## ECE 274 Tutorial for Lab1

This Course will use MAX+PLUS II software for Verilog design.

In the first lab, please generate a AND gate step by step follow the following instructions:

1: Open the MAX+PLUS II software and generate a new project - choose File - Project to generate project "andgate"

Project Name	×
Project Name: andgate	
Directory is: z:\ece274\ece561	
Files: andgate.v	Directories:
	Drives:
🔲 Show Only Tops of Hierarchie	es
OK	Cancel

2: Open the text editor and input the Verilog code for AND gate:

*andgate* takes three arguments: two of them are inputs and the third is the output, which is specified in the header of the module.

'~' operator negates the input. '&' operator ANDs two inputs, '|' operator ORs two inputs, while '^' operator performs a logical XOR on two inputs. Hence, ~ ( $a \mid b$ ) implements a NOR function where a and b are taken to be two inputs. Verilog uses C-style comments. '//' marks the beginning of a line of comments. Moreover, anything between /\* and \*/ is also treated as a comment.

*always* is a statement that lets you define blocks of code, in which assignments will only happen as selected signals (specified in the header of the statement) change.

For example:

sets out to d only if the value of a or the value of d changes. Notice a slightly different syntax (we're using the word 'or' as opposed to the symbol '|', and '=' instead of 'assign').

Notice that all signals on the left-hand side of the '=' sign inside *always* blocks must be declared as *reg*, not wire.

If you need to perform more than one operation under the always block, you need to enclose the operations with begin and end.

*always* is most often combined with if (and possibly else if and else) to create powerful sequential logic systems. Study the following example:



3. Save the Verilog code as .v file.

Please pay attention here; the file name must be the same as the module name. So this will be saved as andgate.v

Save As	×
File Name: andgate.v	
Directory is: z:\ece274\	ece561
Files: *.tdf	Directories:
	2:\
	ece274
	Drives
I	
Automatic Extension:	.tdf 💌
ОК	Cancel

4. Use the compiler to compile the code:

<b>1</b>	MAX+plus II	- z:\e	ece274	4\ece561\	. <b>lab1 - [</b> a	indgate.	v - Text	Editor	]	
8	MAX+plus II	File	Edit	Templates	Assign	Utilities	Options	Wind	ow I	Help
	Hierarchy I Graphic Edi Symbol Edi Text Editor Waveform Floorplan E Compiler Simulator Timing Ana Programme Message Pi	Displa; itor Editor (ditor lyzer er	y , sor	e(a,b,d	<b>₩?</b> ▲		S 🔊 🥸			

Choose "Processing" – "Functional SNF Extractor"

MAX+plus II -	z:\ece274\ece561\lab1	a Wadaw Val				
	Design Doctor Design Doctor Design Doctor Settings				_	
andgate 🐖 Com	Functional SNF Extractor		_		×	_D×
Con	Optimize Timing SNF Linked SNF Extractor	jic hthesizer	Partitioner	Fitter	Timing SNF Extractor	Assembler
	Fitter Settings Report File Settings		X			
0	Generate AHDL TDO File		50			100
	Smart Recompile Total Recompile			01	1	
	Preserve All Node Name Synonyms			Stob		
Line 8	Col 1 INS (	_				

Click "Start" to run the compiler.

MAX+plus II File Processing Interfaces Ass	ign Options Window Help	K E E	
Compiler			
Compiler Netlist	Databas Builder	se	Functional SNF
Extractor			snf
<u>_</u>	50		100
	Start	Stop	
	MAX+pl	ıs II - Comniler	
	<b></b>	Project compilation was successfu 0 errors	
		OK	
	Ľ	≜N 🗆	

The compiler will show there are no errors if the code is OK.

## 5. Simulation

First, choose the "Waveform Editor".



Then choose "node" and select "Enter Nodes From SNF"

式 Untitled4 - Wave	eform Edito	r				
Ref: 0.0ns		<b>+ +</b>	ïme: 21.8ns	Interval:	21.8ns	
Name:	Value:	0.0ns 1			100.0ns	200
			Cut Copy Paste Delete Insert Node Enter Nodes from S Enter Text	iF		
•						

Click list.

Enter Nodes fr	rom SNF	×
Node / Group:	×	List
Available Nod	es & Groups:	Selected Nodes & Groups:
b (I) a (I) c (O) ◀		
Type		Preserve Existing Nodes
		Show All Node Name Synonyms
Group	Memory Bit	
	Memory Word	OK Cancel Clear

Click ">>" button. All the input and output pins will be showed in the window "selected nodes & groups".

Enter Nodes fr	om SNF				×
Node / Group:	×				List
Available Nod	es & Groups:		Selected	Nodes & Groups:	
b ( ) a ( ) c (0)			⊨> [] b (I) a (I) c (O)		
		<u> </u>	<=		Þ
Г Туре			🔽 Preser	ve Existing Nodes	
🔽 Inputs	Registered		🗖 Show.	All Node Name Syn	onyms
🔽 Outputs	Combinatorial				
🗖 Group	🗖 Memory Bit				
	Memory Word		OK	Cancel	Clear

Save as "andgate.scf"

Save As	×
File Name: andgate.scf	
Directory is: z:\ece274\	ece561
Files: *.scf	Directories:
	🗁 z:\
	Con ece274
	ecepp1
	Drives:
	포 z: 🔻
· · · · · · · · · · · · · · · · · · ·	
Automatic Extension:	.scf 💌
ОК	Cancel

Change the input signals a and b manually to "a=0, b=0", "a=0, b=1", "a=1, b=0", "a=1, b=1", 20 ns a period.

₩	AX+plu	ıs II - z:\e	ece274	ece561	l\andga	ate - [l	Intitleo	15 - Wav	eform Ed	itor]								
<u>88</u> I	1AX+plu	is II File	Edit V	iew No	de Ass	ign Ut	ilities	Options	Window	Help								
	🖻 🖪	🖨 🕺	<u></u> <b>b f</b>	<b>N</b>	▶?	<u>ک</u> 🔬	5	ء 😕 🕒	🛃 🔝 🛛	a a	🖬 🗐 🖴	A BA		<u>號</u> 昵	<u>a</u>			
	Ref:	0.0ns			+ +	Time:	32.9	Ins		Interva	ıl: 32.9ns							
À					_0.0ns													
Đ	Name	e:	_~	ʻalue: 🚶	Ţ		20.0	)ns	40	).Ons	60	.Ons	80.0	Ons	100.0ns	120.0ns	140.0ns	1
	n)— b			1	<b></b>							1						
	<b>—</b> a			1	<b></b>							1						
	<b>-</b> 💿 o			х	****	****	***	*****	*****	*****	******		*****	*****				*****
•						~~~~	~~~~	~~~~~	~~~~~	******	^^^^		~~~~~	******	~~~~~~~~	~~~~~	*****	*****
Q																		
E																		
-0-1																		
1																		
×																		
Z																		
NK																		
Xa																		
XC																		
XG																		
XS																		

Choose Simulator



Set end time to 100ns and click start,

🚍 Simulator: Functio	onal Simulation	
Simulation Input: Simulation Time: St <u>a</u> rt Time: 0.0ns Use Device Setup/Hol <u>d</u> Check Outputs	andgate.scf 1.0us End Time: ☑ Oscillation ☑ <u>G</u> litch	1.0us
o <u>S</u> tart <u>F</u>	50 Pause Stop MAX+plus II - Simulator	100 Ope <u>n</u> SCF
	Project simulation Circuit stabilized Simulation ended Simulation covers 0 errors 0 warnings	n was successful at 180.0ns l at 1.0us age: 100%

Check the result waveform with the "and" truth table. And show the result to TA.



By doing the design of "andgate" you will become familiar with the software and Verilog language. Please design the OR and Inverter gate using Verilog by yourself. Compile the code and check the waveform. Show the result to TA.