ELECTRONIC DESIGN for WINDOWS EDWINXP

VISIONICS

CONTENTS

HIERARCHY

Hierarchical Circuits – the concept

Circuits in the project may be arranged hierarchically. More advanced usage of project hierarchical structure would be instances, when the user wishes to accommodate the information about the circuitry for electronic equipment consisting of several PCB within one project database.

Most natural in such a case would be to document the block diagram of the designed equipment as a top (MAINHIER) hierarchy. This hierarchy would contain only schematic diagram where all parts (single PCBs) are symbolically presented as functional blocks. The form and size of these blocks are user defined and their function in the project (database) is dual. Mainly their graphical form and location on the diagram should describe the hierarchical structure of the designed equipment. The other function is to provide handles for accessing individual sub-circuits.

The procedure for creating hierarchical projects (databases) of this type involves as the first step; edition of symbolic parts graphically representing the functional blocks of the designed equipment. Such a symbolic part are created normal part in the project (local) library using functions provided by the Library Part Editor. (Any part, indeed, may be used as a handle to hierarchical sub-circuit).
The parts representing functional blocks may then be placed on the diagram of the main (top) hierarchy page as normal schematic components. Interconnections between the blocks are routed as wires and buses in the same way as when capturing any other schematic diagram

The Add / Delete Circuit item in EDWinXP Project Explorer| Project opens a dialog box to create/ add new circuits. Function tool "Hierarchy down" of Schematic Editor allows to assign sub-circuits to schematic components representing them in the circuits of higher level. This procedure may be repeated on lower hierarchy circuits as long as it doesn't produce a structure which is recursive. In this way the width and depth of the hierarchical structure may be freely controlled according to

particular requirements. For example one component may open the lower hierarchy circuit where some other component open a circuit on even lower level.
The symbol entry names of component associated with sub-circuits are used to name basic nets defining inputs or outputs to/ from the sub-circuits. It happens when a circuit is for the first time
assigned as a sub-circuit to a component. This feature is called net hook-ups and produces automatically a vector of sub-circuit I/O nodes. If an entry in such a symbol is named for example

CLOCK then the net CLOCK will be created in the sub-circuit together with labeled single segment hook-up wire. This provides starting points for routing wires in the sub-circuit with net names consistent with names used in the upper hierarchy. Practically, the hook-ups present then interconnections between circuit and its sub-circuits graphically. The connectivity within hierarchical structure is solved first when the hierarchical circuit is simulated.

Let us consider an example. Suppose you want to design a data terminal system consisting of three PCBs. The basic parts may be described as the input unit (keyboard), the output unit and the central control unit. The input unit inputs data into the control unit and the results are displayed in the output unit. Each of these units has their own complex circuits.

To start with, we create three symbolic parts to graphically represent three functional blocks of this unit (using Library Editor) at place them as components in the circuit at the top hierarchy level (MAINHIER). Names assigned to the entries in these symbols will be used to name the hook-up nets in the circuits at the lower level within hierarchy structure. The next step will be to edit the block diagram of the designed equipment itself. Functional blocks are 'components' on this diagram. Interconnections between the blocks are routed as wires as in any other circuit diagram using functions provided in Schematic Editor.

Dialog box opened by Add/ Delete Circuit item from EDWinXP Project Explorer| Project enables to add the necessary number of sub-circuits.
The depth and width of this structure as well as the purpose of its usage is determined entirely by the user. In another words, user may select which circuit is at

the moment at the top of the hierarchy – by simply selecting it for edition from the Project Explorer. Farther navigation up and down hierarchical tree is done using Hierarchy up and Hierarchy down functions included in the editing features of Schematic Editor as well as Simulators.

How to create a hierarchy?

Hierarchies are a way of splitting up and organizing, large electronic projects into a systematic set of manageable small pieces. Hierarchies are created using the Add Circuit option in Project Explorer. Right click on the task PROJECT and select Add Circuit from the function list to pop up a window.
The process is briefly as follows:
Create a hierarchy by entering its name in the text box 'Create New Circuit' provided and confirm by clicking ADD button. The status bar will display the 'Created O.K' message on completion.
Following the above procedure one can create upto a maximum of 99 hierarchies. The Schematic of each one of the 99 possible Hierarchies can be spread over a maximum of 99 Pages.

Hierarchical Subcircuits

Subcircuits can be of two types - hierarchical subcircuits and library subcircuits. Hierarchical subcircuits are hard coded within the project itself, i.e., the internal circuitry of a circuit element is explicitly created within the project as a different hierarchy. The created subcircuit may still contain circuit element associated with yet another hierarchy. Thus the hierarchy stack may have at the most 99 levels. To simulate such a design, switch to each hierarchy and set the component parameters and the check subcircuit I/O nodes appropriately for each hierarchy. The subcircuit I/O nodes are specified using the menu item Setup/ Subcircuit I/O nodes in the simulator. It is not necessary that the subcircuit should be saved (File/ Save as Subcircuit) which, if done saves the subcircuit as a library subcircuit. Thereafter, it may be used as a library subcircuit also so that it can be assigned to components in any project.

How to Create Hierarchical Subcircuit(s) for EDSpice?

This procedure applies to the Schematic Editor module, the Library Editor and the EDSpice Interactive Module.

- Create the hierarchy from Project Explorer.
- Assign this hierarchy to relevant circuit element in higher hierarchy.
- Create and layout the schematic diagram for the subcircuit ensuring that you use the hookups provided (Schematics Editor).
- If you place the current hierarchy as the top most hierarchy, its circuit may be simulated as the nominal circuit. This should be done to ensure that the subcircuit works as expected. (EDSpice Simulator).
- Check the subcircuit I/O nodes by selecting the option Subcircuit I/O Nodes from the Setup menu of the EDSpice Interactive Module. The I/O nodes should be assigned to the hookups that were generated when the hierarchy for this subcircuit was assigned (EDSpice Simulator).

Please ensure that you specify the subcircuit description, while you are assigning the subcircuit I/O nodes.

- Reset the hierarchy structure to the level where you placed the original component using Hierarchy Down tools.
- Check I/O nodes assignment to symbol entries of this circuit element by selecting Set Parameters/Models button from the toolbar. As you can see the information about the subcircuit you created in step six is automatically displayed. Assign the correct nodes to the symbol entries if necessary and click Accept. You should always preprocess the circuit to ensure correct syntax (EDSpice Simulator).

Subcircuit I/O node

According to SPICE convention, the input and output nodes of a subcircuit are listed in the .SUBCKT line which marks the beginning of the subcircuit in the netlist. The functions of these nodes depend on the way they are connected within the subcircuit.

Whenever a subcircuit stored in the Subcircuit Library is associated with a symbol, its input and output nodes must be assigned to appropriate entries. For example, if a subcircuit executes the logic function of a two input NAND gate, it is necessary to define the input nodes and the output node. The sequence of nodes is irrelevant as long as they are properly assigned to the entries of the symbol, which is linked to this subcircuit.

Consider the line: -

.SUBCKT NAND 1 2 3.

It lists the nodes in the following sequence: -

input 1, input 2 and output

The assignment of the nodes should be made as follows: -

- The dialog window "Subcircuit I/O Nodes " allows to select subcircuit input and output nodes and arrange them in a proper sequence in the node list. Moreover, a short description may be attached to each node to explain its function. This description may be helpful for proper recognition of the nodes when a subcircuit is associated with a symbol. The column Type allows to select the node type for the subcircuit. Digital input and digital output nodes can be selected by cyclically clicking the column. When this subcircuit is used in a circuit, if required EDSpice will automatically insert proper bridges to the nodes after checking this node type.

- The dialog window is invoked by selecting the option Subcircuit I/O Nodes from the Setup menu. The procedure of defining subcircuit's nodes involves the selection of the nets from the drop- down list attached to the dialog window (or from the schematic diagram) and adding them in the required sequence to the table. Please note that a subcircuit name and description also have to be provided.

WORKAROUND - Tutorial

Creation of a half adder

For defining hierarchy, the steps shown below has to be followed

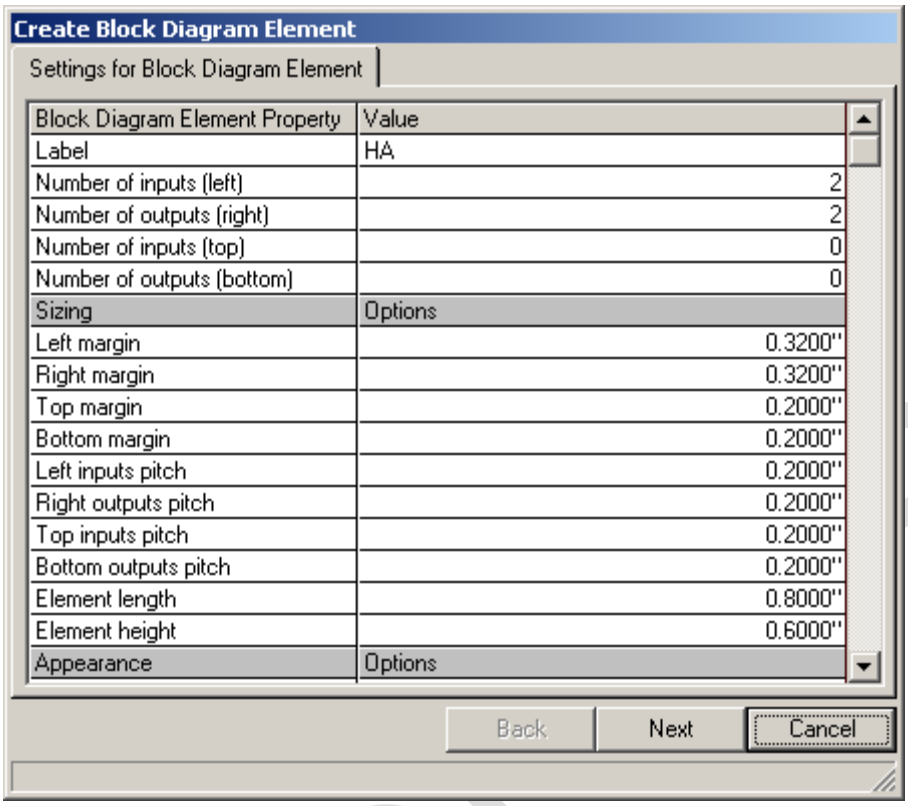
1. Create a Half adder circuit using 7408 and 7432 ICs and wire it as having inputs A and B, and output as SUM and CARRY. Its net names should be as per the inputs and outputs.
   Save this project with relevant information as a circuit file from Project ➔ Circuit ➔Save Circuit ➔HA.epc.

   Let this be the internal circuit for the Hierarchical Main Circuit

2. New Project ➔Schematic Editor ➔ Create a main symbol HA from Schematic Editor ➔ Tools➔ Components➔ Add Block Diagram Elements

   Label : HA
   Number of Inputs : 2
   Number of Outputs : 2

   (eg: a block with 2 input nodes and 2 output node using the Block Diagram editor) in the Schematic page of circuit MAINHEIR



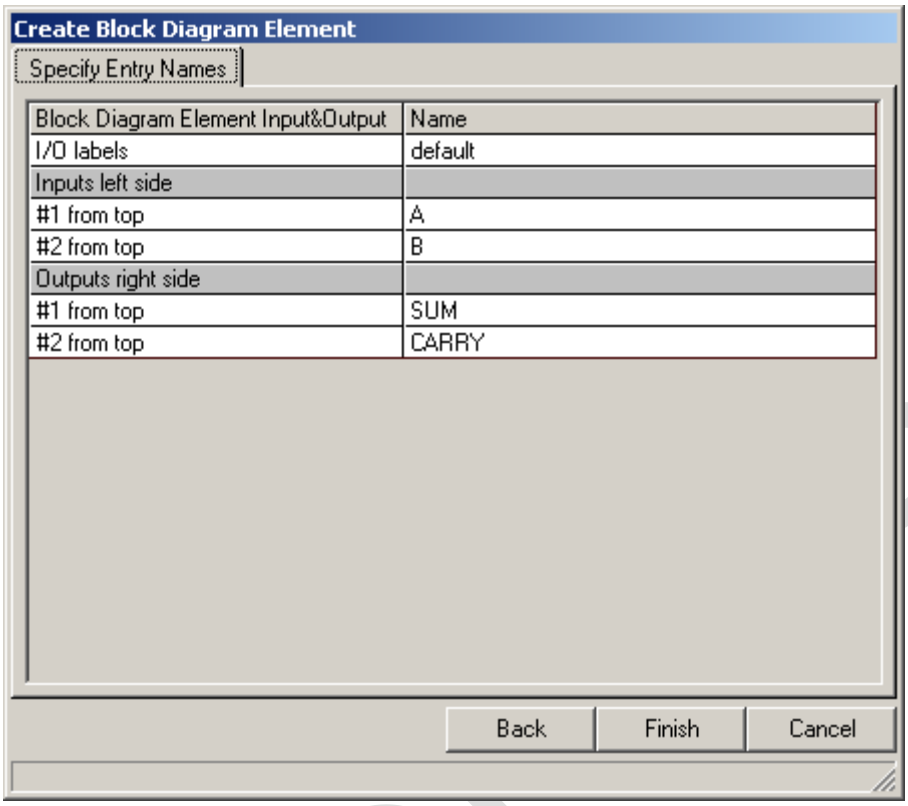
Inputs from left side

- ➔ #1 from top - **A**
- ➔ #2 from top - **B**

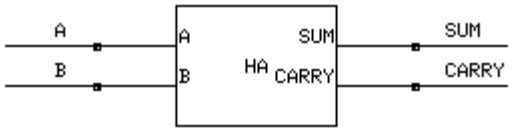
Outputs from right side

Add

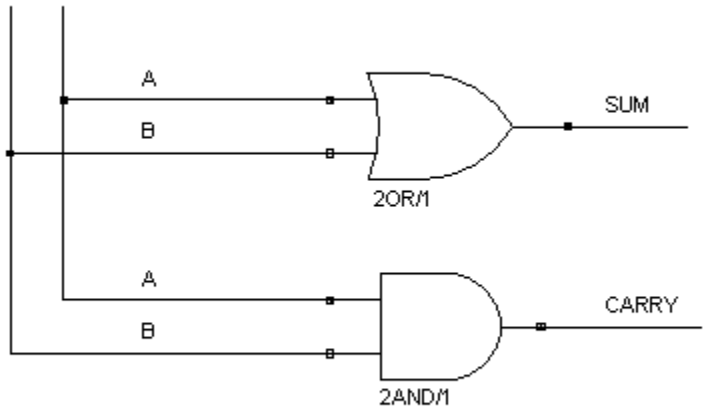
- ➔ #1 from top - SUM,
- ➔ #2 from top – CARRY



Click on **Finish** so that the changes get effected. A block as shown below will be generated.



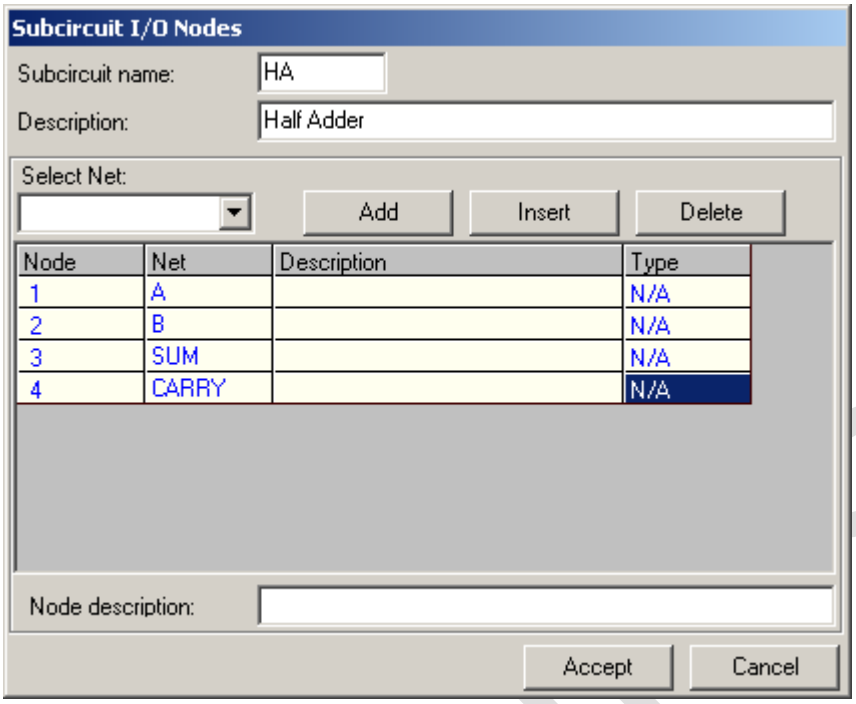
2. Load the internal circuit of the HA in the same project. Let the name of the new circuit is INTERNAL_CKT.



3. The sub circuit I/O nodes for the nets (the name of the nets in the *main circuit* and in the *internal circuit* has to be the same) in the internal circuit can be set next. For this load the internal circuit schematic and set the sub circuit I/O nodes for the nets for which u want to set the sub circuit tags.

This can be accomplished from tools -> Components -> hierarchy down -> Set sub circuit I/O

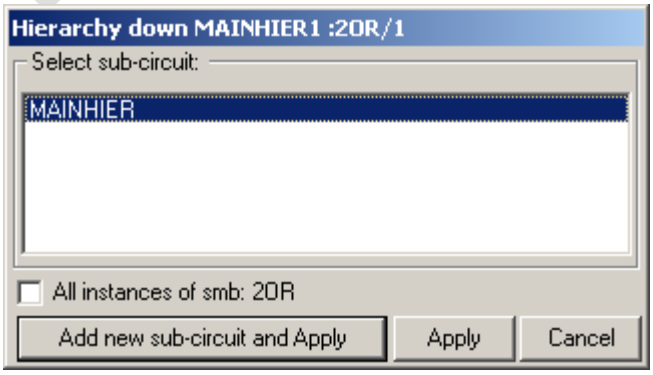
Nodes ➔ Give a suitable name as "Subcircuit name" with Description. Then select net➔Click "Add". Node description can be given to each of the nodes. ➔Accept

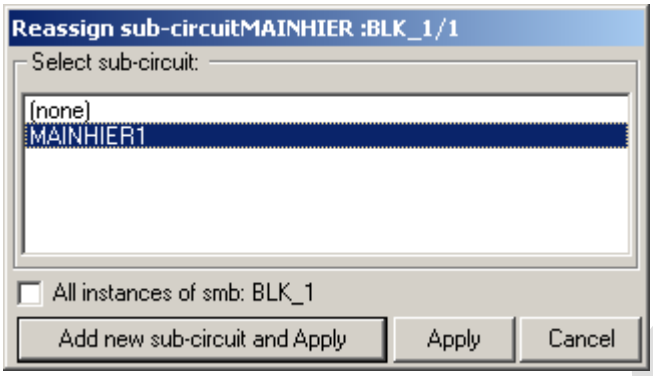


Save this project as a circuit file. This can be done from **Project -> Circuit -> Save Circuit.**

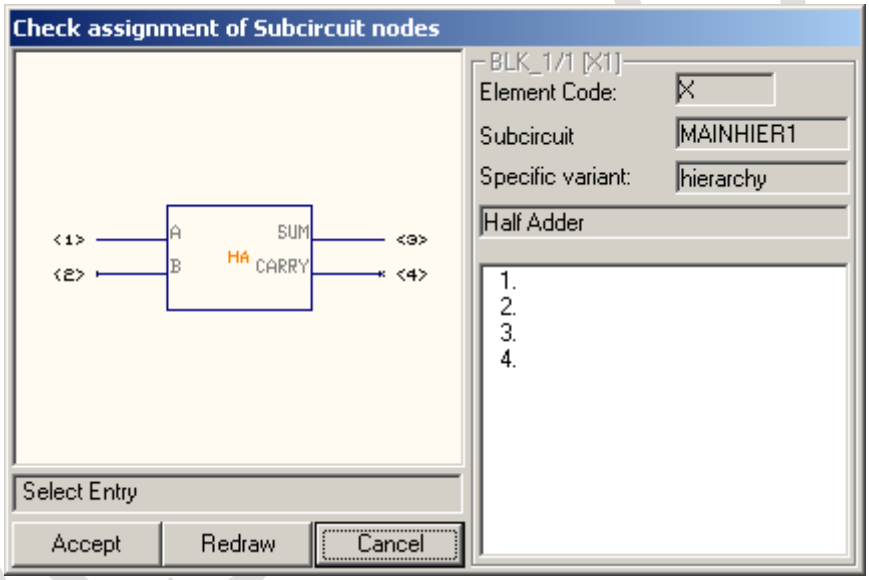
4. Now click on any of the node in sub circuit schematic and enable **Tools ➔Components ➔ Hierarchy Down ➔ Hierarchy up (F1).** Click on the entry points of the pins to which it is assigned .The hookups will appear automatically.

To set





Check assignment of Subcircuit tags



What Are "Hookups"?

When the newly created hierarchy is assigned to a component, the system automatically creates hookup nets in this hierarchy. A hookup is represented as horizontal wire labeled with net name. These hookup nets constitute the input/output node vector of this hierarchical subcircuit. The number of hookups is directly related to the number of entries/ports supported by the component symbol. For example, if you created a hierarchy for a NAND gate then you would expect at least three

hookups (INPUT1, INPUT2, and OUTPUT) that correspond to the entries/ports of the NAND gate symbol

Hookup wires can be moved and rerouted. *The purpose of hookups is to provide informative input and output signals to the hierarchy* (connecting the hierarchy to either the next upper level or lower level hierarchy) and to act as a starting pointing for routing connections on the diagram.