative-edge of **reset**. Close the file.

Documents ×		counter.v	×	
🔰 counter.v	1	module		nter (clk, reset, count);
	2			put clk, reset;
	3		ou	tput reg [3:0] count;
	4			
	5		al	ways @(posedge clk or negedge reset)
	6		be	gin
	7			if (reset == 0)
	8			count <= 0;
	9			else
	10			count <= count +1;
	11		en	d
	12	endmodu	ıle	
	13			
	14			

Figure 2 Verilog Code for 4-bit Counter

Select counter_tb.v and right-click→Edit. Figure 3 shows the test bench. Read the code to understand the test bench. Replace "XXX" with "BEH" for this exercise.
 When the variable BEH is defined, the file "./RTL/counter.v" will be read/included by the simulator as it contains the module counter to be tested. Save and close the file.

Documents	× 🗇 counter_tb.v ×
🕽 counter_tb.v	1`timescale lns/10ps
	2 `define PERIOD 10
	3 `define PERIOD_HALF 5
	4
	5//comments of BEH, SYN, PAR
	6//RTL verilog> define BEH
	7//After synt> define SYN
	8//After P&R> define PAR
	9
	10 //***********************************
	11 //**********************************
	12 define XXX
	13///**********************************
	15 `ifdef BEH
	16 `include "./RTL/counter.v"
	17 elsif SYN
	18 //directory is wrt where you launch nclaunch
	19 `include "./RTL/counter synt.v"
	20 `include "./RTL/typical.v"
	21 `else
	22 `include "./RTL/counter pnr.v"
	23 `include "./RTL/typical.v"
	24 endif
	25
	26 module counter tb;
	27 reg clk_tb, reset_tb;
	28 wire [3:0] count_tb;
	29 integer i;
	30 parameter width=4;
	31
	<pre>32 counter ul (.clk(clk_tb), .reset(reset_tb), .count(count_tb));</pre>

Figure 3 Verilog Code for Test bench

5. Select **counter.v** and **counter_tb.v** (to select, press **Ctrl** & **click** on the file, as shown in Figure 4). **Right-click→NCVlog**. Then click **OK**. This will compile the verilog files to C code for faster simulation.

NCLaunch :	/DIPLOMA/Year3/EG3010/g4 /term2	
<u>F</u> ile <u>E</u> dit <u>T</u> ools <u>U</u> tilities <u>F</u>	lug-Ins <u>H</u> elp	cādence [°]
Browsers: 📝 湕 To Directory: IPLOMA/Year3/EG30	ols: 🗽 😫 🐼 🐼 💽 💥 🐇	/term2/cds.lik
· · · · · · · · · · · · · · · · · · ·	ambit	
EN EN	Compile Verilog	×
B MincA_libs B IEF B Min LIB	File // DIPLOMA/Year3/EG3010/ /DIPLOMA/Year3/EG3010/	
BB. RC EE. RTL	Work Library	•
counter.v*	Overwrite log file 🛁 ncvlog.log	
typical.v* ⊕ math constraints	📕 Error Limit 15 🚆	
🗄 🛅 waves.shm	📕 Update if needed	
	📕 Enable line debug	
	🔟 Define Macro	
	☐ Include Directories	
Filters: *.v *.vhd *.vhdl		
nclaunch>	☐ Other Options	
	Advanced Options.	
	OK Cancel Apply Help	selected

Figure 4 NCLaunch Window

6. Select **worklib/counter_tb** (as shown in Figure 5). **Right-click→NCElab**. Then click **OK**. This will link up all the related compiled modules for simulation.

Elaborate	×	ear3/E	G3010	/g· term2	
Design Unit worklib.counter_tb					cādence [°]
		B			
([lib.]cell[:view])	1		.	 cds_spicelib	
🔟 Snapshot Name				connectLib	
Work Library	•		· _	eee ncinternal	
Overwrite log file [ncelab.log				ncmodels	
Overwrite log file — ncelab.log			∎ <mark> </mark>]	ncutils	
📕 Error Limit 15 🛢				sdilib	
🔟 Update if needed				std synopsys	
				vital_memory	
Access Visibility All	-			worklib	
🔟 Executable Filename	[]			counter	
(Default: ncelab)					
Other Options				Snapshots	E E
					Δ
Advanced Op	otions	pmpiling	design (unit worklib.counter.	
		N data =	= 21.2M t	ntal	
	1		1/0.24 0		
OK Cancel Apply H	Help				1 items selected

Figure 5 NCElab Window

7. Select **Snapshots/worklib.counter_tb:module** (as shown in Figure 6). **Rightclick→NCSim**. Then click **OK**. This will allow you to setup the environment for simulation.

S	imulate 🛛 🗙	
		cādence
Snapshot	worklib.counter_tb:module ([lib.]cell[:view])	
Command Filename		connectLib
Overwrite log file 🛁	ncsim.log	eee • • • • ncinternal
Error Limit	15	• nomernal • • • • • • • • • • • • • • • • • • •
Update if needed Update source files	, <u>a</u>	er∎ ncutils er∎ sdilib er∎ std
Run Mode	Graphical UI 🚽	B I synopsys B I vital_memory
SimVision Argument	s	e∰ worklib e
☐ Executable Filename		e counter_tb
Other Options	(Default: ncsim)	Snapshots worklib.counter_tb:module
	Advanced Options	ear3/EG3010/ -logfile ncelab.lc A e Design Systems, Inc. i = 42.0M total ital (1.9s, 2.9% cpu)
OK Cancel	Apply Help	
		1 items selected

Figure 6 NCSim Window

8. Select **counter_tb** (as shown in Figure 7). **Right-click→Send To New→ Waveform Window**. The related signals of counter_tb will be probed for display.

🕅 Design Browser	1 - SimVision	_ 🗆 🗙
$\underline{E}ile \underline{E}dit \underline{V}iew \underline{S}elect \mathbf{E}\underline{x}plore Sim\underline{u}lation \underline{W}indow$	s <u>H</u> elp	cādence ⁻
🎥 🐚 🖉 🐟 🕅 💥 🖉 🔤	🎽 🖡 🔶 Send To: 🗽 🚝 🗟	📲 📰 🔲 🔳
: № TimeA - = 0	│ Search Times: Value ▼	🗖 N., N.,
] 💽 ▾ 💷 🔣 [🖶 🖶 📮 [🌰 [፻፷፮ 0+0		
Design Browser 🛛 🗙 🤆	Objects Methods	
🛛 📕 Browse: 🕶 🎯 All Available Data 🛛 🔽 🐘 📰	Name 👻	Value (as recorded)
simulator Counter_tb	Clk_tb	x 'h x 'd x x 'd 4
Leaf Filter: *		2 Filter: [*] ■



9. Select count_tb[3:0] (the output of counter as shown in Figure 8). Simulation→Run. Then View→Zoom→Full X. Use the Sliding Bar and View→Zoom→In X (or Alt-i) to zoom in to count_tb[3:0]=F as shown. Click on the waveform screen to bring TimerA to clock edge as shown. There is no delay between the positive-edge of clk_tb and output of counter since the counter is modeled by verilog behavioral code.

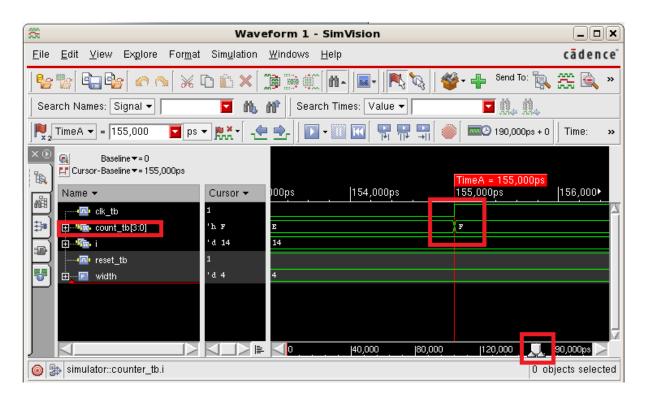


Figure 8 SimVision/Waveform Window

10. Exit all the software.

- **Exercise 2 :** To synthesize the 4-Bit Counter, of Exercise 1, using Cadence Software RTL Compiler.
- 1. Open a new terminal and type:

pwd
If you're not in your home directory, do a cd ~ to change to your home directory
...g4x_xx.
cd term2
source cshrc
cd RC

2. Type *rc* to launch the RTL Compiler. You should see the RC-shell prompt as shown in Figure 9.

				Te	erminal	
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>T</u> erminal	Ta <u>b</u> s	<u>H</u> elp	
Send	us f	eedba	ck at rc_	feedba	ack@cadence.com.	
rc:/:						==

Figure 9 RC-shell Prompt

3. Type:

source counter.tcl

counter.tcl contains the instructions for the setup of various libraries, constraints and etc. for the software to produce the netlist from your behavioral verilog code.

4. Type:

gui_show

This will show you the schmatic of the synthesized circuit. 4 registers, 2 gates and 1 inverter are used to realise the function of the 4-bit counter.

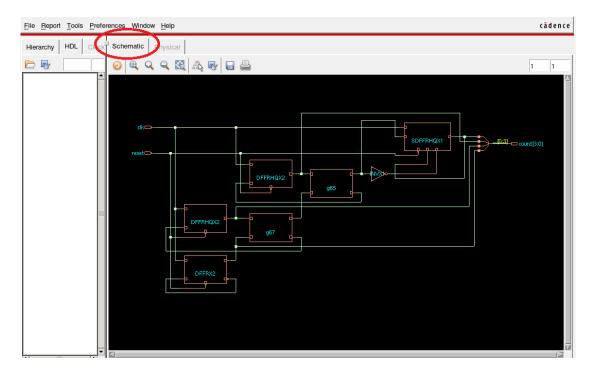


Figure 10 Schematic

- Close the window. You can type *gui_hide* to close the window.
 Type *exit* to exit RTL Compiler.
- 6. Type:

pwd to know which directory you are in.

You can cd .. to go up to the parent directory or cd ~ to your home directory if you get lost.

Make sure you are in .../term2/RC, type *more counter.tcl* to view the setup of the synthesis run you have just performed. A lot of knowledge and expertise are required to do a synthesis for complex digital circuit. Use **space bar key** to scroll throught the file.

7. The synthesized netlist file is in .../term2/RTL.
Go to that directory and do a *more counter_synt.v* to view the netlist. You can see the 7 gates used for the 4-bit counter.

This exercise is optional:

Exercise 3 : Post-synthesize simulation using Cadence Incisive Simulator

- Type:
 cd .. to go up to .../term2.
 nclaunch&
- 2. **Right-click→Edit** on counter_tb.v

Change the variable to **SYN** as shown in Figure 11. Save and close the file. typical.v is the standard cell library that contains the modules of the gates that we're going to use.

	1
2	counter_tb.v (~/term2/RTL) - gedit
<u>File Edit View Sea</u>	arch Tools Documents Help
ि ⋛ 📁 → 🔓 New Open → Sav	
Documents ×	🗇 counter_tb.v 🗙
Counter_tb.v	<pre>1 `timescale lns/l0ps 2 `define PERIOD 10 3 `define PERIOD_HALF 5 4 5 //comments of BEH, SYN, PAR, ANNOTATE 6 //RTL verilog> define BEH 7 //After synt> define SYN 8 //After P&R> define PAR 9 10 //***********************************</pre>
	16 `include "./RTL/counter.v"
	<pre>17 `elsif SYN 18//directory is wrt where you launch nclaunch 19 `include "./RTL/counter_synt.v" 20 `include "./RTL/typical.v"</pre>
	21 else 22 `include "./RTL/counter_pnr.v" 23 `include "./RTL/typical.v" 24 `endif

Figure 11 Test Bench

3. We are going to simulate the test bench like what we did in Exercise 1. Refer to it if you not sure.

Select **counter_synt.v**, **counter_tb.v** and **typical.v** and **right-click→NCVlog** (Figure 12).

Note : This may take a while for the **typical.v** file to be compiled.

<u>File E</u> dit <u>T</u> ools <u>U</u> tilities <u>P</u> lug-Ins <u>F</u>	<u>l</u> elp cādence
Browsers: 📝 🜊 🛛 Tools: 🥸	
Directory: IPLOMA/Year3/EG3010/g4 EN EN INCA_libs ELEF ELIB RC RTL Counter_synt.v counter_tb.v* typical.v*	Image: Construction Image: Constructi
E waves.shm Filters: [*] .v *.vhd *.vhdl Inclaunch>	 wital_memory worklib Snapshots
	3 items selected

Figure 12 NCLaunch Window

4. Select worklib/counter_tb and right-click→NCElab.
Select Snapshot/worklib.counter_tb:module and right-click→NCSim.
Select counter_tb and right-click→Send To New→Waveform Window.
Select count_tb[3:0]. Simulation→Run.
Zoom in to count_tb[3:0]=F as shown in Figure 13.
Click on waveform to bring TimerA to the desired location.
Right-click→Create a marker to put additional marker.

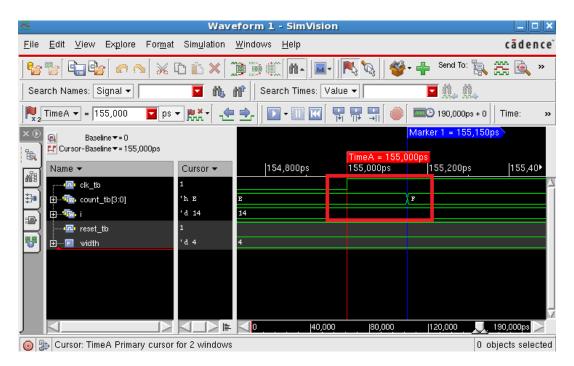


Figure 13 SimVision/Waveform Window

5. As you can see, there is delay of **150ps** between postive-edge of **clk_tb** and counter's output. This is due to the gate delay since now we are simulating a real circuit.



School of Engineering

Diploma in Electronics Computer & Communications Engineering (EGDF01)

EXPERIMENT NO	:	Lab 6 (Duration : 2 hours)
EXPERIMENT TITLE	:	Place and Route of Counter using Cadence Encounter
OBJECTIVE	:	To perform place and route of Counter

Exercise 1

To do place and route of the counter synthesized in previous lab using Cadence software - Encounter.

1.	Open a new Terminal. Type :			
	pwd	(show present working directory)		
	Cd (change directory to home directory			
	cd term2	(change driectory to term2 directory)		
	source cshrc	(execute a setup file called cshrc)		

 Change directory. Type : *cd EN encounter* to launch the software. You should see Figure 1.

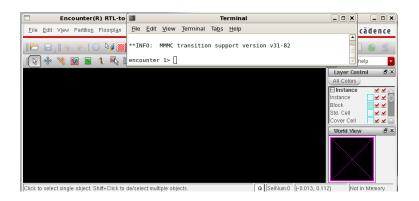


Figure 1 Encounter-shell Prompt

3. In Encounter GUI, File→Import Design.

As shown in Figure 2, click **Load**, select **counter.globals** and click **Open**. Click **OK** to import the design.

	Design Import	
Netlist:		
 Verilog 		
Files:	/RTL/counter_synt.v	
	Top Cell: 🔾 Auto Assign 💿 By User: 🛛 counter	
O OA		
Library:		
Cell:		
View:		
Technology/Physical Libr	raries:	
O O A		
Reference Libraries:		
Abstract View Names:		
Layout View Names:		
LEF Files	./LEF/st_gsclib045.lef	<u>.</u>
Floorplan		
IO Assignment File:		P
Power		
Power Nets:	VDD	
Ground Nets:	VSS	
CPF File:		P
Analysis Configuration –		
MMMC View Definition File	e: eg3010.view	P
	Create Analysis Configuration	
ОК	Save (Load) Cancel Hel	q

Figure 2 Import Design

4. Hit **f** to see the full view. Click on the icon (red cycle) as shown in Figure 3.

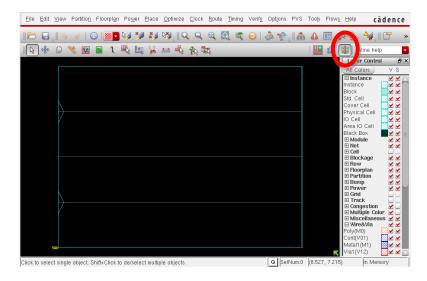


Figure 3 Ready for Floorplanning

5. Floorplanning:

Floor plan→Specify Floorplan. Key in the parameters as shown in Figure 4 and click OK.

asic Advanced		
Design Dimensions		
Specify By: 🖲 Size 🔾 Die/IO/Core	Coordinates	
🖲 Core Size by: 🖲 Aspect Ratio:	Ratio (H/W):	1
	Ore Utilization:	0.6
	Cell Utilization:	0.699702
Dimension:	Width:	9.22
	Height:	6.84
🔾 Die Size by:	Width:	29.22
	Height:	26.91
Core Margins by: 🧿 Core to IO Bo	oundary	
Core to Die B	loundary	
Core to Left: 10	.0 Core to Top:	10.0
Core to Right: 10	.0 Core to Bottom:	10
Die Size Calculation Use: 🥥 Max	: IO Height 💿 Min IO	D Height
Floorplan Origin at: 🛛 💿 Lowe	r Left Corner 🔾 Cen	ter
		Unit: Mic

Figure 4 Floorplanning

6. We are going to add 2 supply rings for VDD and VSS.

Power→Power Planning→Add Ring. Key in the parameters as shown in Figure 5 and click **OK**.

Netto	VDD VSS				1
Ring Typ					
	ring(s) contou	-	Alexa 1	10. h	
	ound core bo		 Along I 	/O boundary	
	clude selecte ring(s) aroun	· ·			
	ach block	u			
-	ach piock ach reef				
	elected power	domain/fenc	es/reefs		
	ach selected b			rows	
	usters of sele				
	With shared				
🔾 User d	defined coordi	inates:			MouseC
	Core ring	O Block rin	α		
	nfiguration -		-		
	Тор:	Bottom:	Left:	Right:	
Laver:	~	10	10	V Netal2 V	4
Width:	4	4	4	4	
Spacing		0.8	0.8	0.8	Update
	Center in	channel) Specify		
Oliset		0.165	0.165	0.165	

Figure 5 Add Power Ring

7. The layout should be the same as shown in Figure 6. It is time to save the work thus far. File→SaveDesign, check Encounter and call it step1.enc Click OK.

Encounter	(R) RTL-to	-GDSII	Save De	sign		/term2/EN - co	unter	_ 🗆 🗙
<u>F</u> ile <u>E</u> dit V <u>i</u> ew Partitio	o <u>n</u> Floorpl <u>a</u> r	ı Po <u>w</u> er	Data Type: 🖲 Encounter 🔾	OA		'S Too <u>i</u> s Flow <u>s</u>	Help C	ādence
 ⊂	0 %/ 1 🖳	. •	File Name: step1 enc		B	≥ 🚴 🔌 ा । 🛄 & (Layer Contr	
			<u>ОК</u> <u>А</u> рріу	Cancel	Help		All Colors	× × ×
			VDD				Cover Cell Physical Cell IO Cell Area IO Cell	
	VSS	4DD		DDV	SSN		Black Box Black Blob Hodule Net Cell Blockage Row Floorplan Partition	
			VDD				⊞ Bump ⊞ Power ⊞ Grid World View	× ×
			V55				World View	
Click to select single object	t. Shift+Click	to de/selec	t multiple objects.		Q Sell	um:0 (-8.472, 18.9	07) In Me	mory

Figure 6 Save Design

8. Before we proceed further, we need to instruct Encounter to connect signals VDD or VSS, and signals that connect to logic high or logic low, to supply ring VDD and VSS respectively. This can be done via the **Encounter-shell** as shown in Figure 7.

After typing and finish the first line of command, hit **ENTER**. To save effort in typing, hit **UP Arror Key** to bring up the previous command for modification. Key in the 4 commands as shown in the figure.

0.000M) *						 	
encounter encounter	2> 3> 4>	globalNetConnect \ globalNetConnect \ globalNetConnect \ globalNetConnect \	/SS /DD	-type -type	pgpin tiehi		

Figure 7 Power Pins Connection

9. Gates are placed in the layout row-by-row. We need to provide supply rails for them. **Route→Special Route.** Key in parameters as shown in Figure 8.

Block Pins 🗹 Pad P	ins 🗹 Pad Rings 🗹 Follow Pins 🗔 Second	ary Power Pins
louting Control		
Layer Change Control Top Layer: Metal5 ►	Bottom Layer: Metal1 >	
🗹 Allow Jogging	🗹 Allow Layer Change	
Area X1: X2: Connect to Targe Delete Existing Routes Generate Progress Me	essages	Power Domain Selection • All • Selected • Named:
Extra Config File:		tra Config Editing

Figure 8 Supply Rails for Cells

10. Perform placement:

Place → Place Standard Cell. The cells should be placed as shown in Figure 9. If not, View → Redraw.

		VDI		
ş	ğ	ont sym cont sym cont men cont wen		
		VD		

Figure 9 Placement

11. Routing

Route → Nanoroute → Route to complete the routing. Place → Physical Cell → Add Filler and select FILL1, FILL1A, FILL2, FILL4, FILL8 as shown in Figure 11, by using Ctrl + Click and Add. This will fill up the empty space.

Add Filler	
Cell Name (6) FILL1 FILL1A FILL2 FILL4 FILL8	(Select)
Prefix FILLER	
Power Domain	Select
No DRC	
Mark Fixed	
Fill Area Draw View Area	
lix liy	
urx ury	
OK Apply Mode Cancel	<u>H</u> elp

Figure 10 Add Filler

12. File→Save Netlist.

Navigate to **../RTL** and provide file name of **counter_pnr.v** The netlist will be saved as **../RTL/counter_pnr.v**

Save Netlist - □ × Include Intermediate Cell Definition Include Leaf Cell Definition Netlist File: ./RTL/counter_pnr.v OK Cancel Help		
— N	letlist File	×
Look in: E/DIPLOMA/Year3/EG3010/g	term2/RTL	3 5 10 10 10 10
Compu Name Counter.v Counter.v Counter_synt.v Counter_tb.v Counter_tb.v Counter_tb.v Counter_tb.v Counter_tb.v Counter_tb.v Counter_tb.v	 ∧ Size Type 231tes v File 222tes v~ File 809tes v File 1 KB v File 1 KB v~ File 594 KB v File 	28 Dec7:36:47
File <u>n</u> ame: counter_pnr.v		<u>S</u> ave
Files of type: Verilog Files (*.v*)		Cancel

Figure 11 Save Netlist

13. **Timing→Extract RC** (as shown in Figure 12) to extract R and C of all the wires.

- 3	Extract RC						
Save RC							
🔲 Save Setload to	Save Setload to counter.setload						
📃 Save Set Resista	Save Set Resistance to counter.setres						
🔲 Save SPF to 🛛 co	Save SPF to counter.spf						
🗹 Save SPEF to 🛛	Save SPEF to counter.spef						
RC Corner to Output	typ						
<u>o</u> k	pply	<u>C</u> ancel	<u>H</u> elp				

Figure 12 RC Extraction

14. From the Extracted RC file, a file that describes delay of all the wires can be produced.

Timing \rightarrow Write SDF.

As shown in Figure 13, change to directory **RTL** and provide a file name of **counter.sdf**

Click Save and then OK.

We are ready to simulate the circuit again with the delay of wires included. You can exit all software.

Calculate Delay	
Delay Calculation Option	
🗹 Ideal Clock	
SDF Output File:	
OK <u>Apply</u> <u>Cancel</u> <u>Help</u>	
	SDF File X
Look in: COMPLOMA/Year3/EG3010/	🚺 🛃 🕞 🤄 🗁 🖻 🖪 🔄
Compu Name	Size Type Date Modified
File <u>n</u> ane: counter.sdf	<u>S</u> ave
Files of type: Delay Files (*.sdf*)	

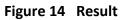
Figure 13 Delay Calculation

14. This is the complete layout produce by the software. File→SaveDesign to counter.enc

Exercise 2 : Post-layout simulation

- 1. We are going to simulate the circuit again but this time with all the delay of wiring (post-layout) taken care of. Refer to Lab5 Exercise 1 if you are not sure about the commands of the simulator.
- Type : cd cd term2 source cshrc nclaunch&
- Select RTL/counter_tb.v and right-click→Edit.
 For the statement "define XXX", replace it with "define PAR". Save and close the file.
 Select counter_pnr.v and counter_tb.v and right-click→NCVlog.
- 4. Select worklib/counter_tb and right-click→NCElab.
 Select Snapshot/worklib.counter_tb:module and right-click→NCSim.
 Select counter_tb and right-click→Send To New→Waveform Window.
 Select count_tb[3:0]. Simulation→Run.
 Zoom in to count_tb[3:0]=F as shown in Figure 14.
 Click on waveform to bring TimerA to the desired location.
 Right-click→Create a marker to put additional marker.

8			Wavefo	orm 1 - SimV	ision				,
<u>F</u> ile	<u>E</u> dit <u>V</u> iew Ex <u>p</u> lore For <u>m</u> at	Sim <u>u</u> lation	<u>W</u> indows <u>H</u> elp						cādence
	💁 🦦 📭 📭 🕼 🗠 🚿 🗅 🛍 🗙 🇊 🗊 📖 📶 🔤 🖉 🥵 😓 🔤 📖 💷 🔳								
Se	Search Names: Signal 🗸 🔽 🆍 🎢 Search Times: Value 🗸 🗖 🎊 🎊								
	TimeA ▼ = 155,000 🔽 ps	- R*- (- 🤄	: 💁 🛛 🖸 🛄		¦ ●) 190,000ps + 0	🛛 Time: 🖁 🖶	154,750ps : 1	155,51 <mark>-</mark> ×
×®	Baseline ▼= 0 H Cursor-Baseline ▼= 155,000ps				TimeA = 155	000ns		Marker 1 = 1	55,290ps
	Name 🕶	Cursor 👻	0ps 154,800ps	154,900ps	155,000ps	155,100ps	155,200ps	155,300ps	155,400ps
\mathbf{H}		1							
₽	🕀 📲 count_tb[3:0]	'h E	Е					F	
	🕂 🕀 👘 i	'd 14	14						
		1							
5	🗄 🔤 width	'd 4	4						



5. As you can see, there is delay of **290ps** between postive-edge of **clk_tb** and counter's output. This is due to the gate delay and wire delay, since now we are simulating a post-layout netlist.



School of Engineering

Diploma in Electronics Computer & Communications Engineering (EGDF01)

EXPERIMENT NO	:	Lab 7 (Duration : 2 hours)
EXPERIMENT TITLE	:	Place and Route of Shifter using Cadence Encounter
OBJECTIVE	:	To perform place and route of Shifter

Exercise 1

To do place and route of the Shifter synthesized in Lab9 using Cadence Encounter

- Open a new Terminal Type : cd cd term2 source cshrc
- Change directory Type : cd EN encounter to launch the software.
- In Encounter GUI, File→Import Design.
 Click Load, select shifter_top.globals and click Open. OK to load the design.
- Floorplanning:
 Floor plan→Specify Floorplan.
 Ratio=1, Core Utilization=0.6, Core to Left/Right/Top/Bottom=10.
 click OK.
- We are going to add 2 supply rings for VDD and VSS.
 Power→Power Planning→Add Ring.
 Nets: VDD VSS
 Width=4, Spacing=0.8, Center in Channel.
 Click OK.
- 7. Before we proceed further, we need to instruct Encounter to connect signals VDD or VSS, and signals that connect to logic high or logic low, to supply ring VDD and VSS respectively.

This can be done via the **Encounter-shell** as shown in Figure 1. Hit **ENTER** if there is no encounter prompt.

```
0.000M) ***
encounter 1> globalNetConnect VDD -type pgpin -pin VDD -all
encounter 2> globalNetConnect VSS -type pgpin -pin VSS -all
encounter 3> globalNetConnect VDD -type tiehi
encounter 4> globalNetConnect VSS -type tielo
encounter 5>
```

Figure 1 Power Pins Connection

- Gates are placed in the layout row-by-row. We need to provide supply rails for them.
 Route→Special Route.
 Nets: VDD VSS
 Top Layer: Metal5
- 9. Perform placement: Place→Place Standard Cell.
- 10. Routing:

Route → Nanoroute → Route to complete the routing. Place → Physical Cell → Add Filler and select FILL1, FILL1A, FILL2, FILL4, FILL8 The complete layout produce by the software is shown in Figure 2. File → SaveDesign to shifter_layout.enc Maximize the layout window. Hit Print scrn to save the screen/result as

shifter_layout.png in the desktop.

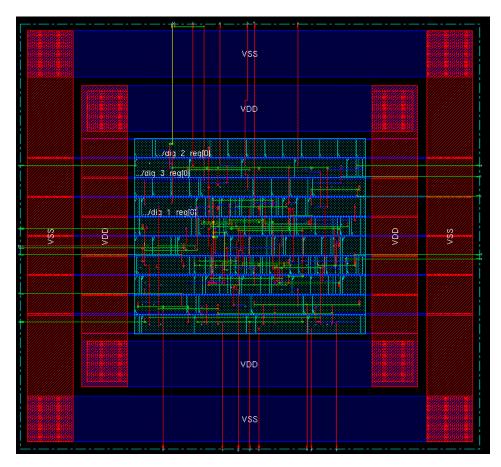


Figure 2 Complete Layout

 File→Save Netlist to save the netlist as ../RTL/shifter_top_pnr.v Timing→Extract RC. Remember to check 'save SPEF ...'.
 Timing→Write SDF. Name the file as ../RTL/shifter_top_pnr.sdf (in /RTL directory). We are ready to simulate the circuit again with the delay of wires included. You can exit all the software.

Exercise 2: Post-layout simulation

- Type:
 cd .. to go up to .../term2.
 nclaunch&
- Right-click→Edit on RTL/shifter_top_tb.v Change the variable to PAR.
 Save and close the file.
- 3. Select **shifter_top_pnr.v**, **shifter_top_tb.v** and **typical.v** and **right-click→NCVlog** Note : This may take a while for the **typical.v** file to be compiled.
- Select worklib/shifter_top_tb and right-click→NCElab.
 Select Snapshot/worklib.shifter_top_tb:module and right-click→NCSim.
 Select shifter_top_tb and right-click→Send To New→Waveform Window.
 Simulation→Run.

Click on waveform to bring **TimerA** to the desired location. **Right-click→Create a marker** to put additional marker.

89 -	🛞 Waveform 1 - SimVision 💶 🗆 🗙										
<u>F</u> ile	<u>F</u> ile <u>E</u> dit <u>V</u> iew Explore Format Simulation <u>W</u> indows <u>H</u> elp cādence										
💁 🤮 🝙 🏤 🔗 🚿 🗅 🛍 🗙 🇊 🟐 📖 🛍 - 🔍 🍢 🏷											
Se	Search Names: Signal - Search Times: Value - Search Times: Value -										
📜 TimeA 👻 = 160,000 🔄 ps 👻 👯 🔸 - 🛨 💁 🗌 🔛 🕶 🔟 🔟 🖤 🔛 👘 🛄 🧼 📾 250,000ps + 0 🛛 Time: 🏤 159,900ps : 160,40 🖬 🔍 = 🔓											
×®	M		,			,	Marker 1 = 160,280ps				
B	Cursor-Baseline - 160,000ps	Cursor 🔻	159,900ps	TimeA = 160,000ps 160,000ps	160,100ps	160,200ps	160,300ps				
	cik	1									
*		0 'h 7	7								
	⊕ ∿ি_ dig_1[6:0]	'h 60	60			(61)	01 OF				
5	⊕ i g_2(6:0)	'h 31	31				20 60				
	⊕ i g_3(6:0)	'h 08	08			(09) (01	<u>X 31</u>				
	±™ segOut[6:0]	'h zz	22								
J		◪◨◪◗◾		40,000 60,000 80,00	<u>, 1100,000 1120,000</u>	140,000 18	0,000 200,000				
0							0 objects selected				



- 5. As you can see, there is a delay of **280ps** between postive-edge of **clk** and shifter's output (when signal dig_1[6:0] finally settled). This is due to the gate delay and wiring since now we are simulating a post-layout circuit.
- 6. Maximize the waveform window. Hit **Print_scrn** to save the screen/result as **shifter_waveform.png** in the desktop.