Application of VHDL in EDWinXP – A knowledgebase

Introduction

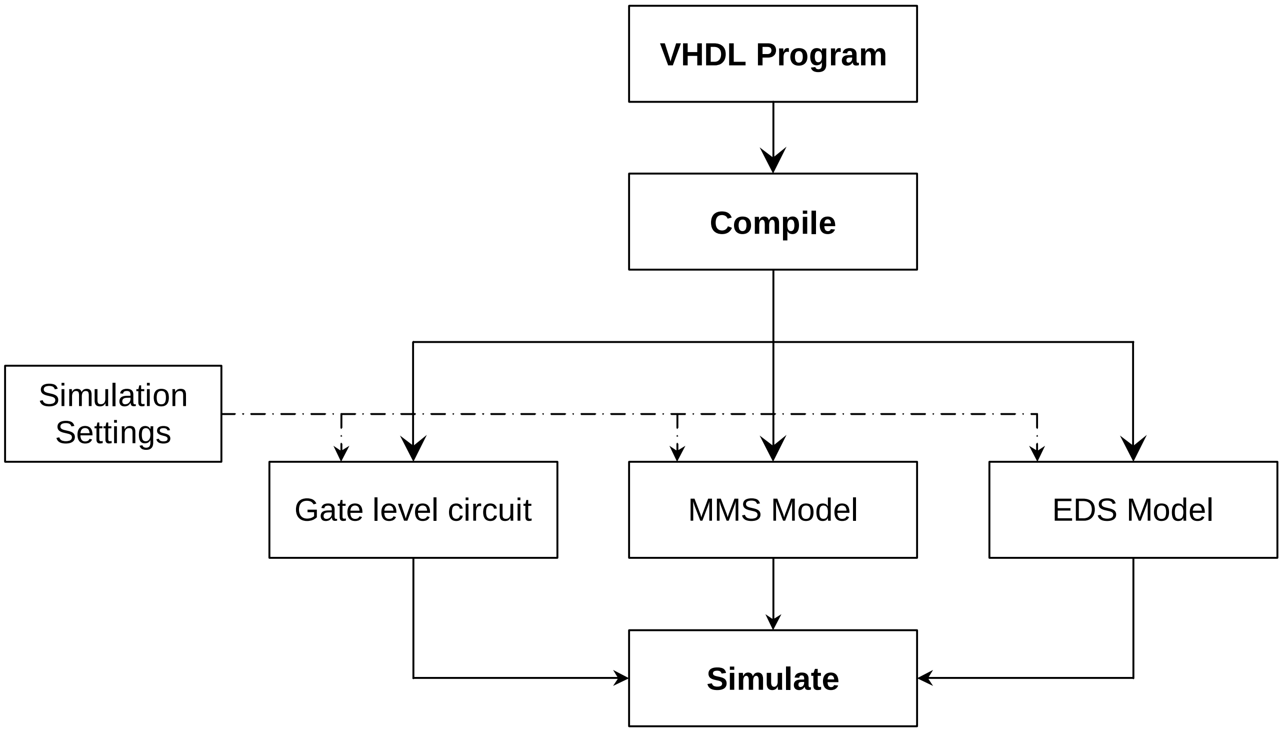
VHDL can be used in EDWinXP for following purposes

1. Creating Schematic diagram from the VHDL program
2. Creating Schematic diagram from the VHDL program and simulating the diagram
3. Creating Mixed mode simulator model[†] and simulating it
4. Creating EDSpice model and simulating it
5. Creation of outputs in *Xilinx* ,*JEDEC* and *CUPL* formats for FPGA/PLD.

In addition to these operations it can also be used for:

6. Hierarchical simulation Using the VHDL program
7. Creating Subcircuit

In this literature, the principle of each topic <u>*in brief*</u> *is explained first and then detailed descriptions are illustrated with pictures and how to do in EDWinXP is also given.*



1. Generation of Gate level circuit

The aim of this is to create the schematic diagram from a VHDL program. The VHDL program written can be compiled to generate EDWin wire list. Hence created wire list can be imported to create the diagram. After importing all the gates (building blocks) will be clustered at (0,0) position of the Page and they can be automatically placed hierarchically using auto placement function.

[†] *Model contains the description of functionality of an electronic device written in high level languages like C++. The model transforms the graphical symbol to an equivalent virtual electronic device that is understandable to the simulator.*

2. **Generation of Gate level circuit and simulation**

The VHDL program once written can be compiled and can generate the equivalent gate level diagram to implement the required logic. Once the program is successfully compiled without any error, this can be imported to create the logic diagram. After importing all the gates (building blocks) will be clustered at (0,0) position of the Page and they can be automatically placed hierarchically. Once it is placed properly, automatic routing function can be used to wire the entire circuit.

The gate level diagram can then be simulated using either Mixed mode simulator or EDSpice simulator since all basic building blocks (gate, flip-flop etc) are simulatable.

3. **Generation of Mixed mode simulator model and simulation**

As did in the earlier case, the program without any error can be compiled to generate the mixed-mode model. After compilation the user has the option to give model number, editing the part etc. The model name also can be specified. Upon "Build" the system will create a simulation function (.dll) in /EDWinXP/Lib and create a Part (optional) under the folder *mmgen.part*. This Part may be loaded (EDWinXP -> Library -> Library Explorer -> Goto MMGEN.PART -> Drag-drop the part to Schematic) to the simulator and it can be simulated by feeding proper inputs. The input/output can be observed in the Waveform Viewer.

4. **Generation of EDSpice simulator model and simulation**

The EDSpice model generation is same as mixed mode model generation. Upon "Build" the system will create a simulation function (.dll) in /EDWinXP/EDS_CML and create a Part (optional) under the folder *edsmgen.part*. This Part may be loaded to the simulator and it can be simulated by feeding proper inputs. The input/output can be observed in the Waveform Viewer.

5. **Hierarchical simulation**

The top most hierarchy can be simulated even though no simulation model is available for it, provided it can be described by lower hierarchies and they are simulatable. We can make a lower hierarchy simulatable by a number of ways viz using the simulatable component, make simulation model for this hierarchy for which VHDL can be used. The model can be created as explained in the topics 3 and 4.

6. Creating subcircuit

A circuit created by means of any of the above explained 5 methods can be converted to a subcircuit. Thus created subcircuit can be used to simulate in any of the SPICE supported simulator without any additional components.

The whole process (1-6) can be explained by using the circuits *full adder* and *half adder.*

1. Procedure for generation of gate level circuit

Write the VHDL program. EDWinXP -> System -> VHDL Editor -> Write the program -> Save. In the build menu -> Compile. Upon compile the system will create a .wrs (wirelist) file. Compile & Import in the *build menu*, will pop up a dialog named 'Netlist / Wirelist Export/Import' -> Click on 'Import'-> The .wrs files will be imported to the page and the required symbols with nets are loaded to the page at (0,0) position. These symbols are then can be placed hierarchically using auto placement function. Then it is wired using autoconnect wire option.

VHDL program for Full Adder

```
library ieee;
use ieee.std_logic_1164.all;

entity FA is
   port (A,B,C:in bit;
   Sum,Carry:out bit);
end entity FA;

architecture F_A of FA is
   begin
   Sum <= A XOR B XOR C;
   Carry <= (A AND B) OR (A AND C) OR (B AND C);
end architecture F_A;
```

Wirelist file obtained when VHDL file compiled

The information written in ***italics*** is the explanations for the wirelist entries.

```
(PATH Generated()
(PORTS                                  description of input out pins
A,DIN                                   DIN-digital input, DOUT-digital output
B,DIN
C,DIN
SUM,DOUT
CARRY,DOUT
)
(COMPONENTS                             components used in the circuit
Comp_1,VHDL_XOR2.VHDL_XOR2
Comp_2,VHDL_XOR2.VHDL_XOR2
Comp_3,VHDL_OR2.VHDL_OR2
Comp_4,VHDL_OR2.VHDL_OR2
Comp_5,VHDL_AND2.VHDL_AND2
Comp_6,VHDL_AND2.VHDL_AND2
Comp_7,VHDL_AND2.VHDL_AND2
)
(NODES                                component connections in the circuit
(UN53                                 Net UN53 connections
Comp_1,I0                             component entries connected to the net
Comp_2,O
)
(C
Comp_1,I1
Comp_6,I1
Comp_7,I1
)
(SUM
Comp_1,O
)
(A
Comp_2,I0
Comp_5,I0
Comp_6,I0
)
(B
Comp_2,I1
Comp_5,I1
Comp_7,I0
)
(UN65
Comp_3,I0
Comp_4,O
)
(UN66
Comp_3,I1
Comp_7,O
)
(CARRY
Comp_3,O
)
(UN67
Comp_4,I0
Comp_5,O
)
(UN68
Comp_4,I1
Comp_6,O
)
)
), Generated                                  end-of-file
```

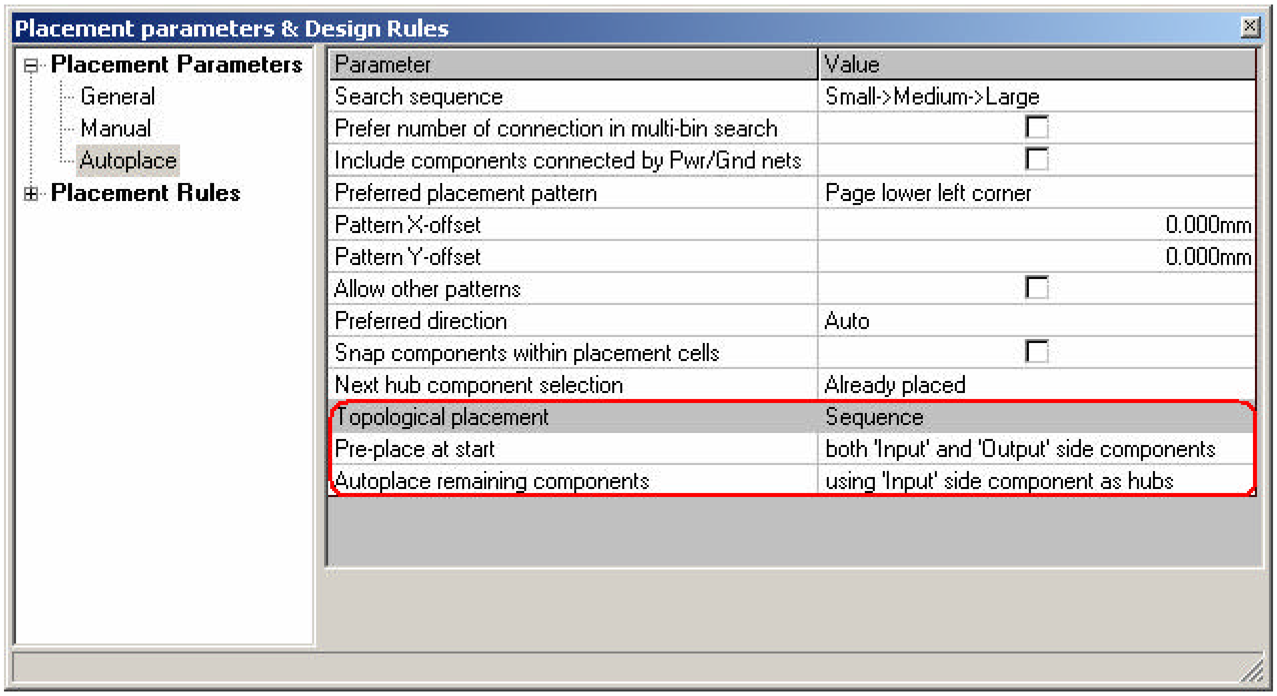
The import option will import the components specified in the wirelist and there will be nets attached to entries as in the wirelist.

Autoplacement & autorouting of components obtained by wirelist import

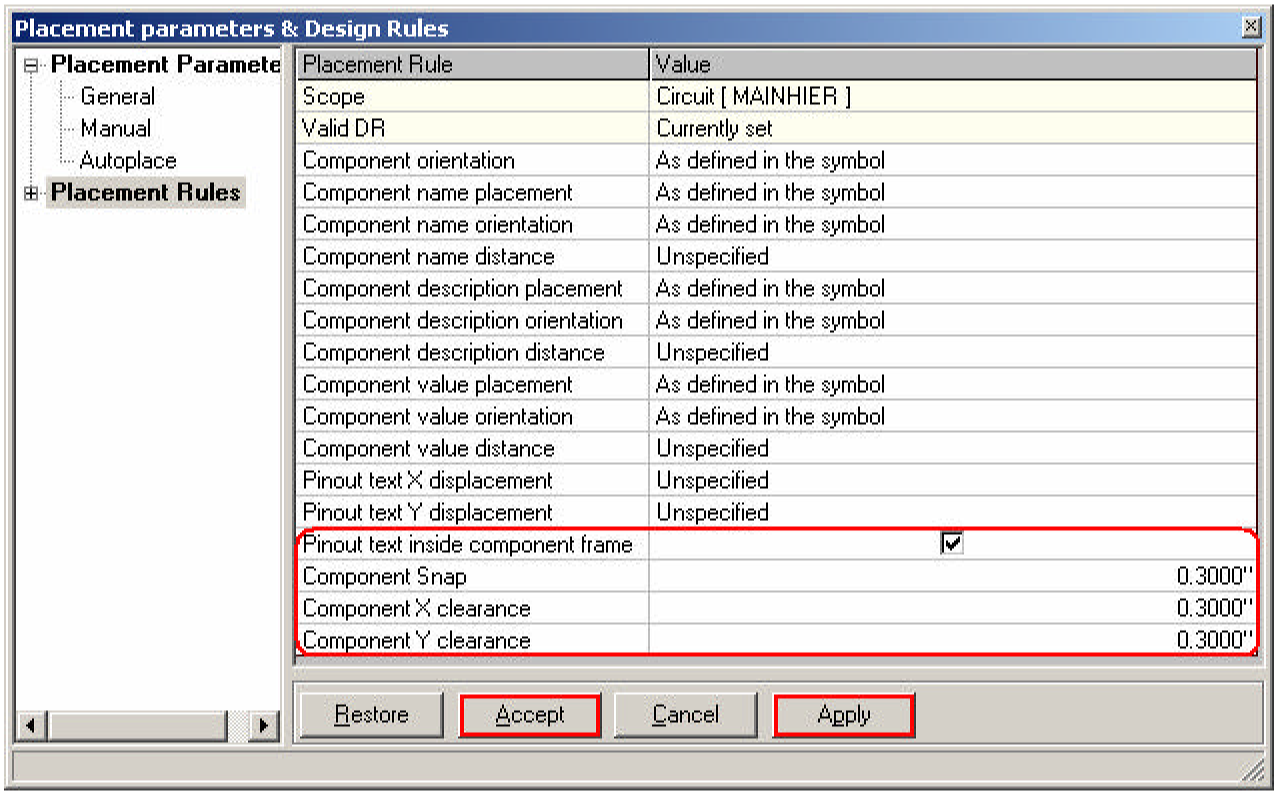
Wirelist import option imports the components with its connections to the schematic editor at (0,0) position. These components can be placed *topologically* (components connected to the input will be

placed at left, components connected to the output will be placed at right) by auto placement function.
Auto place settings

EDWinXP -> Schematic Editor -> Tools -> Auto placement
First function tool 'Auto place parameters' -> 2nd option tool 'Display Parameters' -> 'Placement parameters & Design Rules' dialog will pop up. Do the following settings as shown.



Parameter	Value
Search sequence	Small->Medium->Large
Prefer number of connection in multi-bin search	☐
Include components connected by Pwr/Gnd nets	☐
Preferred placement pattern	Page lower left corner
Pattern X-offset	0.000mm
Pattern Y-offset	0.000mm
Allow other patterns	☐
Preferred direction	Auto
Snap components within placement cells	☐
Next hub component selection	Already placed
Topological placement	Sequence
Pre-place at start	both 'Input' and 'Output' side components
Autoplace remaining components	using 'Input' side component as hubs

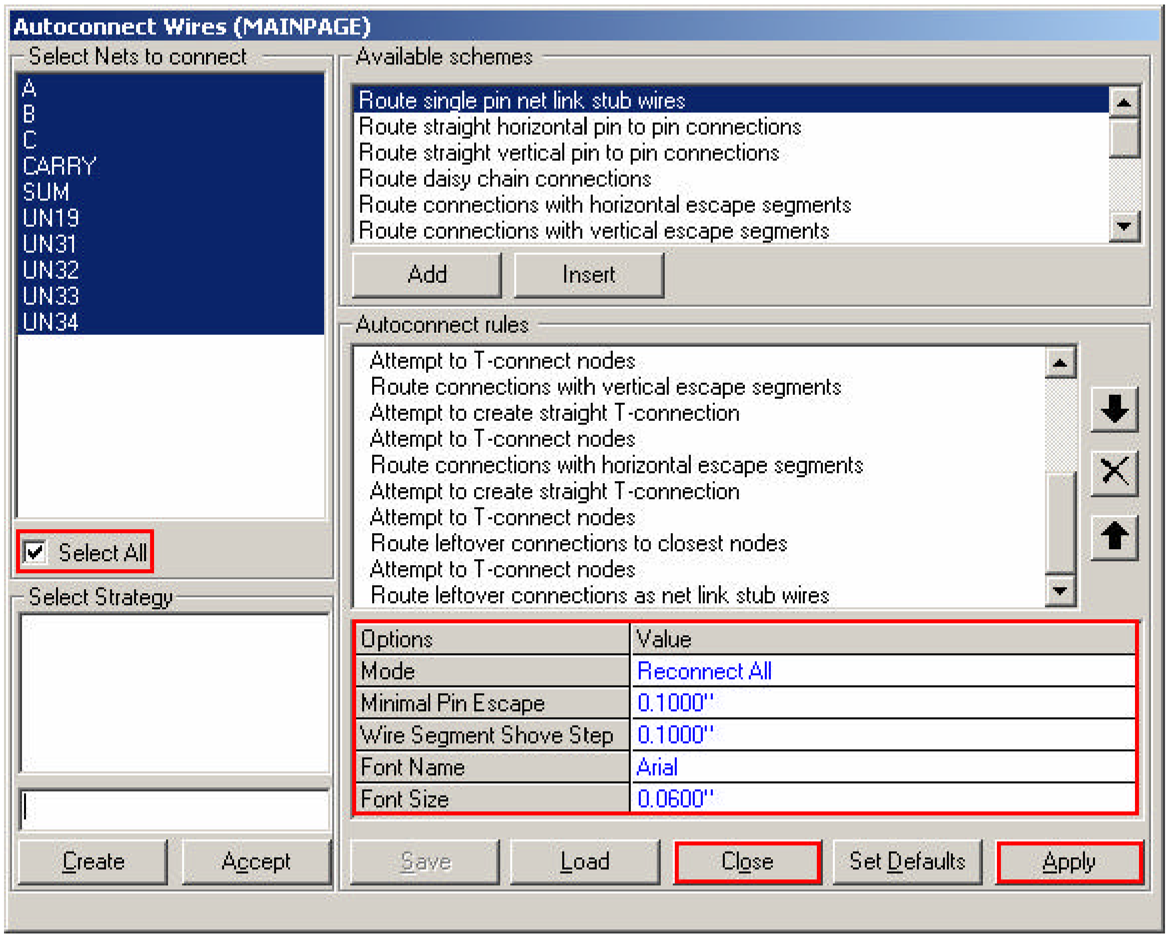


Placement Rule	Value
Scope	Circuit [MAINHIER]
Valid DR	Currently set
Component orientation	As defined in the symbol
Component name placement	As defined in the symbol
Component name orientation	As defined in the symbol
Component name distance	Unspecified
Component description placement	As defined in the symbol
Component description orientation	As defined in the symbol
Component description distance	Unspecified
Component value placement	As defined in the symbol
Component value orientation	As defined in the symbol
Component value distance	Unspecified
Pinout text X displacement	Unspecified
Pinout text Y displacement	Unspecified
Pinout text inside component frame	✓
Component Snap	0.3000''
Component X clearance	0.3000''
Component Y clearance	0.3000''

Function tool ‘Autoplace all components’ -> Option tool ‘Autoplace topologically’. Now all the components will be arranged in the page topologically with the parameters specified.

Autorouting

Function tool -> ‘Reroute connections’ -> Option tool ‘Auto connect wires’-> Do the settings as shown.



Autoconnect Wires (MAINPAGE)

Select Nets to connect

A
B
C
CARRY
SUM
UN19
UN31
UN32
UN33
UN34

[✓] Select All

Select Strategy

Available schemes

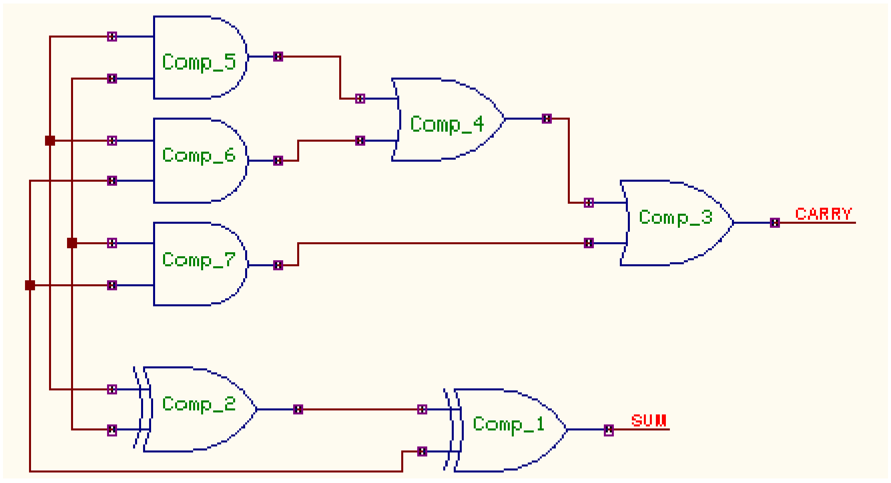
Route single pin net link stub wires
Route straight horizontal pin to pin connections
Route straight vertical pin to pin connections
Route daisy chain connections
Route connections with horizontal escape segments
Route connections with vertical escape segments

Autoconnect rules

Attempt to T-connect nodes
Route connections with vertical escape segments
Attempt to create straight T-connection
Attempt to T-connect nodes
Route connections with horizontal escape segments
Attempt to create straight T-connection
Attempt to T-connect nodes
Route leftover connections to closest nodes
Attempt to T-connect nodes
Route leftover connections as net link stub wires

Options	Value
Mode	Reconnect All
Minimal Pin Escape	0.1000"
Wire Segment Shove Step	0.1000"
Font Name	Arial
Font Size	0.0600"

As the result of autoplacment and autorouting the we will be getting the outputs as shown.



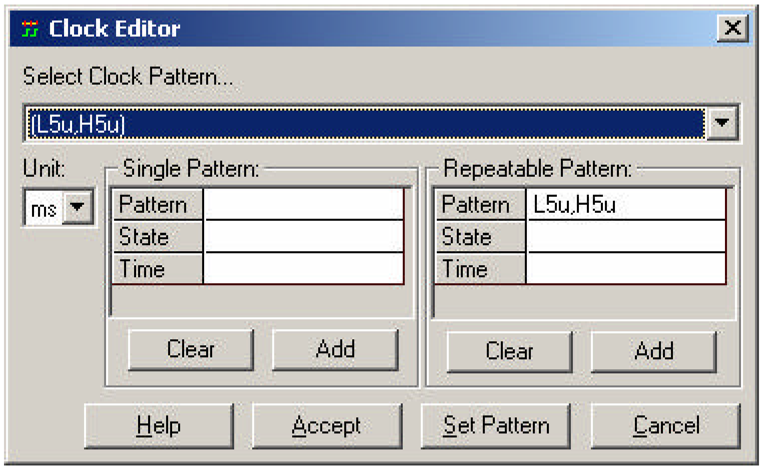
2. Procedure for simulating the circuit obtained through method 1

The circuit obtained through method 1 can be simulated in the following way. As an illustration we will use *mixed mode simulator* for the purpose.

EDWinXP -> Schematic Editor -> Preference -> Mixed mode simulator. The simulation can be preformed by following steps.

Step1: Feeding input signals.

Function tool 'Preset logic states' -> Option tool 'Clock Generator' -> Click on all input nets (here A, B, C) to feed the input clock signals.



Select Clock Pattern: [L5u,H5u]

Unit: ms

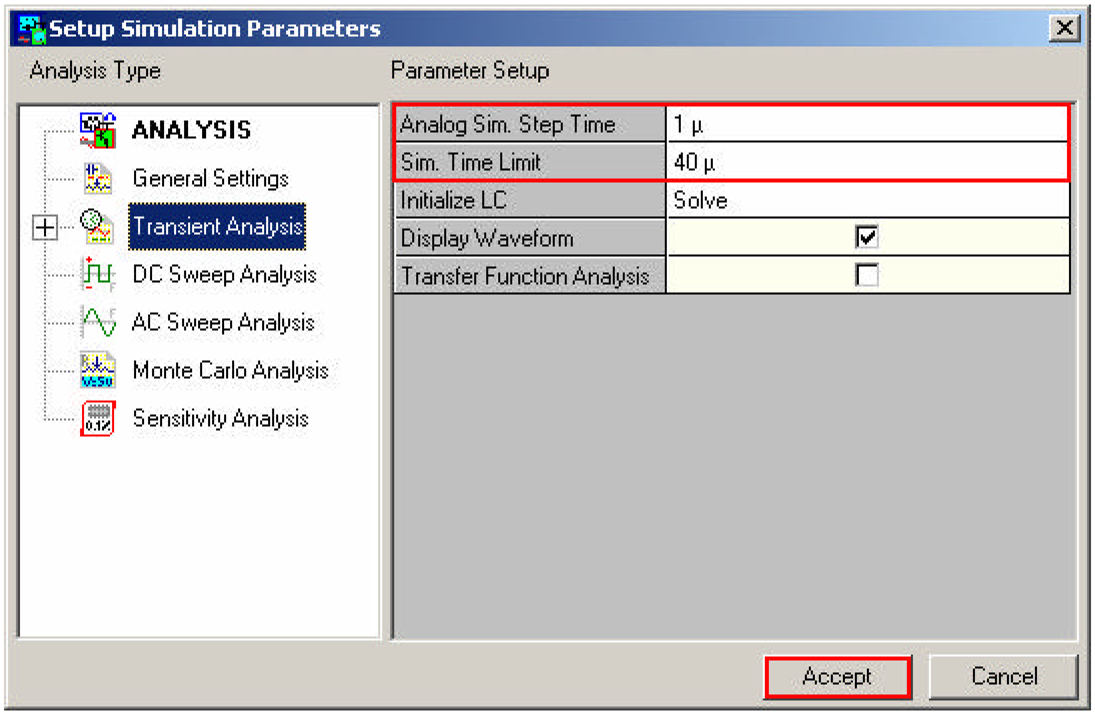
Pattern	L5u,H5u
State	
Time	

Step2: Place the probes to display the waveform at that point

Function tool 'Set waveform contents' -> Option tool 'Logic waveform' -> Click on all inputs and outputs (here A, B, C, Sum and Carry) nets.

Step2: Set simulation parameters

Simulation -> Analysis -> Transient analysis -> Do the settings in 'Setup simulation parameters' dialog -> Accept.


Setup Simulation Parameters

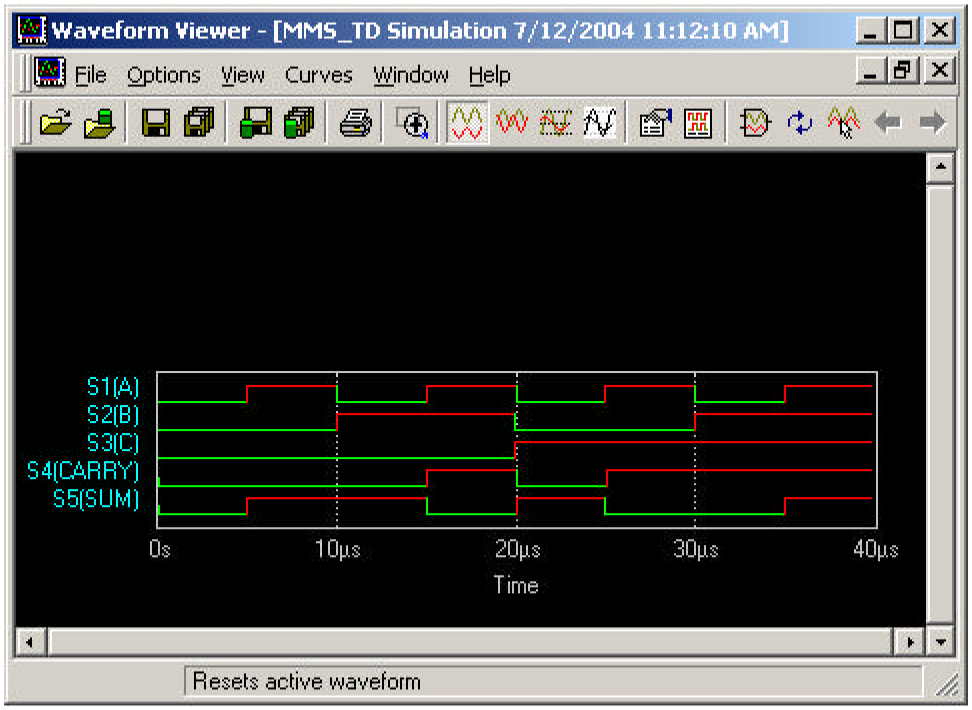
Analysis Type: ANALYSIS — General Settings, Transient Analysis, DC Sweep Analysis, AC Sweep Analysis, Monte Carlo Analysis, Sensitivity Analysis

Parameter Setup

Analog Sim. Step Time	1 µ
Sim. Time Limit	40 µ
Initialize LC	Solve
Display Waveform	✓
Transfer Function Analysis	☐

Step3: Run Transient analysis

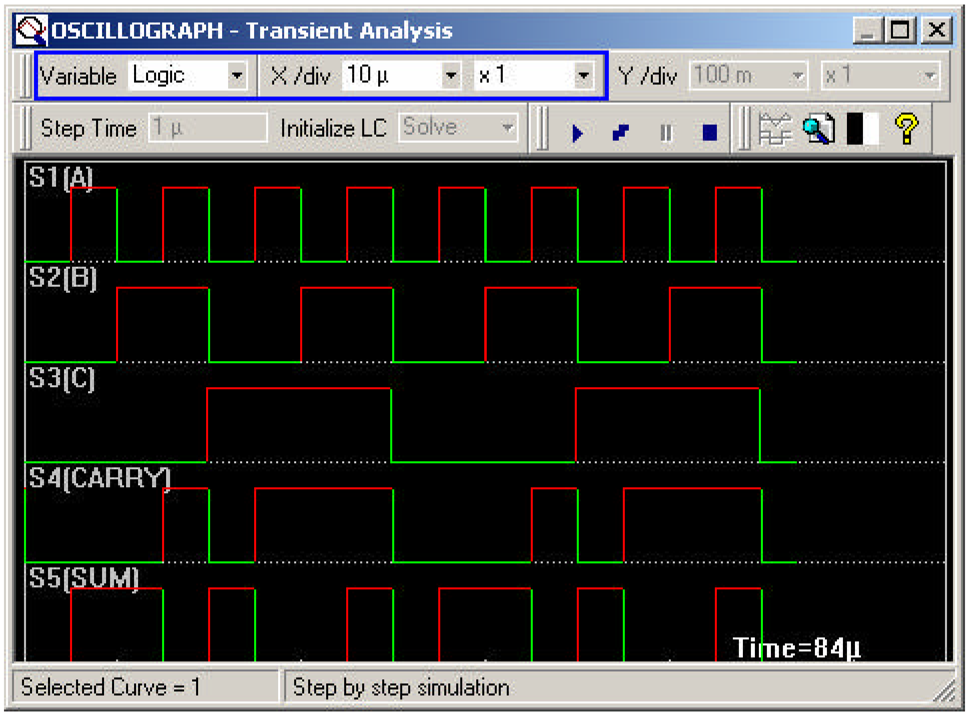
Once the prameters for transient analysis has given Accept -> Run, the simulation results will be displayed in waveform viewer


Waveform Viewer - [MMS_TD Simulation 7/12/2004 11:12:10 AM]

Resets active waveform

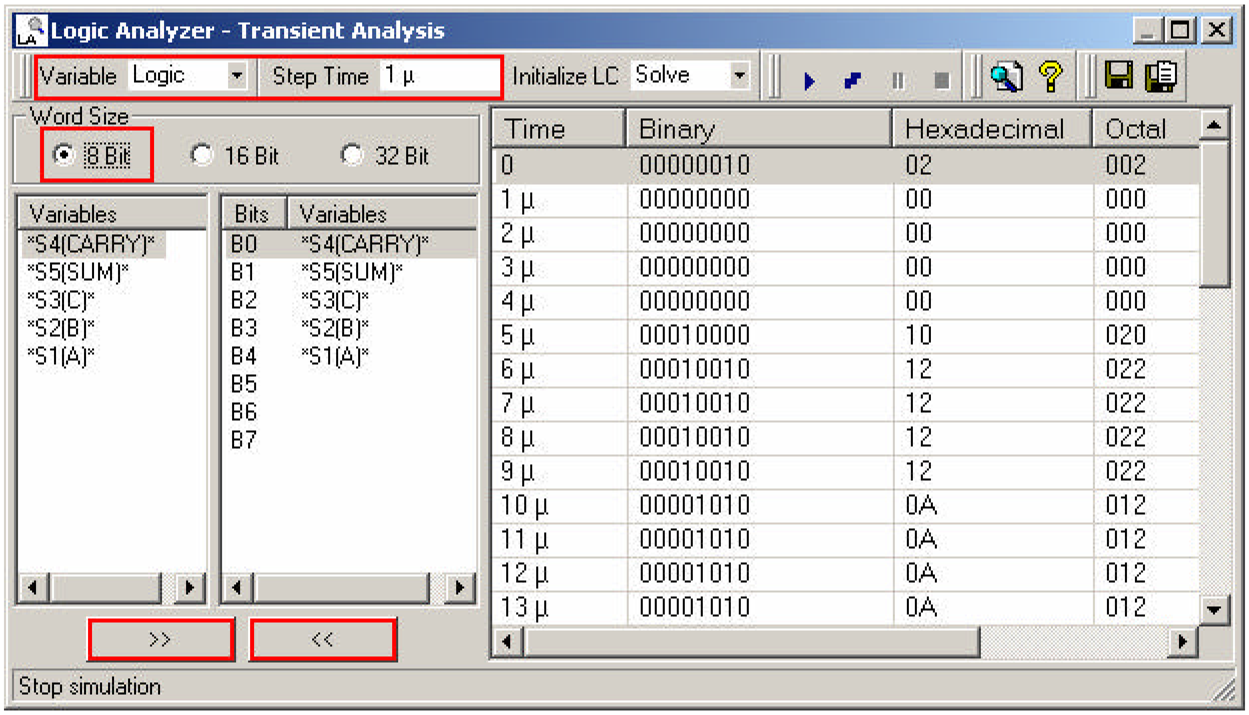
Step4: Run Transient analysis (Oscillograph)

The simulation results can also be observed in the Oscillograph (Simulation -> Run transient analysis (Oscillograph)). The Variable (logic here) and X / div and step has to be selected properly for displaying the result properly.



<u>Step5: Run Transient analysis (Logic Analyzer)</u>

The simulation results can also be observed in the Logic Analyzer (Simulation -> Run transient analysis (Logic Analyzer)). The Variable (logic here) step time and word size has to be selected properly for getting the values at any particular time. The variable required to in the output can be added or deleted using '>>' or '<<' buttons in the dialog.



Variables
S4(CARRY)
S5(SUM)
S3(C)
S2(B)
S1(A)

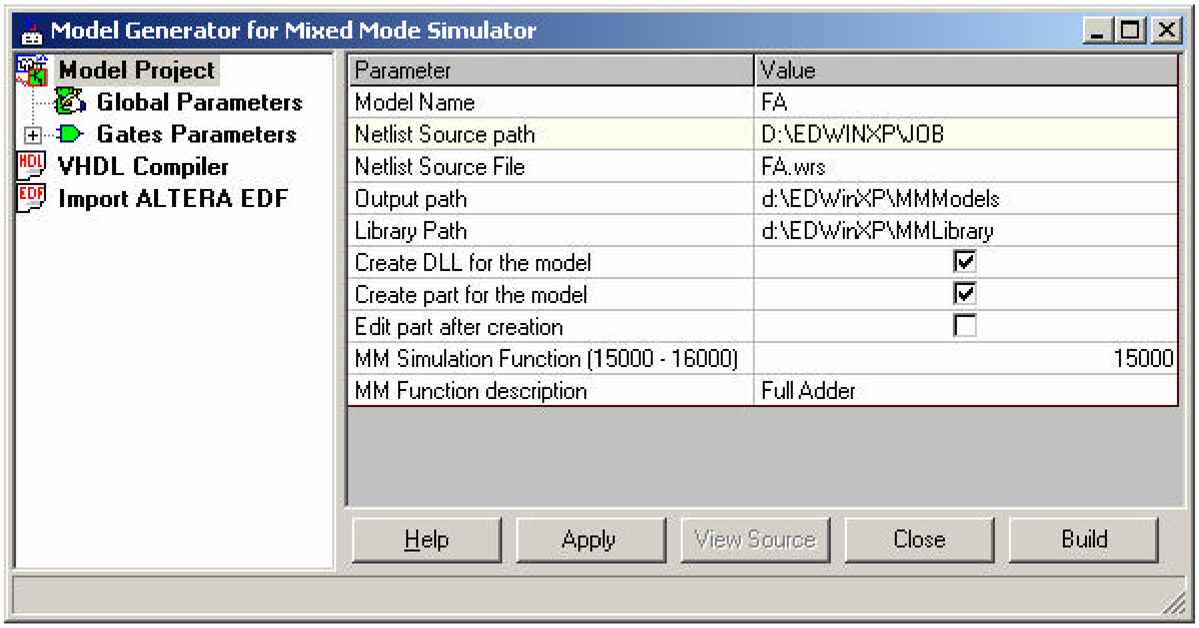
Bits	Variables
B0	*S4(CARRY)*
B1	*S5(SUM)*
B2	*S3(C)*
B3	*S2(B)*
B4	*S1(A)*
B5	
B6	
B7	

Time	Binary	Hexadecimal	Octal
0	00000010	02	002
1 µ	00000000	00	000
2 µ	00000000	00	000
3 µ	00000000	00	000
4 µ	00000000	00	000
5 µ	00010000	10	020
6 µ	00010010	12	022
7 µ	00010010	12	022
8 µ	00010010	12	022
9 µ	00010010	12	022
10 µ	00001010	0A	012
11 µ	00001010	0A	012
12 µ	00001010	0A	012
13 µ	00001010	0A	012

3. Procedure for creating mixed mode simulation model

VHDL Editor -> Build -> Create MM Model -> This will pop up a dialog named ‘Model Generator for Mixed Mode Simulator’.

Various parameter options displayed are illustrated.



Parameter	Value
Model Name	FA
Netlist Source path	D:\EDWINXP\JOB
Netlist Source File	FA.wrs
Output path	d:\EDWinXP\MMModels
Library Path	d:\EDWinXP\MMLibrary
Create DLL for the model	✓
Create part for the model	✓
Edit part after creation	☐
MM Simulation Function (15000 - 16000)	15000
MM Function description	Full Adder

Model name: This is the name of the model which can be set by the user (FA).

Netlist Source path: This will be set to default EDWin/Job path. This can be set in EDWinXP -> System -> Options, if required.

Netlist source file: This is the netlist file which will be converted to mm model. The Netlist file will be obtained by compiling the VHDL code as described in method 1 (FA.wrs).

Output path: This is the path where the mm model will be generated. Default path is /EDWInXP/MMModels

Library path: This is the path of mixed mode model library. Default path is /EDWinXP/MMLibrary

Create dll ad part for the mode: Enable these options to create the simulation dll and equivalent part.

Edit part after creation: This is an option to modify the part once it is created (optional).

MM Simulation function: Each mixed mode simulation function has been assigned with a unique number by which it is identified by the simulator. Models generated using Model generator has been allotted a range of numbers 15000-16000 (15000).

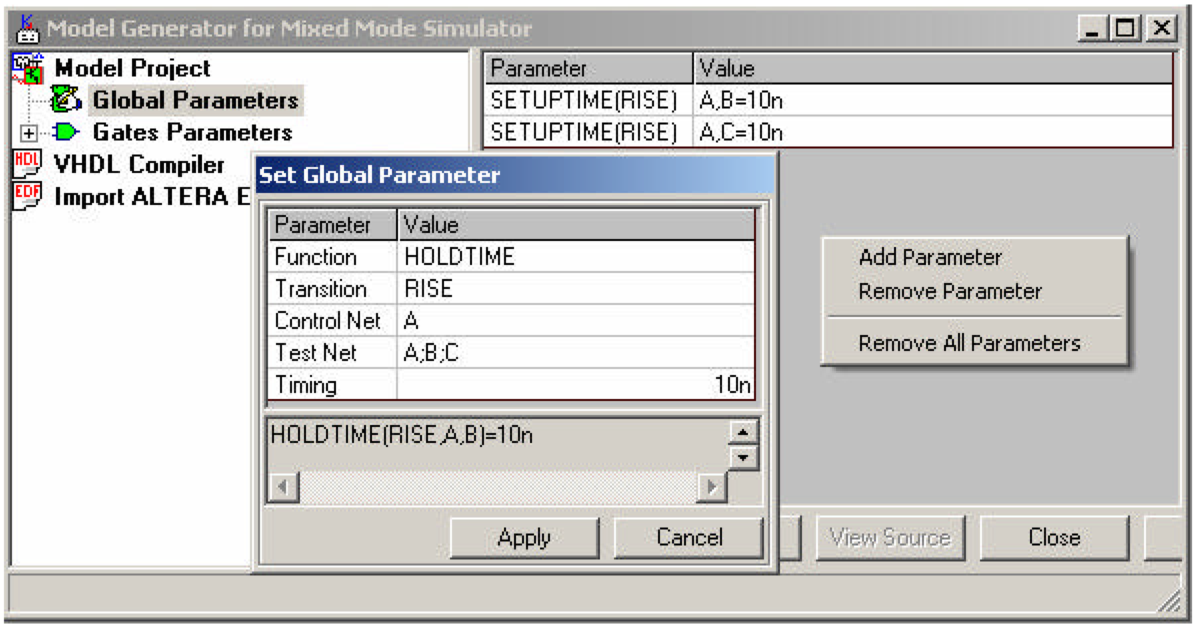
MM Function description: Here the user can enter a meaningful name to the simulation function (Full Adder).

<u>Global Parameters</u>

Right click on the right pane -> Add parameters -> Setup and hold times shown can be specified here. (since full adder is not a clock specific circuit we will not be using these parameters in Full Adder model).

Setup time: The amount of time the synchronous input must be stable before the active edge of clock.

Hold Time: The amount of time the synchronous input must be stable after the active edge of clock.

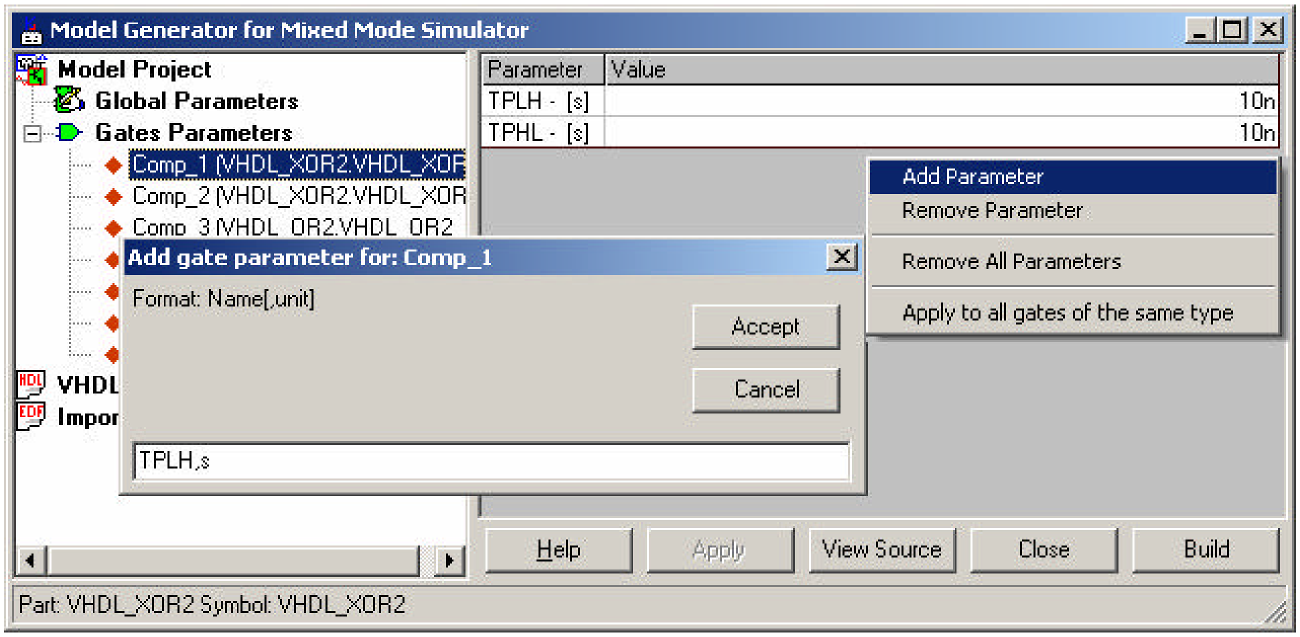


Parameter	Value
SETUPTIME(RISE)	A,B=10n
SETUPTIME(RISE)	A,C=10n

Parameter	Value
Function	HOLDTIME
Transition	RISE
Control Net	A
Test Net	A;B;C
Timing	10n

HOLDTIME(RISE,A,B)=10n

Gate Parameters


Model Generator for Mixed Mode Simulator
Model Project
Global Parameters
Gates Parameters
Comp_1 (VHDL_XOR2.VHDL_XOR
Comp_2 (VHDL_XOR2.VHDL_XOR
Comp_3 (VHDL_OR2.VHDL_OR2
VHDL
Impor
Add Parameter
Remove Parameter
Remove All Parameters
Apply to all gates of the same type
Accept
Cancel
Help
Apply
View Source
Close
Build

Parameter	Value
TPLH - [s]	10n
TPHL - [s]	10n

Add gate parameter for: Comp_1

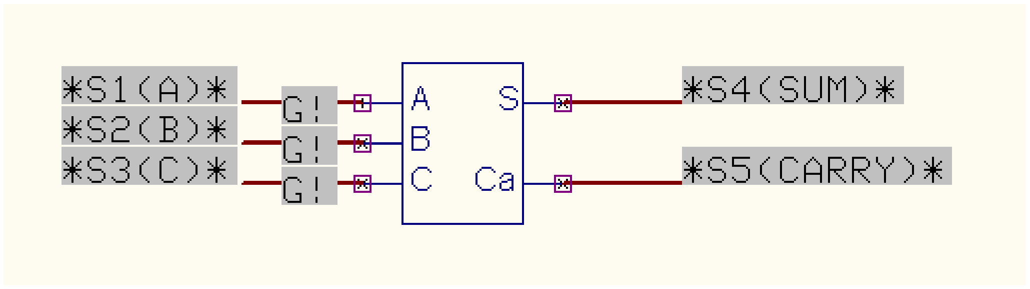
Format: Name[,unit]

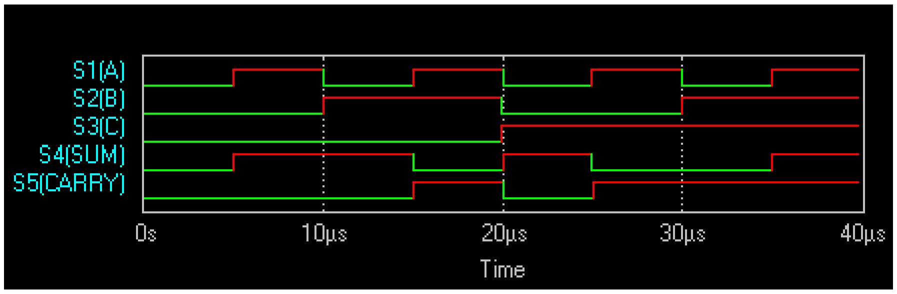
TPLH,s

Part: VHDL_XOR2 Symbol: VHDL_XOR2

Select the gate for which the parameters are to be added -> Then right click on the right pane -> Add parameter -> Type in the parameter (normal parameters to gates are propagation delay L ✍ H and H ✍ L) and its unit (second) -> Now enter the value of parameter. Right click and select the option ‘Apply to all gates of the same type’ to have the same parameters for same type of gates (eg. all 2AND gate). Similarly enter the parameters fro all types of gates -> Apply -> Build. This will create mm model and part in the specified path. The newly created part will be attached with the newly created mm model and hence it can be simulated using mixed mode simulator.

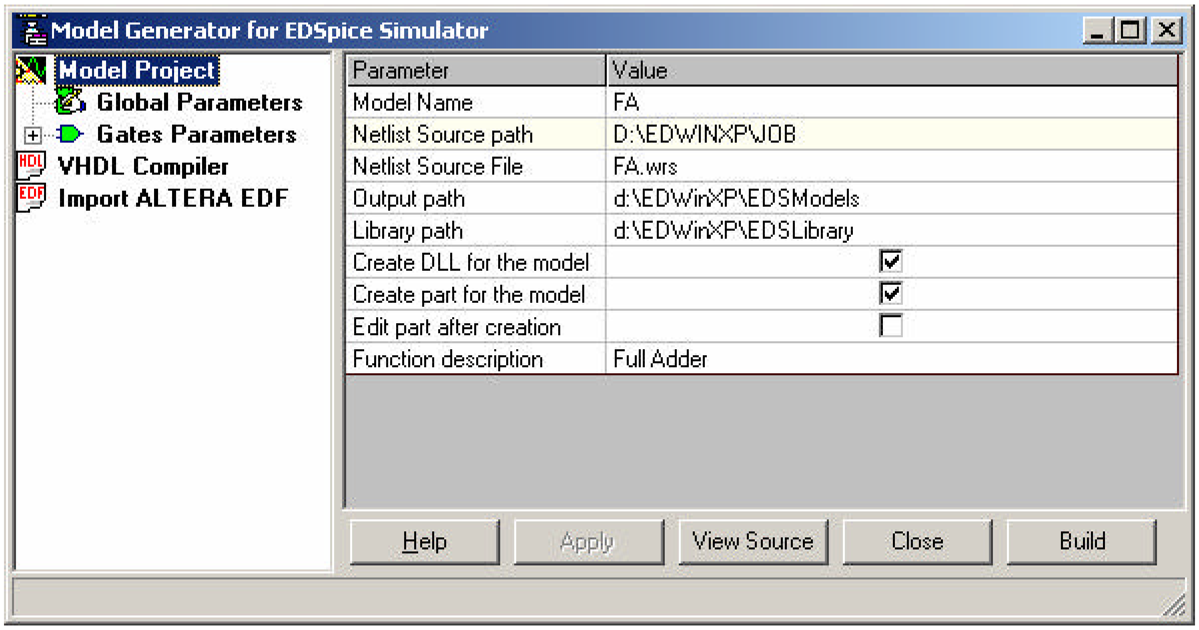
The part thus created can be loaded to schematic and it can be simulated so that it will be working like full adder circuit. So the entire circuit will now converted to a simple block. The settings for simulation has be done as explained in method 2.





4. Procedure for creating EDSpice simulation model

VHDL Editor -> Build -> Create EDSpice Model -> This will pop up a dialog named 'Model Generator for EDSpice Simulator'.



Parameter	Value
Model Name	FA
Netlist Source path	D:\EDWINXP\JOB
Netlist Source File	FA.wrs
Output path	d:\EDWinXP\EDSModels
Library path	d:\EDWinXP\EDSLibrary
Create DLL for the model	☑
Create part for the model	☑
Edit part after creation	☐
Function description	Full Adder

Various parameter options displayed are same as Mixed Mode model generator and only the different options will be explained here.

Output path: This is the path where the mm model will be generated. Default path is /EDWinXP/EDSModels.

Library path: This is the path of mixed mode model library. Default path is /EDWinXP/EDSLibrary

As there was Simulation function in the case Mixed mode model, there is no such item required here.

VHDL Editor -> Build -> Create EDSpice Model -> This will pop up a dialog 'Model Generator for EDSpice Simulator'. Various settings such as Sim function number, description gate propagation delays etc. can be set here. After making settings click on 'Build'. The part if created will be there in EDSMGEN.PART and the simulation function (.dll)will be in EDS_CML folder. All the associate files will be residing under folder EDSModel. *Models.spc* and *elements.spc* will be updated automatically.

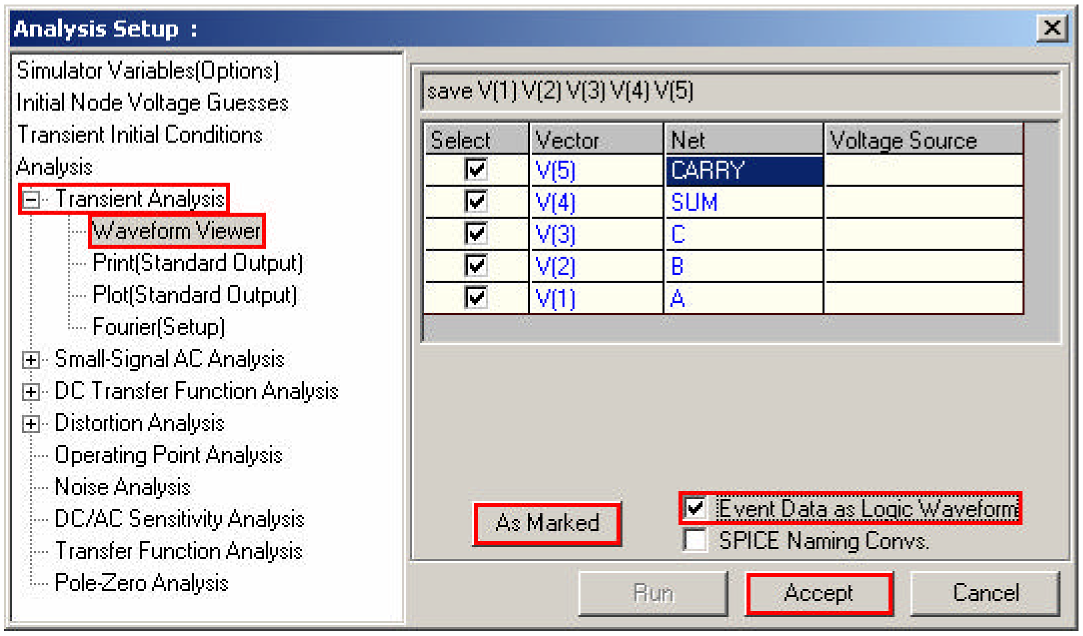
Errors, warnings if any generated will be there in /EDWinXP/Temp/$EDSMcompile.log

Thus generated part can be loaded and simulated (transient analysis) using EDSpice simulator by feeding proper inputs. No need of attaching simulation function to the part as it is done automatically while part creation.

Global parameter and gate parameter can be given as specified in mixed mode model.

Upon 'Build' the system will create the dll and the associate files in the EDSModels folder. It will also update the *models.spc* and *elements.spc* in /EDWinXP/Sys folder.

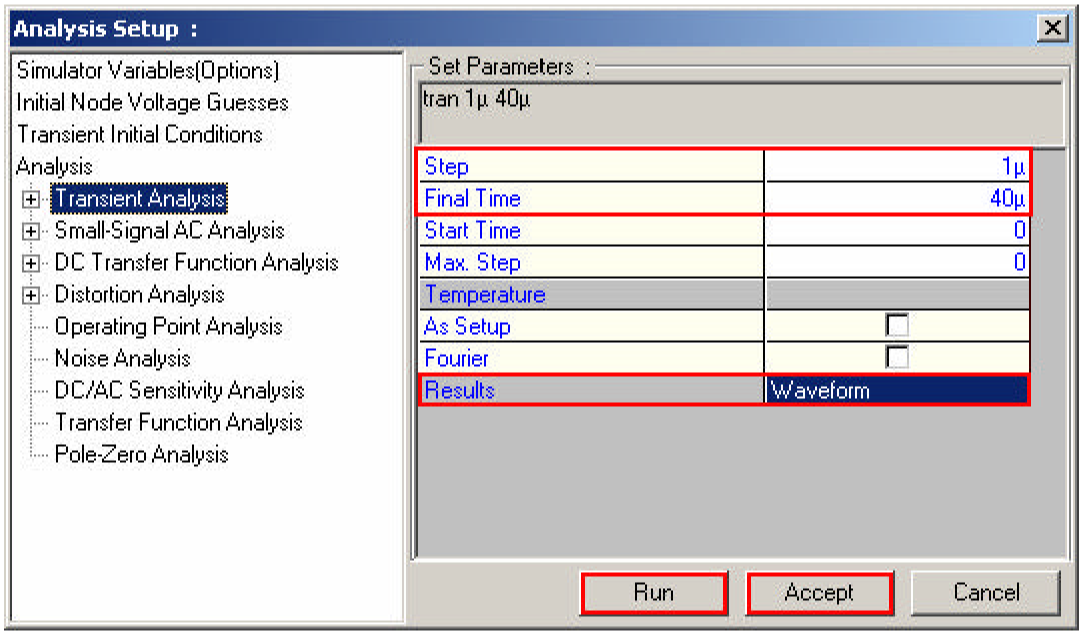
Now load the Part to Schematic and then activate EDSPice Simulator (EDWinXP -> Schematic -> Preference -> EDSpice Simulator) and feed the inputs. Also update the Profile (EDSpice -> Options -> Update Profile Database) files in order to get the latest changes made to *spc* files.



save V(1) V(2) V(3) V(4) V(5)

Select	Vector	Net	Voltage Source
✓	V(5)	CARRY	
✓	V(4)	SUM	
✓	V(3)	C	
✓	V(2)	B	
✓	V(1)	A	

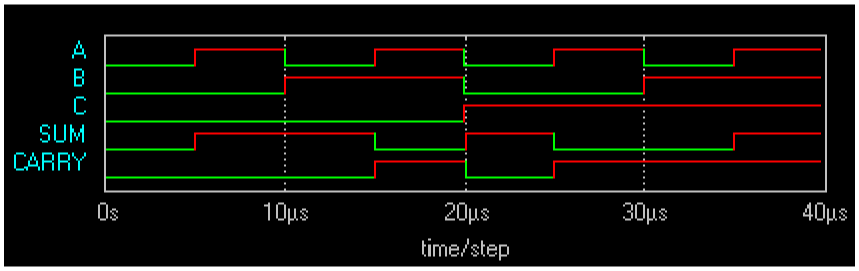
[✓] Event Data as Logic Waveform
[ ] SPICE Naming Convs.


Analysis Setup :
Simulator Variables(Options)
Initial Node Voltage Guesses
Transient Initial Conditions
Analysis
Transient Analysis
Small-Signal AC Analysis
DC Transfer Function Analysis
Distortion Analysis
Operating Point Analysis
Noise Analysis
DC/AC Sensitivity Analysis
Transfer Function Analysis
Pole-Zero Analysis
Run
Accept
Cancel

Set Parameters :

```
tran 1µ 40µ
```

Step	1µ
Final Time	40µ
Start Time	0
Max. Step	0
Temperature	
As Setup	☐
Fourier	☐
Results	Waveform



5. Procedure for creating outputs in Xilinx ,JEDEC and CUPL formats

The VHDL files written without any errors can be converted to Xilinx, JEDEC or CUPL formats. This will be useful when it is required to download the VHDL program to burn a PLD/FPGA.

VHDL Editor -> Build -> Create Xilinx / CUPL / JEDEC Output. Depending upon the output format selected, the system will create .edn (Xilinx), .pld (CUPL), .jed (JEDEC) files.

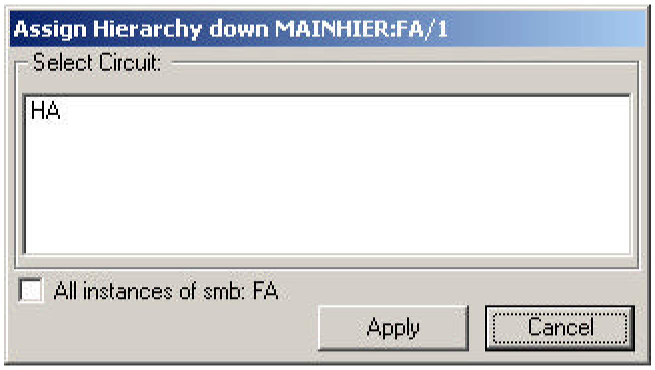
6. Procedure for hierarchical simulation Using the VHDL program

A full adder can be constructed with two Half Adder and an OR gate. If the Half adder and OR gate is simulatable we can simulate the full adder block. For this we don’t need any simulation function for the full adder. This type of simulation is known as hierarchical simulation. ie. We can simulate a block provided its internal building blocks are simulatable.

Here the Full adder doesn’t have any simulation function but its internal blocks are simulatable. ie. We have defined full adder as main hierarchy and half adders are as sub hierarchy. We can create the simulation function for the half adder block as specified in method 3 and 5.

EDWinXP -> Library Editor -> Create the part for Full Adder (FA) -> Load it to schematic editor. This is the main hierarchy of the design. We can wire this FA block in order to give inputs and take outputs. EDWinXP -> Schematic -> Tools - >Connections

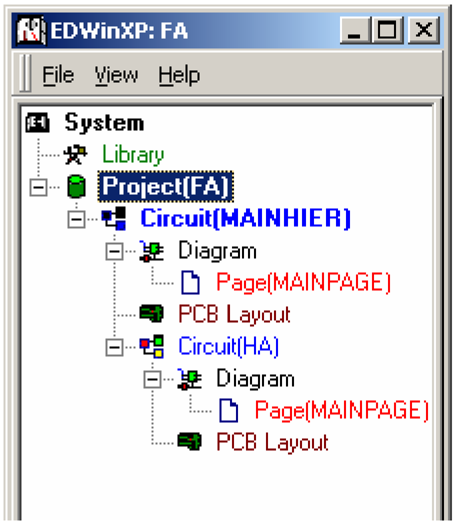
Now we need to create the internal diagram of FA which will be a new hierarchy (circuit). This should be our sub hierarchy. EDWinXP -> Project -> Add Circuit -> Type in the name of hierarchy say HA. Now goto Main hierarchy (FA) -> Schematic -> Tools -> Components -> Function tool 'Hierarchy down' -> Now click on the FA block -> A dialog 'Assign Hierarchy down' will appear, select on the circuit displayed -> Apply. Now this has become the sub hierarchy of FA. This option will also create '*net hooks*' for the pins of main hierarchy which will be acting as the interface between the internal diagram and pins of main hierarchy. Now goto the hierarchy HA and there will be 5 net hooks equivalent to the FA pins. Now create the logic diagram s\using two half adders and an OR gate and connect the net hooks to the pins of this diagram as shown in the project. Now the main hierarchy FA can be simulated either by mixed mode simulator or EDSpice simulator by feeding the inputs and giving the instance parameters as mentioned in method 2 and 3.


Assign Hierarchy down MAINHIER:FA/1

Select Circuit:

HA

☐ All instances of smb: FA


EDWinXP: FA

File View Help

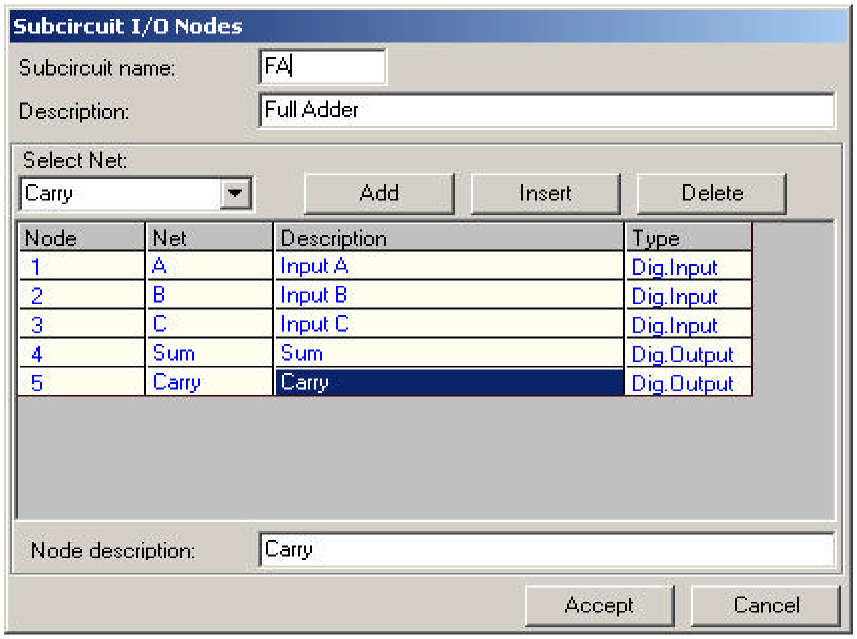
- System
  - Library
  - Project(FA)
    - Circuit(MAINHIER)
      - Diagram
        - Page(MAINPAGE)
      - PCB Layout
      - Circuit(HA)
        - Diagram
          - Page(MAINPAGE)
        - PCB Layout

This is the *logical* hierarchical view of the project. (FA made of two HAs)

7. Procedure for creating Subcircuit

The circuit simulated as per method 6 can be converted to a subcircuit and later this subcircuit can be used to simulate the full adder block. IF we are using the subcircuit for the block, there is no need of internal details as a sub hierarchy.

So first we need to create the subcircuit for the circuit created in method 6. EDWinXP -> Schematic -> Tools -> Components -> Function tool 'Hierarchy down' -> Option tool 'Set subcircuit I/O nodes' -> A pop up dialog 'Subcircuit I/O Nodes' will appear as shown. All the available nets in the circuit will be listed in drop down 'Select Net'. Select the input nets of the circuit and set its properties such as Description and type. Like wise add all input and output nets as shown in the picture. Also set the Subcircuit name and description and Accept. Now EDSpice -> Option s -> Save as Subcircuit -> Select the folder to which the subcircuit has to be saved.


Subcircuit I/O Nodes

Subcircuit name: FA

Description: Full Adder

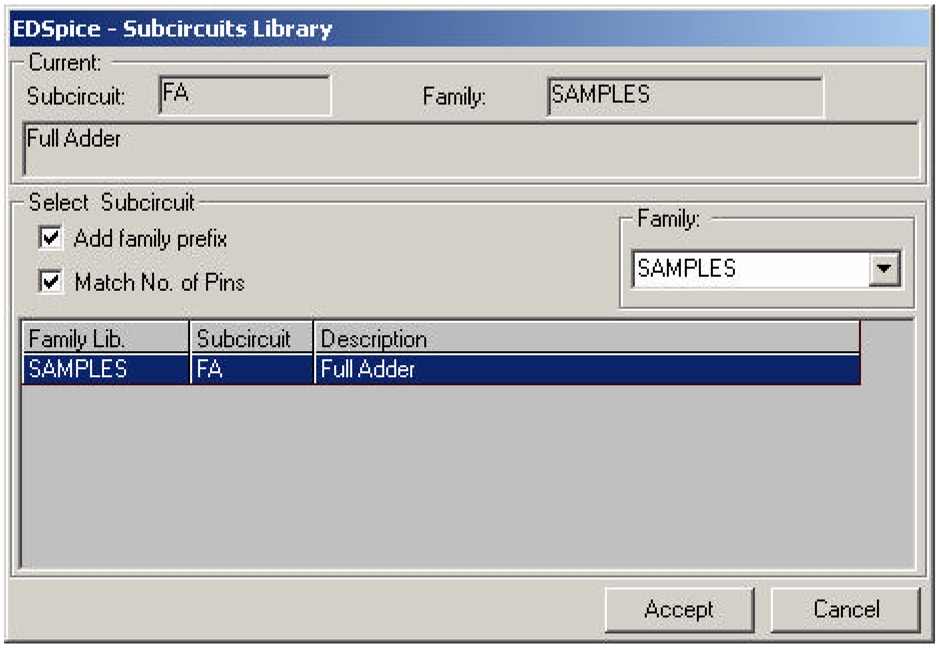
Select Net: Carry

Node	Net	Description	Type
1	A	Input A	Dig.Input
2	B	Input B	Dig.Input
3	C	Input C	Dig.Input
4	Sum	Sum	Dig.Output
5	Carry	Carry	Dig.Output

Node description: Carry

In order to get the updated information we need to update the subcircuit index file. EDSpice -> Options -> Update Profile database. Now the block full adder can be simulated using subcircuit and we do not any simulation information.

Create a Part say FA_SBK -> Load to Schematic -> Preferences -> EDSpice simulator -> Tools -> Instruments -> Function tool 'Component Properties' -> Option tool 'Assign subcircuit'

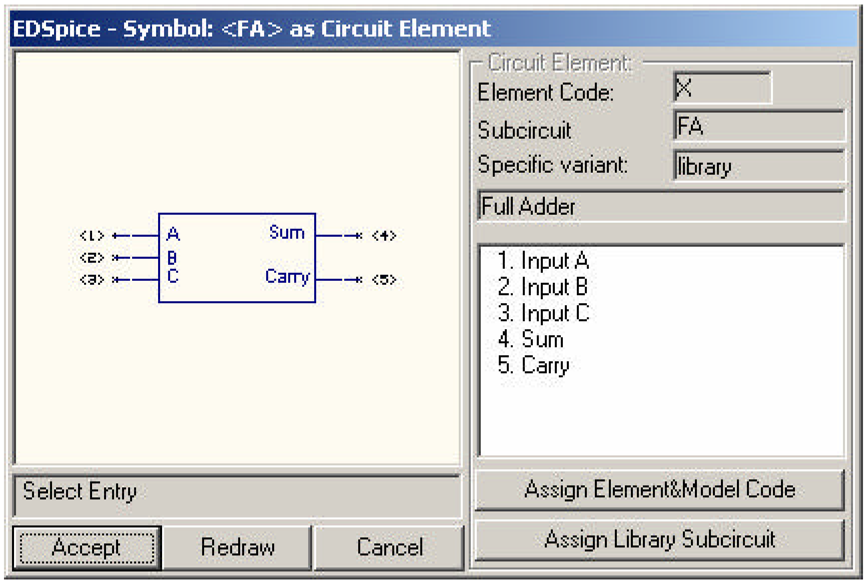


EDSpice - Subcircuits Library

Current:
Subcircuit: FA
Family: SAMPLES
Full Adder

Select Subcircuit
Add family prefix [✓]
Match No. of Pins [✓]
Family: SAMPLES

Family Lib.	Subcircuit	Description
SAMPLES	FA	Full Adder



EDSpice - Symbol: <FA> as Circuit Element

Circuit Element:
Element Code: X
Subcircuit: FA
Specific variant: library
Full Adder

1. Input A
2. Input B
3. Input C
4. Sum
5. Carry

Attach the subcircuit pins to the FA_SBK part as shown. Select the entry on the part fist then click for the list in right side, do the settings for all 5 pins -> Accept. Now this can be simulated as mentioned in method 4.

Praveen kumar E