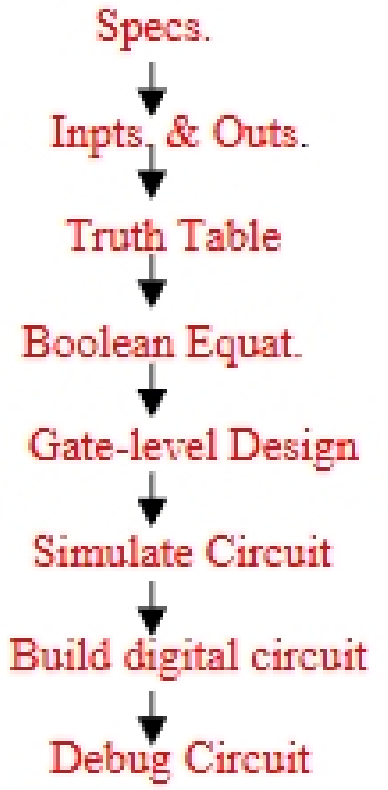
Lab 13: Programmable Logic Design Techniques I

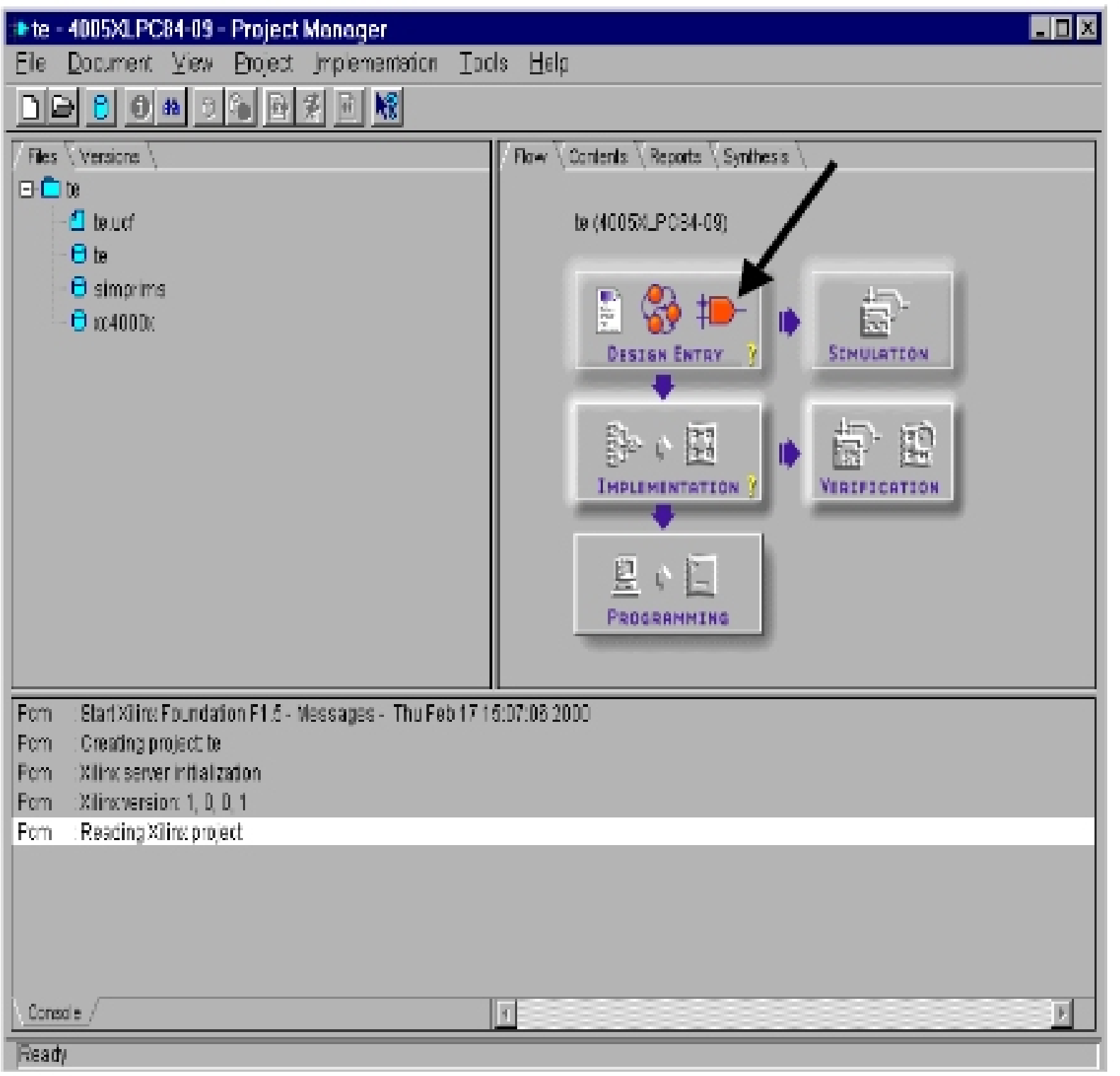
The design of digital circuits is a multi-step process. It starts with specifications describing what the circuit must do. Defining what a circuit receives as inputs and the outputs it generates is the next step. Once these are known the designer has to create a truth table, which lists what values the outputs will have for each possible combination of input values. Once the truth table is written down, the designer has to derive Boolean equations that describe how each binary output can be computed from the binary inputs using logical operations such as AND, OR, NOT etc. Next, the Boolean equations are transformed into a gate-level circuit schematic drawing. Each AND, OR etc. operation in the Boolean equation is replaced with a corresponding AND gate, OR gate etc. in the schematics. Inputs and outputs of these gates are wired to let the passage of binary results between logical operations. Before building the circuit it is a good thing to make sure that all previous steps have been completed correctly. It is done by manual or computer simulation. If successful, a physical circuit is build. Real inputs are applied to the circuit and possible bugs, if any, are fixed.



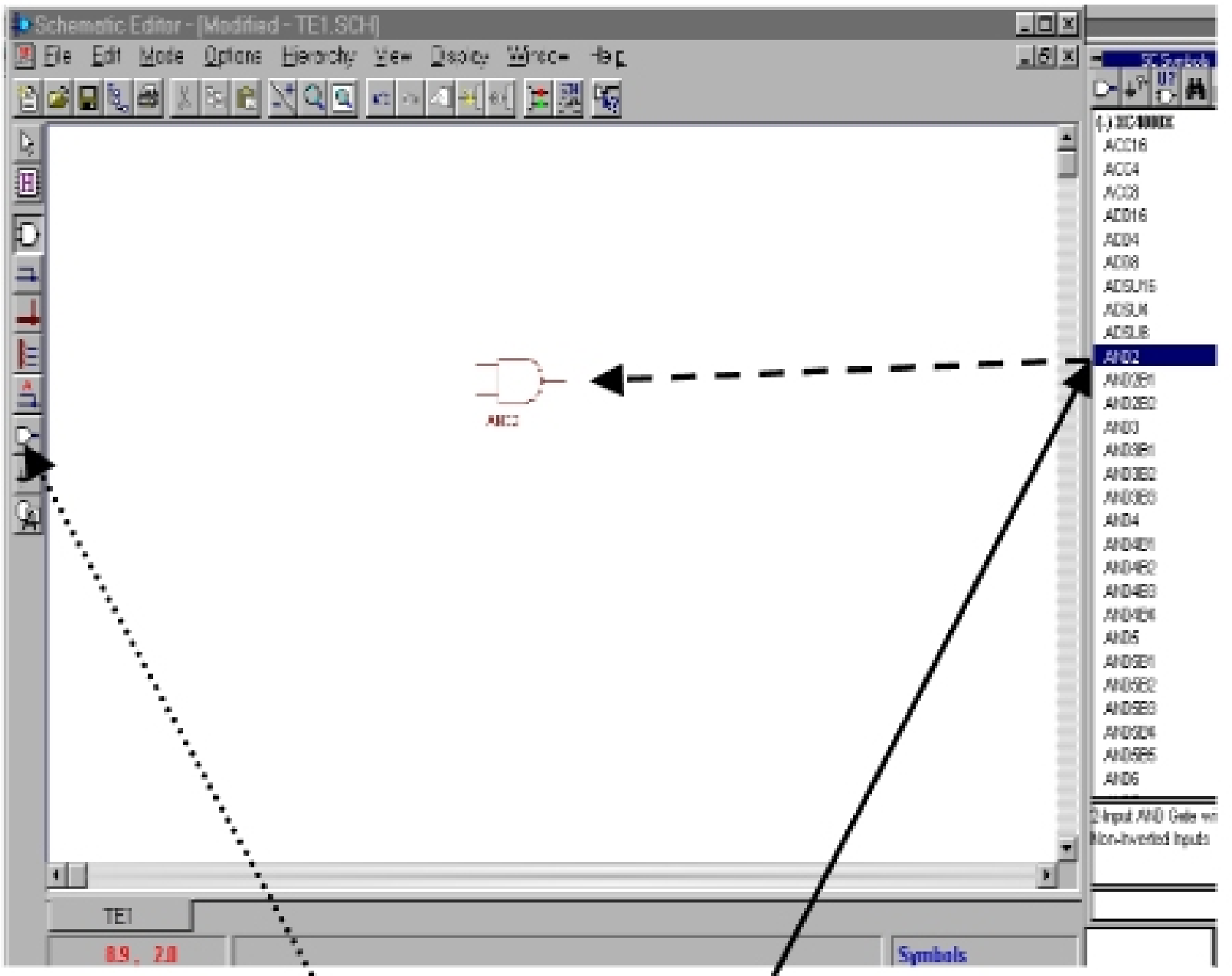
You already built and tested some simple digital circuits using transistor-transistor logic (TTL) devices, and a breadboard with the connections were made using wires. Please, recall Lab 3 and Lab 4 for more details. As you have probably realized, this method of building circuits is not quite convenient: not every type of TTL may be immediately available; wires are often plugged into the wrong place and a lengthy check must be made to find the error; once the circuit works it has to be taken apart to make room for the next circuit; and last but not the least relatively complex digital circuits of more than 10-20 gates are practically impossible to be built and tested. These problems may be eased if a different approach is followed and more elaborated tools employed. The design of digital circuits may still begin by describing the truth table. The details of the logic circuit needed to realize the truth table are, however, worked out by a logic-synthesis program and not by hand. The operation of the "virtual" circuit built is checked using a simulation program. If the circuit simulates correctly, the gates and wires are mapped into a Field Programmable Gate Array (FPGA) using specialized place & route programs. FPGA contains logic gates and the means for interconnecting them within a single Integrated Circuit (IC). The programmed FPGA can be used independently or placed into a larger circuit where it will perform its functions. In this Lab you will be using the XILINX Foundation Series 2.1i software tools to create and test logic designs that can be downloaded into the XC4003E FPGA. *To be successful in this Lab you should review Labs 3 & 4, Chapters 11 & 12 of DH and the Guides to the XILINX software and hardware posted on the PHY440 WWW site.*

Problem 1. Create a digital circuit of a single AND gate and verify its truth table.

To begin, click on the "XILINX" icon. This will bring up the **Project Manager** Window. Select the **File** --> **New project** menu item. Then, enter the project name (on your choice), project directory (keep the default one), type of design flow (Foundation Series 2.1i; Schematic), chip family(XC4003E), chip part number(4003EPC84) and device speed (default). Click **OK** to return to the Project Manager Window.



In the **Project Manager** window click on the **Schematic Editor** button (see the screenshot above) and a schematic editor window will appear (see below).



Select the **Mode** --> **Symbols** menu item and the **SC** window will appear with a list of all type of gates/components we can use. Scroll the list of components in the **SC** window, click on AND2 (two-input AND gate), move the cursor into the drawing area and drop the AND2 gate there. We need to get our inputs and outputs into the circuit as well. To do this click on the **Inputs** button. A dialog window (see below) will appear in which you have to type the name and the type of each input and output.